AIX NetView/6000

by J. H. Chou
C. R. Buckman
T. Hemp
A. Himwich
F. Niemi

AIX® NetView®/6000 is a network management system that manages simple network management protocol (SNMP) devices developed by IBM and other vendors. It provides configuration, fault, and performance applications integrated into an advanced end-user interface (EUI), which incorporates a graphic display of network topology and performance as well as system management functions accessible from both graphic and character-based devices. An application builder and event configurator allow users to generate performance applications and provide automation of management tasks specific to their networks. In addition to providing stand-alone distributed network management, AIX NetView/6000 also provides a bidirectional connection to IBM's mainframe-based NetView product to enable central management of the enterprise network from System/370™ and System/390™ NetView.

Several fundamental changes have occurred in the field of information processing in the past decades. One of the most significant changes has been the shift away from the dominance of large mainframe systems to systems that contain PCs (personal computers), workstations, minisupercomputers, and other special-purpose computers. This change in computing technology was accompanied by a change in communications technologies. Today the local area network (LAN) is as ubiquitous as the personal computer. Bridges, which are used to connect LANs at a single site, and routers, which connect LANs at several sites into a single wide area network (WAN), have become important components of the corporate computing environment.

The structure associated with the information processing industry has also changed. Previously, a handful of companies dominated the industry, and corporations accepted proprietary hardware, operating systems, and communications protocols as normal. A company might, for example, have all IBM hardware linked using Systems Network Architecture (SNA), or it might be a DEC** (Digital Equipment Corporation) company with minicomputers communicating via DECnet. Today, a corporation typically deals with several vendors. This has led to demand for industry-wide standards rather than products based on proprietary technology.

TCP/IP

One example of the shift away from proprietary solutions is the emergence of the Internet Protocol suite as an industry standard. The Internet Protocol suite, usually referred to as TCP/IP (Transmission Control Protocol/Internet Protocol) after its base protocols, is a set of protocols that offers various services including connectionless and connection-oriented data transfer, file transfer, electronic mail, and remote operations. It is available on a wide variety of computers and operating systems from many vendors. (IBM offers it, for example, on machines ranging from

Personal System/2* [PS/2*] running DOS and Operating System/2* [OS/2*], to System/390* running VM and MVS [virtual machine and Multiple Virtual Storage]). Since this suite of protocols is offered by many suppliers, businesses have experienced an increase in connectivity between their various systems even as the heterogeneity of their networks increased, a marked change from the days when the use of systems was often accompanied by difficulties in exchanging data between disparate systems.

The SNMP management framework is very similar to the OSI network management framework.

However, the benefit of increased connectivity and choice in hardware has had some drawbacks. Homogeneous networks (those composed of systems from a single computer manufacturer) in general are easily managed using a particular management system. Heterogeneous networks offer more of a challenge. Since the devices are from various manufacturers, a user might have to work with a number of different management systems.

In order to solve this problem, management protocols are being developed that allow centralized management of heterogeneous networks. The Open Systems Interconnection Common Management Information Protocol (OSI CMIP) is the protocol that most users and vendors expect to support in the future. The current industry standard, however, is Simple Network Management Protocol (SNMP), one of the TCP/IP protocols.

SNMP

The SNMP management framework[1,2] is very similar to the OSI network management framework. In addition to the management protocol,[2] which is used to communicate between the managing system (manager) and the managed system (agent), a management information base (MIB) is also defined;[3-5] this provides a set of common managed object definitions. In addition, the framework also allows vendors to define new, device-specific MIBs using a subset of the OSI ASN.1 (Abstract Syntax Notation 1) specification language.

SNMP enables a user to purchase disparate devices from various vendors and manage them all using a single network management system. This paper describes some of the features of AIX* NetView*/6000, an SNMP network management system that runs on IBM's RISC System/6000* processors under AIX (Advanced Interactive Executive*—IBM's flavor of UNIX**).

AIX NetView/6000 is a network management system that manages SNMP devices and monitors Internal Protocol (IP) devices. Based on technology developed for Hewlett-Packard's OpenView** Network Node Manager product, it is designed to offer an integrated network management system for heterogeneous TCP/IP networks. In addition to the usual configuration, it offers facilities to allow menu-driven development of additional applications, an interface for network management automation, and a bidirectional connection to System/370* and System/390 NetView, in order to allow centralized management of an enterprise network containing both SNA and TCP/IP elements.

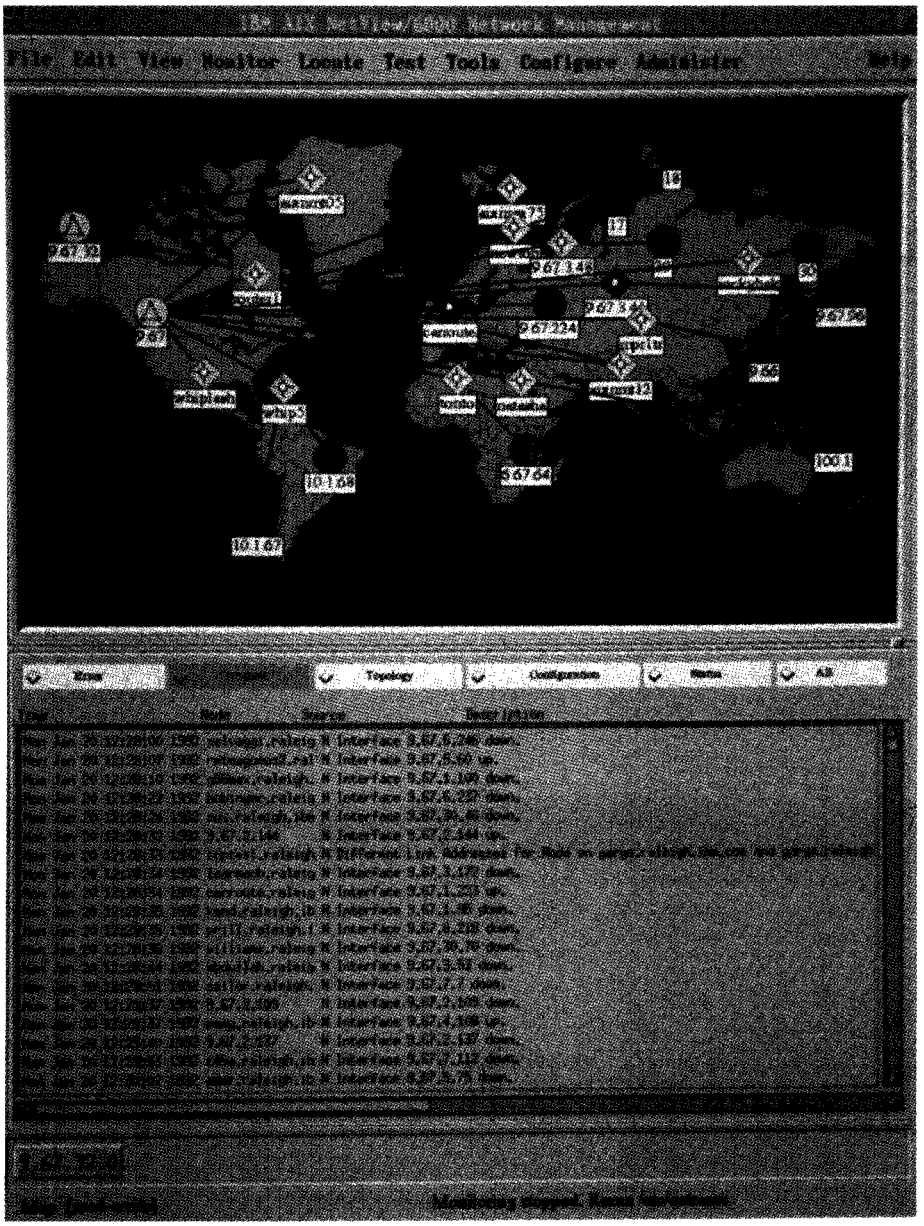
SNMP management applications

AIX NetView provides a number of network management applications. The primary ones are a network configuration application, a set of fault monitoring and diagnostic functions, and performance applications that can be fully customized by a user to monitor device performance and generate an error notification when defined thresholds are exceeded.

Configuration. One of the most onerous tasks faced by a network administrator is the chore of loading data into the network elements of a network management system and keeping the database current. This task has been automated in AIX NetView by a discovery process that generates and maintains a network topology database.

The discovery process is driven by a combination of the SNMP protocols and utilization of the TCP/IP

Figure 1 AIX NetView/6000 main window



network. The efficiency of the system is directly proportional to the number of managed network elements (nodes) that support SNMP agents. These SNMP agents can provide AIX NetView with "hints" about new network elements.

The list for new node discovery, called the "hint list," is generated when a network node is found that supports an SNMP agent. The node is polled for information from its MIB. Data, including the address translation table and the default routing entry, are retrieved. When a new node is discovered, it is added to the topology database and also to the list of nodes that is being monitored. If the node supports an SNMP agent, then the system configuration information, which includes the

system description, system object identifier, IP forwarding status, IP address table, interface table, system location, and system contact, are retrieved. The SNMP GET requests retrieve the values and the data are stored in the database. This system configuration information is polled periodically. Additional information can be stored with each node and network; this allows users to associate enterprise-specific data (such as branch office number or department number) with the predefined generic data generated by AIX NetView.

Discovery need not be completely automatic; users may create "seed" files containing lists of nodes, if desired. The rate of discovery is adjusted automatically, being greatest on initial startup and diminishing as nodes are added.

Fault. Once a database of managed network elements is established, the primary function of a network management system is to detect problems with devices on the network and either notify an operator of a problem, or, ideally, act in an autonomous manner to correct the problem. Fault management applications are those that detect problems, assist an operator with problem determination (in order to isolate the specific cause of a problem), or provide a mechanism for automatic error recovery.

Problem detection. The two primary methods of automated error detection are asynchronous notification (e.g., OSI alarms, SNMP traps, or SNA alerts) of an error by a managed device and synchronous polling by a network manager. The former is preferable since it consumes fewer network resources and less processor time, but polling is always required since a failure on a network element may render it unable to send an error notification. Both mechanisms are supported by AIX NetView.

Asynchronous notification of errors in SNMP is done through the trap mechanism. A trap is a message carrying data about a failure. SNMP defines certain generic traps common to all devices; additionally, an enterprise-specific trap mechanism allows vendors to define new traps for their own devices. AIX NetView understands both generic and enterprise-specific traps. An operator is notified whenever a trap is received.

AIX NetView also provides an SNMP subagent that can be installed on machines running AIX 3.2. This subagent generates enterprise-specific traps when errors are written to the system error log. This provides some integration of system and network management functionality at the AIX NetView console.

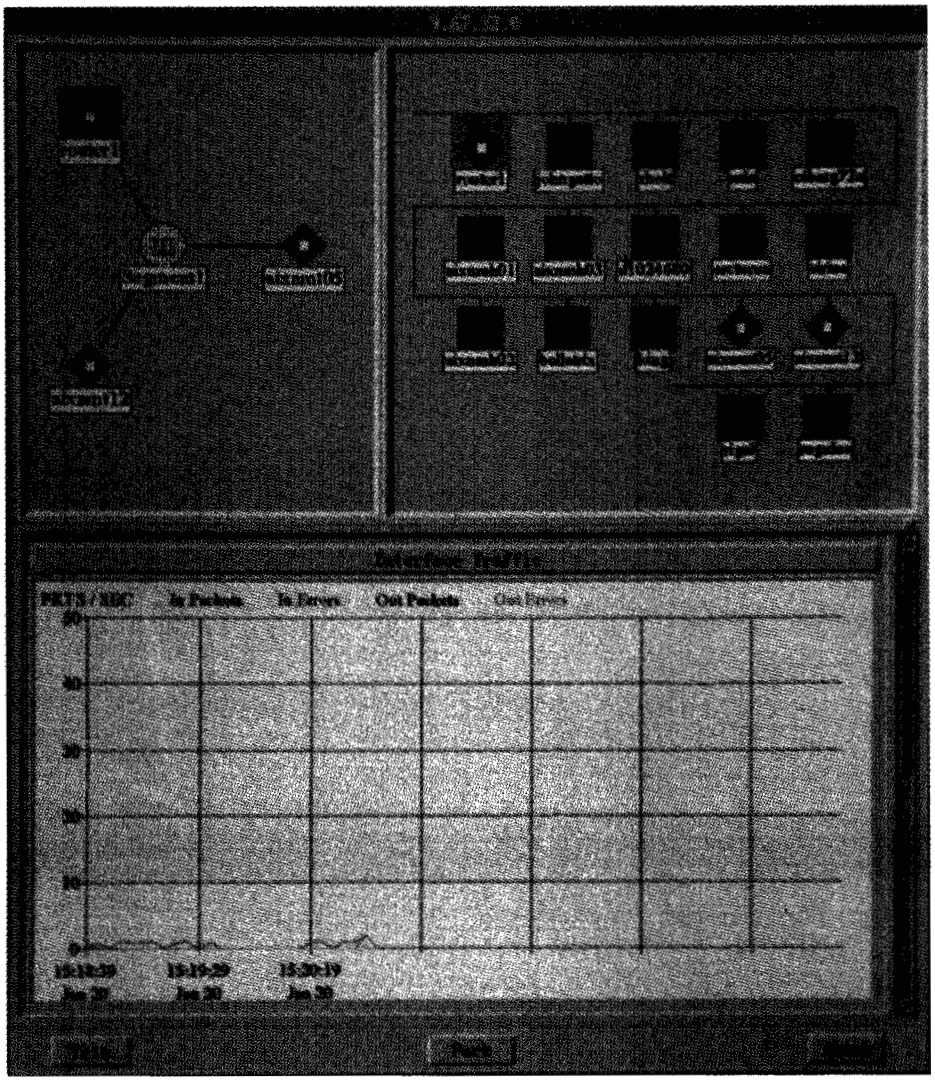
In addition to receiving traps, AIX NetView actively polls all managed devices on the network. There are two types of polling that may be done. One type uses the TCP/IP Internet Control Message Protocol (ICMP) echo mechanism ("ping") and SNMP requests against the standard MIB to verify that a device remains active on a network. This is automatically done at predefined (but user-configurable) intervals against all managed devices. In addition to this, an operator may define thresholds to be monitored on selected devices; when a threshold is exceeded, an error notification is also generated. (This is discussed further in the section on performance.)

Changes in network topology or in the status of network devices is reflected in AIX NetView's end-user interface (EUI). The EUI, based on the industry standard X11R4 window system and Motif** window manager, is centered around a two-pane main window (Figure 1). The top pane (Internet View Pane) shows a color-coded map of all known networks (shown as circles) and the gateways connecting the networks (shown as diamonds). The bottom pane (Control Desk) shows a list of all received events.

When more detail is desired, secondary windows may be brought up that show a specific network and the devices it contains (Figure 2). This secondary window contains a view of the segments comprising the network and the gateways connecting the segment to other networks on the top left, together with a detailed view of a segment in the network showing attached devices on the top right. The bottom view contains applications run against devices in that network; in the case shown, the application is a graph showing interface traffic on a node. Any of these three view windows may be closed.

Problem determination. Once an operator has been notified of an error, the emphasis shifts to the problem determination and recovery phase. Although there may be a limitless number of different errors that may occur in a network, all errors can typically be classified into one of two types, either device errors or connectivity errors. Device errors are those caused by a hardware or

Figure 2 AIX NetView/6000 secondary window, showing more detailed view of a network

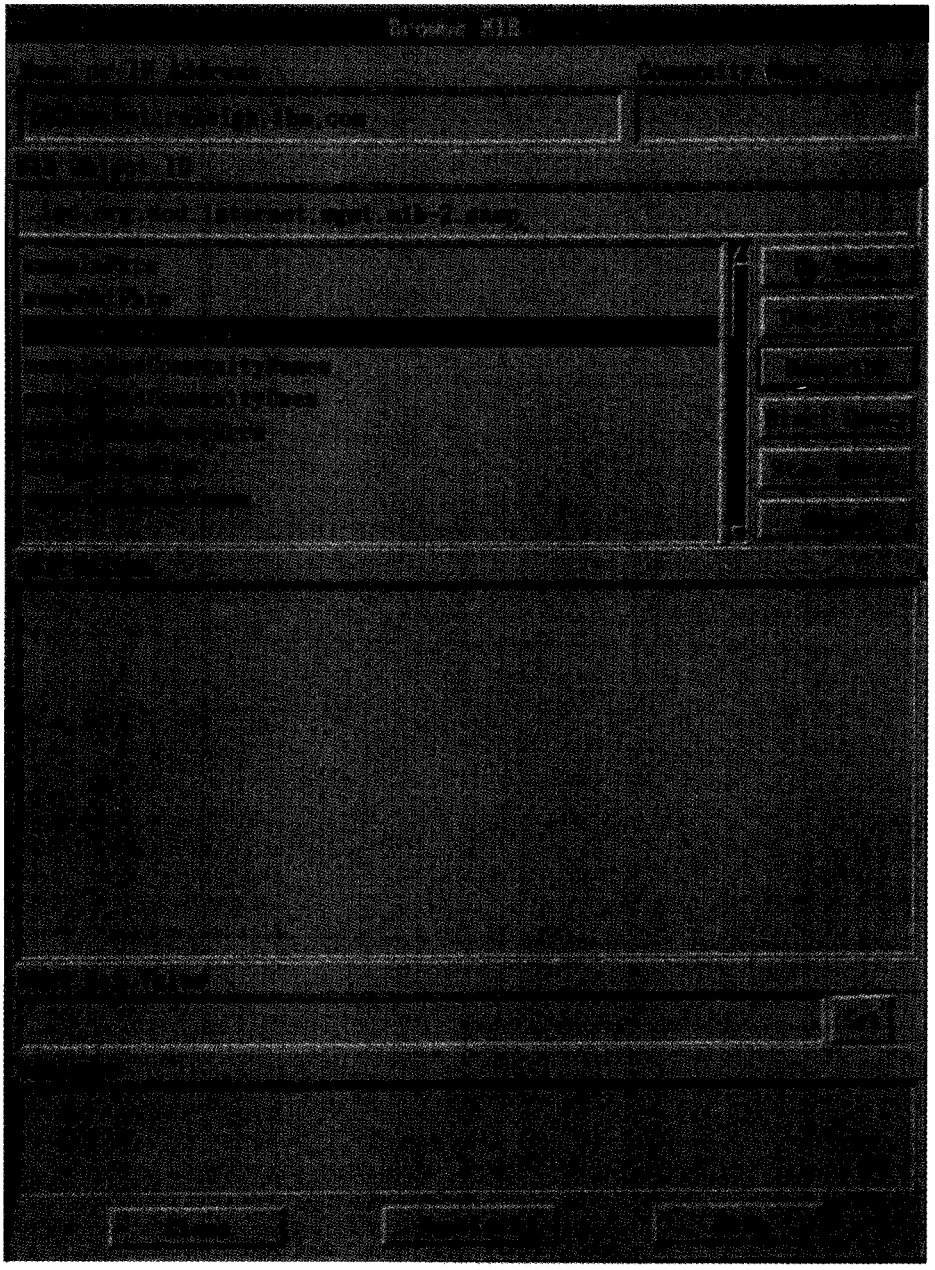


software failure inside a specific node. Connectivity errors are those caused by the failure of a connection between one or more nodes; these usually arise as a result of a device error. For example, the failure of a LAN interface card on a gateway between two networks is a device error that will cause connectivity errors between devices on the two networks.

AIX NetView provides a number of tools to assist with the determination of problems. An operator may manually initiate a poll of a device at various protocol layers, including IP, TCP, and SNMP. This is used to verify connectivity between the management station and the device. Connectivity between two devices on the network at the IP level may also be verified if one of the devices supports a remote ping protocol (currently, this is an extension to a TCP/IP protocol, available for systems running the AIX NetView subagent and for some Hewlett-Packard workstations).

In addition to verifying connectivity, an operator may use AIX NetView to remotely view various

Figure 3 The AIX NetView/6000 MIB Browser



data about a specific device. Some examples of data that may be displayed are a list of network interfaces, including TCP/IP and link addresses (physical addresses), the routing table for the device, a list of TCP/IP services, and a list of currently active TCP/IP connections. In addition to these data, an MIB browsing tool is provided that allows an operator to view any data available from an SNMP agent. The MIB Browser, shown in Figure 3, is one of the most powerful tools that an

operator can use in network problem determination. In the illustration, the MIB tree can be traversed and the MIB values from a selected device can be displayed.

Using the mouse-driven interface, the operator can traverse the entire MIB of a device. Any MIB variable can be retrieved. Numerical MIB values can also be graphed in real time. An operator may also use this tool to set MIB values if the MIB definition specifies the MIB variable as read/write; this is one mechanism by which device operating parameters may be modified.

AIX NetView provides two performance applications that assist in monitoring performance aspects of a network.

When an operator is unable to determine or correct the source of a problem on a device using SNMP, the next course of action is to enter the device and perform system administration and maintenance. If the failing device is an AIX or HP-UX** (Hewlett-Packard's version of UNIX) system, AIX NetView will invoke the appropriate system management tools (System Management Interface Tool [SMIT] or System Administration Manager [SAM]) so an operator can perform remote operations from the main network management console. For other devices, a Telnet[6] session is brought up so the operator can log in to the device.

All the tools described above are invoked either from the menu bar in the AIX NetView main window or from a pop-up context menu. A user may add additional tools to these menus through a registration file. This increases operator productivity by providing integrated access to all problem determination tools from a single location.

Automated error recovery. Optimally, a management system should be able to operate autonomously to correct errors. Unfortunately, this is a difficult problem, one that is unlikely to be solved for the general case in the near future. However, for a given network, usually an operator may be able to specify certain corrective actions to be taken automatically when certain errors are detected. AIX NetView does not attempt to solve the general problem, but it does support automation by allowing an operator to define actions (programs or UNIX shell scripts) that are to be executed when certain events occur.

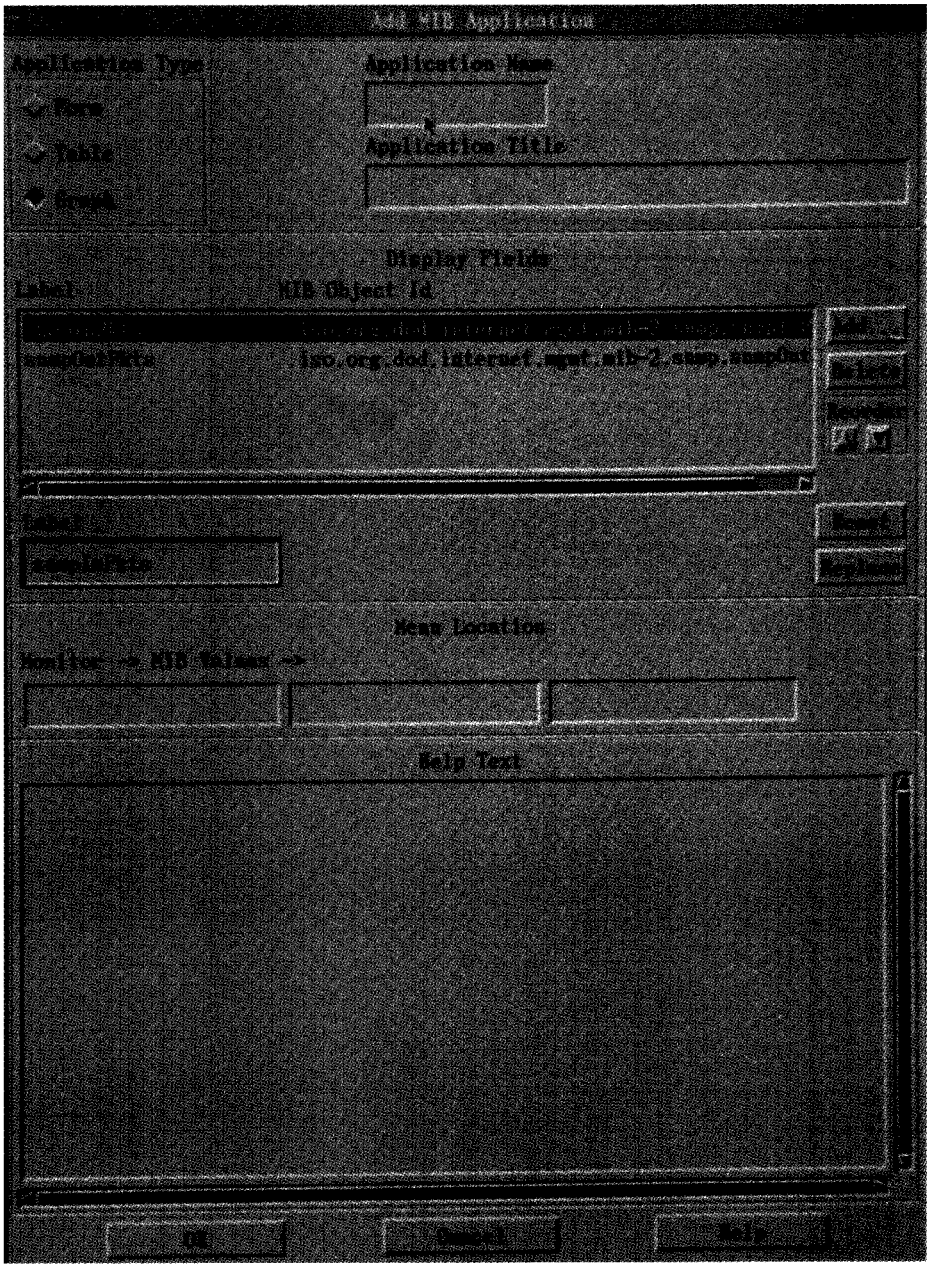
Performance. Fault management applications provide notification of errors that have already occurred. In many cases, it may be possible to act proactively to prevent problems from occurring, especially when these problems are caused by resource exhaustion (for example, exhaustion of disk space, network bandwidth, or processor time). AIX NetView provides two performance applications that assist in monitoring performance aspects of a network. One application, the MIB Application Builder, actually generates applications that allow an operator to monitor a device in real time. Another, the MIB Data Collector, performs the dual functions of collecting data from selected nodes at regular intervals for trend analysis and of generating an error notification when operator-specified thresholds are exceeded on monitored devices.

MIB Application Builder. The MIB Application Builder, shown in Figure 4, is used to generate an application, accessible from AIX NetView's menus, which will query and display MIB variables from selected network nodes. The MIB Application Builder allows programming-free creation of applications to monitor MIB variables.

Any MIB variable defined for a device can be displayed in either a textual format (simple form or table) or, when appropriate, as a graph. The operator selects both the variables to be displayed and the type of display, using a point-and-click interface. The application builder also allows the input of help screens to support the form, table, or graph presented.

AIX NetView/6000 is shipped with some default applications created using the application builder for IBM and other equipment. Vendors may develop applications using this tool and ship them for use with their products, relieving their customers of the need to generate applications.

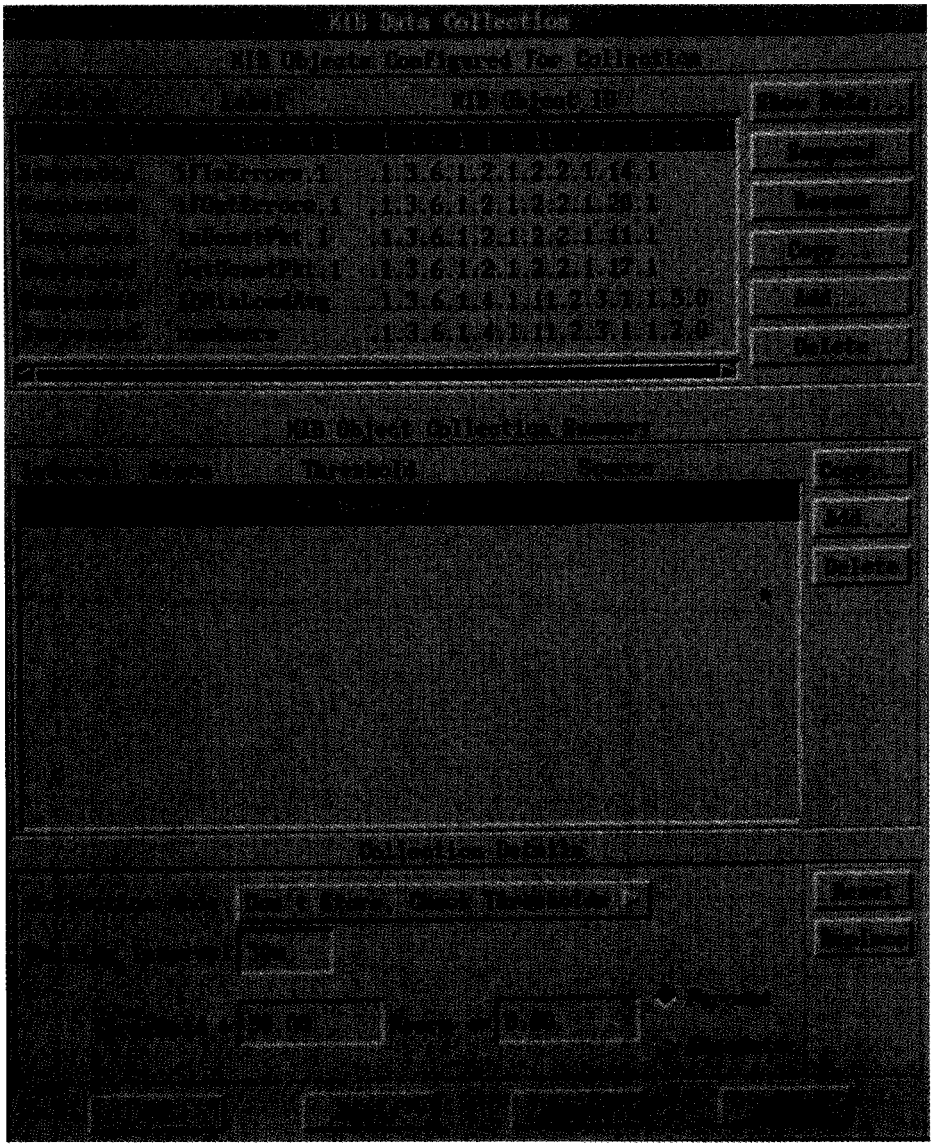
Figure 4 The AIX NetView/6000 MIB Application Builder



MIB Data Collector. The MIB Data Collector periodically polls network nodes for the values of MIB variables. The specific MIB variables that are requested and the polling interval are defined by the operator using the dialog shown in Figure 5. An operator can configure the MIB Data Collector to poll for any MIB variables at defined intervals.

The data collected in this manner can be stored in a file for later analysis; for example, the file can

Figure 5 The AIX NetView/6000 MIB Data Collector configuration



be dumped in a format that can be imported into Lotus 1-2-3**.

The operator can also specify a threshold value for each variable monitored; when this threshold is exceeded, an event is generated that notifies the operator of an impending problem. A re-arm value is also provided in order to provide hysteresis, so that events are not generated continuously after a threshold is exceeded, but only the first time a threshold is exceeded and thereafter only when the monitored variable has gone back below the re-arm value and then exceeded the threshold again.

The utility of the application builder and data collector in network management is, of course, directly proportional to the level of SNMP support in

managed devices. Devices that have more data in their MIB and allow setting of more operating parameters in their MIB will be easier to manage.

NetView connectivity

In today's data processing environment, those responsible for managing networks are confronted with networks that use a variety of telecommunication protocols and corresponding network management facilities. In such heterogeneous networks, cooperation between network management facilities is a requirement. Without this cooperation, users may not continue to be able to benefit from existing installations while expanding the physical network, unless they are willing to constrain their choice of networking protocols.

AIX NetView/6000 offers an answer for the user who uses both SNA and TCP/IP protocols. In this sort of heterogeneous environment, the two network management protocols that must cooperate are the SNA Management Services (MS) Architecture[7] managing the SNA network and SNMP managing the TCP/IP network.

The bridge between these two networking protocols is the AIX Service Point,[8] which possesses the functional base for formatting information as SNA Management Services major vectors to be sent to System/370, System/390 NetView (heretofore referred to simply as NetView), as well as that for processing MS major vectors received from NetView.

The relationship of the AIX Service Point, NetView, and AIX NetView/6000 is shown in Figure 6. As shown in this figure, the AIX node containing the AIX Service Point supports both SNA and TCP/IP networking protocols. It, therefore, can be managed as an SNA node by NetView or as an SNMP node by AIX NetView or any other SNMP manager; it can communicate with both NetView and AIX NetView using the appropriate protocols. The AIX Service Point provides a bridge between SNA- and SNMP-managed networks.

AIX NetView has the ability to convert the SNMP notification message, the trap, to the SNA notification message, the MS major vector (typically an alert major vector), and send it to NetView; this allows NetView to monitor events in the TCP/IP network. This capability of monitoring events in the SNMP environment is complemented by the ability to receive Execute major vectors (X'8061', usually referred to as RUNCMDs) from NetView, execute the desired command on a specified TCP/IP node, and send the results of the operation in a Reply to Execute major vectors (X'0061', RUNCMD reply) to NetView. These abilities allow NetView to play an active role in the SNMP environment.

Trap-to-alert conversion. The trap-to-alert conversion facility of AIX NetView is implemented for users who want to manage heterogeneous networks from a central site. By providing the facility to convert SNMP traps into alerts and forward these alerts to NetView, AIX NetView allows the NetView operator to ascertain the status of the TCP/IP network. Coupling the trap-to-alert conversion together with the ability for the NetView operator to send RUNCMDs to AIX NetView allows the NetView operator to manage TCP/IP networks effectively.

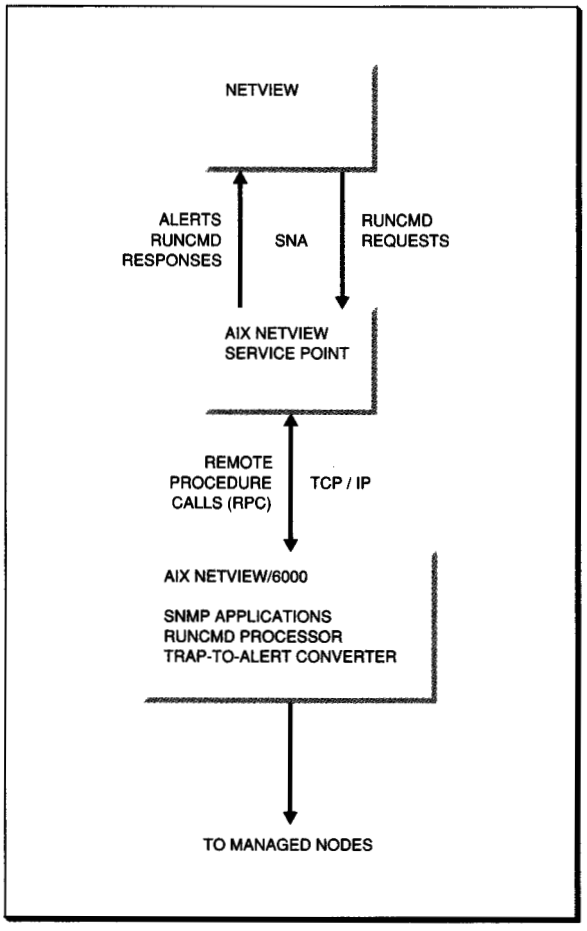
The major goal of the trap-to-alert conversion design was to provide as much information about the trap as possible in the alert. To meet this goal, the following information is essential:

- SNMP object ID (sysObjectID)
- IP address of the origin of the trap
- Type of trap
- For link up and down traps, the affected interface number (ifIndex)
- For Exterior Gateway Protocol (EGP) neighbor loss traps, the IP address of the neighbor
- For enterprise-specific traps, specific trap field and variable bindings
- Time and date of the receipt of the trap
- If agent supports MIB II, the following information is also needed:
  - — System Contact (sysContact)
  - — System Name (sysName)
  - — System Location (sysLocation)

All of this information is included in the Detailed Data Alert MS Subvector of the alert being generated.

Special consideration must be made for preserving the values of the variable-bindings (MIB variables and values) that may be carried in the enterprise-specific trap. These values can be longer than the 44-byte limit in the data field of the Detailed Data Network Alert Common Subfield, and so cannot be sent in the generated alert. These

Figure 6 The AIX Service Point



variables are in the MIB of the device, but they may change dynamically. While the values of these variables may be retrieved from the agent using an SNMP GET request after receipt of the trap, the retrieved values may not correctly reflect the values present at the time the trap occurred; only the variable bindings that were carried in the trap itself are valid.

To solve this problem, every trap whose full contents could not be carried in an alert (due to size or other limitations) is logged with a unique number associated with it. This number is sent in a Detailed Data Network Alert Common Subfield. A RUNCMD can then be issued from NetView to retrieve the entire trap by specifying the unique number associated with the trap. This value is read and returned in the RUNCMD response.

Filtering. In order to reduce the number of alerts sent to NetView, AIX NetView provides filtering on a trap-type basis and on a device basis. An AIX NetView operator can define the specific types of traps that are to be converted to alerts and forwarded to NetView. Only traps that meet this criteria, and that come from nodes on which trap-to-alert conversion has been enabled, will result in an alert being sent to NetView.

Executing commands from NetView. The trap-to-alert conversion facility of AIX NetView allows monitoring of the status of TCP/IP networks from NetView. Complementing this is a facility, the RUNCMD processor, which allows the NetView operator to perform actions on devices in the TCP/IP network in order to correct the cause of a problem.

The sequence of events illustrating how NetView can interact with an SNMP-managed device is shown in Figure 7. A NetView operator or program can issue an Execute major vector for the AIX Service Point by entering a string such as

```
RUNCMD SP=NTFFPU04,APPL=T12EXEC,
  RSH JIM PWD
```

The keyword RUNCMD instructs NetView to construct an Execute major vector in which the Self-Defining Text subvector contains, among other things, a command string destined for AIX NetView/6000. The RUNCMD provides the SNA name (NTFFPU04) for the host on which the AIX Service Point is running as well as the name of a RUNCMD processor (T12EXEC), which will actually execute the command sent in the Execute major vector. The command shown in Figure 7 is, after being translated to lowercase letters,

```
rsh jim pwd
```

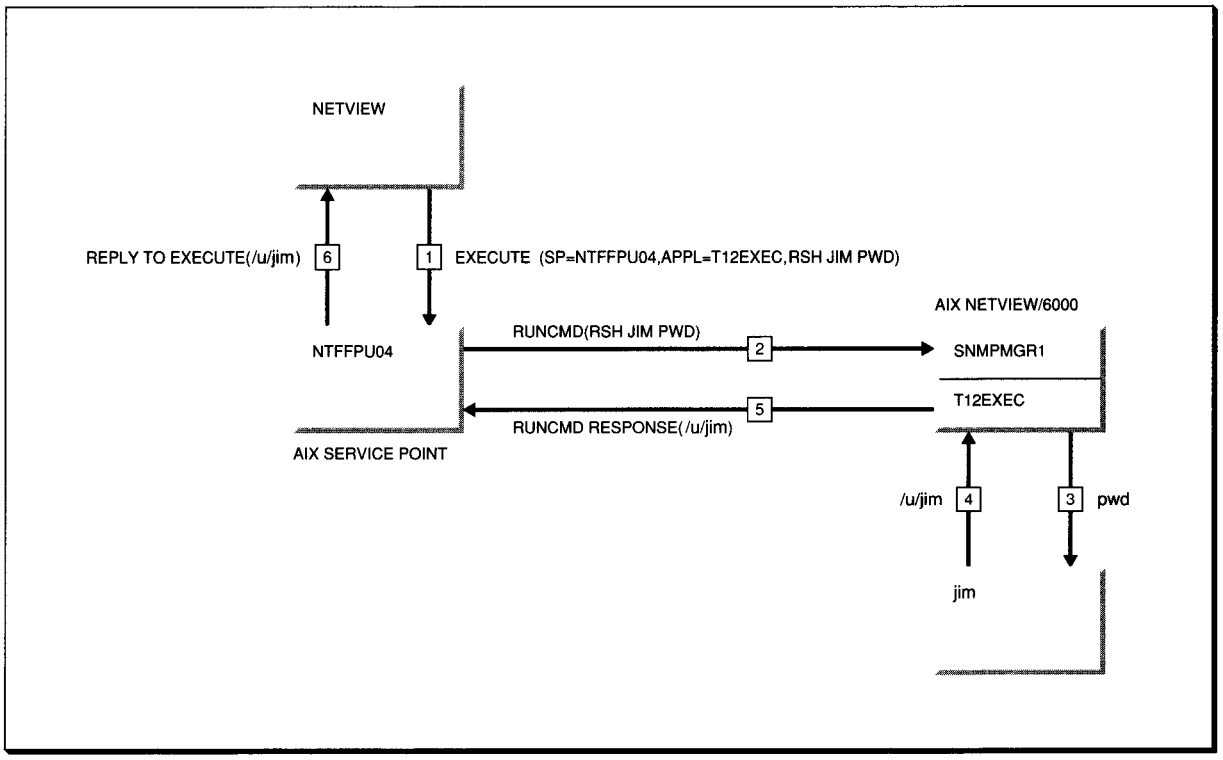
which requests the printing of the current directory on the remote host "jim." The results of this command

```
/u/jim
```

are sent back to the RUNCMD processor, which invokes AIX Service Point functions to transport the result back to NetView as a Reply to Execute major vector.

RUNCMD processor. The function of the RUNCMD processor shown in Figure 7 is to receive the message, the RUNCMD, that the AIX Service

Figure 7 NetView interactions with SNMP-managed devices



Point has extracted from the Execute major vector. The message is routed to the RUNCMD processor named in the Execute major vector. Each RUNCMD processor registers itself with the AIX Service Point as part of its initialization. After receipt of a RUNCMD, the RUNCMD processor executes the command in its native environment. Commands sent in a RUNCMD may be any that can be executed on an AIX host. The only restrictions on the commands that can be successfully executed are those associated with the AIX user with whose account the RUNCMD processor is associated. After execution of the command, the application again invokes AIX Service Point facilities to format and send the Reply to Execute major vector to NetView.

The RUNCMD processor resides on the same host in which AIX NetView is executing. The routing of RUNCMDs is based on both the SNA name of the host on which the AIX NetView Service Point is executing, as well as the name of the RUNCMD processor. Since several different RUNCMD processors, each of which resides on a different host, may use the same AIX NetView Service Point, the name of the RUNCMD processor is critical in routing RUNCMDs. To implement this routing, AIX NetView will automatically generate RUNCMD processor names that are guaranteed to be unique within the scope of any given AIX NetView Service Point. Alternately, users can configure RUNCMD application names to suit their purposes. In this case, the burden of validating the uniqueness of the names of the applications using a given AIX NetView Service point is left to the user.

The correct routing of RUNCMDs is a requirement because the RUNCMD processor makes use of AIX facilities to allow RUNCMD processor to execute programs on managed AIX hosts. To execute programs successfully on remote nodes, some predefinition is required. If the RUNCMDs are not sent to the instance of RUNCMD processor configured to operate with the remote device, the command will fail.

Some danger is inherent in permitting a NetView operator to execute any AIX command on AIX

NetView or a device managed by AIX NetView. The AIX environment offers selected users virtually unrestricted powers. To give access to these powers to a NetView operator who may not be familiar with AIX facilities is unwise. For example, the "ping" command is available to any AIX user. By issuing the command

```
ping 9.67.5.120
```

a user can cause TCP/IP packets to be sent to the device at address 9.67.5.120 continuously. Since the RUNCMD processor waits until the completion of the command, it will not send a response to NetView, since the command never completes. To get a response to this command, therefore, the NetView operator must use the RUNCMD to query the AIX host for the process executing the command and, upon receiving a response to this RUNCMD, issue another to terminate the identified process. To avoid this sort of potential problem, AIX NetView makes use of AIX facilities to offer system administrators the option of restricting the commands that can be executed by the user associated with the AIX NetView Service Point application. Using this option, the AIX command set available to a NetView operator can be configured to be appropriate to the level of AIX expertise of that operator.

Logging. AIX NetView records all interactions with NetView in two files. The first is a log of interactions between NetView and the RUNCMD processor; it is provided as an aid to diagnosing configuration problems between the two. The second log records the SNMP traps converted into SNA alerts when the trap information cannot be put in the corresponding alert. Each record of this log contains the original trap as well as an identifier sent in the trap-prompted alert. This identifier is used by NetView when a RUNCMD is used to retrieve the entire trap.

AIX NetView/6000 internal structure

AIX NetView/6000, like most complex UNIX applications, is composed of a number of interacting processes. These processes are shown in Figure 8.

Major components. The major components of AIX NetView are *trapd*, *netmon*, and *xnm*.

trapd. The *trapd* background process (daemon) is the process that receives traps from the managed nodes as well as internally generated events from *netmon*, *xnm*, and other components. It forwards these events to all other AIX NetView processes and logs them in a disk file.

netmon. The *netmon* daemon is the process that performs most of the configuration and fault management functions. It performs the discovery process in order to detect new nodes in the network. It also periodically polls all managed nodes in order to ascertain their status, and, at less frequent intervals, performs a more comprehensive poll on nodes with SNMP agents in order to maintain more current information regarding certain MIB variables (such as the system contact and location). Whenever the status or configuration of a node changes, it sends a notification of the change to *trapd*, which forwards it to all interested parties.

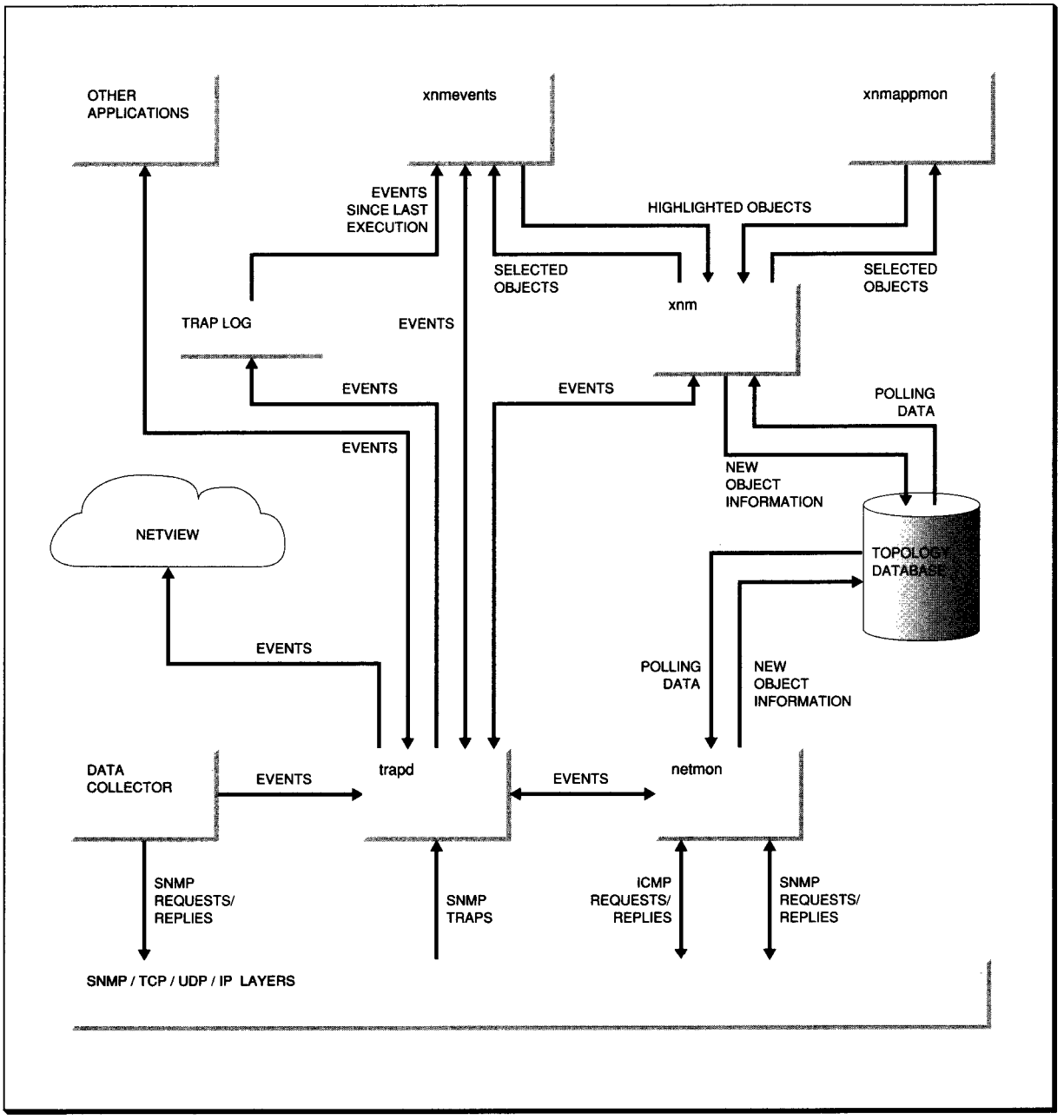
xnm. The component that provides the graphical end-user interface is *xnm*. Upon invocation, it reads the topology database to get the information about managed elements that is needed to generate the topology display. While it is operational, it responds to events generated by *netmon* and other processes by changing the color of the affected nodes appropriately. It also allows a user to edit the topology database in order to manually change the layout of the network (for example, in order to reflect physical placement). Changes made to the network views by the user are stored in the topological database and are sent to *trapd* as internal events.

In addition to drawing the network views, *xnm* also provides an integration point for other applications. It reads an application registration file that defines other applications that may be invoked (via menu selections) from the main window.

Other components. In addition to the components described above, there are a number of other applications which comprise AIX NetView.

xnmevents. The list of events shown in the Control Desk, although apparently part of *xnm*, is actually done by the *xnmevents* process. This process shows all pending events (pending events are events that have occurred on the network, but have not yet been acknowledged by the operator). The events are divided into various categories (i.e., threshold events, error events, status events, etc.) that can be viewed individually or

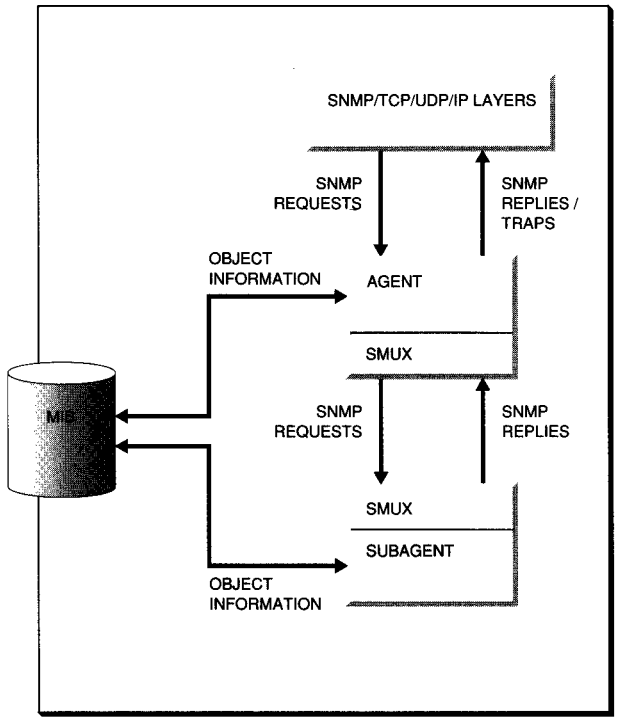
Figure 8 Internal structure of AIX NetView/6000



collectively. At startup, *xnmevents* reads unacknowledged events from disk and displays them. It then updates this list with events from *trapd* when they occur.

xnmappmon. Many of the diagnostic tests described in the previous section on problem determination are initiated by the *xnmappmon* process. When an operator executes certain tests

Figure 9 SNMP agent-subagent relationship



against one or more devices, *xnmappmon* is invoked by *xnm*; it creates the windows used to display the results of the commands that actually perform the diagnostics.

Data collector. The data collector daemon, *snmpCollect*, is the process that periodically polls node for MIB values. It stores the collected information in a file and also sends an event to *trapd* when a threshold is exceeded, as previously described in the previous section on the MIB Data Collector.

tralertd. The *tralertd* daemon is the process that converts traps to alerts. It queries the topology database to determine if a received event is to be sent to NetView as an alert. If so, it extracts information from the event and from the topological database to use in alert generation; these data are then transferred using remote procedure calls (RPCs) to the AIX Service Point, which does the actual sending of an alert to NetView.

AIX NetView subagent. All of the AIX NetView components just described run in the management station; they are part of the network manager itself. AIX NetView/6000 also provides an SNMP subagent that is designed to run on AIX 3.2 nodes in order to increase the level of management available. The AIX 3.2 operating system collects a great deal of error information. The AIX 3.2 agent, however, like most other agents, does not use enterprise-specific traps to notify a manager of these errors. The AIX NetView subagent, when installed on an AIX 3.2 machine, will notify AIX NetView of these system errors using enterprise-specific traps, thus extending the usefulness of the management system.

A diagram of the AIX 3.2 SNMP agent and AIX NetView subagent is shown in Figure 9.

The subagent communicates with the agent via the SNMP Multiplexing (SMUX) protocol. The subagent becomes an SMUX peer by initiating an SMUX association and registering itself with the agent. The subagent can then send traps through the agent when an error is written to the system error log.

Conclusion

AIX NetView/6000 is by no means perfect. One deficiency is the lack of a relational database. Although it provides facilities for dumping data into readable formats, it would be more convenient if all data, both topology data and collected performance data, could be stored in a relation or object-oriented database in order to better enable customers to use these data for their specific management tasks.

A more serious deficiency is the fact that AIX NetView/6000 is solely a TCP/IP manager. It is unable to assist in the diagnosis of events at the physical transport level (as is done by network sniffers) or to display or manage nodes using protocols other than TCP/IP. One thing that all users have demanded is a network management system that can do total network management of both voice and data, regardless of protocol. Only time can cure these problems.

However, all in all, AIX NetView/6000 provides a powerful and flexible network management system. It can operate as a stand-alone system in a distributed management environment, or it can work with NetView to provide the IBM user with

a centralized point of control for heterogeneous networks.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Digital Equipment Corp., UNIX Systems Laboratories, Inc., Hewlett-Packard Co., Open Software Foundation, Inc., or Lotus Development Corporation.

Cited references and note

1. M. Rose and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets," RFC 1155, Internet Activities Board, Marina Del Ray, CA (May, 1990).
2. M. Rose, *The Simple Book*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1991).
3. J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A Simple Network Management Protocol," RFC 1157, Internet Activities Board, Marina Del Ray, CA (May, 1990).
4. K. McCloghrie and M. Rose, "Management Information Base for Network Management of TCP/IP-based Internets," RFC 1156, Internet Activities Board, Marina Del Ray, CA (May, 1990).
5. M. Rose, "Management Information Base for Network Management of TCP/IP-based Internets: MIB-II," RFC 1158, Internet Activities Board, Marina Del Ray, CA (May, 1990).
6. U.S. Department of Defense virtual terminal protocol, based on TCP/IP.
7. *IBM Systems Network Architecture Formats*, GA27-3136, IBM Corporation (1991); available through IBM branch offices.
8. *AIX NetView Service Point Installation, Operation, and Programming Guide*, SC31-6120-0, IBM Corporation (1991); available through IBM branch offices.

Accepted for publication January 15, 1992.

James H. Chou *IBM Corporation, 3039 Cornwallis Road, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Chou is currently a staff programmer in the AIX network management group. He joined IBM in 1984 as an engineer with the token ring test group. He moved to the NetView/PC development group in 1985, where he worked on the alert manager and on the porting of the product to OS/2. In 1988 he entered the resident study program at the University of North Carolina at Chapel Hill, after which he returned to the NetView system design group in the fall of 1990, where he did some of the initial design for AIX NetView/6000. In 1991 he followed the design to the AIX network management development group. Mr. Chou received a B.S. in biology from MIT in 1983, a B.S. in electrical engineering from MIT in 1984, and an M.S. in computer science from the University of North Carolina in 1989.

C. Richard Buckman *IBM Corporation, 3039 Cornwallis Road, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Buckman joined IBM in 1991 as a senior associate programmer with the AIX network management group. Before joining IBM, Mr. Buckman worked as a consultant in the design and implementation of axiological assessment tools. Mr. Buckman received his B.S. in computer science from Western Kentucky University in 1986 and his M.S. in computer science from Vanderbilt University in 1988.

Tom Hemp *IBM Corporation, 3039 Cornwallis Road, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Hemp is currently an advisory programmer in the AIX network management group. He joined IBM in 1977 as an engineer at IBM San Jose. There he contributed to the development of the Olympus Test System. He transferred to Research Triangle Park, North Carolina, in 1980 working in the line-switching area and the 3174 Controller group. In 1990, he joined the AIX network management group. Mr. Hemp received a B.S. in electrical engineering from the University of Virginia in 1975 and an M.S. in electrical engineering from the University of Colorado in 1977.

Alec Himwich *IBM Corporation, 3039 Cornwallis Road, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Himwich is currently an advisory programmer in the AIX network management group. He joined IBM in 1981 as a programmer in Research Triangle Park. There, he initially worked in MVS systems support modeling the performance of large systems. In 1985, he moved to SNA architecture where he worked on the design and metaimplementation of the APPN node. Mr. Himwich received a B.A. in liberal arts from St. John's College in Annapolis, Maryland, and an M.S. in mathematics from the University of Illinois. Before joining IBM, he worked as a programmer of computer-based courseware and simulations on the PLATO system at the University of Illinois and at the University of Connecticut Health Center.

Fred Niemi *IBM Corporation, 3039 Cornwallis Road, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Niemi is currently an advisory engineer in the AIX network management group. He joined IBM in 1978 as an engineer in the IBM Kingston Development Laboratory. There he contributed to the development of the IBM Universal Controller microprocessors used in the 8100 distributed processing systems. Mr. Niemi transferred to Research Triangle Park, North Carolina, in 1982. He joined the network management area in 1984 working on NetView/PC and received an Outstanding Technical Achievement Award for the design and implementation of the NetView/PC Communications Manager. In 1991, Mr. Niemi joined the AIX network management group. Mr. Niemi received a B.S. in electrical and computer engineering from the University of Michigan in 1978.

Reprint Order No. G321-5473.