US 20130064066A1

(19) **United States**

(12) **Patent Application Publication**

GHOSH et al.

(10) **Pub. No.: US 2013/0064066 A1**

(43) **Pub. Date: Mar. 14, 2013**

(54) UPDATING A SWITCH SOFTWARE IMAGE IN A DISTRIBUTED FABRIC PROTOCOL (DFP) SWITCHING NETWORK

(75) Inventors: **NIRAPADA GHOSH**, SUNNYVALE, CA (US); **DAYAVANTI G KAMATH**, SANTA CLARA, CA (US); **KESHAV KAMBLE**, FREMONT, CA (US); **DARREN LEU**, SAN JOSE, CA (US); **NILANJAN MUKHERJEE**, SANTA CLARA, CA (US); **VIJOY PANDEY**, SAN JOSE, CA (US)

(73) Assignee: **INTERNATIONAL BUSINESS MACHINES CORPORATION**, ARMONK, NY (US)

(21) Appl. No.: **13/229,867**

(22) Filed: **Sep. 12, 2011**

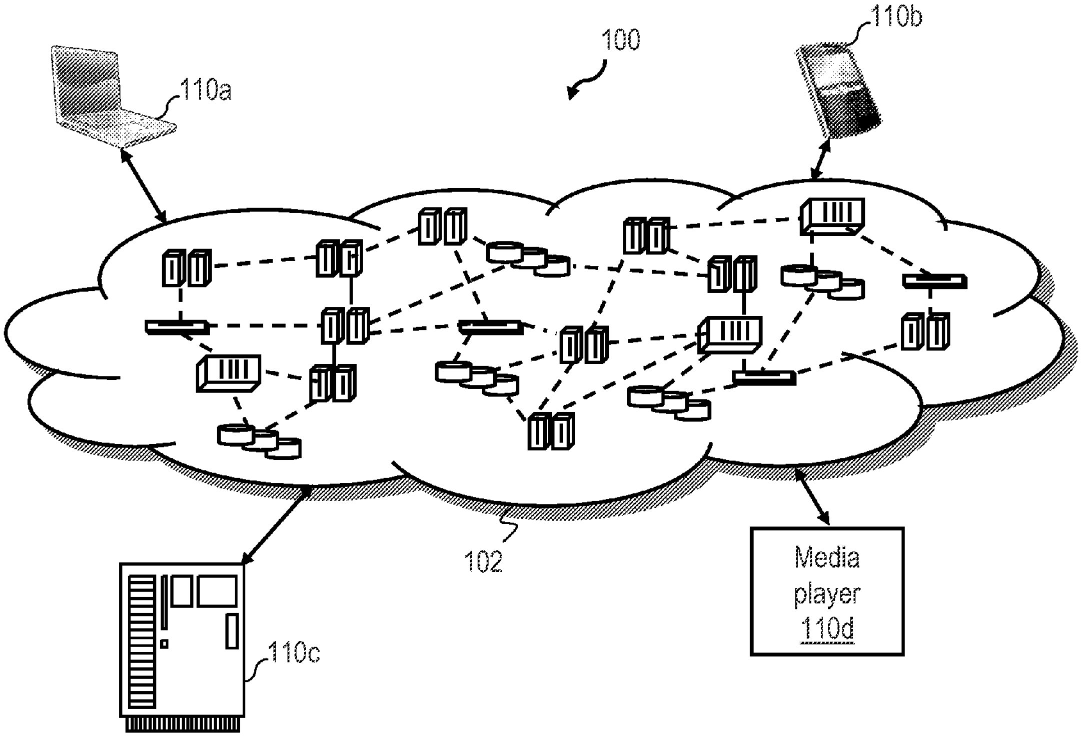
Publication Classification

(51) **Int. Cl.**
H04L 12/26 (2006.01)

(52) **U.S. Cl.**
USPC **370/216**

(57) ABSTRACT

A switching network has a plurality of switches including at least a switch and a managing master switch. At the managing master switch, a first capability vector (CV) is received from the switch. The managing master switch determines whether the first CV is compatible with at least a second CV in a network membership data structure that records CVs of multiple switches in the switching network. In response to detecting an incompatibility, the managing master switch initiates an image update to an image of the switch. In response to a failure of the image update at the switch, the switch boots utilizing a mini-DC module that reestablishes communication between the switch with the managing master switch and retries the image update.



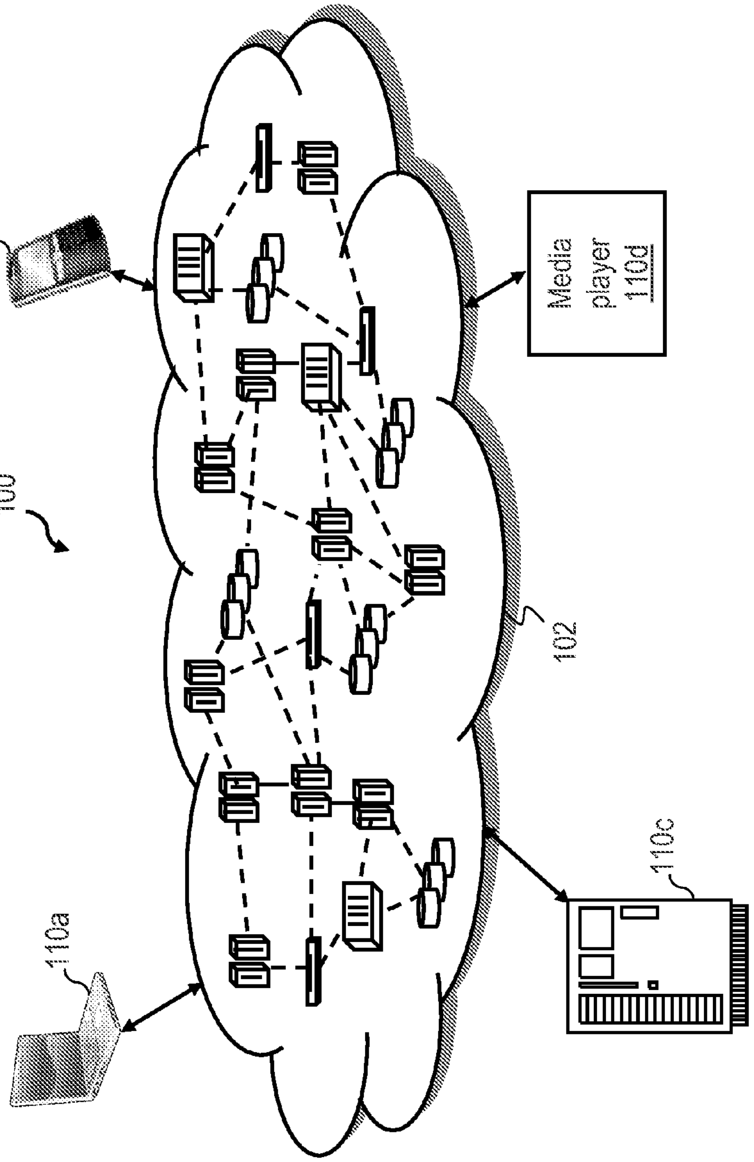


Figure 1

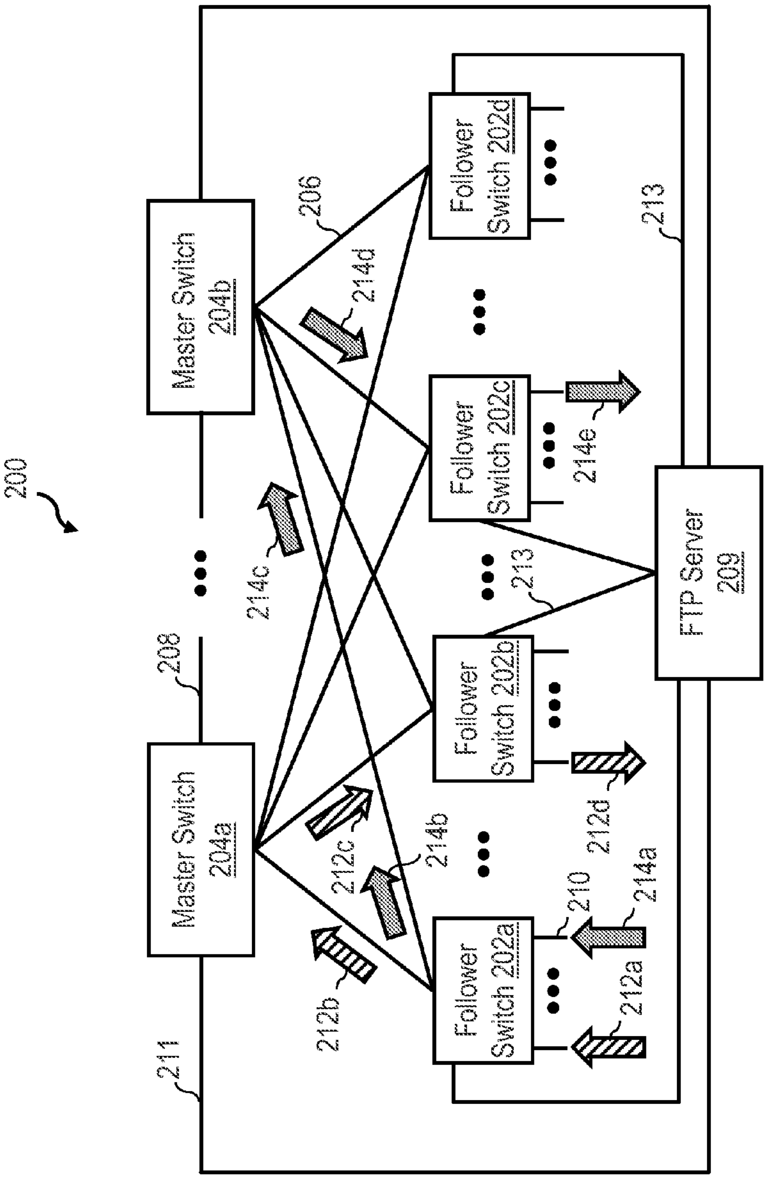

200
Master Switch 204a
Master Switch 204b
208
206
211
214c
214d
212b
212c
214b
Follower Switch 202a
Follower Switch 202b
Follower Switch 202c
Follower Switch 202d
210
212a
214a
212d
213
214e
213
FTP Server 209

Figure 2

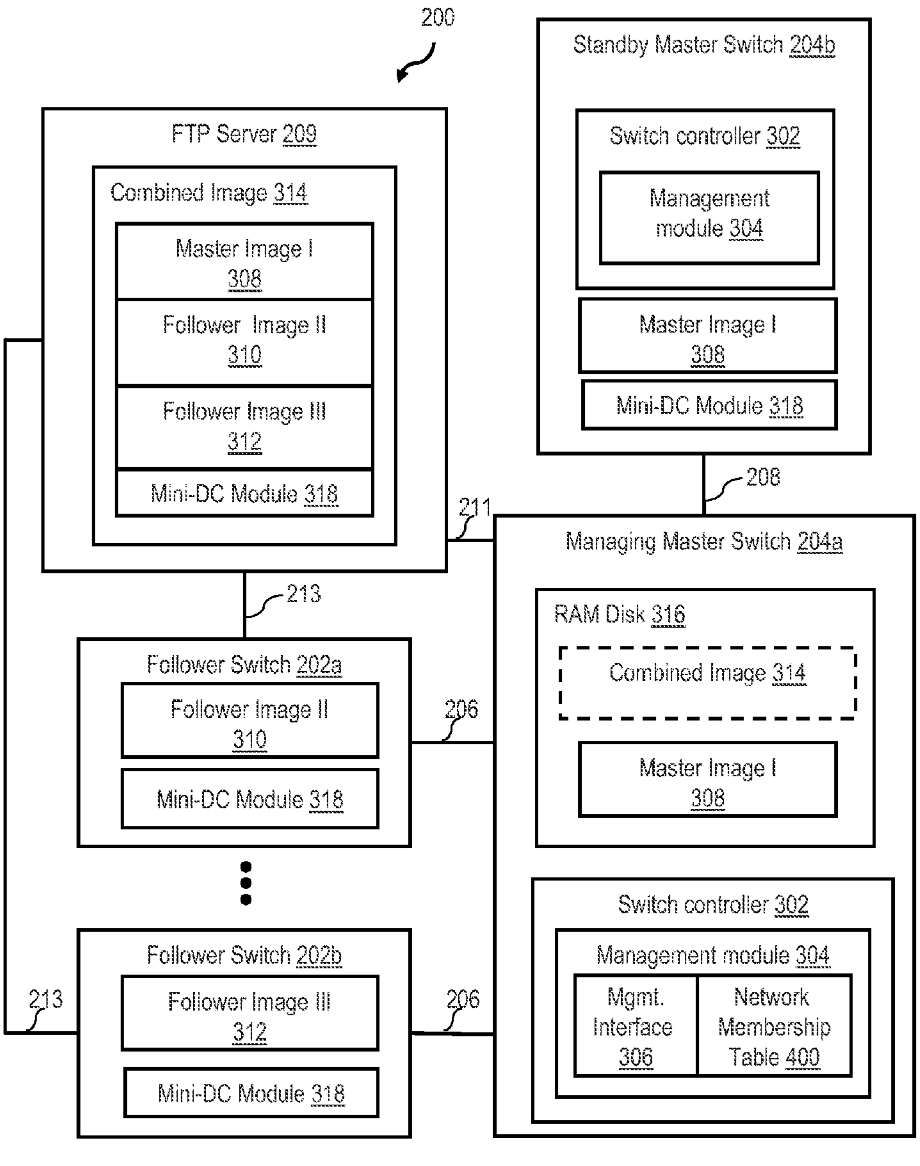

200
FTP Server 209
Combined Image 314
Master Image I 308
Follower Image II 310
Follower Image III 312
Mini-DC Module 318
Standby Master Switch 204b
Switch controller 302
Management module 304
Master Image I 308
Mini-DC Module 318
208
211
Managing Master Switch 204a
213
RAM Disk 316
Combined Image 314
Master Image I 308
Follower Switch 202a
Follower Image II 310
Mini-DC Module 318
206
Switch controller 302
Management module 304
Mgmt. Interface 306
Network Membership Table 400
Follower Switch 202b
Follower Image III 312
Mini-DC Module 318
213
206

Figure 3

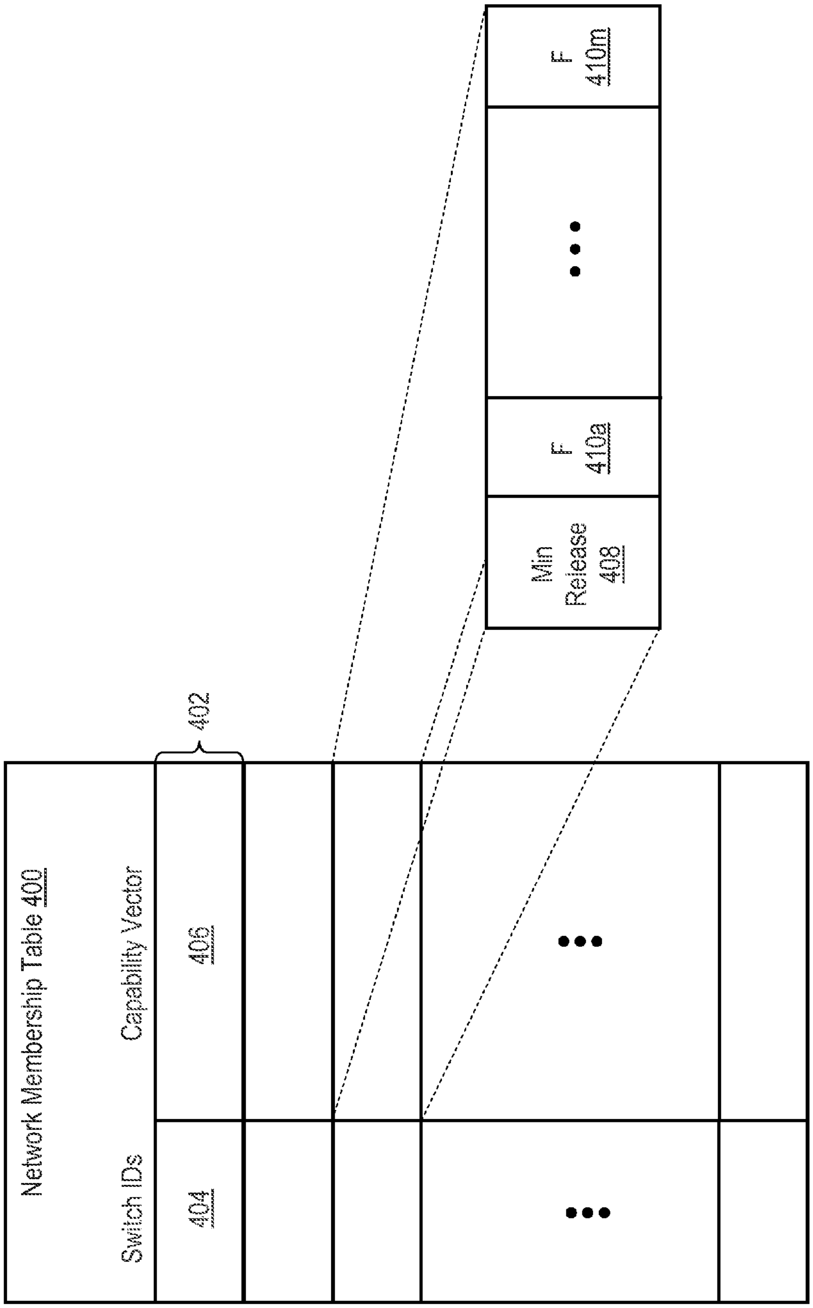

Network Membership Table 400
Switch IDs
404
Capability Vector
406
402
Min Release 408
F 410a
F 410m

Figure 4

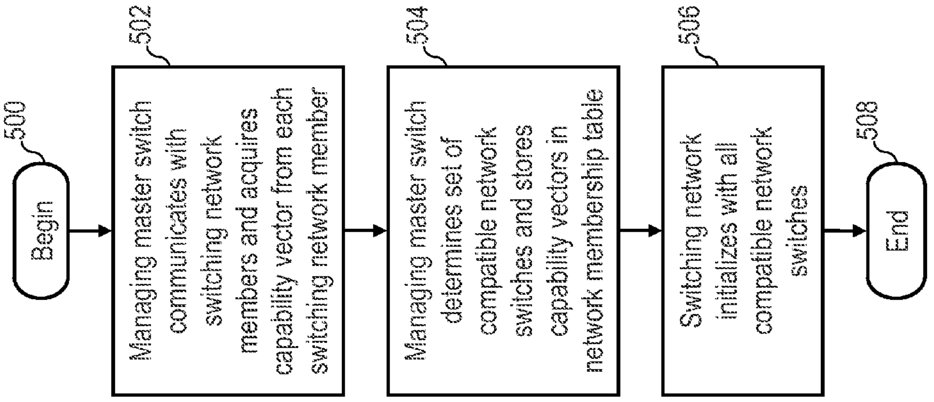

500
Begin
502
Managing master switch communicates with switching network members and acquires capability vector from each switching network member
504
Managing master switch determines set of compatible network switches and stores capability vectors in network membership table
506
Switching network initializes with all compatible network switches
508
End

Figure 5

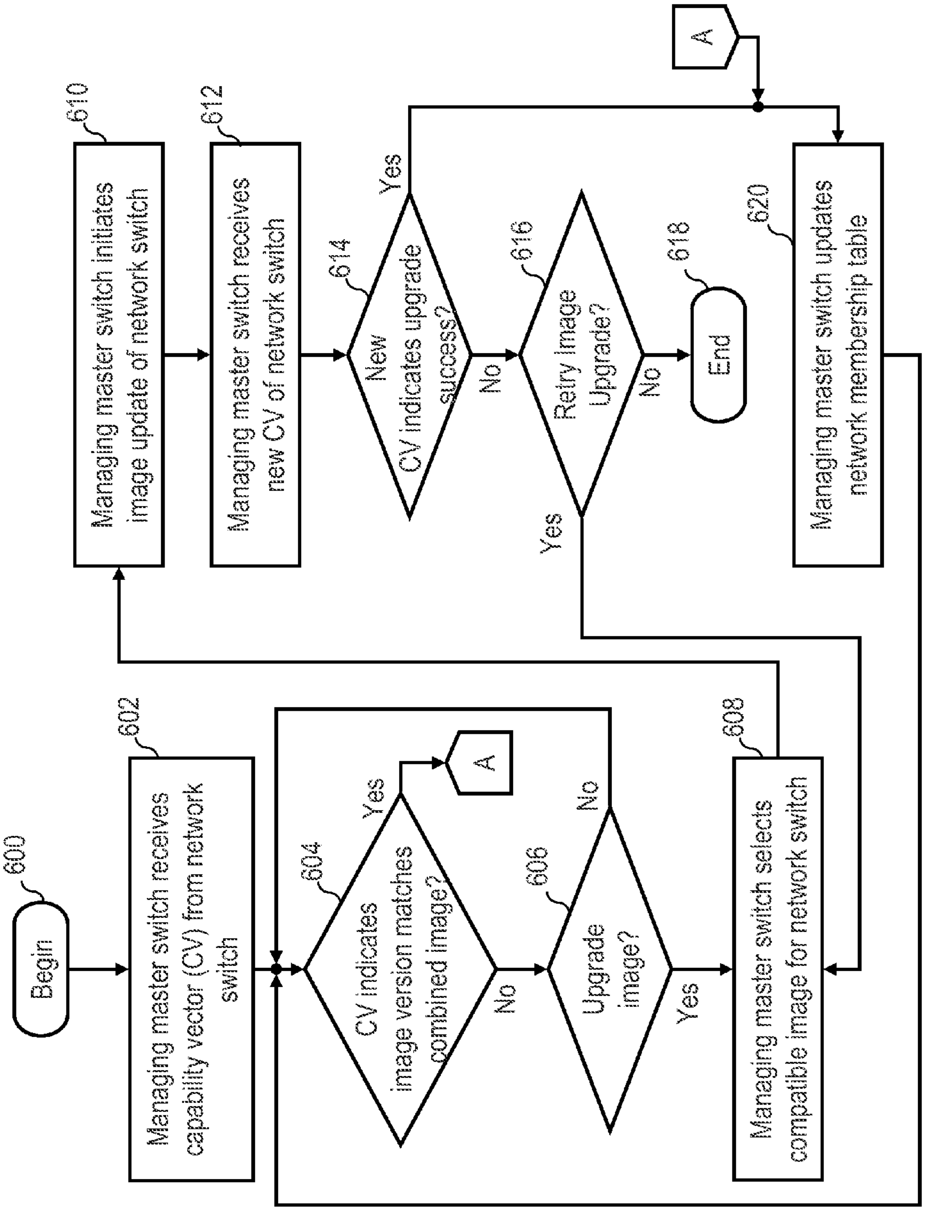

600
Begin
602
Managing master switch receives capability vector (CV) from network switch
604
CV indicates image version matches combined image?
Yes
A
No
606
Upgrade image?
No
Yes
608
Managing master switch selects compatible image for network switch
610
Managing master switch initiates image update of network switch
612
Managing master switch receives new CV of network switch
614
New CV indicates upgrade success?
Yes
No
616
Retry Image Upgrade?
Yes
No
618
End
A
620
Managing master switch updates network membership table

Figure 6

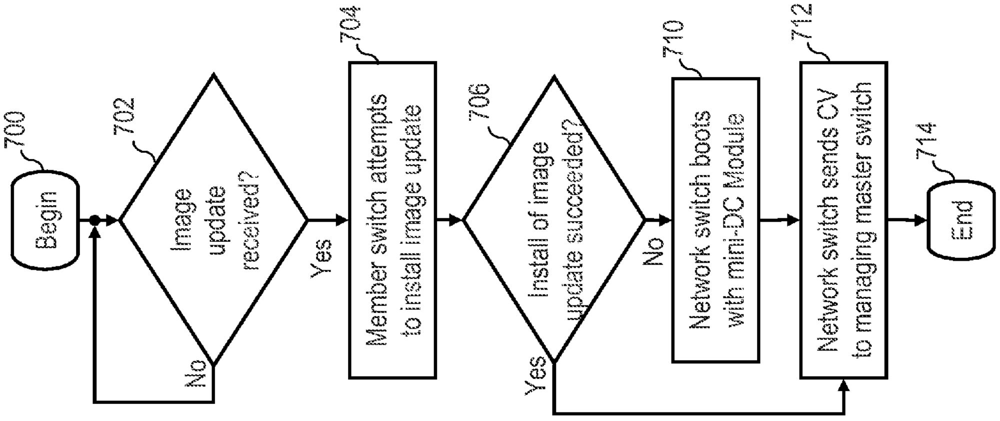

700
Begin
702
Image update received?
No
Yes
704
Member switch attempts to install image update
706
Install of image update succeeded?
Yes
No
710
Network switch boots with mini-DC Module
712
Network switch sends CV to managing master switch
714
End

Figure 7

UPDATING A SWITCH SOFTWARE IMAGE IN A DISTRIBUTED FABRIC PROTOCOL (DFP) SWITCHING NETWORK

BACKGROUND OF THE INVENTION

[0001] 1. Technical Field

[0002] The present invention relates in general to network communication and, in particular, to a system for managing the update of software images for computer networks.

[0003] 2. Description of the Related Art

[0004] As is known in the art, a communications network is a collection of terminals, links, and nodes connected together to enable communication between users of the terminals. Each terminal in the network must have a unique address so messages or connections can be routed to the correct recipients. Messages are generated by a sending or source terminal, then pass through the intermediate network of links and nodes until they arrive at the receiving or destination terminal. The intermediate network nodes handle these messages and route them down the correct network link towards their final destination terminal.

[0005] A large communications network typically includes a many switches, which operate independently at the management, control and data planes. Consequently, in conventional networks, each switch must be individually configured, since each switch implements its own means of handling data, control, and management traffic. Moreover, each switch forwards data, control, and management traffic independently of similar traffic handled by any other of the switches.

[0006] To maintain and/or improve network communication, software or firmware updates to installed network infrastructure (including network switches) are required occasionally. Further, network capacity and functionality is enhanced by installing new switches and/or replacing older switches.

SUMMARY OF THE INVENTION

[0007] In accordance with at least one embodiment, methods, systems and program products for updating system image(s) in a heterogeneous packet-switched network are disclosed.

[0008] In at least one embodiment of a switching network, the switching network has a plurality of switches including at least a switch and a managing master switch. At the managing master switch, a first capability vector (CV) is received from the switch. The managing master switch determines whether the first CV is compatible with at least a second CV in a network membership data structure that records CVs of multiple switches in the switching network. In response to detecting an incompatibility, the managing master switch initiates an image update to an image of the switch. In response to a failure of the image update at the switch, the switch boots utilizing a mini-DC module that reestablishes communication between the switch with the managing master switch and retries the image update.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. **1** is a high level block diagram of a data processing environment in accordance with one embodiment;

[0010] FIG. **2** is a high level block diagram of one embodiment of a distributed fabric protocol (DFP) switching network architecture that can be implemented within the data processing environment of FIG. **1**;

[0011] FIG. **3** is a more detailed block diagram of the DFP switching network architecture in FIG. **2** in accordance with one embodiment;

[0012] FIG. **4** is more detailed view of the network membership table in FIG. **3** in accordance with one embodiment;

[0013] FIG. **5** is a block diagram including additional details of the File Transfer Protocol (FTP) server in FIGS. **2-3** in accordance with one embodiment;

[0014] FIG. **6** is a high level logical flowchart of an exemplary process for initializing a DFP switching network in accordance with one embodiment; and

[0015] FIG. **7** is a high level logical flowchart of an exemplary process for managing the update of an incompatible member switch in accordance with one embodiment.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENT

[0016] Disclosed herein are methods, systems and program products for updating system image(s) in a heterogeneous packet-switched network, which may include switches from multiple vendors and/or switches with differing hardware and/or software. The update(s) of switch image(s) is/are preferably centrally managed by a managing master switch in the packet-switched network. By updating the system images of one or more switches in the packet-switched network, the managing master switch brings the packet-switched network into a consistent state in which all member switches of the packet-switched network are running the same or compatible switch images.

[0017] With reference now to the figures and with particular reference to FIG. **1**, there is illustrated a high level block diagram of an exemplary data processing environment **100** in accordance within one embodiment. As shown, data processing environment **100** includes a collection of resources **102**. Resources **102**, which may include various hosts, clients, switches, routers, storage, etc., are interconnected for communication and may be grouped (not shown) physically or virtually, in one or more public, private, community, public, or cloud networks or a combination thereof. In this manner, data processing environment **100** can offer infrastructure, platforms, software and/or services accessible to various client devices **110**, such as personal (e.g., desktop, laptop, netbook, tablet or handheld) computers **110***a*, smart phones **110***b*, server computer systems **110***c* and consumer electronics, such as media players (e.g., set top boxes, digital versatile disk (DVD) players, or digital video recorders (DVRs)) **110***d*. It should be understood that the types of client devices **110** shown in FIG. **1** are illustrative only and that client devices **110** can be any type of electronic device capable of communicating with and accessing resources **102** via a packet network.

[0018] Referring now to FIG. **2**, there is illustrated a high level block diagram of an exemplary distributed fabric protocol (DFP) switching network architecture that may be implemented within resources **102** in accordance with one embodiment. In the illustrated exemplary embodiment, resources **102** include a plurality of physical and/or virtual network switches forming a DFP switching network **200**. In contrast to conventional network environments in which each switch implements independent management, control, and data planes, DFP switching network **200** implements unified management, control and data planes, enabling all the constituent member switches to be viewed as a unified virtualized switch, thus simplifying deployment, configuration, and management of the network fabric, as discussed further herein.

[0019] DFP switching network **200** includes two or more tiers of switches, which in the instant embodiment includes a lower tier having a plurality of follower switches, including follower switches **202***a*-**202***d,* and an upper tier having a plurality of master switches, including master switches **204***a*-**204***b*. In an embodiment with two tiers as shown, a port of each master switch **204** is directly connected by one of inter-tier links **206** to one of the ports of each follower switch **202**, and a port of each master switch **204** is coupled directly or indirectly to a port at least one other master switch **204** by a master link **208**. A port of each master switch **204***a*-**204***b* and follower switch **202***a*-**202***d* is coupled directly or indirectly to a port of File Transfer Protocol (FTP) server **209** by server-switch links **211** and **213**. When such distinctions are relevant, ports supporting switch-to-switch communication via inter-tier links **206** are referred to herein as "inter-switch ports," and other ports (e.g., of follower switch **202***a*-**202***d* and FTP server **209**) are referred to as "data ports."

[0020] In a preferred embodiment, follower switches **202** are configured to operate on the data plane in a pass-through mode, meaning that all ingress data traffic received at data ports **210** of follower switches **202** (e.g., from host platforms) is forwarded by follower switches **202** via inter-switch ports and inter-tier links **206** to one of master switches **204**. Master switches **204** in turn serve as the fabric for the data traffic (hence the notion of a distributed fabric) and implement all packet switching and routing for the data traffic. With this arrangement data traffic may be forwarded, for example, in the first exemplary flow indicated by arrows **212***a*-**212***d* and the second exemplary flow indicated by arrows **214***a*-**214***e*.

[0021] As will be appreciated, the centralization of data plane switching and routing for follower switches **202** in master switches **204** implies that master switches **204** have knowledge of the ingress data ports of follower switches **202** on which data traffic was received. In a preferred embodiment, switch-to-switch communication via links **206**, **208** employs a Layer 2 protocol, such as the Inter-Switch Link (ISL) protocol developed by Cisco Corporation or IEEE 802.1 QnQ, that utilizes explicit tagging to establish multiple Layer 2 virtual local area networks (VLANs) over DFP switching network **200**. Each follower switch **202** preferably applies VLAN tags (also known as service tags (S-tags)) to data frames to communicate to the recipient master switch **204** the ingress data port **210** on the follower switch **202** on which the data frame was received. In alternative embodiments, the ingress data port can be communicated by another identifier, for example, a MAC-in-MAC header, a unique MAC address, an IP-in-IP header, etc. As discussed further below, each data port **210** on each follower switch **202** has a corresponding virtual port (or vport) on each master switch **204**, and data frames ingressing on the data port **210** of a follower switch **202** are handled as if ingressing on the corresponding vport of the recipient master switch **204**.

[0022] Management of DFP switching network is preferably implemented by a single master switch **204**, for example, master switch **204***a,* herein referred to as the managing master switch. In event of a failure of managing master switch **204***a* (as detected by the loss of heartbeat messaging with managing master switch **204***a* via master link **208**), another master switch **204***b*, which may be predetermined or elected from among the remaining operative master switches **204**, preferably automatically assumes the role of the managing master switch **204***a* and implements centralized management and control of the DFP switching network **200**. In preparation for a failover operation, managing master switch **204***a* pushes its image information to other master switches **204**, thus enabling seamless failover.

[0023] With reference now to FIG. **3**, there is illustrated a more detailed block diagram of features of exemplary DFP switching network **200** of FIG. **2** that may be employed to manage image updates in a homogenous or heterogeneous packet-switched network of member switches.

[0024] To switch data frames, each member switch **202**, **204** within DFP switching network **200** generally includes a plurality of data ports, a switching fabric and a switch controller, which can be implemented with one or more centralized or distributed, special-purpose or general-purpose processing elements or logic devices that implement control entirely in hardware, or more commonly, through the execution of firmware and/or software by a processing element. In master switches **204**, the switch controller **302** includes a management module **304** for managing DFP network **200**. In a preferred embodiment, only the management module **304** of the managing master switch (i.e., managing master switch **204***a* or another master switch **204***b* operating in its stead) is operative at any given time.

[0025] Management module **304** preferably includes a management interface **306**, for example, an XML or HTML interface accessible to an administrator stationed at a network-connected administrator console (e.g., one of clients **110***a*-**110***c*) in response to login and entry of administrative credentials. Management module **304**, which permits the administrator to centrally manage and control all member switches of DFP switching network **200**, preferably presents via management interface **306** a global view of all ports residing on all switches (e.g., master switches **204** and follower switches **202**) in a DFP switching network **200**.

[0026] As further shown in FIG. **3**, management module **304** preferably additionally includes a network membership table **400**, an exemplary embodiment of which is depicted in greater detail in FIG. **4**. As shown in FIG. **4**, network membership table **400** includes n entries **402** each corresponding to a different switch system configuration, as defined by its hardware features. Each entry **402** includes a switch ID field **404** identifying the member switch(es) in DFP switching network **200** having that switch system configuration, as well as a capability vector **406** characterizing the switch system configuration. In the depicted exemplary embodiment, each capability vector **406** includes a minimum release field **408** for identifying the minimum software release version that can be utilized to run that switch system configuration. In addition, each capability vector **406** includes a plurality of feature fields **410***a*-**410***m* each indicating for a respective one of a set of multiple possible hardware or software switch features whether the feature is required, optional or not supported in the switch system configuration.

[0027] Referring again to FIG. **3**, in the depicted embodiment, DFP network system **200** is heterogeneous in that, while master switches **204** all operate under a common image (i.e., master image I **308**), follower switches **202***a* and **202***b* operate under different images (i.e., follower image II **310** and follower image III **312**, respectively). To efficiently manage and deploy these different switch images, managing master switch **204***a* preferably builds and maintains a combined image **314** of all switch images currently supported in DFP switching network **200**. Due to the potentially large number of different switch images that can be supported, combined image **314** can have a large storage footprint. For this reason, combined image **314** is preferably stored and maintained at a remote file storage location, such as File Transfer Protocol (FTP) server **209**. However, in at least some embodiments, combined image **314** can alternatively or additionally be stored within RAM disk **316** of managing master switch **204***a* (as depicted by dashed lines in FIG. **3**).

[0028] As further illustrated in FIG. **3**, each member switch **202**, **204** and combined image **314** preferably includes a mini-DC module **318**. Mini-DC module **318** is a very reliable, compact and lightweight image capable of booting the member switches **202**, **204** of DFP switching network **200** and providing a minimal set of functionality, including communication with master switches **204**. Because of its more limited feature set, mini-DC module **318** has a smaller storage footprint than the regular images **308**, **310** and **312** utilized to operate switches **202**, **204**. In event of a crash of a member switch **202** or **204**, for example, when updating its image, mini-DC module **318** will automatically boot the crashed member switch to an operational state in which communication with master switches **204** is restored, enabling automated recovery from the crash and retry of the image update.

[0029] With reference now to FIG. **5**, there is illustrated a high level logical flowchart of an exemplary process for initializing a DFP switching network in accordance with one embodiment. For convenience, the process of FIG. **5** is described with reference to DFP switching network **200** of FIGS. **2-3**. As with the other logical flowcharts illustrated herein, steps are illustrated in logical rather than strictly chronological order, and at least some steps can be performed in a different order than illustrated or concurrently.

[0030] The process of FIG. **5** begins at block **500** and then proceeds to block **502**, which depicts managing master switch **204***a* communicating with all network switches with which it is directly connected via server-switch links **211** and **213** and acquiring the capability of each such network switch (block **602**). For example, in one embodiment, managing master switch **204***a* may first learn the switches to which it is connected via an automated switch discovery protocol implemented by management module **304**. Based upon the discovered pool of directly connected network switches, management module **304** of managing master switch **204***a* issues a capability vector request to each connected switch, which results in each of the discovered switches directly connected to managing master switch **204***a* reporting a capability vector (e.g., of the form shown in FIG. **4**) to managing master switch **204***a.* Alternatively or additionally, each network switch connected to managing master switch **204***a* may autonomously report its capability vector to managing master switch **204***a,* for example, in conjunction with a "join" request transmitted to managing master switch **204***a* following boot of an image on the network switch.

[0031] In response to acquiring the capability vectors of the network switches to which it is connected, managing master switch **204***a* determines a set of the network switches running compatible images and records the identities of the compatible network switches and their capability vectors in network membership table **400** (block **504**). In addition, managing master switch **204***a* initializes the identified set of compatible network switches as member switches **202**, **204** of DFP switching network (block **506**). Switches running under incompatible images, if any, are not permitted to immediately join DFP switching network **200**, and while capable of communication with master switches **204**, remain under independent management and control until these excluded switches are updated to run under a compatible image, as described below with reference to FIG. **6**. The process of FIG. **5** thereafter ends at block **508**.

[0032] With reference now to FIG. **6**, there is depicted a high level logical flowchart of an exemplary process for managing the update of a network switch in accordance with one embodiment. For ease of understanding, the process will be described with reference to DFP switching network **200** as depicted in FIGS. **2-3**.

[0033] The illustrated process begins at block **600** and then proceeds to block **602**, which depicts managing master switch **204***a* of DFP switching network **200** receiving a capability vector from a network switch to which it is directly connected by an inter-switch link **211**, **213**. The capability vector preferably reports the current version of the image running on the network switch. In response, managing master switch **204***a* determines, via its management module **304**, whether the image version reported by the network member is the same as that contained in the combined image **314**. If the image versions match, no image update is necessary, and the process proceeds through page connector A to block **620**, which is described below.

[0034] If, however, managing master switch **204***a* detects a difference in image versions at block **604**, managing master switch **204***a* determines whether the difference in image versions merits an update of the member switch's image (block **606**). In this regard, it should be noted that it is not always necessary that managing master switch **204***a* and member switches **202**, **204** have the same image version. For example, a follower switch **202** may have a higher release number than managing master switch **204***a* and still share the same capability vector. For this reason, in one embodiment, decision block **606** represents a comparison between the capability vector acquired from the network switch with the capability of the corresponding entry **402** in membership table **400** to determine whether difference in versions causes an incompatibility in capabilities between the images.

[0035] Incompatibility between an installed image and a more recent image within combined image **314** can arise for a number of reasons. For example, one source of incompatibility is a hardware or software update of some, but not all of member switches **202**, **204**. Such an update can lead to an installed image version not supporting a feature that the image version in combined image **314** requires. Other causes of incompatibility include, but are not limited to, protocol updates and changes in management and control data. It should therefore be appreciated that incompatibilities between switch images are not limited to those caused by data plane changes, but can be caused by changes along any of the network planes, including the management plane, control plane, and/or data plane.

[0036] If no incompatibility is detected at block **606**, the process can return to block **604**, and no switch image update is required. However, if an incompatibility is detected at block **606** (or if managing master switch **204***a* optionally determines to update the image despite its compatibility), managing master switch **204***a* automatically selects a compatible image version to which the network switch will be updated (block **608**). Typically, managing master switch **204***a* initially searches FTP server **209** to locate a compatible image with which to perform the image update. If FTP server **209** is not configured or is unavailable, managing master switch **204***a* searches its own local file system (e.g., RAM disk **316**) to locate the compatible image.

[0037] Upon locating the compatible image, managing master switch **204***a* initiates the update of the incompatible network switch (block **610**). In one preferred embodiment, managing master switch **204***a* communicates a push request to FTP server **209** to push the updated switch image to the incompatible network switch. Alternatively, managing master switch **204***a* can communicate a download command to the incompatible member switch, which would in turn download the image directly from FTP server **209**. In another alternative embodiment, managing master switch **204***a* may push the compatible image from its local file system (e.g., RAM disk **316**).

[0038] Next, at block **612**, managing master switch **204***a* receives a revised capability vector from the previously incompatible network switch and updates network membership table **400**. As depicted in decision block **614** managing master switch **204***a* determines whether the newly received capability vector indicates a successful update of the previously incompatible network switch. If the update was successful, that is, the compatibility vector reported the image selected by managing master switch **204***a* at block **608**, the process passes to block **620**, which depicts managing master switch **204***a* updating network membership table **400** with the switch ID and feature information from the compatibility vector. The process thereafter returns to decision block **604**. However, if managing master switch **204***a* determines at block **614** that the update was not successful, managing master switch **204***a* decides at block **616** whether to retry the update to the image of the incompatible network switch. If so, the process returns to block **608**, which depicts managing master switch **204***a* selecting a possibly different compatible image with which to update the network switch. However, if managing master switch **204***a* does not elect to retry the image update, the process terminates at block **618**.

[0039] With reference now to FIG. **7**, there is illustrated a high level logical flowchart of an exemplary process for recovering network communication between a network switch and managing master switch **204***a* in the event of an image update failure in accordance with one embodiment. For ease of understanding, the process of FIG. **7** is also described with reference to DFP switching network **200** of FIG. **2**.

[0040] The depicted process begins at block **700** and thereafter proceeds to block **702**, which depicts a network switch that is directly connected to managing master switch **204***a* determining whether an image update has been received (e.g., due to managing master switch **204***a* pushing an updated image or commanding the network switch to pull the updated image). If not, the process iterates at block **702**. If, however, an image update has been received, the network switch attempts to install the image update (block **704**). As indicated at block **706**, if the installation is successful, the process passes to block **712**, which depicts the network switch transmitting a new capability vector to managing master switch **204***a*, as discussed above with reference to block **612** of FIG. **6**. Thereafter, the image update process of FIG. **7** ends at block **714**.

[0041] Returning to block **706**, if the installation of the updated image fails, meaning that the network switch has crashed, the network switch boots with mini-DC module **318** (block **710**). Mini-DC module **318** is pre-loaded when the network switch is first initialized and serves as a backup/default OS that loads in the event of an image update failure. While mini-DC module **318** contains all the basic hardware and configuration-related information, mini-DC module **318** has a fixed capability vector, which the network switch reports to managing master switch **204***a* at block **712**. In response to receipt of this fixed capability vector, managing master switch **204***a* will discover an incompatibility when the fixed capability vector is compared to that listed in network membership table **400**, which will trigger managing master switch **204***a* initiating an update to the incompatible image (as discussed above with reference to FIG. **6**). Following block **712**, the process depicted in FIG. **7** terminates at block **712**.

[0042] As has been described, a switching network has a plurality of switches including at least a switch and a managing master switch. At the managing master switch, a first capability vector (CV) is received from the switch. The managing master switch determines whether the first CV is compatible with at least a second CV in a network membership data structure that records CVs of multiple switches in the switching network. In response to detecting an incompatibility, the managing master switch initiates an image update to an image of the switch. In response to a failure of the image update at the switch, the switch boots utilizing a mini-DC module that reestablishes communication between the switch with the managing master switch and retries the image update.

[0043] While the present invention has been particularly shown as described with reference to one or more preferred embodiments, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention. For example, although aspects have been described with respect to one or more machines (e.g., FTP server and/or network switches) executing program code (e.g., software, firmware or a combination thereof) that direct the functions described herein, it should be understood that embodiments may alternatively be implemented as a program product including a tangible machine-readable storage medium or storage device (e.g., an optical storage medium, memory storage medium, disk storage medium, etc.) storing program code that can be processed by a machine to cause the machine to perform one or more of the described functions.

1-8. (canceled)

9. A program product, comprising:

a machine-readable storage device; and

program code stored within the machine-readable storage medium for managing an image update in a switching network including a managing master switch and a plurality of switches each coupled to the managing master switch by a respective one of multiple links, wherein the program code, when processed by a machine, causes the machine to perform:

receiving a first capability vector (CV) from the switch at the managing master switch;

the managing master switch determining whether the first CV is compatible with at least a second CV in a network membership data structure that records CVs of multiple switches in the switching network;

in response to detecting an incompatibility, the managing master switch initiating an image update to an image of the switch; and in response to a failure of the image update at the switch, the managing master switch reestablishing communication with the switch as booted under a mini-DC module and retrying the image update of the switch.

10. The program product of claim **9**, wherein:

the incompatibility is in at least one of a control plane and a management plane of the switch.

11. The program product of claim **9**, wherein:

each of the first and second compatibility vectors (CV) indicates whether each feature in a feature set of multiple switch features is supported, as well as a minimum image version corresponding to the feature set.

12. The program product of claim **9**, wherein:

the switching network is a heterogeneous packet-switched network; and

the image of the switch differs from an image of the managing master switch.

13. The program product of claim **9**, wherein the program code further causes the machine to perform:

the managing master switch detecting failure of the image update in response to receipt of a third CV corresponding to a feature set of the switch operating under the mini-DC module.

14. The program product of claim **9**, wherein:

the managing master switch maintains a combined image including a respective image for each of multiple switches in the switching network; and

the initiating the image update includes initiating the image update to the switch from the combined image.

15. The program product of claim **9**, wherein:

the managing master switch provides common data, control and management planes for member switches in the switching network; and

the program code further causes the managing master switch to join the switch to the switching network in response to success of the image update at the switch.

16. A switch for serving as a managing master switch for a switching network including the master switch and a plurality of other switches, the switch comprising:

a plurality of ports by which the managing master switch can connect to the plurality of switches by multiple links;

a switch fabric permitting data exchange between the plurality of ports; and

a switch controller configured to perform:

receiving a first capability vector (CV) from another switch in the switching network;

determining whether the first CV is compatible with at least a second CV in a network membership data structure that records CVs of multiple switches in the switching network;

in response to detecting an incompatibility, initiating an image update to an image of said another switch; and

in response to a failure of the image update at the switch, reestablishing communication with said another switch as booted under a mini-DC module and retrying the image update of said another switch.

17. The switch of claim **16**, wherein:

the incompatibility is in at least one of a control plane and a management plane of the switch.

18. The switch of claim **16**, wherein:

each of the first and second compatibility vectors (CV) indicates whether each feature in a feature set of multiple switch features is supported, as well as a minimum image version corresponding to the feature set.

19. The switch of claim **16**, wherein:

the switching network is a heterogeneous packet-switched network; and

the image of said another switch differs from an image of the managing master switch.

20. The switch of claim **16**, wherein the controller is further configured to perform:

detecting failure of the image update in response to receipt of a third CV corresponding to a feature set of the switch operating under the mini-DC module.

21. The switch of claim **16**, wherein:

the managing master switch maintains a combined image including a respective image for each of multiple switches in the switching network; and

the initiating the image update includes initiating the image update to said another switch from the combined image.

22. The switch of claim **16**, wherein:

the managing master switch provides common data, control and management planes for member switches in the switching network; and

the switch controller is further configured to perform joining said another switch to the switching network in response to success of the image update at said another switch.

23. A switching network comprising:

the switch of claim **16**; and

the plurality of switches connected to ports of the switch.

24. The switching network of claim **23**, wherein:

the image of said another switch is a first image;

the image update is a first image update;

said another switch includes:

a plurality of ports by which said another switch can connect to others of the plurality of switches;

a switch fabric permitting data exchange between the plurality of ports; and

a switch controller configured to perform:

in response to receipt of the first image update, attempting to update said another switch from the first image to a second image; and

in response to a failure of the first image update at said another switch, automatically booting said another switch under a mini-DC module, reestablishing communication with the managing master switch, and again attempting to update said another switch in response to receipt of a second image update.

25. A switch for a switching network including a plurality of switches, the switch comprising:

a plurality of ports by which the switch can connect to others of the plurality of switches;

a switch fabric permitting data exchange between the plurality of ports; and

a switch controller configured to perform:

receiving a first image update intended to update the switch from a first image to a second image;

attempting to update the switch from the first image to the second image; and

in response to a failure of the image update at the switch, automatically booting the switch under a mini-DC module, reestablishing communication with the switching network, and retrying an update to the switch in response to receipt of a second image update.

26. (canceled)

27. A program product, comprising:

a machine-readable storage device; and

program code stored within the machine-readable storage medium for updating an image of a switch in a switching network including a plurality of switches, wherein the program code, when processed by a machine, causes the machine to perform:

the switch receiving a first image update intended to update the switch from a first image to a second image;

attempting to update the switch from the first image to the second image; and

in response to a failure of the image update at the switch, automatically booting the switch under a mini-DC module, reestablishing communication with the switching network, and retrying an update to the switch in response to receipt of a second image update.

* * * * *