SDN Chapter6 Demo

Updated: Mar 30th, 2021

Jing Yan (yanj3, jing.yan@aalto.fi)

TASK 1

Code (Detailed code can be seen in the attachment)

For the configuration of “forward.conf”

```
ubuntu@ubunu2004:~/p4app/examples/forward.p4app$ cat forward.config
# Rewrite the srcmac(h2's mac) and dstmac(h1's mac) for packets sent out from
egress port1(connected to h1)
table_add send_frame rewrite_mac 1 => 00:04:00:00:00:01 00:04:00:00:00:00
# Rewrite the srcmac(h1's mac) and dstmac(h2's mac) for packets sent out from
egress port2(connected to h2)
table_add send_frame rewrite_mac 2 => 00:04:00:00:00:00 00:04:00:00:00:01
# Match packets with dst addr as 10.0.0.10/32 and set egress port as 1
table_add ipv4_lpm set_nhop 10.0.0.10/32 => 10.0.0.10 1
# Match packets with dst addr as 10.0.1.10/32 and set egress port as 2
table_add ipv4_lpm set_nhop 10.0.1.10/32 => 10.0.1.10 2
```

For the configuration of “forward.p4”

```
ubuntu@ubunu2004:~/p4app/examples/forward.p4app$ cat forward.p4
...
control ingress(inout headers hdr, inout metadata meta, inout standard_metadata_t
standard_metadata) {
   action _drop() {
       mark_to_drop(standard_metadata);
   }
   /* Set the next hop and the output port. Decrements ipv4 ttl field. @param
ivp4_dest ipv4 address of next hop, @param port output port, */
   action set_nhop(bit<32> nhop_ipv4, bit<9> port) {
       meta.ingress_metadata.nhop_ipv4 = nhop_ipv4;
       standard_metadata.egress_spec = port;
       hdr.ipv4.ttl = hdr.ipv4.ttl + 8w255;
   }
   table ipv4_lpm {
       actions = {
           _drop;
           set_nhop;
           NoAction;
       }
       key = {
```

```
            hdr.ipv4.dstAddr: lpm;
        }
        size = 1024;
        default_action = NoAction();
    }
    apply {
        if (hdr.ipv4.isValid()) {
          ipv4_lpm.apply();
        }
    }
}
...
```

For the configuration of “parser.p4”

```
ubuntu@ubunu2004:~/p4app/examples/forward.p4app$ cat parser.p4
parser ParserImpl(packet_in packet, out headers hdr, inout metadata meta, inout
standard_metadata_t standard_metadata) {
    state parse_ethernet {
        packet.extract(hdr.ethernet);
        transition select(hdr.ethernet.etherType) {
            16w0x800: parse_ipv4;
            default: accept;
        }
    }
    state parse_ipv4 {
        packet.extract(hdr.ipv4);
        transition accept;
    }
    state start {
        transition parse_ethernet;
    }
}

control DeparserImpl(packet_out packet, in headers hdr) {
    apply {
        packet.emit(hdr.ethernet);
        packet.emit(hdr.ipv4);
    }
}

control verifyChecksum(inout headers hdr, inout metadata meta) {
    apply { }
}

control computeChecksum(inout headers hdr, inout metadata meta) {
    apply {
        update_checksum(
                hdr.ipv4.isValid(),
                { hdr.ipv4.version, hdr.ipv4.ihl, hdr.ipv4.diffserv,
                hdr.ipv4.totalLen, hdr.ipv4.identification,
                hdr.ipv4.flags, hdr.ipv4.fragOffset, hdr.ipv4.ttl,
                hdr.ipv4.protocol, hdr.ipv4.srcAddr, hdr.ipv4.dstAddr },
                hdr.ipv4.hdrChecksum,
                HashAlgorithm.csum16);
    }
}
```

Result

```
ubuntu@ubunu2004:~/p4app$ sudo p4app run examples/forward.p4app
Entering build directory.
Extracting package.
> touch /tmp/p4app_logs/p4s.s1.log
> ln -s /tmp/p4app_logs/p4s.s1.log /tmp/p4s.s1.log
Reading package manifest.
> p4c-bm2-ss --p4v 16 "forward.p4" -o "forward.json"
```

```
> python2 "/scripts/mininet/single_switch_mininet.py" --log-file
"/var/log/forward.p4.log" --cli-message "mininet_message.txt" --num-hosts 2
--switch-config "forward.config" --behavioral-exe "simple_switch" --json
"forward.json"
Adding host h1
Adding host h2
*** Error setting resource limits. Mininet's performance may be affected.
*** Creating network
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 Starting P4 switch s1.
simple_switch -i 1@s1-eth1 -i 2@s1-eth2 --thrift-port 9090 --nanolog
ipc:///tmp/bm-0-log.ipc --device-id 0 forward.json --debugger --log-console
P4 switch s1 has been started.

**********
Network configuration for: h1
Default interface: h1-eth0  10.0.0.10 00:04:00:00:00:00
Default route to switch: 10.0.0.1 (00:aa:bb:00:00:00)
**********
**********
Network configuration for: h2
Default interface: h2-eth0  10.0.1.10 00:04:00:00:00:01
Default route to switch: 10.0.1.1 (00:aa:bb:00:00:01)
**********

Reading switch configuration script: forward.config
Configuring switch...
Obtaining JSON from switch...
Done
Control utility for runtime P4 table manipulation
RuntimeCmd: Adding entry to exact match table send_frame
match key:           EXACT-00:01
action:              rewrite_mac
runtime data:        00:04:00:00:00:01  00:04:00:00:00:00
Entry has been added with handle 0
RuntimeCmd: Adding entry to exact match table send_frame
match key:           EXACT-00:02
action:              rewrite_mac
runtime data:        00:04:00:00:00:00  00:04:00:00:00:01
Entry has been added with handle 1
RuntimeCmd: Adding entry to lpm match table ipv4_lpm
match key:           LPM-0a:00:00:0a/32
action:              set_nhop
runtime data:        0a:00:00:0a  00:01
Entry has been added with handle 0
RuntimeCmd: Adding entry to lpm match table ipv4_lpm
match key:           LPM-0a:00:01:0a/32
action:              set_nhop
runtime data:        0a:00:01:0a  00:02
Entry has been added with handle 1
RuntimeCmd:
Configuration complete.

Ready !

================================================================
Welcome to the BMV2 Mininet CLI!
================================================================
```

```
Your P4 program is installed into the BMV2 software switch
and your initial configuration is loaded. You can interact
with the network using the mininet CLI below.

To inspect or change the switch configuration, connect to
its CLI from your host operating system using this command:
 docker exec -t -i ecaf7419e30b simple_switch_CLI

To view the switch log, run this command from your host OS:
 docker exec -t -i ecaf7419e30b tail -f /var/log/forward.p4.log

To run the switch debugger, run this command from your host OS:
 docker exec -t -i ecaf7419e30b bm_p4dbg


*** Starting CLI:
mininet> links
h1-eth0<->s1-eth1 (OK OK)
h2-eth0<->s1-eth2 (OK OK)
mininet> h1 ifconfig -a | grep Bcast
          inet addr:10.0.0.10  Bcast:10.0.0.255  Mask:255.255.255.0
mininet> h2 ifconfig -a | grep Bcast
          inet addr:10.0.1.10  Bcast:10.0.1.255  Mask:255.255.255.0
mininet> h1 ping -c1 h2
PING 10.0.1.10 (10.0.1.10) 56(84) bytes of data.
64 bytes from 10.0.1.10: icmp_seq=1 ttl=63 time=0.849 ms

--- 10.0.1.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.849/0.849/0.849/0.000 ms
```

TASK 2

Code (Detailed code can be seen in the attachment)

For the configuration of “forward.conf”

```
ubuntu@ubunu2004:~/p4app/examples/forward.p4app$ cat forward.config
# Set the dst mac of the packets
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
# Match packets with dst addr as 10.0.0.10/32 and set egress port as 1
table_add ipv4_lpm set_nhop 10.0.0.10/32 => 10.0.0.10 1
# Match packets with dst addr as 10.0.1.10/32 and set egress port as 2
table_add ipv4_lpm set_nhop 10.0.1.10/32 => 10.0.1.10 2
```

For the configuration of “forward.p4”

```
ubuntu@ubunu2004:~/p4app/examples/forward.p4app$ cat forward.p4
...
control ingress(inout headers hdr, inout metadata meta, inout standard_metadata_t
standard_metadata) {
   action _drop() {
       mark_to_drop(standard_metadata);
   }
    /* Set the next hop and the output port. Decrements ipv4 ttl field. @param
ivp4_dest ipv4 address of next hop, @param port output port, */
   action set_nhop(bit<32> nhop_ipv4, bit<9> port) {
       meta.ingress_metadata.nhop_ipv4 = nhop_ipv4;
       standard_metadata.egress_spec = port;
       hdr.ipv4.ttl = hdr.ipv4.ttl + 8w255;
   }
    /* Set the destination MAC address of the packet, @param dmac destination MAC
address. */
   action set_dmac(bit<48> dmac) {
       hdr.ethernet.dstAddr = dmac;
```

```
    }
    table ipv4_lpm {
        actions = {
            _drop;
            set_nhop;
            NoAction;
        }
        key = {
            hdr.ipv4.dstAddr: lpm;
        }
        size = 1024;
        default_action = NoAction();
    }
    table forward {
        actions = {
            set_dmac;
            _drop;
            NoAction;
        }
        key = {
            meta.ingress_metadata.nhop_ipv4: exact;
        }
        size = 512;
        default_action = NoAction();
    }
    apply {
        if (hdr.ipv4.isValid()) {
          ipv4_lpm.apply();
          forward.apply();
        }
    }
}
...
```

For the configuration of “parser.p4”

```
ubuntu@ubunu2004:~/p4app/examples/forward.p4app$ cat parser.p4
parser ParserImpl(packet_in packet, out headers hdr, inout metadata meta, inout
standard_metadata_t standard_metadata) {
    state parse_ethernet {
        packet.extract(hdr.ethernet);
        transition select(hdr.ethernet.etherType) {
            16w0x800: parse_ipv4;
            default: accept;
        }
    }
    state parse_ipv4 {
        packet.extract(hdr.ipv4);
        transition accept;
    }
    state start {
        transition parse_ethernet;
    }
}

control DeparserImpl(packet_out packet, in headers hdr) {
    apply {
        packet.emit(hdr.ethernet);
        packet.emit(hdr.ipv4);
    }
}

control verifyChecksum(inout headers hdr, inout metadata meta) {
    apply { }
}

control computeChecksum(inout headers hdr, inout metadata meta) {
    apply {
```

```
        update_checksum(
                hdr.ipv4.isValid(),
                { hdr.ipv4.version, hdr.ipv4.ihl, hdr.ipv4.diffserv,
                hdr.ipv4.totalLen, hdr.ipv4.identification,
                hdr.ipv4.flags, hdr.ipv4.fragOffset, hdr.ipv4.ttl,
                hdr.ipv4.protocol, hdr.ipv4.srcAddr, hdr.ipv4.dstAddr },
                hdr.ipv4.hdrChecksum,
                HashAlgorithm.csum16);
    }
}
```

Result

```
ubuntu@ubunu2004:~/p4app$ sudo p4app run examples/forward.p4app
Entering build directory.
Extracting package.
> touch /tmp/p4app_logs/p4s.s1.log
> ln -s /tmp/p4app_logs/p4s.s1.log /tmp/p4s.s1.log
Reading package manifest.
> p4c-bm2-ss --p4v 16 "forward.p4" -o "forward.json"
> python2 "/scripts/mininet/single_switch_mininet.py" --log-file
"/var/log/forward.p4.log" --cli-message "mininet_message.txt" --num-hosts 2
--switch-config "forward.config" --behavioral-exe "simple_switch" --json
"forward.json"
Adding host h1
Adding host h2
*** Error setting resource limits. Mininet's performance may be affected.
*** Creating network
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 Starting P4 switch s1.
simple_switch -i 1@s1-eth1 -i 2@s1-eth2 --thrift-port 9090 --nanolog
ipc:///tmp/bm-0-log.ipc --device-id 0 forward.json --debugger --log-console
P4 switch s1 has been started.

**********
Network configuration for: h1
Default interface: h1-eth0  10.0.0.10 00:04:00:00:00:00
Default route to switch: 10.0.0.1 (00:aa:bb:00:00:00)
**********
**********
Network configuration for: h2
Default interface: h2-eth0  10.0.1.10 00:04:00:00:00:01
Default route to switch: 10.0.1.1 (00:aa:bb:00:00:01)
**********

Reading switch configuration script: forward.config
Configuring switch...
Obtaining JSON from switch...
Done
Control utility for runtime P4 table manipulation
RuntimeCmd: Adding entry to exact match table forward
match key:           EXACT-0a:00:00:0a
action:              set_dmac
runtime data:        00:04:00:00:00:00
Entry has been added with handle 0
RuntimeCmd: Adding entry to exact match table forward
match key:           EXACT-0a:00:01:0a
```

```
action:              set_dmac
runtime data:        00:04:00:00:00:01
Entry has been added with handle 1
RuntimeCmd: Adding entry to lpm match table ipv4_lpm
match key:           LPM-0a:00:00:0a/32
action:              set_nhop
runtime data:        0a:00:00:0a  00:01
Entry has been added with handle 0
RuntimeCmd: Adding entry to lpm match table ipv4_lpm
match key:           LPM-0a:00:01:0a/32
action:              set_nhop
runtime data:        0a:00:01:0a  00:02
Entry has been added with handle 1
RuntimeCmd:
Configuration complete.

Ready !

================================================================
Welcome to the BMV2 Mininet CLI!
================================================================
Your P4 program is installed into the BMV2 software switch
and your initial configuration is loaded. You can interact
with the network using the mininet CLI below.

To inspect or change the switch configuration, connect to
its CLI from your host operating system using this command:
 docker exec -t -i cc4bf632af0f simple_switch_CLI

To view the switch log, run this command from your host OS:
 docker exec -t -i cc4bf632af0f tail -f /var/log/forward.p4.log

To run the switch debugger, run this command from your host OS:
 docker exec -t -i cc4bf632af0f bm_p4dbg


*** Starting CLI:
mininet> h1 ifconfig -a | grep Bcast
         inet addr:10.0.0.10  Bcast:10.0.0.255  Mask:255.255.255.0
mininet> h2 ifconfig -a | grep Bcast
         inet addr:10.0.1.10  Bcast:10.0.1.255  Mask:255.255.255.0
mininet> h1 ping -c1 h2
PING 10.0.1.10 (10.0.1.10) 56(84) bytes of data.
64 bytes from 10.0.1.10: icmp_seq=1 ttl=63 time=3.56 ms

--- 10.0.1.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.561/3.561/3.561/0.000 ms
```