

موسسه بابان

انتشارات بابان و انتشارات راهیان ارشد

درس و کنکور ارشد

شبکه‌های کامپیوتری

(حل فوق تشریحی تست‌های دولتی ۱۳۹۲ تا ۱۳۹۴)

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 ضمیمه 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؟

۸ (۱) ۲۲/۵ (۲) ۹ (۳) ۲۵ (۴)

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؟

۱۱۹ (۱) ۱۸۰ (۲) ۱۴۴ (۳) ۲۰۱ (۴)

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 شکل دهی 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 سرویس دهنده ۱۰۰ میلی ثانیه و زمان ارسال *Object*ها 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) ۳۰۰ و ۹۰۰ (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 میلی ثانیه 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 ضمیمه 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، $T_{Poll} = \frac{1}{10} T_F$ ، بنابراین داریم:

$$T_{total} = 40 \times T_F + 100 \times T_{poll} \xrightarrow{T_{poll} = \frac{1}{10} T_F} T_{total} = 40 T_F + 10 T_F = 50 T_F$$

پس از زمان کل برابر با $50 T_F$ ، $40 T_F$ آن مفید و $10 T_F$ آن غیر مفید است. بنابراین بر اساس رابطه بهره‌وری داریم:

$$U = \frac{\text{مفید}}{\text{مفید} + \text{غیر مفید}} = \frac{40 T_F}{40 T_F + 10 T_F} = \frac{40 T_F}{50 T_F} = 0.8$$

حال نرخ بیتی هر ایستگاه را به دست می‌آوریم:

نرخ انتقال	ایستگاه	
100	$\backslash Mbps$	$\rightarrow R = 10 kbps$
1	R	

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

$$R_e = \text{Throughput} = R \cdot U = 10 kbps \times 0.8 = 8 kbps$$

۲- گزینه (۲) صحیح است.

تعیین 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 پیش بینی نیست.

مثال: مثلاً دسترسی به سیستم آموزشی (پرتال)

۱ - دیدن پرتال از داخل دانشگاه از طریق شبکه محلی (تأخیر در حد 1ms)

۲ - دیدن پرتال از خانه از طریق اینترنت (هنوز در شبکه داخل کشور) (تأخیر در حد 10ms)

۳ - دیدن پرتال از اروپا (تأخیر در حد 100ms)

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 اجرا 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 پراکنندگی نسبت به میانگین، انحراف معیار گفته می‌شود.

محاسبه *timeout*

بر اساس سند *RFC 2988* 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 برابر ۳ یا ۴ است. بر اساس سند RFC ۲۹۸۸ مقدار K عدد ۴ در نظر گرفته شده است. پس داریم:

$$T_{time\ out} = t_{RTT}(new) + 4 \times d_{RTT}(new)$$

تخمین RTT :

$$t_{RTT}(new) = \alpha t_{RTT}(old) + (1 - \alpha) T_n \quad 0 < \alpha < 1$$

T_n : زمان رفت و برگشت اتفاق افتاده در مرحله 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.

$$\text{معدل کل} = \frac{5}{6} \times 15/5 + \frac{1}{6} \times 16$$

چون اهمیت ترم ششم، $\frac{1}{6}$ 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 سند RFC ۲۹۸۸ معمولاً در پروتکل TCP، $\alpha = \frac{1}{8}$ در نظر گرفته می‌شود.

یعنی به پدیده قدیمی وزن $\frac{7}{8}$ و به پدیده جدید وزن $\frac{1}{8}$ می‌دهیم.

تخمین انحراف معیار

$$d_{RTT}(new) = \beta d_{RTT}(old) + (1 - \beta) [T_n - t_{RTT}(new)]$$

نکته: بر اساس سند RFC ۲۹۸۸ معمولاً در پروتکل TCP، $\beta = \frac{3}{4}$ در نظر گرفته می‌شود.

سؤال: برنامه کاربردی که می‌خواهد شروع به کار باید همان لحظه‌ی اول زمان $timeout$ 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:

$$t_{RTT}(new) = \alpha \times t_{RTT}(old) + (1 - \alpha) \times T_n \quad \xrightarrow{\alpha = \frac{\nu}{\lambda}}$$

$$t_{RTT}(new) = \frac{\nu}{\lambda} \times 96 + \frac{1}{\lambda} \times 120 = 99$$

$$d_{RTT}(new) = \beta \times d_{RTT}(old) + (1 - \beta) \left[(t_n - t_{RTT}(new)) \right] \quad \xrightarrow{\beta = \frac{\nu}{\lambda}}$$

$$d_{RTT}(new) = \frac{\nu}{\lambda} \times 20 + \frac{1}{\lambda} \times 21 = 20/25$$

$$T_{time\ out} = t_{RTT}(new) + 4 \times d_{RTT}(new) = 99 + 81 = 180$$

۳- گزینہ (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 $Segment$ شود و بعد ارسال می‌کند. یک $Header$ 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 گرهی 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.

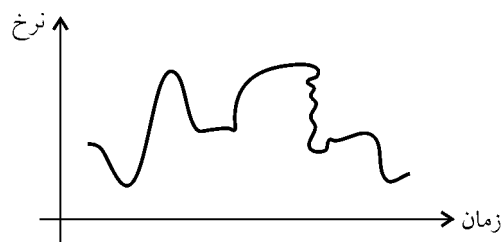
روش‌های کنترل ازدحام

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 این صورت 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 ارسال 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، نتیجه‌گیری 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 $bitrate$ را به شبکه 2mbps اعلام کرده ولی شده $2/2\text{mbps}$ و شبکه بر اساس آن 2mbps گفته بود که ازدحام رخ نمی‌دهد ولی بر اساس $2/2\text{mbps}$ 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

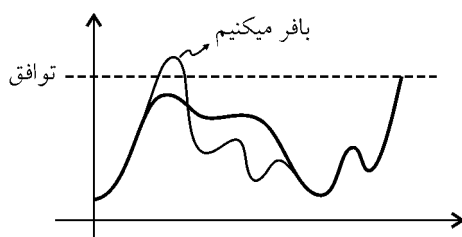
- ۱- سخت‌گیرانه: حذف ترافیک اضافه و بازگشت به ترافیک توافق شده در نتیجه ازدحام در شبکه ایجاد نمی‌شود.
 - ۲- تدبیرگرانه: عدم حذف ترافیک اضافه و نصب بر چسب *tag* (برگه‌ی جریمه) بر روی *Packet*های اضافه، و بعد به *Packet* می‌گوید برو، اما به او می‌گوییم هر جا در گره‌ای ازدحام رخ داد، بسته‌هایی که *tag* 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 مقدرشان 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 2mbps اعلام کرده است ولی الان که دارد ترافیکش را تولید می‌کند، می‌بیند که شده $2/2\text{mbps}$ ، $0/2$ 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 شکل دهی 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 برهم 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 باشد، 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. *word wide web* گفته می‌شود. زیرا *document*‌هایی داریم که *Link*‌ها را به هم وصل می‌کند، پروتکلی که برای آن طراحی شده است، پروتکل *http* (*hypertext Transfer protocol*) نام دارد.

کاری که *HTTP* انجام می‌دهد این است که *client*‌ها، *object*‌ها را به *web server*، *request* می‌دهند و *web server* هم *object*‌ها را می‌آورد.

object‌ها می‌تواند یک فایل *HTML* یا یک تصویر *JPEG* و ... باشد که توسط این پروتکل می‌تواند منتقل شود.

هر *object* ای با یک آدرس منحصر به فرد معرفی می شود که به آن *URL* گفته می شود.

مثال:

www.sbu.ac.ir/index.html

[/home/logo.jpg](http://www.sbu.ac.ir/home/logo.jpg)

[/home/Header.jpg](http://www.sbu.ac.ir/home/Header.jpg)

در صفحه اول دانشگاه ممکن است *n* تا عکس وجود داشته باشد.

پس اولین کاری که می کنیم تا بک صفحه *web* بیاید این است که یک *request* به *Server* بدهیم بدون این که چیزی مشخص کنیم، آن صفحه ی اصلی به فرم *HTML* می آید، در فایل *HTML* گفته شده که در آن چند *object* وجود دارد و بعد *browser* شما *object*ها را به آن شکلی که هست نشان می دهد.

توجه: *RFC* ای که برای *HTTP* وجود دارد *RFC* ۱۹۴۵ و *RFC* ۲۶۱۶ است.

توجه: تمام پروتکل هایی که در شبکه ی اینترنت وجود دارند، دارای *RFC* هستند، مثلاً برای دیدن جزئیات آن ها باید *RFC* شان را بگیریم و مطالعه کنیم یا اگر بخواهیم آن ها را پیاده سازی کنیم باید *RFC* آن ها را تهیه کنیم (خود کتاب شبکه کافی نیست!)
RFC مانند کتاب قانون است، قوانینی دارد که می گوید:

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.

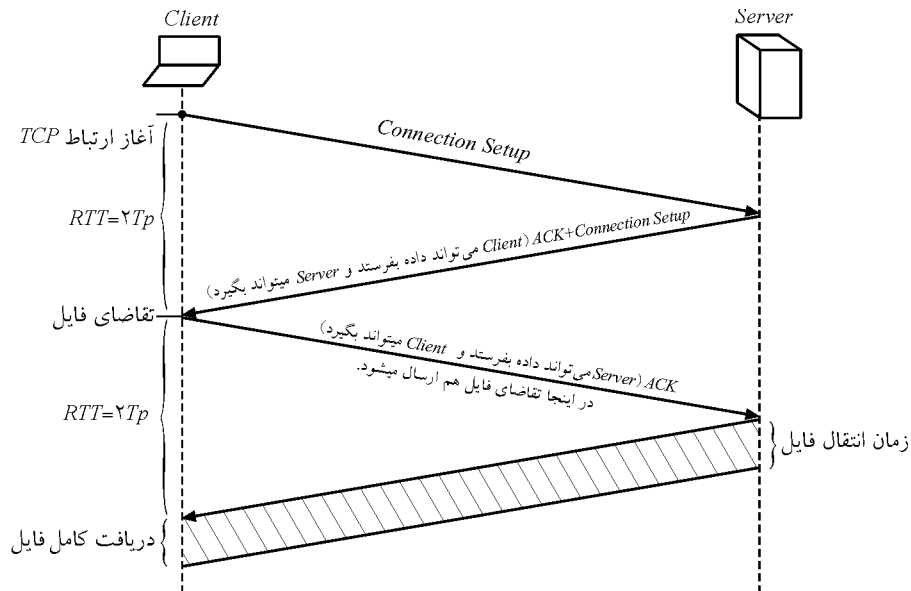
توجه: شرح *RFC*ها در سایت *IETF.ORG* قرار دارد.

نکته: از پروتکل *HTTP* به دو حالت می توان استفاده کرد:

(۱) *non-persistent*: غیر مصر و (۲) *persistent*: مصر

***Non_persistent* غیر مصر**

در حالت *Connection, Non-persistent* را به ازای گرفتن هر *object* می بندیم.



۳-way handshaking

برای ایجاد *connection* در *TCP*، سه پیغام *TCP* رد و بدل می‌شود که به آن *3-way handshaking* گفته می‌شود.

(۱) ابتدا *Client*، درخواست برقراری *Connection* را به *Server* می‌دهد.

(۲) *Server* یک *ACK* به *client* ارسال می‌کند (می‌پذیرد که *Connection* برقرار شود).

توجه: *Server* علاوه بر *ACK* یک درخواست ایجاد *Connection* از سمت *Server* به *Client* هم می‌دهد.

توجه: *TCP Connection*‌ها بیش دو طرفه است، یعنی هم از سمت *Client* به *Server* یک *Connection* ایجاد می‌کند و هم از *Server* به *Client*.

توجه: *Server ACK* و *Connection setup* هر دو با هم از طرف *Server* در قالب یک پیام به سمت *client* ارسال می‌گردد.

(۳) ارسال *ACK* از *Client* به سمت *Server*.

توجه: وقتی *Client*، *ACK* را از *Server* گرفت، *Connection* سمت *Client* به *Server* باز می‌شود، پس *Client* می‌تواند داده بفرستد.

Server هم هنگامی که *ACK* را از *Client* بگیرد، می‌تواند داده بفرستد.

توجه: *Client* این اختیار را دارد که همراه *ACK*، داده هم بفرستد. 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 hand shaking* گفته می‌شود. به معنی دست تکان دهی سه طرفه، در واقع با این کار، دو گره دارند عمل خوشامدگویی انجام می‌دهند و سپس *Connection* برقرار می‌شود.

مثال: مثلاً شما وقتی دوستان را ببینید برای بازکردن سر صحبت یک سری تعارفات اولیه انجام می‌دهید: سلام، ... ، دست دادن ... این‌ها که گفتیم برای فاز برقراری ارتباط بود.

توجه: پس حداقل یک زمان رفت و برگشت طول می‌کشد تا *Client* بتواند داده ارسال کند.

توجه: به این زمان رفت و برگشت *Round Trip Time (RTT)* گفته می‌شود.

توجه: این تأخیر *RTT* از موقعی که *Client* یک *request* به *Server* می‌دهد تا *ACK* آن را دریافت کند یعنی *Connection* برقرار شود (یا از موقعی که پیغام می‌دهد تا جواب آن را بگیرد) شامل تمام تأخیرهای شبکه است؛ (تأخیر ارسال، تأخیر انتشار، تأخیر صف، تأخیر پردازش)

توجه: تا این جا تازه *Connection* ایجاد شد، حالا باید *Client*، *request* بدهد، درخواستش را با پروتکل *HTTP* درست می‌کند و به *Server* می‌دهد، آن هم پاسخ می‌دهد.

نتیجه: پس برای گرفتن یک *object*، حداقل دو *RTT* و یک T_F طول می‌کشد که T_F بستگی به حجم *object* دارد، حال اگر *Non - persistent* استفاده کنیم، به ازای هر *object* اینقدر تأخیر داریم.

$$2RTT + T_F$$

مثال: فرض کنید یک صفحه‌ی وب وجود دارد که ۱۰ تا *Image* در آن هست مثلاً صفحه‌ای اول دانشگاه و می‌خواهید آن را دریافت کنید، چقدر طول می‌کشد تا *browser* کل صفحه را نمایش دهد؟ (به روش *Non - persistent*)

حل:

$$(10 + 1) \times 2RTT + \sum_{i=1}^{10} T_F(i)$$

مثال: اگر یک صفحه‌ی وب *n* تا *object* داشته باشد، چقدر طول می‌کشد؟

Base HTML File

$$(n + 1) \times 2RTT + \sum_{i=1}^n T_F(i)$$

↑
↓
n تا *object*

Persisten : مصر

در این نوع اتصال، سرویس دهنده و گیرنده پس از برقراری اتصال *TCP*، آن را نمی‌بندند در واقع، یک *Connection* برقرار می‌شود و سپس تمام *object*ها از طریق همین *Connection* از *Server* به سمت *Client* منتقل می‌شوند و *request*ها روی همین *connection* ارسال می‌شوند. (به سمت *Server*)

$$3RTT + \sum_{i=1}^n T_F(i)$$

- یک *RTT* برای ارسال درخواست برقراری اتصال *TCP* و دریافت پیام *ACK* از طرف *Server*.
- یک *RTT* برای ارسال *ACK* از *Client* به *Server* به همراه درخواست و دریافت صفحه وب پایه.

• یک *RTT* برای درخواست و دریافت تمام اشیاء

توجه: اگر برای مدتی روی *request*، *Connection* 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)

با توجه با اطلاعات مسأله و روابط فوق داریم:

$$2RTT + \sum_{i=1}^9 T_F(i) = 9 \times 200 = 1800$$

صفر

$$3RTT \times \sum_{i=1}^9 T_F(i) = 3 \times 100 = 300$$

صفر

۵-گزینه (۲) صحیح است.

گذردهی یا نرخ ارسال مؤثر از رابطه‌ی زیر به دست 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 Back N* بدون صرف نظر کردن سربار *ACK* و *Header* و زمان پردازش:

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

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:

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

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

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

محاسبه بهره‌وری $GO\ Back\ N$ با صرف نظر کردن از سربار ACK و $Header$ و زمان پردازش. بنابراین مقدار W_s برابر رابطه‌ی زیر خواهد بود:

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

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

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

بنابراین با توجه به روابط فوق داریم:

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

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 $۲^۳ = ۸$ 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.

$$۱+۲a = ۱+۲ \times \frac{T_P}{T_F} = ۱+۲ \times \frac{۱۳۰ \times ۱۰^{-۳}}{\frac{۲۵۰۰ \times ۸}{۱۰^۶}} = ۱۴$$

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

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

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

حدوداً برابر با یک

بنابراین داریم:

$$R_e = R \times U \rightarrow ۱۰^۶ \times ۰/۵ \simeq ۵۰۰\text{ kbps}$$

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 (*Intra-As*)

با پروتکل دروازه داخلی (*IGP:Interior Gate Way Protocol*)

رایج ترین پروتکل های 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 Protocol

که هر دو از نوع الگوریتم وضعیت پیوند هستند.

ب) پروتکل های مسیریابی برون (*Inter-AS*)

یا پروتکل دروازه خارجی (*EGP:Exterior Gateway Protocol*)

رایج ترین پروتکل مسیریابی *Inter-AS* عبارتند از:

BGP : Border Gateway Protocol

که بر دو نوع زیر می باشد:

IBGP:Internal BGP

EBGP:External BGP

پروتکل *OSPF* یکی از رایج ترین پروتکل های مسیریابی داخل ناحیه یا همان *Intra-AS* است. همان طور که می دانید به *IGP Intra AS* نیز گفته می شود. *OSPF* از الگوریتم وضعیت پیوند استفاده می کند. این پروتکل برای انتخاب مسیر مناسب تر می تواند معیارهای مختلف و گوناگونی را نظیر ترافیک و پهنای باند را در نظر بگیرد.

تست‌های ضمیمه سال ۹۳

۱- فرض کنید ۱۰ کلاينت با استفاده از پروتکل *FTP* به طور همزمان در حال دریافت فایل‌های با حجم زیاد از یک فایل سرور هستند و لینک گلوگاه، لینک متصل به سرور است. اگر یکی از کلاينت‌ها از یک برنامه مدیریت دانلود (*Download Manager*) استفاده کند که به طور همزمان ۹ اتصال همزمان *TCP* باز می‌کند سرعت دانلود این کلاينت نسبت به قبل چند برابر خواهد شد؟

۲ (۱) ۴ (۲) ۵ (۳) ۱۰ (۴)

۲- دلیل اصلی اینکه در پروتکل *IPv6* اجازه *fragmentation* به گره‌های میانی داده نشده است، چیست؟

- ۱) کاهش سربار سرآیند (*header*) بسته‌های *IP* به منظور بالا بردن کارایی پروتکل *IP*
- ۲) اختیاری نمودن *fragmentation* به دلیل انعطاف پذیری *IPv6* در بکارگیری *option*‌ها
- ۳) عدم نیاز به *fragmentation* در پروتکل *IPv6* به دلیل امکان ارسال بسته‌های بزرگ‌تر از ۶۴ کیلوبایت
- ۴) ساده‌تر کردن وظیفه جلورانی (*forwarding*) بسته‌ها به منظور افزایش سرعت سوئیچینگ بسته‌ها

۳- در یک شبکه دیتاگرام به مسیر یاب‌ها اجازه داده می‌شود تا در صورت لزوم بسته‌ها را حذف نمایند. اگر احتمال حذف بسته در هر مسیر یاب ۵۰ درصد باشد، در حالتی که بین گره مبدأ و گره مقصد دو مسیر یاب میانی وجود داشته باشد (بین مبدأ و مقصد سه گام به مقصد وجود دارد) و هر بسته در صورت حذف شدن مجدداً ارسال می‌شود. مطلوب است میانگین تعداد گامی که یک بسته دریافتی طی کرده است؟ (دقت شود که یک بسته ممکن است چند بار ارسال شود تا گیرنده موفق به دریافت آن شود)

۸ (۱) ۷ (۲) ۴ (۳) ۳ (۴)

۴- ایستگاه از طریق دو باس سیمی مختلف به هم متصل شده‌اند. فرض کنید که زمان ارسال هر فریم یک ثانیه طول می‌کشد و زمان نیز به اسلات‌های زمانی یک ثانیه‌ای تقسیم شده است. زمانی

که یک ایستگاه فریمی برای ارسال دارد به طور تصادفی و با احتمال مساوی یکی از باس‌ها را انتخاب کرده و در زمان شروع اسلات بعدی با احتمال p ارسال می‌کند. مقدار p برای حداکثر شدن نرخ ارسال موفقیت‌آمیز چقدر است؟

$$\frac{1}{16} \text{ (۴)} \quad \frac{1}{4} \text{ (۳)} \quad \frac{1}{8} \text{ (۲)} \quad \frac{1}{2} \text{ (۱)}$$

۵- یک شبکه محلی بی سیم برای تبادل داده بین M ایستگاه از یک کانال با نرخ ارسال $25Mbps$ با مکانیزم سرکشی (*polling*) متمرکز با یک ایستگاه پایه (مرکزی) استفاده می‌کند. فرض کنید هر ایستگاه 1000 متر از ایستگاه پایه فاصله دارد، سرعت انتشار امواج $2 \times 10^8 \frac{m}{s}$ ، اندازه فریم سرکشی 500 بیت و اندازه فریم داده 1250 بایت است و هر ایستگاه اتمام ارسال داده خود را با یک فریم 500 بیتی به ایستگاه پایه اعلام می‌کند. اگر هر ایستگاه فقط مجاز باشد یک فریم داده به ازای هر سرکشی ارسال کند حداکثر بهره‌وری این کانال چند درصد است؟

$$95 \text{ (۴)} \quad 93 \text{ (۳)} \quad 91 \text{ (۲)} \quad 89 \text{ (۱)}$$

۶- کدام یک از گزینه‌های زیر در مورد مکانیزم‌های کنترل دسترسی به رسانه (*MAC*) نادرست است؟

- (۱) در بار ترافیکی زیاد، مکانیزم‌های مبتنی بر زمانبندی نرخ گذردهی بیشتری نسبت به مکانیزم‌های رقابتی دارند.
- (۲) پیچیدگی پیاده سازی مکانیزم‌های رقابتی کمتر از مکانیزم‌های مبتنی بر زمانبندی است.
- (۳) در بار ترافیکی کم، مکانیزم‌های رقابتی تأخیر کمتری نسبت به مکانیزم‌های مبتنی بر زمانبندی دارند.
- (۴) سربار پروتکلی مکانیزم‌های مبتنی بر زمانبندی کمتر از مکانیزم‌های رقابتی است.

پاسخ تست‌های ضمیمه ۹۳

۱- گزینه (۳) صحیح است.

سرعت دانلود یک *client* بدون *Download Manager*

در این حالت هر یک از *client*‌ها دارای یک اتصال *TCP* هستند، بنابراین هر یک از *client*‌ها به اندازه $\frac{1}{10}$ از پهنای باند را مصرف می‌کنند.

سرعت دانلود یک *client* با استفاده از *Download Manager*

در این حالت یکی از *client*‌ها که از *Download Manager* استفاده می‌کند به طور همزمان دارای ۹ عدد اتصال *TCP* است و سایر *client*‌ها همچنان دارای ۱ اتصال *TCP* هستند. بنابراین ۱۰ عدد *client* بر روی هم دارای ۱۸ (۹+۹) اتصال *TCP* هستند: که از این ۱۸ اتصال، $\frac{9}{18}$ مربوط به

client است که از *Download Manager* استفاده کرده است. بنابراین $\frac{18}{9} = 2$ برابر سرعت زیادتر شده است.

۲- گزینه (۴) صحیح است.

عمل *Fragmentation* سربار زیادی را بر روی پردازش ایجاد می‌کند.

۳- گزینه (۲) صحیح است.

احتمال حذف بسته برابر ۵۰ درصد است و آن را با $\frac{1}{2} = P$ نشان می‌دهیم. هر بسته ممکن است یک، دو یا سه گام طی کند. برای یک گام، یعنی مسیریاب اول بسته را حذف می‌کند و احتمال آن P است. برای دو گام، یعنی بسته از مسیریاب اول عبور می‌کند ولی مسیریاب دوم آن را حذف می‌کند و احتمال آن برابر است با $P(1-P)$. برای سه گام یعنی بسته از دو مسیریاب عبور می‌کند و احتمال آن برابر است با $(1-P)(1-P)$. بنابراین میانگین تعداد گام برای هر ارسال بسته برابر است با:

$$1 \times P + 2 \times P(1-P) + 3 \times (1-P)(1-P) = P^2 - 3P + 3$$

از طرف دیگر میانگین تعداد ارسال برای ارسال موفق هر بسته برابر است با: $\frac{1}{(1-p)^2}$

میانگین تعداد گام برای ارسال \times میانگین تعداد ارسال موفق = میانگین تعداد گامی که بسته طی می‌کند

$$= \frac{P^2 - 3P + 3}{(1-P)^2}$$

با جایگذاری $P = \frac{1}{4}$ در رابطه داریم:

$$\frac{P^2 - 3P + 3}{(1-P)^2} = \frac{\frac{1}{4}}{\frac{3}{4}} = 1$$

۴- گزینه (۳) صحیح است.

احتمال موفقیت برابر است با احتمال استفاده از یک باس و عدم استفاده از باس دیگر:

$$P = 8 \left(\frac{1}{4} \times P \right) \left(1 - \frac{1}{4} \times P \right)^7 = 4P \left(1 - \frac{1}{4} \times P \right)^7$$

باید P به حداکثر برسد، بنابراین مشتق گرفته و برابر صفر قرار می‌دهیم.

$$4P \left(1 - \frac{1}{4} \times P \right)^7 \rightarrow 4 \left(1 - \frac{1}{4} \times P \right)^7 + 4P \left(7 \times -\frac{1}{4} \right) \left(1 - \frac{1}{4} \times P \right)^6 = 0 \rightarrow P = \frac{1}{4}$$

۵- گزینه (۱) صحیح است.

برای ارسال داده در هر ایستگاه، ۱- ابتدا ایستگاه مرکزی *poling* را برای آن ایستگاه ارسال می‌کند سپس ۲- ایستگاه با تملک *poling* شروع به ارسال فریم داده می‌کند و در نهایت ۳- ایستگاه بلافاصله با ارسال یک فریم به ایستگاه مرکزی اتمام ارسال فریم خود را اعلام می‌کند.

زمان مرحله ۱ برابر است با زمان ارسال فریم سرکشی + زمان انتشار آن

$$T_1 = \frac{500b}{25 \times 10^6 b/s} + \frac{1000m}{2 \times 10^8 m/s} = 2/5 \times 10^{-5} s$$

زمان مرحله ۲ برابر است با زمان ارسال یک فریم داده

$$T_2 = \frac{1250 \times 8b}{25 \times 10^6 b/s} = 4 \times 10^{-4} s$$

زمان مرحله ۳ برابر است با زمان ارسال فریم اتمام + زمان انتشار آن

$$T_3 = \frac{500b}{25 \times 10^6 b/s} + \frac{1000m}{2 \times 10^8 m/s} = 2/5 \times 10^{-5} s$$

زمان مفید فقط زمان مرحله ۲ است بنابراین بهره‌وری کانال برابر است با:

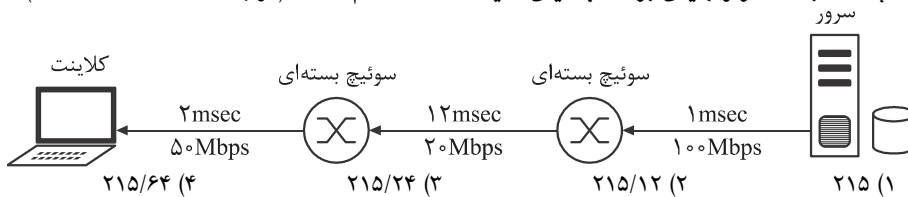
$$U = \frac{T_2}{T_1 + T_2 + T_3} = \frac{4 \times 10^{-4}}{4/5 \times 10^{-4} s} = 0/89$$

۶- گزینه (۴) صحیح است.

شبکه های کامپیوتری دولتی ۱۳۹۴

۵۵- در شبکه‌ای با مسیر شکل زیر بین سرور و کلاینت وجود دارد، حداقل زمان لازم برای انتقال

پانصد بسته هزار بایتی برحسب میلی‌ثانیه (msec) کدام است؟ (توجه: $1\text{Mbps} = 10^6\text{bps}$)



۵۶- کاربری با استفاده از مرورگر وب اقدام به دریافت یک صفحه وب می‌نماید. صفحه وب شامل یک

فایل html و ۹ فایل است. اندازه هر ۱۰ فایل مساوی و پنج هزار بایت است. مرورگر وب از http1.0(non-persistent) استفاده می‌کند. وب سرور حداکثر اجازه پنج ارتباط TCP همزمان

به یک کلاینت را می‌دهد. چنانچه گذردهی شبکه بین کامپیوتر کاربر و وب سرور 10^6bps باشد،

زمان لازم برای دریافت این صفحه برحسب ثانیه (sec) چقدر است؟ زمان رفت و برگشت

(RTT) بین کلاینت و سرور را ۰/۱ ثانیه در نظر بگیرید.

۱/۵ (۴) ۱/۴ (۳) ۱/۲ (۲) ۱/۰ (۱)

۵۷- TCP Reno فایلی را به ۳۲ بسته تبدیل کرده و برای مقصدی ارسال می‌کند. چنانچه بسته ۱۲۷م

به مقصد نرسد چند RTT زمان صرف ارسال فایل خواهد شد. RTT زمان رفت و برگشت بین مبدأ و مقصد است. سطح آستانه پنجره ازدحام را در مرحله «شروع آهسته» چهار در نظر

بگیرید.

۱۲ (۴) ۱۱ (۳) ۹ (۲) ۸ (۱)

۵۸- جدول مسیریابی در یک مسیریاب به صورت زیر است، اگر این مسیریاب بسته‌ای با آدرس مقصد

135.46.52.2 را دریافت کند، گام بعدی این بسته کدام است؟

Network Destination	Next Hop
135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
192.53.40.0/23	Router 1
0.0.0.0/0	Router 2

Interface 1 (۴) Interface 0 (۳) Router 2 (۲) Router 1 (۱)

مانند منوهای موجود در یک سایت خبرگزاری که از بین گروه‌های مختلف خبری، یک گروه خبری مورد علاقه خود را انتخاب می‌کنید و در بخش مورد نظر، بخش بعدی، بعدی و بعدی را انتخاب می‌کنید.

۳- سازگاری واسط کاربر

واسط‌های کاربر از نظر شکل و شمایل باید مطابق قوانین و استانداردهای رسمی و غیر رسمی مقبول جامعه فناوری باشد، به عبارت دیگر با این استانداردها سازگار باشد. برای رسیدن به این هدف رعایت موارد زیر توصیه می‌گردد:

الف) استفاده از نمادها و اشکال سازگار با محیط عملیاتی.

برای مثال، استفاده از نماد کتاب، برای ورود به بخش کتاب در یک فروشگاه اینترنتی.

ب) استفاده از نمادها و اشکال سازگار با محیط نرم‌افزارهای هم خانواده برای مثال، نماد ►، در نرم‌افزارهای پخش صدا و فیلم، فایل مورد نظر را اجرا می‌کند.

ج) سازگاری با اخلاق عمومی نرم‌افزارها.

برای مثال، $Ctrl + S$ ، به طور غیر رسمی، در اغلب نرم‌افزارها، شیوه‌ای عمومی برای ذخیره فایل محسوب می‌گردد. تغییر این پارادایم‌های عمومی، در یک واسط کاربر نرم‌افزار، از نگاه کاربر، ناخوشایند است.

توجه: مشتری و کاربر چیزی را می‌خواهد که برق بزند و به شیوه‌ای کارآمد کار کند، شیوه‌های خلق این خواسته‌ها از نگاه مشتری اهمیتی ندارد!

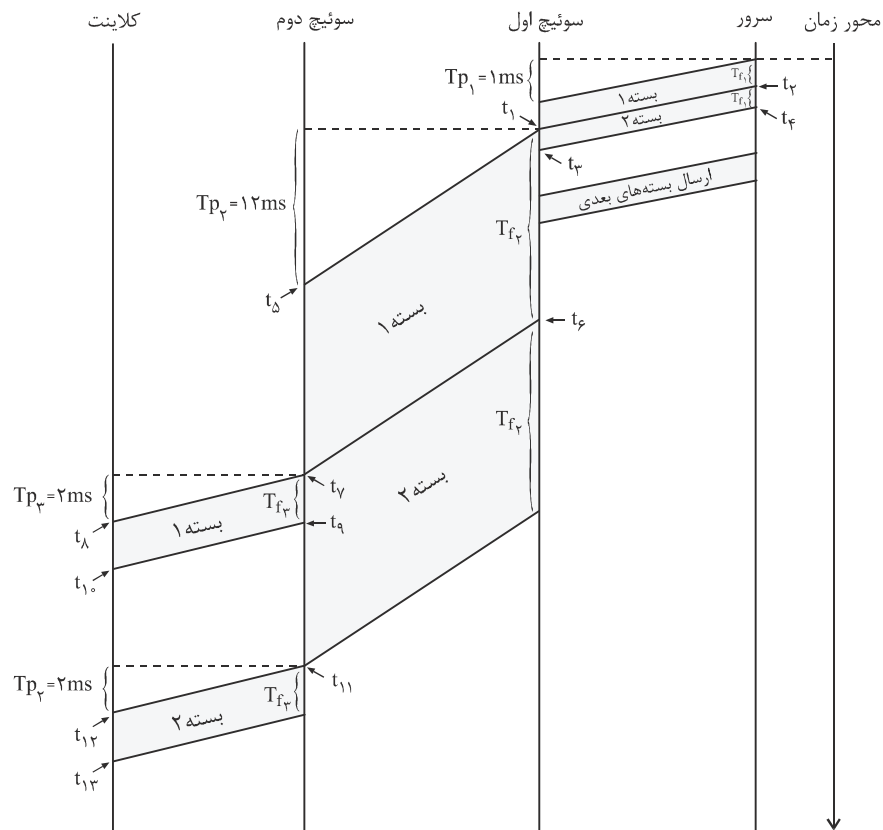
توجه: زمان پاسخ موجود در گزینه سوم، مربوط به نیازمندی‌های غیر وظیفه‌مندی است و نه طراحی واسط کاربر.

۵۴- گزینه (۳). روش‌های ارزیابی و بازبینی برنامه (PERT) و مسیر بحرانی (CPM) دو روش زمان‌بندی برای تولید پروژه‌های نرم‌افزاری هستند. مدل ریاضی این نمودارها گراف است، که به برنامه‌ریز پروژه امکانات زیر را می‌دهد:

- نمایش توالی فعالیت‌ها و مسیرهای بحرانی پروژه
- برآورد زمان لازم برای انجام وظایف از طریق مدل‌های آماری
- محاسبه زمان‌بندی‌های مرزی میان وظایف

شبکه‌های کامپیوتری

۵۵- گزینه (۳).



برای حل مسأله از دیاگرام زمانی روش سوئیچینگ بسته‌ای استفاده می‌شود. این روش مبتنی بر روش Store & Forward است و معمولاً زمان ارسال یک بسته با دریافت بسته قبل از آن همپوشان می‌شود.

$$t_{f_1} = \frac{\text{طول بسته}}{\text{نرخ ارسال بین سرور و سوئیچ اول}} = \frac{1000 \times 8b}{100 \times 10^6 \text{ bps}} = 0.108 \text{ ms}$$

t_{f_1} زمان ارسال بسته (شماره ۱ یا ۲) از سرور به سوئیچ اول به همین ترتیب t_{f_2} زمان ارسال بسته (شماره ۱ یا ۲) از سوئیچ اول به سوئیچ دوم و t_{f_3} زمان ارسال بسته (شماره ۱ یا ۲) از سوئیچ دوم به کلاینت است که به صورت زیر محاسبه می‌شوند:

$$t_{f_2} = \frac{\text{طول بسته}}{\text{نرخ ارسال بین سوئیچ اول و دوم}} = \frac{1000 \times 8b}{20 \times 10^6 \text{ bps}} = 0.14 \text{ ms}$$

$$t_{f_3} = \frac{\text{طول بسته}}{\text{نرخ ارسال بین سوئیچ دوم و کلاینت}} = \frac{1000 \times 8b}{50 \times 10^6 \text{ bps}} = 0.16 \text{ ms}$$

زمان تأخیر انتشار بین سرور و سوئیچ اول، $T_{p_1} = 1ms$ زمان تأخیر انتشار بین

سوئیچ اول و سوئیچ دوم و $T_{p_2} = 2ms$ زمان تأخیر انتشار بین سوئیچ دوم و کلاینت است.

زمان ۰: زمان ارسال بسته ۱ توسط سرور

زمان t_1 : زمان دریافت بسته ۱ توسط سوئیچ اول و همچنین زمان ارسال بسته ۱ توسط سوئیچ

$$t_1 = t_{p_1} + t_f = 1 + 0.08 = 1.08ms \quad \text{اول}$$

$$t_2 = 0 + t_f = 0 + 0.08 = 0.08ms \quad \text{زمان } t_2: \text{ زمان ارسال بسته ۲ توسط سرور}$$

$$t_3 = t_2 + t_f + t_{p_1} = 1.16ms \quad \text{زمان } t_3: \text{ زمان دریافت بسته ۲ توسط سوئیچ اول}$$

زمان t_4 : زمان ارسال بسته‌های بعدی (بسته سوم) و ...

$$t_5 = t_1 + T_{p_2} = 13.08ms \quad \text{زمان } t_5: \text{ زمان دریافت اولین بیت بسته ۱ توسط سوئیچ دوم}$$

$$t_6 = t_1 + t_f = 1.16ms \quad \text{زمان } t_6: \text{ زمان ارسال اولین بیت بسته ۲ توسط سوئیچ اول}$$

زمان t_7 : زمان دریافت آخرین بیت بسته ۱ توسط سوئیچ دوم و همچنین زمان ارسال بسته ۱

$$t_7 = t_5 + t_f = 13.48ms \quad \text{توسط سوئیچ دوم}$$

$$t_8 = t_7 + t_{p_2} = 15.48ms \quad \text{زمان } t_8: \text{ زمان دریافت اولین بیت بسته ۱ توسط کلاینت}$$

$$t_9 = t_7 + t_f = 13.64ms \quad \text{زمان } t_9: \text{ زمان ارسال آخرین بیت بسته ۱ توسط سوئیچ دوم}$$

زمان t_{10} : زمان دریافت آخرین بیت بسته ۱ (دریافت کامل بسته ۱) توسط کلاینت

$$t_{10} = t_8 + t_f = 15.64ms$$

زمان t_{11} : زمان ارسال اولین بیت بسته ۲ توسط سوئیچ دوم - لازم به ذکر است روش سوئیچینگ

بسته‌ای از روش Store & Forward استفاده می‌کند. بنابراین بایستی ابتدا سوئیچ دوم کل بسته ۲ را ذخیره کند تا آن را به سمت کلاینت هدایت کند.

$$t_{11} = t_6 + t_f + t_{p_2} = t_7 + t_f = 13.88ms$$

$$t_{12} = t_{11} + t_{p_2} = 15.88ms \quad \text{زمان } t_{12}: \text{ زمان دریافت اولین بیت بسته ۲ توسط کلاینت}$$

زمان t_{13} : زمان دریافت آخرین بیت بسته ۲ (دریافت کامل بسته ۲) توسط کلاینت

$$t_{13} = t_{11} + t_f + T_{p_2} = t_{12} + t_f = 16.04$$

$$= 0.4ms = t_{13} - t_{10} = \text{زمان دریافت کامل بسته ۱} - \text{زمان دریافت کامل بسته ۲} = \text{تأخیر دریافت بسته ۲ نسبت به}$$

بسته ۱

به همین ترتیب تأخیر دریافت بسته ۳ نسبت به بسته ۲ و تأخیر دریافت بسته ۴ نسبت به بسته ۳ و

... در نهایت تأخیر بسته ۵۰۰ نسبت به بسته ۴۹۹ همگی مساوی و برابر $0.4ms$ خواهد بود.

لذا زمان لازم برای انتقال پانصد بسته عبارت است از:

$$۲۱۵/۲۴ = ۴/۴۹۹ + ۱۵/۶۴ = \text{زمان تأخیر دریافت دو بسته متوالی} \times (۱-۵۰۰) + \text{زمان دریافت}$$

بسته اول

۵۶- گزینه (۱). به طور کلی انواع روش‌های ارسال پروتکل HTTP به چهار طبقه‌ی زیر تقسیم می‌شود:

(۱) روش پایدار ترتیبی

(۲) روش پایدار موازی

(۳) روش ناپایدار ترتیبی

(۴) روش ناپایدار موازی

(۱) روش پایدار و ترتیبی (متوالی) (Persistent)

درخواست و دریافت درخواست و برقراری زمان انتقال درخواست و درخواست و

Object ارتباط فایل اصلی دریافت فایل اصلی برقراری ارتباط

$$(RTT + RTT + T_F) + \left[\begin{array}{c} n \times (RTT) \\ \downarrow \\ \text{تعداد object ها} \end{array} + RTT + \sum_{i=1}^n T_F(i) \right]$$

زمان انتقال object

مثال: فرض کنید شخصی در مرورگر وب خود برای دریافت یک صفحه وب کلیک می‌کند. اگر آدرس IP مربوط به این URL در میزبان محلی وجود داشته باشد و فایل HTML مرتبط با این لینک دارای ۸ object باشد، در صورتی که زمان رفت و برگشت بین سرور و گیرنده و سرور دهنده ۱۰۰ میلی ثانیه و زمان ارسال object ها ناچیز باشد، با استفاده از پروتکل پایدار و ترتیبی از زمانی که شخص روی لینک کلیک می‌کند تا زمانی که صفحه وب را به طور کامل دریافت می‌کند برحسب میلی ثانیه چقدر طول می‌کشد؟

$$(RTT + RTT + T_F) + \left[\begin{array}{c} 8 \times (RTT) \\ \downarrow \\ \text{تعداد object ها} \end{array} + \sum_{i=1}^8 T_F(i) \right]$$

→ ۲۰۰ + ۸۰۰ = ۱۰۰۰ms

(۲) روش پایدار و موازی (Persistent)

درخواست موازی و درخواست و برقراری زمان انتقال درخواست و درخواست و

دریافت موازی Object ها ارتباط فایل اصلی دریافت فایل اصلی برقراری ارتباط

$$(RTT + RTT + T_F) + \left[\begin{array}{c} 1 \times (RTT) \\ \downarrow \\ \text{تعداد موازی} \end{array} + RTT + \sum_{i=1}^n T_F(i) \right]$$

زمان انتقال object

(دوای - ۱۲۹۲)

مثال: مسأله قبل را به روش پایدار و موازی حل کنید:

$$(RTT + RTT + T_F) + \left[\begin{array}{c} 1 \times (RTT) \\ \downarrow \\ \text{تعداد موازی} \end{array} + \sum_{i=1}^n T_F(i) \right]$$

درخواست موازی درخواست و برقراری زمان انتقال درخواست و درخواست و دریافت موازی ارتباط فایل اصلی دریافت فایل اصلی برقراری ارتباط

$$(RTT + RTT + T_F) + \left[\begin{array}{l} n \times (RTT + RTT) \\ \downarrow \\ \text{یک (موازی)} \end{array} + \sum_{i=1}^n T_F(i) \right]$$

زمان انتقال object

مثال: مسأله قبل را به روش ناپایدار و موازی حل کنید:

$$\left(\begin{array}{ccc} RTT & RTT & T_F \\ 100 & 100 & \text{صفر} \end{array} \right) + \left[\begin{array}{l} 1 \times (RTT + RTT) + \sum_{i=1}^n T_F(i) \\ \downarrow \\ \text{صفر} \end{array} \right]$$

$$\rightarrow 200 + 200 = 400$$

مثال ۲: مثال ۲ مسأله قبل را به روش ناپایدار و موازی حل کنید:

$$\left(\begin{array}{ccc} RTT & RTT & T_F \\ 40 & & 28/8 \end{array} \right) + \left[\begin{array}{l} 1 \times (RTT + RTT) + \sum_{i=1}^3 T_F(i) \\ \downarrow \\ 3 \times 28/8 \end{array} \right]$$

برای ۲ سگمنت ۲ بار درخواست و دریافت

$$2 \times 40 = 80$$

$$2 \times 40 = 80$$

$$240 + 115/2 = 355/2 \text{ ms}$$

توجه: یک درخواست برای برقراری ارتباط با مدت زمان ۴۰ms انجام می‌گردد. و یک درخواست موازی برای دریافت موازی ۳ object انجام می‌شود، اما از آنجا که هر object در دو سگمنت جا می‌شود، بنابراین در زمان $2 \times 40 = 80 \text{ ms}$ هر ۳ object دریافت می‌گردند.

با توجه به شرایط ذکر شده در صورت سؤال، مطابق آنچه گفتیم ابتدا بایستی فایل HTML را دریافت کرد و سپس ۹ فایل دیگر را دریافت کرد و حالا با توجه به نوع ارتباط که ناپایدار است و تنها می‌توان ۵ کانکشن موازی داشت بایستی اینگونه عمل کنیم:

ابتدا یک RTT صرف برقراری ارتباط می‌شود، سپس RTT دیگر صرف درخواست فایل و دریافت فایل می‌شود و پس از آن ارتباط قطع می‌شود (توجه داشته باشید که دریافت فایل نیز زمان‌بر است). بعد از دریافت فایل اول، با ۵ کانکشن موازی درخواست برقراری ارتباط را در یک RTT می‌دهیم و در RTT بعدی ۵ فایل را دریافت می‌کنیم و پس از آن ارتباط قطع می‌شود، سپس با توجه به اینکه تنها ۴ فایل باقیمانده است، یک RTT صرف برقراری ۴ کانکشن موازی و یک RTT دیگر صرف درخواست ۴ فایل می‌نماییم و محاسبات بدین ترتیب خواهد بود:

$$RTT + RTT + T_{F1} = 0.1 + 0.1 + \frac{5000 \times 8}{10^6} = 0.24 \text{ s}$$

$$RTT + RTT + T_{f\Delta} = 0.1 + 0.1 + \frac{5 \times 5000 \times 8}{1.6} = 0.4s$$

$$RTT + RTT + T_{f\gamma} = 0.1 + 0.1 + \frac{4 \times 5000 \times 8}{1.6} = 0.36s$$

$$Total = 0.24 + 0.4 + 0.36 = 1s$$

۵۷- گزینه (۱). ابتدا در این سؤال باید توجه داشت که منظور طراح Timeout نیست و تنها یک بسته به مقصد نرسیده است و با توجه به اینکه در TCP Reno پس از وقوع چنین رخدادی ابتدا اندازه آستانه ازدحام برابر با نصف اندازه پنجره کنونی می‌شود و سپس اندازه پنجره ارسال نصف می‌شود و با توجه به اینکه در آغاز ارسال پنجره ازدحام برابر ۴ است می‌توان این سوال را اینگونه حل کرد که بررسی کنیم در هر ارسال کدام بسته‌ها ارسال شده‌اند و با این شرایط ارسال ۳۲ بسته چه زمانی طول می‌کشد، در ادامه اندازه پنجره ارسال و بسته‌های دریافت شده توسط گیرنده در هر ارسال را نشان می‌دهیم:

اندازه پنجره ارسال:	۱	۲	۴	۵	۶	۷	۸	۴	۵
آخرین بسته دریافت شده:	۱	۳	۷	۱۲	۱۸	۲۵	۲۶	۳۰	۳۲

با توجه به توضیحات و روند فوق برای ارسال ۳۲ بسته ۹ زمان رفت و برگشت (RTT) صرف خواهد شد.

۵۸- گزینه (۲). توجه: این سؤال، دقیقاً همان سؤال دولتی ۹۱، مبحث لایه شبکه است. بهترین راه حل برای حل این مدل از مسأله‌ها، مشخص کردن محدودی هر زیرشبکه با استفاده از MASK آن زیر شبکه است. و سپس تعیین اینکه آدرس مقصد در کدام محدوده قرار دارد. توجه: در جدول مسیریابی، برای هر زیرشبکه، آدرس زیرشبکه یا NetID درج می‌گردد. توجه: بخش سمت راست نماد "/" به معنی طول MASK و بخش سمت چپ نماد "/" به معنی NetID زیر شبکه است.

زیر شبکه اول

برای زیر شبکه اول، با توجه به MASK آن که برابر ۲۲ است و به معنی وجود $2^{10} = 1024$ آدرس در این زیر شبکه است، بازه زیر را داریم:

۱۳۵ . ۴۶ . ۵۶ . ۰ → NetID

تا

۱۳۵ . ۴۶ . ۵۹ . ۲۵۵ → Broadcast

توجه: برای ساخت NetID باید، تعداد بیت‌های حاصل از تفاضل $10 = 22 - 32$ یعنی بخش HostID از سمت راست "۰" گردد و برای ساخت آدرس Broad cast، لازم است، تعداد بیت‌های حاصل از تفاضل $10 = 22 - 32$ یعنی بخش HostID از سمت راست "۱" گردد.

۱۲۸ ۶۴ ۳۲ ۱۶ ۸ ۴	۲۱
۱۳۵ . ۴۶ . ۰ . ۰ . ۰ . ۱ . ۱ . ۱ . ۰	۰۰ . ۰ . ۰ . ۰ . ۰ . ۰ . ۰ . ۰ . ۰ . ۰
Net ID	Host ID

۱۳۵ . ۴۶ . ۵۶ . ۰

<table border="1"> <tr> <td>۱۲۸</td><td>۶۴</td><td>۳۲</td><td>۱۶</td><td>۸</td><td>۴</td> </tr> <tr> <td>۱۳۵</td><td>۰</td><td>۴۶</td><td>۰</td><td>۰</td><td>۰</td> </tr> <tr> <td>۰</td><td>۰</td><td>۱</td><td>۱</td><td>۱</td><td>۰</td> </tr> </table> <p>Net ID</p>	۱۲۸	۶۴	۳۲	۱۶	۸	۴	۱۳۵	۰	۴۶	۰	۰	۰	۰	۰	۱	۱	۱	۰	<table border="1"> <tr> <td>۲۱</td> </tr> <tr> <td>۱۱</td><td>۰</td><td>۱</td><td>۱</td><td>۱</td><td>۱</td><td>۱</td><td>۱</td><td>۱</td><td>۱</td> </tr> </table> <p>Host ID</p>	۲۱	۱۱	۰	۱	۱	۱	۱	۱	۱	۱	۱
۱۲۸	۶۴	۳۲	۱۶	۸	۴																									
۱۳۵	۰	۴۶	۰	۰	۰																									
۰	۰	۱	۱	۱	۰																									
۲۱																														
۱۱	۰	۱	۱	۱	۱	۱	۱	۱	۱																					

۱۳۵ . ۴۶ . ۵۹ . ۲۵۵

زیر شبکه دوم

برای زیر شبکه دوم، با توجه به MASK آن که برابر ۲۲ است و به معنی وجود $2^{10} = 1024$ آدرس در این زیر شبکه است، بازه‌ی زیر را داریم:

NetID → ۱۳۵ . ۴۶ . ۶۰ . ۰
تا

Broadcast → ۱۳۵ . ۴۶ . ۶۳ . ۲۵۵

توجه: برای ساخت NetID باید، تعداد بیت‌های حاصل از تفاضل $10 = 32 - 22$ یعنی بخش HostID از سمت راست "۰" گردد و برای ساخت آدرس Broadcast، لازم است، تعداد بیت‌های حاصل از تفاضل $10 = 32 - 22$ یعنی بخش HostID از سمت راست "۱" گردد.

<table border="1"> <tr> <td>۱۲۸</td><td>۶۴</td><td>۳۲</td><td>۱۶</td><td>۸</td><td>۴</td> </tr> <tr> <td>۱۳۵</td><td>۰</td><td>۴۶</td><td>۰</td><td>۰</td><td>۰</td> </tr> <tr> <td>۰</td><td>۰</td><td>۱</td><td>۱</td><td>۱</td><td>۱</td> </tr> </table> <p>Net ID</p>	۱۲۸	۶۴	۳۲	۱۶	۸	۴	۱۳۵	۰	۴۶	۰	۰	۰	۰	۰	۱	۱	۱	۱	<table border="1"> <tr> <td>۲۱</td> </tr> <tr> <td>۰۰</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td> </tr> </table> <p>Host ID</p>	۲۱	۰۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۱۲۸	۶۴	۳۲	۱۶	۸	۴																									
۱۳۵	۰	۴۶	۰	۰	۰																									
۰	۰	۱	۱	۱	۱																									
۲۱																														
۰۰	۰	۰	۰	۰	۰	۰	۰	۰	۰																					

۱۳۵ . ۴۶ . ۶۰ . ۰

<table border="1"> <tr> <td>۱۲۸</td><td>۶۴</td><td>۳۲</td><td>۱۶</td><td>۸</td><td>۴</td> </tr> <tr> <td>۱۳۵</td><td>۰</td><td>۴۶</td><td>۰</td><td>۰</td><td>۰</td> </tr> <tr> <td>۰</td><td>۰</td><td>۱</td><td>۱</td><td>۱</td><td>۱</td> </tr> </table> <p>Net ID</p>	۱۲۸	۶۴	۳۲	۱۶	۸	۴	۱۳۵	۰	۴۶	۰	۰	۰	۰	۰	۱	۱	۱	۱	<table border="1"> <tr> <td>۲۱</td> </tr> <tr> <td>۱۱</td><td>۰</td><td>۱</td><td>۱</td><td>۱</td><td>۱</td><td>۱</td><td>۱</td><td>۱</td><td>۱</td> </tr> </table> <p>Host ID</p>	۲۱	۱۱	۰	۱	۱	۱	۱	۱	۱	۱	۱
۱۲۸	۶۴	۳۲	۱۶	۸	۴																									
۱۳۵	۰	۴۶	۰	۰	۰																									
۰	۰	۱	۱	۱	۱																									
۲۱																														
۱۱	۰	۱	۱	۱	۱	۱	۱	۱	۱																					

۱۳۵ . ۴۶ . ۶۳ . ۲۵۵

زیر شبکه سوم

برای زیر شبکه سوم، با توجه به MASK آن که برابر ۲۳ است و به معنی وجود $2^9 = 512$ آدرس در این زیر شبکه است، بازه‌ی زیر را داریم:

NetID → ۱۹۲ . ۵۳ . ۴۰ . ۰
تا

Broadcast → ۱۳۵ . ۵۳ . ۴۱ . ۲۵۵

توجه: برای ساخت NetID باید، تعداد بیت‌های حاصل از تفاضل $9 = 32 - 23$ یعنی بخش HostID از سمت راست "۰" گردد و برای ساخت آدرس Broadcast، لازم است، تعداد بیت‌های حاصل از تفاضل $9 = 32 - 23$ یعنی بخش HostID از سمت راست "۱" گردد.

<table border="1"> <tr> <td>۱۲۸</td><td>۶۴</td><td>۳۲</td><td>۱۶</td><td>۸</td><td>۴</td><td>۲</td> </tr> <tr> <td>۱۹۲</td><td>۰</td><td>۵۳</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td> </tr> <tr> <td>۰</td><td>۰</td><td>۱</td><td>۰</td><td>۱</td><td>۰</td><td>۰</td> </tr> </table> <p>Net ID</p>	۱۲۸	۶۴	۳۲	۱۶	۸	۴	۲	۱۹۲	۰	۵۳	۰	۰	۰	۰	۰	۰	۱	۰	۱	۰	۰	<table border="1"> <tr> <td>۱</td> </tr> <tr> <td>۰</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td><td>۰</td> </tr> </table> <p>Host ID</p>	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۱۲۸	۶۴	۳۲	۱۶	۸	۴	۲																											
۱۹۲	۰	۵۳	۰	۰	۰	۰																											
۰	۰	۱	۰	۱	۰	۰																											
۱																																	
۰	۰	۰	۰	۰	۰	۰	۰	۰	۰																								

۱۹۲ . ۵۳ . ۴۰ . ۰

۱۹۲	۰۵۳	۰۰۰	۱۲۸	۰۶۴	۰۳۲	۰۱۶	۰۰۸	۰۰۴	۰۰۲	۱	۰	۱	۱	۱	۱	۱	۱	۱	۱
Net ID										Host ID									

۱۹۲.۰۵۳.۰۴۱۰.۲۵۵

حال با توجه به اینکه آدرس مقصد برابر با مقدار ۲ . ۵۲ . ۴۶ . ۱۳۵ است و این آدرس در محدوده هیچیک از سه زیر شبکه فوق قرار ندارد، بنابراین گام بعدی بسته مورد نظر، Router۲ است.

۵۹- گزینه (۳). رابطه کلی برای محاسبه بهره‌وری در stop & wait (IDLE) به صورت زیر است:
 محاسبه بهره‌وری stop & wait بدون صرف نظر کردن از سر بار Ack، Header و زمان پردازش به صورت زیر است:

$$U_{\text{stop \& wait}} = \begin{cases} \frac{1}{w_s} \left(1 - \frac{H}{L}\right) (1 - P_f) & w < w_s \quad (1) \\ \left(1 - \frac{H}{L}\right) \times (1 - P_f) & w \geq w_s \quad (2) \end{cases}$$

توجه: برای بهره‌وری ۱۰۰٪ باید $w \geq w_s$ باشد، بنابراین از رابطه دوم استفاده می‌گردد.

$$w_s = \frac{T_O}{T_F}$$

برابر با «حداقل» اندازه پنجره که برای «ماکزیم کردن راندمان» مورد نیاز است.

T_F و T_O از رابطه‌ی زیر به دست می‌آید:

$$T_O = \text{Total Delay} = T_F + 2T_P + T_{\text{Ack}} + 2T_{\text{Process}}$$

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

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

محاسبه بهره‌وری stop & wait با صرف نظر کردن از سر بار Ack، Heder، و زمان پردازش به صورت زیر است:

$$w_s = \frac{T_F + 2T_P + \cancel{\text{Ack}} + 2\cancel{\text{process}}}{T_F} = \frac{T_F + T_P}{T_F} = \frac{1 + 2\frac{T_P}{T_F}}{1} = \frac{1 + 2a}{1} = 1 + 2a$$

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

بنابراین با توجه به روابط فوق داریم:

$$U_{\text{stop \& wait}} = \begin{cases} \frac{1}{1 + 2a} (1 - P_f) & w < 1 + 2a \quad (1) \\ (1 - P_f) & w \geq 1 + 2a \quad (2) \end{cases}$$

توجه: برای بهره‌وری ۱۰۰٪ باید $w \geq 1 + 2a$ باشد، بنابراین از رابطه دوم استفاده می‌گردد.

از آنجا که مطابق فرض سؤال، از سر بار پیام Ack و سر بار فریم (Header) صرف نظر شده است و مقداری برای زمان پردازش در نظر گرفته نشده است، بنابراین مطابق رابطه دوم، بخش دوم داریم:

$$U_{\text{stop \& wait}} = \frac{1}{1+2a} \times (1 - P_F) \rightarrow \frac{0.8}{1+2a} \geq 0.6$$

$$0.8 \geq 0.6(1+2a) \rightarrow$$

$$8 \geq 6(1+2a) \rightarrow$$

$$8 \geq 6 + 12a \rightarrow$$

$$2 \geq 12a \rightarrow \frac{1}{6} \geq a \rightarrow \frac{1}{6} \geq \frac{T_P}{T_F}$$

$$T_F \geq 6T_P \rightarrow \frac{T_F}{T_P} \geq 6$$

بنابراین مدت زمان انتقال فریم، باید حداقل ۶ برابر تأخیر انتشار یک طرفه باشد.

۶۰- گزینه (۲). CSMA/CD، یک روش مدیریت کانال پخش همگانی است.

توجه: روش CSMA/CD، دارای قدرت شوند و کشف تصادم است.

توجه: مطابق ویژگی الگوریتم CSMA/CD، شرط کشف تصادم در این روش به صورت زیر است:

$$T_F \geq 2T_P$$

$$\frac{L}{R} \geq 2 \frac{D}{V}$$

فرض کنید، اگر حادثه‌ای بر اثر ریزش کوه در جاده رخ دهد، فردی فرستاده شده از سوی شما وظیفه دارد، پس از مشاهده حادثه، وقوع این رخداد را به اطلاع شما برساند، در مدت زمان یک رفت و برگشت، البته تا وقتی که کار از کار نگذشته باشد. اگر فریم در کانال قرار گیرد، دیگر جلوی فریم را نمی‌توان گرفت، در اینجا هم فرض کنید اگر خودروی شما از منزل خارج شود، دیگر جلوی خودرو را نمی‌توان گرفت، و شما با ریزش کوه برخورد خواهید کرد، در حالی که پیام‌رسان جهت اطلاع‌رسانی به شما در راه بود ولی دیر می‌رسد. راه‌حل این است که انقدر اسباب و وسایل (وسایل اصلی و وسایل فرعی) داشته باشید که مدام داخل ماشین قرار دهید که سبب معطلی شما گردد، انقدری که پیام‌رسان برسد، تا دیر نشده است برسد، قبل از حرکت شما برسد. بارها شده است، قصد حرکت به مقصدی را دارید، اما عده‌ای به دلیل با خبر بودن از مسأله‌ای، به هر بهانه‌ای جلوی حرکت شما را می‌گیرند، شما را معطل می‌کنند، با تلفن و با هر امکانی که مانع‌تان شود، تا خبر واقعی از راه برسد.

توجه: بنابر مطلب فوق، مدت زمان انتقال فریم (T_F) به کانال، مشابه قرار دادن وسایل (اصلی و فرعی) در خودرو باید برابر یا بزرگتر از زمان یک رفت و برگشت ($2T_P$)، مشابه رفت و برگشت فرد پیام‌رسان، باشد، تا فرصت جلوگیری از وقوع تصادم ایجاد گردد.

بر اساس رابطه فوق، حداقل طول فریم، به صورت $L = \text{Data} + \text{Header}$ مشخص می‌شود.
 $(L = \text{Data} + 25\text{Byte})$

بر اساس اطلاعات سؤال، برای طول کانال (D)، نرخ انتقال (R) و سرعت انتشار (V) داریم:

$$D = 1\text{KB} = 1000\text{m}, R = 100\text{mbps} = 100 \times 10^6 \text{bps}$$

$$V = 200\text{mpms} = 200 \frac{\text{m}}{\text{ms}} = \frac{200}{10^{-6}} = 2 \times 10^8 \frac{\text{m}}{\text{s}} = 2 \times 10^8 \text{mps}$$

حال در ادامه، با توجه به مقادیر فوق و رابطه مطرح شده داریم:

$$\frac{L}{R} \geq 2 \frac{D}{V} \rightarrow \frac{\text{Data} + 25 \times 8}{100 \times 10^6} \geq 2 \times \frac{1000}{2 \times 10^8}$$

$$\text{Data} + 200 \geq 1000 \rightarrow \text{Data} \geq 800\text{bit} \rightarrow \text{Data} \geq 100\text{Byte}$$

اصول و مبانی مدیریت

۶۱- گزینه (۱). تخصصی کردن کار باعث ظهور متخصصانی می‌شود که به همکاری با یکدیگر نیاز دارند. این همکاری با قرار دادن متخصصان با یکدیگر درون واحدها و تحت نظر مدیر تشدید می‌شود. ایجاد چنین واحدی نوعاً بر پایه این عوامل قرار دارد:
 وظایف کاری که به اجرا در می‌آید (گروه بندی مبتنی بر وظیفه)
 تولید یا خدمتی که عرضه می‌شود (گروه بندی مبتنی بر محصول یا تولید)
 هدف مشتری یا ارباب رجوع (گروه بندی بر اساس مشتری)
 قلمرو جغرافیایی که پوشش داده می‌شود (گروه بندی جغرافیایی)
 فرآیندی که به کار گرفته می‌شود (گروه بندی بر مبنای فرآیند) تا درون داد را به برون داد تبدیل نماید.

۶۲- گزینه (۱). سه ویژگی مشترک همه سازمانها:

هر سازمانی هدفی دارد و متشکل از افرادی است که به شکلی گرد هم آمده‌اند.
 ویژگی اول این است که هدف مشخص یک سازمان معمولاً به صورت یک هدف یا مجموعه‌ای از اهداف بیان می‌شود.
 ویژگی دوم این است که هیچ قصد یا هدفی به خودی خود قابل دستیابی نیست.
 ویژگی سوم اینکه همه سازمانها از ساختار سازمانی، برای هدایت و کنترل رفتار اعضای خود استفاده می‌کنند.
 بنابراین، سازمان به هر مؤسسه‌ای اطلاق می‌شود که دارای هدفی مشخص، افراد یا اعضا و یک ساختار سیستماتیک باشد.

توضیح بیشتر:

سازمان ۱. یک نهاد اجتماعی است ۲. که مبتنی بر هدف باشد ۳. ساختار آن به صورت آگاهانه طرح ریزی شده و دارای سیستمهای هماهنگ و فعال است و سرانجام ۴. با محیط خارجی ارتباط دارد.