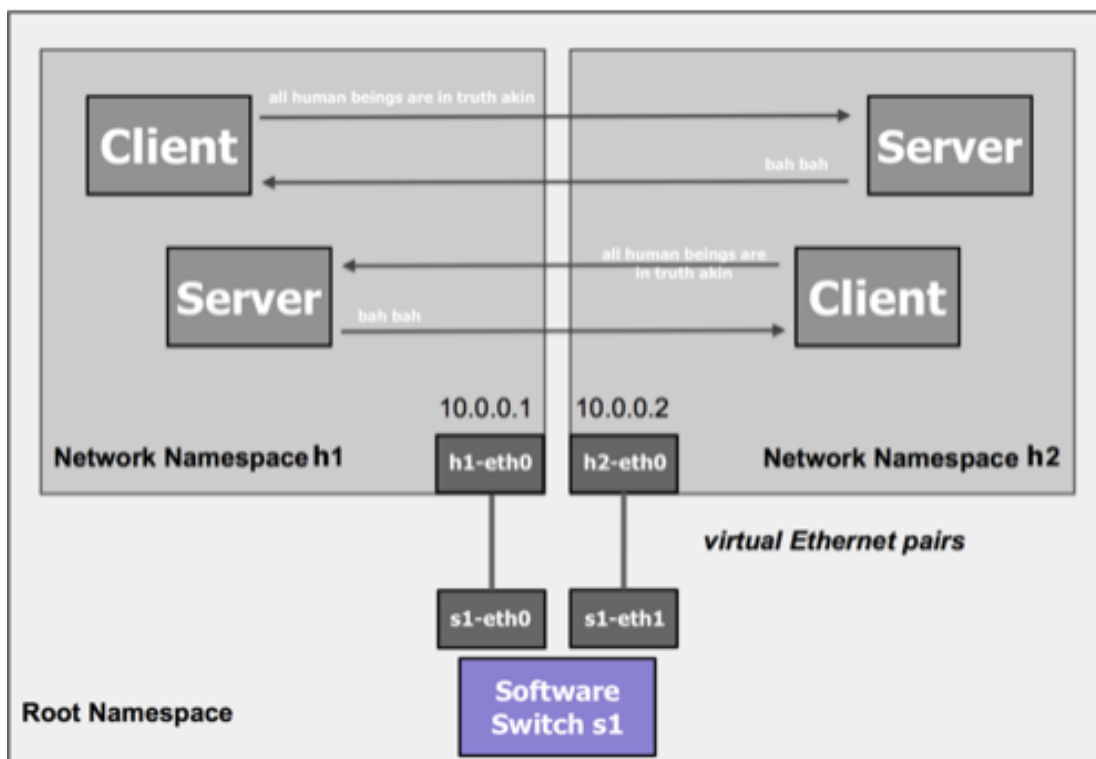


با نام و یاد خدا

پروژه‌ی اول شبکه‌های کامپیوتری

آشنایی با Mininet

بهار ۹۷



0. هدف پروژه

در این پروژه قصد داریم شما را با ابزارهای پایه برای شبیه‌سازی شبکه‌های کامپیوتری و مفهوم Network Namespace آشنا کنیم. یکی از این ابزارها Mininet است که در این پروژه و پروژه‌های بعد با آن کار خواهید کرد. در کنار آشنایی با Mininet مفهومی که قصد داریم که با آن آشنا شوید Network Namespace است که به وسیله آن می‌توان سخت‌افزارهای مجازی شبکه را در سیستم عامل لینوکس درست کرد. این ابزار تولید نرم‌افزارهایی مانند Docker¹ را ممکن کرده است. در بخش آخر آشنایی مختصری با socket خواهیم داشت.

¹ ابزاری برای اجرای قرنطینه پردازنده‌هاست. در این پروژه نیازی به آن نداریم. برای آشنایی بیشتر به منبع [4] یا منبع [5] مراجعه کنید.

۱. مفهوم Network Namespace در لینوکس

قبل از آشنایی با mininet، سراغ Network Namespace می‌رویم تا بستری که نوشتن ابزارهایی مانند mininet را ممکن کرد، معرفی کنیم.

طبیعتاً تا کنون لپ‌تاپ خود را به اینترنت متصل کرده‌اید سیستم عامل سخت‌افزارها و نرم‌افزارهای لازم برای دسترسی به شبکه (مانند کارت و واسطه‌های شبکه و جدول‌های مسیریابی) را مدیریت می‌کند و بین پردازنده‌ها به اشتراک می‌گذارد. منظور از واسط شبکه² نقطه‌ای است که کامپیوتر شما از طریق آن به شبکه متصل می‌شود.

در سیستم عامل لینوکس با اجرای دستور ifconfig واسطه‌های موجود نشان داده می‌شوند. مشابه آن در ویندوز دستور ipconfig است هر کدام از این واسطه‌ها می‌توانند دارای یک آدرس باشند تا به آن‌ها این امکان را بدهد که بسته‌های داده را ارسال و دریافت کنند این دستورات را در ترمینال کامپیوتر خود اجرا کرده، آدرس را مشخص، و تصویر آن را ضمیمه‌ی پروژه خود کنید

سیستم عامل برای اینکه تصمیم بگیرد هر بسته‌ی خروجی را از طریق کدام واسط ارسال کند از جدول مسیریابی (routing table) استفاده می‌کند. این جدول بین پردازنده‌های یک کامپیوتر یکسان است. حال فرض کنید می‌خواهید پردازنده‌هایی ایجاد کنید که مجموعه‌ی متفاوتی از واسطه‌ها و جدول‌های مسیریابی داشته باشند. در این مواقع از Network Namespace استفاده می‌کنیم در ادامه شما را با برخی از دستورات Network Namespace آشنا می‌کنیم. با استفاده از دستور زیر می‌توان دو Network Namespace با نام‌های h1, h2 ساخت.

```
ip netns add h1
```

```
ip netns add h2
```

سپس با کمک دستورات زیر این دو Network Namespace را به یکدیگر متصل می‌کنیم. (این لینک‌ها را مشابه کابل LAN در نظر بگیرید). به‌طور مثال دستور اول یک لینک میان s1-eth1 و h1-eth0 ایجاد می‌کند.

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

```
ip link add s1-eth2 type veth peer name h2-eth0
```

در این مرحله h1-eth0 و h2-eth0 به namespace های مربوطه اختصاص می دهیم.

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

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

اکنون switch را می سازیم:

```
ovs-vsctl add-br s1
```

سپس لینکهای switch را به آنها اختصاص داده و فعال می کنیم:

```
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
```

اکنون برای درک بیشتر این موضوع، به سوالات زیر پاسخ دهید:

۱- چگونه می توان یک دستور را داخل یک namespace به خصوص مثلا h1 اجرا کرد؟

۲- چرا هنگام اجرای دستور زیر با پیغام Network is unreachable رو به رو می شویم؟

برای رفع این مشکل چه راهکاری پیشنهاد می دهید؟

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

۳- چگونه می توان به یک interface متعلق به host آدرس IP اختصاص داد؟

۴- virtual ethernet چیست؟

۵- با چه دستوری می توان یک virtual ethernet ساخت؟

۶- با چه دستوری می توانیم Virtual Ethernet را به یک Network Namespace دیگر

منتقل کنیم؟

۷- اجرا شدن دستور زیر چه اطلاعاتی به ما می دهد؟

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

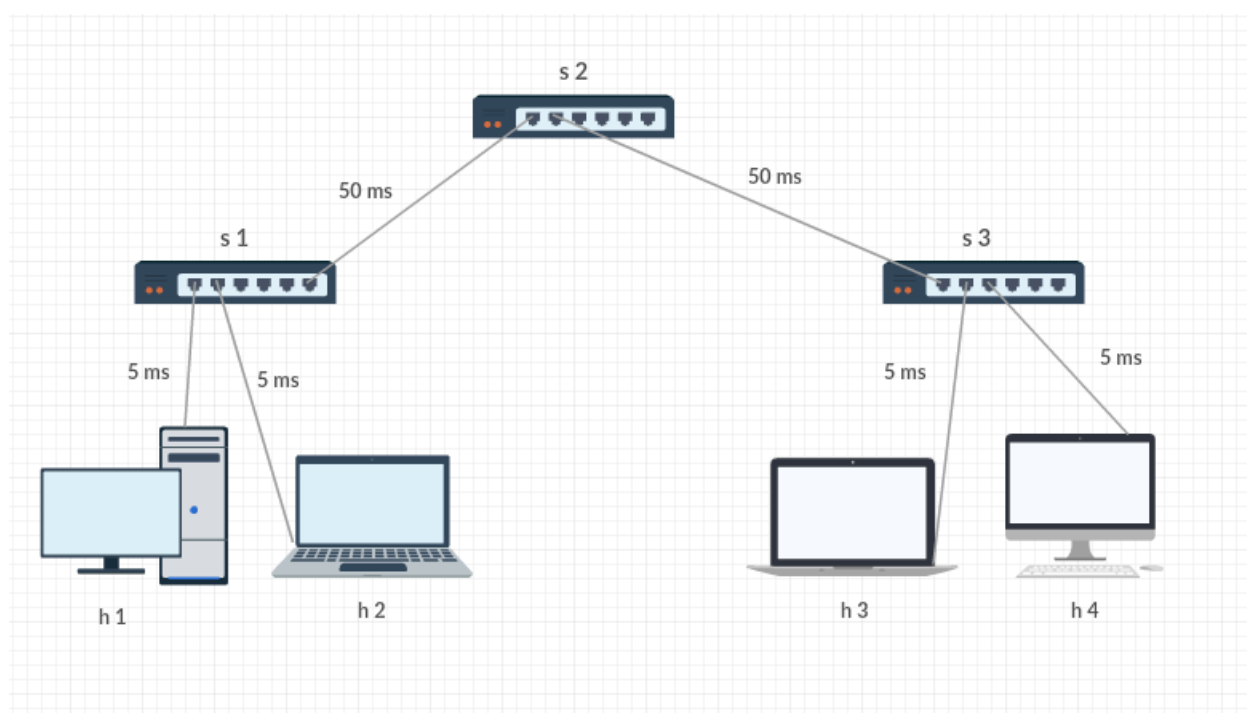
برای پاسخ به این سوالات پیشنهاد می شود به منبع [1] و منبع [2] مراجعه کنید. لطفا هر جا در گزارش نیاز به آوردن دستور دارید قالب آن را بنویسید .
مثلا (`ip netns exec <namespace_name> <command>`) و از کپی کردن دستورات و متن منابع بپرهیزید.

2. نوشتن یک شبیه ساز

- اکنون با استفاده از دستوراتی که در مرحله قبل آموختید،
- دو هاست بسازید به نامهای h1 و h2
 - دو سویچ بسازید به نامهای s1 و s2
 - آدرس IP هاست 10.0.0.1، h1 و هاست 10.0.0.2، h2 قرار دهید
 - لینک بین سویچها و هاستها را به این صورت برقرار کنید: h1 — s1 — s2 — h2
 - و s1-eth2 — s2-eth1 ، s1-eth1 — h1-eth0 لینکها را interface اسم s2-eth2 — h2-eth0 قرار دهید.
 - یک فایل bash بنویسید و دستورات خود را در آن قرار دهید به نحوی که با اجرا کردن آن، توپولوژی فوق ساخته شود.
 - برای این بخش مجازید حداکثر ۳۰ خط کد بنویسید و Commentها جزو تعداد خطوط محاسبه نمی شوند.
 - با اجرای دستور ping ارتباط بین دو هاست و کارکردن loopback interface را بررسی کنید.
- می توانید به کمک دستورهای 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 شبکه‌ی بزرگتر را در کامپیوتر خود ایجاد کنید، پیچیدگی‌های کار با namespace بسیار زیاد خواهد شد. ابزاری به نام Mininet از طریق ارایه API های ساده‌تر (به زبان پایتون) ساختن شبکه‌های مجازی را راحت‌تر کرده‌است. برای آشنایی با Mininet و دستورات آن [۳] را مطالعه کنید (موارد موردنیاز برای انجام این بخش در این منبع به‌طور کامل توضیح داده شده‌است) و پس از مطالعه به پیاده‌سازی موارد زیر بپردازید:



شکل ۲

- یک توپولوژی با ۴ هاست و ۳ سویچ مانند شکل ۲ ایجاد کنید.
- با استفاده از دستور nodes، هاست‌ها و با دستور nets، لینک‌های شبکه را مشاهده کنید.
- تاخیر هر لینک بر روی آن نشان داده شده‌است. 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. عکس این بخش را قرار دهید
عکس تمام خروجی‌هایی که گرفتید را همراه گزارش بفرستید. در گزارش توضیحات و نام عکس متناظر را بنویسید. و از آوردن عکس‌ها در گزارش خودداری کنید

۴. آشنایی با socket

در این بخش قصد داریم تا با نوشتن یک سرور و کلاینت ساده ضمن یادآوری socket programming تاخیر میان هاست‌ها را اندازه‌گیری کنیم. به خاطر سادگی این کار در زبان پایتون توصیه می‌کنیم این بخش را با زبان پایتون انجام دهید.

توپولوژی طراحی شده در بخش ۳ را در نظر بگیرید. (پهنای باند لینک میان switch ها را 1M و پهنای باند لینک میان هاست و switch را 5M در نظر بگیرید.) سرور روی پورت ۸۰۰۰ هرکدام از هاست‌ها منتظر اتصال کلاینت است. برنامه‌ی کلاینت در زمان اجرا، تعدادی آدرس IP از کنسول می‌گیرد. سپس به ترتیب به هر یک از آدرس‌های IP متصل و یک بسته‌ی کوچک (یک کاراکتر) ارسال می‌کند. هر سرور پس از دریافت این بسته با پیام مناسب به کلاینت جواب می‌دهد. کلاینت پس از دریافت پاسخ از سرور فاصله‌ی زمانی میان ارسال پیام به سرور و دریافت پاسخ آن را محاسبه و به واحد میلی ثانیه در خروجی استاندارد چاپ می‌کند. سپس کلاینت عملیات فرستادن بسته، اندازه‌گیری زمان و نمایش آن را با یک بسته‌ی بزرگ تکرار می‌کند. اختلاف این دو زمان را محاسبه و چاپ کنید. اختلاف این دو زمان به صورت تقریبی نشان دهنده‌ی کدام یک از انواع تاخیر است؟ پاسخ خود را به همراه ذکر دلیل در گزارش بنویسید.

توجه داشته‌باشید که شما باید یک نسخه از سرور را بر روی تمام هاست‌های ساخته شده در توپولوژی مرحله‌ی قبل اجرا کنید اما کافیست کلاینت را فقط بر روی یکی از هاست‌ها اجرا کرده و مطمئن شوید اعداد چاپ شده صحیح و منطقی هستند. طبق آدرس‌های IP دریافت شده ممکن است یک کلاینت به سروری که روی همان هاست در حال اجراست متصل شود.

5. توضیحات تکمیلی

- پروژه دو نفره است و نمره افراد لزوماً یکی نیست.
- تنها استفاده از زبان‌های برنامه نویسی پایتون و جاوا مجاز است.
- توضیحات خواسته شده در هر بخش را در گزارش خود بیاورید. در صورتی که کل توضیحات از ۳ صفحه A4 بیشتر شود نمره‌ی قابل قبولی را از دست خواهید داد.
- واضح است که برای تمیز بودن گزارش و واضح بودن کار نمره در نظر گرفته شده است.
- برای ما خیلی مهم است که حاصل کار خودتان را تحویل دهید. برای همین به شدت با تقلب برخورد می‌شود.
- توصیه می‌کنیم هر دو نفر با بخش استفاده از آشنا باشند، زیرا در پروژه‌های Mininet بعدی قطعاً به آن احتیاج خواهید داشت
- حتماً گزارش را فارسی بنویسید. نمره گزارش به زبانی غیر از فارسی صفر است.

6. منابع

1. <http://blog.scottlowe.org/2013/09/04/introducing-linux-network-namespaces>
2. <https://blogs.igalia.com/dpino/2016/04/10/network-namespaces>
3. <http://mininet.org/walkthrough/#part-1-everyday-mininet-usage>
4. <https://hitos.ir/63/docker> -چیسـت-آشنایـی-و-نصب-داکر-قسمت-اول
5. <https://www.infoworld.com/article/3204171/linux/what-is-docker-linux-containers-explained.html>
6. <http://conferences.sigcomm.org/sigcomm/2014/doc/slides/mininet-intro.pdf>
7. <https://sreeninet.wordpress.com/2014/11/30/mininet-internals-and-network-namespaces/> (برای دسترسی به این لینک نیاز به فیلتر شکن دارید)