



دانشگاه صنعتی امیرکبیر  
و فناوری اطلاعات



بسته‌های

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)  
دانشگاه مهندسی کامپیوتر و فناوری اطلاعات



دانشگاه مهندسی کامپیوتر و فناوری اطلاعات  
دانشگاه صنعتی امیرکبیر

۱۳۹۷-۱۳۹۷



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

مسئله	نمره
۹	
۱۰	

مسئله	نمره
۱	
۲	
۳	
۴	
۵	
۶	
۷	
۸	

درس شبکه‌های کامپیوتری، نیم سال اول سال تحصیلی ۹۷-۹۸

تمرین سری چهارم (موعد تحویل: ۱۳۹۷/۰۸/۰۶)

توجه: پاسخ تمرین‌ها باید به صورت دست‌نویس تحویل داده شود.

نام و نام خانوادگی:

شماره دانشجویی:

نمره:

توجه: برای صرفه‌جویی در کاغذ تکالیف را یا دو رو پرینت بگیرید و یا از کاغذهای باطله یک رو سفید استفاده کنید.

**سوال ۱:** هر سه پروتکل IP، TCP و UDP در صورتی که بسته‌ای با خطا در checksum دریافت کنند بدون اطلاع به فرستنده آن را دور می‌ریزند. دلیل اینکار را توضیح دهید.

پروتکل‌های IP و UDP پروتکل‌های قابل اعتمادی نیستند، بنابراین در صورت از دست رفتن بسته‌ها به کاربر اطلاعی نمی‌دهند. در مورد TCP که یک پروتکل قابل اعتماد می‌باشد، ممکن است بسته‌ها از دست بروند اما با استفاده از مکانیزم Timeout دوباره برای آن‌ها تقاضا داده می‌شود. بنابراین محتوای این بسته‌ها از دست نخواهد رفت.

**سوال ۲:** اگر یک رسانه‌ی فیزیکی به صورت قابل اطمینان رفتار کند، آیا هنوز نیاز به لایه‌ی پیوند داده وجود خواهد داشت؟

بله، لایه‌ی پیوند داده در کنار وظیفه‌ی فراهم آوردن قابلیت اطمینان بر بستر فیزیکی وظایف دیگری را نیز بر عهده دارد، مثلاً لایه‌ی پیوند داده framing را انجام می‌دهد یا دسترسی همزمان به بستر فیزیکی را مدیریت می‌کند و ... (ذکر یک مثال از سایر وظایف لایه‌ی پیوند داده لازم می‌باشد).

**سوال ۳:** فرض کنید می‌خواهید یک تراکنش بین مشتری و سرویس‌دهنده که از یکدیگر دور هستند را با بیشترین سرعت ممکن انجام دهید. از UDP استفاده می‌کنید یا TCP؟ پاسخ خود را توضیح دهید.

استفاده از UDP سرعت ارتباط را افزایش می‌دهد، زیرا سربارهای زمانی برای شروع و خاتمه دادن ارتباط را نداشته و کنترلی بر روی جریان ندارد، از این رو استفاده از UDP در ارتباط‌های مالتی‌مدیا به صرفه می‌باشد. البته باید به این نکته دقت داشت که همیشه این افزایش سرعت اتفاق نمی‌افتد مثلاً اگر شبکه بسیار شلوغ باشد ممکن است تعداد زیادی از بسته‌ها با ارسال کورکورانه UDP از بین بروند. (فهم درست تفاوت‌های این دو پروتکل و استنتاج بر اساس این تفاوت‌ها لازم می‌باشد).

توجه داشته باشید که دور بودن سرویس‌گیرنده و سرویس‌دهنده باعث افزایش تاخیر در ارتباط آن‌ها می‌شود و روند ساخت و از بین بردن ارتباط را برای ما هزینه‌بر می‌سازد.



**سوال ۴:** فرض کنید در مرورگر خود بر روی یک لینک برای باز کردن یک صفحه کلیک می‌کنید، آدرس IP مربوط به URL در ماشین شما کش (cache) نشده است پس برای به دست آوردن آدرس IP نیاز به DNS lookup هست. فرض کنید برای به دست آوردن آدرس IP، نیاز به مراجعه به n سرویس-دهنده DNS به صورت پشت سرهم است. فرض کنید زمان لازم برای ارسال و دریافت هر Query به هر سرویس-دهنده DNS را با  $RTT_1$  تا  $RTT_n$  نمایش می‌دهیم. بعلاوه فرض کنید صفحه وب مرتبط با آن لینک یک شی (شامل یک متن کوچک HTML) دارد. فرض کنید  $RTT_0$  به عنوان زمان رفت و برگشت بین ماشین شما و سرویس‌دهنده شامل این شی در نظر گرفته می‌شود. زمان ارسال شی را صفر در نظر بگیرید. از زمانی که شما بر روی این لینک کلیک می‌کنید تا زمانی که شی را دریافت می‌کند، چه زمانی سپری می‌شود؟

کل زمان بدست آوردن آدرس IP:

$$RTT_1 + RTT_2 + \dots + RTT_n$$

زمانی که آدرس IP دریافت شد، به اندازه‌ی  $RTT_0$  برای ایجاد ارتباط TCP زمان می‌خواهیم و در نهایت با زمان  $RTT_0$  یک تقاضا ارسال شده و پاسخ آن دریافت می‌شود بنابراین:

$$2RTT_0 + RTT_1 + RTT_2 + \dots + RTT_n$$

زمان کل این پروسه خواهد بود توجه داشته باشید در اینجا از زمان خاتمه‌ی ارتباط (پیام‌های FIN و ..) صرف نظر شده است.

**سوال ۵:** در سوال ۴ فرض کنید که به سه سرویس-دهنده DNS مراجعه صورت گرفته است و فایل HTML شامل پنج شی بسیار کوچک بر روی همان سرویس‌دهنده است. با صرف نظر کردن از زمان ارسال اشیاء، در هر یک از حالات زیر محاسبه کنید از زمانی که شما بر روی این لینک کلیک می‌کنید تا زمان دریافت کامل صفحه وب، چه زمانی سپری می‌شود؟ در هر مورد دیگرام زمانی تبادل پیام‌ها بین سرویس-گیرنده و سرویس-دهنده را رسم کنید.

- HTTP نا پایا (Non-Persistent) بدون هیچ اتصال موازی TCP
- HTTP نا پایا با ۵ اتصال موازی
- HTTP پایا (Persistent)

- HTTP نا پایا (Non-Persistent) بدون هیچ اتصال موازی TCP

$$RTT_1 + RTT_2 + RTT_3 + 2RTT_0 + 10RTT_0 = 12RTT_0 + RTT_1 + RTT_2 + RTT_3$$

- HTTP نا پایا با ۵ اتصال موازی

$$RTT_1 + RTT_2 + RTT_3 + 2RTT_0 + 2RTT_0 = 4RTT_0 + RTT_1 + RTT_2 + RTT_3$$

- HTTP پایا (Persistent) در حالت پایپ لاین

$$RTT_1 + RTT_2 + RTT_3 + 2RTT_0 + RTT_0 = 3RTT_0 + RTT_1 + RTT_2 + RTT_3$$

- HTTP پایا (Persistent) در حالت غیر پایپ لاین

$$RTT_1 + RTT_2 + RTT_3 + 2RTT_0 + 5RTT_0 = 7RTT_0 + RTT_1 + RTT_2 + RTT_3$$



**سوال ۶:** یک لینک ارتباطی به طول ۱۰ متر را در نظر بگیرید یک فرستنده از طریق آن قادر است با نرخ ۱۵۰ bits/sec در دو جهت ارسال کند. فرض کنید که بسته های شامل داده ۱۰۰۰۰۰ بیت و بسته های شامل درخواست شی و سه مرحله دست تکانی TCP، ۲۰۰ بیت هستند. حال پروتکل HTTP را در نظر بگیرید و فرض کنید که اندازه هر شی ۱۰۰۰۰۰ بیت است و اولین شی دانلود شده به ۱۰ شی دیگر بر روی همان فرستنده ارجاع می کند. فرض کنید از HTTP نا پایا استفاده می کنید و بعد از دریافت اولین شی، ۱۰ اتصال موازی برقرار می کنید. با داشتن N اتصال موازی، هر یک از اتصالات TCP نرخ انتقالی برابر 1/N پهنای باند را دارند. چه زمانی طول می کشد تا همه اشیاء دریافت شوند؟ حال HTTP پایا را در نظر بگیرید آیا انتظار کارایی بیشتری نسبت به مورد اتصال نا پایا دارید؟ پاسخ خود را تشریح کنید. از تاخیر صف و پردازش صرف نظر کنید و صرفا تاخیر انتشار و زمان ارسال بسته ها را در نظر بگیرید.

زمان ارسال (Transmission) بسته کنترلی:

$$t_c = \frac{200}{150} = 1.3s$$

زمان ارسال (Transmission) بسته دیتا:

$$t_d = \frac{100000}{150} = 666.7s$$

زمان هر رفت و برگشت را با یک RTT نمایش می دهیم که برابر با  $2T_{prop}$  می باشد.

زمان دریافت صفحه Base:

$$t_c + t_c + RTT + t_c + t_d + RTT = 1.3 + 1.3 + RTT + t_d + t_c + RTT = 670.6 + 4T_{prop}$$

زمان دریافت بقیه Object ها:

Non-Persistent with 10 Parallel Connections:

زمان ارسال بسته کنترلی (در حالتی که ۱۰ ارتباط موازی وجود دارد، پهنای باند میان این ارتباط ها تقسیم می شود):

$$t'_c = \frac{200}{\left(\frac{150}{10}\right)} = 10t_c = 13s$$

زمان ارسال بسته دیتا (در حالتی که ۱۰ ارتباط موازی وجود دارد، پهنای باند میان این ارتباط ها تقسیم می شود):

$$t'_d = \frac{100000}{\left(\frac{150}{10}\right)} = 10t_d = 6666.67$$

زمان دریافت هر Object:

$$t'_c + t'_c + RTT + t'_c + t'_d + RTT = \frac{200}{\left(\frac{150}{10}\right)} + \frac{200}{\left(\frac{150}{10}\right)} + \frac{200}{\left(\frac{150}{10}\right)} + \frac{100000}{\left(\frac{150}{10}\right)} + 4T_{prop}$$

$$= 3 * 13.33 + 6666.67 + 4T_{prop} = 6706.7 + 4T_{prop}$$

از آنجایی ارتباط ها به صورت موازی شکل می گیرند، زمان کل شامل زمان دریافت Base و همه ی Object ها برابر خواهد بود با:

$$6706.7 + 4T_{prop} + 670.6 + 4T_{prop} = 7377.3 + 8T_{prop}$$

Persistent:

در حالت غیر پایپ لاین، برای هر Object تقاضا داده می شود، در ادامه پس از دریافت آن Object، تقاضا بعدی صورت می گیرد و به همین ترتیب:

$$10 \times (t_c + t_d + RTT) = 10t_c + 10t_d + 10RTT = 6679.7 + 20T_{prop}$$

زمان کل شامل زمان دریافت Base و همه ی Object ها برابر خواهد بود با:



$$10t_c + 10t_d + 20T_{prop} + t_c + t_c + 2T_{prop} + t_c + t_d + 2T_{prop} = 6679.7 + 20T_{prop} + 670.6 + 4T_{prop}$$

$$= 7350.2 + 24T_{prop}$$

در حالت پایپ لاین، برای هر Object به صورت موازی تقاضا داده می شود:

$$t_c + RTT + 10t_d = 1.3 + 666.7 + 2T_{prop} = 668 + T_{prop}$$

زمان کل شامل دریافت Base و همه ی Object ها برابر خواهد بود با:

$$t_c + t_c + 2T_{prop} + t_c + t_d + 2T_{prop} + t_c + RTT + 10t_d$$

سوال ۷: همانطور که می دانید می توانیم برای کاهش میزان ترافیک مصرفی HTTP در یک شبکه از cache استفاده کنیم. در HTTP چه option هایی برای زمانی که یک محتوا در حافظه ذخیره می شود وجود دارند؟

در ادامه این Option ها برای یک Request بررسی می شوند. دقت کنید سناریو به این ترتیب است که Option ها در تقاضایی که یک Client به یک سرور HTTP Proxy ارسال می کند مطرح می شوند. HTTP Proxy می تواند تقاضاهای شما Cache کرده و در آینده به جای برقراری ارتباط با سرور اصلی (Origin Server) از Cache خود برای ارسال پاسخ استفاده نماید. تمامی این Option ها در قالب Header ای با نام Cache-Control در تقاضای HTTP اضافه می شوند.

- **max-age=<seconds>**

Specifies the maximum amount of time a resource will be considered fresh.

- **max-stale[=<seconds>]**

Indicates that the client is willing to accept a response that has exceeded its expiration time.

- **min-fresh=<seconds>**

Indicates that the client wants a response that will still be fresh for at least the specified number of seconds.

- **no-cache**

Forces caches to submit the request to the origin server for validation before releasing a cached copy.

- **no-store**

The cache should not store anything about the client request or server response.

- **no-transform**

No transformations or conversions should be made to the resource. The Content-Encoding, Content-Range, Content-Type headers must not be modified by a proxy.

- **only-if-cached**

Indicates to not retrieve new data. This being the case, the server wishes the client to obtain a response only once and then cache.



**سوال ۸:** آیا برای یک ISP معقول است مانند یک CDN عمل کند؟ یعنی آیا ممکن است ISP بخواهد تعدادی از محتواهای یک CDN را بر روی شبکه خود ذخیره کند؟ این کار چه مزایا و معایبی برای آن دارد.  
بله، برای ISP این کار هزینه‌های زیر را می‌پردازد:

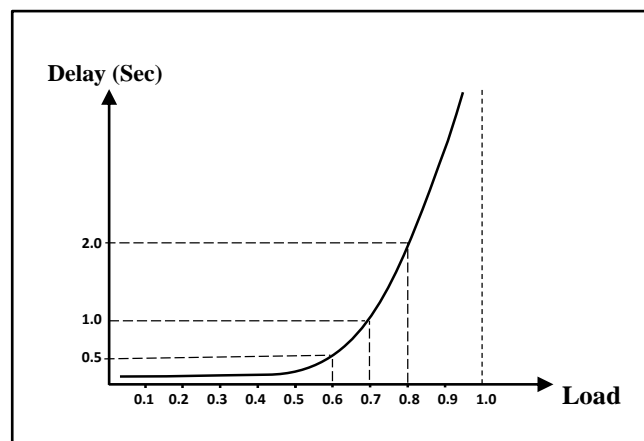
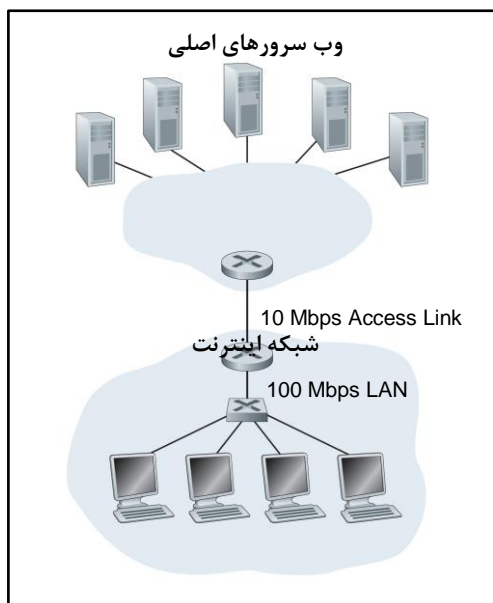
- هزینه‌ی حقوقی: ISP می‌بایست مالکیت معنوی محتوایی که می‌خواهد منتشر کند را بدست آورد.
- هزینه‌ی فنی: ISP می‌بایست زیرساخت مناسب جهت جمع‌آوری این محتوا را فراهم آورد.
- هزینه‌ی فنی: ISP می‌بایست برای به روزرسانی این محتوا زیرساخت مناسب را فراهم آورد.

ISP با نگهداری این اطلاعات به صورت محلی هزینه‌ی انتقال ترافیک به شبکه‌های دیگر را نمی‌پردازد.

**سوال ۹:** فرض کنید یک شخص به‌طور هم‌زمان همه سرویس‌دهنده‌های DNS اینترنت را از کار می‌اندازد این اتفاق چگونه تاثیری در استفاده اینترنت توسط یک کاربر عادی خواهد داشت؟

در این صورت قادر به انجام عمل نگاشت بین دامنه و آدرس IP نخواهیم بود. در نتیجه تنها راه دسترسی به صفحات وب از طریق آدرس IP به جای نام دامنه خواهد بود. در حالی که اکثر ما آدرس IP سروری که به آن دسترسی داریم را نمی‌دانیم بنابراین این نوع موقعیت باعث ناکارآمدی شدید اینترنت می‌گردد. زمانی که سرویس‌های DNS فعال نباشند، جابجایی سرورهایی زمانی که باعث تغییر آدرس IP آن‌ها شود دیگر به سادگی امکان‌پذیر نمی‌باشد.

**سوال ۱۰:** به منظور Web Caching از یک Proxy Server در شبکه محلی سازمان استفاده شده است. شبکه محلی از طریق یک مسیریاب (روتر) با یک لینک ۱۰ Mbps به بیرون متصل است. سرعت خط داخلی ۱۰۰ Mbps است. اگر اندازه پیام‌های Request ناچیز، اندازه پیام‌های Response، ۴۰۰ Kbits و به طور متوسط ۳۰ درخواست برای objectهای وب در هر ثانیه وجود داشته باشد، با فرض آنکه با قرار دادن Proxy، ۵۰٪ درخواست‌ها از طریق Proxy سرویس داده می‌شوند مطلوب است تاخیر متوسط دریافت objectهای وب اگر تاخیر وب سرورهای اصلی تا مسیریاب (تاخیر اینترنت) ۲ ثانیه و تاخیر متوسط مسیریاب‌ها براساس منحنی زیر داده شده باشد.



پروکسی سرور می‌تواند مطابق شکل به سوییچ یا به مسیریاب متصل شود. اگر به سوییچ متصل شود، تاخیر LAN برابر صفر خواهد بود. اگر به مسیریاب متصل شود، بار بر روی لینک ۱۰۰ مگابیتی به صورت زیر محاسبه می‌شود:



$$\frac{30 * 400 * 10^3}{100 * 10^6} = \frac{12}{100}$$

که مطابق نمودار می توان از تاخیر آن چشم پوشی کرد.

۵۰ درصد درخواست ها نیاز به دانلود اطلاعات از وب سرورهای اصلی دارند که بار بر روی لینک ۱۰ مگابیتی به صورت زیر محاسبه می شود:

$$0.5 * \frac{30 * 400 * 10^3}{10 * 10^6} = 0.6$$

که مطابق نمودار تاخیری برابر ۰.۵ ثانیه دارد. تاخیر اینترنت برابر با ۲ ثانیه است و مجموع این دو تاخیر برای این درخواست ها برابر: ۲.۵ است. بنابراین متوسط تاخیر دریافت objectهای وب برابر است با:

$$0.5 * 0 + 0.5 * (2 + 0.5) = 1.25$$