

موسسه بابان

انتشارات بابان و انتشارات راهیان ارشد

درس و کنکور ارشد

شبکه‌های کامپیوتری

(حل تشریحی سوالات دولتی ۱۳۹۷)

ویژه‌ی داوطلبان کنکور کارشناسی ارشد مهندسی کامپیوتر و IT

براساس کتب مرجع

کراس راس و لئون گارسیا

ارسطو خلیلی فر

تست‌های فصل پنجم

۵۵- فرض کنید سه بسته از سه اینترنتی مختلف هم زمان وارد یک مسیریاب از نوع دیتاگرام می‌شوند. مسیریاب عمل مسیریابی برای بسته‌ها را به صورت موازی (هم زمان) انجام می‌دهد و با مشخص شدن اینترنتی خروجی، بسته را برای سوییچینگ در اختیار سوییچ درون مسیریاب قرار می‌دهد. فرض کنید سوییچ درون مسیریاب از نوع bus است و زمان سوییچینگ برای هر بسته ۱ میکروثانیه است.
چنانچه:

- هنگام ورود این سه بسته مسیریاب خالی باشد.
 - مسیریاب هر سه بسته را به یک اینترنتی خروجی یکسان بفرستد.
 - زمان ارسال (transmission) هر بسته از اینترنتی خروجی ۷ میکروثانیه باشد. (۷ میکروثانیه طول می‌کشد تا یک بسته از اینترنتی خروجی ارسال شود).
- هر بسته چه مدت در صف اینترنتی خروجی معطل می‌ماند؟
- (۱) یک بسته در صف خروجی معطل نمی‌شود.
 - یک بسته ۱ میکروثانیه در صف خروجی معطل می‌شود.
 - یک بسته ۷ میکروثانیه در صف خروجی معطل می‌شود.
 - (۲) یک بسته در صف خروجی معطل نمی‌شود.
 - یک بسته ۲ میکروثانیه در صف خروجی معطل می‌شود.
 - یک بسته ۵ میکروثانیه در صف خروجی معطل می‌شود.
 - (۳) یک بسته در صف خروجی معطل نمی‌شود.
 - یک بسته ۶ میکروثانیه در صف خروجی معطل می‌شود.
 - یک بسته ۱۲ میکروثانیه در صف خروجی معطل می‌شود.
 - (۴) یک بسته در صف خروجی معطل نمی‌شود.
 - یک بسته ۷ میکروثانیه در صف خروجی معطل می‌شود.
 - یک بسته ۱۴ میکروثانیه در صف خروجی معطل می‌شود.

پاسخ‌های فصل پنجم

۵۵- گزینه (۳) صحیح است.

آنچه در درس سیستم عامل و شبکه‌های کامپیوتری حائز اهمیت قرار گرفته است، پدیده‌ی زمان است، کاربر نهایی تاخیر کمتر را مطالبه می‌کند و می‌پسندد، بنابراین آنچه در حل اغلب مسائل سیستم عامل و شبکه‌های کامپیوتری با آن مواجه هستیم، بررسی پدیده‌ی زمان است. گاه از جنس زمان کل و بازگشت، گاه از جنس زمان انتظار و گاه از جنس زمان اجرا است. این سوال زمان انتظار را مورد پرسش قرار داده است. جهت درک بهتر و واضح‌تر این سوال، این سوال رو به یک پرسش سیستم عاملی تبدیل می‌کنیم و به همان سبک و سیاق حل خواهیم کرد. در ادامه جهت درک و شناخت عمیق‌تر و لذت بیشتر چند سطر هم از ساختار سیستم عامل و سیستم‌های کامپیوتری با شما می‌گوییم:

انسان به واسطه‌ی نیازهایی که دارد، برای برآورده شدن آنها می‌بایست تلاش و حرکت کند. انسان نیاز به کتاب، کلاس، تحصیل، مشاوره، آموزش، همایش، خوراک، پوشاک، خانه، امنیت، سلامت، آرایش و زیبایی، سرگرمی، مسافرت، اتومبیل، ارتباط، همسر، فرزند و ... دارد.

سوال: کاربرد مباحث کامپیوتر و آی تی در کجاست؟

پاسخ: بهبود فرآیند زندگی انسان‌ها: بهبود فرآیند برآورده‌سازی نیاز انسان‌ها. زندگی راحت‌تر

و بهتر.

این مهم از طریق تولید سیستم‌های کامپیوتری که شامل سخت‌افزارها و نرم‌افزارها می‌باشد محقق می‌گردد.

سخت‌افزار: بخش فیزیکی سیستم کامپیوتری است.

نرم‌افزار: بخش منطقی سیستم کامپیوتری است.

توجه: همکاری بخش فیزیکی و منطقی یک سیستم کامپیوتری، منجر به برآورده‌سازی نیاز

انسان‌ها می‌شود.

در حالت کلی، نرم‌افزارها به دو دسته زیر تقسیم می‌شوند:

۱- نرم‌افزارهای کاربردی

مرتفع‌کننده نیازهای انسان است. برنامه‌هایی که برای رفع نیازهای انسان نوشته می‌شود. مانند نرم‌افزارهای حسابداری و نرم‌افزار فرهنگ لغات.

۲- نرم‌افزارهای سیستمی

مرتفع‌کننده نیازهای انسان و نیازهای سایر برنامه‌ها است. برنامه‌هایی که برای رفع نیازهای انسان یا رفع نیازهای سایر برنامه‌ها نوشته می‌شوند. مانند سیستم عامل‌ها و کامپایلرها.

روش‌های مرتفع‌سازی نیاز انسان‌ها در سیستم‌های کامپیوتری

به طور کلی نیاز انسان‌ها در سیستم‌های کامپیوتری به دو روش زیر مرتفع می‌گردد:

۱- توسط سیستم عامل

در این حالت نیازهای انسان توسط سیستم عامل مرتفع می‌گردد. انسان درخواست‌هایش را تقدیم سیستم عامل می‌کند و سیستم عامل انجام می‌دهد بدون آنکه نیاز باشد انسان از جزئیات سخت‌افزار باخبر باشد. در واقع در این حالت سیستم عامل یک ماشین مجازی به شکلی کاملاً انتزاعی (بیانی) و نزدیک به زبان انسان برای انسان فراهم کرده‌است. انتزاع یعنی حذف جزئیات و کلی‌گویی. مانند درخواست نمایش لیست فایل‌ها، کپی و انتقال فایل‌ها توسط کاربران نهایی.

۲- توسط همکاری برنامه‌های کاربردی و سیستم عامل

در این حالت نیازهای انسان توسط همکاری برنامه‌های کاربردی و سیستم عامل مرتفع می‌گردد. لازم به ذکر است که نیازهای انسان توسط برنامه‌های کاربردی وقتی مرتفع می‌گردد که نیازهای برنامه‌های کاربردی جهت انجام توسط سیستم عامل مرتفع گردد که نتیجه آن برآورده شدن نیاز انسان است. بنابراین وظیفه تخصیص منابع به برنامه‌های کاربردی همچون حافظه اصلی، پردازنده و دستگاه‌های ورودی و خروجی به برنامه‌های کاربردی و همچنین مدیریت تخصیص منابع مابین برنامه‌های کاربردی مختلف موجود در یک سیستم کامپیوتری برعهده سیستم عامل است. سیستم عامل قبل از اجرا (پیش‌اجرا) و پس از اجرا (پسا اجرا) از برنامه کامپیوتری حمایت می‌کند و در زمان اجرای برنامه کامپیوتری توسط پردازنده (حین اجرا) حضور ندارد و سخت‌افزار از برنامه کامپیوتری حمایت می‌کند.

اگر بخواهیم دو عبارت فوق را در یک جمله بیان کنیم، اینطور خواهد بود، سیستم عامل دو وظیفه‌ی ایجاد ماشین مجازی و مدیریت منابع را برعهده دارد.

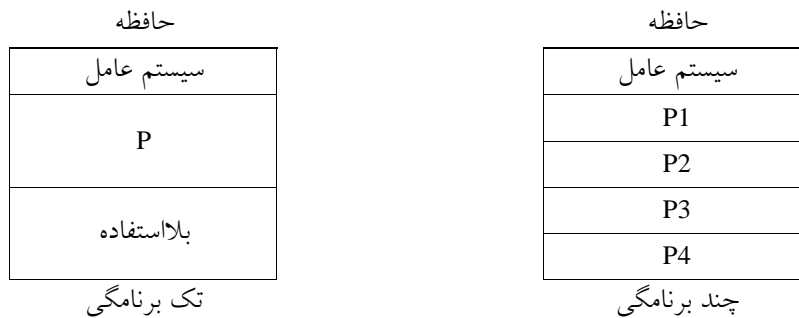
انواع نگاه به سیستم‌های کامپیوتری

به طور کلی از دو منظر می‌توان به سیستم‌های کامپیوتری نگاه کرد:

۱- از نگاه طراحان و سازندگان سیستم‌های کامپیوتری

آنچه از نگاه طراحان و سازندگان سیستم‌های کامپیوتری حائز اهمیت است، استفاده بهینه از منابع سیستم‌های کامپیوتری همچون حافظه اصلی، پردازنده و دستگاه‌های ورودی و خروجی است. که این امر مستلزم مدیریت بهینه منابع است. مانند نگاه شهردار به خیابان‌های شهری که دوست دارد از خیابان‌هایی که فراهم کرده است به طور بهینه استفاده گردد. و این نشود که یک خیابان با بار سنگین ترافیک مواجه گردد و در همان لحظه خیابان دیگری در همان نزدیکی‌ها با بار سبک ترافیک مواجه گردد، بنابراین کنترل و نظارت می‌کنند، گاهی یک طرفه، گاهی دو طرفه. برای مثال در دنیای کامپیوتر نیز انجام چند برنامه‌ی برنامه‌ها بهینه‌تر از انجام تک برنامه‌ها است. در الگوریتم چند برنامه‌ی وقتی یکی از برنامه‌ها منتظر پایان عملیات I/O است، برنامه دیگری می‌تواند از زمان آزاد CPU استفاده کند. به این ترتیب یک برنامه‌ریزی نسبتاً دقیق منجر به آن می‌شود که CPU تقریباً در تمام مواقع مشغول کار مفید باشد.

مفهوم چند برنامه‌ی در حافظه در شکل زیر نشان داده شده است:



تفاوت چند برنامه‌ی و تک برنامه‌ی

۲- از نگاه کاربران نهایی سیستم‌های کامپیوتری

آنچه از نگاه کاربران نهایی سیستم‌های کامپیوتری حائز اهمیت است، انجام سریع کارهاست، زیرا انسان ذاتاً انجام سریع کارها را می‌پسندد، در واقع انجام سریع کارها، زمان انتظار کوتاه، زمان پاسخ کوتاه و زمان گردش کار کوتاه یکی از نیازهای اساسی انسان‌هاست. که این امر مستلزم مدیریت بهینه کارهاست. مانند نگاه شهروند به خیابان‌های شهری که دوست دارد از خیابان‌هایی که توسط شهردار فراهم شده است، به سرعت به مقصد مورد نظر خود برسد. در دنیای کامپیوتر نیز اگر همواره کاری برای اجرا انتخاب شود که از همه کوتاه‌تر است (زمان اجرای کمتری داشته باشد) آنگاه شاهد حداقل زمان انتظار، زمان پاسخ و زمان گردش کار خواهیم بود. به عنوان نمونه‌ای دیگر اگر هنگام دسترسی به محلی از حافظه، آدرس محل حافظه مورد نظر در ثبات‌های پردازنده نگهداری شود به تبع زمان ترجمه آدرس بسیار کوتاه‌تر خواهد بود و در نتیجه زمان دسترسی به محل حافظه مورد نظر نیز کوتاه‌تر خواهد بود. و به طور کلی هر تلاشی جهت انجام

سریع‌تر کارها از سوی کاربر رضایت بخش خواهد بود و این‌ها خوب است، کاربران نهایی هم همین را می‌پسندند. ما برای انسان‌ها کار می‌کنیم، آنچه را که او می‌پسندد باید ایجاد کنیم.
توجه: لحاظ کردن هر دو نگاه در طراحی سیستم‌های کامپیوتری لازم و ضروری است. هم نگاه طراحان و سازندگان و هم نگاه کاربران نهایی.

همانطور که نحوه تخصیص منابع کشور به انسان‌ها توسط دولت‌ها مدیریت می‌شود، نحوه تخصیص منابع سیستم کامپیوتری به برنامه کامپیوتری نیز می‌بایست توسط سیستم عامل‌ها مدیریت شود. همانطور که منابع جهان هستی در خدمت انسان است و باید توسط دولت‌ها مدیریت شود، منابع سیستم کامپیوتری نیز در خدمت برنامه کامپیوتری است و باید توسط سیستم عامل‌ها مدیریت شود. همانطور که منابع جهان هستی جهت رفع نیازهای انسان مورد استفاده قرار می‌گیرد، منابع سیستم‌های کامپیوتری نیز جهت رفع نیازهای برنامه کامپیوتری مورد استفاده قرار می‌گیرد. انسان نیاز به خوراک، پوشاک و مسکن دارد و برنامه کامپیوتری نیز نیاز به حافظه اصلی، پردازنده و دستگاه‌های ورودی و خروجی دارد.

توجه: برای مطرح کردن، تحلیل و بررسی مسائل مربوط به زمانبندی پردازنده در درس سیستم عامل از تعاریف زیر استفاده می‌شود.

۱- زمان ورود

لحظه‌ای که یک فرآیند به لیست فرآیندهای آماده درون صف آماده سیستم اضافه می‌شود.

۲- زمان اجرا

مدت زمانی که یک فرآیند به زمان نیاز دارد که به آن زمان سرویس یا زمان انفجار محاسباتی (CBT = CPU Burst Time) نیز می‌گویند.

۳- زمان انتظار

برابر طول مدت زمانی که فرآیند در صف قرار داشته، آماده‌است و منتظر دریافت پردازنده است.

۴- زمان بازگشت (اتمام کامل)

برابر زمان اجرای فرآیند به علاوه زمان انتظار آن است.

الگوریتم (First Come First Served) FCFS

این الگوریتم ساده‌ترین الگوریتم زمانبندی پردازنده است. در این روش کارها با همان ترتیب ورود به سیستم، در یک صف قرار گرفته و از ابتدای صف به ترتیب، پردازنده را در اختیار می‌گیرند.

توجه: این الگوریتم، (First In First Out) FIFO نیز نامیده می‌شود.

توجه: FCFS یک الگوریتم انحصاری (Non Preemptive) است.

توجه: الگوریتم FIFO مشکل قحطی زدگی ندارد.

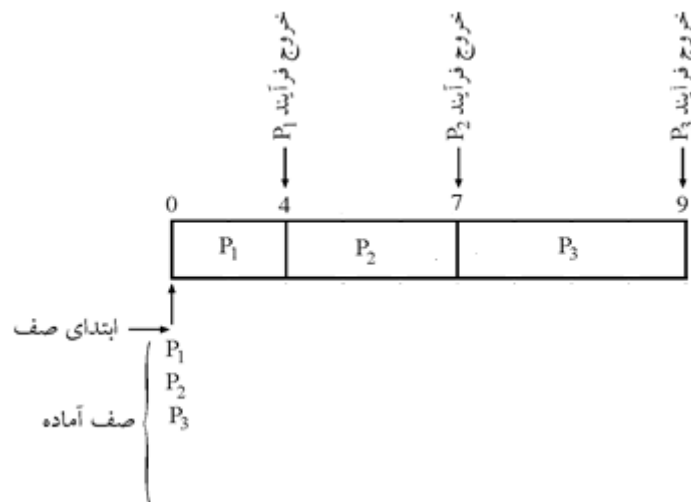
سوال: سه فرآیند P_1 ، P_2 و P_3 را در نظر بگیرید که به ترتیب از P_1 تا P_3 وارد شده‌اند (همگی در لحظه 0 و با اختلاف اندک وارد شده‌اند). با توجه به جدول زیر، زمان بازگشت و انتظار برای هر فرآیند چقدر است؟

فرآیند	مدت زمان اجرا (ms)
P_1	4
P_2	3
P_3	2

حل: با توجه به مفروضات مطرح شده در صورت سؤال داریم:

فرآیند	زمان ورود	زمان اجرا	زمان انتظار +	زمان بازگشت =
P_1	0	4		
P_2	0	3		
P_3	0	2		

فرآیندها به ترتیب P_1 ، P_2 و P_3 وارد شده‌اند، بنابراین نمودار گانت به صورت زیر است:



به نمودار فوق، نمودار گانت (Gantt Chart) گویند.

با توجه به نمودار فوق، فرآیند P_1 به محض ورود، پردازنده را در اختیار می‌گیرد، بنابراین زمان انتظار آن برابر صفر است. فرآیند P_2 ، ۴ میلی‌ثانیه و فرآیند P_3 ، ۷ میلی‌ثانیه منتظر می‌ماند.

زمان ورود فرآیند - زمان خروج فرآیند = زمان بازگشت فرآیند

$$P_1 \text{ زمان بازگشت} = 4 - 0 = 4$$

$$P_2 \text{ زمان بازگشت} = 7 - 0 = 7$$

$$P_3 \text{ زمان بازگشت} = 9 - 0 = 9$$

زمان اجرای فرآیند - زمان بازگشت فرآیند = زمان انتظار فرآیند

$$P_1 \text{ زمان انتظار} = 4 - 4 = 0$$

$$P_2 \text{ زمان انتظار} = 7 - 3 = 4$$

$$P_3 \text{ زمان انتظار} = 9 - 2 = 7$$

با توجه به اطلاعات به دست آمده، جدول قبل، به شکل زیر تکمیل می‌گردد:

فرآیند	زمان ورود	زمان اجرا	زمان انتظار +	زمان بازگشت =
P_1	0	4	0	4
P_2	0	3	4	7
P_3	0	2	7	9

در صورت سوال گفته شده است که

فرض کنید سه بسته از سه اینترفیس مختلف هم زمان وارد یک مسیریاب از نوع دیتاگرام می‌شوند. مسیریاب عمل مسیریابی برای بسته‌ها را به صورت موازی (هم زمان) انجام می‌دهد و با مشخص شدن اینترفیس خروجی، بسته را برای سویچینگ در اختیار سویچ درون مسیریاب قرار می‌دهد. فرض کنید سویچ درون مسیریاب از نوع bus است و زمان سویچینگ برای هر بسته ۱ میکروثانیه است.
چنانچه:

- هنگام ورود این سه بسته مسیریاب خالی باشد.
- مسیریاب هر سه بسته را به یک اینترفیس خروجی یکسان بفرستد.
- زمان ارسال (transmission) هر بسته از اینترفیس خروجی ۷ میکروثانیه باشد. (۷ میکروثانیه طول می‌کشد تا یک بسته از اینترفیس خروجی ارسال شود).
- هر بسته چه مدت در صف اینترفیس خروجی معطل می‌ماند؟

همانطور که گفتیم جهت درک بهتر و واضح‌تر این سوال، این سوال رو به یک پرسش سیستم عاملی تبدیل می‌کنیم و به همان سبک و سیاق حل خواهیم کرد. فرم تبدیل شده پرسش مطرح شده در صورت سوال، به فرم سیستم عاملی به صورت زیر است:

سوال: سه فرآیند P_1 ، P_2 و P_3 را در نظر بگیرید که به ترتیب از P_1 تا P_3 وارد صف آماده (Ready) شده‌اند (فرآیند P_1 پس از یک میکروثانیه تاخیر در لحظه ۱، فرآیند P_2 پس از دو میکروثانیه تاخیر در لحظه ۲ و فرآیند P_3 پس از سه میکروثانیه تاخیر در لحظه ۳ وارد شده‌اند). با توجه به جدول زیر، زمان بازگشت و انتظار برای هر فرآیند چقدر است؟

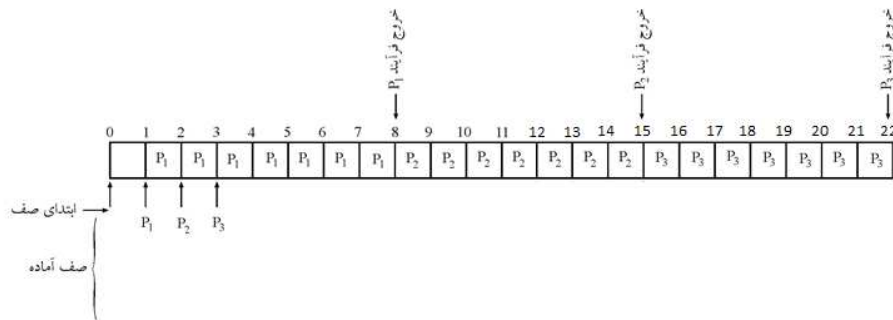
مدت زمان اجرا (میکروثانیه)	فرآیند
----------------------------	--------

P ₁	7
P ₂	7
P ₃	7

حل: با توجه به مفروضات مطرح شده در صورت سؤال داریم:

فرآیند	زمان ورود	زمان اجرا	زمان انتظار+	زمان بازگشت =
P ₁	1	7		
P ₂	2	7		
P ₃	3	7		

فرآیندها به ترتیب P₁، P₂ و P₃ وارد شده‌اند، بنابراین نمودار گانت به صورت زیر است:



زمان ورود فرآیند - زمان خروج فرآیند = زمان بازگشت فرآیند

$$P_1 \text{ زمان بازگشت} = 8 - 1 = 7$$

$$P_2 \text{ زمان بازگشت} = 15 - 2 = 13$$

$$P_3 \text{ زمان بازگشت} = 22 - 3 = 19$$

$$\text{میانگین زمان بازگشت} = \text{ATT} = \frac{7 + 13 + 19}{3} = \frac{39}{3} = 13$$

زمان اجرای فرآیند - زمان بازگشت فرآیند = زمان انتظار فرآیند

$$P_1 \text{ زمان انتظار} = 7 - 7 = 0$$

$$P_2 \text{ زمان انتظار} = 13 - 7 = 6$$

$$P_3 \text{ زمان انتظار} = 19 - 7 = 12$$

$$\text{میانگین زمان انتظار} = \text{AWT} = \frac{0+6+12}{3} = \frac{18}{3} = 6$$

$$\text{میانگین زمان اجرا} = \text{AST} = \frac{7+7+7}{3} = \frac{21}{3} = 7$$

AVG Turnaround Time=AVG Service Time + AVG Waiting Time

میانگین زمان انتظار + میانگین زمان اجرا = میانگین زمان بازگشت

$$13 = 7 + 6$$

توجه: مطابق رابطه فوق، تفاضل میانگین زمان بازگشت و میانگین زمان انتظار باید برابر میانگین زمان اجرا باشد.

توجه: همچنین مطابق رابطه فوق، میانگین زمان بازگشت همواره از میانگین زمان انتظار بیشتر است.

با توجه به اطلاعات به دست آمده، جدول قبل، به شکل زیر تکمیل می‌گردد:

فرآیند	زمان ورود	زمان اجرا	زمان انتظار+	زمان بازگشت =
P ₁	1	7	0	7
P ₂	2	7	6	13
P ₃	3	7	12	19

$$\text{میانگین زمان بازگشت} = \text{میانگین زمان انتظار} + \text{میانگین زمان اجرا}$$

$$13 = 6 + 7$$

توجه: حال بازگردیم به همان مفاهیم درس شبکه‌های کامپیوتری، اگر در محاسبات تاخیر و انتظار بسته‌ها همه چیز را از لحظه‌ی ورود به صف خروجی مسیریاب در نظر بگیریم، آنگاه این صف خروجی مسیریاب همانند صف آماده (Ready) فرآیندها در سیستم عامل آغازگر محاسبات فوق العاده جذابی خواهد شد. واضح و مشخص است که الگوریتم پردازش بسته‌های صف خروجی مسیریاب FIFO است، بنابراین در شبکه‌های کامپیوتری هم، زمان بازگشت، زمان انتظار و زمان اجرا معنا خواهد داشت.

در صورت سوال گفته شده است که

فرض کنید سه بسته از سه اینترنتی مختلف هم زمان وارد یک مسیریاب از نوع دیتاگرام می‌شوند. مسیریاب عمل مسیریابی برای بسته‌ها را به صورت موازی (هم زمان) انجام می‌دهد و با مشخص شدن اینترنتی خروجی، بسته را برای سویچینگ در اختیار سویچ درون مسیریاب قرار می‌دهد. فرض کنید سویچ درون مسیریاب از نوع bus است و زمان سویچینگ برای هر بسته 1 میکروثانیه است.

چنانچه:

- هنگام ورود این سه بسته مسیریاب خالی باشد.
 - مسیریاب هر سه بسته را به یک اینترنتی خروجی یکسان بفرستد.
 - زمان ارسال (transmission) هر بسته از اینترنتی خروجی 7 میکروثانیه باشد. (7 میکروثانیه طول می‌کشد تا یک بسته از اینترنتی خروجی ارسال شود).
- هر بسته چه مدت در صف اینترنتی خروجی معطل می‌ماند؟

توجه: همانطور که گفتیم جهت درک بهتر و واضح‌تر این سوال، این سوال رو به یک پرسش سیستم عاملی تبدیل می‌کنیم و به همان سبک و سیاق حل خواهیم کرد. فرم تبدیل شده پرسش مطرح شده در صورت سوال را به فرم سیستم عاملی بررسی کردیم، حال در ادامه به همان فرم شبکه‌های کامپیوتری مساله را بررسی می‌کنیم، به صورت زیر:

سوال: سه بسته P_1 ، P_2 و P_3 را در نظر بگیرید که به ترتیب از P_1 تا P_3 وارد صف آماده خروجی مسیریاب (Ready) شده‌اند (بسته P_1 پس از یک میکروثانیه تاخیر در لحظه 1، بسته P_2 پس از دو میکروثانیه تاخیر در لحظه 2 و بسته P_3 پس از سه میکروثانیه تاخیر در لحظه 3 وارد شده‌اند). با توجه به جدول زیر، زمان بازگشت و انتظار برای هر فرآیند چقدر است؟

توجه: بسته‌های ورودی مسیریاب پس از عمل مسیریابی و سویچینگ (سویچ درون مسیریاب یا فابریک سویچ)، وارد صف خروجی مسیریاب می‌شوند، در صورت سوال مطرح شده است که سویچ درون مسیریاب از نوع bus است و زمان سویچینگ برای هر بسته 1 میکروثانیه است. بنابراین مطابق خاصیت bus در هر لحظه فقط یک بسته می‌تواند جهت سویچینگ داخل bus قرار گیرد و امکان سویچ همزمان همه بسته‌ها باهم وجود ندارد، پس بسته‌ها یکی یکی وارد فابریک سویچ با فرم bus می‌شوند و یکی یکی هم از فابریک سویچ خارج می‌شوند و به تبع این فرم یکی یکی هم وارد صف خروجی مسیریاب می‌شوند. البته مطابق فرض سوال که گفته شده است، مسیریاب هر سه بسته را به یک اینترنتی خروجی یکسان بفرستد یعنی همان صف آماده خروجی مسیریاب (Ready)، اگر بخواهیم جملات فوق را در یک جمله بیان کنیم، اینطور خواهد بود، بسته P_1 در لحظه 0 وارد فابریک سویچ bus می‌شود و پس از 1 میکروثانیه عمل سویچینگ یعنی لحظه 1 میکروثانیه داخل اینترنتی خروجی یعنی همان صف آماده خروجی مسیریاب (Ready) قرار می‌گیرد، در ادامه نیز بسته P_2 در لحظه 1 وارد فابریک سویچ bus می‌شود و پس از 1 میکروثانیه عمل سویچینگ یعنی لحظه 2 میکروثانیه داخل اینترنتی خروجی یعنی همان صف آماده خروجی مسیریاب (Ready) قرار می‌گیرد و در نهایت بسته P_3 در لحظه 2 وارد فابریک سویچ bus می‌شود و پس از 1 میکروثانیه عمل سویچینگ یعنی لحظه 3 میکروثانیه داخل اینترنتی خروجی یعنی همان صف آماده خروجی مسیریاب (Ready) قرار می‌گیرد.

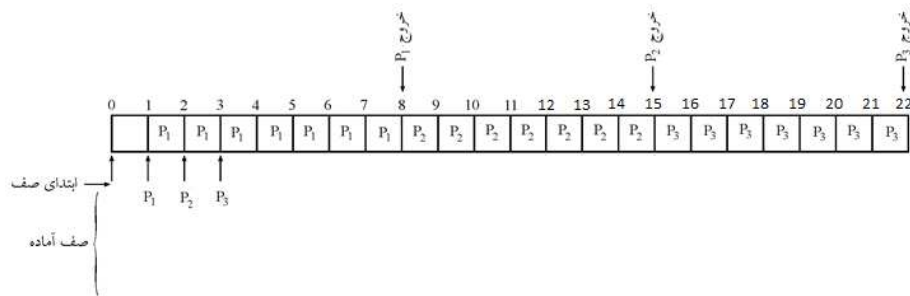
توجه: مسیریاب از نوع دیتاگرام یعنی اینکه فرم مسیریابی از نوع سویچینگ بسته‌ای (packet switching) است و از انواع دیگر همچون سویچینگ مدار مجازی (virtual circuit switching) نیست.

بسته	مدت زمان انتقال (میکروثانیه)
P ₁	7
P ₂	7
P ₃	7

حل: با توجه به مفروضات مطرح شده در صورت سؤال داریم:

بسته	زمان ورود	زمان انتقال	زمان انتظار +	زمان بازگشت =
P ₁	1	7		
P ₂	2	7		
P ₃	3	7		

بسته‌ها به ترتیب P₁، P₂ و P₃ وارد شده‌اند، بنابراین نمودار گانت به صورت زیر است:



زمان ورود بسته - زمان خروج بسته = زمان بازگشت

$$P_1 \text{ زمان بازگشت} = 8 - 1 = 7$$

$$P_2 \text{ زمان بازگشت} = 15 - 2 = 13$$

$$P_3 \text{ زمان بازگشت} = 22 - 3 = 19$$

$$\text{میانگین زمان بازگشت} = \text{ATT} = \frac{7 + 13 + 19}{3} = \frac{39}{3} = 13$$

زمان انتقال بسته - زمان بازگشت بسته = زمان انتظار بسته

$$P_1 \text{ زمان انتظار} = 7 - 7 = 0$$

$$P_2 \text{ زمان انتظار} = 13 - 7 = 6$$

$$P_3 \text{ زمان انتظار} = 19 - 7 = 12$$

$$\text{میانگین زمان انتظار} = AWT = \frac{0+6+12}{3} = \frac{18}{3} = 6$$

$$\text{میانگین زمان اجرا} = AST = \frac{7+7+7}{3} = \frac{21}{3} = 7$$

$$AVG \text{ Turnaround Time} = AVG \text{ Service Time} + AVG \text{ Waiting Time}$$

میانگین زمان انتظار + میانگین زمان اجرا = میانگین زمان بازگشت

$$13 = 7 + 6$$

توجه: مطابق رابطه فوق، تفاضل میانگین زمان بازگشت و میانگین زمان انتظار باید برابر میانگین زمان اجرا باشد.

توجه: همچنین مطابق رابطه فوق، میانگین زمان بازگشت همواره از میانگین زمان انتظار بیشتر است.

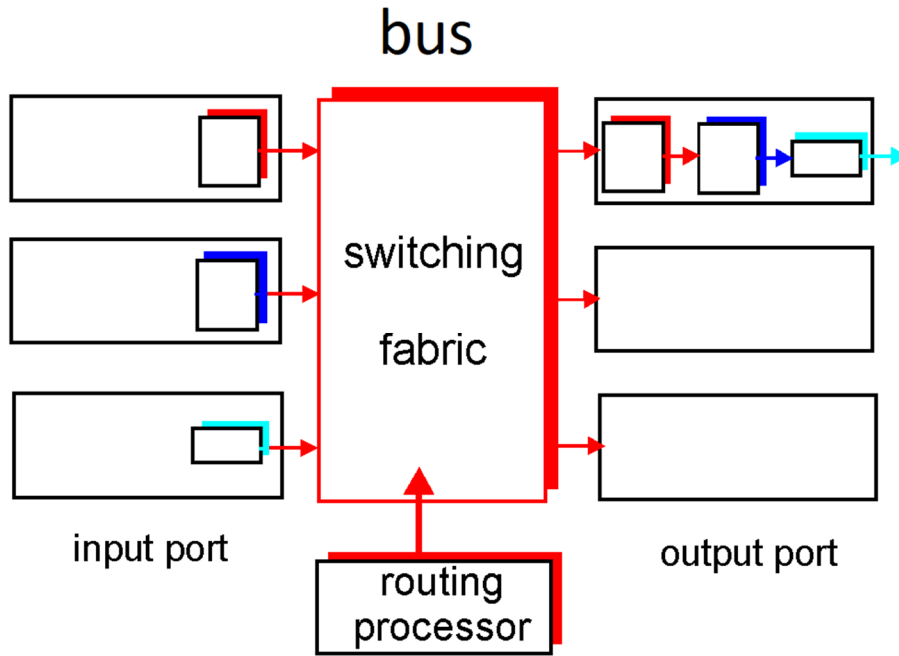
با توجه به اطلاعات به دست آمده، جدول قبل، به شکل زیر تکمیل می گردد:

بسته	زمان ورود	زمان انتقال	زمان انتظار +	زمان بازگشت =
P ₁	1	7	0	7
P ₂	2	7	6	13
P ₃	3	7	12	19

$$\text{میانگین زمان بازگشت} = \text{میانگین زمان انتظار} + \text{میانگین زمان اجرا}$$

$$13 = 6 + 7$$

شکل زیر گویای مطلب است:



بنابراین واضح و مشخص است، که گزینه‌ی سوم پاسخ سوال است.

تست‌های فصل پنجم

۵۶- در هسته شبکه ای شش مسیریاب که با A, B, C, D, E, F مشخص شده‌اند وجود دارد. جداول مسیریابی در مسیریاب‌های A و E به صورت زیر است. در هر جدول، سطر اول مقصد (Dest.) و سطر دوم شماره اینترفیس خروجی در مسیریاب را نشان می‌دهد. مثلاً اگر بسته‌ای به مسیریاب A برسد که مقصد آن مسیریاب B باشد از اینترفیس شماره ۱ ارسال می‌شود. مقادیر زیر هزینه برخی از لینک‌های بین این مسیریاب‌ها را نشان می‌دهد. مثلاً هزینه لینک بین مسیریاب A و مسیریاب B به مقدار ۲ واحد است. حداکثر مقدار برای $Cost(C,E) + Cost(E,D)$ کدام است؟

$$Cost(A,D) = ۵, Cost(B,C) = ۲, Cost(B,D) = ۳, Cost(C,D) = ۲, Cost(D,F) = ۵,$$

$$Cost(A,B) = ۲, Cost(A,C) = ۱$$

Forwarding Table of router A

Dest.	B	C	D	E	F
Out Port	1	2	2	2	2

Forwarding Table of router E

Dest.	A	B	C	D	F
Out Port	1	1	1	2	3

۳ (۱)

۴ (۲)

۵ (۳)

۶ (۴)

پاسخ‌های فصل پنجم

۵۶- گزینه (۱) صحیح است.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه اول را به عنوان پاسخ اعلام کرده بود. که متأسفانه پاسخ درست این سوال در گزینه‌ها نیست.

شرایط تزییر و تقارن:

$cost(A, B) = 2$
 $cost(A, C) = 1$
 $cost(A, D) = 5$

$cost(B, C) = 2$
 $cost(B, D) = 3$

$cost(C, D) = 2$
 $cost(C, E) = 4$

$cost(D, F) = 5$
 $cost(D, E) = 3$

(A)

Dest	B	C	D	E	F
out port	1	2	2	2	2

Port 1
A → B

Port 2
A → C, D, E, F

(E)

Dest	A	B	C	D	F
out port	1	1	1	2	3

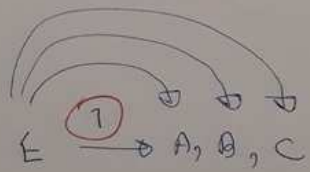
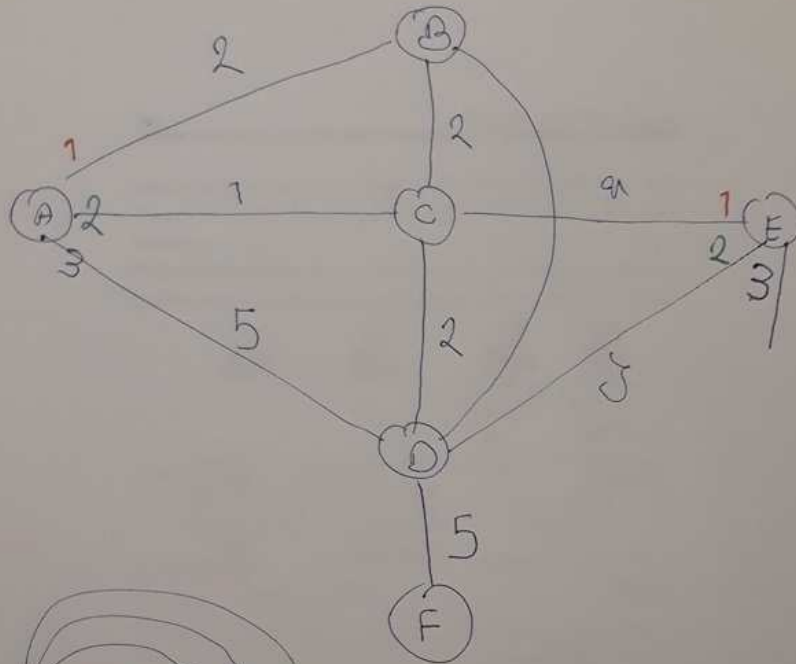
Port 1
E → A, B, C

Port 2
E → D

Port 3
E → F

2

پواسا با شروطت مطرح شده، صورت شکل زیر بدست می آید:



مطابق جدول (E) طرح:

نوع - یا - که به صورتی کنیم:

$a < y + 2$ ← $(a+1) < (y+2+1)$ (A) برای x

$a < y + 1$ ← $(a+2) < (y+3)$ (B) برای x

$a < y + 2$ (C) برای x

از سمت a

از سمت y

③

مسافتها: (E) ۸، (D) ۴

E (2) → D

که از حال می‌گیریم می‌کنیم:

از سمت C: $y < x + 2$

از سمت D: $x < y + 2$

$$\begin{aligned} x &< y + 2 \\ x &< y + 1 \\ x &< y + 2 \end{aligned}$$

$$y < x + 2$$

از موارد مذکور می‌توانیم مقدار x و y را بدیم:

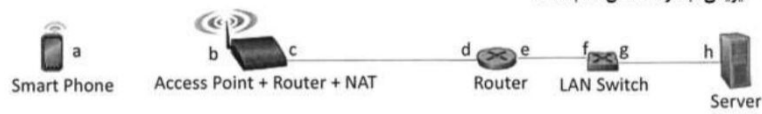
مسافتها: (C, E) ۲، (E, D) ۴

~~$Cost(C, E) + Cost(E, D)$~~

این سؤال باید حذف شود.

تست‌های فصل پنجم

۵۷- شبکه داده شده را در نظر بگیرید. یک دستگاه گوشی هوشمند که با **a** مشخص شده است، با استفاده از ارتباط **wifi** به یک **access point** متصل است. درون **access point** یک مسیریاب و یک **NAT (Network Address Translation)** وجود دارد. در شکل حروف **h, ... , b, a** را مشاهده می‌کنید. این حروف اینترفیس ادوات موجود در این شبکه است. بسته ای که از اینترفیس **a** عازم اینترفیس **h** است را در نظر بگیرید. این بسته هنگام خروج از اینترفیس **e** به ترتیب از راست به چپ آدرس فیزیکی مبدأ و مقصد آن کدام است؟



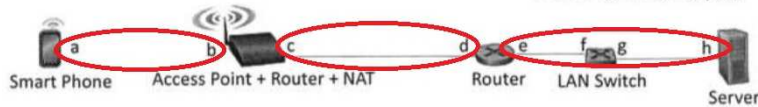
- ۱) آدرس فیزیکی اینترفیس **a** - آدرس فیزیکی اینترفیس **h**
- ۲) آدرس فیزیکی اینترفیس **e** - آدرس فیزیکی اینترفیس **h**
- ۳) آدرس فیزیکی اینترفیس **d** - آدرس فیزیکی اینترفیس **e**
- ۴) آدرس فیزیکی اینترفیس **c** - آدرس فیزیکی اینترفیس **h**

پاسخ‌های فصل پنجم

۵۷- گزینه (۲) صحیح است.

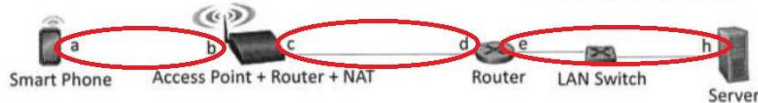
برای حرکت node to node (نقطه به نقطه یا گره به گره) یعنی حرکت محلی داخل یک زیر شبکه بین یک گره محلی با گره محلی دیگر نیاز به آدرس فیزیکی یعنی آدرس MAC است که این وظیفه و جا به جایی توسط لایه پیوند داده انجام می‌شود. اما برای حرکت host to host (میزبان ابتدایی و میزبان انتهایی یعنی مبدا اصلی و مقصد اصلی) یعنی حرکت سراسری داخل کل شبکه بین یک میزبان ابتدایی و میزبان انتهایی نیاز به آدرس منطقی یعنی آدرس IP است که این وظیفه و جا به جایی توسط لایه شبکه انجام می‌شود.

شبکه داده شده را در نظر بگیرید. یک دستگاه گوشی هوشمند که با **a** مشخص شده است، با استفاده از ارتباط **wifi** به یک **access point** متصل است. درون **access point** یک مسیریاب و یک **NAT (Network Address Translation)** وجود دارد. در شکل حروف **a, b, ... , h** را مشاهده می‌کنید. این حروف اینترفیس ادوات موجود در این شبکه است. بسته ای که از اینترفیس **a** عازم اینترفیس **h** است را در نظر بگیرید. این بسته هنگام خروج از اینترفیس **e** به ترتیب از راست به چپ آدرس فیزیکی مبدأ و مقصد آن کدام است؟



در زیر شبکه بین **e** و **h** دستگاه LAN Switch ایجاد زیر شبکه نمی‌کند چون LAN Switch فقط لایه فیزیکی و لایه پیوند داده دارد و لایه شبکه ندارد پس IP هم متوجه نمی‌شود پس زیر شبکه هم ایجاد نمی‌کند. در یک زیر شبکه پورت‌های LAN Switch آدرس ندارند، پس **f** و **g** معنی ندارند و حذف می‌شوند، به صورت زیر:

شبکه داده شده را در نظر بگیرید. یک دستگاه گوشی هوشمند که با **a** مشخص شده است، با استفاده از ارتباط **wifi** به یک **access point** متصل است. درون **access point** یک مسیریاب و یک **NAT (Network Address Translation)** وجود دارد. در شکل حروف **a, b, ... , h** را مشاهده می‌کنید. این حروف اینترفیس ادوات موجود در این شبکه است. بسته ای که از اینترفیس **a** عازم اینترفیس **h** است را در نظر بگیرید. این بسته هنگام خروج از اینترفیس **e** به ترتیب از راست به چپ آدرس فیزیکی مبدأ و مقصد آن کدام است؟



فریم موجود در پورت **e**، آدرس فیزیکی مبدأ آن **e** و آدرس فیزیکی مقصد آن **h** درج می‌شود. آدرس MAC یعنی آدرس فیزیکی توسط لایه پیوند داده درج می‌شود، درج آدرس MAC از وظایف لایه پیوند داده است.

دقت کنید که بسته یعنی Packet تولید شده توسط لایه شبکه در ایستگاه **a**، آدرس منطقی میزبان ابتدایی یعنی آدرس IP میزبان ابتدایی آن برای مثال **X** و آدرس منطقی میزبان انتهایی یعنی آدرس

IP میزبان انتهایی آن Y درج می‌گردد و این آدرس دهی ثابت و ایستا هست و تغییر نمی‌کند تا اینکه packet به ایستگاه نهایی Y برسد که این آدرس دهی برای حرکت سراسری و کل شبکه تنظیم شده است، دقت کنید که آدرس منطقی ایستگاه a آدرس X و آدرس منطقی ایستگاه h آدرس Y لحاظ شده است. اما در حرکت محلی زیر شبکه به زیر شبکه آدرس‌های فیزیکی یعنی آدرس MAC متغیر و پویا است و از یک زیر شبکه به زیر شبکه بعدی تغییر می‌کند. به صورت زیر:

فریم موجود در پورت a، آدرس فیزیکی مبدا آن a و آدرس فیزیکی مقصد آن b درج می‌شود.

در پورت b لایه شبکه توسط آدرس منطقی میزبان انتهایی Y متوجه می‌شود که packet مورد نظر باید در پورت c قرار گیرد. packet موجود در لایه شبکه جهت ادامه مسیر تحویل لایه پیوند داده می‌شود تا به Frame تبدیل گردد.

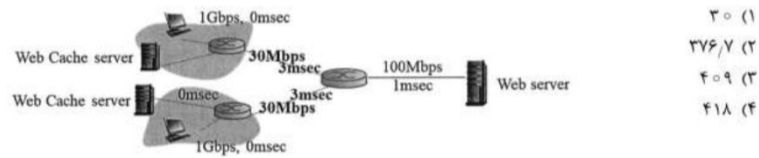
فریم موجود در پورت c، آدرس فیزیکی مبدا آن c و آدرس فیزیکی مقصد آن d درج می‌شود.

در پورت d لایه شبکه توسط آدرس منطقی میزبان انتهایی Y متوجه می‌شود که packet مورد نظر باید در پورت e قرار گیرد. packet موجود در لایه شبکه جهت ادامه مسیر تحویل لایه پیوند داده می‌شود تا به Frame تبدیل گردد.

فریم موجود در پورت e، آدرس فیزیکی مبدا آن e و آدرس فیزیکی مقصد آن h درج می‌شود.

تست‌های فصل پنجم

۵۸- در شبکه زیر وب‌سروری با لینکی به ظرفیت ۱۰۰ مگابیت بر ثانیه به مسیریابی متصل است. این مسیریاب با دو لینک مستقل هر یک به ظرفیت ۳۰ مگابیت بر ثانیه به دو مسیریاب مستقل اتصال دارد. هر یک از این دو مسیریاب به یک شبکه محلی متصل هستند. در هر شبکه محلی یک کامپیوتر کلاینت و یک گش‌سرور وجود دارد. زمان انتشار برای هر لینک روی شکل مشخص شده است. زمان انتشار در شبکه محلی ناچیز است. کلاینت‌ها درخواست‌های خود را به گش‌سرور شبکه خود می‌دهند. اگر نرخ اثابت (hit rate) در گش‌سرورها ۴۰ درصد باشد، هر کلاینت به‌طور متوسط حداکثر با نرخ چند مگابیت بر ثانیه دریافت می‌کند؟



پاسخ‌های فصل پنجم

۵۸- گزینه (۴) صحیح است.

مثال: از شما می‌پرسند معدل کل شما در 6 ترم گذشته چند می‌شود؟
معدل این ترم 16 و میانگین 5 ترم گذشته 15.5

معدل پدیده‌ی لحظه‌ای + تخمین قدیم = تخمین جدید

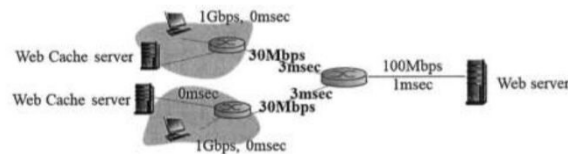
$$\text{معدل کل} = \frac{5}{6} \times 15.5 + \frac{1}{6} \times 16$$

چون اهمیت ترم ششم، $\frac{1}{6}$ است، بنابراین به این ترم نسبت به 5 ترم گذشته وزن می‌دهیم.

یعنی به پدیده‌ی قدیمی وزن $\frac{5}{6}$ و به پدیده‌ی جدید وزن $\frac{1}{6}$ را می‌دهیم.

در صورت سوال گفته شده است که

در شبکه زیر وب‌سروری با لینکی به ظرفیت 100 مگابیت بر ثانیه به مسیریابی متصل است. این مسیریاب با دو لینک مستقل هر یک به ظرفیت 30 مگابیت بر ثانیه به دو مسیریاب مستقل اتصال دارد. هر یک از این دو مسیریاب به یک شبکه محلی متصل هستند. در هر شبکه محلی یک کامپیوتر کلاینت و یک گش‌سرور وجود دارد. زمان انتشار برای هر لینک روی شکل مشخص شده است. زمان انتشار در شبکه محلی ناچیز است. کلاینت‌ها درخواست‌های خود را به گش‌سرور شبکه خود می‌دهند. اگر نرخ اثابت (hit rate) در گش‌سرورها 40 درصد باشد، هر کلاینت به‌طور متوسط حداکثر با نرخ چند مگابیت بر ثانیه دریافت می‌کند؟



داده‌های مسئله به صورت زیر است:

$$R_{\text{LAN-Server}} = 1 \text{ Gbps} = 1 \times 10^3 \text{ Mbps} = 1000 \text{ Mbps}$$

$$R_1 = 30 \text{ Mbps}$$

$$R_{\text{WAN-Server}} = 100 \text{ Mbps}$$

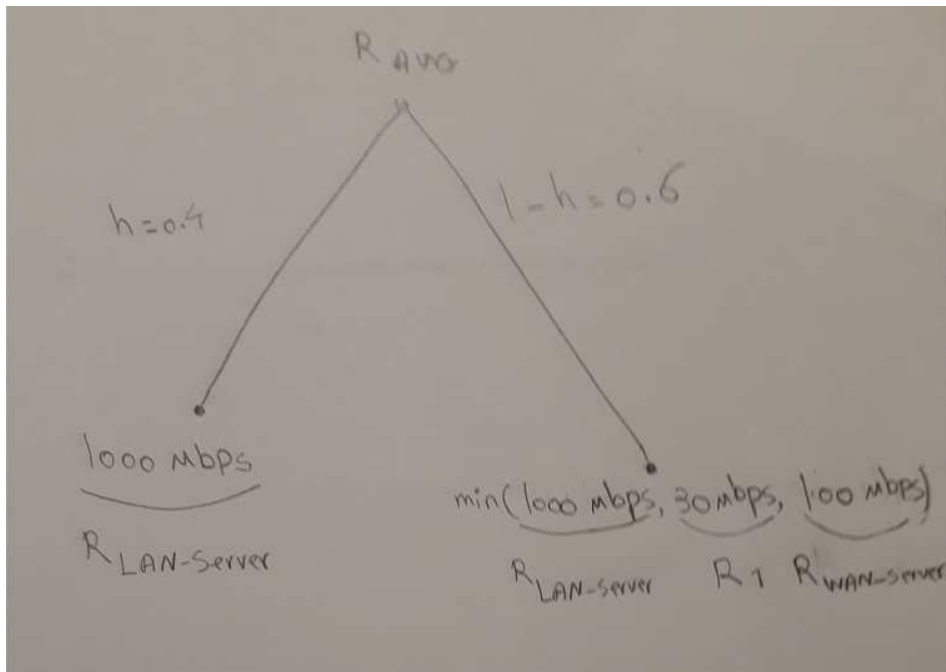
$$h = 0.4, 1-h=0.6$$

توجه: در صورت سوال مطرح شده است که یک Web Server داریم درون یک WAN که ظرفیت این Web Server برابر مقدار 100 Mbps است که متصل شده است به یک مسیریاب، و مسیریاب هم به دو مسیر 30 Mbps متصل شده است، که در نهایت هر مسیر وارد یک LAN می‌شود، یعنی LAN1 و LAN2، هر LAN یک Client و یک Web Cache Server دارد، همچنین زمان انتشار در LAN مطابق فرض سوال ناچیز در نظر گرفته شده است. یعنی T_{prop} برابر مقدار

صفر در نظر گرفته شده است. در شبکه‌ی محلی یا LAN یعنی در سطح لایه پیوند داده، هر RTT برابر دوبرابر T_{prop} است، بنابراین مقدار RTT نیز در شبکه محلی برابر صفر در نظر گرفته می‌شود، پس در شبکه محلی RTT نداریم و فقط زمان‌های انتقال و نرخ انتقال یا گذردهی مورد بررسی قرار می‌گیرد. همچنین در صورت سوال گفته شده است که در هر LAN موجود، هر Client درخواست‌های خودش را ابتدا به Web Cache Server محلی خود می‌دهد و نرخ اجابت یا همان نرخ برخورد یا همان نرخ اصابت هم برابر 40 درصد یعنی 0.4 است. یعنی 40 درصد از پیام‌هایی که Client صادر می‌کند، پاسخ آن داخل Web Cache Server محلی و مرتبط خودش است و به تبع 60 درصد از پیام‌هایی که Client صادر می‌کند، پاسخ آن داخل Web Cache Server محلی و مرتبط خودش نیست و باید پاسخ آن داخل Web Server موجود در WAN جستجو شود. انگار که در ابتدا در یک فضای کوچک مثل یک LAN به دنبال خواسته‌هایمان می‌گردیم و اگر نبود در یک فضای بزرگتر مثل یک WAN به دنبال خواسته‌هایمان می‌گردیم.

در نهایت در صورت سوال این پرسش مطرح شده است که متوسط گذردهی یا همان حداکثر نرخ انتقال برای هر Client چقدر است؟

بنابراین برای محاسبه‌ی میانگین نرخ انتقال چون بحث احتمال مطرح است، باید میانگین دو فرم 40 درصد و 60 درصد را مورد محاسبه قرار دهیم.
برای محاسبه R_{AVG} درخت زیر را در نظر بگیرید:



رابطه درخت فوق به صورت زیر خواهد بود:

$$R_{AVG} = [h \times R_{LAN-Server}] + [(1-h) \times \min(R_{LAN-Server}, R_1, R_{WAN-Server})]$$

یعنی احتمال شاخه ضربدر برگ بعلاوه احتمال شاخه ضربدر برگ.

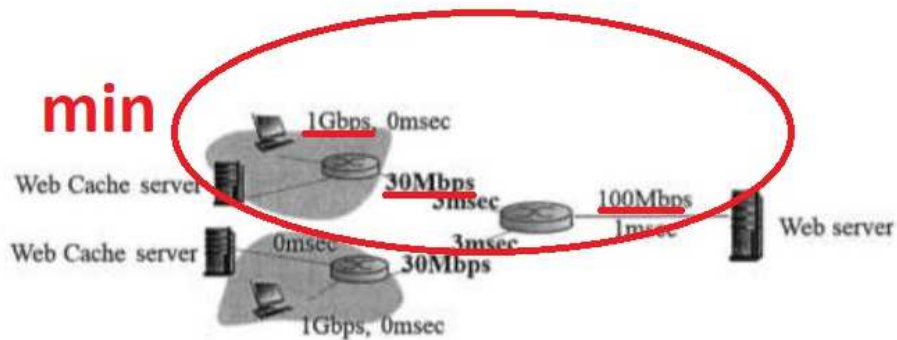
پس از جایگذاری اولیه رابطه زیر را خواهیم داشت:

$$R_{AVG} = [0.4 \times 1000] + [(0.6) \times \min(1000, 30, 100)]$$

پس از جایگذاری نهایی رابطه زیر را خواهیم داشت:

$$R_{AVG} = [0.4 \times 1000] + [0.6 \times 30] = 400 + 18 = 418 \text{ Mbps}$$

توجه: همانطور که گفتیم که در هر LAN موجود، هر Client درخواست‌های خودش را ابتدا به Web Cache Server محلی خود می‌دهد و نرخ اجابت یا همان نرخ برخورد یا همان نرخ اصابت هم برابر 40 درصد یعنی 0.4 است. یعنی 40 درصد از پیام‌هایی که Client صادر می‌کند، پاسخ آن داخل Web Cache Server محلی و مرتبط خودش است و به تبع 60 درصد از پیام‌هایی که Client صادر می‌کند، پاسخ آن داخل Web Cache Server محلی و مرتبط خودش نیست و باید پاسخ آن داخل Web Server موجود در WAN جستجو شود. در این جستجو و حرکت به سمت Web Server مسیر زیر باید طی شود. بنابراین نرخ انتقال Web Server می‌شود مینیمم تمام نرخ انتقال‌های موجود در مسیر، چون مطابق اصل گلوگاه شبکه یا Network Bottleneck بیشتر از این مینیمم امکان انتقال داده در واحد زمان وجود ندارد.



شکل‌های زیر گویای مطلب هستند:



تست‌های فصل ششم

۵۹- فرض کنید که بین یک سرور و کلاینت یک ارتباط TCP برقرار شده و کلاینت درخواست فایلی را به سرور می‌فرستد و سپس سرور فایل را برای کلاینت ارسال می‌کند. سرور فایل را به صورت ۲۰ بسته ۵۰۰ بایتی ارسال می‌کند. زمان رفت و برگشت بین کلاینت و سرور $RTT = 60 \text{ msec}$ است. پهنای باند ارسال سرور ۴۰۰,۰۰۰ بیت بر ثانیه است. از زمانی که کلاینت درخواست خود را ارسال می‌کند تا وقتی فایل را کاملاً دریافت می‌کند چند ثانیه طول می‌کشد؟ (TCP Reno فعال بوده و هیچ بسته‌ای دچار مشکل نمی‌شود.)

○/۴ (۱)

○/۳۶ (۲)

○/۲۲۴ (۳)

○/۲ (۴)

پاسخ‌های فصل ششم

۵۹- گزینه (۱) صحیح است.

توجه: در لایه انتقال، پروتکل TCP وظیفه‌ی کنترل ازدحام را بر عهده دارد که الگوریتم‌های این وظیفه به فرم‌هایی نظیر TCP TAHOE و TCP RENO وجود دارد، با توجه به صورت سوال، الگوریتم کنترل ازدحام ارتباط TCP مطرح شده به صورت TCP RENO است.

توجه: اتصال 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 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

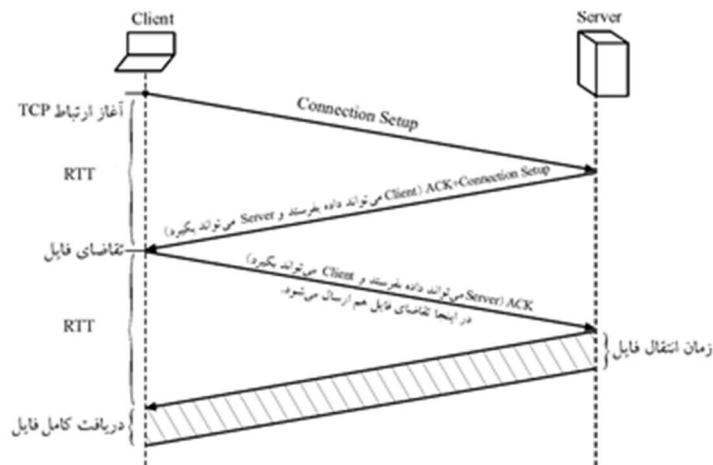
این سه پیغام در TCP اصطلاحاً 3-way handshaking گفته می‌شود. به معنی دست‌تکان‌دهی سه طرفه، در واقع با این کار، دو گره دارند عمل خوشامدگویی انجام می‌دهند و سپس Connection به شکل دو طرفه برقرار می‌شود.

مثال: مثلاً شما وقتی 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 مورد نظر 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: 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 مورد نظر از سمت Client به Server می‌رسد. شکل زیر گویای مطلب می‌باشد:



در صورت سوال گفته شده است که

فرض کنید که بین یک سرور و کلاینت یک ارتباط TCP برقرار شده و کلاینت درخواست فایلی را به سرور می‌فرستد و سپس سرور فایل را برای کلاینت ارسال می‌کند. سرور فایل را به صورت ۲۰ بسته ۵۰۰ بایتی ارسال می‌کند. زمان رفت و برگشت بین کلاینت و سرور $RTT = 60 \text{ msec}$ است. پهنای باند ارسال سرور ۴۰۰,۰۰۰ بیت بر ثانیه است. از زمانی که کلاینت درخواست خود را ارسال می‌کند تا وقتی فایل را کاملاً دریافت می‌کند چند ثانیه طول می‌کشد؟ (TCP Reno فعال بوده و هیچ بسته‌ای دچار مشکل نمی‌شود).

توجه: در صورت سوال فرض شده است که بین client و Server یک ارتباط TCP برقرار شده است.

داده‌های مسئله به صورت زیر است:

$$L_{\text{File}} = 500 \text{ Byte}$$

$$R = 4 \times 10^5 \text{ bps}, \text{ RTT} = 60 \text{ ms}$$

$$\text{Cardinality}(\text{File}) = 20$$

توجه: در صورت سوال مواردی همچون فاز کنونی TCP Reno مثل Slow Start، Congestion Avoidance و Fast Recovery، مقدار اولیه، مقدار MSS و مقدار نقطه آستانه Threshold به وضوح و دقت بیان نشده است. بنابراین فرض می‌کنیم که به طور روال معمول شروع از همان Slow Start بوده است و به تبع چون شروع از Slow Start است، پس مقدار اولیه نیز همان یک خواهد بود. همچنین فرض می‌کنیم اندازه‌ی هر بسته مطرح شده در صورت سوال برابر یک MSS باشد، یعنی هر بسته در یک MSS ارسال شود. مقدار نقطه آستانه Threshold یا ssthresh هم بدون سقف و نامحدود در نظر می‌گیریم. همچنین در صورت سوال مطرح شده است که هیچ بسته‌ای دچار مشکل و مساله نمی‌شود یعنی loss نمی‌شود، بنابراین این نتیجه حاصل می‌شود که همه‌ی ACKها به فرستنده رسیده‌اند.

با توجه به مفروضات در نظر گرفته شده، جدول زیر را خواهیم داشت:

سگمنت	RTT (دور)	شماره بسته‌ها	کل
1	1	1	1
2	2	2, 3	3
4	3	4, 5, 6, 7	7
8	4	8, 9, 10, 11, 12, 13, 14, 15	15
16	5	16, 17, 18, 19, 20	20

slow start
(توجه داشته باشید)

در دور اول فرستنده 1 سگمنت داده ارسال می‌کند یعنی بسته شماره 1 و گیرنده نیز 1 سگمنت ACK تولید می‌کند که به دست فرستنده می‌رسد.

توجه: دقت کنید که در صورت سوال مطرح شده است که هیچ بسته‌ای دچار مشکل و مساله نمی‌شود یعنی loss نمی‌شود، بنابراین این نتیجه حاصل می‌شود که تمامی ACKها به فرستنده می‌رسند.

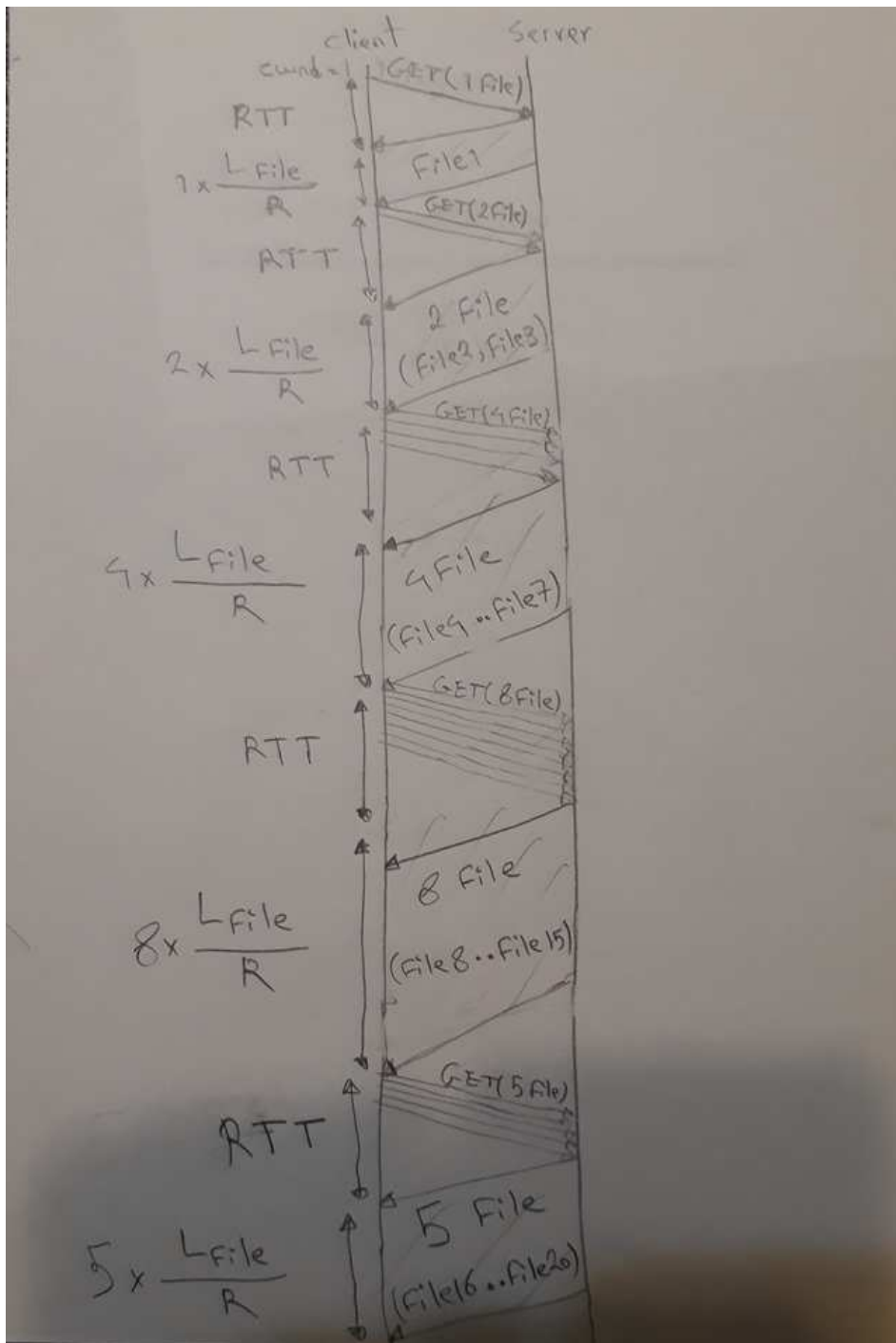
در دور دوم فرستنده 2 سگمنت داده ارسال می‌کند یعنی بسته‌های شماره 2 و 3 و گیرنده نیز 2 سگمنت ACK تولید می‌کند که به دست فرستنده می‌رسد.

در دور سوم فرستنده 4 سگمنت داده ارسال می‌کند یعنی بسته‌های شماره 4، 5، 6 و 7 و گیرنده نیز 4 سگمنت ACK تولید می‌کند که به دست فرستنده می‌رسد.

در دور چهارم فرستنده 8 سگمنت داده ارسال می‌کند یعنی بسته‌های شماره 8، 9، 10، 11، 12، 13، 14 و 15 و گیرنده نیز 8 سگمنت ACK تولید می‌کند که به دست فرستنده می‌رسد.

در دور پنجم با توجه به بسته‌های باقی‌مانده فرستنده 5 سگمنت داده ارسال می‌کند یعنی بسته‌های شماره 16، 17، 18، 19 و 20 و گیرنده نیز 5 سگمنت ACK تولید می‌کند که به دست فرستنده می‌رسد. هرچند که با توجه به ظرفیت cwnd توانایی ارسال 16 سگمنت داده نیز بود. نتیجه اینکه در حالت کلی 20 سگمنت داده توسط فرستنده ارسال شده است و 20 سگمنت ACK نیز توسط گیرنده دریافت شده است.

شکل زیر نیز گویای مطلب است:



بر اساس شکل فوق رابطه‌ی زیر برقرار است:

$$T_{\text{Total}} = \left[(5 \times \text{RTT}) \right] + \left[20 \times (T_{\text{File}}) \right]$$

T_{File} از رابطه زیر بدست می‌آید:

$$T_{\text{File}} = \frac{L_{\text{File}}}{R}$$

T_{File} ، زمان انتقال File به داخل کانال انتقال است.

که L_{File} برابر اندازه File و R برابر نرخ انتقال کانال می‌باشد.

که پس از جایگذاری اولیه رابطه زیر را خواهیم داشت:

$$T_{\text{Total}} = \left[(5 \times 60) \right] + \left[20 \times \left(\frac{500 \times 8}{4 \times 10^5} \times 10^3 \right) \right]$$

پس از جایگذاری نهایی رابطه زیر را خواهیم داشت:

$$T_{\text{Total}} = [300] + [200] = 500 \text{ ms} = \frac{500}{10^3} = 0.5 \text{ s}$$

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه اول را به عنوان پاسخ اعلام کرده بود. که متأسفانه پاسخ درست این سوال در گزینه‌ها نیست.

تست‌های فصل هفتم

۶۰- یک صفحه‌ی وب شامل یک فایل HTML و ۸ آجکت روی یک سرور وجود دارد. اندازه فایل HTML و آجکت‌ها با هم برابر بوده و مقدار هر یک ۵۰۰۰ بایت است. کاربری اقدام به دیدن این صفحه‌ی وب می‌کند. زمان رفت و برگشت بین کامپیوتر کاربر و سرور $RTT = 7msec$ است. متوسط گذردهی ارتباط بین کامپیوتر کاربر و وب سرور برابر با ۴۰ مگابیت بر ثانیه است. فرض کنید کامپیوتر کاربر و سرور از $http$ استفاده می‌کند و سرور اجازه حداکثر پنج ارتباط موازی TCP به کامپیوتر کاربر می‌دهد. از لحظه‌ای که کامپیوتر کاربر $http GET$ را به وب سرور ارسال می‌کند تا زمانی که صفحه‌ی وب را کاملاً دریافت می‌کند چند میلی‌ثانیه زمان صرف می‌شود؟ (توجه داشته باشید که $http$ به صورت $parallel$ و $non-persistent$ (موازی) کار می‌کند.)

۹ (۱)

۲۳ (۲)

۴۴ (۳)

۵۱ (۴)

پاسخ‌های فصل هفتم

۶۰- گزینه (۴) 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 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 پیام رد و بدل نشوند. Connection بین Client و Server ایجاد نشده است، به این سه پیام 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 پیام 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:

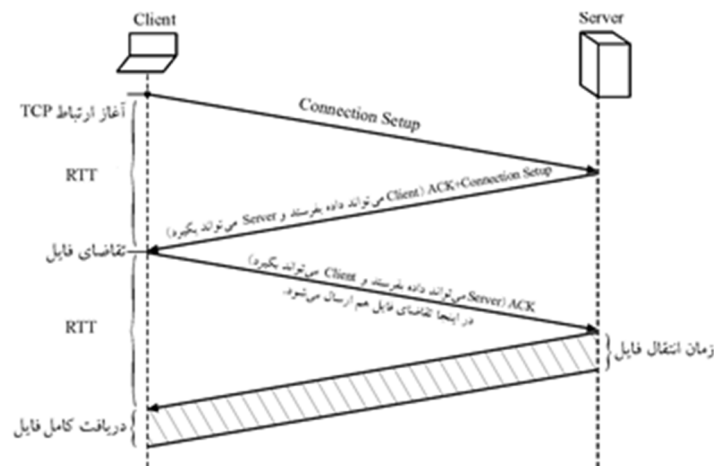
۱- اول این کار را انجام بده

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

RFC یک Reference برای پیاده‌سازی 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 غیرمدام)

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:

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

روش ناپایدار موازی نامحدود:

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

روش ناپایدار موازی محدود:

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

persistent http پایدار (مصر یا مداوم)

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

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

روش پایدار ترتیبی:

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

پایدار موازی نامحدود:

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

روش پایدار موازی محدود:

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

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

یک صفحه‌ی وب شامل یک فایل HTML و ۸ آبجکت روی یک سرور وجود دارد. اندازه فایل HTML و آبجکت‌ها با هم برابر بوده و مقدار هر یک ۵۰۰۰ بایت است. کاربری اقدام به دیدن این صفحه‌ی وب می‌کند. زمان رفت و برگشت بین کامپیوتر کاربر و سرور $RTT = 7 \text{ msec}$ است. متوسط گذردهی ارتباط بین کامپیوتر کاربر و وب سرور برابر با ۴۰ مگابیت بر ثانیه است. فرض کنید کامپیوتر کاربر و سرور از http استفاده می‌کند و سرور اجازه حداکثر پنج ارتباط موازی TCP به کامپیوتر کاربر می‌دهد. از لحظه‌ای که کامپیوتر کاربر http GET را به وب سرور ارسال می‌کند تا زمانی که صفحه‌ی وب را کاملاً دریافت می‌کند چند میلی‌ثانیه زمان صرف می‌شود؟ (توجه داشته باشید که http به صورت parallel و non-persistent (موازی) کار می‌کند.)

داده‌های مسئله به صورت زیر است:

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

$$R_{\text{TOTAL CHANNEL}} = 40 \text{ mbps} = 40 \times 10^6 \text{ bps}, RTT = 7 \text{ ms}$$

$$\text{Cardinality}(\text{Object}) = 8, T_{\text{DNS LOOK UP}} = 0$$

توجه: در صورت سوال محدودیت ترازوی مطرح شده است، پس ترازوی محدود را در نظر می‌گیریم.

رابطه روش ناپایدار (non-persistent) موازی (parallel) محدود به صورت زیر است:

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

$T_{\text{Base HTML}}$ از رابطه زیر بدست می آید:

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

$T_{\text{Base HTML}}$ ، 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 می باشد.

T_{Object} از رابطه زیر بدست می آید:

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

T_{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 می باشد.

با توجه به شرایط ذکر شده در صورت سؤال، مطابق آنچه گفتیم ابتدا بایستی فایل پایه HTML را دریافت کرد و سپس 8 فایل object دیگر را دریافت کرد و حالا با توجه به نوع ارتباط که ناپایدار

موازی محدود است و تنها می توان 5 کانکشن موازی داشت بایستی اینگونه عمل کنیم:

ابتدا یک RTT صرف درخواست و برقراری ارتباط می شود، سپس RTT دیگر صرف درخواست و دریافت فایل پایه HTML می شود و پس از آن ارتباط قطع می شود، بعد از دریافت فایل پایه HTML، با 5 کانکشن موازی درخواست و برقراری ارتباط را در یک RTT می دهیم و در RTT بعدی 5 فایل را دریافت می کنیم و پس از آن ارتباط قطع می شود، سپس با توجه به اینکه تنها 3 فایل باقیمانده است، با 3 کانکشن موازی دیگر درخواست و برقراری ارتباط را در یک RTT می دهیم و در RTT بعدی 3 فایل object باقی مانده را دریافت می کنیم. بنابراین مقدار n در رابطه فوق با عنوان تعداد درخواست ها برابر 2 خواهد بود، زیرا دو کانکشن یکی 5 تایی و دیگری 3 تایی ایجاد کردیم.

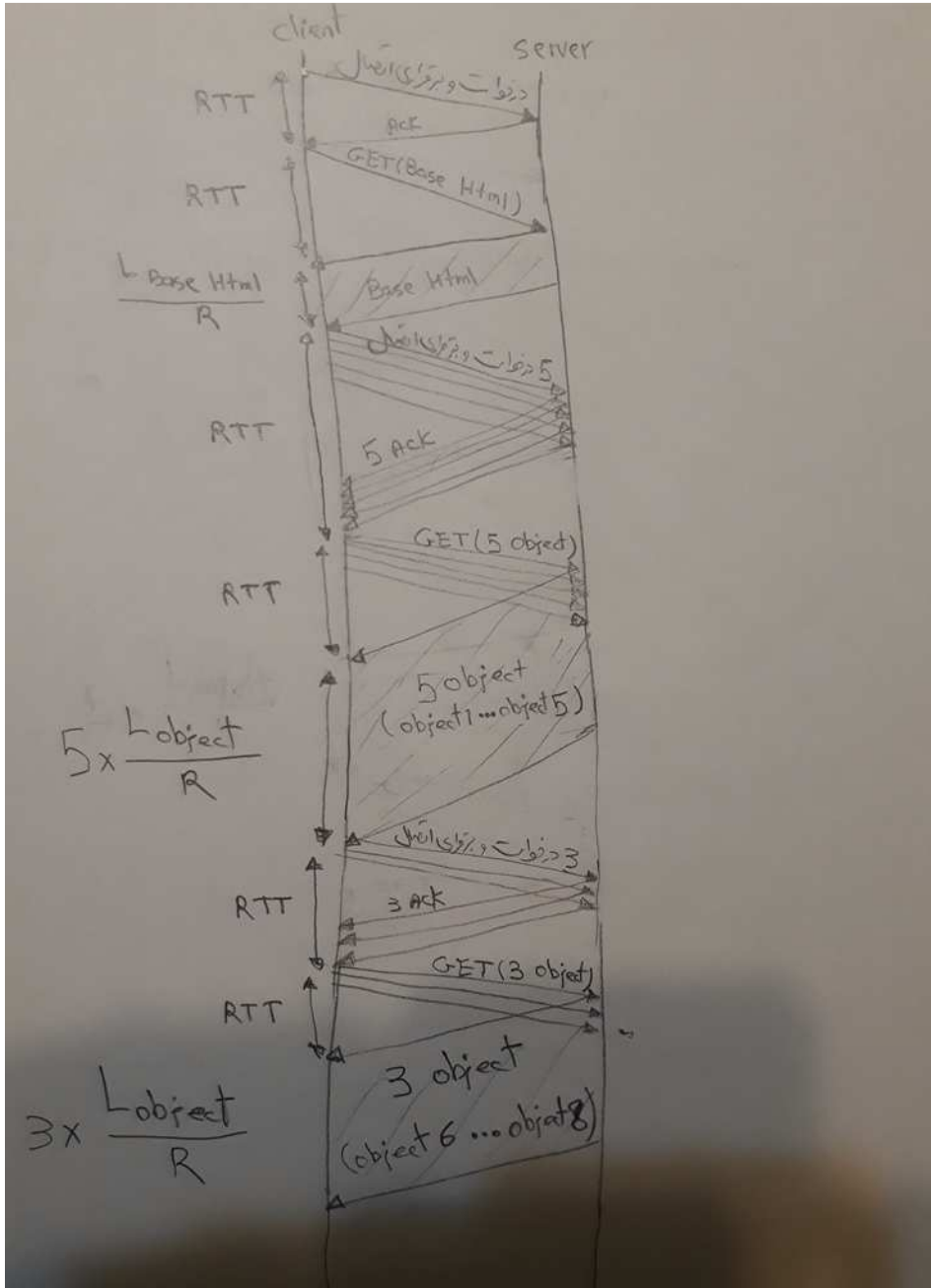
که پس از جایگذاری اولیه رابطه زیر را خواهیم داشت:

$$\left[\left(RTT + RTT + \frac{5000 \times 8}{40 \times 10^6} \times 10^3 \right) \right] + \left[2 \times \left(RTT + RTT \right) + 8 \times \frac{5000 \times 8}{40 \times 10^6} \times 10^3 \right]$$

پس از جایگذاری نهایی رابطه زیر را خواهیم داشت:

$$[(7+7+1)] + [2 \times (7+7) + 8 \times 1] = [(15)] + [28+8] = [(15)] + [36] = 51 \text{ ms}$$

شکل زیر گویای مطلب است:



توجه: البته لازم به ذکر است که طراح به جای اینکه بگوید از لحظه‌ی click روی url، گفته است از لحظه‌ای که کامپیوتر کاربر http GET را به وب سرور ارسال می‌کند، یعنی لحظه‌ی درخواست html پایه (یعنی GET)، که منجر به این می‌شود RTT اول نادیده گرفته شود و پاسخ دقیق‌تر سوال به صورت زیر باشد:

$$[(0+7+1)]+[2\times(7+7)+8\times 1]=[8]+[28+8]=[8]+[36]=44\text{ ms}$$

که البته این مقدار در گزینه‌ی سوم هست.

توجه: سازمان سنجش آموزش کشور، در کلید اولیه و نهایی خود، گزینه چهارم را به عنوان پاسخ اعلام کرده بود.