



بسم خدا
دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

درس شبکه های کامیوتری، نهم سال، دوم سال تحصیلی ۹۹-۰۰

نام و نام خانوادگی:

شماره دانشجویی:

توضیحات:

- مهلت تحویل تمرین ۱۷ اردیبهشت در نظر گرفته شده است و تمدید پذیر نمی باشد.
- پاسخ به تمرین ها به صورت انفرادی باشد و اگر تقلب یافت شود نمره تمرین صفر خواهد شد.
- نظم و خوانایی تمرین از اهمیت بالایی برخوردار می باشد.
- خواهش می شود تمرین خود را در قالب یک فایل PDF با نام "HW2_FirstnameLastName_StdudentNumber" مانند ; HW2_ParsaAliEsfahani_ "9631052.pdf در مهلت یاد شده در سایت بارگزاری فرمایید.
- پرسش های خود درباره این تمرین را می توانید از راه ایمیل AutCnTa@gmail.com بیان کنید.

۱. الف) مشخص کنید سرویس های زیر توسط کدام پروتکل لایه انتقال ارائه می شوند؟

- انتقال داده با قابلیت اطمینان TCP
- امنیت هیچ کدام
- گذردهی بالا هیچ کدام
- کنترل جریان (flow control) TCP
- تضمین دریافت داده در زمانی مشخص هیچ کدام

ب) فرض کنید می خواهید یک تراکنش بین مشتری و سرویس دهنده که از یکدیگر دور هستند را با بیشترین سرعت ممکن انجام دهید. از پروتکل UDP استفاده می کنید یا TCP؟ توضیح دهید.

استفاده از UDP سرعت ارتباط را افزایش می دهد، زیرا سربارهای زمانی برای شروع و خاتمه دادن ارتباط را نداشته و کنترلی بر روی جریان ندارد، از این رو استفاده از UDP در ارتباط های مالی مدیا به صرفه می باشد. البته باید به این نکته دقت داشت که همیشه این افزایش سرعت اتفاق نمی افتد.

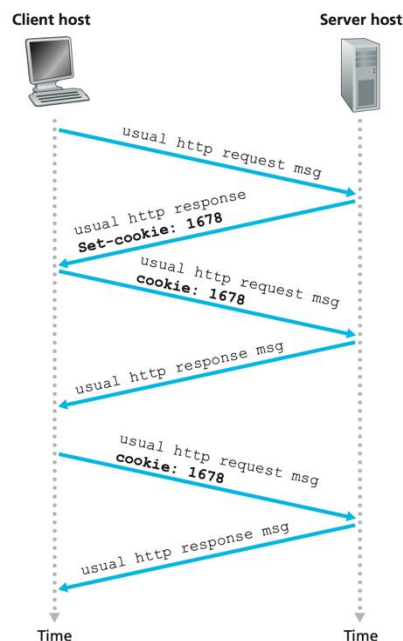
مثلا اگر شبکه دچار ازدحام شده باشد ممکن است تعداد زیادی از بسته ها با ارسال بسته های UDP از بین بروند.

توجه داشته باشید که دور بودن سرویس گیرنده و سرویس دهنده باعث افزایش تاخیر در ارتباط ها می شود و روند ساخت و از بین بردن ارتباط را برای ما هزینه بر می سازد.

۲. هدف استفاده از کوکی ها چیست؟ روند تخصیص کوکی به کاربر را بر حسب هدرهای پروتکل HTTP، به اختصار توضیح دهید.

آیا برای شناسایی کاربران وب، راه دیگری سراغ دارید؟

هنگامی که کاربر برای اولین بار از سایت بازدید می کند، سرور یک شماره شناسایی منحصر به فرد ایجاد می کند و آن را در دیتابیس خود ذخیره می کند. این شماره شناسایی را نیز به در قالب هدر Set-cookie باز می گرداند این شماره در کامپیوتر کاربر ذخیره شده و توسط مرورگر مدیریت می شود. در طول هر بازدید از سایت مذکور، مرورگر آن شماره را در قالب یک هدر Cookie، به وبسایت برمی گرداند. و به این ترتیب وبسایت که این شماره را ذخیره دارد، کاربر را شناسایی می کند.



برای شناسایی کاربران وب، به جز استفاده از کوکی‌ها و Local Storage مرورگر، راه‌های دیگری هم داریم. برای مثال، می‌توانیم آی‌پی کاربر مورد نظر را بررسی کنیم. البته که کوکی‌ها را می‌توان پاک کرد، و آی‌پی را می‌توان تغییر داد. بنا بر این هیچ‌کدام از این راه‌ها قطعی نیستند.

برای مطالعه‌ی بیشتر: یک راه دیگر که جلوی خود داریم، روشی به نام Browser Fingerprinting است. در این روش، به جمع‌آوری تمام اطلاعات قابل برداشت از مرورگر کاربر می‌کنیم. اطلاعاتی از قبیل نوع مرورگر، نسخه‌ی آن، سیستم عامل کاربر، افزونه‌های فعال، منطقه‌ی زمانی، زبان و حتی رزولوشن صفحه‌ی نمایش کاربر. این قبیل اطلاعات یک اثر انگشت (Fingerprint) از کاربر مورد نظر می‌سازند. که به کمک آن می‌توانیم هر کاربر را به صورت یکتا شناسایی کنیم. این روش هم مانند روش‌های قبل به طور همیشگی پاسخ‌گو نیست، اما نتایج یک تحقیق، نشان می‌دهند در عمل، به ازای هر 286,777 مرورگر، ممکن است دو مرورگر اثر انگشت یکسان داشته باشند.

منابع:

<https://coveryourtracks.eff.org/static/browser-uniqueness.pdf>

۳. فرض کنید برای یک وب‌سایت تقاضای HTTP ارسال شده است که فایل HTML شامل n شی بسیار کوچک بر روی همان سرویس‌دهنده است. با صرف نظر کردن از زمان ارسال اشیاء دره‌ریک از حالت‌های زیر محاسبه کنید از زمانی که شما بر روی این لینک کلیک می‌کنید تا زمان دریافت کامل صفحه وب، چه زمانی سپری می‌شود؟ در هر مورد دیاگرام زمانی تبادل پیام‌ها بین سرویس‌گیرنده و سرویس‌دهنده را رسم کنید.

زمان رفت و برگشت تا سرویس‌دهنده را برابر RTT_0 در نظر بگیرید.

- HTTP ناپایا (Non-Persistent) بدون هیچ اتصال موازی TCP

$$2RTT_0 + n \times 2RTT_0 = (2n + 2)RTT_0$$

- HTTP ناپایا با K اتصال موازی

$$2RTT_0 + 2 \times \left\lceil \frac{n}{k} \right\rceil RTT_0$$

- HTTP پایا (Persistent) در حالت غیر پایپ لاین

$$2RTT_0 + nRTT_0$$

- HTTP پایا (Persistent) در حالت پایپ لاین

$$2RTT_0 + RTT_0$$

۴. یک لینک ارتباطی کوتاه ۲۰ متری با نرخ ارسال ۳۰۰۰ بیت بر ثانیه را در نظر بگیرید. اگر طول بسته‌های داده ۸۴۰۰۰ بیت و طول بسته‌های کنترلی ۲۰۰ بیت باشد با فرض اینکه می‌توان n اتصال موازی ایجاد کرد که هر کدام $\frac{1}{n}$ از پهنای باند را می‌گیرند، اگر هر شیء دریافتی در پروتکل HTTP، ۸۴ کیلو بیت باشد و اولین شیء به ۲۰ شیء دیگر ارجاع بدهد، تاخیر دریافت یک صفحه‌ی وب با استفاده از روش HTTP ناپایا (Non-Persistent) و روش HTTP پایا (Persistent) در حالت بدون پایپ لاین و با پایپ لاین چقدر است؟

$$\text{Link Length (d)} = 20 \text{ m}$$

$$\text{Transmission Rate (R)} = 3000 \text{ bps}$$

$$\text{Data Packet Length (L}_d\text{)} = 84000 \text{ bit}$$

$$\text{Control Packet Length (L}_c\text{)} = 200 \text{ bit}$$

$$\text{HTTP Object Length} = 84000 \text{ bit}$$

بنابراین هر شیء HTTP در یک بسته منتقل می‌شود.

$$\text{Propagation Delay (t}_{prop}\text{)} = \frac{d}{v} = \frac{20}{20 * 10^7} = 0.1 \mu\text{sec}$$

$$\text{Round Trip Time (RTT)} = 2t_{prop} = 0.2 \mu\text{sec}$$

ارتباط ناپایا با ۲۰ ارتباط همزمان:

Data Packet Transmission Time for one parallel connection (t_{d1}) = $\frac{L_d}{R} = \frac{84000}{3000} = 28 \text{ sec}$
(base HTML)

Control Packet Transmission Time for one parallel connection (t_{c1}) = $\frac{L_c}{R} = \frac{200}{3000} = 0.067 \text{ sec}$ (base HTML)

Data Packet Transmission Time for 20 parallel connection (t_{d20}) = $\frac{L_d}{R/k} = \frac{84000}{3000/20} = 560 \text{ sec}$

Control Packet Transmission Time for 20 parallel connection (t_{c20}) = $\frac{L_c}{R/k} = \frac{200}{3000/20} = 1.333 \text{ sec}$

$RT_{non-persistent} = (RTT + 2t_{c1} + RTT + t_{c1} + t_{d1}) + (RTT + 2t_{c20} + RTT + t_{c20} + t_{d20}) = 4RTT + 3t_{c1} + t_{d1} + 3t_{c20} + t_{d20} = 8t_{prop} + 3 * 0.067 + 28 + 3 * 1.333 + 560 \approx 592.200$

ارتباط پایا:

Data Packet Transmission Time (t_d) = $\frac{L_d}{R} = \frac{84000}{3000} = 28 \text{ sec}$

Control Packet Transmission Time (t_c) = $\frac{L_c}{R} = \frac{200}{3000} = 0.067 \text{ sec}$

بدون پایپ لاین:

$RT_{persistent-nopipeline} = (RTT + 2t_c + RTT + t_c + t_d) + 20(RTT + t_c + t_d) = 22RTT + 23t_c + 21t_d = 44t_{prop} + 23 * 0.067 + 21 * 28 = 589.541$

با پایپ لاین:

$RT_{persistent-pipeline} = (RTT + 2t_c + RTT + t_c + t_d) + (RTT + t_c + 20t_d) = 3RTT + 4t_c + 21t_d = 6t_{prop} + 4 * 0.067 + 21 * 28 = 588.268$

۵. متن زیر قسمتی از درخواست HTTP از طرف مرورگر به سرور و پاسخ آن از طرف سرور می باشد. با توجه به آن به سوالات زیر پاسخ دهید.

```
GET /users/login HTTP/1.1
Host: stackoverflow.com
HTTP/1.1 200
accept-ranges: bytes
age: 84963
cache-control: private
content-encoding: gzip
content-security-policy: upgrade-insecure-requests
content-type: text/html; charset=utf-8
date: Mon, 19 Apr 2021 14:35:21 GMT
last-modified: Mon, 19 Apr 2021 13:28:24 GMT
```

الف) آدرس URL ای که کاربر در مرورگر وارد کرده است را بنویسید.

stackoverflow.com/users/login

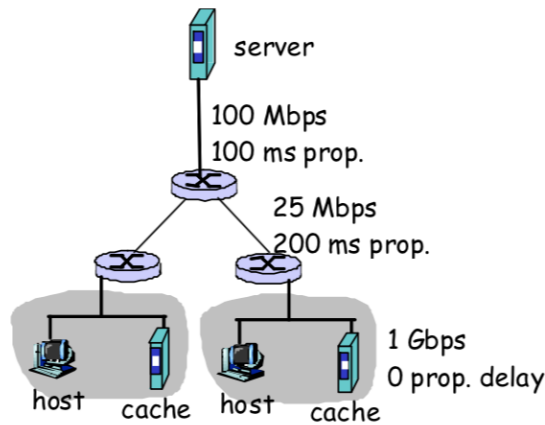
ب) نوع فایل ارسالی توسط سرور چیست و چند بایت دارد؟

[text/html - 84963](#)

ج) زمان آخرین تغییر این فایل در سرور چه زمانی بوده است؟

[Mon, 19 Apr 2021 13:28:24 GMT](#)

۶. شبکه ی شکل زیر را در نظر بگیرید. نرخ انتقال و تأخیر انتشار لینک ها در شکل مشخص شده است. طول بسته ها را ثابت و برابر 10,000 بیت در نظر بگیرید.



الف) تأخیر ارسال یک بسته از سرور به یکی از hostها را محاسبه کنید. فرض کنید از Caching استفاده نمی کنیم، و تأخیرهای صف و پردازش ناچیز هستند.

تأخیر انتقال یک بسته را به صورت زیر محاسبه می کنیم.

$$t_{\text{transmission}} = \frac{L}{R_{\text{Link1}}} + \frac{L}{R_{\text{Link2}}} + \frac{L}{R_{\text{Link3}}} = \frac{10000_b}{100_{\text{bps}} \times 10^6} + \frac{10000_b}{25_{\text{bps}} \times 10^6} + \frac{10000_b}{1_{\text{bps}} \times 10^9} = 510 \text{ (us)}$$

همچنین برای تأخیر انتشار داریم.

$$t_{\text{propagation}} = t_{\text{prop1}} + t_{\text{prop2}} + t_{\text{prop3}} = 300 \text{ (ms)}$$

بنابراین مجموع تأخیر برابر خواهد بود با:

$$t = t_{\text{trans}} + t_{\text{prop}} = 300.510 \text{ (ms)}$$

ب) فرض کنید تنها یک host فعال بوده، و از Caching استفاده نمی کنیم. سرور با حداکثر چه نرخ انتقالی می تواند داده به یکی از hostها ارسال کند؟

سرور حداکثر با ظرفیت کمترین لینک مسیر (Bottleneck) می تواند داده ارسال کند، بنا بر این حداکثر نرخ ارسال برابر با 25Mbps خواهد بود.

ج) بار دیگر فرض کنید تنها یک host فعال بوده، اما این بار از Caching استفاده می کنیم. تمامی درخواست ها ابتدا به Local Cache Server ارسال می شوند. 50% درخواست ها توسط Cache Server با موفقیت پاسخ داده می شوند. در این صورت، حداکثر نرخ دریافت داده ی host چقدر خواهد بود؟

در این صورت نیمی از درخواست ها با نرخ 25Mbps، و نیمی دیگر با نرخ 1Gbps سرویس دهی می شوند. در نتیجه:

$$\bar{R} = \left(\frac{50}{100} \times 10^9_{bps}\right) + \left(\frac{50}{100} \times 25_{bps} \times 10^6\right) = 512.5 (Mbps)$$

د) با فرضیات قسمت ج، در صورتی که این بار هر دو host فعال باشند، حداکثر نرخ قابل دسترسی برای host ها چقدر خواهد بود؟ لینک مشترک به طور مساوی بین host ها تقسیم می شود.

همچنان لینک با کمینه ی ظرفیت، لینک 25Mbps خواهد بود، که میان host ها به اشتراک گذاشته نشده است، پس جواب همچنان همان جواب قسمت قبل خواهد بود.

۷. فرض کنید می خواهید ایمیلی حاوی یک عکس به دوست خود ارسال کنید. می دانیم این پیام از ۴ گره عبور می کند، که به ترتیب عبارتند از کلاینت ایمیل و میل سرور شما، و همچنین میل سرور و کلاینت ایمیل دوست شما.

الف) پروتکل SMTP بین کدام یک از گره ها عمل می کند؟ پروتکل های Mail Access چگونه؟

پروتکل SMTP بین کلاینت ایمیل فرستنده تا میل سرور فرستنده، و همچنین به طور جداگانه بین میل سرور فرستنده تا میل سرور گیرنده عمل می کند. پروتکل های Mail Access بین میل سرور گیرنده تا کلاینت گیرنده عمل می کنند.

ب) تحقیق کنید که با پروتکل SMTP که در قسمت Body خود تنها کاراکترهای ASCII قبول می کند، چگونه می توان عکس مورد نظر را ارسال کرد؟ آیا می توانیم مستقیماً رشته ی باینری عکس مورد نظر را به صورت ۷ بیت ۷ بیت به کاراکترهای ASCII ترجمه و ارسال کنیم؟ (راهنمایی: به RFC 2045 مراجعه کنید)

برای ارسال انواع مختلف پیام در قالب ایمیل، MIME ها مطرح شده اند. MIME که مخفف Multipurpose Internet Mail Extensions می باشد، روشی برای ارسال یک ایمیل با موارد زیر ارائه می کند.

- پیامی شامل کاراکترهایی غیر از کاراکترهای ASCII، (مثلاً کاراکترهای زبان فارسی)

- پیام غیر متنی، مانند عکس و ویدیو

- پیامی شامل چندین بخش (Multipart Message)

- اضافه کردن هدر با کاراکترهایی غیر از ASCII

خیر، امکان کدگذاری مستقیم رشته ی باینری به ASCII را نداریم، چرا که در این صورت در بعضی موارد دنباله ای از صفر و یک ها، ممکن است کاراکترهای کنترلی ناخواسته ایجاد کنند، و پیام را دچار خطا کنند. منظور از کاراکترهای کنترلی، کاراکترهایی با معنای خاص

هستند، مثلاً کاراکتری برای اعلام پایان یک پیام. کدگذاری رشته های باینری به کمک ASCII مشکلات دیگری هم دارد، مانند مشکل ۷ بیتی بودن ASCII. به همین دلیل از تکنیک های کدگذاری ای مانند Base64 استفاده می شود.

ج) چرا نیاز به پروتکل های Mail Access داریم؟ چرا SMTP به صورت مستقیم پیام را بین کلاینت ها منتقل نمی کند؟ پروتکل های Mail Access را به همراه پورت مورد استفاده شان نام ببرید.

پروتکل SMTP از نوع Push بوده، یعنی فرستنده ی پیام است که سعی می کند یک ارتباط TCP با گیرنده باز کند و پیام را ارسال کند. حال اگر بخواهیم پیام را از کلاینت فرستنده به کلاینت گیرنده مستقیم ارسال کنیم، در آن صورت کلاینت گیرنده در همه ی زمان ها روشن و آماده ی دریافت پیام باشد. چون کاربران اینترنت امکان روشن گذاشتن دائمی دستگاه خود را ندارند، بنا بر این از توپولوژی گفته شده در صورت سؤال استفاده می شود که در آن، دو میل سرور میانی، همواره روشن بوده، و وظیفه ی ارسال و دریافت پیام را دارند. برای این که کلاینت گیرنده ایمیل های دریافت شده ی میل سرور را بگیرد، از پروتکل های Mail Access استفاده می کنیم.

چند مورد از این پروتکل ها عبارتند از:

- POP3 با پورت 110
- IMAP با پورت 143
- HTTP با پورت 80

۸. فرض کنید دانشگاه ما قصد دارد آدرس IP وبسایت <https://www.aut.ac.ir> را از ۱۸۵,۲۱۱,۸۸,۱۳۱ به ۱۸۵,۲۱۱,۸۸,۱۳۲ تغییر دهد. آیا به محض این که رکورد مورد نظر در Authoritative name server تغییر کرد، تمام درخواست های <https://www.aut.ac.ir> به آدرس ۱۸۵,۲۱۱,۸۸,۱۳۲ ارسال می شوند؟ چرا؟

خیر. به طور کلی DNS Resolver ها، پس از گرفتن یک رکورد، آن را در خود Cache کرده، و در پرس و جوهای بعدی از آن رکورد، به جای آن که یک درخواست به سرور ایجاد کنند، از Cache خود برای پاسخ گویی استفاده می کنند. این رکوردها یک (TTL Time to Live) دارند، و بعد از اتمام این مدت زمان، دوباره درخواست جدید جهت به روزرسانی آن رکورد ارسال می شود.

در نتیجه، تا زمانی که آن رکورد قدیمی (نگاشت به 185.211.88.131)، از Cache ها پاک نشده باشد، ممکن است کاربرها به آدرس آی پی قدیمی هدایت شوند.

۹. فرض کنید می خواهیم یک فایل $F=1Gb$ را بین N نظیر توزیع کنیم. نرخ بارگذاری (Upload) سرویس دهنده $u_s = 30Mbps$ نرخ دریافت (Download) هر نظیر $d_i = 2Mbps$ و نرخ بارگذاری آن $u = 300Kbps$ است. به ازای $N = 10, 100, 1000$ جدولی رسم کنید و حداقل زمان توزیع را برای هر دو معماری Client - Server و Peer to Peer مشخص کنید.

N

10

100

1000

Client – Server

Peer to Peer

در ارتباط Client-Server، رابطه ی زیر برقرار است:

$$D_{CS} = \max\{NF/u_s, F/d_{min}\}$$

در نتیجه:

N	$\frac{NF}{u_s}$	$\frac{F}{d_{min}}$	max
10	333.3	500	500
1000	3333.3	500	3333.3
10000	33333.3	500	33333.3

همچنین در ارتباط Peer-to-Peer از رابطه زیر استفاده می کنیم:

$$D_{p2p} = \max\{F/u_s, F/d_{min}, \frac{NF}{u_s + \sum u_i}\}$$

N	$\frac{F}{u_s}$	$\frac{F}{d_{min}}$	$\frac{NF}{u_s + \sum u_i}$	max
10	33.3	500	303.3	500
1000	33.3	500	1666.6	1666.6
10000	33.3	500	3030.3	3030.3

(اعداد جدول ها در واحد ثانیه هستند.)

۱۰. برای ارائه ی سرویس Streaming، دو راه Single Server و CDN را پیش روی خود داریم. مزایای استفاده از CDN چیست؟

استفاده از یک سرور مرکزی بزرگ، معایب زیر را به همراه خواهد داشت

- Single Point of Failure : در صورت بروز مشکلی در این سرور، سرویس دهی به تمامی کاربران دچار مشکل خواهد شد.
 - Point of Network Congestion : چون درخواست تمام کاربران در نهایت به این سرور می رسد، در این سرور ازدحام شدیدی رخ خواهد داد.
 - Long Path to Distant Clients : چون تنها یک سرور مرکزی داریم، این سرور به ناچار از بعضی از کاربران دور خواهد بود، که باعث سرویس دهی کندتر به آن دسته از کاربران خواهد شد.
 - Multiple Copies of Video sent over Outgoing link : ویدیوهایی که زیاد مورد درخواست می گیرند، به طور غیربهرینه چندین بار ارسال شوند.
- CDN ها تمام مشکلات بالا را حل خواهند کرد.