تمرین پنجم شبکه‌های کامپیوتری - بهار ۱۴۰۲

کیمیا منتظری ۹۹۳۱۰۷۸

سوال ۱

در رابطه با دو رویکرد کنترل خطای FEC (Forward Error Control) و ARQ (Automatic Repeat reQuest) به سوالات زیر پاسخ دهید.

الف) به طور اجمالی هر یک از این رویکردها را شرح دهید.

در رویکرد FEC، فرستنده با مکانیزم‌های مشخصی متوجه می‌شود که گیرنده پیام را به درستی دریافت نکرده است. در نتیجه، گیرنده خود نیز با ارسال مجدد، خطا را تصحیح می‌کند.

در رویکرد ARQ، گیرنده با مکانیزم‌های مشخصی خطا را تشخصی می‌دهد و به منظور دریافت پیام تصحیح شده، از فرستنده درخواست ارسال مجدد می‌کند.

ب) سربار اصلی در هر یک از رویکردها چیست؟

سربار اصلی در رویکرد FEC تعداد check bit بوده، در حالی در رویکرد ARQ، این سربار به زمان لازم برای retransmission وابسته است.

ج) اگر نرخ خطا کم باشد، عملکرد ARQ بهتر است یا FEC؟

عملکرد رویکرد ARQ زمانی که نرخ خطا کم باشد بهتر است زیرا در این صورت سربار کمتری برای retransmission خواهیم داشت.

د) موادر کاربرد هر یک از این رویکردها را با ذکر چند مثال شرح دهید.

در کانال‌های یک‌طرفه، مانند تلویزیون، تنها می‌توان از رویکرد FEC استفاده کرد.

در شبکه‌هایی که نرخ خطا کم است و ارتباط دوطرفه است (شبکه‌های کامپیوتری)، مانند شبکه اینترنت، رویکرد ARQ عملکرد بهتر و کاربرد بیشتری دارد.

ه) سه روش اصلی کنترل خطای ARQ را نام برده و هر یک را به صورت مختصر شرح دهید.

روش‌های کنترل خطای ARQ شامل Stop & Wait (SW) ، Go-Back-N (GBN)، و Selective Repeat (SR) می‌باشند.

- **SW**: فرستنده یک بسته را ارسال می‌کند و منتظر دریافت ack آن می‌ماند. زمان انتظار نیز یک تخمین از زمان RTT بوده که در صورت timeout شدن تایمر، فرستنده همان بسته را دوباره ارسال می‌کند.
- **GBN**: فرستنده n بسته (به اندازه سایز بافر ارسال یا به عبارتی، به اندازه W_Sتا بسته) ارسال کرده و بعد از آن، منتظر دریافت ack می‌ماند. در این صورت بهره‌وری بیشتری نسبت به SW خواهیم داشت و زمان کمتری هدر می‌رود.
  لازم به ذکر است که در این روش، فرستنده باید یک کپی از بسته ارسالی را، تا زمانی که ack مربوط به آن را دریافت کند، در بافر ارسال خود نگه دارد. در صورت عدم دریافت ack و رخداد timeout، فرستنده بسته مربوط به ackای که دریافت نکرده است و بسته‌هایی که بعد از آن ارسال کرده بود را مجددا ارسال می‌کند.

این روش مانند pipelining عمل می‌کند.

- **SR**: در روش GBN هنگام رخداد خطا در ارسال یک بسته، لازم است تا بسته‌های بعد از آن را هم دوباره ارسال کنیم. زیرا گیرنده این بسته‌ها را به دلیل out of order بودن شماره ترتیبشان، دور ریخته است. در نتیجه، انگیزه‌ای که برای بهبود پروتکل GBN به وجود می‌آید این است که به گیرنده اجازه دریافت بسته‌های خارج از ترتیب را بدهیم تا سربار بازارسال کردن این بسته‌ها را نداشته باشیم. بدین منظور، یک پنجره یا بافر دریافت به اندازه W_R نیز برای گیرنده تعریف می‌کنیم که بیانگر این است که گیرنده می‌تواند بسته‌هایی با شماره ترتیب $N(R)$ تا $N(R) + W_R - 1$ را دریافت کند.

و) عوامل موثر در بهره‌وری هر یک از روش‌های قسمت (ه) را ذکر کنید.

- S&W
  - عامل سربار $\frac{H}{L}$ (که بستگی به پروتکل دارد)
  - عامل سربار ack
  - عامل Delay Bandwidth Product (نرخ ارسال × تاخیر رفت‌وبرگشت) یا $\frac{RTT}{\frac{L}{R}}$
  - عامل خطا (ارسال ناموفق با احتمال p_F)

$$U_{S\&W} = \frac{R_{eff}}{R} = \frac{1 - \frac{H}{L}}{1 + \frac{A}{L} + 2a}(1 - p_F)$$

- GBN
  - عامل سربار $\frac{H}{L}$ (در حالتی که خطا نداشته باشیم، بهره‌وری نزدیک ۱۰۰ درصد خواهد بود)
  - عامل Delay Bandwidth Product که در رابطه بهره‌وری، همان $(W_s - 1)p_F$ می‌باشد. این عامل در روش GBN تاثیر کمتری دارد زیرا برخلاف روش SW، تنها در صورت رخداد خطا تاثیرگذار است.
  - عامل خطا (اگر خطا داشته باشیم، باید به اندازه پنجره ارسال، بسته retransmit کنیم)

$$U_{GBN} = \frac{R_{eff}}{R} = \frac{1 - \frac{H}{L}}{1 + (W_s - 1)p_F}(1 - p_F)$$

- SR: س
  - عامل سربار $\frac{H}{L}$ (این مقدار خیلی کم است. در نتیجه، بهره‌وری نزدیک ۱۰۰ درصد خواهد بود)
  - عامل خطا

$$U_{SR} = \frac{R_{eff}}{R} = (1 - \frac{H}{L})(1 - p_F)$$

در نتیجه، **در شرایط برابر**، بهره‌وری Selective Repeat بهتر است از دو روش دیگر.

سوال ۲

برای هر یک از payloadهای زیر checksum را بدست آورید.

الف) 0x1105 | 0x0209

$0x1105 \equiv 0001\ 0001\ 0000\ 0101$

$0x209 = 0000\ 0010\ 0000\ 1001$

$0x1105 + 0x209 = 0001\ 0011\ 0000\ 1110$

$\text{Checksum (1's complement)} = 1110\ 1100\ 1111\ 0001 \equiv 0xECF1$

ب) 0x1034 | 0x2A22 | 0x3425 | 0xFF37

$0x1034 \equiv 0001\ 0000\ 0011\ 0100$

$0x2A22 \equiv 0010\ 1010\ 0010\ 0010$

$0x3425 \equiv 0011\ 0100\ 0010\ 0101$

$0xFF37 \equiv 1111\ 1111\ 0011\ 0111$

Adding all the above numbers using *wrap around sum* gives us:

$0110\ 1101\ 1011\ 0011$

$\text{Checksum (1's complement)} = 1001\ 0010\ 0100\ 1100 \equiv 0x924C$

سوال ۳

الف) بازه‌های زمانی را که پروتکل در حالت Slow Start کار می‌کند، را مشخص کنید.

[1, 6] and [23, 26]

ب) بازه‌های زمانی را که پروتکل در حالت Congestion Avoidance کار می‌کند را مشخص کنید.

[6, 16] and [17, 22]

ج) مقدار متغیر Slow Start Threshold (ssthresh) را در زمان‌های زیر تعیین کنید:

- **آغاز به کار پروتکل**

$ssthresh = 32\ or\ 33$

- **در ۱۸امین دور ارسال**

در ۱۶امین دور ارسال، از حالت congestion avoidance به حالت fast recovery می‌رویم و در نتیجه خواهیم داشت:

$$ssthresh = \frac{cw}{2} = \frac{42}{2} = 21$$

در ۱۸امین دور ارسال نیز، مقدار ssthresh ثابت و برابر با ۲۱ خواهد بود.

- **در ۲۴امین دور ارسال**
- در ۲۲مین دور ارسال، از حالت congestion avoidance به حالت fast recovery می‌رویم و در نتیجه خواهیم داشت:

$$ssthresh = \frac{cw}{2} = \frac{26}{2} = 13$$

در ۲۴مین دور ارسال نیز، مقدار ssthresh ثابت و برابر با ۱۳ خواهد بود.

د) در کدام دوره بسته شماره ۶۰ ارسال می‌شود؟

در ۶مین دور ارسال

ه) در کدام زمان Packet Loss رخ داده است؟ برای هر کدام مشخص کنید که از دست دادن بسته با استفاده از Triple Duplicate ACK شناسایی شده است یا Timeout؟

در ۱۶مین و ۲۲مین دور ارسال packet loss رخ داده است.

در ۱۶مین دور ارسال، مقدار پنجره ازدحام نصف شده است. پس triple duplicate ACK رخ داده است.

در ۲۲مین دور ارسال، مقدار پنجره ازدحام برابر با ۱ شده است. پس timeout رخ داده است.

سوال ۴

می‌خواهیم با استفاده از پروتکل Stop & Wait یک فایل بزرگ را از گره A به گره B با فاصله ۹۰ کیلومتر منتقل کنیم. اگر از یک ارتباط ماهواره‌ای با نرخ ۲۰ کیلوبیت‌برثانیه استفاده کنیم، اندازه هر بسته تقریبا چقدر باید باشد تا نرخ موثر ارسال اطلاعات از طریق ماهواره معادل نرخ موثر ارسال از طریق یک خط تلفن با نرخ ۱۰ کیلوبیت‌برثانیه باشد (فرض کنید طول کل لینک ماهواره‌ای بین مبدا و مقصد برابر با ۳۰۰۰۰ کیلومتر است)؟

$$R_{eff,1} = R_{eff,2} \rightarrow U_1 \times R_1 = U_2 \times R_2$$

$$U_1 \times 20kbps = U_2 \times 10kbps \rightarrow 2U_1 = U_2$$

$$2 \times \frac{1 - \frac{H}{L}}{1 + 2a_1} = \frac{1 - \frac{H}{L}}{1 + 2a_2}$$

$$2 \times \frac{1}{1 + 2a_1} = \frac{1}{1 + 2a_2}$$

$$1 + 2a_1 = 2 + 4a_2$$

$$we\ know\ that: a = \frac{t_{prop} \times R}{L} \rightarrow 2 \times \frac{\frac{3 \times 10^4 km}{v} \times 20kbps}{L} = 1 + 4 \times \frac{\frac{90km}{v} \times 10kbps}{L}$$

$$if\ v = 3 * 10^8 \rightarrow \frac{4000}{L} = 1 + \frac{12}{L}$$

$$then\ L = 3988\ bit$$

سوال ۵

در یک اتصال TCP در بازه زمانی ۰ تا ۲۶، رخدادهای زیر اتفاق افتاده است:

- **سه پیام تایید تکراری در ۱۶امین دوره دریافت شده است.**
- **در ۲۲امین دوره یک Timeout رخ می‌دهد.**

با فرض اینکه آستانه ازدحام اولیه ssthresh = 32 MSS است. نمودار اندازه پنجره ازدحام بر اساس دوره زمانی را برای TCP Tahoe و TCP Reno رسم کنید و به سوالات زیر پاسخ دهید:

الف) مقدار ssthresh و اندازه پنجره ازدحام در ۱۹امین دوره چقدر است؟

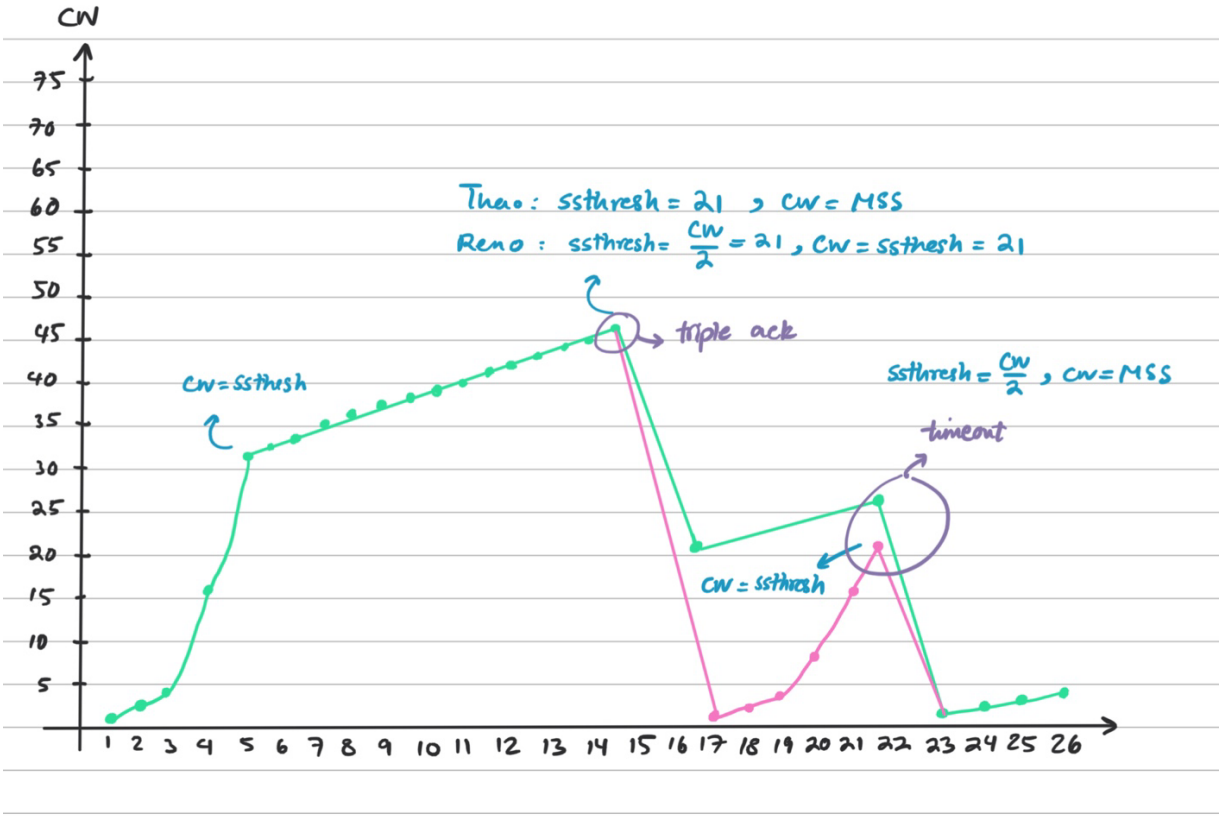
$$\begin{cases} Reno \rightarrow \begin{cases} ssthresh = 21 \\ cw = 23 \end{cases} \\ Thao \rightarrow \begin{cases} ssthresh = 21 \\ cw = 4 \end{cases} \end{cases}$$

ب) تعداد کل بسته‌های ارسال شده در ۲۲امین دوره چقدر است؟

$$\begin{cases} Reno \rightarrow 26 \\ Thao \rightarrow 21 \end{cases}$$

ج) تعداد کل بسته‌های ارسال شده از ۱۷امین دوره تا ۲۲امین دوره چقدر است؟

$$\begin{cases} Reno \rightarrow 21 + 22 + 23 + 24 + 25 + 26 = 141 \\ Thao \rightarrow 1 + 2 + 4 + 8 + 16 + 21 = 52 \end{cases}$$



سوال ۶

$We\ know\ that$:

- $EstimatedRTT = ((1-\alpha)\cdot EstimatedRTT) + (\alpha \cdot SampleRTT)$
- $DevRTT = ((1-\beta)\cdot DevRTT) + (\beta \cdot |EstimatedRTT - SampleRTT|)$
- $Timeout\ Interval = \overline{RTT} + k\sigma_{RTT} \xrightarrow{k=4} EstimatedRTT + 4 \cdot DevRTT$
- $Initial\ EstimatedRTT = 100ms$
- $Initial\ DevRTT = 5ms$

$After\ first\ SampleRTT = 106ms, we\ have$:

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

$After\ second\ SampleRTT = 120ms, we\ have$:

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

$After\ third\ SampleRTT = 140ms, we\ have$:

- $EstimatedRTT = ((1-0.125)(103.15)) + (0.125 \times 140) = 107.76ms$
- $DevRTT = ((1-0.25)(8.75)) + (0.25 \times |140-103.15|) = 15.77ms$
- $Timeout\ Interval = 107.76 + 4 \times 15.77 = 121.75ms$

$After\ fourth\ SampleRTT = 90ms, we\ have$:

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

$After\ fifth\ SampleRTT = 115ms, we\ have$:

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

سوال ۷

چرا پروتکل TCP مقدار ISN (شماره ترتیب اولیه) را از یک ارتباط به ارتباط دیگر تغییر می‌دهد؟ با این کار، TCP از چه خطایی جلوگیری می‌کند؟

حالتی را در نظر بگیرید که یک connection زمانی بسته شود که هنوز تعدادی از بسته‌ها توسط گیرنده دریافت نشده‌اند. حال، اگر یک connection دیگر میان همین گیرنده و فرستنده برقرار شود، ممکن است بسته‌های ارسال نشده، در ارتباط جدید ارسال شوند. اگر شماره ترتیب اولیه در هر connection یکسان باشد، connection دوم متوجه نمی‌شود که این بسته‌ها برای ارتباط قبلی بوده‌اند. با تغییر دادن ISN برای هر ارتباط، بسته‌های مربوط به connectionهای دیگر را می‌توان تشخیص داد.

سوال ۸

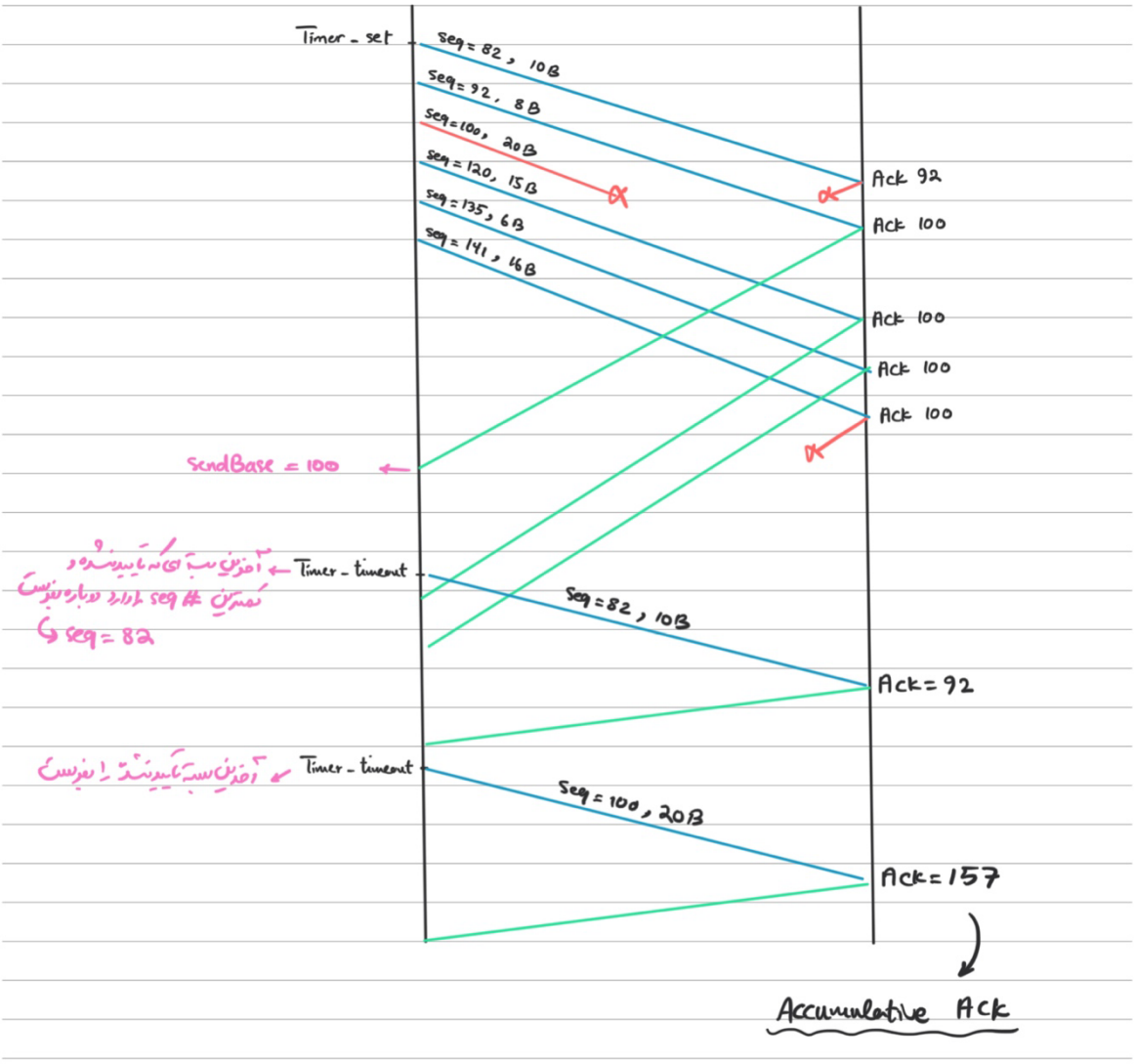
الف) عملکرد پروتکل TCP را برای انتقال قابل اعتماد داده‌ها در قبال رویدادهای مختلف توضیح دهید (Event / Action).

Event	Action
data received from application	1. Create a TCP segment and set `SeqNum = NextSeqNum` 2. If no timer is currently running, start the timer 3. Pass the segment to IP layer 4. Set `NextSeqNum += len(data)`
timer timeout	1. Retransmit the not-yet-acknowledged segment with smallest `SeqNum` 2. Start the timer
ACK received with value *y*	1. If `y > SendBase`, do actions 2 and 3, else do actions 4 and 5 2. Set `SendBase = y` 3. If there exists any not-yet-acknowledged segments, start the timer 4. `NumOfDupAcks++` 5. If `NumOfDupAcks == 3`, then resend the segment with `SeqNum = y`

ب) دیتاگرام زمانی زیر، انتقال مطمئن داده‌ها توسط پروتکل TCP را نشان می‌دهد. با توجه به عملکرد پروتکل TCP، آن را کامل کنید.

به دلیل آنکه سومین ACK = 100 به فرستنده ارسال نشده است، نمی‌توان از مکانیزم Triple Duplicate ACK استفاده کرد و timeout شدن تایمر تنها راه برای retransmit کردن داده با شماره ۱۰۰ است.

اگر در گیرنده، بافر دریافت داشته باشیم (روش selective repeat) آنگاه دیاگرام به صورت زیر خواهد بود (داده‌های ۱۲۰، ۱۳۵، و ۱۴۱ در بافر دریافت نگهداری می‌شوند و زمانی که داده ۱۰۰ دریافت شود، یک ACK تجمعی به منظور اینکه «داده‌ها تا شماره ۱۴۱ تایید شده‌اند» به فرستنده ارسال می‌شود). این روش در TCP پیاده‌سازی شده است:



در غیر این صورت، مانند روش Go-back-n عمل کرده و لازم می‌شود تا داده‌های ۱۲۰، ۱۳۵، و ۱۴۱ نیز دوباره ارسال شوند: