-به نام او

کامیار جان‌پروری – ۹۶۲۶۰۱۵ تمرین سری سوم شبکه‌های کامپیوتری

سؤال اول-

این سؤال را با کمک مثالی پاسخ می‌دهیم:

فرض کنیم دو خانه یکی در یزد و دیگری در کرمان وجود دارد، در هر خانه چند نفر زندگی می‌کنند (افراد این دو خانه فامیل اند) افراد این دو خانه برای یک‌دیگر نامه می‌نویسند و از طریق پست برای هم ارسال می‌کنند. در هر خانه یکی از افراد صندوق پستی را هر روز چک کرده و به فرد مورد نظر می‌دهد. از دید افراد داخل خانه شخصی که پیام ها را می‌آورد و می‌برد این دو نفر اند نه سیستم پستی!

سیستم OSI هم مشابه این است! لایه network مشابه سیستم پستی و لایه transport معادل آن دو نفر و لایه application نامه داخل بسته پستی است. حال اگر سیستم پستی بسته‌های حاوی نامه را در زمان مناسب و به درستی منتقل نکنند از دید افراد داخل خانه مشکل آن دو نفر اند.

در لایه شبکه نیز اگر پهنای باند و تأخیر فراهم نشود منطقا از دید لایه application لایه transport نتوانسته سرویس را به درستی انجام دهد.

سؤال دوم-

بله، در بعضی موارد بسیار معقول است. در کشور ما نیز مشاهده می‌شود برای مثال filimo یک گره در pars online قرار می‌دهد.

در این صورت پهنای باند خارجی این ISP به آن به صفر می‌رسد. یعنی content delivery بسیار بهبود می‌یابد. برای ISP نیز مدیریت ترافیک بهتری در جریان‌های داخل شبکه‌ها حاصل می‌شود. این دو نیز می‌توانند با همکاری هم application های جدیدی به کاربران ارائه کنند که باعث یک رابطه‌ی برد-برد بین آن‌ها می‌شود. اما از طرفی ایجاد ساختارهای آن هزینه بردار است.

سؤال سوم-

زمان بدست آوردن IP : $RTT_1 + RTT_2 + \cdots + RTT_n$

بعد از این‌که IP را بدست آوردیم، با یک RTT_0 یک ارتباط TCP راه‌اندازی می‌شود. سپس با یک RTT_0 دیگر درخواست و پاسخ آن شیئ انجام می‌شود. پس نتیجه‌ی کلی :

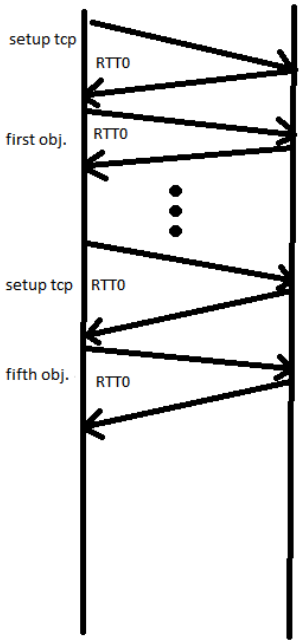
$$RTT_1 + RTT_2 + \cdots + RTT_n + 2\,RTT_0$$

سؤال چهارم-

a) $RTT_1 + RTT_2 + RTT_3 + 12\,RTT_0$

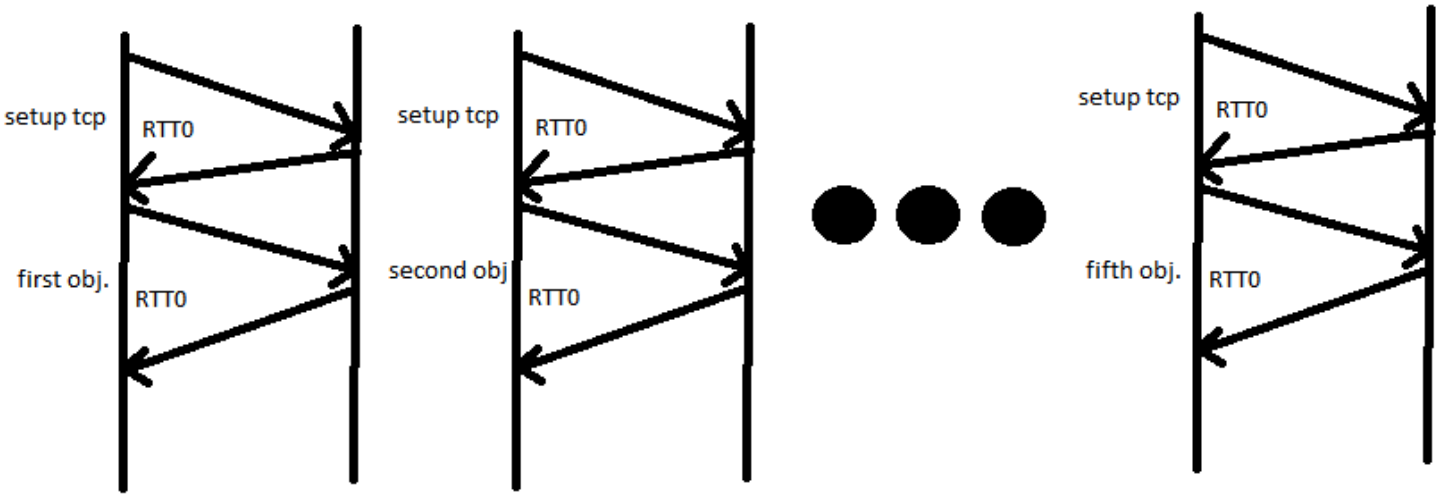
۲ تا RTT0 مربوط به setup connection و دریافت HTML

و سپس ۱۰ تای باقی‌مانده مربوط به دریافت این objectها اند.

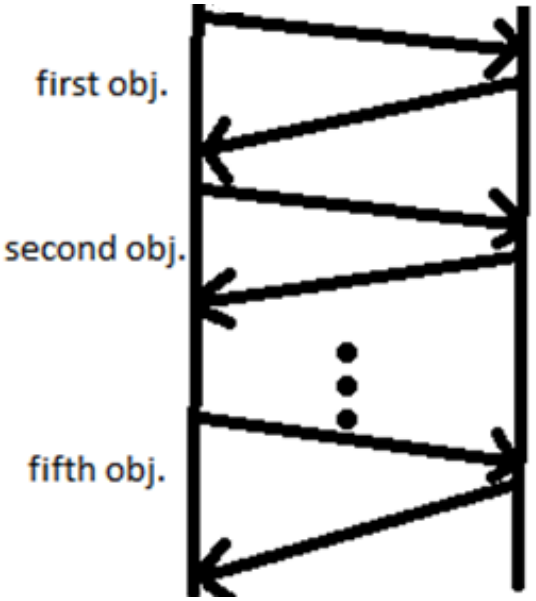


b) $RTT_1 + RTT_2 + RTT_3 + 4\,RTT_0$

۲ RTT0 مربوط به setup connection و دریافت HTML و سپس دریافت objectها به‌طور هم‌زمان انجام می‌شود.



c) $RTT_1 + RTT_2 + RTT_3 + 7\,RTT_0$



2 تا RTT0 برای برقراری ارتباط و دریافت HTML سپس به دلیل این‌که ارتباط پایا است در ادامه به ترتیب objectها دریافت می‌شود.

سؤال پنجم –

نبود!

سؤال ششم-

فرض شده که اندازه پیام‌های HTTP request ناچیز باشد که ترافیکی در access link ایجاد نکند.

و traffic intensity در LAN هست:

(30 requests/sec)·(0.4 Mbits/request)/(100 Mbps)=0.12

و traffic intensity در access link هست:

(30 requests/sec)·(0.4 Mbits/request)/(10Mbps)= 1.2

چون proxy به ۵۰ درصد پاسخ می‌دهد، ۵۰ درصد باقی‌مانده از سرورهای اینترنت سرویس می‌گیرند. حال از آن ۵۰ درصدی که از proxy سرویس می‌گیرند ۲۰ دوباره باید به سرور های اینترنت متصل شوند یعنی:

یعنی در ۶۰ درصد حالات نتیجه را از اینترنت می‌گیرد. (۵۰ درصد که مستقیماً گفته شده و ۱۰ درصد دیگر در صورت نتیجه‌ی نادرستی که از proxy آمده است.

پس با این اوصاف router load را به صورت زیر محاسبه می‌کنیم:

Router load $=0.6 \times 1.2 = 0.72$

پس طبق نمودار تأخیر تقریباً 1.1 است.

$$0.5 \times (D_{router} + D_{internet}) + 0.1 \times (D_{proxy} + D_{router} + D_{internet}) + 0.4 \times (D_{proxy})$$
$$= 0.5 \times (3.1) + 0.1 \times (0 + 3.1) + 0.4 \times 0 \qquad = 1.86s$$

سؤال هفتم-

به خاطر virtual host ها. امروزه متداول است که از یک سرور میزبان (host) چندین domain/site بود. که HTTP 1.1 این را با host header محیا می‌سازد. اگر از HTTP 1.0 استفاده شود دیگر نیازی به این نیست.

سؤال هشتم-

راهکاری به نام DNS balance load وجود دارد که با استفاده از آن می‌توان درخواست‌های client ها به یک دامنه را روی چندین سرور پخش کرد. یعنی در DNS چندین سرور را با یک host name تعریف کرد (برای مثال یک host name می‌تواند هم مربوط به یک website باشد هم یک mailing system و سرویس‌های دیگر.

DNS Balance Load درخواست‌های client ها را با تکنیک‌های به طور بهینه بر روی این سرور ها پخش می‌کند. این تکنیک ها عبارتند از: (در اینجا احتمالا از تکنیک دوم استفاده می‌شود.)

۱- Backup server : یک instance از دامنه ساخته می‌شود تا به عنوان یک DNS ثانویه عمل کند و DNS اصلی ممکن است در زمان اجرا ترافیک را روی این سرور redirect کند.

۲- **Round robin DNS-based load sharing** : DNS درخواست ها را به صورت round robin بین سرور ها می‌چرخواند.

۳- **Dynamic DNS load balancing**: DNS به طور پویا درخواست‌ها را روی سروری که منابع بیشتر و load کمتر دارد rout می‌کند.

سؤال نهم –

شرکت های بزرگ سرورهایی را با کپی از داده ها در مناطق جغرافیایی استراتژیک در سراسر جهان راه اندازی می کنند. به این CDN گفته می شود و به این سرورها edge-server گفته می شود ، زیرا آنها نزدیکترین شبکه این company به end-user هستند.

هنگامی که مرورگر درخواست DNS را برای نام دامنه ای که توسط CDN به کار گرفته شده است انجام می دهد ، فرایند کمی متفاوت از سایت های کوچک و یک IP است. سرور مسؤول handle کردن درخواست DNS برای نام دامنه ، به request دریافت شده برای تعیین بهترین مجموعه از سرورها برای handle کردن آن نگاه می کند. در ساده ترین حالت ، سرور DNS یک جستجوی جغرافیایی را بر اساس آدرس IP مربوط به DNS resolver انجام می دهد و سپس آدرس IP را برای edge-server ای که از لحاظ فیزیکی به آن ناحیه نزدیک است ، بازمی‌گرداند.

سؤال دهم–

دلیل این امر همان دلیلی است که در سؤال هشتم گفته شد.

سرویس DNS از یک الگوریتم round-robin استفاده می‌کند. که ترتیب لیستی که DNS باز می‌گرداند بسته به عملکرد این round-robin است. معمولا client تلاش می‌کند به اولین آدرسی که بازگردانده شده متصل شوند. و منطقی است که در درخواست‌های بعدی DNS برای مدیریت درخواست‌ها و پخش آن‌ها لیست را با ترتیب دیگری بازمی‌گرداند.

سؤال یازدهم

از آن‌جا که فایل HTML در سرور۱ قرار دارد پس ابتدا یک TCP connection با سرور۱ برقرار می‌شود سپس فایل HTML دریافت می‌شود:

Getting HTML FILE : 0.03 + (0.03 + 5000/80000) = 0.1225

با این‌که ارتباط پایا است ولی منطقاً چون client خبر ندارد که دو شیئ دیگر در سرور۱ است پس connection بسته می‌شود و وقتی دید ۲ شیئ دیگر در آن سرور است دوباره ارتباط برقرار می‌شود.

سپس از آن‌جا که به هر سه سرور می‌توان هم‌زمان متصل شد، ابتدا یک TCP connection بین client و هر سه سرور connection بر قرار می‌شود که برای هر کدام یک RTT طول می‌کشد. سپس برای هر شیئ ابتدا یک RTT سپس دریافت شیئ انجام می‌شود.

Server1: 0.03 + (0.03 + 2000/80000) + (0.03 + 4000/80000) = 0.165 s

Server2: 0.04 + (0.04 + 2000/40000) + (0.04 + 4000/40000) = 0.27 s

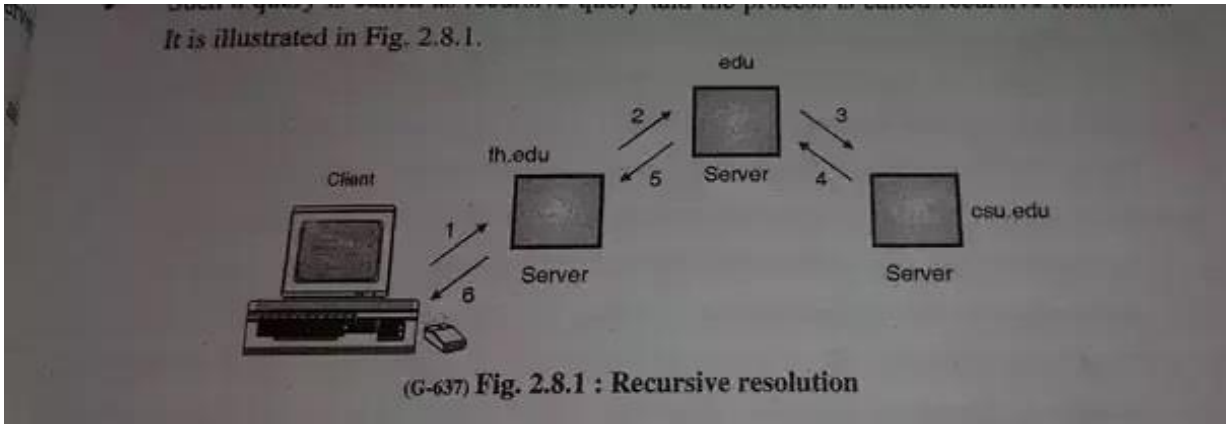
Server3: 0.02 + (0.02 + 5000/80000) + (0.02 + 7000/80000) = 0.21s

چون این‌ها هم‌زمان انجام می‌شود پس زمان کلی دریافت برابر است با:

0.1225 + 0.27 = 0.3925 s

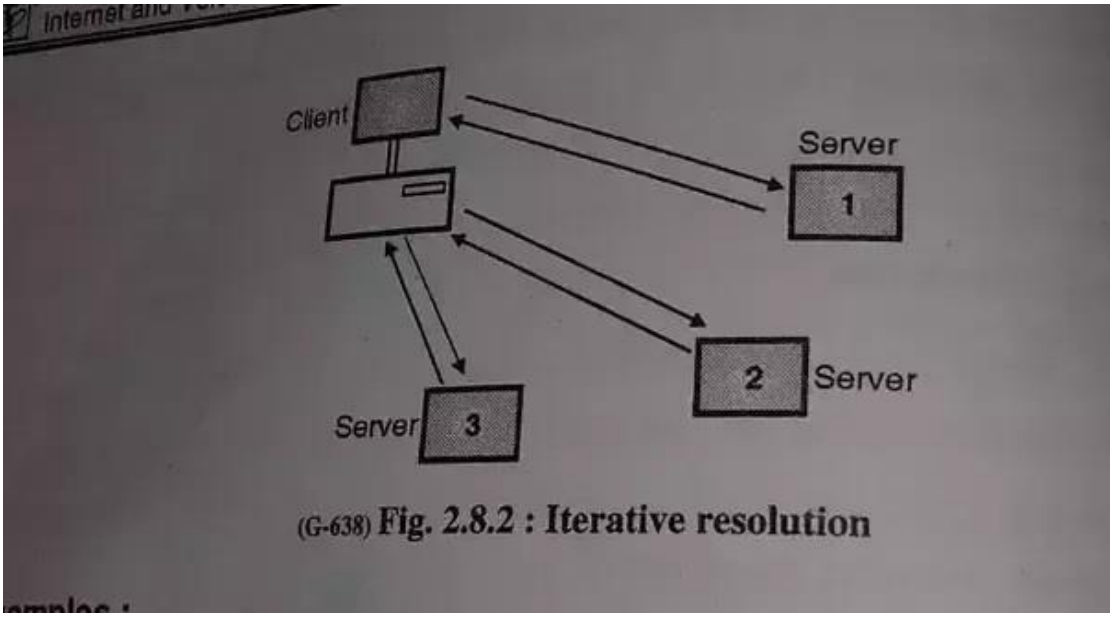
سؤال دوازدهم-

روش recursive در عکس زیر مشاهده می‌شود:



همان‌طور که مشخص است client درخواست خود را به اولین سرور (در اینجا th.edu) می‌فرستد و اگر نام دامنه authorized بود، پاسخ می‌دهد. در غیر این صورت درخواست را به سرور بعدی منتقل می‌کند و این کار ادامه پیدا می‌کند تا سرور درست آن درخواست را resolve کند و از طریق همین مسیر پاسخ به client باز می‌گردد.

روش iterated در عکس زیر نمایش داده شده:



در این روش client درخواست خود را به اولین سرور می‌دهد، اگر authorized بود پاسخ داده می‌شود در غیر این صورت آدرس IP سرور بعدی که می‌تواند درخواست را resolve کند، برگردانده می‌شود. حالا client به سرور بعدی درخواست می‌دهد و این پروسه تا یافتن سرور دست ادامه پیدا می‌کند. (این کارها به جای این‌که توسط client انجام شود می‌تواند توسط یک DNS server انجام شود)

سؤال سیزدهم-

از آنجا که FTP از دو اتصال موازی مختلف برای انتقال file استفاده می کند ، به همین دلیل گفته می شود "خارج از باند" است.

به طور موازی از یک اتصال کنترلی و یک اتصال داده استفاده می کند. اتصال کنترل برای ارسال اطلاعاتی مانند شناسه کاربر ، گذرواژه ، دستورات "PUT" و "GET" file استفاده می شود. از اتصال داده برای ارسال file استفاده می شود. به دلیل این اتصال کنترل (جداگانه) FTP "خارج از باند" است.

سؤال چهاردهم-

- The MAIL FROM : در پروتکل SMTP یک پیام است از SMTP client که فرستنده‌ی mail message را به سرور SMTP مشخص می‌کند.
- THE FROM : در mail message خودش یک پیام SMTP نیست بلکه یک خط در بدنه‌ی mail message است.

نکات:

- Domain فرستنده آدرس From ممکن است با domain مربوط به Mail From یکسان نباشد.
- در حین authentication فریم ورک SPF دامنه مربوط به آدرس MFrom را چک می‌کند، نه آدرس دامنه From.
- هر دو آدرس را می‌توان جداگانه در قسمت header مربوط به email یافت.

سؤال پانزدهم-

DNS به طور پیش‌فرض از UDP روز پورت ۵۳ استفاده می‌کند ولی زمانی که سایز درخواست یا پاسخ بیشتر از سایز یک بسته است (مانند پاسخ‌هایی که recordهای زیادی دارند یا پاسخ های IPv6 یا بیشتر پاسخ‌های DNSSEC

ماکزیموم سایز در اصل ۵۱۲ بایت است ولی extensionی در پروتکل DNS وجود دارد که به client ها اجازه می‌دهد بتوانند از پروتکل UDP پاسخ را تا سایز 5096 بایت بفرستند.

پاسخ‌های DNSSEC معمولا بزرگ‌تر از ماکزیموم سایز UDP اند. همچنین transfer request ها نیز بزرگتر از ماکزیموم سایز UDP اند. پس این‌ها با پروتکل TCP انجام می‌شوند.

سؤال شانزدهم-

الف)

در اطلاعات داده شده، مشاهده می‌شود که در Type مربوط به MX دو سطر وجود دارد mail.domain.com و mail2.domain.com که این دو سرور مربوط به mailing system این دامنه اند.

حال در Type مربوط به CName مشاهده می‌شود که mail و mail2 به ترتیب نام مستعار برای server1 و server 2 اند.

و در Type مربوط به A مشاهده می‌شود که این server1 و server2 مربوط به دو آدرس 10.0.1.5 و 10.0.1.7 اند.

پس ایمیل‌ها به این دو IP Address ارسال می‌شوند.

ب)

در Type مربوط به NS دو سطر دیده می‌شود dns1.domain.com و dns2.domain.com که در Type A آدرس این‌دو به ترتیب 10.0.1.2 و 10.0.1.3 آورده شده است.

پس رکورد ها در این دو ذخیره می‌شوند.