به نام خدا

تمرین برنامه نویسی سوم

درس شبکه‌های کامپیوتری

استاد راهنما: دکتر ناصر یزدانی

رشته: مهندسی کامپیوتر

دستیاران آموزشی:

پارمیدا ضرغامی **شایان شبیهی**

نیمسال اول سال تحصیلی 02 -1401

۱. مقدمه

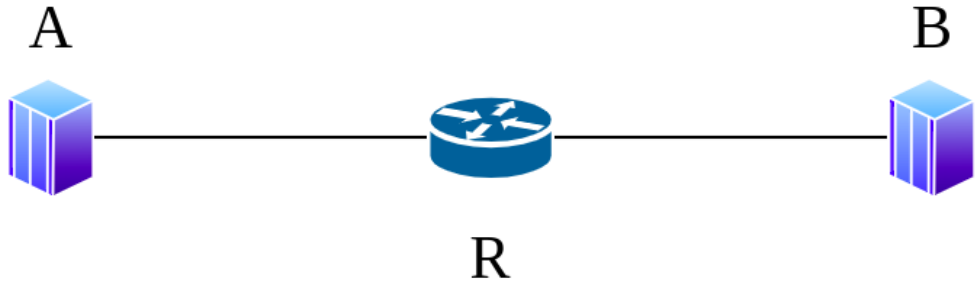
هدف از این تمرین آشنایی با routing در IP و همچنین کارکرد TCP[1] می‌باشد. این پروژه در در دو فاز زیر تعریف میگردد:

1. بخش اول به منظور آشنایی با TCP/UDP بوده و اتصال host هایی در یک network ساده شامل روتر ها میباشد که با استفاده از پروتکل TCP باهم ارتباط دارند.
2. بخش دوم، درای دو فاز است. ابتدا جداول routing در یک شبکه بزرگتر شامل چندین روتر و host تشکیل میگردد، و سپس از آنها جهت ارسال پکت در توپولوژی شبکه با استفاده از پروتکل TCP استفاده میشود.

۲. بخش اول

۲.۱. توضیحات کلی

فرض کنید دو نود A و B با استفاده از یک روتر به هم متصل شده اند. در این بخش، هر کدام از هاست ها و روتر را به صورت یک پردازه[2] یا رشته[3] مستقل در نظر بگیرید. همچنین، در این قسمت از معماری شبکه زیر برای مدلسازی استفاده میکنیم:



در اینجا فرض میکنیم ارسال داده ها از A به B بوده، و برای آن از فایل مشخص ارائه شده با حجم 1MB استفاده میشود. همچنین، میبایست از پروتکل TCP و slow start برای ارسال از طریق پورت استفاده کنید.

[1] Transmission Control Protocol
[2] Process
[3] Thread

پروتکل slow start برای کنترل congestion در شبکه و در استفاده از TCP کاربرد دارد. برای پیاده سازی این پروتکل لازم است شبه کد زیر را پیاده سازی کنید.

```
# Initially:
cwnd = 1;
ssthresh = infinite;

# New ack received:
if (cwnd < ssthresh):
   # Slow Start
   cwnd = cwnd * 2;
else:
   #Congestion Avoidance
     cwnd = cwnd + 1;

# Timeout:
ssthresh = cwnd /2; # Multiplicative decrease
cwnd = 1; # Slow start again
```

در اینجا به دلیل احتمال drop شدن پکت ها در میانه مسیر به دلایل مختلف، لازم است از پروتکلی (Go-Back-N و یا Selective Repeat) برای ارسال مجدد آنها استفاده کنید. پروتکل انتخابی در اینجا به دلخواه میباشد، اما میبایست در گزارش خود به تفصیل نحوه کار و پیاده سازی آن را شرح دهید. همینطور، دلیل انتخاب خود را در گزارش پروژه ذکر کنید.

نکته: ساختار پکت ها را به صورت زیر در نظر بگیرید:

is_ack (1 bit)	#packet_id (16 bits)	src_addr (32 bits)	dest_addr (32 bits)	Data (arbitrary)

نکته: در پیاده سازی پروتکل های ارسال مجدد مذکور، باید از یک دوره timeout استفاده کنید تا بتوانید دریافت و یا عدم دریافت ACK را تشخیص دهید. در این حالت، الگوریتم timeout مورد نظر خود را توسعه دهید و منطق کاری آن را در گزارش ذکر کنید.

۲.۲. شبیه سازی

در ابتدا، اندازه بافر روتر (بصورت FIFO) را بینهایت و سایز پکت ها را 10 بایت در نظر بگیرید و داده های فایل **یک مگابایتی** داده شده را ارسال کنید.

در مرحله بعدی، اندازه بافر روتر را 5 پکت و سایز پکت ها را برابر 10 بایت درنظر بگیرید و داده های فایل 1 مگابایتی داده شده را از A به B ارسال کنید. همچنین فرض کنید روتر با احتمال 10% پکت های دریافتی (شامل پکت های دیتا و یا ACK) را drop میکند و برای ارسال هر بسته هم یک میلی ثانیه تاخیر ایجاد میکند. همچنین در صورت پر بودن بافر روتر تمامی بسته های دریافتی drop میشوند.

نکته: در هر دو مورد لازم است پس از دریافت محتویات فایل مورد نظر در مقصد آنها را در فایل مشابهی ذخیره کرده و با استفاده از ابزار $diff$ یا ابزار های مشابه درستی پروتکل خود را بررسی کنید. تصویر خروجی این ابزار به همراه تعداد bit های هر دو فایل اولیه و ثانویه بصورت دقیق در گزارش شما ذکر گردد.

نکته: زمان تمام شدن ارسال کامل داده ها را در گزارش خود بیاورید و مقادیر Max window size و ssthresh را نیز نمایش دهید.

۲.۳. سوالات

لازم است به موارد زیر پاسخ داده، و پاسخ خود را به طور کامل در گزارش خود شرح دهید:

1. تفاوت های اصلی پروتکل های TCP و UDP چیست؟ از هر یک در چه مواردی استفاده میشود؟
2. مزایا و معایب Selective Repeat و Go-Back-N نسبت یکدیگر چیست؟ دلیل انتخاب خود را ذکر کنید.

۳. بخش دوم

در این بخش به بررسی دو فاز مسیریابی و کارکرد TCP در یک توپولوژی بزرگتر میپردازیم و عوامل مختلف تاثیرگذار در بهینه سازی شبکه مورد نظر را بررسی میکنیم.

۳.۱. فاز اول

هدف از این فاز مسیریابی بر روی یک توپولوژی شبکه با استفاده از الگوریتم مسیریابی DVRP[4] میباشد. در این توپولوژی برای شبیه سازی هاست ها و روتر های شبکه مورد نظر از نود ها و برای شبیه سازی لینک های ارتباطی از یال ها در گراف ورودی استفاده میکنیم. شما میبایست به کمک الگوریتم DVRP مسیریابی را برای تمامی روتر های شبکه به شکلی انجام دهید که هر یک کوتاه ترین مسیر به هر هاست با آدرس IP مشخص را داشته باشند. دقت کنید که در اینجا برای سادگی مسئله از بررسی subnetting برای LAN های مختلف صرف نظر میکنیم ولی در واقعیت معمولا از آن برای افزایش فضای آدرس دهی با تعداد آدرس های IP در حالت base استفاده میشود.

۳.۱.۱. دستورات زمان اجرا

۳.۱.۱.۱. هاست ها و روتر ها

دستور زیر در هر مرحله از اجرای شبیه سازی لیستی از هاست ها و یا روتر های مورد نظر را به توپولوژی اضافه میکند:

```
add <hosts | routers> <ip_1> <ip_2> <ip_3> … <ip_n>
```

۳.۱.۱.۲. لینک ها

دستور زیر در هر مرحله از اجرای شبیه سازی یک لینک را بین دو نود از توپولوژی اضافه یا آپدیت میکند:

```
<add | update> link <ip_1> <ip_2> <cost>
```

[4] Distance-Vector Routing Protocol

همچنین، دستور زیر برای حذف لینک بین دو نود استفاده میشود:

```
remove link <ip_1> <ip_2>
```

همینطور، دستور زیر میتواند در صورت نیاز برای تعیین زمانی برای down شدن خودکار یک لینک و up شدن مجدد پس از مدت زمانی مشخص استفاده شود. در اینجا sd_ms مشخص کننده زمان شروع down شدن به میلی ثانیه (پس از شروع شبیه سازی در صورت در جریان نبودن) و td_ms بیانگر میزان زمان down بودن به میلی ثانیه میباشد. پس از گذشت زمان td_ms از down شدن لینک میبایست بصورت خودکار به وضعیت up برگردد.

```
update link <ip_1> <ip_2> <sd_ms> <td_ms>
```

همچنین، با دستور زیر میتوان از وضعیت هر لینک در هر لحظه آگاهی پیدا کرد. در صورت down بودن موقتی لینک میزان زمان باقی مانده و کل down بودن آن نیز میباست نمایش داده شود:

```
log status <ip_1> <ip_2>
```

۳.۱.۱.۳. شروع اجرای شبیه سازی

پس از اضافه کردن نود ها و لینک های اولیه باید برای شروع شبیه سازی الگوریتم از دستور زیر استفاده شود:

```
run
```

۳.۱.۱.۴. نمایش گراف توپولوژی

دستور زیر در هر زمان (قبل یا بعد از شروع شبیه سازی) گراف شبکه مورد نظر را رسم و نمایش میدهد:

```
draw
```

نکته: برای نمایش گراف شبکه میتوانید از کد موجود در این لینک استفاده کنید.

نکته: در گراف نمایش داده شده میبایست هاست ها و روتر ها به صورت مجزا مشخص بوده، و همچنین آدرس IP هر نود در کنار آن چاپ گردد.

۳.۱.۱.۵. نمایش جدول مسیریابی روتر ها

هر روتر و هاست برای مسیریابی میبایست یک routing table داشته باشند که در آن اطلاعات routing به تمامی آدرس های IP به همراه شماره interface خروجی قرار دارد.

دستور زیر برای چاپ جدول routing مربوط به یک نود از شبکه با آدرس IP مشخص استفاده میشود:

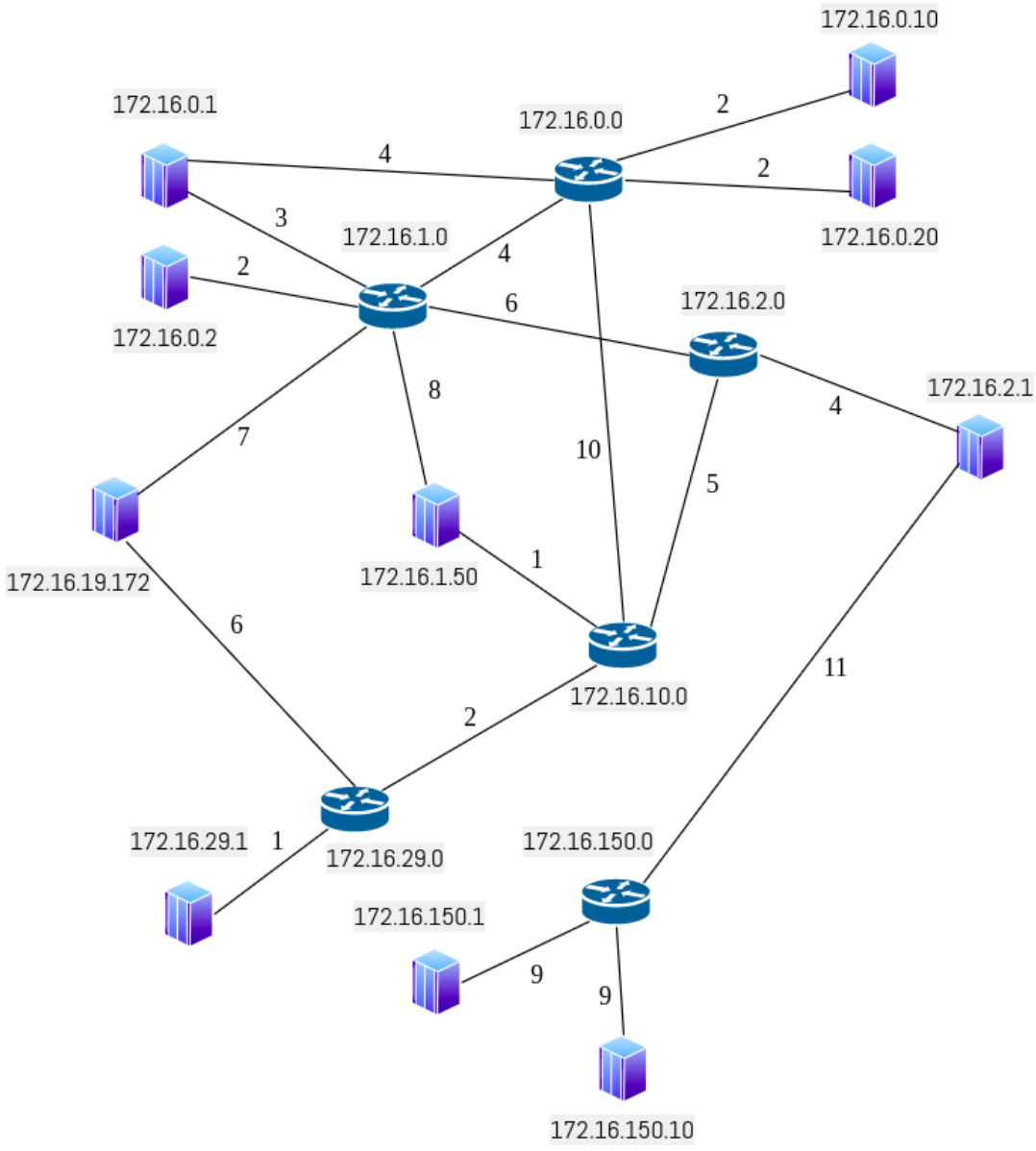
```
show table <ip>
```

۳.۱.۲. تست و شبیه سازی

برای تست اولیه کارکرد صحیح کد خود میبایست توپولوژی زیر را (که در آن آدرس IP نود ها و هزینه لینک های ارتباطی مشخص است) پیاده سازی کرده و نحوه کارکرد دستورات مختلف به همراه تاثیر آنها بر جداول مسیریابی و زمان convergence الگوریتم DVRP را گزارش نمایید. دقت کنید که این قسمت بخش بزرگی از نمره شما در این فاز را تشکیل میدهد و میبایست در آن عوامل موثر بر کاهش یا افزایش هزینه مسیر های مختلف و همچنین زمان convergence را به طور کامل با تصاویر شبیه سازی پوشش دهید. تمامی تصاویر باید شامل خروجی شبیه سازی برنامه و عنصر زمان باشند.

نکته: دقت کنید که الگوریتم مسیریابی شما میبایست مجددا پس از هر تغییر دستی یا خودکار در توپولوژی آپدیت شود و برای این موضوع هم logging مناسب و کامل داشته باشید.

نکته: در logging خود برای هر دو فاز حتما زمان چاپ شدن هر پیام را نیز چاپ کنید.



نکته: دقت کنید که در زمان تحویل کد شما با تست های مجزا بررسی خواهد شد.

۳.۱.۳. سوالات

1. آیا راه بهتری برای آپدیت بهینه تر جداول پس از تغییرات در توپولوژی ورودی (به جای اجرای مجدد الگوریتم برای تمامی نود ها) میشناسید؟ کمی توضیح دهید.
2. LSRP نوع دیگری از الگوریتم های مسیریابی تحت IP است که به صورت گسترده استفاده میشود. تفاوت های اصلی و مزایا و معایب نسبی DVRP و LSRP را گزارش کنید.

۳.۲. فاز دوم

در این قسمت میبایست با استفاده از مسیریابی فاز قبل جداول مسیریابی توپولوژی را بدست آورده و ذخیره کنید، و در مرحله بعد از آنها برای ارسال داده ها با استفاده از TCP استفاده کنید.

۳.۲.۱. توضیحات کلی

فرض کنید توپولوژی فاز قبل به شما داده شده است و هر هاست داده های خود را میبایست از طریق پروتکل slow start ارسال کند. داده های ارسالی با پروتکل TCP به صورت زیر میباشند:

1. داده ی یک مگا بایتی را از هاست 172.16.0.1 به هاست 172.16.29.1 ارسال کنید. این ارسال باید با وقفه های ۴۵ ثانیه ای مجددا تکرار شود و محتویات دریافتی هر بار در یک فایل جدید ذخیره شود.
2. داده ی 20 مگا بایتی از هاست 172.16.1.50 به هاست 172.16.2.1 به صورت مداوم (بدون وقفه زمانی) ارسال میشود.
3. داده ی 1 مگا بایتی از هاست 172.16.19.172 به 172.16.0.10 به صورت مداوم (بدون وقفه زمانی) ارسال میشود.

نکته: برای پکت های ارسال و دریافت شده (شامل ACK)، تکمیل دریافت یا ارسال، و ... در هر یک از موارد بالا logging را در یک فایل مجزا (و نه در terminal) انجام دهید.

نکته: تنها برای مورد اول لازم است محتویات دریافتی هر بار در فایل جدیدی ریخته شوند و برای سایر موارد میتوانید خروجی را صرفا در یک فایل ذخیره کنید.

نکته: دقت داشته باشید که هاست ها توانایی گرفتن پیام و ارسال آن به پورت دیگر را ندارند.

نکته: در این فاز میتوانیم از دستورات فاز قبل برای تغییر توپولوژی شبکه استفاده کنیم و در نتیجه امکان تغییر جداول وجود دارد پس امکان تغییر مسیر پکت ها نیز وجود دارد.

نکته: در صورت نیاز به پردازش multi-threaded از کتابخانه های pthread و یا thread استفاده کنید.

۳.۲.۲. شبیه سازی

در ابتدا سایز پکت ها را 10 بایت در نظر گرفته و اندازه ی بافر روتر (بصورت FIFO) را 5 بسته در نظر بگیرید. روتر ها بعد از یک میلی ثانیه پکت هارا ارسال میکنند.همچنین در صورت پر بودن بافر روتر تمامی بسته های دریافتی drop میشوند.

در مرحله بعد روتر ها مانند مرحله قبل داده ها را ارسال میکنند با این تفاوت که میبایست پروتکل ECN را نیز بر روی روتر ها و هاست ها پیاده سازی کنند.

نکته: هاست ها بعد از گرفتن سیگنال کنترل ازدحام window size خود را نصف میکنند. واضح است برای اضافه کردن این پروتکل ساختار پکت تغییر میکند.

مرحله اول و مرحله ی دوم را از لحاظ زمانی مقایسه کرده و تفاوت های این دو قسمت را بیان کنید. همچنین، بگویید کدام پروتکل برای کد شما بهتر عمل کرده است و چرا؟

۳.۲.۲. سوالات

1. تفاوت پروتکل های Random Early Detection و Explicit Congestion Notification را توضیح دهید. استفاده از هر یک را در چه مواردی پیشنهاد میکنید؟

۴. جمع بندی و نکات پایانی

- برای تحویل این پروژه تا **۷ بهمن ۱۴۰۱** فرصت خواهید داشت.
- پروژه در گروه‌های **دو نفره** انجام می‌شود.
- برای پیاده سازی این پروژه میتوانید از زبان های **C** یا **C++** استفاده کنید. همچنین، میبایست از فایل های مشخص شده در صفحه درس برای ارسال و دریافت در کد خود استفاده کنید.
- ساختار صحیح و تمیزی کد برنامه، بخشی از نمره‌ی این پروژه شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.
- همه اعضای تیم میبایست کار انجام شده را تقسیم کنند. همچنین لازم است از **Git** برای ساختن branch ها و تقسیم issue ها استفاده نمایید.دقت کنید که نمره هر شخص در این پروژه با بررسی **میزان مشارکت در commit ها و issue ها** تعیین میشود.
- پس از تکمیل کد پروژه لازم است در Gitlab یک private repository با نام CN-Final-CA ساخته، و هر دو TA تمرین (شایان شبیهی[5] و پارمیدا ضرغامی[6]) را با دسترسی maintainer به آن اضافه کنید. این repository میبایست شامل گزارش شما و تمامی فایل های مورد نیاز برای اجرای کد باشد.
- لازم است تمامی issue های ایجاد شده و resolve شده در Gitlab قابل مشاهده باشد. همچنین باید این لیست کامل باشد و تمامی issue های ایجاد شده را شامل شود.
- برای آپلود پاسخ خود کافیست لینک repo خود را به همراه مقادیر hash و زمان آخرین commit انجام شده در صفحه درس قرار کنید. دقت کنید که در صورت وجود تاخیر، این مورد با توجه به submission شما در صفحه درس تعیین خواهد شد. همچنین، به کد و یا گزارش های آپلود شده در صفحه درس نمره ای تعلق نمیگیرد.

[5] @shabihish
[6] @p.zarghami

- لازم است کد نهایی و گزارش شما در آخرین commit از شاخه main قابل دسترسی باشد و برای این منظور میتوانید با استفاده از دستور merge شاخه ای دیگر را روی آن merge کنید. همچنین، در زمان تحویل از شما خواسته خواهد شد تا با اجرای دستورات زیر در terminal صحت موارد مذکور در submission خود را نشان دهید:

```
$ git reset --hard
$ git log --oneline --graph --all
```

- میتوانید گزارش خود را در قالب یک فایل README.md در Gitlab قرار دهید و یا گزارشی با فرمت LaTeX ارائه دهید. در هر صورت، این گزارش میبایست به طور کامل تمامی مراحل کار، بخش های مختلف کد، و نتایج نهایی را به همراه screenshot هایی مستند (و با ثبت زمانی) پوشش دهد. برای هر قسمت کد، گزارش دقیق و شفاف بنویسید. کدهای ضمیمه شده بدون گزارش مربوطه نمره‌ای نخواهند داشت.
- سیستم عامل مورد استفاده در زمان کد زنی و تحویل میبایست Linux باشد. همچنین، آشنایی مورد نیاز با دستورات Git در terminal مورد انتظار میباشد.
- هدف این تمرین یادگیری شماست. لطفا تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، **مطابقت سیاست درس با گروه متقلب و تقلب دهنده برخورد خواهد شد**. دقت کنید که در این مورد هیچگونه عذری پذیرفته نخواهد بود.
- سؤالات خود را تا حد ممکن در گروه تلگرامی درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن بهره‌مند شوند. در صورتی که قصد مطرح کردن سؤال خاص‌تری دارید، میتوانید از طریق ایمیل با طراحان تمرین در ارتباط باشید. توجه داشته باشید که سایر شبکه‌های اجتماعی راه ارتباطی رسمی با دستیاران آموزشی نیست و دستیاران آموزشی مؤظف به پاسخگویی در محیط‌های غیررسمی نیستند.

موفق باشید.