OPEN POSSIBILITIES.

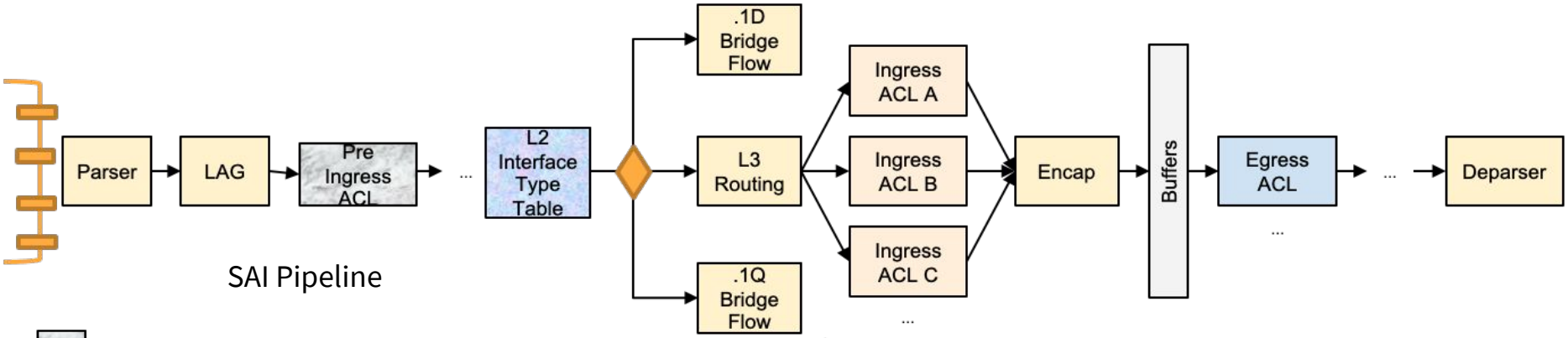
SAI Pipeline Enhancements
- Pre-Ingress ACL Stage
- MyMAC Station Stage

SAI Spec Enhancement
- FEC Modes for >= 200G Ports

SAI pipeline enhancements with Pre-Ingress ACL and MyMAC station stages and enhanced FEC Modes for 200G and above ports

Jai Kumar, Distinguished Engineer, Broadcom
Kishore Gummadidala, Software Engineer, Google
Mike Beresford, Software Engineer, Google

Agenda



SAI Pipeline

Pre Ingress ACL Block – Binding to Switch, New qualifiers and actions

Enhanced L2 Table with MYMAC Entries– Binding to Switch, New qualifiers and actions

Enhanced Port Attributes – New FEC modes

Pre-Ingress ACL Stage - Introduction

NETWORKING

- VRF is currently derived from the Router Interface
- Overriding the VRF based on a packets L2/L3/.. header fields can be useful
- For example: use L3 DSCP to override VRF, and forward high-priority traffic differently from low-priority traffic arriving on the same RIF

Pre-Ingress ACL Stage - Implementation

NETWORKING

- The VRF override should happen before L3 lookup.
- It can be achieved by a match rule in the Port bound ACL via a new Set VRF action.
- In some implementations, port bind points are achieved by adding the port as an ACL match field to the ACL rule.
- If the rules are applicable to multiple (or all) ports, then the rules may need to be instantiated per port hence leading to scaling constraints.
- An ACL bound to the switch is ideal for these rules.

Pre-Ingress ACL Stage- Proposal

NETWORKING

- https://github.com/opencomputeproject/SAI/pull/1185 (merged, included in SAI 1.8)
- Add an ACL stage SAI_ACL_STAGE_PRE_INGRESS with switch bind point SAI_SWITCH_ATTR_PRE_INGRESS_ACL
- Add new ACL action to “Set VRF” SAI_ACL_ENTRY_ATTR_ACTION_SET_VRF
- Existing ACL match fields are sufficient

Pre-Ingress ACL Stage- Example

NETWORKING

- Create a Pre-Ingress ACL table, bind it to switch.
  attr[0].id=SAI_ACL_TABLE_ATTR_ACL_STAGE;
  attr[0].value.s32=SAI_ACL_STAGE_PRE_INGRESS;
  attr[1].id=SAI_ACL_TABLE_ATTR_FIELD_IP_DSCP;
  attr[1].value.booldata = true;
- Add a rule to match on DSCP and assign a VRF
  attr[0].id=SAI_ACL_ENTRY_ATTR_FIELD_DSCP;
  attr[0].value.aclfield.data.u8=3;
  attr[0].value.aclfield.mask.u8=3;
  attr[1].id=SAI_ACL_ENTRY_ATTR_ACTION_SET_VRF;
  attr[1].value.aclaction.parameter.oid=0x3000000000ce9;

MyMac table - Introduction

NETWORKING

- Router Interface (RIF) has a Source MAC address attribute
- Used as SMAC for packets egressing from the Router interface
- Peer device on the other end of the link can discover this MAC address (via ARP, or other mechanisms), and use it as DMAC in packets sent to this device
- On some platforms, packets received from the peer with DMAC matching the RIF's source MAC address, are L3 forwarded.

MyMac table - Use case

NETWORKING

- Allow flexibility by programming the MAC address only (separately from RIF).
- This MAC address is not bound a single RIF.
- This MAC address does not need to be discovered/queried and periodically refreshed, but is signaled out-of-band by a SDN controller.
- This MAC address is used to match against ingress packet's DMAC to L3 forward the traffic.
- Allows for an arbitrary DMAC can be used to send traffic from the peer switch

MyMac table - proposal

NETWORKING

- https://github.com/opencomputeproject/SAI/pull/1243 (merged, included in SAI 1.9)
- New SAI OID object SAI_OBJECT_TYPE_MY_MAC
- Attributes: Port (wildcard if not specified), VLAN (wildcard if not specified), MAC Address with mask
- No change in RIF programming
- PR is reviewed and merged. For any enhancements or suggestions, please bring it to the community.

FEC for 200G+ Ports

NETWORKING

- FEC mode configuration currently limited to None/FC/RS
  - details of RS-FEC mode automatically determined by vendor SAI implementation
- Does not allow specification of the detailed FEC mode
- Example: for 200G PAM4 links, either RS-544 or RS-544 with 2x interleave may be used, no way to specify which is selected

FEC for 200G+ Ports

NETWORKING

- Existing FEC modes
  - SAI_PORT_FEC_MODE_NONE
  - SAI_PORT_FEC_MODE_RS
  - SAI_PORT_FEC_MODE_FC
- Added Extended FEC controls
  - SAI_PORT_FEC_MODE_EXTENDED_NONE
  - SAI_PORT_FEC_MODE_EXTENDED_RS528
  - SAI_PORT_FEC_MODE_EXTENDED_RS544
  - SAI_PORT_FEC_MODE_EXTENDED_RS544_INTERLEAVED
  - SAI_PORT_FEC_MODE_EXTENDED_FC

FEC for 200G+ Ports

NETWORKING

- Example use-cases for extended FEC settings
  - 200G PAM4 ports may use either RS544 or RS544 with interleave - prevents vendor-specific ambiguity
  - Gearbox optics for legacy compatibility
    - 4x25G -> 100G-SR2 <-> 400G-SR8 <- 2x50G
    - 4x25G side only supports RS528, default for 2x50G would be RS544
  - Extensible to additional RS-FEC variants or other FEC modes
- https://github.com/opencomputeproject/SAI/pull/1224
- PR is merged in SAI 1.9. For any enhancements or suggestions, please bring it to the community.

Call to Action

- Get involved in the SAI community at https://www.opencompute.org/wiki/Networking/SAI
- SAI headers with these changes available at https://github.com/opencomputeproject/SAI/

Thank you!

NOVEMBER 9-10, 2021