



به نام خداوند هستی بخش

شبکه‌های کامپیوتری - تابستان ۹۹

پروژه شماره ۳

مهلت تحویل: ۲۹ 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 سوئیچ های مجازی که مجهز به OpenFlow می‌باشند را به کنترلر متصل کنید تا آن‌ها را کنترل کنند:

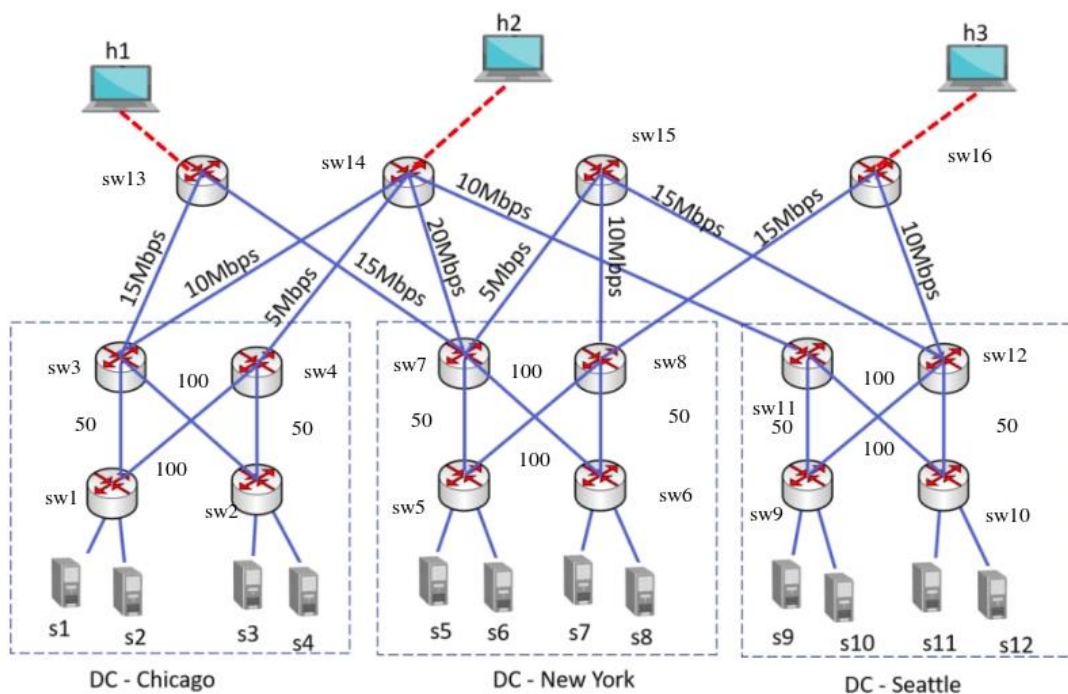
```
sudo mn --controller remote,ip=127.0.0.1,port=6653
```

۵. بروزرسانی پروژه

تمام کدهای لازم برای پروژه، در لینک زیر موجود است:

http://csie.nqu.edu.tw/smallko/sdn/dijkstra_ryu.htm

توجه فرمایید پیاده سازی الگوریتم در محیط ریو و مینی نت به همراه تصاویر اجرای کد موجود است. به نحوه اجرای کد توجه فرمایید. در هنگام اجرا، یکی از از ترمینال های شما باید اپلیکیشن کوتاه ترین مسیر را توسط فرآیند ریو اجرا کنید و در ترمینال دیگر باید ساختار توپولوژی شبکه بر روی مینی نت اجرا شود. به منظور ساده سازی فرآیند اجرای کد، از ساختار درختی زیر برای توپولوژی مینی نت استفاده میکنیم که فاقد حلقه است و پهنای باند لینک‌ها نیز ثابت می‌باشد:



در این میان، شما باید خط به خط مراحل اجرای کد ضمیمه شده را مطالعه کنید. ممکن است نیاز باشد به مستندات ریو هم رجوع کنید. پس از مطالعه کد نیاز است که کد را در محیط مینی نت سیستم خود اجرا کنید و خطاهای احتمالی را رفع کنید. در این مرحله شما باید قادر باشید که کوتاه ترین مسیر را در یک توپولوژی ساختار درختی پیدا کنید. به منظور شفافیت بیشتر، بهتر است با چاپ پکت های دریافتی (به عنوان مثال پینگ از یک هاست به هاست دیگر) در ترمینالی که اپلیکیشن ریو آن در حال اجرا هست، روند منطقی اجرای کد را دنبال کنید. حتما در هنگام مطالعه کد قسمت هایی را که مهم میدانید در ترمینال چاپ کنید. با این فرض، کنترلر بعد از دریافت پکت های هر هاست، اطلاعات را به کاربر نشان میدهد. تحویل پروژه شما به این گونه است که ابتدا باید خطاهای احتمالی اجرای کد فوق را تصحیح کنید. کد موجود را به دقت و با خوانایی بالا مستند سازی کنید و نهایتا دستورات چاپ اضافی برای پیگیری منطق الگوریتم را به کد موجود اضافه کنید. فایل نهایی شما باید قابلیت اجرا با مستندات کافی برای ارائه بدون نقص داشته باشد. هم چنین، دانشجویانی که الگوریتم کد فوق را برای توپولوژی شبکه ای که داری حلقه می باشد و مکانیزم های حذف حلقه در آن در نظر گرفته می شود، تغییر دهند، نمره اضافی خواهد گرفت. راهنمای ذیل به این دانشجویان کمک شایانی می کند:

https://osrg.github.io/ryu-book/en/html/spanning_tree.html

لذا باید موارد زیر به عنوان خروجی پروژه تحویل داده شود:

- ۱- کد اجرایی صحیح همراه با کامنت در خطوط و تولید خروجی در کنسول مطابق با مطالب گفته شده در بالا که نشان دهنده مسیریابی بین گره ها در حین اجرا کد می باشد.
- ۲- توضیح خط به خط کدهای داده شده و توضیح عملکرد آنها
- ۳- اجرای کد بر روی ساختار توپولوژی داده شده
- ۴- اسکرین شات های گرفته شده از اجرای صحیح کدها