به نام خداوند بخشنده مهربان

دانشگاه صنعتی شریف
مدرس: مهدی خرازی

تمرین برنامه نویسی دوم [*]
شبکه‌های رایانه‌ای

دانشکده‌ی مهندسی کامپیوتر
بهار ۱۳۹۵

اهداف تمرین

- آشنایی با شبکه‌های P2P و شبکه‌های RON،
- آشنایی با معیارهای کیفیت شبکه و مسیریابی در لایه‌ی کاربرد،
- آشنایی با DHT Chord.

۱. تمرین تئوری

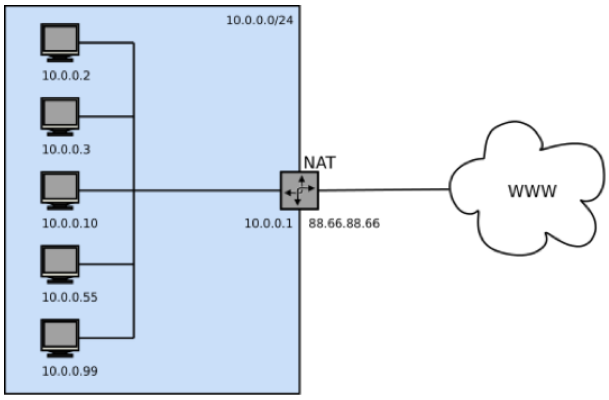
۱.۱ تحلیل ترافیک NAT

یکی از مسائل مهم در زمینه امنیت شبکه، بحث تحلیل ترافیک است. تحلیل ترافیک بدین معناست که بدون توجه به محتوای ترافیک و رمزگشایی و ... بتوانیم اطلاعاتی بدست بیاوریم. تحلیل ترافیک یعنی اینکه درباره‌ی محتوا چه اطلاعاتی می‌توانیم به دست آوریم بدون اینکه به محتوا نگاه کنیم. ارزیابی‌ها نشان می‌دهد که هزینه تحلیل ترافیک بسیار کمتر از روش‌هایی همچون رمزگشایی است. حال در ادامه یک نمونه از تحلیل ترافیک مربوط به NAT را انجام خواهید داد.

فرض کنید علاقه‌مند هستید بدانید که پشت یک NAT چند هاست یکتا وجود دارد. فرض کنید که شما اطلاعات و فیلدهای لایه‌ی IP اعم از شماره شناسه‌ی[1] هر بسته را مشاهده می‌کنید. شماره شناسه‌ای که یک هاست در ارسال اولین بسته‌ی خود تولید می‌کند یک عدد تصادفی است و شماره شناسه‌ی بسته‌های بعدی به ترتیب افزایش پیدا می‌کند. فرض کنید که تمامی بسته‌های IP که توسط هاست‌های پشت NAT تولید شده به سمت شبکه‌ی خارجی ارسال می‌شود.

[*] با سپاس از سولماز سلیمی، زینب ساسان، پیمان عزتی، رضا میرعسگر شاهی، پارسوآ خورسند

[1] identification number



شکل ۱: توپولوژی NAT

- با توجه به توضیحات ذکر شده و با فرض اینکه شما قادر هستید تمامی بسته‌ها را شنود کنید، آیا می‌توانید روشی ارائه کنید که از طریق آن بتوان تشخیص داد که پشت NAT چند هاست یکتا وجود دارد؟ با ذکر جزئیات پاسخ دهید.

- اگر شماره شناسه مربوط به هر بسته به ترتیب افزایش نیابد و تخصیص این شناسه به هر بسته به طور تصادفی صورت گیرد، آیا روشی که در سؤال اول ارائه دادید، صحیح است؟

۲.۱. آشنایی با DHT Chord

در این تمرین قرار است با جداول درهم‌سازی توزیع‌شده که استفاده‌ی فراوانی در شبکه‌های همتا به همتا [۲] دارند، آشنا شوید. همانطور که می‌دانید در شبکه‌های همتا به همتا با حجم زیادی از پرونده‌ها مواجه هستیم که مدیریت این پرونده‌ها می‌تواند به شکل متمرکز یا توزیع‌شده صورت گیرد. تمرکز اصلی بحث و موضوع DHT همان شیوه‌ی توزیع‌شده مدیریت پرونده‌ها است.

به عنوان توپولوژی ساده‌ای از DHT می‌توان حالت دایره‌ای را در نظر گرفت که گره‌ها حول این دایره به حالت ساعتگرد قرار می‌گیرند. گره‌ها در این توپولوژی ماشین‌ها هستند که جداول درهم‌سازی را نگه می‌دارند. هدف اصلی از DHT این است که وقتی به دنبال شناسه‌ی خاصی هستیم بتوانیم از این سامانه‌ی توزیع‌شده بهره ببریم و ماشین‌های دارای آن شناسه را بیابیم. حال این پرس‌وجو می‌تواند از هر یک از گره‌های شبکه انجام شود.

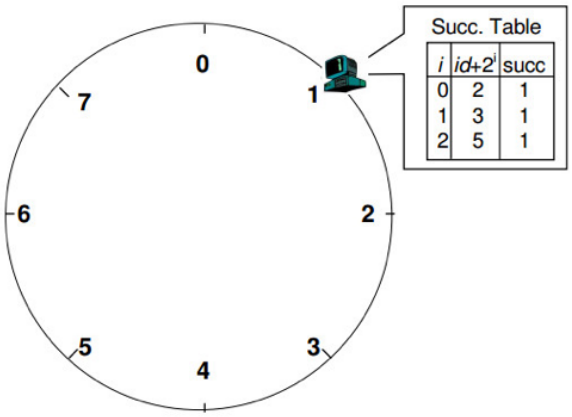
ساختار دایرهای بسیار شبیه به لیست پیوندی است که هر گره دارای یک گره جانشین (خَلَف!) است که اولین گره با شناسه‌ی بزرگ‌تر از گره مورد نظر است. فرض بر این است که در طول درس با روند پرس‌وجو از این سامانه و اضافه شدن یا کم شدن گره‌ها از آن آشنا شده‌اید. نکته‌ی جدیدی که در این تمرین به دنبال یاد گرفتن آن هستیم، finger table در هر گره از DHT Chord است.

روند پرس‌وجو در DHT Chord در حالت ساده، پس ازپرس‌وجو از یک گره، اگر شناسه‌ی مورد نظر را نداشت، ما را برای پرس‌وجو به گره جانشین خود ارجاع می‌دهد و این روند تا جایی ادامه پیدا می‌کند که به گره‌ای برسیم که

[۲] Peer to Peer

دارای شناسه‌ی مورد نظر ما باشد. برای سرعت بخشیدن به روند پرس‌وجو از جداول انگشتی در هر گره استفاده می‌شود. هر گره جدولی با m رکورد را در خود نگه می‌دارد. اگر شناسه‌ی هر گره را n در نظر بگیریم، هر رکورد این finger table برابر است با شناسه‌ی $n + ۲^i$
و گره جانشین آن.

به عنوان مثال فرض کنید m تعداد رکوردهای جدول، ۳ باشد و تنها یک گره با شناسه n=۱ در توپولوژی دایره‌ای وجود داشته باشد، finger table آن به شکل زیر خواهد بود: با اضافه یا کم شدن هر گره، هر گره باید finger



شکل ۲ : جدول نمونه

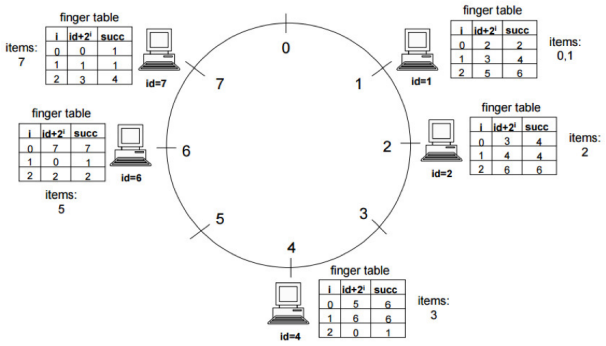
table خود را به‌روزرسانی کند. حال باید ببینیم روند پرس‌وجو چه تغییری کرده است. پرس‌وجو از یک گره برای یافتن آیتم مشخصی شروع می‌شود. گره پرس‌وجوشونده بررسی می‌کند که آیا آیتم مورد نظر را دارد یا خبر. اگر داشت، پرس‌وجو به اتمام می‌رسد. اگر گره مورد نظر شناسه‌ی آیتمی که دنبالش هستیم را نداشت، به شناسه‌های داخل finger table خود نگاه می‌کند، اگردر جدول خود آن شناسه را نداشت بزرگ‌ترین شناسه را انتخاب کرده و به گره جانشین آن، پرس‌وجو را ارجاع می‌دهد.

همچنین باید به این نکته توجه شود که در هر مرحله، گره جانشینی که پرس‌وجو به آن ارجاع داده می‌شود، شناسه‌ای بزرگ‌تر از شناسه‌ی پرس‌وجو نداشته باشد. (نکته: توجه داشته باشید بزرگ‌تر بودن شناسه‌ی هر گره نسبت به دیگری طبق همان حرکت ساعتگرد است.)

حال که با مفاهیم و کارکرد DHT Chord و finger table آشنا شدید، به سراغ سؤال و تمرین اصلی می‌رویم.

فردی شبکه‌ای نظیر به نظیر را راه‌اندازی کرده و برای مدیریت شناسه‌ها و پرس‌وجو در شبکه از DHT Chord استفاده می‌کند. البته شبکه‌ی این فرد بسیار کوچک است و توپولوژی، گره‌های آن و finger table هر گره در شکل زیر نشان داده شده است:

- اگر به دنبال آیتم ۰ باشیم و پرس‌وجو را از گره ۲ شروع کنیم، چه گره‌هایی پرس‌وجو خواهند شد؟ با جزئیات روند پرس‌وجو را توضیح دهید.



شکل ۳: توپولوژی سوال

- فرض کنید گره ۴ از دسترس خارج شود و از گره ۷ برای آیتم ۵ پرس‌وجو صورت گیرد. لیست گره‌هایی که قرار است از آن‌ها پرس‌وجو شود را بنویسید. این نکته را نیز در نظر داشته باشید که گره‌ها فرصت همگرایی و به‌روزرسانی finger table خود را، پس از حذف گره‌ی ۴ دارند.

۳.۱. آشنایی با آدرس‌های IP بدون کلاس

فرض کنید یک روتر در جدول مسیریابی خود رکوردهای (CIDR) زیر را دارد:

Address/mask	Next hop
135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
192.53.40.0/23	Router 1
default	Router 2

حال پاسخ دهید که این روتر با رسیدن بسته‌هایی با آدرس‌های زیر چه عملکردی را اتخاذ خواهد کرد؟

- 135.46.63.10
- 135.46.57.14
- 135.46.52.2
- 192.53.40.7
- 192.53.56.7

۲. تمرین عملی

۱.۲ مقدمه

در درس با شبکه‌های Overlay Network و دو کاربرد مهم این شبکه‌ها یعنی RON[۳] و جداولِ درهم‌سازی توزیع‌شده[۴] آشنا شده‌اید. مهم‌ترین هدف از ایجاد شبکه‌های RON همکاری در یک شبکه‌ی همتا به همتا[۵] به منظور افزایش کیفیت اتصال به اینترنت است. در این شبکه‌ها هر همتا هنگام نیاز به اتصال به اینترنت، با توجه به نوع و کاربرد بسته‌های ارسالی، بهترین مسیر از بین همتاهای مختلف را انتخاب می‌کند. این ایده یعنی با هدف یافتن مسیری بهتر، مسیریابیِ لایه‌ی کاربرد[۶] جایگزین مسیریابی عادی می‌شود. هدف از این تمرین پیاده‌سازی یک شبکه‌ی ساده است که عملکرد RON را شبیه‌سازی می‌کند.

۲.۲ توپولوژی شبکه

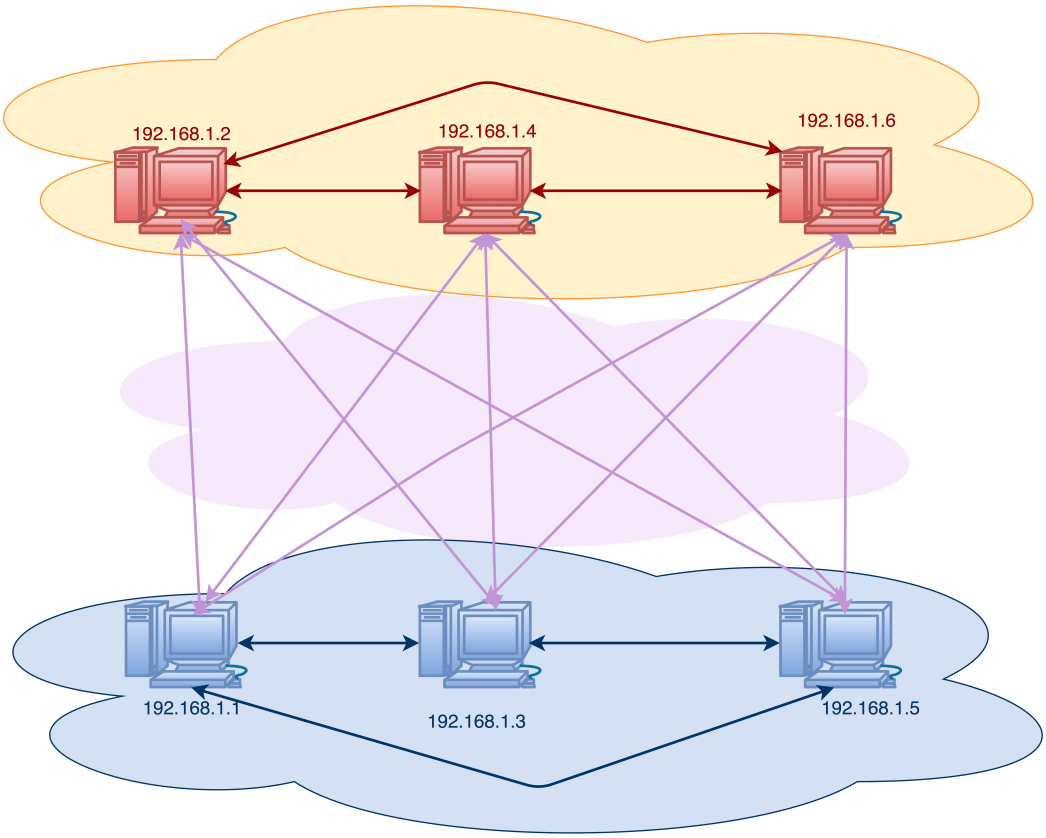
به گره‌های شرکت‌کننده در این شبکه همتا [۷] می‌گوییم. شبکه به نحوی است که اگر آدرس آی‌پی همتا زوج باشد، با همه‌ی همتاهای دیگر که آدرس آی‌پی زوج دارند در یک زیرشبکه (محلی) قرار می‌گیرد و به این ترتیب ارتباطات همتاهای زوج بدون تاخیر است.

همه‌ی همتاهای فرد هم با یکدیگر ارتباط بدون تاخیر دارند. به این ترتیب همه‌ی گره‌های این شبکه یا در شبکه‌ی زوج قرار می‌گیرند یا در شبکه‌ی فرد هستند. اما ارتباط یک همتای زوج با یک همتای فرد لزوماً بدون تاخیر یا گم شدن بسته نیست.

به طور مثال در توپولوژی نمونه که در ادامه آمده است ممکن است یک همتا با آدرس `192.168.1.1` با همتای `192.168.1.6` در شبکه‌ی مقابل با تاخیر `200 ms` اتصال برقرار کند، در صورتی که همتای دیگر فرد مثل `192.168.1.5` با همتای `192.168.1.6` با تاخیر `100 ms` ارتباط برقرار کند. با تاکید بر اینکه تاخیر بین دو گره‌ی `192.168.1.1` و `192.168.1.5` صفر است، اگر گره‌ی `192.168.1.1` بسته‌ای که می‌خواهد به همتای `192.168.1.6` ارسال کند را از راه همتای `192.168.1.5` ارسال کند، بسته با تاخیر کم‌تری به مقصد می‌رسد.

---

[۳] Resilient Overlay Networks
[۴] Distributed Hash Tables
[۵] Peer to Peer
[۶] Application Layer
[۷] Peer



تعداد گره‌های یک شبکه کم‌تر از ۳۲ گره می‌باشد.

در این‌جا هیچ گره‌ای در نقش کارگزار نیست، گره‌های فرد به دنبال سرویس‌هایی هستند که در گره‌های زوج ارائه می‌شود و گره‌های زوج از سرویس‌هایی که گره‌ی فرد ارائه می‌دهد استفاده می‌کنند.

۳.۲ توضیح تمرین

با توجه به توپولوژی مطرح شده، در این تمرین یک شبکه‌ی همتا به همتا مد نظر است که تمام گره‌های همتا در حال اجرای سرویس‌هایی هستند. هر همتا می‌تواند تعدادی برنامه‌ی کاربردی تحت شبکه را اجرا کند که همتاهای مخالف در همین شبکه ارتباط برقرار می‌کنند. هر یک از این برنامه‌ها برای اجرای مناسب، نیازهای خاص خود را از کیفیت شبکه دارند. برای مثال، برنامه‌های بارگیری پرونده [۸] نیاز به پهنای باند بالا دارند، در حالی‌که بازی‌های شبکه‌ای نیاز به تاخیر کم شبکه دارند و پهنای باند برای آن‌ها اهمیت کم‌تری دارد.

در این تمرین، شبکه به دو بخش تقسیم می‌شود؛ گره‌های فرد در یک سمت شبکه قرار دارند و به مجموعه‌ی آن‌ها شبکه‌ی آبی می‌گوییم و گره‌های زوج در سمت دیگر شبکه قرار دارند که به مجموعه‌ی آن‌ها شبکه‌ی قرمز می‌گوییم. در این شبکه‌ی همتا به همتا همه‌ی گره‌ها به یکدیگر دسترسی دارند، اما کیفیت دسترسیِ همتاهای مختلف به سایر همتاها در شبکه‌ی مخالف متفاوت است. برای مثال، ممکن است یک همتا از شبکه‌ی آبی به همتای دیگری در

[۸] Download File

شبکه‌ی قرمز اتصالی با تاخیر زیاد داشته باشد، در حالی که همتای دیگری از شبکه‌ی آبی به همان همتا در شبکه‌ی قرمز دست‌رسی با تاخیر بسیار کمتری دارد.

همتاهای هر دو شبکه می‌خواهند با همکاری یکدیگر از شبکه‌ی مقابل بهترین سرویس را بگیرند. در واقع هر همتا در یکی از شبکه‌ها، بهترین اتصال را از میان اتصالات همتاهای موجود در شبکه‌ی خود برای ارسال بسته انتخاب می‌کند و از طریق همتای انتخاب شده بسته را به شبکه‌ی مقابل ارسال می‌کند و تاخیر ارسال بسته‌ها در هر زیرشبکه ناچیز است و همچنین همه‌ی بسته‌های ارسالی در هر زیرشبکه به همتای مقصد می‌رسد.

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[۹].

همچنین دو برنامه‌ی کاربردیِ فرضی برای هر زیرشبکه در همتاهای شما باید اجرا شوند: `DSA`[۱۰] که برای اجرای مناسب نیاز به تاخیر کم دارد و `LSA`[۱۱] که برای اجرای مناسب نیاز به نرخ از دست دادن بسته‌ی پایین دارد. پیاده‌سازی شما به سه بخش اصل تقسیم می‌شود.

- ابتدا همتاها باید از وضعیت اتصال خود به همتاهای شبکه‌ی مقابل با خبر شوند و با ارتباط برقرار کردن با هر یک از همتاهای شبکه‌ی مقابل، پارامترهای مختلف اتصال خود را بسنجند.
- در قدم بعد، همتاها باید مشخصات به‌دست آمده را به بقیه‌ی همتاهای موجود در شبکه‌ی خود 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.

۴.۲. جمع‌آوری مشخصات اتصال

قدم اول جمع آوری مشخصات کیفیت اتصال همتا به همتاهای شبکه‌ی مقابل است. برای این‌کار، همتا به صورت مداوم با پروتکل ICMP به همتاهای مخالف بسته‌های ping ارسال می‌نماید و زمان دریافت پاسخ از هر همتا را مورد بررسی قرار می‌دهد. برنامه‌ی شما به همه‌ی همتاهای شبکه‌ی مقابل پیغام ping ارسال می‌کند و برای هر همتا، 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.

[۹] Packet Loss Rate

۱.۴.۲ تاخیر شبکه

برنامه‌ی شما باید برای هر همتا در شبکه‌ی مقابل، 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 ارسال ، 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 همتای شبکه‌ی مخالف 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 بسته ی پاسخ از سمت همتای شبکه‌ی مقابل، باید با میانگین گرفتن زمان دریافت پاس بسته با RTT قبل مربوط به آن همتا، مقدار RTT را به روز رسانی کنید.

فرض کنید یکی از همتاهای موجود در شبکه‌ی آبی پس از دریافت پاسخ از همتای `192.168.1.6` از شبکه‌ی قرمز، RTT جدید را برای این همتا به صورت زیر محاسبه می‌کند:

$$\text{مقدار جدید } RTT_{192.168.1.6} = \frac{\text{مقدار قبلی } RTT_{192.168.1.6} + RTT\text{بسته‌ی دریافتی}}{۲} \quad (۱)$$

پیش از ارسال بسته‌های `ping` بسته‌های ارسال شده از 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 صورت که ۳ بسته‌ی متوالی به یک همتا در شبکه‌ی مقابل بدون پاسخ ماندند، وضعیت اتصال همتای فعلی (شما) به آن همتا در شبکه‌ی مقابل 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 قبل را کنار بگذارید. همچنین برای بسته‌ی اول 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 همتاها در شبکه‌ی مقابل را قطع شده در نظر بگیرید.

۲.۴.۲ نرخ از دست دادن بسته‌ها

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

$$\text{نرخ از دست دادن بسته‌ها} = ۱ - \frac{\text{تعداد پاسخ‌های دریافتی از همتا در شبکه‌ی مقابل}}{\text{تعداد بسته‌های ارسالی به همتا در شبکه‌ی مقابل}} \quad (۲)$$

در صورت قطع شدن اتصال به یک همتا در شبکه‌ی مقابل (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 همتا در شبکه‌ی مقابل 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

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 همتا در شبکه‌ی مقابل قطع می‌باشد.

۳.۴.۲. ورودی و خروجی

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

در صورت وارد شدن دستور `stats`، در ورودی شما باید پارامترهای مربوط به هر یک از همتاهای شبکه‌ی مقابل را در خروجی چاپ کنید. به ازای هر همتا در شبکه‌ی مقابل خط زیر را چاپ کنید:

```
<Target PEER IP> <RTT> <Loss Rate> (<Sent> <Received>) connceted|disconnceted
```

`Target PEER IP` آدرس IP همتا در شبکه‌ی مقابل.

`RTT` : مقدار RTT به میلی‌ثانیه. در صورت قطع بودن ارتباط با آن همتا در شبکه‌ی مقابل، عبارت `INF` نمایش داده شود.

`Loss Rate` : نرخ گم شدن بسته‌ها برای آن همتا در شبکه‌ی مقابل 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
185.88.153.2 INF 1.00 (0 0) disconnceted
185.88.153.4 INF 1.00 (0 0) disconnceted
185.88.153.6 INF 1.00 (0 0) disconnceted
```

شکل ۴: قبل از ارسال اولین ping

```
stats
185.88.153.2 2600ms 0.50 (2 1) connceted
185.88.153.4 2600ms 0.50 (2 1) connceted
185.88.153.6 2600ms 0.50 (2 1) connceted
```

شکل ۵: پس از ارسال دو ping

۵.۲ تبادل مشخصات اتصال

همتاهای مستقر در یک زیرشبکه باید اطلاعات به‌دست آمده از وضعیت اتصال خود به همتاهای شبکه‌ی مقابل را به بقیه‌ی همتاها ارسال نمایند. بدین منظور هر همتا باید پیغامی در زیرشبکه‌ی خود broadcast کند 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 زیرشبکه‌ی خود broadcast کنند. پس از سرآیند UDP در این بسته‌ها، به ازای هر همتا در شبکه‌ی مقابل، مشخصات اتصال به آن را با ساختار زیر قرار دهید:

0 15	16 31
Target Peer IP	
RTT	
Sent Count	Received Count

تنها مشخصات همتاهایی از شبکه‌ی مقابل را در بسته قرار دهید که خود همتای فعلی به آن‌ها متصل است. همتاهای مقابل که وضعیت آن‌ها قطع است، برای سایر همتاهای زیرشبکه ارسال نمی‌شوند. پس از سرآیند UDP، به ازای هر همتا در شبکه‌ی مقابل ۱۲ بایت به شکل بالا می‌آید. ترتیب همتاها در این بسته‌ها به ترتیب آدرس‌های آی‌پی از کوچک به بزرگ باشد. در فیلد `Target Peer IP` آدرس همتا در شبکه‌ی مقابل را قرار دهید، در فیلد `RTT` مقدار RTT حساب شده برای همتا در شبکه‌ی مقابل 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 همتاهای شبکه‌ی مقابل (در صورت برقراری اتصال)، 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 در همان زیرشبکه 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 همتاهای شبکه‌ی مقابل مشخص باشد. در ابتدای کار که همتا از همتاهای دیگر زیرشبکه بسته‌ای دریافت نکرده است، بهترین مسیر برای اتصال به هر همتا در شبکه‌ی مقابل از طریق اتصال خود به همان همتا (از طریق ارسال بسته مستقیم به همتا در شبکه‌ی مقابل) است.

آدرس آی‌پی همتا در	مسیر با تاخیر کم		مسیر با نرخ گم شدن کم	
شبکه‌ی مقابل	آدرس گره‌ی بعدی	RTT مسیر	آدرس گره‌ی بعدی	نرخ گم شدن
192.168.1.2	192.168.1.1	200 ms	192.168.1.1	0.3
192.168.1.4	192.168.1.1	150 ms	192.168.1.1	0.4
192.168.1.6	192.168.1.1	200 ms	192.168.1.1	0.5

جدول ۱: 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 همتای `192.168.1.4` 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 همتاها در شبکه‌ی مقابل اتصال خود همتا به آن‌ها است، همه‌ی بسته‌ها از راه خودش ارسال می‌شوند.

با دریافت بسته‌ها و اطلاع پیدا کردن از اتصال همتاهای دیگر در زیرشبکه (در این‌جا زیرشبکه‌ی آبی)، جدول به روز رسانی می‌شود. برای مثال فرض کنید از یک همتای دیگر با آدرس `192.168.1.5` اطلاعات زیر را در مورد اتصال آن به همتاهای شبکه‌ی مقابل به‌دست می‌آوریم:

آدرس آی‌پی همتا در شبکه‌ی مقابل	RTT	نرخ گم شدن بسته‌ها
192.168.1.2	100 ms	0.6
192.168.1.4	200 ms	0.2
192.168.1.6	150 ms	0.4

جدول ۲ : اطلاعات دریافتی از اتصال‌های همتای 192.168.1.5

آدرس آی‌پی همتا در	مسیر با تاخیر کم		مسیر با نرخ گم شدن کم	
شبکه‌ی مقابل	آدرس گره‌ی بعدی	RTT مسیر	آدرس گره‌ی بعدی	نرخ گم شدن
192.168.1.2	192.168.1.5	100 ms	192.168.1.1	0.3
192.168.1.4	192.168.1.1	150 ms	192.168.1.5	0.2
192.168.1.6	192.168.1.5	200 ms	192.168.1.5	0.5

جدول ۳: جدول مسیریابی پس از دریافت پیغام از همتای 192.168.1.5

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 همتاهای زیرشبکه پیغام دریافت می‌کنید، هر بار که از همتاهای شبکه‌ی مقابل پاسخ بسته‌ی ping دریافت می‌کنید نیز باید جدول مسیریابی خود را به روز رسانی کنید. 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 کنید.

۱.۵.۲ ورودی و خروجی

در صورت که دستور `advertise` در ورودی وارد شد، برنامه‌ی شما باید مشخصات اتصال خود را بر روی زیرشبکه `broadcast` کند. گره‌های فرد اگر پیغام `advertise` از گره‌های زوج دریافت کنند باید آن را نادیده بگیرند و گره‌های زوج در صورتی که پیغام `advertise` از گره‌های فرد دریافت کنند باید آن را نادیده بگیرند.

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

```
<Target Peer IP> <D-NextHOP> <D-RTT>
```

`Target PEER IP` : آدرس IP همتا در شبکه‌ی مقابل.

`D-NextHOP` : آدرس گره‌ی بعدی برای مسیر با کمترین تاخیر به همتا در شبکه‌ی مقابل که در حال حاضر مورد

استفاده قرار می‌گیرد.

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

در صورت وارد شدن دستور `ltable` در ورودی برنامه ی شما باید جدول مسیرهای با نرخ گم شدن کم به

```
dtable
185.88.153.1    185.88.153.2    2400
185.88.153.3    185.88.153.2    5000
185.88.153.5    185.88.153.6    4400
```

شکل ۶: خروجی dtable

همتاهای شبکه‌ی مقابل را در خروجی چاپ کند. به ازای هر همتا در شبکه‌ی مقابل خط زیر باید نمایش داده شود:

```
<Target Peer IP> <D-NextHOP> <D-RTT>
```

`Target PEER IP` : آدرس IP همتا در شبکه‌ی مقابل.

`D-NextHOP` s: آدرس گره‌ی بعدی برای مسیر با کمترین نرخ گم شدن بسته‌ها به همتا در شبکه‌ی مقابل که در حال حاضر مورد استفاده قرار می‌گیرد.

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

در این دستورها در صورت که مسیری به همتای شبکه‌ی مقابل وجود ندارد (ارتباط تمام همتاها با یکی از همتاهای شبکه‌ی مقابل قطع باشد(مقدار آدرس گره‌ی بعدی را آدرس خود همتا قرار دهید و مقدار RTT را `INF` و نرخ گم شدن بسته ها را `1.00` گزارش دهید.

```
ltable
185.88.153.1    185.88.153.6    0.33
185.88.153.3    185.88.153.6    0.00
185.88.153.5    185.88.153.6    0.50
```

شکل ۷: خروجی ltable

۶.۲ 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 مرحله ی قبل، بسته را از طریق مناسب ترین مسیر به همتای هدف در شبکه‌ی مقابل 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 codelsa از مسیر با نرخ گم شدن بسته‌ی کم استفاده کنید. در بسته‌های `UDP` ارسال ، 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 همتاها 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 صورت که بهترین مسیر برای ارسال بسته از طریق اتصال خود همتا به همتای شبکه‌ی مقابل باشد، باید بسته را به درگاه اینترنت ارسال نموده و در صورت که بهترین مسیر از طریق یکی از همتاهای دیگر باشد، باید بسته را به آن همتا ارسال کنید و آن همتا بسته را برای همتای شبکه‌ی مقابل می‌فرستد. توجه داشته باشید که در هر دو حالت، آدرس IP مبدا بسته را آدرس IP خود همتا قرار دهید، یعنی بسته مستقیم به دست درخواست‌کننده می‌رسد و در برگشت از بهترین مسیر نمی‌آید. همچنین هنگام ارسال این بسته‌ها به همتاهای دیگر در زیرشبکه، آدرس IP مقصد در بسته‌ها همان آدرس IP همتا در شبکه‌ی مقابل است، شما از راه آدرس `MAC` و لایه‌ی ۲ بسته را به دست همتای دیگر در زیرشبکه‌ی خود می‌رسانید. همچنین برای ارسال بسته به یک همتای خاص لازم است آدرس MAC آن همتا را بدانید. آدرس MAC همتاها را می‌توانید در هنگام دریافت پیغام‌های broadcast از آن‌ها دریافت کنید. هنگامی که از یک همتا پیغام مربوط به اطلاعات اتصال آن به همتاهای شبکه‌ی مقابل را دریافت کردید، آدرس MAC مبدا آن بسته را به عنوان آدرس MAC آن همتا ذخیره کنید.

۱.۶.۲. 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 برنامه‌ي خواسته شده ارسال کنید. علاوه برا این عبارت زیر را در خروجی چاپ کنید:

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

پاسخ به صورت زیر باشد:

```
<Command> packet respond <Port> recieved from <Target Peer IP> in <RTT>ms
```

در ادامه یک نمونه از پاسخ آورده شده است، توجه که در این نمونه فرمان `DSA packet 8001 destined for 185.88.153.4 send to 185.88.153.1` بدون پاسخ

مانده است.

هم‌چنین گام بعدی یا `NextHop IP` اگر هیچ واسطی در میان نباشد، گره‌ی `Target Peer IP` است.

```
lsa 185.88.153.4
LSA packet 8000 destined for 185.88.153.4 send to 185.88.153.4
LSA packet respond 8000 recieved from 185.88.153.4 in 1000ms
dsa 185.88.153.4
DSA packet 8001 destined for 185.88.153.4 send to 185.88.153.1
dsa 185.88.153.4
DSA packet 8002 destined for 185.88.153.4 send to 185.88.153.1
DSA packet respond 8002 recieved from 185.88.153.4 in 800ms
```

شکل ۸: ورودی‌ـخروجی نمونه برای ارسال درخواست‌ها

۷.۲. نکات تکمیلی

۱.۷.۲. توضیح Custom Information

شما می‌توانید از Information Custom چارچوب پرتو آدرس آی‌پی پایه و آدرس مک پایه و تعداد گره‌های شبکه را به دست آورید. در خط اول Information Custom آدرس IP پایه و در خط دوم آدرس MAC پایه وجود دارد. همچنین در خط سوم تعداد همتاهای موجود وجود دارد. برای مثال Information Custom ممکن است به شکل زیر باشد:

185.88.153.0
01:E1:C9:E9:21:00
6

آدرس‌های آی‌پی و مک پایه به ازای تعداد گره‌های شبکه افزایش یک به یک دارند یعنی آدرس‌های آی‌پی برای این شبکه از `185.88.153.1` تا `185.88.153.6` است و آدرس‌های مک هم به همین ترتیب از `01:E1:C9:E9:21:01` تا `01:E1:C9:E9:21:06` خواهد بود.

۲.۷.۲. استفاده از چارچوب

در این تمرین برای اتصال دو زیرشبکه، حاوی گره‌های فرد (شبکه آبی)و زوج (شبکه‌ی قرمز) از یک ارتباط میانی استفاده شده است. برای استفاده از این ارتباط میانی کافی است، پس از `new.sh` اسکریپت

`connect_networks.sh` را از همان مسیر پوشه‌ی حاوی کدهای چارچوب پرتو اجرا کنید.

۳.۷.۲. نکته‌ای در مورد به‌روزرسانی جداول

حالتی را در نظر بگیرید که یک تبلیغ یا `advertise` از گره‌ای مانند x از راه می‌رسد، اطلاعاتی که فعلاً در جدول مسیریابی وجود دارد شامل است گره‌ی x هم می‌شود. حال اگر تبلیغ جدید گره‌ی x از حالت موجود بی‌کیفیت‌تر باشد (چه از لحاظ تاخیر و چه از لحاظ نرخ گم شدن بسته‌ها) آیا جدول باید به‌روزرسانی شود؟
پاسخ بله است، ولی به‌روزرسانی شامل این می‌شود که در این جدول هر جا گره‌ی x به عنوان گره‌ی بهینه استفاده شده به حالت `disconnect` دربیاید، و تا به‌روزرسانی بعدی همین مقدار باشد. این یعنی فقط سطری که شامل x است اگر در مورد نرخ گم شدن بسته‌ها است، آدرس گره‌ی بعدی خود گره قرار بگیرد و نرخ گم شدن بسته‌ها `1` باشد و اگر در مورد تاخیر است، تاخیر مقدار `INF` در نظر گرفته شود و باز هم گره‌ی بعدی آدرس آی‌پی خود گره در نظر گرفته شود.

نکات ضروری

- به علت اینکه نمره‌ی تمرین به صورت خودکار داده می‌شود، ساختار پیام‌های گفته شده باید دقیقاً به صورت گفته شده باشد.
- نحوه‌ی ارزیابی ممکن است دچار تغییراتی شود و تست‌های دیگری اضافه شوند.
- 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 پوشه‌ی user-router را به همراه Makefile فشرده کرده (می‌توانید این کار را با اجرای دستور `make archive` انجام دهید) و از طریق وب‌سایت پرتو ارسال نمایید .
- بخش تمرین تئوری را در قالب فایل PDF به آدرس ایمیل sasan@ce.sharif.edu ارسال کنید و همچنین ایمیل salimi@ce.sharif.edu را CC نمایید. نام فایل ارسالی و عنوان ایمیل باید بصورت YourLastName-443-PA2 باشد.
- از پیوست کردن پرونده‌ی PDF به پوشه‌ای که پرتو ارسال می‌کنید خودداری نمایید. قالب ارسال پرونده‌ها به سایت پرتو قالب Zip است، هر نوع پرونده‌ی دیگری با موفقیت ارسال نمی‌شود.