

Remote Ping and Traceroute

Kenneth White -
wkenneth@us.ibm.com
IBM Corporation - NCSD

DISMAN-MIB

- Many enterprise solutions, need a standards based solution
- Provide a dedicated MIB for remote ping and traceroute
- 2nd Draft 8/98, <draft-ietf-disman-remops-mib-01.txt>
- GET-RESPONSE approach abandoned due to response time issue
- Results table approach used due to 484 OCTET (or 4K?) SNMP message size limit, also simplify response display
- Future work:
 - ▶ Consider <draft-ietf-idmr-traceroute-ipm-03.txt>
 - ▶ Look at ipv6 implementation

REMOPS-MIB Structure

- 2 notifications, 5 global objects, 4 tables with 21 objects
- Target address can be: DNS name, ipv4 or ipv6 address
- remopsPingTable or remopsTraceRouteTable:
 - ▶ allows for parameter specification
 - ▶ entries created and deleted via RowStatus objects
 - ▶ OperStatus object to indicate status of request
 - ▶ OwnerIndex to enable SNMPv3 VACM usage
 - ▶ CtlType objects determine operation behavior
 - ▶ MaxConcurrentRequests global objects to limit concurrent activity
 - ▶ PingPurgeTime global objects to police ResultTable entries
 - ▶ Questions:
 - Are intermediate NOTIFICATIONS of use?
 - Are DNS name support needed?
 - Would current ttl and probe counts be of use?

DISMAN-SCRIPT-MIB

- 2 NOTIFICATIONS, 6 Tables with 42 objects
- Ping and traceroute would be hardwired:
- Both the smLangTable and smExtsnTable can be ignored
- A smScriptTable entry would need to be created:
 - ▶ smScriptOwner -> owning manager name
 - ▶ smScriptName -> hardwired script name
 - ▶ smScriptDescr -> defaulted
 - ▶ smScriptLanguage -> defaulted to 0
 - ▶ smScriptSource -> zero length octet string
 - ▶ smScriptAdminStatus and OperStatus -> supported
 - ▶ smScriptStorageType -> defaults to nonVolatile
 - ▶ smScriptRowStatus -> supported
 - ▶ Ignore smCodeTable

DISMAN-SCRIPT-MIB (continued)

- Create an smLaunchTable entry:
 - ▶ smLaunchOwner -> owning manager name
 - ▶ smLaunchName -> assign name
 - ▶ smLaunchScriptOwner, smLaunchScriptName -> use names
 - ▶ smLaunchArgument -> parameters, need to define format (The target of a request would be a parameter).
 - ▶ smLaunchMaxRunning, smLaunchMaxCompleted, smLaunchLifeTime, smLaunchExpireTime, smLaunchStart, smLaunchControl, smLaunchAdminStatus, OperStatus, smLaunchRunIndexNext -> support
 - ▶ smLaunchStorageType -> ?
 - ▶ smLaunchRowStatus -> support
- smRunTable entries are created and managed via an smlaunchEntry

Comparison

- The result of a traceroute operation can be too large to return as an octet string, smRunResult doesn't work:
 - ▶ ipv4 or ipv6 address -> 17 octets
 - ▶ DNS name -> 66 octets
 - ▶ Response time -> 4 octets
 - ▶ Misc. -> 1 octet
 - ▶ Total -> $88 \times 255 = 22,440$ octets (5,610 without DNS name support, 2,295 without DNS and ipv6 address support)
- Using smRunResult to return pointer is more complicated
- REMOPS-MIB is simpler than a DISMAN-SCRIPT-MIB approach to implement
- Perhaps it make sense to have both:
 - ▶ hardwired scripts for implementations that support DISMAN-SCRIPT-MIB
 - ▶ REMOPS-MIB on systems where DISMAN-SCRIPT-MIB support isn't needed

Guidelines

- In general when should the script MIB be used versus developing a specific MIB
- Considerations:
 - ▶ availability of "tailored" object-types for control and observations
 - ▶ availability of specialised traps/informs (some management systems don't "demultiplex" different traps based on their varbinds or the contents of those varbinds)
 - ▶ processes whose control may be a multi-step or interactive process, rather than "load-and-go"
 - ▶ others?