

第 2 小题：访问限制

一、 实验背景

为了防止外界对服务器进行 DDOS 攻击，限制一定时间内不能访问太过频繁。通过对特定网段、服务建立的访问控制体系，将绝大多数攻击阻止在到达攻击目标之前。

二、 实验目的

搭建简单网络，先使得 PC 机访问服务器成功（即看到服务器的网页），之后限制该 PC 机一定时间（比如一分钟）内再次访问服务器。限制时间过后，PC 机可以成功访问服务器。

三、 实验环境搭建

我们将在 SDN 虚拟机中通过 mininet 搭建一个如图 1.1 所示的简单网络拓扑。

由控制器控制 2 个交换机 S3、S4。

S3 与 S4 连接，S3 连接 1 个主机 h1（代表本题中的台式机），S4 连接 1 个开启 web 服务的主机 h2（代表本题中的服务器）。（如图 1.1 所示）

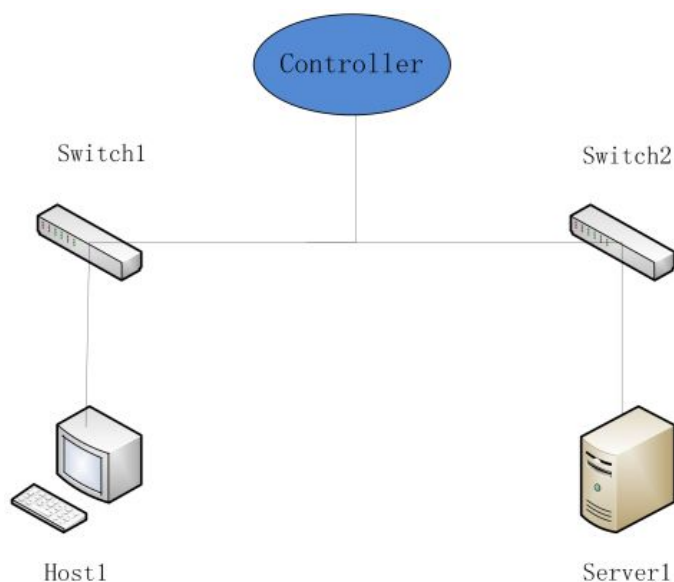


图 1.1 访问限制拓扑图

四、 实验过程及结果

（一）mininet 创建拓扑

在终端中输入以下命令创建拓扑：

```
>>sudo mn --custom /home/ubuntu/mininet/custom/topo-2sw-2host.py --topo mytopo  
--switch ovsk --controller=remote,ip=192.168.181.142,port=6633
```

命令内容已在第一小题中解释，此处不再赘述。

结果如图 1.2，图 1.3 所示，此拓扑由两台主机 h1,h2 和两台交换机 s3,s4 组成，h2 为服务器，拓扑图如图 1.1 所示。

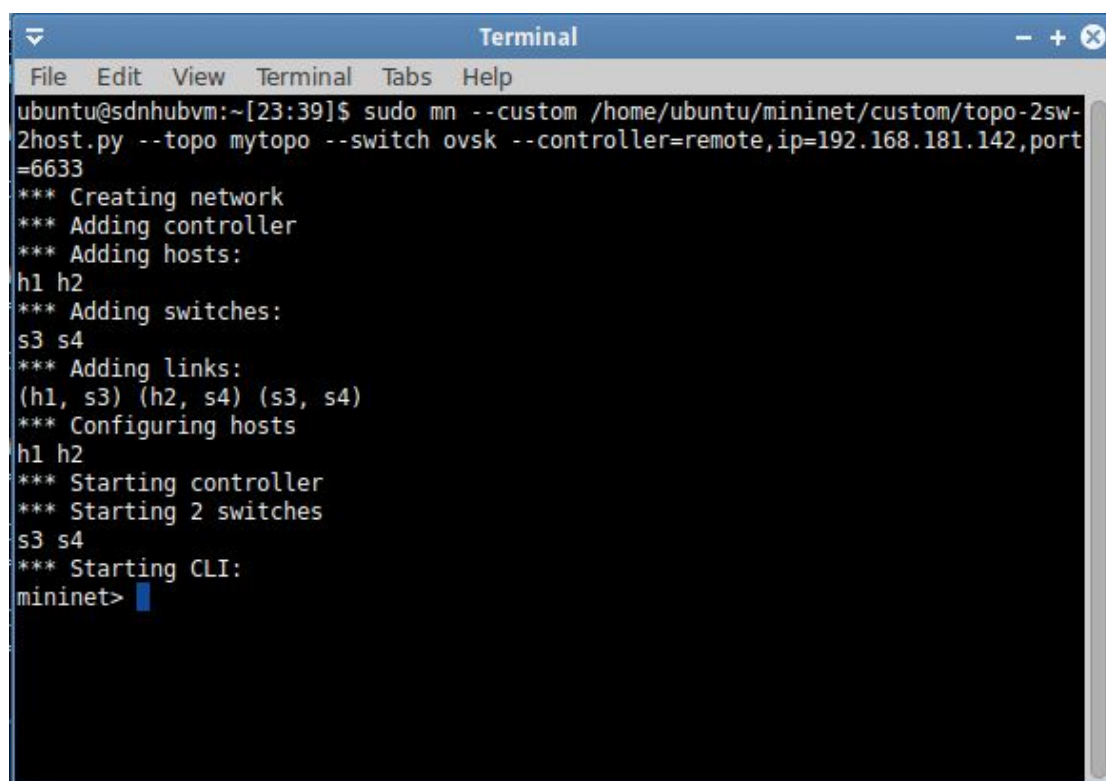
A terminal window titled "Terminal" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command prompt shows the execution of the mininet command: `ubuntu@sdnhubvm:~[23:39]$ sudo mn --custom /home/ubuntu/mininet/custom/topo-2sw-2host.py --topo mytopo --switch ovsk --controller=remote,ip=192.168.181.142,port=6633`. The output displays the network creation process: `*** Creating network`, `*** Adding controller`, `*** Adding hosts:` (listing h1 and h2), `*** Adding switches:` (listing s3 and s4), `*** Adding links:` (listing (h1, s3), (h2, s4), and (s3, s4)), `*** Configuring hosts` (listing h1 and h2), `*** Starting controller`, `*** Starting 2 switches` (listing s3 and s4), and `*** Starting CLI:`. The prompt then changes to `mininet>` with a blue cursor.

图 1.2: mininet 建立拓扑

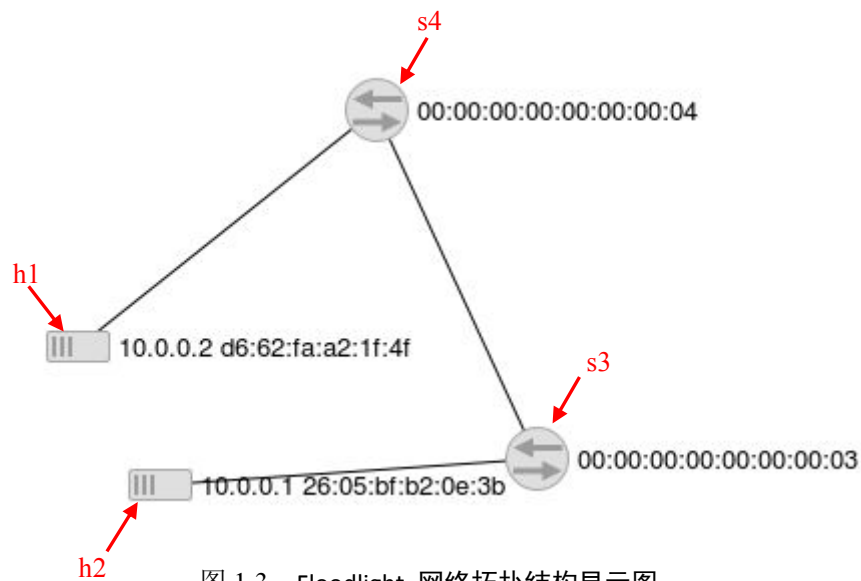


图 1.3: Floodlight 网络拓扑结构显示图

(二) 开启 h2 的 web 服务并访问

在 mininet 中输入以下命令：

```
>>h2 python -m SimpleHTTPServer 80 &    #开启 h2 的 web 服务
>>xterm h1 h2
```

并在/home/ubuntu/处加入 index.html 作为 h2 的 web 页面。
index.html 的内容如图 1.4 所示：

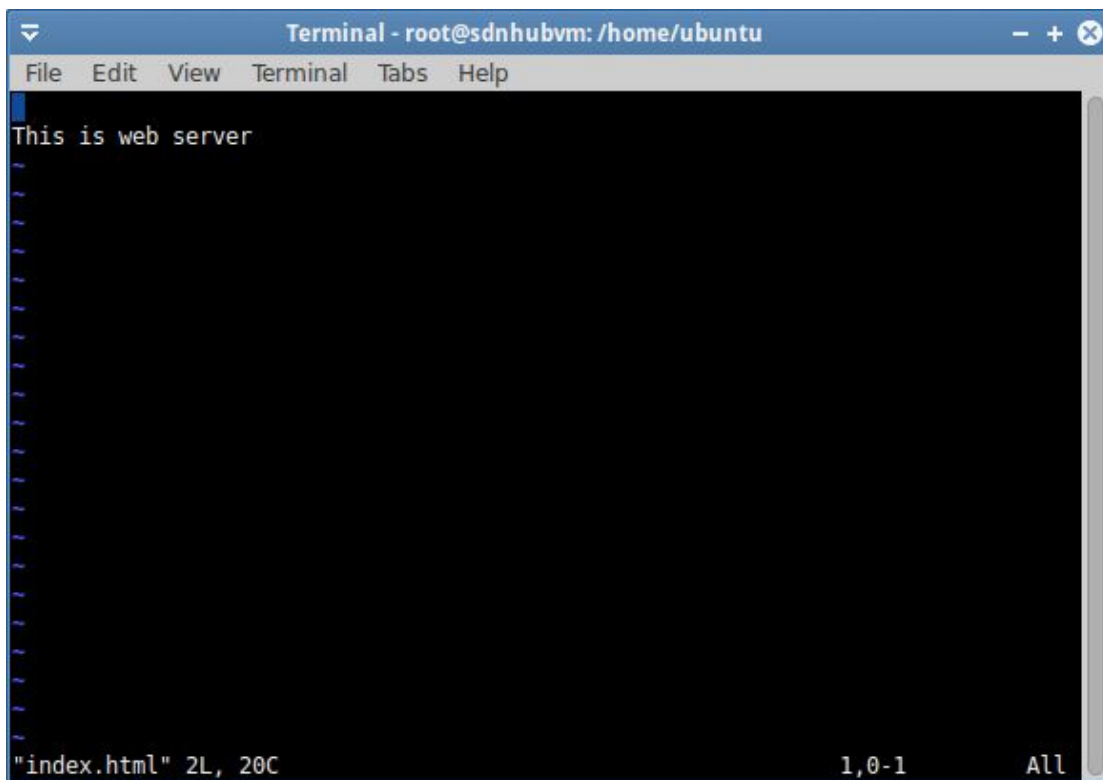
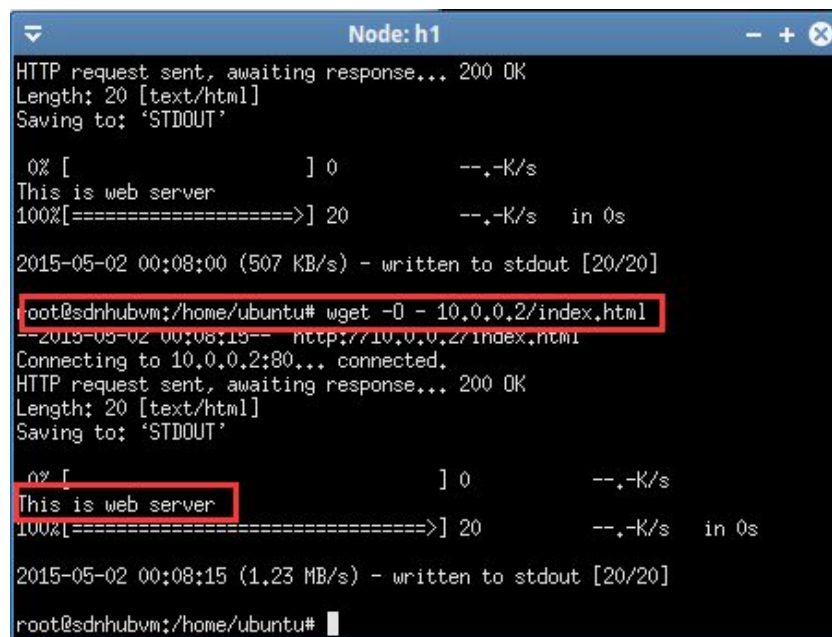


图 1.4: 服务器的 web 页面

在 Node:h1 中输入以下命令，访问服务器 h2 的 web 页面：

```
>>wget -O - 10.0.0.2/index.html
```

获得结果如图 1.5 所示。

A terminal window titled "Node: h1" with standard window controls. It shows the output of a previous wget command and then a new command: "root@sdnhubvm:/home/ubuntu# wget -O - 10.0.0.2/index.html". The command is highlighted with a red box. Below it, the terminal shows the connection process: "--2015-05-02 00:08:15-- http://10.0.0.2/index.html", "Connecting to 10.0.0.2:80... connected.", "HTTP request sent, awaiting response... 200 OK", "Length: 20 [text/html]", "Saving to: 'STDOUT'", a progress bar, and "2015-05-02 00:08:15 (1.23 MB/s) - written to stdout [20/20]". The phrase "This is web server" is also highlighted with a red box. The prompt "root@sdnhubvm:/home/ubuntu#" is visible at the bottom.

```
Node: h1
HTTP request sent, awaiting response... 200 OK
Length: 20 [text/html]
Saving to: 'STDOUT'

0% [ ] 0 --.-K/s
This is web server
100%[=====>] 20 --.-K/s in 0s

2015-05-02 00:08:00 (507 KB/s) - written to stdout [20/20]

root@sdnhubvm:/home/ubuntu# wget -O - 10.0.0.2/index.html
--2015-05-02 00:08:15-- http://10.0.0.2/index.html
Connecting to 10.0.0.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 20 [text/html]
Saving to: 'STDOUT'

0% [ ] 0 --.-K/s
This is web server
100%[=====>] 20 --.-K/s in 0s

2015-05-02 00:08:15 (1.23 MB/s) - written to stdout [20/20]

root@sdnhubvm:/home/ubuntu#
```

图 1.5:PC 机访问服务器成功

由图 1.5 可知主机 h1 访问服务器成功。

（三）添加流表限制主机 h1 在 60s 内无法访问服务器

在终端输入以下命令添加流表：

```
>>ovs-ofctl add-flow s1
```

```
hard_timeout=60,priority=1,in_port=2,actions=drop
```

```
>>ovs-ofctl dump-flows s3
```

这条流表可以使 h1 在 60s 内无法对 h2（即服务器）进行访问。60s 后流表失效，h1 又可以对 h2 进行访问。结果如图 1.6，1.7 所示。

```
Terminal - root@sdnhubvm: /home/ubuntu
File Edit View Terminal Tabs Help
ubuntu@sdnhubvm:~[01:59]$ sudo su
root@sdnhubvm:/home/ubuntu# ovs-ofctl add-flow s3 hard timeout=60,priority=1,in
port=2,actions=drop
root@sdnhubvm:/home/ubuntu# ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=11.679s, table=0, n_packets=0, n_bytes=0, hard_timeout=60,
 idle_age=11, priority=1,in port=2 actions=drop
root@sdnhubvm:/home/ubuntu# ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=46.008s, table=0, n_packets=10, n_bytes=476, hard_timeout=
 60, idle_age=6, priority=1,in port=2 actions=drop
root@sdnhubvm:/home/ubuntu# ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=58.096s, table=0, n_packets=10, n_bytes=476, hard_timeout=
 60, idle_age=18, priority=1,in port=2 actions=drop
root@sdnhubvm:/home/ubuntu# ovs-ofctl dump-flows s3
NXST_FLOW reply (xid=0x4): 60s后流表失效
root@sdnhubvm:/home/ubuntu#
```

图 1.6:设置流表使 h1 在 60s 内无法访问 h2

```
Terminal
File Edit View Terminal Tabs Help
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
mininet>
mininet> h2 python -m SimpleHTTPServer 80 &
mininet> xterm h1
mininet> pingall
*** Ping: testing ping reachability
h1 -> X
h2 -> X
*** Results: 100% dropped (0/2 received)
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
```

图 1.7:h1 在 60s 内不能访问，限制时间过后恢复访问

如图 1.6 所示，我们可以通过 `ovs-ofctl dump-flows s3` 命令观察 s3 的流表信息。从结果图 1.7 可知实现题目要求。