INTERNET-DRAFT
Intended Status: Experimental
Expires:

Haresh Khandelwal & Sushrut Deshpande
Cisco systems

February 2012

MIB Classification based use of SNMP cache or shared database
draft-haresh-sushrut-mib-classification-00

Abstract

This memo defines classification of MIBs to use SNMP cache or shared database mechanism to reduce high CPU usage while SNMP MIBs are polled or GET operations performed from MIB browser.

Status of this Memo

Copyright and License Notice

Table of Contents

1 Introduction

Continuous SNMP polling or GET operations on managed device results into high CPU usage. This is due to high process interactions between SNMP and requested OID's process. This draft suggests way to reduce process interactions in order to reduce CPU usage.

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

When SNMP PDUs are received by SNMP on managed device (router, switch etc...), SNMP will look into ODI and fetch its value either from cache table or shared data base based on its MIB category.

2.1 MIB Classification

i) Dynamic MIBs - Whose values change frequently (for example ifMIB counters). Value of object falls under this MIB category will be fetched from shared Database by SNMP

ii) Relatively Static MIBs - Whose values change occasionally (for example VLAN MIB). Value of object falls under this MIB category will be fetched from SNMP cache table

2.2 SNMP Cache and Shared database

In case of shared database, component or process who falls under "Dynamic MIBs" have to populate SDB (shared database) for SNMP. This particular process is completely responsible for those SDB table values. So, if it is ifMIB counter process for interface then it will be responsible for interfaces counters value in SDB. SNMP will pickup relevant values for asked OIDs from SDB and will send it to MIB browser.

In case of SNMP Caching Mechanism, component or process who falls under "Relatively Static MIBs", SNMP will interact with those processes when OID requested 1st time and SNMP will form its own cache table which in turn, stores values of all processed ODIs. So when next time same OID will be requested, SNMP can response from its own Cache table. Here, if value of OID from "Relatively Static MIBs" changes, that process or component has to inform SNMP regarding its event and new values. This is similar to internal trap

communications. SNMP will accordingly update its Cache table.

SNMP cache table can be flushed in event of SNMP restart/crash/enable-disable. 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. This way, we can reduce interaction messages between SNMP and other processes and so does CPU usage.

3 Security Considerations

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

4 IANA Considerations

5 References

5.1 Normative References

5.2 Informative References

Authors' Addresses

Name
Haresh Khandelwal
Sushrut Deshpande

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