



زمان تحویل: ۲۵ اردیبهشت

بسمه تعالی
شبکه‌های کامپیوتری - بهار ۹۶
پروژه شماره ۵ - درخت پوشا



زمان ارسال پروژه: ۱۵ اردیبهشت



در این پروژه قصد داریم تا با استفاده از کنترلر Floodlight و با استفاده از شبیه سازی شبکه به کمک مینی نت مک آدرس هاست‌های یک شبکه را پیدا کرده و یک spanning tree در شبکه درست کنیم. برای این کار باید یک توپولوژی جدید به مینی نت اضافه کنید و سوییچ‌های آن را طوری تنظیم کنید که یک spanning tree در آن توپولوژی شبیه‌سازی شود.

۱. پیش نیازها

داشتن نرم افزار IntelliJ

داشتن Java 8

۲. راه اندازی Floodlight

۱. دریافت فایل پروژه Floodlight با استفاده از دستور:

```
git clone git://github.com/floodlight/floodlight.git
```

۲. اجرای دستورات زیر:

```
git submodule init
```

```
git submodule update
```

۳. پوشه Floodlight را در IntelliJ ، import کنید (همان‌طور که در کلاس حل توضیح پروژه دیدید)

دقت کنید که در مرحله دوم import باید گزینه دوم، یعنی import project from external model را انتخاب و سپس maven را انتخاب کنید.

۴. هنگامی که پروژه در IntelliJ باز شد باید tab ای با نام maven projects را باز کرده و در آن، داخل قسمت Floodlight قسمتی با نام lifecycle وجود دارد. داخل قسمت lifecycle با اجرای گزینه install پروژه build خواهد شد.

۵. حال با دستور زیر Floodlight را در IntelliJ اجرا کنید:

```
java -jar target/floodlight.jar
```

۳. راه اندازی مینی نت

۱. قبل از اجرای مینی نت وارد virtualbox setting شوید و سپس در قسمت network یکی از adapter ها را بر روی NAT و یکی دیگر را بر روی bridge adapter با انتخاب Name: en0 تنظیم کنید.
۲. مینی نت را اجرا کنید و با نام کاربری و رمز عبور یکسان mininet لاگ این کنید و سپس با دستور زیر شبیه ساز شبکه را با توپولوژی custom خود راه بیندازید.

```
sudo mn --custom ~/shangula.py --topo mytopo --  
controller=remote,ip=192.168.110.2,port=6653 --  
switch=ovsk,protocols=OpenFlow13
```

دقت کنید که ip آدرس ماشینی است که floodlight در حال اجرا بر روی آن است. برای قسمت ۴ از یک توپولوژی ساده و بدون حلقه استفاده کنید. دستورات مربوط به توپولوژی بخش ۵ در همان قسمت ارائه شده اند.

۴. Learning Switch

همان طور که می دانید در حال پیاده سازی یک شبکه ی SDN هستیم. در این شبکه ها ادوات شبکه (مانند سوئیچ ها) به تنهایی مسئول تصمیم گیری در مورد نحوه ی هدایت بسته ها در شبکه نیستند، بلکه آنها پس از مکاتبه با یک کنترلر مرکزی (در این پروژه Floodlight) نحوه ی هدایت بسته را از آن کنترلر می پرسند و در حافظه ی خود ذخیره می کنند. به این ترتیب در این شبکه ها Control Plane از Data Plane جدا شده است.

رفتار یک سوئیچ در یک شبکه ی SDN به این صورت است که پس از دریافت یک بسته اگر بداند که چطور باید آن را هدایت کند (قبلاً در مورد آن از کنترلر پرسیده باشد) به همان ترتیب آن را ارسال خواهد کرد. وگرنه آن بسته را برای کنترلر ارسال می کند و منتظر پاسخ می ماند. پس از دریافت پاسخ، آن را در Forwarding Table خود ذخیره می کند.

کنترلر پس از دریافت هر بسته آن را از طریق ماژول هایش بررسی و نحوه بر خورد با آن را مشخص می کند. در این بخش شما باید ماژول LearningSwitch.java را راه اندازی کنید. در مورد ماژول های Floodlight یک دستورالعمل در سایت درس قرار داده شده است که در این زمینه به شما کمک خواهد کرد. این ماژول همان طور که از نام آن پیداست رفتار یک سوئیچ لایه ۲ را پیاده سازی می کند. به این ترتیب که در صورت دریافت هر بسته ابتدا مشخص می کند که آدرس mac فرستنده و گیرنده چیست (به تابع processPacketInMessage نگاه کنید). در ادامه مشخص می کند که فرستنده بر روی کدام پورت سوئیچ قرار دارد که برای ارسال های بعدی به او مورد استفاده قرار می گیرد، کنترلر این اطلاع را در داده ساختاری به نام swMap که برای هر سوئیچ به صورت جدا ساخته است ذخیره می کند (تابع addToPortMap را ببینید). سپس برای ارسال بسته به مقصد در همین داده ساختار به دنبال آدرس mac مقصد می گردد (تابع getFromPortMap را ببینید) و در صورت یافتن، بسته را فقط بر روی پورت مربوطه ارسال می کند و در غیر این صورت آن را flood می کند. دقت کنید که خود کنترلر در عملیات ارسال نقشی ندارد، بلکه این اطلاع را برای سوئیچ می فرستند و سوئیچ طبق دستور کنترلر عمل می کند.

پس از راه اندازی این ماژول host ها باید بتوانند با یکدیگر ارتباط برقرار کنند (از دستور ping استفاده کنید و تفاوت رفتار شبکه قبل و بعد از راه اندازی ماژول را مشاهده کنید). به سؤالات زیر پاسخ دهید:

۱. در مورد ساختار Forwarding Table که در سوئیچ‌های شبکه‌های SDN مورد استفاده قرار می‌گیرد تحقیق کنید. فیلدهای مهم آن را نام برده و مختصر توضیح دهید.
۲. در حین ارتباط دو host کنترلر را قطع کنید و رفتار شبکه را مشاهده کنید. آیا ارتباط آن‌ها بلافاصله قطع می‌شود یا ادامه دارد؟ برای پاسخ به این بخش در مورد مفهوم timeout برای rule‌های داخل Forwarding Table جستجو کنید و در گزارش ارائه کنید.
۳. در مورد دستور `return Command.CONTINUE` توضیح دهید.

۵. درخت پوشا

در این قسمت شما باید ابتدا یک توپولوژی در می‌نیت ایجاد کنید. این توپولوژی باید حتماً باید دارای یک حلقه و بیش از ۴ سوئیچ باشد که به هر سوئیچ یک host متصل باشد. دوباره floodlight را به همراه ماژول Learning Switch راه‌اندازی کنید و سعی کنید بین دو host ارتباطی برقرار کنید. در مرحله خواهید دید که این ماژول تنها بر روی توپولوژی‌های بدون حلقه کار می‌کند.

در ادامه شما باید یک ماژول جدید به نام MySpanningTree ایجاد کنید. برای شروع از کدهای ماژول LearningSwitch استفاده کنید. شما باید ابتدا توپولوژی شبکه را پیدا کنید و سپس بر روی آن یک درخت پوشا پیدا کنید. به این ترتیب ماژول شما وقتی آدرس مقصد یک بسته را نمی‌داند دیگر نباید آن را بر روی تمام لینک‌ها flood کند، بلکه آن را فقط بر روی لینک‌هایی flood می‌کند که در درخت پوشا قرار دارند. البته در شبکه‌های کنونی پروتکل‌های درخت پوشا پس از اجرا تعدادی از لینک‌ها را غیر فعال می‌کنند ولی شما باید با درج قوانین مناسب در سوئیچ اطمینان حاصل کنید که بر روی آنها عمل flood رخ نمی‌دهد.

شما می‌توانید فرض کنید که توپولوژی را از پیش می‌دانید و به این ترتیب عمل flood را فقط بر روی پورت‌هایی که از قبل می‌دانید flood بر روی آنها منجر به ایجاد ارسال بر روی حلقه نمی‌شود انجام دهید. یعنی ماژول شما دیگر عمل یافتن درخت پوشا را انجام نخواهد داد و فقط باید بداند که بر روی کدام لینک‌ها نباید flood کند. به این ترتیب ماژولی که شما ایجاد خواهید کرد فقط برای توپولوژی که شما ایجاد کرده‌اید کار خواهد کرد. برای دریافت تمام نمره این بخش، ایجاد این ماژول استاتیک کافی است. در صورتی که علاقه‌مند هستید یک ماژول بنویسید که بر روی هر توپولوژی کار کند باید توپولوژی شبکه را بدست بیاورید و بر روی آن الگوریتم درخت پوشا را اجرا کنید و سپس لینک‌هایی که نباید بر روی آنها flood رخ دهد را مشخص کنید.

برای اینکه توپولوژی شبکه را بدست بیاورید باید از ماژول Link Discovery استفاده کنید. به این جهت می‌توانید پس از ایجاد یک نمونه از سرویس `ILinkDiscoveryService` به تمام لینک‌های داخل شبکه و اطلاعات مربوط به آنها (از طریق تابع `getLinks()`)، تمام سوئیچ‌های داخل شبکه (از طریق `getAllSwitchesDpids()`) و سایر اطلاعات شبکه دسترسی پیدا کنید. برای اطلاعات بیشتر فایل آموزشی [۱] می‌تواند کمک کننده باشد. ایجاد ماژول درخت پوشای دینامیک مشمول دریافت نمره اضافه خواهد بود.

دقت کنید در هیچ کدام از قسمت‌های فوق ماژول مسیریاب شما نباید در مورد محل قرارگیری host‌ها پیشفرضی داشته باشد.

به سؤالات زیر پاسخ دهید:

۱. چرا پروتکل Learning Switch بر روی توپولوژی‌های داری حلقه کار نمی‌کند؟
۲. استفاده از درخت پوشا امکان مسیریابی لایه ۲ بر روی توپولوژی‌های دارای حلقه را فراهم می‌کند. به نظر شما این روش چه معایبی دارد؟

۳. (امتیازی) یک توپولوژی ممکن است چندین درخت پوشا داشته باشد. درخت پوشایی که استفاده می شود در زمان تأخیر بین host ها می تواند تأثیرگذار باشد. آیا می توانید با انجام یک آزمایش بر روی مینی نت این مطلب را نشان دهید؟ به نظر شما چه ویژگی هایی می تواند در انتخاب یک درخت پوشا تأثیرگذار باشد؟

۶. تکمیلی

۱. پروژه دو نفرست و نمره افراد لزوما یکی نیست.
۲. حتما توپولوژی منحصر گروه خودتان باشد و حتما طراحی کلی سیستم شبکه خود در کنار کد توپولوژی و ماژول آپلود کنید.
۳. یک گزارش کوتاه از کل روند کارتان و خروجی های لازم (عکس خروجی ها) در کنار کد های خود باید آپلود کنید. گزارش حداکثر ۲ صفحه باشد. کسانی که بخش امتیازی را انجام می دهند یک صفحه بیشتر هم دارند.
۴. کل موارد لازم را به صورت یک فایل فشرده در سایت درس آپلود کنید.
۵. سوالات خود را در فروم مطرح کنید.

References

[1] C. Liang. (2014). Floodlight tutorial [Online]. Available:
www.cs.duke.edu/courses/fall14/compsci590.4/notes/slides_floodlight_updated.pdf