EVERFLOW ALWAYS-ON

Shuotian Cheng shuche@microsoft.com

Motivation

Increase the mobility of packet capture ability by enabling the EVERFLOW always-on feature on SONiC devices. Once the server starts marking packets of interest, the packets are getting mirrored automatically without further intervention on the switches.

Requirement

SONiC

1. Support DSCP value/mask matching on all interfaces
2. Support rate limiting on mirrored traffic while maintain the original flow untouched
    a. Support creating a mirror session with an existing policer
    b. Support attaching a policer to an existing mirror session [low priority]
    c. Support changing the rate settings of an existing policer [low priority]
3. Reserve a stand-alone suite of an ACL rule, an ACL table, a mirror session, and a policer
4. Support both EVERFLOW on-demand and EVERFLOW always-on turned on at the same time – packet matches both criteria will be duplicated twice and sent to corresponding destinations
5. Support the command-line interface to configure the EVERFLOW always-on feature
6. Need to support warm reboot

Configuration

1. Indicate the EVERFLOW always-on feature enabled in minigraph
2. Generate EVERFLOW always-on configuration

To-Do

1. Get the requirement of the policer rate – CBS and CIR
2. Get the requirement of the DSCP value and mask

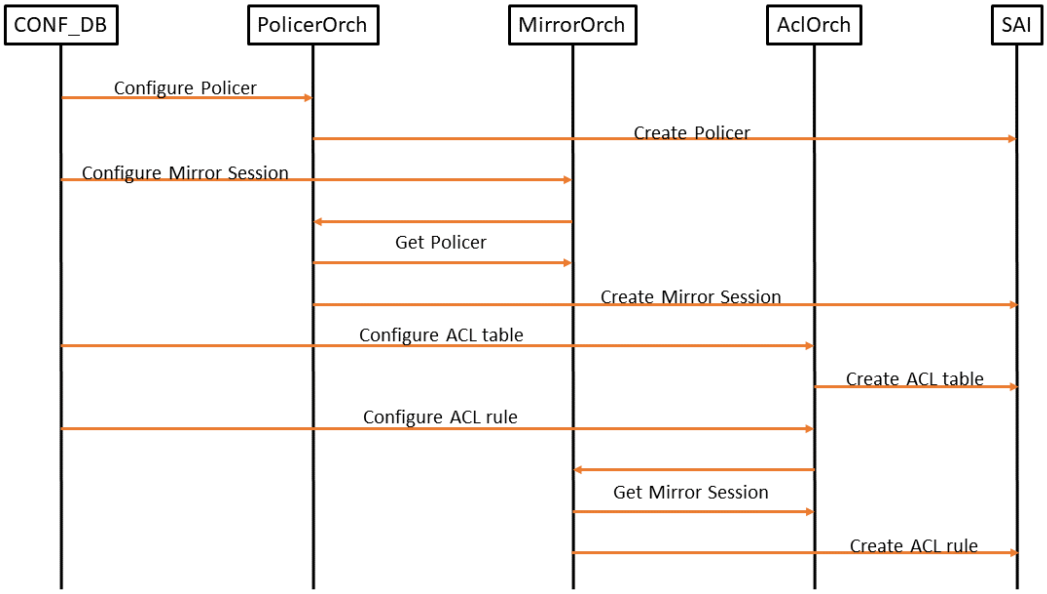
Design

SONiC

1. Add the DSCP value/mask match to support per ACL rule
2. Add the PolicerOrch to create/set/remove the policer
    a. Create/remove the policer
    b. Set rate attributes to an existing policer [low priority]
3. Attach a policer to a mirror session
    a. Create a mirror session with an existing policer
    b. Attach a policer to an existing mirror session [low priority]
4. Add the MIRROR_DSCP type ACL table to support solely DSCP matching
5. Add the CLI support [low priority]
    a. Create/remove the policer
    b. Create/remove the ACL table with MIRROR_DSCP type

c. Create/remove the ACL rule with DSCP match [low priority]
d. Set rate attributes to an existing policer [low priority]

Flow Chart



Integration Testing

[test preparation]

1. Create a policer with the following configuration
2. Create a new mirror session with a different destination other than the original used for EVERFLOW on-demand test
3. Create an ACL table with MIRROR_DSCP type
4. Create a rule with DSCP value and mask

[test core content]

5. Send one packet that matches DSCP value/mask and verify the mirrored packet is received
6. Send one packet that does not match DSCP value/mask and verify no mirrored packet is received
7. Send 100 packets that matches DSCP value/mask consecutively and verify all original packets are received
8. Send 100 packets that matches DSCP value/mask consecutively and verify that NOT all mirrored packets are received due to rate limiting

[test teardown]

9. Remove all created items including the ACL rule, the ACL table, the mirror session and the policer

Configuration Format

Sample:

```
“MIRROR_SESSION”: {
      <mirror_session_name>”: {
            “src_ip”: “<loopback_ip>”,
            “dst_ip”: “<everflow_server_ip>”,
            “dscp”: “<dscp_value>”,
            “ttl”: “<ttle_value>”,
            “queue”: “<queue_value>”,
            “policer”: “<policer_name>”
      }
},
“POLICER_TABLE|<policer_name>”: {
      “meter_type”: “bytes”,
      “mode”: “sr_tcm”,
      “red_action”: “drop”,
      “cbs”: “600”,
      “cir”: “600”
},
“ACL_TABLE|EVERFLOW_DSCP”: {
      “type”: “MIRROR_DSCP”,
      “policy_desc”: “EVERFLOW_ALWAYS_ON”,
      “ports”: <all_front_panel_ports>
},
“ACL_RULE|EVERFLOW_DSCP|RULE_1”: {
      “priority”: “9999”,
      “mirror_action”: “<mirror_session_name>”,
      “dscp”: “<dscp_value>/<dscp_mask>”
}
```