dpctl

Section: OpenFlow Manual (8)
Updated: May 2008
Index Return to Main Contents

NAME

dpctl - administer OpenFlow datapaths

SYNOPSIS

dpctl [*options*] *command* [*switch*] [*args&...*]

DESCRIPTION

The **dpctl** program is a command line tool for monitoring and administering OpenFlow datapaths. It is able to show the current state of a datapath, including features, configuration, and tables entries. When using the OpenFlow kernel module, **dpctl** is used to add, delete, modify, and monitor datapaths.

Most **dpctl** commands take an argument that specifies the method for connecting to an OpenFlow switch. The following connection methods are supported:

nl:*dp_idx*
The local Netlink datapath numbered *dp_idx*. This form requires that the local host has the OpenFlow kernel module for Linux loaded.

ssl:*host*[**:***port*]
The specified SSL *port* (default: 6633) on the given remote *host*. The **--private-key**, **--certificate**, and **--ca-cert** options are mandatory when this form is used.

tcp:*host*[**:***port*]
The specified TCP *port* (default: 6633) on the given remote *host*.

unix:*file*
The Unix domain server socket named *file*.

COMMANDS

With the **dpctl** program, datapaths running in the kernel can be created, deleted, and modified. A single machine may host up to 32 datapaths (numbered 0 to 31). In most situations, a machine hosts only one datapath.

A newly created datapath is not associated with any of the host's network devices thus does not process any incoming traffic. To intercept and process traffic on a given network device, the network device must be explicitly added to a datapath through the **addif** command.

The following commands manage local datapaths.

adddp nl:*dp_idx*
Creates datapath numbered *dp_idx* on the local host. This will fail if *dp_idx* is not in the range 0 to 31, or if the datapath with that number already exists on the host.

deldp nl:*dp_idx*
Deletes datapath *dp_idx* on the local host. *dp_idx* must be an existing datapath. All of a datapath's network devices must be explicitly removed before the datapath can be deleted (see **delif** command).

addif nl:*dp_idx netdev...*
Adds each *netdev* to the list of network devices datapath *dp_idx* monitors, where *dp_idx* is the ID of an existing datapath, and *netdev* is the name of one of the host's network devices, e.g. **eth0**. Once a network device has been added to a datapath, the datapath has complete ownership of the network device's traffic and the network device appears silent to the rest of the system.

delif nl:*dp_idx netdev...*
Removes each *netdev* from the list of network devices datapath *dp_idx* monitors.

get-idx *of_dev*
Prints the datapath index for OpenFlow device *of_dev*.

The following commands can be apply to OpenFlow switches regardless of the connection method.

show *switch*
Prints to the console information on datapath *switch* including information on its flow tables and ports.

status *switch* [*key*]
Prints to the console a series of key-value pairs that report the status of *switch*. If *key* is specified, only the key-value pairs whose key names begin with *key* are printed. If *key* is omitted, all key-value pairs are printed.

(In the OpenFlow reference implementation, the **status** command is implemented in **ofprotocol**(8), not in the kernel module, so the **nl:***dp_idx* connection method should not be used with this command. Instead, specify **-l** or **--listen** on the **ofprotocol** command line and tell **dpctl** to use the connection method specified there.)

show-protostat *switch*
Prints to the OpenFlow protocol statiscal information of *switch*.

(In the OpenFlow reference implementation, the **show-protostat** command is implemented in **ofprotocol**(8), not in the kernel module, so the **nl:***dp_idx* connection method should not be used with this command. Instead, specify **-l** or **--listen** on the **ofprotocol** command line and tell **dpctl** to use the connection method specified there.)

dump-tables *switch*
Prints to the console statistics for each of the flow tables used by datapath *switch*.

dump-ports *switch* [*port number*]
Prints to the console statistics for each interface monitored by *switch*. If port number is specified, print statistics only for the interface corresponding to port number.

mod-port *switch netdev action*
Modify characteristics of an interface monitored by *switch*. *netdev* can be referred to by its OpenFlow assigned port number or the device name, e.g. **eth0**. The *action* may be any one of the following:

up
Enables the interface. This is equivalent to ``ifconfig up'' on a Unix system.

down
Disables the interface. This is equivalent to ``ifconfig down'' on a Unix system.

flood
When a *flood* action is specified, traffic will be sent out this interface. This is the default posture for monitored ports.

noflood
When a *flood* action is specified, traffic will not be sent out this interface. This is primarily useful to prevent loops when a spanning tree protocol is not in use.

dump-flows *switch* [*flows*]
Prints to the console all flow entries in datapath *switch*'s tables that match *flows*. If *flows* is omitted, all flows except emergency flows in the datapath flows are retrieved. See **FLOW SYNTAX**, below, for the syntax of *flows*.

desc *switch string*
Sets the switch description (as returned in ofp_desc_stats) to string (max length is DESC_STR_LEN).

dump-aggregate *switch* [*flows*]
Prints to the console aggregate statistics for flows in datapath WITCH's tables that match *flows*. If *flows* is omitted, the statistics are aggregated across all flows in the datapath's flow tables. See **FLOW SYNTAX**, below, for the syntax of *flows*.

add-flow *switch flow*
Add the flow entry as described by *flow* to the datapath *switch*'s tables. The flow entry is in the format described in **FLOW SYNTAX**, below.

add-flows *switch file*
Add flow entries as described in *file* to the datapath *switch*'s tables. Each line in *file* is a flow entry in the format described in **FLOW SYNTAX**, below.

mod-flows *switch flow*
Modify the actions in entries from the datapath *switch*'s tables that match *flow*. When invoked with the **--strict** option, wildcards are not treated as active for matching purposes. See **FLOW SYNTAX**, below, for the syntax of *flows*.

del-flows *switch* [*flow*]
Deletes entries from the datapath *switch*'s tables that match *flow*. When invoked with the **--strict** option, wildcards are not treated as active for matching purposes. If *flow* is omitted and the **--strict** option is not used, all flows in the datapath's tables are removed. See **FLOW SYNTAX**, below, for the syntax of *flows*.

monitor *switch*
Connects to *switch* and prints to the console all OpenFlow messages received. Usually, *switch* should specify a connection named on **ofprotocol**(8)'s **-m** or **--monitor** command line option, in which the messages printed will be all those sent or received by **ofprotocol** to or from the kernel datapath module. A *switch* of the form **nl:***dp_idx* will print all asynchronously generated OpenFlow messages (such as packet-in messages), but it will not print any messages sent to the kernel by **ofprotocol** and other processes, nor will it print replies sent by the kernel in response to those messages.

The following commands monitor and control the egress queue configuration for an OpenFlow switch if the switch supports such operations. After a queue is created with the add or modify operation, the OpenFlow enqueue action may be specified to direct packets to a particular queue. Queues are associated with specific ports (so the same queue-id may be used on different ports and this will refer to different queues). The only characteristic that may be configured for queues is the minimum bandwidth guarantee. This parameter is specified in tenths of a percent (so full link bandwidth is 1000).

add-queue *switch port q-id* [*bandwidth*]

Connect to *switch* and add an egress queue identified as *q-id* for *port*. If specified, *bandwidth* indicates the minimum bandwidth guarantee for the queue and is specified in tenths of a percent. This is the only characteristic of the queue that may be configured.

mod-queue *switch port q-id bandwidth*
Connect to *switch* and modify the bandwidth setting for an egress queue identified as *q-id* for *port*. The queue need not have been created with **add-queue** previously. The parameter *bandwidth* indicates the minimum bandwidth guarantee for the queue and is specified in tenths of a percent. This is the only characteristic of the queue that may be configured.

del-queue *switch port q-id*
Delete an egress queue identified as *q-id* for *port* which had been created by **add-queue** or **mod-queue**.

dump-queue *switch* [*port* [*q-id*]]
Dump that current queue configuration. A port may be specified. If it is, a queue-id may also be specified.

The following commands can be used regardless of the connection method. They apply to OpenFlow switches and controllers.

probe *vconn*
Connects to *vconn* and sends a single OpenFlow echo-request packet and waits for the response. With the **-t** or **--timeout** option, this command can test whether an OpenFlow switch or controller is up and running.

ping *vconn* [*n*]
Sends a series of 10 echo request packets to *vconn* and times each reply. The echo request packets consist of an OpenFlow header plus *n* bytes (default: 64) of randomly generated payload. This measures the latency of individual requests.

benchmark *vconn n count*
Sends *count* echo request packets that each consist of an OpenFlow header plus *n* bytes of payload and waits for each response. Reports the total time required. This is a measure of the maximum bandwidth to *vconn* for round-trips of *n*-byte messages.

FLOW SYNTAX

Some **dpctl** commands accept an argument that describes a flow or flows. Such flow descriptions comprise a series *field*=*value* assignments, separated by commas or white space.

The following field assignments describe how a flow matches a packet. If any of these assignments is omitted from the flow syntax, the field is treated as a wildcard; thus, if all of them are omitted, the resulting flow matches all packets. The string ***** or **ANY** may be specified a value to explicitly mark any of these fields as a wildcard.

in_port=*port_no*
Matches physical port *port_no*. Switch ports are numbered as displayed by **dpctl show**.

dl_vlan=*vlan*
Matches IEEE 802.1q virtual LAN tag *vlan*. Specify **0xffff** as *vlan* to match packets that are not tagged with a virtual LAN; otherwise, specify a number between 0 and 4095, inclusive, as the 12-bit VLAN ID to match.

dl_src=*mac*

Matches Ethernet source address *mac*, which should be specified as 6 pairs of hexadecimal digits delimited by colons, e.g. **00:0A:E4:25:6B:B0**.

dl_dst=*mac*
Matches Ethernet destination address *mac*.

dl_type=*ethertype*
Matches Ethernet protocol type *ethertype*, which should be specified as a integer between 0 and 65535, inclusive, either in decimal or as a hexadecimal number prefixed by **0x**, e.g. **0x0806** to match ARP packets.

nw_src=*ip*[/*netmask*]
Matches IPv4 source address *ip*, which should be specified as an IP address or host name, e.g. **192.168.1.1** or **www.example.com**. The optional *netmask* allows matching only on an IPv4 address prefix. It may be specified as a dotted quad (e.g. **192.168.1.0/255.255.255.0**) or as a count of bits (e.g. **192.168.1.0/24**).

nw_dst=*ip*[/*netmask*]
Matches IPv4 destination address *ip*.

nw_proto=*proto*
Matches IP protocol type *proto*, which should be specified as a decimal number between 0 and 255, inclusive, e.g. 6 to match TCP packets.

nw_tos=*tos/dscp*
Matches ToS/DSCP (only 6-bits, not modify reserved 2-bits for future use) field of IPv4 header *tos/dscp*, which should be specified as a decimal number between 0 and 255, inclusive.

tp_src=*port*
Matches UDP or TCP source port *port*, which should be specified as a decimal number between 0 and 65535, inclusive, e.g. 80 to match packets originating from a HTTP server.

tp_dst=*port*
Matches UDP or TCP destination port *port*.

icmp_type=*type*
Matches ICMP message with *type*, which should be specified as a decimal number between 0 and 255, inclusive.

icmp_code=*code*
Matches ICMP messages with *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**.

The **add-flow** and **add-flows** commands require an additional field:

actions=target[**,***target*...]
Specifies a comma-separated list of actions to take on a packet when the flow entry matches. The *target* may be a decimal port number designating the physical port on which to output the packet, or one of the following keywords:

output:*port*
Outputs the packet on the port specified by *port*.

enqueue:*port*:*q-id*
Enqueue the packet to the queue specified by *q-id* on the port specified by *port*. See **add-queue** and related commands in this manpage above.

normal
Subjects the packet to the device's normal L2/L3 processing. (This action is not implemented by all OpenFlow switches.)

flood
Outputs the packet on all switch physical ports other than the port on which it was received and any ports on which flooding is disabled (typically, these would be ports disabled by the IEEE 802.1D spanning tree protocol).

all
Outputs the packet on all switch physical ports other than the port on which it was received.

controller:*max_len*
Sends the packet to the OpenFlow controller as a ``packet in'' message. If *max_len* is a number, then it specifies the maximum number of bytes that should be sent. If *max_len* is **ALL** or omitted, then the entire packet is sent.

local
Outputs the packet on the ``local port,'' which corresponds to the **of***n* network device (see **CONTACTING THE CONTROLLER** in **ofprotocol**(8) for information on the **of***n* network device).

mod_vlan_vid:*vlan_vid*
Modifies the VLAN id on a packet. The VLAN tag is added or modified as necessary to match the value specified. If the VLAN tag is added, a priority of zero is used (see the **mod_vlan_pcp** action to set this).

mod_vlan_pcp:*vlan_pcp*
Modifies the VLAN priority on a packet. The VLAN tag is added or modified as necessary to match the value specified. Valid values are between 0 (lowest) and 7 (highest). If the VLAN tag is added, a vid of zero is used (see the **mod_vlan_vid** action to set this).

mod_dl_dst:*dst_mac*
Modifies the destination mac address on a packet, e.g., actions=mod_dl_dst:12:34:56:78:9a:bc

mod_dl_src:*src_mac*
Modifies the source mac address on a packet, e.g., actions=mod_dl_src:12:34:56:78:9a:bc

mod_nw_tos:*tos/dscp*
Modifies the ToS/DSCP (only 6-bits, not modify reserved 2-bits for future use) field of IPv4 header on a packet.

strip_vlan

Strips the VLAN tag from a packet if it is present.

(The OpenFlow protocol supports other actions that **dpctl** does not yet expose to the user.)

The **add-flow**, **add-flows**, **del-flows**, and **del-emerg-flows** commands support an additional optional field:

priority=*value*
Sets the priority of the flow to be added or deleted to *value*, which should be a number between 0 and 65535, inclusive. If this field is not specified, it defaults to 32768.

The **add-flow** and **add-flows** commands support additional optional fields:

idle_timeout=*seconds*
Causes the flow to expire after the given number of seconds of inactivity. A value of 0 prevents a flow from expiring due to inactivity. The default is 60 seconds.

hard_timeout=*seconds*
Causes the flow to expire after the given number of seconds, regardless of activity. A value of 0 (the default) gives the flow no hard expiration deadline.

The **dump-flows**, **dump-aggregate** and **del-flows** commands support the additional optional field:

out_port=*port*
If set, a matching flow must include an output action to *port*.

add-flow, **add-flows**, **del-flows**, **dump-flows**, and **dump-aggregate** commands support the additional optional field:

table=*number*
If specified, limits the flows about which statistics are gathered to those in the table with the given *number*. Normal (non emergency) tables are numbered as shown by the **dump-tables** command.

If this field is not specified, or if *number* is given as **255**, statistics are gathered about flows from all normal (non emergency) tables and flow manipulations are applied to notmal tables.

If this field is given as **254**, statistics are gathered about flows from emergency table and flow manipulations are applied to emergency table.

OPTIONS

--strict
Uses strict matching when running flow modification commands.

-t, **--timeout**=*secs*
Limits **dpctl** runtime to approximately *secs* seconds. If the timeout expires, **dpctl** will exit with a **SIGALRM** signal.

-p, **--private-key**=*privkey.pem*
Specifies a PEM file containing the private key used as the identity for SSL connections to a switch.

-c, **--certificate**=*cert.pem*
Specifies a PEM file containing a certificate, signed by the controller's certificate authority (CA), that certifies the private key to identify a trustworthy controller.

-C, **--ca-cert**=*cacert.pem*
Specifies a PEM file containing the CA certificate used to verify that a switch is trustworthy.

-v*module*[**:***facility*[**:***level*]], **--verbose=***module*[**:***facility*[**:***level*]]

Sets the logging level for *module* in *facility* to *level*:

- *module* may be any valid module name (as displayed by the **--list** action on **vlogconf**(8)), or the special name **ANY** to set the logging levels for all modules.
- *facility* may be **syslog**, **console**, or **file** to set the levels for logging to the system log, the console, or a file respectively, or **ANY** to set the logging levels for both facilities. If it is omitted, *facility* defaults to **ANY**.

  Regardless of the log levels set for **file**, logging to a file will not take place unless **--log-file** is also specified (see below).
- *level* must be one of **emer**, **err**, **warn**, **info**, or **dbg**, designating the minimum severity of a message for it to be logged. If it is omitted, *level* defaults to **dbg**.

-v, **--verbose**
Sets the maximum logging verbosity level, equivalent to **--verbose=ANY:ANY:dbg**.

-vPATTERN:*facility***:***pattern*, **--verbose=PATTERN:***facility***:***pattern*
Sets the log pattern for *facility* to *pattern*. Refer to **vlogconf**(8) for a description of the valid syntax for *pattern*.

--log-file[**=***file*]
Enables logging to a file. If *file* is specified, then it is used as the exact name for the log file. The default log file name used if *file* is omitted is **/var/log/openflow/dpctl.log**.

-h, **--help**
Prints a brief help message to the console.

-V, **--version**
Prints version information to the console.

EXAMPLES

A typical dpctl command sequence for controlling an OpenFlow kernel module:

```
Create datapath numbered 0:

     % dpctl adddp nl:0

Add two network devices to the new datapath:

     % dpctl addif nl:0 eth0
     % dpctl addif nl:0 eth1

Monitor traffic received by the datapath (exit with control-C):

     % dpctl monitor nl:0


View the datapath's table stats after some traffic has passed through:

     % dpctl dump-tables nl:0
```

View the flow entries in the datapath:

```
% dpctl dump-flows nl:0
```

Remove network devices from the datapath when finished:

```
% dpctl delif nl:0 eth0
% dpctl delif nl:0 eth1
```

Delete the datapath:

```
% dpctl deldp nl:0
```

SEE ALSO

ofprotocol(8), **controller**(8), **ofdatapath**(8), **vlogconf**(8)

Index

This document was created by man2html, using the manual pages.
Time: 01:32:28 GMT, October 20, 2012