SNMP

Simple Network Management Protocol
Network Management

- The network management is to
  - Monitor the network
  - Ensure the operations over the network are functional
  - Assure the network works efficiently

- An ounce of prevention is worth a pound of cure
  - Something wrong
    - Service down, fix the problem, resume the service
  - Nothing wrong
    - Service is somewhat abnormal, try to fix it online

- Requirements
  - FCAPS
Requirements of Network Management

- **Fault Management**
  - Detect, isolate, reconfigure and repair the abnormal network environment
  - Problem tracking and control

- **Configuration Management**
  - Gather configuration information of network components

- **Accounting Management**
  - Track the use of network resources by end-user to provide
    - Inappropriate usage tracing, charging, statistics

- **Performance Management**
  - Capacity utilization, throughput, response time, bottleneck
    - Collect information and assess current situation

- **Security Management**
  - Information protection and access control
In that time

- Network environment is simple
  - ICMP is the only way to do network investigation
    - ping, traceroute, ....

- As Internet goes popular, three approaches are proposed:
  - HEMS: High-level Entity Management System
    - Considered to be the first network management tools
  - SGMP and SNMP
    - SNMP was an enhanced version of the Simple Gateway Management Protocol
    - For TCP/IP-based network management standards
    - Supposed to be short-term solution
  - CMIP over TCP/IP (CMOT)
    - Common Management Information Protocol
    - For ISO-based network management standards
    - Supposed to be long-term solution
Introduction

SNMP – Simple Network Management Protocol

- A set of standards for network management
  - Protocol
  - Database structure specification
  - Data objects

- A set of standardized tools that
  - Control costs of network management
  - Across various product types
    - End system, bridges, routers, telecommunications, …

- Two roles
  - Network management station: SNMP collector, manager
  - SNMP agent
History

- In 1989
  - SNMP was adopted as TCP/IP-based Internet standards

- In 1991
  - RMON – Remote network MONitoring
    - Supplement to SNMP to include management of LAN and WAN packet flow

- In 1995
  - SNMPv2 (2c)
    - Functional enhancements to SNMP
    - SNMP on OSI-based networks
  - RMON2
    - Network layer and application layer

- In 1998
  - SNMPv3
    - Precise definition, but the content is the same as SNMPv2
    - Security capability for SNMP
The roles in SNMPv3
Network Management System (1)

- A collection of tools for
  - Network monitoring
  - Network control

- These tools must be integrated
  - Single operator interface with powerful but user-friendly
  - Support of managed equipments.
Network Management System (2)

- Architecture of NMS
  - NMA
    - Operator interface
  - NME
    - Collect statistics
    - Response to NMA
    - Alert NMA when environment changing

**FIGURE 1.1** Elements of a network management system
SNMP Concepts
SNMP Architecture (1)

4 key elements

- Management station
  - Serve as the interface between manager and devices
    - Management applications
    - User-friendly interface
    - Translate manager’s requirements into actual monitoring or control operations
    - Database extracted from MIBs of all managed device

- Management Agent
  - Respond to request from management station
  - Change settings in MIB of managed device
  - Asynchronously report abnormal event (Trap)

- Management Information Base (MIB)
  - Each resource is represented as an object and MIB is a collection of objects

- Network Management Protocol
  - get, setnext, set, getresponse, trap, ...
SNMP Architecture (2)

- SNMP
  - UDP
  - Port 161(snmp)
  - Port 162(snmp-trap)
SNMP Architecture (3)

- SNMP proxy
  - Devices that do not support UDP/IP
    - ex: Bridge, Modem
  - Devices that do not want to add burden of SNMP agent
    - ex: PC, programmable controller
SNMP Message Information

- Management Information Base (MIB)
  - Collection of objects
  - Each object represents certain resource of managed device

- Interoperability of MIB
  - Object that represents a particular resource should be the same cross various system
    - What objects
    - (MIB-I) and MIB-II
  - Common representation format
    - SMI (Structure of Management Information)
SNMP Message Information – SMI (1)

- SMI
  - Structure of Management Information
  - Identify the data type that can be used in MIB
  - How resources are represented and named, including
    - MIB structure
    - Syntax and value of each object
    - Encoding of object value
SNMP Message Information – SMI (2)

MIB structure

- Rooted tree
  - The leaves are the actual managed objects
  - Each object has an identifier (OBJECT IDENTIFIER)
    - Number with dot as delimiter
  - The internet node
    - iso(1) -> org(3) -> dod(6) -> internet(1)
    - object identifier of internet node: 1.3.6.1
  - Under internet node
    - directory(1): OSI X.500 directory
    - mgmt(2): used for objects defined in IAB (Internet Activities Board)
    - experimental(3): used for internet experiments
    - private(4): unilaterally usage
SNMP Message Information – SMI (3)

- MIB Tree
- Define additional objects
  - Under mib-2
    - 1.3.6.1.2.1
  - Under experimental
    - 1.3.6.1.3
  - Under enterprises
    - 1.3.6.1.4.1
SNMP Message Information – Object Syntax (1)

Definition of object

- Data type
  - Application-independent type (UNIVERSAL type)
    - integer, octetstring, null, object identifier, sequence
  - Application-wide types (RFC 1155)
    - Networkaddress → IP Address
    - counter (0 ~ $2^{32}-1$), increasing only, wrap to 0
    - gauge (0 ~ $2^{32}-1$)
    - timeticks
    - opaque (encoded as OCTET STRING for transmission)

- Value ranges

- Relationship with other objects in MIB
SNMP Message Information – Object Syntax (2)

- ASN.1
  - Abstract Syntax Notation One
  - A formal language developed by CCITT and ISO
  - In SNMP, we use macro to define other types used to define managed objects
    - Macro definition (template)
    - Macro instance (particular type)
    - Macro instance value
SNMP Message Information – Object Syntax (3)

- Example: /usr/share/snmp/mibs/BEGEMOT-HOSTRES-MIB.txt

```plaintext
-- Additional stuff for the HOST-RESOURCES MIB.
BEGEMOT-HOSTRES-MIB DEFINITIONS ::= BEGIN

IMPORTS
  MODULE-IDENTITY, OBJECT-TYPE, TimeTicks
  FROM SNMPv2-SMI

begemot
  FROM BEGEMOT-MIB;

begemotHostres MODULE-IDENTITY
  ....
  ::= { begemot 202 }

begemotHostresObjects  OBJECT IDENTIFIER ::= { begemotHostres 1 }

begemotHrStorageUpdate OBJECT-TYPE
  SYNTAX     TimeTicks
  MAX-ACCESS read-write
  STATUS     current
  DESCRIPTION
    "The maximum number of ticks the storage table is cached."
  ::= { begemotHostresObjects 1 }
```
SNMP Message Information – Object Syntax (4)

- OBJECT-Type
  - macro

```plaintext
IMPORTS ObjectName, Object Syntax FROM RFC-1155-SMI

OBJECT-TYPE MACRO ::= BEGIN
  TYPE NOTATION ::= "SYNTAX" type (TYPE ObjectSyntax)
                 "ACCESS" Access
                 "STATUS" Status
                 DescrPart
                 ReferPart
                 IndexPart
                 DefValPart
  VALUE NOTATION ::= value (VALUE ObjectName)

  Access ::= "read-only"|"read-write"|"write-only"|"not-accessible"

  Status ::= "mandatory"|"optional"|"obsolete"|"deprecated"

  DescrPart ::= "DESCRIPTION" value (description DisplayString)|empty

  ReferPart ::= "REFERENCE" value (reference DisplayString)|empty

  IndexPart ::= "INDEX" "(" IndexTypes ")"

  IndexTypes ::= IndexType|IndexTypes "," IndexType

  IndexType ::= value (indexobject ObjectName) --if indexobject, use the SYNTAX
               --value of the correspondent
               --OBJECT-TYPE invocation
               --otherwise use named SMI type;
               --must conform to IndexSyntax below

               |type (indextype)

  DefValPart ::= "DEFVAL" "(" value (defvalue ObjectSyntax ")")" |empty

  DisplayString ::= OCTET STRING SIZE (0..255)

END

IndexSyntax ::= CHOICE { number INTEGER (0..MAX),
                        string OCTET STRING,
                        object OBJECT IDENTIFIER,
                        address NetworkAddress,
                        IpAddress IpAddress }
Example of object definition

- iso.org.dod.internet.mgmt.mib-2.tcp.tcpMaxConn
- 1.3.6.1.2.1.6.4

```
tcpMaxConn OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The limit on the total number of TCP connections the entity can support. In entities where the maximum number of connections is dynamic, this object should contain the value -1."
::= { tcp 4 }
```
SNMP Message Information – Object Syntax (6)

- 2-D table
  - Two-dimensional array with scalar-valued entries
  - Ex: tcpConnTable (RFC1213)

```
tcpConn Table OBJECT-TYPE
  SYNTAX  SEQUENCE OF TcpConnEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
    "A table containing TCP connection-specific information."
  ::= { tcp 13 }

tcpConnEntry OBJECT-TYPE
  SYNTAX  TcpConnEntry
  ACCESS  not-accessible
  STATUS  mandatory
  DESCRIPTION
    "Information about a particular TCP connection. An object of this type is
    transient, in that it ceases to exist when (or soon after) the connection
    makes the transition to the CLOSED state."
  INDEX  { tcpConnLocalAddress, tcpConnLocalPort, tcpConnRemAddress, tcpConnRemPort }
  ::= { tcpConnTable 1 }

TcpConnEntry ::= SEQUENCE { tcpConnState INTEGER,
  tcpConnLocalAddress IpAddress,
  tcpConnLocalPort INTEGER (0..65535),
  tcpConnRemAddress IpAddress
  tcpConnRemPort INTEGER (0..65535)}
```
### SNMP Message Information - Object Syntax (7)

<table>
<thead>
<tr>
<th>tcpConnState OBJECT-TYPE</th>
<th>tcpConnLocalAddress OBJECT-TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcpConnLocalPort OBJECT-TYPE</td>
<td></td>
</tr>
<tr>
<td>tcpConnRemAddress OBJECT-TYPE</td>
<td></td>
</tr>
<tr>
<td>tcpConnRemPort OBJECT-TYPE</td>
<td></td>
</tr>
</tbody>
</table>

**tcpConnState**

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>INTEGER</th>
</tr>
</thead>
<tbody>
<tr>
<td>closed</td>
<td>(1)</td>
</tr>
<tr>
<td>listen</td>
<td>(2)</td>
</tr>
<tr>
<td>synSent</td>
<td>(3)</td>
</tr>
<tr>
<td>synReceived</td>
<td>(4)</td>
</tr>
<tr>
<td>established</td>
<td>(5)</td>
</tr>
<tr>
<td>finWait1</td>
<td>(6)</td>
</tr>
<tr>
<td>finWait2</td>
<td>(7)</td>
</tr>
<tr>
<td>closeWait</td>
<td>(8)</td>
</tr>
<tr>
<td>lastAck</td>
<td>(9)</td>
</tr>
<tr>
<td>closing</td>
<td>(10)</td>
</tr>
<tr>
<td>timeout</td>
<td>(11)</td>
</tr>
<tr>
<td>delete TCB</td>
<td>(12)</td>
</tr>
</tbody>
</table>

**tcpConnLocalAddress**

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>IpAddress</th>
</tr>
</thead>
</table>

**tcpConnLocalPort**

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>INTEGER</th>
</tr>
</thead>
</table>

**tcpConnRemAddress**

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>IpAddress</th>
</tr>
</thead>
</table>

**tcpConnRemPort**

<table>
<thead>
<tr>
<th>SYNTAX</th>
<th>INTEGER</th>
</tr>
</thead>
</table>

**ACCESS**

<table>
<thead>
<tr>
<th>tcpConnState</th>
<th>read-write</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcpConnLocalAddress</td>
<td>read-only</td>
</tr>
<tr>
<td>tcpConnLocalPort</td>
<td>read-only</td>
</tr>
<tr>
<td>tcpConnRemAddress</td>
<td>read-only</td>
</tr>
<tr>
<td>tcpConnRemPort</td>
<td>read-only</td>
</tr>
</tbody>
</table>

**STATUS**

<table>
<thead>
<tr>
<th>tcpConnState</th>
<th>mandatory</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcpConnLocalAddress</td>
<td>mandatory</td>
</tr>
<tr>
<td>tcpConnLocalPort</td>
<td>mandatory</td>
</tr>
<tr>
<td>tcpConnRemAddress</td>
<td>mandatory</td>
</tr>
<tr>
<td>tcpConnRemPort</td>
<td>mandatory</td>
</tr>
</tbody>
</table>

**DESCRIPTION**

- "The state of this TCP connection.
- "The local IP address for this TCP connection. In the case of a connection in the listen state which is willing to accept connections for any IP interface associated with the node, the value 0.0.0.0 is used."
- "The local port number for this TCP connection."
- "The remote IP address for this TCP connection."
- "The remote port number for this TCP connection."
SNMP Message Information – Object Syntax (8)

- iso (1) -> org (3) -> dod (6) -> internet (1) -> mgmt (2)
  - mib-2 (1) -> tcp (6) -> tcpConnTable(13)

<table>
<thead>
<tr>
<th>tcpConnTable (1.3.6.1.2.1.6.13)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcpConnState</td>
</tr>
<tr>
<td>1.3.6.1.2.1.6.13.1.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INDEX</th>
<th>INDEX</th>
<th>INDEX</th>
<th>INDEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.0.0.0.99</td>
<td>12</td>
<td>9.1.2.3</td>
</tr>
<tr>
<td>2</td>
<td>0.0.0.0</td>
<td>99</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.0.0.0.99</td>
<td>14</td>
<td>89.1.1.42</td>
</tr>
</tbody>
</table>
Standard MIBs
MIB-II (1)

- RFC1213
  - MIB-I (RFC 1156)
  - MIB-II is a superset of MIB-I with some additional objects and groups
MIB-II (2)

- First layer under mib-2
  - 1.3.6.1.2.1 (iso.org.dod.internet.mgmt.mib-2)
    - system
      - Overall information about the system
    - interfaces
      - Information about each interface
    - at
      - Address translation (obsolete)
    - ip, icmp, tcp, udp, egp
    - transmission
      - Transmission schemes and access protocol at each system interface
    - snmp
# MIB-II

## system group

- sysServices
  - 1 physical (ex: repeater)
  - 2 datalink/subnetwork (ex: bridge)
  - 3 internet (ex: router)
  - 4 end-to-end (ex: IP hosts)
  - 7 applications (ex: mail relays)

<table>
<thead>
<tr>
<th>Object</th>
<th>Syntax</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysDescr</td>
<td>DisplayString (SIZE [0...255])</td>
<td>RO</td>
<td>A description of the entity, such as hardware, operating system, etc.</td>
</tr>
<tr>
<td>sysObjectID</td>
<td>OBJECT IDENTIFIER</td>
<td>RO</td>
<td>The vendor’s authoritative identification of the network management subsystem contained in the entity</td>
</tr>
<tr>
<td>sysUpTime</td>
<td>TimeTicks</td>
<td>RO</td>
<td>The time since the network management portion of the system was last reinitialized</td>
</tr>
<tr>
<td>sysContact</td>
<td>DisplayString (SIZE [0...255])</td>
<td>RW</td>
<td>The identification and contact information of the contact person for this managed node</td>
</tr>
<tr>
<td>sysName</td>
<td>DisplayString (SIZE [0...255])</td>
<td>RW</td>
<td>An administratively assigned name for this managed node</td>
</tr>
<tr>
<td>sysLocation</td>
<td>DisplayString (SIZE [0...255])</td>
<td>RW</td>
<td>The physical location of this node</td>
</tr>
<tr>
<td>sysServices</td>
<td>INTEGER (0...127)</td>
<td>RO</td>
<td>A value that indicates the set of services this entity primarily offers</td>
</tr>
</tbody>
</table>
MIB-II

interface group (1)
## MIB-II

### interface group (2)

<table>
<thead>
<tr>
<th>Object</th>
<th>Syntax</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ifNumber</td>
<td>INTEGER</td>
<td>RO</td>
<td>The number of network interfaces</td>
</tr>
<tr>
<td>ifTable</td>
<td>SEQUENCE OF ifEntry</td>
<td>NA</td>
<td>A list of interface entries</td>
</tr>
<tr>
<td>ifEntry</td>
<td>SEQUENCE</td>
<td>NA</td>
<td>An interface entry containing objects at the subnetwork layer and below for a particular interface</td>
</tr>
<tr>
<td>ifIndex</td>
<td>INTEGER</td>
<td>RO</td>
<td>A unique value for each interface</td>
</tr>
<tr>
<td>ifDescr</td>
<td>DisplayString (SIZE (0...255))</td>
<td>RO</td>
<td>Information about the interface, including name of manufacturer, product name, and version of the hardware interface</td>
</tr>
<tr>
<td>ifType</td>
<td>INTEGER</td>
<td>RO</td>
<td>Type of interface, distinguished according to the physical/link protocol(s)</td>
</tr>
<tr>
<td>ifMtu</td>
<td>INTEGER</td>
<td>RO</td>
<td>The size of the largest protocol data unit, in octets, that can be sent/received on the interface</td>
</tr>
<tr>
<td>ifSpeed</td>
<td>Gauge</td>
<td>RO</td>
<td>An estimate of the interface's current data rate capacity</td>
</tr>
<tr>
<td>ifPhysAddress</td>
<td>PhysAddress</td>
<td>RO</td>
<td>The interface's address at the protocol layer immediately below the network layer</td>
</tr>
<tr>
<td>ifAdminStatus</td>
<td>INTEGER</td>
<td>RW</td>
<td>Desired interface state (up(1), down(2), testing(3))</td>
</tr>
<tr>
<td>ifOperStatus</td>
<td>INTEGER</td>
<td>RO</td>
<td>Current operational interface state (up(1), down(2), testing(3))</td>
</tr>
<tr>
<td>ifLastChange</td>
<td>TimeTicks</td>
<td>RO</td>
<td>Value of sysUpTime at the time the interface entered its current operational state</td>
</tr>
<tr>
<td>ifInOctets</td>
<td>Counter</td>
<td>RO</td>
<td>Total number of octets received on the interface, including framing characters</td>
</tr>
<tr>
<td>ifInUcastPkts</td>
<td>Counter</td>
<td>RO</td>
<td>Total number of subnetwork-unicast packets delivered to a higher-layer protocol</td>
</tr>
<tr>
<td>ifInNUcastPkts</td>
<td>Counter</td>
<td>RO</td>
<td>Number of nonunicast packets delivered to a higher-layer protocol</td>
</tr>
<tr>
<td>ifInDiscards</td>
<td>Counter</td>
<td>RO</td>
<td>Number of inbound packets discarded, even though no errors had been detected, to prevent their being deliverable to a higher-layer protocol (e.g., buffer overflow)</td>
</tr>
<tr>
<td>ifInErrors</td>
<td>Counter</td>
<td>RO</td>
<td>Number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol</td>
</tr>
<tr>
<td>ifInUnknownProtos</td>
<td>Counter</td>
<td>RO</td>
<td>Number of inbound packets that were discarded because of an unknown or unsupported protocol</td>
</tr>
<tr>
<td>ifOutOctets</td>
<td>Counter</td>
<td>RO</td>
<td>Total number of octets transmitted on the interface, including framing characters</td>
</tr>
<tr>
<td>ifOutUcastPkts</td>
<td>Counter</td>
<td>RO</td>
<td>Total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or otherwise not sent</td>
</tr>
<tr>
<td>ifOutNUcastPkts</td>
<td>Counter</td>
<td>RO</td>
<td>Total number of packets that higher-level protocols requested be transmitted to a nonunicast address, including those that were discarded or otherwise not sent</td>
</tr>
<tr>
<td>ifOutDiscards</td>
<td>Counter</td>
<td>RO</td>
<td>Number of outbound packets discarded even though no errors had been detected to prevent their being transmitted (e.g., buffer overflow)</td>
</tr>
<tr>
<td>ifOutErrors</td>
<td>Counter</td>
<td>RO</td>
<td>Number of outbound packets that could not be transmitted because of errors</td>
</tr>
<tr>
<td>ifOutQLen</td>
<td>Gauge</td>
<td>RO</td>
<td>Length of the output packet queue</td>
</tr>
<tr>
<td>ifSpecific</td>
<td>OBJECT IDENTIFIER</td>
<td>RO</td>
<td>Reference to MIB definitions specific to the particular media being used to realize the interface</td>
</tr>
</tbody>
</table>
MIB-II

tcp group

tcp (mib-2 6)
  tcpRtoAlgorithm (1)
  tcpRtoMin (2)
  tcpRtoMax (3)
  tcpMaxConn (4)
  tcpActiveOpens (5)
  tcpPassiveOpens (6)
  tcpAttemptFails (7)
  tcpEstabResets (8)
  tcpCurrEstab (9)
  tcpInSegs (10)
  tcpOutSegs (11)
  tcpRetransSegs (12)
  tcpConnTable (13)
    tcpConnEntry (1)
      tcpConnState (1)
      tcpConnLocalAddress (2)
      tcpConnLocalPort (3)
      tcpConnRemAddress (4)
      tcpConnRemPort (5)
    tcpInErrors (14)
    tcpOutRsts (15)
MIB-II

ip group

- ipForwarding (1)
- ipDefaultTTL (2)
- ipInReceives (3)
- ipInHdrErrors (4)
- ipInAddrErrors (5)
- ipForwDatagrams (6)
- ipInUnknownProtos (7)
- ipInDiscards (8)
- ipInDelivers (9)
- ipOutRequests (10)
- ipOutDiscards (11)
- ipOutNoRoutes (12)
- ipReasmTimeout (13)
- ipReasmReqds (14)
- ipReasmOKs (15)
- ipReasmFails (16)
- ipFragOKs (17)
- ipFragFails (18)
- ipFragCreates (19)
- ipAddrTable (20)
  - ipAddrEntry (1)
- ipRouteTable (21)
  - ipRouteEntry (1)
- ipNetToMediaTable (22)
  - ipNetToMediaEntry (1)
- ipRoutingDiscards (23)
Host Resource MIB

- RFC2790
  - host OBJECT IDENTIFIER ::= { mib-2 25 }
  - hrSystem OBJECT IDENTIFIER ::= { host 1 }
  - hrStorage OBJECT IDENTIFIER ::= { host 2 }
  - hrDevice OBJECT IDENTIFIER ::= { host 3 }
  - hrSWRun OBJECT IDENTIFIER ::= { host 4 }
  - hrSWRunPerf OBJECT IDENTIFIER ::= { host 5 }
  - hrSWInstalled OBJECT IDENTIFIER ::= { host 6 }
  - hrMIBAdminInfo OBJECT IDENTIFIER ::= { host 7 }
SNMP Protocol
SNMP Protocol

- Supported operations
  - get, getnext, set, getresponse, trap, …

- Simplicity vs. limitations
  - Not possible to change the structure of MIB by adding or deleting object instances
  - Access is provided only to leaf objects
SNMP Protocol – security concern

- In management environment
  - The management station and managed agent
    - One-to-many relationship
    - One station may manage all or a subset of target
  - The managed station and management station
    - One-to-many relationship
    - Each managed agent controls its local MIB and must be able to control the use of that MIB
    - Three aspects
      - Authentication service
      - Access policy
      - Proxy service
SNMP Protocol – communities (1)

- An SNMP community
  - A relationship between an SNMP agent and a set of SNMP managers that defines
    - Authentication, access control and proxy
  - The managed system establishes one community for each combination of authentication, access control and proxy
  - Each community has a unique “community name”
  - Management station use certain community name in all get and set operations
SNMP Protocol – communities (2)

- **Authentication**
  - The community name (password)

- **Access policy**
  - Community profile
    - SNMP MIB view
      - A subset of MIB objects
    - SNMP access mode
      - read-only, read-write, write-only, non-accessible
SNMP Protocol – Where is the security

- SNMPv3
  - User-based Security Model (USM)
    - Message Authentication
      - HMAC
        - MD5, SHA-1
        - Authentication passphrase, secret key
    - Encryption
      - CBC-DES
  - View-based Access Control Model (VACM)
    - Context table
    - Security to group table
    - Access table
    - View tree family table
BSNMP

mini SNMP daemon
BSNMP (1)

- A portable SNMP framework consisting of a daemon (bsnmpd), modules and tools

- Enable bsnmpd
  - /etc/rc.conf
    ```
    bsnmpd_enable="YES"
    ```

- Configure bsnmpd
  - /etc/snmpd.config
    ```
    location := "Room 200"
    contact := "sysmeister@example.com"
    system := 1  # FreeBSD
    traphost := localhost
    trapport := 162
    
    # Change this!
    read := "public"
    
    # Host resources module
    # This requires the mibII module.
    
    #begemotSnmpdModulePath."hostres" = "/usr/lib/snmp_hostres.so"
    ```

- Start bsnmpd
  - /etc/rc.d/bsnmpd start
  - Firewall allows
    - snmpd: udp 161
    - snmptrapd: udp 162
BSNMP (2)

- bsnmp-ucd
  - bsnmpd module that implements parts of UCD-SNMP-MIB
  - The counters will be available under the following MIB: .1.3.6.1.4.1.2021 (UCD-SNMP-MIB::ucdavis)

- Install bsnmp-ucd
  - net-mgmt/bsnmp-ucd

- Configure bsnmpd
  - /etc/snmpd.config

```
begemotSnmpdModulePath."ucd" = "/usr/local/lib/snmp_ucd.so"
```

- Restart bsnmpd

- Other bsnmp modules
  - bsnmp-jails
    - bsnmpd module to measure statistics about jails
  - bsnmp-regexp
    - bsnmpd module allowing creation of counters from log files
Net-SNMP

previously known as "ucd-snmp"
Net-SNMP (1)

- Install net-snmp
  - net-mgmt/net-snmp
  - # make [OPTIONS] install clean

```
DEFAULT_SNMP_VERSION="3"    # Default version of SNMP to use.
NET_SNMP_SYS_CONTACT="nobody@nowhere.invalid"    # Default system contact.
NET_SNMP_SYS_LOCATION="somewhere"    # Default system location.
NET_SNMP_LOGFILE="/var/log/snmpd.log"    # Default log file location for snmpd.
NET_SNMP_PERSISTENTDIR="/var/net-snmp"    # Default directory for persistent data storage.
```

- Firewall allows
  - snmpd: udp 161
  - snmpttrapd: udp 162
Net-SNMP (2)

- After installing...

If you want to auto-start snmpd and snmptrapd:, add the following to /etc/rc.conf:

```plaintext
snmpd_enable="YES"
snmpd_flags="-a"
snmpd_conffile="/usr/local/share/snmp/snmpd.conf /etc/snmpd.conf"

snmptrapd_enable="YES"
snmptrapd_flags="-a -p /var/run/snmptrapd.pid"
```

- /usr/local/share/snmp/snmpd.conf.example

```plaintext
# Full access from the local host
# rocommunity public localhost
rocommunity public default -V systemonly
```
Use `snmpconf` command to generate the configuration files

- `snmpconf -g basic_setup`
- `snmpconf`
  - System Information Setup
    - Location, contact, service
  - Access Control Setup
    - SNMPv3 or SNMPv1 access community
  - Trap Destination
    - Where to send the trap
  - Monitor Various Aspects of the Running Host
    - Process, disk space, load, file
  - Extending the Agent
    - Let snmp agent to return information that yourself define
  - Agent Operating Mode
    - User/group, IP port,...
Net-SNMP (4)

- To get various value
  - man snmpget, snmpgetnext, snmptable
    - `% snmpget -c public -v 1 nasa system.sysContact.0`
    - `% snmpgetnext -c public -v 1 nasa system.sysContact.0`
    - `% snmptable -c public -v 1 nasa mib-2.tcp.tcpConnTable`
    - `% snmpwalk -c public -v 1 nasa system`
    - `% snmpwalk -c public -v 1 nasa iso.org.dod.internet.private.enterprises`
Appendix

Cacti
Cacti(1)

- About
  - Cacti is a complete network graphing solution designed to harness the power of RRDTool's data storage and graphing functionality.
  - Cacti provides a fast poller, advanced graph templating, multiple data acquisition methods, and user management features out of the box.
  - All of this is wrapped in an intuitive, easy to use interface that makes sense for LAN-sized installations up to complex networks with hundreds of devices.

- Install cacti
  - /usr/ports/net/cacti
## Cacti Installation Guide

Thanks for taking the time to download and install cacti, the complete graphing solution for your network. Before you can start making cool graphs, there are a few pieces of data that cacti needs to know.

Make sure you have read and followed the required steps needed to install cacti before continuing. Install information can be found for [Unix](#) and [Win32](#)-based operating systems.

Also, if this is an upgrade, be sure to reading the [Upgrade](#) information file.

Cacti is licensed under the GNU General Public License, you must agree to its provisions before continuing:

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

[Next >>](#)
Cacti(3)

Cacti Installation Guide

Please select the type of installation

New Install

The following information has been determined from Cacti's configuration file. If it is not correct, please edit 'include/config.php' before continuing.

Database User: cactiuser
Database Hostname: localhost
Database: cacti
Server Operating System Type: unix

Next >>
Cacti(4)

Cacti Installation Guide

Make sure all of these values are correct before continuing.

[NOT FOUND] RRDTool Binary Path: The path to the rrdtool binary.
/opt/bin/rrdtool
[ERROR: FILE NOT FOUND]

[FOUND] PHP Binary Path: The path to your PHP binary file (may require a php recompile to get this file).
/usr/bin/php
[OK: FILE FOUND]

[NOT FOUND] snmpwalk Binary Path: The path to your snmpwalk binary.
/opt/bin/snmpwalk
[ERROR: FILE NOT FOUND]

[NOT FOUND] snmpget Binary Path: The path to your snmpget binary.
/opt/bin/snmpget
[ERROR: FILE NOT FOUND]

[NOT FOUND] snmpbulkwalk Binary Path: The path to your snmpbulkwalk binary.
/opt/bin/snmpbulkwalk
[ERROR: FILE NOT FOUND]

[NOT FOUND] snmpgetnext Binary Path: The path to your snmpgetnext binary.
/opt/bin/snmpgetnext
[ERROR: FILE NOT FOUND]

/volume1/web/cacti/log/cacti.log
[OK: FILE FOUND]

SNMP Utility Version: The type of SNMP you have installed. Required if you are using SNMP v2c or don't have embedded SNMP support in PHP.
- NET-SNMP 5.x

NOTE: Once you click "Finish", all of your settings will be saved and your database will be upgraded if this is an upgrade. You can change any of the settings on this screen at a later time by going to "Cacti Settings" from within Cacti.

Finish
Cacti(5)

- Default account/pwd
  - admin/admin
Cacti(6)

You are now logged into Cacti. You can follow these basic steps to get started:

- Create devices for network
- Create graphs for your new devices
- View your new graphs
Cacti(7)