Lab7 SDN 实验

一、 实验目的

熟悉 SDN 的基本操作；修改并增加程序的功能。

提示：在找到 LearningSwitchTutorial.java 后，同一文件夹里的其他四个 java 文件是本次实验的 java code 的答案，大家如果实在不会就 copy 一下吧。还有第四个实验的答案少了一句将 buffer id 定义为 none，大家可以在“Create Learning Switch”中找到这句话，加在 forwardAsLearningSwitch 函数的开始处。

注意：每次进行完一个实验，都要关掉 wireshark 和 beacon（eclipse），并且在 terminal 中输入 exit。

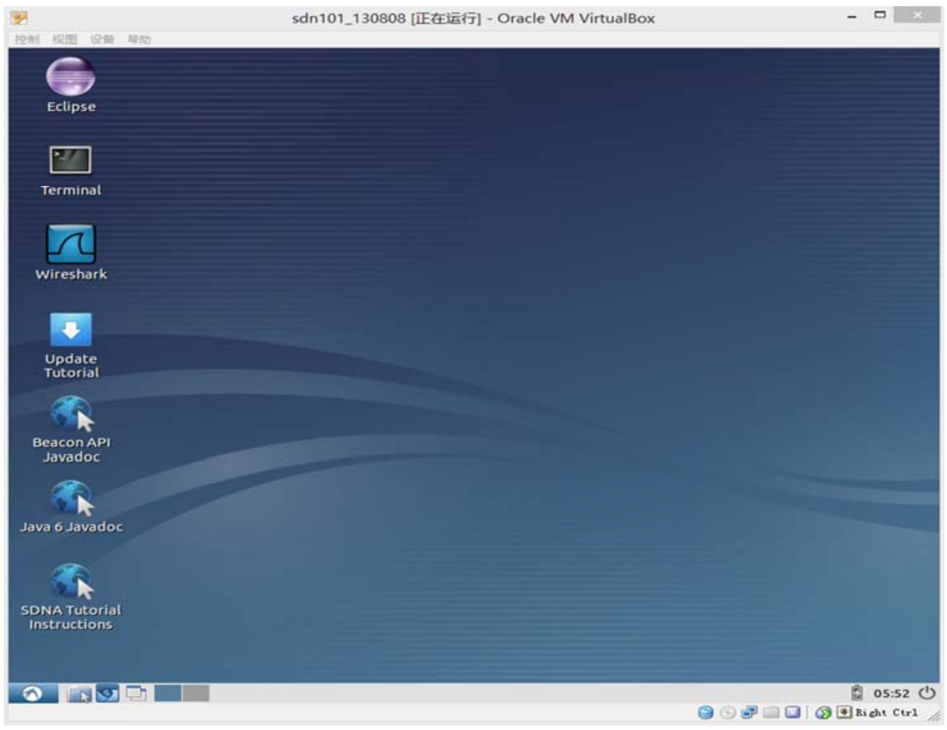
二、实验内容

1. 安装虚拟机

根据电脑操作系统的类型来选择安装虚拟机，包括VirtualBox for Linux，VirtualBox for OS X以及VirtualBox for Windows。

2. 将虚拟电脑导入虚拟机

将sdn101_130808.ova文件导入虚拟机。你将看到下列图像：



3．学习 SDN 实验的基础操作

打开本地文件SDNA Tutorial Instructions，在“Learn Development Tools”中学习SDN实验的基础操作。在java程序了，需要找到LearningSwitchTutorial.java，然后Commenting out the call to forwardAsHub, and uncommenting the call to forwardAsLearningSwitch;在每个实验中都要Change the code in forwardAsLearningSwitch method.

4．进行实验 1

在“Create Learning Switch”中找到phase1，开始实验。在java代码中需要做的改动有：

Build an OFMatch object;

Learn the source Ethernet address;

Look up and send to the destination Ethernet address.

测试步骤是：

Start Eclipse and run the Tutorial Controller, start Wireshark;

Start Mininet (only if not already started) and wait for Beacon's console to report the switch has connected;

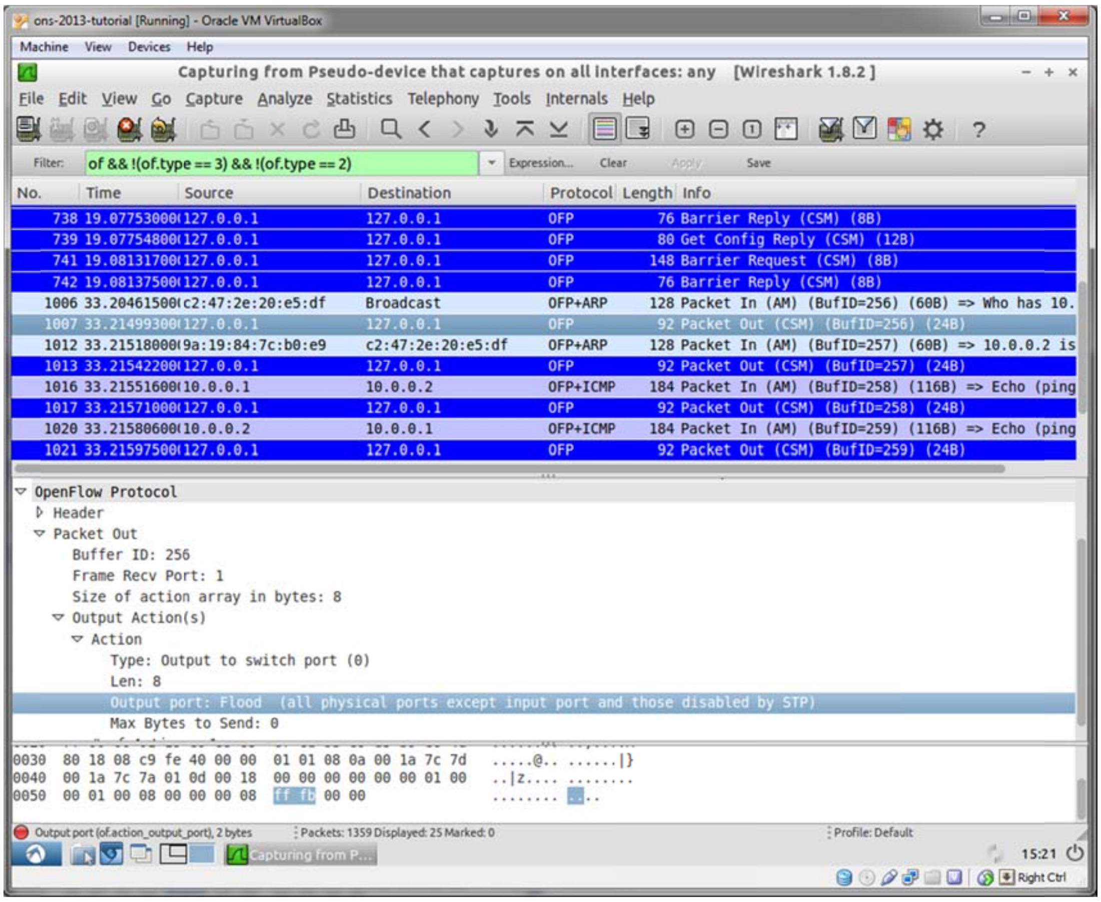
Send a single ping from h1 to h2 and check;

Wireshark view: first Packet Out's output port should be Flood and Subsequent Packet Out actions should be directed to a single port;

Test the speed using iperf.

得到的结果包括：

Wireshark中第一个packetout的output port是flood，第二个则是to switch port：



ons-2013-tutorial [Running] - Oracle VM VirtualBox
Machine View Devices Help
Capturing from Pseudo-device that captures on all interfaces: any [Wireshark 1.8.2]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
Profile: Default
Capturing from P...
15:21
Right Ctrl

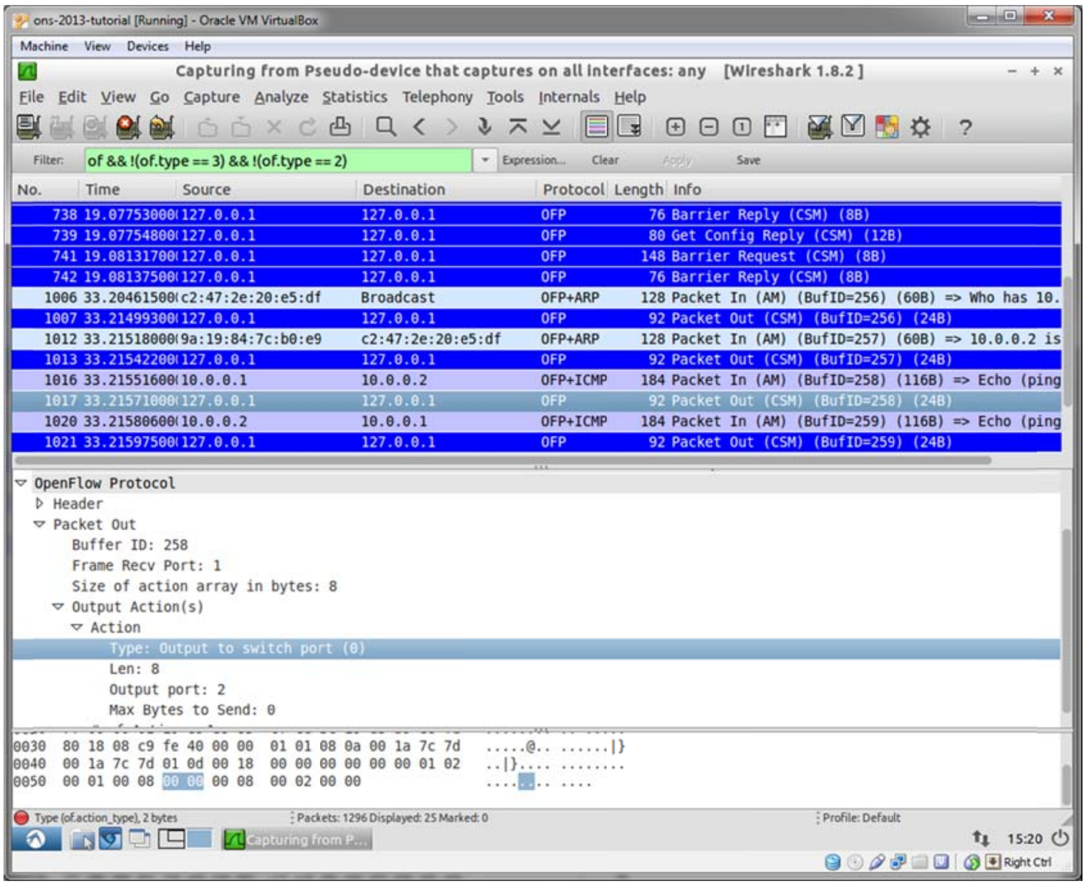
Filter: `of && !(of.type == 3) && !(of.type == 2)`

```
No.   Time             Source              Destination         Protocol Length Info
 738 19.07753000(127.0.0.1           127.0.0.1           OFP          76 Barrier Reply (CSM) (8B)
 739 19.07754800(127.0.0.1           127.0.0.1           OFP          80 Get Config Reply (CSM) (12B)
 741 19.08131700(127.0.0.1           127.0.0.1           OFP         148 Barrier Request (CSM) (8B)
 742 19.08137500(127.0.0.1           127.0.0.1           OFP          76 Barrier Reply (CSM) (8B)
1006 33.20461500(c2:47:2e:20:e5:df   Broadcast           OFP+ARP     128 Packet In (AM) (BufID=256) (60B) => Who has 10.
1007 33.21499300(127.0.0.1           127.0.0.1           OFP          92 Packet Out (CSM) (BufID=256) (24B)
1012 33.21518000(9a:19:84:7c:b0:e9   c2:47:2e:20:e5:df   OFP+ARP     128 Packet In (AM) (BufID=257) (60B) => 10.0.0.2 is
1013 33.21542200(127.0.0.1           127.0.0.1           OFP          92 Packet Out (CSM) (BufID=257) (24B)
1016 33.21551600(10.0.0.1            10.0.0.2            OFP+ICMP    184 Packet In (AM) (BufID=258) (116B) => Echo (ping
1017 33.21571000(127.0.0.1           127.0.0.1           OFP          92 Packet Out (CSM) (BufID=258) (24B)
1020 33.21580600(10.0.0.2            10.0.0.1            OFP+ICMP    184 Packet In (AM) (BufID=259) (116B) => Echo (ping
1021 33.21597500(127.0.0.1           127.0.0.1           OFP          92 Packet Out (CSM) (BufID=259) (24B)
```

```
▽ OpenFlow Protocol
  ▷ Header
  ▽ Packet Out
      Buffer ID: 256
      Frame Recv Port: 1
      Size of action array in bytes: 8
    ▽ Output Action(s)
      ▽ Action
          Type: Output to switch port (0)
          Len: 8
          Output port: Flood  (all physical ports except input port and those disabled by STP)
          Max Bytes to Send: 0
```

```
0030  80 18 08 c9 fe 40 00 00  01 01 08 0a 00 1a 7c 7d   .....@.. ......|}
0040  00 1a 7c 7a 01 0d 00 18  00 00 00 00 00 00 01 00   ..|z.... ........
0050  00 01 00 08 00 00 00 08  ff fb 00 00               ........ ....
```

Output port (of.action_output_port), 2 bytes   Packets: 1359 Displayed: 25 Marked: 0



ons-2013-tutorial [Running] - Oracle VM VirtualBox
Machine View Devices Help
Capturing from Pseudo-device that captures on all interfaces: any [Wireshark 1.8.2]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
Profile: Default
Capturing from P...
15:20
Right Ctrl

Filter: `of && !(of.type == 3) && !(of.type == 2)`

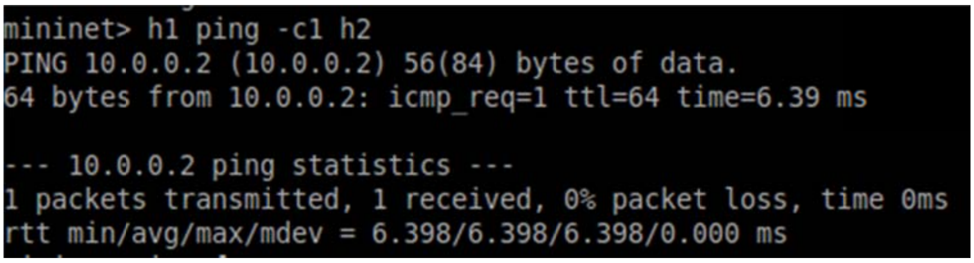
```
No.   Time             Source              Destination         Protocol Length Info
 738 19.07753000(127.0.0.1           127.0.0.1           OFP          76 Barrier Reply (CSM) (8B)
 739 19.07754800(127.0.0.1           127.0.0.1           OFP          80 Get Config Reply (CSM) (12B)
 741 19.08131700(127.0.0.1           127.0.0.1           OFP         148 Barrier Request (CSM) (8B)
 742 19.08137500(127.0.0.1           127.0.0.1           OFP          76 Barrier Reply (CSM) (8B)
1006 33.20461500(c2:47:2e:20:e5:df   Broadcast           OFP+ARP     128 Packet In (AM) (BufID=256) (60B) => Who has 10.
1007 33.21499300(127.0.0.1           127.0.0.1           OFP          92 Packet Out (CSM) (BufID=256) (24B)
1012 33.21518000(9a:19:84:7c:b0:e9   c2:47:2e:20:e5:df   OFP+ARP     128 Packet In (AM) (BufID=257) (60B) => 10.0.0.2 is
1013 33.21542200(127.0.0.1           127.0.0.1           OFP          92 Packet Out (CSM) (BufID=257) (24B)
1016 33.21551600(10.0.0.1            10.0.0.2            OFP+ICMP    184 Packet In (AM) (BufID=258) (116B) => Echo (ping
1017 33.21571000(127.0.0.1           127.0.0.1           OFP          92 Packet Out (CSM) (BufID=258) (24B)
1020 33.21580600(10.0.0.2            10.0.0.1            OFP+ICMP    184 Packet In (AM) (BufID=259) (116B) => Echo (ping
1021 33.21597500(127.0.0.1           127.0.0.1           OFP          92 Packet Out (CSM) (BufID=259) (24B)
```

```
▽ OpenFlow Protocol
  ▷ Header
  ▽ Packet Out
      Buffer ID: 258
      Frame Recv Port: 1
      Size of action array in bytes: 8
    ▽ Output Action(s)
      ▽ Action
          Type: Output to switch port (0)
          Len: 8
          Output port: 2
          Max Bytes to Send: 0
```

```
0030  80 18 08 c9 fe 40 00 00  01 01 08 0a 00 1a 7c 7d   .....@.. ......|}
0040  00 1a 7c 7d 01 0d 00 18  00 00 00 00 00 00 01 02   ..|}.... ........
0050  00 01 00 08 00 00 00 08  00 02 00 00               ........ ....
```

Type (of.action_type), 2 bytes   Packets: 1296 Displayed: 25 Marked: 0

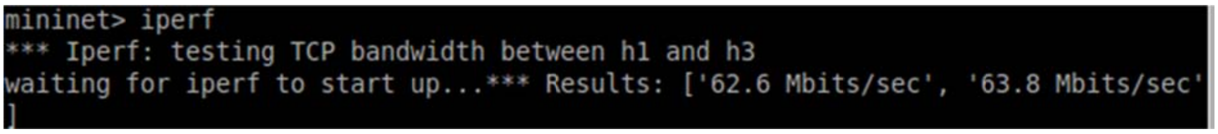
二个host之间可以ping通：



```
mininet> h1 ping -c1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_req=1 ttl=64 time=6.39 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 6.398/6.398/6.398/0.000 ms
```

测速的结果是：



```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
waiting for iperf to start up...*** Results: ['62.6 Mbits/sec', '63.8 Mbits/sec'
]
```

5．进行实验 2

在“Create Learning Switch”中找到phase2，开始实验。在java代码中需要做的改动有：

Install a flow in the network you will create an OFFlowMod object;

Initialize buffer id, match, command, idle timeout and actions;

Create the action that OFFlowMod outputs to the port learned before and set it on the OFFlowMod instance;

Send a message to an OpenFlow switch.

测试步骤是：

Start Eclipse and run the Tutorial Controller, start Wireshark;

Start Mininet (only if not already started) and wait for Beacon's console to report the switch has connected;

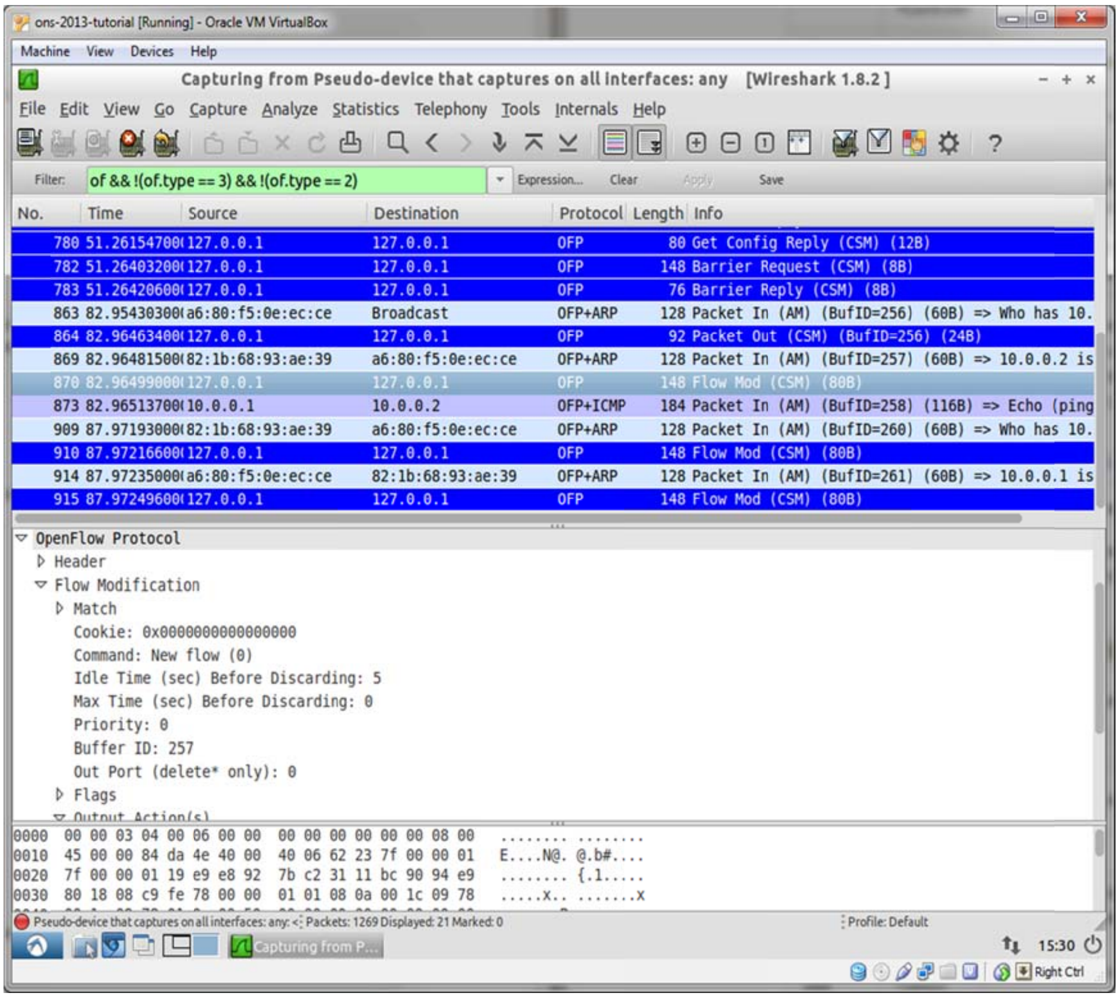
Send a single ping from h1 to h2 and check;

Wireshark view: first Packet Out's output port should be Flood and Subsequent Packet Out actions should be directed to a single port;

Test the speed using iperf.

得到的结果包括：

Wireshark中可以观察到flowmod：


ons-2013-tutorial [Running] - Oracle VM VirtualBox
Machine View Devices Help
Capturing from Pseudo-device that captures on all interfaces: any [Wireshark 1.8.2]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: of && !(of.type == 3) && !(of.type == 2)
Expression... Clear Apply Save
Profile: Default
Capturing from P...
15:30
Right Ctrl

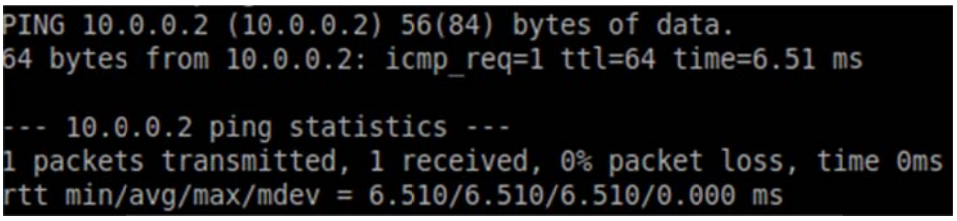
```
No.  Time              Source              Destination         Protocol  Length Info
 780 51.26154700(127.0.0.1              127.0.0.1           OFP          80 Get Config Reply (CSM) (12B)
 782 51.26403200(127.0.0.1              127.0.0.1           OFP         148 Barrier Request (CSM) (8B)
 783 51.26420600(127.0.0.1              127.0.0.1           OFP          76 Barrier Reply (CSM) (8B)
 863 82.95430300(a6:80:f5:0e:ec:ce      Broadcast           OFP+ARP     128 Packet In (AM) (BufID=256) (60B) => Who has 10.
 864 82.96463400(127.0.0.1              127.0.0.1           OFP          92 Packet Out (CSM) (BufID=256) (24B)
 869 82.96481500(82:1b:68:93:ae:39      a6:80:f5:0e:ec:ce   OFP+ARP     128 Packet In (AM) (BufID=257) (60B) => 10.0.0.2 is
 870 82.96499000(127.0.0.1              127.0.0.1           OFP         148 Flow Mod (CSM) (80B)
 873 82.96513700(10.0.0.1               10.0.0.2            OFP+ICMP    184 Packet In (AM) (BufID=258) (116B) => Echo (ping
 909 87.97193000(82:1b:68:93:ae:39      a6:80:f5:0e:ec:ce   OFP+ARP     128 Packet In (AM) (BufID=260) (60B) => Who has 10.
 910 87.97216600(127.0.0.1              127.0.0.1           OFP         148 Flow Mod (CSM) (80B)
 914 87.97235000(a6:80:f5:0e:ec:ce      82:1b:68:93:ae:39   OFP+ARP     128 Packet In (AM) (BufID=261) (60B) => 10.0.0.1 is
 915 87.97249600(127.0.0.1              127.0.0.1           OFP         148 Flow Mod (CSM) (80B)

▽ OpenFlow Protocol
  ▷ Header
  ▽ Flow Modification
    ▷ Match
      Cookie: 0x0000000000000000
      Command: New flow (0)
      Idle Time (sec) Before Discarding: 5
      Max Time (sec) Before Discarding: 0
      Priority: 0
      Buffer ID: 257
      Out Port (delete* only): 0
    ▷ Flags
    ▽ Output Action(s)

0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00   ........ ........
0010  45 00 00 84 da 4e 40 00  40 06 62 23 7f 00 00 01   E....N@. @.b#....
0020  7f 00 00 01 19 e9 e8 92  7b c2 31 11 bc 90 94 e9   ........ {.1.....
0030  80 18 08 c9 fe 78 00 00  01 01 08 0a 00 1c 09 78   .....x.. .......x

Pseudo-device that captures on all interfaces: any: <  Packets: 1269 Displayed: 21 Marked: 0
```

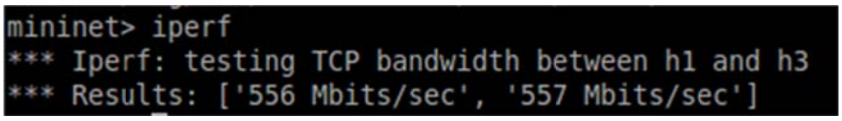
二个host之间可以ping通：



```
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_req=1 ttl=64 time=6.51 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 6.510/6.510/6.510/0.000 ms
```

测速的结果是：



```
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
*** Results: ['556 Mbits/sec', '557 Mbits/sec']
```

6. 进行实验 3

在“Create Learning Switch”中找到Extra Credit 1，开始实验。在java代码中需要做的改动有：

Change the existing single Map macTable to a Map of macTables, indexed by the switch;

Create a macTable once for each switch, and store it into the macTables Map;

In the forwardAsLearningSwitch method, retrieve the proper macTable to use for the current OFPacketIn.

测试步骤是：

Start Eclipse and run the Tutorial Controller, start Wireshark;

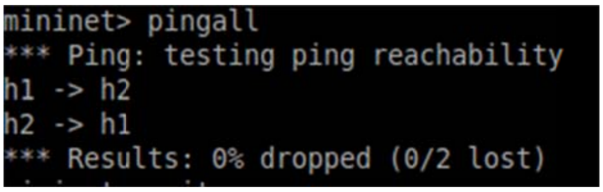
Start Mininet (only if not already started) and wait for Beacon's console to report the switch has connected;

Send a single ping from h1 to h2 and check;

Wireshark view: first Packet Out's output port should be Flood and Subsequent Packet Out actions should be directed to a single port;

Test the speed using iperf.

得到的结果包括二个host之间可以ping通：


```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (0/2 lost)
```

7．进行实验 4

在“Create Learning Switch”中找到Extra Credit 2，开始实验。在java代码中需要做的改动有：

After sending the Flow Mod from phase 2, test if the OFPacketIn's buffer id is none;

Create an OFPacketOut object like phase 1..

测试步骤是：

Start Eclipse and run the Tutorial Controller, start Wireshark;

Start Mininet (only if not already started) and wait for Beacon's console to report the switch has connected;

Send a single ping from h1 to h2 and check;

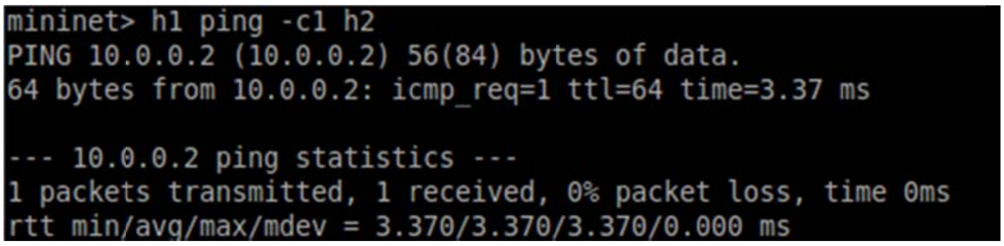
Wireshark view: first Packet Out's output port should be Flood and Subsequent

Packet Out actions should be directed to a single port;

Test the speed using iperf.

得到的结果包括：

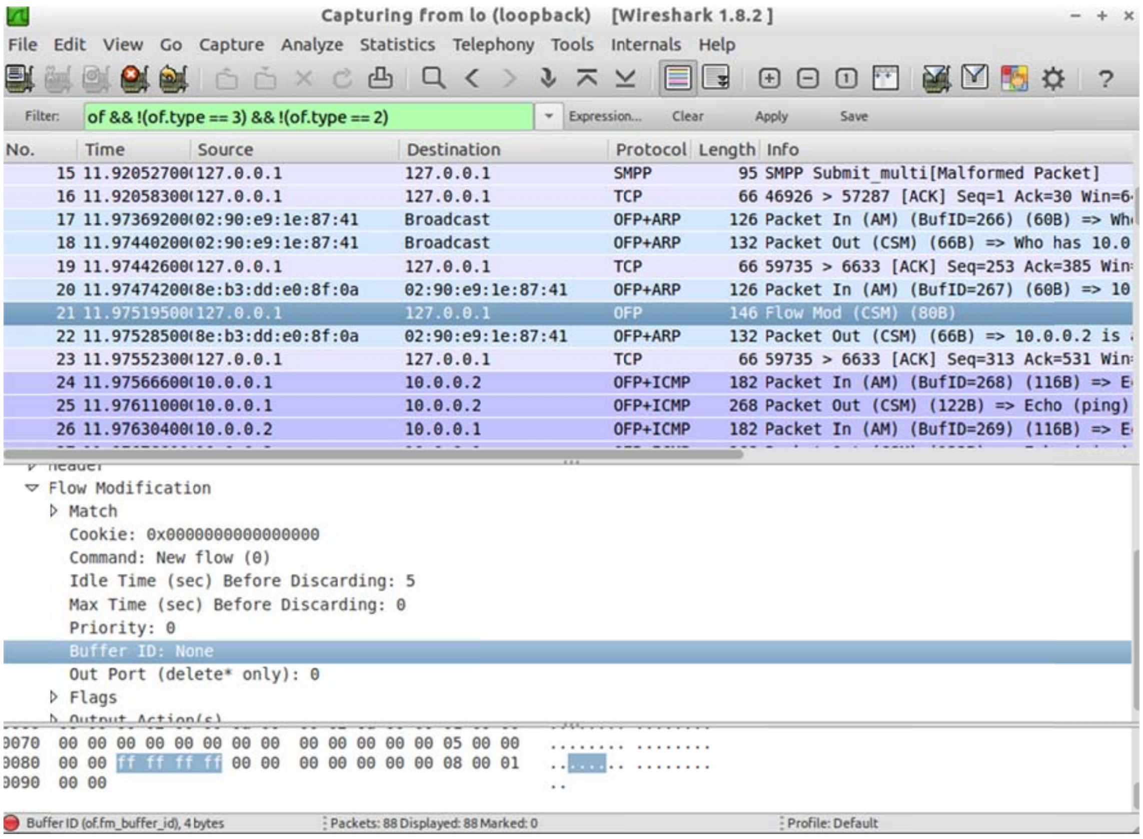
二个host之间可以ping通：



```
mininet> h1 ping -c1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_req=1 ttl=64 time=3.37 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.370/3.370/3.370/0.000 ms
```

Buffer id是none：


Capturing from lo (loopback)   [Wireshark 1.8.2]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

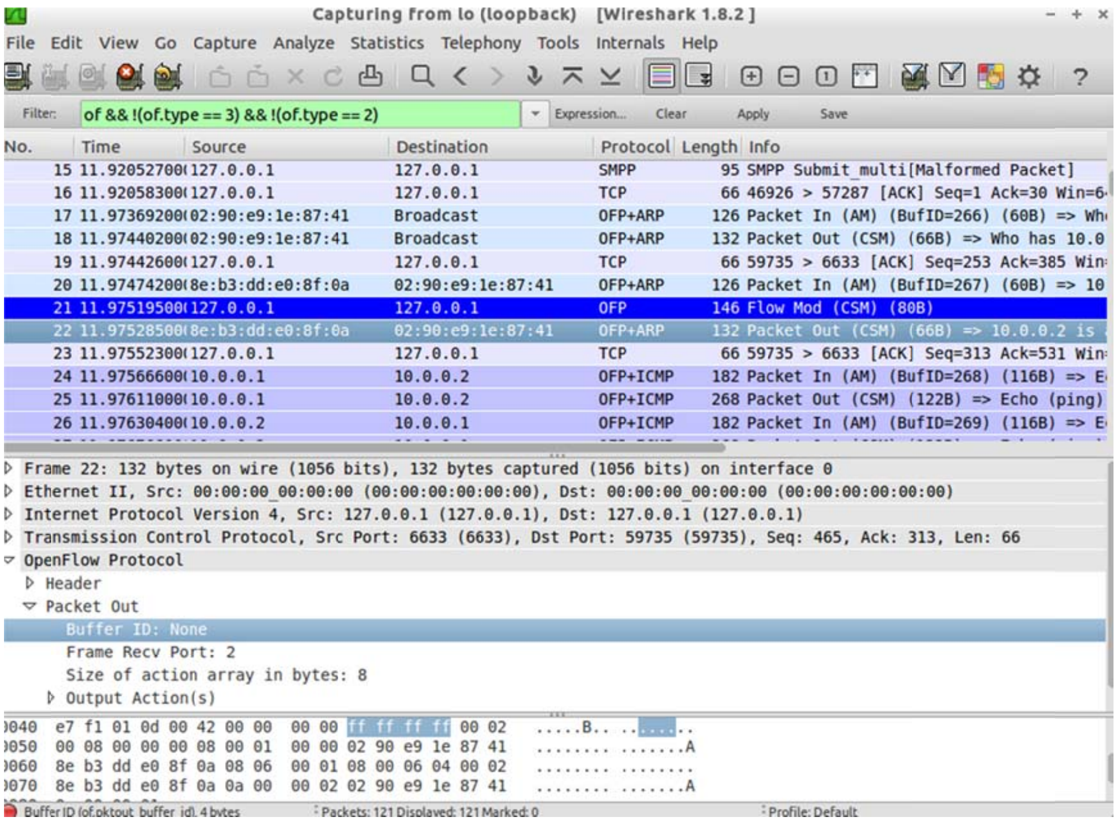
```
of && !(of.type == 3) && !(of.type == 2)
```

```
No.  Time          Source             Destination        Protocol  Length Info
 15 11.92052700(127.0.0.1            127.0.0.1          SMPP          95 SMPP Submit_multi[Malformed Packet]
 16 11.92058300(127.0.0.1            127.0.0.1          TCP           66 46926 > 57287 [ACK] Seq=1 Ack=30 Win=6
 17 11.97369200(02:90:e9:1e:87:41    Broadcast          OFP+ARP      126 Packet In (AM) (BufID=266) (60B) => Wh
 18 11.97440200(02:90:e9:1e:87:41    Broadcast          OFP+ARP      132 Packet Out (CSM) (66B) => Who has 10.0
 19 11.97442600(127.0.0.1            127.0.0.1          TCP           66 59735 > 6633 [ACK] Seq=253 Ack=385 Win
 20 11.97474200(8e:b3:dd:e0:8f:0a    02:90:e9:1e:87:41  OFP+ARP      126 Packet In (AM) (BufID=267) (60B) => 10
 21 11.97519500(127.0.0.1            127.0.0.1          OFP          146 Flow Mod (CSM) (80B)
 22 11.97528500(8e:b3:dd:e0:8f:0a    02:90:e9:1e:87:41  OFP+ARP      132 Packet Out (CSM) (66B) => 10.0.0.2 is
 23 11.97552300(127.0.0.1            127.0.0.1          TCP           66 59735 > 6633 [ACK] Seq=313 Ack=531 Win
 24 11.97566600(10.0.0.1             10.0.0.2           OFP+ICMP     182 Packet In (AM) (BufID=268) (116B) => E
 25 11.97611000(10.0.0.1             10.0.0.2           OFP+ICMP     268 Packet Out (CSM) (122B) => Echo (ping)
 26 11.97630400(10.0.0.2             10.0.0.1           OFP+ICMP     182 Packet In (AM) (BufID=269) (116B) => E
```

```
  Flow Modification
    Match
    Cookie: 0x0000000000000000
    Command: New flow (0)
    Idle Time (sec) Before Discarding: 5
    Max Time (sec) Before Discarding: 0
    Priority: 0
    Buffer ID: None
    Out Port (delete* only): 0
    Flags
```

```
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 05 00 00   ........ ........
0080  00 00 ff ff ff ff 00 00  00 00 00 00 00 08 00 01   ........ ........
0090  00 00                                              ..
```

Buffer ID (of.fm_buffer_id), 4 bytes   Packets: 88 Displayed: 88 Marked: 0   Profile: Default


Capturing from lo (loopback) [Wireshark 1.8.2]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
of && !(of.type == 3) && !(of.type == 2)

No.   Time               Source              Destination         Protocol  Length Info
   15 11.92052700(127.0.0.1            127.0.0.1            SMPP          95 SMPP Submit_multi[Malformed Packet]
   16 11.92058300(127.0.0.1            127.0.0.1            TCP           66 46926 > 57287 [ACK] Seq=1 Ack=30 Win=6
   17 11.97369200(02:90:e9:1e:87:41    Broadcast            OFP+ARP      126 Packet In (AM) (BufID=266) (60B) => Wh
   18 11.97440200(02:90:e9:1e:87:41    Broadcast            OFP+ARP      132 Packet Out (CSM) (66B) => Who has 10.0
   19 11.97442600(127.0.0.1            127.0.0.1            TCP           66 59735 > 6633 [ACK] Seq=253 Ack=385 Win
   20 11.97474200(8e:b3:dd:e0:8f:0a    02:90:e9:1e:87:41    OFP+ARP      126 Packet In (AM) (BufID=267) (60B) => 10
   21 11.97519500(127.0.0.1            127.0.0.1            OFP          146 Flow Mod (CSM) (80B)
   22 11.97528500(8e:b3:dd:e0:8f:0a    02:90:e9:1e:87:41    OFP+ARP      132 Packet Out (CSM) (66B) => 10.0.0.2 is
   23 11.97552300(127.0.0.1            127.0.0.1            TCP           66 59735 > 6633 [ACK] Seq=313 Ack=531 Win
   24 11.97566600(10.0.0.1             10.0.0.2             OFP+ICMP     182 Packet In (AM) (BufID=268) (116B) => E
   25 11.97611000(10.0.0.1             10.0.0.2             OFP+ICMP     268 Packet Out (CSM) (122B) => Echo (ping)
   26 11.97630400(10.0.0.2             10.0.0.1             OFP+ICMP     182 Packet In (AM) (BufID=269) (116B) => E

▷ Frame 22: 132 bytes on wire (1056 bits), 132 bytes captured (1056 bits) on interface 0
▷ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▷ Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▷ Transmission Control Protocol, Src Port: 6633 (6633), Dst Port: 59735 (59735), Seq: 465, Ack: 313, Len: 66
▽ OpenFlow Protocol
  ▷ Header
  ▽ Packet Out
      Buffer ID: None
      Frame Recv Port: 2
      Size of action array in bytes: 8
    ▷ Output Action(s)

)040  e7 f1 01 0d 00 42 00 00   00 00 ff ff ff ff 00 02   .....B.. ........
)050  00 08 00 00 00 08 00 01   00 00 02 90 e9 1e 87 41   ........ .......A
)060  8e b3 dd e0 8f 0a 08 06   00 01 08 00 06 04 00 02   ........ ........
)070  8e b3 dd e0 8f 0a 0a 00   00 02 02 90 e9 1e 87 41   ........ .......A

Buffer ID (of.pktout_buffer_id), 4 bytes    Packets: 121 Displayed: 121 Marked: 0    Profile: Default
```