



دانشکده مهندسی کامپیوتر

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

درس شبکه های کامپیوتری، نیم سال دوم سال تحصیلی ۱۴۰۱-۱۴۰۰
پایخ تمرین سری دوم



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

پاسخ سوال ۱:

در مدل OSI، لایه کاربرد وظیفه ارائه سرویس نظیر ارسال فایل و کنترل از راه دور سیستم ها و ... را برعهده دارد. برنامه های کاربردی که کاربران استفاده می کنند در این لایه قرار دارند. لایه کاربرد واسط ارتباطی کاربر با بقیه لایه ها است. برنامه هایی نظیر ارسال و دریافت ایمیل، به اشتراک گذاری فایل و برنامه های کاربردی دیگر در لایه کاربرد قرار دارند. در مدل OSI، پروتکل لایه کاربرد فقط به ارائه سرویس به کاربران می پردازد و وظایفی از قبیل کدگذاری و یا رمزنگاری اطلاعات و مدیریت نشست یا جلسه را به عهده ندارد.

در مدل TCP/IP، لایه کاربرد علاوه بر ارائه سرویس به کاربر که در مدل OSI شرح داده شد وظایف ارائه صحیح اطلاعات که در مدل OSI به عهده لایه ارائه (Presentation) و وظایف مدیریت جلسه که در مدل OSI به عهده لایه جلسه (Session) است را نیز به عهده دارد. در واقع، لایه کاربرد در مدل TCP/IP وظایف سه لایه بالای مدل OSI یعنی لایه کاربرد، لایه ارائه و لایه جلسه را به عهده دارد.

پاسخ سوال ۲:

(الف)

هر دو معماری برای به اشتراک گذاری منابع و ارائه یا دریافت سرویس استفاده می شوند. در معماری Client-Server نقش برنامه کاربردی ثابت است یعنی یا Client است یا Server. اما در معماری Peer-to-Peer این نقش ثابت نیست و هر برنامه کاربردی هم نقش Client را دارد در زمانی که سرویس دریافت می کند و هم نقش Server را دارد زمانی که سرویس ارائه می کند. جدول مقایسه به صورت زیر است:

ردیف	معیار	P2P	Client-Server
۱	نوع شبکه (از جهت ذخیره داده ها و ارائه سرویس)	Decentralized network هر peer بخشی از داده ها را دارد. peerها می توانند مستقیماً با هم در ارتباط باشند.	Centralized network داده ها در یک سرور مرکزی نگهداری می شوند. کلاینت ها فقط با سرور مرکزی در ارتباط مستقیم هستند.
۲	تمرکز اصلی معماری	ارتباطات و اتصالات communication & connectivity	تبادل داده data exchange
۳	نحوه انتقال داده (نقش ها)	هر گره می تواند هم سرور و هم کلاینت باشد. گره ها همگی یکسان هستند و نقش ثابتی ندارند.	نقش گره ها ثابت است. هر گره یا سرور است یا کلاینت. سرور تمام خدمات و داده ها را ارائه می دهد و کلاینت خدمات و داده ها را درخواست می کنند.
۴	هزینه	نسبت به Client/Server هزینه کمتری دارد زیرا نیازی به سرور مرکزی نیست.	پیاده سازی و نگهداری این شبکه هزینه بالایی دارد زیرا نیاز به سرور مرکزی همیشه در حال کار دارد.
۵	امنیت	از نظر کنترل دسترسی و محرمانگی: آسیب پذیرتر است زیرا گره ها همزمان در نقش سرور و کلاینت عمل می کنند و داده ها در همه گره ها توزیع شده است و کنترل دسترسی به داده ها به صورت متمرکز صورت نمی پذیرد. از نظر دسترسی پذیری:	از نظر کنترل دسترسی و محرمانگی: امنیت بهتری را فراهم می کند زیرا دسترسی به داده ها توسط سرور مرکزی کنترل می شود. از نظر دسترسی پذیری: بدلیل وجود سرور مرکزی در مقابل حملات ممانعت سرویس آسیب پذیرتر است.



	بدلیل ویژگی توزیع شدگی اطلاعات، در مقابل حملات ممانعت از سرویس مقاوم تر هستند.		
۶	کارایی	با افزایش تعداد گره‌ها در این شبکه، کارایی کاهش می‌یابد.	از پایداری بالایی برخوردار است و می‌تواند در صورت نیاز گسترش یابد.
۷	توزیع پهنای باند	در شبکه‌های P2P، از ابتدای اتصال پهنای باند کامل گره‌ها استفاده نمی‌شود. بلکه با توجه به پهنای باند موجود هر گره از آن استفاده و زمانی که مورد نیاز نیست آزاد می‌شود.	پهنای باند بستگی به پهنای باند لینک(های) ارتباطی اتصال سرور به شبکه دارد.

(ب)

از جمله کاربردهای معماری Client-Server در کاربردهای سرویس‌های ایمیل، دسترسی از راه دور، وب و موارد مشابه دیگر است. بدلیل متمرکز بودن سرورها و پایگاه داده‌ها، مدیریت ارتباطات بهتر صورت می‌گیرد و کیفیت سرویس‌دهی و امنیت در این سرویس‌ها تضمین می‌شود. کاربردهای معماری P2P بیشتر در به اشتراک‌گذاری فایل (file sharing)، بروزرسانی سیستم‌های نرم‌افزاری نظیر سیستم عامل و موارد مشابه دیگر است که عموماً حجم داده و ترافیک درخواست‌ها بسیار زیاد است و نیاز به سرورهای قوی (هزینه پیاده‌سازی و نگهداری بالا) وجود دارد. با استفاده از معماری P2P، کاربران با به اشتراک‌گذاری بخش‌هایی از فایل که در اختیار دارند و با سرعت بارگذاری خود به شبکه کمک می‌کنند و بدون نیاز به سرورهای قوی، حجم داده و ترافیک بین Peerها تقسیم می‌شود که باعث کاهش هزینه و سربار نگهداری سرورها می‌شود.

(ج)

علت اصلی استفاده از معماری P2P، ویژگی غیرمتمرکز بودن (Decentralized) است که باعث می‌شود در اشتراک‌گذاری اطلاعات مورد نیاز کلیه یا بخش زیادی از کاربران، کارایی، امنیت بهبود یابد و هزینه پیاده‌سازی و نگهداری کمتری نسبت به معماری Client-Server داشته باشند. چون هیچ سرور مرکزی در سیستم نیست، مهاجم نمی‌تواند با پر کردن ظرفیت آن، ارائه سرویس به کاربران واقعی را متوقف و در عملکرد سیستم اختلال ایجاد کند (از جمله حملات سایبری نظیر DoS). همچنین با از کار افتادن یک گره، فقط همان گره خاموش می‌شود و بقیه شبکه بدون نقص به کار خود ادامه می‌دهد (پایداری سیستم). در معماری P2P، واسطه‌ها حذف می‌شوند و کاربران مستقیماً با یکدیگر در ارتباط هستند که علاوه بر افزایش سرعت و کاهش هزینه، منجر به مصونیت سیستم در برابر سانسور نهادها و مراجع کنترل‌کننده نیز می‌شود. یکی از کاربردهای مهم معماری P2P، زنجیره بلوکی (Blockchain) است. در زنجیره بلوکی داده‌ها به صورت بلوکی و با لحاظ موارد امنیتی به زنجیره اضافه می‌شوند، از آنجایی که همه گره‌ها یک نسخه از زنجیره بلوکی را دارند، امکان دستکاری و تغییر داده‌ها وجود ندارد.

پاسخ سوال ۳:

(الف)

ردیف	سرویس	TCP	UDP
۱	انتقال داده با قابلیت اطمینان	✓	×
۲	امنیت	×	×
۳	ارتباط بین یک فرستنده با چند گیرنده (multicast)	×	✓
۴	گذردهی بالا	×	✓
۵	کنترل جریان (flow control)	✓	×
۶	تضمین دریافت داده در زمانی مشخص	×	×
۷	توانایی ارسال پیغام بزرگ	✓	×



(ب)

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 چندرسانه‌ای بهتر است. البته باید به این نکته نیز دقت شود که ممکن است همیشه افزایش سرعت اتفاق نیافتد. مثلاً اگر شبکه دچار ازدحام شده باشد ممکن است تعداد زیادی از بسته‌ها با ارسال بسته‌های UDP از بین بروند. البته باید توجه داشت که دور بودن سرویس‌گیرنده و سرویس‌دهنده باعث افزایش تأخیر در ارتباط‌ها می‌شود و هزینه ایجاد و از بین بردن اتصال را بالا می‌برد.

(ج)

خیر، در مورد تضمین پهنای باند و تأخیر راه حل انتها به انتها وجود ندارد و نیاز است که گره‌های شبکه نیز در این تضمین مشارکت داشته باشند.

پاسخ سوال ۴:

(الف)

یکی از تفاوت‌های بنیادین بین HTTP/1.1 و نسخه‌های قدیمی‌تر HTTP آن است که در نسخه ۱.۱ اتصال به صورت پیش‌فرض مداوم (persistent) در نظر گرفته می‌شود. در اتصالات مداوم مکانیزمی به نام سیگنالی‌نگ قرار داده شده است که در آن کلاینت و سرور می‌توانند بسته شدن اتصال TCP را اعلام نمایند و اتصال را خاتمه دهند. هنگامی که سیگنال بسته شدن دریافت شد، کلاینت نباید درخواست دیگری بر روی آن اتصال ارسال نماید.

(ب)

هیچ‌گونه سرویس رمزنگاری توسط HTTP ارائه نمی‌گردد.

(ج)

خیر، یک کلاینت نباید بیش از دو اتصال با یک سرور یا پراکسی برقرار کند.

(د)

بله، در هر زمان، یک سرویس‌دهنده یا سرویس‌گیرنده یا پراکسی می‌تواند اتصال را قطع کند. به عنوان مثال، فرض کنید سرور اتصال را بی‌کار (idle) تشخیص می‌دهد و اقدام به بستن اتصال می‌کند و در همان زمان نیز سرویس‌گیرنده اقدام به فرستادن درخواست جدید می‌کند. از دید سرویس‌دهنده، اتصال در حالی که بی‌کار بود بسته می‌شود اما از دید سرویس‌گیرنده درخواست جدیدی در حال انجام است و اتصال بی‌کار نیست.

سوال ۵:

(الف)

زمان بدست آوردن آدرس IP:

$$t_{IP_address_resolve} = RTT_1 + RTT_2 + \dots + RTT_n$$

بعد از بدست آوردن آدرس IP، ابتدا باید اتصال TCP برقرار شود و سپس درخواست HTTP ارسال شده و شی دریافت شود.

$$t_{TCP_connection} = RTT_0$$

$$t_{get_web_object} = RTT_0$$

در نتیجه:

$$Delay_{get_base_HTML} = t_{IP_address_resolve} + t_{TCP_connection} + t_{get_web_object}$$

$$Delay_{get_base_HTML} = RTT_1 + RTT_2 + \dots + RTT_n + RTT_0 + RTT_0$$

$$Delay_{get_base_HTML} = 2RTT_0 + RTT_1 + RTT_2 + \dots + RTT_n$$



(ب)

با توجه به اینکه HTTP به صورت غیر مداوم و بدون اتصال موازی عمل می‌کند بنابراین بعد از دریافت base HTML که زمان دریافت آن در بند الف محاسبه شد، برای دریافت هر شی باید یک اتصال TCP باز شود و سپس درخواست ارسال و شی دریافت شود. در نتیجه:

$$Delay_{get_baseHTML_5_object} = 2RTT_0 + RTT_1 + RTT_2 + \dots + RTT_n + 5 \times (RTT_0 + RTT_0)$$

$$Delay_{get_baseHTML_5_object} = 12RTT_0 + RTT_1 + RTT_2 + \dots + RTT_n$$

سوال ۶:

Data Packet Length: $L_d = 100000 \text{ (bit)}$
Control Packet Length: $L_c = 200 \text{ (bit)}$
HTTP Object Length: $L_o = 100000 \text{ (bit)}$
Number of Objects: $K = 10 \text{ (bit)}$
Transmission Rate: $R \text{ (bps)}$
Link Length: $d \text{ (m)}$
Propagation Speed: $V \text{ (m/s)}$
Propagation Delay: $t_{prop} = d/V \text{ (s)}$
Round Trip Time: $RTT = 2t_{prop} \text{ (s)}$

• روش ناپایا (non-persistent) با ۱۰ اتصال موازی:

$$\text{Data Packet Transmission Time for one parallel connection: } t_{d1} = \frac{L_d}{R}$$

$$\text{Control Packet Transmission Time for one parallel connection: } t_{c1} = \frac{L_c}{R}$$

$$\text{Data Packet Transmission Time for } N \text{ parallel connection: } t_{dN} = \frac{L_d}{R/N} = \frac{NL_d}{R} = Nt_{d1}$$

$$\text{Control Packet Transmission Time for } N \text{ parallel connection: } t_{cN} = \frac{L_c}{R/N} = \frac{NL_c}{R} = Nt_{c1}$$

$$ResponseTime_{NonPersistent} = \underbrace{(t_{c1} + RTT + t_{c1})}_{\text{TCP Connection Recieve Base HTML}} + \underbrace{(t_{c1} + RTT + t_{d1})}_{\text{HTTP Request-Response}} + \underbrace{(t_{c10} + RTT + t_{c10})}_{\text{TCP Connection Recieve other 10 HTML Objects}} + \underbrace{(t_{c10} + RTT + t_{d10})}_{\text{HTTP Request-Response}}$$

$$ResponseTime_{NonPersistent} = 4RTT + 3t_{c1} + t_{d1} + 3t_{c10} + t_{d10} = 4RTT + 33t_{c1} + 11t_{d1}$$

$$= 8t_{prop} + \frac{33 \times 200}{R} + \frac{11 \times 100000}{R} = \frac{8d}{V} + \frac{1106600}{R}$$

• روش پایا (persistent):

$$\text{Data Packet Transmission Time: } t_d = \frac{L_d}{R}$$

$$\text{Control Packet Transmission Time: } t_c = \frac{L_c}{R}$$

الف) پایا خط لوله (pipeline):

$$ResponseTime_{PipelinePersistent} = \underbrace{(t_c + RTT + t_c)}_{\text{TCP Connection Recieve Base HTML}} + \underbrace{(t_c + RTT + t_d)}_{\text{HTTP Request-Response}} + \underbrace{(t_c + RTT + 10 \times t_d)}_{\text{HTTP Request-Response Recieve other 10 HTML Objects}}$$

$$ResponseTime_{PipelinePersistent} = 3RTT + 4t_c + 11t_d = 6t_{prop} + \frac{13 \times 200}{R} + \frac{11 \times 100000}{R} = \frac{6d}{V} + \frac{1100800}{R}$$

ب) پایا بدون خط لوله (no_pipeline):

$$ResponseTime_{NoPipelinePersistent} = \underbrace{(t_c + RTT + t_c)}_{\text{TCP Connection Recieve Base HTML}} + \underbrace{(t_c + RTT + t_d)}_{\text{HTTP Request-Response}} + 10 \times \underbrace{(t_c + RTT + t_d)}_{\text{HTTP Request-Response Recieve other 10 HTML Objects}}$$

$$ResponseTime_{NoPipelinePersistent} = 12RTT + 13t_c + 11t_d = 24t_{prop} + \frac{13 \times 200}{R} + \frac{11 \times 100000}{R} = \frac{24d}{V} + \frac{1102600}{R}$$



مقایسه روش ناپایا با پایا خط لوله:

$$ResponseTime_{NonPresistent} - ResponseTime_{PipelinePresistent} = \frac{2d}{V} + \frac{5800}{R}$$

چون اختلاف مثبت است بنابراین تأخیر روش ناپایا بیشتر از تأخیر روش پایا خط لوله است.

مقایسه روش ناپایا با پایا بدون خط لوله:

$$ResponseTime_{NonPresistent} - ResponseTime_{NoPipelinePresistent} = \frac{4000}{R} - \frac{16d}{V}$$

اگر $R < \frac{250}{d/V}$ باشد آنگاه تأخیر روش ناپایا بیشتر از تأخیر روش پایا بدون خط لوله است. و

اگر $R > \frac{250}{d/V}$ باشد آنگاه تأخیر روش ناپایا کمتر از تأخیر روش پایا بدون خط لوله است. و

اگر $R = \frac{250}{d/V}$ باشد آنگاه تأخیر روش ناپایا مساوی با تأخیر روش پایا بدون خط لوله است.

مقایسه روش پایا بدون خط لوله با پایا خط لوله:

$$ResponseTime_{NoPipelinePresistent} - ResponseTime_{PipelinePresistent} = \frac{18d}{V} + \frac{1800}{R}$$

چون اختلاف مثبت است بنابراین تأخیر روش پایا بدون خط لوله بیشتر از تأخیر روش پایا خط لوله است.

پاسخ سوال ۷:

ردیف	پرسش	جواب
الف)	URL صفحه وب درخواستی چیست؟	http://giaia.cs.umass.edu/cs453/index.html
ب)	مرورگر از چه نسخه‌ای از پروتکل HTTP استفاده می‌کند؟	HTTP Version 1.1
ج)	اتصال مداوم (persistent) و غیرمداوم (non-persistent) را به اختصار تعریف کرده و مشخص کنید در این پیغام‌ها از کدام روش استفاده شده است.	از آنجا که connection از جنس keep-alive می باشد بنابراین این از اتصال persistent استفاده شده است
د)	آیا شیء (Object) درخواستی بر روی سرور موجود بوده است؟	بله
ه)	پاسخ در چه زمانی فراهم شده است؟	Tuesday 07 Mar 2008 12:39:45 GMT
و)	آخرین زمان تغییر شیء درخواستی در چه زمانی بوده است؟	Saturday 10 Dec 2005 18:27:46 GMT
ز)	اندازه شیء (Object) درخواستی چقدر است؟	۳۸۷۴ بایت
ح)	آیا سرور با تقاضای نوع اتصال موافق بوده است؟	بله
ط)	از کدام یک از روش‌های HTTP استفاده شده است؟	از متد GET

پاسخ سوال ۸:

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

$$\frac{30 \times 400 \times 10^3}{100 \times 10^6} = 0.12$$

که مطابق نمودار می‌توان از تاخیر آن چشم‌پوشی کرد. بنابراین تفاوتی در محل قرارگیری پروکسی نخواهد بود.

۵۰ درصد درخواست‌ها نیاز به دانلود اطلاعات از وب سرورهای اصلی دارند. بنابراین بار بر روی لینک دسترسی ۱۰ مگابیتی به صورت زیر محاسبه می‌شود:

$$Load\ on\ The\ Internet\ Access\ Link = 0.5 \times \frac{30 \times 400 \times 10^3}{10 \times 10^6} = 0.6$$

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

$$Access_Router_Delay = 0.5\ s$$



$$\begin{aligned} Delay &= Proxy_Delay \times 0.5 + (Access_Router_Delay + Internet_Delay) \times 0.5 \\ &= 0 \times 0.5 + (2 + 0.5) \times 0.5 = 1.25 \text{ s} \end{aligned}$$

پاسخ سوال ۹:

(الف)

پروتکل FTP یکی از پروتکل‌های پرکاربرد برای تبادل فایل بین سرویس‌دهنده و سرویس‌گیرنده در شبکه اینترنت است. این پروتکل بر اساس مدل Client-Server بوده و سرویس‌دهنده یک فضای ذخیره‌سازی (درایو) در اختیار سرویس‌گیرنده (ها) قرار می‌دهد. سرویس‌گیرنده پس از ورود به (log in) سرویس‌دهنده می‌تواند عملیاتی مشابه با عملیات خط فرمان (command line) یونیکس یا ویندوز را انجام دهد. عملیاتی نظیر دریافت لیست فایل‌ها، ایجاد دایرکتوری، تغییر دایرکتوری و موارد مشابه دیگر. از جمله عملیات مهم پروتکل FTP دریافت کردن (Download) و یا کپی کردن (Upload) فایل از یا به سرویس‌دهنده است. پروتکل FTP برای ارسال فرمان‌ها (commands) و ارسال و یا دریافت فایل از دو سوکت TCP همزمان بر روی پورت‌های ۲۱ و ۲۰ استفاده می‌کند. اتصال TCP بر روی سوکت ۲۱ برای تبادل فرمان‌های کنترلی و اتصال TCP بر روی سوکت ۲۰ برای تبادل فایل استفاده می‌شود. جداسازی کانال کنترلی از کانال داده در FTP این مزیت را دارد که همزمان با ارسال و یا دریافت فایل، سرویس‌گیرنده می‌تواند فرمان جدید به سرویس‌دهنده بدهد.

(ب)

به کانال کنترل (سیگنالینگ) مجزا از کانال داده، اصطلاحاً کانال کنترلی خارج بانده (Out-of-band) می‌گویند. در مقابل آن کانال کنترلی داخل بانده (In-band) را داریم که فرمان‌های کنترلی و داده در یک کانال ارسال می‌شوند. مزیت کانال کنترلی داخل بانده (In-band) نسبت به خارج بانده (Out-of-band) این است که منابع کمتری استفاده می‌کند و برای سرویس‌هایی نظیر سرویس تلفن که بعد از برقراری ارتباط سرویس‌گیرنده فرمان‌های زیادی ارسال نمی‌کند، مناسب هستند.

در کانال کنترلی خارج بانده (Out-of-band) منابع بیشتری برای ایجاد دو کانال ارتباطی، یکی برای کنترل و یکی برای داده، استفاده می‌شود اما مزیت کانال کنترلی خارج بانده (Out-of-band) نسبت به کانال کنترلی داخل بانده (In-band) این است که سرویس‌گیرنده همزمان با دریافت سرویس می‌تواند درخواست‌های جدیدی ارسال نماید و سرویس‌دهنده به موازات به این درخواست‌ها پاسخ دهد. کانال کنترلی خارج بانده (Out-of-band) در کاربردهایی نظیر FTP که معمولاً کاربر همزمان با دریافت سرویس، فرمان‌هایی برای مدیریت سرویس و یا درخواست سرویس جدید ارسال می‌کند مناسب است.

پاسخ سوال ۱۰:

MTA مخفف Mail Transfer Agent یا عامل انتقال ایمیل است. یک میزبان، ایمیل خود را به یک MTA می‌فرستد. سپس آن ایمیل دنباله‌ای از MTAها را دنبال می‌کند تا به گیرنده برسد. همانطور که مشاهده می‌کنید، این ایمیل spam، زنجیره‌ای از MTAها را دنبال می‌کند. یک MTA درستکار باید گزارش کند که از کجا ایمیل را دریافت کرده است. در این ایمیل، MTA با نام "asus-4b96 (58.88.21.177)" جایی که این ایمیل را دریافت کرده است را گزارش نمی‌کند. از آنجایی که فرض بر این است که تولید کننده ایمیل مغرض است، بنابراین، میزبان مغرضی که این ایمیل را تولید کرده است، میزبان "asus-4b96 (58.88.21.177)" است.

پاسخ سوال ۱۱:

برای محاسبه حداقل زمان توزیع فایل در معماری Client-Server از رابطه زیر استفاده می‌شود:

$$D_{CS} = \max \{ NF/u_s, F/d_{min} \}$$

و برای محاسبه حداقل زمان توزیع فایل در معماری نظیر به نظیر از رابطه زیر استفاده می‌شود:

$$D_{P2P} = \max \left\{ F/u_s, F/d_{min}, NF / \left(u_s + \sum_{i=1}^N u_i \right) \right\}$$



طبق صورت سؤال داریم:

$$F = 1 \text{ Gbits} = 1 * 1024 \text{ Mbits}$$

$$u_s = 30 \text{ Mbps}$$

$$d_{min} = d_i = 2 \text{ Mbps}$$

(دقت کنید کیلو (K) در حجم داده $2^{10} = 1024$ ، ولی در نرخ ارسال داده‌ها $10^3 = 1000$ است.)

N	D_{CS}			D_{P2P}		
	10	100	1000	10	100	1000
$u = 300 \text{ Kbps}$	536.9	3579.1	35791.4	536.9	1789.6	3253.8
$u = 700 \text{ Kbps}$	536.9	3579.1	35791.4	536.9	1073.7	1470.9
$u = 2 \text{ Mbps}$	536.9	3579.1	35791.4	536.9	536.9	536.9

سوال ۱۲:

(الف)

توزیعی فایلی را در نظر بگیرید که سرویس‌دهنده به صورت موازی و با نرخ μ_s/N فایلی را به سمت هر سرویس‌گیرنده ارسال می‌کند. با توجه به فرض داده شده $(\mu_s/N \leq d_{min})$ ، پس نرخ ارسال سرویس‌دهنده (μ_s/N) از نرخ دالود هر سرویس‌گیرنده (d_{min}) کمتر است. بنابراین هر سرویس‌گیرنده حداکثر با نرخ μ_s/N فایلی را دریافت می‌کند. از آنجایی که همه سرویس‌گیرنده‌ها به طور همزمان با نرخ دریافت μ_s/N فایلی با اندازه F را دریافت می‌کنند، پس تأخیر (زمان) دریافت فایلی برابر است با:

$$D_{CS} = \frac{F}{\mu_s/N} = \frac{NF}{\mu_s} \quad (1)$$

(ب)

توزیعی فایلی را در نظر بگیرید که همه سرویس‌گیرندگان به صورت موازی و با نرخ d_{min} فایلی را از سرویس‌دهنده دریافت می‌کنند. با توجه به فرض داده شده $(d_{min} \leq \mu_s/N)$ ، پس مجموع نرخ دریافت همه سرویس‌گیرنده‌ها از نرخ ارسال سرویس‌دهنده کمتر است $(Nd_{min} \leq \mu_s)$. بنابراین هر سرویس‌گیرنده حداکثر با نرخ d_{min} فایلی را دریافت می‌کند. از آنجایی که همه سرویس‌گیرنده‌ها به طور همزمان با نرخ دریافت d_{min} فایلی با اندازه F را دریافت می‌کنند، پس تأخیر (زمان) دریافت فایلی برابر است با:

$$D_{CS} = \frac{F}{d_{min}} \quad (2)$$

(ج) با در نظر گرفتن نتایج بدست آمده در بخش‌های (الف) و (ب):

$$\text{From (1): if } \mu_s/N \leq d_{min} \text{ then } D_{CS} = \frac{F}{\mu_s/N} = \frac{NF}{\mu_s}$$

$$\text{From (2): if } \mu_s/N \leq d_{min} \text{ then } D_{CS} = \frac{F}{d_{min}}$$

در نتیجه:

$$D_{CS} = \max \{NF/\mu_s, F/d_{min}\}$$

سوال ۱۳:

(الف)

$u = u_1 + u_2 + \dots + u_N$ را تعریف می‌کنیم، با فرض داده شده:

$$u_s \leq (u_s + u)/N \Rightarrow$$

$$(N-1)u_s \leq u \quad (\text{رابطه ۱})$$

در طرح توزیع، این فایلی را به N قسمت به گونه‌ای تقسیم می‌کنیم که اندازه قسمت i ام برابر $(u_i/u)F$ باشد. سرویس‌دهنده قسمت i ام را با نرخ $r_i = (u_i/u)u_s$ به نظیر (peer) i ارسال می‌کند. دقت کنید که $r_1 + r_2 + \dots + r_N = u_s$ است. در نتیجه مجموع نرخ سرویس‌دهنده از نرخ ارسال



لینک سرویس دهنده بیشتر نمی شود. همچنین نظیر i بیت های دریافتی را با نرخ ارسال r_i به $N - 1$ نظیر دیگر ارسال می کند. مجموع نرخ ارسال نظیر i نام برابر است با $(N - 1)r_i$. همچنین با در نظر گرفتن رابطه ۱ داریم:

$$(N - 1)r_i = (N - 1)(u_s u_i / u) = ((N - 1)u_s / u)u_i \leq u_i$$

بنابراین مجموع نرخ ارسال نظیر i کمتر از پهنای باند لینک ارسال آن (u_i) است.

با این طرح توزیع فایل، نرخ دریافت نظیر i نام برابر است با:

$$r_i + \sum_{\substack{j=1, \\ j \neq i}}^N r_j = u_s$$

بنابراین هر نظیر این فایل را در زمان F/u_s دریافت می کند.

(ب)

مجدداً $u = u_1 + u_2 + \dots + u_N$ را تعریف می کنیم، با فرض داده شده:

$$u_s \geq (u_s + u)/N$$

در نظر بگیرید:

$$r_i = u_i / (N - 1)$$

$$r_{N+1} = (u_s - u / (N - 1)) / N$$

در طرح توزیع، این فایل را به $N + 1$ قسمت تقسیم می کنیم. سرویس دهنده به صورت موازی هر قسمت نام را با نرخ r_i به نظیر (peer) نام ارسال می کند. هر نظیر i بیت های دریافتی را با نرخ ارسال r_i به $N - 1$ نظیر دیگر ارسال می کند. علاوه بر آن، سرویس دهنده بیت های قسمت $N + 1$ را با نرخ r_{N+1} به همه N نظیر ارسال می کند. نظیرها قسمت $N + 1$ را ارسال نمی کنند.

مجموع نرخ ارسال سرویس دهنده برابر است با:

$$r_1 + r_2 + \dots + r_N + Nr_{N+1} = u / (N - 1) + u_s - u / (N - 1) = u_s$$

بنابراین مجموع نرخ ارسال سرویس دهنده از پهنای باند لینک ارسال سرویس دهنده u_s بیشتر نمی شود.

مجموع نرخ ارسال نظیر i نام برابر است با:

$$(N - 1)r_i = u_i$$

بنابراین مجموع نرخ ارسال هر نظیر از پهنای باند لینک ارسال آن (u_i) بیشتر نمی شود.

در این طرح توزیع فایل، نرخ دریافت نظیر i نام برابر است با:

$$r_i + r_{N+1} + \sum_{\substack{j=1, \\ j \neq i}}^N r_j = u / (N - 1) + (u_s - u / (N - 1)) / N = (u_s + u) / N$$

بنابراین هر نظیر این فایل را در زمان $NF / (u_s + u) = NF / (u_s + u_1 + u_2 + \dots + u_N)$ دریافت می کند.

در این طرح، برای سادگی از تعیین اندازه هر قسمت صرف نظر کردیم. حال در اینجا اندازه هر قسمت را نیز تعیین می کنیم. فرض کنید مدت زمان ارسال هر قسمت τ باشد، اگر اندازه هر قسمت را با F_i نشان دهیم، آنگاه:

$$F_i = r_i \tau = \frac{u_i \tau}{N - 1}, \quad \text{for } i = 1, 2, \dots, N$$

$$\text{and } F_{N+1} = r_{N+1} \tau = \frac{((N - 1)u_s - u) \tau}{N(N - 1)}$$

از طرف دیگر:

$$F_1 + F_2 + \dots + F_N + F_{N+1} = F$$

با جایگذاری مقادیر F_i در رابطه بالا مدت زمان ارسال هر قسمت به صورت زیر بدست می آید:



$$\tau = \frac{NF}{u_s + u}$$

در نتیجه:

$$F_i = r_i \tau = \left(\frac{(N/(N-1))u_i}{u_s + u} \right) F, \quad \text{for } i = 1, 2, \dots, N$$

$$\text{and } F_{N+1} = r_{N+1} \tau = \left(\frac{u_s - u/(N-1)}{u_s + u} \right) F$$

(ج)

با در نظر گرفتن نتایج بدست آمده در بخش‌های (الف) و (ب):

$$(الف): \text{ if } u_s \leq (u_s + u_1 + \dots + u_N)/N \text{ then } D_{P2P} = F/u_s$$

$$(ب): \text{ if } u_s \geq (u_s + u_1 + \dots + u_N)/N \text{ then } D_{P2P} = NF/(u_s + u_1 + u_2 + \dots + u_N)$$

در نتیجه:

$$D_{P2P} = \max \{ \{F/u_s, NF/(u_s + u_1 + \dots + u_N)\} \}$$