(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 November 2005 (10.11.2005)

PCT

(10) International Publication Number
WO 2005/106694 A2

(51) International Patent Classification[7]: G06F 15/173

(21) International Application Number:
PCT/US2005/013999

(22) International Filing Date: 25 April 2005 (25.04.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/564,837	23 April 2004 (23.04.2004)	US
60/565,064	23 April 2004 (23.04.2004)	US
11/112,624	22 April 2005 (22.04.2005)	US
11/112,942	22 April 2005 (22.04.2005)	US

(71) Applicant *(for all designated States except US)*: **ONARO** [US/US]; 46 Waltham Street, 6th Floor, Boston, MA 02118 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **ALON, Roee** [IL/IL]; 183 St. Botolph Street, Boston, MA 02115 (US). **LEVY, Assaf** [IL/IL]; 13 Nahal Kane Street, 44245 Kfsar-Saba (IL). **SCHARF, Shai** [IL/IL]; 20 Dubnov Street, Apt. 9, 64368 Tel Aviv (IL). **YAHALOM, Raphael** [US/US]; Onaro, 46 Waltham Street, 6th Floor, Boston, MA 02118 (US).

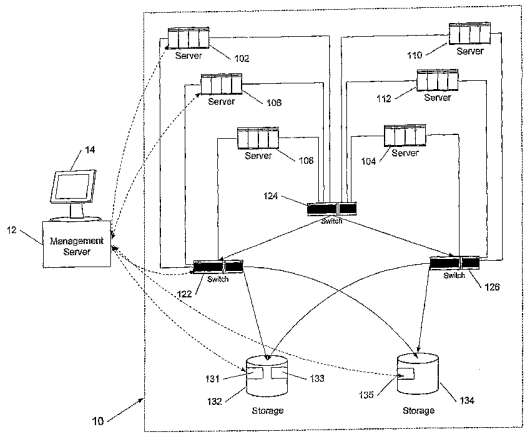
(74) Agents: **KELLY, Edward, J.** et al.; Ropes & Gray LLP, One International Place, Boston, MA 02110-2624 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO,

[Continued on next page]

(54) Title: METHODS AND SYSTEMS FOR HISTORY ANALYSIS AND PREDICTIVE CHANGE MANAGEMENT FOR ACCESS PATHS IN NETWORKS



(57) Abstract: A method and system for constructing and analyzing an access path event history and for predictively managing changes in a network is provided. Inconsistencies in the physical and logical access paths when making changes in the network are detected and analyzed. Component events associated with network appliances are recorded and an event sequence is constructed. Root causes for the inconsistencies are identified and the access paths are repaired in conformance with the network access path policy. For predictive change management, a configuration change can be specified and pre-validated according to the access path policy, before the actual changes are implemented. The conformance of the implemented configuration change with the access path policy is validated after implementation, because of possible consequential effects of the configuration change. If a change in one or more access paths is specified, then associated component events can be simulated before implementation, so that root causes for resulting access path violations can be detected and remedied before the changes are made. The disclosed systems and methods are useful for managing storage area networks (SAN).

WO 2005/106694 A2

SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHODS AND SYSTEMS FOR HISTORY ANALYSIS AND PREDICTIVE CHANGE MANAGEMENT FOR ACCESS PATHS IN NETWORKS

FIELD OF THE INVENTION

The invention is directed to methods and systems for constructing and analyzing logical end-to-end access path histories and predictive change management of these access paths in data networks. The methods and systems can be advantageous for managing access paths in storage area networks (SAN).

BACKGROUND OF THE INVENTION

Changes to IT infrastructures are the prime reasons for problem, disruptions, and vulnerabilities in such environments. The number of such changes in most environments is high due to growth, changing needs, and technological progress. Each such infrastructure change often consists of multiple individual tasks which need to be performed consistently, potentially in different locations by different persons. In most environments there are no effective end-to-end automated feedback mechanisms to ensure the consistency, which is important for changes to end-to-end access paths in data network infrastructures. Data networks are employed to transmit messages and data from a network appliance which initiates a network event, such as a query or a data transfer, subsequently also referred to as an initiator, application, server or host, to another network appliance which can respond to the event, for example, a data storage device. In various types of networks, for example in Storage Area Networks (SAN), defined access paths between the network appliances, may have to conform to an access path policy. The defined access paths are physical and logical pathways,

which can include the initiators, their particular components and ports, such as Host Bus Adapters (HBA), a switch fabric with switches, routers and the like, and end devices, such as physical storage devices, containing Logical Unit Numbers (LUN). The state of each of the components has to be properly logically configured to enable appropriate flow along the pathway. In addition, the physical pathways typically have to comply with a policy, also referred to as access path policy, which includes policy attributes, such as path redundancy, path connectivity characteristics, and the like.

One example of a data network with defined logical access path is a storage area network which enables multiple applications on servers access to data stored in consolidated, shared storage infrastructures. Enterprises increasingly deploy large-scale, complex networks to gain economies-of-scale business benefits, and are performing and planning extensive business-critical migration processes to these new environments.

Data networks are constantly undergoing changes, upgrades and expansion, which increases their complexity. The number of components and links which may be associated with the data transfer between a given initiator and one or more of its data network appliances may increase exponentially with the size of the network.

This complexity, which is compounded by the heterogeneity of the different network devices, leads to high risk and inefficiency. Changes to the network, which can happen frequently, may take a long time to complete by groups of network managers, and are error-prone. For example, in many existing enterprises a routine change, such as adding a new server to a network, may take 1-2 weeks to complete, and a high percentage (sometime 30-40%) of these change processes include at least one error. It is estimated that around 80% of enterprise outage events are a result of

events related to changes in the network infrastructure.

To implement the access relationships and related access characteristics, multiple underlying devices of different types and related physical/logical access paths between these devices need to be set up. The physical and logical set-up can include multiple actions (sometime tens per a single logical change), which need to be set up at different locations and with device types, with perfect mutual consistency.

It would therefore be desirable to detect inconsistencies in the physical and logical access paths when making changes in the network, to analyze these inconsistencies and to find the root causes for the inconsistencies between the changed access path and the access path policy for the network. It would also be desirable to detect such inconsistencies when planning changes in the network, to analyze these inconsistencies and to remedy the inconsistencies before the actual implementation of the changes.

SUMMARY OF THE INVENTION

The invention addresses the deficiencies in the prior art by, in various embodiments, providing methods and systems for constructing and analyzing logical access path histories in a network. In one embodiment, time-based sequences of consistent global snapshots of the network are constantly constructed and maintained for future reference. In another embodiment, planned changes in devices and device configurations of the devices connected to the network fabric and to the network fabric itself are analyzed and mapped to a consistent global snapshot of the network. The predictive change management process includes both pre-validation consistency checks before actions are taken, and post-validation consistency checks after the

action are taken. The consistency checks consider the current state of the access paths in the network, the current state of the access path policy and the set of new events , planned or executed, to determine conformance, or establish violation and its root cause.

A consistent global snapshot of a network is a representation which correctly reflects the actual status of all of the end-to-end access paths in the network at a particular point in time that are consistent with or conform to a defined network access path policy. The network access path policy specifies which access paths in the network between an initiator, such as an application in a SAN, and a data appliance, such as a LUN in a SAN storage appliance, should not exist, which should exist, and what should be the end-to-end attributes of each access path.

The status of each access path includes high level path attributes derived from the physical and logical characteristics of the components which provide the access relationship between a given initiator and a data appliance.

A management server automatically collects information about events and from devices distributed across the network, before and/or after they are implemented, using a variety of non-intrusive mechanisms. It identifies violations of actual access paths relative to the required access paths as determined by the policy. It provides notifications about violations, with all their relevant context information, to an appropriate target recipient. The management server, using network customized graph-based algorithms, analyzes the information and its impact on network access paths and compliance with the access path policy.

According to one aspect, the invention provides a process for constructing and

analyzing an access path event history in a network, which includes detecting events of one or more components in the network, determining a logical order or a temporal order, or both, of the events; and generating an event sequence based on the logical or temporal order.

According to another aspect, the invention provides a process for constructing and analyzing an access path event history in a network, which includes detecting at least one event causing nonconformance with a network access path policy, determining a root cause for the nonconformance, reconfiguring network resources based on the root cause to bring the reconfigured network in conformance with the network access path policy, and validating the reconfigured network..

According to still another aspect, the invention provides a process for updating logical access paths in a network, which includes detecting at least one component event in the network, checking conformance of a configuration change caused by the at least one component event with a network access path policy, and if the configuration change is in conformance, validating the logical access path.

According to an aspect, the invention provides a process for predictive change management of access paths in a network, which includes specifying one or more planned change tasks, pre-validating the one or more planned change tasks according to an access path policy of the network, implementing the one or more planned change tasks, tracking implementation of the one or more changes, and post-validating the implemented changes for conformance of with the access path policy.

According to another aspect, the invention provides a process for managing an access path change in a network, which includes specifying a change in an access path

policy in the network, associating with the specified change at least one component event, determining an effect of the least one component event by evaluating conformance of the changed access path policy, and in the event of nonconformance, determining a root cause for the nonconformance.

Advantageous embodiments of the invention may include one or more of the following features. Detecting an event may involve obtaining physical and logical state parameters from the components in the network. An access path violation may be associated with a first event in the event sequence, with the first event representing a root cause for the access path violation. The root cause can be analyzed and a corrective action for the component causing the access path violation can be determined. For example, the root cause can be determined based on event timing and reconciling discrepancies in the event timing.

Event timing can include receiving from network components timestamps associated with the events. Reconciling discrepancies in the event timing may include determining a relative time order of the timestamps based on semantics of a timestamp, a route for transmission of a timestamp, a multiplicity of messages with different timestamps for an identical event, and a causal relationship between events. Alternatively or in addition, reconciling discrepancies may include correcting, or proposing corrective action for, logical/physical path connectivity of the network components to bring the reconfigured network in conformance with the network access path policy. Moreover, a logical access path associated with the event can be revised, if the event is caused by a logical access path that is not conforming to the access path policy.

Events and associated root causes related to nonconformance can be successively

recorded and optionally collapsed into an updated access path representation. The recorded events and/or the collapsed access path representation can be visualized, for example, on a display.

One or more subsets of the events or of the network can be selected to provide a summary statistics of the events or a network status. The events can also be filtered so as to record state changes within a certain time interval, state changes affecting a specific component, or state changes impacting the access paths, or a combination thereof.

Other advantageous embodiments of the invention may include one or more of the following features. A planned change task may include adding, removing and/or changing a physical network component, a physical link between network components, a port configuration and/or a LUN mapping. The physical component can be a storage device, a switch or a server. The access path policy may be updated by adding an access path to the policy, deleting an access path from the policy, or changing an attribute of an access path. Moreover, a planned change task may involve modifying an access path attribute.

Pre-validating may include detecting a nonconformance of access paths with the access path policy and modifying the planned change task to conform the access paths to the access path policy. Post-validating may include detecting a nonconformance of the implemented change with the access path policy and notifying a user or determining a root cause of the nonconformance. The root cause may be determined from a logical order and/or a temporal order of the component event, and by generating an event sequence based on the logical and/or temporal order. Generally, the root cause can be associated with a first event in the event sequence and may be

eliminated by proposing a correction of one or more access paths.

Generating the event sequence may include determining a relative order of the component events based on semantics of a timestamp, a route for transmission of a timestamp, a multiplicity of messages with different timestamps for an identical event, and/or a causal relationship between events.

A planned change task, a pre-validated change task, an implemented change, and/or a post-validated configuration change may be visualized graphically and/or textually.

Further features and advantages of the present invention will be apparent from the following description of preferred embodiments and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The following figures depict certain illustrative embodiments of the invention in which like reference numerals refer to like elements. These depicted embodiments are to be understood as illustrative of the invention and not as limiting in any way.

FIG. 1 shows a schematic diagram of an exemplary network with physical links and a management server according to the invention;

FIG. 2 shows details of the management server of FIG. 1;

FIG. 3 shows two exemplary logical access path between a server and a storage device that comply with a network access path policy;

FIG. 4 is a schematic high-level flow diagram for constructing and analyzing an access path event history;

FIG. 5 shows a planned change in the logical access paths of the network after adding another server;

FIG. 6 is a schematic high-level flow diagram for managing predictive changes to access paths in a network;

FIG. 7 shows an exemplary visualization of a access path change plan and pre-validation;

FIG. 8 shows an exemplary textual visualization of a change execution tracking;

FIG. 9 shows an exemplary graphical visualization of a change execution tracking;

FIG. 10 shows an exemplary textual and graphical visualization of a violations analysis; and

FIG. 11 shows an exemplary textual and graphical visualization of a change recording and history management.

DETAILED DESCRIPTION

The methods and systems described herein enable multiple efficient, effective, and risk-free changes to access path environments in which initiators on multiple network appliances, such as servers or applications, to access (read and write) data which is stored on multiple other shared network appliances, such as storage devices. Exemplary embodiments of the methods and systems will be described with reference to a SAN having specialized SAN devices (such as different types of switches) which are interlinked, and may employ various transfer protocols (such as Fibre Channel and iSCSI). Each server is connected to the network through one or more specialized network cards (such as a Host Bus Adapter, or HBA). Application data can be stored on a storage device in storage units called LUNs (LUN = Logical Unit Number).

Although the invention will be scribed with reference to a SAN, a SAN is only one particular example of a network with defined physical and logical access paths and an associated access path policy, and it will be understood that the invention can be used with other types of networks, such as local area networks, wide area networks, extranets, remote-site connections, inter-domain networks, intranets, and possible the Internet, as long as access paths can be defined, set up, and monitored in such a network.

In the context of the invention, the following terminology will be used:

Access Path – a physical and logical connection or link between components in a network, which enables an initiator on one node of the network to access a data set on another node, while disabling other initiators from accessing the same data set, unless provisions are made for controlled sharing of the same data between several initiators. More particularly, when applied to the exemplary storage area network (SAN), the term *Logical Access Path* refers to a logical channel between a given application and a given LUN along which data can flow. The logical state (configuration state) of each component along the way in that sequence (for example the HBA, the storage controller, and each of the switches) is set so as not to disable data flow between that specific application and that specific LUN along that specific sequence of components.

Access Path Policy – specifies all the access paths that should exist, and required end-to-end properties for each of these access-path at any given point in time. For example, the access path policy can specify redundancy and replication of components and network appliances, number of hops, latency, constraints on component types inter-connections along the path, etc.

Access Path Event – a physical or logical change of a state of one of the components which are part of an access path. An access path event can be caused, for example, by a node failure, a faulty cable connection, or a change in a cable connection, or a zone configuration change.

Access Path Violation – a discrepancy at some point in time between the access path policy and the access paths in the network environment, which can be caused by an access path event and can be reflected, for example, in the existence/absence of an access path or in a change in the properties of one or more of the existing access paths.

FIG. 1 shows a topological view of an exemplary network 10, such as a storage area network (SAN), with several network appliances, for example peripherals such as servers 102, 104, 106, 108, 110, 112, switches 122, 124, 126, and application data storage devices 132, 134. The storage devices can be, for example, disk drives, such as RAID devices, tape drives, or other types of mass-storage devices. The physical connection paths between different network appliances are indicated by solid lines. Not all physical access paths are also logical access paths, because some physical access paths alone or in combination with other access paths may have characteristics that cause nonconformance with the access path policy. The storage devices 132, 134 can be further partitioned into data storage regions, such as unique LUNs 131, 133, 135.

The network 10 of FIG. 1 includes a management server 12 which for the exemplary network 10 is configured to communicate with various network components, including the network appliances, i.e., the servers 102, 104, 106, ... , 112, the storage devices 132, 134 and respective LUNs 131, 133, 135, and switches

122, 124, 126, to monitor the network and network resources and assure conformance between the logical access paths and the access path policy. This communication can take place via the communication channels used by the network for data transfer or via separate communication channels.

FIG. 2 shows in more detail an exemplary configuration of the management server 12. The management server 12 can include, inter alia, a Components Interaction Engine 202 which obtains information from the various network components in a manner described above. An Information Normalization Engine 204 converts the obtained information to a standard, device-independent representation, with the Information Reconciliation Engine 206 reconciling conflicts, removing redundant information and identifying incomplete information. Event Correlation Engine 208 establishes relationships between events and establishes an temporal and logical event sequence. The Validation Analysis Engine 210 compares the actual access paths with the access path policy and identifies access path violations. History Analysis Engine 212 selects events, for example, based on their causal relationships, filters events according to defined filter criteria, and performs statistical and trending analysis. The Root Cause Analysis Engine 214 analyzes the root cause(s) of detected violations and can optionally generate a root cause decision tree 216. An Event Repository 218 stores access path violations, events leading to these violations, root causes, etc., whereas a Policy History Repository 220 stores access path policies and changes in the access path policies.

FIG. 3 depicts the exemplary network 10 with logical access paths (shown as bold lines) set up between a network appliance 106, such as a server or an application, and a LUN, such as LUN 135, on storage device 134. The intermediate components

along the access path include, among others, intermediate nodes, such as switch 122 in one of the logical access paths, and switches 124, 126, in the other access path. It can be inferred from the network diagram of FIG. 3 that the access path policy requires redundant access paths. It will be understood that the illustrated network configuration serves as an example only, and that the configuration and routing of the access paths will depend on the type of device and number of devices employed at the network nodes (e.g., switches 122, 124, 126). Each network device may be set to logically constrain traffic flows through that device to specific respective end-points only (using different methods depending on the type of device and on the vendor). For example, each switch typically supports the definition of different type of *zones* which are sets of device ports between which data may flow via that switch. Storage devices typically support LUN masking which imposes constraints on the servers that may access a LUN. Similarly, a server's HBA (host bus adapter) typically supports types of LUN masking that constrain which LUNs can be accessed from that particular server.

Consequently, to enable a data flow end-to-end from a particular given initiator or application to a particular given data LUN, both physical constraints (at least one physical path must exist between the corresponding server and the corresponding storage LUN) and logical constraints (the zoning in each switch and the LUN masking at the HBA and storage device should be set so as not to disable data traffic between these end points) need to be satisfied in all the devices along that route.

For example, the logical setup on each of the two HBAs on server 106, the zone set up in each of the switches 122, 124, 126, as well as the LUN masking 135 at the storage device must be set to enable flows along each of these logical channels

between these two end points 106 and 135. For example, the zoning on switch 122 needs to be defined such that the port corresponding to server 106 and the other port corresponding to the storage device of LUN 135 are in the same zone. Logical access paths can include associated path attributes which can be considered as particular properties that characterize each end-to-end Logical Access Path, describing, for example, aspects of availability, performance and security of each given end-to-end logical channel. In other words, given a particular logical access path, a particular value can be computed for each one of the defined attributes (that value represents a particular property value for that logical access path instantiation).

The computation of a particular attribute value for a given logical access path, also referred to as path attributes, can be based on information related to the sequence of linked components, as well as on information about the types and internal configuration states of any number of components contained in that logical access path. The path attributes represent the characteristics of the end-to-end data flow between an initiator and its data. The path attributes relate to, for example, levels of end-to-end availability, performance, and security, all of which characterize data flows along a logical access path.

Monitoring the performance and compliance of a network with defined logical paths requires monitoring network appliances to detect access path events, analyzing the access path events to detect access path violations, and constructing and maintaining an access path history of access path violations through a sequence of end-to-end path snapshots of the access paths. These snapshots may have divergent and/or contradicting information and further information about the temporal and/or

logical sequence of events may hence be required. Discrepancies requiring reconciliation may be due to:

- Multiple devices of different types with a variety of interface protocols
- Different information contents from different component sources
- Different information representation formats and methods
- Different level of reliability of devices
- Different clock settings
- Varying information flow time delays
- Missing information due to failures of components and links

Returning to the exemplary network 10 depicted in FIG. 3, status information received from a variety of distributed sources, such as server 106, switch 122 and LUN 135, is processed at a central server, such as the network management server 12. Each such status information received from a component is parsed and translated from the source-specific and protocol-specific context representation to a normalized representation designed to capture device- and protocol-independent access path status information. Normalization is desired because, depending on the component type and the actual protocol used to communicate with the component, the information received from each component has a different syntactic and semantic representation, as well as variations in the information contents.

Normalization takes into account, among others:

- component identity
- component type and model
- component software version
- component physical connectivity
- component configuration information (representing logical connectivity constraints)
- component information about other components.

Status information can be received from each component source, either as an update response to a request or as a pre-defined, optionally periodically transmitted, component status update message. Each status update can contain a summary of the current state of the component, or information about a new component event that occurred at that component or at other parts of the network. For example, the switch 124 in network 10 may have failed, disabling one of the redundant access paths between server 106 and LUN 135, so that the logical access path between these two network appliances is no longer in compliance with the access path policy of the network 10 which requires redundancy. Similarly, access path policy may specify level storage redundancy and data redundancy (not shown).

As mentioned above, the physical connectivity information may include the identity of other components which are connected to this component's ports. The logical connectivity information may include various types of information flow constraints through the component, such as zoning, port binding, LUN masking, and the like.

The normalization process involves mapping the component-specific status information into an access path context paradigm. That is, deducing from the status of a component (physical and logical constraints) which access paths (potential data flows, from certain sources to certain destinations) are enabled or disabled by this specific component.

For example, a "device down" event (for example, inferred when no status response is received within a certain time after a request) may indicate that no data can currently be transmitted through the device. A "link down" event from a component may indicate that no data flow can currently be transmitted through a

particular port of that component. A "soft zone update" event from a component (e.g. switch) may imply that data can only flow between network appliances in a new zone configuration. A "LUN masking update" event from a component (e.g. a storage device) may imply that data can only flow between a specified source in the new LUN masking configuration to the corresponding destination storage LUN.

The normalized representation of component status information can then be aggregated consistently, as will be described below, to determine the status of all the access paths in the network at any point in time. Moreover, the information received from the various distributed components about local component events, such as their temporal and logical order, can be processed by the management server 12 to determine an event sequence of these events.

Determining the sequential ordering of the underlying distributed events may not be straightforward for various reasons. For example, some event information may not contain timestamps, or timestamps from different sources may have different semantics, or clocks of different distributed sources may not be fully synchronized. Alternatively or in addition, timestamps from different sources may represent time of message generation (or transmission) rather than time of a specific underlying event, or messages from different distributed sources may incur different levels of transmission delays, or event information from various sources may be duplicated or partially duplicated, or, due to transmission failure or misrouting, messages may become reordered or lost.

The management server 12 analyzes the messages received from each component and determines the "correct" event sequence, i.e., the relative time order or timeline of the underlying events based on:

- The semantics of the timestamp in the message as determined by the type of the source device and the nature of the interaction protocol between the central server and the source device. For example, in a switch that transmits, in response to a polling request, the updated zone configuration state, any specific new zone change event must be associated with a time point between the current snapshot time and the previous snapshot time for that switch.
- The characteristics of the route between the source device and the central server. For example, a component event message from a component connected to the central server via a direct local-area channel is likely generated at a more predictable recent point in time than an event message transmitted via a number of network hops.
- Elimination of multiple messages representing the same underlying event. For example, two different components may generate separate messages related to the same failure event. For example, a port failure at switch 124 may cause messages from both switch 124 and switch 126.
- Assessment of causal relationships between different events. For example, a zone configuration change in one switch in a fabric can trigger a propagation of corresponding update events at other switches in that fabric. In other words, the update events at the other switches are causally dependent on the configuration change in the first switch and therefore occur after the original event, i.e., the configuration change. In another example, disconnection of a cable from one port must have occurred before reconnection of the same cable to another port.

The aforedescribed exemplary steps are only illustrative and by no means exhaustive. The temporal event order taken together with the logical event order make it possible to create a causal and consistent representation of an event sequence of multiple component events in the order in which the events actually occurred in the network.

The management server 12 analyzes the consistent event sequence representation and maps the event sequence to a higher level access path representation by performing the following operations:

- Correlating multiple low-level events to determine a corresponding high-level state change. For example, a change event received from more than one source, for example, from switches 124 and 126 in an existing logical path, may indicate a new connection between two switches which represents a high level state change. Another example of a high level state is a zoning change in a switch fabric which may involve several low-level state changes in several switches in the fabric.
- Correlating multiple low-level events to identify redundant information about the same state change and deleting duplicative low level event information. For example, a device failure represents a high level state change which may result in one or more redundant messages from devices in addition to the time-out message from the failed device itself.
- Correlating multiple low-level events to obtain additional complementary information about a high-level event. For example, correlating between a host WWN identifier and its IP identifier may lead to additional information regarding a state change associated with that host.
- Correlating multiple low-level events to determine conflicting information relating to the same state change, and attempting to resolve the identified conflict based on consistency of information from different sources and known reliability of the components. For example, when a new switch or a new host is added to a switch fabric, some neighboring devices may detect and report this event, while others may be updated later and hence still provide outdated state information.
- Correlating multiple low-level events with information about a previously planned change in the network fabric. For example, one or more low-level component events, such as events received from a switch, may initially indicate a component failure, such as a disabled connection on one of the

switch ports may indicate. However, when the same information correlated with the planned change information, the failure may be diagnosed as being most likely due to a change in cabling as part of a planned migration task.

- Correlating multiple low-level events with a network-specific access path policy. For example, the network-specific access path policy may require dual fabric redundancy, so that a new low-level component event may indicate a new cable connection between a server and a storage device as part of a redundant path being set up.

- Establishing the impact of network state changes on the access paths and the access path attributes. For example, a zoning change or a LUN-masking change may imply that one or more new end-to-end (application/hosts to data/LUNs) access paths are established, or that one or more access paths cease to exist.

The above process defines a consistent event sequence of component state changes and their impact on the end-to-end access paths. This can be represented and visualized as a consistent access path event history in the network.

The consistent access path event history is used as a basis for various analysis tasks and to provide control, diagnostics, management, and audit functions. For example, the management server may record every time-stamped low level component event and store it in a dedicated repository. The server can determine from the event sequence of the low level component events corresponding high level access paths and compare the access path with the access path policy. The management server may also record the derived higher level access paths state changes in the repository. In addition, the management server may store the access path policy and changes to the access path policy of the network or of at least the part of the network managed by the management server.

Access path violations may occur for a variety of reasons, including planning mistakes, component failures, and human errors. One challenge addressed by the process of the invention is to identify the root cause of an access path violation. A root cause is defined as an event in the network fabric that causes a violation of access path policy, without being itself caused by an earlier, correctly time-stamped event that violated the access path policy. Whenever an access path violation is detected, effective corrective actions can be performed once the root cause is established.

Whenever the management server detects an access path violation, for example, caused by a component event, the server determines whether the access path was ever set-up correctly, i.e., did not have a preceding violation. If the access path had been set-up correctly, then at least the subset of access paths associated with the access path event history of that path is examined, and the earliest state change in the event sequence from a state without a violation to a state with a violation is identified as the root cause. The state change is presented, for example displayed on display 14, with its context information, including the corresponding low-level component events. Determining the appropriate corrective action is described below. In most situations, i.e. in the absence of consequential components events that lead to failure of another device, or of other unrelated component events, remedying the root cause will cure the access path violation due to the root cause.

If, on the other hand, the access path associated with the violation was not set up correctly (for example, because access path is just being constructed in conjunction with a specific policy update event), then the root cause of the violation may be due to one or more "missing state changes." Identifying these missing state changes is performed as part of the corrective action analysis.

The process of establishing the appropriate corrective actions for a given violation is performed by a combination of several approaches. For certain types of the root cause events and violations, a corresponding counter-event mapping is pre-defined. In general, a graph-analysis is performed to determine the optimal route (or change state sequences) which will lead from the current access path state (which includes the violation) to the target state (as specified by the corresponding policy). This analysis can also be supported by a knowledge-base representing pre-defined network procedures that can provide helpful hints and suggestions to users in cases where no single final determination about the best corrective action can be derived.

The aforedescribed process forms a basis for constructing and analyzing the access path history in a network. As shown in FIG. 4, the process 40 starts with a network state where all network appliances are communicating via valid physical and logical access paths that conform to the defined access path policy for that network or section of network, step 402. The management server 12 collects information from the network appliances, either continuously or periodically, as described above, enabling the server to identify one or more component events, step 404. The server then attempts to determine a temporal sequence of the component events based, for example, on timestamps of these events, step 406. However, since a single component event, such as a component fault or a configuration change, can cause multiple events to be indicated to the server 12, which are a consequence of the first event without representing in themselves a component malfunction, the temporal sequence is analyzed and transformed into a consistent event sequence based on likely or necessary causal relationships between the individual events, step 408, and mapped

onto a logical access path representation, step 410, which can be visualized on monitor 14.

The access paths are then compared 412 with the defined access path policy 414 which can be stored 416 in a policy repository. If the logical access paths comply with the access path policy, no action is required, and the event is logged in an access path history file, step 430, the access path representation is updated, if necessary, step 432, and the server 12 continues to monitor the network.

Conversely, if the logical access paths do not comply with the access path policy, then the root cause for the access path violation(s) is determined, step 418, based, for example, on the consistent event sequence determined in step 408. It is first determined in step 420, if the access path was set up correctly, because the root cause event can also be triggered, for example, by a component change which complies with a presumably correct access path that was, however, set up incorrectly. In this case, the missing or incorrect state change is identified, step 424, and changes in the logical/physical connectivity are proposed, step 426. If, on the other hand, the access path was set up correctly, as determined in step 420, then the earliest component event (root cause) identified in step 418 is assumed to have been the trigger event, step 422, and the component is repaired or the physical connection redefined and/or rerouted, step 426, and the resulting logical path is validated and mapped onto the access path representation. The changes are then logged, step 430, and the access path representation updated, step 432, as described before.

The mechanisms for analyzing and correcting root cause violations are applicable to "actual violations" as well as "pending violations." Actual violations reflect events that have already occurred in the network environment. Pending

violations reflect planned state change events which have not yet been performed in the network environment. As both these events are similarly represented in the state change history structure (with past timestamps and future timestamps, respectively), the analysis mechanisms for both these cases can be constructed in an analogous manner.

Predictive Change Management, as described below, is designed to improve the reliability and efficiency of access path change processes in IT infrastructures, such as managed networks having an access path policy, for example, storage area networks (SAN). According to one embodiment, the management server 12 interacts with the network appliances and network resources in the network fabric and implements a process with the following main aspects:

- Specifying an Access Path Policy or a change in the current Access Path Policy
- Determining a current valid global state of the network
- Specifying a proposed change event in network
- Pre-validating the proposed change event
- Implementing the pre-validated change
- Validating the implemented change
- Updating the global state
- Record changes and implications for validation as an audit trail

Before contemplating a change in the current network configuration that could potentially affect the access paths, the management server receives state information from the various network appliances and fabric components, correlates the information, reconciles inconsistencies, and constructs a representation of the current state of the infrastructure environment and of all access paths that exist in the environment at that point in time, as well as their access path attributes.

The representation of the existing access paths and path attributes is compared with the corresponding representation in the access path policy repository 214 of the management server 12 (see FIG. 2), violations are identified, and appropriate notifications are generated. The types of possible violations include, without being limited thereto, "access path does not exist", "access path should not exist", and "access path attribute value discrepancy". When appropriate corrective actions are taken, the server obtains the updated state information, and the violations are deleted from the list.

Access path policy may change by, for example, adding one or more new access paths between two network appliances (with particular attributes), deleting one or more access path, or changing access path attributes. The network configuration can also change due to one or more actions related to component changes, such as: connecting or removing a link between two device ports; connecting devices to or disconnecting devices from links; and/or changing a logical access state configuration on a device to enable data flows between devices and/or ports. For example, the action "logical access configuration change at device R" is mapped to a detailed zoning configuration or LUN masking.

Referring now to FIG. 5, in the depicted example, a new server 103 is slated to be added to the network. The network policy stipulates that the server 103 is connected to storage LUN 135 with dual fabric redundancy and no more than one hop to maintain a low latency. One access path meeting this access path attribute can be established between an HBA of server 103 and LUN 135 via switch 126. However, a second possible access path via switches 124 and 122 has two hops and therefore violates the access path attributes. No second access path exists that only includes a

single hop. Accordingly, a new access path 142 is established between switch 124 and LUN 135. Although no additional network components are required and no other changes need to be made to add the access path 142, this may not be the case in larger and/or more complex networks. Adding or changing one or more access path may involve addition or reconfiguration of a number of components, which may in turn affect other access paths that previously conformed to the access path policy.

FIG. 6 shows an exemplary process 50 for predictive change management according to the invention. The sequential order of the depicted steps may be changed and additional process steps may be performed as long as the illustrative process enables predictive change management of access paths in a network.

The network is presumed to have a defined access path policy, step 502, so that a valid state of existing access paths can be established, step 504. A proposed network change plan is specified, which may add and/or change physical network components, links, port settings, LUN masking, and the like, step 506. The details of each proposed change in the plan are pre-validated after specification and before their implementation, step 508. Pre-validation is performed by simulating the effect of constituent proposed actions, i.e., the addition of server 103 and the two redundant links via switch 124 and 126. In addition, the effect of these actions on the representation of the infrastructure is determined, and any deviations in the resulting state representation from the specified required policy rules are identified.

Specifically, the effect of each action on the environment is simulated and a list of access paths is derived. For example, a specific logical configuration update of a single component can open new access paths, close existing access paths, as well as change some attributes of existing access paths. For example, the addition of path

142 between switch 124 and LUN 135 also opens connections between servers 102, 106, 108 and LUN 135 having a lesser number of hops. Any identified deviations from policy are presented, analyzed, and can be corrected, simulated, and pre-validated again in an iterative process, step 514.

The result of a successful pre-validation phase is a detailed execution plan for the change which includes the actions to be performed including their detailed device-specific action implementation details. Moreover, a failure in an access path can be correlated with the performance of a component in the access path, which would allow the generation of a forward-looking temporal sequence or timeline of future component events in the access path. For example, by collecting and analyzing time-stamped information from the various components in a simulated implementation, a root cause for an access path failure can be determined before the changes are implemented, as discussed below. Accordingly, necessary repairs and an access path reconfiguration can be easily and predictably pinpointed and cost-effectively performed.

Once the proposed change plan is pre-validated, it may be implemented in the infrastructure environment based on a pre-established action plan, step 516. The change implementation can be performed in a variety of ways, including physical changes in the environment (re-cabling, connecting components), logical re-configuration using component-specific management interfaces or other provisioning solutions. Different parts of the change plan can be implemented in parallel by diverse IT personnel.

The actual implementation of the change plan is continuously tracked and analyzed by the server based on update messages received from the components in the

network and mapped to the change execution plan and the access path policy, step 520. The server records the individual state change actions (what was performed, where, when, by whom), and the evolving network state until the planned changes are completed.

Validation of each implemented change includes establishing its consistency with respect to the pre-validated change plan as well as its consistency with respect to the specified policy, step 522. Any deviation from the change plan or from the specified policy triggers appropriate notifications, step 524.

Each such notification can include context information suggesting a root cause, step 526, and/or specifying proposed corrective actions, step 528. The process 50 then returns to step 516, so that each corrective action can be iteratively processed through the predictive change phases cycle, or parts of it, until successful completion. After all changes have been implemented and found to be consistent with the access path policy, as determined in step 522, also referred to as post-validation, details of the performed changes can be recorded in an access path history file, such as the Event History Repository 218 (see FIG. 2), which can be maintained as long as necessary for future reference, for change cycle statistics, and as a guideline for managing future access path changes.

Post-validation of the changes may be necessary not only because certain actions, for example, installation of a cable connection, may have errors, but also other access path may have been affected by the changes, or other access paths may have been changed from their state before pre-validation due to events unrelated to the specifically implemented changes.

FIG. 7 represents an example of a visualization of access paths change plan to be pre-validated. Outlined are the planned high level tasks, the detailed change low-level individual actions to be pre-validated and performed, and the consequent future changes implied by these low-level actions. The pre-validation is performed based on the current state of the access paths in the network, the access path policy, and the set of planned low-level actions. Any future violation implied by these is detected , highlighted, and notifications are generated.

FIG. 8 represents an example of a visualization of change execution tracking. Outlined are these low-level actions which the system detects as being completed, by interacting with the components in the network environment. In the example, the system detected that actions 1, 2 , and 5 of the plan were the only ones completed at that point in time, as denoted by the ticks inserted to the boxes next to these.

FIG. 9 represents an example of a graphical visualization of change execution tracking. The current state of the network and access path is depicted, with the impact of the latest detected change action highlighted. In this example the state of the network is depicted after the execution of action 5 in the plan, in which a cable was connected between component UP2001 and component USCG_D1_SW4.

FIG. 10 represents an example of a textual and graphical visualization of violations and their analysis. Depicted are a list of violations detected in the environment at a particular point in time, for each the graphical representation of the affected access paths, as well as the details of the changes that are associated with these violations. For example the 1st violation shown represent a redundancy violation in an access path between host 3 and storage 2, created by a change that occurred 9/23/03 at 5:32pm.

FIG. 11 represents an example of a textual and graphical visualization of change history. It contains all the change events that occurred in the system at any point in the past up to now, ordered correctly on a timeline. For each change event, context information is provided, and the state of the access paths in the network at the point in time in which the change event occurred, is depicted graphically.

As the data structure which includes the incremental component events and the incremental access path changes may grow rapidly, the low level event sequences may be collapsed into a new global state representation after violations of the access path policy have been rectified and/or changes to the access path policy have been implemented. Each such global state is a representation of the network at a specific point in time and can be viewed graphically. Each collapsed global state representation is computed by starting from the last collapsed state and applying sequentially each new state change. Zoom-in and zoom-out capabilities, aided by graphic visualization, enable to view details of low-level events, corresponding higher level state changes, affected access paths, and/or the corresponding network state representation.

A planned change task, a pre-validated change task, an implemented change and/or a post-validated configuration change can be displayed graphically or in a text window, with tasks that have been performed or that still have to be performed, marked on the graphs or in the corresponding text fields.

A variety of filtering and query capabilities enable selection and presentation of subsets of the history and subsets of the network, according to any selection criteria. Indexing structures enable selection, for example, of all the state changes within a certain time interval, affecting a specific component, or impacting certain access

paths, and others. Moreover, comprehensive summary statistics on either network states or on the change processes themselves can be prepared.

While the invention has been disclosed in connection with the preferred embodiments shown and described in detail, various modifications and improvements may be made thereto without departing from the spirit and scope of the invention. By way of example, although the illustrative embodiments are depicted with reference to a storage area network (SAN), this need not be the case. Instead other networks with defined access paths may employ the method of the invention, and the network fabric may include any type of device that provides the described access constraints between network appliances. Accordingly, the spirit and scope of the present invention is to be limited only by the following claims.

What is claimed is:

CLAIMS:

1. A process for constructing and analyzing an access path event history in a network, comprising:

detecting events of one or more components in the network;

determining a logical order or a temporal order, or both, of the events; and

generating an event sequence based on the logical or temporal order.

2. The process of claim 1, including associating an access path violation with a first event in the event sequence, said first event representing a root cause for the access path violation.

3. The process of claim 2, including analyzing the root cause and determining a corrective action for the component causing the access path violation.

4. The process of claim 1, wherein determining the temporal order includes receiving from the one or more components a timestamp associated with the events.

5. The process of claim 1, wherein generating the event sequence includes determining a relative order of the events based on semantics of a timestamp, a route for transmission of a timestamp, a multiplicity of messages for an identical event, or a causal relationship between events, or a combination thereof.

6. The process of claim 3, including validating the access path after the corrective action was performed.

7. The process of claim 2, including analyzing the root cause and determining a causal relationship with other access path violations in addition to the access path violation caused by the at least one event.

8. The process of claim 8, further including selecting one or more subsets of the events or of the network, or both, for providing a summary statistics of the events or a network status.

9. The process of claim 8, further including filtering the events so as to record state changes within a certain time interval, state changes affecting a specific component, or state changes impacting the access paths, or a combination thereof.

10. A process for constructing and analyzing an access path event history in a network, comprising:

detecting at least one event causing nonconformance with a network access path policy,

determining a root cause for the nonconformance,

reconfiguring network resources based on the root cause to bring the reconfigured network in conformance with the network access path policy, and

validating the reconfigured network.

11. The process of claim 10, wherein determining the root cause includes determining event timing of the at least one event, and reconciling discrepancies in the event timing of the at least one event.

12. The process of claim 10, wherein determining the event timing includes receiving timestamps from network components, the timestamps being associated with the at least one event.

13. The process of claim 11, wherein reconciling discrepancies in the event timing includes determining a relative time order of the timestamps based on semantics of a timestamp, a route for transmission of a timestamp, a multiplicity of messages with different timestamps for an identical event, and a causal relationship between events.

14. The process of claim 11, wherein reconciling discrepancies includes correcting, or proposing corrective action for, logical/physical path connectivity of the network components to bring the reconfigured network in conformance with the network access path policy.

15. The process of claim 10, wherein detecting an event includes obtaining physical and logical state parameters from the components.

16. The process of claim 10, further comprising revising at least the logical access path associated with the at least one event, if the event is caused by a logical access path that is not conforming to the access path policy.

17. The process of claim 10, further comprising successively recording events and associated root causes for the nonconformance.

18. The process of claim 17, further comprising collapsing the successively recorded events into an updated access path representation.

19. The process of claim 17, further comprising visualizing the recorded events.

20. The process of claim 18, further comprising visualizing the collapsed access path representation.

21. The process of claim 10, further including selecting one or more subsets of the recorded events or of the network, or both, for providing a summary statistics of the events or a network status.

22. The process of claim 10, further including filtering the events so as to record state changes within a certain time interval, state changes affecting a specific component, or state changes impacting certain access paths, or a combination thereof.

23. A process for updating logical access paths in a network, comprising:

detecting at least one component event in the network,

checking conformance of a configuration change caused by the at least one component event with a network access path policy, and

if the configuration change is in conformance, validating the logical access path.

24. A process for predictive change management of access paths in a network, comprising:

specifying one or more planned change tasks,

pre-validating the one or more planned change tasks according to an access path policy of the network,

implementing the one or more planned change tasks,

tracking implementation of the one or more changes,

checking conformance of the implemented changes with the access path policy, and

post-validating the access path in the network.

25. The process of claim 24, wherein specifying a planned change task includes at least one of adding, removing and changing a physical network component, a physical link between network components, a port configuration or a LUN mapping, or a combination thereof.

26. The process of claim 25, wherein the physical component is selected from the group consisting of storage device, switch and server.

27. The process of claim 24, wherein the access path policy is updated by adding an access path to the policy, deleting an access path from the policy, or changing an attribute of an access path.

28. The process of claim 24, wherein specifying a planned change task includes modifying an access path attribute.

29. The process of claim 24, wherein pre-validating includes detecting a nonconformance of access paths with the access path policy and modifying the planned change task to conform the access paths to the access path policy.

30. The process of claim 24, wherein post-validating includes detecting a nonconformance of the implemented change with the access path policy and notifying a user.

31. The process of claim 24, wherein post-validating includes detecting a nonconformance of the implemented change with the access path policy and determining a root cause of the nonconformance.

32. The process of claim 31, further including proposing a correction of at least one access path to eliminate the root cause.

33. The process of claim 24, further including visualizing graphically or textually, or both, a planned change task, a pre-validated change task, an implemented change, or a post-validated configuration change, or a combination thereof.

34. A process for managing an access path change in a network, comprising:

specifying a change in an access path configuration in the network,

associating with the specified change at least one component event,

determining an effect of the least one component event by evaluating conformance of the changed access path configuration with an access path policy of the entwork, and

in the event of nonconformance, determining a root cause for the nonconformance.

35. The process of claim 34, wherein determining the root cause includes:

determining a logical order or a temporal order, or both, of the at least one component event; and

generating an event sequence based on the logical or temporal order.

36. The process of claim 35, wherein the root cause is associated with a first event in the event sequence.

37. The process of claim 35, wherein generating the event sequence includes determining a relative order of the at least one component event based on

semantics of a timestamp, a route for transmission of a timestamp, a multiplicity of messages with different timestamps for an identical event, or a causal relationship between events, or a combination thereof.

38. A process for constructing and analyzing an access path event history in a network, comprising:

detecting nonconformance between an access path state and an access path policy, and

determining at least one future change event which, if it occurs, will result in conformance of the network access paths with the access path policy.

39. The process of claim 38, further comprising validating the conformance of the access paths after the change events have occurred.

40. A network management system for managing access path in a network, comprising:

a components interaction engine receiving event information from network components,

an event correlation engine determining a logical or temporal order, or both, of events, and generating an event sequence based on the logical or temporal order.

41. The network management system of claim 40, further comprising an information normalization engine for converting the received event information to a normalized, device-independent event representation.40,

further comprising an information reconciliation engine for correcting or eliminating inconsistent event information received from network components.

42. The network management system of claim 40, further comprising a root cause analysis engine which associates a first event in the event sequence with an access path violation, said first event representing a root cause for the access path violation.

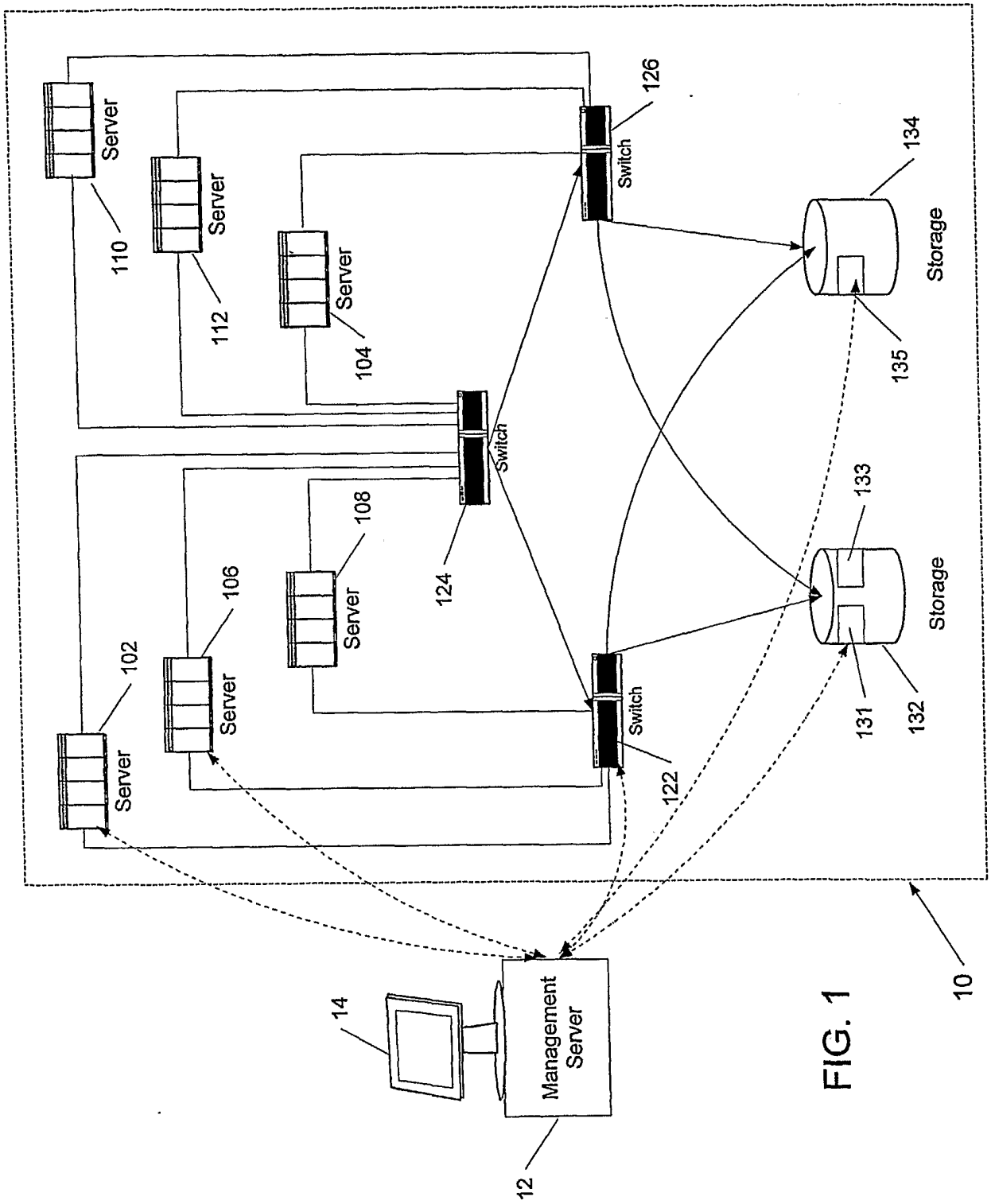

Server
110
Server
112
Server
104
Switch
126
Storage
134
135
Switch
124
Server
108
Server
106
Server
102
Switch
122
Storage
133
131
132
Management Server
12
14
10

FIG. 1

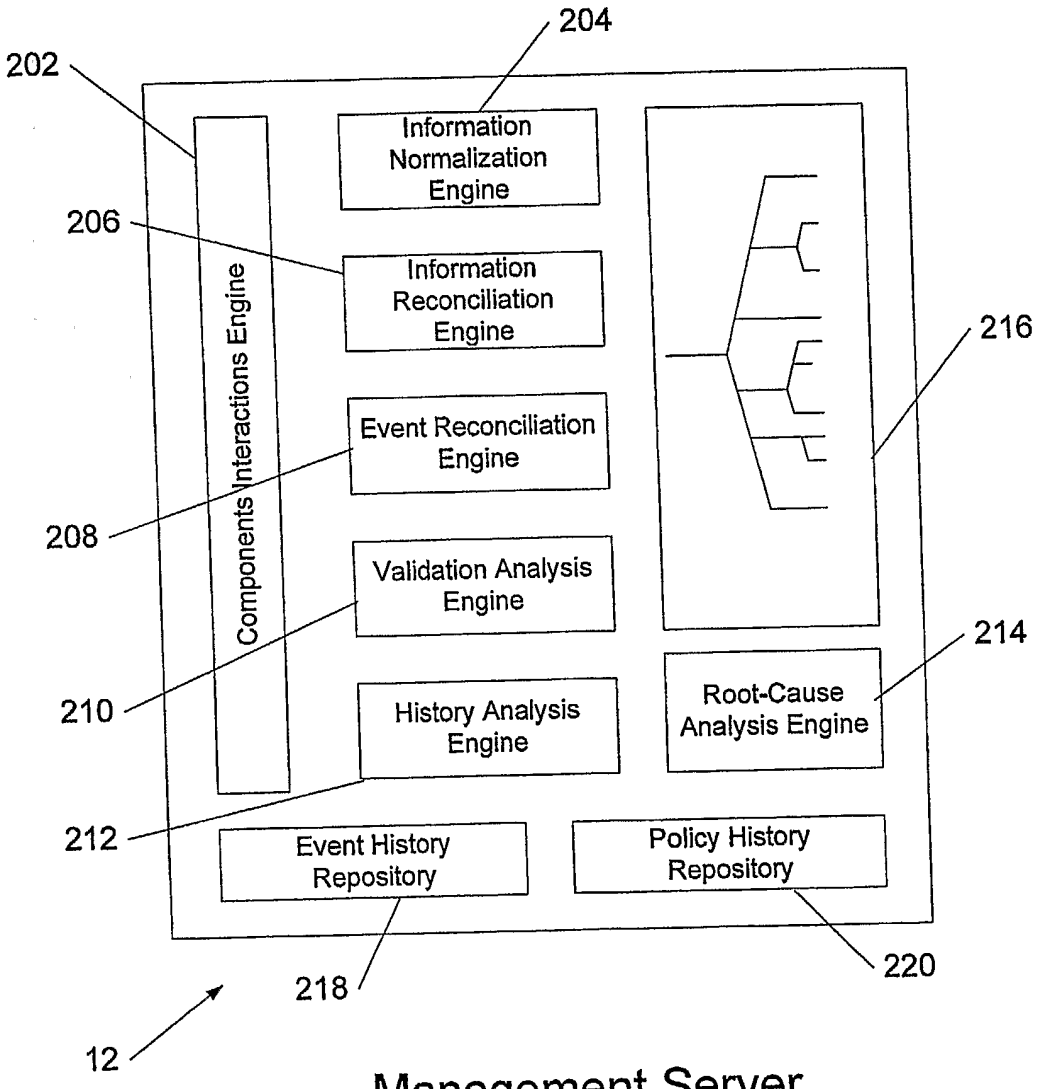

202
204
206
208
210
212
214
216
218
220
12
Components Interactions Engine
Information Normalization Engine
Information Reconciliation Engine
Event Reconciliation Engine
Validation Analysis Engine
History Analysis Engine
Root-Cause Analysis Engine
Event History Repository
Policy History Repository
Management Server

FIG. 2

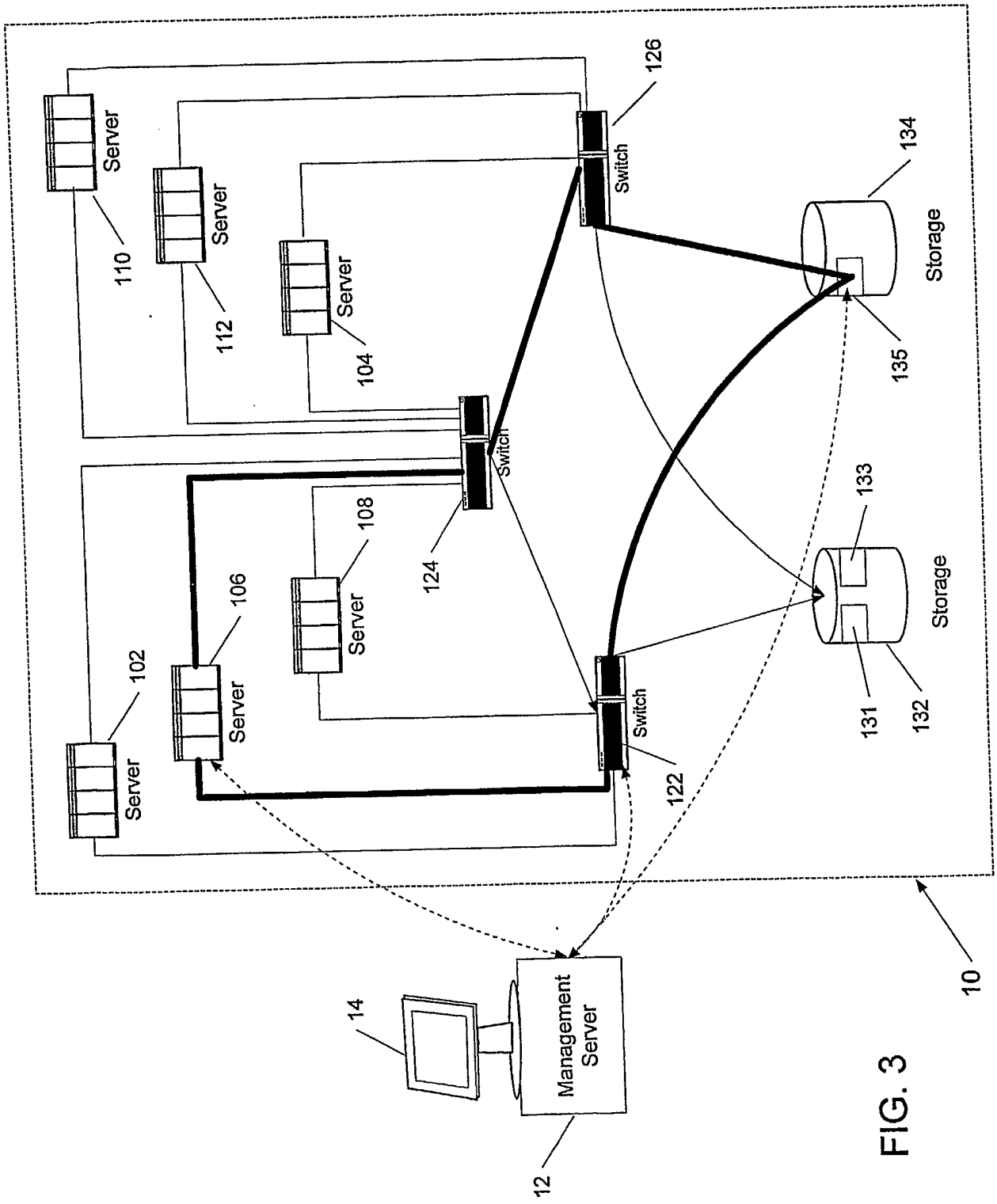

Server
110
Server
112
Server
104
126
Switch
134
Storage
135
Switch
108
Server
124
106
Server
102
Server
133
Storage
131
132
Switch
122
10
14
Management Server
12

FIG. 3

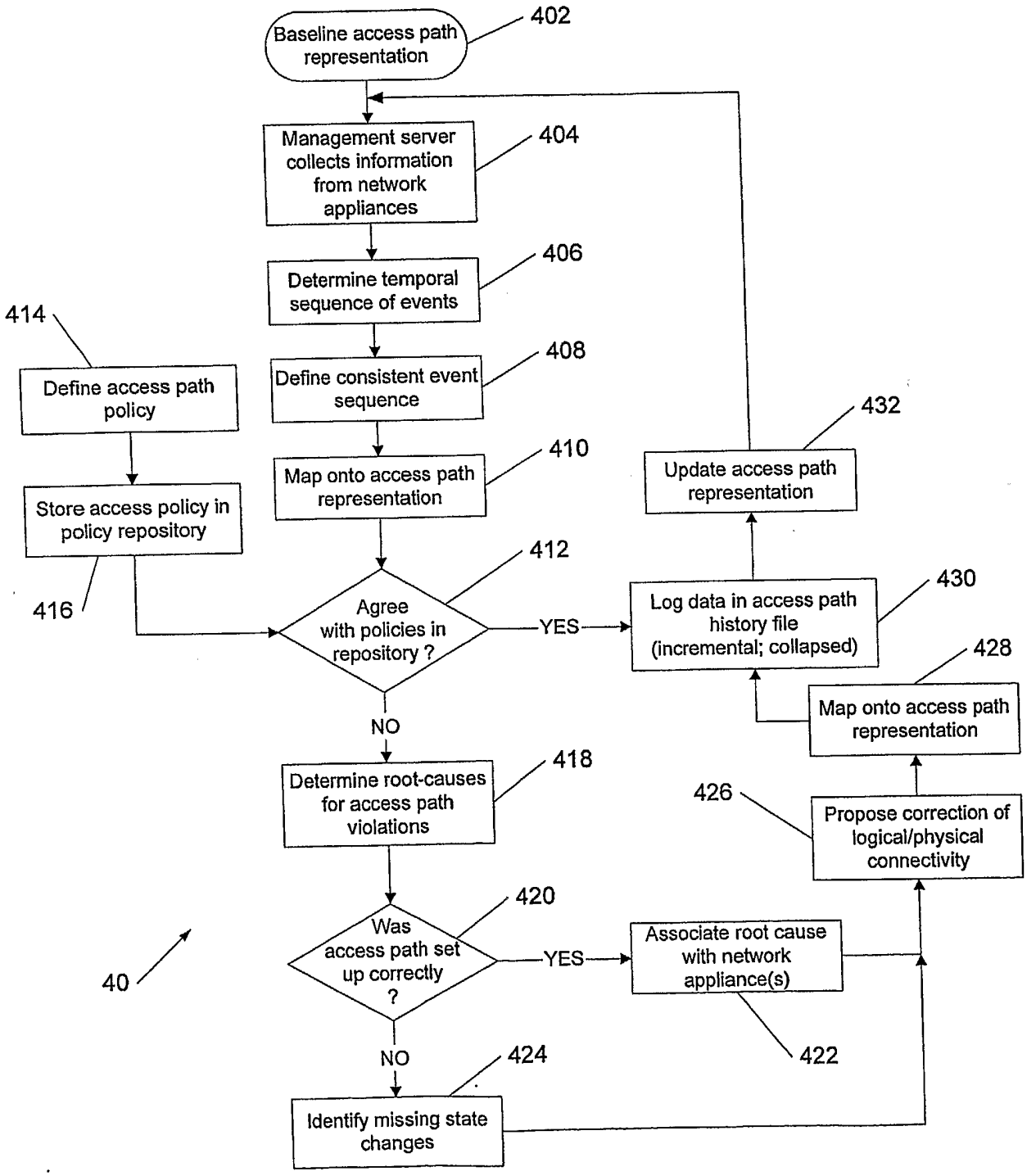

Baseline access path representation
402
Management server collects information from network appliances
404
Determine temporal sequence of events
406
Define consistent event sequence
408
Map onto access path representation
410
Agree with policies in repository ?
412
YES
NO
Define access path policy
414
Store access policy in policy repository
416
Log data in access path history file (incremental; collapsed)
430
Update access path representation
432
Map onto access path representation
428
Propose correction of logical/physical connectivity
426
Determine root-causes for access path violations
418
Was access path set up correctly ?
420
YES
NO
Associate root cause with network appliance(s)
422
Identify missing state changes
424
40

FIG. 4

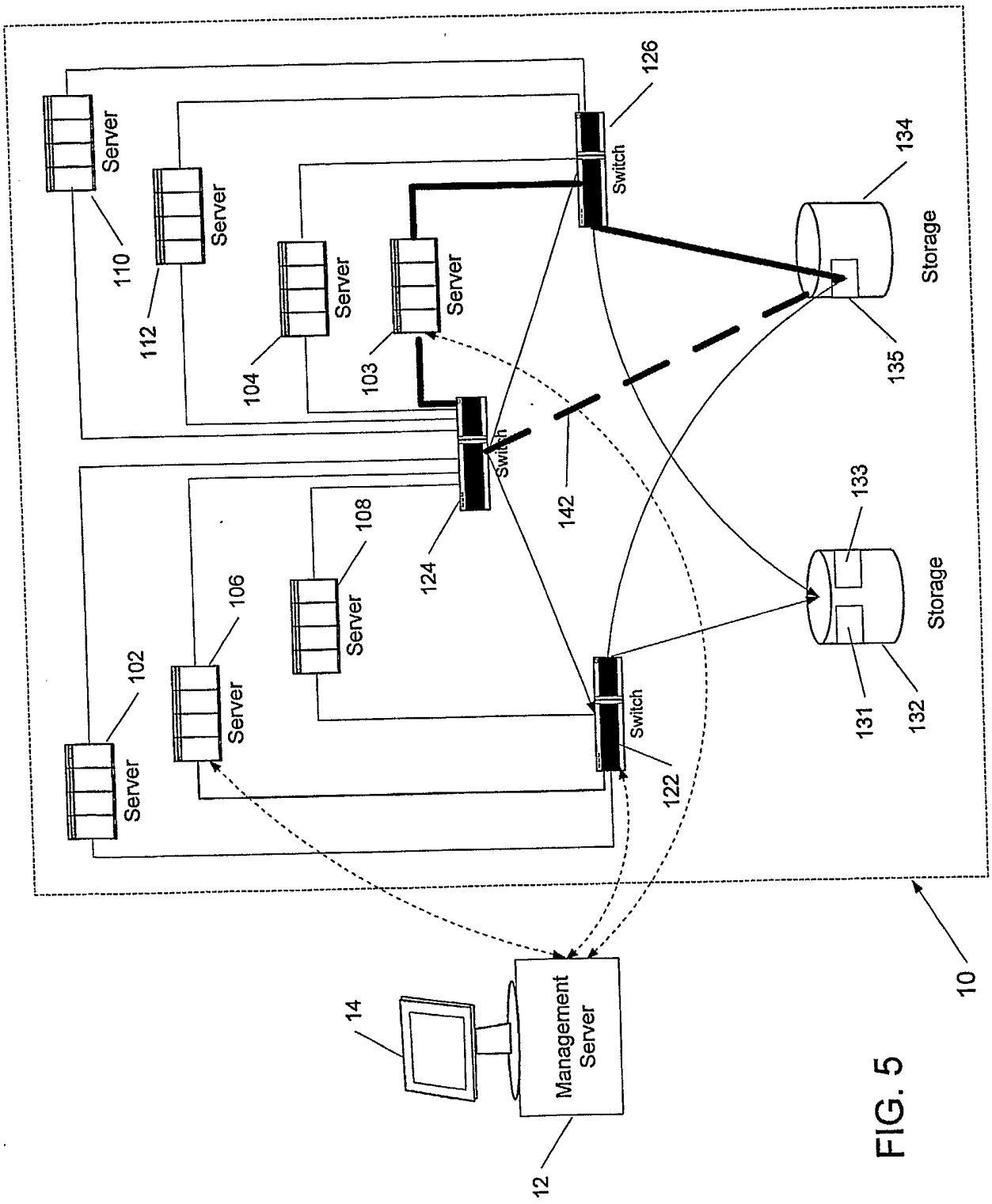

Server
110
Server
112
Server
104
Server
103
126
Switch
134
Storage
135
142
Switch
108
Server
124
106
Server
102
Server
Switch
122
133
Storage
131
132
10
14
Management Server
12

FIG. 5

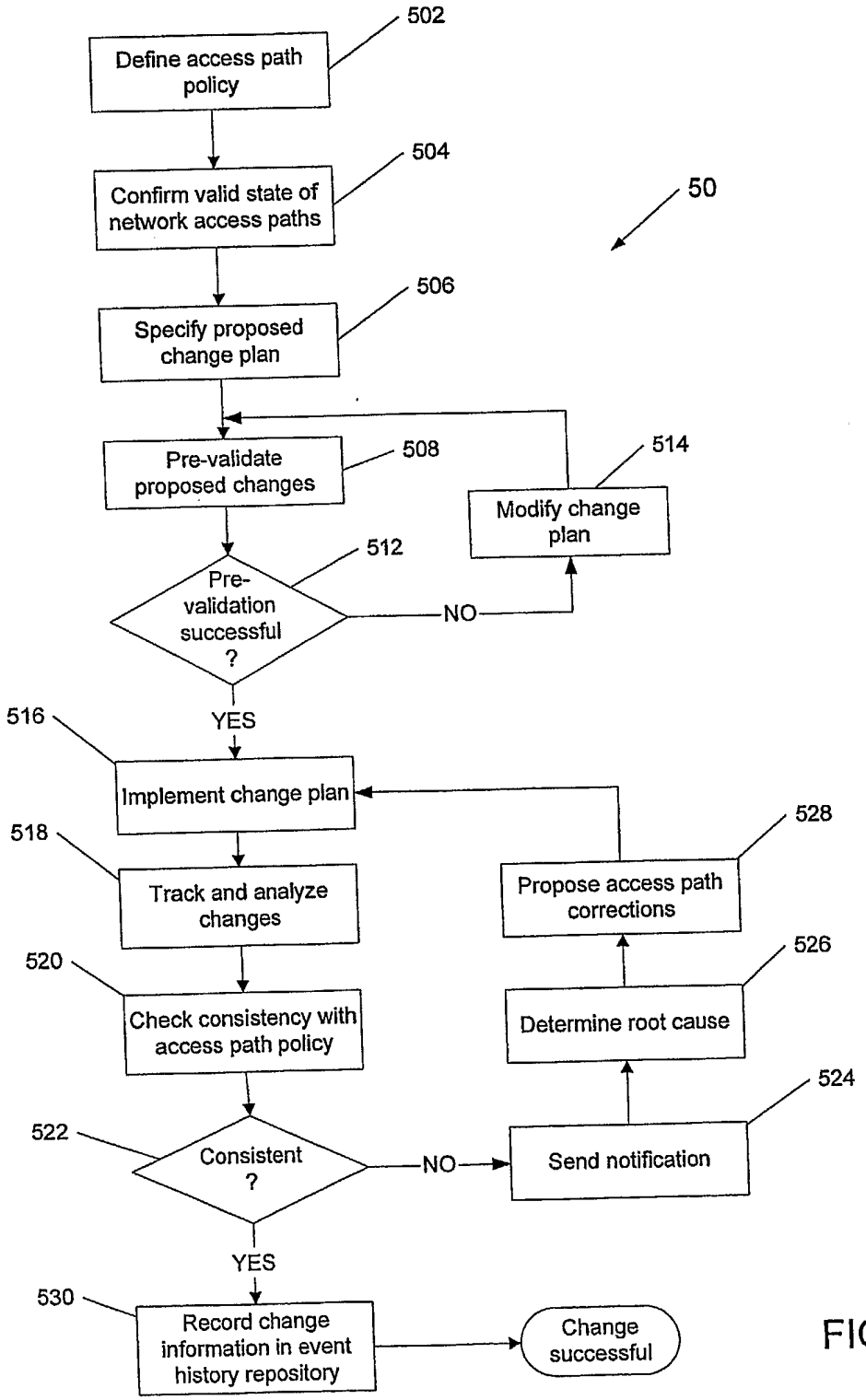

Define access path policy
502
Confirm valid state of network access paths
504
50
Specify proposed change plan
506
Pre-validate proposed changes
508
514
Modify change plan
512
Pre-validation successful ?
NO
YES
516
Implement change plan
518
Track and analyze changes
528
Propose access path corrections
520
Check consistency with access path policy
526
Determine root cause
524
522
Consistent ?
NO
Send notification
YES
530
Record change information in event history repository
Change successful

FIG. 6

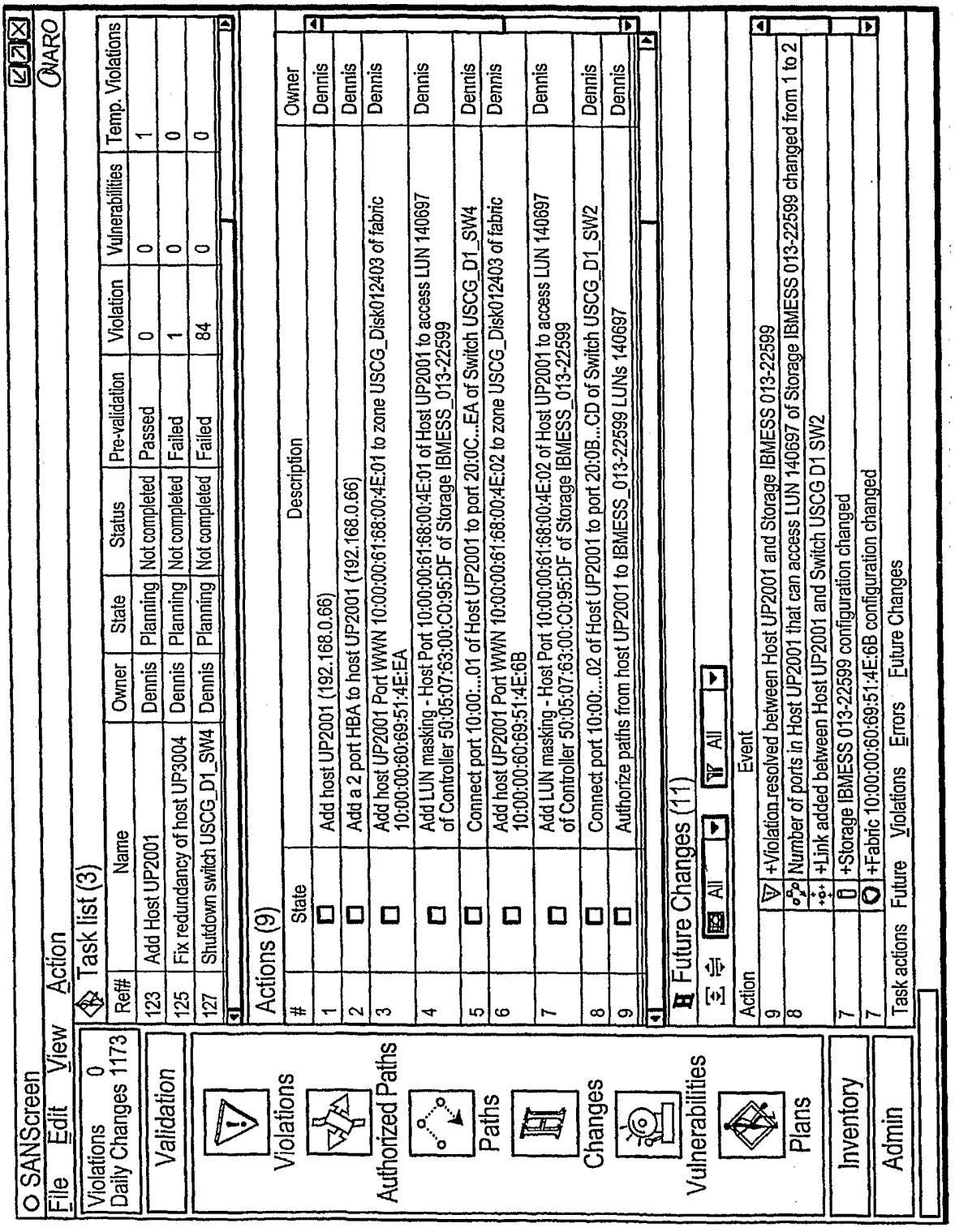

SANScreen
File Edit View Action
NARO
Violations 0
Daily Changes 1173
Validation
Violations
Authorized Paths
Paths
Changes
Vulnerabilities
Plans
Inventory
Admin
Task list (3)
Ref# Name Owner State Status Pre-validation Violation Vulnerabilities Temp. Violations
123 Add Host UP2001 Dennis Planning Not completed Passed 0 0 1
125 Fix redundancy of host UP3004 Dennis Planning Not completed Failed 1 0 0
127 Shutdown switch USCG_D1_SW4 Dennis Planning Not completed Failed 84 0 0
Actions (9)
State Description Owner
1 Add host UP2001 (192.168.0.66) Dennis
2 Add a 2 port HBA to host UP2001 (192.168.0.66) Dennis
3 Add host UP2001 Port WWN 10:00:00:61:68:00:4E:01 to zone USCG_Disk012403 of fabric 10:00:00:60:69:51:4E:EA Dennis
4 Add LUN masking - Host Port 10:00:00:61:68:00:4E:01 of Host UP2001 to access LUN 140697 of Controller 50:05:07:63:00:C0:95:DF of Storage IBMESS_013-22599 Dennis
5 Connect port 10:00:...01 of Host UP2001 to port 20:0C...EA of Switch USCG_D1_SW4 Dennis
6 Add host UP2001 Port WWN 10:00:00:61:68:00:4E:02 to zone USCG_Disk012403 of fabric 10:00:00:60:69:51:4E:6B Dennis
7 Add LUN masking - Host Port 10:00:00:61:68:00:4E:02 of Host UP2001 to access LUN 140697 of Controller 50:05:07:63:00:C0:95:DF of Storage IBMESS_013-22599 Dennis
8 Connect port 10:00:...02 of Host UP2001 to port 20:0B...CD of Switch USCG_D1_SW2 Dennis
9 Authorize paths from host UP2001 to IBMESS_013-22599 LUNs 140697 Dennis
Future Changes (11)
All
All
Action Event
9 +Violation resolved between Host UP2001 and Storage IBMESS 013-22599
8 Number of ports in Host UP2001 that can access LUN 140697 of Storage IBMESS 013-22599 changed from 1 to 2
+Link added between Host UP2001 and Switch USCG D1 SW2
7 +Storage IBMESS 013-22599 configuration changed
7 +Fabric 10:00:00:60:69:51:4E:6B configuration changed
Task actions Future Violations Errors Future Changes

FIG. 7

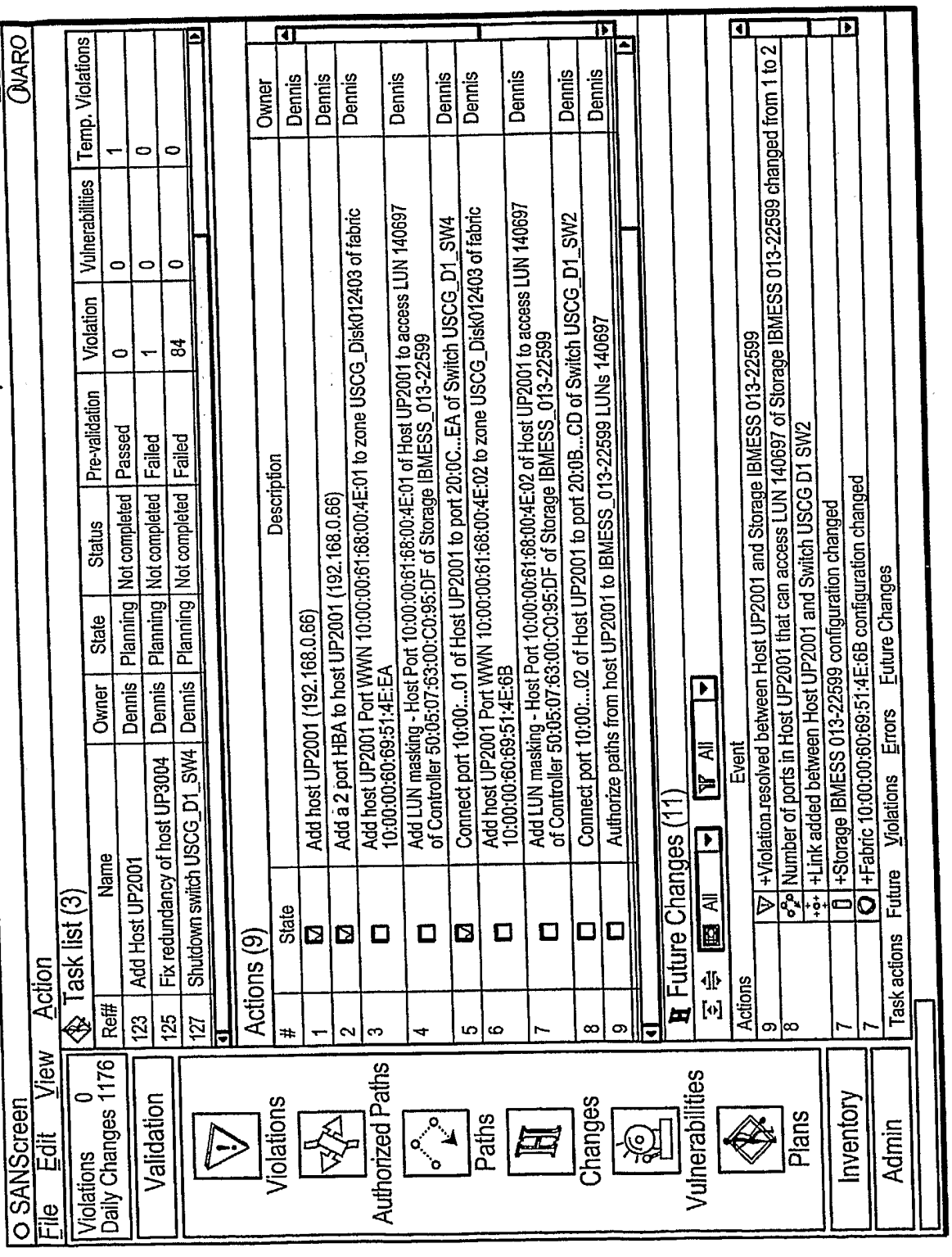
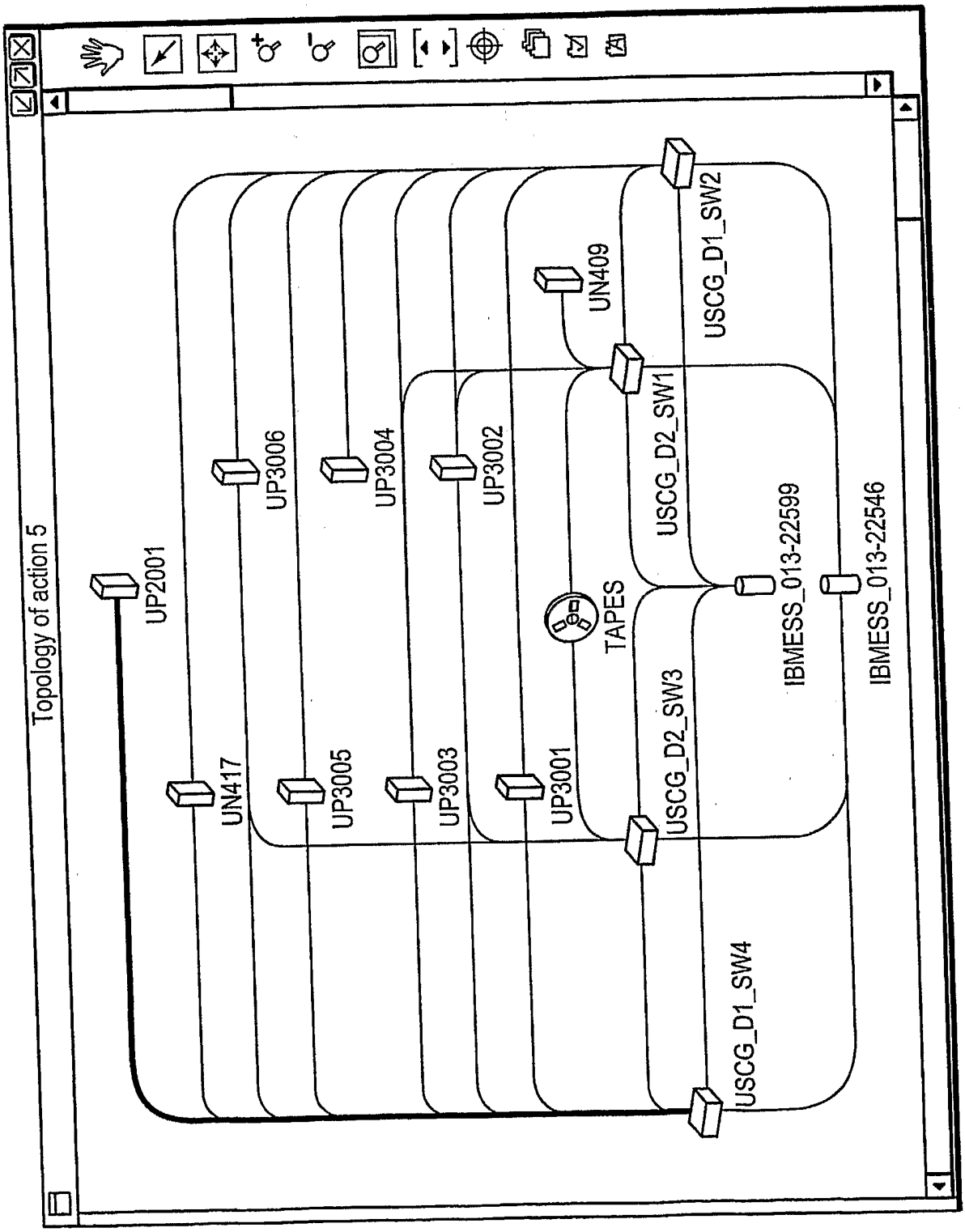

SANScreen
ONARO
File Edit View Action
Violations 0
Daily Changes 1176
Validation
Task list (3)
Reff# Name Owner State Status Pre-validation Violation Vulnerabilities Temp. Violations
123 Add Host UP2001 Dennis Planning Not completed Passed 0 0 1
125 Fix redundancy of host UP3004 Dennis Planning Not completed Failed 1 0 0
127 Shutdown switch USCG_D1_SW4 Dennis Planning Not completed Failed 84 0 0
Violations
Authorized Paths
Paths
Changes
Vulnerabilities
Plans
Actions (9)
State Description Owner
1 Add host UP2001 (192.168.0.66) Dennis
2 Add a 2 port HBA to host UP2001 (192.168.0.66) Dennis
3 Add host UP2001 Port WWN 10:00:00:61:68:00:4E:01 to zone USCG_Disk012403 of fabric 10:00:00:60:69:51:4E:EA Dennis
4 Add LUN masking - Host Port 10:00:00:61:68:00:4E:01 of Host UP2001 to access LUN 140697 of Controller 50:05:07:63:00:C0:95:DF of Storage IBMESS_013-22599 Dennis
5 Connect port 10:00:...01 of Host UP2001 to port 20:0C...EA of Switch USCG_D1_SW4 Dennis
6 Add host UP2001 Port WWN 10:00:00:61:68:00:4E:02 to zone USCG_Disk012403 of fabric 10:00:00:60:69:51:4E:6B Dennis
7 Add LUN masking - Host Port 10:00:00:61:68:00:4E:02 of Host UP2001 to access LUN 140697 of Controller 50:05:07:63:00:C0:95:DF of Storage IBMESS_013-22599 Dennis
8 Connect port 10:00:...02 of Host UP2001 to port 20:0B...CD of Switch USCG_D1_SW2 Dennis
9 Authorize paths from host UP2001 to IBMESS_013-22599 LUNs 140697 Dennis
Future Changes (11)
All All
Actions Event
9 +Violation resolved between Host UP2001 and Storage IBMESS 013-22599
8 Number of ports in Host UP2001 that can access LUN 140697 of Storage IBMESS 013-22599 changed from 1 to 2
+Link added between Host UP2001 and Switch USCG D1 SW2
7 +Storage IBMESS 013-22599 configuration changed
7 +Fabric 10:00:00:60:69:51:4E:6B configuration changed
Inventory
Admin
Task actions Future Violations Errors Future Changes

FIG. 8

FIG. 9


Topology of action 5
UP2001
UP3006
UP3004
UP3002
UN409
USCG_D1_SW2
USCG_D2_SW1
TAPES
IBMESS_013-22599
IBMESS_013-22546
UN417
UP3005
UP3003
UP3001
USCG_D2_SW3
USCG_D1_SW4

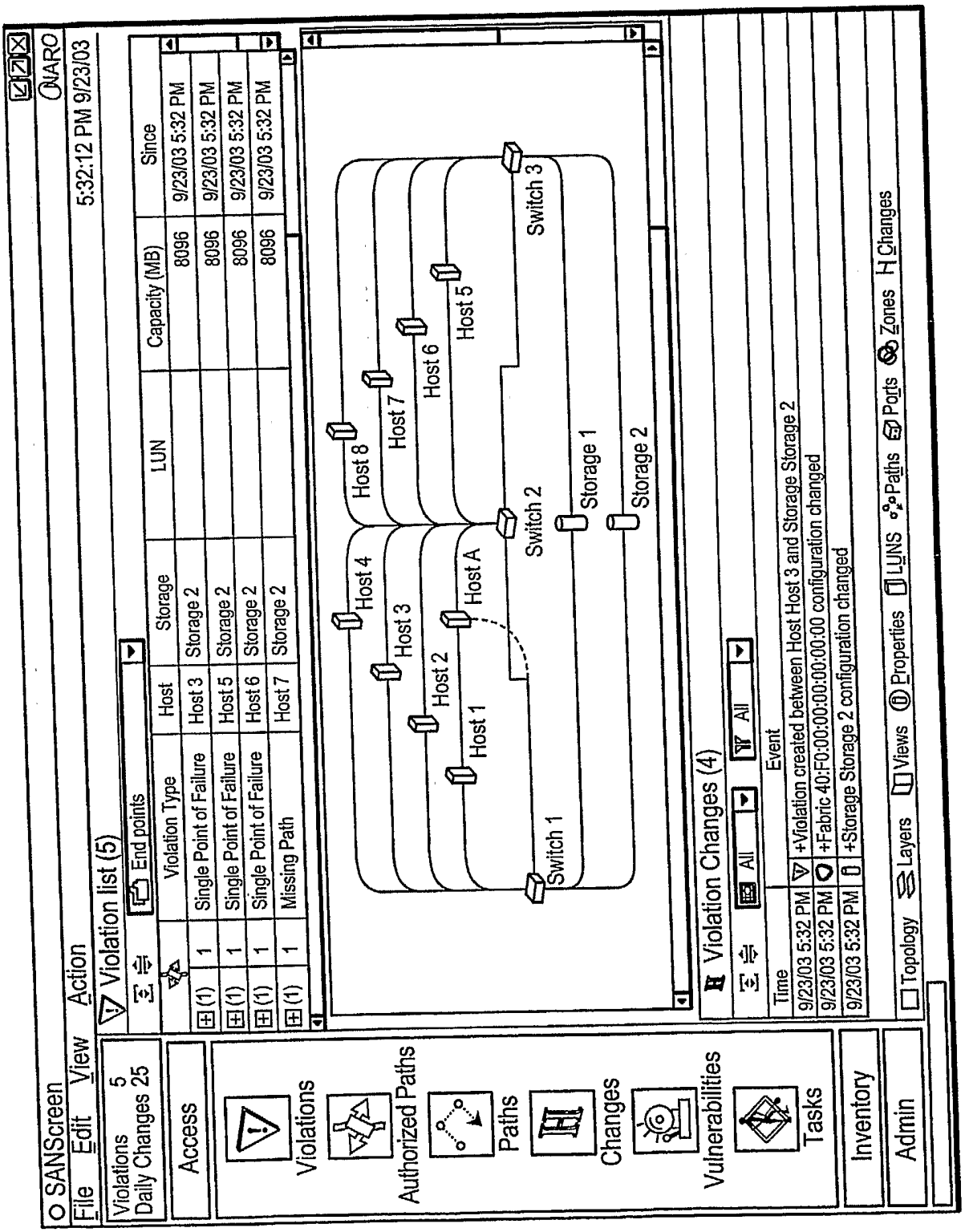

SANScreen
File Edit View Action
Violations 5
Daily Changes 25
Violation list (5)
End points
5:32:12 PM 9/23/03
Violation Type
Host
Storage
LUN
Capacity (MB)
Since
Single Point of Failure
Host 3
Storage 2
8096
9/23/03 5:32 PM
Single Point of Failure
Host 5
Storage 2
8096
9/23/03 5:32 PM
Single Point of Failure
Host 6
Storage 2
8096
9/23/03 5:32 PM
Missing Path
Host 7
Storage 2
8096
9/23/03 5:32 PM
Host 8
Host 4
Host 7
Host 3
Host 6
Host 2
Host 5
Host 1
Host A
Switch 3
Switch 2
Switch 1
Storage 1
Storage 2
Violation Changes (4)
All
All
Time
Event
9/23/03 5:32 PM
+Violation created between Host Host 3 and Storage Storage 2
9/23/03 5:32 PM
+Fabric 40:F0:00:00:00:00:00:00 configuration changed
9/23/03 5:32 PM
+Storage Storage 2 configuration changed
Topology
Layers
Views
Properties
LUNS
Paths
Ports
Zones
Changes
Access
Violations
Authorized Paths
Paths
Changes
Vulnerabilities
Tasks
Inventory
Admin

FIG. 10

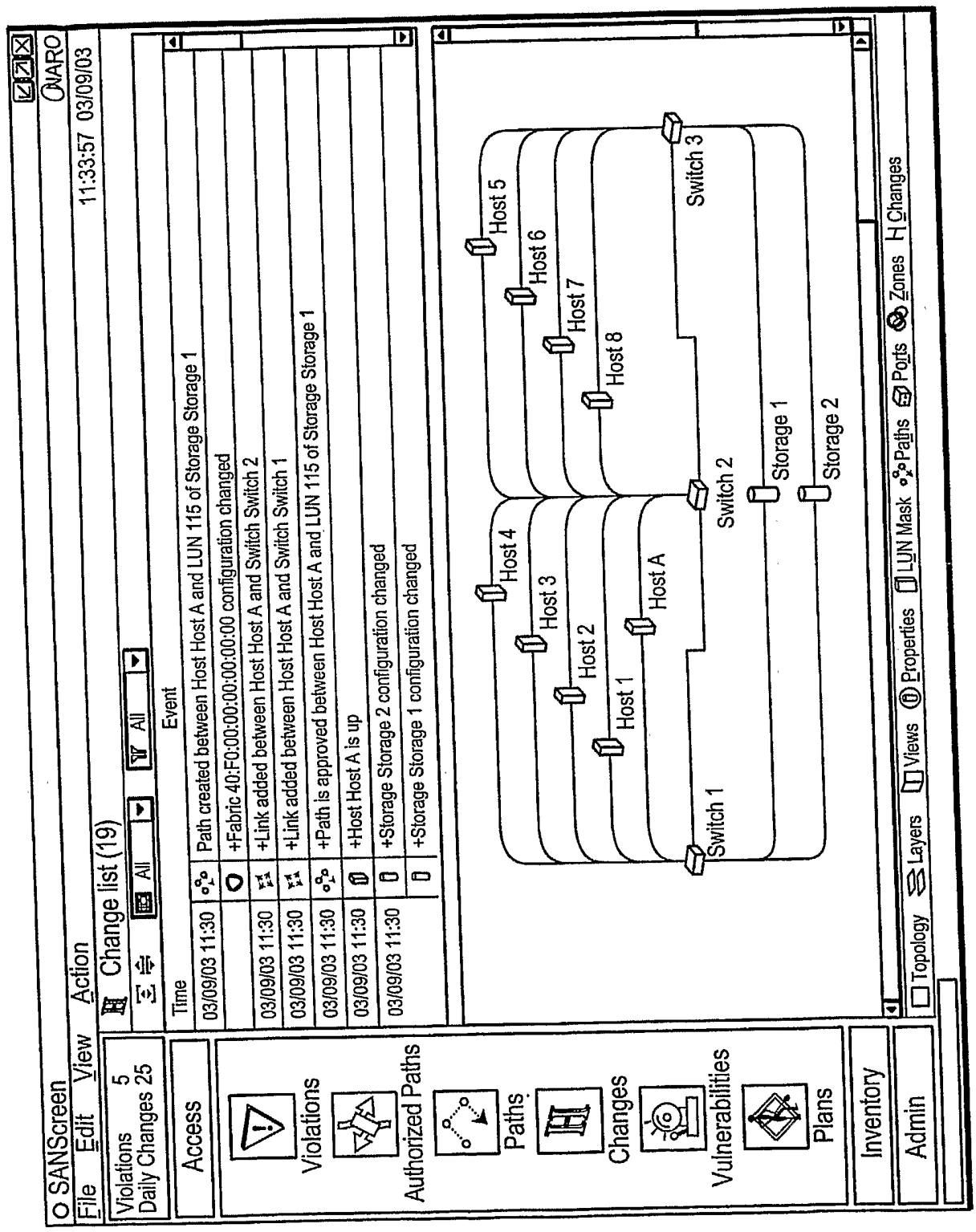

SANScreen
File Edit View Action
NARO
Violations 5
Daily Changes 25
Change list (19)
11:33:57 03/09/03
All
All
Access
Time
Event
03/09/03 11:30 Path created between Host Host A and LUN 115 of Storage Storage 1
+Fabric 40:F0:00:00:00:00:00:00 configuration changed
03/09/03 11:30 +Link added between Host Host A and Switch Switch 2
03/09/03 11:30 +Link added between Host Host A and Switch Switch 1
03/09/03 11:30 +Path is approved between Host Host A and LUN 115 of Storage Storage 1
03/09/03 11:30 +Host Host A is up
03/09/03 11:30 +Storage Storage 2 configuration changed
03/09/03 11:30 +Storage Storage 1 configuration changed
Violations
Authorized Paths
Paths
Changes
Vulnerabilities
Plans
Inventory
Admin
Host 5
Host 6
Host 7
Host 8
Switch 3
Host 4
Host 3
Host 2
Host 1
Host A
Switch 2
Switch 1
Storage 1
Storage 2
Topology
Layers
Views
Properties
LUN Mask
Paths
Ports
Zones
Changes

FIG. 11