



تمرین شماره هشت + حل و بررسی

درس شبکه های کامپیوتری - فصل سوم

مدرس: دکتر هاشمی

۱- پروتکل GBN را در نظر بگیرید که اندازه ی پنجره ی فرستنده در آن ۴ و محدوده ی شماره ترتیب^۱ ۱۰۲۴ تنظیم شده است. فرض کنید که در لحظه ی t بسته ای که گیرنده منتظر دریافت آن است شماره ترتیب k داشته باشد، و فرض دیگر این است که کانال ترتیب بسته ها را به هم نمی ریزد. حال به پرسش های زیر پاسخ دهید:

الف. در لحظه ی t مجموعه های احتمالی شماره ترتیب در پنجره ی فرستنده چیست؟

اگر تمامی ACK ها رسیده باشد: $[k, k+1, k+2, K+3]$

اگر هیچ کدام از ACK ها به فرستنده نرسیده باشد: $[k-4, k-3, k-2, K-1]$

در نتیجه این حالت ها ممکن است: $[k, k+1, k+2, K+3]$ ، $[k-1, k, k+1, K+2]$ ،

$[k-2, k-1, k, K+1]$ ، $[k-3, k-2, K-1, k]$ ، $[k-4, k-3, k-2, K-1]$

ب. همه ی مقادیر احتمالی فیلد های ACK در هر بسته ای که در لحظه ی t در حال انتشار به سمت فرستنده می باشد، چیست؟

$k-4, k-3, k-2, K-1$

۲- پروتکل های Go Back N و Selective Repeat در نظر بگیرید. محدوده ی شماره ترتیب را k در نظر بگیرید. بزرگ ترین اندازه ی پنجره ی فرستنده در این پروتکل ها، به طوری که از مشکلاتی مانند آن چه که در تصویر پیوست یک رخ داده جلوگیری شود، چقدر است؟

فرض کنید کوچکترین شماره ای که گیرنده منتظر آن است m باشد. در این حالت پنجره ی گیرنده $[m, m+w-1]$ است. در این حالت ACK های شماره ی $m-w$ تا $m-1$ ممکن است هنوز به فرستنده نرسیده باشند. (روی کانال باشند). در نتیجه در بدترین حالت پنجره ی فرستنده $[m-w, m-1]$ خواهد بود.

در این حالت کوچکترین شماره ی پنجره ی فرستنده $m-w$ و بزرگترین شماره ی پنجره ی گیرنده $m+w-1$ است. برای این که لبه ی بالایی پنجره گیرنده و لبه ی پایینی پنجره فرستنده روی هم نیفتند، باید محدوده ی شماره ترتیب حداقل دو برابر اندازه ی پنجره باشد. ($k \geq 2w$)

۳- درستی یا نادرستی گزاره های زیر را با یک توضیح کوتاه مشخص کنید.

الف. در پروتکل SR ممکن است ACK هایی به فرستنده برسد که مربوط به بسته ای خارج از پنجره ی فعلی اش باشد.

بله، فرض کنید که در لحظه ای time out رخ دهد و فرستنده بسته ها را دوباره ارسال کند. حال اگر ACK های مربوط به ارسال اولیه برسند، پنجره رو به جلو رفته و ACK های مربوط به ارسال مجدد خارج از پنجره ی فعلی می رسند.

ب. در پروتکل GBN ممکن است ACK هایی به فرستنده برسد که مربوط به بسته ای خارج از پنجره ی فعلی اش باشد. بله، همانند الف

پ. پروتکل alternating-bit protocol مانند یک پروتکل SR با اندازه پنجره ی فرستنده و گیرنده برابر با یک است. بله، در این حالت Cumulative ACK ها همانند یک ACK عادی هستند.

ت. پروتکل alternating-bit protocol مانند یک پروتکل GBN با اندازه پنجره ی فرستنده و گیرنده برابر با یک است. بله همانند پ

۴- انتقال یک فایل بزرگ L بایتی از میزبان A به میزبان B را در نظر بگیرید. MSS را 536 بایت در نظر بگیرید.

الف. بیشترین اندازه ی بسته (L_{max}) را بیابید، به طوری که شماره ترتیب های TCP دور نخورد.

باید شماره ی سگمنت ها با یک عدد 32 بیتی قابل نمایش باشد. آخرین بسته که حاوی 536 بایت آخر فایل است نهایتاً می تواند شماره ای برابر 2^{32} داشته باشد. در نتیجه حداکثر سائز فایل برابر با $536 + 2^{32}$ بیت است. ($= 4,294,967,832$ بایت معادل 4 گیگا بایت)

ب. از روی اندازه ی بسته در بخش الف زمان ارسال کل بسته را به دست آورید. این فرض ها را هم داشته باشید: هر سگمنت 42 بایت افزونه ($header$) دارد؛ نرخ ارسال 100 مگابیت در ثانیه است؛ سگمنت ها پشت سر هم و بدون وقفه ارسال می شوند.

$$\#segments = 4,294,967,832 / 536 = 8,013,000$$

$$Total\ bytes\ to\ be\ sent = (536 + 42) * 8,013,000 = 4,631,514,000\ bytes$$

$$delay = 4,631,514,000 * 8 / 100M = 370.5\ sec$$

۵- دو کامپیوتر A و B در حال برقراری ارتباط TCP با یکدیگر هستند و B در یک لحظه ی مشخص 126 بایت از A دریافت کرده است. بعد از آن A دو بسته ی 70 و 30 بایتی ارسال می کند. اولین بسته شماره ی 127 دارد و پورت مبدا در آن 302 و پورت مقصد نیز 80 ذکر شده است.

الف. شماره ترتیب، پورت مبدا و مقصد برای دومین بسته چیست؟

به ترتیب: 197 ($127 + 70$)، 302 ، 80

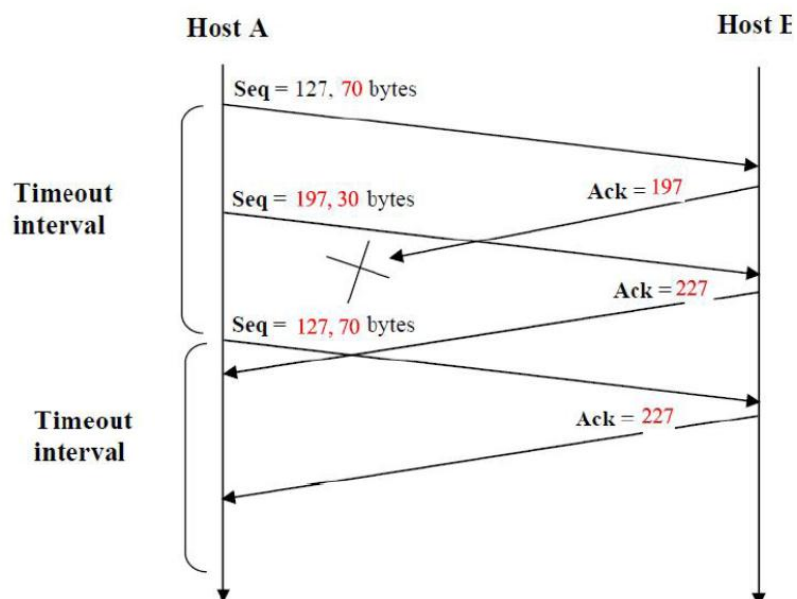
ب. اگر بسته ی اول زودتر برسد، شماره ترتیب و پورت مبدا و مقصد ACK ای را که برای بسته ی اول ارسال می شود بنویسید.

۱۹۷ (منتظر بایت ۱۹۷ ام است.)، ۸۰، ۳۰۲

پ. قسمت ب را با فرض این که بسته ی دوم زودتر برسد پاسخ دهید.

۱۲۷، ۸۰، ۳۰۲

ت. سناریوی زیر را که در آن ACK بسته ی اول گم می شود، کامل کنید. ACK دوم نیز چنان چه دیده می شود بعد از time interval می رسد.



۶- جدول زیر مقدار ۴ تا از SampleRTT های ثبت شده را نشان می دهد. (بر حسب میلی ثانیه)

۸۵	۱۳۰	۱۱۰	۱۰۶
----	-----	-----	-----

الف. EstimatedRTT را بعد از دریافت هر کدام از این مقادیر محاسبه کنید. (مقدار آلفا را ۰,۱۲۵ و مقدار EstimatedRTT قبل از دریافت این بسته ها را ۱۰۰ میلی ثانیه در نظر بگیرید).

$$\text{EstimatedRTT} = (1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT}$$

$$\text{EstimatedRTT} = (1-0.125) * 100 + 0.125*85 = \underline{98.13}$$

$$\text{EstimatedRTT} = (1-0.125) * \underline{98.13} + 0.125*130 = \underline{102.1}$$

$$\text{EstimatedRTT} = (1-0.125) * \underline{102.1} + 0.125*110 = \underline{103.1}$$

$$\text{EstimatedRTT} = (1-0.125) * \underline{103.1} + 0.125*106 = 103.5$$

ب. DevRTT را بعد از دریافت هر کدام از این مقادیر محاسبه کنید. (مقدار بتا را ۰,۲۵ و مقدار DevRTT قبل از دریافت این بسته ها را ۵ میلی ثانیه در نظر بگیرید).

$$\text{DevRTT} = (1 - \beta) * \text{DevRTT} + \beta * (\text{EstimatedRTT} - \text{SampleRTT})$$

$$\text{DevRTT} = (1-0.25) * 5 + 0.25*(98.13-85) = 7.03$$

$$\text{DevRTT} = (1-0.25) * 7.03 + 0.25*(102.1-130) = 12.25$$

$$\text{DevRTT} = (1-0.25) * 12.25 + 0.25*(103.1-110) = 10.91$$

$$\text{DevRTT} = (1-0.25) * 10.91 + 0.25*(103.5-106) = 8.8$$

پ. TCPTimeoutInterval را بعد از دریافت هر کدام از این مقادیر محاسبه کنید.

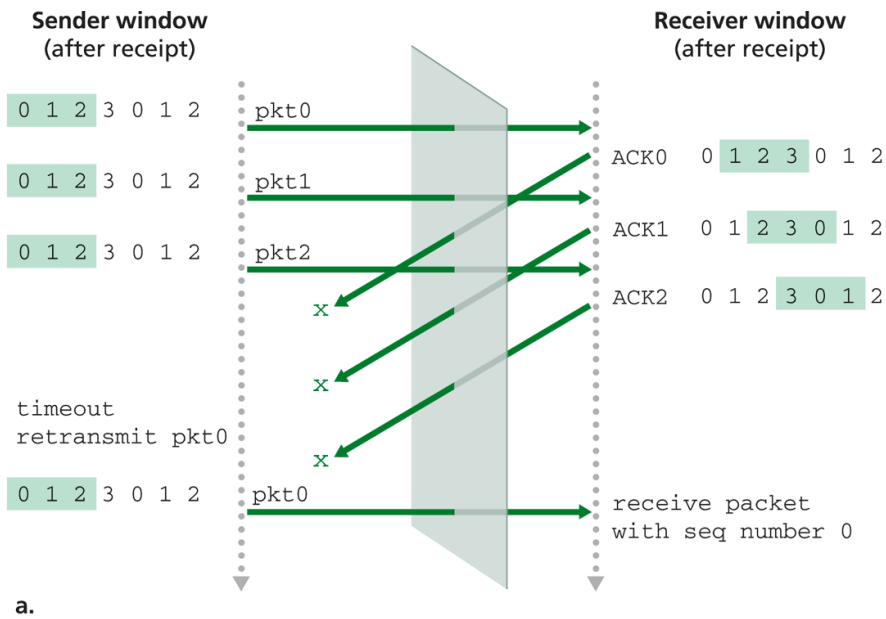
$$\text{TCPTimeoutInterval} = \text{EstimatedRTT} + 4*\text{DevRTT}$$

126.25, 151.1, 146.74, 136.75

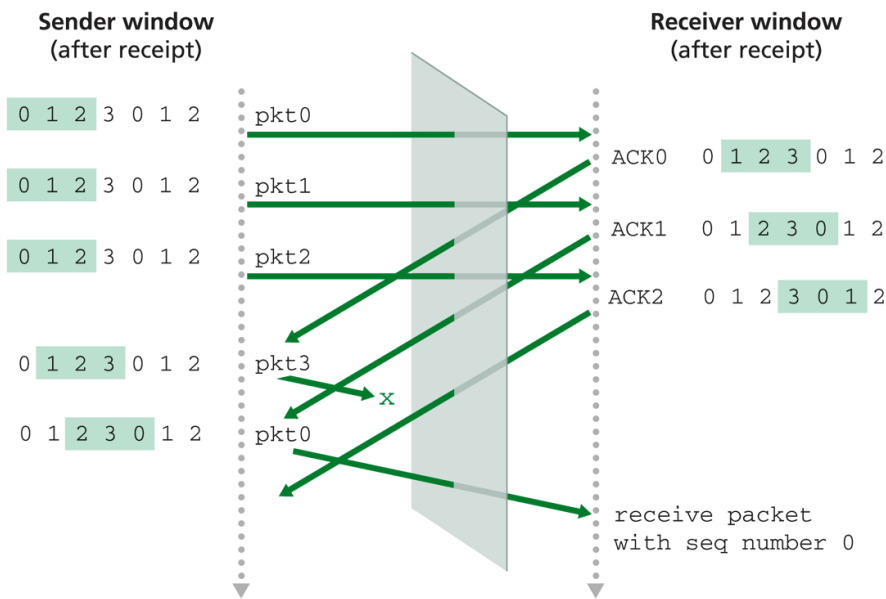
۷- به نظر شما چرا پروتکل TCP، SampleRTT را برای بسته های دوباره ارسال شده در نظر نمی گیرد؟

چون ممکن است ACK ای که برای بسته ی دوباره ارسال شده آمده است، مربوط به بسته ی اولی بوده که با تاخیر رسیده است. در نتیجه این زمان واقعی نیست.

پیوست یک: شکل شماره ۲۷ فصل سوم



a.



b.

Figure 3.27 ♦ SR receiver dilemma with too-large windows: A new packet or a retransmission?