HP Networking OpenFlow Workshop

DPCTL

- dpctl: It is a command-line utility that sends basic OpenFlow messages to a switch
  - View switch port and flow statistics
  - View flow entries (FlowMods)
  - Add and delete FlowMods
- Useful tool for learning and debugging
- dpctl communicates directly with the switch and does not need a controller
- Configuration
  - Switch is the server—listening port must be configured on the switch
  - Default port for HP OF firmware is 975

For more information on DPCTL you can visit **www.openflow.org**

Dpctl command list

show SWITCH	show basic information
status SWITCH [KEY]	report statistics (about KEY) (Not on HP)
show-protostat SWITCH	report protocol statistics (Not on HP)
dump-desc SWITCH	print switch description
dump-tables SWITCH	print table stats
mod-port SWITCH IFACE ACT	modify port behavior
dump-ports SWITCH [PORT]	print port statistics
desc SWITCH STRING	set switch description
dump-flows SWITCH	print all flow entries
dump-flows SWITCH FLOW	print matching FLOWs
dump-aggregate SWITCH	print aggregate flow statistics
dump-aggregate SWITCH FLOW	print aggregate stats for FLOWs
add-flow SWITCH FLOW	add flow described by FLOW
add-flows SWITCH FILE	add flows from FILE
mod-flows SWITCH FLOW	modify actions of matching FLOWs
del-flows SWITCH [FLOW]	delete matching FLOWs
monitor SWITCH	print packets received from SWITCH
execute SWITCH CMD [ARG...]	execute CMD with ARGS on SWITCH

Dpctl example usage

Flow fields and syntax:

- nw_tos=tos/dscp
- tp_dst=port
- icmp_type=type
- icmp_code=code

The following shorthand notations are also available:

- ip Same as dl_type=0x0800
- icmp Same as dl_type=0x0800,nw_proto=1
- tcp Same as dl_type=0x0800,nw_proto=6
- udp Same as dl_type=0x0800,nw_proto=17
- arp Same as dl_type=0x0806

Dpctl example usage

Flow fields and syntax:

- in_port=port_no
- dl_vlan=vlanID
- dl_src=mac
- dl_dst=mac
- dl_type=ethertype (EtherType see: http://en.wikipedia.org/wiki/EtherType)
- nw_src=ip[/netmask]
- nw_dst=ip[/netmask]
- nw_proto=proto

Dpctl example usage

What could be done with static flow rules?

- Using nw_src – Load balancing flows from different sources to webservers
- Using dl_src – sort wireless from wired clients by looking for mac-address prefixes coming from wireless vendors
- Using In_port + Out_port – create basic virtual circuits between end points

Dpctl example usage

dpctl show tcp:10.1.1.5:975

```
features_reply (xid=0x21355842): ver:0x1, dpid:a2c27d7772d80
n_tables:2, n_buffers:256
features: capabilities:0x87, actions:0x7ff
 23(23): addr:2c:27:d7:77:2d:a9, config: 0, state:0
     current:    1GB-FD AUTO_NEG
     supported:  10MB-HD 10MB-FD 100MB-HD 100MB-FD 1GB-FD AUTO_NEG
 24(24): addr:2c:27:d7:77:2d:a8, config: 0, state:0
     current:    1GB-FD AUTO_NEG
     supported:  10MB-HD 10MB-FD 100MB-HD 100MB-FD 1GB-FD AUTO_NEG
LOCAL(local): addr:2c:27:d7:77:2d:80, config: 0, state:0
get_config_reply (xid=0xec6a5f73): miss_send_len=0
```

DPID: Unique identifier assigned by the switch for this OpenFlow instance

Number of tables and buffer size

Bitmaps of capabilities and actions (see spec)

Port Information

Capabilities

Capabilities

- Flow Stats 0x01
- Table Stats 0x02
- Port Stats 0x04
- STP 0x08
- Reserved
- Can reassemble IP 0x20
- Queue Stats 0x40
- ARP Match 0x80

Dpctl example usage

$ dpctl dump-flows tcp:15.255.124.107:6633

- Gives us information about the flows installed
- Rule itself
- Timeouts
- Actions
- Packets and bytes processed by flow

$ dpctl dump-ports tcp:15.255.124.107:6633

- Gives physical port information
- Rx, Tx counters
- Error counters

Dpctl example usage

$ dpctl **mod-port** tcp:15.255.124.107:6633 17 down

Allows manipulation of the switch ports

- Up
- Down
- Flood
- Noflood

$ dpctl mod-port tcp:15.255.124.107:6633 2 down

Ping should fail now

$ dpctl mod-port tcp:15.255.124.107:6633 2 up

Ping works again

Dpctl example usage

Let us add some flow entries so we can ping from host1 to host2

Test to ping Host 1 from Host 2 (should fail as we do not have any flow entries yet)

Add the flow entries (change port numbers):

$ dpctl add-flow tcp:15.255.124.107:6633 in_port=10,actions=output:14

$ dpctl add-flow tcp:15.255.124.107:6633 in_port=14,actions=output:10

Ping should work now!

Dpctl example usage

Let us add some IP based flow entries to test example of ping we did before

Test to ping Host 1 (IP addr = 10.10.10.1) from Host 2 (IP addr = 10.10.10.2)(should fail as we do not have any flow entries yet)

Add the IP flow entries:

$ dpctl add-flow tcp:15.255.124.107:6633 ip,nw_dst=10.10.10.1,actions=output:2

$ dpctl add-flow tcp:15.255.124.107:6633 ip,nw_dst=10.10.10.2,actions=output:17

Try pinging!

Dpctl example usage

Now let us add some ARP flow entry

$ dpctl add-flow tcp:15.255.124.107:6633 arp,actions=NORMAL

Now try pinging Host1 to Host2

Ping should be successful now!

Dpctl example usage

Following flow entries should now be shown:

$ dpctl dump-flows tcp:15.255.124.107:6634

- stats_reply (xid=0xd7d42712): flags=none type=1(flow)
- cookie=0, duration_sec=21s, duration_nsec=0s, table_id=2, priority=32768, n_packets=0, n_bytes=0, idle_timeout=60, hard_timeout=0, arp, actions=NORMAL
- cookie=0, duration_sec=7s, duration_nsec=36000000s, table_id=0, priority=32768, n_packets=0, n_bytes=0, idle_timeout=60, hard_timeout=0,ip,nw_dst=10.10.10.1,actions=output:2
- cookie=0, duration_sec=3s, duration_nsec=954000000s, table_id=0, priority=32768, n_packets=0, n_bytes=0, idle_timeout=60, hard_timeout=0,ip,nw_dst=10.10.10.2,actions=output:17

Dpctl example usage

$ dpctl dump-ports tcp:15.255.124.107:6633

- stats_reply (xid=0xb2eeb981): flags=none type=4(port)
- 3 ports
- port 2: rx pkts=2756, bytes=527428, drop=0, errs=0, frame=?, over=?, crc=? tx pkts=2721, bytes=523911, drop=0, errs=0, coll=?
- port 17: rx pkts=2733, bytes=525187, drop=0, errs=0, frame=?, over=?, crc=? tx pkts=2727, bytes=525296, drop=0, errs=0, coll=?
- port 65534: rx pkts=?, bytes=?, drop=?, errs=?, frame=?, over=?, crc=? tx pkts=?, bytes=?, drop=?, errs=?, coll=?

Dpctl example usage

Lets change the priority of flow

$ dpctl add-flow tcp:15.255.124.107:6633 ip,nw_dst=10.10.10.1,priority=1,actions=output:2

$ dpctl add-flow tcp:15.255.124.107:6633 ip,nw_dst=10.10.10.2,priority=2,actions=output:17

Lets see the flows in the switch

$ dpctl dump-flows tcp:15.255.124.107:6634

stats_reply (xid=0x8422afe4): flags=none type=1(flow)

cookie=0, duration_sec=3s, duration_nsec=899000000s, table_id=0, priority=1, n_packets=0, n_bytes=0, idle_timeout=60,hard_timeout=0,ip,nw_dst=10.10.10.1,actions=output:2

cookie=0, duration_sec=16s, duration_nsec=882000000s, table_id=0, priority=2, n_packets=0, n_bytes=0, idle_timeout=60,hard_timeout=0,ip,nw_dst=10.10.10.2,actions=output:17

Priority changed!

Dpctl example usage

dpctl benchmark tcp:15.255.124.107:6633 100 100

- Sending 100 packets * 108 bytes (with header) = 10800 bytes total
- Finished in 9.7 ms (10349 packets/s) (1117665 bytes/s)

dpctl benchmark tcp:15.255.124.107:6633 1000 1000

- Sending 1000 packets * 1008 bytes (with header) = 1008000 bytes total
- Finished in 94.4 ms (10594 packets/s) (10678984 bytes/s)

THANK YOU