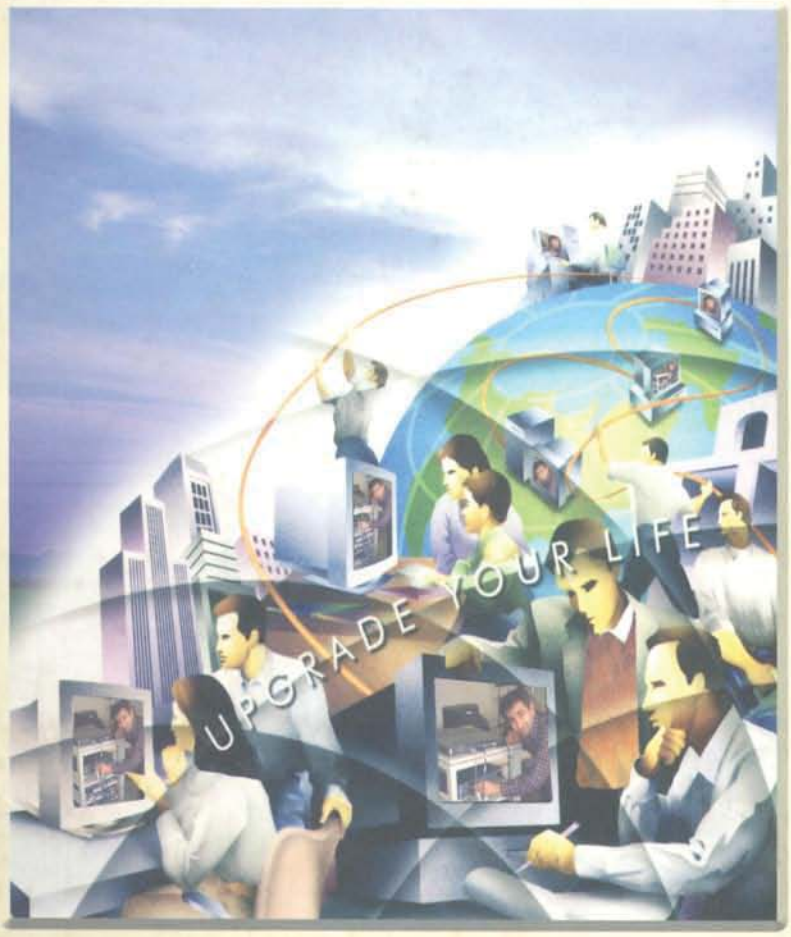
پروتوکلهای TCP/IP

جلد اول



مؤلف: دکتر بهروز فروزان

مترجم: مهندس ادهم صادقی

www.Abiri.ir

به نام خدا

پروتوکلهای TCP/IP

"جلد اوّل"



مؤلف : دکتر بهروز فروزان

مترجم : مهندس ادهم صادقی

www.Abiri.ir

فروزان، بهروز، ۱۳۲۳ – Forouzan, Behrouz A.
پروتوکلهای TCP/IP [تی. سی. پی. /آی. پی]/ نویسنده بهروز فروزان؛ [باهمکاری چونگ سوفیا فگان]؛ مترجم ادهم صادقی.-- تهران : تیزهوشان سرزمین کهن ، ۱۳۸۴ – .
۲ ج.: مصور، جدول.
ISBN 964-96198-5-2:(ج. ۲)
ISBN 964-96198-3-6:(ج. ۱)
ISBN 964-96198-4-4:(دوره)
فهرستنویسی براساس اطلاعات فیپا.
عنوان اصلی: TCP./IP protocol suite, 3rd ed. c2006
۱. تی. سی. پی. / آی. پی. (پروتکل شبکه کامپیوتری). الف.فگان، سوفیا چونگ ، Fegan, Sophia Chung. ب.صادقی، ادهم ، ۱۳۴۶ – ، مترجم. ج.عنوان.
TK ۵۱۰۵/۵۸۵ /ف۴پ۴ ۱۳۸۴ ۰۰۴ /۶۲
کتابخانه ملی ایران ۸۴-۳۵۷۹۶م

نام اصلی کتاب: TCP/IP Protocol Suite
نام کتاب : پروتوکلهای TCP/IP (جلد اول)
مؤلف : دکتر بهروز فروزان
مترجم: مهندس ادهم صادقی
ناشر: انتشارات تیزهوشان سرزمین کهن
تلفن : ۵-۸۸۷۸۳۲۲۴ روزهای زوج، ساعت ۹ الی ۱۳
نوبت چاپ: اول
تاریخ نشر: ۱۳۸۴
تیراژ: ۱۰۰۰ نسخه
قیمت: ۵۰۰۰ تومان
شابک: ۴-۴-۹۶۱۹۸-۹۶۴ (دوره ۲ جلدی) ISBN 964-96198-4-4 (2 Vol. SET)
شابک : ۶-۳-۹۶۱۹۸-۹۶۴ (جلد ۱) ISBN 964-96198-3-6 (Vol. 1)
شابک: ۲-۵-۹۶۱۹۸-۹۶۴ (جلد ۲) ISBN 964-96198-5-2 (Vol. 2)

هرگونه چاپ و تکثیر از محتویات این کتاب بدون اجازه کتبی ناشر ممنوع است.
متخلفان به موجب قانون حمایت حقوق مؤلفان، مصنفان و هنرمندان تحت پیگرد قانونی قرار می‌گیرند.

www.Abiri.ir

تقدیم

این کتاب به دانشجویان عزیزی که در دروس اصول ارتباطات داده‌ها، TCP/IP و CCNA داشته‌ام و همواره منتظر چاپ این اثر بوده‌اند، تقدیم می‌گردد. نام برخی از این عزیزان با رتبه علمی فوق دیپلم، لیسانس و فوق لیسانس تا آنجا که ذهنم یاری می‌دهد در اینجا ذکر می‌گردد:

خانمها و آقایان: محتشمی، خورسندیان، فایند، غلامی جورشری، ماسوله، شریعتی، ایرانشاهی، بهرنگی، غفاری، کریمی، اشتری، طاهری نیک، کیامنش، داداشی، همتی، عبدیان، افتخارزاده، جاهدیان، سوهانیان، وکیلی، نظام آبادی، رسولیان، هادی زاده، درزی، اکبری، کهرکبودی، شمس قصبه، بهروز اقدم، هانی طبایی زواره، منتظری، رنجبر، رضوانی قدیمی، رزاقی، خوش لهجه مطلق، حمدقداری، سنایی، درخشی، میرزاخانی، میرزابیگی، جمشیدبیک، لحافی، ابراهیم بیگی، مکرمی، کاظم‌پور، احمدی، عبدی پور، وافی، ذاکری، محمدیان، خادمی، طحان، قنبریان، غم خواری پیشه، آهنگرانی فراهانی، زندی، شعبانیان، ذبیحی، احمدزاده، اعظامی، ترابی، شمس خامنه، آقامیرزایی، رنجبر، کاشانی، فرجی، مهدی پور، کردبچه، حیدرزاده، اعتمادراد، هدایت نسب، عدالتی عباسپور، تدینی، جهان‌بخش، انوشه، فرخ پی، نوده فراهانی، حاجی کرمی خوزانی، سلیمی، حجتی، ابوالحسنی خواجه، فرهنگ زارع، طاهری مشهدی تفرشی، غضنفریان، خیری فام، مهدوی صفت، باقری، خلفی، حسینی، صادقی نیک، جدی، شالچی زاده، پاشاخانلو، قنبرزاده لک، کیسار، سجادی نژاد، قنادی، صفائی حائری، محمدی، حیدری، عابدین مغانکی، کاظمی، رستم فرودی، مهرزاد، شهباز، نصری، فلاحت‌گر، فنایی، رستگار، حسن نیا، باقری محمدی پور، منصوری، دلیری، خانی، مهیار، بیدار، یوسفی، قبادی، شیروار، رضایی، محمدی مقدم، نویدی، غفرانی، قربانی، زالی، مرادپور هفشجانی، پایدار، ضیاءعلی نسب پور، مرصع فر، امامقلی کرامتی، عاشق حسینی، طاهری، حاصلی، گنجی، حبیبی، طبیب، امیر حسینی، حسینی اصفهانی، سلطانی، جهان‌بخت، نظریان نائینی، نجمی زاده، عزیزخانی، امینی، اسکندری، شیری، روشن فرزاد، شوبیری، لشکری، احسانی، محمدی، خدابنده‌لو، رهبر، رنگین‌کمان، کریمی رادپور، الحسینی، جعفری، نقیبی، بختیاری، بهرامی، نوری، جلالی‌فر، نجفی، عظیمی مهرو، معیاری، کرامتی پور، عطار، سالاری فرزانه، معظمی گودرزی، شیریندخت، عراقچیان، سلیمانی، بنی حسینی، احمدزاده، زارعی، مرادی، حسنی، گل محمدی، رزاق دوست حقیقی، منصوریان، ایزدی، محمودیان، عبدالکریم، جعفریان، کرمی، غفرانی اوچقاز، بابایی پیله‌ور، قزوینی نژاد، زیباکلام، ربیعی کاشانکی، قاطع، خجیر، شکاری، محمدی، غضائری، توفیقی، سمیع زاده، پیرزاده آق قلعه، اسحاقی، حارث آبادی، دارآفرین، شاکوئی، دامک، سمنانی، جعفری فیض، علی محمدی، خاکی، بتوئی، نظری پویا، عظیمی، جمالی، جنابی، شیرازی، طاهری، طاهری شعار، طبایی، کشکولی، مشتاق زاده، معینی شکوه، ونایی، اقبالی،

www.Abiri.ir

هاشمی سجادی، صارمی، ذوالاکتاف، فیاضی، ملکشاهی‌نژاد، حاتمی، باختری توانا، رمضانی، خاتمی، ترحمی، محمدی، پیروزخواه، بشیری، اظهری خامنه، رهنی، یزدانی، عادلی، طرئی، بیدگلی، جام کی، خدادوستان شهرکی، اصغری، عطایی، شیرافکن، معصومی، اردیبهشت، عابدنظری، فرآریا، بردستیان، طباطبایی، اردکانی، ایرانتاج، پوراسکویی، شریف زاده، طایفه، مرادیان، ممدوح، پاکزادکیوی، لتیباری، نوبخت، پورصنعتی، عمویی، لتباری، دارایی، مهرجو، مؤمنی ایزد موسی، احسانی، سلطان محمدی، فرهمند، پورصالحیان، پوریا، صادق عزیزی، برهانی، درخشنده، للهی، شمس، رفضی، رهدار، عباسی، بیرامی، حسینی، حاجی تقی، وطن‌خواه، فاضل‌نژاد، مدرسی، سیدعلیخانی، محمد علی‌پور، زارع، کاظم، مقدم، لطفی، خالق‌پناه، زابلی، ارشدی، اردشیری، قادری، اسدی، باقری‌نسب، عسگری منصورخانی، رفیعی، رحیمی کازرونی، بهادری، گرجی‌پور، محمدی، روزبهی، پوزش، نعمت‌پور، رحیم‌زاده، نیکبخت، ظفرپور، غفاری، قاری، سلیمانی، عجم‌حسنی، موسوی، عیدی، سوداگر، جوادی، هاشمیان، قربانی، نعمتی، ونکی فراهانی، کلانتری، حسین‌خانی، بک مرادی، اصغرپور، نصرالله زاده ممقانی، پورلیمایی، مرادی، افلاکی، اکرمی، پارسایی فرد، طهماسبی، جلیلی، حجازی، رادمنش، اصل‌بند، میرزاد، گل‌خوشبو، بابایی، گودرزی، برنجی، زندیه، ساعدی، خاطری، قاسمی، کربلایی، خداویسی، جباری، میرتقی، بیگی، صوفی، جنوبی، فیضی، قاسمی‌نجم، دادخواه، صاحبی، صلواتی، دل روشن، نادری، طبری، پیرکماجی، مهرجردی، آگاه، کشاورز کیاسرایی، قیومی بیدهندی، طاهری‌زاده، نوفلاح، آزادیان، اصفی عسگرآباد، شادپور کوچصفهانی، فهیمی، اسفندانی، دارایی، عراقی، جاهدیان، نورزاد، طریقی، اشجاری ماهری، امیری، علیقلی، حنیفه‌زاده، روحی، کشوری، نیک‌آذر، صدرزاده، چتر‌روز، خیری‌فام، همدانلو، عبدالهی نمینی، انوری، سبزواریان، کمالی، محمدی غزال، فردی، مینقی، خیرپور کرمانشاهی، جلالی، شیری اسبوئی، بصیری، کرمیان، متانتی، آزادی‌مقدم، برزگر، تشکری، جمشیدی‌فر، چیت‌ساز، ذوقعلی، راجی، سعیدی‌منش، شهابی، فرهادی، مخملی، آقایی میبدی، تبریزی، حاجی حسینی، رستگارفرد، جوکار، خیرخواه، آزما، الماسی‌راد، کتیرائی، دلوچی، اسدی مهمان‌دوستی، ایلخانی، میراخورلو، خان‌محمدی، ظهوری‌فر، تبرزن، بختیارکیا، مازوجی، قادران، آذرتاش نمین، امیرحسینی، خاکپور، پوربختیاری، غلمان، کاوه‌فر، کبیرسلطانی، مسعودی، ارزاقی، میران، توابی، چگینی، روشندل، موعودی، توفیق واحد، معرفتی، نادعلی، نجبائی، رحیمی‌زاده، .....

مهندس ادهم صادقی

www.Abiri.ir

مقدمه ناشر

بسم الله الرّحمن الرّحیم

حمد و سپاس خدا را سزاست که قدرت بیان و توان کتابت را چون دو نعمت بزرگ الهی به انسان ارزانی داشت. همان خدایی که سرچشمه همه نعمتهاست، الرَّحمن؛ عطاکننده گرامیترین نعمت به انسان، عَلَّمَ الْقُرآن؛ خالق انسانی با قدرت نطق و بیان، خَلَقَ الْانسان عَلَّمَهُ الْبَیان.

پس آغاز می‌کنیم به نام او که خالق مُلک وجود است، اِقْرَأْ بِاسْمِ رَبِّکَ الَّذی خَلَق؛ خالقی که اشرف مخلوقات را از خون بسته‌ای آفرید، خَلَقَ الْاِنْسانَ مِنْ عَلَق؛ به نام پروردگار کریمی که نعمت بنده را فوق استحقاقش دهد، اِقْرَأْ وَ رَبُّکَ الْاَکْرَم؛ رحیمی که قلم را در دست بنده‌اش نهاد تا بنگارد، اَلَّذی عَلَّمَ بِالْقَلَم؛ آنچه را که نمی‌دانست و خدایش به او آموخت، عَلَّمَ الْاِنْسانَ ما لَمْ یَعْلَم؛ هدایتگر بشریت را، همان قرآن کریم را، آن که شایسته قسم خوردن است، ن وَ الْقَلَمِ وَ ما یَسْطُرُون؛ چرا که بیان می‌کند آنچه که مایه هدایت انسان است، هذا بَیانٌ لِلنّاسِ وَ هُدًی وَ مَوعِظَةٌ لِلْمُتَّقین.

ای خدای جهان آفرین، قلم را به دست بنده‌ات نهادی تا حروف را یک به یک به یکدیگر بچسباند و کلمه‌ای بنگارد همانگونه که تو خود گلبرگها را به هم چسباندی و گُلی آفریدی؛ کلمه‌ها را به دنبال هم قرار دهد تا جمله‌ای بر صفحه کاغذ نقش بندد همانگونه که تو خود، گلها را کنار یکدیگر بر بستر خاک نهادی تا گلستانی حاصل آید؛ از آن جمله مفهومی در ذهن نقش بندد معطرکننده فضای تفکّر آدمیت، همانگونه که تو به برکت وجود گلستانت، فضای آسمان را معطّر نمودی؛ تا روح انسان اوج گیرد، به معراج رسد و ملحق شود به آن که از او آمده است که اِنّا لله وَ اِنّا اِلَیْهِ راجِعون.

پس دست دعا به سویت دراز می‌کنیم که اگر مقدر فرموده‌ای تا کلامی از قلم این خادمان درگاهت به گوش بندگانت رسد، فانوسی گردد راهنمای آنکه کشتی افکارش در تلاطم دریای علمش ره به سوی منزلی باید، که اِهْدِنا الصِّراطَ الْمُسْتَقیم، صِراطَ الَّذینَ اَنْعَمْتَ عَلَیْهِمْ غَیْرِ الْمَغْضوبِ عَلَیْهِمْ وَ لَاالضّالّین.

پروردگارا بپذیر از ما، همانا تویی شنونده دانا

رَبَّنا تَقَبَّلْ مِنّا إِنَّکَ أَنْتَ السَّمیعُ الْعَلیم

انتشارات تیزهوشان سرزمین کهن
adhamsadeghi@hotmail.com

www.Abiri.ir

تشکر و قدردانی

کشتی زندگی هر انسانی در مسیر رسیدن به اهداف عالیه انسانی بر اساس مقدرات الهی در اقیانوسی در حرکت است که خالی از طوفانها نیست اما در همه تلاطمهای زندگی، گرمی دست پدر، مادر، خواهر، برادران، خواهرزاده‌ها و برادرزادگان عزیزم را در دستانم احساس نموده‌ام، لذا آرزوی خوشبختی آنها، همواره سرلوحه همه آمال و آرزوهایم می‌باشد و امیدوارم که خداوند متعال، مرا برای آنها و آنها را برای من، در سایه الطاف بی‌کرانش، نگهدارد.

نویسنده کتاب، پروفسور دکتر فروزان که در کالیفرنیا تدریس می‌نمایند، ماحصل تلاشهای علمی شبانه‌روزی خود را به صورت کتابهایی درآورده‌اند که اینجانب افتخار ترجمه آنها را به زبان فارسی دارم و تشکر صمیمانه خود را خدمت ایشان عرض می‌کنم که مایه افتخار و مباهات جامعه علمی ایران می‌باشند. ایشان در کتاب خود از زحمات اساتیدی تشکر کرده‌اند که هر یک به نوعی در تهیه کتاب اصلی، یاریشان داده‌اند لذا بر خود لازم می‌دانم که نام آنها را همانگونه که در کتاب اصلی آمده است، در اینجا ذکر نمایم:

Walter Read, California State University, Fresno,
Wayne D. Smith, Mississippi State University,
Ronald R. Srodawa, Oakland University,
Rod Fatoohi, San Jose State University,
Abdullah Abonamah, University of Akron,
Douglas Jacobson, Iowa State University,
Sam C. Hsu, Florida Atlantic University,
Dale Liu, Indiana University-Purdue University-Indianapolis
Dennis Karvelas, New Jersey Institute of Technology,
Paul D. Amer, University of Delaware,
Edward Chlebus, Illinois Institute of Technology,
Anthony Chung, DePaul University,
Isaac Ghansah, California State University, Sacramento,
Khalen Harfoush, North Carolina State University,
Doug Jacobson, Iowa State University,
Tulin Mangir, California State University
Xiao Su, San Jose State University,
Mark Weise, Oklahoma State University

اگر کتاب حاضر ماحصل همکاری جمعی از اندیشمندان است، پس حمد و سپاس مر خدا را سزاست که آسمانها و زمین و آنچه میان آنهاست را به دست توانای خود، بدون نیاز به یار و یاور، آفرید و برای شناخت اسرار نهفته در آنها، انسان را آفرید و به او قدرت تفکر داد تا بیندیشد و به وجود خالقی با این عظمت و شکوه، پی ببرد. پس ای خدای کریم، به ما کمک کن تا اگر قدمی برمی‌داریم، در جهت رضای تو باشد و اگر مدد می‌طلبیم، از تو باشد، که همانا تویی نعم المولی و نعم النصیر.

مهندس ادهم صادقی

www.Abiri.ir

فهرست

فصل اول : مقدمه ۳۵

۱-۱ تاریخچه مختصری از اینترنت ۳۵

ARPANET ۳۶

تولد اینترنت ۳۷

TCP/IP ۳۷

MILNET ۳۸

CSNET ۳۸

NSFNET ۳۸

ANSNET ۳۹

اینترنت امروزی ۳۹

ISP های بین‌المللی ۳۹

ISP های ملی ۴۰

ISP های ناحیه‌ای ۴۱

ISP های محلی ۴۱

تاریخها ۴۱

رشد اینترنت ۴۲

۱-۲ پروتوکلها و استانداردها ۴۲

پروتوکلها ۴۲

ساختار ۴۳

معنا ۴۳

زمان‌بندی ۴۳

استانداردها ۴۳

استانداردهای طبق واقعیت ۴۴

استانداردهای طبق قانون ۴۴

۱-۳ سازمان‌های استانداردسازی ۴۴

کمیته‌های تولید استاندارد ۴۴

سازمان استانداردهای بین‌المللی : ISO ۴۴

اتحادیه بین‌المللی مخابره از راه دور ITU ۴۵

www.Abiri.ir

انستیتوی استانداردهای ملی آمریکا : ANSI ۴۵
انستیتوی مهندسین برق و الکترونیک : IEEE ۴۶
انجمن صنایع الکترونیک : EIA ۴۶
گردهم‌آیی : Forum ۴۶
گردهم‌آییِ Frame Relay ۴۶
گردهم‌آییِ ATM ۴۷
نمایندگی‌های تنظیم‌کننده ۴۷
کمیسیون مخابرات فدرال : FCC ۴۷
۱-۴ استانداردهای اینترنت ۴۷
سطوح تکامل RFC ها ۴۸
استاندارد پیشنهادی ۴۸
طرح اولیه استاندارد ۴۸
استاندارد اینترنت ۴۸
تاریخچه‌ای ۴۹
آزمایشی ۴۹
اطلاعاتی ۴۹
سطوح نیازمندی ۵۰
لازم ۵۰
توصیه شده ۵۰
انتخابی ۵۰
کاربرد محدود ۵۱
توصیه نشده ۵۱
۱-۵ مدیریت اینترنت ۵۱
انجمن اینترنت : ISOC ۵۱
هیئت معماری اینترنت : IAB ۵۲
نیروی مهندسی اینترنت : IETF ۵۲
نیروی تحقیقاتی اینترنت : IRTF ۵۳
سازمان تخصیص اعداد در اینترنت : IANA ۵۳
شرکت اینترنتی برای اعداد و اسامی تخصیص یافته : ICANN ۵۳
مرکز اطلاعات شبکه ۵۳
۱-۶ تاریخچه اینترنت، مقالهٔ مترجم ۵۳

www.Abiri.ir

۱-۷ عبارات کلیدی ۶۰
۱-۸ خلاصه ۶۱
۱-۹ مجموعه تمرین‌ها ۶۲
تمرین‌ها ۶۲
فعالیتهای تحقیقاتی ۶۳

فصل دوم : مدل OSI و پشته پروتوکل TCP/IP ۶۵
۲-۱ مدل OSI ۶۵
معماری لایه‌ای ۶۶
پروسه‌های نظیر به نظیر ۶۷
رابطه بین لایه‌ها ۶۸
سازمان‌دهی لایه‌ها ۶۸
کپسول‌بندی ۷۰
۲-۲ لایه‌ها در مدل OSI ۷۰
لایه فیزیکی ۷۰
مشخصه فیزیکی رابط و محیط انتقال ۷۱
نمایش بیتها ۷۱
نرخ بیت ۷۱
سنکرون کردن بیتها ۷۱
پیکربندی خط ۷۱
توپولوژی فیزیکی ۷۲
مد انتقال ۷۲
لایه دیتالینک ۷۲
فریم‌سازی ۷۳
آدرس‌دهی فیزیکی ۷۳
کنترل فلو ۷۳
کنترل خطای بیت ۷۳
کنترل دسترسی ۷۴
لایه شبکه ۷۴
آدرس‌دهی منطقی ۷۵
مسیردهی ۷۵

www.Abiri.ir

لایه انتقال ۷۶
آدرس‌دهی نقطه سرویس ۷۷
سگمنت‌سازی و مونتاژ مجدد سگمنتها ۷۷
کنترل اتصال ۷۸
کنترل فلوی سگمنتها ۷۸
کنترل خطای سگمنتها ۷۸
لایه جلسه کاری ۷۸
کنترل دیالوگ ۷۹
سنکرونیزاسیون ۷۹
لایه نمایش ۸۰
ترجمه ۸۰
رمزگذاری ۸۰
فشرده‌سازی ۸۱
لایه کاربرد ۸۱
ترمینال مجازی شبکه ۸۲
FTAM ۸۲
سرویسهای email ۸۳
سرویسهای دایرکتوری ۸۳
خلاصه وظایف لایه‌ها ۸۳
۲-۳ پشته پروتوکل TCP/IP ۸۳
لایه‌های فیزیکی و دیتالینک ۸۵
لایه شبکه ۸۵
پروتوکل IP ۸۵
پروتوکل ARP ۸۶
پروتوکل RARP ۸۶
پروتوکل ICMP ۸۶
پروتوکل IGMP ۸۷
لایه انتقال ۸۷
پروتوکل UDP ۸۷
پروتوکل TCP ۸۷
پروتوکل SCTP ۸۸

www.Abiri.ir

لایه کاربرد ۸۸
۲-۴ انواع آدرس‌ها ۸۸
آدرس فیزیکی ۸۹
آدرس‌های فیزیکی تک‌گیرنده، چندگیرنده و همگانی ۹۰
مثال ۱ ..... ۹۰
مثال ۲ ..... ۹۱
آدرس منطقی ۹۱
آدرس‌های منطقی تک‌گیرنده، چندگیرنده و همگانی ۹۲
مثال ۳ ..... ۹۲
مثال ۴ ..... ۹۲
آدرس پورت ۹۳
مثال ۵ ..... ۹۴
مثال ۶ ..... ۹۵
۲-۵ نسخه‌های مختلف IP ۹۵
نسخه ۴ ..... ۹۵
نسخه ۵ ..... ۹۵
نسخه ۶ ..... ۹۵
۲-۶ عبارات کلیدی ۹۶
۲-۵ خلاصه ۹۸
۲-۵ مجموعه تمرین‌ها ۹۹
تمرین‌ها ۹۹
فعالیتهای تحقیقاتی ۱۰۰

فصل سوم : تکنولوژی‌های زیربنائی ۱۰۳
۳-۱ شبکه‌های محلی ۱۰۴
LANهای سیمی : اترنت ۱۰۴
اترنت با سرعت 10 Mbps ۱۰۴
متد دسترسی CSMA/CD ۱۰۵
لایه‌ها ۱۰۶
فریم اترنت ۱۰۷
آدرس‌دهی ۱۰۹

www.Abiri.ir

پیاده‌سازی ۱۰۹
اترنت سریع ۱۱۰
پیاده‌سازی اترنت سریع ۱۱۱
گیگابیت اترنت ۱۱۱
پیاده‌سازی گیگابیت اترنت ۱۱۲
LAN بی‌سیم : IEEE 802.11 ..... ۱۱۳
معماری ۱۱۳
لایه فیزیکی ۱۱۵
IEEE 802.11 FHSS ۱۱۵
IEEE 802.11 DSSS ۱۱۶
IEEE 802.11a OFDM ۱۱۶
IEEE 802.11b HR-DSSS ۱۱۷
IEEE 802.11g OFDM ۱۱۷
زیرلایه MAC ۱۱۷
فریم MAC ۱۱۹
مکانیزم آدرس‌دهی ۱۲۰
۳-۲ شبکه‌های گسترده نقطه به نقطه ۱۲۰
لایه فیزیکی ۱۲۱
مدم‌های 56k ۱۲۱
تکنولوژی‌های DSL ۱۲۱
DSL نامتقارن یا ADSL ۱۲۲
DSL با نرخ بیت بالا یا HDSL ۱۲۵
DSL با نرخ بیت بسیار بالا یا VDSL ۱۲۵
مدم کابلی ۱۲۵
شبکه‌های کابلی معمولی ۱۲۵
شبکه HFC ۱۲۶
پهنای باند ۱۲۶
به اشتراک‌گذاری ۱۲۷
وسائل شبکه تلویزیون کابلی ۱۲۸
خطوط T ۱۲۸
خط T-1 ..... ۱۲۹

www.Abiri.ir

خط T-3 ..... ۱۳۰
SONET ۱۳۰
لایه دیتالینک ۱۳۰
پروتوکل PPP ۱۳۱
لایه‌های PPP ۱۳۱
پروتوکل کنترل لینک ۱۳۲
پروتوکل کنترل شبکه ۱۳۲
PPPoE ۱۳۳
۳-۳ WAN های سویچ‌شده ۱۳۳
X.25 ..... ۱۳۴
Frame Relay ۱۳۴
معماری Frame Relay ۱۳۶
مدارهای مجازی ۱۳۶
لایه‌های Frame Relay ۱۳۶
ATM ۱۳۶
اهداف طراحی ۱۳۷
شبکه‌های سلولی ۱۳۸
ATM آسنکرون ۱۳۸
معماری ATM ۱۳۸
اتصال مجازی ۱۳۹
سلولها ۱۴۰
لایه‌های ATM ۱۴۱
AAL ۱۴۱
AAL1 ..... ۱۴۲
AAL2 ..... ۱۴۲
AAL3/4 ..... ۱۴۲
AAL5 ..... ۱۴۲
لایه ATM ۱۴۳
لایه فیزیکی ۱۴۳
۳-۴ وسایل متصل‌کننده ۱۴۳
تکرارکننده‌ها ۱۴۳

www.Abiri.ir

هابها ۱۴۵
پلها ۱۴۶
فیلتر کردن ۱۴۶
پل شفاف ۱۴۷
به پیش‌رانی ۱۴۸
یادگیری ۱۴۸
سویچ دولایه ۱۴۹
روترها ۱۵۰
سویچ سه‌لایه ۱۵۲
۳-۵ عبارات کلیدی ۱۵۲
۳-۶ خلاصه ۱۵۵
۳-۷ مجموعه تمرین‌ها ۱۵۷
تمرین‌ها ۱۵۷
فعالیتهای تحقیقاتی ۱۵۸

فصل چهارم : آدرس‌های IP – آدرس‌دهی کلاس‌دار ۱۶۱
۴-۱ مقدمه ۱۶۱
فضای آدرس ۱۶۲
نماد ۱۶۲
نماد باینری ۱۶۲
نماد اعداد صحیح نقطه‌دار ۱۶۳
مثال ۱ ..... ۱۶۳
مثال ۲ ..... ۱۶۳
مثال ۳ ..... ۱۶۴
نماد هگزادسیمال ۱۶۴
مثال ۴ ..... ۱۶۴
۴-۲ آدرس‌دهی کلاس‌دار ۱۶۵
تشخیص کلاسها ۱۶۶
پیدا کردن کلاس آدرس با نماد باینری ۱۶۶
مثال ۵ ..... ۱۶۷
مثال ۶ ..... ۱۶۷

www.Abiri.ir

پیدا کردن کلاس با نماد اعداد صحیح نقطه‌دار ۱۶۸
مثال ۷ ..... ۱۶۸
مثال ۸ ..... ۱۶۸
شماره شبکه و شماره ماشین ۱۶۹
کلاسها و بلاکها ۱۷۰
کلاس A ۱۷۰
کلاس B ۱۷۱
کلاس C ۱۷۳
کلاس D ۱۷۴
کلاس E ۱۷۴
آدرس شبکه ۱۷۵
مثال ۹ ..... ۱۷۵
مثال ۱۰ ..... ۱۷۵
مثال ۱۱ ..... ۱۷۵
اطلاعات کافی ۱۷۶
مَسک ۱۷۶
مفهوم مسک ۱۷۶
عملگر AND ۱۷۷
مسکهای پیش‌فرض ۱۷۷
اِعمال مسکها ۱۷۸
مثال ۱۲ ..... ۱۷۸
مثال ۱۳ ..... ۱۷۹
مثال ۱۴ ..... ۱۷۹
نماد CIDR ۱۷۹
کمبود آدرس ۱۸۰
۳-۴ سایر مطالب درباره آدرس‌دهی ۱۸۰
وسائل چندآدرسه ۱۸۰
محل وسیله در شبکه ۱۸۰
آدرسهای خاص ۱۸۱
آدرس شبکه ۱۸۱
آدرس همگانی صریح ۱۸۲

www.Abiri.ir

آدرس همگانی محدود ۱۸۲
این ماشین روی این شبکه ۱۸۴
یک ماشین مشخص روی این شبکه ۱۸۴
آدرس‌برگردان ۱۸۵
آدرسهای خصوصی ۱۸۵
آدرسهای تک‌گیرنده، چندگیرنده و همگانی ۱۸۶
آدرسهای تک‌گیرنده ۱۸۷
آدرسهای چندگیرنده ۱۸۷
آدرسهای چندگیرنده تخصیص یافته ۱۸۷
آدرسهای همگانی ۱۸۸
مثال ۱۵ ..... ۱۸۹
۴-۴ زیرشبکه‌سازی و سوپرشبکه‌سازی ۱۹۰
زیرشبکه‌سازی ۱۹۰
طبقه‌بندی دو سطحی ۱۹۱
طبقه‌بندی سه سطحی ۱۹۳
مسک زیرشبکه ۱۹۳
مسک پیوسته زیرشبکه و مسک ناپیوسته زیرشبکه ۱۹۴
پیدا کردن آدرس زیرشبکه ۱۹۴
مثال ۱۶ ..... ۱۹۵
مسک پیش‌فرض و مسک زیرشبکه ۱۹۵
تعداد زیرشبکه‌ها ۱۹۵
تعداد آدرس‌ها در هر زیرشبکه ۱۹۶
آدرسهای خاص در زیرشبکه‌سازی ۱۹۶
نماد CIDR برای زیرشبکه‌ها ۱۹۶
سوپرشبکه ۱۹۶
مسک سوپرشبکه ۱۹۷
نماد CIDR برای سوپرشبکه‌ها ۱۹۸
منسوخ‌شده‌ها ۱۹۹
۴-۵ عبارات کلیدی ۱۹۹
۴-۶ خلاصه ۲۰۰
۴-۷ مجموعه تمرین‌ها ۲۰۱

www.Abiri.ir

تمرین‌ها ۲۰۱
فعالیتهای تحقیقاتی ۲۰۵

فصل پنجم : آدرس‌های IP – آدرس‌دهی بدون کلاس ۲۰۷

۵-۱ بلاکهای آدرس با طول متغیر ۲۰۸

تعداد آدرسها در یک بلاک آدرس ۲۰۸
اولین آدرس ۲۰۸
مثال ۱ ..... ۲۰۹
مثال ۲ ..... ۲۰۹
مثال ۳ ..... ۲۰۹
مسک ۲۱۰
پیشوند و طول پیشوند ۲۱۱
پسوند و طول پسوند ۲۱۱
پیدا کردن بلاک ۲۱۲
پیدا کردن اولین آدرس ۲۱۲
مثال ۴ ..... ۲۱۲
میان‌بر اول ۲۱۲
مثال ۵ ..... ۲۱۳
میان‌بر دوم ۲۱۳
مثال ۶ ..... ۲۱۳
پیدا کردن تعداد آدرسها در بلاک ۲۱۴
مثال ۷ ..... ۲۱۴
پیدا کردن آخرین آدرس در بلاک ۲۱۴
مثال ۸ ..... ۲۱۵
مثال ۹ ..... ۲۱۵
مثال ۱۰ ..... ۲۱۵
مثال ۱۱ ..... ۲۱۶
بلاک آدرس ۲۱۶

۵-۲ زیرشبکه‌سازی ۲۱۷

پیدا کردن مسک زیرشبکه ۲۱۷
مثال ۱۲ ..... ۲۱۸

www.Abiri.ir

پیدا کردن آدرسهای زیرشبکه ۲۱۸
مثال ۱۳ ..... ۲۱۸
زیرشبکه‌های با طول متغیر ۲۱۹
مثال ۱۴ ..... ۲۱۹
مثال ۱۵ ..... ۲۲۱
۵-۳ تخصیص آدرسها ۲۲۲
مثال ۱۶ ..... ۲۲۲
۵-۴ گراف زیرشبکه‌سازی، ابداع مترجم ۲۲۴
مثال ۱۷ ..... ۲۲۴
مثال ۱۸ ..... ۲۳۱
مثال ۱۹ ..... ۲۳۲
۵-۵ عبارات کلیدی ۲۳۴
۵-۶ خلاصه ۲۳۴
۵-۷ مجموعه تمرین‌ها ۲۳۵
تمرین‌ها ۲۳۵
فعالیتهای تحقیقاتی ۲۳۷

فصل ششم : مسیردهی بسته‌های IP ۲۳۹
۶-۱ تحویل بسته‌های IP ۲۳۹
انواع سرویس‌ها در سطح لایه شبکه ۲۳۹
سرویس اتصال‌گرا ۲۴۰
سرویس فاقد اتصال ۲۴۰
روشهای مختلف تحویل بسته‌های IP ۲۴۰
تحویل مستقیم بسته‌های IP ۲۴۰
تحویل غیرمستقیم بسته‌های IP ۲۴۱
۶-۲ هدایت بسته‌های IP ۲۴۲
متدهای مختلف هدایت بسته‌های IP ۲۴۲
متد هاپ بعدی ۲۴۳
متد شبکه- مشخص ۲۴۳
متد کامپیوتر- مشخص ۲۴۴
متد پیش‌فرض ۲۴۴

www.Abiri.ir

هدایت بسته‌های IP بر اساس آدرس‌دهی کلاس‌دار ۲۴۶
هدایت بسته‌های IP بدون زیرشبکه‌سازی ۲۴۶
مثال ۱ ..... ۲۴۸
مثال ۲ ..... ۲۴۹
مثال ۳ ..... ۲۴۹
هدایت بسته‌های IP به همراه زیرشبکه‌سازی ۲۵۰
مثال ۴ ..... ۲۵۰
مثال ۵ ..... ۲۵۱
مثال ۶ ..... ۲۵۱
هدایت بسته‌های IP بر اساس آدرس‌دهی بدون کلاس ۲۵۲
چند مثال ۲۵۲
مثال ۷ ..... ۲۵۳
مثال ۸ ..... ۲۵۳
مثال ۹ ..... ۲۵۴
مثال ۱۰ ..... ۲۵۴
مثال ۱۱ ..... ۲۵۴
تجمع آدرس ۲۵۶
مطابقت با بلندترین مسک ۲۵۷
مسیردهی طبقاتی ۲۵۷
مثال ۱۲ ..... ۲۵۹
مسیردهی جغرافیایی ۲۶۰
الگوریتمهای جستجو در جدول مسیردهی ۲۶۰
جستجو در آدرس‌دهی کلاس‌دار ۲۶۱
جستجو در آدرس‌دهی بدون‌کلاس ۲۶۱
ترکیب ۲۶۱
۶-۳ مسیردهی بسته‌های IP ۲۶۲
جدول مسیردهی استاتیک و دینامیک ۲۶۲
جدول مسیردهی استاتیک ۲۶۲
جدول مسیردهی دینامیک ۲۶۲
ساختار کلی جدول مسیردهی ۲۶۳
مثال ۱۳ ..... ۲۶۴

www.Abiri.ir

۶-۴ کلیاتی درباره روتر ۲۶۶
اجزای یک روتر ۲۶۶
پورتهای ورودی ۲۶۷
پورتهای خروجی ۲۶۷
پردازشگر مسیردهی ۲۶۷
سخت‌افزار سویچینگ ۲۶۸
سویچ ضربدری ۲۶۸
سویچ بَنیان ۲۶۹
سویچ باچربَنیان ۲۷۰
۶-۵ عبارات کلیدی ۲۷۱
۶-۶ خلاصه ۲۷۲
۶-۷ مجموعه تمرین‌ها ۲۷۳
تمرین‌ها ۲۷۳
فعالیتهای تحقیقاتی ۲۷۴

فصل هفتم : پروتوکلهای ARP و RARP ۲۷۷
۷-۱ پروتوکل ARP ۲۷۹
فرمت بسته ARP ۲۸۱
کپسول‌بندی ۲۸۲
طرز کار پروتوکل ARP ۲۸۳
چهار مورد مختلف به کارگیری ARP ۲۸۴
مثال ۱ ..... ۲۸۵
ARP روی ATM ۲۸۶
Proxy ARP ۲۸۶
۷-۲ نرم‌افزار ARP ۲۸۷
اجزای نرم‌افزار ARP ۲۸۸
جدول حافظه نهان ۲۸۸
صفها ۲۹۰
ماجول خروجی ۲۹۰
جریان رویدادهای ماجول خروجی ۲۹۱
ماجول ورودی ۲۹۲

www.Abiri.ir

جریان رویدادهای ماجول ورودی ۲۹۲
ماجول کنترل‌کننده حافظه نهان ۲۹۳
جریان رویدادهای ماجول کنترل‌کننده حافظه نهان ۲۹۳
مثالهای بیشتر ۲۹۴
مثال ۲ ..... ۲۹۴
مثال ۳ ..... ۲۹۵
مثال ۴ ..... ۲۹۵
مثال ۵ ..... ۲۹۶
۷-۳ پروتوکل RARP ۲۹۶
فرمت بسته RARP ۲۹۸
کپسول‌بندی ۲۹۸
سرور RARP ۲۹۸
راه‌حلهای دیگری به جای RARP ۳۰۰
۷-۴ عبارات کلیدی ۳۰۰
۷-۵ خلاصه ۳۰۰
۷-۶ مجموعه تمرین‌ها ۳۰۱
تمرین‌ها ۳۰۱
فعالیتهای تحقیقاتی ۳۰۳

فصل هشتم : پروتوکل IP ۳۰۵
۸-۱ بسته IP ۳۰۶
فرمت هدر بسته IP ۳۰۶
VER (Version) : ویراست ۳۰۶
HLEN (Header length) : طول هدر ۳۰۷
تفسیر قدیمی و جدید بایت دوم هدر IP ۳۰۸
Service type : نوع سرویس ۳۰۸
زیرفیلد Preference : ارجحیت ۳۰۸
زیرفیلد TOS : نوع سرویس ۳۰۸
DS : سرویسهای ناهمسان ۳۰۹
زیرفیلد Codepoint : نقطه کدگذاری ۳۱۰
Total length : کل طول ۳۱۱

www.Abiri.ir

Identification : شناسه بسته ۳۱۲
Flags : پرچمها ۳۱۳
Fragmentation offset : مکان نسبی این قطعه در بسته ۳۱۳
TTL (Time To Live) : طول عمر ۳۱۳
Protocol : پروتوکل ۳۱۴
Checksum : جمع‌آزما ۳۱۴
Source address : آدرس مبدأ ۳۱۵
Destination address : آدرس مقصد ۳۱۵
مثال ۱ ..... ۳۱۵
مثال ۲ ..... ۳۱۵
مثال ۳ ..... ۳۱۵
مثال ۴ ..... ۳۱۶
۸-۲ قطعه قطعه سازی بسته IP ۳۱۶
ماکزیمم واحد انتقال : MTU ۳۱۶
فیلدهای مرتبط با قطعه قطعه سازی بسته IP ۳۱۸
Identification : شناسه بسته ۳۱۸
Flags : پرچمها ۳۱۹
Fragmentation offset : مکان نسبی این قطعه در کل بسته ۳۱۹
مثال ۵ ..... ۳۲۱
مثال ۶ ..... ۳۲۱
مثال ۷ ..... ۳۲۲
مثال ۸ ..... ۳۲۲
مثال ۹ ..... ۳۲۲
۸-۳ options : گزینه‌ها ۳۲۲
فرمت کلی یک option ۳۲۳
Code : کد ۳۲۳
Copy : کپی ۳۲۳
Class : کلاس ۳۲۴
Number : شماره ۳۲۴
Length : طول ۳۲۴
Data : دیتا ۳۲۴

www.Abiri.ir

انواع optionها ۳۲۴
NOP : ترازبندی میان گزینه‌ها ۳۲۵
EOP : ترازبندی انتهای گزینه‌ها ۳۲۵
Record Route : ضبط کردن مسیر ۳۲۵
Strict Source Route : مسیردهی قطعی از مبدأ ۳۲۷
Loose Source Route : مسیردهی نه چندان قطعی از مبدأ ۳۲۹
Timestamp : برچسب زمانی ۳۳۰
مثال ۱۰ ..... ۳۳۱
مثال ۱۱ ..... ۳۳۲
مثال ۱۲ ..... ۳۳۳
مثال ۱۳ ..... ۳۳۳
مثال ۱۴ ..... ۳۳۳
مثال ۱۵ ..... ۳۳۴
مثال ۱۶ ..... ۳۳۴
۸-۴ Checksum : جمع آزما ۳۳۴
محاسبات مربوط به Checksum ۳۳۴
محاسبه Checksum در فرستنده ۳۳۵
محاسبه Checksum در گیرنده ۳۳۵
پیاده‌سازی Checksum در بسته IP ۳۳۵
مثال ۱۷ ..... ۳۳۷
مثال ۱۸ ..... ۳۳۷
۸-۵ نرم‌افزار IP ۳۳۸
ماجول اضافه‌کننده هدر ۳۴۰
جریان رویدادهای ماجول اضافه‌کننده هدر ۳۴۰
ماجول پردازش‌کننده ۳۴۰
جریان رویدادهای ماجول پردازش‌کننده ۳۴۰
صفها ۳۴۱
جدول مسیردهی ۳۴۱
ماجول هدایت بسته IP ۳۴۱
جدول MTU ۳۴۲
ماجول قطعه قطعه سازی بسته IP ۳۴۲

www.Abiri.ir

جریان رویدادهای ماجول قطعه قطعه سازی بسته IP ۳۴۲
جدول مونتاژ مجدد ۳۴۳
ماجول مونتاژ مجدد ۳۴۳
جریان رویدادهای ماجول مونتاژ مجدد ۳۴۴
۸-۶ عبارات کلیدی ۳۴۵
۸-۷ خلاصه ۳۴۶
۸-۸ مجموعه تمرین‌ها ۳۴۷
تمرین‌ها ۳۴۷
فعالیتهای تحقیقاتی ۳۴۹

فصل نهم : پروتوکل ICMP ۳۵۱
۹-۱ پروتوکل ICMP ۳۵۱
۹-۲ انواع پیغامهای ICMP ۳۵۲
فرمت کلی پیغامهای ICMP ۳۵۳
پیغامهای ICMP از نوع گزارش خطا ۳۵۴
پیغام مقصد، غیر قابل دسترسی است ۳۵۵
پیغام مبدأ، کمتر بسته بفرست ۳۵۸
پیغام مهلت مقرر سپری شده است ۳۶۰
پیغام مشکلی در هدر وجود دارد ۳۶۱
پیغام مسیردهی این بسته باید اصلاح شود ۳۶۲
پیغامهای ICMP از نوع درخواست و پاسخ ۳۶۴
پیغام درخواست و پاسخ انعکاس ۳۶۵
پیغام درخواست و پاسخ برچسب زمانی ۳۶۶
پیغام درخواست و پاسخ مسک ۳۶۹
پیغام درخواست از روتر و پیغام اطلاع‌رسانی روتر ۳۷۰
Checksum در ICMP ۳۷۱
محاسبه Checksum در فرستنده ۳۷۲
تست Checksum در گیرنده ۳۷۲
مثال ۱ ..... ۳۷۲
۹-۳ ابزارهای خطایابی ۳۷۳
ابزار Ping ۳۷۳

www.Abiri.ir

مثال ۲ ..... ۳۷۴
مثال ۳ ..... ۳۷۵
ابزار traceroute ۳۷۶
مثال ۴ ..... ۳۷۸
مثال ۵ ..... ۳۷۸
مثال ۶ ..... ۳۷۹
مثال ۷ ..... ۳۷۹
۹-۴ نرم‌افزار ICMP ۳۸۰
ماجول ورودی ۳۸۰
جریان رویدادهای ماجول ورودی ۳۸۱
ماجول خروجی ۳۸۱
جریان رویدادهای ماجول خروجی ۳۸۲
۹-۵ عبارات کلیدی ۳۸۲
۹-۶ خلاصه ۳۸۳
۹-۷ مجموعه تمرین‌ها ۳۸۴
تمرین‌ها ۳۸۴
فعالیتهای تحقیقاتی ۳۸۶

فصل دهم : پروتوکل IGMP ۳۸۷
۱۰-۱ پروتوکل IGMP ۳۸۷
مدیریت گروه ۳۸۷
۱۰-۲ پیغامهای IGMP ۳۸۸
فرمت پیغام IGMP ۳۸۹
۱۰-۳ نحوه کار IGMP ۳۹۰
ملحق شدن به یک گروه ۳۹۱
ترک گروه ۳۹۲
کنترل عضویت‌ها ۳۹۲
ارسال پاسخ با تأخیر ۳۹۴
مثال ۱ ..... ۳۹۵
روتر پرسش‌کننده ۳۹۶
۱۰-۴ کپسول‌بندی ۳۹۶

www.Abiri.ir

لایه شبکه ۳۹۶
فیلد Protocol ۳۹۷
فیلد TTL ۳۹۷
فیلد آدرس IP مقصد ۳۹۷
لایه دیتالینک ۳۹۸
با پشتیبانی از آدرس فیزیکی multicast ۳۹۸
مثال ۲ ..... ۴۰۰
مثال ۳ ..... ۴۰۰
بدون پشتیبانی از آدرس فیزیکی multicast ۴۰۰
ابزار netstat ۴۰۱
مثال ۴ ..... ۴۰۱
۱۰-۵ نرم‌افزار IGMP ۴۰۲
جدول گروهها ۴۰۳
تایمرها ۴۰۴
ماجول ملحق شدن به گروه ۴۰۴
جریان رویدادهای ماجول ملحق شدن به گروه ۴۰۴
ماجول ترک گروه ۴۰۵
جریان رویدادهای ماجول ترک گروه ۴۰۵
ماجول ورودی ۴۰۵
جریان رویدادهای ماجول ورودی ۴۰۶
ماجول خروجی ۴۰۶
جریان رویدادهای ماجول خروجی ۴۰۶
۱۰-۶ عبارات کلیدی ۴۰۷
۱۰-۷ خلاصه ۴۰۷
۱۰-۸ مجموعه تمرین‌ها ۴۰۸
تمرین‌ها ۴۰۸
فعالیتهای تحقیقاتی ۴۱۱

فصل یازدهم : پروتوکل UDP ۴۱۳
۱۱-۱ مخابره برنامه به برنامه ۴۱۴
شماره پورت ۴۱۵

www.Abiri.ir

تقسیم‌بندی شماره پورتها توسط ICANN ۴۱۸
شماره پورتهای کاملاً شناخته شده برای UDP ۴۱۸
مثال ۱ ..... ۴۱۹
آدرسهای سوکت ۴۲۰
۱۱-۲ دیتاگرام UDP ۴۲۰
فرمت هدر دیتاگرام UDP ۴۲۰
Source port number : شماره پورت مبدأ ۴۲۰
Destination port number : شماره پورت مقصد ۴۲۱
total length : کل طول ۴۲۱
Checksum : جمع‌آزما ۴۲۲
۱۱-۳ Checksum در دیتاگرام UDP ۴۲۲
محاسبه Checksum در فرستنده ۴۲۳
محاسبه Checksum در گیرنده ۴۲۴
یک مثال ۴۲۴
استفاده اختیاری از Checksum ۴۲۴
۱۱-۴ طرز کار UDP ۴۲۵
سرویسهای فاقد اتصال ۴۲۵
کنترل فلو و خطا ۴۲۶
کپسول‌بندی و خارج سازی از کپسول ۴۲۶
کپسول‌بندی ۴۲۶
خارج سازی از کپسول ۴۲۷
در صف گذاشتن ۴۲۷
مالتی‌پلکسینگ و دی‌مالتی‌پلکسینگ در UDP ۴۲۹
مالتی‌پلکسینگ ۴۲۹
دی‌مالتی‌پلکسینگ ۴۳۰
۱۱-۵ کاربردهای UDP ۴۳۰
۱۱-۶ نرم‌افزار UDP ۴۳۱
جدول کنترل بلاک ۴۳۱
صفهای ورودی ۴۳۱
ماجول کنترل بلاک ۴۳۲
جریان رویدادهای ماجول کنترل بلاک ۴۳۲

www.Abiri.ir

ماجول ورودی ۴۳۳
جریان رویدادهای ماجول ورودی ۴۳۳
ماجول خروجی ۴۳۳
جریان رویدادهای ماجول خروجی ۴۳۳
مثالها ۴۳۴
مثال ۲ ..... ۴۳۴
مثال ۳ ..... ۴۳۴
مثال ۴ ..... ۴۳۵
مثال ۵ ..... ۴۳۵
۱۱-۷ عبارات کلیدی ۴۳۶
۱۱-۸ خلاصه ۴۳۶
۱۱-۹ مجموعه تمرین‌ها ۴۳۸
تمرین‌ها ۴۳۸
فعالیتهای تحقیقاتی ۴۳۹

فصل دوازدهم : پروتوکل TCP ۴۴۱
۱۲-۱ سرویسهای TCP ۴۴۲
سرویس مخابراتی برنامه به برنامه ۴۴۲
مثال ۱ ..... ۴۴۲
سرویس تحویل دنباله‌ای از بایتها ۴۴۲
بافرهای فرستنده و گیرنده ۴۴۴
سگمنتها ۴۴۵
سرویس مخابراتی کاملاً دو طرفه ۴۴۶
سرویس اتصال‌گرا ۴۴۶
سرویس قابل اطمینان ۴۴۷
۱۲-۲ ویژگیهای TCP ۴۴۷
سیستم شماره‌گذاری ۴۴۷
شماره بایت ۴۴۸
شماره ترتیب ۴۴۸
مثال ۲ ..... ۴۴۸
شماره تأییدیه ۴۴۹

www.Abiri.ir

کنترل فلو ۴۵۰
کنترل خطا ۴۵۰
کنترل ازدحام ۴۵۰
۱۲-۳ سگمنت TCP ۴۵۰
فرمت سگمنت TCP ۴۵۰
Source port address : آدرس پورت مبدأ ۴۵۱
Destination port address : آدرس پورت مقصد ۴۵۱
Sequence number : شماره ترتیب ۴۵۱
Acknowledgment number : شماره تأییدیه ۴۵۲
Header length : طول هدر ۴۵۲
Reserved : رزرو شده ۴۵۲
Flags : پرچمها ۴۵۲
Window size : سایز پنجره ۴۵۳
Checksum : جمع‌آزما ۴۵۳
Urgent pointer : نشانه‌گر دیتای اورژانسی ۴۵۳
Options : گزینه‌ها ۴۵۴
کپسول‌بندی ۴۵۴
۱۲-۴ فازهای مختلف یک اتصال TCP ۴۵۴
فاز برقراری اتصال ۴۵۵
تبادل سه‌گانه ۴۵۵
باز شدن به طور همزمان ۴۵۸
حمله سیل‌آسای SYN ها ۴۵۸
فاز رد و بدل شدن دیتا ۴۵۹
ارسال بلادرنگ دیتا ۴۶۰
دیتای اورژانس ۴۶۰
فاز قطع اتصال ۴۶۲
تبادل سه‌گانه ۴۶۲
تبادل چهارگانه به همراه نیمه بستن اتصال ۴۶۳
حالتهای خاص ۴۶۴
ریست کردن اتصال ۴۶۴
رد کردن یک اتصال ۴۶۵

www.Abiri.ir

لغو یک اتصال ۴۶۵
خاتمه دادن به یک اتصال غیر فعال ۴۶۶
۱۲-۵ دیاگرام گذر از حالتها ۴۶۶
سناریوها ۴۶۸
برقراری اتصال و در نهایت، خاتمه اتصال ۴۶۸
حالتهای سرویس‌گیرنده ۴۶۸
حالتهای سرور ۴۷۱
خاتمه اتصال با استفاده از تبادل سه‌گانه ۴۷۲
باز شدن به طور همزمان ۴۷۲
بسته شدن به طور همزمان ۴۷۳
رد یک اتصال ۴۷۴
لغو یک اتصال ۴۷۵
۱۲-۶ کنترل فلو ۴۷۶
الگوریتم پنجره لغزان ۴۷۷
مثال ۳ ..... ۴۷۸
مثال ۴ ..... ۴۷۸
مثال ۵ ..... ۴۷۹
مثال ۶ ..... ۴۷۹
مثال ۷ ..... ۴۸۰
مثال ۸ ..... ۴۸۰
مثال ۹ ..... ۴۸۱
متوقف کردن پنجره ۴۸۲
علامت پنجره نامعقول ۴۸۲
علامت توسط فرستنده تولید شده است ۴۸۳
الگوریتم ناگل ۴۸۳
علامت توسط گیرنده تولید شده است ۴۸۴
راه حل کلارک ۴۸۴
ارسال ACK با تأخیر ۴۸۵
۱۲-۷ کنترل خطا ۴۸۵
Checksum ۴۸۶
تأییدیه ۴۸۶

www.Abiri.ir

تولید تأییدیه‌ها ۴۸۶
نوع تأییدیه ۴۸۷
تأییدیه تجمعی : ACK ۴۸۷
تأییدیه انتخابی : SACK ۴۸۸
ارسال مجدد ۴۸۸
ارسال مجدد بعد از RTO ۴۸۹
ارسال مجدد بعد از دریافت سه ACK تکراری ۴۸۹
سگمنتهای خارج از نوبت ۴۸۹
چند سناریو ۴۹۰
کار طبیعی TCP ۴۹۰
سگمنت گم شده ۴۹۱
ارسال مجدد سریع ۴۹۳
سگمنت به تأخیر افتاده ۴۹۳
سگمنت تکراری ۴۹۴
تصحیح اتوماتیک ACK گم شده ۴۹۴
جبران ACK گم شده از طریق ارسال مجدد یک سگمنت ۴۹۴
توقف گفتگو ناشی از ACK گم شده ۴۹۶
۱۲-۸ کنترل ازدحام ۴۹۶
عملکرد شبکه ۴۹۷
منحنی تأخیر- بار ۴۹۸
منحنی نرخ عبور - بار ۴۹۸
مکانیزم کنترل ازدحام ۴۹۹
کنترل حلقه باز ازدحام ۴۹۹
سیاست ارسال مجدد ۵۰۰
سیاست ارسال تأییدیه ۵۰۰
سیاست دور انداختن بسته‌ها ۵۰۰
کنترل حلقه بسته ازدحام ۵۰۰
درخواست دیتای ارسالی کمتر ۵۰۰
نقطه انسداد ۵۰۱
نشانه ضمنی وقوع ازدحام ۵۰۱
نشانه صریح وقوع ازدحام ۵۰۱

www.Abiri.ir

کنترل ازدحام در TCP ۵۰۱
پنجره ازدحام ۵۰۱
نحوه کنترل ازدحام ۵۰۲
شروع آهسته : افزایش نمائی ۵۰۲
جلوگیری از وقوع ازدحام : افزایش خطی ۵۰۴
آشکارسازی ازدحام : کاهش مضربی ۵۰۵
خلاصه‌ای از عملکرد TCP در قبال ازدحام ۵۰۶
۱۲-۹ تایمرهای TCP ۵۰۷
تایمر ارسال مجدد ۵۰۷
زمان رفت و برگشت : RTT ۵۰۸
RTT اندازه‌گیری شده : RTT_M ۵۰۸
RTT هموار شده : RTTs ۵۰۸
میزان انحرافت RTT : RTT_D ۵۱۰
مهلت مقرر برای ارسال مجدد : RTO ۵۱۰
مثال ۱۰ ..... ۵۱۰
الگوریتم کارن ۵۱۲
مثال ۱۱ ..... ۵۱۲
تایمر اصرار بر ارسال ۵۱۳
تایمر زنده نگه دارِ اتصال ۵۱۴
تایمر انتظار قبل از قطع کامل ۵۱۵
۱۲-۱۰ Options : گزینه‌ها ۵۱۵
گزینه EOP ۵۱۶
گزینه NOP ۵۱۷
گزینه ماکزیمم سایز سگمنت : MSS ۵۱۷
ضریب بزرگنمایی پنجره ۵۱۸
گزینه برچسب زمانی ۵۱۹
اندازه‌گیری RTT ۵۲۰
مثال ۱۲ ..... ۵۲۱
حفاظت در مقابل چرخش شماره‌های ترتیبی : PAWS ۵۲۲
گزینه SACK و گزینه SACK - مجاز ۵۲۳
گزینه SACK - مجاز ۵۲۴

www.Abiri.ir

گزینه SACK ۵۲۴
مثال ۱۳ ..... ۵۲۵
مثال ۱۴ ..... ۵۲۶
مثال ۱۵ ..... ۵۲۷
۱۲-۱۱ نرم‌افزار TCP ۵۲۷
بلاکهای کنترل انتقال : TCB ۵۲۸
تایمرها ۵۳۰
ماجول اصلی ۵۳۰
جریان رویدادهای ماجول اصلی ۵۳۱
ماجول پردازش‌کننده ورودی ۵۳۴
ماجول پردازش‌کننده خروجی ۵۳۵
۱۲-۱۲ عبارات کلیدی ۵۳۵
۱۲-۱۳ خلاصه ۵۳۶
۱۲-۱۴ مجموعه تمرین‌ها ۵۳۹
تمرین‌ها ۵۳۹
فعالیتهای تحقیقاتی ۵۴۳

www.Abiri.ir

فصل اول

مقدمه

اینترنت، تحول عظیمی در بسیاری از جنبه‌های زندگی روزمره ما پدید آورده است. اینترنت نحوه کسب و کار و نیز نحوه گذراندن اوقات فراغت ما را تحت تأثیر قرار داده است. به خاطر آورید که اخیراً چه استفاده‌هایی از اینترنت کرده‌اید. احتمالاً یک ***پست الکترونیکی***[1] برای شرکتی ارسال نموده‌اید، صورتحساب آب، برق و یا از این قبیل را پرداخته‌اید. روزنامه شهر دوردستی را خوانده‌اید و یا برنامه زمان‌بندی فیلمهای سینمایی را رؤیت نموده‌اید. همه این کارها را شما به کمک اینترنت انجام داده‌اید. ممکن است درباره یک موضوع پزشکی، مراجع تحقیقاتی گردآوری نموده باشید، اتاقی را در هتلی رزرو کرده باشید، با دوستی، مکاتبه کرده باشید و یا حتی اتومبیلی را خریده باشید. همه این کارها نیز به کمک اینترنت انجام شده‌اند. اینترنت، یک سیستم مخابراتی است که اطلاعات فراوانی را به شکل سازمان‌یافته در اختیار ما گذاشته است.

اینترنت، یک سیستم ساختاریافته و سازمان‌دهی شده است. قبل از آنکه به بحث پیرامون طرز کار آن و ارتباط آن با TCP/IP بپردازیم، ابتدا مختصری درباره تاریخچه اینترنت می‌گوییم و سپس مفهوم پروتوکل و استاندارد و رابطه بین آنها را روشن می‌نماییم. آنگاه سازمانهای مختلفی که استانداردهایی برای اینترنت تولید می‌کنند را معرفی می‌کنیم. این استانداردها توسط هیچ سازمان خاصی تولید نشده‌اند بلکه محصول همکاری مابین کاربران اینترنت هستند. مکانیزمی که طی آن، چنین استانداردهایی ساخته و پرداخته می‌شوند، مورد بحث قرار خواهد گرفت. همچنین در این فصل مقدماتی، به گروههایی اشاره می‌کنیم که مدیریت اینترنت را در دست دارند.

۱-۱ تاریخچه مختصری از اینترنت

شبکه[2] به گروهی از وسایل مخابراتی متصل به هم، مانند کامپیوترها و پرینترها، گفته می‌شود. یک اینترنت (که با i کوچک به صورت internet نوشته می‌شود) دو یا چند شبکه است که می‌توانند دیتایی به یکدیگر مخابره نمایند. قابل توجه‌ترین اینترنت (که با I بزرگ به صورت Internet نوشته می‌شود و به نام اینترنت جهانی موسوم است) از صدها یا هزاران شبکه متصل

1- email
2- network

www.Abiri.ir

به هم تشکیل شده است. اشخاص حقیقی و نیز سازمانهای مختلفی مانند نمایندگی‌ها، مدارس، مراکز تحقیقاتی، شرکتها و کتابخانه‌های بیش از ۱۰۰ کشور دنیا از اینترنت جهانی استفاده می‌کنند. میلیونها نفر، کاربر اینترنت می‌باشند. این سیستم مخابراتی خارق‌العاده از سال ۱۹۶۹ پا به عرصه وجود گذاشته است.

ARPANET

در اواسط دهه ۱۹۶۰، سوپرکامپیوترهای مراکز تحقیقاتی به صورت مستقل کار می‌کردند به طوری که سوپرکامپیوترهای تولید شده توسط تولیدکنندگان مختلف، نمی‌توانستند با یکدیگر کار کنند. ***نمایندگی پروژه‌های تحقیقاتی پیشرفته***[1] در ***سازمان دفاع***[2] به دنبال راهی برای اتصال کامپیوترها بود به طوری که محققین تحت حمایت مالی آنها بتوانند یافته‌های علمی خود را با یکدیگر به اشتراک گذارند و این امر، به کاهش هزینه‌ها و جلوگیری از دوباره‌کاری‌ها منجر شود.

در سال ۱۹۶۷، در یک نشست مربوط به ***انجمن تجهیزات کامپیوتری***[3] ، ARPA ایده‌های خود درباره تحقق ARPANET به عنوان شبکه کوچکی از کامپیوترهای متصل به هم را مطرح کرد. ایده این بود که هر ***کامپیوتری***[4] که الزاماً نمی‌بایست از تولیدات سازنده واحدی باشد، به یک کامپیوتر ***واسط به نام رابط پردازشگر پیغام***[5] متصل شود و سپس IMP ها به یکدیگر متصل گردند. هر IMP می‌بایست با سایر IMP ها و نیز با کامپیوتری که به او متصل است، به مخابره دیتا بپردازد.

در سال ۱۹۶۹، ایده ARPANET به حقیقت پیوست. چهار سوپر کامپیوتر (که هر کدام از آنها، یک ***گره***[6] نامیده می‌شد) در ***دانشگاه کالیفرنیا در لوس‌آنجلس***[7] ، ***دانشگاه کالیفرنیا در سانتاباربارا***[8] ، ***مرکز تحقیقات استنفورد***[9] و دانشگاه Utah از طریق IMPها به هم متصل شدند و شبکه‌ای را ساختند. نرم‌افزاری به نام ***پروتوکل کنترل شبکه***[10] مخابره بین کامپیوترها را امکان‌پذیر می‌کرد.

1- Advanced Research Project Agency : ARPA
2- Department of Defence : DOD
3- Association for Computing Machinery: ACM
4- host
5- Interface Message Processor : IMP
6- node
7- University of California at Los Angeles : UCLA
8- University of California at Santa Barbara : UCSB
9- Stanford Research Institute : SRI
10- Network Control Protocol : NCP

www.Abiri.ir

❊ تولد اینترنت

در سال ۱۹۷۲، Vint Cerf و Bob Kahn که هردو آنها بخشی از گروه اصلی ARPANET بودند، بر روی ساخت چیزی که نام آن را ***پروژه شبکه‌بندی***[1] گذاشته بودند، همکاری خود را آغاز کردند. آنها می‌خواستند شبکه‌های مختلف را به یکدیگر متصل کنند به طوری که کامپیوتری متصل به یک شبکه بتواند با کامپیوتری متصل به شبکه دیگری به مخابره دیتا بپردازد. آنها می‌بایست مسائل زیادی از قبیل بسته‌های اطلاعاتی با سایزهای مختلف، انواع رابطها، نرخهای مختلف انتقال اطلاعات و بالاخره قابلیتهای اطمینان متفاوتی که مورد نیاز بودند را حل کنند. Kahn و Cerf ایده استفاده از وسیله‌ای به نام ***گیت‌وی***[2] را مطرح کردند که به صورت یک سخت‌افزار واسط برای انتقال دیتا از یک شبکه به شبکه دیگر به کار گرفته شود.

❊ TCP/IP

مقاله‌ای که Cerf و Kahn در سال ۱۹۷۳ نوشتند، اشاره به پروتوکلی داشت که می‌توانست ***تحویل پایانه به پایانه***[3] دیتا را فراهم نماید. این پروتوکل، نسخه جدیدی از NCP بود که به آن ***پروتوکل کنترل انتقال***[4] گفته شد و شامل مفاهیمی از قبیل کپسولبندی، دیتاگرام و عملیات مربوط به گیت‌وی‌ها می‌شد. یک ایده کلیدی آن بود که مسئولیت تصحیح خطا از IMP به عهده کامپیوتر گذاشته شود. ARPA به نقطه تمرکز تلاشهای مخابراتی تبدیل شد. در خلال همین زمان بود که مسئولیت ARPANET به ***نمایندگی مخابراتی وزارت دفاع***[5] واگذار شد.

در اکتبر ۱۹۷۷، اینترنتی که از سه شبکه مختلف به نامهای Packet Satellite، ARPANET و Radio Packet ساخته شده بود می‌توانست به شکل موفقیت‌آمیزی کار کند و لذا مخابره بین شبکه‌ها، امکان‌پذیر شده بود.

کمی بعد از آن، مسئولین تصمیم گرفتند که TCP را به دو پروتوکل، تفکیک کنند: TCP و IP . IP وظیفه مسیردهی بسته‌ها را به عهده داشت در حالی که TCP به عملیات سطوح بالاتر مانند سگمنت‌سازی، مونتاژ مجدد و آشکارسازی خطا می‌پرداخت. این پروتوکل شبکه‌بندی، TCP/IP نام گرفت.

در سال ۱۹۸۱، UC Berkeley تحت قراردادی با DARPA، سیستم عامل Unix را

1- Interneting Project
2- gateway
3- end-to-end
4- Transmission Control Protocol : TCP
5- Defense Communication Agency : DCA

www.Abiri.ir

چنان اصلاح کرد که TCP/IP را نیز دربرگیرد. ادغام یک نرم‌افزار شبکه در یک سیستم عامل متداول، تأثیر بسیاری در همه‌جاگیر شدن شبکه‌بندی داشت. پیاده‌سازی مستقل از سازنده یا ***باز***[1] سیستم عامل Berkeley Unix ، کدی را در اختیار سازندگان گذاشت که بر اساس آن می‌توانستند محصولات شبکه‌بندی خود را بسازند.

در سال ۱۹۸۳، پروتوکل اولیه ARPANET منسوخ شد و TCP/IP پروتوکل رسمی ARPANET شد. کسانی که می‌خواستند به کمک اینترنت جهانی، به کامپیوتری در شبکه دیگری دسترسی یابند، می‌بایست TCP/IP را روی کامپیوترشان داشته باشند.

MILNET

در ۱۹۸۳، ARPANET به دو شبکه تفکیک شد: MILNET برای کاربران نظامی و ARPANET برای کاربران غیرنظامی.

CSNET

رویداد مهم دیگری که در تاریخ اینترنت رخ داده است، به وجود آمدن CSNET در سال ۱۹۸۱ بود. CSNET شبکه‌ای بود که سرمایه راه‌اندازی آن از طرف ***سازمان پایه‌گذار علوم ملی***[2] تأمین می‌شد. ایده راه انداختن شبکه CSNET به فکر دانشگاههایی خطور کرد که به دلیل فقدان امکانات ایمنی لازم، اجازه اتصال به شبکه ARPANET را نداشتند. CSNET شبکه ارزان‌تری بود. هیچگونه لینک اضافه‌ای وجود نداشت و نرخ انتقال دیتا، کمتر بود. او ویژگی‌های ARPANET را داشت و به کمک Telnet، اولین سرویس تجاری جابجایی بسته‌های اطلاعات، می‌توانستند در آن، به جابجایی اطلاعات بپردازند.

تا اواسط دهه ۱۹۸۰، اکثر مراکز تحقیقاتی دانشگاه‌های آمریکا، به بخشی از CSNET تبدیل شده بودند. سایر انستیتوها و شرکتها نیز شبکه‌های خود را به کمک پروتوکل TCP/IP به یکدیگر متصل کردند. در ابتدا لغت اینترنت جهانی، به شبکه‌هایی گفته می‌شد که توسط بودجه دولتی به هم متصل شده بودند، ولی امروزه به شبکه‌هایی اتلاق می‌گردد که توسط پروتوکل TCP/IP به مخابره اطلاعات با هم می‌پردازند.

NSFNET

پس از موفقیت CSNET در سال ۱۹۸۶ شبکه‌ای به نام NSFNET توسط NSF به راه

1- open
2- National Science Foundation : NSF

www.Abiri.ir

افتاد و ***ستون فقراتی***[1] شد که می‌توانست پنج مرکز سوپرکامپیوتر که در سراسر آمریکا گسترده بودند را به هم متصل کند. به شبکه‌های مخابراتی اجازه دسترسی به این ستون فقرات به کمک خطوط T1 با سرعتی برابر با 1.544 ***میلیون بیت در ثانیه***[2] ، داده شد و لذا شبکه NSFNET کل آمریکا را تحت پوشش خود قرار داد.

ANSNET*

در سال ۱۹۹۱، تصمیم دولت آمریکا بر این شد که NSFNET توان حمل ترافیک رو به رشد اینترنت را ندارد. سه شرکت به نامهای MCI, Merit, IBM با تشکیل یک سازمان غیرانتفاعی به نام ***سرویسها و شبکه‌های پیشرفته***[3] تصمیم گرفتند یک ستون فقرات پرسرعت به نام ANSNET بسازند.

اینترنت امروزی

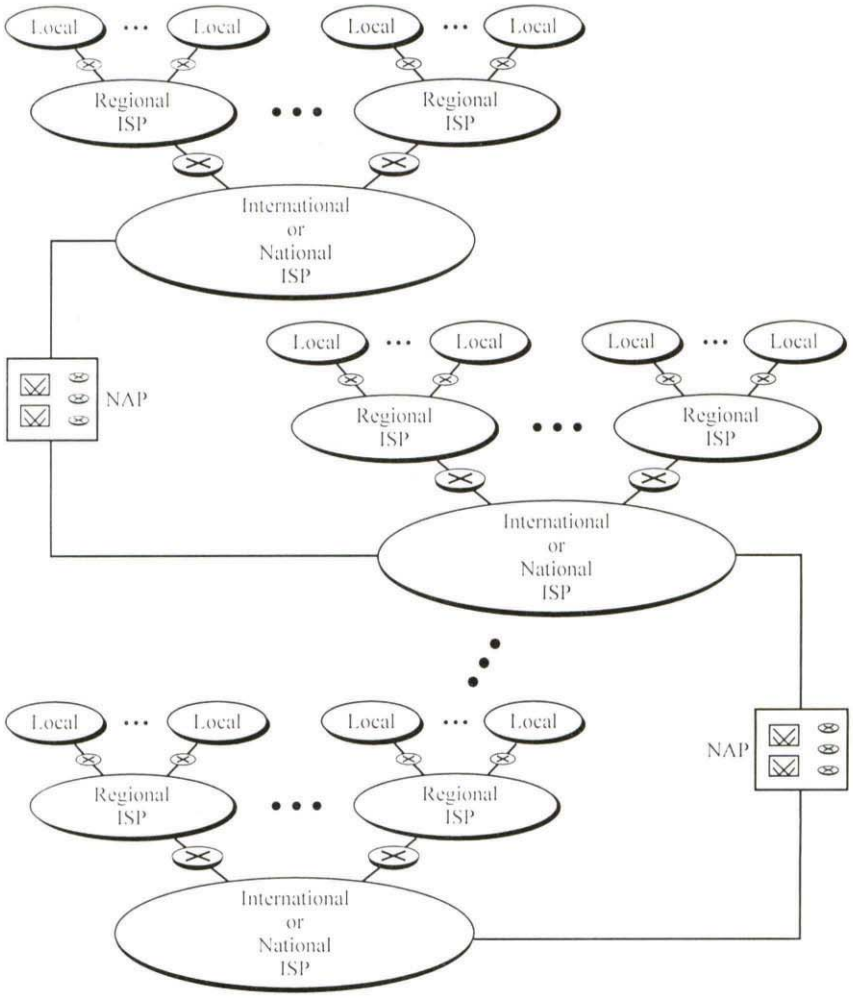
اینترنت امروزی، یک ساختار طبقاتی ساده نیست. اینترنت از تعداد زیادی شبکه‌های گسترده و محلی ساخته شده است که توسط وسائل متصل‌کننده، به یکدیگر متصل شده‌اند. ارائه نمایش صحیحی از اینترنت، کار مشکلی است. دلیل آن این است که اینترنت، دائماً در حال تغییر است، شبکه‌های جدیدی به آن اضافه می‌شوند، شبکه‌های موجود به آدرسهای بیشتری نیاز دارند و باید شبکه‌های مربوط به کمپانی‌های تعطیل شده را از اینترنت حذف کرد. امروزه اکثر کاربرانی که می‌خواهند به اینترنت متصل شوند، از خدمات ***سرویس‌دهندگان اینترنت***[4] و یا ISPها استفاده می‌کنند. ISPهای بین‌المللی، ملی، ناحیه‌ای و محلی، سرویس اتصال به اینترنت را مهیا نموده‌اند. اینترنت امروزی، توسط شرکتهای خصوصی و نه دولتی، خدمات خود را ارائه می‌دهد. شکل ۱-۱ یک شمای کلی (نه جغرافیایی) از اینترنت را نشان می‌دهد.

* ISPهای بین‌المللی

در سطح فوقانی این سلسله مراتب، ***ISPهای بین‌المللی***[5] واقع شده‌اند که کشورهای مختلف را به هم وصل می‌کنند.

1- backbone
2- Mega bit per second : Mbps
3- Advanced Network and Services : ANS
4- Internet Service Provider : ISP
5- International ISP

www.Abiri.ir



شکل ۱-۱: اینترنت امروزی

* ISPهای ملی

ISPهای ملی[1] ، شبکه‌های ستون فقراتی هستند که توسط کمپانی‌های خاصی راه‌اندازی و نگهداری می‌شوند. ISPهای ملی زیادی در شمال آمریکا فعالیت می‌کنند. برخی از مهم‌ترین آنها عبارتند از: Sprintlink ، PSINet ، UUNet Technology ، AGIS ، internet MCI .

1- National ISP

www.Abiri.ir

به منظور برقراری ارتباط بین مشترکین شبکه‌های مختلف، شبکه‌های ستون فقرات توسط ایستگاههای پیچیده سویچینگ (که غالباً توسط شرکتهای دیگری نگهداری می‌شوند) و به نام ***نقاط دسترسی به شبکه***[1] موسومند، به یکدیگر متصل می‌شوند. برخی از ISPهای ملی، توسط ایستگاههای سویچینگ خصوصی به نام ***نقاط نظیر***[2] با یکدیگر ارتباط برقرار می‌کنند. ISPهای ملی غالباً با سرعت بالایی (تا 600 Mbps) کار می‌کنند.

❊ISP های ناحیه‌ای

ISPهای ناحیه‌ای[3] ،ISPهای کوچکی هستند که به یک یا چند ISP ملی متصل شده‌اند. آنها در سطح سوم از تراز سلسله مراتب طبقاتی قرار گرفته‌اند و نرخ انتقال دیتای آنها کمتر از ISPهای ملی است.

❊ISP های محلی

ISPهای محلی[4] مستقیماً به کاربرها، سرویس می‌دهند. می‌توان ISPهای محلی را به ISPهای ناحیه‌ای و یا مستقیماً به ISPهای ملی متصل کرد. اکثر کاربران اینترنت به ISPهای محلی متصل می‌شوند. توجه کنید که در این حالت، یک ISP محلی می‌تواند شرکتی باشد که صرفاً خدمات اینترنتی می‌دهد، سازمان یا شبکه‌ای باشد که خدماتی را فقط به کارمندان خود می‌دهد و یا یک مرکز غیرانتفاعی مانند دانشگاه باشد که شبکه خود را نگهداری می‌کند. هر یک از اینها را می‌توان به یک ISP ناحیه‌ای یا ملی متصل کرد.

❊ تاریخها

لیست زیر، رویدادهای مهم اینترنت را با ترتیب زمانی رخداد آنها نشان می‌دهد.

۱۹۶۹: ARPANET با چهار گره راه اندازی شد.

۱۹۷۰: hostهای ARPA از پروتوکل NCP استفاده کردند.

۱۹۷۳: تهیه پشته پروتوکل TCP/IP آغاز شد.

۱۹۷۸: UNIX در اختیار مراکز علمی / تحقیقاتی قرار گرفت.

۱۹۸۱: CSNET راه‌اندازی شد.

۱۹۸۳: TCP/IP، پروتوکل رسمی ARPANET شد.

1- Network Access Points : NAP
2- Peering Points
3- Regional ISP
4- Local ISP

www.Abiri.ir

۱۹۸۳: MILNET متولد شد.
۱۹۸۶: NSFNET راه‌اندازی شد.
۱۹۹۰: ARPANET جای خود را به NSFNET داد.
۱۹۹۵: NSFNET مجدداً به یک شبکه تحقیقاتی تبدیل شد.
۱۹۹۵: شرکتهایی به نام ISP شکل گرفتند.

*** رشد اینترنت**

اینترنت، رشد بسیاری داشته است. ظرف چند دهه، تعداد شبکه‌های آن، از دهها شبکه به صدها هزار شبکه رسیده است. در همین حال، تعداد کامپیوترهای متصل به شبکه، از صدها به هزاران میلیون کامپیوتر افزایش یافته است. هنوز اینترنت در حال رشد است. مواردی که بر این رشد، اثر داشته‌اند، عبارتند از:

پروتوکلهای جدید: پروتوکلهای جدید باید اضافه شوند و پروتوکلهای منسوخ شده حذف گردند. به عنوان مثال، پروتوکلی که از بسیاری از جنبه‌ها بر IPv4 ارجحیت دارد، به صورت یک استاندارد درآمده است، ولی هنوز به طور کامل پیاده‌سازی نشده است (IPv6 در فصل بیست و هفتم آمده است).

تکنولوژی جدید: تکنولوژی‌های جدیدی در حال توسعه هستند و ظرفیت شبکه‌ها را افزایش می‌دهند تا به این ترتیب، پهنای باند بیشتری در اختیار کاربران اینترنت، قرار گیرد.

استفاده رو به رشد مولتی مدیا: پیش‌بینی می‌شود اینترنت، که زمانی صرفاً وسیله‌ای برای به اشتراک گذاشتن دیتا بود، هرچه بیشتر برای مولتی مدیا (صوت و ویدئو) به کار رود.

۱-۲ پروتوکلها و استانداردها

تعاریفی که در این بخش ارائه می‌شوند، خلاصه‌ای از تعاریفی هستند که به طور کامل در کتاب اصول ارتباطات داده‌ها (جلد اول) از همین مترجم آمده‌اند. ابتدا پروتوکل را تعریف می‌کنیم که مترادف با ***قاعده***[1] است و سپس به استانداردها می‌پردازیم.

پروتوکلها

در شبکه‌های کامپیوتری، مخابره دیتا بین سیستمهای مختلف صورت می‌گیرد. اما دو سیستم نمی‌توانند صرفاً دنباله بیتها را برای هم بفرستند و انتظار داشته باشند که مخابره‌ای بین آنها صورت گیرد. به منظور آنکه مخابره‌ای صورت گیرد، باید دو سیستم مخابره کننده دیتا

1- rule

www.Abiri.ir

از یک پروتوکل مورد توافق طرفین استفاده کنند. پروتوکل، مجموعه قواعد حاکم بر یک سیستم مخابراتی است. پروتوکل می‌گوید که چه چیزی مخابره می‌شود، چگونه مخابره می‌شود و چه زمانی مخابره می‌شود. المانهای کلیدی یک پروتوکل عبارتند از: ***ساختار***[1] ، ***معنا***[2] و ***زمان‌بندی***[3].

*** ساختار**

ساختار به فرمت دیتا اشاره می‌کند و ترتیب و اندازه فیلدهای سازنده پروتوکل را نشان می‌دهد. به عنوان مثال ممکن است در یک پروتوکل، آدرس فرستنده، ۸ بیت اول پروتوکل را دربرگیرد. بعد از آن به طول ۸ بیت، آدرس گیرنده بیاید و سپس اصل پیغام قرار گیرد.

*** معنا**

معنا به معنای هر فیلد از پروتوکل اشاره می‌کند. چگونه باید هر یک از فیلدهای پروتوکل را تفسیر کرد و بر اساس تفسیر مربوطه، چه عملی باید انجام شود؟ به عنوان مثال، آیا یک آدرس، مسیری که باید طی شود را مشخص می‌کند و یا مقصد نهایی پیغام را نشان می‌دهد؟

*** زمان‌بندی**

زمان‌بندی به دو مشخصه اشاره می‌کند: چه زمانی باید دیتا را ارسال کرد و سرعت ارسال آن باید چقدر باشد. به عنوان مثال، اگر فرستنده با سرعت 100 Mbps دیتا تولید کند، اما سرعت پردازش گیرنده فقط 1 Mbps باشد، گیرنده از دیتا لبریز خواهد شد و لذا حجم زیادی از اطلاعات، از دست خواهد رفت.

استانداردها

استانداردها در ایجاد یک بازار باز رقابتی مابین تولیدکنندگان تجهیزات و نیز برای آنکه سیستمهای مخابراتی ملی و بین‌المللی بتوانند به درستی به مبادله اطلاعات بپردازند، ضروری هستند. آنها خط‌مشی‌هایی را در اختیار تولیدکنندگان، نمایندگی‌های دولتی و سایر سرویس‌دهندگان قرار می‌دهند تا در سطح مخابرات بین‌المللی، تضمینی بر عملکرد صحیح سیستمهای مخابراتی، بوجود آید.

1- Syntax
2- Semantics
3- Timing

www.Abiri.ir

استانداردهای سیستمهای مخابره دیتا، به دو گروه تقسیم می‌شوند: ***استانداردهای طبق واقعیت***[1] و ***استاندارد طبق قانون***[2].

❊ استانداردهای طبق واقعیت

استانداردهایی که توسط یک سازمان استانداردسازی ارائه نشده‌اند اما استفاده گسترده‌ای دارند، به نام استانداردهای طبق واقعیت موسومند. اساساً این استانداردها توسط تولیدکنندگانی ارائه می‌شوند که می‌خواهند طرز کار یک محصول یا تکنولوژی جدید را بیان کنند.

❊ استانداردهای طبق قانون

این استانداردها توسط یک سازمان استانداردسازی اعلام شده‌اند.

۱-۳ سازمانهای استانداردسازی

در اثر همکاری ***کمیته‌های تولید استاندارد***[3]، ***گردهم‌آیی‌ها***[4] و ***نمایندگی‌های تنظیم‌کننده دولتی***[5]، استانداردها تولید می‌شوند.

کمیته‌های تولید استاندارد

در عین حال که سازمانهای زیادی به تولید استاندارد مشغول هستند، مخابره دیتا در شمال آمریکا، اساساً برمبنای استانداردهای صادره از سازمانهای زیر، صورت می‌گیرد:

❊ سازمان استانداردهای بین‌المللی : ISO

ISO[6] که به نام International Organization for Standardization نیز موسوم می‌باشد، یک سازمان چند ملیتی است که اعضای آن را کمیته‌های تولیدکننده استاندارد دولتهای کشورهای مختلف جهان، تشکیل داده‌اند. ISO در سال ۱۹۴۷ تأسیس شد و یک سازمان کاملاً داوطلبانه است و برای رسیدن به توافقاتی در سطح استانداردهای بین‌المللی برپا شده است. عضویت آن را نمایندگی‌های کشورهای صنعتی شکل داده‌اند و ISO با ارائه مدلهایی

1- de facto standards
2- de jure standards
3- Standards Creation Committees
4- Forums
5- government regulatory agencies
6- International Standards Organization

www.Abiri.ir

برای سازگاری، کیفیت بهتر، کارآیی بیشتر و قیمت پایین‌تر تولیدات سازندگان مختلف، عملاً تبادل بین‌المللی کالاها و خدمات را آسان‌تر کرده است. فعالیت ISO ایجاد همکاری در زمینه‌های علمی، تکنولوژیکی و اقتصادی می‌باشد. توجه اصلی این کتاب به فعالیت ISO در بخش تکنولوژی اطلاعات معطوف شده است و این بخش، به تولید ***مدل اتصال سیستمهای باز***[1] یا OSI برای شبکه‌های مخابراتی، پرداخته است. ANSI عضو رأی دهنده آمریکا در ISO می‌باشد.

* اتحادیه بین‌المللی مخابره از راه دور : ITU

تا اواسط دهه ۱۹۷۰، تعدادی از کشورها، استانداردهای ملی را برای مخابره از راه دور به کار می‌بردند، اما استانداردهای ملی، سازگاری کمی در سطح بین‌المللی داشتند. پاسخ ملل متحد این بود که به عنوان بخشی از ***اتحادیه بین‌المللی مخابره از راه دور خود***[2] کمیته‌ای به نام ***کمیته مشاوره برای تلفن و تلگراف بین‌المللی***[3] را شکل داد. این کمیته به تحقیقات و برقراری استانداردهایی برای مخابره از راه دور به طور اعم و سیستم‌های دیتا و تلفنی به طور اخص، اختصاص یافت. در اول مارس ۱۹۹۳، این کمیته به نام ***اتحادیه بین‌المللی مخابره از راه دور- بخش استانداردهای مخابره از راه دور***[4] تغییر نام داد.

* انستیتوی استانداردهای ملی آمریکا : ANSI

علیرغم نامش، ***انستیتوی استانداردهای ملی آمریکا***[5] یک شرکت غیرانتفاعی کاملاً خصوصی است و مشارکتی با دولت فدرال آمریکا ندارد. با این وجود، تمام فعالیتهای ANSI تحت حمایت ایالات متحده بوده و مردم نقش مهمی را ایفا می‌کنند. اهداف مطرح شده ANSI شامل ارائه خدمات، به عنوان انستیتوی هماهنگ‌کننده ملی برای استانداردهای ملی در ایالات متحده و به پیش بردن پذیرفتن استانداردها به عنوان راهی برای پیشبرد اقتصاد آمریکا و حصول اطمینان از دخیل نمودن و حفاظت از علایق عمومی می‌باشد. اعضاء ANSI شامل انجمنهای حرفه‌ای، انجمنهای صنفی، تشکیلات دولتی و تنظیم‌کننده و گروهی از مصرف‌کنندگان می‌باشد.

1- Open Systems Interconnection : OSI
2- International Telecommunications Union : ITU
3- Consultative Committee for International Telegraphy and Telephony : CCITT
4- International Telecommunications Union - Telecommunication Standards Sector: ITU-T
5- American National Standards Institute : ANSI

www.Abiri.ir

* انستیتوی مهندسین برق و الکترونیک : IEEE

انستیتوی مهندسین برق و الکترونیک[1] بزرگترین انجمن مهندسین حرفه‌ای در دنیا می‌باشد. با حیطه عمل بین‌المللی، هدف آن پیشبرد تئوری، خلاقیت و کیفیت محصولات، در حیطه مهندسی برق، الکترونیک و رادیو و همچنین در تمام شاخه‌های مربوط به مهندسی می‌باشد. به عنوان یکی از اهداف، IEEE بر توسعه و پذیرش استانداردهای بین‌المللی برای کامپیوتر و مخابرات نظارت دارد.

* انجمن صنایع الکترونیک : EIA

هم‌گام با ANSI ، ***انجمن صنایع الکترونیک***[2] یک سازمان غیرانتفاعی برای ارتقاء کارهای مربوط به ساخت ادوات الکترونیکی است. فعالیتهای آن شامل آموزشهای عمومی و سرویس‌های رأی‌گیری، مضاف بر تولید استاندارد می‌باشد. در حوزه تکنولوژی اطلاعات، EIA با تعریف رابطهای فیزیکی و مشخصه‌های سیگنالینگ الکترونیکی برای مخابره دیتا، نقش عمده‌ای ایفا نموده است.

گردهم‌آیی : Forum

تکنولوژی مخابره از راه دور، سریعتر از توان کمیته‌های استانداردسازی برای تصویب استانداردها، به پیش می‌رود. کمیته‌های استانداردسازی، تشکیلات مرحله مرحله‌ای هستند و لذا حرکت آنها، کند است. برای رسیدن به مدلهای مورد توافق و به منظور آسانتر کردن فرایند استانداردسازی، بسیاری از گروه‌های علاقمند، ***گردهم‌آیی‌ها***[3] تشکیل داده‌اند و اعضاء آنها، نمایندگانی از شرکتهای علاقه‌مند می‌باشند. این گردهم‌آیی‌ها با دانشگاهها و کاربران، در زمینه ارزیابی و استاندارد کردن تکنولوژی‌های جدید، همکاری می‌کنند. با تمرکز تلاشهای خود بر یک تکنولوژی خاص، این گردهم‌آیی‌ها توانستند به پذیرش و استفاده از آن تکنولوژی‌ها در جامعه مخابره از راه دور، سرعت بخشند. آنها تصمیمات اتخاذ شده را به سازمانهای استانداردسازی ارائه می‌کنند. بعضی از گردهم‌آیی‌های مهم برای صنعت مخابره از راه دور، به شرح زیر می‌باشند:

* گردهم‌آیی Frame Relay

گردهم‌آیی Frame Relay توسط Stratacom , Cisco , Northern Telecom , DEC

1- Institute of Electrical and Electronics Engineers
2- Electronic Industries Alliance
3- Forums

www.Abiri.ir

تشکیل شد تا پذیرش و پیاده‌سازی Frame Relay را ارتقاء بخشند. امروزه این گردهم‌آیی، ۴۰ عضو دارد که نمایندگان آمریکای شمالی، اروپا و کشورهای ناحیه اقیانوس آرام می‌باشند. مواردی که در دست بازنگری می‌باشند، عبارتند از کنترل فلو، کپسول‌بندی، ترجمه و پخش عمومی. نتایج برای ISO ارسال می‌شوند.

* گردهم‌آیی ATM

این گردهم‌آیی و کنسرسیوم برای ارتقاء پذیرش و استفاده از تکنولوژی ***مد انتقال آسنکرون***[1] تشکیل شده است. گردهم‌آیی ATM از فروشندگان تجهیزات مورد استفاده مشتری (مانند سیستم‌های PBX) و فراهم کنندگان تجهیزات مراکز تلفنی (مانند Telephone exchanges) تشکیل شده است. هدف آن، استاندارد نمودن سرویس‌ها برای حصول اطمینان از عملکرد صحیح آنهاست.

نمایندگی‌های تنظیم‌کننده

تمام تکنولوژی‌های مخابراتی در معرض تنظیمات اعمال شده توسط ***نمایندگی‌های تنظیم‌کننده***[2] می‌باشند. این نمایندگی در آمریکا، ***کمیسیون مخابرات فدرال***[3] نامیده می‌شود. هدف این نمایندگی‌ها، حفاظت از علائق عمومی توسط اعمال نظم و انضباطی به مخابره سیمی، کابلی، تلویزیونی و رادیو است.

* کمیسیون مخابرات فدرال : FCC

FCC نظارتی بر آن بخش از تجارت درون ایالتی و بین‌المللی آمریکا دارد که به مخابرات مربوط می‌شود.

مهم: FCC یک نمایندگی تنظیم‌کننده می‌باشد.

۱-۴ استانداردهای اینترنت

استاندارد اینترنت، یک مشخصه کاملاً تست شده است که نه تنها برای کاربران اینترنت مفید است، بلکه مورد استفاده همیشگی آنهاست. روند تولید یک استاندارد برای اینترنت، یک روند کاملاً مشخص است و برای آنکه مشخصه‌ای به استانداردی برای اینترنت تبدیل شود، باید

1- Asynchronous Transfer Mode : ATM
2- Regulatory Agencies
3- Federal Commuincations Commission : FCC

www.Abiri.ir

روال معینی را طی کند. هر مشخصه‌ای با یک ***طرح اولیه اینترنتی***[1] آغاز می‌شود. یک طرح اولیه اینترنتی، یک مشخصه در حال اجرا در اینترنت است که هنوز به صورت رسمی درنیامده است و طول عمر آن شش ماه است. با گرفتن تأییدیه مسئولین اینترنت، می‌توان طرح اولیه را به صورت یک ***نظرخواهی***[2] منتشر کرد.

RFCها، ***سطوح مختلف تکامل***[3] را طی می‌کنند و بسته به سطح نیاز به آنها، گروه‌بندی می‌شوند.

سطوح تکامل RFCها

یک RFC در مدت زمان حیات خود، در یکی از شش سطح تکامل قرار دارد: ***استاندارد پیشنهادی***[4] ، ***طرح اولیه استاندارد***[5] ، ***استاندارد اینترنت***[6] ، ***تاریخچه‌ای***[7] ، ***آزمایشی***[8] و ***اطلاعاتی***[9] (شکل ۱-۲ را ببینید).

* استاندارد پیشنهادی

یک استاندارد پیشنهادی، مشخصه باثبات و خوب درک شده‌ای است که جامعه اینترنت به آن علاقه‌مند هستند. غالباً مشخصه‌ای در این سطح، توسط چندین گروه مختلف، تست و پیاده‌سازی می‌شود.

* طرح اولیه استاندارد

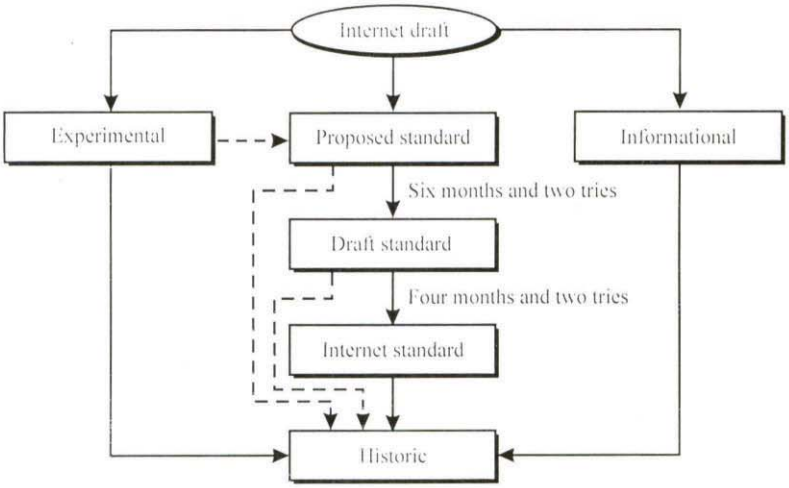
یک استاندارد پیشنهادی، بعد از حداقل دو پیاده‌سازی مستقلی که بتوانند با هم کار کنند، به سطح طرح اولیه استاندارد، ارتقاء می‌یابد. در صورت مشاهده مشکلاتی، اصلاحات مربوطه اعمال می‌شوند و طرح اولیه استاندارد، به یک استاندارد اینترنت تبدیل می‌شود.

* استاندارد اینترنت

بعد از آنکه پیاده‌سازی‌ها، گواه موفقیت‌آمیز بودن طرح اولیه بودند، طرح اولیه استاندارد به سطح استاندارد اینترنت ارتقاء می‌یابد.

1- Internet draft
2- Request For Comment : RFC
3- maturity level
4- proposed standard
5- draft standard
6- Internet standard
7- historic
8- experimental
9- informational

www.Abiri.ir



شکل ۱-۲ : سطوح تکامل یک RFC

❊ تاریخچه‌ای

RFCهایی که جنبه تاریخی دارند، یا مشخصه‌ای هستند که توسط یک مشخه آتی منسوخ شده‌اند و یا هرگز تا سطح یک استاندارد اینترنت، ارتقاء سطح نیافته‌اند و لذا صرفاً از نظر تاریخی اهمیت دارند.

❊ آزمایشی

RFCهایی که در سطح آزمایشی هستند، به کاری اشاره دارند که به یک وضعیت آزمایشی در اینترنت مربوط می‌شود و لذا عملکرد اینترنت را تحت تأثیر قرار نمی‌دهند. چنین RFCهایی را نباید در یک سرویس عملیاتی اینترنتی، پیاده‌سازی کرد.

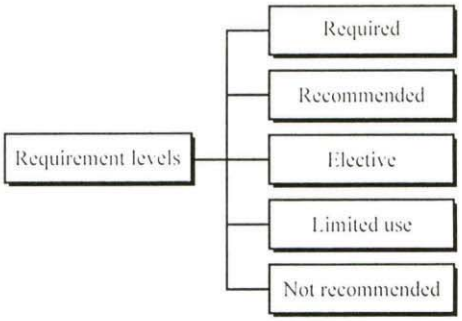
❊ اطلاعاتی

یک RFC که در سطح اطلاعاتی قرار دارد، حاوی اطلاعات آموزشی، تاریخی و عمومی درباره اینترنت است. معمولاً کسی که عضو یک سازمان غیراینترنتی است، مانند یک تولیدکننده تجهیزات، چنین RFCای را می‌نویسد.

www.Abiri.ir

سطوح نیازمندی

RFCها به پنج سطح نیازمندی دسته‌بندی می‌شوند: ***لازم***[1]، ***توصیه شده***[2]، ***انتخابی***[3]، ***کاربرد محدود***[4] و ***توصیه نشده***[5] (شکل ۱-۳ را ببینید).



شکل ۱-۳: سطوح نیازمندی یک RFC

❊ لازم

اگر یک RFC از سوی کلیه سیستمهای اینترنتی برای رسیدن به حداقل سطح هماهنگی، لازم‌الاجراء باشد، به سطح نیازمندی « لازم » می‌رسد. به عنوان مثال IP (فصل هشتم) و ICMP (فصل نهم) پروتوکلهای لازم هستند.

❊ توصیه شده

RFC که برچسب « توصیه شده » را خورده است، برای رسیدن به یک حداقل سطح هماهنگی، لازم نیست، اما به دلیل مفید بودن آن، توصیه شده است. به عنوان مثال FTP (فصل نوزدهم) و Telnet (فصل هجدهم) پروتوکلهای توصیه شده هستند.

❊ انتخابی

یک RFC با سطح نیازمندی « انتخابی » نه لازم و نه توصیه شده است. با این وجود، یک سیستم می‌تواند برای استفاده‌های خود از آن بهره‌مند شود.

1- required
2- recommended
3- elective
4- limited use
5- not recommended

www.Abiri.ir

❊ کاربرد محدود

RFCای که با سطح نیازمندی « کاربرد محدود » معرفی شده باشد را تنها می‌توان با شرایط محدودی به کار برد. اکثر RFCهای آزمایشی، در این گروه قرار دارند.

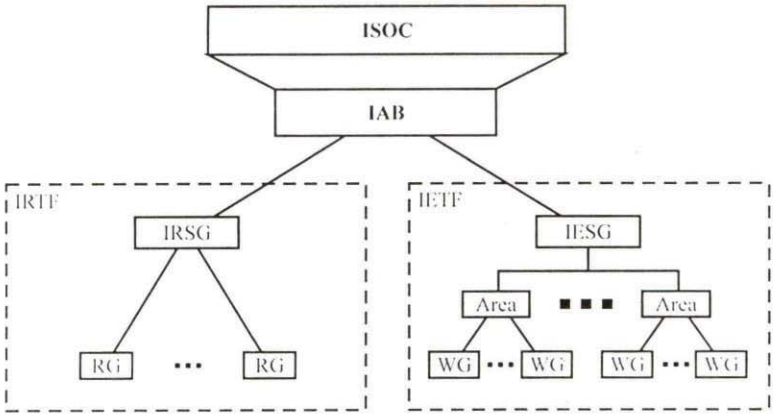
❊ توصیه نشده

یک RFC با سطح نیازمندی « توصیه نشده » برای کاربرد عمومی مناسب نیست. معمولاً یک RFC (منسوخ شده) تاریخی، ممکن است در این گروه قرار گیرد.

مهم: میتوان RFCها را در سایت www.faqs.org/rfcs پیدا کرد.

۱-۵ مدیریت اینترنت

اینترنت که ابتدا برمبنای کارهای تحقیقاتی شکل گرفت، توسعه پیدا کرد و کاربران بیشتری را با فعالیتهای اقتصادی قابل توجهی به خود جلب کرد. گروه‌های متعددی که هماهنگی امورات اینترنت را برعهده دارند، به این رشد و توسعه، جهت بخشیدند. شکل ۱-۴ ساختار کلی سازماندهی مدیریت اینترنت را نشان می‌دهد.



شکل ۱-۴ : مدیریت اینترنت

انجمن اینترنت : ISOC

انجمن اینترنت[1] یک سازمان غیرانتفاعی بین‌المللی است که در سال ۱۹۹۲ به منظور

1- Internet SOCiety

www.Abiri.ir

پشتیبانی از روند تولید استانداردهایی برای اینترنت، شکل گرفت. ISOC از طریق حمایت و پشتیبانی از سایر تشکیلات مدیریتی اینترنت مانند IANA ، IRTF ، IETF و IAB (بخشهای زیر را ببینید) به این هدف، جامه عمل می‌پوشاند. همچنین ISOC سبب رشد و توسعه فعالیتهای تحقیقاتی مرتبط با اینترنت شده است.

هیئت معماری اینترنت : IAB

هیئت معماری اینترنت[1] مشاور فنی ISOC است. هدف اصلی IAB ، نظارت بر ادامه روند توسعه پشته پروتوکل TCP/IP و نیز ارائه مشاوره فنی به محققین جامعه اینترنت است. IAB از طریق دو رکن اصلی خود به این هدف نایل می‌شود: ***نیروی مهندسی اینترنت***[2] و ***نیروی تحقیقاتی اینترنت***[3] . مسئولیت دیگر IAB، مدیریت تهیه و تدوین RFCهاست که قبلاً بحث آنها آمد. همچنین IAB نقش واسط بین اینترنت و سایر سازمانهای استانداردسازی و نیز گروه‌های علاقه‌مند را ایفا می‌کند.

* نیروی مهندسی اینترنت : IETF

IETF از ***گروه‌های کاری***[4] علاقه‌مند، تحت مدیریت ***گروه پیشران مهندسی اینترنت***[5] تشکیل شده است. IETF مسئول پیدا کردن مشکلات عملی اینترنت و سپس ارائه راه حلهایی برای برطرف کردن آنهاست. همچنین IETF، مشخصه‌هایی که قرار است به صورت استانداردهای اینترنت درآیند را مورد بازنگری و بررسی قرار می‌دهد. فعالیتهای گروه‌های کاری در زمینه‌های مختلفی است که هر یک از آنها، یک ***حوزه کاری***[6] نامیده می‌شود. در حال حاضر، ۹ حوزه کاری تعریف شده‌اند و عبارتند از:

۱-کاربردها (Applications)
۲- پروتوکلهای اینترنت (Internet Protocols)
۳- مسیردهی (Routing)
۴- عملیات (Operations)
۵- سرویسهای کاربر (User Services)
۶- مدیریت شبکه (Network Management)

1- Internet Architecture Board : IAB
2- Internet Engineering Task Force : IETF
3- Internet Research Task Force : IRTF
4- WG: Working Groups
5- Internet Engineering Steering Group : IESG
6- area

www.Abiri.ir

۷- انتقال (Transport)
۸- نسل بعدی پروتوکل اینترنت (Internet Protocol next generation : IPng)
۹- امنیت (Security)

* نیروی تحقیقاتی اینترنت :IRTF

IRTF از ***گروه‌های تحقیقاتی***[1] تحت مدیریت ***گروه پیشران تحقیقات اینترنت***[2] تشکیل شده است. تمرکز کاری IRTF بر موضوعات تحقیقاتی بلندمدت مرتبط با پروتوکلها، کاربردها، معماری و تکنولوژی اینترنت است.

سازمان تخصیص اعداد در اینترنت[3] : IANA

شرکت اینترنتی برای اعداد و اسامی تخصیص یافته[4] :ICANN

تا اکتبر ۱۹۹۸، IANA با حمایت از طرف دولت آمریکا، مسئول مدیریت ***نام حوزه‌ها***[5] و نیز آدرسها در اینترنت بود. از آن سال به بعد، شرکت خصوصی و غیرانتفاعی ICANN به مدیریت یک هیئت بین‌المللی، مسئولیتهای IANA را به عهده گرفت.

مرکز اطلاعات شبکه[6]

مسئولیت NIC، جمع‌آوری و توزیع اطلاعات درباره پروتوکلهای TCP/IP است.

مهم: مسئولیت مدیریت نام حوزه‌ها به عهده ICANN می‌باشد.

۱-۶ تاریخچه اینترنت، مقالهٔ مترجم

امروزه مردم زیادی از اینترنت به طرق مختلف استفاده می‌کنند اما شاید تعداد کمی بدانند که شبکه جهانی اینترنت چگونه ساخته شد. پی بردن به چگونگی پیشرفت اینترنت از آغاز تاکنون، دید بهتری نسبت به نحوه تکامل این اختراع بشری ارائه می‌نماید و لذا خواننده در جریان روند پیشرفت اینترنت قرار می‌گیرد و به دنبال آن، درمی‌یابد که چگونه می‌توان از شبکه جهانی به نحو مؤثرتری استفاده کرد. لذا در این بخش، تاریخچه اینترنت از آغاز یعنی از سال

1- Research Groups: RG
2- Internet Research Steering Group: IRSG
3- Internet Assigned Numbers Authority
4- Internet Corporation for Assigned Names and Numbers
5- domain names
6- Network Information Center: NIC

www.Abiri.ir

۱۹۴۰ تاکنون ارائه می‌شود و در ادامه نشان داده خواهد شد که کشورهای جهان چگونه از شبکه جهانی برای پیشرفت مملکت خود بهره برده و می‌برند.

در سال ۱۹۴۰ شخصی به نام Vannevar Bush از طرف روزولت، رئیس جمهور وقت آمریکا، به سرپرستی ***مرکز تحقیقات سازمان دفاع ملی آمریکا***[1] برگزیده شد. از آنجا که در آن سالها، خبر از وقوع یک جنگ جهانی گسترده در میان بود، آمریکا تلاش می‌کرد تا در طول جنگ، توان علمی ساخت تجهیزات جدید جنگی را برای خود مهیا نماید.

پس از یک سال، یعنی در سال ۱۹۴۱، این شخص به سمت ***مدیریت سازمان توسعه و تحقیقات علمی***[2] برگزیده شد. در طول جنگ جهانی دوم، وظیفه او جمع‌آوری دانشمندان و محققین از سراسر جهان به منظور ساخت تسلیحات جدید نظامی بود. Vannevar توانست در این مرکز تحقیقات که نقطه همکاری دانشگاه و ارتش در طول جنگ جهانی دوم محسوب می‌شد، تا پایان جنگ جهانی دوم، به تعداد ۶۰۰۰ نفر دانشمند و محقق را گرد هم آورد.

در پایان جنگ جهانی دوم، یعنی سال ۱۹۴۵، Vannevar این ایده را مطرح کرد که آیا می‌توان از طریق تجهیزات الکترونیکی به اطلاعات دست یافت؟ در آن زمان هنوز دانش تئوریک لازم برای تحقق چنین ایده‌ای فراهم نبود.

در سال ۱۹۴۶ ، Vannevar به سمت ***ریاست گروه همکاری تحقیقات و توسعه***[3] منصوب شد و به منظور فراهم کردن پتانسیل مالی لازم برای حمایت از تحقق ایده خود، یک نمایندگی به نام ***نمایندگی پروژه‌های تحقیقاتی پیشرفته وزارت دفاع***[4] یا به اختصار DARPA را راه‌اندازی کرد. قرار بر این شد که هر زمان، پایه علمی عملی لازم برای تحقق چنین ایده‌ای فراهم گردید، این نمایندگی حمایت مالی لازم را ارائه نماید.

دو سال بعد، یعنی سال ۱۹۴۸، ریاضیدانی به نام ***شانون***[5] تئوری اطلاعات را به شکل ریاضی آن پایه‌ریزی کرد و گفت که می‌توان اطلاعات را به شکل ۰ و ۱ نمایش داد. سپس فرمول ظرفیت یک کانال مخابراتی را ارائه کرد و نشان داد که کانالهای مخابراتی ظرفیت مشخصی برای حمل اطلاعات دارند. پایه تئوری مناسبی که توسط شانون بنا گردید، سبب شد که ایده ذخیره و انتقال اطلاعات از طریق ادوات الکترونیکی تقویت گردد اما در آن زمان هنوز تکنولوژی، امکان ساخت چنین ادواتی را فراهم نکرده بود.

ده سال بعد، یعنی در سال ۱۹۵۸ ، چیپهای سیلیکان، ساخته شدند و به دنبال آن امکان ذخیره‌سازی و انتقال اطلاعات به شکل عملی میسر شد.

1- National Defence Research
2- Director of office of Scientific Research & Development
3- Chairman of Joint Research & Development Board
4- Defence Advanced Research Project Agency : DARPA
5- Shannon

www.Abiri.ir

در سال ۱۹۶۰ ، شرکت honeywell با استفاده از این چیپهای سیلیکان، سوپر کامپیوتر DDP 516 را با 12 KW حافظه و سرعت پردازش 1.1 MHZ ساخت.

در این هنگام پایه تئوریک و تکنولوژیکی لازم، برای ذخیره و انتقال اطلاعات از طریق ادوات الکترونیکی، فراهم شده بود و لذا زمانی که Baran در سال ۱۹۶۲ وارد نیروی هوایی آمریکا شد، این ایده را مطرح کرد که یک شبکه مخابراتی ساخته شود، که با تکیه بر انتقال اطلاعات از طریق ادوات الکترونیکی، ویژگی اصلی پایداری در مقابل حملات هوایی را از خود نشان دهد. تجربه جنگ جهانی دوم، نشان داده بود که شبکه مخابراتی، یکی از حساسترین نقاطی است که اگر مورد حمله دشمن قرار گیرد، می‌تواند ارتباط بین نیروها را از بین ببرد و لذا صدمات جبران‌ناپذیری را وارد نماید.

Baran به مدت دو سال بر روی این ایده کار کرد و سپس در سال ۱۹۶۴ ساختار یک شبکه مخابراتی بر اساس ***سویچینگ بسته‌های اطلاعاتی***[1] را مطرح کرد. بسته‌ها در این سیستم، نه تنها حامل اطلاعات هستند بلکه آدرس لازم برای رسیدن به مقصد را نیز در بردارند و لذا می‌توانند به طور مستقل از سایر بسته‌ها، به مقصد برسند. این شبکه در واقع مشابه شبکه پستی عمل می‌کرد که در آن هر نامه، آدرس مقصد خود را دارد و به طور مستقل از سایر نامه‌ها، به سمت مقصد مورد نظر، مسیردهی می‌شود.

لازم بود که این ایده در سازمانی به مرحله اجرا برسد و لذا در سال ۱۹۶۵ شخصی به نام Bob Taylor که مدیر ***سازمان تکنیکهای پردازش اطلاعات***[2] بود، فردی به نام Roberts را به منظور تحقق چنین شبکه مخابراتی استخدام کرد.

زمانی که خبر ساخت چنین شبکه مخابراتی در سال ۱۹۶۶ به گوش سرپرست DARPA یعنی آقای Herzfeld رسید، وی یک میلیون دلار به Bob Taylor پیشنهاد داد که این کار را با جدیت تمام دنبال نماید و Bob نیز انجام کلیه مراحل لازمه را به Roberts محول کرد.

چیزی که در آن زمان Roberts با آن مواجه بود، امکانات مالی کافی برای تحقق ایده‌ای بود که خود او هیچگونه اطلاعات فنی از آن نداشت، لذا تنها کاری که می‌توانست انجام دهد این بود که در کنفرانسها شرکت کند و درباره این ایده، با افراد مختلف به نشست و گفتگو بپردازد. یک سال بعد، یعنی در سال ۱۹۶۷ ، در یکی از این کنفرانسها، Roberts شخصی به نام Weily Clark را ملاقات کرد و او به Roberts پیشنهاد کرد که برای تحقق این شبکه مخابراتی، به جای آنکه تلاش کند تا تغییراتی در کامپیوترها ایجاد کند تا آنها بتوانند با یکدیگر

1- Packet Switching
2- Information Processing Techniques Office : IPTO

www.Abiri.ir

اطلاعاتی را مخابره کنند، از یک مینی کامپیوتر به عنوان ***رابط پردازشگر پیغامها***[1] یا IMP استفاده نماید. سپس هر کامپیوتر را به یک IMP جداگانه وصل کند تا مکاتبه بین کامپیوترها از طریق IMPها صورت گیرد. در این مرحله، مدل ساخت شبکه فراهم شده بود و لازم بود تا در مکان مناسبی این مدل، پیاده‌سازی شود.

در سال ۱۹۶۸ ، شرکت ***BBN***[2] قراردادی با Roberts منعقد کرد و بر اساس آن قرار شد تا چهار نفر از چهار دانشگاه، شبکه‌ای با چهار ***گره***[3] (کامپیوتر) را بسازند. جدول ۱-۱ را ببینید.

دانشگاه	UCLA	Stanford	UCSB	Utah
کامپیوتر	SDS Sigma 7	SDS 940	IBM 360/75	Dec PDP 10
سیستم‌عامل	Sex	Genie	OS/MVT	Tenex
سرپرست	Crocker	Rulifson	Storghton	Carr

جدول ۱-۱: شبکه اولیه با چهار گره

علت آنکه کار در UCLA شروع شد، این بود که Kleinrock در آنجا سرپرست ***مرکز اندازه‌گیری‌های شبکه***[4] بود و لذا کار بر روی شبکه در آنجا آغاز شده بود. به همین دلیل، اولین کامپیوتر این شبکه، همان سوپرکامپیوتر دانشگاه UCLA یعنی SDS Sigma 7 بود و Kleinrock مشغول نوشتن نرم‌افزاری شد که لازم بود تحت سیستم عامل این کامپیوتر کار کند و مکاتبه سوپر کامپیوتر با IMP را امکان‌پذیر سازد. رابط ۳۲ سیمی که این کامپیوتر را به IMP مربوطه متصل می‌کرد، توسط Wingfeld در تابستان ۱۹۶۹ ساخته شد که ارتباط کاملاً دو طرفه را بین کامپیوتر و IMP امکان‌پذیر می‌ساخت. Bob Kahn از طرف شرکت BBN وظیفه نوشتن نرم‌افزار IMP را به عهده گرفت. او که تجربه کاری در زمینه کار با تجهیزات دیجیتالی را در شرکت AT&T کسب کرده بود، کار نوشتن نرم افزار IMP را روی مینی کامپیوتر DDP 516 آغاز کرد. به این ترتیب در اواسط اکتبر ۱۹۶۹ اولین گره در این شبکه مخابراتی در دانشگاه UCLA شکل گرفت. لازم بود تا گره دوم در دانشگاه Stanford ساخته شود و یک لینک مخابراتی این دو گره را به هم متصل کند. در اواخر اکتبر ۱۹۶۹ این لینک

1- Interface Message Processor : IMP
2- Bolt, Beranek, Newman : BBN
3- Node
4- Network Measurment Center : NMC

www.Abiri.ir

مخابراتی به طول ۳۵۰ مایل و با سرعت ۵۰ کیلو بیت در ثانیه توسط شرکت AT&T ساخته شد و کار ساخت گره دوم نیز در دانشگاه Stanford به پایان رسید.

Kleinrock و همکارانش در دانشگاه UCLA در تاریخ ۲۹ اکتبر سال ۱۹۶۹ ساعت ۱۰:۳۰ شب، تصمیم به تست شبکه متشکل از دو گره یاد شده گرفتند. Kleinrock به شاگردش می‌گوید بزن L و او دکمه L را می‌زند. از طریق تلفن از Stanford خبر رسید که got the L یعنی کاراکتر L دریافت شد. Kleinrock می‌گوید بزن O و شاگردش دکمه O را می‌زند. خبر دریافت کاراکتر O به صورت got the O از طریق تلفن دریافت شد. می‌گوید بزن G و شاگردش دکمه G را می‌زند که ناگهان از طریق تلفن، پیغام System Crash دریافت شد و معنای آن این بود که سیستم از کار افتاده است. چند ساعت طول کشید تا مشکل سیستم برطرف شود و سپس کلمه LOGIN به عنوان اولین کلمه در این شبکه مخابراتی که نام آن را ARPANET گذاشته بودند، بین دو کامپیوتر، یکی در UCLA و دیگری در Stanford ،مخابره شد.

لازم بود که کامپیوترها و یا همان hostهای شبکه ARPANET با استفاده از پروتوکلی با یکدیگر به مخابره دیتا بپردازند و لذا در سال ۱۹۷۰، ***پروتوکل کنترل شبکه***[1] یا NCP ساخته شد.

در سال ۱۹۷۳ ، شخصی به نام Vint Cerf که تجربه کاری با پروتوکل NCP داشت به همراه Bob Kahn که نرم‌افزار IMP را نوشته بود، مقاله‌ای تحت عنوان ***پروتوکلی برای اتصال شبکه‌های بسته‌ها***[2] نوشتند که نقطه تولد پروتوکل ***TCP***[3] محسوب می‌شود.

Cerf در سال ۱۹۷۴ به همراه دانشجویان خود در Stanford ، کار ساخت پروتوکل TCP/IP را آغاز کرد. آدرس IP به صورت یک آدرس سی و دو بیتی معرفی شد که در آن از یک شناسه هشت بیتی برای مشخص کردن شبکه و یک شناسه بیست و چهار بیتی برای مشخص کردن کامپیوتری در آن شبکه استفاده می‌شد. به چنین آدرسی، آدرس کلاس A گفته می‌شد که می‌توانست ۲۵۶ شبکه و هر شبکه حاوی ۱۶ میلیون کامپیوتر را آدرس‌دهی کند. در آن زمان، فرض بر این بود که تعداد محدودی از سوپرکامپیوترها وجود دارند که شبکه‌ها را می‌سازند و بقیه، مشترک این شبکه‌ها می‌شوند لذا تعداد ۲۵۶ آدرس برای مشخص کردن شبکه‌ها، کافی به نظر می‌رسید که البته با پدیدار شدن کامپیوترهای شخصی، چهره کار به طور کامل تغییر کرد.

1- Network Control Protocol : NCP
2- A Protocol for Packet Network Interconnection (TCP)
3- Transmission Control Protocol : TCP

www.Abiri.ir

لازم بود که شبکه‌ها به کمک وسایلی به نام ***روتر***[1] به هم متصل شوند. بنابراین یک زن و شوهر که در دانشگاه Stanford بر روی روترها کار می‌کردند، در سال ۱۹۷۷ شرکتی به نام cisco را با هدف تولید روترها، تأسیس کردند. در همین هنگام ARPA قراردادی نیز با شرکت BBN برای ساختن اولین روتر، منعقد کرد.

در سال ۱۹۸۱، پروتوکل TCP/IP که امتحان خود را روی شبکه ARPANET با موفقیت پس داده بود، به همراه سیستم عامل UNIX ، آزادانه در اختیار مؤسسات تحقیقاتی قرار گرفت تا در سایتهای آنها مورد استفاده قرار گیرد.

در سال ۱۹۸۳، به دلیل موفقیت عملکرد TCP/IP ، این پروتوکل به عنوان پروتوکل رسمی شبکه ARPANET شناخته شد. در همین هنگام شبکه MILNET متولد شد و بخش نظامی ARPANET را از آن جدا کرد.

در سال ۱۹۸۴، ARPANET به INTERNET تغییر نام داد و این تاریخ را می‌توان نقطه تولد اینترنت، در نظر گرفت. در این زمان، تعداد کاربران شبکه به حدود ۱۰۰۰ نفر رسیده بود.

محققین بسیاری از اینرنت برای بحث پیرامون یافته‌های علمی خود استفاده می‌کردند و در این میان فردی به نام Tim Berners Lee که در آزمایشگاه فیزیک ذره‌ای CERN کار می‌کرد، به منظور ساده‌تر کردن نقل و انتقال یافته‌های تحقیقاتی از طریق اینترنت، در سال ۱۹۸۹ ***زبان علامت‌گذاری ابرمتن***[2] یا HTML را اختراع کرد. از این زمان بود که ساخت صفحات وب آغاز شد و در ضمن، تعداد کاربران شبکه به حدود ۱۰۰۰۰۰ نفر رسیده بود.

در سال ۱۹۹۲، یعنی زمانی که تعداد کاربران شبکه به حدود ۱۰۰۰۰۰۰ نفر رسید، لازم بود که استفاده از اطلاعات موجود روی اینترنت، برای عموم آسان‌تر گردد. لذا شخصی به نام Mark Andresson به همراه Eric bina در دانشگاه Illinois ، کار بر روی ساختن یک ***مرورگر***[3] را آغاز کردند و هدف از ساخت آن، این بود که بتوان به سهولت، صفحات وب را مشاهده کرد. آنها کار خود را به عنوان یک پروژه دانشگاهی در ***مرکز ملی محاسبات سوپرکامپیوترها***[4] در دانشگاه Illinois آغاز کردند.

در سال ۱۹۹۳ ، مرورگر Mosaic توسط این دو نفر ساخته شد. اما مرکز NCSA مرورگر ساخته شده را متعلق به خود دانست و لذا Andresson کار خود را بر روی ساخت مرورگر دیگری آغاز کرد که منجر به ساخت مرورگر Netscape در سال ۱۹۹۴ شد.

1- Router
2- Hyper Text Markup Language : HTML
3- Browser
4- National Center for Supercomputing Applications : NCSA

www.Abiri.ir

در سال ۱۹۹۵ ، شرکتهایی که دسترسی به اینترنت را برای مردم فراهم می‌کردند و به آنها ***ISP***[1] گفته می‌شد، به سرعت رشد کردند و ***تجارت الکترونیکی***[2] نیز به دنبال آن آغاز شد.

در سال ۱۹۹۸ ، مرورگر Netscape توسط Andresson به AOL که یکی از بزرگترین سرویس‌دهندگان اینترنت در آمریکا است، فروخته شد. تعداد کاربران اینترنت به سرعت افزایش می‌یافت، به طوری که در سال ۲۰۰۲، تعداد کاربران اینترنت به بیش از ۲۰۰ میلیون نفر رسید.

شاید در نگاه اول، اهمیت ۲۰۰ میلیون نفر کاربر اینترنت روشن نباشد اما ذکر این نکته لازم است که اگر فرض کنیم که شما یک صفحه وب را روی اینترنت بگذارید و با احتمال یک در میلیون، این سایت مورد بازدید همگان قرار گیرد، بنابراین روزانه حداقل ۲۰۰ نفر از سایت شما بازدید خواهند کرد. به همین دلیل است که خبر از خرید و فروشهایی در محیط اینترنت به گوش می‌رسد که پیش از این، حتی امکان آن نیز نبود. مثلاً ملکه‌ای از سایتی که یک عده بافنده تهیه کرده‌اند، بازدید می‌کند و به نَنویی که آنها ساخته‌اند، علاقه‌مند می‌شود و سفارش خرید چند عدد نَنو را به آنها می‌دهد!

این مسئله باعث شده است که سازمان ملل، پیشرفت کشورها را در آگاه شدن آنها از اطلاعات موجود در دنیا و نه صنعتی شدن آنها، بداند. تا پیش از این، فرض بر این بود که اگر کشوری بخواهد به رشد اقتصادی نائل شود، لازم است آن کشور به سمت صنعتی شدن حرکت کند اما امروزه کافی است که بدانیم چه کسی، چه چیزی می‌سازد و سپس این اطلاعات با ارزش را به کسانی که خواستار آن هستند، بفروشیم. از سوی دیگر، کالایی که در یک کشور به قیمت ناچیز تولید می‌شود، وقتی که در معرض خرید جهانی قرار می‌گیرد، می‌تواند سود قابل توجهی را نصیب تولیدکنندگان آن کند و به این ترتیب رنج وسیعی از ***شغلهای کوچک***[3] اما سودآور شکل می‌گیرند.

مواردی که به آنها اشاره شد، نشان می‌دهند که اینترنت، در کشورهایی که ساختار بانکی و پستی آنها رشد کافی نموده‌اند، می‌تواند سبب رشد اقتصادی کشور گردد. وجود یک ساختار بانکی و پستی مناسب، لازمه تحقق تجارت اینترنتی است زیرا که اولاً، باید مبادله پول به راحتی امکان‌پذیر باشد و ثانیاً، زمانی که قرار است یک کالای فیزیکی از دست تولیدکننده به دست مصرف‌کننده برسد، ساختار پستی مناسب، بتواند امکان تحویل ارزان کالا را فراهم نماید. در کشورهای در حال توسعه مانند سنگاپور، مالزی، فیلیپین و غیره، از این ایده استفاده شده است و سبب گردیده است که در سالهای اخیر، شاهد رشد اقتصادی چشمگیری در این

1- Internet Service Provider : ISP
2- E-commerce
3- Small Business

www.Abiri.ir

کشورها باشیم. آنها همچنین بخش عمده‌ای از کارهای دولتی را از طریق اینترنت انجام می‌دهند و لذا از اینترنت، نه تنها در بهبود وضعیت اقتصادی خود استفاده کرده‌اند، بلکه به کمک آن، خدمات دولتی را نیز بهتر نموده‌اند. تجربه موفق این کشورها نشان می‌دهد که اینترنت، چه اثر سازنده‌ای در بهبود وضعیت یک کشور می‌تواند داشته باشد، به همین دلیل سازمان ملل، مقالاتی را تهیه نموده است (www.eprimers.org) که همگی اشاره به نحوه پیشرفت کشورها از طریق به کارگیری هر چه بیشتر اینترنت دارند.

با امید به آنکه شاهد پیشرفت هر چه بیشتر تمام کشورهای در حال توسعه از طریق به کارگیری شبکه جهانی باشیم.

۱-۷ عبارات کلیدی

نمایندگی پروژه‌های تحقیقاتی پیشرفته
Advanced Research Project Agency: ARPA
انستیتوی استانداردهای ملی آمریکا
American National Standards Institute: ANSI
ANSNET
ARPANET
ATM Forum
کمیته مشاوره‌ای برای تلفن و تلگراف بین‌المللی
Consultative Committee for International Telegraphy and Telephony : CCITT
CSNET
استاندارد طبق واقعیت de facto standard
استاندارد طبق قانون de jure standard
انجمن صنایع الکترونیک Electronic Industries Alliance: EIA
کمیسیون مخابرات فدرال Federal Communications Commission: FCC
Frame Relay Forum
انستیتوی مهندسین برق و الکترونیک
Institute of Electrical and Electronic Engineers: IEEE
سازمان بین المللی استانداردها
International Standards Organization: ISO
اتحادیه بین‌المللی مخابره از راه دور – بخش استانداردهای مخابره از راه دور
International Telecommunications Union-Telecommunications Standards Sector
اینترنت جهانی : Internet
هیئت معماری اینترنت Internet Architecture Board : IAB
سازمان تخصیص اعداد در اینترنت
Internet Assigned Numbers Authority : IANA
شرکت اینترنتی برای اعداد و اسامی تخصیص یافته

www.Abfr.ir

Internet Corporation for Assigned Names and Numbers : ICANN
طرح اولیه استاندارد draft standard
نیروی کار مهندسی اینترنت Internet Engineering Task Force: IETF
نیروی کار تحقیقاتی اینترنت Internet Research Task Force: IRTF
سرویس‌دهنده اینترنت Internet Service Provider: ISP
جامعه اینترنت Internet Society: ISOC
استاندارد اینترنت Internet Standard
سرویس‌دهندگان محلی اینترنت Local ISP
سطوح تکامل Maturity Levels
MILNET
شبکه Network
نقاط دسترسی به شبکه Network Access Points : NAP
مرکز اطلاعات شبکه Network Information Center : NIC
NSFNET
پروتوکل Protocol
نظرخواهی Request For Comment: RFC
سطوح نیازمندی requirement levels
معنا Semantics
ساختار Syntax
زمان‌بندی timing
پروتوکل کنترل انتقال / پروتوکل اینترنت
Tramsmission Control Protocol / Internetworking Protocol

۱-۸ خلاصه

- اینترنت جهانی، مجموعه‌ای از صدها هزار شبکه مختلف است.
- ARPANET به عنوان شبکه‌ای با چهار گره آغاز به کار کرد.
- TCP/IP پشته پروتوکل اینترنت است.
- CSNET مخابره‌ای بین شبکه‌هایی برقرار کرد که اجازه اتصال به ARPANET را نداشتند.
- NSFNET مخابره‌ای بین کلیه شبکه‌های موجود در آمریکا را فراهم نمود.
- ISPهای محلی، کاربران را به اینترنت متصل می‌کنند.
- ISPهای ناحیه‌ای، ISPهای محلی را به اینترنت متصل می‌کنند.

www.Abiri.ir

- ISPهای ملی، شبکه‌های ستون فقرات هستند که توسط شرکتهای خاصی نگهداری می‌شوند.
- پروتوکل، مجموعه قواعد حاکم بر یک سیستم مخابراتی است. المانهای اصلی یک پروتوکل عبارتند از: ساختار، معنا و زمان‌بندی.
- به منظور حصول اطمینان از اینکه محصولات تولیدکنندگان مختلف بتوانند با هم کار کنند، به استانداردها نیازمندیم.
- ISO ، ITU-T ، ANSI ، IEEE و EIA برخی از سازمانهایی هستند که در تولید استانداردها نقش دارند.
- Forumها، گروه‌های علاقه‌مندی هستند که به سرعت، تکنولوژیهای جدید را ارزیابی می‌کنند و آنها را به سمت استاندارد شدن، سوق می‌دهند. دو Forum مهم عبارتند از: ATM Forum و Frame Relay Forum.
- FCC، یک نمایندگی تنظیم‌کننده است که به مخابره رادیو، تلویزیون و تلفن سیمی نظم می‌بخشد.
- RFC یک نظر یا ایده است که پیش درآمد یک استاندارد اینترنتی می‌باشد.
- هر RFC در یکی از گروه‌های لازم، توصیه شده، انتخابی، کاربرد محدود و توصیه نشده قرار می‌گیرد.
- ISOC به تشویق و پیشبرد فعالیتهای تحقیقاتی در رابطه با اینترنت، می‌پردازد.
- IAB، مشاور فنی ISOC است.
- IETF از گروه‌های کاری علاقه‌مندی تشکیل شده است که مشکلات علمی اینترنت را پیدا می‌کنند و سپس راه حلهایی را برای برطرف کردن آن مشکلات، ارائه می‌نمایند.
- IRTF از گروههای تحقیقاتی تشکیل شده است که تمرکز کاری آنها، بر موضوعات تحقیقاتی بلندمدت مرتبط با پروتوکلها، کاربردها، معماری و تکنولوژی اینترنت است.
- ICANN که قبلاً IANA نامیده می‌شد، مسئول مدیریت اسم حوزه‌ها و آدرسها در اینترنت می‌باشد.
- NIC مسئول جمع‌آوری و توزیع اطلاعات درباره پروتوکلهای TCP/IP است.

۱-۹ مجموعه تمرین‌ها

تمرین‌ها

۱- با استفاده از اینترنت جهانی، شماره RFCها را پیدا کنید.

۲- با استفاده از اینترنت جهانی، ببینید RFC 1603 و RFC 2418 درباره چه هستند.

۳- با استفاده از اینترنت جهانی، RFCای را پیدا کنید که پیرامون خط‌مشی‌های گروه‌های

www.Abiri.ir

تحقیقاتی IRTF و روال و روند کار آنها بحث می‌کند.

۴- با استفاده از اینترنت جهانی، دو مثال برای یک RFC تاریخچه‌ای ارائه دهید.

۵- با استفاده از اینترنت جهانی، دو مثال برای یک RFC آزمایشی ارائه دهید.

۶- با استفاده از اینترنت جهانی، دو مثال برای یک RFC اطلاعاتی ارائه دهید.

۷- با استفاده از اینترنت جهانی، RFCای را پیدا کنید که درباره پروتوکل FTP است.

۸- با استفاده از اینترنت جهانی، RFCای را پیدا کنید که درباره پروتوکل IP است.

۹- با استفاده از اینترنت جهانی، RFCای را پیدا کنید که درباره پروتوکل TCP است.

۱۰- با استفاده از اینترنت جهانی، RFCای را پیدا کنید که درباره روند ایجاد استانداردی در اینترنت است.

فعالیتهای تحقیقاتی

۱۱- سه استاندارد پیدا کنید که توسط ITU-T تولید شده باشند.

۱۲- سه استاندارد پیدا کنید که توسط ANSI تولید شده باشند.

۱۳-EIA استانداردهایی برای رابطها تولید کرده است. دو تا از این استانداردها را پیدا کنید. EIA-232 چیست؟

۱۴-سه تا از تنظیماتی که توسط FCC در رابطه با AM و FM مطرح شده‌اند، را پیدا کنید.

www.Abiri.ir

فصل دوم

مدل OSI و پشته پروتوکل TCP/IP

مدل لایه‌ای حاکم بر مخابره دیتا و شبکه‌ها تا قبل از سال ۱۹۹۰، مدل ***OSI***[1] بود. همه فکر می‌کردند که مدل OSI، استاندارد نهایی مخابره دیتا خواهد شد اما چنین چیزی رخ نداد. پشته پروتوکل TCP/IP ، به صورت ساختار حاکم بر اینترنت درآمد، زیرا که بطور گسترده‌ای در اینترنت به کار گرفته شد و مورد ارزیابی قرار گرفت. هیچگاه مدل OSI بطور کامل، پیاده‌سازی نشد.

در این فصل، ابتدا بطور خلاصه درباره مدل OSI صحبت می‌کنیم و سپس تمرکز کار خود را بر پشته پروتوکلTCP/IP می‌گذاریم.

۲-۱ مدل OSI

ISO که در سال ۱۹۴۷ تأسیس شده است، یک سازمان چند ملیتی است که برای رسیدن به توافقات جهانی دربارهٔ استانداردهای بین‌المللی شکل گرفته است. یک استاندارد ISO که کلیه جنبه‌های مخابره در شبکه‌ها را شامل می‌شود، مدل OSI است. اواخر دهه ۱۹۷۰، مدل OSI برای اولین بار معرفی شد. یک ***سیستم باز***[2] مجموعه‌ای از پروتوکلهاست که دو سیستم متفاوت را قادر می‌سازد تا علی رغم تفاوت موجود در تکنولوژی‌های زیربنائی آنها، با یکدیگر به مخابره دیتا بپردازند. هدف مدل OSI ، ساده کردن مخابره دیتا بین سیستم‌های مختلف، بدون نیاز به اعمال تغییراتی در سخت‌افزار یا نرم‌افزار آن سیستم‌ها، می‌باشد. مدل OSI ، یک پروتوکل نیست. مدل OSI ، مدلی برای درک و طراحی یک معماری شبکه‌ای با ویژگی‌های قابلیت انعطاف، قابلیت اطمینان و امکان برقراری ارتباط با سایر شبکه‌ها، می‌باشد.

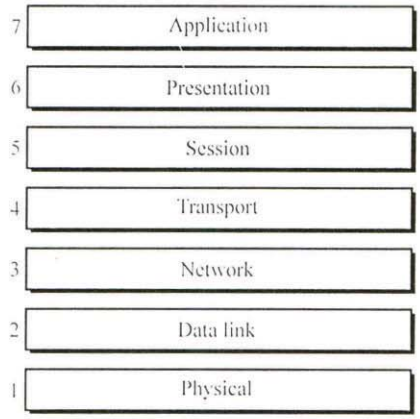
مهم : ISO نام یک سازمان است. OSI نام یک مدل است.

مدل OSI ،یک ساختار لایه‌ای برای طراحی شبکه‌هایی است که مخابره دیتا بین انواع سیستم‌های کامپیوتری را امکان‌پذیر می‌نمایند. مدل OSI از هفت لایه مجزا ولی مرتبط ساخته

1- Open Systems Interconnection
2- Open system

www.Abiri.ir

شده است و هر یک از لایه‌ها، حاوی بخشی از فرایند جابجائی اطلاعات در یک شبکه است (شکل ۲-۱ را ببینید). درک اصول زیر بنائی مدل OSI ، پایه محکمی برای بحث پیرامون مخابره دیتا می‌باشد.



شکل ۲-۱ : مدل OSI

معماری لایه‌ای

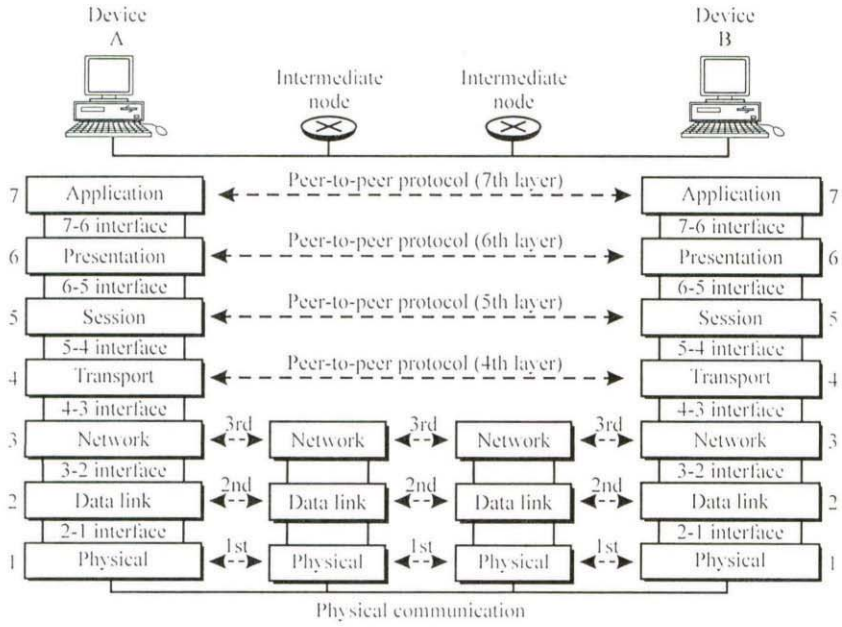
مدل OSI از هفت لایه تشکیل شده است: فیزیکی (لایه ۱)، دیتالینک (لایه ۲) ، شبکه (لایه ۳)، انتقال (لایه ۴)، جلسه کاری (لایه ۵)، نمایش (لایه ۶) و کاربرد (لایه ۷). شکل ۲-۲ نشان می‌دهد در ارسال پیغامی از وسیلهٔ A به وسیله B ، چه لایه‌هایی درگیر می‌شوند. همچنان که پیغام از وسیله A به وسیله B می‌رود، ممکن است از چندین گره میانی عبور کند. غالباً این گره‌های میانی، فقط سه لایه اول مدل OSI را در برمی‌گیرند.

طراحان در ایجاد مدل OSI، فرایند انتقال دیتا را به عمده‌ترین المانهای آن، تجزیه کرده‌اند. آنها تعیین کردند که چه عملیاتی در شبکه، به هم مرتبط می‌باشند و سپس آن عملیات مرتبط را به صورت یک لایه در آورده‌اند. لذا هر لایه، خانواده‌ای از عملیات را شامل می‌شود که متفاوت با خانواده عملیات سایر لایه‌ها می‌باشد. طراحان با دسته‌بندی کردن عملیات مرتبط با هم و قرار دادن هر دسته از عملیات مرتبط در یک لایه، معماری را تولید کردند که هم جامع و هم قابل انعطاف است. مهم‌تر از همه آنکه مدل OSI ، امکان برقراری اتصال بین شبکه‌هایی را فراهم کرده است که بدون مدل OSI ، سیستم‌های ناسازگار و غیرقابل اتصال، محسوب می‌شدند.

هر لایه در یک ماشین منفرد، سرویسهای لایه‌ای که دقیقاً زیر آن لایه قرار گرفته است را به کار می‌بندد. به عنوان مثال لایه ۳، از سرویسهای لایه ۲ استفاده می‌کند و به لایه ۴ سرویس می‌دهد. بین ماشینها ، لایه X از یک ماشین، با لایه X از ماشین دیگر، به مخابره اطلاعات می‌پردازد. این مخابره، تحت حاکمیت یک سری از قوانین مورد توافق ماشینهاست که

www.Abiri.ir

پروتوکل نامیده می‌شوند. پروسه‌ای در هر ماشین در سطح یک لایه، با پروسه نظیر خود در ماشین دیگر، دیتایی را رد و بدل می‌کند. به این دو پروسه، ***پروسه‌های نظیر به نظیر***[1] می‌گویند. بنابراین مخابره بین ماشینها، تحت پروسه‌های نظیر به نظیر صورت می‌گیرد و در آن، از پروتوکل‌های مناسب هر لایه، استفاده می‌شود.



شکل ۲-۲ : لایه‌های مدل OSI

* پروسه‌های نظیر به نظیر

مخابره در سطح لایه فیزیکی، بطور مستقیم صورت می‌گیرد. در شکل ۲-۲ ، وسیله A ، دنباله‌ای از بیتها را برای وسیله B (از طریق گره‌های میانی) ارسال می‌کند. با این حال، برای داشتن مخابره‌ای در سطح لایه‌های بالاتر، ابتدا باید پیغام از میان لایه‌های وسیله A ، پایین آید، خود را به وسیله B برساند و سپس از لایه‌های وسیله B بالا رود. هر لایه در وسیله فرستنده پیغام، اطلاعات خود را به پیغامی که از لایه بالاتر از خود دریافت می‌کند، اضافه می‌نماید و سپس مجموعه حاصل را به لایه پایین‌تر از خود، تحویل می‌دهد.

1- Peer–to–peer processes

www.Abiri.ir

در لایه ۱، کل اطلاعات ارسالی به فرمی در آمده است که قابل ارسال برای وسیله گیرنده می‌باشد. در وسیله گیرنده، لایه به لایه، اطلاعاتی که به پیغام اصلی اضافه شده‌اند، از آن برداشته می‌شوند. به عنوان مثال ،لایه ۲ ماشین گیرنده، اطلاعاتی که لایه ۲ ماشین فرستنده به پیغام اضافه کرده است را از آن برمی‌دارد و آنچه باقی می‌ماند را به لایه بالاتر از خود، یعنی لایه ۳ تحویل می‌دهد. لایه ۳ ماشین گیرنده، اطلاعاتی که لایه ۳ ماشین فرستنده به پیغام اضافه کرده است را از آن بر می‌دارد و آنچه باقی می‌ماند را به لایه بالاتر از خود، یعنی لایه ۴، تحویل می‌دهد و این روند، در لایه‌های بالاتر ادامه می‌یابد.

* رابط بین لایه‌ها

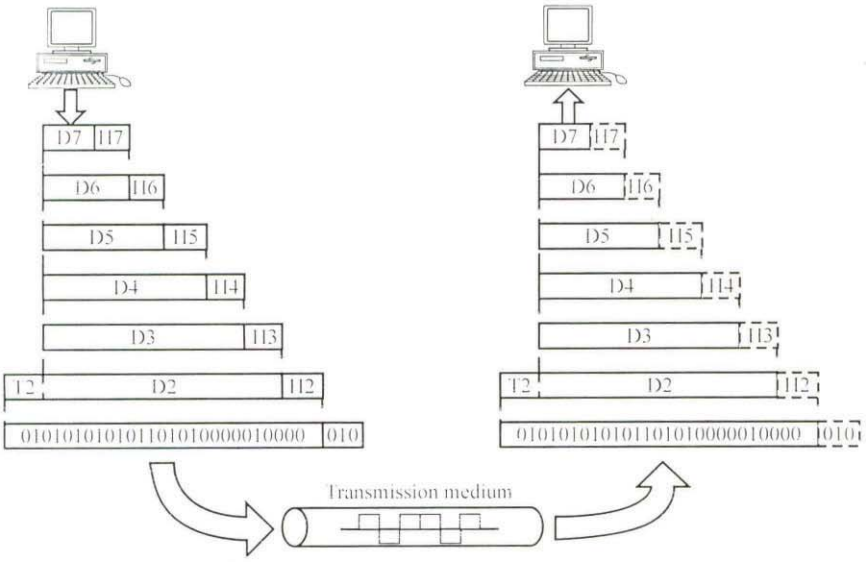
پایین آمدن دیتا از میان لایه‌های ماشین فرستنده و اضافه شدن اطلاعات شبکه به آن، همچنین بالا رفتن دیتا از میان لایه‌های ماشین گیرنده و حذف اطلاعات شبکه از آن، از طریق وجود رابطی بین هر جفت از لایه‌های مجاور هم صورت می‌گیرد. هر رابط، مشخص می‌کند که یک لایه، چه اطلاعات و سرویس‌هایی را باید به لایه بالاتر از خود بدهد. رابطها و عملیات لایه‌ها که به خوبی تعریف شده باشند، ویژگی ماجولار بودن را به شبکه می‌دهند. تا زمانی که یک لایه بتواند سرویسهای مورد نظر را به لایه بالاتر از خود بدهد، نحوه پیاده‌سازی عملیات آن لایه را می‌توان بدون اعمال تغییراتی به سایر لایه‌ها، اصلاح کرد و یا بطورکامل، جایگزین نمود.

* سازمان‌دهی لایه‌ها

می‌توان این طور تصور کرد که هفت لایه مدل OSI ، متعلق به سه زیرگروه می‌باشند. لایه‌های ۱ و ۲ و۳ (فیزیکی ، دیتالینک و شبکه) لایه‌هایی هستند که از شبکه پشتیبانی می‌کنند. آنها به جنبه‌های فیزیکی جابجائی دیتا، از یک وسیله به وسیله دیگر، می‌پردازند (مانند مشخصه‌های الکتریکی، اتصالات فیزیکی، آدرس‌دهی فیزیکی، زمان جابجائی دیتا و قابلیت اطمینان) . می‌توان لایه های ۵ ، ۶ ، ۷ (جلسه کاری، نمایش و کاربرد) را لایه‌هایی در نظر گرفت که از کاربر، پشتیبانی می‌کنند. آنها امکان برقراری ارتباط بین سیستم‌های نرم‌افزاری متفاوت را فراهم می‌کنند. لایه ۴ (لایه انتقال) دو زیرگروه یاد شده را به هم پیوند می‌زند و این اطمینان را بوجود می‌آورد که آنچه لایه‌های زیرین منتقل کرده‌اند، به شکلی بوده است که برای لایه‌های بالاتر، قابل استفاده می‌باشد. تقریباً همواره، لایه‌های بالاتر مدل OSI ، به صورت نرم‌افزاری پیاده‌سازی می‌شوند. لایه‌های پایین‌تر، ترکیبی از سخت‌افزار و نرم‌افزار هستند و البته لایه فیزیکی از این امر، مستثنی است، زیرا که اکثراً به صورت سخت‌افزاری پیاده‌سازی می‌شود.

www.Abiri.ir

شکل ۲-۳ ، یک دید کلی از لایه‌های مدل OSI را ارائه می‌دهد. دراین شکل ، D7 به معنی واحد دیتا درلایه ۷، D6 به معنی واحد دیتا در لایه ۶ و الی آخر می‌باشد. عملیات نقل و انتقال پیغام از لایه ۷ (لایه کاربرد) شروع می‌شوند و سپس لایه به لایه به سمت پایین، ادامه می‌یابند. در هر لایه، هدری به واحد دیتا اضافه می‌شود. در لایه ۲، دنباله‌ای نیز به واحد دیتا اضافه می‌شود. وقتی که دیتای فرمت شده، از لایه فیزیکی (لایه ۱) عبور می‌کند، به سیگنال الکترومغناطیس تبدیل می‌شود و در امتداد یک لینک فیزیکی، منتقل می‌شود.



شکل ۲-۳ : جابجائی پیغام در مدل OSI

بعد از آنکه سیگنال به مقصد رسید، از میان لایه ۱ عبور می‌کند و مجدداً به صورت دنباله‌ای از بیتها در می‌آید. سپس دیتای دریافتی، از لایه‌های وسیله گیرنده بالا می‌رود. همچنان که دیتای دریافتی، از یک لایه به لایه بالاتر از خود می‌رود، عملیات مورد نظر در آن لایه، انجام می‌شوند، هدر و دنباله‌ای که لایه متناظر با آن لایه، در فرستنده، به دیتا اضافه کرده بود، از دیتا برداشته می‌شوند و سپس آنچه که باقی می‌ماند، به لایه بالاتر، تحویل داده می‌شود. وقتی که دیتا به لایه ۷ می‌رسد، مجدداً به فرمی است که برای گیرنده پیغام، مناسب می‌باشد و به این ترتیب، پیغام ارسالی به دست گیرنده نهائی، می‌رسد.

www.Abiri.ir

* کپسول‌بندی

شکل ۲-۳ جنبه دیگری از مخابره دیتا در مدل OSI را نشان می‌دهد: ***کپسول‌بندی***[1]. دیتای لایه ۷، در لایه پایین‌تر از خود یعنی لایه ۶ ، کپسول‌بندی می‌شود. کل دیتای لایه ۶ ، در لایه پایین‌تر از خود، یعنی لایه ۵ ، کپسول‌بندی می‌شود و الی آخر.

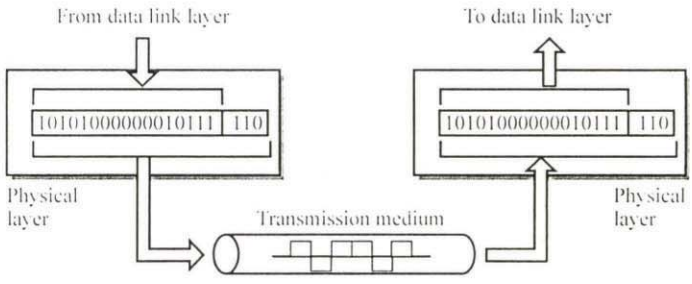
به عبارت دیگر، بخش دیتای یک بسته در لایه N ، کل دیتای دریافتی از لایه بالاتر (لایه N – 1) که شامل اطلاعات و هدرها و دنباله‌های اضافه شده به آن می‌باشد را در برمی‌گیرد. به این عمل، کپسول‌بندی می‌گویند، زیرا لایه N ، نمی‌داند چه بخشی از دیتایی که حمل می‌کند، اطلاعات اصلی و چه بخشی، هدرها و دنباله‌هایی است که به آن اضافه شده‌اند. از نظر لایه N ،کل دیتایی که از لایه بالاتر یعنی لایه N –1 می‌آید، به عنوان یک واحد دیتا تلقی می‌شود و لایه N ، به هیچ عنوان نمی‌داند که ساختار دیتای دریافتی از لایه بالاتر، چیست.

۲-۲ لایه‌ها در مدل OSI

در این بخش، به اختصار عملیات لایه‌های مختلف مدل OSI را شرح می‌دهیم.

لایه فیزیکی

لایه فیزیکی، عملیات لازم برای جابجائی دنباله‌ای از بیتها روی یک محیط انتقال فیزیکی را، در برمی‌گیرد. او شامل مشخصه‌های الکتریکی و مکانیکی رابط و محیط انتقال است. همچنین این لایه، عملیاتی را تعریف می‌کند که وسائل فیزیکی و رابطها برای جابجائی دنباله بیتها، باید انجام دهند. شکل ۲-۴ ، جایگاه لایه فیزیکی را نسبت به محیط انتقال و لایه دیتالینک، نشان می‌دهد.



شکل ۲-۴ : لایه فیزیکی

1- Encapsulation

www.Abiri.ir

مهم : لایه فیزیکی، مسئول جابجائی بیتها از یک گره به گره بعدی است.

همچنین لایه فیزیکی، به موارد زیر رسیدگی می‌کند:

*** مشخصه فیزیکی رابط و محیط انتقال**

لایه فیزیکی، مشخصه رابط بین وسائل و محیط انتقال را تعریف می‌کند. همچنین این لایه، نوع محیط انتقال را نیز تعیین می‌کند (فصل سوم را ببینید).

*** نمایش بیتها**

دیتای لایه فیزیکی، دنباله‌ای از بیتها (دنباله صفرها و یکها) بدون هیچگونه تفسیری از آنهاست. برای انتقال بیتها، باید آنها را به صورت سیگنال الکتریکی یا نوری درآوریم. لایه فیزیکی، چگونگی تبدیل صفرها و یکها به سیگنال را مشخص می‌کند. به این عمل، ***سیگنال‌سازی***[1] می‌گویند.

*** نرخ بیت**

لایه فیزیکی، تعداد بیتی که در هر ثانیه ارسال می‌شود را نیز تعیین می‌کند. به عبارت دیگر لایه فیزیکی، طول زمانی یک بیت را تعریف می‌کند.

*** سنکرون کردن بیتها**

نه تنها فرستنده و گیرنده باید با یک نرخ بیت یکسان کار کنند، بلکه باید در سطح یک بیت، با یکدیگر سنکرون باشند. به عبارت دیگر، کلاکهای فرستنده و گیرنده باید با هم سنکرون باشند.

*** پیکربندی خط**

لایه فیزیکی باید وسائل را به محیط انتقال متصل کند. در پیکربندی خط از نوع ***نقطه به نقطه***[2]، دو وسیله از طریق یک لینک اختصاصی به هم متصل می‌شوند. در پیکربندی خط از نوع ***چند نقطه***[3] ، چندین وسیله بطور مشترک از یک لینک استفاده می‌کنند.

1-Encoding
2- point-to-point
3- multipoint

www.Abiri.ir

* توپولوژی فیزیکی

توپولوژی فیزیکی، نحوه اتصال وسایل و لینکهای یک شبکه را مشخص می‌کند. می‌توان وسائل را با توپولوژی ***مش***[1] (هر وسیله به وسیله دیگر متصل شود)، توپولوژی ***ستاره***[2] (وسایل از طریق یک وسیله مرکزی به هم متصل شوند)، توپولوژی ***حلقه***[3] (هر وسیله به وسیله همسایه متصل شود و همه وسایل تشکیل یک حلقه دهند)، توپولوژی ***باس***[4] (همه وسایل به یک لینک مشترک متصل شوند) و سایر توپولوژیها به هم وصل کرد.

* مد انتقال

لایه فیزیکی، جهت انتقال اطلاعات بین دو وسیله را نیز تعریف می‌کند که می‌تواند به صورت ***یک طرفه***[5] ، ***نیمه دوطرفه***[6] یا ***کاملاً دوطرفه***[7] باشد. در مد یک طرفه، فقط یک وسیله می‌تواند اطلاعات را ارسال کند و وسیله دیگر فقط می‌تواند اطلاعات را دریافت نماید. بنابراین مد یک طرفه، مخابره یک طرفه اطلاعات می‌باشد. در مد نیمه دوطرفه، دو وسیله می‌توانند هم اطلاعات بفرستند و هم دریافت کنند اما نه بطور هم زمان. در مد کاملاً دوطرفه، هر دو وسیله می‌توانند بطور هم زمان، اطلاعات ارسال و دریافت کنند.

لایه دیتالینک

لایه فیزیکی که صرفاً اطلاعات را منتقل می‌کند، به کمک لایه دیتالینک، به یک لینک ارتباطی قابلیت اطمینان تبدیل می‌شود و به این ترتیب، لایه بالاتر (لایه شبکه)، لایه فیزیکی را بدون خطا خواهد دید. شکل ۲-۵ رابطه بین لایه دیتالینک، لایه فیزیکی و لایه شبکه را نشان می‌دهد.

مهم: لایه دیتالینک، مسئول جابجائی فریمها از یک گره به گره دیگر است.

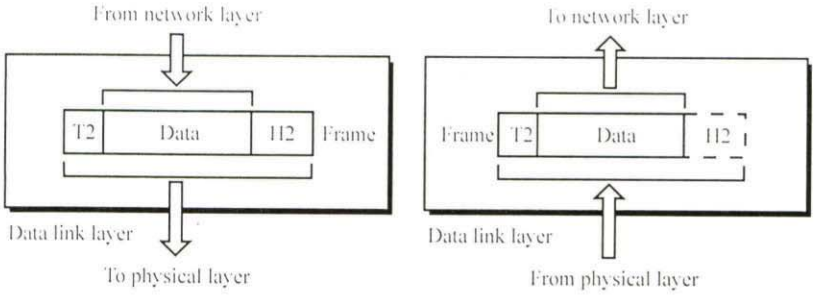
سایر وظایف لایه دیتالینک شامل موارد زیر است:

1- mesh
2- star
3- ring
4- bus
5- Simplex
6- Half duplex
7- Full duplex

www.Abiri.ir

❊ فریم سازی

لایه دیتالینک، دیتایی که از لایه شبکه دریافت کرده است را به صورت واحدهای دیتای قابل مدیریت، به نام ***فریم***[1] ، در می‌آورد.



شکل ۲-۵ : لایه دیتالینک

❊ آدرس‌دهی فیزیکی

لایه دیتالینک به منظور رساندن فریمها به دست وسایل مختلف متصل به یک شبکه واحد، آدرس فرستنده و گیرنده را در هدر فریم می‌نویسد. اگر قرار است که فریم به دست وسیله‌ای خارج از آن شبکه برسد، آدرس گیرنده، آدرس روتری خواهد بود که شبکه فرستنده را به شبکه بعدی وصل می‌کند.

❊ کنترل فلو

اگر سرعت پردازش دیتا در گیرنده، کمتر از سرعت ارسال دیتا توسط فرستنده باشد، لایه دیتالینک، مکانیزم کنترل فلو را اعمال می‌کند تا مانع از سرریز شدن گیرنده از دیتا گردد.

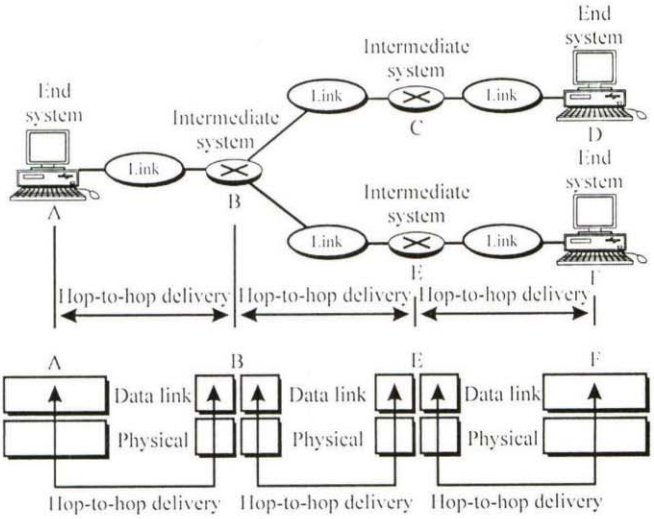
❊ کنترل خطای بیت

لایه دیتالینک، به لایه فیزیکی قابلیت اطمینان می‌دهد و این کار را با وارد کردن مکانیزمی، جهت آشکارسازی خطای بیت و ارسال مجدد فریمهای مخدوش شده، انجام می‌دهد. معمولاً کنترل خطای بیت، با اضافه کردن دنباله‌ای به انتهای فریم، حاصل می‌گردد.

1- Frame

www.Abiri.ir

وقتی که دو یا چند وسیله، به لینک مشترکی متصل می‌کردند، لایه دیتالینک کار
ند تا مشخص باشد در هر لحظه، چه کسی می‌تواند از آن لینک مشترک، استفاده کند
، ۲-۶ ، تحویل ***گره به گره***[1] یا ***هاپ به هاپ***[2] اطلاعات توسط لایه دیتالینک را نشار
هد.



شکل ۲-۶ : تحویل هاپ به هاپ

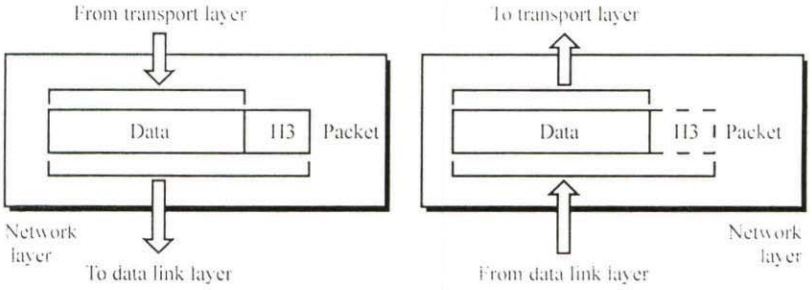
شبکه

لایه شبکه، مسئول تحویل یک بسته اطلاعات از ماشین مبدأ به ماشین مقصد است
ن است که این ماشین‌ها، به شبکه‌های مختلفی متصل شده باشند. در حالی که لای
بنک، ناظر بر انتقال اطلاعات، بین دو ماشین متصل به یک شبکه واحد می‌باشد، لای
، بر انتقال اطلاعات، از یک ماشین در یک شبکه، به ماشین دیگری در شبکه دیگری
ت دارد.

اگر دو ماشین یا سیستم، به لینک مشترکی متصل شده باشند، غالباً نیازی به لای
، نخواهد بود. اما اگر دو سیستم، به شبکه‌های مختلفی متصل شده باشند و ارتباط بین
شبکه‌ها، از طریق روترها برقرار شده باشد، لایه شبکه مورد نیاز خواهد بود تا اطلاعات را ا

www.Abiri.ir

ماشین مبدأ در یک شبکه، به ماشین مقصد در شبکه دیگری برساند. شکل ۲-۷ رابطه؛لایه شبکه را با لایه دیتالینک و لایه انتقال نشان می‌دهد.



شکل ۲-۷ : لایه شبکه

مهم: لایه شبکه، مسئول تحویل بسته‌های اطلاعات از ماشین مبدأ در یک شبکه، به ماشین مقصد در شبکه دیگر است.

سایر مسئولیتهای لایه شبکه به قرار زیر هستند:

* آدرس‌دهی منطقی

آدرس‌دهی فیزیکی که توسط لایه دیتالینک پیاده‌سازی می‌شود، بطور محلی به مسئله آدرس‌دهی می‌پردازد. اگر بسته‌ای از مرز شبکه عبور کند، به سیستم آدرس‌دهی دیگری برای مشخص کردن سیستم های مبدأ و مقصد نیاز داریم. لایه شبکه، به اطلاعاتی که از لایه بالاتر از خود دریافت می‌کند، هدری اضافه می‌کند که علاوه بر سایر موارد، حاوی آدرس منطقی فرستنده و گیرنده است.

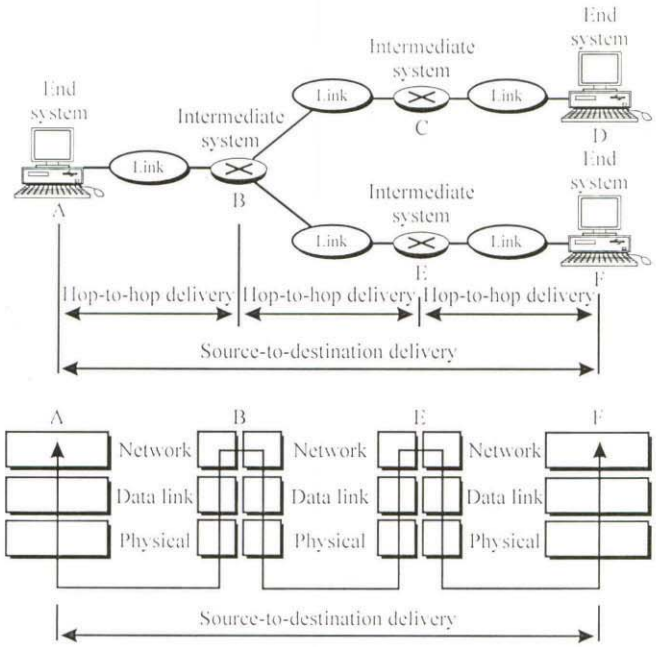
* مسیردهی

وقتی که شبکه‌های مستقل، از طریق روترها به هم متصل می‌شوند تا یک شبکه بزرگتر را بسازند (که به آن، ***شبکه‌بندی***[1] گفته می‌شود)، این وظیفه روترها خواهد بود که بسته‌های اطلاعات را به سمت مقصد مورد نظر، مسیردهی کنند. یکی از عملیات لایه شبکه،

1- internetworking

www.Abiri.ir

فراهم کردن این مکانیزم است. شکل ۲-۸ ، تحویل ***پایانه به پایانه***[1] اطلاعات توسط لایه شبکه را نشان می‌دهد.



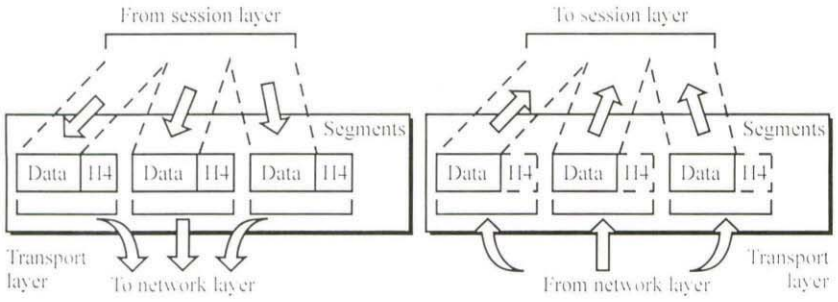
شکل ۲-۸ : تحویل پایانه به پایانه اطلاعات

لایه انتقال

لایه انتقال، مسئول تحویل ***پروسه به پروسه***[2] کل پیغام است. یک پروسه (یا برنامه)، نرم‌افزاری است که روی ماشین، در حال اجراست. با وجود آنکه لایه شبکه، مسئول تحویل اطلاعات از ماشین مبدأ به ماشین مقصد است، لایه شبکه، هیچگونه آگاهی از رابطه مابین بسته‌های جداگانه اطلاعات، ندارد. لایه شبکه با هر بسته، به صورت یک بسته مجزا رفتار می‌کند و کاری ندارد که این بسته‌ها، متعلق به یک پیغام و یا پیغامهای متفاوتی هستند. اما لایه انتقال این اطمینان را بوجود می‌آورد که کل پیغام، صحیح و سالم از برنامه مبدأ، به دست برنامه مقصد برسد و در این نقل و انتقال، کنترل فلو و کنترل خطا را نیز اعمال می‌کند. شکل ۲-۹ ، رابطه لایه انتقال با لایه شبکه و لایه جلسه کاری را نشان می‌دهد.

1- end-to-end
2- process-to-process

www.Abiri.ir



شکل ۲-۹ : لایه انتقال

مهم : لایه انتقال، مسئول تحویل پیغام از برنامه مبدأ به برنامه مقصد است.

سایر مسئولیتهای لایه انتقال، شامل موارد زیر است:

* آدرس دهی نقطه سرویس

غالباً به طور همزمان، برنامه‌های متعددی در یک کامپیوتر در حال اجر هستند. به همین دلیل، تحویل اطلاعات از مبدأ به مقصد، نه تنها شامل تحویل اطلاعات از ماشین مبدأ به ماشین مقصد است، بلکه تحویل اطلاعات از برنامه مبدأ به برنامه مقصد را نیز در برمی‌گیرد. بنابراین هدر لایه انتقال، باید شامل نوعی از آدرس به نام آدرس نقطه سرویس و یا پورت باشد. لایه شبکه، یک بسته اطلاعات را به ماشین مقصد می‌رساند و لایه انتقال، کل پیغام را به برنامه مقصد (در ماشین مقصد) می‌رساند.

* سگمنت‌سازی و مونتاژ مجدد سگمنتها

هر پیغام به سگمنتهای قابل انتقال تفکیک می‌شود و هر سگمنت حاوی یک ***شماره ترتیب***[1] است. شماره ترتیبها به لایه انتقال کمک می‌کنند تا سگمنتهای دریافتی در مقصد را با همان ترتیب اولیه، مونتاژ کند و نیز تشخیص دهد که چه سگمنتهایی حین انتقال ،گم شده‌اند و لذا مجدداً آنها را از فرستنده، درخواست نماید.

1- Sequence number

www.Abiri.ir

* کنترل اتصال

لایه انتقال می‌تواند ***اتصال‌گرا*** [1] یا ***فاقد اتصال*** [2] باشد. انتقال فاقد اتصال با هر دیتاگرام، به صورت یک واحد اطلاعات منفرد، جداگانه و مستقل رفتار می‌کند و آن را مستقل از سایر دیتاگرامها، به برنامه مقصد تحویل می‌دهد. در انتقال اتصال‌گرا، ابتدا با لایه انتقال ماشین مقصد، اتصالی برقرار می‌شود و سپس سگمنتهای پیغام برای او، ارسال می‌شوند. پس از آنکه کل اطلاعات ارسال شد، اتصال قطع می‌گردد.

* کنترل فلوی سگمنتها

لایه انتقال نیز مانند لایه دیتالینک، مسئول کنترل فلو می‌باشد. در لایه دیتالینک، کنترل فلو مابین دو ماشین متصل به یک لینک یا یک شبکه واحد است اما در لایه انتقال، کنترل فلو مابین دو برنامه است که روی دو ماشین متصل به دو شبکه متفاوت، در حال اجرا هستند.

* کنترل خطای سگمنتها

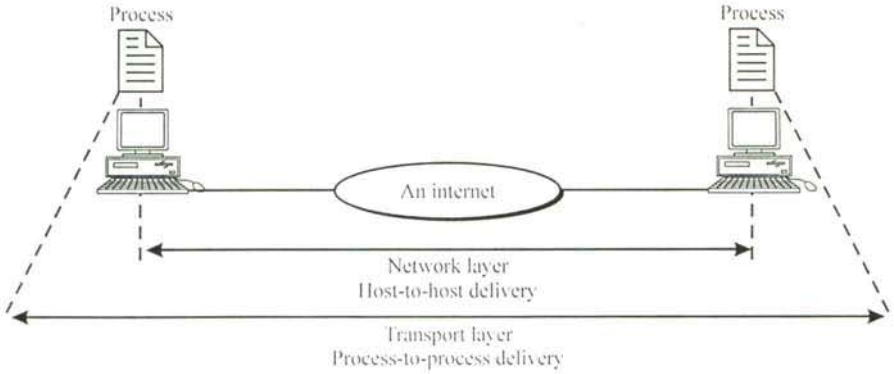
مانند لایه دیتالینک، لایه انتقال نیز مسئول کنترل خطاست. در لایه دیتالینک، کنترل خطا مابین دو ماشینی است که روی یک لینک یا یک شبکه واحد واقع شده‌اند اما در لایه انتقال، کنترل خطا مابین دو برنامه است که روی دو ماشین متصل به دو شبکه جداگانه، در حال اجرا هستند. لایه انتقال این اطمینان را بوجود می‌آورد که کل پیغام، بدون هیچ گونه خطائی (مخدوش شدن، گم شدن یا تکرار شدن) به دست لایه انتقال ماشین گیرنده، برسد. اگر خطائی رخ داده باشد، غالباً درخواست ارسال مجدد، به فرستنده داده خواهد شد. شکل ۲-۱۰، تحویل پروسه به پروسه پیغام را نشان می‌دهد.

لایه جلسه‌کاری

سرویسهایی که توسط اولین سه لایه (فیزیکی، دیتالینک و شبکه) فراهم شده‌اند، برای بعضی از برنامه‌ها، کافی نیستند. لایه جلسه‌کاری، کنترل‌کننده دیالوگها در شبکه است. او ارتباطی بین سیستم‌های مخابراتی، برقرار می‌کند و کنترلی بر پیغامهای رد و بدل شده، اعمال می‌کند. وظایف لایه جلسه‌کاری به شرح زیرند:

1- Connetion-oriented
2- Connectionless

www.Abiri.ir



شکل ۲-۱۰ : تحویل پروسه به پروسه پیغام

❊ کنترل دیالوگ

لایه جلسه‌کاری، به دو سیستم امکان برقراری دیالوگی را می‌دهد. او یک مخابره نیمه دو طرفه و یا کاملاً دو طرفه را مابین دو برنامه، امکان‌پذیر می‌نماید.

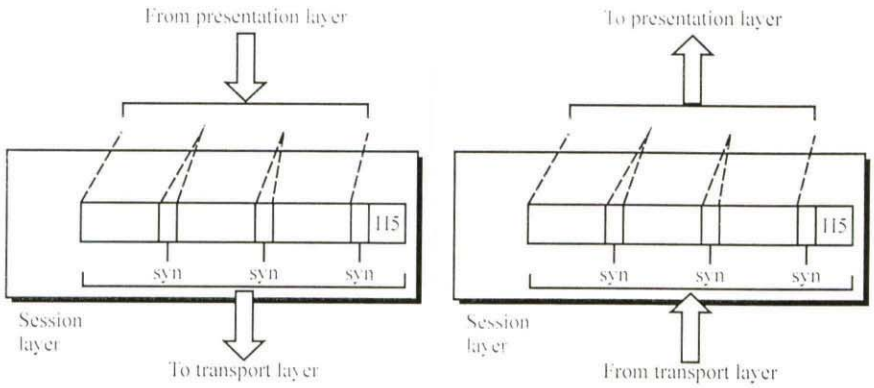
❊ سنکرونیزاسیون

لایه جلسه‌کاری، این امکان را به برنامه می‌دهد که ***نقاط سنکرونیزاسیون***[1] را به داخل پیغام، وارد نماید. به عنوان مثال، اگر قرار است یک فایل ۲۰۰۰ صفحه‌ای ارسال گردد، عاقلانه است که بعد از هر ۱۰۰ صفحه، نقاط سنکرونیزاسیون گذاشته شوند تا این اطمینان حاصل گردد که هر واحد ۱۰۰ صفحه‌ای، به صورت صحیح و سالم، به دست گیرنده رسیده است، زیرا که گیرنده باید تأییدیه‌ای، مبنی بر صحت دریافت آنها، ارسال نماید. اگر به عنوان مثال در این حالت، حین انتقال صفحه ۵۲۳ خرابی در سیستم رخ دهد، پس از برطرف کردن خرابی، صرفاً کافی است که صفحه‌های ۵۰۱ تا ۵۲۳ مجدداً برای گیرنده ارسال شوند و لازم نیست که صفحات قبل از صفحه ۵۰۱ نیز، مجدداً برای گیرنده ارسال شوند. شکل ۲-۱۱، رابطه بین لایه جلسه‌کاری با لایه‌های انتقال و نمایش را نشان می‌دهد.

مهم: لایه جلسه‌کاری، مسئول کنترل دیالوگ و گذاشتن نقاط سنکرونیزاسیون در پیغام است.

1- Synchronization points

www.Abiri.ir



شکل ۲-۱۱ : لایه جلسه کاری

لایه نمایش

لایه نمایش، ساختار و معنای اطلاعات جابجا شده بین دو سیستم را، تعیین می‌کند. شکل ۲-۱۲، رابطه بین لایه نمایش، لایه کاربرد و لایه جلسه‌کاری را نشان می‌دهد. وظایف لایه نمایش به شرح زیرند:

* ترجمه

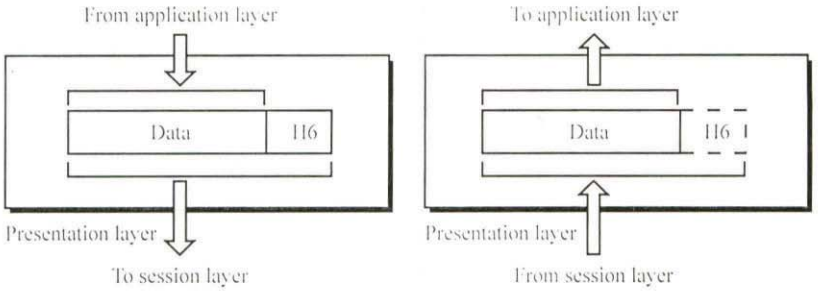
برنامه‌های مبدأ و مقصد، غالباً اطلاعاتی را به فرم دنباله کاراکترها، اعداد و غیره با یکدیگر مبادله می‌کنند. قبل از آنکه اطلاعات ارسال گردند، باید به شکل دنباله‌ای از بیتها درآیند. از آنجا که سیستم‌های مختلف از کدینگهای مختلف استفاده می‌کنند، این وظیفه لایه نمایش است که امکان مبادله اطلاعات بین روشهای مختلف کدینگ را فراهم نماید. لایه نمایش ماشین فرستنده، فرمت اطلاعات را از فرمتی که وابسته به ماشین فرستنده باشد، خارج می‌کند و به فرمت مشترک قابل فهم، برای لایه نمایش ماشین گیرنده، درمی آورد. لایه نمایش ماشین گیرنده، پس از دریافت اطلاعات، آن را به فرمتی که وابسته به ماشین گیرنده است، درمی‌آورد و سپس آن را در اختیار برنامه ماشین گیرنده قرار می‌دهد.

* رمزگذاری

در انتقال اطلاعات محرمانه توسط سیستم، باید این اطمینان وجود داشته باشد که اطلاعات منتقل شده، به صورت خصوصی باقی می‌مانند. رمزگذاری به این معناست که سیستم، قبل از ارسال اطلاعات، شکل نمایشی آن را به صورت رمز شده درآورد و سپس اطلاعات رمز

www.Abiri.ir

شده را از طریق شبکه، ارسال نماید. در رمزگشایی، عکس عمل فوق انجام می‌شود، یعنی شکل نمایشی اطلاعات را از حالت رمز شده، به شکل اولیه آن، باز می‌گرداند.



شکل ۲-۱۲ : لایه نمایش

* فشرده‌سازی

فشرده‌سازی، تعداد بیتهای موجود در پیغام را کم می‌کند. در انتقال مولتی مدیا که شامل متن، صوت و ویدئو است، فشرده‌سازی اطلاعات، از اهمیت خاصی برخوردار است.

مهم : لایه نمایش، مسئول کد کردن، فشرده‌سازی و رمزگذاری اطلاعات و نیز عملهای معکوس متناظر با هر یک از آنهاست.

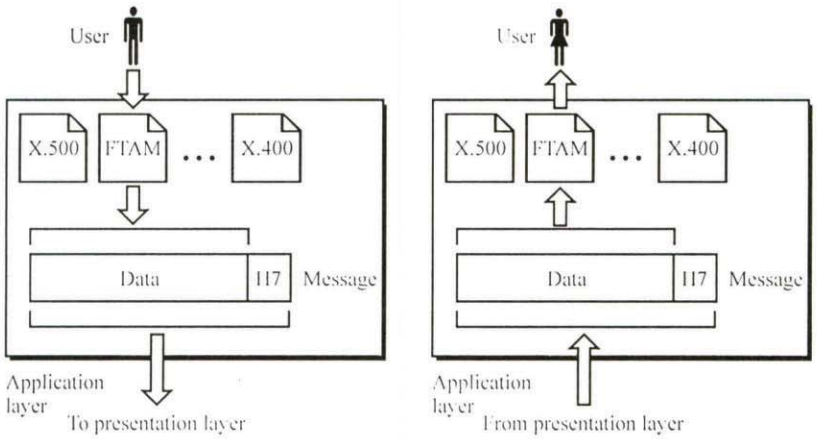
لایه کاربرد

لایه کاربرد به کاربر، انسان یا نرم افزار، امکان دسترسی به شبکه را می‌دهد. او رابطی برای کاربر فراهم می‌کند و سرویسهایی مانند email ، دسترسی به فایل از راه دور، مدیریت دیتابیس‌های به اشتراک گذاشته شده و سایر انواع سرویسهای اطلاعات گسترده را در اختیار کاربر می‌گذارد. شکل ۲-۱۳، رابطه بین لایه کاربرد با کاربر و لایه نمایش را نشان می‌دهد.

در شکل ۲-۱۳، از سرویسهای مختلف موجود برای لایه کاربرد، فقط سه مورد را مشاهده می‌نمایید: X.400 (سرویس email) ، X. 500 (سرویس دایرکتوری) و ***سرویس انتقال، دسترسی و مدیریت فایل***[1] . در این مثال، کاربر با استفاده از سرویس X. 400 ، یک e-mail می‌فرستد. سرویسهایی که لایه کاربرد می‌دهد، شامل موارد زیر است:

1- File Transfer Access Management : FTAM

www.Abiri.ir



شکل ۲-۱۳: لایه کاربرد

❊ ترمینال مجازی شبکه

یک ترمینال مجازی شبکه، نسخه نرم‌افزاری یک ترمینال فیزیکی است و کاربر به کمک آن می‌تواند به یک ***ماشین میزبان***[1] یا کامپیوتر دوردست، متصل گردد. لایه کاربرد برای انجام این کار، یک شبیه‌سازی نرم افزاری از یک ترمینال فیزیکی را در کامپیوتر دوردست ایجاد می‌کند. کاربر از طریق کامپیوتر خود، با ترمینال نرم‌افزاری ایجاد شده در کامپیوتر دوردست، صحبت می‌کند و سپس آن ترمینال مجازی، با خود کامپیوتر دوردست، صحبت می‌نماید و به این ترتیب، ارتباطی بین کاربر و کامپیوتر دوردست، برقرار می‌شود. البته عکس این عمل نیز انجام می‌شود. در این سناریو، کامپیوتر دوردست تصور می‌کند که در حال گفتگو با یکی از ترمینال‌های فیزیکی خود است و لذا به کاربر، امکان برقراری ارتباط را می‌دهد.

❊ FTAM

این پروتوکل لایه کاربرد، امکان دسترسی به یک فایل در کامپیوتر دوردست (برای اعمال تغییراتی در آن و یا خواندن آن)، فراخوانی فایلی از کامپیوتر دوردست (برای استفاده از آن در ماشین محلی) و یا مدیریت و کنترل فایلهای کامپیوتر دوردست را به کاربر می‌دهد.

1- Host

www.Abiri.ir

❊ سرویسهای email

پروتوکل X.400 ، کارهای لازم برای ارسال و دریافت و ذخیره email را انجام می‌دهد.

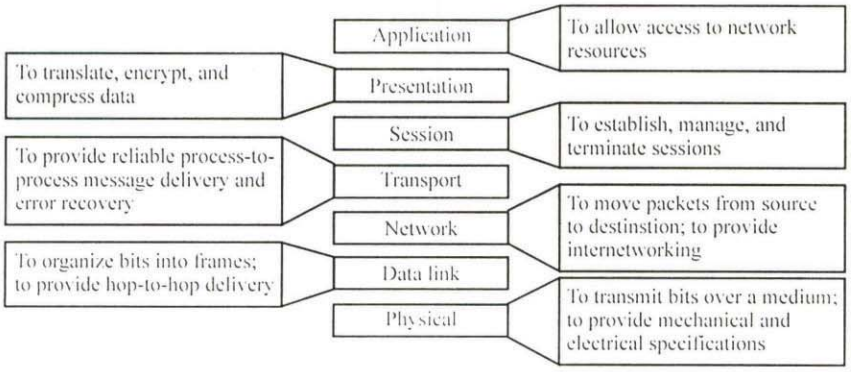
❊ سرویسهای دایرکتوری

پروتوکل X.500 ، امکان دسترسی به اطلاعات گسترده‌ای درباره انواع اشیاء و سرویسهای موجود را فراهم می‌نماید.

مهم: لایه کاربرد، مسئول ارائه سرویس به کاربر می‌باشد.

خلاصه وظایف لایه ها

شکل ۲- ۱۴ ، خلاصه‌ای از وظایف هر لایه را نشان می‌دهد.

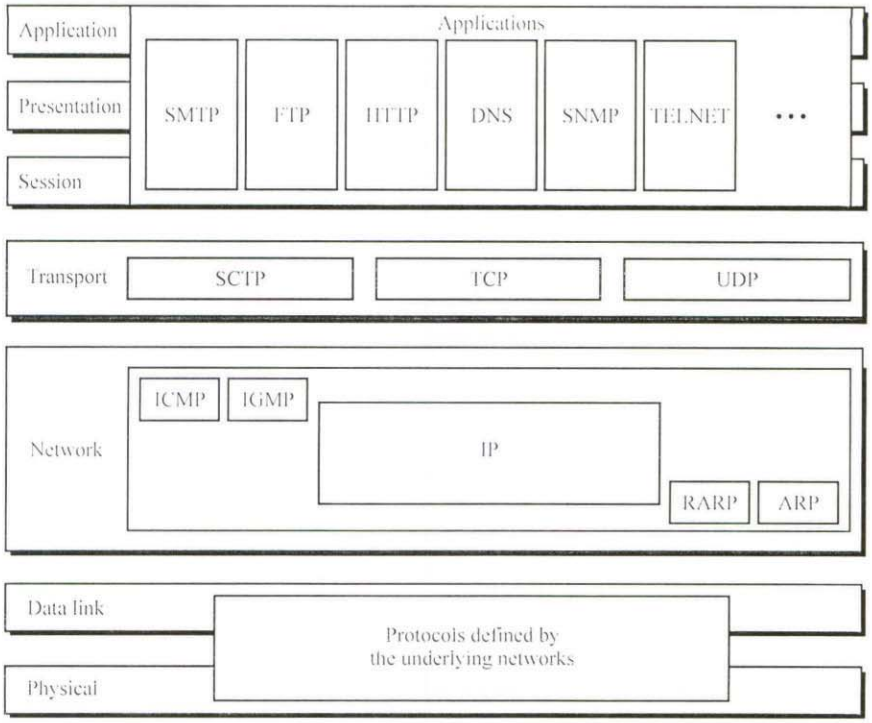


شکل ۲-۱۴ : خلاصه وظایف لایه‌ها

۲-۳ پشته پروتوکل TCP/IP

پشته پروتوکل TCP/IP قبل از مدل OSI ساخته شد. به همین دلیل، لایه‌های پشته پروتوکل TCP/IP، دقیقاً منطبق با لایه‌های مدل OSI نیستند. پشته پروتوکل TCP/IP از پنج لایه ساخته شده است: فیزیکی، دیتالینک، شبکه، انتقال و کاربرد. اولین چهار لایه، عملیات فیزیکی، برقراری ارتباط با شبکه، شبکه‌بندی و عملیات انتقال را انجام می‌دهند که متناظر با چهار لایه اول مدل OSI هستند. اما سه لایه بالایی مدل OSI (جلسه کاری، نمایش و کاربرد) در پشته پروتوکل TCP/IP ، توسط یک لایه به نام لایه کاربرد ، نمایش داده می‌شوند (شکل ۲-۱۵ را ببینید).

www.Abiri.ir



شکل ۲-۱۵ : مدل OSI و پشته پروتوکل TCP/ IP

TCP/IP یک پشته پروتوکل طبقاتی است و از ماجولهایی ساخته شده است و هر ماجول، کار خاصی را انجام می‌دهد. ماجولها به هم وابسته نیستند، اما با هم کار می‌کنند. درحالی که مدل OSI مشخص می‌کند که هر لایه، چه کاری باید انجام دهد، لایه‌های پشته پروتوکل TCP/IP ، حاوی پروتوکلهای نسبتاً مستقلی هستند که بسته به نیاز سیستم، می‌توان تعدادی از آنها را برای انجام کار مورد نظر، انتخاب کرد. منظور از « طبقاتی » این است که هر پروتوکل لایه بالاتر، از یک یا چند پروتوکل لایه پایین‌تر از خود، سرویس می‌گیرد.

پشته پروتوکل TCP/IP، سه پروتوکل را در لایه انتقال تعریف می‌کند: ***پروتوکل کنترل انتقال***[1]، ***پروتوکل دیتاگرام کاربر***[2] و ***پروتوکل انتقال، با کنترل جریان دیتا***[3].

1- Transmission Control Protocol : TCP
2- User Datagram Protocol : UDP
3- Stream Control Transmission Protocol : SCTP

www.Abiri.ir

در لایه شبکه، پروتوکل اصلی که توسط پشته پروتوکل TCP/IP تعریف شده است، ***پروتوکل شبکه‌بندی***[1] است اما باید توجه داشت که در این لایه، پروتوکلهای دیگری را نیز داریم که از جابه‌جایی دیتا در این لایه، پشتیبانی می‌کنند.

لایه‌های فیزیکی و دیتالینک

TCP/IP برای لایه‌های فیزیکی و دیتالینک، هیچ پروتوکل خاصی را معرفی نمی‌کند. او از همه پروتوکل‌های استاندارد و انحصاری که بتوان در این لایه‌ها استفاده کرد، پشتیبانی می‌کند. در شبکه‌بندی به کمک پشته پروتوکل TCP/IP، یک شبکه، هم می‌تواند یک ***شبکه محلی***[2] و هم یک ***شبکه گسترده***[3] باشد.

لایه شبکه

پشته پروتوکل TCP/IP در لایه شبکه، حاوی پروتوکل IP است. IP به نوبه خود از چهار پروتوکل پشتیبان به نامهای ARP ، RARP ، ICMP و IGMP استفاده می‌کند. در فصلهای آتی، به بررسی هر یک از این پروتوکلها خواهیم پرداخت.

* پروتوکل IP

پروتوکلهای پشته پروتوکل TCP/IP ، از پروتوکل IP به عنوان مکانیزم انتقال بسته‌های اطلاعات، استفاده می‌کنند. پروتوکل IP ، یک پروتوکل غیر قابل اطمینان و فاقد اتصال است و در واقع پروتوکلی است که براساس ***نهایت سعی***[4] ، بسته‌ها را به مقصد تحویل می‌دهد. عبارت « نهایت سعی » به این معناست که IP ، خطاهای وارده را ردگیری نمی‌کند. IP فرض را بر این می‌گذارد که لایه‌های پایین‌تر از آن، غیر قابل اطمینان هستند و لذا صرفاً نهایت سعی خود را می‌کند تا بسته‌های اطلاعات را به مقصد برساند، اما هیچ تضمینی در این کار، وجود ندارد.

IP ، دیتا را به صورت ***بسته‌های***[5] جداگانه‌ای، به سمت مقصد مورد نظر ارسال می‌کند. بسته‌ها ممکن است از مسیرهای مختلفی به مقصد برسند و لذا ترتیب دریافت آنها در مقصد، مشابه با ترتیب ارسال آنها از مبدأ، نباشد و یا ممکن است بسته‌ها در راه، تکرار شوند و یا گم شوند. IP گزارشی از مسیر طی شده توسط بسته‌ها را نگه نمی‌دارد و هیچ کاری برای مرتب‌سازی مجدد بسته‌های رسیده به گیرنده، انجام نمی‌دهد.

1- Internetworking Protocol : IP
2- LAN
3- WAN
4- Best-effort
5- Packet

www.Abiri.ir

با این حال، عملیات محدود IP ، نباید نقطه ضعف آن محسوب شوند. IP عملیات اصلی لازم برای انتقال یک بسته، از ماشین مبدأ به ماشین مقصد را انجام می‌دهد و دست کاربر را باز می‌گذارد تا هرگونه عملیات اضافه‌ای که لازم می‌داند را به دلخواه خود، در لایه‌های بالاتر انجام دهد و به این ترتیب، کارآیی IP به حداکثر می‌رسد. به IP ، پروتوکل ***ماشین به ماشین***[1] نیز گفته می‌شود، به این معنی که می‌تواند یک بسته اطلاعات را از ماشین مبدأ به ماشین مقصد برساند. در فصل هشتم، درباره IP صحبت خواهیم کرد.

* پروتوکل ARP

پروتوکل تعیین آدرس یا ***ARP***[2] برای پیدا کردن آدرس فیزیکی متناظر با یک آدرس IP داده شده، به کار می‌رود. هر وسیله‌ای در یک شبکه فیزیکی نوعی مانند یک LAN ، توسط یک آدرس فیزیکی، که غالباً ***داخل کارت شبکه***[3] به صورت سخت افزاری مشخص شده است، شناخته می‌شود. وقتی که آدرس IP ماشین، معلوم باشد، به کمک پروتوکل ARP می‌توان آدرس فیزیکی آن را بدست آورد. در فصل هفتم، درباره ARP صحبت خواهیم کرد.

* پروتوکل RARP

پروتوکل معکوس ARP یا همان ***RARP***[4] به کامپیوتر این امکان را می‌دهد که با دانستن آدرس فیزیکی خود، آدرس IP اختصاص داده شده به او را به دست آورد. وقتی که یک کامپیوتر فاقد هارد دیسک و متصل به شبکه، بوت می‌شود، به کمک این پروتوکل، آدرس IP اختصاص داده شده به او را از RARP Server موجود در آن شبکه، می‌گیرد. در فصل هفتم، درباره RARP صحبت خواهیم کرد.

* پروتوکل ICMP

پروتوکل پیغامهای کنترلی اینترنت[5] مکانیزمی است که به کمک آن، کامپیوترها و روترها می‌توانند، مشکلاتی که حین انتقال بسته‌ها برای بسته‌ها رخ می‌دهند را به اطلاع فرستنده بسته‌ها، برسانند. به کمک ICMP می‌توان در سطح لایه شبکه، پیغامهای گزارش خطا

1- host-to-host
2- Address Resolution Protocol : ARP
3- Network Interface Card : NIC
4- Reverse Address Resolution Protocol : RARP
5- Internet Control Message Protocol : ICMP

www.Abiri.ir

و یا پیغامهای حاوی پرسش و پاسخ، ارسال کرد. در فصل نهم، درباره ICMP صحبت خواهیم کرد.

* پروتوکل IGMP

پروتوکل مدیریت گروهها در اینترنت[1] ، ارسال پیغام برای گروهی از گیرنده‌ها را آسان می‌کند. در فصل دهم، بطور کامل درباره IGMP صحبت خواهیم کرد.

لایه انتقال

در روزهای اول ساخته شدن پشته پروتوکل TCP/IP، در لایه انتقال، دو پروتوکل TCP و UDP را داشتیم. UDP و TCP ، پروتوکل‌هایی در ***سطح انتقال***[2] هستند یعنی می‌توانند پیغامی را از برنامه مبدأ به برنامه مقصد برسانند. به منظور برطرف ساختن نیازهای برخی از برنامه‌های کاربردی جدید، پروتوکل دیگری به نام SCTP در لایه انتقال، تعریف شده است.

* پروتوکل UDP

پروتوکل دیتاگرام کاربر[3] در مقایسه با TCP، پروتوکل ساده‌تری است. UDP یک پروتوکل ***پروسه به پروسه***[4] است که فقط اطلاعات مربوط به پورت مبدأ، پورت مقصد، Checksum و طول پیغام را به دیتایی که از لایه بالاتر خود دریافت کرده است، اضافه می‌نماید. در فصل یازدهم، درباره UDP صحبت خواهیم کرد.

* پروتوکل TCP

پروتوکل کنترل انتقال[5] ، سرویس لایه انتقال را بطور کامل در اختیار پروتوکلهای لایه کاربرد می‌گذارد. TCP پروتوکل قابل اطمینانی در ارسال ***جریان داده‌ها***[6] است. منظور ما از جریان داده‌ها، همان اتصال‌گرا بودن است. قبل از آنکه دیتایی ارسال گردد، اتصالی بین لایه‌های کاربرد ماشین مبدأ و ماشین مقصد، برقرار می‌شود.

1- Internet Group Management Protocol : IGMP
2- Transport level
3- User Datagram Protocol : UDP
4- Process-to-Process
5- Transmission Control Protocol : TCP
6- Stream

www.Abiri.ir

TCP در سمت فرستنده پیغام، جریان داده‌ها یا پیغامها را به واحدهای کوچکتری به نام ***سگمنت***[1] ، تقسیم می‌کند. هر سگمنت، حاوی یک شماره ترتیب است تا بتوان پس از دریافت سگمنتها درگیرنده، آنها را مجدداً مرتب نمود و نیز حاوی یک ***شماره تأییدیه***[2] است تا بتوان صحت سگمنتهای دریافتی را به اطلاع فرستنده رساند. سگمنتها از طریق بسته‌های IP در شبکه، جابجا می‌شوند. در سمت گیرنده، وقتی که سگمنتها از راه می‌رسند، توسط TCP گردآوری می‌شوند و به کمک شماره ترتیب موجود در آنها، مجدداً به صورت پیغام اولیه، مونتاژ می‌شوند. در فصل دوازدهم درباره TCP صحبت خواهیم کرد.

* پروتوکل SCTP

پروتوکل انتقال، با کنترل جریان دیتا[3] از برنامه‌های کاربردی جدید مانند IP Telephony پشتیبانی می‌کند. SCTP یک پروتوکل لایه انتقال است که ویژگیهای خوب پروتوکلهای UDP و TCP را در هم ادغام می‌کند. در فصل سیزدهم، درباره پروتوکل SCTP صحبت خواهیم کرد.

لایه کاربرد

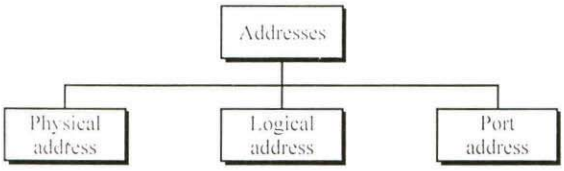
لایه کاربرد در پشته پروتوکل TCP/IP ، معادل با ترکیب سه لایه جلسه کاری ، نمایش و کاربرد در مدل OSI است. در لایه کاربرد پشته پروتوکل TCP/IP، پروتوکلهای زیادی تعریف شده‌اند. در فصلهای آتی، درباره اکثر پروتوکلهای استانداردی که برای این لایه تعریف شده‌اند، صحبت خواهیم کرد.

۲-۴ انواع آدرس‌ها

در اینترنت، از پشته پروتوکل TCP/IP با سه سطح مختلف آدرس‌دهی، استفاده می‌شود: ***آدرس فیزیکی***[4] که به آن آدرس لینک نیز گفته می‌شود، ***آدرس منطقی***[5] که به دلیل استفاده از پروتوکل IP در لایه شبکه، به آن آدرس IP نیز گفته می‌شود و بالاخره ***آدرس پورت***[6]. شکل ۲-۱۶ را ببینید.

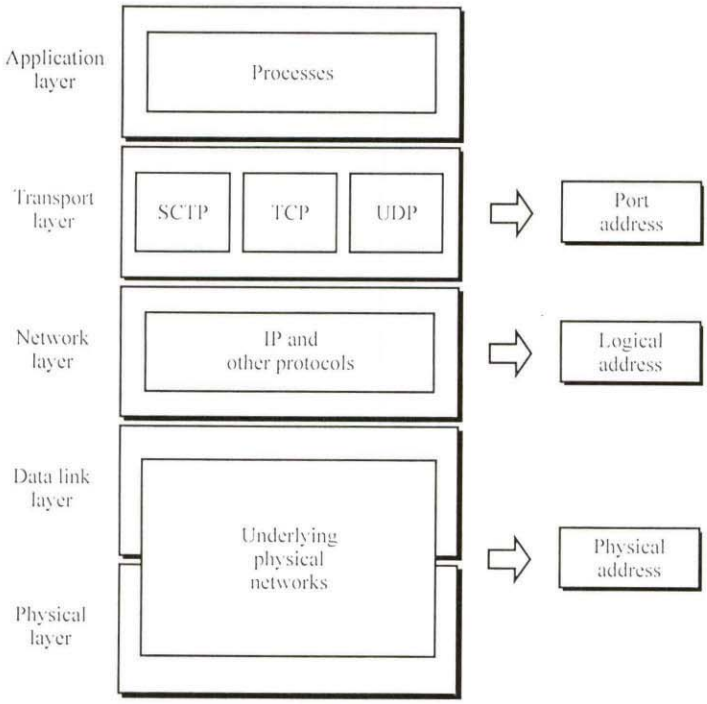
1- Segment
2- Acknowledgment number
3- Stream Control Transmission Protocol : SCTP
4- Physical address
5- Logical address
6- Port address

www.Abiri.ir



شکل ۲-۱۶ : انواع آدرس‌ها در TCP/IP

همانطور که شکل ۲-۱۷ نشان می‌دهد، هر آدرس، متعلق به یک لایه مشخص از ساختار لایه‌ای TCP/IP است.



شکل ۲-۱۷ : رابطه بین لایه‌ها و آدرس‌ها در TCP/IP

آدرس فیزیکی

آدرس فیزیکی که به آن آدرس لینک نیز گفته می‌شود، آدرس یک گره یا یک وسیله در شبکه LAN یا WAN می‌باشد. پروتوکل لایه دیتالینک، این آدرس را در هدر فریمهای خود می‌نویسد. آدرس فیزیکی، پایین‌ترین سطح آدرس است.

www.Abiri.ir

محدوده عمل یک آدرس فیزیکی، در سطح همان شبکه LAN یا WANای است که وسیله حاوی آن آدرس، به آن شبکه، متصل شده است. اندازه و فرمت آدرس‌های فیزیکی، بستگی به نوع شبکه دارد. به عنوان مثال، وسیله متصل به ***شبکه اترنت***[1] از یک آدرس فیزیکی ۶ بایتی (۴۸ بیتی) استفاده می‌کند که روی کارت شبکه آن وسیله، به صورت سخت‌افزاری مشخص شده است. به عنوان یک مثال دیگر، وسیله متصل به شبکه Local Talk از یک آدرس فیزیکی یک بایتی دینامیک استفاده می‌کند و هر زمان که این وسیله راه‌اندازی می‌شود، مقدار این آدرس فیزیکی، تغییر می‌کند.

* آدرس‌های فیزیکی تک‌گیرنده، چندگیرنده و همگانی

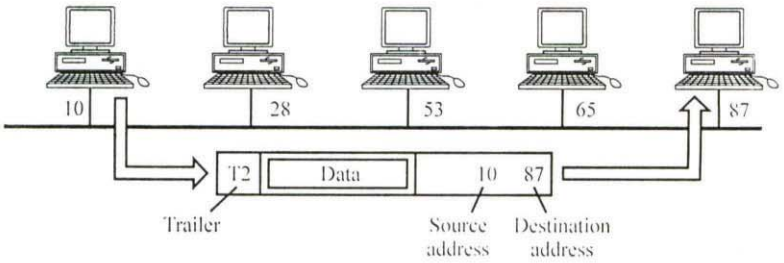
در ***آدرس فیزیکی تک‌گیرنده***[2] ، گیرنده، فقط یک وسیله است. در ***آدرس فیزیکی چندگیرنده***[3] ، گیرنده، یک گروه مشخص شده از وسایل است. در ***آدرس فیزیکی همگانی***[4] ، گیرنده، همه وسایل متصل به یک شبکه می‌باشد. بعضی از انواع شبکه‌ها، از هر سه نوع آدرس استفاده می‌کنند و اترنت یکی از این نوع شبکه‌هاست (فصل سوم را ببینید). برخی دیگر از شبکه‌ها، آدرس‌های چندگیرنده و همگانی ندارند. در این نوع شبکه‌ها، اگر قرار باشد فریمی برای گروهی از وسائل و یا همه وسائل ارسال شود، باید آدرس‌های چندگیرنده و همگانی، به کمک آدرس تک‌گیرنده، شبیه‌سازی شوند. به این معنی که بسته اطلاعات مورد نظر، به کمک آدرس تک‌گیرنده، برای هر یک از گیرنده‌ها به طور جداگانه، ارسال شود.

* مثال ۱

در شکل ۲-۱۸، وسیله‌ای به آدرس فیزیکی ۱۰ ، فریمی را برای وسیله دیگری به آدرس فیزیکی ۸۷، ارسال می‌کند. این دو وسیله توسط یک لینک به هم متصل شده‌اند. هدر این فریم در سطح لایه دیتالینک، حاوی آدرس‌های فیزیکی یاد شده است و اینها، تنها آدرس‌های مورد نیاز می‌باشند. بقیه هدر، حاوی سایر اطلاعات مورد نیاز در سطح لایه دیتالینک است. دنباله فریم، غالباً حاوی بیتهای اضافه‌ای است که برای آشکارسازی خطا به کار می‌روند.

1- Ethernet
2- Unicast physical address
3- Multicast physical address
4- Broadcast physical address

www.Abiri.ir



شکل ۲- ۱۸ : آدرس‌های فیزیکی

*** مثال ۲**

همانطور که در فصل سوم خواهیم دید، اکثر شبکه‌های LAN ، از یک آدرس فیزیکی ۶ بایتی (۴۸ بیتی) استفاده می‌کنند که به صورت دوازده رقم هگزا نوشته می‌شود و مشابه نمونه زیر، هر دو بایت آن توسط یک ***دو نقطه***[1] از بقیه بایتها، جدا می‌شود.

07 : 01 : 02: 01: 2C : 4B
نمونه‌ای از یک آدرس فیزیکی ۶ بایتی (۱۲ رقم هگزا)

آدرس منطقی

آدرس‌های منطقی، برای سرویسهای مخابراتی سراسری که باید مستقل از شبکه‌های فیزیکی زیرساخت کار کنند، مورد نیاز می‌باشند. در یک شبکه‌بندی که از اتصال شبکه‌های مختلف ساخته شده است و هر شبکه، فرمت متفاوتی برای آدرس‌های فیزیکی خود دارد، نمی‌توان به کمک آدرس‌های فیزیکی، یک مخابره سراسری داشت. یک سیستم آدرس‌دهی سراسری که در آن بتوان هر کامپیوتر را، به طور منحصر به فردی و بدون توجه به شبکه فیزیکی زیرساخت آن، آدرس‌دهی کرد، مورد نیاز می‌باشد.

آدرس‌های منطقی برای نیل به همین هدف ساخته شده‌اند. در حال حاضر، یک آدرس منطقی در اینترنت، یک آدرس IP سی و دو بیتی است که می‌تواند بطور منحصر به فردی، یک وسیله متصل شده به اینترنت را مشخص کند. هیچ دو وسیله متفاوتی در اینترنت، نمی‌توانند آدرس منطقی (یا آدرس IP) یکسانی داشته باشند.

1- Colon

www.Abiri.ir

* آدرسهای منطقی تک‌گیرنده، چندگیرنده و همگانی

گیرنده در ***آدرس منطقی تک‌گیرنده***[1]، یک وسیله، در ***آدرس منطقی چندگیرنده***[2] یک گروه مشخص از وسائل، و در ***آدرس منطقی همگانی***[3] همه وسائل متصل به شبکه است. لازم به ذکر است که آدرس‌های همگانی، محدودیتهایی دارند. در فصل چهارم، درباره این سه نوع آدرس منطقی، صحبت خواهیم کرد.

* مثال ۳

در شکل ۲- ۱۹ ، می‌خواهیم دیتایی را از وسیله‌ای با آدرس منطقی A و آدرس فیزیکی ۱۰ متصل به یک شبکه LAN ، به وسیله دیگری با آدرس منطقی P و آدرس فیزیکی ۹۵ متصل به یک شبکه دیگر LAN، ارسال کنیم. از آنجا که این دو وسیله، به شبکه‌های مختلفی متصل شده‌اند، نمی‌توانیم فقط از آدرس‌های فیزیکی استفاده کنیم. آدرس‌های فیزیکی فقط کاربرد محلی دارند. چیزی که در اینجا به آن نیاز داریم، آدرس‌های سراسری هستند که بتوانند از مرز LAN عبور کنند. آدرسهای منطقی، چنین ویژگی دارند. هدر هر بسته اطلاعات در سطح لایه شبکه، حاوی آدرسهای منطقی مبدأ و مقصد بسته است و این آدرسها در طول زمان حرکت بسته از مبدأ تا مقصد، ثابت و بدون تغییر باقی می‌مانند (آدرسهای A و P در شکل ۲-۱۹). وقتی که بسته، از یک شبکه به شبکه دیگر می‌رود، آدرس‌های فیزیکی تغییر می‌کنند اما آدرسهای منطقی، ثابت باقی می‌مانند. در شکل ۲-۱۹ ، بیضی‌هایی که با نام روتر مشخص شده‌اند، وسایل متصل کننده شبکه‌ها هستند و در فصل سوم، درباره آنها صحبت خواهیم کرد.

* مثال ۴

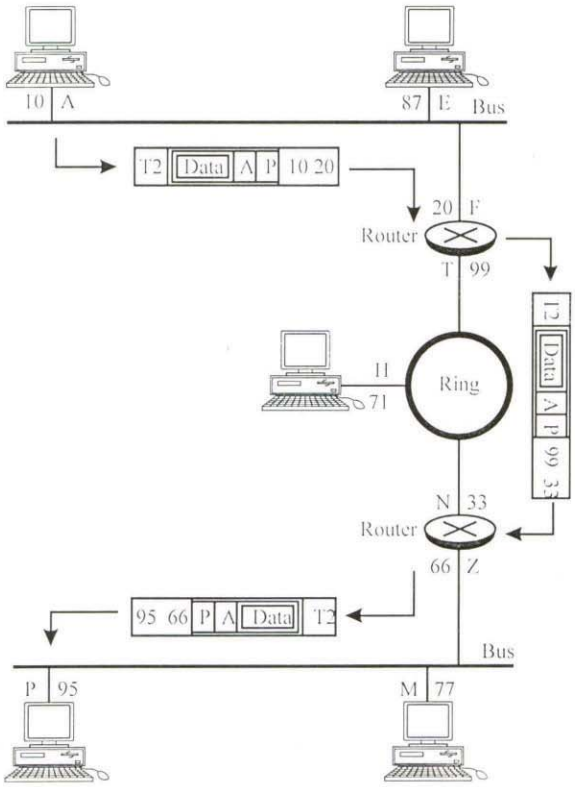
همانطور که در فصل چهارم خواهیم دید، یک آدرس منطقی در اینترنت که از پروتوکل IPv4 استفاده می‌کند، به طول ۳۲ بیت می‌باشد. این آدرس به صورت چهار عدد صحیح نوشته می‌شود، هر عدد معرف یک بایت است و بین اعداد، یک نقطه گذاشته می‌شود. نمونه زیر، مثالی از چنین آدرسی است.

132 . 24 . 75 . 9

یک آدرس اینترنتی به فرمت IPv4 با نماد اعداد صحیح

1- Unicast logical address
2- Multicast logical address
3- Broadcast logical address

www.Abiri.ir



شکل ۲-۱۹ : آدرس‌های IP

آدرس پورت

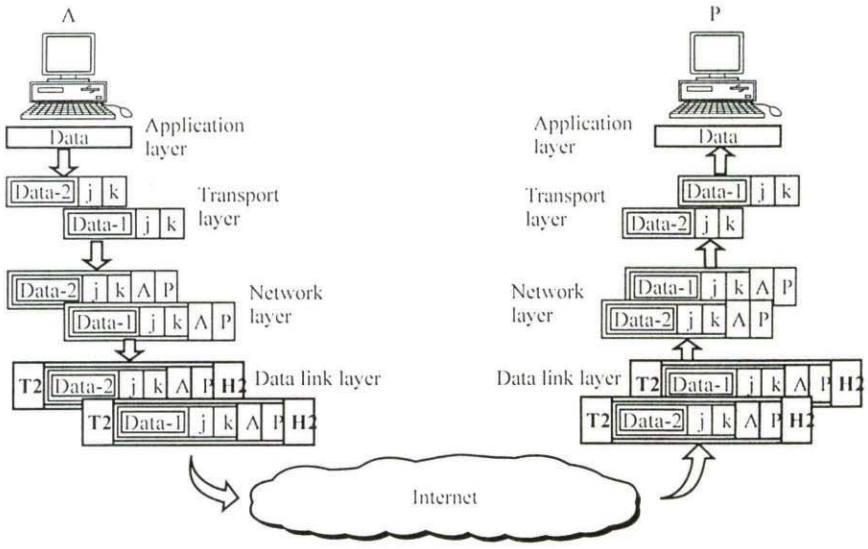
برای آنکه دیتا از ماشین مبدأ به دست ماشین مقصد برسد، هم آدرس فیزیکی و هم آدرس IP مورد نیاز است. با این وجود، صرف رسیدن اطلاعات به ماشین مقصد، هدف نهایی مخابره دیتا در اینترنت، نیست. سیستمی که فقط دیتایی از یک کامپیوتر به کامپیوتر دیگر بفرستد، یک سیستم کامل نیست. امروزه کامپیوترها وسائلی هستند که می‌توانند چندین برنامه را بطور همزمان اجرا کنند. هدف نهائی مخابره دیتا در اینترنت آن است که یک برنامه بتواند به برنامه دیگری، دیتا بفرستد. به عنوان مثال، کامپیوتر A بتواند با کامپیوتر C با استفاده از TELNET ، دیتایی رد و بدل نماید و در همان لحظه، کامپیوتر A بتواند با کامپیوتر B با استفاده از FTP به رد و بدل کردن دیتا بپردازد. برای آنکه چندین برنامه روی یک کامپیوتر بتوانند بطور همزمان از طریق شبکه با سایر برنامه‌ها دیتایی رد و بدل کنند، به آدرسی نیاز داریم که برنامه‌های مختلف روی یک کامپیوتر را از هم تفکیک نماید. در معماری لایه‌ای

www.Abiri.ir

TCP/IP ، آدرسی که به یک برنامه داده می‌شود، **پورت**[1] نامیده می‌شود. آدرس پورت در TCP/ IP به طول ۱۶ بیت است.

*** مثال ۵**

شکل ۲-۲۰ مثالی از مخابره دیتا در سطح لایه انتقال را نشان می‌دهد. دیتایی که از لایه‌های بالاتر می‌آید، حاوی آدرس‌های پورت است که به صورت J (آدرس پورت برنامه فرستنده) و K (آدرس پورت برنامه گیرنده) نشان داده شده‌اند. از آنجا که اندازه دیتای دریافتی از لایه بالاتر، بزرگتر از اندازه‌ای است که لایه شبکه بتواند با آن کار کند، دیتای دریافتی به دو بسته تقسیم شده است و دیتای هر بسته، حاوی همان آدرسهای پورت J , K است. سپس در لایه شبکه، آدرس‌های منطقی (یعنی Λ و P) به هر بسته، اضافه می‌شوند. بسته‌ها می‌توانند مسیرهای مختلفی را طی کنند و لذا به ترتیب و یا خارج از ترتیب اولیه به مقصد برسند. هدرهای این دو بسته، پس از رسیدن به لایه انتقال وسیله گیرنده، حذف می‌شوند و دیتای آنها مجدداً به همان ترتیب اولیه، مونتاژ می‌شود و به لایه بالاتر تحویل داده می‌شود.



شکل ۲-۲۰ : آدرس پورت

1- Port

www.Abiri.ir

*** مثال ۶**

همانطور که در فصلهای یازدهم، دوازدهم و سیزدهم خواهیم دید، آدرس پورت، یک آدرس ۱۶ بیتی است که توسط یک عدد صحیح نمایش داده می‌شود و نمونه‌ای از آن را در زیر می‌بینید:

753

یک آدرس پورت ۱۶ بیتی که به صورت یک عدد صحیح نمایش داده شده است

۲-۵ نسخه‌های مختلف IP

IP در سال ۱۹۸۳ به عنوان پروتوکل رسمی اینترنت (و یا آرپانت آن زمان) شناخته شد. با رشد و توسعه اینترنت، IP نیز توسعه یافت. تاکنون IP ، شش نسخه مختلف را پشت سر گذاشته است. در اینجا، نگاهی به آخرین سه نسخه آن می‌اندازیم.

نسخه ۴

در حال حاضر اکثر شبکه‌های متصل به اینترنت، از ***IP نسخه ۴***[1] استفاده می‌کنند. با این حال، این نسخه از IP ، محدودیتهایی دارد. مشکل اصلی آن است که آدرس IP فقط به طول ۳۲ بیت است و فضای آدرس به کلاسهای مختلفی تقسیم شده است. با رشد سریع اینترنت، این سیستم آدرس‌دهی، نمی‌تواند جوابگوی تعداد رو به رشد کاربران اینترنت باشد.

نسخه ۵

نسخه ۵، پیشنهادی بر مبنای مدل OSI بود. این نسخه هیچگاه از سطح پیشنهاد بالاتر نرفت، زیرا که تغییرات زیادی در لایه‌ها می‌داد و لذا پیاده‌سازی آن، بسیار هزینه بر بود.

نسخه ۶

IETF نسخه جدیدی به نام ***IP نسخه ۶***[2] طراحی کرده است. در این نسخه، تنها پروتوکلهایی که تغییر کرده‌اند، پروتوکل‌های لایه شبکه می‌باشند. IPv4 به IPv6 و ICMPv4 به ICMPv6 تبدیل شده‌اند. IGMP و ARP در ICMPv6 ادغام شده‌اند و RARP حذف شده است.

1- IPv4
2- IPv6

www.Abiri.ir

IPv6 که به آن ***IPng***[1] گفته می‌شود، از آدرسهای ۱۲۸ بیتی (۱۶ بایتی) استفاده می‌کند. تعداد این آدرسها، بسیار بیشتر از آدرسهای ۳۲ بیتی (۴ بایتی) IPv4 است که در حال حاضر به کار می‌روند. بنابراین IPv6 می‌تواند نیاز کاربران بیشتری را برطرف نماید. در IPv6 ، ساختار هدر بسته‌ها، ساده‌تر شده است اما در عین حال، قابلیت انعطاف آن نیز افزایش یافته است تا بتوان در آینده، قابلیتهای بیشتری را به آن اضافه کرد.

IPv6 در سطح لایه شبکه، از ***تشخیص هویت***[2] ، ***حفظ ماهیت دیتا***[3] و ***محرمانه نگهداشتن***[4] آن ، پشتیبانی می‌کند. IPv6 برای جابجائی ***دیتای آنی***[5] مانند صوت و ویدئو و نیز جابجائی دیتای سایر پروتوکلها، طراحی شده است. همچنین IPv6 بهتر از IPv4 می‌تواند با مشکل ازدحام، مقابله کند.

عمده مطالب این کتاب، برمبنای IPv4 است و فصل بیست و هفتم به IPv6 اختصاص داده شده است.

۲-۶ عبارات کلیدی

کنترل دسترسی (access control)
پروتوکل تعیین آدرس (Address Resolution Protocol : ARP)
لایه کاربرد (application layer)
تحویل بر اساس نهایت سعی (best-effort delivery)
بیتها (bits)
آدرس فیزیکی همگانی (broadcast physical address)
توپولوژی باس (bus topology)
فشرده سازی (compression)
کنترل اتصال (connection control)
لایه دیتالینک (data link layer)
کنترل دیالوگ (dialog control)
سرویسهای دایرکتوری (directory services)
سیگنال‌سازی (encoding)
رمزگذاری (encryption)

1- IP next generation
2- Authentication
3- Data integrity
4- Confidentiality
5- Real time

www.Abiri.ir

کنترل خطا (error control)

انتقال، دسترسی و مدیریت فایل
(File Transfer, Access and Management : FTAM)

کنترل فلو (flow control)

فریم‌ها (frames)

مد کاملاً دوطرفه (full-duplex mode)

مد نیمه طرفه (half-duplex mode)

پروتوکل ماشین به ماشین (host–to– host protocol)

رابط (Interface)

سازمان استانداردهای بین‌المللی (International Standards Organization : ISO)

پروتوکل پیغامهای کنترلی اینترنت (Internet Control Message Protocol : ICMP)

پروتوکل مدیریت گروهها در اینترنت
(Internet Group Management Protocol : IGMP)

پروتوکل شبکه‌بندی (Internetworking Protocol : IP)

شبکه‌بندی (Internetwork)

IP نسخه ۴ (IPv4)

IP نسخه ۶ (IPv6)

پیکربندی خط (line configuration)

آدرس منطقی (logical address)

آدرس‌دهی منطقی (logical addressing)

سرویسهای پستی (mail services)

توپولوژی مش (mesh topology)

آدرس فیزیکی چندگیرنده (multicast physical address)

پیکربندی خط از نوع چندنقطه (multipiont line configuration)

پیکربندی خط از نوع نقطه به نقطه (point-to-point line configuration)

لایه شبکه (network layer)

ترمینال مجازی شبکه (network virtual terminal)

سیستم باز (open system)

مدل اتصال سیستمهای باز (Open System Interconnection model: OSI model)

پروسه‌های نظیر به نظیر (peer–to–peer processes)

آدرس فیزیکی (physical address)

لایه فیزیکی (physical layer)

www.Abiri.ir

توپولوژی فیزیکی (physical topology)
آدرس پورت (port address)
لایه نمایش (presentation layer)
تحویل پروسه به پروسه (process–to–process delivery)
معکوس ARP (Reverse address Resolution Protocol : RARP)
توپولوژی حلقه (ring topology)
مسیردهی (routing)
سگمنت‌سازی (segmentation)
آدرس‌دهی نقطه سرویس (service–piont addressing)
لایه جلسه کاری (session layer)
مد یک طرفه (simplex mode)
تحویل از مبدأ به مقصد (source-to-destination delivery)
توپولوژی ستاره (star topology)
پروتوکل انتقال، با کنترل جریان دیتا
(Stream Control Transmission Protocol : SCTP)
نقاط سنکرونسازی (synchronization points)
پشته پروتوکل (TCP/IP stack protocol)
ترجمه (translation)
پروتوکل کنترل انتقال (Transmission Control Protocol : TCP)
مد انتقال (transmission mode)
نرخ انتقال (transmission rate)
لایه انتقال (transport layer)
پروتوکل در سطح انتقال (transport level protocol)
آدرس فیزیکی تک‌گیرندگی (unicast physical address)
پروتوکل دیاگرام کاربر (User Datagram Protocol : UDP)

۲-۷ خلاصه

- ISO مدلی به نام OSI ارائه کرده است که مخابره دیتا بین سیستم‌های مختلف را امکان‌پذیر می‌نماید.
- مدل هفت لایه‌ای OSI، خط مشی‌هایی برای تولید پروتوکل‌های سازگار در سراسر شبکه، ارائه نموده است.

www.Abiri.ir

- لایه‌های فیزیکی، دیتالینک و شبکه، از شبکه پشتیبانی می‌کنند.
- لایه‌های جلسه‌کاری، نمایش و کاربرد، از کاربر پشتیبانی می‌کنند.
- لایه انتقال، لینکی بین لایه‌های پشتیبان شبکه و لایه‌های پشتیبان کاربر، برقرار می‌نماید.
- لایه فیزیکی، عملیات لازم برای جابجائی بیتها روی یک محیط انتقال فیزیکی را با هم هماهنگ می‌سازد.
- لایه دیتالینک مسئول تحویل فریمها، از یک وسیله به وسیله دیگری روی همان شبکه، می‌باشد.
- لایه شبکه مسئول تحویل بسته‌ها، از وسیله مبدأ به وسیله مقصد، با وجود گذر از چندین شبکه میان راهی، می‌باشد.
- لایه انتقال مسئول تحویل پروسه به پروسه کل پیغام است.
- لایه جلسه‌کاری، مخابره دیتا بین وسائل مخابراتی را برقرار می‌کند و تداوم می‌بخشد.
- لایه نمایش با تبدیل فرمت نمایش دیتا به یک فرمت از پیش توافق شده، امکان مخابره دیتا بین وسائل مختلف مخابراتی را فراهم می‌نماید.
- لایه کاربرد، دسترسی به شبکه را برای کاربران امکان‌پذیر می‌نماید.
- TCP/ IP یک پشته پروتوکل طبقاتی پنج لایه است که قبل از مدل OSI ساخته شد.
- لایه کاربرد TCP/IP ، معادل با ترکیب لایه‌های جلسه‌کاری، نمایش و کاربرد درمدل OSI است.
- سیستم‌هایی که از پشته پروتوکل TCP/IP استفاده می‌کنند، سه نوع آدرس را به کار می‌برند. آدرس فیزیکی، آدرس منطقی (یا آدرس IP) و آدرس پورت.
- آدرس فیزیکی که آدرس لینک نیز نامیده می‌شود، آدرس یک وسیله در یک LAN و یا یک WAN است.
- آدرس IP به طور منحصر به فردی، یک وسیله را در اینترنت مشخص می‌کند.
- آدرس پورت، یک پروسه (یا یک برنامه) در حال اجرا در یک کامپیوتر را مشخص می‌کند.
- اکثر شبکه‌ها از IPv4 استفاده می‌کنند.
- قرار است که در آینده نزدیک، IPv6 جایگزین IPv4 شود.

۲-۸ مجموعه تمرین‌ها

تمرین‌ها

۱- ISO و OSI چه ارتباطی با هم دارند؟

www.Abiris.ir

۲- موارد زیر را با یک یا چند لایه از مدل OSI مطابقت دهید.

الف – مسیردهی

ب- کنترل فلو

ج – رابط به محیط انتقال

د- اتصال کاربر به شبکه

۳- موارد زیر را با یک یا چند لایه از مدل OSI مطابقت دهید.

الف – تحویل قابل اطمینان پروسه به پروسه پیغام

ب – انتخاب مسیر

ج- تعریف فریمها

د- سرویسهایی مانند email و انتقال فایل را به کاربر می‌دهد.

هـ- دنباله بیتها را در امتداد محیط انتقال، جابجا می‌کند.

۴- موارد زیر را با یک یا چند لایه از مدل OSI مطابقت دهید.

الف – مستقیماً با برنامه کاربردی کاربر، به مخابره دیتا می‌پردازد.

ب – تصحیح خطا و درخواست ارسال مجدد

ج- رابط مکانیکی، الکتریکی و عملیاتی

د – وظیفه جابجائی فریمها بین وسائل موجود در یک شبکه

۵- موارد زیر را با یک یا چند لایه از مدل OSI مطابقت دهید.

الف – سرویس‌های تبدیل فرمت و کد پیغام

ب - یک جلسه کاری را برقرار می‌کند، مدیریت می‌کند و خاتمه می‌دهد.

ج - انتقال قابل اطمینان دیتا را تضمین می‌کند.

د- روتین‌های ***اتصال به شبکه*** [1] و ***قطع از شبکه*** [2] را انجام می‌دهد.

هـ - مخابره دیتا را از فرمتهای مختلف نمایش دیتا، مستقل می‌کند.

فعالیتهای تحقیقاتی

۶- پروتوکل ***سیستم نام حوزه*** [3] یا DNS که در فصل هفدهم مورد بحث قرار خواهد گرفت، یک پروتوکل لایه کاربرد، در پشته پروتوکل TCP/IP است. اگر معادل این پروتوکل در مدل OSI وجود دارد، آن را پیدا کنید. سپس دو پروتوکل را با هم مقایسه کنید.

1- login
2- log-out
3- Domain Name System : DNS

www.Abiri.ir

۷- ***پروتوکل انتقال فایل***[1] یا FTP که در فصل نوزدهم مورد بحث قرار خواهدگرفت، یک پروتوکل لایه کاربرد، در پشته پروتوکل TCP/IP است. اگر معادل این پروتوکل در مدل OSI وجود دارد، آن را پیدا کنید. سپس دو پروتوکل را با هم مقایسه کنید.

۸- ***پروتوکل انتقال فایلهای کوچک***[2] یا TFTP که در فصل نهم مورد بحث قرار خواهد گرفت، یک پروتوکل لایه کاربرد در پشته پروتوکل TCP/IP است. اگر معادل این پروتوکل در مدل OSI وجود دارد، آن را پیدا کنید. سپس دو پروتوکل را با هم مقایسه کنید.

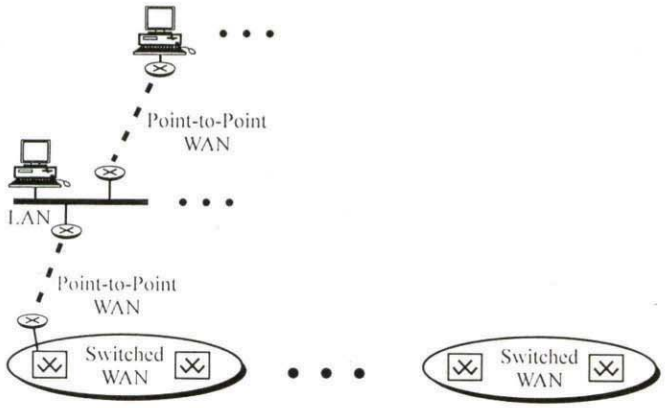
1- File Transfer Protocol : FTP
2- Trivial File Transfer Protocol : TFTP

www.Abiri.ir

فصل سوم

تکنولوژی‌های زیربنائی

می‌توان اینترنت جهانی را متشکل از یک سری شبکه‌های ***ستون فقرات***[1] دانست که نگهداری آنها به عهده ISP های ناحیه‌ای ، ملی و بین‌المللی می‌باشد. شبکه‌های ستون فقرات توسط وسائل متصل‌کننده، مانند روترها یا ایستگاههای سویچ، به یکدیگر متصل می‌شوند. کاربران نهایی یا مستقیماً به شبکه LAN مربوط به ISP محلی متصل می‌شوند و یا اتصال آنها به شبکه‌های LAN از طریق شبکه‌های نقطه به نقطه می‌باشد. مطابق شکل ۳-۱ می‌توان اینترنت جهانی را به عنوان مجموعه‌ای از ***شبکه‌های گسترده سویچ شده***[2] (شبکه‌های ستون فقرات)، شبکه‌های محلی، ***شبکه‌های گسترده نقطه به نقطه***[3] و وسایل متصل‌کننده مانند روترها در نظر گرفت.



شکل ۳-۱ : اینترنت جهانی

گرچه پشته پروتوکل TCP/IP غالباً به صورت پنج لایه‌ای نشان داده می‌شود، اما این پشته پروتوکل، فقط پروتوکلهای سه لایه بالاتر، یعنی لایه‌های شبکه، انتقال و کاربرد را تعریف می‌کند. بنابراین TCP/IP فرض را بر این گذاشته است که شبکه‌های گسترده، شبکه‌های محلی

1- backbone
2- switched WANs
3- point-to-points WANs

www.Abiri.ir

و وسایل متصل‌کننده آنها در لایه‌های مذکور، از پروتوکلهایی استفاده می‌کنند که پشته پروتوکل TCP/IP تعیین کرده است.

ما هم همین کار را انجام می‌دهیم و فرض را بر این می‌گذاریم که خواننده، هم‌اکنون با تکنولوژیهای زیربنائی مخابره دیتا، شبکه‌های محلی، شبکه‌های گسترده و وسایل متصل‌کننده آنها، آشناست. با این وجود به عنوان یک مرور سریع، نگاهی به این تکنولوژیها می‌اندازیم. توجه نمایید که درباره هیچ کدام از این موضوعات، به طور مفصل صحبت نخواهیم کرد. برای کسب اطلاعات بیشتر، کتاب اصول ارتباطات داده‌ها از همین مترجم را مطالعه فرمایید.

۳-۱ شبکه‌های محلی

یک ***شبکه محلی***[1] ، یک سیستم مخابره دیتاست که وسائل مستقل از هم را قادر می‌سازد تا در یک سطح جغرافیایی محدود، مانند یک ساختمان، یک سازمان و یا یک کمپ، با یکدیگر به مخابره مستقیم دیتا بپردازند. یک سازمان خیلی بزرگ ممکن است به چندین LAN متصل شده نیاز داشته باشد. متداولترین LANها ، اترنت و ***LAN بی‌سیم***[2] می‌باشند . در این بخش به اختصار، به بررسی این تکنولوژیها می‌پردازیم.

LANهای سیمی: اترنت

اترنت[3] شبکه محلی است که اکثراً از آن استفاده می‌شود . این پروتوکل در سال ۱۹۷۳ توسط آقای Bob Metcalfe در کمپانی Xerox با نرخ بیت 10 Mbps (10 میلیون بیت در ثانیه) روی توپولوژی باس ساخته شد. امروزه سرعت آن به 100 Mbps و 1 Gbps (یک گیگا بیت در ثانیه یا یک میلیارد بیت در ثانیه) رسیده است. اترنت به طور رسمی توسط استاندارد IEEE 802.3 تعریف شده است.

* اترنت با سرعت 10 Mbps

اترنت اولیه که به نام ***اترنت سنتی***[4] نیز موسوم است، نرخ بیتی برابر با 10 Mbps دارد. ابتدا درباره این نسخه از اترنت صحبت می‌کنیم .

1- Local Area Network : LAN
2- Wireless LAN
3- Ethernet
4- Traditional ethernet

www.Abiri.ir

**** متد دسترسی CSMA/CD**

استاندارد IEEE 802.3 ، ***متد دسترسی چندگانه به همراه تشخیص حامل با آشکارسازی تصادف***[1] را به عنوان متد دسترسی به محیط انتقال، پذیرفت. ایستگاههای متصل به شبکه اترنت، از طریق یک باس فیزیکی یا توپولوژی ستاره به هم متصل گردیده‌اند، اما توپولوژی منطقی، همواره یک باس است. منظور این است که محیط انتقال بین وسائل، مشترک است و فقط یک وسیله در هر زمان، می‌تواند از آن استفاده کند. همچنین به این معناست که فریم ارسالی توسط یک وسیله را، همه وسایل می‌توانند دریافت کنند (که به این عمل، ***ارسال همگانی***[2] می‌گویند). وسیله‌ای که مقصد واقعی فریم است، فریم را نگه می‌دارد و بقیه وسائل، آن را دور می‌اندازند. در چنین وضعیتی، چطور می‌توان مطمئن بود که دو وسیله، به طور همزمان فریمهایشان را روی محیط انتقال نمی‌فرستند؟ اگر چنین کاری کنند، فریمهایشان با هم ***تصادف***[3] خواهد کرد. CSMA/CD به کمک اصول زیر، این مشکل را حل می‌کند:

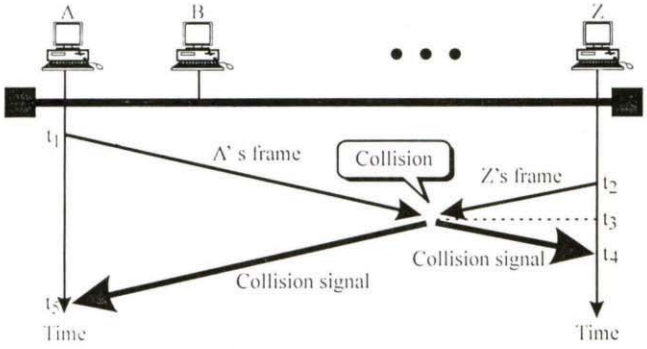
۱- هر وسیله، حق مشابهی در استفاده از محیط انتقال دارد (***دسترسی چندگانه***[4]).

۲- هر وسیله‌ای که می‌خواهد فریمی را ارسال کند، ابتدا به محیط انتقال گوش می‌کند تا بداند آیا سیگنالی روی محیط انتقال هست (***تشخیص حامل***[5]). اگر سیگنالی روی محیط انتقال حس نگردد، وسیله می‌تواند فریم خود را ارسال نماید.

۳- ممکن است این اتفاق بیفتد که دو ایستگاه به طور همزمان به محیط انتقال گوش دهند و سیگنالی روی آن نبینند و لذا هر دو با هم، فریمشان را روی محیط انتقال بگذارند . در این حالت، تصادف رخ می‌دهد. CSMA/CD از وسیله می‌خواهد که با شروع گذاشتن فریم خود روی خط انتقال، همچنان به خط انتقال گوش دهد تا اگر تصادف رخ داد، آن را آشکار کند (***آشکارسازی تصادف***[6]). وقتی که وسائل، تصادفی را آشکار کردند، همه آنها یک ***نویز قوی***[7] روی خط می‌گذارند تا فریمی که روی خط جابجا می‌شود را از بین ببرد و سپس هر وسیله، یک مدت زمان ***اتفاقی***[8] صبر می‌کند و سپس مجدداً فریم خود را روی محیط انتقال می‌گذارد. زمانهای اتفاقی، مانع از بروز مجدد تصادف می‌شوند. شکل ۳-۲ نحوه عملکرد CSMA/CD را نشان می‌دهد.

1- Carrier Sense Multiple Access with Collision Detection : CSMA/CD
2- Broadcasting
3- Collision
4- Multiple Access
5- Carrier Sense
6- Collision Detection
7- Jam signal
8- Random

www.Abiri.ir



شکل ۳-۲ : متد دسترسی CSMA/CD

سه پارامتر در متد دسترسی CSMA/CD نقش دارند: حداقل طول فریم، نرخ انتقال بیت و ***حوزه تصادف***[1] . زمانی که وسیله باید صبر کند تا مطمئن شود که دیتای دیگری روی محیط انتقال در حال جابجایی نیست، برابر است با حداقل طول فریم، تقسیم بر نرخ انتقال بیت و نتیجه حاصل، برابر با مدت زمان لازم برای ارسال فریمی با حداقل طول، می‌باشد. این زمان متناسب است با مدت زمان لازم برای آنکه اولین بیت فریم، ماکزیمم مسافت ممکن روی شبکه، که همان حوزه تصادف می‌باشد، را طی کند. به عبارت دیگر می‌توانیم بگوییم که حداقل طول فریم تقسیم بر نرخ انتقال بیت، متناسب است با حوزه تصادف، تقسیم بر سرعت انتشار سیگنال.

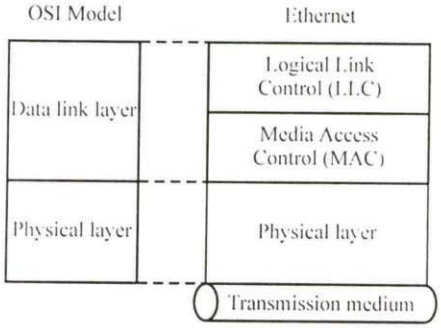
حداقل سایز یک فریم اترنت، 72 بایت یا 576 بیت است. ارسال 576 بیت با سرعت 10 Mbps ، به مدت 57.6 میکروثانیه به طول خواهد انجامید. قبل از آنکه آخرین بیت فرستاده شود، اولین بیت باید به انتهای حوزه تصادف رسیده باشد و اگر تصادفی رخ دهد، باید توسط فرستنده حس گردد. به عبارت دیگر، تصادف باید در خلال زمان 57.6 میکروثانیه آشکار گردد. این زمان برای آنکه سیگنال، مسافت رفت و برگشت 5000 متر را با سرعت انتشار نزدیک سرعت نور داخل یک محیط انتقال نوعی مانند کابل زوج سیم به هم تابیده طی کند، کافی است. بنابراین حوزه تصادف که نصف این مقدار است، برابر با 2500 متر می‌باشد.

1- Collision domain

www.Abiri.ir

❊ لایه‌ها

شکل ۳-۳ لایه‌های اترنت 10 Mbps را نشان می‌دهد. لایه دیتالینک دو زیر لایه دارد: زیر ***لایه کنترل منطقی لینک***[1] و زیر لایه ***کنترل دسترسی به محیط انتقال***[2].



شکل ۳-۳ : لایه‌های اترنت

در لایه دیتالینک، زیر لایه LLC مسئول کنترل خطا و کنترل فلو می‌باشد . زیر لایه MAC مسئول اجرای متد دسترسی CSMA/CD است. همچنین زیر لایه MAC ، دیتای دریافتی از زیر لایه LLC را به صورت فریم در می‌آورد و فریمها را به لایه فیزیکی می‌دهد . لایه فیزیکی، فریم دریافتی را به صورت سیگنال الکتریکی در می‌آورد و از طریق محیط انتقال، آن را برای ایستگاه بعدی ارسال می‌کند . همچنین لایه فیزیکی، وقوع تصادفها را آشکار می‌کند و آنها را به اطلاع لایه دیتالینک می‌رساند.

❊ فریم اترنت

IEEE 802.3 فریم اترنت را با هفت فیلد، معرفی کرده است :

Preamble : مقدمه

SFD (Start Field Delimiter) : شاخص شروع فیلد

DA (Destination Address) : آدرس فیزیکی مقصد

SA (Source Address) : آدرس فیزیکی مبدأ

Length / Type : نوع / طول

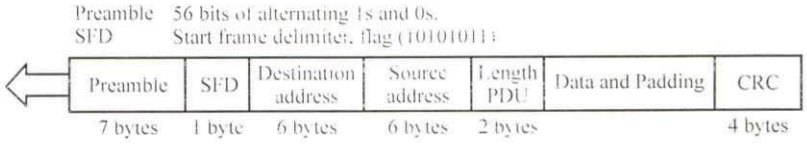
فریم 802.2 که همان LLC است.

CRC (Cyclic Redundancy Check) : یک متد آشکارسازی خطای بیت است.

1- Logical Link Control : LLC
2- Media Access Control : MAC

www.Abiri.ir

اترنت هیچ مکانیزمی برای ارسال تأییدیه بر فریمهای دریافتی، ندارد و به همین دلیل است که آن را به یک پروتوکل غیر قابل اطمینان تبدیل کرده است. ارسال تأییدیه‌ها باید در لایه‌های بالاتر صورت گیرد. فرمت فریم MAC اترنت، در شکل ۳-۴ نشان داده شده است.



شکل ۳-۴ : فریم اترنت

Preamble : این فیلد در فریم اترنت، حاوی هفت بایت (۵۶ بیت) است که در آن، بیتهای 0 و 1 به صورت یک در میان (.....10101010) به دنبال هم، می‌آیند و آمدن فریم جدید را به اطلاع گیرنده، می‌رسانند و به این ترتیب، گیرنده می‌تواند کلاک خود را با بیتهای فریم جدید، سنکرون نماید. در واقع Preamble ، در لایه فیزیکی اضافه می‌شود و جزئی از فریم لایه دیتالینک نیست.

SFD: این فیلد به طول یک بایت و برابر با 10101011 می‌باشد و شروع فریم اترنت را نشان می‌دهد. SFD آخرین فرصت سنکرونیزاسیون را به کلاک گیرنده می‌دهد. آخرین دو بیت این فیلد، به صورت 11 است و نشان می‌دهد که آنچه بعد از این دو بیت خواهد آمد، آدرس فیزیکی مقصد است.

DA : یک فیلد شش بایتی است و حاوی آدرس فیزیکی مقصد است.

SA : یک فیلد شش بایتی است و حاوی آدرس فیزیکی مبدأ است.

Length / Type : این فیلد، به یکی از دو معنای زیر است: اگر مقدار این فیلد کوچکتر از 1518 باشد، به معنای Length است و طول دیتایی که به دنبال آن می‌آید را بر حسب بایت، نشان می‌دهد. اگر مقدار این فیلد بزرگتر از 1536 باشد، به معنای Type است و پروتوکل لایه بالاتر (لایه شبکه) که از سرویس لایه دیتالینک استفاده می‌کند را مشخص می‌کند.

Data : دیتایی که از لایه بالاتر می‌آید، در فیلد Data قرار می‌گیرد و به عبارت دیگر، توسط لایه دیتالینک، کپسول‌بندی می‌شود . حداقل طول آن، 46 بایت و حداکثر طول آن، برابر با 1500 بایت است .

CRC : آخرین فیلد در فریم اترنت، حاوی اطلاعات مربوط به آشکارسازی خطای بیت است که در این حالت از CRC-32 استفاده شده است.

www.Abiri.ir

❊❊ آدرس‌دهی

هر وسیله‌ای مانند یک PC یا پرینتر متصل به شبکه اترنت، ***کارت شبکه***[1] خودش را دارد. NIC داخل وسیله قرار می‌گیرد و یک آدرس فیزیکی به آن وسیله می‌دهد. آدرس فیزیکی اترنت، ۶ بایتی (۴۸ بیتی) است که معمولاً با نماد هگزا دسیمال نوشته می‌شود و بین بایتها، یک ***دو نقطه***[2] گذاشته می‌شود. به مثال زیر توجه فرمایید :

07:01:02:01:2C:4B

آدرس فیزیکی در اترنت به صورت بایت به بایت، از چپ به راست ارسال می‌شود. با این وجود برای هر بایت، ابتدا کم ارزش‌ترین بیت و در نهایت، با ارزش‌ترین بیت ارسال می‌گردد.

در اترنت، سه نوع آدرس داریم: ***تک‌گیرنده***[3] ، ***چندگیرنده***[4] و ***همگانی***[5]. در یک آدرس تک‌گیرنده، کم ارزش‌ترین بیت اولین بایت (بایت سمت چپ آدرس) برابر با 0 است. در آدرس چندگیرنده، کم ارزش‌ترین بیت اولین بایت، برابر با 1 است. آدرس همگانی، از 48 بیت 1 ساخته شده است. آدرس SA همواره تک‌گیرنده است. آدرس DA می‌تواند از هر سه نوع باشد.

❊❊ پیاده‌سازی

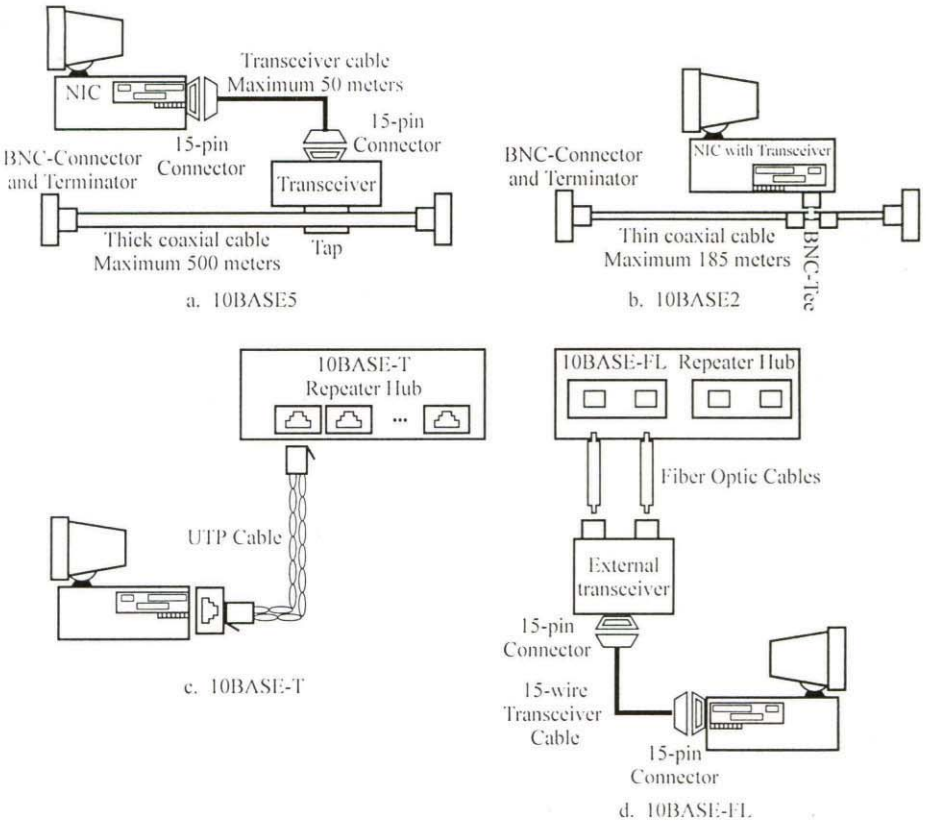
IEEE ، چهار نوع پیاده‌سازی را برای اترنت ارائه کرده است . شکل۳-۵ نمایش ساده‌ای از این چهار نوع پیاده‌سازی را نشان می‌دهد. ***فرستنده/گیرنده***[6] که می‌تواند ***خارجی***[7] یا ***داخلی***[8] باشد، مسئول سیگنال‌سازی، آشکارسازی تصادف و ارسال/دریافت سیگنالهاست.

10BASE5 : به آن، ***اترنت ضخیم***[9] نیز گفته می‌شود. این پیاده‌سازی از توپولوژی باس استفاده می‌کند و محیط انتقال آن، کابل کواکسیال ضخیم است.

10BASE2 : به آن، ***اترنت نازک***[10] نیز گفته می‌شود. این پیاده‌سازی از توپولوژی باس استفاده می‌کند و محیط انتقال آن، کابل کواکسیال نازک است.

1-Network Interface Card : NIC
2- Colon
3- Unicast
4- Multicast
5- Broadcast
6- Transceiver
7- External
8- Internal
9- Thick ethernet
10- Thin ethernet - cheapernet

www.Abiri.ir



شکل ۳-۵ : پیاده‌سازی‌های اترنت

10BASE-T : به آن، اترنت با ***زوج سیم به هم تابیده***[1] نیز گفته می‌شود. این پیاده‌سازی از توپولوژی ستاره استفاده می‌کند و در آن هر وسیله، با دو جفت زوج سیم به هم تابیده، به هاب وصل می‌شود.

10BASE-FL : به آن، ***اترنت فیبر نوری***[2] نیز گفته می‌شود. این پیاده‌سازی، از توپولوژی ستاره استفاده می‌کند و در آن هر وسیله، با یک جفت فیبر نوری، به هاب وصل می‌شود.

* اترنت سریع

نیاز به نرخ بیت بالاتر، به ایجاد ***اترنت سریع***[3] با سرعت 100 Mbps منجر شد. در زیر لایه MAC ، اترنت سریع از همان اصول اترنت (CSMA/CD) تبعیت می‌کند و تنها این تفاوت را دارد که نرخ انتقال، از 10Mbps به 100Mbps ، افزایش یافته است. وقتی که نرخ بیت

1- Twisted-pair ethernet
2- Fiber link ethernet
3- Fast Ethernet

www.Abiri.ir

افزایش می‌یابد، زمان کمتری طول می‌کشد تا یک فریم، ایستگاه فرستنده را ترک گوید. معنی دیگر آن این است که ایستگاه فرستنده، زمان کمتری برای گوش دادن به خط و پی بردن به وقوع تصادفی روی خط، دارد. از آنجا که سرعت انتشار سیگنال در محیط انتقال، مقدار ثابتی است، این بدان معناست که حوزه تصادف باید کوتاهتر گردد.

برای آنکه CSMA/CD کار کند، دو کار می‌توانیم انجام دهیم: یا حداقل طول فریم را افزایش دهیم و یا اینکه حوزه تصادف را کوتاهتر کنیم (سرعت نور را نمی‌توان تغییر داد). افزایش حداقل طول فریم، مستلزم اضافه کردن ***اطلاعات بالاسری***[1] بیشتری به فریم است. اگر دیتایی که قرار است ارسال کنیم، به اندازه کافی بلند نباشد، باید بایتهای بیشتری را به صورت اطلاعات بالاسری، به فریم اضافه کنیم و این عمل، کارآیی اترنت را کاهش می‌دهد. اترنت سریع، راه دوم را برگزیده است و لذا حوزه تصادف را 10 مرتبه کاهش داده است (از 2500 متر به 250 متر). در توپولوژی ستاره، 250 متر قابل قبول است. اترنت سریع در لایه فیزیکی، از روشهای مختلف سیگنال‌سازی و انواع محیطهای انتقال استفاده می‌کند تا بتواند به نرخ بیت 100 Mbps برسد.

❊❊ پیاده‌سازی اترنت سریع

اترنت سریع یا به صورت 100BASE-X پیاده‌سازی می‌شود که خود بر دو نوع است: 100BASE-TX (دو زوج سیم به هم تابیده) و 100BASE-FX (دو عدد فیبر نوری) و یا به صورت 100BASE-T4 پیاده‌سازی می‌گردد که در آن، از چهار زوج سیم به هم تابیده استفاده می‌شود. بنابراین برای اترنت سریع، سه نوع پیاده‌سازی 100BASE-TX ، 100BASE-FX و 100BASE-T4 را داریم (شکل ۳-۶ را ببینید).

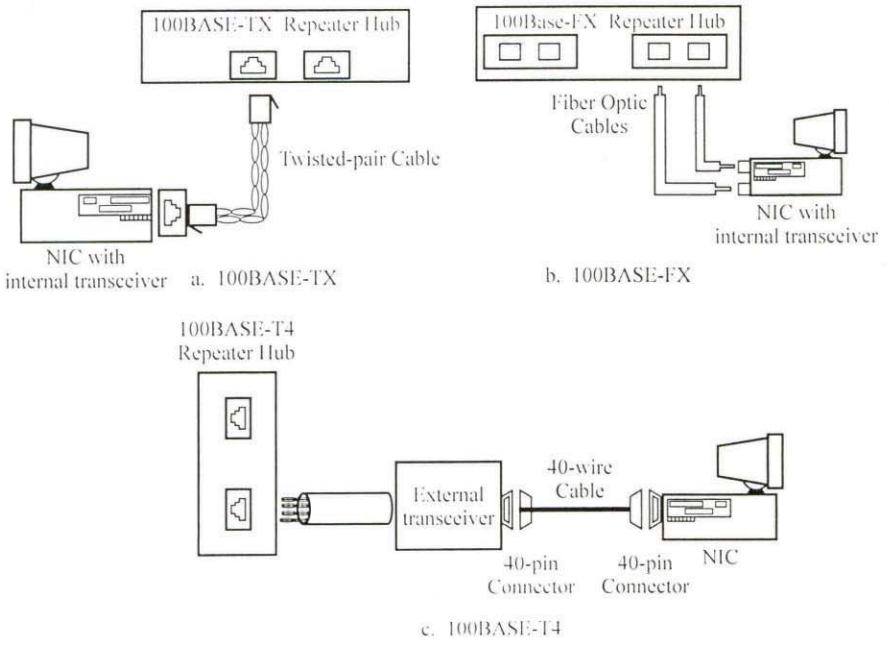
❊ گیگابیت اترنت

نیاز به نرخ بیت بالاتر از 100 Mbps ، به تولید گیگابیت اترنت با سرعت 1000 Mbps و یا 1 Gbps منجر شد. زیر لایه MAC برای رسیدن به این سرعت، دو راه دارد: یا از CSMA/CD استفاده کند و یا آن را حذف نماید. در مورد راه اول که نگه داشتن CSMA/CD است، مجدداً یا باید حداقل طول فریم را افزایش دهد و یا حوزه تصادف را کوتاهتر نماید. از آنجا که 10 مرتبه کاهش مجدد حوزه تصادف و رسیدن به 25 متر، مقدار غیر قابل قبولی است، حداقل طول فریم به روش بسیار جالبی افزایش یافته است. در راه حل دوم که عدم استفاده از CSMA/CD است، هر وسیله توسط دو مسیر جداگانه به هاب مرکزی متصل شده

1- overhead

www.Abiri.ir

است و به آن ***اترنت کاملاً دو طرفه***[1] گفته می‌شود و به دلیل آنکه هیچ تصادفی رخ نمی‌دهد، نیازی به CSMA/CD ندارد. در لایه فیزیکی تغییرات عمده‌ای داده شده است تا ارسال اطلاعات" با سرعت 1 Gbps ، امکان‌پذیر شود.



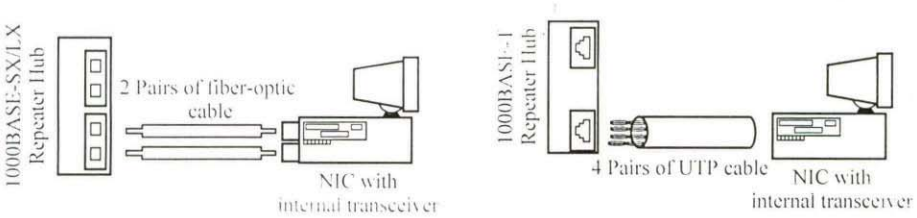
شکل ۳-۶ : پیاده‌سازی اترنت سریع

** پیاده‌سازی گیگابیت اترنت

گیگابیت اترنت یا به صورت 1000BASE-X پیاده‌سازی می‌شود که خود بر دو نوع است: 1000BASE-SX (دو زوج فیبر نوری که سیگنالهای لیزر با ***طول موج کوتاه***[2] را منتقل می‌کنند) و 1000BASE-LX (دو زوج فیبر نوری که سیگنالهای لیزر با ***طول موج بلند***[3] را منتقل می‌کنند) و یا به صورت 1000BASE-T پیاده‌سازی می‌شود که از چهار زوج سیم به هم تابیده استفاده می‌کند. این پیاده‌سازی‌ها در شکل ۳-۷ نشان داده شده‌اند.

1- Full-duplex ethernet
2- Short wave
3- Long wave

www.Abiri.ir



a. 1000BASE-SX LX b. 1000BASE-T

شکل ۳-۷ : پیاده‌سازی گیگابیت اترنت

LAN بی‌سیم : IEEE 802.11

در این بخش، درباره یکی از انواع تکنولوژیهای موجود برای ***LANهای بی‌سیم***[1] صحبت می‌کنیم که IEEE 802.11 می‌باشد. گاهی اوقات به LAN بی‌سیم، اترنت بی‌سیم نیز گفته می‌شود. ما به این تکنولوژی، به چشم ***لینکهای بی‌سیم***[2] نگاه می‌کنیم یعنی لینکهایی که به صورت بی‌سیم، می‌توانند ما را به شبکه متصل کنند. اینکه چگونه اتصال برقرار می‌شود و مخابره پایانه به پایانه صورت می‌گیرد، موضوع فصلهای بعدی است.

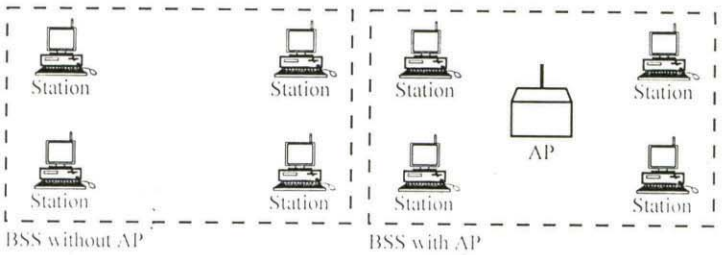
IEEE مشخصه یک LAN بی‌سیم استاندارد به نام IEEE 802.11 را ارائه کرده است که لایه‌های فیزیکی و دیتالینک را در بر می‌گیرد. اما قبل از بحث پیرامون این لایه‌ها، کلیاتی درباره معماری این پروتوکل، خواهیم گفت.

* معماری

در این استاندارد، دو نوع سرویس تعریف می‌شوند: ***مجموعه سرویسهای پایه***[3] و ***مجموعه سرویسهای توسعه یافته***[4]. BSS ، واحد سازنده یک LAN بی‌سیم است. یک BSS شامل ***ایستگاههای ثابت***[5] و ***متحرک***[6] بی‌سیم می‌باشد و ممکن است یک ایستگاه پایه مرکزی، به نام ***نقطه دسترسی***[7] نیز داشته باشد. شکل ۳-۸ ، BSS با AP و BSS بدون AP را نشان می‌دهد.

1- Wireless LAN
2- Wireless Link
3- Basic Service Set : BSS
4- Extended Service Set : ESS
5- stationary
6- mobile
7- Access Point : AP

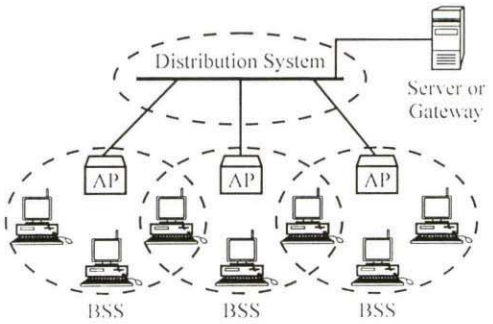
www.Abiri.ir



شکل ۳-۸ : BSS ها

BSS بدون AP ، یک شبکه ***مجزا و منفرد***[1] است و نمی‌تواند دیتایی برای سایر BSSها ارسال نماید. این همان چیزی است که ***معماری مختص همین مورد***[2] نامیده می‌شود. در این معماری، ایستگاهها می‌توانند بدون نیاز به یک AP ، یک شبکه را بسازند. ایستگاهها می‌توانند محل یکدیگر را پیدا کنند و قبول داشته باشند که همگی، متعلق به یک BSS هستند.

یک ESS از دو یا چند BSS با AP ساخته شده است. در این حالت، APهای BSSها به کمک یک ***سیستم توزیع***[3] که غالباً یک LAN سیمی است، به هم متصل می‌شوند. IEEE 802.11 هیچ محدودیتی بر سیستم توزیع اعمال نمی‌کند و لذا سیستم توزیع می‌تواند هر LAN معرفی شده توسط IEEE ، مانند اترنت، باشد. توجه نمایید که ESS از دو نوع ایستگاه استفاده می‌کند: متحرک و ثابت. ایستگاههای متحرک، همان ایستگاههای معمولی داخل یک BSS هستند. ایستگاههای ثابت APها هستند که بخشی از یکLAN سیمی (سیستم توزیع) محسوب می‌شوند. شکل ۳-۹ یک ESS را نشان می‌دهد.



شکل ۳-۹ : ESS

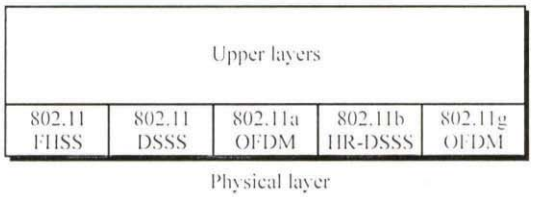
1- Stand-alone
2- Ad hoc architecture
3- Distribution system

www.Abiri.ir

وقتی که BSS ها از طریق یک سیستم توزیع به هم متصل می‌شوند، یک ***شبکه زیرساخت***[1] شکل می‌گیرد. در این شبکه، ایستگاههایی که در محدوده دسترسی همدیگر واقع شده‌اند، می‌توانند مستقیماً بدون استفاده از AP ، به یکدیگر دیتا مخابره کنند. با این وجود، معمولاً مخابره بین دو ایستگاه متعلق به دو BSS متفاوت، از طریق دو AP موجود در آن دو BSS صورت می‌پذیرد. اگر هر BSS را به عنوان یک ***سلول***[2] و هر AP را به عنوان یک ***ایستگاه پایه***[3] در نظر بگیریم، می‌توان این شبکه را مشابه با ***شبکه مبایل***[4] دانست. توجه نمایید که یک ایستگاه مبایل در هر لحظه، می‌تواند متعلق به بیش از یک BSS باشد.

* لایه فیزیکی

IEEE 802.11 مشخصه‌های متعددی را برای لایه فیزیکی که وظیفه تبدیل بیتها به سیگنال را دارد، تعریف کرده است. یک مشخصه، از فرکانسهای نور مادون قرمز استفاده می‌کند که در اینجا مورد بحث قرار نمی‌گیرد. پنج مشخصه دیگر در رنج فرکانسهای رادیویی ارائه شده‌اند و شکل ۳-۱۰ آنها را نشان می‌دهد.



شکل ۳-۱۰ : لایه فیزیکی

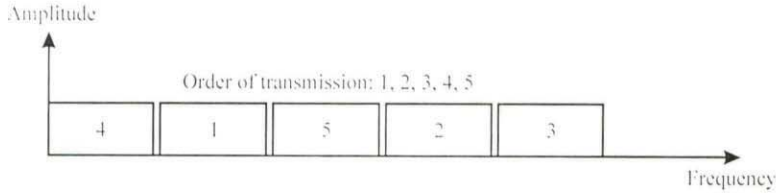
** IEEE 802.11 FHSS

این استاندارد از متد ***طیف گستردهٔ جهش فرکانسی***[5] برای تولید سیگنال در باند 2.40-2.48 GHZ استفاده می‌کند و نرخ بیت آن 1-2 Mbps است. FHSS متدی است که در آن، فرستنده برای یک طول زمانی کوتاه، یک فرکانس کاریر را ارسال می‌کند، سپس به فرکانس کاریر دیگری جهش می‌نماید و مجدداً به همان طول زمانی قبلی، فرکانس کاریر جدید را ارسال می‌نماید. پس از آن دوباره به فرکانس کاریر دیگری جهش می‌نماید و به همان طول

1- Infrastructure network
2- Cell
3- Base station
4- Cellular network
5- Frequency Hopping Spread Spectrum : FHSS

www.Abiri.ir

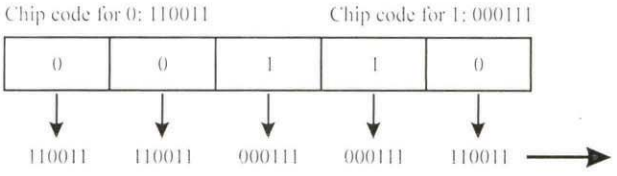
زمانی قبلی، از این فرکانس جدید استفاده می‌نماید. پس از آنکه N مرتبه، به فرکانسهای مختلف جهش کرد، مجدداً سیکل جهشها را به صورت پریودیک، تکرار می‌نماید (شکل ۳-۱۱ را ببینید). اگر پهنای باند سیگنال اصلی B هرتز باشد، پهنای باند سیگنال حاصل از جهشهای فرکانسی برابر با NxB هرتز خواهد بود.



شکل ۳-۱۱ : FHSS

IEEE 802.11 DSSS ❊❊

این استاندارد از متد ***طیف گستردهٔ دنباله مستقیم***[1] برای تولید سیگنال در باند 2.40-2.48 GHZ استفاده می‌کند و نرخ بیت آن 1 Mbps یا 2 Mbps است. در DSSS ، فرستنده به جای ارسال هر بیت، ***قطعه کد***[2] متناظر با آن بیت را ارسال می‌کند . با این وجود به منظور عدم استفاده از بافر، زمان لازم برای ارسال یک قطعه کد، باید برابر با زمان لازم برای ارسال بیت اصلی متناظر با آن قطعه کد باشد. اگر یک قطعه کد حاوی N بیت باشد، در این صورت نرخ بیت ارسال قطعه کدها، N برابر نرخ بیت دنباله اولیه بیتها، خواهد بود. شکل ۳-۱۲ مثالی از DSSS را نشان می‌دهد.



شکل ۳-۱۲: DSSS

IEEE 802.11a OFDM ❊❊

این استاندارد از متد ***مالتی‌پلکسینگ متعامد تقسیم فرکانسی***[3] و یا به اختصار FDM متعامد، با نرخ بیت 18 Mbps و 54 Mbps استفاده می‌کند.

1- Direct Sequence Spread Spectrum : DSSS
2- chip code
3- Orthogonal Frequency Division Multiplexing : OFDM

www.Abiri.ir

IEEE 802.11b HR-DSSS ❊❊

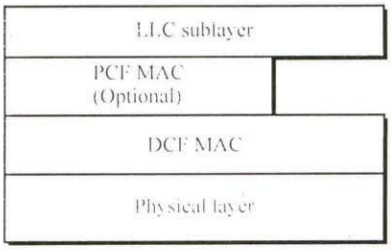
این استاندارد از متد ***DSSS پر سرعت***[1] برای تولید سیگنال در باند 2.40-2.48 GHZ با نرخ بیت 1 Mbps ، 2 Mbps ، 5.5 Mbps و 11 Mbps استفاده می‌کند. این متد با متد DSSS سازگار است و از یک متد کدینگ به نام ***کلید زنی کد مکمل***[2] استفاده می‌کند.

IEEE 802.11g OFDM ❊❊

این استاندارد از نسخه نسبتاً جدیدی از OFDM برای تولید سیگنال در باند 2.40-2.48 GHZ با نرخ بیت 54 Mbps استفاده می‌کند و با استفاده از یک تکنیک مدولاسیون پیچیده، به نرخ بیت بالایی دست می‌یابد.

❊ زیر لایه MAC

IEEE 802.11 دو زیر لایه MAC را تعریف می‌کند: ***تابع هماهنگی توزیع شده***[3] و ***تابع هماهنگی نقطه‌ای***[4]. شکل ۳-۱۳ را ببینید.



شکل ۳-۱۳: زیرلایه‌های MAC در استاندارد IEEE 802.11

PCF یک متد دسترسی پیچیده و دلخواه است که باید در یک شبکه زیرساخت پیاده‌سازی شود (نمی‌توان آن را در یک شبکه ad hoc پیاده‌سازی کرد) و لذا در اینجا، به بحث پیرامون آن نمی‌پردازیم .

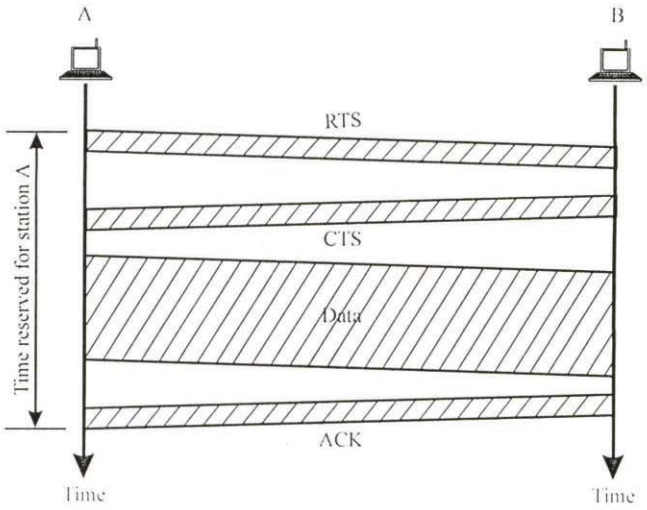
DCF از متد ***دسترسی چندگانه به همراه تشخیص حامل و جلوگیری از وقوع تصادف***[5] استفاده می‌کند . اولین دو اصل CSMA/CD در مورد CSMA/CA نیز قابل استفاده می‌باشند. هر ایستگاه حق دسترسی یکسانی به محیط انتقال دارد (***دسترسی چندگانه***[6]) و

1- High-Rate DSSS
2- Complementary Code Keying : CCK
3- Distributed Coordination Function : DCF
4- Point Coordination Function : PCF
5- Carrier Sense Multiple Access with Collision Avoidance : CSMA/CA
6- Multiple Access : MA

www.Abiri.ir

هر ایستگاه، قبل از آنکه فریمی ارسال کند، به محیط انتقال گوش می‌دهد (***تشخیص حامل***[1]). با این وجود، بدلایل مختلف نمی‌توان وقوع تصادف را آشکار کرد و عمده‌ترین دلیل آن، مشکلی به نام ***مشکل ترمینال پنهان***[2] است. برای درک مشکل ترمینال پنهان، فرض کنید که سه ایستگاه داریم (۱ و ۲ و ۳). ایستگاه ۱ برای ایستگاه ۲ ، فریمی را ارسال می‌کند و در همان زمان، ایستگاه ۳ نیز فریمی را برای ایستگاه ۲ ارسال می‌نماید. ممکن است ایستگاههای ۱ و ۳ به دلیل وجود مانعی ما بین آنها (مثلاً یک دیوار یا یک کوه) نتوانند سیگنالهای یکدیگر را دریافت کنند اما وقتی که سیگنالهای آنها به ایستگاه ۲ می‌رسد، در محل ایستگاه ۲ ، تصادف رخ دهد و ایستگاههای ۱و ۳ از وقوع این تصادف، بی‌خبر بمانند. آنها فکر می‌کنند که فریمهایشان صحیح و سالم به ایستگاه ۲ رسیده است.

برای ممانعت از وقوع چنین حالتی، باید از بروز تصادف جلوگیری کرد (***جلوگیری از وقوع تصادف***[3]). هر ایستگاهی اعلام می‌کند که چه مدت زمانی به محیط انتقال نیاز دارد و سایر ایستگاهها از ارسال فریمهایشان در خلال این مدت زمان، امتناع می‌کنند. شکل ۳-۱۴ این روش را نشان می‌دهد.



شکل ۳-۱۴: CSMA/CA

روش کار به صورت زیر است:

1- Carrier Sense
2- hidden terminal problem
3- Collision Avoidance

www.Abiri.ir

۱- ایستگاه فرستنده پس از گوش دادن به محیط انتقال و بی کار یافتن آن، یک فریم کوچک به نام ***درخواست ارسال***[1] را می‌فرستد.

۲-گیرنده، یک فریم کوچک به نام ***آمادگی برای ارسال***[2] را به صورت همگانی برای کلیه ایستگاهها می‌فرستد و به این ترتیب به همه ایستگاهها اعلام می‌نماید که آماده دریافت فریم دیتای فرستنده‌ای است که فریم RTS را ارسال کرده است.

۳-فرستنده، فریم دیتای خود را ارسال می‌کند.

۴-گیرنده، تأییدیه‌ای بر دریافت فریم دیتا، می‌فرستد.

**** فریم MAC**

مطابق شکل ۳-۱۵ ، فریم زیر لایه MAC حاوی ۹ فیلد است.

2 bytes	2 bytes	6 bytes	6 bytes	6 bytes	2 bytes	6 bytes	0 to 2312 bytes	4 bytes
FC	D	Address 1	Address 2	Address 3	SC	Address 4	Frame body	FCS

شکل ۳-۱۵: فریم زیرلایه MAC

FC (Frame Control) : فریم کنترل. فیلد FC دو بایتی است و نوع فریم و برخی اطلاعات کنترلی دیگر را تعریف می‌کند .

D : این فیلد در همه انواع فریمهای کنترلی، به استثناء یک مورد، طول زمانی ارسال را مشخص می‌کند و در آن مورد استثناء، ID یا شناسه فریم را مشخص می‌نماید.

Addresses : آدرس‌ها. چهار فیلد آدرس داریم که هر یک به طول ۶ بایت می‌باشد. معنی هر فیلد آدرس بستگی به مقدار زیر فیلدهای To DS و From DS دارد که در قسمت مربوطه، راجع به آنها صحبت خواهیم کرد.

SC (Sequence Control) : کنترل دنباله. این فیلد، شماره ترتیب فریم را مشخص می‌کند و این شماره ترتیب، در کنترل فلو به کار می‌رود.

Frame body : بدنه فریم. این فیلد که می‌تواند به طول صفر بایت تا 2312 بایت باشد، حاوی اطلاعاتی است که بستگی به مقدار type (نوع) و subtypeای (زیر نوع) دارد که توسط فیلد FC تعیین می‌شوند.

1- Request to send :RTS
2- Clear To Send : CTS

www.Abiri.ir

FCS (Frame Check Sequence) : دنباله چک کننده فریم. فیلد FCS به طول 4 بایت است و حاوی دنباله‌ای برای آشکارسازی خطا با استفاده از CRC-32 است.

* مکانیزم آدرس‌دهی

مکانیزم آدرس‌دهی IEEE 802.11 پیچیده است. پیچیدگی ریشه در این واقعیت دارد که ممکن است ایستگاههای میانی (همان APها) وجود داشته باشند. کلاً چهار حالت در نظر گرفته شده است که توسط دو بیت To DS و From DS موجود در فیلد FC تعیین می‌شوند. هر بیت (یا هر **پرچم**[1]) می‌تواند 0 یا 1 باشد و لذا دو بیت می‌توانند چهار حالت داشته باشند. مطابق جدول ۳-۱ ، نحوه تفسیر چهار فیلد آدرس یعنی Address1 ، Address2 ، Address3 و Address4 موجود در فریم MAC، به مقدار این دو بیت بستگی دارد.

To DS	*From DS*	*Address 1*	*Address 2*	*Address 3*	*Address 4*
0	0	Destination station	Source station	BSS ID	N/A
0	1	Destination station	Sending AP	Source station	N/A
1	0	Receiving AP	Source station	Destination station	N/A
1	1	Receiving AP	Sending AP	Destination station	Source station

جدول ۳-۱ : آدرسها در IEEE 802.11

توجه نمایید که Address1 همواره آدرس وسیله بعدی است و Address2 ، همواره آدرس وسیله قبلی است. Address3 همواره آدرس مقصد نهایی است مشروط برآنکه Address1 ، حاوی آدرس مقصد نهایی نباشد. Address4 ، آدرس مبدأ اولیه است مشروط بر آنکه Address4 ، مشابه Address2 نباشد.

۳-۲ شبکه‌های گسترده نقطه به نقطه

نوع دوم شبکه‌ها که در اینترنت با آن برخورد خواهیم کرد، ***شبکه‌های گسترده نقطه به نقطه***[2] می‌باشند. چنین شبکه‌هایی، دو وسیله دوردست را با استفاده از لینکی که از یک

1- flag
2- Point-to-Point WAN

www.Abiri.ir

شبکه عمومی مانند یک شبکه تلفنی گرفته شده است، به هم متصل می‌نمایند. در اینجا، لایه‌های فیزیکی و دیتالینک این نوع شبکه‌ها را مورد بحث و بررسی قرار می‌دهیم.

لایه فیزیکی

اتصال نقطه به نقطه دو وسیله در سطح لایه فیزیکی را می‌توان به کمک یکی از سرویسهای موجود امروزی مانند مدم، یک خط DSL، مدم کابلی، یک خط T و یا SONET برقرار کرد .

* مدم‌های 56K

مطابق شکل۳-۱۶ ، هنوز برای ***دیتادهی***[1] به اینترنت و ***دیتاگیری***[2] از اینترنت، مدم‌های معمولی مورد استفاده قرار می‌گیرند.

در دیتادهی، از سیگنال آنالوگ در مرکز سویچینگ، نمونه‌برداری می‌شود و سپس نمونه‌ها، کوانتیزه می‌شوند. به دلیل جمع شدن نویز کوانتیزیشن با سیگنال اصلی، نرخ بیت در دیتادهی به 33.6Kbps محدود می‌شود اما در دیتاگیری، نمونه‌برداری انجام نمی‌شود و کوانتیزیشن هم نداریم، پس نویز کوانتیزیشن با سیگنال جمع نمی‌شود. به همین دلیل نرخ بیت در عمل دیتاگیری، بر اساس قانون شانون محدود نمی‌شود. لذا ماکزیمم نرخ بیت در دیتادهی برابر با 33.6 Kbps و در دیتاگیری برابر با 56 Kbps است.

ممکن است این سؤال پیش آید که چرا 56 Kbps ؟ شرکتهای تلفنی در هر ثانیه از صدا، 8000 نمونه برمی‌دارند و هر نمونه را با 8 بیت نمایش می‌دهند. در هر نمونه، یک بیت برای کارهای کنترلی به کار می‌رود و لذا در هر نمونه، 7 بیت برای دیتا باقی می‌ماند. بنابراین نرخ دیتا برابر خواهد بود با 8000x7 = 56000 bps = 56 Kbps

مدمهای استاندارد V.90 و V.92 در اتصال کامپیوتر به اینترنت، دیتاگیری را با سرعت 56K انجام می‌دهند.

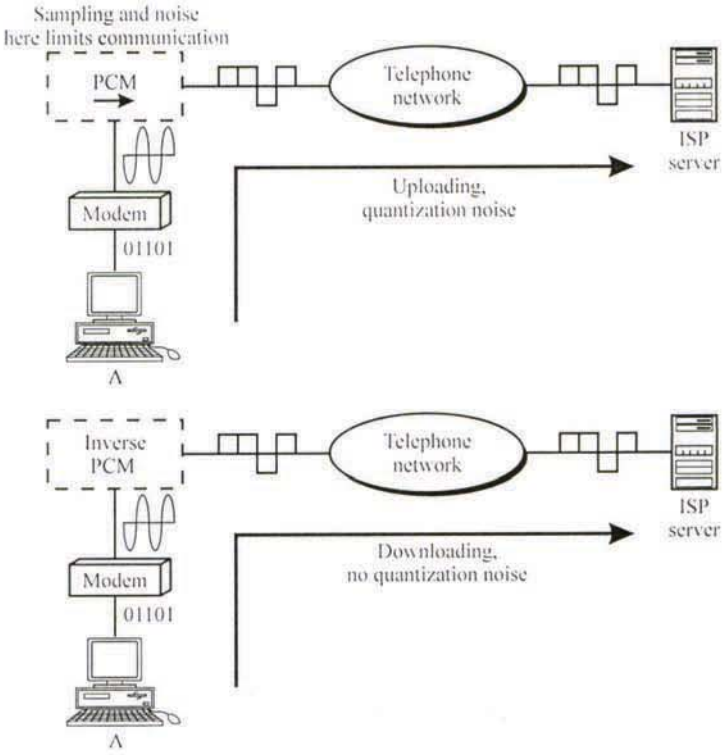
* تکنولوژی DSL

بعد از آنکه مدمهای معمولی به پیک نرخ دیتای خود رسیدند، شرکتهای تلفنی تکنولوژی دیگری به نام DSL را ابداع نمودند تا مشترکین بتوانند با سرعت بالاتری به اینترنت متصل شوند. تکنولوژی ***خط دیجیتالی مشترک***[3] یکی از بهترین تکنولوژیهای موجود برای

1- Upload
2- Download
3- Digital Subscriber Line : DSL

www.Abiri.ir

پشتیبانی از مخابره دیتا با سرعت بالا، روی خطوط تلفنی موجود است. تکنولوژی DSL، مجموعه‌ای از تکنولوژی‌ها را در برمی‌گیرد که تفاوت نام آنها در حرف ابتدایی آنهاست. این تکنولوژیها عبارتند از ADSL ، VDSL ، HDSL و SDSL . غالباً به این خانواده از تکنولوژی‌ها، خانواده xDSL گفته می‌شود و می‌توان حروف A ، V ، H و S را به جای حرف x قرار داد.



شکل ۳-۱۶: مدم 56K

❊❊ DSL نامتقارن یا ADSL

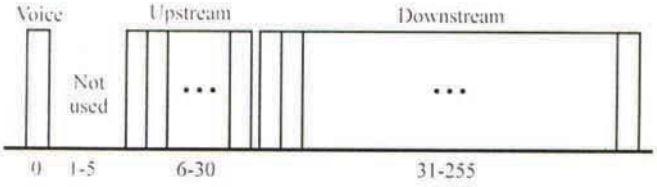
اولین تکنولوژی از این خانواده، ***DSL نامتقارن***[1] یا ADSL است. ADSL مشابه با مدم 56K ، نرخ بیت بیشتری را در عمل دیتاگیری نسبت به دیتادهی در اختیار مشترک می‌گذارد و به همین دلیل به آن، نامتقارن گفته می‌شود. طراحان ADSL پهنای باند خط تلفنی را به صورت نامتقارن بین دیتاگیری و دیتادهی تقسیم کرده‌اند. این سرویس برای شرکتهایی که به

1- Asymmetric DSL : ADSL

www.Abiri.ir

نرخ بیت بالا در هر دو جهت دیتاگیری و دیتادهی نیاز دارند، مناسب نیست. شکل ۳-۱۷ نحوه تقسیم پهنای باند را نشان می‌دهد.

مهم: ADSL یک تکنولوژی مخابراتی نامتقارن است که برای مشترکین منازل مسکونی مناسب است و برای شرکتها مناسب نیست.



شکل ۳-۱۷ : نحوه تقسیم پهنای باند در ADSL

Voice : صدا. کانال 0 برای ارسال صدا به کار می‌رود .

Idle : بی‌کار. از کانالهای 1 تا 5 استفاده نمی‌شود تا یک فاصله فرکانسی بین مخابره صدا و مخابره دیتا بوجود آید.

Upstream : جریان دیتا و کنترل از سمت مشترک به سمت اینترنت. ما در اینجا با همان نام دیتادهی به آن اشاره می‌کنیم. کانالهای 6 تا 30 (جمعاً 25 کانال) برای دیتادهی به کار
می‌روند که از این میان، یک کانال برای کنترل و 24 کانال برای دیتا، مورد استفاده قرار می‌گیرند. با داشتن 24 کانال، هر یک به عرض 4 KHZ (از 4.312 KHZ موجود) و ارسال 15 بیت در هر هرتز، می‌توان به نرخ بیت 24x4000x15 = 1.44 Mbps در جهت دیتادهی، دست یافت.

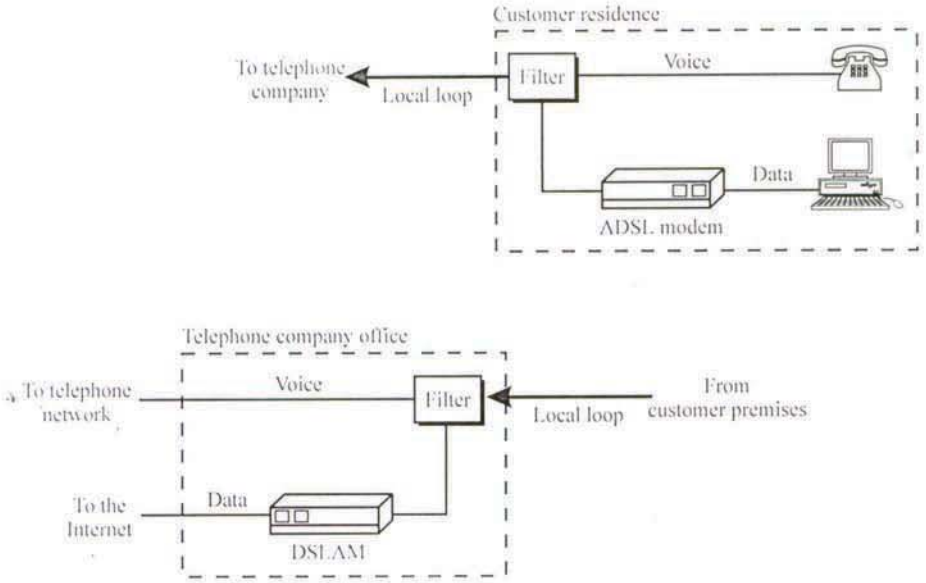
Downstream : جریان دیتا و کنترل از سمت اینترنت به سمت مشترک. ما در اینجا با همان نام دیتاگیری به آن اشاره می‌کنیم. کانالهای31 تا 255 (جمعاً 225 کانال) برای دیتاگیری به کار می‌روند که از این میان، یک کانال برای کنترل و 224 کانال برای دیتا مورد استفاده قرار می‌گیرند. با داشتن 224 کانال، هر یک به عرض 4 KHZ و ارسال 15 بیت در هر هرتز، می‌توان به نرخ بیت 224x4000x15 = 13.44 Mbps دست یافت.

برای تأمین نسبت مطلوب قدرت سیگنال به قدرت نویز (S/N)، در عمل نمی‌توان به سرعتهای فوق رسید و لذا نرخ بیت به مقادیر زیر محدود می‌شود :

Upstream : 64 Kbps to 1 Mbps
Downstream : 500 Kbps to 8 Mbps

www.Abiri.ir

شکل ۳-۱۸ یک مدم DSL را نشان می‌دهد که در محل مشترک، نصب شده است. خط مشترک به فیلتری نصب می‌شود که دیتا و صدا را از هم جدا می‌کند. مدم ADSL ، دیتا را مدوله می‌کند و جریانهای دیتادهی و دیتاگیری را ایجاد می‌کند.



شکل ۳-۱۸: ADSL و DSLAM

وضعیت در سمت دیگر خط تلفن یعنی در مرکز تلفن، متفاوت است. به جای مدم ADSL ، وسیله دیگری به نام ***مالتی‌پلکسر دسترسی به خط دیجیتالی مشترک***[1] نصب شده است که مشابه مدم ADSL عمل می‌کند. به علاوه DSLAM از روی دیتای دریافتی، بسته‌هایی تولید می‌کند و آنها را از طریق اینترنت، ارسال می‌نماید. شکل ۳-۱۸، پیکربندی مربوطه را نشان می‌دهد.

ADSL مخابره نامتقارن را فراهم می‌نماید زیرا که نرخ دیتاگیری، خیلی بیشتر از نرخ دیتادهی است. گرچه این ویژگی، نیاز اکثر مشترکین منازل مسکونی را تأمین می‌نماید، برای شرکتهایی که مقدار زیادی دیتا را ارسال و دریافت می‌نمایند، مناسب نیست. این مشترکین از DSL ***متقارن***[2] که همان SDSL است، استفاده می‌کنند. SDSL پهنای باند را به طور مساوی بین جریانهای دیتادهی و دیتاگیری تقسیم می‌نماید.

1- Digital Subscriber Line Access Multiplexer : DSLAM
2- Symmetric DSL : SDSL

www.Abiri.ir

**** DSL با نرخ بیت بالا یا HDSL**

HDSL[1] به عنوان جایگزینی برای خطوط T-1 (1.544 Mbps) طراحی شد. خط T-1 (که بعداً بحث آن خواهد آمد) از روش تبدیل *AMI*[2] استفاده می‌کند که در فرکانسهای بالا، حساسیت زیادی نسبت به تضعیف دارد. به همین دلیل، طول یک خط T-1 به یک کیلومتر محدود می‌شود و برای رسیدن به سرعتهای بالاتر، باید از ***تکرارکننده***[3] استفاده کرد که هزینه را بالا می‌برد.

**** DSL با نرخ بیت بسیار بالا یا VDSL**

VDSL[4] که روش دیگری مشابه با ADSL است، از کابل کواکسیال، فیبرنوری و زوج سیم به هم تابیده در فواصل کوتاه (از 300m تا 1800m) استفاده می‌کند . تکنولوژی مدولاسیون به کار رفته، تکنولوژی ***DMT***[5] است که نرخ بیت 50 Mbps و 55 Mbps را برای دیتاگیری و 1.5Mbps تا 2.5 Mbps را برای دیتادهی، فراهم می‌نماید.

*** مدم کابلی**

شرکتهای تلفنی سعی می‌کنند، دسترسی به اینترنت با نرخ بیت بالا را در اختیار مشترکین خود گذارند. تکنولوژی DSL چنین امکانی را با استفاده از خطوط تلفنی موجود فراهم کرده است اما با این وجود، DSL از زوج سیمهای بدون حفاظ استفاده می‌کند که بسیار نسبت به تداخل امواج، حساس هستند. به همین دلیل نرخ بیتی که ازطریق این خطوط ارسال می‌شود، حد بالایی دارد. راه حل دیگر، استفاده از شبکه تلویزیون کابلی است .

**** شبکه‌های کابلی معمولی**

در جاهایی که سیگنالهای رادیویی تلویزیونی، یا اصلاً دریافت نمی‌شد و یا با قدرت بسیار کم دریافت می‌شد، از شبکه ***تلویزیون کابلی***[6] استفاده کردند. به آن، ***تلویزیون با آنتن دسته جمعی***[7] نیز گفته می‌شد، زیرا آنتنی که در بالای یک تپه یا ساختمان بلند قرار داشت، سیگنال تلویزیونی را از ایستگاههای رادیویی پخش سیگنال تلویزیونی، دریافت می‌کرد و سپس از طریق کابلهای کواکسیال، آن را به جمع مشترکین خود می‌رساند.

1- High bit rate DSL : HDSL
2- Alternative Mark Inversion : AMI
3- Repeater
4- Very high bit rate DSL : VDSL
5- Discrete Multi Tone : DMT
6- Cable TV
7- Community Antenna TV : CATV

www.Abiri.ir

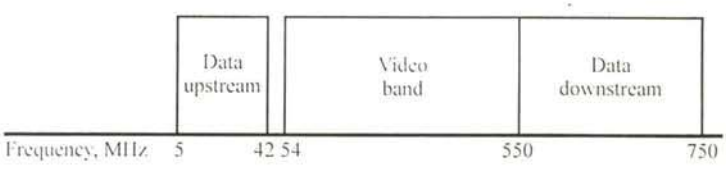
بنابراین مرکز تلویزیون کابلی که head end نیز نامیده می‌شد، سیگنال تلویزیونی را از ایستگاههای رادیویی پخش سیگنالهای تلویزیونی، می‌گرفت و آن را روی کابلهای کواکسیال می‌گذاشت. سیستم معمولی تلویزیون کابلی، از محل تغذیه کابل تا محل مشترک، از کابل کواکسیال استفاده می‌کند. به دلیل تضعیف سیگنال در طول کابل و استفاده از تعداد زیادی تقویت‌کننده، مخابره سیگنال درسیستم معمولی تلویزیون کابلی، به صورت یک طرفه است. سیگنالهای تلویزیونی به صورت یک‌طرفه از head end به سمت منازل مشترکین، ارسال می‌شد.

* شبکه HFC

نسل دوم شبکه‌های تلویزیون کابلی، ***شبکه توأم فیبر و کواکسیال***[1] نامیده می‌شود. این شبکه از کابلهای کواکسیال و فیبرنوری به صورت توأم استفاده می‌کند. محیط انتقال از head end تا جعبه تقسیمی به نام ***گره نوری***[2] از نوع فیبر نوری است. هنوز از محل تقسیم تا منزل مشترکین و نیز داخل منازل مشترکین، از کابل کواکسیال استفاده می‌شود. یک دلیل استفاده توأم فیبر وکابل کواکسیال، دو طرفه کردن مخابره سیگنال، روی شبکه کابلی است.

** پهنای باند

حتی درسیستم HFC، قسمت انتهایی شبکه، یعنی از جعبه تقسیم تا منزل مشترک، هنوز به صورت کابل کواکسیال است. پهنای باند این کابل کواکسیال به طور تقریبی، از 5 MHZ تا 750 MHZ است. شرکتهای تلویزیون کابلی، این پهنای باند را به سه باند مجزا تقسیم کرده‌اند: ویدئو، دیتاگیری و دیتادهی (شکل ۳-۱۹را ببینید).



شکل ۳-۱۹: پهنای باند تلویزیون کابلی

باند فرکانسی ویدئویی: باند ویدئویی که برای ارسال برنامه‌های تلویزیونی از head endبه مشترک، به کار می‌رود، فرکانسهای از 54 MHZ تا 550MHZ را می‌پوشاند. از آنجا که

1- Hybrid Fiber-Coaxial network : HFC
2- Fiber node

www.Abiri.ir

هرکانال تلویزیونی در استاندارد آمریکایی، پهنای باندی برابر با 6MHZ دارد، می‌توان با استفاده از این پهنای باند، بیش از 80 کانال تلویزیونی داشته باشیم.

باند فرکانسی دیتاگیری: جریان دیتاگیری (از اینترنت به مشترک)، باند فرکانسی بالاتر، یعنی از 550 MHZ تا 750 MHZ را به کار می‌برد. این باند را نیز به کانالهای 6MHZ تقسیم می‌کنند. می‌توان از طریق این باند، دیتا را با سرعت 30 Mbps دریافت کرد. استاندارد، عدد 27 Mbps را تعیین کرده است. با این وجود، از آنجا که ***مدم کابلی***[1] از طریق یک کابل 10BASE-T به کامپیوتر متصل می‌شود، سرعت به 10Mbps محدود می‌گردد.

باند فرکانسی دیتادهی: جریان دیتادهی (از مشترک به اینترنت)، باند فرکانسی پایین‌تر یعنی از 5 MHZ تا 42 MHZ را به کار می‌برد. این باند نیز به کانالهای 6 MHZ تقسیم می‌شود. از آنجا که دیتادهی از باند فرکانسی پایین‌تر استفاده می‌کند، حساسیت بیشتری به نویز و تداخل امواج دارد. از نظر تئوری، می‌توان با سرعت 12 Mbps که از رابطه 6 MHz x (2 bits / Hz) بدست می‌آید، دیتادهی را انجام داد. با این وجود، معمولاً نرخ بیت دیتادهی، از 12 Mbps کمتر است.

❊❊ به اشتراک‌گذاری

مشترکین از باندهای دیتادهی و دیتاگیری، به طور مشترک استفاده می‌کنند. پهنای باند دیتادهی فقط 37 MHZ است و معنای آن این است که در جهت دیتادهی، فقط شش کانال 6 MHZ داریم. مشترکین برای ارسال دیتای خود، نیاز به یک کانال در باند فرکانسی دیتادهی دارند. حال این سؤال مطرح می‌شود که چگونه می‌توان از شش کانال موجود، برای ارسال دیتای هزار، دو هزار و یا حتی صد هزار مشترک، موجود در یک منطقه، استفاده کرد؟ راه حل، استفاده از ***تقسیم زمانی***[2] است. شرکت تلویزیون کابلی به طور استاتیک یا دینامیک، یک کانال را در اختیار یک گروه از مشترکین، قرار می‌دهد. اگر مشترکی از یک گروه بخواهد از کانال مربوطه استفاده کند، باید صبر نماید تا کانال، خالی شود. الگوریتم دسترسی به کانال، مشابه با الگوریتم CSMA است که در LANها مورد بحث قرار گرفت.

در جهت دیتاگیری نیز وضعیت مشابهی داریم. باند فرکانسی دیتاگیری حاوی 33 کانال، هر یک به عرض 6 MHZ می‌باشد. هر مشترک برای دریافت دیتای خود از اینترنت، به یک کانال نیاز دارد، اما احتمالاً تعداد مشترکین، بیش از 33 نفر (تعداد کانالهای موجود) می‌باشد. لذا از هر کانال باید برای گروهی از مشترکین استفاده کرد. با این وجود، وضعیت در دیتاگیری قدری با دیتادهی متفاوت است. در دیتاگیری، دیتا به صورت چندگیرندگی، برای

1- Cable Modem
2- time sharing

www.Abiri.ir

گروهی که از یک کانال استفاده می‌کنند، ارسال می‌شود. اگر دیتایی برای یک یا چند نفر از گروه وجود داشته باشد، آن دیتا از طریق کانال مربوط به آن گروه، برای کل آن گروه ارسال می‌شود و لذا، همه مشترکین واقع در آن گروه، دیتای ارسالی را دریافت می‌کنند. اما از آنجا که هر مشترک، آدرس مشخصی دارد که از شبکه تلویزیون کابلی دریافت کرده است، مدم کابلی هر مشترک، آدرس خود را با آدرس دیتای دریافتی، مقایسه می‌کند. اگر دیتا، مربوط به این مشترک باشد، آن را نگه می‌دارد ولی اگر دیتا، مربوط به مشترک دیگری باشد، آن را دور می‌اندازد.

❊❊ وسایل شبکه تلویزیون کابلی

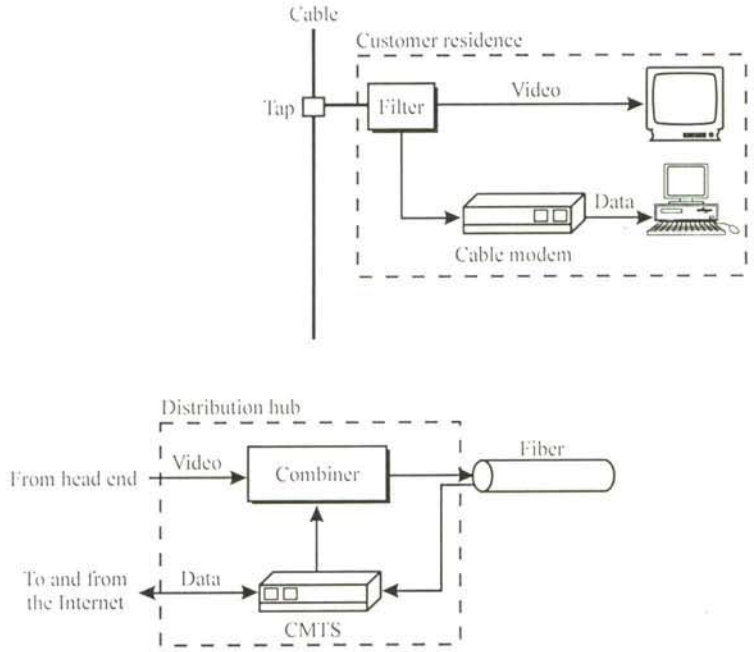
به منظور استفاده از شبکه تلویزیون کابلی برای انتقال دیتا، به دو وسیله کلیدی نیازمندیم : یک ***مدم کابلی***[1] و یک ***سیستم انتقال مدم کابلی***[2]. CM در منزل مشترک نصب می‌شود. CM مشابه با یک مدم ADSL است. شکل ۳-۲۰، محل نصب آن را نشان می‌دهد. CMTS توسط شرکت تلویزیون کابلی در ***مرکز توزیع***[3] نصب می‌شود. CMTS دیتا را از اینترنت دریافت می‌کند و آن را به ***ترکیب‌کننده***[4] می‌دهد تا با برنامه‌های تلویزیونی همراه شود و مجموعه حاصل، از طریق فیبر نوری برای استفاده مشترکین، ارسال گردد. دیتایی که از طرف مشترکین می‌آید نیز از طریق CMTS به روی اینترنت ارسال می‌گردد. شکل ۳-۲۰ مکان نصب CMTS را نشان می‌دهد.

❊ خطوط T

خطوط T خطوط دیجیتالی استانداردی هستند که در سیستمهای آمریکایی به کار می‌روند و عمل مالتی‌پلکسینگ کانالهای صوتی (پس از دیجیتالی کردن) را انجام می‌دهند. اما امروزه از خطوط T ، برای اتصال مشترکین به اینترنت استفاده می‌گردد. همچنین خطوط T ، برای برقراری یک لینک فیزیکی بین مراکز سویچینگ تلفنی نیز مورد استفاده قرار می‌گیرند. خطوط T به فرم تجاری آنها موجود می‌باشند: T-1 و T-3 (جدول ۳-۲ را ملاحظه نمایید).

1- Cable Modem : CM
2- Cable Modem Transmission System : CMTS
3- distribution hub
4- Combiner

www.Abiri.ir



شکل ۳-۲۰: پیکربندی مدم کابلی

Line	*Rate (Mbps)*
T-1	1.544
T-3	44.736

جدول ۳-۲ : نرخ بیت خطوط T

❊ خط T-1

نرخ بیت یک خط T-1 برابر با 1.544 Mbps می‌باشد. از 24 کانال صوتی، نمونه‌برداری می‌شود و هر نمونه با 8 بیت نشان داده می‌شود. برای ایجاد سنکرونیزاسیون، یک بیت به مجموعه اضافه می‌گردد و طول کل فریم را به 193 بیت (193 = 24x8+1) می‌رساند. با ارسال 8000 فریم درثانیه، به نرخ بیت 193x8000=1.544 Mbps می‌رسیم. وقتی که ازخط T-1 برای اتصال مشترک به اینترنت استفاده می‌کنیم، می‌توان از کل ظرفیت خط و یا از قسمتی از ظرفیت آن استفاده کرد.

www.Abiri.ir

❊ خط T-3

نرخ بیت یک خط T-3 برابر با 44.736 Mbps است. این معادل 28 خط T-1 می‌باشد. خیلی از مشترکین ممکن است به کل ظرفیت یک خط T نیاز نداشته باشند. برای برطرف کردن نیاز این کاربران، شرکتهای تلفنی، سرویس ***کسری از خطوط T***[1] را ارائه کرده‌اند و به این ترتیب با مالتی‌پلکس کردن دیتای مشترکین، می‌توان از یک خط T-1 برای نقل و انتقال دیتای همه آنها، استفاده کرد .

❊ SONET

پهنای باند زیاد فیبر نوری برای انتقال دیتای تکنولوژیهای امروزی که به نرخ دیتای بالایی نیاز دارند (مانند کنفرانس ویدئویی) و برای انتقال همزمان دیتای تعداد بسیار زیادی از تکنولوژیهایی که هر یک، به نرخ دیتای پایینی نیاز دارد، مناسب می‌باشد. برای استفاده از کابلهای فیبر نوری، یکسری از استانداردها به نام ***شبکه نوری سنکرون***[2] توسط ANSI ایجاد شده‌اند و کاریرهای پرسرعت دیتا را تعریف می‌کنند.

SONET ابتدا مجموعه‌ای از سیگنالهای الکتریکی به نام ***سیگنالهای انتقال سنکرون***[3] را تعریف کرد. سپس آنها را به سیگنالهای نوری به نام ***کاریر نوری***[4] تبدیل کرد. سرعت انتقال سیگنالهای نوری برابر با 8000 فریم در ثانیه است.

جدول ۳-۳ نرخ بیت سیگنالهای STS و OC را نشان می‌دهد. دقت نمایید که کمترین سرعت در جدول فوق، برابر با 51.840 Mbps است که از سرعت یک خط T-3 یعنی 44.736 Mbps بیشتر است.

لایه دیتالینک

کاربر برای داشتن یک اتصال نقطه به نقطه قابل اطمینان، به پروتوکلی در لایه دیتالینک نیاز دارد. برای نیل به این منظور، متداولترین پروتوکل، ***پروتوکل نقطه به نقطه***[5] می‌باشد.

1- fractional T Line
2- Synchronous Optical Network : SONET
3- Synchronous Transport Signal : STS
4- Optical Carrier : OC
5- Point-to-Point Protocol : PPP

www.Abiri.ir

* پروتوکل PPP

شرکت تلفنی، یک لینک فیزیکی را در اختیار مشترک می‌گذارد اما برای کنترل و مدیریت نقل و انتقال دیتا بر روی آن لینک، به یک پروتوکل خاص در لایه دیتالینک نیاز دارد. برای پاسخگویی به این نیاز، پروتوکل PPP ساخته شد.

STS	*OC*	*Rate (Mbps)*
STS-1	OC-1	51.840
STS-3	OC-3	155.520
STS-9	OC-9	466.560
STS-12	OC-12	622.080
STS-18	OC-18	933.120
STS-24	OC-24	1244.160
STS-36	OC-36	1866.230
STS-48	OC-48	2488.320
STS-96	OC-96	4976.640
STS-192	OC-192	9953.280

جدول ۳-۳: نرخ بیت SONET

** لایه‌های PPP

PPP فقط حاوی لایه‌های فیزیکی و دیتالینک می‌باشد. PPP هیچ پروتوکل خاصی را برای لایه فیزیکی تعریف نکرده است و این وظیفه، به عهده پیاده‌کننده PPP گذاشته شده است که از هر پروتوکل موجودی برای لایه فیزیکی، استفاده کند.

هر پروتوکلی که مورد تأیید ANSI باشد، در لایه فیزیکی PPP قابل استفاده است. PPP در لایه دیتالینک، فرمت فریم و پروتوکلی که کنترل لینک و جابجایی دیتای کاربر توسط آن صورت می‌گیرد را تعریف کرده است. فرمت فریم PPP در شکل ۳-۲۱ نشان داده شده است.

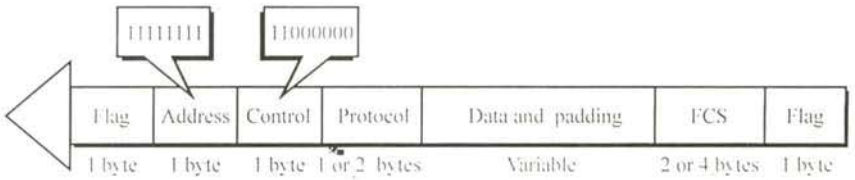
فیلدها به شرح زیرند:

Flag : پرچم. این فیلد، شروع و پایان فریم PPP را مشخص می‌کند. مقدار آن برابر با 01111110 است.

Address : آدرس. از آنجا که PPP در پیکربندی نقطه به نقطه به کار می‌رود، از آدرس همگانی اکثر LANها، یعنی 11111111 ، استفاده می‌کند و به این ترتیب PPP، نیازی به آدرس فیزیکی در لایه دیتالینک ندارد.

www.Abiri.ir

Control : کنترل. به این فیلد، مقدار ثابت 11000000 داده شده است تا مشابه اکثر LANها، نشان دهد که فریم‌ها، فاقد شماره ترتیب هستند و معنی آن این است که هر فریم، مستقل از سایر فریمهاست.



شکل ۳-۲۱ : فریم PPP

Protocol : پروتوکل. این فیلد، نوع دیتایی که در فیلد data حمل می‌شود را تعیین می‌کند. این دیتا می‌تواند دیتای کاربر و یا اطلاعات دیگری باشد.

Data : دیتا. این فیلد، یا حاوی دیتای کاربر و یا حاوی سایر اطلاعات است.

FCS (Frame Check Sequence) : دنباله چک کننده فریم. یک CRC دو بایتی و یا چهار بایتی است که برای آشکارسازی خطا به کار می‌رود.

❊❊ پروتوکل کنترل لینک

پروتوکل کنترل لینک[1] مسئول برقراری ارتباط، مدیریت آن و سپس قطع ارتباط می‌باشد. وقتی که فیلد Data در فریم PPP حاوی اطلاعات مربوط به این پروتوکل باشد، معنی آن این است که PPP در حال برقراری ارتباط، مدیریت آن و یا قطع ارتباط است و لذا در آن لحظه، PPP حامل دیتای کاربر نیست.

❊❊ پروتوکل کنترل شبکه

پروتوکل کنترل شبکه[2] برای دادن قابلیت انعطاف به PPP تعریف شده است. PPP می‌تواند اطلاعات پروتوکلهای مختلف لایه شبکه مانند IP را حمل کند. بعد از آنکه ارتباط برقرار شد، PPP می‌تواند در فیلد Data، بسته‌های IP را جابجا کند.

1- Link Control Protocol : LCP
2- Network Control Protocol : NCP

www.Abiri.ir

PPPoE ***

PPP برای اتصال یک تک کاربر به اینترنت، با استفاده از یک مدم معمولی و یک خط تلفن، طراحی شد. امروزه تکنولوژیهای DSL ، مدم کابلی و بی‌سیم، این امکان را فراهم کرده‌اند که گروهی از کاربران متصل به یک شبکه اترنت، از طریق یک لینک فیزیکی واحد به اینترنت متصل شوند. به عبارت دیگر، کامپیوترهای متصل به شبکه، می‌توانند برای دسترسی به اینترنت، به طور مشترک از یک لینک فیزیکی استفاده کنند. ***PPP روی اترنت***[1] یک پروتوکل جدید است که با استفاده از یک ***تکنیک کشف***[2]، آدرس اترنت کامپیوتری که قرار است به اینترنت وصل شود را پیدا می‌کند. بعد از پیدا شدن آدرس، پروتوکل معمولی PPP برای برقراری ارتباط و سپس جابجایی دیتا، به کار خواهد رفت.

۳-۳ WANهای سویچ شده

شبکه‌های ستون فقرات در اینترنت معمولاً از ***WANهای سویچ شده***[3] استفاده می‌کنند. یک WAN سویچ شده، یک شبکه گسترده است که یک محدوده وسیع (یک استان یا یک کشور) را می‌پوشاند و در چندین نقطه، امکان دسترسی به شبکه را برای کاربران فراهم می‌کند. داخل شبکه WANهای سویچ شده، یک توپولوژی مش از سویچهایی را داریم که به صورت نقطه به نقطه به هم متصل شده‌اند. سویچها (کانکتورهای چند پورت) اتصال چندین ورودی و خروجی به هم را امکان‌پذیر می‌سازند.

تکنولوژی WANهای سویچ شده از جنبه‌های مختلفی با تکنولوژی LAN تفاوت دارد. اول آنکه به جای یک توپولوژی باس یا ستاره، از سویچها استفاده شده است تا چندین مسیر را برقرار کنند. دوم آنکه، تکنولوژی LAN ، یک تکنولوژی ***فاقد اتصال***[4] در نظر گرفته شده است. هیچ رابطه‌ای بین بسته‌های ارسال شده از سوی فرستنده، وجود ندارد. از طرف دیگر، تکنولوژی WAN سویچ شده، یک تکنولوژی ***اتصال‌گرا***[5] است. قبل از آنکه فرستنده بتواند بسته‌ای ارسال کند، باید اتصالی بین فرستنده و گیرنده، برقرار شود. به آن اتصال برقرار شده، یک ***شناسه***[6] داده می‌شود و در طول زمان نقل و انتقال دیتا، از این شناسه استفاده می‌شود. وقتی که جابجایی دیتای کاربر، پایان یافت، به اتصال، خاتمه داده می‌شود. بنابراین به جای آنکه مانند تکنولوژی LAN ، از آدرس مبدأ و مقصد استفاده کنیم، در تکنولوژی WAN سویچ شده، از

1- PPP over Ethernet
2- discovery technique
3- Switched WANs
4- connectionless
5- connection oriented
6- Identifier

www.Abiri.ir

شناسه اتصال استفاده می‌شود. در این بخش درباره سه نوع تکنولوژی متداول برای WANهای سویچ شده، صحبت می‌کنیم. اولین نوع آن یعنی X.25 تقریباً منسوخ شده است. دومین نوع آن یعنی Frame Relay ، هنوز تا چند سال دیگر قابل استفاده است. نوع سوم یعنی ATM ، تکنولوژی غالب می‌باشد. درباره دو مورد اول به اختصار صحبت می‌کنیم و وقت بیشتری را صرف ATM می‌کنیم.

X.25

X.25 که در دهه ۱۹۷۰ معرفی شد، اولین WAN سویچ شده‌ای بود که در اروپا و آمریکا متداول گشت. گرچه هنوز در اروپا از آن استفاده می‌کنند، اما در آمریکا در حال منسوخ شدن می‌باشد. اکثراً از آن به عنوان یک شبکه عمومی برای اتصال کامپیوترهای تکی و یا شبکه‌های LAN استفاده می‌شود. X.25 یک سرویس ***پایانه به پایانه***[1] فراهم می‌نماید. گرچه X.25 به عنوان یک WAN برای حمل بسته‌های IP از یک نقطه دنیا به نقطه دیگری از دنیا به کار رفته است و هنوز هم تا حدودی به کار می‌رود، همواره مشکلی در استفاده از IP روی شبکه X.25 وجود داشته است. IP پروتوکل لایه شبکه (لایه سوم) است. فرض بر این است که بسته‌های IP توسط فریم لایه دیتالینک (لایه دوم) حمل می‌شوند. X.25 که قبل از اینترنت ساخته شد، حاوی سه لایه است و لذا X.25 در لایه شبکه نیز پروتوکل خودش را دارد. به منظور جابجایی بسته‌های IP در شبکه X.25 ، باید آنها را در پروتوکل لایه شبکه X.25 کپسول‌بندی کنیم. این وضعیت مشابه کسی است که اتومبیلی دارد، اما برای جابجایی، مجبور باشد اتومبیلش را روی یک تریلی بگذارد تا بتواند از یک نقطه شهر به نقطه دیگری برود.

مشکل دیگر X.25 آن است که زمانی ساخته شد که محیطهای انتقال، خیلی قابل اطمینان نبودند (از فیبر نوری استفاده نمی‌شد). به همین دلیل، X.25 کارهای زیادی را برای کنترل خطا انجام می‌دهد. این کارها باعث کاهش سرعت انتقال دیتا می‌شود و با تقاضای رو به رشد برای سرعتهای بالاتر، اصلاً سازگاری ندارد.

به دلایل گفته شده، به احتمال زیاد در آینده نزدیک، شاهد حذف X.25 از اینترنت خواهیم بود.

Frame Relay

Frame Relay به عنوان یک تکنولوژی WAN سویچ شده طراحی شد تا جای X.25 را بگیرد. این تکنولوژی حاوی لایه فیزیکی و لایه دیتالینک است. Frame Relay چندین مزیت نسبت به X.25 دارد که می‌توان به موارد زیر اشاره کرد:

1- end-to-end

www.Abiri.ir

نرخ بیت بالا : اگرچه در ابتدا، تکنولوژی Frame Relay برای سرعت 1.544 Mbps (معادل خط T-1) طراحی شد، اما امروزه اکثر پیاده‌سازی‌های آن، می‌توانند تا سرعت 44.736 Mbps (معادل خط T-3) نیز کار کنند.

دیتای دسته‌ای: بعضی از سرویسهای ارائه شده توسط سرویس دهندگان WAN ، فرض را بر این می‌گذارند که کاربر، به نرخ دیتای ثابتی نیاز دارد. به عنوان مثال خط T-1 برای کاربری طراحی شده است که می‌خواهد از خط، با سرعت ثابت 1.544 Mbps استفاده کند. امروزه این نوع سرویس، برای بسیاری از کاربرانی که نیاز به ارسال ***دیتای دسته‌ای***[1] دارند، مناسب نیست زیرا که در ارسال دیتای دسته‌ای، سرعت دیتا در زمانهای مختلف، متفاوت است. به عنوان مثال، ممکن است کاربری بخواهد دیتا را به مدت 2 ثانیه با سرعت 6 Mbps ، به مدت 7 ثانیه با سرعت 0 Mbps (هیچ چیز) و به مدت 1 ثانیه با سرعت 3.44 Mbps ارسال نماید. به عبارت دیگر، در مدت زمان جمعاً 10 ثانیه، به اندازه 15.44 Mbits دیتا بفرستد. گرچه متوسط نرخ بیت ارسالی همان 1.544 Mbps است، اما یک خط T-1 نمی‌تواند چنین سرویسی را به مشترک بدهد و دلیل آن این است که خط T-1 برای سرعت ثابت و نه سرعت متغیر (دیتای دسته‌ای) طراحی شده است. دیتای دسته‌ای به چیزی نیاز دارد که به اصطلاح به آن ***پهنای باند طبق درخواست***[2] گفته می‌شود. کاربر در زمانهای مختلف، به پهنای باندهای مختلف نیاز دارد. Frame Relay می‌تواند دیتای دسته‌ای را بپذیرد. در این تکنولوژی به کاربر، یک نرخ بیت متوسط داده می‌شود ولی کاربر می‌تواند به هنگام نیاز، نرخ بیت لحظه‌ای را از این مقدار متوسط، بالاتر ببرد.

اطلاعات بالاسری کمتر، به دلیل بهبود کیفیت محیطهای انتقال: در خلال دهه گذشته، کیفیت محیطهای انتقال به مقدار قابل توجهی افزایش یافته است. محیطهای انتقال، قابل اطمینان‌تر شده‌اند و خطای کمتری دارند. WANای مورد نیاز نیست که پردازش زیادی را صرف چک کردنهای مکرر خطاها نماید. X.25 عملیات زیادی را برای چک کردن و کنترل خطا انجام می‌دهد. Frame Relay فاقد عملیات چک کردن خطا و فاقد ***تأییدیه‌ها***[3] در لایه دیتالینک می‌باشد. به جای آن، کلیه عملیات چک کردن خطا، بر عهده پروتوکلهای لایه شبکه و لایه انتقال، که از سرویس Frame Relay استفاده می‌کنند، گذاشته شده‌اند.

1- bursty data
2- bandwidth on demand
3- Acknowledgment

www.Abiri.ir

* معماری Frame Relay

وسایلی که کاربر را به شبکه متصل می‌کنند، ***تجهیزات ترمینال دیتا***[1] نامیده می‌شوند. سویچهایی که فریمها را در داخل شبکه سویچها هدایت می‌کنند، ***تجهیزات ختم کننده مدار دیتا***[2] نامیده می‌شوند (شکل ۳-۲۲ را ببینید). Frame Relay غالباً به عنوان یک WAN برای اتصال LANها یا سوپرکامپیوترها به کار می‌رود. در حالت اول، یک ***روتر***[3] یا ***پل***[4] می‌تواند نقش یک DTE را ایفاء کند و به کمک ***خطوط استیجاری***[5]، LAN را به سویچ Frame Relay که یک DCE در نظر گرفته می‌شود، متصل نماید. در حالت دوم، با نصب نرم‌افزار مناسب، خود سوپرکامپیوتر را می‌توان به عنوان یک DTE به کار برد.

* مدارهای مجازی

Frame Relay مشابه سایر WANهای سویچ شده، از ***مدار مجازی***[6] و ***شناسه مدار مجازی***[7] به نام DLCI استفاده می‌کند.

* لایه‌های Frame Relay

Frame Relay فقط حاوی لایه‌های فیزیکی و دیتالینک است. در Frame Relay هیچ پروتوکل خاصی برای لایه فیزیکی تعریف نشده است. به جای آن، این وظیفه به عهده پیاده‌کننده Frame Relay گذاشته شده است که از یک پروتوکل موجود برای لایه فیزیکی، استفاده نماید. هر پروتوکلی در لایه فیزیکی که مورد تأیید ANSI باشد، در Frame Relay قابل استفاده است.

Frame Relay در لایه دیتالینک، از پروتوکل ساده‌ای برای تحویل دیتا از یک DTE به DTE دیگر، استفاده می‌کند.

ATM

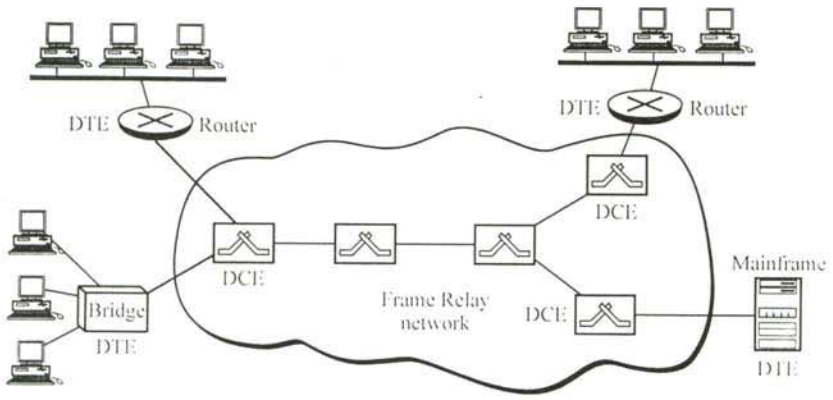
پروتوکل ***مد انتقال آسنکرون***[8] پروتوکل ***رله کردن سلولهای حاوی دیتا***[9] می‌باشد که توسط گردهم‌آیی ATM طراحی شده است و ITU-T آن را پذیرفته است.

1- Data Terminal Equipment : DTE
2- Data Circuit-terminating Equipment : DCE
3- router
4- bridge
5- Leased line
6- Virtual Circuit : VC
7- Virtual Circuit Identifier : VCI
8- Asynchronous Transfer Mode : ATM
9- cell relay protocol

www.Abiri.ir

❋ اهداف طراحی

طراحان ATM ، شش هدف عمده را دنبال کرده‌اند: اولین هدف، که مهمترین هدف نیز محسوب می‌شد، نیاز به یک سیستم انتقال بود که بتواند به شکل بهینه‌ای، از محیطهای انتقال پرسرعت، به ویژه فیبرنوری، به خوبی استفاده نماید. دومین هدف، نیاز به سیستمی بود که بتواند با سیستمهای موجود، مانند شبکه‌های ***سویچینگ بسته‌ها***[1] کار کند و در اتصال به آنها، نیازی به جایگزینی سیستمهای موجود، نداشته باشد و در ضمن، کارآیی آنها را نیز کاهش ندهد. سومین هدف، نیاز به طراحی بود که پیاده‌سازی آن با هزینه کمتری امکان‌پذیر باشد به طوری که هزینه، مانعی در برابر پذیرش آن ایجاد نکند. اگر قرار است که ATM به صورت ستون فقرات مخابرات بین‌المللی درآید، و البته این موضوع، مورد توجه طراحان نیز می‌باشد، باید با هزینه کمی در اختیار کاربرانی قرار گیرد که مایل به استفاده از آن هستند. چهارمین هدف آنکه سیستم جدید باید توانایی کار با سلسله مراتب تجهیزات مخابراتی (لوپهای محلی، مراکز مخابراتی محلی، مراکز مخابراتی راه دور و ...) و پشتیبانی از آنها را داشته باشد. پنجمین هدف، اتصال‌گرا بودن سیستم است تا تحویل صحیح و قابل پیش‌بینی دیتا را تضمین نماید.



شکل ۳-۲۲ : شبکه Frame Relay

بالاخره آخرین هدف آنکه، بسیاری از عملیات لازم الاجرا را تا حد امکان، به سمت سخت‌افزاری شدن و لذا کاهش عملیات نرم‌افزاری، سوق دهد تا سرعت بالا رود.

1- Packet Switching

www.Abiri.ir

* شبکه‌های سلولی

ATM یک ***شبکه سلولی***[1] است. یک ***سلول***[2]، یک واحد دیتای کوچک به طول ثابت است که واحد تبادل دیتا در یک شبکه سلولی محسوب می‌شود. در این نوع شبکه، دیتا به داخل سلولهایی به طول مشابه، حمل می‌شود و این سلولها می‌توانند به شکلی قابل پیش‌بینی و یکنواخت، در شبکه جابه‌جا شوند. سلولها با سایر سلولها، مالتی‌پلکس می‌شوند و داخل شبکه سلولی، مسیردهی می‌شوند. از آنجا که تمام سلولها، کوچک و هم طول هستند، هر مشکلی در رابطه با مالتی‌پلکس کردن بسته‌های با طول متفاوت، برطرف خواهد شد.

مهم: شبکه سلولی از سلول، به عنوان واحد پایه تبادل دیتا، استفاده می‌کند. یک سلول، به عنوان یک بلاک کوچک و به طول ثابت از اطلاعات، تعریف می‌شود.

* ATM آسنکرون

ATM از ***مالتی‌پلکسینگ تقسیم زمانی آسنکرون***[3] استفاده می‌کند (و به همین دلیل به آن، مد انتقال آسنکرون، گفته می‌شود) تا سلولهایی که از کانالهای مختلف می‌آیند را با هم، مالتی‌پلکس کند. ***شیارهای زمانی***[4] خروجی آن، به طول ثابت و برابر با اندازه یک سلول می‌باشند. مالتی‌پلکسرهای ATM یک شیار زمانی را با سلولی که از یکی از کانالهای ورودی خود دریافت کرده است، پر می‌نماید. اگر هیچ یک از کانالهای ورودی، دیتایی برای ارسال نداشته باشند، شیار زمانی خروجی مالتی‌پلکسر، خالی می‌ماند.

شکل ۳-۲۳ نشان می‌دهد که چگونه سلولهای سه ورودی، با هم مالتی‌پلکس می‌شوند. کانال دوم در اولین تیک کلاک، هیچ سلولی ندارد، بنابراین مالتی‌پلکسر، شیار زمانی خروجی خود را با سلول دریافتی از کانال سوم، پر می‌کند. وقتی که همه سلولهای دریافتی از همه کانالها، مالتی‌پلکس شدند، دیگر چیزی برای ارسال روی شیارهای زمانی خروجی وجود نخواهد داشت و لذا آنها، خالی می‌مانند.

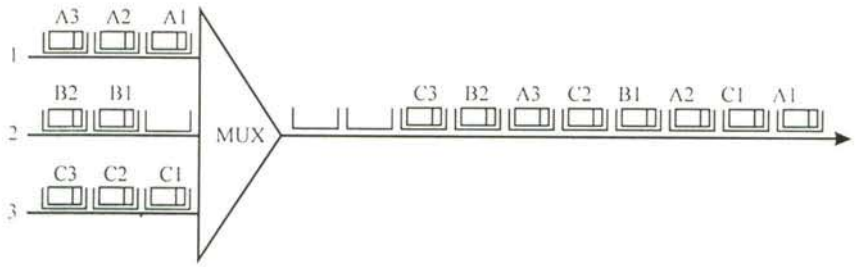
* معماری ATM

ATM یک شبکه سویچ شده است. وسایلی که کاربر به کمک آنها به شبکه متصل می‌شود و ***نقاط پایانی***[5] نامیده می‌شوند، ازطریق یک ***رابط کاربر به شبکه***[6] به سویچهای

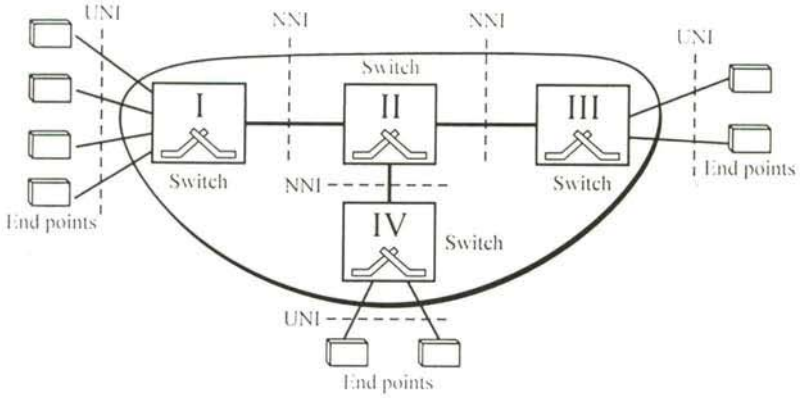
1- cell network
2- cell
3- Asynchronous Time Division Multiplexing
4- time slot
5- end points
6- User to Network Interface : UNI

www.Abiri.ir

داخل شبکه، متصل می‌شوند. رابط بین سویچها، از نوع ***رابط شبکه به شبکه***[1] است. شکل ۳-۲۴ مثالی از یک شبکه ATM را نشان می‌دهد.



شکل ۳-۲۳: مالتی‌پلکسینگ ATM



شکل ۳-۲۴ : معماری یک شبکه ATM

*** اتصال مجازی**

اتصال بین دو نقطه پایانی، از طریق ***مسیرهای انتقال***[2] ، ***مسیرهای مجازی***[3] و ***مدارهای مجازی***[4] برقرار می‌شود. یک TP ، یک اتصال فیزیکی (سیم، کابل، ماهواره و غیره) مابین یک نقطه پایانی و یک سویچ و یا مابین دو سویچ است. دو سویچ را به منزله دو شهر در

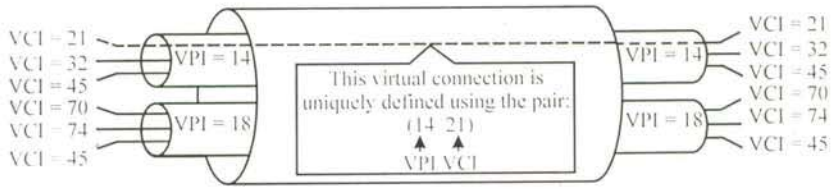
1- Network to Network Interface : NNI
2- Transmission Path : TP
3- Virtual Path : VP
4- Virtual Circuit : VC

www.Abiri.ir

نظر بگیرید. یک TP مشابه مجموعه کلیه اتوبانهایی است که مستقیماً دو شهر را به هم وصل می‌کنند.

یک TP به چندین VP تقسیم می‌شود. یک VP ، یک اتصال یا مجموعه‌ای از اتصالات را بین دو سویچ، برقرار می‌نماید. یک VP را اتوبانی در نظر بگیرید که دو شهر را به هم وصل می‌کند. هر اتوبان، یک VP است. مجموعه همه اتوبانهای بین دو شهر، یک TP است.

شبکه‌های سلولی بر اساس VCها کار می‌کنند. تمام سلولهای متعلق به یک پیغام واحد، در یک VC حرکت می‌کنند و سلولها تا رسیدن به مقصد نهایی، ترتیب اولیه خود را حفظ می‌کنند. مطابق شکل ۳-۲۵، یک VC را به عنوان یکی از باندهای یک اتوبان (VP)، در نظر بگیرید.



شکل ۳-۲۵: مدارهای مجازی (VCها)

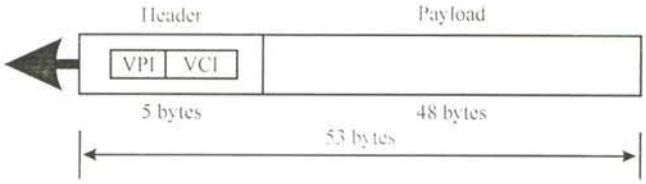
شکل ۳-۲۵ همچنین رابطه بین یک TP (یک اتصال فیزیکی) با VPها (دسته‌ای از VCها که به دلیل آنکه بخشی از مسیرهای آنها مشابه است، در یک دسته قرارگرفته‌اند) و VCها (که به طور منطقی، اتصال بین دو نقطه را فراهم می‌کنند) را نشان می‌دهد.

مهم : توجه نمایید که یک اتصال مجازی، با یک جفت عدد تعریف می‌شود: VPI و VCI.

* سلولها

طول یک سلول، 53 بایت است که از آن، 5 بایت برای هدر و 48 بایت برای حمل دیتای کاربر (که ممکن است کمتر از 48 بایت باشد)، در نظر گرفته شده است. مقادیر VPI و VCI ، بخش عمده‌ای از هدر را پرکرده‌اند. شکل ۳-۲۶ ساختار سلول را نشان می‌دهد.

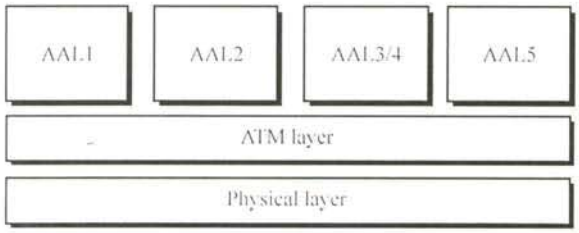
www.Abiri.ir



شکل ۳- ۲۶: یک سلول ATM

❊ لایه‌های ATM

استاندارد ATM سه لایه را تعریف می‌کند. مطابق شکل ۳-۲۷ ، این لایه‌ها از بالا به پایین عبارتند از: ***لایه تطبیق با کاربرد***[1]، لایه ATM و لایه فیزیکی.



شکل ۳- ۲۷ : لایه‌های ATM

❊❊ AAL

AAL ، امکان اتصال شبکه‌های موجود (مانند شبکه‌های جابه‌جایی بسته‌ها) به تجهیزات ATM را فراهم می‌نماید. پروتوکلهای AAL ، اطلاعاتی که از لایه‌های بالاتر می‌آید (مانند اطلاعات بسته‌ها) را دریافت می‌کنند و آنها را به داخل سلولهای با طول ثابت ATM قرار می‌دهند. اطلاعات دریافتی می‌تواند از هر نوعی (صدا، دیتا، صوت، ویدئو) و با نرخ ثابت و یا متغیر، باشد. در گیرنده، معکوس فرایند فوق انجام می‌شود، یعنی سلولها مجدداً به فرمت اولیه، سرهم‌بندی می‌شوند و اطلاعات حاصل به گیرنده، تحویل داده می‌شود.

1- Application Adaptation Layer : AAL

www.Abiri.ir

AAL1 ***

AAL1 از کاربردهایی نظیر ویدئو و صدا پشتیبانی می‌کند که اطلاعات را با نرخ بیت ثابتی ارسال می‌کنند. ATM به کمک AAL1 می‌تواند شبکه‌های تلفنی دیجیتالی موجود، مانند کانالهای صوتی و خطوط T را به هم متصل نماید.

AAL2 ***

ابتدا قرار بود AAL2 از نرخ بیت متغیر، پشتیبانی کند، اما طراحی آن عوض شد. درحال حاضر AAL2 برای حمل ترافیک دیتا با سرعت کم و فریمهای کوتاه، مانند صوت (فشرده شده یا فشرده نشده) و نیز برای حمل ویدئو و فاکس به کار می‌رود. مثال خوبی از کاربرد AAL2 در تلفن مبایل است. به کمک AAL2 می‌توان فریمهای کوتاه را به داخل یک سلول، مالتی‌پلکس کرد.

AAL3/4 ***

ابتدا قرار بود AAL3 ، از سرویس اتصال‌گرا و AAL4 ، از سرویس فاقد اتصال پشتیبانی کنند. اما وقتی که طراحی جلوتر رفت، معلوم شد که عملیات اساسی این دو پروتوکل، یکسان هستند. لذا این دو پروتوکل با هم ترکیب شدند و نام واحد AAL3/4 را به خود گرفتند.

AAL5 ***

AAL3/4 مکانیزمهای پیچیده کنترل خطا و کنترل ترتیب را به کار می‌برد که برای همه کاربردها، مورد نیاز نمی‌باشند. طراحان ATM برای این کاربردها، AAL پنجمی به نام ***لایه تطبیق ساده و کارآمد***[1] را ارائه کرده‌اند. AAL5 فرض را بر این می‌گذارد که کلیه سلولهای متعلق به یک پیغام، به دنبال هم حرکت می‌کنند و عملیات کنترلی، در لایه‌های بالاتر فرستنده پیغام، انجام می‌شوند. AAL5 برای پروتوکل‌های فاقد اتصال که بسته‌ها را جابجا می‌کنند (مانند پروتوکل IP در TCP/IP)، ساخته شده است.

مهم : پروتوکل IP از زیرلایه AAL5 استفاده می‌کند.

1- Simple and Efficient Adaptation Layer : SEAL

www.Abiri.ir

**** لایه ATM**

لایه ATM ، سرویسهای مسیردهی، مدیریت ترافیک، سویچینگ و مالتی‌پلکسینگ سلولها را فراهم می‌نماید. سگمنتهای 48 بایتی که توسط AALها ساخته می‌شوند، به لایه ATM داده می‌شوند و لایه ATM با اضافه کردن یک هدر 5 بایتی به آنها، سلولهای 53 بایتی می‌سازد و این سلولها را برای جابجایی در شبکه، تحویل لایه فیزیکی می‌دهد.

**** لایه فیزیکی**

لایه فیزیکی، نوع محیط انتقال، نرخ بیت، نوع سیگنال‌دهی و نحوه تبدیل سیگنال الکتریکی به سیگنال نوری و بالعکس را تعریف می‌کند. لایه فیزیکی، مکانیزم لازم برای انتقال جریان سلولها به صورت جریانی از بیتها را، به کمک پروتوکلهای فیزیکی انتقال مانند SONET و T3 ، فراهم می‌نماید.

مهم : در فصل بیست و سوم درباره IP over ATM صحبت خواهیم کرد.

۳-۴ وسایل متصل‌کننده

معمولاً LANها یا WANها، جدا از هم کار نمی‌کنند بلکه به یکدیگر و یا به اینترنت وصل شده‌اند. برای اتصال LANها یا WANها، از ***وسایل متصل‌کننده***[1] استفاده می‌کنیم. این وسایل در لایه‌های مختلف مدل OSI کار می‌کنند و ما درباره سه نوع از این وسایل، صحبت می‌کنیم: ***تکرارکننده***[2] یا ***هاب***[3] ، ***پل***[4] یا سویچ دو لایه‌ای و ***روتر***[5] یا سویچ سه‌لایه‌ای. تکرارکننده و هاب در لایه اول مدل OSI کار می‌کنند. پل و سویچ دو لایه‌ای، در اولین دو لایه کار می‌کنند. روترها و سویچهای سه لایه‌ای، در اولین سه لایه کار می‌کنند. شکل ۳-۲۸ لایه‌هایی را نشان می‌دهد که هر وسیله در آنها کار می‌کند.

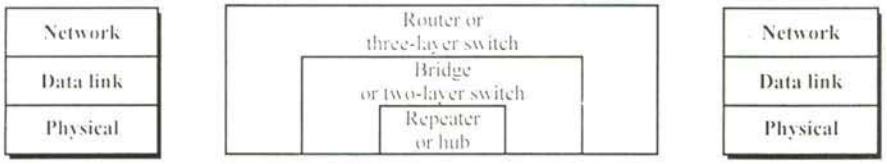
تکرارکننده‌ها

تکرارکننده وسیله‌ای است که فقط در لایه فیزیکی کار می‌کند. سیگنالهایی که در شبکه، حامل اطلاعات هستند، می‌توانند قبل از آنکه تضعیف سبب تغییر ماهیت آنها گردد، تا مسافت مشخصی به پیش روند. تکرارکننده، سیگنال را دریافت می‌کند و قبل از آنکه سیگنال،

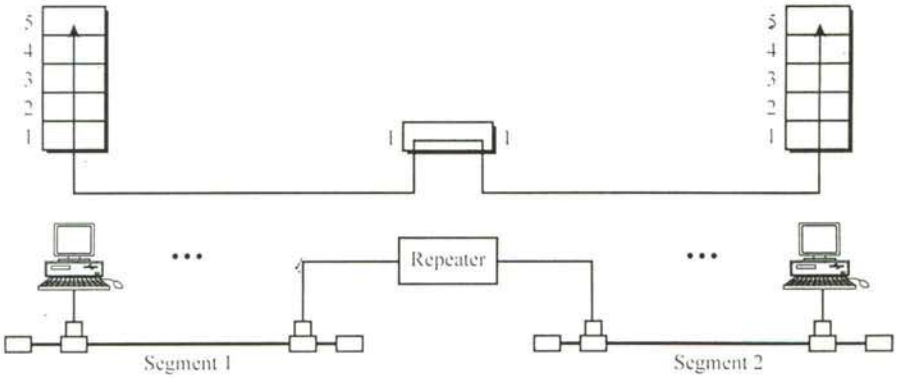
1- Connecting devices
2- Repeater
3- Hub
4- bridge
5- Router

www.Abiri.ir

بسیار ضعیف شود، الگوی بیت اولیه را بازسازی می‌کند و سپس سیگنال بازسازی شده را مجدداً ارسال می‌نماید. مطابق شکل ۳-۲۹ ، تکرارکننده می‌تواند طول فیزیکی یک LAN را افزایش دهد.



شکل ۳-۲۸: وسایل متصل‌کننده



شکل ۳-۲۹: تکرارکننده

در واقع یک تکرارکننده، دو LAN را به هم وصل نمی‌کند بلکه دو قطعه یا سگمنت از یک LAN مشابه را به هم وصل می‌نماید. سگمنتهایی که به هم متصل می‌شوند، هنوز قسمتی از یک LAN واحد هستند.

مهم: تکرارکننده، سگمنتهای یک LAN را به هم متصل می‌کند.

تکرارکننده می‌تواند بر محدودیت طول اترنت 10BASE5 فائق آید. در این استاندارد، طول کابل به ۵۰۰ متر محدود شده است. برای افزایش طول، می‌توان بین سگمنتهایی که هر یک به طول ۵۰۰ متر هستند، از تکرارکننده استفاده کرد. دقت نمایید که هنوز کل شبکه را یک LAN واحد در نظر می‌گیریم. آن قسمتهایی از شبکه که توسط تکرارکننده‌ها از هم جدا

www.Abiri.ir

شده‌اند، **سگمنت**[1] نامیده می‌شوند. تکرارکننده به صورت یک ***گره دو پورت***[2] در لایه فیزیکی کار می‌کند. وقتی که تکرارکننده فریمی را از یک پورت دریافت می‌کند، آن را بازسازی می‌کند و سپس روانه پورت دیگر خود می‌نماید.

مهم: تکرارکننده، همه بیتها را منتقل می‌کند و هیچ بیتی را فیلتر نمی‌کند.

به نظر می‌رسد که می‌توان تکرارکننده را با ***تقویت‌کننده***[3] مقایسه کرد، اما این مقایسه صحیح نیست. یک تقویت کننده نمی‌تواند تفاوتی بین سیگنال مطلوب و نویز قائل شود و لذا هرچه به آن داده می‌شود را تقویت می‌نماید. تکرارکننده، سیگنال را تقویت نمی‌کند بلکه آن را بازسازی می‌نماید. وقتی که تکرارکننده یک سیگنال ضعیف شده یا خراب شده را دریافت می‌کند، سیگنال اولیه را بیت به بیت، با همان قدرت اولیه، بازسازی می‌کند.

مهم: تکرارکننده، سیگنال را بازسازی می‌کند و لذا یک تقویت‌کننده نیست.

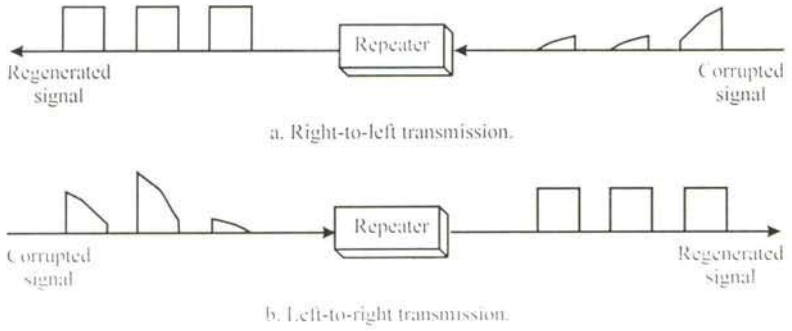
محل تکرارکننده روی لینک، مهم است. تکرارکننده باید جایی روی لینک قرار گیرد که قبل از آنکه نویز، معنی بیتهای سیگنال را عوض کند، سیگنال بتواند خود را به تکرارکننده برساند. نویز کم می‌تواند ولتاژ بیت را تغییر دهد، بدون آنکه معنی بیت عوض شود (شکل ۳-۳۰ را ببینید). با این وجود اگر همان بیت، مسافت بیشتری را طی کند، نویز جمع شده می‌تواند معنی آن را تغییر دهد. در این حالت، ولتاژ اولیه قابل بازیافت نیست و خطای وارده باید تصحیح شود. اگر قبل از آنکه معنی بیتها تغییرکنند، تکرارکننده‌ای روی لینک قرار داده شود، می‌توان ولتاژ بیتها را به درستی تشخیص داد و لذا ولتاژ اولیه آنها را مجدداً بازسازی نمود.

هابها

گرچه درحالت کلی، کلمه هاب می‌تواند به هر وسیله متصل‌کننده‌ای اشاره کند، اما هاب معنی خاصی دارد. یک هاب، در واقع یک تکرارکننده چند پورت است. غالباً از هاب برای ایجاد اتصالاتی بین ایستگاهها در یک توپولوژی ستاره استفاده می‌شود.

1- Segment
2- two- port node
3- amplifier

www.Abiri.ir



شکل ۳-۳۰: کار یک تکرارکننده

پلها

پل[1] هم در لایه فیزیکی و هم در لایه دیتالینک کار می‌کند. پل در لایه فیزیکی، سیگنالی که دریافت می‌کند را دوباره بازسازی می‌کند و در لایه دیتالینک، می‌تواند آدرس فیزیکی (MAC) مبدأ و مقصد موجود در فریم را تشخیص دهد. دقت کنید که اگر تکرارکننده‌ها در فواصل مناسبی روی لینکها قرار داده شوند، می‌توان بیتها را تا هر فاصله دلخواهی منتقل کرد. با این وجود، حوزه تصادف اجازه چنین کاری را نمی‌دهد. برای غلبه بر محدودیت حوزه تصادف، به پلها نیاز داریم.

* فیلتر کردن

ممکن است این سؤال پیش آید که تفاوت بین پل و تکرارکننده چیست؟ پل، توانایی فیلتر کردن دارد. پل می‌تواند آدرس مقصد فریم را چک کند و تصمیم بگیرد که باید فریم را به پیش راند یا آن را دور اندازد. اگر قرار است فریم را به پیش راند، باید پورت خروجی، مشخص شود. پل جدولی دارد که آدرسها را به پورتها مرتبط می‌نماید.

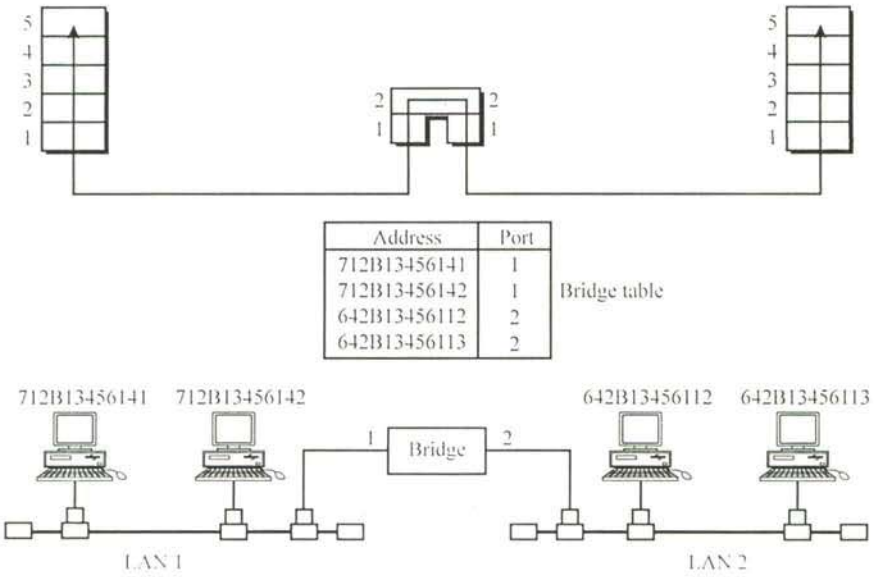
مهم : پل جدولی دارد که در فیلتر کردن به او کمک می‌کند.

اجازه دهید مثالی بزنیم. در شکل ۳-۳۱، دو LAN توسط یک پل به هم متصل شده‌اند. اگر فریمی که برای وسیله 712B13456142 فرستاده شده است، به پورت 1 برسد، پل از جدول خود کمک می‌گیرد تا پورتی را پیدا کند که این وسیله به آن متصل است. بر اساس محتویات این جدول، فریمهای ارسالی برای وسیله 712B13456142 باید از پورت 1 خارج شوند و از آنجا که فریم دریافتی از طریق همین پورت وارد پل شده است، لذا نیازی نیست که

1- bridge

www.Abiri.ir

پل، فریم دریافتی را مجدداً روی پورت 1 بگذارد (یعنی عمل ***به پیش‌رانی***[1] لازم نیست) و لذا آن را دور می‌اندازد. از سوی دیگر اگر فریمی برای ماشین 712B13456141 ، از طریق پورت 2 پل دریافت شود، فریم باید از طریق پورت 1 خارج شود و لذا عمل به پیش‌رانی، لازم است. در حالت اول، LAN2 ترافیکی نخواهد داشت. در حالت دوم، هر دو LANها، ترافیک دارند. در این مثال، یک پل با دو پورت را نشان دادیم اما یک پل واقعی، پورتهای بیشتری دارد. همچنین توجه نمایید که یک پل، آدرسهای فیزیکی موجود در فریم را تغییر نمی‌دهد.



مهم : پل، آدرسهای فیزیکی موجود در فریم را تغییر نمی‌دهد.

* پل شفاف

پل شفاف[2] پلی است که وسایل متصل به آن، کاملاً از وجود آن بی‌خبر هستند. اگر چنین پلی از سیستم، خارج شود و یا به آن اضافه شود، پیکربندی مجدد ایستگاهها، مورد نیاز نیست. بنابه مشخصه IEEE 801.1d ، سیستمی که مجهز به پلهای شفاف است، باید دارای شرایط زیر باشد:

۱- فریمها باید از یک ایستگاه به ایستگاه دیگر به پیش‌رانده شوند.

1- forwarding
2- transparent bridge

www.Abiri.ir

۲- جدولی که به کار می‌رود و به آن ***جدول به پیش‌رانی***[1] گفته می‌شود، باید به طور اتوماتیک از روی فریمهایی که در سیستم جابه‌جا می‌شوند، ساخته و پر شود.

۳- نباید **مسیر بسته**[2] در سیستم ایجاد گردد.

* به پیش‌رانی

همانطور که در بخش قبل توضیح داده شد، یک پل شفاف باید بتواند به طور صحیحی فریمها را به پیش‌راند.

* یادگیری

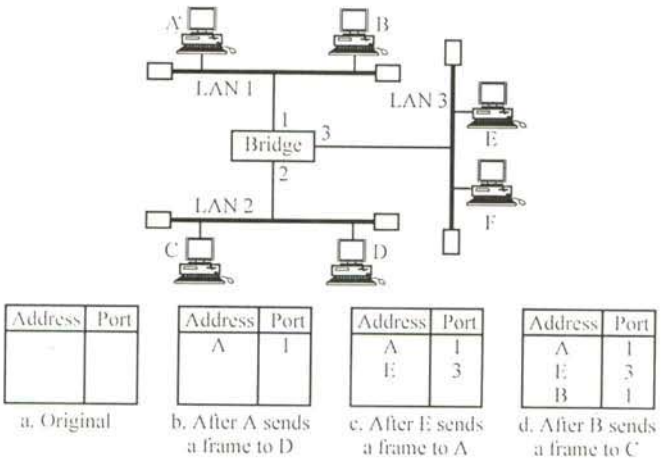
اولین پلها جداولی داشتند که استاتیک یا ثابت بودند. درخلال نصب و راه اندازی پل، مدیر سیستم می‌بایست به طور دستی المانهای جدول را وارد کند. این فرآیندی ساده اما غیر عملی بود. اگر ایستگاهی اضافه یا حذف می‌شد، لازم بود که به صورت دستی، محتویات جدول تغییر داده شود. همین وضعیت در حالتی که آدرس فیزیکی وسیله‌ای عوض می‌شد (که غالباً اتفاق می‌افتد) نیز رخ می‌دهد. به عنوان مثال، نصب یک کارت شبکه جدید به معنی یک آدرس فیزیکی جدید است.

راه حل بهتر از به کارگیری جدول استاتیک، استفاده از یک جدول دینامیک است که به طور اتوماتیک، آدرسها را به پورتها می‌نگارد. برای آنکه جدولی دینامیک شود، به پلی نیاز داریم که محتوای جدول آن، به طور اتوماتیک از روی جابه‌جایی فریمها پر شود. پل برای انجام این کار، هم آدرس مبدأ و هم آدرس مقصد را بررسی می‌کند. آدرس مقصد برای تصمیمات به پیش‌رانی فریمها به کار می‌رود. آدرس مبدأ برای اضافه کردن المانها به جدول و به روزرسانی محتوای جدول، به کار می‌رود. اجازه دهید با استفاده از شکل ۳-۳۲ ، این موضوع را بیشتر بررسی کنیم .

۱-وقتی که ایستگاه A، فریمی را برای ایستگاه D می‌فرستد، پل برای ایستگاه A و ایستگاه D ، المانی در جدول خود ندارد. پل، فریم را روی همه پورتهای خود می‌گذارد و به عبارت دیگر، عمل ***ارسال سیل‌آسا***[3] که به معنی گذاشتن فریم روی همه پورتها است را انجام می‌دهد. با این وجود، پل با نگاه کردن به آدرس مبدأ، یاد می‌گیرد که ایستگاه A باید روی LANای قرار داشته باشد که به پورت 1 پل متصل است. پل این المان را به جدول خود اضافه می‌کند. اکنون جدول، اولین المان خود را بدست آورده است.

1- forwarding table
2- loop
3- flooding

www.Abiri.ir



شکل ۳-۳۲: پل در وضعیت یادگیری

۲- ایستگاه E فریمی را برای ایستگاه A می‌فرستد. با توجه به آنکه پل، المانی در جدول خود برای A دارد که می‌گوید وسیله A به پورت شماره 1 پل متصل شده است، بنابراین فریم را فقط روی پورت شماره 1 خود قرار می‌دهد. این عمل، ***فیلتر کردن***[1] نامیده می‌شود. بنابراین در این حالت، عمل ارسال سیل‌آسا صورت نمی‌گیرد و فریم فقط روی یک پورت گذاشته می‌شود. به علاوه پل، از آدرس مبدأ فریم یعنی آدرس ایستگاه E ، استفاده می‌کند و المان دوم را در جدول خود می‌نویسد.

۳- وقتی که ایستگاه B، فریمی را برای ایستگاه C می‌فرستد، پل، المانی در جدول خود درباره C ندارد و لذا فریم دریافتی را روی همه پورتهای خود قرار می‌دهد (ارسال سیل‌آسا) و سپس المان دیگری را به جدول خود اضافه می‌کند.

۴- همچنان که پل، فریمها را به پیش می‌راند، عمل ***یادگیری***[2] نیز انجام می‌شود و لذا نحوه اتصال وسایل به پورتهای پل، به طور اتوماتیک مشخص می‌شود.

سویچ دو لایه

کلمه ***سویچ***[3] را باید با دقت به کار بریم زیرا که سویچ ممکن است دو معنی متفاوت داشته باشد و به همین دلیل باید با اضافه کردن ***سطحی***[4] که سویچ در آن کار می‌کند، منظورمان از کلمه سویچ را روشن‌تر بیان کنیم. می‌توانیم یک ***سویچ دو لایه***[5] و یا یک

1- filtering
2- learning
3- Switch
4- Level
5- two-layer switch

www.Abiri.ir

سویچ سه لایه[1] داشته باشیم. یک سویچ سه لایه، در لایه شبکه کار می‌کند و لذا یک نوع روتر است. سویچ دو لایه، در لایه دیتالینک کار می‌کند.

یک سویچ دو لایه، یک پل با تعداد زیادی پورت است و طراحی آن، عملکردی بهتر از پل را به آن می‌دهد. یک پل با تعداد کمی پورت، صرفاً می‌تواند تعداد محدودی LAN را به هم متصل نماید. سویچی که پورتهای زیادی دارد، می‌تواند به هر وسیله، یک پورت بدهد و لذا هر وسیله، می‌تواند پورت خاص خودش را روی سویچ داشته باشد. در این حالت، هیچگونه تصادفی نیز پیش نخواهد آمد. در این کتاب به منظور روشن بودن مطلب، از کلمه پل برای سویچ دو لایه استفاده می‌کنیم.

روترها

روتر یک وسیله سه لایه است. روتر در لایه شبکه و لایه‌های پایین‌تر آن یعنی لایه‌های دیتالینک و فیزیکی کار می‌کند. روتر در لایه فیزیکی، پس از دریافت سیگنال، آن را بازسازی می‌کند. روتر در لایه دیتالینک، آدرسهای فیزیکی (مبدأ و مقصد) موجود در فریم را چک می‌کند. روتر در لایه شبکه، آدرسهای لایه شبکه (مثلاً آدرسهای IP) را چک می‌کند. دقت کنید که پلها، حوزه تصادف را تغییر می‌دهند اما روترها، ***حوزه پخش همگانی***[2] را تغییر می‌دهند.

مهم : روتر یک وسیله سه لایه (فیزیکی، دیتالینک و شبکه) است.

یک روتر می‌تواند LANها را به یکدیگر متصل نماید، می‌تواند WANها را به یکدیگر متصل نماید و بالاخره می‌تواند LANها و WANها را به یکدیگر متصل نماید. به عبارت دیگر، روتر یک وسیله شبکه‌بندی است و می‌تواند شبکه‌های مختلف را به هم وصل کند و یک شبکه‌بندی بسازد. طبق این تعریف، دو شبکه (WAN یا LAN) که توسط یک روتر به هم وصل شده‌اند، یک ***اینترنت***[3] را شکل می‌دهند.

سه تفاوت عمده بین روتر و تکرارکننده یا پل وجود دارد:

۱- یک روتر، برای هر یک از ***رابطهایش***[4] یک آدرس منطقی و برای رابطهای اترنت خود، علاوه بر آدرس منطقی، یک آدرس فیزیکی نیز دارد.

1- three-layer switch
2- broadcast domain
3- internet
4- Interface

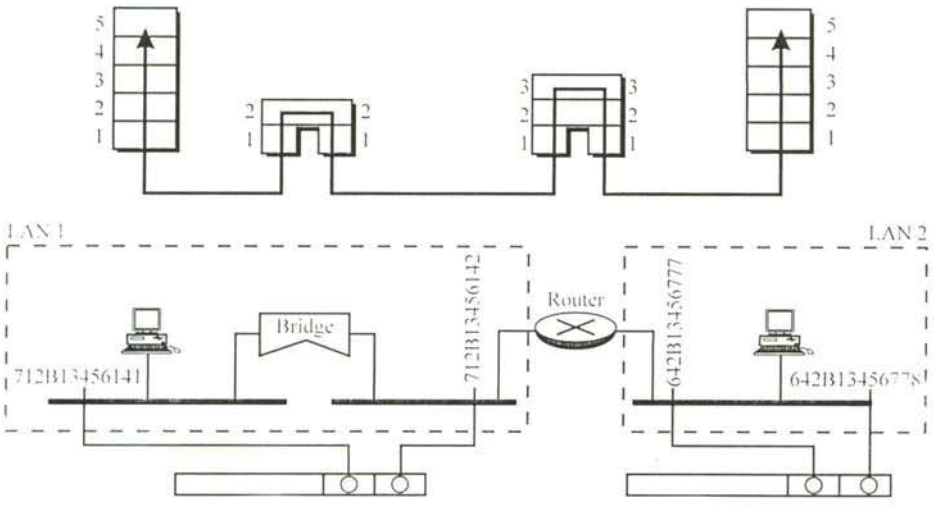
www.Abiri.ir

۲- روتر فقط روی آن بسته‌هایی عمل می‌کند که آدرس فیزیکی مقصد آنها برابر با آدرس فیزیکی رابطی از روتر باشد که بسته از طریق آن رابط، دریافت شده است.

۳- وقتی که روتر بسته را به پیش می‌راند، آدرسهای فیزیکی مبدأ و مقصد بسته را تغییر می‌دهد.

مهم : یک تکرارکننده یا یک پل، سگمنتهای یک LAN را به هم متصل می‌نماید. یک روتر، LANها یا WAN های مستقل را به هم متصل می‌کند و یک اینترنت می‌سازد.

اجازه دهید مثالی بزنیم. درشکل ۳-۳۳، دو LAN را نشان داده‌ایم که روتری آنها را به هم وصل کرده است. LAN سمت چپ، دو سگمنت دارد که یک پل آنها را به هم وصل کرده است. روتر، آدرسهای فیزیکی مبدأ و مقصد بسته را تغییر می‌دهد. وقتی که بسته در LAN سمت چپ حرکت می‌کند، آدرس فیزیکی مبدأ آن، آدرس ایستگاه فرستنده است و آدرس فیزیکی مقصد آن، آدرس روتر است. وقتی که همان بسته در LAN سمت راست حرکت می‌کند، آدرس فیزیکی مبدأ آن، آدرس روتر است و آدرس فیزیکی مقصد آن، آدرس ایستگاه مقصد است.



شکل ۳-۳۳: مثالی از مسیردهی

روترها بسته‌ها را مابین شبکه‌های متصل به هم، جابه‌جا می‌کنند. آنها بسته‌ها را از یک شبکه، به شبکه مقصد مورد نظر در اینترنت می‌رسانند. روترها در یک شبکه، مانند ایستگاهها عمل می‌کنند اما روترها برخلاف اکثر ایستگاهها که فقط به یک شبکه متصل هستند، به چندین شبکه متصل می‌باشند.

www.Abiri.ir

مهم : یک روتر، آدرس‌های فیزیکی موجود در یک بسته را تغییر می‌دهد.

پس از آنکه درباره آدرس‌های IP صحبت کردیم، مطالب بیشتری درباره روترها و مسیردهی خواهیم آموخت.

سویچ سه لایه

یک سویچ سه لایه، یک روتر است که طراحی آن چنان بهینه شده است که عملکرد بهتری را از خود نشان دهد. یک سویچ سه لایه می‌تواند یک بسته را خیلی سریعتر از یک روتر معمولی دریافت کند، پردازش نماید و در نهایت به سمت مقصد نهایی، هدایت نماید اما دقت کنید که تفاوتی بین عملیات یک سویچ سه لایه و یک روتر وجود ندارد. در این کتاب به منظور روشن بودن مطلب، از کلمه سویچ سه لایه استفاده نمی‌کنیم و صرفاً کلمه روتر را به کار می‌بریم.

۳-۵ عبارات کلیدی

1000BASE-CX
1000BASE-LX
1000BASE-SX
1000BASE-T
100BASE-FX
100BASE-T4
100BASE-TX
100BASE-X
10BASE2
10BASE5
10BASE-FL
10BASE-T
نقطه دسترسی (Access Point : AP)
تقویت کننده (amplifier)
لایه تطبیق با کاربرد (Application Adaptation Layer : AAL)
خط دیجیتالی نامتقارن مشترک (Asymmetric Digital Subscriber Line : ADSL)
مالتی‌پلکسینگ تقسیم زمانی آسنکرون
(Asynchronous Time-Division Multiplexing)
مد انتقال آسنکرون (Asynchronous Transfer Mode : ATM)
سویچ ATM (ATM switch)
پهنای باند طبق درخواست (bandwidth on demand)
مجموعه سرویسهای پایه (Basic Service Set : BSS)

www.Abiri.ir

پل (bridge)

دیتای دسته‌ای (bursty data)

مدم کابلی (Cable Modem : CM)

سیستم انتقال مدم کابلی (Cable Modem Transmission System : CMTS)

تلویزیون کابلی (Cable TV)

دسترسی چندگانه به همراه تشخیص حامل و جلوگیری از وقوع تصادف
(Carrier Sense Multiple Access with Collision Avoidance : CSMA/CA)

دسترسی چندگانه به همراه تشخیص حامل با آشکارسازی تصادف
(Carrier Sense Multiple Access with Collision Detection : CSMA/CD)

سلول (Cell)

تصادف (Collision)

تلویزیون با آنتن دسته جمعی (Community Antenna TV : CATV)

وسیله متصل‌کننده (Connecting device)

آدرس مقصد (Destination Address : DA)

خط دیجیتالی مشترک (Digital Subscriber Line : DSL)

مالتی‌پلکسر دسترسی به خط دیجیتالی مشترک
(Digital Subscriber Line Access Multiplexer : DSLAM)

طیف گستردهٔ دنباله مستقیم
(Direct Sequence Spread Spectrum : DSSS)

تابع هماهنگی توزیع شده (Distributed Coordination Function : DCF)

دیتاگیری (Downloading)

اترنت (Ethernet)

مجموعه سرویسهای توسعه یافته (Extended Service Set : ESS)

اترنت سریع (Fast Ethernet)

فیلتر کردن (Filtering)

فریم (Frame)

دنباله چک کننده فریم (Frame Check Sequence : FCS)

Frame Reley

طیف گستردهٔ جهش فرکانسی (Frequency Hoppig Spread Spectrum : FHSS)

گیگابیت اترنت (Gigabit Ethernet)

مشکل ترمینال پنهان (hidden terminal problem)

DSL با نرخ بیت بالا (High bit rate Digital Subscriber Line : HDSL)

www.Abiri.ir

DSSS پر سرعت (High-Rate DSSS : HR-DSSS)

هاب (hub)

شبکه توام فیبر و کواکسیال (Hybrid Fiber – Coaxial network : HFC network)

پروتوکل کنترل لینک (Link Control Protocol : LCP)

شبکه محلی (Local Area Network : LAN)

لوپ محلی (Local Loop)

پروتوکل کنترل شبکه (Network Control Protocol : NCP)

کارت شبکه (Network Interface Card : NIC)

رابط شبکه به شبکه (Network to Network Interface)

کاریر نوری (Optical Carrier : OC)

مالتی‌پلکسینگ متعامد تقسیم فرکانسی

(Ortogonal Ferquency Division Multiplexing : OFDM)

تابع هماهنگی نقطه‌ای (Point Coordination Function : PCF)

پروتوکل نقطه به نقطه (Point to Point Protocol : PPP)

PPP روی اترنت (PPP over Ethernet : PPPoE)

مقدمه (Preamble)

خط دیجیتالی نامتقارن مشترک با قابلیت تطبیق نرخ دیتا

(Rate Adaptive Asymmetrical Digital Subscriber Line : RADSL)

تکرارکننده (Repeater)

روتر (Router)

سگمنت (Segment)

لایه تطبیق ساده و کارآمد (Simple and Efficient Adaptation Layer : SEAL)

آدرس فیزیکی (Source Address : SA)

طیف گسترده (Spread Spectrum)

شاخص شروع فریم (Start Frame Delimiter : SFD)

سویچ (Switch)

اترنت سویچ شده (Switched ethernet)

خط دیجیتالی متقارن مشترک (Symmetric Digital Subscriber Line : SDSL)

سلسله طبقات دیجیتالی سنکرون (Synchronous Digital Hierarchy : SDH)

شبکه نوری سنکرون (Synchronous Optical Network : SONET)

ماجول انتقال سنکرون (Synchronous Transport Module : STM)

www.Abiri.ir

سیگنال انتقال سنکرون (Synchronous Transport Signal : STS)

خطوط T (T lines)

سویچ سه لایه (Three layer switch)

مسیر انتقال (Transmission Path : TP)

پل شفاف (transparent bridge)

سویچ دو لایه (two layer switch)

خط T-1 (T-1 line)

خط T-3 (T-3 line)

دیتادهی (Uploading)

باند فرکانسی جریان دیتادهی (Upstream frequency band)

رابط کاربر به شبکه (User-to-Network Interface : UNI)

DSL با نرخ بیت بسیار بالا (Very high bit rate Digital Subscriber Line : VDSL)

مدار مجازی (Virtual Circuit : VC)

شاخص مدار مجازی (Virtual Circuit Identifier : VCI)

شاخص اتصال مجازی (Virtual Connection Identifier : VCI)

مسیر مجازی (Virtual Path : VP)

شاخص مسیر مجازی (Virtual Path Identifier :VPI)

V.90

V.92

شبکه گسترده (Wide Area Network : WAN)

X.25

۳-۶ خلاصه

- اینترنت پروتوکلی است که در شبکه‌های LAN ، بسیار به کار می‌رود.
- اترنت معمولی از CSMA/CD با نرخ بیت 10 Mbps و حوزه تصادف 2500 متر استفاده می‌کند.
- لایه دیتالینک اترنت از زیرلایه LLC و زیرلایه MAC ساخته شده است.
- زیرلایه MAC ، مسئول اجرای CSMA/CD است .
- هر ایستگاهی روی شبکه اترنت، یک آدرس 48 بیتی دارد که روی کارت شبکه آن، حک شده است.
- پیاده‌سازی‌های متداول اترنت 10 Mbps عبارتند از :10BASE5، 10BASE2، 10BASE-T و 10BASE-FL.

www.Abiri.ir

- اترنت سریع از CSMA/CD با نرخ بیت 10 Mbps و حوزه تصادف 250 متر استفاده می‌کند.
- 100BASE-TX، 100BASE-FX و 100BASE-T4 پیاده‌سازی‌های متداول اترنت سریع می‌باشند.
- نرخ بیت گیگابیت اترنت برابر با 1000 Mbps است. پیاده‌سازی‌های متداول آن عبارتند از: 1000BASE-SX، 1000BASE-LX و 1000BASE-T.
- استاندارد IEEE 802.11 برای LANهای بی‌سیم، دو مجموعه سرویس را تعریف می‌کند: مجموعه سرویس‌های پایه (BSS) و مجموعه سرویسهای توسعه‌یافته (ESS). یک ESS از دو یا چند BSS ساخته شده است. هر BSS باید یک نقطه دسترسی AP داشته باشد.
- لایه فیزیکی LANهای بی‌سیم از متد طیف گستردهٔ جهش فرکانسی (FHSS)، طیف گستردهٔ دنباله مستقیم (DSSS) و مالتی‌پلکسینگ متعامد تقسیم فرکانسی (OFDM) و DSSS پر سرعت (HR-DSSS) استفاده می‌کنند.
- FHSS یک متد تولید سیگنال است که در آن دنباله‌ای از فرکانسهای کاریر به کار می‌رود تا محافظتی در مقابل هکرها ایجاد کند.
- درDSSS یک تکه کد (Chip Code) به جای هر بیت، گذاشته می‌شود.
- در OFDM ، هر منبع از تمام کانالهای پهنای باند موجود استفاده می‌کند.
- HR-DSSS یک DSSS با متد کدینگی به نام کلیدزنی مکمل کد (CCK) است.
- متد دسترسی درLAN بی سیم، CSMA/CA است.
- یک اتصال نقطه به نقطه به اینترنت، با استفاده از خطوط معمولی تلفن و مدمهای معمولی، خطوط DSL، مدمهای کابلی، خطوط T یا شبکه‌های SONET امکان‌پذیر است.
- پروتوکل PPP برای کاربرانی طراحی شده است که نیاز به یک اتصال نقطه به نقطه به اینترنت دارند.
- PPP درلایه‌های فیزیکی و دیتالینک مدل OSI کار می‌کند.
- X.25 یک WANسویچ شده است ولی به تدریج سایر تکنولوژی‌ها، جای آن را می‌گیرند.
- Frame Relay کنترل خطایی که به طور گسترده در پروتوکل X.25 انجام می‌شود را حذف کرده است. Frame Relay در لایه‌های فیزیکی و دیتالینک مدل OS1 کار می‌کند.
- ATM یک پروتوکل رله کردن سلولهاست که برای پشتیبانی از جابه‌جایی دیتا، صدا و ویدئو، روی محیطهای انتقال پرسرعت مانند فیبر نوری، ساخته شده است.
- یک واحد دیتای ATM ، یک سلول 53 بایتی، متشکل از 5 بایت هدر و 48 بایت دیتا، می‌باشد.

www.Abiri.ir

- استاندارد ATM ، سه لایه را تعریف می‌کند: AAL، لایه ATM و لایه فیزیکی.
- چهار نوع مختلف AAL وجود دارد که هر یک، برای نوع خاصی از دیتای کاربر به کار می‌رود. TCP/IP از AAL5 استفاده می‌کند و AAL5 دیتایی که از یک شبکه فاقد اتصال (مانند شبکه سویچینگ بسته‌ها) می‌آید را به فرم سلولهای 48 بایتی ATM در می‌آورد. 5 بایت هدر توسط لایه ATM اضافه خواهد شد.
- وسایل متصل‌کننده می‌توانند سگمنتهای یک شبکه را به هم متصل کنند. همچنین آنها می‌توانند شبکه‌ها را به هم متصل کنند و یک اینترنت را شکل دهند.
- سه نوع وسیله متصل‌کننده وجود دارد: تکرارکننده و هاب، پل و سویچهای دو لایه، روتر و سویچهای سه لایه.
- تکرارکننده، سیگنال را در لایه فیزیکی، دوباره بازسازی می‌کند. هاب، یک تکرارکننده چند پورت است.
- پل، به آدرس ایستگاهها دسترسی دارد و می‌تواند بسته‌ای را در شبکه، فیلتر کند و یا به پیش‌راند. پل در لایه‌های فیزیکی و دیتالینک کار می‌کند. یک سویچ دو لایه، یک پل پیچیده است.

۳-۷ مجموعه تمرین‌ها

تمرین‌ها

۱- فکر می‌کنید چرا یک فریم اترنت، باید حداقل طولی برای دیتا داشته باشد؟

۲- تصور کنید طول کابل 10BASE5 برابر با 2500 متر است. اگر سرعت انتشار سیگنال در یک کابل کواکسیال ضخیم برابر با 200,000,000 m/s باشد، چه مدت طول می‌کشد تا بیتی از ابتدای کابل، به انتهای کابل برسد؟ از هرگونه تأخیر انتشار در تجهیزات، صرف نظر کنید.

۳- با استفاده از راه حل تمرین ۲ ، حداکثر زمانی که طول می‌کشد تا وقوع تصادف احساس گردد را محاسبه نمایید. بدترین حالت زمانی رخ می‌دهد که دیتا، توسط ایستگاهی متصل به یک سرکابل، تولید شود و تصادف در محل ایستگاهی متصل به انتهای دیگر کابل، رخ دهد. به خاطر داشته باشید که برای حس کردن تصادف، سیگنال باید مسیر رفت و برگشت را طی نماید.

۴- نرخ بیت 10BASE5 برابر با 10 Mbps است. چه مدت طول می‌کشد تا کوچکترین فریم ساخته شود؟ نحوه محاسبه را نشان دهید.

۵- با استفاده از نتایج تمرین‌های ۳ و ۴، حداقل طول یک فریم اترنت را بدست آورید به طوری که آشکارسازی تصادف، به درستی انجام شود.

www.Abiri.ir

۶- زیرلایه MAC اترنت، 42 بایت دیتا از زیرلایه LLC دریافت می‌کند. چه تعداد بایت صفر باید به دیتا اضافه کند تا طول فریم اترنت، به حداقل طول مجاز برسد؟ به این عمل، zero padding می‌گویند.

۷- یک زیرلایه MAC اترنت، 1510 بایت دیتا را از زیرلایه LLC دریافت می‌کند. آیا می‌توان این اندازه دیتا را در یک تک فریم اترنت کپسول‌بندی کرد؟ اگر نه، چه تعداد فریم باید ارسال کرد؟ طول دیتا در هر فریم چقدر است؟

۸- CSMA/CD را با CSMA/CA مقایسه کنید. تفاوتها و تشابهات آنها را بیان نمایید.

۹- با استفاده از جدول ۳-۴ ، فیلدهای IEEE 802.3 را با IEEE 802.11 مقایسه نمایید. تشابهات و تفاوتهای آنها را بیان نمایید.

فعالیتهای تحقیقاتی

۱۰- اینترنت معمولی از نسخه‌ای از متد دسترسی CSMA/CD استفاده می‌کند. نام آن CSMA/CD با یک اصرارکننده (CSMA/CD with 1-Persistent) است. اطلاعاتی درباره این متد بدست آورید.

۱۱- نوع دیگری از LAN بی‌سیم که امروزه بکار می‌رود، Bluetooth است. قدری اطلاعات درباره این نوع LANبی‌سیم بدست آورید و بگویید که چگونه می‌توان از آن در اینترنت استفاده کرد.

۱۲-DSL از یک نوع مدولاسیون به نام DMT استفاده می‌کند. قدری اطلاعات درباره این مدولاسیون بدست آورید و بیان نمایید که چطور می‌توان از آن درDSL استفاده کرد.

۱۳-PPP از فازهای مختلفی عبور می‌کند و می‌توان آنها را توسط یک دیاگرام گذر حالت نشان داد. برای یک اتصال PPP ، دیاگرام گذر حالت را ترسیم نمایید.

۱۴- فرمتِ یک بسته LCP که در یک فریم PPP کپسول‌بندی می‌شود را بدست آورید. همه فیلدها، کدهای آنها و هدف از بکارگیری آنها را بیان کنید.

۱۵- فرمت یک بسته NCP که در یک فریم PPP کپسول‌بندی می‌شود را بدست آورید. همه فیلدها، کد آنها و هدف از بکارگیری آنها را بیان کنید.

۱۶-PPP از دو پروتوکل تعیین هویت استفاده می‌کند: CHAP و PAP . اطلاعاتی درباره این دو پروتوکل و اینکه چگونه می‌توان از آنها درPPP استفاده کرد، بدست آورید.

۱۷- فرمت یک بسته PPPoE را بدست آورید. همه فیلدها و هدف از بکارگیری آنها را بیان نمایید.

۱۸- ببینید چگونه می‌توان یک بسته IP را در سلولهای ATM ، با استفاده از AAL5 کپسول‌بندی کرد.

www.Abiri.ir

Fields	*IEEE 802.3 Field Size*	*IEEE 802.11 Field Size*
Destination address		
Source address		
Address 1		
Address 2		
Address 3		
Address 4		
FC		
D/ID		
SC		
PDU length		
Data and padding		
Frame body		
FCS (CRC)		

جدول ۳-۴: تمرین ۹

۱۹- برای جلوگیری از ایجاد لوپها، در شبکه‌ای که از پل شفاف استفاده می‌کند، می‌توان از الگوریتم Spanning Tree استفاده کرد. قدری اطلاعات درباره این الگوریتم بدست آورید و نشان دهید که چگونه این الگوریتم می‌تواند از وقوع لوپ جلوگیری کند.

www.Abiri.ir

فصل چهارم

آدرس‌های IP – آدرس‌دهی کلاس‌دار

برای آنکه مخابره دیتا به صورت ***سرتاسری***[1] مابین وسایل متصل به اینترنت امکان‌پذیر باشد، باید هر وسیله را به طور منحصر به فردی در لایه شبکه (لایه IP) مشخص کنیم. این کار مشابه با سیستم تلفنی است که در آن هر مشترک تلفنی، یک شماره تلفن منحصر به فرد دارد و کد کشور و کد ناحیه، بخشی از الگوی مشخص کردن یک شماره تلفن می‌باشد.

در این فصل، مفهوم کلی آدرسهای IP ، ***آدرس‌دهی کلاس‌دار***[2] و نیز مکانیزمهای آدرس‌دهی به کار رفته در روزهای اول به راه افتادن اینترنت را مورد بحث قرار می‌دهیم. در فصل پنجم، ***آدرس‌دهی بدون‌کلاس***[3] را بررسی می‌کنیم که مکانیزم غالب آدرس‌دهی در اینترنت است.

۴-۱ مقدمه

شاخصی که در لایه IP پشته پروتوکل TCP/IP برای مشخص کردن هر وسیله متصل به اینترنت به کار می‌رود، آدرس IP نامیده می‌شود. یک آدرس IP ، یک آدرس 32 بیتی است که به طور منحصر به فرد و سرتاسری، اتصال وسیله‌ای به اینترنت را مشخص می‌نماید.

مهم: یک آدرس IP، یک آدرس 32 بیتی است.

آدرسهای IP منحصر به فرد هستند. منحصر به فرد بودن آنها به این معنی است که هر آدرس IP فقط و فقط اتصال یک وسیله را به اینترنت نشان می‌دهد. دو وسیله متصل به اینترنت هرگز نمی‌توانند آدرس IP مشابهی داشته باشند. اگر وسیله‌ای از طریق دو شبکه، در دو نقطه به اینترنت متصل شده باشد، دو آدرس IP دارد.

مهم: آدرسهای IP منحصر به فرد هستند.

1- Global
2- Classful addressing
3- Classless addressing

www.Abiri.ir

آدرسهای IP از این بابت جامعیت دارند که هر وسیله‌ای که بخواهد به اینترنت متصل شود، باید این سیستم آدرس‌دهی را بپذیرد.

فضای آدرس

پروتوکلی مانند IP که آدرسها را تعریف می‌کند، یک ***فضای آدرس***[1] دارد. فضای آدرس، کل تعداد آدرسهایی است که پروتوکل از آنها استفاده می‌کند. اگر پروتوکلی از N بیت برای تعریف یک آدرس استفاده کند، فضای آدرس حاوی 2^N آدرس خواهد بود زیرا که هر بیت می‌تواند دو حالت (0 یا 1) داشته باشد و N بیت می‌توانند 2^N حالت داشته باشند. IPv4 از آدرسهای 32 بیتی استفاده می‌کند و لذا فضای آدرس آن حاوی 2^{32} یا 4294967296 (بیش از 4 میلیارد آدرس) می‌باشد. این بدان معناست که از نظر تئوری، می‌توانیم بیش از 4 میلیارد وسیله را به اینترنت وصل کنیم اما کمی بعد خواهیم دید که به دلیل محدودیتهای موجود، تعداد واقعی وسائلی که می‌توان به اینترنت وصل کرد، خیلی کمتر از این تعداد است.

مهم: فضای آدرس IPv4 برابر با 2^{32} و یا 4294967296 است.

نماد

برای نشان دادن یک آدرس IP، سه نماد متداول وجود دارد: ***نماد باینری***[2] ، ***نماد اعداد صحیح نقطه‌دار***[3] و ***نماد هگزادسیمال***[4] .

* نماد باینری

در نماد باینری، آدرس IP به صورت 32 بیت نوشته می‌شود. برای آنکه آدرس، خواناتر شود، یک یا چند فضای خالی بعد از هر هشت بیت (یک octet) گذاشته می‌شود. به هر هشت بیت، یک ***بایت***[5] گفته می‌شود. بنابراین ممکن است که به یک آدرس IP به عنوان یک آدرس 32 bit ، 4 octet و یا 4 byte اشاره شود. مثالی از یک آدرس IP با نماد باینری به صورت زیر است:

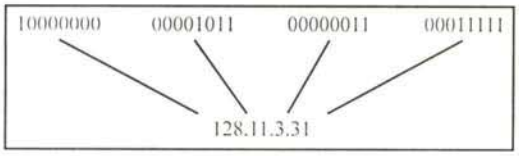
01110101 10010101 00011101 11101010

1- Address Space
2- Binary notation
3- dotted-decimal notation
4- hexadecimal notation
5- byte

www.Abiri.ir

❊ نماد اعداد صحیح نقطه‌دار

برای فشرده‌تر کردن و خواناتر کردن یک آدرس IP ، معمولاً آدرسهای IP را به فرم اعداد صحیحی که توسطه نقطه از هم جدا شده‌اند، نشان می‌دهند. شکل ۴-۱ ،یک آدرس IP را نشان می‌دهد که با نماد اعداد صحیح نقطه‌دار نشان داده شده است. دقت نمایید از آنجا که هر بایت، فقط هشت بیت است، هر عدد در نماد اعداد صحیح نقطه‌دار مابین 0 و 255 است.



شکل ۴-۱: نماد اعداد صحیح نقطه‌دار

مهم: در نماد باینری، آدرس IP به صورت 32 بیت، نوشته می‌شود.

❊❊ مثال ۱

آدرسهای IP زیر را از نماد باینری به نماد اعداد صحیح نقطه‌دار تغییر دهید:

الف – 11101111 00001011 00001011 10000001

ب- 11111111 00011011 10000011 11000001

ج- 01101111 10001011 11011011 11100111

د- 00001111 11111011 10011011 11111001

راه حل: به جای هر هشت بیت، عدد صحیح معادل آن را قرار می‌دهیم و برای جدا ساختن آنها، بین اعداد، نقطه می‌گذاریم.

الف- 129.11.11.239

ب- 193.131.27.255

ج- 231.219.139.111

د- 249.155.251.15

❊❊ مثال ۲

آدرسهای IP زیر را از نماد اعداد صحیح نقطه‌دار به نماد باینری تغییر دهید.

الف – 111.56.45.78

ب- 221.34.7.82

www.Abiri.ir

ج- 241.8.56.12

د- 75.45.34.78

راه حل: به جای هر عدد صحیح، معادل باینری آن را قرار می‌دهیم:

الف – 01001110 00101101 00111000 01101111

ب- 01010010 00000111 00100010 11011101

ج- 00001100 00111000 00001000 11110001

د- 01001110 00100010 00101101 01001011

**** مثال ۳**

اگر خطایی در آدرسهای IP زیر وجود دارد، آن را پیدا کنید:

الف- 111.56.045.78

ب- 221.34.7.8.20

ج- 75.45.301.14

د- 11100010.23.14.67

راه حل: هر مورد را جداگانه بررسی می‌کنیم:

الف- در نماد اعداد صحیح نقطه‌دار، نباید قبل از عدد، صفر بگذاریم (045 صحیح نیست).

ب- آدرس IP بیش از چهار عدد صحیح ندارد.

ج- هر عدد در نماد اعداد صحیح نقطه‌دار، کوچکتر از 256 است و 301 بزرگتر از 256 می‌باشد.

د- مخلوطی از نماد باینری و نماد اعداد صحیح نقطه‌دار، مجاز نیست.

*** نماد هگزادسیمال**

گاهی اوقات یک آدرس IP را با نماد هگزادسیمال می‌نویسیم. هر رقم هگزادسیمال معادل چهار بیت است. معنی آن این است که یک آدرس 32 بیتی حاوی 8 رقم هگزا است. غالباً در برنامه‌نویسی شبکه از این نماد استفاده می‌شود.

**** مثال ۴**

آدرسهای IP زیر را از نماد باینری به نماد هگزادسیمال تغییر دهید.

الف – 11101111 00001011 00001011 10000001

ب- 11111111 00011011 10000011 11000001

www.Abiri.ir

راه حل: به جای هر گروه 4 بیتی، معادل هگزادسیمال آن را قرار می‌دهیم. دقت نمایید که نماد هگزادسیمال معمولاً هیچ نقطه یا فضای خالی اضافه‌ای ندارد. با این وجود، 0X (و یا 0x) در ابتدای عدد و یا زیرنویس 16 در انتهای عدد، معرف هگزادسیمال بودن آن است.

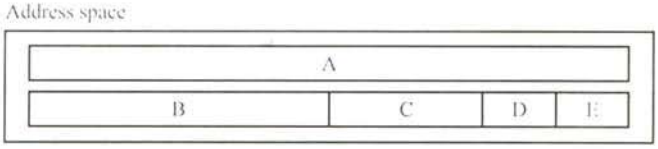
الف- $810B0BEF_{16}$ یا 0x810B0BEF

ب- $C1831BFF_{16}$ یا 0xC1831BFF

۴-۲ آدرس‌دهی کلاس‌دار

وقتی که چند دهه پیش، استفاده از آدرسهای IP آغاز شد، از مفهوم کلاس آدرس استفاده می‌کردند. این معماری، ***آدرس‌دهی کلاس‌دار***[1] نامیده می‌شد. در اواسط دهه ۱۹۹۰، معماری جدیدی به نام ***آدرس‌دهی بدون‌کلاس***[2] معرفی شد و به تدریج جای آدرس‌دهی کلاس‌دار را می‌گیرد به طوری که گرچه هنوز بخشی از اینترنت از آدرس‌دهی کلاس‌دار استفاده می‌کند، حرکت از آدرس‌دهی کلاس‌دار به سمت آدرس‌دهی بدون‌کلاس بسیار سریع است. در این فصل، مفهوم آدرس‌دهی کلاس‌دار را معرفی می‌کنیم و در فصل بعد، آدرس‌دهی کلاس‌دار را مورد بحث قرار خواهیم داد. برای درک مفهوم آدرس‌دهی بدون‌کلاس، درک مفهوم آدرس‌دهی کلاس‌دار الزامی است.

در آدرس‌دهی کلاس‌دار، فضای آدرس IP به پنج کلاس تقسیم می‌شود: A ، B ، C ، D و E. هر کلاس، بخشی از کل فضای آدرس را می‌پوشاند. شکل ۴-۲ به صورت تقریبی نشان می‌دهد که هر کلاس آدرس، چقدر از فضای آدرس را در بر می‌گیرد.



شکل ۴-۲ : مقدار تقریبی از فضای آدرس که هر کلاس در بر می‌گیرد

می‌توان از شکل دید که کلاس A ، نیمی از فضای آدرس را پوشانده است و این یک عیب عمده در طراحی است. کلاس B ، یک چهارم فضای آدرس را می‌پوشاند و این عیب دیگری در طراحی است. کلاس C ، یک هشتم فضای آدرس را می‌پوشاند و کلاسهای D و E

1- Classful addressing
2- Classless addressing

www.Abiri.ir

هر کدام یک شانزدهم فضای آدرس را می‌پوشانند. جدول ۴-۱ تعداد آدرسهای موجود در هر کلاس را نشان می‌دهد.

Class	*Number of Addresses*	*Percentage*
A	2^{31} = 2,147,483,648	50%
B	2^{30} = 1,073,741,824	25%
C	2^{29} = 536,870,912	12.5%
D	2^{28} = 268,435,456	6.25%
E	2^{28} = 268,435,456	6.25%

جدول ۴-۱: تعداد آدرسهای موجود در هر کلاس آدرس

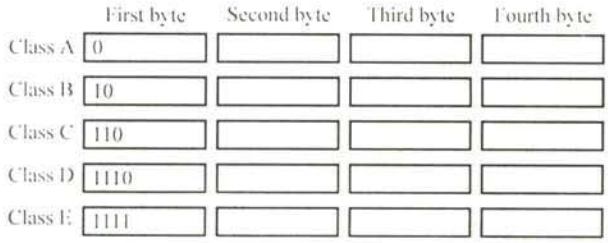
مهم: در آدرس‌دهی کلاس‌دار، فضای آدرس به پنج کلاس تقسیم شده است: A ، B ، C ، D و E .

تشخیص کلاسها

وقتی که آدرس با نماد باینری یا نماد اعداد صحیح نقطه‌دار داده شده باشد، می‌توان کلاس آدرس را پیدا کرد.

❊ پیدا کردن کلاس آدرس با نماد باینری

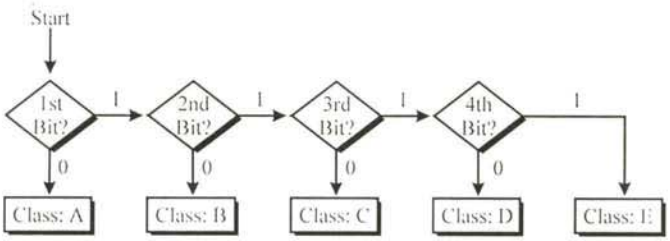
مطابق شکل ۴-۳ وقتی که آدرس با نماد باینری داده شده باشد، اولین بیتهای آن آدرس می‌توانند بلافاصله، کلاس آدرس را به ما بدهند.



شکل ۴-۳: پیدا کردن کلاس آدرس با نماد باینری

www.Abiri.ir

برای پیدا کردن کلاس آدرس می‌توان از روش ارائه شده در شکل ۴-۴ استفاده کرد و با چک کردن سیستماتیک بیتها، کلاس آدرس را پیدا کرد. می‌توان این روش را به راحتی با هر زبانی، برنامه‌نویسی کرد.



شکل ۴-۴: پیدا کردن کلاس آدرس

دقت نمایید که برخی از آدرسهای خاص، در کلاس A یا E قرار می‌گیرند. بر این نکته تأکید می‌نماییم که این آدرسهای خاص، استثناتاتی در کلاس آدرسها هستند.

**** مثال ۵**

چطور می‌توانیم ثابت نماییم که در کلاس A به تعداد 2147483648 آدرس داریم؟

راه حل: در کلاس A ، فقط یک بیت، کلاس را تعریف می‌کند. 31 بیت باقیمانده برای خود آدرس به کار می‌روند. با 31 بیت می‌توانیم 2^{31} یا 2147483648 آدرس داشته باشیم.

**** مثال ۶**

کلاس هر آدرس را پیدا کنید:

الف- **0**0000001 00001011 00001011 11101111

ب- **110**00001 10000011 00011011 11111111

ج- **10**100111 11011011 10001011 01101111

د- **1111**0011 10011011 11111011 00001111

راه حل: روش ارائه شده در شکل ۴-۴ را ببینید.

الف- اولین بیت، 0 است. این یک آدرس کلاس A است.

ب- اولین دو بیت، 1 هستند و سومین بیت، 0 است. این یک آدرس کلاس C است.

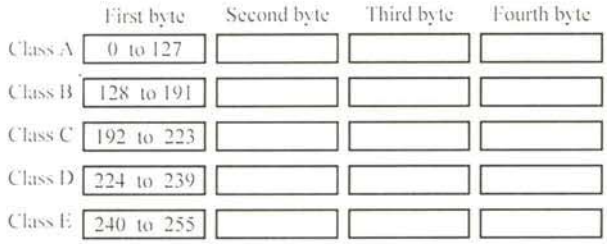
ج- اولین بیت، 1 است. دومین بیت، 0 است. این یک آدرس کلاس B است.

د- اولین چهار بیت، 1 هستند. این یک آدرس کلاس E است.

www.Abiri.ir

❊ پیدا کردن کلاس آدرس با نماد اعداد صحیح نقطه‌دار

وقتی که آدرس با نماد اعداد صحیح نقطه‌دار داده شده باشد، برای پیدا کردن کلاس آدرس، لازم است که فقط به اولین بایت (اولین عدد) نگاه کنیم. اولین بایت هر کلاس آدرس، یک محدوده مشخص از اعداد صحیح را می‌پوشاند. معنای آن این است که اگر اولین بایت آدرس، عددی بین 0 تا 127 باشد، کلاس A است. اگر اولین بایت بین 128 و 191 باشد، کلاس B است و همین طور الی آخر. شکل ۴-۵ این ایده را نشان می‌دهد.



شکل ۴-۵: پیدا کردن کلاس آدرس با نماد اعداد صحیح نقطه‌دار

❊❊ مثال ۷

کلاس هر آدرس را پیدا کنید:

الف- **227**.12.14.87

ب- **193**.14.56.22

ج- **14**.23.120.8

د- **252**.5.15.111

ه - **134**.11.78.56

راه حل: هر مورد را جداگانه بررسی می‌کنیم:

الف- اولین بایت برابر با 227 (بین 224 و 239) است. کلاس D است.

ب- اولین بایت برابر با 193 (بین 192 و 223) است. کلاس C است.

ج- اولین بایت برابر با 14 (بین 0 و 127) است. کلاس A است.

د- اولین بایت برابر با 252 (بین 240 و 255) است. کلاس E است.

هـ - اولین بایت برابر با 134 (بین 128 و 191) است. کلاس B است.

❊❊ مثال ۸

در مثال پنجم نشان دادیم که کلاس A حاوی 2^{31} و یا 2147483648 آدرس است. چطور می‌توانیم با استفاده از نماد اعداد صحیح نقطه‌دار، این مطلب را اثبات کنیم؟

www.Abiri.ir

راه حل: آدرسهای کلاس A در رنج از 0.0.0.0 تا 127.255.255.255 قرار دارند. لازم است نشان دهیم اختلاف بین این دو آدرس برابر با 2147483648 می‌باشد. این تمرین خوبی است زیرا که نشان می‌دهد چطور می‌توانیم تعداد آدرسهای بین دو حد را تعیین نماییم. دقت نمایید که در اینجا با اعدادی در مبنای 256 سروکار داریم. هر بایتی در این نماد، وزنی دارد. وزنها به صورت زیر هستند :

$$256^3 \text{ ، } 256^2 \text{ ، } 256^1 \text{ ، } 256^0$$

حال برای پیدا کردن مقدار عددی متناظر با هر آدرس، کافی است که هر بایت آدرس را در وزن آن ضرب کنیم:

آخرین آدرس = $127x256^3+255x256^2+255x256^1+255x256^0=2147483647$

اولین آدرس = 0

اگر حد پایین را از حد بالا کم کنیم و یک واحد به نتیجه حاصل اضافه نماییم (به خاطر داشته باشید که برای پیدا کردن تعداد آدرسها در یک محدوده، باید یک واحد اضافه کنیم) به عدد 2147483648 و یا 2^{31} می‌رسیم.

شماره شبکه و شماره ماشین

در آدرس‌دهی کلاس‌دار، یک آدرس IP در کلاسهای A ، B و C به ***شماره شبکه***[1] و ***شماره ماشین***[2] تقسیم می‌شود. تعداد بایتهای شماره شبکه و شماره ماشین، بستگی به کلاس آدرس دارد. شکل ۴-۶ بایتهای شماره شبکه و شماره ماشین را نشان می‌دهد. دقت نمایید که کلاسهای D و E بر حسب شماره شبکه و شماره ماشین بیان نمی‌شوند.

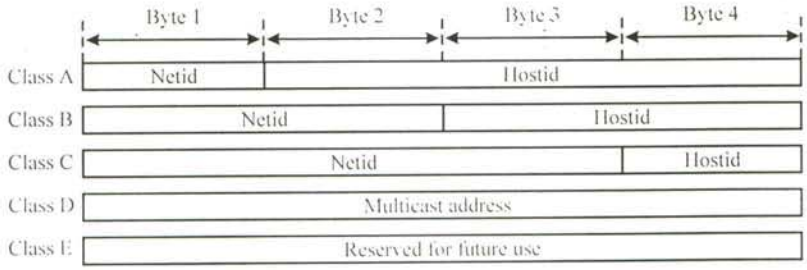
در کلاس A ، شماره شبکه، یک بایتی و شماره ماشین، سه بایتی است.

در کلاس B ، شماره شبکه، دو بایتی و شماره ماشین، دو بایتی است.

در کلاس C ، شماره شبکه، سه بایتی و شماره ماشین، یک بایتی است.

1- Netid
2- Hostid

www.Abiri.ir



شکل ۴-۶ : شماره شبکه و شماره ماشین

کلاسها و بلاکها

یک مسئله در رابطه با آدرس‌دهی کلاس‌دار این است که هر کلاسی به تعداد مشخص و ثابتی از بلاکها تقسیم می‌شود و طول هر بلاک، مقدار ثابتی است. اجازه دهید نگاهی به هر کلاس بیندازیم.

❊ کلاس A

کلاس A به 128 بلاک تقسیم می‌شود و هر بلاک، یک شماره شبکه متفاوت دارد. اولین بلاک، آدرسهای از 0.0.0.0 تا 0.255.255.255 (شماره شبکه 0) را می‌پوشاند. بلاک دوم، آدرسهای از 1.0.0.0 تا 1.255.255.255 (شماره شبکه 1) را می‌پوشاند. آخرین بلاک، آدرسهای از 127.0.0.0 تا 127.255.255.255 (شماره شبکه 127) را می‌پوشاند. دقت نمایید که برای هر بلاکی از آدرسها، اولین بایت (شماره شبکه) ، مقدار ثابتی است اما سه بایت بعدی (شماره ماشین) می‌توانند هر مقداری در محدوده داده شده را داشته باشند.

اولین بلاک (شماره شبکه 0) و آخرین بلاک (شماره شبکه 127) در این کلاس، برای اهداف خاصی رزرو شده‌اند. به علاوه یک بلاک (شماره شبکه 10) برای ***آدرسهای خصوصی***[1] به کار می‌رود. می‌توان 125=3-128 بلاک آدرس باقیمانده را به سازمانها داد. معنی آن این است که فقط 125 سازمان می‌توانند آدرس کلاس A داشته باشند. با این حال، هر بلاک درَ کلاس A حاوی 16777216 آدرس است و معنی آن این است که سازمانی که یک بلاک آدرس کلاس A می‌گیرد، باید واقعاً بزرگ باشد تا بتواند از همه این آدرسها استفاده کند. شکل ۴-۷ بلاکهای آدرس کلاس A را نشان می‌دهد.

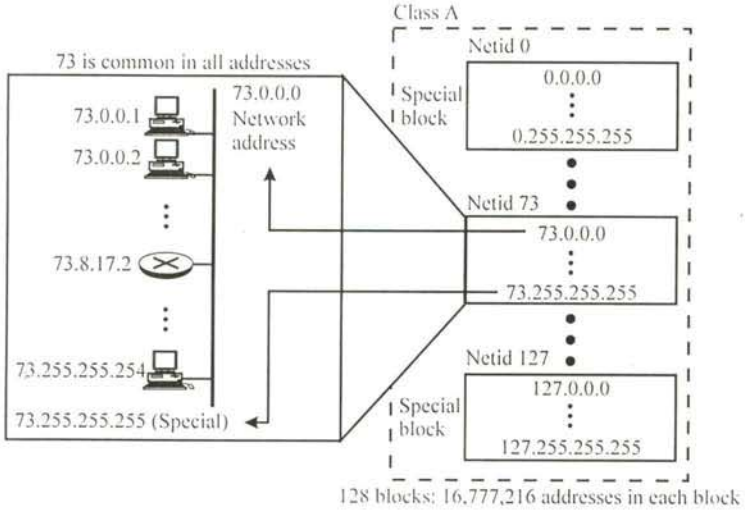
همچنین شکل ۴-۷ نشان می‌دهد سازمانی که یک بلاک آدرس با شماره شبکه 73 به او داده شده است، چگونه از آدرسهایش استفاده می‌کند. اولین آدرس در بلاک آدرس برای

1- Private addresses

www.Abiri.ir

معرفی کردن سازمان به بقیه اینترنت به کار می‌رود. به این آدرس، ***آدرس شبکه***[1] گفته می‌شود. این آدرس، شبکه و نَه تک تک ماشین‌ها را مشخص می‌کند. سازمان مجاز نیست که از آخرین آدرس استفاده کند زیرا که این آدرس برای منظور خاصی رزرو شده است.

آدرسهای کلاس A برای سازمانهای بزرگی که تعداد زیادی کامپیوتر و روتر به شبکه آنها متصل شده است، طراحی شده بود. با این وجود، احتمالاً تعداد آدرسهای موجود در هر بلاک آدرس کلاس A یعنی 16777216 آدرس، بیش از نیاز سازمانها است. بنابراین در این کلاس، بسیاری از آدرسها تلف می‌شوند.



شکل ۴-۷ : بلاکهای آدرس در کلاس A

مهم: میلیونها آدرس کلاس A تلف شده‌اند.

❊ کلاس B

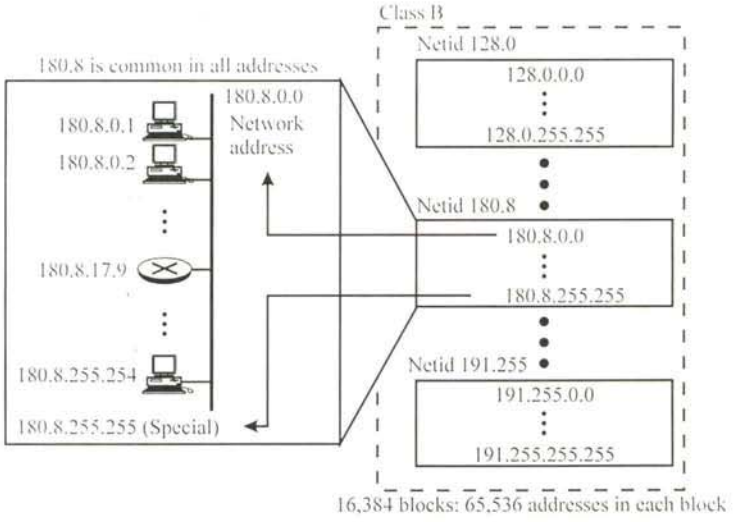
کلاس B به 16384 بلاک تقسیم می‌شود و هر بلاک، یک شماره شبکه متفاوت دارد. 16 بلاک آدرس (از شماره شبکه 172.16 تا شماره شبکه 172.31) برای آدرسهای خصوصی رزرو شده‌اند و لذا 16384-16=16368 بلاک آدرس برای اختصاص دادن به سازمانها باقی می‌ماند. اولین بلاک آدرس، آدرسهای از 128.0.0.0 تا 128.0.255.255 را می‌پوشاند (شماره

1- Network address

www.Abiri.ir

شبکه 128.0). آخرین بلاک، آدرسهای از 191.255.0.0 تا 191.255.255.255 (شماره شبکه 191.255) را می‌پوشاند. دقت نمایید که برای هر بلاک از آدرسها، اولین دو بایت (شماره شبکه) مشابه هستند ولی دو بایت بعدی (شماره ماشین) می‌توانند هر مقداری در محدوده 0 تا 255 را داشته باشند.

16368 بلاک آدرس داریم که می‌توان آنها را به سازمانها اختصاص داد. معنی آن این است که کل تعداد سازمانهایی که می‌توانند آدرس کلاس B داشته باشند برابر با 16368 سازمان است. با این وجود، از آنجا که هر بلاک حاوی 256x256=65536 آدرس است، سازمان باید به اندازه کافی بزرگ باشد تا بتواند از همه این آدرسها استفاده کند. شکل ۴-۸ بلاک آدرسها در کلاس B را نشان می‌دهد.



شکل ۴-۸ : بلاک آدرسها در کلاس B

همچنین شکل ۴-۸ نشان می‌دهد سازمانی که یک بلاک آدرس با شماره شبکه 180.8 را گرفته است، چگونه از آدرسهایش استفاده می‌کند. اولین آدرس در این محدوده، آدرس شبکه است. آخرین آدرس برای منظور خاصی رزرو شده است. آدرسهای کلاس B برای سازمانهای متوسطی که ممکن است چند ده هزار کامپیوتر و یا روتر به شبکه‌هایشان متصل شده باشد، به کار می‌روند. با این وجود، احتمالاً تعداد آدرسهای موجود در هر بلاک، بیش از تعداد آدرس مورد نیاز یک سازمان است. در این کلاس نیز بسیاری از آدرسها تلف می‌شوند.

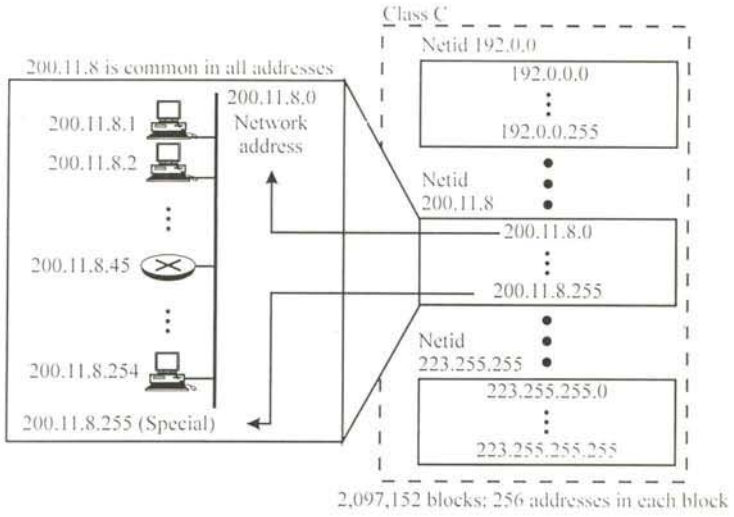
www.Abiri.ir

مهم: بسیاری از آدرسهای کلاس B تلف شده‌اند.

❊ کلاس C

کلاس C به 2097152 بلاک آدرس تقسیم می‌شود و هر بلاک، شماره شبکه متفاوتی دارد. 256 بلاک برای آدرسهای خصوصی به کار می‌روند (از شماره شبکه 192.168.0 تا شماره شبکه 192.168.255) و لذا 2097152-256=2096896 بلاک آدرس برای اختصاص دادن به سازمانها باقی می‌ماند. اولین بلاک، آدرسهای از 192.0.0.0 تا 192.0.0.255 (شماره شبکه 192.0.0) را می‌پوشاند. آخرین بلاک، آدرسهای از 223.255.255.0 تا 223.255.255.255 (شماره شبکه 223.255.255) را می‌پوشاند. دقت نمایید که برای هر بلاکی از آدرسها، اولین 3 بایت (شماره شبکه) مقدار مشابهی است ولی بایت چهارم (شماره ماشین) می‌تواند هر مقداری در محدوده 0 تا 255 را داشته باشد.

2096896 بلاک آدرس وجود دارد که می‌توان آنها را به سازمانها اختصاص داد. معنای آن این است که کل تعداد سازمانهایی که می‌توانند یک آدرس کلاس C داشته باشند برابر با 2096896 سازمان است. با این وجود، تعداد آدرسهای هر بلاک برابر با 256 آدرس است و لذا سازمان باید به اندازه کافی کوچک باشد تا کمتر از 256 آدرس نیاز داشته باشد. شکل ۴-۹ بلاکهای آدرس را در کلاس C نشان می‌دهد.



شکل ۴-۹ : بلاکهای آدرس در کلاس C

www.Abiri.ir

همچنین شکل ۴-۹ نشان می‌دهد سازمانی که به او یک بلاک آدرس با شماره شبکه 200.11.8 داده شده است، چگونه از آدرسهایش استفاده می‌کند. اولین آدرس، آدرس شبکه است. آخرین آدرس، برای منظور خاصی رزرو شده است.

آدرسهای کلاس C برای سازمانهای کوچک طراحی شده‌اند که تعداد کمی کامپیوتر و یا روتر به شبکه آنها متصل است. تعداد آدرسها در هر بلاک آدرس آنقدر محدود است که اکثر سازمانها، بلاکی از این کلاس را نمی‌خواهند.

مهم: تعداد آدرسهای یک بلاک آدرس کلاس C ، کمتر از نیاز اکثر سازمانها است.

* کلاس D

کلاس D فقط یک بلاک آدرس دارد و آدرسهای کلاس D برای ***آدرس‌دهی چندگیرنده***[1] به کار می‌روند. هر آدرس در این کلاس برای تعریف یک گروه از ماشینها در اینترنت به کار می‌رود. وقتی به گروهی، یک آدرس از این کلاس داده می‌شود، هر ماشینی که عضوی از این گروه باشد، علاوه بر ***آدرس تک‌گیرنده***[2] خود، یک ***آدرس چندگیرنده***[3] نیز خواهد داشت.

مهم: آدرسهای کلاس D برای آدرس‌دهی چندگیرنده به کار می‌روند. این کلاس فقط یک بلاک آدرس دارد.

* کلاس E

کلاس E فقط یک بلاک آدرس دارد. آدرسهای این کلاس برای اهداف آتی رزرو شده‌اند.

مهم: آدرسهای کلاس E برای اهداف آتی رزرو شده‌اند. اکثر آدرسهای این کلاس، تلف شده‌اند.

1- Multicast addressing
2- Unicast address
3- Multicast address

www.Abiri.ir

آدرس شبکه

آدرس شبکه[1] ، نقش مهمی را در آدرس‌دهی کلاس‌دار بازی می‌کند. یک آدرس شبکه، چندین خاصیت دارد:

۱- آدرس شبکه، اولین آدرس در بلاک است.

۲- آدرس شبکه، شبکه را به بقیه اینترنت معرفی می‌کند. در فصلهای آتی، یاد می‌گیریم که روترها یک بسته را بر اساس آدرس شبکه، مسیردهی می‌کنند.

۳- با داشتن آدرس شبکه، می‌توان کلاس آدرس، بلاک آدرس و محدوده آدرسهای بلاک را تعیین کرد.

مهم: در آدرس‌دهی کلاس‌دار، آدرس شبکه (اولین آدرس در بلاک آدرس) آدرسی است که به سازمان داده می‌شود زیرا که محدوده آدرسهای بلاک مربوط به آن آدرس شبکه را می‌توان از آدرس شبکه تشخیص داد.

*** مثال ۹**

با داشتن آدرس شبکه 17.0.0.0 ، کلاس، بلاک آدرس و محدوده آدرسها را پیدا کنید.

راه حل: کلاس A است زیرا اولین بایت بین 0 و 127 است. شماره شبکه بلاک آدرس برابر با 17 است. محدوده آدرسهای بلاک از 17.0.0.0 تا 17.255.255.255 است.

*** مثال ۱۰**

با داشتن آدرس شبکه 132.21.0.0 ، کلاس، بلاک آدرس و محدوده آدرسها را پیدا کنید.

راه حل: کلاس B است زیرا که اولین بایت بین 128 و 191 است. شماره شبکه بلاک آدرس برابر با 132.21 است. محدوده آدرسهای بلاک از 132.21.0.0 تا 132.21.255.255 است.

*** مثال ۱۱**

با داشتن آدرس شبکه 220.34.76.0 کلاس، بلاک و محدوده آدرسها را پیدا کنید.

راه حل: کلاس C است به دلیل آنکه اولین بایت بین 192 و 223 است. شماره شبکه بلاک برابر با 220.34.76 است. محدوده آدرسها از 220.34.76.0 تا 220.34.76.255 است.

1- Network address

www.Abiri.ir

اطلاعات کافی

خواننده ممکن است دقت کرده باشد که در آدرس‌دهی کلاس‌دار، آدرس شبکه اطلاعات کافی درباره شبکه می‌دهد. با داشتن آدرس شبکه، می‌توانیم تعداد آدرسهای بلاک آدرس را بدست آوریم. دلیل آن این است که تعداد آدرسها در بلاک آدرس، از قبل تعیین شده است. تعداد آدرس هر بلاک آدرس کلاس A ، یکسان و برابر با 256x256x256=16777216 آدرس است. تعداد آدرس هر بلاک آدرس کلاس B ، یکسان و برابر با 256x256=65536 آدرس است. تعداد آدرس هر بلاک آدرس کلاس C ، یکسان و برابر با 256 آدرس است.

مَسک

در بخش قبل گفتیم که با داشتن آدرس شبکه، می‌توانیم بلاک آدرس و تعداد آدرسهای موجود در آن را بدست آوریم. معکوس آن چطور؟ اگر آدرسی را داشته باشیم، آیا می‌توانیم آدرس شبکه (اولین آدرس در بلاک آدرس) را بدست آوریم؟ این مطلب مهم است به دلیل آنکه روتر برای مسیردهی یک بسته به سمت شبکه مقصد آن بسته، باید آدرس شبکه مقصد را از آدرس مقصد بسته که در هدر آن نوشته شده است، بدست آورد.

یک راه برای بدست آوردن آدرس شبکه این است که ابتدا کلاس آدرس و شماره شبکه را بدست آوریم و سپس شماره ماشین را برابر با صفر قرار دهیم تا آدرس شبکه را پیدا کنیم. به عنوان مثال، اگر آدرس 134.45.78.2 داده شده باشد، فوراً می‌توانیم بگوییم که این آدرس متعلق به کلاس B است. شماره شبکه آن برابر با 134.45 (دو بایت اول) و آدرس شبکه آن 134.45.0.0 است.

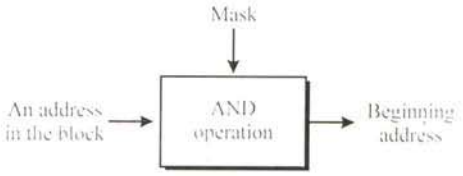
روش فوق زمانی صحیح است که زیرشبکه‌سازی نکرده باشیم. روش کلی که همیشه می‌توانیم برای پیدا کردن آدرس شبکه از روی یک آدرس داده شده به کار ببریم، استفاده از **مَسک**[1] است.

* مفهوم مسک

مسک، یک عدد 32 بیتی است که وقتی با آدرسی در بلاک، بیت به بیت، AND می‌شود، اولین آدرس بلاک که همان آدرس شبکه است را می‌دهد. شکل ۴-۱۰ مفهوم استفاده از مسک را نشان می‌دهد.

1- Mask

www.Abiri.ir



شکل ۴-۱۰ : مفهوم اعمال مسک به آدرس

❊ عملگر AND

در اعمال مسک به یک آدرس، از عملگر AND استفاده می‌کنیم. این عملگر، بیت به بیت، به آدرس و مسک اعمال می‌شود. برای کار ما، کافی است بدانیم که عملگر AND به صورت زیر عمل می‌کند:

۱- اگر بیتی در مسک برابر با 1 باشد، بیت متناظر با آن در آدرس، در خروجی باقی می‌ماند (تغییر نمی‌کند).

۲- اگر بیتی در مسک برابر با 0 باشد، بیت متناظر با آن در آدرس، در خروجی به 0 تبدیل می‌شود.

به عبارت دیگر، بیتهایی از آدرس که با 1 های مسک، AND می‌شوند، به همان مقدار اولیه خود باقی می‌مانند و بیتهایی که با 0 های مسک، AND می‌شوند، به 0 تبدیل می‌شوند. شکل ۴-۱۱ این دو حالت را نشان می‌دهد.



شکل ۴-۱۱ : عملگر AND

❊ مسکهای پیش‌فرض

در عملگر AND برای آدرس‌دهی کلاس‌دار، از سه **مسک پیش‌فرض**[1] ، یکی برای هر کلاس آدرس استفاده می‌شود. جدول ۴-۲، مسک پیش‌فرض را برای هر کلاس آدرس نشان می‌دهد. برای کلاس A ، مسک از هشت 1 و بیست و چهار 0 ساخته شده است. برای کلاس B

1- Default mask

www.Abiri.ir

مسک از شانزده 1 و شانزده 0 ساخته شده است. برای کلاس C ، مسک از بیست و چهار 1 و هشت 0 ساخته شده است. 1 ها شماره شبکه را حفظ می‌کنند و 0 ها شماره ماشین را برابر با 0 قرار می‌دهند. به خاطر داشته باشید که آدرس شبکه در هر کلاس از قرار دادن شماره شبکه در کنار شماره ماشینی برابر با تمام صفر به دست می‌آید. جدول ۴-۲ مسک پیش‌فرض را برای هر کلاس نشان می‌دهد.

Class	*Mask in binary*	*Mask in dotted-decimal*
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

جدول ۴-۲ : مسکهای پیش‌فرض

دقت کنید که تعداد 1 های مسک هر کلاس آدرس، مطابق با تعداد بیتهای شماره شبکه و تعداد 0ها مطابق با تعداد بیتهای شماره ماشین در آن کلاس آدرس است. به عبارت دیگر، وقتی که مسک با یک آدرس AND می‌شود، شماره شبکه، بدون تغییر نگه داشته می‌شود ولی شماره ماشین برابر با تمام صفر می‌شود.

مهم: آدرس شبکه، اولین آدرس یک بلاک آدرس است. می‌توان آن را با اِعمال مسک پیش‌فرض به یکی از آدرسهای بلاک آدرس، بدست آورد. این عمل، شماره شبکه را نگه می‌دارد و شماره ماشین را به تمام صفر تبدیل می‌کند.

* اِعمال مسکها

اِعمال مسک به شبکه‌ای که زیرشبکه‌سازی نشده باشد، ساده است. بدون نیاز به اِعمال عملگر AND به هر بیت، می‌توان با استفاده از دو قاعده ساده، آدرس شبکه را پیدا کرد:

۱- اگر بایتی در مسک برابر با 255 باشد، بایت متناظر با آن را در آدرس نگه دارید.

۲- اگر بایتی در مسک برابر با 0 باشد، بایت متناظر با آن در آدرس را برابر با 0 قرار دهید.

** مثال ۱۲

با داشتن آدرس 23.56.7.91 آدرس شبکه را بدست آورید.

www.Abiri.ir

راه حل: مسک پیش‌فرض کلاس A برابر با 255.0.0.0 می‌باشد و با اِعمال این مسک به آدرس داده شده، فقط بایت اول آن باقی می‌ماند و سه بایت دیگر به 0 تبدیل می‌شوند و لذا آدرس شبکه برابر با 23.0.0.0 خواهد شد.

**** مثال ۱۳**

با داشتن آدرس 132.6.17.85 آدرس شبکه را بدست آورید.

راه حل: مسک پیش‌فرض کلاس B برابر با 255.255.0.0 می‌باشد و با اعمال این مسک به آدرس داده شده، فقط دو بایت اول آن باقی می‌مانند و دو بایت بعدی به 0 تبدیل می‌شوند و لذا آدرس شبکه برابر با 132.6.0.0 می‌شود.

**** مثال ۱۴**

با داشتن آدرس 201.180.56.5 آدرس شبکه را بدست آورید.

راه حل: مسک پیش‌فرض کلاس C برابر با 255.255.255.0 می‌باشد و با اِعمال این مسک به آدرس داده شده، سه بایت اول باقی می‌مانند و بایت آخر به 0 تبدیل می‌شود و لذا آدرس شبکه برابر با 201.180.56.0 می‌شود.

مهم: دقت نمایید که نباید مسک پیش‌فرض یک کلاس را به آدرسی از کلاس دیگری اِعمال کرد.

نماد CIDR

با وجود آنکه در آدرس‌دهی کلاس‌دار، هر آدرس حاوی یک مسک پیش‌فرض است، گاهی اوقات مناسب است (و ضمناً با آدرس‌دهی بدون‌کلاس که در فصل بعد خواهد آمد، مطابقت دارد) که به طور صریح، مسک را بیان کنیم. نماد ***CIDR***[1] به همین منظور به کار می‌رود. در این نماد، عددی برابر با تعداد 1های مسک، بعد از یک / (slash) در انتهای آدرس نوشته می‌شود. به عنوان مثال، آدرس 18.46.74.10 که یک آدرس کلاس A با مسک پیش‌فرض 255.0.0.0 است، به صورت 18.46.74.10/8 نوشته می‌شود تا نشان دهد که مسک، هشت 1 دارد. به طور مشابه، آدرس 141.24.74.69 به صورت 141.24.74.69/16 نوشته می‌شود تا نشان دهد که یک آدرس کلاس B است و مسک حاوی شانزده 1 است. به همین ترتیب آدرس 200.14.70.22 به صورت 200.14.70.22 / 24 نوشته می‌شود. در فصل پنجم خواهیم دید که CIDR علی‌الخصوص در آدرس‌دهی بدون‌کلاس مفید است.

1- Classless InterDomain Routing

www.Abiri.ir

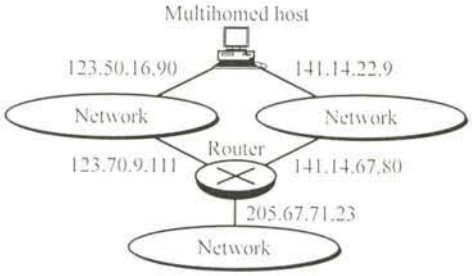
۲ سایر مطالب درباره آدرس‌دهی

ل چند آدرسه

وسیله در شبکه

www.Abiri.ir

پیامدهای آن این است که با جابه‌جایی کامپیوتر از یک شبکه به شبکه دیگر، آدرس IP آن کامپیوتر نیز باید عوض شود.



شکل ۴-۱۲ : وسایل چند آدرسه

آدرسهای خاص

بعضی از قسمتهای فضای آدرس برای آدرسهای خاص به کار می‌روند (جدول ۴-۳ را ببینید).

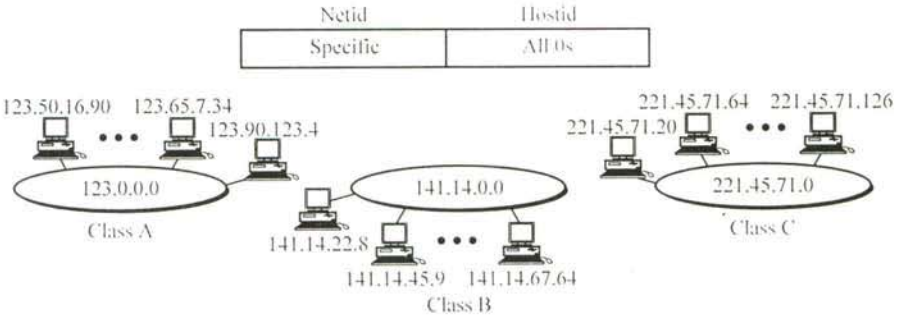
Special Address	*Netid*	*Hostid*	*Source or Destination*
Network address	Specific	All 0s	None
Direct broadcast address	Specific	All 1s	Destination
Limited broadcast address	All 1s	All 1s	Destination
This host on this network	All 0s	All 0s	Source
Specific host on this network	All 0s	Specific	Destination
Loopback address	127	Any	Destination

جدول ۴-۳ : آدرسهای خاص

❊ آدرس شبکه

هم‌اکنون درباره موضوع آدرسهای شبکه صحبت کردیم. اولین آدرس در یک بلاک آدرس (کلاسهای A ، B و C)، آدرس شبکه را تعریف می‌کند. شکل ۴-۱۳ سه مثال از آدرس شبکه، یکی برای هر کلاس آدرس را نشان می‌دهد.

www.Abiri.ir



شکل ۴-۱۳ : آدرس شبکه

❊ آدرس همگانی صریح

در کلاسهای A ، B و C اگر شماره ماشین تمام 1 باشد، به آن ***آدرس همگانی صریح***[1] گفته می‌شود. به این دلیل به آن « صریح » گفته می‌شود که صراحتاً شماره شبکه مورد نظر در این آدرس، قید شده است و البته لازم به ذکر است که شماره ماشین در این آدرس، تمام 1 می‌باشد. برای آنکه بسته‌ای برای کلیه ماشینهای داخل یک شبکه مشخص ارسال شود، از این آدرس استفاده می‌شود. همه ماشینها، بسته‌ای با این نوع آدرس مقصد را می‌پذیرند. دقت نمایید که این آدرس، فقط به صورت آدرس مقصد در یک بسته IP قابل استفاده است. همچنین توجه نمایید که این آدرس خاص، از تعداد شماره ماشینهای قابل تخصیص در یک شبکه، یکی کم می‌کند. در شکل ۴-۱۴ روتر یک بسته IP را به آدرس مقصدی که شماره ماشین آن تمام 1 است، ارسال می‌کند. همه وسایل روی این شبکه، بسته IP را دریافت و پردازش می‌کنند.

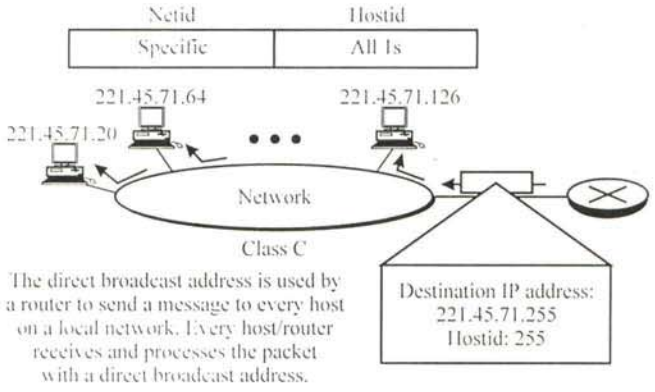
❊ آدرس همگانی محدود

در کلاسهای A ، B و C آدرسی که حاوی تمام 1 است (255.255.255.255)، یک آدرس همگانی برای شبکه می‌باشد و به آن ***آدرس همگانی محدود***[2] گفته می‌شود. به این دلیل به آن « محدود » گفته می‌شود که این آدرس همگانی، فقط محدود به همان شبکه‌ای است که وسیله در آن، قرار گرفته است زیرا روتر اجازه عبور این آدرس و انتقال آن به شبکه دیگری را نمی‌دهد. ماشینی که می‌خواهد بسته‌ای را برای هر ماشین دیگری که در همان

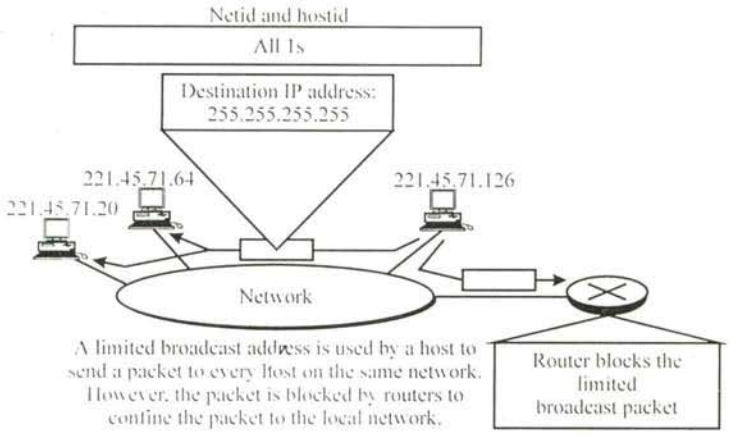
1- Direct broadcast address
2- Limited broadcast address

www.Abiri.ir

شبکه ماشین قرار دارد، ارسال کند، از این آدرس به عنوان آدرس مقصد بسته IP استفاده می‌کند. با این حال همانطور که توضیح دادیم، وقتی که روتر چنین بسته‌ای را دریافت می‌کند، آن را عبور نمی‌دهد تا عمل پخش همگانی این آدرس، فقط به همان شبکه‌ای که ماشین در آن واقع است، محدود شود. دقت نمایید که این آدرس، یک آدرس از کلاس E است. در شکل ۴-۱۵ ماشینی یک بسته IP به آدرس مقصد 255.255.255.255 ارسال می‌کند. همه وسایل متصل به این شبکه، بسته ارسالی را دریافت می‌کنند و آن را پردازش می‌نمایند.



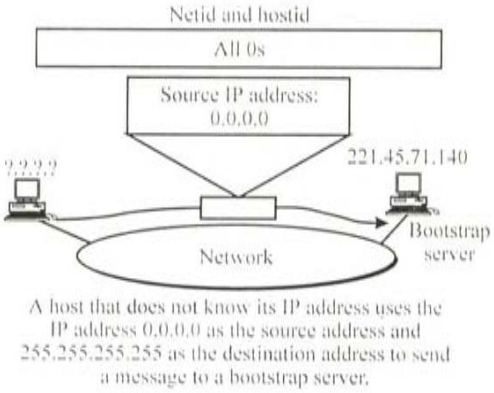
شکل ۴-۱۴ : مثالی از آدرس همگانی صریح



شکل ۴-۱۵ : مثالی از آدرس همگانی محدود

www.Abiri.ir

اگر آدرس IP از تمام 0 ساخته شده باشد، به معنی ***این ماشین روی این شبکه*** . وقتی که یک ماشین به هنگام بوت شدن، آدرس IP خود را نمی‌داند، از این آدرس اده می‌کند. ماشین به هنگام بوت شدن، یک بسته IP برای یک سرور bootstrap (ی گفته می‌شود که به هنگام بوت شدن کامپیوترها، به آنها آدرس IP و سایر اطلاعات را می‌دهد) می‌فرستد که آدرس مبدأ آن برابر با همین آدرس یعنی 0.0.0.0 و آدرس د آن برابر با آدرس همگانی محدود یعنی 255.255.255.255 است و به این ترتیب سعی ند از سرور bootstrap یک آدرس IP بگیرد. دقت نمایید که 0.0.0.0 فقط به عنوان آدرس به کار می‌رود. همچنین توجه نمایید که این آدرس، یک آدرس کلاس A است و لذا تعداد های آدرس کلاس A را یکی کم می‌کند (شکل ۴-۱۶ را ببینید).

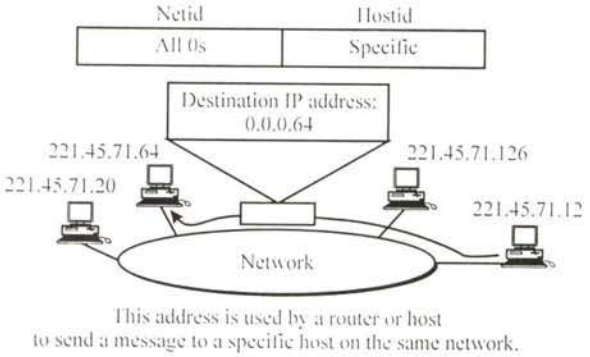


شکل ۴-۱۶: مثالی از آدرس این ماشین روی این شبکه

❋ یک ماشین مشخص روی این شبکه

یک آدرس IP با شماره شبکه تمام 0 به معنی ***یک ماشین مشخص روی این شبکه*** . وقتی که یک ماشین روی این شبکه بخواهد بسته‌ای را برای ماشین دیگری روی همین ه ارسال کند، از این آدرس استفاده می‌کند. از آنجا که روتر جلو عبور بسته‌ای با این آدرس د را می‌گیرد، به این ترتیب می‌توان بسته را به همان شبکه‌ای که ماشین در آن است ود کرد. دقت نمایید که این آدرس فقط به عنوان آدرس مقصد قابل استفاده است

www.Abiri.ir



شکل ۴-۱۷ : مثالی از یک ماشین مشخص روی این شبکه

❊ آدرس برگردان

یک آدرس IP که اولین بایت آن 127 است، به عنوان آدرس ***برگردان***[1] به کار می‌رود و از این آدرس برای تست نرم‌افزار استفاده می‌شود. وقتی که این آدرس به کار می‌رود، بسته IP مربوطه هرگز ماشین ارسال‌کننده بسته را ترک نمی‌کند و به خود نرم‌افزار ارسال‌کننده بسته برمی‌گردد و لذا از آن برای تست نرم‌افزاری که بسته IP را ارسال کرده است، استفاده می‌شود. به عنوان مثال، یک برنامه کاربردی مانند ping می‌تواند بسته‌ای به آدرس مقصد برگردان را ارسال کند تا ببیند که آیا نرم‌افزار IP کامپیوتر، توانایی ارسال و دریافت بسته‌های IP را دارد یا نه. به عنوان یک مثال دیگر، یک برنامه client می‌تواند برای سرور مورد نظر خود که روی همان کامپیوتر نصب شده است، از طریق آدرس برگردان بسته‌هایی را ارسال کند. دقت نمایید که آدرس برگردان فقط به عنوان آدرس مقصد بسته IP به کار می‌رود. همچنین توجه نمایید که آدرس برگردان، یک آدرس کلاس A است و لذا تعداد بلاک آدرسها در کلاس A را یکی کم می‌کند (شکل ۴-۱۸ را ببینید).

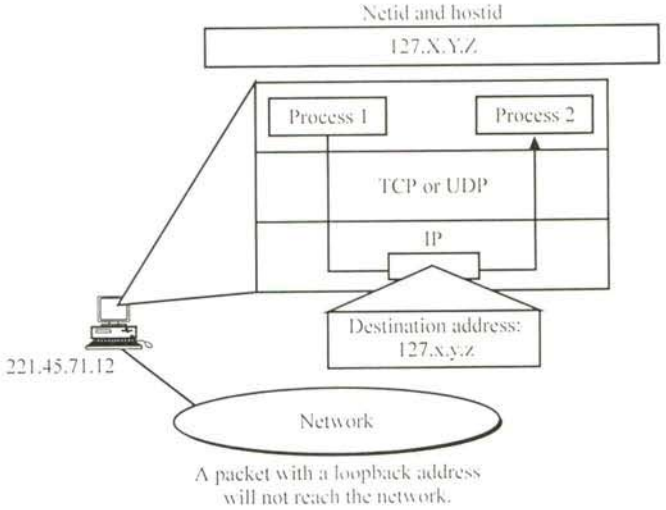
آدرسهای خصوصی

تعدادی از بلاک آدرسها در هر کلاس، به عنوان ***آدرسهای خصوصی***[2] به کار می‌روند به این معنی که در اینترنت قابل استفاده نیستند. این بلاک آدرسها در جدول ۴-۴ نشان داده شده‌اند. یا از این آدرسها در شبکه‌های جدا از اینترنت استفاده می‌شود و یا اگر قرار است

1- Loopback
2- Private addresses

www.Abiri.ir

شبکه‌ای از این آدرسها استفاده کند و به اینترنت متصل شود، باید عمل ***ترجمه آدرس***[1] را انجام دهد که در فصل بیست و ششم درباره آن صحبت خواهیم کرد.



شکل ۴-۱۸ : مثالی از آدرس برگردان

Class	*Netids*	*Blocks*
A	10	1
B	172.16 to 172.31	16
C	192.168.0 to 192.168.255	256

جدول ۴-۴ : آدرسهای خصوصی

آدرسهای تک‌گیرنده، چندگیرنده و همگانی

می‌توان با استفاده از آدرسهای ***تک‌گیرنده***[2]، ***چندگیرنده***[3] و ***همگانی***[4] در اینترنت به مخابره دیتا پرداخت.

1- Address translation
2- Unicast
3- Multicast
4- Broadcast

www.Abiri.ir

❊ آدرسهای تک‌گیرنده

آدرس تک‌گیرنده، یک مخابره ***یک به یک***[1] را فراهم می‌نماید. وقتی که بسته‌ای از یک مبدأ واحد برای یک مقصد واحد ارسال می‌شود، یک مخابره تک‌گیرنده صورت گرفته است. همه سیستم‌هایی که مستقیماً به اینترنت متصل شده‌اند، حداقل یک آدرس تک‌گیرنده منحصر به فرد دارند. آدرسهای تک‌گیرنده متعلق به کلاسهای A ، B و C هستند.

❊ آدرسهای چندگیرنده

آدرسهای چندگیرنده، یک مخابره ***یک به چند***[2] را فراهم می‌نمایند. وقتی که بسته‌ای از یک مبدأ واحد برای گروهی از وسایل ارسال می‌شود، یک مخابره چندگیرنده صورت گرفته است. یک آدرس چندگیرنده، یک آدرس کلاس D است. هر آدرس چندگیرنده، یک شماره را برای یک گروه از وسایل به نام ***شماره گروه***[3] تعریف می‌کند. یک وسیله متصل به اینترنت می‌تواند علاوه بر آدرس تک‌گیرنده خود، یک یا چند آدرس چندگیرنده نیز داشته باشد. اگر یک سیستم (معمولاً یک کامپیوتر) ، هفت آدرس چندگیرنده داشته باشد، به این معنی است که آن سیستم، متعلق به هفت گروه مختلف است. دقت نمایید که یک آدرس کلاس D را می‌توان فقط به عنوان آدرس مقصد به کار برد. چندگیرندگی در اینترنت می‌تواند در سطح ***محلی***[4] و یا ***سرتاسری***[5] باشد. در سطح محلی، ماشینهای روی یک LAN می‌توانند تشکیل یک گروه بدهند و به این گروه، یک آدرس چندگیرنده داده می‌شود. در سطح سرتاسری، ماشینهای روی شبکه‌های مختلف می‌توانند تشکیل یک گروه بدهند و یک آدرس چندگیرنده به آنها داده می‌شود.

مهم: در فصل دهم و پانزدهم به طور کامل درباره چندگیرندگی صحبت خواهیم کرد.

❊❊ آدرسهای چندگیرنده تخصیص یافته

مقامهای مسئول اینترنت، چند آدرس چندگیرنده را مشخص کرده‌اند که برای گروههای خاص به کار می‌روند. در اینجا به دو تا از آنها اشاره می‌کنیم.

1- One to one
2- One to many
3- Groupid
4- Local
5- Global

www.Abiri.ir

۱- ***یک رده کاری***[1] : بعضی از آدرسهای چندگیرنده، به کاربردهای خاصی اختصاص داده شده‌اند. این آدرسهای چندگیرنده با پیشوند 224.0.0 شروع می‌شوند. جدول ۴-۵ بعضی از این آدرسها را نشان می‌دهد.

Address	*Group*
224.0.0.0	Reserved
224.0.0.1	All SYSTEMS on this SUBNET
224.0.0.2	All ROUTERS on this SUBNET
224.0.0.4	DVMRP ROUTERS
224.0.0.5	OSPFIGP All ROUTERS
224.0.0.6	OSPFIGP Designated ROUTERS
224.0.0.7	ST Routers
224.0.0.8	ST Hosts
224.0.0.9	RIP2 Routers
224.0.0.10	IGRP Routers
224.0.0.11	Mobile-Agents

جدول ۴-۵ : آدرسهای رده‌های مختلف کاری

۲- کنفرانس : بعضی از آدرسهای چندگیرنده برای ***کنفرانسها***[2] و ***کنفرانس از راه دور***[3] تعریف شده‌اند. این آدرسهای چندگیرنده با پیشوند 224.0.1 شروع می‌شوند. جدول ۴-۶ بعضی از این آدرسها را نشان می‌دهد.

❊ آدرسهای همگانی

آدرسهای همگانی یک مخابره ***یک به همه***[4] را فراهم می‌کنند. اینترنت، ارسال همگانی بسته‌ها را فقط در سطح محلی امکان‌پذیر کرده است. تا اینجا راجع به دو آدرس همگانی صحبت کرده‌ایم که در سطح محلی به کار می‌روند: آدرس همگانی محدود (255.255.255.255) و آدرس همگانی صریح (شماره شبکه : یک مقدار مشخص ، شماره ماشین : تمام ۱).

1- Category
2- Conferencing
3- Teleconferencing
4- One to all

www.Abiri.ir

در سطح سرتاسری نمی‌توان از ارسال همگانی استفاده کرد. معنی آن این است که یک کامپیوتر و یا یک روتر نمی‌تواند بسته‌ای را برای همه ماشینها و روترهای اینترنت ارسال کند. می‌توانید تصور کنید که اگر این محدودیت وجود نداشت، چه ترافیک زیادی رخ می‌داد.

Address	*Group*
224.0.1.7	AUDIONEWS
224.0.1.10	IETF-1-LOW-AUDIO
224.0.1.11	IETF-1-AUDIO
224.0.1.12	IETF-1-VIDEO
224.0.1.13	IETF-2-LOW-AUDIO
224.0.1.14	IETF-2-AUDIO
224.0.1.15	IETF-2-VIDEO
224.0.1.16	MUSIC-SERVICE
224.0.1.17	SEANET-TELEMETRY
224.0.1.18	SEANET-IMAGE

جدول ۴-۶ : آدرسهای چندکیرنده برای کنفرانسها

مثال ۱۵

شکل ۴-۱۹ بخشی از یک اینترنت با پنج شبکه را نشان می‌دهد.

۱-یک LAN با آدرس شبکه 220.3.6.0 (کلاس C)

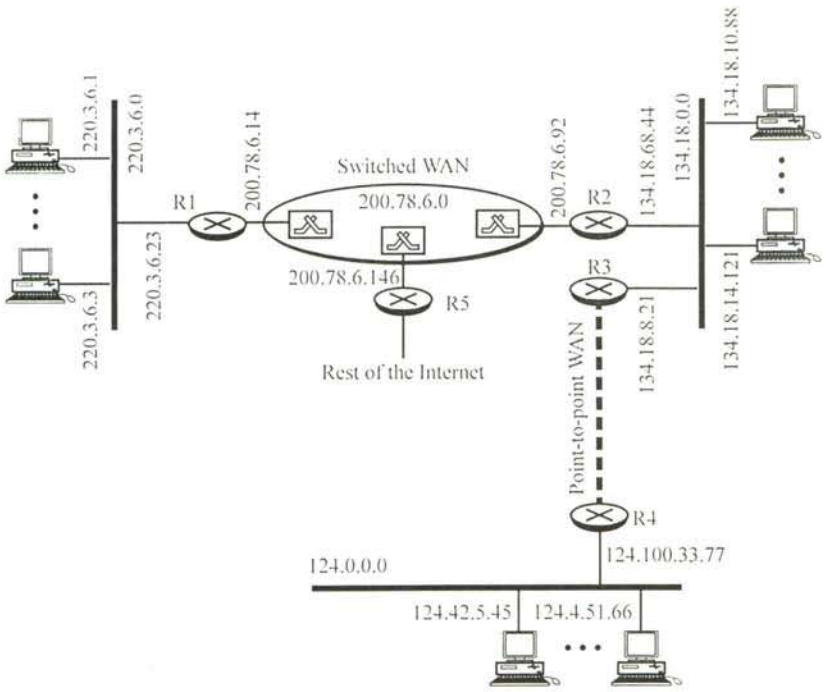
۲-یک LAN با آدرس شبکه 134.18.0.0 (کلاس B)

۳-یک LAN با آدرس شبکه 124.0.0.0 (کلاس A)

۴-یک WAN نقطه به نقطه (خط چین). این شبکه (به عنوان یک خط T-1) صرفاً دو روتر را به هم وصل می‌کند. هیچ ماشینی بین آنها نیست. در این حالت، برای صرفه‌جویی در آدرسهای IP ، هیچ آدرس شبکه‌ای به این نوع از WAN داده نشده است.

۵-یک WAN سویچ شده (مانند Frame Relay یا ATM) که می‌تواند به چندین روتر متصل شود. ما سه تا را نشان داده‌ایم. روتر R1، WAN را به LAN سمت چپ وصل می‌کند. روتر R2، WAN را به LAN سمت راست وصل می‌کند. روتر R5، WAN را به بقیه اینترنت وصل می‌کند.

www.Abiri.ir



شکل ۴-۱۹: اینترنت نمونه

۴-۴ زیرشبکه‌سازی و سوپرشبکه‌سازی

در بخش قبل درباره مشکلات آدرس‌دهی کلاس‌دار صحبت کردیم. به ویژه اشاره کردیم که آدرسهای موجود برای اختصاص دادن به شبکه‌ها در حال اتمام هستند. به دلیل درخواستهای پیاپی از سوی سازمانهای متقاضی آدرس IP، مشکل کمبود آدرسهای IP دو چندان شده است. در این بخش به اختصار درباره دو راه حل صحبت می‌کنیم: ***زیرشبکه‌سازی***[1] و ***سوپرشبکه‌سازی***[2].

زیرشبکه‌سازی

در زیرشبکه‌سازی، یک شبکه به چندین زیرشبکه کوچکتر تقسیم می‌شود به طوری که هر زیرشبکه، آدرس زیرشبکه خودش را دارد.

1- Subnetting
2- Supernetting

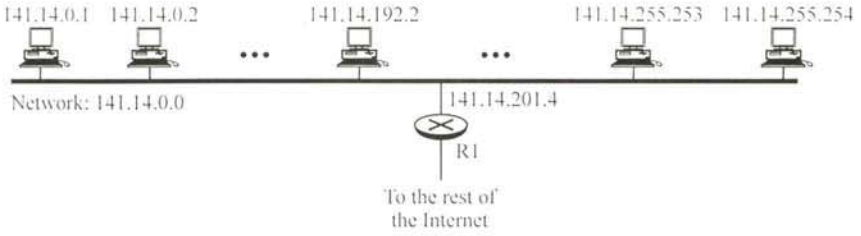
www.Abiri.ir

* طبقه‌بندی دو سطحی

همانطور که قبلاً یاد گرفتیم، یک آدرس IP حاوی 32 بیت است. بخشی از آدرس، شبکه (یا شماره شبکه) را نشان می‌دهد و بخش دیگری از آدرس، ماشین (یا شماره ماشین) را در آن شبکه نشان می‌دهد. معنی آن این است که یک نوع طبقه‌بندی در آدرس‌دهی IP وجود دارد. برای رسیدن به ماشینی در اینترنت، ابتدا باید با استفاده از قسمت اول آدرس (شماره شبکه) به شبکه‌ای برسیم که آن ماشین در آن شبکه واقع شده است. سپس با استفاده از قسمت دوم آدرس (شماره ماشین)، به خود ماشین برسیم. به عبارت دیگر، آدرسهای IP حاوی یک طبقه‌بندی دو سطحی هستند.

مهم: آدرسهای IP با یک طبقه‌بندی دو سطحی طراحی شده‌اند.

با این وجود، این طبقه‌بندی دو سطحی در بسیاری از موارد کافی نیست. به عنوان مثال سازمانی را با آدرس شبکه 141.14.0.0 (یک بلاک کلاس B) در نظر بگیرید. این سازمان یک آدرس‌دهی طبقاتی دو سطحی (شماره شبکه و شماره ماشین) دارد اما همانطور که شکل ۴-۲۰ نشان می‌دهد، این سازمان نمی‌تواند بیش از یک شبکه فیزیکی داشته باشد. دقت نمایید که مسک پیش‌فرض (255.255.0.0) به این معنی است که همه آدرسهای این سازمان در 16 بیت اول خود مقدار یکسانی دارند (141.14). بقیه آدرسها، آدرس ماشینهای مختلف در این شبکه را تعریف می‌کنند. همچنین دقت نمایید که آدرس شبکه، اولین آدرس بلاک است. در واقع در آدرس شبکه، شماره ماشین برابر با تمام صفر است.

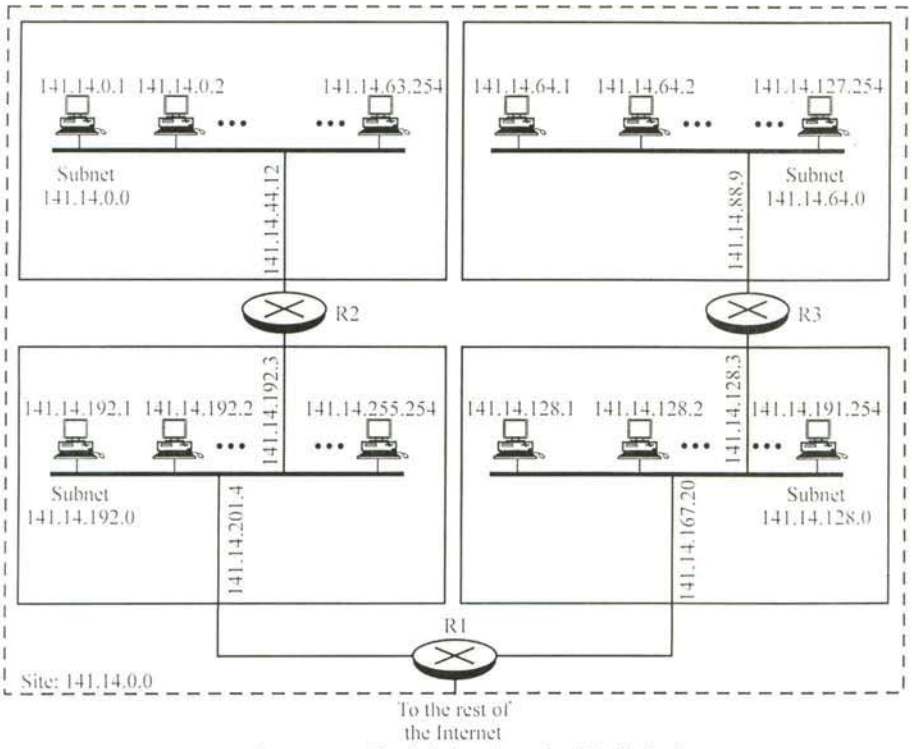


شکل ۴-۲۰: یک شبکه با آدرس IP طبقاتی دوسطحی (بدون زیرشبکه‌سازی)

سازمان با این طرح آدرس‌دهی به یک آدرس‌دهی طبقاتی دو سطحی محدود شده است. نمی‌توان ماشینها را به صورت گروه سازماندهی کرد. همه ماشینها در یک سطح هستند. سازمان، یک شبکه با تعداد زیادی ماشین دارد.

www.Abiri.ir

یک راه حل این مسئله، استفاده از زیرشبکه‌سازی است به این معنی که شبکه را به شبکه‌های کوچکتری به نام **زیرشبکه**[1] تقسیم کنیم. به عنوان مثال شکل ۴-۲۱ نشان می‌دهد که شبکه شکل ۴-۲۰ به چهار زیرشبکه تقسیم شده است.



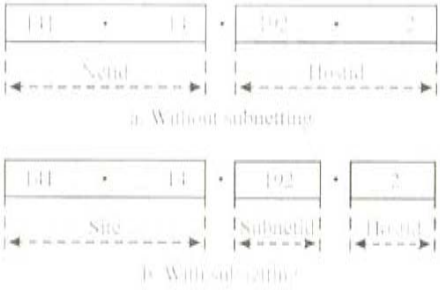
شکل ۴-۲۱ : شبکه‌ای با طبقه‌بندی سه سطحی

در این مثال، بقیه اینترنت از این موضوع خبر ندارد که شبکه به زیرشبکه‌های فیزیکی تقسیم شده است. هنوز زیرشبکه‌ها توأماً از دید بقیه اینترنت، یک شبکه واحد هستند. بسته‌ای که برای مقصد 141.14.192.2 ارسال شده است، هنوز هم ابتدا به دست روتر R1 می‌رسد. با این حال زمانی که بسته به روتر R1 می‌رسد، تفسیر آدرس IP مقصد بسته تغییر می‌کند. روتر R1 می‌داند شبکه 141.14.0.0 به صورت فیزیکی به چهار زیرشبکه تقسیم شده است. او می‌داند که بسته باید به زیرشبکه 141.14.192.0 تحویل داده شود.

1- Subnet

www.Abiri.ir

اضافه کردن زیرشبکه، یک سطح مبانی به سیستم آدرس‌دهی اضافه می‌کند. حا
سطح داریم: *سایت*[1]، زیرشبکه و *ماشین*[2]. سایت، اولین سطح است، سطح دوم، زیر
ست. ماشین، سطح سوم است. او اتصال ماشین به زیرشبکه را تعریف می‌کند. شکل ۲
ببینید.



شکل ۲ ۲۲ : آدرسها در یک شبکه با زیرشبکه‌سازی و بدون زیرشبکه‌سازی

حالا بسته‌دهی یک بسته IP شامل سه مرحله می‌شود. ابتدا تحویل بسته به
سس تحویل بسته به زیرشبکه و در نهایت، تحویل بسته به ماشین.

این مشابه با شماره تلفن ده رقمی در امریکاست. همانطور که شکل ۲ ۲۳
می‌دهد، یک شماره تلفن به سه سطح شکسته می‌شود: *کد ناحیه*[3]، *شماره مرکز تلف*
شماره مشترک[4].



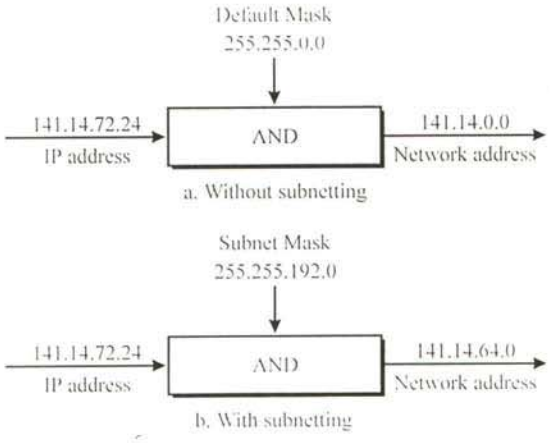
شکل ۲ ۲۳ : مفهوم طبقاتی یک شماره تلفن

◊ مسک زیرشبکه

قبلا درباره مسک بس‌فرض بحث کردیم. وقتی که شبکه‌ای، زیرشبکه‌سازی
باشد، از مسک بس‌فرض استفاده می‌کنید. مسک بس‌فرض برای تعدا کردن اولین اثر

www.Abiri.ir

بلاک آدرس که همان آدرس شبکه است، به کار می‌رود. با این وجود وقتی که شبکه‌ای، زیرشبکه‌سازی می‌شود، وضعیت تغییر می‌کند. باید یک مسک زیرشبکه داشته باشیم. مسک پیش‌فرض، یکهای بیشتری دارد. شکل ۴-۲۴ وضعیت را در دو شبکه پیشین نشان می‌دهد (شکل ۴-۲۰ و ۴-۲۱ را ببینید). مسک پیش‌فرض، آدرس شبکه را می‌دهد. مسک زیرشبکه، آدرس زیرشبکه را می‌دهد.



شکل ۴-۲۴ : مسک پیش‌فرض و مسک زیرشبکه

❊❊ مسک پیوسته زیرشبکه و مسک ناپیوسته زیرشبکه

در روزهای اول زیرشبکه‌سازی، ممکن بود از یک ***مسک ناپیوسته زیرشبکه***[1] استفاده شود. منظورمان از ناپیوسته این است که دنباله‌ای از صفرها به دنبال دنباله‌ای از یکها نمی‌آیند بلکه مخلوطی از صفرها و یکها به کار می‌روند. اما امروزه فقط از ***مسکهای پیوسته زیرشبکه***[2] که در آن دنباله‌ای از صفرها به دنبال دنباله‌ای از یکها می‌آیند، استفاده می‌شود.

❊❊ پیدا کردن آدرس زیرشبکه

با داشتن یک آدرس IP، می‌توان به همان روشی که در فصل قبل، آدرس شبکه را بدست آوردیم، در اینجا، آدرس زیرشبکه را بدست آوریم. به این منظور، مسک را به آدرس اعمال می‌کنیم.

1- Noncontiguous subnet mask
2- Contiguous subnet mask

www.Abiri.ir

***** مثال ۱۶**

اگر آدرس مقصد برابر با 200.45.34.56 باشد و مسک زیرشبکه برابر با 255.255.240.0 باشد، آدرس زیرشبکه را بدست آورید.

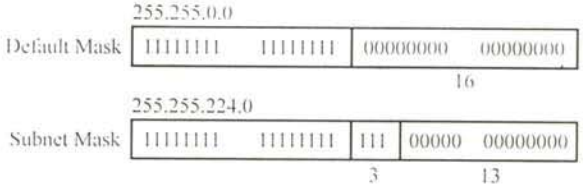
راه حل: آدرس مقصد را با مسک زیرشبکه AND می‌کنیم.

```
Address            → 11001000 00101101 00100010 00111000
Subnet Mask        → 11111111 11111111 11110000 00000000
Subnetwork Address → 11001000 00101101 00100000 00000000
```

آدرس زیرشبکه برابر با 200.45.32.0 بدست می‌آید.

**** مسک پیش‌فرض و مسک زیرشبکه**

تعداد 1 ها در مسک پیش‌فرض از قبل تعیین شده است (8 ، 16 و یا 24). در یک مسک زیرشبکه، تعداد 1 ها بیش از تعداد 1 های مسک پیش‌فرض متناظر با آن است. به عبارت دیگر برای بدست آوردن مسک زیرشبکه، برخی از 0 های سمت چپ دنباله صفرها در مسک پیش‌فرض را به 1 تبدیل می‌کنیم تا مسک زیرشبکه بدست آید. شکل ۴-۲۵ تفاوت بین یک مسک پیش‌فرض کلاس B و مسک زیرشبکه را برای همان بلاک نشان می‌دهد.



شکل ۴-۲۵: مقایسه مسک پیش‌فرض با مسک زیرشبکه

**** تعداد زیرشبکه‌ها**

تعداد زیرشبکه‌ها را می‌توان با شمردن یکهای اضافه‌ای که به مسک پیش‌فرض اضافه شده‌اند تا مسک زیرشبکه را بسازند، بدست آورد. به عنوان مثال در شکل ۴-۲۵ ، تعداد یکهایی که به مسک پیش‌فرض اضافه شده است برابر با سه تا یک است. بنابراین، تعداد زیرشبکه‌ها برابر با 2^3 یا 8 خواهد بود.

www.Abiri.ir

❊❊ تعداد آدرسها در هر زیرشبکه

می‌توان با شمردن تعداد صفرها در مسک زیرشبکه، تعداد آدرسها در یک زیرشبکه را بدست آورد. به عنوان مثال در شکل ۴-۲۵ ، تعداد صفرها برابر با 13 صفر است. بنابراین، تعداد آدرسها در هر زیرشبکه برابر با $2^{13} = 8192$ آدرس می‌باشد.

❊ آدرسهای خاص در زیرشبکه‌سازی

با زیرشبکه‌سازی، دو آدرس به لیست آدرسهای خاص که قبلاً بحث آنها آمد، اضافه می‌شود. اولین آدرس در هر زیرشبکه (با شماره ماشین تمام صفر)، آدرس زیرشبکه است. آخرین آدرس در هر زیرشبکه (با شماره ماشین تمام یک) ، آدرس همگانی محدود در آن زیرشبکه است. بعضی از آدرسهای دیگر نیز ابتدا به عنوان آدرسهای خاص، رزرو شده بودند اما همانطور که خواهیم دید با ابداع آدرس‌دهی بدون‌کلاس، این ایده کنار گذاشته شده است.

❊ نماد CIDR برای زیرشبکه‌ها

وقتی که زیرشبکه داریم نیز می‌توان از CIDR استفاده کرد. می‌توان با استفاده از این نماد به راحتی یک آدرس را در یک زیرشبکه تعریف کرد. به عنوان مثال، نماد 141.14.192.3/16 یک آدرس کلاس B را نشان می‌دهد ولی آدرس 141.14.192.3/18 نشان می‌دهد که این آدرس متعلق به زیرشبکه‌ای با مسک 255.255.192.0 است. وقتی که در فصل بعد، موضوع آدرس‌دهی بدون‌کلاس را مطرح می‌کنیم، بیشتر درباره این موضوع بحث خواهیم کرد.

سوپرشبکه

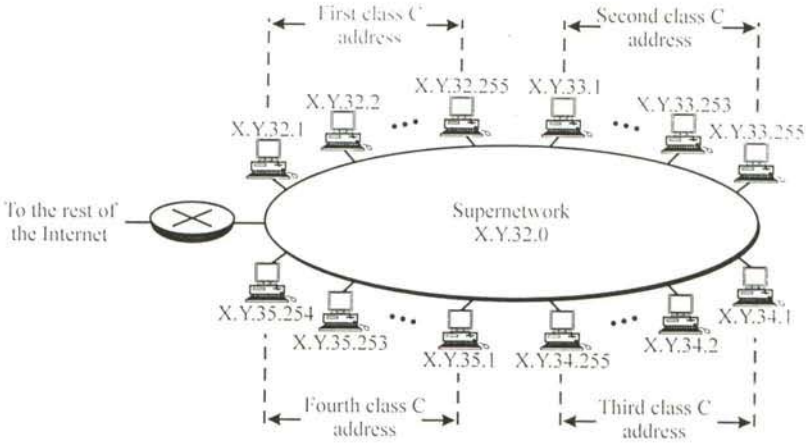
گرچه آدرسهای کلاس A و کلاس B تقریباً تمام شده‌اند، آدرسهای کلاس C هنوز موجود می‌باشند. با این وجود، تعداد 256 آدرس موجود در یک بلاک آدرس کلاس C ممکن است کمتر از آدرسهای مورد نیاز یک سازمان باشد. حتی یک سازمان متوسط نیز ممکن است به آدرسهای بیشتری نیاز داشته باشد.

یک راه حل، استفاده از ***سوپرشبکه‌سازی***[1] است. در این روش، سازمان می‌تواند با ادغام چند بلاک آدرس کلاس C ، یک بلاک آدرس بزرگتری را بسازد. به عبارت دیگر، چندین شبکه را می‌توان در هم ادغام کرد و یک سوپرشبکه ساخت. برای انجام این کار، سازمان می‌تواند مجموعه‌ای از بلاکهای آدرس کلاس C را به جای صرفاً یک بلاک آدرس کلاس C

1- Supernetting

www.Abiri.ir

درخواست نماید. به عنوان مثال، به سازمانی که 1000 آدرس نیاز دارد، می‌توان چهار بلاک آدرس کلاس C داد. سپس سازمان می‌تواند مطابق شکل ۴-۲۶ این آدرسها را در یک سوپرشبکه به کار ببرد.



شکل ۴-۲۶ : یک سوپرشبکه

* مسک سوپرشبکه

وقتی یک بلاک آدرس (کلاس A ، B یا C) به سازمانی داده می‌شود، اولین آدرس بلاک و مسک پیش‌فرض، کل آدرسهای بلاک را تعریف می‌کنند. از آنجا که مسک پیش‌فرض برای هر کلاس آدرس مقدار مشخصی است، صرف دانستن آدرس شبکه برای مشخص کردن کل آدرسهای بلاک آدرس، کفایت می‌کند.

وقتی که سازمانی، بلاک آدرس را به زیرشبکه‌ها تقسیم می‌کند، اولین آدرس در زیرشبکه و مسک زیرشبکه، کلیه آدرسهای موجود در یک زیرشبکه را تعریف می‌کنند. در این حالت بر خلاف حالت قبل، فقط اولین آدرس بلاک کافی نیست بلکه باید مسک زیرشبکه را نیز بدانیم.

به طور مشابه وقتی که سازمانی چندین بلاک آدرس را به صورت یک سوپرشبکه در هم ادغام می‌کند، باید اولین آدرس بلاک و مسک سوپرشبکه را بدانیم. در اینجا نیز آدرس شبکه به تنهایی نمی‌تواند بلاک آدرس را تعریف کند. یک مسک سوپرشبکه نیاز داریم تا بدانیم باید چند بلاک آدرس را در هم ادغام کنیم تا یک سوپرشبکه بسازیم.

www.Abiri.ir

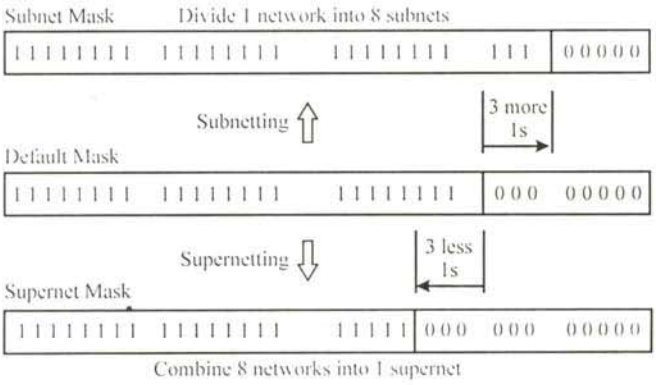
مهم: در زیرشبکه‌سازی، برای تعریف بلاک آدرسها باید اولین آدرس زیرشبکه و مسک زیرشبکه را داشته باشیم. در سوپرشبکه‌سازی برای تعریف بلاک آدرسها باید اولین آدرس سوپرشبکه و مسک سوپرشبکه را داشته باشیم.

یک مسک سوپرشبکه، معکوس یک مسک زیرشبکه است. یک مسک زیرشبکه برای کلاس C ، یکهای بیشتری نسبت به مسک پیش‌فرض کلاس C دارد. یک مسک سوپرشبکه کلاس C ، یکهای کمتری نسبت به مسک پیش‌فرض کلاس C دارد.

شکل ۴-۲۷ تفاوت بین یک مسک زیرشبکه و یک مسک سوپرشبکه را نشان می‌دهد. یک مسک زیرشبکه که بلاک آدرسها را به هشت زیربلاک آدرس تقسیم می‌کند، سه یک (2^3=8) بیشتر از مسک پیش‌فرض دارد. یک مسک سوپرشبکه که هشت بلاک آدرس را در هم ادغام می‌کند و یک سوپربلاک می‌سازد، سه یک، کمتر از مسک پیش‌فرض دارد.

* نماد CIDR برای سوپرشبکه‌ها

نماد CIDR را می‌توان برای سوپرشبکه‌ها نیز به کار برد. یک آدرس در یک سوپرشبکه را می‌توان به راحتی با استفاده از این نماد تعریف کرد. به عنوان مثال نماد 192.14.192.3/24 یک آدرس کلاس C را نشان می‌دهد اما آدرس 192.14.192.3/21 نشان می‌دهد که این آدرس متعلق به یک سوپرشبکه با مسک 255.255.248.0 است.



شکل ۴-۲۷ : مقایسه مسکهای پیش‌فرض، زیرشبکه و سوپرشبکه

www.Abiri.ir

منسوخ شده‌ها

با ابداع آدرس‌دهی بدون‌کلاس ، ایده زیرشبکه‌سازی و سوپرشبکه‌سازی آدرسهای کلاس‌دار تقریباً منسوخ شده است. در اینجا به بحث مختصری پیرامون آدرسهای کلاس‌دار پرداختیم تا خواننده را برای آشنا شدن با مفهوم آدرس‌دهی کلاس‌دار که در فصل پنجم خواهد آمد، آماده کنیم.

مهم: ایده زیرشبکه‌سازی و سوپرشبکه‌سازی آدرسهای کلاس‌دار تقریباً منسوخ شده است.

۵-۴ عبارات کلیدی

فضای آدرس (address space)
عملگر AND (AND operation)
نماد باینری (binary notation)
بلاک آدرسها (block of addresses)
آدرس کلاس A (class A address)
آدرس کلاس B (class B address)
آدرس کلاس C (class C address)
آدرس کلاس D (class D address)
آدرس کلاس E (class E address)
آدرس‌دهی کلاس‌دار (Classful addressing)
آدرس‌دهی بدون‌کلاس (Classless addressing)
نماد CIDR (Classless InterDomain Routing Notation)
مسک پیوسته (Contiguous mask)
مسک پیش‌فرض (default mask)
آدرس همگانی صریح (direct broadcast address)
نماد اعداد صحیح نقطه‌دار (dotted decimal notation)
نماد هگزادسیمال (hexadecimal notation)
شماره ماشین (hostid)
آدرس همگانی محدود (limited broadcast address)
آدرس برگردان (loopback address)
مسک (mask)

www.Abiri.ir

وسیله چند آدرسه (multihomed device)

شماره شبکه (netid)

آدرس شبکه (network address)

مسک ناپیوسته (noncontiguous mask)

یک ماشین مشخص روی این شبکه (specific host on this network)

زیرشبکه (subnet)

مسک زیرشبکه (subnet mask)

زیرشبکه‌سازی (subnetting)

مسک سوپرشبکه (supernet mask)

سوپرشبکه‌سازی (supernetting)

این ماشین روی این شبکه (this host on this network)

۶-۴ خلاصه

- در لایه شبکه، یک سیستم آدرس‌دهی سرتاسری لازم است تا هر روتر و کامپیوتر متصل به شبکه را به طور منحصر به فردی مشخص نماید و به این ترتیب معلوم باشد که بسته برای رسیدن به مقصد نهایی خود باید از چه وسایلی عبور کند.
- آدرس IP، 32 بیتی است (برای IPv4) و به طور منحصر به فرد و سرتاسری یک کامپیوتر یا روتر را روی اینترنت مشخص می‌کند.
- آن قسمت از آدرس IP که شبکه را مشخص می‌کند به نام شماره شبکه موسوم است.
- آن قسمت از آدرس IP که یک کامپیوتر یا روتری را روی شبکه مشخص می‌کند، به نام شماره ماشین موسوم است.
- یک آدرس IP ، اتصال یک وسیله به اینترنت را مشخص می‌کند.
- پنج کلاس آدرس IP وجود دارد. کلاسهای A ، B و C در تعداد ماشینهای هر بلاک آدرس، با هم تفاوت دارند.کلاس D برای چندگیرندگی بکار می‌رود و کلاس E رزرو شده است.
- با بررسی اولین بایت آدرس، می‌توان به سهولت، کلاس یک شبکه را تعیین کرد.
- یک وسیله چند آدرسه به چندین شبکه متصل شده است و برای هر شبکه‌ای که به آن متصل شده است، یک آدرس IP دارد.
- برای کلاسهای A ، B و C اگر روتری بخواهد بسته‌ای را برای همه ماشینهای داخل یک شبکه مشخص ارسال کند، از یک آدرس همگانی صریح (شماره ماشین تمام یک) استفاده می‌کند.

www.Abiri.ir

- یک آدرس همگانی محدود (255.255.255.255) برای ارسال بسته‌ای برای تمام ماشینهای واقع در شبکه‌ای که ماشین فرستنده بسته نیز در آن شبکه واقع شده است، به کار می‌رود.
- اگر یک ماشین به هنگام بوت شدن آدرس IP خود را نداند، از یک آدرس IP مبدأ تمام صفر 0.0.0.0 استفاده می‌کند.
- اگر یک ماشین بخواهد بسته‌ای را برای ماشین دیگری واقع در همان شبکه ماشین فرستنده بسته، ارسال نماید، شماره شبکه آدرس مقصد بسته IP را برابر با تمام صفر قرار می‌دهد.
- برای آنکه یک ماشین، نرم‌افزار داخلی خود را تست کند، از یک آدرس برگردان استفاده می‌کند که اولین بایت آن 127 است.
- اکثر آدرسهای کلاسهای A ، B و C برای مخابره تک‌گیرنده به کار می‌روند.
- آدرسهای کلاس D برای مخابره چندگیرنده به کار می‌روند.
- اینترنت از ارسال همگانی به صورت سرتاسری پشتیبانی نمی‌کند.
- زیرشبکه‌سازی، یک شبکه بزرگ را به چندین شبکه کوچکتر تقسیم می‌کند.
- زیرشبکه‌سازی، یک سطح میانی به سلسله مراتب موجود در آدرس‌دهی IP اضافه می‌کند.
- می‌توان کلاسهای A ، B و C را زیرشبکه‌سازی کرد.
- به کمک مسک زیرشبکه می‌توان آدرس زیرشبکه را از یک آدرس IP داده شده بدست آورد.
- با AND کردن آدرس IP و مسک می‌توان آدرس شبکه و یا آدرس زیرشبکه را بدست آورد
- مفهوم آدرسهای خاص در آدرس‌دهی IP در زیرشبکه‌ها نیز به کار می‌رود.
- یک مسک پیوسته (دنباله‌ای از صفرها بعد از دنباله‌ای از یکها) بر یک مسک ناپیوسته، ارجحیت خیلی بیشتری دارد.
- سوپرشبکه‌سازی از ادغام چندین شبکه و ساختن یک شبکه بزرگتر بدست می‌آید.

۴-۷ مجموعه تمرین‌ها

تمرین‌ها

۱-فضای آدرس در هر یک از سیستم‌های زیر چیست؟

الف- سیستمی با آدرسهای 8 بیتی

ب- سیستمی با آدرسهای 16 بیتی

www.Abiri.ir

ج- سیستمی با آدرسهای 64 بیتی

۲- یک فضای آدرس حاوی 1024 آدرس است. برای نمایش یک آدرس، به چه تعداد بیت نیاز داریم؟

۳- یک فضای آدرس از سه سمبول 0 و 1 و 2 در نمایش آدرسهای خود استفاده می‌کند. اگر هر آدرس از 10 سمبول ساخته شود، در این سیستم چند آدرس داریم؟

۴- آدرسهای IP زیر را از نماد اعداد صحیح نقطه‌دار به نماد باینری تغییر دهید.

الف – 114.34.2.8

ب- 129.14.6.8

ج- 208.34.54.12

د- 238.34.2.1

ه - 241.34.2.8

۵- آدرسهای IP زیر را از نماد اعداد صحیح نقطه‌دار به نماد هگزادسیمال تغییر دهید.

الف- 114.34.2.8

ب- 129.14.6.8

ج- 208.34.54.12

د- 238.34.2.1

ه - 241.34.2.8

۶- آدرسهای IP زیر را از نماد هگزادسیمال به نماد باینری تغییر دهید.

الف- 0x1347FEAB

ب- 0xAB234102

ج- 0x0123A2BE

د- 0x00001111

۷- در هر یک از کلاسهای زیر برای نمایش شماره شبکه با نماد هگزادسیمال، به چند رقم نیاز داریم؟

الف- کلاس A

ب- کلاس B

ج- کلاس C

۸- آدرسهای IP زیر را از نماد باینری به نماد اعداد صحیح نقطه‌دار تغییر دهید.

الف- 01111101 01100111 11110000 01111111

ب- 00011101 11111000 11000000 10101111

ج- 01011101 00011111 10110000 11011111

www.Abiri.ir

د- 11101111 11110111 11000111 00011101

ه - 11110111 11110011 10000111 11011101

۹- کلاس آدرسهای IP زیر را بدست آورید.

الف- 208.34.54.12

ب- 238.34.2.1

ج- 114.34.2.8

د- 129.14.6.8

ه - 241.34.2.8

۱۰- کلاس آدرسهای IP زیر را بدست آورید :

الف- 11110111 11110011 10000111 11011101

ب- 10101111 11000000 11110000 00011101

ج- 11011111 10110000 00011111 01011101

د- 11101111 11110111 11000111 00011101

ه- 01111111 11110000 01100111 01111101

۱۱- شماره شبکه و شماره ماشین آدرسهای IP زیر را بدست آورید.

الف- 114.34.2.8

ب- 132.56.8.6

ج- 208.34.54.12

۱۲- کامپیوتری با آدرس IP برابر با 128.23.67.3 پیغامی را برای کامپیوتر دیگری با آدرس IP برابر با 193.45.23.7 ارسال می‌کند. آیا این پیغام از روتری عبور می‌کند؟ فرض کنید که زیرشبکه نداریم.

۱۳- کامپیوتری با آدرس IP برابر با 128.23.67.3 پیغامی را برای کامپیوتر دیگری با آدرس IP برابر با 128.45.23.7 ارسال می‌کند. آیا این پیغام از روتری عبور می‌کند؟ فرض کنید که زیرشبکه نداریم.

۱۴- کامپیوتری با آدرس IP برابر با 128.23.67.3 پیغامی را برای کامپیوتر دیگری با آدرس IP برابر با 128.23.23.7 ارسال می‌کند. آیا این پیغام از روتری عبور می‌کند؟ فرض کنید که زیرشبکه نداریم.

۱۵- دیاگرامی برای شبکه‌ای به آدرس 8.0.0.0 ترسیم نمایید که از طریق یک روتر به شبکه‌ای با آدرس IP برابر با 131.45.0.0 متصل شده است. برای هر اینترفیس روتر یک آدرس IP انتخاب کنید. چند کامپیوتر را نیز به همراه آدرسهای IP آنها در هر شبکه نشان دهید. کلاس هر شبکه چیست؟

www.Abiri.ir

۱۶- آدرس IP اینترفیسی از روتر که به شبکه‌ای متصل است برابر با 108.5.18.22 می‌باشد. روتر یک بسته به آدرس همگانی صریح برای تمام کامپیوترها در آن شبکه می‌فرستد. آدرس IP مبدأ و مقصد به کار رفته در این بسته ارسالی چیست؟

۱۷- کامپیوتری با آدرس IP برابر با 108.67.18.70 یک بسته به آدرس همگانی محدود برای تمام کامپیوترها در همان شبکه ارسال می‌کند. آدرس IP مبدأ و مقصد به کار رفته در این بسته ارسالی چیست؟

۱۸- یک کامپیوتر با آدرس IP برابر با 185.67.89.34 به تست نیاز دارد. آدرسهای مبدأ و مقصد کدامند؟

۱۹- یک کامپیوتر با آدرس IP برابر با 123.27.19.24 پیغامی را برای کامپیوتر دیگری با آدرس IP برابر با 123.67.89.56 با استفاده از آدرس خاص « یک کامپیوتر مشخص روی این شبکه » ارسال می‌کند. آدرسهای مبدأ و مقصد کدامند؟

۲۰- یک کامپیوتر در کلاس C ، آدرس IP خودش را نمی‌داند لذا با ارسال پیغامی برای یک سرور bootstarp آدرس خودش را سؤال می‌کند. آدرسهای مبدأ و مقصد کدامند؟

۲۱- آیا می‌توانیم آدرسی به صورت x.y.z.t/32 داشته باشیم؟ توضیح دهید.

۲۲- در کلاس A، اولین آدرس در شبکه‌ای (آدرس شبکه) برابر با 20.0.0.0 می‌باشد. دویست و بیست هزارمین آدرس کدام است؟ راهنمایی: از سیستم شماره‌گذاری مبنای 256 استفاده کنید.

۲۳- در یک شبکه، آدرس یک کامپیوتر برابر با 201.78.24.56 و آدرس کامپیوتر دیگری برابر با 201.78.120.202 است. چه تعداد آدرس بین این دو حد وجود دارد؟ راهنمایی: از سیستم شماره‌گذاری مبنای 256 استفاده کنید.

۲۴- در یک زیرشبکه کلاس A، آدرس IP و مسک یکی از کامپیوترها را به صورت زیر می‌دانیم:

آدرس IP : 25.34.12.56

مسک : 255.255.0.0

اولین آدرس چیست (آدرس زیرشبکه) ؟

۲۵- در یک زیرشبکه کلاس B ، آدرس IP و مسک یکی از کامپیوترها را به صورت زیر می‌دانیم:

آدرس IP : 125.134.112.66

مسک : 255.255.224.0

اولین آدرس چیست (آدرس زیرشبکه) ؟

۲۶- در یک زیرشبکه کلاس C ، آدرس IP و مسک یکی از ماشینها را به صورت زیر می‌دانیم:

www.Abiri.ir

آدرس IP : 182.44.82.16

مسک : 255.255.255.192

اولین آدرس چیست (آدرس زیرشبکه) ؟

۲۷- در هر یک از حالات زیر، مسک پیوسته را بدست آورید:

الف- 1024 زیرشبکه در کلاس A

ب- 256 زیرشبکه در کلاس B

ج- 32 زیرشبکه در کلاس C

د- 4 زیرشبکه در کلاس C

۲۸- در هر یک از حالات زیر، حداکثر تعداد زیرشبکه‌ها چندتاست؟

الف- کلاس A ، مسک 255.255.192.0

ب- کلاس B ، مسک 255.255.192.0

ج- کلاس C ، مسک 255.255.255.192

د- کلاس C ، مسک 255.255.255.240

فعالیتهای تحقیقاتی

۲۹- آیا مدرسه یا سازمان شما از یک آدرس کلاس‌دار استفاده می‌کند؟ اگر این چنین است، کلاس آدرس را پیدا کنید.

۳۰- همه RFCهایی که درباره آدرسهای IP هستند را پیدا کنید.

۳۱- همه RFCهایی که درباره آدرس برگردان هستند را پیدا کنید.

۳۲- همه RFCهایی که درباره آدرسهای خصوصی هستند را پیدا کنید.

۳۳- همه RFCهایی که درباره آدرس همگانی صریح هستند را پیدا کنید.

۳۴- همه RFCهایی که درباره آدرسهای چندگیرنده هستند را پیدا کنید.

www.Abiri.ir

فصل پنجم

آدرس‌های IP – آدرس‌دهی بدون‌کلاس

استفاده از ***آدرس‌دهی کلاس‌دار***[1] مشکلات بسیاری را بوجود آورده است. تا اواسط دهه ۱۹۹۰، دنباله‌ای از آدرسها به معنی بلاکی از آدرسهای کلاس A ، B یا C بود. حداقل تعداد آدرسی که به یک سازمان داده می‌شد برابر با 256 آدرس (کلاس C) و حداکثر تعداد آدرس برابر با 16777216 آدرس بود (کلاس A). بین این دو حد، یک سازمان می‌توانست یک بلاک آدرس کلاس B یا چند بلاک آدرس کلاس C داشته باشد. تخصیص این آدرسها همواره به صورت مضربی از 256 صورت می‌گرفت. ولی شرکت کوچکی که فقط به 16 آدرس نیاز داشت، چطور؟ یا مشترک یک منزل مسکونی که فقط به دو آدرس نیاز داشت، چطور؟

در خلال دهه ۱۹۹۰، ***ارائه‌دهندگان سرویسهای اینترنتی***[2] کاری در این رابطه انجام دادند. یک ISP، شرکتی است که برای افراد، شرکتهای کوچک و سازمانهای متوسطی که نمی‌خواهند یک سایت اینترنتی راه بیندازند و درگیر ارائه خدمات اینترنتی شوند (مانند سرویس email)، امکان دسترسی به اینترنت را فراهم می‌کند. محدوده‌ای از آدرسهای IP به یک ISP داده می‌شود، و سپس آن ISP ، آدرسهای دریافتی را به صورت گروههای 2 تایی، 4 تایی، 8 تایی، 16 تایی و غیره در اختیار مشترکین منازل مسکونی و شرکتهای کوچک قرار می‌دهد. مشترکین با ***شماره‌گیری از طریق یک مدم***[3] ، خط DSL ، یا مدم کابلی به ISP وصل می‌شوند. با این حال هر مشترک، به یک آدرس IP نیاز دارد (سایر راه حلها مانند استفاده از آدرسهای خصوصی و نیز ترجمه آدرسها را در فصل بیست و ششم خواهیم دید).

برای حل مشکلات آدرس‌دهی کلاس‌دار و امکان ارائه خدمات فوق الذکر، مسئولین اینترنت در سال ۱۹۹۶، معماری جدیدی به نام ***آدرس‌دهی بدون‌کلاس***[4] را ارائه کرده‌اند که به تدریج جای آدرس‌دهی کلاس‌دار را گرفته است.

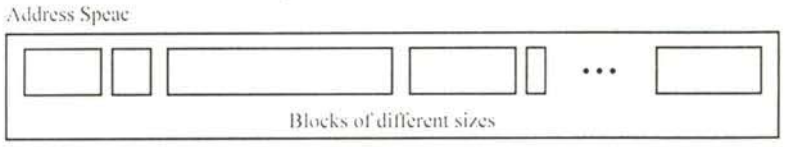
1- Classful addressing
2- Internet Service Provider : ISP
3- Dial – up modem
4- Classless addressing

www.Abiri.ir

۵-۱ بلاکهای آدرس با طول متغیر

در آدرس‌دهی بدون‌کلاس، بلاکهای آدرس با طول متغیر به متقاضی داده می‌شود و این بلاکهای آدرس، متعلق به هیچ کلاس آدرسی نیستند. می‌توانیم یک بلاک آدرس حاوی 2 آدرس، 4 آدرس، 128 آدرس و غیره داشته باشیم. البته محدودیتهایی وجود دارد اما در حالت کلی، یک بلاک آدرس می‌تواند تعداد کم و یا تعداد زیادی آدرس داشته باشد.

در این معماری، کل فضای آدرس (2^{32} آدرس) به بلاکهایی با طول متغیر تقسیم می‌شود. به یک سازمان، یک بلاک آدرس، مناسب با نیازهای آن سازمان، داده می‌شود. شکل ۵-۱ معماری آدرس‌دهی بدون‌کلاس را نشان می‌دهد. این شکل را با شکل ۴-۲ فصل چهارم مقایسه کنید.



شکل ۵-۱ : بلاکهای آدرس به طول متغیر

آدرس‌دهی بدون‌کلاس، بسیاری از مسائل را حل می‌کند. با این وجود محدودیتهایی لازم است تا این ساختار آدرس‌دهی، کار کند.

تعداد آدرسها در یک بلاک آدرس

فقط یک محدودیت بر تعداد آدرسهای یک بلاک آدرس اعمال می‌شود و آن اینکه تعداد آدرسهای یک بلاک آدرس باید توانی از 2 (2 ، 4 ، 8 ، ...) باشد. به عنوان مثال، می‌توان به یک مشترک منزل مسکونی، یک بلاک آدرس حاوی 2 (2^1) آدرس داد. می‌توان به یک شرکت کوچک، یک بلاک آدرس حاوی 16 (2^4) آدرس داد. می‌توان به یک سازمان بزرگ، یک بلاک آدرس حاوی 1024 (2^{10}) آدرس داد.

* اولین آدرس

اولین آدرس باید بر تعداد آدرسهای بلاک، قابل قسمت باشد. به عنوان مثال اگر بلاکی حاوی 4 آدرس است، اولین آدرس بلاک باید بر 4 قابل قسمت باشد. اگر بلاکی حاوی 16 آدرس است، اولین آدرس بلاک باید بر 16 قابل قسمت باشد.

www.Abiri.ir

اگر بلاکی حاوی 256 آدرس یا کمتر باشد، فقط لازم است که بایت سمت راست آدرس را چک کنیم. اگر بلاکی حاوی 65536 آدرس یا کمتر باشد، فقط لازم است که دو بایت سمت راست آدرس را چک کنیم.

*** مثال ۱**

کدام یک از آدرسهای زیر می‌تواند آدرس شروع یک بلاک آدرس، حاوی 16 آدرس باشد؟

الف – 205.16.37.32

ب – 190.16.42.44

ج – 17.17.33.80

د - 123.45.24.52

راه حل : فقط دو مورد الف و ج صحیح هستند. آدرس 205.16.37.32 می‌تواند اولین آدرس باشد زیرا که 32 بر 16 قابل قسمت است. آدرس 17.17.33.80 می‌تواند اولین آدرس باشد زیرا که 80 بر 16 قابل قسمت است.

*** مثال ۲**

کدام یک از آدرسهای زیر می‌تواند اولین آدرس یک بلاک آدرس حاوی 256 آدرس باشد؟

الف – 205.16.37.32

ب – 190.16.42.0

ج – 17.17.32.0

د- 123.45.24.52

راه حل : در این حالت، بایت سمت راست آدرس باید صفر باشد. همانطور که در فصل چهارم اشاره کردیم، آدرسهای IP از پایه 256 استفاده می‌کنند. وقتی که بایت سمت راست برابر با 0 باشد، آدرس بر 256 قابل قسمت است. فقط آدرسهای ب و ج اینطور هستند.

*** مثال ۳**

کدام یک از آدرسهای زیر می‌تواند آدرس شروع یک بلاک آدرس، حاوی 1024 آدرس باشد؟

الف – 205.16.37.32

ب – 190.16.42.0

ج – 17.17.32.0

www.Abiri.ir

د - 123.45.24.52

راه حل : در این حالت، لازم است که دو بایت را چک کنیم زیرا که 1024=4*256 بنابراین بایت اول از سمت راست آدرس باید بر 256 و بایت دوم از سمت راست آدرس باید بر 4 قابل قسمت باشد. این دو شرط فقط در مورد ج صدق می‌کنند.

مَسک

در آدرس‌دهی کلاس‌دار، **مسک**[1] پیش فرض هر کلاس آدرس به طور ضمنی مشخص است. مسک یک بلاک آدرس کلاس A برابر با 255.0.0.0 (و یا 8/) می‌باشد. مسک یک بلاک آدرس کلاس B برابر با 255.255.0.0 (و یا 16/) می‌باشد. مسک یک بلاک آدرس کلاس C برابر با 255.255.255.0 (و یا 24/) است. وقتی که آدرس داده می‌شود، می‌توانیم با استفاده از اولین بایت آدرس، کلاس آن را بدست آوریم و سپس با AND کردن مسک با آدرس داده شده، آدرس شروع بلاک و نیز محدوده آدرسهای موجود در آن بلاک آدرس را بدست آوریم.

از سوی دیگر در آدرس‌دهی بدون‌کلاس، وقتی که آدرس داده می‌شود، فقط به شرطی می‌توان بلاکی که آدرس متعلق به آن است را پیدا کنیم که مسک را داشته باشیم. به عبارت دیگر در آدرس‌دهی بدون‌کلاس، باید به همراه آدرس، مسک را نیز بدهیم. در نماد CIDR ، مسک را با نمایش تعداد 1 های موجود در آن، نشان می‌دهند. یک آدرس در آدرس‌دهی بدون‌کلاس، به فرمت شکل ۵-۲ می‌باشد.

x.y.z.t/n

شکل ۵-۲ : فرمت آدرس‌دهی بدون‌کلاس

n بعد از / (slash) ، نشان دهنده تعداد بیتهایی است که مقدار آنها در همه آدرسهای موجود در یک بلاک آدرس، یکسان است. بنابراین اگر n برابر با 20 باشد، به این معناست که 20 بیت اول همه آدرسها در آن بلاک آدرس، یکسان هستند و تفاوت آدرسها، در 12 بیت سمت راست آنهاست. با استفاده از این اطلاعات، به راحتی می‌توان آخرین آدرس بلاک و نیز تعداد آدرسهای موجود در بلاک را بدست آورد.

1- Mask

www.Abiri.ir

❋ پیشوند و طول پیشوند

دو عبارت که غالباً در آدرس‌دهی بدون‌کلاس به کار می‌روند، ***پیشوند***[1] و ***طول پیشوند***[2] می‌باشند. پیشوند، نام دیگری برای بخش مشترک آدرسهای موجود در یک بلاک آدرس است (مشابه با netid). طول پیشوند، تعداد 1 های موجود در پیشوند است (n در نماد CIDR). همانطور که جدول ۵-۱ نشان می‌دهد، یک رابطه یک به یک بین مسک و طول پیشوند وجود دارد.

/n	Mask	/n	Mask	/n	Mask	/n	Mask
/1	128.0.0.0	/9	255.128.0.0	/17	255.255.128.0	/25	255.255.255.128
/2	192.0.0.0	/10	255.192.0.0	/18	255.255.192.0	/26	255.255.255.192
/3	224.0.0.0	/11	255.224.0.0	/19	255.255.224.0	/27	255.255.255.224
/4	240.0.0.0	/12	255.240.0.0	/20	255.255.240.0	/28	255.255.255.240
/5	248.0.0.0	/13	255.248.0.0	/21	255.255.248.0	/29	255.255.255.248
/6	252.0.0.0	/14	255.252.0.0	/22	255.255.252.0	/30	255.255.255.252
/7	254.0.0.0	15	255.254.0.0	/23	255.255.254.0	/31	255.255.255.254
/8	255.0.0.0	16	255.255.0.0	24	255.255.255.0	/32	255.255.255.255

جدول ۵-۱ : طولهای مختلف پیشوند

توجه نمائید که سطر آخر جدول فوق، مسک پیش فرض را برای کلاسهای A ، B و C نشان می‌دهد و معنی آن این است که آدرس‌دهی کلاس‌دار، حالت خاصی از آدرس‌دهی بدون‌کلاس با نماد CIDR محسوب می‌شود.

❋ پسوند و طول پسوند

دو عبارتی که غالباً در آدرس‌دهی بدون‌کلاس به کار می‌روند، عبارتند از ***پسوند***[3] و ***طول پسوند***[4]. پسوند، قسمت سمت راست آدرس (مشابه با hostid) است و مقدار آن برای هر آدرسی در بلاک آدرس، مقدار متفاوتی است. طول پسوند، در نماد CDIR از رابط 32-n بدست می‌آید که در آن n طول پیشوند است.

1- Prefix
2- Prefix length
3- Suffix
4- Suffix length

www.Abiri.ir

❊ پیدا کردن بلاک

در آدرس‌دهی بدون‌کلاس، یک بلاک آدرس به یک سازمان داده می‌شود و سایز بلاک بستگی به نیاز سازمان دارد. زمانی که یک آدرس بدون‌کلاس داده می‌شود، می‌توانیم بلاک آدرس را پیدا کنیم. همچنین می‌توانیم اولین آدرس، تعداد آدرسها و آخرین آدرس را بدست آوریم.

❊ پیدا کردن اولین آدرس

طول پیشوند در آدرس‌دهی بدون‌کلاس، همان مسک را تعیین می‌کند. از آنجا که آدرسهای موجود در یک بلاک آدرس بدون‌کلاس، حتماً پیوسته هستند و پیشوند، تعداد 1 های مسک را تعیین می‌کند، لذا می‌توان با AND کردن مسک و یکی از آدرسهای موجود در بلاک آدرس، اولین آدرس بلاک را پیدا کرد. برای این کار کافی است که اولین n بیت آدرس را نگه داریم و بقیه بیتهای آدرس را صفر کنیم.

❊❊ مثال ۴

اگر یکی از آدرسها 167.199.170.82 / 27 باشد، اولین آدرس بلاک را بدست آورید.

راه حل : طول پیشوند برابر با 27 است و به این معنی است که باید اولین 27 بیت را نگه داریم و بقیه بیتها را 0 کنیم. آنچه در ذیل می‌آید، روند کار را نشان می‌دهد.

آدرس در نماد باینری به صورت زیر است:

10100111 11000111 10101010 01010010

با نگه داشتن 27 بیت سمت چپ و صفر کردن بقیه بیتها حاصل زیر بدست می‌آید:

10100111 11000111 10101010 01000000

نتیجه حاصل با نماد CIDR به صورت 167.199.170.64 / 27 می‌باشد.

❊ میان بر اول

می‌توانیم برای پیدا کردن اولین آدرس بلاک، از میان بر زیر استفاده کنیم:

۱- طول پیشوند را به چهار گروه، متناظر با چهار بایت یک آدرس، تقسیم کنید و تعداد 1 ها در هر گروه را بدست آورید.

۲- اگر تعداد یکها در یک گروه برابر با 8 باشد، بایت متناظر با آن گروه در اولین آدرس بلاک، همان بایت داده شده است.

۳- اگر تعداد 1 ها در یک گروه، برابر با 0 باشد، بایت متناظر با آن گروه در اولین آدرس بلاک، برابر با 0 است.

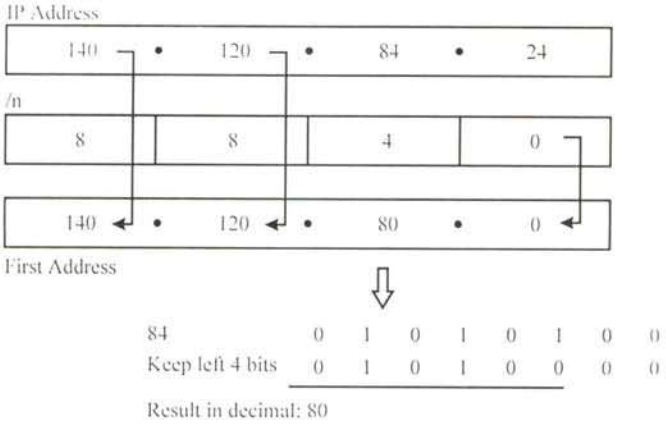
www.Abiri.ir

۴- اگر تعداد 1 ها در یک گروه، بین 0 و 8 باشد، به تعداد 1 ها از بیتهای سمت چپ بایت داده شده را نگه می‌داریم و بقیه بیتهای بایت داده شده را صفر می‌کنیم.

**** مثال ۵**

اگر یکی از آدرسهای یک بلاک آدرس برابر با 140.120.84.24 / 20 باشد، اولین آدرس بلاک چیست؟

راه حل : شکل ۵-۳ را ببینید. اولین، دومین و چهارمین بایت، ساده هستند. برای سومین بایت، بیتهای متناظر با یکهای گروه سوم را نگه می‌داریم و بقیه بیتهای آن گروه را صفر می‌کنیم. اولین آدرس برابر با 140.120.80.0 / 20 بدست می‌آید.



شکل ۵-۳ : مثال ۵

*** میان بر دوم**

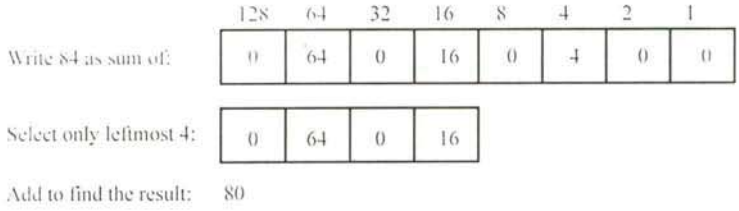
برای امتناع از بکارگیری اعداد باینری، روش دیگری را برای بایتی که تمام 1 و یا تمام 0 نیست، ارائه می‌کنیم. ابتدا بایت مورد نظر از آدرس را به صورت حاصل جمع اعدادی که توان 2 هستند (1, 2, 4, 8, 16, 32, 64, 128) می‌نویسیم. اگر ارزش مکانی یکی از این توانهای 2 در بایت داده شده وجود نداشته باشد، به جای آن، 0 می‌گذاریم. سپس m تا از بالاترین توانها را جدا می‌کنیم که در آن m ، برابر با طول پیشوند مربوط به آن بایت است.

**** مثال ۶**

اگر یکی از آدرسهای بلاک برابر با 140.120.84.24 / 20 باشد، اولین آدرس بلاک را بدست آورید.

www.Abiri.ir

راه حل : اولین، دومین و چهارمین بایت، مطابق با مثال پیش هستند. برای پیدا کردن بایت سوم، 84 را به صورت حاصل جمع توانهای 2 می‌نویسیم و فقط 4 بیت سمت چپ (m=20-2*8=4) را مطابق با شکل ۵-۴ نگه می‌داریم. اولین آدرس برابر با 140.120.80.0 / 20 می‌شود.



شکل ۵-۴ : مثال ۶

* پیدا کردن تعداد آدرسها در بلاک

پیدا کردن تعداد آدرسها در یک بلاک آدرس، کار ساده‌ای است. کل تعداد آدرسها در یک بلاک آدرس برابر با 2^{32-n} است.

** مثال ۷

اگر یکی از آدرسها برابر با 140.120.84.24 / 20 باشد، تعداد آدرسهای بلاک را بدست آورید.

راه حل : طول پیشوند برابر با 20 است. تعداد آدرسهای بلاک برابر با 2^{32-20} و یا 4096 است. توجه نمائید که این، یک بلاک بزرگ آدرس با 4096 آدرس است.

* پیدا کردن آخرین آدرس در بلاک

دو متد برای پیدا کردن آخرین آدرس در یک بلاک آدرس، ارائه می‌دهیم. در اولین متد، یکی کمتر از تعداد آدرسهای بلاک را با اولین آدرس بلاک جمع می‌کنیم تا آخرین آدرس بلاک بدست آید. دقت نمائید که یکی از تعداد آدرسها کم می‌کنیم زیرا که اولین و آخرین آدرس، در شمارش تعداد آدرسها وارد می‌شوند. در روش دوم، اولین آدرس را با مکمل مسک جمع می‌کنیم. مکمل مسک را با تغییر 0 ها به 1 و 1 ها به 0 بدست می‌آوریم.

www.Abiri.ir

مثال ۸

با استفاده از روش اول، اگر یکی از آدرسهای بلاک برابر با 140.120.84.24 / 20 باشد، آخرین آدرس بلاک آدرس را بدست آورید.

راه حل : در مثال قبل دیدیم که اولین آدرس برابر با 140.120.80.0 / 20 و تعداد آدرسها برابر با 4096 است. برای پیدا کردن آخرین آدرس، باید 4095=1-4096 را با اولین آدرس، جمع کنیم. برای رعایت نماد اعداد صحیح نقطه دار، باید 4095 را در مبنای 256 نمایش دهیم و محاسبات را در مبنای 256 انجام دهیم. 4095 را به صورت 15.255 می‌نویسیم و سپس مطابق با آنچه که در ذیل می‌آید، اولین آدرس را با این عدد (در مبنای 256) جمع می‌کنیم تا آخرین آدرس را بدست آوریم.

```
140 .120 .80 .0
      15 .255
---------------------
140. 120. 95. 255
```

آخرین آدرس برابر با 140. 120. 95. 255 / 29 می‌باشد.

مثال ۹

با استفاده از روش دوم، آخرین آدرس بلاک را مشروط بر آنکه یکی از آدرسها برابر با 140.120.84.24 / 20 باشد، بدست آورید.

راه حل : مسک حاوی بیست بیت 1 و دوازده بیت 0 است. مکمل مسک حاوی بیست بیت 0 و دوازده بیت 1 است. به عبارت دیگر مکمل مسک برابر است با:

00000000 00000000 00001111 11111111

می‌توانیم آن را به صورت 0.0.15.255 بنویسیم. برای بدست آوردن آخرین آدرس، مکمل مسک را با آدرس شروع، جمع می‌کنیم.

```
140. 120.  80.  0
  0.   0.  15.  255
----------------------
140. 120.  95.  255
```

آخرین آدرس برابر با 140.120.95.255 / 20 می‌باشد.

مثال ۱۰

اگر یکی از آدرسها برابر با 190.87.140.202 / 29 باشد، بلاک آدرس را پیدا کنید.

www.Abiri.ir

راه حل : برای پیدا کردن اولین آدرس، تعداد آدرسها و آخرین آدرس، از روش مثال قبل استفاده می‌کنیم. برای پیدا کردن اولین آدرس، دقت می‌کنیم که مسک (29 /) حاوی 5=8*3-29 یک، در آخرین بایت است. بنابراین آخرین بایت را به صورت توان 2 می‌نویسیم و مطابق آنچه در ذیل می‌آید، پنج توان 2 از سمت چپ را نگه می‌داریم:

202 → 128 + 64 + 0 + 0 + 8 + 0 + 2 + 0
پنج عدد سمت چپ → 128 + 64 + 0 + 0 + 8
اولین آدرس → 190.87.140.200 / 29

تعداد آدرسها برابر با $2^{32-29} = 8$ آدرس می‌باشد. برای پیدا کردن آخرین آدرس، از مکمل مسک استفاده می‌کنیم. مسک حاوی 29 یک است، بنابراین مکمل مسک حاوی 3=29-32 یک است. مکمل مسک برابر با 0.0.0.7 می‌باشد. اگر این مقدار را با اولین آدرس جمع کنیم، به 190.87.140.207 / 29 می‌رسیم. به عبارت دیگر اولین آدرس، 190.87.140.200 / 29 و آخرین آدرس 190.87.140.207 / 29 می‌باشد. فقط 8 آدرس در این بلاک وجود دارد.

❊❊ مثال ۱۱

برای بلاک مثال قبل، یک شبکه ترسیم نمایید.

راه حل : سازمانی که به او، بلاک آدرس مثال قبل داده شده است، می‌تواند آدرسهای بلاک را به کامپیوترهای شبکه خود اختصاص دهد. با این حال، اولین آدرس، آدرس شبکه و آخرین آدرس، ***آدرس همگانی محدود***[1] می‌باشند. شکل ۵-۵ نشان می‌دهد که چگونه می‌توان از بلاک آدرس در سازمان استفاده کرد. دقت نمایید که بر خلاف آدرس‌دهی کلاس‌دار که در آن آخرین آدرس به 255 ختم می‌شود، در اینجا آخرین آدرس به 207 ختم می‌شود.

مهم: در آدرس‌دهی کلاس‌دار، لزومی ندارد که آخرین آدرس بلاک به 255 ختم شود.

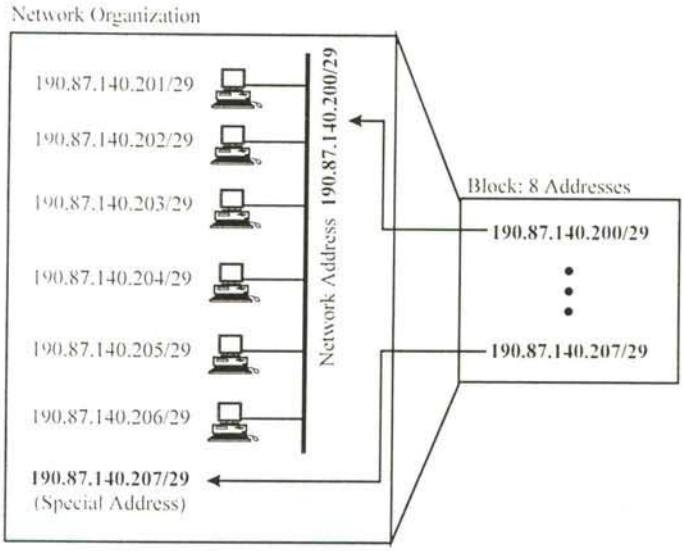
❊ بلاک آدرس

همانطور که می‌دانیم، بلاک آدرسها توسط یک ISP به متقاضی داده می‌شود. بلاک آدرس داده شده، توسط اولین آدرس و طول پیشوند تعریف می‌شود. همانطور که دیدیم، این

1- limited broadcast address

www.Abiri.ir

نماد CIDR بطور کامل، بلاک آدرس را مشخص می‌کند. مثلاً در مثال قبل، بلاک آدرس به صورت 190.87.140.200 /29 تعریف می‌شود.



شکل ۵-۵ : مثال ۱۱

مهم: در نماد CIDR ، بلاک آدرس داده شده توسط اولین آدرس و طول پیشوند تعیین می‌شود.

۵-۲ زیرشبکه‌سازی

البته می‌توانیم با آدرس‌دهی بدون‌کلاس، ***زیرشبکه‌سازی***[1] نیز داشته باشیم. وقتی که به سازمانی یک بلاک آدرس داده می‌شود، او می‌تواند برای برآوردن نیازهای خود، زیرشبکه‌سازی کند. مدیر شبکه می‌تواند برای هر زیرشبکه، یک **مسک زیرشبکه**[2] بسازد.

پیدا کردن مسک زیرشبکه

تعداد زیرشبکه‌های مطلوب، پیشوند زیرشبکه را تعیین می‌کند. اگر تعداد زیرشبکه‌ها برابر با S باشد، تعداد یکهای اضافه در پیشوند، برابر با لگاریتم S در مبنای 2 می‌باشد و لذا S برابر است با 2 به توان تعداد 1 های اضافه در پیشوند. دقت نمایید که اگر

1- Subnetting
2- Subnet mask

www.Abiri.ir

زیرشبکه‌هایی به طول ثابت بخواهیم (تعداد آدرسهای زیرشبکه‌ها، یکسان باشد)، تعداد زیرشبکه‌ها باید توانی از 2 باشد.

مهم: در زیرشبکه‌سازی که طول زیرشبکه‌ها، یکسان است، تعداد زیرشبکه‌ها باید توانی از 2 باشد.

*** مثال ۱۲**

بلاک 130.34.12.64 / 26 به سازمانی داده شده است. این سازمان به چهار زیرشبکه نیاز دارد. طول پیشوند زیرشبکه چیست؟

راه حل : به چهار زیرشبکه نیاز داریم و معنی آن این است که باید دو تا یک بیشتر (لگاریتم 4 در مبنای 2 ، برابر با 2 می‌باشد) به پیشوند اضافه کنیم. بنابراین پیشوند زیرشبکه برابر با 28/ می‌شود.

پیدا کردن آدرسهای زیرشبکه‌ها

بعد از پیدا کردن مسک زیرشبکه، پیدا کردن آدرسهای هر زیرشبکه کار ساده‌ای است.

*** مثال ۱۳**

برای هر یک از زیرشبکه‌های مثال قبل، آدرسهای زیرشبکه و محدوده آدرسها چیست؟

راه حل : شکل ۵-۶ یک پیکربندی را برای شبکه نشان می‌دهد.

این سایت حاوی $2^{32-26}=64$ آدرس است و هر زیرشبکه حاوی $2^{32-28}=16$ آدرس است. حال اجازه دهید، اولین و آخرین آدرس هر زیرشبکه را پیدا کنیم.

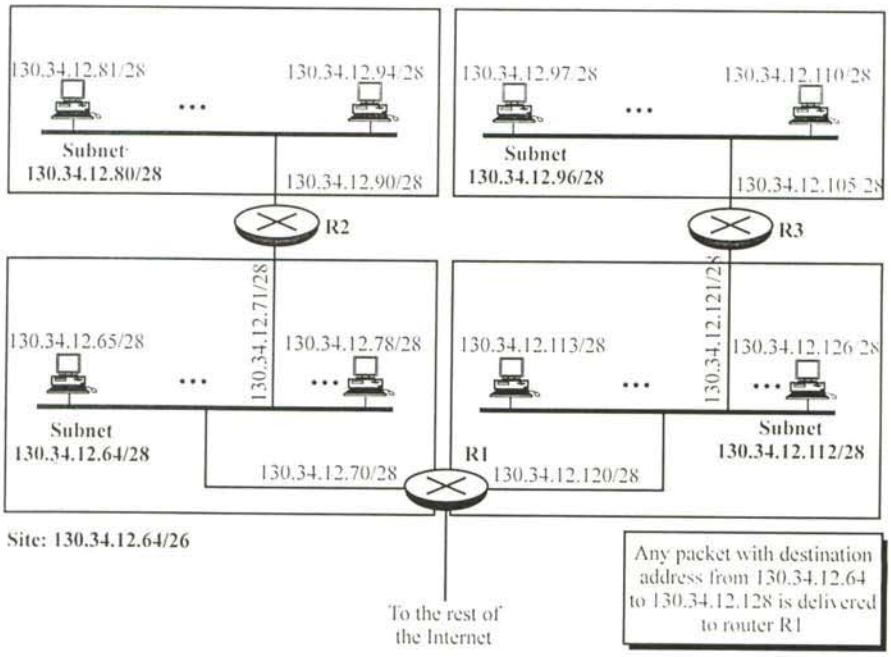
۱- با استفاده از روشی که در مثالهای قبل نشان دادیم، اولین آدرس در اولین زیرشبکه برابر با 130.34.12.64 / 28 می‌باشد. دقت نمائید که اولین آدرس اولین زیرشبکه، برابر با اولین آدرس بلاک است. آخرین آدرس زیرشبکه را می‌توان با اضافه کردن 15 (1-16) به اولین آدرس بدست آورد. آخرین آدرس برابر با 130.34.12.79 / 28 می‌باشد.

۲- اولین آدرس در زیرشبکه دوم برابر با 130.34.12.80 / 28 می‌باشد. آن را با اضافه کردن 1 به آخرین آدرس زیرشبکه قبلی بدست آورده‌ایم. مجدداً با اضافه کردن 15 به اولین آدرس، آخرین آدرس برابر با 130.34.12.95 / 28 را بدست می‌آوریم.

۳- بطور مشابه، اولین آدرس زیرشبکه سوم را برابر با 130.34.12.96 / 28 و آخرین آدرس را برابر با 130.34.12.111 / 28 بدست می‌آوریم.

www.Abiri.ir

۴- بطور مشابه، اولین آدرس زیرشبکه چهارم را برابر با 130.34.12.112 / 28 و آخرین آدرس را برابر با 130.34.12.127 / 28 بدست می‌آوریم.



شکل ۵-۶ : مثال ۱۳

زیرشبکه‌های با طول متغیر

در مثال قبل، همه زیرشبکه‌های ما، مسک مشابه (طول n مشابه) داشتند. همچنین می‌توانیم زیرشبکه‌هایی بسازیم که مسکهای با طول متغیر داشته باشند. به عبارت دیگر می‌توان زیرشبکه‌هایی با سایز متفاوت طراحی کرد. به این ترتیب می‌توان به یک سازمان، بر اساس نیاز زیرشبکه‌هایش، آدرس اختصاص داد. این روش مشابه آن چیزی است که قبلاً بحث آن آمد. مفهوم آن را با یک مثال نشان می‌دهیم.

* مثال ۱۴

به سازمانی یک بلاک آدرس، با آدرس شروع 14.24.74.0 / 24 داده شده است. این بلاک حاوی $2^{32-24}=256$ آدرس است. این سازمان به یازده زیرشبکه، مطابق مشخصات زیر نیاز دارد:

الف – دو زیرشبکه، هر کدام حاوی 64 آدرس

www.Abiri.ir

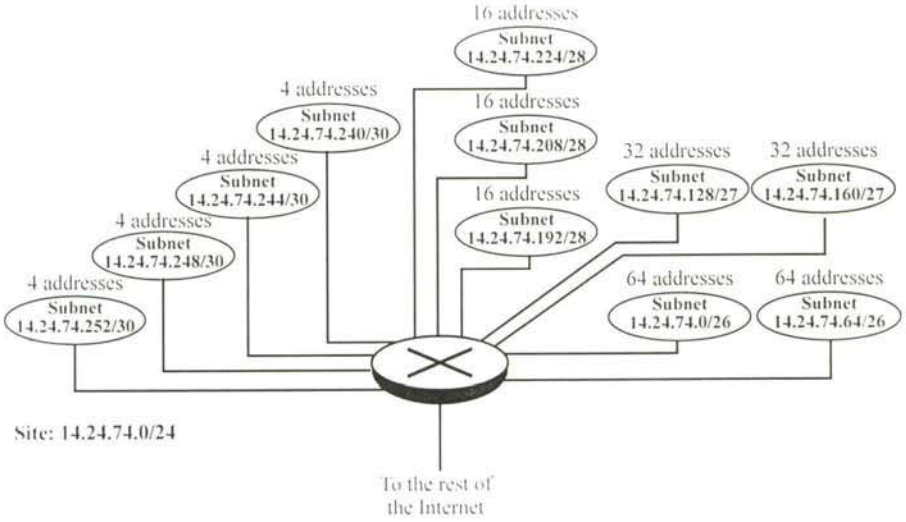
ب – دو زیرشبکه، هر کدام حاوی 32 آدرس

ج – سه زیرشبکه، هر کدام حاوی 16 آدرس

د – چهار زیرشبکه، هر کدام حاوی 4 آدرس

زیرشبکه‌ها را طراحی نمائید.

راه حل : شکل ۵-۷ یک پیکربندی را برای این شبکه نشان می‌دهد.



شکل ۵-۷ : مثال ۱۴

۱- از اولین 128 آدرس برای اولین دو زیرشبکه، هر یک حاوی 64 آدرس استفاده می‌کنیم. دقت نمائید که مسک هر زیرشبکه 26 / است(32-6=26 , $64=2^6$). آدرس پایه هر زیرشبکه، در شکل ۵-۷ داده شده است.

۲- از 64 آدرس بعدی برای دو زیرشبکه بعدی، هر یک حاوی 32 آدرس استفاده می‌کنیم. دقت نمائید که مسک هر زیر شبکه، 27 / است (32-5=27 , $32=2^5$). آدرس پایه هر زیرشبکه، در شکل ۵-۷ داده شده است.

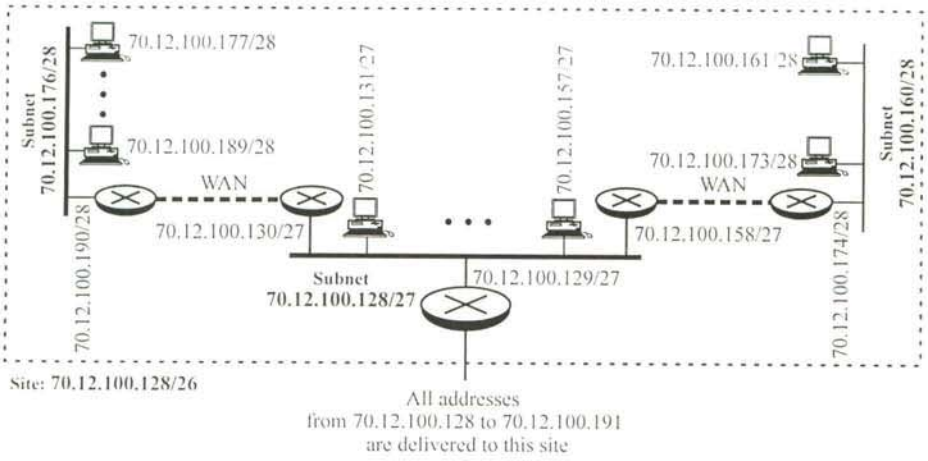
۳- از 48 آدرس بعدی، برای سه زیرشبکه بعدی هر یک حاوی 16 آدرس، استفاده می‌کنیم. دقت نمائید که مسک هر زیرشبکه، 28 / است (32-4=28 , $16=2^4$) آدرس پایه هر زیرشبکه، در شکل ۵-۷ داده شده است.

۴- از آخرین 16 آدرس برای آخرین چهار زیرشبکه، هر یک حاوی 4 آدرس استفاده می کنیم. دقت نمائید که مسک هر زیرشبکه برابر با 30 / است (32-2=30 , $4=2^2$). آدرس پایه هر زیرشبکه، در شکل ۵-۷ داده شده است.

www.Abiri.ir

*** مثال ۱۵**

به عنوان یک مثال دیگر، فرض کنید شرکتی دارای سه شعبه است: مرکزی، شرقی و غربی. شعبه مرکزی از طریق لینکهای WAN نقطه به نقطه خصوصی، به شعب شرقی و غربی متصل شده است. به این شرکت، یک بلاک آدرس حاوی 64 آدرس با آدرس شروع 70.12.100.128 / 26 داده شده است. مدیریت تصمیم گرفته است که 32 آدرس را به شعبه مرکزی بدهد و بقیه آدرسها را بین دو شعبه دیگر تقسیم کند. شکل ۵-۸ پیکربندی طراحی شده توسط مدیریت را نشان می‌دهد.



شکل ۵-۸ : مثال ۱۵

این شرکت، سه زیرشبکه دارد: اولی در مرکز، دومی در شرق و سومی در غرب. لیست زیر بلاکهایی که به هر زیرشبکه داده شده است، به دنبال می‌آید:

الف- شعبه مرکزی از آدرس شبکه 70.12.100.128 / 27 استفاده می‌کند. این اولین آدرس است و مسک 27 / نشان می‌دهد که این زیرشبکه حاوی 32 آدرس است($2^5=32$, 32-27=5) دقت نمائید که سه تا از این آدرسها برای روترها به کار می‌روند و شرکت، آخرین آدرس زیر بلاک را رزرو کرده است. آدرسهای این زیرشبکه از 70.12.100.128 / 27 تا 70.12.100.159 / 27 هستند(128+32-1=159). توجه نمائید که آن اینترفیسی از روتر که زیرشبکه مرکزی را به WAN متصل می‌کند، نیازی به آدرس ندارد زیرا که یک لینک نقطه به نقطه است.

ب – شعبه غربی از آدرس زیرشبکه 70.12.100.160 / 28 استفاده می‌کند. مسک 28 / نشان می‌دهد که فقط 16 آدرس در این زیرشبکه وجود دارد ($2^4=16$, 32-28=4). دقت نمائید که یکی از این آدرسها برای روتر به کار می‌رود و شرکت، آخرین آدرس موجود در

www.Abiri.ir

زیربلاک را رزرو کرده است. آدرسهای این زیرشبکه از 70.12.100.160 / 28 تا 70.12.100.175 / 28 هستند (160+16-1=175). توجه نمائید که آن اینترفیسی از روتر که زیرشبکه غربی را به WAN متصل می‌کند، نیازی به آدرس ندارد زیرا یک اتصال نقطه به نقطه است.

ج- شعبه شرقی از آدرس زیرشبکه 70.12.100.176 / 28 استفاده می‌کند. مسک 28 / نشان می‌دهد که فقط 16 آدرس در این زیرشبکه وجود دارد. دقت نمائید که یکی از این آدرسها برای روتر به کار می‌رود و شرکت، آخرین آدرس زیر بلاک را رزرو کرده است. آدرسهای این زیرشبکه از 70.12.100.176 / 28 تا 70.12.100.191 / 28 هستند (191=1-16+176). توجه نمائید که آن اینترفیسی از روتر که زیرشبکه شرقی را به WAN متصل می‌کند، نیازی به آدرس ندارد زیرا یک اتصال نقطه به نقطه است.

۵-۳ تخصیص آدرسها

مطلب بعدی در آدرس‌دهی بدون‌کلاس، تخصیص آدرسهاست. چگونه بلاک آدرسها به سازمانها اختصاص داده می‌شوند؟ مسئولیت نهائی تخصیص آدرس، به یک سازمان مسئول جهانی به نام ***شرکت اینترنتی برای اعداد و اسامی تخصیص یافته***[1] داده شده است. با این وجود، معمولاً ICANN آدرسها را به سازمانهای منفرد نمی‌دهد. او یک بلاک بزرگ آدرس را به یک ISP می‌دهد. سپس هر ISP به نوبه خود، بلاک آدرس دریافتی را به بلاکهای کوچکتری تقسیم می‌کند و زیربلاکها را به متقاضیان خود می‌دهد. به عبارت دیگر، یک ISP یک بلاک بزرگ آدرس دریافت می‌کند و آن را میان کاربران اینترنتی خود تقسیم می‌کند. این عمل به نام ***انبوه آدرس***[2] موسوم است. تعداد زیادی از بلاکهای آدرس به صورت یک بلاک بزرگ آدرس درمی‌آیند و به یک ISP داده می‌شوند.

مثال ۱۶

به یک ISP یک بلاک آدرس که با 190.100.0.0 / 16 (65536 آدرس) شروع می‌شود، داده شده است. ISP این آدرس را به صورت زیر بین گروههای مختلف مشترکین خود تقسیم می‌کند.

الف – اولین گروه، 64 مشترک دارد؛ هر مشترک به 256 آدرس نیاز دارد.

ب – دومین گروه، 128 مشترک دارد؛ هر مشترک به 128 آدرس نیاز دارد.

ج – سومین گروه، 128 مشترک دارد؛ هر مشترک به 64 آدرس نیاز دارد.

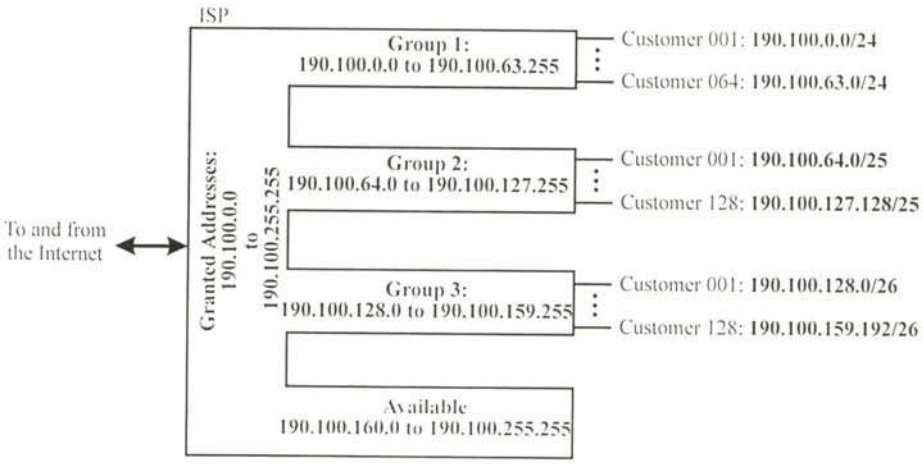
1- Internet Corporation for Assigned Names and Numbers : ICANN

2- Address aggregation

www.Abiri.ir

زیرشبکه‌ها را طراحی کنید و محاسبه کنید که بعد از این تخصیص آدرسها، هنوز چند آدرس باقی می‌ماند.

راه حل : شکل ۵-۹ وضعیت را نشان می‌دهد.



شکل ۵-۹ : مثال ۱۶

۱- گروه ۱

در این گروه، هر مشترک به 256 آدرس نیاز دارد. معنی آن این است که طول پسوند برابر با 8 (2^8=256) است. بنابراین طول پیشوند برابر با 24=8-32 می‌باشد. آدرسها عبارتند از:

1st Customer 190.100.0.0 / 24 190.100.0.255 / 24
2nd Customer 190.100.1.0 / 24 190.100.1.255 / 24

.
.
.

64th Customer 190.100.63.0 / 24 190.100.63.255 / 24
Total = 64 * 256 = 16384

۲ – گروه ۲

برای این گروه، هر مشترک به 128 آدرس نیاز دارد. معنی آن این است که طول پسوند برابر با 7 (2^7=128) است. بنابراین طول پیشوند برابر با 25=7-32 می‌باشد. آدرسها عبارتند از:

1st Customer 190.100.64.0 / 25 190.100.64.127 / 25
2nd Customer 190.100.64.128 / 25 190.100.64.255 / 25

.
.
.

128th Customer 190.100.127.128 / 25 190.100.127.255 / 25

www.Abfr.ir

Total = 128 * 128 = 16384

۳ - گروه ۳

برای این گروه، هر مشترک به 64 آدرس نیاز دارد. معنی آن این است که طول پسوند برابر با 6 ($2^6 = 64$) است. بنابراین طول پیشوند برابر با 26=6-32 است. آدرسها عبارتند از:

1st Customer 190.100.128.0 / 26 190.100.128.63 / 26
2nd Customer 190.100.128.64 / 26 190.100.128.127 / 26
.
.
.
128th Customer 190.100.159.192 / 26 190.100.159.255 / 26
Total = 128 * 64 = 8192

تعداد آدرسهای داده شده به ISP برابر است با : 65536
تعداد آدرسهای اختصاص یافته به گروهها توسط ISP : 40960
تعداد آدرسهای باقیمانده : 24576

۵-۴ گراف زیرشبکه‌سازی، ابداع مترجم

همواره داشتن نمایش دقیقی از نحوه تخصیص آدرسهای IP ، می‌تواند کمک شایان توجهی به طراح شبکه محسوب گردد. اگر این نمایش به صورت یک گراف باشد، آدرسهای تخصیص یافته و آدرسهای باقیمانده، همواره قابل رؤیت می‌باشند و لذا مدیریت آنها، بسیار ساده‌تر می‌شود. به این منظور، گرافی ابداع نموده‌ام که نحوه به کارگیری آن را از طریق مثالهایی توضیح می‌دهم.

مثال ۱۷

محدوده آدرسهای IP زیر را داریم:

آدرس پایه شبکه : 10.0.0.0
مسک : 255.255.255.192

چگونه می‌توانیم این محدوده از آدرسها را به 7 زیرشبکه به گونه‌ای تقسیم کنیم که یک زیرشبکه با 16 آدرس و یک زیرشبکه دیگر با 8 آدرس، جزء زیرشبکه‌های ما باشند؟ تعداد آدرسها در سایر زیرشبکه‌ها را به دلخواه خود انتخاب نمایید.

راه حل: قدم به قدم پیش می‌رویم:

قدم ۱- کلاً چند تا آدرس داریم؟

www.Abiri.ir

برای پیدا کردن تعداد آدرسهای موجود، باید عدد 2 را به توان تعداد صفرهای سمت راست مسک برسانیم. اگر مسک را به صورت باینری بنویسیم، به صورت زیر درمی‌آید:

11111111.11111111.11111111.11000000

از آنجا که 6 صفر در سمت راست مسک داریم، لذا کل آدرسهای موجود برابر با $2^6 = 64$ آدرس است.

قدم ۲- آیا مسک داده شده با آدرس پایه داده شده، همخوانی دارند؟

زمانی همخوانی بین مسک و آدرس پایه وجود دارد که تعداد صفرهای سمت راست آدرس پایه ، حداقل به تعداد صفرهای سمت راست مسک باشند. در مثال ما، به تعداد 6 صفر در سمت راست مسک داریم، بنابراین شرط همخوانی مسک و آدرس پایه این است که در سمت راست آدرس پایه، حداقل 6 صفر داشته باشیم. معادل باینری آدرس پایه داده شده به صورت زیر است:

00001010.00000000.00000000.00000000

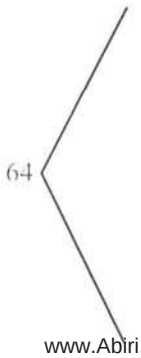
مشاهده می‌نمایید که در سمت راست آدرس پایه، به تعداد 25 صفر داریم و این تعداد از حداقل 6 صفر مورد نیاز، بیشتر است، بنابراین شرط همخوانی آدرس پایه و مسک، برقرار است.

قدم ۳- مرحله به مرحله، گراف زیرشبکه‌سازی را ترسیم می‌نماییم.

گفتیم که به تعداد 64 آدرس داریم. عدد 64 را می‌نویسیم:

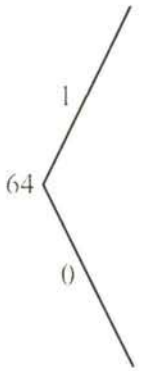
64

سپس دو تکه خط را به گونه‌ای می‌کشیم که رأس آنها، روی عدد 64 باشد:

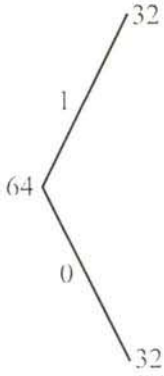


www.Abiri.ir

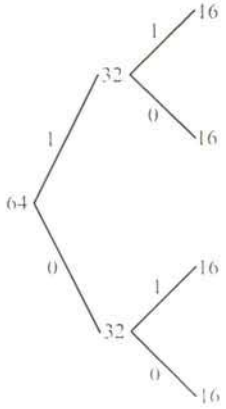
حالا عدد 1 را روی تکه خط بالایی و عدد 0 را روی تکه خط پایینی می‌نویسیم:



یعنی تعداد 64 آدرس موجود را به دو زیرشبکه تقسیم کردیم و هر زیرشبکه، حاوی نیمی از آدرسهای موجود، یعنی 32 آدرس می‌باشد:

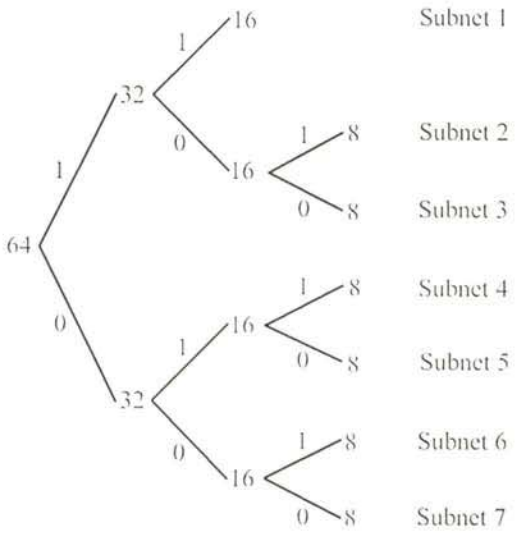


از آنجا که به 7 زیرشبکه نیاز داریم، پس باید به تقسیم زیرشبکه‌ها ادامه دهیم و لذا هر یک از زیرشبکه‌های گراف فوق را به دو زیرشبکه تقسیم می‌نماییم:



www.Abiri.ir

بر اساس خواست مسئله، یک زیرشبکه نیاز داریم که حاوی 16 آدرس باشد. بنابراین یکی از زیرشبکه‌های 16تایی را کنار می‌گذاریم. اگر هر یک از زیرشبکه‌های باقیمانده را به دو زیرشبکه تقسیم کنیم، جمعاً 7 زیرشبکه خواهیم داشت:



شکل ۵-۱۰ : مثال ۱۷

حالا 7 زیرشبکه داریم که در آن زیرشبکه شماره 1 حاوی 16 آدرس، زیرشبکه شماره 2 حاوی 8 آدرس و بقیه زیرشبکه‌ها نیز هر یک حاوی 8 آدرس می‌باشند و این همان چیزی است که مسئله از ما می‌خواست.

قدم ۴- اکنون می‌دانیم که هر زیرشبکه حاوی چند آدرس است، اما نمی‌دانیم که آدرس پایه هر زیرشبکه چیست. بنابراین در این مرحله باید آدرس پایه هر زیرشبکه را پیدا کنیم. از آنجا که در سمت راست مسک، به تعداد 6 صفر داریم، آدرس پایه را به صورت زیر می‌نویسیم:

10.0.0.00 | 000000

طبق شرط همخوانی آدرس پایه و مسک، مطمئن هستیم که در سمت راست آدرس پایه به تعداد 6 عدد صفر داریم. این 6 عدد صفر را توسط یک میله (|) از بقیه آدرس، جدا کردیم زیرا که تمام کار ما با زیرشبکه‌سازی، به همین 6 صفر سمت راست آدرس پایه مربوط می‌شود. ارقامی که سمت چپ میله قرار می‌گیرند، در تمام زیرشبکه‌ها، مقدار یکسانی خواهند

www.Abiri.ir

داشت. حالا به گراف شکل ۵-۱۰ مراجعه می‌کنیم. از Subnet 1 شروع می‌کنیم. از آخرین گره مربوط به این زیرشبکه که عدد 16 را نشان می‌دهد، تا مبدأ گراف که عدد 64 است، به ترتیب، اعداد 1 و 1 روی تکه‌خطها قرار گرفته‌اند. دو 1 را روی گراف، از راست به چپ خواندیم، حالا آنها را به همان ترتیب از راست به چپ، در سمت راست میله، قرار می‌دهیم.

Subnet 1 : 10.0.0.00 | 11
←

همانطور که گراف شکل ۵-۱۰ نشان می‌دهد، اعدادی که روی تکه خطهای واصل از گره مربوط به Subnet 2 تا مبدأ گراف قرار گرفته‌اند، به ترتیب، اعداد 1 0 1 می‌باشند. بنابراین برای زیرشبکه شماره 2 داریم:

Subnet 1 : 10.0.0.00 | 101
←

اگر این کار را برای هر 7 زیرشبکه گراف شکل ۵-۱۰ تکرار کنیم، نتیجه به صورت زیر خواهد بود:

Subnet 1 : 10.0.0.00 | 11
Subnet 2 : 10.0.0.00 | 101
Subnet 3 : 10.0.0.00 | 100
Subnet 4 : 10.0.0.00 | 011
Subnet 5 : 10.0.0.00 | 010
Subnet 6 : 10.0.0.00 | 001
Subnet 7 : 10.0.0.00 | 000

یک آدرس IP حاوی 32 بیت است، بنابراین برای کامل شدن آدرس پایه، باید آنقدر صفر جلو آدرس‌های پایه فوق بگذاریم، که کل تعداد بیتها در هر آدرس پایه، به 32 بیت برسد. می‌دانیم که از ابتدا به تعداد 6 صفر در سمت راست میله داشتیم، پس حالا که دو رقم در سمت راست میله مربوط به آدرس پایه Subnet 1 قرار گرفته است، پس با گذاشتن 4=2-6 بیت 0 در انتهای آدرس پایه مربوط به Subnet 1 ، کل تعداد بیتهای آن، به 32 بیت می‌رسد. در مورد بقیه آدرسهای پایه، چون سه رقم در سمت راست میله آدرس پایه آنها قرار گرفته است، بنابراین، با گذاشتن 3=3-6 بیت 0 در انتهای آنها، به آدرسهای پایه کاملی می‌رسیم. نتیجه به صورت زیر است :

Subnet 1 : 10.0.0.00 | 11**0000**
Subnet 2 : 10.0.0.00 | 101**000**
Subnet 3 : 10.0.0.00 | 100**000**
Subnet 4 : 10.0.0.00 | 011**000**
Subnet 5 : 10.0.0.00 | 010**000**
Subnet 6 : 10.0.0.00 | 001**000**
Subnet 7 : 10.0.0.00 | 000**000**

www.Abiri.ir

بیتهای 0 اضافه‌ای که آدرس را به طول 32 بیت می‌رسانند، به صورت ضخیم نشان داده شده‌اند تا بتوانید آنها را از بیتهایی که از گراف خوانده‌اید، به راحتی تشخیص دهید. وقتی که این محاسبات را به صورت دستی انجام می‌دهید، بهتر است از رنگ دیگری برای نشان دادن این بیتهای 0 اضافه، استفاده کنید تا همواره از بیتهایی که از گراف می‌خوانید، قابل تمیز باشند. حالا کافیست که معادل اعشاری آدرسهای پایه فوق را بنویسیم که به صورت زیر درمی‌آید:

Subnet 1 : 10.0.0.48
Subnet 2 : 10.0.0.40
Subnet 3 : 10.0.0.32
Subnet 4 : 10.0.0.24
Subnet 5 : 10.0.0.16
Subnet 6 : 10.0.0.8
Subnet 7 : 10.0.0.0

قدم ۵- حالا باید مسک مربوط به هر زیرشبکه را بدست آوریم. مسک اولیه برابر با 255.255.255.192 بود که معادل باینری آن را می‌نویسیم و 6 بیت سمت راست آن را با یک میله، جدا می کنیم.

255.255.255.11 | 000000

لزومی ندارد که به جای 255 ها، 8 بیت یک بگذاریم زیرا آنچه که مهم است، بیتهای سمت راست میله می‌باشند. معادل باینری آدرس پایه Subnet 1 به صورت زیر بود:

10.0.0.00 | 11**0000**

سمت راست میله نشان می‌دهد که دو بیت را از گراف خوانده‌ایم (بقیه، 0 های اضافه هستند). بنابراین در سمت راست میله، در مسکی که نوشتیم، دو بیت 1 می‌گذاریم و بقیه را با 0 های اضافه، پر می‌کنیم. نتیجه به صورت زیر خواهد شد:

Subnet 1: 255.255.255.11 | 11**0000**

در مورد سایر زیرشبکه‌ها، سه بیت را از گراف خوانده‌ایم، پس به تعداد سه بیت 1 را در سمت راست میله، به جای 0 های مسک، قرار می‌دهیم. نتیجه به این صورت خواهد شد:

All other Subnets : 255.255.255.11 | 111**000**

بنابراین، معادل باینری مسکها برای 7 زیرشبکه، به صورت زیر درمی‌آید:

www.Abiri.ir

Subnet 1 : 255.255.255.11 | 110000
Subnet 2 : 255.255.255.11 | 111000
Subnet 3 : 255.255.255.11 | 111000
Subnet 4 : 255.255.255.11 | 111000
Subnet 5 : 255.255.255.11 | 111000
Subnet 6 : 255.255.255.11 | 111000
Subnet 7 : 255.255.255.11 | 111000

و معادل اعشاری آنها، برابر است با:

Subnet 1 : 255.255.255.240
Subnet 2 : 255.255.255.248
Subnet 3 : 255.255.255.248
Subnet 4 : 255.255.255.248
Subnet 5 : 255.255.255.248
Subnet 6 : 255.255.255.248
Subnet 7 : 255.255.255.248

قدم ۶ : اکنون می‌توانیم، هر زیرشبکه را با نوشتن آدرس پایه و مسک آن، کاملاً مشخص نماییم:

Subnet 1: 10.0.0.48
255.255.255.240

Subnet 2: 10.0.0.40
255.255.255.248

Subnet 3: 10.0.0.32
255.255.255.248

Subnet 4: 10.0.0.24
255.255.255.248

Subnet 5: 10.0.0.16
255.255.255.248

Subnet 6: 10.0.0.8
255.255.255.248

Subnet 7: 10.0.0.0
255.255.255.248

www.Abiri.ir

مثال ۱۸

در مثال ۱۷ دیدیم که زیرشبکه شماره 7 ، به تعداد 8 آدرس دارد. از این 8 آدرس، کدامیک را می‌توانیم به کارت شبکه کامپیوترهایی اختصاص دهیم که در این زیرشبکه داریم؟

راه حل : در مثال ۱۷ دیدیم که محدوده آدرسهای زیرشبکه شماره 17 به صورت زیر است:

Subnet 7: 10.0.0.0
255.255.255.248

همچنین می‌دانیم که این زیرشبکه حاوی 8 آدرس است اما اگر این را از مثال ۱۷ نمی‌دانستیم و می‌خواستیم با محاسبه بدست آوریم، ابتدا معادل باینری مسک را می‌نوشتیم:

255.255.255.11111000

حالا تعداد 0 های سمت راست مسک را می‌شماریم که 3 تاست. بنابراین در این زیرشبکه به تعداد $2^3=8$ آدرس داریم. می‌خواهیم همه این آدرسها را بنویسیم. اولین آدرس، همان آدرس پایه یعنی 10.0.0.0 است. به این آدرس پایه عدد 1 را اضافه می‌کنیم، آدرس بعدی به صورت 10.0.0.1 بدست می‌آید. همین کار را آنقدر ادامه می‌دهیم تا بقیه آدرسها نیز به دست آیند. نتیجه به صورت زیر است:

10.0.0.0
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.5
10.0.0.6
10.0.0.7

بنابراین حسن استفاده از مسک این است که یک محدوده از آدرسهای پیوسته را به ما می‌دهد که از یک آدرس پایه شروع می‌شوند. همانطور که ملاحظه می‌کنید، این 8 آدرس، یکی پس از دیگری است و هیچ فاصله خالی بین آنها وجود ندارد و دلیل نشان دادن یک محدوده از آدرسها به کمک آدرس پایه و مسک نیز همین است.

از این میان، اولین آدرس یعنی 10.0.0.0 همان آدرس پایه است که نمی‌توان آن را به کارت شبکه یا اینترفیس روترها اختصاص داد زیرا که این آدرس، معرف کل زیرشبکه است. آخرین آدرس از این محدوده یعنی 10.0.0.7 همان آدرس همگانی محدود است. این آدرس را نیز نمی‌توان به کارت شبکه یا اینترفیس روترها اختصاص داد اما اگر بسته‌ای به این آدرس

www.Abiri.ir

مقصد، به این زیرشبکه برسد، آن بسته به کلیه کامپیوترها و روترهای موجود در این زیرشبکه، تحویل داده خواهد شد و این همان مفهوم آدرس همگانی محدود است.

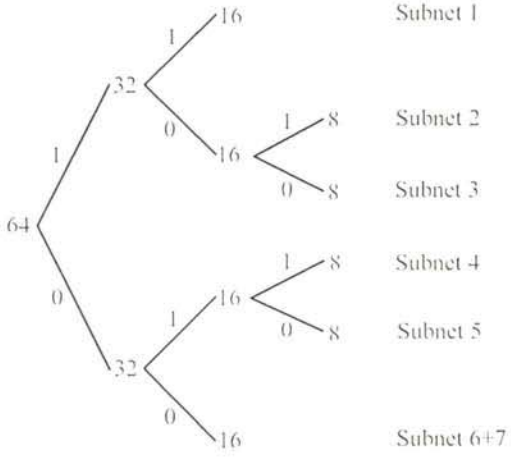
بنابراین می‌توانیم از 6=2-8 آدرس باقیمانده زیر، برای اختصاص دادن به کارت شبکه کامپیوترها و یا روترهای موجود در این زیرشبکه استفاده کنیم:

10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.4
10.0.0.5
10.0.0.6

مثال ۱۹

در مثال ۱۷ دیدیم که فقط می‌توانیم به 6 کامپیوتر، آدرس IP بدهیم. حال اگر در زیرشبکه شماره 7 ، به سه کامپیوتر دیگر نیاز داشته باشیم، چطور می‌توانیم به این سه کامپیوتر، آدرس IP بدهیم؟ فرض کنید که آدرسهای زیرشبکه شماره 6 به صورت رزرو هستند و هنوز استفاده‌ای از آنها نکرده‌ایم.

راه حل: به گراف شکل ۵-۱۰ مراجعه می‌کنیم. حسن این گراف این است که دقیقاً نشان می‌دهد که تاکنون، چگونه از آدرسهای IP استفاده کرده‌ایم. همانطور که صورت مسئله می‌گوید، از آدرسهای زیرشبکه شماره 6 استفاده‌ای نکرده‌ایم بنابراین می‌توانیم زیرشبکه‌های 6 و 7 را در هم ادغام کنیم و یک زیرشبکه بزرگتر بسازیم. اگر نام این زیرشبکه جدید را Subnet 6+7 بگذاریم، گراف جدید به صورت شکل ۵-۱۱ در می‌آید:



شکل ۵-۱۱ : مثال ۱۹

www.Abiri.ir

همانطور که گراف نشان می‌دهد، اینک به جای دو زیرشبکه 6 و 7 ، هر یک حاوی 8 آدرس، یک زیرشبکه بزرگتر، حاوی 16 آدرس داریم. برای پیدا کردن آدرس پایه این زیرشبکه، مطابق مثال ۱۷ پیش می‌رویم. از آنجا که مبدأ گراف عدد 64 یعنی 2^6 را نشان می‌دهد، آدرس پایه اولیه که 10.0.0.0 بود را به صورت زیر می‌نویسیم:

10.0.0.00 | 000000

حال از گره مربوط به Subnet 6+7 شروع می‌کنیم و اعداد کنار تکه خطها را می‌خوانیم تا به مبدأ گراف برسیم. به ترتیب از راست به چپ، اعداد 0 و 0 بدست می‌آیند. این دو رقم را در سمت راست میله قرار می‌دهیم و بقیه را با 4=2-6 صفر اضافه، پر می‌کنیم:

10.0.0.00 | 00**0000**

و معادل اعشاری آن برابر با 10.0.0.0 می‌باشد.

برای بدست آوردن مسک، یک راه ساده‌تر را معرفی می‌کنیم. می‌دانیم که مسک، تعداد آدرسهای موجود در یک زیرشبکه را نشان می‌دهد. از آنجا که این زیرشبکه حاوی 16 آدرس است، پس باید در سمت راست مسک آن، به تعداد 4 (2^4=16) صفر داشته باشیم. بنابراین معادل باینری مسک به صورت زیر است :

255.255.255.11110000

و معادل اعشاری آن برابر با 255.255.255.240 می‌باشد.

بنابراین می‌توانیم زیرشبکه جدید را به صورت زیر نشان دهیم:

Subnet 6+7: 10.0.0.0
255.255.255.240

این محدوده از آدرسها را با محدوده آدرسهای زیرشبکه‌های شماره 6 و 7 در مثال ۱۷ مقایسه نمایید:

Subnet 6: 10.0.0.8
255.255.255.248

Subnet 7: 10.0.0.0
255.255.255.248

www.Abiri.ir

بنابراین می‌بینیم که آدرسهای از 10.0.0.0 تا 10.0.0.7 (جمعاً 8 آدرس) در Subnet 7 و آدرسهای از 10.0.0.8 تا 10.0.0.15 (جمعاً 8 آدرس) در Subnet 6 ، با هم ادغام شده‌اند و آدرسهای از 10.0.0.0 تا 10.0.0.15 (جمعاً 16 آدرس) در Subnet 6+7 را تولید کرده‌اند. از میان این آدرسها، اولی یعنی 10.0.0.0 ، آدرس پایه، و آخری یعنی 10.0.0.15 ، آدرس همگانی محدود است و لذا به تعداد 14=2-16 آدرس را می‌توانیم به کامپیوترهای موجود در این زیرشبکه اختصاص دهیم که از تعداد قبلی یعنی 6 آدرس، بیشتر است و این همان چیزی است که مسئله می‌خواست.

۵-۵ عبارات کلیدی

انبوه آدرس (address aggregation)

آدرس‌دهی بدون‌کلاس (Classless addressing)

پیشوند (Prefix)

طول پیشوند (Prefix length)

مسک زیرشبکه (subnet mask)

پسوند (Suffix)

طول پسوند (Suffix length)

زیرشبکه‌سازی با طول متغیر (variable length subnetting)

۵-۶ خلاصه

- در آدرس‌دهی بدون‌کلاس، می‌توانیم فضای آدرس را به بلاکهای با طول متغیر تقسیم کنیم.
- در آدرس‌دهی بدون‌کلاس، سه نوع محدودیت وجود دارد:

الف – تعداد آدرسها باید توانی از 2 باشد.

ب – برای تعریف یک بلاک آدرس، مسک نیز باید داده شود.

ج – آدرس شروع باید بر تعداد آدرسهای موجود در بلاک آدرس، قابل قسمت باشد.

- مسک در آدرس‌دهی بدون‌کلاس با نماد CIDR، به صورت طول پیشوند (/n) بیان می‌شود.
- برای پیدا کردن اولین آدرس یک بلاک، باید مسک را با یکی از آدرسها AND کنیم.
- برای پیدا کردن تعداد آدرسها در بلاک، 2^{32-n} را محاسبه می‌کنیم که در آن، n طول پیشوند است.

www.Abiri.ir

• برای پیدا کردن آخرین آدرس در بلاک، اولین آدرس را با تعداد آدرسها جمع می‌کنیم و سپس یک واحد کم می‌کنیم.

• می‌توانیم زیرشبکه‌سازی با طول ثابت و نیز با طول متغیر داشته باشیم. در زیرشبکه‌سازی با طول ثابت، تعداد آدرسها در زیر شبکه‌ها، یکسان است. در زیرشبکه‌سازی با طول متغیر، تعداد آدرسها در زیر شبکه‌های مختلف، متفاوت است.

• در زیرشبکه‌سازی با طول ثابت، تعداد زیرشبکه‌ها باید توانی از 2 باشد. چنین محدودیتی در زیرشبکه‌سازی با طول متغیر وجود ندارد.

• زیرشبکه‌سازی مقدار n را افزایش می‌دهد.

• مقام مسئول جهانی برای تخصیص آدرسها، ICANN است. معمولاً ICANN ، بلاکهای بزرگ آدرس را به ISP می‌دهد و آنها به نوبه خود، زیربلاکهای کوچکتر را به متقاضیان می‌دهند.

۵-۷ مجموعه تمرین‌ها

تمرین‌ها

۱- در یک بلاک آدرس، می‌دانیم آدرس IP یک کامپیوتر برابر با 25.34.12.56 / 16 است. اولین آدرس (آدرس شبکه) و آخرین آدرس (آدرس همگانی محدود) در این بلاک، کدامند؟

۲- در یک بلاک آدرس، می‌دانیم آدرس IP یک کامپیوتر برابر با 182.44.82.16 / 26 است. اولین آدرس (آدرس شبکه) و آخرین آدرس (آدرس همگانی محدود) در این بلاک، کدامند؟

۳- اگر تعداد زیرشبکه‌های مطلوب برابر با _____ باشد، تعداد 1 ها در زیرشبکه‌سازی با طول ثابت را بدست آورید.

الف – 2

ب – 62

ج - 122

د - 250

۴- اگر طول پیشوند یک بلاک برابر با هر یک از مقادیر زیر باشد، ماکزیمم تعداد زیر شبکه‌ها در هر حالت، چندتاست؟

الف – 18

ب – 10

ج – 27

www.Abiri.ir

د- 31

۵- بلاک آدرس 16.0.0.0 / 8 به یک سازمان داده شده است. مدیر شبکه می‌خواهد 500 زیرشبکه با طول ثابت ایجاد کند.

الف – مسک زیرشبکه را پیدا کنید.

ب – تعداد آدرسها در هر زیرشبکه را بدست آورید.

ج – در اولین زیر شبکه، اولین و آخرین آدرس را بدست آورید.

د – در آخرین زیرشبکه، اولین و آخرین آدرس را بدست آورید.

۶- بلاک آدرس 130.56.0.0 / 16 به یک سازمان داده شده است. مدیر شبکه می‌خواهد 1024 زیرشبکه ایجاد کند.

الف - مسک زیرشبکه را پیدا کنید.

ب – تعداد آدرسها در هر زیرشبکه را بدست آورید.

ج – در اولین زیر شبکه، اولین و آخرین آدرس را بدست آورید.

د- در آخرین زیرشبکه (زیر شبکه 1024)، اولین و آخرین زیرشبکه را پیدا کنید.

۷- بلاک آدرس 211.17.180.0 / 24 به سازمانی داده شده است. مدیر شبکه می‌خواهد 32 زیرشبکه ایجاد کند.

الف- مسک زیرشبکه را بدست آورید.

ب – تعداد آدرسها در هر زیرشبکه را بدست آورید.

ج – اولین و آخرین آدرس در اولین زیرشبکه را بدست آورید.

د- در آخرین زیرشبکه (زیرشبکه شماره 32) اولین و آخرین آدرس را بدست آورید.

۸- مسکهای زیر را با نماد slash (/n) بنویسید.

الف- 255.255.255.0

ب – 255.0.0.0

ج – 255.255.224.0

د - 255.255.240.0

۹- در بلاکهای زیر، محدوده آدرسها را بدست آورید.

الف- 123.56.77.32 / 29

ب – 200.17.21.128 / 27

ج – 17.34.16.0 / 23

د- 180.34.64.64 / 30

۱۰- یک بلاک آدرس که با آدرس 150.80.0.0 / 16 شروع می‌شود، به یک ISP داده شده است. ISP می‌خواهد این آدرسها را به صورت زیر، بین 1000 نفر از مشترکین خود تقسیم کند:

www.Abiri.ir

الف – اولین گروه، 200 شرکت متوسط تحت پوشش دارد و هر یک به 128 آدرس نیاز دارد.
ب – دومین گروه، 400 شرکت کوچک تحت پوشش دارد و هر یک به 16 آدرس نیاز دارد.
ج – سومین گروه، 2000 منزل مسکونی تحت پوشش دارد و هر یک به 4 آدرس نیاز دارد.
زیربلاکها را طراحی کنید و برای هر زیربلاک از نماد slash استفاده کنید. پیدا کنید که بعد از این تخصیص آدرسها، هنوز چند آدرس باقی می‌ماند .

۱۱- یک بلاک آدرس که با 120.60.4.0 / 20 شروع می‌شود، به یک ISP داده شده است. ISP می‌خواهد به 100 سازمان، بلاکهای آدرس، هر کدام به طول 8 آدرس بدهد. زیربلاکها را طراحی کنید و هر یک را با نماد slash مشخص نمائید. پیدا کنید که بعد از این تخصیص آدرسها، هنوز چند آدرس باقی می‌ماند.

۱۲- یک ISP، یک بلاک آدرس حاوی 1024 آدرس دارد. او می‌خواهد این آدرسها را بین 1024 مشترک خود تقسیم کند. آیا باید زیرشبکه‌سازی کند؟ پاسخ خود را توضیح دهید.

فعالیتهای تحقیقاتی

۱۳- RFC هایی را پیدا کنید که درباره آدرس‌دهی بدون‌کلاس هستند.

۱۴- بلاک آدرسی که به سازمان یا انستیتوی شما داده شده است را پیدا کنید.

۱۵- اگر به کمک یک ISP از منزل به اینترنت متصل می‌شوید، نام ISP و بلاک آدرسهایی که به او داده شده است را پیدا کنید.

۱۶- بعضی‌ها معتقدند که می‌توان کل فضای آدرس را به عنوان یک بلاک واحد در نظر گرفت که در آن هر دنباله‌ای از آدرسها، یک زیربلاک از این بلاک واحد است. این مطلب را بررسی نمائید. اگر این مفهوم را بپذیریم، زیرشبکه‌سازی چطور می‌شود؟

www.Abiri.ir

فصل ششم

مسیردهی بسته‌های IP

در این فصل درباره ***مسیردهی***[1] ، ***هدایت***[2] و ***تحویل***[3] بسته‌های IP به مقصد نهائی آنها صحبت می‌کنیم. مفاهیمی مانند اتصال‌گرا و فاقد اتصال و نیز تحویل مستقیم و غیر مستقیم را مورد بحث قرار خواهیم داد. هدایت بسته که به نحوه حرکت بسته از یک ایستگاه به ایستگاه بعدی اشاره می‌کند را بررسی می‌کنیم و درباره چندین روش هدایت بسته‌ها صحبت خواهیم کرد. مسیردهی که به نحوه تشکیل جداول مسیردهی برای کمک به هدایت بسته‌ها اشاره می‌کند و نیز دو نوع مسیردهی، ***استاتیک***[4] و ***دینامیک***[5] را مورد تجزیه و تحلیل قرار می‌دهیم.

۶-۱ تحویل بسته‌های IP

لایه شبکه بر چگونگی عمل شبکه‌های فیزیکی زیربنائی، روی بسته‌های IP نظارت دارد. عملی که این شبکه‌ها روی بسته انجام می‌دهند تا بسته به مقصد برسد را تحویل بسته‌ها نامیده‌ایم. دو مفهوم مهمی که درباره آنها صحبت خواهیم کرد، نوع سرویس لایه شبکه و تحویل مستقیم و غیرمستقیم بسته‌هاست.

انواع سرویسها در سطح لایه شبکه

تحویل بسته‌ها در سطح لایه شبکه، یا توسط یک سرویس فاقد اتصال و یا توسط یک سرویس اتصال‌گرا صورت می‌گیرد.

در سرویس اتصال‌گرا، تصمیم درباره مسیری که دنباله بسته‌ها از یک مبدأ مشابه به سمت یک مقصد مشابه باید طی کنند، همان ابتدا که اتصال برقرار می‌شود، گرفته می‌شود. روترها برای هر بسته‌ای، مجدداً مسیر را محاسبه نمی‌کنند.

1- Routing
2- Forwarding
3- Delivery
4- Static
5- Dynamic

www.Abiri.ir

* سرویس اتصال‌گرا

پروتوکل لایه شبکه در ***سرویس اتصال‌گرا***[1] ، قبل از آنکه بسته‌ای را ارسال کند، اتصالی را با پروتوکل لایه شبکه سمت گیرنده برقرار می‌کند. وقتی که اتصال برقرار شد، می‌توان دنباله‌ای از بسته‌ها را، یکی پس از دیگری به سمت گیرنده ارسال کرد.

در این حالت، رابطه‌ای بین بسته‌ها وجود دارد. آنها به ترتیب پشت سر هم روی یک مسیر مشابه به طرف مقصد ارسال می‌شوند. از نظر منطقی، یک بسته به بسته‌ای که قبل از او ارسال شده است و نیز به بسته‌ای که بعد از او ارسال می‌شود، متصل است. وقتی که تمام بسته‌های یک پیغام تحویل داده شدند، اتصال خاتمه می‌یابد.

* سرویس فاقد اتصال

در یک ***سرویس فاقد اتصال***[2]، لایه شبکه با هر بسته به صورت مستقل از سایر بسته‌ها رفتار می‌کند به طوری که هر بسته، هیچ رابطه‌ای با سایر بسته‌ها ندارد. بسته‌های یک پیغام ممکن است از طریق مسیرهای متفاوتی خود را به مقصد برسانند. در یک سرویس فاقد اتصال، هر روتر به صورت مستقل از سایر روترها درباره مسیری که بسته باید طی کند، تصمیم می‌گیرد. پروتوکل IP یک پروتوکل فاقد اتصال است. او یک سرویس فاقد اتصال ارائه می‌دهد.

مهم: IP یک پروتوکل فاقد اتصال است.

روشهای مختلف تحویل بسته‌های IP

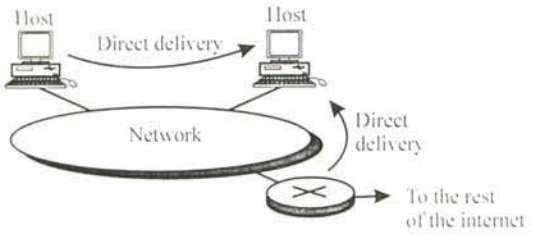
تحویل یک بسته به مقصد نهائی آن توسط یکی از دو متد تحویل انجام می‌شود: مستقیم و غیر مستقیم.

* تحویل مستقیم بسته‌های IP

در ***تحویل مستقیم***[3] ، ***کامپیوتر***[4] مقصد بسته، به همان شبکه‌ای متصل است که کامپیوتر ارسال کننده بسته به آن متصل شده است. تحویل مستقیم زمانی رخ می‌دهد که کامپیوتر مبدأ و کامپیوتر مقصد به شبکه فیزیکی مشابهی متصل شده‌اند و یا اینکه بسته از آخرین روتر به دست کامپیوتر مقصد می‌رسد (شکل ۶-۱ را ببینید).

1- Connection oriented service
2- Connectionless service
3- Direct delivery
4- Host

www.Abiri.ir



شکل ۶-۱: تحویل مستقیم بسته‌های IP

فرستنده به راحتی می‌تواند تعیین کند که آیا تحویل، مستقیم است یا نه. او می‌تواند آدرس شبکه مقصد (با استفاده از مسک) را به دست آورد و این آدرس را با آدرس شبکه‌هایی که به آنها متصل است، مقایسه نماید. اگر تطابق وجود داشت، تحویل مستقیم است.

در تحویل مستقیم، فرستنده از آدرس IP مقصد استفاده می‌کند تا آدرس فیزیکی مقصد را به دست آورد. سپس نرم‌افزار IP، هم آدرس IP و هم آدرس فیزیکی مقصد را به لایه دیتالینک می‌دهد تا بسته را به مقصد تحویل دهد. این فرایند، نگاشت آدرس IP به آدرس فیزیکی نامیده می‌شود. گرچه این عمل نگاشت را می‌توان با جستجو در یک جدول انجام داد، در فصل هفتم خواهیم دید که پروتوکلی به نام *ARP*[1] به طور دینامیک، یک آدرس IP را به آدرس فیزیکی متناظر با آن، می‌نگارد.

❋ تحویل غیرمستقیم بسته‌های IP

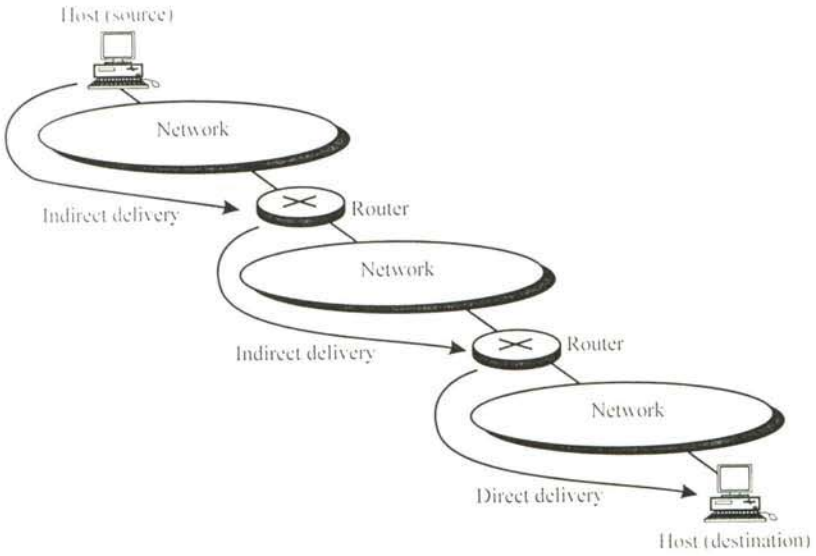
اگر کامپیوتر مقصد به همان شبکه کامپیوتر ارسال‌کننده بسته متصل نباشد، تحویل غیر مستقیم نامیده می‌شود. در تحویل غیر مستقیم، بسته از یک روتر به روتر دیگر می‌رود تا در نهایت به روتری می‌رسد که به شبکه‌ای متصل است که کامپیوتر مقصد نیز در آن شبکه واقع شده است.

دقت نمایید که یک تحویل، همواره شامل یک تحویل مستقیم و همچنین، صفر یا چند تحویل غیر مستقیم است. ضمناً توجه نمائید که آخرین تحویل نیز همواره مستقیم است.

در یک تحویل غیر مستقیم، فرستنده از آدرس IP مقصد و یک جدول مسیردهی برای پیداکردن آدرس IP روتر بعدی که بسته باید به او داده شود، استفاده می‌کند. سپس فرستنده برای پیدا کردن آدرس فیزیکی روتر بعدی، از ARP استفاده می‌کند. دقت نمایید که در تحویل مستقیم، نگاشت آدرس، بین آدرس IP و آدرس فیزیکی مقصد نهائی است. در تحویل غیر مستقیم، نگاشت آدرس بین آدرس IP و آدرس فیزیکی روتر بعدی است.

1- Address Resolution Protocol : ARP

www.Abiri.ir



شکل ۶-۲ : تحویل غیر مستقیم بسته‌های IP

۶-۲ هدایت بسته‌های IP

معنی هدایت بسته‌های IP این است که بسته را در مسیر رسیدن به مقصد خود قرار دهیم. برای انجام این عمل لازم است که کامپیوتر یا روتر، یک جدول مسیردهی داشته باشند. وقتی که کامپیوتر، بسته‌ای برای ارسال دارد و یا وقتی که روتر، بسته‌ای را برای هدایت دریافت می‌کند، به جدول مسیردهی خود نگاه می‌کند تا مسیر رسیدن بسته به مقصد را پیدا کند. با این حال، امروزه این راه حل ساده در شبکه‌ای مانند اینترنت، غیر عملی است زیرا تعداد المانهای لازم در جدول مسیردهی، متناظر با همه مسیرهای ممکن در شبکه، بسیار زیاد می‌شود و جستجو در جدول را نیز غیرکارآمد می‌سازد.

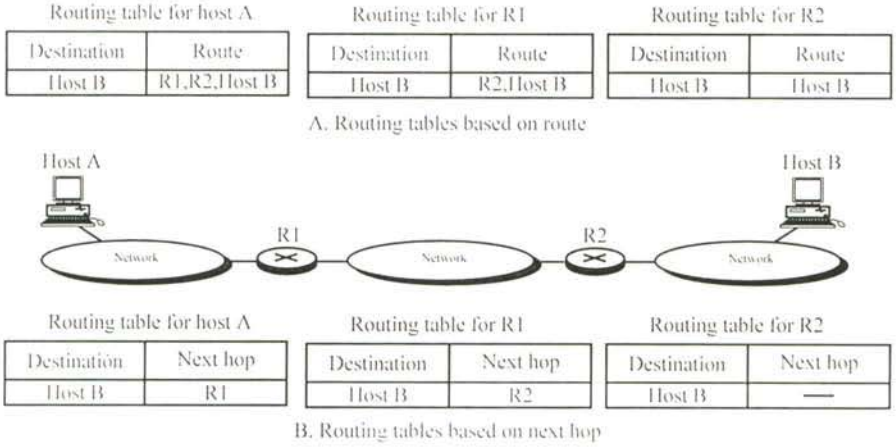
متدهای مختلف هدایت بسته‌های IP

می‌توان با استفاده از چندین روش، مدیریتی بر سایز جدول مسیردهی اعمال کرد و همچنین به مسائلی مانند امنیت نیز رسیدگی کرد. در اینجا به اختصار درباره این روشها صحبت می‌کنیم.

www.Abiri.ir

❊ متد هاپ بعدی

یک روش برای کاهش تعداد المانهای جدول مسیردهی، ***روش هاپ بعدی***[1] می‌باشد. در این روش به جای آنکه جدول مسیردهی، اطلاعات کل مسیر را نگه دارد، فقط آدرس هاپ بعدی را نگه می‌دارد. شکل ۶-۳ نشان می‌دهد که چگونه می‌توان با استفاده از این روش، جدول مسیردهی را کوچک نمود.



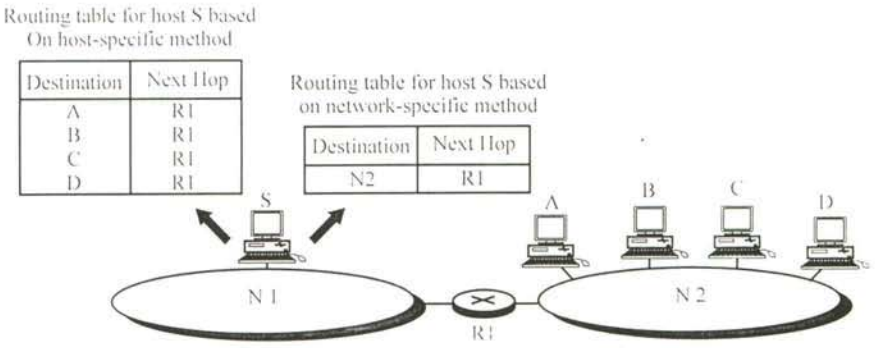
شکل ۶-۳ : متد هاپ بعدی

❊ متد شبکه- مشخص

روش دیگری برای کاهش سایز جدول مسیردهی و ساده کردن فرایند جستجو، **متد شبکه- مشخص**[2] می‌باشد. در این روش به جای آنکه برای هر کامپیوتر متصل به شبکه مقصد، یک المان در جدول داشته باشیم، فقط یک المان برای خود شبکه مقصد خواهیم داشت. به عبارت دیگر، تمام کامپیوترهای متصل به یک شبکه، به عنوان یک المان واحد در نظر گرفته می‌شوند. به عنوان مثال اگر 1000 کامپیوتر به شبکه مشابهی متصل باشند، فقط یک المان در جدول مسیردهی نوشته خواهد شد. شکل ۶-۴ این مفهوم را نشان می‌دهد.

1- Next hop method
2- Network specific method

www.Abiri.ir



شکل ۶-۴ : متد شبکه- مشخص

* متد کامپیوتر- مشخص

در ***متد کامپیوتر- مشخص***[1] ، آدرس کامپیوتر مقصد در جدول مسیردهی نوشته می‌شود. منطقی که پشت این قضیه است، معکوس متد شبکه مشخص است. در اینجا کارآیی را فدای سایر مزایا می‌کنیم.

گرچه گذاشتن آدرس کامپیوتر در جدول مسیردهی چندان کارآیی ندارد، مواردی هست که مدیر شبکه می‌خواهد کنترل بیشتری بر مسیردهی داشته باشد. به عنوان مثال در شکل ۶-۵ ، اگر مدیر شبکه بخواهد تمام بسته‌هایی که برای کامپیوتر B ارسال می‌شوند، به جای آنکه به R1 تحویل داده شوند، به R3 تحویل گردند، یک المان در جدول مسیردهی کامپیوتر A می‌تواند به طور صریح این مسیر را تعریف کند.

مسیردهی کامپیوتر- مشخص برای اهدافی نظیر چک کردن مسیر یا فراهم کردن امنیت لازم، به کار می‌رود.

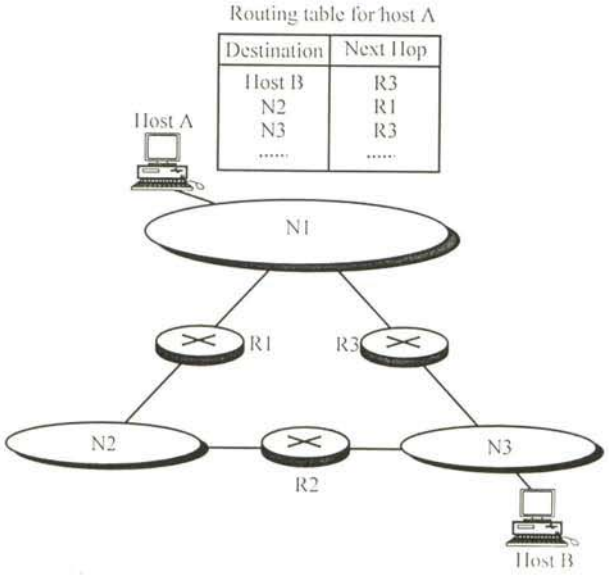
* متد پیش‌فرض

روش دیگری برای ساده‌ترکردن مسیردهی، ***متد پیش‌فرض***[2] نامیده می‌شود. در شکل ۶-۶ ، کامپیوتر A به شبکه‌ای با دو روتر متصل است. کامپیوتر A به کمک روتر R1 بسته‌ها را برای کامپیوترهای متصل به شبکه N2 می‌فرستد و برای بقیه اینترنت، از روتر R2 استفاده می‌کند. بنابراین به جای آنکه کامپیوتر A لیست تمام شبکه‌های متصل به اینترنت را داشته

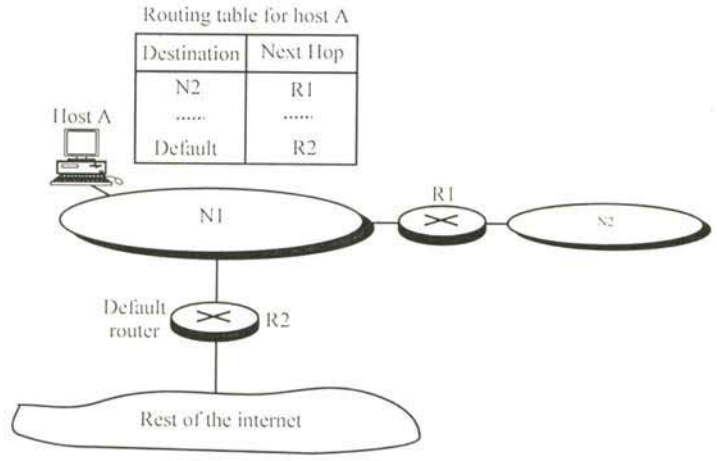
1- Host specific method
2- Default method

www.Abiri.ir

باشد، می‌تواند صرفاً یک المان به نام **پیش‌فرض**[1] داشته باشد (که معمولاً به صورت آدرس شبکه 0.0.0.0 تعریف می‌شود).



شکل ۶-۵ : مسیردهی کامپیوتر- مشخص



شکل ۶-۶ : مسیردهی پیش‌فرض

1- Default

www.Abiri.ir

هدایت بسته‌های IP بر اساس آدرس‌دهی کلاس‌دار

همانطور که در فصل قبل اشاره کردیم، آدرس‌دهی کلاس‌دار، معایب متعددی دارد. با این حال وجود یک مسک پیش‌فرض در یک آدرس کلاس‌دار، فرایند هدایت بسته‌های IP را ساده می‌کند. در این بخش ابتدا محتوای جدول مسیردهی و ماجول هدایت بسته‌های IP را برای وضعیتی که در آن زیرشبکه وجود ندارد، نشان می‌دهیم. سپس نشان می‌دهیم اگر زیرشبکه‌سازی داشته باشیم، چگونه این ماجول تغییر می‌کند.

* هدایت بسته‌های IP بدون زیرشبکه‌سازی

در آدرس‌دهی کلاس‌دار، اکثر روترهای موجود در اینترنت جهانی در زیرشبکه‌سازی وارد نمی‌شوند. زیرشبکه‌سازی داخل سازمان رخ می‌دهد. در این حالت می‌توان با استفاده از سه جدول، یکی برای هر ***آدرس تک‌گیرنده***[1] (کلاس A، B و C)، ماجول هدایت بسته IP را طراحی کرد. اگر روتر از عمل ***چندگیرنده***[2] پشتیبانی کند، می‌توان جدول دیگری را برای کار کردن با آدرسهای چندگیرنده (کلاس D) به کار برد. داشتن سه جدول متفاوت، عمل جستجو را کارآمدتر می‌کند. هر جدول مسیردهی حداقل سه ستون دارد:

۱- آدرس شبکه که معرف شبکه مقصد می‌باشد، نشان می‌دهد که کامپیوتر مقصد کجا واقع شده است. دقت نمائید که از روش شبکه- مشخص برای هدایت بسته استفاده می‌کنیم و متد غیر متداول کامپیوتر- مشخص را به کار نمی‌بریم.

۲- آدرس هاپ بعدی به ما می‌گوید که برای یک تحویل غیرمستقیم، بسته باید به کدام روتر تحویل داده شود. این ستون برای یک تحویل مستقیم، خالی است.

۳- شماره اینترفیس، پورت خروجی که بسته باید از طریق آن به بیرون ارسال شود را مشخص می‌کند. یک روتر معمولاً به چندین شبکه متصل است. هر اتصال، یک اینترفیس متفاوت دارد. ما آنها را به صورت M1, M0 و غیره نشان می‌دهیم. شکل ۶-۷ یک ماجول ساده شده را نشان می‌دهد.

ماجول هدایت بسته‌های IP در ساده‌ترین شکل آن، کارهای زیر را انجام می‌دهد:

۱- آدرس مقصد بسته استخراج می‌شود.

۲- برای پیدا کردن کلاس آدرس، از یک کپی از آدرس مقصد استفاده می‌شود. نحوه کار به این ترتیب است که با شیفت دادن کپی آدرس به اندازه 28=4-32 بیت به سمت راست،

1- Unicast address
2- Multicasting

www.Abiri.ir

کلاس آدرس به دست می‌آید. نتیجه یک عدد 4 بیتی بین 0 و 15 خواهد بود. اگر نتیجه حاصل:

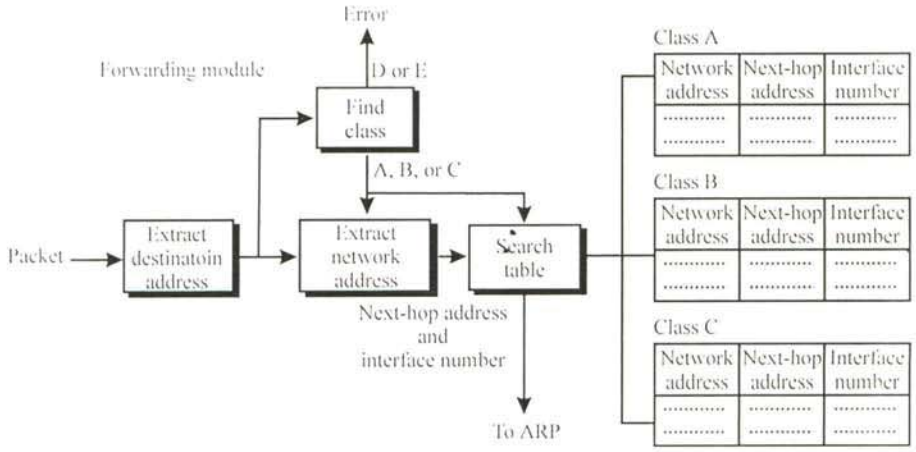
الف- 0 تا 7 باشد، کلاس A است.

ب- 8 تا 11 باشد، کلاس B است.

ج- 12 تا 13 باشد، کلاس C است.

د- 14 باشد، کلاس D است.

ه- 15 باشد، کلاس E است.



شکل ۶-۷: ماجول ساده شده هدایت بسته‌های IP در آدرس‌دهی کلاس‌دار، بدون زیرشبکه‌سازی

۳- از نتیجه مرحله ۲ (برای کلاس A ، B یا C) و نیز آدرس مقصد، برای استخراج آدرس شبکه استفاده می‌شود. برای این کار، 8، 16 یا 24 بیت سمت راست آدرس را ، بسته به کلاس آدرس، صفر می‌کنیم تا آدرس شبکه به دست آید.

۴- برای پیدا کردن اطلاعات مربوطه به هاپ بعدی، کلاس آدرس و آدرس شبکه به کار می‌روند. کلاس، جدولی را تعیین می‌کند که باید عمل جستجو را در آن انجام داد. ماجول در این جدول به دنبال آدرس شبکه می‌گردد. اگر یک ***تطابق***[1] پیدا شود، ***آدرس هاپ بعدی***[2] و شماره اینترفیس پورت خروجی از جدول استخراج می‌شوند. اگر تطابقی پیدا نشود، پیش فرض به کار می‌رود.

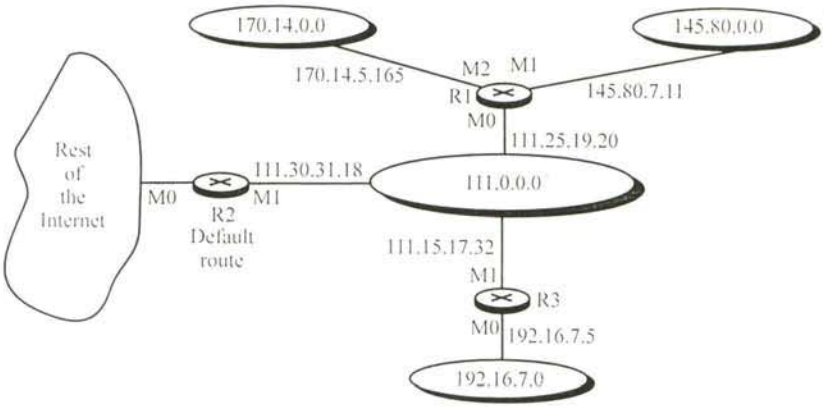
1- Match
2- Next hop address

www.Abiri.ir

۵-ماجول ARP (فصل هفتم) از آدرس IP هاپ بعدی و شماره اینترفیس برای پیدا کردن آدرس فیزیکی روتر بعدی استفاده می‌کند. سپس از لایه دیتالینک می‌خواهد که بسته را به هاپ بعدی تحویل دهد.

**** مثال ۱**

شکل ۶-۸ یک بخش فرضی از اینترنت را نشان می‌دهد. جدول مسیردهی را برای روتر R1 نشان دهید.



شکل ۶-۸ : پیکربندی شبکه برای مثال ۱

راه حل : شکل ۶-۹ سه جدولی که روتر R1 از آنها استفاده می‌کند را نشان می‌دهد. دقت نمایید بعضی از فیلدهای ستون آدرس هاپ بعدی، خالی هستند و دلیل آن این است که در این حالتها، مقصد در همان شبکه‌ای است که روتر به آن متصل است (تحویل مستقیم).

همانطور که در فصل هفتم خواهیم دید، در این حالتها آدرس هاپ بعدی که ARP از آن استفاده می‌کند، همان آدرس مقصد بسته است.

Class A

Network address	Next-hop address	Interface
111.0.0.0	----------	M0

Class B

Network address	Next-hop address	Interface
145.80.0.0	----------	M1
170.14.0.0	----------	M2

Class C

Network address	Next-hop address	Interface
192.16.7.0	111.15.17.32	M0

Default: 111.30.31.18, m0

شکل ۶-۹ : جداول مثال ۱

www.Abiri.ir

**** مثال ۲**

روتر R1 در شکل ۶-۸ بسته‌ای با آدرس مقصد 192.16.7.14 را دریافت می‌کند. نشان دهید که بسته چگونه هدایت می‌شود.

راه حل : آدرس مقصد با نماد باینری به صورت زیر است:

11000000 00010000 00000111 00001110

یک کپی از آدرس را به اندازه 28 بیت به سمت راست شیفت می‌دهیم. نتیجه برابر خواهد بود با :

00000000 00000000 00000000 0000**1100**

که برابر با عدد 12 است و بنابراین شبکه مقصد، کلاس C است. آدرس شبکه با جدا کردن 24 بیت سمت چپ آدرس مقصد به دست می‌آید و نتیجه برابر با 192.16.7.0 می‌باشد. در جدول مربوط به کلاس C ، جستجو به عمل می‌آید. آدرس شبکه در اولین سطر جدول پیدا می‌شود. آدرس IP هاپ بعدی یعنی 111.15.17.32 و شماره اینترفیس یعنی M0 به ARP داده می‌شوند.

**** مثال ۳**

روتر R1 در شکل ۶-۸ بسته‌ای به آدرس مقصد 167.24.160.5 را دریافت می‌کند. نشان دهید که بسته چگونه هدایت می‌شود.

راه حل: آدرس مقصد با نماد باینری به صورت زیر است:

10100111 00011000 10100000 00000101

یک کپی از آدرس را به اندازه 28 بیت به سمت راست شیفت می‌دهیم. نتیجه برابر است با:

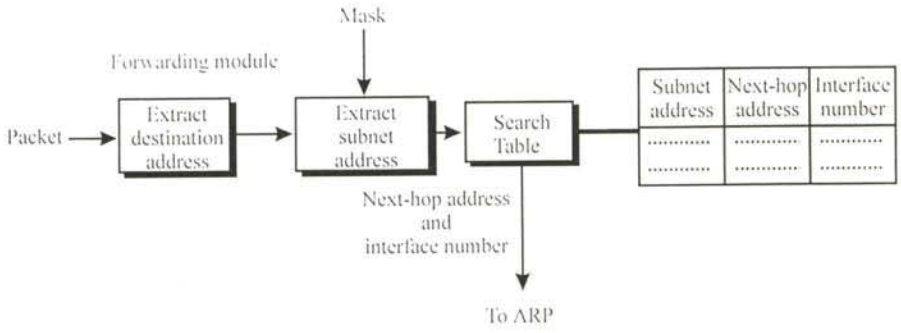
00000000 00000000 00000000 0000**1010**

که معادل 10 اعشاری می‌باشد و بنابراین کلاس آدرس، B است. آدرس شبکه را می‌توان با جدا کردن 16 بیت سمت چپ آدرس مقصد به دست آورد که نتیجه برابر با 167.24.0.0 می‌شود. در جدول کلاس B ، جستجو صورت می‌گیرد. هیچ آدرس شبکه‌ای که تطابق داشته باشد، پیدا نمی‌شود. باید بسته را به روتر پیش‌فرض بدهیم (شبکه مقصد، جای دیگری در اینترنت است). آدرس IP هاپ بعدی یعنی 111.30.31.18 و شماره اینترفیس یعنی M0 به ARP داده می‌شوند.

www.Abiri.ir

* هدایت بسته‌های IP به همراه زیرشبکه‌سازی

در آدرس‌دهی کلاس‌دار، زیرشبکه‌سازی در داخل سازمان انجام می‌شود. روترهایی که عمل زیرشبکه‌سازی را انجام می‌دهند یا در مرز سایت سازمان واقع شده‌اند و یا داخل سازمان هستند. اگر سازمان از زیرشبکه‌سازی با طول متغیر استفاده کند، به چندین جدول نیاز داریم. در غیر این صورت، فقط به یک جدول نیاز داریم. شکل ۶-۱۰ یک ماجول ساده شده‌ای را برای زیرشبکه‌سازی با طول ثابت نشان می‌دهد.



شکل ۶-۱۰: ماجول ساده شده هدایت بسته‌های IP در آدرس‌دهی کلاس‌دار با زیرشبکه‌سازی

۱- ماجول، آدرس مقصد را از شبکه استخراج می‌کند.

۲- آدرس مقصد و مسک برای استخراج آدرس زیرشبکه به کار می‌روند.

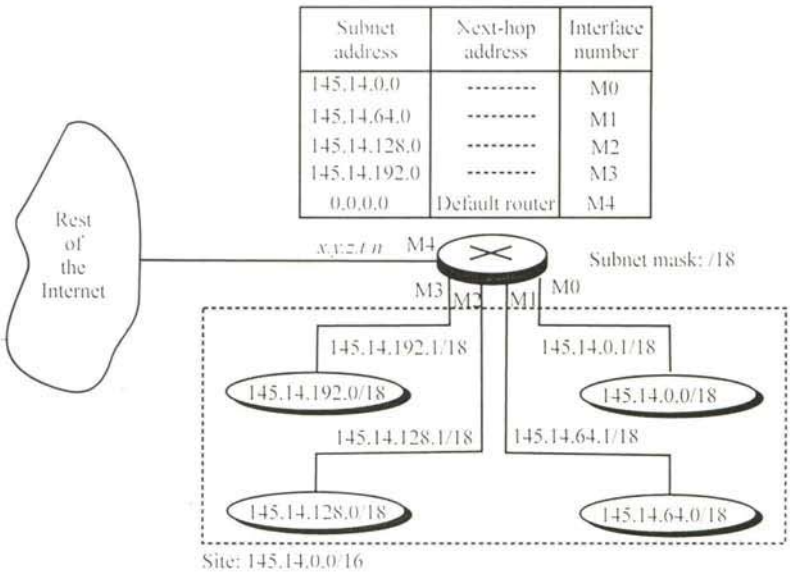
۳- با استفاده از آدرس زیرشبکه، جستجو در داخل جدول صورت می‌گیرد تا آدرس IP هاپ بعدی و شماره اینترفیس پیدا شود. اگر تطابقی پیدا نشود، از پیش‌فرض استفاده می‌شود.

۴- آدرس IP هاپ بعدی و شماره اینترفیس به ARP داده می‌شوند.

** مثال ۴

شکل ۶-۱۱ روتری را نشان می‌دهد که به چهار زیرشبکه متصل شده است. به چندین نکته توجه نمائید. اول آنکه آدرس سایت برابر با 145.14.0.0/16 (یک آدرس کلاس B) می‌باشد. هر بسته‌ای که آدرس مقصد آن در محدوده از 145.14.0.0 تا 145.14.255.255 باشد، به اینترفیس M4 داده می‌شود و توسط روتر، روانه زیرشبکه مقصد می‌شود. دوم آنکه ما از آدرس x.y.z.t/n برای اینترفیس M4 استفاده کرده‌ایم زیرا نمی‌دانیم این روتر به چه شبکه‌ای متصل شده است. سوم آنکه جدول، یک المان پیش‌فرض برای بسته‌هایی که باید به خارج از این سایت ارسال شوند، دارد. روتر چنان پیکربندی شده است که مسک زیر شبکه 18/ را به هر آدرس مقصدی اعمال کند.

www.Abiri.ir



شکل ۶-۱۱: پیکربندی شبکه برای مثال ۴

❊❊ مثال ۵

روتر شکل ۶-۱۱ بسته‌ای به آدرس مقصد 145.14.32.78 را دریافت می‌کند. نشان دهید که بسته چگونه هدایت می‌شود.

راه حل : مسک برابر با /18 است. بعد از اِعمال مسک، آدرس زیرشبکه برابر با 145.14.0.0 بدست می‌آید. بسته به همراه آدرس IP هاپ بعدی یعنی 145.14.32.78 و اینترفیس خروجی M0 به ARP داده می‌شوند.

❊❊ مثال ۶

کامپیوتری در شبکه 145.14.0.0 در شکل ۶-۱۱ بسته‌ای برای ارسال به کامپیوتری با آدرس 7.22.67.91 دارد. نشان دهید که بسته چگونه مسیردهی می‌شود.

راه حل : روتر بسته را دریافت می‌کند و مسک /18 را به آن اِعمال می‌کند. آدرس شبکه برابر با 7.22.64.0 می‌باشد. جدول جستجو می‌شود و آدرس پیدا نمی‌شود. روتر از آدرس روتر پیش‌فرض (که در شکل نشان داده نشده است) استفاده می‌کند و بسته را برای آن روتر پیش‌فرض می‌فرستد.

www.Abiri.ir

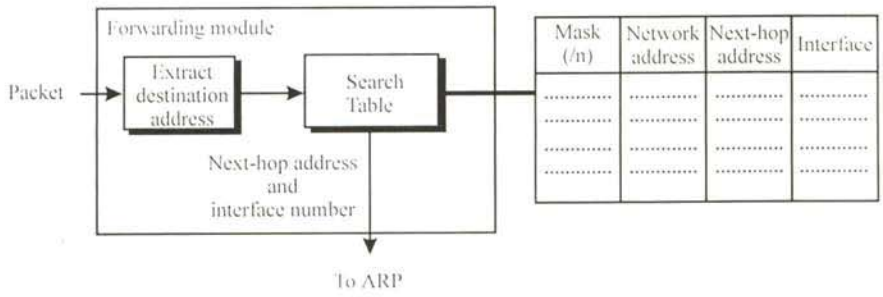
هدایت بسته‌های IP بر اساس آدرس‌دهی بدون‌کلاس

کل فضای آدرس در آدرس‌دهی بدون‌کلاس، یک واحد یکپارچه است و لذا هیچ کلاسی وجود ندارد. معنی آن این است که هدایت بسته IP ، به یک سطر اطلاعات برای هر بلاک آدرس نیاز دارد. باید بر اساس آدرس شبکه (اولین آدرس در بلاک آدرس) جستجو را انجام داد. متأسفانه آدرس مقصد در بسته، هیچ سرنخی درباره آدرس شبکه نمی‌دهد (در صورتی که در آدرس‌دهی کلاس‌دار اینطور نیست).

برای حل مسئله، باید مسک (/n) را در جدول وارد کنیم. باید یک ستون اضافه داشته باشیم که حاوی مسک برای بلاک آدرس متناظر باشد. به عبارت دیگر، گرچه می‌توان یک جدول مسیردهی کلاس‌دار را با سه ستون طراحی کرد، جدول مسیردهی بدون‌کلاس به حداقل چهار ستون نیاز دارد.

مهم: در آدرس‌دهی کلاس‌دار می‌توانیم یک جدول مسیردهی با سه ستون داشته باشیم. در آدرس‌دهی بدون‌کلاس، به حداقل چهار ستون در جدول مسیردهی نیاز داریم.

شکل ۶-۱۲ یک ماجول ساده هدایت بسته را برای آدرس‌دهی بدون کلاس نشان می‌دهد. دقت نمایید که استخراج آدرس شبکه، همزمان با جستجوی جدول صورت می‌گیرد زیرا برای استخراج آدرس شبکه هیچ اطلاعاتی در آدرس مقصد وجود ندارد که بتوان در این رابطه از آن استفاده کرد.



شکل ۶-۱۲: ماجول ساده شده هدایت بسته‌های IP در آدرس‌دهی بدون‌کلاس

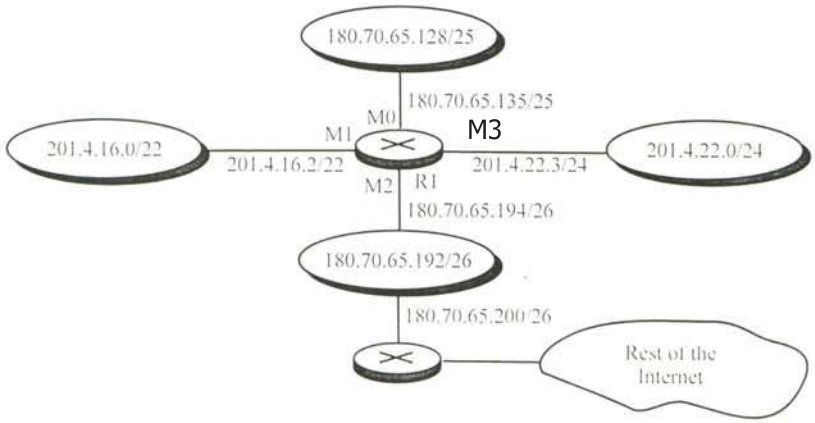
* چند مثال

سعی می‌کنیم به کمک مثالهای زیر، نحوه هدایت بسته‌های IP در سیستم آدرس‌دهی بدون‌کلاس را توضیح دهیم.

www.Abiri.ir

**** مثال ۷**

با استفاده از پیکربندی شکل ۶-۱۳ ، یک جدول مسیردهی برای روتر R1 بسازید.



شکل ۶-۱۳: پیکربندی شبکه برای مثال ۷

راه حل: جدول ۶-۱ ، جدول مطلوب را نشان می‌دهد.

Mask	Network Address	Next-Hop	Interface
/26	180.70.65.192	-	m2
/25	180.70.65.128	-	m0
/24	201.4.22.0	-	m3
/22	201.4.16.0	------	m1
Default	Default	180.70.65.200	m2

جدول ۶-۱: جدول مسیردهی برای روتر R1 در شکل ۶-۱۳

**** مثال ۸**

اگر در شکل ۶-۱۳ بسته‌ای به آدرس مقصد 180.70.65.140 به روتر R1 برسد، فرایند هدایت بسته‌های IP را نشان دهید.

راه حل : روتر کارهای زیر را انجام می‌دهد:

۱- اولین مسک (/26) به آدرس مقصد اعمال می‌شود. نتیجه برابر با 180.70.65.128 خواهد بود که با آدرس شبکه متناظر با این مسک در جدول ۶-۱ مطابقت ندارد.

۲- دومین مسک (/25) به آدرس مقصد اعمال می‌شود. نتیجه برابر با 180.70.65.128 خواهد بود که با آدرس شبکه متناظر با این مسک در جدول ۶-۱ مطابقت دارد. آدرس IP هاپ

www.Abiri.ir

بعدی (در این حالت، آدرس مقصد بسته) و شماره اینترفیس M0 برای پردازش بیشتر به ARP داده می‌شوند.

**** مثال ۹**

اگر در شکل ۶-۱۳ بسته‌ای به آدرس مقصد 201.4.22.35 به R1 برسد، فرایند هدایت بسته را نشان دهید.

راه حل : روتر عملیات زیر را انجام می‌دهد:

۱- اولین مسک (26/) به آدرس مقصد اعمال می‌شود. نتیجه برابر با 201.4.22.0 است که با آدرس شبکه سطر اول مطابقت ندارد.

۲-دومین مسک (25/) به آدرس مقصد اعمال می‌شود. نتیجه برابر با 201.4.22.0 است که با آدرس شبکه سطر دوم مطابقت ندارد.

۳-سومین مسک (24/) به آدرس مقصد اعمال می‌شود. نتیجه برابر با 201.4.22.0 است که با آدرس شبکه سطر سوم مطابقت دارد. آدرس مقصد بسته و شماره اینترفیس M3 به ARP داده می‌شود.

**** مثال ۱۰**

اگر بسته‌ای در شکل ۶-۱۳ با آدرس مقصد 18.24.32.78 به روتر R1 برسد، فرایند هدایت بسته را نشان دهید.

راه حل: سطر به سطر، کلیه مسکها را به آدرس مقصد اعمال می‌کنیم اما هیچ آدرس شبکه‌ای پیدا نمی‌شود که مطابقت داشته باشد. وقتی که ماجول به انتهای جدول می‌رسد، آدرس IP هاپ بعدی یعنی 180.70.65.200 و شماره اینترفیس یعنی M2 را به ARP می‌دهد. این بسته باید به کمک روتر پیش‌فرض به جای دیگری در اینترنت ارسال شود.

**** مثال ۱۱**

حال اجازه دهید نوع مثال را تغییر دهیم. اگر فقط جدول مسیردهی یک روتر را بدانیم، آیا می‌توانیم توپولوژی شبکه متصل به آن را پیدا کنیم؟ مثلاً جدول مسیردهی روتر R1 در جدول ۶-۲ داده شده است. آیا می‌توانیم توپولوژی شبکه متصل به آن را ترسیم نماییم؟

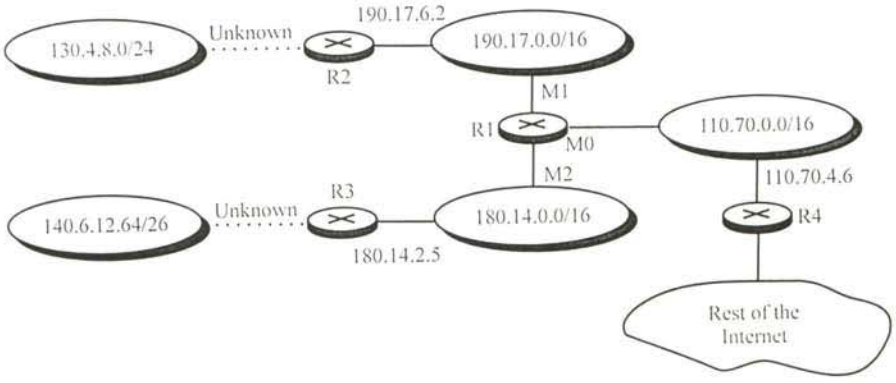
راه حل : بعضی از حقایق را می‌دانیم اما برای ارائه یک توپولوژی مشخص، کافی نیست. می‌دانیم روتر R1 سه اینترفیس دارد. M0 ، M1 و M2 . می‌دانیم سه شبکه وجود دارد که مستقیماً به روتر R1 متصل هستند. می‌دانیم دو شبکه وجود دارد که به طور غیرمستقیم به R1 متصل شده‌اند. باید حداقل سه روتر دیگر نیز وجود داشته باشد (ستون هاپ بعدی را ببینید).

www.Abiri.ir

Mask	Network Address	Next-Hop Address	Interface Number
/26	140.6.12.64	180.14.2.5	m2
/24	130.4.8.0	190.17.6.2	m1
/16	110.70.0.0	----------	m0
/16	180.14.0.0	----------	m2
/16	190.17.0.0	----------	m1
Default	Default	110.70.4.6	m0

جدول ۶-۲: جدول مسیردهی برای مثال ۱۱

با نگاه کردن به آدرس IP آنها می‌فهمیم که این روترها به چه شبکه‌هایی متصل هستند. بنابراین می‌توانیم آنها را در جای مناسب قرار دهیم. می‌دانیم یک روتر، روتر پیش‌فرض، به بقیه اینترنت متصل است. اما یک سری از اطلاعات وجود ندارند . نمی‌دانیم که آیا شبکه 130.4.8.0 مستقیماً به روتر R2 متصل است و یا اینکه از طریق یک روتر دیگر و یک اتصال نقطه به نقطه، به روتر R2 متصل شده است. نمی‌دانیم که آیا شبکه 140.6.12.64 مستقیماً به روتر R3 متصل شده است و یا از طریق روتر دیگری و یک لینک نقطه به نقطه به آن متصل است. معمولاً شبکه‌های نقطه به نقطه، المانی در جدول مسیردهی ندارند زیرا هیچ کامپیوتری به آنها متصل نیست. شکل ۶-۱۴ توپولوژی حدسی ما را نشان می‌دهد.



شکل ۶-۱۴: حدس ما از توپولوژی مثال ۶

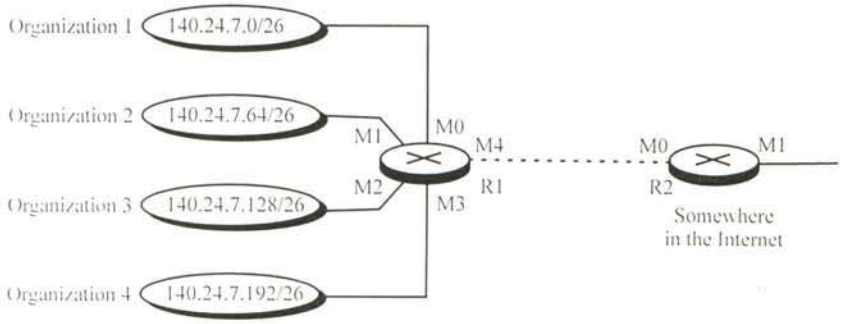
www.Abiri.ir

* تجمع آدرس

وقتی که از آدرس‌دهی کلاس‌دار استفاده می‌کنیم، فقط یک المان در جدول مسیردهی برای هر سایتی خارج از سازمان وجود دارد. آن المان، حتی اگر آن سایت، زیرشبکه‌سازی شده باشد، سایت را تعریف می‌کند. وقتی که بسته‌ای به روتر می‌رسد، روتر المان متناظر را چک می‌کند و بر اساس آن، بسته‌ها را هدایت می‌کند.

وقتی که از آدرس‌دهی بدون‌کلاس استفاده می‌کنیم، احتمال آن وجود دارد که تعداد المانهای جدول مسیردهی افزایش یابد. دلیل آن این است که هدف از به کار بردن آدرس‌دهی بدون‌کلاس، تقسیم کردن کل فضای آدرس به بلاکهای آدرس قابل مدیریت است. بزرگ شدن اندازه جدول مسیردهی، باعث زیاد شدن وقت لازم جهت جستجو در جدول می‌شود.

برای حل این مسئله، ایده ***تجمع آدرس***[1] مطرح شد. در شکل ۶-۱۵ دو روتر داریم.



Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	---------	M0
/26	140.24.7.64	---------	M1
/26	140.24.7.128	---------	M2
/26	140.24.7.192	---------	M3
0	0.0.0.0	Default router	M4

Routing table for R1

Mask	Network address	Next-hop address	Interface
/24	140.24.7.0	---------	M0
/0	0.0.0.0	Default router	M1

Routing table for R2

شکل ۶-۱۵: تجمع آدرس

R1 به شبکه‌های چهار سازمان متصل شده است و هر کدام 64 آدرس دارند. R2 جائی دورتر از R1 است. جدول مسیردهی R1 ، بزرگتر است زیرا که هر بسته باید به طور صحیح به سمت سازمان مناسب، مسیردهی شود. از طرف دیگر، R2 می‌تواند یک جدول مسیردهی خیلی کوچک داشته باشد. برای R2 ، هر بسته‌ای با آدرس مقصد از 140.24.7.0 تا 140.24.7.255 ، بدون توجه به شماره سازمان، از اینترفیس M0 به سمت روتر R1 فرستاده می‌شود. این عمل

1- Address aggregation

www.Abiri.ir

به نام تجمع آدرس موسوم است زیرا که می‌توان بلاکهای آدرس چهار سازمان را، در داخل یک بلاک آدرس بزرگتر در نظر گرفت (140.24.7.0/24). اگر هر سازمانی آدرسهایی داشت که نمی‌شد آنها را داخل یک بلاک قرار داد، جدول مسیردهی R2 بزرگتر می‌شد.

توجه نمائید گرچه ایده تجمع آدرس، مشابه با ایده زیرشبکه‌سازی است اما در اینجا یک سایت مشترک نداریم. شبکه هر سازمان، مستقل است. به علاوه، چندین سطح از تجمع آدرس می‌توانیم داشته باشیم.

* مطابقت با بلندترین مسک

اگر یکی از سازمانهای شکل قبل از نظر جغرافیایی نزدیک به سه سازمان دیگر نباشد، چه اتفاقی می‌افتد؟ به عنوان مثال اگر بنا به دلایلی نتوان سازمان 4 را به روتر R1 متصل کرد، آیا هنوز می‌توانیم از ایده تجمع آدرس استفاده کنیم و بلاک 140.24.7.192/26 را به سازمان 4 اختصاص دهیم؟ پاسخ، مثبت است زیرا که مسیردهی در آدرس‌دهی بدون‌کلاس، اصل دیگری را به کار می‌برد که ***مطابقت با بلندترین مسک***[1] نامیده می‌شود. این قاعده چنین بیان می‌کند که مرتب سازی جدول مسیردهی، از بلندترین مسک آغاز می‌شود و به کوتاهترین مسک ختم می‌شود. به عبارت دیگر اگر سه مسک 27/ ، 26/ و 24/ را داشته باشیم، مسک 27/ باید اولین المان جدول مسیردهی و مسک 24/ آخرین المان جدول مسیردهی باشند. اجازه دهید ببینیم آیا این قاعده، وضعیتی را که در آن سازمان 4 از سه سازمان دیگر جدا باشد را حل می‌کند یا نه. شکل ۶-۱۶ وضعیت را نشان می‌دهد.

فرض کنید بسته‌ای با آدرس مقصد 140.24.7.200 برای سازمان 4 از راه می‌رسد. در روتر R2 اولین مسک اعمال می‌شود که آدرس شبکه 140.24.7.192 را می‌دهد. بسته به طور صحیح از طریق اینترفیس M1 به سمت سازمان 4 مسیردهی می‌شود اما اگر ترتیب جدول مسیردهی بر اساس قرار دادن بلندترین مسک در ابتدای جدول مسیردهی نبود، اعمال مسک 24/ سبب می‌شد که بسته به غلط به سمت روتر R1 مسیردهی شود.

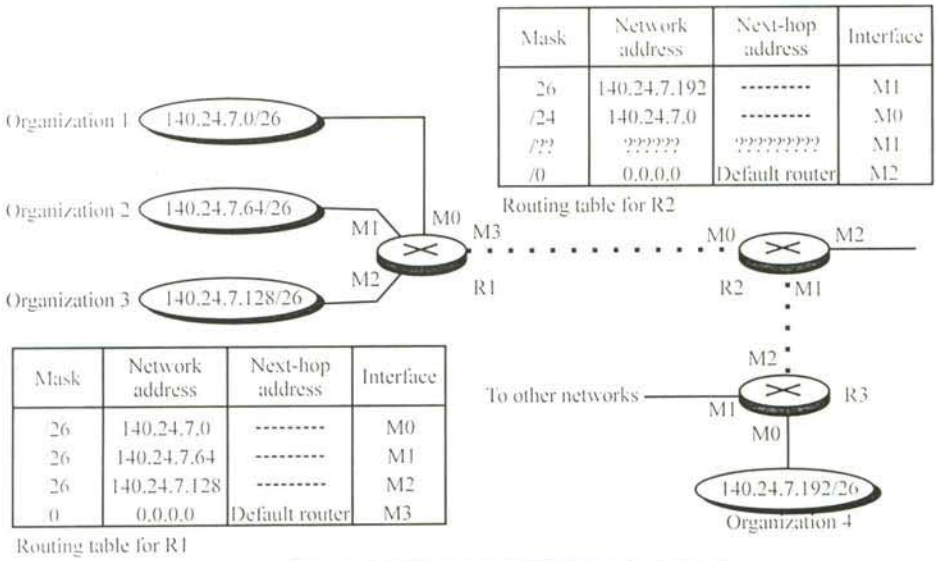
* مسیردهی طبقاتی

برای حل مشکل جداول مسیردهی بسیار بزرگ، می‌توانیم سلسله مراتبی را در جدول مسیردهی ایجاد کنیم. در فصل اول اشاره کردیم که امروزه اینترنت یک حالت طبقاتی دارد. گفتیم که اینترنت به ISPهای ملی و بین‌المللی تقسیم می‌شود. ISP های ملی به ISP های ناحیه‌ای تقسیم می‌شوند و ISP های ناحیه‌ای به ISP های محلی تقسیم می‌گردند. اگر جدول

1- Longest mask matching

www.Abiri.ir

مسیردهی یک حالت طبقاتی مانند معماری طبقاتی اینترنت داشته باشد، می‌توان سایز جدول مسیردهی را کاهش داد.



Mask	Network address	Next-hop address	Interface
26	140.24.7.192	---------	M1
/24	140.24.7.0	---------	M0
/??	??????	?????????	M1
/0	0.0.0.0	Default router	M2

Routing table for R2

Mask	Network address	Next-hop address	Interface
26	140.24.7.0	---------	M0
26	140.24.7.64	---------	M1
26	140.24.7.128	---------	M2
0	0.0.0.0	Default router	M3

Routing table for R1

Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	---------	M0
/??	????????	?????????	M1
/0	0.0.0.0	Default router	M2

Routing table for R3

شکل ۶-۱۶: مطابقت با بلندترین مسک

اجازه دهید یک ISP محلی را در نظر بگیریم. می‌توان آن را یک بیشوند به طول مشخص، یک بلاک آدرس بزرگ را به یک ISP محلی اختصاص داد. ISP محلی می‌تواند این بلاک را به بلاکهای کوچکتری با سایز متفاوت تقسیم کند و این بلاکهای کوچکتر را به کاربران و سازمانهای منفرد بدهد. اگر بلاکی که به ISP محلی داده می‌شود، با آدرس a.b.c.d/n شروع شود، ISP می‌تواند بلاکهایی ایجاد نماید که با e.f.g.h/m شروع می‌شوند و در آنها m برای هر متقاضی مقدار متفاوتی می‌تواند داشته باشد و این مقدار از n بزرگتر است.

چگونه این عمل، سایز جدول مسیردهی را کوچک می‌کند؟ لازم نیست که بقیه اینترنت از این تقسیم‌بندی با خبر باشند. همه مشترکین ISP محلی از دید بقیه اینترنت به صورت a.d.c.d/n تعریف شده‌اند. هر بسته‌ای که برای این بلاک بزرگ آدرس فرستاده شده باشد، به سمت این ISP محلی مسیردهی می‌شود. در جدول مسیردهی هر روتری در دنیا، فقط یک المان برای کلیه مشترکین این ISP محلی وجود دارد. همه مشترکین این ISP محلی در واقع یک گروه واحد را تشکیل می‌دهند. البته داخل ISP محلی، روتر باید زیربلاکها را تشخیص دهد و بسته را به سمت مشترک مورد نظر هدایت نماید. اگر یکی از مشترکین، خودش یک

www.Abiri.ir

سازمان بزرگ باشد، او هم می‌تواند یک سطح دیگر از طبقه‌بندی را به کمک زیرشبکه‌سازی ایجاد نماید و زیربلاک اختصاص یافته به خود را به بلاکهای کوچکتری تقسیم نماید. در مسیردهی بدون‌کلاس، تا زمانی که از قوانین آدرس‌دهی بدون‌کلاس تبعیت نمائیم، به تعداد زیادی می‌توان سلسله مراتب طبقه‌بندی ایجاد کرد.

**** مثال ۱۲**

به عنوان مثالی از ***مسیردهی طبقاتی***[1]، اجازه دهید شکل ۶-۱۷ را در نظر بگیریم. 16384 آدرس که از آدرس 120.14.64.0 شروع می‌شوند، به یک ISP ناحیه‌ای داده شده‌اند. ISP ناحیه‌ای تصمیم گرفته است که این بلاک آدرس را به چهار زیربلاک، هر یک حاوی 4096 آدرس، تقسیم نماید. سه تا از این زیربلاکها به سه ISP محلی اختصاص داده شده‌اند و زیربلاک دوم برای استفاده آتی رزرو شده است. دقت نمایید که مسک هر زیربلاک برابر با /20 (32-12=20 , $4096=2^{12}$) می‌باشد زیرا که بلاک اصلی با مسک /18 (32-14=18 , $16384=2^{14}$) به چهار زیربلاک تقسیم شده است.

اولین ISP محلی، زیربلاک اختصاصی خود را به 8 بلاک کوچکتر تقسیم کرده است و هر یک از آنها را به یک ISP کوچک داده است. هر ISP کوچک سرویسهایی را به 128 منزل (H001 تا H128) می‌دهد و هر یک از این منازل، از چهار آدرس استفاده می‌کنند. دقت نمایید که اینک، مسک برای هر ISP کوچک برابر با /23 (23=20+3 , $8=2^3$) است زیرا که بلاک به 8 زیربلاک کوچکتر تقسیم شده است. هر منزل یک مسک /30 دارد زیرا که هر منزل، فقط $2^{32-30}=4$ آدرس دارد.

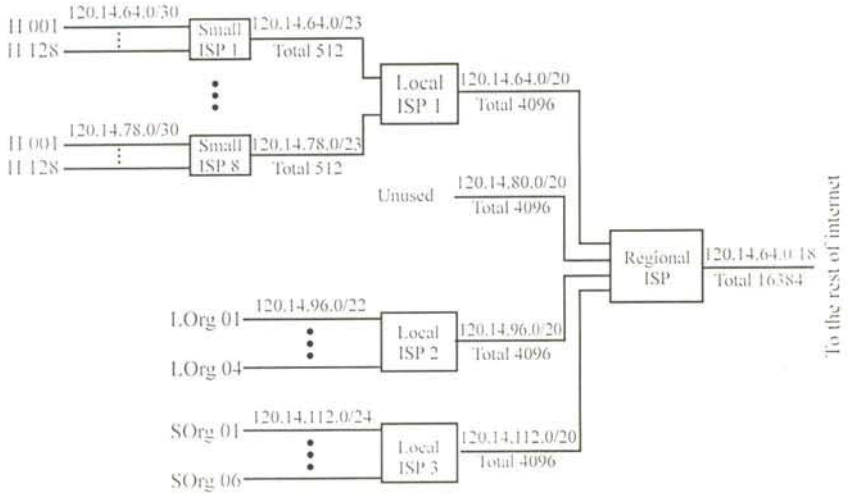
ISP محلی دوم، بلاک آدرس خود را به 4 زیربلاک تقسیم کرده است و آدرسها را به 4 سازمان بزرگ (LOrg01 تا LOrg04) داده است. دقت نمایید که هر سازمان بزرگ حاوی 1024 آدرس است و دارای مسک /22 (22=20+2 , $4=2^2$) می‌باشد.

ISP محلی سوم، بلاک آدرس خود را به 16 زیربلاک تقسیم کرده است و هر زیربلاک را به یک سازمان کوچک (SOrg01 تا SOrg15) داده است. هر سازمان کوچکی دارای 256 آدرس است و مسک آن برابر با /24 (24=20+4 , $16=2^4$) می‌باشد.

این نوع پیکربندی یک حالت طبقاتی دارد. همه روترها در اینترنت، بسته‌ای با آدرس مقصد از 120.14.64.0 تا 120.14.127.255 را برای ISP ناحیه‌ای می‌فرستند.ISP ناحیه‌ای ، هر بسته‌ای با آدرس مقصد 120.14.64.0 تا 120.14.79.255 را برای اولین ISP محلی می‌فرستد. اولین ISP محلی، هر بسته‌ای با آدرس مقصد از 120.14.64.0 تا 120.14.64.3 را برای H001 می‌فرستد.

1- Hierarchical routing

www.Abiri.ir



شکل ۶-۱۷ : مسیردهی طبقاتی با ISP ها

* مسیردهی جغرافیایی

برای آنکه باز هم تا حد امکان سایز جدول مسیردهی را کوچکتر کنیم، لازم است که مسیردهی طبقاتی را توسعه دهیم و مسیردهی جغرافیایی را نیز در آن وارد کنیم. باید کل فضای آدرس را به چند بلاک بزرگ تقسیم نمائیم و یک بلاک به آمریکای شمالی، یک بلاک به اروپا، یک بلاک به آسیا و یک بلاک به آفریقا و غیره اختصاص دهیم. روترهای ISP هایی که خارج از اروپا هستند، فقط یک المان در جدول مسیردهی خود برای بسته‌هایی که به سمت اروپا می‌روند، دارند. روترهای ISPهایی که خارج از آمریکای شمالی هستند، در جدول مسیردهی خود فقط یک المان برای مسیردهی بسته‌هایی دارند که به سمت آمریکای شمالی می‌روند. همین‌طور الی آخر.

الگوریتمهای جستجو در جدول مسیردهی

برای آنکه مسیردهی بدون‌کلاس کارآمدتر شود باید الگوریتمهای آدرس‌دهی کلاس‌دار که به جستجو در جداول مسیردهی می‌پردازند را تغییر داد. این امر شامل الگوریتمهایی که جداول مسیردهی را به روزرسانی می‌کنند نیز می‌شود. در فصل چهاردهم درباره این تکنیکهای به روزرسانی جداول مسیردهی صحبت خواهیم کرد.

www.Abiri.ir

* جستجو در آدرس‌دهی کلاس‌دار

در آدرس‌دهی کلاس‌دار، جدول مسیردهی به صورت یک لیست، سازماندهی می‌شود. با این وجود برای آنکه جستجو کارآمدتر شود، می‌توان جدول مسیردهی را به سه جدول تقسیم کرد (که گاهی اوقات، ***ظرف***[1] نیز نامیده می‌شود) و هر جدول را برای یک کلاس آدرس بکار برد. روتر وقتی که بسته‌ای از راه می‌رسد، مسک پیش‌فرض (که از خود آدرس به راحتی قابل استخراج است) را به آدرس مقصد بسته اعمال می‌کند تا جدول متناظر با آن (A ، B یا C) را بدست آورد. سپس روتر عمل جستجو را صرفاً در جدول مربوطه و نه در کل جدول مسیردهی انجام می‌دهد. حتی بعضی از روترها، 8 جدول را برای کلاس A ، 4 جدول را برای کلاس B و 2 جدول را برای کلاس C اختصاص می‌دهند و این عمل را بر اساس نتیجه حاصل از فرایند پیدا کردن کلاس، انجام می‌دهند.

* جستجو در آدرس‌دهی بدون‌کلاس

در آدرس‌دهی بدون‌کلاس هیچ گونه اطلاعاتی درباره شبکه در آدرس مقصد وجود ندارد. ساده‌ترین و البته نه کارآمدترین متد جستجو، ***مطابقت با بلندترین مسک***[2] نامیده می‌شود و قبلاً درباره آن صحبت کردیم. می‌توان جدول مسیردهی را به جداول کوچکتری تقسیم کنیم به طوری که برای هر پیشوند، یک جدول داشته باشیم. روتر ابتدا بلندترین پیشوند را امتحان می‌کند. اگر آدرس مقصد در این جدول پیدا شود، عمل جستجو کامل شده است. اگر آدرس پیدا نشد، جدول مربوط به پیشوند بعدی جستجو می‌شود و همین‌طور الی آخر. واضح است که این نوع جستجو، مدت زمان زیادی به طول می‌انجامد.

یک راه حل آن است که ساختار دیتای بکار رفته برای جستجو را تغییر دهیم. به جای یک لیست می‌توان از سایر ساختارهای دیتا (مانند یک درخت یا یک درخت باینری) استفاده کنیم. یک روش، استفاده از نوعی درخت جستجو به نام tric است اما صحبت درباره این مطلب خارج از موضوع این کتاب است.

* ترکیب

باید اشاره کنیم که روترهای مدرن همگی بر اساس آدرس‌دهی بدون‌کلاس کار می‌کنند. همه آنها در جدول مسیردهی خود، مسک را نیز می‌گذراند. تا زمانی که مسیردهی

1- Bucket
2- Longest mask matching

www.Abiri.ir

کلاس‌دار به طور کامل از اینترنت حذف شود، با آن به عنوان حالت خاصی از مسیردهی بدون‌کلاس برخورد می‌شود و لذا از مسکهای 24/ ، 16/ و 8/ نیز استفاده می‌شود.

۳-۶ مسیردهی بسته‌های IP

تولید و نگهداری جداول مسیردهی، موضوع مسیردهی بسته‌های IP می‌باشد.

جداول مسیردهی استاتیک و دینامیک

یک کامپیوتر یا روتر برای مسیردهی بسته‌های IP حاوی یک جدول مسیردهی است که در آن یک المان برای هر مقصد یا ترکیبی از مقصدها وجود دارد. جدول مسیردهی می‌تواند استاتیک یا دینامیک باشد.

❊ جدول مسیردهی استاتیک

یک ***جدول مسیردهی استاتیک***[1] حاوی اطلاعاتی است که به صورت دستی نوشته می‌شوند. مدیر شبکه، مسیر رسیدن به هر مقصدی را داخل جدول می‌نویسد. وقتی که جدول نوشته می‌شود، اگر تغییری در اینترنت رخ دهد، جدول نمی‌تواند به طور اتوماتیک محتوای خود را به روزرسانی کند. مدیرشبکه باید به صورت دستی، جدول را تغییر دهد.

می‌توان در یک شبکه‌بندی کوچک که اغلب ثابت است و تغییر نمی‌کند از جدول مسیردهی استاتیک استفاده کرد. این مطلب درباره شبکه‌بندی‌هایی که برای کارهای آزمایشی نیز به کار می‌روند، صادق است. اما استفاده از جدول مسیردهی استاتیک در یک شبکه‌بندی بزرگ مانند اینترنت، استراتژی خوبی نیست.

❊ جدول مسیردهی دینامیک

یک ***جدول مسیردهی دینامیک***[2] با استفاده از یکی از پروتوکلهای مسیردهی دینامیک مانند RIP ، OSPF یا BGP (فصل چهاردهم را ببینید) به طور پریودیک، به روزرسانی می‌شود. هر وقت که تغییری در اینترنت رخ دهد، مثلاً یک روتر از کار بیفتد یا لینکی قطع شود، پروتوکلهای مسیردهی دینامیک، تمام جداول مسیردهی روترها را به طور اتوماتیک به روزرسانی می‌کنند.

1- Static routing table
2- Dynamic routing table

www.Abiri.ir

به منظور تحویل کارآمد بسته‌های IP ، باید روترهای یک شبکه‌بندی بزرگ مانند اینترنت را به صورت دینامیک، به روزرسانی کنیم. در فصل چهاردهم به طور مفصل درباره سه پروتوکل مسیردهی دینامیک صحبت خواهیم کرد.

ساختار کلی جدول مسیردهی

همانطور که قبلاً اشاره کردیم، یک جدول مسیردهی برای آدرس‌دهی بدون‌کلاس، حداقل دارای چهار ستون است. با این حال، امروزه جدول مسیردهی برخی از روترها، حتی ستونهای بیشتری دارند. باید این را بدانیم که تعداد ستونهای جدول مسیردهی روتر بستگی به سازنده روتر دارد و همه روترها دارای همه ستونها نیستند.

Mask	Network address	Next-hop address	Interface	Flags	Reference count	Use
.....	.....	.....	.....	.....	.....	.....

شکل ۶-۱۸: فیلدهای متداول در یک جدول مسیردهی

Mask : مسک. این فیلد، مسکی که برای آن سطر از جدول مسیردهی به کار می‌رود را نشان می‌دهد.

Network address : آدرس شبکه. این فیلد، آدرس شبکه‌ای را تعریف می‌کند که بسته در نهایت به آن تحویل داده می‌شود. در حالت مسیردهی به روش کامپیوتر- مشخص، این فیلد، آدرس کامپیوتر مقصد را نشان می‌دهد.

Next-hop address : آدرس هاپ بعدی. این فیلد، آدرس IP روتر هاپ بعدی که بسته به آن تحویل داده می‌شود را نشان می‌دهد.

Interface : رابط. این فیلد، نام رابط را نشان می‌دهد.

Flags : پرچمها. این فیلد تا پنج پرچم را تعریف می‌کند. پرچمها، سویچهای روشن/ خاموش هستند و بود / نبود پرچم مربوطه را نشان می‌دهند. پنج پرچم عبارتند از U ، G ، H ، D و M .

الف- **U** (Up) : در حال کار. پرچم U نشان می‌دهد که روتر در حال کار است. اگر این پرچم، خاموش باشد (وجود نداشته باشد)، نشان می‌دهد که روتر کار نمی‌کند و لذا بسته را نمی‌توان هدایت کرد و باید آن را دور انداخت.

ب- **G** (Gateway) : گیت‌وی. پرچم G نشان می‌دهد که مقصد در شبکه دیگری است. برای آنکه بسته به مقصد برسد (تحویل غیرمستقیم)، باید به روتر هاپ بعدی داده شود.

www.Abiri.ir

وقتی که این پرچم، خاموش باشد، معنی آن این است که مقصد، در این شبکه است (تحویل مستقیم).

ج- **H** (Host - specific) : کامپیوتر- مشخص. پرچم H نشان می‌دهد که المان موجود در فیلد آدرس شبکه، یک آدرس کامپیوتر- مشخص است. وقتی که این پرچم، خاموش باشد، به این معنی است که آدرس، فقط آدرس شبکه مقصد است.

د- **D** (added by redirection) : توسط پیغام ICMP جهت‌دهی مجدد، اضافه شده است. پرچم D نشان می‌دهد که اطلاعات مسیردهی برای این مقصد، توسط یک پیغام ICMP ***جهت‌دهی مجدد***[1] به جدول مسیردهی کامپیوتر اضافه شده است. در فصل نهم درباره پروتوکل ICMP و پیغام فوق‌الذکر صحبت خواهیم کرد.

ه – **M** (Modified by redirection) : توسط پیغام ICMP جهت‌دهی مجدد، اصلاح شده است. پرچم M نشان می‌دهد که اطلاعات مسیردهی برای این مقصد، توسط یک پیغام ICMP جهت‌دهی مجدد، اصلاح شده است. در فصل نهم به بررسی پروتوکل ICMP و پیغام فوق‌الذکر خواهیم پرداخت.

Reference count : تعداد کاربران این مسیر. این فیلد، تعداد کاربرانی که در این لحظه از این مسیر استفاده می‌کنند را نشان می‌دهد. به عنوان مثال اگر پنج نفر به طور همزمان از طریق این روتر به یک وسیله متصل شده‌اند، مقدار این ستون برابر با 5 خواهد بود.

Use : میزان استفاده. این فیلد، تعداد بسته‌هایی که از طریق این روتر برای مقصد مربوطه ارسال شده‌اند را نشان می‌دهد.

* مثال ۱۳

در UNIX یا LINUX برنامه‌ای که می‌توان برای یافتن محتوای جدول مسیردهی یک کامپیوتر یا روتر به کار برد، netstat است. آنچه که در ذیل می‌آید، لیستی از محتوای سرور پیش‌فرض را نشان می‌دهد. ما از دو گزینه r و n استفاده کرده‌ایم. گزینه r نشان می‌دهد که می‌خواهیم محتوای جدول مسیردهی را ببینیم و گزینه n نشان می‌دهد که به دنبال آدرسهای عددی می‌گردیم. دقت نمایید که این، جدول مسیردهی یک کامپیوتر و نه یک روتر است. گرچه در طول این فصل درباره جدول مسیردهی روتر صحبت کردیم، یک کامپیوتر نیز حاوی جدول مسیردهی است.

1- Redirection

www.Abiri.ir

```
$ netstat -r n
Kernel IP routing table
```

Destination	Gateway	Mask	Flags	Iface
153.18.16.0	0.0.0.0	255.255.240.0	U	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	lo
0.0.0.0	153.18.31.254	0.0.0.0	UG	eth0

همچنین توجه نمایید که ترتیب ستونها، متفاوت با آن چیزی است که قبلاً نشان دادیم. در اینجا ستون **مقصد**[1] ، آدرس شبکه را تعریف می‌کند. کلمه gateway که یونیکس از آن استفاده می‌کند، مترادف با کلمه روتر است. این ستون در واقع آدرس IP هاپ بعدی را نشان می‌دهد.

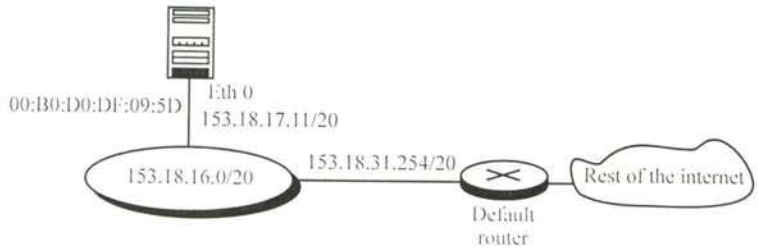
مقدار 0.0.0.0 نشان می‌دهد که تحویل به صورت مستقیم انجام می‌شود. آخرین المان یک پرچم G دارد و معنی آن این است که می‌توان از طریق یک روتر پیش‌فرض به مقصد رسید.

ستون Iface اینترفیس را مشخص می‌کند. کامپیوتر فقط یک اینترفیس واقعی دارد: eth0 و به معنی اینترفیس 0 است که به شبکه اترنت متصل شده است. اینترفیس دوم، lo ، در واقع یک اینترفیس ***برگردان***[2] مجازی است و نشان می‌دهد که کامپیوتر، بسته‌هایی با آدرس برگردان برابر با 127.0.0.0 را می‌پذیرد.

اطلاعات بیشتر درباره آدرس IP و آدرس فیزیکی سرور را می‌توان با دستور ifconfig روی اینترفیس داده شده (eth0) به دست آورد.

```
$ ifconfig eth0
eth0  Link encap:Ethernet Hwaddr 00:B0:D0:DF:09:5D
inet addr: 153.18.17.11 Bcast:153.18.31.255 Mask: 255.255.240.0
...
```

از اطلاعات داده شده می‌توان پیکربندی سرور را مطابق شکل ۶-۱۹ به دست آورد.



شکل ۶-۱۹: پیکربندی سرور برای مثال ۱۳

1- Destination
2- Loopback

www.Abiri.ir

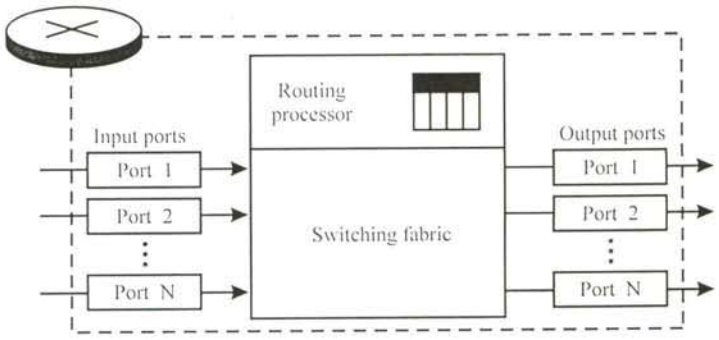
دقت نمایید که دستور ifconfig ، آدرس IP و آدرس فیزیکی اینترفیس را می‌دهد.

۶-۴ کلیاتی درباره روتر

در بحثمان درباره هدایت و مسیردهی بسته‌ها، یک روتر را به صورت یک جعبه نشان دادیم که از یکی از اینترفیسهای ورودی خود، بسته‌های ورودی را دریافت می‌کند و پس از به کارگیری یک جدول مسیردهی برای پیدا کردن اینترفیس خروجی که بسته‌ها باید از طریق آن به سمت مقصد خود هدایت شوند، بسته‌های دریافتی را روی خروجی مربوطه قرار می‌دهد. در این بخش، این جعبه را باز می‌کنیم و به داخل آن نگاه می‌کنیم. با این حال بحث ما خیلی مفصل نخواهد بود. کتابهایی درباره روترها نوشته شده‌اند. صرفاً یک بازنگری کلی درباره روتر را به خواننده ارائه می‌نمائیم.

اجزای یک روتر

می‌توانیم بگوئیم که یک روتر از چهار جزء ساخته شده است: ***پورتهای ورودی***[1] ، ***پورتهای خروجی***[2] ، ***پردازشگر مسیردهی***[3] و ***ساختار سویچینگ***[4] . شکل ۶-۲۰ را ببینید.



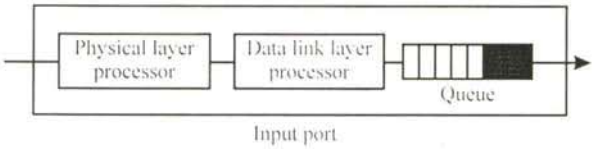
شکل ۶-۲۰: اجزای یک روتر

1- Input ports
2- Ourput ports
3- Routing processor
4- Switching fabric

www.Abiri.ir

❊ پورتهای ورودی

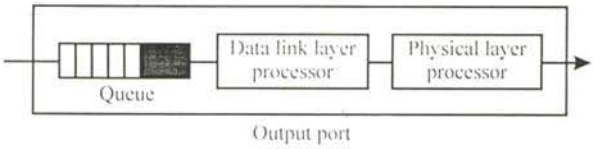
یک پورت ورودی، عملیات لایه‌های فیزیکی و دیتالینک روتر را انجام می‌دهد. بیتها از روی سیگنال ورودی ساخته می‌شوند و بسته، از کپسول فریم دریافتی استخراج می‌شود. خطاها آشکار و اصلاح می‌شوند و بسته، آماده مسیردهی توسط لایه شبکه می‌گردد. پورت ورودی علاوه بر پردازنده لایه فیزیکی و پردازنده لایه دیتالینک، بافرهایی (صفهایی) دارد تا قبل از ارسال بسته‌ها برای قسمت سویچینگ، آنها را در بافرها قرار دهد. شکل ۶-۲۱ دیاگرام ساده‌شده‌ای از یک پورت ورودی را نشان می‌دهد.



شکل ۶-۲۱: پورت ورودی

❊ پورتهای خروجی

یک پورت خروجی، همان عملیات پورت ورودی را انجام می‌دهد با این تفاوت که ترتیب عملیات در اینجا معکوس پورت ورودی است. ابتدا بسته‌های خروجی در صف خروجی قرار می‌گیرند و سپس بسته‌ها به داخل فریمهای خروجی، کپسول‌بندی می‌شوند و در نهایت عملیات لایه فیزیکی به فریم اعمال می‌شوند تا سیگنال قابل ارسال روی خط خروجی تولید شود. شکل ۶-۲۲ دیاگرام ساده شده‌ای از پورت خروجی را نشان می‌دهد.



شکل ۶-۲۲: پورت خروجی

❊ پردازشگر مسیردهی

پردازشگر مسیردهی، عملیات لایه شبکه را انجام می‌دهد. برای پیدا کردن آدرس هاپ بعدی، از آدرس مقصد استفاده می‌شود و در همان لحظه، شماره پورت خروجی که از آنجا بسته باید خارج شود نیز به دست می‌آید. گاهی اوقات به این عمل، ***جستجوی جدولی***[1] گفته

1- Table lookup

www.Abiri.ir

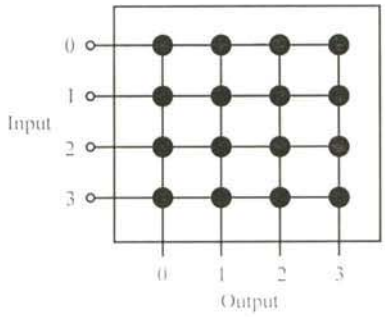
می‌شود زیرا که پردازشگر مسیردهی، داخل جدول مسیردهی را جستجو می‌کند. در روترهای جدیدتر، این عمل پردازشگر مسیردهی، جزء عملیات پورتهای ورودی است تا سرعت کار و سهولت انجام آن، بیشتر شود.

* سخت‌افزار سویچینگ

مشکلترین کار یک روتر این است[2] که بسته‌ها را از صف ورودی به صف خروجی منتقل کند. سرعت انجام این کار، سایز صفهای ورودی/ خروجی و تأخیر کلی در تحویل بسته‌ها را تحت تأثیر قرار می‌دهد. در گذشته، وقتی که یک روتر عملاً یک کامپیوتر خاص بود، حافظه کامپیوتر و یا یک ***باس***[1] به عنوان ساختار سویچینگ به کار می‌رفت. پورت ورودی بسته را داخل حافظه می‌گذاشت و پورت خروجی آن را از حافظه بر می‌داشت. امروزه روترها مکانیزمهای خاصی دارند که از انواع ساختارهای سویچینگ استفاده می‌کنند. در اینجا به بعضی از این ساختارها اشاره می‌کنیم.

** سویچ ضربدری

ساده‌ترین ساختار، ***سویچینگ ضربدری***[2] است که در شکل ۶-۲۳ نشان داده شده است. یک سویچ ضربدری، در یک آرایش شبکه مانند، با استفاده از میکروسویچهای الکترونیکی واقع در ***نقاط تلاقی***[3]، n ورودی را به n خروجی وصل می‌کند.



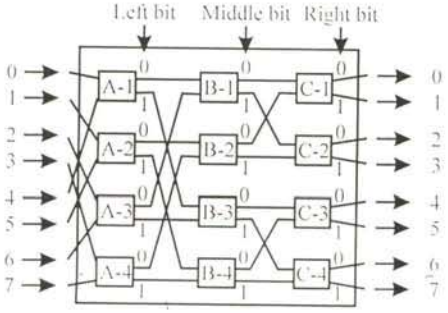
شکل ۶-۲۳: سویچ ضربدری

1- Bus
2- Crossbar switch
3- Crosspoint

www.Abiri.ir

** سویچ بَنیان

سویچی که پیاده‌سازی آن عملی‌تر از سوئیچ ضربدری است، ***سوئیچ بَنیان***[1] (بَنیان نام یک نوع انجیر است) می‌باشد. یک سویچ بَنیان، یک سویچ چندطبقه است و در هر طبقه، میکروسویچهایی واقع شده‌اند که بسته‌ها را بر اساس پورت خروجی که توسط یک دنباله باینری نمایش داده شده است، به سمت خروجی هدایت می‌کنند. برای n ورودی و n خروجی، لگاریتم n در مبنای 2 ، طبقه و در هر طبقه، n/2 میکروسویچ خواهیم داشت. اولین طبقه، بسته را بر اساس بیتی که در بالاترین مرتبه در دنباله باینری قرار گرفته است، به سمت خروجی هدایت می‌کند. دومین طبقه، بسته را بر اساس بیتی که در مرتبه دوم اهمیت در دنباله باینری واقع شده است، به سمت خروجی هدایت می‌کند. شکل ۶-۲۴ یک سوئیچ بَنیان با 8 ورودی و 8 خروجی را با سه طبقه (لگاریتم 8 در مبنای 2) نشان می‌دهد.

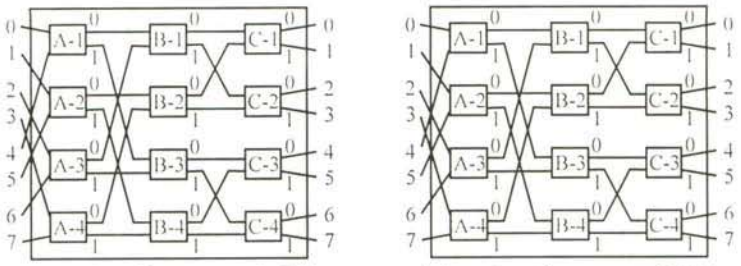


شکل ۶-۲۴: یک سوئیچ بَنیان

شکل ۶-۲۵ نحوه عمل سویچ را نشان می‌دهد. در قسمت a ، بسته‌ای به پورت ورودی شماره 1 رسیده است و باید به پورت خروجی 6 (110 بر حسب باینری) برود. اولین میکروسوئیچ (A-2) بسته را بر اساس اولین بیت (1) هدایت می‌کند. میکروسویچ دوم (B-4) بسته را بر اساس دومین بیت (1) هدایت می‌کند و سومین میکروسویچ (C-4) بسته را بر اساس سومین بیت (0) هدایت می‌نماید. در قسمت b ، بسته‌ای به پورت ورودی 5 رسیده است و باید به پورت خروجی 2 (010 برحسب باینری) برود. اولین میکروسوئیچ (A-2) بسته را بر اساس اولین بیت (0) هدایت می‌کند. میکروسویچ دوم (B-2) بسته را بر اساس دومین بیت (1) هدایت می‌کند و بالاخره میکروسویچ سوم (C-2) بسته را بر اساس بیت سوم (0) هدایت می‌کند.

1- Banyan switch

www.Abiri.ir

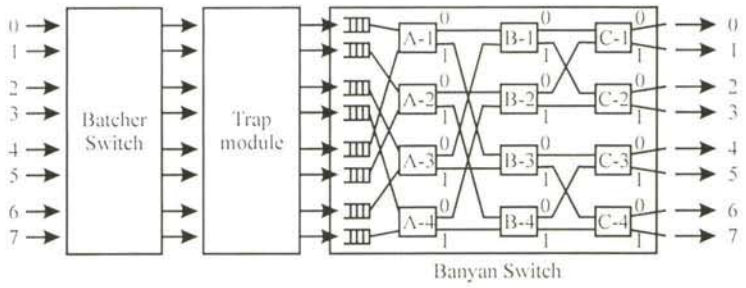


شکل ۶-۲۵: مثالهایی از نحوه جابجایی بسته‌ها در سوئیچ بَنیان

❊❊ سوئیچ باچر- بَنیان

مشکلی که سویچ بَنیان دارد این است که حتی اگر دو بسته به سمت خروجی یکسانی نروند، باز هم امکان تصادف داخلی بسته‌ها وجود دارد. می‌توانیم این مشکل را با مرتب‌سازی بسته‌های ورودی بر اساس پورت مقصد آنها برطرف کنیم.

K.E.Batcher سویچی را طراحی کرد که قبل از سویچ بَنیان قرار می‌گیرد و بسته‌های ورودی را بر اساس مقصد نهائی آنها مرتب می‌نماید. ترکیب این دو سویچ، ***سویچ باچر- بَنیان***[1] نامیده می‌شود. سویچ مرتب‌سازی از ادغام سخت‌افزاری استفاده می‌کند ولی ما درباره آن صحبت نمی‌کنیم. معمولاً مطابق شکل ۶-۲۶ ، ماجول سخت‌افزاری دیگری به نام ***تله***[2] بین سویچ باچر و سوئیچ بَنیان قرار داده می‌شود. ماجول تله مانع از آن می‌شود که بسته‌های هم خروجی (بسته‌هایی با مقصد خروجی مشابه) به طور همزمان وارد سویچ بَنیان شوند. در هر ***تیک***[3] فقط یک بسته برای هر خروجی اجازه عبور دارد. اگر بیش از یک بسته برای خروجی مشابهی وجود داشته باشند، باید تا تیک بعدی صبر کنند.



شکل ۶-۲۶: سویچ باچر- بَنیان

1- Batcher- banyan- switch
2- Trap
3- Tick

www.Abiri.ir

۶-۵ عبارات کلیدی

تجمع آدرس (address aggregation)
سویچ بَنیان (banyan switch)
سویچ باچر- بَنیان (Batcher- banyan switch)
سرویس فاقد اتصال (connectionless service)
سرویس اتصال‌گرا (connection –oriented service)
سویچ ضربدری (crossbar switch)
نقطه تلاقی (crosspoint)
روش پیش‌فرض (default method)
تحویل (delivery)
تحویل مستقیم (direct delivery)
متد مسیردهی دینامیک (dynamic routing method)
جدول مسیردهی دینامیک (dynamic routing table)
هدایت بسته IP (forwarding IP packet)
مسیردهی طبقاتی (hierarchical routing)
متد کامپیوتر- مشخص (host – specific method)
دستور ifconfig (ifconfig command)
تحویل غیرمستقیم (indirect delivery)
پورتهای ورودی (input ports)
مطابقت با بلندترین مسک (longest mask matching)
دستور netstat (netstat command)
متد شبکه مشخص (network- specific method)
متد هاپ بعدی (next hop method)
پورتهای خروجی (output ports)
مسیردهی (routing)
پردازشگر مسیردهی (routing processor)
جدول مسیردهی استاتیک (static routing table)
ساختار سویچینگ (switching fabric)

www.Abiri.ir

۶-۶ خلاصه

- در یک سرویس اتصال‌گرا، پروتوکل لایه شبکه مبدأ قبل از آنکه بسته‌ای را برای سایت مقصد ارسال کند، ابتدا اتصالی را با پروتوکل لایه شبکه مقصد برقرار می‌کند.
- در یک سرویس فاقد اتصال، پروتوکل لایه شبکه با هر بسته به طور مستقل از سایر بسته‌ها رفتار می‌کند و این طور فرض می‌کند که هیچ ارتباطی بین بسته‌ها وجود ندارد و بسته‌های یک پیغام ممکن است از مسیرهای متفاوتی به مقصد برسند. پروتوکل IP ، یک پروتوکل فاقد اتصال است.
- اگر کامپیوتر یا روتری که بسته را ارسال می‌کند به همان شبکه‌ای متصل باشد که کامپیوتر مقصد متصل است، به آن تحویل مستقیم گفته می‌شود.
- اگر کامپیوتر یا روتری که بسته را ارسال می‌کند به شبکه‌ای متفاوت با شبکه مقصد متصل باشد، به آن تحویل غیرمستقیم گفته می‌شود.
- در روش هاپ بعدی، به جای آنکه کل مسیری که بسته باید طی کند تا به مقصد برسد را در جدول مسیردهی بنویسیم، صرفاً آدرس هاپ بعدی را در جدول مسیردهی می‌گذاریم.
- در روش شبکه- مشخص، همه کامپیوترهای متصل به یک شبکه، از یک المان مشابه در جدول مسیردهی، استفاده می‌کنند.
- در روش کامپیوتر- مشخص، آدرس کامل IP یک وسیله در جدول مسیردهی نوشته می‌شود.
- در روش پیش‌فرض، تمام بسته‌هایی که در جدول مسیردهی، تطابقی برای آنها پیدا نمی‌شود به سمت یک روتر پیش‌فرض روانه می‌شوند.
- جدول مسیردهی برای هدایت بسته‌های IP بر اساس آدرس‌دهی کلاس‌دار، می‌تواند سه ستون داشته باشد.
- جدول مسیردهی برای هدایت بسته‌های IP بر اساس آدرس‌دهی بدون‌کلاس، به حداقل چهار ستون نیاز دارد.
- تعداد ستونها در یک جدول مسیردهی، بستگی به نظر سازنده روتر دارد.
- در آدرس‌دهی بدون‌کلاس، تجمع آدرس، عمل هدایت بسته‌ها را ساده‌تر می‌کند.
- مطابقت با بلندترین مسک در آدرس‌دهی بدون‌کلاس، مورد نیاز است.
- آدرس‌دهی بدون‌کلاس به مسیردهی طبقاتی و جغرافیایی نیاز دارد تا مانع از ایجاد جداول بسیار بزرگ شود.
- الگوریتمهای جستجو برای آدرس‌دهی بدون‌کلاس، برای آدرس‌دهی کلاس‌دار، کارآمد نیستند.

www.Abiri.ir

- محتوای یک جدول مسیردهی استاتیک به صورت دستی توسط مدیر شبکه به روزرسانی می‌شود.
- محتوای جدول مسیردهی دینامیک به صورت اتوماتیک توسط یک پروتوکل مسیردهی به روزرسانی می‌شود.
- معمولاً یک روتر از چهار جزء ساخته می‌شود. پورتهای ورودی، پورتهای خروجی، پردازشگر مسیردهی و ساختار سویچینگ.

۶-۷ مجموعه تمرین‌ها

تمرین‌ها

۱- کامپیوتری با آدرس IP برابر با 137.23.56.23/16 بسته‌ای را برای کامپیوتری با آدرس IP برابر با 137.23.67.9/16 می‌فرستد. آیا این تحویل از نوع مستقیم یا غیرمستقیم است؟ فرض کنید زیرشبکه‌سازی نداریم.

۲- کامپیوتری با آدرس IP برابر با 137.23.56.23/16 بسته‌ای را برای کامپیوتری با آدرس IP برابر با 142.3.6.9/24 می‌فرستد. آیا این تحویل از نوع مستقیم یا غیرمستقیم است؟ فرض کنید زیرشبکه‌سازی نداریم.

۳- در شکل ۶-۸، جدول مسیردهی برای روتر R2 را پیدا کنید.

۴- در شکل ۶-۸، جدول مسیردهی برای روتر R3 را پیدا کنید.

۵- بسته‌ای به آدرس مقصد 192.16.7.42 مطابق شکل ۶-۸ به روتر R1 می‌رسد. نشان دهید که چگونه هدایت می‌شود.

۶- بسته‌ای به آدرس مقصد 145.80.14.26 مطابق شکل ۶-۸ به روتر R1 می‌رسد. نشان دهید که چگونه هدایت می‌شود.

۷- بسته‌ای به آدرس مقصد 147.26.50.30 مطابق شکل ۶-۸ به روتر R1 می‌رسد. نشان دهید که چگونه هدایت می‌شود.

۸- بسته‌ای به آدرس مقصد 145.14.192.71 مطابق شکل ۶-۱۱ به روتر می‌رسد. نشان دهید که چگونه هدایت می‌شود.

۹- بسته‌ای به آدرس 135.11.80.21 مطابق شکل ۶-۱۱ به روتر می‌رسد. نشان دهید که چگونه هدایت می‌شود.

۱۰- بسته‌ای به آدرس مقصد 201.4.16.70 مطابق شکل ۶-۱۳ به روتر R1 می‌رسد. نشان دهید که چگونه هدایت می‌شود.

۱۱- بسته‌ای به آدرس مقصد 202.70.20.30 مطابق شکل ۶-۱۳ به روتر R1 می‌رسد. نشان دهید که چگونه هدایت می‌شود.

www.Abiri.ir

۱۲- یک جدول مسیردهی برای کامپیوتری را نشان دهید که کاملاً از بقیه جدا شده است.
۱۳- جدول مسیردهی برای کامپیوتری را نشان دهید که بدون آنکه به اینترنت متصل باشد، به یک LAN متصل شده است.
۱۴-اگر جدول ۶-۳، جدول مسیردهی روتر R1 باشد، توپولوژی شبکه را به دست آورید.

Mask	Network Address	Next-Hop Address	Interface
/27	202.14.17.224	----------	m1
/18	145.23.192.0	----------	m0
Default	Default	130.56.12.4	m2

جدول ۶-۳: جدول مسیردهی برای تمرین ۱۴

۱۵- آیا روتر R1 در شکل ۶-۱۶ می‌تواند بسته‌ای با آدرس مقصد 140.24.7.194 را دریافت کند؟ پاسخ خود را توضیح دهید.
۱۶-آیا روتر R1 در شکل ۶-۱۶، می‌تواند بسته‌ای با آدرس مقصد 140.24.7.42 را دریافت کند؟ پاسخ خود را توضیح دهید.
۱۷-جدول مسیردهی ISP ناحیه‌ای شکل ۶-۱۷ را نشان دهید.
۱۸- جدول مسیردهی ISP محلی شکل ۶-۱۷ را نشان دهید.
۱۹- جدول مسیردهی ISP محلی شماره ۲ در شکل ۶-۱۷ را نشان دهید.
۲۰- جدول مسیردهی ISP محلی شماره ۳ شکل ۶-۱۷ را نشان دهید.
۲۱- جدول مسیردهی ISP کوچک شماره ۱ در شکل ۶-۱۷ را نشان دهید.

فعالیتهای تحقیقاتی

۲۲- اگر به یونیکس (یا لینوکس) دسترسی دارید ، از فرمان netstat استفاده کنید و جدول مسیردهی سروری که به آن متصل هستید را پیدا کنید.
۲۳- اگر به یونیکس (یا لینوکس) دسترسی دارید، از فرمان ifconfig استفاده کنید و مشخصات اینترفیسی از سرور که به آن متصل هستید را پیدا کنید.
۲۴- پیدا کنید چگونه ISP شما از قواعد تجمع آدرس و مطابقت با بلندترین مسک استفاده می‌کند.
۲۵- پیدا کنید که آیا آدرس IP شما بخشی از تخصیص آدرس جغرافیایی است یا نه.
۲۶- RFC های مربوط به تجمع آدرس را پیدا کنید.
۲۷-RFC های مربوط به مطابقت با بلندترین مسک را پیدا کنید.

www.Abiri.ir

۲۸- اگر از یک روتر استفاده می‌کنید، تعداد و نام ستونهای جدول مسیردهی آن را پیدا کنید.
۲۹- cisco یکی از تولیدکنندگان اصلی روترهاست. اطلاعاتی درباره انواع روترهایی که توسط این شرکت ساخته می‌شوند را پیدا کنید.

www.Abiri.ir

فصل هفتم

پروتوکل‌های ARP و RARP

اینترنت، مجموعه‌ای از شبکه‌های فیزیکی است که توسط وسایل شبکه‌بندی مانند روترها به هم متصل شده‌اند. بسته‌ای که حرکت خود را از یک کامپیوتر مبدأ آغاز می‌کند، ممکن است قبل از آنکه نهایتاً به کامپیوتر مقصد برسد، از میان چندین شبکه فیزیکی متفاوت عبور کند.

کامپیوترها و روترها در سطح لایه شبکه، توسط آدرس منطقی آنها شناخته می‌شوند. یک ***آدرس منطقی***[1]، یک آدرس شبکه‌بندی است. محدوده عمل آن به صورت سرتاسری است و منحصر به فرد نیز می‌باشد. از آنجا که یک آدرس منطقی غالباً به صورت نرم‌افزاری پیاده‌سازی می‌شود، نام آن را آدرس منطقی گذاشته‌اند. هر پروتوکلی که با شبکه‌بندی شبکه‌ها سروکار داشته باشد، به آدرسهای منطقی نیاز دارد. آدرسهای منطقی در پشته پروتوکل TCP/IP ، ***آدرسهای IP***[2] نامیده می‌شوند و 32 بیتی هستند.

با این حال، بسته‌ها برای رسیدن به این کامپیوترها و روترها باید از میان شبکه‌های فیزیکی عبور کنند. در سطح فیزیکی، کامپیوترها و روترها توسط آدرس فیزیکی آنها شناخته می‌شوند. یک ***آدرس فیزیکی***[3] ، یک آدرس محلی است. محدوده عمل آن در حد یک شبکه محلی است. منحصر به فرد بودن آن به صورت محلی، کافی است و لازم نیست که به صورت سرتاسری منحصر به فرد باشد. از آنجا که یک آدرس فیزیکی، غالباً (نه همیشه) به صورت سخت‌افزاری پیاده‌سازی می‌شود، به آن آدرس فیزیکی گفته می‌شود. مثالهایی از آدرسهای فیزیکی، MAC آدرسهای 48 بیتی در پروتوکل اترنت هستند که در کارت شبکه‌ای که در کامپیوتر نصب می‌شود، حک شده‌اند.

آدرسهای فیزیکی و منطقی، دو شاخص متفاوت هستند. ما به هر دو آنها نیاز داریم، زیرا که یک شبکه فیزیکی مانند اترنت می‌تواند در سطح لایه شبکه، دو پروتوکل مانند IP و IPX (در شبکه‌های ناوِل) را به طور همزمان داشته باشد. به طور مشابه، بسته‌ای در سطح لایه شبکه مانند IP ممکن است از شبکه‌های فیزیکی مختلفی مانند اترنت و LocalTalk (apple) عبور کند.

1- Logical address
2- IP addresses
3- Physical address

www.Abiri.ir

معنای آن این است که تحویل یک بسته به کامپیوتر یا یک روتر، به دو سطح آدرس‌دهی نیاز دارد: منطقی و فیزیکی. بنابراین لازم است بتوانیم آدرس منطقی را به آدرس فیزیکی متناظر با آن و بالعکس بنگاریم. این کارها را می‌توان هم به کمک ***نگاشت استاتیک***[1] و هم به کمک ***نگاشت دینامیک***[2] انجام دهیم.

نگاشت استاتیک به این معناست که جدولی بسازیم و در آن هر آدرس منطقی را به یک آدرس فیزیکی، مرتبط کنیم. این جدول باید در هر ماشینی در شبکه، نصب شود. هر ماشینی که مثلاً آدرس IP ماشین دیگری را می‌داند اما آدرس فیزیکی آن را نمی‌داند، می‌تواند آدرس فیزیکی آن ماشین را در این جدول پیدا کند. این روش، محدودیتهایی دارد زیرا آدرسهای فیزیکی ممکن است به طرق زیر تغییر کنند:

۱- ممکن است کارت شبکه یک ماشین تغییر کند و لذا آدرس فیزیکی آن عوض شود.

۲- در بعضی از LANها مانند LocalTalk، هر زمان که ماشینی روشن می‌شود، آدرس فیزیکی آن تغییر می‌کند.

۳- یک کامپیوتر متحرک می‌تواند از یک شبکه فیزیکی به شبکه فیزیکی دیگری برود و لذا آدرس فیزیکی آن تغییر کند.

برای پیاده‌سازی این تغییرات، باید یک جدول مسیردهی استاتیک را متناوباً به روزرسانی کنیم. اینگونه عملیات اضافی می‌توانند بر عملکرد شبکه اثر منفی گذارند.

در نگاشت دینامیک، اگر ماشین، یکی از دو نوع آدرس (منطقی یا فیزیکی) را بداند، می‌تواند از پروتوکلی استفاده کند و دیگری را به دست آورد.

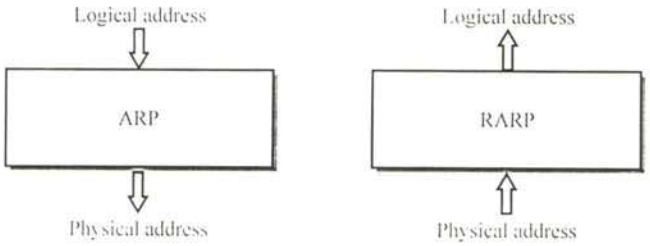
دو پروتوکل برای نگاشت دینامیک طراحی شده است: ***پروتوکل پیدا کردن آدرس***[3] و ***پروتوکل معکوس ARP***[4]. ARP یک آدرس منطقی را به یک آدرس فیزیکی می‌نگارد و RARP یک آدرس فیزیکی را به یک آدرس منطقی می‌نگارد. شکل ۷-۱ این ایده را نشان می‌دهد.

ARP , RARP از آدرسهای فیزیکی تک‌گیرنده و همگانی استفاده می‌کنند. در فصل سوم درباره آدرسهای فیزیکی تک‌گیرنده و همگانی صحبت کردیم. به عنوان مثال اشاره کردیم که اترنت از آدرس تمام 1 (FF:FF:FF:FF:FF:FF) به عنوان آدرس همگانی استفاده می‌کند.

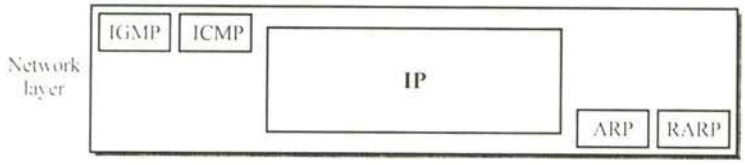
شکل ۷-۲ مکان پروتوکلهایARP و RARP را در پشته پروتوکل TCP/IP نشان می‌دهد.

1- Static mapping
2- Dynamic mapping
3- Address Resolution Protocol : ARP
4- Reverse Address Resolution Protocol : RARP

www.Abiri.ir



شکل ۷-۱: RARP , ARP



شکل ۷-۲: مکان ARP و RARP در پشته پروتوکل TCP/IP

۷-۱ پروتوکل ARP

هر وقت که یک کامپیوتر یا یک روتر، یک بسته IP برای ارسال به کامپیوتر یا روتر دیگری داشته باشد، از آدرس منطقی (IP) گیرنده که در اختیار دارد باید استفاده کند. اما بسته IP باید داخل یک فریم کپسول‌بندی شود تا بتواند از میان شبکه فیزیکی عبور کند. معنی آن این است که فرستنده، به آدرس فیزیکی گیرنده نیز نیاز دارد. یک نگاشت، آن آدرس منطقی را به یک آدرس فیزیکی می‌نگارد. همانطور که قبلاً گفتیم، می‌توان این کار را هم به صورت استاتیک و هم دینامیک انجام داد. رابطه بین آدرسهای فیزیکی و منطقی را می‌توان به صورت استاتیک داخل یک جدول نوشت. فرستنده می‌تواند به جدول نگاه کند و آدرس فیزیکی متناظر با یک آدرس منطقی را پیدا کند. ولی همانطور که قبلاً بحث آن آمد، این راه حل خوبی نیست. هر وقت که یک آدرس فیزیکی تغییر کند، جدول باید به روزرسانی شود. به روزرسانی جداول در تمام ماشینها در فاصله زمانیهای مناسب، کار سنگینی است.

با این حال می‌توان نگاشت را به صورت دینامیک انجام داد به این ترتیب که هر زمان فرستنده به آدرس فیزیکی گیرنده نیاز داشته باشد، آن را از گیرنده سؤال کند. ARP برای تحقق همین خواسته، طراحی شده است.

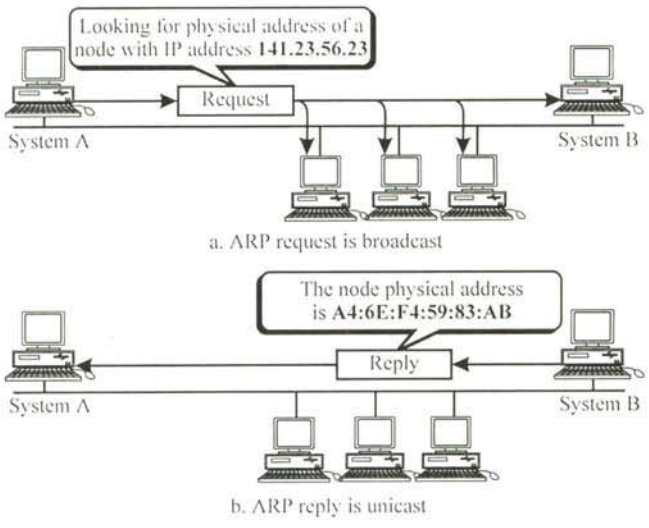
www.Abiri.ir

ARP یک آدرس IP را به یک آدرس فیزیکی می‌نگارد. در یک شبکه فیزیکی معمولی مانند یک LAN ، هر وسیله متصل به شبکه توسط یک آدرس فیزیکی که معمولاً روی کارت شبکه آن وسیله حک شده است، مشخص می‌شود.

هر وقت که یک کامپیوتر یا یک روتر نیاز به آدرس فیزیکی کامپیوتر یا روتر دیگری که به همان شبکه متصل است، داشته باشد، یک بسته درخواست ARP را ارسال می‌کند. این بسته حاوی آدرسهای فیزیکی و IP فرستنده و آدرس IP گیرنده است. از آنجا که فرستنده، آدرس فیزیکی گیرنده را نمی‌داند، درخواست ARP به صورت همگانی روی شبکه ارسال می‌شود.

هر کامپیوتر یا روتر متصل به شبکه، بسته درخواست ARP را دریافت و پردازش می‌کند، ولی فقط گیرنده مورد نظر، آدرس IP خود را در بسته می‌یابد و لذا یک بسته پاسخ ARP را برای فرستنده درخواست ARP، ارسال می‌کند. پاسخ ARP حاوی آدرسهای فیزیکی و IP گیرنده مورد نظر است. اینک فرستنده‌ای که درخواست ARP را ارسال کرده بود، می‌تواند با داشتن آدرس فیزیکی گیرنده مورد نظر، بسته IP خود را در فریم لایه دیتالینک، کپسول‌بندی کند و آن را برای گیرنده، بفرستد.

در شکل ۷-۳ الف، سیستم سمت چپ (A) بسته‌ای دارد که باید برای سیستم دیگری (B) به آدرس IP برابر با 141.23.56.23 ارسال نماید. سیستم A باید برای ارسال بسته، آن را تحویل لایه دیتالینک بدهد، اما آدرس فیزیکی گیرنده را نمی‌داند.



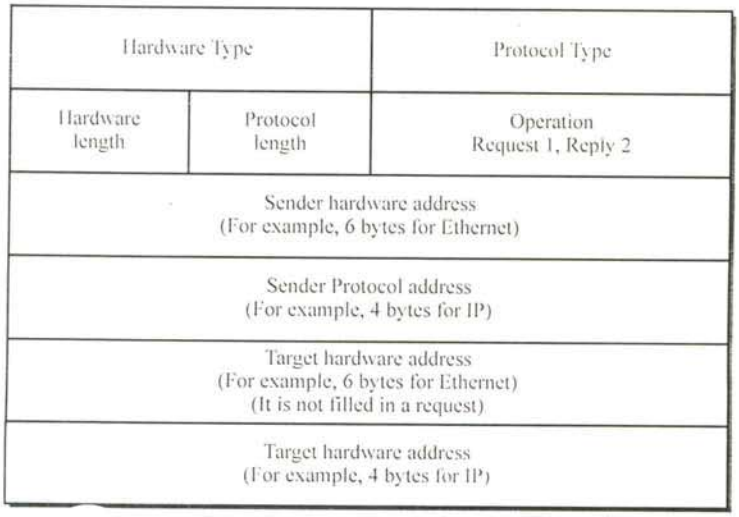
شکل ۷-۳: نحوه عملکرد ARP

www.Abiri.ir

او از سرویس ARP استفاده می‌کند و ARP یک بسته درخواست ARP را به صورت همگانی ارسال می‌کند و آدرس فیزیکی سیستمی با آدرس IP برابر با 141.23.56.23 را سؤال می‌کند. همانطور که شکل ۷-۳ ب نشان می‌دهد، این بسته توسط همه سیستمهای متصل به شبکه دریافت می‌شود ولی فقط سیستم B پاسخ می‌دهد. حال سیستم A به کمک آدرس فیزیکی که دریافت کرده است، می‌تواند بسته‌اش را برای مقصد (B) ارسال کند.

فرمت بسته ARP

شکل ۷-۴، فرمت یک بسته ARP را نشان می‌دهد.



شکل ۷-۴: فرمت بسته ARP

فیلدهای بسته ARP عبارتند از:

Hardware type : نوع سخت‌افزار. این یک فیلد 16 بیتی است و نوع شبکه‌ای را مشخص می‌کند که بسته‌های ARP روی آن ارسال می‌شوند. به هر یک از شبکه‌های فیزیکی، یک عدد صحیح اختصاص داده شده است مثلاً عدد تخصیص یافته به اترنت، برابر 1 است. ARP را می‌توان روی هر شبکه فیزیکی به کار برد.

Protocol type : نوع پروتوکل. این یک فیله 16 بیتی است که پروتوکل لایه شبکه را مشخص می‌کند. به عنوان مثال، مقدار این فیلد برای پروتوکل IPv4 برابر با 0800_{16} است. می‌توان ARP را با هر نوع پروتوکلی در لایه شبکه، به کار برد.

www.Abiri.ir

Hardware length : طول سخت‌افزار. این یک فیله 8 بیتی است که طول آدرس فیزیکی را برحسب بایت نشان می‌دهد. به عنوان مثال، مقدار این فیلد برای اترنت برابر با 6 است.

Protocol length : طول پروتوکل. این یک فیلد 8 بیتی است که طول آدرس منطقی را بر حسب بایت نشان می‌دهد. به عنوان مثال، مقدار این فیلد برای پروتوکل IPv4 برابر با 4 است.

Operation : نوع پیغام. این یک فیله 16 بیتی است و نوع پیغام ARP را مشخص می‌کند. ARP دو نوع پیغام دارد: درخواست ARP (1) و پاسخ ARP (2).

Sender hardware address : آدرس سخت‌افزاری فرستنده. این یک فیلد با طول متغیر است و آدرس فیزیکی فرستنده را مشخص می‌کند. به عنوان مثال، آدرس فیزیکی اترنت، 6 بایتی است و لذا طول این فیلد برای اترنت، برابر با 6 بایت می‌باشد.

Sender Protocol address : آدرس پروتوکل فرستنده. این یک فیلد با طول متغیر است و آدرس منطقی (به عنوان مثال آدرس IP) فرستنده را مشخص می‌کند. برای پروتوکل IP، طول این فیلد برابر با 4 بایت است.

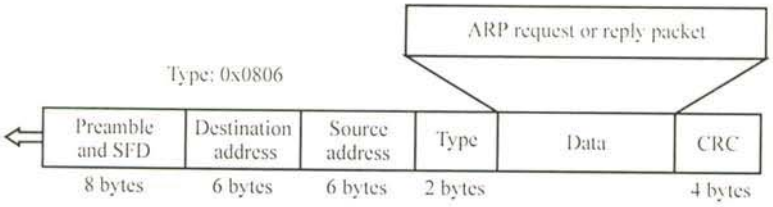
Target hardware address : آدرس سخت‌افزاری هدف. این یک فیلد با طول متغیر است و آدرس فیزیکی گیرنده را مشخص می‌کند. به عنوان مثال برای اترنت، طول این فیلد برابر با 6 بایت است. در یک پیغام درخواست ARP، این فیلد برابر با تمام صفر است زیرا که فرستنده، آدرس فیزیکی گیرنده را نمی‌داند.

Target Protocol Address : آدرس پروتوکل هدف. این یک فیلد به طول متغیر است و آدرس منطقی (به عنوان مثال آدرس IP) گیرنده را مشخص می‌کند. برای پروتوکل IP ، طول این فیلد برابر با 4 بایت است.

کپسول‌بندی

بسته ARP مستقیماً داخل فریم لایه دیتالینک کپسول‌بندی می‌شود. به عنوان مثال در شکل ۷-۵ یک بسته ARP داخل یک فریم اترنت، کپسول‌بندی می‌شود. دقت نمایید که فیلد type (نوع) موجود در فریم نشان می‌دهد که فریم، حاوی بسته ARP است (مقدار آن برای بسته ARP برابر با 0806_{16} و یا 0x0806 می‌باشد).

www.Abiri.ir



شکل ۷-۵: کپسول‌بندی یک بسته ARP

طرز کار پروتوکل ARP

اجازه دهید ببینیم چگونه ARP روی یک شبکه نوعی عمل می‌کند. ابتدا قدم به قدم کار ARP را شرح می‌دهیم. سپس چهار مورد را بیان می‌کنیم که کامپیوتر یا روتر در آن چهار مورد، نیاز به استفاده از ARP دارند.

فرایند استفاده از سرویس ARP شامل مراحل زیر است :

۱- فرستنده، آدرس IP ماشین گیرنده را می‌داند. کمی بعد خواهیم دید که فرستنده چگونه این اطلاعات را به دست می‌آورد.

۲- ماجول IP ، آدرس فیزیکی فرستنده، آدرس IP فرستنده و آدرس IP گیرنده را در اختیار پروتوکل ARP می‌گذارد و از او می‌خواهد یک بسته درخواست ARP ارسال نماید. فیلد آدرس فیزیکی گیرنده، با تمام صفر پر می‌شود.

۳- پیغام درخواست ARP به لایه دیتالینک داده می‌شود و در آنجا در فریمی کپسول‌بندی می‌شود که آدرس فیزیکی مبدأ آن، همان آدرس فیزیکی فرستنده و آدرس فیزیکی مقصد آن، آدرس فیزیکی همگانی است.

۴- هر کامپیوتر یا روتر متصل به شبکه، فریم ارسالی را دریافت می‌کند. از آنجا که فریم حاوی آدرس همگانی است، همه ایستگاهها آن را دریافت می‌کنند و پیغام داخل آن را به پروتوکل ARP خود می‌دهند. همه ماشینها به جز گیرنده مورد نظر، درخواست ARP را دور می‌اندازند زیرا که به جز گیرنده مورد نظر، هیچکدام از ماشینها، آدرس IP خود را داخل آن نمی‌بینند.

۵- گیرنده مورد نظر، با یک پیغام پاسخ ARP که حاوی آدرس فیزیکی اوست، پاسخ می‌دهد. پیغام به صورت تک‌گیرنده برای فرستنده درخواست ARP ارسال می‌شود.

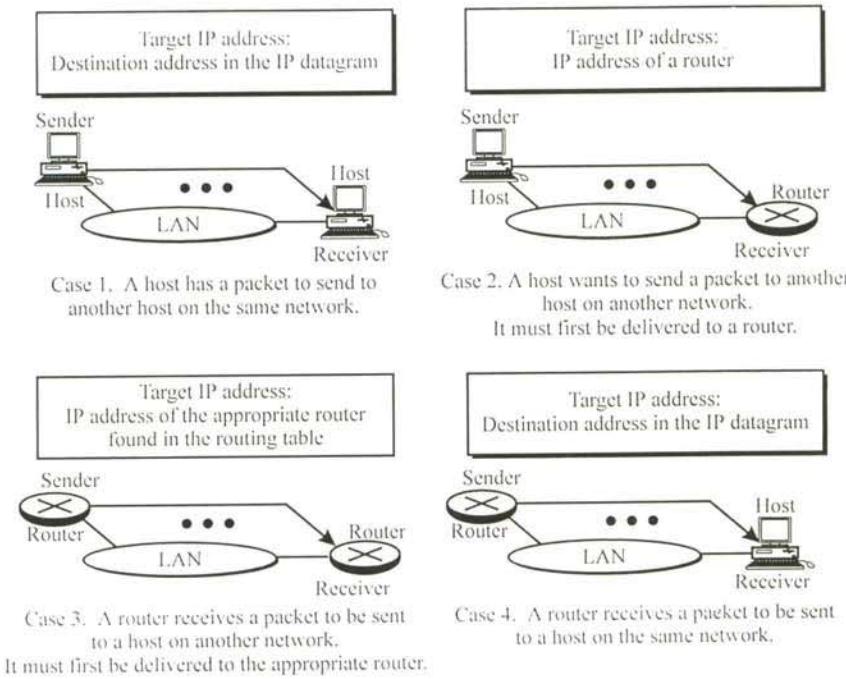
۶- فرستنده درخواست ARP، پاسخ ARP را دریافت می‌کند. حالا او آدرس فیزیکی گیرنده مورد نظر را می‌داند.

www.Abiri.ir

۷- حالا فرستنده می‌تواند بسته IP خود را داخل فریمی به آدرس فیزیکی گیرنده مورد نظر قرار دهد و آن را به صورت تک‌گیرنده برای گیرنده، ارسال کند.

چهار مورد مختلف به کارگیری ARP

چهار مورد مختلفی که می‌توان در آنها از سرویس ARP استفاده کرد را در شکل ۷-۶ می‌بینید.



شکل ۷-۶ : چهار مورد مختلف استفاده از ARP

۱- فرستنده، کامپیوتری است که می‌خواهد بسته‌ای را برای کامپیوتر دیگری روی همان شبکه ارسال نماید. در این حالت، آدرس منطقی که باید به یک آدرس فیزیکی نگاشته شود، آدرس IP مقصد است که در هدر IP ، در محل آدرس مقصد، نوشته شده است.

۲- فرستنده، کامپیوتری است که می‌خواهد بسته‌ای را برای کامپیوتر دیگری که در شبکه دیگری واقع شده است، ارسال نماید. در این حالت، کامپیوتر به جدول مسیردهی خود نگاه می‌کند تا آدرس IP روتر بعدی برای رسیدن به این مقصد را به دست آورد. اگر جدول

www.Abiri.ir

مسیردهی نداشته باشد، به دنبال آدرس IP روتر پیش‌فرض می‌گردد. آدرس IP روتر، همان آدرس IP ای می‌شود که باید آدرس فیزیکی متناظر با آن به کمک ARP به دست آید.

۳- فرستنده، روتری است که یک بسته IP را دریافت کرده است و این بسته IP باید به سمت مقصدی مسیردهی شود که در شبکه دیگری واقع شده است. او به داخل جدول مسیردهی خود نگاه می‌کند و آدرس IP روتر بعدی برای مسیردهی این بسته به سمت مقصدش را پیدا می‌کند. آدرس IP روتر بعدی، همان آدرس IP ای می‌شود که باید آدرس فیزیکی متناظر با آن به کمک پروتوکل ARP به دست آید.

۴- فرستنده، روتری است که یک بسته IP را دریافت کرده است و این بسته IP باید تحویل کامپیوتری گردد که روتر مستقیماً به شبکه حاوی آن کامپیوتر متصل است. در این حالت آدرس IP مقصد موجود در بسته IP دریافتی، همان آدرس IP است که باید آدرس فیزیکی متناظر با آن را به دست آورد.

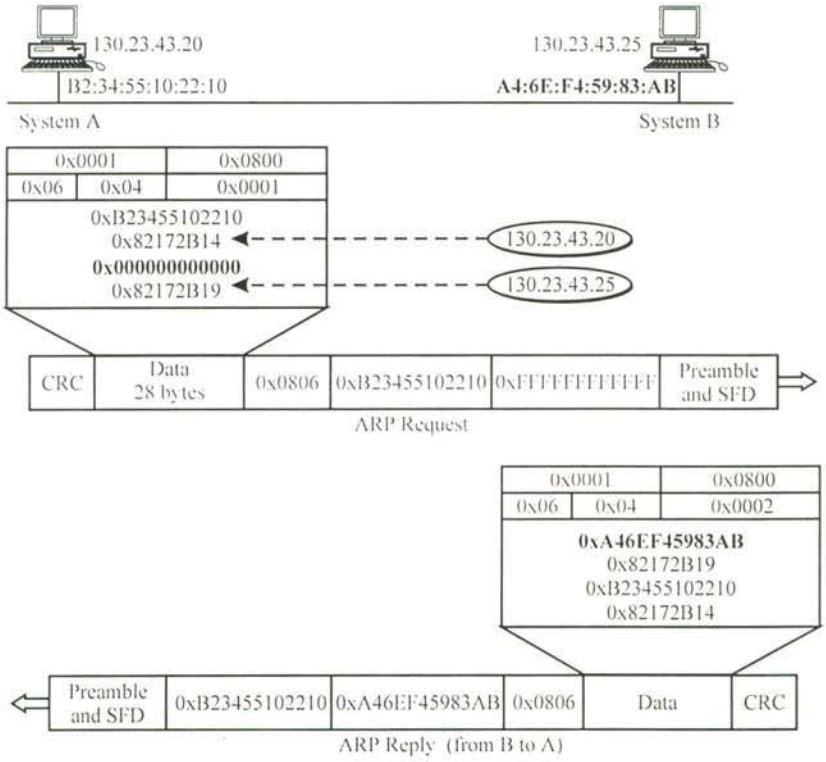
مهم: درخواست ARP به صورت همگانی ارسال می‌شود. پاسخ ARP به صورت تک‌گیرنده ارسال می‌شود.

*** مثال ۱**

یک کامپیوتر با آدرس IP برابر با 130.23.43.20 و آدرس فیزیکی برابر با B2:34:55:10:22:10 ، بسته‌ای برای ارسال به کامپیوتر دیگری با آدرس IP برابر با 130.23.43.25 و آدرس فیزیکی برابر با A4:6E:F4:59:83:AB دارد (و کامپیوتر ارسال کننده بسته، این آدرس فیزیکی را نمی‌داند). این دو کامپیوتر روی شبکه اترنت مشابهی واقع شده‌اند. بسته‌های درخواست و پاسخ ARP که در فریم اترنت کپسول‌بندی می‌شوند را نشان دهید.

راه حل : شکل ۷-۷ بسته‌های درخواست و پاسخ ARP را نشان می‌دهد. دقت نمایید که طول دیتای فریم اترنت در ارسال بسته‌های ARP برابر با 28 بایت است. به همین دلیل، 18 صفر باید به دنبال آن اضافه شوند تا حداقل طول آن را به 46 بایت برسانند. همچنین توجه نمایید که آدرسهای فیزیکی در محدوده 4 بایت قرار نمی‌گیرند زیرا که فیلد مربوط به آنها 6 بایتی است و به همین دلیل برای نمایش این آدرسها در شکل ۷-۷ از مرزبندی 4 بایتی استفاده نکرده‌ایم. نکته بعدی آنکه آدرسهای IP به صورت هگزادسیمال نشان داده شده‌اند.

www.Abiri.ir



شکل ۷-۷: مثال ۱

ARP روی ATM

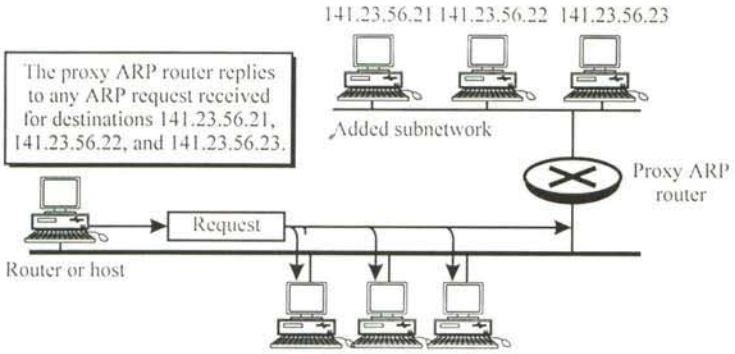
وقتی که می‌خواهیم یک بسته IP را در یک شبکه ATM جابه‌جا کنیم، در این مورد نیز از ARP استفاده می‌کنیم. وقتی که درباره IP روی ATM (IP over ATM) در فصل بیست و سوم صحبت کردیم، به این موضوع نیز خواهیم پرداخت.

Proxy ARP

برای ایجاد حالتی مانند زیرشبکه‌سازی می‌توان از Proxy ARP استفاده کرد. Proxy ARP یک ARP است که به صورت نماینده یک سری کامپیوتر عمل می‌کند. هر وقت روتری که یک Proxy ARP را اجرا می‌کند، یک درخواست ARP را دریافت نماید که حاوی آدرس IP یکی از این کامپیوترها باشد و آدرس فیزیکی متناظر با آن را بخواهد، روتر حاوی Proxy ARP ، آدرس فیزیکی خود را به صورت یک پاسخ ARP ارسال می‌کند. وقتی که روتر، بسته IP را دریافت کرد، آن را برای کامپیوتر مورد نظر ارسال می‌کند.

www.Abiri.ir

اجازه دهید مثالی بزنیم. در شکل ۷-۸ ، ARPای که روی کامپیوتر سمت راست در حال اجراست فقط به درخواست ARP حاوی آدرس IP برابر با 141.23.56.23 پاسخ می‌دهد.



شکل ۷-۸ : Proxy ARP

با این حال ممکن است مدیر شبکه بخواهد بدون تغییر دادن کل سیستم، یک ویژگی مشابه با ویژگی زیرشبکه‌سازی، به سیستم بدهد. یک راه حل آن است که روتری حاوی Proxy ARP به شبکه اضافه کند. در این حالت، این روتر به عنوان نماینده همه کامپیوترهایی که در شبکه واقع شده‌اند، عمل خواهد کرد. وقتی که روتر یک درخواست ARP حاوی آدرس IP یکی از کامپیوترها (141.23.56.23 , 141.23.56.22 , 141.23.56.21) را دریافت می‌کند، یک پاسخ ARP ارسال می‌کند و آدرس فیزیکی خود را در آن اعلام می‌نماید. وقتی که روتر، بسته IP را دریافت کرد، آن را برای کامپیوتر مربوطه ارسال می‌کند.

۷-۲ نرم‌افزار ARP

در این بخش، نمونه‌ای از نرم‌افزار ساده شده ARP را ارائه می‌نماییم. هدف آن است که اجزای یک نرم‌افزار فرضی ARP و رابطه بین آنها را نشان دهیم.

www.Abiri.ir

اجزای نرم‌افزار ARP

می‌توانیم بگوییم نرم‌افزار ARP حاوی پنج جزء است: ***جدول حافظه نهان***[1]، ***صفها***[2]، ***ماجول خروجی***[3]، ***ماجول ورودی***[4] و ***ماجول کنترل کننده حافظه نهان***[5] (به این دلیل به حافظه cache، حافظه نهان گفته می‌شود که CPU چیزی را داخل آن، ذخیره یا نهان می‌کند تا اگر مجدداً به آن نیاز پیدا کرد، بدون آنکه سراغ حافظه اصلی سیستم برود، از همان جا آن را بردارد و لذا این یک حافظه نهان است که در اختیار CPU می‌باشد). شکل ۷-۹ این پنج جزء و رابطه بین آنها را نشان می‌دهد. نرم‌افزار ARP ، یک بسته IP را دریافت می‌کند که برای کپسول‌بندی به داخل فریم لایه دیتالینک، نیاز به آدرس فیزیکی مقصد دارد. اگر نرم‌افزار ARP بتواند این آدرس را پیدا کند، بسته IP و آدرس فیزیکی پیدا شده را توأماً به لایه دیتالینک تحویل می‌دهد.

* جدول حافظه نهان

فرستنده معمولاً بیش از یک بسته IP برای ارسال به مقصد مشابهی دارد. اگر بخواهیم برای هر یک از بسته‌های IP که به سمت مقصد مشابهی می‌روند، از ARP استفاده کنیم، کارآیی بالایی بدست نمی‌آید. راه حل، استفاده از جدول حافظه نهان است و هر جا که در این بخش، به جدول اشاره کردیم، منظورمان همین جدول نهان است. وقتی که کامپیوتر یا روتر آدرس فیزیکی متناظر با آدرس IP را دریافت می‌کند، آن را برای استفاده آتی، داخل جدول می‌نویسد. می‌توان به مدت چند دقیقه، از این آدرس فیزیکی برای کلیه بسته‌های IP که به سمت همان مقصد می‌روند، استفاده کرد. با این حال از آنجا که فضای موجود در جدول محدود است، این آدرس فیزیکی برای مدت طولانی داخل جدول نگه داشته نمی‌شود.

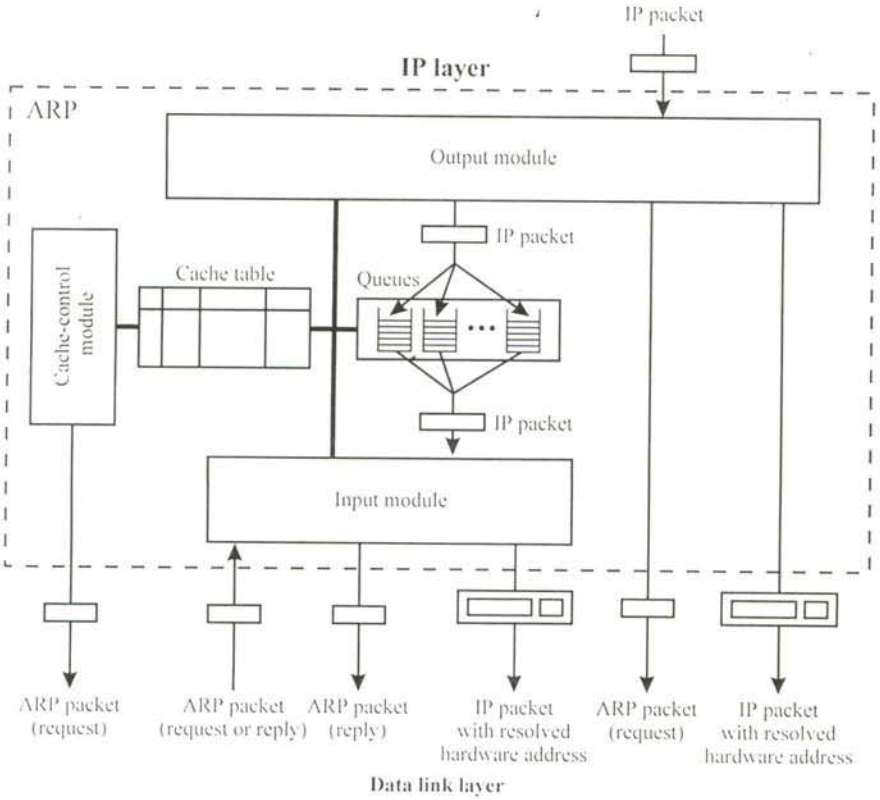
جدول به صورت آرایه‌ای از المانها، پیاده‌سازی می‌شود. در نرم‌افزار ما، هر المان حاوی فیلدهای زیر است :

State : حالت. این ستون، حالت یا وضعیت المان موجود در جدول را نشان می‌دهد و می‌تواند یکی از این سه مقدار را داشته باشد : ***آزاد***[6]، ***معلق***[7] یا ***قطعی***[8]. حالت F نشان می‌دهد که طول عمر این المان جدول، به سر رسیده است و می‌توان از جای آن برای نگهداری

1- Cache table
2- Queues
3- Output module
4- Input module
5- Cache control module
6- FREE : F
7- PENDING : P
8- RESOLVED : R

www.Abiri.ir

المان دیگری در جدول استفاده کرد. حالت P نشان می‌دهد که یک درخواست ARP برای این المان ارسال شده است، اما هنوز پاسخ ARP دریافت نشده است. حالت R نشان می‌دهد که آدرس فیزیکی مورد سؤال، به دست آمده است و لذا اطلاعات این المان جدول، کامل و قطعی است. بسته‌ای که در صف انتظار نشسته است تا به سمت مقصد ارسال گردد، می‌تواند از اطلاعات این المان جدول استفاده کند.



شکل ۷-۹: اجزاء ARP

Hardware type : نوع سخت‌افزار. این فیلد، مشابه با فیلد متناظر با آن در بسته ARP است.

Protocol type : نوع پروتوکل. این فیلد مشابه با فیلد متناظر با آن در بسته ARP است.

Hardware length : طول سخت‌افزار. این فیلد مشابه با فیلد متناظر با آن در بسته ARP است.

www.Abiri.ir

Protocol length : طول پروتوکل. این فیلد مشابه با فیلد متناظر با آن در بسته ARP است.

Interface number : شماره اینترفیس. یک روتر (یا یک کامپیوتر چند آدرسه) را می‌توان از طریق اینترفیس‌های مختلف با شماره‌های مختلف، به شبکه‌های مختلف وصل کرد. نوع سخت‌افزار و پروتوکل لایه شبکه در هر یک از این شبکه‌ها می‌تواند متفاوت باشد.

Queue number : شماره صف. ARP از صفهای شماره‌دار برای نگهداری موقت بسته‌هایی استفاده می‌کند که منتظر مشخص شدن آدرس فیزیکی گیرنده مورد نظرشان هستند. بسته‌هایی که به سمت گیرنده مشابهی ارسال شده‌اند، در صف مشابهی قرار می‌گیرند.

Attempts : تلاشها. این فیلد، تعداد دفعاتی که درخواست ARP برای این المان جدول ارسال شده است را نشان می‌دهد.

Time-out : مهلت مقرر. این ستون، طول عمر یک المان جدول را برحسب ثانیه نشان می‌دهد.

Hardware address : آدرس سخت‌افزار. این فیلد، آدرس فیزیکی گیرنده را نشان می‌دهد و تا زمانی که توسط یک پاسخ ARP پر نشده است، خالی می‌ماند.

Protocol address : آدرس پروتوکل. این فیلد، آدرس IP گیرنده مورد نظر را نشان می‌دهد.

❊ صفها

نرم‌افزار ARP حاوی یک سری صفها است، یکی برای هر مقصد، تا بسته‌های IP قبل از معلوم شدن آدرس فیزیکی گیرنده مورد نظر، در آن صفها بنشینند. ماجول خروجی، بسته‌هایی که هنوز آدرس فیزیکی آنها معلوم نیست را داخل این صفها می‌گذارد. ماجول ورودی پس از مشخص شدن آدرس فیزیکی، بسته‌ای را از صف برمی‌دارد و آن را به همراه آدرس فیزیکی مشخص شده به لایه دیتالینک می‌دهد.

❊ ماجول خروجی

ماجول خروجی منتظر دریافت یک بسته IP از نرم‌افزار IP می‌نشیند. ماجول خروجی داخل جدول نهان به دنبال المانی متناظر با آدرس IP مقصد این بسته می‌گردد. آدرس IP مقصد بسته، باید با مقدار نوشته شده در فیلد «آدرس پروتوکل» آن المان جدول، مطابقت داشته باشد.

www.Abiri.ir

اگر چنین المانی پیدا شد و حالت آن برابر با R بود، آدرس فیزیکی نوشته شده در فیلد «آدرس فیزیکی» آن المان به همراه بسته IP دریافتی به لایه دیتالینک داده می‌شوند تا به سمت مقصد، روانه شوند.

اگر چنین المانی پیدا شد و حالت آن برابر با P بود، بسته IP تا معلوم شدن آدرس فیزیکی گیرنده، صبر می‌کند. از آنجا که وضعیت، P است، هم اکنون صفی برای این مقصد ایجاد شده است. بنابراین ماجول خروجی می‌تواند بسته را تا معلوم شدن آدرس فیزیکی، داخل این صف بگذارد.

اگر چنین المانی پیدا نشد، ماجول یک صف را تولید می‌کند و بسته را داخل آن صف می‌گذارد. المانی با حالت P برای این مقصد، در جدول ایجاد می‌کند و فیلد «تلاشها» را برابر با 1 قرار می‌دهد. سپس یک درخواست ARP به صورت همگانی ارسال می‌کند.

❊❊ جریان رویدادهای ماجول خروجی

برای کسب اطلاعات بیشتر درباره نحوه تنظیم جریان رویدادهای یک برنامه نرم‌افزاری، می‌توانید به کتاب «مدل‌سازی با UML و پیاده‌سازی با JAVA» از همین مترجم، مراجعه فرمایید. جریان رویدادهای ماجول خروجی به شرح زیر است:

۱- صبر می‌کند تا یک بسته IP را از نرم‌افزار IP دریافت کند.
۲- داخل جدول به دنبال المانی می‌گردد که متناظر با مقصد این بسته IP باشد.
۳- اگر پیدا شد :
  ۱- اگر حالت آن المان برابر با R باشد :
    ۱- مقدار آدرس فیزیکی را از المان استخراج می‌کند.
    ۲- بسته IP و آدرس فیزیکی را توأماً به لایه دیتالینک می‌دهد.
    ۳- برمی‌گردد.
  ۲- اگر حالت آن المان برابر با P باشد :
    ۱- بسته را در صف متناظر با آن المان قرار می‌دهد.
    ۲- برمی‌گردد.
  ۴- اگر پیدا نشد :
۱- یک المان در جدول ایجاد می‌کند، حالت آن را برابر با P قرار می‌دهد و در فیلد «تلاشها» ، عدد 1 را می‌نویسد.
  ۲- یک صف ایجاد می‌کند.
  ۳- بسته را در آن صف می‌گذارد.
  ۴- یک درخواست ARP ارسال می‌کند.
  ۵- برمی‌گردد.

www.Abiri.ir

* ماجول ورودی

ماجول ورودی صبر می‌کند تا یک بسته ARP (درخواست یا پاسخ) از راه برسد. ماجول ورودی، داخل جدول به دنبال المانی متناظر با این بسته ARP می‌گردد. باید مقدار فیلد «آدرس پروتوکل» در آن المان، با مقدار نوشته شده در فیلد «آدرس پروتوکل هدف» در بسته ARP دریافتی، مطابقت داشته باشد.

اگر این المان پیدا شد و حالت آن برابر با P بود، ماجول با کپی کردن آدرس فیزیکی نوشته شده در فیلد «آدرس فیزیکی هدف» بسته ARP دریافتی، به داخل فیلد «آدرس فیزیکی» المان و نیز نوشتن وضعیت R در فیلد حالت آن المان، محتوای المان جدول را به روزرسانی می‌کند. همچنین مقدار فیلد «مهلت مقرر» این المان نیز روی یک مقدار اولیه ست می‌شود. سپس بسته‌های IP موجود در صف انتظار را یکی یکی از صف خارج می‌کند و آنها را به همراه آدرس فیزیکی به دست آمده به لایه دیتالینک می‌دهد.

اگر المان پیدا شد و حالت آن R بود، ماجول همچنان المان را به روزرسانی می‌کند. این مورد ممکن است به دلیل تغییر آدرس فیزیکی گیرنده، پیش آمده باشد. فیلد «مهلت مقرر» نیز مجدداً مقدار اولیه‌ای را به خود می‌گیرد.

اگر المان پیدا نشد، ماجول یک المان جدید ایجاد می‌کند و آن را در جدول قرار می‌دهد. روال کار این پروتوکل آن است که هرگونه اطلاعات دریافتی، برای استفاده آتی داخل جدول نوشته شود. حالت المان به R تبدیل می‌شود و مقدار فیلد «مهلت مقرر» نیز روی یک مقدار اولیه ست می‌شود.

حال ماجول چک می‌کند ببیند آیا بسته ARP دریافتی، یک درخواست است. اگر چنین باشد، بلافاصله ماجول، یک پیغام پاسخ ARP تولید می‌کند و آن را برای فرستنده ارسال می‌کند.

در پاسخ ARP ، مقدار فیلد «نوع پیغام» از 1 (برای درخواست) به 2 (برای پاسخ) تغییر می‌کند و آدرس فیزیکی، داخل فیلد «آدرس فیزیکی هدف» نوشته می‌شود.

** جریان رویدادهای ماجول ورودی

۱- صبر می‌کند تا یک بسته ARP (درخواست یا پاسخ) از راه برسد.
۲- داخل جدول به دنبال المان متناظر با این بسته ARP می‌گردد.
۳- اگر پیدا شد :
  ۱- المان را به روزرسانی می‌کند.
  ۲- اگر حالت المان برابر با P باشد:
    ۱- تا زمانی که صف خالی نشده است:

www.Abiri.ir

۱- یک بسته از صف برمی‌دارد.
۲- بسته و آدرس فیزیکی را به لایه دیتالینک می‌دهد.
۴- اگر پیدا نشد :
۱- یک المان جدید ایجاد می‌کند.
۲- المان را به جدول اضافه می‌کند.
۵- اگر بسته، یک درخواست است:
۱- یک پاسخ ARP ارسال می‌کند.
۶- برمی‌گردد.

* ماجول کنترل‌کننده حافظه نهان

ماجول کنترل‌کننده حافظه نهان، مسئول نگهداری جدول حافظه نهان است. این ماجول به صورت تناوبی (مثلاً هر ۵ ثانیه)، المان به المان محتوای جدول را چک می‌کند.

اگر حالت المان، F باشد، سراغ المان بعدی می‌رود. اگر حالت، P باشد، مقدار فیلد «تلاشها» را یک واحد افزایش می‌دهد. سپس مقدار جدید آن را چک می‌کند. اگر مقدار جدید آن بیشتر از ماکزیمم تلاشهای مجاز باشد، حالت آن المان به F تبدیل می‌شود و صف متناظر با آن المان، پاک می‌شود ولی اگر تعداد تلاشها، کمتر از ماکزیمم تلاش مجاز باشد، ماجول، یک درخواست جدید ARP تولید و ارسال می‌کند.

اگر حالت المان برابر با R باشد، ماجول از مقدار فیلد «مهلت مقرر» به اندازه زمان سپری شده از آخرین به روزرسانی جدول، کم می‌کند. اگر مقدار جدید آن برابر با صفر یا کمتر از صفر شد، حالت این المان به F تبدیل می‌شود و صف مرتبط با آن پاک می‌شود.

** جریان رویدادهای ماجول کنترل‌کننده حافظه نهان

۱- صبر می‌کند تا تایمر چک کردن تناوبی جدول، زمان چک کردن را نشان دهد.
۲- برای هر المانی در جدول :
۱- اگر حالت آن المان برابر با F باشد:
۱- ادامه می‌دهد.
۲- اگر حالت آن المان برابر با P باشد:
۱- مقدار فیلد «تلاشها» را، یک واحد افزایش می‌دهد.
۲- اگر مقدار فیلد «تلاشها» بیش از ماکزیمم مقدار مجاز شد:
۱- حالت آن المان را برابر با F قرار می‌دهد.
۲- صف متناظر با آن المان را پاک می‌کند.

www.Abiri.ir

۳- در غیر این صورت :
۱- یک درخواست ARP ارسال می‌کند.
۴- ادامه می‌دهد.
۳- اگر حالت آن المان برابر با R باشد:
۱- مقدار فیلد «مهلت مقرر» را به اندازه زمان سپری شده از آخرین چک کردن جدول، کاهش می‌دهد.
۲- اگر مقدار جدید فیلد «مهلت مقرر» کمتر یا مساوی با صفر باشد:
۱- حالت آن المان را برابر با F قرار می‌دهد.
۲- صف متناظر با آن المان را از بین می‌برد.
۳- برمی‌گردد.

مثالهای بیشتر

در این بخش، مثالهای بیشتری درباره نحوه عملکرد پروتوکل ARP و نحوه تغییر محتوای جدول حافظه نهان، ارائه می‌دهیم. جدول ۷-۱ محتوای اولیه بعضی از فیلدهای این جدول، پیش از شروع مثالهایمان را نشان می‌دهد.

State	*Queue*	*Attempt*	*Time-Out*	*Protocol Addr.*	*Hardware addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
F					
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

جدول ۷-۱: محتوای جدول حافظه نهان که در مثالها به کار می‌رود

* مثال ۲

ماجول خروجی ARP از لایه شبکه، یک بسته IP به آدرس مقصد 114.5.7.89 دریافت می‌کند. او محتوای جدول را بررسی می‌کند و درمی‌یابد که المانی برای این مقصد با حالت R وجود دارد. او آدرس فیزیکی 457342ACAE32 را از جدول استخراج می‌کند و بسته IP و

www.Abiri.ir

آدرس فیزیکی استخراج شده را به لایه دیتالینک می‌دهد تا به سمت مقصد، ارسال گردد. محتوای جدول، تغییر نمی‌کند.

❊ مثال ۳

بیست ثانیه بعد، ماجول خروجی ARP از لایه شبکه، یک بسته IP به آدرس مقصد 116.1.7.22 را دریافت می‌کند. سپس محتوای جدول را بررسی می‌کند اما المانی متناظر با این آدرس در جدول، پیدا نمی‌کند. ماجول خروجی، المانی با حالت P و مقدار فیلد «تلاشها» برابر با 1 به جدول اضافه می‌کند. سپس صفی برای این مقصد می‌سازد و بسته IP را در آن صف قرار می‌دهد. آنگاه یک درخواست ARP برای یافتن آدرس فیزیکی متناظر با این مقصد، به لایه دیتالینک می‌دهد. جدول ۷-۲ محتوای جدید جدول را نشان می‌دهد.

State	*Queue*	*Attempt*	*Time-Out*	*Protocol Addr.*	*Hardware addr.*
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	23	1		116.1.7.22	
R	9		60	19.1.7.82	4573E3242ACA
P	18	3		188.11.8.71	

جدول ۷-۲ : جدول حافظه نهان به‌روزرسانی شده برای مثال ۳

❊ مثال ۴

پانزده ثانیه بعد، ماجول ورودی ARP ، یک بسته ARP دریافت می‌کند که فیلد «آدرس پروتوکل هدف» آن برابر با 188.11.8.71 است. ماجول ورودی، جدول را چک می‌کند و این آدرس را می‌یابد. سپس مقدار حالت را به R تبدیل می‌کند و فیلد «مهلت مقرر» را برابر با 900 ثانیه قرار می‌دهد. سپس ماجول ورودی، آدرس فیزیکی را از فیلد «آدرس فیزیکی هدف» بسته ARP دریافتی، می‌خواند (E34573242ACA) و آن را به جدول اضافه می‌کند. آنگاه سراغ صف شماره 18 می‌رود و تمام بسته‌های موجود در این صف را یکی یکی به لایه دیتالینک می‌دهد. محتوای جدید جدول در جدول ۷-۳ نشان داده شده است.

www.Abiri.ir

State	Queue	Attempt	Time-Out	Protocol Addr.	Hardware addr.
R	5		900	180.3.6.1	ACAE32457342
P	2	2		129.34.4.8	
P	14	5		201.11.56.7	
R	8		450	114.5.7.89	457342ACAE32
P	12	1		220.55.5.7	
P	23	1		116.1.7.22	
R	9		60	19.1.7.82	4573E3242ACA
R	18		900	188.11.8.71	E34573242ACA

جدول ۷-۳: جدول حافظه نهان به روزرسانی شده برای مثال ۴

*** مثال ۵**

بیست و پنج ثانیه بعد، ماجول کنترل حافظه نهان، المانهای موجود در جدول را به روزرسانی می‌کند. مقدار فیلد «مهلت مقرر» برای اولین سه المان با حالت R ، به اندازه 60 ثانیه و برای آخرین المان با حالت R ، به اندازه 25 ثانیه کاهش داده می‌شود. حالت یک المان مانده به آخر جدول به F تبدیل می‌شود زیرا که مقدار جدید فیلد «مهلت مقرر» آن برابر با صفر شده است. برای هر یک از المانها با حالت P، مقدار فیلد «تلاشها» یکی افزایش می‌یابد. بعد از یک واحد افزایش، مقدار فیلد «تلاشها» برای یک المان (آنکه آدرس IP برابر با 201.11.56.7 دارد) بیش از حداکثر مقدار مجاز (5) می‌شود و لذا حالت آن المان به F تبدیل می‌شود. صف متناظر با آن المان، پاک می‌شود و یک پیغام ICMP خطا برای مبدأ اولیه ارسال می‌شود (فصل نهم را ببینید). جدول ۷-۴ محتوای جدید جدول را نشان می‌دهد.

RARP ۳-۷

RARP آدرس منطقی ماشینی را پیدا می‌کند که فقط آدرس فیزیکی خودش را می‌داند. به هر کامپیوتر یا روتر، یک یا چند آدرس منطقی IP داده شده است که منحصر به فرد و مستقل از آدرس فیزیکی ماشین، می‌باشند. برای تولید یک بسته IP، یک کامپیوتر یا روتر باید آدرس‌های IP خودش را بداند. آدرس IP یک کامپیوتر معمولاً از فایل پیکربندی آن که در هارددیسک ذخیره شده است، خوانده می‌شود.

با این حال، یک کامپیوتر فاقد هارددیسک، معمولاً از ROM بوت می‌شود و در ROM اطلاعات کمی برای بوت شدن کامپیوتر وجود دارد زیرا که ROM توسط سازنده کامپیوتر،

www.Abiri.ir

نصب می‌شود. ROM نمی‌تواند حاوی آدرس IP باشد زیرا که آدرسهای IP کامپیوتر متصل به یک شبکه، توسط مدیر شبکه تعیین می‌شوند.

State	*Queue*	*Attempt*	*Time-Out*	*Protocol Addr.*	*Hardware addr.*
R	5		840	180.3.6.1	ACAE32457342
P	2	3		129.34.4.8	
F					
R	8		390	114.5.7.89	457342ACAE32
P	12	2		220.55.5.7	
P	23	2		116.1.7.22	
F					
R	18		875	188.11.8.71	E34573242ACA

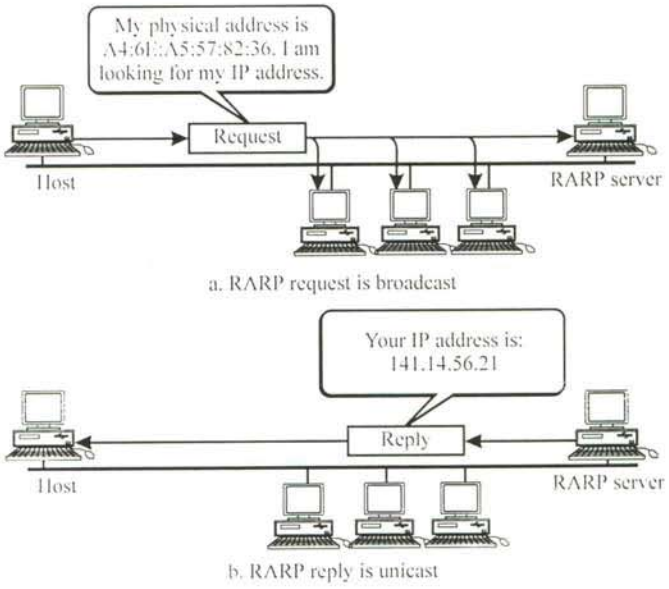
جدول ۷-۴: جدول حافظه نهان به روزرسانی شده برای مثال ۵

کامپیوتر می‌تواند آدرس فیزیکی خود را با خواندن آن از روی کارت شبکه‌اش به دست آورد. این آدرس، منحصر به فرد است (و البته منحصر به فرد بودن آن در سطح محلی کفایت می‌کند). سپس ماشین می‌تواند با استفاده از این آدرس فیزیکی و پروتوکل RARP، آدرس منطقی خود را به دست آورد. برای این کار، کامپیوتر یک درخواست RARP تولید می‌کند و آن را به صورت همگانی، روی شبکه محلی می‌فرستد. کامپیوتر دیگری روی همان شبکه که از همه آدرسهای IP باخبر است، یک پاسخ RARP ارسال می‌کند. یک برنامه RARP client باید روی کامپیوتر درخواست‌کننده آدرس IP در حال اجرا باشد و کامپیوتری که پاسخ می‌دهد باید یک RARP server در حال اجرا داشته باشد (شکل ۷-۱۰ را ببینید).

مهم: بسته‌های درخواست RARP به صورت همگانی ارسال می‌شوند. بسته‌های پاسخ RARP به صورت تک‌گیرنده ارسال می‌شوند.

در شکل ۷-۱۰ a، کامپیوتر فاقد هارد دیسک سمت چپ، بوت می‌شود. این کامپیوتر برای گرفتن آدرس IP خود، یک درخواست RARP را به صورت همگانی برای تمام کامپیوترهای متصل به شبکه ارسال می‌کند. همه کامپیوترها یا روترهای متصل به شبکه، این درخواست را دریافت می‌کنند اما مطابق شکل ۷-۱۰b ، فقط سرور RARP که روی ماشین سمت راست در حال اجراست، پاسخ RARP را ارسال می‌کند. بسته‌ای که سرور به عنوان پاسخ RARP ارسال می‌کند، حاوی آدرسِ IP ماشینِ متقاضیِ آدرسِ IP است.

www.Abiri.ir



شکل ۷-۱۰: نحوه عملکرد RARP

فرمت بسته RARP

فرمت بسته RARP ، دقیقاً مشابه با فرمت بسته ARP است، با این تفاوت که مقدار فیلد «نوع پیغام» آن برای درخواست RARP ، برابر با 3 و برای پاسخ RARP ، برابر با 4 است. شکل ۷-۱۱ را ببینید.

کپسول‌بندی

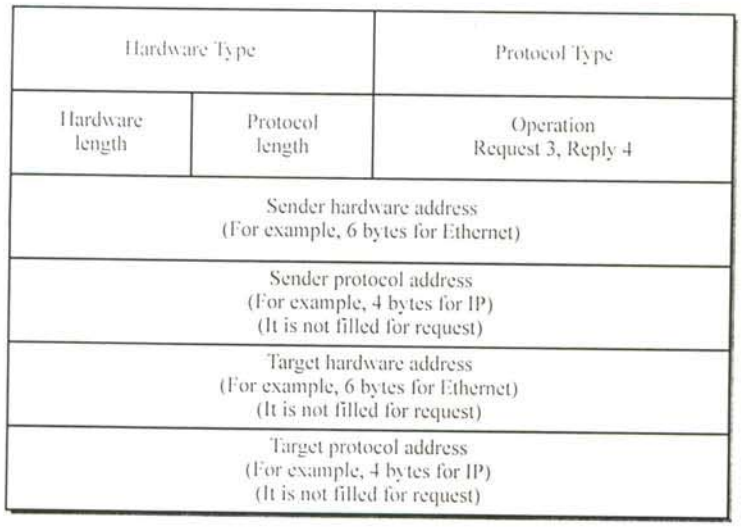
یک بسته RARP مستقیماً به داخل فریم لایه دیتالینک کپسول‌بندی می‌شود. به عنوان مثال، شکل ۷-۱۲ بسته RARP را نشان می‌دهد که داخل یک فریم اترنت، کپسول‌بندی شده است. دقت نمایید که محتوای فیلد type در این حالت، یعنی 0x8035 ، نشان می‌دهد که این فریم، حاوی یک بسته RARP است.

سرور RARP

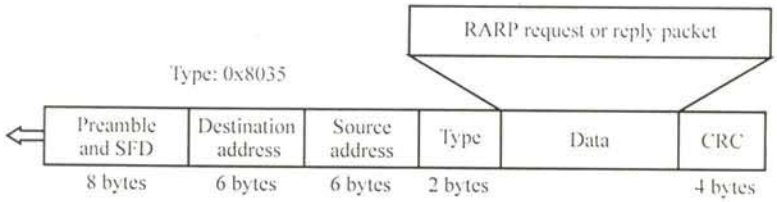
سرور RARP، نگاشت یک آدرس فیزیکی به یک آدرس IP را انجام می‌دهد. نگاشت مذکور داخل فایلی روی هارد دیسک ذخیره می‌شود. نکته جالب و تا حدودی پیچیده آن است که معمولاً سرور، در لایه کاربرد پیاده‌سازی می‌شود و فایلهایی که سرور از آنها استفاده می‌کند، در همان سطح، قابل دسترسی هستند. اما برای سرور RARP ، در لایه دیتالینک

www.Abiri.ir

پیاده‌سازی می‌شود. سرور RARP برای دسترسی به یک فایل، به کمک سیستم عامل ماشین، مثلاً UNIX ، نیاز دارد.



شکل ۷-۱۱ : بسته RARP



مثال ۷-۱۲: کپسول‌بندی بسته RARP

مسئله دیگر در رابطه با سرور RARP این است که گاهی اوقات سرور RARP می‌تواند ***غیرفعال***[1] باشد. برای آنکه پاسخهای RARP همچنان برای کامپیوترهای فاقد هارد دیسک ارسال شوند، معمولاً مدیرشبکه، بیش از یک سرور RARP را نصب می‌کند. اگر همه سرورهای RARP با هم در حال کار باشند، چندین پاسخ RARP به طور همزمان روی شبکه ارسال خواهند شد و ممکن است ترافیک بالایی را ایجاد کنند.

1- down

www.Abiri.ir

راه حلهای دیگری به جای RARP

وقتی یک کامپیوتر فاقد هارد دیسک بوت می‌شود، به اطلاعاتی بیش از آدرس IP خود نیاز دارد و این اطلاعات شامل مَسک زیرشبکه، آدرس IP روتر پیش‌فرض و آدرس IP سرور ***DNS***[1] می‌باشد. RARP نمی‌تواند این اطلاعات اضافه را ارائه دهد. در فصل شانزدهم درباره دو پروتوکل BOOTP و DHCP صحبت خواهیم کرد که می‌توان از آنها به جای RARP استفاده کرد.

۷-۴ عبارات کلیدی

پروتوکل یافتن آدرس (Address Resolution Protocol : ARP)
ماجول کنترل حافظه نهان (cache control module)
جدول حافظه نهان (cache table)
نگاشت دینامیک (dynamic mapping)
کپسول‌بندی (encapsulation)
ماجول ورودی (input module)
آدرس IP (IP address)
آدرس منطقی (logical address)
ماجول خروجی (output module)
آدرس فیزیکی (physical address)
Proxy ARP
صف (queue)
معکوس ARP (Reverse Address Resolution Protocol : RARP)
نگاشت استاتیک (Static mapping)

۷-۵ خلاصه

- تحویل یک بسته به یک کامپیوتر یا روتر، به دو سطح آدرس نیاز دارد : منطقی و فیزیکی.
- آدرس منطقی، کامپیوتر یا روتر را در سطح لایه شبکه مشخص می‌کند. TCP/IP نام این آدرس منطقی را آدرس IP می‌گذارد.
- یک آدرس فیزیکی، کامپیوتر یا روتر را در سطح لایه دیتالینک مشخص می‌کند.

1- Domain Name System : DNS

www.Abiri.ir

- نگاشت یک آدرس منطقی به یک آدرس فیزیکی می‌تواند استاتیک یا دینامیک باشد.
- نگاشت استاتیک حاوی لیستی از تناظر یک به یک آدرسهای IP و آدرسهای فیزیکی است. به روز نگه داشتن چنین لیستی، کار زیادی می‌طلبد.
- ARP یک متد نگاشت دینامیک است که با دادن یک آدرس IP به آن، آدرس فیزیکی متناظر با آن را به ما می‌دهد.
- درخواست ARP به صورت همگانی برای تمام وسایل متصل به شبکه ارسال می‌شود.
- پاسخ ARP به صورت تک‌گیرنده برای کامپیوتر ارسال‌کننده درخواست ARP ، فرستاده می‌شود.
- در Proxy ARP، یک روتر به صورت نماینده چندین ARP عمل می‌کند. وقتی که درخواست ARP به دنبال آدرس فیزیکی یکی از کامپیوترهای این مجموعه می‌گردد، روتر، آدرس فیزیکی خود را اعلام می‌کند. این کار، یک ویژگی مشابه با زیرشبکه‌سازی، به شبکه می‌دهد.
- نرم‌افزار ARP حاوی پنج جزء است: یک جدول حافظه نهان، صفها، یک ماجول خروجی، یک ماجول ورودی و یک ماجول کنترل‌کننده حافظه نهان.
- جدول حافظه نهان حاوی آرایه‌ای از المانهاست که توسط پیغامهای ARP ، تولید و به روزرسانی می‌شوند.
- یک صف حاوی بسته‌هایی است که به سمت مقصد مشابهی می‌روند.
- ماجول خروجی، بسته‌ای را از لایه IP می‌گیرد و آن را یا به لایه دیتالینک می‌دهد و یا در یک صف می‌گذارد.
- ماجول ورودی، از بسته ARP دریافتی برای به روزرسانی محتوای جدول حافظه نهان، استفاده می‌کند. ماجول ورودی همچنین می‌تواند یک پاسخ ARP ارسال کند.
- ماجول کنترل‌کننده حافظه نهان، با به روزرسانی هر فیلدی در جدول حافظه نهان، آن را به روز نگه می‌دارد.
- RARP ، یک نوع نگاشت دینامیک است که در آن، با داشتن یک آدرس فیزیکی، می‌توان آدرس IP متناظر با آن را به دست آورد.

۷-۶ مجموعه تمرین‌ها

تمرین‌ها

۱- آیا طول بسته ARP مقدار ثابتی است؟ توضیح دهید.

۲- آیا طول بسته RARP مقدار ثابتی است؟ توضیح دهید.

۳- اگر از اترنت و IP استفاده کنیم، طول بسته ARP چقدر خواهد بود؟

www.Abiri.ir

۴- اگر از اترنت و IP استفاده کنیم، طول بسته RARP چقدر خواهد بود؟

۵- طول یک فریم اترنت حامل یک بسته ARP چقدر است؟

۶- طول یک فریم اترنت حامل یک بسته RARP چقدر است؟

۷- آدرس همگانی در اترنت چیست؟

۸- روتری با آدرس IP برابر با 125.45.23.12 و آدرس اترنت برابر با 23:45:AB:4F:67:CD ، بسته‌ای را برای ارسال به کامپیوتری به آدرس IP برابر با 125.11.78.10 و آدرس اترنت برابر با AA:BB:A2:4F:67:CD ، دریافت کرده است. محتوای بسته درخواست ARPای که توسط روتر ارسال می‌شود را نشان دهید. فرض کنید که زیرشبکه‌سازی نشده است.

۹- در پاسخ به درخواست ARP ارسالی در مثال هشتم، یک پاسخ ARP ارسال می‌شود. محتوای فیلدهای آن را نشان دهید.

۱۰- بسته درخواست ARP مثال هشتم را داخل یک فریم لایه دیتالینک کپسول‌بندی کنید. همه فیلدهای فریم را پر کنید.

۱۱- بسته پاسخ ARP مثال نهم را داخل یک فریم لایه دیتالینک کپسول‌بندی کنید. همه فیلدهای فریم را پر کنید.

۱۲- روتری با آدرس IP برابر با 195.5.2.12 و آدرس اترنت برابر با AA:25:AB:1F:67:CD بسته‌ای برای مقصدی به آدرس IP برابر با 185.11.78.10 دریافت کرده است. وقتی که روتر، جدول مسیردهی خود را بررسی می‌کند، درمی‌یابد که این بسته باید به روتری با آدرس IP برابر با 195.5.2.6 و آدرس فیزیکی برابر با AD:34:5D:4F:67:CD تحویل داده شود. محتوای فیلدهای بسته درخواست ARPای که توسط روتر ارسال می‌شود را نشان دهید. فرض کنید که زیرشبکه‌سازی نداریم.

۱۳- محتوای فیلدهای بسته ARPای که در پاسخ بسته ارسالی مثال ۱۲ فرستاده می‌شود را نشان دهید.

۱۴- نتیجه مثال ۱۲ را در یک فریم لایه دیتالینک کپسول‌بندی کنید. همه فیلدها را پر کنید.

۱۵- نتیجه مثال ۱۳ را در یک فریم لایه دیتالینک کپسول‌بندی کنید. همه فیلدها را پر کنید.

۱۶- یک کامپیوتر فاقد هارد دیسک، بوت شده است و آدرس اترنت کارت شبکه آن برابر با 98:45:23:4F:67:CD می‌باشد. محتوای فیلدهای بسته درخواست RARP ای که این کامپیوتر به هنگام بوت شدن، ارسال می‌کند را نشان دهید.

۱۷- محتوای فیلدهای بسته پاسخ RARP ارسالی، برای تمرین شانزدهم را نشان دهید. فرض کنید آدرس IP ماشین درخواست‌کننده برابر با 200.67.89.33 باشد. آدرسهای منطقی و فیزیکی مناسبی را برای سرور RARP انتخاب کنید. فرض کنید سرور RARP به همان شبکه کامپیوتر متقاضی آدرس IP ، متصل شده است.

www.Abiri.ir

۱۸- نتیجه تمرین شانزدهم را در یک فریم لایه دیتالینک، کپسول‌بندی کنید. همه فیلدها را پرکنید.
۱۹- نتیجه تمرین هفدهم را در یک فریم لایه دیتالینک، کپسول‌بندی کنید. همه فیلدها را پر کنید.

فعالیتهای تحقیقاتی

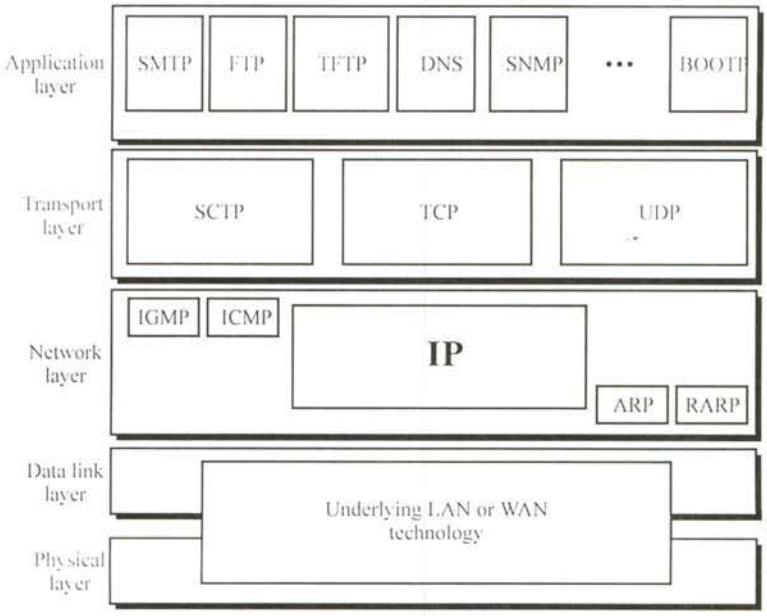
۲۰- RFCهایی که ARP را توصیف می‌کنند، کدامند؟
۲۱- RFCهایی که RARP را توصیف می‌کنند، کدامند؟
۲۲- نام فایلی که نگاشت استاتیک آدرس فیزیکی به آدرس منطقی را برای سرور RARP تحت سیستم عامل یونیکس، نگه می‌دارد، چیست؟

www.Abiri.ir

فصل هشتم

پروتوکل IP

پروتوکل IP[1] ، مکانیزم انتقال بسته‌ها در لایه شبکه است و پروتوکلهای پشته پروتوکل TCP/IP از آن استفاده می‌کنند. شکل ۸–۱، جایگاه پروتوکل IP در این پشته پروتوکل را نشان می‌دهد.



شکل ۸–۱ : جایگاه پروتوکل IP در پشته پروتوکل TCP/IP

IP یک پروتوکل غیر قابل اطمینان و فاقد اتصال است و در واقع یک ***سرویس تحویل بسته‌ها بر اساس نهایت تلاش***[2] می‌باشد. عبارت نهایت تلاش به این معنی است که IP ، هیچگونه خطایی را پیگیری نمی‌کند. IP فرض را بر غیر قابل اطمینان بودن لایه‌های پایین‌تر می‌گذارد و نهایت تلاش خود را می‌کند تا بسته‌ای را به مقصد برساند اما تضمینی نمی‌دهد. اگر

1- Internet Protocol
2- best effort delivery service

www.Abiri.ir

قابلیت اطمینان اهمیت داشته باشد، باید IP را با یک پروتوکل قابل اطمینان لایه بالاتر مانند TCP ، جفت کرد.

مثالی از یک سرویس متداول تحویل بر اساس نهایت تلاش، سرویس اداره پست است. اداره پست نهایت تلاش خود را برای تحویل نامه‌ها می‌کند اما همیشه موفق نمی‌شود. اگر یک نامه سفارشی نشده گم شود، این وظیفه فرستنده و یا گیرنده است که گم شدن نامه را پیگیری نماید. اداره پست، هیچگونه نامه سفارشی نشده‌ای را ردگیری نمی‌کند و لذا نمی‌تواند به فرستنده یا گیرنده، خبر گم شدن یا مخدوش شدن نامه را بدهد. می‌توان IP را یک پروتوکل فاقد اتصال در شبکه سویچینگ بسته‌ها نیز در نظر گرفت (فصل ششم را ببینید) . معنی آن این است که IP با هر بسته‌ای به طور مستقل از سایر بسته‌ها رفتار می‌کند و لذا هر بسته‌ای می‌تواند از مسیر متفاوتی خود را به مقصد برساند . بنابراین بسته‌هایی که از یک مبدأ به سمت یک مقصد مشابه ارسال می‌شوند، ممکن است خارج از نوبت به مقصد برسند. همچنین برخی از آنها ممکن است در حین انتقال، گم یا مخدوش شوند. IP برای رسیدگی به اینگونه مشکلات، روی یک پروتوکل لایه بالاتر حساب می‌کند.

۸-۱ بسته IP

به بسته‌های لایه IP، ***دیتاگرام***[1] نیز گفته می‌شود. شکل ۸-۲ فرمت یک بسته IP را نشان می‌دهد. بسته IP ، یک بسته با طول متغیر است و از دو بخش تشکیل شده است : هدر و دیتا. هدر به طول 20 بایت تا 60 بایت است و حاوی اطلاعاتی است که برای مسیردهی و تحویل بسته به مقصد، ضروری هستند.

فرمت هدر بسته IP

در TCP/IP ، متداول است که هدر را به صورت بخشهای چهار بایتی نشان دهیم. تعریف مختصری از هر فیلد هدر بسته IP به دنبال می‌آید.

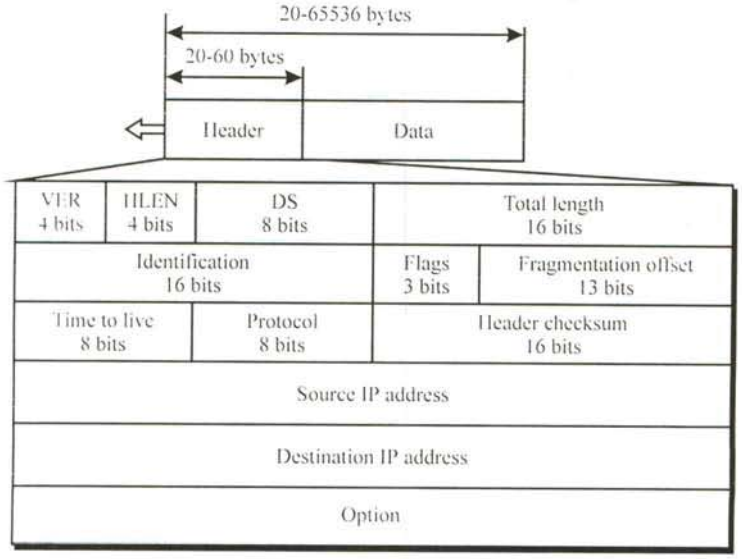
❊ VER (Version) : ویراست

این فیلد چهار بیتی، ویراست پروتوکل IP را نشان می‌دهد. ویراست فعلی، 4 است اما در آینده نزدیک، ویراست 6 (یا IPng) به طور کامل جای ویراست 4 را می‌گیرد. این فیلد، به نرم افزار IP که در ماشین در حال اجراست می‌گوید که فرمت هدر IP بر اساس ویراست 4 آن

1- datagram

www.Abiri.ir

می‌باشد. تفسیر همه فیلدها بر اساس آن چیزی است که در ویراست 4 این پروتوکل آمده است. ماشینی که از یک ویراست IP استفاده می‌کند، اگر بسته‌ای با ویراست متفاوتی را دریافت کند، بدون آنکه فیلدهای این ویراست IP را به اشتباه تفسیر کند، صرفاً آن را دور می‌اندازد.



شکل ۸-۲: فرمت بسته IP

❋HLEN (Header Length) : طول هدر

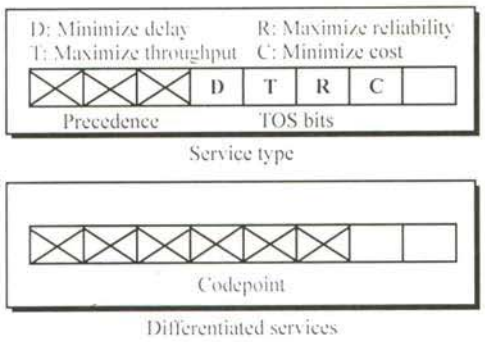
این فیلد 4 بیتی، کل طول هدر بسته IP را به صورت مضربی از 4 بایت نشان می‌دهد. از آنجا که طول هدر IP متغیر است (از 20 تا 60 بایت)، این فیلد مورد نیاز است. وقتی که هدر فاقد ***گزینه***[1] باشد، طول هدر برابر با 20 بایت است و لذا مقدار این فیلد برابر با 20/4=5 است. وقتی که حداکثر طول ممکن را برای گزینه‌ها به کار ببریم، طول هدر IP به 60 بایت می‌رسد و لذا مقدار فیلد فوق برابر با 60/4=15 می‌شود.

1- option

www.Abiri.ir

❊ تفسیر قدیمی و جدید بایت دوم هدر IP

IETF دو گونه تفسیر برای بایت دوم هدر IP ارائه کرده است که در شکل ۸-۳ ملاحظه می‌نمائید. تفسیر قدیمی آن به صورت ***نوع سرویس***[1] و تفسیر جدید آن به صورت ***سرویسهای ناهمسان***[2] می‌باشد.



شکل ۸-۳ :دو گونه تفسیر قدیمی و جدید بایت دوم هدر IP

❊❊ Service type : نوع سرویس

در این تفسیر، سه بیت اول، ***بیتهای ارجحیت***[3] نامیده می‌شوند. به چهار بیت بعدی، ***نوع سرویس***[4] گفته می‌شود و بیت آخر به کار نمی‌رود. زیرفیلدها به شرح زیر می‌باشند:

❊❊❊ زیرفیلد Precedence : ارجحیت

یک زیرفیلد 3 بیتی است و لذا مقدار آن از 0 (000) تا 7 (111) می‌باشد. ارجحیت در مسائلی مانند ***ازدحام***[5] نشان می‌دهد که نگه داشتن کدام بسته IP ، واجبتر است. اگر روتری دچار ازدحام شده باشد و مجبور به دور انداختن بعضی از بسته‌های IP باشد، اول آن بسته‌هایی را دور می‌اندازد که حداقل ارجحیت را دارند یعنی مقدار زیرفیلد ارجحیت آنها از بقیه کمتر است. بعضی از بسته‌های IP در اینترنت از بقیه بسته‌ها، مهم‌تر هستند. به عنوان مثال، بسته‌ای که برای مدیریت شبکه به کار می‌رود، مهم‌تر و اضطراری‌تر از بسته‌ای است که صرفاً حاوی یک سری اطلاعات جانبی درباره یک گروه است.

1- Service type
2- Differentiated services
3- Precedence bits
4- Type of Service : TOS
5- Congestion

www.Abiri.ir

مهم: زیرفیلد ارجحیت، بخشی از ویراست 4 پروتوکل IP بود اما هرگز به کار نرفت.

❊❊❊ زیرفیلد TOS : نوع سرویس

یک زیرفیلد 4 بیتی است و هر بیت، معنی خاصی دارد. گرچه یک بیت می‌تواند 0 یا 1 باشد، اما در هر بسته IP ، فقط و فقط یکی از این 4 بیت می‌تواند 1 باشد. جدول ۸-۱ الگوهای ممکن این 4 بیت و نحوه تفسیر آنها را نشان می‌دهد. از آنجا که در هر بسته IP فقط یکی از این چهار بیت می‌تواند 1 باشد، می‌توانیم پنج نوع سرویس مختلف داشته باشیم.

TOS Bits	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

جدول ۸-۱ : مقادیر مختلف TOS

پروتوکلهای لایه کاربرد می‌توانند سرویس مورد نظر خود را درخواست کنند. نوع سرویس پیش‌فرض برای برخی از پروتوکلهای لایه کاربرد در جدول ۸-۲ آمده است. از جدول ۸-۲ پیداست که فعالیتهای ***فعل و انفعالی***[1] که به توجه فوری نیاز دارند و فعالیتهایی که پاسخ سریع می‌طلبند، از سرویس ***حداقل تاخیر***[2] استفاده می‌کنند. فعالیتهایی که حجم انبوهی از دیتاها را ارسال می‌کنند، به ***حداکثر نرخ ارسال دیتا***[3] نیازمندند. فعالیتهای مدیریتی، به ***حداکثر قابلیت اطمینان***[4] نیاز دارند. فعالیتهای ***پس‌زمینه‌ای***[5] به ***حداقل هزینه مالی***[6] نیازمندند.

❊❊ DS (Differentiated Services) : سرویسهای ناهمسان

در تفسیر سرویسهای ناهمسان، اولین 6 بیت، زیر فیلد ***نقطه کدگذاری***[7] نامیده می‌شوند و از دو بیت آخر، استفاده‌ای نمی‌شود.

1- interactive
2- minimum delay
3- maximum throughput
4- maximum reliability
5- background
6- minimum cost
7- Codepoint : CP

www.Abiri.ir

Protocol	TOS Bits	Description
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP(data)	0100	Maximize throughput
FTP(control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP(command)	1000	Minimize delay
SMTP(data)	0100	Maximize throughput
DNS(UDP query)	1000	Minimize delay
DNS(TCP query)	0000	Normal
DNS(zone)	0100	Maximize throughput

جدول ۸-۲ : نوع سرویس پیش‌فرض برای برخی از پروتوکلهای لایه کاربرد

***** زیرفیلد Codepoint : نقطه کدگذاری**

می‌توان به دو شکل مختلف از زیرفیلد Codepoint که آن را به اختصار با CP نشان می‌دهیم، استفاده کرد:

الف. وقتی که 3 بیت سمت راست برابر با 000 باشند، سه بیت سمت چپ، مشابه بیتهای Precedence در تفسیر « نوع سرویس » به کار می‌روند. به عبارت دیگر، سازگاری با تفسیر قدیمی‌تر را فراهم می‌نماید.

ب. وقتی که 3 بیت سمت راست برابر با 000 نباشند، 6 بیت CP می‌توانند 64 سرویس را تعریف کنند و همانطور که جدول ۸-۳ نشان می‌دهد، مقامات مسئول اینترنت (Internet در جدول ۸-۳) و یا مقامات مسئول محلی (Local در جدول ۸-۳) ، این CP را به سرویسهای مختلفی اختصاص می‌دهند. گروه اول (xxxxx0) حاوی 32 سرویس است، گروه دوم (xxxx11) و گروه سوم (xxxx01) ، هر یک حاوی 16 سرویس می‌باشد. مقامهای مسئول اینترنت (که منظور همان IETF است)، مقدار CP گروه اول (اعداد 0 ، 2 ، 4 ، ... ، 62) را به سرویسهای مورد نظرشان اختصاص داده‌اند. مقدار CP گروه دوم (اعداد 3 ، 7 ، 11 ، 15 ، ... ، 63) توسط مقامات مسئول محلی (سازمانها) به سرویسهای مورد نظرشان اختصاص داده می‌شوند. مقدار CP گروه سوم (اعداد 1 ، 5 ، 9 ، ... ، 61) موقتی هستند و برای کارهای آزمایشی به کار می‌روند. دقت کنید که پیوستگی در اعداد هیچکدام از این گروهها، وجود ندارد. اگر قرار بود

www.Abiri.ir

اعداد فوق پیوسته باشند، می‌بایست از 0 تا 31 را به گروه اول، از 32 تا 47 را به گروه دوم و از 48 تا 63 را به گروه سوم اختصاص دهیم. اما اینگونه تخصیص اعداد به گروهها با تفسیر قدیمی « نوع سرویس » سازگاری ندارد زیرا در آن تفسیر، مقدار CP برابر با xxx000 می‌شود و اعداد 0 ، 8 ، 16 ، 24 ، 32 ، 40 ، 48 و 56 را در برمی‌گیرد و لذا در هر یک از سه گروه فوق‌الذکر، از این اعداد خواهیم داشت. اما اگر از اعداد گروهها مطابق با جدول ۸–۳ استفاده کنیم (اعداد ناپیوسته) در این صورت همه اعداد مربوط به تفسیر قدیمی « نوع سرویس » در گروه اول واقع خواهند شد. دقت نمایید که هنوز تخصیص اعداد جدول ۸-۳ ، قطعی نشده است.

Category	*Codepoint*	*Assigning Authority*
1	XXXXX0	Internet
2	XXXX11	Local
3	XXXX01	Temporary or experimental

جدول ۸-۳ : مقادیر ممکن Codepoint ها

* Total length : کل طول

این یک فیلد 16 بیتی است که کل طول (هدر و دیتا) بسته IP را بر حسب بایت نشان می‌دهد. برای پیدا کردن طول دیتایی که از لایه بالاتر می‌آید، باید طول هدر را از کل طول کم کنیم. می‌توان طول هدر را با ضرب کردن مقدار فیلد HLEN در عدد چهار بدست آورد.

طول هدر – کل طول = طول دیتا

از آنجا که این یک فیلد 16 بیتی است، کل طول بسته IP به 2^{16}-1=65535 بایت محدود شده است که از این طول، 20 تا 60 بایت می‌تواند هدر باشد و بقیه آن، دیتایی است که از لایه بالاتر آمده است.

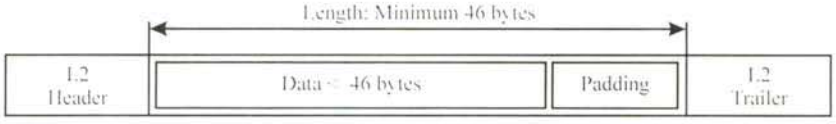
مهم: فیلد Total length ، کل طول بسته IP شامل هدر آن را نشان می‌دهد.

گرچه طول 65535 بایت ممکن است بزرگ به نظر برسد، اما در آینده نزدیک که تکنولوژیهای زیر بنایی اجازه نرخ بیت بالاتری را خواهند داد، ممکن است حداکثر طول بسته‌های IP نیز افزایش یابد.

www.Abiri.ir

وقتی که در بخش بعدی درباره ***قطعه قطعه کردن*** [1] بسته صحبت کنیم، خواهیم دید که بعضی از شبکه‌های فیزیکی نمی‌توانند یک بسته IP به طول 65535 بایت را در فریم خود کپسول‌بندی کنند و لذا باید بسته را قطعه قطعه کنند تا قابل انتقال در شبکه فیزیکی آنها باشد.

یک نفر ممکن است سئال کند که کلاً چرا ما به این فیلد نیاز داریم. وقتی که یک ماشین (کامپیوتر یا روتر) بسته‌ای را دریافت می‌کند، هدر و دنباله آن را حذف می‌کند و دیتای آن باقی می‌ماند. بنابراین چرا باید یک فیلد total length داشته باشیم که طول این دیتا را نشان دهد زیرا که با داشتن خود دیتا، طول آن نیز معلوم است. پاسخ این است که در بسیاری از موارد نیز واقعاً به این فیلد نیاز نداریم. با این حال، مواقعی هست که فقط بسته IP، داخل یک فریم کپسول‌بندی نمی‌شود و ممکن است صفرهایی نیز به دنبال دیتا اضافه شده باشند (که به آن Padding گفته می‌شود). به عنوان مثال پروتوکل اترنت، محدودیتی بر حداقل و حداکثر طول دیتایی که می‌توان در یک فریم اترنت کپسول‌بندی کرد (حداقل برابر با 46 بایت و حداکثر برابر با 1500 بایت) اعمال می‌کند. اگر طول یک بسته IP کمتر از 46 بایت باشد، صفرهایی به دنبال آن اضافه می‌شوند تا حداقل طول را به 46 بایت برسانند. در این حالت، وقتی که ماشین، بسته IP را از فریم خارج می‌کند، از فیلد total length استفاده می‌کند تا بداند چه طولی از آن، همان بسته IP واقعی است (طولی که total length نشان می‌دهد) و بقیه آن صفرهایی است که برای جبران حداقل طول به آن اضافه شده‌اند (شکل ۸-۴ را ببینید).



شکل ۸-۴ : کپسول‌بندی یک بسته IP کوچک در فریم اترنت

❋Identification : شناسه بسته

از این فیلد در قطعه قطعه سازی بسته استفاده می‌شود (در بخش بعدی درباره آن صحبت خواهیم کرد).

1- Fragmentation

www.Abiri.ir

* **Flags : پرچم‌ها**

از این فیلد در قطعه قطعه سازی بسته استفاده می‌شود (در بخش بعدی درباره آن صحبت خواهیم کرد).

* **Fragmentation Offset : مکان نسبی این قطعه در کل بسته**

از این فیلد در قطعه قطعه سازی بسته استفاده می‌شود (در بخش بعدی درباره آن صحبت خواهیم کرد).

* **TTL (Time To Live) : طول عمر**

یک بسته IP می‌تواند برای مدت محدودی در اینترنت حرکت کند . ابتدا هدف از طراحی این فیلد آن بود که حاوی یک مقدار زمانی باشد و هر روتری که بسته را دریافت می‌کرد، قدری از آن کم کند و وقتی که مقدار آن به صفر رسید، آن بسته را دور اندازد. با این وجود برای آنکه این طرح کار کند، کلاک همه ماشینها باید سنکرون باشند و باید بدانند که چه مدت زمانی طول می‌کشد تا بسته‌ای از یک ماشین به ماشین بعدی برود. امروزه از این فیلد اکثراً برای محدود کردن حداکثر تعداد هاپهایی که یک بسته می‌تواند طی کند (حداکثر از میان چند روتر عبور کند) استفاده می‌شود. وقتی که کامپیوتر مبدأ، بسته‌ای را ارسال می‌کند، عددی را در این فیلد می‌نویسد. مقدار این فیلد تقریباً دو برابر تعداد روترهایی است که بین دو کامپیوتر مبدأ و مقصد واقع شده‌اند. هر روتری که بسته را پردازش می‌کند، یک واحد از آن کم می‌کند. اگر بعد از کاهش یک واحد از این فیلد، مقدار آن صفر شد، روتر باید بسته IP را دور اندازد.

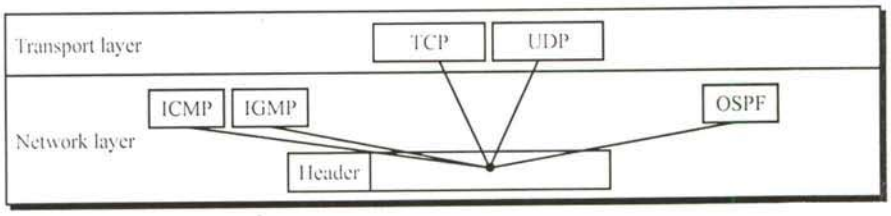
این فیلد مورد نیاز است زیرا که ممکن است جداول مسیردهی روترها خراب شوند و در چنین حالتی ممکن است یک بسته بدون آنکه به کامپیوتر مقصد برسد، برای مدت زمان زیادی بین دو و یا چند روتر جابه‌جا شود اما وجود این فیلد سبب می‌گردد طول عمر یک بسته IP محدود شود.

استفاده دیگر این فیلد آن است که عمداً بخواهیم طول عمر یک بسته در شبکه را محدود کنیم. به عنوان مثال اگر مبدأ بخواهد بسته را به شبکه محلی خود محدود کند، می‌تواند عدد 1 را در این فیلد بنویسد. وقتی که این بسته به دست اولین روتر برسد، مقدار این فیلد، یک واحد کاهش داده می‌شود و به دلیل صفر شدن آن، بسته دور انداخته می‌شود.

www.Abiri.ir

❊ Protocol : پروتوکل

این فیلد 8 بیتی، پروتوکل لایه بالاتر که از سرویس IP استفاده می‌کند را مشخص می‌نماید. یک بسته IP می‌تواند دیتای چندین پروتوکل لایه بالاتر مانند TCP ، UDP ، ICMP و IGMP را در خود کپسول‌بندی کند. این فیلد نشان می‌دهد که بسته IP در مقصد باید تحویل چه پروتوکل لایه بالاتری گردد. به عبارت دیگر، از آنجا که پروتوکل IP ، دیتای پروتوکلهای مختلف لایه بالاتر را کپسول‌بندی می‌کند، وقتی که بسته به مقصد می‌رسد، مقدار این فیلد در فرایند خارج کردن دیتا از کپسول IP و تحویل آن به پروتوکل مورد نظر لایه بالاتر کمک می‌کند (شکل ۸-۵ را ببینید).



شکل ۸-۵ : برخی از پروتوکلهایی که در شبکه توسط بسته IP حمل می‌شوند

مقدار این فیلد برای پروتوکلهای مختلف لایه بالاتر در جدول ۸-۴ نشان داده شده است.

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

جدول ۸-۴ : مقادیر مختلف فیلد Protocol برای پروتوکلهای مختلف

❊ Checksum : جمع آزما

یعنی با جمع‌زدن بیتها و یک سری عملیات دیگر، درمی‌یابیم خطای بیتی رخ داده است یا خیر. در بخشهای بعدی، نحوه محاسبه Checksum را بررسی خواهیم کرد.

www.Abiri.ir

❋ Source address : آدرس مبدأ

این فیلد 32 بیتی، آدرس IP کامپیوتر مبدأ را نشان می‌دهد که بسته را ارسال کرده است. در خلال زمانی که بسته IP از کامپیوتر مبدأ تا کامپیوتر مقصد حرکت می‌کند، مقدار این فیلد، ثابت می‌ماند.

❋ Destination address : آدرس مقصد

این فیلد 32 بیتی، آدرس IP کامپیوتر مقصدی را نشان می‌دهد که بسته به سمت آن می‌رود. در خلال زمانی که بسته IP از کامپیوتر مبدأ تا کامپیوتر مقصد حرکت می‌کند، مقدار این فیلد، ثابت می‌ماند.

❋ مثال ۱

یک بسته IP از راه رسیده است که هشت بیت اول آن به صورت زیر است:

← 01000010

گیرنده، بسته را دور می‌اندازد. چرا؟

راه حل: خطایی در این بسته وجود دارد. چهار بیت سمت چپ 0100، ویراست IP را نشان می‌دهد که صحیح است. چهار بیت بعدی 0010 ، طول هدر را به درستی نشان نمی‌دهند زیرا معادل اعشاری این عدد برابر با 2 است و لذا طول هدر برابر با 2x4=8 بایت می‌شود در حالی که حداقل طول هدر باید 20 بایت باشد و لذا این بسته در مسیر، خراب شده است.

❋ مثال ۲

در یک بسته IP، مقدار فیلد HLEN بر حسب باینری برابر با 1000 می‌باشد. این بسته، چند بایت option را حمل می‌کند؟

راه حل : مقدار HLEN بر حسب اعشاری برابر با 8 است و به این معنی است که طول هدر برابر با 8x4=32 بایت است. اولین 20 بایت، هدر اصلی است و 12 بایت بعدی، option است.

❋ مثال ۳

در یک بسته IP، مقدار HLEN برابر با 5_{16} و مقدار فیلد total length (کل طول) برابر با 0028_{16} می‌باشد. این بسته حاوی چند بایت دیتا است؟

www.Abiri.ir

راه حل : مقدار HLEN برابر با 5 است و معنی آن این است که طول هدر برابر با 5x4=20 بایت است (هیچ option ای وجود ندارد). کل طول بر حسب اعشاری برابر با 2x16+8=40 بایت است و لذا این بسته حاوی 40-20=20 بایت دیتا است.

* مثال ۴

یک بسته IP از راه رسیده است که ارقام اول آن برابر با مقادیر زیر است :

← 450000280001000000102...

قبل از آنکه بسته دور انداخته شود، چند هاپ را می‌تواند طی کند؟ دیتای این بسته متعلق به چه پروتوکل لایه بالاتری است؟

راه حل : برای پیدا کردن مقدار فیلد TTL ، 8 بایت اول (اولین 16 رقم هگزا) را کنار می‌گذاریم. فیلد TTL ، بایت نهم است که برابر با 01 می‌باشد و به این معنی است که این بسته حداکثر یک هاپ را می‌تواند طی کند. فیلد Protocol، بایت بعدی است که برابر با 02 می‌باشد و لذا دیتای این پروتوکل متعلق به پروتوکل IGMP است (جدول ۸-۴ را ببینید).

۸-۲ قطعه قطعه سازی بسته IP

یک بسته IP ممکن است از میان شبکه‌های مختلفی عبور کند. هر روتر از فریمی که دریافت می‌کند، بسته IP را استخراج می‌کند، آن را پردازش می‌کند و سپس آن را در فریم دیگری کپسول‌بندی می‌کند. فرمت و سایز فریم دریافتی بستگی به پروتوکل شبکه فیزیکی دارد که بسته از طریق آن دریافت شده است. فرمت و سایز فریم ارسالی بستگی به پروتوکل شبکه فیزیکی دارد که بسته از طریق آن ارسال خواهد شد. به عنوان مثال اگر یک روتر، یک LAN را به یک WAN متصل کند، فریمی را به فرمت LAN دریافت می‌کند و فریمی را به فرمت WAN ارسال می‌کند.

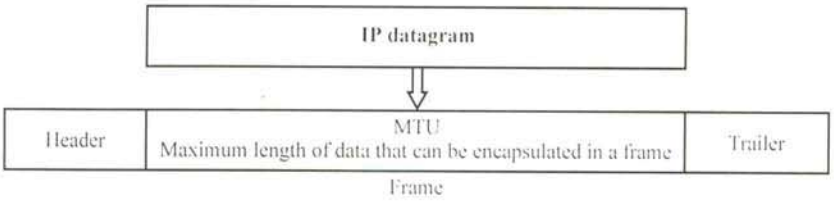
ماکزیمم واحد انتقال : MTU

هر پروتوکل لایه دیتالینک، فرمت مشخصی برای فریمهای خود دارد. یکی از مواردی که این فرمت تعیین می‌کند، حداکثر طول دیتایی است که فریم می‌تواند حمل کند و به آن ***ماکزیمم واحد انتقال***[1] گفته می‌شود. به عبارت دیگر وقتی که یک بسته IP داخل یک فریم، کپسول‌بندی می‌شود، کل طول بسته IP باید از مقدار MTU کمتر باشد و این محدودیت از

1- Maximum Transfer Unit : MTU

www.Abiri.ir

روی نوع سخت‌افزار و نرم‌افزاری که شبکه از آن استفاده می‌کند، اعمال می‌شود (شکل ۸-۶ را ببینید).



شکل ۸-۶ : MTU

از یک شبکه فیزیکی به شبکه فیزیکی دیگر، مقدار MTU فرق می‌کند. جدول ۸-۵ مقدار MTU را برای برخی از پروتوکلها نشان می‌دهد.

Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

جدول ۸-۵ : MTU برای برخی از پروتوکلها

به منظور مستقل ساختن پروتوکل IP از شبکه فیزیکی، طراحان تصمیم گرفتند ماکزیمم طول بسته IP را برابر با 65535 بایت قرار دهند. اگر از پروتوکلی استفاده کنیم که MTU آن برابر با این مقدار باشد، انتقال کارآمدتری خواهیم داشت. با این حال برای آنکه این بسته بتواند از میان سایر شبکه‌ها نیز عبور کند، باید بسته را به قطعات کوچکتری تقسیم کنیم. به این عمل، قطعه قطعه سازی بسته گفته می‌شود.

مبدأ غالباً بسته IP را قطعه قطعه نمی‌کند بلکه لایه انتقال چنان سگمنتهای دیتایش را انتخاب می‌کند که هم پروتوکل IP و هم پروتوکل لایه دیتالینک بتوانند آنها را حمل کنند. وقتی که یک بسته IP به قطعات کوچکتری تقسیم می‌شود، هر قطعه، هدر خودش را دارد و اکثر فیلدهای هدر بسته اولیه، در آن تکرار شده‌اند اما البته برخی از آنها نیز تغییر می‌کنند. اگر

www.Abiri.ir

یک قطعه به شبکه‌ای برسد که MTU آن کوچکتر از اندازه‌ای است که این قطعه بتواند در آن شبکه جابه‌جا شود، مجدداً به قطعات کوچکتری تقسیم می‌شود. به عبارت دیگر قبل از آنکه بسته‌ای به مقصد نهایی خود برسد، ممکن است چندین مرتبه قطعه قطعه شود.

کامپیوتر مبدأ یا هر روتری در مسیر ممکن است بسته را قطعه قطعه کند. مونتاژ مجدد قطعات فقط در کامپیوتر مقصد قابل انجام است زیرا که ممکن است هر قطعه از مسیر متفاوتی خود را به کامپیوتر مقصد برساند. از آنجا که ما کنترلی بر مسیری که قطعات طی می‌کنند تا به مقصد برسند، نداریم لذا فقط در کامپیوتر مقصد است که می‌توانیم آنها را مجدداً مونتاژ کنیم. دلیل دیگری که قطعات را قبل از رسیدن به مقصد نهائی مجدداً مونتاژ نمی‌کنیم این است که کارآیی پایین می‌آید.

وقتی که یک بسته IP به قطعات کوچکتری تقسیم می‌شود، بخشهای لازم هدر IP را باید در تمام قطعات، کپی کنیم. همانطور که در بخش بعد خواهیم دید، فیلد option ممکن است کپی شود و ممکن است کپی نشود. کامپیوتر یا روتری که بسته‌ای را قطعه قطعه می‌کند، باید مقدار سه فیلد را تغییر دهد : Flags ، Fragmentation offset و Total Length . بقیه فیلدها را باید کپی کرد. البته مقدار فیلد checksum را مجدداً باید محاسبه کرد.

فیلدهای مرتبط با قطعه قطعه سازی بسته IP

فیلدهایی که به قطعه قطعه سازی و مونتاژ مجدد قطعات یک بسته IP مربوط می‌شوند، Identification ، Flags و Fragmentation offset هستند:

❊ Identification : شناسه بسته

این فیلد 16 بیتی، که به اختصار آن را با ID نشان خواهیم داد، یک بسته IP را مشخص می‌کند که از کامپیوتر مبدأ راه افتاده است. وقتی که بسته‌ای، کامپیوتر مبدأ را ترک می‌کند، ترکیب فیلدهای ID و آدرس IP مبدأ، باید به طور منحصر به فردی آن بسته را مشخص کنند. برای تضمین منحصر به فرد بودن، پروتوکل IP از یک شمارنده برای دادن ID به بسته‌ها استفاده می‌کند. ابتدا شمارنده روی یک مقدار اولیه مثبت ست می‌شود. وقتی که پروتوکل IP یک بسته IP را ارسال می‌کند، مقدار فعلی شمارنده را در فیلد ID کپی می‌کند و مقدار شمارنده را یک واحد افزایش می‌دهد. تا زمانی که شمارنده را در حافظه اصلی نگه داریم، منحصر به فرد بودن، تضمین شده است. وقتی که یک بسته IP قطعه قطعه می‌شود، مقدار فیلد ID به داخل همه قطعات، کپی می‌شود. به عبارت دیگر، همه قطعات دارای شماره ID یکسانی هستند که برابر با مقدار آن در بسته IP اولیه است. شماره ID در مونتاژ بسته IP در مقصد

www.Abiri.ir

کمک می‌کند. او می‌داند همه قطعاتی که مقدار ID مشابهی دارند باید در مقصد به صورت یک بسته IP واحد، مونتاژ شوند.

* Flags : پرچمها

این یک فیلد سه بیتی است. اولین بیت، رزرو شده است. دومین بیت، بیت ***قطعه نکن***[1] نامیده شده است. اگر مقدار DF برابر با 1 باشد، ماشین نباید بسته را قطعه قطعه کند. اگر او نتواند بدون قطعه قطعه کردن بسته IP، آن را از طریق شبکه‌های فیزیکی موجود خود به سمت مقصد روانه کند، آن بسته IP را دور می‌اندازد و یک پیغام خطای ICMP برای کامپیوتر مبدأ ارسال می‌کند (فصل نهم را ببینید). اگر مقدار بیت DF برابر با 0 باشد، در صورت نیاز می‌توان بسته IP را قطعه قطعه کرد. بیت سوم، بیت ***قطعه بیشتر***[2] نامیده شده است. اگر مقدار بیت MF برابر با 1 باشد، به این معنی است که آن قطعه، آخرین قطعه نیست و پس از آن قطعات دیگری نیز از راه می‌رسند. اگر مقدار آن 0 باشد، به این معنی است که آن، آخرین قطعه و یا تنها قطعه است (شکل ۸-۷ را ببینید).

D: Do not fragment
M: More fragments



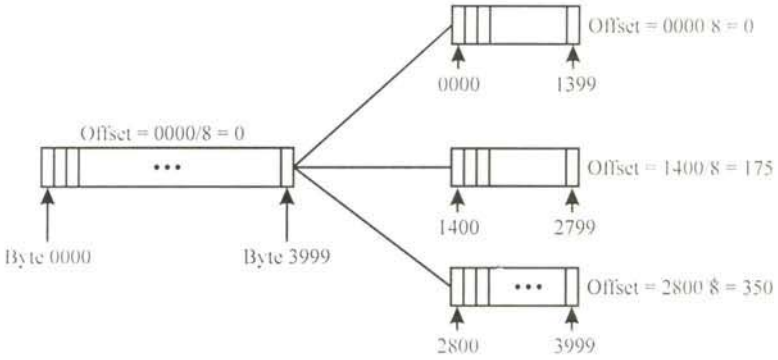
شکل ۸-۷ : فیلد Flags

* Fragmentation Offset : مکان نسبی این قطعه در کل بسته

این فیلد 13 بیتی، مکان نسبی این قطعه را در کل بسته نشان می‌دهد و در واقع offsetای است که صفر آن، در آغاز دیتای بسته اولیه واقع شده است و هر یک واحد افزایش آن، معرف هشت بایت پیشروی در دیتا است. شکل ۸-۸ یک بسته IP را نشان می‌دهد که طول دیتای آن 4000 بایت است و به سه قطعه تقسیم شده است. بایتهای بسته IP اولیه از 0 تا 3999 شماره‌گذاری شده‌اند. قطعه اول، بایتهای 0 تا 1399 را حمل می‌کند. مقدار offset برای این قطعه برابر با 0/8=0 است. قطعه دوم، بایتهای 1400 تا 2799 را حمل می‌کند. مقدار offset برای این قطعه برابر با 1400/8=175 است. سرانجام، قطعه سوم، بایتهای 2800 تا 3999 را حمل می‌کند. مقدار offset برای این قعطه برابر با 2800/8=350 است.

1- Don't Fragment : DF
2- More Fragment : MF

www.Abiri.ir



شکل ۸-۸ : مثالی از قطعه‌قطعه‌سازی بسته IP

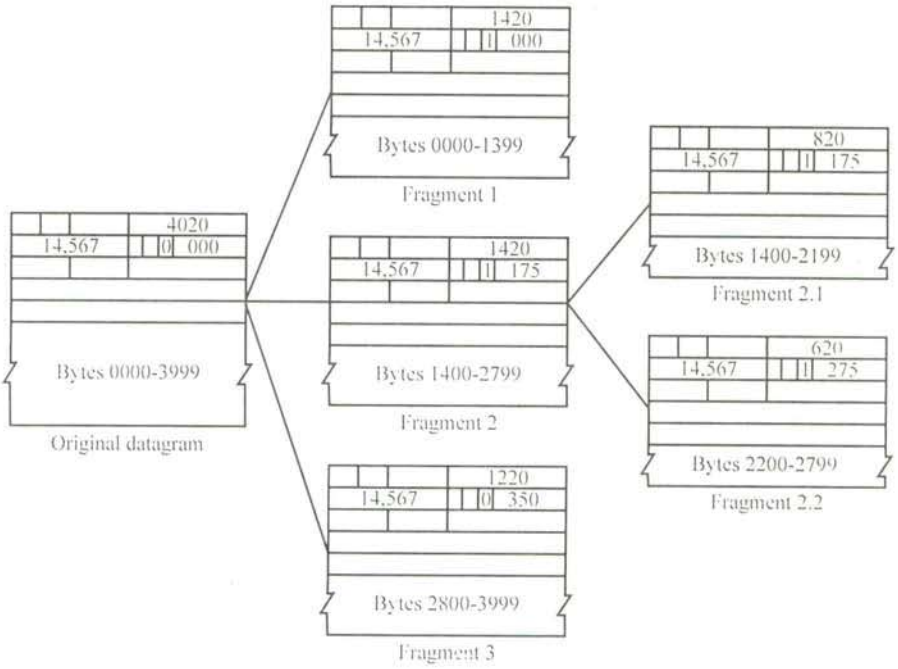
به خاطر داشته باشید که مقدار offset بر حسب واحدهای 8 بایتی اندازه‌گیری می‌شود. دلیل آن این است که فیلد offset ، 13 بیتی است و نمی‌تواند عددی بزرگتر از 2^{13}-1=8191 را نشان دهد. این محدودیت باعث می‌شود، کامپیوترها یا روترهایی که بسته IP را قطعه قطعه می‌کنند، طول هر قطعه را چنان انتخاب کنند که اولین شماره بایت دیتای هر قطعه (یعنی همان offset) بر عدد 8 قابل قسمت باشد. شکل ۸-۹ نمای بازشده‌تری از قطعات شکل قبل را نشان می‌دهد. دقت نمائید که مقدار فیلد ID در تمام قطعات، یکسان است. همچنین توجه نمایید که بیت MF در تمام قطعات به جز قطعه آخر، برابر با 1 است. شکل ۸-۹ مقدار فیلد offset برای هر قطعه را نیز نشان می‌دهد. علاوه بر آن شکل ۸-۹ نشان می‌دهد که خود یک قطعه، چگونه می‌تواند به قطعات کوچکتری تقسیم شود. در این حالت مقدار فیلد offset همواره نسبت به بسته IP اولیه سنجیده می‌شود. به عنوان مثال در شکل ۸-۹، قطعه دوم به دو قطعه کوچکتر، یکی حاوی 800 بایت و دیگری حاوی 600 بایت، تقسیم شده است. ولی مقدار offset ، همچنان مکان نسبی دیتای قطعات را نسبت به دیتای بسته IP اولیه نشان می‌دهد.

واضح است اگر هر قطعه از مسیر متفاوتی خود را به مقصد برساند و ترتیب آنها به هم بخورد، کامپیوتر نهائی می‌تواند بسته IP اولیه را از روی قطعات دریافتی (به شرط گم نشدن آنها) با دنبال کردن استراتژی زیر، بسازد :

الف ـ می‌دانیم مقدار فیلد offset اولین قطعه، برابر با 0 است.

ب ـ طول اولین قطعه را بر 8 تقسیم کنید. مقدار offset قطعه دوم، برابر با نتیجه حاصل است.

www.Abiri.ir



شکل ۸-۹ : مثالی از قطعه قطعه سازی

ج ـ مجموع طول قطعه اول و قطعه دوم را بر 8 تقسیم کنید. مقدار offset قطعه سوم، برابر با نتیجه حاصل است.

د - این روند را ادامه دهید. بیت MF آخرین قطعه برابر با صفر است.

*** مثال ۵**

بسته‌ای با بیت MF برابر با 0 از راه رسیده است. آیا این اولین قطعه، آخرین قطعه یا یک قطعه میانی است؟ آیا می‌توانیم بگوییم که بسته، قطعه قطعه شده است یا خیر؟

راه حل : اگر بیت MF برابر صفر باشد، به این معنی است که قطعات بیشتری وجود ندارند و این قطعه، آخرین قطعه است. با این حال، نمی‌توانیم بگوئیم که آیا بسته اولیه، قطعه قطعه شده است یا خیر زیرا که یک بسته قطعه قطعه نشده را نیز می‌توان آخرین قطعه در نظر گرفت.

*** مثال ۶**

بسته‌ای از راه رسیده است که بیت MF آن برابر با 1 است. آیا این اولین قطعه، آخرین قطعه یا یک قطعه میانی است؟ آیا می‌توانیم بگوییم که این بسته، قطعه قطعه شده است یا خیر؟

www.Abiri.ir

راه حل : اگر بیت MF برابر با 1 باشد، به این معنی است که حداقل یک قطعه دیگر وجود دارد. این قطعه می‌تواند اولین قطعه و یا یک قطعه میانی باشد اما نمی‌تواند آخرین قطعه باشد. نمی‌دانیم که آیا اولین قطعه و یا یک قطعه میانی است و لذا به اطلاعات بیشتری نیازمندیم (مقدار offset). مثال بعدی را نیز ببینید.

*** مثال ۷**

بسته‌ای از راه رسیده است که بیت MF آن برابر با 1 و offset آن برابر با 0 است. آیا این اولین قطعه، آخرین قطعه یا یک قطعه میانی است؟

راه حل : از آنجا که بیت MF برابر با 1 است، یا اولین قطعه و یا یک قطعه میانی است. از آنجا که مقدار offset برابر با صفر است، اولین قطعه است.

*** مثال ۸**

بسته‌ای از راه رسیده است که فیلد offset آن برابر با 100 است. شماره اولین بایت دیتای این بسته چقدر است؟ آیا شماره آخرین بایت را می‌توانیم بدست آوریم؟

راه حل : برای بدست آوردن شماره اولین بایت، مقدار offset را در عدد 8 ضرب می‌کنیم. لذا شماره اولین بایت برابر با 800 است. تا زمانی که طول دیتای بسته را ندانیم، نمی‌توانیم شماره آخرین بایت را بدست آوریم.

*** مثال ۹**

بسته‌ای از راه رسیده است که مقدار offset آن برابر با 100 ، مقدار HLEN برابر با 5 و مقدار فیلد total length برابر با 100 است. شماره اولین بایت و آخرین بایت، چند است؟

راه حل : شماره اولین بایت برابر با 100x8=800 است. کل طول، 100 بایت و طول هدر، 5x4=20 بایت است و به این معناست که این بسته حاوی 80=20-100 بایت است. اگر شماره اولین بایت، 800 باشد، شماره آخرین بایت باید 879=1-80+800 باشد.

۳-۸ options : گزینه‌ها

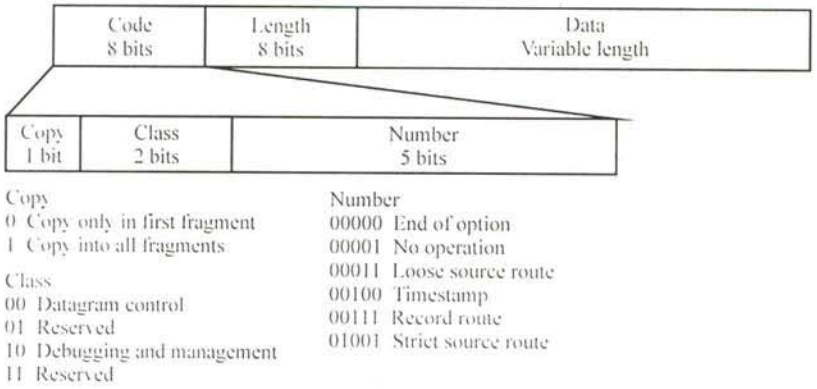
هدر بسته IP از دو بخش ساخته شده است: یک بخش ثابت و یک بخش متغیر. قبلاً درباره بخش ثابت هدر IP که به طول 20 بایت است، صحبت کردیم. بخش متغیر که گزینه‌ها (options) را در بر می‌گیرد، می‌تواند حداکثر به طول 40 بایت باشد.

www.Abiri.ir

گزینه‌ها همانگونه که نامشان نشان می‌دهد، حتماً نباید داخل هدر IP باشند و لذا وجودشان اختیاری است. می‌توان از آنها برای تست شبکه و خطازدائی استفاده کرد. گرچه وجود گزینه‌ها در هدر IP، الزامی نیست، نرم‌افزار IP باید قابلیت پردازش این گزینه‌ها را داشته باشد. بنابراین هرگونه پیاده‌سازی IP باید قابلیت پردازش گزینه‌هایی که ممکن است در هدر IP وجود داشته باشند را داشته باشد.

فرمت کلی یک option

شکل ۸-۱۰ فرمت یک گزینه را نشان می‌دهد. هر گزینه از یک فیلد یک بایتی ***کد***[1]، یک فیلد یک بایتی ***طول***[2] و یک فیلد ***دیتا***[3] با طول متغیر ساخته شده است.



شکل ۸-۱۰ : فرمت گزینه‌ها در هدر IP

❊ Code : کد

این فیلد به طول یک بایت است و از سه زیرفیلد ساخته شده است: ***کپی***[4]، ***کلاس***[5] و ***شماره***[6].

❊❊ Copy : کپی

این زیرفیلد یک بیتی نشان می‌دهد که آیا پس از قطعه قطعه سازی بسته IP، باید این گزینه را در هدر IP همه قطعات کپی کنیم یا فقط باید آن را در هدر IP اولین قطعه بنویسیم.

1- Code
2- Length
3- Data
4- Copy
5- Class
6- Number

www.Abiri.ir

وقتی که مقدار آن برابر با 0 است، به این معنی است که گزینه باید فقط در هدر IP اولین قطعه، نوشته شود و نیازی به کپی کردن آن در هدر IP سایر قطعات نیست. اگر مقدار آن برابر با 1 باشد، به این معنی است که باید این گزینه را در هدر IP همه قطعات، کپی کنیم.

Class : کلاس

این زیرفیلد 2 بیتی، هدف اصلی از به کارگیری این گزینه را نشان می‌دهد. وقتی که مقدار آن برابر با 00 باشد، به این معناست که این گزینه برای کارهای کنترلی است. اگر مقدار آن برابر با 10 باشد، یعنی از این گزینه برای ***خطازدائی و مدیریت***[1] استفاده می‌شود. دو مقدار ممکن دیگر این زیر فیلد 2 بیتی (01 و 11) هنوز تعریف نشده‌اند.

Number : شماره

این زیر فیلد 5 بیتی، شماره گزینه را تعریف می‌کند. گرچه 5 بیت فقط می‌تواند تا 32 شماره مختلف را تعریف کنند، در حال حاضر فقط از 6 شماره استفاده شده است. در بخش آتی در مورد آنها صحبت خواهیم کرد.

Length : طول

این فیلد، کل طول گزینه شامل فیلد Code و خود فیلد Length را بر حسب بایت نشان می‌دهد . همه گزینه‌ها، حاوی این فیلد نیستند.

Data : دیتا

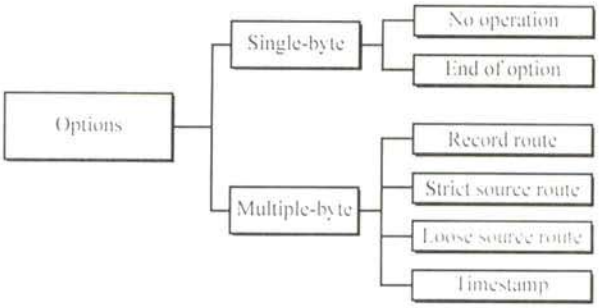
این فیلد حاوی دیتایی است که برخی از گزینه‌ها به آن نیاز دارند. مشابه فیلد Length، این فیلد نیز در همه گزینه‌ها وجود ندارد.

انواع optionها

همانطور که قبلاً اشاره کردیم، در حال حاضر فقط از شش گزینه استفاده می‌کنیم. دو تا از این گزینه‌ها، یک بایتی هستند و آنها نیازی به فیلد Length و Data ندارند. چهار تا از آنها، گزینه‌های چند بایتی هستند و لذا آنها به فیلدهای Data و Length نیاز دارند (شکل ۸– ۱۱ را ببینید).

1- Debugging and Management

www.Abiri.ir



شکل ۸-۱۱ : انواع گزینه‌ها

❋ NOP : ترازبندی میان گزینه‌ها

گزینه ***ترازبندی میان گزینه‌ها***[1] یک گزینه یک بایتی است و از آن برای پر کردن فاصله خالی مابین گزینه‌ها و رساندن طول کل هدر به مضربی از 4 استفاده می‌شود و لذا می‌توان برای ترازبندی هدر از آن استفاده کرد (شکل ۸-۱۲ را ببینید).

❋ EOP : ترازبندی انتهای گزینه‌ها

گزینه ***ترازبندی انتهای گزینه‌ها***[2] نیز یک گزینه یک بایتی است که در انتهای گزینه‌ها برای ***صفرگذاری***[3] از آن استفاده می‌شود. با این وجود فقط می‌توان از آن به عنوان آخرین گزینه، آن هم فقط به تعداد یک بایت، استفاده کرد. بعد از این گزینه، گیرنده به دنبال دیتای بسته IP می‌گردد. بنابراین اگر برای ترازبندی هدر IP و رساندن طول آن به مضربی از 4، بیش از یک بایت مورد نیاز باشد، ابتدا باید به تعداد مورد نیاز از NOP استفاده کرد و سپس در انتهای آنها، یک بایت EOP قرار داد (شکل ۸-۱۳ را ببینید).

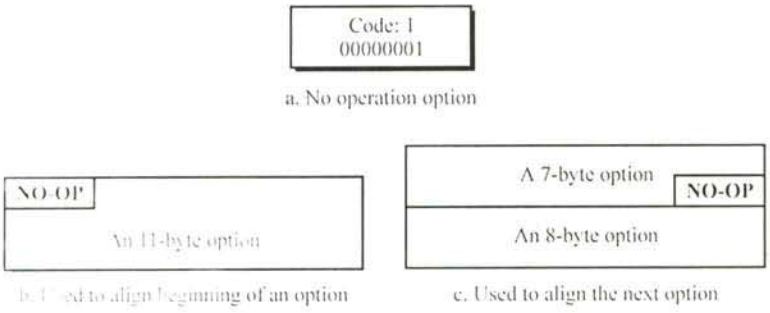
❋ Record Route : ضبط کردن مسیر

گزینه ***ضبط کردن مسیر***[4] برای ضبط آدرس روترهایی که بسته از آنها عبور کرده است (مسیری که بسته طی کرده است) استفاده می‌شود. از آنجا که حداکثر طول هدر، 60 بایت است و فقط 40 بایت فضای آزاد برای گزینه‌ها وجود دارد، می‌توان فقط 9 آدرس IP را

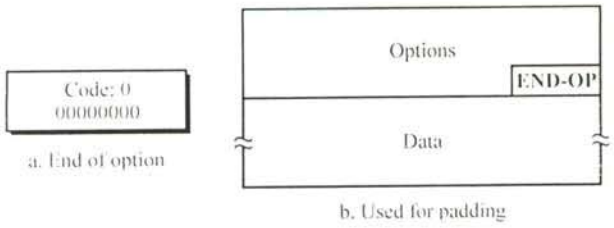
1- No operation : NOP
2- End of Options : EOP
3- Padding
4- Record Route

www.Abiri.ir

ضبط کرد (هر آدرس به 4 بایت نیاز دارد و لذا 9 آدرس به 4x9=36 بایت نیاز دارد. فیلدهای code ، length و Pointer ، هر کدام به یک بایت نیاز دارند که جمعاً 36+3=39 بایت می‌شود. در یک بایت باقیمانده، نمی‌توان آدرس IP دیگری نوشت). کامپیوتر مبدأ که بسته را تولید می‌کند، جای خالی لازم برای ضبط آدرسهای IP را در هدر بسته، فراهم می‌کند. شکل ۸–۱۴ فرمت این گزینه را نشان می‌دهد.



شکل ۸–۱۲ : گزینه NOP



شکل ۸–۱۳ : گزینه EOP

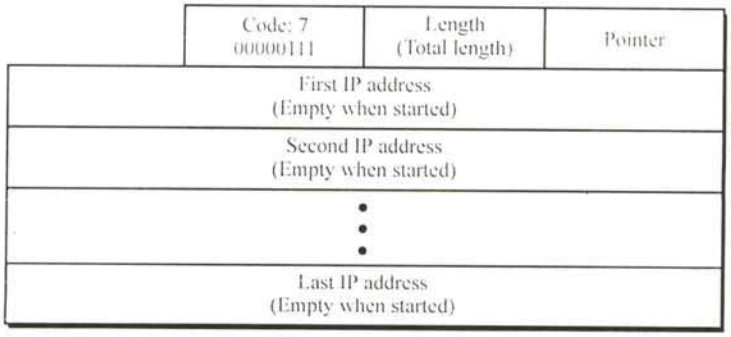
درباره فیلدهای code و length صحبت کردیم . فیلد ***نشانه‌گر***[1] یک عدد صحیح است که به صورت یک offset ، مکان اولین المان خالی برای ثبت آدرس IP در فضای خالی موجود در این گزینه را نشان می‌دهد. به عبارت دیگر، به اولین المان خالی موجود، اشاره می‌کند.

مبدأ تولید کننده بسته، فضای خالی برای ثبت آدرسهای IP را در قسمت دیتای این گزینه فراهم می‌کند. وقتی که بسته IP از مبدأ راه می‌افتد، همه جاهای خالی ثبت آدرسهای IP، در اختیار می‌باشند و اولین جای خالی را، مقدار اولیه Pointer که 4 است، نشان می‌دهد.

1- Pointer

www.Abiri.ir

وقتی که بسته در شبکه جابه‌جا می‌شود، هر روتری که بسته را پردازش می‌کند، مقدار pointer را با مقدار length مقایسه می‌کند. اگر مقدار pointer بزرگتر از مقدار length باشد، گزینه پر شده است و تغییر دیگری در آن داده نمی‌شود. اما اگر مقدار pointer بزرگتر از مقدار length نباشد، روتر، آدرس IP اینترفیسی که بسته باید از آن طریق، از روتر خارج شود را در جای خالی موجود بعدی می‌نویسد (دقت نمائید که یک روتر، بیش از یک آدرس IP دارد). سپس روتر مقدار Pointer را چهار واحد افزایش می‌دهد. شکل ۸-۱۵ نحوه پر شدن جاهای خالی موجود در گزینه Record Route ، وقتی که بسته از چپ به راست در شبکه حرکت می‌کند را نشان می‌دهد.



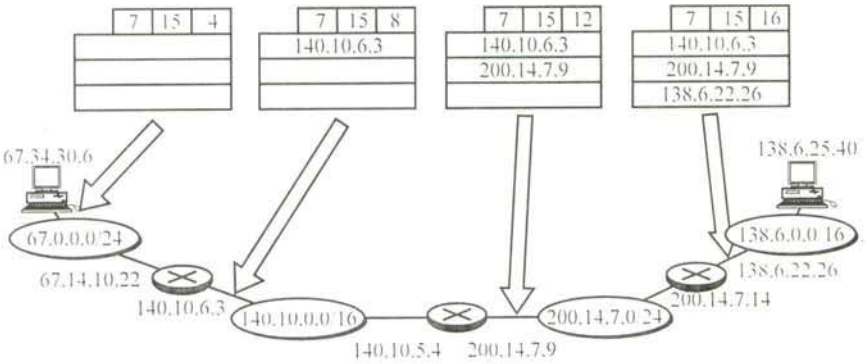
شکل ۸-۱۴ : گزینه Record Route

❊ Strict Source Route : مسیردهی قطعی از مبدأ

گزینه ***مسیردهی قطعی از مبدأ***[1] توسط مبدأ ارسال کننده بسته به کار می‌رود تا مسیری را به طور قطعی برای بسته‌ای که در شبکه حرکت می‌کند، تعیین نماید. تعیین مسیر توسط مبدأ، می‌تواند فوائدی داشته باشد. فرستنده می‌تواند مسیری را انتخاب کند که نوع مشخصی از سرویس، مانند ***حداقل تأخیر***[2] یا ***حداکثر نرخ انتقال***[3] را داشته باشد. یا اینکه ممکن است مبدأ، مسیری مطمئن‌تر را انتخاب کند. به عنوان مثال ممکن است روتر بخواهد مسیری را انتخاب کند که از میان شبکه رقیب عبور نکند.

1- Strict Source Route
2- Minimum Delay
3- Maximum Throughput

www.Abiri.ir



شکل ۸-۱۵ : مفهوم کزینه Record Route

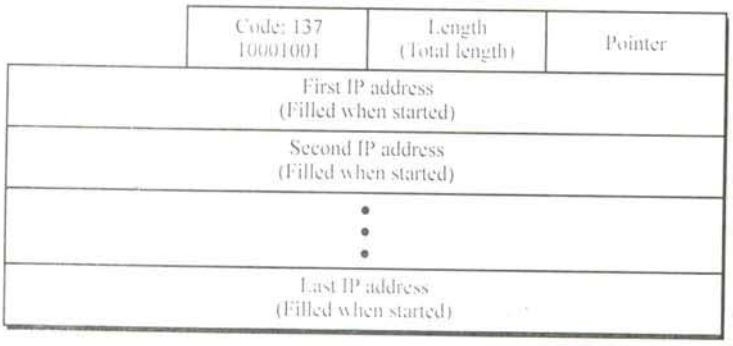
اگر مبدأ، مسیری را مشخص کند، بسته باید از تمام روترهایی که این گزینه مشخص کرده است، عبور نماید. اگر آدرس IP روتری در این گزینه، ذکر نشده باشد، بسته نباید از آن روتر عبور کند. اگر بسته از روتری عبور کند که آدرس آن در لیست آدرسهای موجود این گزینه نیامده باشد، آن بسته دور انداخته می‌شود و یک پیغام ICMP خطا برای مبدأ ارسال کننده بسته، فرستاده می‌شود. اگر بسته به مقصد برسد و از برخی از روترهایی که در لیست روترهای این گزینه آمده است، عبور نکرده باشد، باز هم بسته دور انداخته می‌شود و یک پیغام ICMP خطا برای فرستنده بسته، ارسال می‌شود.

با این حال، کاربران عادی اینترنت معمولاً از توپولوژی فیزیکی اینترنت مطلع نیستند. بنابراین اکثر کاربران از این گزینه استفاده نمی‌کنند. شکل ۸-۱۶ فرمت این گزینه را نشان می‌دهد.

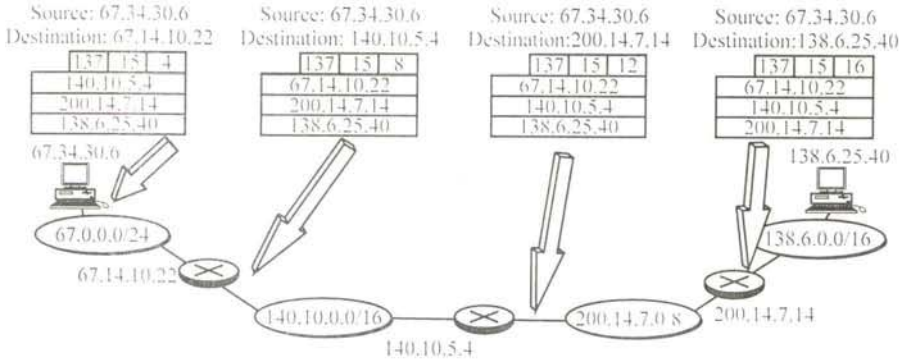
فرمت این گزینه شبیه به گزینه Record Route است با این تفاوت که فرستنده بسته، تمام آدرسهای IP را از ابتدا در این گزینه می‌نویسد. وقتی که بسته IP در شبکه حرکت می‌کند، هر روتری که بسته را پردازش می‌کند، مقدار pointer را با مقدار length مقایسه می‌کند. اگر مقدار pointer بزرگتر از مقدار length باشد، بسته از همه روترهای از قبل تعیین شده، عبور کرده است و هنوز به مقصد نرسیده است لذا بسته نمی‌تواند بیش از این حرکت کند و لذا دور انداخته می‌شود و یک پیغام ICMP خطا برای فرستنده بسته، ارسال می‌شود. اگر مقدار pointer بزرگتر از مقدار length نباشد، روتر، آدرس IP مقصد را با آدرس IP اینترفیسی که از طریق آن، بسته را دریافت کرده است، مقایسه می‌کند. اگر مساوی باشند، بسته را پردازش می‌کند. جای آدرس IP ای که pointer به آن اشاره می‌کند را با جای آدرس IP مقصد بسته عوض می‌کند، 4 واحد به مقدار pointer اضافه می‌کند و سپس بسته را به پیش

www.Abiri.ir

می‌راند. اگر مساوی نباشند، بسته را دور می‌اندازد و یک پیغام ICMP خطا برای مبدأ ارسال‌کننده بسته، می‌فرستد. شکل ۸–۱۷ نشان می‌دهد ضمن حرکت بسته از مبدأ تا مقصد، هر روتر چه عملی را انجام می‌دهد.



شکل ۸–۱۶: گزینه Strict Source Route

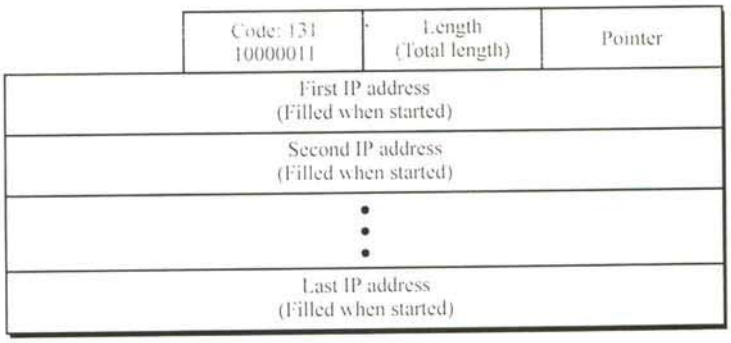


شکل ۸–۱۷ : نحوه عمل گزینه Strict Source Route

❋ Loose Source Route : مسیردهی نه چندان قطعی از مبدأ

گزینه مسیردهی نه چندان قطعی از مبدأ (Loose Source Route) شباهت زیادی به گزینه Strict Source Route دارد اما دارای انعطاف بیشتری است. بسته باید از روتری که آدرس IP آن در لیست آدرسهای IP این گزینه آمده است، عبور کند اما می‌تواند از روترهای دیگری نیز عبور نماید. شکل ۸–۱۸ فرمت گزینه Loose Source Route را نشان می‌دهد.

www.Abiri.ir



شکل ۸-۱۸ : گزینه Loose Source Route

* Timestamp : برچسب زمانی

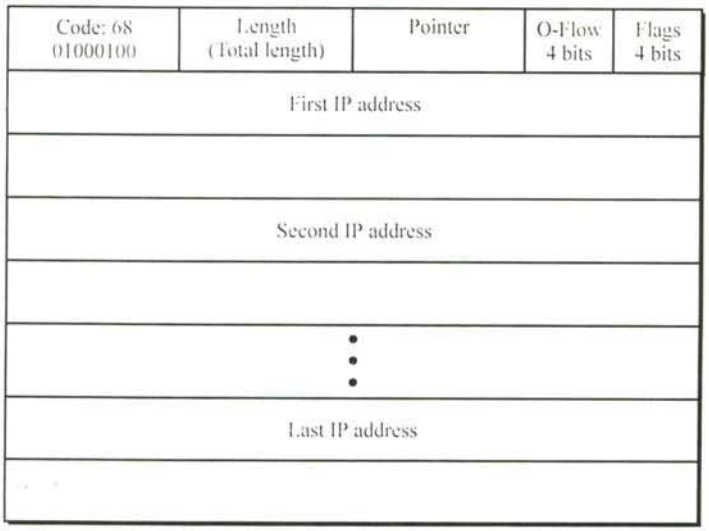
گزینه ***برچسب زمانی***[1] برای ثبت زمان پردازش بسته توسط روتر، به کار می‌رود. زمان بر حسب میلی ثانیه‌ای که از نیمه شب به ***وقت گرینویچ***[2] گذشته است، بیان می‌شود . دانستن زمان پردازش یک بسته می‌تواند به کاربران و مدیران در شناخت نحوه رفتار روترها در شبکه، کمک کند. می‌توانیم زمانی که طول می‌کشد تا بسته‌ای از یک روتر به روتر دیگر برود را تخمین بزنیم. از کلمه تخمین زدن استفاده کردیم زیرا ممکن است کلاک روترها، سنکرون نباشد. با این حال، کاربران عادی اینترنت غالباً از توپولوژی فیزیکی اینترنت مطلع نیستند. در نتیجه، اکثر کاربران از این گزینه استفاده نمی‌کنند. شکل ۸-۱۹، فرمت گزینه Timestamp را نشان می‌دهد.

در این شکل، تعریف فیلدهای Code و length مشابه سابق است. فیلد overflow (سر ریز) نشان دهنده تعداد روترهایی است که به دلیل نبودن جای خالی بیشتر، نتوانسته‌اند مقدار Timestamp خود را بنویسند. فیلد Flags نشان می‌دهد که هر روتری پس از دریافت بسته‌ای حاوی گزینه فوق‌الذکر، چه عملی را باید انجام دهد. اگر مقدار Flags برابر با 0 باشد، هر روتر، فقط مقدار Timestamp را در جای خالی مربوطه در گزینه، ثبت می‌کند. اگر Flags برابر با 1 باشد، هر روتر، هم آدرس IP خروجی که بسته باید از طریق آن به بیرون ارسال شود و هم مقدار Timestamp را ثبت می‌کند. اگر Flags برابر با 3 باشد، آدرسهای IP داده شده‌اند و هر روتر باید آدرس IP داده شده را با آدرس IP اینترفیسی که بسته از طریق آن دریافت شده است، مقایسه کند. اگر یکسان باشند، روتر، آدرس IP اینترفیسی که بسته باید از طریق آن از

1- Timestamp
2- Universal Time

www.Abiri.ir

روتر خارج شود را به جای آدرس IP داده شده، می‌نویسد و مقدار Timestamp را در جای خالی مربوطه در گزینه ثبت می‌کند (شکل ۸–۲۰ را ببینید).



شکل ۸–۱۹ : گزینه Timestamp

*** مثال ۱۰**

کدام یک از شش گزینه، باید در همه قطعات حاصل از یک بسته قطعه قطعه شده، کپی شوند؟

راه حل : برای هر یک از انواع گزینه‌ها، به بیت سمت چپ فیلد Code آنها نگاه می‌کنیم:

شکل ۸–۲۱ عملیاتی که هر روتر در هدایت بسته از مبدأ تا مقصد انجام می‌دهد را نشان می‌دهد. در این شکل فرض بر آن است که مقدار فیلد Flags برابر با 1 می‌باشد.

الف ـ Code : No operation برابر با 00000001 می‌باشد، کپی نمی‌شود.

ب ـ Code : End of option برابر با 00000000 می‌باشد، کپی نمی‌شود.

ج ـ Code : Record Route برابر با 00000111 می‌باشد، کپی نمی‌شود.

د ـ Code : Strict Source Route برابر با 10001001 می‌باشد، کپی می‌شود.

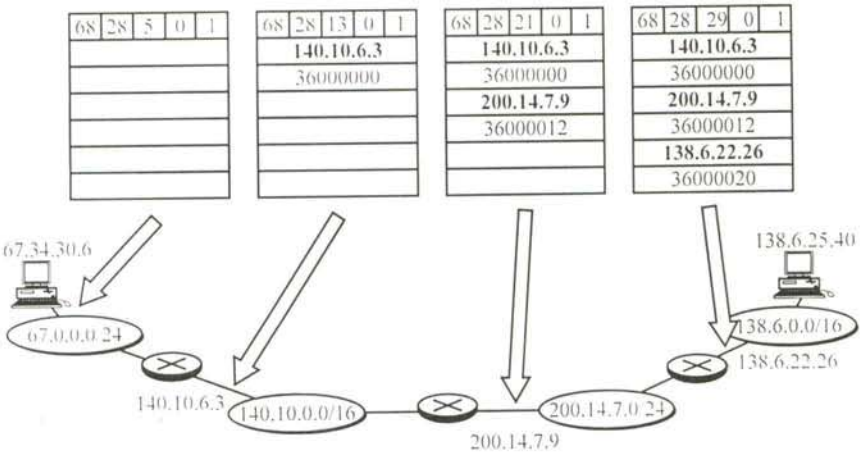
ه ـ Code : Loose Source Route برابر با 10000011 می‌باشد، کپی می‌شود.

ی ـ Code : Timestamp برابر با 01000100 می‌باشد، کپی نمی‌شود.

www.Abiri.ir



شکل ۸-۲۰ : استفاده از Flags در Timestamp



شکل ۸-۲۱ : نحوه عمل گزینه Timestamp

* مثال ۱۱

کدامیک از شش گزینه برای کنترل و کدامیک برای خطایابی و مدیریت به کار می‌روند.

راه حل : به بیت دوم و سوم (از سمت چپ) فیلد Code نگاه می‌کنیم.

الف – Code : No operation برابر با 00000001 می‌باشد : کنترل

ب – Code : End of option برابر با 00000000 می‌باشد : کنترل

ج- Code : Record Route برابر با 00000111 می‌باشد : کنترل

د – Code : Strict Source Route برابر با 10001001 می‌باشد : کنترل

ه- Code : Loose Source Route برابر با 10000011 می‌باشد : کنترل

ی – Code : Timestamp برابر با 01000100 می‌باشد : خطایابی و مدیریت

www.Abiri.ir

* مثال ۱۲

یکی از برنامه‌های موجود در یونیکس برای چک کردن حرکت بسته‌های IP، برنامه ping است. در فصل بعدی، به طور کاملتر درباره برنامه ping صحبت خواهیم کرد. در این مثال می‌خواهیم نشان دهیم که چگونه از این برنامه استفاده کنیم تا بدانیم که آیا یک کامپیوتر قابل دسترسی است یا نه. سروری در De Anza College به نام fhda.edu را ping می‌کنیم. نتیجه نشان می‌دهد که آدرس IP مقصد برابر با 153.18.8.1 است.

```
$ ping fhda.edu
PING fhda.edu ( 153.18.8.1 ) 56 ( 84 ) bytes of data.
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ): icmp_seq = 0 ttl=62 time=1.87 ms
...
```

نتیجه فوق، تعداد بایتهای مورد استفاده را نیز نشان می‌دهد.

* مثال ۱۳

همچنین می‌توانیم از ping با گزینه R- برای پیاده‌سازی گزینه Record Route استفاده کنیم.

```
$ ping -R fhda.edu
PING fhda.edu ( 153.18.8.1 ) 56 ( 124 ) bytes of data.
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ) : icmp_seq=0 ttl=62 time=2.70 ms
RR: voyager.deanza.fhda.edu ( 153.18.17.11 )
Dcore_G0_3-69.fhda.edu ( 153.18.251.3)
Dbackup_V13.fhda.edu ( 153.18.191.249 )
Tiptoe.fhda.edu ( 153.18.8.1 )
Dbackup_V62.fhda.edu ( 153.18.251.34 )
Dcore_G0_1-6.fhda.edu ( 153.18.31.254 )
Voyager.deanza.fhda.edu ( 153.18.17.11 )
```

نتیجه فوق همچنین اینترفیس‌ها و آدرسهای IP را نشان می‌دهد.

* مثال ۱۴

برنامه traceroute همچنین می‌تواند برای نشان دادن مسیری که بسته طی می‌کند، به کار رود.

```
$ traceroute fhda.edu
traceroute to fhda.edu ( 153.18.8.1 ), 30 hops max, 38 byte packets
1 Dcore_G0_1-6.fhda.edu ( 153.18.31.254 ) 0.972 ms 0.902 ms 0.881ms
2 Dbackup_V69.fhda.edu ( 153.18.251.4 ) 2.113 ms 1.996 ms 2.059 ms
3 tiptoe.fhda.edu ( 153.18.8.1 ) 1.791 ms 1.741 ms 1.751 ms
```

نتیجه فوق، سه روتری که بسته از آنها عبور کرده است را نشان می‌دهد.

www.Abiri.ir

* مثال ۱۵

می‌توان از برنامه traceroute برای پیاده‌سازی گزینه Loose Source Route استفاده کرد. گزینه g- به ما امکان مشخص کردن روترهایی که بسته از آنها باید عبور کند را می‌دهد. آنچه که به دنبال می‌آید، نشان می‌دهد که چطور می‌توانیم بسته‌ای را برای سرور fhda.edu بفرستیم و در ضمن از او بخواهیم که از روتر 153.18.251.4 نیز عبور کند.

```
$ traceroute -g 153.18.251.4 fhda.edu.
traceroute to fhda.edu ( 153.18.8.1 ), 30 hops max, 46 byte packets
1 Dcore_G0_1-6.fhda.edu ( 153.18.31.254 ) 0.976 ms 0.906 ms 0.889 ms
2 Dbackup_V69.fhda.edu ( 153.18.251.4 ) 2.168 ms 2.148 ms 2.037 ms
```

* مثال ۱۶

می‌توان از برنامه traceroute برای پیاده‌سازی گزینه Strict Source Route استفاده کرد. گزینه G- بسته را مقید می‌کند، از روترهایی که لیست آنها در خط فرمان آمده است، عبور نماید. آنچه که به دنبال می‌آید، نشان می‌دهد چگونه می‌توانیم بسته‌ای را برای سرور fhda.edu ارسال کنیم و همچنین بسته را مقید می‌کنیم که از روتر 153.18.251.4 عبور کند.

```
$ traceroute -G 153.18.251.4 fhda.edu.
traceroute to fhda.edu ( 153.18.8.1 ), 30 hops max, 46 byte packets
1 Dbackup_V69.fhda.edu ( 153.18.251.4 ) 2.168 ms 2.148 ms 2.037 ms
```

۸-۴ Checksum : جمع آزما

متد آشکارسازی خطا که اکثر پروتوکلهای TCP/IP از آن استفاده می‌کنند، Checksum است. اگر پیغام ارسالی حین عبور در شبکه دچار خطا شده باشد، Checksum آن را آشکار می‌کند. می‌توان آن را اطلاعات زائدی (redundant) در نظر گرفت که برای آشکارسازی خطا به پیغام اضافه می‌شود.

محاسبات مربوط به Checksum

مقدار Checksum در فرستنده محاسبه می‌شود و مقدار بدست آمده به همراه بسته ارسال می‌شود. گیرنده عملیات مشابهی را روی کل بسته و نیز Checksum انجام می‌دهد. اگر نتیجه رضایتبخش باشد، بسته پذیرفته می‌شود، در غیر این صورت، بسته دور انداخته می‌شود.

www.Abiri.ir

در فرستنده، بسته به بخشهای n بیتی (که n معمولا برابر با 16 است) تقسیم می
بن بخشها با استفاده از ریاضیات مکمل یک با هم جمع می‌شوند و نتیجه حاصل، یک ع
جمع n بیتی است. سپس مکمل حاصل جمع محاسبه می‌شود (0 ها به 1 و 1 ها به 0 ت
ی‌شوند) و مقدار Checksum به این ترتیب به دست می‌آید. فرستنده برای مح
Checksum عملیات زیر را انجام می‌دهد :

۱- بسته به k بخش، هر یک حاوی n بیت، تقسیم می‌شود.

۲- با استفاده از ریاضیات مکمل یک، همه بخشها را با هم جمع می‌کنیم.

۳- مکمل حاصل جمع بدست آمده را محاسبه می‌کنیم و به این ترتیب Checksum بدست می‌آید.

* محاسبه Checksum در گیرنده

گیرنده، بسته دریافتی را به k بخش تقسیم می‌کند و آن قسمتها را با هم جمع می
سپس مکمل حاصل جمع را به دست می‌آورد. اگر نتیجه نهائی برابر با 0 باشد، بسته پذ
می‌شود، در غیر این صورت دور انداخته می‌شود. شکل ۸–۲۲ به صورت گرافیکی نشان م
که در فرستنده و گیرنده چه اتفاقی می‌افتد.

گفتیم اگر ضمن حرکت بسته در شبکه خطایی رخ ندهد، باید پس از مکمل گر
حاصل جمع بدست آمده در گیرنده، به تمام صفر برسیم و این مطلب بنابه قواعد ریا
مکمل یک صحیح است.

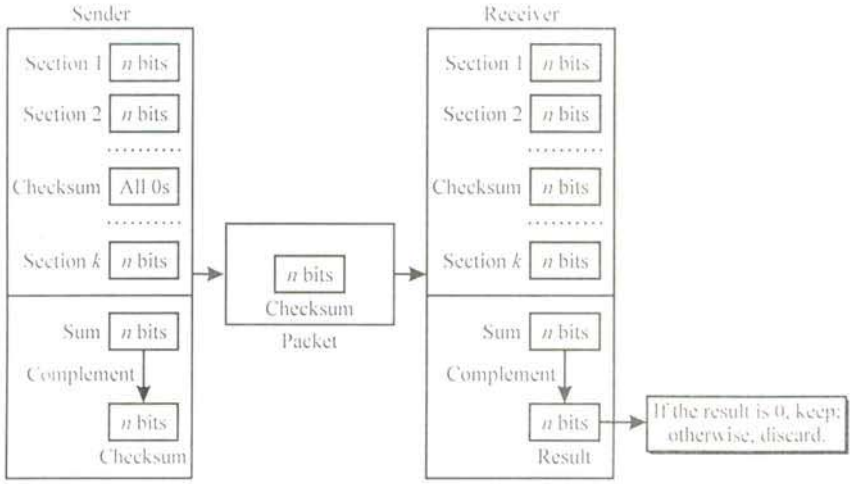
فرض کنید پس از جمع کردن تمام بخشها در فرستنده، به عددی برابر با T ب
وقتی که مکمل آن را با استفاده از ریاضیات مکمل یک بدست می‌آوریم، به T- می
بنابراین اگر مجموع همه قسمتها برابر با T باشد، مکمل آن برابر با T- خواهد بود.

وقتی که گیرنده، بسته را دریافت می‌کند، همه قسمتها را با هم جمع می‌کند. ب
در واقع T و T- با هم جمع می‌شوند که در ریاضیات مکمل یک مقدار 0- را می‌دهد. وق
مکمل آن محاسبه می‌شود، 0- به 0 تبدیل می‌شود. بنابراین اگر نتیجه نهایی برابر 0 باشد
پذیرفته می‌شود، در غیر این صورت، بسته دور انداخته می‌شود (شکل ۸–۲۳ را ببینید).

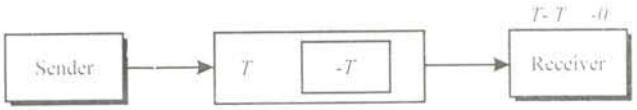
www.Abiri.ir

* پیاده‌سازی Checksum در بسته IP

تقسیم می‌شود و این بخشها با هم جمع می‌شوند. مکمل حاصل جمع محاسبه می‌شود و نتیجه در فیلد Checksum نوشته می‌شود.



شکل ۸-۲۲ : مفهوم Checksum



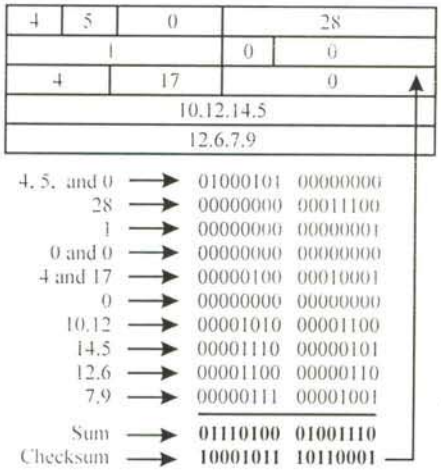
شکل ۸-۲۳ : Checksum در ریاضیات مکمل یک

Checksum در بسته IP ، فقط هدر را در برمی‌گیرد و لذا دیتا را پوشش نمی‌دهد. دو دلیل خوب برای این کار وجود دارد: اول آنکه تمام پروتوکلهای لایه بالاتر که دیتا را در بسته IP کپسول‌بندی می‌کنند، یک فیلد Checksum دارند که کل بسته را در برمی‌گیرد. بنابراین لازم نیست که Checksum بسته IP، دیتای کپسول‌بندی شده را چک کند. دوم آنکه، هر روتری که بسته را می‌گیرد، هدر بسته IP را تغییر می‌دهد، اما دیتای بسته IP تغییر نمی‌کند. بنابراین Checksum فقط شامل آن قسمتی می‌شود که تغییر کرده است. اگر دیتای بسته در محاسبه Checksum وارد می‌شد، همه روترهایی که بسته را دریافت می‌کنند، می‌بایست مقدار Checksum را برای کل بسته محاسبه کنند و لذا زمان پردازش افزایش می‌یافت.

www.Abiri.ir

*** مثال ۱۷**

شکل ۸–۲۴ مثالی از محاسبه Checksum برای یک بسته IP بدون گزینه را نشان می‌دهد. هدر به بخشهای 16 بیتی تفکیک می‌شود. همه بخشها با هم جمع می‌شوند و مکمل حاصل جمع، محاسبه می‌شود. نتیجه در فیلد Checksum نوشته می‌شود.

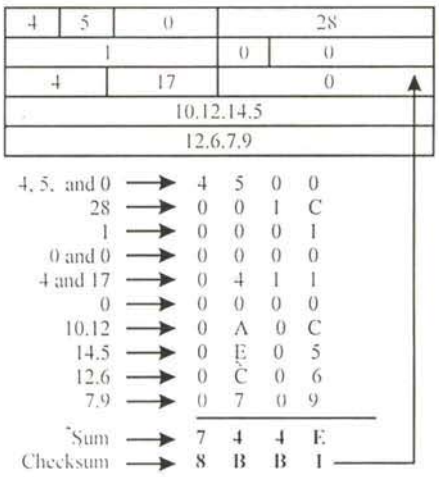


شکل ۸–۲۴ : مثالی از محاسبه Checksum بر حسب باینری

*** مثال ۱۸**

اجازه دهید محاسبات مثال هفدهم را به صورت هگزادسیمال انجام دهیم. هر سطر، حاوی چهار رقم هگزادسیمال است. ابتدا حاصل جمع را محاسبه می‌کنیم. توجه نمائید که اگر نتیجه جمع در هر ستون، بیش از یک رقم هگزادسیمال باشد، رقم سمت راست، زیر همان ستون نوشته می‌شود و رقم سمت چپ، به ستون بالاتر منتقل می‌شود. با مکمل گرفتن از حاصل جمع، مقدار Checksum را بدست می‌آوریم. برای انجام این کار، کافی است هر رقم هگزا را از 15 کم کنیم (همانگونه که در ریاضیات باینری، هر رقم را از 1 کم می‌کنیم). بنابراین مکمل E (14) برابر با 1 و مکمل 4 برابر با B (11) است. شکل ۸–۲۵ نحوه محاسبه را نشان می‌دهد. دقت نمائید که نتیجه (8BB1) دقیقاً همان نتیجه‌ای است که در مثال هفدهم به آن رسیدیم.

www.Abiri.ir



شکل ۸-۲۵ : مثالی از محاسبه Checksum بر حسب هگزادسیمال

۸-۵ نرم‌افزار IP

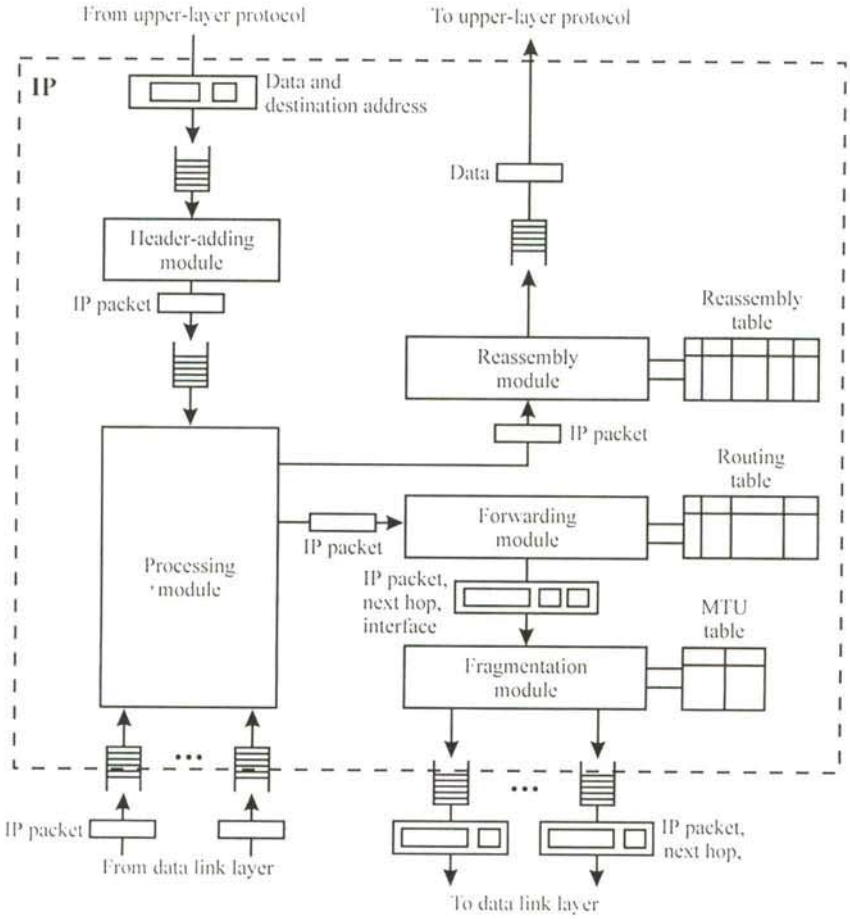
در این بخش، مثال ساده شده‌ای از نرم‌افزارIP را ارائه می‌دهیم. هدف ما نشان دادن روابط بین مفاهیم مختلفی است که در این فصل مورد بحث و بررسی قرار گرفتند. شکل ۸-۲۶ هفت جزء نرم‌افزار ساده شده IP و رابطه بین آنها را نشان می‌دهد.

گرچه IP حاوی گزینه‌هایی است اما به منظور ساده‌تر کردن فهم نرم‌افزار IP ، در این سطح، پردازش گزینه‌ها را حذف کردیم. به علاوه در طرح این نرم افزار، بیشتر به ساده‌سازی پرداخته‌ایم و کمتر به بالا بردن راندمان کاری آن، توجه نموده‌ایم.

می‌توانیم بگوئیم که نرم‌افزار IP حاوی هشت جزء است : ***ماجول اضافه کننده هدر***[1]، ***ماجول پردازش‌کننده***[2]، ***ماجول هدایت بسته***[3] ، ***ماجول قطعه قطعه سازی***[4] ، ***ماجول مونتاژ مجدد***[5]، یک جدول مسیردهی ، یک جدول MTU و یک جدول مونتاژ. به علاوه، نرم افزار IP حاوی صفهای ورودی و خروجی است.

1- Header Adding Module
2- Processing Module
3- Forwarding Module
4- Fragmentation Module
5- Reassembly Module

www.Abiri.ir



شکل ۸-۲۶ : اجزای نرم‌افزار ساده شده IP

نرم افزار IP، بسته‌ای را یا از لایه دیتالینک زیرین و یا از یک پروتوکل لایه بالاتر دریافت می‌کند. اگر بسته از یک پروتوکل لایه بالاتر بیاید، به لایه دیتالینک داده می‌شود تا منتقل شود (مگر آنکه حاوی آدرس برگردان 127.x.y.z باشد). اگر بسته از لایه دیتالینک بیاید، یا پس از مسیردهی شدن مجدداً به لایه دیتالینک داده می‌شود (مثلاً در یک روتر) و یا اگر آدرس IP مقصد بسته، برابر با آدرس IP ایستگاه دریافت کننده بسته باشد، آن را به پروتوکل لایه بالاتر تحویل می‌دهد. دقت نمائید که از چندین صف برای ارتباط با لایه دیتالینک استفاده کردیم زیرا که روتر حاوی چندین اینترفیس است.

www.Abiri.ir

ماجول اضافه کننده هدر

ماجول اضافه کننده هدر، دیتا و نیز آدرس IP مقصد را از لایه بالاتر دریافت می‌کند و سپس دیتا را با اضافه کردن یک هدر IP، داخل یک بسته IP، کپسول‌بندی می‌کند.

* جریان رویدادهای ماجول اضافه کننده هدر

ورودی: دیتا، آدرس مقصد.

۱- دیتا را در یک بسته IP کپسول‌بندی می‌کند.

۲- Checksum را حساب می‌کند و آن را در فیلد Checksum قرار می‌دهد.

۳- بسته IP حاصل را در صف مربوطه قرار می‌دهد.

۴- برمی‌گردد.

ماجول پردازش‌کننده

ماجول پردازش‌کننده، بخش اصلی نرم افزار IP است. در نرم‌افزار ما، ماجول پردازش‌کننده، یک بسته IP را از یک اینترفیس و یا از ماجول اضافه کننده هدر، دریافت می‌کند. هر دو حالت، مشابه است. مهم نیست که بسته IP از کجا آمده است، فقط باید پردازش شود و به سمت مقصد، مسیردهی شود.

ماجول پردازش‌کننده ابتدا چک می‌کند ببیند آیا بسته دریافتی حاوی یک آدرس برگردان (آدرس مقصد 127.x.y.z) است و یا بسته‌ای است که به مقصد نهائی خود رسیده است. در هر دو حالت، بسته به ماجول مونتاژ مجدد، داده می‌شود.

اگر ماشین، یک روتر باشد، یک واحد از فیلد TTL کم می‌کند. اگر مقدار جدید TTL برابر یا کمتر از صفر باشد، بسته IP را دور می‌اندازد و یک پیغام ICMP (فصل نهم را ببینید) برای فرستنده، ارسال می‌شود. اگر بعد از یک واحد کاهش، مقدار TTL بزرگتر از صفر باشد، ماجول پردازش‌کننده، بسته IP را برای ماجول هدایت بسته می‌فرستد (شکل ۸– ۲۶ را ببینید).

* جریان رویدادهای ماجول پردازش‌کننده

۱ – یک بسته IP را از یکی از صفهای ورودی برمی‌دارد.

۲ ـ اگر آدرس مقصد برابر با 127.x.y.z باشد و یا با یکی از آدرسهای محلی، مطابقت داشته باشد:

۱- بسته IP را به ماجول مونتاژ مجدد می‌دهد.

۲- برمی‌گردد.

www.Abiri.ir

۳- اگر ماشین، یک روتر باشد :
۱- یک واحد از TTL کم می‌کند.
۴- اگر TTL کمتر از یا برابر با صفر است :
۱- بسته IP را دور می‌اندازد.
۲- یک پیغام خطای ICMP را ارسال می‌کند.
۳- برمی‌گردد.
۵ - بسته IP را به ماجول هدایت بسته می‌دهد.
۶ - برمی‌گردد.

صفها

نرم‌افزار ما از دو نوع صف استفاده می‌کند: صفهای ورودی و صفهای خروجی. صفهای ورودی، بسته‌های IP که از لایه دیتالینک یا از پروتوکلهای لایه بالاتر می‌آیند را در خود ذخیره می‌کنند. صفهای خروجی، بسته‌های IP که به لایه دیتالینک و یا پروتوکلهای لایه‌های بالاتر می‌روند را ذخیره می‌کنند. ماجول پردازش‌کننده، بسته IP را از صفهای ورودی برمی‌دارد. ماجولهای قطعه قطعه سازی و مونتاژ مجدد، بسته‌های IP را داخل صفهای خروجی قرار می‌دهند.

جدول مسیردهی

در فصل ششم درباره جدول مسیردهی صحبت کردیم. ماجول هدایت بسته از جدول مسیردهی استفاده می‌کند تا آدرس هاپ بعدی بسته را تعیین کند.

ماجول هدایت بسته IP

در فصل ششم درباره ماجول هدایت بسته IP صحبت کردیم. ماجول هدایت بسته IP، یک بسته IP را از ماجول پردازش‌کننده دریافت می‌کند. اگر قرار است که بسته به سمت مقصد هدایت شود، باید به این ماجول تحویل داده شود. این ماجول، آدرس IP ایستگاه بعدی به همراه شماره اینترفیسی که بسته باید روی آن اینترفیس قرار گیرد را پیدا می‌کند. سپس بسته را به همراه این اطلاعات به ماجول قطعه قطعه سازی می‌دهد.

www.Abiri.ir

جدول MTU

ماجول قطعه قطعه سازی بسته از جدول MTU استفاده می‌کند تا ماکزیمم واحد انتقال یک اینترفیس مشخص را پیدا کند. شکل ۸-۲۷ فرمت یک جدول MTU را نشان می‌دهد.

Interface Number	MTU
.....	.....

شکل ۸-۲۷ : جدول MTU

ماجول قطعه قطعه سازی بسته IP

در نرم‌افزار ما، ماجول قطعه قطعه سازی بسته IP ، یک بسته IP را از ماجول هدایت بسته IP دریافت می‌کند. ماجول هدایت بسته IP، آدرس IP هاپ بعدی (مقصد نهائی در تحویل مستقیم و یا روتر بعدی در تحویل غیر مستقیم) و شماره اینترفیسی که بسته IP از طریق آن باید به بیرون ارسال شود را به ماجول قطعه قطعه سازی بسته می‌دهد. این ماجول پس از مشورت با جدول MTU، مقدار MTU متناظر با شماره اینترفیس داده شده را بدست می‌آورد. اگر طول بسته IP از مقدار MTU بزرگتر باشد، ماجول قطعه قطعه سازی، آن بسته را قطعه قطعه می‌کند، یک هدر IP به هر قطعه اضافه می‌کند و بسته‌های IP حاصل را به نرم‌افزار ARP (فصل هفتم را ببینید) می‌دهد تا آدرس فیزیکی ایستگاه بعدی بدست آید.

* جریان رویدادهای ماجول قطعه قطعه سازی بسته IP

ورودی : یک بسته IP از ماجول مسیردهی

۱- سایز بسته IP را محاسبه می‌کند.

۲- اگر سایز بسته از MTU اینترفیسی که بسته باید روی آن گذاشته شود، بزرگتر باشد:

۱- اگر بیت DF برابر با 1 باشد :

۱- بسته IP را دور می‌اندازد.

۲- یک پیغام خطای ICMP می‌فرستد (فصل نهم را ببینید).

۳- برمی‌گردد.

۲- در غیر این صورت :

www.Abiri.ir

۱- ماکزیمم سایز را محاسبه می‌کند.
۲- بسته IP را به چند قطعه، تقسیم می‌کند.
۳- هدر IP را به هر قطعه اضافه می‌کند.
۴- گزینه‌های لازم را به هر قطعه اضافه می‌کند.
۵- قطعات را ارسال می‌کند.
۶- برمی‌گردد.
۳- در غیر این صورت :
۱- بسته IP را ارسال می‌کند.
۴- برمی‌گردد.

جدول مونتاژ مجدد

ماجول مونتاژ مجدد از این جدول که در نرم‌افزار ما حاوی این پنج فیلد است، استفاده می‌کند: ***حالت***[۱] ، ***آدرس IP مبدأ***[۲] ، ***شناسه بسته IP***[۳] ، ***مهلت مقرر***[۴] و ***قطعات***[۵] .

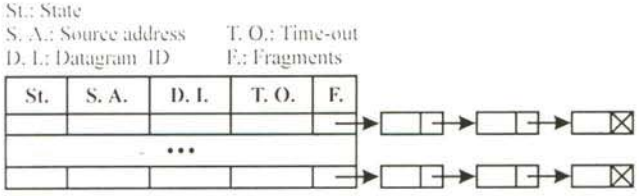
مقدار فیلد حالت می‌تواند ***آزاد***[۶] یا ***در حال استفاده***[۷] باشد. فیلد آدرس IP مبدأ، نشان می‌دهد که این بسته IP از کجا آمده است. ID شناسه‌ای است که به طور منحصر به فردی یک بسته IP و تمام قطعات متعلق به آن را تعیین می‌کند. مهلت مقرر، مقدار زمان از قبل ست شده‌ای است که تا قبل از انقضای آن، کلیه قطعات باید از راه برسند. بالاخره، فیلد قطعات، حاوی ***نشانه‌گری***[۸] به ***لیست دنباله قطعات***[۹] است.

ماجول مونتاژ مجدد

ماجول مونتاژ مجدد، قطعات IP ای که به مقصد نهایی خود رسیده‌اند را از ماجول پردازش‌کننده دریافت می‌کند. در نرم‌افزار ما، ماجول مونتاژ مجدد، با یک بسته IP ای که قطعه قطعه نشده است، به عنوان یک بسته IP قطعه قطعه شده‌ای رفتار می‌کند که فقط حاوی یک قطعه است.

1- State
2- Source IP address
3- Datagram ID
4-Time-out
5- Fragments
6- FREE
7- IN-USE
8- Pointer
9- Linked list of fragments

www.Abiri.ir



شکل ۸-۲۸ : جدول مونتاژ مجدد

از آنجا که پروتوکل IP، یک پروتوکل فاقد اتصال است، هیچ تضمینی وجود ندارد که قطعات به ترتیب به مقصد برسند. به علاوه، قطعات یک بسته IP ممکن است به همراه قطعات یک بسته IP دیگر دریافت شوند. برای کنترل این اوضاع، همانطور که قبلاً گفتیم ماجول مونتاژ مجدد از یک جدول مونتاژ و لیست دنباله قطعات استفاده می‌کند.

کار ماجول مونتاژ مجدد، پیدا کردن بسته IP ای است که قطعات، متعلق به آن می‌باشند تا بتواند کل قطعات یک بسته IP را با نظم و ترتیب اولیه مجدداً مونتاژ کند. اگر مهلت مقرر مربوطه منقضی شود و یک و یا تعدادی از قطعات از راه نرسیده باشند، بقیه قطعات دریافتی نیز دور انداخته می‌شوند.

*** جریان رویدادهای ماجول مونتاژ مجدد**

ورودی : یک بسته IP از ماجول پردازش‌کننده

۱- اگر مقدار offset برابر با 0 و بیت MF برابر با 0 باشد :
  ۱- بسته IP را داخل صف مناسب قرار می‌دهد.
  ۲- برمی‌گردد.
۲- داخل جدول مونتاژ مجدد به دنبال المان متناظر با این بسته IP می‌گردد.
۳- اگر پیدا نشد :
  ۱- یک المان جدید در جدول ایجاد می‌کند.
۴- قطعه را در مکان مناسب داخل لیست دنباله قطعات قرار می‌دهد.
  ۱- اگر همه قطعات از راه رسیده باشند :
    ۱- قطعات را مجدداً مونتاژ می‌کند.
    ۲- بسته IP را به پروتوکل متناظر لایه بالاتر می‌دهد.
    ۳- برمی‌گردد.
  ۲- در غیر این صورت :
    ۱- مقدار مهلت مقرر را چک می‌کند.

www.Abfn.ir

۲- اگر از مهلت مقرر گذشته باشد :
۱- همه قطعات دریافتی را دور می‌اندازد.
۲- یک پیغام خطای ICMP ارسال می‌کند (فصل نهم را ببینید)
۵- برمی‌گردد

۸-۶ عبارات کلیدی

تحویل بر اساس نهایت تلاش (best effort delivery)
جمع‌آزما (Checksum)
فیلد کد (code field)
نقطه کدگذاری (Codepoint)
فیلد دیتا (data field)
دیتاگرام (datagram)
آدرس مقصد (destination address)
سرویسهای ناهمسان (differentiated services)
گزینه انتهای گزینه‌ها (End of option)
ماجول هدایت بسته (forwarding module)
قطعه قطعه سازی (fragmentation)
ماجول قطعه قطعه سازی (fragmentation Module)
مکان نسبی این قطعه در کل بسته (fragmentation offset)
ماجول اضافه کننده هدر (header adding module)
طول هدر (header length)
صف ورودی (input queue)
پروتوکل اینترنت (internet protocol : IP)
فیلد طول (length field)
گزینه مسیردهی نه چندان قطعی از مبدأ (Loose Source Route option)
ماکزیمم واحد انتقال (Maximum Transfer Unit : MTU)
صف خروجی (output queue)
ابزار ping
فیلد نشانه‌گر (pointer field)
ارجحیت (precedence)
ماجول پردازش‌کننده (processing module)

www.Abiri.ir

ماجول مونتاژ مجدد (reassembly module)
جدول مونتاژ مجدد (reassembly table)
گزینه ضبط مسیر (record route option)
نوع سرویس (Service Type)
آدرس مبدأ (Source address)
گزینه مسیردهی قطعی از مبدأ (Strict Source Route option)
طول عمر (Time To Live)
گزینه بر چسب زمانی (Timestamp option)
ابزارِ traceroute
نوع سرویس (Type Of Service : TOS)

۸-۷ خلاصه

- IP یک پروتوکل فاقد اتصال غیر قابل اطمینان برای تحویل بسته از مبدأ به مقصد است.
- بسته‌ها در لایه IP ، دیتاگرام نامیده می‌شوند.
- یک بسته IP حاوی یک هدر (20 الی 60 بایتی) و دیتا است.
- هدر IP حاوی اطلاعات زیر است: شماره ویراست، طول هدر، سرویسهای ناهمسان، طول بسته، شناسه بسته، پرچمهای قطعات، مکان نسبی قطعه در کل بسته، طول عمر، پروتوکل، checksum، آدرس IP مبدأ، آدرس IP مقصد و گزینه‌ها.
- ماکزیمم طول یک بسته IP برابر با 65535 بایت است.
- MTU ماکزیمم تعداد بایتی است که پروتوکل لایه دیتالینک می‌تواند در خود کپسول‌بندی کند. از یک پروتوکل به پروتوکل دیگر مقدار MTU تغییر می‌کند.
- قطعه قطعه سازی، تقسیم یک بسته IP به قطعات کوچکتر است به طوری که این قطعات، داخل فیلد دیتای پروتوکل لایه دیتالینک به حداکثر طولِ مجاز MTU، جای گیرند.
- فیلدهایی از هدر IP که مرتبط با قطعه قطعه سازی هستند، عبارتند از شناسه بسته، پرچمهای قطعه قطعه سازی و مکان نسبی قطعه در بسته.
- هدر بسته IP حاوی یک بخش ثابت 20 بایتی و یک بخش متغیر به طول حداکثر 40 بایت است که گزینه‌ها را در بر می‌گیرد.
- گزینه‌ها در هدر IP برای تست و خطایابی در شبکه به کار می‌روند.

www.Abiri.ir

• هر گزینه حاوی اطلاعات زیر است: فیلد کد که خود گزینه را مشخص می‌کند، فیلد طول که طول گزینه بر حسب بایت را نشان می‌دهد و بالاخره فیلد دیتا که دیتای خاص آن گزینه است.

• شش گزینه، هر کدام برای انجام کار بخصوصی هستند. کارهایی که انجام می‌دهند عبارتند از : الف – NOP : بین سایرگزینه‌ها قرار می‌گیرد و عمل ترازبندی هدر را انجام می‌دهد ب – EOP : به صورت یک صفر در انتهای گزینه‌ها قرار می‌گیرد. ج- Record Route : مسیری که بسته طی می‌کند را ضبط می‌کند. د- Strict Source Route : مسیر را از طرف مبدأ ارسال‌کننده بسته تعیین می‌کند. ه- Loose Source Route : روترهایی که حتماً بسته باید از آنها عبور کند را تعیین می‌کند. ی- Timestamp : زمان پردازش بسته در روترها را ثبت می‌کند.

• برنامه‌های ping و traceroute در یونیکس را می‌توان برای پیاده‌سازی برخی از گزینه‌های IP بکار برد.

• فیلد آشکارسازی خطا که IP از آن استفاده می‌کند، Checksum است.

• Checksum برای جمع زدن بخشهای هم اندازه هدر IP ، از ریاضیات مکمل یک استفاده می‌کند. مکمل حاصل جمع، داخل فیلد Checksum نوشته می‌شود. گیرنده نیز از ریاضیات مکمل یک برای چک کردن هدر استفاده می‌کند.

• نرم‌افزار IP می‌تواند شامل اجزای زیر باشد: ماجول اضافه کننده هدر، ماجول پردازش‌کننده، ماجول هدایت بسته، ماجول قطعه قطعه سازی، ماجول مونتاژ مجدد، جدول مسیردهی، جدول MTU و جدول مونتاژ مجدد.

۸–۸ مجموعه تمرین‌ها

تمرین‌ها

۱- کدام یک از فیلدهای هدر IP ، از یک روتر به روتر بعد تغییر می‌کنند.

۲- اگر کل طول برابر با 1200 بایت باشد و 1176 بایت آن، دیتایی باشد که از لایه بالاتر آمده باشد، مقدار HLEN را محاسبه کنید.

۳- جدول ۸–۵ مقدار MTU را برای پروتوکلهای مختلف نشان می‌دهد . مقدار MTU از 296 تا 65535 است. مزیتِ داشتنِ یک MTU بزرگ چیست؟ مزیتِ داشتنِ یک MTU کوچک چیست؟

۴- با داشتن یک بسته IP قطعه قطعه شده که مقدار offset آن برابر با 120 است، چگونه می‌توان شماره اولین بایت و آخرین بایت را تعیین کرد؟

www.Abiri.ir

۵- یک بسته IP باید از میان روتر 128.46.10.5 عبور کند. هیچ محدودیت دیگری بر روترهایی که بسته باید از آنها عبور کند، وجود ندارد. گزینه‌ها را به همراه مقدارشان ترسیم نمائید.

۶- اگر فیلد Flags در گزینه Timestamp برابر با 1 باشد، حداکثر آدرس IP چند روتر را می‌توان ثبت کرد؟ چرا؟

۷- آیا مقدار HLEN در یک بسته IP می‌تواند کوچکتر از 5 باشد؟ چه زمانی دقیقاً برابر با 5 است؟

۸- مقدار HLEN در یک بسته IP برابر با 7 است. چند بایت option در هدر IP وجود دارد؟

۹- سایز فیلد option یک بسته IP برابر با 20 است. مقدار HLEN چقدر است؟ این مقدار بر حسب باینری، چقدر است؟

۱۰- مقدار فیلد total length در یک بسته IP برابر با 36 و مقدار فیلد HLEN برابر با 5 است. این بسته، چند بایت دیتا را حمل می‌کند؟

۱۱- یک بسته IP، 1024 بایت دیتا را حمل می‌کند. اگر هیچگونه option ای وجود نداشته باشد، مقدار فیلد HLEN چقدر است؟ مقدار فیلد کل طول چقدر است؟

۱۲- یک کامپیوتر در حال ارسال 100 بسته IP برای کامپیوتر دیگری است. اگر شناسه اولین بسته برابر با 1024 باشد، شناسه آخرین بسته چقدر است؟

۱۳- یک بسته IP از راه رسیده است که در آن مقدار offset برابر با 0 و مقدار بیت M نیز برابر با 0 است. آیا این بسته، اولین قطعه، یک قطعه میانی یا قطعه آخر است؟

۱۴- یک قطعه IP با مقدار offset برابر با 100 از راه رسیده است. قبل از دیتای این قطعه، چند بایت دیتا توسط فرستنده ارسال شده است؟

۱۵- یک بسته IP که هدر آن حاوی اطلاعات زیر است (بر حسب ارقام هگزا) از راه رسیده است :

45 00 00 54 00 03 00 00 20 06 00 00 7C 4E 03 02 B4 0E 0F 02

الف ـ آیا هیچ option ای وجود دارد؟

ب ـ آیا این بسته، قطعه قطعه شده است؟

ج - سایز دیتا چقدر است؟

د ـ آیا از checksum استفاده شده است؟

ه- این بسته از چند روتر دیگر می‌تواند عبور کند؟

و ـ شناسه بسته چند است؟

ی ـ نوع سرویس چیست؟

www.Abiri.ir

۱۶- در یک بسته IP، بیت M برابر با صفر، مقدار HLEN برابر با 5، مقدار کل طول برابر با 200 و مقدار offset برابر با 200 است. در این بسته، شماره اولین بایت و شماره آخرین بایت چیست؟ آیا این آخرین قطعه، اولین قطعه یا یک قطعه میانی است؟

فعالیتهای تحقیقاتی

۱۷- برای چک کردن مسیر یک بسته تا مقصد، از برنامه ping با گزینه R- استفاده کنید. نتیجه را تفسیر کنید.

۱۸- از برنامه traceroute با گزینه g- برای پیاده‌سازی گزینه Loose Source Route استفاده کنید. از چند روتر بین مبدأ و مقصد استفاده کنید. نتیجه را تفسیر کنید و ببینید آیا بسته از همه روترها عبور کرده است.

۱۹- از برنامه traceroute با گزینه G- برای پیاده‌سازی گزینه Strict Source Route استفاده کنید. از چند روتر بین مبدأ و مقصد استفاده کنید. نتیجه را تفسیر نمایید و ببینید آیا بسته از تمام روترها عبور کرده است و آیا از روتری که آدرس IP آن نیامده است نیز عبور کرده است یا خیر.

۲۰- یک کار تحقیقی انجام دهید و RFC های مرتبط با پروتوکلهای IP را پیدا کنید. کدامیک، قطعه قطعه سازی را تعریف می‌کند؟

۲۱- RFC های مرتبط با گزینه‌های IP را پیدا کنید. کدامیک گزینه Record Route را تعریف می‌کند؟ کدامیک گزینه Loose Source Route را تعریف می‌کند؟ کدامیک گزینه Strict Source Route را تعریف می‌کند؟

www.Abiri.ir

فصل نهم

پروتوکل ICMP

همانطور که در فصل هشتم بحث آن آمد، IP سرویس غیر قابل اطمینان و فاقد اتصال، برای تحویل بسته‌هاست. IP به این صورت طراحی شده است تا به شکل مؤثری از منابع شبکه ، استفاده گردد. پروتوکل IP ، سرویس تحویل بسته‌های IP از مبدأ اولیه به مقصد نهایی، بر اساس نهایت تلاش است. با این حال، دو نقیصه دارد. فاقد کنترل خطا و فاقد مکانیزمهای کمک‌رسانی است.

پروتوکل IP حاوی هیچ‌گونه مکانیزم تصحیح خطا یا گزارش خطا نیست. اگر اشتباهی پیش آید، چه رخ می‌دهد؟ اگر روتری به دلیل پیدا نکردن روتر بعدی برای تحویل بسته به مقصد و یا به دلیل صفر شدن فیلد طول عمر بسته، مجبور به دور انداختن بسته IP شود، چه رخ می‌دهد؟ اگر کامپیوتر مقصد نهایی، به دلیل عدم دریافت همه قطعات یک بسته در خلال یک فاصله زمانی از قبل تعیین شده، مجبور به دور انداختن سایر بسته‌های دریافتی گردد، چه اتفاقی می‌افتد؟ اینها نمونه‌هایی هستند از اینکه خطایی رخ داده است ولی پروتوکل IP هیچ‌گونه مکانیزمی برای گزارش خطا به کامپیوتر ارسال کننده بسته IP ندارد.

همچنین پروتوکل IP فاقد مکانیزمی برای ارسال درخواست و دریافت پاسخ از کامپیوترها و روترهاست. گاهی اوقات یک کامپیوتر می‌خواهد بداند که آیا کامپیوتر و یا روتر دیگری به شبکه متصل است یا خیر. گاهی اوقات نیز مدیران شبکه نیاز به دریافت اطلاعاتی از یک کامپیوتر و روتر دیگر دارند.

۹-۱ پروتوکل ICMP

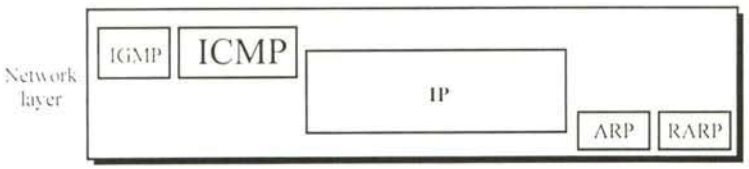
پروتوکل پیغامهای کنترلی اینترنت[1] برای جبران دو نقیصه فوق الذکر طراحی شده است. ICMP همراهی برای IP است. شکل ۹-۱ جایگاه ICMP در لایه شبکه را نشان می‌دهد.

خود ICMP ، یک پروتوکل لایه شبکه است. با این حال بر خلاف انتظار، پیغامهای او مستقیماً به لایه دیتالینک داده نمی‌شوند، بلکه ابتدا به داخل بسته‌های IP کپسول‌بندی می‌شوند و سپس به لایه دیتالینک داده می‌شوند (شکل ۹-۲ را ببینید).

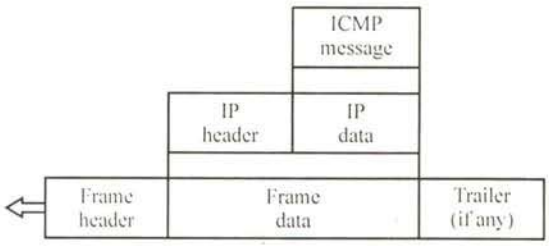
وقتی که بسته IP ،حامل پیغام ICMP باشد، مقدار فیلد protocol آن برابر با 1 می‌باشد.

1- Internet Control Message Protocol : ICMP

www.Abiri.ir



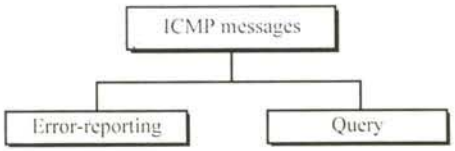
شکل ۹-۱ : جایگاه ICMP در لایه شبکه



شکل ۹-۲ : کپسول‌بندی ICMP

۹-۲ انواع پیغامهای ICMP

پیغامهای ICMP به دو گروه عمده تقسیم می‌شوند : ***پیغامهای گزارش خطا***[1] و ***پیغامهای درخواست و پاسخ***[2]. شکل ۹-۳ را ببینید.



شکل ۹-۳ : پیغامهای ICMP

پیغامهای گزارش خطا مشکلاتی را گزارش می‌کنند که یک روتر و یا کامپیوتر (مقصد) ممکن است حین پردازش یک بسته IP با آن مواجه شوند.

1- error reporting messages
2- query messages

www.Abiri.ir

پیغامهای درخواست و پاسخ که به صورت جفتی (درخواست و پاسخ) می‌باشند، به کامپیوتر یا مدیر شبکه کمک می‌کنند تا اطاعات مشخصی را از یک روتر یا کامپیوتر دیگری بگیرند. به عنوان مثال، وسائل به کمک این پیغامها، می‌توانند همسایگان خود را پیدا کنند. همچنین کامپیوترها به کمک این پیغامها، می‌توانند از وجود روترهای متصل به شبکه با خبر شوند و نیز روترها می‌توانند به کامپیوتر در تصحیح نحوه مسیردهی آن، کمک کنند. جدول ۹-۱ لیست پیغامهای ICMP را در هر گروه نشان می‌دهد.

Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection

Category	*Type*	*Message*
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply
	17 or 18	Address mask request or reply
	10 or 9	Router solicitation or advertisement

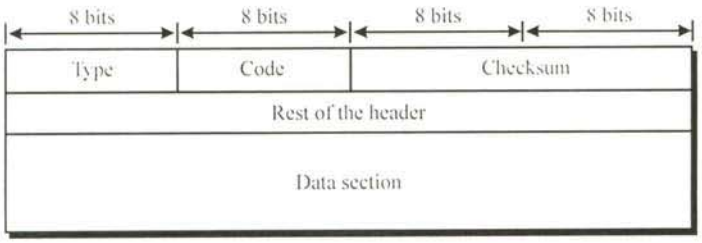
جدول ۹-۱ : پیغامهای ICMP

فرمت کلی پیغامهای ICMP

یک پیغام ICMP حاوی یک هدر 8 بایتی و یک بخش دیتا به طول متغیر است و فرمت کلی هدر برای هر نوع پیغام، متفاوت است. اولین 4 بایت در بین تمام انواع پیغامها، یکسان هستند. همانطور که شکل ۹-۴ نشان می‌دهد، اولین فیلد یعنی type، نوع پیغام را تعیین می‌کند. فیلد Code نشان می‌دهد که آن نوع پیغام، به چه دلیلی تولید شده است. آخرین فیلد مشترک بین انواع پیغامها، فیلد Checksum است. 4 بایت بعدی هدر، بستگی به نوع پیغام ICMP دارد.

بخش دیتای پیغامهای گزارش خطا حاوی اطلاعاتی است که برای پیدا کردن بسته اولیه‌ای که تولید خطا کرده است، به کار می‌رود. در پیغامهای درخواست و پاسخ، بخش دیتا حاوی اطلاعات اضافه‌ای بر اساس نوع درخواست و پاسخ است.

www.Abiri.ir



شکل ۹-۴ : فرمت کلی پیغامهای ICMP

پیغامهای ICMP از نوع گزارش خطا

یکی از مسئولیتهای اصلی ICMP ، گزارش خطاست. گرچه تکنولوژی توانسته است محیطهای انتقالی با قابلیت اطمینان بالا تولید کند، هنوز خطاهایی رخ می‌دهند که باید بررسی شوند. همانطور که در فصل هشتم بحث آن آمد، IP یک پروتوکل غیر قابل اطمینان است. معنی آن این است که کنترل خطا موضوعی نیست که IP به آن بپردازد. تا حدودی علت طراحی ICMP ، جبران این نقیصه بوده است. با این حال، ICMP خطاها را جبران نمی‌کند بلکه فقط آنها را گزارش می‌کند. تصحیح خطا به عهده پروتوکلهای لایه‌های بالاتر گذاشته شده است. گزارشهای خطا فقط برای مبدأ ارسال‌کننده پیغام ارسال می‌شوند زیرا که بسته IP ، فقط آدرسهای مبدأ و مقصد را درباره مسیر می‌داند. ICMP از آدرس IP مبدأ، برای ارسال پیغام خطا برای مبدأ بسته، استفاده می‌کند.

مهم : ICMP همواره پیغامهای خطا را برای مبدأ ارسال‌کننده پیغام می‌فرستد.

پنج نوع گزارش خطا داریم: ***مقصد، غیر قابل دسترسی است*** [1] ، ***مبدأ، کمتر بسته بفرست*** [2] ، ***مهلت مقرر سپری شده است*** [3] ، ***مشکلی در هدر وجود دارد*** [4] و ***مسیردهی این بسته باید اصلاح شود*** [5] . شکل ۹-۵ را ببینید .

مهم : نکات زیر درباره پیغامهای گزارش خطای ICMP مهم هستند :

1- Destination unreachable
2- Source quench
3- Time exceeded
4- Parameter problem
5- Redirection

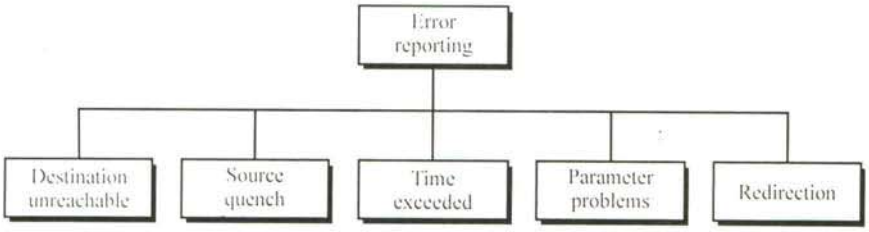
www.Abiri.ir

الف. اگر یک بسته IP حاوی یک گزارش خطای ICMP باشد و این بسته IP دچار خطا شود، هیچ پیغام خطای ICMP ای برای آن ارسال نخواهد شد.

ب. هیچ پیغام خطای ICMP برای قطعه‌ای از یک بسته IP که اولین قطعه نباشد، ارسال نمی‌شود.

ج. هیچ پیغام خطای ICMP برای یک بسته IP که حاوی آدرس **چندگیرنده**[1] باشد، ارسال نمی‌شود.

د. هیچ پیغام خطای ICMP برای بسته‌ای حاوی یک آدرس خاص نظیر 127.0.0.0 یا 0.0.0.0 ارسال نمی‌شود.



شکل ۹-۵ : پیغامهای گزارش خطا

دقت نمایید که همه پیغامهای خطا حاوی یک بخش دیتا هستند که شامل هدر بسته IP تولیدکننده خطا و نیز 8 بایت اول دیتای آن بسته، می‌شود. هدر بسته IP تولیدکننده خطا باید به گزارش اضافه شود تا مبدأ ارسال‌کننده آن بسته، بداند کدام بسته ارسالی او، خطا ایجاد کرده است. 8 بایت از دیتای بسته تولیدکننده خطا نیز به گزارش اضافه می‌شود زیرا همانطور که در فصلهای یازدهم و دوازدهم درباره پروتوکلهای UDP و TCP خواهیم دید، اولین 8 بایت دیتای بسته IP در چنین مواردی حاوی شماره پورتها (در مورد TCP و UDP) و شماره ترتیب (در مورد TCP) می‌باشد. این اطلاعات مورد نیاز است تا مبدأ دریافت‌کننده گزارش خطا بتواند به پروتوکلهای TCP و UDP درباره خطای رخ داده، گزارش دهد. ICMP یک گزارش خطا تولید می‌کند که داخل یک بسته IP کپسول‌بندی می‌شود (شکل ۹-۶ را ببینید).

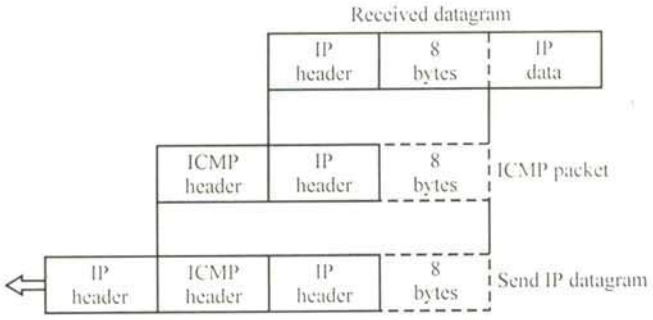
❊ پیغامِ مقصد، غیر قابل دسترسی است

وقتی که روتری نمی‌تواند بسته‌ای را مسیردهی کند و یا یک کامپیوتر نمی‌تواند بسته‌ای را به مقصد تحویل دهد، آن بسته دور انداخته می‌شود و روتر یا کامپیوتر،

1- multicast

www.Abiri.ir

یک ***پیغامِ مقصد، غیر قابل دسترسی است***[1] برای مبدأ ارسال‌کننده پیغام، ارسال می‌کند. شکل ۹-۷ فرمت این پیغام را نشان می‌دهد. فیلد Code برای این نوع از پیغام، علت غیرقابل دسترسی بودن مقصد را نشان می‌دهد.



شکل ۹-۶ : محتوای فیلد دیتا برای پیغامهای گزارش خطا

Type: 3	Code: 0 to 15	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

شکل ۹-۷ : فرمت پیغام مقصد، غیر قابل دسترسی است

Code 0 : شبکه مقصد، غیر قابل دسترسی است. دلیل آن ممکن است یک مشکل سخت‌افزاری باشد. فقط یک روتر می‌تواند این نوع پیغام خطا را تولید کند.

Code 1 : کامپیوتر مقصد، غیر قابل دسترسی است. دلیل آن ممکن است یک مشکل سخت‌افزاری باشد. فقط یک روتر می‌تواند این نوع پیغام خطا را تولید کند.

Code 2 : پروتوکل، غیر قابل دسترسی است. یک بسته IP می‌تواند دیتایی که از لایه‌های بالاتر آمده است مانند UDP ، TCP و OSPF را حمل کند. به عنوان مثال اگر کامپیوتر مقصد، یک بسته IP را دریافت کند که مثلاً باید به پروتوکل TCP تحویل داده شود، اما در حال حاضر پروتوکل TCP در حال اجرا نیست، پیغام خطای ICMP با Code برابر با 2 تولید می‌شود. این نوع پیغام فقط توسط کامپیوتر مقصد قابل تولید است.

1- destination unreachable message

www.Abiri.ir

Code 3 : پورت، غیر قابل دسترسی است. پروتوکل لایه کاربرد که بسته IP برای او ارسال شده است، در حال حاضر کار نمی‌کند.

Code 4 : قطعه قطعه سازی مورد نیاز است اما بیت DF برابر با 1 است. به عبارت دیگر، فرستنده بسته خواسته است که این بسته، قطعه قطعه نشود اما بدون قطعه قطعه کردن بسته، امکان مسیردهی آن نیست.

Code 5 : نمی‌توان مسیردهی درخواستی از طرف مبدأ را انجام داد. به عبارت دیگر بسته نمی‌تواند از یک یا چند تا از روترهایی که آدرس آنها در گزینه « مسیردهی قطعی از مبدأ» و یا گزینه « مسیردهی نه چندان قطعی از مبدأ » ثبت شده است، عبور کند.

Code 6 : شبکه مقصد، ناشناخته است. این حالت با Code 0 متفاوت است. در Code 0 روتر می‌داند که شبکه مقصد وجود دارد اما در حال حاضر آن شبکه، غیر قابل دسترسی است. برای Code 6 ، روتر هیچگونه اطلاعی از شبکه مقصد ندارد.

Code 7 : کامپیوتر مقصد ناشناخته است. این حالت با Code 1 متفاوت است. در Code 1 روتر می‌داند که کامپیوتر مقصد وجود دارد اما در حال حاضر آن کامپیوتر، غیر قابل دسترسی است. برای Code 7 ، روتر از وجود شبکه مقصد بی‌خبر است.

Code 8 : این کد، منسوخ شده است.

Code 9 : مدیر شبکه، مخابره با شبکه مقصد را ممنوع کرده است.

Code 10 : مدیر شبکه، مخابره با کامپیوتر مقصد را ممنوع کرده است.

Code 11 : شبکه مقصد برای نوع سرویس درخواستی، غیر قابل دسترسی است. این حالت با Code 0 تفاوت دارد. در اینجا اگر مبدأ، یک سرویس موجود را درخواست کرده بود، روتر می‌توانست بسته را مسیردهی نماید.

Code 12 : برای نوع سرویس داده شده، کامپیوتر غیر قابل دسترسی است. این حالت با Code 1 تفاوت دارد. در اینجا اگر مبدأ، یک سرویس موجود را درخواست کرده بود، روتر می‌توانست بسته را مسیردهی کند.

Code 13 : کامپیوتر مقصد، غیر قابل دسترسی است زیرا که مدیر شبکه، فیلتری در سر راه رسیدن به آن قرار داده است.

Code 14 : کامپیوتر مقصد، غیر قابل دسترسی است زیرا که عدد قید شده برای ***ارجحیت***[1] در هدر بسته IP ، آنقدر بزرگ نیست که مجاز به دسترسی به کامپیوتر مقصد مورد نظر باشد. این پیغام توسط روتر ارسال می‌شود.

1- precedence

www.Abiri.ir

Code 15 : کامپیوتر مقصد، غیر قابل دسترسی است زیرا که ارجحیت قید شده در هدر بسته IP کمتر از حداقل ارجحیت تعیین شده توسط مدیر شبکه برای دسترسی به شبکه مقصد است.

دقت نمائید که یک روتر یا کامپیوتر مقصد می‌تواند پیغامهای « مقصد، غیر قابل دسترسی است» را تولید کند. پیغامهای Code 2 و Code 3 تنها توسط کامپیوتر مقصد قابل تولید هستند و پیغامهای سایر کدها را فقط روترها می‌توانند تولید کنند.

مهم : پیغامهای « مقصد، غیر قابل دسترسی است » با Codeهای 2 یا 3 را فقط می‌توان توسط کامپیوتر مقصد تولید کرد. سایر Codeها، فقط توسط روترها قابل تولید هستند.

توجه به این نکته لازم است که حتی اگر یک روتر، پیغام خطایی از نوع « مقصد، غیر قابل دسترسی است » را گزارش نکند، الزاماً به این معنی نیست که بسته IP ، تحویل داده شده است. به عنوان مثال اگر بسته‌ای در میان یک شبکه اترنت جابه‌جا شود، از آنجا که اترنت فاقد مکانیزم ارسال ACK (تأییدیه دریافت) است، هیچ راهی وجود ندارد که روتر دریابد آیا بسته ارسالی، به کامپیوتر مقصد یا روتر بعدی، تحویل داده شده است یا خیر.

مهم : یک روتر نمی‌تواند تمام مسائلی که مانع از تحویل بسته IP به مقصد می‌شود را گزارش نماید.

* پیغامِ مبدأ، کمتر بسته بفرست

پروتوکل IP یک پروتوکل فاقد اتصال است. به این معنی که کامپیوتر مبدأ (به عنوان ارسال‌کننده بسته IP)، روترها (به عنوان هدایت کننده بسته IP) و کامپیوتر مقصد (به عنوان پردازش‌کننده بسته IP) تحت پروتوکل IP ، از وضع هم با خبر نمی‌شوند و یکی از پیامدهای آن، نداشتن کنترل فلوست. IP فاقد یک متد کنترل فلوی داخلی است و این ویژگی می‌تواند مشکل عمده‌ای به نام ***ازدحام***[1] را در شبکه ایجاد کند. کامپیوتر مبدأ که بسته‌ها را ارسال می‌کند، هرگز نمی‌داند آیا روترها یا کامپیوتر مقصد از بسته‌های IP لبریز شده‌اند یا خیر. کامپیوتر مبدأ هرگز خبر ندارد که آیا بسته‌ها را سریع‌تر از قابلیت هدایت آنها توسط روترها و یا قابلیت پردازش آنها توسط کامپیوتر مقصد، تولید می‌کند یا خیر.

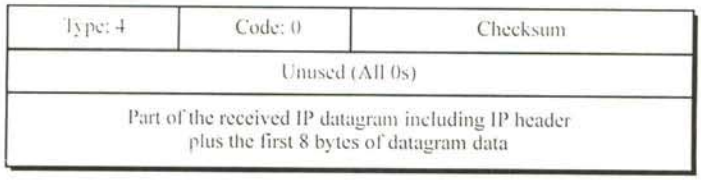
مهم : پروتوکل IP فاقد مکانیزم کنترل فلوست.

1- Congestion

www.Abiri.ir

فقدان کنترل فلو در پروتوکل IP می‌تواند در روترها یا در کامپیوتر مقصد تولید ازدحام کند. یک روتر، صف (بافر) محدودی برای هدایت بسته‌های دریافتی و یک کامپیوتر، صف (بافر) محدودی برای پردازش بسته‌های دریافتی دارد. اگر سرعت دریافت بسته‌ها، بیش از سرعت هدایت آنها توسط روترها و یا سرعت پردازش آنها توسط کامپیوتر مقصد باشد، صف از بسته‌ها لبریز می‌شود. در این حالت، تنها کاری که روتر یا کامپیوتر مقصد می‌تواند انجام دهد این است که برخی از بسته‌های IP دریافتی را دور بیندازد.

پیغامِ مبدأ، کمتر بسته بفرست[1] به منظور دادن قابلیت کنترل فلو به IP ، طراحی شده است. وقتی که یک روتر یا کامپیوتر، بسته‌ای را در اثر ازدحام دور می‌اندازد، یک پیغام خطای ICMP از این نوع، برای فرستنده بسته IP ارسال می‌کند. هدف از ارسال این پیغام، دو چیز است: اول آنکه به مبدأ ارسال کننده بسته IP خبر دهد که آن بسته IP دور انداخته شده است. دوم آنکه به مبدأ خبر دهد که در نقطه‌ای از مسیر رسیدن بسته به مقصد، ازدحامی رخ داده است و لذا مبدأ ارسال‌کننده بسته باید سرعت ارسال بسته‌هایش را پایین آورد. فرمت این پیغام در شکل ۹-۸ نشان داده شده است.



شکل ۹-۸ : فرمت پیغامِ مبدأ، کمتر بسته بفرست

مهم : پیغام « مبدأ، کمتر بسته بفرست »، به مبدأ خبر می‌دهد که در اثر وقوع ازدحام در نقطه‌ای از مسیر رسیدن بسته به مقصد یا در کامپیوتر مقصد، بسته مربوطه دور انداخته شده است. تا زمانی که ازدحام از بین برود، مبدأ باید سرعت ارسال بسته‌هایش را کم کند.

مواردی وجود دارند که توضیح بیشتری را می‌طلبند. اول آنکه روتر یا کامپیوتر مقصد که دچار ازدحام شده است، به ازای هر بسته IP ای که در اثر ازدحام، دور انداخته می‌شود، چنین پیغام خطایی را ارسال می‌کند. دوم آنکه هیچ مکانیزمی برای خبر دادن به مبدأ وجود ندارد که ازدحام برطرف شده است و لذا او می‌تواند با همان سرعت قبلی، بسته‌ها را ارسال کند. کاری که مبدأ می‌کند این است که آنقدر سرعت ارسال بسته‌ها را کم می‌کند تا دیگر پیغام

1- Source quench message

www.Abiri.ir

خطای « مبدأ، کمتر بسته بفرست » را دریافت نکند. سوم آنکه هم مخابره یک به یک و هم مخابره چند به یک می‌تواند ازدحام ایجاد کند. در مخابره یک به یک، یک کامپیوتر پرسرعت ممکن است بسته ها را با سرعتی بیش از توان هدایت روترها و یا توان پردازش کامپیوتر مقصد ارسال نماید. در این حالت، پیغامهای خطای « مبدأ، کمتر بسته بفرست » مفید خواهند بود، زیرا از مبدأ می‌خواهند که سرعت ارسال بسته‌هایش را کم کند، اما در یک مخابره چند به یک، چندین مبدأ، بسته‌هایی تولید می‌کنند که روتر یا کامپیوتر مقصد باید روی آنها کار کنند. در این حالت ممکن است سرعت ارسال بسته‌ها توسط یک مبدأ، کم و سرعت ارسال بسته‌ها توسط مبدأ دیگر، زیاد باشد. در این مورد ممکن است پیغام خطای « مبدأ، کمتر بسته بفرست » خیلی مفید نباشد زیرا روتر یا کامپیوتر مقصد هیچ اطلاعی ندارد که کدامیک از این کامپیوترها، مسئول ایجاد ازدحام است. او ممکن است بسته‌های کامپیوتری را دور بیندازد که با سرعت خیلی کمی در حال ارسال بسته‌هاست و در عین حال، بسته‌های کامپیوتری که با سرعت زیاد در حال ارسال بسته‌هاست، دور انداخته نشوند.

مهم : به ازای هر بسته‌ای که در اثر ازدحام دور انداخته می‌شود، یک پیغام « مبدأ، کمتر بسته بفرست » برای مبدأ ارسال‌کننده بسته فرستاده می‌شود.

❊ پیغام مهلت مقرر سپری شده است

در دو حالت، ***پیغام مهلت مقرر سپری شده است***[1] تولید می‌شود:

الف. همانطور که در فصل ششم دیدیم، روترها از جدول مسیردهی برای پیدا کردن هاپ بعدی (روتر بعدی) که باید بسته را دریافت کند، استفاده می‌نمایند. اگر یک یا چند تا از جداول مسیردهی دچار خطا شده باشند و یک بسته داخل لوپ بیفتد، آن بسته بدون آنکه به مقصد نهایی برود، صرفاً از یک روتر به روتر دیگر می‌رود. همانطور که در فصل هشتم دیدیم، هر بسته IP حاوی یک فیلد ***طول عمر***[2] است که این وضعیت را کنترل می‌کند. وقتی که بسته به روتری می‌رسد، یک واحد از این فیلد کم می‌شود. اگر مقدار جدید TTL برابر با صفر باشد، بسته دور انداخته می‌شود و یک پیغام « مهلت مقرر سپری شده است » برای مبدأ ارسال‌کننده بسته، فرستاده می‌شود.

مهم : اگر روتر با کاهش یک واحد از مقدار TTL ، به TTL برابر با صفر برسد، بسته IP را دور می‌اندازد و پیغام « مهلت مقرر سپری شده است » را برای مبدأ، می‌فرستد.

1- Time exceeded message
2- Time to Live : TTL

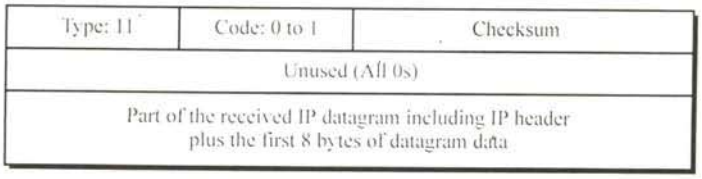
www.Abiri.ir

ب. دومین حالتی که این پیغام خطا ارسال می‌شود زمانی است که تمام قطعات سازنده یک بسته IP در یک مدت زمان از پیش تعیین شده، به مقصد نرسیده‌اند. وقتی که اولین قطعه از راه می‌رسد، کامپیوتر مقصد یک تایمر را راه‌اندازی می‌کند. اگر پس از انقضای مهلت مقرر، همه قطعات بسته از راه نرسیده باشند، مقصد، سایر قطعات دریافتی را نیز دور می‌اندازد و یک پیغام خطای « مهلت مقرر سپری شده است » را برای مبدأ ارسال‌کننده بسته اصلی، می‌فرستد.

مهم : وقتی که مقصد نهایی، تمام قطعات یک بسته IP را در یک مدت زمان از قبل تعیین شده، دریافت نکند، قطعات دریافتی را دور می‌اندازد و یک پیغام « مهلت مقرر سپری شده است » را برای مبدأ اولیه می‌فرستد.

شکل ۹-۹ فرمت پیغام « مهلت مقرر سپری شده است » را نشان می‌دهد. اگر روتر بسته را به دلیل صفر شدن TTL آن دور بیندازد، مقدار Code برابر با 0 است. Code 1 زمانی تولید می‌شود که فقط برخی از قطعات بسته اولیه دریافت شده‌اند و سایر قطعات در مدت زمان تعیین شده ، دریافت نشده‌اند.

مهم : در پیغام « مهلت مقرر سپری شده است » روترها از Code 0 برای نشان دادن صفر شدن TTL استفاده می‌کنند. کامپیوتر مقصد از Code 1 برای نشان دادن عدم دریافت همه قطعات بسته در خلال یک مدت زمان از قبل تعیین شده، استفاده می‌کند.



شکل ۹-۹ : فرمت پیغام مهلت مقرر سپری شده است

* پیغام مشکلی در هدر وجود دارد

همچنان که بسته IP در شبکه جابه‌جا می‌شود، هرگونه ابهامی در بخش هدر بسته IP می‌تواند مشکلات جدی ایجاد کند. اگر روتر یا کامپیوتر نهایی در مورد یکی از فیلدهای هدر

www.Abiri.ir

بسته IP دچار ابهامی شود، بسته IP را دور می‌اندازد و یک ***پیغام مشکلی در هدر وجود دارد***[1] برای مبدأ ارسال‌کننده پیغام می‌فرستد.

مهم : یک روتر یا کامپیوتر مقصد ممکن است پیغام خطای « مشکلی در هدر وجود دارد » را تولید کند.

شکل ۹-۱۰ فرمت پیغام « مشکلی در هدر وجود دارد » را نشان می‌دهد. مقدار فیلد Code در این حالت، علت دور انداختن بسته IP را نشان می‌دهد:

Code 0 : خطا یا ابهامی در یکی از فیلدهای هدر وجود دارد. در این حالت، فیلد Pointer به بایت حاوی خطا در هدر بسته IP اشاره می‌کند. به عنوان مثال، اگر این مقدار برابر با صفر باشد، در این صورت اولین بایت هدر IP ، مقدار معتبری نیست.

Code 1 : اطلاعات یک گزینه در هدر کامل نیست. در این حالت از pointer استفاده نمی‌شود.

Type: 12	Code: 0 to 1	Checksum
Pointer	Unused (All 0s)	
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

شکل ۹-۱۰ : فرمت پیغام مشکلی در هدر وجود دارد

*** پیغام مسیردهی این بسته باید اصلاح شود**

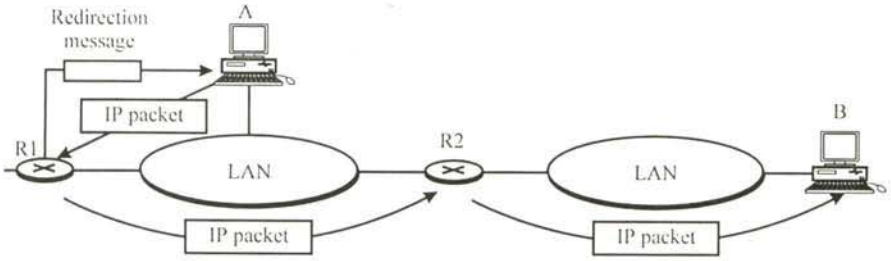
وقتی که روتر می‌خواهد بسته‌ای را برای شبکه دیگری بفرستد، باید آدرس IP روتر مناسب بعدی را بداند. اگر فرستنده بسته، یک کامپیوتر باشد نیز همین مطلب صحیح است. هم روتر و هم کامپیوتر باید جدول مسیردهی داشته باشند تا به کمک آن، آدرس روتر یا روتر بعدی را بدست آورند. همانطور که در فصل چهاردهم خواهیم دید، روترها در فرایند به روز رسانی جدول مسیردهی، شرکت می‌جویند و لذا محتوای این جدول مکرراً به روز رسانی می‌شود. عمل مسیردهی به صورت دینامیک صورت می‌گیرد.

اما برای بالا بردن کارآیی، کامپیوترها در امر به روز رسانی جداول مسیردهی شرکت نمی‌کنند زیرا که تعداد کامپیوترهای موجود در اینترنت، بسیار بیشتر از تعداد روترها هستند.

1- Parameter Problem message

www.Abiri.ir

به روز رسانی جداول مسیردهی کامپیوترها به صورت دینامیک، ترافیک غیر قابل قبولی را ایجاد می‌کند. به همین دلیل معمولاً کامپیوترها از مسیردهی استاتیک استفاده می‌کنند. وقتی که کامپیوتری روشن می‌شود، المانهای محدودی داخل جدول مسیردهی آن است. معمولاً تنها المان موجود در جدول مسیردهی روتر، آدرس IP ***روتر پیش‌فرض***[1] است. حال اگر دو روتر به شبکه LAN متصل شده باشند و یکی به عنوان روتر پیش‌فرض برای کامپیوتر تعریف شده باشد، کامپیوتر برای رسیدن بسته به مقصد، آن را تحویل روتر پیش‌فرض می‌دهد اما ممکن است روتر دوم، مناسبتر باشد. در این حالت وقتی که روتر پیش‌فرض، بسته را دریافت می‌کند، باید آن را به دست روتر دوم بدهد و برای آنکه از آن به بعد، بسته‌هایی که برای آن مقصد ارسال می‌شوند، مستقیماً به دست روتر دوم داده شوند، یک پیغام « مسیردهی این بسته باید اصلاح شود » را برای کامپیوتر ارسال می‌کند. این مفهوم در شکل ۹-۱۱ نشان داده شده است. کامپیوتر A می‌خواهد بسته‌ای را برای کامپیوتر B ارسال کند. بدیهی است که روتر R2 ، بهترین انتخاب برای مسیردهی است اما روتر R1 به عنوان روتر پیش‌فرض برای کامپیوتر A تعریف شده است بنابراین بسته را برای روتر R1 ارسال می‌کند. روتر R1 پس از مراجعه به جدول مسیردهی خود درمی‌یابد که بسته می‌بایست به روتر R2 تحویل داده شود لذا او بسته را برای R2 می‌فرستد و در همان لحظه، یک پیغام « مسیردهی این بسته باید اصلاح شود » برای کامپیوتر A فرستاده می‌شود. حال کامپیوتر A می‌تواند جدول مسیردهی خود را اصلاح کند.



شکل ۹-۱۱ : مفهوم پیغام مسیردهی این بسته باید اصلاح شود

مهم : یک کامپیوتر معمولاً با یک جدول مسیردهی کوچک شروع می‌کند و به تدریج به محتوای آن می‌افزاید و آن را اصلاح می‌کند. یکی از ابزارهای انجام این کار، استفاده از پیغام « مسیردهی این بسته باید اصلاح شود » می‌باشد.

1- Default Gateway

www.Abiri.ir

فرمت پیغام « مسیردهی این بسته باید اصلاح شود » در شکل ۹-۱۲ آمده است. دقت نمائید که آدرس IP روتر مناسب در سطر دوم آمده است.

Type: 5	Code: 0 to 3	Checksum
IP address of the target router		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

شکل ۹-۱۲ : فرمت پیغام مسیردهی این بسته باید اصلاح شود

گرچه این پیغام، یک پیغام گزارش خطا در نظر گرفته شده است، اما با سایر پیغامهای خطا تفاوت دارد. روتر در این حالت، بسته دریافتی را دور نمی‌اندازد بلکه آن را برای روتر مناسب می‌فرستد. فیلد Code در این پیغام می‌تواند مقادیر زیر را به خود بگیرد:

Code 0 : وقتی که شبکه مشخص شده باشد.

Code 1 : وقتی که کامپیوتر مشخص شده باشد.

Code 2 : وقتی که شبکه و نوع سرویس مشخص شده باشند.

Code 3 : وقتی که کامپیوتر و نوع سرویس مشخص شده باشند.

مهم : پیغام « مسیردهی این بسته باید اصلاح شود » از یک روتر، برای کامپیوتری ارسال می‌شود که به همان شبکه روتر، متصل است.

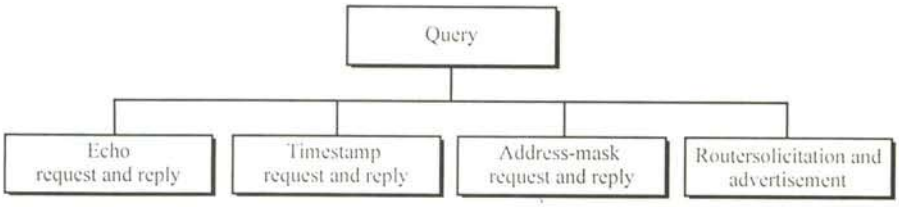
پیغامهای ICMP از نوع درخواست و پاسخ

ICMP می‌تواند علاوه بر گزارش خطا، یک سرویس درخواست و پاسخ در سطح لایه شبکه نیز باشد. این کار توسط پیغامهای درخواست و پاسخ انجام می‌شود که جمعاً چهار نوع پیغام مختلف است و در شکل ۹-۱۳ نشان داده شده‌اند. در این نوع از پیغامهای ICMP ، مبدأ، ***درخواستی***[1] را مطرح می‌کند که به فرمت مشخصی از سوی مقصد، ***پاسخ***[2] داده می‌شود. دقت نمائید که در ابتدا دو نوع دیگر نیز از این پیغامها، وجود داشت (***پیغام درخواست اطلاعات***[3] و ***پیغام پاسخ اطلاعات***[4]) اما امروزه آنها منسوخ شده‌اند. آنها به این منظور

1- Request
2- Reply
3- information request message
4- information reply message

www.Abiri.ir

طراحی شده بودند که یک کامپیوتر بتواند به هنگام روشن شدن، آدرس IP خودش را بگیرد. اینک این عمل توسط RARP (فصل هفتم) و BOOTP (فصل شانزدهم) انجام می‌شود.



شکل ۹-۱۳ : پیغامهای درخواست و پاسخ ICMP

* پیغام درخواست و پاسخ انعکاس

پیغامهای ***درخواست انعکاس***[1] و ***پاسخ انعکاس***[2] به منظور تست اتصالات در شبکه طراحی شده‌اند. مدیران شبکه و کاربران به کمک این یک زوج پیغام می‌توانند به مشکلات شبکه پی ببرند. پیغامهای درخواست و پاسخ Echo نشان می‌دهند که آیا ماشین (کامپیوتر یا روتر) ارسال‌کننده درخواست Echo ، می‌تواند مخابره‌ای با ماشینی داشته باشد که ارسال پاسخ Echo را از او مطالبه کرده است یا خیر.

یک کامپیوتر یا روتر می‌تواند یک پیغام درخواست Echo برای کامپیوتر یا روتر دیگری بفرستد. کامپیوتر یا روتری که پیغام درخواست Echo را دریافت می‌کند، یک پیغام پاسخ Echo را تولید می‌کند و آن را برای فرستنده اصلی پیغام، برمی‌گرداند.

مهم : یک کامپیوتر یا یک روتر می‌تواند یک پیغام درخواست Echo ارسال کند. یک پیغام پاسخ Echo توسط کامپیوتر یا روتری که پیغام درخواست Echo را دریافت می‌کند، ارسال می‌شود.

پیغامهای درخواست و پاسخ Echo را می‌توان برای تعیین آنکه آیا بین دو وسیله در سطح IP ، مخابره‌ای وجود دارد یا خیر، به کار برد. از آنجا که پیغامهای ICMP در بسته‌های IP کپسول‌بندی می‌شوند، وقتی که ماشین ارسال کننده درخواست Echo ، پاسخ Echo را دریافت می‌کند، اثباتی بر این مطلب است که پروتوکلهای IP این دو ماشین می‌توانند مخابره‌ای با هم داشته باشند.

1- Echo request
2- Echo reply

www.Abiri.ir

مهم : مدیران شبکه از پیغامهای درخواست و پاسخ Echo استفاده می‌کنند تا صحت عملکرد پروتوکل IP را تست کنند.

یک کامپیوتر نیز از پیغامهای درخواست و پاسخ Echo استفاده می‌کند تا ببیند آیا کامپیوتر دیگری در سطح لایه شبکه، قابل دسترسی است یا خیر. برنامه ***ping***[1] (بسته‌ای که در اینترنت، کورمال کورمال حرکت می‌کند) این کار را انجام می‌دهد. امروزه اکثر سیستم عاملها نسخه‌ای از برنامه ping را پیاده‌سازی کرده‌اند که می‌تواند به جای صرفاً یک پیغام درخواست و پاسخ Echo، دنباله‌ای از پیغامهای درخواست و پاسخ Echo را تولید کند و به این ترتیب، یک اطلاعات آماری درباره عملکرد شبکه به کاربر بدهد. در انتهای فصل، نحوه استفاده از این برنامه را خواهیم دید.

مهم : می‌توان از پیغامهای درخواست و پاسخ Echo برای تست قابل دسترس بودن یک کامپیوتر استفاده کرد. این کار غالباً با صدا زدن فرمان ping انجام می‌شود.

پیغام درخواست Echo به همراه پیغام پاسخ Echo می‌تواند صحت کار وسیله‌ای در سطح لایه شبکه را تست کنند. به این منظور برای وسیله‌ای که باید تست شود، یک پیغام درخواست Echo ارسال می‌شود. بخش دیتای پیغام درخواست Echo حاوی یک دیتای اتفاقی است که عیناً برای ارسال کننده درخواست Echo برگردانده می‌شود. شکل ۹-۱۴ فرمت پیغام درخواست و پاسخ Echo را نشان می‌دهد. تعریف خاصی برای فیلدهای ***شناسه***[2] و ***شماره ترتیب***[3] در پروتوکل ICMP نیامده است و لذا فرستنده می‌تواند به طور دلخواه از آنها استفاده کند. فرستنده هر مقداری که در این فیلدها قرار دهد، گیرنده عیناً آن را برمی‌گرداند. identifier غالباً همان ***شناسه برنامه***[4] است.

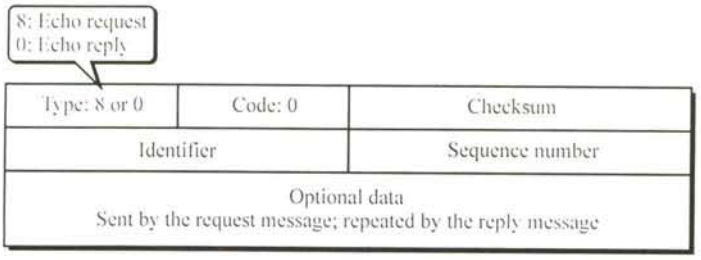
* پیغام درخواست و پاسخ برچسب زمانی

دو ماشین (کامپیوتر یا روتر) می‌توانند از پیغام ***درخواست برچسب زمانی***[5] و ***پاسخ برچسب زمانی***[6] استفاده کنند تا زمان لازم برای رفت و برگشت یک بسته IP بین آنها را

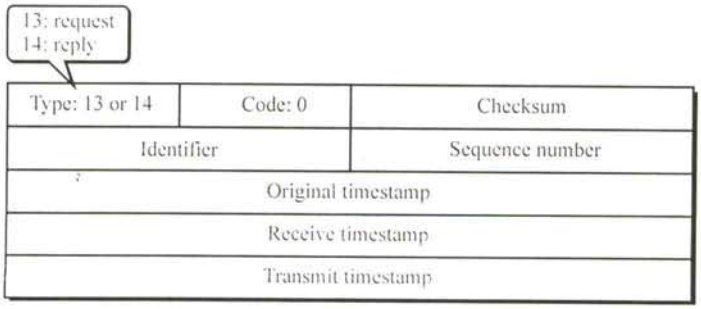
1- packet internet groper : ping
2- identifier
3- Sequence number
4- Process ID
5- Timestamp request
6- Timestamp reply

www.Abiri.ir

تعیین نماید. همچنین می‌توان از آنها برای سنکرون کردن کلاک دو ماشین استفاده کرد. فرمت این دو پیغام در شکل ۹-۱۵ آمده است.



شکل ۹-۱۴ : پیغامهای درخواست و پاسخ Echo



شکل ۹-۱۵ : فرمت پیغام درخواست و پاسخ Timestamp

هر یک از سه فیلد Timestamp به طول 32 بیت می‌باشند. می‌توان در هر فیلد، عددی گذاشت که نمایانگر زمان سپری شده بر حسب میلی ثانیه از نیمه شب به ***وقت گرین‌ویچ***[1] باشد. دقت نمائید که یک عدد 32 بیتی می‌تواند از 0 تا 4294967295 باشد اما مقدار Timestamp در این حالت نمی‌تواند از 24x60x60x1000 = 86400000 میلی‌ثانیه بیشتر شود.

مبدأ، یک پیغام درخواست Timestamp تولید می‌کند. زمانی که فرستنده می‌خواهد پیغام را ارسال کند، محتوای کلاک خود را در فیلد ***برچسب زمانی اولیه***[2] می‌نویسد و فیلد ***برچسب زمانی دریافت***[3] را با صفر پر می‌کند.

1- Greenwich Mean Time or Universal Time
2- Original Timestamp
3- Receive Timestamp

www.Abiri.ir

مقصد، پیغام پاسخ Timestamp را تولید می‌کند. مقصد برای این کار، مقدار نوشته شده در فیلد Original Timestamp را از پیغام دریافتی استخراج می‌کند و آن را در فیلدی به همین نام در پیغام پاسخ Timestamp می‌نویسد. همچنین از کلاک خود درمی‌یابد که چه زمانی پیغام درخواست Timestamp را دریافت کرده است و مقدار آن را در فیلد Receive Timestamp می‌نویسد. سرانجام زمانی که می‌خواهد پیغام پاسخ Timestamp را ارسال نماید، فیلد Transmit Timestamp (برچسب زمانی ارسال) را با مقداری که کلاک نشان می‌دهد، پر می‌کند. زمانی که این پیغام به دست فرستنده اصلی می‌رسد، او به کلاک خود نگاه می‌کند و مقدار Return Time (زمان برگشت) را ثبت می‌کند. می‌توان از پیغامهای درخواست و پاسخ Timestamp برای محاسبه زمان رفت و برگشت یک بسته IP از یک مبدأ به یک مقصد، استفاده کرد. فرمول محاسبه، چنین است:

زمان ارسال = Original Timestamp–Receive Timestamp
زمان دریافت = Return Time - Transmit Timestamp
زمان برگشت + زمان رفت = زمان رفت و برگشت

دقت نمائید که فقط به شرط سنکرون بودن کلاکهای ماشینهای فرستنده و گیرنده، محاسبات زمان رفت و برگشت، صحیح خواهند بود. با این حال محاسبه مربوط به زمان رفت و برگشت، حتی اگر دو کلاک سنکرون نباشند نیز صحیح است زیرا که هر کلاک، دو مرتبه در محاسبه زمان رفت و برگشت وارد می‌شود و بنابراین هر تفاوتی در سنکرونیزاسیون کلاکها، حذف خواهد شد.

مهم : حتی اگر کلاکهای ماشین فرستنده و گیرنده با هم سنکرون نباشند، می‌توان با استفاده از پیغامهای درخواست و پاسخ Timestamp ، زمان رفت و برگشت را محاسبه کرد.

به عنوان مثال با دانستن اطلاعات زیر:

Original Timestamp = 46
Receive Timestamp = 59
Transmit Timestamp = 60
Return Time = 67

www.Abiri.ir

می‌توان به روش زیر، مقدار زمان رفت و برگشت را برابر با 20 میلی‌ثانیه محاسبه کرد:

زمان رفت = 59 - 46 = 13 milliseconds
زمان برگشت = 67 - 60 = 7 milliseconds
زمان رفت و برگشت = 13 + 7 = 20 milliseconds

با داشتن ***زمان واقعی رفت یا برگشت***[1]، می‌توان با استفاده از فرمول زیر، تفاوت زمانی بین کلاکهای دو ماشین را محاسبه کرد :

Receive Timestamp − (Original Timestamp + زمان واقعی رفت یا برگشت)

زمان رفت واقعی را می‌توان با تقسیم کردن زمان رفت و برگشت بر دو (اگر مطمئن باشیم که زمان واقعی رفت با زمان واقعی برگشت، مساوی است) و یا به کمک سایر روشها، محاسبه کرد. به عنوان مثال، می‌توان گفت که دو کلاک مثال قبل به اندازه 3 میلی‌ثانیه اختلاف زمانی دارند زیرا که :

اختلاف زمانی کلاکها = 59 - (46 + 10) = 3

مهم : می‌توان از پیغامهای درخواست و پاسخ Timestamp برای سنکرون کردن کلاک دو ماشین استفاده کرد مشروط بر آنکه زمان واقعی رفت (یا برگشت) مشخص باشد.

* درخواست و پاسخ مَسک

یک کامپیوتر ممکن است آدرس IP خود را بداند اما ممکن است مسک متناظر با آن را نداند . به عنوان مثال ممکن است کامپیوتر بداند که آدرس IP آن برابر با 159. 31.17.24 است اما ممکن است نداندکه مسک متناظر با آن برابر با 24/ است.

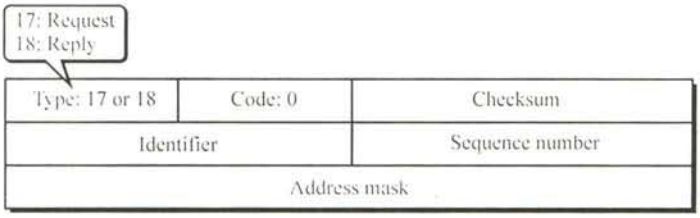
کامپیوتر برای بدست آوردن مسک خود، یک پیغام ***درخواست مسک***[2] را برای روتری روی LAN ارسال می‌کند. اگر کامپیوتر آدرس روتر را بداند، درخواست را مستقیماً برای روتر می‌فرستد. اگر نداند، آن را به صورت پخش همگانی ارسال می‌کند. روتری که این درخواست را دریافت می‌کند با یک پیغام ***پاسخ مسک***[3] جواب می‌دهد و به این ترتیب مسک لازم را در

1- one way time
2- Address mask request
3- Address mask reply

www.Abiri.ir

اختیار کامپیوتر قرار می‌دهد. می‌توان این مسک را به آدرس IP اعمال کرد تا آدرس زیرشبکه بدست آید.

فرمت پیغامهای درخواست و پاسخ مسک در شکل ۹-۱۶ نشان داده شده است. فیلد ***مسک آدرس***[1] در پیغام درخواست با صفر پر می‌شود. وقتی که روتر پیغام پاسخ مسک را برای کامپیوتر برمی‌گرداند، این فیلد حاوی مسک واقعی خواهد بود (1 ها در قسمت netid و 0ها در قسمت hostid).



شکل ۹-۱۶ : فرمت پیغام درخواست و پاسخ مسک

کامپیوترهایی که فاقد هارد دیسک هستند، در زمان روشن شدن می‌توانند از این پیغام استفاده کنند. وقتی که یک کامپیوتر فاقد هارد دیسک، برای اولین بار روشن می‌شود، ممکن است با استفاده از پروتوکل RARP ، آدرس IP خود را سؤال کند (فصل هفتم را ببینید). بعد از دریافت آدرس IP خود، ممکن است از پیغام درخواست و پاسخ مسک استفاده کند تا بفهمد که کدام قسمت از آدرس، بخش زیرشبکه را تعریف می‌کند.

روش دیگر برای گرفتن اطلاعات مربوط به مسک، استفاده از پروتوکل BOOTP است و این مطلب را در فصل شانزدهم خواهیم دید.

* پیغام درخواست از روتر و پیغام اطلاع‌رسانی روتر

همانطور که در بخش پیغام « مسیردهی این بسته باید اصلاح شود » گفتیم، کامپیوتری که می‌خواهد دیتایی را برای کامپیوتر دیگری ارسال کند، باید آدرس روترهای متصل به شبکه خود را بداند. همچنین کامپیوتر باید بداند که آیا روترها در حال کار هستند یا خیر. در این حالت، پیغامهای ***درخواست از روتر***[2] و ***اطلاع‌رسانی روتر***[3] می‌توانند کمک کنند. یک کامپیوتر می‌تواند یک پیغام « درخواست از روتر» را به صورت پخش همگانی (یا

1- address-mask
2- router solicitation
3- router advertisement

www.Abiri.ir

چند گیرنده) ارسال نماید. روتر یا روترهایی که این پیغام را دریافت می‌کنند، اطلاعات خود درباره روترهای متصل به شبکه کامپیوتر را توسط پیغامهای « اطلاع‌رسانی روتر» ارسال می‌کنند. حتی اگر هیچ کامپیوتری پیغام « درخواست از روتر» را ارسال نکرده باشد، خود روترها می‌توانند به طور پریودیک پیغامهای « اطلاع‌رسانی روتر » را ارسال کنند. دقت نمائید وقتی که روتری یک پیغام « اطلاع رسانی روتر » را ارسال می‌کند، نه تنها حضور خود را خبر می‌دهد بلکه همچنین حضور همه روترهای متصل به شبکه کامپیوتر، که آن روتر از وجود آنها با خبر است را نیز اعلام می‌نماید. شکل ۹-۱۷ فرمت پیغام « درخواست از روتر» را نشان می‌دهد.

Type: 10	Code: 0	Checksum
Identifier		Sequence number

شکل ۹-۱۷ : فرمت پیغام درخواست از روتر

شکل ۹-۱۸ فرمت پیغام « اطلاع رسانی روتر » را نشان می‌دهد. فیلد ***طول عمر***[1] مدت زمانی را نشان می‌دهد که می‌توان روی صحت اطلاعات موجود در پیغام « اطلاع رسانی روتر » حساب کرد. در این پیغام، دو فیلد برای هر روتری که درباره او خبر داده می‌شود، وجود دارد : ***آدرس روتر***[2] و ***سطح ارجحیت آدرس***[3]. « سطح ارجحیت آدرس » میزان ارجحیت روتر مربوطه در انتخاب شدن به عنوان روتر پیش‌فرض را نشان می‌دهد. اگر مقدار آن برابر با 0 باشد، روتر مربوط به آن المان، به عنوان روتر پیش‌فرض در نظر گرفته می‌شود. اگر مقدار آن برابر با 80000000_{16} باشد، آن روتر را هرگز نباید به عنوان روتر پیش‌فرض انتخاب کرد.

Checksum در ICMP

در فصل هشتم، مفهوم و ایده Checksum را یاد گرفتیم. در ICMP ، مقدار Checksum روی کل پیغام (هدر و دیتا) محاسبه می‌شود.

1- lifetime
2- router address
3- address preference level

www.Abiri.ir

Type: 9	Code: 0	Checksum
Number of addresses	Address entry size	Lifetime
Router address 1		
Address preference 1		
Router address 2		
Address preference 2		
• • •		

شکل ۹-۱۸ : فرمت پیغام اطلاع رسانی روتر

❊ محاسبه Checksum در فرستنده

فرستنده با استفاده از ریاضیات مکمل یک، قدمهای زیر را برمی‌دارد:

۱- فیلد Checksum برابر با صفر قرار داده می‌شود.

۲- جمع همه بخشهای 16 بیتی (هدر و دیتا) محاسبه می‌شود.

۳- مکمل حاصل‌جمع محاسبه می‌شود.

۴- مقدار بدست آمده در بند ۳، در فیلد Checksum نوشته می‌شود.

❊ تست Checksum در گیرنده

گیرنده با استفاده از ریاضیات مکمل یک قدمهای زیر را برمی‌دارد :

۱- مجموع همه بخشهای 16 بیتی (هدر و دیتا) محاسبه می‌شود. دقت نمائید که مقدار Checksum نیز در این محاسبه وارد می‌شود.

۲- مکمل مجموع، محاسبه می‌شود.

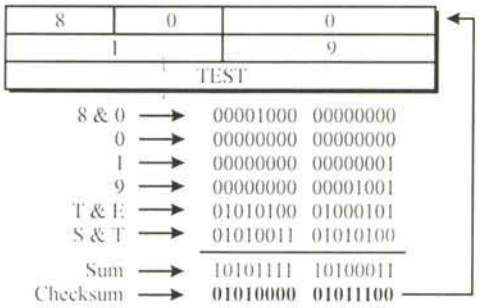
۳- اگر نتیجه بدست آمده در بند ۲ برابر با تمام صفر باشد، پیغام پذیرفته می‌شود و در غیر این صورت، دور انداخته می‌شود.

❊ مثال ۱

شکل ۹-۱۹ مثالی از نحوه محاسبه Checksum برای یک پیغام ساده درخواست Echo را نشان می‌دهد (شکل ۹-۱۴ را ببینید). به طور تصادفی مقدار فیلد identifier را برابر با 1 و

www.Abiri.ir

مقدار فیلد Sequence را برابر با 9 انتخاب کردیم. پیغام به بخشهای 16 بیتی تقسیم می‌شود. این بخشها با هم جمع می‌شوند و مکمل مجموع محاسبه می‌گردد. حالا فرستنده می‌تواند این مقدار را در فیلد Checksum بنویسد.



شکل ۹-۱۹ : مثالی از نحوه محاسبه Checksum

۹-۳ ابزارهای خطایابی

می‌توان از چندین ابزار برای ***خطایابی***[1] در اینترنت استفاده کرد. می‌توانیم بفهمیم که آیا یک کامپیوتر به شبکه متصل و در حال کار است یا خیر. می‌توانیم مسیری که یک بسته طی می‌کند را مشخص کنیم. دو ابزار که برای خطایابی به کار می‌روند و از ICMP استفاده می‌کنند را معرفی می‌کنیم: ping و traceroute . بعد از آنکه در فصلهای آتی پروتوکلهای دیگری را معرفی کردیم، ابزارهای بیشتری را معرفی خواهیم کرد.

ابزار ping

می‌توانیم از برنامه ping استفاده کنیم و به کمک آن دریابیم که آیا کامپیوتری به شبکه متصل است و در حال کار است یا خیر. از برنامه ping در فصل هشتم استفاده کردیم تا گزینه « ضبط کردن مسیر » را شبیه سازی کنیم. با جزئیات بیشتری درباره ping صحبت خواهیم کرد تا ببینیم که چگونه از بسته‌های ICMP استفاده می‌کند.

کامپیوتر مبدأ، پیغامهای درخواست Echo که ICMP ای با type=8 و Code=0 است را ارسال می‌کند. کامپیوتر مقصد اگر به شبکه متصل و در حال کار باشد، با پیغام پاسخ Echo که ICMP ای با type=0 و Code=0 است، پاسخ می‌دهد. برنامه ping ، فیلد identifier در پیغام درخواست و پاسخ را ست می‌کند و فیلد Sequence number را از مقدار 0 شروع می‌کند. هر

1- debugging

www.Abiri.ir

زمان که یک پیغام جدید فرستاده می‌شود، یک واحد به مقدار این فیلد افزوده می‌شود. توجه نمائید که ping می‌تواند زمان رفت و برگشت را محاسبه کند. ping زمان ارسال را در بخش دیتای پیغام می‌نویسد. وقتی که بسته برمی‌گردد، زمان دریافت از زمان ارسال کم می‌شود تا زمان رفت و برگشت (Round Trip Time : RTT) محاسبه شود.

* مثال ۲

از برنامه ping استفاده می‌کنیم تا سرور fhda.edu را تست کنیم. نتیجه در زیر، نشان داده شده است. توجه نمایید که این برنامه ping ، روی سیستم عامل یونیکس است و هر سیستم عاملی ، برنامه ping خاص خود را دارد.

```
$ ping fhda.edu
PING fhda.edu ( 153.18.8.1 ) 56 ( 84 ) bytes of data.
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ): icmp_seq = 0    ttl=62   time=1.91ms
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ): icmp_seq = 1    ttl=62   time=2.04ms
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ): icmp_seq = 2    ttl=62   time=1.90ms
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ): icmp_seq = 3    ttl=62   time=1.97ms
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ): icmp_seq = 4    ttl=62   time=1.93ms
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ): icmp_seq = 5    ttl=62   time=2.00ms
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ): icmp_seq = 6    ttl=62   time=1.94ms
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ): icmp_seq = 7    ttl=62   time=1.94ms
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ): icmp_seq = 8    ttl=62   time=1.97ms
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ): icmp_seq = 9    ttl=62   time=1.89ms
64 bytes from tiptoe.fhda.edu ( 153.18.8.1 ): icmp_seq = 10   ttl=62   time=1.98ms

--- fhda.edu ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10103 ms
rtt min/avg/max = 1.899/1.955/2.041 ms
```

برنامه ping بسته‌هایی را که مقدار Sequence number آنها از صفر شروع می‌شود را ارسال می‌کند. برای هر بسته، مقدار زمان RTT محاسبه می‌شود. فیلد TTL بسته IP که حاوی یک پیغام ICMP است، برابر با 62 ست شده است و به این معناست که این بسته IP نمی‌تواند بیش از 62 هاپ را طی کند. در آغاز، ping حجم دیتای پیغام را برابر با 84 بایت قرار داده است. روشن است که اگر ما 8 بایت هدر ICMP و 20 بایت هدر IP را به 56 بایت دیتای ICMP اضافه کنیم، نتیجه 84 بایت بدست می‌آید. با این حال توجه نمائید که به ازای هر بسته ICMP که ارسال می‌شود، عبارت 64 bytes در ابتدای هر سطر خروجی برنامه ping نوشته شده است. 64 بایت، کل بایتهای پیغام ICMP (56 بایت دیتا و 8 بایت هدر) را نشان می‌دهد.

www.Abiri.ir

اگر با استفاده از ***کلید وقفه***[1] (به عنوان مثال ctrl+c)، ارسال بسته‌های ping را متوقف نکنیم، برنامه ping به طور پیوسته پیغامهای درخواست Echo را ارسال می‌کند. بعد از آنکه ping متوقف شد، آمارگانی درباره بسته‌های ارسالی را نشان می‌دهد. این آمارگان، تعداد بسته ارسالی، دریافتی، کل زمان سپری شده، حداقل زمان RTT ، ماکزیمم زمان RTT و متوسط آن را نشان می‌دهد. بعضی از سیستم‌ها ممکن است اطلاعات بیشتری را نشان دهند. دقت نمائید که این نوع طرز کار برنامه ping عمومیت ندارد و برنامه ping سیستم عامل شما ممکن است خروجی متفاوت دیگری داشته باشد. وجه تشابه همه برنامه‌های ping این است که پیغامهای درخواست Echo ، ارسال می‌شوند و پیغامهای پاسخ Echo ، دریافت می‌گردند.

* مثال ۳

به عنوان دومین مثال، می‌خواهیم بدانیم که آیا یک سرور پستی (mail server) روی کامپیوتر adelphianet به شبکه متصل و در حال کار است یا خیر. نتیجه به دنبال می‌آید:

```
$ ping mail.adelphia.net
PING mail.adelphia.net ( 68.168.78.100 ) 56 ( 84 ) bytes of data.
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 0    ttl=48   time=85.4ms
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 1    ttl=48   time=84.6ms
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 2    ttl=48   time=84.9ms
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 3    ttl=48   time=84.3ms
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 4    ttl=48   time=84.5ms
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 5    ttl=48   time=84.7ms
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 6    ttl=48   time=84.6ms
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 7    ttl=48   time=84.7ms
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 8    ttl=48   time=84.4ms
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 9    ttl=48   time=84.2ms
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 10   ttl=48   time=84.9ms
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 11   ttl=48   time=84.6ms
64 bytes from mail.adelphia.net (68.168.78.100 ): icmp_seq = 12   ttl=48   time=84.5ms

--- mail.adelphia.net ping statistics ---
14 packets transmitted, 13 received, 7% packet loss, time 13129 ms
rtt min/avg/max/mdev = 84.207/84.694/85.469
```

دقت نمائید که در این مورد، 14 بسته فرستاده‌ایم، اما فقط 13 بسته دریافت کرده‌ایم. ممکن است قبل از رسیدن پاسخ آخرین بسته با Sequence number برابر با 13 ، برنامه را متوقف کرده باشیم.

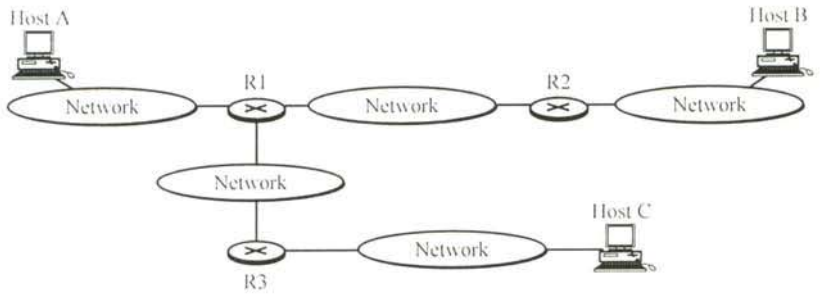
1- interrupt key

www.Abiri.ir

ابزار traceroute

برنامه traceroute در یونیکس و یا tracert در ویندوز را می‌توان برای دنبال کردن مسیر یک بسته از مبدأ تا مقصد، به کار برد. در فصل قبل، یکی از کاربردهای برنامه traceroute برای پیاده‌سازی گزینه‌های « مسیردهی قطعی از مبدأ » و « مسیردهی نه چندان قطعی از مبدأ » در بسته IP را دیدیم. در این فصل، از این برنامه در رابطه با بسته‌های ICMP استفاده می‌کنیم.

برنامه traceroute به نحو جالبی از دو پیغام ICMP به نامهای « مهلت مقرر سپری شده است » و « مقصد، غیر قابل دسترسی است » استفاده می‌کند تا مسیر یک بسته را پیدا کند. traceroute یک برنامه در سطح لایه کاربرد است که از سرویسهای UDP استفاده می‌کند (فصل یازدهم را ببینید). اجازه دهید طرز کار برنامه traceroute را با استفاده از شکل ۹-۲۰ توضیح دهیم.



شکل ۹-۲۰ : طرز کار برنامه traceroute

در شکل ۹-۲۰ ، بسته‌ای از کامپیوتر A به کامپیوتر B فرستاده می‌شود و می‌بینیم که این بسته از میان روترهای R1 و R2 عبور می‌کند. با این حال در بیشتر مواقع از توپولوژی شبکه مطلع نیستیم و لذا از A تا B می‌تواند مسیرهای مختلفی وجود داشته باشند. برنامه traceroute از پیغامهای ICMP و فیلد TTL در بسته IP برای پیدا کردن مسیر استفاده می‌کند.

۱- برنامه traceroute قدمهای زیر را برای پیدا کردن آدرس روتر R1 و زمان رفت و برگشت بین کامپیوتر A و روتر R1 ، برمی‌دارد.

الف. برنامه traceroute در کامپیوتر A با استفاده از UDP ، بسته‌ای را برای کامپیوتر B ارسال می‌کند. این پیغام در یک بسته IP با مقدار TTL برابر با 1 کپسول‌بندی می‌شود. برنامه، زمان ارسال بسته را ثبت می‌کند.

www.Abiri.ir

ب. روتر R1 بسته را دریافت می‌کند و از مقدار TTL یک واحد کم می‌کند و مقدار جدید TTL برابر با صفر می‌شود و لذا بسته را دور می‌اندازد. سپس روتر، یک پیغام ICMP از نوع Time exceeded (type=11 , Code=0) ارسال می‌کند تا نشان دهد که مقدار TTL برابر با صفر شده است و لذا بسته IP دور انداخته شده است.

ج- برنامه traceroute پیغام ICMP را دریافت می‌کند و با استفاده از آدرس مبدأ بسته IP ای که پیغام ICMP را کپسول‌بندی کرده است، آدرس IP روتر R1 یعنی ارسال‌کننده پیغام ICMP را بدست می‌آورد. برنامه زمان دریافت بسته را نیز ثبت می‌کند. تفاوت بین این زمان با زمان ثبت شده بند الف، مقدار زمان رفت و برگشت بسته را می‌دهد.

برنامه traceroute مراحل الف تا ج را سه مرتبه تکرار می‌کند تا مقدار متوسط بهتری برای زمان رفت و برگشت بدست آورد. اولین زمان رفت و برگشت ممکن است خیلی بیشتر از زمانهای رفت و برگشت دوم و سوم باشد به دلیل آنکه یک مدت زمانی طول می‌کشد تا برنامه ARP بتواند آدرس فیزیکی روتر R1 را پیدا کند. برای مرتبه دوم و سوم، آدرس فیزیکی روتر R1 در حافظه arp cache وجود خواهد داشت.

۲- برنامه traceroute مراحل قبل را تکرار می‌کند تا آدرس روتر R2 و زمان رفت و برگشت بین کامپیوتر A و روتر R2 را بدست آورد. با این حال این بار مقدار TTL را برابر با 2 قرار می‌دهد. بنابراین روتر R1 بسته را هدایت می‌کند درحالی که روتر R2 آن را دور می‌اندازد و یک پیغام ICMP از نوع Time exceeded برای کامپیوتر مبدأ می‌فرستد.

۳- برنامه traceroute قدمهای قبلی را تکرار می‌کند تا آدرس کامپیوتر B و زمان رفت و برگشت بین کامپیوتر A و کامپیوتر B را بدست آورد. وقتی که کامپیوتر B، بسته را دریافت می‌کند، یک واحد از مقدار TTL کم می‌کند اما پیغام را دور نمی‌اندازد زیرا که به مقصد نهایی خود رسیده است. چطور کامپیوتر B می‌تواند یک پیغام ICMP برای کامپیوتر A بفرستد؟ در اینجا برنامه traceroute از استراتژی دیگری استفاده می‌کند. پورت مقصد بسته UDP برابر با مقداری قرار داده می‌شود که پروتوکل UDP از آن پشتیبانی نمی‌کند. وقتی که کامپیوتر B بسته را دریافت می‌کند، نمی‌تواند برنامه‌ای را پیدا کند تا بسته را به آن تحویل دهد. بنابراین بسته را دور می‌اندازد و یک پیغام ICMP از نوع « مقصد، غیر قابل دسترسی است » (type=3 , Code=3) برای کامپیوتر A می‌فرستد. دقت نمائید که چنین وضعیتی در روتر R1 و روتر R2 رخ نمی‌دهد زیرا که یک روتر، هدر UDP را چک نمی‌کند. برنامه Traceroute ، آدرس مقصد بسته IP دریافتی را ثبت می‌کند و زمان رفت و برگشت را محاسبه می‌کند. دریافت یک پیغام « مقصد، غیر قابل دسترسی است » با مقدار Code=3 نشانه‌ای است از اینکه کل مسیر، پیدا شده است و نیازی به ارسال بسته‌های بیشتر نیست.

www.Abiri.ir

* مثال ۴

از برنامه traceroute برای پیدا کردن مسیر از کامپیوتر voyager.deanza.edu به سرور fhda.edu استفاده می‌کنیم. آنچه بدنبال می‌آید، نتیجه را نشان می‌دهد:

```
$ traceroute fhda.edu
traceroute to fhda.edu ( 153.18.8.1 ), 30 hops max, 38 byte packets
1  Dcore.fhda.edu     (153.18.31.254)  0.995 ms  0.899 ms  0.878 ms
2  Dbackup.fhda.edu   (153.18.251.4)   1.039 ms  1.064 ms  1.083 ms
3  tiptoe.fhda.edu    (153.18.8.1)     1.797 ms  1.642 ms  1.757 ms
```

سطرِ بدونِ شماره بعد از فرمان traceroute نشان می‌دهد که مقصد، 153.18.8.1 است، مقدار TTL برابر با 30 هاپ است و بسته حاوی 38 بایت است: 20 بایت هدر IP و 8 بایت هدر UDP و 10 بایت دیتا.

اولین خط، اولین روتری که بسته از آن عبور کرده است را نشان می‌دهد. نام روتر Dcore.fhda.edu با آدرس IP برابر با 153.18.31.254 است. اولین، دومین و سومین زمان رفت و برگشت بسته به ترتیب برابر با 0.995 میلی‌ثانیه، 0.899 میلی‌ثانیه و 0.878 میلی‌ثانیه است.

سطر دوم، دومین روتری که بسته از آن عبور کرده است را نشان می‌دهد. نام روتر Dbackup.fhda.edu و آدرس IP آن برابر با 153.18.251.4 است. زمان رفت و برگشت اول، دوم و سوم نیز نشان داده شده‌اند.

سطر سوم، کامپیوتر مقصد را نشان می‌دهد. می‌دانیم که این کامپیوتر،کامپیوتر مقصد است زیرا که سطر دیگری بعد از آن نیامده است. کامپیوتر مقصد fhda.edu است اما نام آن را tiptoe.fhda.edu با آدرس IP برابر با 153.18.8.1 گذاشته‌اند. سه تا زمانهای رفت و برگشت نیز نشان داده شده‌اند.

* مثال ۵

در این مثال، مسیر طولانی‌تری تا Xerox.com را طی می‌کنیم.

```
$ traceroute xerox.com
traceroute to xerox.com ( 13.1.64.93), 30 hops max, 38 byte packets
1  Dcore.fhda.edu     153.18.31.254     0.622 ms  0.891 ms  0.875 ms
2  Ddmz.fhda.edu      153.18.251.40     2.132 ms  2.266 ms  2.094 ms
3  Cinic.fhda.edu     153.18.253.126    2.110 ms  2.145 ms  1.763 ms
4  Cenic.net          137.164.32.140    3.069 ms  2.875 ms  2.930 ms
5  Cenic.net          137.164.22.31     4.205 ms  4.870 ms  4.197 ms
6  Cenic.net          137.164.22.167    4.250 ms  4.159 ms  4.078 ms
7  Cogentco.com       38.112.6.225      5.062 ms  4.825 ms  5.020 ms
8  Cogentco.com       66.28.4.69        6.070 ms  6.207 ms  5.653 ms
```

www.Abiri.ir

9	Cogentco.com	66.28.4.94	6.070 ms	5.928 ms	5.499 ms
10	Cogentco.com	154.54.2.226	6.545 ms	6.399 ms	6.535 ms
11	Sbcglobal.net	151.164.89.241	6.379 ms	6.370 ms	6.210 ms
12	Sbcglobal.net	64.161.1.45	6.908 ms	6.748 ms	7.359 ms
13	Sbcglobal.net	64.161.1.29	7.023 ms	7.040 ms	6.734 ms
14	Snfc21.pbi.net	151.164.191.49	7.656 ms	7.129 ms	6.866 ms
15	Sbcglobal.net	151.164.243.58	7.844 ms	7.545 ms	7.353 ms
16	Pacbell.net	209.232.138.114	9.857 ms	9.535 ms	9.603 ms
17	209.233.48.223	209.233.48.223	10.634ms	10.771ms	10.592ms
18	Alpha.Xerox.COM	13.1.64.93	11.172ms	11.048ms	10.922ms

در اینجا بین مبدأ و مقصد هفده هاپ داریم. دقت نمائید که بعضی از زمانهای رفت و برگشت به نظر غیر طبیعی می‌آیند. ممکن است به این دلیل باشد که روتر خیلی مشغول است و لذا نمی‌تواند بلافاصله بسته‌ای را پردازش نماید.

* مثال ۶

نکته جالب این است که کامپیوتر می‌تواند خودش را traceroute کند. به این ترتیب که آدرس خودش را در خط فرمان بنویسد. همانطور که انتظار داریم، بسته‌ها برای آدرس ***برگردان***[1] فرستاده می‌شوند.

```
$ traceroute voyager.deanza.edu
traceroute to voyager.deanza.edu (127.0.0.1), 30 hops max, 38 byte packets
1  voyager  (127.0.0.1)  0.178 ms  0.086 ms  0.055 ms
```

* مثال ۷

در نهایت از برنامه traceroute برای پیدا کردن مسیر بین fhda.edu و mhhe.com (سرور McGraw-Hill) استفاده می‌کنیم. توجه داریم که نمی‌توان کل مسیر را پیدا کرد. وقتی که traceroute در خلال 5 ثانیه پاسخی دریافت نمی‌کند، یک ستاره را به منزله وقوع مشکلی نمایش می‌دهد و سپس سراغ هاپ بعدی می‌رود.

```
$ traceroute mhhe.com
traceroute to mhhe.com ( 198.45.24.104), 30 hops max, 38 byte packets
1  Dcore.fhda.edu  153.18.31.254  1.025 ms  0.892 ms  0.880 ms
2  Ddmz.fhda.edu  153.18.251.40  2.141 ms  2.159 ms  2.103 ms
3  Cinic.fhda.edu  153.18.253.126  2.159 ms  2.050 ms  1.992 ms
4  Cenic.net  137.164.32.140  3.220 ms  2.929 ms  2.943 ms
5  Cenic.net  137.164.22.59  3.217 ms  2.998 ms  2.755 ms
```

1- loopback

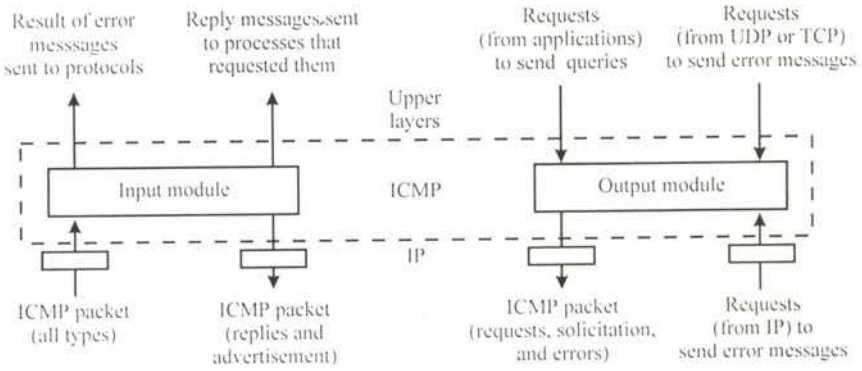
www.Abiri.ir

6	SanJose1.net	209.247.159.109	10.65ms	10.63ms	10.61ms
7	SanJose2.net	64.159.2.1	10.80ms	10.79ms	10.63ms
8	Denver1.Level3.net	64.159.1.114	43.40ms	43.36ms	43.41ms
9	Denver2.Level3.net	4.68.112.162	43.33ms	43.29ms	43.34ms
10	unknown	64.156.40.134	55.50ms	55.46ms	55.64ms
11	Mcleodusa1.net	64.198.100.2	60.96ms	55.68ms	55.46ms
12	Mcleodusa2.net	64.198.101.202	55.69ms	55.61ms	55.50ms
13	Mcleodusa3.net	64.198.101.142	56.05ms	55.62ms	56.33ms
14	Mcleodusa4.net	209.253.101.178	297.1ms	192.7ms	250.5ms
15	Eppg.com	198.45.24.246	71.21ms	70.53ms	70.66ms
16	***				
17	***				

.....

۹-۴ نرم‌افزار ICMP

برای آنکه ایده‌ای درباره نحوه ارسال و دریافت پیغامهای ICMP داشته باشیم، نسخه خودمان از نرم‌افزار ICMP که از دو ماجول ساخته شده است را ارائه می‌دهیم: یک ماجول ورودی و یک ماجول خروجی. شکل ۹-۲۱ این دو ماجول را نشان می‌دهد.



شکل ۹-۲۱ : نرم‌افزار ICMP

ماجول ورودی

ماجول ورودی[1] همه پیغامهای ICMP دریافتی را پردازش می‌کند. وقتی که یک پیغام ICMP از لایه IP به نرم‌افزار ICMP داده می‌شود، ماجول ورودی فراخوانی می‌شود. اگر بسته دریافتی، یکی از انواع درخواستها باشد، ماجول ورودی، پاسخ مربوطه را تولید و ارسال می‌کند.

1- input module

www.Abiri.ir

اگر بسته دریافتی یک « درخواست روتر» باشد، ماجول ورودی پیغام « اطلاع رسانی روتر» را تولید و ارسال می‌کند.

اگر بسته دریافتی یک پیغام « مسیردهی این بسته باید اصلاح شود » باشد، ماجول ورودی از اطلاعات درون آن استفاده می‌کند و جدول مسیردهی را بر آن اساس، به روزرسانی می‌کند. اگر پیغام دریافتی یک گزارش خطا باشد، ماجول ورودی، پروتوکل ICMP را از وقوع خطا با خبر می‌کند. **شبه کد**[1] ماجول ورودی به دنبال می‌آید.

```
* جریان رویدادهای ماجول ورودی
دریافت : یک پیغام ICMP از لایه IP
۱- اگر مقدار type برابر با یکی از سه نوع درخواستهای ICMP باشد:
    ۱- پاسخی را ایجاد می‌کند.
    ۲- پاسخ را ارسال می‌کند.
۲- اگر مقدار type معرف پیغام « درخواست روتر » باشد :
    ۱- اگر این وسیله که پیغام را دریافت کرده است، یک روتر است:
        ۱- یک پیغام « اطلاع رسانی روتر » تولید می‌کند.
        ۲- پیغام را ارسال می‌کند.
۳- اگر type برابر با یکی از پیغامهای پاسخ یا پیغام « اطلاع‌رسانی روتر» باشد :
    ۱- اطلاعات موجود در بخش دیتای بسته را استخراج می‌کند.
    ۲- اطلاعات استخراج شده را به برنامه‌ای که آن را درخواست کرده است، می‌دهد.
۴- اگر type معرف پیغام « مسیردهی این بسته باید اصلاح شود » باشد :
    ۱- جدول مسیردهی را به روز رسانی می‌کند.
۵- اگر type نشان دهنده یک پیغام ICMP خطا به غیر از « مسیردهی این بسته باید
اصلاح شود » باشد :
    ۱- وضعیت را به اطلاع کامپیوتر مبدأ ارسال کننده بسته مولد خطا می‌رساند.
۶- برمی‌گردد.
```

ماجول خروجی

ماجول خروجی[2] مسئول تولید پیغامهای درخواست، پیغام « درخواست روتر» یا پیغامهای خطایی است که یک پروتوکل لایه بالاتر مانند TCP و UDP و یا پروتوکل IP ارسال آن را

1- Pseudocode
2- output module

www.Abiri.ir

درخواست کرده است. اگر درخواست ارسال پیغام خطای ICMP از سوی پروتوکل IP باشد، ابتدا ماجول خروجی باید چک کند ببیند آیا این خواسته، مجاز است یا خیر. به خاطر داشته باشید که در چهار حالت نمی‌توان یک پیغام خطای ICMP را تولید کرد: الف. یک بسته IP که یک پیغام خطای ICMP را حمل می‌کند. ب. یک قطعه از بسته IP ج. یک بسته IP به آدرس مقصد چندگیرنده د. یک بسته IP که حاوی آدرس 0.0.0.0 یا 127.x.y.z است.

همچنین ماجول خروجی ممکن است خواسته را از یک برنامه کاربردی برای ارسال یکی از انواع پیغامهای درخواست و یا « درخواست روتر » دریافت کرده باشد. شبه کد ماجول خروجی به دنبال می‌آید.

*** جریان رویدادهای ماجول خروجی**

دریافت : یک خواسته
۱- اگر خواسته، ارسال یک پیغام خطا باشد :
  ۱- اگر این خواسته از سوی پروتوکل IP صادر شده باشد :
    ۱- اگر ارسال پیغام خطا برای این خواسته، ممنوع باشد :
      ۱- برمی‌گردد.
    ۲- اگر type معرف یک پیغام « مسیردهی این بسته باید اصلاح شود » باشد :
      ۱- اگر این ماشین، یک روتر است:
        ۱- برمی‌گردد.
    ۳- پیغام خطا را با استفاده از type ، code و بسته IP مولد خطا، تولید می‌کند.
۲- اگر خواسته، ارسال یکی از انواع پیغامهای درخواست و یا « درخواست روتر » باشد :
  ۱- آن پیغام درخواست یا « درخواست روتر » را تولید می‌کند.
۳- پیغام را ارسال می‌کند.
۴- برمی‌گردد.

۹-۵ عبارات کلیدی

پیغام پاسخ مسک (address mask reply message)
پیغام درخواست مسک (address mask request message)
پیغام مقصد، غیر قابل دسترسی است (destination unreachable message)
پیغامهای درخواست echo و پاسخ echo (echo request and echo reply message)
پیغام گزارش خطا (error reporting message)
پروتوکل پیغامهای کنترلی اینترنت (Internet Control Message Protocol : ICMP)

www.Abiri.ir

پیغام مشکلی در هدر وجود دارد (Parameter problem message)
پیغام درخواست و پاسخ (query message)
مسیردهی این بسته باید اصلاح شود (redirection)
زمان رفت و برگشت (Round Trip Time : RTT)
پیغامهای درخواست روتر و اطلاع‌رسانی روتر(router solicitation and router advertisement)
پیغام مبدأ، کمتر بسته بفرست (Source quench message)
پیغام مهلت مقرر سپری شده است (time exceeded message)
پیغام درخواست برچسب زمانی و پاسخ برچسب زمانی (timestamp request and reply)
ابزار traceroute

۹-۶ خلاصه

- ICMP پنج نوع پیغام گزارش خطا و چهار جفت پیغامهای درخواست و پاسخ را به منظور پشتیبانی از پروتوکل فاقد اتصال و غیر قابل اطمینان IP ، ارائه کرده است.
- پیغامهای ICMP در بسته‌های IP کپسول‌بندی می‌شوند.
- وقتی که نمی‌توان یک بسته IP را به مقصد تحویل داد، پیغام خطای « مقصد، غیر قابل دسترسی است » برای مبدأ ارسال می‌شود.
- به منظور برطرف کردن ازدحام، پیغام خطای « مبدأ، کمتر بسته بفرست » برای مبدأ ارسال می‌شود.
- پیغام « مهلت مقرر سپری شده است » به کامپیوتر مبدأ خبر می‌دهد که یا فیلد TTL بسته IP به صفر رسیده است و یا کلیه قطعات یک بسته IP در خلال یک مدت زمان از قبل تعیین شده از راه نرسیده‌اند.
- پیغام « مشکلی در هدر وجود دارد » به کامپیوتر مبدأ خبر می‌دهد که مشکلی در یکی از فیلدهای هدر بسته IP وجود دارد.
- به منظور کارآمدتر کردن جدول مسیردهی کامپیوتر، پیغام « مسیردهی این بسته باید اصلاح شود » ارسال می‌شود.
- اتصال موجود بین دو وسیله متصل به شبکه را می‌توان به کمک پیغام درخواست Echo و پاسخ Echo تست کرد.
- پیغامهای درخواست مسک و پاسخ مسک برای پیدا کردن مسک زیرشبکه به کار می‌روند.
- پیغامهای « درخواست از روتر » و « اطلاع رسانی روتر » به کامپیوترها امکان به روز رسانی جداول مسیردهی خود را می‌دهند.

www.Abiri.ir

- فیلد Checksum در پیغام ICMP از روی هدر و دیتای فیلد پیغام ICMP محاسبه می‌شود.
- ping یک برنامه کاربردی است که از سرویس ICMP برای تست قابلیت دسترسی به کامپیوتری در شبکه استفاه می‌کند.
- یک طرح ساده نرم‌افزاری ICMP می‌تواند از یک ماجول ورودی که بسته‌های ICMP وروردی را پردازش می‌کند و یک ماجول خروجی که خواسته‌های مبنی بر ارسال پیغامهای ICMP را برآورده می‌سازد، تشکیل شده باشد.

۹-۷ مجموعه تمرین‌ها

تمرین‌ها

۱- کامپیوتر A یک پیغام درخواست Timestamp را برای کامپیوتر B می‌فرستد و هرگز پاسخی را دریافت نمی‌کند. سه علت ممکن و عمل متناظر با آنها را مورد بحث قرار دهید.

۲- چرا اگر یک پیغام خطای ICMP دچار خطا شود، نباید یک پیغام خطای ICMP برای آن ارسال کرد؟

۳- کامپیوتر A بسته‌ای را برای کامپیوتر B ارسال می‌کند. کامپیوتر B هرگز بسته را دریافت نمی‌کند و کامپیوتر A هرگز از خطای رخ داده با خبر نمی‌شود. دو توضیح مختلف ارائه دهید که نشان دهد چه اتفاقی ممکن است افتاده باشد.

۴- هدف از ارسال هدر بسته IP مولد خطا و هشت بایت اول دیتای آن به همراه پیغام ICMP گزارش خطا چیست؟

۵- ماکزیمم مقدار فیلد Pointer در یک پیغام گزارش خطای « مشکلی در هدر وجود دارد » چیست؟

۶- مثالی از وضعیتی ارائه دهید که در آن یک کامپیوتر، هرگز پیغام « مسیردهی این بسته باید اصلاح شود » را دریافت نکند.

۷- جدولی بسازید که نشان دهد کدام پیغامهای ICMP توسط روترها، کدام توسط کامپیوترهایی که مقصد نهایی نیستند و کدام توسط کامپیوترهای مقصد نهایی ارسال می‌شوند.

۸- آیا محاسبات مربوط به زمان ارسال، زمان دریافت یا زمان رفت و برگشت می‌تواند به یک مقدار منفی منجر شود؟ چرا بله ، چرا نه ؟ مثالی ارائه دهید.

۹- چرا زمان واقعی رفت (یا برگشت) یک بسته صرفاً زمان رفت و برگشت تقسیم بر دو نیست؟

۱۰ – حداقل سایز یک بسته ICMP چیست؟ ماکزیمم سایز یک بسته ICMP چیست؟

www.Abiri.ir

۱۱- حداقل سایز یک بسته IP که حاوی یک پیغام ICMP است، چیست؟ ماکزیمم سایز آن چیست؟

۱۲- حداقل سایز یک فریم اترنت حاوی یک بسته IP که آن نیز حاوی یک پیغام ICMP است، چقدر است؟ ماکزیمم سایز چقدر است؟

۱۳- چگونه می‌توان تعیین کرد که آیا یک بسته IP حاوی یک پیغام ICMP است یا خیر.

۱۴- برای پیغام ICMP زیر، مقدار Checksum را محاسبه نمائید.

۱۵- روتر یک بسته IP با آدرس مبدأ 130.45.3.3 و آدرس مقصد 201.23.4.6 را دریافت می‌کند. روتر نمی‌تواند آدرس IP مقصد را در جدول مسیردهی خودش پیدا کند. فیلدهای پیغام ICMP ای که ارسال می‌شود را تا آنجا که می‌توانید پر کنید.

۱۶ – TCP سگمنتی با آدرس پورت مقصد 234 را دریافت می‌کند. TCP چک می‌کند و می‌بیند که نمی‌تواند یک پورت آزاد را برای این مقصد باز کند. فیلدهای پیغام ICMP ای که ارسال می‌شود را پر کنید.

۱۷- یک پیغام ICMP دریافت شده است که هدر آن حاوی مقادیر زیر (بر حسب هگزادسیمال) است :

03 03 10 20 00 00 00 00

این چه نوع پیغام ICMP است؟ مقدار Code آن چقدر است؟ هدف از ارسال این پیغام چیست؟

۱۸- یک پیغام ICMP دریافت شده است که هدر آن حاوی مقادیر زیر (برحسب هگزادسیمال) است:

05 00 11 12 11 0B 03 02

این چه نوع پیغام ICMP ای است؟ مقدار Code آن چقدر است؟ هدف از ارسال این پیغام چیست؟ مقدار آخرین چهار بایت معرف چیست؟ آخرین بایتها بر چه مطلبی صحه می‌گذارند؟

۱۹- یک کامپیوتر یک پیغام درخواست Timestamp را ارسال می‌کند. اگر کلاک، مقدار 5:20:30 AM را نشان دهد (به وقت گرین‌ویچ)، محتوای پیغام را نشان دهید.

۲۰- تمرین نوزدهم را برای کلاک 3:40:30 PM (به وقت گرین‌ویچ) تکرار کنید.

۲۱- یک کامپیوتر یک پیغام درخواست timestamp را از کامپیوتر دیگری در ساعت 2:34:20 PM دریافت می‌کند. مقدار فیلد Original Timestamp برابر با 52453000 می‌باشد. اگر کلاک فرستنده به اندازه 5 میلی‌ثانیه عقب باشد، مقدار زمان واقعی رفت (یا برگشت) چقدر است؟

www.Abiri.ir

۲۲- کامپیوتر یک پیغام درخواست timestamp را برای کامپیوتر دیگری می‌فرستد. ساعت 3:46:07 AM پاسخ timestamp متناظر با آن را دریافت می‌کند. مقدار فیلدهای Original Timestamp ، Receive Timestamp و Transmit Timestamp به ترتیب برابر با 13560000 ، 13562000 و 13564300 است. زمان ارسال چقدر است؟ زمان دریافت چقدر است؟ زمان رفت و برگشت چقدر است؟ تفاوت زمانی بین کلاک فرستنده و کلاک گیرنده چقدر است؟

۲۳- اگر فاصله بین دو کامپیوتر 5000 مایل باشد، حداقل زمان لازم برای آنکه پیغامی از یک کامپیوتر به کامپیوتر دیگر برود، چقدر است؟

فعالیتهای تحقیقاتی

۲۴- از برنامه ping برای تست کامپیوتر خودتان (loopback) استفاده کنید.

۲۵- از برنامه ping برای تست یک کامپیوتر در آمریکا استفاده کنید.

۲۶- از برنامه ping برای تست یک کامپیوتر خارج از آمریکا استفاده کنید.

۲۷- از برنامه traceroute (یا tracert) برای پیدا کردن مسیر از کامپیوتر خودتان تا کامپیوتری در یک کالج یا دانشگاه استفاده کنید.

۲۸- نشان دهید که چگونه با استفاده از تمرین قبل می‌توانید RTT بین دو روتر را بدست آورید.

۲۹- RFCهای مربوط به پروتوکل ICMP را پیدا کنید. آیا برای هر نوع از انواع پیغامهای ICMP ، یک RFC وجود دارد؟

www.Abiri.ir

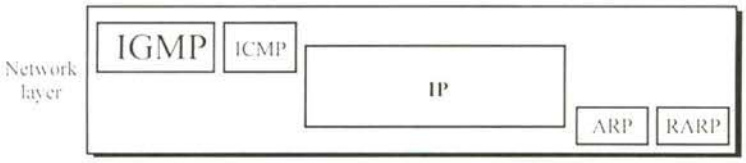
فصل دهم

پروتوکل IGMP

در دو نوع مخابره دیتا می‌توان از پروتوکل IP استفاده کرد: ***تک‌گیرندگی***[1] و ***چندگیرندگی***[2] . تک‌گیرندگی، مخابره بین یک فرستنده و یک گیرنده است و لذا یک مخابره یک به یک است. اما برخی از برنامه‌ها به ارسال یک پیغام مشابه به طور همزمان برای تعداد زیادی از گیرندگان، نیاز دارند. به این نوع مخابره، چندگیرندگی گفته می‌شود و یک مخابره یک به چند است که کاربردهای متعددی دارد. به عنوان مثال می‌توان اطلاعات سهام را به طور همزمان برای چندین سهامدار ارسال کرد و یا خبر لغو شدن یک پرواز را می‌توان برای چندین آژانس هواپیمایی ارسال کرد. سایر کاربردها شامل یادگیری از راه دور و ویدئو طبق درخواست می‌باشد.

۱۰-۱ پروتوکل IGMP

IGMP[3] (پروتوکل مدیریت گروهها در اینترنت) یکی از پروتوکلهای لازم اما نه کافی است که در multicasting از آن استفاده می‌شود. پروتوکل IGMP همراه پروتوکل IP است. شکل ۱۰-۱ جایگاه پروتوکل IGMP را نسبت به سایر پروتوکلها در لایه شبکه نشان می‌دهد.



شکل ۱۰-۱: جایگاه پروتوکل IGMP در لایه شبکه

مدیریت گروه

برای آنکه در اینترنت بتوانیم multicasting را اجرا کنیم به روترهایی نیاز داریم که بتوانند بسته‌هایی با آدرسهای مقصد ***چندگیرنده***[4] را مسیردهی کنند. به کمک یکی از

1- unicasting
2- multicasting
3- Internet Group Managment Protocol : IGMP
4- multicast

www.Abiri.ir

پروتوکلهای مسیردهی چند گیرندگی[1] که در فصل پانزدهم مورد بحث قرار خواهند گرفت، باید جدول مسیردهی این روترها به روزرسانی شود.

IGMP یک پروتوکل مسیردهی چندگیرندگی نیست بلکه ***پروتوکل مدیریت گروهها***[2] می‌باشد. در هر شبکه، یک یا چند ***روتر چندگیرندگی***[3] وجود دارد که پیغامهای با آدرس مقصد multicast را برای کامپیوترها یا سایر روترها ارسال می‌کنند.

پروتوکل IGMP ، اطلاعات عضویت کامپیوترها و روترهای متصل به شبکه در گروهها را به روترهای multicast می‌دهد. یک روتر multicast ممکن است روزانه هزاران بسته حاوی آدرسهای multicast که برای گروههای مختلف ارسال شده‌اند را دریافت کند. اگر روتر هیچگونه اطلاعی درباره وضعیت عضویت کامپیوترها در گروهها نداشته باشد، باید بسته‌های چند گیرندگی را به صورت ***همگانی***[4] ارسال نماید. این عمل، ترافیک زیادی ایجاد می‌کند و پهنای باند زیادی را مصرف می‌کند. یک راه حل بهتر آن است که روتر multicast ، لیستی از گروههای موجود در شبکه را نگه دارد به طوری که هر گروه حداقل یک عضو داشته باشد. IGMP به روتر multicast کمک می‌کند تا چنین لیستی را ایجاد کند و بروز رسانی نماید.

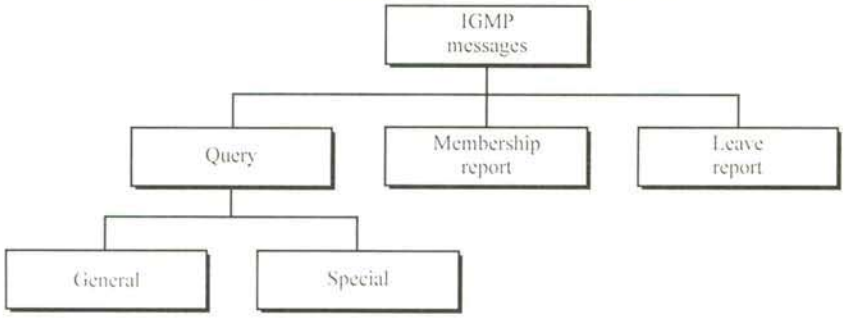
مهم: IGMP یک پروتوکل مدیریت گروه است. او به روتر multicast کمک می‌کند تا لیستی از گروهها با حداقل یک عضو در هر گروه را برای هر یک از اینترفیسهای روتر، نگهداری نماید.

۱۰-۲ پیغامهای IGMP

دو نسخه از IGMP نوشته شده است و ما درباره IGMPv2 که نسخه جاری است، صحبت می‌کنیم. IGMPv2 سه نوع پیغام دارد: پیغام ***پرسشی***[5] ، ***گزارش عضویت در گروه***[6] و ***گزارش ترک گروه***[7] . پیغامهای پرسشی، خود بر دو نوعند: ***عمومی***[8] و ***خاص***[9] . شکل ۱۰-۲ را ملاحظه فرمائید.

1- multicasting routing protocol
2- group management protocol
3- multicast router
4- broadcast
5- query
6- membership report
7- leave report
8- general
9- special

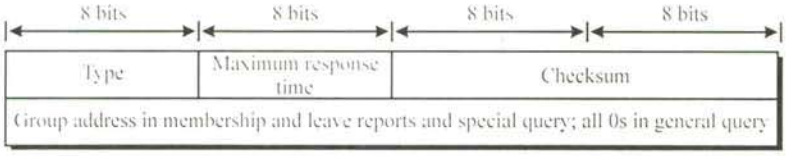
www.Abiri.ir



شکل ۱۰- ۲ : انواع پیغامهای IGMP

فرمت پیغام IGMP

شکل ۱۰-۳ فرمت یک پیغام IGMP را نشان می‌دهد :



شکل ۱۰-۳ : فرمت پیغام IGMP

Type : نوع . این فیلد هشت بیتی، نوع پیغام را مطابق جدول ۱۰-۱ نشان می‌دهد. مقدار type ، هم به صورت هگزادسیمال و هم به صورت باینری نشان داده شده است.

Type	*Value*
General or Special Query	0x11 or 00010001
Membership Report	0x16 or 00010110
Leave Report	0x17 or 00010111

جدول ۱۰-۱: فیلد type در IGMP

Maximum Response Time : ماکزیمم زمان پاسخگویی. این فیلد هشت بیتی، محدوده زمانی را نشان می‌دهد که طی آن باید پاسخی برای پرسش مطرح شده دریافت گردد. واحد آن، دهم ثانیه است. به عنوان مثال اگر مقدار این فیلد برابر با 100 باشد، به معنی 10 ثانیه است. مقدار این فیلد در « پیغام پرسشی »، مخالف با صفر است و در دو نوع دیگر پیغام، برابر با صفر می‌باشد. کاربرد آن را کمی بعد خواهیم دید.

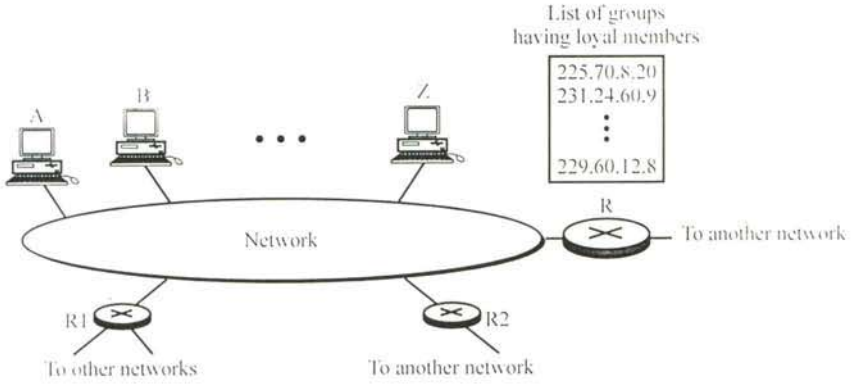
Checksum : جمع‌آزما. این یک فیلد 16 بیتی است که حاوی Checksum است. مقدار Checksum روی 8 بایت پیغام IGMP محاسبه می‌شود.

www.Abirr.ir

Group address : آدرس گروه. مقدار این فیلد در « پیغام پرسش عمومی » برابر با صفر است و در « پیغام پرسش خاص »، « پیغام گزارش عضویت در گروه » و « پیغام گزارش ترک گروه » مقداری مخالف صفر است.

۱۰-۳ نحوه کار IGMP

IGMP به صورت محلی کار می‌کند. یک روتر multicast که به شبکه متصل است، حاوی لیستی از آدرسهای multicast است به طوری که هر آدرس multicast، معرف یک گروه می‌باشد و این گروه، حاوی حداقل یک عضو است (شکل ۱۰-۴ را ببینید).



شکل ۱۰-۴: نحوه عمل IGMP

برای هر گروه، یک روتر وجود دارد که وظیفه توزیع بسته‌های multicast دریافتی برای آن گروه را به عهده دارد. معنی آن این است که اگر سه روتر multicast به شبکه متصل باشند، فقط یک روتر، بسته‌های multicast دریافتی برای آن گروه را در آن شبکه، توزیع می‌کند. به عنوان مثال در شکل ۱۰-۴ ، فقط روتر R بسته‌های multicast ارسالی برای گروهی به آدرس 225.70.8.20 را در این شبکه، توزیع می‌کند و روترهای R1 و R2 کاری به توزیع بسته‌های multicast فرستاده شده برای این گروه در همین شبکه، ندارند.

عضو یک گروه، می‌تواند یک کامپیوتر و یا یک روتر باشد. وقتی که یک کامپیوتر عضو یک گروه است، به این معنی است که برنامه‌ای در آن کامپیوتر، بسته‌های multicast ارسالی آن گروه را دریافت می‌کند. وقتی یک روتر عضو یک گروه است، به این معنی است که یکی از اینترفیسهای دیگر روتر، این بسته‌ها را دریافت می‌کند. می‌گوئیم که این کامپیوتر یا روتر به

www.Abiri.ir

این گروه، ***علاقه***[1] دارد. در هر دو حالت، کامپیوتر یا روتر، لیستی از آدرس گروهها (یا همان groupids) را نگه می‌دارد و این علاقه را به اطلاع ***روتر توزیع کننده***[2] می‌رساند.

به عنوان مثال در شکل ۱۰-۴ روتر R ، روتر توزیع کننده است. دو روتر multicast دیگر (R1 ، R2) نیز وجود دارند که آنها نیز پیغامهای multicast توزیع شده توسط روتر R برای گروههای موجود در شبکه را دریافت می‌کنند ولی آنها نقشی در توزیع این پیغامها در همین شبکه ندارند. البته ممکن است R1 و R2 وظیفه توزیع این پیغامها را در شبکه‌های دیگری بر عهده داشته باشند.

ملحق شدن به یک گروه

برنامه‌های کامپیوترها می‌توانند به گروهها ملحق شوند. برای این کار، کامپیوتر، لیست برنامه‌هایی که عضو یک گروه هستند را در خود نگه می‌دارد. وقتی که برنامه‌ای می‌خواهد به گروهی ملحق شود، درخواست خود را برای کامپیوتر می‌فرستد. کامپیوتر نام برنامه و نام گروه مورد درخواست او را به لیست خود اضافه می‌کند. اگر این برنامه، اولین عضو این گروه باشد، کامپیوتر یک پیغام « گزارش عضویت در گروه » را ارسال می‌کند. اگر این برنامه، اولین عضو گروه نباشد، نیازی به ارسال پیغام « گزارش عضویت در گروه » نیست زیرا که کامپیوتر هم‌اکنون عضو گروه است و لذا بسته‌های multicast ای که برای این گروه ارسال می‌شوند را دریافت می‌کند.

اینترفیسهای روترها نیز می‌توانند به گروهها ملحق شوند. یک روتر نیز لیستی از گروهها دارد که نشان می‌دهد هر اینترفیس روتر به چه گروهی علاقه‌مند است. وقتی که اینترفیسی می‌خواهد علاقه خود به گروهی را اعلام نماید، روتر یک پیغام « گزارش عضویت در گروه » ارسال می‌کند. به عبارت دیگر، روتر مانند یک کامپیوتر عمل می‌کند اما لیست روتر، طولانی‌تر است زیرا که علاقه همه اینترفیسهای روتر به گروهها (در صورت وجود) را نشان می‌دهد. دقت نمائید که پیغام « گزارش عضویت در گروه » از طریق تمام اینترفیسها به جز آن اینترفیسی که علاقه خود به آن گروه را اعلام کرده است، ارسال می‌شود.

شکل ۱۰-۵ یک پیغام « گزارش عضویت در گروه » را نشان می‌دهد که توسط یک کامپیوتر یا روتر ارسال شده است. پروتوکل IGMP ایجاب می‌کند که پیغام « گزارش عضویت در گروه »، دو مرتبه، یکی چند لحظه بعد از دیگری، ارسال شود. به این ترتیب اگر اولین گزارش، گم یا مخدوش شود، گزارش دوم جای آن را می‌گیرد.

1- interest
2- distributing router

www.Abiri.ir

مهم : در IGMP یک گزارش عضویت، دو مرتبه، یکی پس از دیگری، ارسال می‌شود.



شکل ۱۰-۵: گزارش عضویت

ترک گروه

وقتی که کامپیوتری ببیند که هیچیک از برنامه‌های او به گروه مشخصی علاقه‌مند نیستند، درباره آن گروه، یک پیغام « گزارش ترک گروه » ارسال می‌کند. به طور مشابه، وقتی که روتری ببیند که هیچیک از اینترفیسهایش به گروه مشخصی علاقه‌مند نیستند، درباره آن گروه، یک پیغام « گزارش ترک گروه » می‌فرستد.

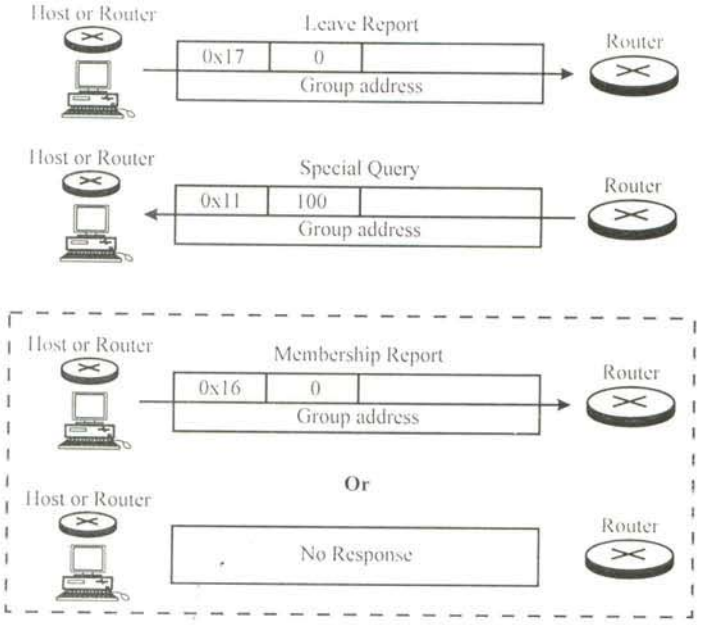
با این حال، وقتی که یک روتر multicast یک پیغام « گزارش ترک گروه » دریافت می‌کند، نمی‌تواند بلافاصله آن گروه را از لیستش حذف کند زیرا که گزارش فقط از یک کامپیوتر یا روتر آمده است. ممکن است کامپیوترها یا روترهای دیگری باشند که هنوز به آن گروه علاقه‌مند باشند. برای حصول اطمینان، روتر multicast یک پیغام « پرسش خاص » را به آدرس multicast گروه مورد نظر ارسال می‌کند. روتر یک زمان مشخصی مهلت می‌دهد تا در خلال آن، هر کامپیوتر یا روتری فرصت پاسخ داشته باشد. اگر در خلال این زمان، هیچ پیغام « گزارش عضویت در گروه » (مبنی بر علاقه کسی به آن گروه) به دست نرسد، روتر فرض را بر این می‌گذارد که این گروه، هیچ عضوی ندارد و لذا آن گروه را از لیست خود حذف می‌کند. شکل ۱۰-۶ مکانیزم ترک یک گروه را نشان می‌دهد.

کنترل عضویتها

یک کامپیوتر یا روتر می‌تواند با ارسال یک پیغام « گزارش عضویت در گروه » ، به گروهی ملحق شود. آنها می‌توانند با ارسال یک پیغام « گزارش ترک گروه »، گروهی را ترک کنند. با این حال، ارسال این دو نوع گزارش، کافی نیست. وضعیتی را در نظر بگیرید که در آن فقط یک کامپیوتر به یک گروه علاقه‌مند است و آن کامپیوتر، یا خاموش است و یا از سیستم جدا شده است. به این ترتیب روتر multicast هرگز برای آن گروه، « گزارش ترک گروه » را دریافت نخواهد کرد. چطور به این وضع رسیدگی شود؟ روتر multicast مسئول کنترل عضویت تمام کامپیوترها یا روترهای یک LAN است تا ببیند که آیا آنها می‌خواهند به عضویت خود در گروه ادامه دهند یا خیر.

www.Abiri.ir

روتر به طور پریودیک (به طور پیش‌فرض هر 125 ثانیه) یک پیغام «پرسش عمومی» ارسال می‌کند. در این پیغام، فیلد آدرس گروه برابر با 0.0.0.0 ست شده است و معنی آن این است که این پرسش درباره تداوم عضویت برای تمام گروهها و نه فقط یک گروه مشخص، ارسال شده است.



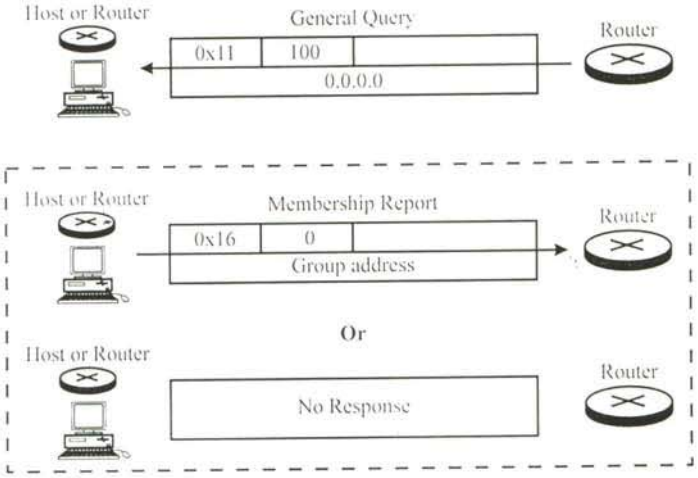
شکل ۱۰-۶ : گزارش ترک گروه

مهم : پیغام « پرسش عمومی » ، یک گروه خاص را مشخص نمی‌کند.

روتر انتظار دارد که برای هر گروهی در لیست گروههای خود، پاسخی را دریافت کند. حتی گروههای جدید ممکن است پاسخ دهند. مقدار فیلد maximum response time (ماکزیمم زمان پاسخ‌دهی) برابر 10 ثانیه است (عددی که داخل این فیلد نوشته می‌شود، در واقع عدد 100 است اما واحد آن دهم ثانیه است). وقتی یک کامپیوتر یا روتری « پیغام پرسش عمومی » را دریافت می‌کند، اگر به گروهی علاقه‌مند باشد، با یک « گزارش عضویت در گروه » پاسخ می‌دهد. با این وجود، اگر علاقه مشترکی وجود داشته باشد (به عنوان مثال، دو کامپیوتر به یک گروه مشابه علاقه‌مند باشند)، به منظور جلوگیری از ترافیک اضافی، فقط یک « گزارش عضویت در گروه » برای آن گروه ارسال می‌شود. به این عمل، ارسال پاسخ با تأخیر گفته

www.Abiri.ir

می‌شود. دقت نمائید که « پیغام پرسش عمومی » فقط باید توسط یک روتر (که معمولاً ***روتر پرسش‌کننده***[1] نامیده می‌شود) ارسال شود تا به این ترتیب نیز، مانع از بروز ترافیک اضافی گردد. شکل ۱۰-۷ مکانیزم ارسال « پرسش عمومی » را نشان می‌دهد.



شکل ۱۰-۷ : پیغام پرسش عمومی

ارسال پاسخ با تأخیر

برای آنکه مانع از بروز ترافیک اضافی شویم، IGMP از یک ***استراتژی ارسال پاسخ با تأخیر***[2] استفاده می‌کند. وقتی که یک کامپیوتر یا روتر، یک « پیغام پرسشی » را دریافت می‌کند، بلافاصله پاسخ نمی‌دهد بلکه با تأخیر پاسخ می‌دهد. هر کامپیوتر یا روتر از یک عدد تصادفی برای ست کردن مقدار اولیه یک تایمر کاهنده استفاده می‌کند و این عدد، مهلت مقرر را بین 1 تا 10 ثانیه قرار می‌دهد. زمان انقضاء مهلت مقرر می‌تواند با پله‌های زمانی یک ثانیه یا کمتر باشد. برای هر گروهی در این لیست، تایمری ست می‌شود. به عنوان مثال، تایمر اولین گروه ممکن است بعد از 2 ثانیه به انقضاء مهلت مقرر برسد اما تایمر گروه سوم ممکن است بعد از 5 ثانیه به انقضاء مهلت مقرر برسد. هر کامپیوتر یا روتر قبل از ارسال یک « پیغام گزارش عضویت در گروه »، صبر می‌کند تا تایمرش به انقضاء مهلت مقرر برسد. در خلال همین زمان، اگر تایمر کامپیوتر یا روتر دیگری برای همان گروه، زودتر به انقضاء مهلت مقرر برسد، آن

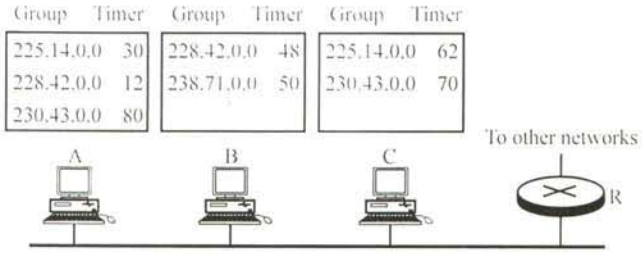
1- query router
2- delayed response strategy

www.Abiri.ir

کامپیوتر یا روتر یک « گزارش عضویت در گروه » ارسال می‌کند. از آنجا که گزارش به صورت همگانی ارسال می‌شود، کامپیوتر یا روتری که منتظر انقضای مهلت مقرر تایمر خود برای آن گروه است تا پیغام « گزارش عضویت در گروه » را ارسال کند، « گزارش عضویت در گروه » ارسالی از سوی کامپیوتر یا روتر فوق‌الذکر را دریافت می‌کند و در می‌یابد که نیازی به ارسال گزارش دیگری نیست و لذا تایمر خود، مربوط به آن گروه را خاموش می‌کند.

* مثال ۱

فرض کنید مطابق شکل ۱۰-۸ ، سه کامپیوتر در شبکه وجود داشته باشند:



شکل ۱۰-۸ : مثال ۱

در لحظه صفر، یک « پیغام پرسشی » دریافت شده است. مهلت مقرر تایمر هر گروه، که یک عدد تصادفی با واحد دهم ثانیه است، در کنار آدرس هر گروه، نوشته شده است. دنباله گزارشها را نشان دهید.

راه حل : رویدادها به ترتیب زیر رخ می‌دهند :

الف. لحظه 12 : تایمر مربوط به گروه 228.42.0.0 در کامپیوتر A به پایان مهلت مقرر می‌رسد و لذا کامپیوتر A ، یک « گزارش عضویت در گروه » می‌فرستد. روتر و هر کامپیوتری شامل کامپیوتر B ، این گزارش را دریافت می‌کنند و کامپیوتر B نیز که به این گروه علاقه‌مند است، تایمر خود برای این گروه یعنی 228.42.0.0 را خاموش می‌کند.

ب. لحظه 30 : تایمر مربوط به گروه 225.14.0.0 در کامپیوتر A به پایان مهلت مقرر می‌رسد و لذا کامپیوتر A ، یک « گزارش عضویت در گروه » می‌فرستد. هر روتر و هر کامپیوتری شامل کامپیوتر C ، این گزارش را دریافت می‌کنند و کامپیوتر C نیز که به این گروه علاقه‌مند است، تایمر خود برای این گروه یعنی 225.14.0.0 را خاموش می‌کند.

www.Abiri.ir

ج. لحظه 50 : تایمر مربوط به گروه 238.71.0.0 در کامپیوتر B به پایان مهلت مقرر می‌رسد و لذا کامپیوتر B یک « گزارش عضویت در گروه » می‌فرستد. روتر و هر کامپیوتری، این گزارش را دریافت می‌کنند.

د. لحظه 70 : تایمر مربوط به گروه 230.43.0.0 در کامپیوتر C ، به پایان مهلت مقرر می‌رسد و لذا کامپیوتر C ، یک « گزارش عضویت در گروه » می‌فرستد. هر روتر و هر کامپیوتری شامل کامپیوتر A ، این گزارش را دریافت می‌کنند و کامپیوتر A نیز که به این گروه علاقه‌مند است، تایمر خود برای این گروه یعنی 230.43.0.0 را خاموش می‌کند.

دقت نمائید که اگر هر کامپیوتری، یک گزارش برای هر گروه موجود در لیست خودش ارسال می‌کرد، هفت « گزارش عضویت در گروه » ارسال می‌شد اما با این استراتژی، فقط چهار گزارش ارسال شده است.

روتر پرسش‌کننده

« پیغامهای پرسشی » ممکن است پاسخهای فراوانی را تولید کنند. IGMP برای ممانعت از بروز ترافیک اضافی، یک روتر را برای هر شبکه، به عنوان ***روتر پرسش‌کننده***[1] تعریف می‌کند. فقط این روتر مشخص شده، « پیغامهای پرسشی » را ارسال می‌کند و سایر روترها در آن شبکه، « پیغامهای پرسشی » را ارسال نمی‌کنند بلکه آنها صرفاً صبر می‌کنند تا پاسخ ارسالی به این پرسشها را دریافت کنند و سپس بر اساس آن، لیستهای گروههای خود را بروزرسانی کنند .

۱۰-۴: کپسول‌بندی

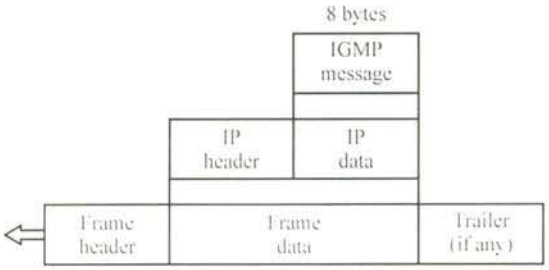
پیغام IGMP در یک بسته IP کپسول‌بندی می‌شود و خود این بسته IP ، در یک فریم لایه دیتالینک، کپسول‌بندی می‌شود. شکل ۱۰-۹ را ببینید. چندین مطلب درباره ***کپسول‌بندی***[2] وجود دارد که باید در اینجا مورد بحث قرار دهیم. ابتدا به لایه شبکه و سپس به لایه دیتالینک می‌پردازیم.

لایه شبکه

در سطح لایه شبکه یا همان IP، سه فیلد مورد توجه ما می‌باشد: فیلد protocol، فیلد TTL و فیلد آدرس IP مقصد.

1- query router
2- encapsulation

www.Abiri.ir



شکل ۱۰-۹: کپسول‌بندی پیغام IGMP

❊ فیلد protocol

در پروتوکل IGMP ، مقدار این فیلد برابر با 2 است. بنابراین اگر در فیلد protocol یک بسته IP ، عدد 2 نوشته شده باشد، دیتای آن بسته، باید به پروتوکل IGMP تحویل داده شود.

مهم: مقدار فیلد protocol یک بسته IP حامل پیغام IGMP ، برابر با 2 است.

❊ فیلد TTL

وقتی که پیغام IGMP در بسته IP کپسول‌بندی می‌شود، باید در فیلد TTL آن بسته IP ، عدد 1 نوشته شود. این کار لازم است زیرا که حوزه عمل IGMP ، محدوده یک LAN است. هیچ پیغام IGMP ای نباید از مرز یک LAN عبور کند. مقدار TTL برابر با 1 این تضمین را به وجود می‌آورد که IGMP ، محدوده LAN را ترک نمی‌کند زیرا که روتر بعدی پس از یک واحد کم کردن از TTL ، به مقدار صفر می‌رسد و لذا باید بسته را دور اندازد.

مهم: فیلد TTL یک بسته IP که پیغام IGMP را حمل می‌کند، برابر با 1 است.

❊ فیلد آدرس IP مقصد

جدول ۱۰-۲، آدرس IP مقصد برای هر نوع پیغام را نشان می‌دهد. یک « پیغام پرسشی » به آدرس multicast برابر با 224.0.0.1 (تمام سیستمهای متصل به این زیرشبکه، مطابق با جدول ۴-۵ فصل چهارم) ارسال می‌شود. همه کامپیوترها و همه روترها، این پیغام را دریافت می‌کنند.

www.Abiri.ir

Type	*IP Destination Address*
Query	224.0.0.1 All systems on this subnet
Membership Report	The multicast address of the group
Leave Report	224.0.0.2 All routers on this subnet

جدول ۱۰-۲: آدرسهای IP مقصد

یک « گزارش عضویت در گروه » ، به آدرس multicast گروهی که گزارش درباره آن گروه می‌باشد، ارسال می‌گردد (که به آن groupid نیز گفته می‌شود). هر وسیله‌ای (کامپیوتر یا روتر) که بسته را دریافت می‌کند، می‌تواند بلافاصله گروهی که درباره او، گزارش ارسال شده است را تعیین نماید (با بررسی هدر). همانطور که بحث آن قبلاً آمد، می‌توان تایمرهایی که هنوز به انقضاء مهلت مقرر نرسیده‌اند، اما مربوط به همان گروه هستند را خاموش کرد. لازم نیست ایستگاهها برای پیداکردن groupid ، بسته را بازکنند زیرا که groupid در دو محل در یک بسته نوشته شده است : الف. بخشی از خود پیغام IGMP ب. آدرس IP مقصد در هدر بسته IP . وقتی که groupid در دو محل نوشته شود، مانع از بروز خطا می‌شود.

یک پیغام « گزارش ترک گروه » به آدرس multicast برابر با 224.0.0.2 (همه روترها روی این زیرشبکه) ارسال می‌شود تا اینکه همه روترها، این پیغام را دریافت کنند. کامپیوترها نیز این پیغام را دریافت می‌کنند اما آن را دور می‌اندازند.

لایه دیتالینک

در لایه شبکه، پیغام IGMP در یک بسته IP کپسول‌بندی می‌شود و لذا به صورت یک بسته IP درمی‌آید. با این حال، از آنجا که آدرس مقصد این بسته IP، یک آدرس multicast است، پروتوکل ARP نمی‌تواند ***آدرس فیزیکی***[1] متناظر با این آدرس multicast را پیدا کند و آن را به لایه دیتالینک بدهد. حالا چه می‌شود؟ بستگی به این دارد که آیا پروتوکل لایه دیتالینک می‌تواند از آدرس فیزیکی multicast پشتیبانی کند یا خیر.

* با پشتیبانی از آدرس فیزیکی multicast

اکثر LAN ها از آدرس فیزیکی multicast پشتیبانی می‌کنند. یکی از آنها اترنت است. یک آدرس فیزیکی اترنت به طول شش بایت (48 بیت) است. اگر اولین (از سمت چپ) 25

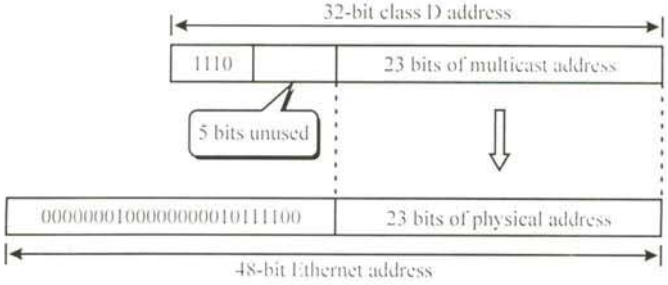
1- MAC address

www.Abiri.ir

بیت یک آدرس فیزیکی اترنت برابر با مقدار زیر باشد، یک آدرس فیزیکی multicast برای پروتوکل TCP/IP را نشان می‌دهد:

00000001000000000010111100

از 23=25-48 بیت باقیمانده، می‌توان برای تعریف یک گروه استفاده کرد. برای تبدیل یک آدرس IP multicast به یک آدرس فیزیکی اترنت، روتر multicast ، کم‌ارزشترین 23 بیت (از سمت راست) آدرس IP multicast ، که یک آدرس کلاس D است را استخراج می‌کند و آنها را به دنبال 25 بیتی که در بالا ذکر آن آمد، قرار می‌دهد و به این ترتیب یک آدرس 48 بیتی اترنت، معادل با آدرس IP multicast داده شده، ساخته می‌شود. شکل ۱۰-۱۰ را ببینید.



شکل ۱۰-۱۰: نگاشت آدرس IP multicast به یک آدرس فیزیکی اترنت

از آنجا که فقط 23 بیت سمت راست یک آدرس IP multicast کلاس D را به عنوان شاخص گروه در آدرس فیزیکی معادل، قرار دادیم لذا از 5 بیت این آدرس، استفاده‌ای نکرده‌ایم. (چهار بیت سمت چپ آن، 1110 است که معرف کلاس D است، بنابراین 5=23-4-32 بیت، باقی می‌ماند). لذا می‌توان گفت 2^5=32 آدرس IP multicast به یک آدرس multicast اترنت نگاشته می‌شوند. پس یک نگاشت چند به یک، و نَه یک به یک، خواهیم داشت.

اگر این 5 بیت برابر با تمام صفر نباشد، کامپیوتر ممکن است بسته‌هایی را دریافت کند که واقعاً برای گروهی که کامپیوتر عضوی از آن است، ارسال نشده باشند. پس صرف دریافت بسته‌های multicast در سطح لایه دیتالینک کافی نیست و کامپیوتر حتماً باید خود آدرس IP multicast بسته دریافتی را نیز چک کند و ببیند آیا بسته ارسالی، واقعاً برای گروهی است که کامپیوتر عضوی از آن است یا خیر. اگر مطابقت نداشته باشد، باید بسته دریافتی را دور بیندازد.

سایر LANها نیز از مفهوم مشابهی پشتیبانی می‌کنند ولی روشهای مختلفی برای نگاشت آدرس IP multicast به آدرس multicast فیزیکی دارند.

www.Abiri.ir

مهم: یک آدرس فیزیکی multicast اترنت در محدوده از 01:00:5E:00:00:00 تا 01:00:5E:7F:FF:FF است.

**** مثال ۲**

آدرس IP multicast برابر با 230.43.14.7 را به آدرس فیزیکی اترنت تبدیل کنید.

راه حل : می‌توانیم این کار را در دو مرحله انجام دهیم:

الف: 23 بیت سمت راست آدرس IP multicast را به صورت هگزادسیمال می‌نویسیم. برای این کار، سه بایت سمت راست را به صورت هگزادسیمال می‌نویسیم و سپس اگر رقم سمت چپ بزرگتر یا مساوی با 8 باشد، عدد 8 را از آن کم می‌کنیم. در مثال ما، نتیجه برابر با 2B:0E:07 است زیرا که سه بایت سمت راست برابر با 43.14.7 است که به صورت هگزادسیمال و گذاشتن « : » به جای « . » به صورت 2B:0E:07 درمی‌آید. رقم سمت چپ برابر با 2 است که از 8 کوچکتر می‌باشد.

ب: نتیجه قسمت الف را با آدرس پایه multicast در اترنت که برابر با 01:00:5E:00:00:00 است، جمع می‌کنیم. نتیجه برابر است با 01:00:5E:2B:0E:07

**** مثال ۳**

آدرس IP multicast برابر با 238.212.24.9 را به یک آدرس اترنت multicast تبدیل کنید.

راه حل: الف. سه بایت سمت راست برحسب هگزادسیمال برابر با D4:18:09 می‌باشد. از آنجا که رقم سمت چپ یعنی D از 8 بزرگتر است، باید 8 را از آن کم کنیم (معادل اعشاری D عدد 13 است و لذا 8 واحد از آن کم کنیم، عدد 5 باقی می‌ماند). نتیجه حاصل برابر است با 54:18:09

ب. نتیجه قسمت الف را با آدرس پایه multicast در اترنت، جمع می‌کنیم. نتیجه برابر است با 01:00:5E:54:18:09

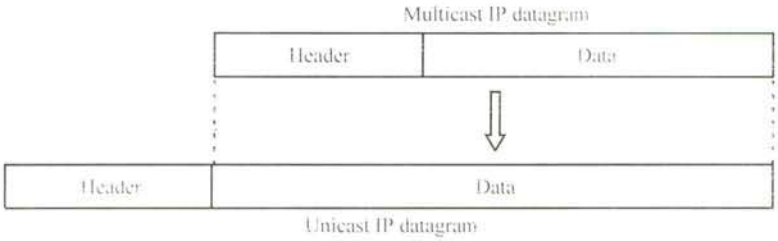
*** بدون پشتیبانی از آدرس فیزیکی multicast**

اکثر WAN ها از آدرس فیزیکی multicast پشتیبانی نمی‌کنند. برای ارسال یک بسته multicast در این شبکه‌ها، از پروسه‌ای به نام ***تونل زدن***[1] استفاده می‌شود. در این روش، بسته multicast در یک بسته تک‌گیرنده، کپسول‌بندی می‌شود و در میان شبکه حرکت می‌کند و

1- tunneling

www.Abiri.ir

زمانی که به آن سوی شبکه رسید، مجدداً به صورت یک بسته multicast ظاهر می‌شود (شکل ۱۰-۱۱ را ببینید).



شکل ۱۰-۱۱: تونل زدن

ابزار netstat

در فصلهای قبل، از برنامه netstat استفاده کردیم. در این بخش نیز، از آن استفاده می‌کنیم تا آدرسهای multicast ای که یک اینترفیس از آنها پشتیبانی می‌کند را بدست آوریم.

* مثال ۴

از netstat با سه گزینه n- ، r- و a- استفاده می‌کنیم. گزینه n- ، معادل عددی آدرسهای IP را می‌دهد، گزینه r- محتوای جدول مسیردهی را می‌دهد و گزینه a- همه آدرسها (تک‌گیرنده و چندگیرنده) را می‌دهد. دقت نمائید که فقط فیلدهایی که به بحث ما مربوط می‌شوند را نشان می‌دهیم.

```
S netstat -nra
Kernel IP routing table
```

Destination	Gateway	Mask	Flags	Iface
153.18.16.0	0.0.0.0	255.255.240.0	U	eth0
169.254.0.0	0.0.0.0	255.255.0.0	U	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	Lo
224.0.0.0	**0.0.0.0**	**224.0.0.0**	**U**	**eth0**
0.0.0.0	153.18.31.254	0.0.0.0	UG	eth0

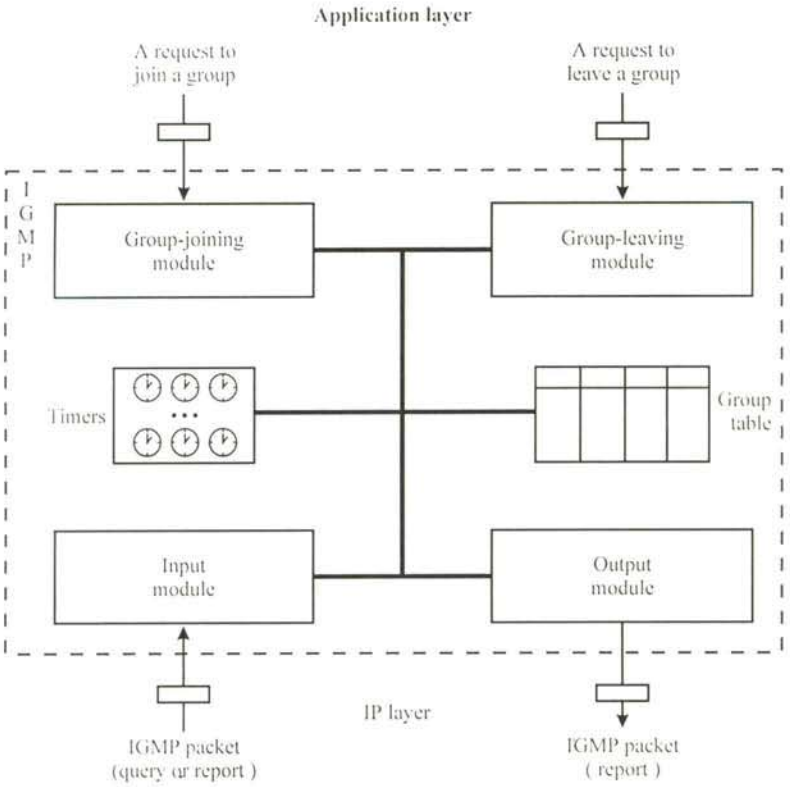
دقت نمائید که آدرس multicast را به صورت درشت نشان داده‌ایم. هر بسته‌ای با آدرس multicast از 224.0.0.0 تا 239.255.255.255 دریافت می‌شود و به اینترفیس اترنت داده می‌شود.

www.Abiri.ir

۱۰-۵ نرم افزار IGMP

به کمک نرم‌افزار IGMP ساده خودمان می‌خواهیم نشان دهیم که چگونه IGMP می‌تواند بسته‌های IGMP را ارسال و دریافت کند.

این نرم‌افزار فقط ماجولهایی را نشان می‌دهد که در یک کامپیوتر IGMP به کار می‌روند. روتر IGMP را به عنوان یک تمرین باقی می‌گذاریم. در این نرم‌افزار IGMP ، یک جدول از گروهها، یک تعداد تایمر و چهار ماجول نرم‌افزاری داریم: ***ماجول ملحق شدن به گروه***[1] ، ***ماجول ترک گروه***[2] ، ***ماجول ورودی***[3] و ***ماجول خروجی***[4] . شکل ۱۰-۱۲ این شش جزء و رابطه بین آنها را نشان می‌دهد.



شکل ۱۰-۱۲: نرم‌افزار IGMP

1- group joining module
2- group leaving module
3- input module
4- output module

www.Abiri.ir

جدول گروهها

جدول گروهها[1] اطلاعاتی درباره یک گروه (یک آدرس multicast) می‌دهد که حداقل یک برنامه، عضو آن است. این جدول، حاوی چهار فیلد است: ***حالت***[2]، ***شماره اینترفیس***[3]، ***آدرس گروه***[4] و ***تعداد علاقه‌مندان***[5]. شکل ۱۰-۱۳ را ببینید.

State	Interface No.	Group Address	Reference Count
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....

شکل ۱۰-۱۳: جدول گروه

State : حالت . این فیلد، حالت یا وضعیت المان جدول را نشان می‌دهد و می‌تواند یکی از مقادیر زیر را داشته باشد: ***آزاد***[6]، ***به تأخیر افتاده***[7] یا ***بی‌کار***[8]. وقتی که حالت، برابر با FREE است، هیچ برنامه‌ای عضو این گروه نیست. وقتی که حالت، برابر با DELAYING است، باید پس از انقضاء مهلت مقرر تایمر، یک « گزارش عضویت در گروه » درباره این المان، ارسال شود. اگر حالت برابر با IDLE است، تایمری برای این المان در حال کار نیست.

Interface number : شماره اینترفیس. این فیلد، اینترفیسی را تعریف می‌کند که از طریق آن، بسته multicast ارسال و دریافت می‌شود.

Group address : آدرس گروه. این یک آدرس multicast است و گروه را تعریف می‌کند.

Reference count : تعداد علاقه‌مندان. تعداد برنامه‌هایی را نشان می‌دهد که هنوز به گروه علاقه‌مند هستند. هر وقت که برنامه‌ای به گروه ملحق می‌شود، یک واحد به این تعداد اضافه می‌شود. هر زمان که برنامه‌ای این گروه را ترک می‌کند، یک واحد از این تعداد کم می‌شود. وقتی که تعداد به صفر رسید، حالت المان جدول به FREE تبدیل می‌شود.

1- group table
2- state
3- interface number
4- group address
5- reference count
6- FREE
7- DELAYING
8- IDLE

www.Abiri.ir

تایمرها

هر المان جدول، در حالت DELAYING ، تایمری دارد که حاکم بر ارسال گزارشهاست. هر تایمر، یک زمان انقضاء دارد که به طور تصادفی انتخاب شده است تا مانع از ارسال انبوهی از گزارشات شود. وقتی که تایمر، به انقضاء مهلت مقرر می‌رسد، سیگنالی به ماجول خروجی داده می‌شود که باید یک گزارش ارسال نماید.

ماجول ملحق شدن به گروه

برنامه‌ای که می‌خواهد به گروهی ملحق شود، ***ماجول ملحق شدن به گروه***[1] را صدا می‌زند. این ماجول، در جدول گروهها، به دنبال المانی با آدرس multicast برابر با آدرس گروه مورد نظر، می‌گردد. اگر پیدا کرد، یک واحد به مقدار reference count اضافه می‌کند تا نشان دهد که برنامه دیگری به این گروه ملحق شده است. اگر آدرس multicast گروه مورد نظر، در جدول گروهها پیدا نشد، این ماجول، یک المان جدید را در جدول ایجاد می‌کند و مقدار reference count را برابر با 1 قرار می‌دهد. در این حالت، این ماجول، هم به ماجول خروجی و هم به لایه دیتالینک پیغامی می‌فرستد. او به ماجول خروجی می‌گوید که باید یک « گزارش عضویت در گروه » ارسال کند و به لایه دیتالینک می‌گوید که باید جدول پیکربندی خود را به روزرسانی کند تا بتواند این نوع بسته multicast را نیز دریافت کند.

*** جریان رویدادهای ماجول ملحق شدن به گروه**

ورودی: درخواست یک برنامه برای ملحق شدن به یک گروه

۱- به دنبال المان متناظر با این گروه در جدول می‌گردد.

۲- اگر پیدا کرد:

۱- مقدار reference count را یک واحد افزایش می‌دهد.

۳- اگر پیدا نکرد:

۱- المانی با reference count برابر با 1 تولید می‌کند.

۲- المان را به جدول اضافه می‌کند.

۳- از ماجول خروجی می‌خواهد که یک « گزارش عضویت در گروه » ارسال کند.

۴- به لایه دیتالینک خبر می‌دهد که جدول پیکربندی خود را به روزرسانی کند.

1- group joining module

www.Abiri.ir

ماجول ترک گروه

برنامه‌ای که می‌خواهد گروهی را ترک کند، ***ماجول ترک گروه***[1] را صدا می‌زند. این ماجول در جدول گروهها به دنبال المانی با آدرس multicast مشابه با گروه مورد نظر می‌گردد. اگر چنین المانی را پیدا کرد، یک واحد از مقدار reference count کم می‌کند. اگر مقدار جدید آن صفر شد، حالت آن المان به FREE تبدیل می‌شود و یک « گزارش ترک گروه » ارسال می‌شود.

❊ **جریان رویدادهای ماجول ترک گروه**

ورودی: درخواست از سوی برنامه‌ای برای ترک گروه

۱- به دنبال المان متناظر در جدول می‌گردد.

۲- اگر پیدا کرد :

۱- از مقدار reference count یک واحد کم می‌کند.

۲- اگر مقدار reference count برابر با صفر است :

۱- اگر تایمری برای این المان در حال کار است :

۱- تایمر را خاموش می‌کند.

۲- حالت المان را به FREE تبدیل می‌کند.

۳- از ماجول خروجی می‌خواهد که یک « گزارش ترک گروه » را ارسال کند.

۳- برمی‌گردد.

ماجول ورودی

یک پیغام IGMP ، ***ماجول ورودی***[2] را صدا می‌زند. اگر پیغام دریافتی، یک « پیغام پرسشی » است، ماجول ورودی، تایمری را برای هر المانی در جدول گروهها که حالت IDLE دارد، راه می‌اندازد و حالت آنها را به DELAYING تغییر می‌دهد. ماجول برای انجام این کار، یک عدد تصادفی بین صفر و ماکزیم تأخیر، تولید می‌کند، تایمری را روی این مقدار، ست می‌کند و سپس آن را راه‌اندازی می‌نماید. به منظور آنکه ماجول خروجی، گزارشهایش را در زمانهای مختلف ارسال کند، تا ازدحام پیش نیاید، لازم به استفاده از مولد عدد تصادفی است.

اگر پیغام دریافتی، یک پیغام « گزارش عضویت در گروه » باشد، ماجول ورودی به دنبال المانی متناظر با آن در جدول گروهها می‌گردد. اگر پیدا کرد و حالت آن DELAYING

1- group leaving module
2- input module

www.Abiri.ir

است، به این معناست که کامپیوتر دیگری در شبکه، یک « گزارش عضویت در گروه » را برای این گروه ارسال کرده است و لذا دیگر ضرورتی ندارد که این کامپیوتر نیز گزارش دیگری ارسال نماید. بنابراین ماجول ورودی تایمر را خاموش می‌کند و حالت آن المان را به IDLE تغییر می‌دهد. به خاطر داشته باشید که کامپیوترها، پیغامهای « گزارش ترک گروه » را دریافت نمی‌کنند.

*** جریان رویدادهای ماجول ورودی**

ورودی : یک پیغام IGMP

۱- نوع پیغام را چک می‌کند

۲- اگر پیغام از نوع « پیغام پرسشی » است :

۱- برای هر المانی در جدول با حالت IDLE ، تایمری را راه‌اندازی می‌کند.

۲- حالت IDLE آن المان را به حالت DELAYING تغییر می‌دهد.

۳- برمی‌گردد.

۳- اگر پیغام از نوع پیغام « گزارش عضویت در گروه » است :

۱- به دنبال المان متناظر با آن در جدول گروهها می‌گردد.

۲- اگر پیدا کرد و حالت آن DELAYING است :

۱- تایمر را برای این المان، خاموش می‌کند.

۲- حالت آن المان را به IDLE تغییر می‌دهد.

۴- برمی‌گردد.

ماجول خروجی

اگر تایمری به انقضاء مهلت مقرر رسیده باشد یا درخواستی برای ملحق شدن به یک گروه آمده باشد و یا درخواستی برای ترک گروهی صادر شده باشد، ماجول خروجی فراخوانی می‌شود. سپس به دنبال المان متناظر با آن گروه می‌گردد و اگر حالت آن DELAYING باشد، یک گزارش ایجاد می‌کند، آن را ارسال می‌نماید و سپس حالت آن المان را به IDLE تغییر می‌دهد.

*** جریان رویدادهای ماجول خروجی**

ورودی : سیگنالی از یک تایمر یا درخواستی از طرف ماجول ملحق شدن به گروه یا ماجول ترک گروه

۱- اگر پیغام از سوی یک تایمر آمده باشد :

www.Abiri.ir

۱- اگر المان متناظر با آن در جدول پیدا شد و حالت آن DELAYING است :
۱- یک « گزارش عضویت در گروه » ایجاد می‌کند.
۲- حالت المان را به IDLE تغییر می‌دهد.
۲- اگر پیغام از سوی ماجول ملحق شدن به گروه آمده باشد :
۱- یک « گزارش عضویت در گروه » ایجاد می‌کند.
۳- اگر پیغام از سوی ماجول ترک گروه آمده باشد :
۱- یک « گزارش ترک گروه » ایجاد می‌کند.
۴- پیغام را می‌فرستد.
۵- برمی‌گردد.

۱۰-۶ عبارات کلیدی

استراتژی پاسخ با تأخیر (delayed response strategy)
پیغام پرسش عمومی (general query message)
عضویت گروه (group membership)
شناسه گروه (groupid)
پروتوکل مدیریت گروه در اینترنت
Internet Group Management Protocol : IGMP
گزارش ترک گروه (leave report)
گزارش عضویت در گروه (membership report)
آدرس چند گیرندگی (multicast address)
روتر چند گیرندگی (multicast router)
چند گیرندگی (multicasting)
پیغام پرسشی (query message)
روتر پرسش کننده (query router)
پیغام پرسش خاص (special query message)
تونل زدن (tunneling)

۱۰-۷ خلاصه

- چند گیرندگی، ارسال یک پیغام مشابه برای بیش از یک گیرنده به طور همزمان است.
- IGMP به روترهای multicast کمک می‌کند تا لیستی از اعضای مرتبط با یک اینترفیس روتر را تولید کنند.

www.Abiri.ir

- سه نوع پیغام IGMP عبارتند از : پیغام پرسشی، گزارش عضویت در گروه و گزارش ترک گروه.
- IGMP به صورت محلی کار می‌کند.
- یک کامپیوتر یا روتر می‌تواند عضو یک گروه باشد.
- یک کامپیوتر، لیستی از برنامه‌ها را نگه می‌دارد که عضوی از یک گروه هستند.
- یک روتر یا کامپیوتر، یک « گزارش عضویت در گروه » برای ملحق شدن به یک گروه می‌فرستد.
- یک روتر یا کامپیوتر، یک « گزارش ترک گروه » برای ترک یک گروه ارسال می‌کند.
- یک روتر، یک « پیغام پرسش عمومی » ارسال می‌کند تا عضویت در گروه را چک کند.
- استراتژی پاسخ با تأخیر، مانع از بروز ترافیک غیر ضروری در یک LAN می‌شود.
- پیغام IGMP در یک بسته IP کپسول‌بندی می‌شود.
- اکثر LAN ها شامل اترنت از آدرس فیزیکی multicast پشتیبانی می‌کنند.
- WAN هایی که از آدرس فیزیکی چندگیرنده پشتیبانی نمی‌کنند نمی‌توانند از روشی به نام تونل زدن برای ارسال بسته‌های multicast استفاده کنند.
- یک نرم‌افزار IGMP می‌تواند شامل یک جدول گروه، یک تعداد تایمر و چهار ماجول نرم‌افزاری باشد : یک ماجول ورودی، یک ماجول خروجی، یک ماجول ملحق شدن به گروه و یک ماجول ترک گروه.

۱۰-۸ مجموعه تمرین‌ها

تمرین‌ها

۱- چرا لزومی ندارد که یک پیغام IGMP ، به خارج از شبکه خودش برود ؟

۲- لیست گروه‌های یک روتر multicast حاوی چهار گروه است (z,y,x,w) . سه کامپیوتر متصل به LAN وجود دارد. کامپیوتر A ، سه عضو در گروه w و یک عضو در گروه x دارد. کامپیوتر B ، دو عضو در گروه w و یک عضو در گروه y دارد. کامپیوتر C ، هیچ برنامه‌ای متعلق به هیچ گروهی ندارد. پیغامهای IGMP ای که برای چک کردن عضویت گروهها ارسال می‌شوند را نشان دهید.

۳- آدرس multicast برای یک گروه 231.24.60.9 است. آدرس 48 بیتی اترنت آن برای LAN ای که از TCP/IP استفاده می‌کند، چیست؟

۴- اگر روتری 20 المان در جدول گروه خود داشته باشد، آیا به طور پریودیک باید 20 پرسش مختلف ارسال کند یا فقط یکی کافی است؟ پاسخ خود را توضیح دهید.

www.Abiri.ir

۵- اگر کامپیوتری بخواهد عضویت خود را در پنج گروه حفظ نماید، آیا باید پنج پیغام مختلف « گزارش عضویت در گروه » بفرستد و یا یکی کافی است؟

۶- یک روتر با آدرس IP برابر با 202.45.33.21 و آدرس اترنت فیزیکی 23:4A:45:12:EC:D1 یک پیغام پرسش عمومی IGMP ارسال می‌کند. تمام المانهای موجود در پیغام را نشان دهید.

۷- پیغام تمرین ۶ را در یک بسته IP کپسول‌بندی کنید. تمام فیلدها را پر کنید.

۸- پیغام تمرین ۷ را در یک فریم اترنت کپسول‌بندی کنید. تمام فیلدها را پر کنید.

۹- یک کامپیوتر با آدرس IP برابر با 124.15.13.1 و آدرس اترنت فیزیکی 4A:22:45:12:E1:E2 یک پیغام « گزارش عضویت در گروه » درباره گروهی به آدرس multicast برابر با 228.45.23.11 ارسال می‌کند. تمام المانهای موجود در پیغام را نشان دهید.

۱۰- پیغام تمرین ۹ را در یک بسته IP کپسول‌بندی کنید و همه فیلدها را پر کنید.

۱۱- پیغام تمرین ۱۰ را در یک فریم اترنت کپسول‌بندی کنید. همه فیلدها را پر کنید.

۱۲- یک روتر متصل به یک شبکه اترنت، یک بسته IP چندگیرنده با groupid (یا آدرس multicast گروه) برابر با 226.17.18.4 را دریافت کرده است. وقتی که کامپیوتر جدول گروه خود را چک می‌کند، این آدرس را پیدا می‌کند. نشان دهید که روتر چگونه این بسته را با کپسول‌بندی بسته IP در یک فریم اترنت، برای گیرندگانش می‌فرستد. همه المانهای فریم اترنت را نشان دهید. آدرس IP خروجی روتر برابر با 185.23.5.6 و آدرس فیزیکی خروجی آن برابر با 4A224512E1E2 است. آیا این روتر به سرویسهای ARP نیاز دارد؟

۱۳- اگر روتر تمرین ۱۲ نتواند groupid گروه مورد نظر را در جدولش پیدا کند، چه رخ می‌دهد ؟

۱۴- تمرین ۱۲ را با یک شبکه فیزیکی که از آدرس‌دهی فیزیکی multicast پشتیبانی نمی‌کند، تکرار کنید.

۱۵- یک کامپیوتر با آدرس IP برابر با 114.45.7.9 ، یک پرسش IGMP را دریافت می‌کند. وقتی که جدول گروه خود را چک می‌کند، هیچ المانی را پیدا نمی‌کند. کامپیوتر چه عملی باید انجام دهد؟ آیا پیغامی باید بفرستد؟ اگر چنین است، فیلدهای بسته را نشان دهید.

۱۶- کامپیوتری با آدرس IP برابر با 222.5.7.19 ، یک پرسش IGMP دریافت می‌کند. وقتی که جدول مسیردهی خود را چک می‌کند، دو المان را در جدولش پیدا می‌کند: 227.4.3.7 و 229.45.6.23 . کامپیوتر چه عملی باید انجام دهد؟ آیا باید پیغامی ارسال کند؟ اگر چنین است، چه نوع پیغامی و چند تا ؟ فیلدها را نشان دهید.

۱۷- کامپیوتری با آدرس IP برابر با 186.4.77.9 ، درخواستی از برنامه‌ای مبنی بر ملحق شدن آن به گروهی با groupid برابر با 230.44.101.34 دریافت می‌کند. وقتی که کامپیوتر جدول

www.Abiri.ir

گروه خود را چک می‌کند، هیچ المانی برای این groupid پیدا نمی‌کند. کامپیوتر باید چه عملی انجام دهد؟ آیا باید پیغامی ارسال کند؟ اگر چنین است ، فیلدهای بسته را نشان دهید.

۱۸- تمرین ۱۷ را در حالتی تکرار کنید که کامپیوتر المانی را در جدول خود پیدا کند.

۱۹- یک روتر با آدرس IP برابر با 184.4.7.9 گزارشی از کامپیوتر دریافت می‌کند که می‌خواهد به یک گروه با groupid برابر با 232.54.10.34 ملحق شود. وقتی که روتر، جدول گروه خود را چک می‌کند، المانی برای این groupid پیدا نمی‌کند. روتر چه عملی باید انجام دهد؟ آیا باید پیغامی ارسال کند؟ اگر چنین است، فیلدهای بسته را نشان دهید.

۲۰- تمرین ۱۹ را برای حالتی که روتر، المانی را در جدول پیدا کند، تکرار کنید.

۲۱- روتر، یک پرسش را ارسال می‌کند و فقط سه گزارش درباره groupid های 225.4.6.7 ، 225.32.56.8 و 226.34.12.9 دریافت می‌کند. وقتی که جدول مسیردهی خود را چک می‌کند، پنج المان پیدا می‌کند : 225.4.6.7 ، 225.11.6.8 ، 226.34.12.9 ، 226.23.22.67 و 229.12.4.89 . چه عملی باید انجام شود ؟

۲۲- محتوای یک پیغام IGMP بر حسب هگزادسیمال برابر است با:

11 00 EE FF 00 00 00 00

به سؤالات زیر پاسخ دهید:

الف- نوع پیغام چیست؟

ب- مقدار checksum چقدر است؟

ج-groupid چند است؟

۲۳-محتوای یک پیغام IGMP بر حسب هگزادسیمال برابر است با :

16 00 F9 C0 E1 2A 13 14

به سؤالات زیر پاسخ دهید:

الف) نوع پیغام چیست؟

ب) مقدار checksum چقدر است؟

ج) groupid چند است؟

۲۴- آیا در نمایش هگزادسیمال پیغام IGMP زیر، خطائی وجود دارد؟

11 00 A0 11 E1 2A 13 14

۲۵- آیا در نمایش هگزادسیمال پیغام IGMP زیر خطائی وجود دارد؟

17 00 A0 11 00 00 00 00

۲۶- اترنت از چند آدرس multicast برای پروتوکل IP می‌تواند پشتیبانی کند؟

۲۷- پروتوکل IP از چند آدرس multicast می‌تواند پشتیبانی کند؟

۲۸- وقتی که یک آدرس IP multicast را به یک آدرس multicast اترنت تبدیل می‌کنیم، سایز فضای آدرس از دست رفته چیست؟

www.Abiri.ir

۲۹- آدرسهای IP multicast را به آدرسهای multicast اترنت تبدیل کنید. چند تا از آنها، آدرس اترنتی مناسبی می‌باشند؟

الف) 224.18.72.8

ب) 235.18.72.8

ج) 237.18.6.88

د) 224.88.12.8

فعالیتهای تحقیقاتی

۳۰- نرم‌افزار IGMP موجود در متن را چنان تغییر دهید که بتوان از آن در یک روتر نیز استفاده کرد.

۳۱- درباره IGMPv1 کمی تحقیق کنید. سایز فیلد type چیست؟ آیا فیلدی در نسخه اول این پروتوکل هست که در نسخه دوم آن نیست؟ آیا نسخه‌های اول و دوم سازگار هستند؟ اگر روتری که از نسخه دوم پشتیبانی می‌کند، پیغامی را تحت نسخه اول دریافت کند، روتر چه کار می‌تواند بکند ؟ اگر روتری که از نسخه اول پشتیبانی می‌کند، پیغامی تحت نسخه دوم دریافت کند، روتر چه کار می‌تواند بکند ؟

۳۲-با استفاده از netstat ببینید آیا سرور از آدرس‌دهی multicast پشتیبانی می‌کند یا خیر؟

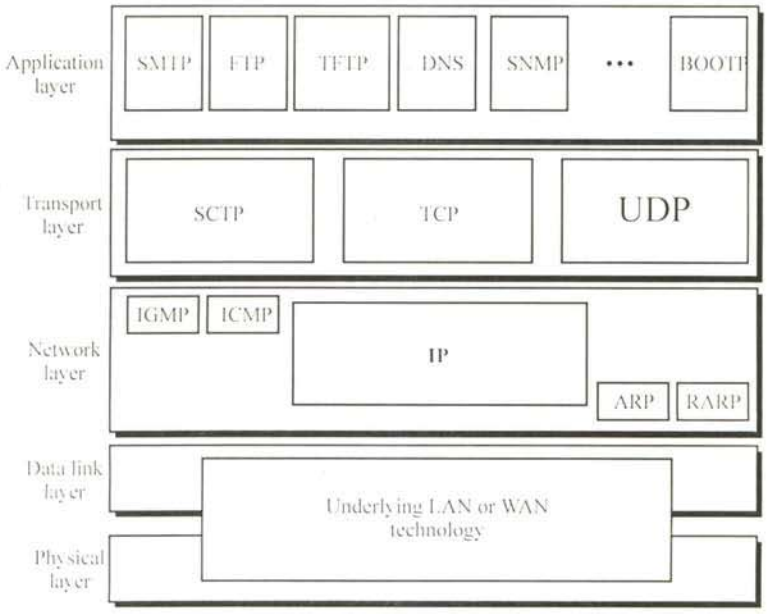
۳۳-RFCهای مرتبط با پروتوکلهای IGMP را پیدا کنید.

www.Abiri.ir

فصل یازدهم

پروتوکل UDP

TCP/IP در ابتدا، دو پروتوکل را برای لایه انتقال تعریف کرده بود : UDP و TCP . قبل از آنکه در فصل دوازدهم درباره TCP صحبت کنیم، ابتدا درباره UDP که ساده‌تر است، صحبت می‌کنیم. در فصل سیزدهم درباره یک پروتول جدید لایه انتقال به نام SCTP صحبت خواهیم کرد. شکل ۱۱-۱ رابطه UDP با سایر پروتوکلها و لایه‌های پشته پروتوکل TCP/IP را نشان می‌دهد. مکان UDP بین لایه کاربرد و لایه IP است و مشابه TCP ، می‌توان آن را رابطی بین برنامه‌های کاربردی و عملیات شبکه، در نظر گرفت.



شکل ۱۱-۱ : جایگاه UDP در پشته پروتوکل TCP/IP

یک پروتوکل لایه انتقال معمولاً مسئولیتهای متعددی دارد. یکی این است که یک مخابره برنامه به برنامه (برنامه یا **پروسه**[1] ، یک نرم افزار در حال اجرا است) برقرار کند. UDP

1- Process

www.Abiri.ir

برای انجام این کار، از شماره پورت استفاده می‌کند. مسئولیت دیگر پروتوکل لایه انتقال، برقراری مکانیزمهای کنترلی در لایه انتقال است. UDP این کار را در سطح بسیار پایینی انجام می‌دهد زیرا که UDP هیچ مکانیزمی برای کنترل فلو ندارد و هیچ تأییدیه‌ای برای بسته‌های دریافتی ارسال نمی‌کند. با این حال UDP تا حدودی، خطا را کنترل می‌کند. به این ترتیب که اگر UDP در بسته دریافتی خطائی را آشکار کند، آن بسته را دور می‌اندازد.

همچنین لایه انتقال، یک مکانیزم کنترلی را برای برنامه‌ها فراهم می‌نماید. برنامه‌ها باید بتوانند ***دنباله‌ای از دیتا***[1] را برای لایه انتقال ارسال نمایند. این وظیفه لایه انتقال کامپیوتر فرستنده است که اتصالی را با لایه انتقال کامپیوتر گیرنده برقرار نماید، دنباله دیتا را به واحدهای قابل انتقال تقسیم نماید، آنها را شماره‌گذاری کند و سپس یکی یکی آنها را ارسال نماید. وظیفه لایه انتقال کامپیوتر گیرنده این است که صبر کند تا تمام واحدهای دیتای متعلق به یک برنامه از راه برسند، آنها را از نظر وقوع خطا بررسی نماید و سپس آنها که فاقد خطا هستند را به صورت دنباله‌ای از دیتا، به برنامه گیرنده تحویل دهد. لایه انتقال پس از آنکه کل دنباله دیتا ارسال شد، اتصال را می‌بندد. UDP هیچکدام از کارهای فوق را انجام نمی‌دهد. UDP فقط می‌تواند یک واحد دیتا را از برنامه فرستنده بگیرد و سپس آن را به صورت غیر قابل اطمینان به برنامه گیرنده تحویل دهد. برای آنکه واحد دیتا در یک دیتاگرام UDP جا شود، باید آن واحد دیتا، به اندازه کافی کوچک باشد.

UDP یک پروتوکل فاقد اتصال و غیر قابل اطمینان لایه انتقال است. او صرفاً به جای مخابره ماشین به ماشین، یک مخابره برنامه به برنامه را برقرار می‌کند و به غیر از این، چیزی به سرویسهای IP اضافه نمی‌کند. همچنین UDP قابلیت خیلی محدودی برای خطایابی دارد.

اگر UDP اینقدر ضعیف است، چرا ممکن است برنامه‌ای بخواهد از آن استفاده کند؟ با هر نقیصه‌ای، فایده‌ای نیز همراه است. UDP یک پروتوکل بسیار ساده با حداقل اطلاعات بالاسری است. اگر برنامه‌ای بخواهد پیغام کوچکی را ارسال نماید و توجه چندانی به قابلیت اطمینان نیز نداشته باشد، می‌تواند از UDP استفاده کند. در ارسال پیغامهای کوچک، عملیات متقابل لازم بین فرستنده و گیرنده برای UDP، کمتر از TCP است.

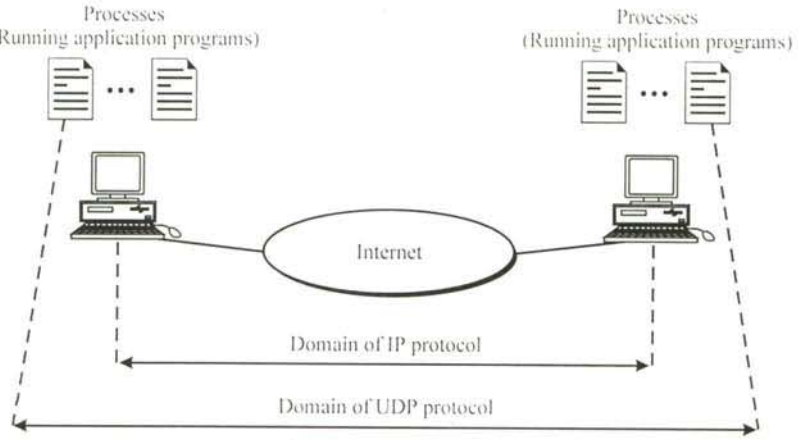
۱۱-۱ مخابره برنامه به برنامه

قبل از بررسی UDP ، ابتدا باید مفهوم مخابره ماشین به ماشین و مفهوم مخابره برنامه به برنامه و نیز تفاوت بین آنها را درک کنیم.

1- Stream

www.Abiri.ir

IP مسئول مخابره دیتا در سطح لایه شبکه (مخابره ماشین به ماشین) است. IP به عنوان یک پروتوکل لایه شبکه، فقط می‌تواند پیغام را به ماشین مقصد تحویل دهد. اما این کار، کامل نیست زیرا که پیغام باید به دست پروسه یا برنامه گیرنده برسد. در اینجاست که UDP به عنوان یک پروتوکل لایه انتقال وارد عمل می‌شود. UDP مسئول تحویل پیغام به برنامه مناسب است. شکل ۱۱-۲ حوزه عمل IP و UDP را نشان می‌دهد.



شکل ۱۱-۲ : حوزه عمل UDP و IP

شماره پورت

گرچه راههای متعددی برای برقراری مخابره برنامه به برنامه وجود دارد، متداولترین آن، از نوع ***سرویس‌گیرنده / سرویس‌دهنده***[1] است. برنامه‌ای روی ماشین مبدأ (یا ماشین محلی) به نام سرویس‌گیرنده، به سرویسهای برنامه‌ای روی ماشین مقصد (یا ماشین دوردست) به نام سرویس‌دهنده یا سرور، نیاز دارد.

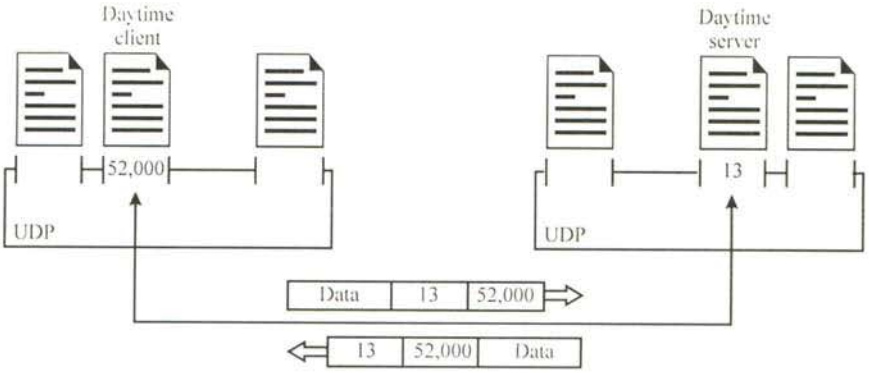
هر دو برنامه (سرویس‌گیرنده و سرویس‌دهنده) نام مشابهی دارند. به عنوان مثال برای گرفتن روز و ساعت از یک ماشین دور دست، به برنامه سرویس‌گیرنده Daytime روی ماشین مبدأ و برنامه سرور Daytime روی ماشین مقصد، نیاز داریم.

با این حال، سیستم عاملهای امروزی، هم از ***محیطهای چندکاربره***[2] و هم از ***محیطهای چند کاره***[3] پشتیبانی می‌کنند. یک کامپیوتر دوردست می‌تواند به طور همزمان،

1- client/server
2- multiuser environments
3- multiprogramming environments

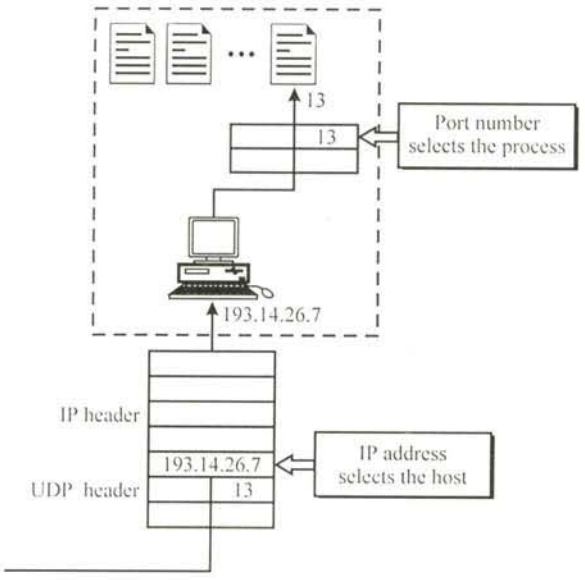
www.Abiri.ir

www.Abiri.ir



شکل ۱۱-۳ : شماره پورت

تاکنون باید روشن شده باشد که آدرسهای IP و شماره پورتها، در انتخاب مقصد نهائی دیتا، نقشهای متفاوتی را ایفاء می‌کنند. آدرس IP مقصد، یک کامپیوتر را مابین همه کامپیوترهای دنیا، مشخص می‌کند. بعد از آنکه کامپیوتر مشخص شد، شماره پورت، یکی از برنامه‌هایی که روی این کامپیوتر خاص در حال اجرا است را مشخص می‌کند (شکل ۱۱-۴ را ببینید.

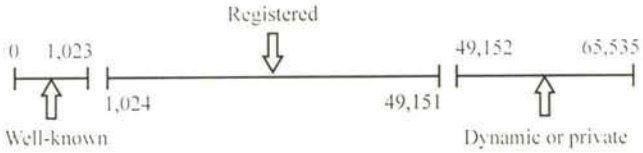


شکل ۱۱-۴ : آدرسهای IP در مقایسه با شماره پورتها

www.Abiri.ir

تقسیم‌بندی شماره پورتها توسط ICANN

ICANN شماره پورتها را به سه دسته تقسیم کرده است: ***کاملاً شناخته شده***[1]، ***ثبت شده***[2] و ***دینامیک***[3] یا ***خصوصی***[4] (شکل ۱۱-۵ را ببینید).



شکل ۱۱-۵ : تقسیم‌بندی شماره پورتها توسط ICANN

پورتهای کاملاً شناخته شده : کنترل و تخصیص پورتهای از 0 تا 1023 توسط ICANNصورت می‌گیرد. اینها پورتهای کاملاً شناخته شده هستند.

پورتهای ثبت شده : کنترل و تخصیص پورتهای از 1024 تا 49151 با ICANN نیست. اما می‌توان به منظور امتناع از تعاریف چندگانه برای این پورتها، آنها را درICANN برای کاربرد مورد نظر، به ثبت رساند.

پورتهای دینامیک یا خصوصی : پورتهایی از 49152 تا 65535 نَه کنترل می‌شوند و نَه ثبت می‌شوند. می‌توان از آنها به عنوان پورتهای موقتی یا خصوصی استفاده کرد. توصیه اولیه این بود که از این محدوده برای شماره پورتهای موقتی سرویس‌گیرنده‌ها استفاده شود. با این وجود اکثر سیستمها، از این توصیه پیروی نمی‌کنند.

مهم: شماره پورتهای کاملاً شناخته شده از 1024 کوچکتر هستند.

* شماره پورتهای کاملاً شناخته شده برایUDP

جدول ۱۱- ۱ برخی از شماره پورتهای کاملاً شناخته شده‌ای را نشان می‌دهد که UDP از آنها استفاده می‌کند. برخی از شماره پورتها را هم توسط UDP و هم توسط TCP می‌توان به کار برد. وقتی که در فصل دوازدهم درباره TCP صحبت کردیم، درباره آنها بحث خواهیم کرد.

1- well-known
2- registered
3- dynamic
4- private

www.Abiri.ir

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the data and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Nameserver	Domain Name Service
67	Bootps	Server port to download bootstrap information
68	Bootpc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

جدول ۱۱-۱ : شماره پورتهای کاملاً شناخته شده و مورد استفاده UDP

**** مثال ۱**

در یونیکس، لیست شماره پورتهای کاملاً شناخته شده در فایلی به نام /etc/services نگهداری می‌شود. هر سطر این فایل، نام سرور و شماره پورت کاملاً شناخته شده متناظر با آن را نشان می‌دهد. می‌توانیم از برنامه grep برای استخراج سطر متناظر با کاربرد مورد نظر، استفاده کنیم. نمونه زیر، شماره پورت TFTP را نشان می‌دهد. دقت نمایید که TFTP می‌تواند از پورت شماره 69 ، هم با TCP و هم با UDP ، استفاده کند.

```
$grep   tftp /etc/services
tftp            69/tcp
tftp            69/udp
```

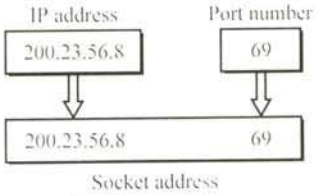
همانطور که در فصل بیست و یکم خواهیم دید، SNMP می‌تواند از دو شماره پورت (161 و 162) ، هر کدام برای یک کاربرد متفاوت، استفاده کند.

```
$grep   snmp/etc/services
snmp            161/tcp         #Simple Net Mgmt Proto
snmp            161/udp         #Simple Net Mgmt Proto
snmptrap        162/udp         #Traps for SNMP
```

www.Abiri.ir

آدرسهای سوکت

همانطور که دیدیم UDP برای برقراری یک اتصال، در هر دو طرف، به آدرس IP و شماره پورت نیاز دارد. ***آدرس سوکت***[1] از ترکیب یک آدرس IP و یک شماره پورت بدست می‌آید. آدرس سوکت سرویس‌گیرنده، به طور منحصر به فردی سرویس‌گیرنده را مشخص می‌کند و آدرس سوکت سرور، به طور منحصر به فردی، سرور را مشخص می‌کند (شکل ۱۱-۶ را ببینید).



شکل ۱۱-۶ : آدرس سوکت

برای استفاده از سرویسهای UDP ، به یک جفت آدرس سوکت نیاز داریم: آدرس سوکت سرویس‌گیرنده و آدرس سوکت سرور. این چهار تکه اطلاعات، بخشی از هدر IP و هدر UDP هستند. هدر IP حاوی آدرسهای IP است و هدر UDP حاوی شماره پورتهاست.

۱۱-۲ دیتاگرام UDP

واحدهای دیتای UDP که به نام ***دیتاگرامهای کاربر***[2] موسومند، هدری به طول ثابت 8 بایت دارند. شکل ۱۱-۷ فرمت یک دیتاگرام کاربر را نشان می‌دهد.

فرمت هدر دیتاگرام UDP

هدر دیتاگرام UDP از چهار فیلد زیر ساخته شده است:

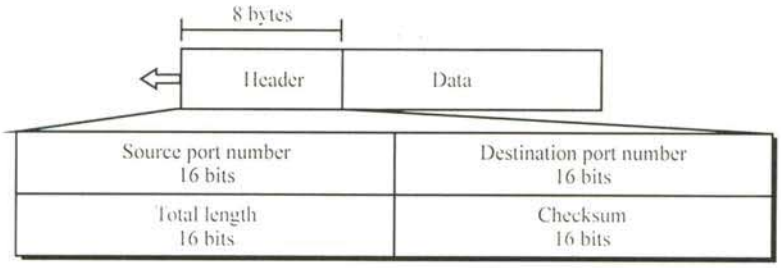
* **Source port number : شماره پورت مبدأ**

این شماره پورتی است که برنامه در حال اجرا روی ماشین مبدأ از آن استفاده می‌کند. طول آن 16 بیت است و لذا شماره پورت می‌تواند عددی از 0 تا 65535 باشد. اگر ماشین مبدأ، سرویس‌گیرنده‌ای باشد که درخواستی را ارسال کرده است، شماره پورت در اکثر حالات، یک

1- socket address
2- user datagrams

www.Abiri.ir

شماره پورت موقتی خواهد بود که برنامهٔ سرویس‌گیرنده آن را درخواست کرده است و نرم افزار UDPای که روی ماشین مبدأ در حال اجراست، آن شماره پورت را برایش انتخاب کرده است. اگر ماشین مبدأ، سرور باشد (سروری که درخواستی را ارسال می‌کند)، شماره پورت در اکثر حالات، یک شماره پورت کاملاً شناخته شده خواهد بود.



شکل ۱۱-۷ : فرمت دیتاگرام کاربر

❋ Destination port number : شماره پورت مقصد

این شماره پورتی است که برنامه در حال اجرا روی ماشین مقصد، از آن استفاده می‌کند. این شماره پورت نیز 16 بیتی است. اگر ماشین مقصد یک سرور باشد، شماره پورت در اکثر حالات، یک شماره پورت کاملاً شناخته شده است. اگر ماشین مقصد، سرویس‌گیرنده باشد، شماره پورت در اکثر حالات، یک شماره پورت موقتی است.

❋ total length : کل طول

این یک فیلد 16 بیتی است که کل طول دیتاگرام کاربر شامل هدر و دیتا را نشان می‌دهد. 16 بیت می‌تواند کل طول از 0 تا 65535 بایت را نشان دهد. با این حال، کل طول باید خیلی کوچکتر باشد زیرا که یک دیتاگرام UDP ، در یک بسته IP حمل می‌شود که طول آن باید از 65535 بایت کمتر باشد.

فیلد total length در دیتاگرام UDP واقعاً مورد نیاز نیست. یک دیتاگرام UDP در یک بسته IP کپسول‌بندی می‌شود. فیلدی در بسته IP وجود دارد که کل طول بسته IP را نشان می‌دهد. فیلد دیگری در بسته IP وجود دارد که طول هدر بسته IP را نشان می‌دهد . بنابراین اگر طول هدر بسته IP را از کل طول بسته IP کم کنیم، می‌توانیم طول دیتاگرام UDP که در بسته IP کپسول‌بندی شده است را بدست آوریم.

طول هدر IP − کل طول IP = طول UDP

www.Abirl.ir

با این حال، طراحان پروتوکل UDP احساس کردند که اگر پروتوکل UDP مقصد، طول UDP را از اطلاعات موجود در هدر دیتاگرام UDP استخراج کند، کارآمدتر از درخواست آن از نرم‌افزار IP است. باید به خاطر بسپاریم وقتی که نرم‌افزار IP، دیتاگرام UDP را به پروتوکل UDP تحویل می‌دهد، ابتدا هدر IP را حذف می‌کند و سپس آنچه باقی می‌ماند را به پروتوکل UDP تحویل می‌دهد. پس طول آنچه که به پروتوکل IP تحویل داده می‌شود نیز معرف کلِ طولِ دیتاگرام UDP است.

* **Checksum : جمع‌آزما**

از این فیلد برای آشکارسازی خطای کل دیتاگرام UDP (هدر و دیتا) استفاده می‌شود.

۱۱-۳ checksum در دیتاگرام UDP

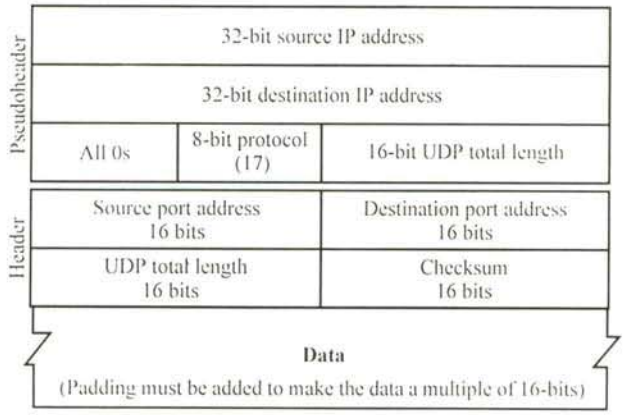
در فصل هشتم درباره مفهوم checksum و نحوه محاسبه آن صحبت کردیم. همچنین نشان دادیم که چگونه می‌توان checksum را برای بسته IP و ICMP حساب کرد. حالا نشان می‌دهیم که چگونه می‌توان این کار را برای UDP انجام داد.

محاسبه checksum برای UDP ، متفاوت با محاسبه آن برای IP و ICMP است. در اینجا checksum ، سه قسمت را در بر می‌گیرد: الف. **شبه هدر**[1] ب. هدر UDP ج. دیتایی که از لایه کاربرد می‌آید. شبه هدر، قسمتی از هدر بسته IP است و این بسته IP ، دیتاگرام UDP را در خود کپسول‌بندی کرده است (شکل ۱۱- ۸ را ببینید). در شکل ۱۱-۸ به تشابهات بین فیلدهای شبه هدر و آخرین 12 بایت هدر IP دقت نمایید.

اگر checksum ، شبه هدر را در بر نگیرد، ممکن است دیتاگرام UDP ، صحیح و سالم به مقصد برسد، اما اگر هدر IP خراب شده باشد، ممکن است آن بسته، به کامپیوتر دیگری تحویل داده شود. فیلد Protocol به این دلیل به شبه هدر اضافه شده است تا نشان دهد که این بسته IP ، متعلق به UDP و نه TCP است. اگر برنامه‌ای بتواند هم از UDP و هم از TCP استفاده کند، پورت مقصد می‌تواند مشابه باشد. مقدار فیلد Protocol برای UDP برابر با 17 است. اگر مقدار این فیلد در خلال انتقال بسته تغییر کند، محاسبه checksum گیرنده، خطای وارده را آشکار می‌کند و لذا UDP ، بسته را دور می‌اندازد و به این ترتیب مانع از تحویل دادن آن به پروتوکل دیگری می‌شود.

1- pseudoheader

www.Abiri.ir



شکل ۱۱-۸ : شبه هدر برای محاسبه checksum

محاسبه checksum در فرستنده

فرستنده برای محاسبه checksum ، به ترتیب مراحل زیر را طی می‌کند:

۱- شبه هدر را به دیتاگرام UDP اضافه می‌کند.

۲- فیلد checksum را با صفر پر می‌کند.

3 ـ کل بیتها را به بخشهای 16 بیتی (دو بایتی) تفکیک می‌نماید.

۴- اگر کل تعداد بایتها، عدد زوجی نباشد، یک بایت صفر به دنبال آن اضافه می‌کند. اضافه کردن این بایت تمام صفر فقط برای محاسبه checksum است و بعد از محاسبه checksum ، دور انداخته می‌شود.

۵- همه قسمت های 16 بیتی را با استفاده از ریاضیات مکمل یک با هم جمع می‌کند.

۶- مکمل نتیجه را حساب می‌کند (تمام 0 ها را به 1 و تمام 1 ها را به 0 تبدیل می‌کند) که یک عدد 16 بیتی می‌شود و سپس آن را در فیلد checksum قرار می‌دهد.

۷- شبه هدر و هر بایت صفری که اضافه شده است را حذف می‌کند.

۸- دیتاگرام UDP را برای کپسول‌بندی، به نرم افزار IP تحویل می‌دهد.

دقت نمایید که ترتیب سطرها در شبه هدر، هیچ اثری در محاسبه checksum ندارد. همچنین اضافه کردن 0 نیز اثری در محاسبه checksum ندارد. به همین دلیل، نرم افزاری که checksum را محاسبه می‌کند، می‌تواند کل هدر IP (20 بایت) را به دیتاگرام UDP اضافه کند، اولین 8 بایت هدر IP و فیلد TTL آن را با صفر پر کند، در جای فیلد checksum هدر IP مقدار فیلد UDP total length (کل طول UDP) را قرار دهد و سپس checksum را محاسبه کند. نتیجه، مشابه خواهد بود.

www.Abiri.ir

محاسبه checksum در گیرنده

گیرنده برای محاسبه checksum ، به ترتیب مراحل زیر را طی می‌کند :

۱- شبه هدر را به دیتاگرام UDP اضافه می‌کند.

۲- اگر کل تعداد بایتها، فرد باشد، یک بایت تمام صفر به دنبال آن قرار می‌دهد.

۳- کل بیتها را به بخشهای 16 بیتی تفکیک می‌کند.

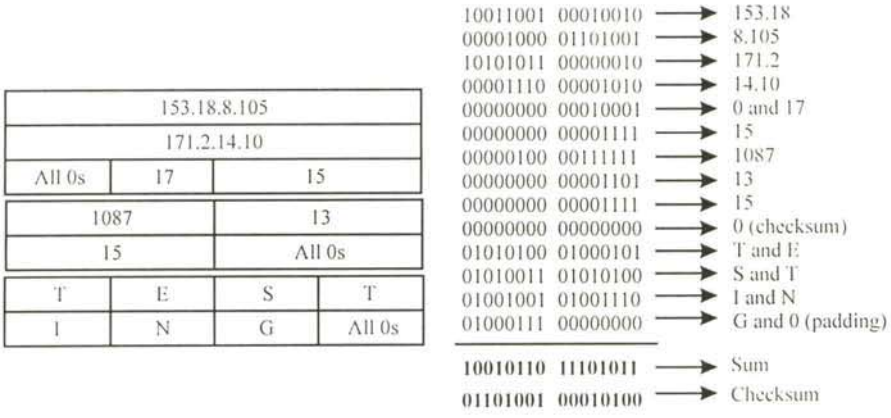
۴- همه بخشهای 16 بیتی را با استفاده از ریاضیات مکمل یک با هم جمع می‌کند.

۵- مکمل نتیجه را محاسبه می‌کند.

۶- اگر نتیجه، تمام صفر باشد، شبه هدر و هر بایت صفر اضافه شده‌ای را حذف می‌کند و دیتاگرام UDP را می‌پذیرد. اگر نتیجه چیز دیگری باشد، دیتاگرام UDP را دور می‌اندازد.

* یک مثال

شکل ۱۱-۹ محاسبه checksum را برای یک دیتاگرام کوچک UDP که فقط حاوی 7 بایت دیتا است، نشان می‌دهد. از آنجا که کل تعداد بایتهای دیتا، عدد فردی است لذا یک بایت 0 به دنبال آن اضافه شده است. وقتی که دیتاگرام UDP به IP تحویل داده می‌شود، شبه هدر و بایت صفر اضافه شده، حذف می‌گردند.



شکل ۱۱-۹ : نحوه محاسبه checksum در یک دیتاگرام کوچک UDP

استفاده اختیاری از checksum

محاسبه checksum و گذاشتن آن در فیلد checksum یک دیتاگرام UDP ، اختیاری است. اگر checksum محاسبه نشود، این فیلد با صفر پر می‌شود. ممکن است این سؤال پیش آید که وقتی نرم‌افزار UDP در کامپیوتر مقصد، یک دیتاگرام UDP با checksum برابر با صفر

www.Abiri.ir

دریافت کند، چطور می‌تواند تشخیص دهد که آیا از محاسبه checksum در فرستنده، صرف نظر شده است و یا اینکه نتیجه محاسبه به تمام 0 منجر شده است؟ پاسخ این سؤال بسیار ساده است. اگر مبدأ، مقدار checksum را محاسبه کرده باشد و از قضا به حاصل جمع تمام 0 رسیده باشد، باید مکمل آن را حساب کند. بنابراین چیزی که به عنوان checksum ارسال می‌شود، تمام 1 است. دقت نمایید که مقدار checksum در فرستنده نمی‌تواند تمام 0 شود زیرا این حالت مستلزم آن است که حاصل جمع در فرستنده برابر با تمام 1 باشد و از آنجا که در محاسبه checksum از ریاضیات مکمل یک استفاده می‌شود، امکان ندارد که حاصل جمع تمام 1 بدست آید.

۴-۱۱ طرز کار UDP

UDP از مفاهیم متداول در لایه انتقال استفاده می‌کند. در اینجا به اختصار درباره این مفاهیم صحبت می‌کنیم و سپس وقتی که در فصل بعد به پروتوکل TCP رسیدیم، این مفاهیم را بسط می‌دهیم.

سرویسهای فاقد اتصال

همانطور که قبلاً اشاره شد، UDP یک ***سرویس فاقد اتصال***[1] را فراهم می‌نماید. معنی آن این است که هر دیتاگرام ارسالی توسط UDP ، مستقل از سایر دیتاگرامهاست. حتی اگر دیتاگرامهای UDP از یک برنامه مبدأ مشابه سرچشمه بگیرند و به سمت یک برنامه مقصد مشابه حرکت کنند، هیچ رابطه‌ای بین آنها وجود ندارد. دیتاگرامهای UDP شماره‌گذاری نمی‌شوند. همچنین برخلاف TCP ، نَه اتصالی برقرار می‌شود و نَه اتصالی خاتمه می‌یابد. مفهوم آن این است که هر دیتاگرام UDP می‌تواند از مسیر متفاوتی به مقصد برسد.

یکی از پیامدهای فاقد اتصال بودن برای برنامه‌ای که از UDP استفاده می‌کند، این است که نمی‌تواند دنباله پیوسته‌ای از دیتا را به UDP بدهد و از UDP انتظار داشته باشد که آن را به دیتاگرامهای متعدد مرتبط با هم تفکیک کند، بلکه هر پیغامی که برای ارسال به UDP تحویل داده می‌شود باید به اندازه‌ای کوچک باشد که در یک دیتاگرام UDP جا شود. بنابراین فقط برنامه‌هایی که پیغامهای کوتاه ارسال می‌کنند می‌توانند از UDP استفاده کنند.

1- Connectionless service

www.Abiri.ir

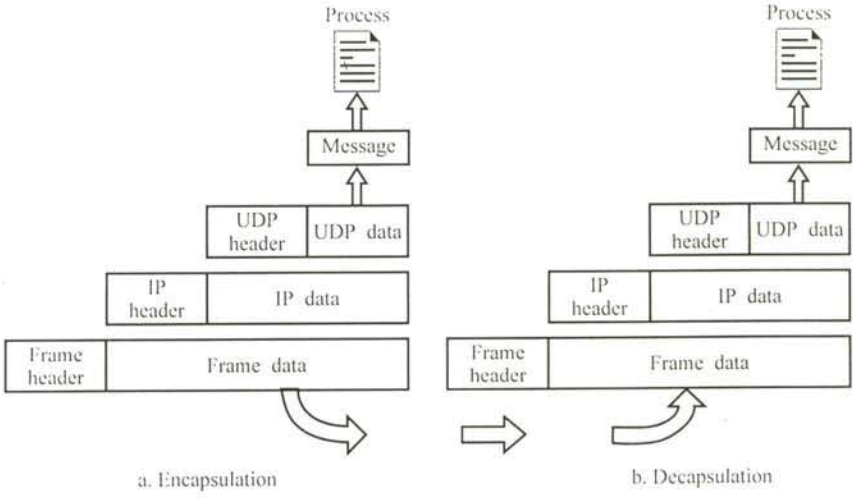
کنترل فلو و خطا

UDP یک پروتوکل بسیار ساده و غیر قابل اطمینان است. UDP مکانیزمی برای کنترل فلو ندارد و لذا از پنجره کنترل فلو استفاده نمی‌کند. بنابراین ممکن است گیرنده از پیغامهای دریافتی، لبریز گردد.

UDP بجز checksum فاقد هر گونه مکانیزم دیگری برای کنترل خطاست و لذا فرستنده نمی‌فهمد که آیا پیغام ارسالی او، گم شده و یا در شبکه، تکرار شده است. وقتی که گیرنده به کمک محاسبه checksum متوجه بروز خطایی می‌شود، صرفاً دیتاگرام دریافتی را دور می‌اندازد. فقدان مکانیزمهای کنترل خطا و کنترل فلو به این معنی است که برنامه استفاده کننده از UDP ، باید خودش چنین مکانیزمهایی را فراهم نماید.

کپسول‌بندی و خارج سازی از کپسول

برای ارسال پیغامی از یک برنامه به برنامه دیگر، پروتوکل UDP فرستنده، پیغام را کپسول‌بندی می‌کند و پروتوکل UDP گیرنده، آن را از کپسول، خارج می‌کند (شکل ۱۱-۱۰ را ببینید).



شکل ۱۱-۱۰ : کپسول‌بندی و خارج کردن از کپسول

* کپسول‌بندی

وقتی که برنامه‌ای ، پیغامی برای ارسال از طریق UDP دارد، پیغام را به همراه یک جفت آدرس سوکت و نیز طول دیتا، به نرم‌افزار UDP می‌دهد. نرم افزار UDP پیغام را دریافت می‌کند و هدر UDP را به آن اضافه می‌کند. سپس دیتاگرام UDP را با آدرس سوکتها، به

www.Abiri.ir

نرم‌افزار IP می‌دهد. IP هدر خود را به آن اضافه می‌کند و با قرار دادن عدد 17 در فیلد Protocol هدر خود، نشان می‌دهد که پیغام دریافتی از پروتوکل UDP آمده است. سپس بسته IP به لایه دیتالینک داده می‌شود. لایه دیتالینک، بسته IP را دریافت می‌کند، هدر و دنباله را به آن اضافه می‌کند و سپس آن را به لایه فیزیکی می‌دهد. لایه فیزیکی، بیتها را به صورت سیگنالهای الکتریکی یا نوری در می‌آورد و آنها را روی محیط انتقال، قرار می‌دهد.

* خارج سازی از کپسول

وقتی که پیغام به ماشین مقصد می‌رسد، لایه فیزیکی، سیگنالهای دریافتی را به بیت تبدیل می‌کند و آنها را به لایه دیتالینک می‌دهد. لایه دیتالینک با استفاده از هدر (و دنباله) صحت دیتای دریافتی را چک می‌کند. اگر خطایی رخ نداده باشد، هدر و دنباله، حذف می‌شوند و دیتای دریافتی به لایه IP تحویل داده می‌شود. نرم افزار IP عملیات تست خطای خود را انجام می‌دهد. اگر خطایی رخ نداده باشد، هدر IP حذف می‌شود و دیتاگرام UDP به همراه آدرسهای IP فرستنده و گیرنده، به نرم‌افزار UDP تحویل داده می‌شود. UDP برای خطایابی کل دیتاگرام UDP ، از checksum استفاده می‌کند. اگر خطایی رخ نداده باشد، هدر UDP حذف می‌شود و دیتا به همراه آدرس سوکت فرستنده، به برنامه گیرنده داده می‌شود و اگر برنامه گیرنده بخواهد پاسخ دهد، می‌تواند از این آدرس استفاده کند.

در صف گذاشتن

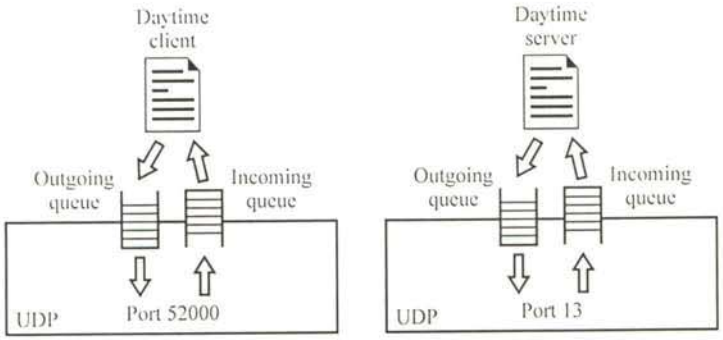
درباره پورتها، بدون اینکه بگوئیم چطور می‌توان آنها را پیاده‌سازی کرد، صحبت کردیم. در UDP ، تناظری بین صفها و پورتها ایجاد می‌شود (شکل ۱۱-۱۱ را ببینید).

در سمت سرویس‌گیرنده وقتی برنامه‌ای اجرا می‌شود، شماره پورتی را از سیستم عامل درخواست می‌نماید. برخی از پیاده‌سازیها به هر برنامه، هم یک صف ورودی و یک صف خروجی اختصاص می‌دهند. برخی دیگر از پیاده‌سازیها، فقط یک صف ورودی به هر برنامه اختصاص می‌دهند. دقت نمایید حتی اگر برنامه‌ای بخواهد با چندین برنامه دیگر مخابره دیتا داشته باشد، فقط یک شماره پورت می‌گیرد و لذا فقط یک صف ورودی و فقط یک صف خروجی در اختیار او قرار می‌گیرد. در اکثر حالتها، صفهایی که توسط سرویس‌گیرنده باز می‌شوند، توسط شماره پورتهای موقتی، مشخص می‌شوند. تا زمانی که برنامه‌ای در حال اجراست، آن صفها کار می‌کنند. وقتی که اجرای برنامه پایان می‌یابد، صفها نیز پاک می‌شوند.

برنامه سرویس‌گیرنده می‌تواند با استفاده از شماره پورت مبدأ که در درخواست آمده است، پیغام خود را در صف خروجی قرار دهد. UDP یکی یکی پیغامها را از صف خروجی بر

www.Abiri.ir

می‌دارد و بعد از اضافه کردن هدر UDP به آنها، آنها را به پروتوکل IP تحویل می‌دهد . ممکن است صف خروجی، لبریز شود. اگر چنین اتفاقی رخ دهد، سیستم عامل می‌تواند از برنامه سرویس‌گیرنده بخواهد تا در ارسال پیغامهای بعدی، قدری صبر کند.



شکل ۱۱-۱۱ : صفها در UDP

وقتی که پیغامی برای سرویس‌گیرنده می‌رسد، UDP چک می‌کند ببیند آیا یک صف ورودی برای شماره پورتی که در فیلد ***شماره پورت مقصد***[1] دیتاگرام UDP آمده است، ایجاد شده است یا خیر. اگر چنین صفی وجود داشته باشد، UDP دیتاگرام دریافتی را در انتهای صف قرار می‌دهد. اگر چنین صفی وجود نداشته باشد، UDP دیتاگرام دریافتی را دور می‌اندازد و از پروتوکل ICMP می‌خواهد یک پیغام « پورت، غیر قابل دسترسی است » برای سرور بفرستد. تمام پیغامهایی که برای یک برنامه مشخص سرویس‌گیرنده از راه می‌رسند، چه از طرف سرور مشابه و چه از طرف سرور متفاوت، در صف ورودی یکسانی قرار می‌گیرند و لذا ممکن است صف ورودی لبریز کند. اگر چنین اتفاقی بیفتد، UDP دیتاگرام را دور می‌اندازد و از ICMP می‌خواهد یک پیغام « پورت، غیر قابل دسترسی است » برای سرور بفرستد.

در سمت سرور، مکانیزم تولید صفها، متفاوت است. در ساده‌ترین شکل آن، وقتی که سرور راه می‌افتد، با استفاده از پورت کاملاً شناخته شده خود از صفهای ورودی و خروجی استفاده می‌کند. تا زمانی که سرور در حال کار است، این صفها باز می‌مانند. وقتی که پیغامی برای سروری از راه می‌رسد، UDP چک می‌کند ببیند آیا یک صف ورودی برای شماره پورتی که در فیلد شماره پورت مقصد دیتاگرام دریافتی مشخص شده است، ایجاد شده است یا خیر. اگر چنین صفی ایجاد شده باشد، UDP دیتاگرام دریافتی را در انتهای صف قرار می‌دهد. اگر چنین صفی وجود نداشته باشد، UDP دیتاگرام را دور می‌اندازد و از ICMP می‌خواهد یک

1- destination port number

www.Abiri.ir

پیغام « پورت، غیر قابل دسترسی است » را برای سرویس‌گیرنده ارسال کند. همه پیغامهای ورودی برای یک سرور مشخص، چه از طرف یک سرویس‌گیرنده و چه از طرف چندین سرویس‌گیرنده آمده باشند، در صف یکسانی قرار می‌گیرند و لذا ممکن است صف ورودی، لبریز کند. اگر چنین اتفاقی بیفتد، UDP دیتاگرام دریافتی را دور می‌اندازد و از ICMP می‌خواهد یک پیغام « پورت، غیر قابل دسترسی است » را برای سرویس‌گیرنده بفرستد.

وقتی که سروری می‌خواهد به یک درخواست گیرنده پاسخ دهد، با استفاده از شماره پورت مبدأ که در درخواست آمده است، پاسخهایش را در صف خروجی قرار می‌دهد. UDP یکی یکی پیغامها را بر می‌دارد و بعد از اضافه کردن هدر UDP ، آنها را به IP تحویل می‌دهد. یک صف خروجی ممکن است لبریز کند. اگر چنین اتفاقی رخ دهد، سیستم عامل از سرور می‌خواهد که قبل از ارسال پیغامهای دیگرش، قدری صبر کند.

مالتی‌پلکسینگ و دی‌مالتی‌پلکسینگ در UDP

در ماشینی که پشته پروتوکل TCP/IP در حال اجراست، فقط یک UDP وجود دارد اما ممکن است چندین برنامه وجود داشته باشند که همگی بخواهند از سرویسهای UDP استفاده کنند. برای عملی شدن این کار، UDP از ***مالتی‌پلکسینگ***[1] و ***دی‌مالتی‌پلکسینگ***[2] برنامه‌های متقاضی سرویس UDP ، استفاده می‌کند (شکل ۱۱-۱۲ را ببینید).

* مالتی‌پلکسینگ

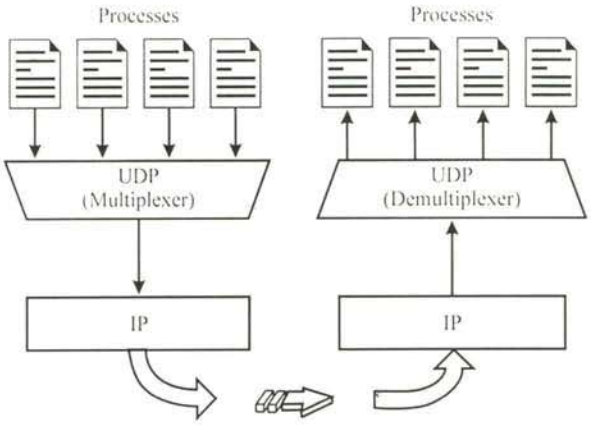
ممکن است در سمت فرستنده، برنامه‌های متعددی وجود داشته باشند که بخواهند دیتاگرامهای UDP را ارسال نمایند. با این حال، فقط یک UDP وجود دارد. این یک وضعیت **چند به یک**[3] است و به مالتی‌پلکسینگ نیاز دارد. UDP پیغامهای صادره از سوی برنامه‌های مختلف را می‌گیرد و تفاوت آنها را از طریق شماره پورتی که به هر کدام از برنامه‌ها اختصاص داده شده است، تشخیص می‌دهد. UDP بعد از اضافه کردن هدر UDP ، دیتاگرام UDP را به IP تحویل می‌دهد.

1- multiplexing
2- Demultiplexing
3- many-to-one

www.Abiri.ir

*** دی‌مالتی‌پلکسینگ**

در سمت گیرنده، فقط یک UDP وجود دارد. با این حال ممکن است چندین برنامه داشته باشیم که بتوانند دیتاگرامهای UDP را دریافت کنند. این یک رابطه ***یک به چند***[1] است و به دی‌مالتی‌پلکسینگ نیاز دارد. UDP دیتاگرامهای UDP را از IP تحویل می‌گیرد. UDP بعد از چک خطا و حذف هدر UDP ، هر پیغام را بر اساس شماره پورت مقصد، به برنامه مربوطه تحویل می‌دهد.



شکل ۱۱-۱۲: مالتی‌پلکسینگ و دی‌مالتی‌پلکسینگ در UDP

۱۱-۵ کاربردهای UDP

لیست زیر، برخی از کاربردهای پروتوکل UDP را نشان می‌دهد:

الف. سرویس UDP برای کاربردی مفید است که به مخابره ساده ***درخواست- پاسخ***[2] نیاز داشته باشد و توجه چندانی نیز به کنترل خطا و کنترل فلو نکند. معمولاً UDP برای پروتوکلی مانند FTP که می‌خواهد حجم زیادی از دیتا را ارسال کند، مناسب نیست (فصل نوزدهم را ببینید).

ب. UDP برای کاربردی که مکانیزم داخلی کنترل فلو و کنترل خطا داشته باشد، مناسب است. به عنوان مثال، پروتوکل TFTP حاوی مکانیزم داخلی کنترل خطا و کنترل فلو است (فصل نوزدهم را ببینید). TFTP می‌تواند به راحتی از UDP استفاده کند.

1- one-to- many
2- request-response

www.Abiri.ir

ج. UDP یک پروتوکل مناسب لایه انتقال برای ***چندگیرندگی***[1] است. قابلیت multicasting در نرم‌افزار UDP اِعمال شده است اما نرم‌افزار TCP چنین قابلیتی را ندارد.

د. UDP در کاربردهای مدیریتی مانند پروتوکل SNMP (فصل بیست و یکم را ببینید) به کار می‌رود.

ه. UDP در برخی پروتوکلهای مسیردهی مانند RIP (فصل چهاردهم را ببینید) به کار می‌رود.

۱۱-۶ نرم‌افزار UDP

برای آنکه نشان دهیم چگونه UDP دیتاگرامهای UDP را ارسال و دریافت می‌کند، نمونه ساده‌ای از نرم‌افزار UDP را ارائه می‌نمائیم.

می‌توان گفت نرم‌افزار UDP از پنج جزء ساخته شده است: یک ***جدول کنترل بلاک***[2] ، ***صفهای ورودی***[3] ، یک ***ماجول کنترل بلاک***[4] ، یک ***ماجول ورودی***[5] و یک ***ماجول خروجی***[6]. شکل ۱۱-۱۳ این پنج جزء و ارتباط بین آنها را نشان می‌دهد.

جدول کنترل بلاک

در نرم افزار ما، UDP حاوی یک جدول کنترل بلاک است که نشان می‌دهد کدام پورتها، باز هستند. هر المانی در این جدول، حداقل چهار فیلد دارد:

State : حالت. می‌تواند دو مقدار ***آزاد***[7] و ***در حال استفاده***[8] را به خود بگیرد.

Process ID : شناسه برنامه.

Port number : شماره پورت.

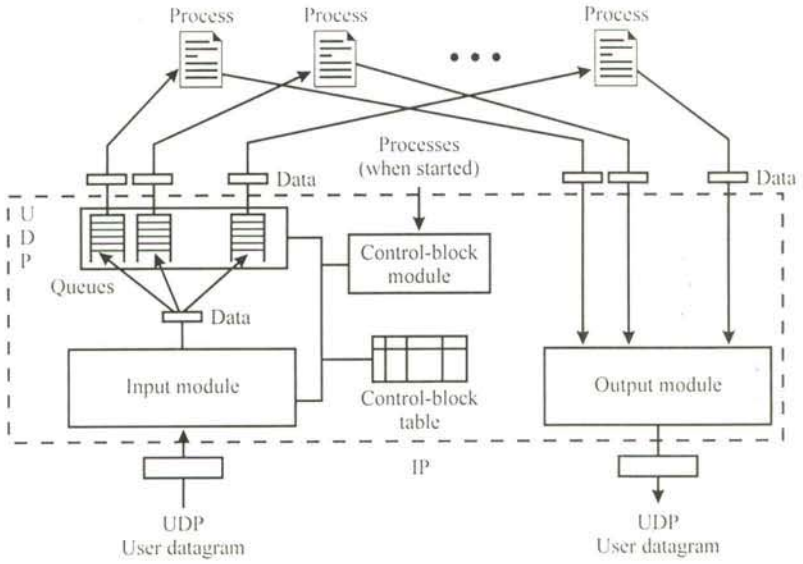
Queue number : شماره صف.

صفهای ورودی

نرم‌افزار UDP ما از مجموعه‌ای از صفها، یکی برای هر برنامه، استفاده می‌کند. در این طرح، از صفهای خروجی استفاده نمی‌کنیم.

1- multicasting
2- control block table
3- input queues
4- control block module
5- input module
6- output module
7- FREE
8- IN-USE

www.Abiri.ir



شکل ۱۱-۱۳ : نرم‌افزار UDP

ماجول کنترل بلاک

ماجول کنترل بلاک، مسئول مدیریت جدول کنترل بلاک است. وقتی که برنامه‌ای اجرا می‌شود، یک شماره پورت را از سیستم عامل درخواست می‌کند. سیستم عامل، شماره پورتهای کاملاً شناخته شده را به سرورها و شماره پورتهای موقتی را به سرویس‌گیرنده‌ها اختصاص می‌دهد. برنامه، مقدار شناسه برنامه و شماره پورت را به ماجول کنترل بلاک می‌دهد تا المانی را برای آن برنامه در جدول کنترل بلاک، ایجاد نماید. این ماجول، صفها را ایجاد نمی‌کند. مقدار فیلد شماره صف، برابر با صفر است. دقت نمایید که در این نرم‌افزار، کاری برای وضعیتی که جدول پر شود، انجام نداده‌ایم.

* جریان رویدادهای ماجول کنترل بلاک

ورودی : یک شناسه برنامه و یک شماره پورت

۱- در جدول کنترل بلاک به دنبال یک المان Free می‌گردد.

۱- اگر پیدا نکرد :

۱- با استفاده از یک استراتژی از قبل تعریف شده، یک المان را پاک می‌کند.

۲- یک المان جدید با حالت IN-USE می‌سازد.

۳- شناسه برنامه و شماره پورت را داخل فیلدهای مربوط به این المان جدید

www.Abiri.ir

می‌نویسد.
۲- برمی‌گردد.

ماجول ورودی

ماجول ورودی، یک دیتاگرام UDP را از IP دریافت می‌کند. او در جدول کنترل بلاک به دنبال المانی می‌گردد که شماره پورت آن، مشابه با شماره پورت مقصد در این دیتاگرام باشد. اگر چنین المانی پیدا شد، ماجول از اطلاعات آن المان استفاده می‌کند تا دیتاگرام دریافتی را در صف ورودی مربوطه قرار دهد. اگر چنین المانی پیدا نشد، یک پیغام ICMP تولید می‌کند.

* جریان رویدادهای ماجول ورودی

ورودی: یک دیتاگرام UDP از IP
۱- به دنبال المان متناظر در جدول کنترل بلاک می‌گردد.
۱- اگر پیدا کرد :
۱- فیلد **صف**[1] را چک می‌کند ببیند آیا صفی اختصاص یافته است.
۱- اگر صفی اختصاص نیافته است :
۱- صفی اختصاص می‌دهد.
۲- دیتا را داخل صف اختصاص یافته قرار می‌دهد.
۲- اگر پیدا نکرد :
۱- از ماجول ICMP می‌خواهد یک پیغام « پورت، غیر قابل دسترسی است » را ارسال کند.
۲- دیتاگرام دریافتی را دور می‌اندازد.
۲- برمی‌گردد.

ماجول خروجی

ماجول خروجی مسئول تولید و ارسال دیتاگرامهای UDP است.

* جریان رویدادهای ماجول خروجی

ورودی : دیتا و اطلاعات از یک برنامه
۱- یک دیتاگرام UDP را می‌سازد.

1- queue

www.Abiri.ir

۲- دیتاگرام را ارسال می‌کند.
۳- برمی‌گردد.

مثالها

در این بخش، چند مثال ارائه می‌دهیم که نشان می‌دهند چگونه نرم‌افزار ما به ورودی و خروجی پاسخ می‌دهد. جدول کنترل بلاک که در مثالها از آن استفاده خواهیم کرد، در اینجا آمده است.

State	*Process ID*	*Port Number*	*Queue Number*
IN-USE	2,345	52,010	34
IN-USE	3,422	52,011	
FREE			
IN-USE	4,652	52,012	38
FREE			

جدول ۱۱-۲ : جدول کنترل بلاک مورد استفاده در مثالها

* مثال ۲

اولین رخداد این است که یک دیتاگرام با شماره پورت مقصد 52012 از راه می‌رسد. ماجول ورودی در جدول کنترل بلاک، به دنبال این شماره پورت می‌گردد و آن را پیدا می‌کند. شماره صف 38 ، به این پورت اختصاص یافته است و معنی آن این است که قبلاً از این پورت استفاده شده است. ماجول ورودی، دیتاگرام دریافتی را در صف شماره 38 قرار می‌دهد. جدول کنترل بلاک تغییر نمی‌کند.

* مثال ۳

بعد از چند ثانیه، برنامه‌ای آغاز به کار می‌کند. او از سیستم عامل می‌خواهد که یک شماره پورت در اختیارش بگذارد و شماره پورت 52014 به او داده می‌شود. حالا برنامه، شماره ID خود (4978) و شماره پورت اهداء شده را به ماجول کنترل بلاک می‌دهد، تا المانی را در جدول کنترل بلاک ایجاد کند. ماجول کنترل بلاک، اولین المان FREE موجود در جدول را پیدا می‌کند، اطلاعات مذکور را در آن می‌نویسد و حالت آن المان را به IN-USE تغییر می‌دهد. در این لحظه ماجول کنترل بلاک، صفی به این المان اختصاص نمی‌دهد زیرا که هنوز دیتاگرامی برای این مقصد از راه نرسیده است (جدول ۱۱-۳ را ببینید).

www.Abiri.ir

State	Process ID	Port Number	Queue Number
IN-USE	2,345	52,010	34
IN-USE	3,422	52,011	
IN-USE	**4,978**	**52,014**	
IN-USE	4,652	52,012	38
FREE			

جدول ۱۱-۳ : محتوای جدول کنترل بلاک بعد از اعمال نتیجه مثال ۳

*** مثال ۴**

اکنون یک دیتاگرام UDP برای پورت 52011 از راه می‌رسد. ماجول ورودی، محتوای جدول کنترل بلاک را چکّ می‌کند و در می‌یابد که هنوز صفی به این مقصد اختصاص داده نشده است زیرا که برای اولین بار است که یک دیتاگرام UDP برای این مقصد، از راه رسیده است. ماجول ورودی، صفی ایجاد می‌کند و به آن شماره 43 را می‌دهد (جدول ۱۱-۴ را ببینید).

State	Process ID	Port Number	Queue Number
IN-USE	2,345	52,010	34
IN-USE	3,422	52,011	**43**
IN-USE	4,978	52,014	
IN-USE	4,652	52,012	38
FREE			

جدول ۱۱-۴ : محتوای جدول کنترل بلاک بعد از اعمال نتیجه مثال ۴

*** مثال ۵**

پس از چند ثانیه، یک دیتاگرام UDP به مقصد پورت شماره 52222 از راه می‌رسد. ماجول ورودی، جدول کنترل بلاک را چک می‌کند و نمی‌تواند المانی برای این مقصد پیدا کند. دیتاگرام دریافتی، دور انداخته می‌شود و از ICMP خواسته می‌شود یک پیغام « پورت، غیر قابل دسترسی است » را برای مبدأ ارسال کند.

www.Abiri.ir

۱۱-۷ عبارات کلیدی

برنامه کاربردی (application program)
جمع‌آزما (checksum)
سرویس گیرنده (client)
سرویس فاقد اتصال(Connectionless service)
پروتوکل غیر قابل اطمینان و فاقد اتصال لایه انتقال
(connectionless unreliable transport protocol)
از کپسول خارجی سازی (decapsulation)
پورت دینامیک (dynamic port)
کپسول‌بندی (encapsulation)
شماره پورت موقتی (ephemeral port number)
کنترل خطا (error control)
کنترل فلو (flow control)
ICANN
مالتی‌پلکسینگ (multiplexing)
شماره پورت (port number)
برنامه (process)
مخابره برنامه به برنامه (process-to-process communication)
شبه هدر (pseudoheader)
صف (queue)
پورت ثبت شده (registered port)
سرور (server)
آدرس سوکت (socket address)
لایه انتقال (transport layer)
دیتاگرام کاربر (user datagram)
پروتوکل دیتاگرام کاربر(User Datagram protocol : UDP)
شماره پورت کاملاً شناخته شده (well-known port number)

۱۱-۸ خلاصه

- UDP یک پروتوکل لایه انتقال است که مخابره برنامه به برنامه را برقرار می‌کند.

www.Abiri.ir

- UDP یک پروتوکل غیر قابل اطمینان و فاقد اتصال است که به اطلاعات بالاسری کمی نیاز دارد و سرعت تحویل پیغام آن، سریع است.
- در طرح سرویس‌گیرنده / سرویس‌دهنده ، یک برنامه کاربردی در سمت ماشین محلی به نام سرویس‌گیرنده، به سرویسهای یک برنامه کاربردی در سمت ماشین دور دست به نام سرویس‌دهنده یا سرور، نیاز دارد.
- هر برنامه کاربردی، یک شماره پورت منحصر به فرد دارد که وجه تمایز او از سایر برنامه‌هایی است که در همان لحظه و روی همان ماشین، در حال اجرا هستند.
- به برنامه سرویس‌گیرنده، یک شماره پورت تصادفی به نام شماره پورت موقتی، اختصاص داده می‌شود.
- به برنامه سرور، یک شماره پورت سرتاسری به نام شماره پورت کاملاً شناخته شده، اختصاص داده می‌شود.
- ICANN محدوده‌هایی را برای انواع مختلف شماره پورتها مشخص کرده است.
- ترکیب آدرس IP و شماره پورت که آدرس سوکت نامیده می‌شود، بطور منحصر به فردی یک برنامه در حال اجرا روی یک ماشین را مشخص می‌کند.
- UDP به یک جفت آدرس سوکت نیاز دارد: آدرس سوکت سرویس‌گیرنده و آدرس سوکت سرور.
- به بسته‌های UDP ، دیتاگرام کاربر گفته می‌شود.
- تنها تلاش UDP برای کنترل خطا، استفاده از checksum است.
- اضافه کردن شبه هدر به محاسبات مربوط به checksum ، امکان آشکارسازی خطاهای رخ داده در آدرسهای IP مبدأ و مقصد را فراهم می‌نماید.
- UDP مکانیزمی برای کنترل فلو ندارد.
- یک دیتاگرام UDP در فیلد دیتای یک بسته IP ، کپسول‌بندی می‌شود.
- صفهای ورودی و خروجی، پیغامهایی که به UDP تحویل داده می‌شوند و یا از UDP تحویل گرفته می‌شوند را در خود نگه می‌دارند.
- UDP برای ارسال دیتاگرامهای چندین برنامه که روی یک ماشین در حال اجرا هستند، از مالتی پلکسینگ استفاده می‌کند.
- UDP با استفاده از دی مالتی پلکسینگ، دیتاگرامهای دریافتی را تحویل برنامه‌های مختلفی که روی یک ماشین واحد در حال اجرا هستند، می‌دهد.
- یک نرم افزار UDP می‌تواند حاوی پنج جزء باشد : یک جدول کنترل بلاک، یک ماجول کنترل بلاک، صفهای ورودی، یک ماجول ورودی و یک ماجول خروجی.
- صفهای ورودی، دیتاگرامهای ورودی را در خود جای می‌دهند.

www.Abiri.ir

- ماجول کنترل بلاک، مسئول مدیریت المانهای موجود در جدول کنترل بلاک است.
- ماجول ورودی، صفهای ورودی را ایجاد می‌کند. ماجول خروجی، دیتاگرامهای کاربر را ارسال می‌نماید.

۱۱-۹ مجموعه تمرین‌ها

تمرین‌ها

۱- در مواردی که قابلیت اطمینان در درجه اول اهمیت قرار ندارد، UDP می‌تواند پروتوکل خوبی برای لایه انتقال محسوب گردد. مثالهایی از چنین مواردی ارائه نمائید.

۲- آیا UDP و IP به یک اندازه غیر قابل اطمینان هستند؟ چرا بله یا چرا نه؟

۳- آیا لازم است آدرس پورتها، منحصر به فرد باشند؟ چرا بله یا چرا نه؟ چرا شماره پورتها کوتاهتر از آدرسهای IP هستند؟

۴- لغت ephemeral (موقتی) در کتاب لغت، چگونه تعریف شده است؟ چگونه این تعریف را می‌توان در مورد شماره پورتهای موقتی به کار برد؟

۵- محتوای هدر UDP ، دیتاگرامی را نشان می‌دهد که پیغامی را از یک سرویس‌گیرنده TFTP به یک سرور TFTP می‌برد. فیلد checksum را با صفر پر کنید. از یک شماره پورت موقتی مناسب و شماره پورت کاملاً شناخته شده صحیح استفاده کنید. طول دیتا، 40 بایت است. با استفاده از فرمت شکل ۱۱- ۷ دیتاگرام UDP را نشان دهید.

۶- یک سرویس‌گیرنده SNMP که روی ماشینی به آدرس IP برابر با 122.45.12.7 در حال اجراست، پیغامی را برای یک سرور SNMP که روی ماشین دیگری با آدرس IP برابر با 200.112.45.90 در حال اجراست، ارسال می‌کند. چه جفت سوکتی در این مخابره به کار می‌رود؟

۷- یک سرور TFTP که روی ماشینی به آدرس IP برابر با 130.45.12.7 در حال اجراست، پیغامی را برای یک سرویس‌گیرنده TFTP که روی ماشین دیگری با آدرس IP برابر با 14.90.90.33 در حال اجراست، می‌فرستد. چه جفت سوکتی در این مخابره به کار می‌رود؟

۸- حداقل سایز یک دیتاگرام UDP چیست؟

۹- حداکثر سایز یک دیتاگرام UDP چیست؟

۱۰- حداقل سایز دیتایی که می‌توان در یک دیتاگرام UDP کپسول‌بندی کرد، چقدر است؟

۱۱- حداکثر سایز دیتایی که می‌توان در یک دیتاگرام UDP کپسول‌بندی کرد، چقدر است؟

۱۲- یک سرویس‌گیرنده بسته‌ای به طول 68000 بایت دارد. نشان دهید که چطور می‌توان فقط با استفاده از یک دیتاگرام UDP ، این بسته را منتقل کرد.

www.Abiri.ir

۱۳- یک سرویس‌گیرنده با استفاده از UDP دیتایی را برای یک سرور می‌فرستد. طول دیتا برابر با 16 بایت است. راندمان این انتقال (نسبت بایتهای مفید به کل تعداد بایتهای ارسالی) را در سطح UDP محاسبه نمایید.

۱۴- مثال ۱۳ را تکرار نمایید و این بار راندمان را در سطح IP محاسبه نمائید. فرض کنید که هدر IP از هیچ optionای استفاده نمی‌کند.

۱۵- مثال ۱۳ را تکرار کنید و این بار راندمان را در سطح لایه دیتالینک محاسبه نمائید. فرض کنید IP از هیچ option ای استفاده نمی‌کند و فرض کنید که پروتوکل لایه دیتالینک، اترنت است.

۱۶- آنچه به دنبال می‌آید، محتوای هدر UDP یک دیتاگرام فرضی بر حسب هگزادسیمال است

06 32 00 0D 00 1C E2 17

الف- شماره پورت مبدأ چیست؟

ب ـ شماره پورت مقصد چیست؟

پ – کل طول دیتاگرام UDP چقدر است؟

ت – طول دیتا چقدر است؟

ث ـ آیا این بسته، از یک سرویس‌گیرنده به سمت یک سرور و یا بالعکس حرکت می‌کند؟

ج – چه پروتوکل لایه کاربردی از سرویس UDP استفاده می‌کند؟

فعالیتهای تحقیقاتی

۱۷- با استفاده از برنامه grep ، پورتهای کاملاً شناخته شده بیشتری را پیدا کنید.

۱۸- RFCهای مربوط به پروتوکل UDP را پیدا کنید.

۱۹- RFCهای مربوط به شماره پورتها را پیدا کنید. آیا هیچ RFC خاصی مرتبط با پورتهای کاملاً شناخته شده یا پورتهای موقتی وجود دارد؟

۲۰- اطلاعات بیشتری درباره ICANN پیدا کنید. قبل از آنکه نام آن تغییر کند، چه نامی داشت؟

۲۱- ثابت کنید که محاسبه checksum در UDP هرگز به نتیجه تمام صفر منجر نمی‌شود.
راهنمایی : از ریاضیات ماجولار (modular arithmetic) استفاده کنید.

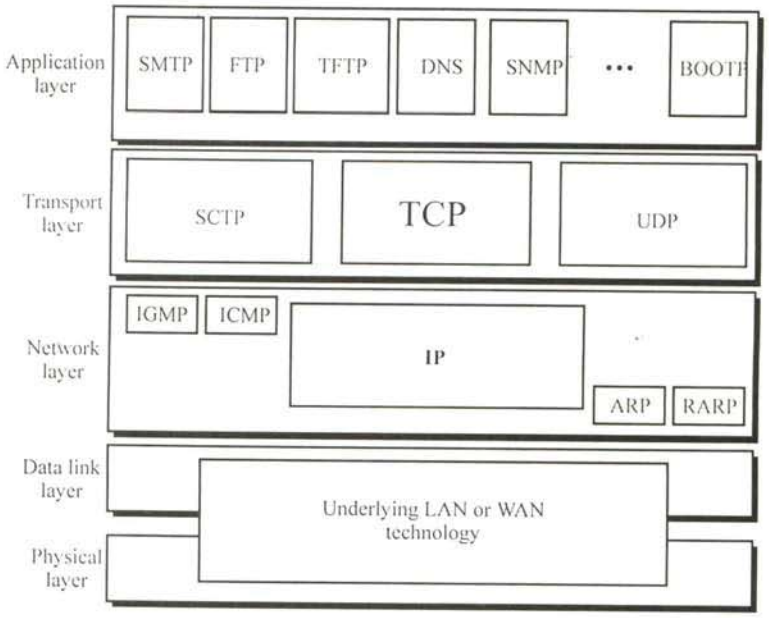
www.Abiri.ir

فصل دوازدهم

پروتوکل TCP

در ابتدا، پشته پروتوکل TCP/IP دو پروتوکل را برای لایه انتقال معرفی کرده بود : UDP و UDP. TCP را در فصل یازدهم مطالعه کردیم و در این فصل، TCP را بررسی خواهیم کرد. امروزه برخی از پیاده‌سازیها از یک پروتوکل جدید لایه انتقال به نام SCTP نیز استفاده می‌کنند که در فصل سیزدهم به مطالعه آن خواهیم پرداخت.

شکل ۱۲-۱ رابطه TCP با سایر پروتوکلها را در پشته پروتوکل TCP/IP نشان می‌دهد. TCP بین لایه کاربرد و لایه شبکه قرار گرفته است و به عنوان واسطه‌ای بین برنامه‌های کاربردی و عملیات شبکه به کار می‌رود.



شکل ۱۲-۱: پشته پروتوکل TCP/IP

www.Abiri.ir

TCP مشابه UDP یک پروتوکل ***برنامه به برنامه***[1] است. بنابراین TCP مانند UDP از شماره پورت استفاده می‌کند. برخلاف UDP ، TCP یک پروتوکل اتصالگراست یعنی برای ارسال دیتا، بین TCP فرستنده و TCP گیرنده، یک اتصال مجازی برقرار می‌شود. به علاوه TCP از مکانیزمهای کنترل خطا و کنترل فلو در سطح لایه انتقال استفاده می‌کند.

به طور خلاصه TCP یک پروتوکل اتصالگرا و قابل اطمینان است. او به سرویسهای IP ویژگیهای اتصالگرا بودن و مطمئن بودن را می‌دهد.

۱۲-۱ سرویسهای TCP

قبل از آنکه به طور مفصل درباره TCP صحبت کنیم، اجازه دهید سرویسهایی که TCP به برنامه‌های لایه کاربرد می‌دهد را بررسی کنیم.

سرویس مخابراتی برنامه به برنامه

مشابه UDP ، TCP مخابره برنامه به برنامه را با استفاده از شماره پورتها برقرار می‌کند (فصل یازدهم را ببینید). جدول ۱۲-۱، لیست برخی از شماره پورتهای کاملاً شناخته شده‌ای که TCP از آنها استفاده می‌کند را نشان می‌دهد.

- **مثال ۱**

همانطور که در فصل یازدهم گفتیم، در یونیکس، پورتهای کاملاً شناخته شده در فایلی به نام /etc/services ذخیره می‌شوند. هرسطری در این فایل، نام سرور و شماره پورتهای کاملاً شناخته شده را نشان می‌دهد. می‌توانیم با استفاده از برنامه grep ، سطر متناظر با کاربرد مطلوب را پیدا کنیم. آنچه که در ذیل می‌آید، پورتهای FTP را نشان می‌دهد.

```
$grep  ftp  /etc/services
ftp-data        20/tcp
ftp-control     21/tcp
```

سرویس تحویل دنباله‌ای از بایتها

TCP برخلاف UDP ، یک پروتوکل ***جریان‌گرا***[2] (برای ارسال دنباله‌ای از بایتها) است. UDP پیغامهایی را از یک برنامه می‌گیرد و آنها را به مقصد می‌رساند. UDP هدر خود را به هر

1- process-to-process or program-to-program
2- Stream-oriented

www.Abiri.ir

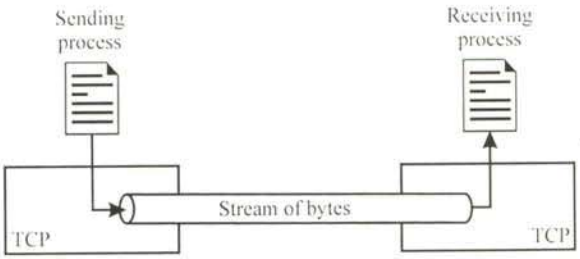
یک از این پیغامها اضافه می‌کند و سپس آنها را برای انتقال، به IP می‌دهد. به پیغامی که حاوی هدر UDP است، یک دیتاگرام UDP نامیده می‌شود و سرانجام به صورت یک بسته IP درمی‌آید. نَه IP و نَه UDP ، هیچ رابطه‌ای را بین دیتاگرامها تشخیص نمی‌دهند. از سوی دیگر TCP، برنامه فرستنده پیغام را قادر می‌سازد تا دنباله‌ای از بایتها را به گونه‌ای ارسال نماید که برنامه گیرنده بتواند آن دنباله را با همان نظم و ترتیب اولیه، دریافت کند. TCP محیطی فراهم می‌نماید که در آن، به نظر می‌رسد دو برنامه توسط یک ***لوله***[1] مجازی به هم متصل شده‌اند و بایتها از طریق این لوله مابین آن دو برنامه، جابجا می‌شوند. این محیط مجازی در شکل ۱۲-۲ نشان داده شده است. برنامه فرستنده، دنباله بایتها را تولید می‌کند (می‌نویسد) و برنامه گیرنده، دنباله بایتها را مصرف می‌کند (می‌خواند).

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FTP,Data	File Transfer Protocol (data connection)
21	FTP,Control	File Transfer Protocol (control connection)
23	TELENT	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name System
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote procedure Call

جدول ۱۲-۱ : پورتهای کاملاً شناخته شده‌ای که TCP به کار می‌برد

1- tube

www.Abiri.ir



شکل ۱۲-۲: تحویل دنباله‌ای از بایتها

* بافرهای فرستنده و گیرنده

از آنجا که ممکن است سرعت ارسال بایتها توسط برنامه‌های فرستنده و گیرنده، متفاوت باشد، لذا TCP برای ذخیره بایتها از بافر استفاده می‌کند. دو نوع بافر وجود دارد: بافر فرستنده و بافر گیرنده. برای هر جهت، یک بافر به کار می‌رود (این بافرها برای مکانیزمهای کنترل خطا و کنترل فلویی که TCP از آنها استفاده می‌کند نیز لازم هستند). یک راه پیاده‌سازی بافر این است که از یک ***آرایه مدور***[1] حاوی خانه‌های یک بایتی، مطابق شکل ۱۲-۳، استفاده کنیم. به منظور ساده‌سازی، دو بافرِ 18 بایتی را نشان داده‌ایم. معمولاً بسته به نوع پیاده‌سازی، بافرهای واقعی به طول صدها یا هزاران بایت می‌باشند. برای سادگی بیشتر، بافرها را هم طول نشان داده‌ایم اما در عمل، غالباً اینطور نیست.

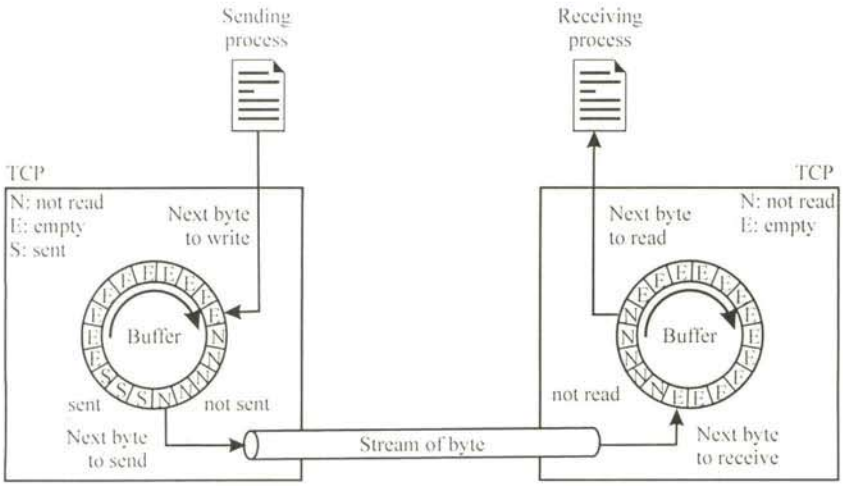
آنچه که در شکل ۱۲-۳ می‌بینید، جابجایی یک طرفه بایتهاست. در سمت ارسال، بافر حاوی سه قسمت متفاوت است. قسمت اول (خانه‌های E) حاوی خانه‌های خالی است که برنامه فرستنده (مولد) می‌تواند آنها را با دیتا پر کند. قسمت دوم (خانه‌های S)، بایتهایی را نشان می‌دهد که ارسال شده‌اند اما هنوز تأییدیه‌ای بر دریافت آنها به دست فرستنده نرسیده است. TCP تا زمانی که تأییدیه دریافت کند، این بایتها را در بافر خود نگه می‌دارد. قسمت سوم (خانه‌های N) بایتهایی را در بر گرفته است که قرار است TCP ارسال کننده، آنها را ارسال نماید. با این حال، همانطور که بعداً در این فصل خواهیم دید، ممکن است TCP بتواند فقط بخشی از این قسمت سوم را ارسال نماید. ممکن است دلیل آن، کندی برنامه گیرنده و یا احتمالاً بروز ازدحام در شبکه باشد. همچنین توجه نمایید که پس از دریافت تأییدیه برای بایتهای ارسال شده قسمت دوم آرایه، خانه‌های مربوط به این قسمت، در اختیار ارسال‌کننده

1- circular-array

www.Abiri.ir

قرار می‌گیرد تا سایر بایتها را بتواند ارسال نماید. به همین دلیل است که از بافر مدور استفاده می‌کنیم.

عمل بافر در سمت گیرنده، ساده‌تر است. بافر مدور به دو قسمت تقسیم شده است: قسمت اول (خانه‌های E) حاوی خانه‌های خالی است که بایتهای دریافتی از شبکه، آنها را پر می‌کند. قسمت دوم (خانه‌های N) حاوی بایتهای دریافت شده‌ای است که برنامه گیرنده می‌تواند آنها را بخواند. وقتی که برنامه گیرنده بایتی را می‌خواند، خانه‌های مربوطه دوباره به صورت خالی در اختیار TCP گیرنده قرار می‌گیرند .



شکل ۱۲-۳: بافرهای فرستنده و گیرنده

❊ سگمنتها

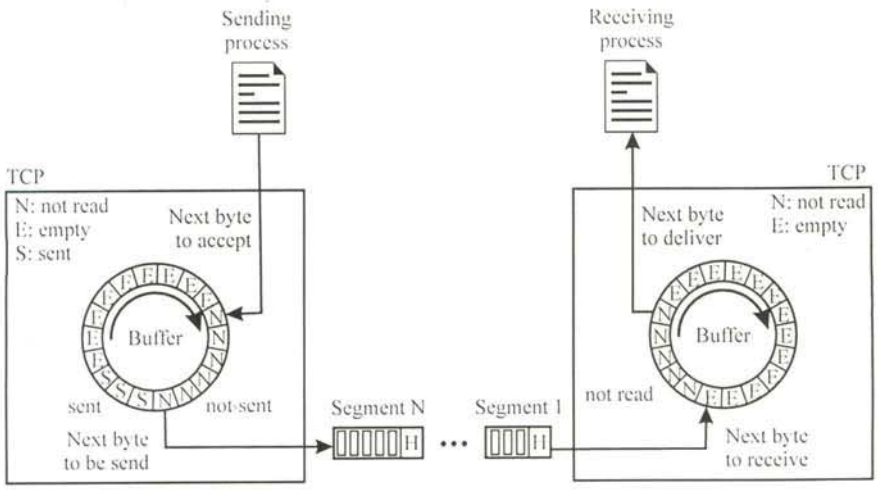
گرچه استفاده از بافر، تفاوت بین سرعت ارسال و سرعت دریافت دیتا را جبران می‌نماید، اما قبل از آنکه بتوانیم دیتایی ارسال کنیم، باید یک قدم بیشتر برداریم. لایه IP به عنوان پروتوکلی که TCP از او سرویس می‌گیرد، بسته‌هایش را به صورت جداگانه ارسال می‌کند و لذا بسته‌های ارسالی IP، یک دنباله پیوسته از بایتها نیست. TCP در لایه انتقال، یک تعدادی از بایتها را به صورت یک گروه درمی‌آورد و نام آن گروه را یک **سگمنت**[1] می‌گذارد. TCP برای انجام کارهای محوله، به هر سگمنت، یک هدر TCP اضافه می‌کند و سپس آن را به لایه IP تحویل می‌دهد تا به ماشین مقصد برساند. لایه IP، هر سگمنت را در یک بسته IP کپسول‌بندی می‌کند و سپس به سوی ماشین مقصد، ارسال می‌نماید. کل این عملیات از دید

1- Segment

www.Abiri.ir

برنامه گیرنده، پنهان باقی می‌ماند. ممکن است سگمنتها بدون نظم و ترتیب اولیه به مقصد برسند و یا گم شوند و یا مخدوش شوند و لذا نیاز به ارسال مجدد داشته باشند. TCP همه این کارها را بدون آنکه برنامه گیرنده از این فعالیتها آگاه باشد، انجام می‌دهد. شکل ۱۲-۴ نشان می‌دهد که چطور سگمنتها از روی بایتهای موجود در بافرها، ساخته می‌شوند.

دقت نمائید که لازم نیست حتماً سایز سگمنتها یکسان باشد. در شکل ۱۲-۴ به منظور سهولت بیشتر، سگمنتی را نشان داده‌ایم که 3 بایت را حمل می‌کند و سگمنت دیگری را نشان داده‌ایم که 5 بایت را حمل می‌نماید. سگمنتها در پیاده‌سازیهای واقعی، صدها و یا هزاران بایت را حمل می‌کنند.



شکل ۱۲-۴: سگمنتهای TCP

سرویس مخابراتی کاملاً دو طرفه

TCP *سرویس کاملاً دوطرفه*[1] ارائه می‌دهد و لذا دیتا می‌تواند به طور همزمان در هر دو طرف جریان داشته باشد. TCP یک بافر فرستنده و یک بافر گیرنده دارد و سگمنتها به هر دو طرف حرکت می‌کنند.

سرویس اتصال‌گرا

TCP برخلاف UDP ، یک پروتوکل اتصالگراست. وقتی که برنامه‌ای در سایت A می‌خواهد دیتایی از برنامه دیگری در سایت B دریافت کند و یا برای او ارسال نماید، مراحل زیر طی می‌شوند:

1- full duplex service

www.Abiri.ir

الف. دو TCP ، یک اتصال بین خود برقرار می‌کنند.
ب. دیتا در هر دو جهت رد و بدل می‌شود.
ج. اتصال خاتمه می‌یابد.

دقت نمائید که این یک اتصال مجازی و نه یک اتصال فیزیکی است. سگمنت TCP در یک بسته IP کپسول‌بندی می‌شود و ممکن است حین انتقال، خارج از نظم و ترتیب اولیه به مقصد برسد، گم شود یا مخدوش شود و لذا نیاز به ارسال مجدد داشته باشد. هر سگمنتی ممکن است از مسیر متفاوتی به مقصد برسد. هیچ اتصال فیزیکی وجود ندارد. TCP محیطی فراهم می‌کند که تحت آن می‌توان جریان پیوسته‌ای از بایتها را با نظم و ترتیب اولیه آنها به گیرنده تحویل داد. به چنین محیطی، ***جریان‌گرا***[1] می‌گویند. این وضعیت مشابه ساختن پلی است که چندین جزیره را به هم وصل می‌کند و بایتها می‌توانند با استفاده از این اتصال برقرار شده، از یک جزیره به جزیره دیگر بروند تا به مقصد برسند.

سرویس قابل اطمینان

TCP یک پروتوکل قابل اطمینان لایه انتقال است. TCP از یک مکانیزم ارسال تأییدیه استفاده می‌کند تا از صحیح و سلامت رسیدن دیتا به مقصد، مطمئن شود.

۱۲-۲ ویژگیهای TCP

TCP برای ارائه سرویسهایی که در بخش قبل به آنها اشاره شد، ویژگیهای متعددی دارد که در این بخش به اجمال آنها را بررسی می‌کنیم.

سیستم شماره‌گذاری

گرچه نرم‌افزار TCP ، کنترلی بر دنباله سگمنتهایی که ارسال و دریافت می‌شوند، دارد اما در هدر TCP ، هیچ فیلدی برای شماره سگمنت وجود ندارد. به جای آن، دو فیلد به نامهای ***شماره ترتیب***[2] و ***شماره تأییدیه***[3] وجود دارد. این دو فیلد، شماره بایتها را نشان می‌دهند و لذا شماره سگمنت نیستند.

1- Stream-Oriented
2- Sequence number
3- acknowledgment number

www.Abiri.ir

*** شماره بایت**

TCP تمام بایتهای دیتا که در یک اتصال منتقل می‌شوند را شماره‌گذاری می‌کند. در هر جهت، شماره‌گذاری به صورت مستقل از جهت دیگر انجام می‌شود. وقتی که TCP بایتهای دیتا را از یک برنامه دریافت می‌کند، آنها را در بافر فرستنده ذخیره می‌کند و شماره‌گذاری می‌نماید. TCP از یک عدد تصادفی بین 0 تا 2^{32}-1 به عنوان شماره اولین بایت استفاده می‌کند و بنابراین شماره‌گذاری، الزاماً از صفر شروع نمی‌شود. به عنوان مثال، اگر اتفاقاً شماره اولین بایت برابر با 1057 باشد و کل دیتایی که باید ارسال شود، 6000 بایت باشد، شماره بایتها از 1057 تا 7056 خواهد بود. این شماره‌گذاری برای کنترل فلو و کنترل خطا نیز مورد استفاده قرار می‌گیرد.

مهم: TCP بایتهایی که در یک اتصال جابجا می‌شوند را شماره‌گذاری می‌کند. شماره‌گذاری بایتها با یک عدد تصادفی شروع می‌شود.

*** شماره ترتیب**

بعد از آنکه بایتها شماره‌گذاری شدند، TCP در فیلد sequence number هر سگمنتی که ارسال می‌شود، یک شماره ترتیب را می‌نویسد. شماره ترتیبی که در هدر سگمنت نوشته می‌شود، برابر با شماره اولین بایتی است که آن سگمنت حمل می‌کند.

**** مثال ۲**

فرض کنید که یک اتصال TCP در حال انتقال فایلی به طول 5000 بایت است. شماره 10001 به اولین بایت داده شده است. اگر دیتا را به صورت 5 سگمنت که هریک حاوی 1000 بایت است، ارسال نماییم، شماره ترتیب هر سگمنت چه عددی خواهد بود؟

راه حل : آنچه به دنبال می‌آید، شماره ترتیب را برای هر سگمنت نشان می‌دهد:

Segment 1 → Sequence Number: 10001 (range: 10001 to 11000)
Segment 2 → Sequence Number: 11001 (range: 11001 to 12000)
Segment 3 → Sequence Number: 12001 (range: 12001 to 13000)
Segment 4 → Sequence Number: 13001 (range: 13001 to 14000)
Segment 5 → Sequence Number: 14001 (range: 14001 to 15000)

مهم: مقداری که فیلد شماره ترتیب هدر یک سگمنت TCP نشان می‌دهد، معرف شماره اولین بایتی است که آن سگمنت حمل می‌کند.

www.Abiri.ir

وقتی که سگمنتی، ترکیبی از اطلاعات کنترلی و دیتا را حمل می‌کند (که به اصطلاح به آن، ***بردوش‌گذاری***[1] گفته می‌شود)، از یک شماره ترتیب استفاده می‌کند. اگر سگمنتی حاوی دیتا نباشد، شماره ترتیب نیز نخواهد داشت. گرچه در این حالت نیز هدر TCP حاوی یک فیلد شماره ترتیب است، اما مقدار موجود در آن معتبر نیست. با این حال، برخی از سگمنتهایی که فقط اطلاعات کنترلی حمل می‌کنند نیز به شماره ترتیب نیاز دارند تا بتوانند از طرف گیرنده، تأییدیه دریافت کنند. این سگمنتها برای برقراری تماس و خاتمه تماس به کار می‌روند. در این سگمنتها، فرض بر این است که یک بایت دیتا را حمل می‌کنند و لذا یک واحد به شماره ترتیب اضافه می‌کنند، اما در واقع دیتایی حمل نمی‌شود. بنابراین اگر شماره ترتیب تصادفی تولید شده برابر با X باشد، شماره اولین بایت دیتا برابر با X+1 خواهد بود. بایت X ، یک ***بایت قلابی***[2] در نظر گرفته می‌شود که برای برقرار کردن اتصال، در یک سگمنت کنترلی به کار می‌رود.

* شماره تأییدیه

همانطور که قبلاً بحث آن آمد، مخابره دیتا در TCP کاملاً دوطرفه است. وقتی که اتصالی برقرار می‌شود، هر دو طرف به طور همزمان می‌توانند دیتا ارسال و دریافت کنند. هر طرف، بایتها را شماره‌گذاری می‌کند و شماره اولین بایت، یک عدد تصادفی است. شماره ترتیب در هر جهت، شماره اولین بایتی که سگمنت حمل می‌کند را نشان می‌دهد. همچنین هر طرف از یک شماره تأییدیه برای تأیید بایتهای دریافتی استفاده می‌کند. با این حال شماره تأییدیه، معرف شماره بایت بعدی است که گیرنده انتظار دریافت آن را دارد. به علاوه شماره تأییدیه، ***تجمعی***[3] است، یعنی گیرنده به شماره آخرین بایتی که صحیح و سالم دریافت کرده است یک واحد اضافه می‌کند و نتیجه حاصل را به عنوان شماره تأییدیه اعلام می‌نماید. عبارت تجمعی به این معناست که اگر شماره تأییدیه‌ای که گیرنده اعلام می‌نماید برابر با 5643 باشد، یعنی گیرنده تمام بایتهای از شماره اولین بایت تا شماره 5642 را دریافت کرده است و اینک منتظر دریافت بایت شماره 5643 است. دقت نمائید که الزاماً، تعداد 5642 بایت دریافت نشده است زیرا که شماره اولین بایت، حتماً از 0 شروع نمی‌شود.

مهم: مقدار فیلد تأییدیه در هدر یک سگمنت TCP ، معرف شماره بایت بعدی است که گیرنده انتظار دریافت آن را دارد. شماره تأییدیه، یک عدد تجمعی است.

1- Piggybacking
2- Phony byte
3- comulative

www.Abiri.ir

کنترل فلو

TCP برخلاف UDP ، مجهز به مکانیزم کنترل فلوست. گیرنده دیتا، حجم دیتایی که فرستنده می‌فرستد را کنترل می‌نماید. به این ترتیب گیرنده از دیتا لبریز نخواهد شد. سیستم شماره‌گذاری، TCP را قادر می‌سازد تا از یک کنترل فلوی ***بایت‌گرا***[1] استفاده کند.

کنترل خطا

TCP برای ارائه یک سرویس قابل اطمینان، مکانیزم کنترل خطا را پیاده‌سازی می‌نماید. گرچه مکانیزم کنترل خطا، از یک سگمنت به عنوان واحد دیتا، برای آشکارسازی خطا (گم شدن یا مخدوش شدن سگمنت) استفاده می‌کند، کنترل خطا به صورت بایت‌گرا اعمال می‌شود.

کنترل ازدحام

TCP برخلاف UDP ، بروز ازدحام در شبکه را مد نظر دارد. حجم دیتایی که فرستنده می‌فرستد، فقط توسط گیرنده کنترل نمی‌شود (مکانیزم کنترل فلو) بلکه سطح ***ازدحام***[2] موجود در شبکه نیز، آن را محدود می‌نماید.

۱۲-۳ سگمنت TCP

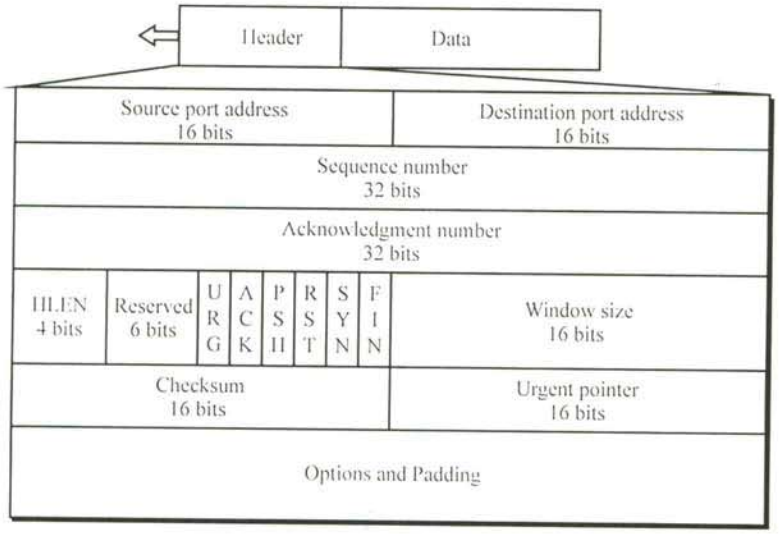
قبل از آنکه با جزئیات بیشتری درباره TCP صحبت کنیم، اجازه دهید درباره واحد دیتای TCP که **سگمنت**[3] نامیده می‌شود، قدری بیشتر بحث نمائیم.

فرمت سگمنت TCP

فرمت یک سگمنت در شکل ۱۲-۵ نشان داده شده است. سگمنت TCP حاوی یک هدر 20 تا 60 بایتی است که به دنبال آن، پیغام صادره از لایه کاربرد می‌آید. اگر Option نداشته باشیم، طول هدر برابر با 20 بایت است و اگر option داشته باشیم، طول هدر می‌تواند حداکثر تا 60 بایت باشد. در این بخش، درباره برخی از فیلدهای هدر TCP صحبت می‌کنیم. همچنان که پیش می‌رویم، معنی و هدف استفاده از این فیلدها روشنتر خواهد شد.

1- byte-oriented
2- Congestion
3- Segment

www.Abiri.ir



شکل ۱۲-۵: فرمت سگمنت TCP

❋ Source port address : آدرس پورت مبدأ

یک فیلد 16 بیتی است و شماره پورتی را در ماشین فرستنده پیغام نشان می‌دهد که پروتوکل لایه کاربرد از آن استفاده می‌کند و هدف از بکارگیری آن، دقیقاً همان چیزی است که در فصل یازدهم درباره فیلد آدرس پورت مبدأ در هدر UDP گفتیم.

❋ Destination port address : آدرس پورت مقصد

یک فیلد 16 بیتی است و شماره پورتی را در ماشین گیرنده پیغام نشان می‌دهد که پروتوکل لایه کاربرد از آن استفاده می‌کند و هدف از بکارگیری آن، دقیقاً همان چیزی است که در فصل یازدهم، درباره فیلد آدرس پورت مقصد در هدر UDP گفتیم.

❋ Sequence number : شماره ترتیب

این فیلد 32 بیتی، شماره‌ای که به اولین بایت دیتای موجود در سگمنت داده شده است را نشان می‌دهد. همانطور که قبلاً گفتیم، TCP یک پروتوکل انتقال جریانی از بایتهاست. برای اطمینان از پیوستگی بایتهای ارسالی، هر بایتی که ارسال می‌شود، شماره دارد. شماره ترتیب به مقصد می‌گوید که اولین بایت این سگمنت، کدام بایت از دنباله بایتهای ارسالی است. در خلال فاز برقراری اتصال (بخش ۱۲-۱۲ را ببینید)، هر طرف از یک مولد عدد تصادفی

www.Abiri.ir

استفاده می‌کند تا یک ***عدد ترتیبی اولیه***[1] تولید نماید و مقدار آن غالباً برای هر جهت دیتا، مقدار متفاوتی است.

❊ Acknowledgment number : شماره تأییدیه

این فیلد 32 بیتی، شماره بایتی را نشان می‌دهد که گیرنده، انتظار دریافت آن را از طرف دیگر دارد. اگر گیرنده، شماره بایت X را با موفقیت از فرستنده دریافت کرده باشد، شماره X+1 را به عنوان شماره تأییدیه ارسال می‌کند. دیتا و شماره تأییدیه را می‌توان به همراه هم ارسال کرد که به آن، ***بردوش‌گذاری***[2] گفته می‌شود.

❊ Header length : طول هدر

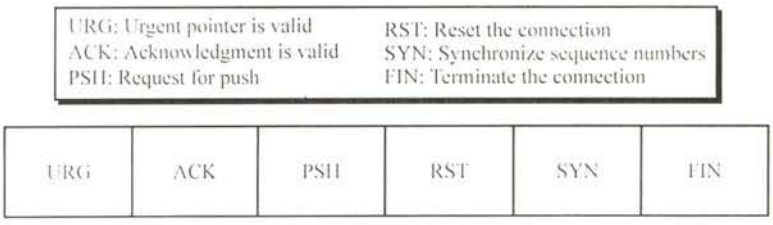
این فیلد 4 بیتی، تعداد 4 بایتی‌های موجود در هدر TCP را نشان می‌دهد. طول هدر می‌تواند بین 20 تا 60 بایت باشد. بنابراین، مقدار این فیلد می‌تواند عددی بین 20/4=5 تا 60/4=15 باشد.

❊ Reserved : رزرو شده

این یک فیلد 6 بیتی است که برای استفاده آتی رزرو شده است.

❊ Flags : پرچمها

همانطور که شکل ۱۲-۶ نشان می‌دهد، این فیلد، 6 بیت کنترلی و یا **پرچم**[3] را نشان می‌دهد. هر بار می‌توان یک یا چند تا از این بیتها را 1 کرد.



شکل ۱۲-۶ : پرچمها و یا فیلد کنترل

1- Initial Sequence Number: ISN
2- Piggybacking
3- flag

www.Abiri.ir

این بیتهای کنترلی، فلو، برقراری و خاتمه اتصال، لغو اتصال و مد انتقال دیتا در TCP را کنترل می‌کنند. جدول ۱۲-۲ ، توصیف مختصری از هر بیت را نشان می‌دهد.

Flag	*Description*
URG	The value of the urgent pointer field is valid
ACK	The value of the acknowledgment field is valid
PSH	Push the data
RST	The connection must be reset
SYN	Synchronize sequence numbers during connection
FIN	Terminate the connection

جدول ۱۲-۲: توصیف بیتهای کنترلی در فیلد کنترل

❋ Window size : سایز پنجره

این فیلد، سایز پنجره‌ای را برحسب بایت نشان می‌دهد که یک سمت، برای سمت دیگر تعیین می‌کند. دقت نمائید که این، یک فیلد 16 بیتی است و به این معناست که ماکزیمم سایز پنجره برابر با 65535 بایت است. معمولاً به آن، سایز ***پنجره دریافت***[1] گفته می‌شود. وقتی که یک طرف، مقداری برای آن تعیین می‌کند، سمت مقابل باید از مقدار تعیین شده، تبعیت کند.

❋ Checksum : جمع آزما

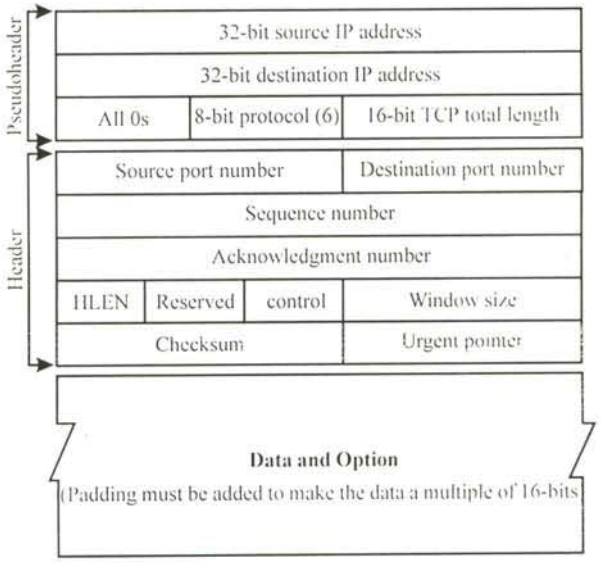
در این فیلد 16 بیتی، مقدار checksum نوشته می‌شود. محاسبه checksum برای TCP همان روندی را طی می‌کند که برای UDP در فصل یازدهم گفتیم. با این حال، محاسبه checksum در UDP ، دلخواه است اما محاسبه checksum در TCP ، اجباری است. از همان شبه هدری که برای UDP گفتیم و به همان منظور، در اینجا نیز استفاده می‌شود، فقط در اینجا مقدار فیلد protocol برابر با 6 است. شکل ۱۲-۷ را ببینید.

❋ Urgent Pointer : نشانه‌گر دیتای اورژانسی

وقتی که سگمنت TCP حاوی دیتای اورژانسی باشد و پرچم URG نیز برابر 1 باشد، مقدار این فیلد 16 بیتی معتبر خواهد بود. عددی که این فیلد نشان می‌دهد را باید با شماره ترتیب جمع کنیم تا شماره اولین بایت اورژانس موجود در دیتای سگمنت، به دست آید.

1- received window: rwnd

www.Abiri.ir



شکل ۱۲-۷ : شبه هدر در TCP

❊ **Options : گزینه‌ها**

هدر TCP می‌تواند حاوی حداکثر 40 بایت گزینه‌های مختلف باشد.

مهم: محاسبه checksum در TCP اجباری است.

کپسول‌بندی

همانطور که شکل ۱۲-۸ نشان می‌دهد، یک سگمنت TCP در یک بسته IP و یک بسته IP در فریم لایه دیتالینک، کپسول‌بندی می‌شوند.



شکل ۱۲-۸: کپسول‌بندی و خارج‌سازی از کپسول

۱۲-۴ فازهای مختلف یک اتصال TCP

TCP اتصالگراست. یک پروتوکل لایه انتقال اتصالگرا، یک مسیر مجازی بین مبدأ و مقصد برقرار می‌کند. سپس تمام سگمنتهای متعلق به یک پیغام از طریق این مسیر مجازی

www.Abiri.ir

ارسال می‌شوند. استفاده از یک مسیر مجازی برای کل پیغام، روند ارسال تأییدیه و نیز ارسال مجدد سگمنتهای گم شده یا مخدوش شده را آسانتر می‌کند. ممکن است تعجب نمائید که چطور TCP که از سرویسهای یک پروتکل فاقد اتصال به نام IP استفاده می‌کند، می‌تواند اتصالگرا باشد. نکته در اینجاست که اتصال TCP ، مجازی و نَه فیزیکی است. TCP در لایه بالاتر از لایه شبکه (IP) کار می‌کند و از سرویسهای IP برای تحویل تک تک سگمنتها به مقصد استفاده می‌کند، اما خودش اتصال را کنترل می‌نماید. اگر سگمنتی گم شود و یا مخدوش شود، مجدداً ارسال می‌گردد. برخلاف TCP ، IP از ارسال مجدد سگمنتها بی اطلاع است. اگر سگمنتی خارج از نوبت به مقصد برسد، TCP آن را نگه می‌دارد تا سایر سگمنتها نیز از راه برسند و نظم و ترتیب اولیه مجدداً برقرار شود، اما IP از این مرتب‌سازی سگمنتها بی خبر است.

در TCP ، انتقال اتصالگرا نیاز به سه فاز دارد: فاز ***برقراری اتصال***[1] ، فاز ***رد و بدل شدن دیتا***[2] و فاز ***قطع اتصال***[3].

فاز برقراری اتصال

TCP دیتا را در مد کاملاً دو طرفه ارسال می‌کند. وقتی که دو TCP در دو ماشین، اتصالی را بین خود برقرار می‌کنند، می‌توانند سگمنتهایی را به طور همزمان برای هم ارسال نمایند. معنای ضمنی آن این است که هر طرف باید درخواست برقراری اتصال با سمت دیگر را بدهد و قبل از آنکه دیتایی رد و بدل شود، تأییدیه سمت مقابل برای برقراری اتصال را دریافت نماید.

* تبادل سه‌گانه

برقراری اتصال در TCP ، طی روندی به نام ***تبادل سه‌گانه***[4] صورت می‌گیرد. در مثال ما، برنامه‌ای به نام سرویس‌گیرنده، می‌خواهد با استفاده از TCP به عنوان پروتوکل لایه انتقال، اتصالی را با برنامه دیگری به نام سرور برقرار کند.

کار از طرف سرور شروع می‌شود. برنامه سرور به TCP خودش می‌گوید که آماده پذیرفتن یک اتصال است و این کار را با دادن فرمان ***به صورت پسیو باز شو***[5] به TCP ، انجام

1- Connection establishment
2- data transfer
3- Connection termination
4- Three-way Handshaking
5- Passive open

www.Abiri.ir

می‌دهد. گرچه اینک سرور آماده پذیرفتن یک اتصال از سوی هر ماشینی در شبکه است، اما خود او نمی‌تواند اتصال را برقرار کند.

برنامه سرویس‌گیرنده به TCP خودش فرمان می‌دهد که ***به صورت اکتیو باز شو***[1]. سرویس‌گیرنده‌ای که می‌خواهد با یک سرور آماده برای برقراری اتصال، اتصالی را برقرار کند، به TCP خودش می‌گوید که می‌خواهد به آن سرور متصل شود. اینک همانطور که شکل ۱۲-۹ نشان می‌دهد، TCP می‌تواند برای برقراری اتصال، تبادل سه‌گانه را آغاز نماید. برای نشان دادن نحوه کار، از دو محور عمودی زمان، یکی در هر طرف، استفاده می‌کنیم. هر سگمنت حاوی مقادیری برای کلیه فیلدهای هدر خود و احتمالاً مقادیری برای فیلدهای گزینه‌های انتخابی می‌باشد، اما ما فقط آن فیلدهایی را نشان می‌دهیم که برای درک تبادل سه‌گانه لازم هستند. بنابراین صرفاً مقدار فیلدهای شماره ترتیب، شماره تأییدیه، پرچمهای کنترلی (فقط آنهایی که 1 شده‌اند) و سایز پنجره را به شرط خالی نبودن آنها، نشان می‌دهیم. سه مرحله‌ای که در تبادل سه‌گانه طی می‌شوند عبارتند از:

الف. سگمنت اول که سگمنت SYN است (بیت SYN در آن، 1 است)، از طرف سرویس گیرنده ارسال می‌شود. این سگمنت برای سنکرونیزاسیون شماره ترتیبها به کار می‌رود. سرویس‌گیرنده در مثال ما یک عدد تصادفی را به عنوان اولین شماره ترتیب انتخاب می‌کند و آن را برای سرور ارسال می‌نماید. به این شماره ترتیب، ***عدد ترتیبی اولیه***[2] می‌گوییم. دقت نمائید که این سگمنت، حاوی شماره تأییدیه نیست و سایز پنجره را نیز تعریف نمی‌کند. زمانی مشخص کردن سایز پنجره با معنی است که سگمنت، حاوی تأییدیه باشد. سگمنت می‌تواند حاوی گزینه‌ها نیز باشد. این سگمنت، یک شماره ترتیب را مصرف می‌کند. وقتی که انتقال دیتا آغاز می‌شود، یک واحد به این شماره ترتیب اضافه می‌شود. می‌توانیم بگوییم که سگمنت SYN حامل هیچ دیتایی نیست، اما می‌توانیم اینطور تصور کنیم که حامل یک بایت دیتای فرضی است.

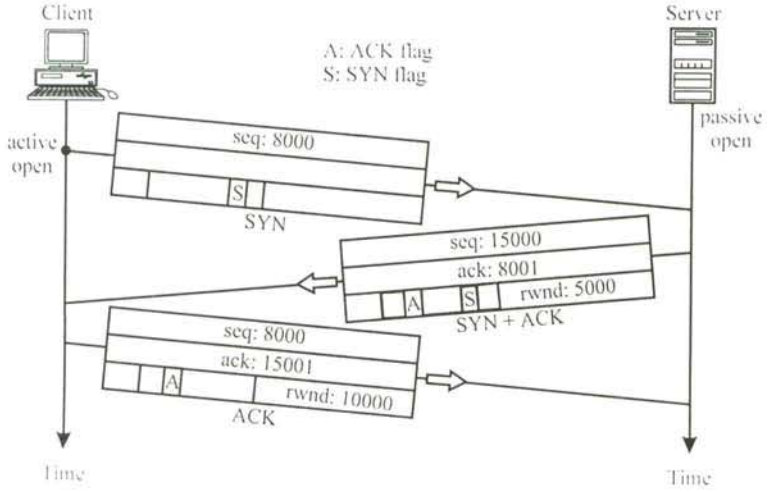
مهم: سگمنت SYN نمی‌تواند حامل دیتا باشد، اما یک شماره ترتیب را مصرف می‌کند.

ب. سگمنت دوم که سگمنت SYN+ACK است از طرف سرور ارسال می‌شود و در این سگمنت، دو پرچم SYN و ACK ست شده‌اند. این سگمنت، دو کار انجام می‌دهد. اول آنکه می‌توان آن را یک سگمنت SYN صادره از سوی سرور تلقی کرد یعنی سرور نیز می‌خواهد اتصالی را با سرویس‌گیرنده برقرار کند. سرور با استفاده از این سگمنت، شماره ترتیب شروعی که او برای شماره‌گذاری بایتهایش از آن استفاده خواهد کرد را به اطلاع سرویس‌گیرنده

1- Active open
2- Initial Sequence Number: ISN

www.Abiri.ir

می‌رساند. دوم آنکه با ست کردن پرچم ACK ، در واقع تأییدیه‌ای بر دریافت سگمنت SYN ارسالی از سوی سرویس‌گیرنده، می‌فرستد و در فیلد شماره تأییدیه، شماره ترتیب بایت بعدی که انتظار دریافت آن از سوی سرویس‌گیرنده دارد را نشان می‌دهد. از آنجا که این سگمنت، حاوی مقداری برای شماره تأییدیه است، باید سایز پنجره rwnd را نیز مشخص کند. سرویس‌گیرنده از سایز پنجره rwnd اعلام شده از سوی سرور، برای کنترل فلو استفاده می‌کند.



شکل ۱۲-۹: برقراری اتصال با استفاده از تبادل سه‌کانه

مهم: سگمنت SYN+ACK نمی‌تواند حامل دیتا باشد، اما یک شماره ترتیب را مصرف می‌کند.

ج. سگمنت سوم را سرویس‌گیرنده می‌فرستد. این فقط یک سگمنت ACK است. او با ست کردن پرچم ACK ، دریافت سگمنت SYN+ACK صادره از سوی سرور را تأیید می‌کند و مقداری که در فیلد شماره تأییدیه می‌نویسد، شماره ترتیب بایت بعدی است که انتظار دریافت آن را در سگمنت بعدی سرور دارد. دقت نمائید که شماره ترتیب در این سگمنت، همان شماره ترتیب موجود در سگمنت SYN است. سگمنت ACK هیچ شماره ترتیبی را مصرف نمی‌کند. سرویس‌گیرنده نیز باید سایز پنجره rwnd را برای سرور تعیین کند. برخی از پیاده‌سازیها به این سگمنت سوم اجازه می‌دهند تا در همین مرحله برقراری اتصال، اولین تکه دیتا را نیز از سرویس‌گیرنده بگیرد و به سمت سرور ببرد. در این حالت، سگمنت سوم باید یک شماره ترتیب جدید داشته باشد که شماره اولین بایت دیتا را نشان دهد. در حالت کلی، معمولاً

www.Abiri.ir

سگمنت سوم حامل دیتا نیست و لذا هیچ شماره ترتیبی را مصرف نمی‌کند اما اگر در مثال ما، می‌خواست حامل دیتا باشد، از شماره ترتیب 8001 استفاده می‌کرد.

مهم: اگر سگمنت ACK حامل دیتا نباشد، هیچ شماره ترتیبی را مصرف نمی‌کند.

* باز شدن به طور همزمان

حالتی که ندرتاً ممکن است پیش آید، این است که هر دو طرف به طور همزمان فرمان « به صورت اکتیو باز شو » را صادر نمایند، یعنی به طور همزمان، هر طرف برای طرف دیگر، یک سگمنت SYN ارسال نماید. در این حالت دو TCP در پاسخ به یکدیگر، سگمنت SYN+ACK را ارسال می‌کنند اما فقط یک اتصال بین آنها برقرار می‌شود. وقتی که در بخش بعدی درباره ***دیاگرام گذر از حالتها***[1] صحبت کردیم، این حالت را نشان خواهیم داد.

* حمله سیل آسای SYN ها

روند برقراری اتصال در TCP ، نسبت به یک مشکل امنیتی جدی به نام ***حمله سیل آسای SYN ها***[2] حساس است. این اتفاق زمانی رخ می‌دهد که یک ***مهاجم***[3] تعداد زیادی سگمنتهای SYN را برای سروری ارسال می‌کند و با جعل آدرس IP مبدأ سگمنتها، اینطور وانمود می‌کند که این سگمنتها از سوی سرویس‌گیرنده‌های مختلفی ارسال شده‌اند. سرور فرض را بر این می‌گذارد که سرویس‌گیرنده‌های مختلفی، درخواست اتصال باز اکتیو را صادر کرده‌اند (سگمنت SYN از سوی سرویس‌گیرنده‌های مختلفی آمده است) و لذا منابع لازم مانند جداول TCB (بعداً در این فصل توضیح می‌دهیم) و تایمرهای لازم را به هر یک از اتصالات درخواستی، اختصاص می‌دهد. سپس سرور، سگمنتهای SYN+ACK را برای سرویس‌گیرنده‌های جعلی ارسال می‌نماید که البته این سگمنتها در شبکه گم می‌شوند زیرا که واقعاً چنین سرویس‌گیرنده‌هایی وجود ندارند. اما در خلال این زمان، مقدار زیادی از منابع سرور، بدون آنکه از آنها استفاده‌ای شود، صرفاً اشغال می‌شوند. اگر در خلال این زمان کوتاه، تعداد سگمنتهای SYN دریافتی زیاد باشد، سرور با کمبود منابع مواجه می‌شود و ممکن است از کار بیفتد. این نوع حمله، یکی از انواع حمله‌هایی است که در حالت کلی، ***حمله عدم توانایی برای سرویس‌دهی***[4] نامیده می‌شود. در این نوع حمله، مهاجم،

1- transition diagram
2- SYN Flooding Attack
3- attacker
4- denial of service attack

www.Abiri.ir

درخواستهای زیادی را برای یک سیستم می‌فرستد و از آنجا که سیستم توان پاسخگویی به آن همه درخواست را ندارد، از کار می‌افتد.

برخی از پیاده‌سازیهای TCP از استراتژیهایی استفاده می‌کنند تا به کمک آنها مشکل حمله سیل‌آسای SYN ها را برطرف نمایند. برخی از آنها، در طول یک فاصله زمانی مشخص، حد بالایی بر تعداد اتصالات قابل درخواست، قرار می‌دهند. برخی دیگر از پیاده‌سازیها، بسته‌هایی که حاوی آدرس مبدأ غیرمجاز هستند را دور می‌اندازند. استراتژی که اخیراً از آن استفاده می‌شود، این است که تخصیص منابع به یک اتصال را تا زمانی که کل اتصال برقرار نشده است، به تعویق می‌اندازند و برای انجام این کار از چیزی به نام ***کوکی***[1] استفاده می‌کنند. SCTP که یک پروتوکل جدید لایه انتقال است و در فصل سیزدهم درباره آن صحبت خواهیم کرد، از این تکنیک استفاده می‌کند.

فاز رد و بدل شدن دیتا

بعد از آنکه اتصال برقرار شد، فاز ***رد و بدل شدن دیتا***[2] به صورت دوطرفه آغاز می‌گردد. سرویس‌گیرنده و سرور می‌توانند دیتا و تأییدیه‌ها را در هر دو جهت ارسال نمایند. در حال حاضر، همین اندازه کافی است که بدانیم انتقال دیتا و تأییدیه، می‌تواند توسط یک سگمنت واحد به طور همزمان صورت بگیرد و به این عمل، گذاشتن تأییدیه بر دوش دیتا یا ***بر دوش گذاری***[3] گفته می‌شود.

شکل ۱۲-۱۰ مثالی را نشان می‌دهد. در این مثال پس از آنکه اتصال برقرار شد (در شکل ۱۲-۱۰ نشان داده نشده است) سرویس‌گیرنده دو سگمنت، هر یک حاوی 1000 بایت دیتا را ارسال می‌نماید. سپس سرور یک سگمنت، حاوی 2000 بایت دیتا را ارسال می‌کند. سرویس‌گیرنده، سگمنت دیگری را می‌فرستد. اولین سه سگمنت حاوی هم دیتا و هم تأییدیه می‌باشند، اما آخرین سگمنت فقط حاوی تأییدیه است، زیرا که سرویس‌گیرنده، دیتای دیگری برای ارسال ندارد. به مقادیر شماره ترتیب و شماره تأییدیه دقت نمایید. پرچم PSH در سگمنتهایی که سرویس‌گیرنده ارسال کرده است، ست شده است تا TCP سمت سرور بداند که به محض دریافت سگمنتها، باید آنها را به سرور تحویل دهد. همانطور که می‌بینیم، این پرچم در سگمنتی که سرور می‌فرستد، ست نشده است و دلیل آن این است که اکثر پیاده‌سازیهای TCP در ست کردن یا ست نکردن این پرچم، حق انتخاب دارند.

1- cookie
2- data transfer
3- piggybacking

www.Abiri.ir

* ارسال بلادرنگ دیتا

دیدیم TCP فرستنده از یک بافر برای ذخیره دنباله بایتهایی که از برنامه فرستنده می‌آید، استفاده می‌نماید. TCP فرستنده می‌تواند سایز سگمنتهای ارسالی خود را انتخاب کند. TCP گیرنده نیز بایتهای دریافتی را در بافر قرار می‌دهد و زمانی که برنامه گیرنده آماده دریافت آنها باشد و یا تحویل آنها در آن لحظه برای TCP گیرنده، مناسب باشد، بایتهای دریافتی را به برنامه گیرنده تحویل می‌دهد. این نوع قابلیت انعطاف، کارآیی TCP را بالا می‌برد.

اما مواقعی هست که برنامه‌ها نیازی به این قابلیت انعطاف ندارند. به عنوان مثال برنامه‌ای را در نظر بگیرید که ***به طور محاوره‌ای***[1] با برنامه دیگری به مخابره دیتا می‌پردازد. این برنامه می‌خواهد کلیدی که فشار داده شده است را برای برنامه سمت دیگر ارسال نماید و سریعاً پاسخی را دریافت نماید و لذا این برنامه نمی‌تواند انتقال توأم با تأخیر و یا تحویل توأم با تأخیر را بپذیرد.

TCP می‌تواند پاسخگوی چنین وضعیتی باشد. برنامه سمت فرستنده می‌تواند درخواست **ارسال بلادرنگ دیتا**[2] را بدهد. به این معنی که TCP فرستنده نباید منتظر پر شدن پنجره بنشیند، بلکه باید بلافاصله یک سگمنت تولید کند و آن را ارسال نماید. برای این کار، TCP فرستنده باید بیت ارسال بلادرنگ یعنی ***بیت PSH***[3] را ست کند تا TCP گیرنده بفهمد که سگمنت دریافتی حامل دیتایی است که هرچه سریعتر باید به برنامه گیرنده تحویل داده شود و نباید منتظر بماند تا سایر دیتاها را نیز دریافت کند و سپس همه را با هم به برنامه گیرنده تحویل دهد.

گرچه برنامه‌ها می‌توانند درخواست ارسال بلادرنگ دیتا را بدهند، اما اکثر پیاده‌سازیهای فعلی TCP به چنین درخواستی، اهمیتی نمی‌دهند. خود TCP بین استفاده یا عدم استفاده از ویژگی ارسال بلادرنگ دیتا، حق انتخاب دارد.

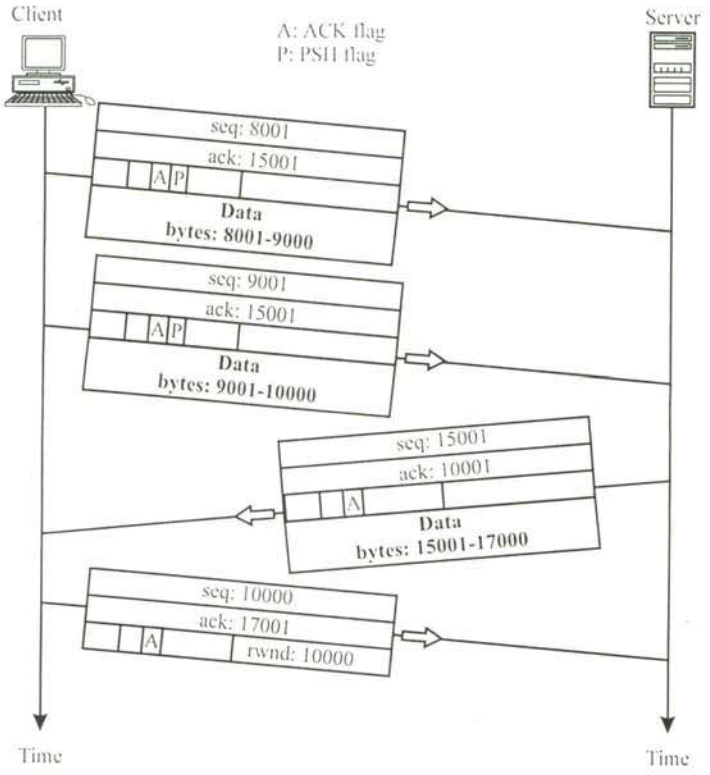
* دیتای اورژانس

TCP یک پروتوکل ***جریان‌گرا***[4] است. به این معنی که دیتا به صورت جریان پیوسته‌ای از بایتها، از برنامه به دست TCP می‌رسد. هر بایت دیتا، محل مشخصی در این جریان پیوسته از بایتها دارد. اما مواردی هست که برنامه فرستنده باید ***بایتهای اورژانسی***[5] ارسال نماید و گیرنده باید آنها را خارج از نوبت، پردازش نماید. به عنوان مثال فرض کنید که برنامه فرستنده

1- Interactively
2- Push
3- Push bit
4- Stream-Oriented
5- urgent bytes

www.Abiri.ir

در حال ارسال دیتایی است که باید توسط برنامه گیرنده پردازش شود اما وقتی که برنامه فرستنده، نتیجه پردازش گیرنده را دریافت می‌کند، درمی‌یابد که همه چیز به اشتباه پردازش شده است. برنامه فرستنده می‌خواهد که از ارسال بقیه دیتا صرفنظر کند، اما می‌بیند حجم زیادی از دیتا را ارسال کرده است. اگر برنامه فرستنده از یک Control C استفاده نماید، این دو کاراکتر در انتهای بافر TCP گیرنده قرار خواهند گرفت. بنابراین زمانی این دو کاراکتر به دست برنامه گیرنده می‌رسند که همه دیتاها به غلط پردازش شده‌اند.



شکل ۱۲-۱۰: انتقال دیتا

راه حل، ارسال یک سگمنت است که در آن بیت اورژانس یعنی ***بیت URG***[1] ست شده باشد. برنامه فرستنده به TCP فرستنده می‌گوید که آن قطعه از دیتا، اورژانس است. TCP

1- urgent bit

www.Abiri.ir

فرستنده یک سگمنت می‌سازد و دیتای اورژانسی را در انتهای سگمنت قرار می‌دهد. قسمت ابتدائی سگمنت، می‌تواند حامل دیتای معمولی باشد که از بافر می‌آید. عددی که ***فیلد نشانه‌گر اورژانس***[1] نشان می‌دهد را باید با شماره ترتیب، جمع کنیم تا شماره اولین بایت اورژانس موجود در دیتای سگمنت، به دست آید.

وقتی که TCP گیرنده، سگمنتی را دریافت می‌کند که در آن بیت URG ست شده است، با استفاده از نشانه‌گر اورژانس، دیتای اورژانس را از سگمنت خارج می‌سازد و خارج از نوبت، آن را به برنامه گیرنده تحویل می‌دهد.

فاز قطع اتصال

هریک از دو طرفی که در تبادل دیتا شرکت می‌کنند (سرور و سرویس‌گیرنده) می‌توانند اتصال را قطع کنند اما معمولاً سرویس‌گیرنده به اتصال خاتمه می‌دهد. اکثر پیاده‌سازیهای امروزی TCP ، از دو روش برای خاتمه اتصال استفاده می‌کنند: تبادل سه‌گانه و تبادل چهارگانه با استفاده از ***گزینه نیمه- بسته***[2].

* تبادل سه‌گانه

همانطور که شکل ۱۲-۱۱ نشان می‌دهد، اکثر پیاده‌سازیهای امروزی TCP از ***تبادل سه‌گانه***[3] برای خاتمه اتصال استفاده می‌کنند.

الف. در شرایط طبیعی، TCP سمت سرویس‌گیرنده، پس از دریافت فرمان قطع از سوی برنامه سرویس‌گیرنده، اولین سگمنت قطع اتصال به نام سگمنت FIN که در آن پرچم FIN ست شده است را ارسال می‌نماید. دقت نمائید که مطابق شکل ۱۲-۱۱، سگمنت FIN ، هم می‌تواند حاوی آخرین تکه دیتای ارسالی سرویس‌گیرنده و هم صرفاً یک سگمنت کنترلی باشد. اگر فقط یک سگمنت کنترلی باشد، فقط یک شماره ترتیب را مصرف می‌کند.

مهم: اگر سگمنت FIN حامل دیتا نباشد، فقط یک شماره ترتیب را مصرف می‌کند.

ب. TCP سمت سرور پس از دریافت سگمنت FIN ، وضعیت را به اطلاع برنامه سرور می‌رساند و دومین سگمنت، یعنی سگمنت FIN+ACK را می‌فرستد تا هم تأییدیه‌ای بر دریافت سگمنت FIN از سوی سرویس‌گیرنده باشد و هم درخواست قطع اتصال را صادر کرده

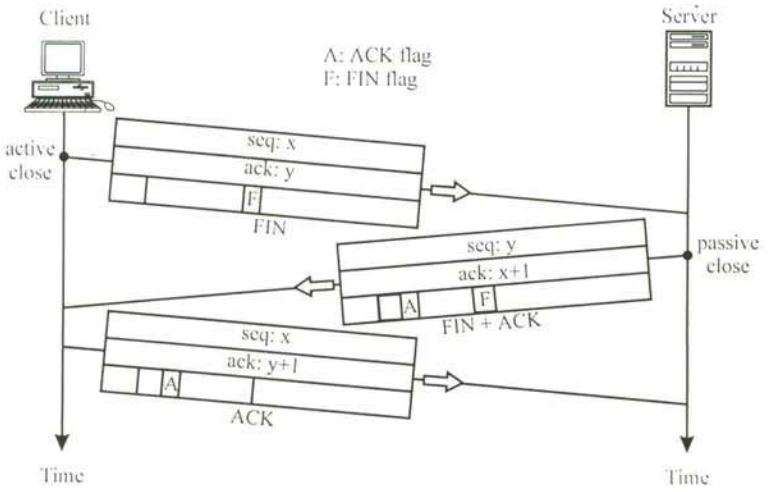
1- urgent pointer field
2- half-close option
3- three-way handshaking

www.Abiri.ir

باشد. این سگمنت نیز می‌تواند حاوی آخرین تکه دیتا از سوی سرور باشد. اگر حامل دیتا نباشد، فقط یک شماره ترتیب را مصرف می‌کند.

مهم: اگر سگمنت FIN+ACK حامل دیتا نباشد، فقط یک شماره ترتیب را مصرف می‌کند.

ج. TCP سمت سرویس‌گیرنده، آخرین سگمنت که یک سگمنت ACK است را به منظور تأییدیه‌ای بر دریافت سگمنت FIN از TCP سمت سرور، ارسال می‌نماید. این سگمنت حاوی شماره تأییدیه است (y+1) و مقدار آن، یک واحد بیشتر از شماره ترتیب سگمنت FIN دریافتی از سوی سرور است (y). این سگمنت نمی‌تواند حامل دیتا باشد و هیچ شماره ترتیبی را مصرف نمی‌کند.



شکل ۱۲-۱۱: قطع اتصال با استفاده از تبادل سه‌گانه

*** تبادل چهارگانه به همراه نیمه بستن اتصال**

در TCP ، یک طرف می‌تواند همچنان به دریافت دیتا ادامه دهد، اما ارسال دیتا را متوقف نماید. به این حالت، ***نیمه- بسته***[1] گفته می‌شود. گرچه هر طرف می‌تواند درخواست نیمه بستن اتصال را بدهد، اما معمولاً چنین درخواستی را سرویس‌گیرنده می‌دهد. چنین وضعیتی می‌تواند زمانی رخ دهد که سرور، قبل از پردازش دیتا، همه دیتا را نیاز داشته باشد.

1- half-close

www.Abiri.ir

برنامه‌سازی برای سرور می‌فرستد. سرور قبل از شروع برنامه‌سازی، باید همه دیتا ر
ت نماید.

بنابراین سرویس‌گیرنده، پس از ارسال همه دیتا، می‌تواند درخواست بسته بستن اتصال
هد یعنی او دیگر دیتایی ندارد که بفرستد و باید منتظر بماند تا سرور، دیتای برنامه‌شده ر
او بفرستد. سرور پس از دریافت تمام دیتا، برای برداشتن دیتای دریافتی، به زمان نیاز دا
ه دیگر دیتایی دریافت نمی‌کند. اما ارسال دیتای آن باید همچنان باز باشد تا پس ا
کردن دیتا، بتواند آن را برای سرویس‌گیرنده بفرستد.

شکل ۱۲-۱۲ نمونه‌ای از نحوه بستن اتصال را نشان می‌دهد. سرویس‌گیرنده با ارسال
گمنت FIN ، اتصال را به صورت نیمه بسته درمی‌آورد. سرور با ارسال سگمنت ACK
سته شدن اتصال را می‌پذیرد و انتقال دیتا از سمت سرویس‌گیرنده به سمت سرور پایان
د. با این حال، سرور هنوز می‌تواند دیتا را ارسال نماید. وقتی که سرور، تمام دیتای
ین شده را ارسال کرد، یک سگمنت FIN ارسال می‌کند که توسط یک ACK از سوی
س‌گیرنده، تأیید می‌شود.

بعد از نیمه بستن اتصال، دیتا می‌تواند از سمت سرور به سمت سرویس‌گیرنده ر
دها از سمت سرویس‌گیرنده به سمت سرور بروند. اما سرویس‌گیرنده نمی‌تواند دیتای
ی برای سرور ارسال نماید. به شماره ترتیب‌هایی که به کار برده‌اید، دقت نمایید. سگمنت
سگمنت ACK) هیچ شماره ترتیبی را مصرف نمی‌کند. گرچه سرویس‌گیرنده شمار
۱۰۱ را دریافت کرده است و انتظار دریافت شماره ترتیب ۷ را دارد، شماره ترتیب سرو
همان ۱۰۱ است. وقتی که سرانجام اتصال بسته می‌شود، شماره ترتیب آخرین سگمنت
۸ هنوز برابر با ۸ است زیرا که در خلال جابجایی دیتا در این جهت، هیچ شماره ترتیبی
نشده است.

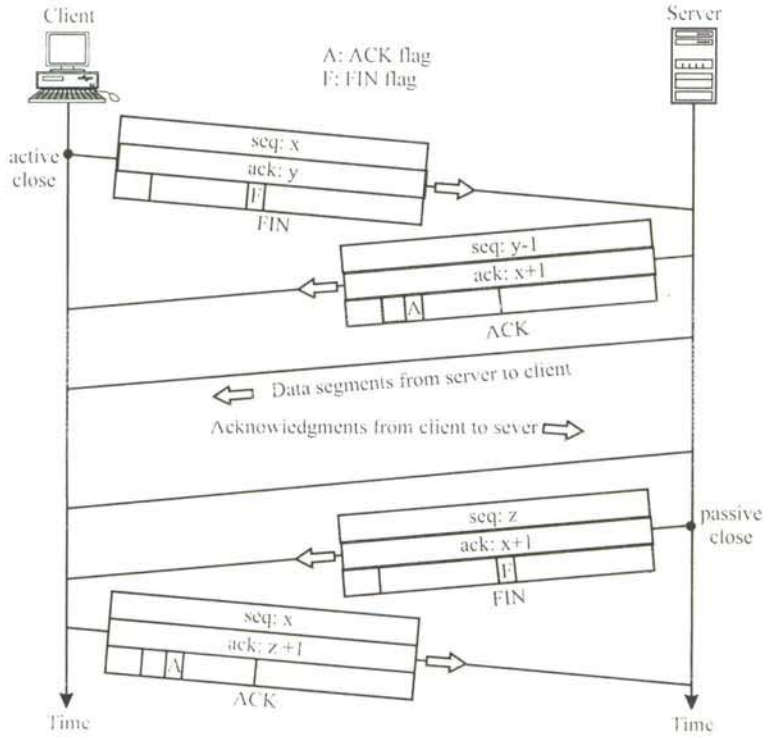
❋ حالتهای خاص

حالتهای خاصی نیز وجود دارند که منجر به خاتمه اتصال می‌شوند و به شرح زیرند:

❋❋ ریست کردن اتصال

www.Abiri.ir

TCP یک طرف ممکن است درخواست برقراری یک اتصال را رد کند، اتصالی را لغو
و یا به یک اتصال غیرفعال، خاتمه دهد. همه این کارها توسط بیت ریست یعنی



شکل ۱۲-۱۲: نیمه بسته شدن اتصال

❊❊ رد کردن یک اتصال

فرض کنید که TCP یک طرف، برقراری اتصالی را به پورتی در سمت دیگر، درخواست کرده است اما آن پورت، بسته است. TCP سمت دیگر ممکن است سگمنتی بفرستد که بیت RST آن ست شده باشد و با این کار، درخواست برقراری اتصال را رد کند.

❊❊ لغو یک اتصال

ممکن است به دلیل شرایط غیرطبیعی، TCP یک طرف بخواهد اتصالی را لغو نماید. او می‌تواند یک سگمنت RST برای بستن اتصال بفرستد. در بخش بعدی، مثالی از این حالت را نیز نشان خواهیم داد.

www.Abiri.ir

❊ خاتمه دادن به یک اتصال غیر فعال

TCP یک طرف ممکن است TCP سمت دیگر را برای یک مدت طولانی، غیر فعال ببیند. او می‌تواند یک سگمنت RST بفرستد تا به اتصال خاتمه دهد. روند کار مشابه با لغو یک اتصال است.

۱۲-۵ دیاگرام گذر از حالتها

برای پیگیری تمام رویدادهای مختلفی که حین برقراری اتصال، نقل و انتقال دیتا و در نهایت خاتمه اتصال رخ می‌دهند، نرم‌افزار TCP به صورت ***یک ماشین با تعداد حالات محدود***[1] پیاده‌سازی می‌شود. چنین ماشینی از تعداد محدودی از حالتها ساخته شده است. در هر لحظه، ماشین در یکی از این حالتها می‌باشد. او تا زمانی که رویدادی رخ نداده است، در آن حالت باقی می‌ماند. آن رویداد ممکن است سبب گردد که ماشین، عملیاتی را انجام دهد. به عبارت دیگر، آن رویداد، یک ورودی است که به یک حالت ماشین اعمال می‌شود. آن رویداد می‌تواند باعث تغییر حالت ماشین و نیز تولید یک خروجی گردد. جدول ۱۲-۳ یازده حالت مختلف TCP را نشان می‌دهد.

State	*Description*
CLOSED	There is no connection
LISTEN	Passive open received; waiting for SYN
SYN-SENT	SYN sent; waiting for ACK
SYN-RCVD	SYN + ACK sent; waiting for ACK
ESTABLISHED	Connection established; data transfer in progress
FIN-WAIT-1	First FIN sent;waiting for ACK
FIN-WAIT-2	ACK to first FIN received; waiting for second FIN
CLOSE-WAIT	First FIN received, ACK sent; waiting for application to close
TIME-WAIT	Second FIN received, ACK sent; waiting for 2MSL time-out
LAST-ACK	Second Fin sent; waiting for ACK
CLOSING	Both sides have decided to close simultaneously

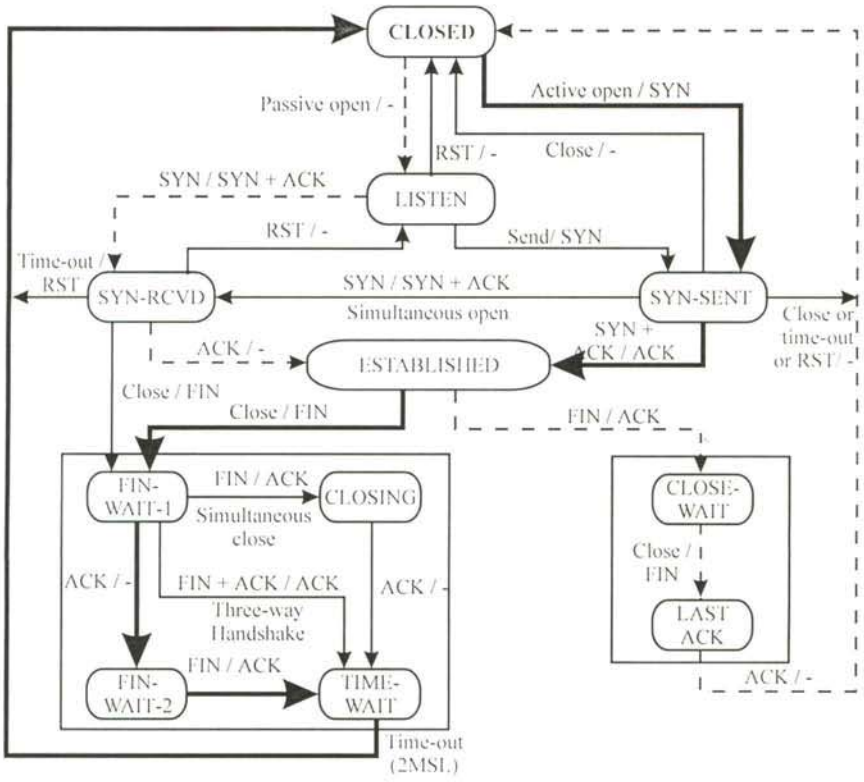
جدول ۱۲-۳: یازده حالت مختلف TCP

1- Finite State Machine

www.Abiri.ir

شکل ۱۲-۳ ***دیاگرام گذر از حالتهای***[1] TCP را نشان می‌دهد که در آن ورودی‌ها و خروجی‌های مختلف، نشان داده شده‌اند.

بیضی‌ها معرف حالتها هستند. گذر از یک حالت به حالت دیگر، توسط خطوط جهت‌دار نشان داده شده‌اند. در کنار هر خط، دو رشته کاراکتری که مابین آنها یک slash (/) است، نوشته شده است. رشته کاراکتری سمت چپ، ورودی است که TCP دریافت می‌کند. رشته کاراکتری سمت راست، خروجی است که TCP می‌فرستد. شکل ۱۲-۱۳ دیاگرام گذر از حالتها را هم برای سرویس‌گیرنده و هم برای سرور نشان می‌دهد. خط چین‌ها، گذرهای سرور از یک حالت به حالت دیگر را نشان می‌دهند، خطوط ضخیم، گذرهای سرویس‌گیرنده از یک حالت به حالت دیگر را نشان می‌دهند. خطوط نازک، حالتهای خاص را نشان می‌دهند.



شکل ۱۲-۱۳ : دیاگرام گذر از حالتهای نرم‌افزار TCP

1- transition diagram

www.Abiri.ir

سناریوها

در این بخش برای درک حالتهای TCP و دیاگرامهای گذر، به بررسی برخی از سناریوها می‌پردازیم.

* برقراری اتصال و در نهایت، خاتمه اتصال

شکل ۱۳-۱۴ سناریویی را نشان می‌دهد که در آن برنامه سرور، فرمانهای ***به صورت پسیو باز شو***[1] و سپس بعد از طی مراحل میانی، فرمان ***به صورت پسیو بسته شو***[2] را صادر کرده است و برنامه سرویس‌گیرنده، فرمان ***به صورت اکتیو باز شو***[3] و پس از طی مراحل میانی، فرمان ***به صورت اکتیو بسته شو***[4] را صادر کرده است. وضعیت ***نیمه- بسته***[5] نیز قابل مشاهده است. حالتهای مختلف سرور و سرویس‌گیرنده و نیز طول زمانی نسبی آنها، روی محورهای عمودی زمان، نشان داده شده‌اند.

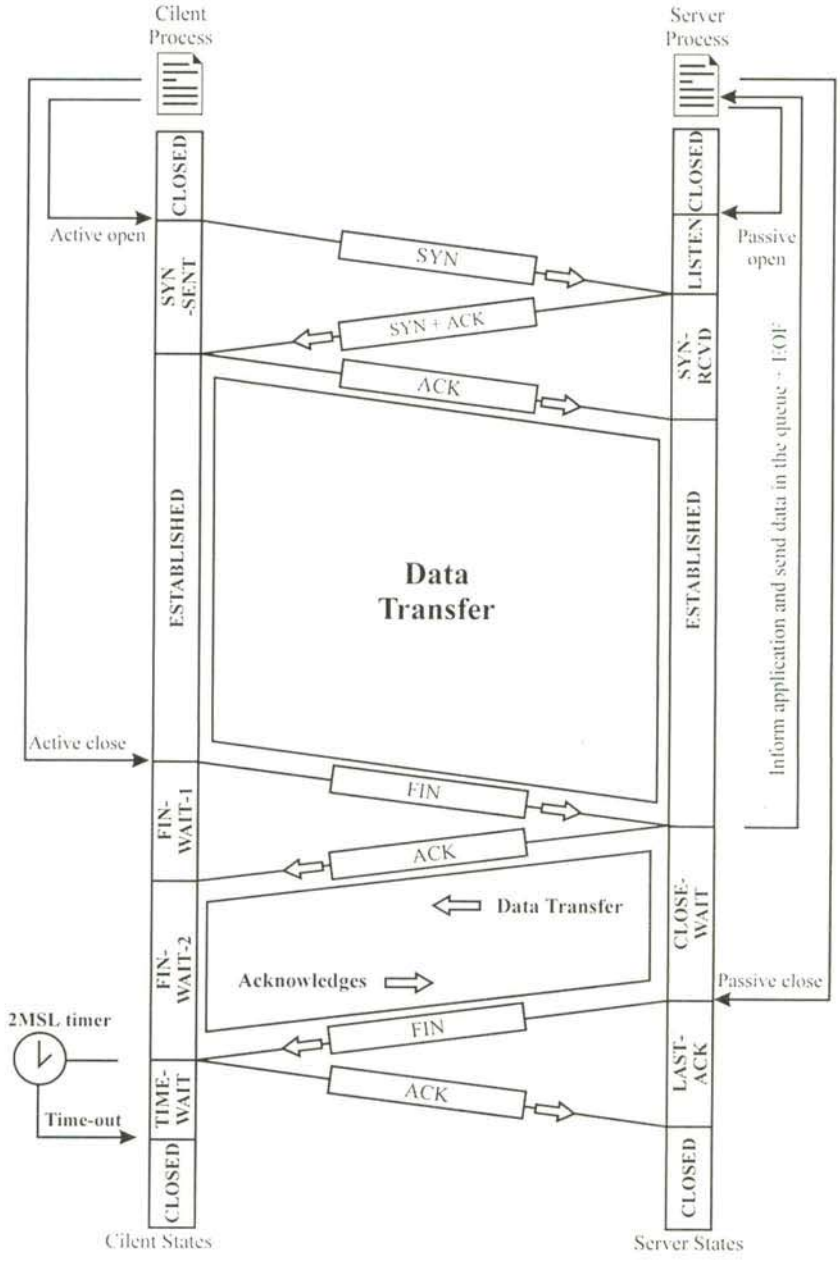
* حالتهای سرویس‌گیرنده

برنامه سرویس‌گیرنده، فرمان « به صورت اکتیو باز شو » را به TCP خودش می‌دهد و به این ترتیب، برقراری اتصالی به یک آدرس سوکت مشخص شده را مطالبه می‌نماید. TCP برای برقراری اتصال، یک سگمنت SYN ارسال می‌کند و وارد حالت SYN-SENT می‌شود. پس از آنکه TCP ، سگمنت SYN+ACK را از سمت سرور دریافت کرد، یک سگمنت ACK ارسال می‌نماید و وارد حالت ESTABLISHED می‌شود. دیتا در هر دو جهت رد و بدل می‌شود و تأییدیه‌های مربوطه ارسال می‌شوند.

وقتی که برنامه سرویس‌گیرنده، دیتای بیشتری برای ارسال ندارد، فرمان « به صورت اکتیو بسته شو » را صادر می‌نماید. TCP برای خاتمه اتصال، یک سگمنت FIN ارسال می‌کند و وارد حالت FIN-WAIT-1 می‌شود. وقتی که او ACK مربوط به FIN ارسالی را از سمت سرور دریافت کرد، وارد حالت FIN-WAIT-2 می‌شود و تا زمانی که سگمنت FIN را از سرور دریافت نکرده است، در همان حالت باقی می‌ماند. وقتی که سگمنت FIN از سوی سرور دریافت شد، سرویس‌گیرنده، یک سگمنت ACK ارسال می‌کند و وارد حالت TIME-WAIT می‌شود و

1- Passive open
2- Passive close
3- active open
4- active close
5- half-close

www.Abiri.ir



شکل ۱۲-۱۴: برقراری اتصال، رد و بدل شدن دیتا و در نهایت، خاتمه اتصال

www.Abiri.ir

تایمری را روی دو برابر ***حداکثر طول عمر یک سگمنت***[1] در شبکه، ست می‌کند. MSL ماکزیمم زمانی است که یک سگمنت قبل از دور انداخته شدن می‌تواند در شبکه، حرکت کند. دقت نمایید که یک سگمنت TCP ، در یک بسته IP کپسول‌بندی می‌شود و خود بسته IP ، طول عمر محدودی (برابر با مقدار TTL) دارد. وقتی که بسته IP ، به پایان عمر خود برسد و دور انداخته شود، سگمنتTCP ای که در آن کپسول‌بندی شده است نیز دور انداخته می‌شود. مقدار متداول برای MSL بین 30 ثانیه تا 1 دقیقه است. دو دلیل برای داشتن حالت TIME-WAIT و تایمر 2 MSL وجود دارد :

الف. TCP سمت سرور، پس از ارسال FIN ، تایمری را ست می‌کند. اگرACK ای که سرویس‌گیرنده پس از دریافت این FIN ، ارسال می‌کند، گم شود، تایمر فوق به پایان مهلت مقرر برای دریافت ACK از سوی سرویس‌گیرنده می‌رسد و لذا سرور دوباره FIN را ارسال می‌کند. حال اگر سرویس‌گیرنده، قبل از آنکه به اندازه 2 MSL صبر کند، بلافاصله وارد حالت CLOSED شود و اتصال را ببندد، سرویس‌گیرنده هرگز FIN ای که سرور مجدداً ارسال کرده است را دریافت نمی‌کند و لذا سرور هرگز آخرین ACK را از سوی سرویس‌گیرنده دریافت نخواهد کرد و بنابراین سرور، اتصال را نمی‌بندد. تایمر 2 MSL سبب می‌گردد که سرویس‌گیرنده، به اندازه کافی صبر کند تا اگر آخرین ACK صادره از سوی او گم شود (یک MSL) و سرور، FIN دیگری بفرستد (یک MSL دیگر) ، این FIN قبل از آنکه اتصال بسته شود، به دست سرویس‌گیرنده برسد. اگر زمانی که سرویس‌گیرنده در حالت TIME-WAIT است، FIN جدیدی از راه برسد، سرویس‌گیرنده یک ACK جدید می‌فرستد و مجدداً تایمر 2 MSL خود را ست می‌کند.

ب. کپی سگمنتی مربوط به اتصال قبلی که هنوز در شبکه وجود دارد، ممکن است در اتصالی که جدیداً باز می‌شود، ظاهر گردد. فرض کنید که سرویس‌گیرنده و سرور، اتصالی را بسته‌اند. پس از گذشت یک مدت زمان کوتاه، آنها اتصال جدیدی را با همان آدرس سوکتهای قبلی (آدرس IP مبدأ و مقصد مشابه و پورتهای مبدأ و مقصد مشابه) باز می‌کنند. این اتصال جدید، ***ظهور جدیدی***[2] از اتصال قبلی است. کپی یک سگمنت از اتصال قبلی که هنوز در شبکه سرگردان است، ممکن است به داخل این اتصال جدید وارد شود و این توهم را به وجود آورد که متعلق به همین اتصال جدید است و این حالت زمانی رخ می‌دهد که یک مدت زمان کافی بین بسته شدن اتصال قبلی و باز شدن اتصال فعلی با همان آدرس سوکتها، سپری نشده باشد. برای جلوگیری از وقوع چنین مشکلی، TCP تا مدت زمانی برابر با MSL 2 از لحظه بسته شدن اتصال قبلی نگذشته باشد، اجازه باز شدن اتصال جدیدی با همان آدرس سوکتهای اتصال

1- Maximum Segment Lifetime : MSL
2- incarnation

www.Abiri.ir

قبلی را نمی‌دهد. با این حال برخی از پیاده‌سازیها، در صورتی که عدد ترتیبی اولیه (ISN) اتصال جدید، بزرگتر از آخرین شماره ترتیب به کار رفته در اتصال قبلی باشد، از این قاعده صرفنظر می‌کنند.

* حالتهای سرور

در این سناریو، برنامه سرور « فرمان به صورت پسیو باز شو » را صادر می‌نماید. این فرمان باید قبل از صدور فرمان « به صورت اکتیو باز شو » از سوی سرویس‌گیرنده باشد. پس از صدور این فرمان، TCP سمت سرور وارد حالت LISTEN می‌شود و تا زمانی که یک سگمنت SYN از سوی سرویس‌گیرنده دریافت نکرده است، به صورت پسیو در همین حالت، باقی می‌ماند. وقتی که TCP سمت سرور، یک سگمنت SYN دریافت کرد، یک سگمنت SYN+ACK ارسال می‌کند و وارد حالت SYN-RCVD می‌شود و سپس صبر می‌کند تا سگمنت ACK از سوی سرویس‌گیرنده را دریافت کند. وقتی که سرور، سگمنت ACK را دریافت کرد، وارد حالت ESTABLISHED می‌شود و در این حالت است که رد و بدل شدن دیتا، آغاز می‌شود.

TCP سمت سرور، در این حالت باقی می‌ماند تا اینکه یک سگمنت FIN از TCP سمت سرویس‌گیرنده، دریافت کند و نشان دهد که سرویس‌گیرنده دیتای دیگری برای ارسال ندارد و متقاضی بسته شدن اتصال است. سرور در این لحظه، یک ACK را برای سرویس‌گیرنده ارسال می‌کند، دیتاهایی که در صفهایش مانده‌اند را می‌فرستد و سپس وارد حالت CLOSE-WAIT می‌شود. در این سناریو، فرض را بر نیمه- بسته شدن اتصال می‌گذاریم. TCP سمت سرور، هنوز می‌تواند برای سرویس‌گیرنده دیتا بفرستد و تاییدیه‌های مربوطه را دریافت کند، اما سرویس‌گیرنده دیگر نمی‌تواند دیتایی بفرستد. به این معنی که TCP سمت سرور در این حالت باقی می‌ماند تا برنامه سمت سرور، فرمان « به صورت پسیو بسته شو » را صادر نماید. اکنون TCP سمت سرور یک سگمنت FIN را برای سرویس‌گیرنده می‌فرستد تا نشان دهد که او نیز می‌خواهد اتصال را ببندد و سپس وارد حالت LAST-ACK می‌شود. TCP سمت سرور در این حالت باقی می‌ماند تا ACK را از سمت سرویس‌گیرنده دریافت کند و پس از دریافت آن، وارد حالت CLOSED می‌شود. فاز خاتمه اتصال که از لحظه ارسال اولین FIN از سوی سرویس‌گیرنده آغاز می‌شود و در لحظه ارسال آخرین ACK از سمت سرویس‌گیرنده، خاتمه می‌یابد، به نام ***تبادل چهارگانه***[1] موسوم است .

1- Four-way handshaking

www.Abiri.ir

* خاتمه اتصال با استفاده از تبادل سه‌گانه

همانطور که قبلاً اشاره شد، در فاز خاتمه اتصال، می‌توان از روش دیگری به نام تبادل سه‌گانه نیز استفاده کرد. در این حالت، سمتی که اولین سگمنت FIN را دریافت می‌کند، یک سگمنت FIN+ACK را ارسال می‌کند و به این ترتیب سگمنتهای FIN و ACK به کار رفته در تبادل چهارگانه، به صورت یک سگمنت در هم ادغام می‌شوند. سرویس‌گیرنده، حالت FIN-WAIT-2 را حذف می‌کند و مستقیماً وارد حالت TIME-WAIT می‌شود. شکل ۱۲-۱۵ نشان می‌دهد که سرویس‌گیرنده بعد از فاز رد و بدل شدن دیتا، فرمان « به صورت اکتیو بسته شو » را صادر کرده است. TCP سمت سرویس‌گیرنده، یک سگمنت FIN می‌فرستد و وارد حالت FIN-WAIT-1 می‌شود. TCP سمت سرور، پس از دریافت سگمنت FIN ، کلیه دیتاهای موجود در صفهای خود را با گذاشتن شاخص EOF در انتهای آنها، به برنامه سرور می‌دهد و معنی شاخص EOF این است که اتصال باید بسته شود. او وارد حالت CLOSE-WAIT می‌شود، اما ارسال ACK برای FIN دریافتی از سوی سرویس‌گیرنده را تا زمانی که یک درخواست بسته شدن پسیو را از سوی برنامه خود دریافت نکرده است، به تعویق می‌اندازد. پس از دریافت فرمان بسته شدن پسیو، یک سگمنت FIN+ACK برای سرویس‌گیرنده می‌فرستد و وارد حالت LAST-ACK می‌شود و منتظر آخرین ACK می‌ماند. بقیه کار مانند تبادل چهارگانه است.

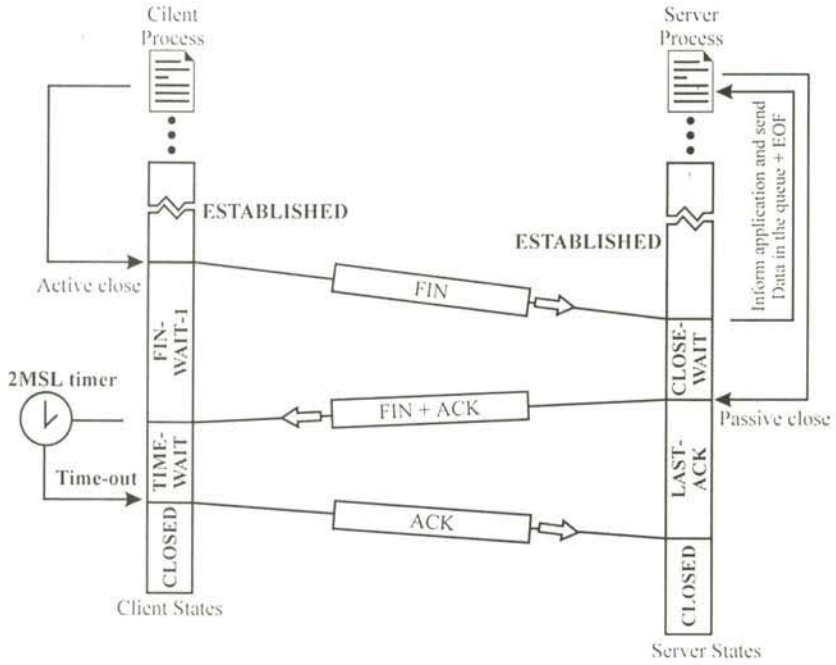
* باز شدن به طور همزمان

در ***باز شدن به طور همزمان***[1] برنامه‌های دو طرف به طور همزمان فرمان « به صورت اکتیو باز شو » را صادر می‌کنند. این یک وضعیت بسیار نامحتمل است و در آن مخابره، بین سرور و سرویس‌گیرنده نیست بلکه مخابره بین دو برنامه است که شماره پورتهای محلی همدیگر را می‌شناسند. TCP اجازه چنین حالتی را می‌دهد اما وقوع آن بسیار نامحتمل است زیرا که هر دو طرف باید برای هم، سگمنت SYN ارسال کنند و سگمنتهای ارسالی با هم به دست طرفین اتصال برسند. شکل ۱۲-۱۶ فاز برقراری اتصال را برای این سناریو نشان می‌دهد. هر دو TCP ها قبل از آنکه وارد حالت ESTABLISHED شوند، از حالتهای SYN-SENT و SYN-RCVD عبور می‌کنند. یک نگاه دقیق به موضوع نشان می‌دهد که هر دو طرف به طور همزمان، هم به صورت سرور و هم به صورت سرویس‌گیرنده عمل می‌کنند. دو سگمنت SYN+ACK ، تأییدیه‌ای بر سگمنتهای SYN دریافتی خواهند بود و اتصال را برقرار می‌کنند.

1- Simultaneous open

www.Abiri.ir

دقت نمائید که برقراری اتصال، شامل یک تبادل چهارگانه می‌شود. فازهای رد و بدل شدن دیتا و خاتمه اتصال، مشابه مثالهای قبل خواهند بود و در شکل ۱۲-۱۶ نشان داده نشده‌اند.



شکل ۱۲-۱۵: خاتمه اتصال با استفاده از تبادل سه‌گانه

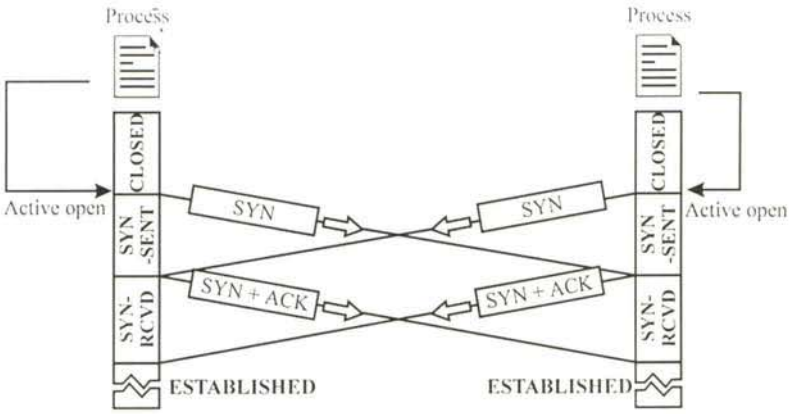
* بسته شدن به طور همزمان

یک سناریوی غیرمحتمل دیگر، وضعیتی است که ***بسته شدن به طور همزمان***[1] نامیده می‌شود و در شکل ۱۲-۱۷ نشان داده شده است. در این وضعیت، هر دو طرف فرمان « به صورت اکتیو بسته شو » را صادر می‌نمایند. هر دو TCP ، یک سگمنت FIN ارسال می‌کنند و سپس وارد حالت FIN-WAIT-1 می‌شوند. فرض می‌کنیم FINهای ارسالی به طور همزمان به دست طرف مقابل برسند. هر طرف پس از دریافت FIN ، یک سگمنت ACK ارسال می‌کند و وارد حالت CLOSING می‌شود. حالت CLOSING به جای حالت FIN-WAIT-2 یا CLOSE-WAIT در سناریوی متداول خاتمه اتصال، به کار رفته است. هر طرف پس از دریافت سگمنت ACK ، وارد حالت TIME-WAIT می‌شود. دقت نمائید که زمان انتظار در حالت TIME-WAIT برای هر دو طرف، الزامی است زیرا هر دو طرف، ACK فرستاده‌اند که ممکن

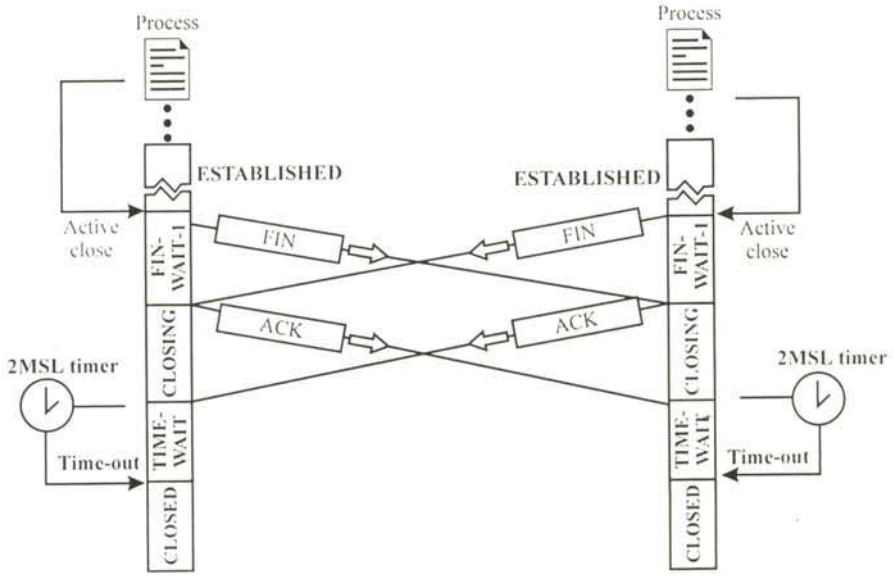
1- Simultaneous close

www.Abiri.ir

است گم شود. در شکل ۱۲-۱۷، فازهای برقراری اتصالی و رد و بدل شدن دیتا را حذف کرده‌ایم.



شکل ۱۲-۱۶: باز شدن به طور همزمان



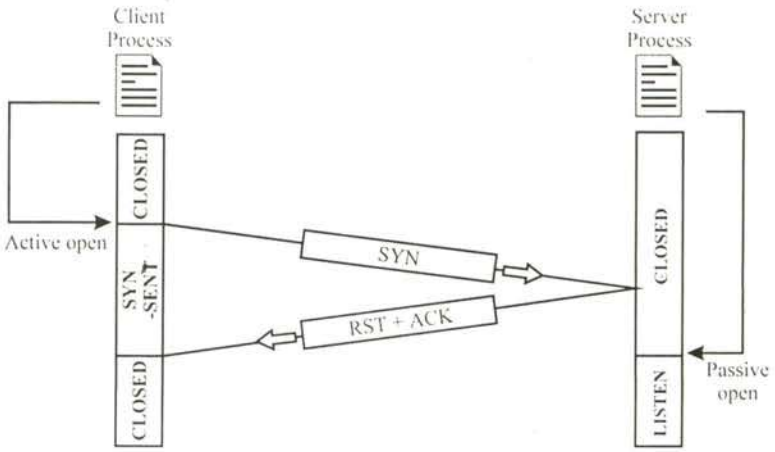
شکل ۱۲-۱۷: بسته شدن به طور همزمان

* رد یک اتصال

وضعیتی که غالباً ممکن است رخ دهد این است که شماره پورت مقصد در سگمنت SYN ای که سرویس‌گیرنده ارسال کرده است، به سروری اشاره کند که در آن لحظه، در

www.Abiri.ir

حالت LISTEN نیست و لذا TCP سمت سرور مجبور می‌شود سگمنت SYN دریافتی را دور بیندازد. TCP سمت سرور پس از دریافت سگمنت SYN ، یک سگمنت RST+ACK ارسال می‌کند و با این کار، هم دریافت سگمنت SYN را تأیید می‌نماید و هم اتصال را رد می‌کند. سرویس‌گیرنده پس از دریافت RST+ACK ، وارد حالت CLOSED می‌شود. به این سناریو، ***رد یک اتصال***[1] گفته می‌شود. بنابراین TCP سمت سرور فقط زمانی وارد حالت LISTEN می‌شود که قبلاً فرمان « به صورت پسیو باز شو » را از برنامه سرور دریافت کرده باشد. شکل ۱۲-۱۸ این وضعیت را نشان می‌دهد.



شکل ۱۲-۱۸: رد یک اتصال

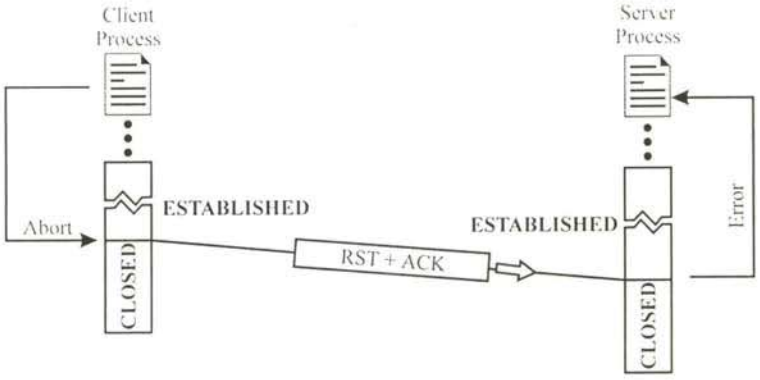
❊ لغو یک اتصال

یک برنامه می‌تواند اتصال موجودی را لغو کند که به آن، **لغو یک اتصال**[2] گفته می‌شود. اگر کار برنامه‌ای مختل شود (مثلاً به دلیل افتادن در لوپ بی‌انتها) و یا نخواهد دیتای موجود در صف، ارسال شود (به دلیل بروز خطایی در دیتا) چنین وضعیتی ممکن است رخ دهد. TCP نیز ممکن است بخواهد اتصالی را لغو کند. چنین وضعیتی ممکن است زمانی رخ دهد که او سگمنتی متعلق به یک اتصال قدیمی را دریافت کرده باشد. در تمام این موارد، TCP می‌تواند برای لغو اتصال، یک سگمنت RST+ACK ارسال نماید. شکل ۱۲-۱۹ وضعیتی

1- Denying a connection
2- Aborting a connection

www.Abiri.ir

را نشان می‌دهد که در آن برنامه سرویس‌گیرنده، می‌خواهد یک فرمان لغو را ارسال نماید. برای این کار TCP سرویس‌گیرنده، یک سگمنت RST+ACK را ارسال می‌نماید و همه دیتاهای موجود در صف را دور می‌ریزد. TCP سمت سرور نیز همه دیتاهای موجود در صف را دور می‌ریزد و از طریق یک پیغام خطا، مشکل را به اطلاع سرور می‌رساند. هر دو TCP فوراً وارد حالت CLOSED می‌شوند. دقت نمایید که در پاسخ به سگمنت RST+ACK، هیچ سگمنت ACKای تولید نمی‌شود.



شکل ۱۲-۱۹ : لغو یک اتصال

۱۲-۶ کنترل فلو

کنترل فلو[1] ، کنترلی بر حجم دیتایی است که مبدأ، قبل از دریافت تأییدیه از سمت مقابل، مجاز به ارسال آن است. در پایین‌ترین حد آن، پروتوکل لایه انتقال می‌تواند یک بایت دیتا بفرستد و قبل از ارسال بایت بعدی، منتظر دریافت تأییدیه بنشیند. این وضعیت، یک نقل و انتقال بسیار کند را نشان می‌دهد. اگر قرار باشد دیتا، مسافت زیادی را طی کند، در طول زمانی که مبدأ منتظر دریافت تأییدیه است، باید غیر فعال باقی بماند.

بالاترین حد آن این است که پروتوکل لایه انتقال، بدون آنکه منتظر دریافت تأییدیه از سمت مقابل بنشیند، همه دیتا را ارسال نماید. این عمل، سرعت را بالا می‌برد، اما ممکن است گیرنده از دیتا، لبریز شود. بعلاوه اگر بخشی از دیتا، گم شود، تکرار شود، خارج از نوبت دریافت شود یا خراب شود، تا زمانی که کل دیتا توسط گیرنده، چک نشود، مبدأ از خرابی رخ داده، مطلع نخواهد شد.

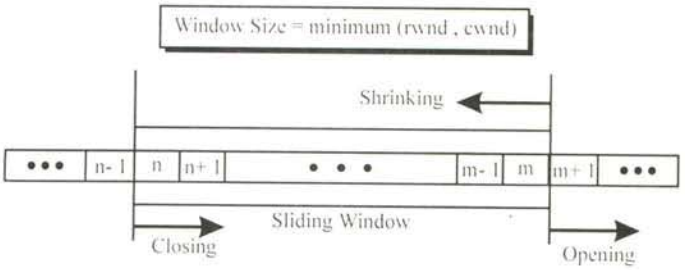
1- Flow control

www.Abiri.ir

TCP راه حلی مابین این دو حالت حدی، ارائه کرده است. او پنجره‌ای را تعریف می‌کند که به بافر اعمال می‌شود. این بافر حاوی دیتایی است که برنامه، متقاضی ارسال آن است. TCP فقط آن مقدار از دیتا را ارسال می‌کند که ***الگوریتم پنجره لغزان***[1] اجازه ارسال آن را داده است.

الگوریتم پنجره لغزان

برای انجام کنترل فلو، TCP از الگوریتم پنجره لغزان استفاده می‌کند. در این الگوریتم، کامپیوتر از یک پنجره برای ارسال دیتا استفاده می‌کند. پنجره، فقط بخشی از بافر حاوی دیتا را در برمی‌گیرد. بایتهای داخل پنجره را می‌توان بدون آنکه منتظر دریافت ACK از سمت مقابل بنشینیم، ارسال نماییم. این پنجره فرضی، دو دیوار دارد: یکی دیوار سمت چپ و دیگری دیوار سمت راست. به این پنجره، یک ***پنجره لغزان***[2] گفته می‌شود زیرا که مطابق شکل ۱۲-۲۰ ، دیواره‌های سمت چپ و راست آن می‌توانند بلغزند. شماره‌ها، شماره بایتهای داخل و خارج از پنجره را نشان می‌دهند.



شکل ۱۲-۲۰: پنجره لغزان

پنجره می‌تواند ***باز شود***[3] ، ***بسته شود***[4] و یا ***کوچک شود***[5] . همانطور که خواهیم دید، این سه فعالیت، تحت کنترل گیرنده و نَه فرستنده هستند (و همچنین به ازدحام موجود در شبکه بستگی دارند). در کنترل فلو، فرستنده باید تابع دستورات گیرنده باشد.

باز کردن پنجره به معنی حرکت دادن لبه سمت راست پنجره به سمت راست است. با این کار، فرستنده اجازه ارسال بایتهای بیشتری را می‌گیرد. بستن پنجره به معنی حرکت دادن

1- Sliding window algorithm
2- Sliding window
3- opened
4- Closed
5- Shrunk

www.Abiri.ir

لبه سمت چپ پنجره به سمت راست است و معنی آن این است که ACK برای برخی از بایتهای ارسالی، دریافت شده است و لازم نیست که فرستنده دیگر نگران آنها باشد. کوچک کردن پنجره به معنی حرکت دادن لبه سمت راست پنجره به سمت چپ است. شدیداً توصیه شده است که از این کار، اجتناب گردد. لذا برخی از پیاده‌سازیهای TCP ، اجازه چنین کاری را نمی‌دهند زیرا معنی آن این است که اجازه ارسال برخی از بایتها را لغو نماییم. اگر فرستنده هم‌اکنون برخی از این بایتها را ارسال کرده باشد، لغو اجازه ارسال آنها، مشکل‌ساز خواهد شد. دقت نمایید که لبه سمت چپ پنجره نمی‌تواند به سمت چپ حرکت کند زیرا که این عمل، لغو برخی از ACKهایی است که قبلاً دریافت شده‌اند.

مهم: برای آنکه هم کارآیی انتقال دیتا، بالاتر رود و هم فلوی دیتا، کنترل شود تا گیرنده از دیتا لبریز نشود، از پنجره لغزان استفاده می‌شود. پنجره لغزان TCP ، بایتگراست.

سایز پنجره در هر طرف را، مینیمم سایز این دو پنجره، تعیین می‌کند: ***پنجره گیرنده***[1] و ***پنجره ازدحام***[2] . rwnd مقداری است که گیرنده در فیلد window size سگمنت ACKای که ارسال می‌کند، می‌نویسد و برابر با تعداد بایتی است که گیرنده می‌تواند بدون لبریز شدن بافرش از دیتا، بپذیرد. پنجره ازدحام، مقداری است که توسط شبکه برای ممانعت از بروز ازدحام در شبکه تعیین می‌شود.

*** مثال ۳**

سایز پنجره rwnd برای کامپیوتر A چقدر است مشروط بر آنکه گیرنده مورد نظر یعنی کامپیوتر B ، بافری به سایز 5000 بایت داشته باشد و نیز 1000 بایت دیتا را دریافت کرده باشد که هنوز پردازش نشده‌اند.

راه حل : مقدار rwnd برابر است با 5000-1000=4000 . قبل از آنکه بافر کامپیوتر B لبریز کند، می‌تواند تا 4000 بایت دیگر را دریافت نماید. کامپیوتر B در سگمنت ACKای که برای کامپیوتر A می‌فرستد، عدد 4000 را در فیلد window size می‌نویسد.

*** مثال ۴**

اگر مقدار rwnd برابر با 3000 بایت و مقدار cwnd برابر با 3500 بایت باشد، سایز پنجره کامپیوتر A چقدر است؟

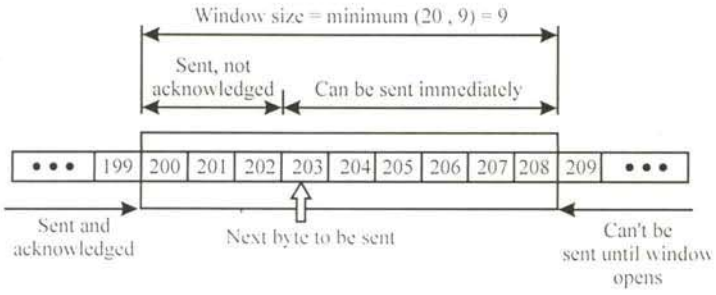
1- receiver window: rwnd
2- congestion window: cwnd

www.Abiri.ir

راه حل : سایز پنجره کامپیوتر A برابر با مینیمم دو مقدار داده شده برای rwnd و cwnd است و بنابراین برابر با 3000 بایت می‌باشد.

* مثال ۵

شکل ۱۲-۲۱ یک نمونه غیر واقعی از یک پنجره لغزان را نشان می‌دهد. فرستنده، بایتهای تا شماره 202 را ارسال کرده است. فرض می‌کنیم که cwnd برابر با 20 بایت باشد (مقداری که واقعاً برای cwnd به کار می‌رود، چند هزار بایت می‌باشد). گیرنده، شماره تأییدیه برابر با 200 را به همراه rwnd برابر با 9 بایت (مقداری که واقعاً برای rwnd به کار می‌رود، چند هزار بایت است) ارسال کرده است. سایز پنجره فرستنده، مینیمم دو مقدار rwnd و cwnd است و لذا برابر با 9 بایت می‌باشد. بایتهای 200 تا 202 ارسال شده‌اند اما تأییدیه‌ای برای آنها، دریافت نشده است. می‌توان بایتهای 203 تا 208 را ارسال کرد. بایتهای 209 به بعد را فعلاً نمی‌توان ارسال کرد.



شکل ۱۲-۲۱: مثال ۵

* مثال ۶

فرستنده در شکل ۱۲-۲۱ ، یک سگمنت ACK با شماره تأییدیه برابر با 202 و سایز پنجره rwnd برابر با 9 را دریافت می‌کند. فرستنده، قبل از دریافت این ACK، بایتهای 203 ، 204 و 205 را نیز ارسال کرده است. مقدار cwnd هنوز برابر با 20 است. پنجره جدید را نشان دهید.

راه حل : شکل ۱۲-۲۲ پنجره جدید را نشان می‌دهد. دقت نمائید که این موردی است که در آن، میزان بسته شدن پنجره از سمت چپ و باز شدن آن از سمت راست، به تعداد مساوی از بایتهاست. بنابراین سایز پنجره تغییر نکرده است. شماره تأییدیه 202 نشان می‌دهد

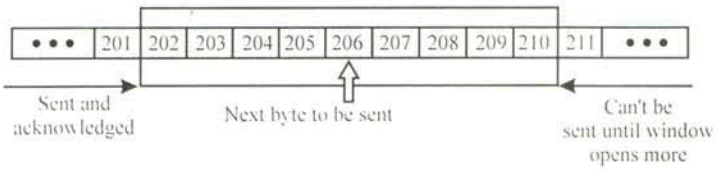
www.Abiri.ir

که بایتهای 200 و 201 دریافت شده‌اند و لزومی ندارد که فرستنده نگرانی دیگری درباره آنها داشته باشد. پنجره می‌تواند بلغزد و آنها را پشت سر بگذارد.

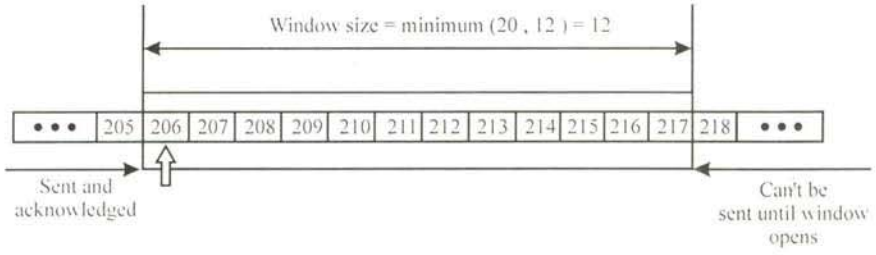
*** مثال ۷**

فرستنده در شکل ۱۲-۲۲ ، یک ACK جدید با شماره تأییدیه برابر با 206 و سایز پنجره rwnd برابر با 12 را دریافت می‌کند. فرستنده قبل از دریافت این ACK، هیچ بایت دیگری را ارسال نکرده است. مقدار cwnd هنوز برابر با 20 است. پنجره جدید را نشان دهید.

راه حل : مقدار rwnd کمتر از cwnd است، بنابراین سایز پنجره برابر با 12 است. شکل ۱۲-۲۳ پنجره جدید را نشان می‌دهد. دقت نمائید که پنجره از سمت راست به اندازه 7 خانه باز شده است و از سمت چپ به اندازه 4 خانه بسته شده است. سایز پنجره، افزایش یافته است.



شکل ۱۲-۲۲: مثال ۶



شکل ۱۲-۲۳: مثال ۷

*** مثال ۸**

فرستنده در شکل ۱۲-۲۳، یک ACK دیگر با شماره تأییدیه 210 و سایز پنجره rwnd برابر با 5 را دریافت می‌کند. فرستنده قبل از دریافت این ACK، بایتهای 206 ، 207 ، 208 و 209 را ارسال کرده است. مقدار cwnd هنوز برابر با 20 است. پنجره جدید را نشان دهید.

www.Abiri.ir

راه حل : مقدار rwnd کمتر از cwnd است، بنابراین سایز پنجره برابر با 5 است. شکل ۱۲-۲۴ این وضعیت را نشان می‌دهد. دقت نمائید که اکثر پیاده‌سازیها اجازه چنین وضعیتی را نمی‌دهند زیرا که پنجره در این حالت، نسبت به حالت قبل، کوچکتر شده است و لذا اگر قبلاً فرستنده، مجاز به ارسال بایتهای 215 ، 216 و 217 بود، حالا دیگر اجازه ارسال آنها را ندارد.

*** مثال ۹**

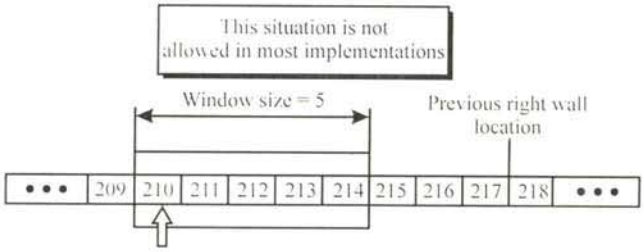
چطور گیرنده می‌تواند مانع از کوچک شدن پنجره در مثال ۸ گردد؟

راه حل : گیرنده همواره باید آخرین مقدار شماره تأییدیه (last ACK) و آخرین مقدار سایز rwnd (last rwnd) را مد نظر داشته باشد زیرا با جمع زدن این دو مقدار، شماره بایت بعد از لبه سمت راست پنجره بدست می‌آید. اگر می‌خواهیم مانع از حرکت لبه سمت راست پنجره به سمت چپ شویم (کوچک شدن پنجره)، باید همواره رابطه زیر را داشته باشیم:

$$\text{new ACK} + \text{new rwnd} \geq \text{last ACK} + \text{last rwnd}$$

یا

$$\text{new rwnd} \geq (\text{last ACK} + \text{last rwnd}) - \text{new ACK}$$



شکل ۱۲-۲۴: مثال ۸

در مثال ۸، شماره تأییدیه جدید (new ACK) برابر با 210 و سایز جدید rwnd (new rwnd) برابر با 5 است. شماره آخرین تأییدیه (last ACK) برابر با 206 و آخرین سایز rwnd (last rwnd) برابر با 12 بود. رابطه فوق به صورت 210 - (12+206) ≥ 5 درمی‌آید که صحیح نیست. بنابراین قبل از آنکه گیرنده بتواند ACK بفرستد، باید صبر کند تا فضای بیشتری در بافرش خالی گردد.

مهم : برای ممانعت از کوچک شدن پنجره فرستنده، گیرنده باید صبر کند تا فضای بیشتری در بافرش، خالی شود.

www.Abiri.ir

متوقف کردن پنجره

گفتیم که کاملاً مخالف کوچک کردن پنجره هستیم. اما یک استثناء وجود دارد: گیرنده می‌تواند با ارسال یک rwnd برابر با 0 ، موقتاً ***پنجره را متوقف کند***[1]. این وضعیت ممکن است زمانی رخ دهد که گیرنده، به دلیلی نمی‌خواهد تا مدتی، دیتای بیشتری از فرستنده دریافت کند. سایز پنجره در این وضعیت، کوچک نمی‌شود بلکه فرستنده تا دریافت خبر جدید از گیرنده، ارسال دیتا را متوقف می‌کند. زمانی که پنجره با آمدن فرمانی از سوی گیرنده، متوقف می‌شود، فرستنده، سگمنتی حاوی فقط یک بایت دیتا را مکرراً ارسال می‌کند. به این سگمنت، ***سگمنت کاوش***[2] می‌گویند و برای ممانعت از ***توقف گفتگوی فرستنده و گیرنده***[3] به کار می‌رود (بخش مربوط به تایمرهای TCP را ببینید).

مهم: برخی نکات درباره پنجره‌های لغزان TCP

الف. سایز پنجره برابر با مینیمم سایز rwnd و cwnd است.

ب. مبدأ مجبور نیست دیتایی به اندازه یک پنجره کامل ارسال نماید.

ج. گیرنده می‌تواند پنجره را باز یا بسته کند، اما نمی‌تواند آن را کوچک کند.

د. مقصد هر زمانی که بخواهد می‌تواند ACK بفرستد، اما مشروط بر آنکه منجر به کوچک شدن پنجره نشود.

ه. گیرنده می‌تواند موقتاً پنجره را متوقف کند. با این حال، بعد از متوقف شدن پنجره، فرستنده همواره می‌تواند سگمنت یک بایتی ارسال نماید.

علامت پنجره نامعقول

یک مشکل جدی زمانی در عملکرد پنجره لغزان رخ می‌دهد که یا برنامه فرستنده، به آهستگی دیتا تولید کند و یا برنامه گیرنده، به آهستگی دیتا را مصرف نماید و یا اینکه هر دو مورد فوق رخ دهد. هر یک از این حالتها سبب می‌گردند تا ارسال دیتا توسط سگمنتهای بسیار کوچک صورت گیرد که کارآیی را پایین می‌آورد. به عنوان مثال، اگر TCP سگمنتهای حاوی 1 بایت دیتا ارسال نماید، به این معنی است که یک بسته 41 بایتی (20 بایت هدر TCP و 20 بایت هدر IP و 1 بایت دیتا) فقط 1 بایت دیتا را جابجا می‌کند. در این حالت میزان اطلاعات بالاسری به نسبت 41/1 است و نشان می‌دهد که از ظرفیت شبکه، با راندمان بسیار کمی استفاده می‌شود. وقتی که چنین مشکلی رخ می‌دهد می‌گویند TCP یک طرف، ***علامت پنجره***

1- window shutdown
2- probe
3- deadlock

www.Abiri.ir

نامعقول[1] را از خود نشان می‌دهد. ابتدا در هر طرف نشان می‌دهیم که چرا این مشکل به وجود می‌آید و سپس راه حلی را پیشنهاد می‌کنیم .

*** علامت توسط فرستنده تولید شده است**

اگر TCP سمت فرستنده، علامت پنجره نامعقول را از خود نشان دهد، در حال ارسال دیتای برنامه‌ای است که با سرعت بسیار کمی، مثلاً هر بار فقط یک بایت، دیتا به TCP می‌دهد. به عبارت دیگر، برنامه هر بار فقط یک بایت دیتا، داخل بافر TCP فرستنده می‌گذارد. اگر دستورات خاصی به TCP فرستنده داده نشود، ممکن است سگمنتهایی حاوی یک بایت دیتا تولید نماید. نتیجه، تعداد زیادی سگمنتهای 41 بایتی است که فقط حاوی یک بایت دیتا می‌باشند.

راه حل این است که نگذاریم TCP فرستنده، یک بایت یک بایت، دیتا را ارسال کند. باید TCP فرستنده را مجبور کنیم تا صبر کند و پس از جمع کردن حجم قابل توجهی از دیتا، آنها را ارسال نماید. چه مدت زمانی TCP فرستنده باید صبر کند؟ اگر مدت زمان زیادی صبر کند، وقفه‌ای در روند ارسال دیتا پیش می‌آید و اگر به اندازه کافی صبر نکند، سگمنتهای کوچکی ارسال خواهند شد. شخصی به نام ***ناگل***[2] راه حل جالبی پیدا کرد.

**** الگوریتم ناگل**

الگوریتم ناگل[3] بسیار ساده است:

۱- TCP فرستنده اولین قطعه دیتایی که از برنامه فرستنده دریافت می‌کند، حتی اگر فقط یک بایت باشد، را ارسال می‌کند.

۲- بعد از ارسال اولین سگمنت، TCP فرستنده دیتای دریافتی از برنامه فرستنده را در بافر خروجی خود ذخیره می‌کند و تا زمانی صبر می‌کند که یا TCP گیرنده، یک ACK ارسال نماید و یا دیتای داخل بافر خروجی بتواند ***سگمنتی به طول حداکثر***[4] را پر کند. در این لحظه TCP فرستنده می‌تواند سگمنت را ارسال کند.

۳- TCP فرستنده، مرحله قبل را برای بقیه دیتاهای دریافتی از برنامه فرستنده، تکرار خواهد کرد. بنابراین سگمنت سوم را زمانی ارسال می‌کند که یا یک ACK از سمت گیرنده

1- silly window syndrome
2- Nagle
3- Nagle's Algorithm
4- Maximum Segment Size

www.Abiri.ir

برای سگمنت دوم ارسالی، دریافت کرده باشد و یا دیتای کافی در بافر خروجی آن، جمع شده باشد به طوری که سگمنتی به طول حداکثر را پر کند.

زیبایی الگوریتم ناگل در سادگی آن و نیز این حقیقت است که این الگوریتم، هم سرعت تولید دیتا توسط برنامه فرستنده و هم سرعت انتقال شبکه را مد نظر دارد. اگر برنامه، سریعتر از شبکه باشد، سگمنتها، بلندتر (به طول حداکثر) خواهند بود و اگر برنامه، کندتر از شبکه باشد، سگمنتها، کوچکتر (کوچکتر از طول حداکثر) خواهند بود.

* علامت توسط گیرنده تولید شده است

ممکن است علامت پنجره نامعقول، از سوی TCP گیرنده مشاهده شود و دلیل آن این باشد که او دیتا را به برنامه‌ای تحویل می‌دهد که آن برنامه، دیتا را بسیار آهسته مصرف می‌کند. مثلاً فرض کنید که برنامه فرستنده، دیتا را به صورت بلاکهای یک کیلو بایتی تولید می‌کند، اما برنامه گیرنده، فقط یک بایت یک بایت، دیتا را مصرف می‌کند. همچنین فرض کنید که بافر ورودی TCP گیرنده، فقط گنجایش 4 کیلو بایت دیتا را داشته باشد، فرستنده اولین 4 کیلو بایت دیتا را ارسال می‌کند و گیرنده آن را دریافت می‌کند و در بافر خود می‌گذارد. حال بافر گیرنده از دیتا پر شده است. بنابراین گیرنده با اعلام سایز 0 برای پنجره، از فرستنده می‌خواهد که دیتای بیشتری ارسال نکند. برنامه گیرنده، یک بایت از بافر TCP گیرنده را می‌خواند. بنابراین اکنون بافر TCP گیرنده به اندازه یک بایت، فضای خالی دارد. TCP گیرنده با اعلام سایز پنجره برابر با یک، به فرستنده اعلام می‌کند که به اندازه یک بایت دیتای بیشتر می‌تواند ارسال کند. TCP فرستنده که مشتاقانه منتظر دریافت خبری از سوی TCP گیرنده است تا بتواند بایتهای بیشتری را بفرستد، همین خبر را یک خبر خوب تلقی می‌کند و لذا یک سگمنت حاوی یک بایت دیتا برای گیرنده می‌فرستد و این روند ادامه می‌یابد. یعنی برنامه گیرنده، یک بایت از دیتا را می‌خواند و TCP فرستنده یک سگمنت حاوی یک بایت دیتا ارسال می‌نماید. در این حالت نیز کارآیی پایین می‌آید و علامت پنجره نامعقول نمایان می‌شود.

دو راه حل برای رفع این مشکل، پیشنهاد شده است:

** راه حل کلارک

اولین راه حل که ***راه حل کلارک***[1] نامیده می‌شود این است که TCP گیرنده به محض دریافت دیتا، ACK را ارسال کند، اما زمانی که بافرش پر شد و لذا TCP گیرنده، سایز پنجره را برابر با 0 اعلام کرد، سایز پنجره را همچنان 0 اعلام کند تا زمانی که یا در بافرش، جای کافی

1- Clark's solution

www.Abiri.ir

برای ذخیره یک سگمنت دیتا به طول حداکثر را داشته باشد و یا نیمی از بافرش، خالی شده باشد. در یکی از این دو حالت، سایز پنجره را مخالف با 0 اعلام کند.

❊ ارسال ACK با تأخیر

دومین راه حل این است که گیرنده، ACK را با تأخیر ارسال نماید. به این معنی که TCP گیرنده پس از دریافت یک سگمنت، بلافاصله ACK را ارسال نکند. TCP گیرنده تا زمانی که فضای خالی قابل توجهی در بافر ورودی‌اش نداشته باشد، ارسال ACK را به تعویق می‌اندازد. به تعویق انداختن ارسال ACK مانع از این می‌شود که TCP فرستنده، پنجره‌اش را به جلو حرکت دهد و لذا بعد از آنکه TCP فرستنده دیتای داخل پنجره‌اش را ارسال کرد، متوقف می‌شود و سگمنتهای کوچک تولید نخواهند شد.

این روش، فایده دیگری نیز دارد : ترافیک را کاهش می‌دهد زیرا که گیرنده مجبور نیست برای هرسگمنتی، ACK بفرستد. با این وجود، عیبی نیز وجود دارد و آن اینکه به تأخیر انداختن ارسال ACK ، ممکن است سبب گردد فرستنده، دیتاهایی که ACK نشده‌اند را مجدداً ارسال نماید.

این راه حل سعی می‌کند توازنی بین حسن و عیب یاد شده، ایجاد نماید، به این ترتیب که نمی‌گذارد مقدار به تأخیر افتادن ارسال ACK از 500 ms بیشتر شود.

۱۲-۷ کنترل خطا

TCP یک پروتوکل قابل اطمینان لایه انتقال است. معنای آن این است که برنامه فرستنده که دنباله‌ای از بایتها را به TCP فرستنده می‌دهد، تا آنها را برای برنامه سمت گیرنده ارسال کند، فرض را بر این می‌گذارد که TCP می‌تواند بدون خطا و بدون آنکه بخشی از دیتا گم شود یا تکرار شود، آن را صحیح و سالم و با همان نظم و ترتیب اولیه، به دست برنامه سمت مقابل برساند.

TCP قابلیت اطمینان را با استفاده از ***کنترل خطا***[1] ، به وجود می‌آورد. کنترل خطا شامل مکانیزمهایی برای آشکارسازی سگمنتهای مخدوش شده، سگمنتهای گم‌شده، سگمنتهای خارج از نوبت دریافت شده و سگمنتهای تکراری است. کنترل خطا همچنین حاوی مکانیزمی برای تصحیح خطا، پس از آشکارسازی آن می‌باشد. TCP با استفاده از سه ابزار ساده، عمل آشکارسازی و تصحیح خطا را انجام می‌دهد : checksum ، تأییدیه و time-out .

1- error control

www.Abiri.ir

checksum

هر سگمنت حاوی یک فیلد checksum است که مخدوش شدن سگمنت را آشکار می‌کند. اگر سگمنتی مخدوش شده باشد، توسط TCP مقصد، دور انداخته می‌شود و یک سگمنت گم شده تلقی می‌گردد. TCP از یک checksum به طول 16 بیت استفاده می‌کند که وجود آن در هر سگمنت، اجباری است. در فصل بعد خواهیم دید که برای پروتوکل جدید لایه انتقال به نام SCTP ، یک checksum به طول 16 بیت، کافی نیست. اما نمی‌توان این طول را برای TCP تغییر داد، زیرا که نیاز به تغییر مجدد پیکربندی هدر TCP دارد.

تأییدیه

TCP برای تأیید سگمنتهای دریافتی از ***تأییدیه***[1] استفاده می‌کند. تأییدیه برای سگمنتهای کنترلی که حاوی دیتا نیستند، اما یک شماره ترتیب را مصرف می‌کنند، نیز ارسال می‌شود. تأییدیه هرگز برای خود سگمنتهای ACK ، ارسال نمی‌شود.

مهم: سگمنتهای ACK هیچ شماره ترتیبی را مصرف نمی‌کنند و تأییدیه‌ای نیز برای آنها، ارسال نمی‌شود.

❊ تولید تأییدیه‌ها

چه زمانی گیرنده، تأییدیه ارسال می‌کند؟ در خلال زمانی که TCP شکل می‌گرفت، قواعد متعددی توسط پیاده‌سازیهای مختلف TCP ، تعریف شدند و به کار رفتند. در اینجا متداولترین قواعد را ارائه می‌دهیم. ترتیب بیان قواعد، نشان‌دهنده میزان اهمیت آنها نیست.

۱- وقتی که یک طرف، یک سگمنت دیتا برای طرف دیگر ارسال می‌کند، باید به همراه آن، شماره تأییدیه بایتهای دریافتی را نیز ارسال نماید (piggybacking) . این قاعده، تعداد سگمنتهای لازم را کاهش می‌دهد و بنابراین باعث کم شدن ترافیک می‌شود.

۲- وقتی که گیرنده، دیتای دیگری برای ارسال ندارد و یک سگمنت دیتا با شماره ترتیب مورد انتظار را از سمت مقابل دریافت می‌کند، اگر تأییدیه دیتای دریافتی قبلی را ارسال کرده باشد، باید ارسال تأییدیه برای این سگمنت را تا زمانی به تأخیر بیندازد که یا یک سگمنت دیتای دیگر از سمت مقابل دریافت کند و یا یک زمان مشخصی (معمولاً 500 میلی‌ثانیه) سپری گردد. به عبارت دیگر اگر فقط یک سگمنت است که باید برای آن تأییدیه

1- Acknowledgment : ACK

www.Abiri.ir

ارسال شود، گیرنده باید ارسال تأییدیه برای آن یک سگمنت واحد را به تأخیر بیندازد. این قاعده نیز مانع از آن می‌شود که سگمنتهای ACK ، سبب ایجاد ترافیک اضافی شوند.

۳- وقتی که سگمنتی از راه می‌رسد که شماره ترتیب آن عددی است که گیرنده انتظار دریافت آن را دارد، اگر گیرنده یک سگمنت دیگر نیز قبلاً دریافت کرده باشد و برای آن تأییدیه ارسال نکرده باشد، گیرنده بلافاصله سگمنت ACK را ارسال می‌کند. به عبارت دیگر، نباید در هر لحظه بیش از دو سگمنت در دست گیرنده باشد که هنوز برای آنها، تأییدیه صادر نکرده باشد. این کار مانع از آن می‌شود که فرستنده، به دلیل عدم دریافت ACK، مجدداً سگمنتهایی را ارسال کند که گیرنده، قبلاً آنها را دریافت کرده است.

۴- زمانی که یک سگمنت از راه می‌رسد که شماره ترتیب آن بزرگتر از عدد مورد انتظار است، گیرنده بلافاصله یک سگمنت ACK ارسال می‌کند و طی آن، شماره ترتیبی را اعلام می‌نماید که انتظار دریافت آن را دارد. این کار باعث ارسال مجدد سگمنتهای گم‌شده می‌گردد.

۵- زمانی که سگمنت گم شده از راه می‌رسد، گیرنده یک سگمنت ACK ارسال می‌کند و طی آن، شماره ترتیب بعدی که انتظار دریافت آن را دارد، اعلام می‌نماید. این کار به اطلاع فرستندهٔ سگمنتها می‌رساند که سگمنتی که به عنوان گم شده، گزارش شده بود، دریافت شده است.

۶- اگر یک سگمنت تکراری از راه برسد، گیرنده بلافاصله ACK ارسال می‌نماید. این کار بعضی از مشکلاتی که در اثر گم شدن خود سگمنت ACK پیش می‌آید را برطرف می‌کند.

* نوع تأییدیه

TCP در گذشته، فقط از یک نوع تأییدیه استفاده می‌کرد: ***تأییدیه تجمعی***[1] . امروزه برخی از پیاده‌سازیهای TCP از ***تأییدیه انتخابی***[2] نیز استفاده می‌کنند.

** تأییدیه تجمعی: ACK

TCP از ابتدا این طور طراحی شده بود که دریافت سگمنتها را به صورت تجمعی تأیید نماید. به این ترتیب که گیرنده، بایت بعدی که انتظار دریافت آن را دارد، اعلام می‌نماید و از تمام سگمنتهایی که خارج از نوبت دریافت شده‌اند، صرفنظر می‌کند. گاهی اوقات به آن، ***تأییدیه تجمعی مثبت***[3] و یا همان ACK گفته می‌شود. کلمه « مثبت» به این معناست که

1- accumulative acknowledgment
2- selective acknowledgment
3- Positive accumulative acknowledgment

www.Abiri.ir

تأییدیه، برای سگمنتهای گم شده، دور انداخته شده و یا تکراری، ارسال نمی‌شود. فیلد 32 بیتی ACK در هدر TCP ، برای تأییدیه‌های تجمعی به کار می‌رود و مقدار آن فقط زمانی که بیت ACK ، برابر با 1 باشد، معتبر است.

**** تأییدیه انتخابی : SACK**

نوع دیگری از تأییدیه به نام ***تأییدیه انتخابی***[1] در پیاده‌سازیهای جدید نمایان شده است. SACK جایگزین ACK نشده است، بلکه فقط اطلاعات بیشتری را به فرستنده می‌دهد. SACK بلاکی از دیتا که خارج از نوبت دریافت شده است و همچنین بلاکی از دیتا که تکرار شده است را به فرستنده گزارش می‌نماید. با این وجود، از آنجا که هدر TCP ، جایی برای اضافه کردن این نوع اطلاعات ندارد، SACK به صورت یک ***گزینه***[2]، در انتهای هدر TCP پیاده‌سازی شده است. وقتی که درباره گزینه‌های انتخابی TCP صحبت می‌کنیم، درباره این نوع گزینه انتخابی نیز بحث خواهیم کرد.

*** ارسال مجدد**

قلب مکانیزم کنترل خطا، ارسال مجدد سگمنتها است. وقتی که سگمنتی مخدوش می‌شود، گم می‌شود و یا به تأخیر می‌افتد، باید دوباره ارسال شود. در پیاده‌سازیهای جدید، در دو حالت، سگمنتی مجدداً ارسال می‌شود. الف - وقتی که تایمر ارسال مجدد به انتهای مهلت مقرر می‌رسد. ب- زمانی که فرستنده، سه تا ACK تکراری دریافت می‌کند.

مهم: در پیاده‌سازیهای جدید TCP ، زمانی ارسال مجدد رخ می‌دهد که تایمر ارسال مجدد، به انتهای مهلت مقرر برسد و یا سه سگمنت ACK تکراری، از راه رسیده باشند.

دقت کنید برای سگمنتهایی که شماره ترتیب مصرف نمی‌کنند، ارسال مجددی رخ نخواهد داد. به ویژه، یک سگمنت ACK ، مجدداً ارسال نخواهد شد.

مهم: برای یک سگمنت ACK ، تایمر ارسال مجدد، ست نمی‌شود.

1- Selective acknowledgment : SACK
2- option

www.Abiri.ir

❊ ارسال مجدد بعد از RTO

TCP فرستنده برای هر سگمنتی که می‌فرستد، تایمر ارسال مجدد را روی ***مهلت مقرر ارسال مجدد***[1]، ست می‌کند. وقتی که تایمر به انتهای زمان ست شده خود می‌رسد، سگمنت متناظر با آن، گم شده و یا مخدوش شده، در نظر گرفته می‌شود و لذا آن سگمنت، باید مجدداً ارسال شود. ممکن است دلیل عدم دریافت ACK ، ناشی از سگمنت به تأخیر افتاده، ACK به تأخیر افتاده و یا یک ACK گم شده باشد. دقت نمایید که هیچ تایمری برای سگمنتی که فقط حامل ACK است، ست نمی‌شود زیرا که چنین سگمنتی هرگز دوباره ارسال نخواهد شد. خواهیم دید که مقدار RTO در TCP ، مقدار دینامیکی است و بر اساس ***زمان رفت و برگشت***[2] سگمنتها، به روزرسانی می‌شود. RTT جمع زمانی است که طول می‌کشد تا سگمنتی به مقصد برسد و سپس فرستنده سگمنت، ACK مربوط به آن را دریافت کند.

❊ ارسال مجدد، بعد از دریافت سه ACK تکراری

اگر مقدار RTO خیلی بزرگ نباشد، قاعده قبلی درباره ارسال مجدد یک سگمنت، کافی است. با این حال گاهی اوقات یک سگمنت گم می‌شود و گیرنده آنقدر سگمنتهای خارج از نوبت دریافت می‌کند که ذخیره کردن همه آنها (به دلیل سایز محدود بافر گیرنده) امکانپذیر نیست. برای جبران این مشکل، اکثر پیاده‌سازیهای جدید از قاعده سه ACK تکراری استفاده می‌کنند. به این ترتیب که پس از دریافت سه ACK تکراری، سگمنت گم شده را، مجدداً ارسال می‌کنند. این ویژگی، ***ارسال مجدد سریع***[3] نامیده شده است.

❊ سگمنتهای خارج از نوبت

وقتی که سگمنتی به تأخیر می‌افتد، گم می‌شود و یا تکرار می‌شود، سگمنتهایی که به دنبال آن سگمنت، به دست گیرنده می‌رسند، همگی خارج از نوبت خواهند بود. در ابتدا TCP چنان طراحی شده بود که تمام سگمنتهای خارج از نوبت را دور بیندازد و نتیجه آن، ارسال مجدد سگمنت گم شده و تمام سگمنتهای بعد از آن بود. اکثر پیاده‌سازیهای جدید TCP ، سگمنتهای خارج از نوبت را دور نمی‌اندازند بلکه آنها را موقتاً ذخیره می‌کنند و تا زمانی که سگمنتهای گم شده از راه برسند، آنها را تحت عنوان سگمنتهای خارج از نوبت، نگه می‌دارند. با این حال دقت نمائید که سگمنتهای خارج از نوبت، به برنامه گیرنده تحویل داده نمی‌شوند، زیرا

1- Retransmission Time-Out : RTO
2- Round Trip Time: RTT
3- Fast retransmission

www.Abiri.ir

که TCP تضمین کرده است که دیتا را با همان نظم و ترتیب اولیه، به برنامه گیرنده تحویل دهد.

مهم: دیتا ممکن است خارج از نوبت از راه برسد و موقتاً توسط TCP گیرنده، ذخیره شود. اما TCP تضمین کرده است که هیچ سگمنتی خارج از نوبت، به برنامه گیرنده، تحویل داده نشود.

چند سناریو

در این بخش، چند سناریو از طرز کار TCP ارائه می‌نمائیم. در این سناریوها، یک سگمنت را با یک مستطیل نشان می‌دهیم. اگر سگمنت حامل دیتا باشد، محدوده شماره بایتها و نیز محتوای ***فیلد تأییدیه***[1] را نشان می‌دهیم. اگر سگمنت فقط حامل ACK باشد، فقط محتوای فیلد تأییدیه را در یک مستطیل کوچکتر، نشان می‌دهیم.

* کار طبیعی TCP

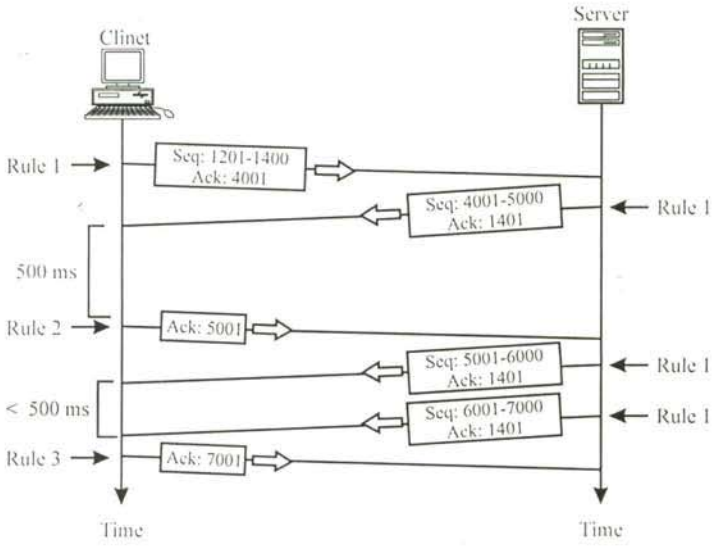
سناریوی ***کار طبیعی TCP***[2] مطابق شکل ۱۲-۲۵ ، جابجا شدن دو طرفه دیتا را بین دو سیستم، نشان می‌دهد. TCP سمت سرویس‌گیرنده، یک سگمنت را ارسال می‌کند. TCP سمت سرور، سه سگمت را ارسال می‌نماید. شکل ۱۲-۲۵ نشان می‌دهد که کدام قاعده به هر تأییدیه اعمال می‌شود. برای اولین سگمنتِ سرویس‌گیرنده و هر سه سگمنتِ سرور، قاعده اول اعمال شده است. یعنی دیتایی برای ارسال وجود دارد و بنابراین سگمنت، بایت مورد انتظار بعدی را نشان می‌دهد. وقتی که سرویس‌گیرنده، اولین سگمنت را از سرور دریافت می‌کند، دیتای بیشتری برای ارسال ندارد. او فقط یک سگمنت ACK می‌فرستد. با این حال بنا به قاعده شماره ۲، ارسال ACK باید به اندازه 500 میلی‌ثانیه به تأخیر بیفتد تا سرویس‌گیرنده ببیند آیا سگمنت دیگری از راه می‌رسد یا نَه. وقتی که تایمر به انتهای زمان ست شده خود رسید، سرویس‌گیرنده یک ACK می‌فرستد. دلیل آن این است که سرویس‌گیرنده نمی‌داند آیا سگمنت دیگری از راه می‌رسد یا نَه و بنابراین نمی‌تواند ارسال ACK را برای همیشه به تأخیر بیندازد. وقتی که سگمنت بعدی از راه می‌رسد، تایمر دیگری برای ارسال ACK ست می‌شود. اما قبل از آنکه این تایمر به انتهای زمان ست شده خود برسد، سگمنت سوم از راه می‌رسد. رسیدن سگمنت سوم، بنابه قاعده شماره ۳، ارسال ACK دیگری را از سوی سرویس‌گیرنده، سبب می‌گردد.

1- acknowledgment field
2- normal operation of TCP

www.Abiri.ir

* سگمنت گم شده

در سناریوی **سگمنت گم شده**[1] ، نشان می‌دهیم که اگر سگمنتی گم شود و یا مخدوش شود، چه اتفاقی می‌افتد. گیرنده در قبال یک سگمنت گم شده و یا یک سگمنت مخدوش شده، رفتار مشابهی دارد. یک سگمنت گم شده، جایی در شبکه دور انداخته شده است. یک سگمنت مخدوش شده، توسط خود گیرنده دور انداخته می‌شود. هر دو حالت را، گم شدن سگمنت در نظر می‌گیریم. شکل ۱۲-۲۶ وضعیتی را نشان می‌دهد که در آن، سگمنتی گم شده است. ممکن است دلیل آن این باشد که روتری در شبکه، آن سگمنت را دور انداخته است و ممکن است دلیل آن، وقوع ازدحام باشد.



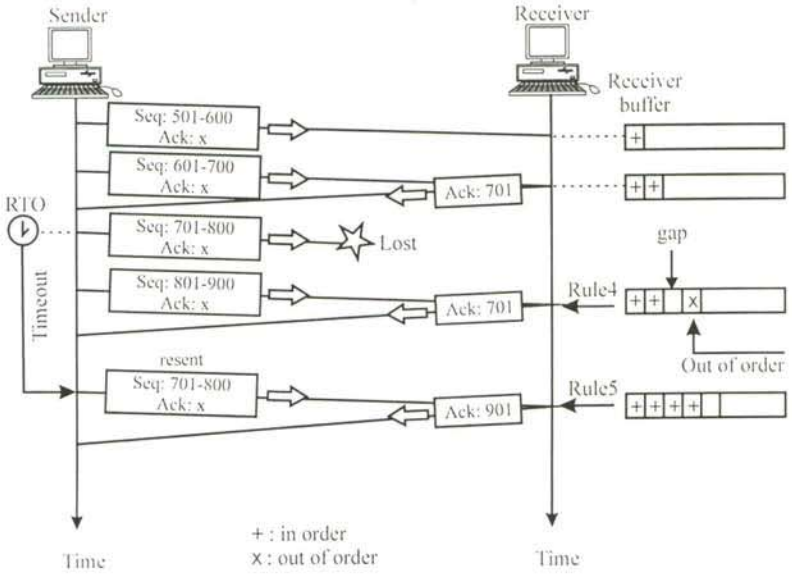
شکل ۱۲-۲۵: کار طبیعی TCP

فرض بر این است که انتقال دیتا، یک طرفه است: یک طرف، دیتا می‌فرستد و طرف دیگر، دیتا می‌گیرد. فرستنده در این سناریو، سگمنتهای اول و دوم را می‌فرستد که طبق قاعده ۳ ، بلافاصله برای آنها ACK صادر می‌شود. اما سگمنت سوم فرستنده، گم می‌شود و لذا گیرنده، سگمنت چهارم فرستنده را خارج از نوبت دریافت می‌کند. گیرنده، دیتای سگمنت چهارم را در بافرش ذخیره می‌کند اما یک جای خالی می‌گذارد تا نشان دهد که دیتای دریافتی، پیوسته نیست. طبق قاعده ۴، گیرنده بلافاصله یک ACK برای فرستنده ارسال

1- lost segment

www.Abiri.ir

می‌کند تا بایت بعدی که انتظار دریافت آن را دارد، اعلام کند. دقت نمایید که گیرنده، بایتهای 801 تا 900 (سگمنت چهارم فرستنده) را ذخیره می‌کند اما تا زمانی که جای خالی، پر نشده باشد، آنها را به برنامه گیرنده تحویل نمی‌دهد.



شکل ۱۲-۲۶: سگمنت گم شده

مهم : TCP گیرنده فقط دیتا را به صورت مرتب شده تحویل برنامه گیرنده می‌دهد.

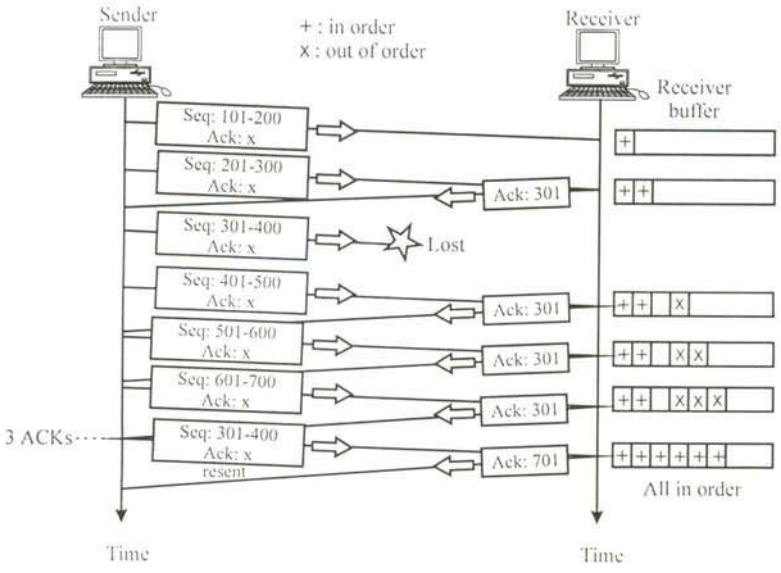
گرچه TCP فرستنده، یک تایمر RTO را برای هر سگمنتی که ارسال می‌کند، ست می‌کند، اما ما فقط این تایمر را برای سگمنتی که گم شده است، نشان داده‌ایم. تایمر مربوط به این سگمنت، قطعاً به انتهای زمان ست شده می‌رسد، زیرا گیرنده هرگز برای سگمنت گم شده یا سگمنتی که خارج از نوبت دریافت شده است، ACK نمی‌فرستد. وقتی که تایمر به انتهای زمان ست شده‌اش می‌رسد، TCP فرستنده، مجدداً سگمنت سوم را می‌فرستد که این بار، به دست گیرنده می‌رسد و به طور مناسب، ACK برای آن ارسال می‌شود (قاعده ۵). دقت نمایید که بر اساس قاعده یاد شده، مقدار فیلد تأییدیه برای ACK دوم و ACK سوم، متفاوت است.

www.Abiri.ir

* ارسال مجدد سریع

در این سناریو، می‌خواهیم ایده ***ارسال مجدد سریع***[1] را نشان دهیم. این سناریو مشابه با سناریوی دوم است با این تفاوت که مقدار RTO در این حالت، بیشتر است (شکل ۱۲-۲۷ را ببینید).

وقتی که گیرنده، سگمنتهای چهارم، پنجم و ششم را دریافت می‌کند، برای هرکدام، یک ACK می‌فرستد. فرستنده، چهار ACK را دریافت کرده است که مقدار فیلد تأییدیه همه آنها، مقدار مشابه 301 است (یکی ACK اولی و سه تا ACK تکراری پس از آن). گرچه تایمر برای سگمنت سوم، هنوز به انتهای زمان ست شده‌اش نرسیده است، قاعده « ارسال مجدد سریع » ایجاب می‌کند که سگمنت سوم، سگمنتی که همه این ACKها انتظار دریافت آن را دارند، فوراً مجدداً ارسال شود (در شکل ۱۲-۲۷ با علامت resent نشان داده شده است).



شکل ۱۲-۲۷: ارسال مجدد سریع

* سگمنت به تأخیر افتاده

سناریوی چهارم درباره یک ***سگمنت به تأخیر افتاده***[2] است. TCP از سرویسهای IP که یک پروتوکل فاقد اتصال است، استفاده می‌کند. هر بسته IP که یک سگمنت TCP را

1- fast retransmission
2- Delayed segment

www.Abiri.ir

کپسول‌بندی می‌کند، ممکن است از مسیر متفاوتی و با تأخیر متفاوتی خود را به مقصد برساند و بنابراین ممکن است سگمنتهای TCP ، به تأخیر بیفتند. گیرنده با سگمنتهای به تأخیر افتاده، مشابه با سگمنتهای گم شده یا مخدوش شده رفتار می‌کند و ممکن است حالتی پیش آید که سگمنتی به تأخیر بیفتد و TCP فرستنده، آن را مجدداً ارسال کند و در نتیجه گیرنده، هم سگمنت به تأخیر افتاده و هم سگمنت مجدداً ارسال شده را دریافت کند و لذا حالت سگمنت تکراری پیش آید.

* سگمنت تکراری

ممکن است مثلاً TCP فرستنده، وقتی که سگمنتی به تأخیر می‌افتد و گیرنده با آن به عنوان یک سگمنت گم شده رفتار می‌کند، یک ***سگمنت تکراری***[1] ایجاد کند. برای TCP مقصد، رسیدگی به یک سگمنت تکراری، کار ساده‌ای محسوب می‌شود. TCP مقصد انتظار جریان پیوسته‌ای از بایتها را دارد. وقتی که سگمنتی از راه می‌رسد که حاوی شماره ترتیبی کمتر از بایتهای تأیید شده قبلی است و لذا معرف یک سگمنت تکراری است، دور انداخته می‌شود.

* تصحیح اتوماتیک ACK گم شده

این سناریو وضعیتی را نشان می‌دهد که در آن یک ACK گم شده، به طور اتوماتیک توسط ACK بعدی جایگزین می‌شود. شکل ۱۲-۲۸ یک ACK گم شده را نشان می‌دهد که توسط گیرنده، ارسال شده بود. در مکانیزم ارسال تأییدیه، ممکن است TCP مبدأ نیز از گم شدن ACK با خبر نشود. از آنجا که TCP از یک سیستم تأییدیه تجمعی استفاده می‌کند، می‌توانیم بگوییم که ACK بعدی (901) به طور اتوماتیک، گم شدن ACK قبلی (701) را جبران می‌نماید.

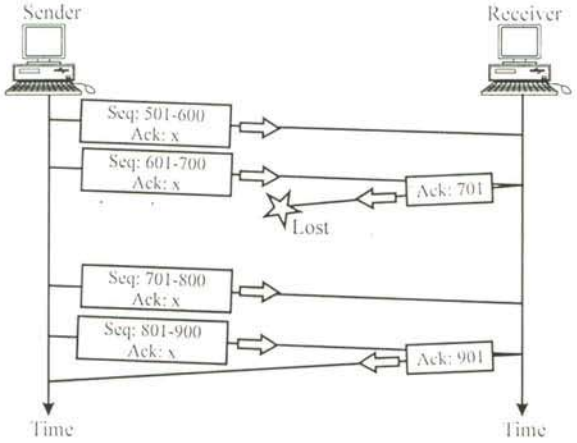
** جبران ACK گم شده از طریق ارسال مجدد یک سگمنت

اگر ACK بعدی برای مدت زیادی به تأخیر بیفتد و یا اصلاً ACK بعدی وجود نداشته باشد (ACK گم شده، آخرین ACK ارسال شده باشد)، تایمر RTO است که مشکل را برطرف می‌کند. به این ترتیب که پس از به پایان رسیدن مهلت ست شده این تایمر و عدم دریافت ACK، سگمنتهای تأیید نشده، مجدداً توسط فرستنده ارسال می‌شوند. نتیجه آن، ایجاد سگمنتهای تکراری است. گیرنده با دریافت سگمنتهای تکراری، آنها را دور می‌اندازد اما

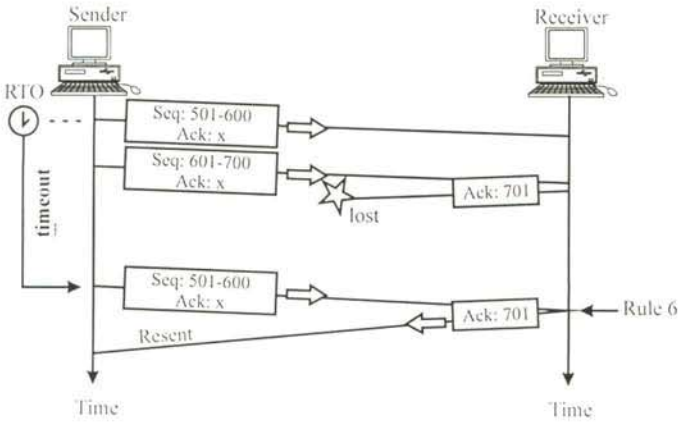
1- Duplicate segment

www.Abiri.ir

بلافاصله ACK آخری را نیز مجدداً ارسال می‌کند تا به اطلاع فرستنده سگمنتها برساند که سگمنت یا سگمنتهای ارسالی اول، صحیح و سالم دریافت شده‌اند. شکل ۱۲-۲۹ این سناریو را نشان می‌دهد.



شکل ۱۲-۲۸: ACK کم شده



شکل ۱۲-۲۹: جبران ACK گم شده از طریق ارسال مجدد یک سگمنت

دقت نمایید که گرچه ACK برای دو سگمنت دریافت نشده است، فقط یک سگمنت مجدداً ارسال می‌شود. وقتی که فرستنده، ACK مجدداً ارسال شده را دریافت می‌کند، می‌فهمد که هر دو سگمنتها، صحیح و سالم به مقصد رسیده‌اند زیرا که ACK ، حالت تجمعی دارد.

www.Abiri.ir

**** توقف گفتگو ناشی از ACK گم شده**

این یک وضعیت است که ممکن است ACK گم شده، منجر به توقف در گفتگو گردد و آن زمانی است که گیرنده، یک ACK با rwnd برابر با 0 ارسال می‌کند و موقتاً از فرستنده می‌خواهد که پنجره‌اش را متوقف کند. پس از مدتی، گیرنده می‌خواهد محدودیت فوق را بردارد، اما دیتایی برای ارسال ندارد. بنابراین یک سگمنت ACK می‌فرستد و با غیرصفر اعلام کردن rwnd ، محدودیت فوق را برمی‌دارد. مشکل زمانی پیش می‌آید که این ACK ، گم شود. فرستنده منتظر دریافت ACK ای است که مقدار rwnd را مخالف با 0 اعلام نماید و گیرنده فکر می‌کند که فرستنده، چنین ack ای را دریافت کرده است و لذا منتظر دریافت دیتای فرستنده می‌نشیند. چنین وضعیتی، ***توقف گفتگوی فرستنده و گیرنده***[1] نامیده می‌شود. هر طرف منتظر دریافت پاسخی از سمت مقابل است و هیچ اتفاقی نمی‌افتد. تایمر ارسال مجدد نیز ست نمی‌شود. برای ممانعت از بروز چنین وضعیتی، یک تایمر به نام ***تایمر اصرار بر ارسال***[2] طراحی شده است.

مهم: اگر به‌ack های گم شده به درستی رسیدگی نشود، ممکن است به توقف گفتگوی فرستنده و گیرنده منجر شود.

۱۲-۸ کنترل ازدحام

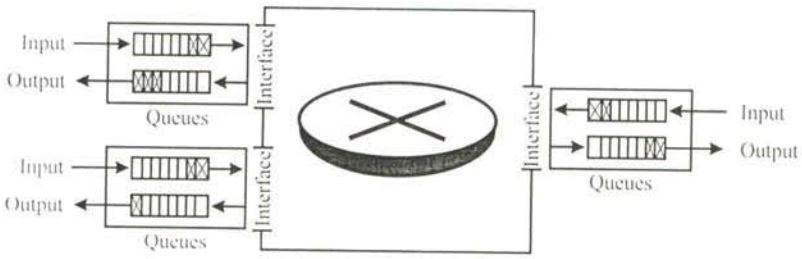
یک مطلب مهم در شبکه، ***ازدحام***[3] است. ازدحام در شبکه زمانی رخ می‌دهد که بار اعمال شده به شبکه (یعنی تعداد بسته‌های ارسالی به روی شبکه)، بیش از ***ظرفیت***[4] شبکه (یعنی تعداد بسته‌هایی که شبکه می‌تواند جابجا کند) باشد. ***کنترل ازدحام***[5] به مکانیزمها و تکنیکهایی برای کنترل ازدحام و نگه داشتن بار شبکه، کمتر از ظرفیت شبکه، اشاره می‌کند.

ممکن است این سؤال مطرح شود که چرا در شبکه ازدحام رخ می‌دهد. ازدحام در هر سیستمی که ممکن است دچار وضعیت انتظار شود، رخ می‌دهد. به عنوان مثال، ممکن است در یک اتوبان، ازدحام رخ دهد، زیرا که هر خللی در جریان طبیعی حرکت ماشینها، مثلاً وقوع یک تصادف در ساعات ترافیک سنگین، می‌تواند منجر به انسداد جاده‌ها و ایجاد وضعیت انتظار گردد.

1- deadlock
2- persistence timer
3- Congestion
4- capacity
5- Congestion control

www.Abiri.ir

از آنجا که سویچها و روترها، دارای صف (بافرهایی که بسته‌ها، قبل و بعد از پردازش در آنها می‌نشینند) می‌باشند، در یک شبکه یا شبکه‌بندی ازدحام رخ می‌دهد. به عنوان مثال، یک روتر برای هر اینترفیسش، یک صف ورودی و یک صف خروجی دارد. همانطور که شکل ۱۲-۳۰ نشان می‌دهد، وقتی که بسته‌ای به یک اینترفیس روتر می‌رسد، قبل از ترک روتر، سه عمل صورت می‌گیرد:



شکل ۱۲-۳۰: صفهای روتر

۱- بسته در انتهای صف ورودی منتظر پردازش می‌نشیند.

۲- وقتی که بسته به جلوی صف می‌رسد، ماجول پردازشگر روتر، بسته را از صف ورودی برمی‌دارد و به کمک جدول مسیردهی و آدرس مقصد، شماره اینترفیسی را پیدا می‌کند که بسته باید از طریق آن اینترفیس، به سمت مقصدش ارسال گردد.

۳- بسته در صف خروجی اینترفیس مناسب قرار می‌گیرد و منتظر نوبتش می‌ماند تا ارسال شود.

لازم است که از دو مطلب، مطلع باشیم. اول آنکه اگر نرخ ورود بسته‌ها بیش از نرخ پردازش آنها باشد، صفهای ورودی، بلندتر و بلندتر می‌شوند. دوم آنکه اگر نرخ خروج بسته‌ها، کمتر از نرخ پردازش آنها باشد، صفهای خروجی، بلندتر و بلندتر می‌شوند.

عملکرد شبکه

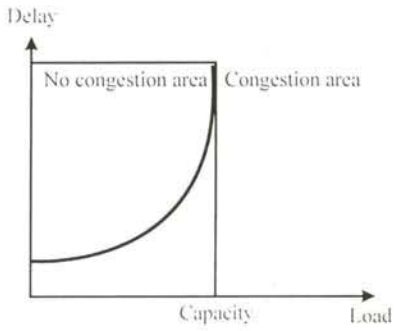
کنترل ازدحام شامل دو فاکتور است که عملکرد یک شبکه را مشخص می‌کنند: *تأخیر*[1] و *نرخ عبور دیتا*[2].

1- delay
2- throughput

www.Abiri.ir

❊ منحنی تأخیر- بار

شکل ۱۲-۳۱ رابطه بین تأخیر بسته و بار شبکه را به صورت ***منحنی تأخیر- بار***[1] نشان می‌دهد. دقت نمایید زمانی که بار شبکه خیلی کمتر از ***ظرفیت***[2] شبکه باشد، حداقل تأخیر را خواهیم داشت. این حداقل تأخیر، از تأخیر انتظار و تأخیر پردازش تشکیل شده است و هر دو آنها قابل صرفنظر کردن می‌باشند. با این حال، وقتی که بار شبکه به حد ظرفیت شبکه می‌رسد، تأخیر به سرعت افزایش می‌یابد زیرا که در این حالت لازم است زمان انتظار در صفها (برای همه روترهای موجود در مسیر) را به کل تأخیر، اضافه کنیم. دقت نمایید زمانی که بار شبکه از ظرفیت شبکه بیشتر باشد، میزان تأخیر به بینهایت می‌رسد. اگر این مطلب، واضح نیست، سایز صفها را زمانی که تقریباً هیچ بسته‌ای به مقصد نمی‌رسد و یا با تأخیر بینهایت به مقصد می‌رسند را در نظر بگیرید. صفها، بلندتر و بلندتر می‌شوند. تأخیر یک اثر منفی روی بار و در نتیجه ازدحام دارد. وقتی که بسته‌ای دچار تأخیر می‌شود، مبدأ در اثر عدم دریافت ACK، بسته را مجدداً ارسال می‌کند که تأخیر و ازدحام را بدتر می‌کند.



شکل ۱۲-۳۱: منحنی تأخیر- بار

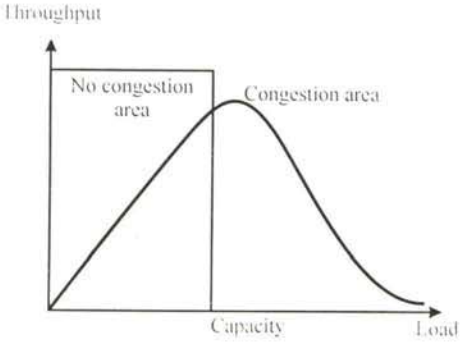
❊ منحنی نرخ عبور- بار

می‌توان نرخ عبور دیتا در شبکه را به عنوان تعداد بسته‌هایی که در یک واحد زمانی از میان شبکه عبور می‌کنند، تعریف کرد. حال می‌توانیم مطابق شکل ۱۲-۳۲ ، میزان نرخ عبور دیتا در شبکه را برحسب بار شبکه به صورت منحنی ***نرخ عبور- بار***[3] ترسیم نماییم.

1- Delay versus Load
2- Capacity
3- Throughput versus Load

www.Abiri.ir

دقت نمایید که وقتی بار شبکه کمتر از ظرفیت شبکه است، نرخ عبور دیتا در شبکه متناسب با بار شبکه، افزایش می‌یابد. انتظار داریم بعد از آنکه بار شبکه به حد ظرفیت شبکه رسید، نرخ عبور، ثابت باقی بماند، اما آنچه واقعاً اتفاق می‌افتد این است که نرخ عبور، به شدت کاهش می‌یابد. دلیل آن این است که در این حالت روترها، شروع به دور انداختن بسته‌ها می‌کنند. وقتی که بار شبکه از حد ظرفیت شبکه فراتر می‌رود، صفها پر می‌شوند و روترها باید برخی از بسته‌ها را دور بیندازند. دور انداختن بسته‌ها باعث کاهش تعداد بسته‌های موجود در شبکه نمی‌شود زیرا وقتی که فرستندهٔ بسته‌ها، به کمک تایمر ارسال مجدد خود، متوجه نرسیدن بسته‌ها به مقصد می‌گردد، آنها را مجدداً ارسال می‌کند.



شکل ۱۲-۳۲: منحنی نرخ عبور - بار

مکانیزم کنترل ازدحام

کنترل ازدحام به تکنیکها و مکانیزمهایی اشاره دارد که یا قبل از وقوع ازدحام، مانع از بروز آن می‌شوند و یا بعد از آنکه ازدحام رخ داد، ازدحام را از بین می‌برند. در حالت کلی می‌توانیم مکانیزمهای کنترل ازدحام را به دو گروه عمده تقسیم کنیم: ***کنترل حلقه باز ازدحام***[1] که مانع از بروز ازدحام می‌شود و ***کنترل حلقه بسته ازدحام***[2] که پس از وقوع ازدحام، آن را از بین می‌برد.

* کنترل حلقه باز ازدحام

در این متد کنترل ازدحام، سیاستهایی اعمال می‌شوند که مانع از بروز ازدحام می‌گردند. کنترل ازدحام در این مکانیزمها، یا از طریق مبدأ و یا از طریق مقصد، اعمال می‌شود. خلاصه‌ای از سیاستهایی که می‌توانند مانع از بروز ازدحام شوند را ارائه می‌نماییم.

1- Open-loop Congestion control
2- Closed-loop Congestion control

www.Abiri.ir

❊❊ سیاست ارسال مجدد

یک ***سیاست ارسال مجدد***[1] مناسب، می‌تواند مانع از بروز ازدحام شود. سیاست ارسال مجدد و تایمرهای ارسال مجدد باید به گونه‌ای طراحی شوند که کارآیی را بهینه نمایند و در عین حال، مانع از بروز ازدحام شوند.

❊❊ سیاست ارسال تأییدیه

سیاست ارسال تأییدیه[2] که توسط گیرنده اعمال می‌شود، نیز ممکن است بر ازدحام اثر گذارد. اگر گیرنده برای هر بسته‌ای که دریافت می‌کند، ACK نفرستد، می‌تواند با این کار، سرعت فرستنده را پایین آورد و مانع از بروز ازدحام شود.

❊❊ سیاست دورانداختن بسته‌ها

یک ***سیاست دور انداختن بسته‌ها***[3] که روترها به طور مناسب اعمال کنند، نه تنها می‌تواند مانع از بروز ازدحام شود، بلکه به ماهیت انتقال نیز لطمه‌ای وارد نخواهد شد. به عنوان مثال در انتقال صوت، اگر سیاست دور انداختن بسته‌ها این باشد که هنگام محتمل شدن وقوع ازدحام، بسته‌های کم اهمیت دور انداخته شوند، با این کار، نه تنها هنوز کیفیت صدا حفظ می‌شود، بلکه مانع از بروز ازدحام نیز می‌گردد.

❊ کنترل حلقه بسته ازدحام

مکانیزمهای کنترل حلقه بسته ازدحام، سعی می‌کنند که پس از وقوع ازدحام، آن را برطرف نمایند. پروتوکلهای مختلف، از مکانیزمهای مختلفی استفاده کرده‌اند. در اینجا درباره برخی از آنها صحبت می‌کنیم.

❊❊ درخواست دیتای ارسالی کمتر

وقتی که روتری دچار ازدحام می‌شود، می‌تواند از روتر قبل از خود که برای او دیتا می‌فرستد بخواهد نرخ ارسال بسته‌هایش را پایین آورد. این عمل می‌تواند همینطور ادامه یابد تا به روتری که دقیقاً بعد از کامپیوتر فرستنده بسته‌ها قرار گرفته است، برسد. به این مکانیزم، ***کم کردن فشار***[4] گفته می‌شود.

1- retransmission policy
2- acknowledgment Policy
3- discard policy
4- back pressure

www.Abiri.ir

❊ نقطه انسداد

نقطه انسداد[1] ، بسته‌ای است که روتر برای کامپیوتر ارسال کننده بسته‌ها می‌فرستد و طی آن وقوع ازدحام را به اطلاع او می‌رساند. این نوع کنترل، مشابه با پیغام ICMP « مبدا، کمتر بسته بفرست » می‌باشد.

❊ نشانه ضمنی وقوع ازدحام

کامپیوتر ارسال کننده بسته‌ها می‌تواند به طور ضمنی دریابد که ازدحامی در شبکه رخ داده است و لذا نرخ ارسال دیتایش را پایین آورد. به این نوع نشانه از وقوع ازدحام، ***نشانه ضمنی***[2] گفته می‌شود. به عنوان مثال، افزایش تأخیر در دریافت ACK می‌تواند نشانه‌ای از بروز ازدحام در شبکه باشد.

❊ نشانه صریح وقوع ازدحام

روترهایی که از وقوع ازدحام باخبر می‌شوند، می‌توانند به عنوان مثال، بیتی را در یک بسته ست کنند و به این ترتیب صریحاً به اطلاع فرستنده یا گیرنده برسانند که ازدحامی رخ داده است. به این نوع نشانه از وقوع ازدحام، ***نشانه صریح***[3] گفته می‌شود.

کنترل ازدحام در TCP

پارامترهای مختلفی در کنترل ازدحام در TCP دخیل هستند که به شرح زیر می‌باشند:

❊ پنجره ازدحام

قبلاً درباره کنترل فلو صحبت کردیم و سعی کردیم راه حلهایی برای جلوگیری از لبریز شدن گیرنده از دیتا ارائه نماییم. گفتیم که سایز پنجره فرستنده، توسط فضای موجود در بافر گیرنده (rwnd) تعیین می‌شود. به عبارت دیگر فرض کردیم، فقط گیرنده می‌تواند سایز پنجره فرستنده را برای او تعیین نماید و لذا از اثر شبکه در این رابطه، کاملاً صرفنظر کردیم. اما اگر شبکه نتواند با سرعتی که فرستنده دیتا را تولید می‌کند، آنها را به دست گیرنده برساند، باید به گونه‌ای از فرستنده بخواهد، سرعت ارسال دیتا را کم کند. به عبارت دیگر علاوه بر گیرنده، شبکه نیز در تعیین اندازه پنجره فرستنده، مؤثر است.

1- choke point
2- implicit signaling
3- Explicit signaling

www.Abiri.ir

امروزه سایز پنجره فرستنده، نه تنها توسط گیرنده، بلکه توسط ازدحام شبکه نیز تعیین می‌شود. فرستنده حاوی دو دسته اطلاعات است: سایز پنجره‌ای که گیرنده اعلام کرده است (rwnd) و سایز پنجره ازدحام (cwnd). سایز واقعی پنجره برابر با مینیمم این دو عدد است:

سایز واقعی پنجره =minimum (rwnd, cwnd)

* نحوه کنترل ازدحام

عملکرد کلی TCP برای مقابله با وقوع ازدحام در شبکه، حاوی سه مرحله است: ***شروع آهسته : افزایش نمایی***[1] ، ***جلوگیری از وقوع ازدحام : افزایش خطی***[2] و ***آشکارسازی ازدحام : کاهش مضربی***[3] . در مرحله شروع آهسته، فرستنده با سرعت خیلی کمی شروع به ارسال دیتا می‌کند، اما سریعاً به نرخ ارسال دیتا می‌افزاید تا جایی که به یک ***حد آستانه***[4] می‌رسد. وقتی که به حد آستانه رسید، نرخ ارسال دیتا کاهش داده می‌شود تا مانع از وقوع ازدحام در شبکه شود. بالاخره اگر ازدحام آشکار شود، فرستنده مجدداً وارد مرحله شروع آهسته و یا مرحله جلوگیری از وقوع ازدحام می‌شود و این بستگی به نحوه آشکارسازی ازدحام دارد.

** شروع آهسته : افزایش نمایی

یکی از الگوریتمهایی که در کنترل ازدحام TCP به کار می‌رود، الگوریتم شروع آهسته : افزایش نمایی نامیده می‌شود. در این الگوریتم، سایز پنجره ازدحام (cwnd) ، از سایزی برابر با ***ماکزیمم سایز سگمنت***[5] آغاز می‌شود. در خلال مرحله برقراری اتصال، مقدار MSS توسط ***گزینه‌ای***[6] به همین نام تعیین می‌شود. هر زمان که ACK برای سگمنتی دریافت می‌شود، سایز پنجره به اندازه یک MSS افزایش داده می‌شود.

همانطور که نام این الگوریتم نشان می‌دهد، شروع این الگوریتم، آهسته است اما سرعت آن به صورت نمایی افزایش می‌یابد. برای روشن شدن مطلب، اجازه دهید نگاهی به شکل ۱۲-۳۳ بیندازیم. دقت نمایید که برای قابل فهم‌تر شدن موضوع و کوچکتر شدن شکل، سه نوع ساده‌سازی را به کار برده‌ایم: الف- به جای استفاده از شماره بایت، از شماره سگمنت

1- Slow start : Exponential increase
2- Congestion avoidance : Additive increase
3- Congestion detection : Multiplicative Decrease
4- threshold
5- Maximum Segment Size: MSS
6- option

www.Abiri.ir

استفاده کرده‌ایم (گرچه هر سگمنت فقط حاوی یک بایت است). ب- فرض کرده‌ایم که rwnd خیلی بزرگتر از cwnd است بنابراین همواره سایز پنجره فرستنده برابر با cwnd خواهد بود. ج- اینگونه در شکل ۱۲-۳۳ نشان داده‌ایم که ACKها فقط پس از دریافت سگمنتهایی به طول پنجره، ارسال می‌شوند و معنی آن این است که از اصول تولید ACKها در اینجا صرفنظر کرده‌ایم.

فرستنده با cwnd= 1 MSS شروع می‌کند. به این معنا که فرستنده فقط می‌تواند یک سگمنت بفرستد. بعد از دریافت ACK برای سگمنت شماره 1 ، یک واحد به سایز پنجره ازدحام افزوده می‌شود و لذا سایز پنجره ازدحام به 2 می‌رسد. حالا فرستنده می‌تواند دو سگمنت دیگر بفرستد. وقتی که ACK برای این دو سگمنت دریافت شد، سایز پنجره به اندازه 2 واحد افزایش می‌یابد و لذا سایز cwnd به 4 می‌رسد. حال فرستنده می‌تواند چهار سگمنت دیگر را ارسال نماید. وقتی که ACK برای این چهار سگمنت دریافت شد، سایز پنجره به اندازه 4 واحد افزایش می‌یابد و به cwnd=8 می‌رسد. دقت نمایید که اصلاً مهم نیست که ACK برای تک تک سگمنتها، دو تا دو تا از سگمنتها و یا ... ارسال شود، زیرا که نتیجه کار، مشابه خواهد بود. آنچه که به دنبال می‌آید، بیانگر نمایی بودن نرخ افزایش سایز پنجره cwnd است (RTT همان ***زمان رفت و برگشت***[1] می‌باشد).

Start	→	cwnd = 1		
After 1 RTT	→	cwnd = 1x2 = 2	→	2^1
After 2 RTT	→	cwnd = 2x2 = 4	→	2^2
After 3 RTT	→	cwnd = 4x2 = 8	→	2^3

الگوریتم شروع آهسته نمی‌تواند تا بینهایت ادامه یابد. باید یک حد بالا یا آستانه برای توقف این مرحله وجود داشته باشد. فرستنده، متغیری به نام ***آستانه شروع آهسته***[2] دارد. وقتی که سایز پنجره برحسب بایت به این حد می‌رسد، فاز شروع آهسته، متوقف می‌شود و فاز بعدی آغاز می‌شود. در بسیاری از پیاده‌سازیها، مقدارssthresh برابر با 65535 بایت است.

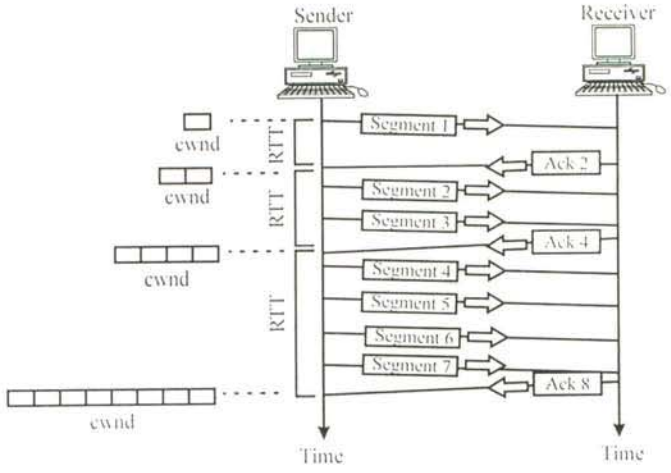
مهم: در الگوریتم شروع آهسته، سایز پنجره ازدحام تا رسیدن به یک حد بالا یا آستانه، به طور نمایی افزایش می‌یابد.

1- Round Trip Time : RTT
2- Slow start threshold: ssthresh

www.Abiri.ir

❊❊ جلوگیری از وقوع ازدحام: افزایش خطی

اگر با الگوریتم شروع آهسته آغاز نماییم، سایز پنجره ازدحام با سرعت نمایی افزایش می‌یابد. اگر بخواهیم قبل از وقوع ازدحام، مانع از بروز آن شویم، چیزی باید این سرعت رشد نمایی را کاهش دهد. TCP از الگوریتم دیگری به نام الگوریتم « جلوگیری از وقوع ازدحام : افزایش خطی » استفاده می‌کند که به جای نمایی، به صورت خطی افزایش می‌یابد. وقتی که سایز پنجره ازدحام به آستانه تعریف شده در الگوریتم شروع آهسته می‌رسد، فاز شروع آهسته متوقف می‌شود و فاز افزایش خطی آغاز می‌شود. در این الگوریتم، هر زمان که ACK برای یک پنجره کامل از سگمنتها ارسال شد. فقط یک واحد به سایز پنجره ازدحام اضافه می‌شود. برای نشان دادن ایده کار، این الگوریتم را در مورد همان سناریویی که برای

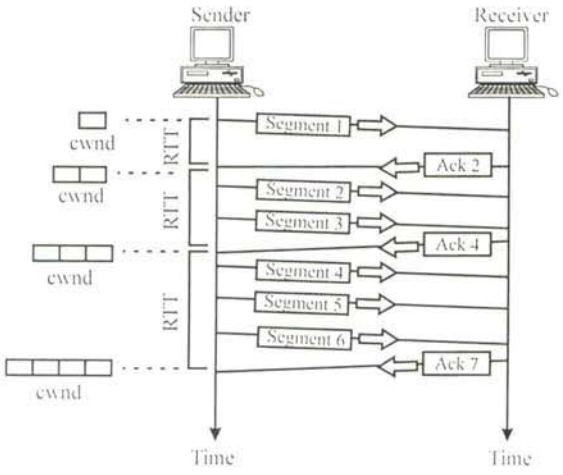


شکل ۱۲-۳۳: الگوریتم شروع آهسته : افزایش نمایی

الگوریتم شروع آهسته داشتیم، به کار می‌بریم، ولی باید توجه نمایید که در عمل، زمانی از الگوریتم جلوگیری از وقوع ازدحام استفاده می‌شود که سایز پنجره، خیلی بزرگتر از 1 باشد. شکل ۱۲-۳۴ این ایده را نشان می‌دهد. در این حالت زمانی که فرستنده برای یک پنجره کامل از سگمنتها، ACK را دریافت کرد، سایز پنجره فقط به اندازه یک واحد افزایش می‌یابد. آنچه که به دنبال می‌آید، نرخ افزایش خطی سایز پنجره cwnd را نشان می‌دهد:

Start → cwnd = 1
After 1 RTT → cwnd = 1+ 1 = 2
After 2 RTT → cwnd = 2 + 1= 3
After 3 RTT → cwnd = 3 + 1= 4

www.Abiri.ir



شکل ۱۲-۳۴: الگوریتم جلوگیری از وقوع ازدحام : افزایش خطی

مهم: در الگوریتم جلوگیری از وقوع ازدحام، سایز پنجره ازدحام به صورت خطی افزایش می‌یابد تا جایی که ازدحام، آشکار شود.

**** آشکارسازی ازدحام: کاهش مضربی**

اگر ازدحام رخ دهد، باید سایز پنجره ازدحام کاهش داده شود. تنها راهی که فرستنده می‌تواند حدس بزند که ازدحام رخ داده است، زمانی است که از فرستنده خواسته می‌شود تا سگمنتی را دوباره ارسال کند. در یکی از این دو مورد ممکن است نیاز به ارسال مجدد باشد : الف- وقتی که تایمر RTO به پایان مهلت مقرر خود برسد. ب- وقتی که سه ACK تکراری دریافت شوند. در هر دو حالت، مقدار آستانه برابر با نصف سایز پنجره فعلی، ست می‌شود. به این الگوریتم، الگوریتم « آشکارسازی ازدحام : کاهش مضربی » گفته می‌شود. اکثر پیاده‌سازیهای TCP دو عکس‌العمل را از خود نشان می‌دهند:

۱- اگر تایمر RTO به پایان مهلت مقرر خود برسد، احتمال بیشتری هست که ازدحام رخ داده باشد. احتمالاً سگمنتی در شبکه، گم شده است و هیچ خبری درباره سگمنتهایی که به دنبال آن ارسال شده‌اند، نیز وجود ندارد. در این حالت TCP ، شدیداً به صورت زیر، عکس‌العمل نشان می‌دهد:

الف- مقدار آستانه را برابر با نصف سایز پنجره فعلی، ست می‌کند.

ب - cwnd را برابر با سایز یک سگمنت، ست می‌کند.

ج- فاز شروع آهسته را مجدداً آغاز می‌نماید.

www.Abiri.ir

۲- اگر سه ACK تکراری دریافت شده باشد، احتمال کمتری هست که ازدحام رخ داده باشد. ممکن است سگمنتی گم شده باشد اما چند تا از سگمنتهای بعد از آن، صحیح و سالم به مقصد رسیده‌اند، زیرا که سه ACK تکراری دریافت شده است. به این حالت، ***ارسال سریع و احیای سریع***[1] گفته می‌شود. در این حالت، همانطور که مطلب زیر نشان می‌دهد، TCP عکس‌العمل ضعیفتری را از خود نشان می‌دهد:

الف- مقدار آستانه را برابر با نصف سایز پنجره فعلی، ست می‌کند.

ب- cwnd را برابر با مقدار آستانه ست می‌کند (برخی از پیاده‌سازی‌ها، 3 MSS به آستانه اضافه می‌کنند).

ج- فاز جلوگیری از وقوع ازدحام را راه می‌اندازد.

مهم: اکثر پیاده‌سازیها در مقابل آشکار شدن ازدحام، به طور متفاوتی عکس‌العمل نشان می‌دهند:

الف- اگر به پایان رسیدن مهلت مقرر تایمر RTO ، وقوع ازدحام را خبر داده باشد، یک فاز جدید شروع آهسته، آغاز می‌گردد.

ب- اگر سه ACK تکراری، وقوع ازدحام را خبر داده باشند، یک فاز جدید جلوگیری از وقوع ازدحام، آغاز می‌گردد.

❊ خلاصه‌ای از عملکرد TCP در قبال ازدحام

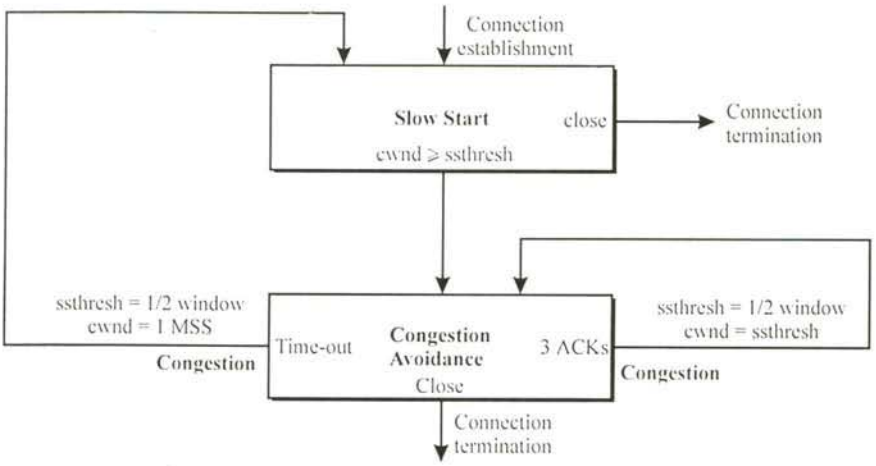
در شکل ۱۲-۳۵ ، عملکرد TCP در قبال ازدحام و نیز رابطه بین سه مرحله یا فاز مختلف را به طور خلاصه نشان داده‌ایم.

مثالی در شکل ۱۲-۳۶ ارائه کرده‌ایم. فرض می‌کنیم ماکزیمم سایز پنجره برابر با 32 سگمنت باشد. آستانه برابر با 16 سگمنت (نصف ماکزیمم سایز پنجره) ست شده است. در فاز « شروع آهسته : افزایش نمایی » سایز پنجره از 1 شروع می‌شود و به صورت نمایی تا رسیدن به آستانه، افزایش می‌یابد. پس از رسیدن به آستانه، الگوریتم « جلوگیری از وقوع ازدحام : افزایش خطی » اجازه می‌دهد که سایز پنجره به صورت خطی افزایش یابد تا اینکه یا تایمر به پایان مهلت مقرر برسد و یا اینکه پنجره به سایز ماکزیمم خود برسد. در شکل ۱۲-۳۶، زمانی که سایز پنجره به 20 می‌رسد، تایمر به پایان مهلت مقرر می‌رسد. در این لحظه، الگوریتم « آشکارسازی ازدحام : کاهش مضربی »، کنترل کار را به دست می‌گیرد و مقدار آستانه را برابر با نصف سایز پنجره، قبل از پایان مهلت مقرر تایمر، ست می‌کند. از آنجا که هنگام پایان مهلت

1- Fast transmission and fast recovery

www.Abiri.ir

مقرر تایمر، مقدار سایز پنجره برابر با 20 بود، مقدار جدید آستانه برابر با 10 می‌شود (Threshold = 20/2 = 10).



شکل ۱۲-۳۵: خلاصه‌ای از عملکرد TCP در قبال ازدحام

TCP مجدداً وارد فاز « شروع آهسته : افزایش نمایی » می‌شود و با سایز پنجره 1 شروع می‌کند و وقتی که به حد آستانه جدید می‌رسد، وارد فاز « جلوگیری از وقوع ازدحام : افزایش خطی » می‌گردد. وقتی که سایز پنجره به 12 می‌رسد، سه ACK تکراری دریافت می‌شوند. مجدداً فاز « آشکار سازی ازدحام : کاهش مضربی » کار را به دست می‌گیرد. این بار مقدار آستانه برابر با 12/2=6 می‌شود و TCP وارد فاز « جلوگیری از وقوع ازدحام : افزایش خطی » می‌شود. او در این فاز باقی می‌ماند تا اینکه تایمر مجدداً به انقضای مهلت مقرر برسد یا رویداد سه ACK تکراری مجدداً رخ دهد.

۱۲-۹ تایمرهای TCP

همانطور که شکل ۱۲-۳۷ نشان می‌دهد، برای آنکه TCP ، هموار کار کند، اکثر پیاده‌سازیهای TCP از حداقل چهار تایمر استفاده می‌کنند.

تایمر ارسال مجدد

TCP از یک ***تایمر ارسال مجدد***[1] استفاده می‌کند که زمان انتظار برای دریافت ACK را نگه می‌دارد و لذا نشان می‌دهد که چه زمانی باید دوباره سگمنتی را ارسال کرد. وقتی که

1- retransmission timer

www.Abiri.ir

TCP، سگمنتی را ارسال می‌کند، یک تایمر RTO را برای آن سگمنت، ست می‌کند. دو حالت ممکن است رخ دهد:

۱- اگر قبل از آنکه این تایمر به انتهای مهلت مقرر برسد، برای آن سگمنت، ACK دریافت شود، این تایمر خاموش می‌شود.

۲- اگر قبل از آنکه ACK از راه برسد، تایمر RTO به پایان مهلت مقرر برسد، سگمنت دوباره ارسال می‌شود و تایمر RTO مجدداً ست می‌شود.

* زمان رفت و برگشت: RTT

برای محاسبه مهلت مقرر برای ارسال مجدد (RTO) ، ابتدا باید ***زمان رفت و برگشت***[1] را محاسبه کنیم. محاسبه RTT در TCP یک فرایند مفصل است و لذا آن را قدم به قدم به کمک چند مثال توضیح می‌دهیم.

** RTT اندازه‌گیری شده : RTT_M

لازم است بدانیم چه مدت زمانی طول می‌کشد تا سگمنتی را ارسال کنیم و تأییدیه‌ای برای آن دریافت کنیم. این معنی ***RTT اندازه‌گیری شده***[2] است. باید به خاطر داشته باشیم که سگمنتها و تأییدیه‌های متناظر با آنها، یک رابطه یک به یک ندارند، یعنی ممکن است یک تأییدیه، برای چندین سگمنت ارسال شود. زمان رفت و برگشت اندازه‌گیری شده برای یک سگمنت، زمان لازم برای رسیدن سگمنت به مقصد و دریافت تأییدیه مبنی بر رسیدن آن است. البته تأییدیه ممکن است سگمنتهای دیگری را نیز در برگیرد. دقت نمایید که در TCP، در هر زمان فقط یک اندازه‌گیری RTT می‌تواند در دست محاسبه باشد. معنی آن این است که وقتی اندازه‌گیری RTT آغاز شد، تا زمانی که مقدار آن بدست نیامده است، اندازه‌گیری RTT دیگری آغاز نخواهد شد.

مهم : در هر لحظه در TCP ، فقط اندازه‌گیری یک RTT می‌تواند در دست محاسبه باشد.

** RTT هموار شده : RTT_S

مقدار اندازه‌گیری شده RTT (RTT_M) ، مرتباً تغییر می‌کند و مقدار نوسانات آن در اینترنت امروزی، آنقدر زیاد است که نمی‌توان مستقیماً از آن به عنوان زمان مقرر برای ارسال

1- Round Trip Time : RTT
2- Measured RTT : RTT_M

www.Abiri.ir

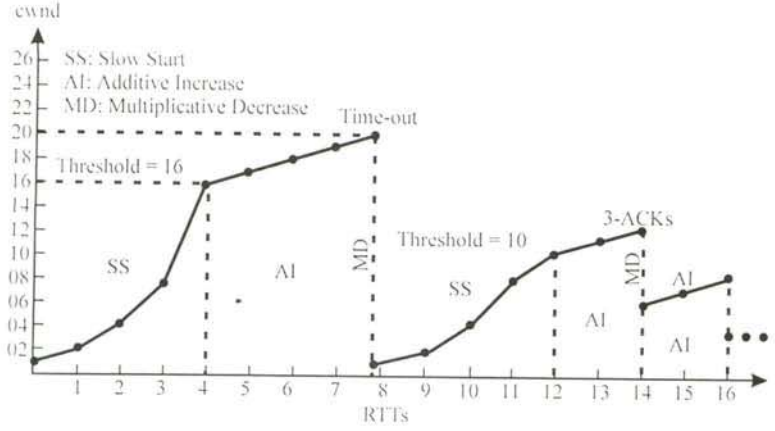
مجدد (RTO) استفاده کرد. اکثر پیاده‌سازی‌ها از یک ***RTT هموار شده***[1] استفاده می‌کنند که با اعمال وزنهایی به RTT_M و RTT_S که در مرحله قبل محاسبه شده است، به دست می‌آید:

مقدار اولیه ندارد → مقدار اولیه

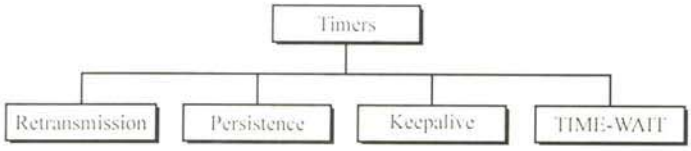
بعد از اولین اندازه‌گیریِ RTT_M → $RTT_S = RTT_M$

بعد از هر اندازه‌گیری دیگرِ RTT_M → $RTT_S = (1 - a)\, RTT_S + a.RTT_M$

مقدار ضریب a بستگی به نوع پیاده‌سازی دارد اما غالباً برابر با 1/8 است. به عبارت دیگر مقدار جدید RTT_S به صورت حاصل جمع 7/8 مقدار قبلی RTT_S و 1/8 مقدار فعلی RTT_M محاسبه می‌شود.



شکل ۱۲-۳۶: مثالی از ازدحام



شکل ۱۲-۳۷: تایمرهای TCP

1- Smoothed RTT : RTT_S

www.Abiri.ir

**** میزان انحراف RTT : RTT_D**

اکثر پیاده‌سازی‌ها صرفاً از RTT_S استفاده نمی‌کنند. آنها همچنین ***میزان انحراف RTT***[1] را نیز به کار می‌برند که با استفاده از فرمول زیر از روی مقادیر RTT_S و RTT_M محاسبه می‌شود:

مقدار اولیه ← مقدار اولیه

بعد از اولین اندازه‌گیریِ RTT_M ← $RTT_D = RTT_M / 2$

بعد از هر اندازه‌گیری دیگرِ RTT_M ← $RTT_D = (1-b)\, RTT_D + b.|\, RTT_S - RTT_M\,|$

مقدار b نیز وابسته به نوع پیاده‌سازی است اما مقدار آن معمولاً برابر با 1/4 است.

*** مهلت مقرر برای ارسال مجدد : RTO**

مقدار RTO بستگی به مقدار هموار شده زمان رفت و برگشت (RTT_S) و نیز میزان انحراف آن (RTT_D) دارد. اکثر پیاده‌سازی‌ها از فرمول زیر برای محاسبه RTO استفاده می‌کنند:

مقدار اولیه ندارد ← مقدار اولیه

بعد از هر اندازه‌گیری ← $RTO = RTT_S + 4RTT_D$

به عبارت دیگر، آخرین مقدار RTT_S را با 4 برابر مقدار RTT_D (که غالباً عدد کوچکی است) باید جمع کنیم تا مقدار RTO بدست آید.

**** مثال ۱۰**

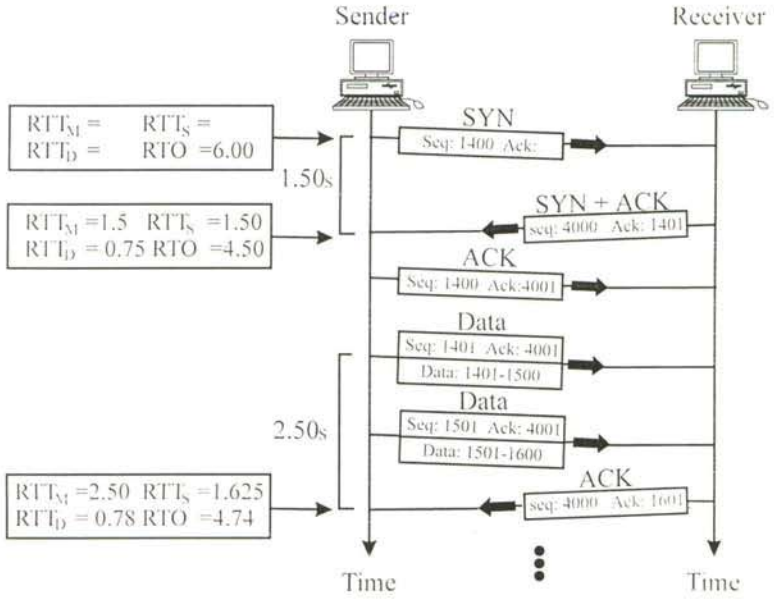
اجازه دهید یک مثال فرضی بزنیم. شکل ۱۲-۳۸ بخشی از یک اتصال را نشان می‌دهد. فاز برقراری اتصال و بخشی از فاز رد و بدل شدن دیتا در شکل ۱۲-۳۸ نشان داده شده‌اند.

۱- وقتی که سگمنت SYN ارسال می‌شود، RTT_M ، RTT_S و RTT_D مقداری ندارند. مقدار RTO برابر با 6.00 ثانیه ست می‌شود. بنابراین در این لحظه، مقدار RTO برابر است با :

$RTO = 6$

1- RTT Deviation : RTT_D

www.Abiri.ir



شکل ۱۲-۳۸ : مثال ۱۰

۲- وقتی که سگمنت SYN+ACK از راه می‌رسد، RTT_M اندازه‌گیری می‌شود و مقدار 1.5 ثانیه برای آن بدست می‌آید. در این لحظه، مقدار متغیرها عبارتند از :

$RTT_M = 1.5$
$RTT_S = 1.5$
$RTT_D = 1.5 / 2 = 0.75$
$RTO = 1.5 + 4 \times 0.75 = 4.5$

۳- وقتی که اولین سگمنت دیتا ارسال می‌شود، اندازه‌گیری جدید RTT آغاز می‌شود. دقت نمایید که وقتی که فرستنده، سگمنت ACK را می‌فرستد، محاسبه RTT را آغاز نمی‌کند زیرا که ***شماره ترتیبی***[1] مصرف نمی‌شود و ***مهلت مقرری***[2] هم وجود ندارد. برای دومین سگمنت دیتا نیز، اندازه‌گیری RTT آغاز نمی‌شود زیرا که یک اندازه‌گیری هم اکنون در جریان است. برای محاسبه مقدار بعدی RTT_M ، از رسیدن آخرین سگمنت ACK استفاده می‌شود. گرچه آخرین سگمنت ACK ، تأییدیه‌ای بر دریافت هر دو سگمنت دیتا است (حالت تجمعی دارد)، رسیدن آن به دست فرستنده، امکان محاسبه RTT_M را برای اولین سگمنت، مهیا می‌سازد. اکنون مقدار متغیرها برابر هستند با :

$RTT_M = 2.5$

1- Sequence number
2- time-out

www.Abiri.ir

$RTT_S = 7/8\ (\ 1.5\ ) + 1/8\ (\ 2.5\ ) = 1.625$
$RTT_D = 3/4\ (\ 0.75\ ) + 1/4\ |\ 1.625 - 2.5\ | = 0.78$
$RTO = 1.625 + 4\ (\ 0.78\ ) = 4.74$

*** الگوریتم کارن**

فرض کنید که در خلال زمان ارسال مجدد، تأییدیه دریافت سگمنت ارسال شده، به دست فرستنده نرسد و لذا آن سگمنت، مجدداً ارسال شود. وقتی که TCP فرستنده، تأییدیه را دریافت می‌کند، او نمی‌داند که این تأییدیه برای سگمنت اصلی است و یا برای سگمنتی است که مجدداً ارسال شده است. مقدار RTT جدید بستگی به این دارد که چه زمانی سگمنت را ارسال کرده‌ایم. اما اگر سگمنت اصلی گم شود و تأییدیه دریافتی برای سگمنتی باشد که مجدداً ارسال شده است، مقدار RTT را باید از زمانی که سگمنت را مجدداً ارسال کرده‌ایم، محاسبه کنیم. برای حل این مشکل، ***کارن***[1] یک راه حل ساده را ارائه کرد. در محاسبه مقدار جدید RTT ، زمان رفت و برگشت سگمنتی که مجدداً ارسال شده است را در نظر نگیرید. بنابراین فقط زمانی مقدار RTT را به روز رسانی کنید که سگمنتی را ارسال کرده‌اید و بدون اینکه نیازی به ارسال مجدد باشد، تأییدیه دریافت آن به دست فرستنده رسیده باشد.

مهم : TCP در محاسبه یک RTO جدید، مقدار RTT سگمنتی که مجدداً ارسال شده است را در نظر نمی‌گیرد.

اگر یک ارسال مجدد رخ دهد، مقدار RTO چقدر خواهد بود؟ اکثر پیاده‌سازی‌های TCP از یک ***استراتژی عقب‌نشینی نمایی***[2] استفاده می‌کنند، به این ترتیب که مقدار RTO به ازای هر ارسال مجدد، دو برابر می‌شود. بنابراین اگر سگمنتی، مجدداً یک بار دیگر ارسال شود، مقدار جدید RTO ، دو برابر مقدار قبلی خواهد شد. اگر باز هم همان سگمنت، ارسال شود، مقدار جدید RTO ، مجدداً دو برابر خواهد شد (و لذا چهار برابر مقدار اولیه می‌شود) و این روند افزایش نمایی برای ارسالهای مجدد بعدی نیز ادامه خواهد یافت.

**** مثال ۱۱**

شکل ۱۲-۳۹ ، ادامه مثال قبل می‌باشد. سگمنتی فرستاده می‌شود، سپس ارسال مجدد صورت می‌گیرد و الگوریتم کارن اعمال می‌شود.

1- Karn
2- exponential backoff strategy

www.Abiri.ir

اولین سگمنت در شکل ۱۲-۳۹ ارسال می‌شود، اما گم می‌شود. تایمر RTO بعد از 4.74 ثانیه به انتهای مهلت مقرر می‌رسد. به دلیل عدم دریافت ACK ، مجدداً سگمنت ارسال می‌شود و مقدار تایمر روی دو برابر مقدار قبلی یعنی مقدار جدید 2x4.74=9.48 ثانیه، ست می‌شود. این بار قبل از آنکه تایمر به انتهای مهلت مقرر برسد، ACK دریافت می‌شود. همانطور که می‌بینید، مقدار جدید RTO پس از ارسال مجدد سگمنت، محاسبه نشده است بلکه صرفاً قاعده « عقب‌نشینی نمایی » به آن اعمال شده است و لذا مقدار آن دو برابر شده است اما اندازه‌گیری جدیدی صورت نگرفته است (طبق قاعده کارن).

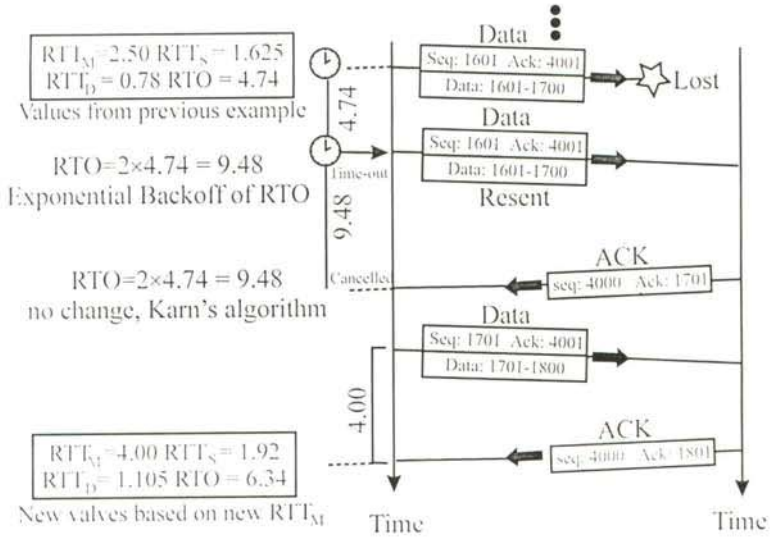
تایمر اصرار بر ارسال

برای رسیدگی به حالتی که پنجره‌ای به سایز 0 از طرف گیرنده، اعلام شده باشد، TCP به تایمر دیگری نیاز دارد. اگر TCP گیرنده، پنجره‌ای به سایز 0 را اعلام کرده باشد، TCP فرستنده، ارسال سگمنتها را متوقف می‌کند تا اینکه TCP گیرنده یک ACK با پنجره‌ای مخالف 0 را اعلام نماید. ممکن است این ACK ، گم شود. به خاطر داشته باشید که در TCP ، تأییدیه‌ای بر دریافت ACK ها فرستاده نمی‌شود. اگر این ACK گم شود، TCP گیرنده تصور می‌کند که کارش را انجام داده است و منتظر می‌ماند تا TCP فرستنده، سگمنتهای بیشتری را ارسال نماید. هیچ تایمر ارسال مجددی برای سگمنتی که فقط حاوی ACK است، ست نمی‌شود. TCP فرستنده، ACK را دریافت نکرده است و منتظر است تا TCP سمت مقابل، ACK ای بفرستد که در آن، سایز پنجره، مخالف با 0 اعلام شده باشد. هر دو TCPها منتظر خبری از یکدیگر می‌مانند و لذا ***توقف گفتگوی فرستنده و گیرنده***[1] پیش می‌آید.

برای برطرف کردن این مشکل، TCP از ***تایمر اصرار بر ارسال***[2] برای هر اتصالی که برقرار می‌کند، استفاده می‌کند. وقتی که TCP فرستنده، یک ACK با سایز پنجره برابر با 0 دریافت می‌کند، این تایمر را راه می‌اندازد. وقتی که این تایمر، به پایان مهلت مقرر برسد و هنوز ACK با سایز پنجره مخالف صفر، دریافت نشده باشد، TCP فرستنده، یک سگمنت به خصوص به نام ***سگمنت کاوش***[3] را ارسال می‌کند. این سگمنت حاوی فقط یک بایت دیتا است. یک شماره ترتیب نیز در این سگمنت هست اما هیچگاه تأییدیه‌ای بر این شماره ترتیب ارسال نخواهد شد. سگمنت probe به TCP گیرنده خبر می‌دهد که ACK ، گم شده است و باید مجدداً ارسال شود.

1- deadlock
2- persistence timer
3- Probe

www.Abiri.ir



شکل ۱۲-۳۹ : مثال ۱۱

مقدار تایمر اصرار بر ارسال، روی مقدار مهلت مقرر برای ارسال مجدد (RTO)، ست می‌شود. اگر در خلال مهلت مقرر این تایمر، خبری از گیرنده دریافت نشد، TCP فرستنده، یک سگمنت probe دیگر، می‌فرستد و تایمر اصرار بر ارسال را روی دو برابر مقدار قبل، ست می‌کند. در صورت عدم دریافت خبر از گیرنده، روند ارسال probe و دو برابر کردن‌های مقدار مهلت مقرر تایمر تا آنجا ادامه می‌یابد که به یک حد ماکزیممی می‌رسد (معمولاً 60 ثانیه). بعد از رسیدن به این حد ماکزیمم، در صورت عدم دریافت خبر از گیرنده، مشابه سابق probe ارسال می‌شود اما مهلت مقرر تایمر اصرار بر ارسال روی همان حد ماکزیمم 60 ثانیه ، ست می‌شود و این عمل تا آنجا ادامه می‌یابد که بالاخره یک ACK با سایز پنجره مخالف 0 از طرف TCP گیرنده، دریافت شود.

تایمر زنده نگه دارِ اتصال

برخی از پیاده‌سازی‌ها از یک تایمر، به نام ***تایمر زنده نگه دار اتصال***[1] برای ممانعت از غیر فعال ماندن یک اتصال TCP برای یک مدت طولانی، استفاده می‌کنند. فرض کنید که یک سرویس‌گیرنده، یک اتصال TCP را با یک سرور برقرار کند، قدری دیتا ارسال کند و سپس

1- keepalive Timer

www.Abiri.ir

ساکت بماند. ممکن است سرویس‌گیرنده، یک مشکل سخت‌افزاری پیدا کرده باشد. در این حالت، اتصال برقرار شده برای همیشه باز می‌ماند.

اکثر پیاده‌سازی‌ها برای برطرف کردن این مشکل، یک تایمر زنده نگه دار اتصال را در سرور قرار می‌دهند. هر بار که سرور، خبری از سرویس‌گیرنده دریافت می‌کند، این تایمر را مجدداً ست می‌کند. مهلت مقرری که این تایمر، روی آن، ست می‌شود، معمولاً برابر با 2 ساعت است. اگر سرور به مدت 2 ساعت، خبری از سرویس‌گیرنده دریافت نکند، یک سگمنت probe ارسال می‌کند. اگر بعد از ارسال 10 سگمنت probe به فاصله زمانی معمولاً 75 ثانیه، باز هم خبری از سرویس‌گیرنده نشد، فرض را بر این می‌گذارد که سرویس‌گیرنده، مشکلی پیدا کرده است و لذا سرور به اتصال، خاتمه می‌دهد.

تایمر انتظار قبل از قطع کامل

قبل از آنکه سرویس‌گیرنده به طور کامل اتصالی را قطع کند، از این ***تایمر انتظار قبل از قطع کامل***[1] استفاده می‌کند که مهلت مقرر آن برابر با MSL 2 است. قبلاً در بخشی که درباره دیاگرام گذر از حالتها صحبت کردیم، علت استفاده از این تایمر، مطرح شد.

۱۰-۱۲ options : گزینه‌ها

حجم ***گزینه‌ها***[2] در هدر TCP ، حداکثر می‌تواند 40 بایت باشد. گزینه‌ها یا برای ترازبندی هدر به کار می‌روند یا اطلاعات مضاعفی را به مقصد می‌رسانند. می‌توانیم دو گروه از گزینه‌ها را تعریف کنیم: گزینه‌های ***1 بایتی***[3] و ***چند بایتی***[4]. گروه اول شامل دو نوع گزینه است : گزینه ***EOP***[5] (که برای ترازبندی در انتهای لیست گزینه‌ها قرار می‌گیرد) و گزینه ***NOP***[6] (که برای ترازبندی، در ابتدا و یا در وسط سایر گزینه‌ها قرار می‌گیرد). اکثر پیاده‌سازی‌ها حاوی چند نوع گزینه در گروه دوم هستند : گزینه ***ماکزیمم سایز سگمنت***[7] ، گزینه ***ضریب بزرگنمایی پنجره***[8] ، گزینه ***برچسب زمانی***[9] ، گزینه ***SACK- مجاز***[10] و گزینه SACK (شکل ۱۲-۴۰ را ببینید).

1- TIME-WAIT Timer
2- options
3- Single-byte
4- Multiple-byte
5- End of option
6- No operation
7- Maximum Segment Size
8- Window scale factor
9- timestamp
10- SACK-permitted

www.Abiri.ir

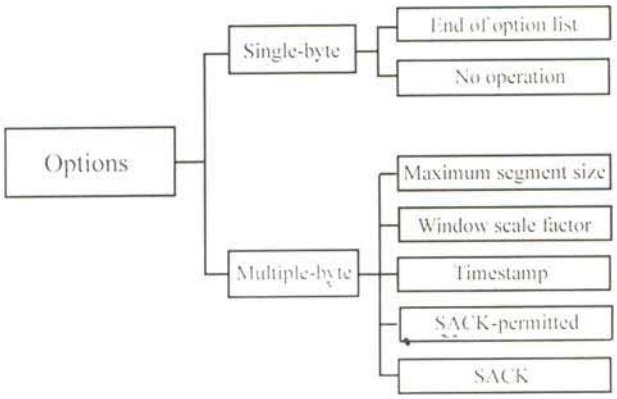
گزینه EOP

گزینه EOP به طول یک بایت است و برای ترازبندی، در انتهای گزینه‌های هدر قرار می‌گیرد. می‌توان از EOP فقط به عنوان آخرین گزینه و فقط به تعداد یکی، استفاده کرد. بعد از این گزینه، گیرنده انتظار دریافت دیتا را دارد. شکل ۱۲-۴۱، مثالی را نشان می‌دهد. بعد از هدر اصلی، یک گزینه سه بایتی آمده است و باید بعد از آن، دیتا قرار گیرد. اما به منظور ترازبندی هدر و رساندن طول آن به مضربی از چهار بایت، ابتدا یک گزینه EOP در انتهای گزینه 3 بایتی قرار می‌گیرد و سپس دیتا می‌آید.

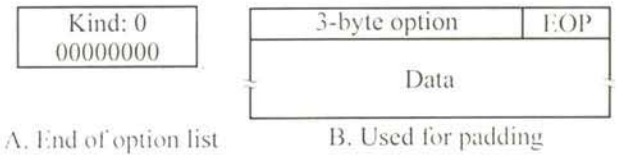
گزینه EOP ، دو گونه اطلاعات را به مقصد می‌رساند:

۱- گزینه دیگری بعد از آن در هدر نیست.

۲- دیتای لایه کاربرد، بعد از EOP ، قرار گرفته است.



شکل ۱۲-۴۰ : گزینه‌ها در TCP



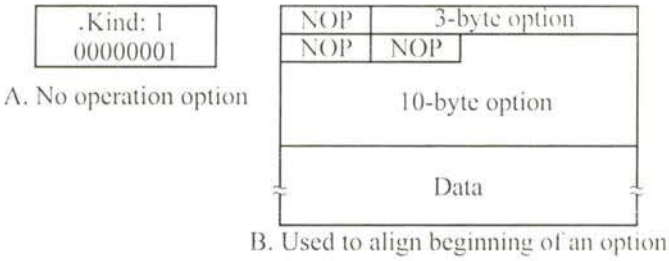
شکل ۱۲-۴۱ : گزینه EOP

مهم : فقط می‌توان از یک بایت EOP در هدر TCP استفاده کرد.

www.Abiri.ir

گزینه NOP

گزینه NOP نیز به طول 1 بایت است و در ابتدا و یا مابین سایر گزینه‌ها قرار می‌گیرد تا سایز هدر به مضربی از 4 بایت، برسد. معمولاً NOP قبل از گزینه می‌آید و طول آن را به مضربی از 4 بایت می‌رساند. به عنوان مثال در شکل ۱۲-۴۲ ، از NOP برای ترازبندی یک گزینه سه بایتی مانند گزینه « ضریب بزرگنمایی پنجره » و یک گزینه 10 بایتی مانند گزینه « برچسب زمانی » استفاده شده است.



شکل ۱۲-۴۲ : گزینه NOP

مهم : می‌توان از NOP ، بیش از یک بار استفاده کرد.

گزینه ماکزیمم سایز سگمنت : MSS

گزینه ماکزیمم سایز سگمنت (MSS)، بزرگترین واحد طول دیتایی که TCP گیرنده می‌تواند تحت یک سگمنت، دریافت کند را تعیین می‌نماید. MSS علی‌رغم نامش، ماکزیمم طول دیتای قابل دریافت تحت یک سگمنت، و نَه ماکزیمم طول خود سگمنت را نشان می‌دهد. شکل ۱۲-۴۳ فرمت گزینه ماکزیمم سایز سگمنت را نشان می‌دهد. مطابق شکل ۱۲-۴۳ یک فیلد Maximum segment size در این گزینه وجود دارد که طول آن 16 بیت است و مقدار آن را با نماد MSS نشان می‌دهیم. بنابراین ماکزیمم طول دیتای هر سگمنت TCP می‌تواند از 0 بایت تا 65535 بایت باشد.

در خلال فاز برقراری اتصال، مقدار MSS تعیین می‌شود. در زمانی که اتصال برقرار می‌شود، هر طرف، مقدار مورد نظر خود برای MSS را به اطلاع طرف مقابل می‌رساند (مقدار مورد نظر را در فیلد Maximum segment size گزینه ماکزیمم سایز سگمنت می‌نویسد). اگر یک طرف نخواهد مقداری برای MSS تعیین کند، از مقدار پیش فرض 536 بایت استفاده می‌شود.

www.Abiri.ir

Kind: 2 00000010	Length: 4 00000100	Maximum segment size
1 byte	1 byte	2 bytes

شکل ۱۲-۴۳ : گزینه ماکزیمم سایز سگمنت

مهم : در خلال برقراری اتصال، مقدار MSS تعیین می‌شود و در طول زمان رد و بدل شدن دیتا، مقدار آن، ثابت باقی می‌ماند.

*** ضریب بزرگنمایی پنجره**

فیلد سایز پنجره (window size) در هدر TCP ، سایز پنجره لغزان را تعیین می‌کند. این فیلد به طول 16 بیت است و به این معنی است که سایز پنجره می‌تواند از 0 تا 65535 بایت باشد. گرچه ممکن است حد بالای 65535 بایت، عدد بزرگی به نظر برسد، اما ممکن است در مواردی که قرار است دیتا از یک کانال طویل که پهنای باند زیادی هم دارد (که به آن ***کانال طویل و ضخیم***[1] گفته می‌شود) عبور کند، کافی نباشد.

برای افزایش سایز پنجره، از یک گزینه به نام ***ضریب بزرگنمایی پنجره***[2] استفاده می‌شود. شکل ۱۲-۴۴ فرمت این گزینه را نشان می‌دهد.

Kind: 3 00000011	Length: 3 00000011	Scale factor
1 byte	1 byte	1 byte

شکل ۱۲-۴۴ : گزینه ضریب بزرگنمایی پنجره

مطابق شکل ۱۲-۴۴، فیلدی به نام ضریب بزرگنمایی (scale factor) در این گزینه وجود دارد. برای محاسبه مقدار سایز پنجره، ابتدا 2 را به توان عدد موجود در این فیلد می‌رسانیم و سپس مقدار حاصل را در مقداری که فیلد window size در هدر TCP نشان می‌دهد، ضرب می‌کنیم. این مطلب را می‌توان با فرمول زیر بیان کرد:

$$\text{مقدار جدید window size} = \text{TCP مقدار window size در هدر} \times 2^{\text{scale factor}}$$

1- long fat pipe
2- window scale factor

www.Abiri.ir

گاهی اوقات به ضریب بزرگنمایی ، ***مقدار شیفت***[1] نیز گفته می‌شود ، زیرا ضرب یک عدد در توانی از 2 ، معادل با شیفت بیتهای آن عدد بر حسب باینری به سمت چپ می‌باشد. به عبارت دیگر می‌توان مقدار سایز پنجره را اینگونه نیز بدست آورد که آنچه در فیلد window size در هدر TCP نوشته شده است را بر حسب باینری بنویسیم و سپس به اندازه عددی که ضریب بزرگنمایی نشان می‌دهد، آن را به سمت چپ شیفت دهیم.

به عنوان مثال فرض کنید که مقدار ضریب بزرگنمایی برابر با 3 و عددی که برای window size در هدر TCP اعلام شده است، برابر با 32768 باشد. بنابراین سایز پنجره‌ای که می‌توان به کار برد برابر است با $32768 \times 2^3 = 262144$. اگر عدد 32768 را به صورت باینری بنویسیم و آن را به اندازه 3 بیت به سمت چپ شیفت دهیم، باز هم به همین نتیجه می‌رسیم.

گرچه ضریب بزرگنمایی می‌تواند تا 255 باشد، بزرگترین مقداری که TCP اجازه داده است برابر با 14 می‌باشد و لذا حداکثر سایز پنجره برابر با $2^{16} \times 2^{14} = 2^{30}$ می‌باشد که البته کوچکتر از ماکزیمم مقدار ممکن برای شماره ترتیب، یعنی 2^{32} است و این هم لازم است زیرا که سایز پنجره، نمی‌تواند بزرگتر از ماکزیمم مقدار مجاز برای شماره ترتیب باشد.

مقدار ضریب بزرگنمایی پنجره را نیز فقط می‌توان در فاز برقراری اتصال، تعیین کرد. در فاز رد و بدل شدن دیتا، مقدار ضریب بزرگنمایی پنجره، مقدار ثابتی خواهد ماند اما البته می‌توانیم مقدار window size که در هدر TCP وجود دارد را تغییر دهیم.

مهم : فقط در فاز برقراری اتصال می‌توان مقدار ضریب بزرگنمایی پنجره را تعیین کرد. در خلال نقل و انتقال دیتا، نمی‌توان مقدار آن را تغییر داد.

دقت نمایید که یک طرف ممکن است مقدار ضریب بزرگنمایی پنجره را برابر با 0 قرار دهد و معنی آن این خواهد بود که گرچه از این گزینه، پشتیبانی می‌کند اما نمی‌خواهد از آن برای این اتصال استفاده کند (وقتی که 2 به توان 0 برسد، برابر با 1 می‌شود و ضرب کردن 1 در window size، تغییری در آن ایجاد نمی‌کند).

گزینه برچسب زمانی

این یک گزینه 10 بایتی است و فرمت آن در شکل ۱۲-۴۵ نشان داده شده است. دقت نمایید سَمتی که اتصال را برقرار می‌کند، به همراه سگمنت برقراری اتصال (سگمنت حاوی SYN)، گزینه برچسب زمانی را نیز به کار می‌برد. اگر در سگمنت بعدی (سگمنت حاوی SYN+ACK) که از سَمت مقابل می‌آید، گزینه برچسب زمانی را دریافت کند، مجاز است که

1- shift count

www.Abiri.ir

از این گزینه استفاده نماید. در غیر این صورت، دیگر از این گزینه استفاده نخواهد کرد. گزینه برچسب زمانی دو کاربرد دارد: الف- به کمک آن می‌توان زمان رفت و برگشت را اندازه‌گیری کرد ب – مانع از آن شد که شماره ترتیبها ، تکرار شوند (که اصطلاحاً به آن **چرخش دوره‌ای**[1] گفته می‌شود).

Kind: 8 00001000	Length: 10 00001010
Timestamp value	
Timestamp echo reply	

شکل ۱۲-۴۵ : گزینه برچسب زمانی

❋ اندازه‌گیری RTT

می‌توان از گزینه برچسب زمانی برای اندازه‌گیری ***زمان رفت و برگشت***[2] استفاده کرد. وقتی که TCP آماده ارسال یک سگمنت است، مقدار فعلی کلاک سیستم را می‌خواند و مقدار آن را به صورت یک عدد 32 بیتی در فیلد timestamp value (مقدار برچسب زمانی) می‌نویسد. وقتی که گیرنده، یک ACK برای این سگمنت ارسال می‌کند، مقدار نوشته شده در فیلد timestamp value را در فیلد timestamp echo reply (پاسخ انعکاس برچسب زمانی) کپی می‌کند. وقتی که فرستنده، ACK را دریافت می‌کند، مقدار نوشته شده در فیلد timestamp echo reply را از مقداری که در آن لحظه، کلاک سیستم نشان می‌دهد، کم می‌کند تا مقدار RTT را بدست آورد.

دقت نمایید که لازم نیست کلاکهای فرستنده و گیرنده با هم سنکرون باشند زیرا که تمام محاسبات بر اساس کلاک فرستنده، انجام می‌شوند. همچنین دقت نمایید که فرستنده نیازی به ذخیره زمان ارسال سگمنت ندارد زیرا که این مقدار، در فیلد timestamp value موجود در گزینه برچسب زمانی که به همراه سگمنت ارسال می‌شود، نوشته شده است.

گیرنده باید مقدار دو متغیر را نگه دارد. اولین متغیر، lastack ، مقدار فیلد acknowledgment در آخرین ACK ای است که گیرنده ارسال کرده است. دومین متغیر، tsrecent ، مقدار timestamp value دریافتی است که هنوز گیرنده برای آن، پاسخی را ارسال نکرده است. وقتی که گیرنده، سگمنتی را دریافت می‌کند که شماره ترتیب بایتی از آن، با

1- wrap-around
2- Round Trip Time : RTT

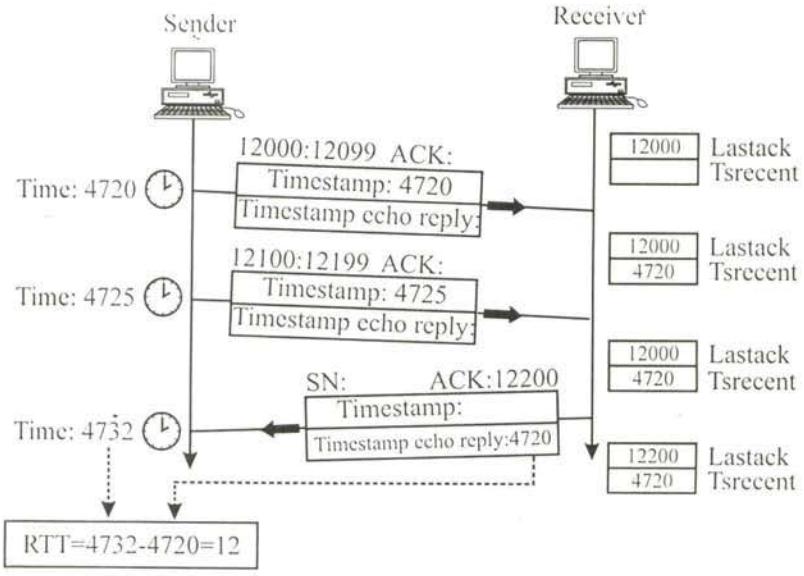
www.Abiri.ir

مقدار نوشته شده در lastack مطابقت داشته باشد، مقدار فیلد timestamp value سگمنت دریافتی را در متغیر tsrecent می‌نویسد. وقتی که گیرنده ACK می‌فرستد، مقدار tsrecent را در فیلد timestamp echo reply می‌نویسد.

مهم : یک کاربرد گزینه برچسب زمانی، محاسبه RTT است.

**** مثال ۱۲**

شکل ۱۲-۴۶ نحوه محاسبه RTT را برای یک طرف، نشان می‌دهد. همین روش محاسبه را می‌توان برای طرف دیگر به کار برد.



شکل ۱۲-۴۶ : مثال ۱۲

فرستنده، مقدار کلاک (به عنوان مثال، ثانیه‌هایی که از نیمه شب گذشته است) را در فیلد timestamp value اولین و دومین سگمنت می‌نویسد (به ترتیب 4720 و 4725). وقتی که ACK دریافت می‌شود (سگمنت سوم)، مقداری که فیلد timestamp echo reply نشان می‌دهد(4720) از مقدار فعلی کلاک (4732) کم می‌شود و در این مثال، مقدار RTT برابر با 4732-4720=12 ثانیه بدست می‌آید.

کار گیرنده ، بیشتر است. او مقدار فیلد acknowledgment در آخرین ACK ای که ارسال کرده است را در متغیر laskack نوشته است (12000). وقتی که سگمنت اول در شکل

www.Abiri.ir

۱۲-۴۶ ، به دست گیرنده می‌رسد، حاوی بایتهای از شماره ترتیب 12000 تا شماره ترتیب 12099 است. مقدار شماره ترتیب اولین بایت(12000)، مطابق با مقدار lastack می‌باشد. بنابراین گیرنده، مقدار نوشته شده در فیلد timastamp یعنی 4720 را داخل متغیر tsrecent کپی می‌کند. مقدار lastack هنوز برابر با 12000 است (گیرنده، هیچ ACK جدیدی نفرستاده است). وقتی که سگمنت دوم از راه می‌رسد، از آنجا که شماره ترتیب هیچ یک از بایتهای داخل این سگمنت (از 12100 تا 12199)، شامل مقدار نوشته شده در متغیر lastack نیست(12000)، از مقدار نوشته شده در فیلد timestamp (4725) صرف نظر می‌شود. وقتی که گیرنده می‌خواهد یک ACK با مقدار 12200 در فیلد acknowledgment بفرستد، مقدار جدید فیلد acknowledgment یعنی 12200 را در متغیر lastack می‌نویسد و مقدار نوشته شده در متغیر tsrecent (4720) را در فیلد timestamp echo reply می‌نویسد. مقدار متغیر tsrecent تغییری نمی‌کند تا زمانی که گیرنده، سگمنتی را دریافت کند که حاوی شماره ترتیب 12200 باشد (سگمنت بعدی).

دقت نمایید همانطور که مثال ۱۲ نشان می‌دهد، RTT محاسبه شده برابر با اختلاف زمان بین لحظه ارسال سگمنت اول (4720) و لحظه دریافت سگمنت ACK (4732) است. این مقدار با مفهوم RTT کاملاً مطابقت دارد : یعنی اختلاف زمانی بین بسته‌ای که ارسال شده است و ACK ای که دریافت شده است. سگمنت سوم، تأییدیه‌ای بر دریافت سگمنت اول و دوم است.

* حفاظت در مقابل چرخش شماره‌های ترتیبی : PAWS

گزینه برچسب زمانی کاربرد دیگری نیز دارد و آن ***حفاظت در مقابل چرخش شماره‌های ترتیبی***[1] است. فیلد ***شماره ترتیب***[2] که در هدر TCP آمده است، به طول 32 بیت می‌باشد. گرچه می‌توان اعداد از 0 تا 2^{32}-1 را با آن نشان داد و ممکن است عدد بزرگی به نظر برسد، اما وقتی که سرعت انتقال دیتا، بالا باشد، سریعاً دنباله اعداد تا 2^{32}-1 طی می‌شوند و دوباره به ابتدای این دنباله، برمی‌گردیم. معنی ضمنی آن این است که اگر در لحظه‌ای از زمان، مقدار فیلد شماره ترتیب برابر با n باشد، ممکن است در خلال زمان رد و بدل شدن دیتا، مجدداً به این عدد n برگردیم. حال اگر سگمنت قدیمی که شماره ترتیب آن n بود، در جایی از شبکه، تکرار شود و از قضا زمانی به دست گیرنده برسد که می‌بایست یک سگمنت جدید به همان شماره n به دست گیرنده می‌رسید، گیرنده، سگمنت قدیمی n را دریافت می‌کند و

1- Protection Against Wrapped Sequence numbers : PAWS
2- Sequence number

www.Abiri.ir

اینطور تصور می‌کند که سگمنت جدید n ، که کمی بعد از راه رسیده است، جایی در شبکه، تکرار شده است و لذا آن را دور می‌اندازد.

یک راه حل برای این مشکل، افزایش طول فیلد Sequence number (شماره ترتیب) است اما برای افزایش طول این فیلد، باید طول فیلد window size (سایز پنجره) را نیز بزرگتر کرد که مستلزم تغییر فرمت هدر TCP و خیلی تغییرات دیگر است. ساده‌ترین راه حل این است که مقدار timestamp value (برچسب زمانی) را در فیلد identification (شناسه) سگمنت، وارد کنیم. به عبارت دیگر، برای تمایز سگمنتها از هم، از ترکیب توأم شماره ترتیب و برچسب زمانی استفاده کنیم. به عنوان مثال اگر سگمنتی با شماره ترتیب 12001 در زمان 400 فرستاده شده باشد و سگمنت دیگری به دلیل چرخش شماره ترتیبها، دوباره با همان شماره ترتیب 12001 ارسال شود، این سگمنت دومی نمی‌تواند در همان زمان 400 ارسال شده باشد و مثلاً زمان ارسال او 700 است. پس 400:12001 و 700:12001 قطعاً دو سگمنت مختلف (علی‌رغم یکسان بودن شماره ترتیب آنها) را نشان می‌دهند.

مهم : گزینه timestamp را می‌توان برای PAWS نیز به کار برد.

گزینه SACK و گزینه SACK-مجاز

همانطور که قبلاً بحث آن آمد، فیلد acknowledgment در هدر TCP به صورت تجمعی ، طراحی شده است و معنی آن این است که تأییدیه‌ای بر دریافت آخرین دنباله‌ای از بایتهاست که تا آن لحظه، به صورت پیوسته دریافت شده‌اند و لذا برای بایتهایی که خارج از نوبت دریافت شده‌اند، نمی‌توان توسط این فیلد، تأییدیه فرستاد. این موضوع شامل بایتهایی که تکرار شده‌اند نیز می‌شود. این وضعیت ممکن است یک اثر منفی بر عملکرد TCP بگذارد. اگر برخی از بسته‌ها گم شده یا دور انداخته شده باشند، فرستنده باید صبر کند تا تایمرش به پایان مهلت مقرر برسد و سپس تمام بسته‌هایی که برای آنها، تأییدیه دریافت نشده است را مجدداً بفرستد. به این ترتیب ممکن است گیرنده، بسته‌های تکراری را دریافت کند. برای بهتر کردن عملکرد، پیشنهاد شده است که از گزینه ***تأییدیه انتخابی***[1] استفاده گردد. گزینه SACK به فرستنده، ایده بهتری درباره اینکه کدام سگمنتها، واقعاً گم شده‌اند و کدام یک، خارج از نوبت دریافت شده‌اند، می‌دهد. پیشنهاد جدید حتی حاوی لیستی برای سگمنتهای تکراری است. به کمک گزینه SACK ، فرستنده می‌فهمد که فقط باید آن سگمنتهایی که واقعاً گم شده‌اند را مجدداً ارسال نماید. لیست سگمنتهای تکراری به فرستنده می‌فهماند که کدام یک از سگمنتها

1- Selective Acknowledgment : SACK

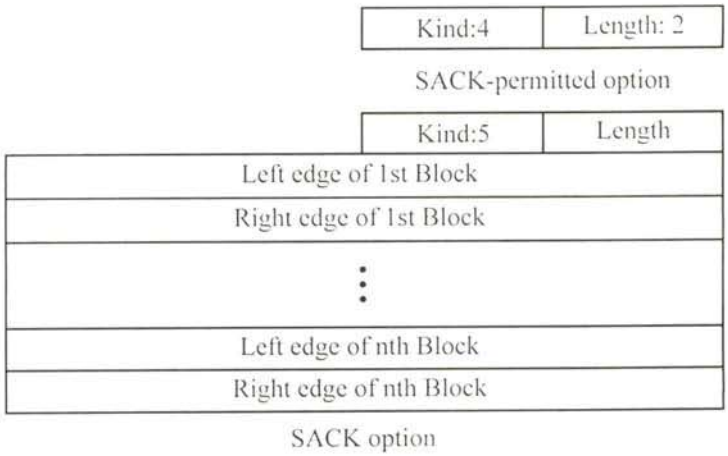
www.Abiri.ir

به دلیل کم بودن مهلت مقرر تایمر RTO ، مجدداً ارسال شده‌اند و لذا به صورت تکراری به دست گیرنده رسیده‌اند.

این پیشنهاد، دو گزینه را تعریف می‌کند : گزینه SACK- مجاز و گزینه SACK. فرمت این دو گزینه را در شکل ۱۲-۴۷ ملاحظه می‌نمایید.

* گزینه SACK- مجاز

گزینه ***SACK- مجاز***[1] به طول 2 بایت، فقط در زمان برقراری اتصال، به کار می‌رود. کامپیوتری که سگمنت SYN را می‌فرستد، این گزینه را نیز می‌تواند اضافه کند تا به سمت مقابل بفهماند که او می‌تواند از گزینه SACK پشتیبانی کند. اگر سمت دیگر که سگمنت حاوی SYN+ACK را می‌فرستد، گزینه SACK- مجاز را در سگمنت خود قرار دهد، او نیز اعلام می‌کند که می‌تواند از گزینه SACK پشتیبانی کند. اگر هر دو طرف، پشتیبانی از گزینه SACK را اعلام نمایند، می‌توانند در خلال نقل و انتقال دیتا، از گزینه SACK استفاده کنند.



شکل ۱۲-۴۷ : گزینه SACK و گزینه SACK- مجاز

* گزینه SACK

طول این گزینه، متغیر است و فقط زمانی می‌توان در خلال نقل و انتقال دیتا از آن استفاده کرد که دو طرف، ضمن برقراری اتصال، گزینه SACK– مجاز را برای یکدیگر، فرستاده

1- SACK-permitted

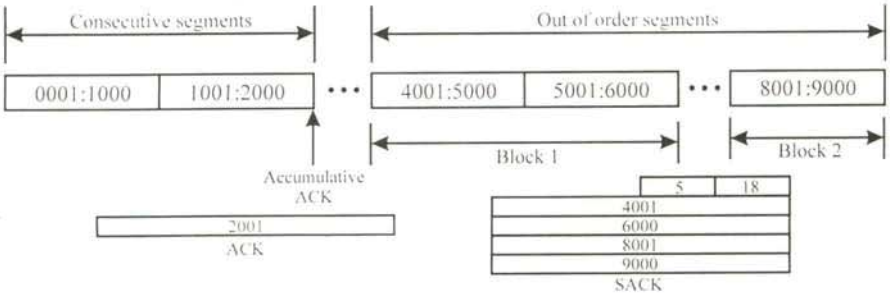
www.Abiri.ir

باشند. گزینه SACK حاوی لیست بلاکهایی است که خارج از نوبت، به دست گیرنده رسیده‌اند. هر بلاک، دو عدد 32 بیتی (چهار بایتی) را می‌پوشاند که نشان‌دهنده شماره ترتیب بایتهای شروع و پایان بلاکها می‌باشند. طی مثالهایی که به دنبال می‌آیند، نحوه استفاده از این گزینه را نشان خواهیم داد. در حال حاضر فقط به خاطر داشته باشید که سایز مجاز یک گزینه در هدر TCP می‌تواند حداکثر 40 بایت باشد. بنابراین گزینه SACK نمی‌تواند بیش از 4 بلاک را تعریف کند. فضای لازم برای 5 بلاک برابر است با : 42 = 2 + x4 (5x2) بایت که بزرگتر از حداکثر 40 بایت، فضای موجود برای گزینه‌ها در هدر TCP است. اگر در کنار SACK از سایر گزینه‌ها نیز استفاده کنیم، قطعاً کمتر از 4 بلاک را می‌توان در گزینه SACK تعریف کرد.

اولین بلاک گزینه SACK را می‌توان برای گزارش سگمنتهای تکراری به کار برد. فقط زمانی می‌توان از این ویژگی استفاده کرد که پیاده‌سازی، امکان آن را فراهم کرده باشد.

** مثال ۱۳

اجازه دهید ببینیم چطور می‌توان با استفاده از گزینه SACK ، لیست بلاکهایی که خارج از نوبت دریافت شده‌اند را نشان داد. شکل ۱۲-۴۸ نشان می‌دهد که یک طرف، پنج سگمنت دیتا را دریافت کرده است.



شکل ۱۲-۴۸ : مثال ۱۳

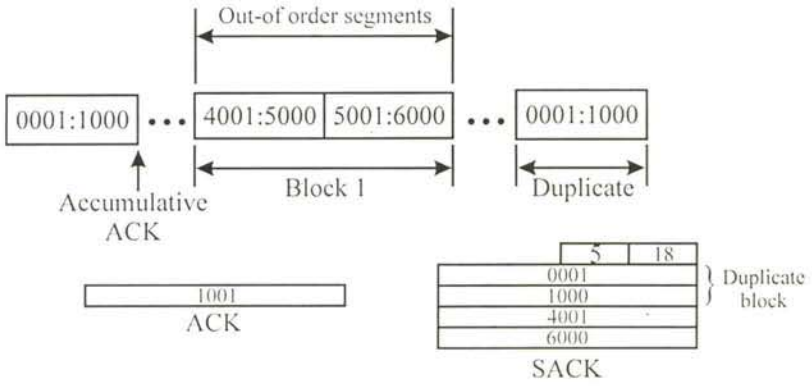
سگمنتهای اول و دوم، به ترتیب هستند. می‌توان یک ACK به عنوان تأییدی بر دریافت این دو سگمنت، ارسال کرد (ACK با شماره 2001). با این حال سگمنتهای سوم ، چهارم و پنجم خارج از نوبت هستند و یک فضای خالی از بایتها بین سگمت دوم (1001:2000) و سوم (4001:5000) و نیز یک فضای خالی از بایتها بین سگمنت چهارم (5001:6000) و پنجم (8001:9000) وجود دارد. یک ACK و SACK با هم می‌توانند

www.Abiri.ir

وضعیت را برای فرستنده، روشن کنند. مقدار ACK برابر با 2001 است و به این معنی است که لازم نیست فرستنده، نگران دریافت بایتهای تا 2000 باشد. SACK حاوی دو بلاک است. اولین بلاک اعلام می‌کند که بایتهای از 4001 تا 6000 ، خارج از نوبت از راه رسیده‌اند. بلاک دوم نشان می‌دهد که بایتهای 8001 تا 9000 نیز خارج از نوبت دریافت شده‌اند. معنی ضمنی آن این است که بایتهای 2001 تا 4000 و بایتهای 6001 تا 8000 گم شده‌اند و یا دور انداخته شده‌اند. فرستنده فقط باید این بایتها را ارسال کند.

**** مثال ۱۴**

مثال شکل ۱۲-۴۹ نشان می‌دهد که چطور می‌توان با ترکیب ACK و SACK ، یک سگمنت تکراری را آشکار کرد. در این مثال، سگمنتهایی داریم که خارج از نوبت دریافت شده‌اند (Block 1) و نیز سگمنتی داریم که تکراری است(Duplicate). SACK برای نشان دادن سگمنتی که تکرار شده است، از اولین بلاک خود استفاده می‌کند و در سایر بلاکها، سگمنتهایی که خارج از نوبت دریافت شده‌اند را معرفی می‌کند. دقت نمایید که فقط از اولین بلاک گزینه SACK می‌توان برای معرفی سگمنتهای تکراری استفاده کرد. سؤالی که طبیعتاً مطرح می‌شود این است که وقتی فرستنده، ACK و SACK را دریافت می‌کند، از کجا بداند که اولین بلاک موجود در گزینه SACK ، مربوط به سگمنتهایی است که تکرار شده‌اند و یا مربوط به سگمنتهایی است که خارج از نوبت دریافت شده‌اند (این را با مثال قبل مقایسه کنید). پاسخ این است که دریافت بایتهایی که توسط اولین بلاک معرفی شده‌اند (از 0001 تا 1000)، هم اکنون توسط مقداری که ACK نشان می‌دهد (1001) تأیید شده است و این دلیلی بر تکراری بودن این بایتهاست.

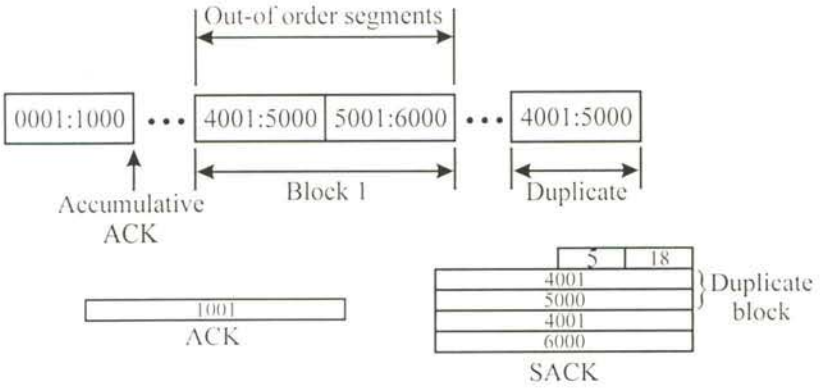


شکل ۱۲-۴۹ : مثال ۱۴

www.Abiri.ir

**** مثال ۱۵**

مثال شکل ۱۲-۵۰ نشان می‌دهد که اگر یکی از سگمنتهایی که خارج از نوبت دریافت شده است، از قضا، تکرار هم شده باشد، چه اتفاقی می‌افتد. در این مثال، سگمنتی که شماره ترتیب بایتهای آن از 4001 تا 5000 است (سگمنت 4001:5000)، هم خارج از نوبت دریافت شده است و هم تکرار شده است. گزینه SACK ، ابتدا این سگمنت تکراری و سپس سگمنتهایی که خارج از نوبت دریافت شده‌اند را معرفی می‌کند. گرچه در این حالت، دریافت دیتای تکراری، هنوز توسط ACK ، تأیید نشده است اما از آنجا که جزئی از محدوده سگمنتهای خارج از نوبت است (4001:5000 جزئی از 4001:6000 است)، فرستنده در می‌یابد که این، یک سگمنت تکراری است.



شکل ۱۲-۵۰ : مثال ۱۵

۱۲-۱۱ نرم‌افزار TCP

TCP یک پروتوکل بسیار پیچیده است. TCP یک پروتوکل اتصالگرا برای تحویل دنباله‌ای از بایتها به گیرنده است و حاوی یک ***دیاگرام گذر از حالتهای***[1] مفصل است. TCP از کنترل فلو و کنترل خطا استفاده می‌کند. TCP آنقدر پیچیده است که کد نرم‌افزاری آن از دهها هزار سطر تشکیل شده است.

در این بخش، یک ساختار نرم‌افزاری ساده‌ای را ارائه می‌نماییم که اصل مطلب را نشان می‌دهد. می‌خواهیم اصل کار TCP که توسط دیاگرام گذر از حالتهای آن، نشان داده می‌شود را شبیه‌سازی کنیم.

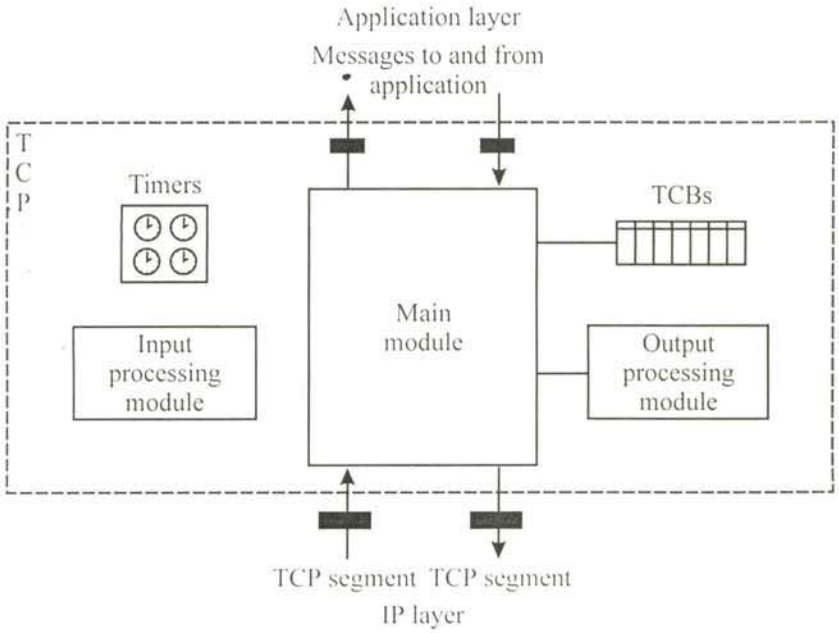
1- State transition diagram

www.Abiri.ir

بسته نرم‌افزاری ما شامل یک سری جدول به نام ***بلاکهای کنترل انتقال***[1] ، یک سری تایمر و سه ماجول نرم‌افزاری است: یک ماجول اصلی، یک ماجول پردازش‌کننده ورودی و یک ماجول پردازش‌کننده خروجی. شکل ۱۲-۵۱ این پنج جزء و رابطه بین آنها را نشان می‌دهد.

بلاکهای کنترل انتقال : TCB

TCP یک پروتوکل اتصالگرای لایه انتقال است. یک اتصال ممکن است به مدت زیادی، باز بماند. TCP برای کنترل اتصال، از بلاک کنترل انتقال(TCB) برای نگهداری اطلاعات مربوط به آن اتصال، استفاده می‌کند. از آنجا که در هر لحظه ممکن است چندین اتصال باز داشته باشیم، TCP ، آرایه‌ای از TCB ها را به صورت یک جدول ، نگه می‌دارد. معمولاً به این جدول، TCB گفته می‌شود (شکل ۱۲-۵۲ را ببینید).



شکل ۱۲-۵۱ : شمای کلی نرم‌افزار TCP

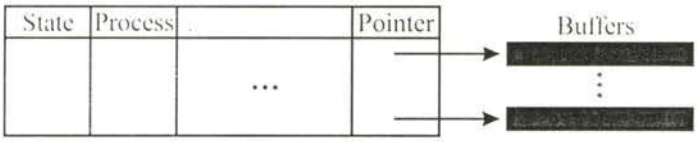
هر TCB می‌تواند شامل فیلدهای متعددی باشد که در اینجا، فقط مهمترین آنها را معرفی می‌کنیم:

1- Transmission Control Block : TCB

www.Abiri.ir

State : حالت. این فیلد، حالت فعلی اتصال را مطابق با دیاگرام گذر از حالتها، نشان می‌دهد.

Process : برنامه . این فیلد، برنامه‌ای (سرور یا سرویس‌گیرنده) که در حال حاضر در این کامپیوتر از این اتصال استفاده می‌کند را مشخص می‌نماید.



شکل ۱۲-۵۲ : TCB ها

Local IP address : آدرس IP محلی. این فیلد، آدرس IP کامپیوتر محلی (همین کامپیوتر) را نشان می‌دهد.

Local port number : شماره پورت محلی . این فیلد، شماره پورت محلی (متعلق به برنامه‌ای روی همین کامپیوتر) را نشان می‌دهد.

Remote IP address : آدرس IP دوردست. این فیلد، آدرس IP کامپیوتر دوردست (کامپیوتر سمت مقابل) را نشان می‌دهد.

Remote port number : شماره پورت دوردست. این فیلد، شماره پورت دوردست (متعلق به برنامه‌ای روی کامپیوتر سمت مقابل) را نشان می‌دهد.

Interface : رابط. این فیلد، شماره اینترفیس روی همین کامپیوتر را نشان می‌دهد.

Local window : پنجره محلی. این فیلد که خود می‌تواند شامل چندین زیرفیلد باشد، اطلاعاتی درباره پنجره کنترل فلوی مورد استفاده در TCP همین طرف از اتصال را نگه می‌دارد.

Remote window : پنجره دوردست. این فیلد که خود می‌تواند شامل چندین زیرفیلد باشد، اطلاعاتی درباره پنجره کنترل فلوی مورد استفاده در TCP سمت دیگر اتصال را نگه می‌دارد.

Sending sequence number : شماره ترتیب ارسال. این فیلد، شماره ترتیب سگمنتهای ارسالی را در خود نگه می‌دارد.

Receiving sequence number : شماره ترتیب دریافت. این فیلد، شماره ترتیب سگمنتهای دریافتی را در خود نگه می‌دارد.

Sending ACK number : شماره ACK ارسالی. این فیلد، شماره به کار رفته در فیلد acknowledgment در ارسال ACK ها را در خود نگه می‌دارد.

www.Abiri.ir

Round Trip Time : زمان رفت و برگشت. ممکن است چندین زیرفیلد، اطلاعات مربوط به RTT را در خود نگه دارند.

Time out value : مقدار مهلت مقرر. ممکن است چندین زیرفیلد، اطلاعات مربوط به مهلت مقرر انواع تایمرها مانند تایمر ارسال مجدد، تایمر اصرار بر ارسال، تایمر زنده نگه دارِ اتصال و غیره را در خود نگه دارند.

Buffer size : سایز بافر. این فیلد، سایز بافر نگهداری دیتای دریافتی را در TCP همین طرف از اتصال، نشان می‌دهد.

Buffer pointer : نشانه‌گر بافر. این نشانه‌گر به بافری اشاره می‌کند که دیتاهای دریافتی را در خود نگه می‌دارد تا توسط برنامه کاربردی مربوطه، خوانده شود.

تایمرها

قبلاً درباره تایمرهایی که TCP برای دنبال کردن عملیاتش از آنها استفاده می‌کند، صحبت کردیم.

ماجول اصلی

پیغامی از یک برنامه کاربردی، سیگنالی از تایمری که به پایان مهلت مقرر خود رسیده است و یا دریافت یک سگمنت TCP ، ***ماجول اصلی***[1] را فراخوانی می‌کند. این یک ماجول بسیار پیچیده است زیرا عملی که باید انجام شود، بستگی به حالت فعلی TCP دارد. برای پیاده‌سازی دیاگرام گذر از حالتهای TCP، از روشهای مختلفی استفاده شده است که در این میان می‌توان از بکارگیری یک پروسه، برای هر حالت و نیز استفاده از یک جدول (آرایه دو بعدی) نام برد. برای ساده نگه داشتن کار خودمان، **مورد به مورد**[2]، به حالتها رسیدگی می‌کنیم. تعداد 11 حالت مختلف داریم، بنابراین در برنامه خود از 11 تا case استفاده می‌کنیم. پیاده‌سازی هر case بر اساس حالت مربوطه در دیاگرام گذر از حالتها است. حالت ***اتصال- برقرار***[3] به توضیح بیشتری نیاز دارد. وقتی که TCP در این حالت است و یک سگمنت دیتا یا تأییدیه از راه می‌رسد، ماجول دیگری به نام ماجول پردازش کننده ورودی، فراخوانی می‌شود تا به این سگمنت دریافتی، رسیدگی کند. همچنین وقتی که TCP در این حالت است و یک فرمان send data (دیتا بفرست) توسط یک برنامه، صادر می‌شود، ماجول دیگری به نام ماجول پردازش کننده خروجی فراخوانی می‌شود تا به این فرمان، رسیدگی کند.

1- main module
2- case
3- ESTABLISHED

www.Abiri.ir

* جریان رویدادهای ماجول اصلی

ورودی : دریافت یک سگمنت TCP، دریافت یک پیغام از یک برنامه کاربردی یا دریافت سیگنالی از تایمری که به پایان مهلت مقرر رسیده است.

۱- داخل جدول TCB را جستجو می‌کند.
۲- اگر المان مربوطه را در جدول TCB پیدا نکند :
  ۱- یک المان با حالت CLOSED در TCB تولید می‌کند.
۳- به state (حالت) المان مربوطه در TCB نگاه می‌کند.
۴- case (state) : بر اساس اینکه مقدار state،کدامیک از مقادیر زیر باشد، عملیات مربوطه را انجام می‌دهد (به خطوطی در شکل ۱۲-۱۳ نگاه کنید که از هر حالت، خارج می‌شوند).

: CLOSED

۱- اگر برنامه کاربردی فرمان passive open (به صورت پسیو، باز شو) را صادر کرده است:
  ۱- مقدار حالت را به LISTEN تغییر می‌دهد.
۲- اگر برنامه کاربردی فرمان active open (به صورت اکتیو، باز شو) را صادر کرده است:
  ۱- یک سگمنت SYN را ارسال می‌کند
  ۲- مقدار حالت را به SYN-SENT تغییر می‌دهد.
۳- اگر سگمنتی دریافت شده است:
  ۱- یک سگمنت حاوی RST (بیت RST در آن، 1 است) ارسال می‌کند.
۴- اگر هرگونه پیغام دیگری دریافت شده است:
  ۱- پیغام خطایی را صادر می‌نماید.
۵- برمی‌گردد.

: LISTEN

۱- اگر برنامه کاربردی، فرمان send data (دیتا بفرست) را صادر کرده است:
  ۱- یک سگمنت SYN می‌فرستد.
  ۲- مقدار حالت را به SYN-SENT تغییر می‌دهد.
۲- اگر سگمنت SYN (بیت SYN در آن، 1 است) دریافت شده باشد:
  ۱- یک سگمنت SYN+ACK (بیتهای SYN و ACK در آن، 1 هستند) را ارسال می‌کند.
  ۲- مقدار حالت را به SYN-RCVD تغییر می‌دهد.

www.Abiri.ir

۳- اگر هر پیغام دیگری دریافت شده است:
۱- پیغام خطایی را صادر می‌نماید.
۴- برمی‌گردد.

: SYN-SENT
۱- اگر مهلت مقرر برقراری اتصال، سپری شده است:
۱- حالت را به CLOSED تغییر می‌دهد.
۲- اگر سگمنت SYN دریافت شده است:
۱- یک سگمنت SYN+ACK را ارسال می‌کند.
۲- مقدار حالت را به SYN-RCVD تغییر می‌دهد.
۳- اگر سگمنت SYN+ACK دریافت شده است:
۱- یک سگمنت ACK را ارسال می‌کند.
۲- مقدار حالت را به ESTABLISHED تغییر می‌دهد.
۴- اگر هر پیغام یا سگمنت دیگری دریافت شده باشد:
۱- پیغام خطایی را صادر می‌نماید.
۵- برمی‌گردد.

: SYN-RCVD
۱- اگر سگمنت ACK دریافت شده است:
۱- مقدار حالت را به ESTABLISHED تغییر می‌دهد.
۲- اگر مهلت مقرر برای برقراری اتصال، سپری شده است:
۱- یک سگمنت RST ارسال می‌کند.
۲- مقدار حالت را به CLOSED تغییر می‌دهد.
۳- اگر برنامه کاربردی، فرمان close (بسته شو)را صادر کرده است:
۱- یک سگمنت FYN (بیت FIN در آن، 1 است) ارسال می‌کند.
۲- مقدار حالت را به FIN-WAIT-1 تغییر می‌دهد.
۴- اگر سگمنت RST دریافت شده است:
۱- مقدار حالت را به LISTEN تغییر می‌دهد.
۵- اگر هر پیغام یا سگمنت دیگری دریافت شده است:
۱- پیغام خطایی را صادر می‌نماید.
۶- برمی‌گردد.

www.Abiri.ir

: ESTABLISHED

۱- اگر سگمنت FIN دریافت شده است:

۱- یک سگمنت ACK می‌فرستد.

۲- مقدار حالت را به CLOSE-WAIT تغییر می‌دهد.

۲- اگر برنامه کاربردی فرمان close را صادر کرده است:

۱- یک سگمنت FIN را ارسال می‌کند.

۲- مقدار حالت را به FIN-WAIT-1 تغییر می‌دهد.

۳- اگر سگمنت RST یا SYN دریافت شده است:

۱- پیغام خطایی را صادر می‌کند.

۴- اگر سگمنت دیتا یا سگمنت ACK دریافت شده باشد:

۱- ماجول ورودی را صدا می‌زند.

۵- اگر برنامه کاربردی، فرمان send (بفرست) را صادر کرده باشد:

۱- ماجول خروجی را صدا می‌زند.

۶- برمی‌گردد.

: FIN-WAIT-1

۱- اگر سگمنت FIN دریافت شده است:

۱- یک سگمنت ACK می‌فرستد.

۲- مقدار حالت را به CLOSING تغییر می‌دهد.

۲- اگر سگمنت FIN+ACK دریافت شده است:

۱- یک سگمنت ACK می‌فرستد.

۲- مقدار حالت را به TIME-WAIT تغییر می‌دهد.

۳- اگر سگمنت ACK دریافت شده است:

۱- مقدار حالت را به FIN-WAIT-2 تغییر می‌دهد.

۴- اگر هر پیغام یا سگمنت دیگری دریافت شده است:

۱- پیغام خطایی را صادر می‌نماید.

۵- برمی‌گردد.

: FIN-WAIT-2

۱- اگر سگمنت FIN دریافت شده است:

۱- یک سگمنت ACK می‌فرستد.

۲- مقدار حالت را به TIME-WAIT تغییر می‌دهد.

۲- برمی‌گردد.

www.Abiri.ir

: CLOSING

۱- اگر سگمنت ACK دریافت شده است:

۱- مقدار حالت را به TIME-WAIT تغییر می‌دهد.

۲- اگر سگمنت یا پیغام دیگری دریافت شده است:

۱- پیغام خطایی را صادر می‌نماید.

۳- برمی‌گردد.

: TIME-WAIT

۱- اگر مهلت مقرر این تایمر، سپری شده است:

۱- مقدار حالت را به CLOSED تغییر می‌دهد.

۲- اگر هر سگمنت یا پیغام دیگری دریافت شده است:

۱- پیغام خطایی را صادر می‌نماید.

۳- برمی‌گردد.

: CLOSE-WAIT

۱- اگر برنامه کاربردی، فرمان close را صادر کرده است:

۱- یک سگمنت FIN ارسال می‌کند.

۲- مقدار حالت را به LAST-ACK تغییر می‌دهد.

۲- اگر هر سگمنت یا پیغام دیگری دریافت شده است:

۱- پیغام خطایی را صادر می‌نماید.

۳- برمی‌گردد.

: LAST-ACK

۱- اگر سگمنت ACK دریافت شده است:

۱- مقدار حالت را به CLOSED تغییر می‌دهد.

۲- اگر هر سگمنت یا پیغام دیگری دریافت شده است :

۱- پیغام خطایی را صادر می‌کند.

۳- برمی‌گردد.

ماجول پردازش‌کننده ورودی

در طرح ما، وقتی که TCP در حالت ESTABLISHED است، همه کارهای لازم برای پردازش دیتا یا تأییدیه دریافتی، توسط ماجول پردازش کننده ورودی انجام می‌شوند. اگر لازم

www.Abiri.ir

باشد این ماجول، یک ACK می‌فرستد، به تغییر مقدار سایز پنجره رسیدگی می‌کند، آشکارسازی خطا را انجام می‌دهد و غیره. جزئیات این ماجول در این کتاب نیامده است.

ماجول پردازش‌کننده خروجی

در طرح ما، وقتی که TCP در حالت ESTABLISHED است، همه کارهای لازم برای ارسال دیتایی که برنامه کاربردی، درخواست ارسال آن را داده است، توسط ماجول پردازش‌کننده خروجی انجام می‌شوند. تایمرهای ارسال مجدد، اصرار بر ارسال و غیره توسط این ماجول، راه‌اندازی و چک می‌شوند. یکی از راه‌های پیاده‌سازی این ماجول، استفاده از یک دیاگرام گذر از حالتهای ساده برای رسیدگی به وضعیتهای مختلف خروجی است. جزئیات این ماجول، در این کتاب نیامده است.

۱۲-۱۲ عبارات کلیدی

افزایش جمع شونده (Additive increasing)
نقطه انسداد (Choke point)
راه حل کلارک (Clark's solution)
ازدحام (Congestion)
امتناع از بروز ازدحام (Congestion avoidance)
کنترل ازدحام (Congestion control)
آشکارسازی ازدحام (Congestion detection)
اتصال (Connection)
لغو اتصال (Connection abortion)
برقراری اتصال (Connection establishment)
ریست کردن اتصال (Connection resetting)
خاتمه اتصال (Connection termination)
کوکی (Cookie)
شماره ترتیب اولیه (initial sequence number)
الگوریتم کارن (Karn's algorithm)
تایمر زنده نگه دار اتصال (Keepalive timer)
ماکزیمم سایز سگمنت (Maximim Segment Size : MSS)
گزینه ماکزیمم سایز سگمنت (Maximum Segment Size option)
کاهش مضربی (multiplicative decrease)

www.Abiri.ir

الگوریتم ناگل (Nagle's algorithm)
گزینه NOP
تایمر اصرار بر ارسال (persistence timer)
بردوش‌گذاری (piggybacking)
حفاظت در مقابل چرخش شماره‌های ترتیبی
(Protection Against Wrapped Sequence numbers : PAWS)
مهلت مقرر ارسال مجدد (Retransmission Time-Out : RTO)
تایمر ارسال مجدد (Retransmission Timer)
زمان رفت و برگشت (Round Trip Time : RTT)
گزینه SACK (SACK option)
گزینه SACK- مجاز (SACK-permitted option)
سگمنت (Segment)
شماره ترتیب (Sequence number)
علامت پنجره نامعقول (Silly window syndrome)
بسته شدن همزمان (simultaneous close)
باز شدن همزمان (simultaneous open)
پنجره لغزان (sliding window)
پروتوکل پنجره لغزان (sliding window protocol)
شروع آهسته (slow start)
دیاگرام گذر از حالتها (state transition diagram)
حمله سیل‌آسای SYNها (SYN flooding attack)
تبادل سه‌گانه (three-way handshaking)
نرخ عبور (throughput)
گزینه برچسب زمانی (timestamp option)
تایمر TIME-WAIT (TIME-WAIT timer)
پروتوکل کنترل انتقال (Transmission Control Protocol : TCP)
ضریب بزرگنمایی پنجره (window scale factor)

۱۲-۱۳ خلاصه

• پروتوکل TCP یکی از پروتوکلهای لایه انتقال در پشته پروتوکل TCP/IP است. TCP یک سرویس انتقال دیتای برنامه به برنامه، اتصالگرا و کاملاً دو طرفه را ارائه می‌دهد. واحد

www.Abiri.ir

دیتایی که توسط TCP بین دو برنامه جابجا می‌شود، سگمنت نامیده می‌شود که حاوی یک هدر 20 تا 60 بایتی است و به دنبال آن، دیتای برنامه کاربردی می‌آید.

- یک اتصال TCP از ابتدا تا پایان، سه فاز را در برمی‌گیرد: فاز برقراری اتصال، فاز نقل و انتقال دیتا و فاز خاتمه اتصال.
- فاز برقراری اتصال، با دست به دست کردن سه سگمنت و فاز خاتمه اتصال، با دست به دست کردن سه یا چهار سگمنت، انجام می‌شوند.
- نرم‌افزار TCP به صورت یک ماشین با حالتهای محدود، پیاده‌سازی می‌شود.
- TCP برای ممانعت از لبریز شدن گیرنده از دیتا، از یک مکانیزم کنترل فلو که توسط پنجره لغزان پیاده‌سازی می‌شود، استفاده می‌کند.
- سایز پنجره TCP فرستنده توسط سایز پنجره اعلام شده توسط گیرنده (rwnd) و سایز پنجره ازدحام (cwnd) ، هر کدام که کوچکتر باشد، تعیین می‌شود. گیرنده می‌تواند سایز پنجره rwnd را کم یا زیاد نماید.
- TCP بایتهای دیتایی که در هر اتصال جابجا می‌شوند را شماره‌گذاری می‌کند. شماره‌گذاری با یک عدد تصادفی اولیه آغاز می‌شود.
- TCP برای فراهم کردن یک سرویس قابل اطمینان، از کنترل خطا استفاده می‌کند. checksum ، ارسال ACK و به کارگیری تایمرها روشهایی هستند که در فرایند کنترل خطا مورد استفاده قرار می‌گیرند. سگمنتهای مخدوش شده و گم شده، مجدداً ارسال می‌شوند و سگمنتهای تکراری، دور انداخته می‌شوند. ممکن است دیتا خارج از نوبت به گیرنده برسد. TCP آنها را ذخیره می‌کند و این تضمین را می‌دهد که هیچگاه دنباله ناقصی از دیتاها را به لایه کاربرد ندهد.
- در پیاده‌سازی‌های جدید، اگر تایمر ارسال مجدد به پایان مهلت مقرر برسد، یا سه ACK تکراری از راه برسند، ارسال مجدد صورت می‌گیرد.
- TCP برای امتناع از وقوع ازدحام و نیز آشکارسازی ازدحام در شبکه، از کنترل ازدحام استفاده می‌کند. استراتژی‌های « آشکارسازی ازدحام : کاهش مضربی » ، « جلوگیری از وقوع ازدحام : افزایش خطی » و بالاخره « شروع آهسته : افزایش نمایی » برای کنترل ازدحام به کار می‌روند.
- در الگوریتم « شروع آهسته : افزایش نمایی » ، سایز پنجره ازدحام به صورت نمایی افزایش می‌یابد تا جایی که به یک حد بالا می‌رسد.
- در الگوریتم « جلوگیری از وقوع ازدحام : افزایش خطی » ، سایز پنجره ازدحام به صورت خطی افزایش می‌یابد تا زمانی که ازدحامی آشکار گردد.

www.Abiri.ir

• اکثر پیاده‌سازی‌ها به طرق مختلفی در مقابل آشکارسازی ازدحام، عکس‌العمل نشان می‌دهند: اگر پایان مهلت مقرر یک تایمر، معرف بروز ازدحام باشد، یک فاز جدید « شروع آهسته : افزایش نمایی » ، آغاز می‌گردد. اگر دریافت سه ACK تکراری معرف بروز ازدحام باشد، یک فاز جدید « جلوگیری از وقوع ازدحام : افزایش خطی » ، آغاز می‌گردد.

• TCP در کار خود از چهار تایمر (ارسال مجدد، اصرار بر ارسال، زنده نگه دار اتصال و انتظار قبل از قطع کامل) استفاده می‌کند.

• در هر لحظه در TCP ، فقط اندازه‌گیری یک RTT می‌تواند در دست محاسبه باشد.

• TCP در محاسبه RTO جدید، RTT سگمنتی که مجدداً ارسال شده است را در نظر نمی‌گیرد.

• TCP از گزینه‌ها برای ارائه سرویسهای بیشتر، استفاده می‌کند.

• گزینه EOP برای ترازبندی به کار می‌رود و نشان می‌دهد که گزینه دیگری بعد از آن در هدر نیست. فقط یکبار می‌توان از EOP استفاده کرد.

• گزینه NOP برای ترازبندی به کار می‌رود. می‌توان از چندین بایت NOP ، در ابتدا و یا مابین گزینه‌ها استفاده کرد.

• گزینه ماکزیمم سایز سگمنت در فاز برقراری اتصال به کار می‌رود و بلندترین سایز دیتای موجود در یک سگمنت را تعیین می‌کند. مقدار MSS در فاز برقراری اتصال، تعیین می‌شود و در فاز رد و بدل شدن دیتا، مقدار ثابتی باقی می‌ماند.

• ضریب بزرگنمایی پنجره، مقدار سایز پنجره را افزایش می‌دهد.

• گزینه برچسب زمانی، مدت زمانی که طول می‌کشد تا دیتا از فرستنده به گیرنده برسد را نشان می‌دهد. یکی از کاربردهای آن، محاسبه زمان رفت و برگشت (RTT) دیتا بین فرستنده و گیرنده است. کاربرد دیگر آن در PAWS است.

• پیاده‌سازی‌های جدید TCP ، از دو گزینه دیگر نیز استفاده می‌کنند: گزینه SACK- مجاز و گزینه SACK. این دو گزینه به گیرنده کمک می‌کنند تا به صورت انتخابی، تأییدیه‌ای بر دریافت سگمنتها ارسال نماید.

• بلاکهای کنترل انتقال (TCB) اطلاعاتی را درباره هر اتصال TCP نگه می‌دارند.

• یک بسته نرم‌افزاری TCP می‌تواند شامل TCBها ، تایمرها، یک ماجول اصلی، یک ماجول پردازش کننده ورودی و یک ماجول پردازش کننده خروجی باشد.

www.Abiri.ir

۱۲-۱۴ مجموعه تمرین‌ها

تمرین‌ها

۱- هدر TCP را با هدر UDP مقایسه کنید. لیست فیلدهایی را نشان دهید که در هدر TCP وجود دارند اما هدر UDP فاقد آنهاست و علت آن را بیان نمایید.

۲- یک بسته IP ، یک سگمنت TCP را به آدرس مقصد 130.14.16.17/16 حمل می‌کند. آدرس IP مقصد نوشته شده در هدر بسته IP ، حین انتقال، دچار خطا می‌شود و لذا این بسته اشتباهاً به مقصدی به آدرس 130.14.16.19/16 می‌رسد. TCP در مقابل دریافت چنین سگمنتی، چه عکس‌العملی نشان می‌دهد؟

۳- همانطور که در فصل نهم دیدیم، یک پیغام خطای ICMP ، مربوط به غیر قابل دسترس بودن پورت مقصد است. چطور TCP می‌تواند خطای موجود در پورت مقصد را آشکار کند؟

۴- UDP یک پروتوکل ***پیغام‌گرا***[1] است. TCP یک پروتوکل ***بایت‌گرا***[2] می‌باشد. اگر برنامه کاربردی بخواهد نگرانی درباره طول پیغامهایش نداشته باشد، از UDP باید استفاده کند یا از TCP ؟

۵- حداکثر طول هدر TCP چقدر است؟ حداقل طول هدر TCP چقدر است؟

۶- اگر مقدار فیلد HLEN برابر با 0111 باشد، چند بایت گزینه در هدر سگمنت TCP وجود دارد؟

۷- محتوای هدر TCP سگمنتی که پیغام را از یک سرویس‌گیرنده FTP برای یک سرور FTP می‌برد را نشان دهید. فیلد checksum را با 0 پر کنید. از یک پورت موقتی مناسب و یک پورت کاملاً شناخته شده استفاده کنید. طول دیتا برابر با 40 بایت است.

۸- درباره سگمنت TCP که در آن مقدار فیلد کنترل، یکی از مقادیر زیر است، چه می‌توان گفت:

الف- 000000

ب- 000001

پ- 010001

ت – 000100

ث – 000010

ج- 010010

۹- آنچه که در ذیل می‌آید، نمایش محتوای هدر TCP بر حسب هگزادسیمال است:

05320017 00000001 00000000 500207FF 00000000

1- message-oriented

2- byte-oriented

www.Abiri.ir

الف – شماره پورت مبدأ چیست؟

ب- شماره پورت مقصد چیست؟

پ- شماره ترتیب چند است؟

ت- شماره تأییدیه چند است؟

ث- طول هدر چقدر است؟

ج- نوع سگمنت چیست؟

چ- سایز پنجره چیست؟

۱۰- فیلد کنترل در یک سگمنت TCP به طول 6 بیت است. می‌توانیم 64 ترکیب مختلف از بیتها را داشته باشیم. برخی از ترکیبات معتبر را بنویسید.

۱۱- اکثر سیستم‌ها برای شروع شماره ترتیب با یک عدد تصادفی، یک شمارنده 32 بیتی را به هنگام راه‌اندازی سیستم، روی عدد 1 ست می‌کنند و سپس بعد از گذشت هر نیم ثانیه، عدد 64000 را به آن اضافه می‌کنند. چه مدت زمانی طول می‌کشد تا این شمارنده، یک دور بزند و دوباره از اول شروع کند؟

۱۲- با استفاده از یک دیاگرام گذر از حالتها، نشان دهید که اگر سرور به جای سرویس‌گیرنده، فرمان active close (به صورت اکتیو، بسته شو) را صادر نماید، چه اتفاقی می‌افتد؟

۱۳- در یک اتصال TCP، مقدار اولیه شماره ترتیب در سمت سرویس گیرنده برابر با 2171 است. سرویس‌گیرنده، اتصال را باز می‌کند، یک سگمنت که فقط حاوی 1000 بایت دیتا است را می‌فرستد و سپس اتصال را می‌بندد. در هر یک از سگمنتهای زیر که توسط سرویس‌گیرنده ارسال می‌شوند، مقدار شماره ترتیب، چقدر است؟

الف- سگمنت SYN

ب- سگمنت دیتا

ج- سگمنت FIN

۱۴- در یک اتصال، مقدار cwnd برابر با 3000 و مقدار rwnd برابر با 5000 است. کامپیوتر به تعداد 2000 بایت دیتا ارسال کرده است که هنوز تأییدیه‌ای بر دریافت آنها به دستش نرسیده است. چند بایت بیشتر را می‌توان ارسال کرد؟

۱۵- TCP با استفاده از یک عدد ترتیبی اولیه (ISN) برابر با 14534 ، اتصالی را باز می‌کند. سمت مقابل، اتصال را با ISN برابر با 21732 باز می‌کند. سه سگمنت TCP که در فاز برقراری اتصال، دست به دست می‌شوند را نشان دهید.

۱۶- با استفاده از مثال ۱۵، اگر یک طرف پیغام « سلام مشتری عزیز » را بفرستد و سمت دیگر با سگمنتی حاوی « سلام فروشنده عزیز » پاسخ دهد، محتوای سگمنتهایی که این پیغامها را حمل می‌کنند، را نشان دهید.

www.Abiri.ir

۱۷- با استفاده از تمرین‌های ۱۵ و ۱۶ محتوای سگمنتهایی که در فاز خاتمه اتصال، جابجا می‌شوند را نشان دهید.

۱۸- سرویس‌گیرنده‌ای از TCP برای ارسال دیتا به سمت سرور مورد نظرش استفاده می‌کند. حجم دیتا، 16 بایت است. در سطح TCP ، راندمان این انتقال (نسبت بایتهای مفید به کل بایتها) را محاسبه نمایید. راندمان انتقال را در سطح IP محاسبه نمایید. فرض کنید که هدر IP حاوی هیچ گزینه‌ای نیست. راندمان انتقال را در سطح لایه دیتالینک محاسبه نمایید. فرض کنید که هیچ گزینه‌ای در هدر IP وجود ندارد و در لایه دیتالینک از پروتوکل اترنت استفاده می‌شود.

۱۹- TCP با سرعت 1 مگا بایت در ثانیه، دیتا می‌فرستد. اگر مقدار اولیه شماره ترتیب برابر با 7000 باشد، چه مدت طول می‌کشد تا شماره ترتیب، به صفر برسد؟

۲۰- یک اتصال TCP از پنجره‌ای به سایز 10000 بایت استفاده می‌کند و مقدار قبلی شماره تأییدیه برابر با 22001 می‌باشد. او سگمنتی با شماره تأییدیه برابر با 24001 و سایز پنجره اعلام شده برابر با 12000 دریافت می‌کند. دیاگرامی بکشید و در آن وضعیت پنجره را قبل و بعد از دریافت این سگمنت تأییدیه نشان دهید.

۲۱- پنجره‌ای حاوی شماره بایتهای از 2001 تا 5000 است. شماره بایت بعدی که باید ارسال شود برابر با 3001 است. شکلی بکشید و در آن، وضعیت پنجره را بعد از دو رویداد زیر، نشان دهید.

الف- یک سگمنت ACK با شماره تأییدیه برابر با 2500 و سایز پنجره اعلام شده برابر با 4000 دریافت می‌شود.

ب- سگمنتی حاوی 1000 بایت، ارسال می‌شود.

۲۲- یک اتصال TCP در حالت ESTABLISHED است. رویدادهای زیر، یکی بعد از دیگری رخ می‌دهند:

الف – یک سگمنت FIN دریافت می‌شود.

ب- برنامه کاربردی یک فرمان close صادر می‌نماید.

حالت جدید اتصال، بعد از هر رویداد، چه خواهد بود؟ چه خروجی تولید خواهد شد؟

۲۳- یک اتصال TCP در حال ESTABLISHED است. رویدادهای زیر یکی بعد از دیگری رخ می‌دهند:

الف – برنامه کاربردی یک فرمان close صادر می‌نماید.

ب- یک سگمنت ACK دریافت می‌شود.

www.Abiri.ir

۲۴- کامپیوتر دیتایی برای ارسال ندارد. سگمنتهای زیر در زمانهای داده شده (با فرمت میلی‌ثانیه : ثانیه : دقیقه : ساعت بعد از نیمه شب) دریافت می‌شوند. تأییدیه‌هایی که توسط کامپیوتر ارسال می‌شوند را نشان دهید.

الف – سگمنت 1 در زمان 0:0:0:000 دریافت شده است.

ب- سگمنت 2 در زمان 0:0:0:027 دریافت شده است.

پ- سگمنت 3 در زمان 0:0:0:400 دریافت شده است.

ت- سگمنت 4 در زمان 0:0:1:200 دریافت شده است.

ث- سگمنت 5 در زمان 0:0:1:208 دریافت شده است.

۲۵- کامپیوتری پنج بسته را می‌فرستد و سه تأییدیه را دریافت می‌کند. زمان با فرمت ثانیه : دقیقه : ساعت نشان داده شده است.

الف- سگمنت 1 در زمان 0:0:00 فرستاده شده است.

ب- سگمنت 2 در زمان 0:0:05 فرستاده شده است.

پ- ACK برای سگمنتهای 1 و 2 در زمان 0:0:07 دریافت شده است.

ت- سگمنت 3 در زمان 0:0:20 فرستاده شده است.

ث- سگمنت 4 در زمان 0:0:22 فرستاده شده است.

ج- سگمنت 5 در زمان 0:0:27 فرستاده شده است.

چ- ACK برای سگمنتهای 1 و 2 در زمان 0:0:45 دریافت شده است.

ح- ACK برای سگمنت 3 در زمان 0:1:5 دریافت شده است.

اگر مقدار اولیه RTO برابر با 6 ثانیه باشد، مقادیر RTT_M ، RTT_S ، RTT_D و RTO را محاسبه نمایید. آیا حالتی رخ داده است که فرستنده، سگمنتی را مجدداً ارسال کرده باشد اما آن را در محاسبات مقادیر فوق، وارد نکرده باشد؟ نشان دهید که کدام سگمنتها می‌بایست دوباره ارسال شوند و زمان ارسال مجدد آنها را نیز تعیین نمایید. رویدادها که شامل زمان ارسال مجدد نیز می‌شود را مجدداً بنویسید.

۲۶- اگر کامپیوتری بایتهای 2001 تا 3000 را به ترتیب دریافت کرده باشد، بایتهای 4001 تا 6000 را خارج از نوبت دریافت کرده باشد و بایتهای 3501 تا 4000 تکرار شده باشند، محتوای گزینه SACK ای که باید ارسال شود را نشان دهید.

۲۷- با استفاده از سناریوی زیر، یک دیاگرام کنترل ازدحام مشابه با شکل ۱۲-۳۶ را نشان دهید. فرض کنید که ماکزیمم سایز پنجره برابر با 64 سگمنت است.

الف- بعد از چهارمین RTT ، سه ACK تکراری دریافت شده است.

ب- بعد از ششمین RTT ، تایمر به پایان مهلت مقرر می‌رسد.

www.Abiri.ir

فعالیتهای تحقیقاتی

۲۸- چند RFC درباره پروتوکل TCP پیدا کنید.

۲۹- RFCهایی درباره گزینه SACK پیدا کنید.

۳۰- RFCهایی درباره الگوریتم کارن پیدا کنید.

۳۱- ما همه قواعد درباره دیاگرام گذر از حالتهای TCP را ارائه نکرده‌ایم. برای کامل کردن مطلب، باید نشان دهیم که برای هر نوع سگمنتی که دریافت می‌شود، حالت بعدی چه باید باشد. همچنین TCP باید بداند که وقتی در یک حالت قرار می‌گیرد، چه عملیاتی را باید انجام دهد. بعضی از قوانینی که در این رابطه به کار می‌روند را پیدا کنید.

۳۲- مورد نیمه- باز (half-open) در TCP چیست؟

۳۳- مورد بسته - نیمه دو طرفه (half duplex close) در TCP چیست؟

۳۴- فرمان tcpdump در یونیکس یا لینوکس را می‌توان برای نمایش هدرهای بسته‌هایی که از طریق کارت شبکه دریافت می‌شوند، به کار برد. با استفاده از tcpdump ، سگمنتهایی که ارسال و دریافت می‌شوند را مشاهده نمایید.

۳۵- فرمان ping در سیستم عامل ویندوز حاوی سویچهایی است که با بکارگیری آنها، یک سری عملیات اضافه‌تری را نسبت به کار معمول فرمان ping می‌توان انجام داد. ببینید هر یک از اعمال زیر را با چه سویچی می‌توانید انجام دهید. لازم به ذکر است که فرمان ping در سطح لایه شبکه کار می‌کند اما از آنجا که سگمنتهای TCP از طریق بسته‌های IP جابجا می‌شوند، لذا درک بهتری از چگونگی کار این فرمان، می‌تواند در خطایابی‌های شبکه، کمک کند.

الف- می‌خواهیم طول عمر بسته IP ، برابر با 3 هاپ باشد.

ب- می‌خواهیم مسیری که بسته IP طی می‌کند، در هدر بسته، ثبت شود.

پ- می‌خواهیم از ابتدا مشخص کنیم که بسته IP باید از چه مسیری عبور کند. برای این کار، لیست اینترفیسهای ورودی روترهایی که در مسیر از مبدأ تا مقصد قرار گرفته‌اند را باید به فرمان ping بدهیم و بسته را چنان مقید کنیم که فقط از همین اینترفیسها وارد روترها شود و هیچ مسیر دیگری را طی نکند.

ت- می‌خواهیم از ابتدا مشخص کنیم که بسته IP باید از چه مسیری عبور کند. برای این کار، لیست اینترفیسهای ورودی روترهایی که در مسیر از مبدأ تا مقصد قرار گرفته‌اند را باید به فرمان ping بدهیم. اگرچه بسته را مقید می‌کنیم که حتماً از این اینترفیسها وارد روترها شود، اما اگر علاوه بر آنها، از اینترفیس دیگری نیز وارد شود، اشکالی ندارد.

ث- می‌خواهیم هم زمانی که بسته به روتری می‌رسد، در هدر بسته IP ثبت شود و هم نشان دهد که بسته IP برای رسیدن به مقصد، از چه روترهایی خارج شده است.

www.Abiri.ir

ج- طول دیتایی که ping ویندوز به طور پیش فرض حمل می‌کند، برابر با 32 بایت است. می‌خواهیم این طول به 100 بایت برسد.

چ- به طور معمول، ping ویندوز، چهار بسته می‌فرستد و پاسخهای مربوطه را نشان می‌دهد. می‌خواهیم 10 بسته ارسال شود و پاسخهای مربوطه، نشان داده شوند.

ح- به طور معمول، ping ویندوز، چهار بسته می‌فرستد و پاسخهای مربوطه را نشان می‌دهد. می‌خواهیم تا زمانی که Control C را نزده‌ایم، فرمان Ping ، به ارسال بسته‌ها و نمایش پاسخ آنها، ادامه دهد.

خ- نمی‌خواهیم بسته ارسالی فرمان ping ، توسط روترهای میان‌راهی، قطعه‌قطعه شود.

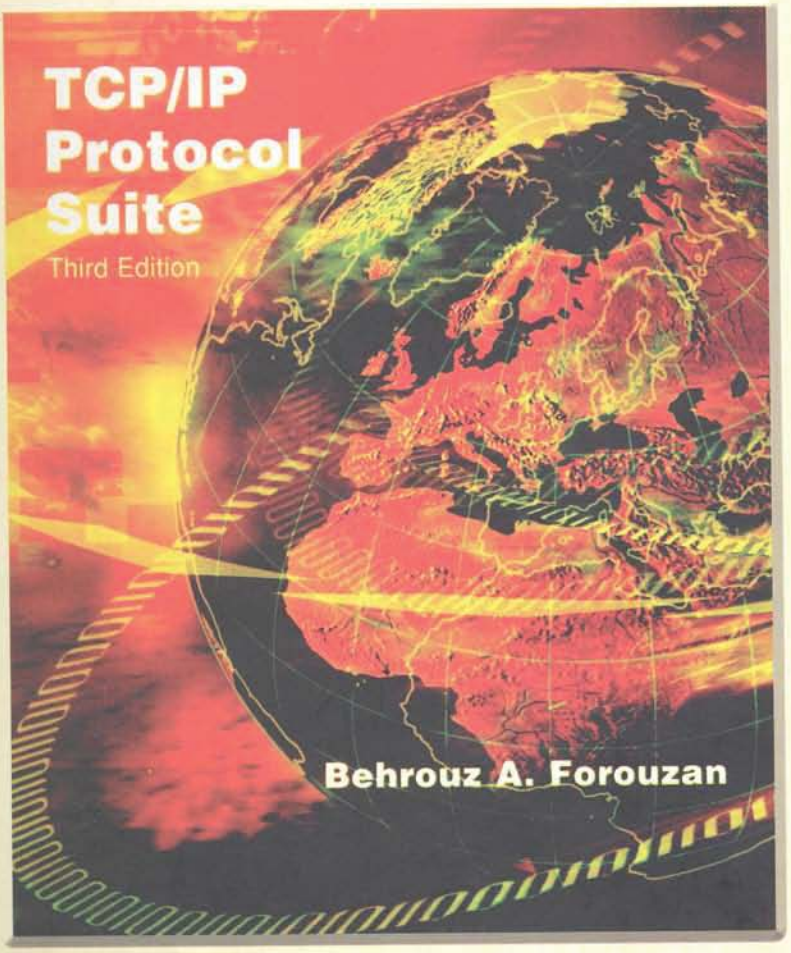
۳۶- فرمان netstat در سیستم عامل ویندوز، برای نشان دادن اتصالاتی که هم‌اکنون برقرار شده‌اند، به کار می‌رود.

الف- ببینید سویچ n- چه عملی انجام می‌دهد.

ب- ببینید سویچ a- چه عملی انجام می‌دهد.

www.Abiri.ir

TCP/IP Protocol Suite



ISBN:964-96198-3-6
9 789649 619835

3rd edition

Update

2006

Translated by: Adham Sadeghi

www.Abiri.ir