۳ لایه پیوند داده

همانطور که در فصل قبل بیان شد. لایه فیزیکی در سمت فرستنده داده‌های دریافتی از لایه پیوند داده را بدون توجه به محتوا و از طریق کانال انتقال به صورت بیت به بیت به سمت گیرنده ارسال می‌کند و لایه فیزیکی گیرنده با دریافت داده‌ها، آنها را بدون توجه به محتوا به لایه پیوند داده تحویل می‌دهد. در واقع لایه فیزیکی وظیفه دارد داده‌هایی که لایه پیوند داده مشخص می‌کند، ارسال کند.

قبل از ارسال اطلاعات، لایه پیوند داده، آدرس فیزیکی مقصد را مشخص می‌کند تا داده‌ها از طریق کانال انتقال (لایه فیزیکی) به این آدرس ارسال شوند. همانطور که گفته شد، داده به شکل بیت‌های پشت سرهم ارسال می‌شوند، از وظایف لایه پیوند داده تعیین ابتدا و انتهای هر فریم است یعنی به شکلی مرز ابتدا و انتهای فریم، علامت‌گذاری شود، به این دلیل که لایه پیوند سمت گیرنده بتواند محدوده یک فریم را در بیت‌های دریافتی تشخیص دهد. به این وظیفه لایه پیوند داده، فریم‌بندی گفته می‌شود.

از دیگر وظایف لایه پیوند، قرار دادن آدرس فیزیکی در فریم اطلاعاتی، کنترل خطا، کنترل جریان و مدیریت کانال انتقال می‌باشد که در ادامه فصل به طور مفصل توضیح داده می‌شود، در کل لایه پیوند داده به دو زیر لایه MAC(Control Media Access) وLLC(Logical Link Control) تقسیم می‌شود. زیر لایه MAC وظیفه قرار دادن آدرس فیزیکی در فریم اطلاعاتی، فریم‌بندی، کنترل خطا، کنترل جریان و مدیریت کانال انتقال را بر عهده دارد. زیر لایه LLC نیز وظیفه دارد ارتباط بین دو ایستگاه در حال انتقال اطلاعات را مدیریت کند و لایه فیزیکی و توپولوژی استفاده شده را از لایه بالاتر یعنی لایه شبکه مخفی کند و به لایه شبکه خدمات ارائه دهد.

توجه: همواره هر لایه به لایه بالاتر از خود سرویس و خدمات ارائه می‌دهد.

در ادامه به تحلیل وظایف لایه پیوند داده می‌پردازیم.

وظیفه اول، تعیین نحوه ارتباط بین ایستگاه مبدأ و مقصد

همانطور که بیان شد زیر لایه LLC وظیفه دارد به لایه شبکه (لایه بالاتر) خدمات ارائه نماید. در این راستا زیر لایه LLC وظیفه دارد چگونگی ارتباط بین ایستگاه مبدأ و مقصد را مدیریت کند. برای این امر سه روش وجود دارد که بررسی می‌کنیم.

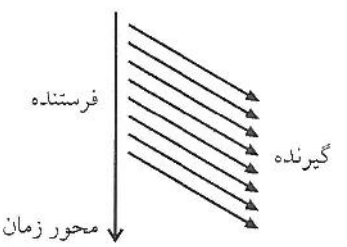
۱- سرویس بدون اتصال و بدون دریافت پیغام پاسخ از گیرنده
(Connection less without Acknowledge)

منظور از بدون اتصال این است که دو ایستگاه فرستنده و گیرنده قبل از شروع انتقال، با هم اطلاعات کنترلی رد و بدل نمی‌کنند، در واقع فرستنده قبل از ارسال، روشن یا خاموش بودن و یا آمادگی ایستگاه گیرنده برای دریافت داده را بررسی نمی‌کند.

پیغام پاسخ (ACK)، پیغامی است که لایه پیوند داده ایستگاه گیرنده بعد از دریافت هر فریم برای فرستنده ارسال می‌کند. دریافت بدون خطای فریم (تأکید بر ارسال فریم بعدی) و یا دریافت خطادار فریم (تأکید بر ارسال دوباره فریم قبلی) در پیغام پاسخ ذکر می‌شود.

در این سرویس ابتدا هیچ اتصالی برقرار نمی‌شود و فرستنده بدون انتظار برای پیغام پاسخ، اطلاعات را به شکل فریم‌های متوالی به سمت گیرنده ارسال می‌کند. در این صورت اگر ایستگاه گیرنده، از ابتدا خاموش باشد و یا آمادگی برای دریافت داده نداشته باشد و یا اطلاعات ارسالی در بین مسیر دچار خطا شوند و یا به هر دلیلی به دست گیرنده نرسند، ایستگاه فرستنده متوجه نخواهد شد.

با توجه به توضیحات بال، این روش در مقابل رخداد خطا به هیچ عنوان قابل اعتماد نیست و بهتر است از رسانه‌هایی که خطای انتقال کمتری دارند مانند فیبر نوری در این روش استفاده شود. به این علت که فرستنده منتظر پاسخ (ACK) نمی‌ماند و پشت سرهم داده‌ها را ارسال می‌کند ، نرخ انتقال در این سرویس بالا می‌رود و این سرویس را برای سیستم‌های بی‌درنگ مناسب می‌کند. به شکل زیر توجه کنید:

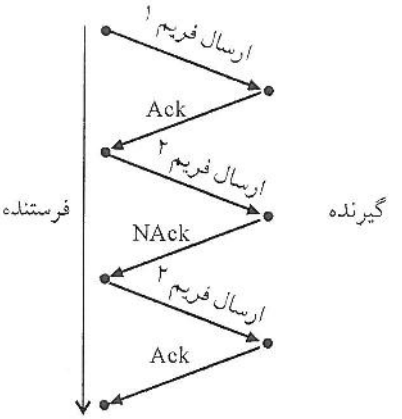


شکل بالا نمایش دیاگرام زمانی سرویس بدون اتصال و بدون دریافت پیغام پاسخ است. در این دیاگرام جهت انتقال اطلاعات با فلش مورب نشان داده شده است. در واقع مورب بودن آن، نمایشی از پدیده تأخیر انتشار می‌باشد که زمان دریافت مدتی بعد از ارسال است. محور زمان نیز به شکل عمودی و در جهت پایین نشان داده می‌شود.

۲- سرویس بدون اتصال، همراه با دریافت پاسخ از گیرنده
(Connection less with Acknowledge)

در این سرویس فرستنده بدون برقراری اتصال شروع به ارسال فریم‌های داده می‌کند، اما بعد از ارسال هر فریم منتظر پیغام پاسخ از طرف گیرنده می‌ماند. در صورتی که پاسخ تصدیق (ACK) دریافت کند، فریم بعدی را ارسال خواهد کرد و اگر پاسخ عدم تصدیق (NACK) دریافت کند، آخرین فریم ارسالی را دوباره ارسال خواهد کرد. در صورتی که تا مدت مشخص از طرف گیرنده پاسخی دریافت نشود، فرستنده دوباره اقدام به ارسال آخرین فریم خواهد کرد.

این روش به علت استفاده از پیغام‌های پاسخ، قابلیت اطمینان بالایی نسبت به روش قبلی دارد و برای کانال های انتقال نویزپذیر، مناسب می‌باشد. با توجه به انتظار فرستنده برای دریافت پیغام پاسخ مناسب، نرخ انتقال در این روش پایین است. شکل زیر دیاگرام زمانی این سرویس را نشان می‌دهد.

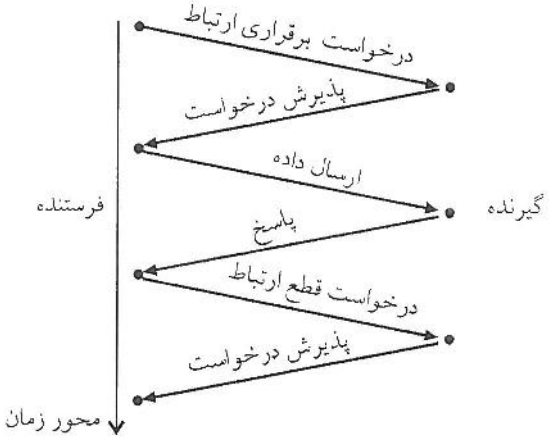


توجه کنید که در شکل بالا منظور از ACK، پاسخ تصدیق (دریافت بدون خطا) یا Acknowledge است و منظور از NACK پاسخ عدم تصدیق (دریافت خطادار) یا Not Acknowledge می‌باشد.

۳- سرویس اتصال‌گرا (Connection Oriented)

در این سرویس قبل از ارسال اطلاعات، عمل اتصال اولیه و ارسال پیغام‌های کنترلی بین فرستنده و گیرنده انجام می‌شود. بعد از اینکه فرستنده از آمادگی گیرنده برای دریافت اطلاعات اطمینان حاصل کرد، شروع به ارسال اطلاعات می‌کند. گیرنده با دریافت هر فریم، پاسخ مناسبی مبنی بر دریافت بدون خطا یا خطادار فریم را برای فرستنده ارسال می‌کند. فرستنده بعد از دریافت پیغام تصدیق، فریم بعدی را ارسال خواهد کرد و در صورت عدم تصدیق، فریم قبلی را دوباره برای گیرنده خواهد فرستاد. بعد از اتمام اطلاعات ارسالی، در انتها دو ایستگاه اتصال خود را قطع می‌کنند. این روش بسیار قابل اطمینان است و برای انتقال فایل استفاده می‌شود. همچنین نرخ انتقال در این سرویس از روش‌های قبلی پایین‌تر است. شکل زیر دیاگرام زمانی این سرویس را

نمایش می‌دهد.

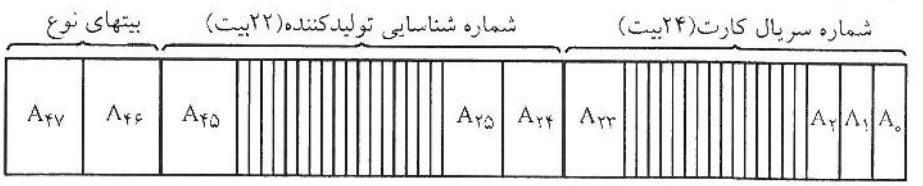


وظیفه دوم، قرار دادن آدرس فیزیکی در فریم اطلاعاتی

آدرس MAC

در شبکه‌های محلی برای انتقال اطلاعات و برقراری ارتباط بین ایستگاه‌ها هر کارت شبکه دارای یک آدرس سخت‌افزاری یکتا و منحصر به فرد در دنیا می‌باشد که درون ROM کارت شبکه ذخیره شده و هویت یک ایستگاه را مشخص می‌کند. به این آدرس، آدرس MAC یا آدرس فیزیکی نیز گفته می‌شود. هر گاه ایستگاهی اقدام به ارسال فریمی نماید، لایه پیوند داده در ایستگاه فرستنده، آدرس فیزیکی مبدأ و مقصد را در فیلد «آدرس مبدأ» و «آدرس مقصد» از فریم قرار داده و برای عمل ارسال به لایه فیزیکی تحویل می‌دهد.

همانطور که در شکل زیر مشاهده می‌کنید، آدرس فیزیکی ۴۸ بیتی و شامل ۱۲ عدد در مبنای شانزده می‌باشد و از سه بخش تشکیل شده است:



- بیت‌های A_0 تا A_{23} شماره‌ی سریال کارت شبکه را نشان می‌دهد. تولید کننده‌ی کارت شبکه، این بخش از شماره را به دلخواه انتخاب و در درون کارت شبکه تنظیم می‌کند. هیچ دوکارت شبکه‌ی تولید یک کارخانه، دارای شماره سریال یکسان نیستند.
- بیت‌های A_{24} تا A_{45} شماره شناسایی کارخانه‌ی تولید کننده‌ی کارت شبکه را مشخص می‌کند. از آن جایی که این شماره یکتاست می‌توان مطمئن بود که هیچ دو کارت شبکه‌ای در دنیا آدرس مشابهی نخواهند داشت. هر تولید کننده‌ای که بخواهد اقدام به تولید کارت

شبکه‌ی اترنت کند، می‌تواند با مرکزی در کالیفرنیا که اکنون تحت مدیریت IEEE است مکاتبه کرده و برای خود یک شناسه‌ی منحصر به فرد دریافت کند.

- بیت $A_{۴۶}$ (مجاور بیت پُر ارزش) سراسری یا محلی بودن آدرس را مشخص می‌کند. اگر این بیت صفر باشد بدین معناست که این آدرس توسط مدیر شبکه‌ی محلی تعیین شده و در خارج از شبکه هیچ ارزشی ندارد. اگر این بیت به یک تنظیم شده باشد بیانگر آن است که این آدرس توسط IEEE به ثبت رسیده و اعتبار جهانی دارد.

قبل از آنکه پر ارزش‌ترین بیت آدرس یعنی $A_{۴۷}$ را توضیح دهیم اشاره به نکاتی در خصوص آدرس دهی در شبکه‌های محلی اهمیت حیاتی دارد، در شبکه‌های محلی یک فریم را می‌توان به دو شکل ارسال کرد:

آدرس تک پخش (Unicast Address)

بدین معنا که گیرنده‌ی فریم یک ایستگاه واحد است. (فرستنده یکی و گیرنده یکی). پُر ارزش‌ترین بیت (یعنی $A_{۴۷}$) در آدرس‌های تک پخشی الزاماً صفر است.

آدرس پخش همگانی (Broad cast Address)

بدین معنا که گیرنده‌ی فریم تمامی ایستگاه‌های متصل به کانال خواهند بود. اگر تمام بیت‌های فیلد آدرس مقصد به ۱ تنظیم شده باشند تمام ایستگاه‌ها فریم را از روی کانال دریافت کرده و آن را پردازش خواهند کرد. در برخی از کاربردها استفاده از این آدرس اهمیت حیاتی دارد.

اگر بخواهیم دو پاراگراف فوق را به بیانی دیگر تکرار کنیم یک ایستگاه در شبکه‌ی محلی، در دو حالت ممکن است فریمی را از روی کانال برداشته و پردازش کند: (الف) در فیلد آدرس مقصد از فریم، آدرس فیزیکی خود را ببیند. (ب) در فیلد آدرس مقصد، تمام بیت‌ها ۱ باشد.

فرمت استاندارد آدرس فیزیکی از شش بایت تشکیل شده است و اغلب در نمایش آن‌ها تمایز مرز بایت‌ها با یک خط تیره یا نقطه مشخص می‌شود. به عنوان مثال دو آدرس زیر نمایش آدرس‌های فیزیکی هستند:

آدرس تک پخشی 07 - 01 - 02 - A1 - 2C - 4B

آدرس پخش همگانی FF - FF - FF - FF - FF – FF

☜ ***نکته:*** شرحی که برای آدرس فیزیکی ارائه کردیم، برای تمامی شبکه‌هایی که توسط IEEE استانداردسازی شده صادق است. به عنوان نمونه شبکه‌های محلی اترنت، حلقه و بی‌سیم wifi نیز به همین ترتیب آدرس دهی شده‌اند. به این آدرس‌ها در عبارتی عام، آدرس‌های IEEE می‌گویند.

کد شناسایی برخی تولید کنندگان مشهور کارت شبکه:

شرکت سازنده	کد شناسایی		
Cisco	00	00	OC
Intel	00	55	0C
D-Link	00	80	C7
Apple	08	00	07
IBM	10	00	5A

لایه پیوند داده در ایستگاه فرستنده، آدرس فیزیکی خود و آدرس فیزیکی مقصد را در فریم داده قرار داده و برای عمل ارسال به لایه فیزیکی تحویل می‌دهد. در سمت گیرنده با دریافت فریم داده، پیغام تصدیق به آدرس مبدأ که در فریم داده ذکر شده است، ارسال خواهد شد.

وظیفه سوم، فریم‌بندی

همانطور که می‌دانید اطلاعات توسط فرستنده در رسانه انتقال به شکل بیت‌های پشت سر هم ارسال می‌شوند و همچنین می‌دانیم که هر لایه به اطلاعات دریافتی از لایه بالاتر، اطلاعات کنترلی خاص خود را می‌افزاید و واحد داده‌ی پروتکل مخصوص به خود را می‌سازد. واحد داده لایه پیوند داده فریم نام دارد و لایه فیزیکی فریم را از لایه پیوند داده گرفته و بیت به بیت منتقل می‌کند. در سمت گیرنده لایه فیزیکی بیت‌های دریافتی را به لایه پیوند داده تحویل می‌دهد اما لایه پیوند داده باید بتواند ابتدا و انتهای داده‌های دریافتی یعنی محدوده‌ی یک فریم را مشخص کند تا اطلاعات کنترلی خود را از آن حذف کند و داده را به لایه‌ی بالاتر تحویل نماید.

فریم‌بندی یعنی لایه پیوند داده فرستنده، اطلاعات را در یک قالب مشخص و مورد توافق فرستنده و گیرنده قرار بدهد که بتوان مرز ابتدا و انتهای آن را طبق توافق به راحتی مشخص نمود. فریم‌بندی در شبکه‌های کامپیوتری به روش‌های زیر انجام می‌شود:

۱- درج فاصله زمانی (سنکرون)

در این روش فرستنده و گیرنده فواصل زمانی خاصی را برای ارسال فریم‌های داده انتخاب می‌کنند و گیرنده بیت‌های دریافتی در این زمان‌ها را به عنوان فریم می‌شناسد. مشکل این روش این است که همزمان کردن زمان و ساعت دو ایستگاه به سادگی انجام نمی‌شود. این روش به علت وجود مشکلات مطرح شده، در شبکه‌های فعلی استفاده نمی‌شود و منسوخ شده است.

۲- استفاده از بیت‌های کنترلی (روش آسنکرون یا ناهمزمان)

با توجه به نام این روش، متوجه خواهید شد که این روش نیازی به همزمان سازی دو ایستگاه فرستنده و گیرنده ندارد بلکه با قرار دادن یک الگوی بیتی، شروع و پایان داده‌ها را مشخص می‌کند. در این روش فرستنده هر زمانی می‌تواند ارسال را آغاز کند و نیازی به فواصل زمانی

ندارد. عمل ارسال در این روش به صورت زیر است:
عدم ارسال در این روش با ارسال مقدار یک باینری انجام می‌شود یعنی عدم ارسال داده با ارسال مداوم یک باینری مشخص می‌شود. وقتی فرستنده قصد ارسال داده دارد یک بیت صفر (بیت آغازین) و پشت سر آن یک بایت داده ارسال می‌کند در پایان هر بایت داده نیز یک یا دو بیت با مقدار یک باینری (بیت پایان) به نشانه اتمام یک بایت ارسال می‌کند البته بر طبق توافق ممکن است بعد هر بایت یک بیت توازن (در آینده توضیح داده می‌شود) ارسال شود.
فرستنده برای ارسال هر بایت مراحل گفته شده را اجرا می‌کند. گیرنده از بیت‌های دریافتی صفر اول و یک آخر را دور می‌ریزد و بقیه را به عنوان داده می‌شناسد. به شکل زیر توجه کنید:

بیت‌های ارسالی

عدم ارسال (ارسال ۱)	1	********	0	عدم ارسال (ارسال ۱)

بیت پایان — داده — بیت آغازین

همانطور که گفته شد برای ارسال هر هشت بیت داده (یک بایت) در این روش حداقل ده بیت ارسال می‌شود که دو بیت آن مربوط به بیت‌های آغاز و پایان که بیت‌های کنترلی هستند، می‌باشد و مابقی داده‌ی اصلی هستند، پس در واقع برای ارسال هر هشت بیت داده دو بیت اضافی ارسال می‌شود که باعث سربار و پایین آمدن کارآیی ارسال خواهد شد. برای بدست آوردن درصد سرباری بیت‌های کنترلی از فرمول استفاده می‌شود:

$$\text{درصد سرباری} = \frac{\text{مقدار اطلاعات غیر مفید (کنترلی) ارسالی}}{\text{مقدار اطلاعات غیر مفید (کنترلی)} + \text{مقدار اطلاعات مفید (داده)}} \times ۱۰۰$$

منظور از مقدار اطلاعات غیر مفید، تعداد بیت‌های ارسالی برای آغاز، پایان و یا بیت توازن می‌باشد. در واقع هر اطلاعاتی که اجباراً به داده اصلی اضافه و برای عملیات کنترلی استفاده می‌شوند ، «اطلاعات غیر مفید» نام دارند.

مثال: اگر سیستمی از روش آسنکرون برای ارسال اطلاعات استفاده کند و با هر بایت داده، یک بیت آغازین، دو بیت پایانی ارسالی نماید، درصد سرباری را برای این روش بدست آورید.

$$\text{درصد سرباری} = \frac{۳}{۸+۳} \times ۱۰۰ = ۲۷/۲۷\%$$

مثال : در انتقال آسنکرون، اگر برای ارسال هر ۸ بیت، یک start bit، یک stop bit و یک parity bit استفاده شود، برای ارسال ۲KB داده، چند بیت باید ارسال شود؟

پاسخ: به ازای هر کاراکتر ۸ بیتی، ۳ بیت (بیت شروع، بیت پایان و بیت پریتی) افزونگی داریم:

داده اصلی	داده کنترلی
۸	۳
۲×۱۰۲۴×۸	x

$$\rightarrow \quad x=\frac{۳\times۲\times۱۰۲۴\times۸}{۸}=۶\times۱۰۲۴ \text{ بیت}$$

حال برای تبدیل به کیلوبایت داریم:

$$\frac{۶\times۱۰۲۴}{۸\times۱۰۲۴}=۰/۷۵\,KB$$

بنابراین حجم کل داده ارسالی (داده اصلی + داده کنترلی) برابر مقدار زیر خواهد بود:

$$۲KB+۰/۷۵KB=۲/۷۵KB$$

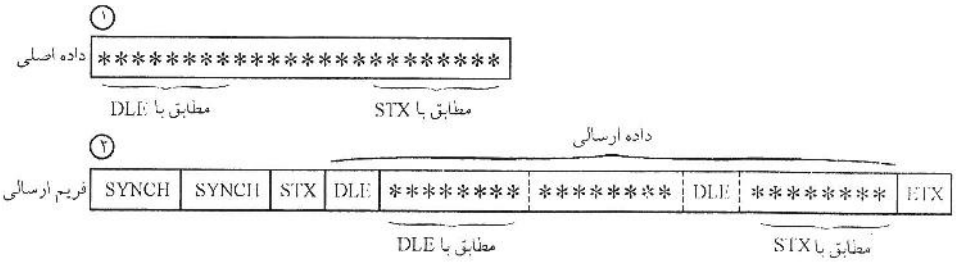
۳- روش سنکرون بایت گرا

همانطور که توضیح داده شد در روش «استفاده از بیت‌های کنترلی» فریم داده‌ها به صورت بایت به بایت ارسال می‌شوند، در این صورت با استفاده از بیت‌های کنترلی، سربار داده در یک فریم بسیار زیاد می‌شود و توان عملیاتی و پهنای باند رسانه انتقال به هدر می‌رود، برای حل این مشکل از روش سنکرون (همزمان) بایت‌گرا استفاده می‌شود. در این روش در هر بار ارسال یک فریم داده انتقال می‌یابد و برای فریم‌بندی این داده‌ها از چهار بایت کنترلی به نام‌های STX، ETX، DLE و SYNCH به شرح زیر استفاده می‌شود.

هرگاه فرستنده قصد ارسال داده داشته باشد، یک یا چند بایت SYNCH برای گیرنده ارسال می‌کند، بایت کنترلی SYNCH مجموعه‌ای از هشت بیت صفر و یک متوالی می‌باشد که در واقع پالس ساعت فرستنده را به گیرنده می‌دهد، این عمل باعث می‌شود که عمل همزمان سازی به خوبی انجام شود و گیرنده در زمان مناسب آماده دریافت بیت‌های داده باشد. حال فرستنده قصد دارد داده‌ها را ارسال کند، برای اینکه شروع شدن ارسال داده را به گیرنده اطلاع دهد، یک بایت STX (start of text) برای گیرنده می‌فرستد و بعد داده‌های خود را ارسال می‌کند.

در پایان نیز برای نشان دادن اتمام ارسال داده، یک بایت ETX (End of text) ارسال می‌کند.

اگر در متن داده اصلی، بایتی مطابق با بایت‌های کنترلی وجود داشته باشد، برای اینکه ایستگاه گیرنده دچار اشتباه نشود (بایت داده را از بایت کنترلی تشخیص ندهد) باید قبل از بایت داده مذکور در داده اصلی، یک بایت کنترلی DLE (Data Link Eskape) اضافه کرد. گیرنده با دریافت هر بایت کنترلی DLE، آن را حذف کرده و بایت بعد از آن را که حتماً مطابق با یکی از بایت‌های کنترلی (DLE , ETX , STX , SYNCH) است، به عنوان بایت داده می‌شناسد. به شکل زیر توجه کنید.



مثال: در شبکه‌ای که فرستنده و گیرنده آن از روش سنکرون بایت‌گرا استفاده می‌کنند؟

گیرنده کد زیر را دریافت می‌کند.

(SYNCH)(STX)(A)(DLE)(ETX)(B)(DLE)(DLE)(DLE)(STX)(C)(DLE)(DLE)(DLE)(DLE)(ETX)

داده اصلی را محاسبه نمایید:

پاسخ: فرم استاندارد داده اصلی و داده کنترلی در روش سنکرون بایت گرا به صورت زیر است:

(SYNCH)(STX)(DATA)(ETX)

بنابراین اطلاعات مابین STX ابتدایی و ETX انتهایی، مربوط به بخش داده اصلی است.

مطابق الگوریتم، گیرنده در صورتی که به یک DLE رسید آن را حذف و در صورتی که به دو DLE رسید، اولی را حذف و دومی را به عنوان داده اصلی نگهداری می‌کند. با حذف DLEهای کنترلی، داده اصلی به صورت زیر خواهد بود:

(A)~~(DLE)~~(ETX)(B)~~(DLE)~~(DLE)~~(DLE)~~(STX)(C)~~(DLE)~~(DLE)~~(DLE)~~(DLE)

داده اصلی:

(A)(ETX)(B)(DLE)(STX)(C)(DLE)(DLE)(DLE)

☜ ***نکته:*** توجه داشته باشید که مراحل بالا به طور مجزا برای هر فریم داده اجرا می‌شود.

توجه: در این روش نیز مانند روش آسنکرون برای محاسبه درصد سربار از رابطه زیر استفاده می‌شود.

$$\text{درصد سربار} = \frac{\text{مقدار اطلاعات غیر مفید (کنترلی) ارسالی}}{\text{مقدار اطلاعات غیر مفید (کنترلی)} + \text{مقدار اطلاعات مفید (داده)}} \times ۱۰۰$$

البته با توجه به ماهیت روش سنکرون بایت گرا، می‌توان واحد اطلاعات در رابطه بالا را بایت در نظر گرفت.

مثال: در ارسال یک فایل داده با حجم 1MB، با تکنیک آسنکرون برای هر بایت داده یک بیت start bit و یک بیت stop bit ارسال می‌شود. این در حالی است که برای ارسال این داده با تکنیک سنکرون بایت گرا، یک بایت STX در ابتدا و یک بایت ETX در انتها ارسال می‌شود. با فرض اینکه

در روش سنکرون بایت‌گرا، احتمال اینکه هر بایت داده یکی از کاراکترهای کنترلی باشد $\frac{۳}{۲۵}$ است، مقدار سربار این دو روش را محاسبه کنید:

پاسخ: روش آسنکرون:

$$\frac{\overbrace{۱}^{\text{start Bit}} + \overbrace{۱}^{\text{stop Bit}}}{\underbrace{۱}_{\text{start Bit}} + ۸\,\text{Bit Data} + \underbrace{۱}_{\text{stop Bit}}} = \frac{۲}{۱۰} = ۲۰\%$$

روش سنکرون:

$$\frac{۳}{۲۵}\ \text{over Head} \rightarrow \underset{۱۰^{۶}}{\text{Data}}\ \underset{x}{\text{DLE}}\ \rightarrow\ x = ۱۲\times۱۰^{۴}\ \text{Byte DLE}$$

$$\frac{\overbrace{۱}^{\text{SYCH}} + \overbrace{۱}^{\text{STX}} + \overbrace{۱۲\times۱۰^{۴}}^{\text{DLE}} + \overbrace{۱}^{\text{ETX}}}{\underbrace{۱}_{\text{SYCH}} + \underbrace{۱}_{\text{STX}} + \underbrace{۱۲\times۱۰^{۴}}_{\text{DLE}} + \underbrace{۱۰^{۶}}_{\text{File}} + \underbrace{۱}_{\text{EXT}}} \cong ۱۱\%$$

۴- روش سنکرون بیت‌گرا، استفاده از پروتکل HDLC (پرچم و جایگزینی بیت)

در این روش داده‌ها به صورت رشته‌ای از بیت‌ها که ابتدا و انتهای آن با یک پرچم به صورت 01111110 مشخص شده است، ارسال می‌شوند. در صورتی که در داده اصلی بایتی به شکل 01111110 وجود داشته باشد، گیرنده این بایت را به عنوان پرچم پایانی در نظر می‌گیرد و این اشتباه باعث خرابی در داده های دریافتی می‌شود. برای حل این مشکل از روش جایگزینی بیت یا اضافه کردن بیت (bit stuffing) استفاده می‌شود به این صورت که فرستنده در متن داده اصلی یک بیت صفر بعد از هر پنج بیت متوالی یک اضافه می‌کند تا به هیچ عنوان شش بیت متوالی یک در متن داده اصلی وجود نداشته باشد. در سمت گیرنده لایه پیوند داده بعد از دریافت داده ارسالی و حذف بایت‌های پرچم، بیت‌های صفر اضافه شده را نیز حذف می‌کند (Bit Destuffing) به شکل زیر توجه کنید:

الف) درج بیت در فرستنده

داده اصلی

0110111111111100

بعد از فریم بندی و اضافه کردن بیت

01111110011011111<u>0</u>111110<u>0</u>0001111110

پرچم پرچم

ب) حذف بیت‌های اضافه شده در گیرنده

داده دریافتی

$$\underbrace{01111110}_{\text{پرچم}}\underbrace{01101111101111100001}_{\text{داده}}\underbrace{01111110}_{\text{پرچم}}$$

بعد از حذف پرچم‌ها و بیت‌های صفر اضافه شده

0110111111111100

۵- قرار دادن طول فریم در سر فصل فریم

در این روش فرستنده طول فریم (تعداد بایت‌های یک فریم) را در ابتدای (سرفصل) فریم قرار می‌دهد و گیرنده بعد از دریافت داده و با بررسی ابتدای آن می‌تواند محدوده‌ی فریم را مشخص کند. مشکل این روش این است که در صورت بروز خطا و تغییر در مقدار طول فریم بر اثر نویز نه تنها فریم جاری خراب خواهد شد بلکه محدوده تمام فریم‌های بعدی نیز غیر قابل تشخیص می‌شود و فریم‌های دیگر از بین خواهند رفت.

ساختار کلی فریم در لایه پیوند داده

همانطور که در فصل یک اشاره شد، واحد داده پروتکل (PDF) مختص به لایه پیوند داده را فریم می‌نامند. ساختار فریم با توجه به پروتکل های استفاده شده در لایه پیوند داده متغیر است ولی در کل تمام ساختارهای فریم از ساختار شکل زیر تبعیت می‌کنند:

فیلد شروع فریم	فیلد آدرس مقصد	فیلد آدرس مبدأ	فیلد طول	داده	فیلد کنترل خطا	فیلد انتهای فریم

در این ساختار فیلدهای شروع و انتها دارای مقادیری هستند که در فریم‌بندی، محدوده فریم را مشخص می‌کنند به عنوان مثال STX و ETX در روش سنکرون بایت‌گرا در این فیلدها قرار می‌گیرند. محتوای فیلد آدرس مبدأ و مقصد، آدرس فیزیکی (MAC) دو ایستگاه فرستنده و گیرنده می‌باشد. با توجه به پروتکل استفاده شده فیلد طول می‌تواند نشان دهنده طول داده و یا کل فریم باشد.

زمانی که فرستنده قصد ارسال اطلاعات دارد با توجه به توافق باگیرنده، یک سری محاسبات بر روی داده و یا کل فریم انجام می‌دهد و با توجه به محاسبات مقداری را در فیلد کنترل خطا قرار می‌دهد. گیرنده با دریافت فریم داده همان اعمال را انجام می‌دهد، در صورتی که فیلد کنترل خطا مقدار درستی داشته باشد گیرنده، فریم را بدون خطا دریافت کرده است.

وظیفه چهارم، کنترل خطا

زمانی که داده‌ها در حال طی مسیر بین فرستنده و گیرنده هستند، ممکن است در مواجه شدن با نویزهای مختلف و یا تضعیف، دچار خطا شوند. منظور از خطا تغییر در بیت‌های داده است به طوری که بیت صفر باینری به یک و یک باینری به صفر تبدیل شود. در این صورت گیرنده داده‌ها

را خطادار دریافت می‌کند و بین داده ارسالی و دریافتی تناقض به وجود می‌آید.
برای حل این مشکل از مکانیزم‌های کنترل خطا استفاده می‌شود. ابتدا فرستنده و گیرنده روی یکی از مکانیزم‌های کشف خطا توافق می‌کنند. فرستنده داده‌ها را در این مکانیزم قرار می‌دهد و نتیجه را (در برخی موارد) در فیلد کنترل خطا می‌گذارد. گیرنده با دریافت داده‌ها، فیلد کنترل خطای داده‌ها را با توجه به مکانیزم توافق شده بررسی می‌کند، در صورت عدم تناقض، داده دریافتی بدون خطا می‌باشد. گیرنده می‌تواند از دو روش کشف خطا و تصحیح خطا برای عمل کنترل خطا استفاده کند.

در مکانیزم کشف خطا، گیرنده متوجه وجود خطا می‌شود ولی آن را تصحیح نمی‌کند و از فرستنده می‌خواهد که داده‌ها را دوباره ارسال کند. در اکثر شبکه‌های کامپیوتری از این روش استفاده می‌شود زیرا سربار کمتری روی محاسبات ایستگاه گیرنده می‌گذارد.

در مکانیزم تصحیح خطا، گیرنده علاوه بر کشف خطا، اقدام به تصحیح خطا می‌کند. در واقع زمانی گیرنده از این مکانیزم استفاده می‌کند که دسترسی به فرستنده نداشته باشد. در شبکه‌های کاملاً یک طرفه از این مکانیزم استفاده می‌شود.

توجه: به طور کلی دو روش برای کنترل خطا وجود دارد:

۱- **روش تصحیح خطای پیش رو(Forward Error connection)** : در این روش مقادیر کنترل خطا (داده‌های افزونه) و داده اصلی برای گیرنده ارسال می‌شود، گیرنده به وسیله این داده‌های افزونه خطاهای احتمالی را کشف و تصحیح می‌کند.

۲- **روش تصحیح خطای پس رو (Backward Error connection)**: در این روش، مقادیر کنترلی، فقط برای کشف خطا به گیرنده کمک می‌کنند و گیرنده در صورت کشف خطا، از فرستنده برای ارسال دوباره اطلاعات تقاضا می‌کند.

فاصله همینگ

برای درک بهتر مکانیزم کشف و تصحیح خطا فاصله همینگ را شرح می‌دهیم:
فرض کنید ایستگاه فرستنده برای انتقال اطلاعات تنها می‌تواند از کدهای مجاز زیر استفاده کند (یعنی تنها باید داده خود را با ترتیب‌های مختلفی از این کدها بیان نماید)

$C_0 = 00000$

$C_1 = 01001$

فاصله همینگ بین دو کد، برابر با تعداد بیت‌های متفاوت در آن دو کد است. به عنوان مثال فاصله همینگ دو کد C_0 و C_1 برابر با ۲ می‌باشد.

$C_0 = 00000$ $\quad C_1 = 01001$ $\quad D(C_0, C_1) = 2$

در واقع اگر دو کد C_0 و C_1 را XOR کنیم فاصله همینگ برابر با تعداد بیت‌های یک باینری نتیجه عمل XOR خواهد بود.

اگر فاصله همینگ بین دو کد برابر D باشد یعنی باید D خطای تک بیتی رخ دهد تا این که یکی از این دو کد به دیگری تبدیل شود. به عنوان مثال اگر کد ارسالی توسط فرستنده $C_۰$=۰۰۰۰۰ باشد و در بین مسیر، این کد دچار نویز شود و دو بیت اول و چهارم آن تغییر کند، کد دریافتی سمت گیرنده به کد ۱۰۰۱۰ که همان $C_۱$ (کد مجاز) است تبدیل می‌شود. در واقع تعداد خطاها برابر با فاصله همینگ (D) می‌باشد و در این صورت گیرنده متوجه خطا نخواهد شد چون کد دریافتی با یکی از کدهای مجاز مطابقت دارد. گیرنده زمانی متوجه بروز خطا می‌شود که حتماً تعداد خطاها از فاصله همینگ کمتر باشد. یعنی حتماً کد ارسالی بر اثر نویز، در سمت گیرنده به یک کد غیر مجاز تبدیل شده باشد، به رابطه زیر توجه کنید:

$$d = D - ۱$$

در رابطه بالا d حداکثر تعداد خطاهایی است که در صورت بروز در کد، قابل تشخیص می‌شود و D فاصله همینگ بین کدهای مجاز است. برای بدست آوردن فاصله همینگ برای بیش از دو کد کافی است فاصله همینگ دو به دوی تمام کدها را بدست آورد و بعد کمترین آنها را به عنوان فاصله همینگ تمام کدها در نظر گرفت. به عنوان مثال فاصله همینگ کدهای زیر برابر با ۲ می‌باشد.

$C_۰$=۰۰۰۰۰ $D(C_۰,C_۱)=۲$

$C_۲$=۰۱۰۰۱ $D(C_۰,C_۲)=۵$ $D(C_۰,C_۱,C_۲)=۲$

$C_۱$=۱۱۱۱۱ $D(C_۱,C_۲)=۳$

اگر تعداد بیت‌های خطا برابر با $\left\lfloor \frac{D-۱}{۲} \right\rfloor$ باشد، بیت‌های خطا قابل تصحیح می‌شوند. فرض کنید دو کد $C_۱$=۰۱۰۰۱ و $C_۲$=۱۱۱۱۱ وجود دارند. (فرستنده فقط می‌تواند از این دو کد برای ارسال داده استفاده کند) فاصله همینگ این دو کد برابر با ۳ می‌باشد (D=۳). تعداد بیت‌های خطا که به صورت بروز قابل تصحیح هستند برابر است با:

$$\left\lfloor \frac{D-۱}{۲} \right\rfloor = \left\lfloor \frac{۳-۱}{۲} \right\rfloor = ۱$$

به عنوان مثال فرستنده داده $C_۱$ را ارسال می‌کند ولی به علت وجود نویز، گیرنده داده‌ی C_X=۰۱۱۰۱ را دریافت می‌کند و آن را با داه‌های مجاز یعنی $C_۱$ و $C_۲$ مقایسه می‌کند با توجه به اینکه فقط یک بیت خطا اتفاق افتاده است، می توان تشخیص داد که کد دریافتی $C_۱$ می‌باشد. در واقع در این مقایسه شبیه‌ترین کد مجاز به کد دریافتی به عنوان کد تصحیح شده در نظر گرفته می‌شود.

حال اگر فرض کنید که فرستنده داده $C_۲$=۱۱۱۱۱ را ارسال نموده و گیرنده داده‌ی C_X=۰۱۱۰۱ را دریافت کرده است، تعداد بیت خطاهایی که رخ داده برابر ۲ می‌باشد در این صورت گیرنده با مقایسه‌ی C_X با کدهای مجاز شبیه‌ترین کد یعنی $C_۱$ را به عنوان کد اصلی در نظر خواهد گرفت و تصحیح به درستی انجام نمی‌شود.

کشف خطا به وسیله بیت توازن

روش بیت توازن یکی از پرکاربردترین و ساده‌ترین روش‌های تشخیص خطا می‌باشد. مکانیزم بیت توازن به دو دسته بیت توازن فرد و بیت توازن زوج تقسیم می‌شود. در مکانیزم بیت توازن به ازای هر هشت بیت داده اصلی، بلافاصله یک بیت به نام بیت توازن اضافه و در مجموع ۹ بیت ارسال می‌شود یعنی برای هر هشت بیت داده یک بیت سربار وجود دارد. در مکانیزم توازن زوج مقدار بیت توازن به شکلی تعیین می‌شود که تعداد بیت‌های یک در ۹ بیت ارسالی، (هشت بیت داده اصلی به علاوه یک بیت توازن)، زوج باشد. در مکانیزم توازن فرد نیز باید بیت توازن طوری مقدار دهی شود که تعداد بیت‌های یک در ۹ بیت ارسالی، فرد باشد.

به عنوان مثال، داده ارسالی برابر با ۰۱۱۰۱۰۱۱ می‌باشد که تعداد بیت‌های یک آن ۵ عدد است. اگر قرار باشد از بیت توازن زوج استفاده شود مقدار بیت توازن برابر با یک می‌شود که در مجموع ۹ بیت ارسالی، تعداد ۶ عدد یک وجود خواهد داشت. اگر قرار باشد بیت توازن فرد استفاده شود مقدار بیت توازن برابر با صفر خواهد بود.

توجه داشته باشید فرستنده و گیرنده قبل از شروع انتقال داده در مورد انتخاب روش بیت توازن فرد یا زوج تصمیم‌گیری می‌کنند. فرستنده در هنگام ارسال داده به ازای هر هشت بیت داده، بیت توازن را محاسبه کرده و بلافاصله به آن اضافه می‌کند. در طرف گیرنده با دریافت هر ۹ بیت تعداد یک‌های آن شمارش می‌شود و در صورت تناقض، خطا تشخیص داده می‌شود. منظور از تناقض، ناهمخوانی زوج یا فرد بودن تعداد یک‌ها نسبت به روش توازن می‌باشد.

بیت توازن باعث می‌شود که فاصله همینگ مجموعه کدهای اسکی برابر با $D = ۲$ شود و در این صورت مکانیزم کشف خطا می‌تواند خطا را تشخیص دهد ولی نمی‌تواند خطا را تصحیح کند.

به عنوان مثال فرض کنید ایستگاه فرستنده برای ارسال اطلاعات تنها از کدهای زیر استفاده می‌کند (داده خود را با ارسال ترتیب‌های مختلف از کدهای زیر ارسال می‌کند):

$C_۰ = ۰۰$

$C_۱ = ۰۱$

$C_۲ = ۱۰$

$C_۳ = ۱۱$

همان‌طور که مشخص است، فاصله همینگ کدهای بالا برابر با حداقل فاصله همینگ بین دو به دوی کدها و برابر با یک می‌باشد.

$D(C_۰, C_۱, C_۲, C_۳) = ۱$

اکنون به هر کدام از کدهای بالا یک بیت اضافه می‌کنیم (به عنوان مثال توازن زوج) و نتیجه به شکل زیر خواهد بود:

$C_۰ = ۰۰۰$

$C_۱ = ۰۱۱$

$C_2 = 101$
$C_3 = 110$

حال اگر فاصله همینگ کدهای جدید را محاسبه کنید در می‌یابید که حداقل فاصله همینگ بین این کدها برابر با دو می‌باشد. این مطلب برای توازن فرد نیز صادق است.

توازن دو بعدی

برای افزایش قدرت تشخیص خطا، علاوه بر بیت توازن که برای هر بایت (هشت بیت) داده محاسبه می‌شود، می‌توان برای بیت‌های هم شماره و بیت توازن بایت‌های ارسالی یک بیت توازن محاسبه کرد و در انتهای فریم، ارسال نمود به شکل زیر توجه کنید:

	Pb	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
بیت توازن فرد سطری (برای هر بایت داده)	0	0	1	1	0	1	0	1	1	0
	1	1	0	1	1	0	1	0	0	0
	0	0	1	1	1	1	0	0	0	1
بیت توازن زوج ستونی (بیت‌های هم‌شماره)	1	1	0	1	0	0	1	1	1	1

اگر یک خطا در هر بیت دلخواه رخ دهد، بیت‌های توازن سطر و ستون مربوطه هر دو این خطا را آشکار می‌کنند.پس نه تنها موفق به تشخیص خطا شده‌ایم، بلکه مکان وقوع خطا را نیز در تلاقی این سطر و ستون پیدا کرده و قادر به تصحیح آن هستیم. اگر دو خطا در یک سطر یا یک ستون رخ دهد، اگرچه یکی از توازن‌های سطری یا ستونی خطا را کشف نمی‌کنند، دیگری قادر به آشکارسازی آن خواهد بود. اگر سه خطا رخ دهد و این سه خطا در سه رأس یک مثلث قائم‌الزاویه قرار گیرند، وقوع خطا قابل کشف است ولی چون فقط یک سطر و ستون خطا را نشان می‌دهند، گیرنده به اشتباه محل تلاقی این سطر و ستون خطا را نشان می‌دهند، گیرنده به اشتباه محل تلاقی این سطر و ستون را تصحیح می‌کند. اگر چهار خطا در چهار رأس یک مستطیل رخ دهد، خطا غیرقابل تشخیص خواهد بود.

جمع مقابله‌ای (Checksum)

در این روش لایه پیوند داده ایستگاه فرستنده، تمام بایت‌های فریم ارسالی (داده اصلی و تمام فیلدهای کنترلی به جز فیلد Checksum) را با هم جمع می‌کند و مکمل یک می‌گیرد و نتیجه را در فیلد Checksum قرار می‌دهد و فریم را ارسال می‌کند. گیرنده با دریافت فریم داده، تمام بایت‌های آن از جمله Checksum را با هم جمع می‌کند در صورتی که حاصل این محاسبه غیر از صفر باشد، حتماً خطا رخ داده است.

☜ **نکته:** توجه داشته باشید که حاصل جمع یک عدد با مکمل ۱ آن عدد برابر با صفر خواهد بود.

یادآوری: برای تبدیل اعداد باینری به شکل مکمل ۱ آن کافی است تمام بیت‌های یک آن عدد را

به صفر و تمام بیت های صفر را به یک تبدیل نمایید. برای بدست آوردن معادل دهدهی یک عدد مکمل ۱، ارزش هر بیت به جز بیت پر ارزش به همان شکل معمول یعنی توان‌های دو در نظر گرفته می‌شود. ولی پُر ارزش بیت پُر ارزش یک واحد کم و منفی می‌شود به عنوان مثال:

$$\begin{pmatrix} -۱۵ & ۸ & ۴ & ۲ & ۱ \\ ۱ & ۱ & ۰ & ۰ & ۱ \end{pmatrix} = -۱۵+۸۱ = -۶$$

مثال: فرض کنید مجموع بایت‌های یک فریم برابر با ۱۰۱۰۱۱۰۱ می‌باشد برای بدست آوردن مقدار Checksum کافی است آن را به مکمل ۱ تبدیل نماییم که نتیجه این عمل برابر با ۰۱۰۱۰۰۱۰ خواهد بود. در سمت گیرنده، مجموع تمام بایت‌های فریم و Checksum با هم جمع می‌شوند، عمل جمع به وسیله عملگر XOR و بدون در نظر گرفتن ارقام نقلی انجام می‌پذیرد:

۱۰۱۰۱۱۰۱

XOR ۰۱۰۱۰۰۱۰

۱۱۱۱۱۱۱۱

اکنون برای بدست آوردن معادل دهدهی نتیجه، کافی است به شکل زیر عمل کرد:

$$\begin{matrix} -۱۲۷ & ۶۴ & ۳۲ & ۱۶ & ۸ & ۴ & ۲ & ۱ \\ ۱ & ۱ & ۱ & ۱ & ۱ & ۱ & ۱ & ۱ \end{matrix} = -۱۲۷+۶۴+۳۲+۱۶+۸+۴+۲+۱ = ۰$$

کدهمینگ

برای اینکه بتوان خطا در هر بایت ارسالی را تشخیص داد و یک بیت خطا در آن را تصحیح نمود. از روش کد همینگ استفاده می‌شود. کد همینگ را با یک مثال توضیح می‌دهیم:

فرض کنید فرستنده قصد ارسال داده ۱۰۰۱ را دارد و طبق توافق بین فرستنده و گیرنده قرار است از کد همینگ برای کشف خطا استفاده شود. برای اینکه بتوان در هر بایت از داده خطا را تشخیص داد باید برای هر m بیت داده اصلی، r بیت داده کنترلی یا افزونه اضافه کرد. به این شکل که داده‌های کنترلی در بیت‌هایی با اندیس‌هایی از توان دو (۱ و ۲ و ۴ و ۸ و ...) و داده‌های اصلی در اندیس‌های باقی‌مانده (۳ و ۵ و ۶ و ۷) قرار می‌گیرند. به شکل زیر توجه کنید:

$r_۱$	$r_۲$	$m_۳$	$r_۴$	$m_۵$	$m_۶$	$m_۷$
؟	؟	۱	؟	۱	۰	۱

برای بدست آوردن مقادیر بیت‌های افزونه به شکل زیر عمل می‌شود:

ابتدا باید شماره اندیس بیت‌های داده اصلی را با استفاده از اعداد توان ۲ بدست آوریم، به عنوان مثال عدد ۷ از مجموع اعداد ۴ و ۲ و ۱ که همه آنها اعداد توان ۲ هستند، بدست می‌آید:

۱ + ۲ + ۴ = ۷

۲ + ۴ = ۶

$۱ + ۴ = ۵$

$۱ + ۲ = ۳$

حال برای بدست آوردن مقدار بیت‌های افزونه کافی است مقدار اندیس‌های داده اصلی که اندیس بیت افزونه در بدست آوردن شماره اندیس آنها نقش داشته است، با هم XOR شوند و در اندیس مورد نظر قرار گیرند. به عنوان مثال برای بدست آوردن $r_۱$ باید مقادیر بیت‌های با اندیس ۷ و ۵ و ۳ را با هم XOR نمود:

$r_۱ = m_۳ \oplus m_۵ \oplus m_۷ \Rightarrow r_۱ = ۱ \oplus ۰ \oplus ۱ = ۰$

$r_۲ = m_۳ \oplus m_۶ \oplus m_۷ \Rightarrow r_۲ = ۱ \oplus ۰ \oplus ۱ = ۰$

$r_۴ = m_۵ \oplus m_۶ \oplus m_۷ \Rightarrow r_۴ = ۰ \oplus ۰ \oplus ۱ = ۱$

در نتیجه کد ارسالی برابر می‌شود با:

$r_۱$	$r_۲$	$m_۳$	$r_۴$	$m_۵$	$m_۶$	$m_۷$
۰	۰	۱	۱	۰	۰	۱

در یک راه‌حل دیگر، می‌توان بدون در نظر داشتن روابط کد همینگ، مقادیر داده‌های کنترلی را به سادگی از روی مقادیر داده‌های اصلی استخراج نمود.

مطابق فرض مثال مطرح شده، فرستنده قصد ارسال عدد ۹ با فرمت باینری ۱۰۰۱ را در قالب کد همینگ دارد. همچنین مطابق رابطه همینگ، تعداد بیت‌های داده‌های کنترلی برابر ۳ بیت است.

رابطه همینگ

همینگ یک رابطه کلی برای محاسبه حداقل افزونگی لازم برای تصحیح یک بیت خطا به دست آورده است که این رابطه نه تنها برای کد همینگ بلکه برای هر کدی که قادر به تصحیح یک بیت خطا باشد، صادق است. مطابق این رابطه، در هر کدی n بیتی که بخواهد یک بیت خطا را تصحیح کند، حداقل افزونگی لازم (r) از رابطه زیر به دست می‌آید:

$m + r + ۱ \leq ۲^r$

توجه: داده اصلی، m بیتی است $(n = m + r)$.

مثال: برای ارسال ۱ بیت داده اصلی، حداقل چند بیت داده کنترلی (داده افزونه)، اضافه گردد تا بتوان یک بیت خطا را تصحیح کرد؟ برای ۴ بیت داده اصلی چطور؟ برای ۱۰۰۰ بیت داده اصلی چطور؟

پاسخ:

برای ۱ بیت داده اصلی داریم:

$m + r + ۱ \leq ۲^r \rightarrow ۱ + r + ۱ \leq ۲^r \rightarrow ۲ + r \leq ۲^r \rightarrow r_{min} = ۲$

برای ۴ بیت داده اصلی داریم:

$m+r+1 \le 2^r \rightarrow 4+r+1 \le 2^r \rightarrow 5+r \le 2^r \rightarrow r_{min}=3$

برای ۱۰۰۰ بیت داده اصلی داریم:

$m+r+1 \le 2^r \rightarrow 1000+r+1 \le 2^r \rightarrow 1001+r \le 2^r \rightarrow r_{min}=10$

بنابراین ۳ بیت داده کنترلی مطابق الگوی همینگ باید به داده اصلی اضافه گردد. بنابراین داریم:

r_1	r_2	m_3	r_4	m_5	m_6	m_7
؟	؟	۱	؟	۰	۰	۱

برای استخراج داده‌های کنترلی از روی داده‌های اصلی جدول زیر مورد استفاده قرار می‌گیرد:

شماره بیتی که محتوی آن یک باشد	معادل باینری		
۳	۰	۱	۱
۷	۱	۱	۱
	$r_4=1$	$r_2=0$	$r_1=0$

در جدول فوق شماره بیت‌هایی از داده اصلی که مقدار ۱ دارند در ستون مربوطه درج می‌شود، سپس در بخش معادل باینری به صورت عمودی (ستونی) عمل XOR انجام می‌شود، که نتیجه حاصل به ترتیب از چپ به راست $r_1=0$، $r_2=0$ و $r_4=1$ خواهد بود.

بنابراین داده‌ای که باید ارسال شود به صورت زیر خواهد بود:

r_1	r_2	m_3	r_4	m_5	m_6	m_7
۰	۰	۱	۱	۰	۰	۱

کنترل خطا در کد همینگ

گیرنده با دریافت داده‌ها، برای عمل کنترل خطا مقدار اندیس‌های داده که اندیس بیت افزونه در بدست آوردن شماره اندیس آنها نقش دارد با مقدار همان بیت افزونه XOR می‌کند، اگر نتیجه ۰ باشد خطایی رخ نداده است:

$s_1 = r_1 \oplus m_3 \oplus m_5 \oplus m_7 \Rightarrow s_1 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$

$s_2 = r_2 \oplus m_3 \oplus m_6 \oplus m_7 \Rightarrow s_2 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$

$s_4 = r_4 \oplus m_5 \oplus m_6 \oplus m_7 \Rightarrow s_4 = 1 \oplus 0 \oplus 0 \oplus 1 = 0$

در صورتی که در یک بیت، خطا رخ داده باشد محل وقوع خطا را می‌توان با معادل مقدار باینری $s_4\, s_2\, s_1$ بدست آورد. به عنوان مثال فرض کنید که گیرنده داده زیر را دریافت نموده است.

داده دریافتی

r_1	r_2	m_3	r_4	m_5	m_6	m_7
۰	۰	۱	۱	۱	۰	۱

$s_1 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$

$s_2 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$

$S_۴ = ۱ \oplus ۱ \oplus ۰ \oplus ۱ = ۱$

$S_۴ S_۲ S_۱ = (۱۰۱)_۲ = (۵)_{۱۰}$

در نتیجه بیت شماره ۵ دچار خطا شده است و قابل تصحیح می‌باشد.

در یک راه حل دیگر، می‌توان بدون در نظر گرفتن روابط کد همینگ، محل وقوع خطا را از روی داده‌های اصلی و کنترلی استخراج نمود.

مطابق مثال مطرح شده، فرض کنید گیرنده، داده خطادار زیر را در الگوی کد همینگ دریافت کرده است.

$r_۱$	$r_۲$	$m_۳$	$r_۴$	$m_۵$	$m_۶$	$m_۷$
۰	۰	۱	۱	۱	۰	۱

در حالی که باید داده زیر را دریافت می‌کرد:

$r_۱$	$r_۲$	$m_۳$	$r_۴$	$m_۵$	$m_۶$	$m_۷$
۰	۰	۱	۱	۰	۰	۱

در واقع در بیت پنجم یعنی $m_۵$ خطا رخ داده است.

برای کشف محل وقوع خطا، جدول زیر مورد استفاده قرار می‌گیرد.

شماره بیتی که محتوی آن یک باشد	معادل باینری		
۳	۰	۱	۱
۴	۱	۰	۰
۵	۱	۰	۱
۷	۱	۱	۱
	$S_۴ = ۱$	$S_۲ = ۰$	$S_۱ = ۱$

بنابراین مطابق الگوی زیر:

$S_۴ S_۲ S_۱ = (۱۰۱) = (۵)_{۱۰}$

در بیت شماره ۵ یعنی $m_۵$ خطا رخ داده است و باید تصحیح گردد.

کد خطی

کد خطی یک گروه m بیتی داده اصلی را به صورت $(b_۱ , b_۲ , ... , b_m)$ گرفته و یک کلمه کد n بیتی $(b_۱ , b_۲ , ... , b_n)$ را تولید می‌کند که در آن m بیت داده اصلی و $r = n - m$ بیت دیگر، مربوط به چک کردن خطاست (داده کنترلی یا داده افزونه). این کد را به صورت کد خطی از درجه (n,m) نشان می‌دهند. برای مثال کد همینگ استاندارد که برای ۴ بیت داده اصلی از ۳ بیت افزونه استفاده می‌کند و یک کد ۷ بیتی را تولید می‌نماید، یک نوع کد خطی $(۷,۴)$ محسوب

می‌شود.

توجه: در حل مسائل، در صورتی که از کد همینگ اسمی برده نشود، کد خطی معمولی در نظر گرفته می‌شود. در چنین کدهایی، صورت سؤال باید الگوی ساخت بیت‌های کنترلی از روی داده‌های اصلی را بدهد.

توجه: برای به دست آوردن حداقل فاصله همینگ در کدهای خطی، باید تمام کدهای خطی را تولید کرد، سپس وزن (یا همان تعداد یک‌های) هر یک از کدها را به دست آورد و کمترین وزن غیرصفر را به عنوان حداقل فاصله همینگ در نظر گرفت.

مثال: در یک کد خطی (۳ و ۶) بیت‌های چک‌کننده (check bits) به صورت زیر محاسبه می‌شوند.

$b_4 = b_1 \oplus b_3$

$b_5 = b_1 \oplus b_2$

$b_6 = b_2 \oplus b_3$

حداقل فاصله همینگ در این کد چقدر است؟

۱) ۴ ۲) ۳ ۳) ۲ ۴) ۵

پاسخ: گزینه (۲) صحیح است.

مطابق جدول زیر برای کد خطی (۶,۳) داریم:

$(n,m) \rightarrow (6,3) \rightarrow r = 6-3 = 3$ بیت داده کنترلی

مطابق فرض سؤال، الگوی محاسبه بیت‌های کنترلی به صورت زیر است:

$b_4 = b_1 \oplus b_3$

$b_5 = b_1 \oplus b_2$

$b_6 = b_2 \oplus b_3$

داده اصلی			داده کنترلی			وزن
b_1	b_2	b_3	b_4	b_5	b_6	W
۰	۰	۰	۰	۰	۰	۰
۰	۰	۱	۰	۱	۱	۳
۰	۱	۰	۱	۰	۱	۳
۰	۱	۱	۱	۱	۰	۴
۱	۰	۰	۱	۱	۰	۳
۱	۰	۱	۱	۰	۱	۴
۱	۱	۰	۰	۱	۱	۴
۱	۱	۱	۰	۰	۰	۳

$\rightarrow W_{min} = 3 \rightarrow D_{min} = 3$

کد افزونه چرخشی (Cyclic Redundancy check :CRC)

این روش برخلاف روش‌های قبلی بر مبنای بیت توازن عمل نمی‌کند. در این روش فرستنده داده‌های فریم ارسالی (رشته‌ای از بیت‌ها) را بر یک چند جمله‌ای مولد که قبلاً بین فرستنده و گیرنده توافق شده است، تقسیم می‌کند و باقی‌مانده را به عنوان کد CRC به همراه داده اصلی برای گیرنده می‌فرستد. چند جمله‌ای مولد که داده ارسالی بر آن تقسیم می‌شود ((G(x)) معادل یک کد باینری است. به عنوان مثال چند جمله‌ای x^4+x^2+1 برابر با عدد باینری ۱۰۱۰۱ می‌باشد، به شکل زیر توجه کنید:

۴	۳	۲	۱	۰
۱	۰	۱	۰	۱

$x^4+x^2+x^0$

در واقع هر اندیس از کد باینری که عدد ۱ باینری در آن قرار داشته باشد، در چند جمله‌ای به عنوان توان یک جمله استفاده می‌شود. نکته مهم در این روش این است که در عدد باینری معادل G(x) باید کم ارزش‌ترین و پرارزش‌ترین بیت دارای مقدار یک باینری باشند.

در عمل تقسیم داده بر چند جمله‌ای G(x)، به جای عمل تفریق معمولی از تفریق به شکل modul-2 استفاده می‌شود، در واقع به جای تفریق، بیت‌های متناظر را XOR می‌کنیم و نتیجه را بدست می‌آوریم.

در مکانیزم CRC، فرستنده قبل از انجام عمل تقسیم، تعداد k بیت صفر به سمت راست داده‌ی اصلی اضافه می‌کند (k یک واحد کمتر از تعداد بیت‌های چند جمله‌ای مولد است) سپس عمل تقسیم را انجام می‌دهد و باقی‌مانده تقسیم را به عنوان کد CRC در سمت راست داده اصلی قرار داده و ارسال می‌کند. توجه داشته باشید هنگام ارسال، k بیت صفر اضافه شده به داده اصلی حذف می‌شود.

☜ **نکته:** تعداد بیت‌های معادل یک چند جمله‌ای از درجه n برابر n + ۱ می‌باشد.

مثال:

x^6+x^2+1

۶	۵	۴	۳	۲	۱	۰
۱	۰	۰	۰	۱	۰	۱

در واقع با احتساب توان صفر تعداد بیت‌های یک چند جمله‌ای برابر با n + ۱ می‌شود. در مقابل می‌توان اینگونه بیان کرد که تعداد بیت صفر اضافه شده به سمت راست داده اصلی برابر با درجه چند جمله‌ای مولد می‌باشد (k=n).

به عنوان مثال فرستنده قصد دارد داده‌ی ۱۱۰۱۰۱۱۱۰۱ را ارسال کند و چند جمله‌ای مولد G(x) مورد توافق فرستنده و گیرنده ۱۰۰۱۱ (x^4+x+1) می‌باشد فریم ارسالی را به شکل زیر بدست

می‌آوریم:
ابتدا به سمت راست داده اصلی، چهار صفر (یکی کمتر از تعداد بیت‌های چندجمله‌ای مولد یا برابر درجه چندجمله‌ای مولد) اضافه می‌کنیم و سپس عمل تقسیم را به شکل زیر انجام می‌دهیم و چهار بیت کم ارزش باقی مانده را به عنوان کد CRC در نظر می‌گیریم.

```
۱۱۰۱۰۱۱۱۰۱۰۰۰۰ | ۱۰۰۱۱
۱۰۰۱۱           | ۱۱۰۰۰۰۰۱۱۰۰
-----
۰۱۰۰۱۱
  ۱۰۰۱۱
-----
۰۰۰۰۰۱۱۰۱۰
     ۱۰۰۱۱
-----
۰۱۰۰۱۰
   ۱۰۰۱۱
-----
۰۰۰۰۱۰۰
```

حال فریم ارسالی به شکل زیر خواهد بود:

داده اصلی	CRC
۱۱۰۱۰۱۱۱۰۱	۰۱۰۰

گیرنده با دریافت این فریم، کل داده (CRC و داده اصلی) را یکجا به جمله مولد G(x) توافق شده تقسیم می‌کند. در صورتی که باقی مانده تقسیم صفر باشد، به این معنی است که هیچ خطایی رخ نداده است و اگر نه خطا رخ داده و گیرنده برای ارسال مجدد داده‌ها به فرستنده در خواست می‌دهد.

توجه: اضافه شدن k بیت صفر به سمت راست داده اصلی به این علت است که در سمت گیرنده، CRC در جای آن k بیت قرار بگیرد و باقی مانده عمل تقسیم را برای تشخیص وقوع خطا تعیین کند.

وظیفه پنجم، کنترل جریان

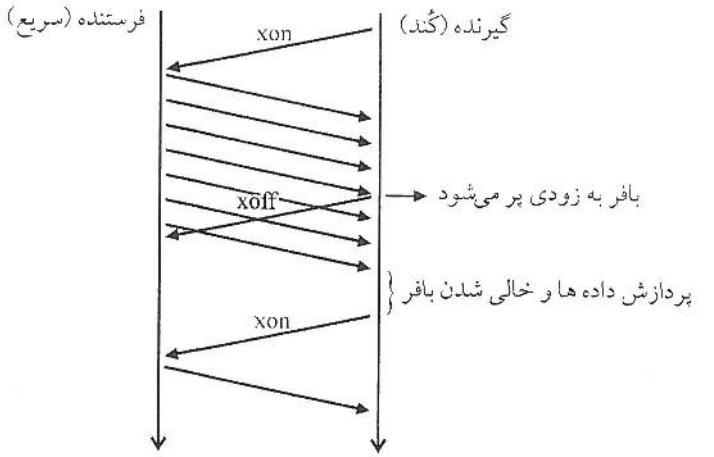
ایستگاه‌های تشکیل دهنده‌ی یک شبکه کامپیوتری الزاماً دارای سرعت و قدرت یکسان نمی‌باشند و ممکن است دو ایستگاه با توان متفاوت به برقراری ارتباط و رد و بدل داده‌ها بپردازند. همان‌طور که می‌دانید ایستگاه گیرنده بعد از دریافت داده‌های منتقل شده، آنها را در یک بافر قرار می‌دهد و هنگام پردازش، داده‌ها را از بافر بر می‌دارد. اگر سرعت پردازش گیرنده بیشتر از سرعت دریافت داده باشد، هیچ‌گاه بافر گیرنده پُر نخواهد شد، ولی اگر سرعت پردازش گیرنده کمتر از سرعت دریافت داده باشد، بافر پُر خواهد شد در این صورت گیرنده دیگر نمی‌تواند داده‌های دریافتی را

بافر نماید و داده‌ها را از دست می‌دهد. برای حل این مشکل گیرنده باید کمی قبل از پُر شدن بافر، به فرستنده اعلام کند که ارسال را متوقف نماید و بعد از خالی شدن بافر دوباره از فرستنده درخواست ارسال اطلاعات نماید. به این امر کنترل جریان داده گفته می‌شود. توجه داشته باشید که سرعت انتقال داده به ایستگاه فرستنده و کانال انتقال وابسته است.

کنترل جریان داده در لایه پیوند داده به روش‌های زیر انجام می‌شود:

۱- کنترل جریان نرم‌افزاری XOFF (Transmission OFF) و XON (Transmission ON)

در این روش گیرنده از دو پیغام کنترلی xon و xoff برای کنترل جریان داده استفاده می‌کند. xoff به معنی توقف ارسال داده و xon به معنی شروع و ادامه ارسال داده توسط فرستنده می‌باشد به شکل زیر توجه کنید:



اکنون سؤال مهم این است که گیرنده باید چه زمانی پیغام xoff را ارسال نماید؟ فاصله زمانی بین ارسال داده توسط فرستنده و دریافت آن توسط گیرنده را با T_p نشان می‌دهند. فرستنده دقیقاً تا قبل از دریافت xoff در حال ارسال داده می‌باشد، پس فاصله زمانی بین ارسال xoff و دریافت آخرین داده ارسالی توسط گیرنده برابر با $2T_p$ می‌باشد.

اگر داده‌ها با نرخ R_s توسط گیرنده دریافت و در بافر قرار بگیرند و با نرخ R_p از بافر برداشته و پردازش شوند اگر $R_p \geq R_s$ باشد بافر هیچ‌گاه پُر نخواهد شد. یعنی گیرنده با همان سرعتی که داده‌ها را دریافت و بافر می‌کند، داده‌ها را از بافر برداشته و پردازش می‌نماید. در غیر اینصورت اگر $R_p < R_s$ باشد، برای بدست آوردن زمان مناسب برای ارسال xoff یعنی محاسبه اینکه بافر باید چه مقدار فضای خالی داشته باشد تا گیرنده مجبور به ارسال xoff شود، از رابطه زیر استفاده می‌نماییم:

$$\Delta R = R_s - R_p$$

$$L = 2T_p \times \Delta R$$

رابطه بالا به این معنی است که در مدت زمان $۲T_p$ چه مقدار داده وارد بافر می‌شود که گیرنده فرصت نمی‌کند آنها را برای پردازش از بافر خارج کند.

مثال: ایستگاه A با نرخ ۱۰ مگابیت در ثانیه داده‌هایی را برای ایستگاه B ارسال می‌کند. ایستگاه B داده‌های دریافتی را در بافر دریافت خود قرار داده و با نرخ ۹/۲ بیت در ثانیه آن‌ها را پردازش می‌کند. اگر ایستگاه B برای کنترل جریان از پیام‌های کنترلی X-OFF و X-ON استفاده کند. با فرض اینکه تأخیر یک طرفه ارتباط ۱۰ میلی‌ثانیه باشد. ایستگاه B در زمانی که بافر دریافتش چند بایت فضای خالی دارد باید پیام کنترلی X-OFF را ارسال کند تا بافرش سرریز نشود؟

۱) ۴۰۰۰ ۲) ۱۰۰۰ ۳) ۲۰۰۰ ۴) ۳۰۰۰

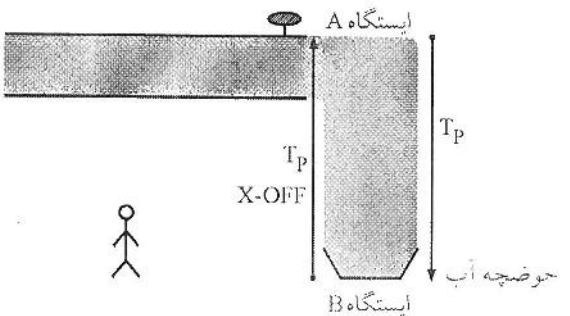
پاسخ: داده‌های مسئله به صورت زیر است:

$$R_{A-Send} = ۱۰ \text{ Mbps} = ۱۰^۷ \text{ bps} \;,\; R_{B-Process} = ۹/۲ \text{ Mbps} = ۹/۲ \times ۱۰^۶ \text{ bps}$$

$$T_P = ۱۰ \text{ ms} = ۱۰^{-۲} \text{s}$$

برای درک بهتر مثال زیر را در نظر بگیرید:

مطابق شکل زیر فرض کنید، در پایین یک آبشار با نرخ انتقال A-Send یک حوضچه آب با نرخ پردازش B-Process قرار گرفته است، تأخیر انتشار آب از بالای آبشار تا حوضچه آب برابر T_P است، حداقل مقدار فضای خالی حوضچه آب چقدر باشد، تا از لحظه‌ای که نگهبان حوضچه آب از پایین به بالا فرمان بستن آبشار را صادر می‌کند، حوضچه آب سرریز نشود؟



دو ایستگاه A و B و یک شبکه دو طرفه داریم. براساس الگوریتم X-ON و X-OFF، وقتی بافر مقصد یعنی B در حال پُر شدن است، باید برای ایستگاه A پیغام X-OFF را ارسال کند. اما به دلیل پدیده تأخیر انتشار هنگامی که پیغام X-OFF از ایستگاه B به سمت ایستگاه A ارسال می‌شود. بعد از صرف T_p به ایستگاه A می‌رسد.

اما در طول این زمان T_p و قبل از رسیدن پیغام X-OFF به ایستگاه A، ایستگاه A، همچنان همینطور پشت سر هم در حال ارسال داده برای ایستگاه B است. پس ایستگاه B باید بتواند این داده‌ها را نیز بافر کند. از طرف دیگر، اگر ایستگاه A پس از مدت زمان T_p یعنی دقیقاً از لحظه

ارسال پیغام X-OFF از ایستگاه B به ایستگاه A، این پیغام X-OFF را دریافت کند و به تبع ارسال داده را متوقف کند، داده‌هایی که تا کمی قبل از دریافت X-OFF در حال ارسال آنها بوده است نیز پس از یک T_p به طور کامل به ایستگاه B خواهند رسید. (مانند آب‌هایی که پس از بستن دریچه آبشار، میان حوضچه و آبشار قرار دارد که تمام این حجم آب پس از یک T_p، به حوضچه خواهند رسید.) پس ایستگاه B باید به قدری بافر داشته باشد که به اندازه زمان $2T_p$ (یک T_p برای زمانی که پیغام X-OFF از ایستگاه B به ایستگاه A برسد و یک T_p هم برای اینکه داده‌های میان ایستگاه A و ایستگاه B، پس از دریافت پیغام X-OFF توسط ایستگاه B و به تبع بسته شدن کانال به ایستگاه B برسد) بتواند داده‌های ارسالی از ایستگاه A به ایستگاه B را در خود ذخیره‌سازی و بافر کند.

در مثال آبشار، اگر در هر ثانیه ۱۰ سطل آب وارد حوضچه شود و نگهبان حوضچه در هر ثانیه، ۶ سطل آب را از حوضچه خارج کند، و تأخیر انتشار از دریچه آبشار تا حوضچه ۳ ثانیه باشد، چقدر حوضچه باید فضای خالی داشته باشد، تا بتواند پیغام X-OFF را به سمت دریچه آبشار ارسال کند، تا پس از دریافت پیغام X-OFF توسط ایستگاه A و به تبع بسته شدن دریچه آبشار در ایستگاه A، حوضچه موجود در ایستگاه B سرریز نشود؟

سطل در هر ثانیه $R_{A-Send} = ۱۰$ = نرخ ارسال دریچه آبشار

سطل در هر ثانیه $R_{B\ Process} = ۶$ = نرخ پردازش حوضچه

توجه: در هر ثانیه ۴ سطل (۴=۶-۱۰) در حوضچه باقی می‌ماند.

$$\Delta R = (R_{A\ Send} - R_{B-Process}) = (۱۰-۶) = ۴ \quad \text{سطل}$$

زمان	آب باقی‌مانده
۱s	$\Delta R = ۴$
$2T_p = ۲\times۳ = ۶$	L

$$\Rightarrow \quad L = 2T_p \times \Delta R = ۲\times۳\times۴ = ۲۴ \quad \text{سطل}$$

بنابراین دقیقاً در لحظه ارسال پیغام X-OFF از ایستگاه B به ایستگاه A، باید ۲۴ سطل فضای خالی در حوضچه وجود داشته باشد تا پس از دریافت پیغام X-OFF توسط ایستگاه A و به تبع بسته شدن دریچه آبشار در ایستگاه A، حوضچه موجود در ایستگاه B سرریز نشود.

با توجه به مثال مطرح شده، اگر در هر ثانیه ۱۰×۱۰ bps داده از ایستگاه A به ایستگاه B وارد شود و ایستگاه B در هر ثانیه $۹/۲\times۱۰^۶$ bps داده را پردازش و از بافر ایستگاه B خارج کند، و تأخیر انتشار از ایستگاه A تا ایستگاه B، ۱۰ میلی‌ثانیه باشد، چقدر ایستگاه B باید فضای خالی داشته باشد، تا بتواند پیغام X-OFF را به سمت ایستگاه A ارسال کند، تا پس از دریافت پیغام X-OFF توسط ایستگاه A و به تبع بسته شدن ارسال داده توسط ایستگاه A، بافر ایستگاه B سرریز نشود؟

نرخ ارسال ایستگاه A = $R_{A-Send} = ۱۰\times۱۰^۴$ bps

$R_{B-Process}$ = نرخ پردازش ایستگاه B $= ۹/۲\times۱۰^{۶}$ bps

توجه: در هر ثانیه $۰/۸\times۱۰^{۶}$ بیت $(۰/۸\times۱۰^{۶} = (۹/۲\times۱۰^{۶}) - ((۱۰\times۱۰^{۶}))$ در بافر ایستگاه B باقی می‌ماند.

$$\Delta R = (R_{A-Send} - R_{B-Process}) = ((۱۰\times۱۰^{۶}) - (۹/۲\times۱۰^{۶})) = ۰/۸\times۱۰^{۶} \text{ bit}$$

زمان	داده باقی‌مانده
۱s	$\Delta R = ۰/۸\times۱۰^{۶}$ bit
$۲T_p = ۲\times۱۰ms = ۲\times۱۰^{-۲}s$	L

$$L = ۲T_p \times \Delta R$$

$$L = ۲\times۱۰^{-۲} \times ۰/۸\times۱۰^{۶} = ۱۶۰۰۰ \text{ bit} = ۲۰۰۰ \text{ Byte}$$

بنابراین دقیقاً در لحظه ارسال پیغام X-OFF از ایستگاه B به ایستگاه A، باید ۲۰۰۰Byte فضای خالی در ایستگاه B وجود داشته باشد تا پس از دریافت پیغام X-OFF توسط ایستگاه A و به تبع بسته شدن ارسال داده در ایستگاه A، بافر ایستگاه B سرریز نشود.

۲- کنترل جریان سخت‌افزاری RTS-CTS

در این روش فرستنده و گیرنده برای عمل کنترل جریان از یک کانال مجزای کنترل استفاده می‌کنند. سیگنال RTS (Request to send) به معنی درخواست فضا برای ارسال است که از طرف فرستنده برای گیرنده، هر زمان که نیاز به انتقال داده داشته باشد، ارسال می‌شود.

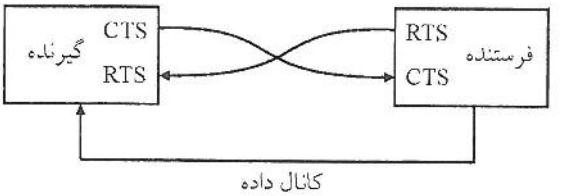
سیگنال CTS (Clear to send) به معنی آمادگی برای دریافت (خالی بودن بافر) است که از طرف گیرنده برای فرستنده ارسال می‌شود و این سیگنال‌ها می‌توانند مقادیر صفر و یک داشته باشند.

مکانیزم کاری دو ایستگاه فرستنده و گیرنده در این روش به دو شکل است:

۱- هر فرستنده که قصد ارسال داده برای گیرنده داشته باشد، یک سیگنال RTS با مقدار یک برای گیرنده می‌فرستد، گیرنده نیز موافقت خود را با ارسال سیگنال CTS با مقدار یک اعلام می‌کند و ارسال آغاز می‌شود.

۲- هرگاه گیرنده نیاز به دریافت داده داشته باشد، یک سیگنال CTS با مقدار یک به معنی درخواست برای دریافت اطلاعات برای فرستنده ارسال می‌کند و فرستنده موافقت خود را با ارسال یک RTS با مقدار یک اعلام می‌کند و ارسال داده آغاز می‌شود.

هرگاه بافر گیرنده در حال پُرشدن باشد، یک سیگنال CTS با مقدار صفر برای فرستنده ارسال می‌کند و فرستنده با دریافت آن، ارسال را متوقف می‌کند. گیرنده پس از خالی شدن بافر با یک سیگنال CTS با مقدار یک آمادگی خود را برای دریافت بقیه اطلاعات اعلام می‌نماید.



نکته: منظور از مقدار یک برای سیگنال، بالابردن سطح سیگنال و منظور از مقدار صفر سیگنال، پایین آمدن سطح سیگنال می‌باشد.

نکته: روش سخت‌افزاری نسبت به روش نرم‌افزاری سریع‌تر است زیرا به جای ارسال پیغام کنترلی روی کانال داده از سیگنال کنترلی صفر یا یک استفاده می‌کند. اما روش سخت‌افزاری هزینه بیشتری دارد و علاوه بر کانال داده نیاز به کانال کنترلی هم دارد. بنابراین فقط مناسب ارتباطات نزدیک است، زیرا چند خط اضافی برای سیگنال‌های کنترلی استفاده می‌شود که هزینه کابل‌کشی آن برای راه دور بالاست.

وظیفه ششم، مدیریت کانال انتقال

مدیریت کانال انتقال یکی از جامع‌ترین وظایف لایه پیوند داده است. منظور از مدیریت کانال انتقال، مشخص نمودن چگونگی برقراری ارتباط، جلوگیری و تشخیص خطا و نحوه دسترسی به کانال انتقال و کنترل جریان داده می‌باشد.

همان‌طور که در فصل اول بیان شد، شبکه‌ها به دو دسته شبکه‌های نقطه به نقطه و شبکه‌های پخش همگانی تقسیم می‌شوند. در شبکه‌های نقطه به نقطه بین دو ایستگاه فرستنده و گیرنده، کانال از پیش مشخص شده وجود دارد و دیگر ایستگاه‌ها به طور همزمان از این کانال استفاده نمی‌کنند، مدیریت کانال نقطه به نقطه ساده‌تر است زیرا یک کانال فقط در اختیار دو ایستگاه در حال ارتباط قرار دارد. در شبکه‌های پخش همگانی چندین ایستگاه از یک کانال مشترک استفاده می‌کنند و ممکن است در یک لحظه چند ایستگاه در حال استفاده از کانال باشند.

مدیریت کانال نقطه به نقطه

مدیریت کانال نقطه به نقطه ساده‌تر از مدیریت کانال پخش همگانی است. مدیریت کانال نقطه به نقطه به سه روش زیر انجام می‌گیرد:

۱- روش IDLE ARQ یا Stop & Wait (توقف و انتظار)

۲- روش Selective Repeat ARQ (تکرار انتخابی)

۳- روش GO Back N ARQ (برگشت به n فریم قبلی)

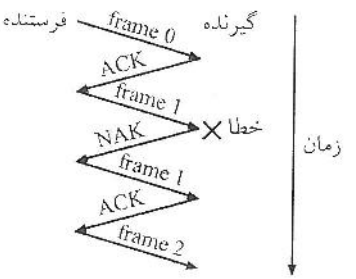
به این سه روش اصطلاحاً روش‌های ARQ (Automatic Repeat Request) یا درخواست تکرار خودکار گفته می‌شود. در سه روش بالا فرستنده اطلاعات را در قالب فریم‌های شماره‌گذاری شده و مرتب ارسال می‌نماید. گیرنده پس از دریافت هر کدام از این فریم‌ها، با توجه به مکانیزم کشف

خطای توافق شده بین فرستنده و گیرنده، فریم را بررسی می‌کند و در صورت عدم کشف خطا یک پیغام ACK (Acknowledge) که شماره فریم مذکور را در خود دارد و به سمت فرستنده ارسال می‌کند. در صورتی که فریم دریافتی دچار خطا شده باشد یک پیغام NACK همراه شماره فریم برای فرستنده ارسال می‌شود و فرستنده با دریافت آن، فریم مورد نظر را دوباره ارسال می‌کند.

اکنون به شرح سه روش ARQ می‌پردازیم:

۱- روش توقف و انتظار(IDLE ARQ یا Stop & Wait)

در این روش فرستنده بعد از ارسال هر فریم داده، منتظر پاسخ مناسب از طرف گیرنده می‌ماند فرستنده در این صورت تا مدتی (حداقل دو برابر فاصله زمانی بین فرستنده و گیرنده) بیکار می‌ماند. در صورتی که فرستنده پاسخ ACK از طرف گیرنده دریافت نماید، اقدام به ارسال فریم بعدی می‌کند ولی اگر پاسخ NACK دریافت کند، فریم مورد نظر را دوباره ارسال خواهد کرد. توجه داشته باشید در صورتی که در مدت زمان مشخصی پاسخی از طرف گیرنده برای دریافت خطادار یا بدون خطا دریافت نشود، فرستنده آخرین فریم را دوباره ارسال خواهد کرد. به شکل زیر توجه کنید:



نمودار stop & wait

☜ **نکته :** با توجه به توضیحات بالا متوجه خواهید شد که فرستنده و گیرنده تنها کافی است بافری با اندازه یک فریم داشته باشد.

توجه: مدت زمان لازم برای دریافت پیام تصدیق (ACK) یا عدم تصدیق (NACK)، حداقل دو برابر فاصله زمانی بین فرستنده و گیرنده ($2T_p$) است که به آن RTT (Round Trip Time) نیز گفته می‌شود.

تعریف پنجره زمانی فرستنده: به تعداد فریم‌هایی که یک فرستنده به شکل پشت سرهم و بدون دریافت پاسخ می‌تواند ارسال کند پنجره زمانی فرستنده گفته می‌شود.

در واقع فرستنده تا زمانی که پاسخ مناسب مبنی بر دریافت صحیح فریم ارسالی از طرف گیرنده را دریافت نکند، فریم ارسالی را در بافر خود (آماده برای ارسال مجدد) نگه می‌دارد.

تعریف پنجره زمانی گیرنده: به تعداد فریمی که گیرنده می‌تواند بدون ارسال پاسخ مناسب دریافت نماید، پنجره گیرنده گفته می‌شود.

توجه: پنجره فرستنده و گیرنده در روش stop & wait برابر با یک می‌باشد.

رابطه کلی برای محاسبه بهره‌وری در stop & wait (IDLE) به صورت زیر است:

محاسبه بهره‌وری stop & wait بدون صرف نظر کردن از سربار ACK و زمان پردازش به صورت زیر است:

$$U_{\text{stop \& wait}} = \begin{cases} \frac{1}{W_S}\left(1-\frac{H}{L}\right)(1-P_F) & W < W_S \quad (1) \\ \left(1-\frac{H}{L}\right)\times(1-P_F) & W \geq W_S \quad (2) \end{cases}$$

توجه: برای بهره‌وری ۱۰۰٪ باید $W \geq W_S$ باشد، بنابراین از رابطه دوم استفاده می‌گردد.

توجه: در روابط فوق مقدار پنجره فرستنده یعنی W برابر یک است.

توجه: W، اندازه پنجره سمت فرستنده و $W_S = \frac{T_O}{T_F}$ برابر با «حداقل» اندازه پنجره که برای «ماکزیمم کردن راندمان» مورد نیاز است.

T_O و T_F از رابطه‌ی زیر به دست می‌آید:

$$T_O = \text{Total Delay} = T_F + 2T_P + T_{ACK} + 2T_{Process}$$

$$T_F = \frac{L}{R}$$

T_F، زمان انتقال فریم داده به داخل کانال انتقال است.

که L برابر اندازه فریم و R برابر نرخ انتقال می‌باشد.

T_P، $T_{Process}$ و T_{ACK} از روابط زیر بدست می‌آید:

$$T_P = \frac{D}{V}$$

T_P، زمان تأخیر انتشار است.

که D برابر طول کانال و V برابر سرعت انتشار می‌باشد.

$$T_{Process} = \frac{b}{R}$$

$T_{Process}$، زمان پردازش مربوط به فریم‌بندی و فریم گشایی فریم داده و فریم ACK در مبدأ و مقصد است.

که b برابر تعداد بیت لازم برای پردازش و R برابر نرخ انتقال می‌باشد.

$$T_{ACK} = \frac{L_{ACK}}{R}$$

T_{ACK}، زمان انتقال فریم ACK به داخل کانال انتقال است.

که L_{ACK} برابر اندازه فریم ACK و R برابر نرخ انتقال می‌باشد.
محاسبه بهره‌وری stop & wait با صرف نظر کردن از سربار ACK و زمان پردازش به صورت زیر است:

$$W_S = \frac{T_F + ۲T_P + \overbrace{T_{ACK}}^{\text{ناچیز}} + ۲\overbrace{T_{Process}}^{\text{ناچیز}}}{T_F} = \frac{T_F + T_P}{T_F} = \frac{۱ + ۲\frac{T_P}{T_F}}{۱} = \frac{۱+۲a}{۱} = ۱+۲a$$

توجه: a برابر $\frac{T_P}{T_F}$ در نظر گرفته شده است.

بنابراین با توجه به روابط فوق داریم:

$$U_{stop\ \&\ wait} = \begin{cases} \frac{۱}{۱+۲a}\left(۱-\frac{H}{L}\right)(۱-P_F) & W < ۱+۲a \quad (۱) \\ \left(۱-\frac{H}{L}\right)(۱-P_F) & W \geq ۱+۲a \quad (۲) \end{cases}$$

توجه: برای بهره‌وری ۱۰۰٪ باید $W \geq ۱+۲a$ باشد، بنابراین از رابطه دوم استفاده می‌گردد.
توجه: در روابط فوق مقدار پنجره فرستنده یعنی W برابر یک می‌باشد.
توجه: در stop & wait، پنجره سمت فرستنده برابر ۱ و پنجره سمت گیرنده برابر ۱ و مجموع آن‌ها که تعداد شماره ترتیب‌های لازم را تشکیل می‌دهد ۱+۱ است. پس داریم:

$$\text{Sequence Number} = ۲^n = W + W = ۱+۱ = ۲$$

$$n = \text{Log}_۲^۲ = ۱\ \text{bit}$$

بنابراین تعداد بیت لازم برای مشخص کردن شماره ترتیب فریم‌ها برابر ۱ بیت است.
توجه: در روش stop & wait، برای هر فریمِ یک پیام تصدیق ACK یا عدم تصدیق NACK به شکل منفرد از سوی گیرنده به فرستنده ارسال می‌گردد.
توجه: در روش stop & wait، اگر خطا برای فریمی رخ دهد، روش کنترل خطای پس‌رو (BEC) به معنی ارسال مجدد فریم از سوی فرستنده استفاده می‌شود و گیرنده به جای پیام تصدیق ACK، پیام عدم تصدیق NACK به سمت فرستنده ارسال می‌کند، و فریم خطادار، مجدداً از سوی فرستنده ارسال می‌گردد.
توجه: احتمال خطا در هر بیت ارسالی برابر P_{bit} و احتمال خطا در یک فریم به طول L، با P_F نشان داده می‌شود. نحوه محاسبه P_F به صورت زیر است:

$$P_F = \begin{cases} ۱-(۱-P_{bit})^L & \text{رابطه دقیق} \\ L \times P_{bit} & \text{رابطه تقریبی اول} \\ ۱-e^{-LP_{bit}} & \text{رابطه تقریبی دوم} \end{cases}$$

مثال: فرستنده و گیرنده‌ای از پروتکل stop & wait ARQ برای انتقال اطلاعات استفاده می‌کنند. اگر نرخ انتقال Mbps ۲۰، اندازه فریم kByte ۲۰۰، طول کانال km ۶۰۰۰، سرعت انتشار ۲×۱۰^{۸} mps و تعداد بیت لازم برای پردازش برابر bit ۲ باشد، بهره‌وری کانال را محاسبه کنید:

پاسخ: داده‌های مسئله به صورت زیر است:

$$D = ۶\times۱۰^{۷}\,m\ ,\ L = ۲\times۱۰^{۵}\ Byte = ۱۶\times۱۰^{۵}\,bit\ ,\ R = ۲\times۱۰^{۷}\ bps\ ,\ V = ۲\times۱۰^{۸}\ mps\ ,\ b = ۲\ bit$$

$$T_{Process} = \frac{b}{R} = \frac{۲}{۲\times۱۰^{۷}} = ۱۰^{-۷}$$

$$T_P = \frac{D}{V} = \frac{۶\times۱۰^{۷}}{۲\times۱۰^{۸}} = ۳\times۱۰^{-۱}$$

$$T_F = \frac{L}{R} = \frac{۱۶\times۱۰^{۵}}{۲\times۱۰^{۷}} = ۸\times۱۰^{-۲}$$

$$T_O = T_F + ۲T_P + ۲T_{Process} = ۸\times۱۰^{-۲} + ۲\times۳\times۱۰^{-۱} - ۲\times۱۰^{-۷} = ۰/۰۸ + ۰/۶ + ۲\times۱۰^{-۷} \simeq ۰/۶۸$$

$$W_S = \frac{T_O}{T_F} = \frac{۰/۶۸}{۰/۰۸} = ۸/۵$$

بنابراین $W < W_S$ یعنی $۱ < ۸/۵$ است، لذا رابطه زیر برای محاسبه بهره‌وری مورد استفاده قرار می‌گیرد:

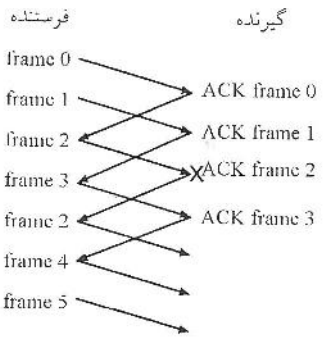
$$U_{stop\ \&\ wait} = \frac{۱}{W_S}(۱-\frac{H}{L})(۱-P_F) \qquad W < W_S$$

توجه: در صورت سؤال، صحبتی از درصد خطا (P_F) و سربار سرآیند ($\frac{H}{L}$) نشده است، بنابراین در رابطه فوق، مقدار آن، صفر در نظر گرفته می‌شود. پس داریم:

$$U_{stop\ \&\ wait} = \frac{۱}{۸/۵}\times۱۰۰ = ۱۱\%$$

۲- روش تکرار انتخابی (Selective Repeat ARQ یا Selective Reject)

در این روش فرستنده فریم‌های داده را پشت سرهم و بدون انتظار برای دریافت پاسخ ACK به سمت گیرنده ارسال می‌کند. گیرنده نیز با دریافت هر کدام از فریم‌ها، بعد از بررسی احتمال خطا، پاسخ مناسب را برای فرستنده ارسال می‌نماید. در صورتی که فرستنده یک پاسخ NACK (که حاوی شماره فریم مورد نظر است) دریافت کند، تنها فریم خطادار را دوباره ارسال می‌کند و بعد از آن به ارسال مابقی فریم‌ها می‌پردازد. به شکل زیر توجه کنید:



نمودار Selective Repeat

در این روش فرستنده می‌تواند تعداد W فریم را بدون دریافت پاسخ ACK پشت سر هم ارسال نماید(پنجره زمانی)، در واقع تصور شده است که بافر گیرنده به اندازه W فریم ظرفیت دارد. فریم‌های ارسالی که هنوز پاسخ ACK آنها دریافت نشده است در بافر فرستنده باقی می‌ماند تا احتمالاً در صورت دریافت پاسخ NACK، فرستنده بتواند بلافاصله دوباره آنها را ارسال نماید، در صورتی که W فریم از طرف فرستنده ارسال شود و از طرف گیرنده برای هیچ کدام ACK دریافت نشود، فرستنده باید عمل ارسال را متوقف کند (ارسال فریم جدید متوقف می‌شود و بعد از مدتی فریم‌های قبلی تکرار می‌شوند).

رابطه کلی برای محاسبه بهره‌وری در Selective Repeat (SR) به صورت زیر است:

محاسبه بهره‌وری Selective Repeat بدون صرف‌نظر کردن از سربار ACK و زمان پردازش به صورت زیر است:

$$U_{SR} = \begin{cases} \frac{W}{W_S}\left(1-\frac{H}{L}\right)(1-P_F) & W < W_S \quad (1) \\ \left(1-\frac{H}{L}\right)\times(1-P_F) & W \geq W_S \quad (2) \end{cases}$$

توجه: برای بهره‌وری ۱۰۰٪ باید $W \geq W_S$ باشد، بنابراین از رابطه دوم استفاده می‌گردد.

توجه: W، اندازه پنجره سمت فرستنده و $W_S = \frac{T_O}{T_F}$ برابر با «حداقل» اندازه پنجره که برای «ماکزیمم کردن راندمان» مورد نیاز است.

T_O و T_F از رابطه‌ی زیر به دست می‌آید:

$$T_O = \text{Total Delay} = T_F + 2T_P + T_{Ack} + 2T_{Process}$$

$$T_F = \frac{L}{R}$$

T_P، زمان انتقال فریم داده به داخل کانال انتقال است.

که L برابر اندازه فریم و R برابر نرخ انتقال می‌باشد.

T_P، $T_{Process}$ و T_{ACK} از روابط زیر به دست می‌آید:

$$T_P = \frac{D}{V}$$

T_P، زمان تأخیر انتشارات است.

که D برابر طول کانال و V برابر سرعت انتشار می‌باشد.

$$T_{Process} = \frac{b}{R}$$

$T_{Process}$، زمان پردازش مربوط به فریم‌بندی و فریم گشایی فریم داده و فریم ACK در مبدأ و مقصد است.

که b برابر تعداد بیت لازم برای پردازش و R برابر نرخ انتقال می‌باشد.

$$T_{ACK} = \frac{L_{ACK}}{R}$$

T_{ACK}، زمان انتقال فریم ACK به داخل کانال انتقال است.

که L_{ACK} برابر اندازه فریم ACK و R برابر نرخ انتقال می‌باشد.

محاسبه بهره‌وری Selective Repeat با صرف‌نظر کردن از سربار ACK و زمان پردازش به صورت زیر است:

$$W_S = \frac{T_F + ۲T_P + \overbrace{T_{ACK}}^{\text{ناچیز}} + ۲\overbrace{T_{Process}}^{\text{ناچیز}}}{T_F} = \frac{T_F + T_P}{T_F} = \frac{۱ + ۲\frac{T_P}{T_F}}{۱} = \frac{۱+۲a}{۱} = ۱+۲a$$

توجه: a برابر $\frac{T_P}{T_F}$ در نظر گرفته شده است.

بنابراین با توجه به روابط فوق داریم:

$$U_{SR} = \begin{cases} \frac{W}{۱+۲a}\left(۱-\frac{H}{L}\right)(۱-P_F) & W < ۱+۲a \quad (۱) \\ \left(۱-\frac{H}{L}\right)(۱-P_F) & W \geq ۱+۲a \quad (۲) \end{cases}$$

توجه: برای بهره‌وری ۱۰۰٪ باید $W \geq ۱+۲a$ باشد. بنابراین از رابطه دوم استفاده می‌گردد.

توجه: در Selective Repeat، پنجره سمت فرستنده برابر W و پنجره سمت گیرنده برابر W و مجموع آن‌ها که تعداد شماره ترتیب‌های لازم را تشکیل می‌دهد ۲W است. پس داریم:

$$\text{Sequence Number} = ۲^n = ۲ \times W_{SR}$$

$$n = \lceil \text{Log}_۲^{۲ \times W_{SR}} \rceil$$

توجه: W_{SR}، اندازه پنجره سمت فرستنده و گیرنده به شکل برابر است.

بنابراین تعداد بیت لازم برای مشخص کردن شماره ترتیب فریم‌ها برابر n بیت است.

در Selective Repeat براساس رابطه زیر:

$$\text{Sequence Number} = ۲^n = ۲ \times W_{SR}$$

داریم:

$$W_{SR} = \frac{۲^n}{۲} = ۲^{n-۱}$$

توجه: در روش Selective Repeat، برای هر فریم یک پیام تصدیق ACK یا عدم تصدیق NACK به شکل منفرد از سوی گیرنده به فرستنده ارسال می‌گردد. همچنین در این روش به تعداد معینی فریم پشت سر هم که به درستی به مقصد رسیده‌اند، می‌توان یک پیام تصدیق انبوه، یعنی یک ACK برای تعداد انبوهی از فریم‌های درست و متوالی از سوی گیرنده به فرستنده ارسال کرد.

توجه: در روش Selective Repeat، اگر خطا برای فریمی رخ دهد، روش کنترل خطای پس‌رو (BEC) به معنی ارسال مجدد فریم از سوی فرستنده استفاده می‌شود و گیرنده به جای پیام تصدیق ACK، پیام عدم تصدیق NACK برای فریم مورد نظر به سمت فرستنده ارسال می‌کند، و فریم خطادار، مجدداً از سوی فرستنده ارسال می‌گردد.

توجه: احتمال خطا در هر بیت ارسالی برابر P_{bit} و احتمال خطا در یک فریم به طول L، با P_F نشان داده می‌شود. نحوه محاسبه P_F به صورت زیر است:

$$P_F = \begin{cases} ۱-(۱-P_{bit})^L & \text{رابطه دقیق} \\ L \times P_{bit} & \text{رابطه تقریبی اول} \\ ۱-e^{-LP_{bit}} & \text{رابطه تقریبی دوم} \end{cases}$$

☜ **نکته:** در بین روش‌های ARQ، این روش بالاترین کارآیی را دارد.

مثال: کانالی از پروتکل Selective Repeat استفاده می‌کند، در صورتی که اندازه پنجره سمت فرستنده ۶ و نرخ ارسال ۱۰ Mbps و طول هر فریم ۱۰۰۰ bit باشد، بهره‌وری کانال با سرعت انتشار ۲×۱۰^۸ mps و طول ۲۰۰km چقدر خواهد بود؟

داده‌های مسئله به صورت زیر است:

$$D = ۲۰۰\ km = ۲۰۰\times۱۰^۳\ m\ ,\ L = ۱۰۰۰ bit\ ,\ R = ۱۰\ Mbps = ۱۰\times۱۰^۶\ bps$$

$$V = ۲\times۱۰^۸\ mps\ ,\ w = ۶$$

در صورت سؤال، صحبتی از سربار ACK و زمان پردازش نشده است، پس روابط زیر برای محاسبه بهره‌وری می‌تواند مورد استفاده قرار گیرد:

$$U_{SR} = \begin{cases} \frac{W}{۱+۲a}\left(۱-\frac{H}{L}\right)(۱-P_F) & W < ۱+۲a \quad (۱) \\ \left(۱-\frac{H}{L}\right)(۱-P_F) & W \geq ۱+۲a \quad (۲) \end{cases}$$

در مسئله گفته شده است، که اندازه پنجره سمت فرستنده برابر W=۶ است. حال باید بررسی کنیم

که W با ۱+۲a چه نسبتی دارد.

$$T_F = \frac{L}{R} = \frac{۱۰۰۰}{۱۰\times۱۰^۶} = ۱۰^{-۴}$$

$$T_P = \frac{D}{V} = \frac{۲۰۰\times۱۰^۳}{۲\times۱۰^۸} = ۱۰^{-۴}$$

$$۱+۲a = ۱+۲\times\frac{T_P}{T_F} = ۱+\frac{۱۰^{-۲}}{۱۰^{-۴}} = ۱+۲\times۱۰ = ۲۱$$

بنابراین $W < ۱+۲a$ یعنی ۲۱ > ۶است، لذا رابطه زیر برای محاسبه بهره‌وری مورد استفاده قرار می‌گیرد:

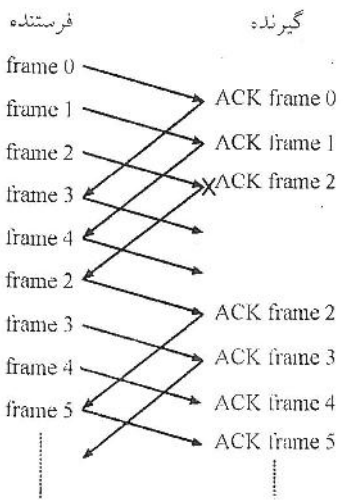
$$U_{SR} = \frac{W}{۱+۲a}\left(۱-\frac{H}{L}\right)(۱-P_F) \qquad W < ۱+۲a$$

توجه: در صورت سؤال صحبتی از درصد خطا (P_F) و سربار سرآیند ($\frac{H}{L}$) نشده است، بنابراین در رابطه فوق، مقدار آن، صفر در نظر گرفته می‌شود. پس داریم:

$$U_{SR} = \frac{W}{۱+۲a} = \frac{۶}{۲۱}\times۱۰۰ = ۲۸\%$$

۳- روش بازگشت به N فریم قبلی (Go Back N)

در این روش نیز فرستنده فریم‌های داده را پشت سرهم و بدون انتظار برای دریافت پاسخ مناسب (ACK) ارسال می‌کند. در صورتی که فرستنده پاسخ NACK (که در خود شماره فریم مورد نظر را دارد) دریافت کند، به عقب بر می‌گردد و فریم خطادار و تمام فریم‌های بعد از آن را یک بار دیگر ارسال می‌نماید. در این روش بافر سمت گیرنده به اندازه یک فریم و بافر سمت فرستنده به اندازه پنجره زمانی (W) ظرفیت دارد.



رابطه کلی برای محاسبه بهره‌وری در Go Back N (GBN) به صورت زیر است:
محاسبه بهره‌وری Go Back N بدون صرف نظر کردن از سربار ACK و زمان پردازش به صورت زیر است:

$$U_{\text{Go Back N}} = U_{GBN} = \begin{cases} \frac{W}{W_S}\left(1-\frac{H}{L}\right)\frac{(1-P_F)}{1+(W-1)P_F} & W < W_S \quad (1) \\ \left(1-\frac{H}{L}\right)\frac{(1-P_F)}{1+(W_S-1)P_F} & W \geq W_S \quad (2) \end{cases}$$

توجه: برای بهره‌وری ۱۰۰٪ باید $W \geq W_S$ باشد، بنابراین از رابطه دوم استفاده می‌گردد.

توجه: W، اندازه پنجره سمت فرستنده و $W_S = \frac{T_O}{T_F}$ برابر با «حداقل» اندازه‌ی پنجره که برای «ماکزیمم کردن راندمان» مورد نیاز است.

T_O و T_F از رابطه‌ی زیر به دست می‌آید:

$$T_O = \text{Total Delay} = T_F + 2T_P + T_{ACK} + 2T_{Process}$$

$$T_F = \frac{L}{R}$$

T_F، زمان انتقال فریم داده به داخل کانال انتقال است.
که L برابر اندازه فریم و R برابر نرخ انتقال می‌باشد.
T_P، $T_{Process}$ و T_{ACK} از روابط زیر بدست می‌آید:

$$T_P = \frac{D}{V}$$

T_P، زمان تأخیر انتشار است.
که D برابر طول کانال و V برابر سرعت انتشار می‌باشد.

$$T_{Process} = \frac{b}{R}$$

$T_{Process}$، زمان پردازش مربوط به فریم‌بندی و فریم گشایی فریم داده و فریم ACK در مبدأ و مقصد است.
که b برابر تعداد بیت لازم برای پردازش و R برابر نرخ انتقال می‌باشد.

$$T_{ACK} = \frac{L_{ACK}}{R}$$

T_{ACK}، زمان انتقال فریم ACK به داخل کانال انتقال است.
که L_{ACK} برابر اندازه فریم ACK و R برابر نرخ انتقال می‌باشد.
محاسبه بهره‌وری Go Back N با صرف نظر کردن از سربار ACK و زمان پردازش به صورت زیر است:

$$W_S = \frac{T_F + 2T_P + \cancel{T_{ACK}}^{\text{ناچیز}} + 2\cancel{T_{Process}}^{\text{ناچیز}}}{T_F} = \frac{T_F + 2T_P}{T_F} = \frac{1 + 2\frac{T_P}{T_F}}{1} = \frac{1+2a}{1} = 1+2a$$

توجه: a برابر $\frac{T_P}{T_F}$ در نظر گرفته شده است.

بنابراین با توجه به روابط فوق داریم:

$$U_{GBN} = \begin{cases} \frac{W}{1+2a}\left(1-\frac{H}{L}\right)\frac{(1-P_F)}{1+(W-1)P_F} & W < 1+2a \quad (1) \\ \left(1-\frac{H}{L}\right)\frac{1-P_F}{1+2aP_F} & W \geq 1+2a \quad (2) \end{cases}$$

توجه: برای بهره‌وری ۱۰۰٪ باید $W \geq 1+2a$ باشد، بنابراین از رابطه دوم استفاده می‌گردد.

$$1+(W_S-1)P_F \xrightarrow{W_S=1+2a} 1+(1+2a-1)P_F = 1+2aP_F$$

توجه: در Go Back N، پنجره سمت فرستنده برابر W و پنجره سمت گیرنده برابر ۱ و مجموع آن‌ها که تعداد شماره ترتیب‌های لازم را تشکیل می‌دهد. $W_{GBN}+1$ است، پس داریم:

$$\text{Sequence Number} = 2^n = W_{GBN} + 1$$

$$n = \left\lceil \text{Log}_2^{W_{GBN}+1} \right\rceil$$

توجه: W_{GBN} ، اندازه پنجره سمت فرستنده است.

بنابراین تعداد بیت لازم برای مشخص کردن شماره ترتیب فریم‌ها برابر n بیت است.

توجه: در Go Back N براساس رابطه زیر:

$$\text{Sequence Number} = 2^n = W_{GBN} + 1$$

داریم:

$$W_{GBN} = 2^n - 1$$

توجه: در روش Go Back N، برای هر فریم یک پیام تصدیق ACK یا عدم تصدیق NACK به شکل منفرد از سوی گیرنده به فرستنده ارسال می‌گردد. همچنین در این روش، به تعداد معینی فریم پشت سر هم که به درستی به مقصد رسیده‌اند، می‌توان یک پیام تصدیق انبوه، یعنی یک ACK برای تعداد انبوهی از فریم‌های درست و متوالی از سوی گیرنده به فرستنده ارسال کرد.

توجه: در روش Go Back N، اگر خطا برای فریمی رخ دهد، روش کنترل خطای پس‌رو (BEC) به معنی ارسال مجدد فریم از سوی فرستنده استفاده می‌شود و گیرنده به جای پیام تصدیق ACK، پیام عدم تصدیق NACK برای فریم موردنظر به سمت فرستنده ارسال می‌کند و فریم خطادار و تمام فریم‌های ارسال شده بعد از آن نیز مجدداً از سوی فرستنده ارسال می‌گردند.

توجه: احتمال خطا در هر بیت ارسالی برابر P_{bit} و احتمال خطا در یک فریم به طول L، با P_F

نشان داده می‌شود. نحوه محاسبه P_F به صورت زیر است:

$$P_F = \begin{cases} 1-(1-P_{bit})^L & \text{رابطه دقیق} \\ L \times P_{bit} & \text{رابطه تقریبی اول} \\ 1-e^{-LP_{bit}} & \text{رابطه تقریبی دوم} \end{cases}$$

مثال: دو گره از طریق یک پیوند ارتباطی با پهنای باند ۱ مگابیت بر ثانیه و تأخیر انتشار ۱۳۰ میلی‌ثانیه به هم متصل هستند. برای کنترل خطا از روش Go Back N ARQ با شماره ترتیب ۳ بیتی استفاده می‌کنند. اگر اندازه هر فریم ۲۵۰۰ بایت و نرخ خطای هر بیت ۰/۰۰۰۰۱ باشد آنگاه حداکثر نرخ ارسال مؤثر در این پیوند بر حسب کیلوبیت بر ثانیه تقریباً برابر است با:

۱) ۱۰۰۰ ۲) ۵۰۰ ۳) ۷۵۰ ۴) ۲۵۰

پاسخ: گزینه (۲) صحیح است.

گذردهی یا نرخ ارسال مؤثر یا نرخ انتقال واقعی از رابطه زیر به دست می‌آید:

$$R_e = R \times U$$

که R برابر مقدار نرخ انتقال اسمی (پهنای باند) و U برابر بهره‌وری می‌باشد. مقدار R برابر ۱ مگابایت می‌باشد، بنابراین در ادامه باید مقدار U محاسبه گردد.

توجه: در صورت سؤال، صحبتی از سربار ACK و زمان پردازش نشده است، بنابراین روابط زیر برای محاسبه بهره‌وری می‌تواند مورد استفاده قرار گیرد:

$$U_{GBN} = \begin{cases} \frac{W}{1+2a}\left(1-\frac{H}{L}\right)\left(\frac{1-P_F}{1+(W-1)P_F}\right) & W < 1+2a \quad (1) \\ \left(1-\frac{H}{L}\right)\frac{1-P_F}{1+2aP_F} & W \geq 1+2a \quad (2) \end{cases}$$

در مسئله گفته شده است که شماره ترتیب ارسال یک عدد ۳ بیتی است، بنابراین تعداد شماره ترتیب $2^3 = 8$ است.

در Go Back N، پنجره سمت فرستنده W و پنجره سمت گیرنده ۱ و مجموع آنها که تعداد شماره ترتیب‌های لازم را تشکیل می‌دهد W+۱ است. پس داریم:

$$\text{Sequence Number} = 8 = W_{GBN} + 1 \rightarrow W_{GBN} = 7$$

داده‌های مسئله به صورت زیر است:

$$R = 1\ \text{Mbps} = 1\times10^6\ \text{bps}\ ,\ T_p = 130\ \text{ms} = 130\times10^{-3}\ \text{s}\ ,\ n = 3\ \text{bit}$$

$$L = 2500\ \text{Byte} = 2500\times8\ \text{bit}\ ,\ P_F = 0/00001 = 10^{-5}$$

حال باید بررسی کنیم که W با 1+2a چه نسبتی دارد.

$$۱+۲a = ۱+۲\times\frac{T_P}{T_F} = ۱+۲\times\frac{۱۳۰\times۱۰^{-۳}}{\frac{۲۵۰۰\times۸}{۱۰^{۶}}} = ۱۴$$

بنابراین $W < ۱+۲a$ است، یعنی $۱۴ > ۷$. لذا رابطه زیر برای محاسبه بهره‌وری مورد استفاده قرار می‌گیرد:

$$U_{GBN} = \frac{W}{۱+۲a}\left(۱-\frac{H}{L}\right)\left(\frac{۱-P_F}{۱+(W-۱)P_F}\right) \qquad W < ۱+۲a$$

توجه: در صورت سؤال، صحبتی از سربار سرآیند $(\frac{H}{L})$ نشده است، بنابراین در رابطه فوق، مقدار آن، صفر در نظر گرفته می‌شود، پس داریم:

$$U_{GBN} = \frac{۷}{۱۴}\left(\frac{۱-۱۰^{-۴}}{۱+(۷-۱)\times۱۰^{-۴}}\right) = \frac{۷}{۱۴} = ۰/۵$$

حدوداً برابر یک است.

بنابراین مطابق رابطه گذردهی یا نرخ ارسال مؤثر یا نرخ انتقال واقعی، داریم:

$$R_e = R\times U = ۱۰^{۶}\times ۰/۵ = ۵۰۰\ \text{kbps}$$

حداقل شماره‌های ترتیب فریم‌ها در سه روش ARQ

stop & wait : برای این روش می‌توان حداقل از دو شماره برای ترتیب دادن به فریم‌های ارسالی استفاده نمود. فرض کنید فرستنده، فریم شماره ۰ را برای گیرنده ارسال می‌کند، گیرنده با دریافت داده صحیح، یک پاسخ ACK برای فرستنده ارسال می‌کند ولی به هر دلیلی پاسخ ACK بدست فرستنده نمی‌رسد، در این صورت فرستنده بعد از مدتی دوباره فریم شماره ۰ را ارسال می‌کند. پس اگر در این روش تنها از یک شماره (شماره ۰) برای ارسال استفاده شود، گیرنده نمی‌تواند تشخیص دهد که فریم دریافتی ، فریم جدید است یا تکرار فریم قبل و دچار اشتباه می‌شود.

Selective Repeat : برای این روش حداقل می‌توان از ۲W شماره استفاده کرد. W اندازه پنجره فرستنده است. فرض کند اگر فرستنده W فریم را پشت سر هم برای گیرنده ارسال نماید و گیرنده پس از دریافت صحیح تمام فریم‌ها پیغام ACK صادر کند و تمام ACKها در مسیر به هر دلیلی خراب شوند، فرستنده بعد از مدتی دوباره اقدام به ارسال این W فریم می‌کند، در این صورت گیرنده باید بفهمد که مجموعه جدید فریم‌های دریافت، تکرار مجموعه W فریم قبلی است یا مجموعه فریم جدید است.

GO Back N: در روش GO Back N حداقل تعداد شماره فریم‌ها باید $W+۱$ (۰ تا W) باشد. W اندازه پنجره فرستنده است. فرض کنید فرستنده W عدد فریم را ارسال نموده (۰.....W − ۱) و گیرنده با دریافت آنها برای تمام W فریم پاسخ ACK ارسال نموده و پاسخ‌های ACK در مسیر به هر دلیلی خراب شده و به دست فرستنده نرسیده‌اند. فرستنده بعد از مدتی با دریافت نکردن

پاسخ مناسب دوباره شروع به ارسال همان W فریم می‌کند. در این صورت اگر تعداد شماره فریم‌ها W عدد باشد، گیرنده با دریافت اولین فریم یعنی فریم شماره صفر نمی‌تواند تشخیص دهد که این مجموعه دریافتی یک مجموعه جدید و یا تکرار مجموعه قبلی است. در واقع چون گیرنده انتظار دارد مجموعه جدید دریافت کند، مجموعه W فریم دریافتی را به عنوان مجموعه جدید می‌شناسد و داده‌ها خراب می‌شوند.

مقایسه اندازه پنجره و تعداد شماره ترتیب روش‌های ARQ

جدول زیر اندازه پنجره گیرنده، پنجره فرستنده و تعداد شماره ترتیب لازم در روش پنجره لغزان را برای سه روش کنترل خطای متفاوت نشان می‌دهد:

پروتکل کنترل خطا	اندازه پنجره فرستنده	اندازه پنجره گیرنده	تعداد شماره ترتیب لازم
Stop & wait	1	1	$2^n = 2$
Selective-Repeat	W	W	$2^n = 2W$
Go-back-N	W	1	$2^n = W+1$

تذکر: مقدار n در جدول فوق برابر بیت‌های لازم برای شماره ترتیب پنجره‌ها می‌باشد.

مدیریت کانال پخش همگانی در فصل آینده توضیح داده می‌شود.