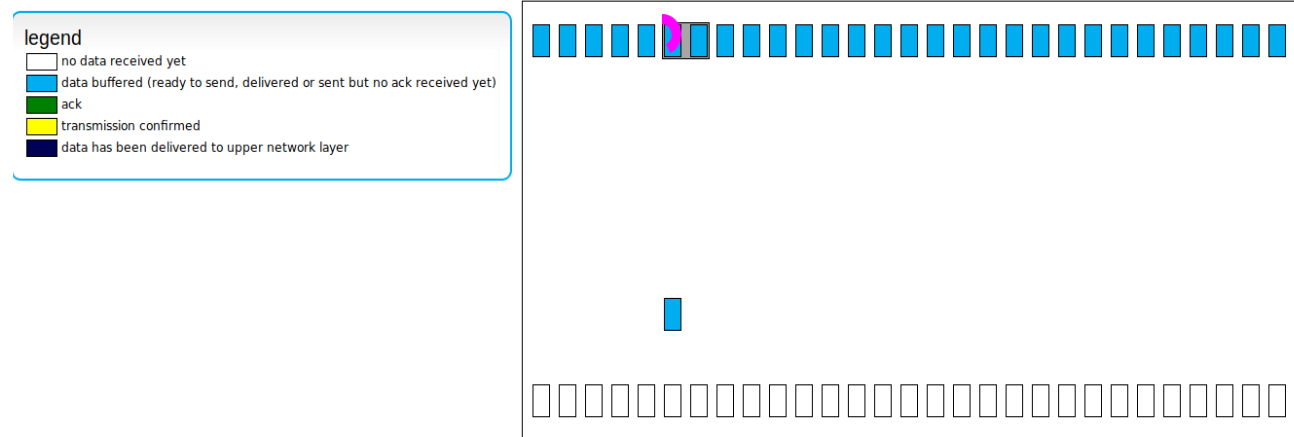
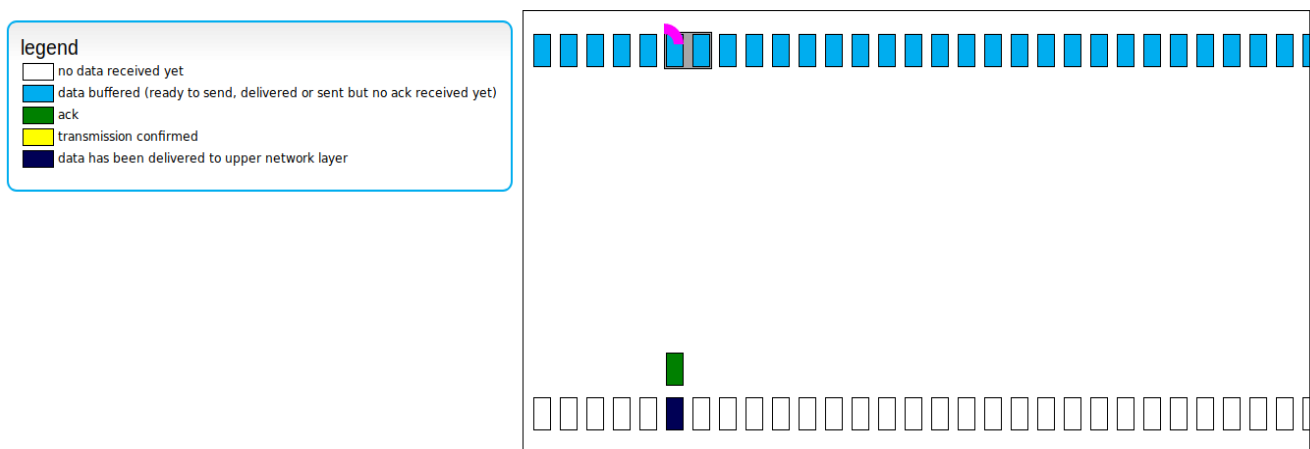


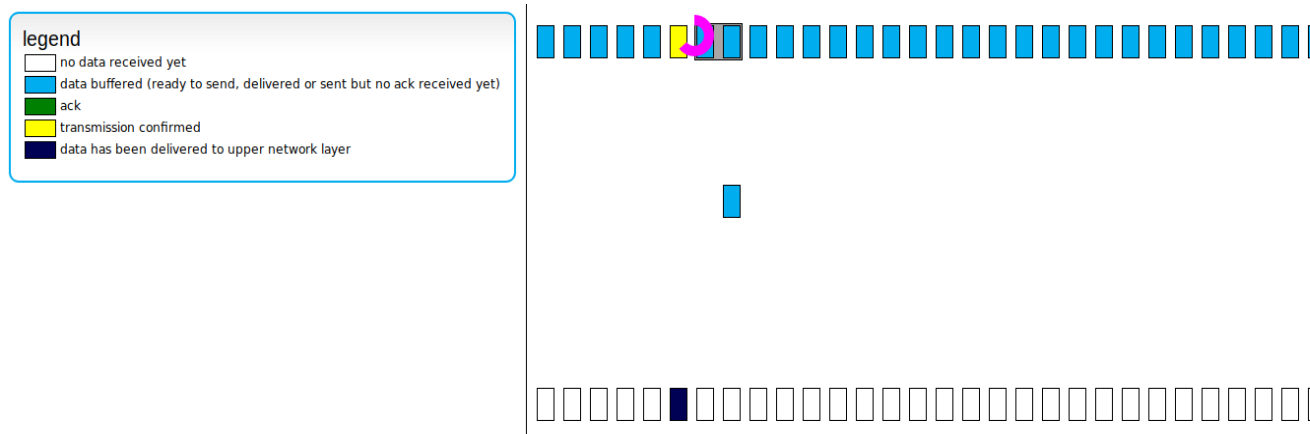
پاسخ سوال اول قسمت (الف)



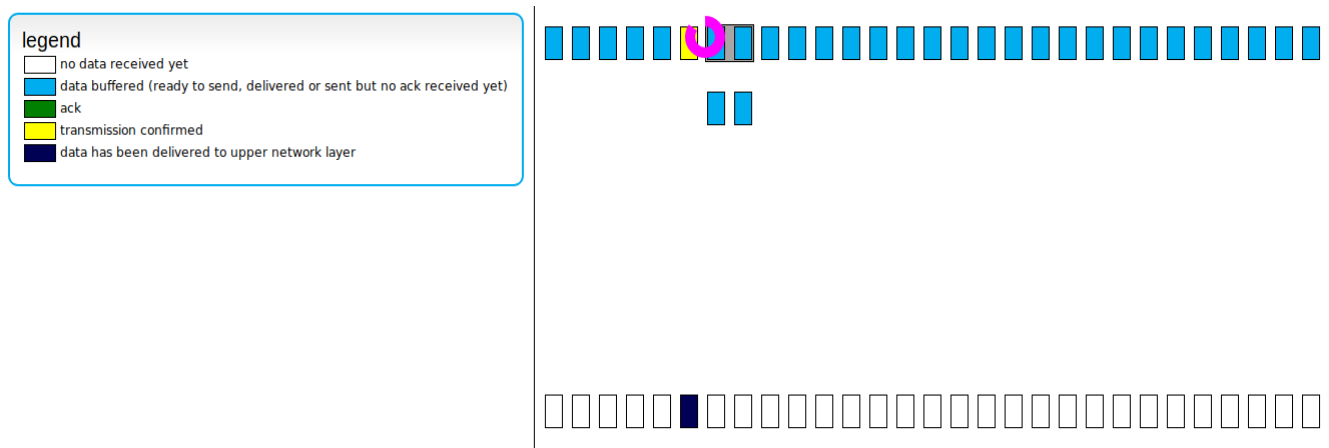
طبق عکس فوق ، پکت یک گم شده است پس فقط پکت 0 به مقصد میرسد.



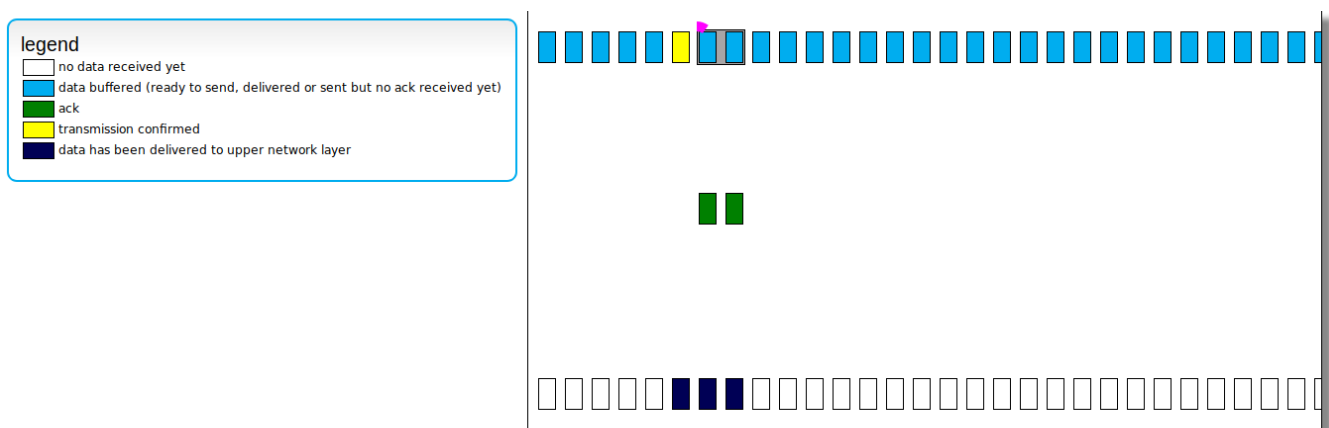
طبق عکس فوق ، Ack پکت یک از مشتری به سرور ارسال میشود.



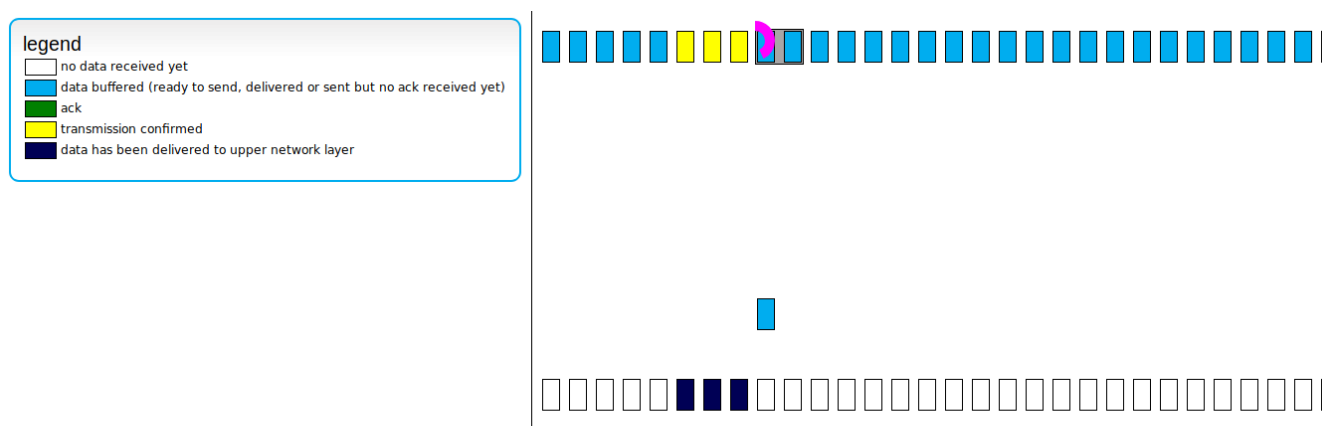
قبل از timeout شدن پکت 1، چون ack پکت 1 میرسه، پنجره یکی به جلو میره و پکت 2 ارسال میشه.



بعد از timeout شدن پکت 1، چون آخرین آکی که رسیده مربوط به پکت 0 بوده و از آنجایی که پنجره در حال حاضر شامل پکت 1 و 2 است پس این دو بسته مجدداً ارسال میشوند.

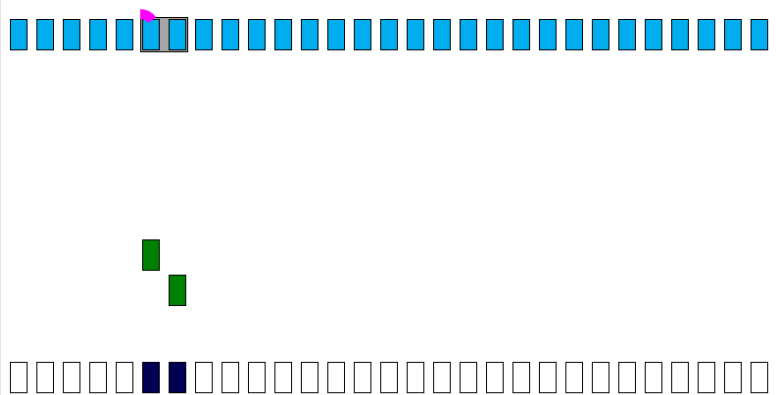
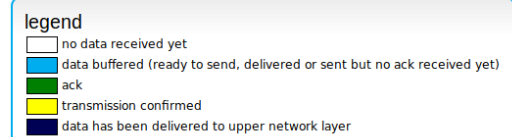


بعد از اینکه مشتری پکت های 1 و 2 را دریافت کرد برای هر یک به صورت مجزا آک برمیگرداند (اگر آک بسته ی 1 گم شود هیچ اشکالی پیش نمی آید، زیرا پروتکل GBN آک تجمعی دارد)

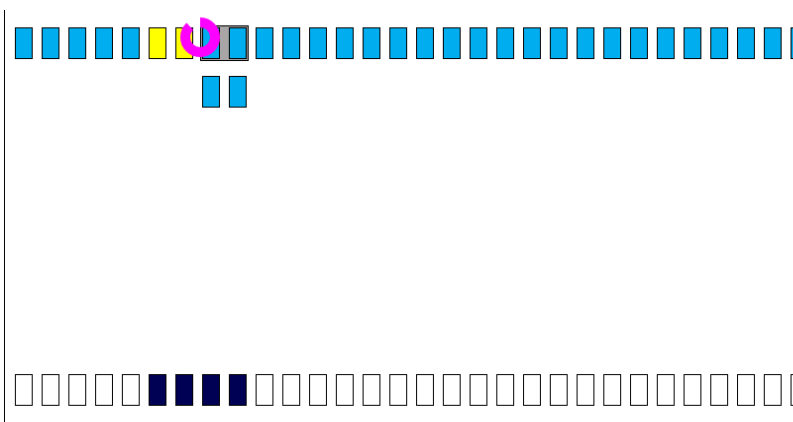
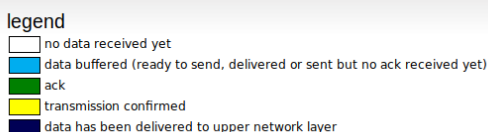


بعد از دریافت آک های مربوط به بسته های 1 و 2، بسته ی آخر یعنی پکت 3 نیز برای مشتری ارسال میشود. در نهایت هم مشتری آک مربوط به بسته ی سه را برای سرور میفرستد.

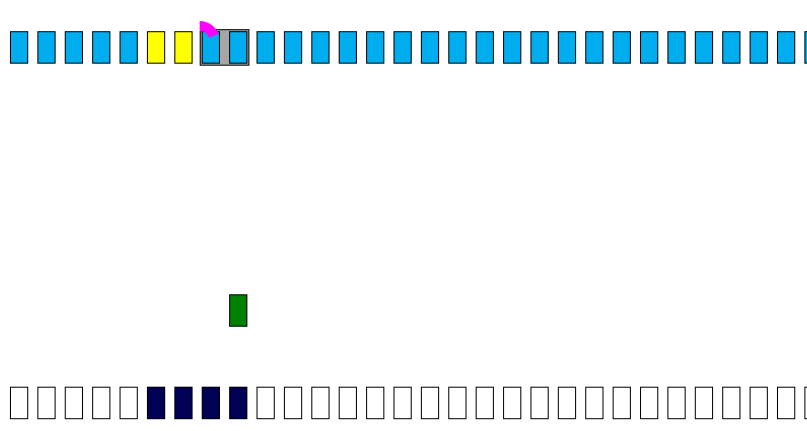
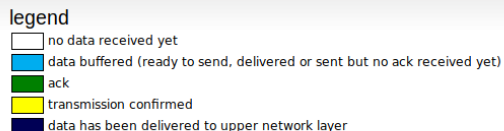
قسمت (ب)



دو پکت 0 و 1 با موفقیت ارسال میشوند و اک آنها هم بدون هیچ مشکلی به سرور ارسال خواهد شد. بعد از آن پنجره دو تا به جلو میرود و پکت 2 و 3 با موفقیت ارسال می شوند اما در مرحله ی برگشت اک ، هر دو اک گم میشوند... پس سرور بعد از timeout شدن مجدداً این دو بسته را ارسال میکند.

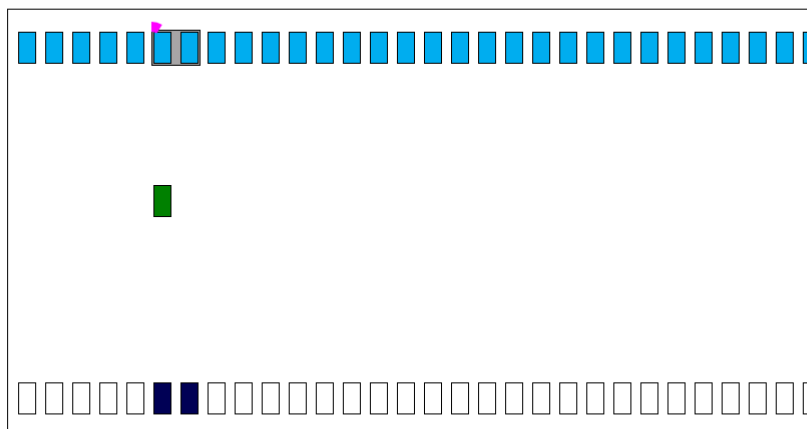
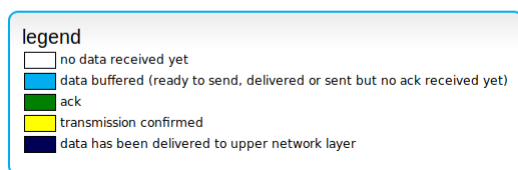


(در عکس هم پیداست که دو پکت 2 و 3 به مشتری رسیده اند اما چون سرور اکی از آن دو دریافت نکرده مجدداً این دو بسته را ارسال کرده)

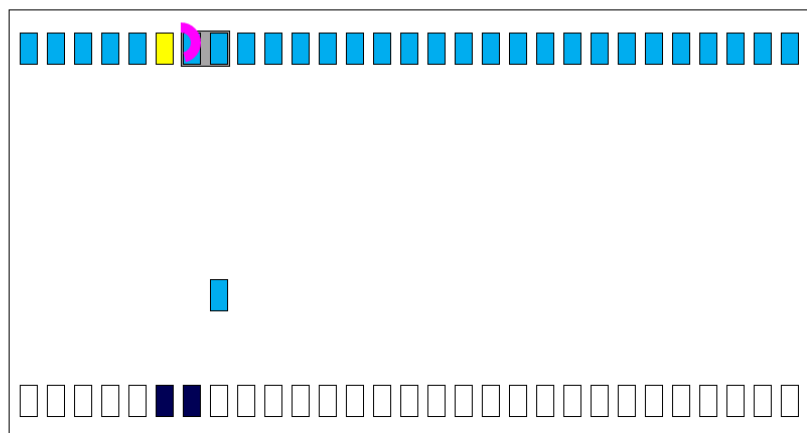
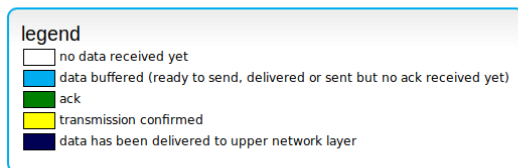


در نهایت هم مشتری یک اک تجمعی برای سرور برمیگرداند.

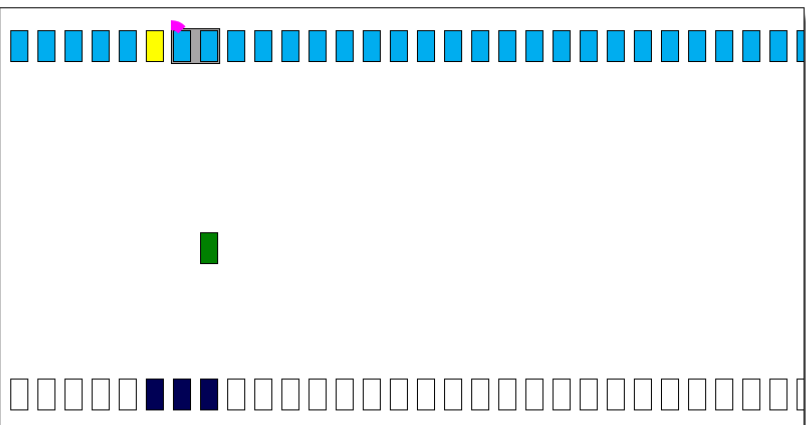
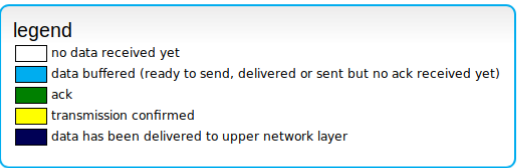
قسمت (ج)



آک بسته ی ۱ گم هنگام بازگشت گم میشود.

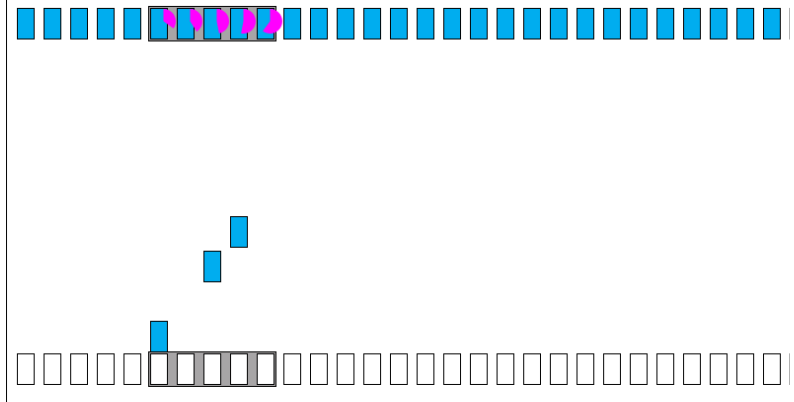
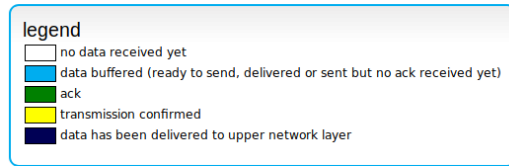


فرستنده بعد از دریافت آک بسته ی ۰ ، پنجره اش یکی به جلو میرود و بسته ی ۲ را ارسال میکند

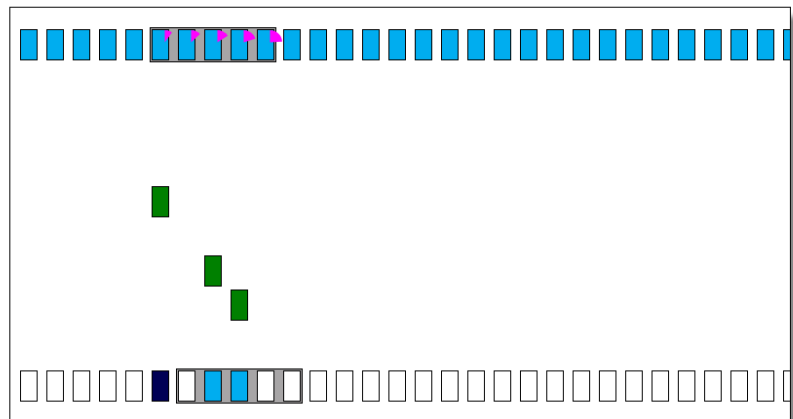
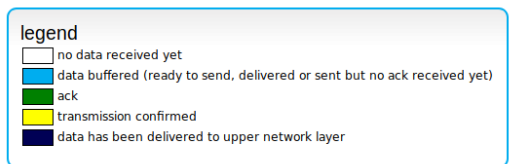


سپس آک بسته ی ۲ برمیگردد و از آنجایی که در پروتکل GBN آک تجمعی داریم، پس برگشتن آک ۲ به این معناس که بسته های ۰ تا ۲ به درستی توسط گیرنده دریافت شده اند. پس پنجره از موضع فعلیش دو تا جلو میره و بسته ی ۳ رو ارسال و آک آن را هم بدون هیچ مشکلی دریافت میکند.

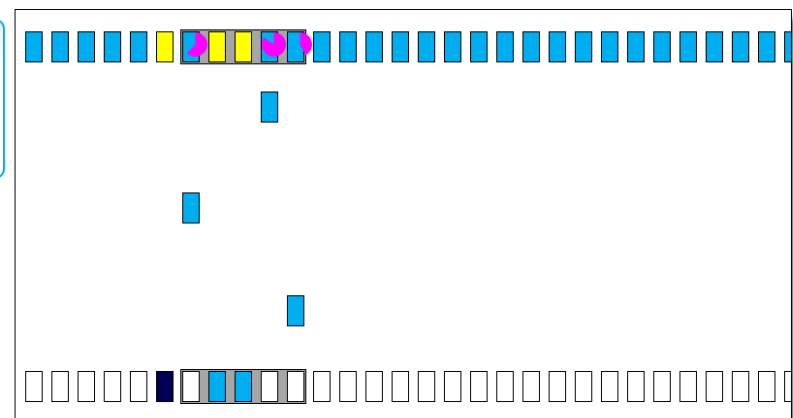
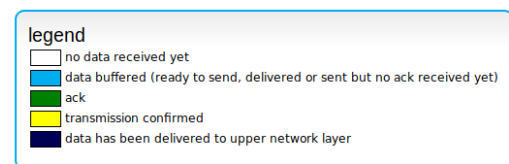
پاسخ سوال دوم قسمت (الف)



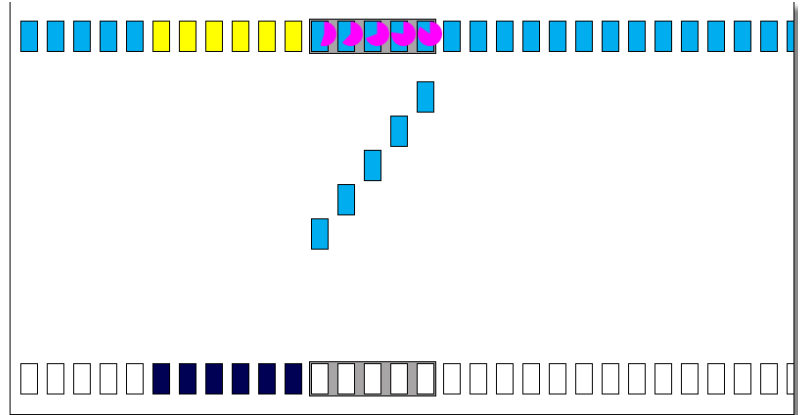
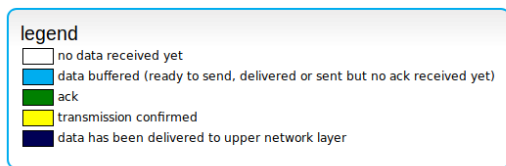
پکت ۱ و ۴ گم شده اند.



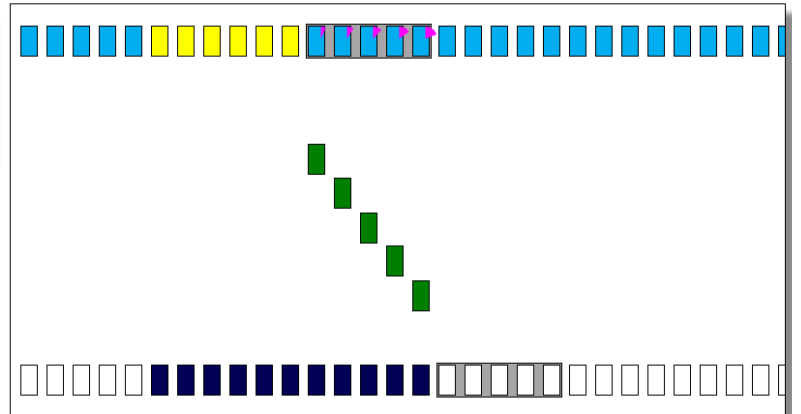
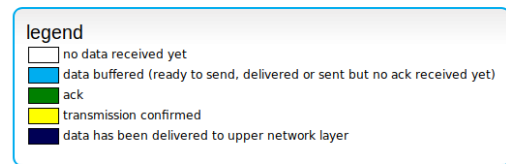
اک پکت های 0 و 2 و 3 برمیگردد.



طبق عکس فوق ، بعد از رسیدن اک پکت های 0 و 2 و 3 پنجره یکی به جلو میرود (چون اک پکت ۱ نرسیده پنجره نمیتواند به اندازه ی ۴ تا جلو برود ولی اک های سه بسته ای که دریافت شده را نگه میدارد.) و بلافاصله پکت ۵ ارسال میشود. ضمناً بعد از timeout شدن پکت های 1 و 4 این دو بسته مجدداً ارسال میشوند.

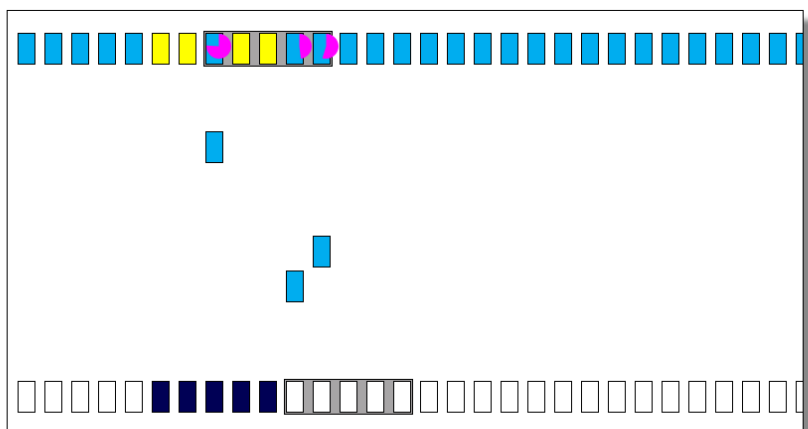


بعد از ارسال پکت 5، 1 و 4 و نیز دریافت آک این پکت ها، پنجره از آفرین موضع خود 5 تا جلو میرود و پکت 6، 7 و 8 ارسال میشوند. (پکت های 9 و 10 را نادیده بگیرید :) !)

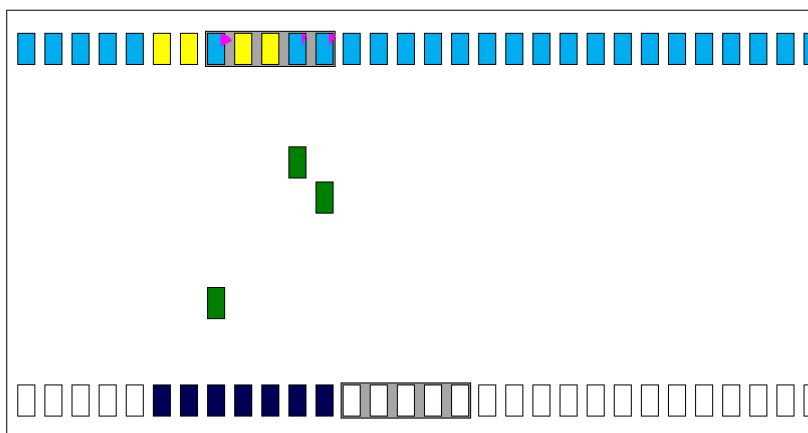


در نهایت هم آک پکت های 6 تا 8 بدون هیچ مشکلی به سرور برمیگردد .

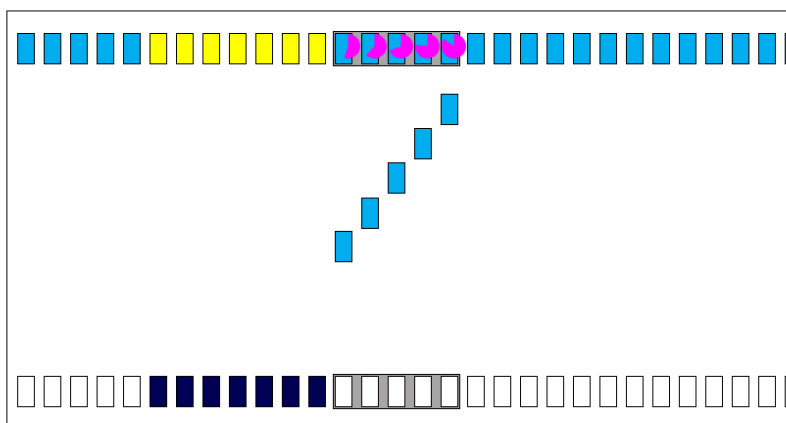
© 2006 The Authors
Journal compilation © 2006 Blackwell Publishing Ltd



timeout شدن یکت ۲ ، این بسته نیز مجدداً ارسال میشود.



آک بسته های ۲، ۵ و ۶ نیز به سمت فرستنده برمیگردد.



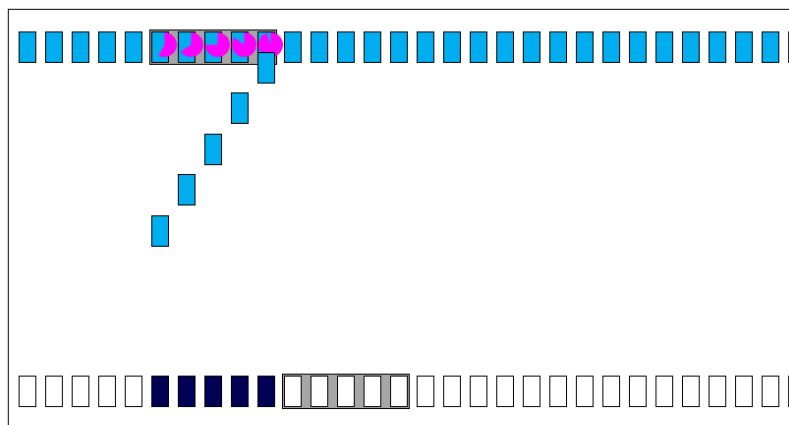
بعد از رسیدن اک بسته های ۲، ۵ و ۶ به فرستنده، پنجره فرستنده از موضع قبل خود به اندازه ی ۵ تا جلو می‌رود

و بسته های ۷ و ۸ نیز ارسال میشوند (به مابقی بسته ها توجه نکنید) در نهایت هم اک این بسته ها با

موفقیت (!) در فرستنده دریافت میشود.

legend

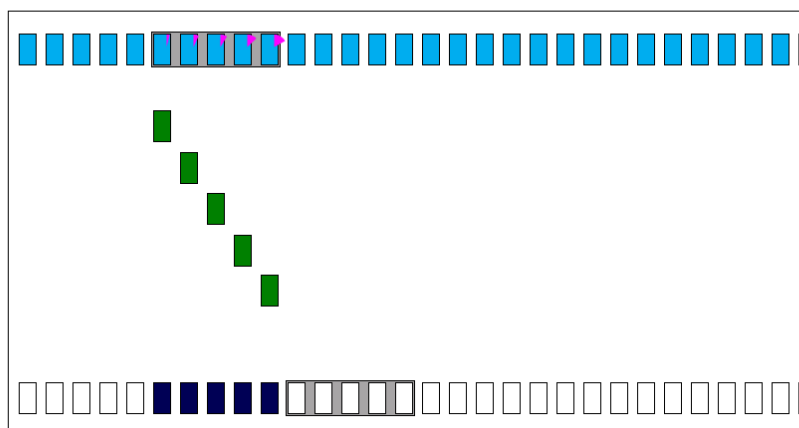
- no data received yet
- data buffered (ready to send, delivered or sent but no ack received yet)
- ack
- transmission confirmed
- data has been delivered to upper network layer



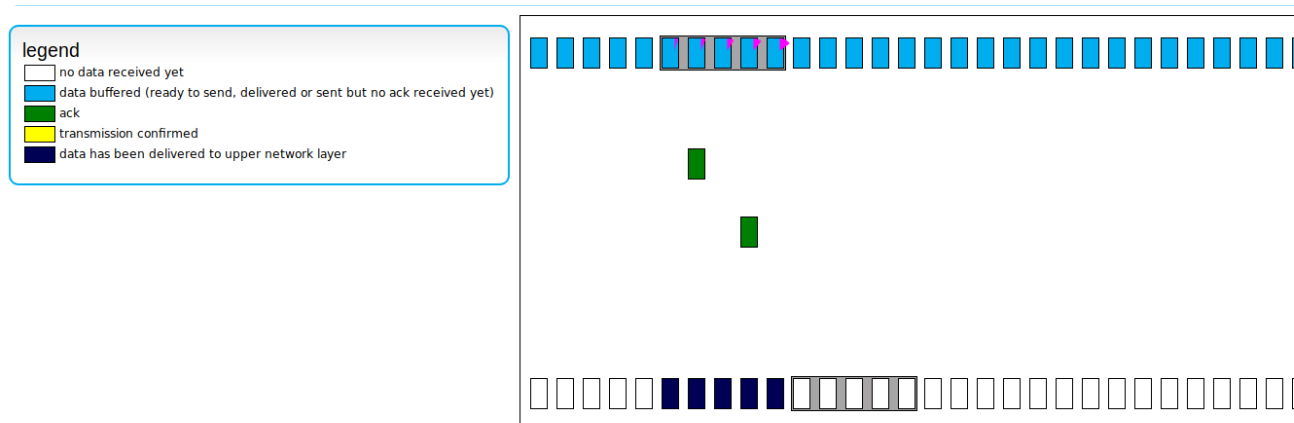
بعد از گم شدن آک بسته های ۰ تا ۴ (۵ بسته ی اول) ، پنجره ی دریافت کننده یا مشتری ۵ تا به جلو میرود چون هیچ مشکلی در دریافت بسته ها نداشته ولی پنجره ی فرستنده حرکتی نمیکند و بعد از timeout شدن بسته های ۰ تا ۴ ، مجدداً آنها را ارسال میکند.

legend

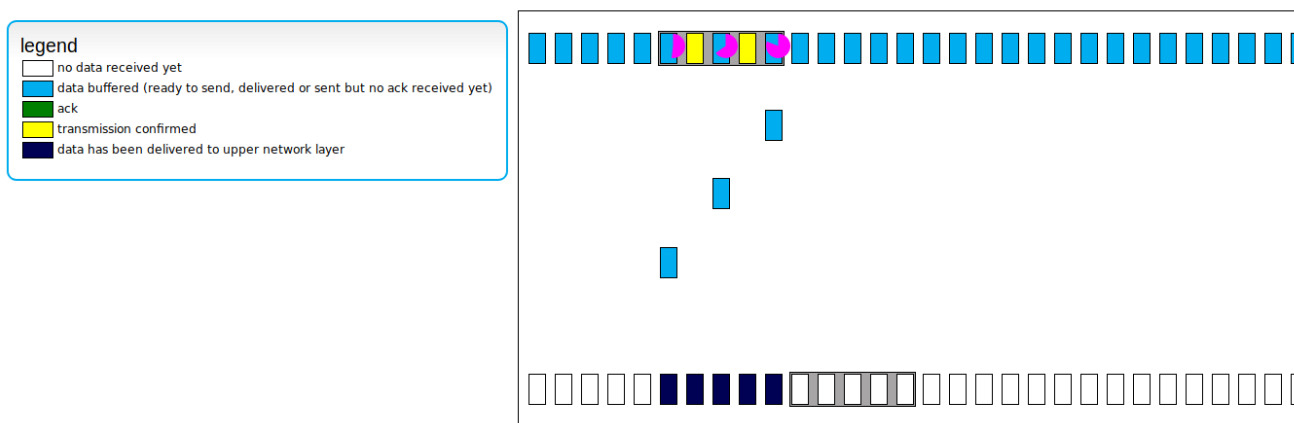
- no data received yet
- data buffered (ready to send, delivered or sent but no ack received yet)
- ack
- transmission confirmed
- data has been delivered to upper network layer



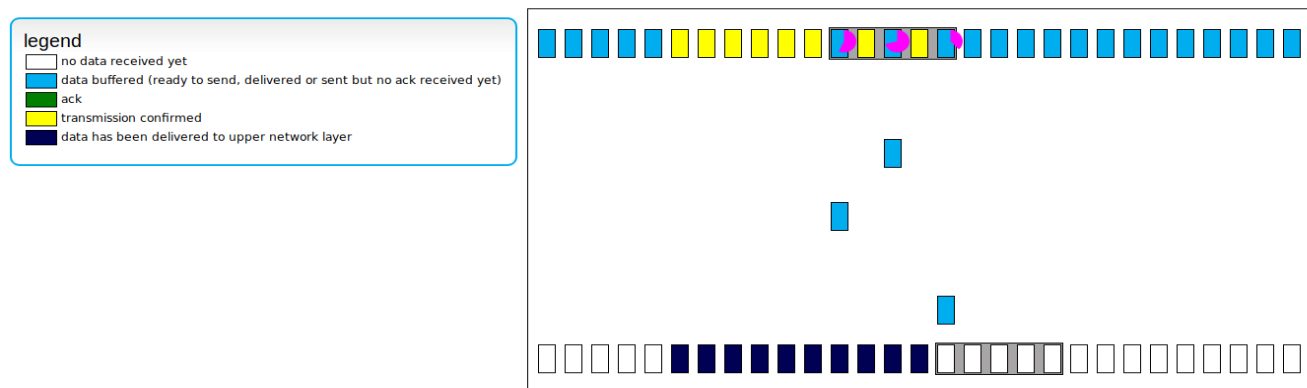
دریافت کننده بعد از دریافت ۵ بسته ی اول ، برای هر یک از آنها آک مجزا ارسال میکند . بعد از این مرحله هم ، پنجره ی فرستنده ۵ تا به جلو میرود و بدون هیچ مشکلی ، مابقی بسته ها (بسته های ۵ تا ۸) را ارسال کرده و آک آنها را نیز با موفقیت دریافت میکند !



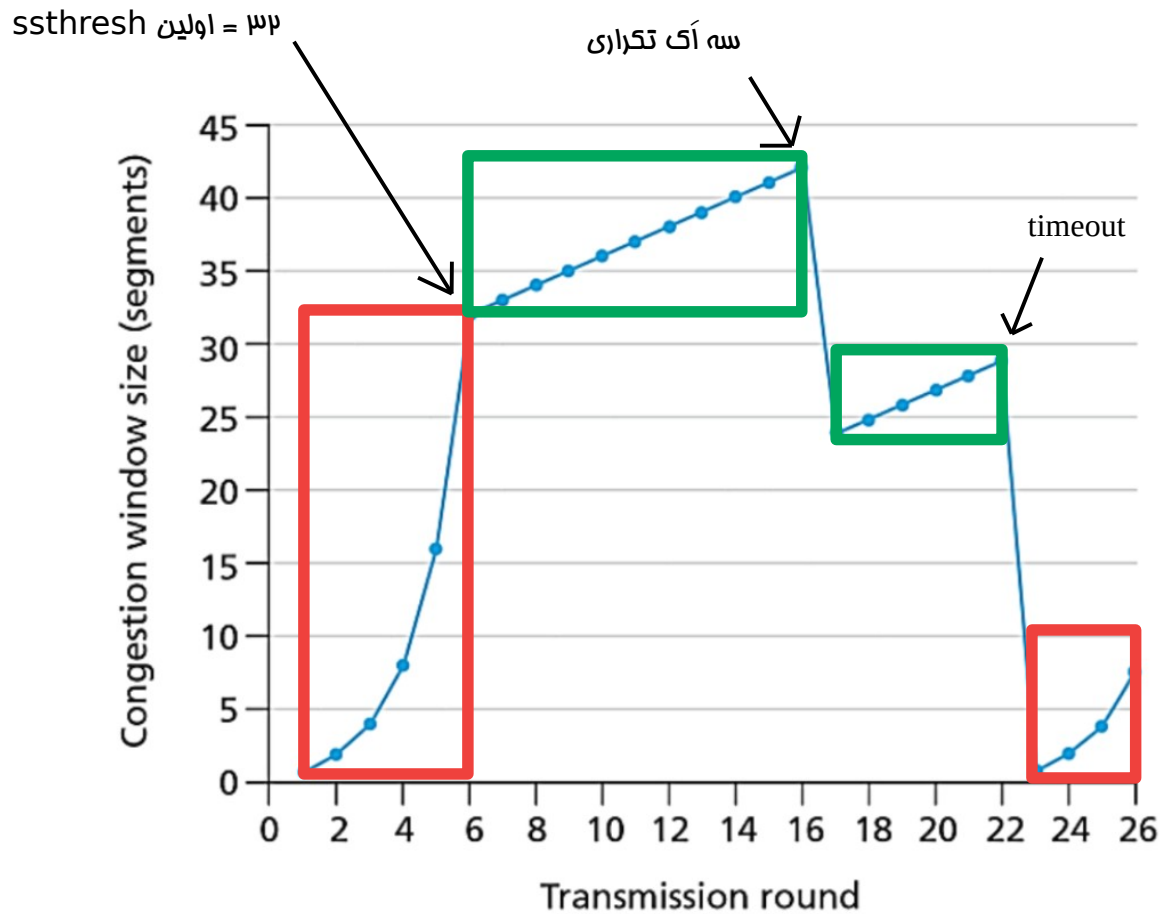
آک بسته های ۰ ، ۲ و ۴ گم شده اند. پنجره ی دریافت کننده بعد از دریافت بسته ها ، ۵ تا به جلو میرود



بعد از timeout شدن بسته های زوج، فرستنده مجدداً این بسته ها رو ارسال میکنه. چون آخرین بسته ی داخل پنجره ی فرستنده ، یک بسته زوج (بسته ی ۴) است پس پنجره ی فرستنده تغییر موضع نمیدهد. بعد از ارسال مجدد بسته های مذکور و دریافت آک آنها ، پنجره ی فرستنده ۵ تا جلو میره و مابقی بسته ها (بسته های ۵ تا ۸) رو ارسال میکنه. مجدداً هنگام برگشت آک های بسته های زوج (بسته های ۶ و ۸) سری دوم ، این آک ها گم میشوند. چون اولین بسته ی داخل پنجره ی فرستنده ، بسته ی ۵ است و آک این بسته به درستی برمیگردد ، پس پنجره ی فرستنده یکی به جلو میرود همچنین پنجره ی گیرنده به صورت کامل جلو میرود چون بسته ها رو به صورت کامل دریافت کرده.



بعد از timeout شدن بسته های ۶ و ۸ ، مجدداً این دو بسته ارسال میشوند. در انتها هم آک این دو بسته برمیگردد.



در دو نامیه ی مستطیل قرمز رنگ مشخص شده، در موقعیت «شروع کند» یا slow start هستیم. یعنی بازه های [۱ ، ۶] و [۲۳ ، ۲۶]

در هیچ بازه ی زمانی در این نمودار در موضع Fast Recovery نیستیم. فقط در دور ۱۷ وارد این موضع شده و با دریافت آک جدید مجدداً به موضع CA بازگشته ایم. در دو نامیه ی مستطیل سبز رنگ مشخص شده (یعنی بازه های [۶ ، ۱۶] و [۱۷ ، ۲۲]) در موقعیت «ممانعت از تراکم» یا congestion avoidance هستیم.

قسمت (ج)

طبیعتاً در نظر گرفته نشده !

قسمت (د)

کافیست محدوده ی فطی دوم نمودار یعنی بازه ی [۲۲ ، ۱۷] را به شکل نمایی رسم کنیم تا نشان دهیم که در موضع FR هستیم.

قسمت (ه)

سه اک تکراری

قسمت (و)

timeout رخ داده

قسمت (ز)

MSS ۳۲

قسمت (ح)

MSS ۲۱ = [۴۲/۲]

قسمت (ط)

MSS ۱۴ = [۲۹/۲]

قسمت (ی)

پاسخ : ۷

دور	۱	۲	۳	۴	۵	۶	۷
cwnd	۱	۲	۴	۸	۱۶	۳۲	۳۳
کل قطعات	۱	۳	۷	۱۵	۳۱	۶۳	۹۶

بسته ی هفتم ارسال میشود

قسمت (ک)

در دور ۲۶ ام اندازه ی پنجره ۸ است حال اگر سه آک تکراری دریافت کند اندازه ی ssthresh برابر با نصف اندازه پنجره یعنی ۴ خواهد شد و طول پنجره هم میشود سه به علاوه ی ssthresh یعنی ۷

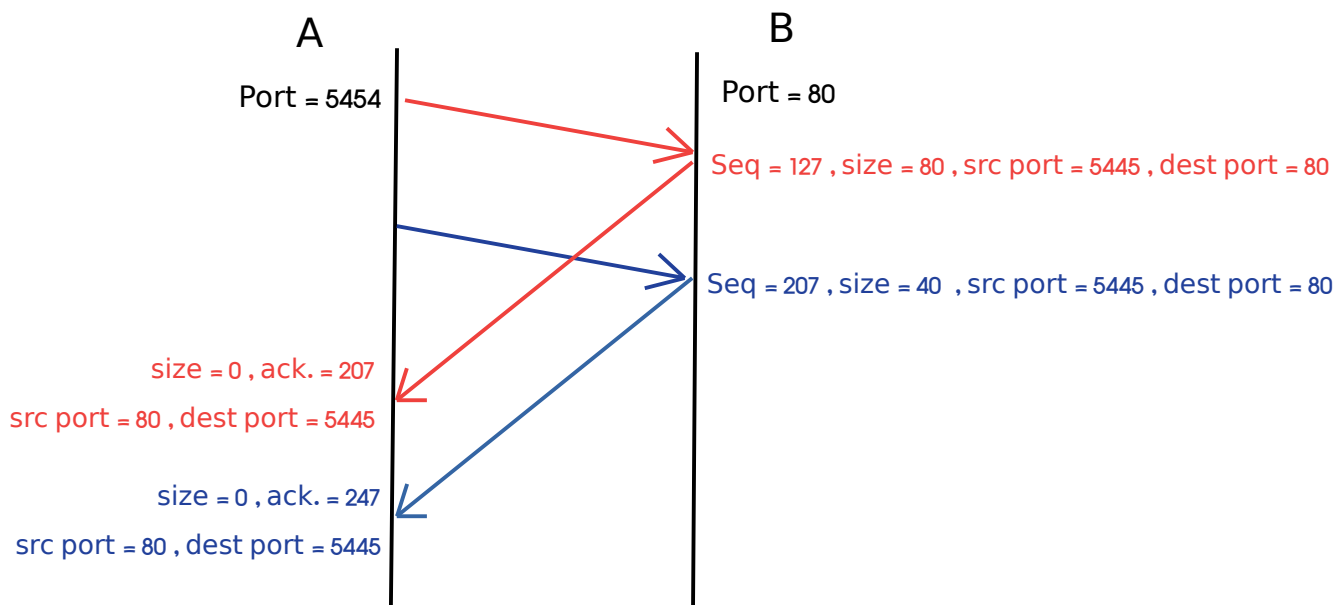
به صورت خلاصه : $ssthresh = ۴$ و $cwnd = ۷$

قسمت (الف)

seq. Number = 207 & source Port = 5445 & destination Port = 80

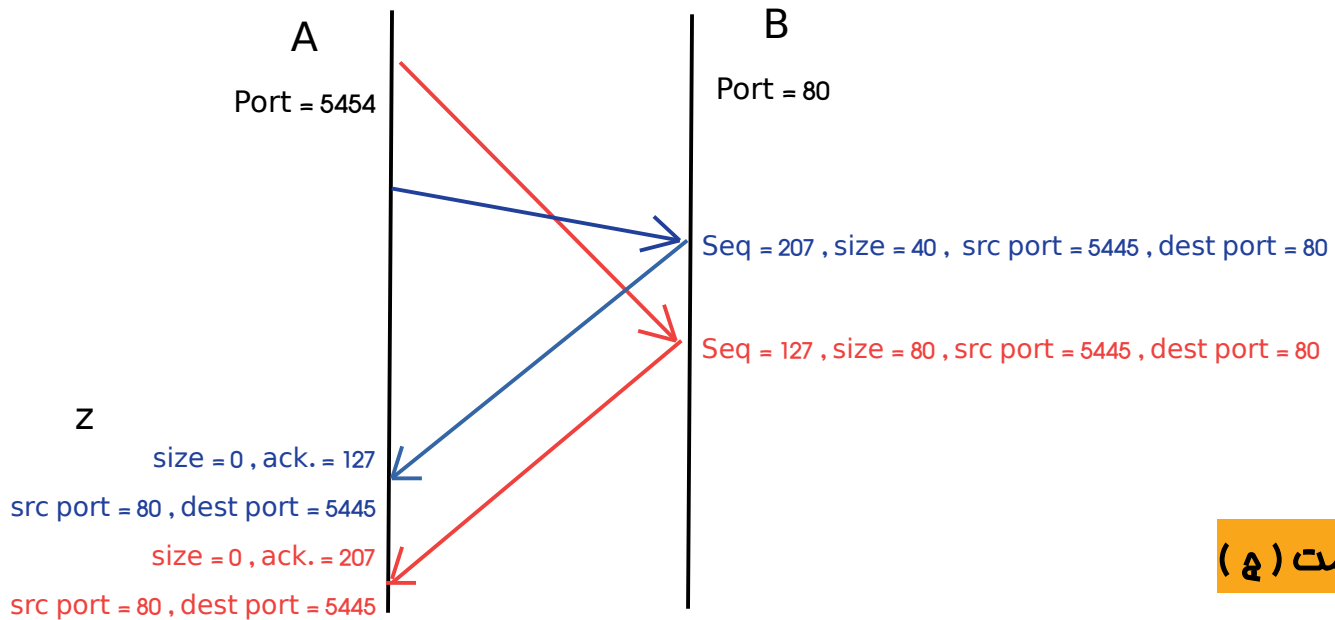
قسمت (ب)

ack. Number = 207 & source port = 80 & destination port = 5445

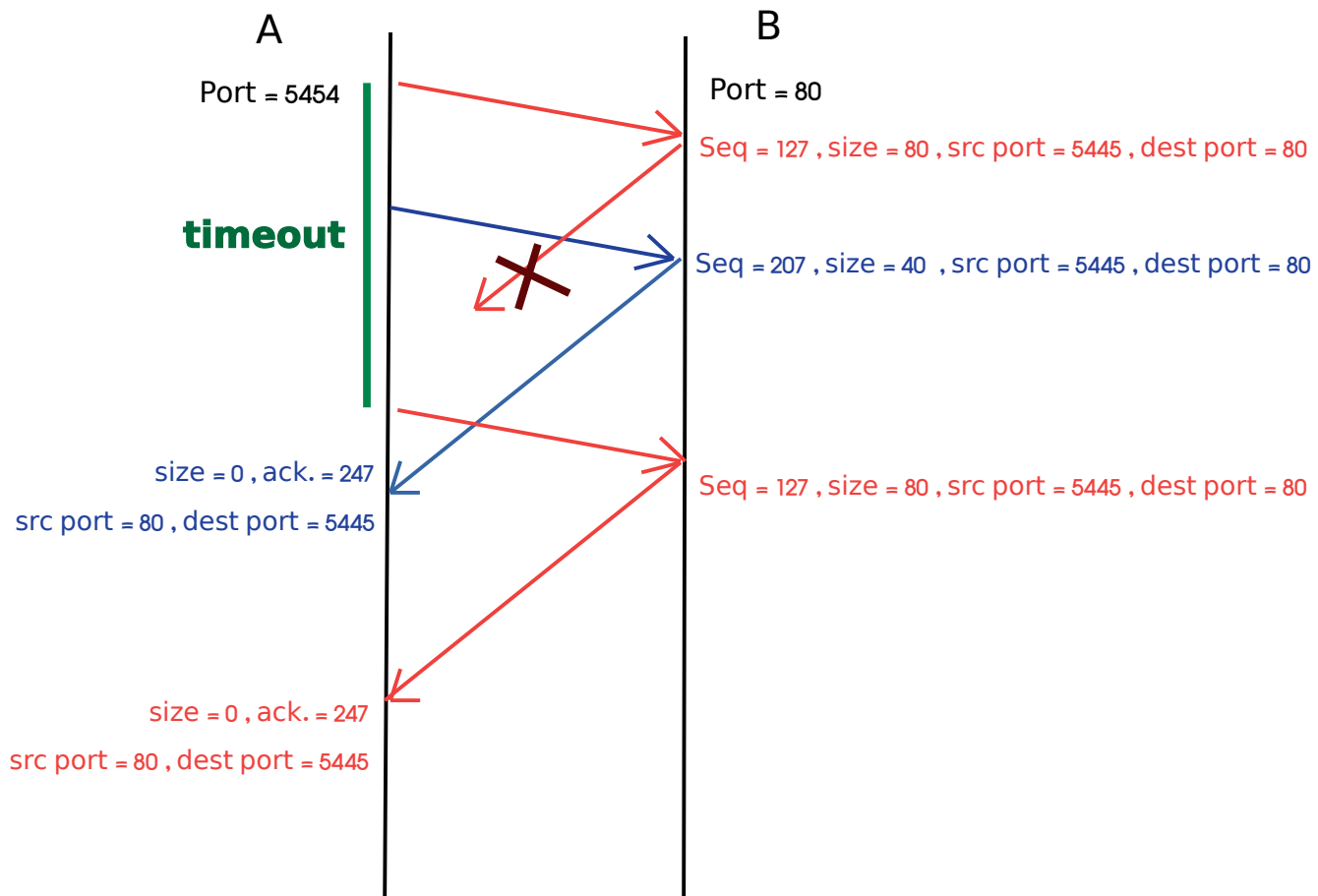


قسمت (ه)

ack. Number = 127 & source port = 80 & destination port = 5445



قسمت (ه)



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

$$estimatedRTT_1 = (1 - 0.125) \times 100 + 0.125 \times 115 = 101.875$$

$$estimatedRTT_2 = (1 - 0.125) \times 101.875 + 0.125 \times 90 = 100.390$$

$$estimatedRTT_3 = (1 - 0.125) \times 100.390 + 0.125 \times 140 = 105.341$$

$$estimatedRTT_4 = (1 - 0.125) \times 105.341 + 0.125 \times 120 = 107.174$$

$$estimatedRTT_5 = (1 - 0.125) \times 107.174 + 0.125 \times 106 = 107.027$$

.....

$$DevRTT = (1 - \beta) \times DevRTT + \beta \times (SampleRTT - EstimateddRTT)$$

$$DevRTT_1 = (1 - 0.25) \times 5 + 0.25 \times (115 - 101.875) = 7.03125$$

$$DevRTT_2 = (1 - 0.25) \times 7.03125 + 0.25 \times (100.39 - 90) = 7.8709375$$

$$DevRTT_3 = (1 - 0.25) \times 7.8709375 + 0.25 \times (140 - 105.341) = 14.567953125$$

$$DevRTT_4 = (1 - 0.25) \times 14.567953125 + 0.25 \times (120 - 107.174) = 14.1324648438$$

$$DevRTT_5 = (1 - 0.25) \times 14.1324648438 + 0.25 \times (107.027 - 106) = 10.8560986329$$