دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

عنوان:

شبکه‌های کامپیوتری - تمرین دوم

Computer Networks - HW2

شماره درس

۴۰۴۴۳

تاریخ تحویل

۱۴۰۴/۰۲/۱۲

مباحث

فصل سوم - لایه انتقال

استاد درس

دکتر سید امیر مهدی صادق زاده مسگر

نیم سال دوم سال تحصیلی ۱۴۰۴-۱۴۰۳

پیش از پاسخ به سوالات، به نکات زیر توجه فرمایید.

- خروجی تمرین شما می‌بایست یک فایل ZIP باشد.
- پاسخ‌های بخش عملی و نظری می‌بایست طبق استاندارد زیر در فایل PDF آورده شده باشد.
- لازم به ذکر است که اگر در سوالات بخش عملی از شما کدی خواسته شود آن را در دارکتوری‌های مجزا، طبق استاندارد زیر قرار دهید.

```
CN_HW#_STDID.pdf <--|
(DIR)  Practical Section Codes <--|
    (DIR) Q# <--|
        Server.py <--|
        Client.py <--|
        Report.pdf <--|
        Others <--|
```

- اشکالات یا ابهامات خود را از طریق تالار پرسش و پاسخ در نظر گرفته شده برای تمرین مطرح نمایید.
- هر گونه نسخه‌برداری از تمرین‌های دیگران منجر به از دست رفتن نمره تمرین خواهد شد.
- در مجموع ۱۰ روز تاخیر مجاز خواهید داشت که برای هر تمرین ۳ روز را می توانید استفاده کنید.
- در صورت استفاده از هر گونه منبع برای پاسخ به سوالات، ذکر اسم و نشانی دقیق و آدرس دسترسی به صفحه مورد نظرالزامی است.
- بارم‌بندی سوالات به قرار زیر است.

بخش	سوال	بارم
سوالات نظری	سوال ۱	۱۵
	سوال ۲	۵
	سوال ۳	۲۰
	سوال ۴	۱۰
	سوال ۵	۵
	سوال ۶	۱۰
سوالات عملی	سوال ۷	۳۵
جمع نمرات		۱۰۰

سوالات نظری

۱. (۱۵ نمره) به سوالات زیر پاسخ کوتاه دهید.

(آ) (۵ نمره) فرض کنید پردازه‌ی A و B در حال ارسال ترافیک با پروتکل لایه ۷ام HTTP به پردازه‌ی C هستند. در این ارتباطات، پورت مربوط به هر کدام از پردازه‌ها را مشخص کنید.

(ب) (۵ نمره) آیا دو پردازه‌ی روی یک هاست، می‌توانند پورت یکسانی داشته باشند؟ چرا؟ آیا این مورد استثنائی هم دارد؟

(ج) (۵ نمره) با توجه به اینکه در برخی پروتکل‌های لایه‌ی لینک، مانند Ethernet، روشی برای تشخیص خطا قرار دارد چه نیازی به checksum در پروتکل UDP داریم؟ توجه کنید که می‌خواهیم پیام برنامه‌ها به صورت end-end بی‌خطا برسد و ممکن است خطاهایی در روترها و لینک‌ها رخ دهد که توسط چک خطا در لایه لینک تشخیص داده نشود.

۲. (۵ نمره) چرا توسعه‌دهندگان یک اپلیکیشن، در بعضی موارد ترجیح می‌دهند از UDP به جای TCP استفاده کنند؟ یک مورد از اپلیکیشن‌های مورد استفاده‌ی خود را که احتمال می‌دهید از UDP استفاده می‌کند، مطرح کنید و توضیح دهید چرا UDP انتخاب بهتری بوده است.

۳. (۲۰ نمره) به سوالات زیر درباره پروتکل‌های ARQ پاسخ دهید.

(آ) (۵ نمره) در پروتکل Go Back N، اگر اندازه پنجره بیش از حد بزرگ یا کوچک باشد چه مشکلی پیش می‌آید؟ با توجه به این مشکلات اندازه پنجره بهینه چیست؟

(ب) (۱۰ نمره) در شبکه‌ای host A در حال ارسال داده به host B با RTT=100ms است. فرض کنید طول هر بسته ۱۰۰۰ بایت و نرخ انتقال داده ۱ گیگابیت بر ثانیه است، احتمال رخ دادن خطا در بسته‌ها p است ولی هیچ بسته‌ای گم نمی‌شود و خارج از ترتیب نمی‌رسد. برای هر یک از پروتکل‌های Stop and Wait، Go-Back-N و Selective Repeat (با اندازه پنجره N) کارایی را به دست آورید. فرض کنید N کوچک‌ترین مقداری‌ست که خط لوله با آن پر می‌شود؛ می‌توانید از سربار بسته‌های ACK و سرآمد بسته‌ها صرف‌نظر کنید و کارایی را در حالت بدون خطا برای Go-Back-N و Selective Repeat یک بگیرید. معیار کارایی را utilization بگیرید: درصدی از زمان انتقال که داده برای اولین بار در کانال ارتباطی ارسال می‌شود. همچنین لایه application با سرعتی داده‌های خود را به لایه انتقال می‌فرستد که پنجره همواره پر باشد. با افزایش p کارایی این سه پروتکل را با هم مقایسه کنید و بگویید برای لینک‌هایی که احتمال خطا در آن‌ها زیاد است استفاده از کدام پروتکل مناسب‌تر است.

(ج) (۵ نمره) می‌خواهیم تاثیر گم شدن بسته‌ها در تعداد بسته‌های ارسالی را برای سه پروتکل Go-Back-N، Selective Repeat و TCP (بدون delayed ACK) مقایسه کنیم. host A در حال ارسال داده به host B است. اندازه پنجره را N بگیرید و فرض کنید مقدار timeout به اندازه‌ای است که premature timeout رخ ندهد. با فرض اینکه ACK ها گم نمی‌شوند، به طور متوسط تعداد بسته‌های ارسالی توسط host A و تعداد ACK های ارسالی توسط host B در هر پروتکل چقدر است؟

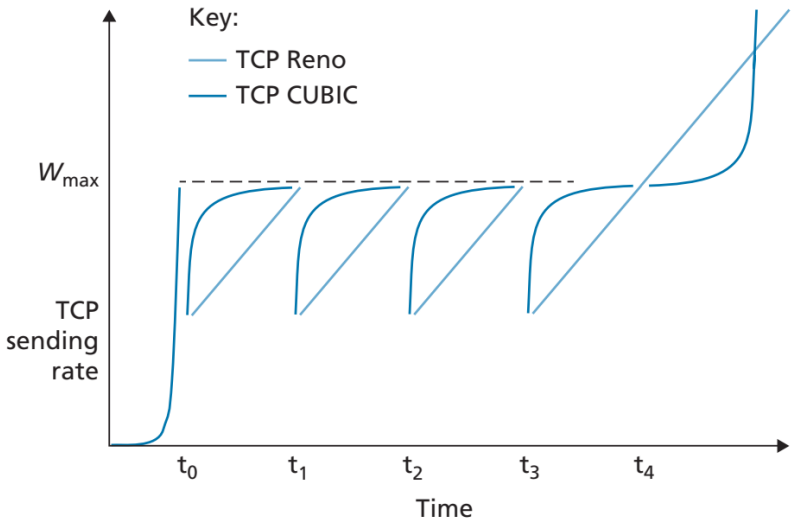
۴. (۱۰ نمره) به سوالات زیر درباره روش‌های کنترل ازدحام در لایه انتقال پاسخ دهید.

(آ) (۳ نمره) host A و host B به کمک لینکی با RTT=300ms به هم متصلند. host A به طور پیوسته به host B داده می‌فرستد و از روش AIMD برای کنترل ازدحام استفاده می‌کند. با فرض اینکه طول سگمنت‌ها ۱۵۰۰ بایت و ظرفیت لینک ۳۰Mbps است محاسبه کنید اندازه پنجره چه زمانی برای اولین بار به حداکثر مقدارش می‌رسد و میانگین throughput این انتقال چقدر است؟

(ب) (۲ نمره) اگر host A مشغول ارسال داده به host B باشد و در زمان t_1 دیگر داده‌ای برای ارسال نداشته باشد و وقفه‌ای در ارسال داده‌ها رخ دهد، زمانی که می‌خواهد داده‌های جدید لایه application را در زمان t_2 ارسال

کند، بهتر است دوباره از اول کنترل ازدحام را شروع کند یا می‌تواند از همان پارامترهای قبلی مانند اندازه پنجره و ssthresh استفاده کند؟

(ج) (۵ نمره) نمودار زیر مقایسه‌ای از عملکرد TCP Reno و TCP CUBIC زمانی که میزان ازدحام شبکه ثابت است را نشان می‌دهد. اگر اندازه پنجره‌ای که در آن congestion loss رخ می‌دهد 0.75 مقدار کنونی‌ش شود، عملکرد این دو پروتکل چگونه خواهد بود؟ اگر دو برابر شود چطور؟ برای هر حالت نمودار سرعت ارسال داده را رسم کنید.



شکل ۱: مقایسه سرعت ارسال داده برای TCP Reno و TCP CUBIC

۵. (۵ نمره) خطوط زیر که مربوط به اجرای دستوری با استفاده از ابزار dig است را توضیح دهید و بیان کنید چه اطلاعاتی در هر کدام از دستورات به دست آمده است.

(آ) (۵.۲ نمره) دستور و خروجی اول:

```
$ dig sharif.edu @4.2.2.4

... Truncated ...

;; ANSWER SECTION:
sharif.edu.  8 IN A 54.13.89.152

;; Query time: 221 msec
;; SERVER: (4.2.2.4)4#53.2.2.4 (UDP)
;; WHEN: Thurs Jan 01 00:00:00 +0330 1970
;; MSG SIZE  rcvd: 55
```

(ب) (۵.۲ نمره) دستور و خروجی دوم:

```
$ dig MX sharif.edu @4.2.2.4

... Truncated ...
```

```
;; ANSWER SECTION:
sharif.edu.  60 IN MX 10 mx02.sharif.edu.
sharif.edu.  60 IN MX 10 mx01.sharif.edu.

;; Query time: 277 msec
;; SERVER: (4.2.2.4)4#53.2.2.4 (UDP)
;; WHEN: Thurs Jan 01 00:00:00 +0330 1970
;; MSG SIZE  rcvd: 81
```

۶. (۱۰ نمره) Port scanning راهی‌ست برای بررسی وضعیت پورت‌های یک host و تشخیص سرویس‌هایی که از آن پورت‌ها استفاده می‌کنند. برای این کار port scanner درخواست‌های متعددی به پورت‌های host موردنظر می‌فرستد و بر اساس پاسخ‌های دریافت شده وضعیت پورت را تشخیص می‌دهد. می‌توان port scanning را بر اساس درخواست ارسال شده به چند دسته تقسیم کرد که شامل TCP scanning، SYN scanning، FIN scanning و UDP scanning می‌شوند.

این چهار روش را با هم مقایسه کنید و توضیح دهید هر یک از چه طریق وضعیت پورت را تشخیص می‌دهند چهار روش زیر را با هم مقایسه کنید و توضیح دهید هر یک از چه طریق وضعیت پورت را تشخیص می‌دهند.

یکی از برنامه‌هایی که قابلیت اسکن پورت‌ها را ارائه می‌دهد nmap است. فایل nmap.pcap بسته‌هایی که برای اسکن پورت‌های ۲۵، ۸۰، ۴۴۳ و ۵۳ رد و بدل شده را در بر دارد. با توجه به پاسخ‌های دریافت شده هر کدام از این پورت‌ها کدام یک از سه وضعیت open، closed و filtered را داشته است؟

علاوه بر تشخیص وضعیت پورت، nmap می‌تواند سرویسی را که روی پورت در حال اجراست همراه با نسخه آن تشخیص دهد. توضیح دهید این کار چطور انجام می‌شود.

سوالات عملی

۷. (۳۵ نمره) پیاده‌سازی یک پروتکل ارتباطی قابل اتکا بر پایه‌ی UDP

در این تمرین، شما یک کلاینت در فایل `client.py` و یک سرور در فایل `server.py` پیاده خواهید کردید که با استفاده از نسخه‌ای بهبود یافته از پروتکل UDP با یکدیگر ارتباط برقرار می‌کنند. کلاینت از `stdin` ورودی می‌گیرد که این ورودی، نام یک فایل است. این نام باید برای سرور (با استفاده از پروتکل ارتباطی پیاده شده) ارسال شود. در پاسخ، سرور فایل مورد نظر را در صورت وجود، برای کلاینت ارسال می‌کند. فایل‌های در دسترس سرور را در `./server_files` قرار دهید. سرور در صورت در دسترس نبود فایل درخواستی، باید پیام خطا برای کلاینت ارسال کند. کلاینت نیز با دریافت فایل، آن را در مسیر `./client_download_files`، در فایلی با همان نامی که به عنوان ورودی به آن داده شده ذخیره کند.

- موارد درخواستی:

(آ) (۵ نمره) یک کد کلاینت و یک کد سرور در فایل‌هایی با نام‌های گفته شده بنویسید. عملکرد کلاینت و سرور باید به نحوه‌ای که پیش‌تر توضیح داده شد باشد.

(ب) (۵ نمره) در پروتکل ارتباطی خود، پکت‌های UDP را خودتان بسازید. به این صورت که تمام بخش‌های بسته‌ی شامل Header و Payload را ایجاد کرده و این بسته را ارسال کنید.

(ج) (۲۰ نمره) ارتباط کلاینت و سرور نیاز به یک روال برای افزایش اتکاپذیری دارد. می‌توانید با استفاده از مکانیزم ارسال `ACK` و `NACK` این اتکاپذیری را فراهم کنید. طراحی این بخش به عهده‌ی شماست و محدودیتی در این مورد وجود ندارد. توصیه می‌شود از روش‌هایی که با آن‌ها در درس آشنا شده‌اید استفاده کنید. نحوه‌ی عملکرد طراحی خود را توضیح دهید و بیان کنید چگونه روش مورد استفاده‌ی شما اتکاپذیری را افزایش می‌دهد. در پروتکل ارتباطی خود، مقدار checksum را هم در سرور و هم در کلاینت بررسی کنید و در صورت پیدا کردن مشکل، بروز خطا را به فرستنده‌ی بسته اطلاع دهید تا متناسب با طراحی بخش قبلتان، بسته دوباره ارسال شود. برای checksum می‌توانید از استاندارد RFC-۷۶۸ ایده بگیرید. اگر از روشی غیر از روش استاندارد checksum را پیاده کردید، کارایی آن را توضیح دهید. بعد از ۵۰۰ میلی ثانیه اگر `ACK` دریافت نشد، فریم ارسالی باید به عنوان فریم timeout شده در نظر گرفته شود.

(د) (۱۰ نمره) بسته‌هایی که از یک threshold مشخص بزرگ‌تر هستند، باید به چند بسته‌ی کوچک‌تر شکسته شوند و هر بسته جداگانه برای کلاینت ارسال شود. این ترشولد را ۲۰۴۸ در نظر بگیرید.

- نکات مهم:

— با شبیه‌سازی یک لینک غیر قابل اتکا، پروتکل ارتباطی خود را آزمایش کرده و گزارشی از آن را همراه کدهای سرور و کلاینت ارسال کنید. توجه کنید که ارسال گزارشی که بخش‌های مختلف پروتکل را توضیح دهد، شرط لازم دریافت نمره‌ی سوال علمی‌ست.

— توجه کنید که برای پیاده‌سازی تمامی بخش‌های خواسته شده، استفاده از روش ارتباطی دیگری غیر از پروتکل پیاده‌سازی شده‌ی خود (نظیر ارتباط از طریق `Filesystem` یا شکل‌دهی یک ارتباط `TCP` موازی و ...) مجاز نیست.

— حتما نکات مهم پیاده‌سازی خود در گزارش تمرین یادداشت کنید. می‌توانید برخی از نکات را نیز در کد به صورت کامنت قرار دهید.

— در کد سرور و کلاینت خود، لاگ‌های مناسب قرار دهید تا اطلاعات مهم در مورد نحوه‌ی اجرای پروتکل ارتباطی‌تان در لاگ سرویس‌ها دیده شود. به طور مثال، حجم پیام‌های ارسالی/دریافتی، تطابق یا عدم تطابق checksum، تعداد سگمنت‌هایی که برای ارسال کامل یک پیام مورد استفاده قرار گرفته است و مواردی از این قبیل، باید در لاگ سرور و کلاینت باشد.

— برای شبیه‌سازی لینک غیر قابل اتکا، می‌توانید از دستور `tc` استفاده کنید. به عنوان مثال، برای اضافه کردن تاخیر به اینترفیس `lo`، از دستور `tc qdisc add dev lo root netem delay 2000ms` استفاده

کنید. هم‌چنین با این ابزار، می‌توانید درصدی از بسته‌ها را نیز دراپ کنید (برای اطلاعات بیشتر در مورد `tc` می‌توانید از این لینک استفاده کنید). این موارد با ابزار `iptables` نیز قابل اجرا هستند. برای شبیه‌سازی، محدودیتی در ابزارها وجود ندارد، اما در گزارش تمرین، ابزاری که از آن استفاده کردید را به همراه توضیحات لازم، معرفی کنید.

- برای تست کدهای خود، می‌توانید از فایل `test.sh` که در کنار فایل تمرین در اختیار شما قرار گرفته‌است استفاده کنید. برای استفاده از این اسکریپت، کدهای سرور و کلاینت خود را در یک دایرکتوری قرار داده و بعد از اجرا کردن سرور خود، اسکریپت را با `sudo` اجرا کنید.
- کلاینت خود را به گونه‌ای پیاده کنید که بعد از دریافت و ذخیره‌سازی فایل، Terminate شود. این مورد برای استفاده از `test.sh` ضروری‌ست.
- توصیه می شود برای پیاده‌سازی این تمرین، از سیستم عامل لینوکس استفاده کنید. ممکن است در بخش‌هایی از پیاده‌سازی روی سیستم‌های مک مشکلاتی پیش بیاید. هم‌چنین می‌توانید از محیط‌های داکری نیز برای این کار بهره بگیرید