Supporting conntrack timeout policy on OVS

Yi-Hung Wei
yihung.wei@gmail.com
VMWare, OVS team
Open vSwitch

- OVS is a multi-layer switch
- Visibility (NetFlow, sFlow, SPAN/RSPAN)
- Fine-grained ACLs and QoS policies
- Port bonding, LACP, tunneling
- Centralized control through OpenFlow and OVSDB
- Open source using Apache 2 license*
- Multiple ports to physical switches

http://www.openvswitch.org/
OVS Architecture

- **Controller**
- **OVSDB Server** (`ovsdb-server`)
- **OVS-Vswitchd** (`ovs-vswitchd`)

- **Kernel**
  - ovSdb protocol
  - OpenFlow
  - Netlink

- **Userspace**

Upcall when flow cache miss
OVS conntrack action example

# OpenFlow rules that allow new connection from port 0 -> port 1

table=0, in_port=0, ip actions=ct(table=1)
table=0, in_port=1, ip actions=ct(table=1)

table=1, in_port=0, ip, ct_state=+trk+new actions=ct(commit), output:1
table=1, in_port=0, ip, ct_state=+trk+est  actions=output:1
table=1, in_port=1, ip, ct_state=+trk+est  actions=output:0
Customized timeout policy

- Motivation
  - Default timeout is too short
    - Does not want to re-establish long hanging connections
  - Default timeout is too long
    - Want to timeout soon to reclaim resources

- Configuration by iptables
  $ nftct add timeout test-tcp inet tcp established 100 close 10 close_wait 10
  $ iptables -I PREROUTING -t raw -p tcp -j CT --timeout test-tcp
Support timeout policy in OVS

- Just extend the OpenFlow API
  - $ nfct add timeout test-tcp inet tcp established 100 close 10 close_wait 1
  - table=1, in_port=0, ip, ct_state=+trk+new actions=ct(commit,timeout=test_tcp), output:1

- Issues
  - Controller configuration is usual for a group of entities
    - A generic configuration for a set of L4 protocols (TCP, UDP, ICMP, etc..)
    - Break down the controller generic timeout policy into 2 x L4 pieces
  - OpenFlow rules explosion
    - increase the number of conntrack commit flows to # of L4 protocols times
    - ip, tcp actions=ct(commit, timeout=test_tcp)
      ip, udp actions=ct(commit, timeout=test_udp)
      ip, icmp actions=ct(commit, timeout=test_icmp)
Zone-based timeout policy design

controller

ovs-vswitchd

ovsdb-server

openvswitch.ko

nf_conntrack.ko

nf_ct_netlink_timeout.ko

netfilter

ip=xxx, ip ct(commit, zone=1)
ip=yyy, ip ct(commit, zone=2)

zone 1 -> tp-01
zone 2 -> tp-02

Bind zone id to timeout policy during upcall

Break down generic timeout policy to small pieces
Discussion

- Is zone based timeout policy support sounds useful for other netfilter use case?
- Other zone based features?