



به نام خداوند هستی بخش

شبکه‌های کامپیوتری - تابستان ۹۹

پروژه شماره ۳

مهلت تحویل: ۲۵ مرداد ماه



۱. معرفی کنترلر Ryu

کنترلر RYU (ری-یوه) یک کنترلر مبتنی بر کامپوننت و به زبان برنامه‌نویسی پایتون می‌باشد. این کنترلر اجزای نرم افزاری را با API تعریف شده فراهم می‌کند که ایجاد برنامه‌های جدید مدیریت و کنترل شبکه را برای توسعه دهندگان آسان می‌کند. Ryu از پروتکل‌های مختلفی برای مدیریت دستگاه‌های شبکه مانند OpenFlow، Netconf، OF-config و موارد دیگر پشتیبانی می‌کند. درباره OpenFlow، Ryu از ۱،۰، ۱،۲، ۱،۳، ۱،۴، ۱،۵ پشتیبانی می‌کند. همه کدها تحت مجوز Apache 2.0 آزادانه در دسترس هستند. کد این کنترلر در GitHub توسط لینک زیر در دسترس می‌باشد:

<https://github.com/osrg/ryu.git> (or github.com/faucetsdn/ryu)

۲. نصب کنترلر Ryu

برای نصب کنترلر ریو در ابتدا می‌بایست نیاز به نصب Python و Pip می‌باشد. دستورات زیر را به ترتیب در ترمینال وارد کنید تا ابتدا کدهای Ryu از GitHub دریافت و سپس نصب شود:

```
cd
pip install eventlet msgpack-python netaddr oslo.config routes six webob
git clone git://github.com/osrg/ryu.git
cd ryu
python ./setup.py install
```

۳. اجرای کنترلر Ryu

برای اجرای کنترلر RYU کافی است دستور زیر را در ترمینال وارد کنید. با این دستور شما کنترلر ریو را تبدیل به یک سوئیچ معمولی نموده اید:

```
cd ryu  
ryu-manager ryu/app/simple_switch.py
```

۴. اجرای Mininet

برای تست صحت عملکرد کنترلر در ترمینال دیگری دستور زیر را وارد نمایید تا Mininet باز شود. همانطور که می‌دانید با استفاده از مینی نت می‌توانید سوئیچ و هاست مجازی ایجاد کنید و سوئیچ‌های مجازی که مجهز به OpenFlow می‌باشند را به کنترلر متصل کنید تا آن‌ها را کنترل کنند:

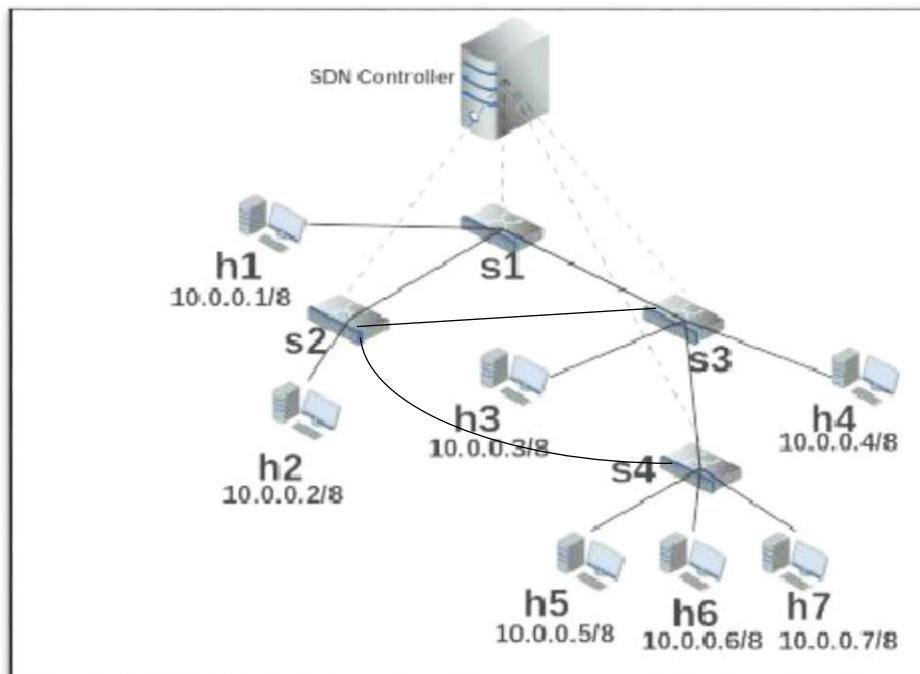
```
sudo mn --controller remote,ip=127.0.0.1,port=6653
```

۵. تعریف پروژه

همانطور که می‌دانیم بر خلاف شبکه‌های سنتی که مسیریابی در آن‌ها توسط روترها به صورت توزیع شده انجام می‌گردد، در SDN مسیریابی (Routing) به صورت متمرکز در داخل کنترلر انجام می‌شود. کنترلر تمام اطلاعات و توپولوژی شبکه را در اختیار داشته و با بروزرسانی جداول flow داخل سوئیچ‌های SDN، مسیریابی را انجام می‌دهد. لذا در حقیقت مسیریابی مشابه آنچه در شبکه‌های سنتی وجود دارد، انجام نمی‌شود.

در این پروژه ما قصد داریم، تا یک مسیریابی ساده با استفاده از الگوریتم Dijkstra را جهت مسیریابی بین گره‌های شبکه SDN در داخل کنترلر Ryu پیاده‌سازی نماییم. در این مسیریابی، میزان تاخیر بین تمام گره‌ها همواره محاسبه شده و مسیر با کوتاه‌ترین تاخیر جهت ارسال بسته‌ها از گره مبدا به گره مقصد انتخاب می‌شود.

برای پیاده‌سازی این پروژه، توپولوژی زیر را در نظر بگیرید:



فرضیات زیر جهت پیاده‌سازی شبکه در نظر گرفته شود:

۱- پهنای باند هر کدام از لینک‌ها به صورت تصادفی هر ۱۰ ثانیه یکبار بین 1Mbps الی 5Mbps در نظر گرفته شود.

۲- به گره‌های h1 الی h7 به دلخواه از subnet مربوطه یک آدرس IP اختصاص دهید.

۳- هر گره در دوره زمانی ۱۰۰ میلی ثانیه، یک گره را به صورت تصادفی انتخاب و سپس داده به اندازه ۱۰۰ کیلو بایت را از طریق TCP به آن ارسال می‌کند.

۴- پیاده سازی کنترلر باید به گونه ای باشد که با دریافت پکت از سمت هاست، با در نظر گرفتن توپولوژی شبکه و اجرا کردن الگوریتم یافتن کوتاه ترین مسیر، پورت ورودی، پورت خروجی و مسیر پیشنهادی را در ترمینال چاپ کند.

۵- هم چنین پیاده سازی باید این قابلیت را داشته باشد که کاربر بتواند وزن هر یک از یال‌های اتصالی گره‌ها (برای تشکیل گراف توپولوژی شبکه) را به صورت دستی از کاربر دریافت کند و الگوریتم کوتاه‌ترین مسیر را بر اساس وزن یال‌ها اجرا کند.

۶. نتیجه‌گیری

مطابق توضیحات داده شده، شبکه ایجاد شده و برای مدت ۱ دقیقه و برای ۵ بار اجرا گردد. نتایج زیر استخراج و در گزارش ذکر گردد:

۱- نمودار میزان زمان ارسال بسته‌ها از گره به گره دیگر محاسبه گردد.

۲- نرخ بروزرسانی جدول flow داخل هر سوئیچ

۳- تاریخچه میر ارسالی بسته‌ها از هر گره به گره دیگر، یعنی بدانیم هر بسته از چه مسیری (و باتوجه به مورد ۱ نیز در چه مدت زمانی) بین گره‌ها ارسال شده است.

با توجه به محدودیت وقت برای این پروژه و عدم امکان پاسخگویی به سوالات دانشجویان، هر گونه ابهام که در پیاده سازی وجود دارد را با پیش فرضات خود انجام داده و این پیش فرضات را در مستندات به طور کامل توضیح دهید.

۷. مراجع و لینک‌ها:

<https://ryu-sdn.org/>

<https://osrg.github.io/ryu-book/en/html/index.html>

موفق باشید