Slices_Demo3

环境要求：

Ubuntu 14.04，64bit； p4c-bm； bmv2。

实验准备：

请根据实际情况对各个脚本中的路径信息进行修改。

我的实验环境中bmv2的路径：`/home/wasdns/bmv2`

实验步骤：

1.启动mininet：

```
./run_demo.sh
```

此时节点信息：

两台主机：h1(IP: 10.0.0.10/32)，h2(IP: 10.0.1.10/32)。

一台交换机：s1。

2.执行pingall，h1与h2无法ping通。

3.打开用于控制交换机的CLI界面，下发命令使h1与h2互ping通。

执行命令如下：

```
table_add send_frame rewrite_mac 1 => 00:aa:bb:00:00:00
table_add send_frame rewrite_mac 2 => 00:aa:bb:00:00:01
table_add forward set_dmac 10.0.0.10 => 00:04:00:00:00:00
table_add forward set_dmac 10.0.1.10 => 00:04:00:00:00:01
table_add ipv4_lpm set_nhop 10.0.0.10/32 => 10.0.0.10 1
table_add ipv4_lpm set_nhop 10.0.1.10/32 => 10.0.1.10 2
```

4.在mininet中执行pingall验证h1与h2能够正常通信。

```
> pingall
```

5.为Match_Table增加一条表项，使计数器实例0开始计数：当经过交换机的包的源IP地址是10.0.0.10时，进行计数。

P4实现：

```
counter Indecounter {
    type : packets;
    static : Match_Table;
    instance_count : 16384;
}

action count_action(index) {
    count(Indecounter, index);
}

table Match_Table {
    reads {
        ipv4.srcAddr : exact;
    }
    actions {
        _noop;
        count_action;
    }
}
```

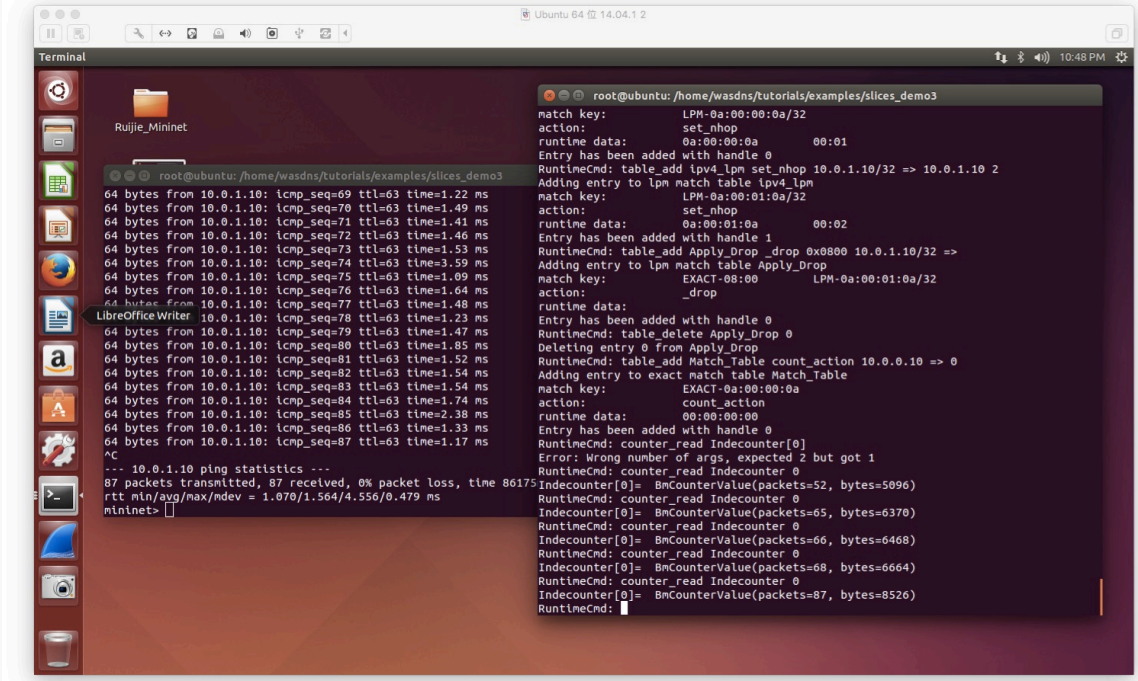
在CLI中下发命令：

```
table_add Match_Table count_action 10.0.0.10 => 0
```

6.验证计数器能够进行计数：

在mininet中执行`h1 ping h2`；

在CLI界面中执行`counter_read Indecounter 0`查看计数器实例0的信息。



7.计数达到100个包之后执行丢包。

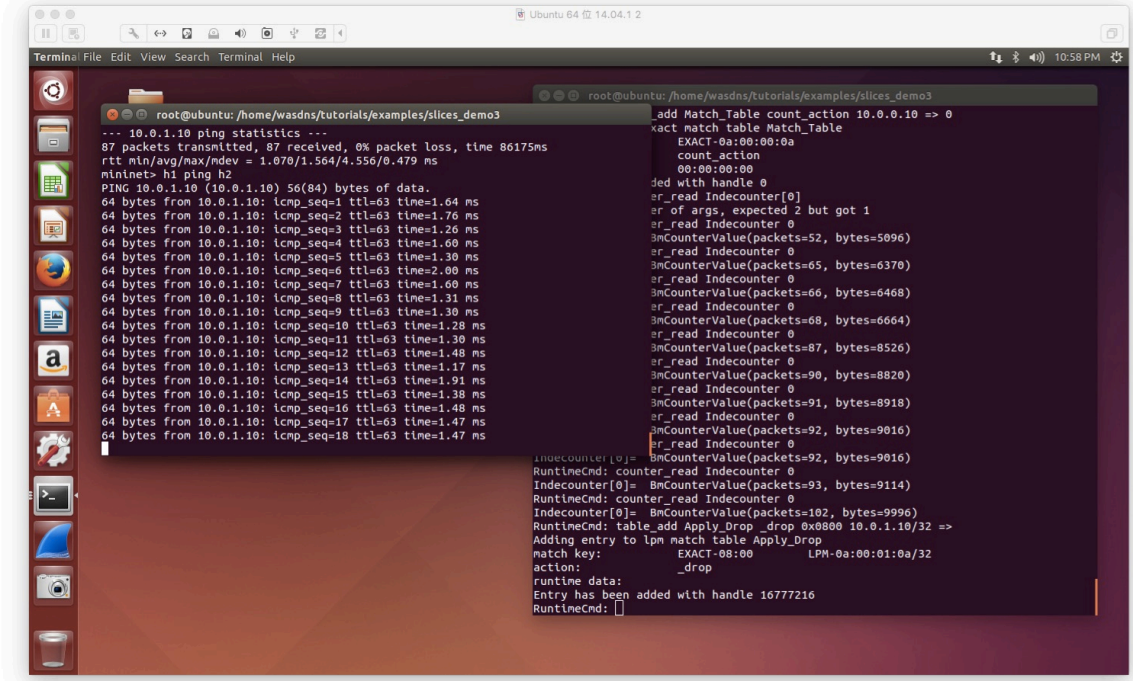
在CLI中通过命令：

```
counter_read Indecounter 0
```

查看计数器实例0的信息，发现超过100个包时，下发命令：

```
table_add Apply Drop 10.0.0.10/32 0x0800 =>
```

使交换机执行丢包，此时h1无法ping通h2。




```
--- 10.0.1.10 ping statistics ---
144 packets transmitted, 18 received, 87% packet loss, time 143057ms
rtt min/avg/max/mdev = 1.173/1.488/2.001/0.224 ms
mininet>
```

实验结论

通过作为控制平面的CLI，我们可以实时查看交换机中的计数器信息，并且可以通过CLI下发指令主机之间的正常通信。

> 2017/1/20