

It is well known that a main telecommunications network may be used to provide communication links between local networks. The main telecommunications network may be
10 a private or public network belonging to a public telecommunications company and the local networks may belong to various other companies. Thus, where a company has local networks at two or more locations, it may use the main network to connect them. It is known to provide network
15 managers for both main networks and for local networks. A network manager receives data from the individual elements of the network which it manages and can send-instructions to these elements. Because these two types of networks have developed separately and have different requirements, the
20 information protocol used by the network manager of a local network is usually different from the information protocol used by the network manager of the main network to which the local network is attached. For example, the SNMP protocol is commonly used in local networks while the CMIP protocol is
25 commonly used in main networks. Although the human operator of a network manager of a local network may be mainly interested in the elements of his network, it is also helpful for him to receive information about some of the elements of the main network which is used to connect his local network
30 to other local networks. Because differing protocols are used in the two types of network, it is difficult to transfer data easily from a network manager of a main network to a network manager of a local network.

According to this invention, there is provided a data
35 storage device for storing data on individual objects of, or related to, a first telecommunications network, said data storage device comprising: means for receiving data according

This Sheet Has Been Replaced



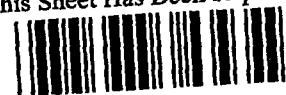
objects of, or related to, the first telecommunications network following its conversion by the data converting means; and means for supplying data from the data storing means on individual objects of, or related to, the first telecommunications network according to the second information protocol to a network manager of a second telecommunications network.

Because the data storage device of this invention converts data from a form used in the first information protocol into a form used in the second information protocol before storing it, it facilitates the supply of data according to the second information protocol to a network manager. Thus, the data storage device of this invention may be used for transferring data from a network manager for a main network to a network manager for a local network. It may also be used, if desired, for transferring data from a network manager for a local network to a network manager for a main network.

According to a second aspect of this invention, there is provided a network management system comprising a network manager for a telecommunications network and a data storage device according to the first aspect of this invention. The network management system may include a network manager for a second telecommunications network.

According to a third aspect of this invention, there is provided a method of operating a data storage device for storing data on individual objects of, or related to, a first telecommunications network, said method comprising the steps of: receiving data according to a first information protocol on individual objects of, or related to, the first telecommunications network from a network manager for that

This Sheet Has Been Replaced



second telecommunications network.

This invention will now be described in more detail, by way of example, with reference to the drawings in which:

10 Figure 1 is a block diagram showing three local networks connected to a main network

Figure 2 is a block diagram showing the relationship between a network manager for a main network, a network manager for a local network and a data storage device
15 embodying this invention;

Figure 3 is a block diagram of the software components of the data storage device shown in Figure 2;

Figure 4 is a diagram illustrating the operating states of the overall control component of the software shown
20 in Figure 3;

Figure 5 is a flow chart of the CMIP application component of the software shown in Figure 3;

Figure 6 is a flow chart of the CMIP/SNMP mapper component of the software shown in Figure 3;

25 Figure 7 shows the CMIP attributes and the corresponding SNMP attributes of the object class for a private circuit used in the software shown in Figure 3;

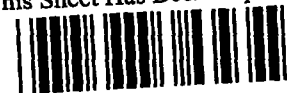
Figure 8 shows the CMIP attributes and the corresponding SNMP attributes for the object class for a
30 router port used in the software shown in Figure 3;

Figure 9 shows the CMIP attributes and the corresponding SNMP attributes for the object class for an access point used in the software of Figure 3; and

Figure 10 is a flow chart of the SNMP application
35 component of the software shown in Figure 3.

Referring now to Figure 1, there is shown a main network 10 and three local networks 12, 13, 14 which are

This Sheet Has Been Replaced



example, the local networks 12, 13, 14 belong to the same organisation which is a completely separate organisation from the owner of the main network 10. The main network 10 has a set of interconnected switching nodes 15 to 20. The local network 12 is connected through a router 21 to the switching node 15, the local network 13 is connected through a router 24 to the switching node 17, and the local network 14 is connected to the switching node 19 through an access point 22 and a gateway 23. The router 21, access point 22 and gateway 23 are associated with and controlled by the main network 10 and so may be regarded as part of the main network 10. The router 24 is associated with and controlled by the local network 13. The dashed line 25 marks the boundary between elements controlled by the local networks 12, 13, 14 and elements controlled by the main network 10.

In the present example, the local networks 12, 13, 14, the router 21, access point 22, gateway 23 and the main network 10 shown in Figure 1 form part of a frame relay system. Although the present invention will be described with reference to a frame relay system, it is to be appreciated that it may be used in other type of telecommunications systems.

Referring now to Figure 2, the main network 10 has a network manager 30. The network manager 30 receives data from the elements of the main network 10 and also the router 21, access point 22 and gateway 23 and sends instructions to them. The network manager 30 is arranged to transmit and receive data according to an information protocol known as the Common Management information protocol (CMIP) and which provides a set of services known as the Common Management Information Services (CMIS). When an element transmits data in another protocol, the data is concerted into CMIP. The

This Sheet Has Been Replaced



construction of network managers is well known, the network managers 30 and 32 will not be described in further detail. As is also well known, a network manager is usually implemented as a general purpose computer provided with
10 appropriate software.

Although the human operator of the network manager 32 for the local network 12 is mainly interested in the elements of that network, it is also valuable for him to receive data on the state of the elements of the main network 10 as well
15 as the router 21, access point 22 and gateway 23 which are used to provide virtual circuits between the local networks 12, 13 and 14. For example, if the port of the router 21 is disabled, it is useful for the human operator to know this so that he can take appropriate action. Because the network
20 managers 30 and 32 use different information protocols to communicate with their respective network elements, data cannot be supplied directly from the network manager 30 to the network manager 32. In order to enable the network manager 32 to receive data on elements in the main network
25 10, the network manager 30 for the main network 10 supplies data on these elements to a data storage device 34 which in turn supplies the data to the network manager 32 for the local network 12. The data storage device 34 will be described in more detail below. As will be explained, the
30 network manager 30 supplies data to the data storage device 34 using CMIP and the data storage device 34 supplies data to the network manager 32 using SNMP.

The data storage device 34 may be implemented as a general purpose computer provided with appropriate software.
35 In the preferred embodiment, the data storage device 34 uses the same computer as the network manager 30 with the result that the network manager 30 and the data storage 34 are

This Sheet Has Been Replaced


known as an object-oriented environment. In an object-oriented environment, computer software is used to model real world physical objects as well as other real world entities.

10 The physical objects and entities are simply known as objects. Where, as in the present case, the computer software is managing real world objects, it models only those attributes of the real world objects which are necessary for management. When objects are modelled in this way, they are

15 known as managed objects. A category of objects of the same or similar type is known as an object type. A particular real world object is said to be an instance of an object type. Thus, for example, router ports might be an example of an object type and the individual ports of router 21 are

20 instances of that object type.

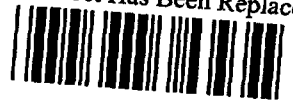
Individual software modules which model real world objects are also known as objects. The software implementation of object type is known as object class. Thus, an individual software module which models a particular

25 real world object is known as an instance of the object class to which that object belongs. Each software object is identified by the name of the object class to which it belongs and by an identifier which is specific to the software object itself. Each object class has a set of pre-

30 defined attributes. For example, in the case of an object class for router ports, the attributes may include the operating state of the port. For an instance of a particular object class, each attribute would have a particular value and the values are liable to change.

35 The network manager 30 uses a set of object classes for modelling the various types of object found in the main network 10. These object classes include three classes

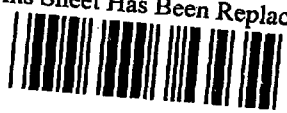
This Sheet Has Been Replaced



and the pre-defined values are appropriate for the network manager 30 and CMIP. Similarly, the object classes used by the network manager 32 include three object classes corresponding to the object classes used by the network manager 30 for private virtual circuits, router ports and access points. The names of the attributes as well as their pre-defined values are appropriate for SNMP. For each of these three object classes used in the network manager 30 and the corresponding object class used in the network manager 32, the names of the attributes and values differ between the two corresponding classes. As will be described in more detail below, the data storage device 34 can convert data on the attributes of a particular object class used in the network manager 30 into an appropriate form for the corresponding object class used in the network manager 32.

In SNMP, a network manager issues commands to obtain information from network elements. These commands include Get and Get-Next. A Get command is a request for the value of a particular attribute of a particular object. A Get-Next command is a request for the value of the same attribute of the next object.

The services provided by CMIS include m-Get and m-EventReport. The m-Get service may be a request for the value of a specified attribute of a specified object or for all the attributes of all the objects belonging to a particular object class. An m-EventReport is an offer, for example by a network element to a network manager, to supply data. The data may be the values of a complete set of attributes of a particular object or a change in the value of a particular attribute of a particular object. An m-EventReport is an unsolicited event report.

This Sheet Has Been Replaced


communications stack 50, a UDP/IP communications stack 52, support files (CFG) 54, an overall control component (O-CTRL) 56 and a trap functions component (USER-EXITS) 58. Figure 3 also shows a multiplexer 60 together with a TCP/IP communications stack 62 and a UDP/IP communications stack 64.

The CMIP application component 42 is responsible for sending requests to, and receiving responses and unsolicited event reports from, the network manager 30. The CMIP/SNMP mapper component 44 is responsible for converting values of the attributes of each object from the form used in CMIP to a form used in SNMP. When data on the values of the attributes of the individual objects of a telecommunications network is stored in a database, the database is known as a management information base (MIB). MIB 46 stores the values of the attributes of the objects of the telecommunications network 10 after conversion by the mapper component 44. Thus, data is stored in MIB 46 in a form suitable for transmission using SNMP. The SNMP application component 48 is responsible for receiving requests from, and sending responses and unsolicited event report to, the network manager 32. In SNMP unsolicited event reports are called traps. The trap functions component 58 is responsible for selecting which of the unsolicited event reports from the network manager 30 should be passed on as traps to the network manager 32. The various components of the data storage device 34 and also the multiplexer 60 will now be described in more detail.

The CMIP stack 40 is responsible for converting CMIS requests from the CMIP application component 42 into a form for transmission to the network manager 30 and for converting responses from the network manager 30 into a form suitable for the CMIP application component 42. The CMIP stack 40

This Sheet Has Been Replaced



the CMIS applications component 42. CMIS and CMIP are defined, respectively, in ISO/IEC Standards 9595 and 9596. A suitable software package for the CMIP handler is available from British Telecommunications plc. A suitable software
10 package for a communication stack is available from Retix Corporation of Santa Monica, California, USA.

The TCP/IP stacks 50, 62 enable data to be transmitted using the TCP/IP protocols. Likewise, the UDP/IP stacks 52 and 64 enable data to be transmitted using the UDP/IP
15 protocols. Suitable software packages for the stacks 50, 52, 62 and 64 are commercially available. For example, the well known Sun operating system includes both TCP/IP and UDP/IP stacks.

Although Figure 2 shows only a single network manager
20 for a local network receiving data from the network manager 30, the present invention may be used to provide data from a network manager for a main network to more than one network manager for a local network. A single data storage device can provide data on the main network to all the local network
25 managers belonging to a particular customer as these local network managers will be interested in the same elements of the main network. However, partly because different customers are interested in different sets of elements of the main network and partly because it may be necessary to
30 restrict the supply of data on network elements for security reasons, it is necessary to provide an individual data storage device for each customer. The data storage devices may be located together or separately. Requests from the various network managers for the local networks are received
35 on a common communication link by a multiplexer 60. Each request includes an identifier for the network manager which is making it. The multiplexer 60 then transmits the request

This Sheet Has Been Replaced



ERS-SENT, UPLOAD-MIB, RUNNING and SHUTDOWN. The SNMP application component 48 exists in five operating states, namely, STARTUP, CONNECTING, REG-REQ-SENT, WAITING-ON-CMIP and RUNNING. The state variable for each of these components is maintained in the overall control component 56. For start-up, the state variable for both components is set to STARTUP. For the CMIP application component, the state variable is changed from one state to another for the next four states by the CMIP application component. As will be explained in more detail below, in the state UPLOAD-MIB, initial data on the elements of the main network 10 are supplied to the CMIP application component. In the state RUNNING, the CMIP application component receives unsolicited event reports from the network manager 30. For shutdown the state variable is changed to SHUTDOWN.

Following start-up the SNMP application component changes its state variable from one state to another for the remaining four operating states. As will be described in more detail below, during the states CONNECTING and REG-REQ-SENT, the SNMP application component establishes a connection with, and registers itself with, the multiplexer 60. During the state WAITING-ON-CMIP, the SNMP application component waits for the completion of supplying the initial data to the CMIP application component. During the state RUNNING, the SNMP application component services requests from the network manager 32 and sends traps to it.

The support files 54 include a list of the CMIP object classes which can be supported by the data storage device 34. In the present example, the data storage device 34 can support CMIP object classes for private virtual circuits, router ports and access points.

This Sheet Has Been Replaced



initialisation routines. Then, in a step S2, it sets the state variable to DELETE-ERS-SENT.

CMIS provides a function known as a filter. When a filter is in place in a MIB, unsolicited event reports are issued when changes occur in the values of specified attributes of objects belonging to specified object classes. In step S3, the CMIP application component sends an instruction to the network manager 30 to delete any filter which is in place with regard to the data storage device 34. The purpose of this is to make sure that any previous filter is cancelled.

Then, in a step S4, the state variable is set to CREATE-ERS-SENT.

Next, in a step S5, the CMIP application component instructs the network manager to create a new filter. This filter specifies both the object classes and the attributes of the three classes for which unsolicited event reports are required. The CMIP application component obtains the data for constructing the filter from appropriate support files 54. The state variable is then changed to UPLOAD-MIB in a step S6.

In order to prevent data storage devices from receiving data to which they are not entitled, the MIB in the main network manager 30 is partitioned and the data storage device 32 has access only to the data contained in the partition associated with it. In a step S7 the CMIP application component sends an m-get-request to the network manager 30 for each object class for which it requires data. The names of these object classes are retrieved from the support files 54. Thus, in the present example, it sends m-get requests for the private virtual circuits, router ports and access points object classes. For each object contained

This Sheet Has Been Replaced



distinguished name for the particular instance of the object class and the name and value of each attribute of the object.

The CMIP application component then passes the data to the CMIP/SNMP mapper 44 and the data for that instance is stored in MIB 46. Thus, in this manner, the data for each object class is uploaded into MIB 46.

After receiving data on each instance, the CMIP application component checks whether that instance is the last instance on which data will be sent. If it is not the last instance, the CMIP application component receives data on the next instance. If it is the last instance, it continues with a step S8 in which the state variable is set to RUNNING.

Then, in a step S9, the CMIP application component repetitively schedules the CMIP stack 40 to determine if any unsolicited event reports have been received.

The operation of the CMIP/SNMP mapper component 44 will now be described with reference to Figures 6 to 9. Figure 6 shows a flow chart for this component. Figures 7 to 9 show respectively the CMIP attribute names together with the SNMP attribute names for the object classes for circuits, router ports and access points.

Referring firstly to Figure 7, it will be seen that the attributes names comprise circuitId, aEndPointName, zEndPointName, administrativeState, operationalState, circuitBandwidth and userLabels. These attribute names refer respectively to the identity of the circuit, the beginning point of the circuit, the end point of the circuit, its administrative state, its operational state, its bandwidth and text which the operator of the network manager may add. Each attribute can have certain values. For example, for a particular circuit, circuitId is the identifier for the

This Sheet Has Been Replaced



pvcCircuitDown and pvcCircuitUp. These two traps are issued, respectively, when the attribute operationalState changes to "disabled" and "enabled".

Referring now to Figure 8, the CMIP attributes for a
10 router port comprise equipmentId, administrativeState,
operationalState, typeText and userLabels. These refer
respectively to the identity of the router port, its
administrative state, its operational state, a verbal
description of it and text which the operator of the main
15 network 30 may add. Each of these may have one of a number
of values. The corresponding SNMP attribute names are shown
beside the CMIP attribute names. For each value of a CMIP
attribute the corresponding SNMP attribute has a
corresponding value. The SNMP object class also has two
20 traps, namely, RouterPortDown and RouterPortUp. These two
traps are issued, respectively, when the attribute
operationalState changes to "disabled" and "enabled".

Referring now to Figure 9, the CMIP attributes for the
access point object class comprise functionId,
25 administrativeState, operationalState, protocolType and
userLabels. These refer respectively to the identity of the
access point, its administrative state, its operational
state, the type of protocol which it uses and text which the
operator of the network manager 30 may add. The
30 corresponding SNMP attribute names are shown beside the CMIP
attribute names. As with the circuit and router port object
classes, each CMIP attribute may take one of a number of
values and for each of these value there is a corresponding
SNMP value. The SNMP object class also has two traps,
35 namely, GNSAccessDown and GNSAccessUp. These two traps are
issued, respectively, when the operationalState changes to
"disabled" and "enabled".

This Sheet Has Been Replaced

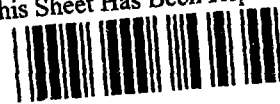

event reports during the operational state RUNNING. The mapper component converts each attribute value in turn and Figure 6 shows the procedure for converting one attribute value.

10 In SNMP, for a particular attribute of a particular instance of a particular class, the combination of the name of the class, the name of the attribute and the distinguished name or identifier for the particular instance of the class is known as the object identifier. In a step S20 for a
15 particular attribute, the mapper component converts the CMIP class name, attribute name and distinguished name into the corresponding SNMP object identifier. The data for converting the class names and attribute names is stored in the mapper component 44. The method of converting the
20 distinguished names will be described below.

In a step S21, the value of the attribute in CMIP is converted into the corresponding value in SNMP. The method for doing this is described below.

In an SNMP MIB, the values of the attributes of the
25 various instances of a particular class are stored in a table dedicated to that class. Each row of the table is associated with a particular instance of a class and each column is associated with a particular attribute of the class. The data is stored as numerals. More specifically, each class is
30 identified by a unique series of numerals and each attribute of the class is identified by a unique series of numerals. The CMIP distinguished name for an instance of the class is converted into ASCII code. Thus, an SNMP object identifier comprises a series of numerals for the class name, a series
35 of numerals for the attribute name and the ASCII code for the distinguished name.

This Sheet Has Been Replaced



and 4. where an attribute value is expressed as text, for example the value of the attribute userLabels, the text is converted into ASCII code.

10 When the local network manager 32 receives an attribute value for a particular object from the data storage device 34, the series of numerals representing the SNMP object identifier and the numeral or numerals giving the attribute value are converted by the network manager into
15 text. Thus, for the attribute operationalState, an attribute value of "1" is converted into "enabled".

 The SNMP application component 48 will now be described with reference to the flow chart shown in Figure 10. The primary purpose of this component is to service Get
20 and Get-Next requests.

 In a step S30, the SNMP application component performs initialisation routines. Then, in a step S31, it sets its state variable to CONNECTING. In a step S32, it forms a connection with the multiplexer 60. In a step S33, it sets
25 its state variable to REG-REQ-SENT. Then, in a step S34, the SNMP application component registers the data storage device 34 with the multiplexer 60 by supplying its identifier to the multiplexer 60.

 In a step S35, the SNMP application component sets its
30 state variable to WAIT-FOR-CMIP-SIDE. Then, in a step S36, it waits until the CMIP application component has received all the initial data from the network manager 30 and loaded the corresponding SNMP data into MIB 46. When all the initial data has been loaded into MIB 46 and the CMIP
35 application component has changed its state variable to RUNNING, the SNMP application component changes its state variable to RUNNING in a step S37. Then, in a step S38, the

This Sheet Has Been Replaced



contain a valid password, the program continues with a step S40 in which the request is ignored.

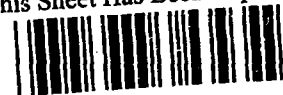
If the request contains a valid password, the SNMP application component continues with a step S41. Each
10 request contains an identifier for the local network manager making the request. In step S41, the SNMP application component checks whether it is authorised to send data to a local network manager having this identifier. If the
15 identifier is not valid, and the request cannot therefore be authorised, the request is ignored in step S40. If the identifier is valid, the SNMP application component continues with step S42.

In step S42, the SNMP application component checks if the request is valid. For example, if the request is
20 corrupted or relates to an object class on which data cannot be supplied, it is not valid. If the request is not valid, the local network manager is informed of this in step S43. If the request is valid, it is executed in a step S42.

After steps S40, S43 and S44, the SNMP application
25 component returns to step S38.

As mentioned above, the SNMP application component sends unsolicited event reports or traps for changes in the values of some attributes. For each object class, the data in the support files 54 specifies the changes in attribute
30 value which will cause traps to be issued. In the present example, for each object class, a change in the value of the attribute operationState to "enabled" or "disabled" causes a trap to be issued. When the CMIP application component 42 receives details of a change in an attribute value, it checks
35 with the support files 54 if the change in the attribute value is one for which a trap is issued. If it is one for which a trap is issued, the CMIP application component 42

This Sheet Has Been Replaced



a network manager for a local network, the present invention may also be used to enable a network manager of a main network to obtain data from a network manager of a local network.

This Sheet Has Been Replaced

A standard 1D barcode with vertical black bars of varying widths on a white background.

means for receiving data according to a first information protocol on individual objects of, or related to, the first telecommunications network from a network manager for that network;

10 means for converting the data received by the data receiving means from a form used in the first information protocol into a form used in a second information protocol;

means for storing data on individual objects of, or related to, the first telecommunications network following
15 its conversion by the data converting means; and

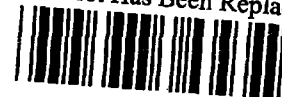
means for supplying data from the data storing means on individual objects of, or related to, the first telecommunications network according to the second information protocol to a network manager of a second
20 telecommunications network.

2. A data storage device as claimed in claim 1, including means for containing support data relating to object classes used in the first information protocol for which data can be
25 stored in the data storage device.

3. A data storage device as claimed in claim 2, in which the data receiving means includes means for requesting data from said network manager for the first telecommunications
30 network for objects belonging to one of said object classes for which data can be stored in the data storage device.

4. A data storage device as claimed in any one of the preceding claims, in which the first information protocol is
35 CMIP, the second information protocol is SNMP, and for each attribute value the data converting means is arranged to convert the CMIP class name, attribute name and distinguished

This Sheet Has Been Replaced



of the second telecommunications network when requested to do so by said network manager of the second telecommunications network.

10

6. A data storage device as claimed in any one of the preceding claims, in which the data supplying means is arranged to provide unsolicited information to said network manager of the second telecommunications network when the value changes of a pre-selected attribute of an object in, or related to, the first telecommunications network.

7. A network manager system comprising a network manager for a first telecommunications network and a data storage device as claimed in any one of the preceding claims.

8. A network management system as claimed in claim 7, in which said network manager for the first telecommunications network and the data storage device are physically located together.

9. A network management system as claimed in claim 7 or claim 8, further including a network manager for a second telecommunications network.

30

10. A network management system as claimed in claim 9, in which the data storage device and said network manager for the second telecommunications network are physically located remotely from each other, the network management system including a telecommunication link for connecting the data storage device and said network manager for the second telecommunications network together.

This Sheet Has Been Replaced


protocol on individual objects or, or related to, the first telecommunications network from a network manager for that network;

10 converting the received data from a form used in the first information protocol into a form used in a second information protocol and storing the converted data; and

15 supplying the converted data on individual objects of, or related to, the first telecommunications network according to the second information protocol to a network manager for a second telecommunications network.

12. A method of operating a data store as claimed in claim 11, including the additional step of requesting data from said network manager for the first telecommunications network for objects belonging to an object class for which data can be stored in the data storage device.

13. A method of operating a data storage device as claimed in claim 11 or claim 12, in which the first information protocol is CMIP and the second information protocol is SNMP, and, in said step of converting the received data, for each attribute value the CMIP object class name, attribute name and distinguished name are converted into the corresponding SNMP object identifier and the CMIP attribute value is converted into the corresponding SNMP attribute value.

14. A method of operating a data storage device as claimed in any one of claims 11 to 13, in which in said step of supplying converted data, data on an object is supplied to said network manager of a second telecommunications network when said network manager of the second telecommunications network requests such data.

This Sheet Has Been Replaced


an object in the first telecommunications network.

This Sheet Has Been Replaced



component 42 which receives data on network objects using the CMIP protocol from the network manager of a main network. For each attribute of each object, a CMIP/SNMP mapper component 44 converts the CMIP object class name, attribute
10 name and distinguished name of the object into the corresponding SNMP object identifier and the CMIP attribute value into the corresponding SNMP attribute value and this data is stored in MIB 46. An SNMP application component 48 receives requests for information from a network manager of
15 a local network. In order to service a valid request, the SNMP application component 48 retrieves the required data from MIB 46 and supplies this data to the network manager for the local network using the SNMP protocol. For changes in the values of certain attributes of certain objects, the SNMP
20 application component issues traps to the manager of the local network.

See Figure 3

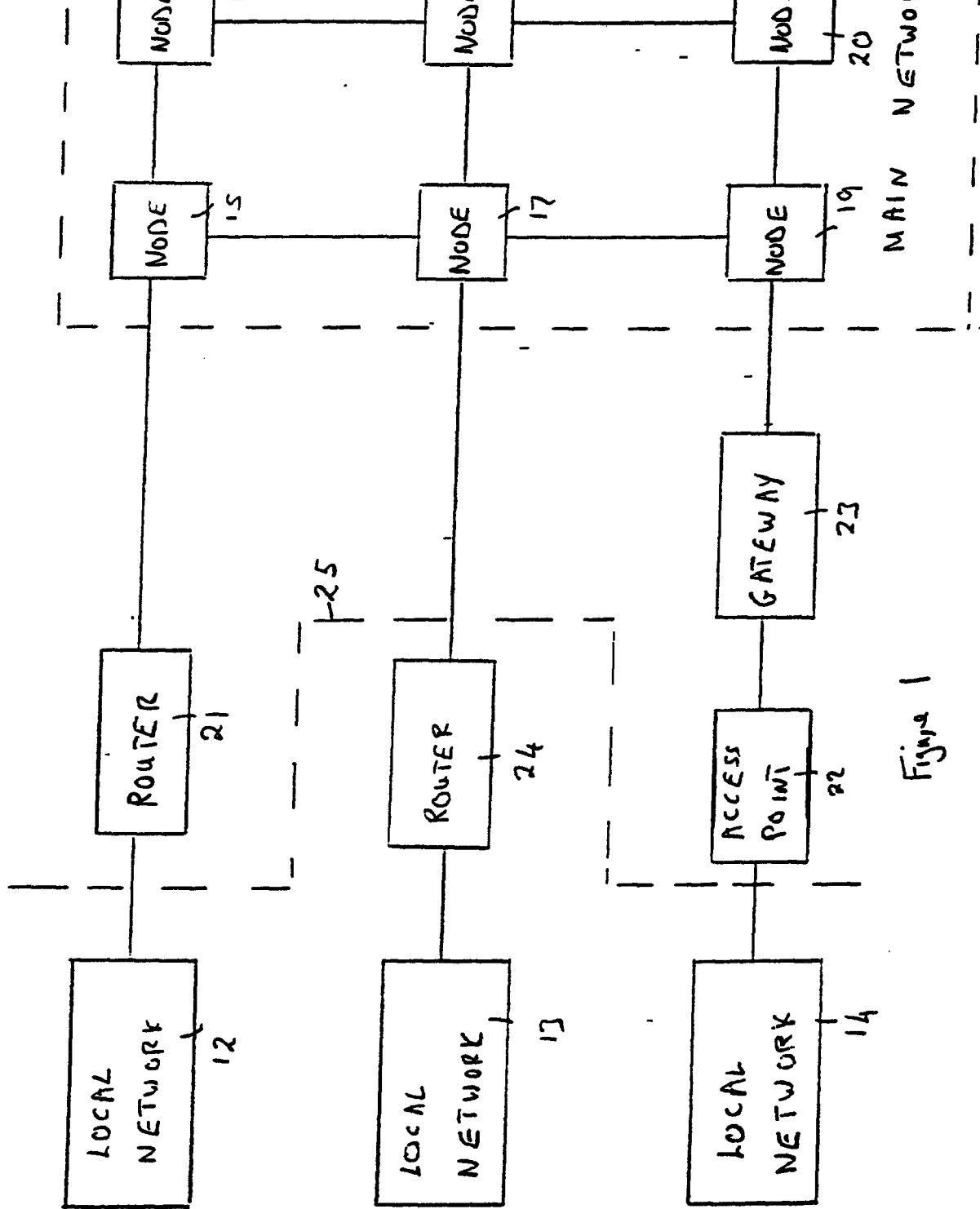
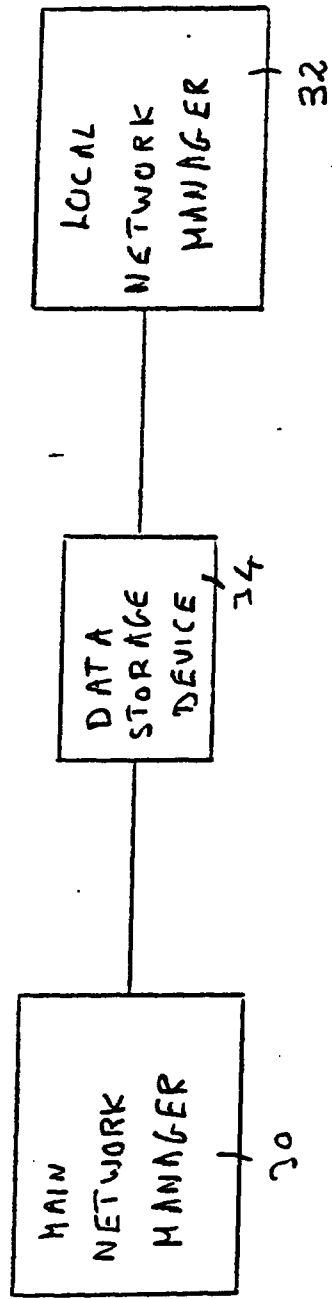


Figure 1



Figures 2

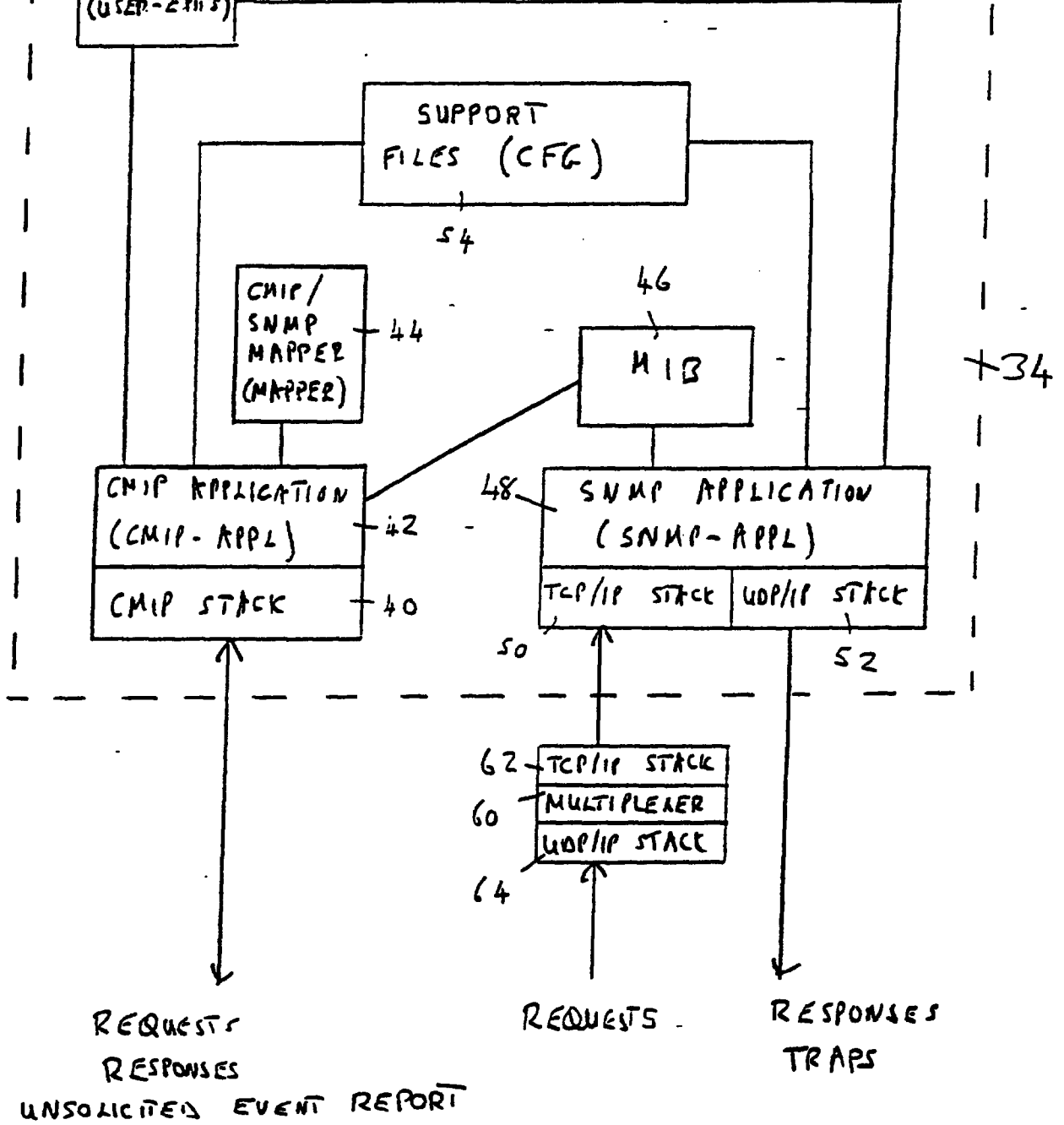


Figure 3

CONNECTING

DELETE-ERS-SENT

REG-REQ-SENT

CREATE-ERS-SENT

WAITING-ON-CHIP

UPLOAD-MIB

RUNNING

RUNNING

SHUTDOWN

SNMP APPLICATION
STATES

CHIP APPLICATION
STATES

Figure 4

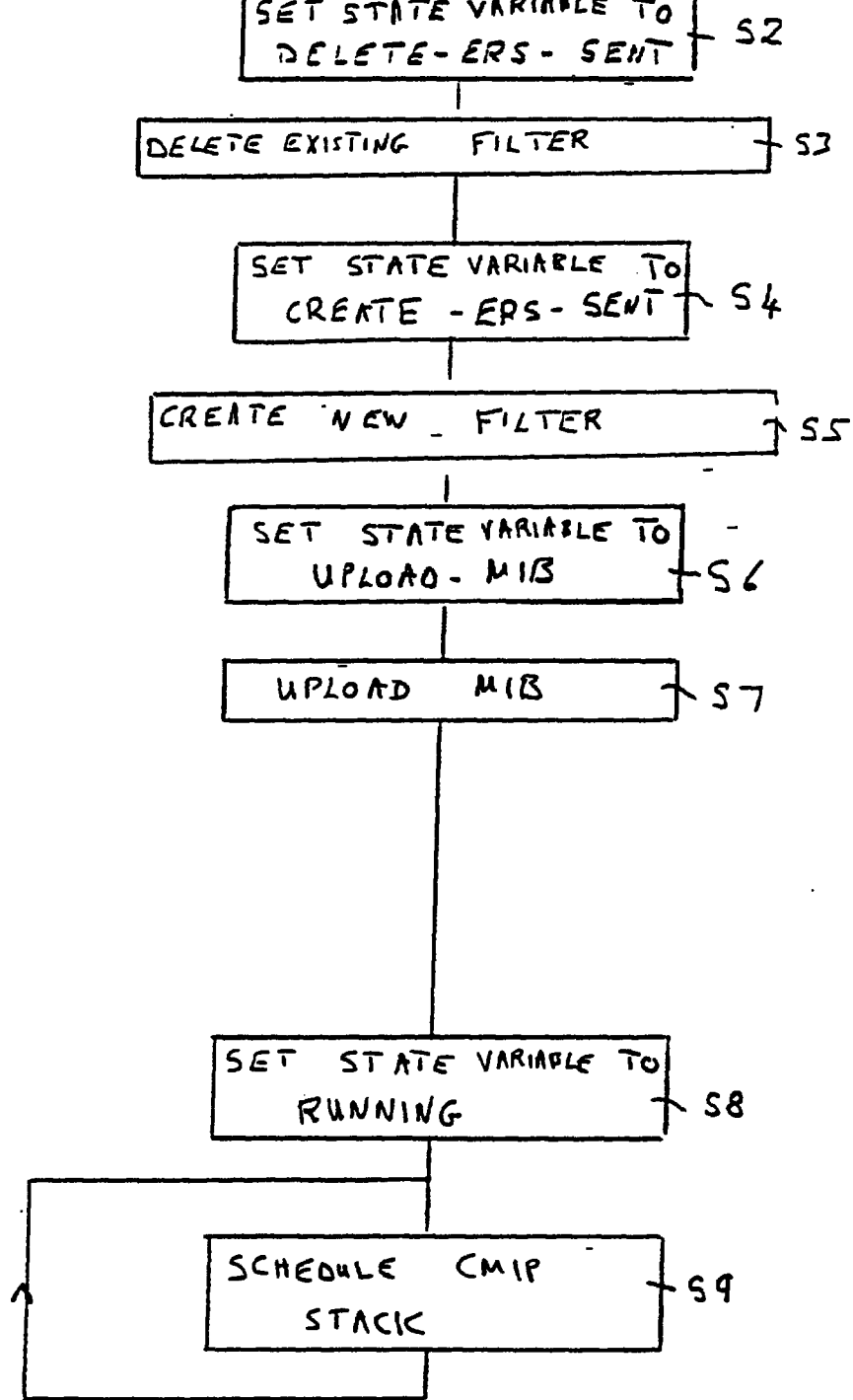


Figure 5

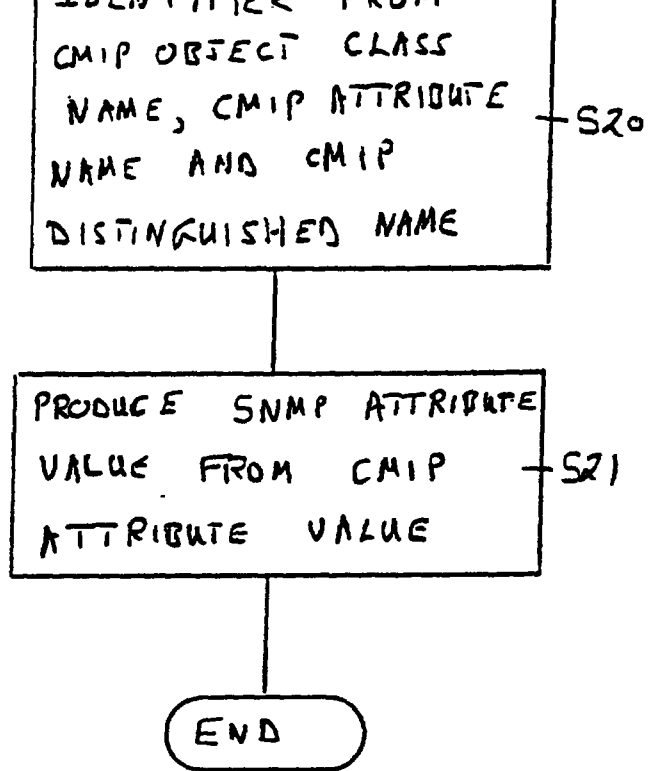


Figure 6

circuitId	pvcId
aEndPointName	pvcEndptA
zEndPointName	pvcEndptZ
administrativeState	pvcAdminState
operationalState	pvcOpState
circuitBandwidth	pvcCIR
userLabels	pvcFreeText

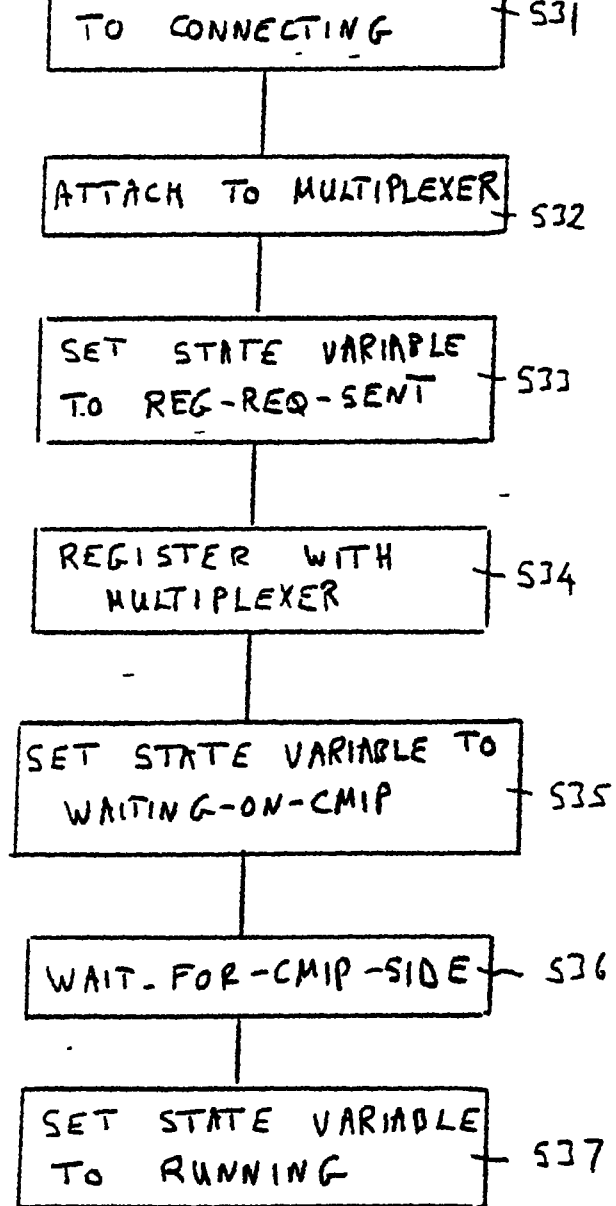
Figure 7

CMIP Attribute	SNMP Attribute
equipmentId	btrouterportId
administrativeState	btrouterportAdminState
operationalState	btrouterportOpState
typeText	btrouterportDesc
userLabels	btrouterportFreeText

Figure 8

CMR Attribute	SNMP Attribute
functionId	btGNSAccessId
administrativeState	btGNSAccessAdminState
operationalState	btGNSAccessOpState
protocolType	btGNSAccessProtocol
userLabels	btGNSAccessFreeText

Figure 9



A

Figure 10a

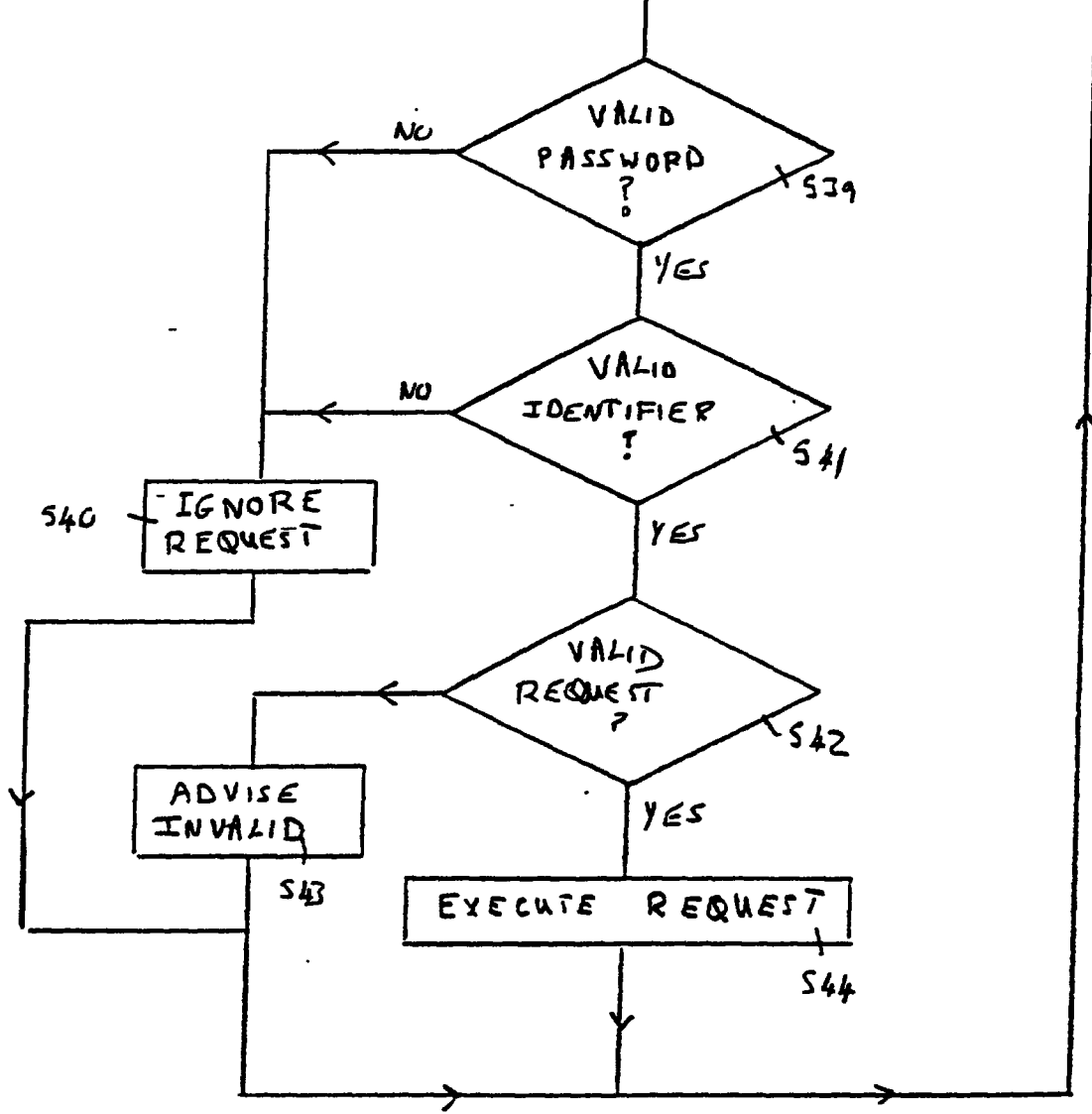


Figure 106

individual objects of a telecommunications network and also to a method of
5 operating such a data storage device.

It is well known that a main telecommunications network may be used to
provide communication links between local networks. The main
telecommunications network may be a private or public network belonging to a
public telecommunications company and the local networks may belong to various
10 other companies. Thus, where a company has local networks at two or more
locations, it may use the main network to connect them. It is known to provide
network managers for both main networks and for local networks. A network
manager receives data from the individual elements of the network which it
manages and can send instructions to these elements. Because these two types
15 of networks have developed separately and have different requirements, the
information protocol used by the network manager of a local network is usually
different from the information protocol used by the network manager of the main
network to which the local network is attached. For example, Simple Network
Management Protocol (SNMP) is commonly used in local networks while Common
20 Management Information Protocol (CMIP) is commonly used in main networks.
Although the human operator of a network manager of a local network may be
mainly interested in the elements of his network, it is also helpful for him to
receive information about some of the elements of the main network which is used
to connect his local network to other local networks. Because differing protocols
25 are used in the two types of network, it is difficult to transfer data easily from a
network manager of a main network to a network manager of a local network.

According to this invention, there is provided a data storage device for
storing data on individual objects of, or related to, a first telecommunications
network, said data storage device comprising: means for receiving data according
30 to a first information protocol on individual objects of, or related to, the first
telecommunications network from a network manager for that network; means for
converting the data received by the data receiving means from a form used in the
first information protocol into a form used in a second information protocol; means

the first telecommunications network according to the second information protocol
5 to a network manager of a second telecommunications network.

Because the data storage device of this invention converts data from a form used in the first information protocol into a form used in the second information protocol before storing it, it facilitates the supply of data according to the second information protocol to a network manager. Thus, the data storage
10 device of this invention may be used for transferring data from a network manager for a main network to a network manager for a local network. It may also be used, if desired, for transferring data from a network manager for a local network to a network manager for a main network.

According to a second aspect of this invention, there is provided a
15 network management system comprising a network manager for a telecommunications network and a data storage device according to the first aspect of this invention. The network management system may include a network manager for a second telecommunications network.

According to a third aspect of this invention, there is provided a method of
20 interconnecting a first network manager of a first telecommunications network using a first information protocol with a second network manager of a second telecommunications network using a second information protocol comprising the steps of:

in a data storage device receiving data according to the first information
25 protocol on individual objects of, or related to, the first telecommunications network from the network manager;

in the data storage device, converting the received data from a form used in the first information protocol into a form used in a second information protocol and storing the converted data; and

30 supplying the converted data on individual objects of, or related to, the first telecommunications network according to the second information protocol to the second network manager for the second telecommunications network.

main network

5 Figure 2 is a block diagram showing the relationship between a network manager for a main network, a network manager for a local network and a data storage device embodying this invention;

 Figure 3 is a block diagram of the software components of the data storage device shown in Figure 2;

10 Figure 4 is a diagram illustrating the operating states of the overall control component of the software shown in Figure 3;

 Figure 5 is a flow chart of the CMIP application component of the software shown in Figure 3;

 Figure 6 is a flow chart of the CMIP/SNMP mapper component of the
15 software shown in Figure 3;

 Figure 7 shows the CMIP attributes and the corresponding SNMP attributes of the object class for a private circuit used in the software shown in Figure 3;

 Figure 8 shows the CMIP attributes and the corresponding SNMP
20 attributes for the object class for a router port used in the software shown in Figure 3;

 Figure 9 shows the CMIP attributes and the corresponding SNMP attributes for the object class for an access point used in the software of Figure 3;
and

25 Figure 10 is a flow chart of the SNMP application component of the software shown in Figure 3.

 Referring now to Figure 1, there is shown a main network 10 and three local networks 12, 13, 14 which are connected to the main network 10. The main network 10 may be a public or private telecommunications network belonging to a
30 public telecommunications company. The local networks 12, 13, 14 represent only three of the many local networks which are connected to the main network 10. In the present example, the local networks 12, 13, 14 belong to the same organisation which is a completely separate organisation from the owner of the

switching node 17, and the local network 14 is connected to the switching node
5 19 through an access point 22 and a gateway 23. The router 21, access point 22
and gateway 23 are associated with and controlled by the main network 10 and so
may be regarded as part of the main network 10. The router 24 is associated with
and controlled by the local network 13. The dashed line 25 marks the boundary
10 between elements controlled by the local networks 12, 13, 14 and elements
controlled by the main network 10.

In the present example, the local networks 12, 13, 14, the router 21,
access point 22, gateway 23 and the main network 10 shown in Figure 1 form
part of a frame relay system. Although the present invention will be described
with reference to a frame relay system, it is to be appreciated that it may be used
15 in other types of telecommunications systems.

Referring now to Figure 2, the main network 10 has a network manager
30. The network manager 30 receives data from the elements of the main
network 10 and also the router 21, access point 22 and gateway 23 and sends
instructions to them. The network manager 30 is arranged to transmit and receive
20 data according to an information protocol known as the Common Management
information protocol (CMIP) and which provides a set of services known as the
Common Management Information Services (CMIS). When an element transmits
data in another protocol, the data is converted into CMIP. The local network 12
has a network manager 32 which receives information from and sends instructions
25 to the elements of the local network 12. The network manager 32 communicates
with these networks using an information protocol known as the Simple Network
Management Protocol (SNMP). As the construction of network managers is well
known, the network managers 30 and 32 will not be described in further detail.
As is also well known, a network manager is usually implemented as a general
30 purpose computer provided with appropriate software.

Although the human operator of the network manager 32 for the local
network 12 is mainly interested in the elements of that network, it is also valuable
for him to receive data on the state of the elements of the main network 10 as

that he can take appropriate action. Because the network managers 30 and 32
5 use different information protocols to communicate with their respective network
elements, data cannot be supplied directly from the network manager 30 to the
network manager 32. In order to enable the network manager 32 to receive data
on elements in the main network 10, the network manager 30 for the main
network 10 supplies data on these elements to a data storage device 34 which in
10 turn supplies the data to the network manager 32 for the local network 12. The
data storage device 34 will be described in more detail below. As will be
explained, the network manager 30 supplies data to the data storage device 34
using CMIP and the data storage device 34 supplies data to the network manager
32 using SNMP.

15 The data storage device 34 may be implemented as a general purpose
computer provided with appropriate software. In the preferred embodiment, the
data storage device 34 uses the same computer as the network manager 30 with
the result that the network manager 30 and the data storage 34 are located
together. Alternatively, the data storage device 34 may use the same computer as
20 the network manager 32 or it may be implemented by using a separate general
purpose computer which can be located with either of the network managers 30 or
32 or at a position which is remote from both of them.

Both the network managers 30 and 32 operate in what is known as an
object-oriented environment. In an object-oriented environment, computer
25 software is used to model real world physical objects as well as other real world
entities. The physical objects and entities are simply known as objects. Where, as
in the present case, the computer software is managing real world objects, it
models only those attributes of the real world objects which are necessary for
management. When objects are modelled in this way, they are known as managed
30 objects. A category of objects of the same or similar type is known as an object
type. A particular real world object is said to be an instance of an object type.
Thus, for example, router ports might be an example of an object type and the
individual ports of router 21 are instances of that object type.

object is known as an instance of the object class to which that object belongs.

5 Each software object is identified by the name of the object class to which it belongs and by an identifier which is specific to the software object itself. Each object class has a set of pre-defined attributes. For example, in the case of an object class for router ports, the attributes may include the operating state of the port. For an instance of a particular object class, each attribute would have a

10 particular value and the values are liable to change.

The network manager 30 uses a set of object classes for modelling the various types of object found in the main network 10. These object classes include three classes which, respectively, model private virtual circuits, router ports and access points. Each of these three object classes has a set of attributes.

15 Each attribute has a pre-defined name and may assume certain pre-defined values. The choice of attributes forming each set, the names of the attributes and the pre-defined values are appropriate for the network manager 30 and CMIP. Similarly, the object classes used by the network manager 32 include three object classes corresponding to the object classes used by the network manager 30 for private

20 virtual circuits, router ports and access points. The names of the attributes as well as their pre-defined values are appropriate for SNMP. For each of these three object classes used in the network manager 30 and the corresponding object class used in the network manager 32, the names of the attributes and values differ between the two corresponding classes. As will be described in more detail

25 below, the data storage device 34 can convert data on the attributes of a particular object class used in the network manager 30 into an appropriate form for the corresponding object class used in the network manager 32.

In SNMP, a network manager issues commands to obtain information from network elements. These commands include Get and Get-Next. A Get command

30 is a request for the value of a particular attribute of a particular object. A Get-Next command is a request for the value of the same attribute of the next object.

The services provided by CMIS include m-Get and m-EventReport. The m-Get service may be a request for the value of a specified attribute of a specified

of attributes of a particular object or a change in the value of a particular attribute
5 of a particular object. An m-EventReport is an unsolicited event report.

The software components of the data storage device 34 are shown in
Figure 3. These comprise a CMIP stack 40, a CMIP application component (CMIP-
APPL) 42, a CMIP/SNMP mapper component (MAPPER) 44, a management
information base (MIB) 46, an SNMP application component (SNMP-APPL) 48, a
10 TCP/IP communications stack 50, a UDP/IP communications stack 52, support files
(CFG) 54, an overall control component (O-CTRL) 56 and a trap functions
component (USER-EXITS) 58. Figure 3 also shows a multiplexer 60 together with
a TCP/IP communications stack 62 and a UDP/IP communications stack 64.

The CMIP application component 42 is responsible for sending requests to,
15 and receiving responses and unsolicited event reports from, the network manager
30. The CMIP/SNMP mapper component 44 is responsible for converting values of
the attributes of each object from the form used in CMIP to a form used in SNMP.
When data on the values of the attributes of the individual objects of a
telecommunications network is stored in a database, the database is known as a
20 management information base (MIB). MIB 46 stores the values of the attributes of
the objects of the telecommunications network 10 after conversion by the mapper
component 44. Thus, data is stored in MIB 46 in a form suitable for transmission
using SNMP. The SNMP application component 48 is responsible for receiving
requests from, and sending responses and unsolicited event reports to, the
25 network manager 32. In SNMP unsolicited event reports are called traps. The trap
functions component 58 is responsible for selecting which of the unsolicited event
reports from the network manager 30 should be passed on as traps to the network
manager 32. The various components of the data storage device 34 and also the
multiplexer 60 will now be described in more detail.

30 The CMIP stack 40 is responsible for converting CMIS requests from the
CMIP application component 42 into a form for transmission to the network
manager 30 and for converting responses from the network manager 30 into a
form suitable for the CMIP application component 42. The CMIP stack 40

unsolicited event reports from the communications stack to the CMIS applications
5 component 42. CMIS and CMIP are defined, respectively, in ISO/IEC Standards
9595 and 9596. A suitable software package for the CMIP handler is available
from British Telecommunications plc. A suitable software package for a
communication stack is available from Retix Corporation of Santa Monica,
California, USA.

10 The TCP/IP stacks 50, 62 enable data to be transmitted using the TCP/IP
protocols. Likewise, the UDP/IP stacks 52 and 64 enable data to be transmitted
using the UDP/IP protocols. Suitable software packages for the stacks 50, 52, 62
and 64 are commercially available. For example, the well known Sun operating
system includes both TCP/IP and UDP/IP stacks.

15 Although Figure 2 shows only a single network manager for a local
network receiving data from the network manager 30, the present invention may
be used to provide data from a network manager for a main network to more than
one network manager for a local network. A single data storage device can
provide data on the main network to all the local network managers belonging to a
20 particular customer as these local network managers will be interested in the same
elements of the main network. However, partly because different customers are
interested in different sets of elements of the main network and partly because it
may be necessary to restrict the supply of data on network elements for security
reasons, it is necessary to provide an individual data storage device for each
25 customer. The data storage devices may be located together or separately.
Requests from the various network managers for the local networks are received
on a common communication link by a multiplexer 60. Each request includes an
identifier for the network manager which is making it. The multiplexer 60 then
transmits the request to the appropriate data storage device. The responses and
30 traps are broadcast to the various network managers on a common communication
link.

The CMIP application component 42 exists in six operating states, namely,
STARTUP, DELETE-ERS-SENT, CREATE-ERS-SENT, UPLOAD-MIB, RUNNING and

RUNNING. The state variable for each of these components is maintained in the overall control component 56. For start-up, the state variable for both components
5 is set to STARTUP. For the CMIP application component, the state variable is changed from one state to another for the next four states by the CMIP application component. As will be explained in more detail below, in the state UPLOAD-MIB, initial data on the elements of the main network 10 are supplied to the CMIP application component. In the state RUNNING, the CMIP application component
10 receives unsolicited event reports from the network manager 30. For shutdown the state variable is changed to SHUTDOWN.

Following start-up the SNMP application component changes its state variable from one state to another for the remaining four operating states. As will be described in more detail below, during the states CONNECTING and REG-REQ-
15 SENT, the SNMP application component establishes a connection with, and registers itself with, the multiplexer 60. During the state WAITING-ON-CMIP, the SNMP application component waits for the completion of supplying the initial data to the CMIP application component. During the state RUNNING, the SNMP application component services requests from the network manager 32 and sends
20 traps to it.

The support files 54 include a list of the CMIP object classes which can be supported by the data storage device 34. In the present example, the data storage device 34 can support CMIP object classes for private virtual circuits, router ports and access points.

25 The CMIP application component (CMIP-APPL) 42 will now be described with reference to the flow chart shown in Figure 5.

Following start up of the data storage device 34, in a step S1 the CMIP application component performs initialisation routines. Then, in a step S2, it sets the state variable to DELETE-ERS-SENT.

30 CMIS provides a function known as a filter. When a filter is in place in an MIB, unsolicited event reports are issued when changes occur in the values of specified attributes of objects belonging to specified object classes. In step S3, the CMIP application component sends an instruction to the network manager 30

Next, in a step S5, the CMIP application component instructs the network
5 manager to create a new filter. This filter specifies both the object classes and the attributes of the three classes for which unsolicited event reports are required. The CMIP application component obtains the data for constructing the filter from appropriate support files 54. The state variable is then changed to UPLOAD-MIB in a step S6.

10 In order to prevent data storage devices from receiving data to which they are not entitled, the MIB in the main network manager 30 is partitioned and the data storage device 32 has access only to the data contained in the partition associated with it. In a step S7 the CMIP application component sends an m-get request to the network manager 30 for each object class for which it requires data.
15 The names of these object classes are retrieved from the support files 54. Thus, in the present example, it sends m-get requests for the private virtual circuits, router ports and access points object classes. For each object contained both within one of these classes and within the partition associated with the data storage device 34, the network manager sends the appropriate data to the CMIP
20 application component. Specifically, for each object, the main network manager sends the name of the object class, the identifier or distinguished name for the particular instance of the object class and the name and value of each attribute of the object.

The CMIP application component then passes the data to the CMIP/SNMP
25 mapper 44 and the data for that instance is stored in MIB 46. Thus, in this manner, the data for each object class is uploaded into MIB 46.

After receiving data on each instance, the CMIP application component checks whether that instance is the last instance on which data will be sent. If it is not the last instance, the CMIP application component receives data on the next
30 instance. If it is the last instance, it continues with a step S8 in which the state variable is set to RUNNING.

The operation of the CMIP/SNMP mapper component 44 will now be described with reference to Figures 6 to 9. Figure 6 shows a flow chart for this component. Figures 7 to 9 show respectively the CMIP attribute names together with the SNMP attribute names for the object classes for circuits, router ports and access points.

Referring firstly to Figure 7, it will be seen that the attributes names comprise circuitId, aEndPointName, zEndPointName, administrativeState, operationalState, circuitBandwidth and userLabels. These attribute names refer respectively to the identity of the circuit, the beginning point of the circuit, the end point of the circuit, its administrative state, its operational state, its bandwidth and text which the operator of the network manager 30 may add. Each attribute can have certain values. For example, for a particular circuit, circuitId is the identifier for the circuit and operationalState can have the values "enabled" and "disabled". The corresponding SNMP attribute names are shown beside the CMIP attributes names. For each value for a CMIP attribute, there is a corresponding value for an SNMP attribute. The SNMP object class also has two traps, namely, pvcCircuitDown and pvcCircuitUp. These two traps are issued, respectively, when the attribute operationalState changes to "disabled" and "enabled".

Referring now to Figure 8, the CMIP attributes for a router port comprise equipmentId, administrativeState, operationalState, typeText and userLabels. These refer respectively to the identity of the router port, its administrative state, its operational state, a verbal description of it and text which the operator of the main network 30 may add. Each of these may have one of a number of values. The corresponding SNMP attribute names are shown beside the CMIP attribute names. For each value of a CMIP attribute the corresponding SNMP attribute has a corresponding value. The SNMP object class also has two traps, namely, RouterPortDown and RouterPortUp. These two traps are issued, respectively, when the attribute operationalState changes to "disabled" and "enabled".

Referring now to Figure 9, the CMIP attributes for the access point object class comprise functionId, administrativeState, operationalState, protocolType and

SNMP attribute names are shown beside the CMIP attribute names. As with the
5 circuit and router port object classes, each CMIP attribute may take one of a
number of values and for each of these value there is a corresponding SNMP value.
The SNMP object class also has two traps, namely, GNSAccessDown and
GNSAccessUp. These two traps are issued, respectively, when the
operationalState changes to "disabled" and "enabled".

10 The mapper component 44 is used by the CMIP application component 42
for converting CMIP attribute values into SNMP attribute values both when
receiving the initial data on the various object classes during the operational state
UPLOAD-MIB and subsequently when receiving unsolicited event reports during the
operational state RUNNING. The mapper component converts each attribute value
15 in turn and Figure 6 shows the procedure for converting one attribute value.

In SNMP, for a particular attribute of a particular instance of a particular
class, the combination of the name of the class, the name of the attribute and the
distinguished name or identifier for the particular instance of the class is known as
the object identifier. In a step S20 for a particular attribute, the mapper
20 component converts the CMIP class name, attribute name and distinguished name
into the corresponding SNMP object identifier. The data for converting the class
names and attribute names is stored in the mapper component 44. The method of
converting the distinguished names will be described below.

In a step S21, the value of the attribute in CMIP is converted into the
25 corresponding value in SNMP. The method for doing this is described below.

In an SNMP MIB, the values of the attributes of the various instances of a
particular class are stored in a table dedicated to that class. Each row of the table
is associated with a particular instance of a class and each column is associated
with a particular attribute of the class. The data is stored as numerals. More
30 specifically, each class is identified by a unique series of numerals and each
attribute of the class is identified by a unique series of numerals. The CMIP
distinguished name for an instance of the class is converted into ASCII code.
Thus, an SNMP object identifier comprises a series of numerals for the class name,

can have only a limited number of states, each state may be identified by a
5 respective numeral. For example, in all three classes, for the attribute
operationalState, the values "disabled", "enabled", "active" and "busy" are
converted, respectively, into "1", "2", "3" and "4". Where an attribute value is
expressed as text, for example the value of the attribute userLabels, the text is
converted into ASCII code.

10 When the local network manager 32 receives an attribute value for a
particular object from the data storage device 34, the series of numerals
representing the SNMP object identifier and the numeral or numerals giving the
attribute value are converted by the network manager into text. Thus, for the
attribute operationalState, an attribute value of "1" is converted into "enabled".

15 The SNMP application component 48 will now be described with reference
to the flow chart shown in Figure 10. The primary purpose of this component is to
service Get and Get-Next requests.

In a step S30, the SNMP application component performs initialisation
routines. Then, in a step S31, it sets its state variable to CONNECTING. In a step
20 S32, it forms a connection with the multiplexer 60. In a step S33, it sets its state
variable to REG-REQ-SENT. Then, in a step S34, the SNMP application component
registers the data storage device 34 with the multiplexer 60 by supplying its
identifier to the multiplexer 60.

In a step S35, the SNMP application component sets its state variable to
25 WAIT-FOR-CMIP-SIDE. Then, in a step S36, it waits until the CMIP application
component has received all the initial data from the network manager 30 and
loaded the corresponding SNMP data into MIB 46. When all the initial data has
been loaded into MIB 46 and the CMIP application component has changed its
state variable to RUNNING, the SNMP application component changes its state
30 variable to RUNNING in a step S37. Then, in a step S38, the SNMP application
component continuously schedules the stack 50 for Get and Get-Next requests.

On receiving a request for an attribute value, in a step S39, the SNMP
application component checks if the request contains a valid password. If the

continues with a step S41. Each request contains an identifier for the local
5 network manager making the request. In step S41, the SNMP application
component checks whether it is authorised to send data to a local network
manager having this identifier. If the identifier is not valid, and the request cannot
therefore be authorised, the request is ignored in step S40. If the identifier is
valid, the SNMP application component continues with step S42.

10 In step S42, the SNMP application component checks if the request is
valid. For example, if the request is corrupted or relates to an object class on
which data cannot be supplied, it is not valid. If the request is not valid, the local
network manager is informed of this in step S43. If the request is valid, it is
executed in a step S44.

15 After steps S40, S43 and S44, the SNMP application component returns
to step S38.

As mentioned above, the SNMP application component sends unsolicited
event reports or traps for changes in the values of some attributes. For each
object class, the data in the support files 54 specifies the changes in attribute
20 value which will cause traps to be issued. In the present example, for each object
class, a change in the value of the attribute operationState to "enabled" or
"disabled" causes a trap to be issued. When the CMIP application component 42
receives details of a change in an attribute value, it checks with the support files
54 if the change in the attribute value is one for which a trap is issued. If it is one
25 for which a trap is issued, the CMIP application component 42 calls the trap
functions component 58 which in turn instructs the SNMP application component
48 to issue a trap to the network manager 32.

Although in the present example the data storage device is used to supply
data relating to a main network to a network manager for a local network, the
30 present invention may also be used to enable a network manager of a main
network to obtain data from a network manager of a local network.

to, a first telecommunications network, said data storage device comprising:

- 5 means for receiving data according to a first information protocol on individual objects of, or related to, the first telecommunications network from a network manager for that network;
- means for converting the data received by the data receiving means from a form used in the first information protocol into a form used in a second
- 10 information protocol;
- means for storing data on individual objects of, or related to, the first telecommunications network following its conversion by the data converting means; and
- means for supplying data from the data storing means on individual
- 15 objects of, or related to, the first telecommunications network according to the second information protocol to a network manager of a second telecommunications network.

2. A data storage device as claimed in claim 1, including means for

20 containing support data relating to object classes used in the first information protocol for which data can be stored in the data storage device.

3. A data storage device as claimed in claim 2, in which the data receiving means includes means for requesting data from said network manager for the first

25 telecommunications network for objects belonging to one of said object classes for which data can be stored in the data storage device.

4. A data storage device as claimed in any one of the preceding claims, in which the first information protocol is CMIP, the second information protocol is

30 SNMP, and for each attribute value the data converting means is arranged to convert the CMIP class name, attribute name and distinguished name into a corresponding SNMP object identifier and the CMIP attribute value into a corresponding SNMP attribute value.

network manager of the second telecommunications network when requested to
5 do so by said network manager of the second telecommunications network.

6. A data storage device as claimed in any one of the preceding claims, in
which the data supplying means is arranged to provide unsolicited information to
said network manager of the second telecommunications network when the value
10 changes of a pre-selected attribute of an object in, or related to, the first
telecommunications network.

7. A network management system comprising a network manager for a first
telecommunications network and a data storage device as claimed in any one of
15 the preceding claims.

8. A network management system as claimed in claim 7, in which said
network manager for the first telecommunications network and the data storage
device are physically located together.

20

9. A network management system as claimed in claim 7 or claim 8, further
including a network manager for a second telecommunications network.

10. A network management system as claimed in claim 9, in which the data
25 storage device and said network manager for the second telecommunications
network are physically located remotely from each other, the network management
system including a telecommunications link for connecting the data storage device
and said network manager for the second telecommunications network together.

30 11. A method of interconnecting a first network manager of a first
telecommunications network using a first information protocol with a second
network manager of a second telecommunications network using a second
information protocol comprising the steps of:

in the data storage device, converting the received data from a form used
5 in the first information protocol into a form used in a second information protocol
and storing the converted data; and

supplying the converted data on individual objects of, or related to, the
first telecommunications network according to the second information protocol to
the second network manager for the second telecommunications network.

10

12. A method as claimed in claim 11, including the additional step of in the
data storage device requesting data from said network manager for the first
telecommunications network for objects belonging to an object class for which
data can be stored in the data storage device.

15

13. A method as claimed in claim 11 or claim 12, in which the first
information protocol is CMIP and the second information protocol is SNMP, and, in
said step of converting the received data, for each attribute value the CMIP object
class name, attribute name and distinguished name are converted into the
20 corresponding SNMP object identifier and the CMIP attribute value is converted
into the corresponding SNMP attribute value.

14. A method as claimed in any one of claims 11 to 13, in which in said step
of supplying converted data, data on an object is supplied to said network manager
25 of a second telecommunications network when said network manager of the
second telecommunications network requests such data.

15. A method as claimed in any one of claims 11 to 14 in which in said step
of supplying converted data, unsolicited information is supplied to said network
30 manager of the second telecommunications network when the value changes of a
pre-selected attribute of an object in the first telecommunications network.