

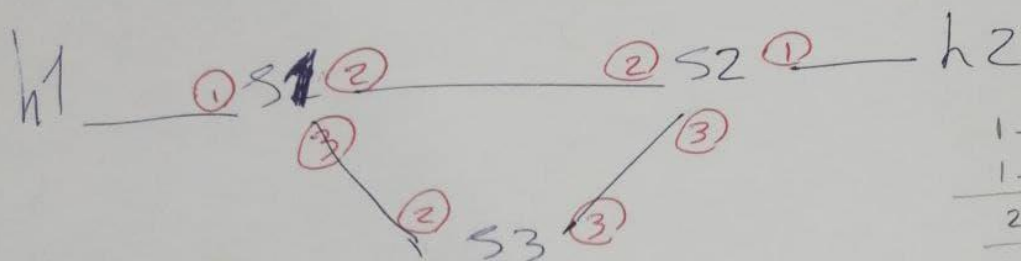


9626903 مهدی حیدری

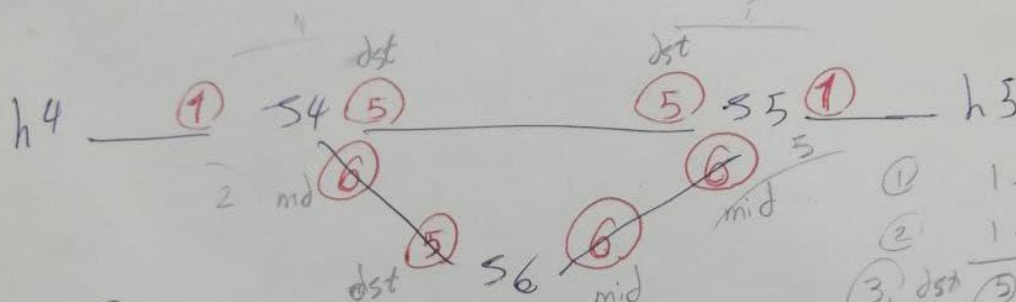
پروژه 3 شبکه 24

1.

مبدا های $h1, h4, h7$ و مقصد های $h2, h5, h8$ در شکل زیر مشخص شده اند.

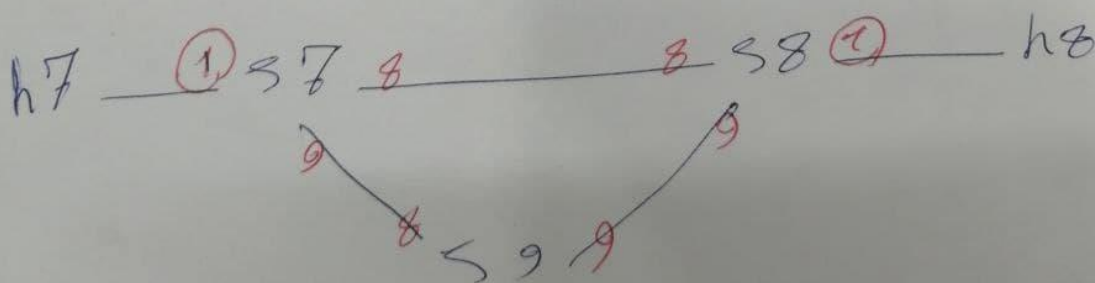


$1 \rightarrow s1 \rightarrow 2$
 $1 \rightarrow s1 \rightarrow 3$
 $2 \rightarrow s3 \rightarrow 3$
 $2 \rightarrow s2 \rightarrow 1$
 $3 \rightarrow s2 \rightarrow 1$



$1 \rightarrow s4 \rightarrow 5$ dst
 $1 \rightarrow s4 \rightarrow 6$ mid
 $5 \rightarrow s6 \rightarrow 6$ mid
 $5 \rightarrow s5 \rightarrow 1$ dst
 $6 \rightarrow s5 \rightarrow 1$ mid

$1 = 4$
 $4, 5$



فایل های ایجاد توپولوژی و فلوهای شبکه پیوست شده اند.

ابتدا با دستور زیر سرور کنترلر floodlight رو اجرا میکنیم.

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

سپس با دستور زیر توپولوژی مدنظر خود را در مینی نت میسازیم.

```
$ sudo mn --controller=remote,ip=127.0.0.1,port=6653 --switch ovsk,protocols=OpenFlow13 --custom  
topo-11sw-11host.py --topo mytopo --mac
```

سپس با دستور زیر entry های لازم رو به شبکه اضافه میکنیم.

```
$ python flow.py
```

در محیط مینی نت هم برای ping کردن با TOS مدنظر از دستور زیر استفاده میکنیم.

```
mininet> h1 ping h2 -Q <TOS, int>
```

برای دیدن flow table سوئیچ ها نیز میتوانید از دستور زیر در استفاده کنیم.

```
mininet> dpctl dump-flows -O OpenFlow13
```

ویدیو توضیحات پروژه:

<https://iutbox.iut.ac.ir/index.php/s/pJwZxsTTWmWj4z8>

1.2) اعمال اولویت بین دو تونل احتمالا بشود از طریق فیلد priority در بسته های دو تونل اعمال کرد. بسته های با اولویت بالاتر دارای priority بالاتر.

روش دیگر ممکن استفاده از QoS است که در ادامه دو لینک مناسب پیوست شده است.

1) نحوه ایجاد QoS با استفاده از floodlight

<https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/1343504/How+to+implement+Quality+Of+Service+using+Floodlight>

2) لینک گیت هاب کدها :

<https://github.com/wallnerryan/floodlight-qos-beta>

2. کنترل OpenDayLight

کنترل OpenDayLight نرم افزار JVM است و به شرط پشتیبانی از جاوا از هر سیستم عامل و سخت افزاری قابل اجرا است. کنترل کننده پیاده سازی مفهوم SDN است و از ابزار های زیر استفاده میکند.

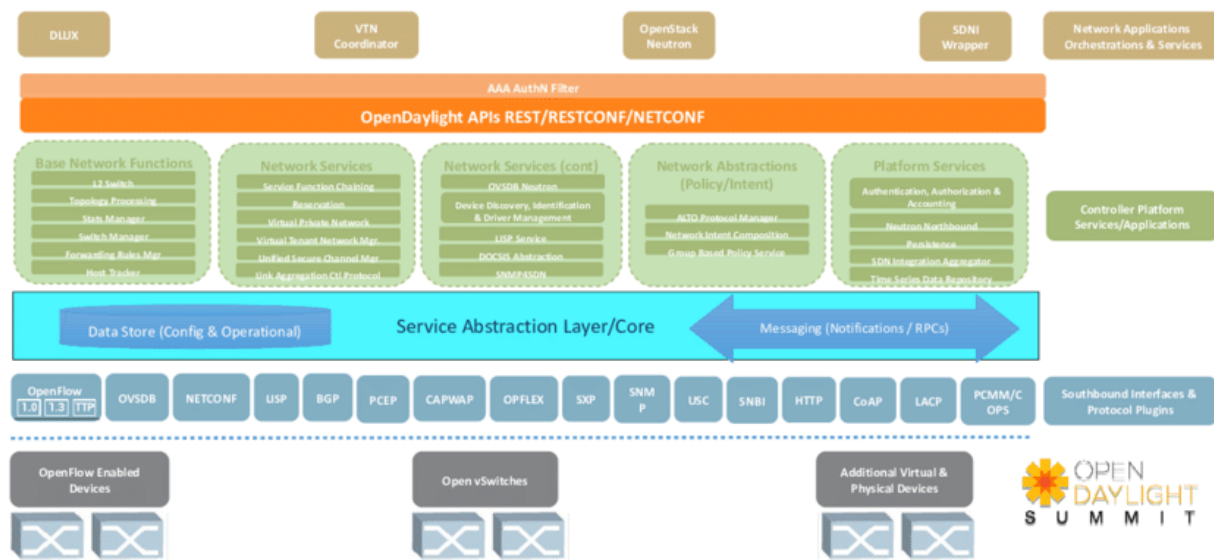
(*Maven:کنترلر ODL از Maven برای اتوماسیون ساخت آسان تر استفاده میکند. Maven از pom.xml برای نوشتن وابستگی های بین بسته نرم افزاری و همچنین توصیف اینکه بسته های نرم افزاری بارگیری و شروع میشود استفاده میکند.

(*OSGi: این چارچوب قسمت اخر ODL است زیرا امکان بارگذاری پویا بسته ها و بسته های پرونده های JAR را فراهم میکند و بسته های نرم افزاری را برای تبادل اطلاعات بهم متصل میکند.

(*رابط های JAVA: رابط های جاوا برای گوش دادن به رویدادها، مشخصات و الگوهای شکل گیری استفاده میشوند. این روش اصلی است که در آن بسته های خاص عملکردهای برگشت تماس را برای رویدادها و همچنین نسان دادن آگاهی از حالت هاص اجرا میکنند.

معماری کنترلر ODL

OpenDaylight Controller



Reference:

<https://docs.opendaylight.org/en/stable-silicon/user-guide/opendaylight-controller-overview.html>

<https://www.sdxcentral.com/networking/sdn/definitions/opendaylight-controller/>