INTERNET-DRAFT Haresh Khandelwal & Sushrut Deshpande
Intended Status: Experimental Cisco Systems
Expires: March 2012

SNMPD to use cache and shared database based on MIB Classification
draft-haresh-sushrut-mib-classification-01

Abstract

This memo defines classification of SNMP MIBs to either use SNMP cache database and shared database (SDB) mechanism to reduce high CPU usage while SNMP GET REQUEST, GETNEXT REQUEST, GETBULK REQUEST are continuously performed from network management system (NMS)/SNMP manager/SNMP MIB browser to managed device.

Table of Contents

1 Introduction

Continuous GET REQUEST, GETNEXT REQUEST, GETBULK REQUEST on managed device results into high CPU usage. High CPU usage is result of high process interactions between SNMP process and requested OID's process when OID came from GET REQUEST, GETNEXT REQUEST, GETBULK REQUEST. This draft suggests the way to reduce these process interactions in order to reduce CPU usage. This approach also suggests way to reduce high CPU usage with accurate OID values.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. MIB classification

2.1 How CPU usage goes high

When SNMP protocol data units (SNMP PDU) are received by SNMPD running on managed device (router, switch etc...), SNMPD de capsulate PDU and retrieves requested ODI from PDU. After getting OIDs, SNMPD interacts with process under whom requested OIDs and its value reside.

For example, NMS sends SNMP GET REQUEST with seeking of interface operational status whose OID is .1.3.6.1.2.1.2.2.1.8. This GET REQUEST PDU is received at managed device's SNMPD which in turn de capsulated PDU and find out that "ifOperStatus" is requested from NMS. SNMPD then interacts with process who handles interface process and gets OID value and builds RESPONSE PDU which sent to NMS.

Like as when NMS keeps polling managed device, it leads very high process interactions between SNMPD and requested OID's. generally, process interactions happen through common mechanism like MTS (message transaction service) in NX-OS. processes talk with each other using such messages. so, when processes have to talk with each other frequently, transactions of interaction messages also go very high. such high amount of transactions increase a CPU usage of the system which in turns result into very high CPU usage. SNMP polling is such a method which also leads to high CPU usage of managed device. generally service providers keep monitoring network devices using SNMP for all time and running other application parallel to monitoring it. so if SNMP polling consume higher CPU, it may lead other processes to CPU starvation which may lead device to strange unknown behavior. To avoid this, we are proposing this solution where

we reduce process interactions, so we can achieve less message transactions and less CPU consumption.

2.2 MIB Classification

MIB Classification provides a solution to avoid high CPU usage. MIB are classified into two categories.

i) Dynamic MIBs - Whose values change frequently (for example ifMIB counters).

ii) Relatively Static MIBs - Whose values change occasionally (for example VLAN MIB).

2.3 How SNMP CACHE and SHARED DATABASE works

SHARED DATABASE (SDB) is a database where component or process who falls under "Dynamic MIBs" has to populate their supported ODIs values. SNMPD will fetch ODI's value from this SHARED DATABASE when ODI belongs to "Dynamic MIBs" category requested from NMS. Processes taking part in SDB are completely responsible for OIDs values. So, if it is ifMIB counter process for interface then it will be responsible for interfaces counter values in SDB. SNMPD will pickup relevant values for asked OIDs from SDB and will response back to MIB browser/NMS.

SNMP CACHING is a database which is maintained by SNMPD of managed device. This database contains ODIs values of "Relatively Static MIBs". These ODI values filled to SNMP CACHE when SNMPD receives ODI's value first time from requested OID's process.SNMPD will form its own cache table which in turn, stores values of all processed ODIs. So when next time same OID will be requested, SNMPD can response from its own Cache table. Now, if value of OID from "Relatively Static MIBs" changes, that process or component has to inform SNMPD regarding its event with new values. SNMPD will update its cache table with new ODI value.

SNMP cache table can be flushed in event of SNMP process restart/crash/enable-disable. New cache table will be formed again after recovery with 1st poll cycle.

SHARED DATABASE will be handled by individual process and should be populated again in event of process crash/restart.

2.4 How CPU usage reduced

For NMS queries with MIB classification approach, SNMPD does not need to talk with individual processes. According to MIB category, SNMP

process will fetch ODIs value from either of these two databases.so, by this way, we can reduce the interaction messages between SNMPD and other processes and still having accurate OID values from accurately maintained database. As high process interactions are reduced, it will reduce CPU usage also.

3 Security Considerations

This design is not changing SNMP packets. It does not apply on SNMP SET operation. Communication between NMS and managed device is also un changed, only internal process interaction changes are proposed based on MIB classification.so, this design does not exhibit any security threat.

4 IANA Considerations

5 References

5.1 Normative References

5.2 Informative References

[RFC2578] McCloghrie, et al., "Structure of Management Information Version 2 (SMIv2)", RFC2578, April 1999.

[RFC3418] Presuhn, et al., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", RFC3418, December 2002.

[RFC3411] Harrington, et al., "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", RFC3411, December 2002.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226,May 2008

[RFC2119] Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC2119, March 1997

Authors' Addresses

Name
Haresh Khandelwal
Sushrut Deshpande

EMail: hkhandel@cisco.com, susdeshp@cisco.com