Stages in Faucet switch dev-cycle

A template for implementing Faucet compatible OF switch.

Nitin Karkhanis- nkarkhan@arista.com

ARISTA

Engineer’s perspective

- Arista has been supporting OpenFlow single-table for a while
- Initial contact with faucet - 2016 - real work was in 2019
- Team comprised of folks with platform and upper layer expertise
- OpenFlow switches are dumb - how hard can this be?

Engineer’s perspective - Denial

- Faucet matched the pipeline of many commercially available ASICs.
- Initial plan was to match faucet table to appropriate ASIC table.

Implementation reality

- Test - faucetTaggedTest - tables used - vlan, ethSrc, ethDst, flood
- Install tables in parallel TCAM lookup with table priority
  vlan - if tagged then goto ethSrc. (ignore goto)
  ethSrc - drops a few Src Macs, table miss punts to controller and goto flood. (ignore goto)
  ethDst - flows installed when MACs are learnt
  flood - sends to all ports except incoming.
  Packets match multiple flows in different tables (even unintended tables) and if there are no conflicting actions then the test passes.

Engineer’s perspective - Anger

- Faucet tables matched ASIC tables in “name”. Functionality expected was very different.

  e.g. TFM for vlan says matching can be on DstMac, EthType and/or InPort. ASIC can probably do vlan assignment based on SrcMac and InPort.
  ethSrc table could match on SrcMac, DstMac, EthType and/or InPort.

  Realization that faucet tables can’t all be mapped to ASIC tables.

  Getting a few tests to pass did not mean we had a workable approach.
  Passing tests at this point

  FaucetTaggedTest

  FaucetTaggedScaleTest
  FaucetTaggedBroadcastTest

Engineer's perspective - Bargaining

- Maybe Faucet is not that detached from ASIC reality, maybe there is no use case for matching any packet field in any table.
  Maybe some hacks are okay
- Maybe this 1 hack to glean router MAC address from flows and programming router MAC table is fine.
- Build a ARP table in SW by combining info from ethDst and FIB.
- Send packets to SW for handling hard to implement flows.

  Implementation reality
  IP packets destined to this router MAC could be processed thru ASIC's routing pipeline and apply operations like decrement TTL change src MAC, dst MAC, Vlan on the packet.
  Faucet test has routed packet in 1 direction and reply packet is bridged but has routing operations.

Engineer’s perspective - Depression

- Dev stage names don’t match reality.
- Faucet pipeline doesn’t match ASICs pipeline.
- Gleaning information from flows to figure out if it was a routed or bridged domain flow was not scalable and would increase complexity.
- If you have custom enhancements/features faucet will never fly. Nobody will write vendor specific code in controller.
- If (vendorName == “Initech”) {
  IpAddrMatchInEthSrcTable = true;
  } // will never fly!
- Switch API has to be kept generic, don’t pollute controller code with vendor specifics

Engineer’s perspective - Acceptance

- Do the right thing.
- Switch table flexibility is critical for Faucet’s success.
- Vendor/Chip specific code on the controller side doesn’t help.

Implementation reality -

- Switch code gets complicated to figure out which ASIC tables to use

Engineer’s perspective -

Current state

- 90 % of faucet tests pass
- Switching/Routing/8021.x tests pass.
- Using the right generic approach got us stacking tests.


Thank You
www.arista.com