Common Monitoring Metrics & Reporting using ITNM

*IBM Tivoli Network Manager - IP Edition (ITNM)*
- v3.9 & v4.1.1

Krishna M Kodali - kmkodali@us.ibm.com

April 28th - 2015
Contents:

1. Target audience – Enterprise customers.

2. ITNM Monitoring
   2.a – ITNM Architecture
   2.b – ITNM Monitoring & Implementation

3. Common Monitoring metrics
   3.a List of Monitoring metrics
      - ICMP
      - SNMP
   3.b Reports based on monitoring metrics
   3.c Validating monitoring policies
      - Using itnm_poller.pl
      - Run OOTB reports
1. Target Audience – Enterprise customers:

This document is an answer to a question about:

What are the common monitoring metrics typically any customer would want to collect when using ITNM as part of their Network Monitoring Solution?

In general, one shouldn't write such document as the requirements typically come from Network Administrators of NoC group. The Application admins (for e.g. ITNM Administrator) then to setup metrics to meet those requirements. This document is intended for guidance purposes only with assumption that the reader has basic understanding of ITNM usage and its functionality.

This document is written based on the study from twelve enterprise customers from various Geographic regions.

2. ITNM Monitoring

ITNM – High level Architecture & Common integration
Brief about ITNM monitoring:

ITNM Monitoring will raise events when problems are detected in the network, and will clear them if they become resolved. Events retrieved from the Object Server for event enrichment and RCA (Root Cause Analysis) might have been generated by ITNM, or could come from external sources (e.g. Syslog, mttrapd probes etc).

ITNM monitoring is performed by a process called 'Poller' and it performs SNMP and ICMP (ping) polling of the discovered network.

ITNM Poller Inputs:

- Poller reads topology from NCIM database
- Processes topology updates by listening to broadcast of ncp_model process.
- Sends packets to devices, processes results

ITNM Poller Outputs:

- Raises events via monitor Probe i.e. 'nco_p_ncpmonitor'
- The monitor probe provides the interface for passing events to the Object Server.

Capabilities of ITNM Poller:

The Poller uses 'Monitors'. There are four types of monitors:

A) Ping - ICMP
B) SNMP Link State
C) SNMP Threshold
   - Generic Threshold
   - Basic Threshold
D) Remote Ping
   - Cisco Remote Ping
   - Juniper Remote Ping

ICMP (ping) Polling:

Poll policies using the Chassis Ping & End Node Ping template type will generate ICMP packets sent to the main node IP address.

Poll policies using the Interface Ping template type will generate ICMP packets sent to the IP address of each interface which has an IP address.
SNMP Link State:

The SNMP Link State is hard-coded to check both `ifOperStatus` and `ifAdminStatus` of interfaces. Events will be raised based on the previous and current values of these MIB variables, as shown in the table below.

<table>
<thead>
<tr>
<th>ifAdminStatus</th>
<th>ifOperStatus</th>
<th>Event raised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remains 1</td>
<td>Transition from 1 to 2</td>
<td>Fail - iface has gone down</td>
</tr>
<tr>
<td>Remains 1</td>
<td>Transition from 2 to 1</td>
<td>Clear - iface has gone up</td>
</tr>
<tr>
<td>Transition from 1 to 2</td>
<td>Transition from 2 to 1</td>
<td>Clear - iface has gone up (the adminStatus says it should be down)</td>
</tr>
<tr>
<td>Transition from 1 to 2</td>
<td>Remains 2</td>
<td>Clear - iface confirmed down</td>
</tr>
<tr>
<td>Transition from 2 to 1</td>
<td>Transition from 1 to 2</td>
<td>Fail - iface has gone down</td>
</tr>
<tr>
<td>Transition from 2 to 1</td>
<td>Remains 2</td>
<td>Fail - iface instructed to go up but didn't</td>
</tr>
</tbody>
</table>

Snmp Threshold:

The information retrieved by each of the Generic Threshold templates is specific to that template. In the GUI, this is a combination of the Triggers and the Descriptions.

The Generate Trigger and Clear Trigger define the thresholds that must be crossed for events to be respectively raised and cleared. If no Clear Trigger is defined, then any raised event will be cleared when the Generate Trigger threshold is no longer satisfied. (ie - If absent, the Clear Trigger is implicitly defined as the inverse of the Generate Trigger.)

At each polling interval, all MIB variables required to calculate both thresholds will be retrieved. This will frequently involve walks of table MIB values, using SNMP GetNext requests.

When a threshold is crossed for a given device, any further MIB values required to populate the Description will be retrieved in a single SNMP Get request.

Remote Ping:

The Remote Ping polls generate both SNMP Get and SNMP Set packets. The latter initiate the remote ping operation from the PE to the CE devices. The former subsequently return the results.
Poller Implementation:

- Monitor implementation is done using Poll definitions and Poll Policies.
- Once can setup Poll definitions and Poll policies via GUI under Administration → Network Polling within TIP navigation tree.

Poll Policies:

- When to poll & Which specific devices
- Device filter & Network Views
- Frequency

Poll Definitions:

- How to poll, what to Poll
- Device Classes
- Interface filter
- Threshold definition
- MIB objects
- Event information
- Event Identifier.
- Severity

3. Common Monitoring Metrics:

Out of the box, ITNM has many and most commonly used monitoring policies without needing much of customisation:

Below are list of Common Monitoring collections:

i) ICMP:
   1.a) Default Chassis Ping – For ICMP polling of Network devices
   1.b) Default Interface Ping – For ICMP polling of Network Interfaces
   1.c) End Node Ping – For ICMP polling of End Nodes (e.g. Servers etc.)
One can setup a single policy with multiple poll definitions to do entire ICMP monitoring.

ii) **Adaptive Polling**:

- This method is used to confirm device/interface ICMP failures using 'Adaptive Polling' without having to wait for 2nd Poll cycle. You can read step by step instructions @ https://ibm.biz/BdXiMe

iii) **SNMP Link State**

- Some customers would like to poll only Uplinks instead of all connected interfaces. Ideally all connected interfaces should be polled for link state alerts. The recommendation is to monitor every link that has connectivity.

In v3.9 following logic can be added to unmanage all ethernet interfaces without any connectivity in $NCHOME/precision/disco/stitchers/TagManagedEntities.stch

(ExtraInfo->m_IfType = 6 and (ExtraInfo->ReverseRelatedTo is NULL and RelatedTo is NULL))

Below is excerpt from TagManagedEntities.stch:

```c
// Modify this statement to unmanage specific types of interface
//
// oq1Update = oq1Update + interfaceFilter + rediscoveryFilter +
// * AND
//
// ExtraInfo->m_IfDescr like 'Dialer'
// OR
// ExtraInfo->m_IfDescr like 'BRI'
// OR
// ExtraInfo->m_IfDescr like 'Async'
// OR
// ExtraInfo->m_IfDescr like 'Virtual'
// OR
ExtraInfo->m_IfType = 6 and (ExtraInfo->ReverseRelatedTo is NULL and RelatedTo is NULL)
// OR
ExtraInfo->m_IfDescr like 'NULL'
```

Unmanage all Ethernet Interfaces that has no Connectivity
In v4.1.1 - similar logic goes under:
$NCHOME/precision/disco/stitchers/DNCIM/PopulateDNCIM_ManagedStatus.stch

iv) **Snmp Threshold:**
- Basic Threshold
- Generic Threshold

ITNM strength is its openness in terms of able to configure any monitoring needs without having to know the product in depth.

Below are some of key common poll policies customers would go as out of the box monitoring features:

4.a) **Bandwidth utilization** *(snmpInBandWidth & snmpOutBandWidth)*
- Recommendation is to go with *Interface Filter*, this is to limit collecting metrics on key router interfaces rather than entire network.

4.b) **CPU Usage** – For e.g. *cpuBusyPoll* (if using older IOS versions) policy can be replaced by *ciscoCPUTotal* policies (if appropriate) as 'avgbusy5' is deprecated in newer IOS versions.
4.c) Memory Metrics.

For e.g. 'memoryPool' policy can be replaced by ciscoMemoryPool & ciscoMemoryPctgUsage policies based on IOS version you are on:

Following new mib variables are being used to calculate memory usage on the network devices:

![MIB Variables](image)

4.d) Interface inbound or outbound packet discards (ifInDisards & ifOutDiscards)

- Recommendation is is to go with Interface filter to collect metrics on the key interfaces.

4.e) Other common metrics – sysUpTime & rebootDetection

- sysUpTime is a requirement to run few Summarization reports.
4.f) An example of a Generic Threshold to monitor 'Stacked Switch module status' that isn't part of Out-of-the-box policy:

Say you got Stacked switches and would like to monitor status of those modules, one could create a simple threshold like below:

<table>
<thead>
<tr>
<th>Specific Object Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
</tr>
<tr>
<td>OID</td>
</tr>
<tr>
<td>Type</td>
</tr>
<tr>
<td>Permission</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Values</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

MIB: CISCO-STACK-MIB: - View Supporting Images

Description:
"The operational status of the module. If the status is not ok, the value of module TestResult gives more detailed information about the module's failure condition(s)."

**Event Trigger Threshold**

```
( eval(int,'SNMP.VALUE.moduleStatus') <> 2 )
```

**Event Description:**

Catalyst Module status is not OK

**Clear Event Threshold**

```
( eval(int,'SNMP.VALUE.moduleStatus') = 2 )
```

**Event Description:**

Catalyst Module in OK status
4.g) Another Generic Threshold policy to monitor 'virtual private dialup network (VPDN)' tunnels:

Note: These examples (4.f & 4.g) are to illustrate how easily one could setup monitoring based on their specific requirements

Below is trigger threshold:
Below is clear threshold:

![Poll Definition Editor - Windows Internet Explorer](image)

**v) Remote Ping:**

- Cisco Remote Ping
- Juniper Remote Ping

Remote Ping requires SNMP **write** access to target devices
3.b) Reports associated with Poll policies

Below are some of key and common reports that rely on polling data (either from local NCIM database or integrated with TDW):

<table>
<thead>
<tr>
<th>S.No</th>
<th>Group</th>
<th>Name of the Report</th>
<th>Purpose of the Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Context Reports</td>
<td>Bandwidth Utilization</td>
<td>Bandwidth usage for selected interfaces</td>
</tr>
<tr>
<td></td>
<td></td>
<td>snmpInBandwidth</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Generic Trend Analysis</td>
<td>Trend of the discarded packets of an interface on a device</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ifInDiscards</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>memoryPoll</td>
<td>Memory trending</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cpuBusyPoll</td>
<td>CPU trending</td>
</tr>
<tr>
<td></td>
<td>Router Health Summary</td>
<td>ciscoMemoryPctUsage, cpuBusyPoll and Chassis Ping</td>
<td>Monitor recent changes of router that can be used to identify bottleneck in the network.</td>
</tr>
<tr>
<td>2</td>
<td>Performance Reports</td>
<td>Device Availability Summarization</td>
<td>Device availability data collected and summarized over the last 7 days.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chassis Ping</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Interface Availability Summarization</td>
<td>snmpInBandwidth</td>
<td>General health trends for inbound traffic on interfaces using data from ifInErrors, ifInDiscards, and bandwidth.</td>
</tr>
<tr>
<td></td>
<td>Interfaces Summary</td>
<td>memoryPoll</td>
<td></td>
</tr>
<tr>
<td></td>
<td>System Availability Summary</td>
<td>sysUpTime</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Summary Reports</td>
<td>Device Availability Summary</td>
<td>chasis ping &amp; sysUpTime</td>
</tr>
<tr>
<td></td>
<td></td>
<td>snmpInBandwidth</td>
<td>General health trends for outbound traffic on interfaces using data from ifOutErrors, ifOutDiscards, and bandwidth.</td>
</tr>
<tr>
<td></td>
<td>Device Egress Traffic Health Summary</td>
<td>memoryPoll</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>snmpOutBandwidth, iOutErrors, &amp; iOutDiscards</td>
<td>General health trends for inbound traffic on interfaces using data from ifInErrors, ifInDiscards, and bandwidth.</td>
</tr>
<tr>
<td></td>
<td>Device Ingress Traffic Health Summary</td>
<td>memoryPoll</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>snmpOutBandwidth</td>
<td>General health trends for outbound traffic on interfaces using data from ifOutErrors, ifOutDiscards, and bandwidth.</td>
</tr>
<tr>
<td></td>
<td>Router Health Summary</td>
<td>ciscoMemoryPctUsage, cpuBusyPoll, &amp; Chassis Ping</td>
<td>Monitor recent changes of router that can be used to identify bottleneck in the network.</td>
</tr>
</tbody>
</table>

It would be ideal to setup a policy exclusive for 'Router Health Summary Report' based on its dependency i.e. the report relies on following three poll definitions

a) ciscoMemoryPctUsage, b) cpuBusyPoll and c) Chassis Ping
Below is an example of Context Report (Router Health Summary) for a Juniper Router:
Above two screen shot excerpts are from a 'Router Health Summary' report.

3.c Validating ITNM monitoring policies
   - itnm_poller.pl script:
     You can validate whether a given poll policy is polling a device or an interface using itnm_poller.pl script. You can read more about the script and its usage @ https://ibm.biz/BdXiVJ
   - Using Out-of-the-box reports:
     Below three reports are to check overall health of ITNM monitoring system and then able to find granular details of a given policy and then to a device and interface level.
Below are screen shot excerpts from 'Monitoring Summary' report for domain: **IBM_NETWORK**

### Network Manager: Monitoring Summary

**Domain Name**: IBM_NETWORK

**First Poll and Last Poll**

<table>
<thead>
<tr>
<th>First Poll</th>
<th>Last Poll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apr 27, 2015 2:55 PM</td>
<td>Apr 28, 2015 11:15 AM</td>
</tr>
</tbody>
</table>

#### pollData Table Utilization

![Poll Data Table Utilization](image)

#### Configured Template and Policy Summary

<table>
<thead>
<tr>
<th>Template Type</th>
<th>Event Name</th>
<th>Severity</th>
<th>Policies with Data Storage</th>
<th>Template Name</th>
<th>Frequency (s)</th>
<th>Share Data</th>
<th>Poller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Threshold</td>
<td>HostDownCpuMemory</td>
<td>1</td>
<td>Router_Name:Summary</td>
<td>CiscoMemory$81276</td>
<td>109</td>
<td>Stored</td>
<td>Default Poller</td>
</tr>
<tr>
<td></td>
<td>InboundBandwidth</td>
<td>0</td>
<td></td>
<td>CiscoMemory$81276</td>
<td>109</td>
<td>Stored</td>
<td>Default Poller</td>
</tr>
<tr>
<td>Cluster Ping</td>
<td>HostDownPingPage</td>
<td>0</td>
<td>ICMP_A</td>
<td>Default Cluster Ping</td>
<td>109</td>
<td>Not Stored</td>
<td>Default Poller</td>
</tr>
<tr>
<td></td>
<td>HostDownPingFail</td>
<td>0</td>
<td>ICMP_A</td>
<td>Default Cluster Ping</td>
<td>109</td>
<td>Not Stored</td>
<td>Default Poller</td>
</tr>
<tr>
<td>Interface Ping</td>
<td>HostDownPingPage</td>
<td>0</td>
<td>Router_Name:Summary</td>
<td>Default Cluster Ping</td>
<td>109</td>
<td>Not Stored</td>
<td>Default Poller</td>
</tr>
<tr>
<td></td>
<td>HostDownPingFail</td>
<td>0</td>
<td>ICMP_A</td>
<td>Default Cluster Ping</td>
<td>109</td>
<td>Not Stored</td>
<td>Default Poller</td>
</tr>
</tbody>
</table>

**Policies Monitored Per Device**: 0

**Devices Not Being Monitored**: 0

Display summary information about Network Manager monitoring policies, polling, configurations and database utilization.

Reference material @ [https://ibm.biz/BdXiAS](https://ibm.biz/BdXiAS)