

۱-

الف) در FEC، فرستنده با استفاده از مکانیزم‌های خاص، قادر است خطاهای دریافت پیام را تشخیص داده و در صورت لزوم، با ارسال مجدد، آن‌ها را تصحیح کند. این مکانیزم‌ها از روش‌هایی مانند بیت پاریتی، کدگذاری همزمان، و استفاده از کدهای همرمز استفاده می‌کنند.

با استفاده از بیت پاریتی، فرستنده یک بیت اضافی به پیام اصلی اضافه می‌کند که نشان دهنده جمع بیت‌های پیام است. گیرنده دریافت کردن پیام را با جمع کردن بیت‌های دریافت شده بررسی می‌کند. در صورتی که جمع بیت‌ها با بیت پاریتی همخوانی نداشته باشد، خطا وجود دارد و گیرنده با ارسال مجدد درخواست تصحیح پیام را به فرستنده می‌دهد.

در کدگذاری همزمان، فرستنده اطلاعات را با استفاده از کدهای خاصی مانند کدهای همرمز، به صورت همزمان ارسال می‌کند. این کدها قادرند خطاهایی که در راه انتقال پیام ایجاد می‌شود را تشخیص داده و تصحیح کنند. گیرنده نیز با استفاده از کدهای همرمز، پیام را بازسازی کرده و در صورت وجود خطا، درخواست ارسال مجدد را به فرستنده می‌دهد.

در ARQ، گیرنده با استفاده از مکانیزم‌های خاص خطاها را تشخیص می‌دهد و برای دریافت پیام تصحیح شده، درخواست ارسال مجدد را به فرستنده ارسال می‌کند. این مکانیزم‌ها ممکن است شامل تأییدیه‌های مثبت و منفی (ACK/NACK) باشند. به طور معمول، فرستنده پس از دریافت NACK از گیرنده، پیام را مجدداً ارسال می‌کند تا گیرنده بتواند آن را به درستی دریافت کند.

با استفاده از ARQ، همچنین می‌توان از روش‌هایی مانند ARQ بازخوردی (ARQ with feedback) و ARQ توقف و انتظار (Stop-and-Wait ARQ) نیز استفاده کرد. در ARQ بازخوردی، گیرنده پس از دریافت پیام، تأییدیه‌ای را به فرستنده می‌دهد که نشان دهنده دریافت موفقیت‌آمیز پیام است. در صورتی که خطا وجود داشته باشد، گیرنده با ارسال NACK به فرستنده اعلام می‌کند که پیام دریافت نشده است و باید مجدداً ارسال شود. در ARQ توقف و انتظار، فرستنده پس از ارسال هر پیام، منتظر تأییدیه گیرنده می‌ماند و دریافت ACK از سوی گیرنده، نشان دهنده دریافت موفقیت‌آمیز پیام است. در صورتی که NACK دریافت شود یا تأییدیه دریافت نشود، فرستنده پیام را مجدداً ارسال می‌کند.

ب) در FEC سربار تعداد check bit است و در ARQ سربار زمان لازم برای retransmission است.

ج) عملکرد ARQ بهتر است چراکه باید سربار خیلی کمتری برای retransmission خواهیم داد.

د) معمولاً در کاربردهایی که یک کانال یک طرفه داریم مانند رادیو و تلویزیون از رویکرد FEC استفاده می‌شود. اما در کاربردهایی که نرخ خطا کمتر است و کانال دوطرفه داریم از ARQ استفاده می‌کنیم. مثل اینترنت.

ه) و و

سه روش اصلی کنترل خطا در ARQ عبارتند از:

1. Stop-and-Wait ARQ:

در این روش، فرستنده پس از ارسال هر بسته داده، منتظر دریافت تأییدیه (ACK) از گیرنده می‌ماند. در صورتی که ACK دریافت شود، فرستنده به بسته داده بعدی می‌رود. در صورتی که ACK دریافت نشود یا NACK دریافت شود، فرستنده مجدداً بسته داده را ارسال می‌کند. این روش ساده است اما باعث ایجاد تأخیر در ارسال داده‌ها می‌شود. عامل موثر در بهره‌وری این روش، زمان پاسخ گیرنده به فرستنده (ACK/NACK) است. اگر پاسخ به موقع داده شود، بهره‌وری افزایش می‌یابد.

2. Go-Back-N ARQ:

در این روش، فرستنده بسته‌های داده را به ترتیب شماره‌گذاری کرده و به صورت پیاپی ارسال می‌کند. گیرنده پس از دریافت بسته‌ها، آن‌ها را به ترتیب دریافت خود نگه می‌دارد و تأییدیه ACK برای بسته‌های دریافت شده را ارسال می‌کند. در صورت دریافت بسته‌ای با خطا، گیرنده NACK می‌فرستد و فرستنده از آن بسته به بعد همه بسته‌ها را مجدداً ارسال می‌کند. عامل موثر در بهره‌وری این روش، سائز پنجره پیام (پنجره ارسال) است. با افزایش سائز پنجره، بهره‌وری نیز افزایش می‌یابد، اما مصرف بیشتری از منابع شبکه دارد.

3. Selective Repeat ARQ:

در این روش، همچنین بسته‌های داده به ترتیب شماره‌گذاری می‌شوند، اما گیرنده بسته‌های دریافت شده را ذخیره می‌کند و تأییدیه ACK برای هر بسته دریافت شده را ارسال می‌کند. در صورت دریافت بسته‌ای با خطا، گیرنده NACK ارسال کرده و فرستنده فقط بسته خطا را مجدداً ارسال می‌کند. عامل موثر در بهره‌وری این روش، سائز پنجره پنجره دریافت است. با افزایش سائز پنجره، بهره‌وری نیز افزایش می‌یابد، اما نیاز به حافظه بیشتری دارد.

عوامل موثر در بهره‌وری هر سه روش عبارتند از:

زمان پاسخ گیرنده: گیرنده باید به سرعت و به موقع پاسخ ACK یا NACK را به فرستنده ارسال کند.
زمان تأخیر شبکه: زمانی که بسته‌ها برای رسیدن به مقصد از شبکه عبور می‌کنند، تأخیر زمانی می‌تواند بهره‌وری را تحت تأثیر قرار دهد. هرچه تأخیر کمتر باشد، بهره‌وری بیشتر خواهد بود.
اندازه پنجره: سائز پنجره پیام در روش‌های Go-Back-N و Selective Repeat تأثیر زیادی بر بهره‌وری دارد. با افزایش سائز پنجره، بهره‌وری افزایش می‌یابد، اما همچنین نیاز به منابع بیشتری دارد.

نرخ خطا: نرخ خطا در شبکه نیز می‌تواند بهره‌وری را تحت تأثیر قرار دهد. با افزایش نرخ خطا، تعداد بسته‌هایی که نیاز به ارسال مجدد دارند، افزایش می‌یابد و بهره‌وری کاهش می‌یابد.

ظرفیت شبکه: ظرفیت شبکه نیز تأثیر مهمی بر بهره وری دارد. در صورتی که ظرفیت شبکه کمتر از نرخ ارسال داده‌ها باشد، ممکن است بسته‌ها تأخیر داشته باشند و بهره وری کاهش یابد. در کل، بهبود بهره وری ARQ بستگی به تنظیم مناسب پارامترهای مختلف آن و بهینه‌سازی عوامل موثر در آن دارد.

-۲

(الف)

$$\begin{array}{r} 0001\ 0001\ 0000\ 0101 \\ +\ 0000\ 0010\ 0000\ 1001 \\ \hline =\ 0001\ 0011\ 0000\ 1110 \end{array}$$

$$(\text{One's complement}) = 1110\ 1100\ 1111\ 0001 = 0x\text{ECF1}$$

(ب)

$$\begin{array}{r} 0001\ 0000\ 0011\ 0100 \\ +\ 0010\ 1010\ 0010\ 0010 \\ +\ 0011\ 0100\ 0010\ 0101 \\ +\ 1111\ 1111\ 0011\ 0111 \\ \hline =\ 0110\ 1101\ 1011\ 0011 \end{array}$$

$$(\text{One's complement}) = 1001\ 0010\ 0100\ 1100 = 0x\text{924C}$$

-۳

(الف)

1 -> 6
23 -> 26

(ب)

6 -> 16
17 -> 22

(ج)

1. 32 or 33

$$2. 42 / 2 = 21$$

$$3. 26 / 2 = 13$$

(د)

در ۶ امین دور ارسال

(ه)

در ۱۶ امین دور ارسال، triple duplicate ACK رخ داده است.

در ۲۲ امین دور ارسال، timeout رخ داده است.

-۴

$$U1 \times R1 = U2 \times R2$$

$$U1 \times 20kbps = U2 \times 10kbps \rightarrow 2U1 = U2$$

$$2 \times 1 - H / L 1 + 2a1 = 1 - H / L 1 + 2a2$$

$$2 \times 1 / (1 + 2a1) = 1 / (1 + 2a2)$$

$$1 + 2a1 = 2 + 4a2$$

$$a = t_{3453} \times RL$$

$$L = 2 \times 3 \times 10^6 \text{ km} / v \times 20kbps$$

$$L = 1 + 4 \times 90 \text{ km} / v \times 10kbps$$

محاسبه L با فرض $v = 3 \times 10^7$:

$$L = 1 + 12 / L = 3988 \text{ bits}$$

-۵

(الف)

$$\text{Reno} \rightarrow \text{issthresh} = 21$$

$$\text{cw} = 23$$

$$\text{Thao} \rightarrow \text{issthresh} = 21$$

$$\text{cw} = 4$$

(ب)

$$\text{Reno} \rightarrow 26$$

$$\text{Thao} \rightarrow 21$$

(ج)

$$\text{Reno} \rightarrow 21 + 22 + 23 + 24 + 25 + 26 = 141$$

$$\text{Thao} \rightarrow 1 + 2 + 4 + 8 + 16 + 21 = 52$$

-۶

- $EstimatedRTT = ((1 - \alpha) \times EstimatedRTT) + (\alpha \times SampleRTT)$
- $DevRTT = ((1 - \beta) \times DevRTT) + (\beta \times |EstimatedRTT - SampleRTT|)$
- $Initial\ EstimatedRTT = 100ms$
- $Initial\ DevRTT = 5ms$

1. با فرض $\alpha = 0.125$ و $\beta = 0.25$:

- اگر $sampleRTT = 106$ میلی ثانیه باشد، محاسبه می کنیم:

- $EstimatedRTT = ((1 - 0.125) \times 100) + (0.125 \times 106) = 100.75$ میلی ثانیه
- $DevRTT = ((1 - 0.25) \times 5) + (0.25 \times |106 - 100|) = 5.25$ میلی ثانیه
- $Timeout\ Interval = 100.75 + 4 \times 5.25 = 121.75$ میلی ثانیه

2. با فرض $sampleRTT = 120$ میلی ثانیه:

- $EstimatedRTT = ((1 - 0.125) \times 100.75) + (0.125 \times 120) = 103.15$ میلی ثانیه
- $DevRTT = ((1 - 0.25) \times 5.25) + (0.25 \times |120 - 100.75|) = 8.75$ میلی ثانیه
- $Timeout\ Interval = 103.15 + 4 \times 8.75 = 138.15$ میلی ثانیه

3. با فرض $sampleRTT = 140$ میلی ثانیه:

- $EstimatedRTT = ((1 - 0.125) \times 103.15) + (0.125 \times 140) = 107.76$ میلی ثانیه
- $DevRTT = ((1 - 0.25) \times 8.75) + (0.25 \times |140 - 103.15|) = 15.77$ میلی ثانیه
- $Timeout\ Interval = 107.76 + 4 \times 15.77 = 170.62$ میلی ثانیه

4. با فرض $sampleRTT = 90$ میلی ثانیه:

- $EstimatedRTT = ((1 - 0.125) \times 107.76) + (0.125 \times 90) = 105.54$ میلی ثانیه
- $DevRTT = ((1 - 0.25) \times 15.77) + (0.25 \times |107.76 - 90|) = 16.27$ میلی ثانیه
- $Timeout\ Interval = 105.54 + 4 \times 16.27 = 170.62$ میلی ثانیه

5. با فرض $sampleRTT = 115$ میلی ثانیه:

- $EstimatedRTT = ((1 - 0.125) \times 105.54) + (0.125 \times 115) = 106.71$ میلی ثانیه
- $DevRTT = ((1 - 0.25) \times 16.27) + (0.25 \times |115 - 105.54|) = 14.57$ میلی ثانیه
- $Timeout\ Interval = 106.71 + 4 \times 14.57 = 165$ میلی ثانیه

- *EstimatedRTT*: [100.75, 103.15, 107.76, 105.54, 106.71] میلی‌ثانیه
- *DevRTT*: [5.25, 8.75, 15.77, 16.27, 14.57] میلی‌ثانیه
- *Timeout Interval*: [121.75, 138.15, 121.75, 170.62, 165] میلی‌ثانیه

-۷

پروتکل TCP برای برقراری ارتباطات اطمینان‌پذیر و قابل اعتماد در شبکه‌های کامپیوتری استفاده می‌شود. یکی از ویژگی‌های اساسی TCP، استفاده از شماره ترتیب اولیه (ISN) برای شروع ارتباطات TCP است.

شماره ترتیب اولیه (ISN) در TCP به عنوان یک مقدار تصادفی انتخاب می‌شود و برای شروع ترتیب بسته‌ها و تشخیص ارتباطات قبلی استفاده می‌شود. هدف اصلی از استفاده از ISN در TCP، جلوگیری از وقوع تداخل‌های بسته‌ها در ارتباطات TCP است.

با تغییر شماره ترتیب اولیه (ISN) در هر ارتباط جدید، احتمال تداخل بسته‌ها از ارتباطات قبلی کاهش می‌یابد. این به معنی این است که حتی اگر یک حمله‌کننده به روش‌هایی مانند تزریق بسته‌های مخرب (Spoofing) یا بازیابی ترتیب بسته‌ها (Replay Attack) سعی کند تلاش کند تا از بین ببرد یا نقض ارتباطات TCP را ایجاد کند، شناسایی این تلاش‌ها و محافظت از ارتباطات TCP ممکن می‌شود. به علاوه، تغییر شماره ترتیب اولیه (ISN) در هر ارتباط جدید باعث می‌شود که ترتیب تزریق بسته‌ها توسط حمله‌کننده پیچیده‌تر شود و تشخیص و جلوگیری از حملات SYN Flooding و TCP Session Hijacking سخت‌تر شود.

بنابراین، با تغییر شماره ترتیب اولیه (ISN) در هر ارتباط جدید، پروتکل TCP می‌تواند از حملات مختلفی مانند SYN Flooding و TCP Session Hijacking جلوگیری کند و امنیت ارتباطات TCP را بهبود بخشد.

-۸

(الف)

پروتکل TCP برای انتقال قابل اعتماد داده‌ها در شبکه از مجموعه‌ای از رویدادها (events) و اقدامات (actions) استفاده می‌کند.

1. برقراری ارتباط (Connection Establishment):

- رویداد: یک میزبان (Host) تلاش می‌کند برای برقراری ارتباط TCP با میزبان دیگر.
- اقدامات:

- میزبان فرستنده (Sender) ارسال یک بسته SYN (Synchronize) حاوی شماره ترتیب اولیه (ISN) به میزبان گیرنده (Receiver).
- میزبان گیرنده دریافت بسته SYN و ارسال یک بسته SYN-ACK حاوی شماره ترتیب اولیه خود و تأیید دریافت بسته SYN به میزبان فرستنده.
- میزبان فرستنده دریافت بسته SYN-ACK و ارسال یک بسته ACK (Acknowledge) حاوی شماره ترتیب درست برای برقراری ارتباط به میزبان گیرنده.

2. انتقال داده‌ها (Data Transmission):

- رویداد: میزبان فرستنده می‌خواهد داده‌ها را به میزبان گیرنده ارسال کند.
- اقدامات:
- میزبان فرستنده تقسیم داده‌ها به بسته‌های کوچکتر (segmentation) و ارسال هر بسته به میزبان گیرنده.
- میزبان گیرنده دریافت بسته‌ها و تأیید دریافت آن‌ها با ارسال بسته‌های ACK به میزبان فرستنده.
- در صورتی که میزبان فرستنده بسته‌ها را به طور کامل دریافت نکند، مجدداً درخواست برای ارسال بسته‌های گمشده را ارسال می‌کند (Selective Repeat و ARQ).

3. کنترل جریان (Flow Control):

- رویداد: میزبان گیرنده قادر به پذیرش داده‌های ورودی نیست یا از سرعت بالایی برای دریافت داده‌ها شکایت می‌کند.
- اقدامات:
- میزبان گیرنده ارسال بسته‌های ACK به میزبان فرستنده برای نشان دادن توانایی دریافت داده‌ها.
- میزبان فرستنده دریافت بسته‌های ACK و بر اساس آن‌ها میزان ارسال داده‌ها را تنظیم می‌کند تا جلوی سربار و اشباع شبکه را بگیرد.

4. تشخیص و بازیابی خطا (Error Detection and Recovery):

- رویداد: بسته‌های ارسالی توسط میزبان فرستنده یا میزبان گیرنده دچار خطا شده‌اند.
- اقدامات:
- استفاده از مجموعه‌ای از کدگذاری و چک‌های تأیید (checksum) برای تشخیص خطا در بسته‌ها.
- در صورت تشخیص خطا، میزبان گیرنده از میزبان فرستنده درخواست برای ارسال مجدد بسته‌های خراب را می‌کند.
- میزبان فرستنده بسته‌های خراب را مجدداً ارسال می‌کند تا دریافت صحیح انجام شود.

5. قطع اتصال (Connection Termination):

- رویداد: میزبان یکی از طرفین می‌خواهد ارتباط TCP را قطع کند.
- اقدامات:
- میزبان فرستنده ارسال بسته FIN (Finish) به میزبان گیرنده برای نشان دادن تمایل به قطع ارتباط.
- میزبان گیرنده دریافت بسته FIN و ارسال یک بسته ACK به میزبان فرستنده برای تأیید دریافت بسته FIN.

- میزبان گیرنده نیز ارسال یک بسته FIN به میزبان فرستنده.
- میزبان فرستنده دریافت بسته FIN و ACK و ارسال یک بسته ACK برای تأیید دریافت بسته FIN.

با استفاده از این رویدادها و اقدامات، پروتکل TCP توانایی انتقال قابل اعتماد داده‌ها در شبکه را فراهم می‌کند. تشخیص و بازیابی خطا، کنترل جریان، برقراری و قطع اتصال بهبودهایی است که TCP ارائه می‌دهد تا ارتباطات شبکه را پایدارتر و قابل اعتمادتر سازد.

(ب)

