

| | |
|----------------------------------|--------------|
| پنج‌شنبه، 10 تیر 1400، 9:50 صبح | شروع |
| پایان یافته | وضعیت |
| پنج‌شنبه، 10 تیر 1400، 11:00 صبح | پایان |
| 1 ساعت 9 دقیقه | زمان صرف شده |
| 79.00 از 92.00 (85.87%) | نمره |

سؤال 1

کامل

نمره 10.00 از

10.00

با وجود اینکه مدل سرویسی که لایه شبکه در اینترنت ارائه می‌دهد از جنس best effort است و یا به عبارتی ضمانتی در مورد پهنای باند، تأخیر و ... نمی‌دهد، دلایل موفقیت نسبی اینترنت در سرویس دهی به کاربردهای حساس به تأخیر و پهنای باند را چه میدانید (حداقل ۲ دلیل بیان کنید)

<> برای این سوال پاسخ خود را باید در ویرایشگر زیر تایپ کنید.

سادگی پیاده سازی اینترنت (simplicity of mechanism)

همچنین provisioning of bandwidth: زیاد بودن پهنای باند لینک ها به اندازه ی کافی برای پوشش دادن مشکلات.

و همچنین: replicated , application layer distributed service

پراکندگی باعث میشود ازدحام روی یک گزینه زیاد نشود و چندین گزینه در دسترس یک هاست باشد.

و کنترل ازدحام برای سرویس های elastic

اینترنت باعث افزایش ارتباطات شده و به عبارت simple connectivity دارد.

داشتن یک پروتکل در لایه ی شبکه (ip) که مشترک است.

به دلیل هوشمندی است که در ساختار اینترنت پیاده سازی شده اینکه قسمت پیچیده را در لبه ی اینترنت به جای

core شبکه پیاده سازی کرده اند باعث بهینه تر شدن اینترنت شده.

۱- در ISP ها از لینکهای با پهنای باند زیاد استفاده می شود و به عبارتی منابع لازم برای دسترسی به سرعت های بالا و با تاخیر کم در هسته اینترنت موجود است. (over-provisioned bandwidth)

۲- استفاده از ایده پخش کردن سرویس در سطح اینترنت (duplicated servers) باعث نزدیک شدن محل سرورها به محل کلاینتها می شود و یا امکان سرویس از چند محل به یک کلاینت را فراهم می کند. این امر به کاهش زمان سرویس دهی کمک می کند.

۳- ریت ارسال سرویس های غیر حساس به تاخیر می تواند بر حسب شرایط ازدحام شبکه کم و یا زیاد شود. در نتیجه میتوان انتظار داشت که در شرایط ازدحام همچنان سرویس های حساس به زمان بتوانند منابع لازم را در اختیار داشته باشند. (congestion control of "elastic" services)

دیدگاه:

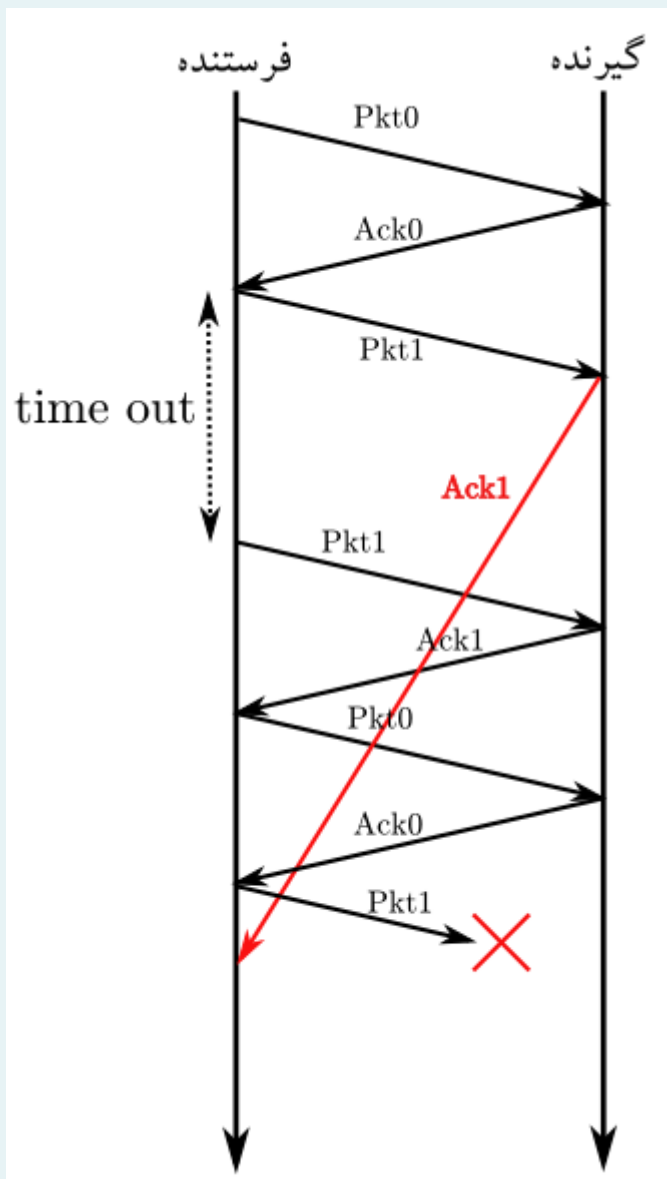
سؤال 2

کامل

نمره 22.00 از

35.00

پروتکل rdt3.0 را در نظر بگیرید. شکل زیر نمونه‌ای از عملکرد این پروتکل در مواجهه با یک کانال نامطمئن را نشان می‌دهد.



الف) با توجه به شکل اثرات مخربی که توسط کانال تاکنون اتفاق افتاده است شامل چه مواردی است؟

ب) اگر در ادامه کانال به صورت ایده‌آل عمل کند، چه بسته‌هایی از سمت فرستنده و گیرنده برای یکدیگر ارسال می‌شوند؟ (برای پاسخ به این سوال می‌بایست یک تصویر که بیانگر ادامه شکل فوق می‌باشد در پیوست این سوال قرار دهید به نحوی که ۲ بسته‌ای که در ادامه هر یک از دستگاه‌های فرستنده و گیرنده ارسال می‌کنند مشخص باشد)

ج) در ادامه قسمت ب، آیا پروتکل rdt3 توانسته است به درستی از پس اثرات مخربی که توسط کانال ایجاد شده است برآید؟ پاسخ خود را توضیح دهید و در صورت منفی بودن جواب جزئیات مشکلی که با وجود استفاده از rdt3.0 بوجود می‌آید را بیان کنید.

د) در صورت منفی بودن جواب به قسمت «ج» تمام راهکارهایی (اصلاحاتی که در پروتکل باید اعمال شود) که می‌توانید پیشنهاد کنید تا مشکل برطرف شود را بنویسید.

<< برای قسمت‌های «الف»، «ج» و «د» پاسخ خود را باید در ویرایشگر زیر تایپ کنید. برای قسمت «ب» عکس تصویر درخواستی را پیوست نمایید.

الف: یک پکت دوم (pkt1) پس از اتمام timeout فرستاده شده و همچنین فرستنده مجدد بسته را با timeout شدن ارسال کرده. بعد با دریافت اک برای فرستادن دوباره، بسته‌های بعدی را می‌فرستد و منتظر می‌شود تا اک یک برای بسته آخر دریافت شود.

فرستنده اک پکت دوم را با تاخیر دریافت میکند و به اشتباه فرض میکند برای بسته ی آخر است (ولی در واقع برای پکت دوم بوده)

اک آخری که دریافت شده برای بسته ی آخری نیست ول

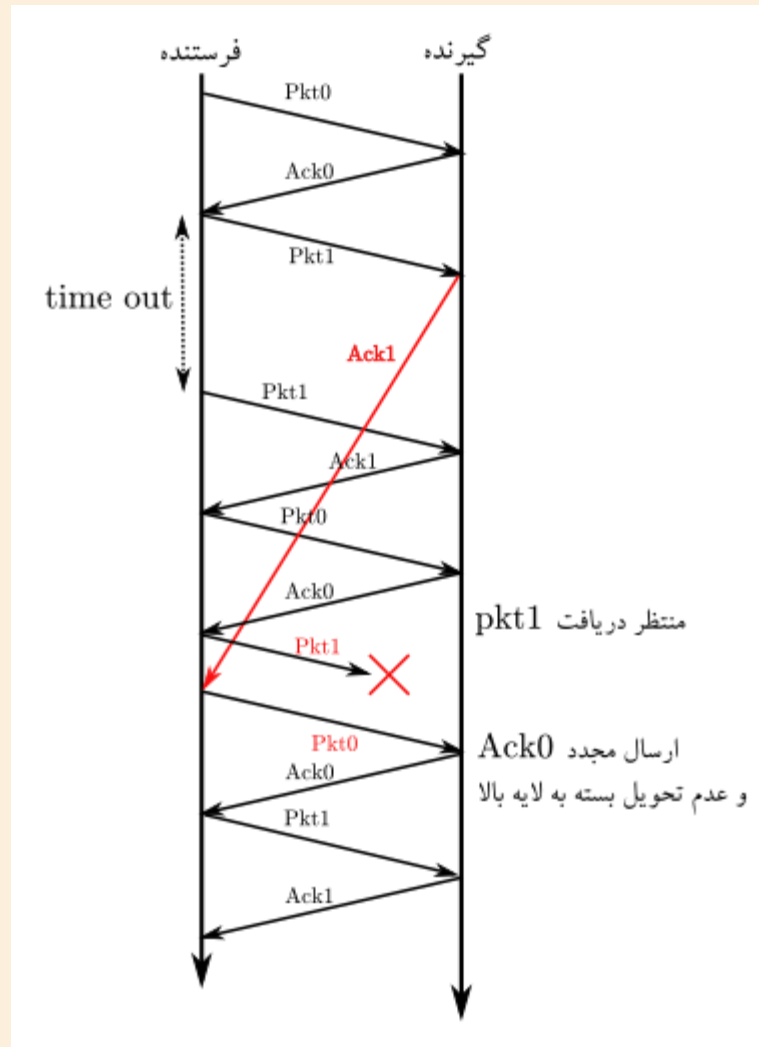
ج) خیر نتوانسته چون پکت آخری که ارسال شده گم شده در مسیر و به دست گیرنده نرسیده و اک یک دریافتی بسته دوم هم که با تاخیر گرفته شده برای بسته آخر در نظر گرفته و همچنین هیچ ارسال مجددی برای بسته آخر که گم شده نداریم.

و همچنین بدون اینکه آخرین بسته به گیرنده برسد بسته ی صفرم ارسال میشود.

د) راهکارهایی که میتواند بهبود دهد: داشتن یک پنجره ارسال با سایز معین و افزایش طول بازه برای شماره گذاری

photo_2021-07-01_10-52-59.jpg 

(ب)



ج) پروتکل rdt3.0 در از بین بردن اثرات مخرب کانال موفق نبوده است. زیرا پس از گم شدن بسته pkt1 و دریافت Ack1 تاخیر یافته، فرستنده بسته جدید با شماره صفر ارسال می کند. در این شرایط چون گیرنده منتظر بسته با شماره ۱ است، تصور می کند که فرستنده بسته قبلی را مجدد ارسال کرده و به همین دلیل یک Ack0 ارسال می کند و بسته جدید را دور میریزد!

در ادامه فرستنده با دریافت Ack0 فرض می کند که بسته اخیر با شماره ۰ به درستی به گیرنده رسیده و بسته جدید با شماره ۱ را برای گیرنده ارسال میکند. بنابراین طبق این توصیف دو بسته که با رنگ قرمز در شکل قسمت «ب» نشان داده

شده اند نهایتاً به دست گیرنده نمی رسند و این موضوع از دید فرستنده و گیرنده مخفی می ماند.
(د) دو راه حل وجود دارد: ۱- افزایش زمان time-out به نحوی که تاخیرهای زیاد کانال را همچنان پوشش دهد. ۲- استفاده از رنج شماره گذاری بزرگتر

دیدگاه:

| نمره گرفته شده | نمره هر قسمت | سؤال ۲ |
|----------------|--------------|--------|
| 2 | 5 | الف |
| 10 | 10 | ب |
| 5 | 10 | ج |
| 5 | 10 | د |
| 22 | 35 | جمع |

سؤال 3

کامل

نمره 32.00 از

32.00

جدول ارسال (forwarding table) یک روتر در یک شبکه دیتاگرام به صورت نمایش داده شده در زیر است. فرض کنید در این شبکه آدرس‌های لایه شبکه ۸ بیتی هستند و روتر از مکانیزم «تطابق بلندترین پیشوند» (longest prefix match) برای تعیین پورت خروجی بسته‌های دریافتی استفاده می‌کند. در این صورت مشخص کنید، از مجموع ۲۵۶ آدرس ممکن، متناظر با هر پورت خروجی چه تعداد آدرس وجود دارد؟ (به عبارت دیگر متناظر با هر پورت خروجی چند آدرس وجود دارد که اگر بسته‌ای یکی از آن آدرس‌ها را به عنوان آدرس مقصد داشته باشد برای ارسال شدن، تحویل آن پورت خروجی داده می‌شود). نوشتن دلیل و توضیح برای عددی که متناظر با هر پورت بیان می‌کنید الزامی است.

<< برای این سوال پاسخ خود را باید در ویرایشگر تایپ کنید.

| پیشوند | شماره پورت خروجی |
|-----------------|------------------|
| 1 | 0 |
| 10 | 1 |
| 001 | 2 |
| 101 | 3 |
| در غیر این صورت | 4 |

طبق مکانیزم بلندترین پیشوند ۱: بسته‌هایی که با بیت‌های ۱۰۱ شروع میشوند به پورت ۳ میروند که تعداد: ۲ به توان (

$$۳۲ = (۸ - ۳)$$

۲: بسته‌هایی که با بیت‌های ۱۱ شروع میشوند به پورت ۰ میروند زیرا ۱۰ به پورت یک و ۳ رفته تعداد: ۲ به توان (۸-۲) = ۶۴ میشود.

۳: بسته‌هایی که با بیت‌های ۱۰۰ شروع میشوند به پورت ۱ میروند (زیرا ۱۰۱ به پورت ۳ رفته) و تعداد آنها: ۲ به توان (۳ -

$$۳۲ = (۸ - ۳)$$

۴: بسته‌هایی که با بیت‌های ۰۰۱ شروع میشوند به پورت ۲ میروند. تعداد: ۲ به توان (۸ - ۳) = ۳۲

تعداد بسته هایی که به پورت ۴ میروند مساوی کل ادرس هایی که میتواند وجود داشته باشد منهای جمع ادرس های بالا:

$$32 \times 3 + 64 = 160$$

$$256 - 160 = 96$$

طبق قانون تطابق بلندترین پیشوند، پیشوند های ۰۰۱ و ۱۰۱ بالاترین اولویت را دارند و هر کدام با ۳۲ آدرس لایه شبکه تطبیق پیدا می کنند. بنابراین برای هر یک از پورت های خروجی ۲ و ۳ تعداد آدرس ها ۳۲ می شود.

برای پورت خروجی ۱، تعداد آدرس های شبکه که به صورت اولیه با پیشوند متناظر این پورت یعنی ۱۰ تطابق پیدا می کنند برابر با ۶۴ آدرس است. اما از آنجایی که این پیشوند، خود برای پیشوند ۱۰۱ (متناظر با پورت ۳) نیز پیشوند است، ۳۲ آدرس از ۶۴ آدرس هم با ۱۰ تطبیق پیدا می کنند و هم با ۱۰۱. از آنجایی که از قانون «تطابق بلندترین پیشوند» استفاده می شود، نهایتاً ۳۲ آدرس لایه شبکه هستند که از پورت ۱ ارسال می شوند.

برای پورت ۰، پیشوند ۱ است که با پیشوندهای ۱۰ و ۱۰۱ هم پوشانی دارد و در نتیجه تعداد آدرس های لایه شبکه که نهایتاً از این پورت ارسال می شوند برابر است با ۳۲-۳۲-۱۲۸=۶۴

در نهایت مابقی آدرس های IP برای پورت ۴ ارسال می شوند ۶۴-۳۲-۳۲-۳۲=۹۶

خلاصه بحث فوق:

پورت ۰: ۶۴

پورت ۱: ۳۲

پورت ۲: ۳۲

پورت ۳: ۳۲

پورت ۴: ۹۶

دیدگاه:

سؤال 4

کامل

نمره 15.00 از

15.00

دو نود x و y را در نظر بگیرید که قصد دارند از طریق لینکی به صورت مشترک با استفاده از پروتکل Slotted Aloha داده‌های خود را ارسال کنند. اگر هر دو نود به اندازه کافی داده برای ارسال داشته باشند و احتمال باز-ارسال نود x برابر با 0.2 باشد، احتمال باز-ارسال نود y باید چقدر باشد تا میانگین دفعاتی که نود y موفق به استفاده از کانال می‌شود 6 برابر دفعاتی باشد که نود x به صورت میانگین موفق به استفاده از کانال می‌شود؟

پاسخ خود را کامل توضیح دهید.

<< برای این سوال می‌توانید پاسخ خود را در وایشگر تایپ کنید و یا عکس راه حل خود را در پیوست قرار دهید.

$p_x = 0.2$ است.

دنبال p_y هستیم. طبق روات پایین جواب پیدا میشود:

[۴.jpg](#) 

احتمال موفقیت نود x برابر است با

$$p_x(1 - p_y)$$

احتمال موفقیت نود x برابر است با

$$p_y(1 - p_x)$$

بنابراین می توان نوشت:

$$p_y(1 - 0.2) = 6 \times 0.2(1 - p_y) \rightarrow p_y = 0.6$$

دیدگاه: