Chapter 15. Network management

With the growth in size and complexity of the TCP/IP-based internetworks the need for network management became very important.

The Internet Architecture Board (IAB) issued an RFC 1052 detailing its recommendation, which adopted two different approaches:

- Simple Network Management Protocol (SNMP).

The IAB said that, in the short term, SNMP should be used. SNMP became so popular that it has become the industry-wide standard for reporting management data for an IP based network.

15.1 Simple Network Management Protocol and MIB overview

The SNMP network management framework consists of:

- A large number of managed nodes, each with an SNMP entity (agent). An SNMP agent is a server at a managed host that responds to SNMP requests from managers. An agent must be present at each IP host in order to enable management of that host by a SNMP manager. The SNMP agent manages/supports the Management Information Base (MIB) on the IP host on which it resides.
- At least one SNMP entity with management applications (manager). An SNMP manager is an application that runs on a management station that typically requests management data from an SNMP agent using the SNMP protocol.
- The management information database (MIB) for each entity. MIBs are the representation of the data objects that are managed by the agents. MIBs are represented using ASN.1 notation.
- A protocol to carry management information between the entities. The protocol that is used between an agent and the manager is SNMP. The protocol used between an agent and a sub-agent can be DPI/SMUX/AgentX or any proprietary protocol.

The IAB recommends that all IP and TCP implementations be network manageable using SNMP, that is, all hosts, gateways, and other IP aware devices should implement at least MIB-II.
Note that the historic protocols Simple Gateway Monitoring Protocol (SGMP, RFC 1028) and MIB-I (RFC 1156) are not recommended for use.

SNMP is an Internet standard protocol. Its status is recommended. Its current specification can be found in RFC 1157 Simple Network Management Protocol (SNMP).

MIB-II is an Internet standard protocol. Its status is recommended. Its current specification can be found in RFC 1213 Management Information Base for Network Management of TCP/IP-based internetworks: MIB-II.

More information can be found in:
1. SMI (RFC 1155) - Describes how managed objects contained in the MIB are defined. (It is discussed in 15.2, “Structure and identification of management information (SMI)” on page 526.)
2. MIB-II (RFC 1213) - Describes the managed objects contained in the MIB. (It is discussed in 15.3, “Management Information Base (MIB)” on page 528.)
3. SNMP (RFC 1157) - Defines the protocol used to manage these objects. (It is discussed in 15.4, “Simple Network Management Protocol (SNMP)” on page 532.)

15.2 Structure and identification of management information (SMI)

The SMI defines the rules for how managed objects are described and how management protocols can access these objects. The description of managed objects is made using a subset of the ASN.1 (Abstract Syntax Notation 1, ISO standard 8824), a data description language. The object type definition consists of five fields:

- Object: A textual name, termed the object descriptor, for the object type along with its corresponding object identifier defined below.
- Syntax: The abstract syntax for the object type. It can be a choice of SimpleSyntax (Integer, Octet String, Object Identifier, Null) or an ApplicationSyntax (NetworkAddress, Counter, Gauge, TimeTicks, Opaque) or other application-wide types (see RFC 1155 for more details).
- Definition: A textual description of the semantics of the object type.
- Access: One of read-only, read-write, write-only or not-accessible.
- Status: One of mandatory, optional, or obsolete.

As an example, we can have:
This example shows the definition of an object contained in the Management Information Base (MIB). Its name is sysDescr and it belongs to the system group (see 15.3, “Management Information Base (MIB)” on page 528).

A managed object not only has to be described but identified, too. This is done using the ASN.1 Object Identifier in the same way as a telephone number, reserving group of numbers to different locations. In the case of TCP/IP-based network management the number allocated was 1.3.6.1.2 and SMI uses it as the base for defining new objects.

The number 1.3.6.1.2 is obtained by joining groups of numbers with the following meaning:

- The first group defines the node administrator:
  - (1) for ISO
  - (2) for CCITT
  - (3) for the joint ISO-CCITT
- The second group for the ISO node administrator defines (3) for use by other organizations.
- The third group defines (6) for the use of the U.S. Department of Defense (DoD).
- In the fourth group the DoD has not indicated how it will manage its group so the Internet community assumed (1) for its own.
- The fifth group was approved by IAB to be:
  - (1) for the use of OSI directory in the Internet
  - (2) for objects identification for management purposes
  - (3) for objects identification for experimental purposes
  - (4) for objects identification for private use.

In the example the “{ system 1 }” beside the object name means that the object identifier is 1.3.6.1.2.1.1.1. It is the first object in the first group (system) in the Management Information Base (MIB).
15.3 Management Information Base (MIB)

The MIB defines the objects that may be managed for each layer in the TCP/IP protocol. There are two versions: MIB-I and MIB-II. MIB-I was defined in RFC 1156, and is now classified as an historic protocol with a status of not recommended.

MIB-II is described in RFC 1213. The group definitions are:

Table 23. Management Information Base II (MIB-II) group definitions

<table>
<thead>
<tr>
<th>Group</th>
<th>Objects for</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Basic system information</td>
<td>7</td>
</tr>
<tr>
<td>Interfaces</td>
<td>Network attachments</td>
<td>23</td>
</tr>
<tr>
<td>AT</td>
<td>Address translation</td>
<td>3</td>
</tr>
<tr>
<td>IP</td>
<td>Internet protocol</td>
<td>42</td>
</tr>
<tr>
<td>ICMP</td>
<td>Internet control message protocol</td>
<td>26</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission control protocol</td>
<td>19</td>
</tr>
<tr>
<td>UDP</td>
<td>User datagram protocol</td>
<td>7</td>
</tr>
<tr>
<td>EGP</td>
<td>Exterior gateway protocol</td>
<td>18</td>
</tr>
<tr>
<td>SNMP</td>
<td>SNMP applications entities</td>
<td>39</td>
</tr>
</tbody>
</table>

Legend: # = Number of objects in the group

There is also space in the definition for a Transmission group, for describing the underlying media.

Each managed node supports only those groups that are appropriate. For example, if there is no gateway, the EGP group need not be supported. If a group is appropriate, all objects in that group must be supported.

The list of managed objects defined has been derived from those elements considered essential. This approach of taking only the essential objects is not restrictive, since the SMI provides extensibility mechanisms such as definition of a new version of the MIB and definition of private or non-standard objects.

Below are some examples of objects in each group. The complete list is defined in RFC 1213. Please also refer to RFC 2011, RFC 2012 and RFC 2013 for updated information of IP, TCP and UDP.
• System Group
  - sysDescr - Full description of the system (version, HW, OS)
  - sysObjectID - Vendor's object identification
  - sysUpTime - Time since last re-initialization
  - sysContact - Name of contact person
  - sysServices - Services offered by device

• Interfaces Group
  - ifIndex - Interface number
  - ifDescr - Interface description
  - ifType - Interface type
  - ifMtu - Size of the largest IP datagram
  - ifAdminStatus - Status of the interface
  - ifLastChange - Time the interface entered in the current status
  - ifInErrors - Number of inbound packets that contained errors
  - ifOutDiscards - Number of outbound packets discarded

• Address Translation Group
  - atTable - deprecated (MIB-I)
  - atTableEntry - deprecated (MIB-I)
  - atIfIndex - IfIndex interface number
  - atPhysAddress - The media-dependent physical address
  - atNetAddress - The network address corresponding to the media-dependent physical address

• IP Group
  - ipForwarding - Indication of whether this entity is an IP gateway
  - ipInHdrErrors - Number of input datagrams discarded due to errors in their IP headers
  - ipInAddrErrors - Number of input datagrams discarded due to errors in their IP address
  - ipInUnknownProtos - Number of input datagrams discarded due to unknown or unsupported protocol
  - ipReasmOKs - Number of IP datagrams successfully re-assembled
  - ipRouteDest - Destination IP address

• ICMP Group
  - icmpInMsgs - Number of ICMP messages received
  - icmpInDestUnreaches - Number of ICMP destination-unreachable messages received
  - icmpInTimeExcds - Number of ICMP time-exceeded messages received
- icmpInSrcQuenchs - Number of ICMP source-quench messages received
- icmpOutErrors - Number of ICMP messages not sent due to problems within ICMP

• TCP Group
- tcpRtoAlgorithm - Algorithm to determine the timeout for retransmitting unacknowledged octets
- tcpMaxConn - Limit on the number of TCP connections the entity can support
- tcpActiveOpens - Number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state
- tcpInSegs - Number of segments received, including those received in error
- tcpConnRemAddress - The remote IP address for this TCP connection
- tcpInErrs - Number of segments discarded due to format error
- tcpOutRsts - Number of resets generated

• UDP Group
- udpInDatagrams - Number of UDP datagrams delivered to UDP users
- udpNoPorts - Number of received UDP datagrams for which there was no application at the destination port
- udpInErrors - Number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port
- udpOutDatagrams - Number of UDP datagrams sent from this entity

• EGP Group
- egpInMsgs - Number of EGP messages received without error
- egpInErrors - Number of EGP messages with errors
- egpOutMsgs - Number of locally generated EGP messages
- egpNeighAddr - The IP address of this entry's EGP neighbor
- egpNeighState - The EGP state of the local system with respect to this entry's EGP neighbor

This is not the complete MIB definition but it is presented as an example of the objects defined in each group. These modules currently support IPv4.

To illustrate this, the Interfaces Group contains two top-level objects: the number of interface attachments on the node (ifNumber) and a table containing information on those interfaces (ifTable). Each entry (ifEntry) in that table contains the objects for a particular interface. Among those, the interface type (ifType) is identified in the MIB tree using the ASN.1 notation by 1.3.6.1.2.1.2.1.3. and for a token-ring adapter the value of the
corresponding variable would be 9, which means iso88025-tokenRing (see Figure 211).

![Diagram showing Object Identifier allocation for TCP/IP-based network](image)

**Figure 211. MIB-II - Object identifier allocation for TCP/IP-based network**

### 15.3.1 IBM-specific MIB part

IBM has added the following objects in the MIB-II database:

* IBM SNMP agent DPI UDP port
  
  DPI_port  
  1.3.6.1.4.1.2.1.1.  number  2

* IBM "ping" round-trip-time table
  
  RTTaddr  
  1.3.6.1.4.1.2.1.3.1. internet  60

  minRTT  
  1.3.6.1.4.1.2.1.3.2. number  60

  maxRTT  
  1.3.6.1.4.1.2.1.3.3. number  60
Where:

- DPI_port returns the port number between the agent and the subagent.
- "RTT" allows an SNMP manager to ping remote hosts. RTT stands for Round Trip Time table.
  - RTTaddr: host address
  - minRTT: minimum round trip time
  - maxRTT: maximum round trip time
  - aveRTT: average round trip time
  - RTTtries: number of pings yet to be performed
  - RTTresponses: number of responses received

15.4 Simple Network Management Protocol (SNMP)

The SNMP added the improvement of many years of experience in SGMP and allowed it to work with the objects defined in the MIB with the representation defined in the SIM.

RFC 1157 defines the Network Management Station (NMS) as the one that executes network management applications (NMA) that monitor and control network elements (NE) such as hosts, gateways and terminal servers. These network elements use a management agent (MA) to perform the network management functions requested by the network management stations. The Simple Network Management Protocol (SNMP) is used to communicate management information between the network management stations and the agents in the network elements (see Figure 212 for more details).
All the management agent functions are only alterations (set) or inspections (get) of variables limiting the number of essential management functions to two and avoiding more complex protocols. In the other direction, from NE to NMS, a limited number of unsolicited messages (traps) are used to indicate asynchronous events. In the same way, trying to preserve the simplicity, the interchange of information requires only an unreliable datagram service and every message is entirely and independently represented by a single transport datagram. This means also that the mechanisms of the SNMP are generally suitable for use with a wide variety of transport services. RFC 1157 specifies the exchange of messages via the UDP protocol, but a wide variety of transport protocols can be used.

The entities residing at management stations and network elements that communicate with one another using the SNMP are termed SNMP application entities. The peer processes that implement it are the protocol entities. An SNMP agent with some arbitrary set of SNMP application entities is called an
SNMP community, where each one is named by a string of octets that need to be unique only to the agent participating in the community.

A message in the SNMP protocol consists of a version identifier, an SNMP community name and a protocol data unit (PDU). It is mandatory that all implementations of SNMP support the five PDUs:

- **GetRequest**: Retrieve the values of a specific object from the MIB.
- **GetNextRequest**: Walk through portions of the MIB.
- **SetRequest**: Alter the values of a specific object from the MIB.
- **GetResponse**: Response from a GetRequest, a GetNextRequest and a SetRequest.
- **Trap**: Capability of the network elements to generate events to network management stations such as agent initialization, agent restart and link failure. There are seven trap types defined in RFC 1157: coldStart, warmStart, linkDown, linkUp, authenticationFailure, egpNeighborLoss and enterpriseSpecific.

The formats of these messages are illustrated in Figure 213.

![SNMP message format](image)

*Figure 213. SNMP message format - Request, set and trap PDU format*
There are three versions of SNMP available, usually referred as SNMPv1, SNMPv2, and SNMPv3, respectively. The security functions provided by the SNMP protocols are categorized into the following two models:

- Community-based security model, whose data is protected by nothing more than a password, namely the community name. This level of security is provided by the SNMPv1 and SNMPv2c Community-Based Security Model.

- User-based security model (USM), which provides different levels of security, based on the user accessing the managed information. To support this security level, the SNMPv3 framework defines several security functions, such as USM for authentication and privacy and view-based access control model (VACM), which provides the ability to limit access to different MIB objects on a per-user basis, and the use of authentication and data encryption for privacy.

### 15.5 Simple Network Management Protocol Version 2 (SNMPv2)

The framework of Version 2 of the Simple Network Management Protocol (SNMPv2) was published in April 1993 and consists of 12 RFCs including the first, RFC 1441, which is an introduction. In August 1993, all 12 RFCs became a proposed standard with the status elective.

This framework consists of the following disciplines:

- **Structure of Management Information (SMI)**
  - Definition of the OSI ASN.1 subset for creating MIB modules. See RFC 2578 for a description.

- **Textual conventions**
  - Definition of the initial set of textual conventions available to all MIB modules. See RFC 2579 for a description.

- **Protocol operations**
  - Definition of protocol operations with respect to the sending and receiving of PDUs. See RFC 1905 for a description.

- **Transport mappings**
  - Definition of mapping SNMPv2 onto an initial set of transport domains because it can be used over a variety of protocol suites. The mapping onto UDP is the preferred mapping. The RFC also defines OSI, DDP, IPX etc. See RFC 1906 for a description.

- **Protocol instrumentation**
Definition of the MIB for SNMPv2. See RFC 1907 for a description.

- Administrative framework

Definition of the administrative infrastructure for SNMPv2, the user-based security model for SNMPv2 and the community-based SNMPv2. See RFCs 1909, 1910 and 1901 for descriptions.

- Conformance statements

Definition of the notation compliance or capability of agents. See RFC 2580 for a description.

The following sections describe the major differences and improvements from SNMPv1 to SNMPv2.

15.5.1 SNMPv2 entity

An SNMPv2 entity is an actual process that performs network management operations by generating and/or responding to SNMPv2 protocol messages by using the SNMPv2 protocol operations. All possible operations of an entity can be restricted to a subset of all possible operations that belong to a particular administratively defined party (please refer to 15.5.2, “SNMPv2 party” on page 536). An SNMPv2 entity could be member of multiple SNMPv2 parties. The following local databases are maintained by an SNMPv2 entity:

- One database for all parties known by the SNMPv2 entity which could be:
  - Operation realized locally
  - Operation realized by proxy interactions with remote parties or devices
  - Operation realized by other SNMPv2 entities

- Another database that represents all managed object resources that are known to that SNMPv2 entity.

- And at least a database that represents an access control policy that defines the access privileges accorded to known SNMPv2 parties.

An SNMPv2 entity can act as an SNMPv2 agent or manager.

15.5.2 SNMPv2 party

An SNMPv2 party is a conceptual, virtual execution environment whose operation is restricted, for security or other purposes, to an administratively defined subset of all possible operations of a particular SNMPv2 entity (please refer to 15.5.1, “SNMPv2 entity” on page 536). Architecturally, each SNMPv2 party comprises:
• A single, unique party identity.
• A logical network location at which the party executes, characterized by a transport protocol domain and transport addressing information.
• A single authentication protocol and associated parameters by which all protocol messages originated by the party are authenticated as to origin and integrity.
• A single privacy protocol and associated parameters by which all protocol messages received by the party are protected from disclosure.

15.5.3 GetBulkRequest

The GetBulkRequest is defined in RFC 1905 and is thus part of the protocol operations. A GetBulkRequest is generated and transmitted as a request of an SNMPv2 application. The purpose of the GetBulkRequest is to request the transfer of a potentially large amount of data, including, but not limited to, the efficient and rapid retrieval of large tables. The GetBulkRequest is more efficient than the GetNextRequest in case of retrieval of large MIB object tables. The syntax of the GetBulkRequest is:

\[ \text{GetBulkRequest} \{ \text{non-repeaters} = N, \text{max-repetitions} = M \} \]
\[
( \text{RequestedObjectName1}, \\
\quad \text{RequestedObjectName2}, \\
\quad \text{RequestedObjectName3} )
\]

Where:

- **RequestedObjectName1, 2, 3** MIB object identifier such as sysUpTime etc. The objects are in a lexicographically ordered list. Each object identifier has a binding to at least one variable. For example, the object identifier ipNetToMediaPhysAddress has a variable binding for each IP address in the ARP table and the content is the associated MAC address.

- **N** Specifies the non-repeaters value, which means that you request only the contents of the variable next to the object specified in your request of the first N objects named between the parentheses. This is the same function as provided by the GetNextRequest.
M Specifies the max-repetitions value, which means that you request from the remaining (number of requested objects - N) objects the contents of the M variables next to your object specified in the request. Similar to an iterated GetNextRequest but transmitted in only one request.

With the GetBulkRequest you can efficiently get the contents of the next variable or the next M variables in only one request.

Assume the following ARP table in a host that runs an SNMPv2 agent:

<table>
<thead>
<tr>
<th>Interface-Number</th>
<th>Network-Address</th>
<th>Physical-Address</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10.0.0.51</td>
<td>00:00:10:01:23:45</td>
<td>static</td>
</tr>
<tr>
<td>1</td>
<td>9.2.3.4</td>
<td>00:00:10:54:32:10</td>
<td>dynamic</td>
</tr>
<tr>
<td>2</td>
<td>10.0.0.15</td>
<td>00:00:10:98:76:54</td>
<td>dynamic</td>
</tr>
</tbody>
</table>

An SNMPv2 manager sends the following request to retrieve the sysUpTime and the complete ARP table:

GetBulkRequest [ non-repeaters = 1, max-repetitions = 2 ]
( sysUpTime,
  ipNetToMediaPhysAddress,
  ipNetToMediaType )

The SNMPv2 entity acting in an agent role responds with a response PDU:

Response (( sysUpTime.0 = "123456" ),
( ipNetToMediaPhysAddress.1.9.2.3.4 = "000010543210" ),
( ipNetToMediaType.1.9.2.3.4 = "dynamic" ),
( ipNetToMediaPhysAddress.1.10.0.0.51 = "000010012345" ),
( ipNetToMediaType.1.10.0.0.51 = "static" ))

The SNMPv2 entity acting in a manager role continues with:

GetBulkRequest [ non-repeaters = 1, max-repetitions = 2 ]
( sysUpTime,
  ipNetToMediaPhysAddress.1.10.0.0.51,
  ipNetToMediaType.1.10.0.0.51 )

The SNMPv2 entity acting in an agent role responds with:
Response (( sysUpTime.0 = "123466" ),
    ( ipNetToMediaPhysAddress.2.10.0.0.15 = "000010987654" ),
    ( ipNetToMediaType.2.10.0.0.15 = "dynamic" ),
    ( ipNetToMediaNetAddress.1.9.2.3.4 = "9.2.3.4" ),
    ( ipRoutingDiscards.0 = "2" ))

This response signals the end of the table to the SNMPv2 entity acting in a manager role. With the GetNextRequest you would have needed four requests to retrieve the same information. If you had set the max-repetition value of the GetBulkRequest to three, in this example, you would have needed only one request.

15.5.4 InformRequest

An InformRequest is generated and transmitted as a request from an application in an SNMPv2 manager entity that wishes to notify another application, acting also in an SNMPv2 manager entity, of information in the MIB view\(^1\) of a party local to the sending application. The packet is used as an indicative assertion to the manager of another party about information accessible to the originating party (manager-to-manager communication across party boundaries). The first two variables in the variable binding list of an InformRequest are sysUpTime.0 and snmpEventID.i\(^2\) respectively. Other variables may follow.

15.6 MIB for SNMPv2

This MIB defines managed objects that describe the behavior of the SNMPv2 entity.

Note:
This is not a replacement of the MIB-II.

Following are some object definitions to get an idea of the contents:

\begin{verbatim}
sysName OBJECT-TYPE
    SYNTAX       DisplayString (SIZE (0..255))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION

1\(^{\text{A MIB view is a subset of the set of all instances of all object types defined according to SMI.}}\)
2\(^{\text{snmpEventID.i is an SNMPv2 manager-to-manager MIB object that shows the authoritative identification of an event.}}\)
"An administratively-assigned name for this managed node. By convention, this is the node's fully-qualified domain name. If the name is unknown, the value is the zero-length string."
::= { system 5 }

warmStart NOTIFICATION-TYPE
STATUS current
DESCRIPTION
"A warmStart trap signifies that the SNMPv2 entity, acting in an agent role, is reinitializing itself such that its configuration is unaltered."
::= { snmpTraps 2 }

15.7 The new administrative model

It is the purpose of the administrative model for SNMPv2 to define how the administrative framework is applied to realize effective network management in a variety of configurations and environments.

The model entails the use of distinct identities for peers that exchange SNMPv2 messages. Thus, it represents a departure from the community-based administrative model of the original SNMPv1. By unambiguously identifying the source and intended recipient of each SNMPv2 message, this new strategy improves upon the historical community scheme both by supporting a more convenient access control model and allowing for effective use of asymmetric (public key) security protocols in the future. Please refer to Figure 214 for the new message format.
Figure 214. SNMP Version 2 message format

- PDU

  Includes one of the following protocol data units:
  - GetNextRequest
  - GetRequest
  - Inform
  - Report
  - Response
  - SNMPv2-Trap
  - SetRequest

The GetBulkRequest has a different PDU format, as shown above (refer to 15.5.3, “GetBulkRequest” on page 537).

**Note:**

The SNMP trap now has the same format as all the other requests.
• SnmpMgmtCom (SNMP Management Communication)
  Adds the source party ID (srcParty), the destination party ID (dstParty) and the context to the PDU. The context specifies the SNMPv2 context containing the management information referenced by the communication.

• SnmpAuthMsg
  This field is used as authentication information from the authentication protocol used by that party. The SnmpAuthMsg is serialized according to ASN.1 BER\(^3\) and can then be encrypted.

• SnmpPrivMsg SNMP Private Message
  An SNMPv2 private message is an SNMPv2 authenticated management communication that is (possibly) protected from disclosure. A private destination (privDst) is added to address the destination party.

  The message is then encapsulated in a normal UDP/IP datagram and sent to the destination across the network.

  For further information please refer to the above mentioned RFCs.

### 15.8 Simple Network Management Protocol Version 3 (SNMPv3)

SNMPv3 is described in RFC 2570, RFC 2571, RFC 2572, RFC 2573 and RFC 2574. SNMPv3 is an extension to the existing SNMP architecture.

SNMPv3 supports the following:
- A new SNMP message format
- Authentication for messages
- Security for messages
- Access control
- Continued support for SNMPv2

The User-based Security Model (USM), described in RFC 2574, specifies using MD5 and hashing algorithms. This provides data integrity, security and privacy. There is support for the authentication protocols HMAC-MD5-96, HMAC-SHA-96, and optional support for the encryption protocol CBC-DES.

The View-based Access Control Model (VACM) in RFC 2575, shows how to define views which are subsets of the full MIB tree. Access control is then available for these views.

\(^3\) ASN.1 BER specifies the Basic Encoding Rules for OSI Abstract Syntax Notation One, according to ISO 8825.
Since SNMP has a modular structure, changes to individual modules do not impact the other modules directly. This allows you to easily define SNMPv3 over the existing model. For example, to add a new SNMP message format, it is sufficient to upgrade the message processing model. Furthermore, since it is needed to support SNMPv1 and SNMPv2 messages as well, it can be achieved by adding the new SNMPv3 message module into the message processing subsystem. The Figure 215 illustrates this structure.

15.8.1 Single authentication and privacy protocol

The authentication protocol provides a mechanism by which SNMPv3 management communications, transmitted by a party, can be reliably identified as having originated from that party.

The privacy protocol provides a mechanism by which SNMPv3 management communications transmitted to a party are protected from disclosure.

Principal threats against which the SNMPv3 security protocol provides protection are:

- Modification of information
- Masquerade
- Message stream modification
- Disclosure

The following security services provide protection against the above threats:

- Data integrity
Provided by the MD5 message digest algorithm. A 128-bit digest is calculated over the designated portion of a SNMPv3 message and included as part of the message sent to the recipient.

- Data origin authentication
  Provided by prefixing each message with a secret value shared by the originator of that message and its intended recipient before digesting.

- Message delay or replay
  Provided by including a timestamp value in each message.

- Data confidentiality
  Provided by the symmetric privacy protocol which encrypts an appropriate portion of the message according to a secret key known only to the originator and recipient of the message. This protocol is used in conjunction with the symmetric encryption algorithm, in the cipher block chaining mode, which is part of the Data Encryption Standard (DES). The designated portion of an SNMPv3 message is encrypted and included as part of the message sent to the recipient.

15.9 References

The following RFCs define the Simple Network Management Protocol and the information kept in a system:

- RFC 1052 – IAB Recommendations for the Development of Internet Network Management Standards
- RFC 1085 – ISO Presentation Services on Top of TCP/IP-Based Internets
- RFC 1155 – Structure and Identification of Management Information for TCP/IP-Based Internets
- RFC 1157 – A Simple Network Management Protocol (SNMP)
- RFC 1213 – Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II
- RFC 1215 – Convention for Defining Traps for Use with the SNMP
- RFC 1228 – SNMP-DPI: Simple Network Management Protocol Distributed Programming Interface
- RFC 1239 – Reassignment of Experimental MIBs to Standard MIBs
- RFC 1351 – SNMP Administrative Model
- RFC 1352 – SNMP Security Protocols
- RFC 1441 – Introduction to Version 2 of the Internet-Standard Network Management Framework
- RFC 1748 – IEEE 802.5 Token-Ring MIB
• RFC 1901 – Introduction to Community-Based SNMPv2
• RFC 1904 – Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)
• RFC 1906 – Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)
• RFC 1907 – Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)
• RFC 1909 – An Administrative Infrastructure for SNMPv2
• RFC 1910 – User-Based Security Model for SNMPv2
• RFC 2011 – SNMPv2 Management Information Base for the Internet Protocol Using SMiv2
• RFC 2012 – SNMPv2 Management Information Base for the Transmission Control Protocol Using SMiv2
• RFC 2013 – SNMPv2 Management Information Base for the User Datagram Protocol Using SMiv2
• RFC 2570 – Introduction to Version 3 of the Internet-standard Network Management Framework.
• RFC 2571 – An Architecture for Describing SNMP Management Frameworks.
• RFC 2573 – SNMP Applications
• RFC 2574 – User-based Security Model (USM) for SNMPv3.
• RFC 2575 – View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)
• RFC 2578 – Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)
• RFC 2580 – Conformance Statements for SMiv2
• RFC 2742 – Definitions of Managed Objects for Extensible SNMP Agents