Introductory and Advanced Topics on P4 Programmable Data Plane Switches

Ali AlSabeh, Jose Gomez
University of South Carolina
http://ce.sc.edu/cyberinfra
gomezgaj@email.sc.edu, gomezgaj@email.sc.edu

WASTC 2022 virtual Faculty Development Weeks (vFDW)
June 13-17, 2022

Measuring Flow Statistics using Direct and Indirect Counters

Lab activities are described in Lab 7, P4 Programmable Data Planes: Applications, Stateful Elements, and Custom Packet Processing lab series

P4 Counters

- Counters are stateful elements used for monitoring tasks such as:
  - Collecting statistics from flows
  - Enforcing Quality of Service (QoS) policies
  - Implementing security features (e.g., detecting and blocking Denial of Service (DoS) attacks)
- The V1Model provides counters as extern objects that can be invoked using the P4 language
- Counters in P4 support packet counters, byte counters, and the combination of both
- A P4 program can update counters but cannot read them
- The control plane can read counter values and use them to implement applications
- P4 offers two types of counters: direct and indirect counters

P4 Direct counters

- Direct counters: these are counters that are directly associated to a match-action table
  - Indirect counter: independent counters that can be referred to specific entries or group of entries in a match-action table
- A P4 program can update counters but cannot read them
- The control plane can read counter values and use them to implement applications

P4 Direct counters

- Direct counters are directly associated to a match-action table

```
 1: control MyIngress(inout header hdr,
 2:                   inout metadata meta,
 3:                   inout standard_metadata_t standard_metadata){
 4:
 5: direct_counter(counterType.packets) my_direct_counter;
 6:
 7: action forward(egressSpect_t port){
 8:       standard_meadata.egress_spec = port;
 9: }
10:
11: action drop(){
12:       mark_to_drop(standard_metadata);
13: }
14:
15: table forwarding {
16:       key = {
17:           hdr.ethernet.dstAddr : exact;
18:        }
19:       actions = {
20:           forward;
21:           drop;
22:           NoAction;
23:       }
24:       size = 32;
25:       default_action = drop();
26:       counters = my_direct_counter;
27: }
28: apply {
29:       if(hdr.ipv4.isValid()){
30:          forwarding.apply();
31:       }
32: }
```

forwarding

Key	Action	Action Data
10.0.0.1	forward	egress port = 1
10.0.0.2	forward	egress port = 2
10.0.0.3	forward	egress port = 3
10.0.0.4	forward	egress port = 4
10.0.0.5	forward	egress port = 0
10.0.0.6	forward	egress port = 0
10.0.0.7	forward	egress port = 0
10.0.0.8	forward	egress port = 0
⋮	⋮	⋮
10.0.0.32	drop	egress port = 0

my_direct_counter

Idx.	Count
0	0
1	728
2	239
3	520
4	837
5	1175
6	0
7	865
⋮	⋮
31	355

P4 Indirect counters

- Indirect counters are independent counters that can be referred to specific entries or group of entries in a match-action table

```
1: control MyIngress(inout header hdr,
2:                   inout metadata meta,
3:                   inout standard_metadata_t standard_metadata){
3:
4: counter(3,counterType.packets) my_indirect_counter;
5:
6: action forward(egressSpect_t port, bit<32> index){
7:       standard_meadata.egress_spec = port;
8:       my_indirect_counter.count(index);
9: }
10: action drop(){
11:       mark_to_drop(standard_metadata);
12: }
13:
14: table forwarding {
15:       key = {
16:           hdr.ethernet.dstAddr : exact;
17:        }
18:       actions = {
19:           forward;
20:           drop;
21:           NoAction;
22:       }
23:       size = 32;
24:       default_action = drop();
25: }
26:
27: apply {
28:       if(hdr.ipv4.isValid()){
29:          forwarding.apply();
30:       }
31: }
```

forwarding

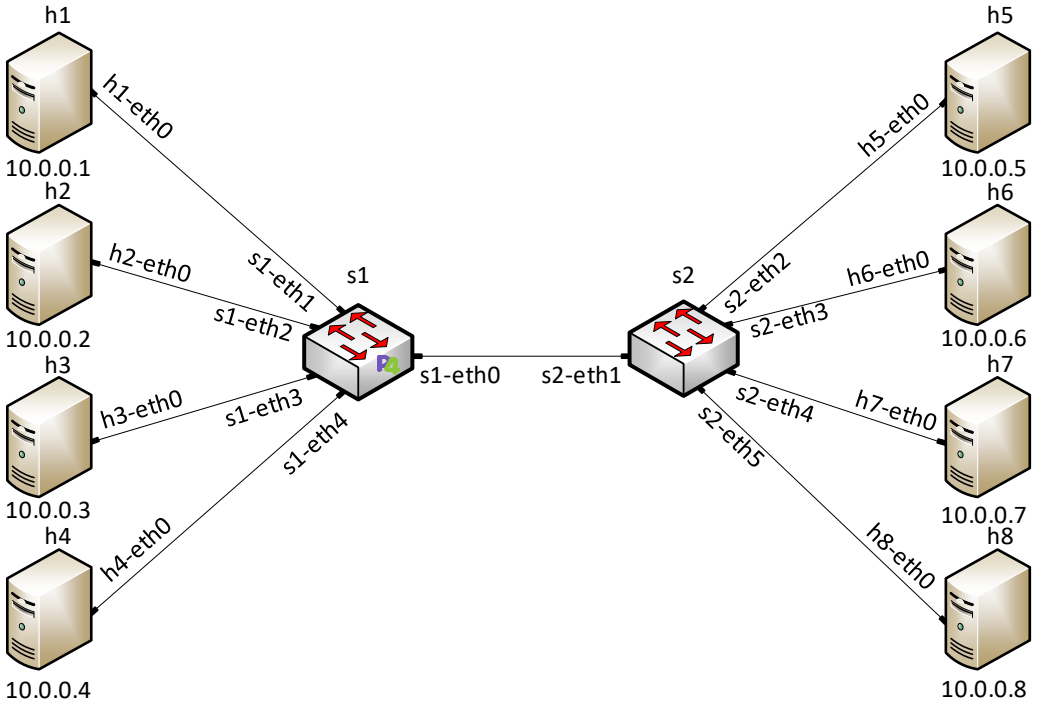
Key	Action	Action Data
10.0.0.1	forward	egress port = 1
10.0.0.2	forward	egress port = 2
10.0.0.3	forward	egress port = 3
10.0.0.4	forward	egress port = 4
10.0.0.5	forward	egress port = 0
10.0.0.6	forward	egress port = 0
10.0.0.7	forward	egress port = 0
10.0.0.8	forward	egress port = 0
⋮	⋮	⋮
10.0.0.32	drop	egress port = 0

my_indirect_counter

Idx.	Count
0	2341
1	1023
2	127

Lab Topology and Objectives

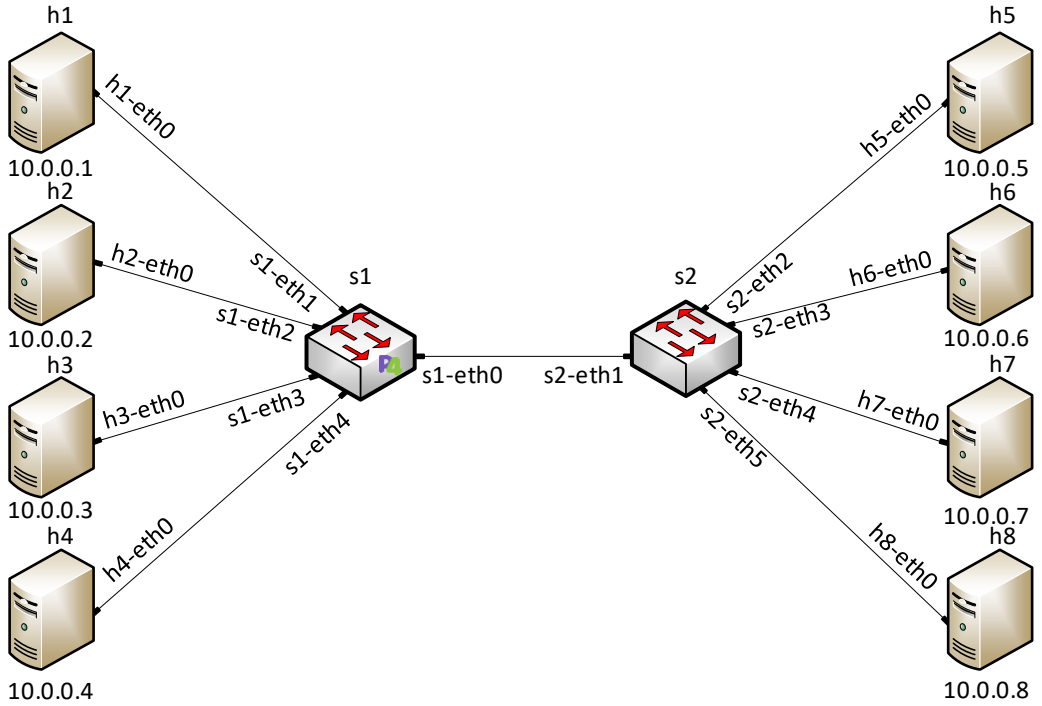
- The topology consists of eight hosts: h1-h8; one P4 switch: s1; and one legacy switch: s2
- The P4 program contains counters to count the bytes per flow
- The user will generate traffic from different hosts and read the counters from the control plane to monitor the statistics of the flows



Lab Topology and Objectives

Iperf3 test between hosts h1 and h5

```
"Host: h1"
root@lubuntu-vm:/home/admin# iperf3 -c 10.0.0.5
Connecting to host 10.0.0.5, port 5201
[  7] local 10.0.0.1 port 36320 connected to 10.0.0.5 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  7]   0.00-1.00   sec  17.7 MBytes   148 Mbits/sec  233   22.6 KBytes
[  7]   1.00-2.00   sec  15.4 MBytes   129 Mbits/sec  192   19.8 KBytes
[  7]   2.00-3.00   sec  17.2 MBytes   144 Mbits/sec  260   21.2 KBytes
[  7]   3.00-4.00   sec  15.5 MBytes   130 Mbits/sec  236   25.5 KBytes
[  7]   4.00-5.00   sec  13.9 MBytes   117 Mbits/sec  208   21.2 KBytes
[  7]   5.00-6.00   sec  15.0 MBytes   126 Mbits/sec  233   25.5 KBytes
[  7]   6.00-7.00   sec  19.0 MBytes   160 Mbits/sec  241   22.6 KBytes
[  7]   7.00-8.00   sec  18.1 MBytes   152 Mbits/sec  219   22.6 KBytes
[  7]   8.00-9.00   sec  14.2 MBytes   119 Mbits/sec  253   22.6 KBytes
[  7]   9.00-10.00  sec  13.8 MBytes   116 Mbits/sec  250   25.5 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  7]   0.00-10.00  sec   160 MBytes   134 Mbits/sec  2325             sender
[  7]   0.00-10.00  sec   160 MBytes   134 Mbits/sec                   receiver

iperf Done.
root@lubuntu-vm:/home/admin#
```



Reading the direct counter associated to the destination IP address 10.0.0.5

```
root@s1: /behavioral-model
RuntimeCmd: counter_read MyIngress.my_direct_counter 4
this is the direct counter for table MyIngress.forwarding
MyIngress.my_direct_counter[4]= (174974860 bytes, 115591 packets)
RuntimeCmd:
```

Lab Topology and Objectives

Reading the direct counter associated to the destination IP address 10.0.0.6 before the Iperf3 test

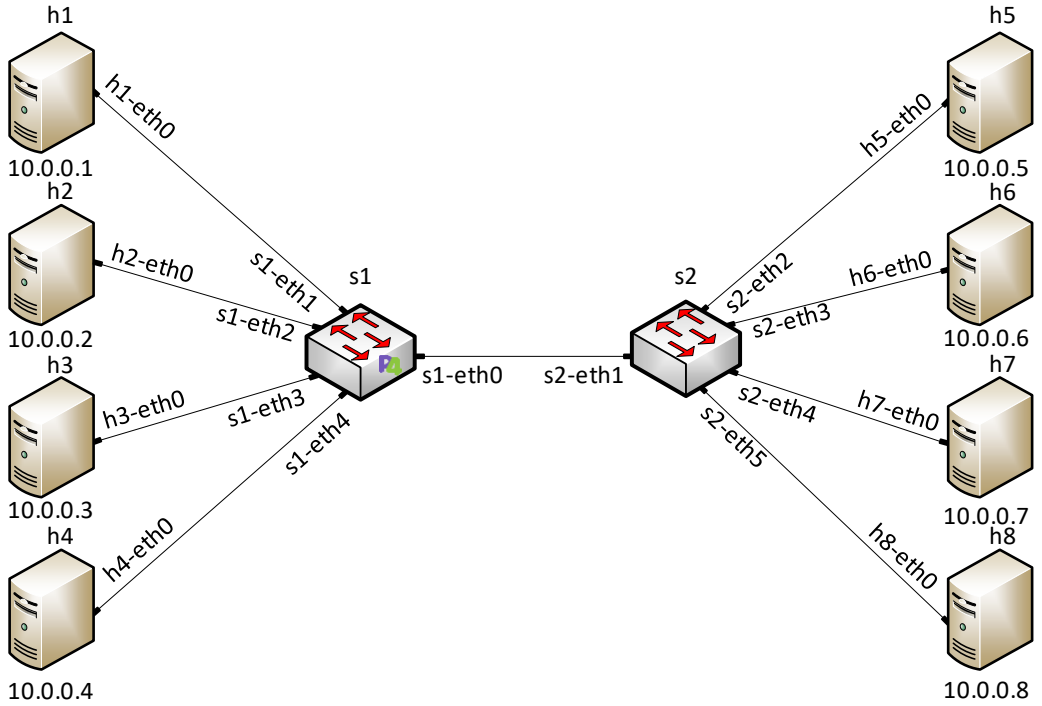
```
root@s1: /behavioral-model
RuntimeCmd: counter read MyIngress.my_direct_counter 5
this is the direct counter for table MyIngress.forwarding
MyIngress.my_direct_counter[5]= (0 bytes, 0 packets)
RuntimeCmd:
```

Iperf3 test between hosts h2 and h6

```
"Host: h2"
root@lubuntu-vm:/home/admin# iperf3 -c 10.0.0.6
Connecting to host 10.0.0.6, port 5201
[  7] local 10.0.0.2 port 43070 connected to 10.0.0.6 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  7]   0.00-1.00   sec  17.7 MBytes   149 Mbits/sec  242   21.2 KBytes

[  7]   1.00-2.00   sec  20.3 MBytes   170 Mbits/sec  233   19.8 KBytes

[  7]   2.00-3.00   sec  20.2 MBytes   169 Mbits/sec  235   24.0 KBytes

[  7]   3.00-4.00   sec  20.3 MBytes   170 Mbits/sec  240   19.8 KBytes
```

Reading the direct counter associated to the destination IP address 10.0.0.5 after the Iperf3 test

```
root@s1: /behavioral-model
RuntimeCmd: counter read MyIngress.my_direct_counter 5
this is the direct counter for table MyIngress.forwarding
MyIngress.my_direct_counter[5]= (218499328 bytes, 144339 packets)
RuntimeCmd:
```



Lab Topology and Objectives

Referring the index 1 of the indirect counter to the flow with destination IP address 10.0.0.7

```
root@s1: /behavioral-model
RuntimeCmd: table_modify MyIngress.forwarding MyIngress.forward 6 0 1
Modifying entry 6 for exact match table MyIngress.forwarding
RuntimeCmd:
```

Iperf3 test between hosts h3 and h7

```
"Host: h3"
root@lubuntu-vm:/home/admin# iperf3 -c 10.0.0.7
Connecting to host 10.0.0.7, port 5201
[  7] local 10.0.0.3 port 38988 connected to 10.0.0.7 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  7]   0.00-1.00   sec  14.1 MBytes   118 Mbits/sec  165   21.2 KBytes
[  7]   1.00-2.00   sec  16.3 MBytes   137 Mbits/sec  212   18.4 KBytes
[  7]   2.00-3.00   sec  16.3 MBytes   137 Mbits/sec  233   24.0 KBytes
```

Reading the indirect counter associated to the destination IP address 10.0.0.7

```
root@s1: /behavioral-model
RuntimeCmd: counter_read MyIngress.my_indirect_counter 1
MyIngress.my_indirect_counter[1]= (172983947 bytes, 114276 packets)
RuntimeCmd:
```