System Health Check

System Health Check

Monitoring systems in a network proactively helps prevent potential issues and take preventive actions. This section illustrates how you can monitor the system health using the health check service. This service helps to analyze the system health by monitoring, tracking and analyzing metrics that are critical for functioning of the router.

The system health can be gauged with the values reported by these metrics when the configured threshold values exceed or are nearing the threshold value.

This table describes the significant fields shown in the display.

Table 1: System Health Check Metrics

Metric	Parameter Tracked	Considered Unhealthy When
Critical System Resources	CPU, free memory, file system, shared memory	The respective metric has exceeded the threshold
Infrastructure Services	Field Programmable Device (FPD), fabric health, platform, redundancy	Any component of the service is down or in an error state
Counters	Interface-counters, fabric-statistics, asic-errors	Any specific counter exhibits a consistent increase in drop/error count over the last n runs (n is configurable through CLI, default is 10)

By default, metrics for system resources are configured with preset threshold values. You can customize the metrics to be monitored by disabling or enabling metrics of interest based on your requirement.

Each metric is tracked and compared with that of the configured threshold, and the state of the resource is classified accordingly.

The system resources exhibit one of these states:

- **Normal:** The resource usage is less than the threshold value.
- **Minor:** The resource usage is more than the minor threshold, but less than the severe threshold value.
- **Severe:** The resource usage is more than the severe threshold, but less than the critical threshold value.
- **Critical:** The resource usage is more than the critical threshold value.

The infrastructure services show one of these states:

- **Normal:** The resource operation is as expected.
- **Warning:** The resource needs attention. For example, a warning is displayed when the FPD needs an upgrade.

The health check service is packaged as an optional RPM. This is not part of the base package and you must explicitly install this RPM.

You can configure the metrics and their values using CLI. In addition to the CLI, the service supports NETCONF client to apply configuration (`Cisco-IOS-XR-healthcheck-cfg.yang`) and retrieve operational data (`Cisco-IOS-XR-healthcheck-oper.yang`) using YANG data models. It also supports subscribing to metrics and their reports to stream telemetry data. For more information about streaming telemetry data, see *Telemetry Configuration Guide for Cisco 8000 Series Routers*.

Configure Health Check

To enable health check, you must configure the following:

- **netconf-yang agent ssh**
- **healthcheck enable**
- From IOS XR Release 7.3.3 onwards, you must also enable Google Remote Procedure Call (gRPC) using the command **grpc local-connection**.

Configuration Example

```
Router# config
Router(config)# netconf-yang agent ssh
Router(config)# grpc local-connection
Router(config)# healthcheck enable
Router(config)# commit
```

To change the preset cadence, use the **healthcheck cadence** *cadence-value* command:

```
Router(config)#healthcheck cadence 30
```

Note

- Healthcheck use-cases will not work if **grpc no-tls** is configured.

Monitoring Critical System Resources

This task explains how to check the health of a system using operational data from the network. The data can be queried by both CLI and NETCONF RPC, and can also be streamed using telemetry.

Step 1 Check the status of all metrics with its associated threshold and configured parameters in the system.

Example:

```
Router#show healthcheck status
Healthcheck status: Enabled

Collector Cadence: 60 seconds

System Resource metrics
  cpu
      Thresholds: Minor: 10%
                  Severe: 20%
                  Critical: 30%

       Tracked CPU utilization: 15 min avg utilization

   free-memory
        Thresholds: Minor: 10%
                    Severe: 8%
                    Critical: 5%

   filesystem
        Thresholds: Minor: 80%
                    Severe: 95%
                    Critical: 99%

   shared-memory
        Thresholds: Minor: 80%
                    Severe: 95%
                    Critical: 99%

Infra Services metrics
   fpd

   fabric-health
```

Step 2 View the health state for each enabled metric.

Example:

```
Router#show healthcheck report
Healthcheck report for enabled metrics

cpu
  State: Normal

free-momry
   State: Normal

shared-memory
   State: Normal

fpd
   State: Warning
```

```
One or more FPDs are in NEED UPGD state

fabric-health
   State: Normal
```

In the above output, the state of the FPD shows a warning message that indicates an FPD upgrade is required.

To further investigate the warning message, check the metric information. Here, for example, check the FPD state.

```
FPD Metric State: Warning
Last Update Time: 17 Feb 18:28:57.917193
FPD Service State: Enabled
Number of Active Nodes: 69

Node Name: 0/0/CPU0
    Card Name: 8800-LC-48H
    FPD Name: Bios
    HW Version: 0.31
    Status: NEED UPGD
    Run Version: 5.01
    Programmed Version: 5.01

-------------Truncated for brevity---------------
```

The `Last Update Time` is the timestamp when the health for that metric was computed. This timestamp gets refreshed with each collector run based on the cadence.

Step 3 Customize the health check threshold value for the following parameters:

- **Metric:** To list the metrics that can be configured, use the command:

```
Router(config)#healthcheck metric ?
  cpu            cpu configurations(cisco-support)
  fabric-health  fabric configurations(cisco-support)
  filesystem     Filesystem usage configurations(cisco-support)
  fpd            FPD configurations(cisco-support)
  free-mem       free memory configurations(cisco-support)
  shared-mem     shared memory configurations(cisco-support)
```

For example, to change the preset value of CPU metric, use the command:

```
Router(config)#healthcheck metric cpu ?
   threshold minor, severe or critical threshold
   avg_cpu_util 1min, 5min or 15min
        ios(config)#healthcheck metric cpu threshold ?
        minor       minor threshold in %
        severe      severe threshold in %
        critical    critical threshold in %
```

- Disable or enable metrics to selectively filter some metrics. By default, all metrics are enabled.

```
Router(config)#[no] healthcheck metric cpu disable
Router(config)#[no] healthcheck metric free-mem disable
```

Monitoring Infrastructure Services

This task explains how to check the health of the infrastructure services of a system. The data can be queried by both CLI and NETCONF RPC, and can also be streamed using telemetry.

Step 1 Check the health status of the infrastructure metrics in the system. By default, the router software enables the health check for infrastructure services.

Example:

The below example shows how to obtain the health-check status for the platform metric:

```
Router# show healthcheck metric platform
Platform Metric State: Normal ==========> Health of the metric
Last Update Time: 25 Jun 05:17:03.508172 =====> Timestamp at which the metric data was collected
Platform Service State: Enabled =====> Service state of Platform
Number of Racks: 1 ======> Total number of racks in the testbed
Rack Name: 0
Number of Slots: 12
Slot Name: RP0
Number of Instances: 2
Instance Name: CPU0
Node Name 0/RP0/CPU0
Card Type 8800-RP
Card Redundancy State Active
Admin State NSHUT
Oper State IOS XR RUN
```

Example:

The below example shows how to obtain the health-check status for the redundancy metric:

```
Router# show healthcheck metric redundancy
Redundancy Metric State: Normal ==========> Health of the metric
Last Update Time: 25 Jun 05:21:14.562291 =====> Timestamp at which the metric data was collected
Redundancy Service State: Enabled =====> Service state of the metric
Active: 0/RP0/CPU0
Standby: 0/RP1/CPU0
HA State: Node Ready
NSR State: Ready
```

Step 2 Disable health-check of any of the metrics, if required. By default, all metrics are enabled.

Example:

The below example shows how to disable the health-check status for the platform metric:

```
Router(config)# healthcheck metric platform disable
Router(config)# commit
```

Example:

The below example shows how to disable the health-check status for the redundancy metric:

```
Router(config)# healthcheck metric redundancy disable
Router(config)# commit
```

Monitoring Counters

This task explains how to check the health of the counters of a system. The counter values that can be monitored are interface-counters, asic-errors and fabric-statistics.

From IOS XR Release 7.3.5 onwards, all interfaces, including bundles, sub-interfaces, physical interfaces, can be monitored via health check. Previously, only physical interfaces could be monitored.

Step 1 Configure the size of the buffer which stores the history of the counter values as shown in the below examples.

Example:

The below example shows how to configure the buffer-size for the **interface-counters** to store values for the last 5 cadence snapshots:

```
Router(config)# healthcheck metric intf-counters counter-size 5
Router(config)# commit
```

Example:

The below example shows how to configure the buffer-size for the **asic-errors** counters to store values for the last 5 cadence snapshots:

```
Router(config)# healthcheck metric asic-errors counter-size 5
Router(config)# commit
```

Example:

The below example shows how to configure the buffer-size for the **fabric-stats** counters to store values for the last 5 cadence snapshots:

```
Router(config)# healthcheck metric fabric-stats counter-size 5
Router(config)# commit
```

Step 2 Configure the list of interfaces for which the **interface-counters** should be tracked as shown in the below examples. This is possible only for the **interface-counters** metric.

Example:

The below example shows how to configure the list of interfaces for which the **interface-counters** need to be tracked:

```
Router(config)# healthcheck metric intf-counters intf-list MgmtEth0/RP0/CPU0/0 HundredGigE0/0/0/0
Router(config)# commit
```

Example:

The below example shows how to configure all the interfaces so that the **interface-counters** are tracked for them:

```
Router(config)# healthcheck metric intf-counters intf-list all
Router(config)# commit
```

Step 3 By default, the router software enables the health-check for counters. Check the health status of the counters in the system as shown in the below examples.

Example:

The below example shows how to obtain the health-check status for the interface-counters:

```
Router# show healthcheck metric interface-counters summary
Interface-counters Health State: Normal ==========> Health of the metric
Last Update Time: 25 Jun 05:59:33.965851 =====> Timestamp at which the metric data was collected
Interface-counters Service State: Enabled =====> Service state of the metric
Interface MgmtEth0/RP0/CPU0/0 =====> Configured interface for healthcheck monitoring
```

```
Counter-Names Count Average Consistently-Increasing
------------------------------------------------------------------------------------------------
output-buffers-failures 0 0 N
Counter-Names =====> Name of the counters
Count =====> Value of the counter collected at "Last Update Time"
Average =====> Average of all values available in buffer
Consistently-Increasing =====> Trend of the counter values, as per data available in buffer
```

```
Router# show healthcheck metric interface-counters detail all
Thu Jun 25 06:02:03.145 UTC
Last Update Time: 25 Jun 06:01:35.217089 =====> Timestamp at which the metric data was collected
Interface MgmtEth0/RP0/CPU0/0 =====> Configured interface for healthcheck monitoring
Following table displays data for last <x=5> values collected in periodic cadence intervals
----------------------------------------------------------------------------------------------------
Counter-name Last 5 values
LHS = Earliest RHS = Latest
----------------------------------------------------------------------------------------------------
output-buffers-failures 0 0 0 0 0
parity-packets-received 0 0 0 0 0
```

Example:

The below example shows how to obtain the health-check status for the asic-errors:

```
Router# show healthcheck metric asic-errors summary
Asic-errors Health State: Normal ==========> Health of the metric
Last Update Time: 25 Jun 06:20:47.65152 =====> Timestamp at which the metric data was collected
Asic-errors Service State: Enabled =====> Service state of the metric
Node Name: 0/1/CPU0 =====> Node name for healthcheck monitoring

Instance: 0 =====> Instance of the Node

Counter-Names Count Average Consistently-Increasing
------------------------------------------------------------------------------------------------
Link Errors 0 0 N
Counter-Names =====> Name of the counters
Count =====> Value of the counter collected at "Last Update Time"
Average =====> Average of all values available in buffer
Consistently-Increasing =====> Trend of the counter values, as per data available in buffer
```

```
Router# show healthcheck metric asic-errors detail all
Thu Jun 25 06:25:13.778 UTC
Last Update Time: 25 Jun 06:24:49.510525 =====> Timestamp at which the metric data was collected
Node Name: 0/1/CPU0 =====> Node name for healthcheck monitoring
Instance: 0 =====> Instance of the Node
Following table displays data for last <x=5> values collected in periodic cadence intervals
----------------------------------------------------------------------------------------------------
Counter-name Last 5 values
LHS = Earliest RHS = Latest
----------------------------------------------------------------------------------------------------
Link Errors            0     0     0     0     0
```

Example:

The below example shows how to obtain the health-check status for the fabric-stats:

```
Router# show healthcheck metric fabric-stats summary
Thu Jun 25 06:51:13.154 UTC
Fabric-stats Health State: Normal ==========> Health of the metric
Last Update Time: 25 Jun 06:51:05.669753 =====> Timestamp at which the metric data was collected
Fabric-stats Service State: Enabled =====> Service state of the metric
Fabric plane id 0 =====> Plane ID
Counter-Names Count Average Consistently-Increasing
------------------------------------------------------------------------------------------------
mcast-lost-cells 0 0 N
Counter-Names =====> Name of the counters
```

```
Count =====> Value of the counter collected at "Last Update Time"
Average =====> Average of all values available in buffer
Consistently-Increasing =====> Trend of the counter values, as per data available in buffer
```

```
Router# show healthcheck metric fabric-stats detail all
Thu Jun 25 06:56:20.944 UTC
Last Update Time: 25 Jun 06:56:08.818528 =====> Timestamp at which the metric data was collected
Fabric Plane id 0 =====> Fabric Plane ID

Following table displays data for last <x=5> values collected in periodic cadence intervals
----------------------------------------------------------------------------------------------------
Counter-name Last 5 values
LHS = Earliest RHS = Latest
----------------------------------------------------------------------------------------------------
mcast-lost-cells 0 0 0 0 0
```

Step 4 If required, disable health-check of any of the counters. By default, all counters are enabled.

Example:

The below example shows how to disable the health-check status for the interface-counters:

```
Router(config)# healthcheck metric intf-counters disable
Router(config)# commit
```

Example:

The below example shows how to disable the health-check status for the asic-errors:

```
Router(config)# healthcheck metric asic-errors disable
Router(config)# commit
```

Example:

The below example shows how to disable the health-check status for the fabric-stats:

```
Router(config)# healthcheck metric fabric-stats disable
Router(config)# commit
```

System Health Check Use-Cases

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
System Health Check Use-cases	Release 7.3.3 Release 7.5.4	System Health Check use-cases are a version of the system health check where the user can monitor specific metrics of the system to determine the health and detect potential failures in the system caused by ASIC reset or packet drops. When seen from the device health point of view, it is conceptually determining and analyzing metrics that detect anomalies in the router. When the metric degrades beyond a certain threshold, the router itself raises the alarm. This service supports NETCONF client retrieve operational data using the following YANG data models: • Cisco-IOS-XR-ofa-npu-stats-oper.yang • Cisco-IOS-XR-infra-syslog-oper.yang This feature introduces two new keywords in the system health check metrics use-case: • asic-reset • packet-drop

System Health Check use-cases are an enhanced version of the system health check where you can monitor NPU traffic related counters. This service helps to monitor, track and analyze these metrics to detect failures in the system caused by:

- ASIC resets
- packet drops

The system health can be gauged with the values reported by these metrics when the configured threshold values exceed or are nearing the threshold value. This feature determines the Packet forwarding state inside the router, and the data is collected and plotted with respect to time to determine if there are any failures that can affect the packet forwarding state of the router. When seen from the device health point of view, it is conceptually determining and analyzing metrics that detect anomalies in the router. When the metric degrades beyond a certain threshold, the router itself raises the alarm.

Once enabled, it collects metrics from Syslogs, NPU traps, and NPU packet counters. It then analyses the raw data per metric and transforms them into actionable metrics. It then correlates the metrics based on the use case and if all conditions are met, it reports the event as a gray-failure. The user can use can then take action and troubleshoot as required.

System Health check and use-cases are not part of the base package and you must explicitly install the '*xr-healthcheck'* optional package to use this service.

Note In Health check use-cases, packet drop is calculated as the total Bytes egressing the NPU subtracted by total bytes ingressing the NPU. If the traffic arriving on the npu via physical ports/interfaces is less than the Inter-fabric traffic on the Cisco 8000 Distributed platform then the trends will not be seen. This will be updated in a future.

Feature Behavior and Guidelines

- Feature drops such as ACL and QoS are also treated as NPU drops.
- Packet replicating features like Multicast, SPAN, LI, can lead to missed packet drop trends.
- If the Total traffic on a given NPU is less than 10mbps, the trends will not trigger an alarm

Trends Supported by Health Check Use-cases

The use-cases, ASIC resets and packet drops, demonstrate three trends:

1. Peak
2. Plateau
3. Recovery

Peak Trend

A Peak trend is observed when there is a sudden spike in the packet drop or npu traps count and the percentage of this spike is higher than the configured tolerance limit.To verify the configured tolerance limit use the **show healthcheck status** command and see the *drop tolerance* value in the output.

Example:

```
asic_reset
       Drop tolerance: 10

   packet_drop
       Drop tolerance: 10
```

Plateau Trend

A Plateau trend is observed when the packet drop or npu traps count remains higher than the tolerance limit for ten consecutive cadence intervals. Cadence is the time period used by the health check use-cases to examine the data received, transmitted packets and the npu traps per npu.To verify the configured cadence value, use the **show healthcheck status** command and see the *Collector Cadence* value in the output.

Example:

```
 Collector Cadence: 30 seconds
```

Recovery Trend

After Peak or Plateau trend, if the packet drop and npu traps count stays within the tolerance limits for ten cadence intervals, then the recovery trend is seen.

ASIC Reset Use-case and Monitoring

A soft reset of the NPU takes place when pre-determined set of *error-interrupts* occur which are serviced by the NPU-driver. In this case, the recommended action is to reset specific set of blocks inside the ASIC. After reset of the ASIC , the NPU-driver will check if these interruptions occur again within a certain time window. This use-case intends to detect these scenarios and alarm the user that traffic did not recover after the ASIC reset.

This section explains how to check the health of a system ASIC reset information. In this use-case, if the NPU does not recover fully from a soft reset and traffic gets dropped, it gets detected and an alarm is triggered.

Configure ASIC Reset Monitoring

Configure the use-case asic-reset drop tolerance threshold. If the traffic input is below the configured threshold, an alarm is triggered.

```
Router(config)# healthcheck
Router(config-healthcheck)# use-case asic-reset drop-tolerance 10
Router(config-healthcheck)# enable
```

Note You can re-configure the drop-tolerance based on the expected drops in your network.

Retrieve ASIC Reset Trends

This example shows how to obtain the status for the asic-reset use-case:

```
Router# show healthcheck use-case asic-reset detail npu all location all
Mon Nov 29 05:06:51.240 UTS
Node: 0/0/CPU0             NPU Instance: 0
 Timestamp: Mon 2021-11-29 05:06:29 GMT
Alerts     :
0. asic reset for NPU 0 location 0/0/CPU0 triggered at Dec 10 2020 10:52:34
1. peak detected in queue drops for NPU 0 location 0/0/CPU1

Node: 0/0/CPU0             NPU Instance: 1
 Timestamp: Mon 2021-11-29 05:06:29 GMT
Alerts     :
0. asic reset for NPU 0 location 0/0/CPU0 triggered at Dec 10 2020 10:52:34
1. peak detected in queue drops for NPU 0 location 0/0/CPU1
```

Show Command Examples for Asic Reset Use-case

Detailed show outputs of this use-case listed below shows the trend of the packet-drops and the time-stamp when asic reset was triggered. If the asic reset stopped the packet drop then the packet-drop trend moves to Recovery and the status is shown as normal.

Initial show output when no asic-reset is triggered:

```
Router# show healthcheck use-case asic-reset summary
Mon Jun  5 11:13:51.901 IST
Use Case Name: asic_reset
Timestamp    : Mon 2023-06-05 11:13:51 IST
State        : Normal
```

Syslogs when asic-reset is triggered

```
 ----------------------------------------------
Router:Jun  5 11:17:31.664 IST: npu_drvr[299]: %FABRIC-NPU_DRVR-3-ASIC_ERROR_ACTION : [8698]

: npu[0]: HARD_RESET needed for hmc_cgm.cgm_int.total_buffers_underflow
Router:Jun  5 11:19:17.883 IST: NOSi[66650]: %OS-SYSLOG-6-DOCKER_APP : 2023-06-05 11:19:17,882
 [WARNING ] NOSi: 2023-06-05 11:19:01 0/1/CPU0 {NPU:0} :PACKET DROP ALERT.
Waiting for dropped packets to recover post asic reset.
```

Show output when asic-reset is triggered

```
Router# show healthcheck use-case asic-reset summary
Mon Jun  5 11:20:28.307 IST
Use Case Name: asic_reset
Timestamp    : Mon 2023-06-05 11:20:02 IST
State        : Warning
Alert        : Usecase asic_reset has warnings and alerts


 Router# show healthcheck use-case asic-reset detail npu all location all
Mon Jun  5 11:20:35.330 IST
Node: 0/1/CPU0     NPU Instance: 0
    Timestamp : Mon 2023-06-05 11:20:02 IST
    Alerts    :
        0. packet-counters location: node-name: 0/1/CPU0 npu-id: 0 trend: PLATEAU at
2023-06-05 11:20:02
        1. syslog location: node-name: 0/1/CPU0 npu-id: 0 event at 2023-06-05 11:17:31
```

Syslog when packet drops recovered after asic-reset:

```
Router:Jun  5 11:28:18.551 IST: NOSi[66650]: %OS-SYSLOG-6-DOCKER_APP : 2023-06-05
11:28:18,550 [INFO    ] NOSi: 2023-06-05 11:28:02 0/1/CPU0 {NPU:0} :PACKET DROP ALERT
CLEARED.
Dropped packet counters within tolerance limits post asic reset
```

Show commands after packet drops recovered

```
Router# show healthcheck use-case asic-reset summary
Mon Jun  5 11:28:47.961 IST
Use Case Name: asic_reset
Timestamp    : Mon 2023-06-05 11:28:02 IST
State        : Normal
```

```
Router# show healthcheck use-case asic-reset detail npu all location all
Mon Jun  5 11:28:54.111 IST
Node: 0/1/CPU0     NPU Instance: 0
    Timestamp : Mon 2023-06-05 11:28:02 IST
    Alerts    :
        0. syslog location: node-name: 0/1/CPU0 npu-id: 0 event at 2023-06-05 11:17:31
        1. packet-counters location: node-name: 0/1/CPU0 npu-id: 0 trend: RECOVERY at
2023-06-05 11:28:02
```

Packet Drop Use-case and Monitoring

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
Improved Packet Loss Detection and Monitoring	Release 24.1.1	You can now set the tolerance to monitor packet drops in the Network Processing Unit (NPU) for three different NPU trap categories at a configurable cadence of your choice. When a NPU trap breaks the configured packet-drop tolerance for that trap category, the router alerts you with a system log message and you can monitor the trend using healthcheck show commands. This enables you to prioritize taking action depending on the trap category for which the router logs the message. In earlier releases, you could only monitor packet-drops globally for all NPU trap cateogories at a fixed cadence. This feature introduces these changes: **CLI:** The **tolerance** and **window-size** keywords are introduced in the **use-case** command. **YANG Data Model:** New xpaths for `Cisco-IOS-XR-healthcheck-cfg.yang` data model. (see GitHub, YANG Data Models Navigator)

NPU traps are signals that the NPU raises in response to certain types of packets received by the router, such as errored packets, packets that will be dropped by the router, or packets that require extra processing by the CPU.

Packet-drop use-case checks the health of the system by monitoring NPU traps to check for packets dropped per NPU for the cadence interval. You can enable the packet-drop use-case using the **use-case packet-drop** command, In this use-case, you can monitor packet-drops in the NPU by configuring a global drop-tolerance value for all NPU traps for a fixed number of cadence intervals.

From Cisco IOS XR Release 24.1.1 onwards, instead of configuring a global drop-tolerance, you can configure separate drop-tolerance values for low, medium and high tolerance NPU traps. You can also configure the number of cadence intervals to alert you of packet drops.

If the packet-drops exceed configured drop-tolerance rates and continues over the set cadence interval, the router software detects it and generates a system log message.

Apart from this, based on the tolerance limit configured the router software also shows the system health trends in the output of the **show healthcheck use-case packet-drop detail npu all location all** command.

Configure Packet-Drop Monitoring

Prior to Cisco IOS XR Release 24.1.1, you could set the **drop-tolerance** threshold as shown in the following code-block.

```
Router(config)# healthcheck
Router(config-healthcheck)# use-case packet-drop drop-tolerance 10
Router(config-healthcheck)# enable
```

From Cisco IOS XR Release 24.1.1 onwards, you must configure separate drop-tolerance values for low, medium or high tolerance NPU traps. You can also configure the cadence interval.

1. **Check NPU traps tolerance:** Execute the **show controllers npu stats traps-all detail instance all location all** command to check whether an NPU trap is low, medium or high tolerance. The **TOL** column in the following show command output shows the tolerance levels for the different NPU traps.

```
Router# show controllers npu stats traps-all detail instance all location all

Trap stats for 0/RP0/CPU0
Traps marked (D*) are punted (post policing) to the local CPU internal VLAN 1586 for
debugging
They can be read using "show captured packets traps" CLI
Traps marked (D) are dropped in the NPU
TOL represent tolerance level for healthcheck monitoring
 - TOL H/M/L represent High/Medium/Low tolerance
Traps punted to internal VLAN 1538 are processed by the process "spp" on the "Punt Dest"
 CPU
They can also be read using "show captured packets traps" CLI
"Configured Rate" is the rate configured by user (or default setting) in pps at the LC
level
"Hardware Rate" is the actual rate in effect after hardware adjustments
Policer Level:
        NPU: Trap meter is setup per NPU in packets per second
        IFG: Trap meter is setup at every IFG in bits per second
        The per IFG meter is converted from the user configured/default rate (pps)
        based on the "Avg-Pkt Size" into bps.
        Due to hardware adjustments, the "Configured Rate" and
        "Hardware Rate" differ in values.


NOTE:The displayed stats are NOT real-time and are updated every 30 SECONDS from the
hardware.

Trap Type                                     NPU  Trap  Punt       Punt  Punt  Punt
Configured Hardware   Policer Avg-Pkt TOL Packets              Packets
                                              ID   ID    Dest       VoQ   VLAN  TC
Rate(pps)  Rate(pps)  Level   Size        Accepted             Dropped
=======================================================================================
ARP                                          0    3    LC_CPU      311   1538  7    542
```

```
        537        IFG      1520     -  18                  0
NOT_MY_MAC(D*)                                0    4    LC_CPU      304   1586  0    67
         132        IFG      64       -  0                    433944239
ONLINE_DIAG                                   0    32   LC_CPU      311   1538  7
8038660    7898712    IFG      64       -  142298               0
UNKNOWN_VLAN_OR_BUNDLE_MEMBER(D*)             0    42   LC_CPU      304   1586  0    67
         132        IFG      64       M  0                    85722172
V4_CHECKSUM_ERROR(D*)                         0    67   LC_CPU      304   1586  0    67
         132        IFG      64       L  0                    38696396
....
```

2. **Configure tolerance and window-size:** Specify the NPU-trap tolerance level and configure the drop-tolerance using the **tolerance** keyword. Configure cadence intervals to alert you of packet-drops using the **window-size** keyword, as shown below:

```
Router# conf t
Router(config)# healthcheck
Router(config-healthcheck)# use-case packet-drop tolerance high 100
Router(config-healthcheck)# use-case packet-drop window-size 5
Router(config-healthcheck)# enable
Router(config-healthcheck)# commit
```

Retrieve Packet-Drop Trends

The following example shows how to obtain the packet-drop use-case trends prior to Cisco IOS XR Release 24.1.1:

```
Router# show healthcheck use-case packet-drop detail npu all location all

Node: 0_0_CPU0     NPU Instance: 0
    Timestamp : Thu 2021-09-02 21:41:00 UTC
    Alerts    :
        0. npu-traps-sum location: node-name: 0/0/CPU0 npu-id: 0 trend: PEAK at 2021-09-02
 21:41:00
       1. packet-counters location: node-name: 0/0/CPU0 npu-id: 0 trend: PEAK at 2021-09-02
 21:40:51
Node: 0_1_CPU0     NPU Instance: 2
    Timestamp : Thu 2021-09-02 21:42:30 UTC
    Alerts    :
```

The following example shows the packet-drop use-case trends from Cisco IOS XR Release 24.1.1 onwards:

```
Router# show healthcheck use-case packet-drop detail npu all location all
Node: 0/3/CPU0     NPU Instance: 0
    Timestamp : Thu 2024-02-08 17:43:45 UTC
    Alerts    :
        1. npu-traps location: node-name: 0/3/CPU0 npu-id: 0 trap-string:
UNKNOWN_VLAN_OR_BUNDLE_MEMBER(D*) trend: RECOVERY at 2024-02-08 17:43:45
Node: 0/3/CPU0     NPU Instance: 0
    Timestamp : Fri 2024-02-09 04:12:25 UTC
    Alerts    :
        1. npu-traps location: node-name: 0/3/CPU0 npu-id: 0 trap-string:
V4_CHECKSUM_ERROR(D*) trend: PLATEAU at 2024-02-09 04:12:22
```

Show Command Examples for Packet Drops Use-case

Show outputs of this use-case listed below shows the trend of the packet-drops use-case.

Initial show output when no packet-drop is seen

```
Router# show healthcheck use-case packet-drop summary
Mon Jun  5 10:38:25.885 IST
Use Case Name: packet_drop
```

```
Timestamp    : Mon 2023-06-05 10:38:25 IST
State        : Normal
```

Syslog when peak trend is seen:

```
Router:Jun  5 10:59:19.164 IST: NOSi[66650]: %OS-SYSLOG-6-DOCKER_APP : 2023-06-05 10:59:19,164
 [WARNING ] NOSi: 2023-06-05
10:59:18 0/1/CPU0 {NPU:0} :PACKET DROP ALERT.
Dropped packets shows PEAK trend due to PEAK trend observed for NPU traps
```

Show command output when Peak trend is seen

```
Router# show healthcheck use-case packet-drop summary
Mon Jun  5 10:59:30.800 IST
Use Case Name: packet_drop
Timestamp    : Mon 2023-06-05 10:59:18 IST
State        : Warning
Alert        : Usecase packet_drop has warnings and alerts
```

```
Router# show healthcheck use-case packet-drop detail npu all location all
Mon Jun  5 10:59:35.820 IST
Node: 0/1/CPU0     NPU Instance: 0
    Timestamp : Mon 2023-06-05 10:59:18 IST
    Alerts    :
        0. npu-traps-sum location: node-name: 0/1/CPU0 npu-id: 0 trend: PEAK at 2023-06-05
 10:59:18
       1. packet-counters location: node-name: 0/1/CPU0 npu-id: 0 trend: PEAK at 2023-06-05
 10:59:02
```

Syslog when Plateau trend is seen

```
Router:Jun  5 11:03:49.417 IST: NOSi[66650]: %OS-SYSLOG-6-DOCKER_APP : 2023-06-05 11:03:49,416
 [WARNING ] NOSi: 2023-06-05
11:03:48 0/1/CPU0 {NPU:0} :PACKET DROP ALERT.
Dropped packets shows PLATEAU trend due to PLATEAU trend observed for NPU traps
```

Show command output when Plateau trend is seen

```
Router# show healthcheck use-case packet-drop summary
Mon Jun  5 11:05:34.428 IST
Use Case Name: packet_drop
Timestamp    : Mon 2023-06-05 11:03:48 IST
State        : Warning
Alert        : Usecase packet_drop has warnings and alerts
```

```
Router# show healthcheck use-case packet-drop detail npu all location all
Mon Jun  5 11:05:39.720 IST
Node: 0/1/CPU0     NPU Instance: 0
    Timestamp : Mon 2023-06-05 11:03:48 IST
    Alerts    :
       0. npu-traps-sum location: node-name: 0/1/CPU0 npu-id: 0 trend: PLATEAU at 2023-06-05
 11:03:48
        1. packet-counters location: node-name: 0/1/CPU0 npu-id: 0 trend: PLATEAU at
2023-06-05 11:03:42
```

Syslog when Recovery trend is seen

```
Router:Jun  5 11:11:49.866 IST: NOSi[66650]: %OS-SYSLOG-6-DOCKER_APP : 2023-06-05
 11:11:49,865 [INFO    ] NOSi: 2023-06-05 11:11:49 0/1/CPU0 {NPU:0} :PACKET DROP ALERT
CLEARED.
NPU trap and dropped packet counters within tolerance limits
```

Show command output when Recovery trend is seen

```
Router# show healthcheck use-case packet-drop summary
Mon Jun  5 11:12:59.387 IST
Use Case Name: packet_drop
```

```
Timestamp     : Mon 2023-06-05 11:11:49 IST
State         : Normal


Router# show healthcheck use-case packet-drop detail npu all location all
Mon Jun  5 11:13:05.155 IST
Node: 0/1/CPU0     NPU Instance: 0
    Timestamp : Mon 2023-06-05 11:11:49 IST
    Alerts    :
       0. npu-traps-sum location: node-name: 0/1/CPU0 npu-id: 0 trend: RECOVERY at 2023-06-05
 11:11:49
        1. packet-counters location: node-name: 0/1/CPU0 npu-id: 0 trend: RECOVERY at
2023-06-05 11:11:42
```

Retrieval of Data

For the purpose of analyzing the metrics or troubleshooting once an alarm is raised, you can retrieve the data. The data can be retrieved using CLI. You can use the following show commands to retrieve the data:

- **show healthcheck use-case asic-reset detail npu all location all**
- **show healthcheck use-case packet-drop detail npu all location all**

The service supports NETCONF client to retrieve operational data (*Cisco-IOS-XR-ofa-npu-stats-oper.yang* and *Cisco-IOS-XR-infra-syslog-oper.yang*) using YANG data models.

It also supports subscribing to metrics and their reports to stream telemetry data. For more information about streaming telemetry data, see *Telemetry Configuration Guide for Cisco 8000 Series Routers*. You can also view the data model definitions using the YANG Data Models Navigator tool.