Dual TOR use case for Single NIC servers
Dual TOR use case for Single NIC servers

Lawrence Lee, Software Engineer, Microsoft
Kamini Santhanagopalan, Product Line Manager, Broadcom
Agenda

• Background
• Mux Cable Overview
• Active/Standby Control
• Dual ToR Data Plane
• SONiC Dual ToR Overview
• Broadcom Gemini Support
• Demo
• Open Discussion
Background
Data Center Topology

Tier 3 – Regional

Tier 2 – Spine

Tier 1 – Row Leaf

Tier 0 – Top of Rack

Server Racks
Single Point Of Failure

- ToR is a SPOF for full rack of servers
  - Switches do fail
    - ~2% of switches fail in first 3 months
    - 32% due to hardware failures
    - 27% due to power failures
- Link failure w.r.t single server
- ToR upgrades are a pain point
State-of-the-Art T0 Redundancy

• MCLAG
  • Requires dual NICs, requires Inter-Switch Link to sync state
• vSwitch
  • Requires dual NICs, performance limited by CPU
• SR-IOV
  • Requires dual NICs, VMs are exposed to failures
• NIC offloading
  • Performance/feature set/implementations varies between vendors
Solution Space

ToR

NIC

Host OS

Guest OS
Mux Cable Overview
Mux Cable Hardware

- Y-cable with embedded microcontroller with active/standby configuration
- Hitless firmware upgrades and switchover
- Common control plane on each end

Y-cable assembly from Molex

**OPEN POSSIBILITIES.**
Mux Cable Operation

- Northbound traffic is broadcast to both ToRs
- Southbound traffic from active side is forwarded normally
- Southbound traffic from standby side is dropped
Active/Standby Control
ICMP Heartbeat

- Heartbeat = ICMP echo request/reply
- Continuous heartbeat from both ToRs containing UUID
ICMP Heartbeat (cont.)

- Server only replies to active heartbeat
- ToRs infer active/standby state based on response from server
Dual ToR Data Plane
Southbound Packet Flow

- Southbound traffic is only possible via active ToR
- Standby sends southbound traffic to active via L3 tunnel
Northbound Packet Flow

- Northbound traffic is broadcast to both ToRs
- Data plane traffic is dropped by the standby ToR
SONiC Dual ToR Overview
SONiC Components

- Mux container (new)
  - Manages mux hardware
  - Manages convergence in failover conditions
- Orchestration agent (updated)
  - Manages L3 tunnel between peered ToRs
  - Configures ACL rules on standby ToR
  - Interact with new SAI APIs/attributes to accomplish above
- Transceiver daemon (updated)
  - Provides common interface for vendors to implement mux cable APIs
  - Directly controls mux cable
Failure Scenarios

- Failure detection utilizes heartbeat
- ToR Down:
  - ToR A is initially active:
    - ToR becomes unhealthy
  - ToR B is initially standby:
    - Detects sustained loss of heartbeat
    - Tears down L3 tunnel for southbound traffic
    - Removes ACLs to drop northbound traffic
    - Signals mux cable to switch itself to active
    - ToR B finishes transition to active
Failure Scenarios

T1

L3 Tunnel

Active ToR

Standby ToR

Mux Cable

Server
Broadcom Dual ToR Support
MSFT + Broadcom Collaboration

- Close engagement on SAI support for Dual TOR use case
  - Support for Dual ToR features on T0 devices (Trident3 & Tomahawk2)
  - Contributed SAI APIs/Attributes to support Dual ToR

- Broadcom’s broader commitment to SONiC and SAI communities
  - Active participation in peer reviews and pull requests
  - Significant contributions (20+) to SAI specification in 2020 and 2021
  - VoQ architecture, SAI pipeline enhancements, switch scoped tunnel features (IP-in-IP/GRE/VXLAN)

- Continued partnership with Microsoft to extend SONiC to new use cases
# Broadcom Dual ToR SAI Contributions

<table>
<thead>
<tr>
<th>SAI API</th>
<th>SAI Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAI TUNNEL Create, Remove, Set/Get</td>
<td>enum (sai_tunnel_ttl_mode_t) SAI_TUNNEL_TTL_MODE_UNIFORM_MODEL</td>
<td>Support of P2P Tunnels (Extended for IPinIP as well for DualToR) with underlay ECMP support to reach destination tunnel end-point</td>
</tr>
<tr>
<td></td>
<td>enum (sai_tunnel_dscp_mode_t) SAI_TUNNEL_DSCP_MODE_UNIFORM_MODEL</td>
<td>Support of both Uniform and Pipe for the DSCP (Native to Outer IP Packet)</td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_ATTR_ENCAP_DST_IP</td>
<td>Support of user defined TTL (For Outer IP Packet) specified by application</td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_ATTR_ENCAP_TTL_MODE</td>
<td>Support of both Uniform and Pipe for the TTL (Native to Outer IP Packet)</td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_ATTR_ENCAP_TTL_VAL</td>
<td>Support of user defined TTL (Native to Outer IP Packet) specified by application</td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_ATTR_ENCAP_DSCP_MODE</td>
<td>Support of both Uniform and Pipe for the DSCP (Native to Outer IP Packet)</td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_ATTR_ENCAP_DSCP_VAL</td>
<td>Support of user defined DSCP (For Outer IP Packet) specified by application</td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_ATTR_DECAP_TTL_MODE</td>
<td>Support of both Uniform and Pipe for the TTL (Native to Outer IP Packet)</td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_ATTR_DECAP_DSCP_MODE</td>
<td>Support of both Uniform and Pipe for the DSCP (Native to Outer IP Packet)</td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_ATTR_LOOPBACK_PACKET_ACTION</td>
<td>Support of Loopback packet action on tunnel to avoid the incoming and outgoing packet on the same tunnel</td>
</tr>
<tr>
<td>SAI TUNNEL Term Table Create, Remove, Set/Get</td>
<td>enum (sai_tunnel_term_table_entry_type_t) SAI_TUNNEL_TERM_TABLE_ENTRY_TYPE_MP2P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_TERM_TABLE_ENTRY_TYPE_MP2MP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_TERM_TABLE_ENTRY_ATTR_DST_IP_MASK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_TERM_TABLE_ENTRY_ATTR_SRC_IP_MASK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_TERM_TABLE_ENTRY_ATTR_IP_ADDR_FAMILY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_ATTR_VXLAN_UDP_SPORT_MODE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SAI_TUNNEL_ATTR_VXLAN_UDP_SPORT</td>
<td></td>
</tr>
</tbody>
</table>
Call to action

Download and run SONiC!
https://azure.github.io/SONiC/

Check out the SONiC OCP page:
https://www.opencompute.org/projects/sonic
(weekly OCP call link)
Open Discussion