Subtree Retrieval MIB

draft-irtf-nmrg-get-subtree-mib-00.txt

http://www.eecs.umich.edu/~thalerd/subtree.txt

Dave Thaler

dthaler@microsoft.com

Problem:

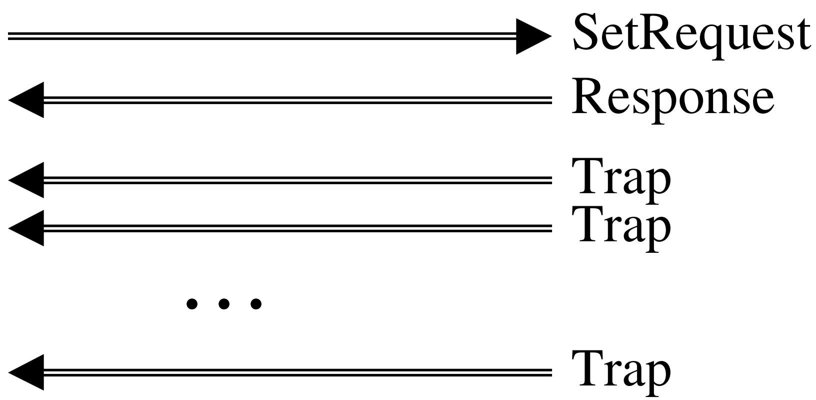
- Bulk subtree(s) retrieval
  - "Overshoot" by guessing max-repetitions too large
  - Or issue sequential requests (larger time window of missing changes, more pkts on wire, etc)
  - Latency
  - etc

Goals:

- Goals in common with new-PDU approach:
  - single "request" message
  - multiple "response" messages, as large as possible
  - make it easy to know what ranges, if any, got dropped (if over udp)
  - allow halting operation in progress (e.g. oid typo)
- Goals in common with FTP approach:
  - **No changes to existing SNMP protocol**
  - **No changes to some master agents** (see Limitations slide later on)

Mechanism:

- "request" is a set (row creations) in a new MIB
- "responses" are traps:



- response destination must be a legal notification target in SNMP-TARGET-MIB
- conceptual row deleted automatically when done
- can cancel operation via row deletion

Limitations:

- Application must be allowed to issue SETs to this MIB
- Command responder/notification originator tightly coupled
- Command generator/notification responder tightly coupled
- Subagent implementing MIB must be able to call back into the SNMP agent to walk other MIBs, without causing a deadlock

Trap contains:

- Read-only objects in “request” row created:
  - seqno (to detect losses)
  - "done" flag
- Bunch of varbinds requested
  - Encoding optimizations possible (same as with other approaches), but not discussed in this document
- App stops waiting for traps when gets “done” and has all seqno’s, or after timeout.

TODO

- Need to update to get multiple subtrees in parallel
  - Initial version didn’t include this just to give the general idea
- Implementation
- Co-author(s)?