A new solution to support connection tracking

Yuan Linsi
Jaguar Microsystems
Agenda

- Community Solution: ct action
- Challenge
- Requirement and Design Goals
- Our Proposed Solution
  - Design
  - Performance
  - Impact on hwol support
- Further work
Community Solution: ct action

TX Recirc once: ct update conn state
RX Recirc twice: ct update conn state; vxlan tunnel decap

ofproto pkt trace result for tx:

```
br(int):... (default)
    priority 1
    resubmit(0)
    06: ip, in_port=1, priority 100
    load:0x16->NN_H�XRESS]
    load:0x16->NN_H�XRESS]
    if(table=(type=port), zone=0)
    mpop
    -> A clone of the packet is forked to recirculate. The forked pipeline will be resumed at table 02.
    Final flow: ip, in_port=1, vlan_tci=0x0000, dl_src=fa:c8:0a:2c, dl_dst=fa:c8:0a:2c, ct_proto=6, ct_state=NEW
    datapath actions: CT(09), recirc(80)
    recirc(80) - resume conntrack with default ct_state=NEW (use --ct-next to customize)
    Flow: recircId=80, ct_state=NEW, trk, ct_zone=0, eth, ip, vlan_tci=0x0000,
    dl_src=fa:c8:0a:2c, dl_dst=fa:c8:0a:2c, ct_proto=6, ct_state=NEW, ct_protocol=vxlan,
    nw_tos=0, nw_src=172.16.10.43, nw_dst=172.16.10.43, nw_proto=vxlan
    bridge(“br-int”)... (default)
    thaw
    Resuming from table 02
    priority 1
    resubmit(0)
    ...# CT_STATE=NEW+TRK, ip, regSmux, priority 99
    ct(commit, zone=0)
    drop
    resubmit(0)
    add, priority 1
    NORMAL
    -> no learned MAC for destination, flooding
    bridge(“br-tun”)... (default)
    priority 1
    ...
    add, priority 1
    resubmit(0)
    ...# CT_STATE=NEW, priority 99
    strip_vxlan
    set_tunnel=0x04
    output:
    -> output to native tunnel
    -> tunnel 0x04 to 172.16.10.43 via hcm
```
Community Solution: ct action

1. ct_state: -trk  
   action: ct  
   ct_state: +new+trk

2. ct_state: +new+trk  
   action: output  
   ct_state: +new+trk

- update the conn state in ct action and recirc back to match based on the new conn state
Community Solution: ct action

- The whole design is Flow-oriented, stateless
- rely on recirc to support conn track
Challenge 1: performance degradation

CPU: skylake 6148, turbo on, 3.1Ghz
ovs: 4pmd (2core,4ht), base ovs 2.9
numa: keep the vm running in the same numa node with pmd thread

Test case: vm to vm, running with netperf to generate the traffic

~50% performance degradation
Challenge 2: hard to support hardware offload

- hard to support hardware offload base ct action:
  Require extra hw resource to support multi table to guarantee the hwol performance will not degrade

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>recirc_id(0), in_port(6), eth(), ip v4()...</td>
<td>actions: <code>ct(zone=10), recirc(0xf0a)</code></td>
</tr>
</tbody>
</table>

- hw Nic

```
Parse  Key build  Match  Action
```

```
<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>ct_state(new+trk), recirc_id(0xf0a), eth(), ipv4()</td>
<td>actions: clone(tnl_push(...)), out_port(3), 4</td>
</tr>
</tbody>
</table>

| Parse  Key build  Match  Action |
|------------------------------|------------------------------|
| Pipeline                      | JMP Action                   |

| Parse  Key build  Match  Action |
|------------------------------|------------------------------|
| Pipeline                      |                             |
Requirement and Design Goals

- **Goals:**
  - To get Higher performance in software datapath
  - Simplify the hardware offload support

- **Requirement:**
  - keep control plane programmable
    - up to the control plane to decide whether or not to enable the new logic
  - modular design
    - Support fallback to the old one
New design overview

- Introduce a new action: $\text{sg}(\text{allow}|\text{deny},[\text{nat}()],[\text{alg}()])$
- Introduce a new fastpath cache: \textit{SST} (Stateful Session Table Cache)
- The whole design is Connection oriented

```
parse and extract meta data
stateful session table

\textbf{SST}

\textbf{smc/emc}

dpcls

flow rule parse \&\& xlate
dp flow rule insert and execute actions
```

```
stateless flow table

\textbf{Hit} \quad \textbf{Hit} \quad \textbf{Hit}

\textbf{Recirc}

\textbf{Recirc}

\textbf{Hit} \quad \textbf{Hit} \quad \textbf{Hit}
```
### Design of sg action

**Solution:**
- Control plane no more need to specify the conn state in the match field
- Data plane will do a secondary translate base on the conn state and get final action

#### sg police:

<table>
<thead>
<tr>
<th>Port1</th>
<th>Port2</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td></td>
</tr>
<tr>
<td>est</td>
<td>allow</td>
</tr>
<tr>
<td>new</td>
<td>deny</td>
</tr>
</tbody>
</table>

#### base ct action:
- Table 0, priority=100, ip, ct_state=-trk, action=ct(table=1)
- Table 1, in_port=1, ip, ct_state=+trk+new, action=ct(commit), 2
- Table 1, in_port=1, ip, ct_state=+trk+est, action=2
- Table 1, in_port=2, ip, ct_state=+trk+new, action=drop
- Table 1, in_port=2, ip, ct_state=+trk+est, action=1

#### base sg action:
- Table 0, in_port=1, ip, action=sg(allow), 2
- Table 0, in_port=2, ip, action=sg(deny), 1
Design of sg action

Introduce a new action: `sg(allow|deny,[nat()])`

1. new conn
   - action: `sg(allow)`
     - pass

2. new conn
   - action: `sg(deny)`
     - drop

3. conn exist
   - action: `sg(deny)`
     - pass

**Solution:**
- Control plane no more need to specify the conn state in the match field
- Data plane will do a secondary translate based on the conn state and get final action
Introduction of SST (Stateful Session Table)

Steps to insert an entry to SST:

a) DP translate in sg action
b) create new conn
c) associate the conn with the flow
d) insert conn to SST
e) insert flow to the dpcls and EMC
Introduction of SST (Stateful Session Table)

- **base statefull conn**
  - SST
    - Miss
    - Hit
    - conn:
      - state update
      - && aging
      - && statistic
      - exec associate dpflow action

- **base stateless flow**
  - dpcls
    - Miss
  - ofproto slow path

- **statefull**
- **stateless**
**Introduction of SST (Stateful Session Table)**

- **base statefull conn**
  - SST
  - SST works as L2 cache
  - avoid too much upcall to the ofproto slow path

- **base stateless flow**
  - dpcls
  - Hit flow
  - exec sg action
  - dp
  - xlate
  - new conn
  - insert new conn

- **ofproto slow path**

---

Copyright © 2021 Jaguar Microsystems. All Rights Reserved
More optimizations

Base on the new design, more optimizations can be done now:

• Unify vxlan decap dp flow
  • No more rely on recirc to finish the vxlan decap

• Merge actions
  • Since the dp flow has been unified, no more rely on recirc to support ct, we can merge the actions to simplify the action execute

• Use spin lock instead of mutex
  • Avoid context switch, and spin lock is much more lightweight

• Use hugepage based session memory pool to support session entry allocation
  • Eliminate the TLB miss events and speed up the allocation and free for each session entry

• batch execute tnl_push and tnl_pop
Performance

- **vm**
- **vhostuser**
- **br-int**
- **ct/nat**
- **br-tun**
- **vxlan encap/decap**
- **br-phy**
- **phy-nic**
- TX with ct action
- TX without ct action
- TX without sg action

**CPU:** skylake 6148, turbo on, 3.1Ghz
**ovs:** 4pmd (2core,4ht), base ovs 2.9
**numa:** keep the vm running in the same numa node with pmd thread
**Test case:** vm to vm, running with netperf to generate the traffic

**~260% performance improvement**
Impact on hwol support

hardware offload support base on sg action:

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>recirc_id(0),in_port(6),eth(),ipv4()...</td>
<td>actions:sg(),tnl_push(), output 3</td>
</tr>
</tbody>
</table>

Unify multi flow in ovs layer, no extra work need to do to offload to the HW Nic, much more easy to offload to the hw Nic.
Acknowledgement

**JVS**: Jaguar Virtual Switch

- **support running on different archs:**
  - **CPU**: x86_64 / AMD / ARM
  - **Linux distribution**: Not limited to a specific one

- **Thanks to:**
  - Wang Yao, Baidu
  - Mao Yingming, Baidu
  - Liu Feifei, Baidu

- **Any comments on this design are welcome. Feel free to contact me via email**

- **Contact:**
  - lindsay.yuan@jaguarmicro.com
THANKS FOR WATCHING

Contact: lindsay.yuan@jaguarmicro.com