



به نام خدا

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

درس شبکه‌های کامپیوتری ، نیمسال دوم سال تحصیلی 99-00
تمرین سه



دانشگاه صنعتی امیرکبیر
پلی تکنیک تهران

نام و نام خانوادگی:

شماره دانشجویی:

توضیحات:

- مهلت تحویل تمرین تا جمعه 7 خرداد در نظر گرفته شده است و تمدید پذیر نمی باشد.
- پاسخ به تمرین ها به صورت انفرادی باشد و اگر تقلب یافت شود نمره تمرین صفر خواهد شد.
- نظم و خوانایی تمرین از اهمیت بالایی برخوردار می باشد.
- خواهش می شود تمرین خود را در قالب یک فایل PDF با نام **"HW3_FirstnameLastName_StdudentNumber"** مانند ;
"HW3_ParsaAliEsfahani_ 9631052.pdf" در مهلت یاد شده در سایت بارگزاری فرمایید.
- پرسش های خود درباره این تمرین را می توانید از راه ایمیل های AUTCNTa@gmail.com بیان کنید.

1.

الف) multiplexing و demultiplexing را توضیح دهید.

جمع آوری chunk های داده در هاست مبدا از سوکت‌های مختلف، ایجاد هدر و سگمنت بندی چانک‌ها و انتقال سگمنت‌ها به لایه network را عمل multiplexing می‌گویند. در سمت گیرنده لایه transport فیلدهای موجود در هدر هر سگمنت را بررسی می‌کند و هر سگمنت را به سوکت مربوطه تحویل می‌دهد که به این عمل demultiplexing می‌گویند.

ب) تفاوت های بین connectionless demultiplexing و connection-oriented demultiplexing را شرح بگویید. (دست کم دو مورد)

در UDP(connectionless) demultiplexing داریم که سوکت آن را با دو آرگومان : IP گیرنده و پورت گیرنده مشخص میکنیم و میسازیم که برای فرستادن به سمت گیرنده از آن استفاده میشود. درحالی که در TCP(connection-oriented) سوکت را با چهار آرگومان : IP فرستنده , پورت فرستنده, IP گیرنده و پورت آن مشخص کرده.

در زمان آدرس دهی در connectionless demultiplexing از یک پورت یکسان در سمت گیرنده بهره می گیریم اما در connection oriented demultiplexing از پورت های مختلفی برای هر کانکشن استفاده میکنیم.

۲- برای هر یک از payload های زیر checksum را بدست آورید. (هرکدام را یک بسته کامل در نظر بگیرید)

الف) 0x1105|0x0209

0x1105 = 0001 0001 0000 0101

0x209 = 0000 0010 0000 1001

Addition: 0001000100000101

+ 0000001000001001

= 0001001100001110 ----> Checksum: 1110110011110001 (0xECF1)

ب) 0x1034|0x2A22|0x3425|0xFF37

0x1034 = 0001 0000 0011 0100

0x2A22 = 0010 1010 0010 0010

0x3425 = 0011 0100 0010 0101

0xFF37 = 1111 1111 0011 0111

Addition: 0001000000110100

+ 0010101000100010

= 0011101001010110

0011010000100101

+ 1111111100110111

= 10011001101011100

0011001101011100

+ 1

= 0011001101011101

0011101001010110

+ 0011001101011101

= 0110110110110011 -----> Checksum: 1001001001001100 (0x924C)

3- مقایسه GBN و SR : (اندازه پنجره را w در نظر بگیرید)

الف) در سمت فرستنده GBN و SR چند تایمر نیاز داریم؟

در GBN به یک زمان سنج نیاز داریم زیرا در صورتی که پنجره ارسال timeout شود همه‌ی پنجره دوباره فرستاده می‌شود و این در حالی است که در SR برای هر بسته از پنجره به یک زمان سنج نیاز است بنابراین در GBN به w زمان سنج نیاز است.

ب) اگر w یک عدد زوج باشد و بسته‌ی شماره $w/2$ از بین برود، در هر کدام از پروتکل‌ها کدامین بسته‌ها دوباره ارسال میشوند؟

SR: $w/2$

GBN: $w/2, w/2 + 1, \dots, w + w/2$

ج) اگر محدودیت حافظه داشته باشیم و پهنای باند اضافی داشته باشیم کدام پروتکل را انتخاب میکنید؟ چرا؟

استفاده از GBN به حافظه کمتری نیاز دارد.

د) اگر تاخیر دائلود برای کاربران مهم باشد، کدام پروتکل را انتخاب میکنید؟ چرا؟

استفاده از SR در زمان هایی که loss وجود دارد تاخیر را کاهش میدهد که میتواند به دو دلیل آن اشاره کرد:

نخست اینکه گیرنده بافر دارد و بسته های خارج از ترتیب را حفظ میکند و به محض دریافت بسته ی loss شده همه را به لایه ی بالاتر تحویل میدهد، در حالی که در GBN بسته خارج از ترتیب دریافت نمیشود .

دوم اینکه در پروتکل SR علاوه بر مکانیزم timeout برای تشخیص loss یک بسته یه مکانیزم تشخیص زودهنگام یا متوجه timeout وجود دارد که فرستنده زودتر از مقدار Fast Retransmission NACK شدن یک بسته میشود و ارسال مجدد زودتر انجام میشود loss

۴- با در نظر داشتن پروتکل های S&W پاسخ دهید:

الف) فرض کنید فرستنده میخواهد ۱۰ فریم بفرستد. اگر فریم های دوم و پنجم و پیام های ACK فریم سوم و ششم در نخستین بار فرستادن از بین بروند، ترتیب فرستادن فریم ها را نشان دهید. (اگر فریم ایی بیش از یک بار فرستاده میشود باید به همراه تعداد فرستادنش نشان داده شود).

1 2 2 3 3 4 5 5 6 6 7 8 9 10

ب) با یک نمونه نشان دهید که اگر در این پروتکل پیام های ACK دارای Sequence number نبودند چه مشکلی پیش می آمد.

- ارسال فریم صفر
 - Timeout و ارسال مجدد فریم صفر
 - دریافت Ack مربوط به اولین فریم صفر ارسال
 - ارسال فریم یک که در مسیر از بین میرود
 - دریافت Ack مربوط به دومین ارسال فریم صفر
 - ارسال فریم دوم
- در این حالت پیام Ack به اشتباه به عنوان Ack برای فریم یک در نظر گرفته میشود در حالی که این بسته اصلا به مقصد نرسیده است.

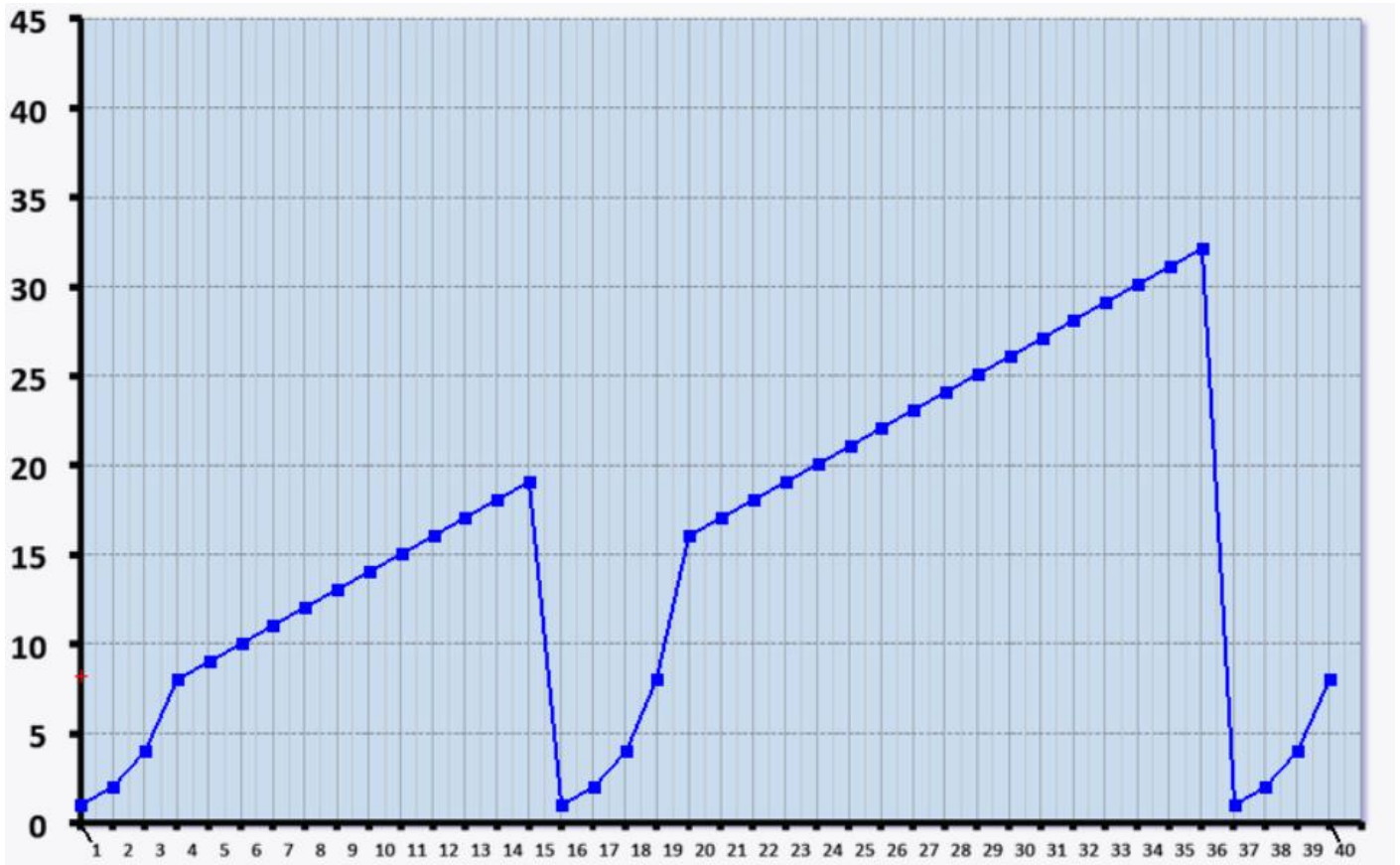
۵- در یک سیستم انتقال اطلاعات مبتنی بر بسته ها، بسته هایی با اندازه ۵۱۲ بایت بر روی یک لینک ارتباطی با نرخ ارسال ۵۱۲ کیلوبیت بر ثانیه و تاخیر انتشار ۲۰ میلی ثانیه ارسال میشوند. اگر برای کنترل خطا از مکانیسم پنجره لغزان استفاده شود، دست کم اندازه پنجره مورد نیاز برای دستیابی به بهره‌وری بهینه را بدست آورید.

$$Utilization = \frac{nL/R}{RTT + L/R}$$

$$T_{trans} = \frac{L}{R} = \frac{512 \times 8}{512 \times 1000} = 8ms$$

$$W_{min} = \left\lceil \frac{T_0}{T_p} \right\rceil = \left\lceil \frac{8 + 40}{8} \right\rceil = 6$$

6- با توجه به شکل به سوالات پاسخ دهید:



الف) در کدام زمان ها پروتکل TCP در مد slow start است؟

1,2,3,16,17,18,19,37,38,39,40

ب) در کدام بازه های زمانی پروتکل TCP در مد congestion avoidance می باشد؟

(4-15)(20-36)

ج) در کدام بازه های زمانی پروتکل TCP در مد fast recovery می باشد؟ -

د) در کدام زمان ها packet loss رخ داده است؟

15, 36

7) فرض کنید مقادیر فعلی پروتکل TCP برای $estimatedRTT$ و $DevRTT$ به ترتیب مقادیر 230 و 31 میلی ثانیه می باشند.

فرض کنید سه عدد اندازه گیری شده بعدی اندازه RTT به ترتیب 250 و 390 و 320 میلی ثانیه می باشند. (مقدار α را برابر 0.125 و β را 0.25 در نظر بگیرید)

الف) مقدار $estimatedRTT$ برای اولین RTT چند است؟

$$estimatedRTT = (1 - \alpha) * estimatedRTT + \alpha * sampleRTT =$$

$$(0.875) * 230 + 0.125 * 250 = 232.5$$

ب) مقدار $DevRTT$ برای اولین RTT چند است؟

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

$$(0.75) * 31 + 0.25 * 20 = 28.25$$

ج) مقدار TCP timeout را برای اولین RTT بدست آورید.

$$TimeoutInterval = EstimatedRTT + 4 * DevRTT =$$

$$232.5 + 4 * 28.25 = 345.5$$

د) مقدار $estimatedRTT$ برای دومین RTT چند است؟

$$estimatedRTT = (1 - \alpha) * estimatedRTT + \alpha * sampleRTT =$$

$$(0.875) * 232.5 + 0.125 * 390 = 252.1875$$

ه) مقدار $DevRTT$ برای دومین RTT چند است؟

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

$$(0.75) * 28.25 + 0.25 * 157.5 = 60.5625$$

و) مقدار TCP timeout را برای دومین RTT بدست آورید.

$$TimeoutInterval = EstimatedRTT + 4 * DevRTT =$$

$$252.1875 + 4 * 60.5625 = 494.4375$$

8) اگر در روش congestion control پروتکل TCP ، فرستنده با دریافت هر ACK جدید، به جای آنکه به اندازه پنجره ضریبی از MSS را اضافه کند (در حالت slow start یک MSS و در حالت congestion avoidance $MSS * (MSS/cwnd)$) به اندازه پنجره ضریب مثبتی از اندازه پنجره اضافه کند یعنی $cwnd = (1+\alpha) cwnd$ آنگاه رابطه ی بین L (نرخ از دست رفتن بسته ها) و W (حداکثر اندازه پنجره ازدحام) را بدست آورید.

مجموع سگمنت های ارسال شده در طول بازه ای که اندازه پنجره از $W/2$ تا W تغییر میکند برابر است با:

$$S = \frac{W}{2} + \left(\frac{W}{2}\right)(1+a) + \left(\frac{W}{2}\right)(1+a)^2 + \left(\frac{W}{2}\right)(1+a)^3 + \dots + \left(\frac{W}{2}\right)(1+a)^k =$$

$$\frac{W}{2} * \frac{(a+1)^{k+1} - 1}{a}$$

$$W = \left(\frac{W}{2}\right)(1+a)^k \rightarrow k = \log_{(1+a)} 2$$

$$S = \left(\frac{W}{2}\right) * \frac{(a+1)^k (a+1) - 1}{a} = W * \frac{2a+1}{2a}$$

$$L = \frac{1}{S} = \frac{2a}{(2a+1)W}$$

9) فرض کنید در یک ارتباط طول بسته ارسالی 84 بیت و سرعت ارسال برابر است با 84000 بیت بر ثانیه و اندازه پنجره را نیز برابر 13 در نظر بگیرید.

در روش Stop & Wait مقدار بهره وری کانال را بدست آورید. (تاخیر انتشار را 84 میلی ثانیه در نظر بگیرید)

$$L = 84 \text{ (bit)}$$

$$R = 84000 \text{ (bit/s)}$$

$$N = 13$$

$$t_{prop} = 84 \text{ (ms)}$$

$$U = \frac{1}{1 + 2 * (\frac{t_{prop}}{\frac{L}{R}})} = \frac{1}{169} = \frac{1}{169} = 0.005917 = 0.5917 \%$$

10) دلیل استفاده از شماره ترتیب (sequence number) و تایمر برای ارسال داده با قابلیت اطمینان چیست؟

گیرنده برای این که تشخیص بدهد که بسته دریافتی حاوی داده جدید است و یا فقط ارسال مجدد است نیاز به داشتن شماره ترتیب دارد.

برای مدیریت کردن بسته های از دست رفته در کانال ارتباطی از تایمر استفاده میشود به این صورت که اگر تا پایان زمان مقرر شده پاسخ بسته فرستاده شده از سوی گیرنده نرسد ، فرض کنیم که بسته ارسالی از دست رفته و مجدداً ارسال را انجام بدهیم.