What Is SNMP?

The core of SNMP is a simple set of operations (and the information these operations gather) that gives administrators the ability to change the state of some SNMP-based device. For example, you can use SNMP to shut down an interface on your router or check the speed at which your Ethernet interface is operating. SNMP can even monitor the temperature on your switch and warn you when it is too high.

SNMP usually is associated with managing routers, but it's important to understand that it can be used to manage many types of devices. While SNMP's predecessor, the Simple Gateway Management Protocol (SGMP) , was developed to manage Internet routers, SNMP can be used to manage Unix systems, Windows systems, printers, modem racks, power supplies, and more. Any device running software that allows the retrieval of SNMP information can be managed. This includes not only physical devices but also software, such as web servers and databases.

Another aspect of network management is network monitoring ; that is, monitoring an entire network as opposed to individual routers, hosts, and other devices. Remote Network Monitoring (RMON) was developed to help us understand how the network itself is functioning, as well as how individual devices on the network are affecting the network as a whole. It can be used to monitor not only LAN traffic, but WAN interfaces as well. We discuss RMON in more detail later in this chapter and in Chapter 2.

RFCs and SNMP Versions

The Internet Engineering Task Force (IETF) is responsible for defining the standard protocols that govern Internet traffic, including SNMP. The IETF publishes Requests for Comments (RFCs), which are specifications for many protocols that exist in the IP realm. Documents enter the standards track first as *proposed* standards, then move to *draft* status. When a final draft is eventually approved, the RFC is given *standard* status—although there are fewer completely approved standards than you might think. Two other standards-track designations, *historical* and *experimental* , define (respectively) a document that has been replaced by a newer RFC and a document that is not yet ready to become a standard. The following list includes all the current SNMP versions and the IETF status of each (see Appendix D for a full list of the SNMP RFCs):

- SNMP Version 1 (SNMPv1) is the initial version of the SNMP protocol. It's defined in RFC 1157 and is a historical IETF standard. SNMPv1's security is based on communities, which are nothing more than passwords: plain-text strings that allow any SNMP-based application that knows the strings to gain access to a device's management information. There are typically three communities in SNMPv1: *read-only*, *read-write*, and *trap*. It should be noted that while SNMPv1 is historical, it is still the primary SNMP implementation that many vendors support.
- SNMP version 2 (SNMPv2) is often referred to as community-string-based SNMPv2. This version of SNMP is technically called SNMPv2c, but we will refer to it throughout this book simply as SNMPv2. It's defined in RFC 3416, RFC 3417, and RFC 3418.
- SNMP version 3 (SNMPv3) is the latest version of SNMP. Its main contribution to network management is security. It adds support for strong authentication and private communication between managed entities. In 2002, it finally made the transition from draft standard to full standard. The following RFCs define the standard: RFC 3410, RFC 3411, RFC 3412, RFC 3413, RFC 3414, RFC 3415, RFC 3416, RFC 3417, RFC 3418, and RFC 2576. Chapter 3 provides a thorough treatment of SNMPv3 and Chapter 6 goes through the SNMPv3 agent configuration for Net-SNMP and Cisco. While it is good news that SNMPv3 is a full standard, vendors are notoriously slow at adopting new versions of a protocol. While SNMPv1 has been transitioned to historical, the vast majority of vendor implementations of SNMP are SNMPv1 implementations. Some large infrastructure vendors like Cisco have supported SNMPv3 for quite some time, and we will undoubtedly begin to see more vendors move to SNMPv3 as customers insist on more secure means of managing networks.

The official site for RFCs is http://www.ietf.org/rfc.html. One of the biggest problems with RFCs, however, is finding the one you want. It is a little easier to navigate the RFC index at Ohio State University (http://www.cse.ohio-state.edu/cs/Services/rfc/index.html).

Managers and Agents

In the previous sections, we've vaguely referred to SNMP-capable devices and network management stations. Now it's time to describe what these two things really are. In the world of SNMP, there are two kind of entities: managers and agents . A *manager* is a server running some kind of software system that can handle management tasks for a network. Managers are often referred to as Network Management Stations (NMSs).[*] An NMS is responsible for polling and receiving traps from agents in the network. A *poll*, in the context of network management, is the act of querying an agent (router, switch, Unix server, etc.) for some piece of information. This information can be used later to determine if some sort of catastrophic event has occurred. A *trap* is a way for the agent to tell the NMS that something has happened. Traps are sent asynchronously, not in response to queries from the NMS. The NMS is further responsible for performing an action[†] based upon the information it receives from the agent. For example, when your T1 circuit to the Internet goes down, your router can send a trap to your NMS. In turn, the NMS can take some action, perhaps paging you to let you know that something has happened.

The second entity, the *agent*, is a piece of software that runs on the network devices you are managing. It can be a separate program (a daemon, in Unix language), or it can be incorporated into the operating system (for example, Cisco's IOS on a router, or the low-level operating system that controls a UPS). Today, most IP devices come with some kind of SNMP agent built in. The fact that vendors are willing to implement agents in many of their products makes the system administrator's or network manager's job easier. The agent provides management information to the NMS by keeping track of various operational aspects of the device. For example, the agent on a router is able to keep track of the state of each of its interfaces: which ones are up, which ones are down, etc. The NMS can query the status of each interface and take appropriate action if any of them are down. When the agent notices that something bad has happened, it can send a trap to the NMS. This trap originates from the agent and is sent to the NMS, where it is handled appropriately. Some devices will send a corresponding "all clear" trap when there is a transition from a bad state to a good state. This can be useful in determining when a problem situation has been resolved. Figure 1-1 shows the relationship between the NMS and an agent.

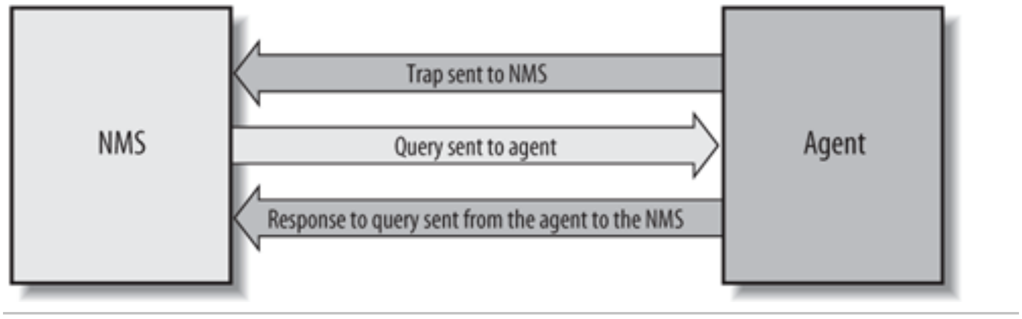

Figure 1-1. Relationship between an NMS and an agent

It's important to keep in mind that polls and traps can happen at the same time. There are no restrictions on when the NMS can query the agent or when the agent can send a trap.

The Structure of Management Information and MIBs

The Structure of Management Information (SMI) provides a way to define managed objects and their behavior. An agent has in its possession a list of the objects that it tracks. One such object is the operational status of a router interface (for example, *up*, *down*, or *testing*). This list collectively defines the information the NMS can use to determine the overall health of the device on which the agent resides.

The Management Information Base (MIB) can be thought of as a database of managed objects that the agent tracks. Any sort of status or statistical information that can be accessed by the NMS is defined in a MIB. The SMI provides a way to define managed objects while the MIB is the definition (using the SMI syntax) of the objects themselves. Like a dictionary, which shows how to spell a word and then gives its meaning or definition, a MIB defines a textual name for a managed object and explains its meaning. Chapter 2 goes into more technical detail about MIBs and the SMI.

An agent may implement many MIBs, but all agents implement a particular MIB called MIB-II [*] (RFC 1213). This standard defines variables for things such as interface statistics (interface speeds, MTU, octets[*] sent, octets received, etc.) as well as various other things pertaining to the system itself (system location, system contact, etc.). The main goal of MIB-II is to provide general TCP/IP management information. It doesn't cover every possible item a vendor may want to manage within its particular device.

What other kinds of information might be useful to collect? First, many draft and proposed standards have been developed to help manage things such as frame relay, ATM, FDDI, and services (mail, Domain Name System (DNS), etc.). A sampling of these MIBs and their RFC numbers includes:

- ATM MIB (RFC 2515)
- Frame Relay DTE Interface Type MIB (RFC 2115)
- BGP Version 4 MIB (RFC 1657)
- RDBMS MIB (RFC 1697)
- RADIUS Authentication Server MIB (RFC 2619)
- Mail Monitoring MIB (RFC 2789)
- DNS Server MIB (RFC 1611)

But that's far from the entire story, which is why vendors, and individuals, are allowed to define MIB variables for their own use.[†] For example, consider a vendor that is bringing a new router to market. The agent built into the router will respond to NMS requests (or send traps to the NMS) for the variables defined by the MIB-II standard; it probably also implements MIBs for the interface types it provides (e.g., RFC 2515 for ATM and RFC 2115 for Frame Relay). In addition, the router may have some significant new features that are worth monitoring but are not covered by any standard MIB. So, the vendor defines its own MIB (sometimes referred to as a *proprietary MIB*) that implements managed objects for the status and statistical information of its new router.

TIP

Simply loading a new MIB into your NMS does not necessarily allow you to retrieve the data/values/objects, etc., defined within that MIB. You need to load only those MIBs supported by the agents from which you're requesting queries (e.g., snmpget, snmpwalk). Feel free to load additional MIBs for future device support, but don't panic when your device doesn't answer (and possibly returns errors for) these unsupported MIBs.

Host Management

Managing host resources (disk space, memory usage, etc.) is an important part of network management. The distinction between traditional system administration and network management has been disappearing over the last decade and is now all but gone. As Sun Microsystems puts it, "The network is the computer." If your web server or mail server is down, it doesn't matter whether your routers are running correctly—you're still going to get calls. The Host Resources MIB (RFC 2790) defines a set of objects to help manage critical aspects of Unix and Windows systems.[*]

Some of the objects supported by the Host Resources MIB include disk capacity, number of system users, number of running processes, and software currently installed. Today, more and more people are relying on service-oriented web sites. Making sure your backend servers are functioning properly is as important as monitoring your routers and other communications devices.

Unfortunately, some agent implementations for these platforms do not implement this MIB since it's not required.

A Brief Introduction to Remote Monitoring (RMON)

Remote Monitoring Version 1 (RMONv1, or RMON) is defined in RFC 2819; an enhanced version of the standard, called RMON Version 2 (RMONv2), is defined in RFC 2021. RMONv1 provides the NMS with packet-level statistics about an entire LAN or WAN. RMONv2 builds on RMONv1 by providing network- and application-level statistics. These statistics can be gathered in several ways. One way is to place an RMON probe on every network segment you want to monitor. Some Cisco routers have limited RMON capabilities built in, so you can use their functionality to perform minor RMON duties. Likewise, some 3Com switches implement the full RMON specification and can be used as full-blown RMON probes.

The RMON MIB was designed to allow an actual RMON probe to run in an offline mode that allows the probe to gather statistics about the network it's watching without requiring an NMS to query it constantly. At some later time, the NMS can query the probe for the statistics it has been gathering. Another feature that most probes implement is the ability to set thresholds for various error conditions and, when a threshold is crossed, alert the NMS with an SNMP trap. You can find a little more technical detail about RMON in the next chapter.