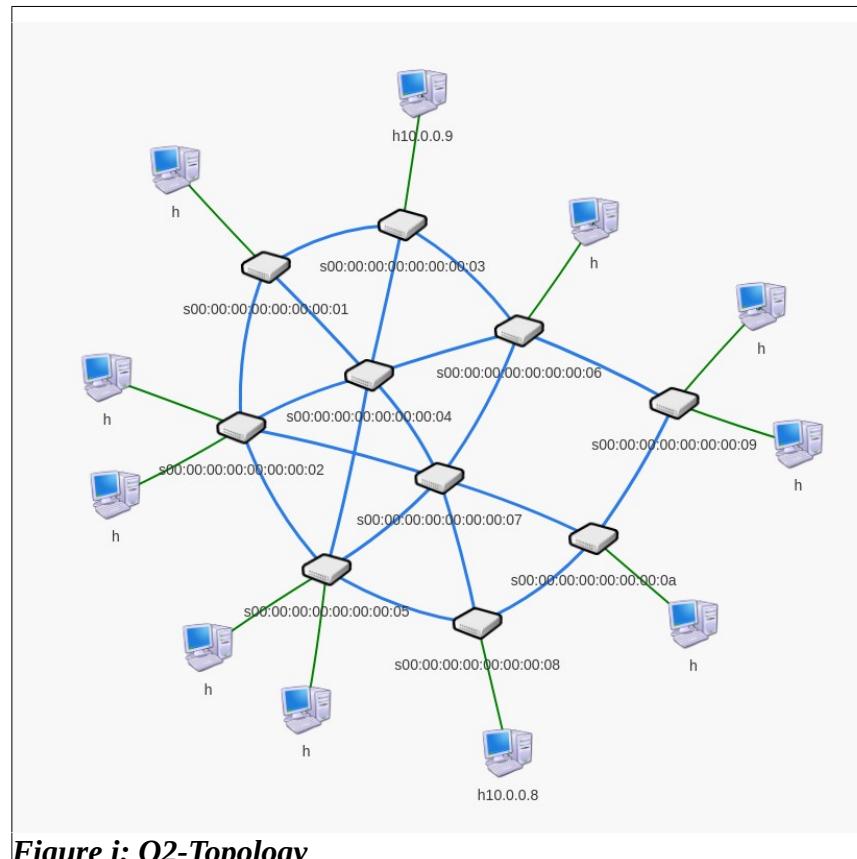


بسمه تعالیٰ
 دانشکدهٔ مهندسی برق و کامپیوٹر
 دانشگاه صنعتی اصفهان

شبکه‌های کامپیووتری ۲ – نیمسال اول ۱۴۰۰-۱۳۹۹
 گزارش پروژه شماره دو – تحویل جمعه ۱۳۹۹/۹/۲۱

مریم سعیدمهر – ش.د. : ۹۶۲۹۳۷۳

شبکه داده شده را با پایتون نوشتم (فایل `Q2.py`) که در پوشه پاسخنامه ضمیمه شده است
 توپولوژی این شبکه در فلودلایت به شکل زیر شد :



سپس با دستور زیر ، آن توپولوژی را همراه با کنترلر فلودلایت در مینی‌نت ایجاد کردم :

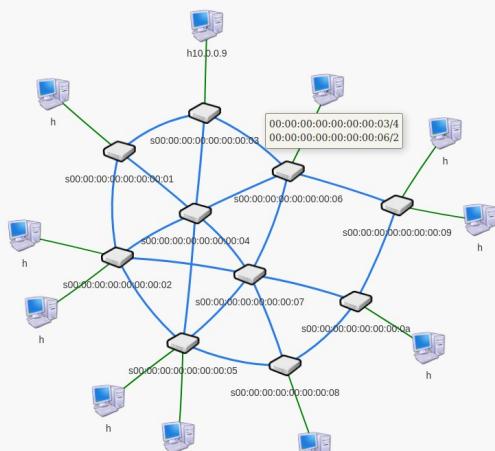
```
~$ sudo mn --custom Q2.py --topo sample --controller=remote,ip=127.0.0.1,port=6653
```

سپس در مینی‌نت (قبل از اضافه کردن هر `flowtable entity` در سویچ‌ها) یک

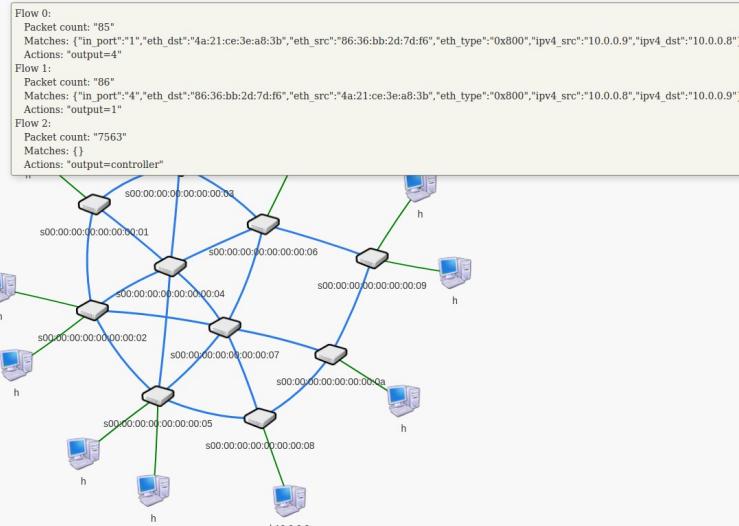
گرفتم. در این زمان که هیچ entity خودمان اضافه نکرده ایم ، خود فلودلایت ، کوتاه ترین مسیر را انتخاب میکند که در اینجا ، کوتاه ترین مسیر بین هاست ۹ و هاست ۸ : $H_9 \rightarrow S_3 \rightarrow S_6 \rightarrow S_7 \rightarrow S_8 \rightarrow H_8$ است.

در ادامه تصاویری از شماره لینکها و flowtable سویچ ها آورده ام :

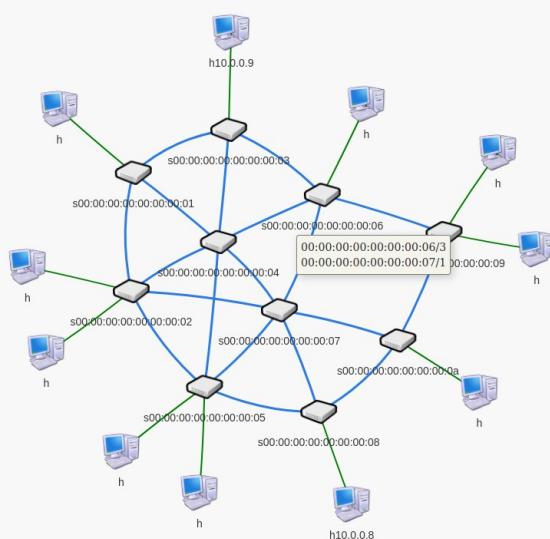
(* تصاویر به ترتیب مشخص شده‌اند).



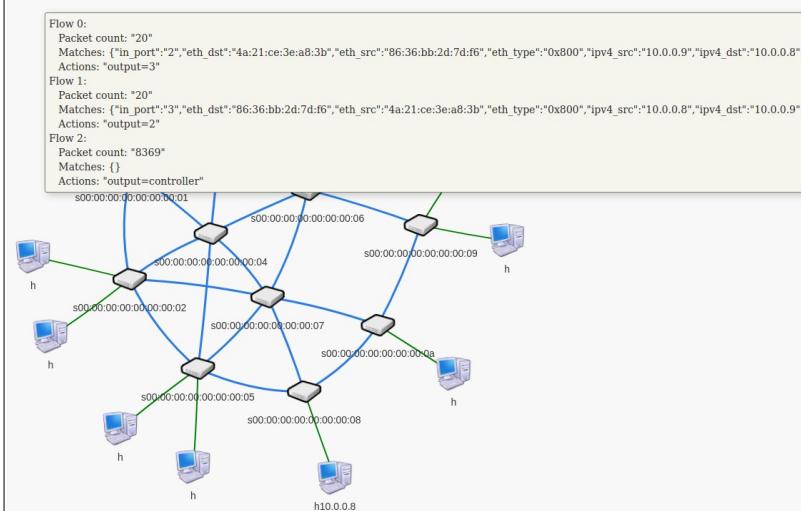
تصویر b: طبق تصویر مقابل ، پورت 4 سویچ 3 را به سویچ 6 متصل میکند.



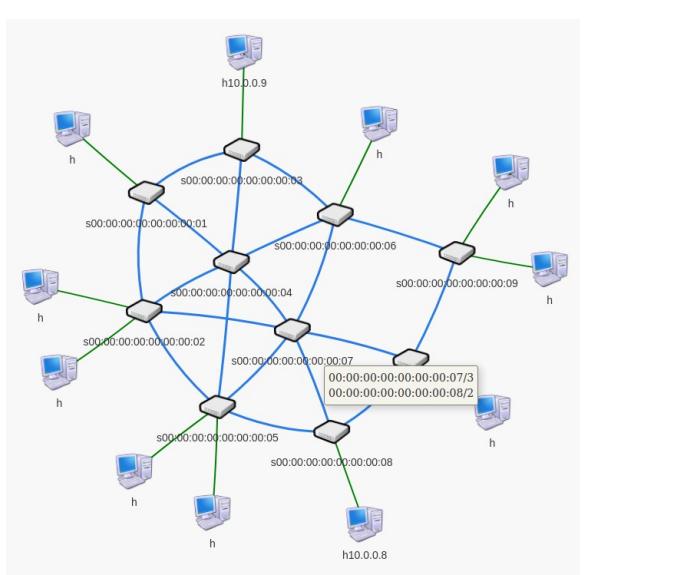
تصویر a: طبق flowtable سویچ 3 ، بسته به مبدأ $h9$ و مقصد $h8$ به پورت 4 هدایت میشود



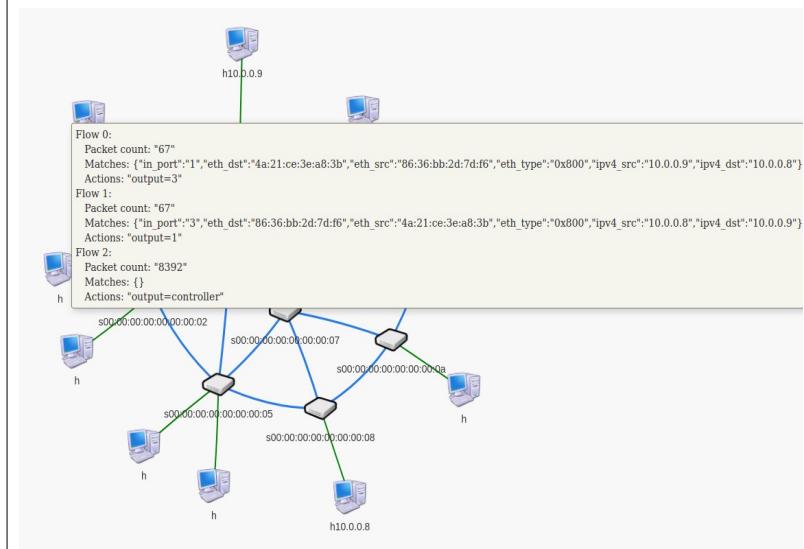
تصویر d: طبق تصویر مقابل ، پورت 3 سویچ 6 را به سویچ 7 متصل میکند



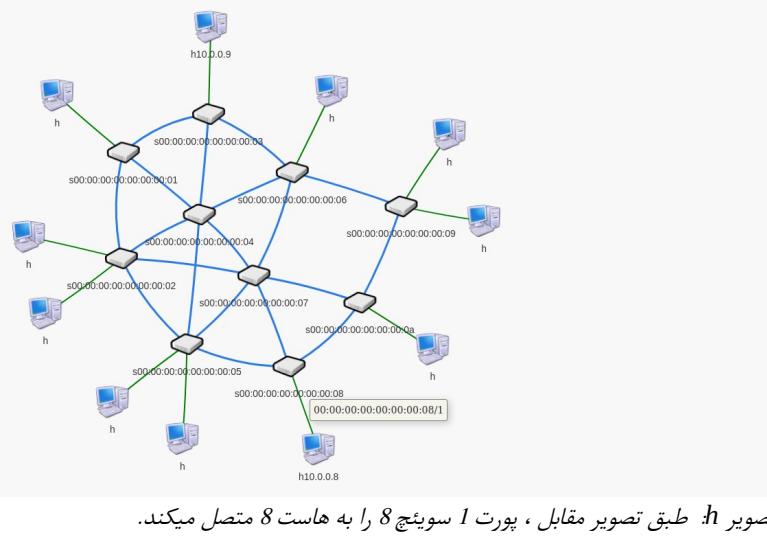
تصویر c: طبق flowtable سویچ 6 ، بسته به مبدأ $h9$ و مقصد $h8$ به پورت 3 هدایت میشود



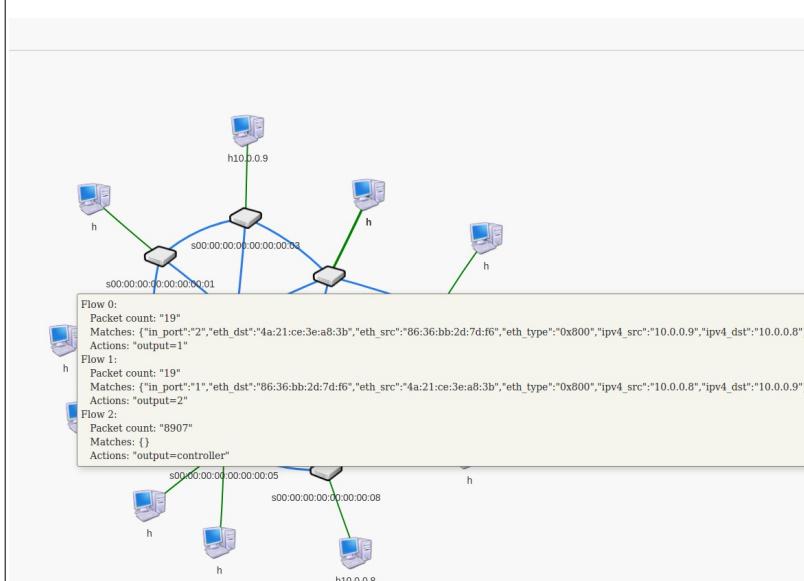
تصویر f: طبق تصویر مقابل ، پورت 3 سویچ 7 را به سویچ 8 متصل میکند



تصویر e: طبق flowtable سویچ 7 ، بسته به مبدأ h9 و مقصد h8 به پورت 3 هدایت میشود



تصویر h: طبق تصویر مقابل ، پورت 1 سویچ 8 را به هاست 8 متصل میکند.



تصویر g: طبق flowtable سویچ 8 ، بسته به مبدأ h9 و مقصد h8 به پورت 1 هدایت میشود

به این ترتیب مسیری که فلودلایت مشخص میکند (کوتاه ترین مسیر) برای بسته های به مقصد 8 به h8 به صورت H9 -> S3 -> S6 -> S7 -> S8 -> H8

همچنین packet count f1ow های مربوط به بسته های هاست 9 به هاست 8 نیز اعداد معناداری هستند که بدین معناست که در لحظه اسکرین شات گرفتن ها، ping همچنان فعال بوده است. کد این شبکه را در صفحه بعدی آورده ام.

```

from mininet.topo import Topo

class myTopo(Topo):
    def __init__(self):
        Topo.__init__(self)
        #hosts definition
        Host1 = self.addHost('h1')
        Host2 = self.addHost('h2')
        Host3 = self.addHost('h3')
        Host4 = self.addHost('h4')
        Host5 = self.addHost('h5')
        Host6 = self.addHost('h6')
        Host7 = self.addHost('h7')
        Host8 = self.addHost('h8')
        Host9 = self.addHost('h9')
        Host10 = self.addHost('h10')
        Host11 = self.addHost('h11')
        #switches definition
        Switch1 = self.addSwitch('s1')
        Switch2 = self.addSwitch('s2')
        Switch3 = self.addSwitch('s3')
        Switch4 = self.addSwitch('s4')
        Switch5 = self.addSwitch('s5')
        Switch6 = self.addSwitch('s6')
        Switch7 = self.addSwitch('s7')
        Switch8 = self.addSwitch('s8')
        Switch9 = self.addSwitch('s9')
        Switch10 = self.addSwitch('s10')
        #links definition
        self.addLink(Host1,Switch1)
        self.addLink(Host2,Switch2)
        self.addLink(Host3,Switch2)
        self.addLink(Host4,Switch9)
        self.addLink(Host5,Switch9)
        self.addLink(Host6,Switch10)
        self.addLink(Host7,Switch5)
        self.addLink(Host8,Switch8)
        self.addLink(Host9,Switch3)
        self.addLink(Host10,Switch6)
        self.addLink(Host11,Switch5)
        self.addLink(Switch1,Switch2)
        self.addLink(Switch1,Switch3)
        self.addLink(Switch1,Switch4)
        self.addLink(Switch2,Switch5)
        self.addLink(Switch3,Switch4)
        self.addLink(Switch4,Switch5)
        self.addLink(Switch3,Switch6)
        self.addLink(Switch6,Switch7)
        self.addLink(Switch4,Switch7)
        self.addLink(Switch7,Switch8)
        self.addLink(Switch6,Switch9)
        self.addLink(Switch2,Switch4)
        self.addLink(Switch7,Switch10)
        self.addLink(Switch10,Switch8)
        self.addLink(Switch8,Switch5)
        self.addLink(Switch9,Switch10)
        self.addLink(Switch2,Switch7)
        self.addLink(Switch7,Switch5)
        self.addLink(Switch4,Switch6)

```

```
topos = { 'sample':( lambda: myTopo() ) }
```

Q2.py

حالا قصد داریم ، entity هایی را به نحوی به flowtable ها اضافه کنیم که بسته ها از مبدأ 9 به H9 -> S3 -> S1 -> S4 -> S6 -> S7 -> S8 -> H8 مقصود هاست 8 از مسیر دلخواه ما یعنی عبور کند.

```

import httplib
import json

class StaticEntryPusher(object):
    def __init__(self,server):
        self.server = server

    def get(self,data):
        ret=self.rest_call({}, 'GET')
        return json.loads(ret[2])

    def Set(self,data):
        ret=self.rest_call(data, 'POST')
        return ret[0]==200

    def remove(self,objtype,data):
        ret=self.rest_call(data, 'DELETE')
        return ret[0]==200

    def rest_call(self,data,action):
        path='/wm/staticentrypusher/json'
        header={
            'Content-type':'application/json',
            'Accept':'application/json'
        }
        body=json.dumps(data)
        Conn=httplib.HTTPConnection(self.server,8080)
        Conn.request(action,path,body,header)
        response=Conn.getresponse()
        ret=(response.status,response.reason,response.read())
        print ret
        Conn.close()
        return ret

pusher=StaticEntryPusher('127.0.0.1')

# route from h9 to h8
# H9 -> S3 -> S1 -> S4 -> S6 -> S7 -> S8 -> H8
Entity1={
    "switch":"00:00:00:00:00:00:00:03",
    "name":"Entity1",
    "eth_type":"0x0800",
    "ipv4_src":"10.0.0.09",
    "ipv4_dst":"10.0.0.08",
    "priority":32768,
    "in_port":1,
    "active":true,
    "actions":output=2
}

```

```

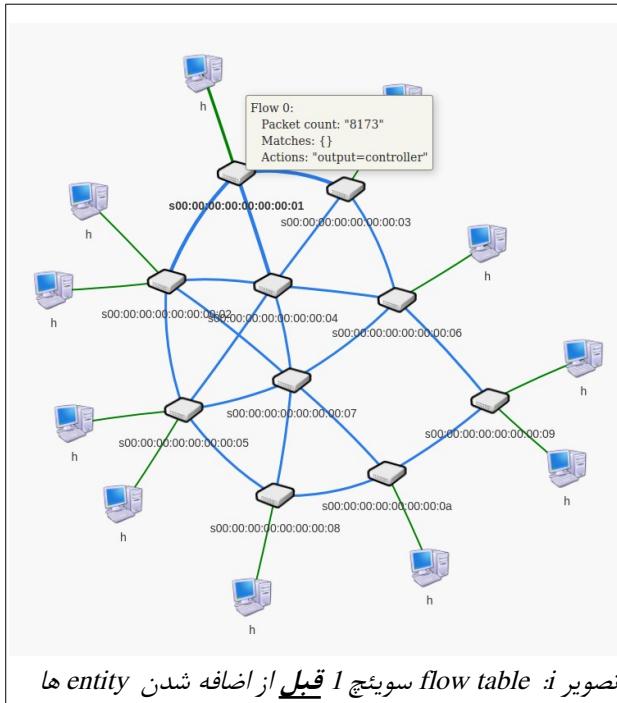
Entity2={
    "switch":"00:00:00:00:00:00:00:01",
    "name":"Entity2",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.09",
    "ipv4_dst": "10.0.0.08",
    "priority": "32768",
    "in_port": "3",
    "active": "true",
    "actions": "output=4"
}
Entity3={
    "switch": "00:00:00:00:00:00:00:04",
    "name": "Entity3",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.09",
    "ipv4_dst": "10.0.0.08",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=6"
}
Entity4={
    "switch": "00:00:00:00:00:00:00:06",
    "name": "Entity4",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.09",
    "ipv4_dst": "10.0.0.08",
    "priority": "32768",
    "in_port": "5",
    "active": "true",
    "actions": "output=3"
}
Entity5={
    "switch": "00:00:00:00:00:00:00:07",
    "name": "Entity5",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.09",
    "ipv4_dst": "10.0.0.08",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=3"
}
Entity6={
    "switch": "00:00:00:00:00:00:00:08",
    "name": "Entity6",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.09",
    "ipv4_dst": "10.0.0.08",
    "priority": "32768",
    "in_port": "2",
    "active": "true",
    "actions": "output=1"
}

pusher.Set(Entity1)
pusher.Set(Entity2)
pusher.Set(Entity3)
pusher.Set(Entity4)
pusher.Set(Entity5)
pusher.Set(Entity6)

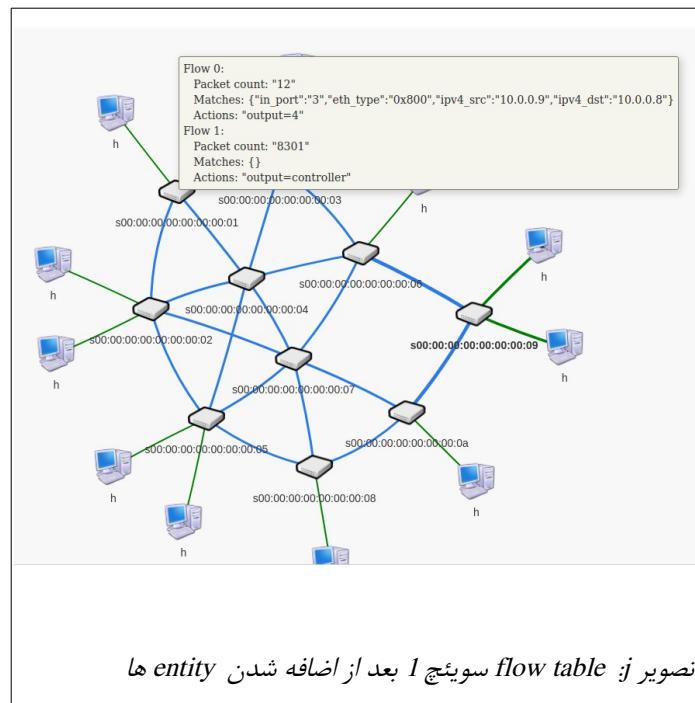
```

کد فوق در فایل `Q2_entity.py` در پوشه پاسخنامه ضمیمه شده است.

سپس با دستور `~$ python Q2_entity.py` سویچ‌ها را به `entity` flowtable اضافه کردم و گرفتم. در ادامه تصویر `Mininet> h9 ping h8` دو نمونه از سویچ‌ها را قبل از اضافه کردن `entity` و بعد از آن می‌آورم..

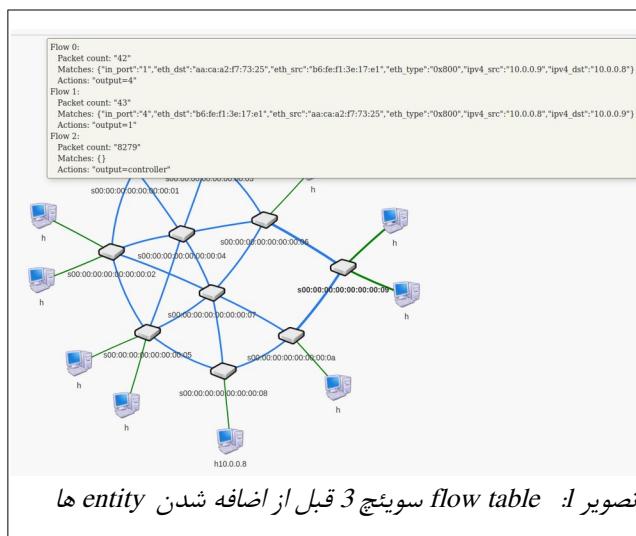


تصویر *j*: سویچ 1 قبل از اضافه شدن `entity` flow table

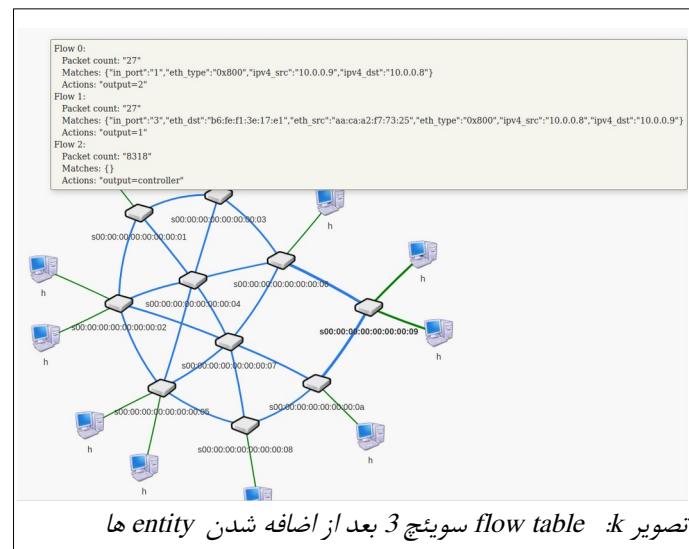


تصویر *j*: سویچ 1 بعد از اضافه شدن `entity` flow table

همان طور که در تصاویر فوق (*j*, *i*) هم پیداست، قبل از اضافه شدن `entity`‌های دلخواه‌مان، فلودلایت کوتاه ترین مسیر را انتخاب می‌کند. در مسیری که فلودلایت برای `h9` و `h8` انتخاب کرده بود یعنی `H9->S3->S6->S7->S8->H8` هیچ ارتباطی با `switch1` نداشته و لذا `entity` این سویچ نیز خالی مانده بوده. بعد از اضافه کردن `entity`‌ها به دلیل اینکه مسیر دلخواه ما از `switch1` نیز عبور می‌کند پس این سویچ بروزرسانی شده و حاوی مقدار است.



تصویر *j*: سویچ 3 قبل از اضافه شدن `entity` flow table



تصویر *j*: سویچ 3 بعد از اضافه شدن `entity` flow table

همان طور که در تصاویر فوق (L, K) هم پیداست ، قبل از اضافه شدن `entity` های دلخواهمان ، فلودلایت کوتاه ترین مسیر را انتخاب میکند. در مسیری که فلودلایت بین `h9` و `h8` انتخاب کرده بود یعنی `h9->S3->S6->S7->S8->H8` لازم بوده که `switch3` ترافیک دریافتی از مبدأ `h9` به مقصد `h8` را به پورت 4 حواله کند تا وارد `switch6` شود. بعد از اضافه کردن `entity` ها به دلیل اینکه ترافیک بین این دو هاست از مسیر دلخواه مان عبور کند `switch3` باید ترافیک دریافتی از مبدأ `h9` به مقصد `h8` را به پورت 2 حواله کند تا به `switch1` برود.

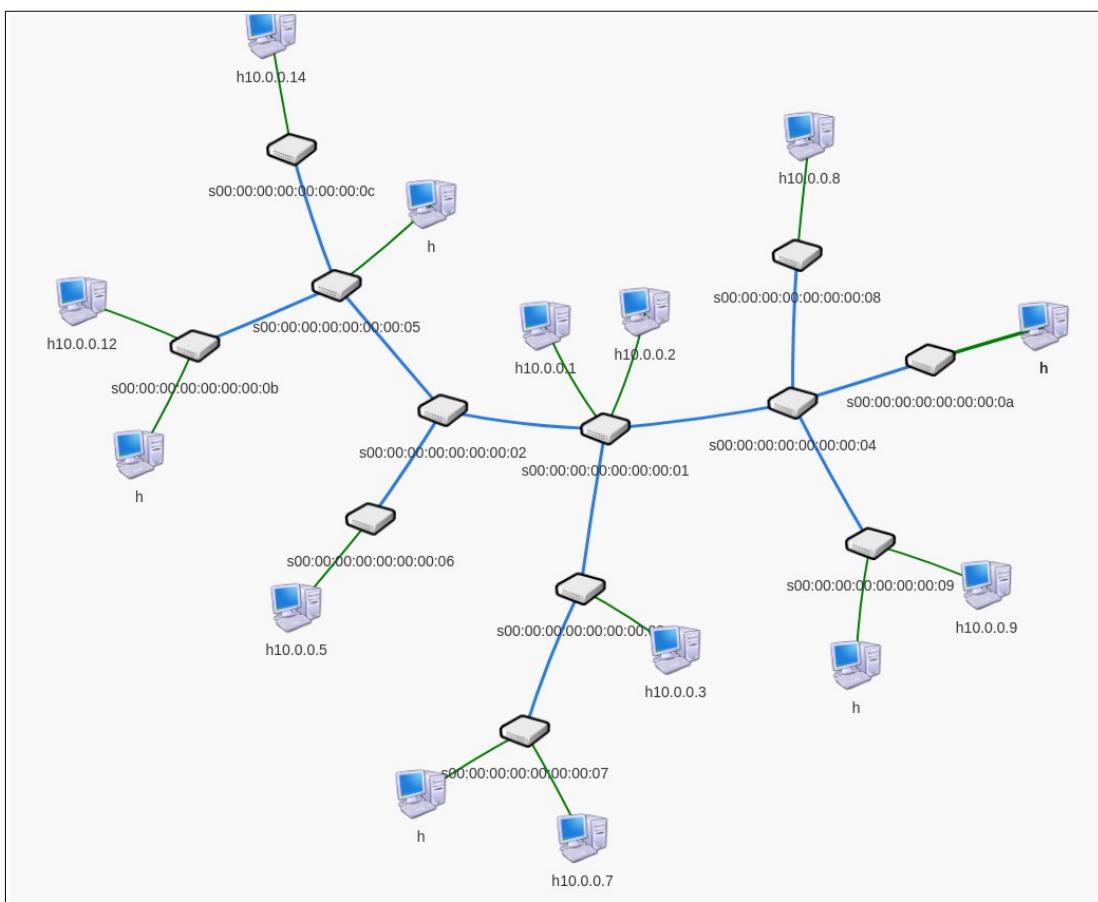
* یادآوری : مسیر دلخواه ما : `H9 -> S3 -> S1 -> S4 -> S6 -> S7 -> S8 -> H8`

بقیه توضیحات در ویدیو آپلود شده در پوشه پاسخنامه موجود است (از قبیل اینکه شماره پورت ها را از کجا پیدا کردم و ...)

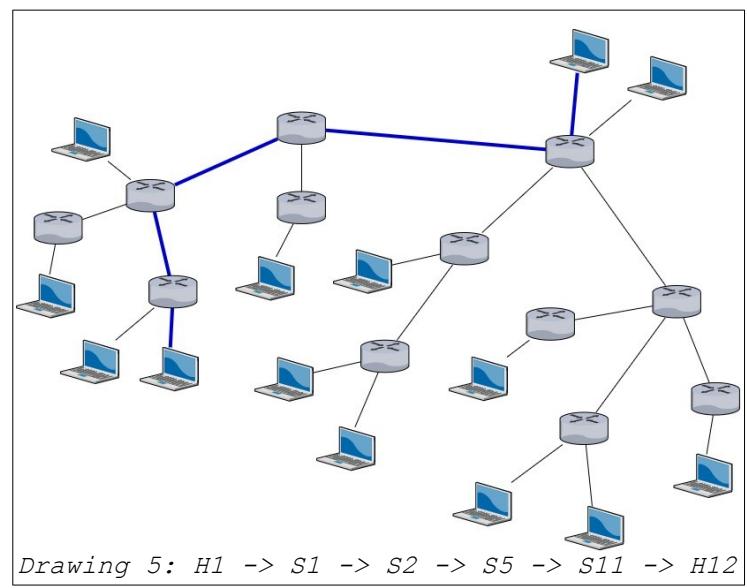
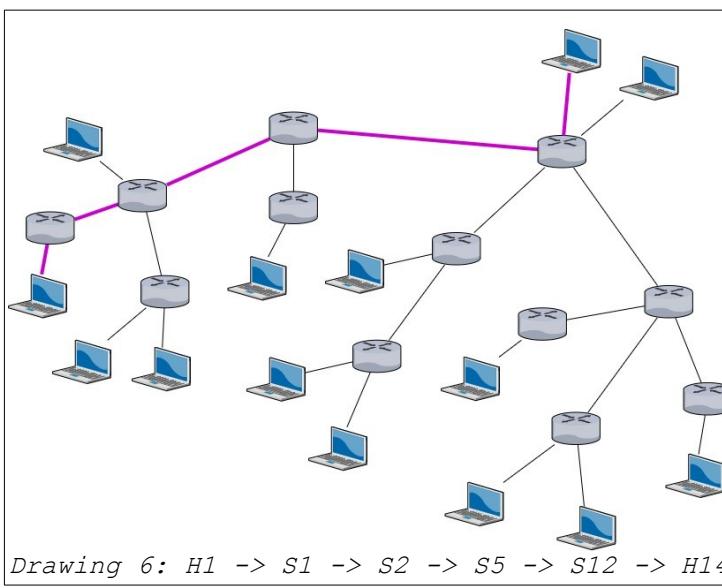
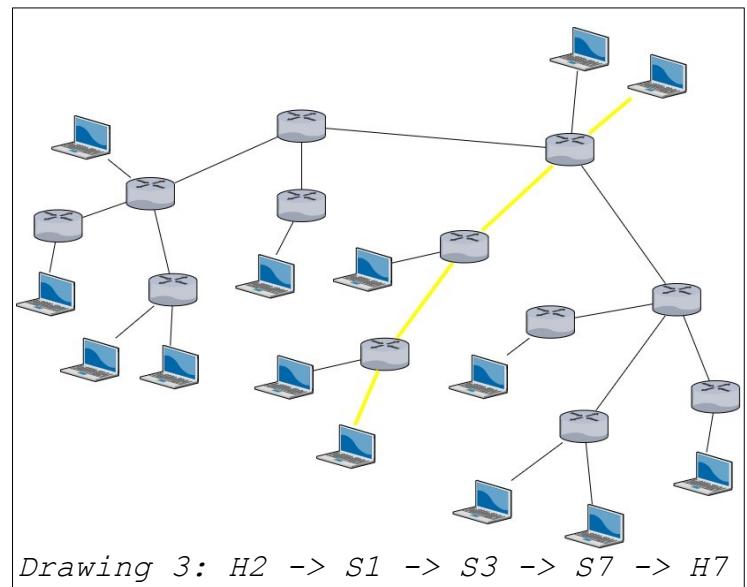
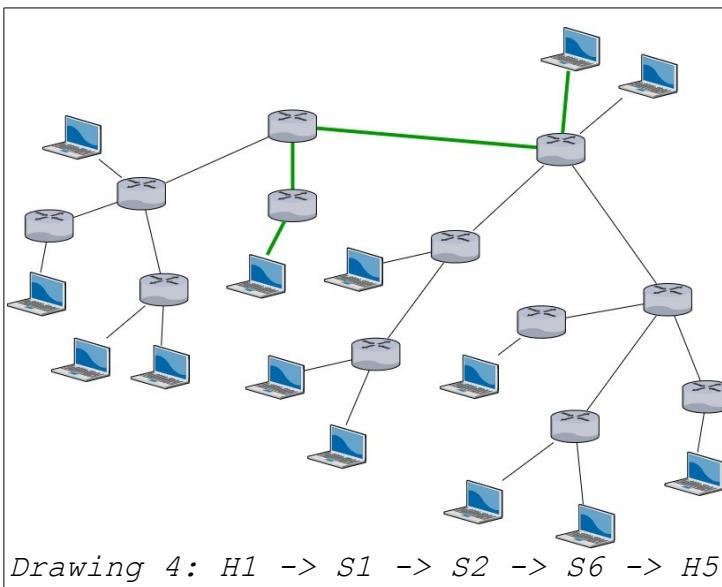
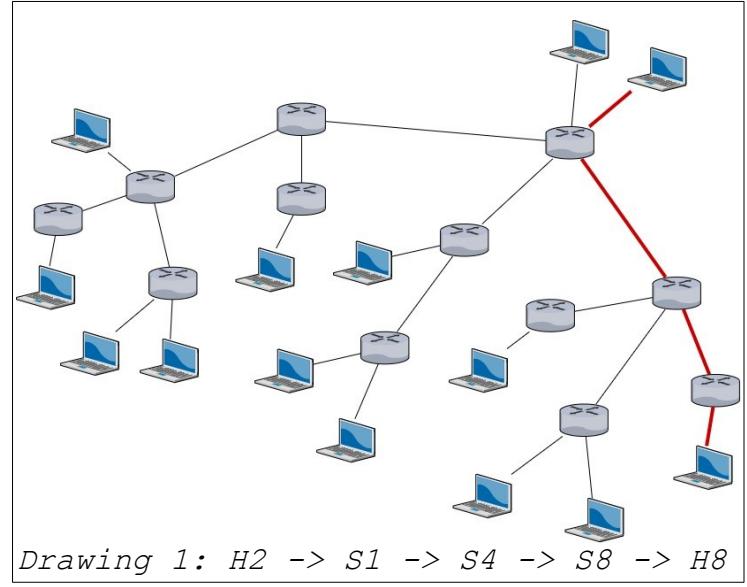
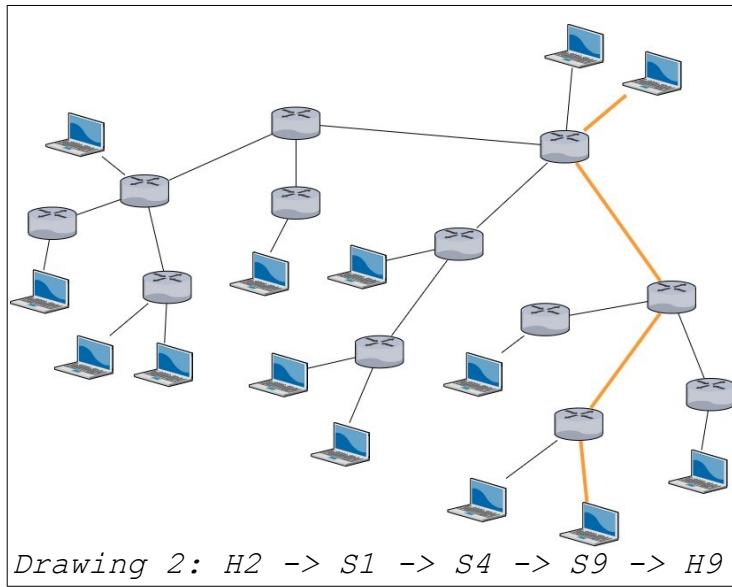
بعد از تعریف کردن `entity` ها به نحوی که بسته های ارسالی از مبدأ `h9` به مقصد `h8` از مسیر دلخواه مان عبور کند ، میتوانیم با استفاده از `entity count` در `entity` هایی که تعریف کرده ایم ، که مرتب در حال اضافه شدن است ، میتوان فهمید که جریان مورد نظر ما از مسیری که میخواستیم در حال عبور است. همچنین با بررسی `ipv4_src` و `ipv4_dst` و همچنین `Action` (پورت خروجی را مشخص کرده) سویچ های طول مسیر ، میتوان مطمئن شد که مسیر به درستی تعریف شده است.

در ویدیو کامل مسیر حرکت بسته ها را توضیح داده ام.

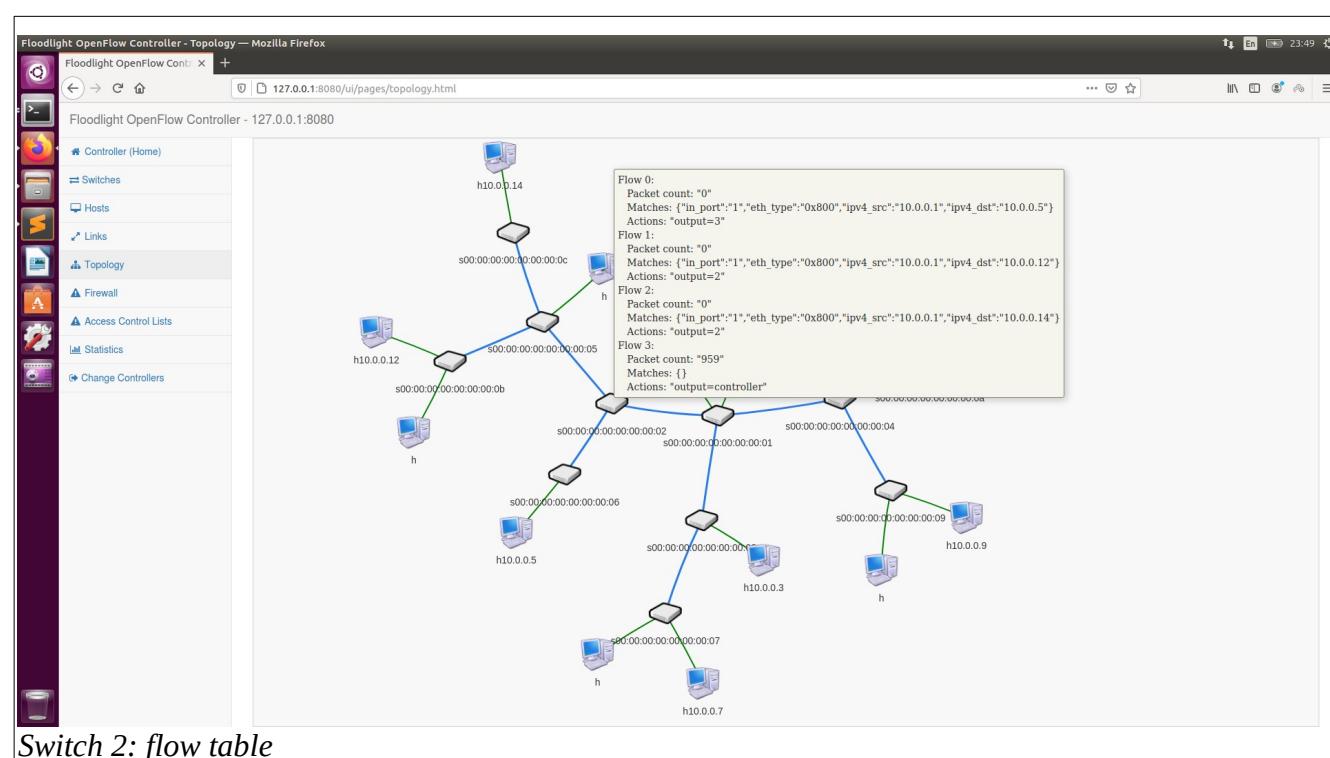
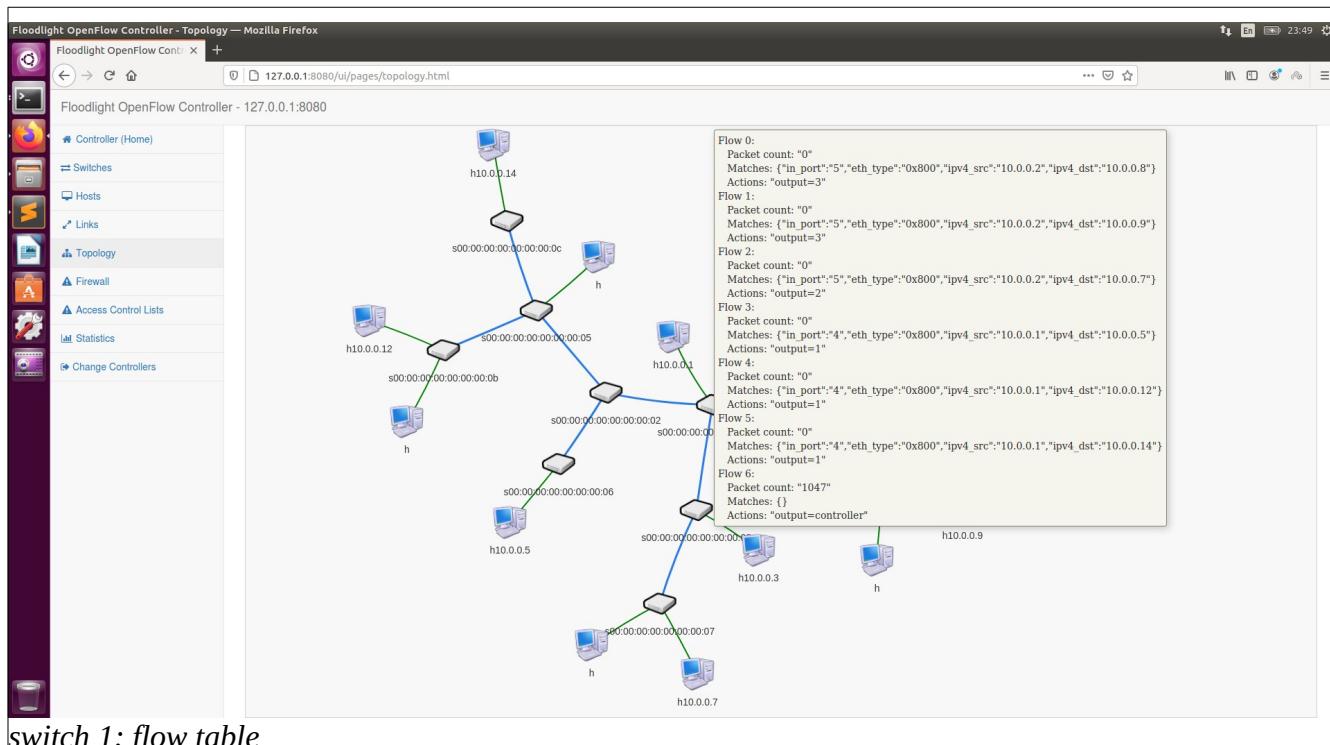
در خصوص شبکه ای که در پژوهه اول طراحی کرده ایم ، محض یادآوری ، توپولوژی شبکه من به شکل زیر بود :

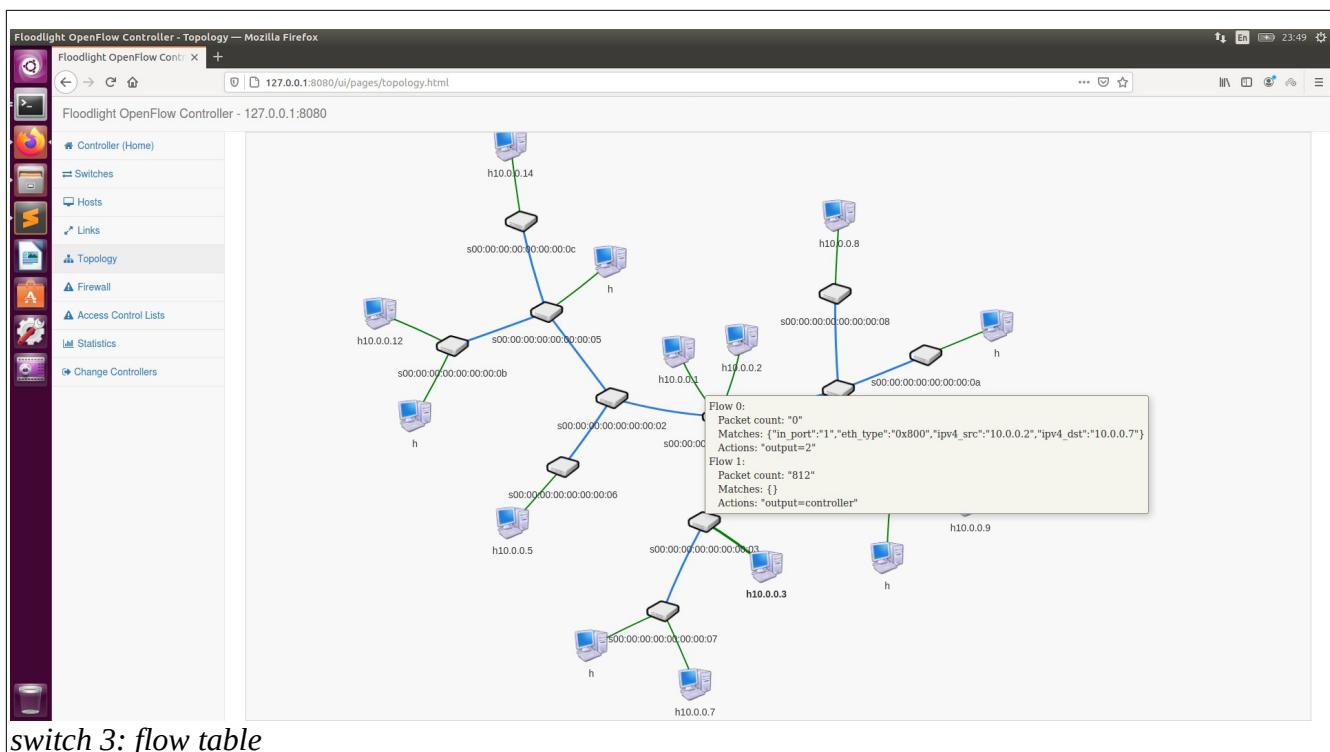


همچنین مسیرهایی که در آن پروژه مشخص کرده بودم به شکل زیر بودند :

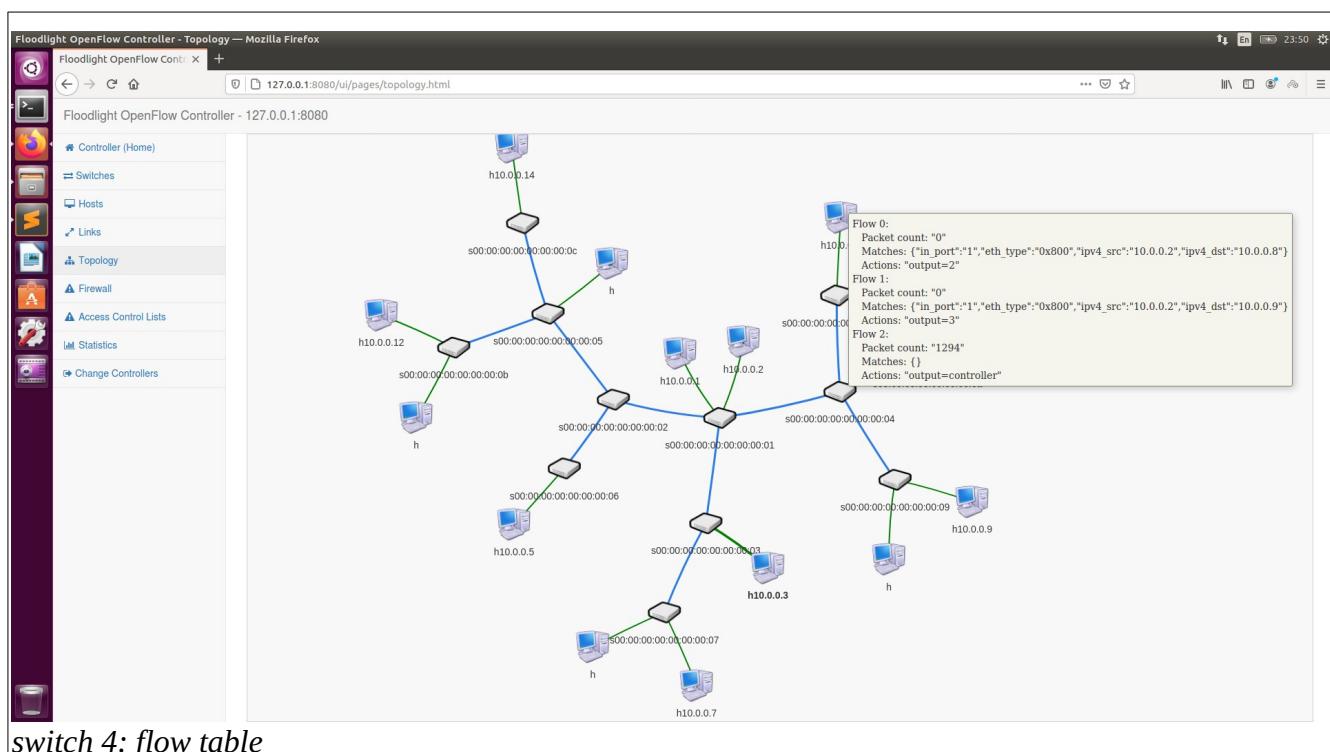


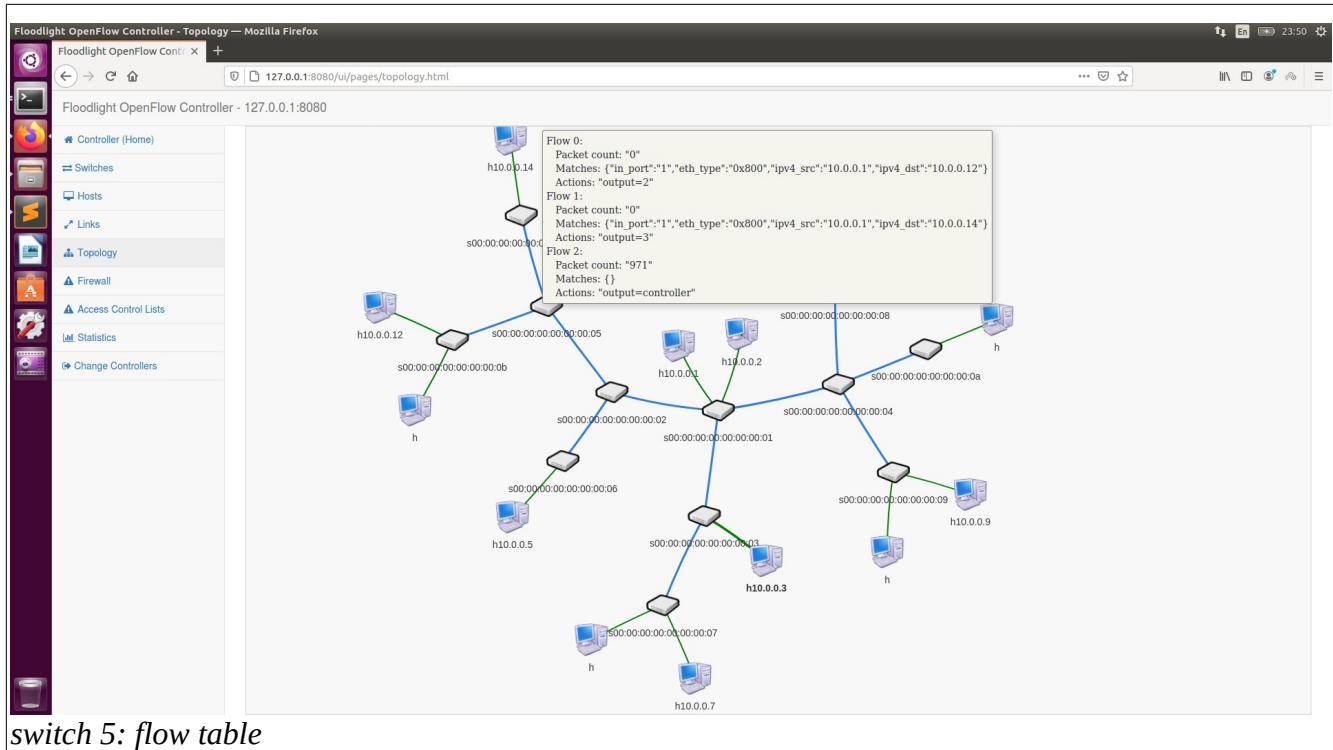
نام هاست ها و سویچ هایی که در مسیر های مشخص شده قرار گرفته اند را با استفاده از توپولوژی که رابط گرافیکی فلودلایت در اختیارم گذاشته بود به علاوه ping گرفتن بین هاست هایی که میخواستم کمک کرد تا به شکل دقیق شماره لینک ها را پیدا کنم. اما ping گرفتن ، مربوط به قبل از اضافه کردن entity ها به سویچ ها بود. برای انجام مطلوب سوال ، یک بار سورف فلودلایت را ری استارت کردم و مجددًا مراحل خواسته شده را انجام دادم. در نهایت از تمام ۱۲ سویچم اسکرین آورده ام.



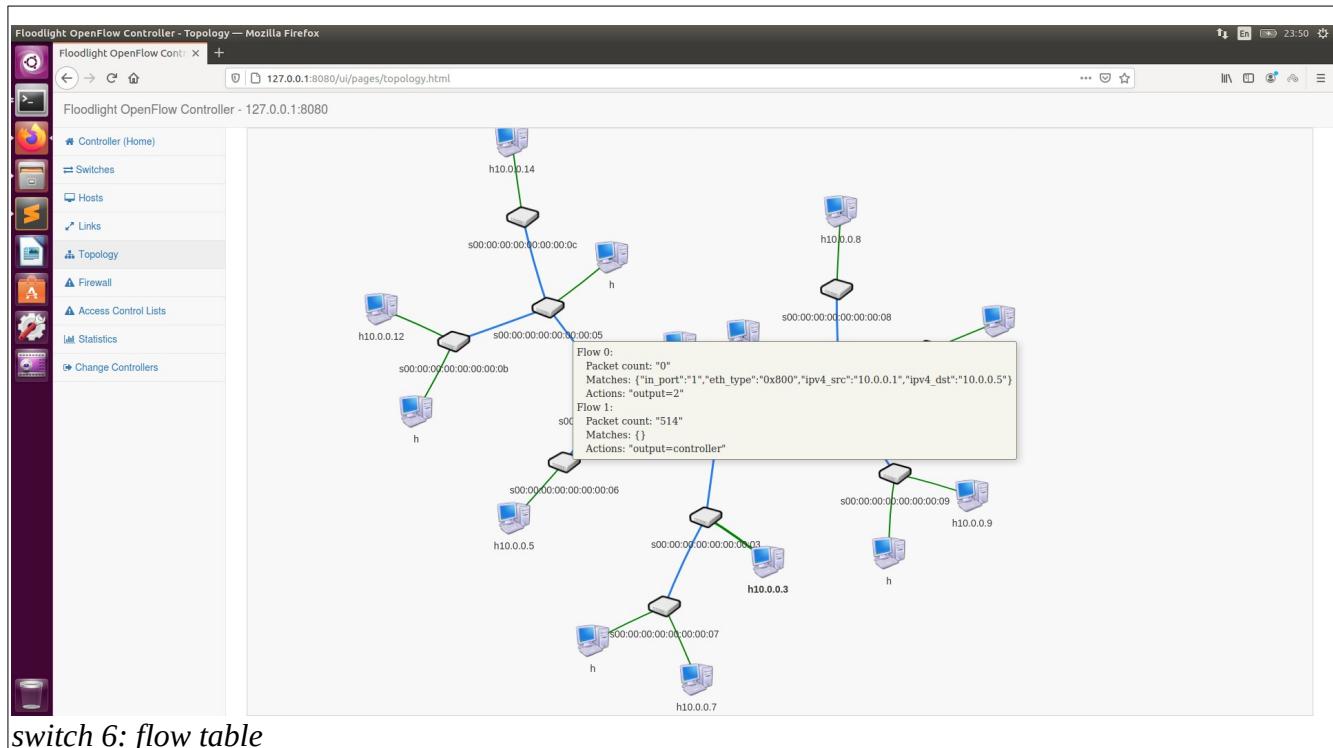


switch 3: flow table

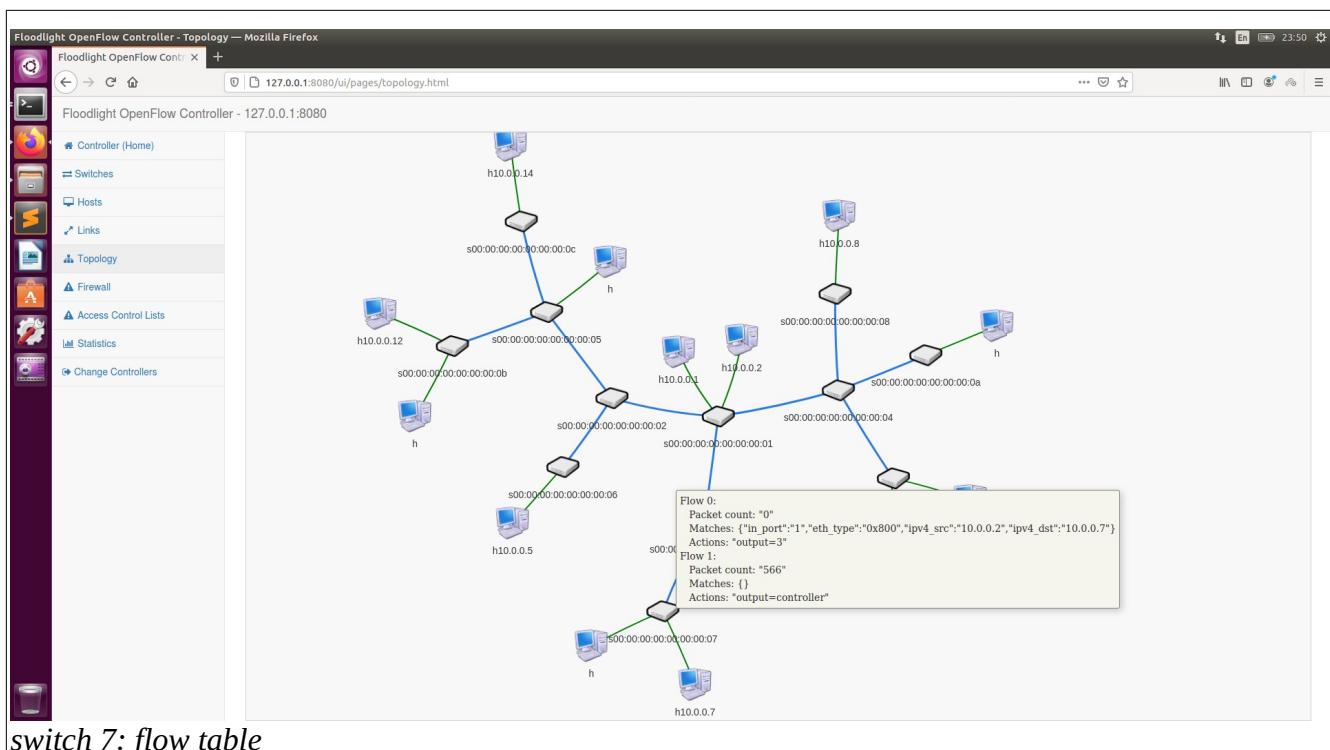




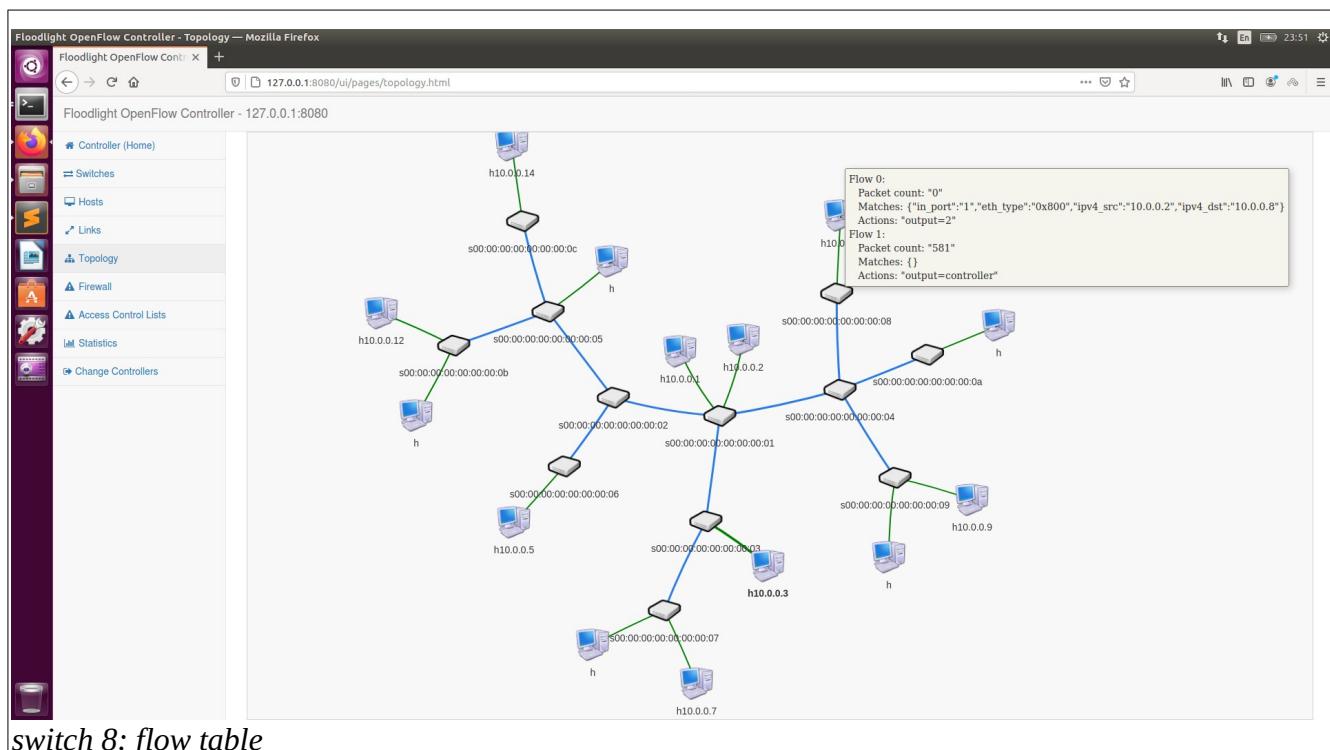
switch 5: flow table



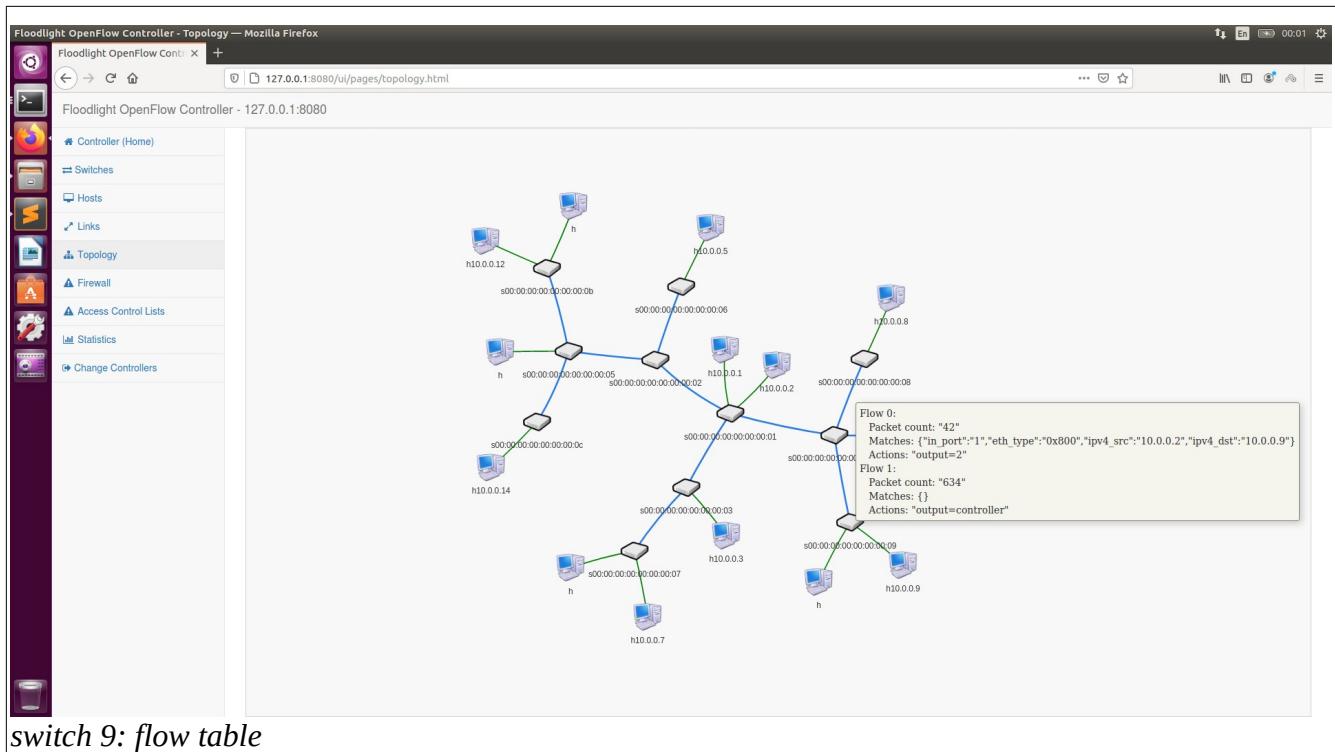
switch 6: flow table



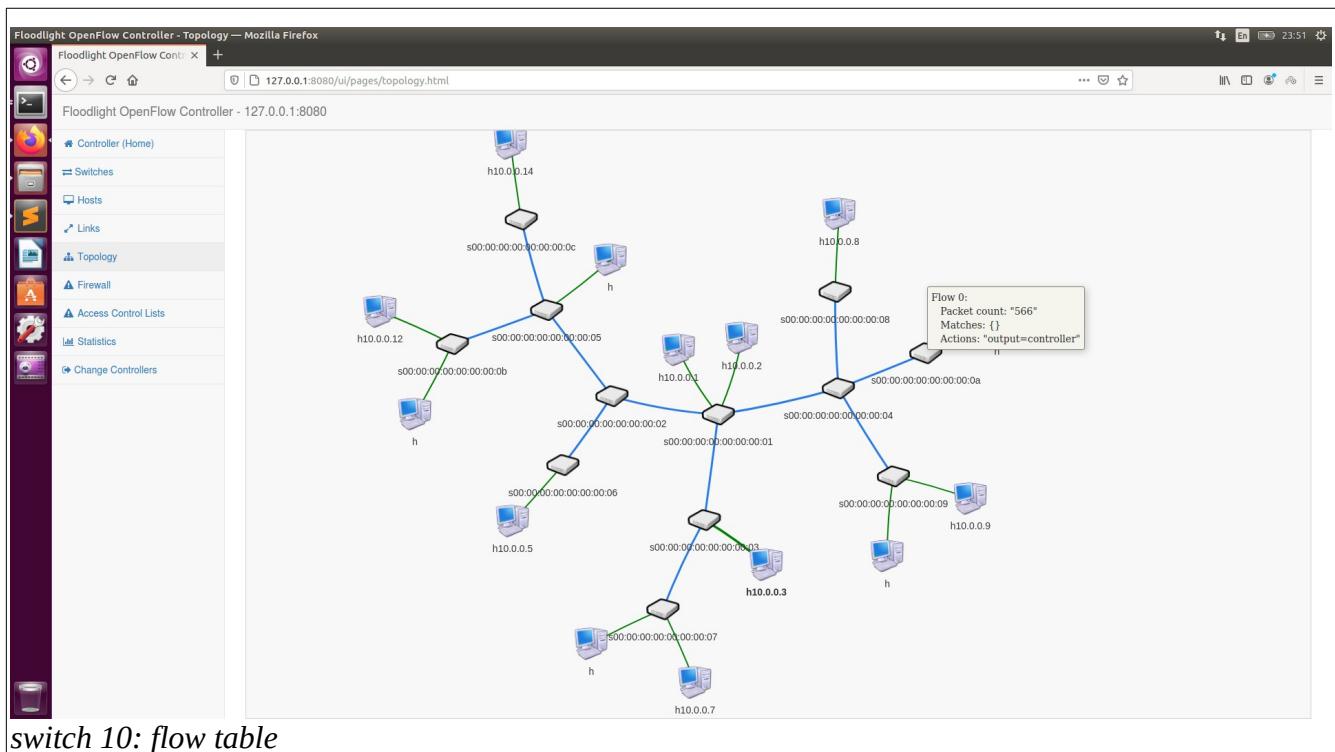
switch 7: flow table



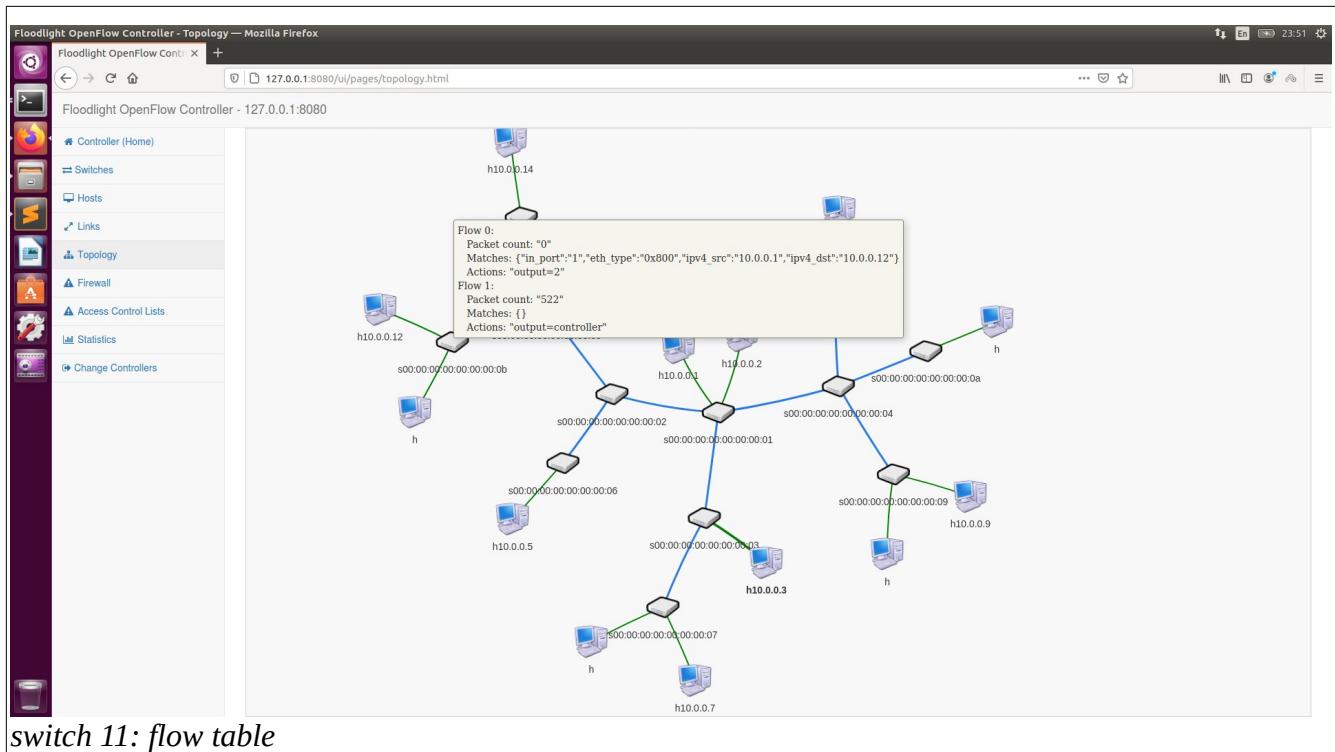
switch 8: flow table



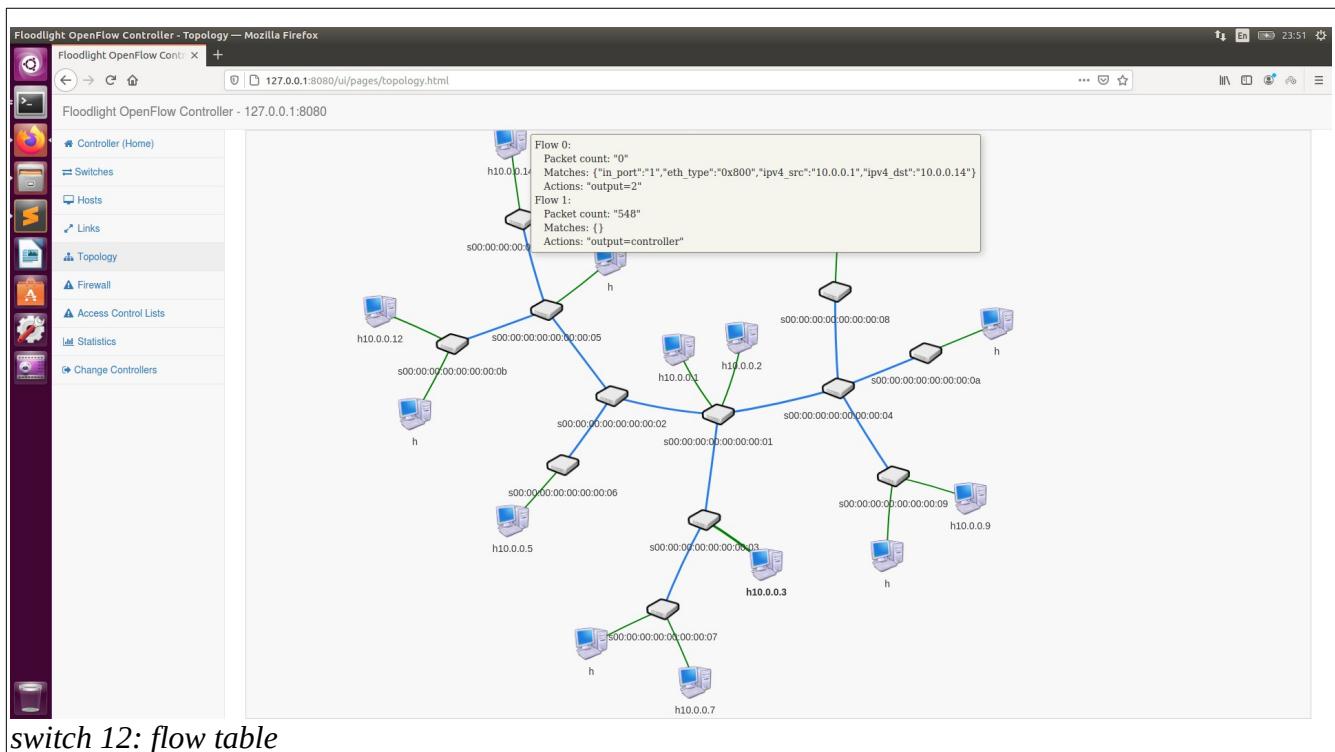
switch 9: flow table



switch 10: flow table



switch 11: flow table



switch 12: flow table

کد اضافه کردن entity ها به شکل زیر است:

```
import httplib
import json

class StaticEntryPusher(object):
    def __init__(self,server):
        self.server = server

    def get(self,data):
        ret=self.rest_call({}, 'GET')
        return json.loads(ret[2])

    def Set(self,data):
        ret=self.rest_call(data, 'POST')
        return ret[0]==200

    def remove(self,objtype,data):
        ret=self.rest_call(data, 'DELETE')
        return ret[0]==200

    def rest_call(self,data,action):
        path='/wm/staticentrypusher/json'
        header={
            'Content-type':'application/json',
            'Accept':'application/json'
        }
        body=json.dumps(data)
        Conn=httplib.HTTPConnection(self.server,8080)
        Conn.request(action,path,body,header)
        response=Conn.getresponse()
        ret=(response.status,response.reason,response.read())
        print ret
        Conn.close()
        return ret

pusher=StaticEntryPusher('127.0.0.1')

# route from h2 to h8
# H2 -> S1 -> S4 -> S8 -> H8
Entity1={
    "switch":"00:00:00:00:00:00:00:01",
    "name":"Entity1",
    "eth_type":"0x0800",
    "ipv4_src":"10.0.0.02",
    "ipv4_dst":"10.0.0.08",
    "priority":32768,
    "in_port":5,
    "active":true,
    "actions":output=3
}
```

```

Entity2={
    "switch":"00:00:00:00:00:00:00:04",
    "name":"Entity2",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.02",
    "ipv4_dst": "10.0.0.08",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=2"
}
Entity3={
    "switch": "00:00:00:00:00:00:00:08",
    "name": "Entity3",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.02",
    "ipv4_dst": "10.0.0.08",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=2"
}

# route from h2 to h9
# H2 -> S1 -> S4 -> S9 -> H9
Entity4={
    "switch": "00:00:00:00:00:00:00:01",
    "name": "Entity4",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.02",
    "ipv4_dst": "10.0.0.09",
    "priority": "32768",
    "in_port": "5",
    "active": "true",
    "actions": "output=3"
}
Entity5={
    "switch": "00:00:00:00:00:00:00:04",
    "name": "Entity5",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.02",
    "ipv4_dst": "10.0.0.09",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=3"
}
Entity6={
    "switch": "00:00:00:00:00:00:00:09",
    "name": "Entity6",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.02",
    "ipv4_dst": "10.0.0.09",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=2"
}

```

```

# route from h2 to h7
# H2 -> S1 -> S3 -> S7 -> H7
Entity7={
    "switch":"00:00:00:00:00:00:00:01",
    "name":"Entity7",
    "eth_type":"0x0800",
    "ipv4_src":"10.0.0.02",
    "ipv4_dst":"10.0.0.07",
    "priority":32768,
    "in_port":5,
    "active":true,
    "actions":"output=2"
}
Entity8={
    "switch":"00:00:00:00:00:00:00:03",
    "name":"Entity8",
    "eth_type":"0x0800",
    "ipv4_src":"10.0.0.02",
    "ipv4_dst":"10.0.0.07",
    "priority":32768,
    "in_port":1,
    "active":true,
    "actions":"output=2"
}
Entity9={
    "switch":"00:00:00:00:00:00:00:07",
    "name":"Entity9",
    "eth_type":"0x0800",
    "ipv4_src":"10.0.0.02",
    "ipv4_dst":"10.0.0.07",
    "priority":32768,
    "in_port":1,
    "active":true,
    "actions":"output=3"
}

# route from h1 to h5
# H1 -> S1 -> S2 -> S6 -> H5
Entity10={
    "switch":"00:00:00:00:00:00:00:01",
    "name":"Entity10",
    "eth_type":"0x0800",
    "ipv4_src":"10.0.0.01",
    "ipv4_dst":"10.0.0.05",
    "priority":32768,
    "in_port":4,
    "active":true,
    "actions":"output=1"
}

```

```

Entity11={
    "switch":"00:00:00:00:00:00:00:02",
    "name":"Entity11",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.01",
    "ipv4_dst": "10.0.0.05",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=3"
}
Entity12={
    "switch":"00:00:00:00:00:00:00:06",
    "name":"Entity12",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.01",
    "ipv4_dst": "10.0.0.05",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=2"
}

# route from h1 to h12
# H1 -> S1 -> S2 -> S5 -> S11 -> H12
Entity13={
    "switch":"00:00:00:00:00:00:00:01",
    "name":"Entity13",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.01",
    "ipv4_dst": "10.0.0.12",
    "priority": "32768",
    "in_port": "4",
    "active": "true",
    "actions": "output=1"
}
Entity14={
    "switch":"00:00:00:00:00:00:00:02",
    "name":"Entity14",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.01",
    "ipv4_dst": "10.0.0.12",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=2"
}
Entity15={
    "switch":"00:00:00:00:00:00:00:05",
    "name":"Entity15",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.01",
    "ipv4_dst": "10.0.0.12",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=2"
}

```

```

Entity16={
    "switch":"00:00:00:00:00:00:00:0b",
    "name":"Entity16",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.01",
    "ipv4_dst": "10.0.0.12",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=2"
}

# route from h1 to h14
# H1 -> S1 -> S2 -> S5 -> S12 -> H15
Entity17={
    "switch":"00:00:00:00:00:00:00:01",
    "name":"Entity17",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.01",
    "ipv4_dst": "10.0.0.14",
    "priority": "32768",
    "in_port": "4",
    "active": "true",
    "actions": "output=1"
}
Entity18={
    "switch":"00:00:00:00:00:00:00:02",
    "name":"Entity18",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.01",
    "ipv4_dst": "10.0.0.14",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=2"
}
Entity19={
    "switch":"00:00:00:00:00:00:00:05",
    "name":"Entity19",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.01",
    "ipv4_dst": "10.0.0.14",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=3"
}
Entity20={
    "switch":"00:00:00:00:00:00:00:0c",
    "name":"Entity20",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.01",
    "ipv4_dst": "10.0.0.14",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=2"
}

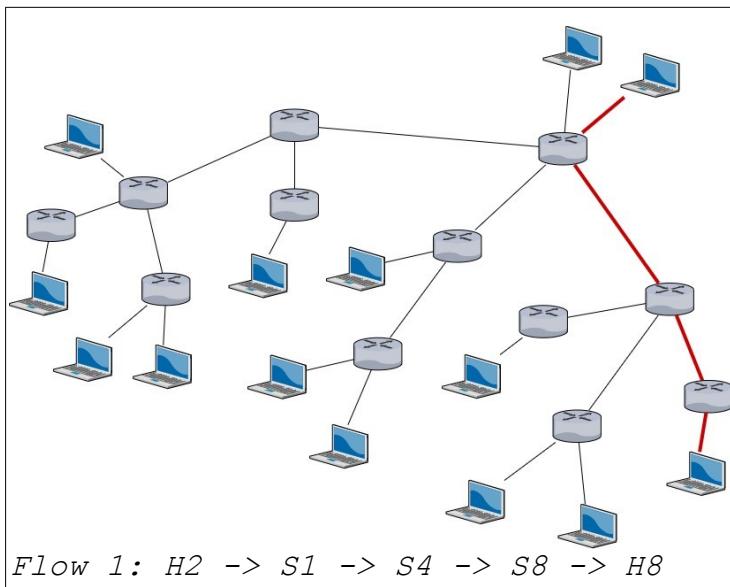
```

```

pusher.Set(Entity1)
pusher.Set(Entity2)
pusher.Set(Entity3)
pusher.Set(Entity4)
pusher.Set(Entity5)
pusher.Set(Entity6)
pusher.Set(Entity7)
pusher.Set(Entity8)
pusher.Set(Entity9)
pusher.Set(Entity10)
pusher.Set(Entity11)
pusher.Set(Entity12)
pusher.Set(Entity13)
pusher.Set(Entity14)
pusher.Set(Entity15)
pusher.Set(Entity16)
pusher.Set(Entity17)
pusher.Set(Entity18)
pusher.Set(Entity19)
pusher.Set(Entity20)

```

کد فوق در فایل `Q3_entity.py` در پوشه پاسخنامه ضمیمه شده است.



مثلاً کد جریان رو به رو به شکل زیر است :

```

# route from h8 to s1
# H2 -> S1 -> S4 -> S8 -> H8
Entity1={
    "switch":"00:00:00:00:00:00:00:01",
    "name":"Entity1",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.02",
    "ipv4_dst": "10.0.0.08",
    "priority": "32768",
    "in_port": "5",
    "active": "true",
    "actions": "output=3"
}

```

```

Entity2={
    "switch":"00:00:00:00:00:00:00:04",
    "name":"Entity2",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.02",
    "ipv4_dst": "10.0.0.08",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=2"
}
Entity3={
    "switch": "00:00:00:00:00:00:00:08",
    "name": "Entity3",
    "eth_type": "0x0800",
    "ipv4_src": "10.0.0.02",
    "ipv4_dst": "10.0.0.08",
    "priority": "32768",
    "in_port": "1",
    "active": "true",
    "actions": "output=2"
}

```

همچنین با مراجعه به تصاویر switch4: flow table و switch1: flow table و switch8: flow table که در صفحات قبل آورده ام میتوان از اضافه شدن درست این entity ها به سویچ ها مطمئن شد. به علاوه entity های فوق را با استفاده از دستور `~$ python Q3_entry.py` به شبکه ام اضافه کردم. نحوه پیدا کردن شماره لینک های هر سویچ را در ویدیو توضیح داده ام.

* نکته مهم در خصوص ویدیو :

به دلیل حجم بالای ویدیو (حدود ۱۲۹ مگابایت) تصمیم گرفتم ویدیو را در آپارات و محض احتیاط در IUTBox قرار دادم و لینک آن را در زیر آورده ام.

[آپارات : ●](https://aparat.com/v/A7nj2)

[IUT باکس : ●](https://iutbox.iut.ac.ir/index.php/s/xY6NLSTjCF7TAB3)