



دانشکده مهندسی کامپیوتر

بستری
دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

درس شبکه‌های کامپیوتری، نیمیال دوم سال تحصیلی ۱۴۰۳-۱۴۰۴

پایخ تمرین سری سوم



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

سوال ۱:

الف) دو معماری سرویس‌دهنده-سرویس‌گیرنده و معماری نظیر به نظیر را تعریف کنید و برای هر کدام دو کاربرد مثال بزنید.

ب) سه اصل امنیت اطلاعات و نقش هر یک در حفاظت از داده‌ها را توضیح دهید.

ج) تعریف سرویس‌دهنده و سرویس‌گیرنده، مستقل از نوع شبکه، به چه صورت است؟

پاسخ:

الف) دو معماری برای پیاده‌سازی برنامه‌های کاربردی در بستر شبکه‌های کامپیوتری، معماری سرویس‌دهنده-سرویس‌گیرنده (Client-Server) و معماری نظیر به نظیر (Peer-to-Peer) هستند. شرح و کاربردهای این دو معماری در جدول زیر آمده است:

ردیف	عنوان	معماری سرویس‌دهنده-سرویس‌گیرنده (Client-Server)	معماری نظیر به نظیر (Peer-to-Peer)
۱	شرح	برنامه کاربردی سمت سرویس‌دهنده (Server) همواره نقش ثابت سرویس‌دهی و برنامه کاربردی سمت سرویس‌گیرنده (Client) همواره نقش ثابت دریافت‌کننده سرویس را دارد. سرویس‌دهنده باید بر روی یک ماشین با آدرس IP ثابت و از طریق یک سوکت با شماره پورت ثابت در حال اجرا باشد و آماده برای دریافت درخواست‌های سرویس‌گیرنده و ارائه سرویس باشد. سرویس‌گیرنده باید از قبل آدرس IP و شماره پورت برنامه کاربردی سرویس‌دهنده را داشته باشد و درخواست دریافت سرویس خود را برای سرویس‌گیرنده ارسال و منتظر دریافت پاسخ باشد. آدرس IP و شماره پورت سرویس‌گیرنده (Client) می‌تواند موقتی باشد.	هر نظیر هم نقش سرویس‌دهی و هم نقش دریافت‌کننده سرویس را دارد. هر نظیر برای دریافت سرویس به یک یا چند نظیر دیگر درخواست دریافت سرویس را می‌دهد و به طور همزمان می‌تواند به نظیرهای دیگر سرویس ارائه دهد. نظیرها می‌توانند به شبکه نظیر به نظیر ملحق شده و یا از آن خارج شوند، بنابر این یکی از پیچیدگی‌های پیاده‌سازی شبکه‌های نظیر به نظیر، مدیریت ورود و خروج نظیرها و همچنین منابع در اختیار برای ارائه سرویس به سایر نظیرها است.
۲	مقیاس‌پذیری	مقیاس‌پذیر نیست، با افزایش تعداد سرویس‌گیرندگان، ظرفیت سرویس‌دهی سرویس‌دهنده نیز افزایش یابد.	مقیاس‌پذیر است، با افزایش تعداد نظیرها، به طور همزمان حجم درخواست و حجم سرویس‌دهی افزایش می‌یابد.
۳	کاربردهای نمونه	سرویس وب، سرویس ایمیل، سرویس DNS	اشتراک‌گذاری فایل (بیت‌تورنت)، اسکایپ

ب) سه اصل امنیت اطلاعات عبارتند از:

- محرمانگی (Confidentiality): جلوگیری از دسترسی غیرمجاز به اطلاعات و حفظ حریم خصوصی
- درستی داده‌ها (Integrity): اطمینان از درستی و صحت اطلاعات (تغییر نکردن اطلاعات در هنگام انتقال و نگهداری اطلاعات).
- در دسترس بودن (Availability): اطمینان از دسترسی کاربران مجاز به اطلاعات در زمان مورد نیاز.



ج) سرویس‌دهنده، برنامه‌کاربردی ارائه‌دهنده سرویس است و سرویس‌گیرنده، برنامه‌کاربردی دریافت‌کننده سرویس است. در شبکه‌های نظیر به نظیر هر نظیر همزمان می‌تواند هر دو نقش سرویس‌دهنده و سرویس‌گیرنده را داشته باشد، اما در شبکه‌های سرویس‌دهنده-سرویس‌گیرنده، سرویس‌دهنده فقط نقش ارائه سرویس را دارد و سرویس‌گیرنده نیز فقط نقش دریافت‌کننده سرویس را دارد.

سوال ۲:

الف) تفاوت بین پروتکل‌های عمومی و پروتکل‌های اختصاصی چیست؟

ب) نقش سوکت (socket) در ارتباط بین فرایندها چیست؟ چه تفاوتی بین سوکت TCP و UDP وجود دارد؟

پاسخ:

الف) پروتکل‌های عمومی، پروتکل‌هایی هستند که تمام جزئیات پیاده‌سازی آن‌ها آزادانه در دسترس عموم قرار دارد و همه می‌توانند از آن‌ها در پیاده‌سازی برنامه‌های کاربردی خود استفاده نمایند. به عنوان نمونه پروتکل HTTP یک پروتکل عمومی است و در پیاده‌سازی برنامه‌های کاربردی مبتنی بر وب استفاده می‌شود. اما، پروتکل‌های اختصاصی متعلق به یک شرکت خاص هستند و فقط در برنامه‌های کاربردی آن شرکت استفاده می‌شوند و جزئیات پیاده‌سازی پروتکل در دسترس عموم قرار ندارد. به عنوان نمونه پروتکل استفاده شده در برنامه‌کاربردی اسکایپ، یک پروتکل اختصاصی است.

ب) هر فرایند (برنامه‌کاربردی در حال اجرا) برای تبادل پیام با فرایند دیگری که بر روی کامپیوتر دیگری در حال اجرا است، می‌بایست از سرویس انتقال پیام ارائه شده توسط لایه انتقال استفاده کند. برای این منظور، هر فرایند باید یک سوکت (درب) با لایه انتقال باز کند تا از طریق آن بتواند ارسال و دریافت پیام با فرایند دیگر را انجام دهد. در واقع سوکت، یک نقطه پایانی است که دو فرایند می‌توانند از طریق آن تبادل پیام انجام دهند. به طور کلی، دو نوع سوکت برای تبادل اطلاعات وجود دارد. سوکت TCP یا Stream Socket که از طریق آن یک سرویس انتقال پیام مطمئن اتصال‌گرا به صورت جریانی از بایت‌ها ارائه می‌شود و سوکت UDP یا Datagram Socket که از طریق آن یک سرویس انتقال غیرمطمئن بدون اتصال پیام‌های مجزا ارائه می‌شود.

سوال ۳:

با توجه به پاسخ دریافتی زیر در پاسخ به یک درخواست GET، به سوالات زیر پاسخ دهید:

```
HTTP/1.1 200 OK<CR><LF>
Date: Sat, 02 Mar 2024 12:00:00 GMT<CR><LF>
Server: Apache/2.4.41 (Ubuntu)<CR><LF>
Content-Type: application/json<CR><LF>
Content-Length: 85<CR><LF>
Connection: keep-alive<CR><LF>
<CR><LF>
{"status": "success", "message": "Welcome to our API", "data": {"user": "JohnDoe", "role": "admin"}}<CR><LF>
```

الف) آیا این درخواست موفق بوده است. از کجا متوجه می‌شوید؟

ب) چه چیزی نشان‌دهنده پایان سرایند و شروع بدنه پاسخ در این پیام است؟

ج) چگونه مقدار Content-Length محاسبه می‌شود و چه واحدی دارد؟

د) کدام یک از سرایندهای بالا اختیاری و کدام یک اجباری هستند؟

ه) چرا (<CR><LF>) در انتهای هر خط از سرایندها استفاده می‌شود؟



پاسخ:

الف) با توجه به کد ۲۰۰ می‌توان نتیجه گرفت که این درخواست موفق بوده است.

ب) بعد از دو تا CRLF پشت سر هم یعنی <CR><LF><CR><LF> بدنه درخواست شروع می‌شود.

ج) مقدار Content-Length تعداد بایت‌های بخش بدنه‌ی پاسخ را نشان می‌دهد.

د) به طور کل در پاسخ به درخواست در HTTP، سرآیند Date و نسخه http (به طور مثال در این سوال HTTP/1.1 هست) و وضعیت درخواست اجباری هستند و سایر سرآیندها در این پروتکل اختیاری هست اما بعضی از سرورها بنا به نیازهای خود سرآیندهایی را اجباری قرار می‌دهند.

ه) در این پروتکل، پایان هر خط از سرآیندها باید با CRLF (Carriage Return + Line Feed) مشخص شود. این کار برای آن هست که در سمت مقابل امکان جداسازی (خط‌های) سرآیندها امکان‌پذیر باشد.

سوال ۴:

بر اساس نمونه تعامل زیر به سوالات پاسخ دهید.

```
220 mail.example.com ESMTP Postfix
HELO client.example.com
250 mail.example.com
MAIL FROM: <sender@example.com>
250 OK
RCPT TO: <recipient@example.com>
250 OK
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: Test Email
Hello, this is a test email.
.
250 OK: queued as 12345
QUIT
221 Bye
```

الف) HELO در فرایند دست‌دهی (handshake) پروتکل SMTP چیست؟

ب) چرا دستور DATA به یک دنباله خاص <CR><LF>.<CR><LF> برای نشان دادن پایان پیام نیاز دارد؟

ج) پاسخ سرور "250 OK: queued as 12345" چه چیزی را نشان می‌دهد؟

د) اگر جلسه SMTP قبل از تکمیل دستور DATA قطع شود، چه اتفاقی خواهد افتاد؟

پاسخ:

الف) سرویس‌گیرنده (Client) بعد از بازکردن اتصال TCP با ارسال پیام HELO همراه نام دامنه خود، درخواست ایجاد اتصال با سرویس‌دهنده (Server) را می‌کند. سرویس‌دهنده نیز با دریافت پیام HELO با ارسال پیام پاسخ OK همراه نام دامنه خود ایجاد اتصال با سرویس‌گیرنده را تایید می‌کند. با رد و بدل شدن پیام HELO و OK بین سرویس‌گیرنده و سرویس‌دهنده، اتصال در وضعیت اولیه که هیچ تراکنشی در حال انجام وجود ندارد و تمامی جدول‌های وضعیت و بافرها پاک شده‌اند قرار می‌گیرد.



ب) از آنجایی که داده‌های ایمیل از طریق کانال انتقال ارسال می‌شوند، باید پایانی برای داده‌های ایمیل مشخص شود تا گفت‌وگوی بین دستورات و پاسخ‌ها بتواند از سر گرفته شوند. SMTP پایان داده‌های ایمیل را با ارسال یک خط که تنها شامل یک نقطه (.) است، مشخص می‌کند.

ج) این پاسخ نشان می‌دهد که سرویس‌دهنده ایمیل پیام را دریافت کرده و آن را در صف پردازش (queue) قرار داده است. عدد ۱۲۳۴۵ یک شناسه (ID) است که به پیام اختصاص داده شده و از آن برای پیگیری یا خطایابی استفاده می‌شود. این پیام نشان می‌دهد که سرویس‌دهنده ایمیل آماده ارسال پیام به گیرنده است.

د) اگر اتصال به‌طور ناگهانی بسته شود، سرویس‌دهنده باید طوری رفتار کند که گویی یک دستور RSET دریافت کرده است (لغو هر تراکنش در حال انتظار، اما بدون بازگرداندن تراکنش‌هایی که قبلاً تکمیل شده‌اند). همچنین، سرویس‌گیرنده باید طوری عمل کند که گویی دستور یا تراکنش در حال اجرا با یک خطای موقت (4xx code) مواجه شده است.

سوال ۵:

دو سرویس‌گیرنده می‌خواهند صفحه‌ای به حجم ۲ مگا بایت که شامل ۸ فایل با حجم ۲۰ مگا بایت است را دریافت کنند. بین هر دو سرویس‌گیرنده تا سرور لینک مشترکی با پهنای باند 128 Mbps وجود دارد و زمان RTT بین هر سرویس‌گیرنده تا سرور نیز ۷۰ میلی‌ثانیه است.

الف) در صورتی که اتصال‌ها غیرمداوم باشند و هر سرویس‌گیرنده بتواند تا ۸ اتصال موازی داشته باشد. حداقل زمانی که طول می‌کشد در هر دو سرویس‌گیرنده صفحه وب به طور کامل دریافت شود چقدر است؟

ب) در صورتی که اتصال‌ها غیرمداوم باشند اما یکی از سرویس‌گیرنده‌ها ۴ اتصال موازی و سرویس‌گیرنده دیگر ۱۲ اتصال موازی داشته باشد، چقدر طول می‌کشد که هر سرویس‌گیرنده صفحه وب را به طور کامل دریافت کند؟

پاسخ:

زمان دریافت اولین شیء (base html) توسط سرویس‌گیرندگان برابر است با:

$$t_{base\ html} = 2 \times 70 \times 10^{-3} + \frac{2 \times 2^{20} \times 8}{\frac{128 \times 10^6}{2}} = 0.14 + 0.26 = 0.40 \text{ Sec}$$

الف) هر سرویس‌گیرنده می‌تواند حداکثر ۸ اتصال موازی باز کند، بنابراین با توجه به اینکه این صفحه وب حاوی ۸ شیء دیگر است هر سرویس‌گیرنده نیز ۸ اتصال موازی برای دریافت شیء‌ها باز خواهد کرد. در نتیجه زمان دریافت این صفحه وب برابر است با زمان دریافت شیء اول (base html) به علاوه زمان دریافت ۸ شیء دیگر.

$$d_{each\ link} = \frac{128 \times 10^6}{8 \times 2} = 8 \times 10^6 = 8 \text{ Mbps}$$

$$t_{files} = 2 \times 70 \times 10^{-3} + \frac{20 \times 2^{20} \times 8}{8 \times 10^6} = 0.14 + 20.97 = 21.11 \text{ Sec}$$

$$t_{total} = 0.40 + 21.11 = 21.51 \text{ Sec}$$

ب) از آنجایی که یکی از سرویس‌گیرنده‌ها (فرض کنید سرویس‌گیرنده اول) فقط می‌تواند ۴ اتصال موازی باز کند، در نتیجه این سرویس‌گیرنده در هر مرحله ۴ شیء و در دو مرحله پشت سرهم ۸ شیء را دریافت می‌کند. اما سرویس‌گیرنده دیگر (سرویس‌گیرنده دوم) در یک مرحله با بازکردن ۸ اتصال موازی هر ۸ شیء را دریافت می‌کند. بنابراین این در مرحله اول مجموعاً ۱۲ اتصال موازی اما در مرحله دوم فقط ۴ اتصال موازی بر روی لینک مشترک به سرویس‌دهنده وجود دارد، در نتیجه زمان انتقال ۸ شیء و زمان کل دریافت کامل صفحه به شرح زیر محاسبه می‌شود:

$$t_{files\ u1} = \left(2 \times 70 \times 10^{-3} + \frac{20 \times 2^{20} \times 8}{\frac{128 \times 10^6}{8 + 4}} \right) + \left(2 \times 70 \times 10^{-3} + \frac{20 \times 2^{20} \times 8}{\frac{128 \times 10^6}{4}} \right)$$

$$= (0.14 + 15.73) + (0.14 + 5.24) = 21.11 \text{ Sec}$$



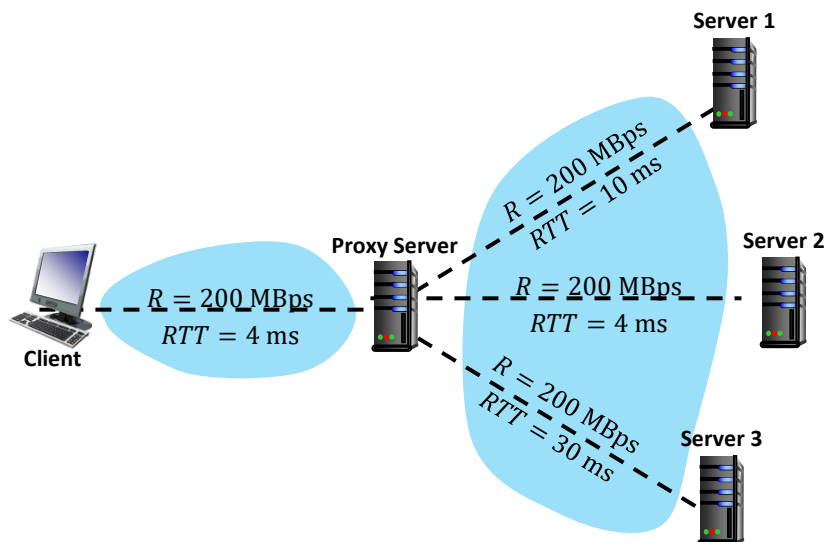
$$t_{files\ u2} = \left(2 \times 70ms + \frac{20 \times 2^{20} \times 8}{\frac{128 \times 10^6}{8 + 4}} \right) = (0.14 + 15.73) = 15.87 \text{ Sec}$$

$$t_{total\ u1} = 0.40 + 21.11 = 21.51 \text{ Sec}$$

$$t_{total\ u2} = 0.40 + 15.87 = 16.27 \text{ Sec}$$

سوال ۶:

یک سرویس‌گیرنده وب می‌خواهد سه شیء وب A، B و C که به ترتیب در سرویس‌دهنده‌های وب ۱، ۲ و ۳ قرار دارند را از طریق یک سرویس‌دهنده پروکسی مطابق با شکل زیر دریافت کند. اندازه‌های این ۳ شیء به ترتیب ۴ مگابایت، ۱۰ مگابایت و ۲۰ مگابایت است و همچنین تأخیر رفت و برگشت (RTT) بین میزبان‌ها و نرخ داده لینک‌ها نیز در شکل مشخص شده است. احتمال وجود شیء‌های A، B و C در حافظه پنهان موقتی (Cache) پروکسی به ترتیب برابر با 0، 0 و p است، در صورتی که از پروتکل HTTP غیرمداوم برای دریافت این ۳ شیء استفاده شده باشد و میانگین مجموع زمان دریافت این شیء‌ها **۳۶۰ میلی ثانیه** شده باشد، مقدار احتمال p را بدست آورید.



پاسخ:

توضیح:

(۱) مجموع دریافت سه شیء ۳۶۰ میلی ثانیه است که اشتباهاً ۲۴۰ میلی ثانیه در صورت سوال ذکر شده بود.

(۲) نرخ لینک‌ها ۲۰۰ مگا بایت در ثانیه (200 MBps) است.

حل:

اگر یک شیء در حافظه پنهان سرویس‌دهنده پروکسی وجود نداشته باشد، زمان دریافت آن شیء برابر است با:

$$t_{non-cached-object} = 2RTT_{client-to-proxy} + t_{proxy-to-client} + 2RTT_{proxy-to-server} + t_{server-to-proxy}$$

و اگر یک شیء در حافظه پنهان سرویس‌دهنده پروکسی وجود داشته باشد، زمان دریافت آن شیء برابر است با:

$$t_{cached-object} = 2RTT_{client-to-proxy} + t_{proxy-to-client}$$

و اگر احتمال اینکه یک شیء در حافظه پنهان سرویس‌دهنده پروکسی وجود داشته باشد، برابر با p باشد، متوسط زمان دریافت آن شیء برابر است با:

$$t_{p-probability-cached-object} = (1 - p) \times t_{non-cached-object} + p \times t_{cached-object}$$



در نتیجه:

$$t_A = 2 \times 4 \times 10^{-3} + \frac{4 \times 2^{20}}{200 \times 10^6} + 2 \times 10 \times 10^{-3} + \frac{4 \times 2^{20}}{200 \times 10^6} = 0.008 + 0.02097 + 0.020 + 0.02097 = 0.06994 \text{ Sec}$$

$$t_B = 2 \times 4 \times 10^{-3} + \frac{10 \times 2^{20}}{200 \times 10^6} + 2 \times 4 \times 10^{-3} + \frac{10 \times 2^{20}}{200 \times 10^6} = 0.008 + 0.05242 + 0.008 + 0.05242 = 0.12084 \text{ Sec}$$

$$t_C = (1 - p) \times \left(2 \times 4 \times 10^{-3} + \frac{20 \times 2^{20}}{200 \times 10^6} + 2 \times 30 \times 10^{-3} + \frac{20 \times 2^{20}}{200 \times 10^6} \right) + p \times \left(2 \times 4 \times 10^{-3} + \frac{20 \times 2^{20}}{200 \times 10^6} \right)$$

$$= (1 - p) \times (0.008 + 0.10486 + 0.060 + 0.10486) + p \times (0.008 + 0.10486) = 0.27772 - 0.16486p$$

با توجه به اینکه میانگین مجموع تأخیر برابر است با ۳۶۰ میلی ثانیه در نتیجه:

$$t_{total} = t_A + t_B + t_C = 0.06994 + 0.12084 + 0.27772 - 0.16486p = 0.46850 - 0.16486p = 0.360$$

$$\Rightarrow p = 0.6584$$

سوال ۷:

یک سرویس‌گیرنده وب می‌خواهد یک صفحه وب را دریافت کند. فرض کنید اندازه اولین شیء (base html) ۸ مگا بایت است و اولین شیء به ۱۰ شیء دیگر ارجاع داده است که اندازه هر یک آن شیء‌ها ۱۶ مگابایت است. اگر شیء اولیه (base html) و ۵ شیء در سرویس‌دهنده A و ۵ شیء دیگر در سرویس‌دهنده B باشند. با توجه به اطلاعات داده در شکل زیر، زمان لازم برای دریافت کامل این صفحه وب را در حالت‌های زیر بدست آورید.

الف) استفاده از اتصال غیرمداوم با امکان ایجاد اتصال موازی بدون محدودیت

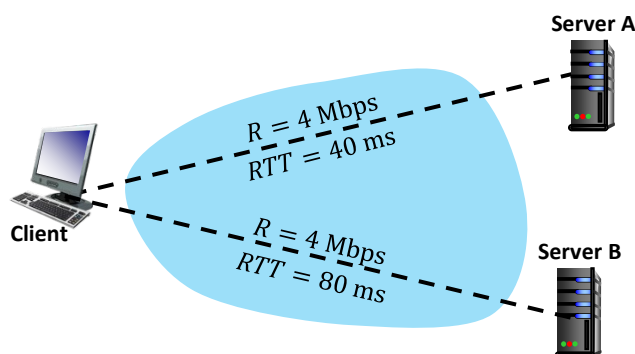
ب) استفاده از اتصال غیرمداوم با ۵ اتصال موازی

ج) استفاده از اتصال مداوم بدون خط لوله و عدم امکان ایجاد اتصال موازی با سرویس‌دهنده‌ها

د) استفاده از اتصال مداوم بدون خط لوله و امکان ایجاد اتصال موازی با سرویس‌دهنده‌ها

ه) استفاده از اتصال مداوم با خط لوله و عدم امکان ایجاد اتصال موازی با سرویس‌دهنده‌ها

و) استفاده از اتصال مداوم با خط لوله و امکان ایجاد اتصال موازی با سرویس‌دهنده‌ها



پاسخ:

Data Packet Transmission Time for one parallel connection:

$$t_{d1} = \frac{L_d}{R}$$

Data Packet Transmission Time for N parallel connection:

$$t_{dN} = \frac{L_d}{\frac{R}{N}} = \frac{NL_d}{R} = Nt_{d1}$$



(الف)

Non-persistent connection with unlimited parallel connections (10 parallel connections):

$$\begin{aligned}
 \text{ResponseTime} &= 2RTT_A + \frac{L_{base}}{R_A} + \max(2RTT_A + t_{5A}, 2RTT_B + t_{5B}) \\
 &= 2 \times 40 \times 10^{-3} + \frac{8 \times 2^{20} \times 8}{4 \times 10^6} + \max\left(2 \times 40 \times 10^{-3} + \frac{16 \times 2^{20} \times 8}{4 \times 10^6}, 2 \times 80 \times 10^{-3} + \frac{16 \times 2^{20} \times 8}{4 \times 10^6}\right) \\
 &= 80 \times 10^{-3} + 16.78 + 160 \times 10^{-3} + 167.78 = 184.70 \text{ Sec}
 \end{aligned}$$

(ب)

Non-persistent connection with 5 parallel connections:

$$\begin{aligned}
 \text{ResponseTime} &= 2RTT_A + \frac{L_{base}}{R_A} + 2RTT_A + t_{5A} + 2RTT_B + t_{5B} \\
 &= 2 \times 40 \times 10^{-3} + \frac{8 \times 2^{20} \times 8}{4 \times 10^6} + 2 \times 40 \times 10^{-3} + \frac{16 \times 2^{20} \times 8}{4 \times 10^6} + 2 \times 80 \times 10^{-3} + \frac{16 \times 2^{20} \times 8}{4 \times 10^6} \\
 &= 80 \times 10^{-3} + 16.78 + 80 \times 10^{-3} + 167.78 + 160 \times 10^{-3} + 167.78 = 352.66 \text{ Sec}
 \end{aligned}$$

(ج)

Persistent no pipeline connection without parallel connection to servers:

$$\begin{aligned}
 \text{ResponseTime} &= 2RTT_A + \frac{L_{base}}{R_A} + 5 \times (RTT_A + t_A) + RTT_B + 5 \times (RTT_B + t_B) \\
 &= 2 \times 40 \times 10^{-3} + \frac{8 \times 2^{20} \times 8}{4 \times 10^6} + 5 \times (40 \times 10^{-3} + \frac{16 \times 2^{20} \times 8}{4 \times 10^6}) + 80 \times 10^{-3} + 5 \times (80 \times 10^{-3} + \frac{16 \times 2^{20} \times 8}{4 \times 10^6}) \\
 &= 80 \times 10^{-3} + 16.78 + 5 \times 33.59 + 80 \times 10^{-3} + 5 \times 33.68 = 353.04 \text{ Sec}
 \end{aligned}$$

(د)

Persistent no pipeline connection with parallel connection to servers:

$$\begin{aligned}
 \text{ResponseTime} &= 2RTT_A + \frac{L_{base}}{R_A} + \max(5 \times (RTT_A + t_A), RTT_B + 5 \times (RTT_B + t_B)) \\
 &= 2 \times 40 \times 10^{-3} + \frac{8 \times 2^{20} \times 8}{4 \times 10^6} + \max(5 \times (40 \times 10^{-3} + \frac{16 \times 2^{20} \times 8}{4 \times 10^6}), 80 \times 10^{-3} + 5 \times (80 \times 10^{-3} + \frac{16 \times 2^{20} \times 8}{4 \times 10^6})) \\
 &= 80 \times 10^{-3} + 16.78 + 80 \times 10^{-3} + 5 \times 33.63 = 185.09 \text{ Sec}
 \end{aligned}$$

(ه)

persistent pipeline connection without parallel connections to servers:

$$\begin{aligned}
 \text{ResponseTime} &= 2RTT_A + \frac{L_{base}}{R_A} + RTT_A + 5t_A + 2RTT_B + 5t_B \\
 &= 2 \times 40 \times 10^{-3} + \frac{8 \times 2^{20} \times 8}{4 \times 10^6} + 40 \times 10^{-3} + 5 \times \frac{16 \times 2^{20} \times 8}{4 \times 10^6} + 2 \times 80 \times 10^{-3} + 5 \times \frac{16 \times 2^{20} \times 8}{4 \times 10^6} \\
 &= 80 \times 10^{-3} + 16.78 + 40 \times 10^{-3} + 5 \times 33.55 + 160 \times 10^{-3} + 5 \times 33.55 = 352.48 \text{ Sec}
 \end{aligned}$$

(و)

persistent pipeline connection with parallel connections to servers:

$$\begin{aligned}
 \text{ResponseTime} &= 2RTT_A + \frac{L_{base}}{R_A} + \max(RTT_A + 5t_A, 2RTT_B + 5t_B) \\
 &= 2 \times 40 \times 10^{-3} + \frac{8 \times 2^{20} \times 8}{4 \times 10^6} + \max(40 \times 10^{-3} + 5 \times \frac{16 \times 2^{20} \times 8}{4 \times 10^6}, 2 \times 80 \times 10^{-3} + 5 \times \frac{16 \times 2^{20} \times 8}{4 \times 10^6}) \\
 &= 80 \times 10^{-3} + 16.78 + 160 \times 10^{-3} + 5 \times 33.55 = 184.77 \text{ Sec}
 \end{aligned}$$



سوال ۸:

الف) فرض کنید که یک درخواست DNS باید از سرور ریشه به سرور TLD و سپس به سرور معتبر هدایت شود. توضیح دهید که فرایند انتقال درخواست DNS چگونه از یک سرور به سرور دیگر صورت می‌گیرد.

ب) زمان انتقال از سرور ریشه به سرور TLD برابر با 35 ms و از سرور TLD به سرور معتبر 50 ms طول می‌کشد. اگر روزانه 10000 درخواست DNS به سرور ارسال شود، مجموع زمان تأخیر برای پردازش این درخواست‌ها چقدر خواهد بود؟

ج) اگر 20 درصد از درخواست‌ها نیازی به ارسال به سرور TLD نداشته باشند و به طور مستقیم از سرور ریشه به سرور معتبر هدایت شوند، چقدر زمان پردازش درخواست‌ها کاهش خواهد یافت؟

پاسخ:

الف) برای تبدیل نام دامنه به آدرس IP دو روش تکراری (iterative) و بازگشتی (recursive) برای ارسال پیام‌های پرس و جو وجود دارد. در روش تکراری ارسال پیام‌های پرس و جو به صورت زیر انجام می‌شود:

۱. ارسال پیام پرس و جو از سرویس‌گیرنده به سرویس‌دهنده DNS ریشه و دریافت آدرس سرویس‌دهنده TLD DNS
۲. ارسال پیام پرس و جو از سرویس‌گیرنده به سرویس‌دهنده TLD DNS، و دریافت آدرس سرویس‌دهنده DNS معتبر
۳. ارسال پیام پرس و جو از سرویس‌گیرنده به سرویس‌دهنده DNS معتبر، و دریافت آدرس IP نام دامنه مورد نظر.

در روش بازگشتی ارسال پیام‌های پرس و جو به صورت زیر انجام می‌شود:

۱. ارسال پیام پرس و جو از سرویس‌گیرنده به سرویس‌دهنده DNS ریشه
۲. ارسال پیام پرس و جو از سرویس‌دهنده DNS ریشه به سرویس‌دهنده TLD DNS
۳. ارسال پیام پرس و جو از سرویس‌دهنده TLD DNS به سرویس‌دهنده DNS معتبر
۴. ارسال پیام پاسخ تبدیل آدرس IP نام دامنه مورد نظر از سرویس‌دهنده DNS معتبر به سرویس‌دهنده TLD DNS
۵. ارسال پیام پاسخ تبدیل آدرس IP نام دامنه مورد نظر از سرویس‌دهنده TLD DNS به سرویس‌دهنده DNS ریشه
۶. ارسال پیام پاسخ تبدیل آدرس IP نام دامنه مورد نظر از سرویس‌دهنده DNS ریشه به سرویس‌گیرنده

(ب)

$$total\ delay = (35 \times 10^{-3} + 50 \times 10^{-3}) \times 10000 = 850\ Sec$$

ج) اگر ۲۰٪ از درخواست‌ها نیازی به مراجعه به سرور TLD نداشته باشند و مستقیماً از سرور ریشه بتوانند نشانی سرور معتبر را بگیرند، عملاً ۲۰٪ درخواست‌ها مسیر TLD به root را طی نمی‌کنند. در نتیجه برای آن ۲۰٪، زمان مورد نیاز فقط بخش ریشه تا سرور معتبر خواهد بود. از آنجا که ۸۰٪ درخواست‌ها هنوز همان ۸۵ میلی‌ثانیه را مصرف می‌کنند و ۲۰٪ باقی‌مانده فقط ۵۰ میلی‌ثانیه می‌گیرند، میانگین زمان هر درخواست برابر است با:

$$T_{avg} = 0.80 \times (35 \times 10^{-3} + 50 \times 10^{-3}) + 0.20 \times 50 \times 10^{-3} = 68 \times 10^{-3} + 10 \times 10^{-3} = 78\ mSec$$

و در نتیجه برای ۱۰,۰۰۰ درخواست در روز مجموع تأخیر برابر است:

$$78 \times 10^{-3} \times 10000 = 780\ Sec$$

و میزان کاهش زمان روزانه برابر است با:

$$850 - 780 = 70\ Sec$$

که به طور نسبی برابر است با:

$$\frac{70}{850} = 8.24\%$$



سوال ۹:

الف) توضیح دهید که چگونه پروتکل‌های P2P مانند Bit Torrent فایل‌ها را بین نظیرها توزیع می‌کنند. چگونه نظیرها به طور همزمان داده‌ها را ارسال و دریافت می‌کنند؟

ب) فرض کنید که اندازه یک فایل 500 MB است و 20 نظیر به طور همزمان اقدام به دریافت این فایل می‌کنند. اگر سرعت ارسال هر نظیر 1 Mbps باشد، زمان لازم برای دریافت کل فایل توسط یک نظیر چقدر است؟

ج) اگر سرعت ارسال داده در شبکه به دلیل بار زیاد کاهش یابد و به 0.5 Mbps برسد، زمان لازم برای دریافت کامل فایل چقدر خواهد بود؟

د) اگر ۱۰۰ نظیر به طور همزمان اقدام به دریافت این فایل کنند و سرعت دانلود برای هر نظیر 2 Mbps باشد، زمان کلی برای توزیع فایل بین نظیرها چقدر خواهد بود؟

پاسخ:

الف) در پروتکل‌های P2P نظیر BitTorrent، یک فایل به قطعه‌هایی (Chunk) کوچک تقسیم می‌شود. هر نظیر (Peer) به محض دریافت یک قطعه، می‌تواند آن قطعه را برای بقیه نظیرها بارگذاری (آپلود) کند. در نتیجه پس از مدتی، بسیاری از نظیرها قطعه‌های مختلف فایل را در اختیار دارند و همه آن‌ها به صورت توزیع‌شده در حال دریافت یا ارسال قطعه‌ها به یکدیگر هستند. این مکانیزم باعث می‌شود توان عملیاتی (Throughput) کل سیستم بالا رفته و توزیع فایل سریع‌تر از زمانی که فقط یک سرور مرکزی وظیفه بارگذاری را برعهده دارد انجام شود.

ب) زمان حداقلی که Seed باید یک نسخه کامل را آپلود کند:

$$F/u_{\text{seed}} = \frac{500 \times 2^{20} \times 8}{1 \times 10^6} = 4194.304 \text{ Sec}$$

زمان لازم برای اینکه همه ۲۰ نفر آن فایل را بگیرند:

$$\frac{N \times F}{u_{\text{seed}} + (N - 1) \times u_{\text{peer}}} = \frac{500 \times 2^{20} \times 8}{1 \times 10^6 + (20 - 1) \times 1 \times 10^6} = 4194.304 \text{ Sec}$$

$$t_{\text{total}} = \max\left(\frac{F}{u_{\text{seed}}}, \frac{N \times F}{u_{\text{seed}} + (N - 1) \times u_{\text{peer}}}\right) = \max(4194.304, 4194.304) = 4194.304 \text{ Sec}$$

(ج)

$$t_{\text{total}} = \max\left(\frac{F}{u_{\text{seed}}}, \frac{N \times F}{u_{\text{seed}} + (N - 1) \times u_{\text{peer}}}\right) = \max\left(\frac{500 \times 2^{20} \times 8}{0.5 \times 10^6}, \frac{20 \times 500 \times 2^{20} \times 8}{0.5 \times 10^6 + (20 - 1) \times 0.5 \times 10^6}\right) = 8388.608 \text{ Sec}$$

(د)

$$t_{\text{total}} = \max\left(\frac{F}{u_{\text{seed}}}, \frac{N \times F}{u_{\text{seed}} + (N - 1) \times u_{\text{peer}}}\right) = \max\left(\frac{500 \times 2^{20} \times 8}{2 \times 10^6}, \frac{100 \times 500 \times 2^{20} \times 8}{2 \times 10^6 + (100 - 1) \times 2 \times 10^6}\right) = 2097.152 \text{ Sec}$$

سوال ۱۰:

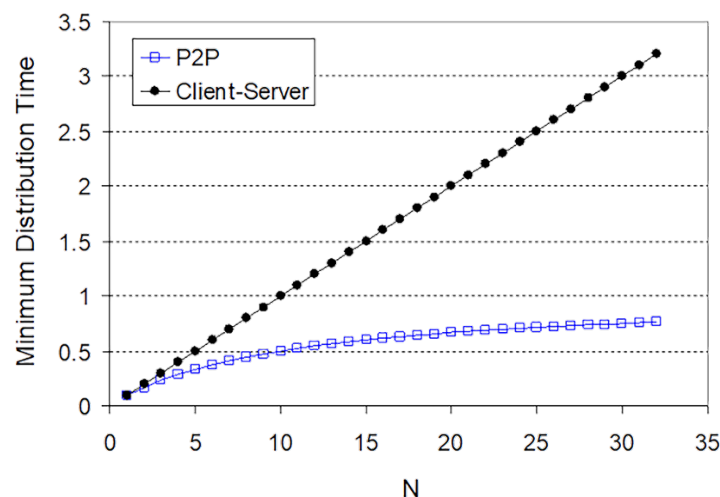
الف) چرا مدل P2P نسبت به مدل Client-Server زمان توزیع داده کمتری دارد و این تفاوت در چه شرایطی بیشتر قابل مشاهده است؟

ب) فرض کنید که تعداد کاربران در مدل P2P از ۳۰ نفر به ۴۰ نفر افزایش یابد. پیش‌بینی کنید که زمان حداقل توزیع داده در این مدل چقدر خواهد بود. سپس، این پیش‌بینی را با زمان حداقل توزیع داده در مدل Client-Server مقایسه کنید.



ج) در مدل Client-Server، اگر ظرفیت سرور به نصف کاهش یابد (یعنی قادر به ارسال فایل به نصف کاربران باشد)، چگونه این تغییر ظرفیت می‌تواند بر زمان حداقل توزیع فایل تأثیر بگذارد؟ چرا این تغییر می‌تواند در مدل Client-Server زمان توزیع را بیشتر از مدل P2P تحت تأثیر قرار دهد؟

د) با توجه به نمودار و روند افزایش زمان توزیع داده در هر دو مدل، در شرایط واقعی و با افزایش تعداد کاربران به مرزهای بالاتر کدام مدل مناسب‌تر برای استفاده در شبکه‌های بزرگ است و چرا؟



پاسخ:

الف) در مدل P2P، برخلاف مدل Client-Server که بارگذاری (آپلود) فقط از طریق سرور مرکزی انجام می‌شود، هر نظیر به محض دریافت هر بخش از فایل می‌تواند آن بخش را برای دیگران بفرستد. به همین دلیل مجموع نرخ آپلود کل سیستم با افزایش تعداد کاربران (نظیرها) بالا می‌رود و زمان توزیع کاهش می‌یابد. هرچه تعداد کاربران (N) بزرگ‌تر باشد، تفاوت کارایی P2P با مدل تک‌سرور بیشتر نمود پیدا می‌کند.

ب) اگر تعداد کاربران در مدل P2P مثلاً از ۳۰ نفر به ۴۰ نفر افزایش یابد، به دلیل آن که این ۱۰ نفر اضافه خودشان نیز در بارگذاری (آپلود) فایل نیز سهیم می‌شوند، رشد زمان توزیع آن قدر زیاد نمی‌شود؛ زیرا توان آپلود کل سیستم (مجموع آپلود همه نظیرها) افزایش می‌یابد. اما در مدل Client-Server، زیاد شدن تعداد کاربران به این معناست که سرور باید برای درخواست‌های تعداد بیشتری کاربر آپلود کند و زمان توزیع به مراتب بالاتر می‌رود.

ج) در مدل Client-Server اگر ظرفیت یا سرعت آپلود سرور نصف شود، عملاً مدت لازم برای توزیع همه داده‌ها تقریباً دو برابر خواهد شد؛ چون تنها منبع آپلود همان سرور است. اما در مدل P2P، حتی اگر توان آپلود اولیه (Seed) کاهش پیدا کند، نظیرهای دیگر می‌توانند بخش‌هایی از فایل را که دریافت کرده‌اند بین خودشان توزیع کنند. بنابراین کاهش سرعت سرور لزوماً ضربه شدیدی به کل زمان توزیع نمی‌زند.

د) با توجه به نموداری که معمولاً برای مقایسه P2P و Client-Server رسم می‌شود (همانند شکل پرسش)، در تعداد کم کاربران، اختلاف دو مدل زیاد نیست؛ اما با افزایش تعداد کاربران، زمان توزیع در مدل Client-Server به صورت خطی بالا می‌رود، درحالی که در P2P شیب افزایش کمتر است. به همین دلیل در شبکه‌های بزرگ و پرجمعیت (مثلاً ده‌ها هزار یا میلیون‌ها کاربر)، مدل P2P بسیار مناسب‌تر و مقیاس‌پذیرتر است. علت اصلی هم تجمع توان آپلود همه نظیرها و توزیع فایل توسط خود کاربران است.