



دانشکده مهندسی کامپیوتر

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

درس شبکه های کامپیوتری، نیم سال دوم سال تحصیلی ۱۴۰۱-۱۴۰۰  
پایخ تمرین سری بهجم



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

پاسخ سوال ۱:

(الف)

شماره ترتیب بایت های سگمنت اول ۹۰ تا ۱۰۹ است. بنابراین مقدار داده های سگمنت اول ۲۰ بایت است.

(ب)

با توجه به اینکه سگمنت اول از دست رفته است، پس گیرنده هنوز منتظر دریافت بایت ۹۰ است و شماره تایید (ACK) با حالت قبل فرقی نکرده و همان ۹۰ است.

پاسخ سوال ۲:

(الف)

$$EstimatedRTT^{(4)} = \alpha SampleRTT_4 + (1 - \alpha) EstimatedRTT^{(3)}$$

$$EstimatedRTT^{(3)} = \alpha SampleRTT_3 + (1 - \alpha) EstimatedRTT^{(2)}$$

$$EstimatedRTT^{(2)} = \alpha SampleRTT_2 + (1 - \alpha) EstimatedRTT^{(1)}$$

$$EstimatedRTT^{(1)} = \alpha SampleRTT_1 + (1 - \alpha) EstimatedRTT^{(0)}$$

$$EstimatedRTT^{(4)} = \alpha SampleRTT_4 + (1 - \alpha)(\alpha SampleRTT_3 + (1 - \alpha)(\alpha SampleRTT_2 + (1 - \alpha)(\alpha SampleRTT_1 + (1 - \alpha) EstimatedRTT^{(0)})))$$

$$EstimatedRTT^{(4)} = \alpha SampleRTT_4 + \alpha(1 - \alpha) SampleRTT_3 + \alpha(1 - \alpha)^2 SampleRTT_2 + \alpha(1 - \alpha)^3 SampleRTT_1 + (1 - \alpha)^4 EstimatedRTT^{(0)}$$

(ب)

$$EstimatedRTT^{(n)} = \alpha SampleRTT_n + (1 - \alpha) EstimatedRTT^{(n-1)}$$

$$EstimatedRTT^{(n-1)} = \alpha SampleRTT_{n-1} + (1 - \alpha) EstimatedRTT^{(n-2)}$$

⋮

$$EstimatedRTT^{(2)} = \alpha SampleRTT_2 + (1 - \alpha) EstimatedRTT^{(1)}$$

$$EstimatedRTT^{(1)} = \alpha SampleRTT_1 + (1 - \alpha) EstimatedRTT^{(0)}$$

$$EstimatedRTT^{(n)} = \alpha SampleRTT_n + (1 - \alpha)(\alpha SampleRTT_{n-1} + (1 - \alpha)(\alpha SampleRTT_{n-2} + \vdots (1 - \alpha)(\alpha SampleRTT_2 + (1 - \alpha)(\alpha SampleRTT_1 + (1 - \alpha) EstimatedRTT^{(0)})) \dots))$$

$$EstimatedRTT^{(n)} = \alpha SampleRTT_n + \alpha(1 - \alpha) SampleRTT_{n-1} + \alpha(1 - \alpha)^2 SampleRTT_{n-2} + \dots + \alpha(1 - \alpha)^{n-2} SampleRTT_2 + \alpha(1 - \alpha)^{n-1} SampleRTT_1 + (1 - \alpha)^n EstimatedRTT^{(0)}$$

$$EstimatedRTT^{(n)} = \alpha \left( \sum_{i=1}^n (1 - \alpha)^{n-i} SampleRTT_i \right) + (1 - \alpha)^n EstimatedRTT^{(0)}$$

با توجه به اینکه  $0 \leq \alpha \leq 1$  است، همانطور که در رابطه فوق مشاهده می شود، برای محاسبه تخمین زمان رفت و برگشت وزن نمونه های قدیمی تر به صورت نمایی کاهش پیدا می کنند و نمونه های جدید وزن بیشتری دارند. به همین دلیل به این روش Exponential Weighted Moving Average گفته می شود.



## پاسخ سوال ۳:

$$EstimatedRTT^{(n)} = \alpha SampleRTT_n + (1 - \alpha) EstimatedRTT^{(n-1)}$$

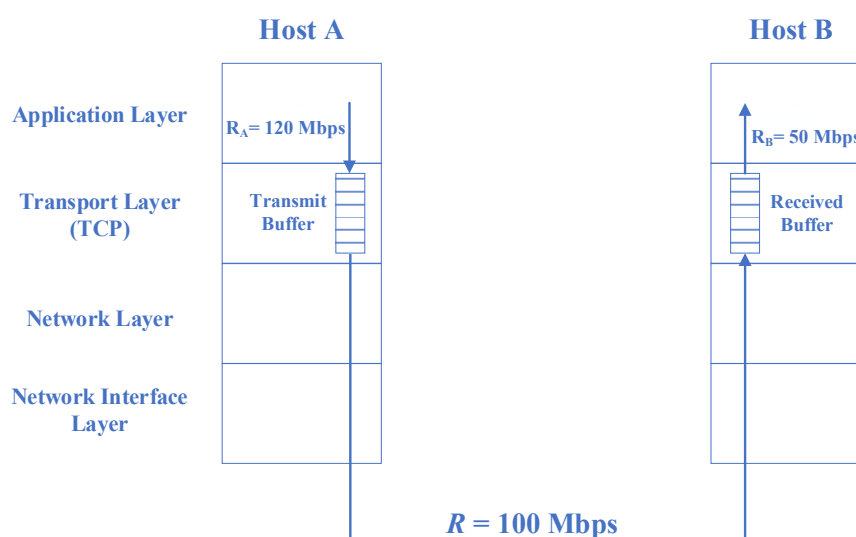
$$DevRTT^{(n)} = \beta |SampleRTT_n - EstimatedRTT^{(n)}| + (1 - \beta) DevRTT^{(n-1)}$$

$$TimeoutInterval^{(n)} = EstimatedRTT^{(n)} + 4 \times DevRTT^{(n)}$$

با توجه به اینکه  $\alpha = 0.125$  و  $\beta = 0.25$  است، مقدار  $EstimatedRTT$ ،  $DevRTT$  و  $TimeoutInterval$  بعد از هر  $SampleRTT$  بر اساس روابط فوق در جدول زیر نشان داده شده است (کلیه مقادیر بر حسب میلی ثانیه است).

$n$	$sampleRTT_n$	$EstimatedRTT^{(n)}$	$DevRTT^{(n)}$	$TimeoutInterval^{(n)}$
0	-	100	5	120
1	120	103	8	135
2	135	107	13	159
3	140	111	17	180
4	85	108	19	182
5	128	110	18	184

## پاسخ سوال ۴:



با توجه به اینکه ظرفیت ارسال لینک  $R = 100 Mbps$  است، بنابراین حداکثر نرخ ارسال لحظه‌ای میزبان A به میزبان B نیز  $100 Mbps$  است. بنابراین در ابتدا، بافر گیرنده با نرخ لحظه‌ای  $100 Mbps$  شروع به پر شدن می‌کند. میزبان B در هر بار ارسال داده یا ACK، اندازه خالی بافر دریافت خود را از طریق فیلد اطلاعاتی Received Buffer در سرآیند سگمنت‌های TCP به صورت Piggybacking به میزبان A اطلاع می‌دهد و میزبان A می‌بایست اندازه پنجره ارسال خود را کوچکتر از اندازه خالی فضای بافر دریافت، گیرنده قرار دهد. با توجه به بزرگتر بودن نرخ لحظه‌ای ارسال میزبان A از نرخ خواندن بافر دریافت گیرنده ( $R_B$ )، حجم داده ارسال شده در یک زمان ارسال و دریافت ACK (زمان رفت و برگشت) بستگی به اندازه خالی بافر دریافت گیرنده دارد که برابر است با حجم داده خوانده شده توسط برنامه کاربردی گیرنده در همان مدت زمان (زمان رفت و برگشت). در نتیجه نرخ ارسال مؤثر میزبان A برابر با نرخ خواندن بافر دریافت توسط برنامه کاربردی در میزبان B یعنی  $50 Mbps$  خواهد شد.

## پاسخ سوال ۵:

با توجه به اینکه سگمنت‌های TCP از طریق سرویس انتقال اطلاعات لایه شبکه یعنی پروتکل IP به گره میزبان مقصد تحویل داده می‌شوند و سرویس IP تضمینی برای حفظ ترتیب ارسال بسته‌ها ندارد، بنابراین احتمال عدم حفظ ترتیب ارسال سگمنت‌ها وجود دارد. وضعیتی را در نظر بگیرید که یک اتصال TCP به هر دلیلی بسته شده و بلافاصله یک اتصال TCP جدید باز شده است. اگر از اتصال TCP بسته شده، داده‌ای در اتصال



TCP جدید دریافت شود که شماره ترتیب ارسال آن مورد قبول گیرنده باشد (یعنی در پنجره دریافت قرار داشته باشد)، آنگاه داده‌های دریافتی معتبر تشخیص داده شده و توسط لایه بالاتر استفاده خواهند شد. TCP برای جلوگیری از وقوع این خطا، شماره ترتیب اولیه هر اتصال را به گونه‌ای انتخاب می‌کند که اگر هنوز سگمنتی از اتصال قبلی در شبکه باقی مانده باشد، شماره ترتیب داده‌های آن، در پنجره ارسال قرار نگیرد و خارج ترتیب تشخیص داده شده و دور ریخته شوند.

### پاسخ سوال ۶:

#### (الف)

دلیل ۱: شناسه‌های فرایندها وابسته به سیستم عامل است و استفاده از آن‌ها باعث می‌شود پروتکل‌های TCP و UDP به سیستم عامل وابسته باشند که این موضوع در مغایرت با مفهوم مدل لایه‌ای است.

دلیل ۲: یک فرایند یک شناسه فرایند دارد ولی ممکن است نیاز به ایجاد چندین کانال ارتباطی داشته باشد. در صورت استفاده از شناسه فرایند، امکان تفکیک کانال‌های ارتباطی به صورت منحصر بفرد وجود ندارد.

دلیل ۳: در مدل Client-Server، کلاینت باید شماره پورت برنامه کاربردی سرور را داشته باشد و سرور نیز باید بر روی پورت‌های شناخته شده گوش دهد. با توجه به اینکه شناسه فرایند توسط سیستم عامل تعیین می‌شود و برای برنامه‌های کاربردی کلاینت و سرور از قبل شناخته شده نیست، در این صورت امکان ارائه سرویس وجود نخواهد داشت.

#### (ب)

$$\begin{array}{rcl}
 & b_1 & 1001000000101010 \\
 + & b_2 & + \quad 0010101010111100 \\
 \hline
 & s_1 & 1011101011100110 \\
 + & b_3 & + \quad 0100010010001001 \\
 \hline
 & s_2 & 1111111101101111 \\
 + & b_4 & + \quad 0100011111100011 \\
 \hline
 & x_2 & 1 \quad 0100011101010010 \\
 + & c & + \quad 1 \\
 \hline
 & s_3 & 0100011101010011 \\
 \hline
 \text{Checksum} & & 1011100010101100
 \end{array}$$

#### (ج)

- برای سادگی محاسبات از مکمل یک استفاده شده است زیرا:
- مکمل یک با معکوس کردن بیت‌ها بدست می‌آید.
  - گیرنده با همان روشی که فرستنده Checksum را محاسبه کرده، مجدداً Checksum را برای کل سگمنت دریافتی محاسبه می‌کند و در صورتی نتیجه عدد صفر در مکمل یک شود (یعنی تمام بیت‌ها یک یا تمام بیت‌ها صفر باشند)، خطایی تشخیص داده نمی‌شود و نیاز به انجام عملیات اضافی دیگری برای تشخیص خطا نیست.
  - اگر از مدار سخت‌افزاری برای محاسبه Checksum استفاده شود، سخت‌افزار تولید Checksum و چک کردن آن به ترتیب در فرستنده و گیرنده یکسان است.
  - چون در مکمل یک، عدد صفر به دو صورت تماماً یک و تماماً صفر نمایش داده می‌شود. بنابراین اگر در یک سگمنت دریافتی فیلد Checksum تماماً صفر باشد، طبق پروتکل TCP و UDP به معنی عدم محاسبه Checksum توسط فرستنده است و نیازی به چک کردن توسط گیرنده نیست.



(د)

با استفاده از روش Checksum تمام خطاهای یک بیتی بطور کامل قابل تشخیص هستند. زیرا فقط یکی از اعداد تغییر می‌کند و حتماً نتیجه جمع نیز تغییر خواهد کرد. اما اگر دو یا بیش از دو بیت خطا رخ دهد، اگر تغییر اعداد به گونه‌ای باشد که نتیجه جمع تغییر نکند آن خطاها قابل تشخیص نیستند. به عنوان مثال اگر دو بیت همزمان با فاصله ۱۶ بیت تغییر کنند، خطا بوجود آمده قابل تشخیص نیست. دقت کنید احتمال رخ دادن خطاهای این چینی بسیار پایین است. به عنوان مثال، همان عدد قسمت (الف) را در نظر بگیرید که خطاهای زیر بر روی آن اتفاق افتاده است.

1001000000101000001010101011111001000100100010010100011111100011

پاسخ سوال ۷:

(الف)

در بازه‌های (۱ تا ۶) و (۲۳ تا ۲۶) پروتکل در فاز Slow Start است، زیرا اندازه پنجره ازدحام در هر زمان رفت و برگشت ( $RTT$ ) دو برابر شده است.

(ب)

در بازه‌های (۶ تا ۱۶) و (۱۷ تا ۲۲) پروتکل در فاز Congestion Avoidance قرار دارد، زیرا اندازه پنجره ازدحام در هر  $RTT$  یک واحد افزایش یافته است.

(ج)

ردیف	زمان (دور)	مقدار $ssthresh$	توضیح
۱	آغاز به کار پروتکل	۳۲	بدلیل اینکه وضعیت پروتکل از فاز Slow Start به فاز Congestion Avoidance تغییر کرده است.
۲	در ۱۸امین دور ارسال	۲۱	مقدار $ssthresh$ برابر است با نصف اندازه پنجره ازدحام در زمان وقوع ازدحام، در دوره ۱۶ که ازدحام تشخیص داده شده است، اندازه پنجره ازدحام ۴۲ بوده است.
۳	در ۲۴امین دور ارسال	۱۳	مقدار $ssthresh$ برابر است با نصف اندازه پنجره ازدحام در زمان وقوع ازدحام، در دوره ۲۲ که ازدحام تشخیص داده شده است، اندازه پنجره ازدحام ۲۶ بوده است.

(د)

در ۶امین دور ارسال

(ه)

ردیف	زمان تشخیص Packet Loss	روش تشخیص	توضیح
۱	دوره ۱۶	دریافت سه ACK تکراری	قرارگرفتن در فاز Congestion Avoidance بعد از تشخیص ازدحام
۲	دوره ۲۲	Timeout	قرارگرفتن در فاز Slow Start بعد از تشخیص ازدحام

پاسخ سوال ۸:

از آنجایی که ازدحامی رخ نمی‌دهد بنابراین بسته‌ای از دست نخواهد رفت، بدین ترتیب اندازه‌ی پنجره ازدحام ( $CW$ ) در فاز Slow Start در هر بار ارسال دو برابر می‌شود. در نتیجه بعد از ۴ زمان رفت و برگشت ( $RTT$ ) اندازه پنجره ازدحام به ۳۲ می‌رسد.

$$2\text{ KB} \rightarrow 4\text{ KB} \rightarrow 8\text{ KB} \rightarrow 16\text{ KB} \rightarrow 32\text{ KB}$$

اما می‌دانیم پنجره ارسال ( $W_s$ ) نمی‌تواند از اندازه بافر دریافت ( $W_a$ ) بزرگتر باشد،

$$W_s = \min(W_a, CW)$$

بنابراین بعد از ۴ زمان رفت برگشت پنجره ارسال برابر با اندازه بافر دریافت خواهد شد. با توجه به اینکه  $RTT = 10\text{ msec}$  است، پس بعد از ۴۰ میلی‌ثانیه پنجره ارسال برابر با پنجره دریافت می‌شود.



### پاسخ سوال ۹:

$$R = 20 \text{ Mbps}$$

$$MSS = 1500 \text{ Bytes}$$

$$RTT = 120 \text{ msec}$$

#### (الف)

اگر  $W$  حداکثر ممکن اندازه پنجره ازدحام بر حسب تعداد سگمنت باشد، آنگاه در نرخ ارسال حداکثر  $R = W \times MSS / RTT = 20 \text{ Mbps}$  بسته‌ها از دست خواهند رفت. در نتیجه:

$$W = \frac{R \times RTT}{MSS} = \frac{20 \times 10^6 \times 120 \times 10^{-3}}{1500 \times 8} = 200 \text{ (Segments)}$$

#### (ب)

با توجه به اینکه این اتصال همیشه در فاز دوری از ازدحام است، بنابراین اندازه پنجره ازدحام بین مقدار  $W/2$  تا  $W$  دائماً در حال تغییر است که متوسط آن  $0.75W$  است. یعنی ۱۵۰ سگمنت در هر  $RTT$  ارسال می‌شود بنابراین گذردهی برابر است با:

$$\text{Throughput} = \frac{0.75W \times MSS}{RTT} = \frac{0.75 \times 200 \times 12000}{0.15} = 12 \text{ Mbps}$$

#### (ج)

زمانی که یک بسته از دست می‌رود، اندازه پنجره ازدحام از مقدار  $W$  به مقدار  $W/2$ ، یعنی از ۲۰۰ به ۱۰۰، کاهش پیدا می‌کند. چون در فاز دوری از ازدحام قرار داریم بعد از هر  $RTT$ ، اندازه پنجره ازدحام یک سگمنت بزرگ‌تر می‌شود، بنابراین بعد از  $(W/2) \times RTT = 100 \times 0.15 = 15 \text{ sec}$  اندازه پنجره دوباره به مقدار حداکثر خود یعنی ۲۰۰ می‌رسد.