OVS connection tracking for Mobile use cases

Anita Tragler, Product Manager - Networking/NFV Platform
Franck Baudin, Principal Product Manager - OpenStack NFV

November, 2017 - OVS Conference
Mobile networks deployment today/yesterday

1 VNF == N x VNFC

1 ATCA blade
== 1 VM
== 1 VNFC

Picture credits: wikipedia
vEPC Mobile traffic profile

- Majority 75% are short duration flows < 100Kbps
- Large number of simultaneous calls - 1 Million flows
- High incoming call rate of 100K - 200K connections per second (cps)
- Need for distributed firewall at the vSwitch
- Statistics for each call for Billing - call duration, bandwidth, source & destination ip
10 Gbps “real Mobile traffic” profile injection

Established user flows [1] (="conntracks")
Bidirectional ⇔ 2 X OpenFlow microflows

10 Gbps, 4 Mpps
Average frame size ~600 B

200k flows/s created
200k flows/s destroyed

First flow expiration

Last flow creation

200k flows/s

1 M flows

5s

305s

[1] one five tuple: L1/L2/L3
Ex: One DNS request: MAC/IP/IUDP
One HTTP session: MAC/IP/TCP
Traffic profile - Key Parameters

Packet size

Typically only the packet header is accessed (one cache line)...

- **Except for virtualization vhost-user/vhost-net (hypervisor on host)** since guest requires payload memcpy
- except for IPSec: segmentation/reassembly or packet ordering; not priority for vswitch
- except when we terminate a connection (SSL, TCP, UDP); not as relevant to NFV

Flows or Connections: creation/destruction of flow per second

Flow Creation: not in flow table and cache, upcall to add flow => bucket allocation

Flow Destruction: TCP FIN + timer, UDP timer, LRU recycling (flow hash table entry recycling)...

Performance depends on

- Number of flows in the flow table and
- Rate of incoming flows
What metrics to measure?

In particular NEPs (VNFs vendors)

1. Performance with the number of cores, minimum OF rules, varying packet sizes
   a. Mpps (cycles/packets)
   b. Latency
   c. Jitter

2. Performance evolution regarding the number of conntrack, IP routes, ...
   a. For various cores numbers
   b. Mpps, Latency, Jitter
Datapath performances: measurement units

Telco VMs (VNFs) typically use cycles/packet internally and Gbps/Mpps externally (marketing)

RFC 2544 permit to find the maximum packet throughput before dropping, i.e. when the target is loaded at 100%:

- X Mpps ⇔ 100% system load ⇔ 0% idle for N cores running at F GHz
  - cycles/packet = (F x 10^3 / X) / N
  - 200 cycles/packet for 10 Mpps per core at 2GHz
  - This measure is an average (bulk)
- Gbps = (“inter-frame gap and preamble equivalent bits” + “frame size”) x Mpps
  - For 64 Bytes frames (CRC included): Gbps = ((20 + 64) x 8) x Mpps
All tests developed within OPNFV VSperf project

All Measures (next slides) done with:

- OVS 2.7
- IPv4 traffic
- straight NUMA
- RFC2544, 0% acceptable loss rate, 2 mins iterations
- UDP flows, 5 Tuple change, referred as “flows” in the next slides
- DPDK testpmd in the VM, so the VM is never the bottleneck (verified)
- We use a Telco grade traffic generator (TRex, could an appliance as well), not iperf!!
Conntrack test results

Thanks to our QE team
- Christian Trautman
- Qi Jun Ding

Dev team
- Flavio Leitner
- Aaron Conole
TCP Stateful conntrack - test profile

Use TRex packet replay
Use 600B IPv4 data packets
Short calls with timeout = 5s
Scale number of connections
Conntrack test configuration

Openvswitch 2.7 and DPDK 16.11

Conntrack rule 4-Tuple - Match source IP, destination IP, src port and dst port

ovs-ofctl add-flow ovsbr0
"table=0,priority=100,ip,nw_src=10.0.0.1/12,nw_dst=20.0.0.1/12,udp,tp_src=1234,tp_dst=1234,ct_state=-trk,action=ct(table=1)"

ovs-ofctl add-flow ovsbr0 "table=1,in_port=10,ip,ct_state=+trk,action=ct(commit),20"

ovs-ofctl add-flow ovsbr0 "table=1,in_port=10,ip,ct_state=+trk,action=output:20"

ovs-ofctl add-flow ovsbr0 "table=1,in_port=20,ip,ct_state=+trk,action=output:10"

ovs-ofctl add-flow ovsbr0 "table=1,in_port=11,ip,ct_state=+trk,action=ct(commit),21"

ovs-ofctl add-flow ovsbr0 "table=1,in_port=11,ip,ct_state=+trk,action=output:21"

ovs-ofctl add-flow ovsbr0 "table=1,in_port=21,ip,ct_state=+trk,action=output:11"

ovs-ofctl add-flow ovsbr0 "table=0,priority=1,action=drop"
## OVS Conntrack - VSPERF Throughput (pps)

### OVS Conntrack - VSPERF Throughput

<table>
<thead>
<tr>
<th>OVS conntrack (pps)</th>
<th>baseline</th>
<th>Src ip</th>
<th>Src &amp; dst ip</th>
<th>4-Tuple</th>
<th>5-Tuple</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1k Flows (with EMC)</strong></td>
<td>431,064</td>
<td>237,490</td>
<td>238,244</td>
<td>228,452</td>
<td>256,320</td>
</tr>
<tr>
<td><strong>1k Flows (EMC disabled)</strong></td>
<td>321,580</td>
<td>256,320</td>
<td>230,712</td>
<td>232,218</td>
<td>269,878</td>
</tr>
<tr>
<td><strong>100k Flows (with EMC)</strong></td>
<td>216,402</td>
<td>151,626</td>
<td>152,380</td>
<td>174,222</td>
<td>180,248</td>
</tr>
<tr>
<td><strong>100K Flows (EMC disabled)</strong></td>
<td>303,359</td>
<td>172,176</td>
<td>151,626</td>
<td>230,424</td>
<td>199,830</td>
</tr>
</tbody>
</table>
OVS-DPDK Conntrack - VSperf Throughput

<table>
<thead>
<tr>
<th>IPv4 (pps)</th>
<th>baseline</th>
<th>src ip</th>
<th>Src &amp; dst ip</th>
<th>4-Tuple</th>
<th>5-Tuple</th>
</tr>
</thead>
<tbody>
<tr>
<td>1k Flows</td>
<td>7,064,494</td>
<td>4,657,578</td>
<td>4,574,882</td>
<td>3,366,854</td>
<td>3,417,136</td>
</tr>
<tr>
<td>10k Flows</td>
<td>6,815,158</td>
<td></td>
<td></td>
<td>3,151,180</td>
<td></td>
</tr>
<tr>
<td>100K Flows</td>
<td>3,913,314</td>
<td>1,928,606</td>
<td>1,820,606</td>
<td>1,630,236</td>
<td>1,597,822</td>
</tr>
</tbody>
</table>
OVS-DPDK Conntrack - VSperf Throughput

Conntrack pps | baseline | Match src ip | Match 4 Tuple |
--- | --- | --- | --- |
100K Flows (with EMC) | 3,913,314 | 1,763,214 | 1,597,822 |
100K Flows (EMC disabled) | 4,053,314 | 1,928,606 | 1,630,236 |

Userspace Conntrack no significant performance improvement with EMC disabled
OVS Kernel: Conntrack Connection Setup Rate

Connection duration 5s, test duration 300s

<table>
<thead>
<tr>
<th>TCP Connection rate (cps)</th>
<th>Steady connections after 5s</th>
</tr>
</thead>
<tbody>
<tr>
<td>5K CPS</td>
<td>25K</td>
</tr>
<tr>
<td>10K CPS</td>
<td>50K</td>
</tr>
<tr>
<td>20K CPS</td>
<td>100K</td>
</tr>
<tr>
<td>50K CPS</td>
<td>250K</td>
</tr>
</tbody>
</table>

Track open connections (number of table entries)
conntrack -C (entries) & conntrack -S (stats)

timeout setting for conntrack in kernel:
- nf_conntrack_tcp_timeout_close_wait=5
- nf_conntrack_tcp_timeout_established=5
- nf_conntrack_tcp_timeout_fin_wait=5
- nf_conntrack_tcp_timeout_last_ack=5
- nf_conntrack_tcp_timeout_max_retrans=5
- nf_conntrack_tcp_timeout_syn_recv=5
- nf_conntrack_tcp_timeout_syn_sent=5
- nf_conntrack_tcp_timeout_time_wait=5
- nf_conntrack_tcp_timeout_unacknowledged=5
- nf_conntrack_udp_timeout=5
- nf_conntrack_udp_timeout_stream=5
OVS-DPDK: Conntrack Connection Setup Rate

- Cannot set **connection timeout**; default timeout = 30s. Connections are timing out @ ~32s.
- Cannot query **conntrack table entries** (# of entries) and stats (similar to `conntrack -S -C`).
- Only support for dumping conntrack table >`ovs-appctl dpctl/dump-conntrack`.
- Max conntrack **table size restricted** to 3M entries, cannot change table size.

Connection duration 5s, test duration 300s

<table>
<thead>
<tr>
<th>TCP Connection rate (cps)</th>
<th>Steady connections after 30s</th>
</tr>
</thead>
<tbody>
<tr>
<td>50K CPS</td>
<td>1.5M connections</td>
</tr>
<tr>
<td>100K CPS</td>
<td>3M connections (Max table size)</td>
</tr>
<tr>
<td>200K CPS (goal)</td>
<td>6M connections</td>
</tr>
</tbody>
</table>
Measure Connection Rate (CPS)

<table>
<thead>
<tr>
<th>Conntrack (cps)</th>
<th>TCP w/o data</th>
<th>HTTP 600B data</th>
<th>HTTP 800B data</th>
<th>UDP 800B data</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVS (kernel)</td>
<td>45K CPS</td>
<td>16K CPS</td>
<td>15K CPS</td>
<td>106K CPS</td>
</tr>
<tr>
<td>OVS-DPDK (userspace)</td>
<td>No configurable timeout*</td>
<td>55K CPS</td>
<td>55K CPS</td>
<td>84K* CPS</td>
</tr>
</tbody>
</table>

userspace conntrack cps performance is lower than expected
In Conclusion
Performance Benchmarking Plan (OPNFV VSPerf)

64B and 9KB Jumbo PVP performance
Metric - throughput, latency
Single numa node, basic multi-queue
vlan, flat, VXLAN networks, bonding

Real traffic profile with T-Rex
Mobile traffic flows
Conntrack - scale flows
Multi-queue w/ RX queue mgmt.
Live Migration, Cross NUMA perf

More overlays (NSH, MPLS...?)
Firewall testing (dynamic rules)
Conntrack - connection rate
SNAT & DNAT rule scale
OVS Hardware Offload
BFD, ECMP, L3 VPN and eVPN

SR-IOV, Base OVS and OVS-DPDK, TestPMD as a switch performance
OVS-DPDK NFV performance ready scale with cores, multi-queue
Real world Mobile traffic flows

vRouter and vFirewall features

We are here!