



## پروژه چهارم

مریم سعیدمهر  
شماره دانشجویی: ۹۶۲۹۳۷۳

### ۱ نصب ODL

مراحل نصب:

۱. لینک دانلود نسخه *0.4.1 - Beryllium - SR1* (۲۹۷ مگابایت)

۲. فایل فشرده را unzip میکنیم

۳. فیچرهای مورد نیاز را نصب میکنیم. (مراجعه به بخش ۲)

### ۲ نصب features

• *odl - l2switch - all*

• *vtn - manager - rest*

• *vtn - manager - neutron*

• *odl - dlux - all*

روند نصب فیچرها:  
بعد از دانلود و خارج کردن آن از حالت فشرده دستورات زیر را به ترتیب در ترمینالی که در پوشه unzip شده باز کرده ایم، اجرا میکنیم:

۱. *./bin/karaf*

۲. اگر پیغام زیر ظاهر شد یعنی سرور ODL ران شده است روی پورت ۶۶۳۳

```

Terminal File Edit View Search Terminal Help
[sudo] password for maryam:
karaf: JAVA_HOME not set; results may vary

Hit '<tab>' for a list of available commands
Hit '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown OpenDaylight.

opendaylight-user@root>

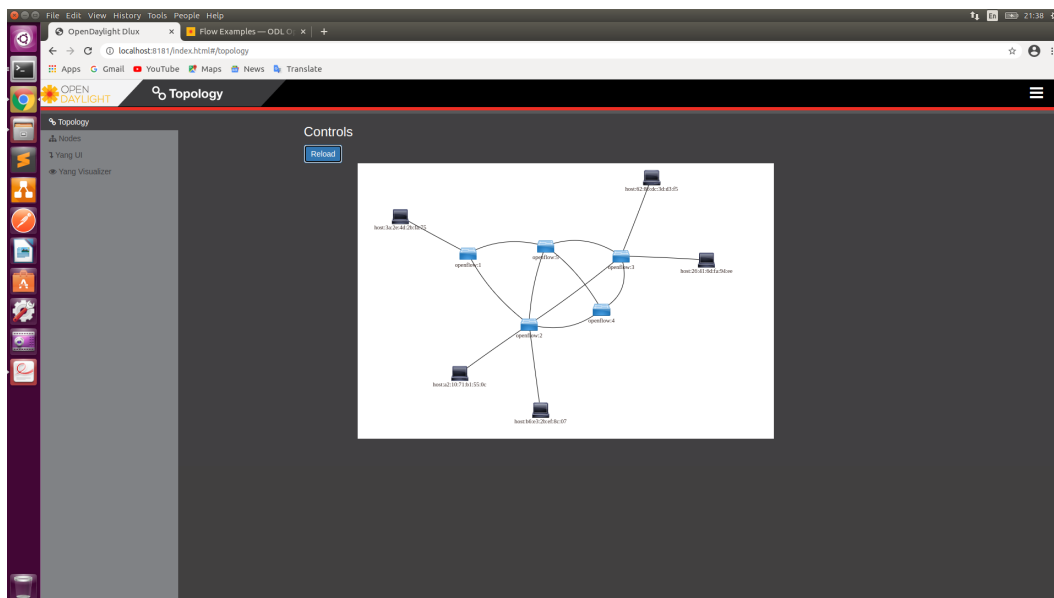
```

شکل ۱: ODL is listening on port 6633

۳. حالا دستور `feature : install <feature_name>` را اجرا میکنیم و به جای `<feature_name>` نام فیچرهایی که در بخش ۲ آمده است را میگذاریم

۳ ساخت توپولوژی با ۵ هاست و سویچ در مینی نت

فایل `mesh.py` در پوشه آپلود شده ضمیمه شده است. شکل توپولوژی طراحی شده به فرم زیر است :



شکل ۲: توپولوژی شبکه مورد نظر

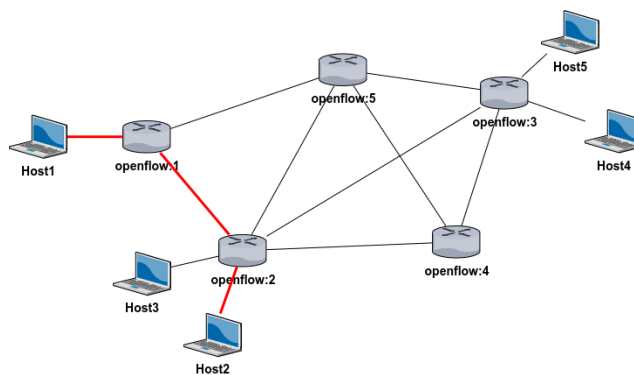
برای ایجاد توپولوژی و اتصال کنترلر ODL به آن دستور :

```
sudo mn --custom mesh.py --topo sample --controller=remote,ip=127.0.0.1,port=6633
```

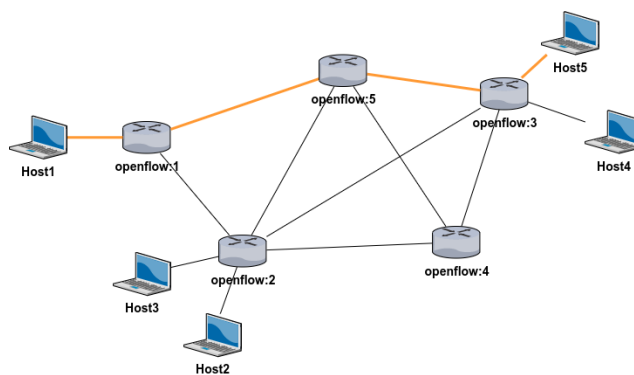
شکل ۳: دستور ایجاد توپولوژی و اتصال کنترلر ODL به آن

#### ۴ کانفیگ کردن سویچ ها

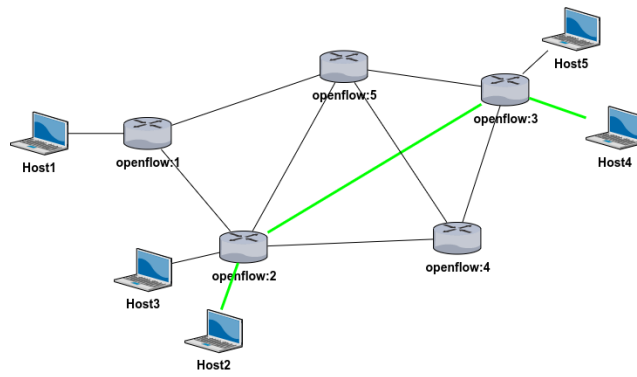
فایل *flow.py* در پوشه آپلود شده ضمیمه شده است.  
مسیرهایی که در نظر گرفته ام عبارت اند از :



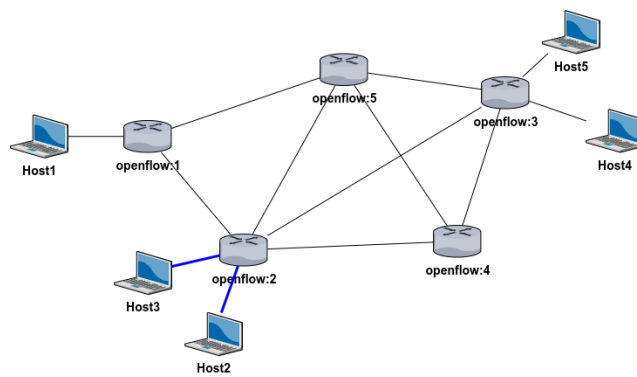
شکل ۴: مسیر از هاست ۱ به هاست ۲



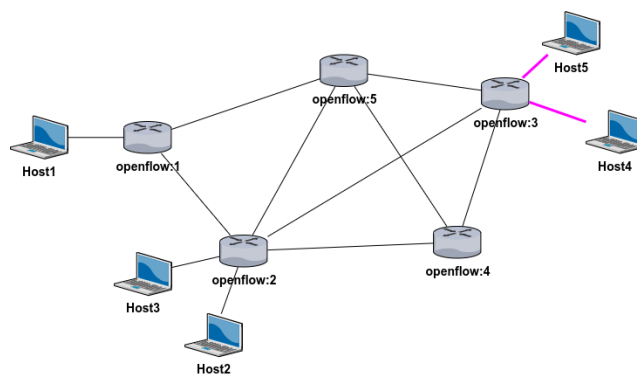
شکل ۵: مسیر از هاست ۱ به هاست ۵



شکل ۶: مسیر از هاست ۲ به هاست ۴



شکل ۷: مسیر از هاست ۲ به هاست ۳



شکل ۸: مسیر از هاست ۴ به هاست ۵

<i>Feature</i>	<i>Floodlight</i>	<i>OpenDaylight</i>
<i>Developer</i>	<i>Big Switch Networks</i>	<i>Linux Foundation</i>
<i>Supporters</i>	<i>Big Switch Networks</i>	<i>Cisco, HP, IBM, Juniper, VMWare, etc.</i>
<i>Written language</i>	<i>Java</i>	<i>Java</i>
<i>Supporting language</i>	<i>Java, Python and any language supports Rest API</i>	<i>Java</i>
<i>REST API</i>	<i>Yes</i>	<i>Yes</i>
<i>OpenStack networking (Quantum)</i>	<i>Yes</i>	<i>Yes</i>
<i>TLS supporting</i>	<i>Yes</i>	<i>Yes</i>
<i>Open-source</i>	<i>Yes</i>	<i>Yes</i>
<i>OF version</i>	<i>Full support for 1.0 and 1.3, experimental support for OpenFlow 1.1 and 1.2</i>	<i>1.0, 1.3</i>
<i>User interface</i>	<i>web, Java</i>	<i>Web</i>
<i>Interfaces</i>	<i>southbound (OpenFlow), northbound (Java, REST)</i>	<i>southbound (OpenFlow and other SB protocols), northbound (Java RPC, REST)</i>
<i>Virtualization</i>	<i>Mininet, OpenVswitch</i>	<i>Mininet, OpenVswitch</i>
<i>Platform</i>	<i>Linux, Mac, Windows</i>	<i>Linux, Windows</i>
<i>Active community</i>	<i>Yes</i>	<i>Yes</i>
<i>Age</i>	<i>4 years</i>	<i>3 years</i>
<i>Documentation</i>	<i>Good (documentations exist in official website or other developers sites)</i>	<i>Medium</i>
<i>Mailing list activity</i>	<i>Very high</i>	<i>Medium</i>
<i>Handling mixed none-OpenFlow and OpenFlow networks</i>	<i>Yes</i>	<i>Yes</i>
<i>Installation</i>	<i>Very easy</i>	<i>Easy</i>
<i>Loop supporting</i>	<i>Topologies</i>	<i>Topologies and OF islands</i>

شکل ۹: مقایسه کنترلر OpenDayLight و Floodlight (۱)

کنترلر Floodlight و OpenDaylight دو کنترلر کننده رایج در بین کنترلر کننده های SDN هستند. هر یک از این دو کنترلر کننده ماژولار هستند و می توانند برای شبکه جدید برنامه ریزی شوند خدمات. هر دو منبع باز با زبان پشتیبانی Java هستند. تفاوت بین Floodlight و OpenDaylight کنترلر Floodlight یک کاربر مبتنی بر جاوا دارد در حالی که OpenDaylight از پروتکل های کنترلر غیر OpenFlow پشتیبانی می کند و یک رابط ارائه می دهد. لایه انتزاعی، بالاتر از پروتکل های محدود به جنوب. کارهایی برای مقایسه وجود دارد کنترلر کننده ها در جنبه های معماری و کارایی مانند مقیاس پذیری و در دسترس بودن و همچنین از نظر ویژگی در این مقاله، تأخیر شبکه و از دست دادن بسته برای کنترلرهای Floodlight و OpenDaylight در بارگذاری های مختلف ترافیکی در توپولوژی های منفرد، خطی و درختی اندازه گیری می شوند. نتایج با اطمینان ۹۵٪ نشان می دهد که کنترلر کننده ها دارای یک رفتار رقابتی دارند. شرایطی وجود دارد که هر دو کنترلر یکسان عمل می کنند. آنجا موقعیت هایی است که شبکه های دارای OpenDaylight تأخیر بهتری دارند، مانند درخت توپولوژی برای شبکه ای با نیمی از ترافیک پهنای باند، در حالی که Floodlight می تواند عملکرد بهتری داشته باشد نسبت به کنترلر OpenDaylight از نظر افت بسته در شبکه بارگیری شده سنگین در توپولوژی درخت. (۱)

## References

- [۱] M. K. SHIVA ROWSHANRAD, VAJIHE ABDI, "Performance evaluation of sdn controllers: Floodlight and opendaylight," IIUM Engineering Journal, .۲۰۱۶