**Broadview IFA Support**

**Feature Testing:**

**Feature Description:**

BroadView INT feature enables easy setup and monitoring for the Inband Telemetry Feature sported on advanced Broadcom Switching silicon such as Trident3.

The BroadView INT feature works in conjunction with the Broadcom IFA (Inband Flow Analyzer) Embedded Application to make the switch assume the roles of different types of  INT nodes listed below:

* Ingress nodes -  that initiate the Inband Telemetry to select traffic,
* Intermediate nodes - that append INT metadata to INT traffic flowing through them, and
* Egress nodes - that are capable of extracting INT metadata stack from the traffic and send the extracted metadata to collectors.

Whole purpose of IFA feature is to know (with the help of meta data) what all nodes (hops) packets have traversed from and how long did it take to reach the destination and was there any congestion.

**Functionality:**

1. Verify that IFA telemetry data (Switch ID, Ingress/Egress Port, Ingress/Egress port timestamp, Congestion bit, Queue ID) is added before forwarding to the next hop.

2. Verify that IFA allows sample rate to be configured for ingress node.

3. Verify that IFA identifies UDP/TCP flow depending on the 5-tuple configured parameters for the flows.

4. Verify that IFA flow identification with the help of 5-tuple, Silicon needs max 2 parameters in 5-tuple and other tuple info can be 'any'.

5. Verify that IFA intermediate switch adds an extra meta-data for this node to each packet with INT header originally inserted by ingress node.

6. Verify that IFA egress switch inserts extra meta-data for this node into each TCP/UDP packet with INT header.

7. Verify that IFA egress switch allows (no sampling) telemetry data of each packet to be sent to collector.

8. Verify that IFA functionality works fine with ingress node, multiple intermediate nodes and egress node to send telemetry packets to collector.

9. Verify INT headers and Metadata headers inserted at each node for their fields.

10. Verify that IFA sampled packets from a flow are inserted with INT header at ingress and MD headers at each switch node.

11. Verify IFA metadata fields (Device-ID, Template-Id, Congestion, IP\_TTL and Queue\_id, Rx Timestamp Seconds, Rx Timestamp Nano-Seconds, Tx Timestamp Nano-Seconds, Egress Port Utilization, Ingress Port and Egress Port, Egress Port Drop Pkt byte Cnt) inserted by each node are correct.

12. Verify IFA inserted INT header contains fields (Prober Marker, Version, Message Type, Flags, Telemetry Request Vector, Hop Limit, Hop Count, Maximum Length, Current Length, Sender’s Handle, Sequence Number) are correct.

13. Verify that max flows can be created and used for IFA operation.

14. Verify that max sessions can be created and used for IFA operation.

**Warm Boot:**

1. Verify the IFA functionality for ingress-node is not disturbed during warm boot.
2. Verify the IFA functionality for intermediate-node is not disturbed during warm boot.
3. Verify the IFA functionality for egress-node is not disturbed during warm boot.

**Hard Boot:**

1. Verify that switch does not malfunction when IFA functionality is being tested and switch is hard rebooted.

**Manageability:**

1. Verify that IFA feature can be configured using REST JSON.

**Functional Interaction:**

1. NA

**Memory Leaks:**

1. Verify that there are no memory leaks when IFA feature is configured and active for 12 hours.
2. Verify that there are no memory leaks when IFA feature is configured and unconfigured repeatedly for 10 iterations with traffic forwarding.

**Scalability:**

1. Verify the max supported flows supported for IFA feature.
2. Verify the max supported sessions supported for IFA feature.

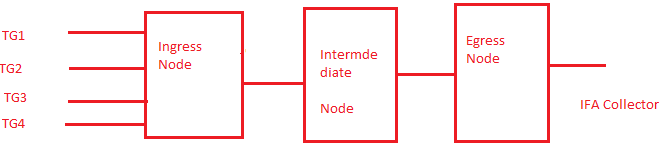
**Stress:**

1. Verify that IFA feature works fine for an overnight long run.

**Performance:**

NA

**Test Topology**



Note – 3 DUTs are needed to check the feature end to end.

**External Dependencies:**

1. Getting the test cases from the community
2. Contributing to the community by identifying the test gaps

**Development Dependencies:**

1. Functional Requirements document
2. Functional Spec
3. UI spec
4. Unit test plan
5. Build deliverables
6. Jira guidelines
7. QA test plan review/approval

**Documents needed for ramp-up/test plan preparation:**

1. Functional Requirements document
2. Functional Spec
3. UI spec

**Software tools needed:**

N/A

**Platform supported:**

Tomahawk2 and TD3

**HW Requirements:**

3qty each of Tomahawk2 and TD3

**Test Gear Requirements:**

4 Traffic Generator Ports

**Automation Plan:**

Using Pytest tool tests are planned to be automated.

**Risks:**

Pytest tool frame work availability/delay will impact the Automation plan mentioned above.

**QA efforts:**

Test Case Count: 41

Automation effort in PDs = 41 / 1.2  = 34 PDs (on an avg per day 2 tests can be automated. )

Additional QA effort(Defects logging/Re-testing unblocked tests) in PDs

34\*0.3 = 10 PDs

New test case automation time per one test case:

0.5 PDs

Feature ramp-up and Test Case writing:

5 PDs