

# Network Management (SNMP)

November 21, 2006

# The Level Of Management Protocols

- Originally, many wide area networks included management protocols as part of their link level protocols. If a packet switch began misbehaving, a network manager could instruct a neighboring packet switch to send a special *control packet* which would cause the switch to suspend normal operation and respond to commands from the manager.
- Unlike a homogeneous WAN, the Internet does not have a single link level protocol, and thus requires a new network management paradigm. Also, the network manager may not be on the same physical network as the offending device.
- As a result, management tools are moved above the transport layer and communicate using TCP/IP.

# Management Protocols (Continued)

- Advantages include one set of protocols for all managed devices.
- A disadvantage is that this approach assumes lower levels are working correctly. If a router's routing table becomes corrupt, it may be impossible to reboot the machine from a remote site. If a router's OS crashes, it will be impossible to reach the application program that implements the internet management protocols even if the router can still process hardware interrupts and forward packets.

- Despite the disadvantages application level management software has worded well in practice.
- Figure 29.1
- Client software runs on the manager's workstation. Each participating network system (router, printer) runs a server program (a *management agent*).
- Network management software uses an authentication mechanism, possibly with multiple levels of authorization.

- TCP/IP protocols divide the management problem into two parts and specify separate standards for each part.
- The first part concerns communication of information (formats).
- The second part concerns the data being managed. Which data items must a managed device keep, what are their names, and what syntax is used.

# A Standard Network Management Protocol

- The TCP/IP standard for network management is the *Simple Network Management Protocol (SNMP)*.
- It has evolved through three generations known as *SNMPv1*, *SNMPv2*, and *SNMPv3*.
- Changes have been minor and many features are backward compatible.
- The SNMP standard defines a small set of operations.

# A Standard For Management Information

- A router keeps statistics on the status of its interfaces, incoming and outgoing traffic, dropped datagrams, and error messages generated. SNMP does not specify exactly which data can be accessed on which devices.
- A separate standard known as the *Management Information Base (MIB)* specifies the data items a managed device must keep, the operations allowed on each, and the meanings. For example, MIB specifies that the software must keep a count of all octets that arrive over each network interface and that management software can only read the count.
- MIB divides information into many categories: Figure 29.2

# Examples Of MIB Variables

- Now there are many (more than 100) individual MIB documents that each specify the variables (total number is more than 10,000) for a specific type of device (hardware bridge, UPS, DSL modem, Ethernet switch). Figure 29.3
- Many of the items listed are numeric, but the MIB also defines more complex structures. For example, the MIB variable *ipRoutingTable* refers to an entire routing table. Additional variables define the contents. MIB specifies the logical definition of each data item, not the data structure used to store the information.

# The Structure Of Management Information

- A separate set of rules (the *Structure of Management Information (SMI)*) define and identify MIB variables. It places restrictions on the types of variables, specifies the rules for naming those variables, and creates rules for creating variable types. For example, the SMI standard includes definitions of terms like *IpAddress* (4 octet string) and *Counter* (32 bit), and specifies that they are terms used to define MIB variables. The rules describe how the MIB refers to tables of values.

# Formal Definitions Using ASN.1

- The SMI standard specifies that all MIB variables must be defined and referenced using ISO's *Abstract Syntax Notatin 1 (ASN.1)* which is a formal language with two main features: a notation used in documents that humans read and a compact encoded representation of the same information used in communication protocols. For example, instead of saying that a variable contains an integer value, a protocol designer must state the exact form and range of each numeric value. This is especially important when implementations include heterogeneous computers that do not all use the same representations.
- In addition ASN.1 defines a set of *Basic Encoding Rules (BER)* that specify how to encode both names and data items in a message.

# Structure And Representation Of MIB Object Names

- Names used for MIB variables are taken from the *object identifier* namespace administered by ISO and ITU. The key idea is to provide a namespace in which all possible objects can be designated.
- The object identifier namespace is *absolute (global)*, meaning that names are structured to make them globally unique. It is hierarchical with authority subdivided.
- The root is unnamed, but has three direct descendants managed by: ISO, ITU, and jointly by ISO and ITU. The descendants are assigned both short text strings and integers that identify them.
- ISO has allocated one subtree for use by other national or international standards organizations.
- Figure 29.4

# MIB Object Names (Continued)

- The name is the sequence of numeric labels on the nodes along a path from the root to the object. For example, the name 1.3.6.1.2 denotes the node labeled *mgmt*.
- Figure 29.5
- A MIB variable named *ipInReceives* is represented as *iso.org.dod.internet.mgmt.mib.ip.ipInReceives* or *1.3.6.1.2.1.4.3*. When management protocols use names in messages, each name has a suffix appended. For simple variables, the suffix 0 refers to the instance of the variable with that name.
- *ipAddrTable* is *iso.org.dod.internet.mgmt.mib.ip.ipAddrTable* or *1.3.6.1.2.1.4.20*. We think of the IP address table as a one-dimensional array, but the router may keep this information in many variables or use a linked list.

# MIB Object Names (Continued)

- There is a significant difference in the way programmers use arrays and the way management software uses tables in MIB.
- MIB tables append a suffix onto the name to select a specific element in the table. To specify the netmask in the IP address table entry corresponding to 128.10.2.3 one uses the name *1.3.6.1.2.1.4.20.1.3.128.10.2.3*. This allows searching tables without knowing the number of items or the type of data.

# Simple Network Management Protocol

- One might expect a large number of commands. Some early protocols supported *reboot*, *add*, *delete*, *disable*, *enable*, *remove* but complexity results.
- SNMP takes an interesting alternative approach by casting all operations in a *fetch-store* paradigm. Conceptually, there are only two commands. All other operations are defined as side-effects of these operations. For example, to reboot declare a data item that gives the time until the next reboot and allow the manager to assign the item a value (zero).
- The chief advantages of using a fetch-store paradigm are stability, simplicity, and flexibility. SNMP is especially stable because its definitions remain fixed. It is simple to implement and debug.
- From a manager's viewpoint, SNMP remains hidden. Vendors sell software that offers a GUI.

# SNMP (Continued)

- SNMP offers other operations: Figure 29.6
- SNMP specifies that all operations must be *atomic*. If a single message specifies on multiple variables, the server must either perform all operations or none of them. No assignments will be made if any of them are in error.
- The *trap* operation allows managers to program servers to send information when an event occurs, say one of the interfaces goes down.

# SNMP Message Format

- SNMP messages do not have fixed fields, but use the standard ASN.1 encoding. Each item in the grammar consists of a descriptive name followed by a declaration of the item's type.
- *msgVersion* INTEGER (0..2147483647) declares the name *msgVersion* to be a nonnegative integer less than or equal to 2147483647.
- Figure 29.7
- Each SNMP message consists of four main parts: an integer that identifies the *version*, additional header data, a set of security parameters, and a data area.
- Figure 29.8

# SNMP Message Format (Continued)

- The data area in an SNMP message is divided into *Protocol Data Units (PDUs)*. Each consists of a client request or an agent response. SNMPv3 allows each PDU to be sent as plain text or to be encrypted for privacy. Thus the grammar specifies a *CHOICE*. In programming language terminology, the concept is known as a *discriminated union*.
- An encrypted PDU begins with an identifier of the *engine* that produced it. The item labeled *data* in the *ScopedPDU* definition has a type *ANY* because field *contextName* defines the exact details of the item. The data must consist of one of the PDUs illustrated in Figure 29.9
- Figure 29.10 shows the definition of a *get-request*.

# An Example Encoded SNMP Message

- The encoded form of ANS.1 uses variable-length fields to represent items. Each field begins with a header that specifies the type of object and its length in bytes. For example, each *SEQUENCE* begins with an octet containing 0x30; the next octet specifies the number of the following octets that constitute the sequence.
- Figure 29.11 contains an example SNMP message, a *get-request* that specifies data item *sysDescr*.

# New Features In SNMPv3

- The primary changes arise in the areas of security and administration. The goals are twofold. First, to have both general and flexible security policies, making it possible for the interactions between a manager and managed devices to adhere to the security policies an organization specifies. Second, the system is designed to make administration of security easy.
- SNMPv3 includes facilities for several aspects of security and allows each to be configured independently. For example, *message authentication*, *privacy*, and *authorization* and *view-based access control*.
- SNMPv3 allows *remote configuration* allowing an authorized manager to change the configuration of the security items listed above without having physical access.