



نام و نام خانوادگی: علی خرمی‌پور

شماره دانشجویی: ۹۶۳۱۴۰۷

1) از بین پرتکل‌های TCP و UDP برای هر کدام از مصارف زیر چه پرتکلی سرویس بهتری ارائه می‌دهد؟ چرا؟

- a. مکالمه زنده: UDP – بسته‌ها باید هر چه سریع‌تر به مقصد برسند
- b. بازدید از صفحات وب: TCP – انتقال داده قابل اطمینان لازم است
- c. دریافت فایل‌های حجیم: TCP – انتقال داده قابل اطمینان، کنترل جریان و کنترل ازدحام لازم است

2) چهار طبقه اساسی از خدماتی را که پرتکل انتقال می‌تواند ارائه دهد را فهرست کنید. برای هریک از سرویسها، بنویسید که کدام یک از پرتکل‌های TCP یا UDP و یا هر دو این سرویس را ارائه می‌دهد.

- Multiplexing/Demultiplexing: هر دو
- انتقال داده قابل اطمینان (Reliable Data Transfer): TCP
- کنترل جریان (Flow Control): TCP
- کنترل ازدحام (Congestion Control): TCP

3) توضیح دهید چگونه Caching در وب می‌تواند تاخیر دریافت یک درخواست را کاهش دهد. آیا Caching در وب تاخیر برای تمام سندهای درخواست شده توسط کاربر را کاهش می‌دهد یا فقط تاخیر برخی از سندها را که در Cache قرار داده را تحت تأثیر قرار می‌دهد؟ دلیل خود را توضیح دهید.

Caching یک فضای ذخیره‌سازی موقت است که برخی از اسناد را در خود ذخیره می‌کند تا نیازی به دریافت دوباره آنها نباشد. اگر ارتباط بین کاربر و Cache سریع‌تر (یا دارای پهنای باند بیشتر) باشد، تاخیر درخواست کاهش می‌یابد.

Caching تاخیر تمام اشیاء را کم می‌کند، حتی آنهایی که Cache نشده‌اند؛ چرا که Caching ترافیک کل ارتباطات خود را کاهش می‌دهد.



4) فرض کنید شما یک صفحه وب را که شامل یک سند و پنج عکس است را درخواست می‌کنید. اندازه سند 1 کیلوبایت، حجم هر تصویر 50 کیلوبایت، سرعت دانلود 1 مگابایت در ثانیه و $RTT = 100\text{ ns}$ است. در حالت‌های زیر چه مدت طول می‌کشد تا کل محتوای سایت را دریافت کنیم؟ (فرض کنید نیازی به DNS Query نیست و زمان جابجایی هدرها و دیگر سربرها در پیام‌های HTTP ناچیز است.)

a. Non-Persistent HTTP با اتصالات سری.

$$RTT(\text{Connection}) + 6 \times RTT(1 \text{ doc} + 5 \text{ pic}) + \frac{(1 + 5 \times 50 \text{ KB}) \times \frac{8 \text{ bit}}{1 \text{ Byte}}}{1 \text{ Mbps} \times \frac{1000 \text{ Kbps}}{1 \text{ Mbps}}}$$

$$= 100\text{ ns} + 6 \times 100\text{ ns} + 2.008 \text{ s} = 2.0080007 \text{ s}$$

b. Non-Persistent HTTP با دو اتصال موازی.

$$RTT(\text{Connection}) + 3 \times RTT(1 \text{ doc} + 5 \text{ pic}, 2 \text{ conn}) + \frac{(1 + 5 \times 50 \text{ KB}) \times \frac{8 \text{ bit}}{1 \text{ Byte}}}{1 \text{ Mbps} \times \frac{1000 \text{ Kbps}}{1 \text{ Mbps}}}$$

$$= 100\text{ ns} + 6 \times 100\text{ ns} + 2.008 \text{ s} = 2.0080004 \text{ s}$$

c. Persistent HTTP با یک اتصال.

$$RTT(\text{Connection} + \text{download}) + \frac{(1 + 5 \times 50 \text{ KB}) \times \frac{8 \text{ bit}}{1 \text{ Byte}}}{1 \text{ Mbps} \times \frac{1000 \text{ Kbps}}{1 \text{ Mbps}}}$$

$$= 100\text{ ns} + 6 \times 100\text{ ns} + 2.008 \text{ s} = 2.0080001 \text{ s}$$

5) چرا در پرتکل‌های ارسال مطمئن داده (RDT) نیاز است از شماره ترتیب و تایمر استفاده کنیم؟
شماره ترتیب: برای این‌که فرستنده تشخیص بدهد که بسته دریافتی دارای اطلاعات جدید است یا ارسال دوباره اطلاعات قبلی
تایمر: اگر در مدتی مشخص بسته دریافت نشود، فرض می‌شود که packet loss رخ داده است؛ برای همین، باید دوباره ارسال شود.

(6) توضیح دهید که به چه دلایلی ممکن است یک توسعه دهنده برنامه، به جای TCP از UDP استفاده کند؟

در سیستم‌های Real-Time، از آن جا که بسته‌ها باید به ترتیب و با تأخیر حداقل منتقل شوند، استفاده از UDP ارجحیت دارد. نمی‌توان از TCP برای انتقال همزمان اطلاعات به چند مقصد استفاده کرد و باید از UDP استفاده شود.

(7) آیا امکان دارد که برنامه‌ای که از UDP استفاده می‌کند از قابلیت انتقال اطلاعات مطمئن هم بهره‌برد؟ به چه صورت؟ در صورت امکان دشواری‌های پیش رو چه هستند؟
بله، می‌توان قابلیت انتقال اطلاعات مطمئن را در لایه کاربرد (Application Layer) پیاده‌سازی کرد که نیازمند اضافه کردن بسیاری از موارد است.

(8) دو نمود و نشانه اصلی رخداد ازدحام چیست؟ (توضیح دهید)

- Lost packets(Buffer overflow at routers)
تعداد بسته‌های ارسالی زیاد بوده و بافرها ظرفیت کافی برای نگهداری همه‌ی آن‌ها را ندارند و بسته‌ها loss می‌شوند.
- Long delays(Queuing in router buffers)
در این حالت بسته‌ها loss نمی‌شوند اما مدت زیادی در صف باقی می‌مانند تا پردازش شوند؛ که این هم نشان‌دهنده ازدحام است.

(9) فرض کنید در یک اتصال UDP گیرنده با محاسبه، Checksum به این نتیجه برسد که Checksum محاسبه شده با مقداری که درون فیلد Checksum بسته دریافتی وجود دارد، یکسان است. آیا می‌توان مطمئن گفت که به طور حتم هیچ خطایی در ارسال بیت‌ها رخ نداده است؟ چرا؟ توضیح دهید.
خیر؛ اگر بیت‌ها به گونه‌ای تغییر کنند که Checksum آن‌ها با Checksum دریافتی یکسان باشد، داده دریافتی دچار تغییر شده ولی خطایی گزارش نمی‌شود.
مثال:

47 6F 6F 64

48 6E 6F 64