

# [CS6250] CN P4 Assignment 0

Team 108

## Members

Abrar Ahmed Mohammed , Rohith Sudheer , Simran Gade

## repeater.p4

```
action forward(bit<9> egress_port){
    standard_metadata.egress_spec = egress_port;
}

table repeater {
    key = {
        standard_metadata.ingress_port: exact; /* TODO 1: Match on the ingress port */
    }
    actions = {
        forward;
        NoAction; /* NoAction results in the matched packets being dropped */
    }
    size = 2; /* TODO 1: Define the size of the table, i.e., the max number of entries */
    default_action = NoAction;
}

apply {
    repeater.apply();
}
```

Figure 1: Ingress portion of repeater.py

## controller.py

```
controller = controllers['s2']
controller.table_clear('repeater')
controller.table_add('repeater', 'forward', ['1'], ['2'])

controller = controllers['s3']
controller.table_clear('repeater')
controller.table_add('repeater', 'forward', ['1'], ['2'])
controller.table_add('repeater', 'forward', ['2'], ['3'])

controller = controllers['s4']
controller.table_clear('repeater')
controller.table_add('repeater', 'forward', ['1'], ['2'])
```

Figure 2: Modified portion of repeater.py

## receive.py

```
root@p4:~/assignment_0#  
root@p4:~/assignment_0#  
root@p4:~/assignment_0#  
root@p4:~/assignment_0#  
root@p4:~/assignment_0#  
root@p4:~/assignment_0# python send.py 10.0.0.2 "Hello1234"  
Sending on interface h1-eth0 to 10.0.0.2  
root@p4:~/assignment_0#  
root@p4:~/assignment_0# python receive.py  
sniffing on h2-eth0  
Packet Received:  
###[ Ethernet ]###  
src: 00:00:0a:00:00:01  
dst: 00:00:0a:00:00:02  
###[ IP ]###  
src: 10.0.0.1  
dst: 10.0.0.2  
###[ MESSAGE ]###  
msg: b'Hello1234'
```

Figure 3: receive.py output on h2

## Match/action table size

We chose the size of the table as 2 since there would be at most 2 rules for each table. In  $s1$  and  $s3$ , only 2 of the ports should receive packets, so there are only 2 rules required. For  $s2$  and  $s4$ , only 1 port should receive packets, so only one rule is required.

## tcpdump on port 2 of switch s1 showing packets from h1 to h2

```
p4@p4:~$ sudo tcpdump -n -i s1-eth2  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on s1-eth2, link-type EN10MB (Ethernet), capture size 262144 bytes  
16:50:15.474758 IP 10.0.0.1 > 10.0.0.2: ip-proto-0 6  
16:50:19.154907 IP 10.0.0.1 > 10.0.0.2: ip-proto-0 6  
^C  
2 packets captured  
2 packets received by filter  
0 packets dropped by kernel  
p4@p4:~$
```

Figure 4: tcpdump on s1-eth2 for traffic from h1 to h2

## tcpdump on port 3 of switch s1 that showing from h2 to h1

```
p4@p4:~$ sudo tcpdump -n -i s1-eth3  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on s1-eth3, link-type EN10MB (Ethernet), capture size 262144 bytes  
16:48:12.119863 IP 10.0.0.2 > 10.0.0.1: ip-proto-0 4  
16:48:15.343650 IP 10.0.0.2 > 10.0.0.1: ip-proto-0 4  
^C  
2 packets captured  
2 packets received by filter  
0 packets dropped by kernel  
p4@p4:~$
```

Figure 5: tcpdump on s1-eth2 for traffic from h2 to h1