Slingshot Troubleshooting

Version:

Contents

1 Slingshot Troubleshooting **5**

1.1 Fabric Manager Health Engine . . . 5
1.1.1 Overview . . . 5
1.1.2 Retrieving the Health Engine state . . . 6
1.2 Fabric Manager Health Monitor services . . . 7
1.2.1 Overview . . . 7
1.2.2 Rules for health monitors . . . 7
1.2.3 Detailed Description of each parameter: . . . 8
1.2.4 Creation of Health Monitors . . . 11
1.2.5 Viewing active Health Monitors . . . 11
1.2.6 Retrieving the health events . . . 12
1.2.7 Known limitations . . . 12
1.3 Example Use Cases: . . . 12
1.3.1 Creating a Health Monitor based on Routing Initialization error events. . . . 12
1.3.2 Creating a rxDiscard counter Health Monitor from rxDsicard counter event. . . . 14
1.4 Incomplete Configuration Synchronization messages . . . 16
1.4.1 Fabric-manager internal task names . . . 16
1.4.2 How to use the message . . . 17
1.5 Slingshot Topology Tool (STT) . . . 17
1.5.1 What is a topology? . . . 17
1.5.2 Function of the STT . . . 17
1.5.3 Installing STT . . . 17
1.5.4 Setting up access to compute nodes for STT . . . 18
1.5.5 Initialize . . . 20
1.5.6 Commands . . . 21
1.5.7 Data Model Load/Update/Clear . . . 52
1.5.8 Disabling diagnostics collection in STT . . . 58
1.5.9 Increasing SSH timeout value in STT . . . 58
1.5.10 Logging mechanism . . . 59
1.5.11 Batch mode . . . 59
1.5.12 Single mode . . . 60
1.5.13 Default topology . . . 60
1.5.14 Compute nodes information from STT . . . 61
1.5.15 Common problems related to Compute Nodes access from . . . 63
1.6 Current Known Behaviors, Limitations and Workarounds with Slingshot Topology Tool . . . 64
1.6.1 SSH key configuration issue impacting running STT for diagnostics . . . 64
1.6.2 “Too many open files” error while running STT . . . 64
1.6.3 Authentication error while accessing Compute Node HSN interfaces . . . 65
1.6.4 STT shows base address of algorithmic MAC address instead of actual MAC address when host-settings “mirrorSynchronization” is set to TRUE in Fabric Manager . . . 65
1.6.5 STT ‘show edge include_hostname’ shows the ‘hostname’ column as ‘unknown’ . . . 65
1.6.6 STT command doesn’t work as expected with ‘–fmn_host’ option specified . . . 66
1.6.7 STT command output redirection does not work sometimes . . . 66
1.6.8 Some scripts execution time won’t get printed while running ‘refresh_data’ . . . 67

1.7 Troubleshooting HSN Links That Are Down . . . 67
1.7.1 Why links go down/why links flap . . . 67
1.7.2 Information needed to open a bug . . . 67
1.7.3 HSN Debug Procedure . . . 68
1.7.4 An example workflow . . . 71
1.7.5 More information . . . 72
1.8 General strategies for debugging HSN links . . . 72
1.8.1 Ensure HSN switch is online . . . 72
1.8.2 Port and link configuration and media issues . . . 72
1.8.3 Other failures . . . 74
1.9 Show link flap events . . . 75
1.9.1 When to use `show-flaps` . . . 75
1.9.2 Example workflow . . . 75
1.10 Why do links flap? . . . 75
1.10.1 Links may flap for benign reasons . . . 75
1.11 How to run show - flaps . . . 76
1.11.1 How to interpret show-flaps output . . . 76
1.11.2 Command line arguments . . . 77
1.12 What to do about links which have a high score in `show-flaps` . . . 79
1.13 Validate HSN cabling . . . 79
1.14 Using linkdbg to debug downed links . . . 79
1.14.1 When to use linkdbg . . . 79
1.14.2 What information linkdbg provides: . . . 79
1.14.3 Why do links fail? . . . 79
1.14.4 How to interpret linkdbg output . . . 80
1.14.5 Specific Features: . . . 81
1.15 linkdbg action code definitions . . . 83
1.15.1 List of all action codes: . . . 83
1.15.2 Action codes, by code: . . . 83
1.16 linkdbg column definitions . . . 85
1.16.1 Error/State code interpretation . . . 85
1.17 Domain Name System (DNS) Troubleshooting . . . 87
1.17.1 Check for DNS Errors . . . 87
1.17.2 Fix P2P file and re-apply DNS Config . . . 89
1.18 HSN link failovers to ClusterStor server . . . 90
1.18.1 HSN links to ClusterStor server NICs can failover if a HSN link is down . . . 90
1.18.2 ClusterStor failovers are possible when Slingshot switches get a firmware upgrade . . . 90
1.19 LLDPAD buffer overflow . . . 91
1.19.1 Symptom . . . 91
1.19.2 Solution . . . 91
1.20 Cray TLV is not broadcasting on node reboot . . . 91
1.20.1 Step 1: Verify Cray TLV is being broadcast . . . 91
1.20.2 Step 2: Restart shasta-network-lldp service . . . 91
1.20.3 Step 3: ifdown/ifup HSN interface . . . 91
1.21 Troubleshooting Issues with River Cable Validator . . . 91
1.21.1 Symptom . . . 91
1.21.2 Description . . . 91
1.21.3 Diagnosis . . . 92
1.21.4 Solution . . . 92

2 Certificates and Authentication **92**
2.1 Slingshot Switch Certificate Provision Failure . . . 92
2.1.1 `fmn-create-certificate` returns error status code 400 . . . 92
2.1.2 Slingshot switch credential not configured properly . . . 94
2.1.3 Expected output of certificate provisioning . . . 94
2.1.4 Expected output responses in verbose mode . . . 95
2.2 Slingshot Certificate Manager . . . 95
2.2.1 Certificate Manager Keystore Corruption . . . 95

2.2.2 Fabric Manager Java Truststore Corruption . . . 96
2.3 Slingshot Fabric Manager Authentication Token Service . . . 97
2.3.1 Cannot retrieve Cray EX Keycloak OpenID Token . . . 97
2.4 Slingshot Switch Passwordless SSH Login from Fabric Management Node (FMN) Failure . . . 98
2.4.1 Symptom . . . 98
2.4.2 Description . . . 98
2.4.3 Solution for Slingshot version 0.8.0A and earlier . . . 98
2.4.4 Solution for later Slingshot versions, post 0.8.0A . . . 98
2.5 Rosetta Redfish credential mismatch . . . 99
2.5.1 Symptom . . . 99
2.5.2 Description . . . 99
2.5.3 Diagnosis . . . 99
2.5.4 Solution . . . 99

3 Compute Nodes **100**
3.1 Compute Node Troubleshooting Guide . . . 100
3.1.1 Confirm connectivity . . . 100
3.2 Troubleshooting Bond0 Interface on FMN . . . 100
3.3 Address Resolution Protocol (ARP) . . . 103
3.3.1 ARP (Address Resolution Protocol) Sizing . . . 103
3.3.2 ARP (Address Resolution Protocol) Lookup and MPI jobs . . . 103

4 Fabric Manager **104**
4.1 Troubleshoot Fabric Property HMSCollector . . . 104
4.1.1 Invalid Hostname Supplied to the HMSCollector in the Fabric Policy's FabricPropertyMap . . . 104
4.2 Troubleshooting Disconnected Groups Errors . . . 105
4.2.1 Disconnected groups in fabric policy . . . 105
4.2.2 Disconnected groups in fabric policy with no proxy . . . 106
4.3 Troubleshooting Partially Connected Groups Issue . . . 107
4.3.1 Partially connected groups in fabric policy with proxy . . . 107
4.4 Troubleshooting Live Topology Errors . . . 107
4.4.1 Live topology not available while checking agent status . . . 108
4.5 Troubleshooting Issues with Fabric Topology . . . 108
4.5.1 Topology Policy has no links . . . 108
4.6 Troubleshooting Fabric Port Link Undefined Error . . . 109
4.6.1 Fabric Port Link Undefined in Fabric Policy . . . 109
4.7 Troubleshooting Issues with Fabric Topology . . . 110
4.7.1 No fabric links on the topology policy, skipping route calculation . . . 110
4.8 OData Query for fabric events and telemetry . . . 111
4.8.1 Available Fields in events . . . 111
4.8.2 Queries . . . 111
4.9 Troubleshooting No Data on Topology Policy During Lag Calculation . . . 112
4.9.1 Fabric Port Link Undefined in Fabric Policy . . . 112
4.10 Troubleshooting IntraGroup Connectivity Errors . . . 114
4.10.1 Intra Group Connectivity in fabric policy . . . 114
4.11 Troubleshooting Issues with Fabric Topology . . . 115
4.11.1 All fabric links are offline on the topology policy, skipping route calculation . . . 115

5 Cassini **116**
5.1 Cassini Troubleshooting . . . 116
5.1.1 Checking and Fixing Misconfigured Non-VLAN Tagged Ethernet Priority Code Point (PCP) Settings 116
5.1.2 Checking and Fixing LLDP Issues on the Fabric . . . 116
5.1.3 Fixing LLDP TLV Settings on a Switch . . . 117
5.2 Cassini Host Troubleshooting . . . 117
5.2.1 Retry Handler . . . 117
5.2.2 Cassini Loopback Mode . . . 117
5.2.3 Ethernet driver configuration . . . 118
5.2.4 Cassini NIC Errors . . . 119

5.2.5 PCIe Interface Troubleshooting . . . 120
5.2.6 Ethernet Interface Troubleshooting . . . 120
5.2.7 RDMA Interface Troubleshooting . . . 120

6 Rosetta **121**
6.1 Rosetta Switch Firmware Soft Reset . . . 121
6.2 Rosetta Switch Firmware Hard Reset . . . 122
6.3 Users may observe log messages with incorrect timestamps on the Rosetta switch controller. . . . 122
6.3.1 Symptom . . . 122
6.3.2 Description . . . 122

7 CXI Diagnostics and Utilities **122**
7.1 Overview . . . 122
7.2 Minimum Setup . . . 123
7.3 Running the Diagnostics . . . 123
7.4 Device Information . . . 123
7.4.1 cxi_stat . . . 123
7.5 Bandwidth . . . 124
7.5.1 cxi_write_bw . . . 124
7.5.2 cxi_read_bw . . . 125
7.5.3 cxi_send_bw . . . 126
7.5.4 cxi_atomic_bw . . . 128
7.5.5 cxi_gpu_bw_loopback . . . 129
7.6 Latency . . . 129
7.6.1 cxi_write_lat . . . 130
7.6.2 cxi_read_lat . . . 131
7.6.3 cxi_send_lat . . . 132
7.6.4 cxi_atomic_lat . . . 133
7.7 Thermal Diagnostic . . . 134
7.7.1 Node Configuration . . . 134
7.7.2 Running the Diagnostic . . . 134
7.8 Troubleshooting . . . 137
7.8.1 Cannot Reach Server . . . 137
7.8.2 No Server Running at Remote Host . . . 137
7.8.3 TCP Port Already in Use . . . 137
7.8.4 Incorrect Program or Version . . . 137
7.8.5 Unexpected Event Type . . . 137
7.8.6 High Rate Puts and Internal-Loopack . . . 138
7.8.7 GPU Libraries . . . 138
7.8.8 Low Bandwidth . . . 138
7.8.9 List Entry Exhaustion . . . 138
7.8.10 Sync Timeout . . . 138
7.8.11 Semaphore File Already Exists . . . 139

8 Network Diagnostics **139**
8.1 Overview . . . 139
8.1.1 Minimum Setup . . . 139
8.1.2 Theory of Operation . . . 140
8.2 dgnettest . . . 140
8.2.1 dgnettest_run.sh . . . 140
8.2.2 Examples . . . 141
8.2.3 Troubleshooting . . . 144
8.3 cxibwcheck.sh . . . 146
8.3.1 Example . . . 147
8.3.2 Troubleshooting . . . 147
8.4 bwcheck.sh . . . 148
8.4.1 Example . . . 148
8.4.2 Troubleshooting . . . 149

8.5 cxiberstat.sh . . . 150
8.5.1 Example Good Run . . . 150
8.5.2 Example Bad Run . . . 150
8.5.3 Troubleshooting . . . 151

9 Firmware Update Using FAS 151
9.1 Upgrade Slingshot switch firmware on HPE Cray EX manually . . . 151
9.1.1 Step 1: Identify the switch firmware RPM file . . . 151
9.1.2 Step 2: Load firmware image into FAS . . . 151
9.1.3 Step 3: Set the permissions for S3 . . . 153
9.1.4 Step 4: Update Slingshot switch firmware . . . 154

10 Copyright and Version 157

1 Slingshot Troubleshooting

1.1 Fabric Manager Health Engine

1.1.1 Overview

Health Engine is a Slingshot Fabric Manager service available since release 1.0. All topology policies created on the Fabric Manager will automatically have a Health Engine associated under `/fabric/health-engines` with the name of the topology-policy they are associated with. This field is named `healthEngineLink` under topology policy:

```
fmctl get topology-policies/template-policy
+----------------------+-------------------------------------------------------+
|         KEY          |                        VALUE                          |
+----------------------+-------------------------------------------------------+
| active               | true                                                  |
| deploymentEngineLink | /fabric/deployment-engines/template-policy            |
| documentKind         | com:services:fabric:TopologyPolicyState               |
| documentSelfLink     | /fabric/topology-policies/template-policy             |
| fabricPropertyMap    | map[LLDP:true                                         |
|                      | MAX_LAG_PORTS:256 MTU:9216]                           |
| healthEngineLink     | /fabric/health-engines/template-policy                |
| healthMapLink        | /fabric/topology-maps/template-map                    |
| lagPropertyMap       | map[]                                                 |
| linkGroupLink        | /fabric/link-groups/template-links                    |
| routingEngineLink    | /fabric/routing-engines/dragonfly/template-routing    |
| routingPropertyMap   | map[maxNumLocalSwitches:2                             |
|                      | numGroups:1]                                          |
| switchGroupLink      | /fabric/switch-groups/template-switches               |
+----------------------+-------------------------------------------------------+
```

The per topology policy Health Engine will periodically retrieve health events (under Fabric Manager metrics) for the duration of configured operational window. Based on these events, it determines the health of the system and provide health recommendations. By default, 3 categories of health conditions are included:

- Configuration, health of Fabric configurations and configuration synchronization
- Runtime, health of runtime environment
- Traffic, health of the network and routing configurations

Each of the categories shown by Health Engine will have an associated health score:

- HEALTHY, when no errors have been detected for the category
- WARNING, configurations should be verified
- CRITICAL, there is at least one active error affecting the category

Besides categories and an overall health score for each, the Health Engine will show:

- Health recommendations, a list of problems detected an a recommendation on what should be done to fix it
- Health window, the period of time in which the errors were detected (in local time)

The health engine's periodic sweep interval (30 second default) and operational window size (2 minute default) attributes are user configurable and can be configured using `fmctl update`.

For example:

1. To update periodic sweep to 60 seconds, for a health engine associated with topology policy `template-policy`, use following command:

```
fmctl update health-engines/template-policy sweepIntervalInMilliseconds=60000
```

2. To update health operational window size to 3 minutes, for a health engine associated with topology policy `template-policy`, use following command:

```
fmctl update health-engines/template-policy operationalWindowSizeInMilliseconds=180000
```

1.1.2 Retrieving the Health Engine state

As a service exposing a REST interface, querying the service can be performed using `fmctl`. For a healthy system where all configurations have been applied and traffic is correctly flowing across the network, the expected status is as shown below.

```
fmctl get health-engines/template-policy
```

```
+-----------------------------------+-----------------------------------------------------------------------------------------+
|            KEY                    |                                VALUE                                                    |
| documentKind                      | com:services:fabric:models:HealthEngineStat                                             |
| documentSelfLink                  | /fabric/health-engines/template-policy                                                  |
| healthIssuesIdentified            | http://127.0.0.1:7777/metrics?$filter=((TimestampLong%20le%201617153456547)%20and%20    |
|                                   (TimestampLong%20ge%201617153426551)%20and%20(ParentalContext%20eq%20CrayFabricHealth)   |
| healthObjectsImpacted             | http://127.0.0.1:7777/metrics?$filter=((TimestampLong%20le%201617153456547)%20and%20    |
|                                   (TimestampLong%20ge%201617153426551)%20and%20(ParentalContext%20eq%20CrayFabricHealth)) |
|                                   | %26%24select%3DLocation%2CPhysicalSubContext                                            |
| healthRecommendations             | map[Traffic:map[]]                                                                      |
| healthStatus                      | map[Configuration:HEALTHY                                                               |
|                                   | Runtime:HEALTHY                                                                         |
|                                   |  Traffic:HEALTHY]                                                                       |
| lastEngineMaintenanceEpoch        | 1.617153456547e+12                                                                      |
| lastMaintenanceEnd                | Tue Mar 30 20:17:36 CDT 2021                                                            |
| lastMaintenanceStart              | Tue Mar 30 20:17:06 CDT 2021                                                            |
| sweepIntervalInMilliseconds       | 30000                                                                                   |
| operationalWindowSizeInMilliseconds | 120000                                                                                |
+-----------------------------------+-----------------------------------------------------------------------------------------+
```

For each health problem detected, the Health Engine will show a recommendation and the service in which the problem has been detected. The example below shows the content of the engine when a switch suddenly becomes offline:

```
fmctl get health-engines/template-policy --raw | jq
```


```
{
  "healthStatus": {
    "Runtime": "HEALTHY",
    "Configuration": "HEALTHY",
    "Traffic": "CRITICAL"
  },
  "lastMaintenanceStart": "Tue Mar 30 20:37:06 CDT 2021",
  "lastMaintenanceEnd": "Tue Mar 30 20:37:37 CDT 2021",
  "healthRecommendations": {
    "Traffic": {
      "CRITICAL": [
        {
          "issueTime": "Tue Mar 30 20:37:20 CDT 2021",
          "resourceLink": "http://127.0.0.1:7777/fabric/agents/x0c0r0b0",
          "description": "Recommend: Check why the Switch went offline ;
                          Cause: The switch is unreachable and the state is
                          unknown, either it went offline or there is
                          a communication issue."
        }
      ]
    }
  },
  "healthObjectsImpacted": "http://127.0.0.1:7777/metrics?$filter=((TimestampLong%20le%201617154657801)%20and%20
  (TimestampLong%20ge%201617153456547)%20and%20(ParentalContext%20eq%20CrayFabricHealth))%26%24select%3DLocation%2CPhysicalSubContext",
  "healthIssuesIdentified": "http://127.0.0.1:7777/metrics?$filter=((TimestampLong%20le%201617154657801)%20and%20
  (TimestampLong%20ge%201617153456547)%20and%20(ParentalContext%20eq%20CrayFabricHealth))",
  "lastEngineMaintenanceEpoch": 1617154657801,
  "sweepIntervalInMilliseconds": 30000,
  "operationalWindowSizeInMilliseconds": 120000,
  "documentVersion": 2,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:models:HealthEngineState",
  "documentSelfLink": "/fabric/health-engines/template-policy",
  "documentUpdateTimeMicros": 1617154657815000,
  "documentUpdateAction": "PUT",
  "documentExpirationTimeMicros": 0,
  "documentOwner": "ae5f6f47-20cd-4094-ad9b-73bc69bccd2a",
  "documentAuthPrincipalLink": "/core/authz/system-user"
}
```

1.2 Fabric Manager Health Monitor services

1.2.1 Overview

The Health Monitor Engine provides a common framework for periodic monitoring and analysis of data generated by microservices provided by the Fabric Manager (FM) or by the Fabric Agent running on the switches. It facilitates reporting deviations in the form of health events that are made available in the metrics store.

All the microservices in FM exposes their data in the form of service documents and this data can be retrieved by making OData query to the service resource links. The Health Monitors can be used to monitor these services at FM or switch level and generate custom health events in the field or by the customers for debugging purposes. By default, it runs all OData queries over metrics store.

Following are the main components of Health Monitor Engine:

Components	Features
Telemetry Store	A repository of Metrics and Events in Fabric Manager and Fabric Agent.
Health Monitor Engine	Facilitates creation and listing of Health Monitors. It is deployed as a microservice in Fabric Manager and Fabric Agent.
Health Monitor	Service that periodically monitors Metrics/Events based on monitoring rules, analyzes and generates health events. It is deployed as a microservice in Fabric Manager and Fabric Agent.
Rules for Health Monitors	Defines the guidelines, used by health monitors to qualify and generate the health events.

Currently we store the generated health events into metrics store.

Subsequent sections of this document provides details about: 1. Description and usage of service Rules 2. Creation of health monitors 3. Fetching/viewing existing active services 4. Retrieval of health events generated by the engine 5. Known limitations

1.2.2 Rules for health monitors

The Rules are set of criteria used by the service to qualify a symptom/behaviour as a health events. It is captured in the form of json file and used for creation of service.

Rules consists of several parameters and is defined in the following format:

- **QueryRulesList** : map/dictionary of Query Rule Name and Query Object.
  - ***query name***
  - ***oDataQuery*** : oData query for retrieving the events.
  - ***locationConstrain*** : optional, per query location constraint.
    - type
    - locationList
  - ***eventCountThresholds*** : thresholds for generating health event for event count based queries.
    - criticalUpperThreshold
    - warningUpperThreshold
  - ***counterThresholds*** : thresholds for generating health event for counter based queries.
    - type
    - criticalUpperThreshold
    - criticalLowerThreshold
    - warningUpperThreshold
    - warningLowerThreshold
  - ***resourceLink*** : optional, resource link for each odata query.
- **resourceLink** : optional, resource link for all the queries, default /metrics.
- **physicalContext** : health category and event type to be used for health event.
- **causeString** : cause string to be used for health event.
- **recommendationString** : recommendation string to be used for health event.
- **sweepIntervalInMilliseconds** : optional, maintenance sweep interval for service, default is 30 secs.

- **operationalWindowSizeInMilliseconds** : optional, size of data sample to be considered for analysis, default is 60 secs.
- **locationConstraint** : optional, global location constraint.
  - type
  - locationList
- **documentSelfLink** : optional, name for monitor service.

Sample service document format in json:

Note: The following template can be reused to create a health monitor. The content can be coppied into a file and relevant fields can be edited as per the requirement.

```
{
  "queryRulesList": {
    "<query_name_string>": {
      "oDataQuery": <oData query_String>,
      "eventCountThresholds": {
        "criticalUpperThreshold": <event threshold value>,
        "warningUpperThreshold": <event threshold value>
      },
      "counterThresholds" : {
        "type": <"DIFF"/"RATE">,
        "criticalUpperThreshold": <counter threshold value>,
        "warningUpperThreshold": <counter threshold value>,
        "criticalLowerThreshold": <counter threshold value>,
        "warningLowerThreshold": <counter threshold value>
      },
      "locationConstraint": {
        "type": <"UNIQUE"/"ANY">,
        "locationList": [<List of resource locations>]
      }
    },
    .
    .
  },
  "resourceLink": <target service link on which to run oDataQuery, default if /metrics store>,
  "locationConstraint": {
    "type": <"UNIQUE"/"ANY">,
    "locationList": [<List of resource locations>]
  },
  "physicalContext": <custom_healthEvent_category>.<custom_event_type>,
  "causeString": <customizable cause string to be used for health events>,
  "recommendationString": <customizable recommendation string to be used for health events>,
  "sweepIntervalInMilliseconds": <periodic maintenance sweep interval for service, default 30 secs>,
  "operationalWindowSizeInMilliseconds": <time window used in OData query to collect past events, default 60 secs>,
  "documentSelfLink": "<name for service>",
}
```

1.2.3 Detailed Description of each parameter:

1.2.3.1 QueryRulesList

One or more rules can be combined to form the criteria for generating a health event. All the rules can be specified under QueryRulesList.

Data Type: map/dictionary Map<Query Rule Name, Query Object>.

Each query rule must contain: 1. A name: This is an unique string which specifies the name of the query and 2. A set of query parameters for analysis as mentioned below.

NOTE: 1. If multiple Query Rules are specified then all the rules should be satisfied to raise a health event or Alerts. 2. The severity of the health event will be least severity from any query rule.

Example: Say, there are two query rules, Rule_1 matches severity of WARNING, while Rule_2 matches severity of CRITICAL. Then the health event will be generated with a severity of WARNING.

1.2.3.1.1 OData Query

The main component of the rule is the OData query filter which will be used to collect data.

Example: OData query filter for Routing Initialization Critical errors:

“oDataQuery” : “(PhysicalContext eq Traffic.RoutingInitialization and PhysicalSubContext eq CRITICAL)”

1.2.3.1.2 Per Query Local Resource link (Optional)

The local resource link on which this query should be performed on. This is an optional field and if not specified, the global resource link will be used. Note: `/metrics` store is the only supported `resourceLink` for oData Qeuries in this release.

1.2.3.1.3 Per Query Local Location Constraints(Optional):

The local Location constraint to be considered for filtering events for this query and generating health events. This is an optional field and if not specified, the global location constraint will be used.

The location constraint parameters are:

LocationConstraintType

1. UNIQUE: If this criteria is set then all the events should be from the same location. For instance, for generating a switch RoutingInit failure health event, the events should occur on the same switch.
2. ANY: If this criteria is set then the events can come from any location. Basically the location field from the event will be ignored for analysis. For instance, if all the switches have RoutingInit failure event then we can raise a health event to check switch credentials.

Location List

A resource link or a set of specific locations for which events must be filtered. If a value is specified, then the string will directly be used as a Location constraint while building OData query.

For example, if user is interested in health events only from a subset of switches/Port. This option can be useful in the field while debugging a problem, for example, if you are interested in health events only from a subset of switches/Port etc.

1.2.3.1.4 Event Count Thresholds

This parameter defines the event count thresholds for generating health events. All threshold parameters are long values.

Critical Threshold of Event Count

The threshold count of events that should be available in the time window to raise a critical event.

Warning Threshold of Event Count

The threshold count of events that should be available in the time window to raise a warning event.

1.2.3.1.5 Counter Thresholds

This parameter defines the various threshold values of a counter for generating health events. The value is calculated by taking the difference of counter value from the first and the last event.

Type (Optional)

1. DIFF: This threshold type indicates use of difference between the counter value from the last and first event for comparisons.
2. RATE: This threshold type indicates use of rate of change in the value of counter per second for comparisons.

This is a optional parameter and the default Threshold type is DIFF.

Critical Upper Threshold Value Count

The upper threshold of the difference in the value between the first and last event in the specified time window for generating a critical event.

Warning Upper Threshold Value Count

The upper threshold of the difference in the value between the first and last event in the specified time window for generating a warning event.

Critical Lower Threshold Value Count

The lower threshold of the difference in the value between the first and last event in the specified time window for generating a critical event.

Warning Lower Threshold Value Count

The lower threshold of the difference in the value between the first and last event in the specified time window for generating a warning event.

1.2.3.2 Global Resource link (Optional)

The resource link on which the query should be performed on. This is a optional parameter and by default the health monitor will look in the Metric service. Note: `/metrics` store is the only supported `resourceLink` for oData Qeuries in this release.

1.2.3.3 Global Location Constrainst(Optional):

The global Location constraint is considered for filtering events of all queries and generating health events.

The location constraint parameters are:

1.2.3.3.1 LocationConstraintType

1. UNIQUE: If this criteria is set then all the events should be from the same location. For example for generating a switch RoutingInit failure health event, the events should occur on the same switch.
2. ANY: If this criteria is set then the events can come from any location. Basically the location field from the event will be ignored from analysis. For example if all the switches has a RoutingInit failure event then we can raise a health event to check switch credentials.

1.2.3.3.2 Location List

A resource link or a set of specific locations for which events must be filtered. If a value is specified, then the string will directly be used as a Location constraint while building OData query.

For example, if user is interested in health events only from a subset of switches/Port. This option can be useful in the field while debugging a problem, for example, if you are interested in health events only from a subset of switches/Port etc.

1.2.3.4 Physical Context

This parameter specifies the name of event category and type to be used while generating health event. The required format for this field: `< Event category>.< Event name>` If Event category is not specified, then the name of the service will be used as the event category.

For example a possible value for a rule which monitors switch reboots can be: `"physicalContext":"SwitchEvent.RouteInit"`

1.2.3.5 Cause and Recommendation String

1.2.3.5.1 Cause String

This parameter will be used as the Cause string while generating the Health event.

1.2.3.5.2 Recommendation String

This parameter will be used as the Recommendation string while generating the Health event.

1.2.3.5.3 Customization of Cause and Recommendation String

For customization of the cause strings below mention special keywords can be used.

1. **$< Query Name>:LOCATION**: If the cause/recommendation string has this keyword, then the value from the "Location" field from the event data of the query rule specified by "Query Name" will be substituted in cause/recommendation string of the health message.
2. **$< Query Name>:COUNT**: If the cause/recommendation string has this keyword, then the event count or the difference of counter value from the event data of the query rule specified by "Query Name" will be substituted in cause/recommendation string of the health message.

3. **$< Query Name>:TIME**: If the cause/recommendation string has this keyword, then the time duration between the first and the last event in seconds from the event data of the query rule specified by "Query Name" will be substituted in cause/recommendation string of the health message.

For example: "causeString" : "Fabric Link $:LOCATION has flapped $:COUNT times in $:TIME."

In the above case the ' $< query_name >$: $LOCATION'willbereplacedbythequeryspecificportname,'< query_name >$: $COUNT'willbereplacedbytotalnumberoftimesspecifiedqueryeventoccuredand'$:TIME' will be replaced by the operational window size in seconds.

"causeString" : "Switch link $:LOCATION reported rxDiscards more than $:COUNT times in last $:TIME",

In the above case the '$< query_name >$: $LOCATION'willbereplacedbythequeryspecificportname,'$:TIME' will be replaced by the time duration between first and last counter reported for specified query in seconds and '$:COUNT' will be replaced by the rxDiscards count for the $TIME duration.

1.2.3.6 Window Size

The time window which will be used in OData query to collect past events.

1.2.3.7 Sweep Interval

The periodic time interval in which the health monitors will perform analysis for generating health events.

1.2.4 Creation of Health Monitors

The health monitor is described by a set of rules for identification and qualification of events and other configuration parameters for generating health events.

User can capture the required configurable parameters in the form of json file and use `POST` request on the Health Monitor Engine to create the health monitor. The json file forms the Body of the POST request.

Health Monitors can also be created programmatically by any service running inside the Fabric Manager or the Fabric Agent using helper routines provided by Health Monitor Engine.

Health Monitors are deployed as a microservices in Fabric Manager and Fabric Agent.

Command to create a service: `fmctl create health-monitors --file <service-description-filename.json>`

Example:

```
$ fmctl create health-monitors --file sample-monitor-service.json
```

Once created, the health monitor will periodically monitor the configured resource link by running the OData queries along with other constraints specified in the rules. The data gathered by the service is analyzed and qualified as a health event if the rules are satisfied.

The details about rules are defined in `Rules for health monitors` section above.

1.2.5 Viewing active Health Monitors

Command to retrive active health monitors:

`fmctl get health-monitors`

Example:

```
$ fmctl get health-monitors
+-----------------------------------------------------------+-----------------------------------------------------------+
|                            KEY                            |                           VALUE                           |
+-----------------------------------------------------------+-----------------------------------------------------------+
| documentCount                                             |                                                         2 |
| documentLinks                                             | /fabric/health-monitors/monitor-routingInit               |
|                                                         1 | /fabric/health-monitors/monitor-rxDiscard_counters        |
| documents./fabric/health-monitors/monitor-routingInit     | map[documentDescription:map[documentIndexingOptions:[] |
|                                                           | serializedStateSizeLimit:32768                            |
|                                                           | serviceCapabilities:[INSTRUMENTATION                      |
|                                                           | PERIODIC_MAINTENANCE REPLICATION OWNER_SELECTION          |
|                                                           | IDEMPOTENT_POST FACTORY_ITEM DOCUMENT_OWNER]              |
|                                                           | versionRetentionFloor:500 versionRetentionLimit:1000]     |
|                                                           | documentExpirationTimeMicros:0                            |
|                                                           | documentUpdateTimeMicros:0 documentVersion:0]             |
```

```
| documents./fabric/health-monitors/monitor-rxDiscard_counters | map[documentDescription:map[documentIndexingOptions:[] |
|                                                              | serializedStateSizeLimit:32768                         |
|                                                              | serviceCapabilities:[INSTRUMENTATION                   |
|                                                              | PERIODIC_MAINTENANCE REPLICATION OWNER_SELECTION        |
|                                                              | IDEMPOTENT_POST FACTORY_ITEM DOCUMENT_OWNER]           |
|                                                              | versionRetentionFloor:500 versionRetentionLimit:1000]  |
|                                                              | documentExpirationTimeMicros:0                         |
|                                                              | documentUpdateTimeMicros:0 documentVersion:0]          |
+--------------------------------------------------------------+--------------------------------------------------------+
```

1.2.6 Retrieving the health events

By default, the health events that are qualified are stored in metrics store. Following commands can be used to retrieve the events: - `GET` request on metrics store.

```
$ fmctl get /metrics --raw expand=true filter="(ParentalContext eq CrayFabricHealth)" expand=true | jq .
```

- `GET` request on the health engine link.

```
$ fmctl get health-engines/template-policy --raw | jq .
```

1.2.7 Known limitations

1. Health Monitoring framework is pre-enabled as part of this release. The scaling and performance aspects would be handled in subsequent releases.
2. `/metrics` store is the only supported `resourceLink` for oData Qeuries in this release.

1.3 Example Use Cases:

1.3.1 Creating a Health Monitor based on Routing Initialization error events.

1.3.1.1 POST request for creating the health monitor

Create json file with service rules in required format and use `fmctl create` by sharing the file as input using `--file`. Here considering the rules in `monitor-routing_init.json` file:

```
$ fmctl create health-monitors --file monitor-routing_init.json
```

```
$ cat monitor-routing_init.json
{
    "queryRulesList" : {
      "query1" : {
        "oDataQuery" : "(PhysicalContext eq Traffic.RoutingInitialization and PhysicalSubContext eq CRITICAL)",
        "eventCountThresholds" : {
          "criticalUpperThreshold" : 20,
          "warningUpperThreshold" : 10
        }
      }
    },
    "physicalContext" : "FabricMonitor.RoutingInit",
    "causeString" : "Switch $query1:LOCATION reported routing initialization failures $query1:COUNT times in last $query1:TIME.",
    "recommendationString" : "Please check the state of switches in the Fabric Policy X<Fabric Policy Resource link> .",
    "sweepIntervalInMilliseconds" : 30000,
    "operationalWindowSizeInMilliseconds" : 3600000,
    "locationConstraint" : {
      "type" : "UNIQUE",
      "locationList" : ["*x0c0*"]
    },
    "documentSelfLink" : "monitor-routingInit"
}
```

1.3.1.2 Health Monitor created from the above POST request

```
$ fmctl get health-monitors/monitor-routingInit --raw | jq
{
  "queryRulesList": {
    "query1": {
      "oDataQuery": "(PhysicalContext eq Traffic.RoutingInitialization and PhysicalSubContext eq CRITICAL)",
      "eventCountThresholds": {
        "criticalUpperThreshold": 20,
        "warningUpperThreshold": 10,
        "documentVersion": 0,
        "documentUpdateTimeMicros": 0,
        "documentExpirationTimeMicros": 0
      },
      "documentVersion": 0,
      "documentUpdateTimeMicros": 0,
      "documentExpirationTimeMicros": 0
    }
```

```
  },
  "resourceLink": "/metrics",
  "locationConstraint": {
    "type": "UNIQUE",
    "locationList": [
      "*x0c0*"
    ],
    "documentVersion": 0,
    "documentUpdateTimeMicros": 0,
    "documentExpirationTimeMicros": 0
  },
  "physicalContext": "FabricMonitor.RoutingInit",
  "causeString": "Switch $query1:LOCATION reported routing initialization failures $query1:COUNT times in last $query1:TIME.",
  "recommendationString": "Please check the state of switches in the Fabric Policy X<Fabric Policy Resource link> .",
  "sweepIntervalInMilliseconds": 30000,
  "operationalWindowSizeInMilliseconds": 3600000,
  "lastMonitorServiceMaintenanceEpoch": 1633030504379,
  "documentVersion": 41,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:health:models:HealthMonitorState",
  "documentSelfLink": "/fabric/health-monitors/monitor-routingInit",
  "documentUpdateTimeMicros": 1633030504379010,
  "documentUpdateAction": "PATCH",
  "documentExpirationTimeMicros": 0,
  "documentOwner": "17bb43d9-377f-440c-abe7-948988727806",
  "documentAuthPrincipalLink": "/core/authz/system-user"
}
```


1.3.1.3 Sample Health events generated in the metric store by this health monitor

```
$ fmctl get /metrics --raw filter="PhysicalContext eq FabricMonitor.RoutingInit" expand=true | jq
```


```
{
  .
  "documentLinks": [
    "/metrics/6edfff39f4faa4755cd3b436a3640"
  ],
  "documents": {
    "/metrics/6edfff39f4faa4755cd3b436a3640": {
      "Timestamp": "2021-09-30T19:15:01.351Z",
      "TimestampLong": 1633029301351,
      "Location": "http://127.0.0.1:8000/fabric/health-monitors/monitor-routingInit",
      "PhysicalContext": "FabricMonitor.RoutingInit",
      "ParentalContext": "CrayFabricHealth",
      "ParentalIndex": 0,
      "Index": 0,
      "SubIndex": 0,
      "PhysicalSubContext": "CRITICAL",
      "Value": "Recommend: Please check the state of switches in the Fabric Policy X<Fabric Policy Resource link> . ; Cause: Switch x0c0r1b0 reported rout
      "documentVersion": 0,
      "documentEpoch": 0,
      "documentKind": "com:services:fabric:telemetry:models:Metric",
      "documentSelfLink": "/metrics/6edfff39f4faa4755cd3b436a3640",
      "documentUpdateTimeMicros": 1633029301352001,
      "documentUpdateAction": "POST",
      "documentExpirationTimeMicros": 1633032901352000,
      "documentOwner": "17bb43d9-377f-440c-abe7-948988727806",
      "documentAuthPrincipalLink": "/core/authz/system-user"
    }
  },
  "documentCount": 1,
  "queryTimeMicros": 1000,
  "documentVersion": 0,
  "documentUpdateTimeMicros": 0,
  "documentExpirationTimeMicros": 0,
  "documentOwner": "17bb43d9-377f-440c-abe7-948988727806"
}
```


1.3.1.4 Sample output from health engine reporting this health event

```
$ fmctl get health-engines/template-policy --raw | jq
```


```
{
  "healthStatus": {
    "Runtime": "CRITICAL",
    "Configuration": "WARNING",
    "FabricMonitor": "CRITICAL",
    "Traffic": "CRITICAL",
    "Security": "HEALTHY"
  },
  "lastMaintenanceStart": "Fri Oct 01 00:43:42 IST 2021",
  "lastMaintenanceEnd": "Fri Oct 01 00:45:42 IST 2021",
  "healthRecommendations": {
    .
    .
    "FabricMonitor": {
```

```
      "CRITICAL": [
        {
          "issueTime": "Fri Oct 01 00:45:31 IST 2021",
          "resourceLink": "http://127.0.0.1:8000/fabric/health-monitors/monitor-routingInit",
          "description": "Recommend: Please check the state of switches in the Fabric Policy X<Fabric Policy Resource link>. ; Cause: Switch x0c0r0b0 repo
        },
        {
          "issueTime": "Fri Oct 01 00:45:31 IST 2021",
          "resourceLink": "http://127.0.0.1:8000/fabric/health-monitors/monitor-routingInit",
          "description": "Recommend: Please check the state of switches in the Fabric Policy X<Fabric Policy Resource link> . ; Cause: Switch x0c0r1b0 rep
        },
       .
       .
      ]
    },
   .
   .
  "healthObjectsImpacted": "http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%201633029342803)%20and%20(TimestampLong%20        ge%20163302922271
  "healthIssuesIdentified": "http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%201633029342803)%20and%20(TimestampLong%2        0ge%2016330292227
  "lastEngineMaintenanceEpoch": 1633029342803,
  "sweepIntervalInMilliseconds": 30000,
  "operationalWindowSizeInMilliseconds": 120000,
  "documentVersion": 3,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:models:HealthEngineState",
  "documentSelfLink": "/fabric/health-engines/template-policy",
  "documentUpdateTimeMicros": 1633029342882002,
  "documentUpdateAction": "PUT",
  "documentExpirationTimeMicros": 0,
  "documentOwner": "17bb43d9-377f-440c-abe7-948988727806",
  "documentAuthPrincipalLink": "/core/authz/system-user"
}
```


1.3.2 Creating a rxDiscard counter Health Monitor from rxDsicard counter event.

1.3.2.1 POST request for creating a Health Monitor to raise events if rxDiscards count is greater than X for any ports in 10mins.

Create json file with service rules in required format and use `fmctl create` by sharing the file as input using `--file`. Here considering that rules are in `monitor-rxDiscards.json` file:

```
$ fmctl create health-monitors --file monitor-rxDiscards.json
```

```
$ cat monitor-rxDiscards.json
    "queryRulesList" : {
      "query1" : {
        "oDataQuery" : "PhysicalContext eq Counters.rxDiscards",
        "counterThresholds" : {
          "criticalUpperThreshold" : 1000,
          "warningUpperThreshold" : 800
        }
      }
    },
    "physicalContext" : "rxRWCounters",
    "causeString" : "Switch link $query1:LOCATION reported rxDiscards more than $query1:COUNT times in last $query1:TIME",
    "locationConstraint" : {
      "type" : "UNIQUE"
    },
    "sweepIntervalInMilliseconds" : 30000,
    "operationalWindowSizeInMilliseconds" : 600000,
    "documentSelfLink" : "monitor-rxdiscards"
}
```

1.3.2.2 Health Monitor created from the above POST request

```
$ fmctl get health-monitors/monitor-rxdiscards --raw | jq
```


```
{
  "queryRulesList": {
    "query1": {
      "oDataQuery": "PhysicalContext eq Counters.rxDiscards",
      "counterThresholds": {
        "criticalUpperThreshold": 1000,
        "warningUpperThreshold": 800,
        "documentVersion": 0,
        "documentUpdateTimeMicros": 0,
        "documentExpirationTimeMicros": 0
      },
      "documentVersion": 0,
      "documentUpdateTimeMicros": 0,
      "documentExpirationTimeMicros": 0
    }
  },
  "resourceLink": "/metrics",
```

```
  "locationConstraint": {
    "type": "UNIQUE",
    "documentVersion": 0,
    "documentUpdateTimeMicros": 0,
    "documentExpirationTimeMicros": 0
  },
  "physicalContext": "rxRWCounters",
  "causeString": "Switch link $query1:LOCATION reported rxDiscards more than $query1:COUNT times in last $query1:TIME",
  "sweepIntervalInMilliseconds": 30000,
  "operationalWindowSizeInMilliseconds": 600000,
  "documentVersion": 0,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:health:models:HealthMonitorState",
  "documentSelfLink": "/fabric/health-monitors/monitor-rxdiscards",
  "documentUpdateTimeMicros": 1631616534230000,
  "documentExpirationTimeMicros": 0,
  "documentOwner": "a6031651-40e5-45fe-a7e4-ad8836b62fbe"
}
```


1.3.2.3 Sample Health events generated in the metric store by this health monitor


```
  "documents": {
    "/metrics/1294016158d578755cbf255bad100": {
      "Timestamp": "2021-09-14T10:49:24.383Z",
      "TimestampLong": 1631616564383,
      "Location": "http://127.0.0.1:8000/fabric/health-monitors/monitor-rxdiscards",
      "PhysicalContext": "monitor-rxdiscards.rxRWCounters",
      "ParentalContext": "CrayFabricHealth",
      "ParentalIndex": 0,
      "Index": 0,
      "SubIndex": 0,
      "PhysicalSubContext": "CRITICAL",
      "Value": "Recommend: null ; Cause: Switch link x1000c0j1p0 reported rxDiscards more than 1900.0 times in last 600 seconds",
      "documentVersion": 0,
      "documentEpoch": 0,
      "documentKind": "com:services:fabric:telemetry:models:Metric",
      "documentSelfLink": "/metrics/1294016158d578755cbf255bad100",
      "documentUpdateTimeMicros": 1631616564384001,
      "documentUpdateAction": "POST",
      "documentExpirationTimeMicros": 1631620164384000,
      "documentOwner": "a6031651-40e5-45fe-a7e4-ad8836b62fbe",
      "documentAuthPrincipalLink": "/core/authz/system-user"
    },
    "/metrics/1294016158d578755cbf25ec2c460": {
      "Timestamp": "2021-09-14T10:51:55.900Z",
      "TimestampLong": 1631616715900,
      "Location": "http://127.0.0.1:8000/fabric/health-monitors/monitor-rxdiscards",
      "PhysicalContext": "monitor-rxdiscards.rxRWCounters",
      "ParentalContext": "CrayFabricHealth",
      "ParentalIndex": 0,
      "Index": 0,
      "SubIndex": 0,
      "PhysicalSubContext": "CRITICAL",
      "Value": "Recommend: null ; Cause: Switch link x1000c0j1p0 reported rxDiscards more than 1900.0 times in last 600 seconds",
      "documentVersion": 0,
      "documentEpoch": 0,
      "documentKind": "com:services:fabric:telemetry:models:Metric",
      "documentSelfLink": "/metrics/1294016158d578755cbf25ec2c460",
      "documentUpdateTimeMicros": 1631616715900001,
      "documentUpdateAction": "POST",
      "documentExpirationTimeMicros": 1631620315900000,
      "documentOwner": "a6031651-40e5-45fe-a7e4-ad8836b62fbe",
      "documentAuthPrincipalLink": "/core/authz/system-user"
    },
```


1.3.2.4 Sample output from health engine reporting this health event

```
$ fmctl get health-engines/test --raw | jq
```


```
{
  "healthStatus": {
    "Runtime": "HEALTHY",
    "Configuration": "HEALTHY",
    "Traffic": "HEALTHY",
    "Security": "HEALTHY",
    "monitor-rxdiscards": "CRITICAL"
  },
  "lastMaintenanceStart": "Tue Sep 14 16:12:07 IST 2021",
  "lastMaintenanceEnd": "Tue Sep 14 16:22:07 IST 2021",
  "healthRecommendations": {
    "monitor-rxdiscards": {
      "CRITICAL": [
        {
          "issueTime": "Tue Sep 14 16:19:24 IST 2021",
          "resourceLink": "http://127.0.0.1:8000/fabric/health-monitors/monitor-rxdiscards",
```

```
        "description": "Recommend: null ; Cause: Switch link x1000c0j1p0 reported rxDiscards more than 1900.0 times in last 600 seconds"
      }
    ]
  }
},
"healthObjectsImpacted": "http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%201631616727568)%20and%20(TimestampLong%20ge%201631616127570)%20and
"healthIssuesIdentified": "http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%201631616727568)%20and%20(TimestampLong%20ge%201631616127570)%20an
"lastEngineMaintenanceEpoch": 1631616727568,
"sweepIntervalInMilliseconds": 30000,
"operationalWindowSizeInMilliseconds": 600000,
"documentVersion": 1,
"documentEpoch": 0,
"documentKind": "com:services:fabric:models:HealthEngineState",
"documentSelfLink": "/fabric/health-engines/test",
"documentUpdateTimeMicros": 1631616727602000,
"documentUpdateAction": "PUT",
"documentExpirationTimeMicros": 0,
"documentOwner": "a6031651-40e5-45fe-a7e4-ad8836b62fbe",
"documentAuthPrincipalLink": "/core/authz/system-user"
}
```


1.4 Incomplete Configuration Synchronization messages

If after running `fmn_status --details` there are "Requiring Review" messages such as following:

```
fmn_status --details
-----------------------------------------
Topology Status
Active: template-policy
Health
------
Runtime:HEALTHY
Configuration:HEALTHY
Traffic:HEALTHY
For more detailed Health - run 'fmctl get health-engines/template-policy'


Edge: 8008 / 8344
Fabric: 17894 / 17928
Ports Reported: 26222 / 26272
Fully Synchronized Switches: 523 / 524

Switches with Incomplete Configuration Synchronization Requiring Review:
x1000c3r3b0, Initialization Sync: FINISHED, Route Sync: FAILED, Configuration Sync: FINISHED, Firmware Sync: FINISHED, /fabric/agents/x1000c3r3b0

Downed links:
<snip>
```

There are potentially 4 tasks' status be shown in the message: "Initialization Sync" is for fabric-manager sending routing initialization to Rosetta switch. "Route Sync" is for fabric-manager sending routing state update to Rosetta switch. "Configuration Sync" is fabric-manger sending other configurations to Rosetta switch. "Firmware Sync" is for updating new Rosetta firmware version.

Because some tasks take time to finally configure the switch, if we see tasks are not FINISHED, please wait up to 30 seconds then execute `fmn_status --details` again.

If any of task is in FAILED status, use fmctl to get more details on failed synchronization task. for example, upper mentioned message from `fmn_status --detail`: Switches with Incomplete Configuration Synchronization Requiring Review: x1000c3r3b0, Initialization Sync: FINISHED, Route Sync: FAILED, Configuration Sync: FINISHED, Firmware Sync: FINISHED, /fabric/agents/x1000c3r3b0

The message shows routing state synchronizing with switch x1000c3r3b0 is failed. Now we can take look at the failed synchronization task by executing fmctl command. For example:

```
fmctl get agents/x1000c3r3b0 --raw | jq .routingUpdateSyncTask.failure.message
"Connection reset by peer"
```

1.4.1 Fabric-manager internal task names

For getting details on "Initialization Sync", use routingInitSyncTask

```
fmctl get agents/x1000c3r3b0 --raw | jq .routingInitSyncTask.failure.message
```

For getting details on "Route Sync", use routingUpdateSyncTask

```
fmctl get agents/x1000c3r3b0 --raw | jq .routingUpdateSyncTask.failure.message
```

For getting details on "Configuration Sync", use switchConfigurationSyncTask

```
fmctl get agents/x1000c3r3b0 --raw | jq .switchConfigurationSyncTask.failure.message
```

For getting details on “Firmware Sync”, use switchFirmwareSyncTask

```
fmctl get agents/x1000c3r3b0 --raw | jq .switchConfigurationSyncTask.failure.message
```

1.4.2 How to use the message

The message reveals that the TCP socket connection got issue with the switch. For example, it might show: “x1000c3r3b0: nodename nor servname provided, or not known”

Which means the switch name was not find in DNS

Sometimes it might show: “Connection reset by peer”

Which means probably the switch is offline, or the fabric-agent on the switch host is not running in good state, etc. Troubleshoot network connectivity, For example, ping the switch to see if it is online or not:

```
PING x1000c3r3b0 (10.104.0.96) 56(84) bytes of data.
^C
--- x1000c3r3b0 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4073ms
```

If ping test is passed, then we can log into the switch to check nginx log, check fabric-agent is running properly, etc.

1.5 Slingshot Topology Tool (STT)

The Slingshot Topology Tool (STT) is a command line interface tool which maintains the Slingshot topology and runs diagnostic tests. STT enables:

- import/export of different file formats (SHCD files, point-to-point files),
- generation of topologies from plugin algorithms,
- inspection, modification, generation, and validation of topologies,
- consolidation of diagnostic test scripts from various repositories, test execution and result reporting.

It is primarily used through a CLI, as described below and can be invoked from a non-compute node.

1.5.1 What is a topology?

For the purpose of the tool, a topology is defined as a map of port-port connections for a Slingshot fabric.

1.5.2 Function of the STT

The slingshot topology tool can inspect, modify, generate, and combine topologies.

File formats used to describe and configure a Slingshot topology:

- SHCD files
- Point-to-point files
- V2 Fabric Templates

STT uses topology generator plugins to generate topologies. Currently, the only plugin supported is the Shasta Cabling Tool.

To describe how topology description file formats may be combined, it is possible to load a Shasta-P2P describing a cabinet and a V2 Fabric Template describing a multi-cabinet system. Then, one can combine them and add fabric links via CLI, verify the topology is a valid network topology using a plugin topology generator, and export the combined topology as a new total Shasta-P2P and V2 Fabric Template.

1.5.3 Installing STT

1.5.3.1 Shasta SLES

This section describes the steps for the installation and removal of STT. STT is a self-contained python package.

1.5.3.1.1 Prerequisites

The CentOS RPM of STT depends on the following rpms along with a few other generic third party python packages, please ensure they are available before installing STT:

- python3-rosetta-dev-centos
- python3-parallelssh

The SLES RPM is dependent on the following package along with a few other generic third party python packages:

- python3-rosetta-dev
- python3-parallelssh

The Compute Node snapshot feature of STT requires the following package to be installed on compute nodes:

- slingshot-utils

  Please ensure that slingshot-utils is installed on compute nodes. The above package ensures the availability of certain scripts in the compute node which provide snapshot data. Refer to the `snapshot_data` command documentation below for instructions on using snapshot feature.

1.5.3.1.2 Installation

1. CentOS RPM

   First, any existing revisions of slingshot-tools need to be uninstalled. Use `yum remove` if slingshot-tools for centos was installed. Use `pip3 uninstall` if the pip module of slingshot_tools was installed.

   Ensure that the prerequisite RPM repositories have been added in the list of yum repos.

   To get open source python3 centOS RPMs, add the following repos:

   ```
   sudo yum -y install --nogpgcheck epel-release
   sudo yum -y install --nogpgcheck https://extras.getpagespeed.com/release-latest.rpm
   ```

   The internal Slingshot RPMs like python3-rosetta-dev-centos, python3-parallelssh and slingshot-tools should be available in the Slingshot centos8_ncn repository.

   Add these repositories using yum-config-manager.

   ```
   yum-config-manager --add-repo <link_to_repo>
   ```

   Install the RPM.

   ```
   yum install -y --nogpgcheck slingshot-tools (or)
   yum install -y --nogpgcheck <link_to_specific_revision_of_RPM>
   ```

2. SLES RPM

   First, any existing revisions of slingshot-tools need to be uninstalled. Use `zypper rm`, if slingshot-tools for sles was installed. Use `pip3 uninstall` if the pip module of slingshot_tools was installed.

   Ensure that the prerequisite RPM repositories have been added in the list of zypper repos.

   The internal Slingshot RPMs like python3-rosetta-dev, python3-parallelssh and slingshot-tools should be available in the Slingshot sles15 sp2 repository. Add these repositories using zypper addrepo.

   ```
   zypper addrepo -n <repo_name> -p 80 -Gcf <link_to_repo> <repo_alias>
   zypper refresh <repo_alias>
   ```

   Install the RPM.

   ```
   zypper in <link_to_SLES_rpm>
   ```

1.5.4 Setting up access to compute nodes for STT

STT provides various commands to diagnose, troubleshoot and run performance benchmarks on compute nodes. As STT is launched at Fabric Manager Node (FMN) or Container, it connects to compute nodes over management ethernet network using SSH and not over HSN interfaces.

STT could access compute nodes to collect the diagnostic data if and only if customer provides compute node SSH password or imports compute node administrator SSH key-pair into Fabric Manager. However, this is an extended diagnostic feature and if the security is a concern, customer can opt out this capability.

STT needs both the compute node management hostname and the HSN interface name to extract information from the compute nodes. It extracts the compute node's management hostname from HSN interface's xname in p2p file and uses that to access using SSH, just the way it does for switches.

STT uses following algorithim to extract compute node's management hostname and HSN interface number information: 1. First split the compute node hostname in the SHCD/P2P file into two parts with the first occurrence of `h`, 2. Left part is used as the compute node management hostname and the right part is used as the HSN interface number. With this naming convention and other prerequisites setup properly, STT can access compute nodes and extract information about right HSN interface.

For example: 1. For HSN interface x9000c1s0b0n0h0 `dsta/b` entry in p2p file, node hostname as x9000c1s0b0n0 will be used for ssh and '0' would be used along the configured hsn_prefix using `compute_nodes_hsn_prefix` STT command (`hsn` is default prefix), as HSN interface name e.g. hsn0 or ens0 etc. 2. For HSN interface 'apollo-1h801' as `dst/b` entry in p2p file along with `ens` as hsn prefix, 'apollo-1' would be used a node hostname for ssh and 'ens801' as HSN interface.

To access compute nodes from STT, following prerequisites shall be setup first at STT host:

1. Password-less ssh login setup

   If password-less ssh to the nodes is set up using non-default SSH key pair on the FMN or uses a username/password instead, the administrator can provision STT to use custom imported SSH private key-pair or password based login credentials to access Compute Nodes. If nothing is provisioned, STT tries to access Compute Nodes without any password. This step needs to be followed on each STT launch before running commands related to compute nodes.

   ```
   ncn-m001# slingshot-topology-tool

   (STT) help compute_nodes_creds

   Login Credentials for Compute Nodes as username/password or ssh key pair.
   Usage: compute_nodes_creds --[ssh-key,password] [SSH key file full path, username:password]
   (STT)
   ```

   Note:

   1. The credentials will not be stored in any persistent store.
   2. The following list of commands will not display correct data, if administrator opt out to setup the compute node credential:

   ```
   show hsn_nodes                                          - Show hsn interfaces of all hsn nodes
   show hsn_nodes lldp                                     - Show lldp info of all hsn nodes
   show hsn_node hsn <node hostname list>                  - Show hsn nodes hsn interface info
   show hsn_node lldp <node hostname list>                 - Show hsn nodes lldp info
   show hsn_traffic ping-all-to-all <nodes_list> <hsn_ip_list> <detailed> - Show all to all connectivity between hsn nodes
   show hsn_traffic roce_perf_check_loopback <nodes_list>  - Show RoCE Perf benchmark results at compute node port level
   snapshot_data compute_nodes <all|list of CNs>           - Takes a backup/snapshot of network config of
                                                             all/given accessible compute nodes
   show cables                                             - Show all cables info
   ```

   3. This command will clear STT data model cache, so any subsequent STT command will refetch data using newly provisioned credentials.

   For example:

   When password based login is used to access compute nodes, below command can provision STT to use them for accessing compute nodes:

   ```
   (STT) compute_nodes_creds --password username:password
   ```

   When compute nodes are provisioned to use non-default ssh public keypair file, provision STT with required private ssh keypair for accessing compute nodes:

   ```
   (STT) compute_nodes_creds --ssh /root/.ssh/id_rsa_private
   ```

2. Compute nodes' xname name resolution to Ethernet IP address

   DNS services or /etc/hosts file shall be configured at FMN to allow translation of compute nodes' xname to its Ethernet based IP address. This will allow STT to access compute nodes over Ethernet based management network. Example:

   ```
   root@host1:~ \# ssh x3000c0s22b1n0
   nid000043:~ \#
   ```

   With above configuration setup properly, confirm proper working of 'ssh ' at FMN level.

STT code do not depend on compute node's hostname anymore (as was in earlier versions with 'nid' prefix), admins can now use any string as CN's hostname.

The following are the list of commands depends on the above mentioned prerequisites:

```
show hsn_nodes                                          - Show hsn interfaces of all hsn nodes
show hsn_nodes lldp                                     - Show lldp info of all hsn nodes
show hsn_node hsn <node hostname list>                  - Show hsn nodes hsn interface info
show hsn_node lldp <node hostname list>                 - Show hsn nodes lldp info
show hsn_traffic ping-all-to-all <nodes_list> <hsn_ip_list> <detailed> - Show all to all connectivity between hsn nodes
show hsn_traffic roce_perf_check_loopback <nodes_list>  - Show RoCE Perf benchmark results at compute node port level
snapshot_data compute_nodes <all|list of CNs>           - Takes a backup/snapshot of network config of
                                                          all/given accessible compute nodes
show cables                                             - Show all cables info
```

1.5.5 Initialize

Launch the STT tool by running runcli.py script, or launch the STT tool using the binary `slingshot-topology-tool`.

1.5.5.1 Use `runcli.py` script

Launch STT by running runcli.py script. Specify locations for template and point-to-point file.

Default filter profile and topology profile will be created from the default path specified.

```
[root@fmn slingshot_tools]# ./runcli.py

STT diags log directory -  /root/slingshot_tools/stt_diags_logs
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/default
Loading  point2point file /opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv to default topology
Loading fabric template file /opt/cray/fabric_template.json to default topology
JSON Validation succeeded
Welcome to the Slingshot Topology Tool v1.3.0.
     General Usage is <command> <arguments>
     Type help or ? to list commands.

(STT)
```

1.5.5.2 Launch STT using the `slingshot-topology-tool` binary

Launch interactive version of the STT. To exit STT, `exit` Note the locations of the log directories and the default topology description files.

```
[root@fmn slingshot_tools]# slingshot-topology-tool
STT diags log directory -  /root/slingshot_tools/stt_diags_logs
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/default
Loading  point2point file /opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv to default topology
Loading fabric template file /opt/cray/fabric_template.json to default topology
JSON Validation succeeded
Welcome to the Slingshot Topology Tool v1.3.0.
     General Usage is <command> <arguments>
     Type help or ? to list commands.

(STT)
```

To see STT options, launch using `-h`.

```
# slingshot-topology-tool --help
usage: slingshot-topology-tool [-h] [--verbose] [--logfile LOGFILE]
                               [--logdir LOGDIR] [--sctpath SCTPATH]
                               [--slingshotdefspath SLINGSHOTDEFSPATH]
                               [--cmd CMD] [--cmdfile CMDFILE] [-t TOPOLOGY]
                               [-p P2P] [--fmn_host FMN_HOST]
                               [--fmn_port FMN_PORT] [--fmn_secure]
                               [--version] [--editConfigOnly]
                               [--ssh_conn_timeout SSH_CONN_TIMEOUT]
                               [--ssh_conn_retries SSH_CONN_RETRIES]
                               [--ssh_remote_cmd_timeout SSH_REMOTE_CMD_TIMEOUT]

CLI for Slingshot Topology Tool.

optional arguments:
  -h, --help            show this help message and exit
  --verbose             Set log level to debug
  --logfile LOGFILE     filename for log file
  --logdir LOGDIR       directory path for all log files
  --sctpath SCTPATH     override default SCT path
  --slingshotdefspath SLINGSHOTDEFSPATH
                        override Slingshot.py path
  --cmd CMD             standalone command to run in STT without entering
                        prompt
```

```
--cmdfile CMDFILE     path to new line separated commands to run
-t TOPOLOGY, --topology TOPOLOGY
                      path to topology template file
-p P2P, --p2p P2P     path to the point-to-point file
--fmn_host FMN_HOST   specific FMN host name to get REDFISH creds
--fmn_port FMN_PORT   specific FMN host port number to get REDFISH creds
--fmn_secure          enables HTTPS (secure) protocol
--version             show program's version number and exit
--editConfigOnly      disable collecting diagnostic information
--ssh_conn_timeout SSH_CONN_TIMEOUT
                      specify SSH connection timeout value for larger
                      systems
--ssh_conn_retries SSH_CONN_RETRIES
                      specify SSH connection retry value for larger systems
--ssh_remote_cmd_timeout SSH_REMOTE_CMD_TIMEOUT
                      specify SSH remote command execution timeout value for
                      larger systems
```

1.5.6 Commands

The following are the list of commands and short descriptions of their capabilities.

As STT is rapidly evolving, this output is representational; actual output may vary.

NOTE:

All tables below may be truncated for readability purposes.

Command	Description
del	Delete topologies or filter profiles.
combine	Combine topologies into a new topology.
add	Add components and links to active topology.
filter	Modify and list attributes of active filter.
list	List created topologies or filter profiles.
new	Create new topologies or filter profiles.
run	Run diagnostic command.
set	Set active working topology or filter profile.
exit	Exits the program.
help	Help menu of various commands within STT.
load	Loads information into the active topology from specified file formats.
remove	Removes components and links from active topology.
save	Save active topology into a file in specified format.
show	Shows components of the active topology and their properties.
copy	Copies switches to another topology to create sub-topologies.
set_active	Set active working topology or filter profile.
history	View, run, edit, save, or clear previously entered commands
shell	Execute a command as if at the OS prompt
generate	Generate links using algorithm
clear_cache	Clear cached SHDT data model.

NOTE: The above set of commands display various metrics collected from different sources. Detailed information about each of the metrics, such as “LinkErrors, Congestion, and different RFCs”, can be found in “Cray Shasta Administration Guide”, under the section: “Monitor the System” –> “Fabric Telemetry Metrics”

1.5.6.1 `help`

List available commands.

```
(STT) help
Documented commands (use 'help -v' for verbose/'help <topic>' for details):
============================================================================
(STT)
add                  copy                 help                 refresh_data         set_active
clear_cache          del                  history              remove               set_threadpool_size
combine              exit                 list                 run                  shell
compute_nodes_creds  filter               load                 save                 show
configure            generate             new                  set                  snapshot_data
```

List specific command's usage.

```
(STT) help add
    Add components and links to active topology
    Usage: add <switch|link|edge> <arguments>
(STT) help set
    Usage: set [-h] [-v] [param] [value]
    Set a settable parameter or show current settings of parameters

positional arguments:

param     parameter to set or view
value     new value for settable

optional arguments:

-h, --help   show this help message and exit
-v, --verbose include description of parameters when viewing
```

1.5.6.2 `refresh_data`

This command refreshes the STT data model by clearing the in-memory cached data. By default, it updates the data model without HSN related data. If the user wants to update the HSN data as well, it can be done by adding the optional flag '–include-hsn-data' to the command.

```
(STT) help refresh_data
Refresh STT data model by clearing any cached data.
Usage: refresh_data [--include-hsn-data]
```

*Note: Depending upon the number of Compute Nodes, this option can take long time to complete.

1.5.6.3 `run`

Run a diagnostic script from within a predefined set of diagnostic scripts. `help run` displays the list of diagnostic scripts that can be run.

```
(STT) help run

        Run diagnostic command
        Usage: run <command> <command options>
        Usage: run <command> <help|summary>

Available list of diagnostic commands:

         dgrperfcheck                  - Rosetta Diagnostics for performance check.
         dgrlinkstat                   - Rosetta Diagnostics for links statistics.
         dgrerrstat                    - Rosetta Diagnostics for error statistics.
         dgrheadshellstat              - Rosetta Diagnostics for Headshell statistics.
         dgrflowdebug                  - Rosetta Diagnostics for flow control debugging.
         dgrcounters                   - Rosetta Diagnostics for capturing counter data.
         check-fabric                  - Validates current state of a fabric with fabric template file.
         check-switches                - Validates L1/L2 cabling with p2p file.
         fmn_status                    - Provides summary of fabric switch ports status.
         simple_discovery              - Finds connected Switch/Node/Chassis BMCs to the SMS/NCN.
         services_rosetta              - Provides summary of services running on Rosetta.
         services_platform             - Provides summary of services running on Switch platform.
         dgrvalidatesyscfg             - Provides health snapshot of switches
         linkdbg                       - Provides health snapshot of links
         fabric_snapshot               - Collects fabric information from the FMN.
         show-flaps                    - Provides link flapping information.
         compute_snapshot              - Collects Network Config Snapshot from the CNs.
         dgrcsr                        - Dumps Rosetta CSR data of switches
```

1.5.6.3.1 `run <command> help`

The supported commands have help text which can be viewed by `run <command> help`. For example:

```
(STT) run check-fabric help
Working with 'default' topology.
usage: check-fabric [-h] [-t TOPOLOGY] [-p PTP] [-a] [-e] ... [--version]


Compares the current running state of a slingshot network with a
fabric template file at /opt/cray/fabric_template.json and displays
any links that are inconsistent with the expected state.


optional arguments:
  -h, --help            show this help message and exit
  -t TOPOLOGY, --topology TOPOLOGY
```

```
                      Topology configuration file
  -p PTP, --ptp PTP   Point to Point file. Allows for host lookup
  -a, --all           Show ports in all status. Default is to show disabled or enabled only
  -e, --enabled       Show enabled ports only. Default is disabled ports only
  -d, --edge          Include edge ports in output
  -l, --local         Include local fabric ports in output
  -g, --globe         Include global fabric ports in output
  -n, --names         Attempt to use xnames to resolve human readable hostnames
  -u, --unused        Include Unused ports in output
  -v, --verbosity
  -V API_VERSION, --api_version API_VERSION
                      API version to use
  --version           show program\'s version number and exit
```

1.5.6.3.2 `run <command> summary`

To summarize lengthy output generated by `run <command>` use `run <command> summary` sub-option.

```
(STT) run dgrlinkstat summary
Working with 'default' topology.
Initializing...
System: fabric_template
Ports: 0xffffffffffffffff
Switches: x0c0r0b0 x0c0r1b0
Getting rates. This could take some time...

SUMMARY
=======
Number of ports with corrected BER >=
1e-04:  0
1e-05:  0
1e-06:  0
1e-07:  7 ==========
1e-08: 19 ==========================
1e-09: 29 ========================================
1e-10: 10 ==============
1e-11:  3 =====
1e-12:  0
1e-13:  0
1e-14:  0
1e-15:  0

No ports with non-zero uncorrected BER

No edge ports with non-zero RX Pause

No edge ports with non-zero TX Pause


(STT)
```

1.5.6.4 `add`

Add fabric components – switch, link, edge – to an active topology.

```
(STT) help add
       Add components and links to active topology
       Usage: add <switch|link|edge|nics> <arguments>
```

Note: The following format must be used for specifying switch, link and edge:

```
add switch <xname> <switch number> <group number> <switch type>
add link <xname1> <xname2>
add edge <switch> <jack> <opt. dst_a> <opt. dst_b>
```

For example, see existing switches, add a switch, and verify. In this example, note the format of the add switch command:

```
(STT) show switches
Working with default topology and default filter profile.
Collecting data using `check-switches` script.
Collecting data using `dgrerrstat` script.
Collecting data using `dgrperfcheck` script.
+------------+----------+------+------+------------+--------------+---------+----------+------+-----------+-------------+-------------+
|   xname    |   type   | snum | gnum | edge_count | fabric_count |  uptime | up_ports | flap | checkidle |    pktIn    |    pktOut   |
+------------+----------+------+------+------------+--------------+---------+----------+------+-----------+-------------+-------------+
| x3000c0r24 | Columbia |  0   |  0   |     12     |      4       | 33 days |  16/64   |  12  |    idle   | 76487990217 | 76492665999 |
| x5000c1r1  | Colorado |  0   |  1   |     16     |      30      | 29 days |  35/64   |  21  |    idle   |  475236261  |  543225882  |
| x5000c1r3  | Colorado |  1   |  1   |     16     |      30      | 32 days |  35/64   |  25  |    idle   |  5934110689 |  6091560233 |
| x5000c1r7  | Colorado |  3   |  1   |     16     |      28      | 32 days |  34/64   |  26  |    idle   |  7386748984 |  7564234631 |
| x5000c3r5  | Colorado |  6   |  1   |     16     |      28      | 24 days |  42/64   |  14  |    idle   | 11978393890 | 12609906010 |
| x5000c3r3  | Colorado |  5   |  1   |     16     |      28      | 24 days |  42/64   |  14  |    idle   | 17867247377 | 18603841637 |
| x5000c3r7  | Colorado |  7   |  1   |     16     |      28      | 24 days |  42/64   |  14  |    idle   | 18938104589 | 19665986812 |
| x5000c3r1  | Colorado |  4   |  1   |     16     |      28      | 24 days |  41/64   |  13  |    idle   | 11952596580 | 12537913899 |
| x5000c1r5  | Colorado |  2   |  1   |     16     |      28      | 32 days |  34/64   |  26  |    idle   |  421871191  |  483842936  |
```

```
+------------+----------+------+------+------------+-------------+---------+----------+------+----------+-------------+-------------+
```

```
(STT) add switch x0c0r1 0 0 Colorado
Working with default topology.
(STT) show switches
Working with default topology and default filter profile.
JSON Validation succeeded
JSON Validation succeeded
Collecting data using `check-switches` script.
Collecting data using `dgrerrstat` script.
Collecting data using `dgrperfcheck` script.
+------------+----------+------+------+------------+-------------+---------+----------+------+----------+-------------+-------------+
|   xname    |   type   | snum | gnum | edge_count | fabric_count |  uptime | up_ports | flap | checkidle |    pktIn    |    pktOut   |
+------------+----------+------+------+------------+-------------+---------+----------+------+----------+-------------+-------------+
| x3000c0r24 | Columbia |  0   |  0   |     12     |      4      | 33 days |  16/64   |  12  |    idle   | 76487990217 | 76492665999 |
| x5000c1r1  | Colorado |  0   |  1   |     16     |      30     | 29 days |  35/64   |  21  |    idle   |  475236261  |  543225882  |
| x5000c1r3  | Colorado |  1   |  1   |     16     |      30     | 32 days |  35/64   |  25  |    idle   |  5934110689 |  6091560233 |
| x5000c1r7  | Colorado |  3   |  1   |     16     |      28     | 32 days |  34/64   |  26  |    idle   |  7386748984 |  7564234631 |
| x5000c3r5  | Colorado |  6   |  1   |     16     |      28     | 24 days |  42/64   |  14  |    idle   | 11978393890 | 12609906010 |
| x5000c3r3  | Colorado |  5   |  1   |     16     |      28     | 24 days |  42/64   |  14  |    idle   | 17867247377 | 18603841637 |
| x5000c3r7  | Colorado |  7   |  1   |     16     |      28     | 24 days |  42/64   |  14  |    idle   | 18938104589 | 19665986812 |
| x5000c3r1  | Colorado |  4   |  1   |     16     |      28     | 24 days |  41/64   |  13  |    idle   | 11952596580 | 12537913899 |
| x5000c1r5  | Colorado |  2   |  1   |     16     |      28     | 32 days |  34/64   |  26  |    idle   |  421871191  |  483842936  |
|   x0c0r1   | Colorado |  0   |  0   |     0      |      0      |         |          |  0   |           |      0      |      0      |
+------------+----------+------+------+------------+-------------+---------+----------+------+----------+-------------+-------------+
```

Similarly, we can add edge ports and verify using `show edge`.

In this example, note the format of add edge command `add edge <switch> <jackno> <opt. dst_a> <opt. dst_b>`

Note: Add edge link command takes either 2 or 4 parameters.

In the example below arguments 3 and 4 are destination ports:

```
(STT) add edge x5000c1r5 102
Working with 'topo1' topology.
(STT) show edge
Working with 'topo1' topology and 'default' filter profile.
All invalid rows would be skipped.
Collecting data using 'check-switches' script.
Collecting data using 'check-fabric' script.
Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.
Collecting data using 'dgrlinkstat' script.
Collecting data using 'dgrperfcheck' script.
+-----------------+------+------+---------------+--------+----------+---------+------+-------------------+-------------+---------+------------+
|      xname      | type | dst  |    hostname   | status |  ptype   | subtype | mtu  |        mac        |   mactype   |  speed  |   media    |
+-----------------+------+------+---------------+--------+----------+---------+------+-------------------+-------------+---------+------------+
| x5000c1r5j102p0 | edge | None | No Connection |  True  | Ethernet |   Edge  | 9216 | 02:00:00:00:08:a3 | algorithmic | BJ_100G | electrical |
| x5000c1r5j102p1 | edge | None | No Connection |  True  | Ethernet |   Edge  | 9216 | 02:00:00:00:08:a2 | algorithmic | BJ_100G | electrical |
+-----------------+------+------+---------------+--------+----------+---------+------+-------------------+-------------+---------+------------+
```

```
(STT) add edge x5000c1r5 107 x5000c1s7b1n0h1 x5000c1s7b1n1h1
Working with 'topo1' topology.
(STT) show edge
Working with `topo1` topology and `default` filter profile.
All invalid rows would be skipped.
Collecting data using 'check-switches' script.
Collecting data using 'check-fabric' script.
Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.
Collecting data using 'dgrlinkstat' script.
Collecting data using 'dgrperfcheck' script.
+-----------------+------+--------------------+-----------+--------+----------+---------+------+-------------------+-------------+---------+------------+
|      xname      | type |        dst         |  hostname | status |  ptype   | subtype | mtu  |        mac        |   mactype   |  speed  |   media    |
+-----------------+------+--------------------+-----------+--------+----------+---------+------+-------------------+-------------+---------+------------+
| x5000c1r5j102p0 | edge | x5000c1s2b1n1h0 | nid001008 |  True  | Ethernet |   Edge  | 9216 | 02:00:00:00:08:b1 | algorithmic | BJ_100G | electrical |
| x5000c1r5j102p1 | edge | x5000c1s2b1n1h1 | nid001008 |  True  | Ethernet |   Edge  | 9216 | 02:00:00:00:08:b0 | algorithmic | BJ_100G | electrical |
| x5000c1r5j107p0 | edge | x5000c1s7b1n0h1 | nid001031 | False  | Ethernet |   Edge  | 9216 | 02:00:00:00:08:82 | algorithmic | BJ_100G | electrical |
| x5000c1r5j107p1 | edge | x5000c1s7b1n1h1 | nid001032 | False  | Ethernet |   Edge  | 9216 | 02:00:00:00:08:83 | algorithmic | BJ_100G | electrical |
+-----------------+------+--------------------+-----------+--------+----------+---------+------+-------------------+-------------+---------+------------+
```

1.5.6.5 `del`

Delete topologies or filter profiles.

```
(STT) help del
       Delete topologies or filter profiles
       Usage: del <topology,filter> <list of names>
```

1.5.6.6 `filter`

Modify/list attributes of an active filter.

Attributes added into filter will not be visible in the show command. during launch default and no filter will be created.

```
(STT) help filter
        Modify and list attributes of active filter.
        Also save the active filter to a file and load from file to active filter.
        Usage: filter <add,remove,list> <filter component for add/remove> <arguments>
        Usage: filter <save,load> <file name>
```

1.5.6.6.1 `filter list`

List filters and their properties.

This list of filters is representational and actual list will vary. Please use `filter list` to see available options.

```
(STT) filter list
Working with 'default' filter profile.
Active Filter Profile Details:

default:
{
        switch     : ['fw', 'pingable', 'dynamic', 'slwInit', 'warnCnts', ... ]
        port       : ['dynamic', 'last_fault', 'corrected_rate', 'uncorrected_rate', ... ]
        cable      : ['srcgrp', 'dstgrp', 'srcsw', 'dstsw', 'dynamic', ... ]
        p2p : ['stage', 'part_number', 'part_length', 'calculated_distance', ... ],
        headshell  : ['num_cmis_jacks', 'num_jacks_with_alarms', 'num_jacks_with_flags', ... ],
        jack       : ['DataPathPwrUp', 'RX_LOL_flags', 'RX_LOS_flags', 'TX_LOL_flags', ... ],
    }
```

1.5.6.6.2 `filter add`

Add a new filter called `status` to the list of switch filters. Add a new filter called `edge` to the list of port filters.

```
(STT) filter add switch status
Working with default filter profile.
['fw', 'pingable', 'dynamic', 'slwInit', 'warnCnts', ... , 'status']


(STT) filter add port edge
Working with default filter profile.
['dynamic', 'last_fault', 'corrected_rate', 'uncorrected_rate', ... , 'edge']
```

1.5.6.6.3 `filter remove`

Remove a particular filter property.

```
(STT) filter remove switch dynamic
Working with default filter profile.
['fw', 'pingable', 'slwInit', 'warnCnts', 'uncorrectable_errs', 'critical_errs',  ... ]
```

1.5.6.6.4 `filter save`

Save a filter to a file.

```
(STT) filter save test.txt
Working with 'default' filter profile.
Filters saved to test.txt
(STT) shell cat test.txt


{"_name": "default",
    "_switch": ["fw", "pingable", "slwInit", "warnCnts", "uncorrectable_errs",
                "critical_errs", "degrd_errs", "corrected_errs", "transient_errs",
                "bad_condition_errs", "status"],
    "_port": ["dynamic", "last_fault", "corrected_rate", "uncorrected_rate", "good_rate",
              "corrected_ber", "uncorrected_ber", "tx_show_pct", "rx_show_pct",
              "tx_pause_state", "rx_pause_state", "edge"],
    "_p2p": ["stage", "part_number", "part_length", "calculated_distance",
    "src_egress_a", "src_egress_b", "dst_egress_a", "dst_egress_b"],
    "_cable": ["srcgrp", "dstgrp", "srcsw", "dstsw", "dynamic"],
    "_headshell": ["num_cmis_jacks", "num_jacks_with_alarms", "num_jacks_with_flags",
    "num_jacks_with_warnings", "num_non_cmis3plus_jacks"],
    "_jack": ["DataPathPwrUp", "RX_LOL_flags", "RX_LOS_flags", "TX_LOL_flags",
              "TX_LOS_flags", "cable_end", "cmis_version", "custom_info", "date_code",
              "flat_memory", "length", "part_number", "temperature", "vendor_name",
              "vendor_oui", "voltage"],
    "_software": ["rossw_module", "roscore_module", "rosnic_module",
    "fabric_agent_listening", "hmpa_daemon", "hmpersistor", "hmsc_daemon",
```

```
                "hmfuse_daemon", "hmjtd_daemon", "nginx_listening"],
    "_error_flags": ["eiq_err_flg", "inq_err_flg", "red_err_flg", "ihdr_err_flg",
    "elu_err_flg", "ibuf_err_flg", "ageq_err_flg"]
  }

(STT)
```

1.5.6.6.5 filter load

Load filter from a file.

```
(STT) filter load test.txt
Working with 'default' filter profile.
Filters loaded from test.txt
(STT) filter list
Working with 'default' filter profile.
Active Filter Profile Details:


    default:
    {
        switch: ['fw', 'pingable', 'slwInit', 'warnCnts', 'uncorrectable_errs',
                    'critical_errs', 'degrd_errs', 'corrected_errs', 'transient_errs',
                    'bad_condition_errs', 'status']
        port       : ['dynamic', 'last_fault', 'corrected_rate', 'uncorrected_rate',
                    'good_rate', 'corrected_ber', 'uncorrected_ber', 'tx_show_pct',
                    'rx_show_pct',
         'tx_pause_state', 'rx_pause_state', 'edge']
        cable      : ['srcgrp', 'dstgrp', 'srcsw', 'dstsw', 'dynamic']
        p2p : ['stage', 'part_number', 'part_length', 'calculated_distance',
               'src_egress_a', 'src_egress_b', 'dst_egress_a', 'dst_egress_b']
        headshell  : ['num_cmis_jacks', 'num_jacks_with_alarms', 'num_jacks_with_flags',
                    'num_jacks_with_warnings', 'num_non_cmis3plus_jacks']
        jack       : ['DataPathPwrUp', 'RX_LOL_flags', 'RX_LOS_flags', 'TX_LOL_flags',
                    'TX_LOS_flags', 'cable_end', 'cmis_version', 'custom_info',
                    'date_code', 'flat_memory', 'length', 'part_number', 'temperature',
                    'vendor_name', 'vendor_oui', 'voltage']
    }
```

1.5.6.7 list

```
(STT) help list
  List created topologies or filter profiles
  Usage: list <topology,filter>
```

1.5.6.7.1 list topology, list filters

List topology and filters.

```
  (STT) list topology
    List of topology profiles['default']
  (STT) list filter
    List of filter profiles['default', 'no_filter']
```

1.5.6.8 new

```
(STT) help new
  Create new topologies or filter profiles
  Usage: new <topology,filter> <list of names>
```

1.5.6.8.1 new topology, new filter

Create new topology and filters.

```
(STT) new topology t1
  STT diags log directory -  /root/slingshot_tools/stt_diags_logs/t1
(STT) new filter filter_1 filter_2
  Creating new filter profile 'filter_1'.
  Creating new filter profile 'filter_2'.
```

1.5.6.9 set

```
(STT) help set
Usage: set [-h] [-v] [param] [value]
Set a settable parameter or show current settings of parameters
positional arguments:
  param          parameter to set or view
  value          new value for settable
optional arguments:
```

```
  -h, --help     show this help message and exit
  -v, --verbose  include description of parameters when viewing
```

1.5.6.10 load

```
(STT) help load
    Load information into the active topology from specified file formats.
    Usage: load <topology|p2p> <filename>
```

1.5.6.10.1 load topology

Load topology from a file.

```
(STT) load
    p2p topology
(STT) load topology /root/slingshot_tools/stt/test/inputs/topo1.json
    Working with 'default' topology.JSON
    Validation succeeded
```

1.5.6.11 remove

```
(STT) help remove
    Removes components and links from active topology
    Usage: remove <switch|link|edge> <arguments>
```

1.5.6.12 save

```
(STT) help save
    Save active topology into a file in specified format.
    Usage: save <topology|p2p> <filename> [--overwrite] [--order]

(STT) show active_topology
Current active topology name is: default

(STT) save topology /tmp/sample_topo
Working with 'default' topology.
JSON Validation succeeded
Wrote 'default' topology to /tmp/sample_topo
```

1.5.6.13 show

Show components of the active topology and their properties.

```
(STT) help show

        Shows components of the active topology and their properties
        Usage: show <active_topology,switches,fabric,edge,topology,headshells,p2p..> [--order] [--meta]
        Usage: show switch <ports,jacks..> <switch xname>
        Usage: show hsn_nodes <lldp>
        Usage: show hsn_node <hsn,lldp> <nodes list>
        Usage: show links health
        Usage: show telemetry <LinkErrors| Congestion| Counters| PortErrors|
                               RoutingErrors| HardErrors| Power| Current|  Energy| Rotational|
                               Temperature| Voltage> <count> <switch xname>

        Details
        =======
         show active_topology                                  - Show active topology
         show active_filter                                    - Show active filter
         show output_format                                    - Show current output format
         show switches                                         - Show all switches info
         show switches software                                - Show all switches software info
         show switches health                                  - Show all switches health info
         show flow_timeout <summary, details>                  - Show stuck flow timeout info
         show flow_timeout summary                             - Show all stuck flow summary info
         show flow_timeout details                             - Show all detailed stuck flow info
         show fabric                                           - Show all fabric ports
         show edge                                             - Show all edge ports
         show edge include_hostname                            - Show all edge ports with HSN hostname
         show topology                                         - Show all contents of topology file
         show p2p                                              - Show contents of Point2Point file
         show cables                                           - Show all cables info
         show cables fabric                                    - Show cables info for fabric connections
         show cables edge                                      - Show cables info for edge connections
         show headshells                                       - Show all headshell info
         show switch ports <switch xname>                      - Show all port info of a switch
         show switch jacks <switch xname>                      - Show all jack info of a switch
         show switch software <switch xname>                   - Show all software info of a switch
         show switch error_flags <switch xname>                - Show all error flags of a switch port
         show switch counters <switch xname>/<port xname>      - Show all counters info of a switch port
```

```
show switch perfcounters <switch xname>/<port xname>   - Show all perf counters info of a switch port
show switch failures <switch xname>                    - Shows list of error within a switch
show switch rossw_info <switch xname>                  - Shows rossw_info of each port of switch
show switch csr_data <switch xname>                    - Shows all CSR data of the switch
show version                                           - Show the current Slingshot Tools version
show hsn_nodes                                         - Show hsn interfaces of all compute nodes
show hsn_nodes lldp                                    - Show lldp info of all compute nodes
show hsn_node hsn <node hostname list>                 - Show compute nodes hsn interface info
show hsn_node lldp <node hostname list>                - Show compute nodes lldp info
show links health                                      - Diagnose downed links
show telemetry <metric> <count> <switch xname>         - Show telemetry of components
show FabricTelemetry ODquery=<query>                   - Show Fabric Telemetry events using OData queries
show hsn_traffic ping-all-to-all <nodes_list> <hsn_ip_list> <detailed> - Show all to all connectivity between hsn nodes
show hsn_traffic roce_perf_check_loopback <nodes_list> - Show RoCE Perf benchmark results at compute node port level
```

1.5.6.13.1 show switches

```
(STT) show switches
Working with 'default' topology and 'default' filter profile.
Collecting data using 'check-switches' script.
Collecting data using 'dgrerrstat' script.
Collecting data using 'dgrperfcheck' script.
```

xname	type	snum	gnum	edge_count	fabric_count	uptime	up_ports	flap	checkidle	pktIn	pktOut	drops	dscrds	mcast_found
x3000c0r24	Columbia	0	0	12	4			0		0	0	0	0	True
x5000c1r1	Colorado	0	1	16	30			0		0	0	0	0	True
x5000c1r3	Colorado	1	1	16	30			0		0	0	0	0	True
x5000c1r7	Colorado	3	1	16	28			0		0	0	0	0	True
x5000c3r5	Colorado	6	1	16	28			0		0	0	0	0	True
x5000c3r3	Colorado	5	1	16	28			0		0	0	0	0	True
x5000c3r7	Colorado	7	1	16	28			0		0	0	0	0	True
x5000c3r1	Colorado	4	1	16	28			0		0	0	0	0	True
x5000c1r5	Colorado	2	1	16	28			0		0	0	0	0	True
x3000c1r40	Columbia	0	0	1	1			0		0	0	0	0	False

1.5.6.13.2 show headshells

```
(STT) show headshells

Working with 'default' topology and 'default' filter profile.
Collecting data using 'dgrheadshellstat' script.
```

hostname	num_jacks_with_trans_normal	num_empty_jacks	num_jacks_with_faults	num_jacks_with_trans_absent	num_jacks_with_trans_error
x0c0r0b0	31	1	0	1	0
x0c0r1b0	30	2	0	1	1

1.5.6.13.3 show cables <edge/fabric>

Extract serial ID's of each end-point based on p2p file from switch controllers or compute nodes using appropriate scripts.

The 'serial_ids' column has this information arranged as ", , , ".

If cable is not connected, the serial_ids are marked as "unknown". If end-points are not reachable, the serial_ids are marked as "Connection Failure" and status field would be "Unknown".

On Mountain systems, for edge connections(Examax), there exists no external cables connected between the compute node and the switch. In that case, serial_ids are marked as "NA".

Status for a connection is derived by comparing `serial_id` in expected cabling pairs as (srca,dsta) and (srcb,dstb). Note, the serial_ids for connected cable pairs must be identical.

For fetching serial ID's of HSN cables from compute nodes, SSH is used. Refer to Setting up access to compute nodes for STT.

This command supports two optional parameters "edge" and "fabric". With these options, 'show cables' will display information only specific to a particular link type. By default, it displays data about all types of links.

New fields

The `vendor`, `media_type` and `health` (default) fields and the `part_number` and `state` (non-default) fields give detailed information regarding the cables.

When the Cable is in "Connected" state or "Not Connected" state, it means that the cables will have a singular value of `vendor`, `media_type`, `health`, `state` or `part_number` information.

However, when the cables are in "Wrong Connection" or "Partial" state, the source and the destination connection of the cables may have different values of the above fields. Hence, in this case, these fields will be populated with comma separated values indicating the src cable and dst cable information respectively.

A sample output of the above fields is present in the section "Updated `show cables` output with Vendor and other cable information" below.

```
(STT) show cables
Working with 'default' topology and 'default' filter profile.
Collecting data using 'check-switches' script.
+-----------------+-----------------+-----------------+-----------------+--------+--------------+-----------------------------------------------------
|       srca      |       srcb      |       dsta      |       dstb      |  type  |    status    |                                serial_ids
+-----------------+-----------------+-----------------+-----------------+--------+--------------+-----------------------------------------------------
| x3000c0r24j31p1 | x3000c0r24j31p0 |  x5000c1r1j23p1 |  x5000c1r1j23p0 | fabric |   Connected  |             UH294G00485, UH294G00485, UH294G00485, U
|  x3000c0r24j1p1 |  x3000c0r24j1p0 |  x5000c1r3j23p1 |  x5000c1r3j23p0 | fabric |   Connected  |             UH294G00294, UH294G00294, UH294G00294, U
| x3000c0r24j12p1 | x3000c0r24j12p0 | x3000c0s9b0n0h1 | x3000c0s9b0n0h0 |  edge  |   Connected  |             0616190017, 0616190017, 0616190017, 0616
| x3000c0r24j14p1 | x3000c0r24j14p0 | x3000c0s7b0n0h1 | x3000c0s7b0n0h0 |  edge  |   Connected  |             APF18458035H8J,APF18458035H8J,APF1845803
|  x5000c1r7j17p1 |  x5000c1r7j17p0 |  x5000c3r5j19p0 |  x5000c3r5j19p1 | fabric |   Connected  |             0211190032, 0211190032, 0211190032, 0211
|  x5000c1r3j18p1 |  x5000c1r3j18p0 |  x5000c3r3j17p1 |  x5000c3r3j17p0 | fabric |   Connected  |             0616190145, 0616190145, 0616190145, 0616
.................
.................
| x5000c3r7j102p1 | x5000c3r7j102p0 | x5000c3s2b1n0h0 |  x5000c3s2b1n1h0 | fabric |    Unknown    | Connection Failure, Connection Failure, Connection Fa
| x5000c3r3j103p1 | x5000c3r3j103p0 | x5000c3s3b0n0h0 |  x5000c3s3b0n1h0 | fabric |    Unknown    | Connection Failure, Connection Failure, Connection Fa
+-----------------+-----------------+-----------------+-----------------+--------+--------------+-----------------------------------------------------

(STT) show cables edge

Working with 'default' topology and 'default' filter profile.
WARNING:root:Model string  via Redfish call for switch x1000c3r5 is
Collecting data using 'check-switches' script.
Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.

check-switches             : Start time: 12/13/2021, 15:07:15, End time: 12/13/2021, 15:07:44, Total time: 0:00:28.377884
+-----------------+-----------------+-----------------+-----------------+------+-----------+-----------------------------------------------------
|       srca      |       srcb      |       dsta      |       dstb      | type |   status  |                                serial_ids
+-----------------+-----------------+-----------------+-----------------+------+-----------+-----------------------------------------------------
| x1002c2r5j100p0 | x1002c2r5j100p1 | x1002c2s0b1n1h1 | x1002c2s0b1n0h1 | edge |   Absent  |                             NA,NA,NA,NA
| x1002c2r5j101p0 | x1002c2r5j101p1 | x1002c2s1b1n1h1 | x1002c2s1b1n0h1 | edge |   Absent  |                             NA,NA,NA,NA
| x1002c2r5j104p0 | x1002c2r5j104p1 | x1002c2s4b1n1h1 | x1002c2s4b1n0h1 | edge |   Absent  |                             NA,NA,NA,NA
...
...
|  x1002c1r5j17p0 |  x1002c1r5j17p1 | x7000c0s31b0n0h0 | x7000c0s31b0n0h1 | edge |  Unknown  | MT1904FT06562, MT1904FT06562, Connection Failure, Connectio
|  x1002c0r7j17p0 |  x1002c0r7j17p1 | x7000c0s28b0n0h0 | x7000c0s28b0n0h1 | edge |  Unknown  | MT1947FT00044, MT1947FT00044, Connection Failure, Connectio
| x3000c0r40j16p0 | x3000c0r40j16p1 | x3000c0s9b0n0h0  | x3000c0s9b0n0h1  | edge | Connected |    APF18458035H8V, APF18458035H8V, APF18458035H8V, APF18458
| x3000c0r39j12p0 | x3000c0r39j12p1 | x3000c0s13b2n0h0 | x3000c0s13b4n0h0 | edge |  Unknown  |       unknown, unknown, Connection Failure, Connection Fail
| x3000c0r39j18p0 | x3000c0r39j18p1 | x3000c0s25b0n0h0 | x3000c0s25b0n0h1 | edge | Connected |           0616190054, 0616190054, 0616190054, 0616190054
| x3000c0r39j16p0 | x3000c0r39j16p1 | x3000c0s7b0n0h0  | x3000c0s7b0n0h1  | edge | Connected |    APF18458035H8F, APF18458035H8F, APF18458035H8F, APF18458
| x3000c0r42j16p0 | x3000c0r42j16p1 | x3000c0s13b0n0h0 | x3000c0s13b0n0h1 | edge | Connected |    APF18458035H8M, APF18458035H8M, APF18458035H8M, APF18458
| x3000c0r42j20p0 | x3000c0r42j20p1 | x3000c0s23b0n0h0 | x3000c0s23b0n0h1 | edge | Connected |             141100027, 141100027, 141100027, 141100027
| x3000c0r41j12p0 | x3000c0r41j12p1 | x3000c0s15b2n0h0 | x3000c0s15b4n0h0 | edge |  Unknown  |       unknown, unknown, Connection Failure, Connection Fail
| x3000c0r41j16p0 | x3000c0r41j16p1 | x3000c0s11b0n0h0 | x3000c0s11b0n0h1 | edge | Connected |    APF18458035H91, APF18458035H91, APF18458035H91, APF18458
+-----------------+-----------------+-----------------+-----------------+------+-----------+-----------------------------------------------------

(STT) show cables fabric

Working with 'default' topology and 'default' filter profile.
WARNING:root:Model string  via Redfish call for switch x1000c3r5 is
Collecting data using 'check-switches' script.
check-switches             : Start time: 12/14/2021, 08:27:30, End time: 12/14/2021, 08:27:41, Total time: 0:00:11.829068
+-----------------+-----------------+-----------------+-----------------+--------+-----------+-----------------------------------------------------+-------
|       srca      |       srcb      |       dsta      |       dstb      |  type  |   status  |                      serial_ids                     |      v
+-----------------+-----------------+-----------------+-----------------+--------+-----------+-----------------------------------------------------+-------
|  x1000c0r1j10p0 |  x1000c0r1j10p1 |  x1000c3r3j11p0 |  x1000c3r3j11p1 | fabric | Connected |   0211190005, 0211190005, 0211190005, 0211190005    | TE Con
|  x1000c0r1j11p0 |  x1000c0r1j11p1 |  x1000c3r5j14p0 |  x1000c3r5j14p1 | fabric | Connected |   0211190040, 0211190040, 0211190040, 0211190040    | TE Con
|  x1000c0r1j12p0 |  x1000c0r1j12p1 |  x1000c3r7j11p0 |  x1000c3r7j11p1 | fabric | Connected |   0211190035, 0211190035, 0211190035, 0211190035    | TE Con
......
......
|
|  x1002c7r3j17p0 |  x1002c7r3j17p1 |  x1003c7r3j17p0 |  x1003c7r3j17p1 | fabric | Connected | UH294G00300, UH294G00300, UH294G00300, UH294G00300 |     Hi
+-----------------+-----------------+-----------------+-----------------+--------+-----------+-----------------------------------------------------+-------
```

Interpret output

The following connection information can be derived from the output:

Status	Condition
Connected	All endpoints of the cable are connected correctly

Status	Condition
Not connected	No cables connected
Partial connection	At least one pair of cable endpoints are connected correctly
Wrong connection	Serial ID's in either of cable pairs do not match
Absent	Indicate edge connections on Mountain systems where external cables are not required
Unknown	One of the endpoints is not reachable

Case 1: Connected

The serial_ids match.

```
+----------------+----------------+----------------+----------------+--------+---------+-----------------------------------------
|      srca      |      srcb      |      dsta      |      dstb      |  type  |status   |     serial_ids                          |
+----------------+----------------+----------------+----------------+--------+---------+-----------------------------------------
| x3000c0r24j14p1 | x3000c0r24j14p0 | x3000c0s7b0n0h1  | x3000c0s7b0n0h0  |  edge  |Connected| APF18458035H8J,APF18458035H8J,APF18458035H8J|
```

In this case, serial_ids are populated for all endpoints and serial_ids of srcb (APF18458035H8J) and dstb (APF18458035H8J) and of srca (APF18458035H8J) and dsta (APF18458035H8J). We classify such connections as connected.

Case 2: Partial connection

```
+--------------+--------------+--------------+--------------+------+-------+-----------------------------------------------------
|      srca    |      srcb    |      dsta    |      dstb    | type |status |        serial_ids                                   |
+--------------+--------------+--------------+--------------+------+-- ----+-----------------------------------------------------
|x3000c0r24j14p|x3000c0r24j14p0|x3000c0s7b0n0h1|x3000c0s7b0n0h0| edge |Partial| APF18458035H8J,APF18458035H8J,0616190017,APF18458035H8J|
```

In this case, serial_ids are populated for all endpoints and serial_ids of srcb (APF18458035H8J) and dstb (APF18458035H8J) are the same, but serial_ids for srca (APF18458035H8J) and dsta(0616190017) are not. We classify such connections as partially connected.

For further debugging, we can use `grep` command to filter data based on serial_ids, to get more information on where the other end of the cable is connected to in case of a such cable swap.

Based on using grep for srca(APF18458035H8J) serial id, we notice how one end of edge link is swapped between two compute nodes.

```
(STT) show cables | grep -e APF18458035H8J
| x3000c0r24j12p1 | x3000c0r24j12p0 | x3000c0s7b0n0h1  | x3000c0s9b0n0h0|edge|Partial|0616190017, 0616190017, APF18458035H8J, 0616190017        |
| x3000c0r24j14p1 | x3000c0r24j14p0 | x3000c0s9b0n0h1  | x3000c0s7b0n0h0|edge|Partial|APF18458035H8J, APF18458035H8J, 0616190017, APF18458035H8J|
```

case 3: Wrong connection

```
+----------------+----------------+----------------+----------------+--------+---------------+-----------------------------------------
|      srca      |      srcb      |      dsta      |      dstb      |  type  |     status    |           serial_ids                    |
+----------------+----------------+----------------+----------------+--------+---------------+-----------------------------------------
| x3000c0r24j31p1 | x3000c0r24j31p0 |  x5000c1r1j23p1  | x5000c1r1j23p0  | fabric |Wrong Connection|UH294G00485,UH294G00485,UH294G00294,UH294G00294|
```

In this case, serial_ids of cable pairs, srca(UH294G00485),dsta(UH294G00294) and srcb(UH294G00485),dstb(UH294G00294) do not match, which indicates that cabling is wrong.

For further debugging, we can use 'grep' command to filter data based on serial ids and get more information on how cables ends are connected. Based on using grep for srca(UH294G00485) and dsta(UH294G00294) serial ids, we can see how cable endpoints are swapped between fabric links.

```
(STT) show cables | grep -e UH294G00485 -e UH294G0029
|x3000c0r24j31p1|x3000c0r24j31p0| x5000c1r1j23p1|x5000c1r1j23p0| fabric|Wrong Connection|UH294G00485,UH294G00485,UH294G00294,UH294G00294|
| x3000c0r24j1p1|x3000c0r24j1p0 | x5000c1r1j23p1|x5000c1r1j23p0| fabric|Wrong Connection|UH294G00294,UH294G00294,UH294G00485,UH294G00485|
```

case 4: Absent

```
+----------------+----------------+----------------+----------------+--------+---------------+-----------------------------------------
|      srca      |      srcb      |      dsta      |      dstb      |  type  |     status    |           serial_ids                    |
+----------------+----------------+----------------+----------------+--------+---------------+-----------------------------------------
| x1000c0r3j100p1 | x1000c0r3j100p0 | x1000c0s0b0n0h0  | x1000c0s0b0n1h0 |  edge  |    Absent     |       NA,NA,NA,NA                       |
+----------------+----------------+----------------+----------------+--------+---------------+-----------------------------------------
```

In this case, serial_ids of cable pairs "NA" indicate that there were no external cables that exist between the switch and the compute nodes. This condition occurs on Mountain systems for edge connections which are called Examax connections.

case 5: Unknown

“‘bash +——————+——————+———————+——————+———+—————-+—————————————— ——— | srca | srcb | dsta | dstb | type | status | serial_ids | +——————+——————+———————+—————— –+———+—————-+—————————————————— | x1003c1r5j17p0 | x1003c1r5j17p1 | x7000c0s30b0n0h0 | x7000c0s30b0n0h1 | edge | Unknown | MT1904FT06595, MT1904FT06595, Connection Failure, Connection Failure |

Updated `show cables` output with Vendor and other cable information

As mentioned above, in the below output, the `vendor`, `media_type` and `health` fields are singular for a cable in “Connected” state. For a cable in “Wrong Connection” state, both source and destination values of cable information is displayed.

```
+-------------+-------------+----------------+---------------------------------------------------+------------------+-------------------+--------+
|    srca     |    srcb     |     status     |                    serial_ids                     |      vendor      |    media_type     | health|
+-------------+-------------+----------------+---------------------------------------------------+------------------+-------------------+--------
| x0c0r0j16p1 | x0c0r0j16p0 |   Connected    | UH294G00846,UH294G00846,UH294G00846,UH294G00846 |     Hisense      |      Optical      |   OK   |
| x0c0r0j17p1 | x0c0r0j17p0 |Wrong Connection| UH294G00668,UH294G00668,UH294G00948,UH294G00948 | Hisense, unknown | Optical, Optical | OK, OK |
| x0c0r0j18p1 | x0c0r0j18p0 |Wrong Connection| UH294G00229,UH294G00229,UH294G00641,UH294G00641 | Hisense, unknown | Optical, Optical | OK, OK |
| x0c0r0j19p1 | x0c0r0j19p0 |Wrong Connection| UH294G01026,UH294G01026,UH294G00932,UH294G00932 | Hisense, unknown | Optical, Optical | OK, OK |

* The output in this table has been truncated for readability.
```

1.5.6.13.4 `show edge`

By default, this command displays all edge port information without HSN hostnames. In order to get the HSN hostnames information as well, add ‘include_hostname’ flag to the command.

```
(STT) show edge
Working with 'default' topology and 'default' filter profile.
Collecting data using 'check-switches' script.
Collecting data using 'check-fabric' script.
Collecting data using 'dgrlinkstat' script.
Collecting data using 'dgrperfcheck' script.
+-----------------+------+----------------+----------+--------+-------+---------+-----+-----+---------+-------+-------+---------+---------+
|      xname      | type |      dst       | hostname | status | ptype | subtype | mtu | mac | mactype | speed | media | mcast_a | mcast_b |
+-----------------+------+----------------+----------+--------+-------+---------+-----+-----+---------+-------+-------+---------+---------+
| x3000c0r24j12p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   480   |
| x3000c0r24j12p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   497   |
| x3000c0r24j14p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   485   |
| x3000c0r24j14p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   2730  |
|  x3000c0r24j8p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   771   |
|  x3000c0r24j8p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   782   |
| x3000c0r24j10p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   478   |
| x3000c0r24j10p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   1115  |
|  x3000c0r24j4p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   1120  |
|  x3000c0r24j4p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   1143  |
|  x3000c0r24j6p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   745   |
|  x3000c0r24j6p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   725   |
| x5000c1r1j103p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r1j103p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r1j101p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    18   |
| x5000c1r1j101p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    18   |
| x5000c1r1j105p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r1j105p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r1j107p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
                                    .................
                                    .................
| x5000c3r1j100p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   833   |
| x5000c3r1j104p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   740   |
| x5000c3r1j104p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c3r1j106p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   749   |
| x5000c3r1j106p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   759   |
| x5000c1r5j103p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r5j103p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r5j101p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    18   |
| x5000c1r5j101p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    18   |
| x5000c1r5j105p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r5j105p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r5j107p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r5j107p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r5j102p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    12   |
| x5000c1r5j102p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    12   |
| x5000c1r5j100p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    12   |
| x5000c1r5j100p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    12   |
| x5000c1r5j104p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r5j104p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r5j106p1 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x5000c1r5j106p0 | edge |      None      |          | False  |       |         |  0  |     |         |       |       |    0    |   703   |
| x3000c1r40j13p0 | edge | x9999c1r0n0j0  |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
+-----------------+------+----------------+----------+--------+-------+---------+-----+-----+---------+-------+-------+---------+---------+

(STT) show edge include_hostname
+-----------------+------+-----------------+----------+----------+--------+---------+---------+-----+-----------------+---------+-----------+--------
```

```
| x3000c0r39j16p0 | edge | x3000c0s7b0n0h0  | ncn-w001 |    31    |  True  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G | electrical |    0
| x3000c0r39j16p1 | edge | x3000c0s7b0n0h1  | ncn-w001 |    30    |  True  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G | electrical |    0
| x3000c0r39j18p0 | edge | x3000c0s25b0n0h0 | ncn-w005 |    47    |  True  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G | electrical |    0
| x3000c0r39j18p1 | edge | x3000c0s25b0n0h1 | ncn-w005 |    46    |  True  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G | electrical |    0
| x3000c0r40j16p0 | edge | x3000c0s9b0n0h0  | ncn-w002 |    31    |  True  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G | electrical |    0
| x3000c0r40j16p1 | edge | x3000c0s9b0n0h1  | ncn-w002 |    30    |  True  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G | electrical |    0
| x3000c0r41j12p0 | edge | x3000c0s15b2n0h0 | unknown  |    26    | False  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G |            |    0
| x3000c0r41j12p1 | edge | x3000c0s15b4n0h0 | unknown  |    27    | False  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G |            |    0
| x3000c0r41j16p0 | edge | x3000c0s11b0n0h0 | ncn-w003 |    31    |  True  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G | electrical |    0
| x3000c0r41j16p1 | edge | x3000c0s11b0n0h1 | ncn-w003 |    30    |  True  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G | electrical |    0
| x3000c0r42j16p0 | edge | x3000c0s13b0n0h0 | ncn-w004 |    31    |  True  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G | electrical |    0
| x3000c0r42j16p1 | edge | x3000c0s13b0n0h1 | ncn-w004 |    30    |  True  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G | electrical |    0
| x3000c0r42j20p0 | edge | x3000c0s23b0n0h0 |  uan01   |    60    |  True  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G | electrical |    0
| x3000c0r42j20p1 | edge | x3000c0s23b0n0h1 |  uan01   |    61    |  True  | Ethernet |   Edge   |  0  | 02:00:00:00:00:00 | BJ_100G | electrical |    0
+-----------------+------+------------------+----------+----------+--------+----------+----------+-----+-------------------+---------+------------+-------
```

1.5.6.13.5 `show fabric`

```
(STT) show fabric
Working with 'default' topology and 'default' filter profile.
Collecting data using 'check-switches' script.
Collecting data using 'check-fabric' script.
Collecting data using 'dgrlinkstat' script.
Collecting data using 'dgrperfcheck' script.
+-----------------+--------+-----+----------+--------+-------+---------+-----+-----+---------+-------+-------+---------+---------+
|      xname      |  type  | dst | hostname | status | ptype | subtype | mtu | mac | mactype | speed | media | mcast_a | mcast_b |
+-----------------+--------+-----+----------+--------+-------+---------+-----+-----+---------+-------+-------+---------+---------+
| x3000c0r24j31p1 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |  116714 |
| x3000c0r24j31p0 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x3000c0r24j1p1 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |  41402  |
|  x3000c0r24j1p0 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j23p1 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |  10094  |
|  x5000c1r1j23p0 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j18p1 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j18p0 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j3p1  | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |   783   |
|  x5000c1r1j3p0  | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j21p1 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j21p0 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j6p1  | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |  18854  |
|  x5000c1r1j6p0  | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j20p1 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j20p0 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j19p1 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j19p0 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j8p1  | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |  23120  |
|  x5000c1r1j8p0  | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r1j5p1  | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |   5518  |
|  x5000c1r1j5p0  | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
                                  ..................
                                  ..................
|  x5000c1r5j22p0 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r5j4p1  | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |   5516  |
|  x5000c1r5j4p0  | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r5j18p1 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r5j18p0 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r5j8p1  | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |  23119  |
|  x5000c1r5j8p0  | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r5j16p1 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
|  x5000c1r5j16p0 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
| x3000c1r40j16p1 | fabric |     |          | False  |       |         |  0  |     |         |       |       |    0    |    0    |
+-----------------+--------+-----+----------+--------+-------+---------+-----+-----+---------+-------+-------+---------+---------+
```

1.5.6.13.6 `show switches flow_timeout summary`

Flow channels (FCs) track every packet in the system. A packet enters on an edge port and then it is assigned to a FC to take it through the whole fabric. An Edge Flow Channel Table (EFCT) at the switch port traces packet movement through the system.

`dgrflowdebug.py` runs on the switch and executes table tests at the switch port EFCTs to find timed out flow channel table entries. It follows entries from one to the next to trace stuck flows through the fabric and then prints either a summary of all stuck flow paths, a detailed trace of a single stuck flow path, or detailed traces of all stuck flow paths.

Also see `show switches flow_timeout details`.

```
(STT) show switches flow_timeout summary

Working with 'default' topology and 'default' filter profile.
Checking dgrflowdebug.py version on switches...
Looking for EFCT timeouts at edge ports...
Found 2 EFCT timeouts
Following flows...
```

```
Splitting mcast flows into individual flow paths...
Removing duplicate flow paths...
Found 2 unique flow paths

Full List of Unique Stuck Flows
===============================
--ingress_edge--> x3000c0r42b0[EFCT_12] --> x3000c0r42b0[IFCT_12] --ERROR->
  Errors:
    IFCT timeout indicates an entry should be found in x3000c0r42b0[OFCT_48] for source_flow_id 460 and source_port 12. None found!

--ingress_edge--> x3000c0r42b0[EFCT_13] --> x3000c0r42b0[IFCT_13] --ERROR->
  Errors:
    IFCT timeout indicates an entry should be found in x3000c0r42b0[OFCT_48] for source_flow_id 461 and source_port 13. None found!

IFCTs where flow tracing ended early
====================================
  x3000c0r42b0: 12 13

OFCTs where timeouts were missing
=================================
  x3000c0r42b0: 48(x2)
```

1.5.6.13.7 show switches flow_timeout details

```
show switches flow_timeout details

Working with 'default' topology and 'default' filter profile.
Checking dgrflowdebug.py version on switches...
Looking for EFCT timeouts at edge ports...
Found 2 EFCT timeouts
Following flows...

Detailed Stuck Flow Trace
=========================
--ingress_edge-->
x3000c0r42b0 -- EFCT 12 -- flow_id: 460
  FGFC: 0x60000 (393216)
  SQ: 0x0 (0)
  VC: 0x0 (0)
  SOURCE_PORT: 0x0 (0)
  SOURCE_FLOW_ID: 0x0 (0) - unmapped: 0x0 (0)
  FLOW_SEPARATOR: 0x40 (64)
  DESTINATION: 0x61 (97)
  FTAG: 0x0 (0)
  PCP: 0x0 (0)
  TIMEOUT: 0x1 (1)
  ACK_FLOW: 0x2b (43)
  DATA_FLOW: 0x30 (48)
-->
x3000c0r42b0 -- IFCT 12 -- flow_id: 460
  INJECTION_LIMIT_ID: 0x0 (0)
  EP_CONGESTION: 0x1 (1)
  FC_STATE: 0x1 (1)
  MC_AND_UC_DATA: 0x28 (40)
  PATH_TO: 0x1 (1)
  BLK_REASON: 0x0 (0)
  FQ_EMPTY: 0x1 (1)
  TIMEOUT: 0x3 (3)
  EPOCH: 0x0 (0)
  FTAG: 0x0 (0)
  FOLLOW_NXT_VC: 0x0 (0)
  FOLLOW_PORT: 0x30 (48)
  WAKEUP_TIME: 0x364c45 (3558469)
  MULTICAST: 0x0 (0)
  UM: 0x0 (0)
  DM: 0x0 (0)
  ACK_FLOW: 0x2b (43)
  DATA_FLOW: 0x30 (48)
  NEXT_DATA_FLOW: 0x30 (48)
--ERROR-->
  Errors:
    IFCT timeout indicates an entry should be found in x3000c0r42b0[OFCT_48] for source_flow_id 460 and source_port 12. None found!

--ingress_edge-->
x3000c0r42b0 -- EFCT 13 -- flow_id: 461
  FGFC: 0x60000 (393216)
  SQ: 0x0 (0)
  VC: 0x0 (0)
  SOURCE_PORT: 0x0 (0)
  SOURCE_FLOW_ID: 0x0 (0) - unmapped: 0x0 (0)
  FLOW_SEPARATOR: 0x40 (64)
  DESTINATION: 0x61 (97)
  FTAG: 0x0 (0)
  PCP: 0x0 (0)
```

```
  TIMEOUT: 0x1 (1)
  ACK_FLOW: 0x2b (43)
  DATA_FLOW: 0x30 (48)
-->
x3000c0r42b0 -- IFCT 13 -- flow_id: 461
  INJECTION_LIMIT_ID: 0x0 (0)
  EP_CONGESTION: 0x1 (1)
  FC_STATE: 0x1 (1)
  MC_AND_UC_DATA: 0x28 (40)
  PATH_TO: 0x1 (1)
  BLK_REASON: 0x0 (0)
  FQ_EMPTY: 0x1 (1)
  TIMEOUT: 0x3 (3)
  EPOCH: 0x0 (0)
  FTAG: 0x0 (0)
  FOLLOW_NXT_VC: 0x0 (0)
  FOLLOW_PORT: 0x30 (48)
  WAKEUP_TIME: 0x1e30d (123661)
  MULTICAST: 0x0 (0)
  UM: 0x0 (0)
  DM: 0x0 (0)
  ACK_FLOW: 0x2b (43)
  DATA_FLOW: 0x30 (48)
  NEXT_DATA_FLOW: 0x30 (48)
--ERROR-->
  Errors:
    IFCT timeout indicates an entry should be found in x3000c0r42b0[OFCT_48] for source_flow_id 461 and source_port 13. None found!
```

1.5.6.13.8 show switch ports <xname>

```
(STT) show switch ports x5000c3r3
Working with 'default' topology and 'default' filter profile.
Collecting data using 'check-switches' script.
Collecting data using 'check-fabric' script.
Collecting data using 'dgrlinkstat' script.
Collecting data using 'dgrperfcheck' script.
```

xname	type	dst	hostname	status	ptype	subtype	mtu	mac	mactype	speed	media	mcast_a	mcast_b
x5000c3r3j17p1	fabric			False			0					0	0
x5000c3r3j17p0	fabric			False			0					0	0
x5000c3r3j7p1	fabric			False			0					0	3832
x5000c3r3j7p0	fabric			False			0					0	0
x5000c3r3j4p1	fabric			False			0					0	12497
x5000c3r3j4p0	fabric			False			0					0	0
x5000c3r3j22p1	fabric			False			0					0	0
x5000c3r3j22p0	fabric			False			0					0	0
x5000c3r3j8p1	fabric			False			0					0	631
x5000c3r3j8p0	fabric			False			0					0	0
x5000c3r3j18p1	fabric			False			0					0	0
x5000c3r3j18p0	fabric			False			0					0	0
...													
x5000c3r3j106p0	edge	None		False			0					0	713
x5000c3r3j107p1	edge	None		False			0					0	0
x5000c3r3j107p0	edge	None		False			0					0	0

1.5.6.13.9 show switch jacks <xname>

```
(STT) show switch jacks x0c0r0
Working with 'default' topology and 'default' filter profile.
```

jack	chfs_transceiver_status	connector	identifier	media_type	module_state	serial_number
1	normal		QSFP+ or later			
2	normal		QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)			
3	normal	Reserved	QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)	Reserved		1106190331
4	normal		QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)			
5	normal		QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)			
6	normal		QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)			
7	normal		QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)			
8	normal		QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)			
9	normal		QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)			
10	normal		QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)			
11	normal		QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)			
12	normal		QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)			
13	normal		QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)			
14	absent					
15	normal		QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)			
16	normal	Reserved	QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)	Reserved	ModuleReady	UH294G00846
17	normal	Reserved	QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)	Reserved	ModuleReady	UH294G00668
18	normal	Reserved	QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)	Reserved	ModuleReady	UH294G00229
19	normal	Reserved	QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628)	Reserved	ModuleReady	UH294G01026

```
|  20  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleReady   |  UH294G00747  |
|  21  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleLowPwr  |  UH294G00670  |
|  22  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleReady   |  UH294G00793  |
|  23  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleReady   |  UH294G00895  |
|  24  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleReady   |  UH294G00593  |
|  25  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleReady   |  UH294G00641  |
|  26  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleReady   |  UH294G00948  |
|  27  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleReady   |  UH294G00499  |
|  28  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleReady   |  UH294G00932  |
|  29  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleReady   |  UH294G00699  |
|  30  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleReady   |  UH294G00924  |
|  31  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleReady   |  UH294G00606  |
|  32  |          normal          |  Reserved | QSFP-DD Double Density 8X Pluggable Transceiver (INF-8628) |  Reserved  | ModuleReady   |  UH294G00544  |
+------+--------------------------+-----------+------------------------------------------------------------+------------+---------------+---------------+
```

1.5.6.13.10 show switch error_flags <xname>

```
(STT) show switch error_flags x5000c3r3
Working with 'default' topology and 'default' filter profile.
Collecting data using 'dgrerrstat' script.
```

xname	ofct_err_flg	obuf_err_flg	ifct_err_flg	eeg_err_flg	cftx_err_flg	frf_err_flg	pml_err_flg	xbar_err_flg
x5000c3r3j17p1	0	0	0	0	0	0	0	0
x5000c3r3j17p0	0	0	0	0	0	0	0	0
x5000c3r3j7p1	0	0	0	0	0	0	0	0
x5000c3r3j7p0	0	0	0	0	0	0	0	0
x5000c3r3j4p1	0	0	0	0	0	0	0	0
x5000c3r3j4p0	0	0	0	0	0	0	0	0
x5000c3r3j22p1	0	0	0	0	0	0	0	0
x5000c3r3j22p0	0	0	0	0	0	0	0	0
x5000c3r3j8p1	0	0	0	0	0	0	0	0
x5000c3r3j8p0	0	0	0	0	0	0	0	0
x5000c3r3j18p1	0	0	0	0	0	0	0	0
x5000c3r3j18p0	0	0	0	0	0	0	0	0
x5000c3r3j20p1	0	0	0	0	0	0	0	0
x5000c3r3j20p0	0	0	0	0	0	0	0	0
x5000c3r3j19p1	0	0	0	0	0	0	0	0
x5000c3r3j19p0	0	0	0	0	0	0	0	0
x5000c3r3j5p1	0	0	0	0	0	0	0	0
x5000c3r3j5p0	0	0	0	0	0	0	0	0
x5000c3r3j21p1	0	0	0	0	0	0	0	0
x5000c3r3j21p0	0	0	0	0	0	0	0	0
x5000c3r3j3p1	0	0	0	0	0	0	0	0
x5000c3r3j3p0	0	0	0	0	0	0	0	0
x5000c3r3j6p1	0	0	0	0	0	0	0	0
x5000c3r3j6p0	0	0	0	0	0	0	0	0
x5000c3r3j16p1	0	0	0	0	0	0	0	0
x5000c3r3j16p0	0	0	0	0	0	0	0	0
x5000c3r3j10p1	0	0	0	0	0	0	0	0
x5000c3r3j10p0	0	0	0	0	0	0	0	0
x5000c3r3j100p1	0	0	0	0	0	0	0	0
x5000c3r3j100p0	0	0	0	0	0	0	0	0
x5000c3r3j101p1	0	0	0	0	0	0	0	0
x5000c3r3j101p0	0	0	0	0	0	0	0	0
x5000c3r3j102p1	0	0	0	0	0	0	0	0
x5000c3r3j102p0	0	0	0	0	0	0	0	0
x5000c3r3j103p1	0	0	0	0	0	0	0	0
x5000c3r3j103p0	0	0	0	0	0	0	0	0
x5000c3r3j104p1	0	0	0	0	0	0	0	0
x5000c3r3j104p0	0	0	0	0	0	0	0	0
x5000c3r3j105p1	0	0	0	0	0	0	0	0
x5000c3r3j105p0	0	0	0	0	0	0	0	0
x5000c3r3j106p1	0	0	0	0	0	0	0	0
x5000c3r3j106p0	0	0	0	0	0	0	0	0
x5000c3r3j107p1	0	0	0	0	0	0	0	0
x5000c3r3j107p0	0	0	0	0	0	0	0	0

1.5.6.13.11 show switch counters <xname>

Either switch xname or port xname can be passed to view the switch/port counters information.

```
(STT) show switch counters x3000c0r39
Working with 'default' topology and 'default' filter profile.
{
"0": {
"ageq_histogram_00": 12693711959829,
"ageq_histogram_16": 12693711995157,
"ibuf_gnt_empty": 12693711273749,
"ibuf_op_idle": 12693711278145,
"ofct_cycles_n_flows_allocated_0": 12693711808345
},
"1": {
```

```
"ageq_histogram_00": 12693713516949,
"ageq_histogram_16": 12693713552269,
"ibuf_gnt_empty": 12693712830661,
"ibuf_op_idle": 12693712835045,
"ofct_cycles_n_flows_allocated_0": 12693713365413
},
"10": {
"ageq_histogram_00": 12693727377841,
"ageq_histogram_16": 12693727413189,
"ibuf_gnt_empty": 12693726691481,
"ibuf_op_idle": 12693726695913,
"ofct_cycles_n_flows_allocated_0": 12693727226297
},
"11": {
"ageq_histogram_00": 12693728908421,
"ageq_histogram_16": 12693728943773,
"ibuf_gnt_empty": 12693728222089,
"ibuf_op_idle": 12693728226525,
"ofct_cycles_n_flows_allocated_0": 12693728756905
},
"12": {
"ageq_histogram_00": 12693730460057,
"ageq_histogram_16": 12693730495457,
"ibuf_gnt_empty": 12693729773781,
"ibuf_op_idle": 12693729778177,
"ofct_cycles_n_flows_allocated_0": 12693730308517
},
```


1.5.6.13.12 show switch counters <port_xname>

```
(STT) show switch counters x3000c0r24j14p1

Working with 'default' topology and 'default' filter profile.
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/switch_topo_x3000c0r24
JSON Validation succeeded
JSON Validation succeeded
Collecting data using 'dgrerrstat' script.
dgrerrstat  :  Start time: 05/31/2021, 04:24:15 , End time: 05/31/2021, 04:24:20
13 Rosetta ASIC counters:
```


```
{
    "ageq_gnt_sq_07": 1004803,
    "ageq_gnt_vc_00": 1004803,
    "ageq_histogram_00": 2553032485814229,
    "ageq_histogram_16": 2553032485848164,
    "ageq_histogram_17": 1405,
    "eeg_itf_sts_rx_ok": 67216,
    "eeg_sts_tx_ok": 76962988,
    "eeg_tx_ok_1024_to_2047": 8,
    "eeg_tx_ok_128_to_255": 144,
    "eeg_tx_ok_2048_to_4095": 76,
    "eeg_tx_ok_256_to_511": 4074,
    "eeg_tx_ok_4096_to_8191": 205,
    "eeg_tx_ok_512_to_1023": 551,
    "eeg_tx_ok_64": 872953,
    "eeg_tx_ok_65_to_127": 227815,
    "eeg_tx_ok_broadcast": 875838,
    "eeg_tx_ok_ieee": 1105826,
    "eeg_tx_ok_multicast": 1101581,
    "eiq_efct_tot": 906,
    "eiq_elu_tot": 906,
    "eiq_inq_tot": 906,
    "eiq_sop_tot": 906,
    "elu_filtered1": 906,
    "elu_lu_in_tot1": 906,
    "elu_rslt_tot1": 906,
    "frf_mcast_drop_edge_cntr": 820,
    "frf_mcast_drop_non_op_cntr": 2,
    "ibuf_gnt_empty": 2553032485105951,
    "ibuf_gnt_last": 906,
    "ibuf_gnt_mult": 9018,
    "ibuf_gnt_tot": 9104,
    "ibuf_op_blk": 6376,
    "ibuf_op_busy": 18208,
    "ibuf_op_idle": 2553032485114129,
    "ibuf_pkt_tot": 906,
    "ibuf_wr_req": 906,
    "ifct_discard_acks_b": 2151,
    "ifct_epc2_acks_a": 86,
    "ifct_frf_route_used_a": 86,
    "ifct_frf_route_used_b": 820,
    "ifct_headroom_acks_b": 1,
    "ifct_high_latency_b": 1,
    "ifct_injected_data_no_data": 86,
    "ifct_low_latency_a": 86,
    "ifct_low_latency_b": 819,
```

```
    "ifct_multicast_frames_b": 820,
    "ifct_new_flow": 86,
    "ifct_normal_acks_b": 6867,
    "ifct_not_blocked_a": 86,
    "ifct_not_blocked_b": 820,
    "ifct_ordered_frames_a": 86,
    "ifct_ordered_frames_b": 820,
    "ifct_queue_empty": 86,
    "ifct_queue_extent_a_empty": 86,
    "ifct_queue_extent_b_empty": 820,
    "ifct_srcdst_to_edge_a": 86,
    "ifct_unicast_frames_a": 86,
    "inq_ibuf_gnt": 9104,
    "inq_mcast_gnt": 9018,
    "inq_mcast_req": 9018,
    "inq_mcast_rte": 820,
    "inq_rcvd_hdr": 906,
    "inq_ucast_gnt": 86,
    "inq_ucast_req": 86,
    "inq_ucast_rte": 86,
    "itf_rx_ok_65_to_127": 906,
    "itf_rx_ok_ieee": 906,
    "itf_rx_ok_multicast": 906,
    "llr_rx_ok_lossy": 906,
    "llr_tx_ok_lossy": 1105826,
    "obuf_efct_ack": 1004803,
    "obuf_efifo_flt": 2079309,
    "obuf_efifo_frm": 1004803,
    "obuf_efifo_stl": 10696,
    "obuf_ln_stl": 10696,
    "obuf_mb_flt": 202046,
    "obuf_mb_frm": 101023,
    "obuf_per_sq_flits_01": 200378,
    "obuf_per_sq_flits_03": 1668,
    "obuf_per_sq_flits_07": 2079309,
    "obuf_pfg_cdt_stl": 808184,
    "obuf_rcvd_flt": 2079309,
    "obuf_rcvd_frm": 1004803,
    "obuf_sent_flt": 2281355,
    "obuf_sent_frm": 1105826,
    "ofct_acks_forwarded": 906,
    "ofct_allocations_a": 86,
    "ofct_cycles_n_flows_allocated_0": 2553032484461663,
    "ofct_cycles_n_flows_allocated_1": 1201046,
    "ofct_deallocations": 906,
    "ofct_final_normal_acks": 1004803,
    "ofct_flow_extent_histogram_00": 906,
    "ofct_hdrs_to_inq": 906,
    "ofct_match_operations_a": 86,
    "ofct_no_of_grants": 1004803
}
```


1.5.6.13.13 `show switch perfcounters <xname>`

Either switch xname or port xname can be passed to view the switch/port perfcounters information.


```
    "port 1": {
        "IFCT_MULTICAST_FRAMES_A": 0,
        "IFCT_MULTICAST_FRAMES_B": 0,
        "NUM_DROPS_DISCARDS_FULL_BUFFERS": 0,
        "NUM_IFCT_NOT_ZERO": 0,
        "ROSEVC_AGEQ_HWM_DISCARD_ERR": 0,
        "ROSEVC_AGEQ_HWM_DISCARD_INFO": 0,
        "ROSEVC_AGEQ_SIZE_DISCARD_ERR": 0,
        "ROSEVC_AGEQ_TO_DISCARD_ERR": 0,
        "ROSEVC_AGEQ_TO_DISCARD_INFO": 0,
        "ROSEVC_EEG_TRAP_DROP_EOPB": 0,
        "ROSEVC_EEG_TRAP_DROP_FULL": 0,
        "ROSEVC_IBUF_IBUF_FULL": 0,
        "ROSEVC_IFCT_ALWAYS_DISCARD": 0,
        "ROSEVC_IFCT_ALWAYS_FQ_DROP_A": 0,
        "ROSEVC_IFCT_ALWAYS_FQ_DROP_B": 0,
        "ROSEVC_IFCT_ALWAYS_PCP_DROP_A": 0,
        "ROSEVC_IFCT_ALWAYS_PCP_DROP_B": 0,
        "ROSEVC_IFCT_FC_DQ_DISCARD": 0,
        "ROSEVC_IFCT_RANDOM_FQ_DROP_A": 0,
        "ROSEVC_IFCT_RANDOM_FQ_DROP_B": 0,
        "ROSEVC_IFCT_RANDOM_PCP_DROP_A": 0,
        "ROSEVC_IFCT_RANDOM_PCP_DROP_B": 0,
        "ROSEVC_INQ_ECN_DROPPED": 0,
        "ROSEVC_LLR_TX_DISCARD": 0,
        "ROSEVC_OBUF_DROP_FLT": 0,
        "ROSEVC_OBUF_DROP_FRM": 0,
        "ROSEVC_OFCT_DISCARD_FORWARD_BLOCK": 0,
        "ROSEVC_OFCT_EPC1_ACKS_DROPPED": 0,
```

```
        "ROSEVC_OFCT_FABRIC_DISCARD_ACKS": 0,
        "ROSEVC_OFCT_FINAL_DISCARD_ACKS": 0,
        "ROSEVC_RED_DATA_DROP": 0
    },

    "port 61": {
        "IFCT_MULTICAST_FRAMES_A": 0,
        "IFCT_MULTICAST_FRAMES_B": 0,
        ...
    },
    "port 62": {
        "IFCT_MULTICAST_FRAMES_A": 0,
        "IFCT_MULTICAST_FRAMES_B": 0,
        ...
    },
    "port 63": {
        "IFCT_MULTICAST_FRAMES_A": 0,
        "IFCT_MULTICAST_FRAMES_B": 465183,
        ...
    },
    "port 7": {
        "IFCT_MULTICAST_FRAMES_A": 0,
        "IFCT_MULTICAST_FRAMES_B": 1074,
        ...
    },
    "port 8": {
        "IFCT_MULTICAST_FRAMES_A": 0,
        "IFCT_MULTICAST_FRAMES_B": 893,
        ...
    },
    "port 9": {
        "IFCT_MULTICAST_FRAMES_A": 0,
        "IFCT_MULTICAST_FRAMES_B": 972,
        ...
    },
    ...
}
```


1.5.6.13.14 show switch perfcounters <port_xname>

```
(STT) show switch perfcounters x3000c0r24j1p0

Working with 'default' topology and 'default' filter profile.
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/switch_topo_x3000c0r24
JSON Validation succeeded
JSON Validation succeeded
Collecting data using 'dgrperfcheck' script.
dgrperfcheck  :  Start time: 06/03/2021, 01:24:15 , End time: 06/03/2021, 01:24:18
port 1 Rosetta ASIC perf counters:
```


```
{
    "IFCT_MULTICAST_FRAMES_A": 0,
    "IFCT_MULTICAST_FRAMES_B": 0,
    "NUM_DROPS_DISCARDS_FULL_BUFFERS": 0,
    "NUM_IFCT_NOT_ZERO": 59,
    "ROSEVC_AGEQ_HWM_DISCARD_ERR": 0,
    "ROSEVC_AGEQ_HWM_DISCARD_INFO": 0,
    "ROSEVC_AGEQ_SIZE_DISCARD_ERR": 0,
    "ROSEVC_AGEQ_TO_DISCARD_ERR": 0,
    "ROSEVC_AGEQ_TO_DISCARD_INFO": 0,
    "ROSEVC_EEG_TRAP_DROP_EOPB": 0,
    "ROSEVC_EEG_TRAP_DROP_FULL": 0,
    "ROSEVC_IBUF_IBUF_FULL": 0,
    "ROSEVC_IFCT_ALWAYS_DISCARD": 0,
    "ROSEVC_IFCT_ALWAYS_FQ_DROP_A": 0,
    "ROSEVC_IFCT_ALWAYS_FQ_DROP_B": 0,
    "ROSEVC_IFCT_ALWAYS_PCP_DROP_A": 0,
    "ROSEVC_IFCT_ALWAYS_PCP_DROP_B": 0,
    "ROSEVC_IFCT_FC_DQ_DISCARD": 0,
    "ROSEVC_IFCT_RANDOM_FQ_DROP_A": 0,
    "ROSEVC_IFCT_RANDOM_FQ_DROP_B": 0,
    "ROSEVC_IFCT_RANDOM_PCP_DROP_A": 0,
    "ROSEVC_IFCT_RANDOM_PCP_DROP_B": 0,
    "ROSEVC_INQ_ECN_DROPPED": 0,
    "ROSEVC_LLR_TX_DISCARD": 0,
    "ROSEVC_OBUF_DROP_FLT": 0,
    "ROSEVC_OBUF_DROP_FRM": 0,
    "ROSEVC_OFCT_DISCARD_FORWARD_BLOCK": 0,
    "ROSEVC_OFCT_EPC1_ACKS_DROPPED": 0,
    "ROSEVC_OFCT_FABRIC_DISCARD_ACKS": 0,
    "ROSEVC_OFCT_FINAL_DISCARD_ACKS": 0,
    "ROSEVC_PCS_UNCORRECTED_CW": 6,
    "ROSEVC_RED_DATA_DROP": 0
}
```

1.5.6.13.15 show links health

This command analyses the downed links and suggests a diagnostic action code based on information fetched from switches and edge nodes. Note: It will clear the STT data model cache, hence any subsequent command will need to re-populate it.

```
(STT) show links health

Working with 'default' topology and 'default' filter profile.
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/dl_subtopo
JSON Validation succeeded
JSON Validation succeeded
Collecting data using 'check-switches' script.
Collecting data using 'check-fabric' script.
Collecting data using 'dgrheadshellstat' script.
Collecting data using 'dgrvalidatesyscfg' script.
Collecting data using 'check_hsn_nics_config' script.
Collecting data using 'dgrlinkstat' script.

Displaying the Downed Links and required diagnostics action for each link:
Querying downed links\' link partners...
+------+------------+----------+---------------+-----------------+---------------+------------+------------+-------------+---------------+
| type |  rosswinfo | sC firmW | sw_medtype-pw |    hdsh state   | lp /etc/hosts | node power | hsn iface  | action_code |lp action_code|
+------+------------+----------+---------------+-----------------+---------------+------------+------------+-------------+---------------+
| Edge | tpml d S l | 1.3.367  | electrical-N/A | ExaMAX connector |   nid001014   | rfish fail | ssh failed |     ros4    |   lp_unresp   |
| Edge | tpml d S l | 1.3.367  | electrical-N/A | ExaMAX connector |   nid001013   | rfish fail | ssh failed |     ros4    |   lp_unresp   |
| Edge | tpml d S l | 1.3.367  | electrical-N/A | ExaMAX connector |   nid001018   | rfish fail | ssh failed |     ros4    |   lp_unresp   |
| Edge | tpml d S l | 1.3.367  | electrical-N/A | ExaMAX connector |   nid001017   | rfish fail | ssh failed |     ros4    |   lp_unresp   |
| Edge | tpml d S l | 1.3.367  | electrical-N/A | ExaMAX connector |   nid001022   | rfish fail | ssh failed |     ros4    |   lp_unresp   |
| Edge | tpml d S l | 1.3.367  | electrical-N/A | ExaMAX connector |   nid001063   | rfish fail | ssh failed |     ros4    |   lp_unresp   |
+------+------------+----------+---------------+-----------------+---------------+------------+------------+-------------+---------------+


To get details about action codes and table header info, use following commands:
         'run linkdbg -a <action_code>'
         'run linkdbg -d'

(STT)
```

1.5.6.13.16 show telemetry Congestion 10 x3000c0r4*

NOTE: For the `show telemetry` command,

are optional parameters. As an alternative, the above command can be run as:

```
show telemetry Congestion 100 x0c0*-->  [location matched to x0c0*]
show telemetry Congestion * x0c0*  -->  [count defaults to 10]
show telemetry Congestion          -->  [count defaults to 10 & location defaults to all switches]

Working with 'default' topology and 'default' filter profile.
Query URL: http://localhost:8000/metrics?$filter=PhysicalContext%20eq%20%27Congestion*%27%20and%20Location%20eq%20%27x3000c0r4*%27&$top=10&$expand&limit=3
+--------------------------+-----------------+-----------------+---------------+-------+----------+-----------------------+----------+
| Timestamp                | Location        | PhysicalContext | ParentalIndex | Index | SubIndex | DeviceSpecificContext | Value    |
+--------------------------+-----------------+-----------------+---------------+-------+----------+-----------------------+----------+
| 2021-07-30T07:09:18.813Z | x3000c0r42j27p0 | Congestion.rxBW | 0             | 0     | 52       | edge                  | 6.724703 |
| 2021-07-30T07:09:18.813Z | x3000c0r42j27p1 | Congestion.txBW | 0             | 0     | 53       | edge                  | 6.058891 |
| 2021-07-30T07:09:18.814Z | x3000c0r42j17p0 | Congestion.rxBW | 0             | 0     | 62       | edge                  | 6.724815 |
| 2021-07-30T07:09:18.814Z | x3000c0r42j17p0 | Congestion.txBW | 0             | 0     | 62       | edge                  | 6.058992 |
| 2021-07-30T07:09:18.814Z | x3000c0r42j17p1 | Congestion.rxBW | 0             | 0     | 63       | edge                  | 6.724815 |
| 2021-07-30T07:09:18.814Z | x3000c0r42j17p1 | Congestion.txBW | 0             | 0     | 63       | edge                  | 6.058992 |
| 2021-07-30T07:09:18.814Z | x3000c0r42j20p0 | Congestion.rxBW | 0             | 0     | 61       | edge                  | 6.724815 |
| 2021-07-30T07:09:18.814Z | x3000c0r42j20p0 | Congestion.txBW | 0             | 0     | 61       | edge                  | 6.058992 |
| 2021-07-30T07:09:18.814Z | x3000c0r42j20p1 | Congestion.rxBW | 0             | 0     | 60       | edge                  | 6.724815 |
| 2021-07-30T07:09:18.814Z | x3000c0r42j20p1 | Congestion.txBW | 0             | 0     | 60       | edge                  | 6.058992 |
+--------------------------+-----------------+-----------------+---------------+-------+----------+-----------------------+----------+
```

1.5.6.13.17 show telemetry LinkErrors 10

```
Working with 'default' topology and 'default' filter profile.
http://localhost:7777/metrics?$filter=PhysicalContext%20eq%20%27LinkErrors.%27%20and%20Location%20eq%20%27%27\&$top=10\&$expand
+--------------------------+------------+----------------------+---------+----------+--------+-----------------------+-------+
| Timestamp                | Location   | PhysicalContext      | GroupId | SwitchId | PortId | DeviceSpecificContext | Value |
+--------------------------+------------+----------------------+---------+----------+--------+-----------------------+-------+
| 2020-11-15T11:59:34.741Z | x0c0r1j15p0 | LinkErrors.LinkResets | 0       | 1        | 30     | local                 | 1     |
| 2020-11-15T11:59:34.741Z | x0c0r1j17p0 | LinkErrors.LinkResets | 0       | 1        | 34     | local                 | 1     |
| 2020-11-15T11:59:34.741Z | x0c0r1j17p1 | LinkErrors.LinkResets | 0       | 1        | 35     | local                 | 1     |
| 2020-11-15T11:59:34.741Z | x0c0r1j18p1 | LinkErrors.LinkResets | 0       | 1        | 37     | local                 | 1     |
| 2020-11-15T11:59:34.741Z | x0c0r1j20p0 | LinkErrors.LinkResets | 0       | 1        | 41     | local                 | 1     |
| 2020-11-15T11:59:34.741Z | x0c0r1j21p1 | LinkErrors.LinkResets | 0       | 1        | 43     | local                 | 1     |
| 2020-11-15T11:59:34.741Z | x0c0r1j22p1 | LinkErrors.LinkResets | 0       | 1        | 45     | local                 | 1     |
| 2020-11-15T11:59:34.741Z | x0c0r1j23p0 | LinkErrors.LinkResets | 0       | 1        | 46     | local                 | 1     |
| 2020-11-15T11:59:34.741Z | x0c0r1j23p1 | LinkErrors.LinkResets | 0       | 1        | 47     | local                 | 1     |
+--------------------------+------------+----------------------+---------+----------+--------+-----------------------+-------+
```

1.5.6.13.18 show FabricTelemetry

ODquery=$filter=PhysicalContext eq Congestion* and Location eq x3000c0r4*\&$top=10

```
Working with 'default' topology and 'default' filter profile.
Query URL: http://localhost:8000/metrics?$filter=PhysicalContext eq Congestion* and Location eq x3000c0r4*&$top=10&$expand
+--------------------------+-----------------+-----------------+---------------+-------+----------+--------------------+-----------------------+----------
| Timestamp                | Location        | PhysicalContext | ParentalIndex | Index | SubIndex | PhysicalSubContext | DeviceSpecificContext | Value
+--------------------------+-----------------+-----------------+---------------+-------+----------+--------------------+-----------------------+----------
| 2021-07-30T06:33:15.441Z | x3000c0r42j10p0 | Congestion.rxBW | 0             | 0     | 8        | N/A                | edge                  | 6.724703
| 2021-07-30T06:33:15.441Z | x3000c0r42j10p0 | Congestion.txBW | 0             | 0     | 8        | N/A                | edge                  | 4.061454
| 2021-07-30T06:33:15.441Z | x3000c0r42j10p1 | Congestion.rxBW | 0             | 0     | 9        | N/A                | edge                  | 6.724703
| 2021-07-30T06:33:15.441Z | x3000c0r42j14p1 | Congestion.txBW | 0             | 0     | 12       | N/A                | edge                  | 4.094745
| 2021-07-30T06:33:15.441Z | x3000c0r42j4p0  | Congestion.rxBW | 0             | 0     | 3        | N/A                | edge                  | 6.724703
| 2021-07-30T06:33:15.441Z | x3000c0r42j5p1  | Congestion.txBW | 0             | 0     | 5        | N/A                | edge                  | 4.061454
| 2021-07-30T06:33:15.442Z | x3000c0r42j14p0 | Congestion.rxBW | 0             | 0     | 13       | N/A                | edge                  | 6.724703
| 2021-07-30T06:33:15.442Z | x3000c0r42j14p0 | Congestion.txBW | 0             | 0     | 13       | N/A                | edge                  | 4.094745
| 2021-07-30T06:33:15.442Z | x3000c0r42j15p1 | Congestion.txBW | 0             | 0     | 15       | N/A                | edge                  | 4.094745
| 2021-07-30T06:33:15.442Z | x3000c0r42j3p0  | Congestion.txBW | 0             | 0     | 19       | N/A                | edge                  | 4.094745
+--------------------------+-----------------+-----------------+---------------+-------+----------+--------------------+-----------------------+----------
```

1.5.6.14 compute_nodes_creds

By default, STT relies on password-less SSH using the default ssh keypair to access Compute Nodes for diagnostics and troubleshooting purposes. If nodes login authentication uses a different ssh keypair or username/password credentials, administrator can provision STT to use custom imported SSH private key-pair or password based login credentials to access Compute Nodes. When this command is not used, STT tries to access Compute Nodes without any password.

```
(STT) help compute_nodes_creds
Login Credentials for Compute Nodes as username/password or ssh key pair.
Usage: compute_nodes_creds --[ssh-key,password] [SSH key file full path, username:password]
(STT)

Usage:
(STT) compute_nodes_creds --password username:password
(STT) compute_nodes_creds --ssh-key <SSH public key file>
```

For example:

```
(STT) compute_nodes_creds --ssh-key /root/.ssh/id_rsa
```

Note: 1. The credentials will not be stored in any persistent store. 2. The following list of commands will not display correct data, if administrator opt out to setup the compute node credential:

```
show hsn_nodes                                          - Show hsn interfaces of all hsn nodes
show hsn_nodes lldp                                     - Show lldp info of all hsn nodes
show hsn_node hsn <node hostname list>                  - Show hsn nodes hsn interface info
show hsn_node lldp <node hostname list>                 - Show hsn nodes lldp info
show hsn_traffic ping-all-to-all <nodes_list> <hsn_ip_list> <detailed> - Show all to all connectivity between hsn nodes
show hsn_traffic roce_perf_check_loopback <nodes_list>  - Show RoCE Perf benchmark results at compute node port level
snapshot_data compute_nodes <all|list of CNs>           - Takes a backup/snapshot of network config of
                                                            all/given accessible compute nodes
show cables                                             - Show all cables info
```

3. This command will clear STT data model cache, so any subsequent STT command will refetch data using newly provisioned credentials.

1.5.6.15 compute_nodes_hsn_prefix

STT commands which accesses Compute Nodes, requires name of HSN interfaces for fetching interface specific information. But Shasta and HPCM systems can have HSN interfaces represented as hsn0/hsn1 or eno1/eno2 etc. or any other name. This command can be used to provision STT to use any specific HSN interface prefix based on compute nodes. Hence forth, all Compute Nodes specific commands will internally use this information to derive out available interfaces on each Compute Node.

Note: STT uses default HSN prefix as **hsn** if nothing is provisioned.

```
(STT) help compute_nodes_hsn_prefix
Usage: compute_nodes_hsn_prefix <hsn,eno,eth..>
(STT)
```

For example:

```
(STT) compute_nodes_hsn_prefix eno
```

Note: This command will clear STT data model cache, so any subsequent STT command will refetch data.

1.5.6.15.1 show hsn_traffic ping-all-to-all

Performs all-to-all ping test between all the configured HSN interfaces of all HSN nodes as per p2p file or user provided comma separated HSN nodes. This command relies on HSN prefix provisioned by `compute_nodes_hsn_prefix` command or user provided list of comma seperated HSN IP addresses to check connectivity. By default, only non-reachable nodes are listed. To get a list of both reachable and non-reachable nodes, use 'detailed' option at end of command.

To run this commmand, set up STT's access to the compute nodes. Refer to Setting up access to compute nodes for STT

```
(STT) show hsn_traffic ping-all-to-all

Working with 'default' topology and 'default' filter profile.

Displaying the results of ping-all-to-all test between hsn nodes:
Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.

+----------------------------+----------------------+-----------------------------------------------------+
| hsn_node (xname)           | hsn_iface (ipv4_addr) |       non-reachable peer_hsn_nodes (ipv4_addr)       |
+----------------------------+----------------------+-----------------------------------------------------+
| ncn-w003 (x3000c0s20b0n0)  |   hsn0 (10.253.0.27)  | nid000001 (not_configured), uan01 (not_configured),  |
+----------------------------+----------------------+-----------------------------------------------------+
.
.
| nid000001 (x3000c0s28b1n0) |        Unknown        |             ssh to nid000001-nmn failed              |
+----------------------------+----------------------+-----------------------------------------------------+
| nid000002 (x3000c0s28b2n0) |   hsn0 (10.253.0.9)   | nid000001 (not_configured), uan01 (not_configured),  |
+----------------------------+----------------------+-----------------------------------------------------+
hsn_traffic ping-all-to-all test:  Start time: 02/24/2021, 14:23:39 , End time: 02/24/2021, 14:24:02

(STT)

(STT) show hsn_traffic ping-all-to-all detailed

Working with 'default' topology and 'default' filter profile.

Displaying the results of ping-all-to-all test between hsn nodes:
Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.

+----------------------------+---------------------+-----------------------------------------------------------------------------------------------------+
| hsn_node (xname)           |hsn_iface (ipv4_addr)|                              reachable peer_hsn_nodes (ipv4_addr)                                   |
+----------------------------+---------------------+-----------------------------------------------------------------------------------------------------+
| ncn-w003 (x3000c0s20b0n0)  |hsn0 (10.253.0.27)   |ncn-w001 (10.253.0.13),ncn-w002 (10.253.0.12),nid000003 (10.253.0.8),nid000002 (10.253.0.9),|
+----------------------------+---------------------+-----------------------------------------------------------------------------------------------------+
.
.
+----------------------------+---------------------+-----------------------------------------------------------------------------------------------------+
| nid000002 (x3000c0s28b2n0) |hsn0 (10.253.0.9)    |ncn-w003 (10.253.0.27),ncn-w001 (10.253.0.13),ncn-w002 (10.253.0.12),nid000003 (10.253.0.8),|
+----------------------------+---------------------+-----------------------------------------------------------------------------------------------------+


+----------------------------+---------------------+-----------------------------------------------------------------------------------------------------+
| hsn_node (xname)           |hsn_iface (ipv4_addr)|                                  non-reachable peer_hsn_nodes (ipv4_addr)                           |
+----------------------------+---------------------+-----------------------------------------------------------------------------------------------------+
| ncn-w003 (x3000c0s20b0n0)  |   hsn0 (10.253.0.27)| nid000004 (not_configured), nid000001 (not_configured), uan01 (not_configured),                     |
+----------------------------+---------------------+-----------------------------------------------------------------------------------------------------+
.
.
+----------------------------+----------------------+-----------------------------------------------------------------------------------------------------+
| nid000002 (x3000c0s28b2n0) |   hsn0 (10.253.0.9)  | nid000004 (not_configured), nid000001 (not_configured), uan01 (not_configured),                     |
+----------------------------+----------------------+-----------------------------------------------------------------------------------------------------+
hsn_traffic ping-all-to-all test:  Start time: 02/24/2021, 14:24:19 , End time: 02/24/2021, 14:24:37

(STT)

(STT) show hsn_traffic ping-all-to-all x3000c0s20b0n0,x3000c0s28b4n0

Working with 'default' topology and 'default' filter profile.

Displaying the results of ping-all-to-all test between hsn nodes:
Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.


hsn_traffic ping-all-to-all test:  Start time: 02/24/2021, 14:32:44 , End time: 02/24/2021, 14:32:47

(STT)

(STT) show hsn_traffic ping-all-to-all x3000c0s20b0n0,x3000c0s28b4n0 detailed

Working with 'default' topology and 'default' filter profile.
```

```
Displaying the results of ping-all-to-all test between hsn nodes:
Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.

+---------------------------+----------------------+--------------------------------------+
| hsn_node (xname)          | hsn_iface (ipv4_addr) | reachable peer_hsn_nodes (ipv4_addr)  |
+---------------------------+----------------------+--------------------------------------+
| ncn-w003 (x3000c0s20b0n0)  |   hsn0 (10.253.0.27)  |       nid000004 (10.253.0.26),        |
+---------------------------+----------------------+--------------------------------------+
| nid000004 (x3000c0s28b4n0) |   hsn0 (10.253.0.26)  |        ncn-w003 (10.253.0.27),        |
+---------------------------+----------------------+--------------------------------------+

hsn_traffic ping-all-to-all test:  Start time: 02/24/2021, 14:31:10 , End time: 02/24/2021, 14:31:13

(STT)

(STT) show hsn_traffic ping-all-to-all 172.31.241.141,172.31.241.147,172.31.241.149 192.168.0.92,192.168.0.220,192.168.0.156 --detailed

Working with 'default' topology and 'default' filter profile.

Displaying the results of ping-all-to-all test between hsn nodes:


+---------------------------------+-------------------------+-----------------------------------------------------------------------+
| hsn_node (xname)                | hsn_iface (ipv4_addr)  |             non-reachable peer_hsn_nodes (ipv4_addr)              |
+---------------------------------+-------------------------+-----------------------------------------------------------------------+
| 172.31.241.141 (172.31.241.141) | ens801 (192.168.0.92)  | 172.31.241.147 (192.168.0.156), 172.31.241.149 (192.168.0.220),  |
+---------------------------------+-------------------------+-----------------------------------------------------------------------+
| 172.31.241.147 (172.31.241.147) | ens1f0 (192.168.0.156) | 172.31.241.141 (192.168.0.92), 172.31.241.149 (192.168.0.220),   |
+---------------------------------+-------------------------+-----------------------------------------------------------------------+
| 172.31.241.149 (172.31.241.149) | ens1f0 (192.168.0.220) | 172.31.241.141 (192.168.0.92), 172.31.241.147 (192.168.0.156),   |
+---------------------------------+-------------------------+-----------------------------------------------------------------------+
hsn_traffic ping-all-to-all test:  Start time: 09/28/2021, 08:59:51 , End time: 09/28/2021, 09:00:05

(STT)
```

1.5.6.15.2 show hsn_traffic roce_perf_check_loopback

Performs RoCE (RDMA over Converged Ethernet) bandwidth and latency benchmark tests using perftest package on mellanox5 devices at Compute Nodes port level. The bandwidth and latency tests operate on a single port and not cross-fabric routes. The test validates underlying software and hardware components between Compute Node port and Rosetta Switch. We have integrated 'ib_read_bw', 'ib_read_lat', 'ib_write_bw' and 'ib_write_lat' benchmarks.

To run this commmand, set up STT's access to the compute nodes. Refer to Setting up access to compute nodes for STT

```
(STT) help show
show hsn_traffic roce_perf_check_loopback <nodes_list> - Show RoCE Perf benchmark results at compute node port level

(STT) show hsn_traffic roce_perf_check_loopback
.
Displaying the results of RoCE Perf benchmark tests at compute node port level:
.
+---------------------------+------------+--------------+-------------------------+--------------+-------------+
|compute_node (xname)       |ibv_version |Mellanox device|ib_read_bw   |ib_read_lat  |ib_write_bw   |ib_write_lat |
                                          |              |(avg[Gb/sec])|(t_avg[usec])| (avg[Gb/sec])|(t_avg[usec])|
+---------------------------+------------+--------------+-------------+-------------+--------------+-------------+
|nid001001 (x1000c0s0b0n0)  | 16.23.1020 |     mlx5_0    | 100.85      | 1.74        | 100.22       | 0.95        |
+---------------------------+------------+--------------+-------------+-------------+--------------+-------------+
| nid001001 (x1000c0s0b0n0) | 16.23.1020 |     mlx5_1    | 100.86      | 1.74        | 100.19       | 0.95        |
+---------------------------+------------+--------------+-------------+-------------+--------------+-------------+
| nid001002 (x1000c0s0b0n1) | 16.23.1020 |     mlx5_0    | 100.83      | 1.73        | 100.43       | 0.94        |
+---------------------------+------------+--------------+-------------+-------------+--------------+-------------+
| nid001002 (x1000c0s0b0n1) | 16.23.1020 |     mlx5_1    | 100.72      | 1.73        | 100.48       | 0.94        |
+---------------------------+------------+--------------+-------------+-------------+--------------+-------------+
.
.
+---------------------------+------------+--------------+-------------+-------------+--------------+-------------+
| nid001162 (x1000c5s0b0n1) | 16.26.4012 | mlx5_1        | 103.84      | 1.74        | 103.60       | 0.95        |
+---------------------------+------------+--------------+-------------+-------------+--------------+-------------+


List of nodes having Mellanox5 devices with state as PORT_DOWN:
+---------------------------+-----------------------------------------------------------------------------------------------------------+
| Mellanox devices (PORT_DOWN) | compute_nodes                                                                                          |
+---------------------------+-----------------------------------------------------------------------------------------------------------+
| mlx5_0                    | nid001034, nid001038                                                                                      |
+---------------------------+-----------------------------------------------------------------------------------------------------------+
| mlx5_1                    | nid001098, nid001102, nid001107, nid001111, nid001112, nid001115, nid001116                               |
+---------------------------+-----------------------------------------------------------------------------------------------------------+
| mlx5_2                    | nid001001, nid001002, nid001005, nid001006, nid001033, nid001034, nid001037, nid001038, nid001065,  |
|                           | nid001066, nid001070, nid001073, nid001074, nid001097, nid001098, nid001101, nid001102, nid001129,  |
```

```
|                              | nid001130, nid001133, nid001137, nid001138, nid001141, nid001142, nid001145, nid001146, nid001149, |
|                              | nid001150, nid001161, nid001162                                                                    |
+------------------------------+----------------------------------------------------------------------------------------------------+
| mlx5_3                       | nid001001, nid001002, nid001005, nid001006, nid001033, nid001034, nid001037, nid001038, nid001065, |
|                              | nid001066, nid001070, nid001073, nid001074, nid001097, nid001098, nid001101, nid001102, nid001129, |
|                              | nid001130, nid001133, nid001137, nid001138, nid001141, nid001142, nid001145, nid001146, nid001149, |
|                              | nid001150, nid001161, nid001162                                                                    |
+------------------------------+----------------------------------------------------------------------------------------------------+

Configuration used for RoCE Perf benchmarks:
Size of the message to exchange for bw test (-s) : 65536 bytes
Size of the message to exchange for latency test (-s) : 16 bytes
TCP port for initial synchronization (-p) : 18523
Test duration in seconds (-D) : 5 secs

hsn_traffic RoCE perf check loopback: Start time: 03/08/2021, 21:57:21 , End time: 03/08/2021, 21:58:03

(STT)
```

1.5.6.16 copy

Copy switches to another topology, thus creating sub-topologies.

```
(STT) help copy
Copies switches and switch info from the current topology to another topology
Usage: copy <sub-topology name> <xname1>,<xname2> [--include-neighbors]
       Eg : copy topo1 x3000c1r1
            copy topo1 x1000c1r[1-7], x1000*
            copy topo2 x5000c1r1 --include-neighbors
```

This command copies a given switch along with its neighboring switches to another topology mentioned by the user. If the specified topology does not exists, a new topology is created and required switches are copied into it. Neighbour switches are picked based on the fabric and edge links(connections) mentioned in the p2p file for the user specified set of switches.

```
(STT) copy topo1 x3000c0r42
STT diags log directory -  /opt/slingshot/stt_diags_logs/topo1

Copied switches ['x3000c0r42'] to new topology topo1

(STT) set_active topology topo1
(STT) show switches

Working with 'topo1' topology and 'default' filter profile.
Collecting data using 'dgrerrstat' script.
Collecting data using 'dgrperfcheck' script.
dgrerrstat                : Start time: 12/15/2021, 10:14:33, End time: 12/15/2021, 10:14:36, Total time: 0:00:03.919427
dgrperfcheck              : Start time: 12/15/2021, 10:14:33, End time: 12/15/2021, 10:14:37, Total time: 0:00:04.088976
+------------+----------+------+------+------------+--------------+----------+--------------+--------------+-------+----------+-------------+-------------
|   xname    |   type   | snum | gnum | edge_count | fabric_count | up_ports |    pktIn     |    pktOut    | drops |  dscrds  | mcast_found | pcs_uncor_cw
+------------+----------+------+------+------------+--------------+----------+--------------+--------------+-------+----------+-------------+-------------
| x1001c6r3  | Colorado |  0   |  1   |     0      |      2       |  49/64   | 145412267548 | 155994570133 |  1131 | 10907840 |     True    |     True
| x1002c2r3  | Colorado |  0   |  2   |     0      |      2       |  48/64   | 133923825261 | 143605246268 |   0   | 79537178 |     True    |     True
| x3000c0r39 | Columbia |  2   |  0   |     0      |      8       |  32/64   |  974262385   |  1019525070  |   0   | 6126678  |     True    |    False
| x3000c0r40 | Columbia |  0   |  0   |     0      |      8       |  30/64   |  1037109327  |  1072807776  |   0   |   593    |     True    |    False
| x3000c0r41 | Columbia |  1   |  0   |     0      |      8       |  30/64   |  1300686632  |  1327436786  |   0   | 6122773  |     True    |    False
| x3000c0r42 | Columbia |  3   |  0   |     4      |      28      |  32/64   |  2472326666  |  2549320675  |   0   |   595    |     True    |    False
+------------+----------+------+------+------------+--------------+----------+--------------+--------------+-------+----------+-------------+-------------

* The output in this table has been truncated for readability.
```

1.5.6.16.1 copy switches with wildcard indication

Two types of wildcards are available for copying several switches at a time.

- The * wildcard
- The range ([]) wildcard

  For * wildcard, you can indicate a set of switches in two ways. Example: x5000c1* or x5000*. This denotes all switches that begin with the specified string.

  For range wild-card, you can indicate a range of switches in the following way: x5000c1r[1-7]. This will copy all switches in the given range of xnames.

```
(STT) copy topo2 x5000c1*
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/topo2

Copied switches ['x5000c1r1', 'x5000c1r3', 'x5000c1r7', 'x5000c1r5'] to new topology topo2

(STT) copy topo2 x5000c3r[1-7]
```

```
Copied switches ['x5000c3r1', 'x5000c3r3', 'x5000c3r5', 'x5000c3r7'] to new topology topo2

(STT) set_active topology topo2
(STT)
(STT) show switches

<snip>

Collecting data using 'dgrperfcheck' script.
+-----------+----------+------+------+------------+--------------+---------+----------+------+-----------+------------+
|   xname   |   type   | snum | gnum | edge_count | fabric_count |  uptime | up_ports | flap | checkidle |   pktIn    |
+-----------+----------+------+------+------------+--------------+---------+----------+------+-----------+------------+
| x5000c1r3 | Colorado |  0   |  0   |     16     |      24      |   1:03  |  36/64   |  0   |    idle   |    2861    |
| x5000c1r7 | Colorado |  1   |  0   |     16     |      24      | 15 days |  34/64   |  16  |    idle   | 8413139568 |
| x5000c1r1 | Columbia |  2   |  0   |     0      |      24      |   1:29  |  36/64   |  4   |    idle   |   16821    |
| x5000c1r5 | Colorado |  3   |  0   |     16     |      24      | 15 days |  34/64   |  16  |    idle   |  15019329  |
| x5000c3r3 | Colorado |  0   |  1   |     16     |      28      | 15 days |  28/64   |  12  |    idle   |  15472448  |
| x5000c3r5 | Colorado |  1   |  1   |     16     |      28      | 15 days |  28/64   |  12  |    idle   |  15840772  |
| x5000c3r7 | Colorado |  2   |  1   |     16     |      28      |   3:42  |  28/64   |  8   |    idle   |   10004    |
| x5000c3r1 | Columbia |  3   |  1   |     0      |      12      | 15 days |  28/64   |  12  |    idle   |  15227797  |
+-----------+----------+------+------+------------+--------------+---------+----------+------+-----------+------------+
* The output in this table has been truncated for readability.
```

1.5.6.16.2 copy switches using –include-neighbors option

Using this option, all the switches that the specified switch has links to will be copied to the new topology with two levels of switches which include neighbouring switches of the specified switch as well as its neighbour's neighbours to provide a complete picture as shown below.

For example: `copy topo1 x3000c0r24 --include-neighbors` will copy all the neighboring switches of x3000c0r24 and the neighboring switches of x3000c0r24 neighbours.

Note: Edge connections of neighboring switches are not copied. Copy the switch specifically in order to copy its edge links as well.

```
(STT) copy topo3 x3000c0r42 --include-neighbors
STT diags log directory -  /opt/slingshot/stt_diags_logs/topo3

Copied switches ['x3000c0r42', 'x1001c6r3', 'x1002c2r3', 'x3000c0r39', 'x3000c0r40',
'x3000c0r41', 'x1001c4r1', 'x1001c4r3', 'x1001c4r5', 'x1001c4r7', 'x1001c5r1', 'x1001c5r3', 'x1001c5r5', 'x1001c5r7', 'x1001c6r1', 'x1002c2r5', 'x1001c6r7
'x1001c6r5', 'x1001c7r3', 'x1001c7r1', 'x1001c7r5', 'x1001c7r7', 'x1001c2r3', 'x1002c0r1', 'x1002c0r3', 'x1002c0r5', 'x1002c0r7', 'x1002c1r1', 'x1002c1r3'
'x1002c1r7', 'x1002c2r1', 'x1002c2r7', 'x1002c3r3', 'x1002c3r1', 'x1002c3r5', 'x1002c3r7', 'x1000c2r3', 'x1003c6r3', 'x1000c6r3', 'x1003c2r3', 'x1002c6r3'
'x1001c0r1', 'x1002c4r1', 'x1000c0r5', 'x1003c0r1', 'x1000c4r5', 'x1003c4r1', 'x1000c1r7', 'x1003c1r3', 'x1001c1r3', 'x1002c5r3', 'x1000c5r7', 'x1003c5r3'
 'x1003c6r5', 'x1002c6r5', 'x1000c2r1', 'x1003c2r5', 'x1001c2r5', 'x1000c7r3', 'x1003c7r7', 'x1000c3r3', 'x1003c3r7', 'x1001c3r7', 'x1002c7r7', 'x1001c0r3
'x1001c0r7', 'x1001c1r1', 'x1001c1r5', 'x1001c1r7', 'x1001c2r1', 'x1001c2r7', 'x1001c3r3', 'x1001c3r1', 'x1001c3r5', 'x1000c0r7', 'x1003c0r3', 'x1000c4r7'
'x1000c1r5', 'x1003c1r1', 'x1000c5r5', 'x1003c5r1', 'x1003c6r7', 'x1003c2r7', 'x1000c7r1', 'x1003c7r5', 'x1000c3r1', 'x1003c3r5', 'x1000c0r1', 'x1000c0r3'
'x1000c1r3', 'x1000c2r7', 'x1000c2r5', 'x1000c3r5', 'x1000c3r7', 'x1003c4r5', 'x1003c4r7', 'x1003c5r5', 'x1003c5r7', 'x1003c6r1', 'x1003c7r3', 'x1003c7r1'
'x1000c4r3', 'x1000c5r1', 'x1000c5r3', 'x1000c6r7', 'x1000c6r5', 'x1000c7r5', 'x1000c7r7', 'x1003c0r5', 'x1003c0r7', 'x1003c1r5', 'x1003c1r7', 'x1003c2r1'
 'x1003c3r1', 'x1002c4r3', 'x1002c4r5', 'x1002c4r7', 'x1002c5r1', 'x1002c5r5', 'x1002c5r7', 'x1002c6r1', 'x1002c6r7', 'x1002c7r3', 'x1002c7r1', 'x1002c7r5
to new topology topo3

(STT) set_active topology topo3
(STT) show switches

Working with 'topo3' topology and 'default' filter profile.
Collecting data using 'dgrerrstat' script.
Collecting data using 'dgrperfcheck' script.
dgrperfcheck              : Start time: 12/15/2021, 10:27:07, End time: 12/15/2021, 10:27:35, Total time: 0:00:27.819535
dgrerrstat                : Start time: 12/15/2021, 10:27:07, End time: 12/15/2021, 10:28:03, Total time: 0:00:55.342093
+-----------+----------+------+------+------------+--------------+----------+--------------+--------------+-------+-----------+-------------+-----------
|   xname   |   type   | snum | gnum | edge_count | fabric_count | up_ports |    pktIn     |    pktOut    | drops |  dscrds   | mcast_found | pcs_uncor_c
+-----------+----------+------+------+------------+--------------+----------+--------------+--------------+-------+-----------+-------------+-----------
| x1000c0r1 | Colorado |  14  |  1   |     16     |      34      |  50/64   | 278899649615 | 293874148742 |   0   |    539    |     True    |     True
| x1000c0r3 | Colorado |  6   |  1   |     16     |      34      |  50/64   | 255632480653 | 270924695949 |   0   |  46606438 |     True    |     True
| x1000c0r5 | Colorado |  12  |  1   |     16     |      34      |  50/64   | 236396565577 | 249956041496 |  5783 |  23963299 |     True    |     True
| x1000c0r7 | Colorado |  4   |  1   |     18     |      32      |  49/64   | 229877869989 | 243811750353 |  3120 |  27169734 |     True    |     True
| x1000c1r1 | Colorado |  10  |  1   |     16     |      34      |  49/64   | 178770195197 | 190753936098 | 42239 | 158685224 |     True    |     True
| x1000c1r3 | Colorado |  8   |  1   |     16     |      34      |  49/64   | 194419475359 | 207400817297 |  8342 |  45605995 |     True    |     True
| x1000c1r5 | Colorado |  11  |  1   |     18     |      32      |  49/64   | 178537210875 | 190632330877 |   0   |  31723900 |     True    |     True
| x1000c1r7 | Colorado |  5   |  1   |     16     |      34      |  50/64   | 182291788946 | 195338279843 |  5130 |  35691660 |     True    |     True
| x1000c2r1 | Colorado |  3   |  1   |     16     |      34      |  49/64   | 183275604606 | 194035813306 |   0   |  31170391 |     True    |     True
| x1000c2r3 | Colorado |  13  |  1   |     16     |      34      |  49/64   | 181817694912 | 192981519615 |  3342 | 131329308 |     True    |     True
| x1000c2r5 | Colorado |  7   |  1   |     16     |      34      |  50/64   | 224355551195 | 236004258651 |  2634 |  74508922 |     True    |     True
| x1000c2r7 | Colorado |  9   |  1   |     16     |      34      |  50/64   | 209558228470 | 221747310193 |  5794 |  49365506 |     True    |     True
| x1000c3r1 | Colorado |  15  |  1   |     16     |      32      |  48/64   | 223933691851 | 234955790751 |   0   |  59913461 |     True    |     True
| x1000c3r3 | Colorado |  0   |  1   |     16     |      34      |  50/64   | 232745295179 | 244877093334 |  6603 |  24817065 |     True    |     True
| x1000c3r5 | Colorado |  1   |  1   |     16     |      34      |  50/64   | 229954921761 | 241113673722 |  6350 |  35074558 |     True    |     True
| x1000c3r7 | Colorado |  2   |  1   |     16     |      34      |  50/64   | 250187201253 | 262193148126 |   0   |  48907387 |     True    |     True
| x1000c4r1 | Colorado |  4   |  2   |     16     |      34      |  50/64   | 211694006554 | 224513651968 | 12784 |  72872923 |     True    |     True
| x1000c4r3 | Colorado |  3   |  2   |     16     |      34      |  50/64   | 204382507988 | 216946147446 |  4241 |  12002364 |     True    |     True
| x1000c4r5 | Colorado |  5   |  2   |     16     |      34      |  50/64   | 197013802810 | 209373914149 |  2617 |  47666865 |     True    |     True
```

x1000c4r7	Colorado	9	2	18	32	49/64	199495459659	211937697046	0	31725610	True	True
x1000c5r1	Colorado	10	2	16	34	48/64	161368031778	173353417003	0	55392196	True	True
x1000c5r3	Colorado	14	2	16	34	48/64	178859160759	191890083179	0	55431494	True	True
x1000c5r5	Colorado	15	2	18	32	47/64	156005769599	167964143541	0	30644980	True	True
x1000c5r7	Colorado	6	2	16	34	48/64	163567339486	175821213780	2692	41309460	True	True
x1000c6r1	Colorado	1	2	16	34	50/64	164676115286	176761346947	2728	47896933	True	True
x1000c6r3	Colorado	8	2	16	34	50/64	167190160320	179856464608	0	22389199	True	True
x1000c6r5	Colorado	7	2	16	34	50/64	182024927554	194495287780	0	35537751	True	True
x1000c6r7	Colorado	12	2	16	34	50/64	172239634551	184924336594	0	216	True	True
x1000c7r1	Colorado	0	2	16	32	48/64	159309884045	171379040008	1329	59344531	True	True
x1000c7r3	Colorado	11	2	16	34	50/64	167812331483	180193083390	0	12639994	True	True
x1000c7r5	Colorado	13	2	16	34	50/64	156521864419	168540273021	2840	60489105	True	True
x1000c7r7	Colorado	2	2	16	34	49/64	177016025091	190084136600	2867	15475546	True	True
x1001c0r1	Colorado	11	3	16	34	50/64	186614094581	198590294318	0	72952418	True	True
x1001c0r3	Colorado	4	3	16	34	50/64	185737943467	198250658575	0	24529714	True	True
x1001c0r5	Colorado	15	3	16	34	50/64	184333928610	196671698200	0	500	True	True
x1001c0r7	Colorado	5	3	18	32	49/64	185693327144	198413063784	1389	23674186	True	True
x1001c1r1	Colorado	1	3	16	34	50/64	178280701061	189717462955	1235	47498056	True	True
x1001c1r3	Colorado	10	3	16	34	50/64	189379925705	201313395238	2672	11439584	True	True
x1001c1r5	Colorado	6	3	18	32	49/64	177670918921	189020914691	2471	38187796	True	True
x1001c1r7	Colorado	2	3	16	34	50/64	186046315333	197945835998	1313	36151303	True	True
x1001c2r1	Colorado	0	3	16	34	50/64	182490355148	194330115190	3804	35944652	True	True
x1001c2r3	Colorado	3	3	16	34	50/64	185935161185	198416166460	0	25071597	True	True
x1001c2r5	Colorado	14	3	16	34	50/64	184705914544	196704077649	1355	49896401	True	True
x1001c2r7	Colorado	7	3	16	34	50/64	186013387654	198520139139	0	47197519	True	True
x1001c3r1	Colorado	9	3	16	32	47/64	161006562137	171980721144	5731	72400781	True	True
x1001c3r3	Colorado	8	3	16	34	49/64	169034397854	180519207634	4828	78969917	True	True
x1001c3r5	Colorado	13	3	16	34	50/64	180785193652	192632311100	2678	52702433	True	True
x1001c3r7	Colorado	12	3	16	34	50/64	190831015719	203211853687	0	54346192	True	True
x1001c4r1	Colorado	14	4	16	34	49/64	155900779989	167377643687	0	43735459	True	True
x1001c4r3	Colorado	3	4	16	34	49/64	150760346033	162802703600	0	29756347	True	True
x1001c4r5	Colorado	1	4	16	34	49/64	142212586643	153501655504	1337	22215192	True	True
x1001c4r7	Colorado	13	4	18	32	48/64	142679059375	154012449067	1143	102817978	True	True
x1001c5r1	Colorado	11	4	16	34	50/64	164696473712	176427846041	1561	95104603	True	True
x1001c5r3	Colorado	8	4	16	34	50/64	177587046055	189374787038	5525	48073864	True	True
x1001c5r5	Colorado	7	4	18	32	49/64	155820476649	167072747888	0	66235862	True	True
x1001c5r7	Colorado	4	4	16	34	50/64	164535030177	176610986367	4180	59438977	True	True
x1001c6r1	Colorado	5	4	16	34	49/64	158316438945	169909320189	1262	33532597	True	True
x1001c6r3	Colorado	6	4	16	34	49/64	159222442540	170703378399	1131	10965422	True	True
x1001c6r5	Colorado	12	4	16	34	48/64	154897989045	166532772126	4363	78612097	True	True
x1001c6r7	Colorado	10	4	16	34	50/64	151061657022	163291425931	0	85914716	True	True
x1001c7r1	Colorado	15	4	16	32	47/64	158668354239	169965531568	4346	58615090	True	True
x1001c7r3	Colorado	2	4	16	34	49/64	166607222956	178664853156	6864	88934912	True	True
x1001c7r5	Colorado	0	4	16	34	49/64	155949973113	167386548884	1420	83489986	True	True
x1001c7r7	Colorado	9	4	16	34	49/64	169196868787	180822532926	6840	52445880	True	True
x1002c0r1	Colorado	12	5	16	34	47/64	131395657310	141652149343	0	30244351	True	True
x1002c0r3	Colorado	0	5	16	34	47/64	129350928703	140060780681	0	18077478	True	True
x1002c0r5	Colorado	9	5	16	34	48/64	145995503808	156492034268	1380	83334464	True	True
x1002c0r7	Colorado	3	5	18	32	47/64	150328519689	160782857864	0	44208324	True	True
x1002c1r1	Colorado	13	5	16	34	50/64	160382537546	171154228985	1253	106341765	True	True
x1002c1r3	Colorado	6	5	16	34	50/64	170652630321	181576012135	2485	53422198	True	True
x1002c1r5	Colorado	15	5	18	32	48/64	155921869710	166464481137	2713	73445421	True	True
x1002c1r7	Colorado	1	5	16	34	49/64	159248936378	170261327422	0	29695168	True	True
x1002c2r1	Colorado	7	5	16	34	48/64	139662530043	150035788916	0	78080332	True	True
x1002c2r3	Colorado	14	5	16	34	48/64	146854198823	157458190482	0	79594553	True	True
x1002c2r5	Colorado	4	5	16	34	48/64	138384300552	148781240251	0	18323588	True	True
x1002c2r7	Colorado	11	5	16	34	48/64	137953098843	148993821188	1326	80939575	True	True
x1002c3r1	Colorado	8	5	16	32	47/64	134064173202	144032385297	0	42330555	True	True
x1002c3r3	Colorado	5	5	16	34	49/64	137467536444	148036351209	0	43680071	True	True
x1002c3r5	Colorado	10	5	16	34	50/64	146013647840	156335715385	0	80955420	True	True
x1002c3r7	Colorado	2	5	16	34	50/64	155242089001	165648848303	0	73735457	True	True
x1002c4r1	Colorado	15	6	16	34	49/64	149653279937	158974041451	0	64594064	True	True
x1002c4r3	Colorado	9	6	16	34	49/64	154082548582	163643015698	0	30198840	True	True
x1002c4r5	Colorado	4	6	16	34	50/64	158762869292	168682274745	0	85237367	True	True
x1002c4r7	Colorado	5	6	18	32	49/64	161600334461	171808774559	0	6175094	True	True
x1002c5r1	Colorado	0	6	16	34	50/64	154553859706	164139177318	0	61056382	True	True
x1002c5r3	Colorado	13	6	16	34	50/64	161530220210	171534574951	0	49444299	True	True
x1002c5r5	Colorado	3	6	18	32	48/64	134691070763	143607502738	0	42190295	True	True
x1002c5r7	Colorado	11	6	16	34	49/64	142227487471	151323708958	0	54528953	True	True
x1002c6r1	Colorado	2	6	16	34	50/64	155205012703	165226474355	0	323	True	True
x1002c6r3	Colorado	7	6	16	34	50/64	158708110237	168916251324	0	668	True	True
x1002c6r5	Colorado	14	6	16	34	50/64	146315973089	155664756571	0	11930802	True	True
x1002c6r7	Colorado	6	6	16	34	50/64	150238384937	159834097430	0	728	True	True
x1002c7r1	Colorado	8	6	16	32	48/64	143949274342	153773410189	0	34159589	True	True
x1002c7r3	Colorado	1	6	16	34	50/64	151286170011	161254842188	0	11751142	True	True
x1002c7r5	Colorado	12	6	16	34	48/64	128791416447	137708666732	0	17920863	True	True
x1002c7r7	Colorado	10	6	16	34	48/64	133576228932	142759751537	0	6178785	True	True
x1003c0r1	Colorado	8	7	16	34	48/64	106748076453	115982895896	0	41002960	True	True
x1003c0r3	Colorado	15	7	16	34	48/64	114473489823	124304526527	0	79111755	True	True
x1003c0r5	Colorado	5	7	16	34	46/64	92374601741	101382126408	0	52300895	True	True
x1003c0r7	Colorado	10	7	18	32	45/64	98697649195	108210572070	0	76603670	True	True
x1003c1r1	Colorado	6	7	16	34	49/64	101371244170	110278294795	0	73546265	True	True
x1003c1r3	Colorado	13	7	16	34	49/64	101339550702	110296746146	0	53021171	True	True
x1003c1r5	Colorado	12	7	18	32	49/64	125386496195	134691576035	0	42258297	True	True
x1003c1r7	Colorado	0	7	16	34	50/64	129412305627	139290476084	0	11762133	True	True
x1003c2r1	Colorado	7	7	16	34	50/64	79800713529	88486947284	0	60537916	True	True

```
| x1003c2r3  | Colorado | 2  | 7  |    16    |      34      | 50/64  | 83428480240  | 92232195122  |  0  |  59284172 |     True     |     True
| x1003c2r5  | Colorado | 14 | 7  |    16    |      34      | 50/64  | 106447111194 | 115582934117 |  0  |  11768606 |     True     |     True
| x1003c2r7  | Colorado | 9  | 7  |    16    |      34      | 50/64  | 114796805100 | 124665848511 |  0  |  59158132 |     True     |     True
| x1003c3r1  | Colorado | 11 | 7  |    16    |      32      | 47/64  | 116841846818 | 126088605022 |  0  |  43084760 |     True     |     True
| x1003c3r3  | Colorado | 1  | 7  |    16    |      34      | 49/64  | 120585721994 | 130445586774 |  0  |  77270526 |     True     |     True
| x1003c3r5  | Colorado | 3  | 7  |    16    |      34      | 49/64  | 117116099565 | 126260096451 |  0  |  29974905 |     True     |     True
| x1003c3r7  | Colorado | 4  | 7  |    16    |      34      | 49/64  | 117397711036 | 126620053938 |  0  | 140291469 |     True     |     True
| x1003c4r1  | Colorado | 8  | 8  |    16    |      34      | 46/64  | 95288315695  | 101468679608 |  0  |  54067173 |     True     |     True
| x1003c4r3  | Colorado | 14 | 8  |    16    |      34      | 46/64  | 99212270268  | 105764653510 |  0  |  92404394 |     True     |     True
| x1003c4r5  | Colorado | 9  | 8  |    16    |      34      | 46/64  | 96778949098  | 103109190487 |  0  |  52649976 |     True     |     True
| x1003c4r7  | Colorado | 1  | 8  |    18    |      32      | 45/64  | 98425342730  | 105089636044 |  0  |  41810908 |     True     |     True
| x1003c5r1  | Colorado | 11 | 8  |    16    |      34      | 48/64  | 109947966777 | 116269512634 |  0  |  18127092 |     True     |     True
| x1003c5r3  | Colorado | 5  | 8  |    16    |      34      | 48/64  | 113396956527 | 119819590101 |  0  |  40883041 |     True     |     True
| x1003c5r5  | Colorado | 12 | 8  |    18    |      32      | 47/64  | 100806761795 | 106815584667 |  0  |  51828839 |     True     |     True
| x1003c5r7  | Colorado | 3  | 8  |    16    |      34      | 48/64  | 108858895663 | 115508686338 |  0  |  6170276  |     True     |    False
| x1003c6r1  | Colorado | 7  | 8  |    16    |      34      | 49/64  | 120260816073 | 127097007715 |  0  |  55487357 |     True     |     True
| x1003c6r3  | Colorado | 15 | 8  |    16    |      34      | 49/64  | 121159479217 | 128175407653 |  0  | 115601113 |     True     |     True
| x1003c6r5  | Colorado | 6  | 8  |    16    |      34      | 49/64  | 128129017366 | 135082183679 |  0  |  61358944 |     True     |     True
| x1003c6r7  | Colorado | 2  | 8  |    16    |      34      | 49/64  | 133592993147 | 140896426767 |  0  |  58495432 |     True     |     True
| x1003c7r1  | Colorado | 10 | 8  |    16    |      32      | 47/64  | 126641235356 | 133086807097 |  0  |  30207773 |     True     |     True
| x1003c7r3  | Colorado | 4  | 8  |    16    |      34      | 49/64  | 134792355686 | 141673955639 |  0  |  30026927 |     True     |     True
| x1003c7r5  | Colorado | 0  | 8  |    16    |      34      | 50/64  | 126770409440 | 133745741091 |  0  |  60221917 |     True     |     True
| x1003c7r7  | Colorado | 13 | 8  |    16    |      34      | 50/64  | 131445353424 | 138607108142 |  0  |  47691399 |     True     |     True
| x3000c0r39 | Columbia | 2  | 0  |    6     |      28      | 32/64  |  975534496   |  1020975919  |  0  |  6184066  |     True     |    False
| x3000c0r40 | Columbia | 1  | 0  |    2     |      28      | 30/64  |  1038550642  |  1074386557  |  0  |    593    |     True     |    False
| x3000c0r41 | Columbia | 3  | 0  |    4     |      28      | 30/64  |  1301029026  |  1327863684  |  0  |  6180166  |     True     |    False
| x3000c0r42 | Columbia | 0  | 0  |    4     |      28      | 32/64  |  2473523522  |  2550751795  |  0  |    595    |     True     |    False
+------------+----------+----+----+----------+--------------+--------+--------------+--------------+-----+-----------+--------------+-----------
```

```
* The output in this table has been truncated for readability.
```

1.5.6.17 combine

```
(STT) help combine
       Combine topologies into a new topology.
       Usage: combine <result> <list of topologies to combine>
```

1.5.6.18 exit

```
(STT) help exit
       Exits the program.
```

1.5.6.19 set_active

```
(STT) help set_active
Set active working topology, filter profile or output format.
If a topology or filter object with the given name does not exist,
then a new topology or filter profile is created.

Usage: set_active   topology        <topology name>
       set_active   filter          <filter profile name>
       set_active   output-format   <TABLE(default)|JSON|CSV>

Tip: Use "list" command to know available topology and filters
```

1.5.6.19.1 set_active topology

Set active topology.

```
(STT) set_active topology default
```

1.5.6.19.2 set_active output-format

Set current display option for the **show** command output. Default output format is TABLE and output is shown as prettytable. The display option can be set to CSV or JSON and STT and produces the results in the specified format.

```
(STT) show output-format
Current output format is: TABLE
(STT) set_active   output-format CSV
(STT) show output-format
Current output format is: CSV


(STT)
(STT) show switches

Working with 'default' topology and 'default' filter profile.
Collecting data using 'check-switches' script.
Collecting data using 'dgrerrstat' script.
Collecting data using 'dgrperfcheck' script.
```

```
xname,type,snum,gnum,edge_count,fabric_count,uptime,up_ports,flap,checkidle,pktIn,pktOut,drops,dscrds,mcast_found
x3000c0r42,Columbia,0,0,12,4,28days,16/64,15,idle,240529432,241484858,0,9578207,True
x9000c1r3,Colorado,0,1,16,26,22days,34/64,25,idle,52348613,53104549,0,8509004,True
x9000c3r7,Colorado,1,1,16,24,7days,26/64,0,idle,75709,239912,0,70867,True
x9000c3r3,Colorado,2,1,16,24,22days,26/64,16,idle,74427,238453,0,70867,True
x9000c1r7,Colorado,3,1,16,26,22days,34/64,15,idle,582681076,583425028,0,70867,True

(STT) set_active   output-format JSON
(STT) show switches

Working with 'default' topology and 'default' filter profile.

{
    "1": {
        "xname": "x3000c0r42",
        "type": "Columbia",
        "snum": "0",
        "gnum": "0",
        "edge_count": "12",
        "fabric_count": "4",
        "uptime": "28days",
        "up_ports": "16/64",
        "flap": "15",
        "checkidle": "idle",
        "pktIn": "240529432",
        "pktOut": "241484858",
        "drops": "0",
        "dscrds": "9578207",
        "mcast_found": "True"
    },
    "2": {
        "xname": "x9000c1r3",
        "type": "Colorado",
        "snum": "0",
        "gnum": "1",
        "edge_count": "16",
        "fabric_count": "26",
        "uptime": "22days",


    ..................
    ..................

    "5": {
        "xname": "x9000c1r7",
        "type": "Colorado",
        "snum": "3",
        "gnum": "1",
        "edge_count": "16",
        "fabric_count": "26",
        "uptime": "22days",
        "up_ports": "34/64",
        "flap": "15",
        "checkidle": "idle",
        "pktIn": "582681076",
        "pktOut": "583425028",
        "drops": "0",
        "dscrds": "70867",
        "mcast_found": "True"
    }
}
```

1.5.6.20 `snapshot_data`

This functionality helps to take a snapshot or a backup of the current STT data model, of the fabric agent, and of compute nodes. The following points explain the individual features of `snapshot_data`.

1. `snapshot_data` command takes a snapshot of the existing data model after a refresh. All the show command outputs will be displayed on the screen. This snapshot can be redirected to a file if necessary.
2. `snapshot_data compute_nodes all|list of CNs` option can be used to take a snapshot of the network configuration of compute nodes.
3. `snapshot_data fmn_fabric` option takes a data backup of the fabric agent.

NOTE: `snapshot_data` compute nodes option requires an additional RPM named `slingshot-utils` to be installed on compute nodes. Please refer to the pre-requisites section of this README.

```
(STT) help snapshot_data
Take a snapshot of the current STT data model after a refresh.
Usage: snapshot_data - Takes a snapshot after a refresh of the STT data model.
       snapshot_data compute_nodes <all|list of CNs>- Takes a backup/snapshot of network config of
```

```
                                                    all/given accessible Compute nodes.
      snapshot_data fmn_fabric - Takes a backup/snapshot of the triage data of fabric agent.
      snapshot_data switch_csr <all|list of xnames>- Takes a backup/snapshot of CSR data of all/given accessible switches.
```

1.5.6.20.1 snapshot_data Snapshot of current STT data model

```
(STT) snapshot_data
Taking a snapshot of the STT data model...
Refreshing STT data model

NOTE: This action may take a while...

Working with 't1' topology.
JSON Validation succeeded
JSON Validation succeeded
Collecting data using 'dgrvalidatesyscfg' script.
Collecting data using 'dgrperfcheck' script.
Collecting data using 'services_platform' script.
Collecting data using 'check-fabric' script.
Collecting data using 'dgrheadshellstat' script.
Collecting data using 'services_rosetta' script.
Collecting data using 'dgrerrstat' script.
Collecting data using 'dgrlinkstat' script.
Collecting data using 'check-switches' script.
Collecting data using 'check_hsn_nics_config' script.
Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.

Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.

services_platform  :  Start time: 01/28/2021, 03:51:17 , End time: 01/28/2021, 03:51:26
services_rosetta  :  Start time: 01/28/2021, 03:51:17 , End time: 01/28/2021, 03:51:30
check-fabric  :  Start time: 01/28/2021, 03:51:17 , End time: 01/28/2021, 03:51:35
An exception occurred while running dgrvalidatesyscfg : Authentication failed.
dgrvalidatesyscfg  :  Start time: 01/28/2021, 03:51:17 , End time: 01/28/2021, 03:51:36
dgrlinkstat  :  Start time: 01/28/2021, 03:51:17 , End time: 01/28/2021, 03:51:45
dgrerrstat  :  Start time: 01/28/2021, 03:51:17 , End time: 01/28/2021, 03:51:51
check-switches  :  Start time: 01/28/2021, 03:51:17 , End time: 01/28/2021, 03:52:10
dgrheadshellstat  :  Start time: 01/28/2021, 03:51:17 , End time: 01/28/2021, 03:53:28

check_hsn_nics_config  :  Start time: 01/28/2021, 03:51:17 , End time: 01/28/2021, 03:53:45
dgrperfcheck  :  Start time: 01/28/2021, 03:51:17 , End time: 01/28/2021, 03:53:46
show active_topology :
Current active topology name is: t1
show active_filter :
Current active filter profile is: default
show output-format :
Current output format is: TABLE
show switches :

Working with 't1' topology and 'default' filter profile.
+-----------+----------+------+------+------------+--------------+----------------------+--------+-------+------+--------+-------+------+
|   xname   |   type   | snum | gnum | edge_count | fabric_count |    firmware_version  |pingable|dynamic|uptime|up_ports|slwInit| flap |
+-----------+----------+------+------+------------+--------------+----------------------+--------+-------+------+--------+-------+------+
| x1000c0r1 | Colorado |  0   |  0   |     16     |      36      | sc.1.5.40-prod-master |  False |  False| 4:58 |  62/64 |    0  |  16  |
| x1000c3r3 | Colorado |  1   |  0   |     16     |      34      | sc.1.5.40-prod-master |  False |  False| 4:58 |  62/64 |    0  |  16  |
| x1000c3r5 | Colorado |  2   |  0   |     16     |      36      | sc.1.5.40-prod-master |  False |  False| 4:58 |  62/64 |    0  |  16  |
| x1000c3r7 | Colorado |  3   |  0   |     16     |      36      | sc.1.5.40-prod-master |  False |  False| 4:57 |  62/64 |    0  |  16  |
| x1000c2r7 | Colorado |  4   |  0   |     16     |      36      | sc.1.5.40-prod-master |  False |  False| 4:57 |  62/64 |    0  |  16  |

.
.
.
<ALL SHOW COMMANDS WITH NO_FILTER>
.
.

* The output in this table has been truncated for readability.
```

1.5.6.20.2 snapshot_data Redirect snapshot to a file

```
(STT) snapshot_data > stt_snapshot_28Jan.txt

(STT) shell cat stt_snapshot_28Jan.txt | head -20
Taking a snapshot of the STT data model...
Refreshing STT data model

NOTE: This action may take a while...

Working with 't1' topology.
Collecting data using 'dgrvalidatesyscfg' script.
```

```
Collecting data using 'dgrperfcheck' script.
Collecting data using 'services_platform' script.
.
.
.
services_platform  :  Start time: 01/28/2021, 04:02:28 , End time: 01/28/2021, 04:02:37
services_rosetta  :  Start time: 01/28/2021, 04:02:28 , End time: 01/28/2021, 04:02:39
check-fabric  :  Start time: 01/28/2021, 04:02:28 , End time: 01/28/2021, 04:02:47
dgrlinkstat  :  Start time: 01/28/2021, 04:02:28 , End time: 01/28/2021, 04:02:58
dgrerrstat  :  Start time: 01/28/2021, 04:02:28 , End time: 01/28/2021, 04:03:06
check-switches  :  Start time: 01/28/2021, 04:02:28 , End time: 01/28/2021, 04:03:26
dgrvalidatesyscfg  :  Start time: 01/28/2021, 04:02:28 , End time: 01/28/2021, 04:03:26
dgrheadshellstat  :  Start time: 01/28/2021, 04:02:28 , End time: 01/28/2021, 04:03:34
dgrperfcheck  :  Start time: 01/28/2021, 04:02:28 , End time: 01/28/2021, 04:03:35
check_hsn_nics_config  :  Start time: 01/28/2021, 04:02:29 , End time: 01/28/2021, 04:05:01
show active_topology :
Current active topology name is: t1
show active_filter :
Current active filter profile is: default
show output-format :
Current output format is: TABLE
.
.
.
show switches health :

Working with 't1' topology and 'default' filter profile.

----------------RESULTS------------------

Health of switch:  x1000c0r1
+-----------------+-------------------+----------------+--------------+-------------------+
| Component       | Parameter Checked | Expected Value | Actual Value | Error Code        |
+-----------------+-------------------+----------------+--------------+-------------------+
| Group           | ID/Numbers        |       0        |      1       | ERR_GROUP_NUM     |
| SerDes firmware | Version           |  0X109B_208D   | 0X109E_208D  | ERR_SERDES_FW_VER |
+-----------------+-------------------+----------------+--------------+-------------------+
.
.
.
```

1.5.6.20.3 `snapshot_data compute_nodes` Take a snapshot of compute node network configuration

To run this commmand, set up STT's access to the compute nodes. Refer to Setting up access to compute nodes for STT

```
1. (STT) snapshot_data compute_nodes all
Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.

Snapshot tool unsuccessful on node nid001026
Snapshot tool unsuccessful on node nid001034
.
.
.
Snapshot tool unsuccessful on node nid001005
Snapshot succesfully collected from nid001003
Snapshot succesfully collected from nid001002
Snapshot succesfully collected from nid001001
Network config snapshot of compute nodes is collected in /root/slingshot_tools/slingshot-cn-snapshot-030321-053307.tgz

2. (STT) snapshot_data compute_nodes nid001001,nid001002,nid001003
Snapshot succesfully collected from nid001003
Snapshot succesfully collected from nid001002
Snapshot succesfully collected from nid001001
Network config snapshot of compute nodes is collected in /root/slingshot_tools/slingshot-cn-snapshot-030321-053412.tgz
```

1.5.6.20.4 `snapshot_data fmn_fabric` Take a snapshot of fabric agent information

```
(STT) snapshot_data fmn_fabric
Wed Mar 17 12:02:02 UTC 2021

/var/tmp/triage-data/
/var/tmp/triage-data/fabric-agent-x1001c7r1b0.txt
/var/tmp/triage-data/fabric-agent-x3001c0r42b0.txt
/var/tmp/triage-data/fabric-agent-x1001c6r7b0.txt
/var/tmp/triage-data/fabric-agent-x1001c5r5b0.txt
/var/tmp/triage-data/fabric-agent-x1000c7r1b0.txt
/var/tmp/triage-data/fabric-agent-x1001c0r3b0.txt
/var/tmp/triage-data/fabric-agent-x1001c5r3b0.txt
/var/tmp/triage-data/fabric-agent-x1000c1r1b0.txt
/var/tmp/triage-data/fabric-agent-x1000c4r3b0.txt
/var/tmp/triage-data/fabric-agent-x3000c0r42b0.txt
/var/tmp/triage-data/fabric-agent-x1000c1r5b0.txt
```

```
/var/tmp/triage-data/fabric-agent-x1000c2r7b0.txt
/var/tmp/triage-data/fabric-agent-x1000c5r1b0.txt
/var/tmp/triage-data/fabric-agent-x1001c3r1b0.txt
/var/tmp/triage-data/fabric-agent-x1000c6r7b0.txt
/var/tmp/triage-data/fabric-agent-x3002c0r42b0.txt
/var/tmp/triage-data/fabric-agent-x1001c2r7b0.txt
/var/tmp/triage-data/fabric-agent-x1001c4r5b0.txt
/var/tmp/triage-data/fabric-agent-x1001c7r3b0.txt
/var/tmp/triage-data/fabric_template.json
.
.
.
```

1.5.6.20.5 `snapshot_data switch_csr` Take a snapshot of CSR data of switches

```
(STT) snapshot_data switch_csr all

/tmp/x1000c3r1-CSR-230321020035.json
/tmp/x1000c7r5-CSR-230321020036.json
/tmp/x1000c3r3-CSR-230321020036.json
/tmp/x1000c6r1-CSR-230321020036.json
/tmp/x3002c0r42-CSR-230321020036.json
/tmp/x1000c7r3-CSR-230321020036.json
/tmp/x1000c2r1-CSR-230321020036.json
/tmp/x1000c2r5-CSR-230321020036.json
/tmp/x1000c1r3-CSR-230321020037.json
/tmp/x3000c0r41-CSR-230321020037.json
.
.
.
/tmp/x1001c2r1-CSR-230321020039.json
/tmp/x1001c1r5-CSR-230321020039.json
/tmp/x1001c2r3-CSR-230321020040.json
/tmp/x1001c4r5-CSR-230321020040.json
/tmp/x1001c5r5-CSR-230321020040.json
/tmp/x1001c5r7-CSR-230321020040.json
Snapshot of switch CSR data is collected in /root/slingshot_tools/slingshot-switch-csr-230321020040.tgz
```

```
(STT) snapshot_data switch_csr x1001c7r1,x1000c5r5,x1001c0r7,x3000c0r42
/tmp/x1000c5r5-CSR-230321021237.json
/tmp/x3000c0r42-CSR-230321021237.json
/tmp/x1001c7r1-CSR-230321021237.json
/tmp/x1001c0r7-CSR-230321021237.json
Snapshot of switch CSR data is collected in /root/slingshot_tools/slingshot-switch-csr-230321021237.tgz
```

1.5.6.21 history

```
(STT) help history
Usage: history [-h] [-r | -e | -o FILE | -t TRANSCRIPT_FILE | -c] [-s] [-x]
               [-v] [-a]
               [arg]
View, run, edit, save, or clear previously entered commands
positional arguments:
  arg                   empty               all history items
                        a                   one history item by number
                        a..b, a:b, a:, ..b  items by indices (inclusive)
                        string              items containing string
                        /regex/             items matching regular expression
optional arguments:
  -h, --help            show this help message and exit
  -r, --run             run selected history items
  -e, --edit            edit and then run selected history items
  -o, --output_file FILE
                        output commands to a script file, implies -s
  -t, --transcript TRANSCRIPT_FILE
                        output commands and results to a transcript file,
                        implies -s
  -c, --clear           clear all history
formatting:
  -s, --script          output commands in script format, i.e. without command
                        numbers
  -x, --expanded        output fully parsed commands with any aliases and
                        macros expanded, instead of typed commands
  -v, --verbose         display history and include expanded commands if they
                        differ from the typed command
  -a, --all             display all commands, including ones persisted from
                        previous sessions
```

The history command lists the commands executed in the near past.

```
(STT) show switches
Working with 'default' topology and 'default' filter profile.
Collecting data using 'check-switches' script.
. . .
(STT) add switch x0c0r1 0 0
```

```
Colorado Working with 'default' topology.
(STT) show switches
Working with 'default' topology and 'default' filter profile.
JSON Validation succeeded
. . .
(STT) history
 1 load topology /root/slingshot_tools/stt/test/inputs/topo1.json
 2 help remove
 3 help save
 4 show active topology
 5 show active_topology
 6 save topology /tmp/sample_topo
 7 help show
 8 show switches
 9 show headshells
 10 refresh_data
 11 show headshells
 12 show cables
 13 show edge
 14 show fabric
 15 show switches
 16 show switch port x5000c3r3
 17 show switch ports x5000c3r3
 18 show switch jacks x5000c3r3
 19 show switch jacks x5000c3r3
 20 show switch jacks x3000c1r40
 21 show cables |grep x5000c3r3
 22 show switch error_flags x5000c3r3
 23 show switch counters x5000c3r3
 24 show switch counters x3000c1r40
 25 help combine
 26 help exit
 27 help set_active
 28 help history
 29 show switches
 30 add switch x0c0r10 0 Colorado
 31 add switch x0c0r1 0 0 Colorado
 32 show switches
```

1.5.6.22 `shell`

Use the shell to execute a command.

```
(STT) help shell
Usage: shell [-h] command ...
Execute a command as if at the OS prompt
positional arguments:
command the command to run command_args arguments to pass to command
optional arguments: -h, --help show this help message and exit
```

For example, execute `cat` within STT.

```
(STT) show cables>> cable_log
(STT) shell cat cable_log

Working with 'default' topology and 'default' filter profile.
Collecting data using 'check-switches' script.

+-------------+-------------+-------------+-------------+--------+------------------+-----------------------------------------------------------------------------------------
|     srca    |     srcb    |     dsta    |     dstb    |  type  |      status      |                                    serial_ids
+-------------+-------------+-------------+-------------+--------+------------------+-----------------------------------------------------------------------------------------
| x10c0r1j10p1| x10c0r1j10p1| x10c5r7j14p1| x10c5r5j14p1| fabric |     Connected    |               57DS9LD2118, 57DS9LD2118, 57DS9LD2118, 57DS9LD2118
| x10c0r1j11p0| x10c0r1j11p0| x10c7r1j14p0| x10c7r3j14p0| fabric |     Connected    |               021321G075, 021321G075, 021321G075, 021321G075
| x10c0r1j11p1| x10c0r1j11p1| x10c7r3j14p1| x10c7r1j14p1| fabric |     Connected    |               021321G075, 021321G075, 021321G075, 021321G075
| x10c0r1j12p0| x10c0r1j12p0| x10c7r5j14p0| x10c7r7j14p0| fabric |     Connected    |               57DS9ND2047, 57DS9ND2047, 57DS9ND2047, 57DS9ND2047
..................
| x0c0r0j16p0 | x0c0r0j16p1 |  x0c0s1j16  |      nc     |  edge  |     Unknown      |               UH294G00846, UH294G00846, Connection Failure, NC
| x0c0r0j17p1 | x0c0r0j17p0 | x0c0r1j24p1 | x0c0r1j24p0 | fabric | Wrong Connection |               UH294G00668,UH294G00668,UH294G00948,UH294G00948
| x0c0r0j18p1 | x0c0r0j18p0 | x0c0r1j23p1 | x0c0r1j23p0 | fabric | Wrong Connection |               UH294G00229,UH294G00229,UH294G00641,UH294G00641
..................
| x0c0r0j16p1 | x0c0r0j16p0 | x0c0r1j16p1 | x0c0r1j16p0 | fabric |     Unknown      | Connection Failure, Connection Failure, Connection Failure, Connecti
| x0c0r0j17p1 | x0c0r0j17p0 | x0c0r1j24p1 | x0c0r1j24p0 | fabric |     Unknown      | Connection Failure, Connection Failure, Connection Failure, Connecti
| x0c0r0j106p1| x0c0r0j106p0| x0c5s7b0n0h0| x0c5s7b0n0h1|  edge  |      Absent      |                                NA,NA,NA,NA
| x0c0r0j107p1| x0c0r0j107p0| x0c5s7b0n1h0| x0c5s7b0n1h1|  edge  |      Absent      |                                NA,NA,NA,NA
+-------------+-------------+-------------+-------------+--------+------------------+-----------------------------------------------------------------------------------------

* The output has been truncated for readability.
```

For example, display a subset of output using `grep` and `cat`.

```
(STT) show cables |grep -i x0c0r0j1 >> cable_log1
(STT) shell cat cable_log1
| x0c0r0j16p1 | x0c0r0j16p0 | x0c0r1j16p1 | x0c0r1j16p0 | fabric |     Unknown      | Connection Failure, Connection Failure, Connection Failure, Connecti
| x0c0r0j17p1 | x0c0r0j17p0 | x0c0r1j24p1 | x0c0r1j24p0 | fabric |     Unknown      | Connection Failure, Connection Failure, Connection Failure, Connecti
| x0c0r0j18p1 | x0c0r0j18p0 | x0c0r1j23p1 | x0c0r1j23p0 | fabric |     Unknown      | Connection Failure, Connection Failure, Connection Failure, Connecti
```

```
| x0c0r0j19p1 | x0c0r0j19p0 | x0c0r1j22p1 | x0c0r1j22p0 | fabric |     Unknown      | Connection Failure, Connection Failure, Connection Failure, Connecti
| x0c0r0j16p1 | x0c0r0j16p0 | x0c0r1j16p1 | x0c0r1j16p0 | fabric |    Connected     |                UH294G00846,UH294G00846,UH294G00846,UH294G00846
| x0c0r0j17p1 | x0c0r0j17p0 | x0c0r1j24p1 | x0c0r1j24p0 | fabric | Wrong Connection |                UH294G00668,UH294G00668,UH294G00948,UH294G00948
| x0c0r0j18p1 | x0c0r0j18p0 | x0c0r1j23p1 | x0c0r1j23p0 | fabric | Wrong Connection |                UH294G00229,UH294G00229,UH294G00641,UH294G00641
| x0c0r0j19p1 | x0c0r0j19p0 | x0c0r1j22p1 | x0c0r1j22p0 | fabric | Wrong Connection |                UH294G01026,UH294G01026,UH294G00932,UH294G00932
```

1.5.7 Data Model Load/Update/Clear

STT helps maintain the inventory and health status of various components in an in-memory database, also referred to as the "data model". The topology configuration is encoded within the data model.

The current version of the tool supports creating a data model for following components:

- cables
- edge
- fabric
- headshells
- switches
- ports within a switch
- jacks within a switch

1.5.7.1 Workflow

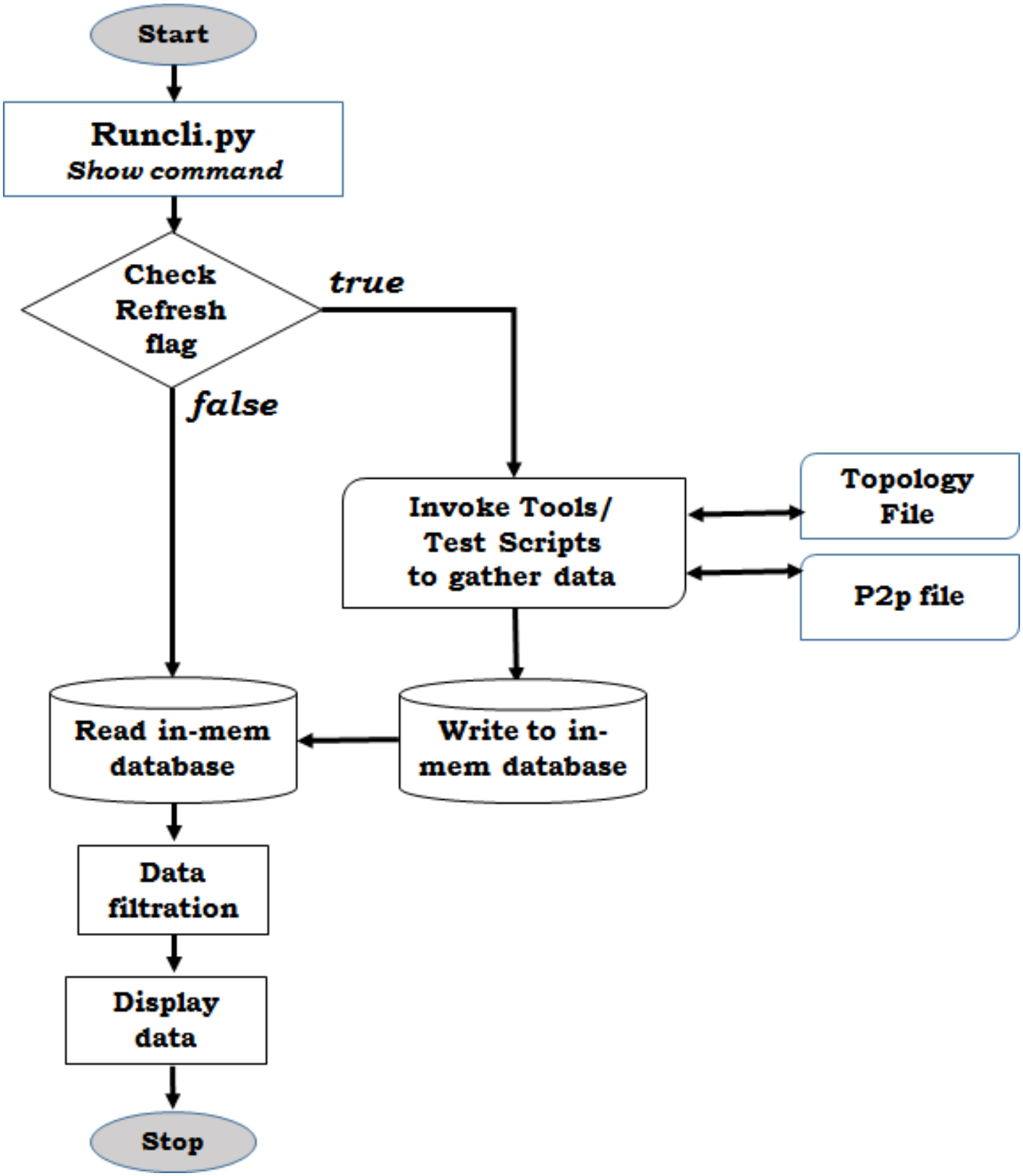
1.5.7.1.1 Build the data model

Scenario 1

The following flow diagram explains the work flow when `show` command is invoked.

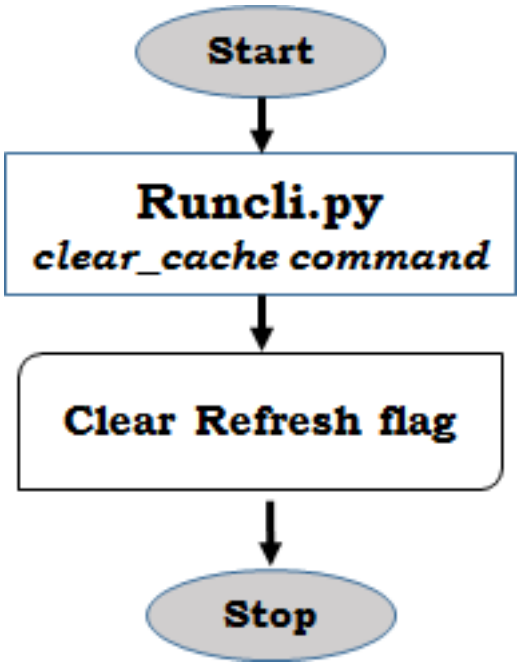
The first time the `show` command is invoked, the refresh flag is set to true by default. Thus, STT would invoke one or more scripts/tools within its scope and mine the data from the target switches.

Subsequently, the scripts might make use of the p2p and topology files available on the system at the default location. The retrieved information is maintained in an in-memory database. Before displaying the content to the user the data is filtered the sub-option specified in show command and displayed in a tabular format.



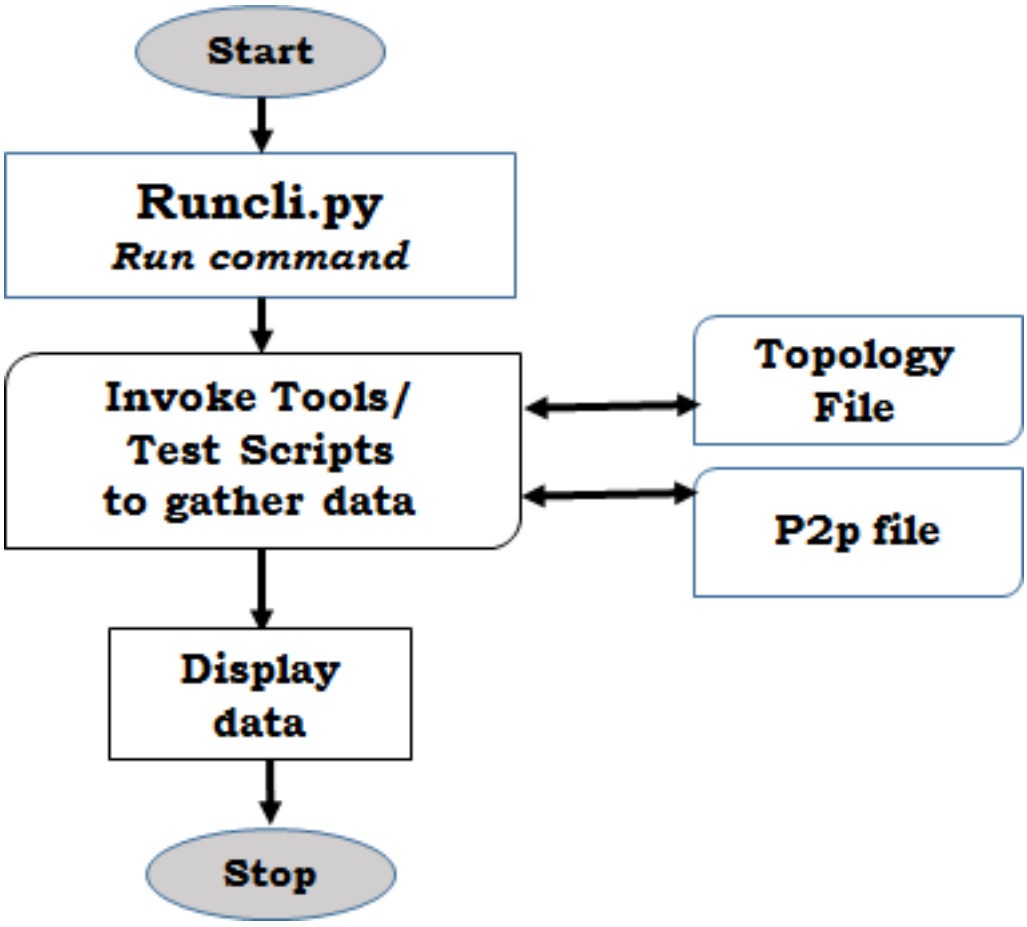
Scenario 2

The following flow diagram explains the work flow when `clear_cache` command is invoked. When the tool is invoked for the first time it gathers the data for each component by invoking corresponding scripts based on the `show` command. The tool would return data from in memory cache for subsequent invocation of `show` command. Users can force the tool to gather fresh data by invoking `clear_cache` command.



Scenario 3:

The following flow diagram explains the work flow when `run` command is invoked. By means of the run command users can invoke diagnostic tools/test scripts from within the scope of STT.



1.5.7.1.2 Update data model

In other words, modify the topology.

STT can help create a topology object and run commands on one or more topology objects.

Topologies can be created by loading from files, creating manually using add commands, or generated using plugins.

Topologies can be saved as a V2 Fabric Template or point-to-point (P2P) file. This section describes various scenarios and use-cases, which use the CLI commands.

Scenario: Convert a P2P to a V2 Fabric Template.

```
load p2p <path_to_current_P2P_file>
save topology <new_file_name>
```

Scenario: Create a template for Colorado switch with all edge links enabled.

```
new "topology/filter" <name>
set_active topology <name>
add switch <switch xname> <switch number> <group number>
add edge <switch xname> 1-24
add edge <switch xname> 100-107
save topology <filename>
```

Scenario: Add an edge link to a V2 Fabric Template.

```
load topology <path_to_current_V2_template_file>
```

For each edge cable (host names are optional):

```
add edge <switch_host_name> <switch_jack_number> <edge_host_name1> <edge_host_name2>
save topology <new_file_name>
```

Scenario: Add a switch with fabric and edge links to a V2 Fabric Template.

There are currently no checks for correct topology. The user must verify that the topology that they have created is correct.

```
load topology <path_to_current_V2_template_file>
add switch <switch_host_name_or_IP> <switch_number> <group_number>
```

For each fabric link:

```
add link <port_xname_1> <port_xname_2>
```

For each edge cable:

```
add edge <switch_host_name> <switch_jack_number> <edge_host_name1> <edge_host_name2>

save topology <new_file_name>
```

Scenario: Run diagnostics on a subset of switches instead of running on all switches.

There are three ways to do this:

1. If the number of switches is small, add fabric components like switch/port manually using “add switch/edge/link ” command to a new topology.

   For instance, if the current default topology has the following switches:

   ```
   (STT) show switches
   ```

xname	type	snum	gnum	edge_count	fabric_count	uptime	up_ports	flap	checkidle	pktIn	pktOut
x3000c0r24	Columbia	0	0	16	8	28 days	16/64	8	idle	5887772588	6203403182
x7000c0r0	Columbia	0	2	16	6	4 days	10/64	5	idle	29504757476	31541437748
x7000c0r1	Columbia	0	3	16	6	28 days	10/64	7	idle	489525220143	497819029438
x9000c1r1	Colorado	0	1	16	30	31 days	46/64	22	idle	94968165307	97659618193
x9000c1r3	Colorado	1	1	16	30	31 days	46/64	22	idle	106562775895	111477837801
x9000c1r5	Colorado	2	1	16	28	31 days	44/64	20	idle	86548533733	89109581331
x9000c1r7	Colorado	3	1	16	28	31 days	44/64	20	idle	106497424137	111719424470
x9000c3r1	Colorado	4	1	16	28	31 days	44/64	20	idle	5582301475	5674157861
x9000c3r3	Colorado	5	1	16	30	31 days	46/64	22	idle	21477841144	22526373110
x9000c3r5	Colorado	6	1	16	28	14 days	44/64	17	idle	1739494433	1746413561
x9000c3r7	Colorado	7	1	16	30	31 days	46/64	22	idle	21114524827	22177522225

   Create a new topology and add the required switches manually using `add switch <switch_name> <switch_num> <group_num>`:

   ```
   (STT) new topology topo1
   STT diags log directory -  /root/slingshot_tools/stt_diags_logs/topo1

   (STT) set_active topology topo1
   ```

```
(STT) add switch x3000c0r24 0 0
Working with 'topo1' topology.

(STT) show switches

+------------+----------+------+------+------------+--------------+---------+----------+------+-----------+------------+------------+
|   xname    |   type   | snum | gnum | edge_count | fabric_count |  uptime | up_ports | flap | checkidle |   pktIn    |   pktOut   |
+------------+----------+------+------+------------+--------------+---------+----------+------+-----------+------------+------------+
| x3000c0r24 | Columbia |  0   |  0   |     0      |      0       | 28 days |  16/64   |  8   |    idle   | 5887773242 | 6203406209 |
+------------+----------+------+------+------------+--------------+---------+----------+------+-----------+------------+------------+
```

2. If the number of switches is big, create a template file or a p2p with the required subset of the original template/p2p and load it using `load topology <path_to_file>` or `load p2p <path_to_file>`.

   For example, consider a system with the following switches:

```
(STT) show switches

+------------+----------+------+------+------------+--------------+---------+----------+------+-----------+--------------+--------------+
|   xname    |   type   | snum | gnum | edge_count | fabric_count |  uptime | up_ports | flap | checkidle |    pktIn     |    pktOut    |
+------------+----------+------+------+------------+--------------+---------+----------+------+-----------+--------------+--------------+
| x3000c0r24 | Columbia |  0   |  0   |     16     |      8       | 28 days |  16/64   |  8   |    idle   |  5887772588  |  6203403182  |
| x7000c0r0  | Columbia |  0   |  2   |     16     |      6       |  4 days |  10/64   |  5   |    idle   | 29504757476  | 31541437748  |
| x7000c0r1  | Columbia |  0   |  3   |     16     |      6       | 28 days |  10/64   |  7   |    idle   | 489525220143 | 497819029438 |
| x9000c1r1  | Colorado |  0   |  1   |     16     |      30      | 31 days |  46/64   |  22  |    idle   | 94968165307  | 97659618193  |
| x9000c1r3  | Colorado |  1   |  1   |     16     |      30      | 31 days |  46/64   |  22  |    idle   | 106562775895 | 111477837801 |
| x9000c1r5  | Colorado |  2   |  1   |     16     |      28      | 31 days |  44/64   |  20  |    idle   | 86548533733  | 89109581331  |
| x9000c1r7  | Colorado |  3   |  1   |     16     |      28      | 31 days |  44/64   |  20  |    idle   | 106497424137 | 111719424470 |
| x9000c3r1  | Colorado |  4   |  1   |     16     |      28      | 31 days |  44/64   |  20  |    idle   |  5582301475  |  5674157861  |
| x9000c3r3  | Colorado |  5   |  1   |     16     |      30      | 31 days |  46/64   |  22  |    idle   | 21477841144  | 22526373110  |
| x9000c3r5  | Colorado |  6   |  1   |     16     |      28      | 14 days |  44/64   |  17  |    idle   |  1739494433  |  1746413561  |
| x9000c3r7  | Colorado |  7   |  1   |     16     |      30      | 31 days |  46/64   |  22  |    idle   | 21114524827  | 22177522225  |
+------------+----------+------+------+------------+--------------+---------+----------+------+-----------+--------------+--------------+
```

   Create a subset p2p which has only the switches you require. For example, here we pick switch x7000c0r0:

```
cat subset_p2p.csv

cable_id,src_conn_a,src_conn_b,dst_conn_a,dst_conn_b,stage,src_egress_a,src_egress_b,dst_egress_a,dst_egress_b,link_type,src_group,\
dst_group,part_number,part_length,calculated_distance,route

7000.7000.01.0000,x7000c0r0j1,none,x7000c0s1b0n3h0,x7000c0s1b0n1h0,1,none,none,none,none,edge,2,n/a,102234600,001.00,000.7878368,[]
7000.7000.01.0001,x7000c0r0j3,none,x7000c0s1b0n5h0,x7000c0s1b0n4h0,1,none,none,none,none,edge,2,n/a,102234600,001.00,000.7878368,[]
7000.7000.01.0002,x7000c0r0j9,none,x7000c0s1b0n6h0,x7000c0s1b0n7h0,1,none,none,none,none,edge,2,n/a,102234600,001.00,000.7878368,[]
7000.7000.01.0003,x7000c0r0j11,none,x7000c0s1b0n8h0,x7000c0s1b0n9h0,1,none,none,none,none,edge,2,n/a,102234600,001.00,000.7878368,[]
7000.7000.01.0004,x7000c0r0j13,none,x7000c0s1b0n3h1,x7000c0s1b0n1h1,1,none,none,none,none,edge,2,n/a,102234600,001.00,000.7878368,[]
7000.7000.01.0005,x7000c0r0j15,none,x7000c0s1b0n5h2,x7000c0s1b0n4h2,1,none,none,none,none,edge,2,n/a,102234600,001.00,000.7878368,[]
7000.7000.01.0006,x7000c0r0j17,none,x7000c0s1b0n6h2,x7000c0s1b0n7h2,1,none,none,none,none,edge,2,n/a,102234600,001.00,000.7878368,[]
7000.7000.01.0007,x7000c0r0j19,none,x7000c0s1b0n8h2,x7000c0s1b0n9h2,1,none,none,none,none,edge,2,n/a,102234600,001.00,000.7878368,[]
```

   Load the subset p2p into a new topology:

```
(STT) new topology topo1
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/topo1

(STT) set_active topology topo1

(STT) load p2p subset_p2p.csv
Working with 'topo1' topology.

Loading p2p from /root/slingshot_tools/subset_p2p.csv
(STT) show switches
+------------+----------+------+------+------------+--------------+--------+----------+------+-----------+-------------+-------------+
|   xname    |   type   | snum | gnum | edge_count | fabric_count | uptime | up_ports | flap | checkidle |    pktIn    |    pktOut   |
+------------+----------+------+------+------------+--------------+--------+----------+------+-----------+-------------+-------------+
| x7000c0r0  | Columbia |  0   |  0   |     16     |      0       | 4 days |  10/64   |  5   |    idle   | 29504780747 | 31541465733 |
+------------+----------+------+------+------------+--------------+--------+----------+------+-----------+-------------+-------------+
```

3. Leverage the `copy` command in STT to copy information from current topology to a new topology and create sub-topologies. Refer to `copy` command documentation above.

   Here are the options that the `copy` command provides:

```
(STT) help copy
Copies switches and switch info from the current topology to another topology
Usage: copy <sub-topology name> <xname1>,<xname2> [--include-neighbors]
    Eg : copy topo1 x3000c1r1
         copy topo1 x1000c1r[1-7], x1000*
         copy topo2 x5000c1r1 --include-neighbors
```

   The following example shows how STT can run on a sub-topology:

   Create a new topology:

```
(STT) new topology topo2
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/topo2
```

Copy the required switches from the default topology to the new topology. Any wildcard representation such as ranges, and '*' can be used as mentioned in the help menu above. Both the wildcard examples are furnished below.

Note: `--include-neighbors` option can also be used, which would copy the switch and all the switches it has links to, to the new topology. Refer to this in detail in the `copy` command documentation above.

```
(STT) copy topo2 x5000c1*
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/topo2

Copied switches ['x5000c1r1', 'x5000c1r3', 'x5000c1r7', 'x5000c1r5'] to new topology topo2

(STT) copy topo2 x5000c3r[1-7]

Copied switches ['x5000c3r1', 'x5000c3r3', 'x5000c3r5', 'x5000c3r7'] to new topology topo2

(STT) set_active topology topo2
(STT)
(STT) show switches

<snip>

Collecting data using 'dgrperfcheck' script.
+-----------+----------+------+------+------------+--------------+---------+----------+------+-----------+------------+------------+-------+
|   xname   |   type   | snum | gnum | edge_count | fabric_count |  uptime | up_ports | flap | checkidle |   pktIn    |   pktOut   | drops |
+-----------+----------+------+------+------------+--------------+---------+----------+------+-----------+------------+------------+-------+
| x5000c1r3 | Colorado |  0   |  0   |     16     |      24      |   1:03  |  36/64   |  0   |    idle   |    2861    |    8925    |   0   |
| x5000c1r7 | Colorado |  1   |  0   |     16     |      24      | 15 days |  34/64   |  16  |    idle   | 8413139568 | 8868241972 |   0   |
| x5000c1r1 | Columbia |  2   |  0   |     0      |      24      |   1:29  |  36/64   |  4   |    idle   |   16821    |   68277    |   0   |
| x5000c1r5 | Colorado |  3   |  0   |     16     |      24      | 15 days |  34/64   |  16  |    idle   |  15019329  |  17229362  |   0   |
| x5000c3r3 | Colorado |  0   |  1   |     16     |      28      | 15 days |  28/64   |  12  |    idle   |  15472448  |  17362716  |   0   |
| x5000c3r5 | Colorado |  1   |  1   |     16     |      28      | 15 days |  28/64   |  12  |    idle   |  15840772  |  17737952  |   0   |
| x5000c3r7 | Colorado |  2   |  1   |     16     |      28      |   3:42  |  28/64   |  8   |    idle   |   10004    |     63     |   0   |
| x5000c3r1 | Columbia |  3   |  1   |     0      |      12      | 15 days |  28/64   |  12  |    idle   |  15227797  |  17085513  |   0   |
+-----------+----------+------+------+------------+--------------+---------+----------+------+-----------+------------+------------+-------+
```

There are currently no checks for correct topology. The user must verify that the topology is correct.

```
(STT) new topology <name>
(STT) set_active topology <name>
(STT) load topology <path_to_V2_template_subset_topology_file>
(STT) refresh_data
(STT) show switches
```

Scenario: Taking a snapshot of STT data

STT has functionality which helps to take a snapshot or a backup of the current STT data model, of the fabric agent, and of compute nodes. The following points explain the individual features of `snapshot_data`.

1. `snapshot_data` command takes a snapshot of the existing data model after a refresh. All the show command outputs will be displayed on the screen. This snapshot can be redirected to a file if necessary.
2. `snapshot_data compute_nodes all|list of CNs` option can be used to take a snapshot of the network configuration of compute nodes.
3. `snapshot_data fmn_fabric` option takes a data backup of the fabric agent.

   NOTE: `snapshot_data` compute nodes option requires an additional RPM named `slingshot-utils` to be installed on compute nodes. Please refer to the pre-requisites section of this README.

```
(STT) help snapshot_data
Take a snapshot of the current STT data model after a refresh.
Usage: snapshot_data - Takes a snapshot after a refresh of the STT data model.
    snapshot_data compute_nodes <all|list of CNs> - Takes a backup/snapshot of network
                                                    config of all/given accessible Compute nodes.
    snapshot_data fmn_fabric - Takes a backup/snapshot of the triage data of
                               fabric agent.
```

Scenario: Specifying a CIDR for IP assignment during Fabric template generation

Using STT, Fabric template files with CIDR IP ranges can be generated. In order to generate such Fabric Templates, STT's "add nics " command can be used as follows:

If you need to exclude IP addresses from being assigned, please use the `--exclude` flag. Example: `--exclude 10.253.1.2,10.253.1.3,10.253.2.2-10.253.2.16`

```
$ slingshot-topology-tool
STT diags log directory - /tmp/stt_diags_logs
```

```
STT diags log directory - /tmp/stt_diags_logs/default
Loading point2point file /opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv to default topology
Loading fabric template file /opt/cray/fabric_template.json to default topology
Welcome to the Slingshot Topology Tool.
General Usage is <command> <arguments>
Type help or ? to list commands.(STT) new topology setup
STT diags log directory - /tmp/stt_diags_logs/setup

(STT) add nics 148.187.112.0/21
Working with 'default' topology.

(STT) save topology /tmp/cidr_template.json
Working with 'default' topology.
Wrote 'default' topology to /tmp/cidr_template.json
```

There is currently no well established way of generating `node_hsn_interface.yaml` from `system_nics.json`.

Scenario: Mapping a Compute Node NIC to a Switch Port

When problems with a specific NIC need to be correlated to its switch port, the two can be linked via the hostname. If you know the NIC's name and you want to map it to the corresponding switch port in STT, you can follow the below steps:

1. Find the xname of the node that the NIC maps to:

   Case 1 : If you only know the nid hostname of the HSN NIC, then you can find the xname by referring to /etc/hosts.

   ```
   $ grep nid001051 /etc/hosts
   10.253.11.81 nid001051-hsn1 x1000c1s4b1n1h1
   ```

   Note: However, /etc/hosts may not be populated in all setups.

   Case 2 : If you have the xname of the HSN NIC, directly proceed to Step 2.

2. At the STT prompt, grep for the xname in `show edge` output.

   ```
   (STT) show edge | grep x1000c1s4b1n1h0

   |      xname      | type |      dst      |     hostname    | port | status |  ptype   | subtype | mtu  |        mac        |   mactype   |  speed
   +-----------------+------+---------------+-----------------+------+--------+----------+---------+------+-------------------+-------------+-------
   | x1000c1r7j104p1 | edge |x1000c1s4b1n1h0| x1000c1s4b1n1h0 |  17  |  True  | Ethernet |   Edge  | 9216 | 02:00:00:00:0b:91 | algorithmic |BS_200G|

   node nid001051 NIC 1 is connected to switch x1000c1r7 port 17
   ```

Note: STT fetches the hostname of a node by directly doing an ssh and verifying the node xname. For example, x9000c1s0b0n0 will be used for hsn interface x9000c1s0b0n0h0.

If the `hostname` field is empty, then it must be due to one of the following:

1. DNS services or /etc/hosts file might not be configured in the system to translate the xname to its equivalent Ethernet based IP address.
2. Passwordless SSH may not be enabled. This can be set up within STT using the `compute_nodes_creds --ssh <ssh private key file>` command.

1.5.8 Disabling diagnostics collection in STT

STT can be used only to create/modify SHCD, Point-to-point and V2 Fabric Templates files by disabling collection of diagnostic in STT. Users can disable diagnostic collection in STT by using the option `--editConfigOnly` while launching the tool.

1.5.9 Increasing SSH timeout value in STT

On bigger systems, in order to avoid SSH connection timeouts while connecting to the switches/CNs through STT, timeout value can be increased using the option `--ssh_conn_timeout` while launching the tool.

```
# slingshot-topology-tool --ssh_conn_timeout 60
Using Fabric Manager URL http://localhost:8000
STT diags log directory -  /root/abhilash/stt_diags_logs
STT diags log directory -  /root/abhilash/stt_diags_logs/default
Loading  point2point file /opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv to default topology
Loading fabric template file /opt/cray/fabric_template.json to default topology
Welcome to the Slingshot Topology Tool v1.2.1-5.
     General Usage is <command> <arguments>
     Type help or ? to list commands.

(STT)
```

1.5.10 Logging mechanism

STT captures the logs generated by scripts in a default/custom specified locations and also exposes a command by which results can be captured into a file as the commands are executed.

1.5.10.1 Script logs

STT invokes one or more scripts upon running the commands and then each of these scripts generates logs. By default these logs are captured under the directory named `stt_diags_logs` under current directory. Users can change the location of log by using the option `--logfile` while launching the tool.

1.5.10.2 Live logs

Within the STT prompt in the cli mode, live logs can be captured by redirecting output of a command to a file.

Capture the logs by redirecting it to a file and subsequently viewing the same using shell command.

```
(STT) show cables >> cable_log
(STT) shell cat cable_log

Working with 'default' topology and 'default' filter profile.
+-------------+-------------+-------------+-------------+--------+------------------+-----------------------------------------------------+
|     srca    |     srcb    |     dsta    |     dstb    |  type  |      status      |                     serial_ids                      |
+-------------+-------------+-------------+-------------+--------+------------------+-----------------------------------------------------+
| x0c0r0j16p1 | x0c0r0j16p0 | x0c0r1j16p1 | x0c0r1j16p0 | fabric |    Connected     | UH294G00846, UH294G00846, UH294G00846, UH294G00846 |
| x0c0r0j17p1 | x0c0r0j17p0 | x0c0r1j24p1 | x0c0r1j24p0 | fabric | Wrong Connection | UH294G00668, UH294G00668, UH294G00948, UH294G00948 |
| x0c0r0j18p1 | x0c0r0j18p0 | x0c0r1j23p1 | x0c0r1j23p0 | fabric | Wrong Connection | UH294G00229, UH294G00229, UH294G00641, UH294G00641 |
| x0c0r0j19p1 | x0c0r0j19p0 | x0c0r1j22p1 | x0c0r1j22p0 | fabric | Wrong Connection | UH294G01026, UH294G01026, UH294G00932, UH294G00932 |
+-------------+-------------+-------------+-------------+--------+------------------+-----------------------------------------------------+
```

1.5.11 Batch mode

Commands can be executed in bulk without having the user get into an interactive mode. STT can be launched along with sub option of –cmdfile

The list of a pre-planned set of commands can be fed to STT in the form of a simple file as illustrated below.

NOTE It is mandatory to load a topology or p2p file using load command for the rest of the commands to use the configuration.

```
(STT) shell cat test_cmds
  load p2p /opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv
  load topology /opt/cray/fabric_template.json
  show switches
  show active topology
  show cables

root@sms1:~ #  slingshot-topology-tool --cmdfile test_cmds
STT diags log directory -  /root/slingshot_tools/stt_diags_logs
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/default
  Working with 'default' topology.
  Working with 'default' topology.

  Working with 'default' topology and 'default' filter profile.
  Collecting data using 'check-switches' script.
  Collecting data using 'dgrerrstat' script.
  Collecting data using 'dgrperfcheck' script.
+--------+----------+------+------+------------+--------------+--------+----------+------+-----------+-------+--------+-------+--------+-------------+
| xname  |   type   | snum | gnum | edge_count | fabric_count | uptime | up_ports | flap | checkidle | pktIn | pktOut | drops | dscrds | mcast_found |
+--------+----------+------+------+------------+--------------+--------+----------+------+-----------+-------+--------+-------+--------+-------------+
| x0c0r0 | Columbia |  0   |  0   |     28     |      36      | 1 day  |   0/64   |  0   |    idle   |   0   |   0    |   0   |   0    |    False    |
| x0c0r1 | Columbia |  1   |  0   |     28     |      36      | 1 day  |   0/64   |  0   |    idle   |   0   |   0    |   0   |   0    |    False    |
+--------+----------+------+------+------------+--------------+--------+----------+------+-----------+-------+--------+-------+--------+-------------+


  Working with 'default' topology and 'default' filter profile.
  Input error, see 'help show'


  Working with 'default' topology and 'default' filter profile.
  Collecting data using 'check-switches' script.
+-------------+-------------+-------------+-------------+--------+------------------+-----------------------------------------------------+
|     srca    |     srcb    |     dsta    |     dstb    |  type  |      status      |                     serial_ids                      |
+-------------+-------------+-------------+-------------+--------+------------------+-----------------------------------------------------+
| x0c0r0j16p1 | x0c0r0j16p0 | x0c0r1j16p1 | x0c0r1j16p0 | fabric |    Connected     | UH294G00846, UH294G00846, UH294G00846, UH294G00846 |
| x0c0r0j17p1 | x0c0r0j17p0 | x0c0r1j24p1 | x0c0r1j24p0 | fabric | Wrong Connection | UH294G00668, UH294G00668, UH294G00948, UH294G00948 |
| x0c0r0j18p1 | x0c0r0j18p0 | x0c0r1j23p1 | x0c0r1j23p0 | fabric | Wrong Connection | UH294G00229, UH294G00229, UH294G00641, UH294G00641 |
| x0c0r0j19p1 | x0c0r0j19p0 | x0c0r1j22p1 | x0c0r1j22p0 | fabric | Wrong Connection | UH294G01026, UH294G01026, UH294G00932, UH294G00932 |
```

```
| x0c0r0j20p1 | x0c0r0j20p0 | x0c0r1j21p1 | x0c0r1j21p0 | fabric | Wrong Connection | UH294G00747, UH294G00747, UH294G00499, UH294G00499 |
+-------------+-------------+-------------+-------------+--------+------------------+----------------------------------------------------+
```

1.5.12 Single mode

Commands can be executed individually or in batch without having the user get into an interactive mode. STT can be launched along with `--cmd` option.

Single commands can be specified as a string or multiple commands can be specified separated by a semicolon as illustrated below.

1.5.13 Default topology

```
# slingshot-topology-tool --cmd "show switches"
STT diags log directory -  /root/slingshot_tools/stt_diags_logs
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/default
Loading  point2point file /opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv to default topology
Loading fabric template file /opt/cray/fabric_template.json to default topology
JSON Validation succeeded

Working with 'default' topology and 'default' filter profile.
Collecting data using 'check-switches' script.
Collecting data using 'dgrerrstat' script.
Collecting data using 'dgrperfcheck' script.
```

xname	type	snum	gnum	edge_count	fabric_count	uptime	up_ports	flap	checkidle	pktIn	pktOut	drops	dscrds	mcast_found
x3000c0r24	Columbia	0	0	12	4	14:29	10/64	0	idle	3362	5445	0	399	True
x5000c1r1	Colorado	0	1	16	30	14:29	30/64	0	idle	383	1818	0	399	True
x5000c1r3	Colorado	1	1	16	30	14:29	30/64	0	idle	384	133	0	1114	True
x5000c1r7	Colorado	2	1	16	28	14:29	28/64	0	idle	384	133	0	1114	True
x5000c3r5	Colorado	3	1	16	28	14:29	28/64	0	idle	384	133	0	1114	True
x5000c3r3	Colorado	4	1	16	28	14:29	28/64	0	idle	384	140	0	1112	True
x5000c3r7	Colorado	5	1	16	28	14:29	28/64	0	idle	384	140	0	1112	True
x5000c3r1	Colorado	6	1	16	28	14:29	28/64	0	idle	384	140	0	1112	True
x5000c1r5	Colorado	7	1	16	28	14:29	28/64	0	idle	383	147	0	1107	True

1.5.13.1 Custom p2p/topology

```
# slingshot-topology-tool -p tests/Shasta_Coke_A5_pt_pt.csv -t Shasta_Coke_topo.json --cmd "show p2p"
STT diags log directory -  /root/slingshot_tools/stt_diags_logs
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/default
Loading  point2point file tests/Shasta_Coke_A5_pt_pt.csv to default topology
Loading fabric template file Shasta_Coke_topo.json to default topology
JSON Validation succeeded

Working with 'default' topology and 'default' filter profile.
```

cable_id	src_conn_a	src_conn_b	dst_conn_a	dst_conn_b	link_type	src_group	dst_group
3000.3000.00.0008	x3000c0r18j15	None	x3000c0r22j15	None	global	0	2
3000.3000.00.0010	x3000c0r18j17	None	x3000c0r20j15	None	global	0	1
3000.3000.00.0014	x3000c0r18j29	None	x3000c0r19j29	None	local	0	0
3000.3000.00.0013	x3000c0r18j30	None	x3000c0r19j30	None	local	0	0
3000.3000.00.0009	x3000c0r19j15	None	x3000c0r22j17	None	global	0	2
3000.3000.00.0011	x3000c0r19j17	None	x3000c0r21j15	None	global	0	1
3000.3000.00.0012	x3000c0r20j17	None	x3000c0r22j19	None	global	1	2
3000.3000.00.0016	x3000c0r20j29	None	x3000c0r21j29	None	local	1	1
3000.3000.00.0015	x3000c0r20j30	None	x3000c0r21j30	None	local	1	1
3000.3000.00.0017	x3000c0r21j17	None	x3000c0r22j21	None	global	1	2
3000.3000.00.0006	x3000c0r20j14	None	x3000c0s11b3n3h0	x3000c0s11b1n1h0	edge	1	None
3000.3000.00.0001	x3000c0r20j16	None	x3000c0s7b0n0h0	x3000c0s5b0n0h0	edge	1	None
3000.3000.00.0005	x3000c0r18j14	None	x3000c0s9b3n3h0	x3000c0s9b1n1h0	edge	0	None
3000.3000.00.0000	x3000c0r18j16	None	x3000c0s3b0n0h0	x3000c0s1b0n0h0	edge	0	None
3000.3000.00.0003	x3000c0r21j16	None	x3000c0s11b4n4h0	x3000c0s11b2n2h0	edge	1	None
3000.3000.00.0002	x3000c0r19j16	None	x3000c0s9b4n4h0	x3000c0s9b2n2h0	edge	0	None
3000.3000.00.0007	x3000c0r22j14	None	x3000c0s13b3n3h0	x3000c0s13b1n1h0	edge	2	None
3000.3000.00.0004	x3000c0r22j16	None	x3000c0s13b4n4h0	x3000c0s13b2n2h0	edge	2	None

1.5.13.2 Multiple commands

```
#slingshot-topology-tool --cmd "set_active output-format CSV; show p2p"

STT diags log directory -  /root/slingshot_tools/stt_diags_logs
STT diags log directory -  /root/slingshot_tools/stt_diags_logs/default
Loading  point2point file /opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv to default topology
Loading fabric template file /opt/cray/fabric_template.json to default topology
JSON Validation succeeded
```

```
Working with 'default' topology and 'default' filter profile.
cable_id,src_conn_a,src_conn_b,dst_conn_a,dst_conn_b,link_type,src_group,dst_group
3000.5000.01.0000,x3000c0r24j1,None,x5000c1r3j23,None,global,0,1
3000.5000.00.0000,x3000c0r24j31,None,x5000c1r1j23,None,global,0,1
5000.5000.00.0054,x5000c1r1j10,None,x5000c3r7j10,None,local,1,1
5000.5000.00.0055,x5000c1r1j16,None,x5000c3r7j16,None,local,1,1
5000.5000.00.0045,x5000c1r1j17,None,x5000c3r3j16,None,local,1,1
5000.5000.00.0045,x5000c1r1j17,None,x5000c3r3j16,None,local,1,1
5000.5000.00.0051,x5000c1r1j18,None,x5000c3r5j16,None,local,1,1
5000.5000.00.0027,x5000c1r1j19,None,x5000c1r7j18,None,local,1,1
5000.5000.00.0027,x5000c1r1j19,None,x5000c1r7j18,None,local,1,1
5000.5000.00.0037,x5000c1r1j20,None,x5000c3r1j16,None,local,1,1
5000.5000.00.0037,x5000c1r1j20,None,x5000c3r1j16,None,local,1,1
5000.5000.00.0015,x5000c1r1j21,None,x5000c1r5j20,None,local,1,1
5000.5000.00.0015,x5000c1r1j21,None,x5000c1r5j20,None,local,1,1
5000.5000.00.0001,x5000c1r1j22,None,x5000c1r3j21,None,local,1,1
```

1.5.14 Compute nodes information from STT

Once STT's access to compute nodes is set up (see Setting up access to compute nodes for STT), launch STT and run commands to check HSN IP and LLDP related configuration on compute nodes. Ping all-to-all connectivity of HSN interfaces between compute nodes and run RoCE based bandwidth and latency checks.

Compute Nodes in Shasta and HPCM systems can have HSN interfaces represented as hsn0/hsn1 or eno1/eno2 etc. or any other name. So, STT needs to be provisioned with appropriate HSN interface prefix based on compute nodes, for compute nodes specific commands to work.

Use `compute_nodes_hsn_prefix` STT command to provision specific hsn interface prefix.

Note: STT uses default HSN prefix as `hsn` if nothing is provisioned.

```
ncn-m001# slingshot-topology-tool
```

Provision STT with compute nodes HSN prefix: e.g. on HPCM systems, compute nodes mostly has HSN interfaces as `eno1/2..`, so provision `eno` as prefix.

```
(STT) compute_nodes_hsn_prefix eno
```

List all compute node related commands:

```
(STT) help show

        show hsn_nodes                                          - Show hsn interfaces of all hsn nodes

        show hsn_nodes lldp                                     - Show lldp info of all hsn nodes

        show hsn_node hsn <node hostname list>                  - Show hsn nodes hsn interface info

        show hsn_node lldp <node hostname list>                 - Show hsn nodes lldp info

        show hsn_traffic ping-all-to-all <nodes_list> <hsn_ip_list> <detailed> - Show all to all connectivity between hsn nodes
        show hsn_traffic roce_perf_check_loopback <nodes_list> - Show RoCE Perf benchmark results at compute node port level
```

Check HSN IP and LLDP configurations at compute nodes.

```
(STT) show hsn_nodes
+----------------+---------+-----------------+----------------------+-----------+-------------+--------------+----------------+--------------+
|    hostname    | if_name |      xname      | accessibility_status | node_type |  if_defined | link_status  | carrier_status | admin_status |
+----------------+---------+-----------------+----------------------+-----------+-------------+--------------+----------------+--------------+
|   nid001001    |   hsn0  | x9000c1s0b0n0h0 |       pingable       |     cn    |   defined   |   detected   |    detected    |      up      |
```

`accessibility_status` is ping status of HSN interface from FMN.

If FMN does not have physical connections to CNs HSN network, `accessibility_status` would be not_pingable.

The output has been truncated for readability.

Show hsn configuration for fewer nodes.

```
(STT) show hsn_node hsn x9000c3s6b0n1,x9000c3s5b0n1
+--------------+--------------+---------+-----------+--------------+------------------+-------------------+-----------------+
|   node_xname | node_hostname | if_name | op_status |  ipv4_addr   |     ipv6_addr    |  config_mac_addr  |  algo_mac_addr  |
+--------------+--------------+---------+-----------+--------------+------------------+-------------------+-----------------+
| x9000c3s6b0n1 |   nid001058   |   hsn3  |     up    | 10.253.9.210 | fe80::ff:fe00:9d2 | 02:00:00:00:09:d2 | no_lldp_mac_tlv |
| x9000c3s6b0n1 |   nid001058   |   hsn2  |     up    | 10.253.9.146 | fe80::ff:fe00:992 | 02:00:00:00:09:92 | no_lldp_mac_tlv |
| x9000c3s6b0n1 |   nid001058   |   hsn1  |     up    | 10.253.9.147 | fe80::ff:fe00:993 | 02:00:00:00:09:93 | no_lldp_mac_tlv |
| x9000c3s6b0n1 |   nid001058   |   hsn0  |     up    | 10.253.9.211 | fe80::ff:fe00:9d3 | 02:00:00:00:09:d3 | no_lldp_mac_tlv |
.
.
```

Show LLDP configuration for compute nodes.

```
(STT) show hsn_nodes lldp
Working with 'default' topology and 'default' filter profile.
+----------------+----------------+-----------+---------------------+-------------------+--------------------+-----------------------+
|    hostname    |     xname      | node_type | accessibility_status | lldp_enable_status | lldp_running_status | lldp_conf_file_status |
+----------------+----------------+-----------+---------------------+-------------------+--------------------+-----------------------+
|    ncn-w003    | x3000c0s11b0n0 |    ncn    |      reachable      |      enabled      |       running      |        corrupt        |
|   nid001001    | x9000c1s0b0n0  |     cn    |      reachable      |      enabled      |       running      |        present        |

'accessibility_status' is ssh status of compute node over management network from FMN.
```

Sample outputs from HPCM managed apollo Systems:

Before proceeding with any compute nodes related commands, please check the compute nodes HSN prefix by logging in to the compute nodes and provision STT with that prefix. e.g. if compute nodes has HSN interfaces as `eno1/eno2`, provision `eno` as HSN prefix.

```
(STT) compute_nodes_hsn_prefix eno

(STT) show hsn_nodes

Working with 'default' topology and 'default' filter profile.
Collecting data using 'check_hsn_nics_config' script.
check_hsn_nics_config  :  Start time: 07/29/2021, 15:46:57 , End time: 07/29/2021, 15:47:04
+------------+----------------------+---------+-----------+---------------+---------------------------+-------------------+-------------------+
| node_xname |    node_hostname     | if_name | hsn_xname |   ipv4_addr   |         ipv6_addr         |  config_mac_addr  |   algo_mac_addr   |
+------------+----------------------+---------+-----------+---------------+---------------------------+-------------------+-------------------+
|  apollo-1  | apollo-1.us.cray.com |   eno1  |  apollo-1 | 172.31.241.141 | fe80::c959:2bb0:1efc:ed6c | a4:bf:01:3e:e0:4d | 3c:2c:30:5e:5d:00 |
|  apollo-1  | apollo-1.us.cray.com |   eno2  |  apollo-1 | not_configured |       not_configured      | a4:bf:01:3e:e0:4e |  no_lldp_mac_tlv  |
|  apollo-2  | apollo-2.us.cray.com |   eno1  |  apollo-2 | 172.31.241.143 | fe80::a243:affd:ede0:19e3 | a4:bf:01:3e:e1:0b | 3c:2c:30:5e:5d:00 |
|  apollo-2  | apollo-2.us.cray.com |   eno2  |  apollo-2 | not_configured |       not_configured      | a4:bf:01:3e:e1:0c |  no_lldp_mac_tlv  |
.
.

* The output in this table has been truncated for readability.

(STT) show hsn_node hsn apollo-1,apollo-2

Working with 'default' topology and 'default' filter profile.
+------------+----------------------+---------+-----------+---------------+---------------------------+-------------------+-------------------+
| node_xname |    node_hostname     | if_name | hsn_xname |   ipv4_addr   |         ipv6_addr         |  config_mac_addr  |   algo_mac_addr   |
+------------+----------------------+---------+-----------+---------------+---------------------------+-------------------+-------------------+
|  apollo-1  | apollo-1.us.cray.com |   eno1  |  apollo-1 | 172.31.241.141 | fe80::c959:2bb0:1efc:ed6c | a4:bf:01:3e:e0:4d | 3c:2c:30:5e:5d:00 |
|  apollo-1  | apollo-1.us.cray.com |   eno2  |  apollo-1 | not_configured |       not_configured      | a4:bf:01:3e:e0:4e |  no_lldp_mac_tlv  |
|  apollo-2  | apollo-2.us.cray.com |   eno1  |  apollo-2 | 172.31.241.143 | fe80::a243:affd:ede0:19e3 | a4:bf:01:3e:e1:0b | 3c:2c:30:5e:5d:00 |
|  apollo-2  | apollo-2.us.cray.com |   eno2  |  apollo-2 | not_configured |       not_configured      | a4:bf:01:3e:e1:0c |  no_lldp_mac_tlv  |
+------------+----------------------+---------+-----------+---------------+---------------------------+-------------------+-------------------+

(STT) show hsn_traffic ping-all-to-all apollo-1,apollo-2 --detailed

Working with 'default' topology and 'default' filter profile.

Displaying the results of ping-all-to-all test between hsn nodes:
+--------------------+----------------------+-----------------------------------------------------------------------------------------------------------------------------+
| hsn_node (xname)   | hsn_iface (ipv4_addr) |                                reachable peer_hsn_nodes (ipv4_addr)                                                         |
+--------------------+----------------------+-----------------------------------------------------------------------------------------------------------------------------+
| apollo-1 (apollo-1) | eno1 (172.31.241.141) |      apollo-2 (172.31.241.143), apollo-4 (172.31.241.147), apollo-6 (172.31.241.151), apollo-5       |
|                    |                      |               (172.31.241.149), apollo-3 (172.31.241.145), apollo-7 (172.31.241.153),                |
+--------------------+----------------------+-----------------------------------------------------------------------------------------------------------------------------+
| apollo-2 (apollo-2) | eno1 (172.31.241.143) |      apollo-1 (172.31.241.141), apollo-4 (172.31.241.147), apollo-3 (172.31.241.145), apollo-7       |
|                    |                      |               (172.31.241.153), apollo-5 (172.31.241.149), apollo-6 (172.31.241.151),                |
+--------------------+----------------------+-----------------------------------------------------------------------------------------------------------------------------+

+--------------------+----------------------+-----------------------------------------+
| hsn_node (xname)   | hsn_iface (ipv4_addr) | non-reachable peer_hsn_nodes (ipv4_addr) |
+--------------------+----------------------+-----------------------------------------+
| apollo-1 (apollo-1) | eno2 (not_configured) |       apollo-2 (not_configured),         |
+--------------------+----------------------+-----------------------------------------+
| apollo-2 (apollo-2) | eno2 (not_configured) |                  None                   |
+--------------------+----------------------+-----------------------------------------+
hsn_traffic ping-all-to-all test:  Start time: 07/29/2021, 15:50:50 , End time: 07/29/2021, 15:50:57

(STT)

* The output in this table has been truncated for readability.

(STT) show hsn_node lldp apollo-1,apollo-2

Working with 'default' topology and 'default' filter profile.
+------------+----------------------+-----------+---------------------+-------------------+--------------------+-------------------------+
| node_xname |    node_hostname     | hsn_xname | accessibility_status | lldp_enable_status | lldp_running_status | lldptool_access_status |
+------------+----------------------+-----------+---------------------+-------------------+--------------------+-------------------------+
|  apollo-1  | apollo-1.us.cray.com |  apollo-1 |      reachable      |      disabled     |       running      |   lldpad_accessible     |
|  apollo-2  | apollo-2.us.cray.com |  apollo-2 |      reachable      |      enabled      |       running      |   lldpad_accessible     |
+------------+----------------------+-----------+---------------------+-------------------+--------------------+-------------------------+
```

```
(STT)

* The output in this table has been truncated for readability.

(STT) show hsn_traffic roce_perf_check_loopback apollo-1,apollo-2

Working with 'default' topology and 'default' filter profile.

Displaying the results of RoCE Perf benchmark tests at compute node port level:
+----------------------+-----------------+--------------------------+---------------------------+--------------------------+---------------------------+
| compute_node (xname) | Mellanox device | ib_read_bw (avg[Gb/sec]) | ib_read_lat (t_avg[usec]) | ib_write_bw (avg[Gb/sec]) | ib_write_lat (t_avg[usec]) |
+----------------------+-----------------+--------------------------+---------------------------+--------------------------+---------------------------+
| apollo-1 (apollo-1)  | mlx5_0          |          100.01          |            1.52           |           99.78          |            0.98           |
+----------------------+-----------------+--------------------------+---------------------------+--------------------------+---------------------------+
| apollo-2 (apollo-2)  | mlx5_0          |          100.44          |            1.71           |           97.98          |            0.99           |
+----------------------+-----------------+--------------------------+---------------------------+--------------------------+---------------------------+

List of nodes having Mellanox5 devices with state as PORT_DOWN:


Configuration used for RoCE Perf benchmarks:
         Size of the message to exchange for bw test (-s)        : 65536 bytes
         Size of the message to exchange for latency test (-s)   : 16 bytes
         TCP port for initial synchronization (-p)               : 18523
         Test duration for bw test in seconds (-D)               : 5 secs

hsn_traffic RoCE perf check loopback:  Start time: 07/29/2021, 15:57:37 , End time: 07/29/2021, 15:57:54
(STT)

* The output in this table has been truncated for readability.
```

The output has been truncated for readability.

```
(STT) show hsn_node lldp x3000c0s5b0n0,x3000c0s3b1n0h0
```

Follow 'help show' for other available commands for compute nodes.

1.5.15 Common problems related to Compute Nodes access from

These are few common problems faced due to wrong setup:

1. STT `show edge` command shows 'hostname' field as 'unknown'. This happens when `/etc/hosts` file at host is not following format specified in step 2 above.

```
(STT) show edge
+-----------------+------+-----------------+----------+----------+--------+----------+---------+----+-----------------+-----------+
|      xname      | type |       dst       | hostname | port_num | status |  ptype   | subtype |mtu |    mac          |  mactype  |
+-----------------+------+-----------------+----------+----------+--------+----------+---------+----+-----------------+-----------+
| x3000c0r42j10p0 | edge | x3000c0s14b1n0h0 | unknown  |    9     |  True  | Ethernet |   Edge  |9216|02:00:00:00:00:08|algorithmic|
+-----------------+------+-----------------+----------+----------+--------+----------+---------+----+-----------------+-----------+
* The output in this table has been truncated for readability.
```

For switch port to endpoint hostname mapping, STT uses p2p file to get `dst_conn_a` and then resolves hostnames based on `/etc/hosts` file.

To fix this problem, update `/etc/hosts` file at host to contain entries in specified format and re-launch STT.

2. STT `show cables` command shows 'serial_ids' for 'dsta' and 'dstb' as 'Connection Failure'.

   This command uses SSH to access compute nodes to fetch serial numbers for HSN cables but if above steps 1 and 2 are not followed properly, we observe this problem.

```
(STT) show cables

Working with 'default' topology and 'default' filter profile.
+-----------------+-----------------+-----------------+-----------------+------+--------------+-----------------------------------------------------------
|       srca      |       srcb      |       dsta      |       dstb      | type |    status    |                          serial_ids
+-----------------+-----------------+-----------------+-----------------+------+--------------+-----------------------------------------------------------
|  x3000c0r42j1p1 |  x3000c0r42j1p0 | x3000c0s26b1n0h0 | x3000c0s25b1n0h0 | edge |    Unknown   | 1019200536, 1019200536, Connection Failure, Connection
+-----------------+-----------------+-----------------+-----------------+------+--------------+-----------------------------------------------------------
* The output in this table has been truncated for readability.
```

To solve this problem, fix hosts `/etc/hosts` file and provision STT with required authentication method and relaunch this command.

3. STT `show hsn_nodes lldp` commands shows `accessibility_status` as `not_reachable`. This is ssh status of compute node over management network from FMN and it could be not reachable due to improper `/etc/hosts` file or use of non-default SSH key pairs for password-less access or other network problems.

   "'bash

(STT) show hsn_nodes lldp +——————-+——————-+————+———————-+——————+——————— +————————+ | hostname | xname | node_type | accessibility_status | lldp_enable_status | lldp_running_status | lldp_conf_file_status | +——————-+——————-+————+———————-+——————+———————+— ———————+ | nid000017-hsn0 | x3000c0s10b0n0 | cn | not_reachable | disabled | not_running | not_present | +——————-+——————-+————+———————-+——————+———————+————————+ * The output in this table has been truncated for readability. "'

To solve this problem, fix `/etc/hosts` file and provision STT with required authentication method and relaunch this command. If the status is still `not_reachable`, then there could be a connectivity problem between FMN and compute node over management network, which needs further debugging.

1.6 Current Known Behaviors, Limitations and Workarounds with Slingshot Topology Tool

1.6.1 SSH key configuration issue impacting running STT for diagnostics

Sometimes, running commands inside STT can result in "SSH remote key changed" error from OpenSSH while communicating with switches using SSH when the remote switch identification changes.

```
(STT) show switch jacks x3000c0r42

Working with 'default' topology and 'default' filter profile.
Collecting data using 'dgrheadshellstat' script.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:QxN9TTUJ/3DqO6K38ch6aXbiqivVc+AmJ9IfS7FIVXY.
Please contact your system administrator.
Add correct host key in /root/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /root/.ssh/known_hosts:4
Password authentication is disabled to avoid man-in-the-middle attacks.
Keyboard-interactive authentication is disabled to avoid man-in-the-middle attacks.
```

1.6.1.1 Workaround/Solution

This is a common SSH issue which occurs while communicating with the switch when the SSH keys change on the remote switch and does not match with the ones in known_hosts file. To resolve this issue, replace old SSH entry of the switch from the known_hosts file using the below command:

ssh-keygen -R xname-of-switch

1.6.2 "Too many open files" error while running STT

Running the slingshot-topology-tool gives "Too many open file errors" and fails to retrieve counter information. Here's a snippet of the error and the command "completes" but the counter information shows "zeroes".

```
(STT) show switches
Working with 'default' topology and 'default' filter profile.
Collecting data using 'check-switches' script.
Collecting data using 'dgrerrstat' script.
Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.
Collecting data using 'dgrperfcheck' script.
An exception occurred while running dgrerrstat : [Errno 24] Too many open files
1.Access to switch x1000c0r3 BMC failed with [Errno 24] Too many open files
1.Access to switch x1000c5r7 BMC failed with [Errno 24] Too many open files
1.Access to switch x1000c7r3 BMC failed with [Errno 24] Too many open files
1.Access to switch x1000c7r7 BMC failed with [Errno 24] Too many open files
```

1.6.2.1 Workaround/Solution

Increase the soft limit of `nofiles` tunable to a higher value as shown below. By default, it is 1024.

```
_echo "*     soft   nofile  65000" >> /etc/security/limits.conf_
```

1.6.3 Authentication error while accessing Compute Node HSN interfaces

While running STT commands which needs access to Compute Nodes, authentication failures can be seen as shown below if STT is not provisioned with right set of Compute Node credentials using compute_nodes_creds command.

```
(STT) show hsn_nodes

Working with 'default' topology and 'default' filter profile.
Collecting data using 'check_hsn_nics_config' script.
Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.

get_accessible_nodes:      : Start time: 12/17/2021, 07:12:25, End time: 12/17/2021, 07:12:28, Total time: 0:00:02.629846
ERROR:pssh.clients.base.parallel:Failed to run on host None - ('Authentication error while connecting to %s:%s - %s', 'x3000c0s07b2n0', 22, Authentication
ERROR:pssh.clients.base.parallel:Failed to run on host None - ('Authentication error while connecting to %s:%s - %s', 'x3000c0s07b1n0', 22, Authentication
ERROR:pssh.clients.base.parallel:Failed to run on host None - ('Authentication error while connecting to %s:%s - %s', 'x3000c0s21b3n0', 22, Authentication
ERROR:pssh.clients.base.parallel:Failed to run on host None - ('Authentication error while connecting to %s:%s - %s', 'x3000c0s23b4n0', 22, Authentication
check_hsn_nics_config      : Start time: 12/17/2021, 07:12:25, End time: 12/17/2021, 07:12:34, Total time: 0:00:08.812230
+---------------+---------------+---------+-----------------+----------------------+-------------+-------------+---------------+-------------+-------
|   node_xname  | node_hostname | if_name |    hsn_xname    | accessibility_status |  if_defined | link_status | carrier_status | admin_status | op_sta
+---------------+---------------+---------+-----------------+----------------------+-------------+-------------+---------------+-------------+-------
| x3000c0s01b1n0 |    unknown    | unknown | x3000c0s01b1n0h0 |       pingable       | not_defined | not_detected |   no_carrier   |     down     |    dow
| x3000c0s01b2n0 |    unknown    | unknown | x3000c0s01b2n0h0 |       pingable       | not_defined | not_detected |   no_carrier   |     down     |    dow

...
```

1.6.3.1 Workaround/Solution

Provide the compute node credentials using either password way or using SSH private key file.

1.6.4 STT shows base address of algorithmic MAC address instead of actual MAC address when host-settings "mirrorSynchronization" is set to TRUE in Fabric Manager

The mirrorSynchronization setting of the host can be obtained by running the following command:

```
# fmctl get /host-settings
+---------------------------+-----------------+
|       KEY                 |       VALUE     |
+---------------------------+-----------------+
| documentSelfLink          | /host-settings  |
| isRunningOnRoseta         | false           |
| mirrorSynchronization     | true            |
| routingUpdateEngine       | true            |
| telemetryRateLimit        |            1000 |
| telemetryRetentionSeconds |            3600 |
+---------------------------+-----------------+
```

When the mirrorSynchronization is set to True in Fabric Manager, from 1.5 release of Slingshot-topology-tool, the "show edge", "show fabric", and "show switch ports" will display the base address of the algorithmic MAC address instead of actual MAC address.

1.6.5 STT 'show edge include_hostname' shows the 'hostname' column as 'unknown'

This behaviour is seen when the data model update is already done by the previous commands and the edge information is coming from data model.

```
(STT) show edge include_hostname

Working with 'default' topology and 'default' filter profile.
+-----------------+------+-----------------+----------+----------+--------+----------+---------+-----+---------+------------+---------+---------+--------
|      xname      | type |       dst       | hostname | port_num | status |  ptype   | subtype | mtu |  speed  |   media    | mcast_a | mcast_b | num_ifc
+-----------------+------+-----------------+----------+----------+--------+----------+---------+-----+---------+------------+---------+---------+--------
| x1000c0r1j100p0 | edge | x1000c0s0b0n1h1 | unknown  |    50    |  True  | Ethernet |   Edge  |  0  | BJ_100G | electrical |    0    |   5073  |
| x1000c0r1j100p1 | edge | x1000c0s0b0n0h1 | unknown  |    51    |  True  | Ethernet |   Edge  |  0  | BJ_100G | electrical |    0    |   3801  |
| x1000c0r1j101p0 | edge | x1000c0s1b0n1h1 | unknown  |    34    |  True  | Ethernet |   Edge  |  0  | BJ_100G | electrical |    0    |   1767  |
| x1000c0r1j101p1 | edge | x1000c0s1b0n0h1 | unknown  |    35    |  True  | Ethernet |   Edge  |  0  | BJ_100G | electrical |    0    |   154   |
| x1000c0r1j102p0 | edge | x1000c0s2b0n0h1 | unknown  |    49    |  True  | Ethernet |   Edge  |  0  | BJ_100G | electrical |    0    |   1800  |
| x1000c0r1j102p1 | edge | x1000c0s2b0n1h1 | unknown  |    48    |  True  | Ethernet |   Edge  |  0  | BJ_100G | electrical |    0    |   2520  |
| x1000c0r1j103p0 | edge | x1000c0s3b0n0h1 | unknown  |    33    |  True  | Ethernet |   Edge  |  0  | BJ_100G | electrical |    0    |   1804  |
| x1000c0r1j103p1 | edge | x1000c0s3b0n1h1 | unknown  |    32    |  True  | Ethernet |   Edge  |  0  | BJ_100G | electrical |    0    |   2003  |
| x1000c0r1j104p0 | edge | x1000c0s4b0n1h1 | unknown  |    17    |  True  | Ethernet |   Edge  |  0  | BJ_100G | electrical |    0    |   9308  |
| x1000c0r1j104p1 | edge | x1000c0s4b0n0h1 | unknown  |    16    |  True  | Ethernet |   Edge  |  0  | BJ_100G | electrical |    0    |   8629  |
...
```

1.6.5.1 Workaround/Solution

Running 'clear_cache' command before running this command will trigger a data model update and the 'hostname' column will be updated properly.

1.6.6 STT command doesn't work as expected with '–fmn_host' option specified

When the Fabric Manager runs on a remote node and if we specify the remote FM details using –fmn_host option while starting STT, STT commands may fail while fetching redfish credentials.

```
(STT) show switches

Working with 'default' topology and 'default' filter profile.
ERROR:root:Cannot find Redfish token file, need to generate token using rosetta_security tools.
ERROR:root:Cannot find Redfish token file, need to generate token using rosetta_security tools.
Traceback (most recent call last):
  File "/usr/lib/python3.6/site-packages/rosetta_dev/redfish_rest_api.py", line 79, in get_rosetta_redfish_credentials
    creds = subprocess.check_output("kubectl get secret rosetta-redfish-token -o json", shell=True, stderr=subprocess.PIPE).decode('utf-8')
  File "/usr/lib64/python3.6/subprocess.py", line 356, in check_output
    **kwargs).stdout
  File "/usr/lib64/python3.6/subprocess.py", line 438, in run
    output=stdout, stderr=stderr)
subprocess.CalledProcessError: Command 'kubectl get secret rosetta-redfish-token -o json' returned non-zero exit status 127.
An exception occurred while running dgrlinkstat : 'NoneType' object is not iterable
dgrlinkstat                : Start time: 12/17/2021, 06:35:53, End time: 12/17/2021, 06:35:54, Total time: 0:00:00.492986
WARNING:root:Access to switch x3002c0r42 BMC failed: 'user'
WARNING:root:Access to switch x3001c0r42 BMC failed: 'user'
WARNING:root:Access to switch x3002c0r41 BMC failed: 'user'
WARNING:root:Access to switch x3001c0r41 BMC failed: 'user'

(STT) show edge

Working with 'default' topology and 'default' filter profile.
Collecting data using 'check-fabric' script.
Collecting data using 'dgrlinkstat' script.
Collecting data using 'dgrperfcheck' script.
Failed to fetch port MTU from agent data. 'actualStatus'
Failed to fetch port MTU from agent data. 'actualStatus'
Failed to fetch port MTU from agent data. 'actualStatus'
Failed to fetch port MTU from agent data. 'actualStatus'
Failed to fetch port MTU from agent data. 'actualStatus'
```

1.6.6.1 Workaround/Solution

Make sure the redfish token file '/root/.config/rosetta_redfish_token' is present. If not, regenerate the redfish token and retry the command.

1.6.7 STT command output redirection does not work sometimes

Redirecting the output of any STT command during data model update is not working.

```
(STT) show switches >switches.out
dgrerrstat                 : Start time: 12/16/2021, 17:19:44, End time: 12/16/2021, 17:20:40, Total time: 0:00:56.087687
dgrperfcheck               : Start time: 12/16/2021, 17:19:44, End time: 12/16/2021, 17:20:58, Total time: 0:01:13.921170
+------------+----------+------+------+------------+-------------+----------+------+--------------+--------------+-------+-----------+-------------+---
|   xname    |   type   | snum | gnum | edge_count | fabric_count | up_ports | flap |     pktIn    |    pktOut    | drops |   dscrds  | mcast_found | tc
+------------+----------+------+------+------------+-------------+----------+------+--------------+--------------+-------+-----------+-------------+---
| x1000c0r1  | Colorado |  0   |  1   |     16     |      34     |  50/64   |  0   | 1320158208100 | 1345679894666 |   0   |    539    |     True    |
| x1000c0r3  | Colorado |  10  |  1   |     16     |      34     |  50/64   |  4   | 1177943079024 | 1202495431707 |   0   |  46606438 |     True    |
| x1000c0r5  | Colorado |  9   |  1   |     16     |      34     |  50/64   |  2   | 1080604908901 | 1104952697235 |  5783 |  23963299 |     True    |
| x1000c0r7  | Colorado |  8   |  1   |     18     |      32     |  49/64   |  1   | 1034324954796 | 1058078415180 |  3120 |  45337439 |     True    |
| x1000c1r1  | Colorado |  11  |  1   |     16     |      34     |  49/64   |  4   |  991864700170 | 1013517811523 | 42239 | 212830874 |     True    |
| x1000c1r3  | Colorado |  12  |  1   |     16     |      34     |  49/64   |  1   | 1123503024561 | 1145996239969 |  8342 |  92809795 |     True    |
| x1000c1r5  | Colorado |  13  |  1   |     18     |      32     |  49/64   |  2   |  860711971897 |  883518997537 |   0   |  91423360 |     True    |
| x1000c1r7  | Colorado |  14  |  1   |     16     |      34     |  50/64   |  2   |  923976787491 |  947554909901 |  5130 |  35691660 |     True    |
| x1000c2r1  | Colorado |  7   |  1   |     16     |      34     |  49/64   |  2   |  891784965288 |  911505485252 |   0   |  64363043 |     True    |
|
...
(STT) shell cat switches.out

Working with 'default' topology and 'default' filter profile.
Collecting data using 'dgrerrstat' script.
Collecting data using 'dgrperfcheck' script.
(STT)
```

1.6.7.1 Workaround/Solution

Re-run the same command again for the redirection to work as given below.

```
(STT) show switches >switches.out
```

```
(STT) shell cat switches.out | more

Working with 'default' topology and 'default' filter profile.
+------------+----------+------+------+------------+--------------+----------+------+--------------+--------------+-------+-----------+------------+---
|   xname    |   type   | snum | gnum | edge_count | fabric_count | up_ports | flap |     pktIn    |    pktOut    | drops |   dscrds  | mcast_found | tc
+------------+----------+------+------+------------+--------------+----------+------+--------------+--------------+-------+-----------+------------+---
| x1000c0r1  | Colorado |  0   |  1   |     16     |      34      |  50/64   |  0   | 1320158208100 | 1345679894666 |   0   |    539    |     True    |
| x1000c0r3  | Colorado |  10  |  1   |     16     |      34      |  50/64   |  4   | 1177943079024 | 1202495431707 |   0   |  46606438 |     True    |
| x1000c0r5  | Colorado |  9   |  1   |     16     |      34      |  50/64   |  2   | 1080604908901 | 1104952697235 |  5783 |  23963299 |     True    |
| x1000c0r7  | Colorado |  8   |  1   |     18     |      32      |  49/64   |  1   | 1034324954796 | 1058078415180 |  3120 |  45337439 |     True    |
| x1000c1r1  | Colorado |  11  |  1   |     16     |      34      |  49/64   |  4   |  991864700170 | 1013517811523 | 42239 | 212830874 |     True    |
```

1.6.8 Some scripts execution time won't get printed while running 'refresh_data'

```
(STT) refresh_data
Refreshing STT data model

NOTE: This action may take a while...

Working with 'default' topology.
Collecting data using 'check-switches' script.
Collecting data using 'dgrheadshellstat' script.
Warning: login credentials for compute nodes is not set in STT.
Use 'compute_nodes_creds' command to input compute node login credentials.
Trying to access compute nodes without password using SSH.

Collecting data using 'dgrerrstat' script.
Collecting data using 'check-fabric' script.
Collecting data using 'dgrlinkstat' script.
Collecting data using 'services_platform' script.
Collecting data using 'dgrperfcheck' script.
Collecting data using 'dgrvalidatesyscfg' script.
Collecting data using 'services_rosetta' script.
dgrvalidatesyscfg          : Start time: 12/16/2021, 17:47:38, End time: 12/16/2021, 17:49:09, Total time: 0:01:30.370956
dgrheadshellstat           : Start time: 12/16/2021, 17:47:38, End time: 12/16/2021, 17:49:09, Total time: 0:01:31.624266
dgrerrstat                 : Start time: 12/16/2021, 17:47:38, End time: 12/16/2021, 17:50:01, Total time: 0:02:23.348458
dgrlinkstat                : Start time: 12/16/2021, 17:47:38, End time: 12/16/2021, 17:50:21, Total time: 0:02:42.986358
```

1.6.8.1 Workaround/Solution

No workaround is available. This behavior does not have major impact on functionality.

1.7 Troubleshooting HSN Links That Are Down

The following sections describe the debug procedure to follow when `fmn_status` indicates downed links. It also provides a recipe for how to open a CAST issue. First, let's talk briefly about why links fail and flap.

1.7.1 Why links go down/why links flap

The software stack responsible for initializing HSN links will also attempt to bring links up in the event that they go down. Several events can occur which may cause a link to go down, such as:

- Physical jostling of the cable itself
- Thermal expansion and contraction
- SerDes firmware corruption or misbehavior
- Cable power or thermal issues
- Cable wire or fiber damage
- ASIC-initiated link down due to: High SerDes error rate, loss of PCS lock or alignment, etc.
- Active Optical Cable (AOC) loss of signal or loss lock.

If the physical layer is partially or fully disconnected, a link may go down and simply not come back up. In that case, it will be reported as down by `linkdbg` or `fmn_status`. In cases where the link is marginal, then it may flap for the reasons above.

1.7.2 Information needed to open a bug

If a problem occurs while following the HSN Debug Procedure, capture system log files by running the script `hsn_triage_capture` on the FMN:

```
$ hsn_triage_capture -h
usage: ./hsn_triage_capture [-h] [-a] [-c] [-f] [-s] [-t TARGET_XNAME[,TARGET_XNAME]]

Collect HSN debug information from the FMN, CMMs and switches,
```

```
  then aggregate into a single tarball.

Note: it is assumed that passwordless ssh is configured to all devices.

optional arguements:
 -h                              Show usage and exit
 -a                              Collect FMN, console, and switch data (default)
 -c                              Collect switch console logs
 -f                              Collect FMN data
 -s                              Collect switch data
 -t TARGET_XNAME[,TARGET_XNAME]  Specific targets only
```

The script `hsn_triage_capture` when run without any options on the FMN will capture state and logs from the FMN, switches and CMMs. When it completes, it prints the path to the tarball of captured data:

```
   Wed Mar 24 20:58:06 UTC 2021: Done! Log tarball is here: /tmp/triage/210324_205805.tar.gz
```

Note that `hsn_triage_capture` assumes network connectivity between the FMN and any CMMs. If the FMN can not communicate with the CMMs, this script will need to be copied to an administrative node that does have connectivity in order to collect the switch console logs. On the admin node, run `hsn_triage_capture` again, this time giving it the `-c` option to collect only the switch console logs and the `-t` option with the list of switches called out when run on the FMN.

```
$ hsn_triage_capture -c -t x1000c1r1b0,x1000c1r3b0,x1000c1r5b0
```

Submit both generated tar files with your bug report.

If you have a very large system you may use the `-t` option to capture data from a subset of the total system.

1.7.3 HSN Debug Procedure

This debug procedure assumes that the cabling is correct. If you are concerned that the cabling is not correct, please read Validate HSN cabling. Please note that when debugging HSN health, we initally focus on the fabric links, because edge links are often unconnected for benign reasons; however, all fabric connections should be cabled up.

1.7.3.1 Step 1: Debug down links

1.7.3.1.1 Initial triage steps

The two utilities used to query system state are `fmn_status` and `linkdbg`. Both `fmn_status` and `linkdbg` can be run directly from the command line of the FMN.

1.7.3.1.2 Quick counts

We suggest that you run `fmn_status` for a short summary of the total number of configured ports that are up vs the total that are supposed to be configured:

```
$ fmn_status
-----------------------------------------
Topology Status
Active: template-policy
Health
------
Runtime:HEALTHY
Configuration:HEALTHY
Traffic:HEALTHY
Security:HEALTHY
For more detailed Health - run 'fmctl get health-engines/template-policy'


Port Policies (online / total ports for each port-policy)
---------------------------------------------------------
cx6-port-policy: 315 / 487
fabric-policy: 1352 / 1356
storage-lag-policy: 120 / 136
offline-policy: 0 / 0
cassini-policy: 0 / 0
edge-policy: 0 / 0
undefined-lag-policy: 0 / 0
cx6-cu-port-policy: 0 / 0


Edge: 435 / 623
Fabric: 1352 / 1356
Ports Reported: 1979 / 1979
Fully Synchronized Switches: 40 / 40
```

The output above contains the summary of Fabric Manager status, but the information most relevant to the debugging of the HSN the report of the `Edge` and `Fabric` ports, which are of the format:

```
port type: number of ports up / number of ports configured to come up
```

Also, the number of `Fully Synchronized Switches` gives us an idea of whether the Fabric Manager has been able to properly configure each switch.

Again, initially, we focus first on the Fabric ports.

1.7.3.1.3 No bad links? Check link health with `show-flaps`.

If all of the ports are up, verify links are not flapping. Please read Show link flap events.

If there are no flapping links, your HSN is fully operational at the L1/L2 network level.

1.7.3.1.4 In the case of downed links,

If you *do* have downed links, run `linkdbg -l`. This will give you a summary of the down ports and their link partners in each of the three categories of links: Edge, Local, and Global. Note that Global and Local links are reported as "Fabric" links.

Using `linkdbg -l` and examining the link pair summary will allow you to prioritize your debug session as to which links you attack first.

Start working on the Fabric links (local and global) first, as this will allow for a more rapid recovery of the overall HSN. A few downed edge links does not affect the efficacy of the HSN, while a few downed fabric links can have a significant impact on performance.

Note that optionally, (again based on the idea we first examine Fabric links), you can filter your results for just fabric links by using the `-L fabric` option.

```
$ linkdbg -l
----------------------------
Downed Links, no diagnostics:
----------------------------
(Edge) x1107c7r3j102p1   <-> x1107c7s2b0n0h0
(Edge) x3000c0r41j24p0   <-> x3000c0s29b4n0h0
(Edge) x3000c0r41j24p1   <-> x3000c0s29b1n0h0
(Edge) x3000c0r42j10p1   <-> x3000c0s29b3n0h0
<...>  # both this example and below truncate the Edge links for brevity
(Edge) x3000c0r42j12p0   <-> x3000c0s29b2n0h0
(Fabric) x1001c7r7j19p0    <-> x1106c7r7j7p0
(Fabric) x1001c7r7j19p1    <-> x1106c7r7j7p1
(Fabric) x3003c0r41j31p0   <-> x3004c0r40j28p0
(Fabric) x3003c0r41j31p1   <-> x3004c0r40j28p1
----------------------------
```

1.7.3.1.5 Detailed link information via linkdbg:

At this point you can make a plan of attack; again, we suggest working on the Fabric links as a first priority. `linkdbg` can be run against all of the above links, but this can be time consuming. If you do decide to run `linkdbg` against all of the down links, you might consider capturing the output by redirecting to a file locally. This will aid in debugging against a system that has many down links; you can refer back to the file instead of running `linkdbg` repeatedly.

Below, see example output for `linkdbg`. The first section is the summary view as seen above. Second is the edge links which are down, and finally the down fabric links.

The value of `linkdbg` is there is an action code for each side of the link, for every link entry in the table. This action code represents the recommended action to get that link to come up. Actions range from bouncing the link to replacing the switch.

```
$ linkdbg
Querying downed links' link partners...
+------+-----------------------------------------+------------+----------+---------------+--------------------+--------------+------------+----------
| type | rosprt xname (pport) <-> link_partner   | rosswinfo  | sC firmW | sw_medtype-pw |      hdsh state    | lp /etc/hosts | node power |      hsn
+------+-----------------------------------------+------------+----------+---------------+--------------------+--------------+------------+----------
| Edge | x1107c7r3j102p1 (48) <-> x1107c7s2b0n0h0  | tpml d S l | 1.5.181  | Electrical-N/A | no errors to report |              |     On     |      ssh
| Edge | x3000c0r41j24p0 (57) <-> x3000c0s29b4n0h0 | tpml d S l | 1.5.181  | Electrical-N/A | no errors to report |              | rfish fail |      ssh
| Edge | x3000c0r41j24p1 (56) <-> x3000c0s29b1n0h0 | tpml d S l | 1.5.181  | Electrical-N/A | no errors to report |              | rfish fail |      ssh
| Edge | x3000c0r42j10p1 ( 8) <-> x3000c0s29b3n0h0 | tpml d s l | 1.5.181  | Electrical-N/A | no errors to report |              | rfish fail |      ssh
<...>  # both this example and above truncate the Edge links for brevity
| Edge | x3000c0r42j12p0 (26) <-> x3000c0s29b2n0h0 | tpml d s l | 1.5.181  | Electrical-N/A | no errors to report |              | rfish fail |      ssh
+------+-----------------------------------------+------------+----------+---------------+--------------------+--------------+------------+----------
```

```
+--------+-----------------------------------------------+------------+----------+----------------+--------------------+-------------+-------------+----
|  type  | rosprt xname (pport) <-> rosprt xname (pport) | rosswinfo  | sC firmW | sw_medtype-pw  |      hdsh state    | lp rosswinfo | lp sC firmW | lp
+--------+-----------------------------------------------+------------+----------+----------------+--------------------+-------------+-------------+----
| Fabric | x1001c7r7j19p0  (41) <-> x1106c7r7j7p0   (10) | tpml d S l | 1.5.181  |   Optical-06   |        RS(ff)      |  tpml d S l  |   1.5.181   |
| Fabric | x1001c7r7j19p1  (40) <-> x1106c7r7j7p1   (11) | tpml d S l | 1.5.181  |   Optical-06   |        RS(ff)      |  tpml d S l  |   1.5.181   |
| Fabric | x3003c0r41j31p0 (49) <-> x3004c0r40j28p0 (37) | tpml d S l | 1.5.181  | Electrical-N/A | no errors to report |  tpml d S l  |   1.5.181   |   U
| Fabric | x3003c0r41j31p1 (48) <-> x3004c0r40j28p1 (36) | tpml d S l | 1.5.181  | Electrical-N/A | no errors to report |  tp-l d S l  |   1.5.181   |   U
+--------+-----------------------------------------------+------------+----------+----------------+--------------------+-------------+-------------+----
```

Below we will discuss in more detail how to read the output above and use it to bring the HSN back to good health.

1.7.3.2 Step 2: Use linkdbg's output to debug iteratively on each downed link.

For extensive details on how linkdbg works and its various options, please read Using linkdbg.

However, for our purposes here in this document, the three most relevant features are described in the next three sections: filtering features, decoding the information in the status/error data columns, and decoding theaction codes in the last two columns.

1.7.3.2.1 Filtering options for linkdbg

To narrow `linkdbg` output, we suggest you use one of the filtering features of `linkdbg`: use `linkdbg`'s `-L` or `-t` options to filter on a subset of link types or switch/jack/ports as follows:

```
#for filtering on link type:
$ linkdbg -L <Fabric, Edge>
#for filtering on switch, jack, or port, respectively
$ linkdbg -t x1001c7r7
$ linkdbg -t x1001c7r7j19
$ linkdbg -t x1001c7r7j19p0
```

In each of the cases above, `linkdbg` will print just the line of output for that particular matching link pair in addition to the accompanying column header information.

1.7.3.2.2 Interpreting the output data columns from linkdbg

Use `run linkdbg -d <column_field_name_without_spaces>` to list the details about table header columns. In the example below, the column for rosswinfo is defined:

```
$ linkdbg -d rosswinfo
rosswinfo
Error/State code interpretation:
    (note that ssh failure indicates that ssh was unable to connect to the controller in question)
rosswinfo:
    switch port's rossw_info file state (tmpl D|d S|s L|l):
        tmpl: (or ---- to indicate error in respective position)
            t: port type has been configured
            p: port has been configured
            m: media (headshell/connector) connected and configured
            l: link has been configured
        d|D: lmon is "dirn up" (d) or "dirn down" (D)
        s|S: serdes is running (s) or not (S)
        l|L: linkstate is up (l) or not (L)
```

For a complete listing of the column headings and their possible error codes, please see linkdbg column definitions.

1.7.3.2.3 Interpreting the action codes from linkdbg

Use `run linkdbg -a <action code>` to list the actions suggested by `linkdbg` for the errors reported in the error and status columns. In example below, the action code for hsh1 (headshell) is defined:

```
linkdbg -a hsh1
Action code hsh1 for hdsh state:
If this link's jack reports "transceiver state = error", or any faults are reported,
then the switch controller is not able to communicate with the cable head.  In this
case, take the following actions, rerunning dgrheadshell in between each one:
- Re-seated HSN cable on the local side, then:
    - Create a CAST issue and distinguish between Columbia and Colorado switches.
    - Run the following from the sC to ensure a reseat has occurred:
        cat /var/log/messages | grep -i "usr event handler" | grep "insert\|remove"
- Replace cable
- Power Cycle the Rosetta on the local side.
    This power cycles the ASIC only, while leaving the sC powered up. Use the command below:
        curl -k -H 'Content-Type: application/json' -d '{"ResetType":"PowerCycle"}' -u <username>:<password>\
        https://<switch bmc xname>/redfish/v1/Chassis/Enclosure/Actions/Chassis.Reset
    Watch ASIC cycle through On -> PoweringOff -> Off -> PoweringOn -> On with the following command (ctrl-C to exit):
        while [ 1 ]; do (curl -k -u <username>:<password> https://<switch bmc xname>/redfish/v1/Chassis/Enclosure) 2>/dev/null | jq .PowerState; sleep 1 ;
```

For a complete listing of the action codes, please see linkdbg action code definitions.

1.7.4 An example workflow

This section will take you though how to use the tools listed above to find and resolve a problem with an HSN link.

As stated in the initial triage steps section, start with the high-level view:

```
$ fmn_status
-----------------------------------------
Topology Status
Active: template-policy
Health
------
Runtime:HEALTHY
Configuration:HEALTHY
Traffic:HEALTHY
Security:HEALTHY
For more detailed Health - run 'fmctl get health-engines/template-policy'


Port Policies (online / total ports for each port-policy)
---------------------------------------------------------
cx6-port-policy: 315 / 487
fabric-policy: 1352 / 1356
storage-lag-policy: 120 / 136
offline-policy: 0 / 0
cassini-policy: 0 / 0
edge-policy: 0 / 0
undefined-lag-policy: 0 / 0
cx6-cu-port-policy: 0 / 0


Edge: 435 / 623
Fabric: 1352 / 1356
Ports Reported: 1979 / 1979
Fully Synchronized Switches: 40 / 40
```

We see some edge and fabric links down, so we run `linkdbg`. We will focus on the first fabric link on the list:

```
$ linkdbg
<snip>
+--------+-------------------------------------------------+------------+----------+---------------+--------------------+-------------+-------------+----
|  type  | rosprt xname (pport) <-> rosprt xname (pport) | rosswinfo  | sC firmW | sw_medtype-pw  |      hdsh state     | lp rosswinfo | lp sC firmW | lp
+--------+-------------------------------------------------+------------+----------+---------------+--------------------+-------------+-------------+----
| Fabric | x1001c7r7j19p0  (41) <-> x1106c7r7j7p0   (10) | tpml d S l | 1.5.181  |   Optical-06   |        RS(ff)       |  tpml d S l  |   1.5.181   |
<snip>
+--------+-------------------------------------------------+------------+----------+---------------+--------------------+-------------+-------------+----
```

Here we have an action code of 'none' on both sides of the link. We check the see linkdbg action code definitions guide for what steps to take in this case. Here, we learn 'none' means `linkdbg` didn't find anything wrong with the configuration. This means we can't jump to an action to fix this link; we will need to read the rest of the columns to see if there's any breadcrumbs to follow.

Starting with `rosswinfo`, we need to check what this data means:

```
$ linkdbg -d rosswinfo
rosswinfo
Error/State code interpretation:
    (note that ssh failure indicates that ssh was unable to connect to the controller in question)
rosswinfo:
    switch port's rossw_info file state (tmpl D|d S|s L|l):
        tmpl: (or ---- to indicate error in respective position)
            t: port type has been configured
            p: port has been configured
            m: media (headshell/connector) connected and configured
            l: link has been configured
        d|D: lmon is "dirn up" (d) or "dirn down" (D)
        s|S: serdes is running (s) or not (S)
        l|L: linkstate is up (l) or not (L)
```

Our link shows an 'S' for SerDes; we now know the SerDes are down on both sides of the link. Similarly, we will query `sw_medtype-pw`. This tells us we've got an AOC with a valid power state. The same is true for the link partner. We've now ruled out the AOC power state as the reason the link is not up.

The next column over is `hdsh state`. `linkdbg -d hdsh` tell us this that the headshell is reporting a `RX LOS`, a receiver loss of signal, across all eight SerDes lanes in the cable. Now we know the SerDes are not tuning and the headshell is not seeing data coming in from the link partner on either side of the link.

To get a bit of additional data, we'll now run `linkdbg` in verbose mode to see what `dmesg` is saying about this link.

```
$ linkdbg -t x1001c7r7j19p0 -v
<snip>
+--------+-----------------------------------------------+------------+----------+---------------+------------+--------------+-------------+--------------
|  type  | rosprt xname (pport) <-> rosprt xname (pport) | rosswinfo  | sC firmW | sw_medtype-pw | hdsh state | lp rosswinfo | lp sC firmW | lp sw_medtype
+--------+-----------------------------------------------+------------+----------+---------------+------------+--------------+-------------+--------------
| Fabric | x1001c7r7j19p0  (41) <-> x1106c7r7j7p0   (10) | tpml d S l | 1.5.181  |   Optical-06  |   RS(ff)   |  tpml d S l  |   1.5.181   |    Optical-07
+--------+-----------------------------------------------+------------+----------+---------------+------------+--------------+-------------+--------------
Verbose output per link:
-----------------------------------------------------------------
(Fabric) x1001c7r7j19p0   phys(41) <-> x1106c7r7j7p0    phys(10)
-----------------------------------------------------------------
Last 30 lines of dmesg log output for x1001c7r7b0:
<snip>
[Thu Mar 25 22:42:15 2021] sbl sbl: p41s0 eye[0] height (0x1) less than requirement (0x25)!

Last 30 lines of dmesg log output for x1106c7r7b0:
<snip>
[Thu Mar 25 22:44:49 2021] sbl sbl: p10s0 eye[0] height (0x0) less than requirement (0x25)!

<snip>
rossw_info entry for x1001c7r7j19p0:
<snip>
serdes: tuning, cnt 3, eff max (2/20), pcal: off
<snip>

rossw_info entry for x1106c7r7j7p0:
<snip>
serdes: tuning, cnt 4, eff max (12/20), pcal: off
<snip>

This link is showing correct software metrics as known by this program. You should inspect the hardware.
```

The output above has been truncated to draw your attention to a few key places. We see in the `dmesg` log from both side of the link that the SerDes eye heights are too small. This is a symptom of a poor tune. We see the driver is indeed continuing to try to tune the SerDes at maximum effort. Finally, `linkdbg` flat out tells us nothing is misconfigured, and it's time to check the hardware.

Referencing the linkdbg guide, the next step for this link would be to attempt to reseat the cable head. If the cable head is not well seated in the headshell, the SerDes may not be able tune to any incoming signal on the cable.

We requested this headshell be reseated on both sides of the link since there's no indication of one side being the source of the problem. In this case, a headshell reseat caused the link to come up and this entry to drop off our list. No additional intervention was needed in software following the reseat.

1.7.5 More information

- General Strategies for debugging HSN links.
- Show link flap events
- Using linkdbg to debug downed links
- linkdbg column definitions
- linkdbg action code definitions
- Validate HSN cabling
- Slingshot Operations Guide

1.8 General strategies for debugging HSN links

In certain situations, error codes or action codes from `linkdbg` may require discussion that is outside the scope of the typical `linkdbg` debugging process for L1 and L2 links.

1.8.1 Ensure HSN switch is online

Use the commands below to query the fabric manager and get the operational state of a target switch:

```
$ TGT_SWITCH=x3011c0r42b0
$ curl -s localhost:8000/fabric/topology-maps/template-map | jq ".fabricStatus[] | select(.switchLink==\"/fabric/switches/$TGT_SWITCH\")" | jq .enabled
"ONLINE"
```

1.8.2 Port and link configuration and media issues

Port and link configuration are required for the link to be brought up properly. Configuration values like port and link speed, initial state, link negotiation protocol and others are required to be set from the Fabric Manager to the switch controller, and if both port and link are not configured, the link will fail to come up.

Misconfiguration can result from the link or port missing from the point to point file or `fabric_template.json` file, or it can result from the information failing to reach the switch controller. The configuration status can be read from the rosswinfo column from `linkdbg`'s output which is gathered from the `sysfs/rossw_info` entry on the switch controller for the particular port. The values returned are parsed and interpreted by `linkdbg`, although the information regarding port and link configuration are reported verbatim as they were read from the `sysfs` entry.

The letters in the configured `tmpl` status code indicate the following:

- t: port type has been configured
- m: media (headshell/connector) is connected and configured
- p: port has been configured
- l: link has been configured

The port, port type, and link (`t`, `p`, and `l`) are configured from the fabric manager on the fmn or pod; the media/headshell/connector (`m`) is configured from the HMS daemons on the switch controller. *It is important to note that the first steps are to make sure that the port and link are configured*...configuring the headshell is important as well, but if the link is missing configuration for the link or port, start there first.

If `t`, `p`, and/or `l` are missing from the code, you will likely need to "bounce" the links by reconfiguring them either manually (light touch) or by restarting the Fabric Manager (big hammer). Last resort is to reset the ASIC and reboot the switch controller (nuclear option). Again, in the case of a configuration issue:

1. Check the files first! It's possible the link is missing from the file.
2. Bounce the links that are down
3. Restart the Fabric Manager and bring up the fabric. See Restart the Fabric Manager.
4. As a last effort, reset the ASIC and reboot the switch controller.

If `m` is missing:

1. Check for issues with the HSN cable; look at the headshell data from `linkdbg`. See Identifiable issues with the QSFP-DD cable.
2. Otherwise ensure that the port is configured in the point to point (p2p) file.
3. Reset the ASIC and reboot the switch controller.

1.8.2.1 Verify link partner HSN port is configured as UP

Edge links: If the link partner is a compute node, verify that the link is trying to come up by logging into the compute node and checking the status of the interface (e.g. for x1002c1s5b1n0h0, we log in to x1002c1s5b1n0):

```
$ ssh root@x1002c1s5b1n0 "ip addr show | grep hsn"
3: hsn0:  mtu 9000 qdisc mq state UP group default qlen 1000
inet 169.0.41.193/12 brd 169.15.255.255 scope global hsn0
4: hsn1:  mtu 9000 qdisc mq state UP group default qlen 1000
inet 169.0.41.129/12 brd 169.15.255.255 scope global hsn1
```

If the output does not show UP for the target interface, verify that the compute resource was configured to bring the interface up.

1.8.2.2 Bounce the links that are down

If the port is in the p2p file, there may have been an issue distributing the configuration using the fabric manager or fabric controller. This is using Fabric Manager:

Fabric Ports:

```
fmn_bounce_offline_edgeports
```

1.8.2.2.1 Edge Ports: 100G or 200G?

NOTE: The following edge link configuration needs to be adapted for edge links that run at 100G or 200G; the following example uses 100G, see below; modify to your needs.

```
EDGE_DOWN_LIST=$(fmn_status --details | grep "Edge" | grep -v "\/" | awk '{ print $2 }')
echo '{"state":"OFFLINE","autoneg":true,"speed":"BJ_100G","precode":"AUTO","flowControl":{"rx":true,"tx":true},"mac":"02:00:00:00:00:00","loopback":"NONE"
curl --request POST -H "Content-Type: application/json" --data @/tmp/edge-offline-port-policy.json http://127.0.0.1:8000/fabric/port-policies
echo '{"portPolicyLinks": ["/fabric/port-policies/edge-offline-port"]}' > /tmp/edge-offline-port-policy.json
echo '{"portPolicyLinks":["/fabric/port-policies/edge-policy"]}' > /tmp/edge-port-policy.json
for i in $EDGE_DOWN_LIST; do curl -X PATCH -H "Content-Type: application/json" -d@/tmp/edge-offline-port-policy.json localhost:8000/fabric/ports/$i ; done
```

```
sleep 5
for i in $EDGE_DOWN_LIST; do curl -X PATCH -H "Content-Type: application/json" -d@/tmp/edge-port-policy.json localhost:8000/fabric/ports/$i ; done
```

1.8.2.3 Restart the Fabric Manager

If the links are not configured correctly, and the issue persists, continue with the following steps.

```
$ systemctl restart fabric-manager
```

Reinitialize the HSN:

```
$ fmn_build_switch_inventory
$ fmn_replace_fabric_policy
$ fmn_replace_fabric_links
```

1.8.2.4 Reset the ASIC and reboot the switch controller

First, reset the switch ASIC. The following command power cycles the ASIC only, while leaving the sC powered up:

```
fmn_switch_reset -k -i <xname>
```

Watch the ASIC cycle through the power states: `On --> PoweringOff --> Off --> PoweringOn --> On` with the following command loop:

```
# type (ctrl-C to exit):
$ while [ 1 ]; do (curl -k -u ${CREDENTIALS} https://${SWITCH_BMC}/redfish/v1/Chassis/Enclosure) 2>/dev/null | jq .PowerState; sleep 1 ; done
```

After running this, log into the switch controller/bmc and reboot.

1.8.3 Other failures

1.8.3.1 Dealing with SerDes tuning issues

If you do not see both rosswinfo codes `s` (`serdes: running`) and `l` (`link state: up`), the SerDes have not been able to tune well, or at all. This could be due to a link partner not being properly configured.

1.8.3.1.1 The link partner is not configured to come up

For example, on one side of the link, the link could be properly configured, but the serdes is not tuning; on the other side of the link, it is missing one of the four configuration status codes (`tpml`).
Essentially, the serdes error in this case is due to the other side of the link not being configured properly.

See Port and link configuration and media issues.

1.8.3.1.2 Identifiable issues with the QSFP cable

There may be identifiable issues with the QSFP-DD cable. This would be the case where `linkdbg` reports that `dgrheadshell` has reported errors.
This information is reported in the error column `hdsh`, and can be decoded by running `linkdbg -d hdsh`.
Also, the action code meanings for headshell issues can be seen anytime by using `linkdbg -a hsh<N>`

See Interpreting the action codes from linkdbg.

1.8.3.2 Link down but no headshell issues on either side of link

There are no headshell issues on either side of link, but link is still down.

It is possible for both sides of the link to be in a state where there are no detectable issues with the cable headshell, but the link does not come up, and the state reported by `rossw_info` will not advance past the `If NOT serdes is running / If NOT link state is up` state.

In this case, perform the following steps. After each step, the entire HSN should be reinitialized. This includes both rebooting and power cycling all switches in the HSN.

1. Re-seat both ends of the HSN cable (fabric link, edge link to another switch), or remove and reinsert compute blade (edge link to compute blade). For 1.2-based sC firmware, a reboot is sometimes needed after a cable reseat. This is not needed for 1.3+.
2. Replace the HSN cable.
3. Power cycle the ASIC on both switches and reboot them.

4. Replace switch on one end.
5. Replace the switch on the other end.

1.9 Show link flap events

A link flap event occurs when a link goes down and then automatically comes back up.

The `show-flaps` tool runs on the FMN and gathers information from each switch in the network. It finds link flap events and displays them to the user. `show-flaps` scores each flapping link and displays them in order of the worst offenders.

1.9.1 When to use `show-flaps`

Use `show-flaps` in three situations:

- after initially assembling a machine in the HPE manufacturing bay,
- after shipping and reassembling the machine at a customer site,
- to spot check machines during their operation once they've been accepted by the customer.

Note that the spot check use case is being built into the architecture as automated monitoring, so this third usage will eventually be deprecated and unnecessary.

1.9.2 Example workflow

Run the following:

```
show-flaps
```

- Go through the `show-flaps` output and look for entries which have a high number of total flaps (20+) AND a high combined score (10+). Reseat these cables. Once the reseat is complete, run the following:

```
show-flaps -S /some/unique-to-you/directory
```

- Wait a couple hours to accumulate system run time, and then run the following:

```
show-flaps -c /some/unique-to-you/directory
```

Note: the `-S` and `-c` options do not save the display output presented to the user to a file; rather, the files it creates for these two options are the metadata it parses to generate the tables it presents. Please redirect to a file anytime you run show-flaps if you want to record the data presented:

```
show-flaps [options] | tee ~/show-flaps.`date +%s`.log
```

- The commands above will show the user the flaps which have occurred since you ran `show-flaps -S <dir>` last: that is, the output that `show-flaps -c /some/unique-to-you/directory` is reporting are the links that flapped between the time `show-flaps -S <dir>` and `show-flaps -c /some/unique-to-you/directory` were run; it's showing you what links have flapped in the last couple hours.
- Go through this `show-flaps` output again, again looking for entries with a high number of total flaps and combined score.
- If any of the cables you've reseated have reappeared on the list, it's time to replace them.
- Continue iterating in this fashion until `show-flaps` shows no entries, or at least no entries which cannot be explained by known hardware actions which are taking place on the machine. Increase the time between the `-S` and `-c` calls to `show-flaps` to find link flaps which are occurring less frequently. Some links may only flap once a day, or even more infrequently.
- An example of a known hardware action would be a node or switch reboot which will cause one or more links to flap. See Links may flap for benign reasons.

1.10 Why do links flap?

A link flap is defined as a link going down and then coming back up. For a full description of why links go down or flap, see Troubleshooting HSN Links That Are Down.

1.10.1 Links may flap for benign reasons

Not all entries in show-flaps represent problems to fix.

For example, a compute node may be rebooted, which will cause an Edge link to flap. There is no corrective action needed here. On a switch, the Rosetta may be power cycled or reinitialized, which will cause the links to go down and come back up later. Again, that is not necessarily a reason for corrective action. So, some common sense is needed to determine which entries in the list need attention. Links which are flapping without any corresponding hardware action are typically the ones which need a closer look.

1.11 How to run show - flaps

Run `show-flaps` from the FMN as follows:

```
# show-flaps
STT diags log directory -  /root/stt_diags_logs/newtopo
Showing links with a flap score of 3 or greater
+-----------------------------------+-----------------+-------------------------------------+-----------------------------------------------+---------
|            xname A<->B            | total flaps A/B |        time since last flap A/B     |                flap period A/B                | flap sco
+-----------------------------------+-----------------+-------------------------------------+-----------------------------------------------+---------
|   x1002c1r1j7p0<->x1004c7r1j15p0  |     52 / 52     |       0:29:57     / 0:29:57         |               5:02:37 / 5:32:21               |     6 /
|   x1003c2r1j7p1<->x1006c2r5j7p1   |    129 / 130    |       1:29:21     / 1:29:22         |               2:00:11 / 1:05:40               |     6 /
|  x1004c3r7j19p1<->x1007c3r7j5p1   |      0 / 0      |   1 day, 0:02:54  / 1 day, 0:02:55  |        2 days, 8:11:20 / 2 days, 8:11:19      |     6 /
|  x3006c0r40j5p1<->x3008c0r42j28p1 |    299 / 299    |      19:10:22     / 19:10:22        |               0:24:12 / 0:24:12               |     2 /
| x1005c3r7j102p0<->x1005c3s2b1n1h0 |     2  / 0      |   1 day, 17:44:41 / 0               |       3 days, 18:48:08 / 0                    |     6 /
<snip>
```

1.11.1 How to interpret show-flaps output

In the example above, we can see that `show-flaps` has identified 4 flapping links. The link is defined as two ports connected by an HSN cable. Below is an explanation of each column:

1.11.1.1 xname A<->B

These are the two port names which are connected together via an HSN cable. Xnames of the format `xXcCrRjJpP` represent switch ports, and xnames of the format `xXcCsSbBnNhH` represent NIC ports (I.E., the last link in the above example has these two formats, respectively: `x1005c3r7j102p0`, and `x1005c3s2b1n1h0`).

1.11.1.2 total flaps A/B

This is the total number of link down events as reported by the Rosetta driver. For NIC ports (as of 03/25/21), this information is not collected and will be 0. Note, the Rosetta driver can reset these counts, so the flap score is NOT based on this data.

1.11.1.3 time since last flap A/B

This is the time which has elapsed since the last link down event has been logged.

1.11.1.4 flap period A/B

Flap period represents the average time between link down events. This is calculated by taking the time elapsed between the oldest run of `show-flaps -S <dir>` and most recent run of `show-flaps -c <dir>` and dividing it by the number of flap events during that window.

1.11.1.5 flap score A/B

A flap score is calculated for each port. The score is a reflection of which ports are currently flapping - not a reflection of how fast they are flapping. A flapping link will score up to six points, with a higher score representing a more unreliable link. Six points means the link is actively flapping; the most recent flap occurred more recently than one flap period in the past. For example, if the link is flapping every 10 minutes, and it last flapped eight minutes ago, then it would score a six. If, on the other hand, it last flapped 12 minutes ago, it would score a five. The idea here is that if the last flap is outside the flap period, there's a greater chance that the flapping has stopped.

The scores for a link (up to 6 points per link partner) rank from 0-12. Flap scores are based on two data points:

First is the `delta` between the last flap and the current time. Second is the `flap period`, which is the time between the first and last flap in the data sample divided by the number of flaps in that sample.

The score is then calculated by determining if the flap period is greater than the delta in 7 grades:

```
6: the delta is less than the flap period
5: the delta is more than the flap period but less than two times the flap period
4: the delta is more than two times the flap period but less than four times the flap period
3: the delta is more than four times the flap period but less than eight times the flap period
2: the delta is more than eight times the flap period
1: At least one flap was reported by the driver
0: No flaps detected
```

Flap history, and thus score, is cleared away when a switch controller is rebooted. No flap history is persisted across a reboot.

1.11.1.6 combined score

This is simply the sum of the two individual flap scores, and is the column by which the `show-flaps` output is sorted in descending order.

1.11.2 Command line arguments

1.11.2.1 -s MIN_SCORE, –min_score MIN_SCORE

Specify minimum combined flap score which will not be filtered out of the output (default 3)

1.11.2.2 -p P2P, –p2p P2P

p2p file location, default `/opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv`

1.11.2.3 -S SAVE_COUNT_DIR, –save_count_dir SAVE_COUNT_DIR

Saves the current flap counts in the specified directory. This allows the users to use the `--compare_count_dir` option to see only new flaps which have occurred since the `--save_count_dir` option was executed.

1.11.2.4 -c COMPARE_COUNT_DIR, –compare_count_dir COMPARE_COUNT_DIR

Show only flaps which have occurred since the `--save_count_dir` option was executed. Provide the same directory as was provided to `--save_count_dir`.

1.11.2.5 -r REMOVE_SAVED_COUNT_DIR, –remove_saved_count_dir REMOVE_SAVED_COUNT_DIR

Deletes saved flap count data in the provided directory.

1.11.2.6 -j FLAP_FILE [FLAP_FILE ...], –flap_files FLAP_FILE [FLAP_FILE ...]

Load flaps from json files rather than querying switches. Takes a list of paths to json files.

1.11.2.7 -J SAVE_FLAPS_TO_DIR, –save_flaps_to_dir SAVE_FLAPS_TO_DIR

Saves the flaps in json form in the specified directory under a new directory named `showflapsjsonfiles`.

1.11.2.8 -F FLAP_FILES_DIR, –flap_files_dir FLAP_FILES_DIR

Directory from which to read the flaps stored in json files per switch.

1.11.2.9 -R, –hard_fail_on_redfish_errors

Redfish errors will prevent show-flaps from presenting partial results by default; just the errors will be displayed. Otherwise, the errors will be displayed in addition to those flaps show-flaps could get according to the other command line option conditions.

1.11.2.10 -v, –version

show program's version number and exit

1.11.2.11 -t, –show_timeline

Shows the complete timeline of link flap events.

1.11.2.12 -n, –omit_nodes

Omit edge links to compute nodes from the output.

1.11.2.13 –fmn_host

Specific FMN host name

1.11.2.14 –fmn_port

Specific FMN host port number

1.11.2.15 –fmn_secure

Use HTTPS if fmn_secure=True else use HTTP

1.11.2.16 -l –target_list

Specify targets: Allows the user to limit the query via a space delimited list of one or more cabinet, chassis, or switch fractional xnames. Note that this will limit the reporting to just the flaps on the target switch(es).

1.11.2.17 -f, –target_list_file

Specify targets: Allows the user to limit the query via a file containing space- or newline-delimited list of one or more cabinet, chassis, or switch fractional xnames. Note that this will limit the reporting to just the flaps on the target switch(es).

1.11.2.18 –not_before

Specify datetime before which flaps will be ignored; use the following format (in quotes!): ”YYYY/mm/dd hh:mm:ss”. Note, this will affect flap score, and may lower it below the default of 3.

1.11.2.19 –num_threads

Specify maximum threads allowed to run simultaneously for redfish queries (default is 32 threads).

1.11.2.20 –num_jobs

Specify maximum jobs allowed to queue in the thread pool to do redfish queries (default is 64 jobs).

1.11.2.21 -D, –debug

Rudimentary debug output flag option, primarily to track threads.

1.11.2.22 -N, –no_status_bar

Suppress status bar to allow for easier stdout redirect.

1.11.2.23 -T, –dbg_time

Gather timestamps and print elapsed time between gather points.

1.12 What to do about links which have a high score in `show-flaps`

At this point, you've run `show-flaps` and have some entries which either show a high number of individual flaps or a high combined score. The corrective action for flapping links is generally a cable reseat operation. The software would have already tried to retune the link after the first flap, so when that cycle continues, we can say the retune is not fixing the problem.

Cleaning flapping links out of the system is an iterative process. Some flapping links will be fixed by a cable reseat, so that's always where we start. Sometimes the reseat does not fix the problem, so we then need to move on to a cable replacement. In rare cases, the switch jack may be damaged, so if a cable replacement doesn't solve the issue, the switch at one or both ends of the cable may need to be replaced.

See the flow described in the "**example workflow**" section at the top of this document for an explanation on how to use `show-flaps` in an iterative manner to clear away flapping links from the system.

1.13 Validate HSN cabling

Launch the Slingshot Topology Tool (STT) with the following command:

```
$ slingshot-topology-tool
```

Please note that when debugging HSN health, we initally focus on the fabric links, because edge links are often unconnected for benign reasons, but all fabric connections should be cabled up. In the following examples, we focus on fabric links. Run the `show cables` command, and check for any fabric links which are not connected.

The following is an example of expected output for a system with no cabling errors

```
(STT) show cables | grep -v "|[ ]*Connected"
Working with default topology and default filter profile.
+----------------+----------------+----------------+----------------+--------+-----------+-------------------------------------------------------------------------
|      srca      |      srcb      |      dsta      |      dstb      |  type  |   status  |                                  serial_ids
+----------------+----------------+----------------+----------------+--------+-----------+-------------------------------------------------------------------------
+----------------+----------------+----------------+----------------+--------+-----------+-------------------------------------------------------------------------
```

For a detailed description of `show cables` and examples of its normal output, please read show cables in the STT section of the trouble shooting guide.

1.14 Using linkdbg to debug downed links

`linkdbg` is run from the FMN and gathers information from each switch in the network to determine what links are down. It then uses the link information to gather the error, status, and state information available on the switch controllers and nodes, which it does through redfish and ssh calls.

1.14.1 When to use linkdbg

linkdbg is used when the user has determined that there is a failure in the HSN either via an observation of performance issues or process failures that indicate communication issues between processes using the HSN.

1.14.2 What information linkdbg provides:

`linkdbg` can display all of the currently downed links, as reported by the fabric agents on all of the system's connected switch controllers. `linkdbg` only knows about links that are reported by the fabric agent; it does so through the Fabric API, and then filters that information through the switch-topology-tool's topology database created from the top level system configuration file, also known as the point to point file.

When run without any arguments, `linkdbg` will display all of the currently downed links with each link partner's associated state via a table where there is one link pair per row, where the columns are error, status, and state codes for each link, and where the last two columns provide action codes suggested to the user as first steps in the debug process for each side of the link.

1.14.3 Why do links fail?

A full description of why links go down is in the intro to Troubleshooting HSN Links That Are Down

1.14.4 How to interpret linkdbg output

1.14.4.1 One link per row

`linkdbg` sorts the links into "Edge" and "Fabric" links. Each Row defines the link partners in a link; the first two columns define the link `type` and `xname (physical port) <-> linkpartner xname` for each partner, and physical ports for each switch link port. In addition to the link partner information, columns contain definitions, error, state, and status codes, as well as action codes for both ports of the link. Here are some screen captures denoting the various fields for each link:

1.14.4.1.1 Edge Link Partners:

Edge links have a switch and node as their respective link partners:

```
+------+-------------------------------------------+
| type | rosprt xname (pport) <-> link_partner     |
+------+-------------------------------------------+
| Edge | x1107c7r3j102p1 (48) <-> x1107c7s2b0n0h0  |
+------+-------------------------------------------+
```

1.14.4.1.2 Fabric Links Partners:

Fabric (local and global) links have switch ports for their respective link partners:

```
+--------+-------------------------------------------------+
|  type  | rosprt xname (pport) <-> rosprt xname (pport) |
+--------+-------------------------------------------------+
| Fabric | x3003c0r41j31p1 (48) <-> x3004c0r40j28p1 (36) |
+--------+-------------------------------------------------+
```

1.14.4.1.3 Switch Port Status Columns:

The first port in the link will always be a switch regardless of its `type`; as such, the next four columns are switch port state: - The `sysfs` info (from rossw_info on switch controller) - Switch controller's firmware version - The switch cable media type - Headshell status parsed from dgrheadshell

```
+------------+----------+---------------+---------------------+
| rosswinfo  | sC firmW | sw_medtype-pw |      hdsh state     |
+------------+----------+---------------+---------------------+
| tpml d S l | 1.5.181  | Electrical-N/A | no errors to report |
+------------+----------+---------------+---------------------+
```

1.14.4.1.4 Edge Link Partner Status Columns:

In an Edge link, the second port is a NIC connection, and there are 3 columns of status for the NIC: - The `/etc/hosts` entry, if applicable or available. - Node power state, (off/on) from redfish endpoints on the node. - The hsn iface, status from the linux utility `ip addr show hsn`

```
+---------------+------------+---------------------+
| lp /etc/hosts | node power |      hsn iface      |
+---------------+------------+---------------------+
|               |     On     |      ssh failed     |
+---------------+------------+---------------------+
```

1.14.4.1.5 Fabric Link Partner Status Columns:

When both partners in the link are switch ports, the four columns of switch error/state information are repeated, but the second set is labeled with the prefix `lp` ("link partner"). Again, the link partner switch have the following columns: - The `sysfs` info (from rossw_info on switch controller) - Switch controller's firmware version - The switch cable media type - Headshell status parsed from dgrheadshell

```
+--------------+-------------+------------------+-----------------------+
| lp rosswinfo | lp sC firmW | lp sw_medtype-pw |     lp hdsh state     |
+--------------+-------------+------------------+-----------------------+
|  tp-l d S l  |   1.5.181   |   Unknown-N/A    | headshell not present |
+--------------+-------------+------------------+-----------------------+
```

1.14.4.1.6 Action Code Columns:

The last two columns are the action codes suggested by linkdbg for both link partners, the switch and its link partner, respectively. Again, the `lp` prefix is used to differentiate.

```
+-------------+----------------+
| action_code | lp action_code |
+-------------+----------------+
|     ros4    |      ros1      |
+-------------+----------------+
```

1.14.4.1.7 -d [DEFINITION_OF_COLUMN], –definition_of_column [DEFINITION_OF_COLUMN]:

Detailed definitions of these columns are available from the command line with the above options: - Allows for printing all or one of the header definitions for each link/error column. - Prints the header definition as well as the possible error codes for each column - Not very forgiving, please print the header name as printed by the program Also, please read linkdbg column definitions for the full listing of column definitions.

1.14.4.1.8 -a ACTION_CODE, –action_code ACTION_CODE:

Action code definitions are available from the command line with the above options: - Defines the last two columns of the default output. - Prints the definitions of the action codes separately from the error codes. - Requires an argument, use `all` to print the possible action codes. - Future update will make the default for no argument to print all action possible codes. Also, please read linkdbg action code definitions for a full listing of action definitions.

1.14.5 Specific Features:

1.14.5.1 Filtering:

Linkdbg has several filtering options:

1.14.5.1.1 -l, –just_down_links:

`linkdbg` can display just the link partner xname pairs using the `-l` option. This is a faster way to get a view the forest, rather than the trees.

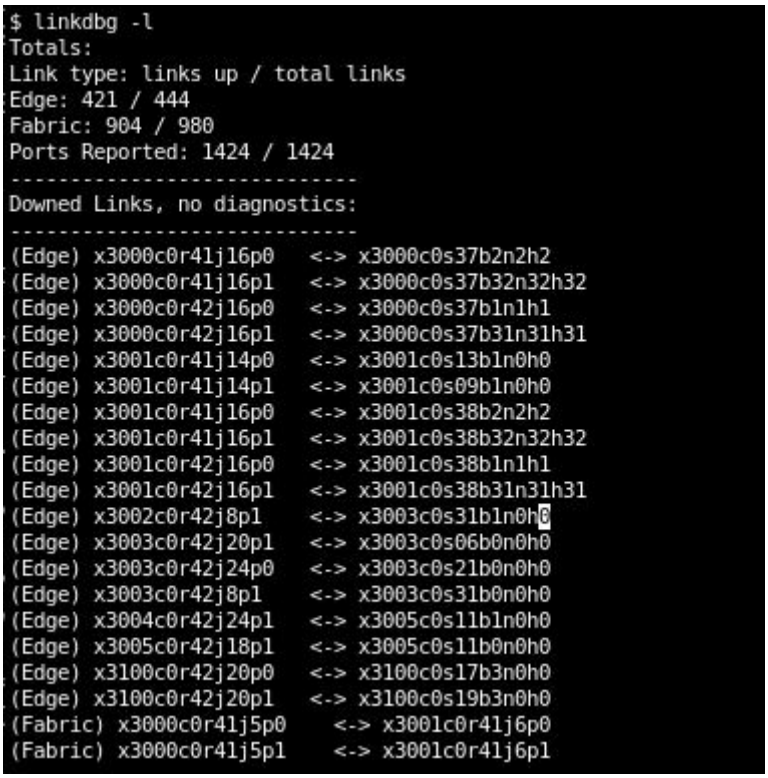


Figure 1: alt text

1.14.5.1.2 -L LINK_TYPE, –link_type LINK_TYPE:

`linkdbg` can also display just the links of a single `type` using the `-L <type>` option. Supported types are link type, fabric (global or local) or edge.

```
$ linkdbg -L Fabric
Totals:
Link type: links up / total links
Edge: 421 / 444
Fabric: 904 / 980
Ports Reported: 1424 / 1424
Querying downed links' link partners...

+--------+-----------------------------------------------+------------+----------+---------------+----------------------+--------
|  type  | rosprt xname (pport) <-> rosprt xname (pport) | rosswinfo  | sC firmW | sw_medtype-pw |       hdsh state     | lp ross
+--------+-----------------------------------------------+------------+----------+---------------+----------------------+--------
| Fabric | x3000c0r41j5p0  ( 5) <-> x3001c0r41j6p0  (21) | tpml d S l | 1.4.407  | Electrical-N/A |  no errors to report  |  tpml d
| Fabric | x3000c0r41j5p1  ( 4) <-> x3001c0r41j6p1  (20) | tpml d S l | 1.4.407  | Electrical-N/A |  no errors to report  |  tpml d
| Fabric | x3000c0r41j6p0  (21) <-> x3001c0r41j5p0  ( 5) | tpml d S l | 1.4.407  | Electrical-N/A |  no errors to report  |  tpml d
| Fabric | x3000c0r41j6p1  (20) <-> x3001c0r41j5p1  ( 4) | tpml d S l | 1.4.407  | Electrical-N/A |  no errors to report  |  tpml d
| Fabric | x3000c0r42j13p0 (28) <-> x3002c0r42j15p0 (15) | tp-- D S L | 1.4.407  |  Unknown-N/A   | headshell not present |  tpml d
| Fabric | x3000c0r42j13p1 (29) <-> x3002c0r42j15p1 (14) | tpml d S l | 1.4.407  |  Unknown-N/A   | headshell not present |  tpml d
| Fabric | x3000c0r42j5p0  ( 5) <-> x3001c0r42j6p0  (21) | tpml d S l | 1.4.407  | Electrical-N/A |  no errors to report  |  tpml d
| Fabric | x3000c0r42j5p1  ( 4) <-> x3001c0r42j6p1  (20) | tpml d S l | 1.4.407  | Electrical-N/A |  no errors to report  |  tpml d
| Fabric | x3000c0r42j8p0  ( 6) <-> x3100c0r40j7p0  (22) | tpml d S l | 1.4.407  |   Optical-06   |  no errors to report  |  tpml d
| Fabric | x3000c0r42j8p1  ( 7) <-> x3100c0r40j7p1  (23) | tpml d S l | 1.4.407  |   Optical-06   |  no errors to report  |  tpml d
| Fabric | x3001c0r41j25p1 (39) <-> x3104c0r39j25p1 (39) | tpml d S l | 1.4.407  |   Optical-06   |  no errors to report  |  tpml d
```

Figure 2: alt text

1.14.5.1.3 -t TARGET_XNAME, –target_xname TARGET_XNAME:

The `linkdbg -t <xname>` option allows you to display just the links associated with a port's partial or full xname: currently chassis, cabinet, slot, jack, and port are supported (cabinet (`xX`), chassis (`xXcC`), slot (`xXcCrR` or `xXcCsS`) are supported. This allows you to examine single links or sets of links in a particular subdivision of the system.

```
$ linkdbg -t x3003c0r42j30p0
Querying downed links' link partners...
+--------+-----------------------------------------------+------------+----------+---------------+----------------------+------
|  type  | rosprt xname (pport) <-> rosprt xname (pport) | rosswinfo  | sC firmW | sw_medtype-pw |       hdsh state     | lp ro
+--------+-----------------------------------------------+------------+----------+---------------+----------------------+------
| Fabric | x3003c0r42j30p0 (50) <-> x3004c0r42j29p0 (34) | tp-- D S L | 1.4.407  |  Unknown-N/A  | headshell not present |  tpml
+--------+-----------------------------------------------+------------+----------+---------------+----------------------+------
Action code ros1 for rosswinfo:
"m" is missing from the configuration quad "tpml":
Check the status of the headshell fields (hdsh state) except in the
case of non compute node endpoints.
```

Figure 3: alt text

1.14.5.2 -T TARGET [TARGET ...], –target_list TARGET [TARGET ...]

List one or more xnames for limiting the query. Must be in xname form, supports mix of cabinet, chassis, switch bmc, or switch port xnames in a space delimited list.

1.14.5.3 -f TARGET_LIST_FILE, –target_list_file TARGET_LIST_FILE

List one or more xnames for limiting the query. Must be in xname form. Supports mix of cabinet, chassis, switch bmc, or switch port xnames in a space delimited list.

1.14.5.4 -x, –exclude_targets

With this option, any targets specified will be excluded from the output. Use this option to exclude all targets given in a target list from examination.

1.14.5.5 Verbose Options:

1.14.5.5.1 -v, –verbose:

- Dumps the relevant links' results from `sysfs` (rossw_info), `dgrheadshell`, and the last 30 lines of `dmesg`.
- The `dgrheadshell` and `dmesg` output is on a per switch basis
- The `sysfs` (rossw_info) output is per port.
- It works in concert with the filtering options so that only the relevant information is presented, no duplication.

1.15 linkdbg action code definitions

Below is the formatted version of the individual action codes printed out by `linkdbg -a <action code>`, which describes each of the actions for various errors as determined by `linkdbg`'s analysis of the errors discovered. The `-a` option does not have a default value, but by giving the value `all` for the action code, you can get a list of the action codes currently defined:

1.15.1 List of all action codes:

1.15.1.1 all

```
valid action codes:
    lp_unresp, lp_cfg_[phys, iface, device, mac]
    ros[1-3]
    aoc[1-2]
    hsh[1-3]
    none
```

1.15.2 Action codes, by code:

1.15.2.1 lp_unresp

Action code lp_unresp for link partners:

```
This link is connected to a link partner that is unresponsive to either ssh
or redfish commands.  Please make sure the link partner is powered up before
proceeding to debug this link. If it is a compute node, it should be booted,
and its HSN interface configured and commanded up.
```

See Ensure HSN switch is online.

1.15.2.2 lp_cfg_phys

Action code lp_cfg_phys for link partners:

```
no signal is detected at the physical level, check link media connections
```

1.15.2.3 lp_cfg_iface

Action code lp_cfg_iface for link partners:

```
network interface is not enabled, run "ifup <device>"
```

1.15.2.4 lp_cfg_device

Action code lp_cfg_device for link partners:

```
network device is not not enabled by the operating system, run "ip link set <device> up"
```

1.15.2.5 lp_cfg_mac

Action code lp_cfg_mac for link partners:

```
algorithmic mac address is not set, node hsn needs to have its AMA configured properly.
```

1.15.2.6 ros1

Action code ros1 for rosswinfo:

```
"m" is missing from the configuration quad "tpml":
Check the status of the headshell fields (hdsh state) except in the
case of non compute node endpoints.
```

1.15.2.7 ros2

Action code ros2 for rosswinfo:

```
The configuration quad "tpml" is incomplete:
For 1.3+:
    Ensure port is configured in the point to point file.
```

1.15.2.8 ros3

Action code ros3 for rosswinfo:

```
Ensure the fabric manager has been configured to initialize the HSN.
As a workaround, bounce the link; see section 1.2.4.4.1 of the Confluence
page:
https://connect.us.cray.com/confluence/display/SASSHOT/Troubleshooting+and+Debugging+Downed+High-Speed+Networ
```

1.15.2.9 ros4

Action code ros4 for rosswinfo:

```
The SerDes have not been able to tune well, or at all. This could be
due to one of the following reasons:
    The link partner is not configured to come up (check hsn iface field),
    There are identifiable issues with the QSFP-DD cable (check hdsh status field), or
    There are silent issues with the QSFP-DD cable:
        The following action should be taken, in order. After each step,
        the entire HSN should be reinitialized. This includes both rebooting
        and power cycling all switches in the HSN:
            -Re-seat both ends of the HSN cable (fabric link, edge link to
            another switch), or remove and reinsert compute blade (edge
            link to compute blade).
            -Replace HSN cable.
            -Power cycle and reboot all switches.
            -Replace switch on both ends, one at a time.
```

1.15.2.10 aoc1

Action code aoc1 for sw_medtype/pw state:

```
Unable to communicate with AOC via i2c. Reseat the cable. If the problem
persists, attempt the following resolutions in order: power cycle the
slot, replace the cable, replace the switch.
```

1.15.2.11 aoc2

Action code aoc2 for sw_medtype/pw state:

```
AOC/optical cables need to be in high power mode.  It is possible you
caught the link in a transient state.  Try re-running this diagnostic
on that link alone.  See help menu.

Otherwise, valid high power modes are 07 and 06. Any other values
indicates a problem.  This action code indicates that this cable is not
in high power mode, so open a CAST and be sure to include the output of
```

```
the command below (in addition to the output from hsn_triage_capture):
    ssh root@<switchbmc_xname_here> "qdddump -p 0 <switch_jack_number_here>"
```

1.15.2.12 hsh1

Action code hsh1 for hdsh state:

```
If this link's jack reports "transceiver state = error", or any faults are reported,
then the switch controller is not able to communicate with the cable head.  In this
case, take the following actions, rerunning dgrheadshell in between each one:
- Re-seated HSN cable on the local side, then:
    - Create a CAST issue and distinguish between Columbia and Colorado switches.
    - Run the following from the sC to ensure a reseat has occurred:
        cat /var/log/messages | grep -i "usr event handler" | grep "insert\|remove"
- Replace cable
- Power Cycle the Rosetta on the local side.
    This power cycles the ASIC only, while leaving the sC powered up. Use the command below:
        curl -k -H 'Content-Type: application/json' -d '{"ResetType":"PowerCycle"}' -u <username>:<password>\
        https://<switch bmc xname>/redfish/v1/Chassis/Enclosure/Actions/Chassis.Reset
    Watch ASIC cycle through On -> PoweringOff -> Off -> PoweringOn -> On with the following command (ctrl-C
        while [ 1 ]; do (curl -k -u <username>:<password> https://<switch bmc xname>/redfish/v1/Chassis/Enclo
```

1.15.2.13 hsh2

Action code hsh2 for hdsh state:

```
dgrheadshell is reporting that the switch cable heads are not seeing a signalcoming in on the local (TX) side
    If so, the cable should be re-seated on the local end.
    If this error persists after a cable re-seat, the cable should be replaced.
```

1.15.2.14 hsh3

Action code hsh3 for hdsh state:

```
dgrheadshell is reporting that the switch cable heads are not seeing a signalcoming in on the remote (RX) sid
    If so, the cable should be re-seated on the remote end.
    If this error persists after a cable re-seat, the cable should be replaced.
```

1.15.2.15 none

Action code none:

This link is showing correct software metrics as known by this program. You should inspect the hardware.

1.16 linkdbg column definitions

Below is a formatted version of the output from the command `linkdbg -d`, which describes each of the column headers and the possible error codes that will appear in them. Note that each of these column headers can be given as an argument to `linkdbg -d <column>` to abbreviate the output and focus on the meaning of one column's error codes.

1.16.1 Error/State code interpretation

Note that ssh failure indicates that ssh was unable to connect to the controller in question. Also, any column headed with "lp" indicates it is the state for the link partner, or the second link in the pair.

1.16.1.1 link type:

```
can be edge, local, or global
```

1.16.1.2 rosprt xname (pport) <-> link_partner

```
Rosetta port name and its physical port <-> link partner.
Note that the link partner can be another Rossetta port (with
physical port) or another type of partner, in which case a
physical port is not listed, but "na" (not applicable) is printed
for the link partners "physical" port.
```

1.16.1.3 rosswinfo:

```
switch port's rossw_info file state (tmpl D|d S|s L|l):
    tmpl: (or ---- to indicate error in respective position)
        t: port type has been configured
        p: port has been configured
        m: media (headshell/connector) connected and configured
        l: link has been configured
    d|D: lmon is "dirn up" (d) or "dirn down" (D)
    s|S: serdes is running (s) or not (S)
    l|L: linkstate is up (l) or not (L)
```

1.16.1.4 sC firmW:

```
switch controller firmware version in the form: major.minor.version
```

1.16.1.5 sw_medtype/pw:

```
This is a composite code: the switch jack media type and its
power state if it's an optical cable.  It is reported primarily
for the case of an AOC, or optical cable, but since it's there,
we can report the type for all switch jacks.  Note that only power
state for optical connectors is reported. The following states
are reported for this column:
    Optical-XX: this is an AOC cable, and the valid power states(XX) are 0x07 and 0x06.
    Electrical-N/A: this is an electrical cable, we do not report power state for these.
    rfish 404: bad URI.
    rfish fail: Most likely a bad field designator given to parse
    the json info returned for the URI.
```

1.16.1.6 hdsh state:

```
headshell status from dgrheadshell (C F RL(xx) RS(xx) TL(xx) TS(xx)):
    C: transceiver state = error
    F: Module State = Fault
    RL(xx): RX_LOL_flag value (in hex)
    RS(xx): RX_LOS_flag value (in hex)
    TL(xx): TX_LOL_flag value (in hex)
    TS(xx): TX_LOS_flag value (in hex)
    (use -d, -v and -t <target> options for more information)
Other values reported are:
    headshell not present
    no errors to report
```

1.16.1.7 lp /etc/hosts:

```
link partner's etc host name
```

1.16.1.8 node power:

```
Redfish query of link partner Node PowerState:
On/Off: are the only two valid values,
```

```
rfish fail: Redfish query failed,
json fail: json field queried via Redfish does not exist,
rfish 404: Redfish URI does not exist.
```

1.16.1.9 hsn iface:

```
This is output from "ip addr show" for the link partner node's hsn:
phup: LOWER_UP in ip output: physical level has a signal
PHDN: NO-CARRIER, no signal is detected at the physical level
ifup: UP in ip output: link was ifup'd, and the network interface is enabled
IFDN: no "UP" in ip output: link was ifdown'd down, network iface is disabled
dvup: "state UP" device is enabled by operating system
DVDN: "state DOWN" device is disabled by operating system
algo: The mac address has 02 in the first couplette indicating it's an algorithmic macaddr.
MAC : The mac address does not have 02 in the first place, indicating that it's
      not configured correctly.
```

1.16.1.10 action:

```
This column is to report a single action code based on internal diagnostics
that analyze the state reported.  For example, if the rossw_info reports
that the link has not been configured (if not tpml, but rather tp-- or
some variant thereof), you'll see an action code of "ros1".  Actions are
prioritized based on prerequisites for link health, and thus only one
action is listed per link.  Fix the problem indicated by the action,
then re-run this program for that specifc link (using the -t or --target
argument listed in the help/usage output).

Also, there is a separate decoder argument availble (-a or --action_code) you
use with the code to determine the action required. To continue with
the previous example, you can run this program with "-a ros1" to
determine the required action.

Lastly, this aspect of this program is a work in progress; as such,
please find more information on debugging links at:
https://connect.us.cray.com/confluence/display/SASSHOT/Troubleshooting+and+Debugging+Downed+High-Speed+Networ
```

1.17 Domain Name System (DNS) Troubleshooting

1.17.1 Check for DNS Errors

Starting in Shasta v1.5 / COS 2.1, there is a script which you can run to diagnose problems with a system's network (Provided by DVS Team). It detects DNS and network interface mismatches.

1. Copy dvs network validation script from worker node01 to master node01

   ```
   ncn-m001:~ # cd slingshot/
   ncn-m001:~/slingshot # rsync -av ncn-w001:/opt/cray/dvs/default/sbin/dvs_validate_network .
   receiving incremental file list
   dvs_validate_network

   sent 43 bytes  received 7,060 bytes  14,206.00 bytes/sec
   total size is 6,960  speedup is 0.98
   ncn-m001:~/slingshot #
   ```

2. Then run dvs_validate_network.

   Check for errors, if you need more information on the error look at first part of the output.

   ```
   ncn-m001:~/slingshot # ./dvs_validate_network
   Getting token to authenticate with the HSM...
   Querying the HSM...
   Getting SSH key for connecting to CNs/UANs...
   Iterating over components returned by HSM...

   Examining:
   {
   ```

```
    "Arch": "X86",
    "Class": "River",
    "Enabled": true,
    "Flag": "OK",
    "ID": "x3000c0s19b2n0",
    "NID": 2,
    "NetType": "Sling",
    "Role": "Compute",
    "SoftwareStatus": "DvsAvailable",
    "State": "Ready",
    "Type": "Node"
}
nodename: nid000002-nmn
NMN: x3000c0s19b2n0 -> 10.252.1.15
NMN: 10.252.1.15 -> x3000c0s19b2n0
NMN interface nmn0 has expected IP address 10.252.1.15
HSN: x3000c0s19b2n0h0 -> 10.253.0.11
HSN: 10.253.0.11 -> x3000c0s19b2n0h0
Error: nid000002-nmn: HSN interface hsn0 does not have expected IP address 10.253.0.11
4: hsn0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP group default qlen 1000
    link/ether 02:00:00:00:00:06 brd ff:ff:ff:ff:ff:ff
    inet 10.253.0.6/16 scope global hsn0
    valid_lft forever preferred_lft forever
    inet6 fe80::ff:fe00:6/64 scope link
    valid_lft forever preferred_lft forever

...
...

Warnings = 0

Errors = 4
ncn-w001-nmn: HSN xname "x3000c0s7b0n0h0" does not map to an IP address
ncn-w002-nmn: HSN xname "x3000c0s9b0n0h0" does not map to an IP address
nid000002-nmn: HSN interface hsn0 does not have expected IP address 10.253.0.11
nid000003-nmn: HSN interface hsn0 does not have expected IP address 10.253.0.6
```

Above Example we have an error on nid000002-nmn ,it expecting 10.253.0.11 but its configured with 10.253.0.6.

Step by step explanation is provided below.

we need to fix either the P2P file and re-apply DNS Config.

3. Decode above output.

   For each error line, you can scroll back up and find the section talking about the node to get more info.

   For example, let's look to see what's causing this error:

   ```
   nid000002-nmn: HSN interface hsn0 does not have expected IP address 10.253.0.11
   ```

   If we look at the script output, we'll find the section for nid000002, which is:

   ```
   Examining:
    {
        "Arch": "X86",
        "Class": "River",
        "Enabled": true,
        "Flag": "OK",
        "ID": "x3000c0s19b2n0",
        "NID": 2,
        "NetType": "Sling",
        "Role": "Compute",
        "SoftwareStatus": "DvsAvailable",
        "State": "Ready",
        "Type": "Node"
    }
    nodename: nid000002-nmn
    NMN: x3000c0s19b2n0 -> 10.252.1.15
    NMN: 10.252.1.15 -> x3000c0s19b2n0
    NMN interface nmn0 has expected IP address 10.252.1.15
    HSN: x3000c0s19b2n0h0 -> 10.253.0.11
    HSN: 10.253.0.11 -> x3000c0s19b2n0h0
    Error: nid000002-nmn: HSN interface hsn0 does not have expected IP address 10.253.0.11
    4: hsn0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP group default qlen 1000
        link/ether 02:00:00:00:00:06 brd ff:ff:ff:ff:ff:ff
        inet 10.253.0.6/16 scope global hsn0
        valid_lft forever preferred_lft forever
        inet6 fe80::ff:fe00:6/64 scope link
        valid_lft forever preferred_lft forever
   ```

   The first part, the part in curly braces, is the Hardware State Manager (HSM) data that describes the node. The big things here which the script cares about there are the Type, Role, Subrole (if there is one), and ID. It use the Type, Role, and Subrole to only query nodes which might actually run DVS (CNs, UANs, and Workers).

The script then uses DNS to find out the nodename that corresponds to the ID field (which is actually an xname). That's represented by this line:

```
nodename: nid000002-nmn
```

Then, it takes the ID/xname and makes sure it has a one-to-one mapping to an IP address. That's these lines:

```
NMN: x3000c0s19b2n0 -> 10.252.1.15
NMN: 10.252.1.15 -> x3000c0s19b2n0
```

Then, it does a SSH into the node and runs "ip a s nmn0" and makes sure the IP address returned matches the IP address from the previous step. That's this line:

```
NMN interface nmn0 has expected IP address 10.252.1.15
```

Then, it appends "h0" on to the xname from the HSM to generate the HSN xname. It repeats the process with the new xname. The one-to-one mapping check gives us these lines:

```
HSN: x3000c0s19b2n0h0 -> 10.253.0.11
HSN: 10.253.0.11 -> x3000c0s19b2n0h0
```

It SSHes into the node and does a "ip a s hsn0" to make sure the hsn0 has the IP address it should, but in this case, it sees a mismatch. The "Error" line says the IP address on hsn0 didn't match what was expected. It's the same line that gets printed at the end of the script output in the "Errors" section. The debugging information is the output of the "ip a s hsn0". As you can see in this case, DNS is saying the IP address is "10.253.0.11", but hsn0 on the node actually has the address "10.253.0.6".

```
Error: nid000002-nmn: HSN interface hsn0 does not have expected IP address 10.253.0.11
4: hsn0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP group default qlen 1000
     link/ether 02:00:00:00:00:06 brd ff:ff:ff:ff:ff:ff
     inet 10.253.0.6/16 scope global hsn0
     valid_lft forever preferred_lft forever
     inet6 fe80::ff:fe00:6/64 scope link
     valid_lft forever preferred_lft forever
```

1.17.2 Fix P2P file and re-apply DNS Config

1. Backup Existing Config and Files. See Backup of a Fabric Manager Configuration for more information.

   login into slingshot fabric manager pod. Do `fmn-backup`, copy config and backup files outside container/fmn.

   ```
   ncn-m001:~/slingshot # kubectl exec -it -n services slingshot-fabric-manager-7dccbfb48c-kqkx7 -- bash

   slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot # fmn-backup
   INFO: Initiating Fabric Manager DB Backup Operation to File /opt/slingshot/backup/fm-fabric-manager-backup-2021-08-03_21-10-53.zip
   INFO: Initiating Fabric Manager Keystore Backup Operation to File /opt/slingshot/backup/fm-java-keystore-backup-2021-08-03_21-10-53.tar
   INFO: Initiating Fabric Manager Slingshot Keystore Backup Operation to File /opt/slingshot/backup/slingshot-keystore-2021-08-03_21-10-53
   INFO: Creating Fabric Manager Backup file /opt/slingshot/backup/fm-fabric-backup-2021-08-03_21-10-53.tar
   INFO: Creating Fabric Manager Backup Checksum file /opt/slingshot/backup/fm-fabric-backup-2021-08-03_21-10-53.tar-checksum
   INFO: Fabric Backup File Checksum:173469797 Checksum Result File:/opt/slingshot/backup/fm-fabric-backup-2021-08-03_21-10-53.tar-checksum
   SUCCESS: Fabric Backup Operation Backup File:/opt/slingshot/backup/fm-fabric-backup-2021-08-03_21-10-53.tar

   slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot # cp /opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv /opt/slingshot/data/Backup_2021_8_3_Sha
   slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot # cp /opt/cray/fabric_template.json /opt/slingshot/data/Backup_2021_8_3_fabric_template.js
   slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot # exit
   exit

   ncn-m001:~/slingshot # kubectl cp services/slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot/backup/fm-fabric-backup-2021-08-03_21-10-53.tar
   ncn-m001:~/slingshot # kubectl cp services/slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot/data/Backup_2021_8_3_Shasta_system_hsn_pt_pt.cs
   ncn-m001:~/slingshot # kubectl cp services/slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot/data/Backup_2021_8_3_fabric_template.json Backu
   ncn-m001:~/slingshot # ls -latr
   -rw-r--r-- 1 root root 225280 Aug  3 21:16 fm-fabric-backup-2021-08-03_21-10-53.tar
   -rw-r--r-- 1 root root   1567 Aug  3 21:17 Backup_2021_8_3_Shasta_system_hsn_pt_pt.csv
   -rw-r--r-- 1 root root  19734 Aug  3 21:17 Backup_2021_8_3_fabric_template.json
   drwxr-xr-x 2 root root    225 Aug  3 21:17 .
   ncn-m001:~/slingshot #
   ```

2. fix P2P file as per your requirements and system setup.

3. login back into slingshot fabric manager pod. following steps will be run inside the container.

   ```
   ncn-m001:~/slingshot # kubectl exec -it -n services slingshot-fabric-manager-7dccbfb48c-kqkx7 -- bash
   ```

4. check the ip address range for dns. In following example its `10.253.0.0/16`

   ```
   slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot # fmn_shasta_dns -p
    HSN
        HERE IS THE SUBNET YOU NEED hsn_base_subnet 10.253.0.0/16
        10.253.0.1 x3000c0s09b0n0h1
        10.253.0.2 x3000c0s09b0n0h0
        10.253.0.3 x3000c0s07b0n0h1
   ```

```
    10.253.0.4 x3000c0s07b0n0h0
...
...
```

5. Delete Existing DNS Records.

```
slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot # fmn_shasta_dns -i /opt/cray/fabric_template.json  -f -x
```

6. Copy New/Fixed P2P File to `/opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv`,Use Slingshot topology tool to create fabric-template from p2p file.

```
slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot # slingshot-topology-tool
Using Fabric Manager URL http://localhost:8000
STT diags log directory -  /opt/slingshot/stt_diags_logs
STT diags log directory -  /opt/slingshot/stt_diags_logs/default
Loading  point2point file /opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv to default topology
Loading fabric template file /opt/cray/fabric_template.json to default topology
Welcome to the Slingshot Topology Tool v1.3.1-8.
    General Usage is <command> <arguments>
    Type help or ? to list commands.

(STT) new topology setup5
STT diags log directory -  /opt/slingshot/stt_diags_logs/setup5
(STT) load p2p /opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv
Working with 'default' topology.
Loading p2p from /opt/cray/etc/sct/Shasta_system_hsn_pt_pt.csv
(STT) add nics 10.253.0.0/16
Working with 'default' topology.
(STT) save topology fabric_template_setup5.json
Working with 'default' topology.
Wrote 'default' topology to fabric_template_setup5.json
(STT)
(STT) exit
Exiting STT
slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot #
slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot # mv fabric_template_setup5.json /opt/cray/fabric_template.json
```

7. Configure DNS and Update all NIC IP's.

```
slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot # fmn_shasta_dns -i /opt/cray/fabric_template.json  -f && fmn_shasta_dns -p
...
...
slingshot-fabric-manager-7dccbfb48c-kqkx7:/opt/slingshot # fmn_replace_fabric_nic_ips
Patching to all fabric agents is completed.

Cray IP/ TLV set on 5 / 5 switches
```

8. Run DVS Script to check if you have any DNS issues. See Check for DNS Errors.

1.18 HSN link failovers to ClusterStor server

1.18.1 HSN links to ClusterStor server NICs can failover if a HSN link is down

ClusterStor storage nodes have a utility that monitors the HSN links, and if a link is down for more than 30 seconds, a failover will be initiated. This is not a fatal condition, but it can be disruptive. As such, when performing maintenance on the network that may result in links to the file system being down for more than 30 seconds, the filesystem needs to be taken off line to prevent failovers.

1.18.2 ClusterStor failovers are possible when Slingshot switches get a firmware upgrade

ClusterStor fail-overs are possible when Slingshot switches get a firmware upgrade, which requires a switch reboot. Unmounting and stopping file system does NOT prevent failover on ClusterStor nodes. Rather, to prevent failover, the link monitor needs to be stopped.

This requires running the following from the ClusterStor management node (node `n00`, or `n01` if `n00` is down) before the firmware update:

```
$ pdsh -g lustre stop_lcnmon
```

When switches are back online, execute the following from the ClusterStor management node:

```
$ pdsh -g lustre start_lcnmon
```

1.19 LLDPAD buffer overflow

1.19.1 Symptom

If we see any HSN NIC TLV issues on Rosetta or compute node, then verify the lldpad daemon on it does not display the following buffer full message. For additional information about HSN NIC TLV diagnositcs, please refer to the Admin Fabric Command Reference.

```
root@ros-0040a682f75d:~# systemctl status lldpad
  lldpad.service - Link Layer Discovery Protocol Agent Daemon.
Loaded: loaded (/lib/systemd/system/lldpad.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2016-11-03 17:16:44 UTC; 4 years 2 months ago
Main PID: 4053 (lldpad)
Tasks: 1 (limit: 4915)
CGroup: /system.slice/lldpad.service
 4053 /usr/sbin/lldpad -t

Nov 03 17:16:44 ros-0040a682f75d systemd[1]: Started Link Layer Discovery Protocol Agent Daemon..
Jan 25 19:53:38 ros-0040a682f75d lldpad[4053]: recvfrom(Event interface): No buffer space available
Jan 25 19:53:38 ros-0040a682f75d lldpad[4053]: recvfrom(Event interface): No buffer space available
Jan 26 02:39:04 ros-0040a682f75d lldpad[4053]: recvfrom(Event interface): No buffer space available
root@ros-0040a682f75d:~#
```

1.19.2 Solution

Run "systemctl restart lldpad" to restart the lldpad daemon.

1.20 Cray TLV is not broadcasting on node reboot

In a Cray EX environment, if some nodes boot but `Cray TLV` is not being broadcast at time of node boot, perform the following steps:

1.20.1 Step 1: Verify Cray TLV is being broadcast

Verify that `Cray TLV` is being broadcast from the switch:

```
lldptool -i hsn0 -t -n
# OR
lldptool -i hsn1 -t -n
```

1.20.2 Step 2: Restart shasta-network-lldp service

Assuming that `Cray TLV` is being broadcast, restart the shasta-network-lldp service:

```
systemctl restart shasta-network-lldp
```

When the shasta-network-lldp service is restarted the `/etc/sysconfig/network/ifcfg-hsnX` files for the hsn0 and hsn1 interfaces should be created.

1.20.3 Step 3: ifdown/ifup HSN interface

```
# Bring down HSN interface
ifdown hsn0
# OR
ifdown hsn1

# Bring up HSN interface
ifup hsn0
# OR
ifup hsn1
```

1.21 Troubleshooting Issues with River Cable Validator

1.21.1 Symptom

IP Configuration on server does not match the `fabric_template.json`

1.21.2 Description

If you are upgrading your Slingshot software from <= 1.7.2 to >= 1.7.3, you should run this tool to ensure the `fabric_template.json` is correct.

The reason for this check is to ensure that Y-Cable connections are correctly configured in the `fabric_template.json`

1.21.3 Diagnosis

The River Cable Validator ensures that P1 / P0 mapping with in `fabric_template.json` file matches physical connection. This script is used check against a Y-Cable being flipped during `fabric_template.json` file generation.

The script does an SSH connection and calls the lldptool and compares the results with whats inside the `fabric_template.json` that was provided.

Parameters:

```
REQUIRED --username :  Username of Compute Server
REQUIRED --password :  Password of Compute Server
REQUIRED --ft       :  Fabric Template file that is currently being used.  Usually this is /opt/cray/fabric_template.json
OPTIONAL --public_key : This can be used instead of --password.  This must point to a file
OPTIONAL --verbose  :  Verbose Output of failures
```

Usage:

```
river_cable_validator --username <USERNAME> --password <PASSWORD> --ft <Fabric_Template_FILE>
```

1.21.4 Solution

If the diagnosis script shows a bad configuration. You will need to re-generate a new fabric_template.json to correct the issue.

2 Certificates and Authentication

2.1 Slingshot Switch Certificate Provision Failure

Use these diagnostic techniques to identify and address certificate provisioning failure on the switch. The switch certificate is provisioned through one of the following commands:

- `fmn-create-certificate`
- `fmn-create-certificate -n <Rosetta switch xname>`

2.1.1 `fmn-create-certificate` returns error status code 400

2.1.1.1 Symptom

The `fmn-create-certificate` command returns status code 400.

```
fmn-create-certificate
```

```
Working on switch: x1004c0r0b0
{"error":
  {"@Message.ExtendedInfo":
    [
      {"@odata.type":"#Message.v1_0_5.Message",
       "Message":"Requested certificate replacement failed",
       "MessageId":"CrayHMS.1.0.BMC002",
       "Resolution":"Verify input parameters and retry request",
       "Severity":"Critical"}
   ],
  "code":"CrayHMS.1.0.BMC002","message":"Requested certificate replacement failed"
  }
 }
STATUS : 400
```

2.1.1.2 Description

The Slingshot switch certificate provisioning failure, status code 400, can occur in following cases:

- Slingshot switch firmware version is older than 1.4 release
- Slingshot Certificate Manager could not generate the switch private and public keys

2.1.1.3 Diagnosis

Run command again with one sample switch with verbose mode on.

```
fmn-create-certificate -n <Rosetta switch xname> --verbose
```

2.1.1.3.1 Firmware version

With verbose mode enabled and test result is same, then the issue is with pre-1.4 firmware installed in the Slingshot switch. Unless `fmn-create-certificate` is older release, `fmn-create-certificate` command automatically handles new certificate provisioning.

2.1.1.3.2 Certificate Manager health

With verbose mode enabled and test result show additional information such as:

```
- Unsupported protocol

- Failed to initialize

- URL malformed

- Could not resolve host

- Fail to connect host

- HTTP page not retrieved

- HTTP range error

- HTTP post error

- SSL handshake error

- Internal error

- The peer SSL certificate fingerprint mismatch

- Server did not respond

- Peer certificate cannot be authenticated with known CA

- Invalid SSL certificate
```

The issue is `fmn-create-certificate` command could not communicate with Slingshot Certificate Manager or Slingshot Certificate Manager may be in an unstable state. Check the Slingshot Certificate Manager health with the following command:

```
# checking the health of Slingshot Certificate Manager in HTTP mode
systemctl status slingshot-certmgr

# checking health of Slingshot Certificate Manager in HTTPS mode
systemctl status slingshot-certmgr-secure
```

The Slingshot Certificate Manager can be restarted to resolve the issue with following command:

```
# checking the health of Slingshot Certificate Manager in HTTP mode
systemctl restart slingshot-certmgr

# checking health of Slingshot Certificate Manager in HTTPS mode
systemctl restart slingshot-certmgr-secure
```

2.1.1.3.3 Certificate Manager keystore corruption

The Certificate Manager keystore corruption could prevent Certificate Manager from generating the switch device certificate. This happens when Fabric Manager is recovering from a file system issue such as file system full or disk failure.

See Certificates and Authentication for detail.

2.1.1.3.4 Collect more diagnostic data

If the above solutions do no resolve the issue, please collect following information and submit them to Slingshot team.

- Software version
- Slingshot Certificate Manager Data Store

```
systemctl stop slingshot-certmgr # Certificate Manager is in HTTPS mode, systemctl stop slingshot-certmgr-secure
cd /opt/slingshot/data/slingshot
tar cvfz slingshot-certmgr-sandbox.tgz certificate-manager
systemctl start slingshot-certmgr
```

2.1.2 Slingshot switch credential not configured properly

2.1.2.1 Symptom

```
fmn-create-certificate
Rosetta credential not available. Please, create <user:password> in /opt/slingshot/config/rosetta-redfish-token/token
```

2.1.2.2 Description

To provision Slingshot switch certificate, the administrator must configure the switch API credential as part of the installation procedure.

2.1.2.3 Diagnosis

Check the credential artifact on the file system and in the Fabric Manager Rosetta Authentication Token service:

```
ls -l /opt/slingshot/config/rosetta-auth-token/token
# Fabric Manager is in HTTP mode
curl http://<Fabric Manager Host>:8000/fabric/rosetta-auth-token | jq -r
# Fabric Manager is in HTTPS mode
curl https://<Fabric Manager Host>:443/fabric/rosetta-auth-token | jq -r
```

2.1.2.4 Initialize Slingshot switch authentication token using following command:

```
fmn-update-password --commit<enter>
Enter username : root
Enter password : ********
Confirm password: ********
```

2.1.3 Expected output of certificate provisioning

Provision TLS certificates and switch SSH authorized keys by running the following command on default switch group. Upon success we see the following:

```
fmn-create-certificate

Provisioning SSH public key is successful on x3000c0r39b0 with http status 204
Provisioning certificate is successful on x3000c0r39b0 with http status 204

Provisioning SSH public key is successful on x3000c0r42b0 with http status 204
Provisioning certificate is successful on x3000c0r42b0 with http status 204

Provisioning SSH public key is successful on x3000c0r41b0 with http status 204
Provisioning certificate is successful on x3000c0r41b0 with http status 204

Provisioning SSH public key is successful on x3000c0r40b0 with http status 204
Provisioning certificate is successful on x3000c0r40b0 with http status 204
```

Provision TLS certificates and Rosetta SSH authorized keys by running the following command on specific switch group. Upon success, we see the following:

```
fmn-create-certificate --group "moa-1"

Provisioning SSH public key is successful on x3000c0r39b0 with http status 204
Provisioning certificate is successful on x3000c0r39b0 with http status 204

Provisioning SSH public key is successful on x3000c0r41b0 with http status 204
Provisioning certificate is successful on x3000c0r41b0 with http status 204
```

Provision TLS certificates and Rosetta SSH authorized keys by running the following command on a single switch. Upon success, we see the following:

```
fmn-create-certificate -n "x3000c0r40b0"

Working on switch: x3000c0r40b0
Provisioning SSH public key is successful on x3000c0r40b0 with http status 204
Provisioning certificate is successful on x3000c0r40b0 with http status 204
```

2.1.4 Expected output responses in verbose mode

Provision TLS certificates and Rosetta SSH authorized keys by running the following command on default switch group in verbose mode. Upon success, we see the following:

```
fmn-create-certificate --verbose

Working on switch: x0c0r1b0

Provision SSH public key on x0c0r1b0
Working on switch: x0c0r0b0

Provision certificate on x0c0r1b0
Provisioning SSH public key is successful on x0c0r1b0 with HTTP status 204
Provisioning certificate is successful on x0c0r1b0 with HTTP status 204

Provision SSH public key on x0c0r0b0
Provision certificate on x0c0r0b0
Provisioning SSH public key is successful on x0c0r0b0 with HTTP status 204
Provisioning certificate is successful on x0c0r0b0 with HTTP status 204
```

Provisioning TLS certificates and Rosetta SSH authorized keys by running the following command on specific switch group in verbose mode. Upon success, we see the following:

```
fmn-create-certificate --verbose --group "template-switches"

Working on switch: x0c0r1b0

Provision SSH public key on x0c0r1b0
Working on switch: x0c0r0b0

Provision certificate on x0c0r1b0
Provisioning SSH public key is successful on x0c0r1b0 with HTTP status 204
Provisioning certificate is successful on x0c0r1b0 with HTTP status 204

Provision SSH public key on x0c0r0b0
Provision certificate on x0c0r0b0
Provisioning SSH public key is successful on x0c0r0b0 with HTTP status 204
Provisioning certificate is successful on x0c0r0b0 with HTTP status 204
```

Provision TLS certificates and Rosetta SSH authorized keys by running the following command on a single switch in verbose mode. Upon success, we see the following:

```
fmn-create-certificate -n "x0c0r0b0" --verbose

Working on switch: x0c0r0b0
Provision SSH public key on x0c0r0b0
Provision certificate on x0c0r0b0
Provisioning SSH public key is successful on x0c0r0b0 with HTTP status 204
Provisioning certificate is successful on x0c0r0b0 with HTTP status 204
```

2.2 Slingshot Certificate Manager

Use these diagnostic techniques to identify and address certificate manager operational failure.

2.2.1 Certificate Manager Keystore Corruption

2.2.1.1 Symptom

The `fmn-create-certificate` command returns error status code 400 since Certificate Manager could not generate the keypairs for the given switches.

```
Working on switch: x1004c0r0b0
{"error":
  {"@Message.ExtendedInfo":
    [
      {"@odata.type":"#Message.v1_0_5.Message",
       "Message":"Requested certificate replacement failed",
       "MessageId":"CrayHMS.1.0.BMC002",
       "Resolution":"Verify input parameters and retry request",
       "Severity":"Critical"}
   ],
  "code":"CrayHMS.1.0.BMC002","message":"Requested certificate replacement failed"
  }
 }
STATUS : 400
```

2.2.1.2 Description

The Slingshot Certificate Manager keystore corruption can occur in following cases:

- Fabric Management Node file system was full and `fmn-create-certificate` command was run during that time.
- Manually tampered by mis-actor

2.2.1.3 Diagnosis

The `/opt/slingshot/data/slingshot/certificate-manager/CertMgrServiceHost.<port>.0.log` contains following error:

```
[15][I][2021-05-25T00:14:51.195Z][40][PlatformUtils][createKeyPair][exit command [/usr/bin/easyrsa, --batch, --pki-dir=data/CA/, build-serverClient-full,
[16][W][2021-05-25T00:14:51.197Z][40][9000][processPendingServiceAvailableOperations][Service
/certmgr/switch-certificates/x0c0r0b0 failed start:
java.lang.IllegalAccessException: fail while generating the certificate]
```

2.2.1.4 Manual restoration of Keystore

Delete all Rosetta switch device certificate related files under following directories:

- `/opt/slingshot/data/CA/private`
- `/opt/slingshot/data/CA/issued`
- `/opt/slingshot/data/CA/reqs`

```
find /opt/slingshot/data/CA -name "x*" -exec rm -f {} \; -print
```

2.2.2 Fabric Manager Java Truststore Corruption

2.2.2.1 Symptom

The `fmn_status` command shows Fully Synchronized Switches: 0 / n whereby n is the number of switches.

```
Edge: 0 / 128
Fabric: 0 / 48
Ports Reported: 0 / 176
Fully Synchronized Switches: 0 / 4
```

2.2.2.2 Description

The Fabric Manager Java Truststore corruption can occur in following cases:

- Cray EX Fabric Manager PVC file system is full and Fabric Manager POD restarted
- Manually tampered by mis-actor

2.2.2.3 Diagnosis

The size of file `/opt/slingshot/data/slingshot-keystore` is 0.

The `keytool -list /opt/slingshot/data/slingshot-keystore` returns no data

```
keytool -list -keystore /opt/slingshot/data/slingshot-keystore -storepass <passcode>`

Keystore type: jks
Keystore provider: SUN
Your keystore contains 0 entries
```

2.2.2.4 Manual restoration of Java Truststore

- Ensure to free up the space on the file system

2.2.2.4.1 Cray EX

```
rm -f /opt/slingshot/data/slingshot-keystore
keytool -import -file /opt/slingshot/data/CA/ca.crt -alias "SLINGSNHOT CA" \
        -keystore /opt/slingshot/data/slingshot-keystore \
        -storepass <passcode> -trustcacerts -noprompt
keytool -import -file /var/run/configmap/ca-public-key/certificate_authority.crt \
        -alias "SHASTA PLATFORM CA"  -keystore /opt/slingshot/data/slingshot-keystore \
        -storepass <passcode> -trustcacerts -noprompt
```

Exit from Fabric Manager POD and restart

```
kubectl rollout restart deployment slingshot-fabric-manager -n services
```

2.2.2.4.2 FMN

```
rm -f /opt/slingshot/data/slingshot-keystore
keytool -import -file /opt/slingsot/CA/ca.crt -alias "SLINGSNHOT CA" \
        -keystore /opt/slingshot/data/slingshot-keystore \
        -storepass <passcode> -trustcacerts -noprompt
systemctl restart fabric-manager
```

2.3 Slingshot Fabric Manager Authentication Token Service

Use following diagnostic techniques to identify and address authentication token services issues.

- `fmctl get health-engines/template-policy`
- `journalctl -t fabric-manager.sh -f`
- Look up log file under `/opt/slingshot/data/slingshot/fabric/8000/FabricHost.8000.0.log`

2.3.1 Cannot retrieve Cray EX Keycloak OpenID Token

2.3.1.1 Symptom

Telemetry data from Rosetta switch is not delivered to Fabric Manager.

Log displays following error messages:

```
[11][S][2021-05-19T07:56:13.818Z][80][8000/fabric/shasta-auth-token][reportErrorAndRecover][Failed
to retrieve OpenID token from IAM server]


[73][W][2021-05-19T07:56:48.283Z][55][8000/fabric/agents/x0c0r0b0][lambda$syncSwitchConfiguration$22][Failed
to retrieve OpenID token from IAM server]
```

Health Event displays following critical event:

```
fmctl get health-engines/topology-template
```

2.3.1.2 Description

In the Cray EX environment, Rosetta switch telemetry data is delivered to Fabric Manager through API GW using Slingshot Fabric Manager NB APIs. The Cray EX API GW will authenticate every telemetry messages with OpenID Token which was issued to Rosetta switch through Fabric Manager. When above symptoms are observed, the Fabric Manager could not acquire the OpenID token from Cray EX keycloak server.

2.3.1.3 Diagnosis

Check the `shasta-auth-token` service `accessToken` to see it contains the token with invalid signature.

```
$ fmctl get shasta-auth-token
+-----------------+-------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|       KEY       |                                                                VALUE                                                                                              |
+-----------------+-------------------------------------------------------------------------------------------------------------------------------------------------------------------+
| accessToken     | eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIwIiwibmFtZSI6IiIsImlhdCI6MH0.0000-0000000000000000000000-000000000000000 |
| clientSecret    | afa30cd3-8f72-4c5b-adce-2688532bfeb7                                                                                                                              |
| documentSelfLink | /fabric/shasta-auth-token                                                                                                                                        |
| expiresIn       |                                                                                                                                                                  10 |
| expiryTimer     |                                                                                                                                                                   0 |
+-----------------+-------------------------------------------------------------------------------------------------------------------------------------------------------------------+
```

The root cause is likely because Cray EX Keycloak server is unreachable due to:

- management network issue
- installation issue
- DNS issue

2.3.1.4 Work-Around

There is no work-around if the Cray EX Keycloak server is not accessible. However, if Fabric Manager is showing the same symptom and Keycloak server is accessible then manual PATCH to the `shasta-auth-token` service is possible.

2.3.1.4.1 Obtain the OpenID Token

```
export client_id=$(cat /var/run/secrets/kubernetes.io/serviceaccount/admin-client-auth/client-id)
export client_secret=$(cat /var/run/secrets/kubernetes.io/serviceaccount/admin-client-auth/client-secret)
curl -ks -d grant_type=client_credentials -d client_id=$client_id \
-d client_secret=$client_secret \
https://api-gw-service-nmn.local/keycloak/realms/shasta/protocol/openid-connect/token | jq -r .access_token
```

2.3.1.4.2 PATCH the shasta-auth-token service to renew the Rosetta Switch OpenID Token

```
fmctl update shasta-auth-token accessToken=<OpenID Token>
```

2.4 Slingshot Switch Passwordless SSH Login from Fabric Management Node (FMN) Failure

2.4.1 Symptom

```
slingshot-fmn> ssh root@<Rosetta switch xname>
root@<Rosetta switch xname>'s password:
```

2.4.2 Description

The Slingshot switch password-less SSH login can be enabled by configuring SSH public key authentication. If it is not configured, you are prompted to enter a password for SSH login.

2.4.3 Solution for Slingshot version 0.8.0A and earlier

Generate payload in `add-authorized-key.json` file

```
{
  "Oem": {
    "SSHAdmin": {
      "AuthorizedKeys":"<copy-and-paste ~/.ssh/id_rsa.pub content>"
    }
  }
}
```

Run Redfish REST API

```
curl -k -u root:<Password> -X PATCH -H "Content-Type: application/json" \
-d @add-authorized-key.json \
https://<Rosetta switch xname>/redfish/v1/Managers/BMC/NetworkProtocol
```

2.4.4 Solution for later Slingshot versions, post 0.8.0A

2.4.4.1 New switch, never provisioned with certificate

For individual switch:

```
fmn-create-certificate -n <Rosetta switch xname>
```

For batch processing:

```
fmn-create-certificate
```

2.4.4.2 Switch has been provisioned

Re-provisioning of the certificate does not harm the switches. However, it is a CPU intensive task which takes a long time to complete. You need `ssh-only` authorized key installation.

For individual switch:

```
fmn-create-certificate -n <Rosetta switch xname> --ssh-only
```

For batch processing:

```
fmn-create-certificate --ssh-only
```

2.5 Rosetta Redfish credential mismatch

2.5.1 Symptom

```
fmn_status
...(omitted)...
Edge: 0 / 8
Fabric: 0 / 24
Ports Reported: 0 / 32
Fully Synchronized Switches: 0 / 4
```

2.5.2 Description

The Rosetta Redifsh credential mismatch results in failure of switch and fabric configuration. The symptom often coming in as unsynchronized switch in `fmn_status` command. The root cause of the issues could vary:

- Undiscovered switch during the Redfish credential update
- Connectivity failure between switch and Fabric Manager during the Redifsh credential update
- Newly added switch

2.5.3 Diagnosis

```
fmn_status --detail
...(omitted)...
Unauthorized switch:

Swtich: x0c0r0b0
```

2.5.4 Solution

2.5.4.1 Rosetta switch with same administrator credential stored in Fabric Manager

Rerun "fmn-update-password"

```
fmctl get rosetta-auth-token
+------------------+-----------------------------+
|       KEY        |           VALUE             |
+------------------+-----------------------------+
| documentSelfLink | /fabric/rosetta-auth-token |
| password         | <fmn current password>     |
| user             | root                        |
+------------------+-----------------------------+

fmn-update-password root:<fmn current password> --commit -n x0c0r0b0
```

2.5.4.2 New Rosetta switch from Factory and Default Credential Already Changed

Rerun "fmn-update-password" with factory default credential

Make sure that new switch is alreay part of the switch inventory service (/fabric/switches)

```
fmctl get rosetta-auth-token
+------------------+-----------------------------+
|       KEY        |           VALUE             |
+------------------+-----------------------------+
| documentSelfLink | /fabric/rosetta-auth-token |
| password         | <fmn current password>     |
| user             | root                        |
+------------------+-----------------------------+

fmn-update-password root:<fmn current password> --commit -n x0c0r0b0 --admin-credential root:initial0 --redfish-only
fmn-create-certificate -n x0c0r0b0
fmn-update-password root:<fmn current password> --commit -n x0c0r0b0 --ssh-only
```

2.5.4.3 Rosetta switch with different administrator credential stored in Fabric Manager

Rerun "fmn-update-password" with current administrator credential known to swtich

```
fmctl get rosetta-auth-token
+------------------+-----------------------------+
|       KEY        |           VALUE             |
+------------------+-----------------------------+
| documentSelfLink | /fabric/rosetta-auth-token |
| password         | <fmn current password>     |
| user             | root                        |
+------------------+-----------------------------+

fmn-update-password root:<fmn current password> --commit -n x0c0r0b0 --admin-credential root:<switch current password>
```

3 Compute Nodes

3.1 Compute Node Troubleshooting Guide

3.1.1 Confirm connectivity

The first step is to confirm the link is up. Do this by executing the 'ip' command for each high speed interface:

```
cn1:~ # ip addr show dev hsn0
2: hsn0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP group default qlen 1000
    link/ether 11:22:33:44:55:66 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/24 scope global hsn0
    ...
```

Observe that the state is **UP**. Repeat for additional interfaces:

```
cn1:~ # ip addr show dev hsn1
3: hsn1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP group default qlen 1000
    link/ether aa:bb:cc:dd:ee:ff brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.101/24 scope global hsn1
```

Next, confirm the route table information:

```
cn1:~ # ip route show dev hsn0
10.0.0.0/24 proto kernel scope link src 10.0.0.1
cn1:~ # ip route show dev hsn1
10.0.0.0/24 proto kernel scope link src 10.0.0.101
cn1:~ #
```

While the interfaces are in the **UP** state with route table entries, ping the neighboring compute nodes. Do this for each high speed interface using the **-I** flag:

```
cn1:~ # ping -c1 -I hsn0 cn2
PING cn2 (10.0.0.2) from 10.0.0.1 hsn0: 56(84) bytes of data.
64 bytes from cn2 (10.0.0.2): icmp_seq=1 ttl=64 time=0.123 ms

--- cn2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.123/0.123/0.123/0.000 ms

cn1:~ # ping -c1 -I hsn1 cn2
PING cn2 (10.0.0.2) from 10.0.0.101 hsn1: 56(84) bytes of data.
64 bytes from cn2 (10.0.0.2): icmp_seq=1 ttl=64 time=0.107 ms

--- cn2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.107/0.107/0.107/0.000 ms
```

3.2 Troubleshooting Bond0 Interface on FMN

On some systems, a system may require that they bond their 1G interfaces for redundancy and/or better performance.

The following is an example of what a bonding configuration should typically look like on the FMN.

```
[root@slingshot-fmn ~]# ip a s bond0
7: bond0: <NO-CARRIER,BROADCAST,MULTICAST,MASTER,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 46:90:89:f2:e5:ba brd ff:ff:ff:ff:ff:ff
    inet 10.100.0.3/17 brd 10.100.127.255 scope global noprefixroute bond0
       valid_lft forever preferred_lft forever

[root@slingshot-fmn ~]# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)
Bonding Mode: IEEE 802.3ad Dynamic link aggregation
Transmit Hash Policy: layer2+3 (2)
MII Status: down
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0
Peer Notification Delay (ms): 0
802.3ad info
LACP rate: slow
Min links: 0
Aggregator selection policy (ad_select): stable
System priority: 65535
System MAC address: 46:90:89:f2:e5:ba
bond bond0 has no active aggregator

[root@slingshot-fmn ~]# cat /etc/sysconfig/network-scripts/ifcfg-bond0
DEVICE=bond0
NAME=bond0
TYPE=Bond
BONDING_MASTER=yes
ONBOOT=yes
```

```
BOOTPROTO=none
IPADDR=10.100.0.3
PREFIX=17
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
GATEWAY=10.100.127.254
DEFROUTE=yes
BONDING_SLAVE_0='enp66s0f0'
BONDING_SLAVE_1='enp66s0f1'
BONDING_OPTS='mode=802.3ad xmit_hash_policy=layer2+3 miimon=100'
```

```
[root@slingshot-fmn ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether b4:2e:99:ab:12:80 brd ff:ff:ff:ff:ff:ff
3: eno2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether b4:2e:99:ab:12:81 brd ff:ff:ff:ff:ff:ff
4: enp65s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 0c:42:a1:87:d3:14 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::bcda:4363:a1eb:92d4/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
5: enp65s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 0c:42:a1:87:d3:15 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::1501:4fdc:a02a:d288/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
6: usb0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
    link/ether 5a:99:08:28:b5:2b brd ff:ff:ff:ff:ff:ff
    inet6 fe80::b6e3:b60c:e27a:84ef/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
7: bond0: <NO-CARRIER,BROADCAST,MULTICAST,MASTER,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 46:90:89:f2:e5:ba brd ff:ff:ff:ff:ff:ff
    inet 10.100.0.3/17 brd 10.100.127.255 scope global noprefixroute bond0
       valid_lft forever preferred_lft forever
```

```
[root@slingshot-fmn network-scripts]# dmesg | grep -i bond
[   13.627090] Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)
[   13.665263] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[   13.671887] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[   14.745304] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[   14.796987] bond0: Invalid ad_actor_system MAC address.
[   14.802581] bond0: option ad_actor_system: invalid value (00:00:00:00:00:00)
[   14.824726] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[13497.218483] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[13497.226652] bond0 (unregistering): Released all slaves
[14154.221230] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[14154.227207] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[14154.234544] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[14154.242198] bond0: Invalid ad_actor_system MAC address.
[14154.247443] bond0: option ad_actor_system: invalid value (00:00:00:00:00:00)
[14154.255442] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
```

```
[root@slingshot-fmn network-scripts]# dmesg | grep enp
[   10.641568] mlx5_core 0000:41:00.0 enp65s0f0: renamed from eth0
[   10.657560] mlx5_core 0000:41:00.1 enp65s0f1: renamed from eth1
[   13.697887] IPv6: ADDRCONF(NETDEV_UP): enp65s0f0: link is not ready
[   14.037558] mlx5_core 0000:41:00.0 enp65s0f0: Link down
[   14.061874] IPv6: ADDRCONF(NETDEV_UP): enp65s0f0: link is not ready
[   14.070890] mlx5_core 0000:41:00.0 enp65s0f0: Link up
[   14.087224] IPv6: ADDRCONF(NETDEV_CHANGE): enp65s0f0: link becomes ready
[   14.123186] IPv6: ADDRCONF(NETDEV_UP): enp65s0f1: link is not ready
[   14.460836] mlx5_core 0000:41:00.1 enp65s0f1: Link down
[   14.484976] IPv6: ADDRCONF(NETDEV_UP): enp65s0f1: link is not ready
[   14.494919] mlx5_core 0000:41:00.1 enp65s0f1: Link up
[   14.508031] IPv6: ADDRCONF(NETDEV_CHANGE): enp65s0f1: link becomes ready
```

enp65s0f0: renamed from eth0

enp65s0f1: renamed from eth1

ifcfg files should be renamed to match!

```
[root@slingshot-fmn network-scripts]# ls -l
total 16
-rw-r--r-- 1 root root 370 Feb 27 03:54 ifcfg-bond0
-rw-r--r-- 1 root root  52 Feb 26 22:45 ifcfg-eno1
-rw-r--r-- 1 root root  97 Feb 27 03:50 ifcfg-enp65s0f0
-rw-r--r-- 1 root root  97 Feb 27 03:51 ifcfg-enp65s0f1
```

```
[root@slingshot-fmn network-scripts]# cat ifcfg-enp65s0f0
DEVICE=enp65s0f0
NAME=bond0-slave
BOOTPROTO=none
ONBOOT=yes
TYPE=Ethernet
MASTER=bond0
SLAVE=yes

[root@slingshot-fmn network-scripts]# cat ifcfg-enp65s0f1
and

[root@slingshot-fmn network-scripts]# cat ifcfg-bond0
DEVICE=bond0
NAME=bond0
TYPE=Bond
BONDING_MASTER=yes
ONBOOT=yes
BOOTPROTO=none
IPADDR=10.100.0.3
PREFIX=17
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
GATEWAY=10.100.127.254
DEFROUTE=yes
BONDING_SLAVE_0='enp65s0f0'
BONDING_SLAVE_1='enp65s0f1'
BONDING_OPTS='mode=802.3ad xmit_hash_policy=layer2+3 miimon=1000'

[root@slingshot-fmn network-scripts]# dmesg | grep -i bond
[   13.627090] Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)
[   13.665263] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[   13.671887] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[   14.745304] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[   14.796987] bond0: Invalid ad_actor_system MAC address.
[   14.802581] bond0: option ad_actor_system: invalid value (00:00:00:00:00:00)
[   14.824726] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[13497.218483] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[13497.226652] bond0 (unregistering): Released all slaves
[14154.221230] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[14154.227207] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[14154.234544] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[14154.242198] bond0: Invalid ad_actor_system MAC address.
[14154.247443] bond0: option ad_actor_system: invalid value (00:00:00:00:00:00)
[14154.255442] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[15166.840899] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[15166.849548] bond0 (unregistering): Released all slaves
[15269.983586] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[15269.989917] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[15269.996689] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[15270.004458] bond0: Invalid ad_actor_system MAC address.
[15270.009700] bond0: option ad_actor_system: invalid value (00:00:00:00:00:00)
[15270.017761] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[15358.895571] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[15358.920418] bond0 (unregistering): Released all slaves
[15384.723362] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[15384.729624] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[15384.753285] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[15384.777877] bond0: Invalid ad_actor_system MAC address.
[15384.783371] bond0: option ad_actor_system: invalid value (00:00:00:00:00:00)
[15384.805367] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[15810.475654] bond0: (slave enp65s0f0): Enslaving as a backup interface with an up link
[15810.491294] IPv6: ADDRCONF(NETDEV_CHANGE): bond0: link becomes ready
[15810.981741] bond0: (slave enp65s0f1): Enslaving as a backup interface with a down link
[15811.808406] bond0: (slave enp65s0f1): link status definitely up, 40000 Mbps full duplex
[15811.816827] bond0: active interface up!

[root@slingshot-fmn network-scripts]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether b4:2e:99:ab:12:80 brd ff:ff:ff:ff:ff:ff
3: eno2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether b4:2e:99:ab:12:81 brd ff:ff:ff:ff:ff:ff
4: enp65s0f0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master bond0 state UP group default qlen 1000
    link/ether 0c:42:a1:87:d3:14 brd ff:ff:ff:ff:ff:ff
5: enp65s0f1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master bond0 state UP group default qlen 1000
    link/ether 0c:42:a1:87:d3:14 brd ff:ff:ff:ff:ff:ff
6: usb0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
    link/ether 5a:99:08:28:b5:2b brd ff:ff:ff:ff:ff:ff
```

```
    inet6 fe80::b6e3:b60c:e27a:84ef/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
10: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 0c:42:a1:87:d3:14 brd ff:ff:ff:ff:ff:ff
    inet 10.100.0.3/17 brd 10.100.127.255 scope global noprefixroute bond0
       valid_lft forever preferred_lft forever
    inet6 fe80::5681:2bad:c638:8b52/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Ping from Admin Node

```
<ADMIN>:~ # ping afmn02
PING afmn02.head.acorn.wcoss2.ncep.noaa.gov (10.100.0.3) 56(84) bytes of data.
64 bytes from afmn02.head.acorn.wcoss2.ncep.noaa.gov (10.100.0.3): icmp_seq=1 ttl=64 time=0.561 ms
64 bytes from afmn02.head.acorn.wcoss2.ncep.noaa.gov (10.100.0.3): icmp_seq=2 ttl=64 time=0.295 ms
```

SSH into FMN

```
<ADMIN>:~ # ssh root@afmn02
Warning: Permanently added 'afmn02' (ECDSA) to the list of known hosts.
Warning: the ECDSA host key for 'afmn02' differs from the key for the IP address '10.100.0.3'
Offending key for IP in /root/.ssh/known_hosts:9
root@afmn02's password:
Last login: Sat Feb 27 03:06:53 2021
[root@slingshot-fmn ~]#
```

3.3 Address Resolution Protocol (ARP)

For any destination IPv4 address that is not in its cache, the Linux network stack will generate a broadcast ARP packet requesting a MAC address for the specified IP address. A switch or the IP address owner will respond with an ARP response packet. Basic, standard ARP default parameters will not scale to support large systems.

Improper sizing can result in

- Connectivity Issues between Compute Nodes
- Jobs Failing to Start and Not able to Complete

3.3.1 ARP (Address Resolution Protocol) Sizing

Sizing parameters for the ARP tables depend on the size of the fabric and number of endpoints. The following parameters should be appropriately set for compute nodes.

neigh/default/gc_thresh1 - INTEGER Minimum number of entries to keep. Garbage collector will not purge entries if there are fewer than this number. Default: 128 : Recommended 4096

neigh/default/gc_thresh2 - INTEGER Threshold when garbage collector becomes more aggressive about purging entries. Entries older than 5 seconds will be cleared when over this number. Default: 512 : Recommended 4096

neigh/default/gc_thresh3 - INTEGER Maximum number of non-PERMANENT neighbor entries allowed. Increase this when using large numbers of interfaces and when communicating with large numbers of directly-connected peers. Default: 1024 :Recommended : 8192

** gc_stale_time** Determines how often to check for stale neighbor entries. Set to 4 minutes to reduce the frequency of ARP broadcasts on the network.

3.3.2 ARP (Address Resolution Protocol) Lookup and MPI jobs

MPI jobs depend on ARP table for establishing TCP/IP connection for setup and tear down information. It is required that the ARP tables are populated with the HSN IP/MAC entries for this connection establishment to be successful. If MPI jobs fail with *Transport retry count exceeded* errors, that could be a result of connectivity issues. Hence it is required to validate
HSN connectivity and ARP table as explained below.

The following steps are recommended before initiating MPI jobs to validate ARP Table.

Ping all to all tests: Perform a ping all to all tests to ensure the connectivity between the compute nodes using HSN NICs. This can be done by one of the methods described below.

1. Slingshot Topology Tool *show hsn_traffic ping-all-to-all* command
2. *ping* command issued as *pdsh* to all compute nodes from an admin node.

This validates HSN connectivity for MPI jobs and also results in the ARP table being loaded with the cache entries.

Example

```
[root@slingshotfmn~](STT) show hsn_traffic ping-all-to-all
```

```
[root@login_node ~]pdsh -w nid00[1000-3559] 'for i in {1000..3559} ; \
                  > do ping -W 1 -c 1 nid00$i | grep packet ; done' | dshbak -c
```

Static/Permanent ARP entries: For environment, where the IP address to MAC mapping is constant, it is recommended to load the ARP entries of all compute nodes HSN IPs as static/permanent entries in all compute nodes. The will result in ARP entries to be not invalidated. This can be done as a part of the compute node boot sequence.

Example: The following example illustrates adding a permanent entry to ARP cache

```
[root@apollo-1 ~]# ip neigh add 192.168.0.220 dev ens801 lladdr \
> 02:00:00:00:08:5c nud permanent
[root@apollo-1 ~]# ip neigh show 192.168.0.220 dev ens801
192.168.0.220 lladdr 02:00:00:00:08:5c PERMANENT
```

4 Fabric Manager

4.1 Troubleshoot Fabric Property HMSCollector

4.1.1 Invalid Hostname Supplied to the HMSCollector in the Fabric Policy's FabricPropertyMap

4.1.1.1 Symptom

Below is an example on how the event will be reported on health engine. Replace "template-policy" with the name of the policy that is being used.

The health engine reports a CRITICAL event with the following information:

```
fmctl get health-engines/template-policy --fmn-endpoint http://127.0.0.1:8001
```

```
+-----------------------------------+-----------------------------------------------------------------------------------------------------------------------+
|                KEY                |                                                        VALUE                                                          |
+-----------------------------------+-----------------------------------------------------------------------------------------------------------------------+
| documentSelfLink                  | /fabric/health-engines/template-policy                                                                                |
| healthIssuesIdentified            | http://172.29.68.22:8001/metrics?$filter=((TimestampLong%20le%201622104192261)%20and%20(TimestampLong%20ge            |
|                                   | %201622104072175)%20and%20(ParentalContext%20eq%20CrayFabricHealth))                                                  |
| healthObjectsImpacted             | http://172.29.68.22:8001/metrics?$filter=((TimestampLong%20le%201622104192261)%20and%20(TimestampLong%20ge            |
|                                   | %201622104072175)%20and%20(ParentalContext%20eq%20CrayFabricHealth))%26%24select%3DLocation                           |
|                                   | %2CPhysicalSubContext                                                                                                 |
| healthRecommendations.Configuration | map[CRITICAL:[map[description:Recommend: Please retry after updating the Fabric Policy 'fabricPropertyMap'          |
|                                   |  with a valid HMS Collector URL at the service link:/fabric/topology-policies/template-policy ;                       |
|                                   | Cause: Invalid Hostname Supplied to HMSCollector in FabricPropertyMap issueTime:Thu May 27 03:29:35 CDT 2021|
|                                   | resourceLink:http://172.29.68.22:8001/fabric/topology-policies/template-policy                                        |
| healthStatus.Configuration        | CRITICAL                                                                                                              |
+-----------------------------------+-----------------------------------------------------------------------------------------------------------------------+
```

Event details:


```
{
    "Timestamp": "2021-05-26T09:54:32.996Z",
    "TimestampLong": 1622022872996,
    "Location": "http://172.29.68.22:8001/fabric/topology-policies/template-policy",
    "PhysicalContext": "Configuration.FabricPolicy",
    "ParentalContext": "CrayFabricHealth",
    "ParentalIndex": 0,
    "Index": 0,
    "SubIndex": 0,
    "PhysicalSubContext": "CRITICAL",
    "Value": "Recommend: Please retry after updating the Fabric Policy 'fabricPropertyMap'
              with a valid HMS Collector URL at the service link: /fabric/
              topology-policies/template-policy ; Cause: Invalid Hostname Supplied to
              HMSCollector in FabricPropertyMap",
    "documentVersion": 0,
    "documentEpoch": 0,
    "documentKind": "com:services:fabric:telemetry:models:Metric",
    "documentSelfLink": "/metrics/8206b8e5c969b4755c338a12998a0",
    "documentUpdateTimeMicros": 1622022872996001,
    "documentUpdateAction": "POST",
    "documentExpirationTimeMicros": 1622109272997000,
    "documentOwner": "99e70fe5-190e-4924-bc2f-9a263be116dd"
}
```

4.1.1.2 Description

The Fabric Manager has detected that the URL configured for the HMS Collector in the fabricPropertyMap of the Fabric Policy is invalid.

When a valid URL is configured for the HMSCollector, the fabricPropertyMap looks like this:

```
"fabricPropertyMap": {
    "MAX_LAG_PORTS": "256",
    "HMSCollector": "https://hpe.com:8080",
    "MTU": "9216",
    "LLDP": "true"
}
```

The impact of this problem is that, when the EventCollectionService tries to fetch the HMSCollector Address from the fabricPropertyMap to forward events, the address will be null. As a result, events may not be sent to the HMSCollector.

4.1.1.3 Diagnosis

In order to resolve this, the recommended action is to modify the HMSCollector URL in the fabricPropertyMap of the Fabric Policy with a valid URL.

4.2 Troubleshooting Disconnected Groups Errors

Use the following event to identify and address disconnected groups issues in Fabric policy

4.2.1 Disconnected groups in fabric policy

4.2.1.1 Symptom

Below is an example on how the event will be reported on health engine. Replace “template-policy” with the name of the policy that is being used.

The health engine reports a WARNING event with the following information:

```
fmctl get health-engines/template-policy
+------------------------------+-----------------------------------------------------------------------------------------------------------------------+
|             KEY              |                                                       VALUE                                                           |
+------------------------------+-----------------------------------------------------------------------------------------------------------------------+
| documentSelfLink             | /fabric/health-engines/template-policy                                                                                |
| healthIssuesIdentified       | http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%20162022028579762)%20and%20(TimestampLong%20ge             |
|                              | %201622028549768)%20and%20(ParentalContext%20eq%20CrayFabricHealth))                                                  |
| healthObjectsImpacted        | http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%20162022028579762)%20and%20(TimestampLong%20ge             |
|                              | %201622028549768)%20and%20(ParentalContext%20eq%20CrayFabricHealth))%26%24select%3DLocation%2C                        |
|                              | PhysicalSubContext                                                                                                    |
| healthRecommendations.Traffic | map[WARNING:[map[description:Recommend: Fabric Admin should verify                                                   |
|                              | the Fabric Policy for switch groups with no inter-group connections                                                   |
|                              | Cause: Multiple disconnected groups are detected on routing update.                                                   |
|                              | Traffic between the groups or overall dragonfly global routing is                                                     |
|                              | likely impacted. issueTime:Wed May 26 06:29:10 CDT 2021                                                               |
|                              | resourceLink:http://127.0.0.1:8000/fabric/topology-policies/template-policy]]]                                        |
| healthStatus.Traffic         | WARNING                                                                                                               |
+------------------------------+-----------------------------------------------------------------------------------------------------------------------+
```

Event details:

```
{
  "Timestamp": "2021-05-26T06:01:40.301Z",
  "TimestampLong": 1620818683301,
  "Location": "http://172.29.68.22:8000/fabric/topology-policies/template-policy",
  "PhysicalContext": "Traffic.DisconnectedGroups",
  "ParentalContext": "CrayFabricHealth",
  "ParentalIndex": 0,
  "Index": 0,
  "SubIndex": 0,
  "PhysicalSubContext": "WARNING",
  "Value": "Recommend: Fabric Admin should verify the Fabric Policy for switch groups with no inter-group connections ;
            Cause: Multiple disconnected groups are detected on routing update.
            Traffic between the groups or overall dragonfly global routing is likely impacted.",
  "documentVersion": 0,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:telemetry:models:Metric",
  "documentSelfLink": "/metrics/4b63dc5b46fc38755c22041dd1070",
  "documentUpdateTimeMicros": 1620818683302001,
  "documentUpdateAction": "POST",
  "documentExpirationTimeMicros": 1620905083302000,
  "documentOwner": "5c07323f-af89-4bef-b804-55948a4ef36c",
  "documentAuthPrincipalLink": "/core/authz/system-user"
}
```

4.2.1.2 Description

The Fabric Manager has detected that the fabric policy has switch groups with no inter-group connections. The impact of this problem is between the groups that are disconnected and overall dragonfly global routing. As a result some traffic may not be routable.

4.2.1.3 Diagnosis

In order to resolve this the recommended action is to verify if the switches in the groups are down. As an alternative the global links between the groups can be verified.

4.2.2 Disconnected groups in fabric policy with no proxy

4.2.2.1 Symptom

Below is an example on how the event will be reported on health engine. Replace "template-policy" with the name of the policy that is being used.

The health engine reports a CRITICAL event with the following information:

```
fmctl get health-engines/template-policy
+------------------------------+-----------------------------------------------------------------------------------------------------------------------+
|             KEY              |                                                       VALUE                                                           |
+------------------------------+-----------------------------------------------------------------------------------------------------------------------+
| documentSelfLink             | /fabric/health-engines/template-policy                                                                                |
| healthIssuesIdentified       | http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%201622028579762)%20and%20(TimestampLong%20ge               |
|                              | %201622028549768)%20and%20(ParentalContext%20eq%20CrayFabricHealth))                                                  |
| healthObjectsImpacted        | http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%201622028579762)%20and%20(TimestampLong%20ge               |
|                              | %201622028549768)%20and%20(ParentalContext%20eq%20CrayFabricHealth))%26%24select%3DLocation%2C                        |
|                              | PhysicalSubContext                                                                                                    |
| healthRecommendations.Traffic | map[CRITICAL:[map[description:Recommend: For additional details                                                      |
|                              | surrounding the routing initialization errors related to Switch                                                       |
|                              | map[description:Recommend: If this problem persists check if switches in                                              |
|                              | group 2 or group 1 is down OR check the global links between the groups ;                                             |
|                              | Cause: Switch interconnectivy issue: Groups 2 and 1 are disconnected and using                                        |
|                              | multicast proxy not fully connected issueTime:Wed May 26 06:29:10 CDT 2021                                            |
|                              | resourceLink:http://127.0.0.1:8000/fabric/topology-policies/template-policy]]]                                        |
| healthStatus.Traffic         | CRITICAL                                                                                                              |
+------------------------------+-----------------------------------------------------------------------------------------------------------------------+
```

Event details:


```
{
  "Timestamp": "2021-05-26T06:01:40.301Z",
  "TimestampLong": 1620818683301,
  "Location": "http://172.29.68.22:8000/fabric/topology-policies/template-policy",
  "PhysicalContext": "Traffic.DisconnectedMulticastGroups",
  "ParentalContext": "CrayFabricHealth",
  "ParentalIndex": 0,
  "Index": 0,
  "SubIndex": 0,
  "PhysicalSubContext": "CRITICAL",
  "Value": "Recommend: If this problem persists check if switches in group 0 or group 1 are down OR check the global links between the groups ;
           Cause: Switch interconnectivy issue: Groups 0 and 1 are disconnected and have no valid multicast proxy. Traffic between the groups
           or overall dragonfly global routing is likely impacted. Some traffic may not be routable.",
  "documentVersion": 0,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:telemetry:models:Metric",
  "documentSelfLink": "/metrics/4b63dc5b46fc38755c22041dd1070",
  "documentUpdateTimeMicros": 1620818683302001,
  "documentUpdateAction": "POST",
  "documentExpirationTimeMicros": 1620905083302000,
  "documentOwner": "5c07323f-af89-4bef-b804-55948a4ef36c",
  "documentAuthPrincipalLink": "/core/authz/system-user"
}
```


4.2.2.2 Description

The Fabric Manager has detected that there are no valid links between the specified groups. In addition to this there is no alternate switch that can act as a proxy between the two disconnected switches. The impact of this problem is between the groups that are disconnected and overall dragonfly global routing. As a result some traffic may not be routable.

4.2.2.3 Diagnosis

In order to resolve this the recommended action is to verify if the switches in the groups mentioned in the error are down. As an alternative the global links between the groups can be verified.

4.3 Troubleshooting Partially Connected Groups Issue

Use the following event to identify and address partially connected groups in Fabric policy.

4.3.1 Partially connected groups in fabric policy with proxy

4.3.1.1 Symptom

Below is an example on how the event will be reported on health engine. Replace "template-policy" with the name of the policy that is being used.

The health engine reports a WARNING event with the following information:

```
fmctl get health-engines/template-policy
+-------------------------------+-----------------------------------------------------------------------------------------------------------------+
|              KEY              |                                                    VALUE                                                        |
+-------------------------------+-----------------------------------------------------------------------------------------------------------------+
| documentSelfLink              | /fabric/health-engines/template-policy                                                                          |
| healthIssuesIdentified        | http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%201622034034535)%20and%20(TimestampLong%20ge         |
|                               | %201622034004544)%20and%20(ParentalContext%20eq%20CrayFabricHealth))                                            |
| healthObjectsImpacted         | http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%201622034034535)%20and%20(TimestampLong%20ge         |
|                               | %201622034004544)%20and%20(ParentalContext%20eq%20CrayFabricHealth))%26%24select%3DLocation%2C                  |
|                               | PhysicalSubContext                                                                                              |
| healthRecommendations.Traffic | map[CRITICAL:[map[description:Recommend: If this problem persists check                                         |
|                               | if switches in group 2 or group 1 are down OR check the global links                                            |
|                               | between the groups ; Cause: Switch interconnectivy issue: Groups 2 and                                          |
|                               | 1 are disconnected and have no valid multicast proxy. Traffic between                                           |
|                               | the groups or overall dragonfly global routing is likely impacted. Some                                         |
|                               | traffic may not be routable. issueTime:Wed May 26 08:00:05 CDT 2021                                             |
|                               | resourceLink:http://127.0.0.1:8000/fabric/topology-policies/template-policy]                                    |
| healthStatus.Traffic          | CRITICAL                                                                                                        |
+-------------------------------+-----------------------------------------------------------------------------------------------------------------+
```

Event details:


```
{
  "Timestamp": "2021-05-26T06:21:10.302Z",
  "TimestampLong": 1620818683302,
  "Location": "http://172.29.68.22:8000/fabric/topology-policies/template-policy",
  "PhysicalContext": "Traffic.DisconnectedMulticastGroups",
  "ParentalContext": "CrayFabricHealth",
  "ParentalIndex": 0,
  "Index": 0,
  "SubIndex": 0,
  "PhysicalSubContext": "WARNING",
  "Value": "Recommend: If this problem persists check if switches in group 0 or group 1 is down OR check the global links between the groups.;
           Cause: Groups 0 and 1 are disconnected and using partially connected multicast proxy",
  "documentVersion": 0,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:telemetry:models:Metric",
  "documentSelfLink": "/metrics/4b63dc5b46fc38755c22041dd1458",
  "documentUpdateTimeMicros": 1620818683303001,
  "documentUpdateAction": "POST",
  "documentExpirationTimeMicros": 1620905083303000,
  "documentOwner": "5c07323f-af89-4bef-b804-55948a4ef36c",
  "documentAuthPrincipalLink": "/core/authz/system-user"
}
```


4.3.1.2 Description

The Fabric Manager has detected that there are no valid links between the specified groups. In addition to this, the event also indicates the presence of a switch that is acting as a proxy between the groups. This indicates that the groups are partially connected.

4.3.1.3 Diagnosis

In order to resolve this the recommended action is to verify if the switches in the groups mentioned in the error are down. As an alternative the global links between the groups can be verified.

4.4 Troubleshooting Live Topology Errors

Use the following event to identify and address issues with live topology map in Fabric policy.

4.4.1 Live topology not available while checking agent status

4.4.1.1 Symptom

Below is an example on how the event will be reported on health engine. Replace “template-policy” with the name of the policy that is being used.

The health engine reports a CRITICAL event with the following information:

```
fmctl get health-engines/template-policy
```

```
+-----------------------------------+-----------------------------------------------------------------------------------------------------+
|                KEY                |                                                VALUE                                                |
+-----------------------------------+-----------------------------------------------------------------------------------------------------+
| documentSelfLink                  | /fabric/health-engines/template-policy                                                              |
| healthIssuesIdentified            | http://http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%201622743955099)%20and%20(Time    |
|                                   | stampLong%20ge%201622743834712)%20and%20(ParentalContext%20eq%20CrayFabricHealth))                  |
| healthObjectsImpacted             | http://http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%201622743955099)%20and%20(Time    |
|                                   | stampLong%20ge%201622743834712)%20and%20(ParentalContext%20eq%20CrayFabricHealth))%26%24select%3D   |
|                                   | Location%2CPhysicalSubContext                                                                       |
| healthRecommendations.Configuration | map[CRITICAL:[map[description:Recommend: Fabric Admin should verify that the Fabric               |
|                                   | Policy has an active enabled live topology map by looking at topology                               |
|                                   | service link at /fabric/topology-maps ; Cause: Fabric Policy’s live                                 |
|                                   | topology map is inaccessible issueTime:Thu Jun 03 23:40:44 IST 2021                                 |
|                                   | resourceLink:http://127.0.0.1:8000/fabric/topology-policies/template-policy]]]                      |
| healthStatus.Configuration        | CRITICAL                                                                                            |
+-----------------------------------+-----------------------------------------------------------------------------------------------------+
```

Event details:


```
{
  "Timestamp": "2021-06-03T18:08:52.931Z",
  "TimestampLong": 1622743732931,
  "Location": "http://127.0.0.1:8000/fabric/topology-policies/template-policy",
  "PhysicalContext": "Configuration.FabricPolicy",
  "ParentalContext": "CrayFabricHealth",
  "ParentalIndex": 0,
  "Index": 0,
  "SubIndex": 0,
  "PhysicalSubContext": "CRITICAL",
  "Value": "Recommend: Fabric Admin should verify that the Fabric Policy has an active enabled live topology map by looking at topology service
          link at /fabric/topology-maps ; Cause: Fabric Policy's live topology map is inaccessible",
  "documentVersion": 0,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:telemetry:models:Metric",
  "documentSelfLink": "/metrics/1311caea074ef4755c3e077c35188",
  "documentUpdateTimeMicros": 1622743732933001,
  "documentUpdateAction": "POST",
  "documentExpirationTimeMicros": 1622747332933000,
  "documentOwner": "17ad9116-cd45-46d0-83c0-ca0533f436d7",
  "documentAuthPrincipalLink": "/core/authz/system-user"
}
```


4.4.1.2 Description

The Fabric Manager has detected that there is no active live topology map. This is likely to happen when the live topology map service itself is deleted or not found while checking agent status.

4.4.1.3 Diagnosis

In order to resolve this the recommended action is to verify if Fabric policy has an active enabled live topology map. If it is not enabled restarting the fabric manager will help.

4.5 Troubleshooting Issues with Fabric Topology

Use these events and techniques to identify and address fabric topology policy related issues.

4.5.1 Topology Policy has no links

4.5.1.1 Symptom

Below is an example on how the event will be reported on health engine. Replace “template-policy” with the name of the policy that is being used.

The health engine reports a CRITICAL event with the following information:

```
fmctl get health-engines/template-policy
+-----------------------------------+-------------------------------------------------------------------------------------------------------------+
|                KEY                |                                                    VALUE                                                    |
+-----------------------------------+-------------------------------------------------------------------------------------------------------------+
| documentSelfLink                  | /fabric/health-engines/template-policy                                                                      |
| healthIssuesIdentified            | http://10.0.2.15:7771/metrics?$filter=((TimestampLong%20le%201622656605764)%20and%20(TimestampLong          |
|                                   | %20ge%201622656485402)%20and%20(ParentalContext%20eq%20CrayFabricHealth))                                   |
| healthObjectsImpacted             | http://10.0.2.15:7771/metrics?$filter=((TimestampLong%20le%201622656605764)%20and%20(TimestampLong          |
|                                   | %20ge%201622656485402)%20and%20(ParentalContext%20eq%20CrayFabricHealth))%26%24select%3DLocation            |
|                                   | %2CPhysicalSubContext                                                                                       |
| healthRecommendations.Traffic     | map[CRITICAL:[map[description:Recommend: Ensure that Fabric                                                 |
|                                   | Policy has associated Fabric Links configured in the switch                                                 |
|                                   | group ; Cause: Fabric Topology Policy does not have Fabric Links                                            |
|                                   | configured, so local and global routes cannot be calculated, and                                            |
|                                   | traffic may not be routable  issueTime:Wed Jun 02 23:24:45 IST 2021                                         |
|                                   | resourceLink:http://10.0.2.15:7771/fabric/topology-policies/template-policy]]]                              |
| healthStatus.Traffic              | CRITICAL                                                                                                    |
+-----------------------------------+-------------------------------------------------------------------------------------------------------------+
```

Event details:


```
{
      "Timestamp": "2021-05-31T15:29:11.378Z",
      "TimestampLong": 1622474951378,
      "Location": "http://10.0.2.15:7771/fabric/topology-policies/template-policy",
      "PhysicalContext": "Traffic.FabricLinks",
      "ParentalContext": "CrayFabricHealth",
      "ParentalIndex": 0,
      "Index": 0,
      "SubIndex": 0,
      "PhysicalSubContext": "CRITICAL",
      "Value": "Recommend: Ensure that Fabric Policy has associated Fabric Links configured in the switch group ;
                Cause: Fabric Topology Policy does not have Fabric Links configured, so local and global routes cannot be calculated,
                    and traffic may not be routable ",
      "documentVersion": 0,
      "documentEpoch": 0,
      "documentKind": "com:services:fabric:telemetry:models:Metric",
      "documentSelfLink": "/metrics/dcb54fab1cd70c755c3a1e3224838",
      "documentUpdateTimeMicros": 1622474951379001,
      "documentUpdateAction": "POST",
      "documentExpirationTimeMicros": 1622478551383000,
      "documentOwner": "eb2d5fb1-6ed7-4e3d-8f62-ebec23f3882e",
      "documentAuthPrincipalLink": "/core/authz/system-user"
}
```


4.5.1.2 Description

The Fabric Manager has detected that there is a fabric topology policy that does not have fabric links configured, which will lead to dropped traffic or inability to route traffic.

4.5.1.3 Diagnosis

To resolve this, ensure that the fabric policy has associated fabric links configured between switches in the switch group.

4.6 Troubleshooting Fabric Port Link Undefined Error

Use the following event to identify and address issues with missing fabric port link issue.

4.6.1 Fabric Port Link Undefined in Fabric Policy

4.6.1.1 Symptom

Below is an example on how the event will be reported on health engine. Replace “template-policy” with the name of the policy that is being used.

The health engine reports a WARNING event with the following information:

```
fmctl get health-engines/template-policy

+-----------------------------------+-------------------------------------------------------------------------------------------------------------+
|                KEY                |                                                    VALUE                                                    |
+-----------------------------------+-------------------------------------------------------------------------------------------------------------+
| documentSelfLink                  | /fabric/health-engines/template-policy                                                                      |
| healthIssuesIdentified            | http://172.29.68.22:8000/metrics?$filter=((TimestampLong%20le%201623319515153)%20and%20(Timestamp           |
|                                   | Long%20ge%201623319395074)%20and%20(ParentalContext%20eq%20CrayFabricHealth))                               |
| healthObjectsImpacted             | http://172.29.68.22:8000/metrics?$filter=((TimestampLong%20le%201623319515153)%20and%20(Timestamp           |
|                                   | Long%20ge%201623319395074)%20and%20(ParentalContext%20eq%20CrayFabricHealth))%26%24select%3D                |
|                                   | Location%2CPhysicalSubContext                                                                               |
| healthRecommendations.Configuration | map[WARNING:[map[description:Recommend: Ensure that Fabric Policy has edge port                           |
```

```
|                                    | with links defined ; Cause: Port x3000c0r3j17p1 identified as fabric port, but                    |
|                                    | no link defined, setting to UNDEFINED issueTime:Thu Jun 10 05:04:25 CDT 2021                      |
|                                    | resourceLink:http://172.29.68.22:8000/fabric/topology-policies/template-policy]]]                 |
| healthStatus.Configuration         | WARNING                                                                                           |
+------------------------------------+---------------------------------------------------------------------------------------------------+
```

Event details:


```
{
  "Timestamp": "2021-06-10T10:04:25.186Z",
  "TimestampLong": 1623319465186,
  "Location": "http://172.29.68.22:8000/fabric/topology-policies/template-policy",
  "PhysicalContext": "Configuration.FabricPolicy",
  "ParentalContext": "CrayFabricHealth",
  "ParentalIndex": 0,
  "Index": 0,
  "SubIndex": 0,
  "PhysicalSubContext": "WARNING",
  "Value": "Recommend: Ensure that Fabric Policy has edge port with links defined ;
          Cause: Port x3000c0r3j17p1 identified as fabric port, but no link defined, setting to UNDEFINED",
  "documentVersion": 0,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:telemetry:models:Metric",
  "documentSelfLink": "/metrics/8c2b83b787a62a755c466841452d0",
  "documentUpdateTimeMicros": 1623319465186001,
  "documentUpdateAction": "POST",
  "documentExpirationTimeMicros": 1623323065186000,
  "documentOwner": "cd9b5694-cbad-466a-8562-1b230c98a883",
  "documentAuthPrincipalLink": "/core/authz/system-user"
}
```


4.6.1.2 Description

The Fabric Manager has detected a fabric port without a link in fabric policy during policy change. As a result of this, Fabric Manager will set the port type as "UNDEFINED" with a warning message and continues with Routing initialization for the rest of the ports.

4.6.1.3 Diagnosis

In order to resolve this, the recommended action is to ensure that Fabric Policy has edge port with links defined properly.

4.7 Troubleshooting Issues with Fabric Topology

Use these events and techniques to identify and address fabric topology policy related issues.

4.7.1 No fabric links on the topology policy, skipping route calculation

4.7.1.1 Symptom

Below is an example on how the event will be reported on health engine. Replace "template-policy" with the name of the policy that is being used.

The health engine reports a INFO event with the following information:

```
fmctl get health-engines/template-policy
+------------------------------------+---------------------------------------------------------------------------------------------------+
|                KEY                 |                                              VALUE                                                |
+------------------------------------+---------------------------------------------------------------------------------------------------+
| documentSelfLink                   | /fabric/health-engines/template-policy                                                            |
| healthIssuesIdentified             | http://10.0.2.15:7774/metrics?$filter=((TimestampLong%20le%201623932602641)%20and%20(Timestamp    |
|                                    | Long%20ge%201623932482471)                                                                        |
|                                    | %20and%20(ParentalContext%20eq%20CrayFabricHealth))                                               |
| healthObjectsImpacted              | http://10.0.2.15:7774/metrics?$filter=((TimestampLong%20le%201623932602641)%20and%20(Timestamp    |
|                                    | Long%20ge%201623932482471)                                                                        |
|                                    | %20and%20(ParentalContext%20eq%20CrayFabricHealth))%26%24select%3DLocation%2CPhysicalSubContext   |
| healthRecommendations.Traffic      | map[CRITICAL:[map[description:Recommend: For additional details                                   |
|                                    | surrounding the routing initialization errors related to Switch                                   |
|                                    | 'x0c0r0b0' please query '/fabric/agents/x0c0r0b0' ; Cause: Failure                                |
|                                    | initializing routes on Switch: x0c0r0b0 issueTime:Thu Jun 17 17:52:42 IST 2021                    |
|                                    | resourceLink:http://10.0.2.15:7774/fabric/topology-policies/template-policy]]                     |
|                                    | INFO:[map[description:Recommend: Please confirm that the Topology                                 |
|                                    | should not have any Fabric Links configured. ; Cause: Topology                                    |
|                                    | Policy does not have Fabric Links configured, fabric route                                        |
|                                    | calculation skipped. issueTime:Thu Jun 17 17:52:53 IST 2021                                       |
|                                    | resourceLink:http://10.0.2.15:7774/fabric/topology-policies/template-policy]]]                    |
| healthStatus.Traffic               | CRITICAL                                                                                          |
+------------------------------------+---------------------------------------------------------------------------------------------------+
```

Event details:


```
"/metrics/4e71c4dee40c44755c4f54427c550": {
  "Timestamp": "2021-06-17T12:22:53.041Z",
  "TimestampLong": 1623932573041,
  "Location": "http://10.0.2.15:7774/fabric/topology-policies/template-policy",
  "PhysicalContext": "Traffic.FabricLinks",
  "ParentalContext": "CrayFabricHealth",
  "ParentalIndex": 0,
  "Index": 0,
  "SubIndex": 0,
  "PhysicalSubContext": "INFO",
  "Value": "Recommend: Please confirm that the Topology should not have any Fabric Links configured. ;
            Cause: Topology Policy does not have Fabric Links configured, fabric route calculation skipped.",
  "documentVersion": 0,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:telemetry:models:Metric",
  "documentSelfLink": "/metrics/4e71c4dee40c44755c4f54427c550",
  "documentUpdateTimeMicros": 1623932573042001,
  "documentUpdateAction": "POST",
  "documentExpirationTimeMicros": 1623936173043000,
  "documentOwner": "99c173c6-c76f-4bea-a24a-f9a50cfb7d09",
  "documentAuthPrincipalLink": "/core/authz/system-user"
```


4.7.1.2 Description

The Fabric Manager has detected that there is a fabric topology policy that does not have Fabric Links configured, which will leads to fabric route calculation skipped.

4.7.1.3 Diagnosis

Ensure that the Topology should not have any Fabric Links configured.

4.8 OData Query for fabric events and telemetry

This document captures the OData query for fabric health events and metric events.

4.8.1 Available Fields in events

S.No.	Parameter	Details
1	Timestamp	Timestamp of when telemetry was collected
2	TimestampLong	Timestamp of when telemetry collected in a format used for sorting
3	Location	Location where metric is collected, for instance port xname
4	Physical Context	The attribute name of the metric/counter , exampleString = txBytes
5	ParentalContext	Default value is : CrayFabricHealth
6	ParentalIndex	The group number where metric was collected when locality enabled
7	Index	The switch number where metric was collected when locality enabled
8	SubIndex	The port number where metric was collected when locality enabled
9	PhysicalSubContext	Severity of the error CRITICAL, WARNING ,INFO, HEALTHY
10	DeviceSpecificContext	The type of the port where metric was collected
11	Value	A string value of the attribute metric/counter, Cause + Action
12	NumberValue	A numerical value of the attribute metric/counter

4.8.2 Queries

The following section captures sample queries for various scenarios.

4.8.2.1 Query for fabric policies

To get the list of topology policies

```
$ fmctl get topology-policies
+---------------+-------------------------------------------+
|      KEY      |                  VALUE                    |
+---------------+-------------------------------------------+
| documentCount |                                         1 |
| documentLinks | /fabric/topology-policies/template-policy |
+---------------+-------------------------------------------+
```

To display the contents

```
# fmctl get "topology-policies/template-policy"
```

To get the active topology

```
# fmctl get topology-policies expand=true filter="active eq true"
```

4.8.2.2 Querying for event by Parental Context

```
# fmctl get /metrics expand=true filter="ParentalContext eq CrayFabricHealth"
```

4.8.2.3 Querying for specific Metric event or health event from the past X duration

The time stamp provided "TimestampLong" is in milliseconds. To get epoch time in milliseconds the following command can be used:

```
Current time in milliseconds since epoch
# date +%s%3N
OR
# echo $(($(date +%s%N)/1000000))
# fmctl get /metrics expand=true filter="((PhysicalContext eq Traffic.RoutingInitialization) and (PhysicalSubContext eq CRITICAL) and
(TimestampLong ge 1622554657582))"
```

4.8.2.4 Querying for specific count of event from past duration

```
# fmctl get /metrics expand=``true`  `filter=``"TimestampLong ge 1622554657582"
```

4.8.2.5 Querying for specific PhysicalContext from past duration.

```
# fmctl get /metrics expand=true filter="((PhysicalContext eq Configuration*) and (TimestampLong ge 1622554657582))"
# fmctl get /metrics expand=true \
filter="((PhysicalContext eq Traffic.FabricLinks) and (TimestampLong ge 1622554657582))&$orderby=Timestamp desc&$top=1"
# fmctl get /metrics expand=true \
filter="PhysicalContext eq Traffic.RoutingInitialization and PhysicalSubContext eq CRITICAL and TimestampLong ge 1622554657582"
```

4.8.2.6 Querying for event by severity

The available severity strings are Warning / Critical / Info/ Healthy

```
# fmctl get /metrics expand=true filter="PhysicalSubContext eq CRITICAL"
```

4.8.2.7 Querying for event by category and severity

```
# fmctl get /metrics expand=true filter="((ParentalContext eq CrayFabricHealth) and (PhysicalSubContext eq CRITICAL))"
```

4.8.2.8 Querying for event by Location / service impacted

```
# fmctl get /metrics expand=true filter="((Location eq *template-policy) and (PhysicalSubContext eq CRITICAL))"
```

4.8.2.9 Querying for event based on Value (cause and action)

```
# fmctl get /metrics expand=true filter="((Value eq *link*) and (PhysicalSubContext eq CRITICAL))"
```

4.8.2.10 All Telemetry Events for a switch

```
To get all switches
# fmctl get /fabric/switches

Apply the switch name to query
# fmctl get /metrics expand=true filter="(Location eq *x0c0r1b0)"
```

4.9 Troubleshooting No Data on Topology Policy During Lag Calculation

Use the following event to identify and address empty topology policy snapshot issue during lag calculation.

4.9.1 Fabric Port Link Undefined in Fabric Policy

4.9.1.1 Symptom

Below is an example on how the event will be reported on health engine. Replace "template-policy" with the name of the policy that is being used.

The health engine reports a WARNING event during lag update with the following information:

```
fmctl get health-engines/template-policy
```

```
+-------------------------------------+-----------------------------------------------------------------------------------------------------------+
|                 KEY                 |                                                VALUE                                                      |
+-------------------------------------+-----------------------------------------------------------------------------------------------------------+
| documentSelfLink                    | /fabric/health-engines/template-policy                                                                    |
| healthIssuesIdentified              | http://10.0.2.15:8000/metrics?$filter=((TimestampLong%20le%201623919916399)%20and%20(Timestamp            |
|                                     | Long%20ge%201623919795993)%20and%20(ParentalContext%20eq%20CrayFabricHealth))                             |
| healthObjectsImpacted               | http://10.0.2.15:8000/metrics?$filter=((TimestampLong%20le%201623919916399)%20and%20(Timestamp            |
|                                     | Long%20ge%201623919795993)%20and%20(ParentalContext%20eq%20CrayFabricHealth))%26%24select%3D              |
|                                     | Location%2CPhysicalSubContext                                                                             |
| healthRecommendations.Configuration | map[WARNING:[map[description:Recommend: Fabric Admin should verify                                        |
|                                     | that the Fabric Policy doesn't have empty topology data at                                                |
|                                     | /fabric/routing-engines/dragonfly/template-routing ; Cause:                                               |
|                                     | Fabric Policy's topology policy snapshot doesn't have any                                                 |
|                                     | data to update LAG issueTime:Thu Jun 17 14:21:06 IST 2021                                                 |
|                                     | resourceLink:http://10.0.2.15:8000/fabric/topology-policies/template-policy]]]                            |
| healthStatus.Configuration          | WARNING                                                                                                   |
+-------------------------------------+-----------------------------------------------------------------------------------------------------------+
```

Event details:


```
{
  "Timestamp": "2021-06-17T08:51:06.432Z",
  "TimestampLong": 1623919866432,
  "Location": "http://10.0.2.15:8000/fabric/topology-policies/template-policy",
  "PhysicalContext": "Configuration.FabricPolicy",
  "ParentalContext": "CrayFabricHealth",
  "ParentalIndex": 0,
  "Index": 0,
  "SubIndex": 0,
  "PhysicalSubContext": "WARNING",
  "Value": "Recommend: Fabric Admin should verify that the Fabric Policy doesn't have empty topology data at
          /fabric/routing-engines/dragonfly/template-routing ; Cause: Fabric Policy's topology policy snapshot doesn't have any data to update LAG",
  "documentVersion": 0,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:telemetry:models:Metric",
  "documentSelfLink": "/metrics/4c25187981c9b6755c4f24ec851d0",
  "documentUpdateTimeMicros": 1623919866435000,
  "documentUpdateAction": "POST",
  "documentExpirationTimeMicros": 1623923466435000,
  "documentOwner": "7e05f368-5695-43d5-8371-90b529d44f9b",
  "documentAuthPrincipalLink": "/core/authz/system-user"
}
```


The health engine reports a WARNING event during lag clear with the following information:

```
fmctl get health-engines/template-policy
```

```
+-------------------------------------+-----------------------------------------------------------------------------------------------------------+
|                 KEY                 |                                                VALUE                                                      |
+-------------------------------------+-----------------------------------------------------------------------------------------------------------+
| documentSelfLink                    | /fabric/health-engines/template-policy                                                                    |
| healthIssuesIdentified              | http://10.0.2.15:8000/metrics?$filter=((TimestampLong%20le%201623933954368)%20and%20(Timestamp            |
|                                     | Long%20ge%201623933834370)%20and%20(ParentalContext%20eq%20CrayFabricHealth))                             |
| healthObjectsImpacted               | http://10.0.2.15:8000/metrics?$filter=((TimestampLong%20le%201623933954368)%20and%20(Timestamp            |
|                                     | Long%20ge%201623933834370)%20and%20(ParentalContext%20eq%20CrayFabricHealth))%26%24select%3D              |
|                                     | Location%2CPhysicalSubContext                                                                             |
| healthRecommendations.Configuration | map[WARNING:[map[description:Recommend: Fabric Admin should                                               |
|                                     | verify that the Fabric Policy doesn't have empty topology                                                 |
|                                     | data at /fabric/routing-engines/dragonfly/template-routing ;                                              |
|                                     | Cause: Fabric Policy's topology policy snapshot doesn't have                                              |
|                                     | any data to clear LAG issueTime:Thu Jun 17 18:15:34 IST 2021                                              |
|                                     | resourceLink:http://10.0.2.15:8000/fabric/topology-policies/template-policy]]]                            |
| healthStatus.Configuration          | WARNING                                                                                                   |
+-------------------------------------+-----------------------------------------------------------------------------------------------------------+
```

Event details:


```
{
  "Timestamp": "2021-06-17T12:45:44.690Z",
  "TimestampLong": 1623933944690,
  "Location": "http://10.0.2.15:8000/fabric/topology-policies/template-policy",
  "PhysicalContext": "Configuration.FabricPolicy",
  "ParentalContext": "CrayFabricHealth",
  "ParentalIndex": 0,
  "Index": 0,
  "SubIndex": 0,
  "PhysicalSubContext": "WARNING",
  "Value": "Recommend: Fabric Admin should verify that the Fabric Policy doesn't have empty topology data at
          /fabric/routing-engines/dragonfly/template-routing ; Cause: Fabric Policy's topology policy snapshot doesn't have any data to clear LAG",
  "documentVersion": 0,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:telemetry:models:Metric",
  "documentSelfLink": "/metrics/2c73346fefbae8755c4f595e97550",
  "documentUpdateTimeMicros": 1623933944690002,
```

```
  "documentUpdateAction": "POST",
  "documentExpirationTimeMicros": 1623937544690000,
  "documentOwner": "f0d0f975-6cdc-4e05-88fc-6f376428fd7d",
  "documentAuthPrincipalLink": "/core/authz/system-user"
}
```


4.9.1.2 Description

The Fabric Manager has detected that there is no data in topology policy snapshot while calculating lag during patch operation. As a result of this, lag calculation will be skipped for the policy.

4.9.1.3 Diagnosis

In order to resolve this the recommended action is to verify that the Fabric Policy doesn't have empty topology data at /fabric/routing-engines/dragonfly/template-routing

4.10 Troubleshooting IntraGroup Connectivity Errors

Use the following event to identify and address intra group connectivity issues in Fabric policy

4.10.1 Intra Group Connectivity in fabric policy

4.10.1.1 Symptom

Below is an example on how the event will be reported on health engine. Replace "template-policy" with the name of the policy that is being used.

The health engine reports a CRITICAL event with the following information:

```
fmctl get health-engines/template-policy
+------------------------------+-----------------------------------------------------------------------------------------------------------------------+
|              KEY             |                                                        VALUE                                                          |
+------------------------------+-----------------------------------------------------------------------------------------------------------------------+
| documentSelfLink             | /fabric/health-engines/template-policy                                                                                |
| healthIssuesIdentified       | http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%201622028579762)%20and%20(TimestampLong%20ge               |
|                              | %201622028549768)%20and%20(ParentalContext%20eq%20CrayFabricHealth))                                                  |
| healthObjectsImpacted        | http://127.0.0.1:8000/metrics?$filter=((TimestampLong%20le%201622028579762)%20and%20(TimestampLong%20ge               |
|                              | %201622028549768)%20and%20(ParentalContext%20eq%20CrayFabricHealth))%26%24select%3DLocation%2C                        |
|                              | PhysicalSubContext                                                                                                    |
| healthRecommendations.Traffic | map[description:Recommend: Fabric Admin should ensure that switch group                                              |
|                              |  0 has at least one switch fully connected to all other switches in the group;                                        |
|                              | Cause: Group 0 in topology policy doesn't have any switch fully                                                       |
|                              |  connected to all other switches of the group.This can impact traffic                                                 |
|                              | performance in Dragonfly topology. issueTime:Wed May 26 06:29:10 CDT 2021                                             |
|                              | resourceLink:http://127.0.0.1:8000/fabric/topology-policies/template-policy]]]                                        |
| healthStatus.Traffic         | CRITICAL                                                                                                              |
+------------------------------+-----------------------------------------------------------------------------------------------------------------------+
```

Event details:


```
{
  "Timestamp": "2021-05-26T06:29:10.301Z",
  "TimestampLong": 1620818683301,
  "Location": "http://172.29.68.22:8000/fabric/topology-policies/template-policy",
  "PhysicalContext": "Traffic.DisconnectedMulticastGroups",
  "ParentalContext": "CrayFabricHealth",
  "ParentalIndex": 0,
  "Index": 0,
  "SubIndex": 0,
  "PhysicalSubContext": "CRITICAL",
  "Value": "Recommend: Fabric Admin should ensure that switch group 0 has at least one switch fully connected to all other switches in the group ;
          Cause: Group 0 in topology policy doesn't have any switch fully connected to all other switches of the group. This can impact traffic performan
  "documentVersion": 0,
  "documentEpoch": 0,
  "documentKind": "com:services:fabric:telemetry:models:Metric",
  "documentSelfLink": "/metrics/4b63dc5b46fc38755c22041dd1070",
  "documentUpdateTimeMicros": 1620818683302001,
  "documentUpdateAction": "POST",
  "documentExpirationTimeMicros": 1620905083302000,
  "documentOwner": "5c07323f-af89-4bef-b804-55948a4ef36c",
  "documentAuthPrincipalLink": "/core/authz/system-user"
}
```


4.10.1.2 Description

The Fabric Manager has detected that in a particular switch group of the fabric policy, there is not a single switch which has inter connections to all the other switches of the group. This can impact traffic performance due to congestion bottlenecks in the dragonfly topology.

4.10.1.3 Diagnosis

In order to resolve this the recommended action is to verify the fabric links between all the switches of mentioned switch group for any missing fabric links or wrong connections.

4.11 Troubleshooting Issues with Fabric Topology

Use these events and techniques to identify and address fabric topology policy related issues.

4.11.1 All fabric links are offline on the topology policy, skipping route calculation

4.11.1.1 Symptom

Below is an example on how the event will be reported on health engine. Replace "template-policy" with the name of the policy that is being used.

The health engine reports a WARNING event with the following information:

```
fmctl get health-engines/template-policy
+-----------------------------------+-----------------------------------------------------------------------------------------------------------+
|                KEY                |                                                VALUE                                                      |
+-----------------------------------+-----------------------------------------------------------------------------------------------------------+
| documentSelfLink                  | /fabric/health-engines/template-policy                                                                    |
| healthIssuesIdentified            | http://10.0.2.15:8003/metrics?$filter=((TimestampLong%20le%201624892333557)%20and%20(Timestamp            |
|                                   | Long%20ge%201624892213435)%20and%20(ParentalContext%20eq%20CrayFabricHealth))                             |
| healthObjectsImpacted             | http://10.0.2.15:8003/metrics?$filter=((TimestampLong%20le%201624892333557)%20and%20(Timestamp            |
|                                   | Long%20ge%201624892213435)%20and%20(ParentalContext%20eq%20CrayFabricHealth))                             |
|                                   | %26%24select%3DLocation%2CPhysicalSubContext                                                              |
| healthRecommendations.Traffic     | map[WARNING:[map[description:Recommend: Admin should Verify and ensure the                                |
|                                   | state of all switches and links are online.  ; Cause: All fabric links are                                |
|                                   | offline,skipping route calculation issueTime:Mon Jun 28 20:28:34 IST 2021                                 |
|                                   | resourceLink:http://10.0.2.15:8003/fabric/topology-policies/template-policy]]]                            |
| healthStatus.Runtime              | HEALTHY                                                                                                   |
| healthStatus.Configuration        | WARNING                                                                                                   |
| healthStatus.Traffic              | WARNING                                                                                                   |
+-----------------------------------+-----------------------------------------------------------------------------------------------------------+
```

Event details:


```
 "documents": {
    "/metrics/b75b3a57c46db2755c5d150593e80": {
      "Timestamp": "2021-06-28T10:54:30.415Z",
      "TimestampLong": 1624877670415,
      "Location": "http://10.0.2.15:8009/fabric/topology-policies/template-policy",
      "PhysicalContext": "Traffic.RedundantLink",
      "ParentalContext": "CrayFabricHealth",
      "ParentalIndex": 0,
      "Index": 0,
      "SubIndex": 0,
      "PhysicalSubContext": "WARNING",
      "Value": "Recommend: Admin should Verify and ensure the state of all switches and links are online.;
                Cause: All fabric links are offline,skipping route calculation",
      "documentVersion": 0,
      "documentEpoch": 0,
      "documentKind": "com:services:fabric:telemetry:models:Metric",
      "documentSelfLink": "/metrics/b75b3a57c46db2755c5d150593e80",
      "documentUpdateTimeMicros": 1624877670416001,
      "documentUpdateAction": "POST",
      "documentExpirationTimeMicros": 1624881270416000,
      "documentOwner": "bb304ef2-7d1c-4329-a2c8-989162cc4473"
    }
```


4.11.1.2 Description

The Fabric Manager has detected that , all fabric links in the topology policy are offline, which leads to fabric route calculation skipped.

4.11.1.3 Diagnosis

Ensure that the state of all switches and fabric links are online.

5 Cassini

5.1 Cassini Troubleshooting

5.1.1 Checking and Fixing Misconfigured Non-VLAN Tagged Ethernet Priority Code Point (PCP) Settings

VLAN tagged Ethernet frames (IEEE 802.1Q) have a PCP field which both Cassini and Rosetta use to map to internal queue resources. For non-VLAN tagged Ethernet frames, both Cassini and Rosetta need to map non-VLAN tagged Ethernet to a PCP still. In addition, failure to use the same PCP value on Cassini and Rosetta will break pause configurations. The following error message on the host could be reported if Cassini and Rosetta have misconfigured PCP values.

```
[ 6283.556807] cxi_core 0000:c1:00.0: HNI error: pfc_fifo_oflw (46) (was first error at 1686:713898606)
[ 6283.565937] cxi_core 0000:c1:00.0:   pfc_fifo_oflw_cntr: 383
[ 6283.571602] cxi_core 0000:c1:00.0: IXE error: pbuf_rd_err (48) (was first error at 1686:713903420)
[ 6283.580551] cxi_core 0000:c1:00.0:   pbuf_rd_errors: 219
```

Note: The above errors, specifically pfc_fifo_oflw errors, can also occur if the Slingshot Fabric Manager is not configured with Cassini QoS settings.

By default, the Slingshot Fabric Manager configures Rosetta to map non-VLAN tagged Ethernet frames to PCP 6. The CXI driver (cxi-core) defines a kernel module parameter, untagged_eth_pcp, to change this value. The following is an example of how to set this parameter.

```
modprobe cxi-core untagged_eth_pcp=6
```

Note: The above method is not the only way to set the cxi-core untagged_eth_pcp parameter. Standard Linux methods can also be used to configure this.

The following provides an example for how to verify what the current cxi-core untagged_eth_pcp value is.

```
# cat /sys/module/cxi_core/parameters/untagged_eth_pcp
6
```

5.1.2 Checking and Fixing LLDP Issues on the Fabric

This procedure only works for ports that are up and connected to nodes that are online. Verify if the interface is up and doesn't have an IP address assigned.

Ensure that the Rosetta switches have their appropriate LLDP TLV settings to automatically configure the NICs on the nodes. Run the following command on the FMN:

```
# cd SWITCHES/
# ./Check_LLDP_All.sh
x1000c0r1b0
     1      { "ip_addr":"10.253.8.0/16","ttl":"forever","mtu": 9000}
     2      { "ip_addr":"10.253.8.1/16","ttl":"forever","mtu": 9000}
     3      { "ip_addr":"10.253.8.2/16","ttl":"forever","mtu": 9000}
     4      { "ip_addr":"10.253.8.3/16","ttl":"forever","mtu": 9000}
     5      { "ip_addr":"10.253.8.16/16","ttl":"forever","mtu": 9000}
     6      { "ip_addr":"10.253.8.17/16","ttl":"forever","mtu": 9000}
     7      { "ip_addr":"10.253.8.18/16","ttl":"forever","mtu": 9000}
     8      { "ip_addr":"10.253.8.19/16","ttl":"forever","mtu": 9000}
     9      { "ip_addr":"10.253.8.32/16","ttl":"forever","mtu": 9000}
    10      { "ip_addr":"10.253.8.33/16","ttl":"forever","mtu": 9000}
    11      { "ip_addr":"10.253.8.34/16","ttl":"forever","mtu": 9000}
    12      { "ip_addr":"10.253.8.35/16","ttl":"forever","mtu": 9000}
    13      { "ip_addr":"10.253.8.48/16","ttl":"forever","mtu": 9000}
    14      { "ip_addr":"10.253.8.49/16","ttl":"forever","mtu": 9000}
    15      { "ip_addr":"10.253.8.50/16","ttl":"forever","mtu": 9000}
    16      { "ip_addr":"10.253.8.51/16","ttl":"forever","mtu": 9000}
x1000c3r3b0
x1000c3r5b0
     1      {"ip_addr":"10.253.8.128/16","ttl":"forever","mtu": 1500}
     2      {"ip_addr":"10.253.8.129/16","ttl":"forever","mtu": 1500}
     3      {"ip_addr":"10.253.8.130/16","ttl":"forever","mtu": 1500}
     4      {"ip_addr":"10.253.8.131/16","ttl":"forever","mtu": 1500}
     5      {"ip_addr":"10.253.8.144/16","ttl":"forever","mtu": 1500}
     6      {"ip_addr":"10.253.8.145/16","ttl":"forever","mtu": 1500}
     7      {"ip_addr":"10.253.8.146/16","ttl":"forever","mtu": 1500}
(SNIP)
    12      { "ip_addr":"10.253.27.227/16","ttl":"forever","mtu": 9000}
    13      { "ip_addr":"10.253.27.240/16","ttl":"forever","mtu": 9000}
    14      { "ip_addr":"10.253.27.241/16","ttl":"forever","mtu": 9000}
    15      { "ip_addr":"10.253.27.242/16","ttl":"forever","mtu": 9000}
    16      { "ip_addr":"10.253.27.243/16","ttl":"forever","mtu": 9000}
```

```
x1001c4r1b0
     1      { "ip_addr":"10.253.35.192/16","ttl":"forever","mtu": 9000}
     2      { "ip_addr":"10.253.35.193/16","ttl":"forever","mtu": 9000}
     3      { "ip_addr":"10.253.35.194/16","ttl":"forever","mtu": 9000}
     4      { "ip_addr":"10.253.35.195/16","ttl":"forever","mtu": 9000}
     5      { "ip_addr":"10.253.35.208/16","ttl":"forever","mtu": 9000}
     6      { "ip_addr":"10.253.35.209/16","ttl":"forever","mtu": 9000}
     7      { "ip_addr":"10.253.35.210/16","ttl":"forever","mtu": 9000}
     8      { "ip_addr":"10.253.35.211/16","ttl":"forever","mtu": 9000}
     9      { "ip_addr":"10.253.35.224/16","ttl":"forever","mtu": 9000}
    10      { "ip_addr":"10.253.35.225/16","ttl":"forever","mtu": 9000}
    11      { "ip_addr":"10.253.35.226/16","ttl":"forever","mtu": 9000}
    12      { "ip_addr":"10.253.35.227/16","ttl":"forever","mtu": 9000}
    13      { "ip_addr":"10.253.35.240/16","ttl":"forever","mtu": 9000}
    14      { "ip_addr":"10.253.35.241/16","ttl":"forever","mtu": 9000}
    15      { "ip_addr":"10.253.35.242/16","ttl":"forever","mtu": 9000}
    16      { "ip_addr":"10.253.35.243/16","ttl":"forever","mtu": 9000}
#
```

Every x100x Series switch should be followed by 16 IP Addresses. If you see any that are not followed by anything, you have a problem and need to go to the next step.

5.1.3 Fixing LLDP TLV Settings on a Switch

To reconfigure a switch that has lost it's LLDP TLV settings, run the following script on the FMN, against that switch:

```
# cd SWITCHES/
# ./Set_LLDP_One.sh x1000c3r3b0
```

5.2 Cassini Host Troubleshooting

5.2.1 Retry Handler

Each Cassini must have a Retry Handler daemon running, and they are indexed by device number (cxi0, cxi1, …). Note: These instructions refer to cxi0 in examples.

`systemd` and `udev` should automatically handle starting the retry handler but in case of issues, it should be checked:

```
# systemctl status -q cxi_rh@cxi0
  cxi_rh@cxi0.service - Cassini Retry Handler on cxi0
   Loaded: loaded (/usr/lib/systemd/system/cxi_rh@.service; static; vendor preset: disabled)
   Active: active (running) since Tue 2021-02-09 10:44:08 CST; 1min 17s ago
```

If the retry handler failed to start, verify that the corresponding `/dev/cxi<n>` device is present, and check for errors in dmesg.

```
# ls /dev/ | grep cxi

# journalctl | grep cxi_core
or
# dmesg -T |grep cxi
```

If there is nothing suspicious, you can manually start the retry handler again

```
# systemctl start cxi_rh@cxi0.service
```

If there are persistent issues or if the retry handler has crashed, it is useful to gather logs:

```
# journalctl --output=short-precise -u cxi_rh@cxi0 >> $LOG_PATH
```

There are retry handler counters that are also useful to gather for debugging. These files can be found in /run/cxi/cxi.

```
# ls /run/cxi/cxi0/
accel_close_complete   nack_no_matching_conn  nacks         sct_in_use    spt_timeouts_o  trs_in_use
cancel_tct_closed      nack_no_target_conn    nack_sequence_error   sct_timeouts  spt_timeouts_u
connections_cancelled  nack_no_target_mst     pkts_cancelled_o      smt_in_use    srb_in_use
ignored_sct_timeouts   nack_no_target_trs     pkts_cancelled_u      spt_in_use    tct_in_use
mst_in_use         nack_resource_busy     rh_sct_status_change  spt_timeouts  tct_timeouts
```

5.2.2 Cassini Loopback Mode

The Cassini NIC has an internal-loopback feature that causes transmitted data to be looped back into the NIC, bypassing any attached cable and switch.

The following command will put a Cassini with the ethernet interface name `hsn0` into loopback mode:

```
ethtool --set-priv-flags hsn0 internal-loopback on
```

Once in loopback mode, tests such as Libfabric's `fi_pingpong` can be run with both processes on the same node.

To exit loopback mode, either restart the node, or reload the Cassini drivers as follows:

```
systemctl stop 'cxi_rh@cxi*.service'
modprobe -r -a cxi-eth cxi-user cxi-core sbl
modprobe -a sbl cxi-core cxi-user cxi-eth
```

5.2.3 Ethernet driver configuration

To optimize performance, the Ethernet driver (`cxi-eth`) offers several parameters that can be changed when loading said driver.

5.2.3.1 Receive Side Scaling (RSS) support

With RSS, each received packet is hashed, and given to the corresponding queue.

The module parameter `max_rss_queues` defines the maximum number of queues that can be used. It defaults to 16, and can be set as a power of 2, up to 64.

The current number of RX queues can be dynamically changed, and must be a power of 2, up to the value defined by `max_rss_queues`.

By default, each interface is configured with only one receive queue.

To change the number of RX queues:

```
$ ethtool -L hsn0 rx 4
```

To see the RSS queues:

```
$ ethtool -l hsn0

Channel parameters for hsn0:
Pre-set maximums:
RX:             4
TX:             1
Other:          0
Combined:       1
Current hardware settings:
RX:             1
TX:             1
Other:          1
Combined:       1
```

When a new number of RSS entries is programmed, the indirection table is reset. The indirection table has 64 entries by default, and that can be changed by the `rss_indir_size` module parameter. The value must be a power of 2, up to 2048.

Show the indirection table:

```
$ ethtool -x hsn0

RX flow hash indirection table for hsn0 with 4 RX ring(s):
    0:      0     1     2     3     0     1     2     3
    8:      0     1     2     3     0     1     2     3
   16:      0     1     2     3     0     1     2     3
   24:      0     1     2     3     0     1     2     3
   32:      0     1     2     3     0     1     2     3
   40:      0     1     2     3     0     1     2     3
   48:      0     1     2     3     0     1     2     3
   56:      0     1     2     3     0     1     2     3
```

In this example, a packet with a hash of 58 will go to RSS queue number 2.

The driver is currently configured with default hashes:

```
C_RSS_HASH_IPV4_TCP
C_RSS_HASH_IPV4_UDP
C_RSS_HASH_IPV6_TCP
C_RSS_HASH_IPV6_UDP
```

There is no way to change these values during operation, as ethtool doesn't exactly support the Cassini hash mechanism.

Program the indirection table to equally split the traffic between queues 0 and 1 only:

```
$ ethtool -X hsn0 equal 2
```

5.2.3.2 Buffer sizing

Small received packets are sharing a set of large buffers, while bigger packets are received in individual smaller buffers. In other words, a large buffer will receive many small packets, while a small buffer (still enough to receive an MTU packet) will receive only one large packet before being re-used.

Internally, each large packet buffer is used and freed after receiving one single packet, while the small packet buffers will receive many small packets before being freed.

The driver provides the following tuning parameters for these buffers:

- `small_pkts_buf_size`: size of one large buffer receiving small packets. It defaults to 1MB.
- `small_pkts_buf_count`: number of large buffers to make available. It defaults to 4.
- `large_pkts_buf_count`: number of small buffers intended to receive larger packets. Their size is always 4KB. Defaults to 64.
- `buffer_threshold`: Ethernet packets with length greater than or equal to this threshold are put in one large packet buffer, otherwise they land in a shared small packet buffer. Defaults to 256.

Depending on the type of traffic, it might be more efficient to have more buffer of one type or the other, and/or have a different threshold.

A jumbo packets (eg. 9000 bytes) will be split into 2 large packets buffers of 4KB, and the remainder will be stored in the current small packets buffer. The linux stack can process these packets without re-assembling the data.

5.2.3.3 SoftRoCE (rxe) support

This support is still experimental, as the rxe driver needs some changes.

Cassini can do RoCE checksum offload when the parameter `roce_csum_offload` is set to True. This is currently set to false.

Special RoCE packet segmentation, which separates headers from payload, and removes the iCRC, marks the packet as valid so the rxe driver will not have to validate it again. This is disabled by default, and can be enabled by ethtool:

```
ethtool --set-priv-flags hsn0 roce-seg on
```

5.2.3.4 Jumbo frames

By default, each ethernet interface has an MTU of 1500. Cassini supports jumbo frames up to 9000, which will improve performances for large transfers.

To change the MTU:

```
ip link set dev hsn0 mtu 9000
```

5.2.3.5 Other Ethernet module parameters

Small Ethernet packets, up to 224 bytes, can be inlined in a Cassini command instead of using a more costly DMA operation. `idc_dma_threshold` sets the threshold for these packets and defaults to the maximum possible of 224 bytes, including MAC headers. This value can be safely changed while the device is active.

The driver also support multiple transmit (TX) queues. Linux will round-robin these queues when sendiong packets. The module parameter `max_tx_queues` defines the number of queues to create, and defaults to 16. `max_tx_queues` can be set from 1 to 64.

`lpe_cdt_thresh_id` controls the Cassini LPE append credit id to use. Its value can be 0 to 3. Do not change.

5.2.4 Cassini NIC Errors

Each block in the The Cassini ASIC contains a set of error flags. Error flags are defined in section 13 of the Cassini Software Developer's Guide. Errors vary in severity. Certain types of errors, like an "invalid VNI" error, are expected after an application terminates abnormally. Others, like a multi-bit error, may signal the need for a NIC reset.

An interrupt is generated when an error is raised. The Cassini driver handles these interrupts. For each interrupt, the Cassini driver reports error events and resets all flags.

Error events are sent to multiple places:

- The kernel console
- Kernel trace events (as used by rasdaemon)
- Netlink sockets
- Error events are rate-limited to avoid overwhelming the console. The driver will mask an error interrupt bit if its rate becomes too high.

An example of an error reported to the kernel console is shown below.

```
# dmesg -T |grep cxi
...
[Fri Feb 19 16:04:20 2021] cxi_core 0000:21:00.0: EE error: eq_rsrvn_uflw (38)
[Fri Feb 19 16:04:20 2021] cxi_core 0000:21:00.0:   C_EE_ERR_INFO_RSRVN_UFLW 1000000001430100
[Fri Feb 19 16:04:20 2021] cxi_core 0000:21:00.0:   eq_rsrvn_uflw_err_cntr: 12
...
```

5.2.5 PCIe Interface Troubleshooting

Cassini supports a PCIe Gen4 x16 interface. The standard Linux lspci command can be used to validate that the device has tuned to its full speed:

```
# lspci -vvvs `ethtool -i hsn0 | grep bus-info | cut -f2 -d' '` | grep LnkSta:
        LnkSta: Speed 16GT/s, Width x16, TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
```

PCIe AER should be enabled in the host OS. If enabled, PCIe errors will be reported to the kernel console. These errors can be aggregated using syslog or rasdaemon.

5.2.6 Ethernet Interface Troubleshooting

This section describes how to monitor the health of the Cassini Ethernet Interface. Standard tools are used to show Ethernet interface health.

5.2.6.1 ip

'ip' is a standard Linux tool to monitor the state of a Linux network device. This tool can be used to view the state of the Cassini Ethernet interface as follows:

```
# ip l show hsn0
3: hsn0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UNKNOWN mode DEFAULT group default qlen 1000
    link/ether 02:00:00:00:00:f3 brd ff:ff:ff:ff:ff:ff
```

This command shows several pieces of important information:

- `LOWER_UP` - This flag shows the state of the L1 link. Note that this is relevant to the RDMA interface as well.
- `UP` - This flag shows the administrative state of the Ethernet L2 interface.
- `state UNKNOWN` - The operational state of the interface. UNKNOWN and UP are valid running states for the L2 Ethernet interface.
- `02:00:00:00:00:f3` - Algorithmic MAC address. The prefix '0x02' indicates a locally administered unicast address. This is one signature of a managed AMA.

5.2.6.2 ping

The standard Linux `ping` command can be used to test basic Ethernet L2 function between a pair of Cassini NICs.

5.2.7 RDMA Interface Troubleshooting

This section describes how to monitor the health of the Cassini RDMA interface.

5.2.7.1 Libcxi Utilities

The libcxi utility package contains a set of diagnostics that has been developed for troubleshooting Cassini RMDA issues without using libfabric. These include a tool to display device information and status, a number of bandwidth and latency benchmarks, and a thermal diagnostic. Detailed documentation is provided by the CXI Diagnostics and Utilities document.

5.2.7.2 Libfabric Utilities

The following subsections address the utilities provided with libfabric package to troubleshoot libfabric over Cassini RDMA issues.

5.2.7.2.1 fi_info

The fi_info utility is provided in the libfabric package. This tool queries available libfabric interfaces. The following output shows a node with one Cassini (CXI) interface available:

```
# fi_info -p cxi
provider: cxi
    fabric: cxi
    domain: cxi0
    version: 0.0
    type: FI_EP_RDM
    protocol: FI_PROTO_CXI
# fi_info -p cxi -d cxi0 -v |grep state
            state: FI_LINK_UP
```

The availability of a CXI interface indicates several key signs of health. An interface is not made available unless it meets the following criteria:

- The interface retry handler is running
- A matching L2 interface is available
- The L1 interface has a temporary, locally administered, unicast address assigned to it. This is presumed to be an AMA applied by the fabric manager.
- Note that the L1 link state is also reported if verbosity is enabled. L1 link state reported by fi_info will match the state reported by the L2 device via the ip tool.

All these checks together make fi_info an excellent first tool to use to check the general health of Cassini RDMA interfaces.

5.2.7.2.2 fi_pingpong

The fi_pingpong utility is provided is also provided with libfabric. This is a basic client-server RDMA test. This is a quick and easy tool to use to validate end-to-end RDMA functionality.

```
// start server:
# fi_pingpong -I 10000 -e rdm -m tagged -p cxi -d cxi1
bytes   #sent   #ack     total       time     MB/sec    usec/xfer   Mxfers/sec
64      10k     =10k     1.2m        0.10s     12.89       4.97       0.20
256     10k     =10k     4.8m        0.08s     60.71       4.22       0.24
1k      10k     =10k     19m         0.09s    233.95       4.38       0.23
4k      10k     =10k     78m         0.09s    881.85       4.64       0.22
64k     10k     =10k     1.2g        0.19s   7067.63       9.27       0.11
1m      10k     =10k     19g         0.99s  21234.98      49.38       0.02
// start client:
#  fi_pingpong -I 10000 -e rdm -m tagged -p cxi -d cxi0 172.29.68.180
bytes   #sent   #ack     total       time     MB/sec    usec/xfer   Mxfers/sec
64      10k     =10k     1.2m        0.10s     12.90       4.96       0.20
256     10k     =10k     4.8m        0.08s     60.83       4.21       0.24
1k      10k     =10k     19m         0.09s    234.34       4.37       0.23
4k      10k     =10k     78m         0.09s    883.30       4.64       0.22
64k     10k     =10k     1.2g        0.19s   7074.11       9.26       0.11
1m      10k     =10k     19g         0.99s  21236.47      49.38       0.02
```

This tool can use any IP interface for bootstrapping the RDMA communication. The ping-pong communication pattern does not achieve the best bandwidth results, however, the 1M transfer size should achieve high enough bandwidth to make it obvious that the link is operating at the expected 200Gbps speed.

This tool can be used between any two endpoints on a fabric or looped back to the same device.

6 Rosetta

6.1 Rosetta Switch Firmware Soft Reset

Remove configurations

1. ssh into each switch
2. Run the following commands:

```
emmc-setup -U
rm -rf /nvram/*
emmc-setup -u
reboot
```

6.2 Rosetta Switch Firmware Hard Reset

Wipe out storage

1. Reboot the switch with a console cable connected
2. On the console, break into UBoot as the switch boots At the UBoot prompt type:

   ```
   run nukeemmc
   reset
   ```

   The switch will then need to tftp load the recovery image. ALL data on the switch, including all 3 boot images, will be erased.

6.3 Users may observe log messages with incorrect timestamps on the Rosetta switch controller.

6.3.1 Symptom

Slingshot Switch starting timestamp for various services can be incorrect; a specfic example is that lldpad may show that it started on Valentine's Day 2019:

```
systemctl show lldpad

[...]
ExecMainStartTimestamp=Thu 2019-02-14 10:12:03 UTC
ExecMainStartTimestampMonotonic=14350577
ExecMainExitTimestampMonotonic=0
ExecMainPID=4208
ExecMainCode=0
ExecMainStatus=0
ExecStart={ path=/usr/sbin/lldpad ; argv[]=/usr/sbin/lldpad -t ; ignore_errors=no ; start_time=[Thu 2019-02-14 10:12:03 UTC] ; stop_time=[n/ a] ; pid=4208
[...]
```

6.3.2 Description

After initial power up and certain types of reboots, the Rosetta/Slingshot switch controller may have log messages with incorrect timestamps due to the fact that it does not possess a battery backed real-time clock on the board. Log messages generated prior to the switch having its time set correctly by NTP can result in the time shown for the start of a service (in particular, `lldpad`) as February 14, 2019. To be more specific, affected logs are any generated before the management ethernet device is online and any messages subsequent to that but before before the NTP service is finished setting the correct time.

While it could be possible to rewrite log messages with correct timestamps after the completion of the NTP service initial setting, it is not done because doing so could mask certain types of failures, such as a failure in the NTP services.

7 CXI Diagnostics and Utilities

7.1 Overview

The Cassini software stack includes the libcxi library, which provides a direct interface to the CXI driver. A set of diagnostics has been developed using this library. These can be used to measure performance and troubleshoot Cassini issues without using libfabric.

The bandwidth and latency utilities can measure either loopback or point-to point performance. They can be used to get a quick system-wide snapshot of NIC and link health, as well as highlight switch edge port configuration issues. When used point-to-point they can be helpful in isolating localized problems. Point-to-point runs may also discover switch fabric port configuration issues, though other tools like dgnettest are better suited for this purpose.

The cray-libcxi-utils RPM is included in the slingshot-compute-cassini tarball in the Slingshot release package.

The cray-libcxi-utils package can be installed on both compute and non-compute nodes. Each program has a corresponding man page, and there is a cxi_diags(7) page with a summary of the diagnostics and their features. Binaries are placed in /usr/bin, and man pages are placed in /usr/share/man/man1 and /usr/share/man/man7.

7.2 Minimum Setup

Always:

- The CXI drivers must be installed.
- The retry handler service must be running for each NIC (cxi_rh@cxi0, cxi_rh@cxi1, etc.).
- The links must have a valid Algorithmic MAC Address (AMA) configured.
- The client and server hosts must each have a network device with an IPv4 address, and they must be able to reach each other through these devices.

When using internal loopback:

- The link must be configured in internal-loopback mode.

When running point-to-point (this includes using the same NIC for client and server, unless internal-loopback is used):

- The fabric switches must have routing and QoS configuration applied.
- The links must be up.

When using GPU buffers with cxi_gpu_loopback_bw or cxi_heatsink_check:

- The appropriate runtime library must be installed (HIP for AMD or NVIDIA, or CUDA for NVIDIA).

7.3 Running the Diagnostics

A subset of the diagnostics are client-server tests that measure point-to-point or loopback bandwidth and latency. These can use any IP interface to bootstrap the high-speed network connection between the client and server. They share many common command line arguments and print results in a similar format. The server accepts optional arguments to specify which CXI device to use and which TCP port to listen on for the client connection. The client must be supplied with the hostname or IPv4 address of the server, along with the port to connect to if the default value was not used with the server. The client must also be supplied with all desired run options, which it shares with the server after establishing a connection.

Generally only a single client-server pair is needed. However, when measuring Atomic Memory Operation (AMO) bandwidth the RDMA write sizes are small enough that a single client-server pair cannot reach full bandwidth. It is possible to run multiple pairs simultaneously by providing a unique TCP port to each pair. If the run duration is sufficiently long, the measured bandwidths can be combined with minimal error due to offsets in the start and end times of each client-server pair.

There are several diagnostics that run as single programs. One, cxi_stat, provides CXI device information. The others, cxi_gpu_bw_loopback and cxi_heatsink_check, generate network traffic but can only be configured to use the same NIC for both initiator and target.

7.4 Device Information

7.4.1 cxi_stat

The cxi_stat utility displays a summary of information provided by the CXI driver. By default it displays information for all available devices. It can be limited to a single device with the `--device` option. Additional HSN link error rate and pause information can be shown with the `--rates` option.

Usage

```
cxi_stat - CXI device status utility
Usage: -hlm
 -h --help             Show this help
 -l --list             List all CXI devices
 -m --mac-list         List all CXI MAC addresses
 -r --rates            Report codeword rates and pause percentages
 -d --device=DEV       List only specified CXI device
                       Default lists all CXI devices
 -V --version          Print the version and exit
```

Example

```
$ cxi_stat -r
Device: cxi0
    Description: SS11 200Gb 2P NIC Mezz
    Part Number: P43012-001
    Serial Number: GR21120003
    FW Version: 1.6.0.326
    Network device: hsn0
    MAC: 02:00:00:00:00:12
    PID granule: 256
    PCIE speed/width: 16 GT/s x16
    PCIE slot: 0000:21:00.0
        Link layer retry: on
        Link loopback: off
        Link media: electrical
        Link MTU: 2112
        Link speed: BS_200G
        Link state: up
    Rates:
        Good CW: 39066186/s
        Corrected CW: 348/s
        Uncorrected CW: 0/s
        Corrected BER: 1.638e-09
        Uncorrected BER: <1.176e-12
        TX Pause state: pfc/802.1qbb
        RX Pause state: pfc/802.1qbb
            RX Pause PCP 0: 0.0%
            TX Pause PCP 0: 0.0%
            RX Pause PCP 1: 0.0%
            TX Pause PCP 1: 0.0%
            RX Pause PCP 2: 0.0%
            TX Pause PCP 2: 0.0%
            RX Pause PCP 3: 0.0%
            TX Pause PCP 3: 0.0%
            RX Pause PCP 4: 0.0%
            TX Pause PCP 4: 0.0%
            RX Pause PCP 5: 0.0%
            TX Pause PCP 5: 0.0%
            RX Pause PCP 6: 0.0%
            TX Pause PCP 6: 0.0%
            RX Pause PCP 7: 0.0%
            TX Pause PCP 7: 0.0%
```

7.5 Bandwidth

Bandwidth is calculated using only the frame payload. The client-server diagnostics can measure uni-directionally or bi-directionally. Uni-directional bandwidth is measured by the client, shared with the server, and reported by both. Bi-directional bandwidth is measured by both the client and server, who share their results and report the combined value.

The diagnostics can be run for a number of iterations or for a duration of time. They can be configured to use a single size or a range of sizes, with the exception of cxi_gpu_bw_loopback which does not support ranges. They can be configured to use either system memory or GPU memory for the initiator and target write buffers, with the exception of cxi_atomic_bw. A summary of the run options is printed during initialization. This is followed by several columns of data, including the transaction size, number of transactions, measured bandwidth, and measured transaction rate.

7.5.1 cxi_write_bw

The cxi_write_bw utility measures one-sided RDMA write bandwidth.

Usage

```
Usage:
  cxi_write_bw [-d DEV] [-p PORT]
                        Start a server and wait for connection
  cxi_write_bw ADDR [OPTIONS]
                        Connect to server with address ADDR

Options:
  -d, --device=DEV      Device name (default "cxi0")
  -p, --port=PORT       The port to listen on/connect to (default 49194)
  -t, --tx-gpu=GPU      GPU index for allocating TX buf (default no GPU)
  -r, --rx-gpu=GPU      GPU index for allocating RX buf (default no GPU)
  -g, --gpu-type=TYPE   GPU type (AMD or NVIDIA) (default type determined
                        by discovered GPU files on system)
  -n, --iters=ITERS     Number of iterations to run (default 1000)
  -D, --duration=SEC    Run for the specified number of seconds
  -s, --size=MIN[:MAX]  Write size or range to use (default 65536)
                        Ranges must be powers of 2 (e.g. "1:8192")
                        The maximum size is 4294967295
```

```
  -l, --list-size=SIZE   Number of writes per iteration, all pushed to
                         the Tx CQ prior to initiating xfer (default 256)
  -b, --bidirectional    Measure bidirectional bandwidth
      --no-hrp           Do not use High Rate Puts for sizes <= 2048 bytes
      --no-idc           Do not use Immediate Data Cmds for sizes <= 224 bytes
      --buf-sz=SIZE      The max TX/RDMA buffer size, specified in bytes
                         If (size * list_size) > buf_sz, writes will overlap
                         (default is 4294967296)
      --buf-align=ALIGN  Byte-alignment of writes in the buffer (default 64)
  -h, --help             Print this help text and exit
  -V, --version          Print the version and exit
```

Example

This example shows a bi-directional run over a range of sizes for 5 seconds each.

Server

```
$ cxi_write_bw
Listening on port 49194 for client to connect...
---------------------------------------------------
    CXI RDMA Write Bandwidth Test
Device           : cxi0
Client TX Mem    : System
Server RX Mem    : System
Test Type        : Duration
Duration         : 5 seconds
Min Write Size   : 1024
Max Write Size   : 65536
List Size        : 256
HRP              : Enabled
IDC              : Enabled
Bidirectional    : Enabled
Max RDMA Buf     : 4294967296
RDMA Buf Align   : 64
Local (server)   : NIC 0x12 PID 0 VNI 10
Remote (client)  : NIC 0x13 PID 0
---------------------------------------------------
RDMA Size[B]      Writes  BW[MB/s]  PktRate[Mpkt/s]
        1024    71568640  29314.66        28.627597
        2048    44694528  36613.15        17.877517
        4096    21218560  34763.74        16.974481
        8192    11737856  38461.53        18.780044
       16384     6154752  40333.92        19.694297
       32768     3153664  41334.64        20.182929
       65536     1522688  40550.70        19.800147
---------------------------------------------------
```

Client

```
$ cxi_write_bw 10.1.1.8 -D 5 -s 1024:65536 -b
---------------------------------------------------
    CXI RDMA Write Bandwidth Test
Device           : cxi0
Client TX Mem    : System
Server RX Mem    : System
Test Type        : Duration
Duration         : 5 seconds
Min Write Size   : 1024
Max Write Size   : 65536
List Size        : 256
HRP              : Enabled
IDC              : Enabled
Bidirectional    : Enabled
Max RDMA Buf     : 4294967296
RDMA Buf Align   : 64
Local (client)   : NIC 0x13 PID 0 VNI 10
Remote (server)  : NIC 0x12 PID 0
---------------------------------------------------
RDMA Size[B]      Writes  BW[MB/s]  PktRate[Mpkt/s]
        1024    71572224  29314.66        28.627597
        2048    44694272  36613.15        17.877517
        4096    21218304  34763.74        16.974481
        8192    11737856  38461.53        18.780044
       16384     6154752  40333.92        19.694297
       32768     3153664  41334.64        20.182929
       65536     1571328  40550.70        19.800147
---------------------------------------------------
```

7.5.2 cxi_read_bw

The cxi_read_bw utility measures one-sided RDMA read bandwidth.

Usage

```
Usage:
  cxi_read_bw [-d DEV] [-p PORT]
                        Start a server and wait for connection
  cxi_read_bw ADDR [OPTIONS]
                        Connect to server with address ADDR

Options:
  -d, --device=DEV      Device name (default "cxi0")
  -p, --port=PORT       The port to listen on/connect to (default 49194)
  -t, --tx-gpu=GPU      GPU index for allocating TX buf (default no GPU)
  -r, --rx-gpu=GPU      GPU index for allocating RX buf (default no GPU)
  -g, --gpu-type=TYPE   GPU type (AMD or NVIDIA) (default type determined
                        by discovered GPU files on system)
  -n, --iters=ITERS     Number of iterations to run (default 1000)
  -D, --duration=SEC    Run for the specified number of seconds
  -s, --size=MIN[:MAX]  Read size or range to use (default 65536)
                        Ranges must be powers of 2 (e.g. "1:8192")
                        The maximum size is 4294967295
  -l, --list-size=SIZE  Number of reads per iteration, all pushed to
                        the Tx CQ prior to initiating xfer (default 256)
  -b, --bidirectional   Measure bidirectional bandwidth
      --buf-sz=SIZE     The max TX/RDMA buffer size, specified in bytes
                        If (size * list_size) > buf_sz, reads will overlap
                        (default is 4294967296)
      --buf_align=ALIGN Byte-alignment of reads in the buffer (default 64)
  -h, --help            Print this help text and exit
  -V, --version         Print the version and exit
```

Example

This example shows a quick run using the default options.

Server

```
$ cxi_read_bw
Listening on port 49194 for client to connect...
----------------------------------------------------
    CXI RDMA Read Bandwidth Test
Device           : cxi0
Client TX Mem    : System
Server RX Mem    : System
Test Type        : Iteration
Iterations       : 1000
Read Size        : 65536
List Size        : 256
Bidirectional    : Disabled
Max RDMA Buf     : 4294967296
RDMA Buf Align   : 64
Local (server)   : NIC 0x12 PID 0 VNI 10
Remote (client)  : NIC 0x13 PID 0
----------------------------------------------------
RDMA Size[B]       Reads  BW[MB/s]  PktRate[Mpkt/s]
       65536           -  21479.54        10.488056
----------------------------------------------------
```

Client

```
$ cxi_read_bw nid000018
----------------------------------------------------
    CXI RDMA Read Bandwidth Test
Device           : cxi0
Client TX Mem    : System
Server RX Mem    : System
Test Type        : Iteration
Iterations       : 1000
Read Size        : 65536
List Size        : 256
Bidirectional    : Disabled
Max RDMA Buf     : 4294967296
RDMA Buf Align   : 64
Local (client)   : NIC 0x13 PID 0 VNI 10
Remote (server)  : NIC 0x12 PID 0
----------------------------------------------------
RDMA Size[B]       Reads  BW[MB/s]  PktRate[Mpkt/s]
       65536      256000  21479.54        10.488056
----------------------------------------------------
```

7.5.3 cxi_send_bw

The cxi_send_bw utility measures two-sided message bandwidth. It can be configured to use eager or rendezvous transactions.

Usage

```
Usage:
  cxi_send_bw [-d DEV] [-p PORT]
                        Start a server and wait for connection
  cxi_send_bw ADDR [OPTIONS]
                        Connect to server with address ADDR

Options:
  -d, --device=DEV      Device name (default "cxi0")
  -p, --port=PORT       The port to listen on/connect to (default 49194)
  -t, --tx-gpu=GPU      GPU index for allocating TX buf (default no GPU)
  -r, --rx-gpu=GPU      GPU index for allocating RX buf (default no GPU)
  -g, --gpu-type=TYPE   GPU type (AMD or NVIDIA) (default type determined
                        by discovered GPU files on system)
  -n, --iters=ITERS     Number of iterations to run (default 1000)
  -D, --duration=SEC    Run for the specified number of seconds
  -s, --size=MIN[:MAX]  Send size or range to use (default 65536)
                        Ranges must be powers of 2 (e.g. "1:8192")
                        The maximum size is 4294967295
  -l, --list-size=SIZE  Number of sends per iteration, all pushed to
                        the Tx CQ prior to initiating xfer (default 256)
  -b, --bidirectional   Measure bidirectional bandwidth
  -R, --rdzv            Use rendezvous PUTs
      --no-idc          Do not use Immediate Data Cmds for sizes <= 224 bytes
      --buf-sz=SIZE     The max TX/RDMA buffer size, specified in bytes
                        If (size * list_size) > buf_sz, sends will overlap
                        (default is 4294967296)
      --buf-align=ALIGN Byte-alignment of sends in the buffer (default 64)
  -h, --help            Print this help text and exit
  -V, --version         Print the version and exit
```

Example

This example shows a quick run using the default options.

Server

```
$ cxi_send_bw
Listening on port 49194 for client to connect...
---------------------------------------------------
    CXI RDMA Send Bandwidth Test
Device           : cxi0
Client TX Mem    : System
Server RX Mem    : System
Test Type        : Iteration
Iterations       : 1000
Send Size        : 65536
List Size        : 256
IDC              : Enabled
Bidirectional    : Disabled
Max RDMA Buf     : 4294967296
RDMA Buf Align   : 64
Local (server)   : NIC 0x12 PID 0 VNI 10
Remote (client)  : NIC 0x13 PID 0
---------------------------------------------------
Send Size[B]       Sends  BW[MB/s]  PktRate[Mpkt/s]
       65536           -  23772.52        11.607674
---------------------------------------------------
```

Client

```
$ cxi_send_bw 192.168.1.1
---------------------------------------------------
    CXI RDMA Send Bandwidth Test
Device           : cxi0
Client TX Mem    : System
Server RX Mem    : System
Test Type        : Iteration
Iterations       : 1000
Send Size        : 65536
List Size        : 256
IDC              : Enabled
Bidirectional    : Disabled
Max RDMA Buf     : 4294967296
RDMA Buf Align   : 64
Local (client)   : NIC 0x13 PID 0 VNI 10
Remote (server)  : NIC 0x12 PID 0
---------------------------------------------------
Send Size[B]       Sends  BW[MB/s]  PktRate[Mpkt/s]
       65536      256000  23772.52        11.607674
---------------------------------------------------
```

7.5.4 cxi_atomic_bw

The cxi_atomic_bw utility measures one-sided AMO bandwidth. It can be configured to use a specific atomic operation and data type. Where possible, target buffer writes are ensured to occur with every AMO. The utility works with or without CPU offload, but enabling that feature in the NIC is left to the user.

Usage

```
Usage:
  cxi_atomic_bw [-d DEV] [-p PORT]
                        Start a server and wait for connection
  cxi_atomic_bw ADDR [OPTIONS]
                        Connect to server with address ADDR

Options:
  -d, --device=DEV      Device name (default "cxi0")
  -p, --port=PORT       The port to listen on/connect to (default 49194)
  -n, --iters=ITERS     Number of iterations to run (default 10000)
  -D, --duration=SEC    Run for the specified number of seconds
                        Ranges must be powers of 2 (e.g. "1:8192")
                        The maximum size is 4294967295
  -l, --list-size=SIZE  Number of writes per iteration, all pushed to the
                        initiator CQ prior to initiating xfer (default 4096)
  -A, --atomic-op       The atomic operation to use (default SUM)
  -C, --cswap-op        The CSWAP operation to use (default EQ)
  -T, --atomic-type     The atomic type to use (default UINT64)
  -b, --bidirectional   Measure bidirectional bandwidth
      --fetching        Use fetching AMOs
      --matching        Use matching list entries at the target
      --unrestricted    Use unrestricted AMOs
      --no-hrp          Do not use High Rate Puts
      --no-idc          Do not use Immediate Data Cmds
  -h, --help            Print this help text and exit
  -V, --version         Print the version and exit

Atomic Ops:
  MIN, MAX, SUM, LOR, LAND, BOR, BAND, LXOR, BXOR, SWAP, CSWAP, AXOR
CSWAP Ops:
  EQ, NE, LE, LT, GE, GT
Atomic Types:
  INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, UINT64, FLOAT, DOUBLE,
  FLOAT_COMPLEX, DOUBLE_COMPLEX, UINT128
```

Example

This example shows a two-second bi-directional run using the CSWAP EQ operation with uint128 data.

Server

```
$ cxi_atomic_bw -p 42000
Listening on port 42000 for client to connect...
------------------------------------------------
    CXI Atomic Memory Operation Bandwidth Test
Device           : cxi0
Test Type        : Duration
Duration         : 2 seconds
Atomic Op        : NON-FETCHING CSWAP EQ
Atomic Type      : UINT128
List Size        : 4096
HRP              : Enabled
IDC              : Enabled
Matching LEs     : Disabled
Restricted       : Enabled
Bidirectional    : Enabled
Local (server)   : NIC 0x12 PID 0 VNI 10
Remote (client)  : NIC 0x13 PID 0
------------------------------------------------
AMO Size[B]         Ops  BW[MB/s]    OpRate[M/s]
         16    67584000   1081.16      67.572741
------------------------------------------------
```

Client

```
$ cxi_atomic_bw cxi-nid0 -p 42000 -A cswap -C eq -T uint128 -D 2 -b
------------------------------------------------
    CXI Atomic Memory Operation Bandwidth Test
Device           : cxi0
Test Type        : Duration
Duration         : 2 seconds
Atomic Op        : NON-FETCHING CSWAP EQ
Atomic Type      : UINT128
List Size        : 4096
HRP              : Enabled
IDC              : Enabled
```

```
Matching LEs     : Disabled
Restricted       : Enabled
Bidirectional    : Enabled
Local (client)   : NIC 0x13 PID 0 VNI 10
Remote (server)  : NIC 0x12 PID 0
-------------------------------------------------
AMO Size[B]         Ops  BW[MB/s]    OpRate[M/s]
         16    67575808   1081.16      67.572741
-------------------------------------------------
```

7.5.5 cxi_gpu_bw_loopback

The cxi_gpu_bw_loopback utility measures one-sided RDMA write bandwidth. When using system memory, it can be configured to use hugepages.

Usage

```
Usage:
  cxi_loopback_bw [OPTIONS]

Options:
  -d, --device=DEV       Cassini device (default "cxi0")
  -t, --tx-gpu=GPU       GPU index for allocating TX buf (default no GPU)
  -r, --rx-gpu=GPU       GPU index for allocating RX buf (default no GPU)
  -g, --gpu-type=TYPE    GPU type (AMD or NVIDIA). Default type determined by
                         discovered GPU files on system.
  -D, --duration=SEC     Run for the specified number of seconds
  -i, --iters=ITERS      Number of iterations to run if duration not
                         specified (default 25)
  -n, --num-xfers=XFERS  Number of transfers per iteration, all pushed to
                         the Tx CQ prior to initiating xfer (default 8192)
  -s, --size=SIZE        Transfer Size in Bytes (default 524288)
                         The maximum size is 4294967295 bytes
  --use-hp               Attempt to use 1GB huge pages when allocating
                         system memory
  -h, --help             Print this help text and exit
  -V, --version          Print the version and exit
```

Example

This example shows a run using GPU memory for both the initiator and target buffers.

```
$ cxi_gpu_loopback_bw -D 10 -s 131072 --tx-gpu 0 --rx-gpu 2
-----------------------------------------------------
    CXI Loopback Bandwidth Test
Device          : cxi0
TX Mem          : GPU 0
RX Mem          : GPU 2
GPU Type        : AMD
Test Type       : Duration
Duration        : 10 seconds
Write Size (B)  : 131072
Cmds per iter   : 8192
Found 4 GPU(s)
-----------------------------------------------------
RDMA Size[B]      Writes  BW[Gb/s]  PktRate[Mpkt/s]
      131072     1810432    189.49        11.565970
-----------------------------------------------------
```

7.6 Latency

Latency is measured by obtaining the start and end times for individual transactions. The start time is the point when software initiates the transaction. The end time is the point when software receives an event from the device indicating that the transaction's final data write to either host has occurred. For cxi_write_lat, this is the write to the target. For cxi_read_lat, this is the write to the initiator. For cxi_atomic_lat, this is the write to the target, or for fetching atomics, the initiator. For cxi_send_lat a message is sent first from client to server and then from server to client. The end time is the point when the second message has been written, and the latency is estimated by halving the round-trip time.

Latency is measured and reported by the client. The diagnostics can be run for a number of iterations or for a duration of time. They can be configured to use a single size or a range of sizes. They can be configured to use either system memory or GPU memory for the initiator and target write buffers, with the exception of cxi_atomic_ . A summary of the run options is printed during initialization. If the `--report-all` option is used, each individually measured latency is printed. Finally several columns of data are printed, including the transaction size, number of transactions, the measured minimum, maximum, and average latencies, as well as the standard deviation.

7.6.1 cxi_write_lat

The cxi_write_lat utility measures one-sided RDMA write latency.

Usage

```
Usage:
  cxi_write_lat [-d DEV] [-p PORT]
                          Start a server and wait for connection
  cxi_write_lat ADDR [OPTIONS]
                          Connect to server with address ADDR

Options:
  -d, --device=DEV        Device name (default "cxi0")
  -p, --port=PORT         The port to listen on/connect to (default 49194)
  -t, --tx-gpu=GPU        GPU index for allocating TX buf (default no GPU)
  -r, --rx-gpu=GPU        GPU index for allocating RX buf (default no GPU)
  -g, --gpu-type=TYPE     GPU type (AMD or NVIDIA) (default type determined
                          by discovered GPU files on system)
  -n, --iters=ITERS       Number of iterations to run (default 100)
  -D, --duration=SEC      Run for the specified number of seconds
      --warmup=WARMUP     Number of warmup iterations to run (default 10)
      --latency-gap=USEC  Number of microseconds to wait between each
                          iteration (default 1000)
  -s, --size=MIN[:MAX]    Write size or range to use (default 8)
                          Ranges must be powers of 2 (e.g. "1:8192")
                          The maximum size is 4294967295
      --no-idc            Do not use Immediate Data Cmds for sizes <= 224 bytes
      --no-ll             Do not use Low-Latency command issuing
      --report-all        Report all latency measurements individually
                          This option is ignored when using --duration
  -h, --help              Print this help text and exit
  -V, --version           Print the version and exit
```

Example

This example shows a run with individual latencies printed.

Server

```
$ cxi_write_lat
Listening on port 49194 for client to connect...
------------------------------------------------------------------------
    CXI RDMA Write Latency Test
Device           : cxi0
Client TX Mem    : System
Server RX Mem    : System
Test Type        : Iteration
Iterations       : 5
Warmup Iters     : 10
Inter-Iter Gap   : 1000 microseconds
Write Size       : 8
IDC              : Enabled
LL Cmd Launch    : Enabled
Results Reported : All
Local (server)   : NIC 0x12 PID 0 VNI 10
Remote (client)  : NIC 0x13 PID 0
------------------------------------------------------------------------
See client for results.
------------------------------------------------------------------------
```

Client

```
$ cxi_write_lat 10.1.1.8 -n 5 --report-all
------------------------------------------------------------------------
    CXI RDMA Write Latency Test
Device           : cxi0
Client TX Mem    : System
Server RX Mem    : System
Test Type        : Iteration
Iterations       : 5
Warmup Iters     : 10
Inter-Iter Gap   : 1000 microseconds
Write Size       : 8
IDC              : Enabled
LL Cmd Launch    : Enabled
Results Reported : All
Local (client)   : NIC 0x13 PID 0 VNI 10
Remote (server)  : NIC 0x12 PID 0
------------------------------------------------------------------------
  WriteNum  Latency[us]
         0        2.008
         1        2.012
         2        2.008
         3        2.007
```

```
        4        2.008
------------------------------------------------------------------------
RDMA Size[B]      Writes     Min[us]     Max[us]    Mean[us]  StdDev[us]
           8           5        2.07        2.12        2.09        0.01
------------------------------------------------------------------------
```

7.6.2 cxi_read_lat

The cxi_read_lat utility measures one-sided RDMA read latency.

Usage

```
Usage:
  cxi_read_lat [-d DEV] [-p PORT]
                          Start a server and wait for connection
  cxi_read_lat ADDR [OPTIONS]
                          Connect to server with address ADDR

Options:
  -d, --device=DEV        Device name (default "cxi0")
  -p, --port=PORT         The port to listen on/connect to (default 49194)
  -t, --tx-gpu=GPU        GPU index for allocating TX buf (default no GPU)
  -r, --rx-gpu=GPU        GPU index for allocating RX buf (default no GPU)
  -g, --gpu-type=TYPE     GPU type (AMD or NVIDIA) (default type determined
                          by discovered GPU files on system)
  -n, --iters=ITERS       Number of iterations to run (default 100)
  -D, --duration=SEC      Run for the specified number of seconds
      --warmup=WARMUP     Number of warmup iterations to run (default 10)
      --latency-gap=USEC  Number of microseconds to wait between each
                          iteration (default 1000)
  -s, --size=MIN[:MAX]    Read size or range to use (default 8)
                          Ranges must be powers of 2 (e.g. "1:8192")
                          The maximum size is 4294967295
      --no-ll             Do not use Low-Latency command issuing
      --report-all        Report all latency measurements individually
                          This option is ignored when using --duration
  -h, --help              Print this help text and exit
  -V, --version           Print the version and exit
```

Example

This example shows a run over a range of sizes for 1 second each.

Server

```
$ cxi_read_lat
Listening on port 49194 for client to connect...
------------------------------------------------------------------------
    CXI RDMA Read Latency Test
Device           : cxi0
Client TX Mem    : System
Server RX Mem    : System
Test Type        : Duration
Duration         : 1 seconds
Warmup Iters     : 10
Inter-Iter Gap   : 1000 microseconds
Min Read Size    : 64
Max Read Size    : 1024
LL Cmd Launch    : Enabled
Results Reported : Summary
Local (server)   : NIC 0x12 PID 0 VNI 10
Remote (client)  : NIC 0x12 PID 1
------------------------------------------------------------------------
See client for results.
------------------------------------------------------------------------
```

Client

```
$ cxi_read_lat 192.168.1.1 -D 1 -s 1:1024
------------------------------------------------------------------------
    CXI RDMA Read Latency Test
Device           : cxi0
Client TX Mem    : System
Server RX Mem    : System
Test Type        : Duration
Duration         : 1 seconds
Warmup Iters     : 10
Inter-Iter Gap   : 1000 microseconds
Min Read Size    : 64
Max Read Size    : 1024
LL Cmd Launch    : Enabled
Results Reported : Summary
Local (client)   : NIC 0x12 PID 1 VNI 10
Remote (server)  : NIC 0x12 PID 0
```

```
-------------------------------------------------------------------------
RDMA Size[B]       Reads     Min[us]     Max[us]    Mean[us]  StdDev[us]
          64       97181        2.61       13.91        2.70        0.07
         128       97080        2.63        5.55        2.71        0.07
         256       96887        2.66        6.17        2.74        0.07
         512       96548        2.71        5.44        2.77        0.05
        1024       95613        2.78       11.57        2.87        0.07
-------------------------------------------------------------------------
```

7.6.3 cxi_send_lat

The cxi_send_bw utility measures two-sided message latency. It can be configured to use eager or rendezvous transactions.

Usage

```
Usage:
  cxi_send_lat [-d DEV] [-p PORT]
                          Start a server and wait for connection
  cxi_send_lat ADDR [OPTIONS]
                          Connect to server with address ADDR

Options:
  -d, --device=DEV        Device name (default "cxi0")
  -p, --port=PORT         The port to listen on/connect to (default 49194)
  -t, --tx-gpu=GPU        GPU index for allocating TX buf (default no GPU)
  -r, --rx-gpu=GPU        GPU index for allocating RX buf (default no GPU)
  -g, --gpu-type=TYPE     GPU type (AMD or NVIDIA) (default type determined
                          by discovered GPU files on system)
  -n, --iters=ITERS       Number of iterations to run (default 100)
  -D, --duration=SEC      Run for the specified number of seconds
      --warmup=WARMUP     Number of warmup iterations to run (default 10)
      --latency-gap=USEC  Number of microseconds to wait between each
                          iteration (default 1000)
  -s, --size=MIN[:MAX]    Send size or range to use (default 8)
                          Ranges must be powers of 2 (e.g. "1:8192")
                          The maximum size is 4294967295
      --no-idc            Do not use Immediate Data Cmds for sizes <= 224 bytes
      --no-ll             Do not use Low-Latency command issuing
  -r, --rdzv              Use rendezvous PUTs
      --report-all        Report all latency measurements individually
                          This option is ignored when using --duration
  -h, --help              Print this help text and exit
  -V, --version           Print the version and exit
```

Example

This example shows a quick run using the default options.

Server

```
$ cxi_send_lat
Listening on port 49194 for client to connect...
-------------------------------------------------------------------------
    CXI RDMA Send Latency Test
Device           : cxi0
Client TX Mem    : System
Server RX Mem    : System
Test Type        : Iteration
Iterations       : 100
Warmup Iters     : 10
Inter-Iter Gap   : 1000 microseconds
Send Size        : 8
IDC              : Enabled
LL Cmd Launch    : Enabled
Results Reported : Summary
Local (server)   : NIC 0x12 PID 0 VNI 10
Remote (client)  : NIC 0x13 PID 0
-------------------------------------------------------------------------
See client for results.
-------------------------------------------------------------------------
```

Client

```
$ cxi_send_lat 192.168.1.1 -s 1:1024
-------------------------------------------------------------------------
    CXI RDMA Send Latency Test
Device           : cxi0
Client TX Mem    : System
Server RX Mem    : System
Test Type        : Iteration
Iterations       : 100
Warmup Iters     : 10
Inter-Iter Gap   : 1000 microseconds
Send Size        : 8
```

```
IDC              : Enabled
LL Cmd Launch    : Enabled
Results Reported : Summary
Local (client)   : NIC 0x13 PID 0 VNI 10
Remote (server)  : NIC 0x12 PID 0
------------------------------------------------------------------------
     Bytes       Sends     Min[us]     Max[us]    Mean[us]  StdDev[us]
         8         100        1.60        2.65        1.64        0.13
------------------------------------------------------------------------
```

7.6.4 cxi_atomic_lat

The cxi_atomic_bw utility measures one-sided AMO latency. It can be configured to use a specific atomic operation and data type. The utility works with or without CPU offload, but enabling that feature in the NIC is left to the user.

Usage

```
Usage:
  cxi_atomic_lat [-d DEV] [-p PORT]
                         Start a server and wait for connection
  cxi_atomic_lat ADDR [OPTIONS]
                         Connect to server with address ADDR

Options:
  -d, --device=DEV       Device name (default "cxi0")
  -p, --port=PORT        The port to listen on/connect to (default 49194)
  -n, --iters=ITERS      Number of iterations to run (default 100)
  -D, --duration=SEC     Run for the specified number of seconds
      --warmup=WARMUP    Number of warmup iterations to run (default 10)
      --latency-gap=USEC Number of microseconds to wait between each
                         iteration (default 1000)
  -A, --atomic-op        The atomic operation to use (default SUM)
  -C, --cswap-op         The CSWAP operation to use (default EQ)
  -T, --atomic-type      The atomic type to use (default UINT64)
      --fetching         Use fetching AMOs
      --matching         Use matching list entries at the target
      --unrestricted     Use unrestricted AMOs
      --no-idc           Do not use Immediate Data Cmds
      --no-ll            Do not use Low-Latency command issuing
      --report-all       Report all latency measurements individually
  -h, --help             Print this help text and exit
  -V, --version          Print the version and exit

Atomic Ops:
  MIN, MAX, SUM, LOR, LAND, BOR, BAND, LXOR, BXOR, SWAP, CSWAP, AXOR
CSWAP Ops:
  EQ, NE, LE, LT, GE, GT
Atomic Types:
  INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, UINT64, FLOAT, DOUBLE,
  FLOAT_COMPLEX, DOUBLE_COMPLEX, UINT128
```

Example

This example shows a quick run using the default options.

Server

```
$ cxi_atomic_lat
Listening on port 49194 for client to connect...
------------------------------------------------------------------------
    CXI Atomic Memory Operation Latency Test
Device           : cxi0
Test Type        : Iteration
Iterations       : 100
Warmup Iters     : 10
Inter-Iter Gap   : 1000 microseconds
Atomic Op        : NON-FETCHING SUM
Atomic Type      : UINT64
IDC              : Enabled
Matching LEs     : Disabled
Restricted       : Enabled
LL Cmd Launch    : Enabled
Results Reported : Summary
Local (server)   : NIC 0x12 PID 0 VNI 10
Remote (client)  : NIC 0x13 PID 0
------------------------------------------------------------------------
See client for results.
------------------------------------------------------------------------
```

Client

```
$ cxi_atomic_lat cxi-nid0
------------------------------------------------------------------------
    CXI Atomic Memory Operation Latency Test
```

```
Device           : cxi0
Test Type        : Iteration
Iterations       : 100
Warmup Iters     : 10
Inter-Iter Gap   : 1000 microseconds
Atomic Op        : NON-FETCHING SUM
Atomic Type      : UINT64
IDC              : Enabled
Matching LEs     : Disabled
Restricted       : Enabled
LL Cmd Launch    : Enabled
Results Reported : Summary
Local (client)   : NIC 0x13 PID 0 VNI 10
Remote (server)  : NIC 0x12 PID 0
------------------------------------------------------------------------
AMO Size[B]         Ops     Min[us]     Max[us]    Mean[us]  StdDev[us]
          8         100        2.48        2.63        2.50        0.06
------------------------------------------------------------------------
```

7.7 Thermal Diagnostic

The cxi_heatsink_check utility is a thermal diagnostic intended to validate that heat is being dissipated properly. It stresses the ASIC by generating a large amount of small RDMA writes.

7.7.1 Node Configuration

When testing a single-NIC device, only one instance of cxi_heatsink_check is needed. However, when testing a dual-NIC device, two instances of the diagnostic must run simultaneously, each targeting one of the two NICs. Dual-NIC pairs can be determined by using cxi_stat to obtain NIC serial numbers. NICs of the same device will have identical serial numbers.

Note: In some compute blades, each NIC of a dual-NIC device belongs to a separate node.

To generate the most heat, the HSN link should be configured in mission-mode. Internal-loopback does not fully exercise the SerDes.

7.7.2 Running the Diagnostic

Running cxi_heatsink_check requires root privileges.

Usage

```
Monitor Cassini NIC temperature and power consumption while stressing
the chip with RDMA writes.
Requirements:
  1. The NIC must be able to initiate writes to itself, either by being
  configured in internal loopback mode, or by having a link partner that is
  configured to allow routing packets back to their source.
  2. When configured in internal loopback mode, the --no-hrp option must be
  used.
  3. When testing a dual-NIC card, the diagnostic should be run for each NIC
  concurrently or the power target will not be reached.

Options:
  -d, --device=DEV       Device name (default "cxi0")
  -t, --tx-gpu=GPU       GPU index for allocating TX buf (default no GPU)
  -r, --rx-gpu=GPU       GPU index for allocating RX buf (default no GPU)
  -g, --gpu-type=TYPE    GPU type (AMD or NVIDIA) (default type determined
                         by discovered GPU files on system)
  -P, --procs=PROCS      Number of write-generating processes
                         (default 1/4 of available processors)
  -c, --cpu-list=LIST    Processors to use when assigning affinities to
                         write-generating processes (default assigned based
                         on device number and socket count)
  -D, --duration=SEC     Run for the specified number of seconds (default 600)
  -i, --interval=INT     Interval in seconds to check and print sensor
                         readings (default 10)
  -s, --size=SIZE        RDMA Write size to use (default 512)
                         The maximum size is 262144
  -l, --list-size=LSIZE  Number of writes per iteration, all pushed to
                         the Tx CQ prior to initiating xfer (default 256)
      --no-hrp           Do not use High Rate Puts for sizes <= 2048 bytes
      --no-idc           Do not use Immediate Data Cmds for high rate put
                         sizes <= 224 bytes and matching put sizes <= 192 bytes
      --no-ple           Append a single use-once list entry for every write
                         Note: Combining this option with large LSIZE and PROCS
                         values may results in NO_SPACE errors
  -h, --help             Print this help text and exit
  -V, --version          Print the version and exit
```

Example of a Successful Run

In this example, the cxi_heatsink_check diagnostic utility successfully executed and all tests passed as expected.

```
# cxi_heatsink_check -D 60
-------------------------------------------------------------
    CXI Heatsink Test
Device          : cxi0
TX Mem          : System
RX Mem          : System
Duration        : 60 seconds
Sensor Interval : 10 seconds
TX/RX Processes : 32
Processor List  : 1-31,0
RDMA Write Size : 512
List Size       : 250
HRP             : On
IDC             : On
Persistent LEs  : On
Local Address   : NIC 0x13
Board Type      : Dual-NIC
-------------------------------------------------------------
Time[s]  Rate[GB/s]  VDD[W]  AVDD[W]  ASIC_0[°C]  ASIC_1[°C]
     10       21.64      18        6          56          56
     20       22.01      18        6          57          57
     30       22.01      18        6          57          57
     40       22.01      18        6          57          57
     50       22.01      18        7          58          58
-------------------------------------------------------------
Cassini 0 Temperature (ASIC_0) under 85 °C:  58 °C       PASS
Cassini 1 Temperature (ASIC_1) under 85 °C:  58 °C       PASS
0.85V S0 Power (VDD) reached 17 W:           18 W        PASS
0.9V S0 Power (AVDD) reached 6 W:            7 W         PASS
Average BW over 19 GB/s:                     21.94 GB/s  PASS
```

Example Power Draw and Bandwidth Target Failures

This example shows two failures. The first failure (**0.85V S0 Power (VDD) reached 17 W**) indicates that the test did not reach the minimum power usage target.This means the test could not validate that the heatsink was functioning as expected due to a low power draw.

The second failure (**Average BW over 19 GB/s**) indicates that the target bandwidth was not reached. This correlates to the insufficient power draw. If full bandwidth is not reached, the power draw target cannot be reached.

```
# cxi_heatsink_check -D 60 -P 1
-------------------------------------------------------------
    CXI Heatsink Test
Device          : cxi0
TX Mem          : System
RX Mem          : System
Duration        : 60 seconds
Sensor Interval : 10 seconds
TX/RX Processes : 1
Processor List  : 1-31,0
RDMA Write Size : 512
List Size       : 250
HRP             : On
IDC             : On
Persistent LEs  : On
Local Address   : NIC 0x12
Board Type      : Dual-NIC
-------------------------------------------------------------
Time[s]  Rate[GB/s]  VDD[W]  AVDD[W]  ASIC_0[°C]  ASIC_1[°C]
     10       10.30      15        6          55          56
     20       10.30      15        6          55          56
     30       10.31      16        6          56          56
     40       10.32      16        6          56          56
     50       10.33      16        6          56          56
-------------------------------------------------------------
Cassini 0 Temperature (ASIC_0) under 85 °C:  56 °C       PASS
Cassini 1 Temperature (ASIC_1) under 85 °C:  56 °C       PASS
0.85V S0 Power (VDD) reached 17 W:           16 W        FAIL
0.9V S0 Power (AVDD) reached 6 W:            6 W         PASS
Average BW over 19 GB/s:                     10.31 GB/s  FAIL
```

Example High Temperature Failure

This example shows a single failure (**Cassini 0 Temperature (ASIC_0) under 85°C**). The Cassini temperature rose higher than expected, which indicates that heat is not being dissipated properly. In this case the Cassini NIC card should be inspected to ensure that the heatsink is properly installed and that the cabinet has adequate cooling. If the problem persists and is local to only one card within the cabinet, then the Cassini NIC card should be replaced.

```
# cxi_heatsink_check -D 12 -i 2
-------------------------------------------------------------
    CXI Heatsink Test
Device          : cxi0
TX Mem          : System
RX Mem          : System
Duration        : 12 seconds
Sensor Interval : 2 seconds
TX/RX Processes : 32
Processor List  : 1-31,0
RDMA Write Size : 512
List Size       : 250
HRP             : On
IDC             : On
Persistent LEs  : On
Local Address   : NIC 0x12
Board Type      : Dual-NIC
-------------------------------------------------------------
Time[s]  Rate[GB/s]  VDD[W]  AVDD[W]  ASIC_0[°C]  ASIC_1[°C]
      2       21.13      18        6          81          79
      4       21.38      18        6          82          80
      6       21.55      18        6          83          81
      8       21.75      18        6          84          82
     10       21.93      18        6          85          83
-------------------------------------------------------------
Cassini 0 Temperature (ASIC_0) under 85 °C:  85 °C       FAIL
Cassini 1 Temperature (ASIC_1) under 85 °C:  83 °C       PASS
0.85V S0 Power (VDD) reached 17 W:           18 W        PASS
0.9V S0 Power (AVDD) reached 6 W:            6 W         PASS
Average BW over 19 GB/s:                     21.55 GB/s  PASS
```

Example Early Stop

This example illustrates an early exit due to the Cassini 0 Temperature continuing to climb into an unsafe range. In this case the test is stopped immediately. If the temperature was allowed to continue to climb, the Cassini NIC card would execute an Emergency Power Off (EPO). The Cassini NIC card should be inspected to ensure that the heatsink is properly installed and that the cabinet has adequate cooling. If the problem persists and is local to only one card within the cabinet, then the Cassini NIC card should be replaced.

```
# cxi_heatsink_check -D 12 -i 1
-------------------------------------------------------------
    CXI Heatsink Test
Device          : cxi0
TX Mem          : System
RX Mem          : System
Duration        : 12 seconds
Sensor Interval : 1 seconds
TX/RX Processes : 32
Processor List  : 1-31,0
RDMA Write Size : 512
List Size       : 250
HRP             : On
IDC             : On
Persistent LEs  : On
Local Address   : NIC 0x12
Board Type      : Dual-NIC
-------------------------------------------------------------
Time[s]  Rate[GB/s]  VDD[W]  AVDD[W]  ASIC_0[°C]  ASIC_1[°C]
      1       20.96      18        6          81          79
      2       21.11      18        6          82          80
      3       21.30      18        6          83          81
      4       21.52      18        6          84          82
      5       21.67      18        6          85          83
      6       21.75      18        6          86          84
      7       21.84      18        6          87          85
      8       21.92      18        6          88          86
      9       21.93      18        6          89          87
     10       21.98      18        6          90          88

Error! Cassini 0 Temperature has exceeded safe range. Exiting early.
-------------------------------------------------------------
Cassini 0 Temperature (ASIC_0) under 85 °C:  90 °C       FAIL
Cassini 1 Temperature (ASIC_1) under 85 °C:  88 °C       FAIL
0.85V S0 Power (VDD) reached 17 W:           18 W        PASS
0.9V S0 Power (AVDD) reached 6 W:            6 W         PASS
Average BW over 19 GB/s:                     21.60 GB/s  PASS
```

7.8 Troubleshooting

7.8.1 Cannot Reach Server

The following error indicates that the client could not reach the remote host. Check that the hostname or address supplied to the client is valid. Verify that the client and server hosts can ping each other.

```
getaddrinfo() failed: Name or service not known
Failed to init control messaging: Input/output error
```

7.8.2 No Server Running at Remote Host

The following error indicates that the client could reach the remote host but found no server listening on the given TCP port (49194 by default). Verify the hostname or address is correct. Verify that the server is running and the server and client are using the same value for the `--port` option.

Note: When launching a server and client from an automated process it can be helpful to add a short wait after launching the server before launching the client.

```
Failed to connect to cxi-nid1: Connection refused
Failed to init control messaging: Connection refused
```

7.8.3 TCP Port Already in Use

The following error indicates that the server's given TCP port is already being used. Often this means another server is already running. If not, or if multiple simultaneous servers are desired, the `--port` option allows selecting a different port.

```
bind() failed: Address already in use
Failed to init control messaging: Address already in use
```

7.8.4 Incorrect Program or Version

The following error indicates that different CXI diagnostics were used for client and server.

```
Client and server program names do not match!
Failed to exchange client/server config: Invalid argument
```

This error indicates that the client and server program versions are incompatible. Major and minor versions must match, while the revision number is allowed to differ.

```
Client and server program versions do not match!
cxi_write_bw: 1.3.0
Failed to exchange client/server config: Invalid argument
```

7.8.5 Unexpected Event Type

The following errors occur when the client sends a packet and does not receive a response.

Server

```
recv() failed: Connection reset by peer
Post-run handshake failed: Connection reset by peer
```

Client

```
Unexpected event type: ACK (9) Expected TRIGGERED_OP
event RC != RC_OK: CANCELED
Failed to get initiator ACK TRIGGERED_OP event: No message of desired type
```

When High rate puts (HRP) are used, the RC might instead be HRP_RSP_DISCARD. This means the packet made it as far as the switch.

```
Unexpected event type: ACK (9) Expected TRIGGERED_OP
event RC != RC_OK: HRP_RSP_DISCARD
Failed to get initiator ACK TRIGGERED_OP event: No message of desired type
```

If the RC is PKTBUF_ERROR, the node logs should be checked for pfc_fifo_oflw errors. This means the switch QoS config has not been applied correctly.

```
Unexpected event type: ACK (9) Expected TRIGGERED_OP
event RC != RC_OK: PKTBUF_ERROR
Failed to get initiator ACK TRIGGERED_OP event: No message of desired type
```

```
[ 1451.997298] cxi_core 0000:21:00.0: HNI error: pfc_fifo_oflw (40)
[ 1452.002035] cxi_core 0000:21:00.0:   pfc_fifo_oflw_cntr: 218
[ 1452.005821] cxi_core 0000:21:00.0: IXE error: pbuf_rd_err (48)
[ 1452.009456] cxi_core 0000:21:00.0:   pbuf_rd_errors: 105
```

These errors indicate that the link is not up or something is misconfigured. Verify the link state on both the NIC and the switch. Verify that the NIC's AMA has been applied and is correct. Verify that the switch routing config and QoS config have been applied and are correct.

7.8.6 High Rate Puts and Internal-Loopack

High rate puts are acknowledged by the Rosetta switch, not the target NIC, so they do not work when the NIC is configured in internal-loopback mode. The `--no-hrp` option can be used to disable high rate puts.

7.8.7 GPU Libraries

When using GPU buffers, the diagnostics attempt to dynamically link the appropriate runtime library, HIP for AMD or CUDA for NVIDIA. The following error occurs if the library has not been installed.

```
Get device count request failed
Failed to init GPU lib: Operation not permitted
```

If both HIP and CUDA are installed and NVIDIA GPUs are being used, HIP is used by default. In that case this error indicates that the HIP library has not been compiled for NVIDIA. Adding -g NVIDIA to the command forces use of the CUDA library.

7.8.8 Low Bandwidth

In general, the bandwidth utilities and cxi_heatsink_check should reach full bandwidth using their default options. However, there are cases where this may not be true. The following may result in improved performance:

- Pin processes to specific cores. This can be done to avoid core 0 and to assign the server and client to separate cores. When using a multi-socket node, it is important to use the socket nearest the NIC under test. cxi_heatsink_check attempts to intelligently set core affinities, but this can be overridden with the `--cpu-list` option.
- Use larger values for `--size`.
- Use larger or smaller values for `--list-size` (called `--num-xfers` in cxi_gpu_bw_loopback). This option specifies the number of transactions that are queued up before telling the NIC to proceed.
- Use larger or smaller values for `--procs` with cxi_heatsink_check. This option specifies the number of write-generating processes that are created.
- Use GPU memory for TX and/or RX buffers.

7.8.9 List Entry Exhaustion

The Cassini NIC has 4 processing engines, each of which has access to 16,384 list entries. When cxi_heatsink_check is run with the `--no-ple` option, every write is preceded by appending a single, use-once list entry to receive the write. If both the number of processes and the list size are set to large values, this can result in exceeding the available number of list entries.

```
event RC != RC_OK: NO_SPACE
Failed to get target LINK event: No message of desired type
```

If this error is seen, try reducing the list size with `--list-size` and the number of processes with `--procs`, or try running without the `--no-ple` option.

7.8.10 Sync Timeout

cxi_heatsink_check spawns multiple processes to generate stress traffic. It synchronizes these processes once before they begin generating traffic and again when they stop.

```
proc0 wait failed: Connection timed out
```

This error message is accompanied by "sync failed" errors from each process. In the absence of other errors, a sync timeout likely indicates that the values of `--procs`, `--size`, and `--list-size` may be too large. The diagnostic generally isn't meant to be run with high values for all three of these arguments. Each of them increases the amount of time needed to initialize the write data buffers to random values, as well as the time needed for the traffic generating processes to quiesce. Try lowering one or more of them.

7.8.11 Semaphore File Already Exists

cxi_heatsink_check uses two semaphore files to synchronize processes. These are named based on the CXI device ID. For example, with cxi0 the files would be /dev/shm/sem.cxi_heatsink_p_cxi0 and /dev/shm/sem.cxi_heatsink_c_cxi0. If either of these already exists, cxi_heatsink_check cannot run.

```
sem_open(/cxi_heatsink_p_cxi0): File exists
```

Before deleting the semaphore files, check to see if cxi_heatsink_check is already running. Wait for any running processes finish, or try to terminate them:

```
pkill cxi_heatsink_check
```

A single SIGINT or SIGTERM signal should be enough to break the processes out of their loops and cause them to clean up all Cassini resources that were allocated through the driver. If they do not finish running within several seconds, they can be killed with a second SIGTERM. Please note that killing the processes with SIGKILL or a second SIGTERM may result in degraded Cassini performance until the next node reboot.

8 Network Diagnostics

8.1 Overview

Several diagnostics have been developed to measure performance or verify link health over the entirety or a subsection of the high speed network.

The dgnettest diagnostic measures various aspects of network performance for a given set of nodes. This includes loopback bandwidth, point-to-point latency, bisectional bandwidth, and all-to-all bandwidth. The dgnettest_run.sh script can be used to execute a common set of test configurations at once. It includes an option to broadcast a copy of the binary to each node. The diagnostic and the run script both run from the user access node (UAN). They use Slurm as a workload manager, and they use MPI for node-to-node communication.

The cxibwcheck.sh and bwcheck.sh diagnostics measure loopback bandwidth for a given set of nodes. cxibwcheck.sh uses a libcxi diagnostic, cxi_write_bw. bwcheck.sh uses a Perftest diagnostic, ib_write_bw. Neither of these rely on MPI. They can be run on a UAN, in which case they use pdsh to execute on the nodes. They also support being executed through Slurm. cxibwcheck.sh is only valid for use with Slingshot NICs, and bwcheck.sh is only valid for use with Infiniband NICs.

The cxiberstat.sh diagnostic measures NIC link corrected and uncorrected bit error rates (BERs) for a given set of nodes. It uses a libcxi diagnostic, cxi_stat to measure the rates. It should be run on a UAN. By default it will interface with Slurm, but there is an option to use pdsh instead.

The cray-diags-fabric RPM is included in the slingshot-host-software tarball in the Slingshot release package.

The cray-diags-fabric package can be installed on both compute and non-compute nodes. Binaries and scripts are placed in /usr/local/diag/bin. dgnettest and cxibwcheck.sh include man pages, which are placed in /usr/local/diag/man/man1.

8.1.1 Minimum Setup

UAN:

- Slurm must be installed (for dgnettest).

Nodes:

- The Cray MPICH, libfabric, and craype libraries must be installed (for dgnettest).
- The cray-libcxi-utils package must be installed (for cxibwcheck.sh and cxiberstat.sh).
- If Cassini NICs are used, the CXI drivers must be installed.
- If Cassini NICs are used, the retry handler service must be running for each NIC (cxi_rh@cxi0, cxi_rh@cxi1, etc.).
- The links must be up.
- The links must have valid IPv4 addresses configured.
- If Cassini NICs are used, the links must have valid algorithmic MAC addresses (AMAs) configured.
- Ping all-to-all connectivity should be verified.
- If use of GPU memory is desired, the appropriate runtime library must be installed (for cxibwcheck.sh).

Fabric:

- The links must be up.

- The switches must have routing and QoS configuration applied.

8.1.2 Theory of Operation

The network diagnostics are useful tools for validating overall system health. This should be done using a methodical approach, testing individual NIC links first and then growing in scope until the entire system is being tested. dgnettest_run.sh offers one version of this methodical testing strategy, with the benefit of being automated by a single script. However, more can be accomplished by running the diagnostics manually and investigating each issue before growing the testing scope.

The following series of steps is a useful place to start. After each step, failures should be investigated and fixed, or the affected nodes should be drained and excluded from further testing.

- Run cxiberstat.sh.
- Run cxibwcheck.sh.
- Run the dgnettest loopback test.
- Run the dgnettest bisection test for nodes attached to each switch. This is done by specifying a set size of 16 with `-s 16` (or 32 in the case of Class 1 topologies).
- Run the dgnettest bisection test for nodes attached to each group. This is done by specifying a set size of 512 with `-s 512`.
- Run the dgnettest bisection test across the whole system. This is done by specifying a set size that matches the total number of nodes in the system.

8.2 dgnettest

The network diagnostic dgnettest is comprised of four individual tests. It interfaces with Slurm to run multiple tests in parallel, setting up MPI communicators for each blade, group, or other set of nodes. It can run on systems of any size. It uses enough processes per node to get good performance while ensuring that memory use remains low.

Loopback Bandwidth Test: The loopback test measures MPI bandwidth with each NIC set to communicate through the connected Rosetta switch back to itself with local memory sharing turned off. It reports bandwidth for each node and highlights nodes that measured outside of a set threshold. This test runs for each NIC on a node in serial, but for each node in parallel.

Latency Test: The latency test measures MPI point-to-point latency from one node to itself and to every other node. The test measures round-trip latency and halves this to estimate one-way latency. It reports a summary of latency statistics along with a histogram to illustrate the measured distribution.

Fabric Bisection Bandwidth Test: The bisection bandwidth test divides the nodes in two and measures bandwidth between the halves. In general the user selects the worst-case cut, but in a dragonfly network (or fat-tree) of reasonable size it is all pretty much the same. The bisection bandwidth is usually limited by the bandwidth of the global links.

Fabric All-to-All Test: The all-to-all diagnostic measures performance of all-to-all communication for sets of nodes corresponding to the physical structure of a system: blades, chassis, groups, and the whole system. The test is designed to run on as many nodes as are available, reporting variation in performance over sets of nodes of a given size. For example, to run 512 instances of a blade level test on 2048 nodes and report variation between them the set size would be 4.
The diagnostic generates a high network load. In particular it stresses the PCIe interfaces that connect each node to its NICs. Poor performance on this test correlates well to high rates of PCIe errors. It is intended that the test be executed on all nodes for a period of ten minutes. This is sufficient to detect nodes with rates of PCIe errors that impact application performance.
The diagnostic uses MPI_Alltoall with the DMAPP optimizations enabled. As such it is representative of a real application.
The diagnostic does not require dedicated access to the whole system. It can be run on a subset of nodes allocated by the batch system. The impact on performance of other applications is low if all nodes in an electrical group are allocated to a test.

8.2.1 dgnettest_run.sh

While dgnettest can be run as a standalone test for troubleshooting specific issues, it would need to be run multiple times to get a full picture of the state of a system. The script dgnettest_run.sh makes this easier by executing dgnettest in a number of configurations. There are four test configurations that dgnettest_run.sh runs.

Loopback: This configuration executes the loopback bandwidth test.

Latency: This configuration executes the latency test. By default, it is excluded from the test list. It can be included by providing dgnettest_run.sh with test list arguments that include `latency`. Note that specifying the test list requires supplying a set size value with `-s` as well.

Switch: This configuration executes the bisect bandwidth and all2all tests using the dgnettest option `-S` to display statistics. The set size is the number of edge ports on a switch, which is chosen based on the network class of the system, optionally specified with the dgnettest_run.sh option `-c`.

Group: This configuration executes the bisect bandwidth and all2all tests using the dgnettest option `-S` to display statistics. The set size is 512, which is the number of nodes in a high-density cabinet.

8.2.2 Examples

8.2.2.1 Typical run on dual-NIC system

```
$ dgnettest_run.sh -n nid[000013-000028]
Running loopback tests on NIC 0 over : nid[000013-000028]
Using NIC 0
nprocs=128 sets=1 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
loopback     1   131072    10307    10503    10664       94     0.9%
loopback     2   131072    10497    10654    10777       83     0.8%
loopback     3   131072    10497    10655    10778       83     0.8%
Final loopback bandwidth stats
loopback     3   131072    10497    10655    10778       82     0.8%
Per node bandwidth stats
nid000013    3   131072    10733    10733    10734        1     0.0%
nid000014    3   131072    10671    10671    10672        1     0.0%
nid000015    3   131072    10688    10688    10689        1     0.0%
nid000016    3   131072    10777    10777    10778        0     0.0%
nid000017    3   131072    10757    10757    10758        0     0.0%
nid000018    3   131072    10592    10593    10593        1     0.0%
nid000019    3   131072    10633    10634    10634        1     0.0%
nid000020    3   131072    10643    10644    10644        0     0.0%
nid000021    3   131072    10638    10638    10638        0     0.0%
nid000022    3   131072    10703    10706    10709        4     0.0%
nid000023    3   131072    10756    10757    10757        0     0.0%
nid000024    3   131072    10613    10614    10615        1     0.0%
nid000025    3   131072    10612    10613    10613        0     0.0%
nid000026    3   131072    10498    10498    10498        0     0.0%
nid000027    3   131072    10654    10654    10655        0     0.0%
nid000028    3   131072    10497    10497    10497        1     0.0%
dgnettest has PASSED

Running loopback tests on NIC 1 over : nid[000013-000028]
Using NIC 1
nprocs=128 sets=1 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes       Min      Mean       Max      Dev       CV
loopback     1   131072      6065      6196      6883      189     3.0%
loopback     2   131072      6141      6235      7013      211     3.4%
loopback     3   131072      6134      6232      6842      169     2.7%
loopback     4   131072      6144      6265      7158      248     4.0%
loopback     5   131072      6144      6258      7322      285     4.6%
Final loopback bandwidth stats
loopback     5   131072      6134      6248      7322      227     3.6%
Per node bandwidth stats
nid000013    5   131072      6147      6167      6192       21     0.3%
nid000014    5   131072      6164      6178      6195       15     0.2%
nid000015    5   131072      6143      6154      6173       14     0.2%
nid000016    5   131072      6161      6185      6204       21     0.3%
nid000017    5   131072      6134      6161      6202       31     0.5%
nid000018    5   131072      6185      6190      6193        4     0.1%
nid000019    5   131072      6144      6150      6155        4     0.1%
nid000020    5   131072      6165      6173      6180        7     0.1%
nid000021    5   131072      6174      6178      6182        4     0.1%
nid000022    5   131072      6166      6186      6237       34     0.6%
nid000023    5   131072      6220      6286      6396       77     1.2%
nid000024    5   131072      6842      6984      7022       76     1.2%
nid000025    5   131072      6167      6215      6315       69     1.1%
nid000026    5   131072      6192      6193      6197        2     0.0%
nid000027    5   131072      6165      6179      6202       17     0.3%
nid000028    5   131072      6242      6281      6313       30     0.5%
dgnettest has PASSED

Running switch tests on NIC 0 over : nid[000013-000028]
Using NIC 0
nprocs=128 sets=2 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes       Min      Mean       Max      Dev       CV
all2all      1   131072      8718      9805     10892     1537    15.7%
all2all      2   131072     10987     11747     12507     1075     9.2%
all2all      3   131072     10652     11217     11782      799     7.1%
Test      Iter    Bytes       Min      Mean       Max      Dev       CV
```

```
bisect       1   131072    18117    18144    18170       38      0.2%
bisect       2   131072    18146    18177    18207       42      0.2%
bisect       3   131072    18152    18182    18212       42      0.2%
dgnettest has PASSED

Running switch tests on NIC 1 over : nid[000013-000028]
Using NIC 1
nprocs=128 sets=2 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
all2all      1   131072     5514     5762     6010      351     6.1%
all2all      2   131072     4857     5128     5398      382     7.5%
all2all      3   131072     6189     6437     6685      351     5.4%
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
bisect       1   131072     6184     6188     6192        6     0.1%
bisect       2   131072     5981     6030     6080       70     1.2%
bisect       3   131072     5769     5843     5917      104     1.8%
bisect       4   131072     5949     6026     6103      109     1.8%
bisect       5   131072     6033     6111     6190      111     1.8%
dgnettest has PASSED

Running group tests on NIC 0 over : nid[000013-000028]
Using NIC 0
nprocs=128 sets=1 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
all2all      1   131072    14073    14073    14073        0     0.0%
all2all      2   131072    12926    12926    12926        0     0.0%
all2all      3   131072    12558    12558    12558        0     0.0%
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
bisect       1   131072    18087    18087    18087        0     0.0%
bisect       2   131072    18197    18197    18197        0     0.0%
bisect       3   131072    18202    18202    18202        0     0.0%
dgnettest has PASSED

Running group tests on NIC 1 over : nid[000013-000028]
Using NIC 1
nprocs=128 sets=1 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
all2all      1   131072    11920    11920    11920        0     0.0%
all2all      2   131072    10832    10832    10832        0     0.0%
all2all      3   131072     9820     9820     9820        0     0.0%
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
bisect       1   131072    12284    12284    12284        0     0.0%
bisect       2   131072     6005     6005     6005        0     0.0%
bisect       3   131072     5979     5979     5979        0     0.0%
bisect       4   131072     5640     5640     5640        0     0.0%
dgnettest has PASSED
```

8.2.2.2 Running with a subset of tests

A list of tests to run can be passed to dgnettest_run.sh. When a test list is provided, the set-size must be specified with `-s` or the test list is ignored.

```
$ dgnettest_run.sh -n nid[000022-000028] -s 16 latency bisect
Running tests over : nid[000022-000028]
nprocs=56 sets=2 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test       Count    Min   Mean    Max    Dev     CV
latency       25   1.86   1.88   1.90   0.01    0.7%
Test      Iter    Bytes  Set      Base  Ranks    Reps     Secs  BW/proc  BW/node
bisect       1   131072    1 000.01.00     32    2791  0.00287  1391.85 11134.77
bisect       1   131072    3 000.03.00     24    2791  0.00155  1936.26 15490.05
bisect       2   131072    1 000.01.00     32    3682  0.00289  1383.68 11069.44
bisect       2   131072    3 000.03.00     24    3682  0.00155  1929.98 15439.84
bisect       3   131072    1 000.01.00     32    2064  0.00418   958.00  7664.01
bisect       3   131072    3 000.03.00     24    2064  0.00155  1933.02 15464.16
bisect       4   131072    1 000.01.00     32    2066  0.00337  1185.33  9482.63
bisect       4   131072    3 000.03.00     24    2066  0.00155  1933.75 15470.04
dgnettest has PASSED
```

8.2.2.3 Running concurrently on node NICs

The loopback test is always executed separately for each NIC number.

```
$ dgnettest_run.sh -n nid[001128-001163] -C
Running loopback tests on NIC 0 over : nid[001128-001163]
nprocs=288 sets=1 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
loopback     1   131072    11112    11184    11258       35      0.3%
loopback     2   131072    11181    11239    11286       27      0.2%
loopback     3   131072    11181    11240    11286       27      0.2%
Final loopback bandwidth stats
loopback     3   131072    11181    11240    11286       27      0.2%
Per node bandwidth stats
nid001128    3   131072    11232    11233    11234        1      0.0%
```

```
nid001129    3   131072    11259    11259    11260        1      0.0%
<lines omitted for brevity>
nid001162    3   131072    11226    11227    11227        1      0.0%
nid001163    3   131072    11230    11230    11230        0      0.0%
dgnettest has PASSED

Running loopback tests on NIC 1 over : nid[001128-001163]
nprocs=288 sets=1 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
loopback     1   131072     6157     6695     8457      649      9.7%
loopback     2   131072     6018     6266     8037      434      6.9%
loopback     3   131072     6026     6264     8051      444      7.1%
loopback     4   131072     6006     6266     8011      441      7.0%
Final loopback bandwidth stats
loopback     4   131072     6006     6265     8051      436      7.0%
Per node bandwidth stats
nid001128    4   131072     6265     6294     6341       42      0.7%
nid001129    4   131072     6152     6175     6209       30      0.5%
<lines omitted for brevity>
nid001162    4   131072     6013     6035     6066       28      0.5%
nid001163    4   131072     6019     6041     6058       20      0.3%
dgnettest has PASSED

Running group tests over : nid[001128-001163]
nprocs=288 sets=1 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
all2all      1   131072     9594     9594     9594        0      0.0%
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
bisect       1   131072    37902    37902    37902        0      0.0%
bisect       2   131072    40094    40094    40094        0      0.0%
bisect       3   131072    40099    40099    40099        0      0.0%
dgnettest has PASSED

Running switch tests over : nid[001128-001163]
nprocs=288 sets=4 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
all2all      1   131072    36344    36443    36542      140      0.4%
all2all      2   131072    19979    20779    21580     1132      5.4%
all2all      3   131072    21386    21925    22464      762      3.5%
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
bisect       1   131072    33610    35609    37498     1900      5.3%
bisect       2   131072    33786    38274    42493     4812     12.6%
bisect       3   131072    33789    38269    42517     4831     12.6%
dgnettest has PASSED
```

8.2.2.4 Custom NIC mapping

By default, NICs are divided and mapped evenly across the NUMA nodes. This can be overridden with a custom mapping of threads to NICs.

If dgnettest is being executed directly, several MPICH environment variables must be set. The dgnettest_run.sh script does this automatically when a custom mapping is provided. For example, two NICs can be split so NIC 0 is mapped to the even ranks and NIC 1 is mapped to the odd ranks.

```
$ export MPICH_OFI_NUM_NICS=2
$ export MPICH_OFI_NIC_POLICY="USER"
$ export MPICH_OFI_NIC_MAPPING="0:0,2,4,6;1:1,3,5,7"
```

The mapping should be provided to dgnettest with the `-N` option or to dgnettest_run.sh with the `-C` option.

Note: Results are not guaranteed to be valid when using a custom mapping, so be sure to validate any errors with a re-run of dgnettest with the default settings.

```
$ dgnettest_run.sh -n nid[000022-000028] -C "0:0,2,4,6;1:1,3,5,7"
Running loopback tests on NIC 0 over : nid[000022-000028]
Using NIC 0
nprocs=56 sets=1 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
loopback     1   131072     3616     3636     3655       12      0.3%
loopback     2   131072     3644     3650     3655        4      0.1%
loopback     3   131072     3644     3650     3656        5      0.1%
Final loopback bandwidth stats
loopback     3   131072     3644     3650     3656        4      0.1%
Per node bandwidth stats
nid000022    3   131072     3645     3646     3646        0      0.0%
nid000023    3   131072     3651     3652     3652        1      0.0%
nid000024    3   131072     3650     3650     3650        0      0.0%
nid000025    3   131072     3655     3656     3656        0      0.0%
nid000026    3   131072     3644     3644     3644        0      0.0%
nid000027    3   131072     3655     3655     3655        0      0.0%
nid000028    3   131072     3647     3647     3647        0      0.0%
dgnettest has PASSED
```

```
Running loopback tests on NIC 1 over : nid[000022-000028]
Using NIC 1
nprocs=56 sets=1 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min      Mean      Max      Dev       CV
loopback     1   131072     4082     5471     8073     1735     31.7%
loopback     2   131072     4181     6877     8943     2037     29.6%
loopback     3   131072     4180     7597     8934     1659     21.8%
Final loopback bandwidth stats
loopback     3   131072     4180     7237     8943     1824     25.2%
Per node bandwidth stats
nid000022    3   131072     8282     8358     8433      107      1.3%   result is high
nid000023    3   131072     4186     6132     8077     2751     44.9%    result is low
nid000024    3   131072     4180     4181     4181        0      0.0%    result is low
nid000025    3   131072     6814     7147     7479      470      6.6%
nid000026    3   131072     6308     7571     8835     1787     23.6%
nid000027    3   131072     8934     8939     8943        6      0.1%   result is high
nid000028    3   131072     8058     8335     8611      391      4.7%   result is high
dgnettest has FAILED
srun: error: nid000022: task 0: Exited with exit code 1

Running group tests over : nid[000022-000028]
Using all NICs
nprocs=56 sets=1 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min      Mean      Max      Dev       CV
all2all      1   131072    13432    13432    13432        0      0.0%
all2all      2   131072    16304    16304    16304        0      0.0%
all2all      3   131072    16969    16969    16969        0      0.0%
Test      Iter    Bytes      Min      Mean      Max      Dev       CV
bisect       1   131072    14623    14623    14623        0      0.0%
bisect       2   131072     8683     8683     8683        0      0.0%
bisect       3   131072    14009    14009    14009        0      0.0%
dgnettest has PASSED

Running switch tests over : nid[000022-000028]
Using all NICs
nprocs=56 sets=2 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min      Mean      Max      Dev       CV
all2all      1   131072    14024    14024    14024        0      0.0%
all2all      2   131072    25106    25106    25106        0      0.0%
all2all      3   131072    18942    18942    18942        0      0.0%
all2all      4   131072    19366    19366    19366        0      0.0%
all2all   test bandwidth low  11055 MB/s for set 0 location 000.03.00 3 nodes: nid[000022,000025-000026]
Test      Iter    Bytes      Min      Mean      Max      Dev       CV
bisect       1   131072     8127    10795    13462     3772     34.9%
bisect       2   131072     8114     8263     8411      210      2.5%
bisect       3   131072     8084     9114    10144     1456     16.0%
bisect    test bandwidth low   8084 MB/s for set 0 location 000.03.00 3 nodes: nid[000022,000025-000026]
dgnettest has FAILED
srun: error: nid000022: task 0: Exited with exit code 1
```

8.2.3 Troubleshooting

8.2.3.1 Test Failures

Test failures indicate NIC or network issues. A failure can often be isolated by running multiple tests using different node configurations. Once isolated, error flags and counters should be checked for the NICs and switches affected. Cabling and link stability should be verified. Algorithmic MAC addresses, switch routing, and switch and NIC QoS settings should be verified.

8.2.3.1.1 Loopback Test

If a node falls outside of a set threshold compared to the mean, a warning is displayed. There can be warnings for both too high and too low results, but only low results cause the test to fail. The inclusion of a too high warning is due to the possibility of extreme outliers skewing the results for other nodes. Begin by looking at the worst offenders and retest once they have been fixed or without them included in the test. The threshold value is set to 90% by default (acceptable values fall within 10% above or below the mean). This can be adjusted with the `-T` option.

```
Running loopback tests on NIC 1 over : nid[000022-000028]
Using NIC 1
nprocs=56 sets=1 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
loopback  autoreps time 0.003501 reps 4284
Test      Iter    Bytes      Min      Mean      Max      Dev       CV
loopback     1   131072     4044     6518     8156     1782     27.3%
loopback     2   131072     4235     7033     8771     1641     23.3%
loopback     3   131072     6653     7910     9145      998     12.6%
loopback     4   131072     4850     6785     9087     1796     26.5%
Final loopback bandwidth stats
loopback     4   131072     4235     7243     9145     1523     21.0%
Per node bandwidth stats
```

```
nid000022    4   131072     7900     8570     9039      596      7.0%   result is high
nid000023    4   131072     6223     6494     6823      304      4.7%    result is low
nid000024    4   131072     4235     6201     9145     2597     41.9%    result is low
nid000025    4   131072     6617     7914     8854     1161     14.7%
nid000026    4   131072     5356     6984     8206     1468     21.0%
nid000027    4   131072     6653     8150     9087     1309     16.1%   result is high
nid000028    4   131072     4850     6385     8045     1601     25.1%    result is low
dgnettest has FAILED
```

8.2.3.1.2 Latency Test

In the latency test the coefficient of variance (CV) is checked to see if it's below a threshold, which is set to 5% by default. If the CV is above this threshold, a warning is printed. In this case the latency histogram likely shows a wide range of latency values. The CV threshold can be adjusted with the `-l` option. Increasing the threshold is necessary when measuring latency for nodes that span multiple switches.

```
$ dgnettest_run.sh -n nid[001000-001009,001011-001081,001211-001219,001221-001279,001412-001499,001501-001511] -s 16 latency
Running tests over : nid[001000-001009,001011-001081,001211-001219,001221-001279,001412-001499,001501-001511]
nprocs=1984 sets=20 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test       Count    Min   Mean    Max    Dev     CV
latency     1596   1.81   2.37   2.93   0.33   13.9%     CV is high
latency histogram
latency   1.75 - 1.80     0
latency   1.80 - 1.85     66
latency   1.85 - 1.90     110
latency   1.90 - 1.95     27
latency   1.95 - 2.00     63
latency   2.00 - 2.05     123
latency   2.05 - 2.10     24
latency   2.10 - 2.15     70
latency   2.15 - 2.20     53
latency   2.20 - 2.25     59
latency   2.25 - 2.30     116
latency   2.30 - 2.35     93
latency   2.35 - 2.40     6
latency   2.40 - 2.45     116
latency   2.45 - 2.50     33
latency   > 2.50          637
dgnettest has PASSED
```

8.2.3.1.3 Bisectional and All-to-All Tests

If the bandwidth for a set of nodes falls below a set threshold compared to the mean, a warning is printed, and the test fails. The threshold value is set to 90% by default. This can be adjusted with the `-T` option.

```
Running group tests on NIC 1 over : nid[000004-000028,000033-000064]
Using NIC 1
nprocs=456 sets=2 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
all2all      1   131072     6357     6637     6918      397      6.0%
all2all      2   131072     6291     7326     8360     1463     20.0%
all2all   test bandwidth low   6291 MB/s for set 1 location 001.00.00 32 nodes: nid[000033-000064]
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
bisect       1   131072     6199     7922     9646     2438     30.8%
bisect       2   131072     5655     7360     9065     2411     32.8%
bisect       3   131072     5580     7356     9132     2511     34.1%
bisect    test bandwidth low   5580 MB/s for set 1 location 001.00.00 32 nodes: nid[000033-000064]
dgnettest has FAILED
```

8.2.3.2 Not Enough Nodes

Due to the nature of the tests, the all2all test must have at least 4 nodes present per set, and the bisect test must have at least 2 nodes present. If a smaller configuration is used, a warning is displayed.

```
Running switch tests over : nid[000001-000004]
Using NIC 0
nprocs=32 sets=2 maxbytes=131072 ppn=8 (Warning: no PMI set manually)
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
all2all      1   131072                         Not enough nodes
all2all      2   131072                         Not enough nodes
all2all      3   131072                         Not enough nodes
Test      Iter    Bytes      Min     Mean      Max      Dev       CV
bisect       1   131072    19374    19621    19868      349      1.8%
bisect       2   131072    19409    19639    19869      326      1.7%
bisect       3   131072    19409    19651    19894      343      1.7%
dgnettest has PASSED
```

8.2.3.3 PMPI_Init Error

By default, dgnettest uses the MPI selection set in Slurm on the system. If this fails with the following error, try using the `--mpi` option to select a different version of MPI. A system's supported versions can be queried with `srun --mpi=list`.

```
aborting job:
Fatal error in PMPI_Init: Other MPI error, error stack:
MPIR_Init_thread(632):
MPID_Init(286).......:  PMI2 init failed: 1
MPICH ERROR [Rank 0] [job id unknown] [Wed Aug 26 14:01:10 2020] [unknown] [nid000009] - Abort(590863) (rank 0 in comm 0): Fatal error in PMPI_Init: Other
MPIR_Init_thread(632):
MPID_Init(286).......:  PMI2 init failed: 1
```

8.2.3.4 Failed to Parse MAC

By default, dgnettest runs with an assumption that the network class of the system is class 2. The supported network classes range from 0 to 4. The following error is seen when an incorrect class type is used. The program may also hang.

```
dgnettest_run.sh -n nid[000005-000009]
Running loopback tests over : nid[000005-000009]
dgnettest: failed to parse MAC addr for Shasta node nid000006 02:00:00:00:00:3e
dgnettest: failed to parse MAC addr for Shasta node nid000009 02:00:00:00:00:2c
dgnettest: failed to parse MAC addr for Shasta node nid000005 02:00:00:00:00:3f
```

If the correct class type is used and MAC parsing errors are still printed, something may be cabled wrong or the MAC addresses may have been assigned incorrectly. The `-i` option can be used to force dgnettest to run despite these warnings.

8.2.3.5 Too Many Nodes for Network Class

The following error indicates that too many nodes were selected for the given network class type.

```
$ dgnettest_run.sh -n nid[001000-001009,001083-001146,001188-001219,001278-001279,001284-001347,001432-001499,001501-001506,001510-001511] -c 0
Running loopback tests on NIC 0 over : nid[001000-001009,001083-001146,001188-001219,001278-001279,001284-001347,001432-001499,001501-001506,001510-001511
Too many nodes selected for network class type 0, please check configuration
Too many nodes selected for network class type 0, please check configuration
...
```

8.2.3.6 Slurm Cannot Find Remote dgnettest Binaries

Running dgnettest requires that a copy of the test is present on each node. For some systems, dgnettest is only installed on the user access node (UAN). In this case, the `-b` option can be provided to use Slurm's broadcast feature to create temporary copies of dgnettest on each node for the duration of the run.

```
$ dgnettest_run.sh -n nid[000002-000028]
Running loopback tests on NIC 0 over : nid[000002-000028]
slurmstepd: error: execve(): /tmp/./dgnettest: No such file or directory
slurmstepd: error: execve(): /tmp/./dgnettest: No such file or directory
slurmstepd: error: execve(): /tmp/./dgnettest: No such file or directory
slurmstepd: error: execve(): /tmp/./dgnettest: No such file or directory
...
```

8.3 cxibwcheck.sh

The network diagnostic cxibwcheck.sh measures bi-directional loopback bandwidth for each Slingshot NIC on each node in a given set of nodes. It targets each node in parallel, measuring bandwidth for all of a node's NICs in series. The script should be installed on each node and optionally on the user access node (UAN) as well. The node copies can be run using Slurm, or the script can be executed from the UAN, where it uses pdsh to execute the node copies. When using pdsh, scalability is limited by the performance of the UAN. Using pdsh allows for testing both compute nodes and non-compute nodes. The diagnostic does not require dedicated access to the whole system. It can be run on a subset of nodes allocated by the batch system. The impact on performance of other applications is low.

Bi-directional RMDA write bandwidth is measured using the CXI diagnostic cxi_write_bw. More information for this diagnostic is provided in the CXI Diagnostics and Utilities documentation. Bandwidth is measured from each NIC to its connected switch and then back to itself.

Several network checks are performed prior to running the test. The nodelist is checked to verify that all nodes are reachable and that SSH can be used. The nodes are checked for properly configured IPv4 addresses and algorithmic MAC addresses (AMAs). A nameserver lookup is performed for each NIC, and the resulting IPv4 addresses are compared to what is configured. The NIC links are checked to ensure that they are up. The NIC PCIe links are verified. After the test completes, the PCIe links are checked again to verify that they have not degraded during the test.

The test measures, and can print, bi-directional bandwidth in stages. First the test is performed for NIC 0 on all nodes, then NIC 1, and so on. Once all NICs have been measured, a summary is printed that includes the number of hosts targeted, the overall NIC count, the number of failing NICs, and the number of NICs excluded from the test by the network checks. This is followed by the min, max, and mean bandwidths, along with a list of NICs whose performance falls below a threshold compared to the mean.

8.3.1 Example

This example shows a run using four dual-injection nodes.

```
# cxibwcheck.sh nid[000008-000011]
cxibwcheck.sh: firmware version: 1.6.0.320
Hosts = 4 Count = 8 Skipped = 0 Failed = 0 Missing = 0
Min = 39819 Mean = 40111 Max = 40987
Test passed
```

This example shows the same run with verbosity set to show the bi-directional bandwidth results for each NIC.

```
# cxibwcheck.sh -v nid[000008-000011]
cxibwcheck.sh: firmware version: 1.6.0.320
nid000009: cxi0: 39819
nid000008: cxi0: 39908
nid000011: cxi0: 39923
nid000010: cxi0: 40052
nid000009: cxi1: 40035
nid000008: cxi1: 39956
nid000011: cxi1: 40214
nid000010: cxi1: 40987
Hosts = 4 Count = 8 Skipped = 0 Failed = 0 Missing = 0
Min = 39819 Mean = 40111 Max = 40987
Test passed
```

8.3.2 Troubleshooting

When failures occur, often the details are contained in error logs that are created on the failing node(s). The log file location is printed for each failing node.

```
nid000002: cxi0: test failed (logfile is nid000002:/tmp/cxibwcheck.sh.errlog.144874)
nid000002: cxi1: test failed (logfile is nid000002:/tmp/cxibwcheck.sh.errlog.144874)
```

8.3.2.1 No Remote Copies of cxibwcheck.sh

Running cxibwcheck.sh requires that a copy of the script is present on each node. For some systems, cxibwcheck.sh is only installed on the UAN. In this case, the `-b` option can be provided to use pdcp to create temporary copies of cxibwcheck.sh on each node.

```
nid000008: bash: ./cxibwcheck.sh: No such file or directory
pdsh@uan: nid000008: ssh exited with exit code 127
```

8.3.2.2 No Remote Copies of cxi_write_bw

cxibwcheck.sh uses the CXI diagnostic cxi_write_bw to measure bandwidth. This is provided by the cray-libcxi-utils package, which should be installed on every node. See the CXI Diagnostics and Utilities documentation for more details.

```
taskset: failed to execute /usr/bin/cxi_write_bw: No such file or directory
```

8.3.2.3 Errors in Nodelist

Before running the test, the node list is verified. If a node cannot be reached, or if SSH key verification fails, an error is printed.

```
cxi-nid0: Host key verification failed.
pdsh@uan: cxi-nid0: ssh exited with exit code 255
cxibwcheck.sh: errors in nodelist (see /tmp/cxibwcheck.sh.log.19071) 0 remain
cxibwcheck.sh: nodelist is empty
```

8.3.2.4 IP Check Errors

The following errors indicate types of problems with the HSN link.

```
test failed for hsn0 link is down
```

```
test failed IP address is not configured for hsn0
```

```
IP lookup for nid000008-hsn0 failed, trying nid000008
IP lookup for nid000008 failed
Please validate the configured IP addresses. The test will continue but errors may be due to address configuration issues.

IP address check failed for hsn0: Configured: 10.150.0.189 Lookup: 10.168.0.49
Please validate the configured IP addresses. The test will continue but errors may be due to address configuration issues.
```

8.3.2.5 Bad AMA Found

This error indicates that the NIC has not been configured with an algorithmic MAC address.

```
Bad hsn0 AMA found:      link/ether 11:22:33:44:55:66 brd ff:ff:ff:ff:ff:ff
```

8.3.2.6 PCIe Check Errors

These errors indicate that a NIC PCIe link did not come up completely. The node should be rebooted, and if the problem persists, the card should be replaced.

```
found PCI width at x4

found PCIe Speed at 8GT/s
```

8.3.2.7 Low Bandwidth

If any NIC bandwidths fall far enough below the mean, they are listed at the end of the test, and the test fails. Low bandwidths indicate NIC or network issues. Cabling and link stability should be verified. Switch routing and switch and NIC QoS settings should be verified. NIC and switch error flags and counters should be checked.

```
# ./cxibwcheck.sh -b -t 1 cxi-nid[0-1]
cxibwcheck.sh: firmware version: 1.6.0.320
Hosts = 2 Count = 2 Skipped = 0 Failed = 1 Missing = 0
Min = 21215 Mean = 31822 Max = 42430
cxi-nid1: cxi0: bandwidth is low 21215 MB/s (66%)
Test failed (logfile is /tmp/cxibwcheck.sh.log.20078)
```

8.4 bwcheck.sh

The network diagnostic bwcheck.sh measures uni-directional loopback bandwidth for each Mellanox NIC on each node in a given set of nodes. It targets each node in parallel, measuring bandwidth for all of a node's NICs in series. The script should be installed on each node and optionally on the user access node (UAN) as well. The node copies can be run using Slurm, or the script can be executed from the UAN, where it will use pdsh to run on the nodes. Using pdsh allows for testing both compute nodes and non-compute nodes. When using pdsh, scalability is limited by the performance of the UAN. The diagnostic does not require dedicated access to the whole system. It can be run on a subset of nodes allocated by the batch system. The impact on performance of other applications is low.

Bi-directional RMDA write bandwidth is measured using the Perftest diagnostic ib_write_bw. Bandwidth is measured from each NIC to its connected switch and then back to itself.

Several network checks are performed prior to running the test. The nodelist is checked to verify that all nodes are reachable and that SSH can be used. The nodes are also checked for properly configured IPv4 addresses.

The test will measure and print uni-directional bandwidth in stages. First the test is performed for NIC 0 on all nodes, then NIC 1, and so on. Once all NICs have been measured, a summary is printed that includes the number of hosts targeted, the overall NIC count, the number of failing NICs, and the number of NICs excluded from the test by the network checks. This is followed by the min, max, and mean bandwidths, along with a list of NICs whose performance falls below a threshold compared to the mean.

8.4.1 Example

This example shows running the script with Slurm and using srun's broadcast option to distribute the script to the nodes.

```
# srun -N6 --bcast=/tmp/bwcheck.sh /tmp/bwcheck.sh
bwcheck.sh: checking nodelist
bwcheck.sh: firmware version: 16.26.1040
bwcheck.sh: starting test
nid003707: mlx5_0: 12288.58
nid003710: mlx5_0: 12311.46
nid003705: mlx5_0: 12300.33
nid003711: mlx5_0: 12326.43
nid003708: mlx5_0: 12288.89
nid003709: mlx5_0: 12263.94
nid003707: mlx5_1: 12272.08
```

```
nid003710: mlx5_1: 12273.13
nid003705: mlx5_1: 12256.29
nid003711: mlx5_1: 12249.00
nid003708: mlx5_1: 12235.52
nid003709: mlx5_1: 12273.98
Hosts = 6 Count = 12 Skipped = 0 Failed = 0 Missing = 0
Min = 12236 Mean = 12278 Max = 12326
Test passed
```

8.4.2 Troubleshooting

8.4.2.1 No Remote Copies of bwcheck.sh

Running bwcheck.sh requires that a copy of the script is present on each node. The following error indicates that the script has not been installed. This can be remedied by installing the cray-diags-fabric package on each node.

```
nid000008: bash: ./bwcheck.sh: No such file or directory
pdsh@uan: nid000008: ssh exited with exit code 127
```

8.4.2.2 No Remote Copies of ib_write_bw

bwcheck.sh uses the Perftest diagnostic ib_write_bw to measure bandwidth. This is provided by the infiniband-diags package.

```
nid000008: taskset: failed to execute /usr/bin/ib_write_bw: No such file or directory
nid000008: taskset: failed to execute /usr/bin/ib_write_bw: No such file or directory
nid000008: mlx5_0: test failed (logfile is nid000008:/tmp/bwcheck.sh.errlog.213234)
nid000008: taskset: failed to execute /usr/bin/ib_write_bw: No such file or directory
nid000008: taskset: failed to execute /usr/bin/ib_write_bw: No such file or directory
nid000008: mlx5_1: test failed (logfile is nid000008:/tmp/bwcheck.sh.errlog.213234)
```

8.4.2.3 Errors in Nodelist

Before running the test, the node list is verified. If a node cannot be reached, or if SSH key verification fails, an error is printed.

```
nid000008: Host key verification failed.
pdsh@uan: nid000008: ssh exited with exit code 255
bwcheck.sh: checking nodelist
bwcheck.sh: errors in nodelist (see /tmp/bwcheck.sh.log.19071) 0 remain
bwcheck.sh: nodelist is empty
```

8.4.2.4 IP Check Errors

The following errors indicate types of problems with the HSN link.

```
nid000008: test failed for eth4 link is down

nid000008: test failed IP address is not configured for eth4

nid000008: IP lookup for nid000008 failed

nid000008: IP address check failed for eth4: Configured: 10.150.0.189 Lookup: 10.168.0.49
```

8.4.2.5 Low Bandwidth

If any NIC bandwidths fall far enough below the mean, they are listed at the end of the test, and the test fails. Low bandwidths indicate NIC or network issues. Cabling and link stability should be verified. Switch routing and QoS settings should be verified. NIC and switch error flags and counters should be checked.

```
# srun -N2 --bcast=/tmp/bwcheck.sh /tmp/bwcheck.sh
bwcheck.sh: checking nodelist
bwcheck.sh: firmware version: 16.26.1040
bwcheck.sh: starting test
nid003707: mlx5_0: 12288.58
nid003710: mlx5_0: 12311.46
nid003707: mlx5_1: 6121.34
nid003710: mlx5_1: 12273.13
Hosts = 2 Count = 4 Skipped = 0 Failed = 1 Missing = 0
Min = 6121 Mean = 10749 Max = 12311
nid003707: mlx5_1: bandwidth is low 6121 MB/s (57%)
Test failed (logfile is /tmp/bwcheck.sh.log.20078)
```

8.5 cxiberstat.sh

The network diagnostic cxiberstat.sh measures NIC link corrected and uncorrected bit error rates (BERs). By default the script uses Slurm to remotely execute cxi_stat on each node. There is an option to use pdsh instead. Using pdsh allows for testing both compute nodes and non-compute nodes, but scalability is limited by the performance of the machine on which the script is running. The script does not need to be installed on each node, as it uses Slurm's broadcast feature to create a temporary copy of itself on each node. When using pdsh instead of slurm, remote execution is simply a series of bash commands. The diagnostic does not require dedicated access to the nodes. It should have no impact on node performance.

Corrected and Uncorrected BERs are measured using the CXI diagnostic cxi_stat. More information about this diagnostic is provided in the CXI Diagnostics and Utilities documentation. The BERs reflect errors on the NIC side of the links only.

8.5.1 Example Good Run

This example shows a run using Slurm and the default measurement duration of 4 seconds

```
# cxiberstat.sh
Note: Using a duration of 4 means the minimum measurable rate is 1.176e-12
      Node  Device    CCW_BER    UCW_BER
nid000006:    cxi0  9.412e-12       <min
nid000006:    cxi1  3.765e-11       <min
nid000009:    cxi0  1.412e-11       <min
nid000009:    cxi1  3.294e-11       <min
nid000013:    cxi0       <min       <min
nid000013:    cxi1       <min       <min
nid000035:    cxi0  2.353e-11       <min
nid000035:    cxi1       <min       <min
nid000036:    cxi0  6.118e-11       <min
nid000036:    cxi1  8.000e-11       <min
nid000037:    cxi0  6.118e-11       <min
nid000037:    cxi1  4.706e-11       <min
nid000047:    cxi0  1.976e-10       <min
nid000047:    cxi1  2.353e-11       <min
nid000048:    cxi0  9.412e-12       <min
nid000048:    cxi1  1.412e-11       <min
nid000049:    cxi0  1.412e-11       <min
nid000049:    cxi1  2.353e-11       <min
nid000050:    cxi0       <min       <min
nid000050:    cxi1  4.565e-10       <min
nid000051:    cxi0  1.412e-11       <min
nid000051:    cxi1  5.176e-11       <min
nid000052:    cxi0       <min       <min
nid000052:    cxi1       <min       <min
nid000060:    cxi0       <min       <min
nid000060:    cxi1  1.882e-11       <min

CCW BER Summary
  Bin    # Histogram
1e-05:  0
1e-06:  0
1e-07:  0
1e-08:  0
1e-09:  0
1e-10:  2 ======
1e-11: 15 ========================================
1e-12:  2 ======
 <min:  7

UCW BER Summary
No UCW BERs above min measurable value
```

8.5.2 Example Bad Run

This example shows a local node run where one device reports a high corrected BER and a non-zero uncorrected BER. If this occurs with a cabled NIC, the cable should be reseated or replaced. If that does not resolve the problem, or if this occurs with a NIC that has a backplane connection, further investigation is needed. The NIC card may need to be reseated or replaced. It may be helpful to check error rates at the connected switch port.

```
# cxiberstat.sh -w x2510c7s6b0n0
x2510c7s6b0n0: Note: Using a duration of 4 means the minimum measurable rate is 1.176e-12
x2510c7s6b0n0:           Node  Device    CCW_BER    UCW_BER
x2510c7s6b0n0: x2510c7s6b0n0:    cxi0       <min       <min
x2510c7s6b0n0: x2510c7s6b0n0:    cxi1  1.412e-11       <min
x2510c7s6b0n0: x2510c7s6b0n0:    cxi2  2.519e-06  4.706e-12
x2510c7s6b0n0: x2510c7s6b0n0:    cxi3  9.412e-12       <min
```

8.5.3 Troubleshooting

8.5.3.1 Unexpected Link Mode

The following error indicates that cxi_stat could not determine the link speed. This likely means that the link went down while the measurement was being performed, perhaps due to a high error rate.

```
x2510c7s6b0n0: Warning: encountered unexpected Link Mode NA for non-zero error rate 595424
```

8.5.3.2 No Remote Copies of cxi_stat

cxiberstat.sh uses the CXI diagnostic cxi_stat to measure BERs. This is provided by the cray-libcxi-utils package, which should be installed on every node. See the CXI Diagnostics and Utilities documentation for more details.

```
# ./cxiberstat.sh
Errors occurred:
/tmp/cxiberstat.sh: line 64: cxi_stat: command not found
/tmp/cxiberstat.sh: line 64: cxi_stat: command not found

# ./cxiberstat.sh -p -w x0c0s0b0n[0-1]
Errors occurred:
x0c0s0b0n0: bash: cxi_stat: command not found
x0c0s0b0n1: bash: cxi_stat: command not found
```

8.5.3.3 No Output

If cxiberstat.sh runs without error but prints no output, this likely means that cxi_stat ran on each node and found no NICs. Verify that the NIC drivers have been installed.

8.5.3.4 Missing Option in cxi_stat

Older versions of cxi_stat do not support specifying the BER measurement duration. If the following error is seen, try running cxiberstat.sh without the `-u` option.

```
Errors occurred:
cxi_stat: invalid option -- 'p'
```

9 Firmware Update Using FAS

9.1 Upgrade Slingshot switch firmware on HPE Cray EX manually

This section describes manual process to update the switch firmware using FAS. Follow these instructions if the `fmc-update-switch-firmware.sh` failed to update the switch firmware.

9.1.1 Step 1: Identify the switch firmware RPM file

The release distribution gzipped tar file consists of Slingshot Switch Firmware RPM. It is extracted in the working directory.

```
ncn-m001:~ # ls
slingshot-x.y.z-bbb/                slingshot-Cray-EX-x.y.z-bbb.tar.gz
sc-firmware-x.y.z-bbbbbbbb_bbbbb.x86_64.rpm  fmc-update-switch-firmware.sh
```

9.1.2 Step 2: Load firmware image into FAS

1. Extract the desired firmware file (.itb) from the RPM.

   NOTE: We will use firmware version 1.6.0-996 for example commands in this section.

   ```
   ncn-m001# rpm2cpio sc-firmware-1.6.0.996-20211103180248_cdaad06.x86_64.rpm | cpio -idmv
   ```

   This will extract controllers-1.6.0-996.itb and controllers-1.6.0-996.json file in opt/cray/FW/sc-firmware/ directory inside the pwd (present working directory).

2. Upload the firmware image (.itb file) to S3 using `cray artifacts create` command.

   - first argument is the bucket-name `fw-update`
   - second argument is the key in that bucket
   - third argument is the file to upload

```
ncn-m001# cray artifacts create fw-update slingshot/controllers-1.6.0-996.itb  opt/cray/FW/sc-firmware/controllers-1.6.0-996.itb

artifact = "slingshot/controllers-1.6.0-996.itb"
Key = "slingshot/controllers-1.6.0-996.itb"
```

3. Create a FAS image record file.

   `slingshotImage.json` is the example image record used in this step. The image record is slightly different from the image meta file (e.g. controllers-1.6.0-996.json) present in the RPM; hence first make a copy of the image meta file (e.g. controllers-1.6.0-996.json) and name it as slingshotImage.json.

```
ncn-m001: # cat slingshotImage.json
{
   "deviceType": "RouterBMC",
   "manufacturer": "cray",
   "models": [
      "ColoradoSwitchBoard_REV_A",
      "ColoradoSwitchBoard_REV_B",
      "ColoradoSwitchBoard_REV_C",
      "ColoradoSwtBrd_revA",
      "ColoradoSwtBrd_revB",
      "ColoradoSwtBrd_revC",
      "ColoradoSWB_revA",
      "ColoradoSWB_revB",
      "ColoradoSWB_revC",
      "ColoradoSwtBrd_REV_A",
      "ColoradoSwtBrd_REV_B",
      "ColoradoSwtBrd_REV_C",
      "101878104_",
      "ColumbiaSwitchBoard_REV_A",
      "ColumbiaSwitchBoard_REV_B",
      "ColumbiaSwitchBoard_REV_D",
      "ColumbiaSwtBrd_revA",
      "ColumbiaSwtBrd_revB",
      "ColumbiaSwtBrd_revD",
      "ColumbiaSWB_revA",
      "ColumbiaSWB_revB",
      "ColumbiaSWB_revD",
      "ColumbiaSwtBrd_REV_A",
      "ColumbiaSwtBrd_REV_B",
      "ColumbiaSwtBrd_REV_D",
      "HPE_Slingshot_1_Switch_Blade",
      "HPE_Slingshot_1_Top-of-Rack_Switch"
   ],
   "target": "BMC",
   "tags": [
      "default"
   ],
   "softwareIds": [
      "sc:*:*:*"
   ],
   "firmwareVersion": "sc.1.6.0-996-prod-master.arm64.2021-11-03T22:35:12+00:00.964b67a",
   "semanticFirmwareVersion": "1.6.0-996",
   "pollingSpeedSeconds": 30,
   "s3URL": "s3:/fw-update/slingshot/controllers-1.6.0-996.itb"
}
```

   Following are some changes to be done to slingshotImage.json file after copying data from the image meta file (e.g. controllers-1.6.0-996.json) :

   Addition of 2 fields in slingshotImage.json : Ensure these two fields `pollingSpeedSeconds`, and `s3URL` are newly added to the image record file slingshotImage.json as shown in the above sample image record file. In s3URL value, update the controllers-x.y.z-bbbb.itb version to match with the version of the switch firmware e.g. 1.6.0-996 that you are using. **NOTE**: For Slingshot versions prior to v1.5, add a `softwareIds` field as shown above to the image record file slingshotImage.json .

   Removal of 1 field in slingshotImage.json : Note that the field `fileName` is present in the image meta file (e.g. controllers-1.6.0-996.json) but needs to be removed from image record file slingshotImage.json.

   Change of 1 field name in slingshotImage.json : Change the key of one field from `targets` to `target` in slingshotImage.json file, and remove the square brackets around its value “BMC”.

4. Create the image record in FAS.

   Note the `imageID` returned from `cray fas images create`. You will use it to verify the status of image creation.

```
ncn-m001:# cray fas images create slingshotImage.json
imageID = "09614e7a-50f2-4752-8073-289319305cf1"
```

5. Verify the image record has been created. Use `imageID` from the above step.

```
ncn-m001:# cray fas images describe "09614e7a-50f2-4752-8073-289319305cf1" --format json
{
"imageID": "09614e7a-50f2-4752-8073-289319305cf1",
"createTime": "2021-11-04T18:03:32Z",
"deviceType": "RouterBMC",
"manufacturer": "cray",
"models": [
   "ColoradoSwitchBoard_REV_A",
   "ColoradoSwitchBoard_REV_B",
   "ColoradoSwitchBoard_REV_C",
   "ColoradoSwtBrd_revA",
   "ColoradoSwtBrd_revB",
   "ColoradoSwtBrd_revC",
   "ColoradoSWB_revA",
   "ColoradoSWB_revB",
   "ColoradoSWB_revC",
   "ColoradoSwtBrd_REV_A",
   "ColoradoSwtBrd_REV_B",
   "ColoradoSwtBrd_REV_C",
   "101878104_",
   "ColumbiaSwitchBoard_REV_A",
   "ColumbiaSwitchBoard_REV_B",
   "ColumbiaSwitchBoard_REV_D",
   "ColumbiaSwtBrd_revA",
   "ColumbiaSwtBrd_revB",
   "ColumbiaSwtBrd_revD",
   "ColumbiaSWB_revA",
   "ColumbiaSWB_revB",
   "ColumbiaSWB_revD",
   "ColumbiaSwtBrd_REV_A",
   "ColumbiaSwtBrd_REV_B",
   "ColumbiaSwtBrd_REV_D",
   "HPE_Slingshot_1_Switch_Blade",
   "HPE_Slingshot_1_Top-of-Rack_Switch"
],
"softwareIds": [
   "sc:*:*:*"
],
"target": "BMC",
"tags": [
   "default"
],
"firmwareVersion": "sc.1.6.0-996-prod-master.arm64.2021-11-03T22:35:12+00:00.964b67a",
"semanticFirmwareVersion": "1.6.0-996",
"pollingSpeedSeconds": 30,
"s3URL": "s3:/fw-update/slingshot/controllers-1.6.0-996.itb"
}
```

9.1.3 Step 3: Set the permissions for S3

For HPE Cray EX v1.3 and v1.4 systems

Only use this procedure for HPE Cray EX v1.3 - v1.4 systems to set permissions for S3.

1. Retrieve the cray-fas-loader job

   Retrieve the job name. In the following example, the returned job name is cray-fas-loader-1, which is the job to rerun in this scenario.

   ```
   ncn-m001:# kubectl -n services get jobs | grep fas-loader
   cray-fas-loader-1 1/1 3m11s 3d7h
   ncn-m001: #
   ```

2. Rerun the cray-fas-loader job. Note, after "-f" there is a "-".

   ```
   ncn-m001:# kubectl -n services get job cray-fas-loader-1 -o json | jq 'del(.spec.selector)' \
    | jq 'del(.spec.template.metadata.labels."controller-uid")' \
    | kubectl replace --force -f -
   job.batch "cray-fas-loader-1" deleted
   job.batch/cray-fas-loader-1 replaced
   ncn-m001:#
   ```

   Make sure the FAS Loader job is complete. Depending on the number of images in FAS, this could take 5-7 minutes.

   ```
   ncn-m001:~/slingshot# kubectl -n services get jobs | awk 'NR == 1 || /fas-loader/'
   NAME                 COMPLETIONS   DURATION    AGE
   cray-fas-loader-1        1/1        7m35s     7m35s
   ```

   If FAS Loader job is not completed, look at the error log using these commands.

   Get the fas loader pod name:

```
ncn-m001:~/slingshot# kubectl get pods -n services | awk 'NR == 1 || /fas-loader/'
NAME                        READY   STATUS      RESTARTS    AGE
cray-fas-loader-1-pnn6c 2/2 Running 2 9m38s
```

Check the logs dumped on screen using this command:

```
ncn-m001:~/slingshot # kubectl logs -n services cray-fas-loader-1-pnn6c -c cray-fas-loader
```

For HPE Cray EX v1.5 systems

Only use this procedure HPE Cray EX v1.5 systems to set permissions for S3.

1. Run the cray fas loader.

   Run the following command. This will return `loaderRunID` which will be used to check the status of the run.

   ```
   ncn-m001# cray fas loader nexus create
   loaderRunID = "8aff49d9-f608-4e90-9911-bd3e4cf52f70"
   ```

2. Check the results of the loader run using the following command. Use `loaderRunID` from the above step.

   ```
   ncn-m001# cray fas loader describe "8aff49d9-f608-4e90-9911-bd3e4cf52f70" --format json | jq '.loaderRunOutput | .[-10:]'
   [
      "2021-10-12T17:44:23Z-FWLoader-INFO-update ACL to public-read for 4c3005db0a8e11ec8d9572a38444a3ba/ilo5_246.bin",
      "2021-10-12T17:44:23Z-FWLoader-INFO-update ACL to public-read for 4ab9e03af0b111eba8d81a28e5cdfae5/ex425.bios-1.5.0.tar.gz",
      "2021-10-12T17:44:23Z-FWLoader-INFO-update ACL to public-read for slingshot/controllers-1.6.0-996.itb",
      "2021-10-12T17:44:23Z-FWLoader-INFO-update ACL to public-read for 3f1a6b5bf0b111eb9fb51a28e5cdfae5/A48_2.40_02_24_2021.signed.flash",
      "2021-10-12T17:44:23Z-FWLoader-INFO-update ACL to public-read for 57797d4e222b11ec98ad72a38444a3ba/controllers-1.5.908.itb",
      "2021-10-12T17:44:23Z-FWLoader-INFO-update ACL to public-read for 2b4ce0bf1d8711ec937772a38444a3ba/controllers-1.5.885.itb",
      "2021-10-12T17:44:23Z-FWLoader-INFO-update ACL to public-read for cc5d52bbefc911eb9d5b1a28e5cdfae5/ex425.bios-1.4.3.tar.gz",
      "2021-10-12T17:44:23Z-FWLoader-INFO-update ACL to public-read for 30c896f81a5c11ec9bb172a38444a3ba/controllers-1.5.857.itb",
      "2021-10-12T17:44:23Z-FWLoader-INFO-finished updating images ACL",
      "2021-10-12T17:44:23Z-FWLoader-INFO-*** Number of Updates: 0 ***"
   ]
   ```

   A successful run will end with `Number of Updates:`. Note that the FAS loader will not overwrite image records already in FAS. `Number of Updates` will be the number of new images found in the RPM. If the number is 0, all images were already in FAS.

9.1.4 Step 4: Update Slingshot switch firmware

1. List available images.

   List available images and select one to use for the update. They are uniquely identified by `imageID`.

   ```
   ncn-m001:# cray fas images list --format json | jq '.[] | .[] | select(.target=="BMC" and .manufacturer=="cray" and .deviceType=="RouterBMC")'
   ```

   If many images fit this this description, use the latest version number.

   ```
   ncn-m001:# cray fas images list --format json | jq '.[] | .[] | select(.target=="BMC" and .manufacturer=="cray" and .deviceType=="RouterBMC" and .s
   ```

   Use `imageID` in the `fas_slingshot_workaround.json` input file when a specific switch fails to update and you want to force the update as shown in one of the examples below.

2. Determine the switch component names (xnames).

   ```
   ncn-m001:~# cray hsm inventory redfishEndpoints list --type RouterBMC --format json \
     | jq -r 'flatten' | jq -c '[ .[] | .ID ]'

     ["x3000c0r15b0","x9000c1r1b0","x9000c1r3b0","x9000c1r5b0","x9000c1r7b0","x9000c3r1b0","x9000c3r3b0","x9000c3r5b0"]
   ```

3. Create an update input file.

   These examples are for 3 separate situations:

   - update specific switches by xname,
   - update all switches,
   - force update to specific switches.

   This example `fas_slingshot.json` input file updates firmware on specific switches identified by xnames. The switch component names listed here are examples and need to be provided from your environment.

   Note `overrideDryrun: false`. This will perform a dryrun, without update. To perform an actual update, set the `overrideDryrun` flag to true.

   ```
   ncn-m001: # more fas_slingshot_dryrun.json
   {
   "stateComponentFilter":
     {
   ```

```
      "xnames":
         ["x3000c0r15b0",
         "x9000c1r1b0",
         "x9000c1r3b0",
         "x9000c1r5b0",
         "x9000c1r7b0",
         "x9000c3r1b0",
         "x9000c3r3b0",
         "x9000c3r5b0"]
   },
"targetFilter":
   {
      "targets":
         [
         "BMC"
         ]
   },
"command":
   {
      "version": "latest",
      "tag": "default",
      "overrideDryrun": false,
      "timeLimit": 1000,
      "restoreNotPossibleOverride": true,
      "overwriteSameImage": false,
      "description": "All Cray RouterBMCs"
   }
}
```

This example input file updates all switches. Note the `overwriteDryrun: true` setting. This would perform an actual update.

```
ncn-m001: # more fas_slingshot.json
{
   "inventoryHardwareFilter": {
     "manufacturer": "cray"
      },
   "stateComponentFilter": {
     "deviceTypes": [
     "routerBMC"
     ]
    },
    "targetFilter": {
        "targets": [
        "BMC"
         ]
    },
   "command": {
    "version": "latest",
    "tag": "default",
    "overrideDryrun": true,
    "timeLimit": 1000,
    "restoreNotPossibleOverride": true,
    "overwriteSameImage": false,
    "description": "All Cray RouterBMCs"
     }
}
```

If a specific switch fails to update, this `fas_slingshot_workaround.json` should force the switch to update. This may be necessary if the switches are running their recovery image. Notice the `imageID`.

```
ncn-m001:# more fas_slingshot_workaround.json
{
   "stateComponentFilter": {
  "xnames": [
  "x3000c0r21b0"
   ]
   },
   "imageFilter": {
  "imageID": "09614e7a-50f2-4752-8073-289319305cf1",
  "overrideImage": true
   },
   "targetFilter": {
  "targets": [
  "BMC"
   ]
   },
   "command": {
  "overrideDryrun": true,
  "restoreNotPossibleOverride": true,
  "overwriteSameImage": false
    }
}
```

4. Dryrun update firmware on Slingshot switches.

Within the Slingshot installation process we are updating the firmware on just the Slingshot switches, deviceType = RouterBMC.

Complete a dryrun. The jsonfile would specify `overrideDryrun": false`. Note the value of `actionID` that is returned.

```
ncn-m001:# cray fas actions create {jsonfile}
```

Poll the status of the action until `state = completed`. Use the `actionID` from the above step.

```
ncn-m001:# cray fas actions describe {actionID} --format json
```

Interpret the outcome of the dryrun.

Look at the counts and determine if the dryrun identified any hardware to update.

For the steps below, the following returned messages will help determine if a firmware update is needed. The following are end states for operations. The Firmware `action` itself should be `completed` once all operations have finished.

- `NoOp`: Nothing to do, already at version.
- `NoSol`: No viable image is available; this will not be updated.
- `succeeded`:

  If `dryrun`: The operation should succeed if performed as a `live update`. `succeeded` means that FAS identified that it COULD update an xname + target with the declared strategy.

  If `live update`: the operation succeeded, and has updated the xname + target to the identified version.

- `failed`:

  If `dryrun`: There is something that FAS could do, but it likely would fail; most likely because the file is missing.

  If `live update`: the operation failed, the identified version could not be put on the xname + target.

If `succeeded` count > 0, now perform a live update.

5. Update firmware on Slingshot switches.

   Update the JSON file `overrideDryrun` to `true` and run the following command:

   ```
   ncn-m001:# cray fas actions create {jsonfile}
   ```

   Note the value of `actionID` that is returned.

   Poll the status of the action until the action `state` is `completed`. Use the `actionID` from the above step

   ```
   ncn-m001:# cray fas actions describe {actionID} --format json
   ```

6. Verify switch firmware update is complete by checking firmware version.

   Log on to the Slingshot fabric manager pod

   ```
   ncn-m001:~# kubectl get pods -A |grep fabric |awk '{print $2}'
   slingshot-fabric-manager-5dc448779c-d8t72

   ncn-m001:~# kubectl exec -it -n services slingshot-fabric-manager-5dc448779c-d8t72 -- /bin/bash
   Defaulting container name to slingshot-fabric-manager.
   ```

   Check the switch firmware version is same as desired version

   ```
   fmn_update_switch_firmware ShowFirmware all

   x3000c0r15b0 has firmware version: 1.6.0-996 Health: OK. State: Enabled
   x9000c1r1b0 has firmware version: 1.6.0-996 Health: OK. State: Enabled
   x9000c1r3b0 has firmware version: 1.6.0-996 Health: OK. State: Enabled
   x9000c1r5b0 has firmware version: 1.6.0-996 Health: OK. State: Enabled
   x9000c1r7b0 has firmware version: 1.6.0-996 Health: OK. State: Enabled
   x9000c3r1b0 has firmware version: 1.6.0-996 Health: OK. State: Enabled
   x9000c3r3b0 has firmware version: 1.6.0-996 Health: OK. State: Enabled
   x9000c3r5b0 has firmware version: 1.6.0-996 Health: OK. State: Enabled
   ```

7. Use fmn_switch_reset to clear the ASIC and switch controller after switch firmware update

   After an update there should be an BMC reboot, followed by a ASCI reset. We do this via the command `fmn_switch_reset`, run twice with a 120 second wait between each command:

For all switches in the configuration:

```
fmn_switch_reset -a -k; sleep 120; fmn_switch_reset -a -r
```

For specific switches:

```
fmn_switch_reset -k -i x0c0r0b0 x1c1r1b1; sleep 120; fmn_switch_reset -r -i x0c0r0b0 x1c1r1b1
```

The Slingshot switch firmware upgrade is complete.

Please return to the HPE Cray EX System Software Getting Started Guide (S-8000) upon completion of this step.

10 Copyright and Version

Slingshot: 1.7.3-77

Doc git hash: 765943227d33c02605d202d8bad1b87a34c5a512

Generated: Fri Jun 24 2022