Bulk MIB Data Transfer Solution Space

Presented by J. Schönwälder on behalf of the

Network Management Research Group (NMRG)

<URL:http://www.irtf.org/charters/management.html>

For additional information:

R. Sprenkels, J. P. Martin-Flatin: Bulk Transfer of MIB Data,
Simple Times 7(1), March 1999
<URL:http://www.simple-times.org/>

Introduction

Definition:

- A bulk MIB data transfer is the transfer of several hundreds of kilobytes of MIB data in a single logical transaction.

Background:

- SNMPv1 was not designed with bulk data transfers in mind.
- Size of MIB tables is constantly increasing.
- Amount of data accessible via SNMP is steadily increasing.
- Improvements by the SNMPv2c/SNMPv3 `get-bulk` operation are insufficient.

Goals:

- Reduce the end-to-end latency for a bulk data transfer.
- Reduce the network overhead.
- Improve the consistency of the retrieved data.
- Simplify the retrieval of large conceptual MIB tables.

Approach #1: Extending SNMP

- Add three new features to the SNMP protocol:
  1. SNMP over TCP transport mapping
  2. SNMP lossless payload compression
  3. `get-subtree` and linked `response` PDUs

Discussion:

- Each feature has value on its own. In combination, they provide a fast pipe to ship SNMP payloads:
  – The SNMP over TCP mapping provides a reliable transport which features flow and congestion control (and most likely segmentation with MTU path discovery).
  – Lossless payload compression reduces the amount of redundant data in SNMP PDUs, which leads to considerable bandwidth savings by reducing the instance specific OID naming overhead.
  – The `get-subtree` PDU is an extension of the `get-next` PDU which iterates over complete subtrees and sends one or more linked `response` PDUs, which greatly reduces latency.
- Solution fits into the administrative framework and the SNMP architecture.

Approach #2: Bulk Transfer MIB

- Layer a bulk transfer mechanism on top of the existing SNMP protocol:
  1. Ordinary MIB objects can be created to initiate and control bulk data transfers.
  2. The actual payload is efficiently encoded in other MIB objects.
  3. The MIB objects that encapsulate the requested data are pushed to the management application via notifications.

Discussion:

- No need to change the SNMP protocol.
- Implementation of the bulk transfer MIB must reside on the master-agent in most sub-agent environments (e.g. AgentX).
- The initiator of a bulk retrieval interaction (a command generator) is not the receiver of the data (a notification receiver).
- Bursty notification traffic may overload notification receivers.
- No flow or congestion control without an SNMP over TCP transport mapping.

Approach #3: Hybrid Solutions

- Combine a bulk transfer control MIB with existing data transfer protocols:
  1. The MIB provides an interface to initiate and control bulk data transfers.
  2. Protocols such as FTP or HTTP are used to push the data from the agent to the management application.

Discussion:

- No need to change the SNMP protocol.
- Implementation of the bulk transfer mechanism must reside on the master-agent in most sub-agent environments (e.g. AgentX).
- Managers and agents must have access to an FTP or HTTP server/client in order to receive/transfer the data.
- Different protocols have a different administrative models and mappings of security configurations (e.g. access control) between them are a potential source of problems.
- Flow and congestion control inherited from TCP (when using FTP or HTTP).

Approach #4: Bulk Transfer without SNMP

- Solve the bulk data transfer problem without SNMP:
  1. Define a new encoding scheme (e.g. XML-based) for MIB data and wrap bulky data in e.g. a MIME envelop.
  2. Any protocol capable to transfer MIME content (e.g. HTTP, SMTP) can be used to transfer MIB data.
  3. Define rules that allow to initiate transfers via special HTTP ports or something equivalent.

Discussion:

- No need to change and use the SNMP protocol.
- Easier access to MIB data from non-SNMP aware applications.
- Different administrative models and mappings of security configurations (e.g. access control) between them are a potential source of problems.
- May lead to a special version of HTTP or another variant of COPS.
- Flow and congestion control inherited from TCP.