



دانشکده مهندسی کامپیوتر

بسمه تعالی
دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

درس شبکه‌های کامپیوتری، نیم سال دوم سال تحصیلی ۱۴۰۱-۱۴۰۲

پاسخنامه‌ی تمرین سوم

سوال ۱:

الف) معماری Client-Server و Peer-To-Peer را توضیح داده و با یکدیگر مقایسه کنید.
شبکه‌های P2P مجموعه‌ای decentralized (غیر متمرکز) از کامپیوترها می‌باشند که به تبادل اطلاعات می‌پردازند. در این شبکه هر گره آن قابلیت ارسال و دریافت اطلاعات را دارد، به عبارت دیگر هر گره می‌تواند هم در نقش Client و هم در نقش Server ظاهر شود و همه‌ی گره‌ها برابر در نظر گرفته می‌شوند.

شبکه‌های Client-Server از یک یا چند سرور Centralized تشکیل شده‌اند که سرویس‌ها و دیتاهای مختلف را میزبانی می‌کنند و Clientها از این سرورها سرویس می‌گیرند. سرور در این معماری می‌تواند در نقش یک واسطه، اطلاعات Clientهای مختلف را به یکدیگر منتقل کند؛ ولی خود Clientها مستقیماً باهم در ارتباط نیستند.

مقایسه:

P2P وابسته به یک گره نیست، و در صورت قطع شدن یک گره، دیگر گره‌ها همچنان در شبکه حاضر بوده و قادر به تبادل اطلاعات هستند. اما در Client-Server در صورتی که Server دچار مشکل شود، Clientها قادر به دریافت از سرویس‌ها و داده‌ها نخواهند بود. به عبارت دیگر، در این معماری، Server یک Single Point of Failure می‌باشد. برای درستی کارکرد شبکه، Server باید همواره آماده‌ی خدمت‌رسانی باشد. پیاده‌سازی Server-Client هزینه بیشتری نسبت به P2P دارد زیرا نیاز به سروری داریم که همیشه سرویس‌ها را اجرا کند و داده‌ها را نگه دارد. امنیت فایلها دریافتی در Server-Client بیشتر است زیرا سرور روی فایلها کنترل دارد. مزیت معماری P2P در قابلیت خودمقیاس‌پذیری (Self-Scaling) آن است که در آن هر یک از گره‌ها، علاوه بر دریافت داده‌ها (خدمت گرفتن)، می‌توانند به بازتوزیع (Redistribution) داده‌ها (خدمت دادن) کمک کنند. در مقابل در معماری Client-Server، نقش توزیع تنها بر عهده‌ی سرور است، در نتیجه با افزایش تعداد Clientها و فشار بیش‌تر روی سرور، زمان توزیع افزایش می‌یابد. شکل زیر مقایسه‌ای از قابلیت Scaling بین این دو معماری در کتاب درس است.

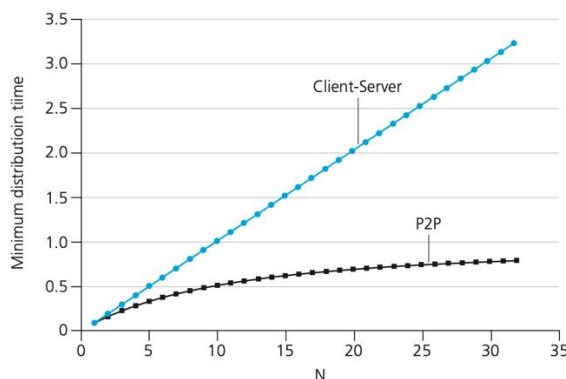


Figure 2.23 ♦ Distribution time for P2P and client-server architectures

سوال ۲:

در یک پروتکل دلخواه لایه‌ی کاربرد، چه مواردی باید تعریف شوند؟ برای هر مورد، مثالی از تعریف آن در پروتکل HTTP بزنید.
در یک پروتکل لایه‌ی Application، موارد زیر تعریف می‌شود:

- نوع پیام‌های رد و بدل شده (برای مثال Request و Response)
- نحو (Syntax) پیام‌ها یعنی فیلدهای پیام و جایگذاری‌شان
- معنانشناخت (Semantics) پیام‌ها یعنی مفهوم هر یک از فیلدها
- قوانینی برای توصیف آن که پیام‌ها چه زمانی و به چه صورت پردازش و پاسخ داده شوند.

برای مثال در پروتکل HTTP که در [RFC 2616](#) تعریف شده است، داریم:

- در بخش 4 و زیربخش 4.1 نوع پیام‌های این پروتکل تعریف شده‌اند.
- به عنوان مثال، سینتکس URI‌ها در این پروتکل، در بخش 3.2.1 تعریف شده‌اند.
- به طور کلی، بخش 3 به توصیف معنای هر یک پارامترهای پروتکل پرداخته است.
- بخش 13 قوانین پردازش Cache در HTTP را معین کرده است.

سوال ۳:

الف) چهار دسته خدمت کلی که پروتکل‌های لایه‌ی انتقال فراهم می‌کنند را نام برده و توضیح دهید هر یک توسط کدام یک از پروتکل‌های UDP یا TCP (و یا هر دو) برآورده می‌شوند.

چهار دسته خدمت کلی به شرح زیر هستند:

- Reliable Data Transfer
- پروتکل TCP، برخلاف پروتکل UDP این خدمت را فراهم می‌کند،

- تضمین نرخ گذردهی (Throughput)
 - تضمین زمان رسیدن داده‌ها (Timing)
 - امنیت: صحت و محرمانگی داده‌ها (Security: Integrity & Confidentiality)
- هیچ یک از پروتکل‌های TCP و UDP این سه خدمت را فراهم نمی‌کنند.



دانشگاه مهندسی کامپیوتر

درس شبکه‌های کامپیوتری، نیم سال دوم سال تحصیلی



دانشگاه صنعتی امیرکبیر
(پاسی تکنیک تهران)

ب) برای سه کاربرد «بازی‌های تعاملی»، «دریافت فایل‌های وب» و «ارسال ایمیل» توضیح دهید کدام یک از خدمات‌های قسمت قبل، ضروری‌ست؟

بازی‌های تعاملی، نیازی به Reliable Data Transfer نداشته و حتی در صورت از دست رفتن بخشی از داده‌ها، همچنان بازی برای کاربران ممکن است. این بازی‌ها روی نرخ گذردهی و زمان رسیدن داده‌ها محدودیت داشته و نرخ گذردهی کم‌تر از یک آستانه (برای مثال چند کیلوبایت بر ثانیه و زمان رسیدن بیش از چند ثانیه) بازی را غیر ممکن می‌کنند. هر دو کاربرد دریافت فایل‌های وب و ارسال ایمیل، نیاز به Reliable Data Transfer دارند، در غیر این صورت، فایل و ایمیل دریافتی برای گیرنده قابل استفاده نخواهد بود. از طرف دیگر این دو کاربرد روی نرخ گذردهی و زمان رسیدن داده، محدودیت ندارند.

سوال ۴:

توضیح دهید که به چه دلیلی ممکن است یک توسعه‌دهنده برنامه، به جای TCP از UDP استفاده کند؟ از جمله دلایل ترجیح پروتکل TCP به UDP به شرح زیر است.

- کنترل دقیق‌تر در سطح application بر روی داده‌ها و زمان ارسال‌شان.
 - پروتکل TCP داده‌های دریافتی از لایه‌ی بالاتر را با مکانیزم‌های Congestion Control خود ارسال می‌کند، به این معنی که در صورت شلوغی بیش از حد پیوندها، نرخ ارسال را محدود می‌کند، همچنین برای رسیدن به Reliable Data Transfer، بسته‌ها ممکن است چندین بار ارسال شوند. در مواقعی برنامه‌ها به چنین نیازمندی‌هایی دچار نیستند (مثلاً برنامه‌های real-time)، استفاده از UDP مناسب‌تر است.
- ایجاد نکردن یک connection و نداشتن connection state
 - پروتکل TCP برای رسیدن به ویژگی‌های گفته شده در قسمت قبل، یک connection و برای آن یک connection state در نظر می‌گیرد که در آن پارامترهای اتصال مشخص شده است. یک سرور با پروتکل UDP که این پارامترها را نگهداری نمی‌کند، می‌تواند به تعداد Client بسیار بیش‌تری خدمت‌رسانی کند.
- سر بار کم‌تر: اندازه‌ی سرآیند این پروتکل (۸ بایت) از پروتکل TCP (۲۰ بایت) کم‌تر است.

سوال ۵:

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



```
GET /wiki/Computer_network HTTP/1.1
Host: en.wikipedia.org:443

HTTP/1.1 200
accept-ranges: bytes
age: 99397
backend-timing: D=146476 t=1550415632220341
cache-control: private, s-maxage=0, max-age=0, must-revalidate
content-encoding: gzip
content-language: en
content-length: 69726
content-type: text/html; charset=UTF-8
date: Mon, 18 Feb 2019 18:37:10 GMT
last-modified: Fri, 15 Feb 2019 15:42:29 GMT
```

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

https://en.wikipedia.org/wiki/Computer_network

ب) زمان آخرین تغییر این فایل در سرور چه زمانی بوده است. این سرآیند، چه کاربردی دارد؟
 زمان آخرین تغییر این فایل در سرآیند last-modified قابل مشاهده است. یکی از کاربردهای این سرآیند، درخواست Conditional GET می‌باشد. حافظه‌های Cache با گذر زمان ممکن است در خود نسخه‌های قدیمی یک داده را ذخیره کرده باشند. بنا بر این با هر بار دریافت یک داده از وب‌سرور اصلی، در کنار خود داده، سرآیند last-modified داده را نیز در خود ذخیره می‌کنند. بنا بر این در درخواست‌های بعدی، Cache ابتدا می‌تواند با سرآیندی به نام if-modified-since از وب‌سرور پرس‌وجو کند، تا مطمئن شود همچنان آخرین نسخه‌ی داده را دارد، و سپس به درخواست Client پاسخ دهد.

ج) نوع فایلی که سرور فرستاده چیست و چند بایت دارد؟

[text/html - 69726](#)

سوال ۶:

یک پیوند ارتباطی ۱۰ متری با نرخ 150 bps را در نظر بگیرید. اگر طول بسته‌های داده $100,000$ بیت و طول بسته‌های کنترلی 200 بیت باشد، با فرض اینکه می‌توان N اتصال موازی ایجاد کرد که هر کدام $1/N$ از پهنای باند را می‌گیرند، اگر هر شیء دریافتی در پروتکل HTTP $100,000$ بیت باشد و اولین شیء به ۱۰ شیء دیگر روی همان وب‌سرور ارجاع دهد، تأخیر دریافت یک صفحه‌ی وب با استفاده از روش non-persistent HTTP (غیر مداوم) و با اتصال‌های موازی هم‌زمان چقدر است؟

Link Length (d) = 10 m
 Transmission Rate (R) = 150 bps
 Data Packet Length (L_d) = $10^5 b$
 Control Packet Length (L_c) = 200b
 HTTP Object Length = 100,000 bit
 Propagation Speed (v)

هر شیء HTTP را می‌توان در قالب یک بسته (L_D) ارسال کرد.



$$RTT = 2T_{prop} = 2\left(\frac{d}{v}\right) = 2\left(\frac{10}{v}\right) = \frac{20}{v}$$

$$\text{Base HTML: Data Packet Transmission Time for a single connection: } t_{d1} = \frac{L_d}{R} = \frac{10^5}{150} = 666. \bar{6}s$$

$$\text{Base HTML: Control Packet Transmission Time for a single connection: } t_{c1} = \frac{L_c}{R} = \frac{200}{150} = 1. \bar{3}s$$

$$\text{Data Packet Transmission Time for 10 Parallel Connections: } t_{d10} = \frac{L_d}{R/k} = \frac{10^5}{3000/10} = 333. \bar{3}s$$

$$\text{Control Packet Transmission Time for 10 Parallel Connections: } t_{c10} = \frac{L_c}{R/k} = \frac{200}{3000/10} = 0. \bar{6}s$$

$$\begin{aligned} \text{total delay} &= \text{delay}_{\text{baseHTML}} + \text{delay}_{10\text{Objects}} = (RTT + 2t_{c1} + RTT + t_{c1} + t_{d1}) + (RTT + 2t_{c10} + RTT + t_{c10} + t_{d10}) \\ &= 4RTT + 3t_{c1} + t_{d1} + 3t_{c10} + t_{d10} = 4RTT + (3 \times 1. \bar{3}) + 666. \bar{6} + (3 \times 0. \bar{6}) + 333. \bar{3} \\ &= 4RTT + 4 + 2 + 1000 = 4RTT + 1006s \end{aligned}$$

سوال ۷:

یک صفحه وب شامل یک فایل HTML=5000 Byte و ۶ شی است. فایل HTML=5000 Byte و شی های 01=2000 Byte و 02 = 4000 Byte روی وب سرور ۱ و شی های 03= 2000 Byte و 04 = 4000 Byte روی وب سرور ۲ و شی های 05=5000 Byte و 06 = 7000 Byte روی وب سرور ۳ قرار دارند.

کاربری مشتاق است که این صفحه وب را ببیند. زمان رفت و برگشت بین کامپیوتر کاربر و سرور ۱ به اندازه RTT1=0.03 sec است. زمان رفت و برگشت بین کامپیوتر کاربر و سرور ۲ به اندازه RTT2=0.04 sec است و زمان رفت و برگشت بین کامپیوتر کاربر و سرور ۳ به اندازه RTT3=0.02 sec است. متوسط گذردهی ارتباط بین کامپیوتر کاربر و سرور ۱ برابر با R1=80000bps است، گذردهی ارتباط بین کامپیوتر کاربر و سرور ۲ برابر با R2=40000bps و گذردهی ارتباط بین کامپیوتر کاربر و سرور ۳ برابر با R3=80000bps است. از لحظه ای که کاربر تقاضا GET را برای دریافت صفحه وب ارسال می کند تا زمانی که صفحه وب را کاملاً دریافت می کند چند میلی ثانیه زمان صرف می شود؟

(فرض کنید ارتباط HTTP با هر یک از سرورها به صورت پایا و غیر پایپ لاین است و ارتباط هم زمان با هر سه سرور می تواند وجود داشته باشد).

ابتدا فایل پایه را از سرور ۱ دریافت کرده و سپس در حالی که ارتباط اول را نگه داشته ایم، دو ارتباط همزمان با هر کدام از دو وب سرور دیگر نیز برقرار می کنیم. سپس مدت زمانی که طول می کشد تا آبجکت ها به صورت کامل دریافت بشوند را بدست آورده و بیشینه ی آن ها را به دست می آوریم. در این صورت مدت زمان کل دریافت صفحه وب برابر با بیشینه ی به دست آمده به علاوه ی مدت زمان دریافت فایل پایه می باشد.

$$\text{Base}=2RTT1+5000*880000=2*0.03+0.5=0.56 \text{ sec}$$

$$01=RTT1+2000*880000=0.03+0.2=0.23 \text{ sec}$$

$$02= RTT1+4000*880000=0.03+0.4=0.43 \text{ sec}$$

$$03=2RTT2+2000*840000=2*0.04+0.4=0.48 \text{ sec}$$

$$04=RTT2+4000*840000=0.04+0.8=0.84 \text{ sec}$$

$$05=2RTT3+5000*880000=2*0.02+0.5=0.54 \text{ sec}$$

$$06=RTT3+7000*880000=0.02+0.7=0.72 \text{ sec}$$

$$01+02=0.23+0.43=0.66 \text{ sec}$$

$$03+04=0.48+0.84=1.32 \text{ sec}$$

$$05+06=0.54+0.72=1.26 \text{ sec}$$

$$\text{Total}=0.56+0.66, 1.32, 1.26=0.56+1.32=1.88 \text{ sec}$$

سوال ۸:

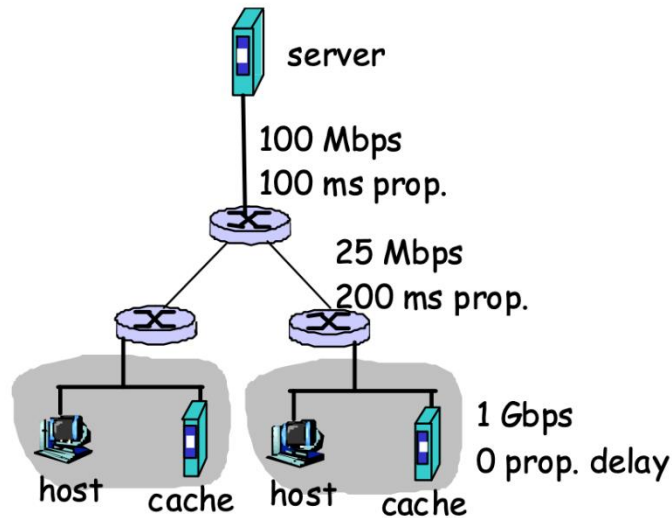
برای احراز هویت پرتال دانشگاه قصد داریم از کوکی‌ها استفاده کنیم. به طور دقیق‌تر، هر بار که یک دانشجو به حساب کاربری‌اش وارد شد، از سمت سرور در پاسخ به او یک کوکی با شناسه‌ای یکتا اختصاص می‌دهیم. به این صورت کاربر با فرستادن کوکی‌اش در کنار درخواست‌های بعدی خود، نیازی به احراز هویت دوباره و وارد کردن رمز عبورش ندارد. آیا این روش به نظر شما امن است؟ تحقیق کنید که آسیب‌پذیری این روش چیست و چگونه باید با آن مقابله کرد؟

از جمله خطرات بالقوه‌ی این روش، امکان وقوع Cookie Hijacking است. در چنین سناریویی، یک مهاجم با دستیابی به کوکی یک کاربر دسترسی غیرمجاز به حساب کاربری وی پیدا می‌کند. سرقت کوکی‌ها به دلایل مختلف ممکن است رخ دهد؛ برای مثال

- در صورتی که پارامتر SecureFlag کوکی تنظیم نشده باشد، کوکی‌ها به صورت خام و بدون رمزنگاری ارسال می‌شوند. در چنین شرایطی امکان شنود ترافیک و دستیابی به کوکی‌ها وجود خواهد داشت. در مقابل اگر این پارامتر تنظیم شده باشد، کوکی‌ها تنها از کانال‌های امن ارتباطی TLS/SSL عبور می‌کنند.
- Cross-site Scripting و به اختصار XSS: در این روش از وبسایت‌هایی که اجازه‌ی پست کردن قطعه‌کدهای HTML و یا Javascript فیلتر نشده و خام می‌دهند، برای سرقت کوکی استفاده می‌شود. برای مثال، کاربر ممکن است روی یک لینک مخرب، که توسط مهاجم پست شده، کلیک کند تا قطعه‌کدی جاوااسکریپتی اجرا شود و کوکی‌های کاربر را برای مهاجم ارسال کند.
- Cookie Tossing: این نوع حمله از مکانیزم به اشتراک‌گذاری کوکی بین دامنه و زیردامنه‌ها سوءاستفاده می‌کند. زمانی که یک زیردامنه کوکی‌ای را تنظیم می‌کند، دامنه‌ی والد می‌تواند آن کوکی را بخواند، اما نمی‌تواند تشخیص دهد که این کوکی توسط خود و یا یکی از زیردامنه‌ها تنظیم شده است. در صورتی که دامنه‌ی والد از کلیدی مشابه با زیردامنه برای یک کوکی استفاده کند، در این صورت ممکن است کوکی تنظیم شده توسط زیردامنه را به اشتباه کوکی خود در نظر گرفته و از آن استفاده کند. در چنین شرایطی مهاجم می‌تواند با تنظیم کوکی مد نظر در یکی از زیردامنه‌ها به مکانیزم احراز هویت دامنه‌ی اصلی نفوذ کند. واضحاً چنین حمله‌ای زمانی رخ می‌دهد که یک وبسایت با دامنه‌ای مشخص، اجازه‌ی استفاده از زیردامنه‌هایش را به افراد غیرمطمئن دهد.
- Cross-site request forgery و به اختصار CSRF

سوال ۹:

سناریوی شکل مقابل را در نظر بگیرید که در آن یک سرور به کمک یک لینک با پهنای باند 100 Mbps و تأخیر انتشار 100 ms به یک مسیریاب متصل شده است. در ادامه مسیریاب به دو مسیریاب دیگر با لینکی با پهنای باند 25 Mbps و تأخیر انتشار 200 ms وصل شده است. یک لینک 1 Gbps هر میزبان و هر Cache (در صورت وجود) را به یکی از این مسیریاب‌ها وصل می‌کند. تأخیر انتشار این لینک، که در یک شبکه‌ی محلی (LAN) قرار دارد، در عمل ناچیز است. تمام بسته‌های شبکه، 10000 بیت هستند.



الف) تأخیر انتها به انتها برای ارسال یک بسته از سرور به میزبان چقدر است؟ در این قسمت فرض کنید هیچ مکانیزم Cache وجود نداشته و تأخیر صف نیز در لینک‌ها و تأخیر پردازش بسته‌ها در مسیریاب‌ها برابر صفر است. تأخیر انتقال یک بسته را به صورت زیر محاسبه می‌کنیم.

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

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 میزبان مستقیماً درخواست‌های فایل‌های خود را به سرور می‌فرستد (به عبارت دیگر از Cache‌ها استفاده نمی‌کند). همچنین تنها یکی از کلاینت‌ها درخواست ارسال می‌کند. در این صورت، بیشینه‌ی نرخ ارسال سرور به یک میزبان چقدر است؟

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

ج) بار دیگر فرض کنید تنها یک میزبان فعال بوده، اما این بار از Caching استفاده می‌کنیم. تمامی درخواست‌ها ابتدا به Local Cache Server ارسال می‌شوند. ۵۰٪ درخواست‌ها توسط Cache Server با موفقیت پاسخ داده می‌شوند. در این صورت، حداکثر نرخ دریافت داده‌ی host چقدر خواهد بود؟

در این صورت نیمی از درخواست‌ها با نرخ 25Mbps، و نیمی دیگر با نرخ 1Gbps سرویس‌دهی می‌شوند. در نتیجه:

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

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



دانشگاه مهندسی کامپیوتر

درس شبکه‌های کامپیوتری، نیم سال دوم سال تحصیلی



دانشگاه صنعتی امیرکبیر
(پاسی تکنیک تهران)

همچنان لینک با کمینه‌ی ظرفیت، لینک 25Mbps خواهد بود، که میان hostها به اشتراک گذاشته نشده است، پس جواب همچنان همان جواب قسمت قبل خواهد بود.