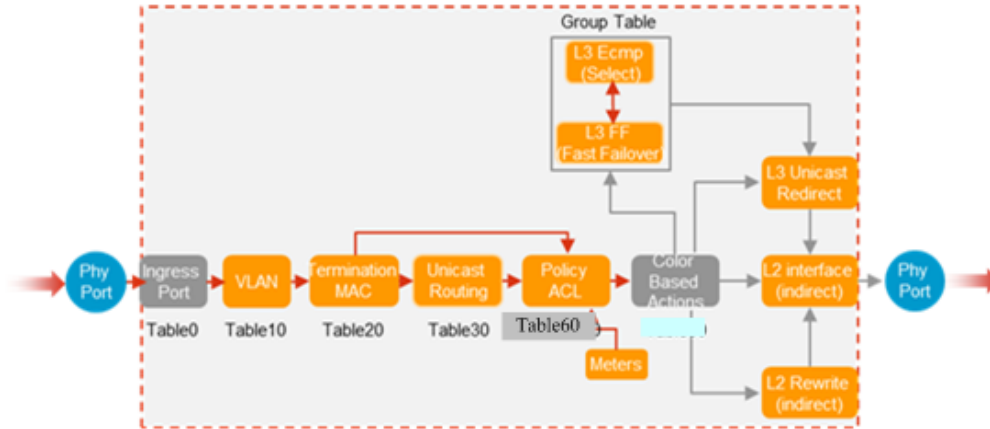


TTP Unicast

Introduction

This document describes the Table Type Pattern (TTP) and its usage.



TTP defines a template of tables that can be configured using open flow. It uses JSON notation to define the data structure for the pipeline defined in the TTP. Open flow instructions can use the parameters defined in the TTP file to program the flows. If a flow cannot be configured, the switch will report an error message.

Enabling TTP Mode

From PicOS version 2.7.1, supporting for TTP mode has been added in our switches. To use the TTP mode, user must first enable this mode. To enable the TTP mode, use the command `"ovs-vsctl set-ttp-enable true"`. To add the TTP file, use the command `"ovs-vsctl set-ttp-file <TTP file>.json"`. After adding the file, restart the switch and the TTP mode should be enabled on the switch.

json file

About .JSON FILE:

- Only "table_map" and "flow_tables" can be modified.
- "table_map": User can modify the table number of "VLAN," "Termination MAC," "Unicast Routing" and "Policy ACL". All of the four table id's must be increasing and different. For example, if user wants table 20 to be the VLAN table, write like this "VLAN:" 20 and so on.
- Flow_tables: Only the priority of all the flows tables can be changed. If user modifies vlan filter tables' priority to 1999, when user adds a flow, user must specify that the priority is 1999.
- path: /ovs/share/openvswitch

At present, users cannot modify the json file. If necessary, users may notify our team, and we may be able to change it.

Table map is:

- VLAN:10
- MAC:20
- ROUTE:30
- ACL:60

Notification

- In the acl table, if any layer 2 header content of packets is modified, previous actions in route table will be invalid.
- At present, the packets can ecmp in src ip by default.

VLAN Table

VLAN Filter Table

In this table, the user can configure the flow matching a vlan and in_port, and the action is dropped. That means that when packets arrive in this table, all the packets that are matching this flow will be dropped, others will pass through the vlan filter table. See the example configuration below:

```
ovs-ofctl add-flow br-s table=10,priority=32768,in_port=77,dl_vlan=20,actions=drop
```

VLAN Assignment Table

In this table, the user can configure a flow to decide the actions of the untagged packets. The flow matching fields are in_port, and vlan_vid=0x0000/0x1000, the action is set_field a vlan and goto terminal mac table. Refer to the example configuration below:

```
ovs-ofctl add-flow br-s table=10,priority=32768,in_port=77,vlan_vid=0x0000/0x1000,actions=set_field:123-\>vlan_vid,goto_table:20
```

VLAN Translate Single Tag Table

In this table, the user can configure a flow to decide the action of the tagged packets. The flow matching fields are in_port and a valid vlan vid, the actions is set_field a vlan and goto terminal mac table. As an example, see the code below:

```
ovs-ofctl add-flow br-s table=10,priority=32768,in_port=77,dl_vlan=10,actions=set_field:123-\>vlan_vid,goto_table:20
```

Terminal MAC Table

In this table, the user can configure a flow to decide if the packets go to the route table or not, do not support a output port in this table. By default, packets will go to the acl table directly, instead of going to route table first. The match fields of the flow supported by this table are dl_dst, dl_vlan, action is goto_table:30. Multicast and broadcast MAC are not supported here.

```
ovs-ofctl add-flow br-s table=20,priority=32768,dl_dst=00:11:22:33:44:56,dl_vlan=2015,actions=goto_table:30
```

L3 Table

In this table, users can direct the packets' route through several different kind of groups by means of writing actions. The matching fields are dl_type, nw_dst, actions are several kinds of groups and dec_nw_ttl, then go to ACL table. According our .json file we can add this kind of flow. Broadcast and multicast are not supported in this table. By default, all packets pass through this table and goto acl table.

Examples

(1) Unicast

L2 interface group:

```
ovs-ofctl add-group br-s group_id=1,type=indirect,bucket=output:79
```

L3 unicast group:

```
ovs-ofctl add-group br-s group_id=2,type=indirect,bucket=set_field:66:66:66:00:00:00-\>dl_src,set_field:66:66:66:11:11:11-\>dl_dst,set_field:200-\>vlan_vid,group:1
```

```
ovs-ofctl add-flow br-s table=30,priority=32768,dl_type=0x0800,nw_dst=192.168.1.100,actions=write_actions\ (group:2,dec_nw_ttl)\,goto_table:60
```

(2) Ecmp

User can also complete the ecmp through a select group.

L2 interface group:

```
ovs-ofctl add-group br-s group_id=1,type=indirect,bucket=output:77
ovs-ofctl add-group br-s group_id=5,type=indirect,bucket=output:79
```

L3 UNICAST group:

```
ovs-ofctl add-group br-s group_id=2,type=indirect,bucket=set_field:66:66:66:00:00:00->dl_src,set_field:66:66:66:11:11:11->dl_dst,set_field:200->vlan_vid,group:1
ovs-ofctl add-group br-s group_id=6,type=indirect,bucket=set_field:22:11:11:11:11:11->dl_src,set_field:44:44:44:11:11:11->dl_dst,set_field:200->vlan_vid,group:5
```

L3 ff group:

```
ovs-ofctl add-group br-s group_id=3,type=fast_failover,bucket=group:2,watch_port:77
ovs-ofctl add-group br-s group_id=7,type=fast_failover,bucket=group:6,watch_port:79
```

L3 ecmp:

```
ovs-ofctl add-group br-s group_id=4,type=select,bucket=group:2,bucket=group:7
```

Add flow and check:

```
ovs-ofctl add-flow br-s table=30,priority=32768,dl_type=0x0800,nw_dst=192.168.1.100,actions=write_actions\
(group:4,dec_nw_ttl),goto_table:60
admin@PicOS-OVS$ovs-ofctl dump-flows br-s
OFPST_FLOW reply (OF1.4) (xid=0x2):
 cookie=0x0, duration=76.544s, table=30, n_packets=n/a, n_bytes=n/a, ip,nw_dst=192.168.1.100
 actions=write_actions(group:9,dec_ttl),goto_table:60
admin@PicOS-OVS$ovs-appctl pica/dump-flows
Ingress Port Table: (Total 0 flows)

VLAN Table: (Total 0 flows)

Termination MAC Table: (Total 0 flows)

Unicast Routing Table: (Total 1 flows)
 ID=1 ip,nw_dst=192.168.1.100, actions:group(id=9,select,n=2,b0(live,group(id=3,ff,n=1,b0(live,group(id=2,
indirect,n=1,b0(live,set(dl_src=66:66:66:00:00:00,dl_dst=66:66:66:11:11:11),set(vlan_vid=200),group(id=1,
indirect,n=1,b0(live,output:77)))))),b1(live,group(id=7,ff,n=1,b0(live,group(id=6,indirect,n=1,b0(live,set
(dl_src=22:11:11:11:11:11,dl_dst=44:44:44:11:11:11),set(vlan_vid=200),group(id=5,indirect,n=1,b0(live,output:
79)))))),goto(Policy ACL Table)

Policy ACL Table: (Total 0 flows)
```

Policy ACL Table

In this table, users can add a flow with any match field supported by our switch. The actions could be meter, group, controller, drop, set_queue.

Example:

```
ovs-ofctl add-flow br-s table=60,priority=65535,in_port=78,dl_type=0x86dd,dl_src=22:22:22:22:22:22,dl_dst=44:44:44:11:11:11,ipv6_src=2015::1,dl_vlan=100,dl_vlan_pcp=3,ip_proto=6,ip_dscp=128,ip_ecn=0,tp_src=1111,tp_dst=1444,
actions=meter:1
```

```
ovs-ofctl add-flow br-s table=60,actions=write_actions\ (group:4\)
```

Check TTP flows

The "ovs-appctl pica/dump-flows [tables]" accept table numbers or strings from version 2.8.0.

1) Ingress Port Table:

```
root@PicOS-OVS$ ovs-appctl pica/dump-flows 0
Ingress Port Table: (Total 0 flows)
root@PicOS-OVS$ ovs-appctl pica/dump-flows port
Ingress Port Table: (Total 0 flows)
```

2) VLAN Table:

```
admin@PicOS-OVS$ ovs-appctl pica/dump-flows 10
VLAN Table: (Total 2 flows)
flow_id=6 in_port=1,dl_vlan=20, actions:drop
flow_id=7 in_port=1,dl_vlan=21, actions:drop
admin@PicOS-OVS$ ovs-appctl pica/dump-flows vlan
VLAN Table: (Total 2 flows)
flow_id=6 in_port=1,dl_vlan=20, actions:drop
flow_id=7 in_port=1,dl_vlan=21, actions:drop
```

3) Termination MAC Table:

```
admin@PicOS-OVS$ ovs-appctl pica/dump-flows 20
Termination MAC Table: (Total 3 flows)
flow_id=6 dl_vlan=2015,dl_dst=00:11:22:33:44:56, actions:goto(Unicast Routing Table)
flow_id=7 dl_vlan=205,dl_dst=22:22:22:33:44:56, actions:goto(Unicast Routing Table)
admin@PicOS-OVS$ ovs-appctl pica/dump-flows mac
Termination MAC Table: (Total 3 flows)
flow_id=6 dl_vlan=2015,dl_dst=00:11:22:33:44:56, actions:goto(Unicast Routing Table)
flow_id=7 dl_vlan=205,dl_dst=22:22:22:33:44:56, actions:goto(Unicast Routing Table)
```

4) Unicast Routing Table:

```
admin@PicOS-OVS$ ovs-appctl pica/dump-flows 30
Unicast Routing Table: (Total 1 flows)
flow_id=6 ip,nw_dst=192.168.1.0/24, actions:group(id=300,indirect,n=1,b0(live,set(dl_src=66:66:66:00:00:00,dl_dst=66:66:66:11:11:11),set(vlan_vid=200),group(id=100,indirect,n=1,b0(live,output:2))))),goto(Policy ACL Table)
admin@PicOS-OVS$ ovs-appctl pica/dump-flows unicast
Unicast Routing Table: (Total 1 flows)
flow_id=6 ip,nw_dst=192.168.1.0/24, actions:group(id=300,indirect,n=1,b0(live,set(dl_src=66:66:66:00:00:00,dl_dst=66:66:66:11:11:11),set(vlan_vid=200),group(id=100,indirect,n=1,b0(live,output:2))))),goto(Policy ACL Table)
```

5) Policy ACL Table:

```

root@PicOS-OVS$ovs-appctl pica/dump-flows 60
Policy ACL Table: (Total 1 flows)
  flow_id=4009 priority=40000,ip,dl_vlan=199, actions:meter(id=2,drop),group(id=5,indirect,n=1,b0(live,output:4))
root@PicOS-OVS$ovs-appctl pica/dump-flows acl
Policy ACL Table: (Total 1 flows)
  flow_id=4009 priority=40000,ip,dl_vlan=199, actions:meter(id=2,drop),group(id=5,indirect,n=1,b0(live,output:4))

```

List TTP System Resources Usage

From version 2.8.0, when ttp is enabled, we can use command `ovs-appctl pica/show tables` to check max limitation or current used of different tables.

```

admin@PicOS-OVS$ovs-appctl pica/show tables
TTP Tables Statistics:
Pica Tables                               Max Limitation           Current Used
-----
VLAN Table (vlan-translate)               512                       512
VLAN Table (vlan-vfilter)                 512                       512
Termination MAC Table                     512                       510
Host Unicast Routing Table                12000                     IPv4(0), IPv6(6000*2)
Route Unicast Routing Table               12000                     IPv4(4261), IPv6(3870*2)
Policy ACL Table                          2046                      2046
admin@PicOS-OVS$

```

Flow_id in TTP flows

`Flow_ids` can match for TTP based flow configuration, and you can check or delete the flow using `flow_id`.

```

admin@PicOS-OVS$ovs-ofctl add-flow br0 table=10,priority=32768,in_port=1,dl_vla
n=199,actions=set_field:123->vlan_vid,goto_table:20
admin@PicOS-OVS$
admin@PicOS-OVS$ovs-ofctl dump-flows br0
OFPST_FLOW reply (OF1.4) (xid=0x2):
  flow_id=6, cookie=0x0, duration=8.424s, table=10, n_packets=n/a, n_bytes=n/a, in_port=1,dl_vlan=199
actions=set_field:123->vlan_vid,goto_table:20
admin@PicOS-OVS$
admin@PicOS-OVS$ovs-appctl pica/dump-flows
Ingress Port Table: (Total 0 flows)

VLAN Table: (Total 1 flows)
  flow_id=6 in_port=1,dl_vlan=199, actions:set(vlan_vid=123),goto(Termination MAC Table)

Termination MAC Table: (Total 0 flows)

Unicast Routing Table: (Total 0 flows)

Policy ACL Table: (Total 0 flows)

admin@PicOS-OVS$

```

Check flows using `flow_id`:

```
admin@PicOS-OVS$ovs-ofctl dump-flows br0 flow_id=6
OFPST_FLOW reply (OF1.4) (xid=0x2):
  flow_id=6, cookie=0x0, duration=23.068s, table=10, n_packets=n/a, n_bytes=n/a, in_port=1,d1_vlan=199
actions=set_field:123->vlan_vid,goto_table:20
admin@PicOS-OVS$ovs-appctl pica/dump-flows flow_id=6
VLAN Table: (Total 1 flows)
  flow_id=6 in_port=1,d1_vlan=199, actions:set(vlan_vid=123),goto(Termination MAC Table)
admin@PicOS-OVS$
```

Delete flows using flow_id:

```
admin@PicOS-OVS$ovs-ofctl del-flows br0 flow_id=6
admin@PicOS-OVS$
admin@PicOS-OVS$ovs-ofctl dump-flows br0
OFPST_FLOW reply (OF1.4) (xid=0x2):
admin@PicOS-OVS$
```