EXTENSIBLE AGENTS

FACILITATE THE EXTENSION OF SNMP AGENTS WITH NEW MIB MODULES

• SEPARATE SNMP PROTOCOL ENGINE FROM MIB INSTRUMENTATION

• ALLOW DYNAMIC ADDITION OF NEW MIB MODULE IMPLEMENTATIONS

• EXTENSIBLE AGENTS SHOULD BE TRANSPARENT
BASIC STRUCTURE

- MASTER AGENT
  - PROTOCOL OPERATIONS
  - ENCODING
- SUB AGENT
  - MIB
  - API
- SUB AGENT
  - MIB
  - API
- SUB AGENT
  - MIB
  - API
- IPC / PROTOCOL
- TRANSPORT
SPLITTING OF VARBIND LIST

AGENT SYSTEM

MANAGER SYSTEM

MASTER AGENT

A

B

C

A

B

SUB AGENT

MIB

C

SUB AGENT

MIB

SUB AGENT

MIB
CHARACTERISTICS

REQUIRES OID REGISTRATION:

- TOP REGISTRATION
  EXAMPLE: REGISTER(mib-2)

- RANGE REGISTRATION
  EXAMPLE REGISTER(interfaces -> tcp)
POTENTIAL PROBLEMS

• TABLE ENTRIES MAY BE CREATED AND DELETED AT RUN-TIME

• ENTRIES OF A SINGLE TABLE MAY BE LOCATED IN DIFFERENT SUBAGENTS

• DUPLICATED OIDs

• GAPS

• SETS

• sysUpTime
EXAMPLE: GAPS

IMAGINE WHAT HAPPENS WITH GET-NEXT OR GETBULK ...
SETS AND ATOMICITY

 TRANSACTION-LIKE APPROACH
  - TEST
  - COMMIT
  - UNDO / CLEAN
HISTORY

SMUX (1991: RFC 1227)
SNMP MULTIPLEXING PROTOCOL

DISTRIBUTED PROTOCOL INTERFACE

RESEARCH PROTOTYPES
FOR EXAMPLE: UNIVERSITY OF TWENTE - UT-SNMPv2

COMMERCIAL PRODUCTS
FOR EXAMPLE: SNMP RESEARCH - EMANATE
(ENHANCED MANAGEMENT AGENT THROUGH EXTENSIONS)

AGENTX

PROPOSED IETF STANDARD
- RFC 2741 & RFC 2742
- http://www.scguild.com/agentx/

HAS EFFICIENT MESSAGE FORMAT AND CODING

SUPPORTS
- SUBAGENTS IMPLEMENTING SEPARATE MIB MODULES
- SUBAGENTS IMPLEMENTING ROWS IN "SIMPLE TABLES"
- SUBAGENTS SHARING TABLES ALONG NON-ROW BORDERS

NON-GOALS
- SUBAGENTS SHARING "COMPLEX TABLES"
- SUBAGENT TO SUBAGENT COMMUNICATION
**EXAMPLE: PDU FORMAT OF GetNext**

<table>
<thead>
<tr>
<th>VERSION</th>
<th>TYPE</th>
<th>FLAGS</th>
<th>RESERVED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SESSION ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANSACTION ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PACKET ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAYLOAD LENGTH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONTEXT (OPTIONAL)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LENGTH</th>
<th>PREFIX</th>
<th>INCLUDE</th>
<th>RESERVED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FIRST SUB IDENTIFIER</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LAST SUB IDENTIFIER</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OBJECT 1</th>
<th>START OF RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>OBJECT 1</td>
<td>END OF RANGE</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>OBJECT N</td>
<td>START OF RANGE</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>OBJECT N</td>
<td>END OF RANGE</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
AGENTX - ADMINISTRATIVE PDUS

Open
Close

AddAgentCaps
RemoveAgentCaps

Register
Unregister

IndexAllocate
IndexDeallocate

Ping

Response
OPEN & CLOSE

TO ESTABLISH A SESSION

A UNIQUE `sessionID` IS ASSIGNED

SUBAGENT SPECIFIES DEFAULT TIME-OUT

RESPONSES FROM MASTER ALWAYS INCLUDE `sysUpTime`

SESSION CAN BE CLOSED BY MASTER OR SUBAGENT
AGENT CAPABILITIES

TO INFORM THE MASTER OF THE AGENT’S CAPABILITIES

CAPABILITIES ARE DEFINED AS:
• AN OBJECT ID
• A HUMAN READABLE STRING

THE CAPABILITIES ARE STORED IN THE sysORTable
REGISTRATION

CHOICE BETWEEN:
- TOP REGISTRATION
- RANGE REGISTRATION

PRIORITITY CAN BE SPECIFIED
- TO DETERMINE THE AUTHORITATIVE SUBAGENT

TIME-OUT CAN BE SPECIFIED
INDEXALLOCATION

TO ALLOCATE ONE OR MORE TABLE ROWS

SUBAGENT REQUESTS ALLOCATION OF:
  • A SPECIFIC INDEX VALUE
  • AN INDEX VALUE THAT IS NOT CURRENTLY ALLOCATED
  • AN INDEX VALUE THAT HAS NEVER BEEN ALLOCATED

MASTER AGENT MAINTAINS DATABASE

AFTER ALLOCATION REGISTRATION IS STILL NEEDED
TO MONITOR IF THE MASTER AGENT IS STILL ABLE TO RECEIVE AND SEND AGENTX PDUs
AGENTX TRANSPORT MAPPINGS

UNIX SOCKETS
- var/agentx/master

TCP
- PORT 705

ANY OTHER IPC MECHANISM
- FOR EXAMPLE: SHARED MEMORY