



دانشکده مهندسی کامپیوتر

بسمه تعالی
دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

درس شبکه های کامپیوتری، نیم سال دوم سال تحصیلی ۱۴۰۱-۱۴۰۲

پایخ تمرین سری پنجم

پاسخ سوال ۱:

(الف)

FEC یک رویکرد کنترل خطا است که اطلاعات اضافی مانند check bit را به بسته اضافه می کند و از این طریق گیرنده می تواند با چک کردن این داده های اضافی، از صحت بسته اطمینان حاصل کند. (بسته به آن که چند بیت برای صحت سنجی اختصاص داده شود، دقت این رویکرد برای تشخیص خطا متفاوت خواهد بود).

ARQ یک رویکرد کنترل خطا است که شامل ارسال مجدد داده هایی می شود که خراب شده اند یا از بین رفته اند. (به طور کلی به صورت درست به مقصد نرسیده اند).

فرستنده داده را می فرستد و تنها در زمانی داده مجدداً ارسال نمی شود که تاییدی از سمت گیرنده مبنی بر دریافت داده (ack)، به دست فرستنده برسد. اگر این تایید دریافت نشد، بسته مجدداً ارسال می گردد. (زمان بین ارسال قبلی و جدید توسط timer مشخص می شود).

(ب)

در رویکرد FEC (Control Error Forward) سربار اصلی check bit ها هستند.

در رویکرد ARQ (reQuest Repeat Automatic) سربار اصلی ارسال مجدد داده ها است.

(ج)

در صورتی که نرخ خطا کم باشد، عملکرد ARQ بهتر از FEC است. زیرا در رویکرد FEC، در هر دو شرایط روز یا عدم بروز خطا، هزینه سربار پرداخته می شود.

(د)

در شرایطی که احتمال خطا زیاد باشد و هزینه ارسال مجدد بالا باشد، FEC گزینه ی بهتری است.

و نیز در ارتباطات یکطرفه مانند رادیو و تلویزیون تنها رویکرد FEC قابل استفاده است. زیرا ARQ نیاز به ارتباط دو طرفه دارد.

در مواردی مثل شبکه internet، ARQ گزینه ی بهتری است. زیرا اگر نرخ خطا کم باشد، سربار ARQ کمتر از FEC است.

(ه)

۱- Stop-and-Wait ARQ: در این روش، بعد از ارسال هر بسته، مبدا منتظر دریافت تأییدیه از مقصد می ماند. در صورتی که مقصد بسته را با موفقیت دریافت کرد، تأییدیه را به مبدا می فرستد و مبدا بسته بعدی را ارسال می کند. در صورتی که مقصد بسته را با خطا دریافت کرد، تأییدیه ارسال نمی شود و مبدا بسته قبلی را مجدداً ارسال می کند.



۲- Go-Back-N ARQ: در این روش، مبدا می‌تواند چندین بسته را پشت سر هم برای فرستنده ارسال کند. در صورتی که بسته‌ای با خطا دریافت شود، تمام بسته‌های که پس از آن بسته ارسال شده‌اند مجدداً ارسال می‌شوند.

۳- Selective Repeat ARQ: مانند روش Go-Back-N ARQ: عمل می‌کند با این تفاوت که در صورت بروز خطا در دریافت یک بسته، فقط همان بسته بازارسال می‌گردد. و بسته‌هایی که شماره ترتیب آن‌ها جلوتر بوده است ولی توسط گیرنده دریافت شده‌اند، حذف می‌شوند و دور ریخته نمی‌شوند.

و

۱- Stop-and-Wait ARQ:

- سر بار header بسته $(\frac{H}{L})$
- سر بار ack $(\frac{A}{L})$
- Delay bandwidth product (ضریب تاخیر \times پهنای باند)

۲- Go-Back-N ARQ:

- سر بار header بسته $(\frac{H}{L})$
- عامل خطا (P_f)
- Delay bandwidth product (ضریب تاخیر \times پهنای باند)

۳- Selective Repeat ARQ:

- سر بار header بسته $(\frac{H}{L})$
- عامل خطا (P_f)

پاسخ سوال ۲:

روش محاسبه IP Checksum به شرح زیر است:

۱- جمع اعداد ۱۶ بیتی به روش carry bit wrap around برای اطمینان از اینکه نتیجه حاصل جمع همواره ۱۶ بیتی است. (در صورت تولید carry، رقم نقلی با حاصل جمع، جمع خواهد شد.)

۲ - مکمل یک کردن نتیجه حاصل جمع نهایی (معکوس کردن بیت‌ها)

الف)

برای اعداد: $b_1 = 0x1105$ و $b_2 = 0x0209$



Binary => 0001000100000101 + 00000001000001001 = 0001001100001110 with carry 0

After addition with carry => 0001001100001110 + 0 = 0001001100001110

Ones' complement => 1110110011110001

برای سادگی در نوشتار همین محاسبات را به صورت هگزادسیمال نیز می توان انجام داد (نتیجه نهایی یکسان خواهد بود). داریم :

Hex => 1105 + 0209 = 130E with carry 0

After addition with carry => 130E + 0 = 130E

Ones' complement => ECF1

(ب)

برای اعداد: $b_4 = 0xFF37$ و $b_3 = 0x3425$ و $b_2 = 0x2A22$ و $b_1 = 0x1034$

Binary => 0001000000110100 + 0010101000100010 = 0011101001010110 + 0 = 0011101001010110

0011101001010110 + 0011010000100101 = 0110111001111011 + 0 = 0110111001111011

0110111001111011 + 1111111100110111 = 0110110110110010 + 1 = 0110110110110011

Ones' complement => 1001001001001100

Hex => 1034 + 2A22 = 3A56 + 0 = 3A56

3A56 + 3425 = 6E7B + 0 = 6E7B

6E7B + FF37 = 6DB2 + 1 = 6DB3

Ones' complement => 924C

پاسخ سوال ۳:

(الف)

در بازه های [۶,۱] و [۲۶,۲۳] در وضعیت Slow Start قرار دارند زیرا اندازه پنجره ازدحام در هر زمان رفت و برگشت (RTT) دو برابر می شود.

(ب)

در بازه های [۱۶,۶] و [۲۲,۱۷] در وضعیت Congestion Avoidance قرار دارد زیرا اندازه پنجره ازدحام در این بازه ها در هر RTT یک واحد افزایش یافته است.



پایخ تمرین سری پنجم

(ج)

- ۳۲ - به دلیل اینکه وضعیت پروتکل از فاز Slowstart به فاز Congestion Avoidance تغییر کرده است.
- ۲۱ - مقدار ssthresh برابر است با نصف اندازه پنجره ازدحام در زمان وقوع ازدحام، در دوره ۱۶ام هنگامیکه packet loss تشخیص داده شده است، اندازه پنجره ازدحام برابر با ۴۲ بوده است. بنابراین در دوره ۱۸ام مقدار Slow Start Threshold برابر ۲۱ خواهد بود.
- ۱۳ - مقدار ssthresh برابر است با نصف اندازه پنجره ازدحام در زمان وقوع ازدحام، در دوره ۲۲ام که timeout رخ داده است، اندازه پنجره برابر ۲۶ بوده است.

(د)

در ششمین دور ارسال

(ه)

دوره‌های ۱۶ و ۲۲

در دوره ۱۶ پس از ازدحام، پروتکل در فاز Congestion Avoidance قرار گرفته است. بنابر این سه ACK تکراری دریافت شده است.

در دوره ۲۲، پس از تشخیص ازدحام پروتکل در فاز Slow Start قرار گرفته است. بنابر این timeout رخ داده است.

پاسخ سوال ۴:

$$R_e(Sat) = R_e(Tel)$$

$$U_{Sat} * R_{Sat} = U_{Tel} * R_{Tel}$$

$$U_{Sat} * 20Kbps = U_{Tel} * 10Kbps$$

$$\frac{1}{1 + 2a_{Sat}} * 2 = \frac{1}{1 + 2a_{Tel}}$$

$$1 + 2a_{Sat} = 2 + 4a_{Tel}$$

$$2 * a_{Sat} = 1 + 4 * a_{Tel}$$

$$2 * \frac{\frac{3 * 10^7}{3 * 10^8}}{\frac{2 * 10^4}{10^4}} = 1 + 4 * \frac{\frac{9 * 10^4}{3 * 10^8}}{\frac{3 * 10^8}{10^4}}$$

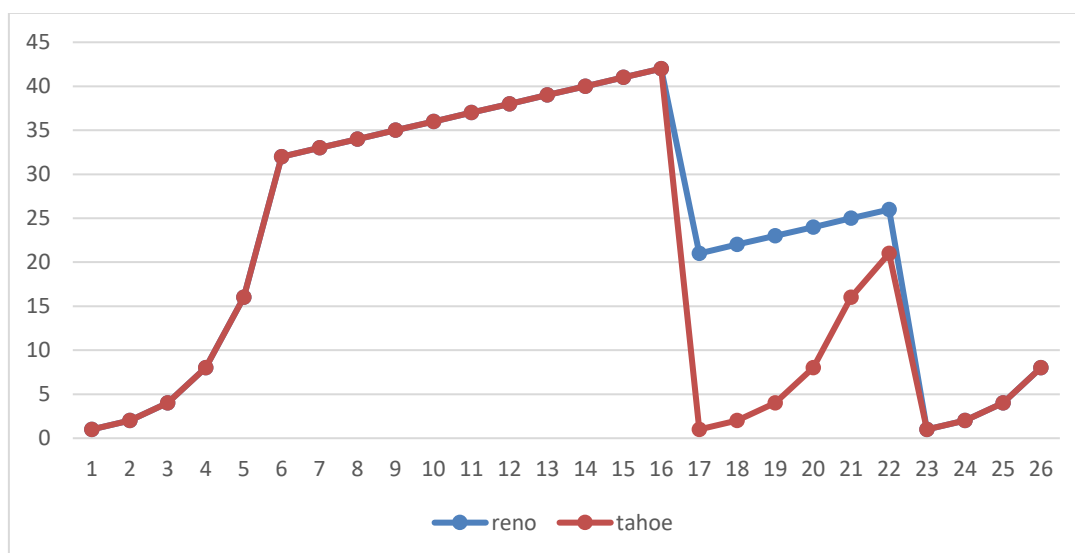
$$\frac{4000}{L} = 1 + \frac{12}{L}$$

و در نهایت خواهیم داشت:



$L = 3988$

پاسخ سوال ۵:



(الف)

$Ssthresh = 21 / Cwnd = 23 \llcorner TCP Reno$

$Ssthresh = 21 / Cwnd = 4 \llcorner TCP Tahoe$

(ب)

$Cwnd = 26 \llcorner TCP Reno$

$/ Cwnd = 21 \llcorner TCP Tahoe$

(ج)

« TCP Reno » تعداد کل بسته‌های ارسال شده از هفدهمین دوره تا بیست و دومین دوره

$$= 21 + 22 + 23 + 24 + 25 + 26 = 141$$

« TCP Tahoe » تعداد کل بسته‌های ارسال شده از هفدهمین دوره تا بیست و دومین دوره

$$= 1 + 2 + 4 + 8 + 16 + 21 = 52$$

پاسخ سوال ۶:

می‌دانیم:

$$DevRTT = \beta |SampleRTT - EstimatedRTT| + (1 - \beta) DevRTT$$

$$EstimatedRTT = \alpha SampleRTT + (1 - \alpha) EstimatedRTT$$



$$TimeoutInterval = EstimatedRTT + 4 \times DevRTT$$

بعد از بدست آوردن اولین $SampleRTT = 106ms$

$$DevRTT = 0.25 \times |106 - 100| + 0.75 \times 5 = 5.25ms$$

$$EstimatedRTT = 0.125 \times 106 + 0.875 \times 100 = 100.75ms$$

$$TimeoutInterval = 100.75 + 4 \times 5.25 = 121.75ms$$

بعد از بدست آوردن دومین $SampleRTT = 120ms$

$$DevRTT = 0.25 \times |120 - 100.75| + 0.75 \times 5.25 = 8.75ms$$

$$EstimatedRTT = 0.125 \times 120 + 0.875 \times 100.75 = 103.15ms$$

$$TimeoutInterval = 103.15 + 4 \times 8.75 = 138.15ms$$

بعد از بدست آوردن سومین $SampleRTT = 140ms$

$$DevRTT = 0.25 \times |140 - 103.15| + 0.75 \times 8.75 = 15.77ms$$

$$EstimatedRTT = 0.125 \times 140 + 0.875 \times 103.15 = 107.76ms$$

$$TimeoutInterval = 107.76 + 4 \times 15.77 = 170.84ms$$

بعد از بدست آوردن چهارمین $SampleRTT = 90ms$

$$DevRTT = 0.25 \times |90 - 107.76| + 0.75 \times 15.77 = 16.27ms$$

$$EstimatedRTT = 0.125 \times 90 + 0.875 \times 107.76 = 105.54ms$$

$$TimeoutInterval = 105.54 + 4 \times 16.27 = 170.62ms$$

بعد از بدست آوردن پنجمین $SampleRTT = 115ms$

$$DevRTT = 0.25 \times |115 - 105.54| + 0.75 \times 16.27 = 14.57ms$$

$$EstimatedRTT = 0.125 \times 115 + 0.875 \times 105.54 = 106.71ms$$

$$TimeoutInterval = 106.71 + 4 \times 14.57 = 165ms$$



پایخ تمرین سری پنجم

پاسخ سوال ۷:

در هر اتصال TCP بسته‌هایی با شماره ترتیب‌هایی ارسال می‌شوند که ممکن است به هر دلیل در شبکه باقی بمانند و هرگز به مقصد نرسند. این بسته‌ها امکان دارد زمانی به مقصد برسند که ارتباط قبلی تمام شده و ارتباط جدیدی شکل گرفته است. اگر شماره ترتیب بسته‌ها یکسان باشد، connection جدید اشتباها بسته‌های connection قبلی را دریافت خواهد کرد. با انتخاب شماره ترتیب اولیه مختلف برای هر ارتباط از پذیرش این بسته‌ها جلوگیری می‌شود.

پاسخ سوال ۸:

(الف)

رویداد (Event)	اقدام (Action)
Time out	۱ - ارسال مجدد قدیمی‌ترین سگمنت ۲ - روشن کردن مجدد تایمر با مقدار دو برابر قبلی
Triple duplicate Ack	ارسال مجدد قدیمی‌ترین سگمنت
New Ack	داده‌های تایید شده از بافر خارج می‌شوند و در صورتی که داده‌های ارسال شده قبلی در بافر وجود دارد که هنوز تایید نشده است، تایمر با مقدار تخمین زده شده روشن می‌شود.
دریافت داده از لایه بالاتر	۱ - تولید یک سگمنت جدید ۲ - اگر تایمر روشن نیست، با مقدار تخمین زده شده روشن می‌شود. ۳ - سگمنت برای ارسال به لایه IP داده می‌شود.

(ب)

تایمر در ابتدا برای دریافت تایید دریافت بسته با شماره ترتیب ۸۲ (ack = ۹۲) روشن شده است. (ack = ۹۲ معنای این است که تا بایت ۹۱ را گیرنده دریافت کرده است و منتظر بایت ۹۲ ام است.) تا زمان timeout فرستنده ack را دریافت نکرده است.

بنابراین بسته با شماره ترتیب ۸۲، retransmit خواهد شد.

بسته ۹۲ را نیز در بافر دارد. پس ack = ۱۰۰ را در پاسخ برمی‌گرداند.

بسته با شماره ترتیب ۱۰۰، با موفقیت ارسال نشده است. دو راه برای retransmit این داده وجود دارد. ۳ تاییدیه تکراری و یا timeout

پایخ تمرین سری پنجم

با توجه به اینکه سه ack تکراری دریافت نشده است این بسته با timeout, retransmit خواهد شد.

دقت شود که TCP تنها یک تایمر دارد. و بعد از اینکه بسته با شماره ترتیب ۸۲ و ۹۲ دریافت شود، تایمر برای داده با شماره ترتیب ۱۰۰ روشن خواهد شد.

قبل از اتمام تایمر نیز ۳ ack تکراری رخ خواهد داد.

بنابر این (۱) و (۲) هر دو برابر خواهند بود با Seq=100, 20 bytes data

دو ack آخر نیز تجمیعی ارسال هستند.

