فصل هفتم

لایه شبکه (Network layer)

همانطور که در فصل اول ذکر شد وظایف لایه شبکه عبارتند از :

- در اینجا بحث Addressing هم مطرح می‌شود یعنی آدرسها باید واحد، یکتا و جامع باشند.
- وظیفه دیگر این لایه Forwarding است یعنی وقتی بسته‌ای وارد مسیریاب (Router) می‌شود باید یک گام (Hop) به سمت مقصد به پیش رانده شود. از روی جداول درون مسیریاب تشخیص داده می‌شود که هر بسته ورودی از کدام درگاه خروجی خارج شود. این تصمیم‌گیری یا براساس آدرس مقصد و یا شماره ارتباط انجام می‌شود.
- Routing که پیدا کردن بهترین یا مناسبترین مسیر بین مبدا و مقصد است از دیگر وظائف این لایه است. این کار یا به ازای هر بسته تکرار می‌شود و یا یک بار در ابتدای مکالمه در فاز برقراری اتصال (Connection Setup) انجام می‌شود. نتیجه عملیات مسیریابی، به روز رسانی جداول درون مسیریابها است.

نکته : مسیریابها هم وظیفه Routing و هم Forwarding را بر عهده دارند.

- وظیفه دیگر این لایه کنترل ازدحام (Congestion Control) است. باید از اعمال بار بیش از حد بر زیر شبکه ارتباطی (Communication Subnet) جلوگیری شود. زیرا چنانچه بار شبکه از یک حد مشخص بیشتر شود کارایی شبکه روند نزولی را طی خواهد کرد.
- از دیگر وظایف مهم این لایه، تطبیق پروتکل‌ها است (Protocol Matching). لینکهای ورودی و خروجی مسیریابها ممکن است دارای پروتکل‌ها و استانداردهای متفاوت و متعلق به شبکه‌های مختلف باشند. وظیفه دیگر مسیریابها تطبیق پروتکل و یا نگاشت (تبدیل) بسته‌های اطلاعاتی از یک پروتکل به پروتکل دیگر می‌باشد (حذف Header مربوط به پروتکل قبلی و افزودن Header مربوط به پروتکل جدید و به‌طور کلی ایجاد فرمت جدید)

سرویسهایی که لایه شبکه به لایه انتقال می‌دهد بر دو نوع است.

۱ـ Connection less یا بدون اتصال :

وظیفه یک مسیریاب در این شبکه هدایت (Forward) بسته‌ها است و نه چیز دیگر. این شبکه‌ها ذاتاً غیرقابل اعتمادند و کنترل خطا و کنترل جریان را به لایه انتقال می‌سپارند. در این شبکه‌ها ممکن است با تغییر پویای جداول مسیریابی درون مسیریاب ها (با توجه به شرایط جدید شبکه) بسته‌های مربوط به یک مکالمه از مسیرهای متفاوتی و با ترتیب متفاوت به مقصد برسند و یا حتی غلط برسند. اینترنت با یک تجربه ۳۰ ساله از این روش استفاده می‌کند و حتی اگر لایه‌های زیرین IP، کنترل خطا و جریان را انجام دهند فقط دوباره‌کاری کرده‌اند زیرا TCP در لایه چهارم این امر را برعهده دارد. نام دیگر این روش ارسال دیتاگرام (Datagram) است. هرگاه حجم اطلاعات رد و بدل شده در یک مکالمه کم باشد این روش مقرون به صرفه است زیرا سربار فاز برقراری اتصال اولیه را ندارد.

۲ـ Connection Oriented یا اتصال‌گرا :

شبکه‌های سوئیچ تلفنی با تجربه بیش از یک قرن از این مکانیزم استفاده می‌کنند. در این روش در فاز برقراری اتصال یک مسیر مشخص بین مبدا و مقصد ایجاد می‌شود و جداول مسیریابی به روز در می‌آیند. این مسیر را در شبکه‌های سوئیچ تلفنی Circuit (مدار) می‌گویند و به روش سوئیچینگ آن هم Circuit Switching می‌گویند اما در شبکه‌های مدرن به آن Virtual Circuit یا مدار مجازی می‌گویند. در فاز برقراری اتصال، منابع شبکه (Resources) مانند پهنای باند link ها ، فضای بافر در حافظه مسیریاب‌ها ، زمان CPU برای پردازش در گره‌های میانی و غیره باید رزرو شوند تا مطمئن شویم بار اضافه بر زیر شبکه ارتباطی تحمیل نخواهد شد. این کار برای جلوگیری از ازدحام و نیز تضمین تحقق معیارهای کیفیت سرویس (QOS) شامل حداکثر تاخیر، حداقل پهنای باند، حداقل گذردهی (Throughput) ، حداکثر نسبت از دست رفتن بسته‌ها (Packet Loss Ratio) PLR، حداکثر لرزش تاخیر (Delay Jitter) ، قابلیت اطمینان (Reliability) و امنیت (Security) انجام می‌گیرد.

در شبکه‌های مدرن پروتکل Resource ReserVation Protocol (RSVP) برای رزرو منابع بکار می‌رود. ارتباطات اتصال‌گرا مطمئن بوده و از کنترل جریان و خطا بهره‌مندند و با توجه به تضمین کیفیت سرویس برای ارتباطات چندرسانه‌ای نظیر کنفرانس تصویری راه دور و پخش فیلم بکار می‌روند. ATM یکی از مهمترین شبکه‌هایی است که از خدمات اتصال‌گرا استفاده می‌کند. اینترنت نیز برای اینکه از این غافله عقب نماند در IPv6 گامهای بزرگی در جهت تحقق ملزومات QOS برداشته است.

مقایسه زیر شبکه‌های مدار مجازی و دیتاگرام

مورد مقایسه	دیتاگرام	مدارمجازی
تنظیم مدار (Circuit Setup)	مسیریاب نیاز به نگهداری اطلاعات در خصوص وضعیت هر اتصال ندارد.	به ازای هر مدار مجازی تمامی مسیریابها باید اطلاعاتی در خصوص وضعیت آن را نگه دارند (برای تضمین QOS)
آدرس‌دهی	براساس آدرسهای مبدا و مقصد است.	بسته‌ها براساس یک شماره ID مخصوص به VC آدرس‌دهی می‌شوند.
مسیریابی (Routing)	بصورت پویا برای هر بسته مستقلاً انجام می‌شود.	فقط یکبار و آن هم در فاز برقرار اتصال و برپاسازی مدار مجازی انجام شده و همه بسته‌های آن اتصال از آن مسیر هدایت می‌شوند.
تاثیر خرابی مسیریاب	فقط بسته‌هایی خراب می‌شوند که در حافظه مسیریاب خراب در آن در لحظه بار شده بودند.	همه مدارهای مجازی که از مسیریاب خراب عبور می‌کرده‌اند قطع می‌شوند.
تضمین QOS (کیفیت سرویس)	بسیار دشوار است. (مطالب اضافه‌تر در سایت IETF موجود می‌باشد)	در فاز برقراری مدار مجازی یک مذاکره بین کاربر و شبکه انجام می‌شود و کاربر ملزومات QOS خود را اعلام می‌کند و چنانچه شبکه قادر باشد بدون ایجاد مشکلاتی مثل ازدحام آن معیارها را تحقق بخشد و تحقق آنها را تضمین نماید پس از رزرو منابع مورد نیاز، مدار مجازی را برقرار می‌کند و در غیراین صورت مکالمه را نمی‌پذیرد مگر اینکه کاربر توقع خود را کاهش دهد. به این فاز مذاکره CAC (Call Admission Control) می‌گویند.
کنترل ازدحام	بسیار دشوار است اما با مسیریابی پویا امکانپذیر است.	با تخصیص منابع شبکه در فاز CAC از ازدحام جلوگیری می‌شود.

الگوریتم‌های مسیریابی

هر یک از الگوریتم‌های مسیریابی به‌طور کلی 6 ویژگی داشته باشند.

۱) صحت عملکرد (Correctness) : الگوریتم باید صحیح عمل کند

۲) سادگی (Simplicity) :

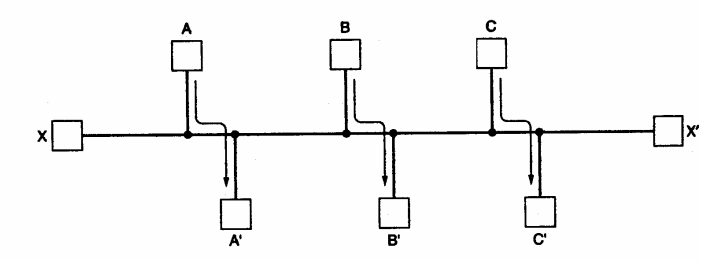
۳) قابلیت تحمل (Robustness) : خرابی سخت‌افزار و نرم‌افزار تاثیری بر عملکرد شبکه نگذارد. (شبکه را از کار نیاندازد)

۴) پایداری(Stability) : الگوریتم همگرا باشد زیرا اگر چنین شرطی وجود نداشته باشد در حلقه ابدی گرفتار خواهد شد.

۵) عدالت و مساوات (Fairness) : منابع به صورت عادلانه تقسیم شوند.

۶) بهینه بودن (Optimality)

برخی از این معیارها متاسفانه با هم در تضاد هستند مثلاً مساوات با بهینگی تضاد دارد و باید موازنه برقرار شود. در شکل زیر برای بهینگی باید ارتباط بین x با x′ قطع باشد تا 3 ارتباط دیگر برقرار شود ولی این با مساوات در تضاد است.



الگوریتم‌های مسیریابی به دو دسته تقسیم می‌شوند:

۱) وفقی (Adaptive) یا پویا ۲) غیروفقی(non Adaptive) یا ایستا

انتخاب مسیر در الگوریتم‌های وفقی بر اساس شرایط فعلی شبکه عوض می‌شود.

از طرف دیگر ا لگوریتم‌های مسیریابی را می‌توان به سه دسته تقسیم کرد:

۱) Centeralized (متمرکز)

۲) Distributed (توزیع شده)

۳) Hierarchical (سلسله مراتبی)

در الگوریتم‌های متمرکز اطلاعات وضعیت شبکه مانند توپولوژی و میزان ترافیک جاری در نقاط مختلف شبکه همگی در یک جا در درون هر مسیریاب متمرکز می‌شوند و هر مسیریاب کل اطلاعات شبکه را در اختیار دارد و تصمیم‌گیری به صورت متمرکز و براساس اطلاعات کامل و سراسری انجام می‌شود. مسیریابی مبدأ یکی از انواع مسیریابی متمرکز است.

اما در الگوریتم‌های مسیریابی توزیع شده تصمیم‌گیری به صورت توزیع شده است و اطلاعات وضعیت شبکه بر روی مسیریاب‌های مختلف توزیع شده است و تصمیم‌گیری (اجرای الگوریتم) نیز به صورت غیر متمرکز و براساس اطلاعات ناقص محلی انجام می‌شود.

در روش سلسله مراتبی برای جلوگیری از بزرگ شدن بیش از حد جداول مسیریابی کل یک شبکه بسیار بزرگ را به تعدادی ناحیه(Region) تقسیم می‌کنیم. هر مسیریاب فقط اطلاعات مسیریابی مربوط به ناحیه خود را دارد ولی چیزی در خصوص جزئیات و ساختار داخلی دیگر نواحی ندارد. البته در شبکه‌های عظیم سلسله مراتب از دو سطح هم بیشتر است. در این شبکه‌ها هر ناحیه به تعدادی خوشه (Cluster) و هر Cluster به تعدادی Zone و هر Zone به تعدادی گروه (Group) تقسیم می‌شوند.

الگوریتم مسیریابی ابتدا کوتاه‌ترین مسیر Shortest Path

در این الگوریتم هر گره دارای یک برچسب دو قسمتی است که حاوی فاصله آن با گره مبدا و نام گره‌ایست که آن گره را به گره مبدا متصل می‌کند. (با فاصله مذکور)

همچنین هر گره در طی پیشرفت الگوریتم یکی از دو وضعیت زیر را دارد:

- T یا Tentative یا موقتی
- P یا Permanent یا دائمی

گره دائمی گره‌ایست که برچسب آن مطمئناً کوتاه‌ترین مسیر تا مبدأ را نشان می‌دهد.

الگوریتم:

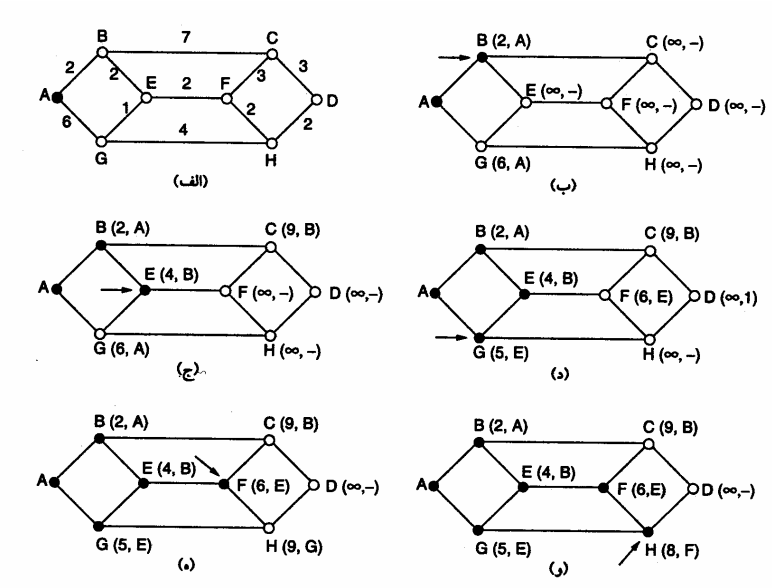
۱) برچسب همه گره‌ها تا مبدا را $(-,\infty)$ قرار دهید (یعنی فاصله آن تا مبدا ∞ و از طریق گره نامشخص)

۲) از گره مبدا شروع می‌کنیم (فرقی کند؛ از مقصد هم می‌توانستیم شروع کرده و تا مبدا ادامه دهیم) آن را دائمی علامت بزنید. این گره را گره کار در نظر می‌گیریم.

۳) برای کلید همسایگان گره کار در صورتی که مجموع برچسب گره کار و فاصله گره کار تا آن گره از برچسب آن گره کوچکتر باشد فاصله هر کدام با گره کار را (وزن link متصل را) با فاصله گره کار تا گره مبدا جمع کنید و به همراه نام گره کار به عنوان برچسب گره همسایه قرار دهید.

۴) به کلیه گره‌های موقتی نگاه کنید. کوچکترین آن‌ها را پیدا کنید و به عنوان گره کار در نظر بگیرید و آن را به صورت دائمی علامت بزنید

۵) اگر همه گره‌ها دائمی نشده‌اند به قسمت ۳ مراجعه کنید.



الگوریتم مسیریابی بردار فاصله یا DVR (Distance Vector Routing)

الگوریتم DVR که نام‌های دیگر آن Bellman-Ford یا Ford-Fulkerson می‌باشد و برای اولین بار در شبکه ARPANET مورد استفاده قرار گرفت و سپس در اینترنت با نام RIP (Routing Information Protocol) به کار گرفته شد.

این الگوریتم به صورت زیر عمل می‌کند:

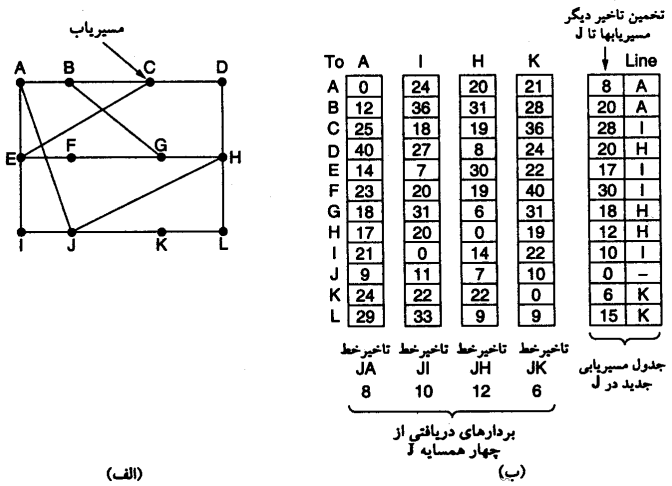
هر مسیریاب یک جدول مسیریابی دارد که به ازای هر مسیریاب موجود در زیر شبکه یک سطر در آن وجود دارد (مراجعه به جدول به کمک اندیس صورت می‌گیرد) در هر سطر دو فیلد زیر وجود دارد:

۱) link خروجی مناسب برای رسیدن به مقصد موردنظر

۲) تخمینی از زمان یا فاصله رسیدن به آن مقصد (این هزینه می‌تواند تعداد گام، تاخیر و یا هر پارامتر دیگر شبکه باشد.)

اگر هزینه نشان‌دهنده تعداد گام است فاصله هر گره با همسایگانش برابر یک در نظر گرفته می‌شود. اگر معیار، طول صف یا تاخیر صف باشد مسیریاب از صف‌های درون خود به سادگی مطلع است و اگر معیار، تاخیر کل، تاخیر انتشار یا صف باشد یک بسته خاص به نام Echo به سمت هر گره همسایه ارسال می‌شود همسایه موظف است فوراً آن را باز گرداند. می‌توان تاخیر کل را فاصله زمانی بین ارسال و دریافت تقسیم بر 2 در نظر گرفت. (با فرض این‌که شبکه متقارن است و زمان رفت و برگشت یکسان است)

شکل زیر نحوه عملکرد این الگوریتم را نشان می‌دهد.



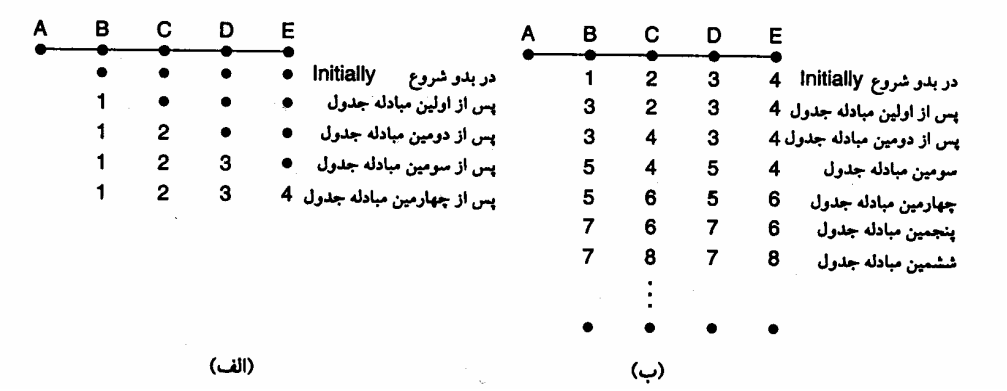
در این شکل می‌بینیم که گره J ابتدا بردار فاصله چهار همسایه خود را (A, I, H, K) را دریافت می‌کند و بر اساس این چهار بردار و فاصله خود از این چهار گره بردار فاصله خود را به روز در می‌آورد.

نکته : این الگوریتم مشکلات اساسی دارد که باعث منسوخ شدن آن شده است. اگرچه از نظر تئوری الگوریتم درست عمل می‌کند اما دو مشکل اساسی زیر دارد:

۱) کندی همگرا شدن

۲) این الگوریتم خبرهای خوب را به سرعت منتقل می‌کند اما در انتقال خبرهای بد واگرا می‌شود و گاهی هرگز همگرا نمی‌شود. خبر خوب یعنی یک نود یا link اضافه شد، ترافیک فلان جا کمتر شد، طول فلان صف کوتاه‌تر شد (برعکس این‌ها خبرهای بدی هستند) به طور کلی این الگوریتم Stable نیست و در برخی شرایط می‌تواند واگرا باشد.

مثال : در این شکل هزینه را تعداد گام می‌گذاریم.



شکل الف انتشار خبر خوب پیوستن A و شکل ب انتشار خبر بد حذف A را نشان می‌دهد. برای حل این مشکل پیشنهاد شده است که حداکثر فاصله را معین کنیم.

مسیریابی حالت پیوند یا LS (Link State)

مشکل شمارش تا بی‌نهایت (∞ Count to Infinity Problem) که در بالا شرح داده شد و الگوریتم RIP یا همان DVR را واگرا می‌کرد و موجب ناپایداری آن می‌شد باعث شد که در سال 1979 الگوریتم دیگری بنام LS جایگزین آن شود. الگوریتم LS مزیت دیگری نیز نسبت به DVR دارد و آن این است که علاوه بر طول صف پهنای باند را نیز در محاسبه تاخیر در نظر می‌گیرد. این الگوریتم در ۵ مرحله زیر عمل می‌کند.

۱) همه همسایگان خود را شناسایی کن و آدرس یکتای هر یک را بدست بیاور

۲) تاخیر یا هزینه (فاصله) هر یک از همسایگان خود را با خود اندازه‌گیری کن (تخمین بزن)

۳) بسته‌ای (Packet) بساز و اطلاعاتی که از همسایگان خود کسب کرده‌ای در آن جاسازی کن

۴) این بسته را برای تمامی مسیریاب‌ها بفرست

۵) با استفاده از الگوریتم کوتاه‌ترین مسیر Dijkstra کوتاه‌ترین مسیر رسیدن به هر یک از مسیریاب های شبکه را محاسبه کن

مرحله ۱) شناسایی همسایه‌ها

هر گاه یک مسیریاب، boot شده و آغاز به کار می‌کند بر روی هر یک از پورت‌های خود بسته‌ای خاص بنام Hello packet را ارسال می‌کند و منتظر می‌نشیند تا پاسخ‌های سلام خود را بشنود. انتظار می‌رود مسیریاب های همسایه در پاسخ سلام آدرس خود را ارسال نمایند.

مرحله ۲) اندازه‌گیری یا تخمین هزینه (تاخیر)

می‌خواهیم ببینیم وضعیت link بین ما با هر یک از همسایگانمان چگونه است و یک تخمین قابل قبول از تاخیر link ها بدست می‌آوریم. برای این کار یک بسته به نام Echo ارسال می‌کنیم و پس از بازگشت بسته Round Trip Time (RTT) را بر 2 تقسیم می‌کنیم. با فرض تقارن شبکه و تکرار این عمل و میانگین‌گیری تقریب خوبی از تاخیر بدست می‌آید.

مرحله ۳) ساخت بسته‌های وضعیت LINK (Link State Packet)

بسته وضعیت link حاوی فیلدهای زیر است:

۱) آدرس فرستنده

۲) شماره ترتیب (اولین بسته از صفر شماره‌گذاری می‌شود)

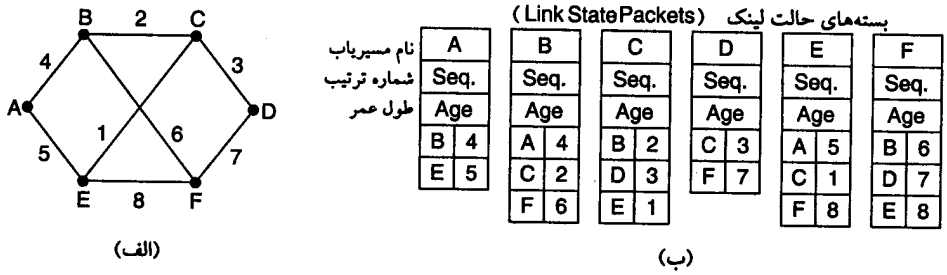
۳) Age یا (Time To Live) TTL کد یک شمارنده است و از مقدار معینی شروع می‌شود و هر دفعه (با عبور از هر مسیریاب یا گذشت یک ثانیه) یک واحد از آن کم می‌شود و هر وقت به صفر رسید این بسته از بین می‌رود.

۴) فهرست همسایه‌ها و وضعیت (تاخیر link بین ما و هر همسایه)

نکته : این بسته‌ها چه زمانی ارسال می‌شود؟ دو راه داریم

الف) پریودیک (در زمان‌های خاص)

ب) هر وقت تغییر ذاتی در توپولوژی شبکه یا وضعیت Link ها (میزان تاخیر و غیره) مشاهده شود.



مرحله ۴) توزیع بسته‌های Link state

مهمترین نکته در توزیع این بسته‌های LS همگام‌سازی مسیریاب های دریافت کننده این بسته‌ها است زیرا اگر بعضی از router ها زودتر این بسته‌ها را دریافت کنند و جداول مسیریابی خود را به روز درآورند ولی هنوز این بسته‌ها توسط مسیریاب های دیگر دریافت نشده باشد اختلاف بین این جداول مشکلاتی از قبیل پیدایش حلقه بی‌نهایت و جدا شدن بعضی از مسیریاب‌ها را در توپولوژی شبکه ایجاد می‌کند.

یک راه‌حل برای این مشکل الگوریتم مسیریابی سیل‌آسا (Flooding) است. که موردبحث قرار خواهد گرفت.

نکته : دقت شود که برای این‌که جداول نگهدارنده این بسته‌ها بیش از حد بزرگ و پردازش آن‌ها پیچیده نشود و اطلاعات زائد در آن نباشد باید آخرین بسته ارسالی از هر مسیریاب را جایگزین قبلی نماییم اما طبق الگوریتم سیل‌آسا ممکن است بسته قدیمی بعد از بسته جدید از راه برسد و در نتیجه اعتبار اطلاعات از بین می‌رود (چون جایگزین اطلاعات جدید می‌شود) راه‌حل این مشکل استفاده از شماره ترتیب است. بنابراین در صورتی بسته دریافتی جایگزین می‌شود که شماره ترتیب آن بزرگتر از قبلی باشد.

نکته : اگر محدودهٔ شماره کوچک باشد مثلاً 4 بیتی بعد از 16 بسته دوباره reset شده و طبق الگوریتم فوق بسته‌های دیگر در نظر گرفته نمی‌شوند. راه‌حل این است که محدوده شماره را 32 بیتی و بزرگ در نظر بگیریم.

نکته : همچنین با گذشت زمان طبق فیلد Age بسته Expired یا منقضی می‌شود.

مرحله ۵) محاسبه مسیرهای جدید

این کار توسط الگوریتم کوتاه‌ترین مسیر Dijkstra به راحتی انجام می‌شود.

پروتکل OSPF (Open Shortest Path First)

الگوریتم باز ابتدا کوتاه‌ترین مسیر (OSPF) یکی از رایج‌ترین الگوریتم‌های مسیریابی شبکه اینترنت است. این پروتکل توسعه یافته الگوریتم LS محسوب می‌شود.

پروتکل IS – IS (Intermediate System Intermrdiate System)

این پروتکل نیز مبتنی بر اطلاعات وضعیت link بوده و توسط شرکت Dec Net با توسعه LS بوجود آمده است. این پروتکل برای لایه شبکه CLNP که در محصولات این شرکت به کار می‌رفت طراحی شد. اما این پروتکل ویژگی بسیار جالبی دارد و قادر است همزمان با چندین پروتکل شبکه کار کند. Novel Netware نیز از این پروتکل برای هدایت بسته‌های IPX در لایه شبکه خود استفاده می‌کرد. به عبارت دیگر همزمان می‌توان چندین استاندارد آدرس‌دهی شبکه مانند Apple talk ، CLNP، IP و IPX را با این پروتکل پشتیبانی کرد.

این ویژگی مهم در OSPF دیده نمی‌شود. IS – IS در ستون فقرات بخش‌های مهمی از شبکه اینترنت به کار رفته است .

الگوریتم سیل‌آسا (Flooding)

در این الگوریتم سیلی از بسته‌ها از مسیرهای مختلف در آن واحد به سمت مقصد (در واقع در همه جهات) ارسال می‌شود. هر مسیریاب موظف است با دریافت آن بسته یک نسخه از آن را به تمام پورت‌های خروجی ارسال کند. واضح است که در این الگوریتم بسته‌های تکراری از مسیرهای مختلف به کلیه گره‌ها خواهد رسید و تولید بسته‌های تکراری موجب ازدحام و اشباع شبکه خواهد شد. برای حل این مشکل پیشنهاداتی ارائه شده است:

۱) یک شمارنده گام (Hop counter) داشته باشیم و در Header بسته قرار دهیم و در هر گام یک واحد از آن کم کنیم و پس از صفر شدن آن، بسته را دور بریزیم.

۲) فهرست بسته‌های سیل‌آسای ارسالی از هر گره مبدا را از طریق شماره ترتیب آن نگهداری نمائید و از ارسال مجدد بسته‌های تکراری جلوگیری کنیم.

۳) برای اجتناب از طولانی شدن این لیست فقط کافی است آخرین بسته (بزرگترین شماره ترتیب) مربوط به هر گره مبدا را لیست کنیم.

مسیریابی انتشاری یا (Broadcast Routing)

برای انتشار بسته‌ها در لایه شبکه در شبکه‌ای مانند اینترنت چه باید کرد؟ هر یک از الگوریتم‌های زیر را می‌توان برای انتشار بسته‌ها از یک مبدا به همه میزبان‌های درون شبکه پیشنهاد کرد. البته هر روش مزایا و معایب خود را دارد.

روش ۱) یک لیست از آدرس همه مقاصد داشته باشیم و در یک حلقه به صورت نقطه به نقطه بسته را به یکایک ماشین‌ها ارسال کنیم. مشکلات این روش عبارتند از اتلاف پهنای باند، کندبودن الگوریتم و نیاز به نگهداری فهرست طولانی از آدرس‌ها

روش ۲) استفاده از الگوریتم مسیریابی سیل‌آسا

روش ۳) مسیریابی چندمقصدی (Multi-Destination Routing) : در این روش در آدرس بسته یک نگاشت بیتی وجود دارد (Bitmap) که هر بیت آن یکی از گره‌های شبکه را نشان می‌دهد. حال فرض کنید یک بسته انتشاری به یک گره می‌رسد بیت مربوط به خود را reset می‌کند و بسته را در صورتی به سمت link های خروجی می‌فرستد که بیت مربوط به گره متصل به آن link یک باشد (reset نشده باشد.)

روش ۴) استفاده از درخت پوشا (Spaning Tree): درخت پوشا درختی است (بدون حلقه) که شامل همه‌گره‌های شبکه می‌شود. اگر بهینه باشد به آن Sink Tree می‌گویند.

نکته : Spaning Tree واحد نیست. کافی است مسیریاب‌ها، اطلاعات یکی از درخت‌های پوشا را داشته باشند و بسته را از طریق این درخت یا شاخه‌های این درخت به همه گره‌ها برسند. مسیریاب با استفاده از اطلاعات وضعیت link ها می‌تواند این درخت را پیدا کند.

روش ۵) Reverse Path Forwarding (هدایت بر روی مسیر معکوس)

هر گره فقط بسته‌های پخشی را در صورتی می‌پذیرد که از مسیری دریافت شده باشد که برای ارسال یک بسته معمولی، آن بسته از طریق آن مسیر به سمت گره مبدا بسته‌های پخشی ارسال می‌شود. به عبارت دیگر بسته‌هایی که از سایر link ها دریافت می‌شود دور ریخته می‌شود تا از تکرار بسته‌های اضافی جلوگیری شود. بسته‌ای که از مسیر معکوس دریافت می‌شود به سمت هر یک از گره‌های مجاور ارسال می‌شود.

مسیریابی در شبکه‌های بی‌سیم متحرک

پیچیدگی این شبکه‌ها بسیار زیاد است زیرا ماشین‌ها حرکت می‌کنند و از یک حوزه وارد حوزه‌های دیگر می‌شوند. مثلا در شبکه تلفنی سلولی از یک سلول وارد سلول‌های دیگر می‌شوند.

در این شبکه‌ها چند مفهوم جدید تعریف می‌شود.

۱) ماشین متحرک یک محل استقرار دائمی دارد! اگرچه ممکن است در آن‌جا نباشد (مثل یک تلفن همراه که اگرچه شماره تهران (محل استقرار دائمی) است اما ممکن است اکنون در تبریز باشد)

۲) یک عامل خانگی (Home Agent) که یک برنامه است در محل استقرار دائمی وجود دارد (برای مثال ما در تهران)

۳) در هر ناحیه خارجی (شبکه از نظر جغرافیایی به چند ناحیه تقسیم می‌شود) یک عامل خارجی (Foreign Agent) وجود دارد وقتی ماشین متحرک واحد ناحیه خارجی می‌شود صبر می‌کند تا یک پیغام از عامل خارجی دریافت کند. این پیغام مبنی بر این است که آیا ماشین متحرک خارجی در این ناحیه وجود دارد؟ اگر ماشین متحرک منتظر شود و این پیغام را دریافت نکند خودش یک پیغام منتشر می‌کند که آیا یک عامل خارجی در این‌جا وجود دارد؟

خلاصه در صورتی‌که عامل خارجی ماشین خارجی را پیدا کند ماشین متحرک در آن عامل خارجی ثبت‌نام می‌کند. عامل خارجی یک پیغام به عامل خانگی می‌فرستد (در مثال ما از تبریز به تهران) تا از این پس بسته‌های به مقصد ماشین متحرک به حوزه خارجی مربوط مسیریابی شود.

کیفیت خدمات (Quality of Services)

در تمامی شبکه‌های کامپیوتری پیشرفته تکنیک‌هایی متعدد وجود دارد که تمرکز ویژه‌ای بر روی تضمین کیفیت خدمات (QoS) متناسب با نیازهای برنامه‌های کاربردی دارند. این نیازها با چهار پارامتر ″ قابلیت اطمینان ″، ″ تاخیر″ ، ″ لرزش″، و ″ پهنای باند″ مشخص می‌شوند. راهکارهای مختلف دستیابی به کیفیت خوب خدمات به شرح زیر می‌باشد:

کنترل ازدحام (Conjestion Control) و شکل‌دهی ترافیک

سیاست‌های مختلفی در لایه‌های مختلف شبکه برای کنترل و پیش‌گیری از ازدحام پیشنهاد شده است. در هر حال دقت کنید که سیاست‌های گوناگونی بر پدیده ازدحام تاثیر مثبت یا منفی می‌گذارند. برای مثال در لایه پیونده داده سیاست ارسال مجدد، سیاست کنترل جریان، سیاست ارسال ACK و سیاست ذخیره بسته‌های خارج از ترتیب بر ازدحام تاثیر می‌گذارند.

همچنین در لایه شبکه سیاست‌هایی از جمله مسیریابی، طول عمر بسته‌ها، روش‌های مدار مجازی و رزرو منابع، مکانیزم‌های صف‌بندی و حذف بسته‌های اضافی بر کاهش ازدحام موثر خواهند بود.

همچنین در لایه انتقال سیاست‌هایی نظیر ارسال مجدد، ACK ذخیره بسته‌های خارج از ترتیب، کنترل جریان و زمان انقضای تایمرها بر ازدحام موثرند.

نکته ۱ : یکی از بهترین مکانیزم‌ها برای جلوگیری از ازدحام ایجاد مدار مجازی و رزرو منابع توسط پروتکل‌هایی نظیر RSVP است

نکته ۲: چگونه می‌توان در روش‌هایی مانند دیتاگرام از ازدحام اجتناب کرد؟

برای کنترل ازدحام در این شبکه‌ها مکانیزم‌های مختلفی پیشنهاد شده است که چند مورد از آن‌ها عبارتند از:

۱) Set کردن بیت هشدار در بسته‌ها در مواقعی که حجم ترافیک از یک حد آستانه بالاتر می‌رود.

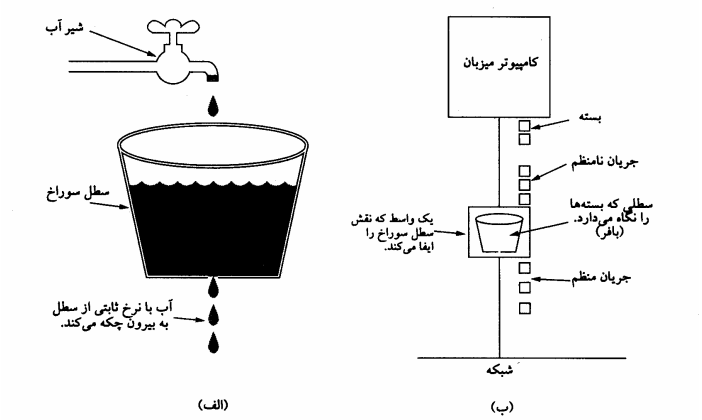
۲) ارسال بسته‌های خاص دعوت به آرامش (Chock) در شرایطی که حجم ترافیک سنگین شده است.

۳) دور ریختن بار اضافی مشتریان در صورتی‌که مشتری‌ها از تعهدات مندرج در مذاکره اولیه تخطی کرده‌اند.

نکته ۳: برای کنترل Jitter چه باید کرد؟

این کار به راحتی انجام می‌شود. بسته‌هایی‌که با نرخ متغیر و فواصل زمانی متفاوت دریافت می‌شوند در یک بافر ذخیره کرده و از یک طرف بسته‌ها را با نرخ ثابت از بافر خارج می‌کنیم.

نکته ۴: الگوریتم سطل سوراخ‌دار (Leaky Bucket) یکی از الگوریتم‌هایی است که در جهت افزایش QOS و کاهش ازدحام و جلوگیری از تحمیل بار اضافی توسط مشتری‌ها یا میزبان‌ها بر شبکه طراحی شده است. فرض کنید ترافیک نامنظمی را که یک کاربر ارسال می‌کند با قطره‌های نامنظم و تصادفی که به یک سطل سوراخ وارد می‌شوند مدل کنیم. همچنین فرض کنید که بافر اولین مسیریاب سر راه این بسته‌ها را تقسیم کنیم و یک فضای خاص با حجم معین به آن مشتری اختصاص دهیم (مدل این بخش از بافر، سطلی است که ظرفیت مشخصی دارد و در صورت پر شدن سرریز شده و بار اضافی ورودی دور ریخته می‌شود.) از آن‌جا که اندازه سوراخ زیر سطل ثابت است قطرات از خروجی سیستم به‌طور منظم و با نرخ ثابت از بافر سطل خارج می‌شوند.



رزرو منابع (Resource Reservation)

توانایی شکل‌دهی و تنظیم ترافیک ارسالی، تمهید خوبی برای تضمین « کیفیت خدمات » (QoS) محسوب می‌شود ولیکن استفاده از این روش‌ها زمانی کارآمد خواهد بود که تمامی بسته‌ها از مسیر یکسانی عبور کنند. پراکندگی تصادفی بسته‌ها بر روی مسیرهای متفاوت، تضمین هر چیزی را بسیار دشوار می‌کند. بنابراین برای تامین کیفیت خدمات باید بین مبدا و مقصد چیزی شبیه به یک مدار مجازی ایجاد و تنظیم شود و تمام بسته‌های یک « جریان » از این مسیر حرکت کنند.

هر گاه برای جریان داده‌ها، مسیر ویژه داشته باشیم می‌توان منابع لازم را در طول این میسر، رزرو کرده و موجود بودن ظرفیت موردنیاز را تضمین کرد. سه نوع متفاوت از منابع را می‌توان از قبل رزرو کرد:

۱ـ پهنای باند

۲ـ فضای بافر

۳ـ سیکلهای CPU [ظرفیت پردازش موردنیاز]

کنترل پذیرش (Admission Control)

حال در مرحله‌ای هستیم که ترافیک ورودی از یک « جریان » (Flow) خاص به خوبی شکل و نظم داده شده و بسته‌ها از یک مسیر واحد حرکت می‌کنند و پیشاپیش ظرفیت موردنیاز در طول مسیر، پیش‌بینی و رزرو شده است. با چنین فرضی، هر گاه جریانی از بسته‌ها به یک مسیریاب تسلیم شود بر اساس ظرفیت موجود خود و سطح تعهداتی که در خصوص دیگر جریان‌ها پذیرفته، باید در خصوص قبول یا رد آن تصمیم بگیرد.

چونکه برای رسیدن به توافق نهایی در خصوص تامین نیازهای یک « جریان » باید مولفه‌های متعددی در مذاکرات شرکت داشته باشند (اعم از فرستنده، گیرنده و تمام مسیریاب‌های واقع بر روی مسیر)، لذا هر « جریان » باید بر حسب پارامترهای مشخصی به‌دقت توصیف شود تا بتوان بر روی این پارامترها مذاکره و توافق کرد. مجموعهٔ چنین پارامترهایی اصطلاحاً « مشخصات توصیفی جریان » (Flow Specification) نامیده می‌شود. بدین‌ترتیب یک فرستنده (مثل سرویس‌دهنده ویدیو) مشخصات توصیفی جریان را به صورت پارامترهای پیشنهادی و موردنظر خود تعریف می‌نماید. این پارامترهای پیشنهادی در طول مسیر منتشر می‌شود و هر مسیریاب واقع بر مسیر آن‌ها را بررسی کرده و درصورت نیاز در آن‌ها تغییراتی ایجاد می‌کند. این تغییرات فقط کاهشی است نه افزایشی (یعنی مثلاً نرخ موردنظر ارسال داده‌ها را کاهش می‌دهد نه افزایش). وقتی این پارامترها به طرف مقابل برسد، به اجرا گذاشته می‌شوند.

زمان‌بندی بسته‌ها

هرگاه یک مسیریاب هدایت چندین « جریان » را بر عهده داشته باشد این خطر وجود داردکه یک «جریان» از حدود و ظرفیت مجاز خود تجاوز نماید و در نتیجه جریان‌های دیگر را با کمبود منابع (Starvation) مواجه سازد. اگر پردازش بسته‌ها به ترتیب ورودشان انجام گیرد باعث می‌شود که یک فرستنده متجاوز بتواند بیشتر ظرفیت مسیریاب‌هایی را که بر روی خط سیر بسته‌های او هستند اشغال کرده و کیفیت خدمات دیگران کاهش یابد. برای خنثی کردن چنین تلاشی، الگوریتم‌هایی جهت زمان‌بندی بسته‌ها پیشنهاد شده است.

یکی از اولین روش‌ها، الگوریتم « صف‌بندی بی‌طرفانه» (Fair Queuing) است. جوهرهٔ این الگوریتم آن است که مسیریاب‌ها باید برای هر خط خروجی و به ازای هر « جریان » که از آن خط خروجی می‌گذرد، صف‌های جداگانه‌ای تشکیل بدهند. هر گاه خطی بیکار شود، مسیریاب‌ها صف‌ها را به ترتیب پویش کرده و از سر هر صف یکی را بر می‌دارد. بدین‌ترتیب، در شرایطی که n ماشین میزبان برای یک خط خروجی رقابت می‌کنند، از هر n بسته ارسالی بر روی خط یک بسته به هر ماشین میزبان تعلق می‌گیرد. افزایش نرخ ارسال بسته‌ها، در نسبت سهم هر ماشین تغییری ایجاد نخواهد کرد.

یک اشکال این الگوریتم آن است که به تمام ماشین‌های میزبان، اولویت یکسانی می‌دهد. در بسیاری از محیط‌ها مطلوب‌تر آن است که به سرویس‌دهنده‌های ویدیو (Video Server) اولویت بیشتری نسبت به یک سرویس‌دهنده معمولی فایل داده شود و در هر تیک ساعت، سهم آن دو یا چند بایت باشد. این الگوریتم اصلاح شده به نام الگوریتم صف‌بندی بی‌طرفانهٔ وزن‌دار» (Weighted Fair Queuing) مشهور است و کاربرد گسترده‌ای دارد.

خدمات مجتمع (Integrated Services)

در خلال سال‌های 1995 تا 1997، تلاش IETF بر آن بود که برای انتقال داده‌های مالتی مدیا (Multimedia Streaming) معماری مناسبی ابداع کند. این پروژه با نام کلی « الگوریتم‌های مبتنی بر جریان» (Flow – based algorithms) یا « خدمات مجتمع » (Integrated Services) شناخته می‌شود و کاربردهای چند پخشی (Multicast) و تک پخشی (Unicast) را در بر می‌گیرد.

به عنوان مثالی از کاربردهای چندپخشی، ایستگاه‌های پخش تلویزیون دیجیتال را در نظر بگیرید که برنامه‌های خود را در قالب جریانی از بسته‌های IP به گیرندگان بی‌شمار و پراکندهٔ خود ارسال می‌دارند.

اصلی‌ترین پروتکل پیشنهاد شده توسط IETF برای ارائه خدمات مجتمع، RSVP نامیده می‌شود و برای رزرو کردن پهنای باند به‌کار می‌آید. RSVP اجازه می‌دهد که چندین فرستنده بتوانند برای چندین گروه از گیرندگان خود داده بفرستند و همچنین امکان آن را فراهم کرده که گیرندگان بتوانند کانال موردنظر خود را آزادانه عوض کنند. در عین حال پروتکل RSVP، استفاده از پهنای باند را بهینه‌سازی کرده و از بروز ازدحام جلوگیری می‌کند.

خدمات متمایز (Differentiated Services)

«الگوریتم‌های مبتنی بر جریان» قابلیت عرضه کیفیت خوب خدمات به یک یا چند جریان را دارند زیرا در طول مسیر هر منبعی را که نیاز است از قبل رزرو می‌کنند. ولی این روش‌ها یک اشکال دارند: در این الگوریتم‌ها نیاز است که برای هر جریان (Flow) پیشاپیش تنظیمات لازم انجام شود در حالی که در مقیاس کلان یعنی وقتی که هزاران یا میلیون‌ها « جریان » وجود دارد قابلیت اجرایی خود را از دست می‌دهند. از طرفی در هر مسیریاب « وضعیت » هر جریان به‌طور جداگانه نگهداری می‌شود و عملکرد این الگوریتم‌ها در مقابل خرابی یک مسیریاب آسیب‌پذیر خواهد بود. نهایتاً آن‌که برای تنظیم و ایجاد « جریان » باید تبادل اطلاعات پیچیده‌ای بین مسیریاب‌ها انجام گیرد. در نتیجه RSVP یا الگوریتم‌های مشابه آن بسیار کم پیاده‌سازی عملی شده‌اند.

به همین دلایل، IETF راهکارهای ساده‌تر برای تامین کیفیت خدمات (QoS) ابداع کرد؛ روشی که بدون نیاز به هیچ تنظیمات قبلی یا تعیین کل مسیر می‌تواند به صورت محلی و مجزا در هر مسیریاب پیاده‌سازی شود. این راهکار اصطلاحاً « روش مبتنی بر کلاس » (Class – Based) برای تضمین کیفیت خدمات نامیده می‌شود (در مقابل روش‌های مبتنی بر جریان). IETF یک معماری مناسب به نام «خدمات متمایز » برای آن طراحی و استانداردسازی کرده است.

« خدمات متمایز » (که به اختصار DS گفته می‌شود) می‌تواند توسط مجموعه‌ای از مسیریاب‌ها که در یک «حوزهٔ مدیریتی واحد» (Administrative Domain) قرار می‌گیرند (مثلاً یک ISP یا شرکت مخابرات)، عرضه شود. مدیریت مسئول شبکه، مجموعه‌ای از کلاس‌های متفاوت خدمات و متناظر با آن، قواعد هدایت بسته‌ها (Forwarding Rules) را تعریف می‌کند. پیاده‌سازی خدمات DS بسیار آسان است.

سوئیچ برچسب و MPLS [۱]

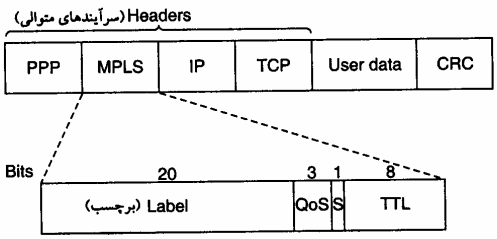
در ابتدای هر بسته یک «برچسب» (Label) اضافه شود و به‌جای آن‌که مسیریابی و هدایت بسته‌ها مبتنی بر آدرس مقصد باشد براساس این «برچسب» انجام شود. با استفاده از این «برچسب» به عنوان یک اندیس در جدول داخلی هر مسیریاب، خط خروجی صحیح و مناسب برای هر بسته پیدا می‌شود. به کمک این روش، مسیریابی بسته‌ها به سرعت انجام شده و منابع موردنیاز در طول مسیر رزرو خواهد شد.

البته برچسب‌گذاری بر روی هر «جریان» شباهت عجیبی به مدارهای مجازی پیدا می‌کند. در شبکه‌های ATM، X.25 و Fram Relay یا هر زیر شبکه مدار مجازی دیگر نیز یک « برچسب» (یا به عبارتی یک شناسهٔ مدار مجازی) در هر بسته قرار داده می‌شود و با استفاده از آن به عنوان یک اندیس برای درایه‌های جدول، مسیر مناسب به‌دست می‌آید.

ایدهٔ جدید سوئیچینگ با نام‌های متنوعی مثل «سوئیچینگ برچسب»[۲] یا « سوئیچینگ علامت»[۳] شناخته می‌شود. در نهایت IETF آن را تحت نام MPLS استاندارد کرد.

مضاف بر این، برخی افراد بین «مسیریابی » و « سوئیچینگ » فرق می‌گذارند. مسیریابی فرآیند جستجو در جدول مسیریابی به دنبال آدرس مقصد هر بسته و پیدا کردن خط مناسب برای آن است. برعکس در فرآیند سوئیچینگ از برچسب هر بسته به عنوان یک اندیس در جدول مسیریابی استفاده می‌شود و با استفاده از این اندیس بلافاصله خط خروجی پیدا می‌شود، بدون آن که نیازی به جستجو باشد. البته این تعاریف و تعابیر جهان شمول و همگانی نیستند.

اولین مسئله آن است که این برچسب در کجا قرار داده شود. از آنجایی که بسته‌های IP برای شبکه‌های مدار مجازی طراحی نشده بودند، طبعاً هیچ فیلدی در سرآیند بستهٔ IP برای درج شماره‌های مدار مجازی وجود ندارد به همین دلیل سرآیند جدید MPLS، باید در جلوی سرآیند هر بسته IP قرار بگیرد. در خطوط مستقیم بین هر دو مسیریاب که مبتنی بر « فریمینگ PPP » کار می‌کنند ترتیب سرآیندها طبق شکل زیر عبارتند از: سرآیند PPP، سرآیند MPLS، سرآیند IP و نهایتاً سرآیند TCP. در واقع باید MPLS را در لایه 2.5 فرض کرد!!!



ارسال یک قطعه TCP (TCP Segment) با استفاده از IP , MPLS ، و PPP.

سرآیند عمومی MPLS (MPLS Header) چهار فیلد دارد که مهمترین آن‌ها فیلد Lable (فیلد برچسب) است که در آن یک اندیس درج می‌شود. فیلد QoS، کلاس خدمات را مشخص می‌کند. فیلد S بدان منظور تعریف شده که در شبکه‌های سلسله مراتبی چندین سرآیند MPLS متوالیاً به بسته اضافه گردد. فیلد TTL زمان حیات بسته را مشخص می‌کند و به ازای هر گام یک واحد از آن کم می‌گردد؛ هر گاه مقدار این فیلد به صفر برسد، بسته حذف می‌شود. این ویژگی بدان منظور مفید است که از حلقهٔ بی‌نهایت که در اثر ناپایداری (

---

[۱] Multi Protocol Label Switching

[۲] Label Switching

[۳] Tag Switching

عدم همگرایی) جدول مسیریابی بروز می‌کند، اجتناب شود.

از آنجایی که سرآیند MPLS بخشی از بسته لایه شبکه یا فریم لایه پیوند داده‌ها محسوب نمی‌شود لذا MPLS تا حد زیادی مستقل از هر دو لایه است. از بین تمام محاسن دیگر، دستاورد ویژگی « استقلال از دیگر لایه‌ها » آن است که می‌توان سوئیچ‌های MPLS را به گونه‌ای ساخت که بتواند هم بسته‌های IP و هم سلول‌های ATM را برحسب مورد، هدایت کند. این ویژگی همانی است که براساس آن کلمه Multiprotocol در ابتدای نام MPLS ظاهر شده است.

وقتی یک بسته یا سول غنی‌شده با سرآیند MPLS در یک مسیریاب MPLS دریافت می‌شود از برچسب آن به عنوان اندیسی در جدول داخلی مسیریاب استفاده شده و خط خروجی متناسب با آن تعیین می‌شود و قبل از خروج بسته از آن خط، برچسب جدیدی در فیلد مربوطه درج می‌گردد. تغییر در برچسب‌ها در تمام زیر شبکه‌های مدار مجازی معمول و متعارف است چرا که برچسب‌ها در هر مسیریاب معنای محلی دارند و دو مسیریاب متفاوت ممکن است بسته‌های نامربوط را با برچسبی یکسان برای مسیریاب دیگر بفرستند چرا که این بسته‌ها همگی در بخشی از مسیر مشترک‌اند. به همین دلیل در هر گام برچسب‌های بسته قبل از انتقال بر روی خط خروجی به برچسب جدید و معتبر در مسیریاب بعدی نگاشته می‌شود.

فصل هشتم

پروتکل اینترنت (IP)

جوهرهٔ اینترنت به گونه ای شکل گرفته است که مجموعه ای از شبکه‌های خودمختار را به همدیگر وصل می‌نماید. قراردادی که حمل و تردّد بسته‌های اطلاعاتی و همچنین مسیریابی صحیح آنها را از مبدأ به مقصد ، مدیریّت و سازماندهی می‌نماید پروتکلIP نام دارد. درحقیقت پروتکل IP که روی تمامی ماشینهای شبکه اینترنت وجود دارد بسته‌های اطلاعاتی را (بسته‌های IP) از مبداء تا مقصد هدایت می‌نماید، فارغ از آنکه آیا ماشینهای مبدأ و مقصد روی یک شبکه هستند یا چندین شبکهٔ دیگر بین آنها واقع شده است.

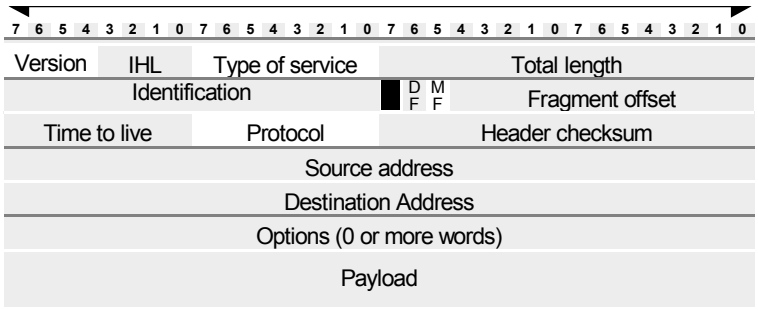
ساده ترین تعریف برای پروتکل IP روی شبکهٔ اینترنت بصورت زیر خلاصه می‌شود: لایهٔ IP یک واحد از داده‌ها را از لایهٔ بالاتر تحویل می‌گیرد؛ به این واحد اطلاعات معمولاً یک "دیتاگرام" گفته می‌شود. امکان دارد طول این دیتاگرام بزرگ باشد ، در چنین موردی لایهٔ IP آنرا به واحدهای کوچکتری که هر کدام "قطعه" (Fragment) نام دارد شکسته و با تشکیل یک بستهٔ IP به ازای هر قطعه ، اطلاعات لازم برای طی مسیر در شبکه را به آنها اضافه میکند و سپس آنها را روی شبکه به جریان می‌اندازد؛ هر مسیریاب با بررسی و پردازش بسته‌ها ، آنها را تا مقصد هدایت می‌کند . هر چند طول یک بسته IP می‌تواند حداکثر 64Kbyte باشد و لیکن در عمل عموماً طول بسته‌ها حدود ۱۵۰۰ بایت است.

در کنار پروتکل IP چندین پروتکل دیگر مثل ICMP، ARP، RARP، RIP و غیره تعریف شده که پروتکل IP را در عملکرد بهتر ، مسیریابی صحیح، مدیریت خطاهای احتمالی یا کشف آدرسهای ناشناخته کمک می‌کنند.

قالب یک بستهٔ IP

شکل زیر قالب یک بسته IP را به تصویر کشیده است. یک بستهٔ IP از دو قسمت سرآیند و قسمت حمل داده تشکیل شده است. مجموعهٔ اطلاعاتی که در سرآیند بستهٔ IP درج می‌شود توسط مسیریابها مورد استفاده و پردازش قرار می‌گیرد.

32 Bits



فیلد Version : اولین فیلد در سرآیند یک بستهٔ IP که چهار بیت است نسخهٔ پروتکلِ IP که این بسته بر اساس آن سازماندهی و ارسال شده است را تعیین می‌کند. در حال حاضر تمامی شبکه‌ها و مسیریابها از نسخهٔ شماره ۴ پروتکل IP پشتیبانی می‌کنند. امروزه نسخهٔ شمارهٔ ۶ پروتکل IP به نامهای IPng یا IPv6 معرفی و در حال بررسی و نصب است. عددی که در حال حاضر در این فیلد قرار می‌گیرد ۴ یا $(0100)_B$ است .

فیلد IHL[1] : این فیلد هم چهاربیتی است و طول کل سرآیند بسته را بر مبنای کلمات ۳۲ بیتی مشخص می‌نماید. غیر از فیلد Options که اختیاری است، وجود تمامی فیلدهای سرآیند الزامی می‌باشد . طول قسمت اجباریِ سرآیند ۲۰ بایت است و بهمین دلیل حداقل عددی که در فیلد IHL قرار می‌گیرد ۵ یا $(0101)_2$ خواهد بود و هر مقدار کمتر از ۵ به عنوان خطا تلقّی شده و منجر به حذف بسته خواهد شد. با توجه به طول ۴ بیتی این فیلد ، بدیهی است که حداکثر مقدار آن ۱۵ یا $(1111)_2$ خواهد بود که در این صورت طول قسمت سرآیند ۶۰ بایت (۴×۱۵) و طول قسمت اختیاری ۴۰ بایت می‌باشد. قسمت اختیاری در سرآیند برای اضافه کردن اطلاعاتی مثل آدرس مسیرهای پیموده شده، "مهر زمان" و برخی دیگر از گزینه‌هاست که در ادامه توضیح داده خواهد شد.

فیلد Type of service : این فیلد هشت بیتی است و توسط آن ماشین میزبان (یعنی ماشین تولید کنندهٔ بستهٔ IP) از مجموعهٔ زیرشبکه (یعنی مجموعهٔ مسیریابهای بین راه) تقاضای سرویس ویژه‌ای برای ارسال یک دیتاگرام می‌نماید. از طریق این فیلد نوع سرویس درخواستی مشخص می‌شود، این فیلد خودش به چند بخش تقسیم شده است :

P2	P1	P0	D	T	R	–	–
تقدم بسته			تأخیر	توان خروجی	قابلیّت اطمینان	بلا استفاده	

الف) سه بیت سمت چپ : اولویت بستهٔ IP را تعیین می‌کند . اگر در این سه بیت صفر قرار گرفته باشد بسته اطلاعاتی از نوع معمولی تلقی می‌شود، یعنی دارای پایین ترین مقدار اولویت است و اگر مقدار ۷ یعنی $(111)_2$ در این سه بیت قرار گرفته باشد بالاترین اولویت برای بسته درنظر گرفته می‌شود.

ب) بیتهای R , T , D : بیت D به معنای تأخیر[2] ، بیت R به معنای قابلیت اطمینان و بیت T به معنای توان خروجی خط[3] است.

اکثر مسیریابهای تجاری فیلد Type of Service را نادیده می‌گیرند و اهمیتی به محتوای آن نمی‌دهند .

---

۱ IP Header Length
۲ Delay
۳ Throughput

فیلد Total Length : در این فیلدِ ۱۶ بیتی عددی قرار می‌گیرد که طول کل بستهٔ IP را که شامل مجموع اندازهٔ سرآیند و ناحیه داده است، تعیین می‌کند. مبنای طول برحسب بایت است و بنابراین حداکثر طول کل بستهٔ IP می‌تواند ۶۵۵۳۵ بایت باشد.

فیلد Identification : همانگونه که قبلاً اشاره شد برخی از مواقع مسیریابها یا ماشینهای میزبان مجبورند یک دیتاگرام را به قطعات کوچکتر بشکنند و ماشین مقصد مجبور است آنها را بازسازی کند ، بنابراین وقتی یک دیتاگرامِ واحد شکسته می‌شود باید مشخصه‌ای داشته باشد تا در هنگام بازسازیِ آن در مقصد بتوان قطعه‌های آن دیتاگرام را از بقیه جدا کرد. در این فیلدِ ۱۶ بیتی عددی قرار می‌گیرد که شمارهٔ یک دیتاگرام واحد را مشخص می‌کند.

فیلد Fragment offset : این فیلد در سه بخش سازماندهی شده است :

الف) بیت DF[1] : با یک شدن این بیت در یک بستهٔ IP هیچ مسیریابی حقّ ندارد آن را قطعه قطعه کند، چرا که مقصد قادر به بازسازی دیتاگرام‌های تکه تکه شده نیست.

ب) بیت MF[2] : این بیت مشخص می‌کند که آیا بستهٔ IP آخرین قطعه از یک دیتاگرام محسوب می‌شود یا باز هم قطعه‌های بعدی وجود دارد. در آخرین قطعه از یک دیتاگرام بیت MF صفر خواهد بود و در بقیه الزاماً ۱ است.

ج) Fragment offset : این قسمت که سیزده بیتی است در حقیقت شماره ترتیب هر قطعه در یک دیتاگرام شکسته شده محسوب می‌شود. با توجه به سیزده بیتی بودن این فیلد، یک دیتاگرام حداکثر می‌تواند به ۸۱۹۲ تکه تقسیم شود.

فیلد Time To Live : این فیلد هشت بیتی در نقش یک شمارنده ، طول عمر بسته را مشخص می‌کند. طول عمر یک بسته بطور ضمنی به زمانی اشاره می‌کند که یک بسته IP می‌تواند بر روی شبکه سرگردان باشد . حداکثر طول عمر یک بسته ، ۲۵۵ خواهد بود که به ازای عبور از هر مسیریاب از مقدار این فیلد یک واحد کم می‌شود. هر گاه یک بستهٔ IP به دلیل بافر شدن در حافظهٔ یک مسیریاب زمانی را معطّل بماند ، به ازای هر ثانیه یک واحد از این فیلد کم خواهد شد. به محض آنکه مقدار این فیلد به صفر برسد بستهٔ IP در هر نقطه از مسیر باشد حذف شده و از ادامهٔ سیرِ آن به سمت مقصد جلوگیری خواهد شد. (البته معمولاً یک پیام هشدار به ماشینی که آن بسته را تولید کرده باز پس فرستاده خواهد شد.) اگرچه بزرگترین عددی که در فیلد طول عمر بسته قرار می‌گیرد ۲۵۵ است ولی در عمل مقداری که سیستمهای عامل در این فیلد قرار می‌دهند چیزی حدود ۳۰ است. (البته می‌توان مقدار پیش فرض آن را عوض کرد)

فیلد Protocol : دیتاگرامی که در فیلد داده از یک بستهٔ IP حمل می‌شود با ساختمان دادهٔ خاصّ از لایهٔ بالاتر تحویلِ پروتکل IP شده تا روی شبکه ارسال شود. بعنوان مثال ممکن است این داده‌ها را پروتکل TCP در لایه بالاتر ارسال کرده باشد و یا ممکن است این کار توسط پروتکل UDP انجام شده باشد. بنابراین مقدار این فیلد شمارهٔ پروتکلی است که در لایه بالاتر تقاضای ارسال یک دیتاگرام کرده است؛ بسته‌ها پس از دریافت در مقصد باید به پروتکل تعیین شده تحویل داده شود.

فیلد Header Checksum : این فیلد که شانزده بیتی است به منظور کشف خطاهای احتمالی در سرآیند هر بستهٔ IP استفاده می‌شود . برای محاسبهٔ کد کشف خطا ، کلّ سرآیند بصورت دو بایت، دوبایت با یکدیگر جمع می‌شود. نهایتاً حاصل جمع به روش "مکمل یک"[1] منفی می‌شود و این عدد منفی در این فیلد از سرآیند قرار می‌گیرد .

---

[1] Don't Fragment

[2] More fragment

در هر مسیریاب قبل از پردازش و مسیریابی ابتدا صحّتِ اطلاعات درون سرآیند بررسی می‌شود و بستهٔ IP فاقد اعتبار حذف خواهد شد.

دقت کنید که فیلد Checksum در هر مسیریاب باید از نو محاسبه و مقداردهی شود زیرا وقتی یک بستهٔ IP وارد یک مسیریاب می‌شود حداقّل فیلد TTL از آن بسته عوض خواهد شد.

فیلد Source Address : هر ماشین میزبان در شبکهٔ اینترنت یک آدرس جهانی و یکتای ۳۲ بیتی دارد. بنابراین هر ماشین میزبان در هنگام تولید یک بستهٔ IP باید آدرس خودش را در این فیلد قرار بدهد .

فیلد Destination Address : در این فیلد آدرس ۳۲ بیتیِ مربوط به ماشین مقصد که باید بسته IP تحویل آن بشود ، قرار می‌گیرد.

فیلد Payload : در این فیلد داده‌های دریافتی از لایهٔ بالاتر قرار می‌گیرد.

مبحث آدرسها در اینترنت و اینترانت

آدرسهای IP درون یک عدد دودویی ۳۲ بیتی درج می‌شوند ولیکن برای سادگی نمایش به چهار قسمت هشت بیتی[۲] تقسیم و بصورت چهار عدد دهدهی که با نقطه از هم جدا شده‌اند ، نوشته می‌شود؛ بعنوان مثال آدرس زیر یک آدرس IP معتبر می‌باشد که در قالب چهار قسمت دهدهی نوشته شده است:

34.21.225.1

این آدرس بصورت زیر در فیلد آدرس از یک بستهٔ IP تنظیم می‌شود:

00100010	00010101	11100001	00000001

کلاسهای آدرسِ IP

با توجه به آنکه اینترنت مجموعه‌ای از شبکه‌های متصل شده به هم می‌باشد، برای آدرس دادن به ماشینهای میزبان بهتر است ۳۲ بیت آدرسِ IP به قسمتهای زیر تقسیم شود:

آدرس ماشین/آدرس زیرشبکه/آدرس شبکه

الف) آدرس شبکه

ب) آدرس زیر شبکه (درصورت لزوم)

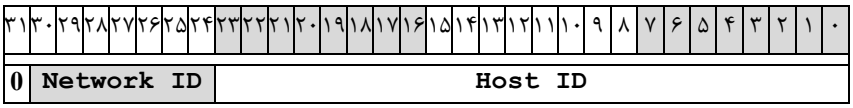
ج) آدرس ماشین میزبان

آدرسهای IP درپنج کلاس E,D,C,B,A معرفی شده‌اند که شما بایستی آنها را بدقت بشناسید و تحلیل کنید. در زیر قالب کلاسهای پنج گانهٔ آدرس IP مشخص شده است:

---

۱ One's Compelement
۲ Octet

آدرسهای کلاس A : قالب ۳۲ بیتیِ آدرس در کلاس A به صورت زیر است:



در کلاس A ، پرارزشترین بیت از آدرس ، مقدار صفر دارد و این بیت ، کلاس A را از دیگر کلاسها متمایز می‌کند؛ ۷ بیت بعدی "مشخصهٔ آدرس شبکه" و سه بایت باقیمانده ، آدرس ماشین میزبان را تعیین می‌کند. بنابراین در کلاس A بایت پرارزش در محدوده صفر تا ۱۲۷ تغییر می‌کند. چون با ۲۴ بیت می‌توان حدود هفده میلیون ماشین میزبان را آدرس دهی کرد ، می‌توان به این نتیجه رسید که آدرسهای کلاس A بایستی برای آژانسهای ستون فقرات اینترنت یا شبکه‌ها بسیار عظیم مثل NSFNet یا ARPANet اختصاص داده شده باشد . مشخصهٔ شبکه در این کلاس بهیچوجه نمی‌تواند اعداد صفر یا ۱۲۷ انتخاب شود چرا که این دو عدد در شبکه معنای دیگری خواهند داشت و بعداً به آن اشاره خواهیم کرد. بنابراین تعداد شبکه‌هائی که در جهان می‌توانند از کلاس A استفاده کنند ۱۲۶ تا خواهد شد که بسیار کم است . امروزه اختصاص آدرسهای کلاس A غیر ممکن است چرا که همهٔ آنها توسط پیشگامان شبکه سالها قبل تملیک شده‌اند.

وقتی به یک آدرس IP که در قالب دهدهی نوشته شده است نگاه می‌کنید براحتی می‌توانید کلاس آنرا تشخیص بدهید . اگر عدد سمت چپ آدرس ، بین صفر تا ۱۲۷ باشد ، آن آدرس از کلاس A خواهد بود:

74. 103 .14 .138

Net ID Host ID

آدرس IP معادل با (**127.0.0.0**) در پروتکل IP ، یک شبکه را تعیین نمی‌کند بلکه بصورت قراردادی بعنوان آدرس **"حلقه بازگشت"**[۱] جهت اهداف اشکال زدایی استفاده شده است چرا که این آدرس عملاً معادل آدرس خود ماشین محلی است .

آدرسهای کلاس B : قالب ۳۲ بیتیِ آدرس در کلاس **B** به صورت زیر است:

هر گاه دو بیت پرارزش از آدرس IP مقدار 10 داشته باشد آن آدرس از کلاس B خواهد بود. ۱۴ بیت باقیمانده از ۲ بایت سمت چپ ، آدرس شبکه را تعیین می‌کند و دو بایت اول از سمت راست (۱۶ بیت) آدرس ماشین میزبان خواهد بود.

در آدرسهای کلاس B ، تعداد ۱۶۳۸۲ (2^{14}-2) شبکهٔ گوناگون قابل تعریف خواهد بود و هر شبکه می‌تواند ۶۵۵۳۴ (2^{16}-2) ماشین میزبان تعریف نماید. اختصاص آدرسهای کلاس B برای شبکه‌های بسیار عظیم مناسب است . هر چند تعداد این شبکه در جهان می‌تواند تا حدود شانزده هزار عدد باشد ولیکن امروزه عملاً نمی‌توان آدرس کلاس B گرفت چرا که تقریباً همهٔ آنها آن تخصیص داده شده‌اند .

اگر آدرس IP به صورت دهدهی نوشته شود و عدد سمت چپِ آن بین ۱۲۸ تا ۱۹۱ باشد ، آن آدرس ، کلاس B خواهد بود:

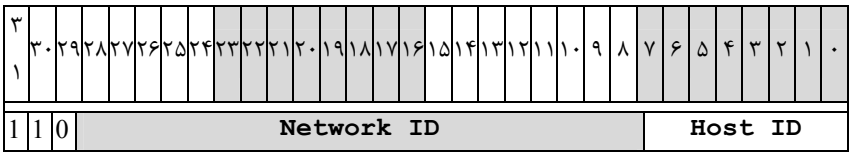
134. 64. 143 . 24

Net ID Host ID

---

[۱] Loopback

آدرس کلاس C : قالب ۳۲ بیتیِ آدرس در کلاس **C** به صورت زیر است:

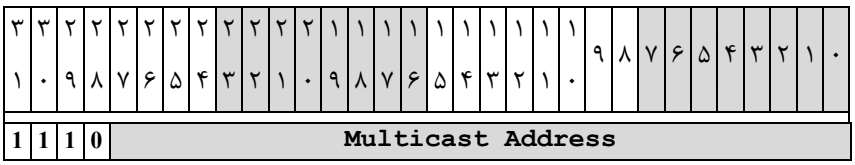


کلاس C مناسب ترین و پرکاربردترین کلاس از آدرس‌های IP است . همانگونه که از شکل مشخص است در این کلاس ، سه بیت پرارزش دارای مقدار 110 است و ۲۱ بیت بعدی از سه بایت سمت چپ برای تعیین آدرس شبکهٔ مورد نظر بکار رفته است . بنابراین در این کلاس می‌توان حدود دو میلیون شبکه را در جهان آدرس دهی کرد و هر شبکه می‌تواند تا ۲۵۴ عدد ماشین میزبان تعریف نماید . برای تشخیص آدرس‌های کلاس C به عدد سمت چپ از آدرس IP که به صورت دهدهی نوشته شده است نگاه کنید. اگر عدد بین ۱۹۲ تا ۲۲۳ بود آن آدرس از کلاس C خواهد بود:

199. 164. 78 .132

Net ID Host ID

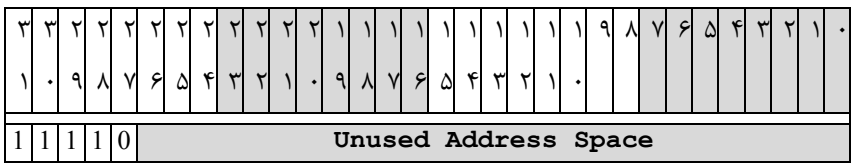
آدرس کلاس D : قالب ۳۲ بیتیِ آدرس در کلاس **D** به صورت زیر است:



در این کلاس ، چهار بیت پرارزش دارای مقدار 1110 است و ۲۸ بیت باقیمانده از کلّ آدرس برای تعیین آدرسهای "چند مقصده"[۱] (آدرسهای گروهی) است .

از این آدرسها برای ارسال یک دیتاگرام به طور همزمان برای چندین ماشین میزبان کاربرد دارد و بمنظور عملیات رسانه ای و چند پخشی بکار می‌رود. توضیح بیشتر در مورد این کلاس در بخشی مجزا ارائه خواهد شد .

آدرس کلاس E : قالب ۳۲ بیتیِ آدرس در کلاس **E** به صورت زیر است:



فعلاً این دسته از آدرسها که پنج بیت پرارزش آنها در سمت چپ 11110 است کاربرد خاصّی ندارند و برای استفاده در آینده بدون استفاده رها شده‌اند. البته گاهی بصورت آزمایشی از این آدرسها استفاده شد ولی تاکنون جهانی نشده‌اند .

[۱] Multicast

آدرسهای خاصّ

در بین تمامی کلاسهای آدرسIP پنج گروه از آدرسها ، معنای ویژه ای دارند و با آنها نمی‌توان یک شبکهٔ خاص را تعریف و آدرس دهی کرد. این پنج گروه آدرس عبارتند از:

الف) آدرس 0.0.0.0 : هر ماشین میزبان که از آدرس IP خودش مطلع نیست این آدرس را بعنوان آدرس خودش فرض می‌کند. البته از این آدرس فقط به عنوان آدرس مبداء و برای ارسال یک بسته می‌توان استفاده کرد و گیرندهٔ بسته نمی‌تواند پاسخی به مبداء بسته برگرداند.

ب) آدرس 0.HostID : این آدرس زمانی به کار می‌رود که ماشین میزبان ، آدرس مشخصهٔ شبکه‌ای که بدان متعلق است را نداند. در این حالت در قسمت NetID مقدار صفر و در قسمت HostID شمارهٔ مشخصهٔ ماشین خود را قرار می‌دهد.

ج) آدرس 255.255.255.255 : برای ارسال پیامهای فراگیر برای تمامی ماشینهای میزبان بر روی شبکهٔ محلی که ماشین ارسال کننده به آن متعلق است .

د) آدرس NetID.255 : برای ارسال پیامهای فراگیر برای تمامی ماشینهای یک شبکهٔ راه دور که ماشین میزبان فعلی متعلق به آن نیست. آدرس شبکهٔ مورد نظر در قسمت NetID تعیین شده و تمامی بیتهای قسمت مشخصهٔ ماشین میزبان ۱ قرار داده می‌شود. البته بسیاری از مسیریابها برای مصون ماندن شبکه از مزاحمتهای بیرونی ، چنین بسته‌هایی را حذف می‌کنند.

ه) آدرس 127.xx.yy.zz : این آدرس بعنوان "آدرس بازگشت" شناخته می‌شود و آدرس بسیار مفیدی برای اشکالزدایی از نرم افزار می‌باشد. به عنوان مثال اگر بسته‌ای به آدرس 127.0.0.1 ارسال شود ، بسته برای ماشین تولیدکنندهٔ آن بر خواهد گشت[۱]؛ در این حالت اگر نرم افزارهای TCP/IP درست و بدون اشکال نصب شده باشد فرستندهٔ بسته باید آنرا مجدداً دریافت کند. همچنین از این آدرس می‌توان برای آزمایش برنامه‌های تحت شبکه ، قبل از نصب آنها بر روی ماشینهای میزبان استفاده کرد.

آدرسهای زیرشبکه

در ادامهٔ بحث بایستی مسئله زیر شبکه را در خصوص آدرس دهی‌ها مطرح نمائیم . مبحث را با یک مثال آغاز می‌نمائیم:

فرض کنید دانشگاه شما یک کلاس C با قابلیت تعریف ۲۵۴ ماشین میزبان ثبت می‌نماید (مثلاً 211.11.121.0) ؛ یعنی شبکهٔ دانشگاه توانایی آدرس دهی ۲۵۵ ایستگاه را در شبکه دارد . در نظر بگیرید که دانشگاه دارای یک شبکهٔ محلی واحد و یکپارچه برای کل دانشگاه نیست بلکه دارای هشت شبکهٔ محلی مجزا است که برای هر دانشکده تهیه دیده شده است؛

برای آنکه بتوان زیرشبکه‌ها[۲] را تفکیک کرد جدای از قسمت آدرس شبکه که کل شبکهٔ دانشگاه شما را مشخص می‌کند بایستی در قسمت مشخصهٔ ماشین میزبان نیز به گونه ای زیر شبکه‌ها مشخص شوند . این کار از طریق مفهومی به نام **"الگوی زیرشبکه"**[۳] انجام می‌شود.

---

[۱] این آدرس همانند آنست که فرستندهٔ یک بستهٔ پستی آدرس دقیق خودش را به عنوان گیرندهٔ آن درج نماید. بنابراین با آدرس 0.0.0.0 تفاوت ذاتی دارد.

[۲] Subnetworks

[۳] Subnet Mask

شما با نگاه اول به اولین عدد سمت چپ متوجه خواهید شد که این آدرس از چه کلاسی است ولی هنوز موارد مبهمی وجود دارد : آیا شبکه ای که آدرس آنرا پیش رو دارید فقط یک شبکه است یا خودش زیر شبکه بندی شده است؛ یعنی از چند شبکهٔ محلی متصل بهم تشکیل شده است؟

تمامی ماشینهای میزبان برای تشخیص محل مقصد یک بستهٔ IP در شبکه احتیاج به یک مشخصهٔ دیگر دارند و آن **"الگوی زیرشبکه"** نامیده می‌شود.

الگوی زیرشبکه یک عدد ۳۲ بیتی دودویی است که برای ماشین میزبان نقش یک مقایسه‌گر را بازی می‌کند تا با استفاده از آن بتواند تشخیص دهد که آیا مقصد روی همین شبکه محلی است که خودش به آن تعلق دارد یا روی شبکه دیگری است .

فرآیند استفاده از "الگوی زیرشبکه" را با استفاده از مثال قبل ولی با آدرس کلاس B آموزش می‌دهیم:

فرض کنید شما کاربری روی یک ایستگاه در شبکه دانشگاه خودتان هستید, آدرس IP متعلق به دستگاه شما بصورت زیر اختصاص داده شده است :

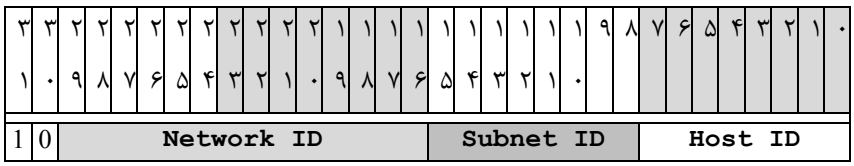
131.55.213.73

با یک نگاه متوجه می‌شوید که آدرس از کلاس B است که مشخصهٔ شبکهٔ آن معادل 131.55.0.0 و مشخصهٔ ماشین شما 0.0.213.73 است؛ ولی هنوز نمی‌دانید شبکه ای که مشخصهٔ آن معادل 131.55 است آیا زیر شبکه دارد یا خیر؟

فرض کنید که دانشگاه شما با آدرس شبکه 131.55.0.0 ، می‌خواهد حداکثر دارای ۲۵۴ زیر شبکه باشد ، بهمین دلیل فرض کرده است که در فیلد مشخصهٔ ماشین میزبان (Host ID) که در کلاس B دو بایت سمت راست را شامل میشود ، بایت دومِ آن به عنوان مشخصهٔ مربوط به زیر شبکه تعریف شود. یعنی فیلد دوبایتی مربوط به مشخصهٔ ماشین میزبان به دو بخش تقسیم شده است:

الف) مشخصهٔ زیرشبکه

ب) مشخصهٔ ماشین میزبان



ماشین شما تصمیم دارد بسته ای را برای یک ماشین میزبان با آدرس IP معادل 131.55.108.75 بفرستد؛ ماشین از کجا می‌تواند بفهمد که مقصد روی همین شبکه محلی که شما بدان متعلق هستید واقع است یا آنکه به شبکهٔ محلی در یک دانشکدهٔ دیگر متعلق است. دانستن این موضوع بسیار با اهمیت خواهد بود چرا که اگر ماشین میزبان مورد نظر روی شبکهٔ دیگری باشد بسته باید با آدرس فیزیکی "مسیریاب پیش فرض"[۱] روی کانال ارسال شود. بنابراین تمام ماشینهای روی شبکه بایستی از وضعیت زیر شبکه‌ها مطلع باشند .

با توجه به آنچه که در بالا اشاره شد دومین بایت از سمت راست بعنوان مشخصهٔ زیر شبکه اختصاص داده شده است و بهمین دلیل هر ماشین برای دانستن آنکه آیا ماشین مقصد در شبکه محلی خودش واقع است یا در خارج از شبکه قرار دارد باید قسمت "مشخصهٔ شبکه" و "مشخصهٔ زیرشبکه" از آدرس IP خودش را با همین مشخصه‌ها از آدرس مقصد مقایسه نماید.

---

[۱] Default Gateway

اینجاست که یک الگوی ۳۲ بیتی تعریف می‌شود که یک عدد ۳۲ بیتی و در این مثال بصورت 255.255.255.0 است:

۳۱	۳۰	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰

هر گاه ماشین بخواهدیک آدرس IP را تحلیل کند . الگوی فوق را با آدرس IP خودش AND می‌کند. (با اینکار در حقیقت Host ID خودش را صفر می‌نماید) سپس مجدداً الگو را با آدرس IP مقصد AND می‌کند (مشخصهٔ ماشین مقصد هم صفر می‌شود) حال نتیجه دو مرحله را با هم مقایسه می‌نماید . اگر نتیجه دو مرحله یکسان بود ، هم مشخصهٔ شبکه و هم مشخصهٔ زیرشبکه از آدرسهای مبدأ و مقصد یکی است و هر دو روی یک شبکهٔ محلی قرار دارند. در صورت عدم تساوی ، ماشین مبدأ به این نتیجه می‌رسد که مقصد مورد نظر روی شبکه محلی خودش نیست و آن بسته بایستی به آدرس فیزیکی مسیریابِ پیش فرض ارسال شود.

فرض کنید بسته ای با آدرسهای مشخص زیر بخواهد ارسال شود:

آدرس ماشین مبدأ: 131.55.213.73

آدرس ماشین مقصد: 131.55.108.75

الگوی زیرشبکه: 255.255.255.0

حاصل AND این دو آدرس با الگوی زیرشبکه مساوی نیستند و بنابراین ماشین مبدأ متوجه خواهد شد که ماشین مقصد روی شبکه محلی خودش نیست و بسته اطلاعاتی را بایستی به آدرس فیزیکی مسیریاب پیش فرض ارسال نماید.

به عنوان مثالی دیگر فرض کنید ماشین مبدأ می‌خواهد برای ماشینی با آدرس IP زیر بسته ای را ارسال نماید:

131.55.213.84

حاصل AND آدرس ماشین مبدأ و مقصد فوق با الگوی زیرشبکه مساوی هستند و بالطبع مبدأ و مقصد روی یک شبکه محلی واقعند و هیچ لزومی ندارد که ارسال بسته به آدرس فیزیکی مسیریاب پیش فرض انجام شود ، بلکه باید مستقیماً از آدرس فیزیکی ایستگاه مقصد استفاده شود .

الگوی زیرشبکه در مثالهای بالا ساده‌ترین حالت بود که به آنها **"الگوی زیرشبکهٔ استاندارد"**[۱] گفته می‌شود چرا که الگوها دقیقاً مبتنی بر تقسیمات هشت‌بیتی هستند.

زیر شبکه‌های غیر استاندارد

الگوهای زیرشبکه برای تقسیم فضای آدرس‌دهی در شبکه‌های کلاس A ، B و C به تعدادی زیرشبکه ، تعریف می‌شوند. در مثالهائی که بررسی کردیم الگوی شبکه بصورت زیر تعریف شده بود:

255.255.255.0

حال الگوی زیر را در نظر بگیرید:

255.255.240.0

---

[۱] Standard Subnet Mask

عدد ۲۴۰ در الگوی زیرشبکه چه چیزی را تعریف می‌کند؟ به فرم دودویی الگوی بالا دقت کنید:

۳۱	۳۰	۲۹	۲۸	۲۷	۲۶	۲۵	۲۴	۲۳	۲۲	۲۱	۲۰	۱۹	۱۸	۱۷	۱۶	۱۵	۱۴	۱۳	۱۲	۱۱	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	۰
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰

بفرض اگر الگوی بالا برای زیر شبکه بندی آدرس کلاس B به کار رفته باشد ، نشان دهنده آن است که چهار بیت پرارزش از بایت دوم برای تعیین شماره زیر شبکه به کار رفته است و ۱۲ بیت باقیمانده بعنوان "مشخصهٔ ماشین میزبان" استفاده شده است؛ بدین معنا که با این الگو می‌توان ۱۴ زیر شبکه (2^4-2) تعریف کرد بگونه ای که در هر زیرشبکه ۴۰۹۴ ، (2^{12}-2) ماشینِ میزبان قابل آدرس دهی خواهد بود.

CIDR[1] : مسیریابی بر اساس آدرس‌های بدون کلاس

برای بسیاری از موسسات یک شبکه کلاس A با شانزده میلیون آدرس، بسیار بزرگ و بی‌مصرف و کلاس C با 256 آدرس بسیار کوچک و ناکافی است؛ فقط شبکه کلاس B با 65536 آدرس مناسب است. این مشکل در فرهنگ رایج اینترنت به نام مشکل « سه خرس » مشهور شده است. (برگرفته از داستان موطلایی و سه خرس)

حقیقت آن‌است که حتی کلاس B هم برای بسیاری از سازمان‌ها و موسسات، بسیار بزرگ است. بررسی‌ها نشان داده که بیش از نیمی از شبکه‌های کلاس B، کمتر از 50 ماشین دارند! برای چنین شبکه‌هایی استفاده از کلاس C کفایت می‌کند ولی بی‌تردید تمام موسساتی که متقاضی کلاس B بوده‌اند بدان می‌اندیشیده‌اند که روزی فیلد هشت بیتی کلاس C برای شبکه آن ها کافی نخواهد بود.

راه‌حل نهایی و پیاده شده در اینترنت که توانست به اینترنت اجازه نفس کشیدن بدهد، روش CIDR (مسیریابی بر اساس آدرس‌های بدون کلاس) بود. ایدهٔ اصلی در CIDR که در سند RFC 1519 تشریح شده آن است که آدرس‌های IP بدون در نظر گرفتن کلاس و به صورت بلوک‌هایی با طول متغیر تخصیص یابد. مثلاً اگر یک سایت نیاز به 2000 آدرس داشته باشد یک بلوک آدرس 2048 تایی به او داده می‌شود.

حذف کلاس‌های آدرس، فرآیند هدایت بسته‌ها را پیچیده‌تر می‌کند. در سیستم مبتنی بر کلاس، فرآیند هدایت بدین نحو بود که وقتی بسته‌ای به یک مسیریاب می‌رسید، یک کپی از آدرس IP به اندازه 28 بیت به راست شیفت داده می‌شد تا فقط چهار بیت سمت چپ آدرس (که کلاس آدرس را مشخص می‌کند) باقی بماند. بر اساس این چهار بیت (16 حالت مختلف) بسته‌ها در یکی از کلاس‌های A و B و C (و D در صورت پشتیبانی از آن) مرتب می‌شدند. (از این 16 حالت مختلف، هشت حالت برای کلاس A است ـ 0xxx ـ چهار حالت برای کلاس B ـ 10xx ـ ، دو حالت برای کلاس C ـ 110x ـ و دو حالت برای کلاس‌های D و E است.) پس از تشخیص کلاس آدرس، برای به‌دست آوردن شماره شبکه، آدرس IP با یکی از الگوهای -8، -16، -24 به صورت بولی AND و بخش شماره ماشین حذف می‌شد. سپس شماره شبکه در هر یک از جدول مربوط به آدرس‌های کلاس A , B و C جستجو می‌شد. جداول مسیریابی برای کلاس‌های A و B بر حسب شماره شبکه ایندکس شده بودند.[2] درعوض جداول مسیریابی برای کلاس C مبتنی بر روش جداول Hash (Hash Table) پیاده

[1] CIDR: Classless InterDomain Routing

[1] به عبارت ساده به دلیل کم بودن تعداد شبکه‌ها جداول مسیریابی برای کلاس‌های A و B در ساختمان داده‌ای شبیه به آرایه ذخیره می‌شد. – م

شده بود. پس از آن‌که درایه متناظر با آدرس شبکه در یکی از این جداول پیدا می‌شد خط خروجی متناسب با آن شبکه مشخص شده و بسته بر روی آن خط هدایت می‌گردید.

در CIDR این الگوریتم ساده، کار نخواهد کرد. درعوض به هر یک از درایه‌های جدول مسیریابی یک فیلد 32 بیتی جدید افزوده شده که الگوی آن را [از طریق یک MASK سی و دو بیتی] مشخص می‌کند. بدین‌ترتیب برای تمام شبکه‌ها فقط یک جدول مسیریابی یکتا وجود دارد که در حقیقت یک آرایه ستونی متشکل از آدرس IP، الگوی زیرشبکه (Subnet Mask) و خط خروجی است. وقتی بسته‌ای وارد می‌شود ابتدا آدرس IP آن استخراج می‌شود. سپس جدول مسیریابی درایه به درایه (Entry by Entry) جستجو و آدرس مقصد بسته پس از AND شدن با الگوی زیر شبکه از هر درایه با آدرس IP از آن درایه مقایسه می‌شود. این فرآیند آن‌قدر تکرار می‌گردد تا به موارد مطابقت برسد. این امکان وجود دارد که چندین درایه با یک آدرس IP مطابقت داشته باشد (به دلیل طول متفاوت ا لگوهای زیر شبکه). در این حالت درایه‌ای که طول الگوی زیر شبکه آن از همه بزرگتر است از بین آن‌ها انتخاب می‌شود. به عبارتی اگر دو مورد مطابق با طول الگوی 20/(255.255.240.0) و الگوی 24 / (255.255.255.0) پیدا شود، درایه دوم انتخاب می‌شود.

برای سرعت بخشیدن به فرآیند جستجو و مطابقت، الگوریتم‌های پیچیده‌ای ابداع شده است. (Ruiz – Sanches et al .2001) مسیریاب‌های تجاری در بازار امروز از تراشه‌های VLSI خاصی بهره گرفته اند که الگوریتم مذکور را به صورت یک « سخت‌افزار درون‌کار» (Embedded Hardware) پیاده‌سازی کرده‌اند.

برای آن‌که فهم فرآیند هدایت بسته‌ها در CIDR را ساده‌ترکنیم مثالی را مدنظر قرار بدهید که در آن میلیون‌ها آدرس تعریف شده است و آدرس شروع 194.24.0.0 است. فرض کنید که دانشگاه کمبریج به 2048 آدرس نیاز دارد و آدرس‌های 194.24.0.0 تا 194.24.7.255 به آن اختصاص داده شده است. (الگوی زیر شبکه نیز 255.255.248.0 است). بعداً دانشگاه آکسفورد تقاضای 4096 آدرس IP می‌دهد. از آنجایی که بلوک‌های آدرس 4096 تایی باید در مرز 4096 بایتی قرار بگیرد نمی‌توان آدرس‌هایی که از 194.24.8.0 شروع می‌شود را به آن اختصاص داد. درعوض آدرس اختصاص داده شده به او در محدودهٔ 194.24.16.0 تا 194.24.31.255 و با الگوی 255.255.240.0 خواهد بود. در این‌جا دانشگاه ادینبورو تقاضای 1024 آدرس داده و فضای 194.24.8.0 تا 194.24.11.255 با الگوی 255.255.252.0 به او تعلق می‌گیرد. این انتساب‌ها در جدول زیر خلاصه شده‌اند.

الگوی نمایش	تعداد آدرس	آخرین آدرس	اولین آدرس	دانشگاه
194.24.0.0/21	2048	194.24.7.	194.24.0.0	Cambridge
194.24.8.0/22	1024	194.24.11.255	194.24.8.0	Edinburgh
194.24.12/22	1024	194.24.15.255	194.24.12.0	در دسترس و آزاد
194.24.16.0/20	4096	194.24.31.255	194.24.16.0	Oxford

انتساب آدرس‌های IP

حال جداول مسیریابی در تمام مسیریاب‌های واقع بر ستون فقرات اینترنت در جهان باید با این سه درایه جدید به هنگام شود. هر درایه یک آدرس مبنا و یک الگوی زیر شبکه است. این درایه‌ها در مبنای دو عبارتند از:

	آدرس				الگوی زیر شبکه (Subnet Mask)			
C:	11000010	00011000	00000000	00000000	11111111	11111111	11111000	00000000
E:	11000010	00011000	00001000	00000000	11111111	11111111	11111100	00000000
O:	11000010	00011000	00010000	00000000	11111111	11111111	11110000	00000000

حال ببینیم وقتی که بسته‌ای با آدرس 194.24.17.4. وارد یک مسیریاب می‌شود چه اتفاقی می‌افتد. این آدرس به صورت دودویی عبارت است از:

11000010 00011000 00010001 00000100

ابتدا این آدرس با الگوی زیر شبکه کمبریج، AND می‌شود و نتیجه زیر به‌دست می‌آید:

11000010 00011000 00010000 00000000

این مقدار با آدرس مبنای دانشگاه کمبریج مطابقت ندارد. حال مجدداً آدرس اصلی با الگوی زیر شبکه دانشگاه ادینبورو AND شده و نتیجه زیر به‌دست می‌آید:

11000010 00011000 00010000 00000000

این مقدار نیز با آدرس مبنای دانشگاه ادینبورو تطابق ندارد و همین کار برای دانشگاه آکسفور تکرار شده مقدار زیر به‌دست می‌آید:

11000010 00011000 00010000 00000000

این مقدار با آدرس مبنای دانشگاه آکسفورد مطابقت دارد. اگر هیچ مورد تطبیق دیگری در جدول یافت نشد بسته بر روی خطی ارسال می‌شود که در درایه متناظر با شبکه دانشگاه آکسفورد درج شده است.

حال اجازه بدهید، آدرس این سه دانشگاه را از دید یک مسیریاب در نبراسکای اوهاما بررسی کنیم. این مسیریاب چهار خط به مینیاپولیس، نیویورک، دالاس و دنور دارد. وقتی نرم‌افزار مسیریاب اوهاما، این سه درایه جدید را جهت درج در جدول مسیریابی خود دریافت می دارد، متوجه می‌شود که قادر است هر سه تای آن‌ها را در یک « درایه واحد و تجمیع شده» (Aggregate Entry) به صورت 194.24.0.0/19 ادغام نماید. [1] آدرس و الگوی زیر شبکه در مبنای دو به صورت زیر است:

11000010 00000000 00000000 00000000 11111111 11111111 11100000 00000000

طبق این درایه تمام بسته‌هایی که به مقصد یکی از این سه دانشگاه روانه شده‌اند به سوی نیویورک هدایت می‌شوند. با تجمیع این سه درایه، مسیریاب اوهاما توانسته به میزان دو درایه حجم جدول خود را کاهش بدهد.

به همین ترتیب اگر مسیریاب نیویورک برای تمام ترافیک منتهی به انگلستان فقط یک خط به لندن داشته باشد او نیز سه درایه فوق را در یک درایه ادغام می‌کند ولیکن اگر برای لندن و ادینبورو دو خط مجزا داشته باشد باید هر سه تای آن‌ها را به‌طور مجزا در جدول ذخیره کند. عمل تجمیع (Aggregation) در اینترنت به‌طور گسترده‌ای مورد استفاده قرار گرفته تا حجم جداول مسیریابی کاهش یابد.

آخرین نکته در این مثال آن است که بر طبق درایه ادغام شده در جدول مسیریابی مسیریاب واقع در اوهاما حتی بسته‌هایی که به آدرس اختصاص داده نشده روانه هستند [یعنی آدرس‌های بین 194.24.12.0 تا 194.24.15.255] نیز به سوی نیویورک هدایت می‌شوند. مادامی که این آدرس‌ها به کسی اختصاص داده نشده هیچ مشکلی به وجود نمی‌آید چرا که بنا نیست بسته‌هایی با این آدرس‌ها تولید شوند. ولی اگر این بلوک آدرس، به شرکتی در کالیفرنیا داده شود باید درایه‌ای جدید به شکل 194.24.12.0/22 در جدول مسیریابی تمام مسیریاب‌ها درج شود تا بسته‌هایی به مقصد این شبکه نیز به درستی مسیریابی شوند.

[1] از آن جهت امکان تجمیع این سه آدرس وجود داشته که بسته‌هایی که مقصدشان هر یک از این سه دانشگاه است باید بر روی خط خروجی یکسان بروند. – م

پروتکل ICMP[۱]

پروتکل IP ، پروتکلی "بدون اتصّال"[۲] و "غیر قابل اعتماد"[۳] است! بدون اتصّال بدین معنا که مسیریاب هر بسته را بدون هیچگونه هماهنگی با مقصد بسته یا مسیریاب بعدی ارسال می‌نماید ، بدون آنکه بتواند اطلاعی از وجود یا عدم وجود مقصد داشته باشد. در ضمن هر مسیریاب پس از ارسال یک بسته آنرا فراموش می‌کند و منتظر "پیام دریافت بسته"[۴] از گیرندهٔ آن نخواهد ماند. اگر یک بستهٔ IP با خطا به مقصد برسد و یا اصلا به مقصد نرسد این پروتکل هیچ اطلاعی در مورد سرنوشت آن به فرستندهٔ بسته نمی‌دهد.

دلایل مختلفی برای نرسیدن یک بسته به مقصد وجود دارد: ممکن است "زمان حیات[۵]" بسته قبل از رسیدن به مقصد منقضی شود؛ ممکن است مسیر یاب بسته را به مسیری اشتباه هدایت کند؛ ممکن است در هنگام قطعه قطعه کردن بسته و ارسال آنها ، یکی از قطعات دچار خطا شود یا به هر دلیلی به مقصد نرسد بنابراین کل دیتاگرام قابل بازسازی نخواهد بود؛ ممکن است مقصدِ بسته آمادگی دریافت بسته را نداشته باشد یا اصلاً وجود خارجی نداشته باشد. در هنگام بروز هرگونه خطا ، پروتکل IP به فرستندهٔ بسته هیچ اطلاعی در مورد سرنوشت آن نخواهد داد .

عدم گزارشِ خطا به تولید کنندهٔ یک بسته منجر به تکرار خطا و حمل بیهوده و زائد بسته‌هایی میشود که محکوم به فنا و حذف در شبکه هستند. به عنوان مثال عدم گزارش در مورد آماده نبودن مقصد برای دریافت بسته باعث خواهد شد که فرستندهٔ آن اقدام به ارسال بسته‌های دیگر کند در حالی که این کار بی ثمر خواهد بود و فقط بار ترافیک شبکه را افزایش می‌دهد و حتی می‌تواند منجر به بروز "ازدحام"[۶] شود.

پروتکلICMP در کنار پروتکل IP ، برای بررسی انواع خطا و ارسال پیام برای مبدأ بسته در هنگام بروز اشکالات ناخواسته استفاده می‌شود. در حقیقت ICMP یک سیستم گزارش خطا است که بر روی پروتکلIP نصب می‌شود تا در صورت بروز هرگونه خطا به فرستندهٔ بسته پیام مناسب را بدهد تا آن خطا تکرار نشود. در واقع ICMP وظیفه ای در قبال وقوع خطا ندارد بلکه فقط پیامی که بیانگر بروز خطا و نوع آن است به فرستنده برمیگرداند . این پروتکل اشکالات موجود را در قالب یکسری پیام گزارش می‌کند که این پیام خود در یک بسته IP قرار می‌گیرد که از جانب یک مسیریاب یا ماشین مقصد به آدرس فرستنده باز می‌گردد.

پروتکل ARP[7]

نکته ظریفی که در مورد شبکه اینترنت وجود دارد آن است که اگر چه تمامی ماشینهای میزبان و ابزارهای شبکه ای از آدرس IP که آدرس منحصر به فرد و یکتا است استفاده می‌کنند ولیکن یک بسته IP فقط در لایه شبکه قابل شناسائی و تحلیل است. یک بسته IP قبل از ارسال روی کانال از لایه اول یعنی لایهٔ فیزیکی عبور می‌کند و ضمن اضافه شدن اطلاعات لازم و تشکیل یک فریم ، روی کانال فیزیکی ارسال می‌شود . بعبارت روشنتر بسته IP قبل از ارسال درون فیلد داده از فریمی قرار می‌گیرد که بعداً در لایه اول تشکیل می‌شود؛ لایه اول وظیفه ای در قبال مسیریابی و کارهائی از این قبیل ندارد و فقط با آدرسهای فیزیکی کار می‌کند . بعنوان مثال اگر ماشین شما بخواهد بسته ای را برای ماشینی که روی شبکه محلی خودتان واقع است بفرستد، در لایه اول الزاماً بایستی آدرس فیزیکی ماشین شما (مبداء) و

---

۱ Internet Control Message Protocol
۲ Connectionless
۳ Unreliable
٤ Acknowledgement Message
٥ Time To Live
٦ Congestion

۷ Address Resolution Protocol

آدرس فیزیکی ماشین طرف مقابل (مقصد) معین باشد . (این آدرسها بصورت سخت افزاری در کارت شبکه درج شده است) عدم دانستن آدرسهای فیزیکی عملاً مساوی عدم توانایی برای ارتباط خواهد بود چرا که روی کانال انتقال آدرسهای IP بی معنا هستند .

وظیفه پروتکل ARP در اینجا آن است که یک "بسته فراگیر"[۱] روی کل شبکه محلی منتشر کند که این بسته در حقیقت سوال می‌کند:

" کسی که آدرس IP او فلان است ، آدرس فیزیکی او چیست؟"

با توجه به آنکه بسته‌های فراگیر توسط تمامی ماشینهای روی شبکهٔ محلی دریافت می‌شود ، ماشینی که آدرس IP خودش را درون این بسته می‌بیند ، بدان پاسخ می‌دهد و آدرس فیزیکی خود را برای ارسال کنندهٔ آن بسته می‌فرستد. پس از آنکه آدرس فیزیکیِ مقصد بدست آمد ، یک فریم اترنت ساخته شده بر روی کانال منتقل می‌شود.

پروتکل RARP[۲]

پروتکل ARP برای یافتن آدرس‌های فیزیکی ایستگاههائی است که آدرس IP خود را می‌دانند. پروتکل RARP دقیقاً عکس پروتکل ARP عمل می‌کند . گاهاً اتفاق می‌افتد که ایستگاه آدرس فیزیکی مورد نظرش را میداند ولیکن آدرس IP آنرا نمی‌داند؛ این قضیه برای ایستگاههائی که بدون دیسکند و از طریق سرویس دهنده بوت می‌شوند صادق است .

در این پروتکل برای شناسایی آدرس IP متناظر با یک آدرس فیزیکی یک بسته فراگیر روی خط ارسال می‌شود که در آن آدرس فیزیکی یک ایستگاه قرار دارد . تمامی‌ایستگاههایی که از پروتکل RARP حمایت می‌کنند و بسته‌های مربوطه را تشخیص می‌دهند ، در صورتی که آدرس فیزیکی خودشان را درون بسته ببینند در پاسخ به آن ، آدرس IP خود را در قالب یک بستهٔ RARP Reply برمی‌گردانند. بعنوان مثال فرض کنید ایستگاهی با قرار دادن بستهٔ RARP و آدرس ۶ بایتی اترنت 14-04-D5-C8-01-25 روی خط ، آدرس IP آنرا طلب می‌کند . هر ماشین که آدرس IP متناظر با آن را می‌داند به این بسته RARP پاسخ می‌دهد.

دقت کنید که بسته‌های RARP, ARP از نوع "فراگیر محلی"[۳] هستند و بالطبع توسط مسیریابها منتقل نمیشوند و فقط در محدوده شبکه محلی عمل می‌کنند. (کلاً بسته‌هایی که درون فریم لایهٔ فیزیکی قرار می‌گیرند -کپسوله می‌شوند- ، فقط قادرند در محدودهٔ شبکهٔ محلی بصورت فراگیر و همگانی ارسال شوند و این بسته‌ها توسط مسیریاب هدایت نخوتهد شد.)

پروتکل BootP

با توجه به آنچه که در مورد RARP گفته شد بسته‌های سوال کننده آدرس IP از نوع محلی هستند و بالطبع این گونه بسته‌ها از مسیریابها به خارج از شبکه منتقل نخواهد شد.

گاهی نیاز است که یک آدرس IP روی چند شبکه محلی جستجو شود که در این حالت RARP جوابگو نیست. (این نیاز برای ایستگاههای بدون دیسک بوجود می‌آید چرا که پس از روشن شدن بایستی از طریق سرویس دهندهٔ شبکه[۴] بوت شوند)

پروتکل BOOTP در چنین محیطهایی کاربرد دارد و از دیتاگرام‌های نوع UDP که در آینده به آنها خواهیم پرداخت استفاده می‌کند و مسیریابها موظف به انتقال آنها هستند . در این پروتکل نکته جالبی وجود دارد و آن هم آنست که در پاسخ به چنین بسته‌هائی به غیر از آدرس IP ایستگاه مورد نظر، اطلاعات لازم جهت بوت شدن سیستم و همچنین "الگوی زیرشبکه" برای ایستگاه تقاضا کننده که احتمالاً یک ایستگاه بدون دیسک است در قالب یک بستهٔ UDP ارسال خواهد شد.

---

[۱] Broadcast

[۲] Reverse Address Resolution Protocol

[۳] Local Broadcast

[۴] Network Server