

به نام هستی بخش

پروژه اول درس شبکه های کامپیوتری

آشنایی با Mininet

زمستان ۱۳۹۸

در این پروژه قصد داریم شما را با ابزارهای پایه برای شبیه سازی شبکه های کامپیوتری و مفهوم Network namespace آشنا کنیم. یکی از این ابزارها mininet است که در این پروژه و پروژه های بعد با آن کار خواهید کرد. در کنار آشنایی با Mininet مفهومی که قصد داریم با آن آشنا شوید Network Namespace است که به وسیله آن میتوان سخت افزارهای مجازی شبکه را در سیستم عامل لینوکس درست کرد. این ابزار تولید نرم افزارهایی مانند Docker (مراجعه به منبع [4] و [5]) را ممکن کرده است. در بخش آخر نیز آشنایی مختصری با مفهوم Ping و بسته های ICMP خواهیم داشت.

توصیه می شود پیش از شروع پروژه، نکات تکمیلی که در انتهای توضیحات آمده اند را به دقت مطالعه فرمایید.

۱. مفهوم Network Namespace در لینوکس

قبل از آشنایی با mininet، سراغ Network Namespace می رویم تا بستری که نوشتن ابزارهایی مانند mininet را ممکن کرده است، معرفی کنیم.

طبیعتاً تاکنون لپتاپ خود را به اینترنت متصل کرده اید؛ سیستم عامل سخت افزار ها و نرم افزار های لازم برای دسترسی به شبکه (مانند کارت و واسطه های شبکه و جدول مسیریابی) را مدیریت می کند و بین پرده ها به اشتراک می گذارد. منظور از واسط شبکه نقطه ای است که کامپیوتر شما از طریق آن به شبکه متصل می شود.

در سیستم عامل لینوکس با اجرای دستور ifconfig واسطه های موجود نشان داده می شوند. مشابه آن در ویندوز دستور ipconfig است. هر کدام از این واسطه ها می توانند دارای یک آدرس باشند تا به آن ها این امکان را بدهد که بسته های داده را ارسال و دریافت کنند. **این دستورات را در ترمینال کامپیوتر خود اجرا کرده، آدرس را مشخص و تصویر آن را در گزارش خود بیاورید.**

سیستم عامل برای اینکه تصمیم بگیرد هر بسته خروجی را از طریق کدام واسط ارسال کند، از جدول مسیریابی (routing table) استفاده می کند. این جدول بین پرده های یک کامپیوتر یکسان است. حال فرض کنید می خواهید پرده هایی ایجاد کنید که مجموعه متفاوتی از واسط ها و جدول های مسیریابی داشته باشند. در این مواقع از Network Namespace استفاده می کنیم. در ادامه شما را با برخی دستورات Network Namespace آشنا می کنیم.

با استفاده از دستورات زیر می توان دو Network Namespace با نام های h1 و h2 ساخت:

```
ip netns add h1
```

```
ip netns add h2
```

سپس با کمک دستور زیر دو کابل شبکه مجازی (مشابه کابل LAN در دنیای واقعی) می‌سازیم و هر سر این لینک‌ها را با نامی مشخص می‌کنیم. برای مثال برای دو سر کابل اول نام‌های s1-eth1 و h1-eth0 انتخاب می‌شوند:

```
ip link add s1-eth1 type veth peer name h1-eth0
```

با اجرای دستورهای زیر یکی از سرهای لینکی که در دستور قبل ساختیم به namespace متناظرشان متصل می‌شود (مشابه اینکه یک سر را به کامپیوتر خود متصل کنیم):

```
ip link set h1-eth0 netns h1
```

```
ip link set h2-eth0 netns h2
```

اکنون switch را می‌سازیم:

```
Ovs-vsctl add-br s1
```

سپس سر دیگر لینک‌ها را به switch اختصاص داده و فعال می‌کنیم:

```
ovs-vsctl add-port s1 s1-eth1
```

```
ovs-vsctl add-port s1 s1-eth2
```

```
ip link set s1-eth1 up
```

```
ip link set s1-eth2 up
```

*دقت کنید که دستورات این بخش کامل نیستند و باید با استفاده از مفاهیمی که آموختید و دستورات گفته شده، آن‌ها را کامل کنید.

اکنون برای درک بیشتر این موضوع به سوالات زیر پاسخ دهید.

- ۱- چگونه می‌توان یک دستور را داخل یک namespace به‌خصوص مثلا h1 اجرا کرد؟
- ۲- چرا هنگام اجرای دستور زیر با پیغام Network is unreachable روبه‌رو می‌شویم؟ برای رفع این مشکل چه راهکاری پیشنهاد می‌دهید؟

```
ip netns exec h1 ping 10.0.0.2
```

- ۳- چگونه می‌توان به یک interface متعلق به host آدرس IP اختصاص داد؟
- ۴- Virtual Ethernet چیست؟
- ۵- با چه دستوری می‌توان یک Virtual Ethernet ساخت؟
- ۶- با چه دستوری می‌توانیم Virtual Ethernet ساخته شده را به یک Network Namespace دیگر منتقل کنیم؟
- ۷- اجرا شدن دستور زیر چه اطلاعاتی به ما می‌دهد؟

```
ip netns exec <my_namespace_name> ip addr list
```

برای پاسخ به این سوالات پیشنهاد می‌شود به منبع [1] و منبع [2] مراجعه کنید. لطفاً هر جا در گزارش نیاز به آوردن دستور دارید قالب آن را بنویسید (مثلاً `ip netns exec <namespace_name> <command>`) و از کپی کردن دستورات و متن منابع بپرهیزید.

۲. نوشتن یک شبیه‌ساز

اکنون با استفاده از دستوراتی که در مرحله قبل آموختید:

- ۳ هاست بسازید با نام‌های `h1`، `h2` و `h3`.
- دو سویچ بسازید با نام‌های `s1` و `s2`.
- آدرس IP هاست `x` را `10.0.0.x` قرار دهید.
- لینک بین سویچ‌ها و هاست‌ها را به این صورت برقرار کنید:

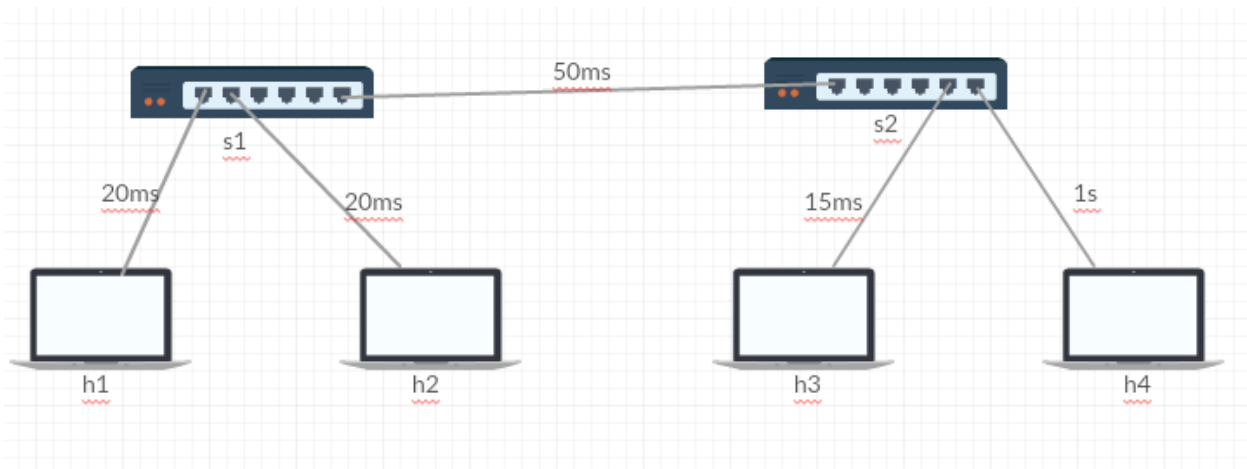
`H1—s1, h2—s1, s1—s2, h3—s2`

- یک فایل `bash` بنویسید و دستورات خود را در آن قرار دهید. به نحوی که با اجرای آن توپولوژی فوق ساخته شده و هاست اول هاست سوم را `ping` کند.
- برای این بخش مجازید حداکثر ۵۰ خط کد بنویسید.
- ✓ می‌توانید به کمک دستورات بخش قبل و منابع [6] و [7] این بخش را تکمیل کنید.

۳. استفاده از Mininet

احتمالا تا اینجا متوجه شده‌اید که اگر بخواهید یک شبکه بزرگتر را در کامپیوتر خود ایجاد کنید، پیچیدگی‌های کار با `namespace` بسیار زیاد خواهد شد. ابزاری به نام `Mininet` از طریق ارائه API‌های ساده‌تر (به زبان پایتون) ساختن شبکه مجازی را راحت‌تر کرده است.

برای آشنایی با `Mininet` و دستورات آن منبع [3] را مطالعه کنید. پس از مطالعه به پیاده‌سازی موارد زیر بپردازید (یک منبع بسیار مفید برای یادگیری این بخش مطالعه `example`‌های موجود در `Mininet` می‌باشد).



- یک توپولوژی با ۴ هاست و ۲ سویچ مطابق شکل بالا ایجاد کنید.
- با استفاده از دستور `nodes`، هاست‌ها و با دستور `net`، لینک‌های شبکه را مشاهده کنید.
- تاخیر هر لینک بر روی آن نشان داده شده است. شما باید این مقدار را هم از طریق API پایتون تنظیم کنید.

پاسخ سوالات زیر را در گزارش خود بیاورید.

- ۱- خروجی دستور pingall در ترمینال Mininet چیست؟
- ۲- با استفاده از دستور xterm در ترمینال Mininet برای هاست‌های h1 و h3 یک ترمینال باز کرده و IP هر هاست را با دستور ifconfig نمایش دهید. تصویر این بخش را در گزارش خود قرار دهید.

۴. بررسی عوامل تاثیرگذار بر ترافیک شبکه

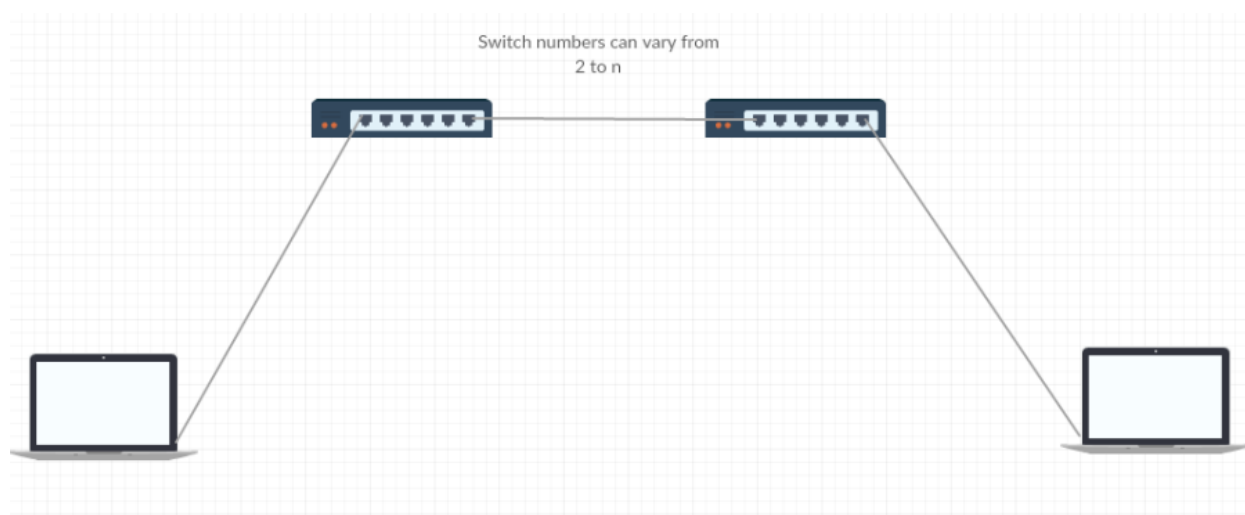
توپولوژی شکل زیر را با استفاده از Mininet پیکربندی بفرمایید. سپس پارامترهای Delay و Bandwidth و همچنین تعداد Switchها و نیز Switch Queue (برای آشنایی با پارامترهای switch به بخش Setting Performance Parameter در منبع [8] مراجعه کنید) تغییر داده و نقش آن‌ها را در زمان ارسال و بازگشت بسته ping از میزبان اول به دوم بررسی کنید. سپس با استفاده از نرم‌افزار وایرشارک تغییرات تعداد بسته‌هایی که میزبان اول به میزبان دوم ارسال می‌کند (بسته‌های ICMP ECHO_REQUEST) پیش از اینکه میزبان اول، اولین بسته پاسخ (ICMP ECHO_REPLY) را از میزبان دوم دریافت کند را با تغییر پارامترهای ذکر شده، مورد تحلیل قرار دهید. در انتها همه این موارد را در گزارش کار خود ذکر کنید.

برای انجام این مقایسه‌ها حالات زیر را در نظر بگیرید:

برای مقایسه‌ی

- Delayها با مقادیر ۲۰ و ۹۰
- bandwidth با مقادیر ۱ و ۱۵
- Switch Queue با مقادیر ۱ و ۱۵
- تعداد سوییچ‌ها با مقادیر ۲ و ۷

کار کنید.



۵. آشنایی با مفهوم Ping و پروتکل ICMP (Internet Control Message Protocol)

فضای ذخیره‌سازی بزی پر شده است و او می‌خواهد به کمک دانش شبکه شما اطلاعات خود را روی شبکه خصوصی خود ذخیره کند. برای آشنایی با چگونگی انجام این کار با ما همراه باشید.

عملکرد ping بدین صورت است که یک بسته را به مقصد موردنظر ارسال کرده (ECHO-REQUEST) و مقصد همان بسته را به عنوان جواب به مبدا برمی‌گرداند (ECHO_REPLY). از این ابزار به منظور کسب اطمینان از صحت اتصال فیزیکی میان میزبان مبدا و مقصد استفاده می‌شود.

طبق شکل زیر که نمایان‌گر یک بسته ICMP است، در قسمت source ip address آدرس IP مبدا و در قسمت destination ip address آدرس IP هاست مقصد قرار می‌گیرد. هنگامی که هاست مقصد بسته REQ را دریافت می‌کند، آدرس IP های مبدا و مقصد را جابجا کرده و باقی بسته را عینا ارسال می‌کند (با فیلدهایی که توسط روتر ها و یا عوامل دیگر در مسیر تغییر می‌کنند کاری نداشته باشید).

IP Datagram

	Bits 0–7	Bits 8–15	Bits 16–23	Bits 24–31
IP Header (20 bytes)	Version/IHL	Type of service	Length	
	Identification		flags and offset	
	Time To Live (TTL)	Protocol	Checksum	
	Source IP address			
	Destination IP address			
ICMP Header (8 bytes)	Type of message	Code	Checksum	
	Header Data			
ICMP Payload (optional)	Payload Data			

تغییری که در این پروژه شامل حال ping می‌شود، به این صورت می‌باشد که هاست مبدا بسته ICMP ECHO_REQUEST را به سمت هاست مقصد ارسال می‌کند. اما در IP Header آن، به جای اینکه IP خودش را در قسمت source ip address قرار دهد، IP یکی دیگر از هاست‌ها را به صورت تصادفی قرار می‌دهد. برای مثال آدرس IP میزبان شماره ۱۱ به صورت 10.0.0.11 می‌باشد. (در مجموع IP میزبان‌ها برای میزبان شماره x به صورت 10.0.0.x است)؛ اما او در بسته‌ای که جهت ارسال آماده می‌کند آدرس IP مبدا را 10.0.0.5 قرار می‌دهد. حال این گونه می‌شود که بسته ICMP ECHO_REQUEST از مبدا به سمت مقصد مدنظر ما می‌رود. در آن جا IP های مبدا و مقصد جابجا شده و عینا بسته برگشت داده می‌شود. اما به جای اینکه به خود ما، که مبدا واقعی بودیم، بازگردد، به میزبانی که IP آن را در فیلد source ip address بسته اولیه قرار دادیم، می‌رسد. حال میزبان دریافت کننده این بسته مجددا عمل ارسال بسته ICMP ECHO_REQUEST را مانند میزبان اولیه (یعنی با Spoof کردن IP بسته ارسالی) را تکرار می‌کند. نتیجه این می‌شود که بسته ما که شامل یک پیام در بخش ICMP Payload می‌باشد، در شبکه‌ای از هاست‌ها که متعلق به خود ما هستند، شروع به چرخیدن می‌کند و در لحظه هیچ کس نمی‌داند که بسته دقیقا کجا قرار دارد.

حال برای اینکه میزبان اول بسته در حال چرخش را بازیابی کند، یک بسته ICMP ECHO_REQUEST که از آن با نام RETURN_HOME یاد می‌کنیم را در شبکه می‌چرخاند. در بخش ICMP Payload این بسته پیام return و آدرس IP خود را قرار داده و مانند بسته‌های قبلی آن را در شبکه می‌چرخاند. حال هر میزبانی در شبکه که بسته RETURN_HOME را دریافت می‌کند این آمادگی را پیدا می‌کند که با دریافت بسته اولیه آن را به سمت میزبان اول که IPش را هم دارد برگرداند.

اما هدف پِزی از انجام این کار صرفاً انتقال یک بسته کوچک ICMP نیست؛ بلکه او می‌خواهد تا فایل‌های خود را اینگونه در شبکه نگهداری کند. بدین منظور برنامه‌ای که شما برای او می‌نویسید باید این قابلیت را داشته باشد که در هر لحظه هر فایلی که پِزی نام آن را وارد کرد را به تیکه‌های به حد کافی کوچک که در یک بسته ICMP جا شوند شکسته و اقدام به چرخاندن آن‌ها در شبکه بکند. شما برای اینکه تمایز این بسته‌ها در شبکه بتوانید مشخص کنید کافی‌ست تا در فیلد id بسته ICMP شماره بسته‌های یک فایل و در ابتدای Payload هر بسته نام یا شماره نظیر فایل مربوط به آن بسته (اینکه در این بخش چه چیزی قرار دهید کاملاً دست خودتان می‌باشد و فقط باید به گونه‌ای باشد که بتوانید متوجه شوید هر بسته متعلق به چه فایلی می‌باشد) را قرار دهید. بدین ترتیب تا اینجا شما توانستید تا چند فایل را در شبکه در چرخش داشته باشید. حال برای اجرای فرمان RETURN_HOME هم کافی‌ست تا نام فایلی که میزبان اول قصد بازگشت آنرا دارد دریافت کنید و فقط بسته‌های مربوط به آن را از شبکه جمع کرده و نهایتاً یک فایل کامل و مرتب به همان صورتی که قبلاً در اختیار او بود به او تحویل دهید.

این نکته را هم مدنظر داشته باشید که نیازی نیست فایل‌های شما خیلی بزرگ باشند و کفایت حجم هر فایل چیزی در حدود ۱ کیلو بایت باشد.

برای سادگی کار می‌توانید فرض کنید تعداد هاست‌های موجود در توپولوژی را در ابتدای شروع برنامه می‌دانید. مثلاً یک عدد از ورودی بخوانید یا هر روشی که دوست دارید!

*هم‌چنین توجه داشته باشید که برای ما بسیار مهم است که تمامی بسته‌های گفته شده حتماً از جنس ICMP باشد. استفاده از لایه‌های بالاتر به‌منظور کمک به پِزی هیچ نمره‌ای برای شما نخواهد داشت.

۶. توضیحات تکمیلی

- پروژه دو نفره است و نمره افراد لزوما یکسان نمی‌باشد.
- تنها استفاده از بان برنامه‌نویسی پایتون مجاز می‌باشد. هم‌چنین جهت سادگی کار نمونه کدی در اختیار شما قرار می‌گیرد که به عنوان ورودی از شما آدرس IP خودتان و مقصد موردنظرتان را گرفته و حاصل انجام ۳ بار عمل ping را به باز می‌گرداند.
- به این نکته توجه داشته باشید که برای انجام این پروژه لزوما باید از ماشین مجازی Mininet که بر روی ابزارهایی نظیر Virtual box اجرا می‌شود استفاده کنید. نصب پکیج Mininet بر روی توزیع‌های مختلف لینوکس مورد قبول ما نیست و نمره‌ای نخواهد داشت.
- توضیحات خواسته شده در هر بخش را در گزارش خود بیاورید. در صورتی که کل توضیحات شما (بدون در نظر گرفتن تصاویر موجود) بیشتر از ۳ صفحه شود، نمره قابل قبولی را از دست خواهید داد.
- در هر بخش که خواسته شده تصویری در گزارش خود بیاورید به این نکته توجه داشته باشید که این تصویر میبایست شامل ساعت و تاریخ سیستم باشد. برای این کار میتوانید کادر مربوط به زمان سیستم عامل را در هنگام تصویربرداری باز نموده، و یا یک Terminal که دستور date درون آن اجرا شده است را در هنگام تصویربرداری در کنار مجموعه قرار دهید.
- برای ما خیلی مهم است که حاصل کار خودتان را تحویل دهید. بنابراین به شدت با تقلب برخورد می‌شود.
- توصیه می‌کنیم هر دو نفر با Mininet آشنا باشند، زیرا در پروژه‌های بعدی به آن احتیاج خواهید داشت.
- حتما گزارش خود را به زبان و رسم الخط فارسی بنویسید. نمره گزارش بدون رعایت کردن این نکته ۰ می‌باشد.
- به این نکته هم توجه کنید که بخش زیادی از ۴ قسمت ابتدایی، مشابه پروژه سال‌های گذشته می‌باشد. اما انجام آن‌ها بسیار ساده بوده و دانش حاصل از آن‌ها هم برای انجام بخش ۵ام پروژه و هم برای انجام پروژه‌های آینده به کمک شما خواهد آمد. هم‌چنین بدانید که این بخش‌ها نمره اندکی از پروژه را به خود اختصاص می‌دهند. بنابراین سعی کنید از کپ زدن آن‌ها بپرهیزید.
- سوالات خود را در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. در صورتی که سوال شما به‌گونه‌ای باشد که جنبه جدیدی به پروژه اضافه کند، فقط گروه شما ملزم به انجام آن بوده و در ارزیابی دیگر گروه‌ها دخالت داده نمی‌شود.

۱. <http://blog.scottlowe.org/2013/09/04/introducing-linux-network-namespaces/>
۲. <https://blogs.igalia.com/dpino/2016/04/10/network-namespaces/>
۳. <http://mininet.org/walkthrough/#part-1-everyday-mininet-usage>
۴. <https://hitos.ir/63/docker-چیست-آشنایی-و-نصب-داکر-قسمت-اول>
۵. <https://www.infoworld.com/article/3204171/linux/what-is-docker-linux-containers-explained.html>
۶. <http://conferences.sigcomm.org/sigcomm/2014/doc/slides/mininet-intro.pdf>
۷. <https://screeninet.wordpress.com/2014/11/30/mininet-internals-and-network-namespaces/> (برای دسترسی به این لینک نیاز به فیلتر شکن دارید)
۸. <https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>