

به نام خدا

تمرین سری سوم درس شبکه های کامپیوتری

---

1. به سوالات زیر پاسخ دهید: (50 نمره)

الف) در صورتی که برنامه روی UDP اجرا شود آیا امکان اینکه برنامه از انتقال داده ی قابل اعتماد بهره ببرد، وجود دارد؟ توضیح دهید.

بله، می توان این قابلیت را در لایه کاربرد یا Application اضافه کرد.

ب) چه نوع خطاهایی توسط checksum و چه نوع خطاهایی توسط sequence number قابل تشخیص هستند؟  
خطاهای بیتی مربوط به بسته ها توسط checksum و خطاهای مربوط به ترتیب و سالم رسیدن بسته ها توسط sequence number قابل تشخیص هستند.

ج) در تعیین زمان timeout پروتکل TCP در صورتی که این زمان را خیلی بزرگ یا خیلی کوچک در نظر بگیریم چه مشکلاتی پیش می آید؟

اگر زمان timeout را خیلی کوچک در نظر بگیریم ممکن است ارسال مجدد های بیهوده زیادی داشته باشیم و باعث ایجاد ترافیک زیاد و شلوغی در شبکه شود؛ اگر زمان timeout را خیلی بزرگ در نظر بگیریم ممکن است سرعت عکس العمل شبکه نسبت به گم شدن بسته کم شود و شبکه ی کندی داشته باشیم.

د) مقدار Internet Checksum برای دو عدد زیر را محاسبه کنید. (نمایش اعداد بصورت hex می باشد)

0x7BE0  
0x8653

$$\begin{array}{r} 0x7BE0 \\ + 0x8653 \\ \hline 0x40233 \\ + 0x0001 \\ \hline 0x0234 \end{array}$$

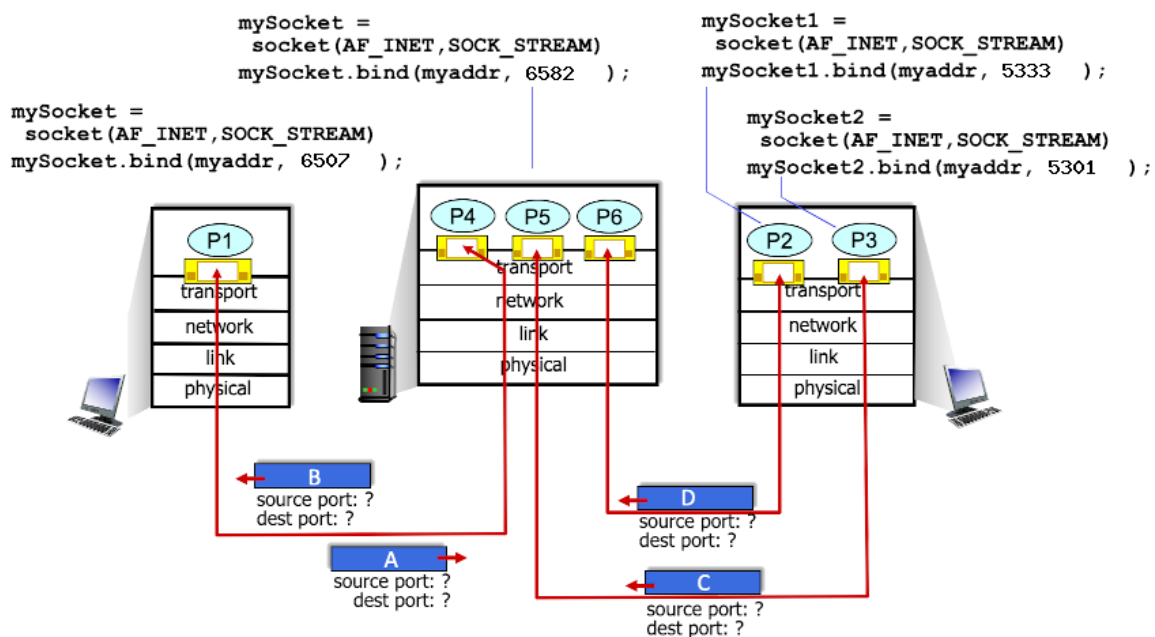
→ Internet checksum =  $0xFFFF - 0x0234 = (0x0234)' = 0xFDCB$

ه) در پروتکل rdt 3.0 بسته های ACK که از مقصد به مبدا در جریان هستند نیازی به sequence number ندارند (یعنی در صورت حذف شدن sequence number ارسال بسته ها با مشکلی مواجه نمی شود)

چرا بسته های ACK به sequence number احتیاج ندارند؟

چون این پروتکل stop & wait است و تا دریافت ACK بسته ارسال شده کنونی به سراغ بسته بعدی نمی رود و آن را ارسال نمی کند.

ز) با توجه به تصویر زیر مقادیر پورت های مبدا و مقصد هر یک از بسته های A تا D را بدست آورید. (با توجه به شکل سوکت وسط مربوط به سرور و سوکت های دیگر مربوط به Client ها می باشند)



با توجه به شکل می توان نوشت:

بسته A: پورت مبدا 6507 و مقصد 6582

بسته B: پورت مبدا 6582 و مقصد 6507

بسته C: پورت مبدا 5301 و مقصد 6582

بسته D: پورت مبدا 5333 و مقصد 6582

2. بسته ای توسط لایه انتقال TCP دریافت شده است و اطلاعات مربوط به هدر TCP این بسته در شکل زیر موجود است. با توجه به داده های موجود در شکل به سوالات زیر پاسخ دهید: (30 نمره)

39	7c	01	bb
b6	c1	40	86
b5	3d	56	00
50	11	02	03
a4	da	00	00

الف) پورت مبدا و مقصد در این بسته چیست؟ دو بایت اول نشان دهنده پورت مبدا و دو بایت بعدی نشان

دهنده پورت مقصد است پس پورت مبدا برابر با 14716 و پورت مقصد برابر با 443 است

ب) این بسته چندمین بایت را ACK می کند؟ آیا این مقدار، یک مقدار valid است؟

با توجه به خط سوم (چهار بایت سوم) این بسته 3,040,695,808 امین بایت را ACK می کند و چون بیت

مربوط به ACK در این بسته برابر با 1 است پس این مقدار یک مقدار valid می باشد

ج) این بسته مربوط به کدام قسمت از یک کانکشن TCP می باشد؟

چون بیت های FIN و ACK این بسته برابر با 1 می باشند پس این بسته مربوط به اتمام یک کانکشن TCP

می باشد

د) Sequence Number این بسته چه مقدار است؟ این عدد نشان دهنده چیست؟

این مقدار نشان دهنده ی شماره اولین بایت ارسالی است. 0xb6c14086

---

3. کاربر A و B با یکدیگر بر بستر TCP ارتباط برقرار کرده اند. همچنین کاربر B تمام بایت ها تا بایت 126م را از کاربر A دریافت کرده است. سپس فرض کنید که کاربر A سه بسته به صورت پشت سر هم با سایز های 80 و 40 و 80 بایت را برای کاربر B ارسال میکند. در اولین بسته sequence number 127 میباشد. پورت مبدا ارسالی 5445 و پورت مقصد دریافتی 80 میباشد. سپس کاربر B اقدام به ارسال بسته ACK میکند. حال به سوالات زیر پاسخ دهید: (90 نمره)

الف) در بسته سوم، مقدار sequence number و پورت مبدأ و پورت مقصد چیست ؟

در این بسته، Sequence Number برابر با 247، پورت مبدا 5445 و مقصد 80 می باشد.

ب) در صورتی که بسته اول قبل از بسته دوم برسد، در بسته ACK ای که پس از دریافت بسته اول توسط

گیرنده ارسال میشود، مقدار ACK number و پورت مبدأ و پورت مقصد چیست ؟

در این بسته، ACK Number برابر با 207، پورت مبدا 80 و مقصد 5445 می باشد.

ج) در صورتی که بسته دوم قبل از بسته اول برسد، در بسته ACK ای که پس از دریافت بسته دوم توسط

گیرنده ارسال میشود، مقدار ACK number و پورت مبدأ و پورت مقصد اولین بسته ACK ای که به مقصد

رسیده چیست ؟

در این بسته، ACK Number برابر با 127، پورت مبدا 80 و مقصد 5445 می باشد.

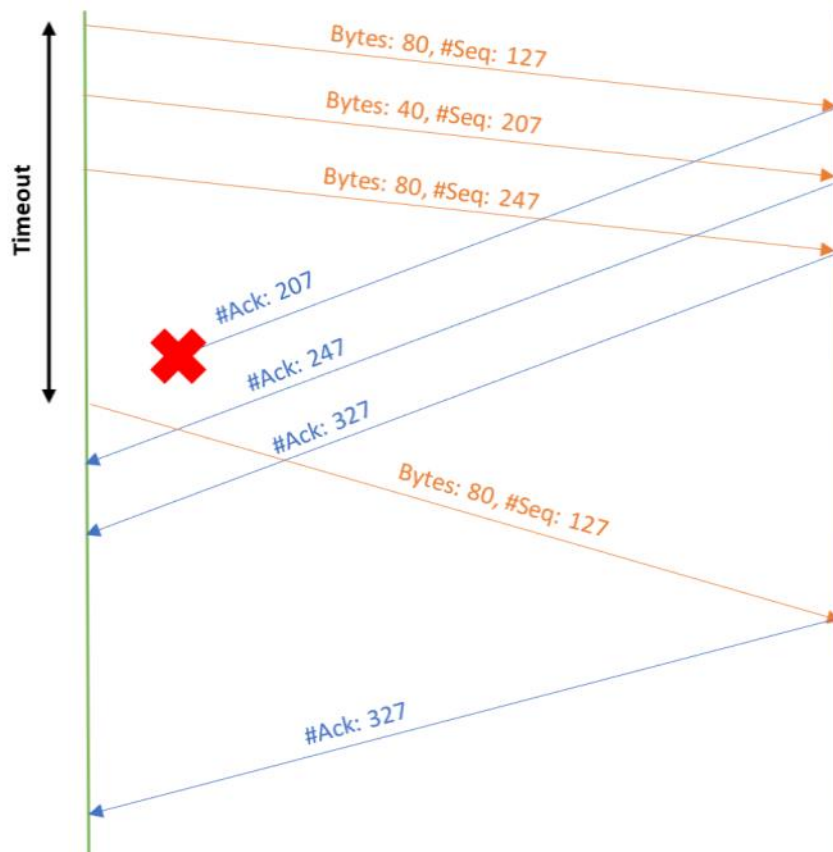
د) اگر به دلیل شلوغی شبکه، ابتدا بسته سوم به مقصد برسد و سپس بسته اول و دوم به ترتیب به B برسند.

در اینصورت ACK هایی که در هر مرحله ارسال می شود به چه صورت می باشند؟

ابتدا ACK با شماره 127 ارسال می شود، سپس 207 و در نهایت در صورت وجود بافر 327 و در صورت

عدم وجود بافر 247 ارسال می شود.

ه) فرض کنید که بسته ها به همان ترتیب که در A ارسال میشود، به دست B میرسد و اولین ACK هم گم شده است و دومین ACK هم بعد از اولین time out میرسد. یک دیاگرام زمانی رسم کنید و تمام اطلاعات این سه بسته و همچنین ACK های آن ها را کامل بیان کنید. (فرض کنید که هیچ گم شدن بسته ی دیگری رخ نمیدهد) برای هر بسته ارسالی در شکل خود، مقدار sequence number و اندازه اطلاعات به بایت و برای هر بسته ی دریافتی مقدار ACK number مشخص باشد.



4. در یک کانکشن TCP، مقدار Estimated RTT اولیه برابر با 120 میلی ثانیه و مقدار DevRTT اولیه برابر با 5ms است. حال فرض کنید که 4 اندازه گیری RTT برای این کانکشن انجام می شود و مقادیر آنها به صورت  $R_1=128ms$  و  $R_2=135ms$  و  $R_3=141ms$  و  $R_4=110ms$  باشد. اگر  $\alpha=0.125$  باشد و  $\beta=0.25$  باشد در اینصورت بعد از هر اندازه گیری مقدار Estimated RTT و Dev RTT و در نهایت Timeout Interval را بدست آورید. (50نمره)

$$Estimated\ RTT = (1 - \alpha) \times Estimated\ RTT + \alpha \cdot Sample\ RTT$$

$$DevRTT = (1 - \beta) \times DevRTT + \beta |Estimated\ RTT - Sample\ RTT|$$

$$\Rightarrow R_1 = 128ms$$

$$\Rightarrow Estimated\ RTT = (1 - 0.125) \times 120 + 0.125 \times 128 = 121ms$$

$$Dev\ RTT = (1 - 0.25) \times 5 + 0.25 \times (128 - 121) = 5.5ms$$

$$R_2 = 135ms$$

$$\Rightarrow Estimated\ RTT = (1 - 0.125) \times 121 + 0.125 \times 135 = 122.75ms$$

$$Dev\ RTT = (1 - 0.25) \times 5.5 + 0.25 \times (135 - 122.75) = 7.1875ms$$

$$R_3 = 141ms$$

$$\Rightarrow Estimated\ RTT = (1 - 0.125) \times 122.75 + 0.125 \times 141 = 125.03125ms$$

$$Dev\ RTT = (1 - 0.25) \times 7.1875 + 0.25 \times (141 - 125.03125) = 9.38281ms$$

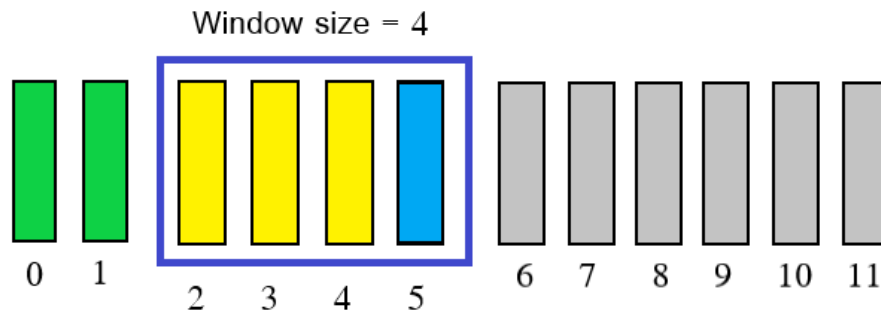
$$R_4 = 110\ ms$$

$$\Rightarrow Estimated\ RTT = (1 - 0.125) \times 125.03125 + 0.125 \times 110 = 123.1523ms$$

$$Dev\ RTT = (1 - 0.25) \times 9.38281 + 0.25 \times (123.1523 - 110) = 10.325195ms$$

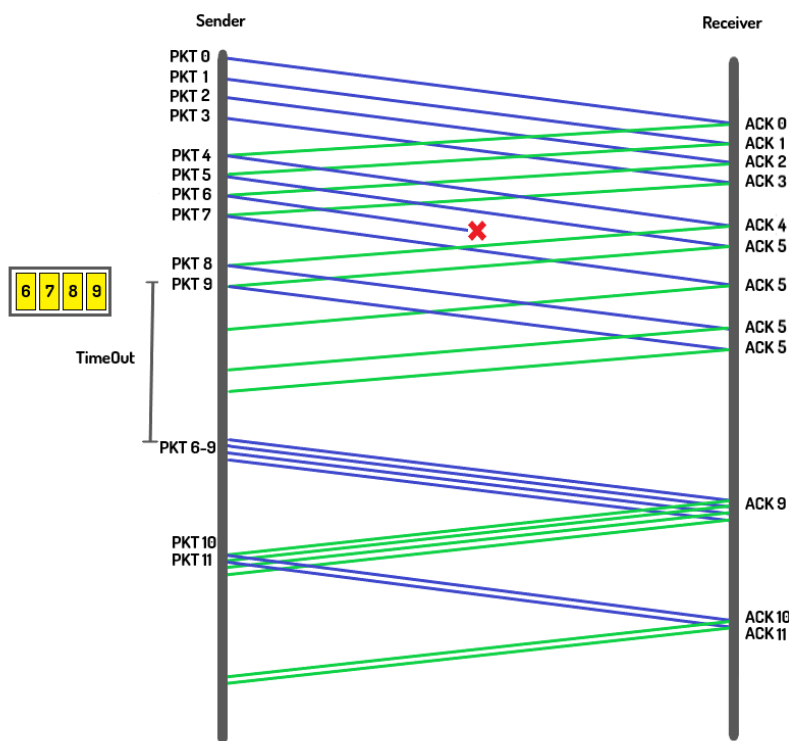
$$Timeout\ Interval = Estimated\ RTT + 4 \times Dev\ RTT = 164.4531\ ms$$

-  Already Acked
-  Sent but not Acked
-  Not yet sent
-  Not Usable



الف) بسته با عدد توالی 6 گم شود.

GBN: در این حالت تمامی بسته های 0 تا 5 به مقصد رسیده و ack آنها نیز توسط مبدا دریافت می شود. هنگامی که ack بسته 2 دریافت می شود، بسته 6 ارسال می شود که به مقصد نمی رسد. اما پس از رسیدن هر ack از بسته های 3 تا 5، پنجره ارسال شیفت پیدا می کند پس هنگامی که بسته 7 به مقصد می رسد، مقصد ack تکراری ارسال می کند (آخرین ack ی که در بافر وجود دارد یعنی ack بسته 5) در این حین، پنجره ارسال هم شیفت پیدا کرده زیرا ack بسته های 4 و 5 دریافت می شود پس بسته های 8 و 9 ارسال شده اند ولی چون ack در این روش تجمعی است، باز هم ack بسته 5 ارسال می شود. در نهایت هنگامی که تایمر پنجره timeout می شود، تمام بسته های حاضر در پنجره یعنی بسته های 6 تا 9 به صورت همزمان ارسال می شوند و ack خود را دریافت می کنند.



SR: در این حالت بسته های 0 تا 5 به مقصد رسیده و ack آنها هم توسط مبدا دریافت می شود. پس از رسیدن ack بسته 2، پنجره ارسال شیفت پیدا می کند و بسته 6 ارسال می شود که به مقصد نمی رسد. با رسیدن ack بسته های 3 و 4 و 5 پنجره ارسال شیفت پیدا می کند و بسته های 7 و 8 و 9 ارسال می شوند و ack آنها هم دریافت می شود. ولی چون بسته 6 به مقصد نرسیده است و ack آن برای مبدا ارسال نشده است پس ack بسته های 7 و 8 و 9 در مبدا بافر می شوند. با timeout شدن تایمر، بسته 6 دوباره ارسال می شود و با دریافت ack آن، پنجره ارسال چهار بار شیفت داده می شود و روند ارسال با رسیدن بسته های 10 و 11 به مقصد و ack آنها به اتمام می رسد.



ب) اگر ACK بسته با عدد توالی 8 گم شود.

GBN: در این حالت تمامی بسته های 0 تا 7 به مقصد رسیده و ack آنها توسط مبدا دریافت می شود. پس از رسیدن ack بسته شماره 4، پنجره ارسال یک شیفت پیدا می کند و تایمر مجدداً start می شود و بسته شماره 8 ارسال می شود. پس از رسیدن هر یک از بسته های 5 و 6 و 7، پنجره ارسال در هر مورد یک بار شیفت پیدا می کند و مجدداً تایمر برای اولین بسته پنجره (بسته شماره 8) شروع به کار می کند. به دلیل اینکه ack بسته های 9 و 10 و 11 قبل از timeout می رسند و ack ها به صورت تجمعی هستند، با رسیدن ack این بسته ها، بسته 8 نیز ack می شود و هیچ ارسال دوباره ای اتفاق نمی افتد. و روند ارسال با رسیدن ack بسته 11 اتمام می شود.

---

SR: در این حالت تایمر بسته های ارسالی مستقل از یکدیگر هستند و تمامی بسته های 0 تا 7 به مقصد می رسند و ack آنها هم توسط مبدا دریافت می شود و با هر بار دریافت ack این بسته ها پنجره ارسال یک شیفت پیدا می کند و هنگامی که ack بسته 8 گم می شود تا هنگامی که تایمر این بسته timeout نشود، پنجره ارسال ثابت می ماند ولی ack بسته های 9 و 10 و 11 دریافت میشود و با ارسال مجدد بسته 8 و دریافت ack پنجره ارسال چهار بار شیفت پیدا می کند.

ج) با توجه به سناریوی قسمت "الف"، فاصله زمانی بین ارسال اولیه بسته 6 و ارسال مجدد آن چقدر است؟

SR: در این حالت چون هر بسته تایمر مربوط به خود را دارد پس از  $RTT \times 1.5$  (مدت زمان timeout) تایمر مربوط به بسته 6 ریست می شود و این بسته دوباره ارسال می شود

GBN: در این حالت با هر بار شیفت پیدا کردن پنجره ارسال تایمر کلی ریست می شود. پس با توجه به شکل پس از ارسال بسته 6، حدود  $RTT \times 1$  طول می کشد تا ack بسته های 4 و 5 به مبدا برسند و پنجره ارسال شیفت پیدا کند و بسته 6 در ابتدای پنجره ارسال قرار گیرد. پس از آن به دلیل آمدن ack های تکراری پنجره شیفت پیدا نمی کند و با گذشتن  $RTT \times 1.5$  (مدت زمان timeout) بسته 6 به همراه بقیه بسته های پنجره ارسال می شوند. پس فاصله زمانی بین اولین ارسال و ارسال مجدد بسته 6 در کل برابر با  $RTT \times 2.5$  است

---