

پروژه ی شبکه دو

مهسا امینی ۹۸۱۷۸۲۳

سوال یک:

رقم یکی به مانده آخر ۲- $n=4=2+2$

1-1

```
mahsaamini@Ubuntu: ~  
mahsaamini@Ubuntu:~$ sudo mn --topo single,4  
[sudo] password for mahsaamini:  
*** Creating network  
*** Adding controller  
*** Adding hosts:  
h1 h2 h3 h4  
*** Adding switches:  
s1  
*** Adding links:  
(h1, s1) (h2, s1) (h3, s1) (h4, s1)  
*** Configuring hosts  
h1 h2 h3 h4  
*** Starting controller  
c0  
*** Starting 1 switches  
s1 ...  
*** Starting CLI:  
mininet>
```

توپولوژی single را با استفاده از ۴ میزبان ایجاد می‌کنیم این توپولوژی شامل یک کنترلر c0 و ۴ هاست h1, h2, h3, h4 و همچنین سویچ s1 است.

اسکیرین شات سه دستور اجرا شده در قسمت بعد قرار داده شده اند.

2-1

دستور nodes

```
mininet> nodes  
available nodes are:  
c0 h1 h2 h3 h4 s1  
mininet>
```

برای نمایش همه ی نود ها از فرمان nodes استفاده می‌شود خروجی این فرمان شامل همه ی نودها است که شامل یک کنترلر c0 و ۴ هاست h1, h2, h3, h4 و همچنین یک سویچ s1 است.

دستور net

```
mininet> net  
h1 h1-eth0:s1-eth1  
h2 h2-eth0:s1-eth2  
h3 h3-eth0:s1-eth3  
h4 h4-eth0:s1-eth4  
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:h3-eth0 s1-eth4:h4-eth0  
c0  
mininet>
```

برای نمایش لینک هایی که بین دیوایس های مختلف است از فرمان net استفاده می شود که می تواند نمای کلی توپولوژی را به نمایش بگذارد خروجی این فرمان نشان می دهد که :

هاست h1 با استفاده از اینترفیس خود یعنی h1-eth0 به سویچ s1-eth1 متصل شده است.
هاست h2 با استفاده از اینترفیس خود یعنی h2-eth0 به سویچ s1-eth2 متصل شده است.
هاست h3 با استفاده از اینترفیس خود یعنی h3-eth0 به سویچ s1-eth3 متصل شده است.
هاست h4 با استفاده از اینترفیس خود یعنی h4-eth0 به سویچ s1-eth4 متصل شده است.

سویچ s1

یک اینترفیس loopback به نام lo دارد.

از طریق s1-eth1 به اینترفیس h1-eth0 متصل است.

از طریق s1-eth2 به اینترفیس h2-eth0 متصل است.

از طریق s1-eth3 به اینترفیس h3-eth0 متصل است.

از طریق s1-eth4 به اینترفیس h4-eth0 متصل است.

کنترلر c0 به عنوان مغز شبکه عمل می کند و اطلاعاتی از کل شبکه و اتصالات دارد. کنترلر به سویچ فرمان می دهد که بسته ها را به چه صورت فوروارد یا دراپ کند.

دستور dump

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=3094>
<Host h2: h2-eth0:10.0.0.2 pid=3099>
<Host h3: h3-eth0:10.0.0.3 pid=3101>
<Host h4: h4-eth0:10.0.0.4 pid=3103>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None,s1-eth4:None
pid=3108>
<Controller c0: 127.0.0.1:6653 pid=3087>
mininet>
```

برای نمایش ip address و ip هر پروسس برای هر نود از فرمان dump استفاده می کنیم.

خروجی دستور به صورت زیر است:

هاست h1 دارای اینترفیس h1-eth0 با ip=10.0.0.1 است و همچنین id پروسس آن ۳۰۹۴ است.

هاست h2 دارای اینترفیس h2-eth0 با ip=10.0.0.2 است و همچنین id پروسس آن ۳۰۹۹ است.

هاست h3 دارای اینترفیس h3-eth0 با ip=10.0.0.3 است و همچنین id پروسس آن ۳۱۰۱ است.

هاست h4 دارای اینترفیس h4-eth0 با ip=10.0.0.4 است و همچنین id پروسس آن ۳۱۰۳ است.

سویچ s1 دارای اینترفیس lo با ip=127.0.0.1 است و ip پروسس آن نیز ۳۱۰۸ است و اینترفیس های s1-eth4, s1-eth3, s1-eth2, s1-eth1 هنوز ip دریافت نکرده اند.

کنترلر c0 با ip=127.0.0.1:6653 است و همچنین ip پروسس آن ۳۰۸۷ است.

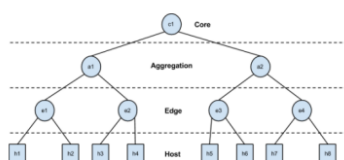
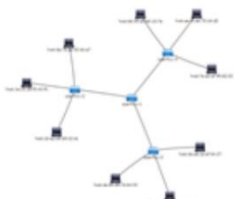
سوال دو:

1-2

یکی از این توپولوژی ها tree است که با دستور زیر میتوان آن را ساخت :

`sudo mn --topo tree, depth=2, fanout=3`

این دستور یک توپولوژی tree ایجاد میکند که شامل $\text{depth} \wedge \text{fanout}$ هاست است و به تعداد $\text{fanout} + 1$ سوئیچ دارد در این توپولوژی، شبکه به صورت درختی با عمق depth است که هاست ها در برگ ها قرار دارد و به سوئیچ هایی که گره های میانی هستند متصل میشوند. همچنین یک یا چند سوئیچ به عنوان ریشه درخت نیز وجود دارد که سوئیچ های میانی به آنها متصل اند. وابسته به عمق درخت سوئیچ های لبه نیز میتوانند بین سوئیچ های میانی و هاست ها قرار بگیرند. در یک توپولوژی tree ساده یک سوئیچ ریشه به n سوئیچ تجمعی میانی و هر یک از آنها به n سوئیچ لبه و هر یک از آنها به n هاست متصل میشود. پس به طور کلی میتوان گفت fanout تعداد فرزند هر گره و depth ارتفاع درخت را تعیین میکند. یک tree توپولوژی با $\text{fanout}=2, \text{depth}=3$ را میبینیم:



خروجی دستور به شکل زیر است:

```
mahsaamini@Ubuntu: ~  
mahsaamini@Ubuntu:~$ sudo mn --topo tree,2,3  
*** Creating network  
*** Adding controller  
*** Adding hosts:  
h1 h2 h3 h4 h5 h6 h7 h8 h9  
*** Adding switches:  
s1 s2 s3 s4  
*** Adding links:  
(s1, s2) (s1, s3) (s1, s4) (s2, h1) (s2, h2) (s2, h3) (s3, h4) (s3, h5) (s3, h6)  
(s4, h7) (s4, h8) (s4, h9)  
*** Configuring hosts  
h1 h2 h3 h4 h5 h6 h7 h8 h9  
*** Starting controller  
c0  
*** Starting 4 switches  
s1 s2 s3 s4 ...  
*** Starting CLI:  
mininet>
```

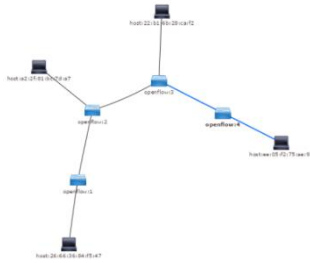
یکی دیگر از توپولوژی ها linear است.

یکی دیگر از این توپولوژی ها linear است که با دستور زیر میتوان آن را ساخت:

```
sudo mn --topo=linear, 4
```

این دستور یک توپولوژی linear ایجاد میکند که شامل ۴ سویچ متصل به هم است که به هر سویچ یک هاست متصل است. همانطور که از اسم آن پیداست به دلیل اتصال سویچ ها به هم این توپولوژی خطی است.

در تصویر زیر یک توپولوژی linear با ۴ سویچ را مشاهده میکنیم:



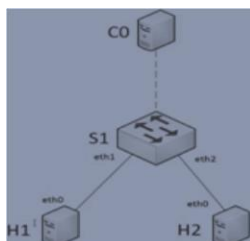
خروجی دستور به شکل زیر است:

```
mahsaamini@Ubuntu: ~  
mahsaamini@Ubuntu:~$ sudo mn --topo linear,4  
[sudo] password for mahsaamini:  
*** Creating network  
*** Adding controller  
*** Adding hosts:  
h1 h2 h3 h4  
*** Adding switches:  
s1 s2 s3 s4  
*** Adding links:  
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s2, s1) (s3, s2) (s4, s3)  
*** Configuring hosts  
h1 h2 h3 h4  
*** Starting controller  
c0  
*** Starting 4 switches  
s1 s2 s3 s4 ...  
*** Starting CLI:  
mininet>
```

توپولوژی minimal نیز وجود دارد که با دستور زیر میتوان آن را ساخت:

```
sudo mn --topo minimal
```

این دستور یک توپولوژی minimal ایجاد میکند که ساده ترین توپولوژی با یک سویچ و دو هاست است که به صورت زیر به هم متصل شده اند



خروجی دستور به شکل زیر است:

```
*** Network complete ***
mahsaamini@Ubuntu:~$ sudo mn --topo minimal
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

2-2

خروجی ها در قسمت قبل به همراه توضیح قرار داده شده است.

سوال سه:

tcpdump یک تحلیلگر بسته خط فرمان قدرتمند است که به ما امکان می دهد ترافیک شبکه را ضبط و آنالیز کنیم. هنگام کار با Mininet، شبیه ساز شبکه، tcpdump می تواند ابزار ارزشمندی برای تشخیص مشکلات مربوط به شبکه باشد.

ابتدا توپولوژی Mininet خود را با استفاده از دستور mn ایجاد و راه اندازی میکنیم. مثلاً:

```
sudo mn --topo linear,2 --controller remote
```

ترمینال های جداگانه ای را برای هاست ها و سوئیچ های مختلف در توپولوژی Mininet خود باز میکنیم. می توانیم این کار را با استفاده از Mininet CLI (فرمان xterm) یا مستقیماً از طریق ترمینال انجام دهیم

هاست ها و رابط هایی را که می خواهیم ترافیک را در آن ثبت کنیم، شناسایی میکنیم. می توانیم از دستور ifconfig برای فهرست کردن رابط های موجود در هر میزبان استفاده کنیم.

ifconfig

در ترمینال میزبانی که به آن علاقه داریم، از دستور tcpdump برای گرفتن بسته ها در یک رابط خاص استفاده میکنیم. مثلاً:

```
sudo tcpdump -i h1-eth0
```

tcpdump شروع به گرفتن بسته ها در رابط مشخص شده می کند. می توانیم ترافیک را در زمان واقعی مشاهده کنیم. اگر با مشکل شبکه مواجه شدیم، می توانیم از tcpdump برای تجزیه و تحلیل بسته ها و شناسایی مشکلات احتمالی استفاده کنیم.

می‌توانیم از فیلترهای مختلف با `tcpdump` برای محدود کردن بسته‌های ضبط شده استفاده کنیم. به عنوان مثال، برای ضبط فقط ترافیک `ICMP`، می‌توانیم از موارد زیر استفاده کنیم:

```
sudo tcpdump -i h1-eth0 icmp
```