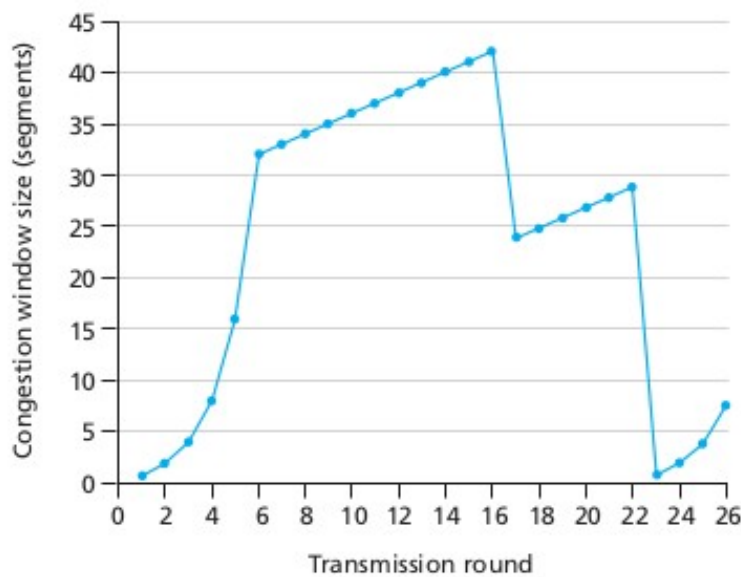


بسمه تعالی  
دانشکده ی مهندسی برق و کامپیوتر  
دانشگاه صنعتی اصفهان

شبکه‌های کامپیوتری ۲ – نیمسال اول ۱۳۹۹-۱۴۰۰  
تکلیف شماره پنج – تحویل یکشنبه ۱۳۹۹/۸/۲۵

مریم سعیدمهر – ش.د.: ۹۶۲۹۳۷۳

- 3.P40. Consider Figure 3.58. Assuming TCP Reno is the protocol experiencing the behavior shown above, answer the following questions. In all cases, you should provide a short discussion justifying your answer.
- Identify the intervals of time when TCP slow start is operating.
  - Identify the intervals of time when TCP congestion avoidance is operating.
  - After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
  - After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?



**3.58** ♦ TCP window size as a function of time

- e. What is the initial value of `ssthresh` at the first transmission round?
- f. What is the value of `ssthresh` at the 18th transmission round?
- g. What is the value of `ssthresh` at the 24th transmission round?
- h. During what transmission round is the 70th segment sent?
- i. Assuming a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, what will be the values of the congestion window size and of `ssthresh`?
- j. Suppose TCP Tahoe is used (instead of TCP Reno), and assume that triple duplicate ACKs are received at the 16th round. What are the `ssthresh` and the congestion window size at the 19th round?
- k. Again suppose TCP Tahoe is used, and there is a timeout event at 22nd round. How many packets have been sent out from 17th round till 22nd round, inclusive?

( a ) در بازه های [1,6] و [23,26] در استیت slowstart هستیم (چون اندازه پنجره به صورت نمایی افزایش یافته)

( b ) در بازه های [6,16] و [17,22] در استیت congested avoidance هستیم (چون اندازه پنجره خطی افزایش یافته)

( c ) در ۱۶ امین راند ارسال، ما با triple duplicate ACK روبه رو شدیم. در TCP Reno در صورت مواجه شدن با timeout اندازه پنجره را روی 1 ست میکند ولی در این راند، اندازه پنجره هم مانند `ssthresh` نصف شده و این یعنی triple duplicate ACK رؤیت شده.

( d ) در ۲۲ امین راند ارسال، ما با timeout روبه رو شده ایم و به همین دلیل اندازه پنجره ارسال روی 1 ست شده است (درقسمت c توضیح دادم!)

( e ) از آنجایی که در استیت slowstart تا جایی اندازه پنجره به صورت نمایی افزایش میابد تا به `ssthresh` برسیم پس در ابتدا  $ssthresh = 32$  بوده است.

(اگر در ابتدای کار باشیم یعنی اولین باریست که TCP Reno را اجرا میکنیم،  $ssthresh = \infty$  ست شده و این افزایش تا مشاهده سه duplicate ACK ادامه میابد که در آنجا مقدار `ssthresh` مقداری به جز  $\infty$  خواهد گرفت و معادل نصف `cwnd` آن لحظه خواهد شد و در ادامه ی کار طبق روالی که عرض کردم پیش میرود یعنی تا رسیدن به مقدار `ssthresh` نمایی رشد میکند و بعد از آن خطی افزایش میابد)

f) در راند ۱۶ ام مقدار ماکزیمم  $cwnd = 42$  بوده (طبق چیزی که از تصویر پیداست!) و در این راند باید  $ssthresh$  نصف این مقدار  $cwnd$  شود چرا که triple duplicate ACK رؤیت شده پس در بازه  $[17,22]$  مقدار  $ssthresh = 21$  .  
 g) در راند ۲۲ ام ، با timeout مواجه شده ایم و ماکزیمم  $cwnd = 29$  است و در بازه  $[23,26]$  مقدار  $ssthresh = 14$  یعنی کف (گرد شده به سمت پایین) نصف  $cwnd$  خواهد شد.

h) اندازه طول پنجره مشخص میکند هر بار چند بسته ارسال شده است. مثلاً در راند ۱ ، بسته اول ارسال شده ، در راند ۲ ، بسته دوم و سوم نیز ارسال شده اند. پس اگر به صورت تجمعی این روند را ادامه دهیم واضح است که در راند ۷ ام بسته های ۶۴ تا ۹۶ ارسال خواهند شد که بسته ۷۰ هم در همین راند ارسال میشود.

i) در راند ۲۶ ام ، فعلاً در استیت slowstart هستیم و اگر triple duplicate ACK دریافت کنیم ، به استیت fast recovery خواهیم رفت و مقدار  $ssthresh = 4$  خواهد شد که معادل نصف  $cwnd$  در راند ۲۶ ام است. همچنین در این استیت مقدار جدید پنجره معادل مقدار جدید  $ssthresh + 3$  خواهد شد یعنی  $cwnd = 7$  میشود.

j) مقدار  $ssthresh$  همان 21 که در قسمت f محاسبه کردیم میشود ( چون به هر حال در هر دو TCP Reno و TCP Tahoe باید مقدار پنجره در لحظه وقوع triple duplicate ACK نصف شود ) اما مقدار پنجره در راند ۱۷ ام روی 1 ست میشود و به صورت نمایی شروع به رشد میکند تا به  $ssthresh$  برسد پس با این حساب در راند ۱۹ ام  $cwnd = 4$  میشود و مقدار  $ssthresh = 21$  است.

k) چون در راند ۱۶ هم timeout داشتیم پس در راند ۱۷ ام مقدار پنجره روی ۱ ست شده و  $ssthresh = 21$  است همچنین در استیت slowstart هستیم و اندازه پنجره به صورت نمایی رشد میکند . پس :

- راند ۱۷ : ۱ بسته
- راند ۱۸ : ۲ بسته
- راند ۱۹ : ۴ بسته
- راند ۲۰ : ۸ بسته
- راند ۲۱ : ۱۶ بسته

- راند ۲۲ : ۲۱ بسته ( به  $ssthresh$  رسیده ایم )

پس به صورت تجمعی ، ۵۲ بسته در بازه  $[17,22]$  ارسال شده است.

- 3.P46. Consider that only a single TCP (Reno) connection uses one 10Mbps link which does not buffer any data. Suppose that this link is the only congested link between the sending and receiving hosts. Assume that the TCP sender has a huge file to send to the receiver, and the receiver's receive buffer is much larger than the congestion window. We also make the following assumptions: each TCP segment size is 1,500 bytes; the two-way propagation delay of this connection is 150 msec; and this TCP connection is always in congestion avoidance phase, that is, ignore slow start.
- What is the maximum window size (in segments) that this TCP connection can achieve?
  - What is the average window size (in segments) and average throughput (in bps) of this TCP connection?
  - How long would it take for this TCP connection to reach its maximum window again after recovering from a packet loss?

( a

$$\frac{W \times MSS}{RTT} = 10 \text{ Mbps}, \text{ MSS} = 1500 \text{ bytes}, \text{ RTT} = 150 \text{ msec}$$

$$\rightarrow W = \frac{10 \times 10^6 \times RTT}{MSS}$$

$$\rightarrow W = \frac{10^7 \times 150 \times 10^{-3}}{1500 \times 8}$$

$$\rightarrow W = 125 \text{ segments}$$

( b طول پنجره از  $W$  تا  $\frac{W}{2}$  متغیر است پس میانگین اندازه پنجره برابر با  $\frac{3W}{4}$  است. همچنین میانگین throughput هم برابر است با  $\frac{W_{avg} \times MSS}{RTT}$  پس در مجموع :

$$W_{avg} = \frac{3W}{4} = \left\lfloor \frac{3 \times 125}{4} \right\rfloor = 94 \text{ segments}$$

$$Throughput_{avg} = \frac{W_{avg} \times MSS}{RTT} = \frac{94 \times 1500 \times 8}{150 \times 10^{-3}} = 7.52 \text{ Mbps}$$

( c ) از آنجایی که طبق صورت سوال ، ما هیچگاه به استیت slowstart نمیرویم لذا در صورت وقوع packet loss اندازه پنجره نصف cwnd میشود (= ۶۲) و حال باید به صورت خطی مجدداً به اندازه اصلی (= ۱۲۵) برگردد و این معدل ۶۳ transmission round است( به صورت خطی یعنی هر راند ، پنجره به اندازه ی یک MSS افزایش می یابد). همچنین میدانیم RTT = 150 msec است پس

$$\text{زمان مطلوب} = 63 \times 150 \text{ msec} = 63 \times 0.15 = 9.45 \text{ sec}$$