به نام خداوند بخشنده‌ی مهربان

دانشگاه صنعتی شریف
مدرس: مهدی خرازی

تمرین برنامه‌نویسی دوم*
شبکه‌های رایانه‌ای

دانشکده‌ی مهندسی کامپیوتر
خزان ۱۳۹۳

۱. هدف‌ها

- آشنایی با شبکه‌های Overlay و P2P،
- آشنایی با معیارهای کیفیت شبکه،
- آشنایی با سرآیندهای Ethernet، IP، UDP و ICMP.

۲. مقدمه

در درس با شبکه‌های Overlay و کاربردهای مختلف آن‌ها آشنا شده‌اید. یکی از این کاربردها شبکه‌ی RON[۱] است. گره‌های شرکت‌کننده در شبکه‌ی RON کیفیت اتصال خود به اینترنت را تحت نظر می‌گیرند و هنگام نیاز به اتصال اینترنت، با توجه به نوع و کاربرد بسته‌های ارسالی، بهترین مسیر از بین گره‌های مختلف را انتخاب می‌کنند. بدین ترتیب یک نوع مسیریابی لایه‌ی کاربرد[۲] بین گره‌ها صورت می‌گیرد با این هدف که از مسیریابی حالت عادی بهتر عمل کند. در این تمرین می‌خواهیم یک شبکه‌ی ساده شبیه به شبکه‌ی RON پیاده‌سازی کنیم.

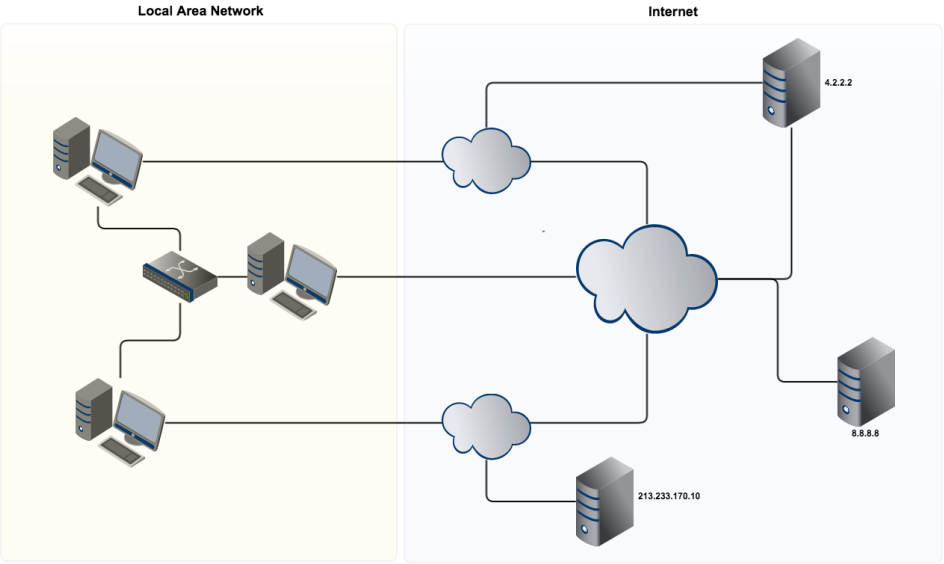
*با سپاس از بهنام مومنی، علی فتّاح‌المنان، عرفان عبدی، سعید محلوجی‌فر، هادی ذوالفقاری و مهران خلدی

[۱] Resilient Overlay Networks

[۲] Application Layer

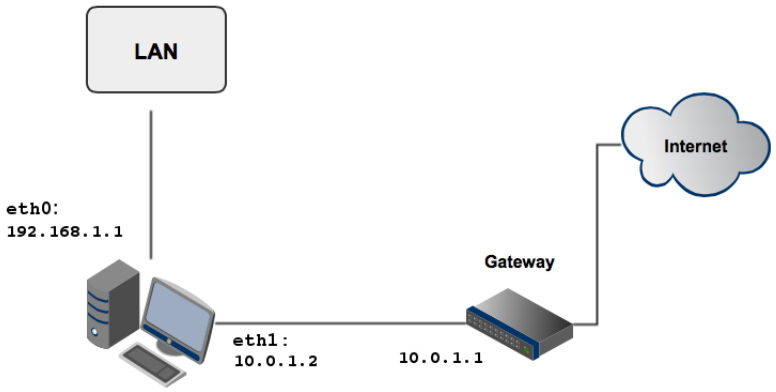
۳. توپولوژی شبکه

به گره‌های شرکت‌کننده در این شبکه، همتا[۳] می‌گوییم. هر همتا، مستقل از همتاهای دیگر، یک اتصال به شبکه‌ی اینترنت دارد. علاوه بر این، همه‌ی همتاها از طریق یک شبکه‌ی محلی[۴] به یکدیگر متصل می‌باشند.



شکل ۱: توپولوژی کلی شبکه

هر همتا دو رابط شبکه[۵] دارد. یک رابط به شبکه‌ی محلی متصل است و رابط دیگر به یک درگاه[۶] برای ارتباط با اینترنت.



شکل ۲: رابط‌های شبکه‌ی همتا

[۳] Peer
[۴] Local Area Network (LAN)
[۵] Network Interface
[۶] Gateway

۴. توضیح تمرین

هر همتا می‌تواند تعدادی برنامه‌ی کاربردی تحت شبکه اجرا کند که با کارگزارهایی در اینترنت ارتباط برقرار می‌کنند. هر یک از این برنامه‌ها برای اجرای مناسب، نیازهای خاص خود را از کیفیت شبکه دارند. برای مثال، برنامه‌های دانلود فایل نیاز به پهنای باند بالا دارند در حالی که برنامه‌هایی مانند بازی‌های شبکه‌ای نیاز به تاخیر کم شبکه دارند و پهنای باند برای آنها اهمیت کمتری دارد.

هر همتا به طور مستقل از سایر همتاها به اینترنت دسترسی دارد و کیفیت دسترسی همتاهای مختلف به کارگزارهای مختلف متفاوت است. برای مثال، ممکن است یک همتا به یک کارگزار، اتصالی با تاخیر زیاد داشته باشد در حالی که اتصال همتای دیگر به این کارگزار تاخیر کمتری داشته باشد. علاوه بر این ممکن است به دلایل مختلف مانند محدودیت‌های سازمانی، یک همتا به بعضی از کارگزارها دسترسی نداشته باشد.

همتاها می‌خواهند با همکاری یکدیگر و با بهره‌گیری از شبکه‌ی محلی بین خود، بهترین مسیر را از میان اتصالات همتاهای مختلف برای ارسال بسته‌های خود انتخاب کنند. بدین ترتیب در صورت بالا بودن تاخیر اتصال خود به اینترنت، یک همتا می‌تواند بسته‌ی خود را به همتای دیگر ارسال کند و آن همتا بسته را به کارگزار مربوطه ارسال می‌نماید. تاخیر ارسال بسته‌ها در شبکه‌ی محلی ناچیز است و همچنین همه‌ی بسته‌های ارسالی در شبکه‌ی محلی به همتای مقصد می‌رسند.

شما باید با استفاده از چارچوب پرتو، برنامه‌ی مربوط به همتاهای این شبکه را پیاده‌سازی کنید. برای این تمرین دو مشخصه برای سنجیدن کیفیت اتصالات در نظر گرفته می‌شوند: تاخیر شبکه و نرخ از دست دادن بسته‌ها[۷]. همچنین دو برنامه‌ی کاربردی فرضی در همتاهای شما باید اجرا شوند: `DSA` [۸] که برای اجرای مناسب نیاز به تاخیر کم دارد و `LSA` [۹] که برای اجرای مناسب نیاز به نرخ از دست دادن بسته‌ی پایین دارد.

پیاده‌سازی شما به سه بخش اصلی تقسیم می‌شود. ابتدا همتاها باید از وضعیت اتصال خود به کارگزارها با خبر شوند و با ارتباط برقرار کردن با کارگزارها، پارامترهای مختلف اتصال خود را بسنجند. در قدم بعد، همتاها باید مشخصات به‌دست آمده را به بقیه‌ی همتاها اطلاع دهند. در آخر نیز همتاها باید قابلیت اجرای دستورات گفته شده را داشته باشند و برای هر دستور، بسته‌ها را از طریق بهترین مسیر به‌دست آمده در مرحله‌ی قبل ارسال نمایند. در ادامه به توضیح جزییات مربوط به هرکدام از این بخش‌ها می‌پردازیم.

[۷] Packet Loss Rate
[۸] Delay Sensitive Application
[۹] Loss Sensitive Application

۵. جمع‌آوری مشخصات اتصال

قدم اول جمع‌آوری مشخصات کیفیت اتصال همتا به کارگزارها است. برای این‌کار، همتا به صورت مداوم با پروتکل ICMP به کارگزارها بسته‌های ping ارسال می‌نماید و زمان دریافت پاسخ از هر کارگزار را مورد بررسی قرار می‌دهد. لیست کارگزارهایی که همتاها قصد اتصال به آنها را دارند در Custom Information چارچوب پرتو به شما داده خواهد شد. برنامه‌ی شما به همه‌ی این کارگزارها پیغام ping ارسال می‌کند و برای هر کارگزار، از طریق پاسخ‌هایی که به ازای بسته‌های ping خود می‌گیرد، پارامترها را محاسبه کرده و ذخیره می‌کند. همان‌طور که گفته شد، شما کافیست دو مشخصه‌ی شبکه را ارزیابی کنید که توضیح آنها در ادامه داده می‌شود.

۱.۵ تاخیر شبکه

برنامه‌ی شما باید برای هر کارگزار، RTT بسته‌های ارسالی به آن را محاسبه کند. بنابراین به ازای هر بسته‌ی ping که همتا می‌فرستد، زمان ارسال باید ثبت شود و پس از دریافت بسته‌ی جواب با توجه به آن زمان ارسال ثبت شده و زمان دریافت بسته، RTT محاسبه شود.

در سرآیند ICMP بسته‌های ارسالی، مقدار فیلد `identifier` را برابر با `0` قرار دهید. فیلد `sequence number` را نیز در بسته‌ی اول برابر با `0` قرار داده و به ازای هر بسته‌ای که ارسال می‌کنید، یک واحد افزایش دهید (برای هر کارگزار جداگانه `sequence number` نگه دارید و افزایش دهید). همچنین پس از سرآیند ICMP، عدد `0x12345678` را در چهار بایت بعدی قرار دهید.

با دریافت هر بسته‌ی پاسخ از سمت کارگزار، باید با میانگین گرفتن زمان دریافت پاسخ بسته با RTT قبلی مربوط به آن کارگزار، مقدار RTT را به روز رسانی کنید. برای مثال، در صورت دریافت پاسخ از کارگزار `4.2.2.4`، RTT جدید آن کارگزار به شکل زیر محاسبه می‌شود:

$$\text{مقدار جدید } \mathrm{RTT}_{4.2.2.4} = \frac{\text{مقدار قبلی } \mathrm{RTT}_{4.2.2.4} + \text{RTT بسته‌ی دریافتی}}{۲}$$

پیش از ارسال بسته‌های ping، بسته‌های ارسالی دور قبل را بررسی کنید و اگر برای بسته‌ای، پاسخی دریافت نشده است آن بسته را گم شده فرض کنید و دیگر منتظر پاسخ آن نباشید. تنها بسته‌هایی که پاسخ آن‌ها تا زمان ارسال ping بعدی دریافت می‌شوند در محاسبه‌ی RTT وارد می‌شوند.

در صورتی که ۳ بسته‌ی متوالی به یک کارگزار بدون پاسخ ماندند، وضعیت اتصال شما به آن کارگزار قطع شده فرض می‌شود و محاسبه‌ی RTT متوقف می‌شود. در این حالت شما باید به ارسال بسته‌های ping ادامه دهید و در صورت دریافت پاسخ، وضعیت به حالت عادی برمی‌گردد و به محاسبه RTT طبق عبارت بالا ادامه می‌دهید. هر بار که از وضعیت قطع شده در می‌آیید، محاسبه‌ی RTT را از اول شروع کرده و مقادیر قبلی را کنار بگذارید. همچنین برای بسته‌ی اولی که دریافت می‌کنید نیازی به استفاده از عبارت بالا نیست و مقدار RTT را برابر با مقدار RTT آن بسته قرار دهید. هنگام دریافت بسته‌های بعدی، RTT طبق عبارت بالا حساب می‌شود. در ابتدای اجرای برنامه،

وضعیت اتصال به همه‌ی کارگزارها را قطع شده در نظر بگیرید.

۲.۵. نرخ از دست دادن بسته‌ها

پارامتر دومی که باید محاسبه کنید، نرخ از دست دادن بسته‌ها می‌باشد که عبارت است از:

$$\text{نرخ از دست دادن بسته‌ها} = ۱ - \frac{\text{تعداد پاسخ‌های دریافتی از کارگزار}}{\text{تعداد بسته‌های ارسالی به کارگزار}}$$

در صورت قطع شدن اتصال به یک کارگزار (پس از ۳ ping ناموفق)، محاسبه‌ی نرخ از دست دادن بسته‌ها را متوقف کنید و بسته‌هایی که در این حالت به مقصد نمی‌رسند را در محاسبه وارد نکنید. در این حالت، نرخ گم شدن بسته‌ها باید برابر ۱ گزارش داده شود. پس از برقراری مجدد اتصال، محاسبه را ادامه دهید (مقادیر قبل از قطع شدن نیز همچنان در نظر گرفته می‌شوند و کنار گذاشته نمی‌شوند). مثال زیر نمونه‌ای از رخدادهایی که ممکن است برای یک کارگزار اتفاق بیفتد را نشان می‌دهد:

Time		↑	↓	Loss	
0	Send Ping #0	0	0	1.00	اتصال در ابتدا قطع است
30	Send Ping #1	0	0	1.00	
32	Receive Reply #1	1	1	0.00	با دریافت پاسخ، اتصال وصل می‌شود
60	Send Ping #2	2	1	0.50	
61	Receive Reply #2	2	2	0.00	
90	Send Ping #3	3	2	0.33	
91	Receive Reply #3	3	3	0.00	
120	Send Ping #4	4	3	0.25	
150	Send Ping #5	5	3	0.40	
152	Receive Reply #5	5	4	0.20	
180	Send Ping #6	6	4	0.33	
210	Send Ping #7	7	4	0.42	
240	Send Ping #8	8	4	0.50	
270	Send Ping #9	8	4	1.00	وضعیت اتصال قطع می‌شود
300	Send Ping #10	8	4	1.00	
330	Send Ping #11	8	4	1.00	
332	Receive Reply #11	9	5	0.45	اتصال دوباره وصل می‌شود

جدول ۱: نمونه‌ای از محاسبه‌ی نرخ گم شدن بسته‌ها برای یک کارگزار

در مثال بالا، ستون ↑ تعداد بسته‌های ارسالی و ستون ↓ تعداد بسته‌های دریافتی که تا آن زمان در محاسبه وارد شده‌اند را نشان می‌دهند. ردیف‌های با رنگ قرمز مربوط به بسته‌های گم شده می‌باشند (بسته‌هایی که تا زمان ارسال بعدی، پاسخی به ازای آن‌ها دریافت نشده است). همچنین زمان‌هایی که با رنگ نارنجی نشان داده شده است مربوط به زمان‌هایی است که اتصال به کارگزار قطع می‌باشد.

ورودی و خروجی

در صورت وارد شدن دستور `ping` ، برنامه‌ی شما باید یک دور از ارسال بسته‌های ping به کارگزارها را اجرا کند. به این معنی که ابتدا برای هر کارگزار بررسی کنید که آیا بسته‌ی دور قبل گم شده است و در صورتی که این سومین بسته‌ی گم شده‌ی متوالی به آن کارگزار بود، آن کارگزار را قطع شده در نظر بگیرید. سپس به همه‌ی کارگزارها بسته‌ی ping جدید ارسال کنید.

در صورت وارد شدن دستور `stats` در ورودی شما باید پارامترهای مربوط به کارگزارها را در خروجی چاپ کنید. به ازای هر کارگزار خط زیر را چاپ کنید

```
<Server IP> <RTT> <Loss Rate> (<Sent> <Received>)
```

`Server IP` : آدرس IP کارگزار.
`RTT` : مقدار RTT به میلی‌ثانیه. در صورت قطع بودن ارتباط با آن کارگزار، عبارت `INF` نمایش داده شود.
`Loss Rate` : نرخ گم شدن بسته‌ها برای آن کارگزار به صورت عدد بین ۰ تا ۱ (تا دو رقم اعشار).
`Sent` : تعداد بسته‌های ارسالی وارد شده در محاسبه‌ی نرخ گم شدن (در صورت وصل بودن اتصال).
`Received` : تعداد بسته‌های دریافتی وارد شده در محاسبه‌ی نرخ گم شدن (در صورت وصل بودن اتصال).

```
> stats
8.8.8.8 150 0.25 (4 3)
4.2.2.4 INF 1.00
```

۶. تبادل مشخصات اتصال

همتاها باید اطلاعات به‌دست آمده از وضعیت اتصال خود به کارگزارها را به بقیه‌ی همتاها ارسال نمایند. بدین منظور هر همتا باید پیغامی در شبکه‌ی محلی broadcast کنند که در آن اطلاعات مربوطه وجود دارد.

همتاها باید بسته‌های UDP بسازند که در آن مشخصات اتصال خود به اینترنت وجود دارد و این بسته‌ها را در شبکه‌ی محلی broadcast کنند. پس از سرآیند UDP در این بسته‌ها، به ازای هر کارگزار، مشخصات اتصال به آن را با ساختار زیر قرار دهید.

0 15	16 31
Server IP	
RTT	
Sent Count	Received Count

تنها مشخصات کارگزارهایی را در بسته قرار دهید که همتا به آن‌ها متصل است. کارگزارهایی که وضعیت آن‌ها قطع است، برای سایر همتاها ارسال نمی‌شوند. پس از سرآیند UDP، به ازای هر کارگزار ۱۲ بایت به شکل بالا می‌آید. ترتیب کارگزارها در این بسته‌ها به همان ترتیبی باشد که در Custom Information می‌خوانید. در فیلد `Server IP` آدرس کارگزار را قرار دهید، در فیلد `RTT` مقدار RTT حساب شده برای کارگزار را قرار دهید و در فیلدهای `Sent Count` و `Received Count` به ترتیب تعداد بسته‌های ارسالی و تعداد بسته‌های دریافتی را قرار دهید. پس از قرار دادن اطلاعات همه‌ی کارگزارها، ۴ بایت صفر قرار دهید تا پایان بسته مشخص شود. همچنین پورت مبدا و مقصد را در سرآیند UDP برابر با `5000` قرار دهید. نیازی به محاسبه‌ی checksum در سرآیند `UDP` نیست و مقدار آن را صفر بگذارید.

همتاهای دیگر لازم است با دریافت این پیغام‌ها، جداولی بسازند که در آن مسیر با کمترین تاخیر و همچنین مسیر با کمترین نرخ از دست دادن بسته‌ها برای اتصال به هر یک از کارگزارها مشخص باشد. در ابتدای کار که همتا از همتاهای دیگر بسته‌ای دریافت نکرده است، بهترین مسیر برای اتصال به هر کارگزار از طریق اتصال خود به اینترنت (از طریق ارسال بسته به درگاه اینترنت خود) خواهد بود.

کارگزار	مسیر با تاخیر کم		مسیر با نرخ گم شدن کم	
	آدرس گره‌ی بعدی	RTT مسیر	آدرس گره‌ی بعدی	نرخ گم شدن
8.8.8.8	10.0.1.1	200ms	10.0.1.1	0.3
4.2.2.4	10.0.1.1	150ms	10.0.1.1	0.3

جدول ۲: جدول مسیریابی قبل از دریافت پیغام از سایر همتاها

در این جدول آدرس گره‌ی بعدی در مسیر (next-hop) نوشته شده است. به عنوان مثال، برای ارسال بسته‌ای به کارگزار `8.8.8.8` از طریق مسیر با کمترین تاخیر، به خانه‌ی مربوطه در جدول نگاه می‌کنیم و بسته را به آن آدرس ارسال می‌نماییم. از آنجا که تاکنون بهترین مسیر برای همه‌ی کارگزارها اتصال خود همتا به اینترنت است، همه‌ی بسته‌ها به درگاه اینترنت مربوط به آن همتا با آدرس `10.0.1.1` ارسال می‌شوند.

با دریافت بسته‌ها و اطلاع پیدا کردن از اتصال همتاهای دیگر، جدول به‌روزرسانی می‌شود. برای مثال فرض کنید از یک همتای دیگر با آدرس `192.168.1.2` اطلاعات زیر را در مورد اتصال آن به کارگزارها به‌دست آوردیم:

کارگزار	RTT	نرخ گم شدن بسته‌ها
8.8.8.8	100ms	0.6
4.2.2.4	200ms	0.2

جدول ۳: اطلاعات دریافتی از اتصال همتا به کارگزارها

حال باید جدول خود را با توجه به این اطلاعات به‌روزرسانی کنیم. در صورتی که این اطلاعات مسیر بهتری به یک کارگزار را نشان دهند آن را جای‌گزین مسیر موجود در جدول پیشین می‌کنیم.

در مثال بالا، اتصال فعلی ما به کارگزار `8.8.8.8` دارای تاخیر ۲۰۰ میلی ثانیه می‌باشد در حالی که همتای `192.168.1.2` دارای اتصالی با تاخیر ۱۰۰ میلی ثانیه می‌باشد. همچنین اتصال این همتا به کارگزار `4.2.2.4` دارای نرخ از دست دادن بسته‌ی کمتری می‌باشد. این دو مورد را در جدول جای‌گزین می‌کنیم:

کارگزار	مسیر با تاخیر کم		مسیر با نرخ گم شدن کم	
	آدرس گره‌ی بعدی	RTT مسیر	آدرس گره‌ی بعدی	نرخ گم شدن
8.8.8.8	192.168.1.2	100ms	10.0.1.1	0.3
4.2.2.4	10.0.1.1	150ms	192.168.1.2	0.2

جدول ۴: جدول مسیریابی پس از به روز رسانی

هنگام مقایسه‌ی مقدار موجود در جدول با مقدار جدید، در صورت برابر بودن دو مقدار، آن خانه از جدول را تغییر ندهید و اولویت را به مقداری بدهید که از قبل در جدول وجود داشته است.

علاوه بر زمان‌هایی که از همتاها پیغام دریافت می‌کنیم، هر بار که از کارگزاری پاسخ بسته‌ی ping دریافت می‌کنیم نیز باید جدول مسیریابی خود را به‌روزرسانی کنیم. در صورتی که پس از دریافت پاسخ و محاسبه‌ی تاخیر و نرخ گم شدن، اتصال خود همتا مسیر بهتری از مسیر موجود در جدول باشد، آن را جای‌گزین می‌کنیم.

یک همتا ممکن است پس از مدتی از شبکه خارج یا قطع شود. در این صورت اگر سایر همتاها در جدول‌های خود این همتا را به عنوان یک مسیر خوب داشته باشند، دچار مشکل خواهند شد. برای مقابله با این مشکل ما مدت اعتبار هر خانه از جدول که مربوط به همتای دیگری است را برابر با فاصله‌ی بین دو ارسال متوالی اطلاعات خود به سایر همتاها می‌گذاریم. پس از هر بار ارسال اطلاعات به همتاهای دیگر، خانه‌هایی از جدول را که مربوط به همتاهای دیگر هستند، حذف می‌کنیم تا در صورت قطع شدن آن همتاها، دیگر بسته‌ای به آن‌ها ارسال نشود.

بنابراین، بعد از ارسال اطلاعات اتصال خود به سایر همتاها، جدول خود را به حالت اول آن برمی‌گردانیم به طوری که همه‌ی مسیرها از طریق درگاه اینترنت خود باشد. علاوه بر این، قبل از برگرداندن خانه‌های جدول به مقادیر اولیه‌ی آن‌ها، آدرس گره‌ی بعدی موجود در جدول برای مسیرها را ذخیره می‌کنیم و در این دوره‌ی کار، یعنی تا ارسال بعدی مشخصات اتصال خود، از این مسیرها استفاده می‌کنیم.

در مثال قبل، پس از ارسال مشخصات اتصال خود به همتاها، جدول مسیریابی به حالت اولیه‌ی آن، مانند جدول ۲، برمی‌گردد و ما اطلاعات زیر را از جدول مسیریابی ذخیره می‌کنیم و تا زمان ارسال بعدی از این مسیرها استفاده می‌کنیم.

کارگزار	گره‌ی بعدی مسیر با تاخیر کم	گره‌ی بعدی مسیر با نرخ گم شدن کم
8.8.8.8	192.168.1.2	10.0.1.1
4.2.2.4	10.0.1.1	192.168.1.2

جدول ۵: مسیرهای ذخیره شده برای استفاده در یک دوره

بنابراین در هر دوره، ما مسیرهای مربوط به دور بعد را به‌دست می‌آوریم. در صورتی که از یک همتای دیگر اطلاعاتی دریافت کردیم که باعث تغییر در جدول شد، آن تغییر را در جدول لحاظ می‌کنیم ولی تا زمان ارسال بعدی خود از آن مسیر استفاده نخواهیم کرد. در صورتی که یک همتا از شبکه قطع شود، اطلاعات مربوط به آن پس از مدتی از جدول همتاهای دیگر حذف خواهد شد و از آن‌جا که همتای قطع شده دیگر اطلاعات خود را به سایر همتاها ارسال نمی‌کند، دیگر در جدول‌ها حضور نخواهد داشت.

ورودی و خروجی

در صورتی که دستور `advertise` در ورودی وارد شد، برنامه‌ی شما باید مشخصات اتصال خود را بر روی شبکه‌ی محلی broadcast کند. همچنین مسیرهای موجود در جدول باید از جدول حذف و ذخیره شوند و تا زمان وارد شدن مجدد این دستور، از آن‌ها استفاده شود.

در صورت وارد شدن دستور `dtable` در ورودی برنامه‌ی شما باید جدول مسیرهای با تاخیر کم به کارگزارها را در خروجی چاپ کند. به ازای هر کارگزار خط زیر باید نمایش داده شود.

```
<Server IP> <D-Current> <D-NextHOP> <D-RTT>
```

`Server IP` : آدرس کارگزار.

`D-Current` : آدرس گره‌ی بعدی برای مسیر با کمترین تاخیر به کارگزار که در حال حاظر برای این دوره مورد استفاده قرار می‌گیرد.

`D-NextHop` : آدرس گره‌ی بعدی برای مسیر با کمترین تاخیر به کارگزار که در جدول مسیریابی وجود دارد.

`D-RTT` : مقدار RTT مربوط به مسیر با کمترین تاخیر به کارگزار در جدول مسیریابی.

```
> dtable
8.8.8.8  192.168.1.2  192.168.1.2  100
4.2.2.4  10.0.1.1     10.0.1.1     150
```

در صورت وارد شدن دستور `ltable` در ورودی برنامه‌ی شما باید جدول مسیرهای با نرخ گم شدن کم به کارگزارها را در خروجی چاپ کند. به ازای هر کارگزار خط زیر باید نمایش داده شود.

```
<Server IP> <L-Current> <L-NextHOP> <L-Loss>
```

`L-Current` : آدرس گره‌ی بعدی در مسیر با کمترین نرخ گم شدن بسته‌ها به کارگزار که در حال حاظر برای این دوره مورد استفاده قرار می‌گیرد.

`L-NextHop` : آدرس گره‌ی بعدی در مسیر با کمترین نرخ گم شدن بسته‌ها که در جدول مسیریابی وجود دارد.

`L-Loss` : نرخ گم شدن بسته‌ها در مسیر با کمترین نرخ که در جدول مسیریابی وجود دارد (تا دو رقم اعشار).

```
> ltable
8.8.8.8  192.168.1.2  10.0.1.1     0.30
4.2.2.4  192.168.1.2  192.168.1.2  0.20
```

در این دستورها در صورتی که مسیری به کارگزار وجود ندارد (ارتباط تمامی همتاها با کارگزار قطع باشد) مقدار آدرس گره‌ی بعدی را آدرس درگاه خود بگذارید و مقدار RTT را `INF` و نرخ گم شدن بسته‌ها را `1.00` گزارش دهید.

۷. ارسال بسته‌ها

در این مرحله کاربر از شما می‌خواهد که برنامه‌های کاربردی را اجرا کنید. شما هنگام ارسال بسته‌های مربوط به این برنامه‌ها باید بهترین مسیر را با توجه به دستور در حال اجرا انتخاب کنید.

شما باید دو دستور `dsa` و `lsa` را پشتیبانی کنید. برای هر دو دستور کافیست بسته‌ی UDP بسازید و با توجه به جداول ساخته شده در مرحله‌ی قبل، بسته را از طریق مناسب‌ترین مسیر به کارگزار ارسال کنید. کارگزار نیز در جواب به شما یک بسته‌ی UDP برمی‌گرداند. برای ارسال بسته‌ی مربوط به برنامه‌ی `dsa` از مسیر با تاخیر کم و برای ارسال بسته‌ی برنامه‌ی `lsa` از مسیر با نرخ گم شدن بسته‌ی کم استفاده کنید.

در بسته‌های UDP ارسالی، پورت مقصد را برای بسته‌های `dsa` برابر با `1000` و برای بسته‌های `lsa` برابر با `2000` بگذارید. همچنین شماره‌ی پورت مبدا را در ابتدا `8000` بگذارید و به ازای هر بسته‌ی `lsa` یا `dsa` که برنامه‌ی شما ارسال می‌کند، آن را یک عدد زیاد کنید (این عدد برای همه‌ی کارگزارها مشترک است و جداگانه زیاد نمی‌شود). نیازی به محاسبه‌ی checksum سرآیند UDP نیست و در فیلد مربوط به آن مقدار `0` قرار دهید. همچنین عدد `0x12345678` را به عنوان داده‌ی بسته‌ها، در چهار بایت بعد از سرآیند UDP قرار دهید.

در صورتی که بهترین مسیر برای ارسال بسته از طریق اتصال خود همتا به اینترنت باشد، باید بسته را به درگاه اینترنت ارسال نموده و در صورتی که بهترین مسیر از طریق یکی از همتاهای دیگر باشد، باید بسته را به آن همتا ارسال کنید و آن همتا بسته را برای کارگزار می‌فرستد.

توجه داشته باشید که در هر دو حالت، آدرس IP مبدا بسته را آدرس IP رابط شبکه‌ی خود که به اینترنت متصل است (رابط شماره‌ی ۱) بگذارید و بسته‌ی پاسخ مستقیما از طریق درگاه اینترنت به دست شما خواهد رسید. همچنین هنگام ارسال این بسته‌ها به درگاه یا به همتاهای دیگر، آدرس IP مقصد در بسته‌ها همان آدرس IP کارگزار است، شما از طریق شبکه‌ی محلی و با استفاده از لایه‌ی ۲ شبکه بسته را به همتای دیگر ارسال می‌نمایید. همتای دیگر نیز تنها با تغییر دادن لایه‌ی ۲ بسته، آن را برای درگاه خود ارسال می‌کند.

در صورتی که یک همتا از همتای دیگر بسته‌ای دریافت کرد، باید از طریق درگاه خود، آن بسته را برای کارگزار ارسال نماید. برای اینکه بسته‌هایی که همتاها برای تبادل مشخصات اتصالات خود بین یک‌دیگر ارسال می‌کنند با بسته‌هایی که به قصد کارگزارها ارسال می‌شوند اشتباه نشوند، می‌توانید فرض کنید که بسته‌هایی که به صورت broadcast ارسال می‌شوند مربوط به مرحله‌ی قبل (تبادل اطلاعات میان همتاها) و بسته‌هایی که به همتای خاصی ارسال می‌شوند مربوط به این مرحله می‌باشند.

همچنین برای ارسال بسته به یک همتای خاص لازم است آدرس MAC آن همتا را بدانید. آدرس MAC همتاها را می‌توانید موقع دریافت پیغام‌های broadcast از آن‌ها دریافت کنید. هنگامی که از یک همتا پیغام مربوط به اطلاعات اتصال آن به کارگزارها دریافت کردید، آدرس MAC مبدا آن بسته را به عنوان آدرس MAC آن همتا ذخیره کنید.

نمره‌ی اضافی

فرض کنید یک همتا بسته‌ای را به یک همتای میانی ارسال کند تا آن همتا برای کارگزار بفرستد. در صورتی که در بسته‌ی نهایی که به کارگزار ارسال می‌شود آدرس IP مبدا، آدرس همتای اول باشد، بسته‌ی پاسخی که از کارگزار دریافت می‌شود مستقیم به دست همتای اول خواهد رسید. این اتفاق خلاف هدف ما می‌باشد چراکه کیفیت مسیر بین کارگزار و همتای دوم بهتر است و ما می‌خواهیم برای بسته‌ی پاسخ نیز از این مسیر استفاده شود. در این قسمت باید کاری کنید که علاوه بر بسته‌های ارسالی به کارگزارها، بسته‌های پاسخ نیز از بهترین مسیر به دست همتا برسند.

برای این‌کار همتاهای میانی هنگام ارسال بسته‌هایی که از همتاهای دیگر دریافت می‌کنند باید آدرس IP مبدا را در بسته‌ها به آدرس خود تغییر دهند تا بسته‌های پاسخ نیز ابتدا به دست خود آن‌ها برسد. همچنین بسته‌های پاسخ دریافت شده در نهایت باید به دست همتای اصلی برسند بنابراین هر همتا باید برای بسته‌هایی که برای همتاهای دیگر ارسال می‌کند اطلاعات مربوطه را ذخیره کند تا هنگام دریافت پاسخ، آن را به دست همتای اصلی آن برساند. برای این‌کار، هنگام ارسال بسته‌ها برای همتاهای دیگر، شماره‌ی پورت مبدا آن را تغییر داده و با شماره‌ی پورتی که برای بسته‌های خود استفاده می‌کنید جای‌گزین کنید (همانطور که گفته شد این شماره‌ی پورت از `8000` شروع می‌شود و به ازای هر بسته یک واحد زیاد می‌شود). حال می‌توانید در یک جدول نگه دارید که هر شماره‌ی پورت استفاده شده، برای بسته‌ی خود این همتا بوده یا برای همتای دیگری بوده و در صورتی که برای همتای دیگری بوده، آدرس IP و پورت اصلی آن را نیز ذخیره کنید. هنگام دریافت پاسخ از یک کارگزار شماره‌ی پورت مقصد بسته‌ی دریافتی را با این جدول تطابق داده و در صورتی که بسته برای همتای دیگری بوده، آدرس و پورت اصلی آن را جای‌گزین کنید و برای آن همتا ارسال کنید.

ورودی و خروجی

برای اجرای یکی از دستورات کاربر عبارت زیر را در ورودی وارد می‌کند.

```
<Command> <Server IP>
```

`Command` یکی از عبارت‌های `dsa` یا `lsa` می‌باشد. پس از وارد شدن این دستور شما باید بسته‌ای بسازید و در بهترین مسیر برای برنامه‌ی خواسته شده ارسال کنید. علاوه بر این عبارت زیر را در خروجی چاپ کنید.

```
<Command> packet <Port> destined for <Server IP> sent to <NextHop IP>
```

`Command` : یکی از عبارت‌های `DSA` یا `LSA` .

`Port` : شماره‌ی پورت مبدا بسته‌ی ارسالی.

`Server IP` : آدرس IP کارگزار مقصد.

`NextHop IP` : آدرس IP گره‌ی بعدی در مسیر که یا یکی از همتاها است یا درگاه

در صورت دریافت پاسخ از کارگزار عبارت زیر را نمایش دهید:

```
<Command> packet <Port> reply received in <RTT>ms
```

`Command` : یکی از عبارت‌های `DSA` یا `LSA`.

`Port` : شماره‌ی پورت مقصد بسته‌ی دریافتی.

`RTT` : مدت زمان بین ارسال بسته به کارگزار و دریافت جواب از آن به میلی‌ثانیه. به مثال زیر توجه کنید:

```
> dsa 8.8.8.8
DSA packet 8002 destined for 8.8.8.8 sent to 192.168.1.2
DSA packet 8002 reply received in 220ms

> lsa 8.8.8.8
LSA packet 8003 destined for 8.8.8.8 sent to 10.0.1.1
LSA packet 8003 reply received in 165ms
```

در صورتی که همتا بسته‌ای از همتای دیگر دریافت کرد تا به کارگزاری ارسال شود عبارت زیر را در خروجی چاپ کنید.

```
<Command> packet forwarded to <Server IP>
```

`Command` : یکی از عبارت‌های `DSA` یا `LSA`.

`Server IP` : آدرس IP کارگزار مقصد.

در صورت پیاده‌سازی نمره اضافی، موقعی که پاسخ این بسته از کارگزار دریافت شد نیز باید عبارت زیر چاپ شود.

```
<Command> forwarded packet reply received from <Server IP>
```

`Command` : یکی از عبارت‌های `DSA` یا `LSA`.

`Server IP` : آدرس IP کارگزار.

برای نمونه:

```
DSA packet forwarded to 8.8.8.8
DSA forwarded packet reply received from 8.8.8.8
```

اینکه بسته‌ها مربوط به کدام برنامه‌ی کاربردی می‌باشند (DSA یا LSA) را می‌توانید با توجه به شماره پورت مقصد بسته‌های ارسالی به کارگزارها و پورت مبدا بسته‌های دریافتی از کارگزارها بفهمید.

۸. نکات تکمیلی

شماره‌ی رابط‌های شبکه

همان‌طور که قبلا اشاره شد، هر همتا دو رابط شبکه دارد. رابط شماره‌ی ۰ به شبکه‌ی محلی متصل است و رابط با شماره‌ی ۱ به درگاه اینترنت. دقت کنید که در هر بسته اطلاعات مربوط به رابط صحیح را قرار دهید و همچنین بسته را بر روی رابط صحیح ارسال کنید.

ساختار Custom Information

شما می‌توانید از Custom Information چارچوب پرتو آدرس درگاه اینترنت خود و کارگزارها را به‌دست آورید. در خط اول Custom Information آدرس IP درگاه و در خط دوم آدرس MAC آن وجود دارد. همچنین در خط سوم تعداد کارگزارها و در خط‌های بعدی آدرس این کارگزارها خواهد آمد. برای مثال Custom Information یک همتا ممکن است به شکل زیر باشد:

```
10.0.1.2
00:24:8C:01:01:02
3
8.8.8.8
4.2.2.4
4.4.4.4
```

۹. نکات ضروری

- در صورتیکه هر مشکل یا پرسشی داشتید که فکر می‌کنید پاسخ آن برای همه مفید خواهد بود، آن را به گروه اینترنتی درس ارسال کنید.
- از فرستادن جواب تمرین به گروه اینترنتی درس خودداری کنید.
- تمام برنامه‌ی شما باید توسط خود شما نوشته شده باشد. فرستادن کل یا قسمتی از برنامه‌تان برای افراد دیگر، یا استفاده از کل یا قسمتی از برنامه‌ی فرد دیگری، حتی با ذکر منبع، تقلب محسوب می‌شود!
- پس از اتمام کارتان لازم است که پوشه‌ی user را به همراه Makefile فشرده کرده (می‌توانید این کار را با اجرای دستور `make submit` انجام دهید) و از طریق وب‌سایت پرتو ارسال نمایید.

۱۰۹. اجرای نمونه

برای روشن‌تر شدن بخش‌های مختلف، یک اجرای نمونه (بدون در نظر گرفتن قسمت نمره‌ی اضافی) از برنامه در ادامه نشان داده شده است:

```
> stats
8.8.8.8 INF 1.00
4.2.2.4 INF 1.00
4.4.4.4 INF 1.00
> dtable
8.8.8.8 10.0.1.1 10.0.1.1 INF
4.2.2.4 10.0.1.1 10.0.1.1 INF
4.4.4.4 10.0.1.1 10.0.1.1 INF
> ltable
8.8.8.8 10.0.1.1 10.0.1.1 1.00
4.2.2.4 10.0.1.1 10.0.1.1 1.00
4.4.4.4 10.0.1.1 10.0.1.1 1.00
> ping
> ping
> stats
8.8.8.8 134 0.00 (2 2)
4.2.2.4 86 0.50 (2 1)
4.4.4.4 INF 1.00
> lsa 8.8.8.8
LSA packet 8000 destined for 8.8.8.8 sent to 10.0.0.1
LSA packet 8000 reply received in 128ms
> dsa 4.4.4.4
DSA packet 8001 destined for 4.4.4.4 sent to 10.0.0.1
> dtable
8.8.8.8 10.0.1.1 10.0.1.1 134
4.2.2.4 10.0.1.1 10.0.1.1 86
4.4.4.4 10.0.1.1 10.0.1.1 INF
```

```
> ltable
8.8.8.8 10.0.1.1 10.0.1.1 0.00
4.2.2.4 10.0.1.1 10.0.1.1 0.50
4.4.4.4 10.0.1.1 10.0.1.1 1.00
* Received an advertisement from a peer *
> dtable
8.8.8.8 10.0.1.1 192.168.1.2 94
4.2.2.4 10.0.1.1 10.0.1.1 86
4.4.4.4 10.0.1.1 192.168.1.2 142
> ltable
8.8.8.8 10.0.1.1 10.0.1.1 0.00
4.2.2.4 10.0.1.1 192.168.1.2 0.25
4.4.4.4 10.0.1.1 192.168.1.2 0.33
> dsa 8.8.8.8
DSA packet 8002 destined for 8.8.8.8 send to 10.0.1.1
DSA packet 8002 reply received in 130ms
> advertise
> dtable
8.8.8.8 192.168.1.2 10.0.1.1 134
4.2.2.4 192.168.1.2 10.0.1.1 86
4.4.4.4 192.168.1.2 10.0.1.1 INF
> ltable
8.8.8.8 10.0.1.1 10.0.1.1 0.00
4.2.2.4 192.168.1.2 10.0.1.1 0.50
4.4.4.4 192.168.1.2 10.0.1.1 INF
> dsa 8.8.8.8
DSA packet 8003 destined for 8.8.8.8 send to 192.168.1.2
DSA packet 8003 reply received in 90ms
```