

قسمت ۱:

سوال ۱: برای اجرای دستور در یک namespace خاص میتوان به شکل زیر عمل کرد که در بخش command دستوری که قصد اجرای آن را داریم مشخص میکنیم.

```
ip netns exec <network_namespace_name> <command>
```

برای مثال اگر namespace ما h1 باشد دستور به شکل زیر درمی آید:

```
ip netns exec h1 <command>
```

سوال ۲:

قبل از اجرای این دستور باید ip مربوط به interface را ست کنیم. چون این کار انجام نشده است با پیغام خطای Network is unreachable مواجه میشویم. برای حل آن میتوانیم از دستورات زیر استفاده میکنیم:

```
ip netns exec <network_namespace_name> ifconfig <veth_interface_name>  
ip_address
```

```
ip netns exec <network_namespace_name> ifconfig lo up
```

سوال ۳:

پس از انتقال virtual ethernet به namespace مورد نظر باید interface های مربوط به آن را تنظیم کنیم. در حالت کلی برای تنظیم این interface ها از دستور ip netns exec استفاده میکنیم. پس از آن نام namespace مورد نظر را مینویسیم و دستور مورد نظرمان را مشخص میکنیم. در این جا که قصد داریم یک آدرس ip را اختصاص دهیم از ifconfig استفاده می کنیم. فرمت کلی این دستور به شکل زیر است:

```
ip netns exec <network_namespace_name> ifconfig <veth_interface_name>  
ip_address up
```

namespace ما h1 باشد و نام veth ما veth1 باشد دستور به شکل زیر درمی آید:

```
ip netns exec h1 ifconfig veth1 10.0.0.2 up
```

سوال ۴:

Virtual ethernet یک اجرای مجازی از ethernet است و اجازه می دهد ارتباط بین پارتیشن های منطقی و ماشین های مجازی بدون نیاز به اختصاص دادن و configure کردن سخت افزار فیزیکی روی آن پارتیشن ها امکان پذیر باشد. یکی از مسایل network namespace ها این است که فقط یک interface میتواند در یک زمان به یک namespace اختصاص داده شود. مثلا اگر root دارای namespace ای با نام eth0 باشد که از طریق آن به دنیای بیرونی دسترسی پیدا میکند؛ فقط برنامه هایی که دارای root namespace هستند میتوانند به اینترنت دسترسی پیدا کنند. یک راهی که برای حل این مساله وجود دارد این است که ارتباط یک namespace را با root namespace از طریق یک جفت veth برقرار کنیم. یک veth pair مانند یک لوله عمل میکند که یک سمت آن به root network namespace اختصاص داده میشود و سمت دیگر آن به namespace مورد نظرمان متصل میشود.

سوال ۵:

ip link add <your favorite name> type veth peer name <your favorite name>

سوال ۶:

با اجرای دستور بیان شده در سوال ۵ یک جفت از veth interface ها ساخته میشوند که به global namespace تعلق دارند. برای انتقال آن ها به یک network namespace میتوانیم از دستور زیر استفاده کنیم:

ip link set <veth interface name> netns <network namespace name>

مثلا اگر بخواهیم veth1 را به namespace با نام h1 منتقل کنیم دستور به شکل زیر درمی آید:

ip link set veth1 netns h1

سوال ۷:

این دستور interface و آدرس های مرتبط با نام namespace را به ما نشان می دهد و هیچ interface یا آدرسی را که مربوط به global namespace باشد را به ما نشان نمیدهد. اجرای این دستور با حذف بخش <network_namespace_name> ip netns exec <network_namespace_name> global namespace را به ما نشان میدهد.

عکس خواسته شده از اجرای دستور ifconfig با نامی به همین عنوان در پوشه موجود است.

قسمت ۲:

ساده ترین کاربرد دستور ping عیب یابی و تست ارتباط بین دو نقطه از شبکه است. در واقع ping بررسی می کند که آیا میزبان فعال است یا نه یا مثلا تاخیر رسیدن پکت به میزبان چقدر است و اینکه آیا packet loss داریم یا خیر.

طرز کار ping به این صورت است که ابتدا بسته ICMP Echo Request را به سمت کامپیوتر مقصد ارسال میکند. اگر کامپیوتر مقابل این بسته را دریافت کند بسته ای به نام ICMP Echo Replay را به سمت مبدا ارسال میکند و خبر دریافت پکت اطلاعات را به طور خودکار ارسال میکند. در این بخش با استفاده از دستورهایی که در قسمت اول توضیح داده شد (ساختن هاست ها و سویچ ، ایجاد veth و منتقل کردن veth به داخل namespace ها ، ست کردن آی پی و ارتباط برقرار کردن بین آن ها) یک ارتباط بین دو هاست برقرار کردیم و اینکه پکت ها بین این دو هاست ارسال یا دریافت شده اند یا پکتی از دست رفته است؛ قابل مشاهده است. همچنین مدت زمان جابجایی پکت بین دو هاست نیز به میلی ثانیه دیده میشود.

قسمت ۳:

در این قسمت سری کارهای زیر را به ترتیب انجام دادیم تا توپولوژی خواسته شده در صورت سوال را بسازیم

1. sudo dhclient eth1
2. sudo ssh -X -v mininet@<vm's ip address>
3. scp path/of/python/file mininet@<mininet ip>:~/path/to
4. sudo mn - -custom ./filename.py - -topo mytopo - -link tc

DHCP

پروتکل ای است که این امکان را فراهم میکند که هاست به یک سرور مرکزی وصل شده و یکی از آدرس های آی پی موجود در آن سرور را بردارد

دستور خط ۲ ما را از ترمینالی بیرون ماشین مینینت به این ماشین وصل میکند تا بتوانیم دستوراتی که می خواهیم را از طریق این ترمینال وارد کنید از طرفی اگر این کار را نکنیم دستوراتی مثل xterm را نمیتوانیم

اجرا کنیم چون خود مینیت فقط همان یک ترمینال/صفحه نمایش را دارد و به خارج از آن یا هیچ رابط گرافیکی نمی‌تواند متصل شود.

دستور خط ۳ کد ما را از کامپیوتر ما به داخل مینیت کپی می‌کند (باید از طریق یک ترمینال که دستور ۲ روی آن اجرا نشده است این عمل را انجام دهیم)

خط ۴ دستوری دارد که توپولوژی نوشته شده توسط ما را اجرا می‌کند، اگر بخش link tc — را ننویسیم ممکن است به bw مشخص شده در لینکها ایراد بگیرد و topo mytopo — نامی است که با صدا کردن آن بخش مربوط به کد توپولوژی ما اجرا می‌شود.

Pingall:

این دستور از توابع/دستورات مینیت است که به صورت زیر تعریف شده است:

```
Def mininet.net.Mininet.pingAll (self, timeout=None)
```

این دستور به این شکل عمل می‌کند که میان هر ۲ تا هاست موجود ping می‌کند و نتیجه کل تعداد پکت‌های گم شده و موفق دریافت شده را بیان می‌کند.

عکسی با نام همین دستور نحوه اجرای آن را برای توپولوژی پیش فرض مینیت نشان می‌دهد.

Xterm:

این دستور ترمینال ماشین هاست شماره x را باز می‌کند و ما می‌توانیم ستوراتی را که اگر واقعا آن هاست وجود داشت روی ترمینالش اجرا می‌کردیم روی این ترمینال اجرا کنیم.

۲ عکس با نامی حاوی نام این دستور در پوشه‌ی عکس‌ها موجود اند.

قسمت ۴:

زمانی که توپولوژی را اجرا کردیم و توابع سرور و کلاینت را در هاست‌ها اجرا کردیم، می‌بینیم که در حالتی که ساینز پکت ارسالی بزرگ می‌شود تاخیر نیز بیشتر می‌شود علت این موضوع را بررسی می‌کنیم:

برای توضیح مثال استاد را مطرح می‌کنیم که گروه دانشجویان که در صفی از کلاس خارج می‌شوند پکت ما هستند و تعداد آن‌ها طول پکت، این افراد پس از خروج از در به سمت مقصد خود می‌روند که مثلاً انقلاب است. حال در هر دو ارسال مقصد یکی است و با یک وسیله‌ی نقلیه هم می‌روند چون پهنای باند و تاخیر زیاد نشده است پس علت تاخیر طول پکت است یا به عبارت دیگر تاخیر ایجاد شده از عبور از در.

با استفاده از اصطلاحات شبکه در واقع این اختلاف بین زمان‌ها transmission delay است که به طول پکت بستگی دارد.

از طرفی زمانی که به تاخیرها نگاه می‌کنیم می‌بینیم که زمانی که کد کلاینت را روی هاست ۱ اجرا می‌کنیم زمان رسید پکت به هاست ۱ و ۲ با اختلاف زیادی از هاست ۲ و ۴ کمتر است و این موضوع به این علت است که برای فرستان پکت به هاست ۳ و یا ۴ از هاست ۱ (کلاینت ۱) باید از یک switch اضافه عبور کنیم که تاخیری برابر با ۱۰۰ میلی ثانیه به تاخیر اضافه می‌کند (به دلیل تاخیر لینک‌ها).

تصویر مربوط به این بخش با عنوان client-server موجود است.