WEPOQ AND OVN
TALK FLOW

- Context
  - my background
  - big picture
- Wepoq Overview
  - sample apps
- Wepoq-OVN Overview
- Next Steps
  - FQDN dns
  - logical trunk port
  - Virtual Client Cloud
MY BACKGROUND

• Worked in UI system software early career
  • First OSS project was X Window System
  • up the stack: Xt, Motif, CDE
• Worked at protocol/interop layer next
  • XML Schema, Linking, RDF, RSS 2.0
• Switched to Media system interop in 2000
  • MultiPhoto/Video (MPV) standard and toolkit
  • Windows Media interop (DRM)
  • Video packaging and distribution (Blu-ray, large scale video streaming)
• Returning to OSS an networking 😊
CLIENTLESS VS SERVERLESS

- Two worlds
- VPC (virtual private cloud)
- VCC (virtual client cloud) aspirational
OVN FUTURES

- L7+
- L2/L3
- eBFF P4
- sfc/nfc
- external endpoints
- scaling
- clientless
- serverless
OVERVIEW OF END TO END PATHS

- default:
  - ua -> uw-gw
  - uw-gw -> srv-gw
  - srv-gw -> vpc
  - vpc -> wl

- sd-wan:
  - ua -> ua-vpn
  - ua-vpn -> srv-gw
  - srv-gw -> srv
  - vpc -> wl

- wepoq:
  - ua -> ua-vpn(ovn)
  - ua-vpn(ovn) -> ovn-gw
  - ovn-gw -> ovn
  - ovn -> wl
WHAT IS WEPOQ

• End to end L3 architecture for logical application networks
• Focus on client-side cloud and making client endpoints 1st class OVN ports
• Hierarchically scoped topology which drives control plane flow (up/down)
• Support for client/server and P2P application networks
• All traffic is from known L2 endpoints, no MAC learning
• Application topology directly expressed in OVN topology and DNS naming

• Current VPC approach is client-less, Wepoq makes it client-yes!
DNS NAMING HIERARCHY

- net.wepoq.ovn
  - { namespace }
  - { app }
    - { room/service }
    - ( "cli" | "svr" )
    - endpoint ( c{n}, s{n} )

<table>
<thead>
<tr>
<th>ep</th>
<th>cli</th>
<th>room/service</th>
<th>app</th>
<th>ns.wepoq.net</th>
</tr>
</thead>
<tbody>
<tr>
<td>ctl</td>
<td>roominator</td>
<td>ns.wepoq.net</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c1</td>
<td>cli</td>
<td>roominator</td>
<td>ns.wepoq.net</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>roominator</td>
<td>ns.wepoq.net</td>
<td></td>
</tr>
<tr>
<td>c32</td>
<td>cli</td>
<td>roominator</td>
<td>ns.wepoq.net</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>roominator</td>
<td>ns.wepoq.net</td>
<td></td>
</tr>
<tr>
<td>vip</td>
<td>svr</td>
<td>roominator</td>
<td>ns.wepoq.net</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>roominator</td>
<td>ns.wepoq.net</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>room1</td>
<td>roominator</td>
<td>ns.wepoq.net</td>
</tr>
<tr>
<td></td>
<td></td>
<td>room3</td>
<td>roominator</td>
<td>ns.wepoq.net</td>
</tr>
<tr>
<td>cl</td>
<td>cli</td>
<td>room3</td>
<td>roominator</td>
<td>ns.wepoq.net</td>
</tr>
<tr>
<td></td>
<td></td>
<td>room3</td>
<td>roominator</td>
<td>ns.wepoq.net</td>
</tr>
<tr>
<td></td>
<td></td>
<td>room3</td>
<td>roominator</td>
<td>ns.wepoq.net</td>
</tr>
</tbody>
</table>
DNS EXAMPLES

- cluster controller
  - vip.svr.ctl.ovn.wepoq.net
- namespace controller
  - vip.svr.ctl.ns.wepoq.net
- app controller
  - vip.svr.ctl.roominator.ns.wepoq.net, vip.svr.ctl.ovn-vote.ns.wepoq.net
- the third room controller
  - vip.svr.ctl.room3.roominator.ns.wepoq.net
- the third client in room3
  - c3.room3.roominator.ns.wepoq.net
- the third client of the ovn-vote result invoking the service
  - c3.cli.result.ovn-vote.ns.wepoq.net ->
    - vip.svr.result.ovn-vote.ns.wepoq.net
WEPOQ-OVN (SECTION)
# Network Interface Hierarchy

<table>
<thead>
<tr>
<th>Level of Host Integration</th>
<th>PIF</th>
<th>VIF</th>
<th>CIF</th>
<th>WIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HOST HW</td>
<td>Host Kernel</td>
<td>Nested Guest Kernel</td>
<td>External host Kernel</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Physical network interface (PIF)
- Virtual network interface (VIF)
- Container network interface (CIF)
- Wepoq network interface (WIF)
**LOGICAL TRUNK PORT (SECTION)**

- ovs tunnel connect internal OVN hypervisors
- vlan trunk ports connect nested server-side hypervisors to OVN
- l4 trunk ports connect external client-side hypervisors to OVN

<table>
<thead>
<tr>
<th>src</th>
<th>dst</th>
<th>connect</th>
<th>signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>HV</td>
<td>HV</td>
<td>L2 tunnel</td>
<td>external</td>
</tr>
<tr>
<td>HV-lite</td>
<td>HV</td>
<td>VLAN trunk</td>
<td>inline</td>
</tr>
<tr>
<td>HV-ext</td>
<td>HV</td>
<td>L4 trunk</td>
<td>payload</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ovs port type</th>
<th>netdev type</th>
<th>encaps layer</th>
<th>relative to L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>tunnel</td>
<td>internal</td>
<td>L2</td>
<td>outer</td>
</tr>
<tr>
<td>CIF trunk</td>
<td>nested</td>
<td>L2.5</td>
<td>inline</td>
</tr>
<tr>
<td>wepoq (*)</td>
<td>external</td>
<td>L4</td>
<td>inner</td>
</tr>
</tbody>
</table>
TYPES OF TRUNK PORTS

• physical 802.1q port
• ipvlan/macvlan port
• logical trunk port
  • vlan container port
  • l4 client port
    • client vpn
    • client ingress router
PHYSICAL TRUNK
CIF AND L4 LOGICAL TRUNK PORT

L4 to vlan Mapping

Br-wepoq

Br-int

vlan mapping

Br-vmimport

CTR_1
CIF_1

CTR_2
CIF_2

VM_1

WH_1

WH_2

CIF AND L4 LOGICAL TRUNK PORT
• wepoq cluster controller manages a set of wepoq apps. Each chassis in the cluster runs the wepoq agent which manages br-wepoq which provides client network access

• br-wepoq creates one or more veth pairs to br-int, to serve as logical trunk ports on that chasis.

• a client wepoq port is represented in ovn-nb in the externals_ids column of the ports table
  • external_ids:wepoq-addr="mac-addr ipv4-addr {protocol/port, ... protocol/port}"
  • external_ids:wepoq-name="c1.cli.room3.roominator.ns.wepoq.net"

• trunk port has
  • external_ids:wepoq="???”
NEXT STEPS
DNS PROPOSAL

- add table to ovn_sb to maintain dns mapping (suggested by ben pfaff)
- add FQDN option in addition to naked hostname
  - enables FQDN to be used instead of ip
  - significant robustness and speed potential
LOGICAL TRUNK PORT

• Generalize container ports to not be specific to VM nesting use case
• Allow multiple methods of signaling OVN on how to handle logical trunk port
  • vlan signalling
  • L4 port signalling
• make logical trunk port an explicit type
  • logical location for metadata about nested/external guests
VIRTUAL CLIENT CLOUD

• Flesh out a complete proposal
• investigate what other capabilities might enable OVN to better support VCC
  • limited resource environments (opposite of current focus)
  • add use cases