

موسسه بابان

انتشارات بابان و انتشارات راهیان ارشد

درس و کنکور ارشد

شبکه‌های کامپیوتری

(حل تشریحی سوالات دولتی ۱۳۹۴)

ویژه‌ی داوطلبان کنکور کارشناسی ارشد مهندسی کامپیوتر و IT

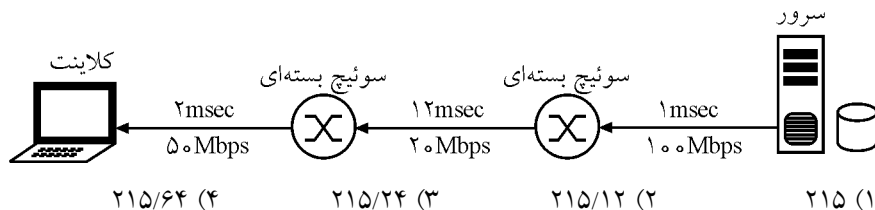
براساس کتب مرجع

کراس راس و لئون گارسیا

ارسطو خلیلی فر

تست‌های سال ۹۴

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



۲- 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-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 10^6 bps باشد، زمان لازم برای دریافت این صفحه بر حسب ثانیه (sec) چقدر است؟ زمان رفت و برگشت (RTT) بین کلاینت و سرور را 0.1 ثانیه در نظر بگیرید.

۱/۵ (۴)
۱/۴ (۳)
۱/۲ (۲)
۱/۵ (۱)

۳- TCP Reno فایلی را به ۳۲ بسته تبدیل کرده و برای مقصدی ارسال می‌کند. چنانچه بسته ۲۷ ام 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)

۴- 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)

پاسخ تست‌های سال ۹۴

۱- گزینه (۳) صحیح است.

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 تأخیر 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)

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

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

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:

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

T_{Prop} ، زمان تأخیر انتشار است.

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

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

$$T_{\text{Process}} = \frac{b}{R}$$

$T_{Process}$ ، زمان پردازش موجود در مسیریاب (گره میانی) مربوط به کنترل خطای فریم، احیاناً 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:

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

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:

$$L = 1000 \text{ Byte}$$

$$T_{\text{Prop1}} = 1 \text{ msec}, T_{\text{Prop2}} = 12 \text{ msec}, T_{\text{Prop3}} = 2 \text{ msec}$$

$$T_{\text{Process1}} = 0, T_{\text{Process2}} = 0$$

$$R_1 = 100 \text{ Mbps}, R_2 = 20 \text{ Mbps}, R_3 = 50 \text{ Mbps}$$

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

$$T_{\text{Total Delay}} = [T_{F1}] + T_{\text{Prop1}} + [T_{\text{Process1}} + T_{F2}] + T_{\text{Prop2}} + [T_{\text{Process2}} + T_{F3}] + T_{\text{Prop3}} + T_{\text{queue}}$$

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:

$$T_{\text{Total Delay}} = [0/0.8] + 1 + [0+0/4] + 12 + [0+0/16] + 2 + (499) \times (0/4)$$

$$T_{\text{Total Delay}} = [0/0.8] + 1 + [0/4] + 12 + [0/16] + 2 + 199/6 = 15/64 + 199/6 = 215/24 \text{ msec}$$

۲- گزینه (۱) صحیح است.

پروتکل HTTP در لایه کاربرد

به برنامه کاربردی که روی اینترنت نوشته شده است، world wide web یا شبکه جهانی وب گفته می‌شود. زیرا documentهایی داریم که Linkها را به هم متصل می‌کند، پروتکلی که برای آن طراحی شده است، پروتکل HTTP (HyperText Transfer Protocol) نام دارد. کاری که HTTP انجام می‌دهد این است که clientها، objectها را به web server، request می‌دهند و web server هم objectها را می‌آورد. objectها می‌توانند یک فایل HTML با یک تصویر JPEG و ... باشند که توسط این پروتکل می‌توانند منتقل شوند. هر object ای در محیط عملیاتی اینترنت با یک آدرس منحصر به فرد معرفی می‌شود که به آن URL گفته می‌شود. URL سرواژه عبارت Uniform Resource Locator می‌باشد.

مثال:

www.iust.ac.ir/index.htm
[/home/logo.jpg](http://www.iust.ac.ir/home/logo.jpg)
[/home/Header.jpg](http://www.iust.ac.ir/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 Client، 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 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.

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 دوستان 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 معمولاً به این صورت است. به این زمان رفت و برگشت اصطلاحاً 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 آن‌ها 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:

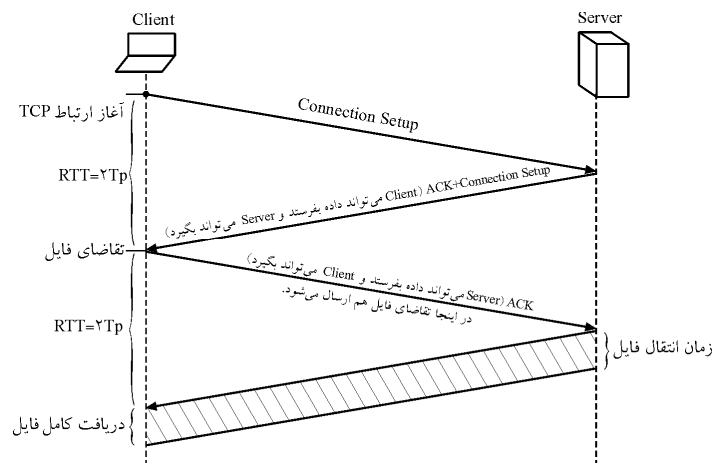
۱- اول این کار را انجام بده

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:

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

توجه: فرض کنید در مرورگر وب خود برای دریافت یک صفحه وب به طور کامل بر روی یک لینک کلیک می‌کنید و آدرس IP مربوط به این URL در میزبان محلی ذخیره نشده است، در نتیجه برای به دست آوردن آدرس IP به یک DNS LOOK UP نیاز است. فرض کنید برای دریافت آدرس IP از طریق سرویس DNS، n سرور DNS ملاقات می‌شوند و تاخیر زمان رفت و برگشت معادل RTT_1 تا RTT_n باشد. بنابراین بدون در نظر گرفتن زمان مربوط به درخواست و دریافت فایل پایه html و objectهای موجود در آن، رابطه زیر را خواهیم داشت:

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

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

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

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

Non-persistent http ناپایدار (غیرمصر یا غیرمداوم)

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

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

روش ناپایدار ترتیبی:

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

روش ناپایدار موازی نامحدود:

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

روش ناپایدار موازی محدود:

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

persistent http (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:

درخواست و برقراری ارتباط	زمان انتقال فایل اصلی	درخواست و برقراری ارتباط	زمان انتقال object
\uparrow	\uparrow	\uparrow	\uparrow
$(RTT + RTT + T_F)$		$n \times (RTT + RTT)$	$\sum_{i=1}^n T_F(i)$
\downarrow	\downarrow	\downarrow	\downarrow
درخواست و دریافت فایل اصلی		تعداد درخواست‌ها درخواست و دریافت object	

پایدار موازی نامحدود:

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

روش پایدار موازی محدود:

درخواست و برقراری ارتباط	زمان انتقال فایل اصلی	درخواست و برقراری ارتباط	زمان انتقال object
\uparrow	\uparrow	\uparrow	\uparrow
$(RTT + RTT + T_F)$		$n \times (RTT + RTT)$	$\sum_{i=1}^n T_F(i)$
\downarrow	\downarrow	\downarrow	\downarrow
درخواست و دریافت فایل اصلی		تعداد درخواست‌ها درخواست موازی و دریافت موازی object	

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

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

در صورت سوال گفته شده است که 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-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 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.

۱/۰ (۱) ۱/۲ (۲) ۱/۴ (۳) ۱/۵ (۴)

داده‌های مسئله به صورت زیر است:

$$L_{\text{Base HTML}} = 5000 \text{ Byte}, L_{\text{Object}} = 5000 \text{ Byte}$$

$$R_{\text{TOTAL CHANNEL}} = 10^6 \text{ bps}, RTT = 0.1 \text{ s}$$

$$\text{Cardinality}(\text{Object}) = 9, T_{\text{DNS LOOK UP}} = 0$$

رابطه روش ناپایدار موازی محدود به صورت زیر است:

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

$T_{F(\text{Base HTML})}$ 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 $L_{\text{Base HTML}}$ برابر اندازه فایل پایه html و $R_{\text{TOTAL CHANNEL}}$ برابر نرخ انتقال کل کانال می‌باشد.

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

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

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

که L_{Object} برابر اندازه Object و $R_{\text{TOTAL CHANNEL}}$ 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 با عنوان تعداد درخواست‌ها برابر ۲ خواهد بود، زیرا دو کانکشن یکی ۵ تایی و دیگری ۴ تایی ایجاد کردیم.

که پس از جایگذاری اولیه رابطه زیر را خواهیم داشت:

$$\left[\left(\text{RTT} + \text{RTT} + \frac{5000 \times 8}{10^6} \right) \right] + \left[2 \times \left(\text{RTT} + \text{RTT} \right) + 9 \times \frac{5000 \times 8}{10^6} \right]$$

پس از جایگذاری نهایی رابطه زیر را خواهیم داشت:

$$\left[\left(\frac{0}{1} + \frac{0}{1} + \frac{0}{0.4} \right) \right] + \left[2 \times \left(\frac{0}{1} + \frac{0}{1} \right) + 9 \times \frac{0}{0.4} \right] = \left[\frac{0}{24} \right] + \left[\frac{0}{4} + \frac{0}{36} \right]$$

$$\rightarrow \left[\frac{0}{24} \right] + \left[\frac{0}{76} \right] = 1 \text{ s}$$

مثال: فرض کنید کامپیوتر A می‌خواهد یک فایل HTML به حجم ۳/۶ کیلوبایت را که خود حاوی 3 object که حجم هر کدام از آن‌ها نیز ۳/۶ کیلوبایت است را از وب سرور B دریافت کند. اگر پهنای باند دریافت A، ۱ Mbps و تأخیر انتشار یک طرفه ۲۰ms باشد و با فرض اینکه طول پیغام‌های کنترل ناچیز باشد و حداکثر اندازه هر سگمنت ۱/۸ Mss کیلوبایت است و اندازه پنجره TCP همواره ثابت و برابر یک باشد و در حین انتقال هیچگونه خطا و ازدحامی رخ ندهد، به روش ناپایدار و ترتیبی دریافت این صفحه وب به طور کامل چقدر طول می‌کشد؟

پاسخ:

ابتدا زمان انتقال یک سگمنت را محاسبه می‌کنیم:

$$T_{\text{segment}} = \frac{L}{R} = \frac{1/8 \times 10^3 \times 8}{10^6} = 14/4 \text{ ms} \rightarrow T_F = 2 \times T_{\text{segment}} = 2 \times 14/4 = 28/8$$

$$\text{RTT} = 2 \times T_p = 2 \times 20 \text{ ms} = 40 \text{ ms}$$

توجه: دقت کنید فایل اصلی و object ها، هر کدام در ۲ سگمنت جا می‌شوند.

$$(\overline{RTT} + \overline{RTT} + \overline{T_F}) + \left[3 \times (\overline{RTT} + \overline{RTT}) + \sum_{i=1}^3 \overline{T_F}(i) \right]$$

$\frac{3 \times 28/8 = 86/4}{40}$

برای ۲ سگمنت ۲ بار برای ۲ سگمنت ۲ بار
 درخواست و دریافت درخواست و دریافت
 $2 \times 40 = 80$ $2 \times 40 = 80$

→ $480 + 115/2 = 595/2$ ms

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

$$(\overline{RTT} + \overline{RTT} + \overline{T_F}) + \left[1 \times (\overline{RTT} + \overline{RTT}) + \sum_{i=1}^3 \overline{T_F}(i) \right]$$

$\frac{3 \times 28/8 = 86/4}{40}$

برای ۲ سگمنت ۲ بار برای ۲ سگمنت ۲ بار
 درخواست و دریافت درخواست و دریافت
 $2 \times 40 = 80$ $2 \times 40 = 80$

→ $240 + 115/2 = 355/2$ 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 $2 \times 40 = 80$ ms هر ۳ object دریافت می‌گردند.

۳- گزینه (۱) صحیح است.

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 RCP 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 سؤال 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.

۴- گزینه (۲) صحیح است.

توجه: این سؤال، دقیقاً همان سؤال دولتی ۹۱، مبحث لایه شبکه است. بهترین راه حل برای حل این مدل از مسئله‌ها، مشخص کردن محدوده‌ی 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 زیرشبکه است.

زیرشبکه اول

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

NetID → ۱۳۵ . ۴۶ . ۵۶ . ۰
تا

Broadcast → ۱۳۵ . ۴۶ . ۵۹ . ۲۵۵

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

$\begin{array}{cccccccc} & 128 & 64 & 32 & 16 & 8 & 4 & \\ 135 & . & 46 & . & 56 & . & 0 & . & 0 \end{array}$	$\begin{array}{cccccccc} 21 & & & & & & & & \\ 00 & . & 0 & . & 0 & . & 0 & . & 0 \end{array}$
---	--

NetID

Host ID

۱۳۵ . ۴۶ . ۵۶ . ۰

$\begin{array}{cccccccc} & 128 & 64 & 32 & 16 & 8 & 4 & \\ 135 & . & 46 & . & 56 & . & 0 & . & 0 \end{array}$	$\begin{array}{cccccccc} 21 & & & & & & & & \\ 11 & . & 1 & . & 1 & . & 1 & . & 1 \end{array}$
---	--

NetID

Host ID

۱۳۵ . ۴۶ . ۵۹ . ۲۵۵

زیرشبکه دوم

برای زیر شبکه دوم، با توجه به MASK آن که برابر ۲۲ است و به معنی وجود $2^{10} = 1024$ آدرس در این زیرشبکه 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 style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; text-align: right;">۱۲۸</td> <td style="width: 25%; text-align: right;">۶۴</td> <td style="width: 25%; text-align: right;">۳۲</td> <td style="width: 25%; text-align: right;">۱۶</td> <td style="width: 25%; text-align: right;">۸</td> <td style="width: 25%; text-align: right;">۴</td> </tr> <tr> <td style="text-align: center;">۱۳۵</td> <td style="text-align: center;">۴۶</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۱</td> <td style="text-align: center;">۱</td> </tr> </table>	۱۲۸	۶۴	۳۲	۱۶	۸	۴	۱۳۵	۴۶	۰	۰	۱	۱	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; text-align: right;">۲۱</td> <td style="width: 25%; text-align: right;">۱۱</td> <td style="width: 25%; text-align: right;">۱۱</td> <td style="width: 25%; text-align: right;">۱۱</td> </tr> <tr> <td style="text-align: center;">۰</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۰</td> </tr> </table>	۲۱	۱۱	۱۱	۱۱	۰	۰	۰	۰
۱۲۸	۶۴	۳۲	۱۶	۸	۴																
۱۳۵	۴۶	۰	۰	۱	۱																
۲۱	۱۱	۱۱	۱۱																		
۰	۰	۰	۰																		

NetID

Host ID

۱۳۵ . ۴۶ . ۶۰ . ۰

<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; text-align: right;">۱۲۸</td> <td style="width: 25%; text-align: right;">۶۴</td> <td style="width: 25%; text-align: right;">۳۲</td> <td style="width: 25%; text-align: right;">۱۶</td> <td style="width: 25%; text-align: right;">۸</td> <td style="width: 25%; text-align: right;">۴</td> </tr> <tr> <td style="text-align: center;">۱۳۵</td> <td style="text-align: center;">۴۶</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۱</td> <td style="text-align: center;">۱</td> </tr> </table>	۱۲۸	۶۴	۳۲	۱۶	۸	۴	۱۳۵	۴۶	۰	۰	۱	۱	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; text-align: right;">۲۱</td> <td style="width: 25%; text-align: right;">۱۱</td> <td style="width: 25%; text-align: right;">۱۱</td> <td style="width: 25%; text-align: right;">۱۱</td> </tr> <tr> <td style="text-align: center;">۱۱</td> <td style="text-align: center;">۱</td> <td style="text-align: center;">۱</td> <td style="text-align: center;">۱</td> </tr> </table>	۲۱	۱۱	۱۱	۱۱	۱۱	۱	۱	۱
۱۲۸	۶۴	۳۲	۱۶	۸	۴																
۱۳۵	۴۶	۰	۰	۱	۱																
۲۱	۱۱	۱۱	۱۱																		
۱۱	۱	۱	۱																		

NetID

Host ID

۱۳۵ . ۴۶ . ۶۳ . ۲۵۵

زیرشبکه سوم

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

۱۹۲ . ۵۳ . ۴۰ . ۰ → NetID

تا

۱۳۵ . ۵۳ . ۴۱ . ۲۵۵ → Broadcast

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 style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; text-align: right;">۱۲۸</td> <td style="width: 25%; text-align: right;">۶۴</td> <td style="width: 25%; text-align: right;">۳۲</td> <td style="width: 25%; text-align: right;">۱۶</td> <td style="width: 25%; text-align: right;">۸</td> <td style="width: 25%; text-align: right;">۴</td> <td style="width: 25%; text-align: right;">۲</td> </tr> <tr> <td style="text-align: center;">۱۹۲</td> <td style="text-align: center;">۵۳</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۱</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۱</td> </tr> </table>	۱۲۸	۶۴	۳۲	۱۶	۸	۴	۲	۱۹۲	۵۳	۰	۰	۱	۰	۱	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; text-align: right;">۲۱</td> <td style="width: 25%; text-align: right;">۱۱</td> <td style="width: 25%; text-align: right;">۱۱</td> <td style="width: 25%; text-align: right;">۱۱</td> </tr> <tr> <td style="text-align: center;">۰</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۰</td> </tr> </table>	۲۱	۱۱	۱۱	۱۱	۰	۰	۰	۰
۱۲۸	۶۴	۳۲	۱۶	۸	۴	۲																	
۱۹۲	۵۳	۰	۰	۱	۰	۱																	
۲۱	۱۱	۱۱	۱۱																				
۰	۰	۰	۰																				

NetID

Host ID

۱۹۲ . ۵۳ . ۴۰ . ۰

<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; text-align: right;">۱۲۸</td> <td style="width: 25%; text-align: right;">۶۴</td> <td style="width: 25%; text-align: right;">۳۲</td> <td style="width: 25%; text-align: right;">۱۶</td> <td style="width: 25%; text-align: right;">۸</td> <td style="width: 25%; text-align: right;">۴</td> <td style="width: 25%; text-align: right;">۲</td> </tr> <tr> <td style="text-align: center;">۱۹۲</td> <td style="text-align: center;">۵۳</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۱</td> <td style="text-align: center;">۰</td> <td style="text-align: center;">۱</td> </tr> </table>	۱۲۸	۶۴	۳۲	۱۶	۸	۴	۲	۱۹۲	۵۳	۰	۰	۱	۰	۱	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; text-align: right;">۲۱</td> <td style="width: 25%; text-align: right;">۱۱</td> <td style="width: 25%; text-align: right;">۱۱</td> <td style="width: 25%; text-align: right;">۱۱</td> </tr> <tr> <td style="text-align: center;">۱</td> <td style="text-align: center;">۱</td> <td style="text-align: center;">۱</td> <td style="text-align: center;">۱</td> </tr> </table>	۲۱	۱۱	۱۱	۱۱	۱	۱	۱	۱
۱۲۸	۶۴	۳۲	۱۶	۸	۴	۲																	
۱۹۲	۵۳	۰	۰	۱	۰	۱																	
۲۱	۱۱	۱۱	۱۱																				
۱	۱	۱	۱																				

NetID

Host ID

۱۹۲ . ۵۳ . ۴۱ . ۲۵۵

حال با توجه به اینکه آدرس مقصد برابر با مقدار ۲ . ۵۲ . ۴۶ . ۱۳۵ است و این آدرس در محدوده هیچ یک از سه زیرشبکه فوق قرار ندارد، بنابراین گام بعدی بسته مورد نظر، Router_۲ است.

۵- گزینه (۳) صحیح است.

رابطه کلی برای محاسبه بهره‌وری در stop & wait (IDLE) به صورت زیر است:
محاسبه بهره‌وری stop & wait بدون صرف نظر کردن از سربار ACK 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.

توجه: در روابط فوق مقدار پنجره فرستنده یعنی W برابر یک است.

توجه: W ، اندازه پنجره سمت فرستنده است و $W_S = \frac{T_O}{T_F}$ برابر با «حداقل» اندازه پنجره که برای «ماکزیمم کردن راندمان» مورد نیاز است.
 T_F و T_O از رابطه‌ی زیر به دست می‌آید:

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

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

T_F ، زمان انتقال فریم داده به داخل کانال انتقال است.

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

محاسبه بهره‌وری stop & wait با صرف نظر کردن از سربار ACK و زمان پردازش، به صورت زیر است:

بنابراین مقدار W_S برابر رابطه‌ی زیر خواهد بود:

$$W_S = \frac{T_F + 2T_p + \overset{\text{ناچیز}}{T_{\text{Ack}}} + \overset{\text{ناچیز}}{2T_{\text{Process}}}}{T_F} = \frac{T_F + T_p}{T_F} = \frac{1 + 2\frac{T_p}{T_F}}{1} = \frac{1 + 2a}{1} = 1 + 2a$$

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

بنابراین با توجه به روابط فوق داریم:

$$U_{\text{stop \& wait}} = \begin{cases} \frac{1}{1 + 2a} \left(1 - \frac{H}{L}\right) (1 - P_F) & W < 1 + 2a \quad (1) \\ \left(1 - \frac{H}{L}\right) (1 - P_F) & W \geq 1 + 2a \quad (2) \end{cases}$$

توجه: برای بهره‌وری ۱۰۰٪ باید $W \geq 1 + 2a$ باشد، بنابراین از رابطه دوم استفاده می‌گردد.
 توجه: در روابط فوق مقدار پنجره فرستنده یعنی W برابر یک است.
 توجه: در صورت سؤال از سر بار ACK صرف نظر شده است و همچنین صحبتی از زمان پردازش فریم نشده است، بنابراین روابط فوق برای محاسبه بهره‌وری مورد استفاده قرار می‌گیرند.
 توجه: در صورت سؤال مطرح شده است که برای داشتن بهره‌وری بیشتر از ۶۰ درصد، مدت زمان ارسال فریم باید حداقل چند برابر تاخیر انتشار یکطرفه باشد؟
 بنابراین $W < 1 + 2a$ است، لذا رابطه‌ی زیر برای محاسبه بهره‌وری مورد استفاده قرار می‌گیرد:

$$U_{\text{stop \& wait}} = \frac{1}{1+2a} \left(1 - \frac{H}{L}\right) (1 - P_F) \quad W < 1 + 2a$$

توجه: در صورت سؤال از سر بار سرآیند فریم $\left(\frac{H}{L}\right)$ صرفه نظر شده است، بنابراین در رابطه فوق، مقدار آن، صفر در نظر گرفته می‌شود، پس داریم:

$$U_{\text{stop \& wait}} = \frac{1}{1+2a} (1 - P_F) \quad W < 1 + 2a$$

در صورت سؤال مطرح شده است که برای داشتن بهره‌وری بیشتر از ۶۰ درصد، مدت زمان ارسال فریم باید حداقل چند برابر تاخیر انتشار یکطرفه باشد؟
 بنابراین مطابق رابطه فوق داریم:
 داده‌های مسئله به صورت زیر است:

$$P_F = 0/2, (1 - P_F) = 0/8$$

$$U_{\text{stop \& wait}} = \frac{1}{1+2a} \times (1 - P_F) \rightarrow \frac{0/8}{1+2a} \geq 0/6$$

$$0/8 \geq 0/6(1+2a) \rightarrow$$

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

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

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

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

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

۶- گزینه (۲) صحیح است.

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. راه حل این است که آنقدر اسباب و وسایل (وسایل اصلی و وسایل فرعی) داشته باشید که مدام داخل ماشین قرار دهید که سبب معطلی شما گردد، آنقدری 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 مسئله‌ای، 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:

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

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

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

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

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