موسسه بابان

انتشارات بابان و انتشارات راهیان ارشد

درس و کنکور ارشد

شبکه‌های کامپیوتری

(حل تشریحی سوالات دولتی ۱۳۹۲)

ویژه‌ی داوطلبان کنکور کارشناسی ارشد مهندسی کامپیوتر و IT

براساس کتب مرجع

کراس راس و لئون گارسیا

ارسطو خلیلی‌فر

کلیه‌ی حقوق مادی و معنوی این اثر در سازمان اسناد و کتابخانه‌ی ملی ایران به ثبت رسیده است.

تست‌های سال ۹۲

۱- اگر از روش سرکشی (Polling) برای به اشتراک گذاری پیوندی با سرعت ۱ مگابیت بر ثانیه بین ۱۰۰ ایستگاه استفاده شده باشد. با فرض اینکه به طور متوسط در هر دور سرکشی ۴۰ درصد ایستگاه‌ها فعال هستند و زمان سرکشی هر ایستگاه ۱۰ درصد زمان ارسال یک فریم است. متوسط ظرفیت ارسال هر ایستگاه بر حسب کیلو بیت بر ثانیه چقدر است؟

۱) ۸ ۲) ۲۲/۵ ۳) ۹ ۴) ۲۵

۲- در محاسبه زمان Timeout در پروتکل TCP اگر مقادیر قبلی t_{RTT} و d_{RTT} به ترتیب برابر با ۹۶ و ۲۰ میلی ثانیه باشند و آخرین زمان رفت و برگشت ۱۲۰ میلی ثانیه باشد، مقدار جدید Timeout محاسبه شده چند میلی ثانیه است؟

۱) ۱۱۹ ۲) ۱۸۰ ۳) ۱۴۴ ۴) ۲۰۱

۳- همه موارد زیر، در خصوص روش‌های کنترل ازدحام پیشگیرانه (Preventive) یا حلقه باز صحیح است بجز :

۱) نیاز به روش‌های کنترل پذیرش اتصال (CAC) است.
۲) نیاز به روش‌های کنترل و نظارت بر ترافیک (Traffic Policing) است.
۳) نیاز به روش‌های دریافت بازخورد (Feedback) است.
۴) بهتر است از روش‌های شکل‌دهی ترافیک (Traffic Shaping) استفاده شود.

۴- فرض کنید شخصی در مرورگر وب خود روی یک لینک برای دریافت یک صفحه وب کلیک می‌کند. اگر آدرس IP مربوط به این URL در میزبان به صورت محلی وجود داشته باشد و فایل HTML مرتبط با این لینک دارای هشت Object باشد، در صورتی که زمان رفت و برگشت بین سرویس گیرنده و سرویس‌دهنده ۱۰۰ میلی ثانیه و زمان ارسال Objectها ناچیز باشد، به ترتیب با استفاده از پروتکل Non-persistent HTTP و HTTP Persistent از زمانی که شخص روی لینک کلیک می‌کند تا زمانی که صفحه وب را به طور کامل دریافت می‌کند بر حسب میلی ثانیه چقدر طول می‌کشد؟

۱) ۱۸۰۰ و ۹۰۰ ۲) ۹۰۰ و ۳۰۰ ۳) ۱۸۰۰ و ۳۰۰ ۴) ۳۰۰ و ۱۰۰

۵- دو گره که از طریق یک پیوند ارتباطی یا پهنای باند ۱ مگابیت بر ثانیه و تأخیر انتشار ۱۳۰ میلی‌ثانیه به هم متصل هستند. برای کنترل خطا از روش Goback N ARQ با شماره ترتیب ۳ بیتی استفاده می‌کنند. اگر اندازه هر فریم ۲۵۰۰ بایت و نرخ هر فریم ۰/۰۰۰۱ باشد آنگاه حداکثر نرخ ارسال مؤثر در این پیوند بر حسب کیلو بیت بر ثانیه تقریباً برابر است با :

۱) ۱۰۰۰ ۲) ۵۰۰ ۳) ۷۵۰ ۴) ۲۵۰

۶- کدام یک از گزینه‌های زیر نادرست است. پروتکل مسیریابی OSPF .....

۱) هزینه مسیر را بر اساس تعداد گام تعیین می‌کند.

۲) بر اساس الگوریتم وضعیت پیوند کار می‌کند.

۳) از نوع پروتکل‌های درون ناحیه است.

۴) از نوع پروتکل‌های IGP است.

پاسخ تست‌های سال ۹۲

۱- گزینه (۱) صحیح است.

در کلید اولیه این سؤال، سازمان سنجش گزینه ۱ را اعلام نمود. اما در کلید نهایی، این سؤال به دلیل اطلاعات ناقص صورت مسأله و وجود ابهام حذف گردید.

به هر حال می‌توان راه حل زیر را برای این سؤال در نظر گرفت:

بر اساس صورت مسأله، در هر دور تنها ۴۰ ایستگاه به اندازه T_F (زمان انتقال فریم) کانال را مشغول نگه می‌دارند (۴۰٪ از ۱۰۰ ایستگاه، یعنی ۴۰ ایستگاه) ولی بنابر روش سرکشی، عملیات سرکشی برای هر ۱۰۰ ایستگاه اتفاق می‌افتد و بنابراین هر ۱۰۰ ایستگاه Poll خواهند شد. از طرفی بنابر صورت مسأله، $T_{Poll}=\frac{۱}{۱۰}T_F$، بنابراین داریم:

$$T_{total}=۴۰\times T_F+۱۰۰\times T_{poll} \xrightarrow{T_{poll}=\frac{۱}{۱۰}T_F} T_{total}=۴۰T_F+۱۰T_F=۵۰T_F$$

پس از زمان کل برابر با $۵۰T_F$، $۴۰T_F$ آن مفید و $۱۰T_F$ آن غیر مفید است، بنابراین بر اساس رابطه بهره‌وری داریم:

$$U=\frac{\text{مفید}}{\text{غیرمفید + مفید}}=\frac{۴۰T_F}{۴۰T_F+۱۰T_F}=\frac{۴۰T_F}{۵۰T_F}=۰/۸$$

حال نرخ بیتی هر ایستگاه را به دست می‌آوریم:

ایستگاه	نرخ انتقال		
۱۰۰	۱Mbps	→	$R=۱۰kbps$
۱	R		

بنابراین بر اساس رابطه گذردهی، نرخ مؤثر هر ایستگاه به صورت زیر محاسبه می‌شود:

$$R_e=\text{Throughput}=R.U=۱۰kbps\times ۰/۸=۸kbps$$

۲- گزینه (۲) صحیح است.

تعیین نرخ زمان timeout خیلی می‌تواند در کارایی شبکه تأثیرگذار باشد، در صورت انتخاب نادرست آن، ممکن است کارایی پروتکل خیلی کاهش یابد.

یادآوری: timeout برای این استفاده می‌شود که اگر داده‌ی ارسالی با خطا روبرو شود، فرستنده بعد از مدت زمان مشخص که Ack یا NACK را دریافت نکرد timeout کند و ارسال مجدد را انجام دهد.

اگر timeout خیلی کوچک باشد، ارسال مجددهای بیهوده خواهیم داشت.

اگر timeout خیلی زیاد باشد، ممکن است داده از بین رفته باشد و خیلی صبر کند تا مجدداً آن را ارسال کند که در این مدت زمان صبر، ظرفیت کانال دارد از دست می‌رود.

☜ ***نکته:*** بهترین زمان انتخاب timeout برابر مدت زمان بین یک رفت و برگشت است.

☜ ***نکته:*** در شبکه‌های کامپیوتری چهار نوع تأخیر داریم:
تأخیر ارسال (T_F)، تأخیر انتشار (T_p)، تأخیر صف (T_q)، تأخیر پردازش $(T_{process})$.

☜ ***نکته:*** تأخیر صف‌بندی داخل گره‌ها، یک تأخیر متغیر است که به حجم ترافیک لحظه عبور از آن گره بستگی دارد.
پس تأخیری که از ابتدا به انتها ایجاد می‌شود، متغیر است و از قبل قابل پیش‌بینی نیست.

مثال: مثلاً دسترسی به سیستم آموزشی (پرتال)
۱- دیدن پرتال از داخل دانشگاه از طریق شبکه محلی (تأخیر در حد ۱ ms)
۲- دیدن پرتال از خانه از طریق اینترنت (هنوز در شبکه داخل کشور) (تأخیر در حد ۱۰ms)
۳- دیدن پرتال از اروپا (تأخیر در حد ۱۰۰ms)
همه برنامه‌های فوق یک کاربرد را دارند و همه از پروتکل یکسان TCP استفاده می‌کنند ولی تأخیرها متفاوت است.

☜ ***نکته مهم:*** timeout باید بر اساس زمان رفت و برگشت لحظه‌ای (که الان دارد اتفاق می‌افتد) تعیین شود، نه این که از قبل تعیین شود، چون که نمی‌دانیم که برنامه کاربردی از کجا دارد چرا می‌شود.

☜ ***نکته:*** تأخیر رفت و برگشت یک متغیر تصادفی است.

☜ ***نکته:*** برای هر متغیر تصادفی یک میانگین و یک انحراف معیار تعریف می‌شود.

☜ ***نکته:*** به پراکندگی نسبت به میانگین، انحراف معیار گفته می‌شود.

محاسبه timeout

بر اساس سند RFC۲۹۸۸ بهترین روش محاسبه timeout بر اساس رابطه‌ی زیر می‌باشد.

$$T_{timeout} = t_{RTT} + kd_{RTT}$$

مقدار K، بستگی به تابع توزیع و کاربرد دارد. معمولاً مقداری برابر۳ یا ۴ است. بر اساس سند RFC۲۹۸۸ مقدار K عدد ۴ در نظر گرفته شده است. پس داریم:

$$\boxed{T_{timeout} = t_{RTT}(new) + ۴ \times d_{RTT}(new)}$$

تخمین RTT:

$$\boxed{t_{RTT}(new) = \alpha\, t_{RTT}(old) + (۱-\alpha)\, T_n \quad ۰<\alpha<۱}$$

T_n: زمان رفت و برگشت اتفاق افتاده در مرحله‌ی n ام

α: نشان می‌دهد که نسبت به پدیده‌ی رفت و برگشت لحظه‌ای که الان اتفاق افتاده است برای T_n چه وزنی را نسبت به داده‌های قبلی قائل هستیم.

مثال: از شما می‌پرسند، معدل کل شما در ۶ ترم گذشته چند می‌شود؟
معدل این ترم ۱۶ و میانگین ۵ ترم گذشته ۱۵/۵.

$$\text{معدل کل} = \frac{۵}{۶} \times ۱۵/۵ + \frac{۱}{۶} \times ۱۶$$

چون اهمیت ترم ششم، $\frac{۱}{۶}$ است. بنابراین به این ترم نسبت به ۵ ترم گذشته وزن می‌دهیم.

توجه: هر چه α بیشتر باشد، اهمیت پدیده‌ی لحظه‌ای کمتر می‌شود.

☜ ***نکته:*** بر اساس RFC۲۹۸۸ معمولاً در پروتکل TCP، $\alpha = \frac{۷}{۸}$ در نظر گرفته می‌شود.
یعنی به پدیده قدیمی وزن $\frac{۷}{۸}$ و به پدیده جدید وزن $\frac{۱}{۸}$ می‌دهیم.

تخمین انحراف معیار

$$d_{RTT}(new) = \beta\, d_{RTT}(old) + (۱-\beta)\left[T_n - t_{RTT}(new)\right]$$

☜ ***نکته:*** بر اساس سند RFC۲۹۸۸ معمولاً در پروتکل TCP، $\beta = \frac{۳}{۴}$ در نظر گرفته می‌شود.

سؤال : برنامه کاربردی که می‌خواهد شروع به کار نماید باید همان لحظه‌ی اول زمان timeout اش را مشخص کند. در حالی که داده‌ای ارسال نکرده است، بنابراین زمان timeout به چه صورت تعیین می‌شود؟ زیرا هنوز که ارسالی را در TCP انجام نداده‌ایم و واقعه‌ای اتفاق نیفتاده است، میانگین و انحراف معیار وجود ندارد (چون هنوز داده‌ای وجود ندارد)
پاسخ: در TCP، برای بار اول تخمین می‌زنند که زمان timeout چقدر باشد و بعد با هر بار ارسال داده و گرفتن ACK این تخمین را تصحیح می‌کنند تا به آن میانگین واقعی نزدیک شوند و بعد بر اساس آن، زمان رفت و برگشت را حساب می‌کنند.

توجه: هر چه واقعه بیشتر اتفاق بیفتد، تخمین دقیق‌تر می‌شود.
در واقع TCP، با یک t_{RTT} و d_{RTT} اولیه شروع به کار می‌کند. و بعد با هر ارسال و زمان دریافت ACK تخمینش را تصحیح می‌کند. بعد از ۸-۹ بار ارسال، تخمینش با واقعیت دارای اختلاف خیلی کمی است و زمان timeout مناسبی را تخمین می‌زند.
مطابق روابط محاسبه timeout داریم:

$$t_{RTT}(new) = \alpha \times t_{RTT}(old) + (۱-\alpha) \times T_n \xrightarrow{\alpha = \frac{۷}{۸}}$$

$$t_{RTT}(new) = \frac{۷}{۸} \times ۹۶ \times \frac{۱}{۸} \times ۱۲۰ = ۹۹$$

$$d_{RTT}(new)=\beta\times d_{RTT}(old)+(۱-\beta)\left[(t_n-t_{RTT}(new)\right]\xrightarrow{\beta=\frac{۳}{۴}}$$

$$d_{RTT}(new)=\frac{۳}{۴}\times ۲۰+\frac{۱}{۴}\times ۲۱=۲۵$$

$$T_{timeout}=t_{RTT}(new)+۴\times d_{RTT}(new)=۹۹+۸۱=۱۸۰$$

۳- گزینه (۳) صحیح است.

☜ ***نکته مهم:*** برنامه کاربردی داده‌ها را به صورت stream (جریانی از بایت‌ها) به TCP می‌دهد، TCP در مبدأ بافر ارسال دارد و هر موقع که برنامه کاربردی به آن داده داد، آن را بلافاصله ارسال نمی‌کند، بلکه در بافرش قرار می‌دهد تا به اندازه‌ی یک Segnent شود و بعد ارسال می‌کند. یک Headr به آن اضافه می‌کند و به IP می‌دهد. IP هم آن Segment را با یک بسته IP می‌فرستد و در شبکه عبور می‌دهد (از طریق لایه شبکه) تا به مقصد برسد. وقتی این بسته IP، به گره مقصد رسید IP، این Segment را به TCP می‌دهد. TCP با دیدن Header این سگمنت، تشخیص می‌دهد که این سگمنت متعلق به کدام برنامه کاربردی است.

روش‌های کنترل ازدحام

۱- پیش‌گیرانه (باز) : کنترل ورودی (پیش‌گیری از وقوع ازدحام) (open loop) در واقع اجازه‌ی وقوع ازدحام را نمی‌دهد.

برای پیش‌گیری : از روش Call Admission Control استفاده می‌شود، یعنی در این شبکه‌ها ابتدا کاربر درخواست ورود جریان ترافیکی‌اش را به شبکه می‌دهد (باید همراه درخواستش، مشخصه ترافیکی‌اش را هم بگوید که چه حجم ترافیک می‌خواهد وارد کند، نرخ ارسالش چقدر است) تا شبکه درخواستش را بررسی کند، در صورتی که این درخواست (اضافه شدن این جریان ترافیکی) باعث ایجاد ازدحام (Congestion) نشود. یعنی حتماً مسیری از مبدأ به مقصد وجود دارد که این ظرفیت ارسال را از خودش عبور دهد و ازدحام به وجود نیاید، با درخواست موافقت می‌گردد، در غیر اینصورت درخواست لغو می‌گردد. بنابراین در این روش از منابع شبکه به خوبی استفاده نمی‌گردد.

در واقع هر منبع برای ارسال داده در شبکه، باید از شبکه اجازه بگیرد، شبکه در این اجازه بررسی می‌کند که آیا congestion رخ می‌دهد یا نه؟

سؤالی که مطرح می‌شود، این است که CAC از چه طریقی تشخیص می‌دهد که ظرفیتی که کاربر می‌خواهد، وجود دارد یا نه؟

پاسخ را می‌توان در اطلاعات آماری شبکه جستجو کرد، اما آیا می‌توان شخصی که بنا به هر دلیلی مثل شانس و اقبال با رتبه‌ی ۱۰۰ در دانشگاه شریف قبول شده است بگوییم، هر سال این اتفاق

می‌افتد؟ کار آماری کار سختی است، هنوز الگویی برای آن وجود ندارد، هر چند خیلی بر روی آن کار شده است، کوچ جمعیت به یک دانشگاه خاص الگوی خاصی ندارد!
جریان‌های ترافیکی موجود در شبکه، عمدتاً دارای نرخ ارسال متغیر هستند، یعنی از ابتدا که برقرار می‌شوند، نرخ ارسالشان ثابت نیست و با زمان تغییر می‌کند.

مثال: اگر خروجی اکثر برنامه‌های کاربردی کامپیوتری را monitor کنید، نرخ ارسالی که تولید می‌کنند یک نرخ ارسال متغیر با زمان است. (نرخ ارسال در واحد زمان کم و زیاد می‌شود.)

☜ ***نکته:*** ظرفیت لینک شبکه یک مقدار ثابت است و روی لینک، جریان‌های مختلف ترافیکی وجود دارد، که هر کدام از این جریان‌ها دارای نرخ بیت متغیر هستند.
حالا این CAC از کجا می‌فهمد که باید این جریان را اضافه کند یا نه؟
پاسخ: یک کار آماری انجام می‌شود و بر اساس آن، نتیجه گیری می‌کنند. (در واحد زمان) و به احتمال فراوان حاصل جمع اطلاعات آماری ایستگاه‌ها دقیق نخواهد بود چون مدام در حال تغییر است!
اگر حاصل جمع آماری دقیق نباشد:
۱- ازدحام ایجاد می‌شود، یعنی جریانی را که نباید بپذیرد، می‌پذیرد.
۲- تلف کردن ظرفیت، یک جریانی را که می‌توانسته بپذیرد و ازدحام اتفاق نمی‌افتاده را نمی‌پذیرد!

Traffic Policing (در ورودی انجام می‌شود)

اگر کاربر آمد و تخلفی انجام داد، مثلاً bitrateاش را به شبکه ۲mbps اعلام کرده ولی شده ۲/۲mbps و شبکه بر اساس آن ۲ mbps گفته بود که ازدحام رخ نمی‌دهد ولی بر اساس ۲/۲mbps ممکن است ازدحام رخ بدهد، پس باید کاربر را کنترل کنیم، برای این کار واحدی به نام Traffic policing وجود دارد.
مانند پلیس راهنمایی و رانندگی که با متخلفین برخورد می‌کنند.
در واقع traffic policing روی ترافیکی که در حال عبور است، نظارت می‌کند و در صورتی که تخلفی مشاهده کرد وارد عمل می‌شود، در صورتی که تخلفی نباشد، هیچ سرباری ایجاد نمی‌کند و packet هایی که درست کار می‌کنند و آن توافق ترافیکی را رعایت می‌کنند کاری ندارد.

نحوه برخورد Traffic Policing

۱- سخت‌گیرانه: حذف ترافیک اضافه و بازگشت به ترافیک توافق شده در نتیجه ازدحام در شبکه ایجاد نمی‌شود.

۲- تدبیرگرانه: عدم حذف ترافیک اضافه و نصب بر چسب tag (برگه‌ی جریمه) بر روی Packetهای اضافه، و بعد به Packet می‌گوید برو، اما به او می‌گوییم هرجا در گره‌ای ازدحام رخ داد، بسته‌هایی که tag خوردند، در آن گره را حذف می‌کنیم.

توجه: نمی‌توان این بسته‌هایی که tag خوردند را نگه داشت و بعداً ارسال کنیم، چون جا نداریم خود ازدحام به دلیل پر بودن بافرها و نداشتن ظرفیت رخ داده است.

راه حل تدبیرگرانه، بهتر است چون احتمال دارد بسته‌هایی که tag خورده‌اند، به مقدارشان به مقصد برسند و بدون اینکه ازدحام رخ دهد.

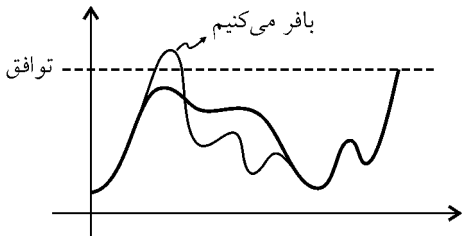
پس دو ماژول در گره‌های شبکه وجود دارد:

۱- CAC : برای پذیرش ارتباط، محتاط عمل می‌کند به طوری که وقتی پذیرش داد، احتمال وقوع ازدحام در شبکه نباشد.

۲- traffic policing : بعد از اینکه یک مکالمه (ارتباطی) برقرار می‌شود، ترافیک آن مکالمه تحت نظارت و کنترل قرار می‌گیرد که اگر تخلفی کرد با آن برخورد شود.

Traffic shaping (در خروجی انجام می‌شود)

کاربر می‌داند که اگر ترافیک بالاتر از توافق تولید کند، ممکن است ترافیک اضافه‌اش توسط Traffic policing حذف گردد. پس بهتر است، ترافیکش را خودش کنترل کند و طبق توافق پیش رود، تا تخلفی برایش از دید شبکه ثبت نشود.



توجه: کاربر حداکثر نرخ ارسالش را ۲mbps اعلام کرده است ولی الان که دارد ترافیکش را تولید می‌کند، می‌بیند که شده ۲/۲mbps، ۰/۲ را خودش در مبدأ بافر می‌کند و در زمانی که نرخ ترافیکش پایین آمد، آن را می‌فرستد.

توجه: به این کاری که کاربر انجام داده است Traffic shaping گفته می‌شود. یعنی شکل‌دهی ترافیک در جهتی که ترافیک به آن شکل مورد توافق در بیاید. کاربر و شبکه دو موجودیت مستقل‌اند:

شبکه: traffic policing انجام می‌دهد.
کاربر، traffic shaping انجام می‌دهد.
توجه: شبکه اصلاً نمی‌داند که کاربر، Traffic shaping انجام می‌دهد یا نه، برای همین همیشه نظارت و traffic policing را انجام می‌دهد.
پلیس نمی‌داند که ما آدم‌هایی هستیم که تخلف می‌کنیم یا نه، چون ممکن است آدم‌هایی باشند که تخلف می‌کنند، بنابراین پلیس همیشه لازم است.
☜ ***نکته:*** traffic shaping، همیشه باعث افزایش تأخیر می‌شود، این افزایش تأخیر می‌تواند باعث بر هم زدن کیفیت سرویس کاربر شود.
پس: در شبکه‌هایی که کنترل ازدحام ، به روش Open Loop (پیشگیرانه) انجام می‌شود، از تمامی ظرفیت شبکه نمی‌توان استفاده کرد.
توجه: روش پیشگیرانه در شبکه‌ی اینترنت استفاده نمی‌شود. شبکه‌های اینترنت برای کنترل ازدحام، از روش‌های واکنشی استفاده می‌کنند.

۲- واکنشی (بسته) (اصلاح ورود به جای کنترل ورودی)

از ابتدا جلوی ازدحام را نمی‌گیریم، بعد از رخ دادن ازدحام، واکنش نشان می‌دهیم و آن را کنترل می‌کنیم.
از ارسال ایستگاهی جلوگیری نمی‌کنیم و آزادانه به هر ایستگاه در هر زمانی این اجازه داده شده است که هر چقدر ترافیک دارد، داده وارد شبکه کند. اما در صورت وقوع ازدحام، واکنش نشان می‌دهیم، در این واکنش به همه‌ی منابع می‌گوییم که از حجم ارسال داده‌شان کم (اصلاح ورودی) کنند البته به میزانی کم کنند تا ازدحام کم شود.
☜ ***نکته:*** در این روش نیازی به اجازه گرفتن از کسی نیست ولی این کنترل می‌شود که منابع به طور دائم از شبکه Feedback بگیرند که آیا در داخل شبکه ازدحام برای این جریان ترافیکی رخ داده است یا نه.
– اگر رخ نداده است، حتی می‌تواند نرخ بیتش را بالاتر هم ببرد.
– اگر رخ داده باشد، ناچار است نرخ بیتش را پایین بیاورد تا شبکه از ازدحام خارج شود.
نتیجه اینکه حلقه بازخورد (Feedback) از جمله ویژگی‌های روش کنترل ازدحام حلقه بسته یا واکنشی است، پس گزینه سوم نادرست است.

۴- گزینه (۳) صحیح است.

پروتکل HTTP در لایه کاربرد

به برنامه کاربردی که روی اینترنت نوشته شده است، world wide web یا شبکه جهانی وب گفته می‌شود. زیرا documentهایی داریم که Linkها را به هم متصل می‌کند، پروتکلی که برای آن طراحی شده است، پروتکل HTTP (HyperText Transfer Protocol) نام دارد.

کاری که HTTP انجام می‌دهد این است که clientها، objectها را به web server، request می‌دهند و web server هم objectها را می‌آورد.
objectها می‌توانند یک فایل HTML با یک تصویر JPEG و ... باشند که توسط این پروتکل می‌توانند منتقل شوند.
هر object ای در محیط عملیاتی اینترنت با یک آدرس منحصر به فرد معرفی می‌شود که به آن URL گفته می‌شود. URL سرواژه عبارت Uniform Resource Locator می‌باشد.

مثال:

www.iust.ac.ir/index.htm
/home/logo.jpg
/home/Header.jpg

در صفحه اول دانشگاه ممکن است n تا object وجود داشته باشد.
پس اولین کاری که می‌کنیم تا یک صفحه web بیاید این است که یک request از سمت Client به Server بدهیم بدون این که چیزی مشخص کنیم، آن صفحه‌ی اصلی به فرم HTML می‌آید، در فایل HTML گفته شده که در آن چند object وجود دارد و بعد browser شما objectها را به آن شکلی که هست نشان می‌دهد.
از آنجاییکه پروتکل http به دلیل دغدغه **صحت** داشتن با پروتکل TCP در لایه انتقال کار می‌کند، در ادامه ابتدا TCP، درخواست Clint به سمت Server را معوق می‌کند تا یک TCP Connection مابین فرستنده و گیرنده برای درخواست و دریافت فایل پایه HTML ایجاد کند. این TCP Connection در سه گام یعنی (۱) فاز برقراری اتصال (3–way handshaking)، (۲) فاز تبادل داده و (۳) فاز رهاسازی اتصال انجام می‌گردد. که در ادامه به بررسی فاز برقراری اتصال (way 3–handshaking) می‌پردازیم:

فاز برقراری اتصال (3–way handshaking)

برای ایجاد TCP Connection ، سه پیغام TCP رد و بدل می‌شود که به آن 3–way handshaking (دست‌تکاندهی سه طرفه) نیز گفته می‌شود. مراحل فاز برقراری اتصال به صورت زیر است:

۱) ابتدا Clinet، درخواست برقراری Connection را به Server می‌دهد.(SYN=1)
۲) Server یک ACK به Client ارسال می‌کند یعنی می‌پذیرد که Connection سمت Client به سمت Server باز شود. همچنین Server علاوه بر ACK یک درخواست ایجاد Connection از سمت Server به Client هم می‌فرستد.(ACK=1, SYN=1)

توجه: Server ACK و درخواست ایجاد Connection هر دو با هم از طرف Server در قالب یک پیام به سمت Client ارسال می‌گردد.

توجـه: وقتـی Client، ACK را از Server گرفـت، Connection سـمت Client بـه Server بـاز می‌شود، پس Client می‌توانـد داده و درخواسـت بفرسـتد. Client ایـن اختیـار را دارد کـه همـراه ACK، داده و درخواست هم بفرستد.

۳) Client یک ACK به Server ارسال می‌کند یعنی می‌پذیرد کـه Connection سـمت Server بـه سمت Client باز شود. (ACK=1)

توجـه: وقتـی Server، ACK را از Client گرفـت، Connection سـمت Server بـه Client بـاز می‌شود، پس Server می‌تواند داده و درخواست بفرستد.

توجه: TCP، Connectionهایش دو طرفه است، یعنی هم از سمت Client به سـمت Server یـک Connection ایجاد می‌کند و هم از سمت Server به سمت Client یک Connection ایجاد می‌کند.

توجه: تا این سه پیغام رد و بدل نشوند. Connection بین Client و Server ایجاد نشده است، به این سه پیغام در TCP اصطلاحاً 3-way handshaking گفته می‌شود. به معنی دست‌تکان‌دهی سه طرفه، در واقع با این کار، دو گره دارند عمل خوشامدگویی انجام می‌دهند و سپس Connection به شکل دو طرفه برقرار می‌شود.

مثال: برای مثال شما وقتی دوستتان را ببینید برای باز کردن سر صحبت یک سـری تعارفـات اولیـه انجام می‌دهید: سلام، ... ، دست دادن ... این‌ها که گفتیم برای فاز برقراری اتصال بود.

توجه: پس حداقل یک زمان رفت و برگشت طول می‌کشد تا Client بتواند یک request مربوط به درخواست و دریافت فایل پایه html را بدهد. البتـه اگـر request اش را همـراه ACK بدهـد، کـه معمولا به این صورت است. به این زمان رفت و برگشـت اصـطلاحا RTT یـا Round Trip Time گفته می‌شود.

توجه: این تأخیر RTT از مـوقعی کـه Client یـک request بـه Server مـی‌دهـد تـا ACK آن را دریافت کند یعنی Connection برقرار شود، یا از موقعی که یـک پیغـام مـی‌دهـد تـا جـواب آن را بگیرد، شامل تمام تأخیرهای شبکه است، تأخیر انتقال (T_F)، تـأخیر انتشـار (T_{Prop})، تـأخیر صـف (T_{queue})، تأخیر پردازش $(T_{process})$.

توجه: RFC ای که برای HTTP وجود دارد RFC۱۹۵۴ و RFC۲۶۱۶ است.

توجه: تمام پروتکل‌هایی که در شبکه‌ی اینترنت وجود دارند، دارای RFC هستند، برای مثال بـرای دیدن جزئیات آن ها باید RFCشان را بگیریم و مطالعه کنیم یا اگر بخـواهیم آن‌هـا را پیـاده‌سـازی کنیم باید RFC آنها را تهیه کنیم. RFC مانند کتاب قانون است، قوانینی دارد که می‌گوید:

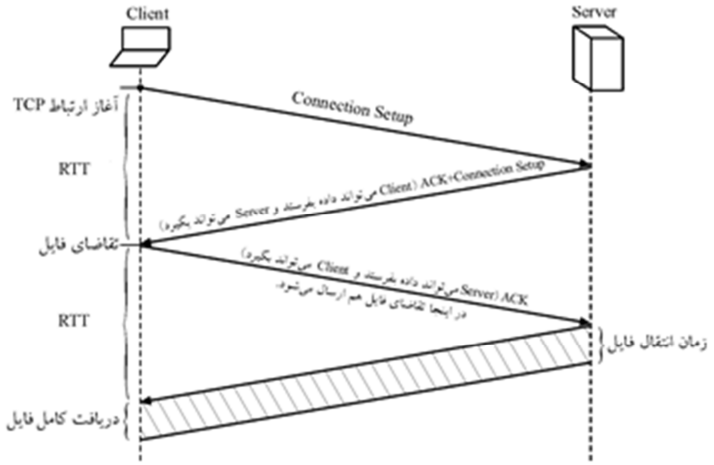
۱- اول این کار را انجام بده

۲- این پیغام را دریافت کردی، بعد این کار را انجام بده و ...

RFC یک Reference برای پیاده سازی بدون ابهام است.

توجه: شرح RFC ها در سایت IETF.ORG قرار دارد.

توجه: پس از آنکه فاز برقراری اتصال (3–way handshaking) انجام شد، یعنی Connection سمت Client به Server باز شد. آنگاه نوبت به ارسال request به معنی درخواست و دریافت فایل پایه HTML از سمت Client به Server می‌رسد، این صفحه‌ی اصلی یعنی فایل پایه HTML به فرمت HTML می‌آید، در فایل پایه HTML گفته شده است که در آن چند object وجود دارد و بعد browser شما objectها را به آن شکلی که هست نشان می‌دهد. در این حالت نقشه درخواست و دریافت objectها در فایل پایه HTML مشخص شده است. شکل زیر گویای مطلب می‌باشد:



به طور کلی زمان دستیابی به یک صفحه وب به طور کامل از رابطه زیر محاسبه می‌گردد:

$$T_{Access\ (Website)} = T_{Translate\ (Domain\ to\ IP)} + T_{Destination} = T_{DNS\ LOOK\ UP} + T_{HTTP}$$

توجه: فرض کنید در مرورگر وب خود برای دریافت یک صفحه وب به طور کامل بر روی یک لینک کلیک می‌کنید و آدرس IP مربوط به این URL در میزبان محلی **ذخیره نشده** است، در نتیجه برای به دست آوردن آدرس IP به یک DNS LOOK UP نیاز است. فرض کنید برای دریافت آدرس IP از طریق سرویس DNS، n سرور DNS ملاقات می‌شوند و تاخیر زمان رفت و برگشت معادل RTT1 تا RTTn باشد. بنابراین بدون در نظر گرفتن زمان مربوط به درخواست و دریافت فایل پایه html و objectهای موجود در آن، رابطه زیر را خواهیم داشت:

$$T_{Access\ (Website)} = T_{Translate\ (Domain\ to\ IP)} + T_{Destination} = T_{DNS\ LOOK\ UP} + T_{HTTP} = \sum_{i=1}^{n} RTT_i + T_{HTTP}$$

حال در ادامه به نحوه‌ی محاسبه T_{HTTP} در شرایط مختلف می‌پردازیم:

توجه: از پروتکل HTTP به دو حالت می‌توان استفاده کرد:

(۱) non-persistent http : ناپایدار و (۲) persistent http : پایدار

Non-persistent http ناپایدار(غیرمصر یا غیرمداوم)

در حالت Non-persistent http یک TCP connection مابین فرستنده و گیرنده برای درخواست و دریافت فایل پایه HTML ایجاد می‌گردد و در انتها Connection بسته می‌شود. در ادامه نیز برای درخواست و دریافت objectها به طور مستقل Connection باز و بسته می‌شود.

حالت Non-persistent http خود به سه روش ترتیبی، موازی نامحدود و موازی محدود وجود دارد، که روابط آن به صورت زیر است:

روش ناپایدار ترتیبی:

$$\left[(\overset{\uparrow}{RTT}+\underset{\downarrow}{RTT}+\overset{\uparrow}{T_F})\right]+\left[\underset{\downarrow}{n}\times(\overset{\uparrow}{RTT}+\underset{\downarrow}{RTT})+\sum_{i=1}^{n}\overset{\uparrow}{T_F}(i)\right]$$

درخواست و برقراری ارتباط — زمان انتقال فایل اصلی — درخواست و دریافت فایل اصلی — درخواست و برقراری ارتباط — تعداد درخواست‌ها — درخواست و دریافت objectها — زمان انتقال object

روش ناپایدار موازی نامحدود:

$$\left[(\overset{\uparrow}{RTT}+\underset{\downarrow}{RTT}+\overset{\uparrow}{T_F})\right]+\left[\underset{\downarrow}{n}\times(\overset{\uparrow}{RTT}+\underset{\downarrow}{RTT})+\sum_{i=1}^{n}\overset{\uparrow}{T_F}(i)\right]$$

درخواست و برقراری ارتباط — زمان انتقال فایل اصلی — درخواست و دریافت فایل اصلی — درخواست و برقراری ارتباط — یک — درخواست موازی و دریافت موازی — زمان انتقال object

روش ناپایدار موازی محدود:

$$\left[(\overset{\uparrow}{RTT}+\underset{\downarrow}{RTT}+\overset{\uparrow}{T_F})\right]+\left[\underset{\downarrow}{n}\times(\overset{\uparrow}{RTT}+\underset{\downarrow}{RTT})+\sum_{i=1}^{n}\overset{\uparrow}{T_F}(i)\right]$$

درخواست و برقراری ارتباط — زمان انتقال فایل اصلی — درخواست و دریافت فایل اصلی — درخواست و برقراری ارتباط — تعداد درخواست‌ها — درخواست موازی و دریافت موازی — زمان انتقال object

persistent http پایدار(مصر یا مداوم)

در حالت persistent http یک TCP connection مابین فرستنده و گیرنده برای درخواست و دریافت فایل پایه HTML ایجاد می‌گردد و در انتها Connection باز می‌ماند. در ادامه نیز برای درخواست و دریافت object ها همان TCP connection اولیه مورد استفاده قرار می‌گیرد.

حالت persistent http خود به سه روش ترتیبی، موازی نامحدود و موازی محدود وجود دارد، که روابط آن به صورت زیر است:

روش پایدار ترتیبی:

$$\left[(\overset{\uparrow}{RTT}+\underset{\downarrow}{RTT}+\overset{\uparrow}{T_F})\right]+\left[\underset{\downarrow}{n}\times(\overset{\uparrow}{\cancel{RTT}}+\underset{\downarrow}{RTT})+\sum_{i=1}^{n}\overset{\uparrow}{T_F}(i)\right]$$

درخواست و برقراری ارتباط — زمان انتقال فایل اصلی — درخواست و برقراری ارتباط — زمان انتقال object

صفر — تعداد درخواست‌ها — درخواست و دریافت فایل اصلی — درخواست و دریافت object

پایدار موازی نامحدود:

$$\left[(\overset{\uparrow}{RTT}+\underset{\downarrow}{RTT}+\overset{\uparrow}{T_F})\right]+\left[\underset{\downarrow}{n}\times(\overset{\uparrow}{\cancel{RTT}}+\underset{\downarrow}{RTT})+\sum_{i=1}^{n}\overset{\uparrow}{T_F}(i)\right]$$

درخواست و برقراری ارتباط — زمان انتقال فایل اصلی — درخواست و برقراری ارتباط — زمان انتقال object

صفر — یک — درخواست و دریافت فایل اصلی — درخواست موازی و دریافت موازی object

روش پایدار موازی محدود:

$$\left[(\overset{\uparrow}{RTT}+\underset{\downarrow}{RTT}+\overset{\uparrow}{T_F})\right]+\left[\underset{\downarrow}{n}\times(\overset{\uparrow}{\cancel{RTT}}+\underset{\downarrow}{RTT})+\sum_{i=1}^{n}\overset{\uparrow}{T_F}(i)\right]$$

درخواست و برقراری ارتباط — زمان انتقال فایل اصلی — درخواست و برقراری ارتباط — زمان انتقال object

صفر — تعداد درخواست‌ها — درخواست و دریافت فایل اصلی — درخواست موازی و دریافت موازی object

توجه: اگر برای مدتی روی Connection ، request ای نیاید، server آن را می‌بندد.

توجه: بستگی به برنامه کاربردی دارد Persistent یا Non persistent را انتخاب کند. پروتکل HTTP به هر دو اجازه می‌دهد.

در صورت سوال گفته شده است که **فرض کنید شخصی در مرورگر وب خود روی یک لینک برای دریافت یک صفحه وب کلیک می کند. اگر آدرس IP مربوط به این URL در میزبان به صورت محلی وجود داشته باشد و فایل HTML مرتبط با این لینک دارای هشت Object باشد، در صورتی که زمان رفت و برگشت بین سرویس گیرنده و سرویس دهنده ۱۰۰ میلی ثانیه و زمان ارسال Objectها ناچیز باشد، به ترتیب با استفاده از پروتکل Non-persistent HTTP و HTTP Persistent از زمانی که شخص روی لینک کلیک می‌کند تا زمانی که صفحه وب را به طور کامل دریافت می‌کند بر حسب میلی ثانیه چقدر طول می‌کشد؟**

۱) ۱۸۰۰ و ۹۰۰ ۲) ۹۰۰ و ۳۰۰ ۳) ۱۸۰۰ و ۳۰۰ ۴) ۳۰۰ و ۱۰۰

داده‌های مسئله به صورت زیر است:

$L_{\text{Base HTML File}} = ۰$, $L_{\text{Object}} = ۰$

$RTT = ۱۰۰ \text{m sec}$

$\text{Cardinality(Object)} = ۸$, $T_{\text{DNS LOOK UP}} = ۰$

روش ناپایدار و ترتیبی:

$$\left[\left(\frac{RTT}{۱۰۰}+\frac{RTT}{۱۰۰}+\underset{\text{صفر}}{T_F}\right)\right]+\left[۸\times\left(\frac{RTT}{۱۰۰}+\frac{RTT}{۱۰۰}\right)+\underset{\text{صفر}}{\sum_{i=۱}^{۸}T_F(i)}\right]$$

$\rightarrow ۲۰۰+۱۶۰۰=۱۸۰۰ \text{ ms}$

روش ناپایدار و موازی نامحدود:

$$\left[\left(\frac{RTT}{۱۰۰}+\frac{RTT}{۱۰۰}+\underset{\text{صفر}}{T_F}\right)\right]+\left[۱\times\left(\frac{RTT}{۱۰۰}+\frac{RTT}{۱۰۰}\right)+\underset{\text{صفر}}{\sum_{i=۱}^{۸}T_F(i)}\right]$$

$\rightarrow ۲۰۰+۲۰۰=۴۰۰ \text{ ms}$

روش پایدار و ترتیبی:

$$\left[\left(\frac{RTT}{۱۰۰}+\frac{RTT}{۱۰۰}+\underset{\text{صفر}}{T_F}\right)\right]+\left[۸\times\left(\frac{RTT}{۱۰۰}\right)+\underset{\text{صفر}}{\sum_{i=۱}^{۸}T_F(i)}\right]$$

$\rightarrow ۲۰۰+۸۰۰=۱۰۰۰ \text{ ms}$

روش پایدار و موازی نامحدود:

$$\left[\left(\frac{RTT}{۱۰۰}+\frac{RTT}{۱۰۰}+\underset{\text{صفر}}{T_F}\right)\right]+\left[۱\times\left(\frac{RTT}{۱۰۰}\right)+\underset{\text{صفر}}{\sum_{i=۱}^{۸}T_F(i)}\right]$$

$\rightarrow ۲۰۰+۱۰۰=۳۰۰ \text{ ms}$

۵- گزینه (۲) صحیح است.

گذردهی یا نرخ ارسال مؤثر از رابطه‌ی زیر بدست می‌آید.

$$R_e = R \times U$$

که R برابر مقدار نرخ انتقال اسمی (پهنای باند) و U برابر بهره‌وری می‌باشد. مقدار R برابر ۱ مگابایت می‌باشد، بنابراین در ادامه باید مقدار U محاسبه گردد.

رابطه‌ی کلی برای محاسبه بهره‌وری در Go Back N به صورت زیر است:

محاسبه بهره‌وری Go Bank N بدون صرف نظر کردن سربار ACK و Header و زمان پردازش:

$$U_{\text{Go Back N}} = U_{GBN} = \begin{cases} \frac{W}{W_s}\left(1-\frac{H}{L}\right)\frac{(1-P_F)}{1+(W-1)P_F} & W < W_s \\ \left(1-\frac{H}{L}\right)\frac{(1-P_F)}{1+(W_s-1)P_F} & W \geq W_s \end{cases}$$

که W، اندازه پنجره سمت فرستنده و $W_s = \frac{T_O}{T_F}$ برابر با «حداقل» اندازه‌ی پنجره که برای «ماکزیمم کردن راندمان» مورد نیاز است.

T_O و T_F از رابطه زیر بدست می‌آید:

$$T_O = \text{Total Delay} = T_F + 2T_P + T_{ACK} + 2T_{process}$$

$$T_F = \frac{L}{R}$$

که L برابر اندازه فریم و R برابر نرخ انتقال می‌باشد.

محاسبه بهره‌وری Go Back N با صرف‌نظر کردن از سربار ACK و Header و زمان پردازش.

بنابراین مقدار W_s برابر رابطه‌ی زیر خواهد بود:

$$W_s = \frac{T_F + 2T_P + \cancel{T_{ACK}}^{\text{ناچیز}} + \cancel{2T_{process}}^{\text{ناچیز}}}{T_F} = \frac{T_F + 2T_P}{T_F}$$

$$W_s = \frac{1+2\frac{T_P}{T_F}}{1} = \frac{1+2a}{1} = 1+2a$$

توجه: a برابر $\frac{T_P}{T_F}$ در نظر گرفته شده است.

بنابراین با توجه به روابط فوق داریم:

$$U_{\text{Go Back N}} = U_{GBN} = \begin{cases} \frac{W}{1+2a}\frac{(1-P_F)}{1+(W-1)R_F} & W < 1+a \\ \frac{1-P_F}{1+2aP_F} & W \geq 1+2a \end{cases}$$

در صورت سؤال صحبتی از سربار ACK، Header و زمان پردازش نشده است، بنابراین برای محاسبه بهره‌وری از رابطه فوق استفاده می‌کنیم.
در مسأله گفته شده است که شماره ترتیب ارسال یک عدد ۳ بیتی است، بنابراین تعداد شماره ترتیب $۸ = ۲^۳$ است.
در Go Back N پنجره سمت فرستنده برابر W و پنجره سمت گیرنده برابر ۱ و مجموع آنها که تعداد شماره ترتیب‌های لازم را تشکیل می‌دهد W+۱ است. پس داریم:

$$\text{Sequence Number} = ۸ = W_{GBN} + ۱ \rightarrow W_{GBN} = ۷$$

حال باید بررسی کنیم که W با ۱+۲a چه نسبتی دارد.

$$۱+۲a = ۱+۲\times\frac{T_P}{T_F} = ۱+۲\times\frac{\frac{۱۳۰\times۱۰^{-۳}}{۲۵۰۰\times۸}}{۱۰^{۶}} = ۱۴$$

بنابراین $W<۱+۲a$ است، لذا از رابطه زیر برای محاسبه‌ی بهره‌وری استفاده می‌کنیم:

$$U_{\text{Go Back N}} = U_{GBN} = \frac{W}{۱+۲a}\times\frac{(۱-P_F)}{۱+(W-۱)P_F} \quad W<۱+۲a$$

$$\rightarrow \quad \frac{۷}{۱۴}\times\frac{۱-۱۰^{-۴}}{۱+(۷-۱)\times۱۰^{-۴}} = \frac{۷}{۱۴} = ۰/۵$$

حدوداً برابر با یک

بنابراین داریم:

$$R_e = R\times U \rightarrow ۱۰^۶\times۰/۵ \simeq ۵۰۰\,\text{Kbps}$$

۶- گزینه (۱) صحیح است.

به فرآیند تعیین مسیر از مبدأ تا مقصد در شبکه، مسیریابی گفته می‌شود.
توجه: الگوریتم مسیریابی، قسمتی از نرم‌افزار لایه شبکه است که وظیفه و مسئولیت تعیین مسیر از بین مسیرهای موجود، بر عهده آن است.

مسیریابی سلسله مراتبی

در این روش، مسیریاب‌ها به صورت مجموعه‌ای از سیستم‌های خود مختار یا (As: Autonomous System) در نظر گرفته می‌شوند.
انواع مسیریابی سیستم‌های خود مختار به صورت زیر است:

الف) پروتکل‌های مسیریابی درون (Inrta-As)

یا پروتکل دروازه داخلی (IGP:Interior Gate Way Protocol)
رایج‌ترین پروتکل‌های مسیریابی Intra-As عبارتند از:

۱- RIP:Routing Information Protocol

که از نوع الگوریتم بردار فاصله است.

۲- OSPF: Open Shortest Path First

IGRP: Interior Gateway Routing Protocol

که هر دو از نوع الگوریتم وضعیت پیوند هستند.

ب) پروتکل‌های مسیریابی برون (Inter-As)

یا پروتکل دروازه خارجی (EGP: Exterior Gateway Protocol)

رایج‌ترین پروتکل مسیریابی Inter-As عبارتند از:

BGP: Border Gateway Protocol

که بر دو نوع زیر می‌باشد:

IBGP: Internal BGP

EBGP: External BGP

پروتکل OSPF یکی از رایج ترین پروتکل‌های مسیریابی داخل ناحیه یا همان Intra-As است. همان طور که می‌دانید به Intra-AS، IGP نیز گفته می‌شود. OSPF از الگوریتم وضعیت پیوند استفاده می‌کند. این پروتکل برای انتخاب مسیر مناسب‌تر می‌تواند معیارهای مختلف و گوناگونی را نظیر ترافیک و پهنای پاند را در نظر بگیرد.