

## تمرین چهارم شبکه های کامپیوتری

### سوال اول

الف) در سگ منت اول می توان ۲۰ بیت ارسال کرد.

ب) شماره ACK برابر ۹۰ خواهد بود.

### سوال دوم

الف) معتقدم صورت سوال به اشتباه نوشته شده است و SampleRTT1 جدید ترین و سپس SampleRTT2 و ...

$$EstimatedRTT = \alpha \cdot SampleRTT_1 + (1 - \alpha) \cdot \alpha \cdot SampleRTT_1 + (1 - \alpha)^2 \cdot \alpha \cdot SampleRTT_1 + (1 - \alpha)^3 \cdot \alpha \cdot SampleRTT_1$$

ب) به طوری کلی می توان فرمول را به شکل مقابل نوشت:

$$EstimatedRTT = \alpha \cdot \sum_{i=1}^{n-1} (1 - \alpha)^i \cdot SampleRTT_i + (1 - \alpha)^{n-1} SampleRTT_n$$

در حالتی که n به بی نهایت میل کند داریم:

$$EstimatedRTT = \frac{\alpha}{1 - \alpha} \cdot \sum_{i=1}^{\infty} (1 - \alpha)^i \cdot SampleRTT_i$$

به عبارتی وزن داده شده به هر SampleRTT نمایی است که مطابق تعریف EMA یا EWMA است.

### سوال سوم

ابتدا فرمول استفاده شده در این سوال را می نویسیم:

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

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

$$TimeOutInterval = EstimatedRTT + (4 \cdot DevRTT)$$

حال سوال را با مقادیر گفته شده حل می کنیم:

• 120ms

$$devRTT = (0.75 * dev) + (0.25 * abs(estimated - 120)) = 8.75$$

$$EstimatedRTT = (0.875 * estimated) + (0.125 * 120) = 102.5$$

$$TimeOutInterval = estimated + (4 * dev) = 137.5$$

• 135ms

$$devRTT = (0.75 * dev) + (0.25 * abs(estimated - 135)) = 15.43$$

$$EstimatedRTT = (0.875 * estimated) + (0.125 * 135) = 106.56$$

$$TimeOutInterval = estimated + (4 * dev) = 168.31$$

• 140ms

$$devRTT = (0.75 * dev) + (0.25 * abs(estimated - 140)) = 19.93$$

$$EstimatedRTT = (0.875 * estimated) + (0.125 * 140) = 110.74$$

$$TimeOutInterval = estimated + (4 * dev) = 190.49$$

• 85ms

$$devRTT = (0.75 * dev) + (0.25 * abs(estimated - 85)) = 20.98$$

$$EstimatedRTT = (0.875 * estimated) + (0.125 * 85) = 107.52$$

$$TimeOutInterval = estimated + (4 * dev) = 191.45$$

## تمرین چهارم شبکه های کامپیوتری

128ms •

$$devRTT = (0.75 * dev) + (0.25 * abs(estimated - 128)) = 20.85$$

$$EstimatedRTT = (0.875 * estimated) + (0.125 * 128) = 110.08$$

$$TimeoutInterval = estimated + (4 * dev) = 193.51$$

### سوال چهارم)

در گام اول به دلیل اینکه لینک ارتباطی بین دو A و B در نهایت ۱۰۰ مگابیت بر ثانیه گنجایش دارد، پس A هم با ۱۰۰ مگابیت بر ثانیه داده منتقل می کند. در گام دوم به دلیل اینکه همچنان نرخ ارسال A دو برابر نرخ دریافت B است شاهد ازدحام و در نتیجه مکانیسمی برای کنترل آن هستیم. به گونه ای که وقتی بافر B پر شود به A سیگنال می دهد تا ارسال داده را به طور موقت متوقف کند. این ارسال و توقف آنقدری ادامه خواهد داشت تا داده منتقل شود. متغیر کنترلی برای متوقف یا ادامه دادن ارسال داده از A به B در اصل RcvWindow بوده که با گرفتن مقدار ۰ دیگر A دیتا نمیفرستد.

### سوال پنجم)

پروتکل TCP با عوض کردن ISN به صورت رندوم از حملات امنیتی احتمالی جلوگیری میکند؛ زیرا این Sequence Number یک پرچم در هدر بسته ارسالی است که مقدار SYN را کنترل می کند.

### سوال ششم)

الف) مورد اول اینکه شماره فرایند با شماره پورت کاملاً مفاهیم مجزایی هستند و اصلاً در ذات و تعریف به هم ربطی ندارند. پس به این دلیل نمی توان به جای هم استفاده کرد. زیرا شماره فرایند می تواند تکراری باشد یا ترتیب متفاوتی داشته باشد. همچنین محدودیت بازه عددی فرایند با پورت متفاوت است (پورت محدود تر است). پس نمیتوان جای هم این دو را استفاده کرد.  
ب) فرایند جمع این چهار عدد:

1001000000101010

0010101010111100 -> 1011101011100110

0100010010001001 -> 1111111101101111

0100011111100011 -> 0100011101010011

سپس مکمل یک جواب: 1011100010101100

ج) دلیل این کار آن است که وقتی مکمل یک حساب شده با تمام checksum ها جمع شود همه بیت ها باید برابر ۱ شوند که مشخصه درست انجام شدن فرایند است.

د) بله امکان دارد. خطا میتواند به صورتی رخ دهد که checksum همچنان ثابت بماند یا بعد تغییر مشابه checksum اولیه شود. اما یک checksum ۱۶ بیتی میتواند خطاهای ۱ تا ۱۶ بیتی را تشخیص دهد اما بیشتر را نه. به عبارت کلی هرچه تعداد بیت های checksum بیشتر شود قادر است بازه بیشتری از خطاها را تشخیص دهد.

### سوال هفتم)

الف) بازه های ۱ تا ۶ و ۲۳ تا ۲۶

ب) بازه های ۶ تا ۱۶ و ۱۷ تا ۲۲

ج)

آغاز به کار پروتکل: ۳۳، زیرا در این نقطه Slow Start به پایان رسیده و congestion avoidance شروع می شود.

در ۱۸امین دور ارسال: حدوداً ۲۱، زیرا مقدار threshold در هر packet loss نصف می شود. چون مقدار قبلی آن

۴۲ بوده، پس مقدار جدید باید ۲۱ شود.

## تمرین چهارم شبکه های کامپیوتری

- در ۲۴امین دور ارسال: ۱۴. مانند بخش قبلی چون مقدار congestion در دور ۲۲ام برابر ۲۹ است و چون packet loss داشتیم مقدار threshold نصف شده و برابر ۱۴.۵ می شود؛ مقدار روند آن ۱۴ (یا ۱۵) است.
- د) ۶ام. طبق فرمول بسته های  $2^{n-1}$  تا  $2^n$  در ارسال nام فرستاده می شوند. پس بسته ششم که بین بسته ۳۲ تا ۶۳ است در ارسال ششم ارسال می شوند.
- ه) در زمان های ۱۶ و ۲۲ شاهد packet loss هستیم. در ارسال ۱۶ام به نوع Triple Duplicate ACK و در ارسال ۲۲ام به نوع Timeout برخوردیم زیرا در بار ۱۶ام پنجره نصف شده اما در بار ۲۲ام پنجره مقدار ۱ گرفته است.

### سوال هشتم)

ز

### سوال نهم)

اگر W برابر بیشترین پنجره اندازه گیری شده هر سگ منت باشد، داریم:

$$W \cdot \frac{MSS}{RTT} = 20Mbps$$

الف) که از رابطه بالا مقدار W برابر ۲۰۰ بدست می آید.

ب) به دلیل آنکه پنجره ازدحام بین W و W/2 متغیر است پس میانگین آن برابر 0.75.W که معادل ۱۵۰ است. مقدار

Average Throughput برابر ۱۵ مگابیت بر ثانیه است.

ج) هنگام رخ دادن ازدحام پنجره از W به W/2 تغییر میکند و این یعنی ۷۵ واحد کاهش در اندازه پنجره. با ضرب این مقدار در تاخیر انتشار، میزان تاخیر باز ارسال ۹ ثانیه می شود.

### سوال دهم)

از مزایای مهم پروتکل QUIC اجازه دادن به ارتباطات فقط رمزنگاری شده است؛ به گونه ای که ارتباطاتی که رمزنگاری نشده باشند پنهان هستند که مزیت مهمی برای امنیت شبکه است.

از طرفی برخلاف TCP فقط به یک Handshake نیاز دارد تا ارتباط را برقرار کند؛ در صورتی که TCP به دو Handshake نیاز دارد.

در ارتباطات HTTP+TLS+TCP هر پروتکل نیاز به Handshake مختص خود دارد اما در QUIC صرفاً یک Handshake ارتباط را برقرار می کند.

همچنین الگوریتم های کنترل ازدحام بهینه تری نسبت به TCP در QUIC استفاده شده است.