Optimizing TCAM Usage

1, extend-group

TCAM flow table support add arp/mpls and other normal flows. Due to ASIC limitation these flows cannot use one same slice, so we use group to devide these flows. In other words, arp and mpls flows use different groups from normal flow, we said them extend groups.

Before PicOS2.8.1, user can add these different groups flow at same time. From PicOS2.8.1, user also can add these flows at same time, but also can control disable or enable extend group for arp/mpls flows. The command as below,

ovs-vsctl disable-extend-group <TRUE|FALSE>

Once disable it, user will not allowed to add arp flow which include arp_tpa or arp_spa, and not allowed to add mpls flow include mpls_label. But support add dl_type=0x0806 and dl_type=0x8847/0x8848.

For example,

```
 admin@PicOS-OVS$ovs-vsctl disable-extend-group TRUE
  Please reboot for the change to take effect!
admin@PicOS-OVS$sudo /etc/init.d/picos restart
[....] Stopping web server: lighttpd.
[....] Stopping: PicOS Open vSwitch/OpenFlowdevice ovs-pica8 left promiscuous mode
device br0 left promiscuous mode
.
[....] Stopping enhanced syslogd: rsyslogd.
[....] Starting enhanced syslogd: rsyslogd.
[....] Stopping internet superserver: xinetd.
[....] Restarting OpenBSD Secure Shell server: sshd.
[....] Starting: PicOS Open vSwitch/OpenFlow.
admin@PicOS-OVS$ovs-pica8 entered promiscuous mode
device br0 entered promiscuous mode
admin@PicOS-OVS$
admin@PicOS-OVS$ovs-vsctl show-extend-group
extend group: disabled
admin@PicOS-OVS$
```

2, match-mode

By default, 2 TCAM entries are used to support all matching tuples for all flows even if the flow does not use all matching tuples. All the groups below can be used in combination.

PicOS allows users to configure the switch in short flow TCAM match mode to optimize the TCAM usage, in this mode, each flow will only consume 1 TCAM entry (doubling the flows capacity in the TCAM).

When this mode is enabled (with the set-match-mode command), only specific fields can be used in the priority range defined by the command.Use " ovs-appctl pica /show tables" to check the max flows numbers can be installed to switch .

From version 2.11,pica8 switch support thirteen match mode groups.

The flows must use the exact fields described below:

mac(in_port: INGRESS_PORTS+INGRESS_LAG**):** ***"in_port,dl_src,dl_dst,dl_vlan,dl_type"***

mac_x(in_port: INGRESS_PORTS)**:** ***"in_port,dl_src,dl_dst,dl_vlan,dl_type"***

ip(in_port: INGRESS_PORTS+INGRESS_LAG)***: "in_port,nw_proto,nw_src,nw_dst,tp_src,tp_dst,dl_type=0x0800"***

ip_x(in_port: INGRESS_PORTS): ***"in_port,nw_proto,nw_src,nw_dst,tp_src,tp_dst,dl_type=0x0800"***

ipv4_quintuple(in_port: INGRESS_PORTS, dl_type: NONE): ***"in_port,dl_vlan,nw_src,nw_dst,tp_src,tp_dst,nw_proto,dl_type=0x0800"***

arp_tpa: ***"in_port,arp_tpa,dl_type=0x0806"***

arp_full: ***"in_port,dl_src,dl_dst,dl_vlan,arp_spa,arp_tpa,arp_op,dl_type=0x0806"***

arp_min: ***"in_port,arp_spa,arp_tpa,arp_op,dl_type=0x0806"***

l2l4: ***"in_port,dl_src,dl_dst,dl_vlan,dl_vlan_pcp,dl_type,nw_proto,nw_src,nw_dst,nw_tos,tp_src,tp_dst"***

ipv6_full: ***"in_port,dl_vlan,ipv6_src,ipv6_dst,tp_src,tp_dst,nw_proto,dl_type=0x86dd"***

ipv6_src: ***"in_port,dl_src,dl_dst,dl_vlan,ipv6_src,nw_proto,dl_type=0x86dd"***

ipv6_dst: ***"in_port,dl_src,dl_dst,dl_vlan,ipv6_dst,nw_proto,dl_type=0x86dd"***

udf ***"in_port,udf0,udf1,udf2,udf3"***

You can use one or more fields in a match mode group.For example, if mac mode is enabled, all the flows must only use one or more fields defined in the mac mode. If mac and ip modes are enabled, then user can configure either mac flows or ip flows based on the fields described above. However, user cannot mixed the fields from mac and ip (that is, dl_src and nw_src).

Each match mode group is configured with a priority range. All the priorities can not be overlapped among all the match mode groups you configured.The max match mode groups you can configure in command "ovs-vsctl set-match-mode" must be not more than 10.

in the example below, all the flows between priority "10 and 1000" have to be Mac flows. All flows between 2000 and 20000 have to be IP flows and all flows between 30000 and 60000 have to be ARP flows, all flows between 50000 and 50001 have to be ipv6_full, all flows between 50002 and 50003 have to be ipv6_dst, all flows between 50004 and 60000 have to be ipv6_src.

1. all platforms support all the match mode groups from 2.11.
2. **all the vcap flows include l2gre encapsulation,vxlan encapsulation,push_pbb can not be supported.**
3. mac_x,ip_x,ipv4_quintuple match modes can not **match lag in port**.
4. udf mode is conflict with the arp match group.After enable udf mode,arp flow in arp match group can not be installed to flow table.
5. .ipv6 match mode group do not support vcap actions (eg:push_vlan,push_pbb)
6. The max flows of each match mode group has nothing to do with the numbers of match fields in a flow,but the width of the group ifself.

Examples

example1:arp_min group

Step1,set match mode to arp_min

```
ovs-vsctl set-match-mode arp_min=1-2000
```

Step2,add a flow

```
ovs-ofctl del-flows br0
ovs-ofctl add-flow br0 priority=1000,in_port=1,arp_spa=192.168.1.100,arp_tpa=192.168.2.100,arp_op=2,
dl_type=0x0806,actions=2
ovs-appctl pica/dump-flows
```

Step3,send packets to te-1/1/1

a.Send arp packets with arp_spa is 192.168.1.100,arp tpa is 192.168.2.100,arp op is 2 to te-1/1/1

result:te-1/1/2 will transmit the arp packets.

b.Send arp packets with arp_spa is 192.168.1.100,arp tpa is 192.168.2.100,arp op is 1 to te-1/1/1

result:te-1/1/2 will not transmit the arp packets.

example2:l2l4 group

Step1,set match mode to l2l4 group

```
ovs-vsctl set-match-mode l2l4=1000-20000
```

Step2,add a flow with all match fields

```
ovs-ofctl del-flows br0
ovs-ofctl add-flow br0 priority=10000,in_port=1,vn_tag=0x01110222,dl_src=22:11:11:11:11:11,dl_dst=22:22:22:22:
22:22,dl_vlan=10,dl_vlan_pcp=3,dl_type=0x0800,nw_proto=6,nw_src=192.168.1.100,nw_dst=192.168.2.100,nw_tos=24,
tp_src=0x1000,tp_dst=0x2000,actions=2
ovs-appctl pica/dump-flows
```

Step3,send packets to te-1/1/1

a.Send ip packets with vntag ethertype is 0x893f,vn_tag is 0x01110222,src mac is 22:11:11:11:11:11,dst mac is 22:22:22:22:22:22,dl_vlan=10,src ip is 192.168.1.100,dst ip is 192.168.2.100,vlan pcp is 3,nw_proto is 6,tos is 24,tp_src is 0x1000,tp_dst is 0x2000 to te-1/1/1.

result:te-1/1/2 will transmit the packets.

example3:udf match group

step1,set udf match mode

```
ovs-vsctl set-match-mode udf=2000-3000
```

step2,set udf mode

```
ovs-vsctl set-udf-mode "udf0(l2,offset=12,length=2)"
```

step3,add ipv4 flow with other priority except 2000-3000

```
ovs-ofctl add-flow br0 priority=500,table=250,in_port=1,udf0=0x0800,actions=14
```