



(51) International Patent Classification:

G06F 13/14 (2006.01) G06F 11/10 (2006.01)
G06F 12/16 (2006.01) G06F 3/06 (2006.01)

(21) International Application Number:

PCT/US2012/030067

(22) International Filing Date:

22 March 2012 (22.03.2012)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/467,715 25 March 2011 (25.03.2011) US

(71) Applicant (for all designated States except US): **ADC TELECOMMUNICATIONS, INC.** [US/US]; 1187 Park Place, Shakopee, Minnesota 55379 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **ANNE, Laxman R.** [IN/US]; 13545 Carmody Drive, Eden Prairie, Minnesota

55344 (US). **MILLER, Jeffrey J.** [US/US]; 2308 Promise Ave., Shakopee, Minnesota 55379 (US).

(74) Agent: **POWERS, Jon M.**; Fogg & Powers LLC, 5810 W. 78th Street, Suite 100, Minneapolis, Minnesota 55439 (US).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU,

[Continued on next page]

(54) Title: DOUBLE-BUFFER INSERTION COUNT STORED IN A DEVICE ATTACHED TO A PHYSICAL LAYER MEDIUM

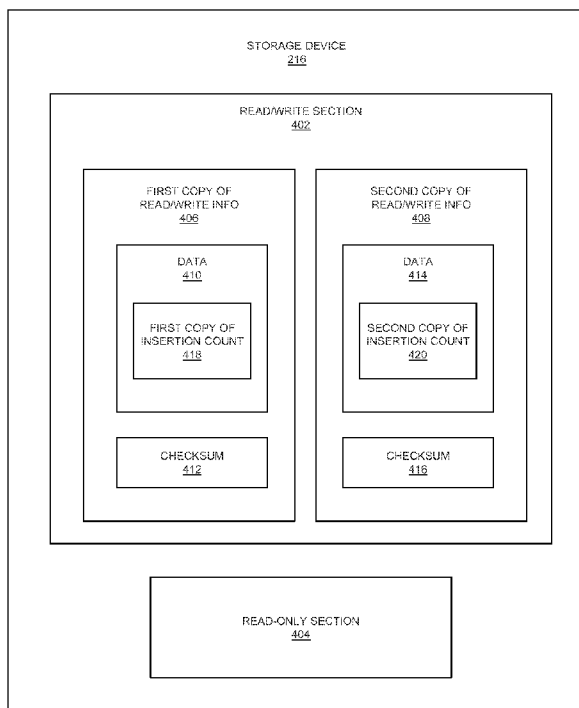


FIG. 4

(57) Abstract: One exemplary embodiment is directed to a connector assembly. The connector assembly comprises a port having a media interface configured to interface with a storage device interface of a connector. The connector is attached to a segment of physical communication media. The connector also includes a storage device. The connector assembly also comprises a programmable processor configured to execute software that stores information to the storage device using a plurality of redundant storage operations by which a plurality of copies of the information is sequentially stored in the storage device.



TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, 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 (Rule 48.2(g))*

DOUBLE-BUFFER INSERTION COUNT STORED IN
A DEVICE ATTACHED TO A PHYSICAL LAYER MEDIUM

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Patent Application Serial No. 61/467,715, filed on March 25, 2011, which is hereby incorporated herein by reference.

[0002] This application is related to the following:

[0003] U.S. Provisional Patent Application Serial No. 61/467,725, filed on March 25, 2011, titled "DYNAMICALLY DETECTING A DEFECTIVE CONNECTOR AT A PORT", which is hereby incorporated herein by reference;

[0004] U.S. Patent Application Serial No. __/__,__, Attorney Docket No. 100.1177US01, filed on even date herewith, titled "DYNAMICALLY DETECTING A DEFECTIVE CONNECTOR AT A PORT", which is hereby incorporated herein by reference;

[0005] U.S. Provisional Patent Application Serial No. 61/467,729, filed on March 25, 2011, titled "IDENTIFIER ENCODING SCHEME FOR USE WITH MULTI-PATH CONNECTORS", which is hereby incorporated herein by reference;

[0006] U.S. Patent Application Serial No. __/__,__, Attorney Docket No. 100.1178US01, filed on even date herewith, titled "IDENTIFIER ENCODING SCHEME FOR USE WITH MULTI-PATH CONNECTORS", which is hereby incorporated herein by reference;

[0007] U.S. Provisional Patent Application Serial No. 61/467,736, filed on March 25, 2011, titled "SYSTEMS AND METHODS FOR UTILIZING VARIABLE LENGTH DATA FIELD STORAGE SCHEMES ON PHYSICAL COMMUNICATION MEDIA SEGMENTS", which is hereby incorporated herein by reference;

[0008] U.S. Patent Application Serial No. __/__,__, Attorney Docket No. 100.1179US01, filed on even date herewith, titled "SYSTEMS AND METHODS FOR UTILIZING VARIABLE LENGTH DATA FIELD STORAGE SCHEMES ON PHYSICAL COMMUNICATION MEDIA SEGMENTS", which is hereby incorporated herein by reference;

[0009] U.S. Provisional Patent Application Serial No. 61/467,743, filed on March 25, 2011, titled "EVENT-MONITORING IN A SYSTEM FOR AUTOMATICALLY OBTAINING AND MANAGING PHYSICAL LAYER INFORMATION USING A RELIABLE PACKET-BASED COMMUNICATION", which is hereby incorporated herein by reference; and

[0010] U.S. Patent Application Serial No. __/__,__, Attorney Docket No. 100.1181US01, filed on even date herewith, titled "EVENT-MONITORING IN A SYSTEM FOR

AUTOMATICALLY OBTAINING AND MANAGING PHYSICAL LAYER INFORMATION USING A RELIABLE PACKET-BASED COMMUNICATION”, which is hereby incorporated herein by reference.

BACKGROUND

[0011] Communication networks typically include numerous logical communication links between various items of equipment. Often a single logical communication link is implemented using several pieces of physical communication media. For example, a logical communication link between a computer and an inter-networking device such as a hub or router can be implemented as follows. A first cable connects the computer to a jack mounted in a wall. A second cable connects the wall-mounted jack to a port of a patch panel, and a third cable connects the inter-networking device to another port of a patch panel. A “patch cord” cross connects the two together. In other words, a single logical communication link is often implemented using several segments of physical communication media.

[0012] A network or enterprise management system (generally referred to here as a “network management system” or “NMS”) is typically aware of the logical communication links that exist in a network but typically does not have information about the specific physical layer media that are used to implement the logical communication links. Indeed, NMS systems typically do not have the ability to display or otherwise provide information about how logical communication links are implemented at the physical layer level.

[0013] Physical layer management (PLM) systems do exist. However, existing PLM systems are typically designed to facilitate the adding, changing, and removing of cross connections at a particular patch panel or a set of patch panels at a given location. Generally, such PLM systems include functionality to track what is connected to each port of a patch panel, trace connections that are made using a patch panel, and provide visual indications to a user at a patch panel. However, such PLM systems are typically “patch-panel” centric in that they are focused on helping a technician correctly add, change, or remove cross connections at a patch panel. Any “intelligence” included in or coupled to the patch panel is typically only designed to facilitate making accurate cross connections at the patch panel and troubleshooting related problems (for example, by detecting whether a patch cord is inserted into a given port and/or by determining which ports are coupled to one another using a patch cord).

[0014] Moreover, any information that such PLM systems collect is typically only used within the PLM systems. In other words, the collections of information that such PLM systems maintain are logical “islands” that are not used at the application-layer level by other systems. Though such PLM systems are sometimes connected to other networks (for example, connected to local area networks or the Internet), such network connections are typically only used to enable a user to remotely access the PLM systems. That is, a user remotely accesses the PLM-related application-layer functionality that resides in the PLM system itself using the external network connection but external systems or networks typically do not themselves include any application-layer functionality that makes use of any of the physical-layer-related information that resides in the PLM system.

SUMMARY

[0015] One exemplary embodiment is directed to a connector assembly. The connector assembly comprises a port having a media interface configured to interface with a storage device interface of a connector. The connector is attached to a segment of physical communication media. The connector also includes a storage device. The connector assembly also comprises a programmable processor configured to execute software that stores information to the storage device using a plurality of redundant storage operations by which a plurality of copies of the information is sequentially stored in the storage device.

DRAWINGS

[0016] FIG. 1 is a block diagram of one exemplary embodiment of a system that includes physical layer information (PLI) functionality as well as physical layer management (PLM) functionality.

[0017] FIG. 2 is a block diagram of one high-level embodiment of a port and media reading interface that are suitable for use in the system of FIG. 1.

[0018] FIGS. 3A-3B are diagrams illustrating exemplary embodiments of patch cords.

[0019] FIG. 4 is a block diagram of one embodiment of a storage device suitable for use in the patch cords of FIGS. 3A-3B.

[0020] FIG. 5 is a flow chart of one exemplary embodiment of a method of updating an insertion count on a storage device of a connector.

[0021] FIG. 6 is a flow chart of one exemplary embodiment of a method of reading current copies of the insertion count from a storage device of a connector.

[0022] FIG. 7 is a flow chart of one exemplary embodiment of a method of redundantly storing an insertion count to a storage device of a connector.

DETAILED DESCRIPTION

[0023] FIG. 1 is a block diagram of one embodiment of a system 100 that includes physical layer information (PLI) functionality as well as physical layer management (PLM) functionality. The system 100 comprises a plurality of connector assemblies 102, where each connector assembly 102 comprises one or more ports 104. In general, the connector assemblies 102 are used to attach segments of physical communication media to one another.

[0024] Each segment of physical communication media is attached to a respective port 104. Each port 104 is used to connect two or more segments of physical communication media to one another (for example, to implement a portion of a logical communication link). Examples of connector assemblies 102 include, for example, rack-mounted connector assemblies (such as patch panels, distribution units, and media converters for fiber and copper physical communication media), wall-mounted connector assemblies (such as boxes, jacks, outlets, and media converters for fiber and copper physical communication media), and inter-networking devices (such as switches, routers, hubs, repeaters, gateways, and access points).

[0025] At least some of the connector assemblies 102 are designed for use with segments of physical communication media that have identifier and attribute information stored in or on them. The identifier and attribute information is stored in or on the segment of physical communication media in a manner that enables the stored information, when the segment is attached to a port 104, to be read by a programmable processor 106 associated with the connector assembly 102. Examples of information that can be stored in or on a segment of physical communication media include, without limitation, an identifier that uniquely identifies that particular segment of physical communication media (similar to an ETHERNET Media Access Control (MAC) address but associated with the physical communication media and/or connector attached to the physical communication media), a part number, a plug or other connector type, a cable or fiber type and length, a serial number, a cable polarity, a date of manufacture, a manufacturing lot number, information about one or more visual attributes of physical communication media or a connector attached to the physical communication media

(such as information about the color or shape of the physical communication media or connector or an image of the physical communication media or connector), and other information used by an Enterprise Resource Planning (ERP) system or inventory control system. In other embodiments, alternate or additional data is stored in or on the media segments. For example, testing or media quality or performance information can be stored in or on the segment of physical communication media. The testing or media quality or performance information, for example, can be the results of testing that is performed when a particular segment of media is manufactured.

[0026] Also, as noted below, in some embodiments, the information stored in or on the segment of physical communication media can be updated. For example, the information stored in or on the segment of physical communication media can be updated to include the results of testing that is performed when a segment of physical media is installed or otherwise checked. In another example, such testing information is supplied to an aggregation point 120 and stored in a data store maintained by the aggregation point 120 (both of which are described below). In another example, the information stored in or on the segment of physical communication media includes a count of the number of times that a connector (not shown) attached to a segment of physical communication media has been inserted into port 104. In such an example, the count stored in or on the segment of physical communication media is updated each time the connector is inserted into port 104. This insertion count value can be used, for example, for warranty purposes (for example, to determine if the connector has been inserted more than the number of times specified in the warranty) or for security purposes (for example, to detect unauthorized insertions of the physical communication media).

[0027] In the particular embodiment shown in FIG. 1, each of the ports 104 of the connector assemblies 102 comprises a respective media interface 108 via which the respective programmable processor 106 is able to determine if a physical communication media segment is attached to that port 104 and, if one is, to read the identifier and attribute information stored in or on the attached segment (if such information is stored therein or thereon). The programmable processor 106 associated with each connector assembly 102 is communicatively coupled to each of the media interfaces 108 using a suitable bus or other interconnect (not shown).

[0028] In the particular embodiment shown in FIG. 1, four exemplary types of connector assembly configurations are shown. In the first connector assembly configuration 110 shown in FIG. 1, each connector assembly 102 includes its own respective programmable processor 106 and its own respective network interface 116 that is used

to communicatively couple that connector assembly 102 to an Internet Protocol (IP) network 118.

[0029] In the second type of connector assembly configuration 112, a group of connector assemblies 102 are physically located near each other (for example, in a bay or equipment closet). Each of the connector assemblies 102 in the group includes its own respective programmable processor 106. However, in the second connector assembly configuration 112, some of the connector assemblies 102 (referred to here as “interfaced connector assemblies”) include their own respective network interfaces 116 while some of the connector assemblies 102 (referred to here as “non-interfaced connector assemblies”) do not. The non-interfaced connector assemblies 102 are communicatively coupled to one or more of the interfaced connector assemblies 102 in the group via local connections. In this way, the non-interfaced connector assemblies 102 are communicatively coupled to the IP network 118 via the network interface 116 included in one or more of the interfaced connector assemblies 102 in the group. In the second type of connector assembly configuration 112, the total number of network interfaces 116 used to couple the connector assemblies 102 to the IP network 118 can be reduced. Moreover, in the particular embodiment shown in FIG. 1, the non-interfaced connector assemblies 102 are connected to the interfaced connector assembly 102 using a daisy chain topology (though other topologies can be used in other implementations and embodiments).

[0030] In the third type of connector assembly configuration 114, a group of connector assemblies 102 are physically located near each other (for example, within a bay or equipment closet). Some of the connector assemblies 102 in the group (also referred to here as “master” connector assemblies 102) include both their own programmable processors 106 and network interfaces 116, while some of the connector assemblies 102 (also referred to here as “slave” connector assemblies 102) do not include their own programmable processors 106 or network interfaces 116. Each of the slave connector assemblies 102 is communicatively coupled to one or more of the master connector assemblies 102 in the group via one or more local connections. The programmable processor 106 in each of the master connector assemblies 102 is able to carry out the processing described below for both the master connector assembly 102 of which it is a part and any slave connector assemblies 102 to which the master connector assembly 102 is connected via the local connections. As a result, the cost associated with the slave connector assemblies 102 can be reduced. In the particular embodiment shown in FIG. 1, the slave connector assemblies 102 are connected to a master connector assembly

102 in a star topology (though other topologies can be used in other implementations and embodiments).

[0031] Each programmable processor 106 is configured to execute software or firmware 190 (shown in FIG. 2) that causes the programmable processor 106 to carry out various functions described below. The software 190 comprises program instructions that are stored (or otherwise embodied) on an appropriate non-transitory storage medium or media 192 (such as flash or other non-volatile memory, magnetic disc drives, and/or optical disc drives). At least a portion of the program instructions are read from the storage medium 192 by the programmable processor 106 for execution thereby. The storage medium 192 on or in which the program instructions are embodied is also referred to here as a “program product”. Although the storage medium 192 is shown in FIG. 2 as being included in, and local to, the connector assembly 102, it is to be understood that remote storage media (for example, storage media that are accessible over a network or communication link) and/or removable media can also be used. Each connector assembly 102 also includes suitable memory (not shown) that is coupled to the programmable processor 106 for storing program instructions and data. In general, the programmable processor 106 (and the software 190 executing thereon) determines if a physical communication media segment is attached to a port 104 with which that processor 106 is associated and, if one is, to read the identifier and attribute information stored in or on the attached physical communication media segment (if the segment includes such information stored therein or thereon) using the associated media interface 108.

[0032] As shown in FIG. 1, in the first, second, and third configurations 110, 112, and 114, each programmable processor 106 is also configured to communicate physical layer information to devices that are coupled to the IP network 118. The physical layer information (PLI) includes information about the connector assemblies 102 associated with that programmable processor 106 (also referred to here as “device information”) as well as information about any segments of physical media attached to the ports 104 of those connector assemblies 102 (also referred to here as “media information”) The device information includes, for example, an identifier for each connector assembly, a type identifier that identifies the connector assembly’s type, and port priority information that associates a priority level with each port. The media information includes identity and attribute information that the programmable processor 106 has read from attached physical media segments that have identifier and attribute information stored in or on it. The media information may also include information about physical communication media that does not have identifier or attribute

information stored in or on it. This latter type of media information can be manually input at the time the associated physical media segments are attached to the connector assembly 102 (for example, using a management application executing on the programmable processor 106 that enables a user to configure and monitor the connector assembly 102).

[0033] In the fourth type of connector assembly configuration 115, a group of connector assemblies 102 are housed within a common chassis or other enclosure. Each of the connector assemblies 102 in the configuration 115 includes their own programmable processors 106. In the context of this configuration 115, the programmable processors 106 in each of the connector assemblies are “slave” processors 106. Each of the slave programmable processor 106 is also communicatively coupled to a common “master” programmable processor 117 (for example, over a backplane included in the chassis or enclosure). The master programmable processor 117 is coupled to a network interface 116 that is used to communicatively couple the master programmable processor 117 to the IP network 118. In this configuration 115, each slave programmable processor 106 is configured to determine if physical communication media segments are attached to its port 104 and to read the identifier and attribute information stored in or on the attached physical communication media segments (if the attached segments have such information stored therein or thereon) using the associated media interfaces 108. This information is communicated from the slave programmable processor 106 in each of the connector assemblies 102 in the chassis to the master processor 117. The master processor 117 is configured to handle the processing associated with communicating the physical layer information read from by the slave processors 106 to devices that are coupled to the IP network 118.

[0034] The system 100 includes functionality that enables the physical layer information that the connector assemblies 102 capture to be used by application-layer functionality outside of the traditional physical-layer management application domain. That is, the physical layer information is not retained in a PLM “island” used only for PLM purposes but is instead made available to other applications. In the particular embodiment shown in FIG. 1, the system 100 includes an aggregation point 120 that is communicatively coupled to the connector assemblies 102 via the IP network 118.

[0035] The aggregation point 120 includes functionality that obtains physical layer information from the connector assemblies 102 (and other devices) and stores the physical layer information in a data store.

[0036] The aggregation point 120 can be used to receive physical layer information from various types of connector assemblies 106 that have functionality for automatically

reading information stored in or on the segment of physical communication media. Examples of such connector assemblies 106 are noted above. Also, the aggregation point 120 can also be used to receive physical layer information from other types of devices that have functionality for automatically reading information stored in or on the segment of physical communication media. Examples of such devices include end-user devices – such as computers, peripherals (such as printers, copiers, storage devices, and scanners), and IP telephones – that include functionality for automatically reading information stored in or on the segment of physical communication media.

[0037] The aggregation point 120 can also be used to obtain other types of physical layer information. For example, in this embodiment, the aggregation point 120 also obtains information about physical communication media segments that is not otherwise automatically communicated to an aggregation point 120. One example of such information is information about non-connectorized physical communication media segments that do not otherwise have information stored in or on them that are attached to a connector assembly (including, for example, information indicating which ports of the devices are connected to which ports of other devices in the network as well as media information about the segment). Another example of such information is information about physical communication media segments that are connected to devices that are not able to read media information that is stored in or on the media segments that are attached to their ports and/or that are not able to communicate such information to the aggregation point 120 (for example, because such devices do not include such functionality, because such devices are used with media segments that do not have media information stored in or on them, and/or because bandwidth is not available for communicating such information to the aggregation point 120). In this example, the information can include, for example, information about the devices themselves (such as the devices' MAC addresses and IP addresses if assigned to such devices), information indicating which ports of the devices are connected to which ports of other devices in the network (for example, other connector assemblies), and information about the physical media attached to the ports of the devices. This information can be provided to the aggregation point 120, for example, by manually entering such information into a file (such as a spreadsheet) and then uploading the file to the aggregation point 120 (for example, using a web browser) in connection with the initial installation of each of the various items. Such information can also, for example, be directly entered using a user interface provided by the aggregation point 120 (for example, using a web browser).

[0038] The aggregation point 120 can also obtain information about the layout of the building or buildings in which the network is deployed, as well as information indicating where each connector assembly 102, physical media segment, and inter-networking device is located within the building. This information can be, for example, manually entered and verified (for example, using a web browser) in connection with the initial installation of each of the various items. In one implementation, such location information includes an X, Y, and Z location for each port or other termination point for each physical communication media segment (for example, X, Y, and Z location information of the type specified in the ANSI/TIA/EIA 606-A Standard (Administration Standard For The Commercial Telecommunications Infrastructure)).

[0039] The aggregation point 120 can obtain and maintain testing, media quality, or performance information relating to the various segments of physical communication media that exist in the network. The testing, media quality, or performance information, for example, can be results of testing that is performed when a particular segment of media is manufactured and/or when testing is performed when a particular segment of media is installed or otherwise checked.

[0040] The aggregation point 120 also includes functionality that provides an interface for external devices or entities to access the physical layer information maintained by the aggregation point 120. This access can include retrieving information from the aggregation point 120 as well as supplying information to the aggregation point 120. In this embodiment, the aggregation point 120 is implemented as “middleware” that is able to provide such external devices and entities with transparent and convenient access to the PLI maintained by the access point 120. Because the aggregation point 120 aggregates PLI from the relevant devices on the IP network 118 and provides external devices and entities with access to such PLI, the external devices and entities do not need to individually interact with all of the devices in the IP network 118 that provide PLI, nor do such devices need to have the capacity to respond to requests from such external devices and entities.

[0041] The aggregation point 120, in the embodiment shown in FIG. 1, implements an application programming interface (API) by which application-layer functionality can gain access to the physical layer information maintained by the aggregation point 120 using a software development kit (SDK) that describes and documents the API. Also, in those embodiments where the connector assemblies 102 include one or more light emitting diodes (LEDs) (for example, where each port 104 has an associated LED), the API and aggregation point 120 can include functionality that enables application-layer functionality to change the state of such LEDs using the API.

[0042] For example, as shown in FIG. 1, a network management system (NMS) 130 includes physical layer information (PLI) functionality 132 that is configured to retrieve physical layer information from the aggregation point 120 and provide it to the other parts of the NMS 130 for use thereby. The NMS 130 uses the retrieved physical layer information to perform one or more network management functions (for example, as described below). In one implementation of the embodiment shown in FIG. 1, the PLI functionality 132 of the NMS 130 retrieves physical layer information from the aggregation point 120 using the API implemented by the aggregation point 120. The NMS 130 communicates with the aggregation point 120 over the IP network 118.

[0043] As shown in FIG. 1, an application 134 executing on a computer 136 can also use the API implemented by the aggregation point 120 to access the PLI information maintained by the aggregation point 120 (for example, to retrieve such information from the aggregation point 120 and/or to supply such information to the aggregation point 120). The computer 136 is coupled to the IP network 118 and accesses the aggregation point 120 over the IP network 118.

[0044] In the embodiment shown in FIG. 1, one or more inter-networking devices 138 used to implement the IP network 118 include physical layer information (PLI) functionality 140. The PLI functionality 140 of the inter-networking device 138 is configured to retrieve physical layer information from the aggregation point 120 and use the retrieved physical layer information to perform one or more inter-networking functions. Examples of inter-networking functions include Layer 1, Layer 2, and Layer 3 (of the OSI model) inter-networking functions such as the routing, switching, repeating, bridging, and grooming of communication traffic that is received at the inter-networking device. In one implementation of such an embodiment, the PLI functionality 140 uses the API implemented by the aggregation point 120 to communicate with the aggregation point 120.

[0045] The PLI functionality 140 included in the inter-networking device 138 can also be used to capture physical layer information associated with the inter-network device 138 and the physical communication media attached to it and communicate the captured physical layer information to the aggregation point 120. Such information can be provided to the aggregation point 120 using the API or by using the protocols that are used to communicate with the connector assemblies 102.

[0046] The aggregation point 120 can be implemented on a standalone network node (for example, a standalone computer running appropriate software) or can be integrated along with other network functionality (for example, integrated with an element management system or network management system or other network server

or network element). Moreover, the functionality of the aggregation point 120 can be distributed across many nodes and devices in the network and/or implemented, for example, in a hierarchical manner (for example, with many levels of aggregation points).

[0047] Moreover, the aggregation point 120 and the connector assemblies 102 are configured so that the aggregation point 120 can automatically discover and connect with devices that provide PLI to an aggregation point 120 (such as the connector assemblies 102 and inter-network device 138) that are on the network 118. In this way, when devices that are able to provide PLI to an aggregation point 120 (such as a connector assembly 102 or an inter-networking device 138) are coupled to the IP network 118, an aggregation point 120 is able to automatically discover the connector assembly 102 and start aggregating physical layer information for that connector assembly 102 without requiring the person installing the connector assembly 102 to have knowledge of the aggregation points 120 that are on the IP network 118. Similarly, when an aggregation point 120 is coupled to the IP network 118, the aggregation point 120 is able to automatically discover and interact with devices that are capable of providing PLI to an aggregation point without requiring the person installing the aggregation point 120 to have knowledge of the devices that are on the IP network 118. Thus, the physical-layer information resources described here can be easily integrated into the IP network 118.

[0048] The IP network 118 can include one or more local area networks and/or wide area networks (including for example the Internet). As a result, the aggregation point 120, NMS 130, and computer 136 need not be located at the same site as each other or at the same site as the connector assemblies 102 or the inter-networking devices 138.

[0049] Various conventional IP networking techniques can be used in deploying the system 100 of FIG. 1. For example, conventional security protocols can be used to secure communications if they are communicated over a public or otherwise unsecure communication channel (such as the Internet or over a wireless communication link).

[0050] In one implementation of the embodiment shown in FIG. 1, each connector assembly 102, each port 104 of each connector assembly 102, and each media segment is individually addressable. Where IP addresses are used to individually address each connector assembly 102, a virtual private network (VPN) dedicated for use with the various connector assemblies 102 can be used to segregate the IP addresses used for the connector assemblies 102 from the main IP address space that is used in the IP network 118.

[0051] Also, power can be supplied to the connector assemblies 102 using conventional “Power over Ethernet” techniques specified in the IEEE 802.3af standard, which is hereby incorporated herein by reference. In such an implementation, a power hub 142 or other power supplying device (located near or incorporated into an inter-networking device that is coupled to each connector assembly 102) injects DC power onto one or more of the wires (also referred to here as the “power wires”) included in the copper twisted-pair cable used to connect each connector assembly 102 to the associated inter-networking device. The interface 116 in the connector assembly 102 picks the injected DC power off of the power wires and uses the picked-off power to power the active components of that connector assembly 102. In the second and third connector assembly configurations 112 and 114, some of the connector assemblies 102 are not directly connected to the IP network 118 and, therefore, are unable to receive power directly from the power wires. These connector assemblies 102 receive power from the connector assemblies 102 that are directly connected to the IP network 118 via the local connections that communicatively couple such connector assemblies 102 to one another. In the fourth configuration 115, the interface 116 picks the injected DC power off of the power wires and supplies power to the master processor 117 and each of the slave processors 106 over the backplane.

[0052] In the particular embodiment shown in FIG. 1, the system 100 also supports conventional physical layer management (PLM) operations such as the tracking of moves, adds, and changes of the segments of physical media that are attached to the ports 104 of the connector assemblies 102 and providing assistance with carrying out moves, adds, and changes. PLI provided by the aggregation point 120 can be used to improve upon conventional “guided MAC” processes. For example, information about the location of the port 104 and the visual appearance (for example, the color or shape) of the relevant physical media segment (or connector attached thereto) can be communicated to a technician to assist the technician in carrying out a move, add, or change. This information can be communicated to a computer or smartphone used by the technician. Moreover, the PLI functionality that resides in the system 100 can also be used to verify that a particular MAC was properly carried out by checking that the expected physical media segment is located in the expected port 104. If that is not the case, an alert can be sent to the technician so that the technician can correct the issue.

[0053] The PLM functionality included in the system 100 can also support conventional techniques for guiding the technician in carrying out a MAC (for example, by illuminating one or more light emitting diodes (LEDs) to direct a technician to a particular connector assembly 102 and/or to a particular port 104 or by displaying

messages on a liquid crystal display (LCD) included on or near the connector assemblies 102).

[0054] Other PLM functions include keeping historical logs about the media connected to the connector assembly. In the embodiment shown in FIG. 1, the aggregation point 120 includes PLM functionality 144 that implements such PLM functions. The PLM functionality 144 does this using the physical layer information that is maintained at the aggregation point 120.

[0055] The IP network 118 is typically implemented using one or more inter-networking devices. As noted above, an inter-networking device is a type of connector assembly (and a particular implementation of an inter-networking device 138 is referenced separately in FIG.1 for ease of explanation only). Generally, an inter-networking device can be configured to read media information that is stored in or on the segments of physical media that are attached to its ports and to communicate the media information it reads from the attached segments of media (as well as information about the inter-networking device itself) to an aggregation point 120 like any other connector assembly described here.

[0056] In addition to connector assemblies 102, the techniques described here for reading media information stored in or on a segment of physical communication media can be used in one or more end nodes of the IP network 118. For example, computers (such as, laptops, servers, desktop computers, or special-purpose computing devices such as IP telephones, IP multi-media appliances, and storage devices) can be configured to read media information that is stored in or on the segments of physical communication media that are attached to their ports and to communicate the media information they read from the attached segments of media (as well as information about the devices themselves) to an aggregation point 120 as described here.

[0057] In one implementation of the system 100 shown in FIG. 1, the ports 104 of each connector assembly 102 are used to implement the IP network 118 over which each connector assembly 102 communicates physical layer information associated with that connector assembly 102. In such an implementation, such physical layer information is communicated over the IP network 118 just like any other data that is communicated over the IP network 118. As noted below, the media interface 108 determines if a physical communication media segment is attached to the corresponding port 104 and, if one is, reads the identifier and attribute information stored in or on the attached segment (if such information is stored therein or thereon) without affecting the normal data signals that pass through that port 104. Indeed, such physical layer information may actually pass through one or more of the ports 104 of connector assemblies 102 in

the course of being communicated to and/or from a connector assembly 102, aggregation point 150, network management system 130, and/or computer 136. By using the IP network 118 to communicate physical layer information pertaining to it, a separate network need not be provided and maintained in order to communicate such physical-layer information. However, in other implementations and embodiments, the physical layer information described above is communicated using a network that is separate from the network to which such physical layer information pertains.

[0058] FIG. 2 is a block diagram of one high-level embodiment of a port 104 and media interface 108 that are suitable for use in the system 100 of FIG. 1.

[0059] Each port 104 comprises a first attachment point 206 and a second attachment point 208. The first attachment point 206 is used to attach a first segment of physical communication media 210 to the port 104, and the second attachment point 208 is used to attach a second segment of physical communication media 212 to the port 104.

[0060] In the particular embodiment shown in FIG. 2, the first attachment point 206 is located near the rear of the connector assembly. As a consequence, the first attachment point 206 and the first segment of physical media 210 attached thereto are also referred to here as the “rear attachment point” 206 and the “rear media segment” 210, respectively. Also, in this embodiment, the rear attachment point 206 is configured to attach the rear media segment 210 to the port 104 in a semi-permanent manner. As used herein, a semi-permanent attachment is one that is designed to be changed relatively infrequently, if ever. This is also referred to sometimes as a “one-time” connection. Examples of suitable rear connectors 206 include punch-down blocks (in the case of copper physical media) and fiber adapters, fiber splice points, and fiber termination points (in the case of optical physical media).

[0061] In the embodiment shown in FIG. 2, the second attachment point 208 is located near the front of the connector assembly 102. As a consequence, the second attachment point 208 and the second segment of physical media 212 are also referred to here as the “front attachment point” 208 and the “front media segment” 212, respectively. In the embodiment shown in FIG. 2, the front attachment point 208 for each port 104 is designed for use with “connectorized” front media segments 212 that have identifier and attribute information stored in or on them. As used herein, a “connectorized” media segment is a segment of physical communication media that includes a connector 214 at at least one end of the segment. The front attachment point 208 is implemented using a suitable connector or adapter that mates with the corresponding connector 214 on the end of the front media segment 212. The connector 214 is used to facilitate the easy and repeated attachment and unattachment of the front media segment 212 to the port 104.

Examples of connectorized media segments include CAT-5, 6, and 7 twisted-pair cables having modular connectors or plugs attached to both ends (in which case, the front connectors are implemented using compatible modular jacks) or optical cables having SC, LC, FC, LX.5, MTP, or MPO connectors (in which case, the front connectors are implemented using compatible SC, LC, FC, LX.5, MTP, or MPO connectors or adapters). The techniques described here can be used with other types of connectors including, for example, BNC connectors, F connectors, DSX jacks and plugs, bantam jacks and plugs, and MPO and MTP multi-fiber connectors and adapters.

[0062] Each port 104 communicatively couples the respective rear attachment point 206 to the respective front attachment point 208. As a result, a rear media segment 210 attached to the respective rear attachment point 206 is communicatively coupled to any front media segment 212 attached to the respective front attachment point 208. In one implementation, each port 104 is designed for use with a rear media segment 210 and a front media segment 212 that comprise the same type of physical communication media, in which case each port 104 communicatively couples any rear media segment 210 attached to the respective rear attachment point 206 to any front media segment 212 attached to the respective front attachment point 208 at the physical layer level without any media conversion. In other implementations, each port 104 communicatively couples any rear media segment 210 attached to the respective rear attachment point 206 to any front media segment 212 attached to the respective front attachment point 208 in other ways (for example, using a media converter if the rear media segment 210 and the front media segment 212 comprise different types of physical communication media).

[0063] In the exemplary embodiment shown in FIG. 2, the port 104 is configured for use with front media segments 212 that include a storage device 216 in which the media information for that media segment 212 is stored. The storage device 216 includes a storage device interface 218 that, when the corresponding connector 214 is inserted into (or otherwise attached to) a front attachment point 208 of the port 104, communicatively couples the storage device 216 to a corresponding media interface 108 so that the associated programmable processor 106 can read the information stored in the storage device 216. In one implementation of the embodiment shown in FIG. 2, each connector 214 itself houses the storage device 216. In another implementation of such an embodiment, the storage device 216 is housed within a housing that is separate from the connector 214. In such an implementation, the housing is configured so that it can be snapped onto the media segment 212 or the connector 214, with the storage device interface 218 positioned relative to the connector 214 so that the storage device

interface 218 will properly mate with the media interface 108 when the connector 214 is inserted into (or otherwise attached to) the front attachment point 208. Although in the exemplary embodiment shown in FIG. 2 only the front media segments 212 include storage devices 216, it is to be understood that in other embodiments connector assemblies and/or other devices are configured to read storage devices that are attached to (or otherwise included with) rear media segments 210 and/or any "auxiliary" media segments (for example, media segments coupled to the network interface 116).

[0064] In some implementations, at least some of the information stored in the storage device 216 can be updated in the field (for example, by having an associated programmable processor 106 cause additional information to be written to the storage device 216 or changing or deleting information that was previously stored in the storage device 216). For example, in some implementations, some of the information stored in the storage device 216 cannot be changed in the field (for example, identifier information or manufacturing information) while some of the other information stored in the storage device 216 can be changed in the field (for example, testing, media quality, or performance information). In other implementations, none of the information stored in the storage device 216 can be updated in the field.

[0065] Also, the storage device 216 may also include a processor or micro-controller, in addition to storage for the media information. In which case, the micro-controller included in the storage device 216 can be used to execute software or firmware that, for example, controls one or more LEDs attached to the storage device 216. In another example, the micro-controller executes software or firmware that performs an integrity test on the front media segment 212 (for example, by performing a capacitance or impedance test on the sheathing or insulator that surrounds the front physical communication media segment 212, (which may include a metallic foil or metallic filler for such purposes)). In the event that a problem with the integrity of the front media segment 212 is detected, the micro-controller can communicate that fact to the programmable processor 106 associated with the port 104 using the storage device interface 218. The micro-controller can also be used for other functions.

[0066] The port 104, connector 214, storage device 216, and media interface 108 are configured so that the information stored in the storage device 216 can be read without affecting the communication signals that pass through the media segments 210 and 212.

[0067] Further details regarding system 100 and the port 104 can be found in the following United States Patent Applications, all of which are hereby incorporated herein by reference: United States Provisional Patent Application Serial No. 61/152,624, filed

on February 13, 2009, titled "MANAGED CONNECTIVITY SYSTEMS AND METHODS" (also referred to here as the "624 Application"); United States Patent Application Serial No. 12/705,497, filed on February 12, 2010, titled "AGGREGATION OF PHYSICAL LAYER INFORMATION RELATED TO A NETWORK" (is also referred to here as the '497 Application); United States Patent Application Serial No. 12/705,501, filed on February 12, 2010, titled "INTER-NETWORKING DEVICES FOR USE WITH PHYSICAL LAYER INFORMATION" (also referred to here as the '501 Application); United States Patent Application Serial No. 12/705,506, filed on February 12, 2010, titled "NETWORK MANAGEMENT SYSTEMS FOR USE WITH PHYSICAL LAYER INFORMATION" (also referred to here as the '506 Application); United States Patent Application Serial No. 12/705,514, filed on February 12, 2010, titled "MANAGED CONNECTIVITY DEVICES, SYSTEMS, AND METHODS" (also referred to here as the '514 Application); United States Provisional Patent Application Serial No. 61/252,395, filed on October 16, 2009, titled "MANAGED CONNECTIVITY IN ELECTRICAL SYSTEMS AND METHODS THEREOF" (also referred to here as the "395 Application"); United States Provisional Patent Application Serial No. 61/253,208, filed on October 20, 2009, titled "ELECTRICAL PLUG FOR MANAGED CONNECTIVITY SYSTEMS" (also referred to here as the "208 Application"); United States Provisional Patent Application Serial No. 61/252,964, filed on October 19, 2009, titled "ELECTRICAL PLUG FOR MANAGED CONNECTIVITY SYSTEMS" (also referred to here as the "964 Application"); United States Provisional Patent Application Serial No. 61/252,386, filed on October 16, 2009, titled "MANAGED CONNECTIVITY IN FIBER OPTIC SYSTEMS AND METHODS THEREOF" (also referred to here as the "386 Application"); United States Provisional Patent Application Serial No. 61/303,961, filed on February 12, 2010, titled "FIBER PLUGS AND ADAPTERS FOR MANAGED CONNECTIVITY" (the "961 Application"); and United States Provisional Patent Application Serial No. 61/303,948, filed on February 12, 2010, titled "BLADED COMMUNICATIONS SYSTEM" (the "948 Application").

[0068] FIG. 3A is a diagram illustrating one exemplary embodiment of a front media segment. In the embodiment shown in FIG. 3A, the front media segment comprises a "patch cord" 312 that is used to selectively cross-connect two ports of the same or different patch panels. The patch cord 312 shown in FIG. 3A is suitable for use with an implementation of a patch panel where the front connectors of the ports are implemented using modular RJ-45 jacks. The patch cord 312 shown in FIG. 3A comprises a copper unshielded twisted-pair (UTP) cable 386. The UTP cable 386 includes eight conductors arranged in four conductor pairs. The patch cord 312 also comprises two RJ-45 plugs 314, one at each end of the cable 386 (only one of which is shown in FIG. 3A). The RJ-45 plugs 314 are designed to be inserted into the RJ-45

modular jacks used as the front connectors. Each RJ-45 plug 314 comprises a contact portion 388 in which eight, generally parallel electrical contacts 390 are positioned. Each of the eight electrical contacts 390 are electrically connected to one of the eight conductors in the UTP cable 386.

[0069] Each plug 314 also comprises (or is attached to) a storage device 392 (for example, an Electrically Erasable Programmable Read-Only Memory (EEPROM) or other non-volatile memory device). The media information described above for the patch cord 312 is stored in the storage device 392. The storage device 392 includes sufficient storage capacity to store such information. Each storage device 392 also includes a storage device interface 394 that, when the corresponding plug 314 is inserted into a front connector of a port 304, communicatively couples the storage device 392 to the corresponding media reading interface so that the programmable processor 320 in the corresponding patch panel 302 can read the information stored in the storage device 392.

[0070] Examples of such a patch cord 312 and plug 314 are described in the '395 Application, the '208 Application, and the '964 Application.

[0071] FIG. 3B is a diagram illustrating another exemplary embodiment of a patch cord 312'. The patch cord 312' shown in FIG. 3B is suitable for use with a fiber patch panel where the front connectors of the ports are implemented using fiber LC adapters or connectors. The patch cord 312' shown in FIG. 3B comprises an optical cable 386'. The optical cable 386' includes an optical fiber enclosed within a suitable sheathing. The patch cord 312' also comprises two LC connectors 314', one at each of the cable 386'. Each LC connector 314' is designed to be inserted into an LC adapter used as the front connector of a port of a fiber patch panel. Each LC connector 314' comprises an end portion 388' at which an optical connection with the optical fiber in the cable 386' can be established when the LC connector 314' is inserted in an LC adapter of a port.

[0072] Each LC connector 314' also comprises (or is attached to) a storage device 392' (for example, an Electrically Erasable Programmable Read-Only Memory (EEPROM) or other non-volatile memory device). The media information described above for the patch cord 312 is stored in the storage device 392'. The storage device 392' includes sufficient storage capacity to store such information. Each storage device 392' also includes a storage device interface 394' that, when the corresponding LC connector 314' is inserted into a front connector of a port, communicatively couples the storage device 392' to the corresponding media reading interface so that the programmable processor in the corresponding fiber patch panel can read the information stored in the storage device 392'.

[0073] In some implementations of the patch cords 312 and 312', the storage devices 392 and 392' are implemented using a surface-mount EEPROM or other non-volatile memory device. In such implementations, the storage device interfaces and media reading interfaces each comprise four leads -- a power lead, a ground lead, a data lead, and an extra lead that is reserved for future use. In one such implementation, an EEPROM that supports serial protocol is used, where the serial protocol is used for communicating over the signal data lead. The four leads of the storage device interfaces come into electrical contact with four corresponding leads of the media reading interface when the corresponding plug or connector is inserted in the corresponding front connector of a port 304. Each storage device interface and media reading interface are arranged and configured so that they do not interfere with data communicated over the patch cord. In other embodiments, other types of interfaces are used. For example, in one such alternative embodiment, a two-line interface is used with a simple charge pump. In other embodiments, additional lines are provided (for example, for potential future applications).

[0074] Examples of such fiber patch cords 312' and connectors 314' are described in United States Provisional Patent Application Serial No. 61/252,386, filed on October 16, 2009, titled "MANAGED CONNECTIVITY IN FIBER OPTIC SYSTEMS AND METHODS THEREOF" (also referred to here as the "'386 Application"), United States Provisional Patent Application Serial No. 61/303,961, filed on February 12, 2010, titled "FIBER PLUGS AND ADAPTERS FOR MANAGED CONNECTIVITY" (the "'961 Application"), and United States Provisional Patent Application Serial No. 61/303,948, filed on February 12, 2010, titled "BLADED COMMUNICATIONS SYSTEM" (the "'948 Application"). The '386 Application, the '961 Application, and the '948 Application are hereby incorporated herein by reference.

[0075] In some implementations of the patch cords 312 and 312', each plug 314 or connector 314' itself houses the respective storage device and storage device interface. In implementations, each storage device and corresponding storage device interface are housed within a housing that is separate from the corresponding plug or connector. In such implementations, the housing is configured so that it can be snapped onto (or otherwise attached to) the cable or the plug or connector, with the storage device interface positioned relative to the plug or connector so that the storage device interface will properly mate with the relevant media reading interface when the plug or connector is inserted into the front connector of the corresponding port.

[0076] As described above, in some embodiments, the information that is stored in the storage device 216 can be updated when the associated connector 214 to which that

storage device 216 is attached is inserted into a port 104 of a connector assembly 102. However, it is possible for that connector 214 to be removed from the port 104 while a write operation is being performed to update the information stored in the storage device 216. This can cause the data that was stored in the storage device 214 to become corrupted. FIGS. 4-7 describe one approach to addressing this issue.

[0077] For ease of explanation, certain processing relating to one or more connector assemblies 102 is described below in connection with FIGS. 4-7 as being performed by the programmable processor 106 and the software 190 executing on programmable processor 106. However, it is to be understood that all or part of the processing described below as being performed by processor 106 and the software 190 could also be performed by other processors and software associated with each connector assembly 102. For example, all or some of such processing can (but need not) be performed by a “master” processor 117 (and the software executing thereon) where a master-slave configuration 115 is used. Also, a particular connector assembly 102 can also include more than one processor 106 (for example, where required by the port density of the connector assembly 102).

[0078] Moreover, functionality described here as being implemented in software executing on a programmable processor can be implemented in other ways. For example, such functionality can be implemented in hardware using discrete hardware, application-specific integrated circuits (ASICs), programmable devices (such as field-programmable gate arrays (FPGAs) or complex programmable logic devices (CPLDs)), and/or combinations of one or more of the foregoing, and/or combinations of one or more of the foregoing along with software executing on one or more programmable processors. For example, the detection of the insertion of a connector 214 into a port 104 of a connector assembly 102 and/or the reading of information from any storage device 216 attached to the connector 214 can be implemented in hardware (for example, using one or more programmable devices and/or an ASIC) in addition to or instead of being implemented as software.

[0079] FIG. 4 is a block diagram of one embodiment of a storage device 216 suitable for use with the connector 214 of FIG. 2, the plug 314 of FIG. 3A, and the plug 314' of FIG. 3B. In the exemplary embodiment shown in FIG. 4, the information that is stored in the storage device 216 includes both read only information and read/write information. In the embodiment shown in FIG. 4, the storage device 216 includes both a read/write section 402 and a read-only section 404. The read/write section 402 of the storage device 216 can be read, updated, overwritten, and erased and is used to store the read/write information. In contrast, the read only section 404 of the storage device 216

can be read, but is write-protected and cannot be updated, overwritten, or erased during normal operation. The read only section 404 is used to store the read only information. In other embodiments, the storage device 216 only includes a read/write section 402 and there is no read-only section 404. The storage device 216 may include an Electrically Erasable Programmable Read-Only Memory (EEPROM) or other non-volatile memory device as discussed above with reference to storage device 392 and 392'.

[0080] The read/write section 402 of the storage device 216 is configured to store two copies of the read/write information. As shown in FIG. 4, first and second copies 406 and 408, respectively, of the read/write information is stored in the read/write section 402 of the storage device 216. In the particular exemplary embodiment shown in FIG. 4, the first copy 406 of the read/write information comprises a data part 410 (also referred to here simply as "data" 410) that includes the read/write information itself and a checksum 412 that is generated from the data 410 in the first copy 406 of the read/write information. Also, the second copy 408 of the read/write information comprises a data part 414 (also referred to here simply as "data" 414) that includes the read/write information itself and a checksum 416 that is generated from the data 414 in the second copy 408 of the read/write information.

[0081] In this exemplary embodiment, the software 190 is configured to only allow write operations to update, overwrite, or erase one of the first copy 406 and the second copy 408 at a time. In addition, the software 190 is further configured so that both the first copy 406 and the second copy 408 cannot be updated, overwritten, or erased by a single write operation. These features ensure that if an error occurs during a write operation to one of the first copy 406 and the second copy 408, the other copy remains intact. In some example implementations of such an embodiment, the processor 106 reads and writes from and to the storage device 216 using a serial protocol, which by its nature ensures that the write operations occur sequentially and not in parallel. In some implementations, the storage device 216 itself (and/or the media interface 108) is configured to preclude simultaneous writes to the first copy 406 and the second copy 408.

[0082] The techniques described here in connection with FIGS. 4-7 can be used for updating any information that is stored in storage device 216. For example, as noted above, the information stored in the storage device 216 can include information indicative of the number of times that a particular connector 214 has been inserted into port 104 (also referred to there as an "insertion count"). As noted above, in such an example, the insertion count stored in storage device 216 is updated each time the connector 214 to which the storage device 216 is attached is inserted into a port 104.

This insertion count value can be used, for example, for warranty purposes (for example, to determine if the connector 214 has been inserted more than the number of times specified in the warranty) or for security purposes (for example, to detect unauthorized insertions of the physical communication media). As shown in FIG. 4, the insertion count can be stored in the read/write section 402 of the storage device 216. A first copy 418 of the insertion count is included in the data 410 stored in the first copy 406 of the read/write information stored in the storage device 216. Also, a second copy 420 of the insertion count is included in the data 414 stored in the second copy 408 of the read/write information stored in the storage device 216. The exemplary embodiments shown in FIGS. 4-7 are described here in connection with updating an insertion count, although the techniques described here in connection with FIGS. 4-7 can be used for updating any information that is stored in the storage device 216.

[0083] FIG. 5 is a flow chart of one exemplary embodiment of a method 500 of updating an insertion count on a storage device of a connector. The particular exemplary embodiment of method 500 shown in FIG. 5 is described here as being implemented using the storage device 216 and the connector assembly 102 described above in connection with FIGS. 1 and 2 (though other embodiments can be implemented in other ways). More specifically, in this exemplary embodiment, at least a portion of method 500 is implemented by the software 190 executing on the programmable processor 106 associated with the port 104 into which the connector 214 is inserted.

[0084] Method 500 comprises detecting an insertion of the connector 214 into the port 104 (block 502). Insertion of the connector 214 can be detected using various approaches, such as by detecting connection of the media interface 108 with the storage device interface 218 through completion of a circuit between the power leads and the ground leads of the media interface 108 and the storage device interface 218 using a pull-up resistor. Examples of such approaches are described in United States Provisional Application Serial No. 61/353,906, titled "SWITCH-STATE INFORMATION AGGREGATION", filed on June 11, 2010, which is hereby incorporated herein by reference. In the exemplary embodiment described here in connection with FIG. 5, the software 190 executing on the programmable processor 106 associated with the port 104 is informed of the insertion of the connector 214 into the port 104.

[0085] After the insertion of the connector 214 is detected, an insertion count for the connector 214 is determined from at least some of the information stored in the storage device 216 attached to the connector 214 (block 504). In one implementation of this embodiment, the software 190 executing on the processor 106 associated with the port 104 into which the connector 214 was inserted reads at least one of the copies 418 and

420 of the insertion count that is stored in the read/write section 402 of the storage device 216. The processor 106 (and the software 190 executing thereon) uses the media interface 108 associated with the port 104 into which the connector 214 is inserted to interact with the storage interface 218 of the storage device 216 in order to read at least one copies 418 and 420 of the insertion count from the storage device 216. The software 190 determines if the one or more of the copies 418 and 420 of the insertion count read from the storage device 216 are valid. If one of the copies 418 and 420 of the insertion count read from the storage device 216 is valid, the software 190 uses the insertion count included in that valid copy 418 or 420 as the current insertion count for the connector 214. The valid copy 418 or 420 of the insertion count read from the storage device 216 that is used as the current insertion count for the connector 214 is also referred to here as the "primary" copy of the insertion count stored on the storage device 216, and the copy 406 or 408 of the read/write information in which the primary copy of the insertion count is included is also referred to here as the "primary" copy of the read/write information stored on the storage device 216. Likewise, the other copy 418 or 420 of the insertion count (which is not used as the current insertion count) is also referred to here as the "secondary" copy of the insertion count stored on the storage device 216, and the copy 406 or 408 of the read/write information in which the secondary copy of the insertion count is included is also referred to here as the "secondary" copy of the read/write information stored on the storage device 216. An example embodiment of a method of determining the current insertion count for a connector 214 referenced in block 504 is described below with reference to FIG. 6.

[0086] After determining the current insertion count for the connector 214, the current insertion count is incremented (block 506). In the exemplary embodiment described here in connection with FIG. 5, the software 190 executing on the processor 106 associated with the port 104 into which the connector 214 was inserted increments the insertion count value by one. In other embodiments, the insertion count may be incremented by another value or decremented from a number of insertion counts remaining.

[0087] After the insertion count is incremented, the updated insertion count is stored in the storage device 216 using a plurality of sequential storage operations (block 508). In the exemplary embodiment described here in connection with FIG. 5, the software 190 executing on the processor 106 associated with the port 104 into which the connector 214 was inserted uses the media interface 108 associated with that port 104 to interact with the storage device interface 218 for the storage device 216 in order to first overwrite the secondary copy 418 or 420 of the insertion count stored in the storage

device 216 with the updated insertion count and then overwrite the primary copy 418 or 420 of the insertion count stored in storage device 216 with the updated insertion count. An example embodiment of a method for performing the sequential storage operations referenced in block 508 is described below with reference to FIG. 7. In other embodiments, the updated insertion count is redundantly stored in other ways.

[0088] FIG. 6 is a flow chart of one exemplary embodiment of a method of determining the current insertion count for a connector. The particular exemplary embodiment of method 600 shown in FIG. 6 is described here as being implemented using the storage device 216 and the connector assembly 102 described above in connection with FIGS. 1, 2, and 4 (though other embodiments can be implemented in other ways). More specifically, in this exemplary embodiment, at least a portion of method 600 is implemented by the software 190 executing on the programmable processor 106 associated with the port 104 into which the connector 214 is inserted.

[0089] Method 600 comprises reading the first copy 418 of the insertion count from the read/write section 402 of the storage device 216 (block 602). Method 600 also comprises reading the second copy 420 of the insertion count from the read/write section 402 of the storage device 216 (block 604). While the read operations referenced at block 602 and block 604 are shown in FIG. 6 as individual and sequential operations, those read operations can also be implemented together in a single read transaction and/or can also be implemented as two parallel read transactions. Moreover, the reading of the second copy 420 of the insertion count can be delayed until after it is determined that the first copy 418 of the insertion count is not valid (as described below in connection with block 608).

[0090] In the exemplary embodiment described here in connection with FIG. 6, the software 190 executing on the processor 106 associated with the port 104 into which the connector 214 is inserted reads all the information stored in the storage device 216 (including the first and second copies 406 and 408 of the read/write information stored in the read/write section 402 and the read only information stored in the read only section 404) in one read transaction. As noted above, the first and second copies 406 and 408 of the read/write information stored in the read/write section 402 of the storage device 216 include respective first and second copies 418 and 420 of the insertion count as well as respective checksums 412 and 416. In other embodiments, the first and second copies 418 and 420 of the insertion count can be read from the storage device 216 in other ways. For example, the first copy 406 (or a portion thereof such as the first copy 418 of the insertion count and/or the checksum 412) can be read from the storage device 216 in a read transaction that is separate from the read

transaction in which the second copy 408 (or a portion thereof such as the second copy 420 of the insertion count and/or the checksum 416) is read from the storage device 216 and/or one or both of the first and second copies 406 and 408 (or portions thereof) can be read from the storage device 216 in one or more read transactions that are separate from the one or more read transactions in which the read only information (or portions thereof) is read from the storage device 216.

[0091] Method 600 also comprises validating the first copy 418 of the insertion count read from the storage device 216 (block 606). In the exemplary embodiment described here in connection with FIG. 6, the checksum 412 associated with the first copy 418 of the insertion count comprises a cyclic redundancy check (CRC) code that is calculated based on the data portion 410 of the first copy 406 of the read/write information. In this example embodiment, the data portion 410 of the first copy 406 of the read/write information read from the storage device 216 is used to calculate a checksum (more specifically, a CRC value). In other embodiments, a CRC or other checksum could be calculated at other stages, such as during or immediately after the read operation referenced at block 602. This calculated checksum is compared against the first checksum 412 included in the first copy 406 of the read/write information read from the storage device 216 in order to determine if these checksums are equal. If the calculated checksum is equal to the first checksum 412 read from the storage device 216, the first copy 406 of the read/write information, including the first copy 418 of the insertion count and the first checksum 412, is considered valid. If the first copy 418 of the insertion count read from the storage device 216 is valid (block 608), then the value for the insertion count stored in the first copy 418 is used as the current insertion count of the connector 214 (block 610).

[0092] If the first copy 418 of the insertion count read from the storage device 216 is not valid, then the second copy 420 read from the storage device 216 is validated (block 612). In the exemplary embodiment described here in connection with FIG. 6, the data portion 414 of the second copy 408 of the read/write information read from the storage device 216 is used to calculate a checksum (more specifically, a CRC value). In other embodiments, a CRC or other checksum could be calculated at other stages, such as during or immediately after the read operation referenced at block 604. This calculated checksum is compared against the second checksum 416 included in the second copy 408 of the read/write information read from the storage device 216 in order to determine if these checksums are equal. If the calculated checksum is equal to the second checksum 416 read from the storage device 216, the second copy 406 of the read/write information, including the second copy 420 of the insertion count and the

second checksum 416, is considered valid. If the second copy 420 of the insertion count from the storage device 216 is valid (block 614), then the value for the insertion count stored in the second copy 420 is used as the current insertion count of the connector 214 (block 616).

[0093] If the second copy 420 of the insertion count read from the storage device 216 is not valid, then the method 600 enters an error condition (block 618). In some embodiments, the error condition indicates that the insertion count for the currently inserted connector 214 has been lost or corrupted. In some embodiments, a user is notified of the error and instructed to perform appropriate tasks in response, such as replacing the connector. In other example embodiments, method 600 is repeated at least one additional time to ensure the error condition is correct. In other embodiments, other actions can be taken by the system or user to resolve the error condition.

[0094] As noted above, the valid copy 418 or 420 of the insertion count read from the storage device 216 that is used as the current insertion count for the connector 214 is also referred to here as the “primary” copy of the insertion count stored on the storage device 216, and the copy 406 or 408 of the read/write information in which the primary copy of the insertion count is included is also referred to here as the “primary” copy of the read/write information stored on the storage device 216. Likewise, the other copy 418 or 420 of the insertion count (which is not used as the current insertion count) is also referred to here as the “secondary” copy of the insertion count stored on the storage device 216, and the copy 406 or 408 of the read/write information in which the secondary copy of the insertion count is included is also referred to here as the “secondary” copy of the read/write information stored on the storage device 216.

[0095] FIG. 7 is a flow chart of one exemplary embodiment of a method of redundantly storing an insertion count in a storage device of a connector. The particular exemplary embodiment of method 700 shown in FIG. 7 is described here as being implemented using the storage device 216 and the connector assembly 102 described above in connection with FIGS. 1, 2, and 4 and in connection with methods 500 and 600 described above in connection with FIGS. 5 and 6 (though other embodiments can be implemented in other ways). More specifically, in this exemplary embodiment, at least a portion of the method 600 is implemented by the software 190 executing on the programmable processor 106 associated with the port 104 into which the connector 214 is inserted.

[0096] In the embodiment described here in connection with FIG. 7, method 700 is performed after the connector 214 has been inserted into a port 104 of a connector assembly 102 and an insertion count has been determined for the connector 214 as described above in connection with FIG. 6. Method 700 comprises calculating an

updated checksum associated with the incremented insertion count (block 702). As described above in connection with block 504, the current insertion count for the connector 214 is determined from the data part 410 or 414 of a valid copy 406 or 408 of the read/write information stored in the storage device 216. This data part 410 or 414 is referred to here as the "valid data part" 410 or 414. In the exemplary embodiment described here in connection with FIG. 7, an updated checksum is calculated by first updating the valid data part 410 or 414 by replacing the copy 418 or 420 of the insertion count read from the storage device 216 with the incremented insertion count and then calculating the checksum for the updated data part 410 or 414 (for example, by calculating a CRC code for the updated data part).

[0097] Then, the secondary copy 406 or 408 of the read/write information stored in the storage device is updated to include the incremented insertion count and the updated checksum (block 704). In one implementation of this embodiment, the secondary copy 406 or 408 of the read/write information that is stored in the storage device 216 is completely overwritten with the updated data part (which includes the incremented insertion count as well as any other read/write information) and the updated checksum calculated from that updated data part. In another implementation of this embodiment, the secondary copy 418 or 420 of the insertion count stored in the storage device 216 is overwritten with the incremented insertion count (but the other read/write information (if any) is not changed) and the associated checksum 412 or 416 is overwritten with the updated checksum calculated for the updated data part.

[0098] As noted above in connection with FIG. 4, in the absence of any errors, the content of the primary copy 406 or 408 of the read/write information stored in the read/write section 402 of the storage device 216 should be the same as the content of the secondary copy 406 or 408 of the read/write information stored in the read/write section 402 of the storage device 216. Thus, after the secondary copy 406 or 408 of the read/write information is updated to include the incremented insertion count and the updated checksum, the primary copy 406 or 408 of the read/write information is updated to include the incremented insertion count and the updated checksum (block 706). In some implementations of the embodiment described here in connection with FIG. 7, the primary copy 406 or 408 of the read/write information is updated to include the incremented insertion count and the updated checksum in the same way the secondary copy 406 or 408 is updated (though in other implementations and embodiments, the primary copy 406 or 408 of the read/write information is updated to include the incremented insertion count and the updated checksum in a way that differs from how the secondary copy 406 or 408 is updated).

[0099] As noted above with reference to FIG. 4, the read/write section 402 is configured to allow only one of the first copy 406 and the second copy 408 to be updated, overwritten, or erased at a time and by using a single write operation at a time. Thus, the distinct write operations described at blocks 602 and 604 do not overlap in time. Because the storage operations do not overlap in time and by updating the secondary copy 406 or 408 of the read/write information first, in the event that the updating of the secondary copy 406 or 408 fails or corrupts the value of the incremented insertion count stored in the secondary copy 406 or 408 (for example, because the connector 214 is pulled out of a port 104 while such operation is still being performed), the primary insertion count (and the primary copy 406 or 408 in which it is included) should still be valid. Similarly, in the event that the updating of the primary copy 406 or 408 fails or corrupts the value of the incremented insertion count stored in the primary copy 406 or 408 (for example, because the connector 214 is pulled out of a port 104 while such operation is still being performed), the secondary insertion count (and the secondary copy 406 or 408 in which it is included) should be valid. In this way, there should always be a valid copy of the incremented insertion count, even if either of the storage operations described at blocks 602 and 604 fails or corrupts the value of the incremented insertion count stored in either the first copy 406 or the second copy 408. As long as one of the first copy 406 and the second copy 408 is valid and not corrupted, the insertion count will be recoverable upon the next insertion of the connector 214 into the port 104 (though the insertion count may be off by one).

[0100] While the storage operations referenced at block 704 and block 708 are described as individual operations in the particular embodiment shown in method 700 of FIG. 7, in other example embodiments, the first copy 406 and the second copy 408 are stored during a single transmission or session in which multiple storage transactions occur – one or more storage transactions for updating the first copy 406 and one or more other storage transactions for updating the second copy 408. Still, the actual storage of the first copy 406 and the second copy 408 occur sequentially and do not overlap in time.

[0101] While the embodiments described above only describe two copies of the incremented insertion count being read and stored using two sets of read operations and two sets of storage operations, more than two redundant sets of storage operations may be used in other embodiments. While the embodiments described above describe the insertion of the connector 214 into the port 104, it is understood that the insertion count can be updated on the storage device of the connector 214 by other compatible

ports and that the port 104 can update the insertion count on other compatible connectors.

[0102] Further details, embodiments, and implementations can be found in the following United States Patent Applications, all of which are hereby incorporated herein by reference: United States Provisional Patent Application Serial No. 61/252,964, filed on 10/19/2009, titled "ELECTRICAL PLUG FOR MANAGED CONNECTIVITY", Attorney Docket No. 02316.3045USP1; United States Provisional Patent Application Serial No. 61/253,208, filed on 10/20/2009, titled "ELECTRICAL PLUG FOR MANAGED CONNECTIVITY", Attorney Docket No. 02316.3045USP2; United States Patent Application Serial No. 12/907,724, filed on 10/19/2010, titled "MANAGED ELECTRICAL CONNECTIVITY SYSTEMS", Attorney Docket No. 02316.3045USU1; United States Provisional Patent Application Serial No. 61/303,948, filed on 2/12/2010, titled "PANEL INCLUDING BLADE FEATURE FOR MANAGED CONNECTIVITY", Attorney Docket No. 02316.3069USP1; United States Provisional Patent Application Serial No. 61/413,844, filed on 11/15/2010, titled "COMMUNICATIONS BLADED PANEL SYSTEMS", Attorney Docket No. 02316.3069USP2; United States Provisional Patent Application Serial No. 61/439,693, filed on 2/4/2011, titled "COMMUNICATIONS BLADED PANEL SYSTEMS", Attorney Docket No. 02316.3069USP3; United States Patent Application Serial No. 13/025,730, filed on 2/11/2011, titled "COMMUNICATIONS BLADED PANEL SYSTEMS", Attorney Docket No. 02316.3069USU1; United States Patent Application Serial No. 13/025,737, filed on 2/11/2011, titled "COMMUNICATIONS BLADED PANEL SYSTEMS", Attorney Docket No. 02316.3069USU2; United States Patent Application Serial No. 13/025,743, filed on 2/11/2011, titled "COMMUNICATIONS BLADED PANEL SYSTEMS", Attorney Docket No. 02316.3069USU3; United States Patent Application Serial No. 13/025,750, filed on 2/11/2011, titled "COMMUNICATIONS BLADED PANEL SYSTEMS", Attorney Docket No. 02316.3069USU4; United States Provisional Patent Application Serial No. 61/303,961; filed on 2/12/2010, titled "Fiber Plug And Adapter For Managed Connectivity", Attorney Docket No. 02316.3071USP1; United States Provisional Patent Application Serial No. 61/413,828, filed on 11/15/2010, titled "Fiber Plugs And Adapters For Managed Connectivity", Attorney Docket No. 02316.3071USP2; United States Provisional Patent Application Serial No. 61/437,504, filed on 1/28/2011, titled "Fiber Plugs And Adapters For Managed Connectivity", Attorney Docket No. 02316.3071USP3 ; United States Patent Application Serial No. 13/025,784, filed on 2/11/2011, titled "Managed Fiber Connectivity Systems", Attorney Docket No. 02316.3071USU1; United States Patent Application Serial No. 13/025,788, filed on 2/11/2011, titled "Managed Fiber Connectivity Systems", Attorney Docket No

02316.3071USU2; United States Patent Application Serial No. 13/025,797, filed on 2/11/2011, titled "Managed Fiber Connectivity Systems", Attorney Docket No. 02316.3071USU3; United States Patent Application Serial No. 13/025,841, filed on 2/11/2011, titled "Managed Fiber Connectivity Systems", Attorney Docket No. 02316.3071USU4; United States Provisional Patent Application Serial No. 61/413,856, filed on 11/15/2010, titled "CABLE MANAGEMENT IN RACK SYSTEMS", Attorney Docket No. 02316.3090USP1; United States Provisional Patent Application Serial No. 61/466,696, filed on 3/23/2011, titled "CABLE MANAGEMENT IN RACK SYSTEMS", Attorney Docket No. 02316.3090USP2; United States Provisional Patent Application Serial No. 61/252,395, filed on 10/16/2009, titled "MANAGED CONNECTIVITY IN ELECTRICAL SYSTEMS", Attorney Docket No. 02316.3021USP1; United States Patent Application Serial No. 12/905,689, filed on 10/15/2010, titled "MANAGED CONNECTIVITY IN ELECTRICAL SYSTEMS", Attorney Docket No. 02316.3021USU1; United States Provisional Patent Application Serial No. 61/252,386, filed on 10/16/2009, titled "MANAGED CONNECTIVITY IN FIBER OPTIC SYSTEMS", Attorney Docket No. 02316.3020USP1; and United States Patent Application Serial No. 12/905,658, filed on 10/15/2010, titled "MANAGED CONNECTIVITY IN FIBER OPTIC SYSTEMS", Attorney Docket No. 02316.3020USU1.

[0103] A number of embodiments of the invention defined by the following claims have been described. Nevertheless, it will be understood that various modifications to the described embodiments may be made without departing from the spirit and scope of the claimed invention. Accordingly, other embodiments are within the scope of the following claims.

32
CLAIMS

What is claimed is:

1. A connector assembly comprising:
a port having a media interface configured to interface with a storage device interface of a connector, the connector attached to a segment of physical communication media, the connector also including a storage device;
a programmable processor configured to execute software that stores information to the storage device using a plurality of redundant storage operations by which a plurality of copies of the information is sequentially stored in the storage device.
2. The connector assembly of claim 1, wherein the software is further configured to read a first copy of the information and a second copy of the information from the storage device.
3. The connector assembly of claim 2, wherein the software is further configured to validate a first checksum of the first copy of the information.
4. The connector assembly of claim 1, wherein the connector assembly comprises a plurality of ports.
5. The connector assembly of claim 1, wherein the processor comprises a slave processor and wherein the connector assembly further comprises a master processor that is communicatively coupled to the slave processor.
6. The connector assembly of claim 5, wherein the connector assembly comprises a plurality of slave processors that are communicatively coupled to the master processor.
7. The connector assembly of claim 1, wherein the connector assembly comprises at least one of a rack-mounted connector assembly, a wall-mounted connector assembly, an inter-networking device, a fiber distribution hub (FDH), a fiber splice panel, and a fiber termination point.
8. The connector assembly of claim 1, wherein the connector assembly comprises at least one of a patch panel, a distribution unit, a media converter, a wall-mounted

connector box, wall-mounted jack, wall-mounted outlet, a wall-mounted media converter, a switch, a bridge, a router, a hub, a repeater, a gateway, and an access points.

9. A connector assembly comprising:
 - a port having a media interface configured to interface with a storage device interface of a connector, the connector attached to a segment of physical communication media, and the connector also including a storage device;
 - a programmable processor configured to execute software that stores an insertion count to the storage device through a plurality of redundant storage operations, wherein the insertion count indicates a number of insertions of the connector into any port.
10. The connector assembly of claim 9, wherein the plurality of redundant storage operations sequentially store a first copy of the insertion count and a second copy of the insertion count to the storage device.
11. The connector assembly of claim 9, wherein the software is further configured to detect an insertion of the connector into the port.
12. The connector assembly of claim 9, wherein the software is further configured to read the insertion count from the storage device.
13. The connector assembly of claim 12, wherein the software is further configured to read a first copy of the insertion count from the storage device and to read a second copy of the insertion count from the storage device.
14. The connector assembly of claim 13, wherein the software is further configured to validate a first checksum of the first copy of the insertion count.
15. The connector assembly of claim 14, wherein the first checksum comprises a cyclic redundancy check (CRC) code.
16. The connector assembly of claim 13, wherein the software is further configured to increment the insertion count from the first copy of the insertion count and then overwrite the insertion count in the second copy with the incremented insertion count and then overwrite the insertion count in the first copy with the incremented insertion count.

17. The connector assembly of claim 9, wherein the software is further configured to increment the insertion count.

18. The connector assembly of claim 9, the connector assembly comprising at least one of a rack-mounted connector assembly, a wall-mounted connector assembly, an inter-networking device, a fiber distribution hub (FDH), a fiber splice panel, and a fiber termination point.

19. The connector assembly of claim 9, the connector assembly comprising at least one of a patch panel, a distribution unit, a media converter, a wall-mounted connector box, a wall mounted jack, a wall-mounted outlet, a wall-mounted media converter, a switch, a bridge, a router, a hub, a repeater, a gateway, and an access point.

20. The connector assembly of claim 9, wherein the storage device comprises non-volatile memory.

21. The connector assembly of claim 9, wherein the segment of physical communication media comprises at least one of a copper patch cord and an optical fiber patch cord.

22. The connector assembly of claim 9, wherein the segment of physical communication media comprises a twisted-pair cable.

23. The connector assembly of claim 22, wherein the twisted-pair cable has an RJ-45 modular jack connected to at least one end.

24. The connector assembly of claim 9, wherein the segment of physical communication media comprises a fiber optic cable.

25. The connector assembly of claim 24, wherein the fiber optic cable has an LC connector, an SC connector, an ST connector, or an MPO connector connected to at least one end thereof.

26. The connector assembly of claim 9, wherein the connector assembly comprises a plurality of ports.

27. The connector assembly of claim 9, wherein the processor comprises a slave processor and wherein the connector assembly further comprises a master processor that is communicatively coupled to the slave processor.
28. The connector assembly of claim 27, wherein the connector assembly comprises a plurality of slave processors that are communicatively coupled to the master processor.
29. A method comprising:
 - storing a first copy of an insertion count to a storage device of a connector during a first set of one or more storage operations, wherein the insertion count indicates a number of insertions of the connector into any port; and
 - storing a second copy of the insertion count to the storage device during a second set of one or more storage operations.
30. The method of claim 29, further comprising:
 - detecting an insertion of the connector into any port.
31. The method of claim 29, further comprising:
 - reading a first copy of a previous insertion count from the storage device.
32. The method of claim 31, further comprising:
 - validating a first checksum of the first copy of the previous insertion count.
33. The method of claim 32, wherein the first checksum comprises a cyclic redundancy check (CRC) code.
34. The method of claim 31, further comprising:
 - incrementing the first copy of the previous insertion count to calculate the current insertion count.
35. The method of claim 29, further comprising:
 - reading a second copy of the previous insertion count from the storage device.
36. The method of claim 35, further comprising:
 - validating a second checksum of the second copy of the previous insertion count.

37. The method of claim 36, wherein the second checksum comprises a cyclic redundancy check (CRC) code.
38. The method of claim 35, further comprising:
incrementing the second copy of the previous insertion count to calculate the current insertion count.
39. A connector assembly comprising:
a port having a media interface configured to couple with a storage device interface of a connector, the connector also including a storage device;
a programmable processor configured to execute software;
wherein an insertion count indicates a number of insertions of a connector into at least one port;
wherein the software is configured to detect an insertion of the connector into the port;
wherein the software is configured to determine an insertion count for the connector from information read;
wherein the software is further configured to increment the insertion count;
wherein the software is further configured to update the information stored in the storage device in order to store the incremented insertion count to the storage device through a plurality of redundant storage operations.
40. The connector assembly of claim 39, wherein the plurality of redundant storage operations includes a first storage operation to store a first copy of the incremented insertion count to the storage device and a second storage operation to store the second copy of the incremented insertion count to the storage device; and
wherein the first storage operation completes before the first copy operation begins.
41. The connector assembly of claim 39, wherein the first checksum and the second checksum include cyclic redundancy checks.
42. A method comprising:
detecting an insertion of a connector into a port, the connector including a storage device;
reading a first copy of an insertion count from the storage device;
validating a first checksum of the first copy of the insertion count;

reading a second copy of the insertion count from the storage device;
validating a second checksum of the second copy of the insertion count;
incrementing the first copy of the insertion count;
storing a first copy of the incremented insertion count to the storage device; and
storing a second copy of the incremented insertion count to the storage device.

43. The method of claim 42, wherein the first copy of the incremented insertion count is stored to the storage device by overwriting the second copy of the insertion count read from the storage device and, then, after the first copy of the incremented insertion count is stored to the storage device by overwriting the second copy of the insertion count read from the storage device, the second copy of the incremented insertion count is stored to the storage device by overwriting the first copy of the insertion count read from the storage device.

44. The method of claim 42, wherein storing the first copy completes before storing the second copy begins.

45. A system comprising:

a connector assembly comprising a port having a media interface configured to interface with a storage device interface of a connector, the connector attached to a segment of physical communication media, and the connector also including a storage device; and

an aggregation point communicatively coupled to the connector assembly, wherein the aggregation point is configured to cause the connector assembly to send to the aggregation point at least some of the data read from the storage device attached to the connector connected to the port of the connector assembly;

wherein the aggregation point is configured to store at least some of the data sent by the connector assembly to the aggregation point; and

wherein the connector assembly is configured to store an insertion count to the storage device through a plurality of redundant storage operations, wherein the insertion count indicates a number of insertions of the connector into any port.

46. The system of claim 45, wherein the plurality of redundant storage operations sequentially store a first copy of the insertion count and a second copy of the insertion count to the storage device.

47. The system of claim 45, wherein the connector assembly is further configured to detect an insertion of the connector into the port.

48. The system of claim 45, wherein the connector assembly is further configured to read the insertion count from the storage device.

49. The system of claim 48, wherein the connector assembly is further configured to read a first copy of the insertion count and a second copy of the insertion count from the storage device.

50. The system of claim 49, wherein the connector assembly is further configured to validate a first checksum of the first copy of the insertion count.

51. The system of claim 45, wherein the connector assembly is further configured to increment the insertion count.

52. The system of claim 45, wherein the connector assembly comprises at least one of a rack-mounted connector assembly, a wall-mounted connector assembly, an inter-networking device, a fiber distribution hub (FDH), a fiber splice panel, and a fiber termination point.

53. The system of claim 45, wherein the connector assembly comprises at least one of a patch panel, a distribution unit, a media converter, a wall-mounted connector box, a wall mounted jack, a wall-mounted outlet, a wall-mounted media converter, a switch, a bridge, a router, a hub, a repeater, a gateway, and an access point.

54. The system of claim 45, wherein the storage device comprises non-volatile memory.

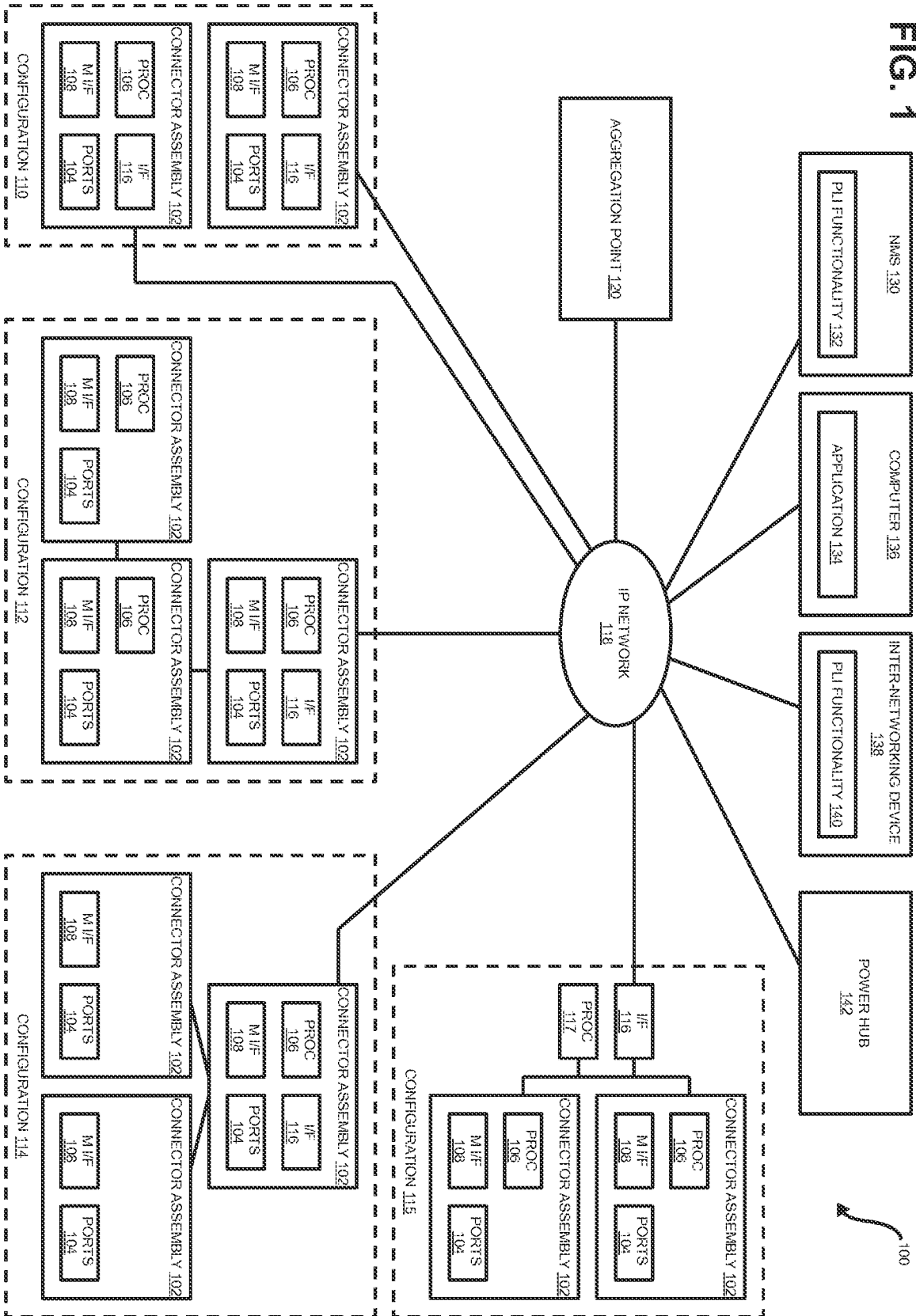
55. The system of claim 45, wherein the segment of physical communication media comprises at least one of a copper patch cord and an optical fiber patch cord.

56. The system of claim 45, wherein the segment of physical communication media comprises a twisted-pair cable.

57. The system of claim 45, wherein the segment of physical communication media comprises a fiber optic cable.

58. The system of claim 45, wherein the connector assembly comprises a plurality of ports.
59. The system of claim 45, wherein the connector assembly comprises a slave processor and a master processor that is communicatively coupled to the slave processor.
60. The system of claim 59, wherein the connector assembly comprises a plurality of slave processors that are communicatively coupled to the master processor.
61. The system of claim 45, wherein the aggregation point is configured to provide at least some of the data stored by the aggregation point to at least one other device.
62. The system of claim 61, wherein at least one other device comprises application-layer functionality executing on a computer communicatively coupled to the aggregation point.
63. The system of claim 45, wherein the aggregation point comprises middleware that provides an application programming interface (API) by which an external entity is able to access at least some of the data stored by the aggregation point.
64. The system of claim 63, wherein the external entity comprises at least one of a computer executing application-layer software, a network management system, an enterprise management system, and an inter-networking device.
65. The system of claim 45, wherein the aggregation point is at least one of:
implemented on a standalone network node;
integrated along with other network functionality;
distributed across multiple nodes in a network; and
implemented in a hierarchy of aggregation points.

FIG. 1



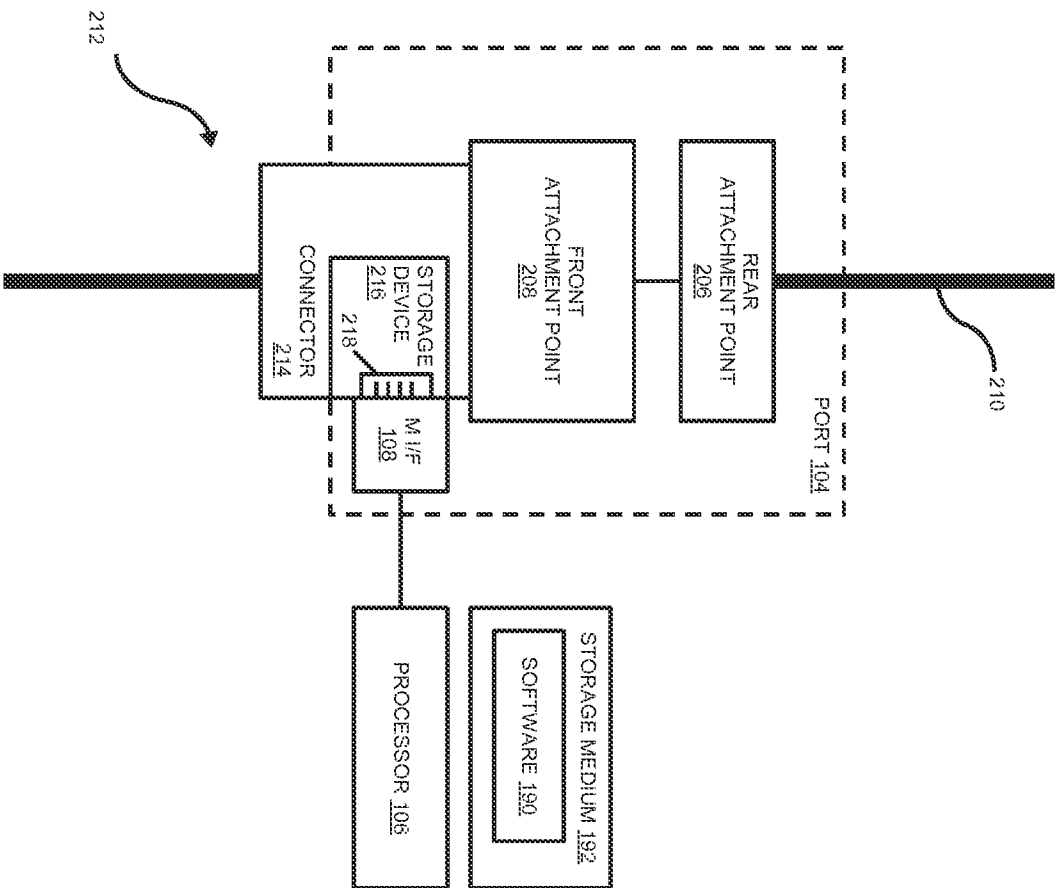


FIG. 2

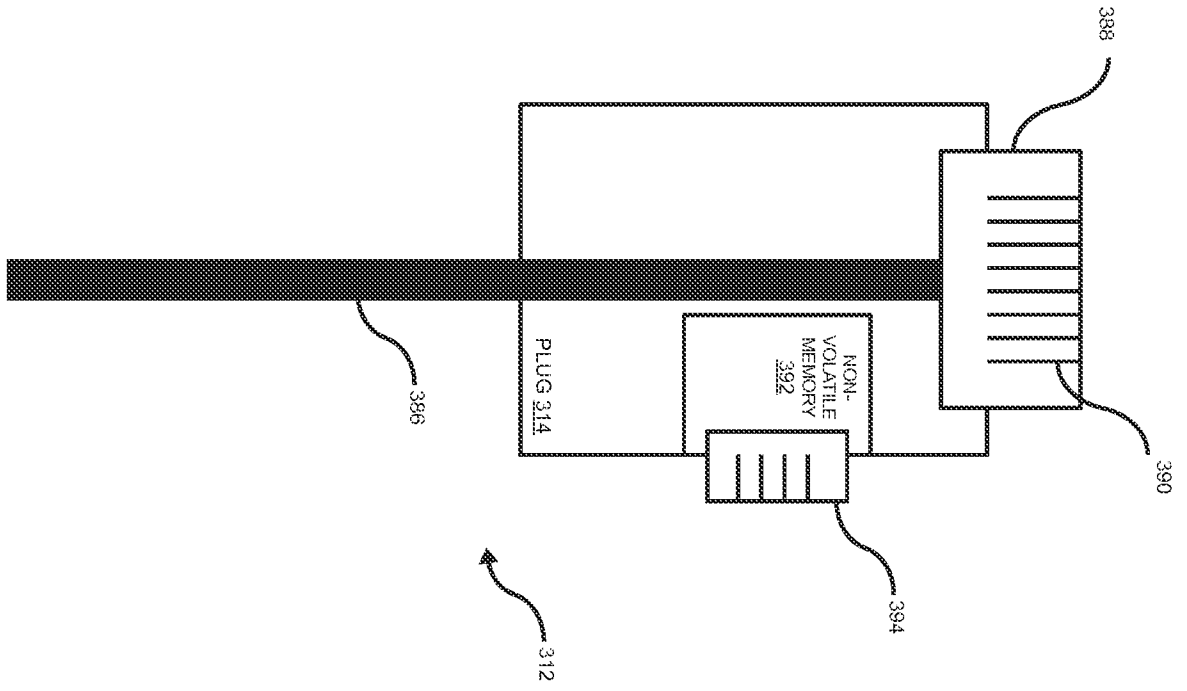


FIG. 3A

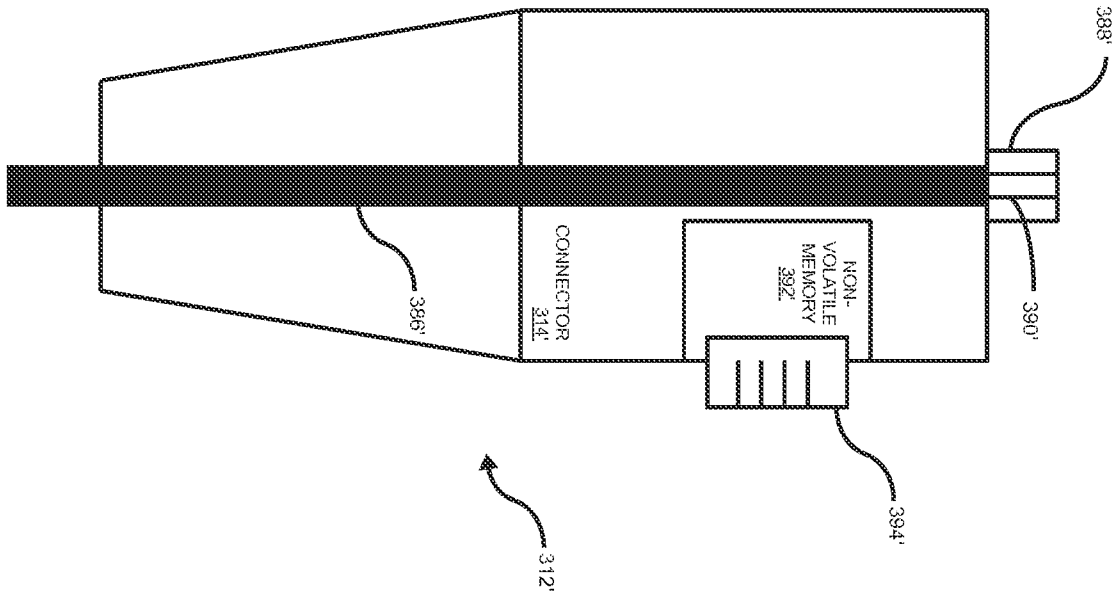


FIG. 3B

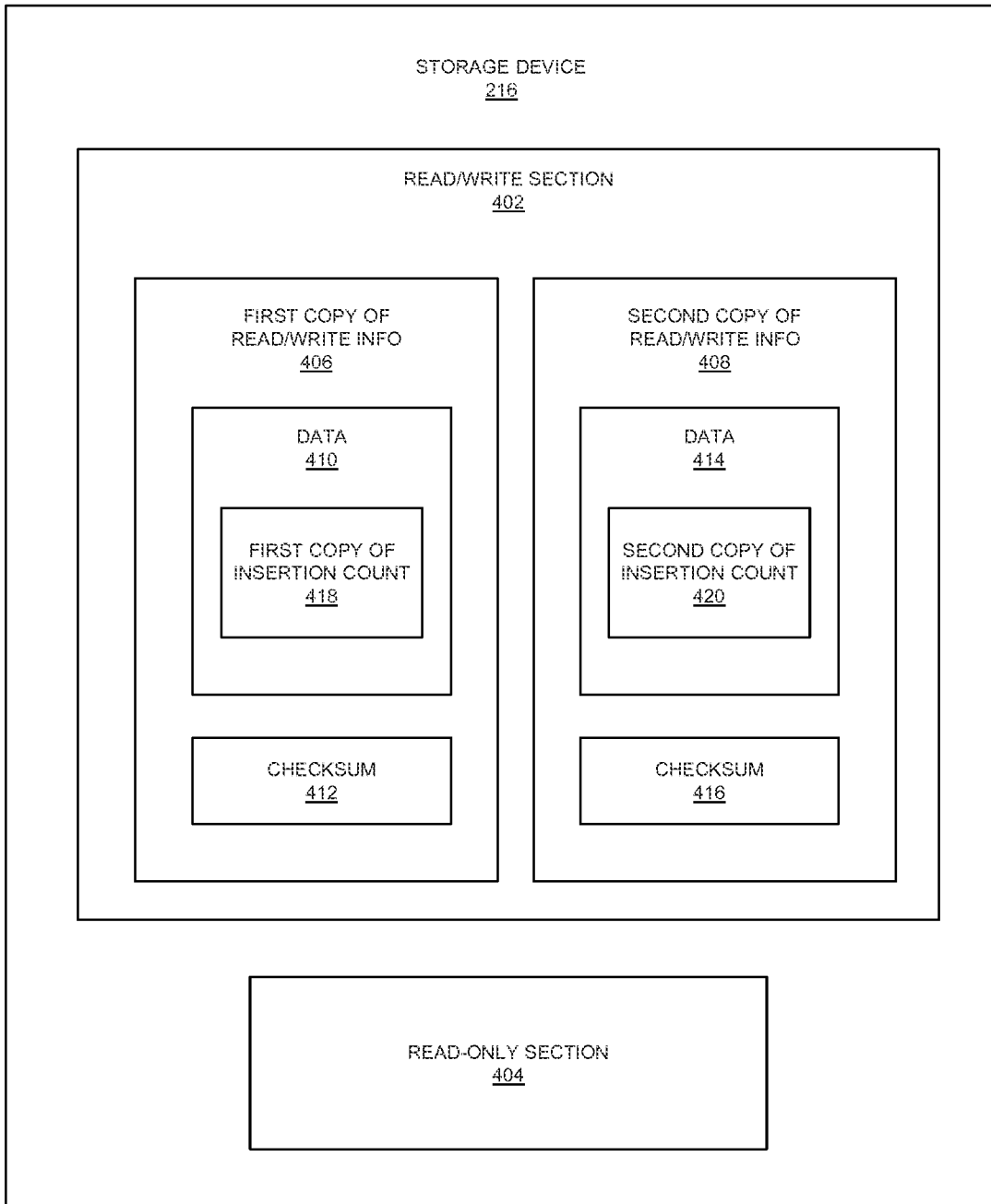


FIG. 4

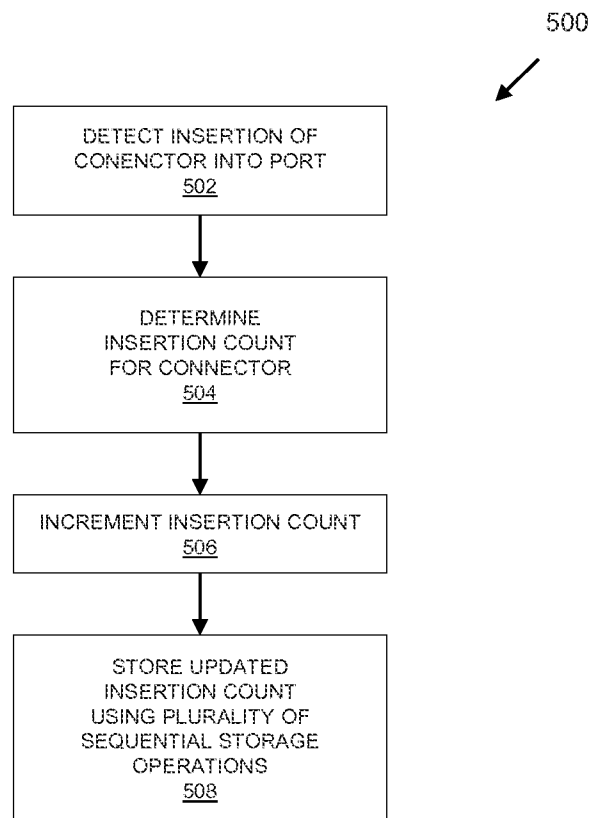


FIG. 5

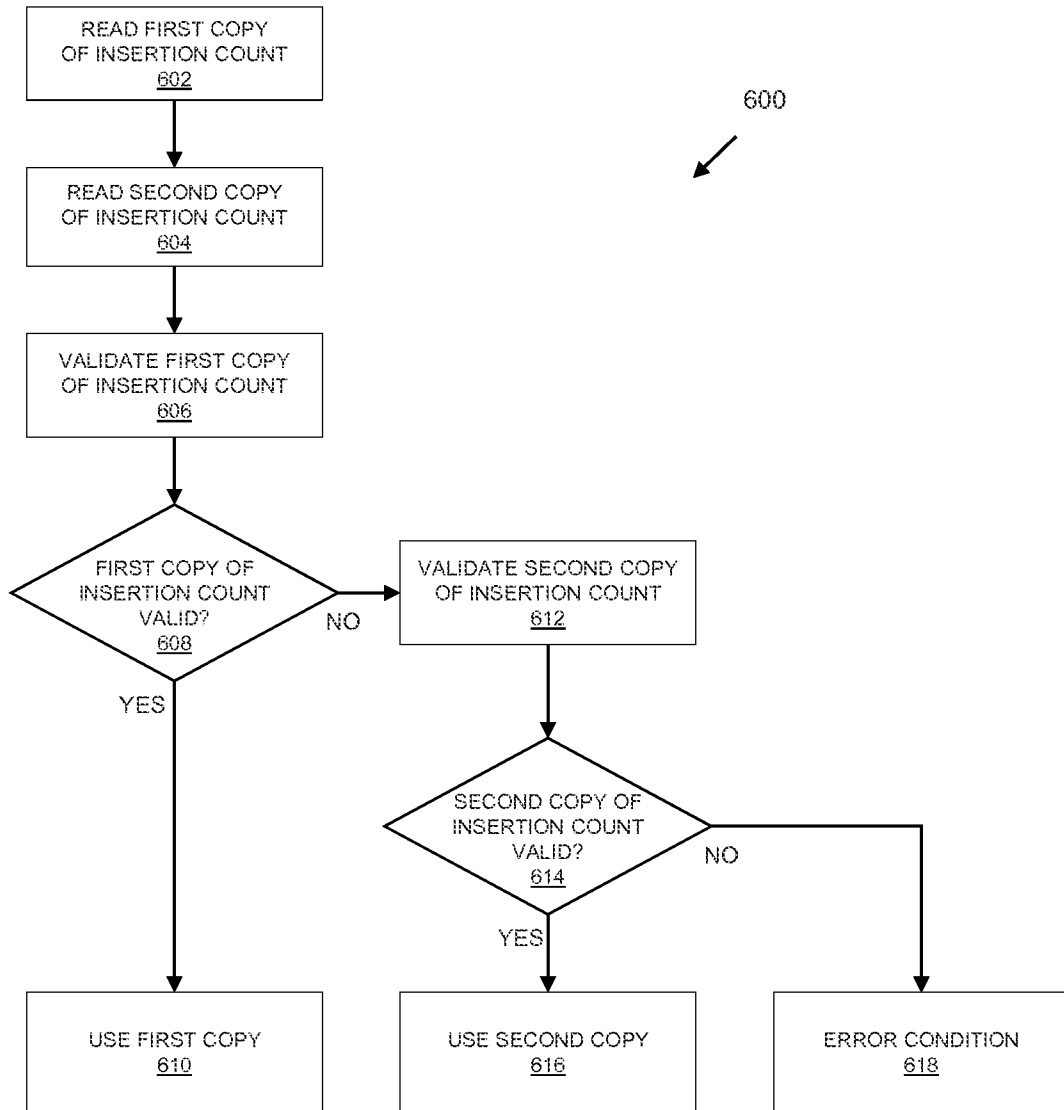


FIG. 6

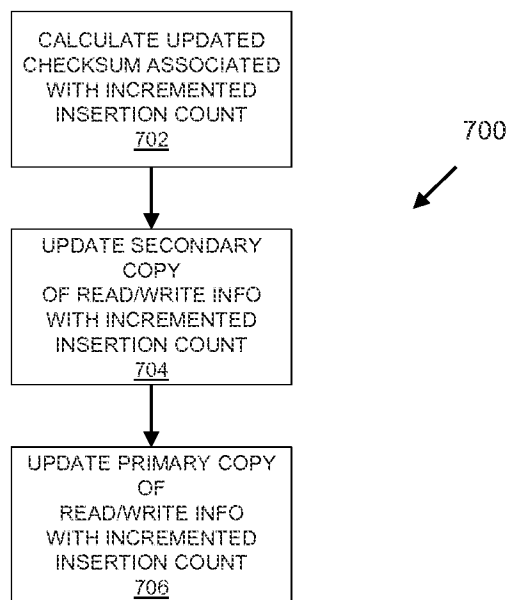


FIG. 7