موسسه بابان

انتشارات بابان و انتشارات راهیان ارشد

درس و کنکور ارشد

شبکه‌های کامپیوتری

(حل تشریحی سوالات دولتی ۱۳۹۵)

ویژه‌ی داوطلبان کنکور کارشناسی ارشد مهندسی کامپیوتر و IT

براساس کتب مرجع

کراس راس و لئون گارسیا

ارسطو خلیلی‌فر

تست‌های سال ۹۵

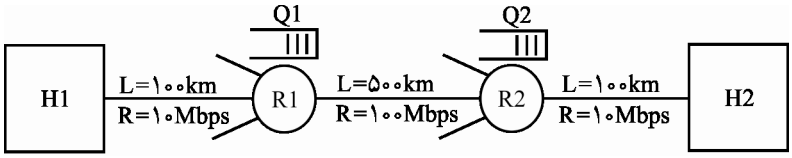
۱- ۱۰۰۰ ایستگاه برای ارسال اطلاعات خود از یک کانال مشترک با روش دسترسی Slotted ALOHA استفاده می‌کنند. اگر هر ایستگاه به طور متوسط ۱۸ ارسال در یک ساعت داشته و اندازه هر برش زمانی (time slot) ۱۰۰ میکروثانیه باشد، بار کانال برحسب تعداد ارسال در هر برش زمانی برابر است با:

۱) ۰/۰۰۵ ۲) ۰/۰۱۰ ۳) ۰/۰۱۵ ۴) ۰/۰۲۰

۲- یک کد همینگ ۷ بیتی با مقدار **1011011** به گیرنده می‌رسد. مقدار صحیح این کد چیست؟

۱) 1011001 ۲) 1011011 ۳) 1011010 ۴) 1011111

** یک جریان داده (صوت و ویدیو) با نرخ ۵ مگابیت بر ثانیه که در بسته‌های ۱۰۰۰۰ بیتی قرار دارند از برنامه‌ای در کامپیوتر **H1** به برنامه‌ای در کامپیوتر **H2** مطابق با شکل زیر، در حال ارسال است. فرض کنید طول صف در بافرهای مسیریاب‌های R_1 و R_2، حداکثر ۴ بسته و حداقل صفر است. با توجه به اطلاعات داده شده در شکل به سؤال‌های ۳ و ۴ پاسخ دهید.



Propagation Speed $= ۲\times۱۰^{۸}$ mps

۳- حداقل و حداکثر تأخیر یک بسته از زمان ارسال از کامپیوتر **H1** و دریافت توسط کامپیوتر **H2** به میلی‌ثانیه چقدر است؟

۱) ۱۳/۶ ،۵/۶ ۲) ۱۳/۶ ،۹/۶ ۳) ۲۱/۶ ،۵/۶ ۴) ۲۱/۶ ،۹/۶

۴- حداقل تأخیر زمانی به میلی‌ثانیه که برنامه در کامپیوتر **H2** باید بافر خود را قبل از پخش داده مشاهده (محاسبه) کند و حداقل اندازه بافر مورد نیاز به کیلو بیت برای پخش بدون وقفه این جریان چقدر هستند؟

۱) ۸۰ ،۸ ۲) ۸۰ ،۱۶ ۳) ۱۶۰ ،۸ ۴) ۱۶۰ ،۱۶

۵- فرض کنید یک برنامه سرویس گیرنده (Client) بعد از پیدا کردن آدرس IP کامپیوتر سرویس‌دهنده (Server) می‌خواهد یک صفحه وب که اندازه فایل اصلی آن ۲۰۰ کیلوبیت و اندازه هر یک از ۳ تصویر قرار گرفته در آن ۳۰۰ کیلوبیت است را از طریق پروتکل HTTP غیرمداوم (Non-Persistent HTTP) که مجاز به ایجاد اتصال موازی نیز است، دریافت کند. اگر زمان رفت و

برگشت (RTT)۲۰۰ میلی ثانیه، نرخ ارسال هر اتصال ۱۰ مگابیت بر ثانیه و اندازه پیام‌های GET ناچیز باشد، تأخیر دریافت کامل این صفحه وب به میلی ثانیه چقدر است؟

۱) ۲۱۱۰ ۲) ۱۶۵۰ ۳) ۱۲۵۰ ۴) ۸۲۰

۶- برنامه مدیریتی Trace Route به چه منظور استفاده می‌شود و از کدام پیام(هـای) پروتکـل ICMP استفاده می‌کند؟

۱) برای تست اتصال مسیر استفاده می‌شود و از پیام‌های Echo Request و Echo Reply استفاده می‌کند.

۲) برای کشف مسیر استفاده می‌شود و از پیام‌های Echo Request و Echo Reply استفاده می‌کند.

۳) برای تست اتصال مسیر استفاده می‌شود و از پیام Time Exceeded استفاده می‌کند.

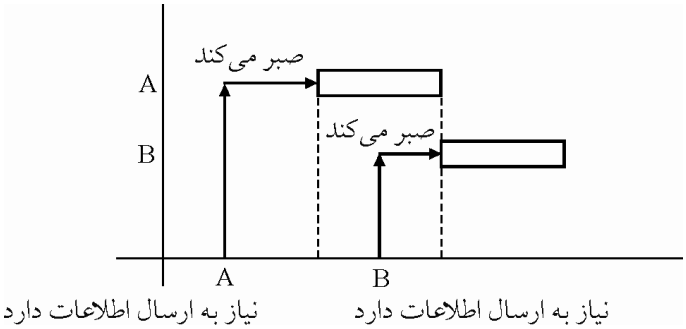
۴) برای کشف مسیر استفاده می‌شود و از پیام Time Exceeded استفاده می‌کند.

پاسخ تست‌های سال ۹۵

۱- گزینه (۱) صحیح است.

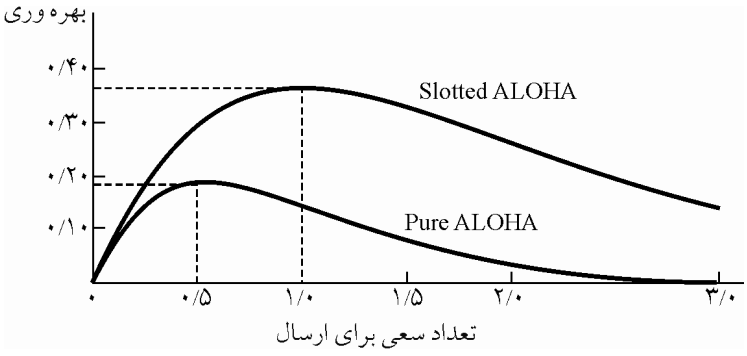
برای کاهش احتمال برخورد فریم‌های ارسالی توسط چند فرستنده از روش Slotted-Aloha استفاده می‌شود. در این روش زمان به بخش‌های مساوی به نام اسلات زمانی تقسیم می‌شود. هر کدام از این بخش‌ها حداکثر زمان کافی برای انتقال یک فریم داده می‌باشد. هر ایستگاه هر زمان که نیاز به ارسال اطلاعات داشته باشد، صبر می‌کند و در شروع اسلات زمانی و بدون بررسی کانال اقدام به انتقال اطلاعات می‌کند در این صورت فقط زمانی برخورد رخ می‌دهد که حداقل دو ایستگاه در ابتدای این اسلات زمانی، اقدام به ارسال داده نمایند.

در این روش احتمال برخورد به میزان قابل توجهی کاهش می‌یابد، به شکل زیر توجه کنید.



در صورتی که برخورد رخ دهد، هر کدام از ایستگاه‌های درگیر یک مدت زمان تصادفی صبر می‌کنند و دوباره مراحل ذکر شده را تکرار می‌کنند. توجه داشته باشید، زمان‌های تصادفی برای هر ایستگاه به شکل جداگانه محاسبه می‌شود.

توجه: شکل زیر رابطه G با بهره‌وری را در دو روش P-ALOHA و S-ALOHA، نمایش می‌دهد:



بازده کانال در روش‌های ALOHA

توجه: احتمال برخورد در P-Aloha نسبت به S-Aloha بیش‌تر است. زیرا در S-Aloha احتمال برخورد فقط در ابتدای ارسال فریم در یک اسلات زمانی خاص وجود دارد. ولی در P-Aloha در هر زمانی از ارسال فریم امکان برخورد وجود دارد. پس کارایی و بهره وری از کانال، در S-Aloha بیشتر است. حدود دو برابر.

توجه: این پروتکل reservation ALOHA یا r-ALOHA نیز نامیده می‌شود.

در یک بیان کلی در روش S-ALOHA، زمان به قسمت‌هایی که آن را اِسلات زمانی (slot) می‌نامند، تقسیم می‌شود. شروع ارسال فریم‌ها تنها در آغاز هر اسلات امکان‌پذیر است (اما در P-ALOHA، در هر زمانی می‌توانست ارسال آغاز شود).

توجه: اندازه اسلات برابر زمان انتقال کامل یک فریم است. $([T_F]+T_P)$

توجه: هر وقت فریمی برای انتقال توسط یک ایستگاه آماده شد، باید تا شروع اسلات بعدی صبر کند.

شبه کد S-ALOHA

۱- اگر فریمی برای ارسال داری، **صبر کن**، تا نوبتت شود (نوبت اسلات بعدی برسد). آنگاه بدون بررسی کانال (عدم شنود کانال) آن را ارسال کن.

۲- اگر فریم‌ها با تصادم مواجه شدند، پس از گذشتن زمان تصادفی، آن‌ها را مجدداً ارسال کن.

روابط S-ALOHA

$$U_{S-ALOHA} = Ge^{-G} \qquad G^{Max}_{S-ALOHA} = ۱ \qquad U^{Max}_{S-ALOHA} = ۰/۳۶۸ = ۳۶/۸\%$$

G : تعداد تلاش‌ها برای ارسال فریم‌ها توسط ایستگاه‌ها در زمان اسلات (T_{slot}).

توجه: مطابق تعریف، مقدار G برابر است با تعداد تلاش‌ها برای ارسال فریم‌ها توسط ایستگاه‌ها در زمان یک اسلات (T_{slot})، بنابراین مطابق این تعریف برای محاسبه مقدار G داریم:

زمان	درخواست
۱s	$M\lambda$
T_{slot}	G

$$\rightarrow \quad G = M\lambda T_{slot}$$

که M برابر تعداد ایستگاه‌ها و λ برابر تعداد فریم‌های ارسالی هر ایستگاه در واحد زمان می‌باشد.

احتمال ارسال موفق یک فریم در همان بار اول $= e^{-G}$

احتمال ارسال موفق پس از بروز k تصادم $= (۱-e^{-G})^k e^{-G}$

تعداد تلاش برای ارسال موفق (امید ریاضی) $= E(x) = \frac{۱}{P} = \frac{۱}{e^{-G}} = e^{G}$

توجه: S-ALOHA فقط حالت اول و حالت دوم تصادم را دارد. زیرا زمانی که فریمی در حال حرکت در کانال است، در یک اسلات تا به مقصد برسد، فریم دیگری در کانال قرار نمی‌گیرد، در

واقع بقیه پشت اسلات می‌مانند.

توجه: مسئله تصادم حالت سوم در روش S-ALOHA توسط مکانیزم اسلات زمانی حل می‌گردد.

مثال: ایستگاه‌های شبکه S-ALOHA در هر ثانیه ۵۰ درخواست تولید می‌کنند، در صورتی که زمان به اسلات‌های ۴۰ms تقسیم شده باشد، به ترتیب احتمال موفقیت در همان بار اول چقدر است؟ احتمال موفقیت بعد از اتفاق دقیقاً k تصادم چقدر است؟ تعداد تلاش مورد انتظار برای ارسال موفق چقدر است؟

پاسخ:

$$\begin{matrix} \text{ثانیه} & \text{درخواست} \\ ۱ & M\lambda = ۵۰ \\ T_{slot} = ۴۰\times۱۰^{-۳} & G \end{matrix} \rightarrow \quad G = M\lambda T_{slot} = ۵۰\times۴۰\times۱۰^{-۳} = ۲$$

احتمال ارسال موفق یک فریم در همان بار اول برابر e^{-G} یعنی $e^{-۲}$ است.

احتمال ارسال موفق پس از بروز k تصادم برابر $(۱-e^{-G})^k e^{-G}$ یعنی $(۱-e^{-۲})^k e^{-۲}$ است.

تعداد تلاش برای ارسال موفق (امید ریاضی) برابر با $\frac{۱}{e^{-G}}$ یا e^{G} یعنی $e^{۲}$ است.

در صورت سوال مطرح شده است که **۱۰۰۰۰ ایستگاه برای ارسال اطلاعات خود از یک کانال مشترک با روش دسترسی Slotted ALOHA استفاده می‌کنند. همچنین گفته شده است که اگر هر ایستگاه به طور متوسط ۱۸ ارسال در یک ساعت داشته و اندازه هر برش زمانی (time slot) ۱۰۰ میکروثانیه باشد، خواسته شده است که بار کانال برحسب تعداد ارسال در هر برش زمانی چقدر است؟** کارایی کانال در روش Slotted ALOHA از رابطه $U_{\text{Slotted ALOHA}} = G\times e^{-G}$ محاسبه می‌شود. G عبارتست از تعداد تلاش برای ارسال فریم در واحد زمان که واحد زمانی، زمان لازم برای ارسال یک فریم است (T_{slot}) .

$$\begin{matrix} \text{ثانیه} & \text{درخواست} \\ ۳۶۰۰ & ۱۸\times۱۰^{۴} \\ ۱ & M\lambda \end{matrix} \Rightarrow \quad M\lambda = ۵۰$$

یعنی تمامی ایستگاه‌ها (روی هم) ۵۰ درخواست در ثانیه تولید می‌کنند:

$$\left.\begin{matrix} \text{ثانیه} & \text{درخواست} \\ ۱ & ۵۰ \\ ۱۰۰\times۱۰^{-۶} & G \end{matrix}\right\} \Rightarrow \quad G = \frac{۱}{۲۰۰} = ۰/۰۰۵$$

مقدار G را می‌توان از رابطه ز یر نیز محاسبه نمود:

$$G = M\lambda T_{Slot} = ۵۰\times۱۰۰\times۱۰^{-۶} = \frac{۱}{۲۰۰} = ۰/۰۰۵$$

یعنی تمامی ایستگاه‌ها (روی هم) به طور میانگین $\frac{۱}{۲۰۰}$ درخواست در هر برش زمانی تولید می‌کنند

که همان G است. بنابراین گزینه اول پاسخ سوال است.
همچنین کارایی کانال از رابطه زیر قابل محاسبه است:

$$U_{\text{Slotted ALOHA}} = Ge^{-G} = \frac{۱}{۲۰۰}\, e^{-\frac{۱}{۲۰۰}}$$

۲- گزینه (۳) صحیح است.

برای اینکه بتوان خطا را در داده‌های ارسالی تشخیص داد و یک بیت خطا را در آن تصحیح نمود. از روش کد همینگ استفاده می‌شود. کد همینگ را با یک مثال توضیح می‌دهیم:
فرض کنید فرستنده قصد ارسال داده ۱۰۱۰ را دارد و طبق توافق بین فرستنده و گیرنده قرار است از کد همینگ برای کشف خطا استفاده شود. برای اینکه بتوان خطا را در داده‌ای ارسالی تشخیص داد باید برای هر m بیت داده اصلی، r بیت داده فرعی (کنترلی یا افزونه) اضافه کرد. به این شکل که داده‌های فرعی در بیت‌هایی با اندیس‌هایی از توان دو (۱ و ۲ و ۴ و ۸ و ...) و داده‌های اصلی در اندیس‌های باقی‌مانده (۳ و ۵ و ۶ و ۷) قرار می‌گیرند. به شکل زیر توجه کنید:

$r_۱$	$r_۲$	$m_۳$	$r_۴$	$m_۵$	$m_۶$	$m_۷$
؟	؟	۱	؟	۰	۱	۰

برای بدست آوردن مقادیر بیت‌های فرعی به شکل زیر عمل می‌شود:
ابتدا باید شماره اندیس بیت‌های داده اصلی را با استفاده از اعداد توان ۲ بدست آوریم، به عنوان مثال عدد ۷ از مجموع اعداد ۴ و ۲ و ۱ که همه آنها اعداد توان ۲ هستند، بدست می‌آید:

$۱ + ۲ + ۴ = ۷$

$۲ + ۴ = ۶$

$۱ + ۴ = ۵$

$۱ + ۲ = ۳$

حال برای بدست آوردن مقدار بیت‌های فرعی کافی است مقدار اندیس‌های داده اصلی که اندیس بیت فرعی در بدست آوردن شماره اندیس آنها نقش داشته است، با هم XOR شوند و در اندیس مورد نظر قرار گیرند. به عنوان مثال برای بدست آوردن $r_۱$ باید مقادیر بیت‌های با اندیس ۷ و ۵ و ۳ را با هم XOR نمود:

$r_۱ = m_۳ \oplus m_۵ \oplus m_۷ \Rightarrow r_۱ = ۱ \oplus ۰ \oplus ۰ = ۱$

$r_۲ = m_۳ \oplus m_۶ \oplus m_۷ \Rightarrow r_۲ = ۱ \oplus ۱ \oplus ۰ = ۰$

$r_۴ = m_۵ \oplus m_۶ \oplus m_۷ \Rightarrow r_۴ = ۰ \oplus ۱ \oplus ۰ = ۱$

در نتیجه کد ارسالی برابر می‌شود با:

$r_۱$	$r_۲$	$m_۳$	$r_۴$	$m_۵$	$m_۶$	$m_۷$
۱	۰	۱	۱	۰	۱	۰

در یک راه‌حل دیگر، می‌توان بدون در نظر داشتن روابط کد همینگ، مقادیر داده‌های فرعی را به سادگی از روی مقادیر داده‌های اصلی استخراج نمود.
مطابق فرض مثال مطرح شده، فرستنده قصد ارسال عدد ۱۰ با فرمت باینری ۱۰۱۰ را در قالب کد همینگ دارد. بنابراین ۳ بیت داده فرعی مطابق الگوی همینگ باید به داده اصلی اضافه گردد. بنابراین داریم:

r_1	r_2	m_3	r_4	m_5	m_6	m_7
؟	؟	۱	؟	۰	۱	۰

برای استخراج داده‌های کنترلی از روی داده‌های اصلی جدول زیر مورد استفاده قرار می‌گیرد:

شماره بیتی که محتوی آن یک باشد	معادل باینری		
۳	۰	۱	۱
۶	۱	۱	۰
	$r_4=1$	$r_2=0$	$r_1=1$

در جدول فوق شماره بیت‌هایی از داده اصلی که مقدار ۱ دارند در ستون مربوطه درج می‌شود، سپس در بخش معادل باینری به صورت عمودی (ستونی) عمل XOR انجام می‌شود، که نتیجه حاصل به ترتیب از چپ به راست $r_1=1$، $r_2=0$ و $r_4=1$ خواهد بود.
بنابراین داده‌ای که باید ارسال شود به صورت زیر خواهد بود:

r_1	r_2	m_3	r_4	m_5	m_6	m_7
۱	۰	۱	۱	۰	۱	۰

کنترل خطا در کد همینگ

گیرنده با دریافت داده‌ها، برای عمل کنترل خطا مقدار اندیس‌های داده که اندیس بیت افزونه در بدست آوردن شماره اندیس آنها نقش دارد با مقدار همان بیت افزونه XOR می‌کند، اگر نتیجه ۰ باشد خطایی رخ نداده است:

$$s_1 = r_1 \oplus m_3 \oplus m_5 \oplus m_7 \Rightarrow s_1 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$s_2 = r_2 \oplus m_3 \oplus m_6 \oplus m_7 \Rightarrow s_2 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

$$s_4 = r_4 \oplus m_5 \oplus m_6 \oplus m_7 \Rightarrow s_4 = 1 \oplus 0 \oplus 1 \oplus 0 = 0$$

در صورتی که در یک بیت، خطا رخ داده باشد محل وقوع خطا را می‌توان با معادل مقدار باینری $s_4 s_2 s_1$ بدست آورد. مطابق صورت سوال گیرنده داده زیر را دریافت نموده است.

r_1	r_2	m_3	r_4	m_5	m_6	m_7
۱	۰	۱	۱	۰	۱	۱

$s_۱ = r_۱ \oplus m_۳ \oplus m_۵ \oplus m_۷ \Rightarrow s_۱ = ۱ \oplus ۱ \oplus ۰ \oplus ۱ = ۱$

$s_۲ = r_۲ \oplus m_۳ \oplus m_۶ \oplus m_۷ \Rightarrow s_۲ = ۰ \oplus ۱ \oplus ۱ \oplus ۱ = ۱$

$s_۴ = r_۴ \oplus m_۵ \oplus m_۶ \oplus m_۷ \Rightarrow s_۴ = ۱ \oplus ۰ \oplus ۱ \oplus ۱ = ۱$

$s_۴ s_۲ s_۱ = (۱۱۱)_۲ = (۷)_{۱۰}$

در نتیجه بیت شماره ۷ دچار خطا شده است و قابل تصحیح است. و باید به صورت زیر تصحیح گردد.

$r_۱$	$r_۲$	$m_۳$	$r_۴$	$m_۵$	$m_۶$	$m_۷$
۱	۰	۱	۱	۰	۱	۰

بنابراین گزینه سوم پاسخ سوال است.

در یک راه حل دیگر، می‌توان بدون در نظر گرفتن روابط کد همینگ، محل وقوع خطا را از روی داده‌های اصلی و فرعی استخراج نمود.

مطابق مثال مطرح شده، فرض کنید گیرنده، داده خطادار زیر را در الگوی کد همینگ دریافت کرده است.

$r_۱$	$r_۲$	$m_۳$	$r_۴$	$m_۵$	$m_۶$	$m_۷$
۱	۰	۱	۱	۰	۱	۱

در حالی که باید داده زیر را دریافت می‌کرد:

$r_۱$	$r_۲$	$m_۳$	$r_۴$	$m_۵$	$m_۶$	$m_۷$
۱	۰	۱	۱	۰	۱	۰

در واقع در بیت پنجم یعنی $m_۷$ خطا رخ داده است.

برای کشف محل وقوع خطا، جدول زیر مورد استفاده قرار می‌گیرد.

شماره بیتی که محتوی آن یک باشد	معادل باینری		
۱	۰	۰	۱
۳	۰	۱	۱
۴	۱	۰	۰
۶	۱	۱	۰
۷	۱	۱	۱
	$s_۴ = ۱$	$s_۲ = ۱$	$s_۱ = ۱$

در جدول فوق شماره بیت‌هایی از داده اصلی و داده فرعی کد دریافتی توسط گیرنده که مقدار ۱

دارند در ستون مربوطه درج می‌شود، سپس در بخش معادل باینری به صورت عمودی (ستونی) عمل XOR انجام می‌شود، که نتیجه حاصل به ترتیب از چپ به راست $s_1=1$، $s_2=1$ و $s_4=1$ خواهد بود.

بنابراین مطابق الگوی زیر:

$$s_4 s_2 s_1 = (111)_2 = (7)_{10}$$

در بیت شماره ۷ یعنی m_7 خطا رخ داده است و باید به صورت زیر تصحیح گردد.

r_1	r_2	m_3	r_4	m_5	m_6	m_7
۱	۰	۱	۱	۰	۱	۰

بنابراین گزینه سوم پاسخ سوال است.

۳- گزینه () صحیح است.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه خود، گزینه سوم را به عنوان پاسخ اعلام کرده بود. اما در کلید نهایی این سوال حذف گردید، که کار درستی بوده است.

توجه: در شبکه‌های کامپیوتری چهار نوع تأخیر داریم:

تأخیر انتقال (T_F)، تأخیر انتشار (T_{Prop})، تأخیر صف (T_{queue})، تأخیر پردازش $(T_{process})$.

توجه: تأخیر صف‌بندی داخل گره‌ها، یک تأخیر متغیر است که به حجم ترافیک لحظه عبور از آن گره بستگی دارد. به عبارت دیگر تاخیر صف در طول زمان نوسان دارد. پس تأخیری که از ابتدا به انتها ایجاد می‌شود، متغیر است و از قبل قابل پیش‌بینی نیست.

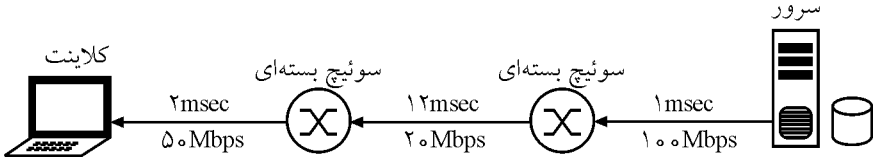
مثال: مثلاً دسترسی به سیستم آموزشی (پرتال)

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)

مثال- در شبکه‌ای با مسیر شکل زیر بین سرور و کلاینت وجود دارد، حداقل زمان لازم برای انتقال پانصد بسته هزار بایتی بر حسب میلی‌ثانیه (msec) کدام است؟ (توجه: $1Mbps = 10^6 bps$)



۱) ۲۱۵ ۲) ۲۱۵/۱۲ ۳) ۲۱۵/۲۴ ۴) ۲۱۵/۶۴

پاسخ- گزینه (۳) صحیح است.

به طور کلی حداقل زمان لازم برای انتقال بسته‌ها مابین دو گره انتهایی از رابطه زیر محاسبه می‌گرد:

$$T_{\text{Total Delay}} = \left[T_{F۱}\right] + T_{Prop۱} + \left[T_{process۱} + T_{F۲}\right] + T_{Prop۲} + \left[T_{process۲} + T_{F۳}\right] + T_{Prop۳} + T_{queue}$$

T_F از رابطه زیر بدست می‌آید:

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

T_F، زمان انتقال بسته به داخل کانال انتقال است.

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

T_{Prop} از رابطه زیر بدست می‌آید:

$$T_{Prop} = \frac{D}{V}$$

T_{Prop}، زمان تأخیر انتشار است.

که D برابر طول کانال و V برابر سرعت انتشار می‌باشد.

$T_{Process}$ از رابطه زیر بدست می‌آید:

$$T_{Proces} = \frac{b}{R}$$

$T_{Process}$، زمان پردازش موجود در مسیریاب (گره میانی) مربوط به کنترل خطای فریم، احیانا قطعه قطعه شدن بسته و مسیریابی بسته است.

که b برابر تعداد بیت لازم برای پردازش و R برابر نرخ انتقال می‌باشد.

T_{queue} از رابطه زیر بدست می‌آید:

$$T_{queue} = (N-۱) \times \left(\frac{L}{\min(R_۱, R_۲, R_۳)} \right)$$

T_{queue}، زمان تأخیر صف است.

که L برابر اندازه بسته، R برابر نرخ انتقال و N برابر تعداد بسته‌ها می‌باشد.

توجه: صف در جایی ایجاد می‌شود که پایین‌ترین نرخ انتقال را دارد یعنی $\min(R_۱, R_۲, R_۳)$ که در این حالت گلوگاه (bottleneck) در آن محل ایجاد شده است.

توجه: مطابق فلش موجود در نمودار صورت سوال حرکت بسته‌ها از سمت راست به چپ یعنی از سمت سرور به سمت کلاینت است.

همچنین داده‌های مسئله به صورت زیر است:

$L = ۱۰۰۰$ Byte

$T_{Prop۱} = ۱$ msec, $T_{Prop۲} = ۱۲$ msec, $T_{Prop۳} = ۲$ msec

$T_{Process۱} = ۰$, $T_{Process۲} = ۰$

$R_۱ = ۱۰۰$Mbps, $R_۲ = ۲۰$Mbps, $R_۳ = ۵۰$ Mbps

همانطور که گفتیم به طور کلی حداقل زمان لازم برای انتقال بسته‌ها مابین دو گره انتهایی از رابطه زیر محاسبه می‌گردد:

$$T_{\text{Total Delay}} = \left[T_{F_1}\right] + T_{Prop_1} + \left[T_{process_1} + T_{F_2}\right] + T_{Prop_2} + \left[T_{process_2} + T_{F_3}\right] + T_{Prop_3} + T_{queue}$$

که پس از جایگذاری اولیه رابطه زیر را خواهیم داشت:

$$T_{\text{Total Delay}} = \left[\frac{L}{R_1}\right] + 1 + \left[0 + \frac{L}{R_2}\right] + 12 + \left[0 + \frac{L}{R_3}\right] + 2 + (N-1) \times \left(\frac{L}{\min(R_1, R_2, R_3)}\right)$$

توجه: در صورت سؤال زمان پردازش موجود در مسیریاب ($T_{Process}$) داده نشده است، بنابراین در رابطه فوق، مقدار زمان پردازش را برابر صفر در نظر گرفتیم.

پس از جایگذاری نهایی رابطه زیر را خواهیم داشت:

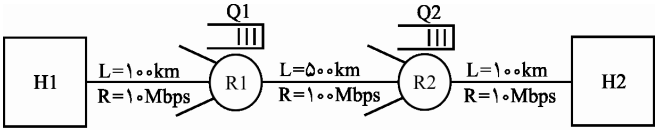
$$T_{\text{Total Delay}} = \left[\frac{1000 \times 8}{100 \times 10^6} \times 10^3\right] + 1 + \left[0 + \frac{1000 \times 8}{20 \times 10^6} \times 10^3\right] + 12 + \left[0 + \frac{1000 \times 8}{50 \times 10^6} \times 10^3\right] + 2 + (499) \times \left(\frac{1000 \times 8}{20 \times 10^6} \times 10^3\right)$$

در نهایت داریم:

$$T_{\text{Total Delay}} = [0/08] + 1 + [0 + 0/4] + 12 + [0 + 0/16] + 2 + (499) \times (0/4)$$

$$T_{\text{Total Delay}} = [0/08] + 1 + [0/4] + 12 + [0/16] + 2 + 199/6 = 15/64 + 199/6 = 215/24 \text{ msec}$$

در صورت سوال گفته شده است که یک جریان داده (صوت و ویدیو) با نرخ ۵ مگابیت بر ثانیه که در بسته‌های ۱۰۰۰۰ بیتی قرار دارند از برنامه‌ای در کامپیوتر H1 به برنامه‌ای در کامپیوتر H2 مطابق با شکل زیر، در حال ارسال است. همچنین فرض شده است که طول صف در بافرهای مسیریاب‌های R1 و R2، حداکثر ۴ بسته و حداقل صفر است. همچنین در ادامه صورت سوال خواسته شده است که با توجه به اطلاعات داده شده در شکل حداقل و حداکثر تاخیر یک بسته از زمان ارسال از کامپیوتر H1 و دریافت توسط کامپیوتر H2 به میلی ثانیه چقدر است؟



Propagation Speed $= 2 \times 10^8$ mps

همچنین داده‌های مسئله به صورت زیر است:

$L = 10000 \text{bit}$

$D_1 = 100 \text{km}, D_2 = 500 \text{km}, D_3 = 100 \text{km}$

$V = 2 \times 10^8$ mps

$T_{Process_1} = 0, T_{Process_2} = 0$

$R_1 = 10 \text{Mbps}, R_2 = 100 \text{Mbps}, R_3 = 10 \text{ Mbps}$

همانطور که گفتیم به طور کلی حداقل زمان لازم برای انتقال بسته‌ها مابین دو گره انتهایی از رابطه زیر محاسبه می‌گردد:

$$T_{\text{Total Delay}} = \left[T_{F1}\right] + T_{Prop1} + \left[T_{process1} + T_{F2}\right] + T_{Prop2} + \left[T_{process2} + T_{F3}\right] + T_{Prop3} + T_{queue}$$

همچنین به طور کلی حداقل زمان لازم برای انتقال «یک بسته» مابین دو گره انتهایی از رابطه زیر محاسبه می‌گرد:

$$T_{\text{Total Delay}} = \left[T_{F1}\right] + T_{Prop1} + \left[T_{process1} + T_{F2}\right] + T_{Prop2} + \left[T_{process2} + T_{F3}\right] + T_{Prop3} + 0$$

در رابطه فوق مقدار T_{queue} یعنی صف حاصل از «**تعدد**» بسته‌های ارسالی مابین یک فرستنده و گیرنده مورد نظر برابر صفر در نظر گرفته شده است. چون در صورت سوال حداقل و حداکثر تاخیر «**یک بسته**» از زمان ارسال از کامپیوتر H1 و دریافت توسط کامپیوتر H2 مورد پرسش قرار گرفته است. دقت کنید که در صورت سوال «**فقط**» حداقل و حداکثر تاخیر «**یک بسته**» از زمان ارسال از کامپیوتر H1 و دریافت توسط کامپیوتر H2 مورد پرسش قرار گرفته است. یعنی مقدار T_{queue} در رابطه فوق برای ارسال **فقط** «**یک بسته**» به صورت زیر محاسبه شده است.

$$T_{queue} = (1-1)\times\left(\frac{L}{\min(R_1, R_2, R_3)}\right) = (0)\times\left(\frac{L}{\min(R_1, R_2, R_3)}\right) = 0$$

کنترل جریان در محیط زیست node-to-node توسط لایه پیوند داده انجام می‌گردد، اما کنترل ازدحام در محیط زیست end-to-end توسط لایه انتقال انجام می‌گردد. چه کنترل جریان در محیط زیست node-to-node باشد و چه کنترل ازدحام در محیط زیست end-to-end باشد، هر دو جلوگیری می‌کنند از سرریزی اما اولی جلوگیری می‌کند از سرریزی در گره‌های انتهایی مربوط به محیط زیست node-to-node یعنی دو گره موجود در دو طرف یک تک یال از گراف شبکه و دومی جلوگیری می‌کند از سرریزی در گره‌های میانی مربوط به محیط زیست end-to-end یعنی گره‌های میانی موجود در گراف کل شبکه. با وجود کنترل جریان لایه پیوند داده اما همچنان به کنترل ازدحام لایه انتقال نیاز است. چون در کنترل جریان لایه پیوند داده فقط سرریزی گره‌های انتهایی مربوط به محیط زیست node-to-node یعنی دو گره موجود در دو طرف یک تک یال از گراف شبکه بررسی می‌شود اما در کنترل جریان لایه انتقال همچنان نیاز است تا سرریزی در گره‌های میانی مربوط به محیط زیست end-to-end یعنی گره‌های میانی موجود در گراف کل شبکه نیز بازهم بررسی گردد. در محیط زیست node-to-node حل مساله سرریزی فقط مابین دو گره انتهایی موجود در دو طرف یک تک یال از گراف شبکه است. یعنی طرفین مساله فقط دو گره انتهایی موجود در دو طرف یک تک یال از گراف شبکه است، که باید این دو مدارا کنند. اما در محیط زیست end-to-end حل مساله سرریزی مابین همه گره‌های انتهایی موجود در گراف کل شبکه است. یعنی طرفین مساله همه گره‌های انتهایی گراف کل شبکه است، که باید همه مدارا

کنند. حل مساله کنترل ازدحام بر عهده پروتکل TCP موجود در لایه انتقال به شیوه واکنشی و ضمنی یعنی بر اساس نشانه‌ها و نیامدن پیام ACK توسط الگوریتم TCP TAHOE یا TCP RENO است. در صورتی که TCP موجود در گره‌های انتهایی (فرستنده‌ها) به شکل بی‌رویه سگمنت‌های زیادی را ارسال کنند، باعث ازدحام در گره‌های میانی (مسیریاب‌ها) می‌شود. همچنین اگر تعداد سگمنت‌های ارسالی کم باشد از ظرفیت شبکه به درستی استفاده نمی‌شود. برای کنترل این مساله باید مکانیزمی وضع شود که مقدار مناسب سگمنت‌ها را مشخص کند، تا هم از ظرفیت شبکه به شکل بهینه استفاده شود و هم از سرریزی بافرهای گره‌های میانی (مسیریاب‌ها) جلوگیری کند. برای این امر مکانیزم دریچه ازدحام یا پنجره ازدحام یا congestion window یا cwnd ایجاد شده است. cwnd تعداد سگمنت‌های ارسالی توسط گره‌های انتهایی (فرستنده‌ها) را بر اساس بازخوردهای دریافتی از گره‌های میانی (مسیریاب‌ها) برعهده دارد. طوری که نه سرریزی در گره‌های میانی (مسیریاب‌ها) رخ دهد و نه از ظرفیت شبکه هدر رود.

همانطور که گفتیم در محیط زیست end-to-end حل مساله سرریزی مابین همه گره‌های انتهایی موجود در گراف کل شبکه است. یعنی طرفین مساله همه گره‌های انتهایی گراف کل شبکه است، که باید همه مدارا کنند. که این **مدارا کردن** و **مدارا نکردن** دو فرم زیر را ایجاد می‌کند:

فرم اول: اگر همه گره‌های انتهایی مرتبط با گره‌های میانی (مسیریاب‌ها) مدارا کنند و به دلیل ارسال نرمال سگمنت‌ها سبب ایجاد صف در هیچ یک از گره‌های میانی (مسیریاب‌ها) نشوند آنگاه **طول صف در بافرهای مسیریاب‌های R1 و R2 برابر صفر خواهد بود. در این حالت روابط زیر را برای مسیریاب‌های R1 و R2 خواهیم داشت:**

$$T^{min}_{queue\text{-}R1} = \left[\text{cardinality(queue-router1)} \times \frac{L}{R_2} \right] = \left[0 \times \frac{L}{R_2} \right] = 0 \text{ m sec}$$

$$T^{min}_{queue\text{-}R2} = \left[\text{cardinality(queue-router2)} \times \frac{L}{R_3} \right] = \left[0 \times \frac{L}{R_3} \right] = 0 \text{ m sec}$$

بنابراین «حداقل» زمان لازم برای انتقال فقط «یک بسته» مابین دو گره انتهایی H1 و H2 و «وابسته» به طول صف موجود در مسیریاب‌های **R1** و **R2 از رابطه زیر محاسبه می‌گرد:**

$$T^{min}_{\text{Total Delay(1)}} = \left[T_{F1} \right] + T_{Prop1} + \left[T^{min}_{queue\text{-}R1} + T_{process1} + T_{F2} \right] + T_{Prop2} + \left[T^{min}_{queue\text{-}R2} + T_{process2} + T_{F3} \right] + T_{Prop3}$$

که پس از جایگذاری اولیه رابطه زیر را خواهیم داشت:

$$T^{min}_{\text{Total Delay(1)}} = \left[\frac{L}{R_1} \right] + \frac{D_1}{V} + \left[0 + 0 + \frac{L}{R_2} \right] + \frac{D_2}{V} + \left[0 + 0 + \frac{L}{R_3} \right] + \frac{D_3}{V}$$

توجه: در صورت سؤال زمان پردازش موجود در مسیریاب ($T_{Process}$) داده نشده است، بنابراین در رابطه فوق، مقدار زمان پردازش را برابر صفر در نظر گرفتیم.

پس از جایگذاری نهایی رابطه زیر را خواهیم داشت:

$$T^{min}_{Total\ Delay(1)} = \left[\frac{10000}{10\times10^6}\times10^3\right] + \frac{100\times10^3}{2\times10^8}\times10^3 + \left[0+0+\frac{10000}{100\times10^6}\times10^3\right] + \frac{500\times10^3}{2\times10^8}\times10^3 + \left[0+0+\frac{10000}{10\times10^6}\times10^3\right] + \frac{100\times10^3}{2\times10^8}\times10^3$$

در نهایت داریم:

$$T^{min}_{Total\ Delay(1)} = [1]+0/5+[0+0+0/1]+2/5+[0+0+1]+0/5 = 5/6\ msec$$

تا به اینجا واضح است که گزینه اول یا سوم پاسخ سوال هستند.

فرم دوم: اگر همه گره‌های انتهایی مرتبط با گره‌های میانی (مسیریاب‌ها) مدارا نکنند و به دلیل ارسال غیرنرمال سگمنت‌ها سبب ایجاد صف درگره‌های میانی (مسیریاب‌ها) بشوند آنگاه **طول صف در بافرهای مسیریاب‌های R1 و R2 مطابق فرض مساله برابر چهار خواهد بود. در این حالت روابط زیر را برای مسیریاب‌های R1 و R2 خواهیم داشت:**

$$T^{max}_{queue\text{-}R1} = \left[cardinality(queue\text{-}router1)\times\frac{L}{R_2}\right] = \left[4\times\frac{10000}{100\times10^6}\times10^3\right] = 4\times0/1 = 0/4\ msec$$

$$T^{max}_{queue\text{-}R2} = \left[cardinality(queue\text{-}router2)\times\frac{L}{R_3}\right] = \left[4\times\frac{10000}{10\times10^6}\times10^3\right] = 4\times1 = 4\ msec$$

بنابراین «حداکثر» زمان لازم برای انتقال فقط «یک بسته» مابین دو گره انتهایی H1 و H2 و «وابسته» به طول صف موجود در مسیریاب‌های **R1** و **R2 از رابطه زیر محاسبه می‌گرد:**

$$T^{max}_{Total\ Delay(1)} = [T_{F1}] + T_{Prop1} + \left[T^{max}_{queue\text{-}R1} + T_{process1} + T_{F2}\right] + T_{Prop2} + \left[T^{max}_{queue\text{-}R2} + T_{process2} + T_{F3}\right] + T_{Prop3}$$

که پس از جایگذاری اولیه رابطه زیر را خواهیم داشت:

$$T^{max}_{Total\ Delay(1)} = \left[\frac{L}{R_1}\right] + \frac{D_1}{V} + \left[T^{max}_{queue\text{-}R1} + 0 + \frac{L}{R_2}\right] + \frac{D_2}{V} + \left[T^{max}_{queue\text{-}R2} + 0 + \frac{L}{R_3}\right] + \frac{D_3}{V}$$

توجه: در صورت سؤال زمان پردازش موجود در مسیریاب ($T_{Process}$) داده نشده است، بنابراین در رابطه فوق، مقدار زمان پردازش را برابر صفر در نظر گرفتیم.

پس از جایگذاری نهایی رابطه زیر را خواهیم داشت:

$$T^{max}_{Total\ Delay(1)} = \left[\frac{10000}{10\times10^6}\times10^3\right] + \frac{100\times10^3}{2\times10^8}\times10^3 + \left[0/4+0+\frac{10000}{100\times10^6}\times10^3\right] + \frac{500\times10^3}{2\times10^8}\times10^3 + \left[4+0+\frac{10000}{10\times10^6}\times10^3\right] + \frac{100\times10^3}{2\times10^8}\times10^3$$

در نهایت داریم:

$$T^{max}_{Total\ Delay(1)} = [1]+0/5+[0/4+0+0/1]+2/5+[4+0+1]+0/5 = 5/6+4/4 = 10\ msec$$

تا به اینجا واضح بود که گزینه اول یا سوم پاسخ سوال هستند. اما دیگر از اینجا به بعد با توجه به مقادیر گزینه اول یا سوم مشخص نیست که گزینه اول یا سوم پاسخ سوال است. همانطور که

گفتیم سازمان سنجش آموزش کشور، در کلید اولیه خود، گزینه سوم را به عنوان پاسخ اعلام کـرده بود. اما در کلید نهایی این سوال حذف گردید، که کار درستی بوده است.

۴- گزینه () صحیح است.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه خود، گزینه دوم را به عنوان پاسخ اعلام کـرده بود. اما در کلید نهایی این سوال حذف گردید، که کار درستی بوده است.

توجه: در شبکه‌های کامپیوتری چهار نوع تأخیر داریم:

تأخیر انتقال (T_F)، تأخیر انتشار (T_{Prop})، تأخیر صف (T_{queue})، تأخیر پردازش $(T_{process})$.

توجه: تأخیر صف‌بندی داخل گره‌ها، یک تأخیر متغیر است که به حجم ترافیک لحظه عبـور از آن گره بستگی دارد. به عبارت دیگر تاخیر صف در طول زمان نوسان دارد. پس تأخیری که از ابتدا به انتها ایجاد می‌شود، متغیر است و از قبل قابل پیش‌بینی نیست.

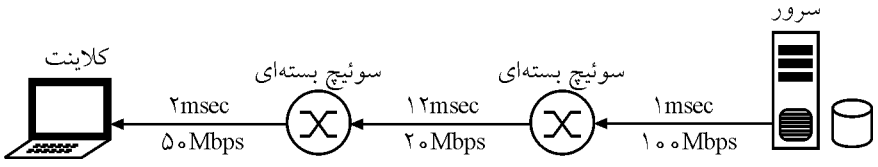
مثال: مثلاً دسترسی به سیستم آموزشی (پرتال)

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)

مثال- در شبکه‌ای با مسیر شکل زیر بین سرور و کلاینت وجود دارد، حداقل زمان لازم بـرای انتقـال پانصد بسته هزار بایتی بر حسب میلی‌ثانیه (msec) کدام است؟ (توجه: $1Mbps = 10^6 bps$)



۱) ۲۱۵ ۲) ۲۱۵/۱۲ ۳) ۲۱۵/۲۴ ۴) ۲۱۵/۶۴

پاسخ- گزینه (۳) صحیح است.

به طور کلی حداقل زمان لازم برای انتقال بسته‌ها مابین دو گره انتهایی از رابطه زیر محاسبه می‌گرد:

$$T_{Total\ Delay} = [T_{F1}] + T_{Prop1} + [T_{process1} + T_{F2}] + T_{Prop2} + [T_{process2} + T_{F3}] + T_{Prop3} + T_{queue}$$

T_F از رابطه زیر بدست می‌آید:

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

T_F، زمان انتقال بسته به داخل کانال انتقال است.

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

T_{Prop} از رابطه زیر بدست می‌آید:

$$T_{Prop} = \frac{D}{V}$$

T_{Prop}، زمان تأخیر انتشار است.
که D برابر طول کانال و V برابر سرعت انتشار می‌باشد.
$T_{Process}$ از رابطه زیر بدست می‌آید:

$$T_{Proces} = \frac{b}{R}$$

$T_{Process}$، زمان پردازش موجود در مسیریاب (گره میانی) مربوط به کنترل خطای فریم، احیانا قطعه قطعه شدن بسته و مسیریابی بسته است.
که b برابر تعداد بیت لازم برای پردازش و R برابر نرخ انتقال می‌باشد.
T_{queue} از رابطه زیر بدست می‌آید:

$$T_{queue} = (N-1)\times\left(\frac{L}{\min(R_1, R_2, R_3)}\right)$$

T_{queue}، زمان تأخیر صف است.
که L برابر اندازه بسته، R برابر نرخ انتقال و N برابر تعداد بسته‌ها می‌باشد.
توجه: صف در جایی ایجاد می‌شود که پایین‌ترین نرخ انتقال را دارد یعنی $\min(R_1, R_2, R_3)$ که در این حالت گلوگاه (bottleneck) در آن محل ایجاد شده است.
توجه: مطابق فلش موجود در نمودار صورت سوال حرکت بسته‌ها از سمت راست به چپ یعنی از سمت سرور به سمت کلاینت است.
همچنین داده‌های مسئله به صورت زیر است:

$L = 1000$ Byte
$T_{Prop1} = 1$ msec, $T_{Prop2} = 12$ msec, $T_{Prop3} = 2$ msec
$T_{Process1} = 0$, $T_{Process2} = 0$
$R_1 = 100$Mbps, $R_2 = 20$Mbps, $R_3 = 50$ Mbps

همانطور که گفتیم به طور کلی حداقل زمان لازم برای انتقال بسته‌ها مابین دو گره انتهایی از رابطه زیر محاسبه می‌گرد:

$$T_{Total\ Delay} = [T_{F1}] + T_{Prop1} + [T_{process1} + T_{F2}] + T_{Prop2} + [T_{process2} + T_{F3}] + T_{Prop3} + T_{queue}$$

که پس از جایگذاری اولیه رابطه زیر را خواهیم داشت:

$$T_{Total\ Delay} = \left[\frac{L}{R_1}\right] + 1 + \left[0 + \frac{L}{R_2}\right] + 12 + \left[0 + \frac{L}{R_3}\right] + 2 + (N-1)\times\left(\frac{L}{\min(R_1, R_2, R_3)}\right)$$

توجه: در صورت سؤال زمان پردازش موجود در مسیریاب ($T_{Process}$) داده نشده است، بنابراین در

رابطه فوق، مقدار زمان پردازش را برابر صفر در نظر گرفتیم.

پس از جایگذاری نهایی رابطه زیر را خواهیم داشت:

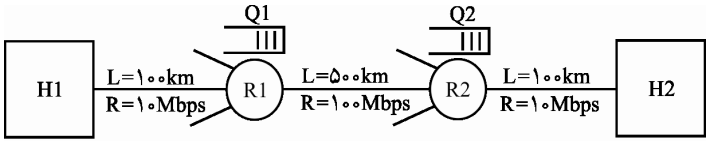
$$T_{\text{Total Delay}} = \left[\frac{۱۰۰۰\times۸}{۱۰۰\times۱۰^{۶}}\times۱۰^{۳}\right]+۱+\left[۰+\frac{۱۰۰۰\times۸}{۲۰\times۱۰^{۶}}\times۱۰^{۳}\right]+۱۲+\left[۰+\frac{۱۰۰۰\times۸}{۵۰\times۱۰^{۶}}\times۱۰^{۳}\right]+۲+(۴۹۹)\times\left(\frac{۱۰۰۰\times۸}{۲۰\times۱۰^{۶}}\times۱۰^{۳}\right)$$

در نهایت داریم:

$$T_{\text{Total Delay}} = [۰/۰۸]+۱+[۰+۰/۴]+۱۲+[۰+۰/۱۶]+۲+(۴۹۹)\times(۰/۴)$$

$$T_{\text{Total Delay}} = [۰/۰۸]+۱+[۰/۴]+۱۲+[۰/۱۶]+۲+۱۹۹/۶ = ۱۵/۶۴+۱۹۹/۶ = ۲۱۵/۲۴ \text{ msec}$$

در صورت سوال گفته شده است که یک جریان داده (صوت و ویدیو) با نرخ ۵ مگابیت بر ثانیه که در بسته‌های ۱۰۰۰۰ بیتی قرار دارند از برنامه‌ای در کامپیوتر H1 به برنامه‌ای در کامپیوتر H2 مطابق با شکل زیر، در حال ارسال است. همچنین فرض شده است که طول صف در بافرهای مسیریاب‌های R1 و R2، حداکثر ۴ بسته و حداقل صفر است. همچنین در ادامه صورت سوال خواسته شده است که با توجه به اطلاعات داده شده در شکل حداقل تاخیر زمانی به میلی‌ثانیه که برنامه در کامپیوتر H2 باید بافر خود را قبل از پخش داده مشاهده (محاسبه) کند و حداقل اندازه بافر مورد نیاز به کیلو بیت برای پخش بدون وقفه این جریان چقدر هستند؟



$$\text{Propagation Speed} = ۲\times۱۰^{۸} \text{ mps}$$

همچنین داده‌های مسئله به صورت زیر است:

$L = ۱۰۰۰۰ \text{bit}$

$D_۱ = ۱۰۰\text{km},\ D_۲ = ۵۰۰\text{km},\ D_۳ = ۱۰۰\text{km}$

$V = ۲\times۱۰^{۸} \text{ mps}$

$T_{\text{Process}۱} = ۰\ ,\ T_{\text{Process}۲} = ۰$

$R_{\text{Multimedia}} = ۵\text{Mbps},$

$R_۱ = ۱۰\text{Mbps},\ R_۲ = ۱۰۰\text{Mbps},\ R_۳ = ۱۰\text{ Mbps}$

توجه: در صورت سوال مفهوم **Jitter** مورد پرسش قرار گرفته است.

Jitter یک پدیده در انتقال صوت و تصویر می‌باشد. برای محاسبه Jitter ابتدا باید **حداقل** زمان و **حداکثر** زمان تاخیر یک بسته محاسبه شود. همانطور که گفتیم در شبکه‌های کامپیوتری چهار نوع تأخیر داریم:

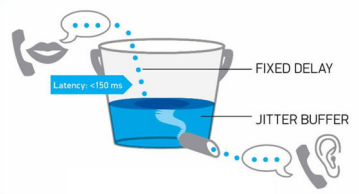
تأخیر انتقال (T_F)، تأخیر انتشار (T_{Prop})، تأخیر صف (T_{queue})، تأخیر پردازش $(T_{process})$.

توجه: تأخیر صف‌بندی داخل گره‌ها، یک تأخیر متغیر است که به حجم ترافیک لحظه عبور از آن گره بستگی دارد. به عبارت دیگر تاخیر صف در طول زمان نوسان دارد. پس تأخیری که از ابتدا به انتها ایجاد می‌شود، متغیر است و از قبل قابل پیش‌بینی نیست. بنابراین مدت زمان انتقال هر بسته ممکن است تغییر کند. این تغییر در تاخیر انتقال بسته‌ها به عنوان «Jitter» یا «تغییر در تاخیر بسته‌ها» نامیده می‌شود. پیوسته و گسسته شدن صوت و تصویر به Jitter وابسته است. اگر Jitter کم باشد صوت و تصویر به صورت پیوسته قابل مشاهده خواهد بود و این یعنی افزایش کیفیت در مشاهده صوت و تصویر. اما اگر Jitter زیاد باشد صوت و تصویر به صورت گسسته قابل مشاهده خواهد بود و این یعنی کاهش کیفیت در مشاهده صوت و تصویر.

Delay : مدت زمانی که یک بسته از یک نقطه شبکه به نقطه دیگه منتقل می‌شود.

Jitter : وقتی که Delay متغییر شد Jitter به وجود می‌آید. بافر گره انتهایی (گیرنده) باید ظرفیت لازم برای دریافت بسته‌ها به اندازه Jitter را داشته باشد.

بسته‌های ارسالی از سمت فرستنده، در سمت گیرنده باید مرتب و به ترتیب دریافت شوند اما این بسته‌ها در مسیر حرکت از سمت فرستنده به گیرنده به ترتیب حرکت نمی‌کنند و ممکن است از مسیرهای متفاوتی به سمت گیرنده روانه گردند و به تبع ممکن است تاخیرهای متفاوتی در مسیر رسیدن به گیرنده تجربه کنند. برای مقابله با پدیده Jitter از مکانیزمی به نام Jitter Buffer استفاده می‌گردد، Jitter Buffer یک فرصت‌ساز برای دریافت بسته‌ای است که باید می‌رسیده است ولی به دلیل تاخیر هنوز نیامده است، در حالی که بسته‌های جلویی آن در سمت گیرنده دریافت شده‌اند. در واقع Jitter Buffer باید آنقدر دلگنده و جادار باشد و انقدر باید بسته‌های غیرمرتبط که شماره ترتیب مد نظر گیرنده نیست را بی‌مورد دریافت کند تا بالاخره بسته مورد نظر دریافتی گیرنده که دارای شماره ترتیب مدنظر گیرنده است از راه برسد. شکل زیر گویای مطلب است:



زمانیکه بسته‌های اطلاعاتی از مبدا به مقصد می‌رسند ابتدا Jitter Buffer یک برآورد از حداقل و حداکثر میزان تاخیر بسته‌ها محاسبه می‌کند و از تفاضل حاصل از حداکثر و حداقل تاخیر مقدار Jitter را محاسبه می‌کند و میزان Jitter Buffer را مشخص می‌کند. البته تا زمانی مکانیزم Jitter Buffer قادر به مقابله با پدیده Jitter است که مقدار Jitter از ۱۵۰ میلی‌ثانیه تجاوز نکند، چون این تاخیر به این میزان برای گیرنده قابل پذیرش نیست.

همانطور که گفتیم به طور کلی حداقل زمان لازم برای انتقال بسته‌ها مابین دو گره انتهایی از رابطه زیر محاسبه می‌گرد:

$$T_{\text{Total Delay}} = \left[T_{F۱}\right] + T_{Prop۱} + \left[T_{process۱} + T_{F۲}\right] + T_{Prop۲} + \left[T_{process۲} + T_{F۳}\right] + T_{Prop۳} + T_{queue}$$

همچنین به طور کلی حداقل زمان لازم برای انتقال «یک بسته» مابین دو گره انتهایی از رابطه زیر محاسبه می‌گرد:

$$T_{\text{Total Delay}} = \left[T_{F۱}\right] + T_{Prop۱} + \left[T_{process۱} + T_{F۲}\right] + T_{Prop۲} + \left[T_{process۲} + T_{F۳}\right] + T_{Prop۳} + ۰$$

در رابطه فوق مقدار T_{queue} یعنی صف حاصل از «**تعدد**» بسته‌های ارسالی مابین یک فرستنده و گیرنده مورد نظر برابر صفر در نظر گرفته شده است. چون در صورت سوال حداقل و حداکثر تاخیر «**یک بسته**» از زمان ارسال از کامپیوتر H1 و دریافت توسط کامپیوتر H2 مورد پرسش قرار گرفته است. دقت کنید که در صورت سوال «**فقط**» حداقل و حداکثر تاخیر «**یک بسته**» از زمان ارسال از کامپیوتر H1 و دریافت توسط کامپیوتر H2 مورد پرسش قرار گرفته است. یعنی مقدار T_{queue} در رابطه فوق برای ارسال **فقط** «**یک بسته**» به صورت زیر محاسبه شده است.

$$T_{queue} = (۱-۱)\times\left(\frac{L}{\min(R_۱, R_۲, R_۳)}\right) = (۰)\times\left(\frac{L}{\min(R_۱, R_۲, R_۳)}\right) = ۰$$

کنترل جریان در محیط زیست node-to-node توسط لایه پیوند داده انجام می‌گردد، اما کنترل ازدحام در محیط زیست end-to-end توسط لایه انتقال انجام می‌گردد. چه کنترل جریان در محیط زیست node-to-node باشد و چه کنترل ازدحام در محیط زیست end-to-end باشد، هر دو جلوگیری می‌کنند از سرریزی اما اولی جلوگیری می‌کند از سرریزی در گره‌های انتهایی مربوط به محیط زیست node-to-node یعنی دو گره موجود در دو طرف یک تک یال از گراف شبکه و دومی جلوگیری می‌کند از سرریزی در گره‌های میانی مربوط به محیط زیست end-to-end یعنی گره‌های میانی موجود در گراف کل شبکه. با وجود کنترل جریان لایه پیوند داده اما همچنان به کنترل ازدحام لایه انتقال نیاز است. چون در کنترل جریان لایه پیوند داده فقط سرریزی گره‌های انتهایی مربوط به محیط زیست node-to-node یعنی دو گره موجود در دو طرف یک تک یال از گراف شبکه بررسی می‌شود اما در کنترل جریان لایه انتقال همچنان نیاز است تا سرریزی در گره‌های میانی مربوط به محیط زیست end-to-end یعنی گره‌های میانی موجود در گراف کل شبکه نیز بازهم بررسی گردد. در محیط زیست node-to-node حل مساله سرریزی فقط مابین دو گره انتهایی موجود در دو طرف یک تک یال از گراف شبکه است. یعنی طرفین مساله فقط دو گره انتهایی موجود در دو طرف یک تک یال از گراف شبکه است، که باید این دو مدارا کنند. اما در محیط زیست end-to-end حل مساله سرریزی مابین همه گره‌های انتهایی موجود در گراف کل شبکه است. یعنی طرفین مساله همه گره‌های انتهایی گراف کل شبکه است، که باید همه مدارا کنند. حل مساله کنترل ازدحام بر عهده پروتکل TCP موجود در لایه انتقال به شیوه واکنشی و ضمنی یعنی بر اساس نشانه‌ها و نیامدن پیام ACK توسط الگوریتم TCP TAHOE یا TCP

RENO است. در صورتی که TCP موجود در گره‌های انتهایی (فرستنده‌ها) به شکل بی‌رویه سگمنت‌های زیادی را ارسال کنند، باعث ازدحام در گره‌های میانی(مسیریاب‌ها) می‌شود. همچنین اگر تعداد سگمنت‌های ارسالی کم باشد از ظرفیت شبکه به درستی استفاده نمی‌شود. برای کنترل این مساله باید مکانیزمی وضع شود که مقدار مناسب سگمنت‌ها را مشخص کند، تا هم از ظرفیت شبکه به شکل بهینه استفاده شود و هم از سرریزی بافرهای گره‌های میانی(مسیریاب‌ها) جلوگیری کند. برای این امر مکانیزم دریچه ازدحام یا پنجره ازدحام یا congestion window یا cwnd ایجاد شده است. cwnd تعداد سگمنت‌های ارسالی توسط گره‌های انتهایی (فرستنده‌ها) را بر اساس بازخوردهای دریافتی از گره‌های میانی (مسیریاب‌ها) برعهده دارد. طوری که نه سرریزی در گره‌های میانی (مسیریاب‌ها) رخ دهد و نه از ظرفیت شبکه هدر رود.

همانطور که گفتیم در محیط زیست end-to-end حل مساله سرریزی مابین همه گره‌های انتهایی موجود در گراف کل شبکه است. یعنی طرفین مساله همه گره‌های انتهایی گراف کل شبکه است، که باید همه مدارا کنند. که این **مدارا کردن** و **مدارا نکردن** دو فرم زیر را ایجاد می‌کند:

فرم اول: اگر همه گره‌های انتهایی مرتبط با گره‌های میانی (مسیریاب‌ها) مدارا کنند و به دلیل ارسال نرمال سگمنت‌ها سبب ایجاد صف در هیچ یک از گره‌های میانی (مسیریاب‌ها) نشوند آنگاه **طول صف در بافرهای مسیریاب‌های R1 و R2 برابر صفر خواهد بود. در این حالت روابط زیر را برای مسیریاب‌های R1 و R2 خواهیم داشت:**

$$T^{min}_{queue\text{-}R1} = \left[\text{cardinality(queue-router۱)} \times \frac{L}{R_۲} \right] = \left[۰ \times \frac{L}{R_۲} \right] = ۰ \text{ m sec}$$

$$T^{min}_{queue\text{-}R۲} = \left[\text{cardinality(queue-router۲)} \times \frac{L}{R_۳} \right] = \left[۰ \times \frac{L}{R_۳} \right] = ۰ \text{ m sec}$$

بنابراین «حداقل» زمان لازم برای انتقال فقط «یک بسته» مابین دو گره انتهایی H1 و H2 و «وابسته» به طول صف موجود در مسیریاب‌های **R1** و **R2 از رابطه زیر محاسبه می‌گرد:**

$$T^{min}_{\text{Total Delay}(۱)} = \left[T_{F۱}\right] + T_{Prop۱} + \left[T^{min}_{queue\text{-}R۱} + T_{process۱} + T_{F۲}\right] + T_{Prop۲} + \left[T^{min}_{queue\text{-}R۲} + T_{process۲} + T_{F۳}\right] + T_{Prop۳}$$

که پس از جایگذاری اولیه رابطه زیر را خواهیم داشت:

$$T^{min}_{\text{Total Delay}(۱)} = \left[\frac{L}{R_۱}\right] + \frac{D_۱}{V} + \left[۰ + ۰ + \frac{L}{R_۲}\right] + \frac{D_۲}{V} + \left[۰ + ۰ + \frac{L}{R_۳}\right] + \frac{D_۳}{V}$$

توجه: در صورت سؤال زمان پردازش موجود در مسیریاب ($T_{Process}$)داده نشده است، بنابراین در رابطه فوق، مقدار زمان پردازش را برابر صفر در نظر گرفتیم.

پس از جایگذاری نهایی رابطه زیر را خواهیم داشت:

$$T^{min}_{Total\ Delay(1)} = \left[\frac{10000}{10\times10^6}\times10^3\right] + \frac{100\times10^3}{2\times10^8}\times10^3 + \left[0+0+\frac{10000}{100\times10^6}\times10^3\right] + \frac{500\times10^3}{2\times10^8}\times10^3 + \left[0+0+\frac{10000}{10\times10^6}\times10^3\right] + \frac{100\times10^3}{2\times10^8}\times10^3$$

در نهایت داریم:

$$T^{min}_{Total\ Delay(1)} = [1]+0/5+[0+0+0/1]+2/5+[0+0+1]+0/5 = 5/6\ msec$$

فرم دوم: اگر همه گره‌های انتهایی مرتبط با گره‌های میانی (مسیریاب‌ها) مدارا نکنند و به دلیل ارسال غیرنرمال سگمنت‌ها سبب ایجاد صف درگره‌های میانی (مسیریاب‌ها) بشوند آنگاه **طول صف در بافرهای مسیریاب‌های R1 و R2 مطابق فرض مساله برابر چهار خواهد بود. در این حالت روابط زیر را برای مسیریاب‌های R1 و R2 خواهیم داشت:**

$$T^{max}_{queue\text{-}R1} = \left[cardinality(queue\text{-}router1)\times\frac{L}{R_2}\right] = \left[4\times\frac{10000}{100\times10^6}\times10^3\right] = 4\times0/1 = 0/4\ msec$$

$$T^{max}_{queue\text{-}R2} = \left[cardinality(queue\text{-}router2)\times\frac{L}{R_2}\right] = \left[4\times\frac{10000}{10\times10^6}\times10^3\right] = 4\times1 = 4\ msec$$

بنابراین «حداکثر» زمان لازم برای انتقال فقط «یک بسته» مابین دو گره انتهایی H1 و H2 و «وابسته» به طول صف موجود در مسیریاب‌های **R1** و **R2 از رابطه زیر محاسبه می‌گرد:**

$$T^{max}_{Total\ Delay(1)} = [T_{F1}] + T_{Prop1} + \left[T^{max}_{queue\text{-}R1} + T_{process1} + T_{F2}\right] + T_{Prop2} + \left[T^{max}_{queue\text{-}R2} + T_{process2} + T_{F2}\right] + T_{Prop3}$$

که پس از جایگذاری اولیه رابطه زیر را خواهیم داشت:

$$T^{max}_{Total\ Delay(1)} = \left[\frac{L}{R_1}\right] + \frac{D_1}{V} + \left[T^{max}_{queue\text{-}R1} + 0 + \frac{L}{R_2}\right] + \frac{D_2}{V} + \left[T^{max}_{queue\text{-}R2} + 0 + \frac{L}{R_3}\right] + \frac{D_3}{V}$$

توجه: در صورت سؤال زمان پردازش موجود در مسیریاب ($T_{Process}$) داده نشده است، بنابراین در رابطه فوق، مقدار زمان پردازش را برابر صفر در نظر گرفتیم.

پس از جایگذاری نهایی رابطه زیر را خواهیم داشت:

$$T^{max}_{Total\ Delay(1)} = \left[\frac{10000}{10\times10^6}\times10^3\right] + \frac{100\times10^3}{2\times10^8}\times10^3 + \left[0/4+0+\frac{10000}{100\times10^6}\times10^3\right] + \frac{500\times10^3}{2\times10^8}\times10^3 + \left[4+0+\frac{10000}{10\times10^6}\times10^3\right] + \frac{100\times10^3}{2\times10^8}\times10^3$$

در نهایت داریم:

$$T^{max}_{Total\ Delay(1)} = [1]+0/5+[0/4+0+0/1]+2/5+[4+0+1]+0/5 = 5/6+4/4 = 10\ msec$$

حال در ادامه برای محاسبه **Jitter** رابطه زیر را خواهیم داشت:

$$T_{Jitter} = T^{max}_{Total\ Delay(1)} - T^{min}_{Total\ Delay(1)} = 10 - 5/6 = 4/4\ msec$$

برای محاسبه **حداقل** اندازه **Jitter Buffer** رابطه زیر را خواهیم داشت:

$$T_{Jitter} = \frac{L_{JITTER\ BUFFER}}{R_{Multimedia}}$$

همچنین بر اساس رابطه فوق داریم:

$$L_{JITTER\ BUFFER} = T_{Jitter} \times R_{Multimedia} \times = ۴/۴ \times ۱۰^{-۳} \times ۵ \times ۱۰^{۳} = ۲۲\ kb$$

اما متأسفانه، این پاسخ در گزینه‌ها موجود نیست. همانطور که گفتیم سازمان سنجش آموزش کشور، در کلید اولیه خود، گزینه دوم را به عنوان پاسخ اعلام کرده بود. اما در کلید نهایی این سوال حذف گردید، که کار درستی بوده است.

۵- گزینه () صحیح است.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه خود، گزینه دوم را به عنوان پاسخ اعلام کرده بود. اما در کلید نهایی این سوال حذف گردید، که کار درستی بوده 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. 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 (3-way 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.

توجه: وقتی Client، ACK را از Server گرفت، Connection سمت Client به Server باز می‌شود، پس Client می‌تواند داده و درخواست بفرستد. Client این اختیار را دارد که همراه ACK، داده و درخواست هم بفرستد.

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

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

توجه: TCP، Connectionهایش دو طرفه است، یعنی هم از سمت Client به سمت Server یک 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: مثلاً شما وقتی دوستتان را ببینید برای باز کردن سر صحبت یک سری تعارفات اولیه 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. البته اگر request اش را همراه ACK بدهد، که معمولا به این صورت است. به این زمان رفت و برگشت اصطلاحا RTT یا Round Trip Time 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 یک پیغام می‌دهد تا جواب آن را بگیرد، شامل تمام تأخیرهای شبکه است، تأخیر انتقال (T_F)، تأخیر انتشار (T_{Prop})، تأخیر صف (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 برای دیدن جزئیات آن ها باید RFCشان را بگیریم و مطالعه کنیم یا اگر بخواهیم آن‌ها را پیاده‌سازی کنیم باید RFC آنها را تهیه کنیم. RFC مانند کتاب قانون است، قوانینی دارد که می‌گوید:

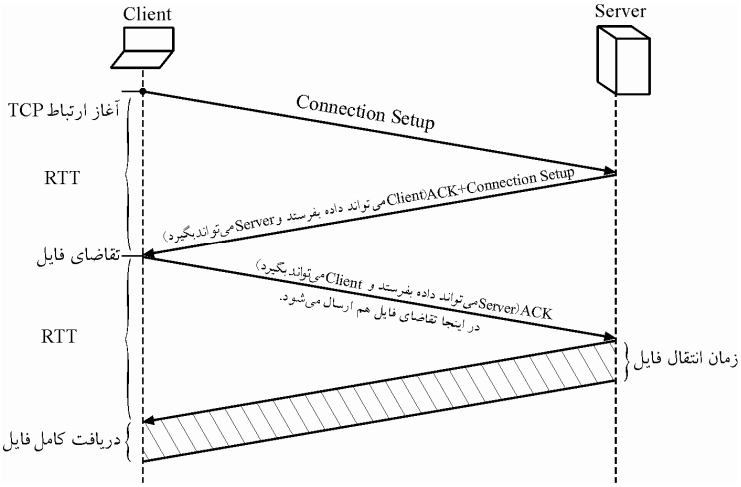
۱- اول این کار را انجام بده

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 بـر روی یـک 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 اسـت. فـرض کنیـد بـرای دریافـت آدرس IP از طریق سرویس DNS، n سرور DNS ملاقات می‌شوند و تاخیر زمان رفت و برگشـت معادل RTT1 تا RTTn باشد. بنابراین بدون در نظر گرفتن زمان مربوط بـه درخواسـت و دریافـت 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 نیـز بـرای درخواست و دریافت objectها به طور مستقل Connection باز و بسته می‌شود.

حالت Non-persistent http خود به سه روش ترتیبی، موازی نامحـدود و مـوازی محـدود وجـود دارد، که روابط آن به صورت زیر است:

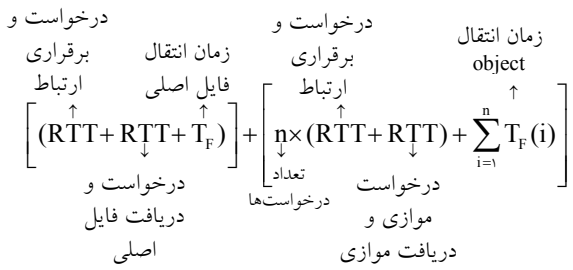
روش ناپایدار ترتیبی:

$$\left[\overset{\text{درخواست و برقراری ارتباط}}{\overset{\uparrow}{RTT}} + \underset{\text{درخواست و دریافت فایل اصلی}}{\underset{\downarrow}{RTT}} + \overset{\text{زمان انتقال فایل اصلی}}{\overset{\uparrow}{T_F}}\right] + \left[\underset{\text{تعداد درخواست‌ها}}{\underset{\downarrow}{n}} \times \left(\overset{\text{درخواست و برقراری ارتباط}}{\overset{\uparrow}{RTT}} + \underset{\text{درخواست و دریافت objectها}}{\underset{\downarrow}{RTT}}\right) + \overset{\text{زمان انتقال object}}{\overset{\uparrow}{\sum_{i=1}^{n} T_F(i)}}\right]$$

روش ناپایدار موازی نامحدود:

$$\left[\overset{\text{درخواست و برقراری ارتباط}}{\overset{\uparrow}{RTT}} + \underset{\text{درخواست و دریافت فایل اصلی}}{\underset{\downarrow}{RTT}} + \overset{\text{زمان انتقال فایل اصلی}}{\overset{\uparrow}{T_F}}\right] + \left[\underset{\text{یک}}{\underset{\downarrow}{n}} \times \left(\overset{\text{درخواست و برقراری ارتباط}}{\overset{\uparrow}{RTT}} + \underset{\text{درخواست موازی و دریافت موازی}}{\underset{\downarrow}{RTT}}\right) + \overset{\text{زمان انتقال object}}{\overset{\uparrow}{\sum_{i=1}^{n} T_F(i)}}\right]$$

روش ناپایدار موازی محدود:

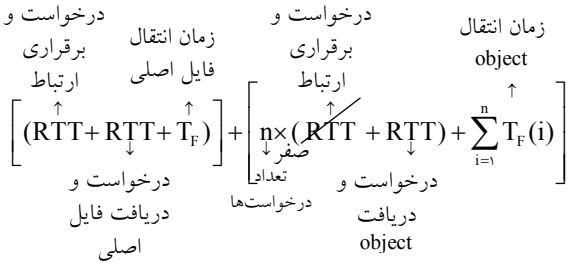


persistent http پایدار (مصر یا مداوم)

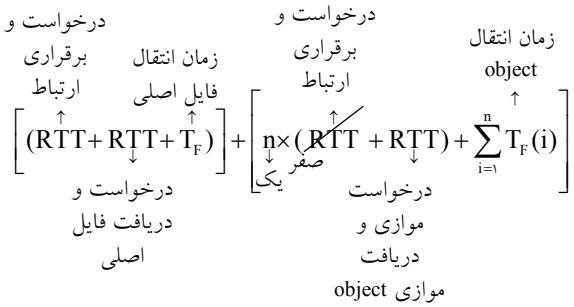
در حالت persistent http یک TCP connection مابین فرستنده و گیرنده برای درخواست و دریافت فایل پایه HTML ایجاد می‌گردد و در انتها Connection باز می‌ماند. در ادامه نیز برای درخواست و دریافت object ها همان TCP connection اولیه مورد استفاده قرار می‌گیرد.

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

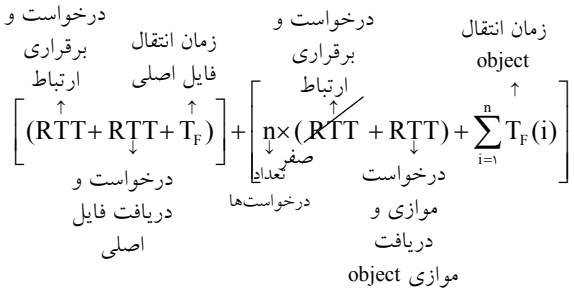
روش پایدار ترتیبی:



پایدار موازی نامحدود:



روش پایدار موازی محدود:



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

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.

مثال- 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 روش ترتیبی و HTTP Persistent روش موازی نامحدود از زمانی که شخص روی لینک کلیک می کند 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

پاسخ- گزینه (۳) صحیح است.

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:

$$\left[\left(\frac{RTT}{۱۰۰} + \frac{RTT}{۱۰۰} + \frac{T_F}{\text{صفر}}\right)\right] + \left[۸ \times \left(\frac{RTT}{۱۰۰} + \frac{RTT}{۱۰۰}\right) + \sum_{i=1}^{۸} \frac{T_F(i)}{\text{صفر}}\right]$$

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

روش ناپایدار و موازی نامحدود:

$$\left[\left(\frac{RTT}{۱۰۰} + \frac{RTT}{۱۰۰} + \frac{T_F}{\text{صفر}}\right)\right] + \left[۱ \times \left(\frac{RTT}{۱۰۰} + \frac{RTT}{۱۰۰}\right) + \sum_{i=1}^{۸} \frac{T_F(i)}{\text{صفر}}\right]$$

$\rightarrow ۲۰۰ + ۲۰۰ = ۴۰۰ \text{ ms}$

روش پایدار و ترتیبی:

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

$$\rightarrow ۲۰۰+۸۰۰=۱۰۰۰\ \text{ms}$$

روش پایدار و موازی نامحدود:

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

$$\rightarrow ۲۰۰+۱۰۰=۳۰۰\ \text{ms}$$

مثال- 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 http 1.0 (non-persistan) استفاده می‌کند. وب سرور حداکثر اجازه پنج ارتباط TCP همزمان به یک کلاینت را می‌دهد. چنانچه گذردهی شبکه بین کامپیوتر کاربر و وب سرور $۱۰^۶$ bps 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) ۱/۰ ۲) ۱/۲ ۳) ۱/۴ ۴) ۱/۵

پاسخ- گزینه (۱) صحیح است.

داده‌های مسئله به صورت زیر است:

$$L_{\text{Base HTML}}=۵۰۰۰\ \text{Byte}\ ,\ L_{\text{Object}}=۵۰۰۰\ \text{Byte}$$

$$R_{\text{TOTAL CHANNEL}}=۱۰^۶\ \text{bps},\ RTT=۰/۱\ \text{s}$$

$$\text{Cardinality(Object)}=۹,\ T_{\text{DNS LOOK UP}}=۰$$

رابطه روش ناپایدار موازی محدود به صورت زیر است:

$$\left[\underbrace{(\overbrace{RTT}^{\text{درخواست و برقراری ارتباط}}+\underbrace{RTT}_{\text{درخواست و دریافت فایل اصلی}}+\overbrace{T_F}^{\text{زمان انتقال فایل اصلی}})}\right]+\left[\underbrace{n}_{\text{تعداد درخواست‌ها}}\times\underbrace{(\overbrace{RTT}^{\text{درخواست و برقراری ارتباط}}+\underbrace{RTT}_{\text{درخواست موازی و دریافت موازی}})}+\sum_{i=۱}^{n}\overbrace{T_F(i)}^{\text{زمان انتقال object}}\right]$$

$T_{F(\text{Base HTML})}$ از رابطه زیر بدست می‌آید:

$$T_{F(\text{Base HTML})}=\frac{L_{\text{Base HTML}}}{R_{\text{TOTAL CHANNEL}}}$$

$T_{F(\text{Base HTML})}$، زمان انتقال فایل پایه html به داخل کانال انتقال است.

که $L_{Base\ HTML}$ برابر اندازه فایل پایه html و $R_{TOTAL\ CHANNEL}$ برابر نرخ انتقال کل کانال می‌باشد.
$T_{F(Object)}$ از رابطه زیر بدست می‌آید:

$$T_{F(Object)} = \frac{L_{Object}}{R_{TOTAL\ CHANNEL}}$$

$T_{F(Object)}$، زمان انتقال Object به داخل کانال انتقال است.
که L_{Object} برابر اندازه Object و $R_{TOTAL\ CHANNEL}$ برابر نرخ انتقال کل کانال می‌باشد.
با توجه به شرایط ذکر شده در صورت سؤال، مطابق آنچه گفتیم ابتدا بایستی فایل پایه HTML را دریافت کرد و سپس ۹ فایل دیگر را دریافت کرد و حالا با توجه به نوع ارتباط که **ناپایدار موازی محدود** است و تنها می‌توان **۵ کانکشن موازی** داشت بایستی اینگونه عمل کنیم:
ابتدا یک RTT صرف درخواست و برقراری ارتباط می‌شود، سپس RTT دیگر صرف درخواست و دریافت فایل پایه HTML می‌شود و پس از آن ارتباط قطع می‌شود، بعد از دریافت فایل پایه HTML ، با ۵ کانکشن موازی درخواست و 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، با ۴ کانکشن موازی دیگر درخواست و برقراری ارتباط را در یک RTT می‌دهیم و در RTT بعدی ۴ فایل باقی‌مانده را دریافت می‌کنیم. بنابراین مقدار n در رابطه فوق باعنوان تعداد درخواست‌ها برابر ۲ خواهد بود، زیرا دو کانکشن یکی ۵ تایی و دیگری ۴تایی ایجاد کردیم.

که پس از جایگذاری اولیه رابطه زیر را خواهیم داشت:

$$\left[\left(\frac{RTT}{۰/۱}+\frac{RTT}{۰/۱}+\frac{۵۰۰۰\times۸}{۱۰^۶}\right)\right]+\left[۲\times\left(\frac{RTT}{۰/۱}+\frac{RTT}{۰/۱}\right)+۹\times\frac{۵۰۰۰\times۸}{۱۰^۶}\right]$$

پس از جایگذاری نهایی رابطه زیر را خواهیم داشت:

$$[(۰/۱+۰/۱+۰/۰۴)]+[۲\times(۰/۱+۰/۱)+۹\times۰/۰۴]=[(۰/۲۴)]+[۰/۴+۰/۳۶]$$

$$\rightarrow [(۰/۲۴)]+[۰/۷۶]=۱\ s$$

در صورت سوال گفته شده است که فرض کنید یک برنامه سرویس‌گیرنده (Client) بعد از پیداکردن آدرس IP کامپیوتر سرویس‌دهنده (Server) می‌خواهد یک صفحه وب که اندازه فایل اصلی آن ۲۰۰ کیلوبیت و اندازه هر یک از ۳ تصویر قرار گرفته در آن ۳۰۰ کیلوبیت است را از طریق پروتکل HTTP غیرمداوم (Non-Persistent HTTP) که مجاز به ایجاد اتصال موازی نیز است، دریافت کند. همچنین گفته شده است که اگر زمان رفت و برگشت (RTT) ۲۰۰ میلی ثانیه، نرخ ارسال هر اتصال ۱۰ مگابیت بر ثانیه و اندازه پیام‌های GET ناچیز باشد، تاخیر دریافت کامل این صفحه وب به میلی‌ثانیه چقدر است؟

داده‌های مسئله به صورت زیر است:

$L_{Base\ HTML} = ۲۰۰\ kbit\ ,\ L_{Object} = ۳۰۰\ kbit$

$R_{TCP\ CHANNEL} = ۱۰\ mbps,\ \ RTT = ۲۰۰\ m\,sec$

$Cardinality(Object) = ۳,\ T_{DNS\ LOOK\ UP} = ۰$

توجه: در صورت سوال محدودیت توازی مطرح نشده است، پس توازی نامحدود را در نظر می‌گیریم.

رابطه روش ناپایدار موازی نامحدود به صورت زیر است:

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

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

اما رابطه روش ناپایدار موازی نامحدود با در نظر گرفتن نرخ انتقال هر اتصال از اتصالات موازی به صورت زیر است:

$$\left[(\overset{\uparrow}{RTT}+\underset{\downarrow}{RTT}+\overset{\uparrow}{T_F})\right]+\left[\underset{\downarrow}{n}\times(\overset{\uparrow}{RTT}+\underset{\downarrow}{RTT})+Max\left(\frac{\overset{\uparrow}{L_{Object1}}}{R_{TCP\ CHANNEL}},\frac{L_{Object2}}{R_{TCP\ CHANNEL}},\frac{L_{Object3}}{R_{TCP\ CHANNEL}}\right)\right]$$

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

توجه: دقت کنید که در صورت سوال گفته شده است که نرخ ارسال هر اتصال ۱۰ مگابیت بر ثانیه است که این به معنی نرخ انتقال هر اتصال از اتصالات موازی است یعنی $R_{TCP\ CHANNEL}$ و نه نرخ انتقال کل کانال یعنی $R_{TOTAL\ CHANNEL}$.

$T_{F(Base\ HTML)}$ از رابطه زیر بدست می‌آید:

$$T_{F(Base\ HTML)} = \frac{L_{Base\ HTML}}{R_{TCP\ CHANNEL}}$$

$T_{F(Base\ HTML)}$، زمان انتقال فایل پایه html به داخل کانال انتقال است.

که $L_{Base\ HTML}$ برابر اندازه فایل پایه html و $R_{TCP\ CHANNEL}$ برابر نرخ انتقال هر اتصال از اتصالات موازی می‌باشد.

$T_{F(Object)}$ از رابطه زیر بدست می‌آید:

$$T_{F(Object)} = \frac{L_{Object}}{R_{TCP\ CHANNEL}}$$

$T_{F(Object)}$، زمان انتقال Object به داخل کانال انتقال است.

که L_{Object} برابر اندازه Object و $R_{TCP\ CHANNEL}$ برابر نرخ انتقال هر اتصال از اتصالات موازی می‌باشد.

با توجه به شرایط ذکر شده در صورت سؤال، مطابق آنچه گفتیم ابتدا بایستی فایل پایه HTML را دریافت کرد و سپس ۳ فایل object دیگر را دریافت کرد و حالا با توجه به نوع ارتباط که **ناپایدار موازی نامحدود** است بایستی اینگونه عمل کنیم:

ابتدا یک RTT صرف درخواست و برقراری ارتباط می‌شود، سپس یک RTT دیگر صرف درخواست و دریافت فایل پایه HTML می‌شود و پس از آن ارتباط قطع می‌شود، بعد از دریافت فایل پایه HTML ، با **کانکشن موازی نامحدود** درخواست و برقراری ارتباط را در یک RTT می‌دهیم یعنی فقط ۳ کانکشن موازی مورد نیاز و در RTT بعدی ۳ فایل object را دریافت می‌کنیم. بنابراین مقدار n در رابطه فوق باعنوان تعداد درخواست‌ها برابر ۱ خواهد بود، زیرا فقط یک کانکشن برای درخواست و دریافت ۳ فایل object ایجاد کردیم.

که پس از جایگذاری اولیه رابطه زیر را خواهیم داشت:

$$\left[(۲۰۰+۲۰۰+\frac{۲۰۰\times۱۰^{۳}}{۱۰\times۱۰^{۶}}\times۱۰^{۳})\right]+\left[۱\times(۲۰۰+۲۰۰)+\text{Max}\left(\frac{۳۰۰\times۱۰^{۳}}{۱۰\times۱۰^{۶}}\times۱۰^{۳},\frac{۳۰۰\times۱۰^{۳}}{۱۰\times۱۰^{۶}}\times۱۰^{۳},\frac{۳۰۰\times۱۰^{۳}}{۱۰\times۱۰^{۶}}\times۱۰^{۳}\right)\right]$$

پس از جایگذاری نهایی رابطه زیر را خواهیم داشت:

$$\left[(۲۰۰+۲۰۰+۲۰)\right]+\left[۱\times(۲۰۰+۲۰۰)+\text{Max}\left(۳۰,۳۰,۳۰\right)\right]$$

$$\rightarrow \left[(۲۰۰+۲۰۰+۲۰)\right]+\left[۱\times(۲۰۰+۲۰۰)+۳۰\right]$$

$$\rightarrow \left[(۴۲۰)\right]+\left[۴۳۰\right]=۸۵۰\ \text{mses}$$

اما متأسفانه، این پاسخ در گزینه‌ها موجود نیست و با توجه به گزینه درست اعلام شده در کلید اولیه سازمان سنجش آموزش کشور یعنی گزینه دوم، مشخص می‌شود که طراح این سؤال، دچار خطای محاسباتی شده است. با کمی دقت شاید بتوان به خطای طراح محترم پی برد، به نظر می‌رسد طراح محترم در محاسبات خود دچار خطا شده است، و مقدار RTT را تاخیر انتشار یک طرفه درنظر گرفته است و به تبع در محاسبات خود مقدار ۴۰۰ را به جای ۲۰۰ قرار داده است که در این حالت می‌توان به گزینه دوم که نظر سازمان سنجش آموزش کشور در کلید اولیه بوده است، رسید. تستی که تا ابد راز نهفته در آن کشف نخواهد شد!

پس از جایگذاری نهایی به فرم محاسبات طراح محترم رابطه زیر را خواهیم داشت:

$$\left[(۴۰۰+۴۰۰+۲۰)\right]+\left[۱\times(۴۰۰+۴۰۰)+\text{Max}\left(۳۰,۳۰,۳۰\right)\right]$$

$$\rightarrow [(۴۰۰+۴۰۰+۲۰)]+[۱\times(۴۰۰+۴۰۰)+۳۰]$$

$$\rightarrow [(۸۲۰)]+[۸۳۰]=۱۶۵۰ \text{ mses}$$

همانطور که گفتیم سازمان سنجش آموزش کشور، در کلید اولیه خود، گزینه دوم را به عنوان پاسخ اعلام کرده بود. اما در کلید نهایی این سوال حذف گردید، که کار درستی بوده است.

۶- گزینه (۴) صحیح است.

لایه شبکه سه مولفه اصلی دارد: (۱) پروتکل IP، (۲) پروتکل‌های مسیریابی اینترنت همچون RIP، OSPF و BGP، و (۳) پروتکل پیام کنترل اینترنت ICMP که در ادامه در مورد آن صحبت می‌کنیم.

پروتکل پیام کنترل اینترنت (Internet Control Message Protocol - ICMP) برای مبادله اطلاعات لایه شبکه مابین میزبان‌ها و مسیریاب‌ها مورد استفاده قرار می‌گیرد. بیشترین کاربرد ICMP برای گزارش کردن خطاهاست. اغلب (به اشتباه) ICMP را بخشی از IP می‌دانند، ولی ICMP از نظر ساختاری درست بالای IP قرار گرفته است، چون پیام‌های ICMP هم (درست مثل سگمنت‌های TCP یا UDP) به عنوان محموله داده در داخل دیتاگرام IP حمل می‌شوند. وقتی یک میزبان دیتاگرامی دریافت می‌کند که ICMP به عنوان پروتکل لایه بالای آن مشخص شده‌است، درست همانند حالتی که لایه شبکه محتویات دیتاگرام‌ها را به TCP یا UDP تحویل می‌دهد، محتویات این دیتاگرام را نیز به ICMP تحویل می‌دهد.

پیام‌های ICMP یک فیلد نوع(type)، یک فیلد کد(code)، و یک سرآیند دارند، آنها همچنین هشت بایت ابتدایی دیتاگرام IP که منشا تولید این پیام ICMP بوده است را در خود حمل می‌کنند (تا فرستنده بتواند تشخیص دهد کدام دیتاگرام باعث بروز خطا شده است) توجه داشته باشید که اعلام وضعیت‌های خطا فقط یکی از کاربردهای ICMP است.

برنامه مدیریتی Trace Route برای کشف مسیر موجود مابین یک گره مبدا و یک گره مقصد مورد استفاده قرار می‌گیرد. همچنین در برنامه مدیریتی Trace Route پیام Time Exceeded موجود در پروتکلICMP مورد استفاده قرار می‌گیرد. برای کشف نام و آدرس مسیریاب‌های واقع در مسیر مابین گره مبدا و گره مقصد، برنامه مدیریتی Trace Route گره مبدا تعدادی دیتاگرام IP معمولی به سمت گره مقصد ارسال می‌کند. هر یک از این دیتاگرام‌ها یک سگمنت UDP با شماره پورتی نامحتمل(شماره‌ای که به احتمال زیاد در هیچ گره‌ای به کار نرفته است) حمل می‌کنند. برنامه مدیریتی Trace Route طول عمر (TTL) بسته اول را به ۱، بسته دوم را به ۲، بسته سوم را به ۳ و الی آخر...مقداردهی می‌کند، و یک تایمر جداگانه برای هر یک از این دیتاگرام‌ها به راه می‌اندازد. وقتی دیتاگرام n ام به مسیریاب n ام برسد، مسیریاب متوجه می‌شود که TTL این دیتاگرام منقضی شده است. بنابر قواعد پروتکلIP، مسیریاب باید این دیتاگرام را دور بیاندازد و یک پیام اخطار ICMP با عنوان Time Exceeded و به معنی انقضای TTL (نوع ۱۱ و کد ۰) به گره مبدا برگرداند، که در این پیام اخطار یعنی Time Exceeded، نام و آدرس IP مسیریاب درج شده است. با رسیدن این پیام ICMP به گره مبدا، برنامه مدیریتی Trace Route می‌تواند (علاوه بر نام و آدرس IP

مسیریاب n ام) به کمک تایمر مربوطه زمان رفت و برگشت آن و به تبع فاصله زمانی تا مسیریاب n ام را نیز به دست آورد. به این صورت که ابتدا اولین دیتاگرام معمولی از گره مبدا به گره مقصد با TTL برابر با مقدار ۱ ارسال می‌گردد. مطابق قواعد پروتکل IP مقدار فیلد TTL در هر مسیریاب، یک واحد کاهش می‌یابد و اگر در مسیریابی مقدار فیلد TTL برابر صفر گردد از سوی مسیریاب موجود یک پیام اخطار ICMP با عنوان Time Exceeded و به معنی انقضای TTL (نوع ۱۱ و کد ۰) به گره مبدا ارسال می‌گردد. بنابراین مقدار فیلد TTL دیتاگرام اول ارسال شده از گره مبدا به گره مقصد در اولین مسیریاب برابر صفر شده و یک پیام اخطار ICMP با عنوان Time Exceeded و به معنی انقضای TTL (نوع ۱۱ و کد ۰) به گره مبدا ارسال می‌گردد. در ادامه دومین دیتاگرام معمولی از گره مبدا به گره مقصد با TTL برابر با مقدار ۲ ارسال می‌گردد. بنابراین مقدار فیلد TTL دیتاگرام دوم ارسال شده از گره مبدا به گره مقصد در دومین مسیریاب برابر صفر شده و یک پیام اخطار ICMP با عنوان Time Exceeded و به معنی انقضای TTL (نوع ۱۱ و کد ۰) به گره مبدا ارسال می‌گردد. همچنین در ادامه سومین دیتاگرام معمولی از گره مبدا به گره مقصد با TTL برابر با مقدار ۳ ارسال می‌گردد. بنابراین مقدار فیلد TTL دیتاگرام سوم ارسال شده از گره مبدا به گره مقصد در سومین مسیریاب برابر صفر شده و یک پیام اخطار ICMP با عنوان Time Exceeded و به معنی انقضای TTL (نوع ۱۱ و کد ۰) به گره مبدا ارسال می‌گردد. این روند کشف مسیریاب‌های مابین گره مبدا و گره مقصد تا کشف آخرین مسیریاب ادامه پیدا می‌کند. اما سوال اینست که گره مبدا برنامه مدیریتی Trace Route چگونه متوجه می‌شود که ارسال قطعه‌های UDP را چه زمانی باید متوقف کند؟ همانطور که گفتیم برنامه مدیریتی Trace Route فیلد TTL هر دیتاگرام ارسالی از گره مبدا به گره مقصد را افزایش می‌دهد، بنابراین یکی از این دیتاگرام‌ها درنهایت به گره مقصد خواهد رسید. از آنجا که (به خاطر نامربوط بودن شماره پورت آن) هیچ فرآیندی در نگاه end-to-end منتظر این سگمنت UDP نیست، گره مقصد یک پیام خطای ICMP با عنوان Port Unreachable و به معنی دسترسی به پورت مقصد مقدور نیست (نوع ۳ و کد ۳) به گره مبدا برمی‌گرداند. با دریافت این پیام ICMP خاص، گره مبدا متوجه می‌شود که دیگر نیازی به ارسال بسته‌های دیتاگرام بعدی نیست. با این روند، گره مبدا می‌تواند از تعداد و هویت مسیریاب‌های مابین خود و گره مقصد، و همچنین زمان رفت و برگشت مابین خود و گره مقصد آگاه شود. بنابراین گزینه چهارم پاسخ سوال است. دقت کنید که برنامه مدیریتی Trace Route باید بتواند به سیستم عامل دستور دهد تا دیتاگرام‌هایی با TTL مشخص تولید کند، و همچنین باید بتواند از دریافت پیام‌های ICMP توسط سیستم عامل مطلع شود. همچنین دقت کنید که تاخیر رفت و برگشت هر دیتاگرام تا مسیریاب مورد نظر شامل تمامی تاخیرها همچون تاخیر انتقال، تاخیر انتشار، تاخیر پردازش در هر مسیریاب و تاخیر صف می‌باشد. از آنجا که تاخیر صف در طول زمان نوسان دارد، گاهی ممکن است تاخیر رفت و برگشت دیتاگرام n (دیتاگرام مخصوص مسیریاب n) بیشتر از تاخیر رفت و برگشت دیتاگرام n+1 (دیتاگرام مخصوص مسیریاب n+1) باشد.

مشاوره تخصصی رشته کامپیوتر و IT

در راستای رسالت **مؤسسهٔ فرهنگی و انتشاراتی بابان** مبنی بر ارتقای سطح علـم و دانش کشور و کمک همه جانبه به دانشجویان و داوطلبان گرامـی، در جهـت قبـولی در کنکور کارشناسی ارشد و دکتری مهندسـی کـامپیوتر و IT دو طـرح زیـر را پایـه‌ریـزی کرده‌ایم:

۱) ارائه مشاوره تخصصی **حضوری** و **غیرحضوری** (تلفنی و آنلاین)

۲) برگزاری کلاس‌های **حضوری** و **غیرحضوری** (فیلم آموزشی و کلاس آنلاین)

بـدین منظـور بـرای تمـاس بـا شـما و تعیـین وقـت قبلـی عـدد **۲** را بـه شـماره **۵۰۰۰۲۰۶۰۴۵** پیامک نمائید.

برای آشنایی بیشتر با خدمات ارائـه شـده توسـط **موسسـه بابـان** بـه وب سـایت **khalilifar.ir** یا کانال تلگرام **@arastookhalilifar** مراجعه فرمایید.

تلفن دفتر مرکزی موسسه بابان: ۷۷۹۷۲۸۶۸-۰۲۱

پایگاه اطلاع رسانی موسسه بابان: www.baban.ir

تلفن دفتر فروشگاه انتشارات بابان: ۷۷۹۷۳۳۸۶-۰۲۱

فروشگاه اینترنتی انتشارات بابان: shop.baban.ir