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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 March 2012 (22.03.2012)

PCT

(10) International Publication Number
WO 2012/037518 A1

(51) International Patent Classification:
H04L 12/24 (2006.01) *H04L 12/56* (2006.01)
H04L 29/08 (2006.01)

(21) International Application Number:
PCT/US2011/052029

(22) International Filing Date:
16 September 2011 (16.09.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

61/384,228	17 September 2010 (17.09.2010)	US
61/484,390	10 May 2011 (10.05.2011)	US
61/493,347	3 June 2011 (03.06.2011)	US
61/493,330	3 June 2011 (03.06.2011)	US
61/498,329	17 June 2011 (17.06.2011)	US

(71) Applicant *(for all designated States except US)*: **ORACLE INTERNATIONAL CORPORATION** [US/US]; 500 Oracle Parkway, M/S 5op7, Redwood Shores, California 94065 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **JOHNSEN, Bjorn-Dag** [NO/NO]; Vilberggrenda 9, N-0687 Oslo (NO). **HOLEN, Line** [NO/NO]; Vitasen 17, N-1900 Fetsund (NO). **MOXNES, Dag Georg** [NO/NO]; Leirskallbakken 25, N-1164 Oslo (NO).

(74) Agents: **MEYER, Sheldon, R.** et al.; Fliesler Meyer LLP, 650 California Street, Fourteenth Floor, San Francisco, California 94108 (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, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, 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:

— *with international search report (Art. 21(3))*

[Continued on next page]

(54) **Title**: SYSTEM AND METHOD FOR FACILITATING PROTECTION AGAINST RUN-AWAY SUBNET MANAGER INSTANCES IN A MIDDLEWARE MACHINE ENVIRONMENT

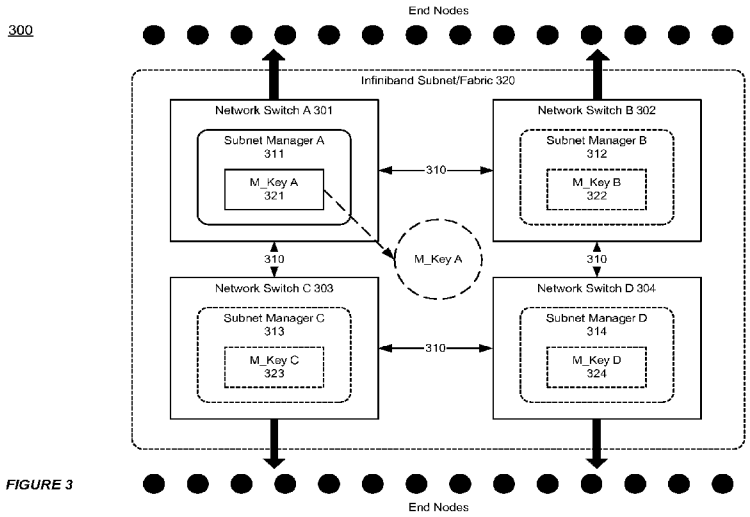

FIGURE 3

(57) **Abstract**: A system and method can support a middleware machine environment. The middleware machine environment can include a set of subnet manager instances that reside on one or more nodes in the middleware machine environment and cooperate to provide a highly available subnet manager service within a subnet, wherein each said subnet manager instance is associated with a different private secure key. The set of subnet manager instances can negotiate with each other and elect a master subnet manager, which is responsible for configuring and managing the middleware machine environment using the private secure key that is associated with the master subnet manager. The subnet is reconfigured to be associated with a different private secure key, when a new subnet manager instance is elected as the master subnet manager. An old master subnet manager can be automatically prevented from resuming normal operations as the master subnet manager, in order to avoid undesired consequence such as a "split brain" scenario.

WO 2012/037518 A1

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

SYSTEM AND METHOD FOR FACILITATING PROTECTION AGAINST RUN-AWAY SUBNET MANAGER INSTANCES IN A MIDDLEWARE MACHINE ENVIRONMENT

Field of Invention:

[0001] The present invention is generally related to computer systems and software such as middleware, and is particularly related to supporting a middleware machine environment.

Background:

[0002] Infiniband (IB) Architecture is a communications and management infrastructure that supports both I/O and interprocessor communications for one or more computer systems. An IB Architecture system can scale from a small server with a few processors and a few I/O devices to a massively parallel installation with hundreds of processors and thousands of I/O devices.

[0003] The IB Architecture defines a switched communications fabric allowing many devices to concurrently communicate with high bandwidth and low latency in a protected, remotely managed environment. An end node can communicate with over multiple IB Architecture ports and can utilize multiple paths through the IB Architecture fabric. A multiplicity of IB Architecture ports and paths through the network are provided for both fault tolerance and increased data transfer bandwidth.

[0004] These are the generally areas that embodiments of the invention are intended to address.

Summary:

[0005] Described herein is a system and method for supporting a middleware machine environment. The middleware machine environment can include a set of subnet manager instances that reside on one or more nodes in the middleware machine environment and cooperate to provide a highly available subnet manager service within a subnet, wherein each said subnet manager instance is associated with a different private secure key. The set of subnet manager instances can negotiate with each other and elect a master subnet manager, which is responsible for configuring and managing the middleware machine environment using the private secure key that is associated with the master subnet manager. The subnet is reconfigured to be associated with a different private secure key, when a new subnet manager

instance is elected as the master subnet manager. An old master subnet manager can be automatically prevented from resuming normal operations as the master subnet manager, in order to avoid undesired consequence such as a "split brain" scenario.

[0006] In one aspect, provided is a subnet manager within a subnet in a middleware machine environment, comprising: an associating module configured to associate a different private secure key with the subnet manager; and a negotiating module configured to allow the subnet manager to negotiate with other subnet managers within the subnet and elect a master subnet manager, which is responsible for configuring and managing the middleware machine environment using the private secure key that is associated with the master subnet manager.

[0007] In some embodiments, the subnet is an Infiniband (IB) subnet.

[0008] In some embodiments, the subnet manager further comprises a communication module configured to allow the subnet manager to communicate with other subnet managers within the subnet using an in-band communication protocol.

[0009] In some embodiments, the subnet can be divided into a dynamic set of resource domains implemented by subnet partitions.

[00010] In some embodiments, the subnet manager further comprises an initialization module configured to, responsive to the subnet manager being elected as the master subnet manager, initialize the master subnet manager using a default patitioning policy when no partitioning policy is specified.

[00011] In some embodiments, the private secure key is a M_key that is a 64 bit secret value that is known only to authorized entities in the subnet.

[00012] In some embodiments, when a subnet management agent (SMA) associated with a port in the subnet is configured with a M_Key value, an in-band request needs to specify the M_Key value in order to change a state associated with the port.

[00013] In some embodiments, the subnet is reconfigured to be associated with a different private secure key, when a different subnet manager is elected as the master subnet manager.

[00014] In some embodiments, each different private secure key is defined in a different range that is known to the other subnet manager within the subnet.

[00015] In some embodiments, the subnet manager further comprises a determining module configured to dynamically determine which private secure key in a defined range is in use depending on which subnet manager is currently the master subnet manager.

[00016] In some embodiments, an old master subnet manager is automatically prevented from resuming normal operations as a master subnet manager after a new master subnet manager is elected in order to prevent a split brain scenario.

[00017] In some embodiments, the master subnet manager can determine that a connection to a new subnet manager is unintentional since the new subnet manager can not recognize the private secure key used in the subnet, and the new subnet manager is prevented to change any

state in the subnet.

[00018] In another aspect, provided is a network switch, comprising: one subnet manager according to one aspect of the disclosure; one or more external ports that are used to connect with an external network; and one or more internal ports that are used to connect with a plurality of host servers in the middleware machine environment.

[00019] In still another aspect, provided is a system for supporting a middleware machine environment, comprising one or more network switches according to another aspect of the disclosure.

[00020] In some embodiments, the system further comprises a separate storage system that connects with the plurality of host servers through said one or more network switches.

[00021] In some embodiments, the system further comprises one or more gateway that can be accessed by a guest.

[00022] In some embodiments, one or more updated subnet configuration policies can be applied in the subnet through one or more subnet managers that can recognize a private secure key associated with the updated subnet configuration policies, and other subnet managers within the subnet that can not recognize the private secure key can be left unaffected by the one or more updated subnet configuration policies.

[00023] In yet still another aspect, provided is a system for supporting a middleware machine environment, comprising: a set of subnet manager that cooperate to provide a highly available subnet manager service within a subnet, wherein each said subnet manager is associated with a different private secure key, and wherein the set of subnet manager can negotiate with each other and elect a master subnet manager, which is responsible for configuring and managing the middleware machine environment using the private secure key that is associated with the master subnet manager.

Brief Description of the Figures:

[00024] **Figure 1** shows an illustration of a middleware machine environment that uses an M_Key, in accordance with an embodiment of the invention.

[00025] **Figure 2** shows an illustration of a middleware machine environment that employs an explicit take-over scheme, in accordance with an embodiment of the invention.

[00026] **Figure 3** shows an illustration of a middleware machine environment that uses an M_Key, in accordance with an embodiment of the invention.

[00027] **Figure 4** illustrates an exemplary flow chart for setting up routing logic in an IB fabric using M_Key, in accordance with an embodiment of the invention.

[00028] **Figure 5** illustrates an exemplary flow chart for setting up routing logic in an IB fabric using M_Key, in accordance with an embodiment of the invention.

[00029] **Figure 6** shows an illustration of a middleware machine environment that supports an

explicit take-over scheme, in accordance with an embodiment of the invention.

[00030] **Figure 7** illustrates an exemplary flow chart for supporting an explicit take-over scheme in a middleware machine environment, in accordance with an embodiment of the invention.

[00031] **Figure 8** shows an illustration of a middleware machine environment that supports robust fencing of subnet manager instances that are not able to receive updated subnet configuration policies, in accordance with an embodiment of the invention.

[00032] **Figure 9** illustrates an exemplary flow chart for supporting robust fencing of subnet manager instances that are not able to receive updated subnet configuration policies in a middleware machine environment, in accordance with an embodiment of the invention.

[00033] **Figure 10** shows an illustration of a middleware machine environment with an accidental subnet merge, in accordance with an embodiment of the invention.

[00034] **Figure 11** illustrates an exemplary flow chart for guarding against negative effect of accidental subnet merge in a middleware machine environment, in accordance with an embodiment of the invention.

[00035] **Figure 12** is a functional block diagram of a subnet manager within a subnet in a middleware machine environment in accordance with some embodiments of the invention.

[00036] **Figure 13** is a functional block diagram of a network switch in accordance with some embodiments of the invention.

[00037] **Figure 14** is a functional block diagram of a system for supporting a middleware machine environment in accordance with some embodiments of the invention.

Detailed Description:

[00038] Described herein is a system and method for providing a middleware machine or similar platform. In accordance with an embodiment of the invention, the system comprises a combination of high performance hardware (e.g. 64-bit processor technology, high performance large memory, and redundant InfiniBand and Ethernet networking) together with an application server or middleware environment, such as WebLogic Suite, to provide a complete Java EE application server complex which includes a massively parallel in-memory grid, that can be provisioned quickly, and that can scale on demand. In accordance with an embodiment of the invention, the system can be deployed as a full, half, or quarter rack, or other configuration, that provides an application server grid, storage area network, and InfiniBand (IB) network. The middleware machine software can provide application server, middleware and other functionality such as, for example, WebLogic Server, JRockit or Hotspot JVM, Oracle Linux or Solaris, and Oracle VM. In accordance with an embodiment of the invention, the system can include a plurality of compute nodes, one or more IB switch gateways, and storage nodes or units, communicating with one another via an IB network. When implemented as a rack configuration,

unused portions of the rack can be left empty or occupied by fillers.

[00039] In accordance with an embodiment of the invention, referred to herein as "Sun Oracle Exalogic" or "Exalogic", the system is an easy-to-deploy solution for hosting middleware or application server software, such as the Oracle Middleware SW suite, or Weblogic. As described herein, in accordance with an embodiment the system is a "grid in a box" that comprises one or more servers, storage units, an IB fabric for storage networking, and all the other components required to host a middleware application. Significant performance can be delivered for all types of middleware applications by leveraging a massively parallel grid architecture using, e.g. Real Application Clusters and Exalogic Open storage. The system delivers improved performance with linear I/O scalability, is simple to use and manage, and delivers mission-critical availability and reliability.

[00040] **Figure 1** shows an illustration of an exemplary configuration for a middleware machine, in accordance with an embodiment of the invention. As shown in Figure 1, the middleware machine 100 uses a single rack configuration that includes two gateway network switches, or leaf network switches, 102 and 103 that connect to twenty-eight server nodes. Additionally, there can be different configurations for the middleware machine. For example, there can be a half rack configuration that contains a portion of the server nodes, and there can also be a multi-rack configuration that contains a large number of servers.

[00041] As shown in Figure 1, the server nodes can connect to the ports provided by the gateway network switches. As shown in Figure 1, each server machine can have connections to the two gateway network switches 102 and 103 separately. For example, the gateway network switch 102 connects to the port 1 of the servers 1-14 106 and the port 2 of the servers 15-28 107, and the gateway network switch 103 connects to the port 2 of the servers 1-14 108 and the port 1 of the servers 15-28 109.

[00042] In accordance with an embodiment of the invention, each gateway network switch can have multiple internal ports that are used to connect with different servers, and the gateway network switch can also have external ports that are used to connect with an external network, such as an existing data center service network.

[00043] In accordance with an embodiment of the invention, the middleware machine can include a separate storage system 110 that connects to the servers through the gateway network switches. Additionally, the middleware machine can include a spine network switch 101 that connects to the two gateway network switches 102 and 103. As shown in Figure 1, there can be optionally two links from the storage system to the spine network switch.

IB Fabric/Subnet

[00044] In accordance with an embodiment of the invention, an IB Fabric/Subnet in a middleware machine environment can contain a large number of physical hosts or servers,

switch instances and gateway instances that are interconnected in a fat-tree topology.

[00045] **Figure 2** shows an illustration of a middleware machine environment, in accordance with an embodiment of the invention. As shown in Figure 2, the middleware machine environment 200 includes an IB subnet or fabric 220 that connects with a plurality of end nodes. The IB subnet includes a plurality of subnet managers 211-214, each of which resides on a node such as one of a plurality of network switches 201-204. The subnet managers can communicate with each other using an in-band communication protocol 210, such as the Management Datagram (MAD) / Subnet Management Packet (SMP) based protocols or other protocol such as the Internet Protocol over IB (IPoIB).

[00046] In accordance with an embodiment of the invention, a single IP subnet can be constructed on the IB fabric allowing the switches to communicate securely among each other in the same IB fabric (i.e. full connectivity among all switches). The fabric based IP subnet can provide connectivity between any pair of switches when at least one route with operational links exists between the two switches. Recovery from link failures can be achieved if an alternative route exists by re-routing.

[00047] The management Ethernet interfaces of the switches can be connected to a single network providing IP level connectivity between all the switches. Each switch can be identified by two main IP addresses: one for the external management Ethernet and one for the fabric based IP subnet. Each switch can monitor connectivity to all other switches using both IP addresses, and can use either operational address for communication. Additionally, each switch can have a point-to-point IP link to each directly connected switch on the fabric. Hence, there can be at least one additional IP address.

[00048] IP routing setups allow a network switch to route traffic to another switch via an intermediate switch using a combination of the fabric IP subnet, the external management Ethernet network, and one or more fabric level point-to-point IP links between pairs of switches. IP routing allows external management access to a network switch to be routed via an external Ethernet port on the network switch, as well as through a dedicated routing service on the fabric.

[00049] The IB fabric includes multiple network switches with management Ethernet access to a management network. There is in-band physical connectivity between the switches in the fabric. In one example, there is at least one in-band route of one or more hops between each pair of switches, when the IB fabric is not degraded. Management nodes for the IB fabric include network switches and management hosts that are connected to the IB fabric.

[00050] A subnet manager can be accessed via any of its private IP addresses. The subnet manager can also be accessible via a floating IP address that is configured for the master subnet manager when the subnet manager takes on the role as a master subnet manager, and the subnet manager is un-configured when it is explicitly released from the role. A master IP address can be defined for both the external management network as well as for the fabric

based management IP network. No special master IP address needs to be defined for point-to-point IP links.

[00051] In accordance with an embodiment of the invention, each physical host can be virtualized using virtual machine based guests. There can be multiple guests existing concurrently per physical host, for example one guest per CPU core. Additionally, each physical host can have at least one dual-ported Host Channel Adapter (HCA), which can be virtualized and shared among guests, so that the fabric view of a virtualized HCA is a single dual-ported HCA just like a non-virtualized/shared HCA.

[00052] The IB fabric can be divided into a dynamic set of resource domains implemented by IB partitions. Each physical host and each gateway instance in an IB fabric can be a member of multiple partitions. Also, multiple guests on the same or different physical hosts can be members of the same or different partitions. The number of the IB partitions for an IB fabric may be limited by the P_key table size.

[00053] In accordance with an embodiment of the invention, a guest may open a set of virtual network interface cards (vNICs) on two or more gateway instances that are accessed directly from a vNIC driver in the guest. The guest can migrate between physical hosts while either retaining or having updated vNIC associates.

[00054] In accordance with an embodiment of the invention, switchs can start up in any order and can dynamically select a master subnet manager according to different negotiation protocols, for example an IB specified negotiation protocol. If no partitioning policy is specified, a default partitioning enabled policy can be used. Additionally, the management node partition and the fabric based management IP subnet can be established independently of any additional policy infomation and independently of whether the complete fabric policy is known by the master subnet manager. In order to allow fabric level configuration policy information to be synchronized using the fabric based IP subnet, the subnet manager can start up initially using the default partition policy. When fabric level synchronization has been achived, the partition configuration, which is current for the fabric, can be installed by the master subnet manager.

M_Key

[00055] In accordance with an embodiment of the invention, a set of subnet manager instances can cooperate to provide a highly available subnet manager service within the IB subnet using a private secure key. One example of such a private secure key is an M_Key that can be used to facilitate protection against undesired consequences of various network abnormalities in an IB fabric. In one embodiment, the M_Key is a 64 bit secret value that has the function of a password, and is known only to authorized entities in the IB fabric. When a Subnet Management Agent (SMA) associated with a port in the IB fabric is configured with an M_Key value, any in-band SMP request to change a state associated with the port has to specify the

correct M_Key value.

[00056] **Figure 3** shows an illustration of a middleware machine environment that uses an M_Key, in accordance with an embodiment of the invention. As shown in Figure 3, an IB subnet or fabric 320 in the middleware machine environment 300 includes a plurality of subnet managers 311-314. Each subnet manager resides on one of a plurality of network switches 301-304 and is associated with a different M_Key 321-324. The subnet managers can communicate with each other using an in-band communication protocol 310.

[00057] The subnet managers can negotiate with each other and elect a master subnet manager, which is responsible for configuring and managing the IB fabric in the the middleware machine environment. In the example as shown in Figure 3, subnet manager A 311 is elected as the master subnet manager. As a result, the M_Key A 321 that is associated with the subnet manager A is selected for configuring and managing the IB subnet. Additionally, each of the subnet managers B-D 312-314 as shown in Figure 3 can monitor subnet the manager A and prepare to take over as the master subnet manager when it is necessary.

[00058] **Figure 4** illustrates an exemplary flow chart for setting up routing logic in an IB fabric using M_Key, in accordance with an embodiment of the invention. As shown in Figure 4, at step 401, the system can associate a different private secure key, such as an M_Key, with each subnet manager instance in the IB fabric. The subnet manager instances cooperate to provide a highly available subnet manager service within an IB subnet. Then, the subnet manager instances can negotiate with each other and elect a master subnet manager at step 402. Finally, at step 403, the master subnet manager can configure and manage the middleware machine environment using the private secure key that is associated with the master subnet manager.

[00059] **Figure 5** illustrates an exemplary flow chart for setting up routing logic in an IB fabric using M_Key, in accordance with an embodiment of the invention. As shown in Figure 5, at step 501, a master subnet manager can first try to discover the complete connected port topology in a physical IB subnet using SMP request packets. Then, at step 502, the master subnet manager can determine what ports it is allowed to control using M_Keys and/or explicit node/port-list configuration information defined via out-of-band policy input. Finally, at step 503, the master subnet manager can set up routing logic in the IB fabric based on the discovered port topology before allowing normal data packets communication in the IB fabric.

Protection against "run-away" subnet manager instances

[00060] In accordance with one embodiment of the invention, each subnet manager instance can be associated with a particular M_Key value/range that is known to the other subnet manager instances in the same IB fabric. The set of subnet manager instances in the IB fabric can dynamically determine which M_Key in a defined range is in use depending on which subnet manager is currently the master subnet manager.

[00061] **Figure 6** shows an illustration of a middleware machine environment that supports an explicit take-over scheme, in accordance with an embodiment of the invention. As shown in Figure 6, the subnet manager A 611 is suspended or failed, for example, the subnet manager A has been prevented from performing normal operations and handshakes with standby subnet manager instance for a period of time, and the subnet manager C 613 takes over the IB subnet 620 subsequently. Accordingly, the M_Key C 623 that is associated with the subnet manager C replaces the old M_Key, M_Key A, and is used in the IB fabric 620.

[00062] In accordance with one embodiment of the invention, an old "run-away" master subnet manager, for example the subnet manager A as shown in Figure 6, can be interrupted and/or prevented from running for a significant period of time (e.g. due to a scheduling problem on the platform it is running on). When the subnet manager A connects back to the IB fabric, the subnet manager A can realize that a new master subnet manager has been elected for the IB subnet and a new M_Key is in use in the IB fabric. Then, the system can automatically prevent the subnet manager A from resuming normal operations as the master subnet manager, in order to avoid undesired consequence such as a "split brain" scenario. The "split brain" scenario can happen when the old master subnet manager performs state updates on the nodes and ports in the subnet in a way that conflicts with concurrent updates from the new master subnet manager.

[00063] **Figure 7** illustrates an exemplary flow chart for supporting an explicit take-over scheme in a middleware machine environment, in accordance with an embodiment of the invention. As shown in Figure 7, at step 701, the system can detect an interruption of an old master subnet manager for a significant period of time. Then, at step 702, the set of subnet manager instances can elect a new master subnet manager. The IB fabric can replace the old M_Key associated with the old master subnet manger with a new M_Key associated with the new master subnet manager at step 703. Finally, at step 704, the IB fabric can automatically prevent the old subnet manager from resuming normal operations as the master subnet manager, in order to avoid undesired consequence such as a "split brain" scenario.

Robust fencing of subnet manager instances that are not able to receive updated subnet configuration policies

[00064] In accordance with an embodiment of the invention, new policies, such as subnet configuration policies, can be applied in an IB subnet by implementing a new set of subnet manager instances or updating an old set of subnet manager instances, without depending on the operational states or reachability of the existing subnet manager instances.

[00065] **Figure 8** shows an illustration of a middleware machine environment that supports robust fencing of subnet manager instances that are not able to receive updated subnet configuration policies, in accordance with an embodiment of the invention. As shown in Figure 8, updated subnet configuration policies 809, can be applied in an IB subnet 820 via a subnet

manager A 811 and a subnet manager C 813. The subnet manager A and the subnet manager C can have M_Key value ranges that are only recognizable among themselves.

[00066] In accordance with an embodiment of the invention, the system can ensure robust fencing of subnet manager instances that are not able to receive updated subnet configuration policies in a coordinated manner, such as in an ACID compliant manner. In the example as shown in Figure 8, the existing subnet managers in the IB subnet, the subnet manager B 812 and the subnet manager D 814, can not recognize the M_Key values associated with the updated subnet configuration policies. Therefore, the subnet manager B and the subnet manager D are not able to receive the updated subnet configuration policies. As a result, the updated configuration policy can be applied in the IB subnet through the subnet managers A and C, leaving the subnet managers B and D unaffected by the new policies. Further, neither subnet managers B or D can change the state of the subnet based on any old and potentially stale configuration policy stored locally on the corresponding nodes.

[00067] **Figure 9** illustrates an exemplary flow chart for supporting robust fencing of subnet manager instances that are not able to receive updated subnet configuration policies in a middleware machine environment, in accordance with an embodiment of the invention. As shown in Figure 9, at step 901, a user can provide an updated subnet configuration policy for the IB fabric. The IB fabric can apply updated subnet configuration policies through one or more subnet managers that can recognize an M_Key associated with the updated subnet configuration policies at step 902. Then, at step 903, the subnet managers in the IB subnet that can not recognize the M_Key can be left unaffected by the new policies. Such subnet managers are thereby effectively excluded from the group of cooperating subnet managers in this subnet.

Guarding against negative effect of accidental subnet merge

[00068] **Figure 10** shows an illustration of a middleware machine environment with an accidental subnet merge, in accordance with an embodiment of the invention. As shown in Figure 10, an IB subnet or fabric 1020 in the middleware machine environment 1000 manages a plurality of end nodes. The subnet manager A 1001 is the master subnet manager that configures and manages the IB subnet using an M_Key A 1021.

[00069] In accordance with an embodiment of the invention, the system can guard against potential negative effect associated with an accidental subnet merge. In the example as shown in Figure 10, a subnet manager E 1005, which is not part of the IB fabric, is accidentally connected to the IB subnet 1020 by mistake via a network connection 1030. The subnet manager E 1015 can first try to discover port topology using M_Keys or explicit node/port-list configuration. Since the connection of the subnet manager E to the IB fabric is unintentional, the subnet manager E can not recognize the M_Key C 1021 that is used in the IB subnet 1020. Accordingly, the subnet manager E will not communicate with the ports/nodes in the IB subnet

1020, in order to prevent undesired consequence such as unauthorized access. Also, the subnet manager A may not know the M_Key value used in the other subnet that the subnet manger E is a part of. Accordingly, the subnet manager A will not try to, or be able to change any state in the other subnet. Accordingly, there can be no change of state in either IB subnet 1020 or the other IB subnet as a result of the said accidental connectivity established between the two IB subnets.

[00070] **Figure 11** illustrates an exemplary flow chart for guarding against negative effect of accidental subnet merge in a middleware machine environment, in accordance with an embodiment of the invention. As shown in Figure 11, at step 1101, a master subnet manger in the IB subnet can detect a new subnet manager from another IB subnet. Then, at step 1102, the master subnet manager can determine that the connection is unintentional, since the new subnet manager can not recognize the M_Key used in the IB subnet. Accordingly, the IB fabric can prevent the new subnet manager from changing any state in the IB subnet at step 1103.

[00071] In accordance with some embodiments, **Figure 12** shows a functional block diagram of a subnet manager 1200, configured in accordance with the principles of the invention as described above, within a subnet in a middleware machine environment; **Figure 13** shows a functional block diagram of a network switch 1300, configured in accordance with the principles of the invention as described above, comprising the subnet manager 1200 as shown in Figure 12; and **Figure 14** shows a functional block diagram of a supporting system 1400, configured in accordance with the principles of the invention as described above, comprising the network switch 1300 as shown in Figure 13. The functional blocks of the subnet manager, network switch and supporting system may be implemented by hardware, software, or a combination of hardware and software to carry out the principles of the invention. It is understood by persons of skill in the art that the functional blocks described in Figures 12-14 may be combined or separated into sub-blocks to implement the principles of the invention as described above. Therefore, the description herein may support any possible combination or separation or further definition of the functional blocks described herein.

[00072] As shown in Figure 12, the subnet manager 1200 can comprise an associating module 1202 and a negotiating module 1204. In some embodiments, the associating module 1202 can be configured to associate a different private secure key with the subnet manager. The negotiating module 1204 can be configured to allow the subnet manager to negotiate with other subnet managers within the subnet and elect a master subnet manager, which is responsible for configuring and managing the middleware machine environment using the private secure key that is associated with the master subnet manager.

[00073] In some embodiments, the subnet is an Infiniband (IB) subnet. In some embodiments, the subnet can be divided into a dynamic set of resource domains implemented by subnet partitions.

[00074] In some embodiments, alternatively, the subnet manager 1200 can further comprise a

communication module 1206 configured to allow the subnet manager to communicate with other subnet managers within the subnet using an in-band communication protocol. In some embodiments, alternatively, the subnet manager 1200 can further comprise an initialization module 1208 configured to, responsive to the subnet manager 1200 being elected as the master subnet manager, initialize the master subnet manager using a default patitioning policy when no partitioning policy is specified.

[00075] In some embodiments, the private secure key is a M_key that is a 64 bit secret value that is known only to authorized entities in the subnet. When a subnet management agent (SMA) associated with a port in the subnet is configured with a M_Key value, an in-band request needs to specify the M_Key value in order to change a state associated with the port.

[00076] In some embodiments, the subnet is reconfigured to be associated with a different private secure key, when a different subnet manager is elected as the master subnet manager. In some embodiments, each different private secure key is defined in a different range that is known to the other subnet manager within the subnet.

[00077] In some embodiments, alternatively, the subnet manager 1200 can further comprise a determining module 1210 configured to dynamically determine which private secure key in a defined range is in use depending on which subnet manager instance is currently the master subnet manager. In some embodiments, an old master subnet manager is automatically prevented from resuming normal operations as a master subnet manager after a new master subnet manager is elected in order to prevent a split brain scenario.

[00078] In some embodiments, the master subnet manager can determine that a connection to a new subnet manager is unintentional since the new subnet manager can not recognize the private secure key used in the subnet, and the new subnet manager is prevented to change any state in the subnet.

[00079] As shown in Figure 13, the network switch 1300 can comprise one subnet manager 1200 as shown in Figure 12, one or more external ports 1304 and one or more internal ports 1308. The external ports 1304 are used to connect with an external network. The internal ports 1308 are used to connect with a plurality of host servers in the middleware machine environment.

[00080] As shown in Figure 14, the supporting system 1400 can further comprise one or more network switches 1300 as shown in Figure 13. In some embodiments, alternatively,the supporting system 1400 can further comprise a separate storage system 1404 that connects with the plurality of host servers through said one or more network switches. In some embodiments, alternatively, the supporting system 1400 can further comprise one or more gateway 1408 that can be accessed by a guest. Although the gateway 1408 is illustrated to be a seperate component from the network switch 1300, in some embodiments, the gateway 1408 can reside on the network switch 1300.

[00081] In some embodiments, one or more updated subnet configuration policies can be applied in the subnet through one or more subnet managers 1200 that can recognize a private secure key associated with the updated subnet configuration policies, and other subnet managers 1200 within the subnet that can not recognize the private secure key can be left unaffected by the one or more updated subnet configuration policies.

[00082] The present invention may be conveniently implemented using one or more conventional general purpose or specialized digital computer, computing device, machine, or microprocessor, including one or more processors, memory and/or computer readable storage media programmed according to the teachings of the present disclosure. Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art.

[00083] In some embodiments, the present invention includes a computer program product which is a storage medium or computer readable medium (media) having instructions stored thereon/in which can be used to program a computer to perform any of the processes of the present invention. The storage medium can include, but is not limited to, any type of disk including floppy disks, optical discs, DVD, CD-ROMs, microdrive, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, DRAMs, VRAMs, flash memory devices, magnetic or optical cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data.

[00084] The foregoing description of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations will be apparent to the practitioner skilled in the art. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention for various embodiments and with various modifications that are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalence.

Claims:

What is claimed is:

1. A system for supporting a middleware machine environment, comprising:

one or more microprocessors;

a set of subnet manager instances that resides on one or more nodes running on the one or more microprocessors in the middleware machine environment, wherein the set of subnet manager instances cooperate to provide a highly available subnet manager service within a subnet, wherein each said subnet manager instance is associated with a different private secure key, and

wherein the set of subnet manager instances can negotiate with each other and elect a master subnet manager, which is responsible for configuring and managing the middleware machine environment using the private secure key that is associated with the master subnet manager.

2. The system according to Claim 1, wherein:

the one or more nodes include one or more network switches, wherein each said network switch provides one or more external ports that are used to connect with an external network, and one or more internal ports that are used to connect with a plurality of host servers in the middleware machine environment.

3. The system according to Claim 2, further comprising:

a separate storage system that connects with the plurality of host servers through said one or more network switches.

4. The system according to any one of Claims 1 to 3, wherein:

the subnet is an Infiniband (IB) subnet.

5. The system according to any one of Claims 1 to 4, further comprising:

one or more gateway instances that can be accessed by a guest.

6. The system according to any one of Claims 1 to 5, wherein:

the subnet manager instances can communicate with each other using an in-band communication protocol.

7. The system according to any one of Claims 1 to 6, wherein:

the subnet can be divided into a dynamic set of resource domains implemented by subnet partitions.

8. The system according to any one of Claims 1 to 7, wherein:

the master subnet manager can use a default patitioning policy for initialization when no partitioning policy is specified.

9. The system according to any one of Claims 1 to 8, wherein:

the private secure key is a M_Key that is a 64 bit secret value that is known only to authorized entities in the subnet.

10. The system according to Claim 9, wherein:

when a subnet management agent (SMA) associated with a port in the subnet is configured with a M_Key value, an in-band request needs to specify the M_Key value in order to change a state associated with the port.

11. The system according to any one of Claims 1 to 10, wherein:

the subnet is reconfigured to be associated with a different private secure key, when a different subnet manager instance is elected as the master subnet manager.

12. The system according to any one of Claims 1 to 11, wherein:

each different private secure key is defined in a different range that is known to other subnet manager instances in the subnet.

13. The system according to any one of Claims 1 to 12, wherein:

the set of subnet manager instances in the subnet can dynamically determine which private secure key in a defined range is in use depending on which subnet manager instance is currently the master subnet manager.

14. The system according to any one of Claims 1 to 13, wherein:

an old master subnet manager is automatically prevented from resuming normal operations as a master subnet manager after a new master subnet manager is elected in order to prevent a split brain scenario.

15. The system according to any one of Claims 1 to 14, wherein:

one or more updated subnet configuration policies can be applied in the subnet through one or more subnet managers that can recognize a private secure key associated with the

updated subnet configuration policies, and other subnet managers in the subnet that can not recognize the private secure key can be left unaffected by the one or more updated subnet configuration policies and be prevented from updating a state of the subnet.

16. The system according to any one of Claims 1 to 15, wherein:

the master subnet manager can determine that a connection to a new node is unintentional since the master subnet manager can not recognize the private secure key used for the node.

17. The system according to any one of Claims 1 to 16, wherein:

the master subnet manager in a remote subnet containing the new node is not allowed to change a state in the subnet and the master subnet manager in the subnet is not allowed to change a state in the remote subnet.

18. A method for supporting a middleware machine environment, comprising:

associating a different private secure key with each subnet manager instance in a set of subnet manager instances, wherein each said subnet manager instance resides on one or more nodes running on one or more microprocessors in the middleware machine environment, wherein the set of subnet manager instances cooperate to provide a highly available subnet manager service within an IB subnet;

allowing the set of subnet manager instances to negotiate with each other and elect a master subnet manager; and

configuring and managing the middleware machine environment using the private secure key that is associated with the master subnet manager.

19. The method according to Claim 18, wherein:

one or more network switches each provide one or more external ports that are used to connect with the external network, and one or more internal ports that are used to connect with a plurality of host servers in the middleware machine environment.

20. The method according to Claim 19, further comprising:

providing a separate storage system connecting with the plurality of host servers through said one or more network switches.

21. The method according to any of Claims 18 to 20, wherein:

the subnet is an Infiniband (IB) subnet.

22. The method according to any of Claims 18 to 21, further comprising:

providing one or more gateway instances that can be accessed by a guest.

23. The method according to any of Claims 18 to 22, further comprising:

the subnet manager instances communicating with each other using an in-band communication protocol.

24. The method according to any of Claims 18 to 23, further comprising:

dividing the subnet into a dynamic set of resource domains implemented by subnet partitions.

25. The method according to any of Claims 18 to 24, further comprising:

the master subnet manager using a default patitioning policy for initialization when no partitioning policy is specified.

26. The method according to any of Claims 18 to 25, wherein:

the private secure key is a M_key that is a 64 bit secret value that is known only to authorized entities in the subnet.

27. The method according to any of Claims 18 to 26, wherein:

when a subnet management agent (SMA) associated with a port in the subnet is configured with a M_Key value, an in-band request needs to specify the M_Key value in order to change a state associated with the port.

28. The method according to any of Claims 18 to 27, further comprising:

reconfiguring the subnet to be associated with a different private secure key, when a different subnet manager instance is elected as the master subnet manager.

29. The method according to any of Claims 18 to 28, further comprising:

defining each different private secure key in a different range that is known to other subnet manager instances in the subnet.

30. The method according to any of Claims 18 to 29, further comprising:

the set of subnet manager instances in the subnet dynamically determining which private secure key in a defined range is in use depending on which subnet manager instance is currently the master subnet manager.

31. The method according to any of Claims 18 to 30, wherein:

an old master subnet manager is automatically prevented from resuming normal operations as a master subnet manager after a new master subnet manager is elected in order to prevent a split brain scenario.

32. The method according to any of Claims 18 to 31, wherein:

one or more updated subnet configuration policies can be applied in the subnet through one or more subnet managers that can recognize a private secure key associated with the updated subnet configuration policies, and other subnet managers in the subnet that can not recognize the private secure key can be left unaffected by the one or more updated subnet configuration policies.

33. The method according to any of Claims 18 to 32, further comprising:

the master subnet manager determining that a connection to a new subnet manager is unintentional since the new subnet manager can not recognize the private secure key used in the subnet, and preventing the new subnet manager from changing any state in the subnet.

34. A computer program comprising program readable instructions that when loaded into and executed by one or more computer systems cause the one or more computer systems to perform the method of any of claims 19 to 33.

35. A computer program product comprising a computer readable storage medium storing the computer program of claim 34.

36. A machine readable medium having instructions stored thereon that when executed cause a system to perform the steps of:

associating a different private secure key with each subnet manager instance in a set of subnet manager instances, wherein each said subnet manager instance resides on one or more nodes running on one or more microprocessors in a middleware machine environment, wherein the set of subnet manager instances cooperate to provide a highly available subnet manager service within an IB subnet;

allowing the set of subnet manager instances to negotiate with each other and elect a master subnet manager; and

configuring and managing the middleware machine environment using the private secure key that is associated with the master subnet manager.

37. A computer program for causing a computer to perform the steps of:

associating a different private secure key with each subnet manager instance in a set of subnet manager instances running on one or more microprocessors, wherein the set of subnet manager instances cooperate to provide a highly available subnet manager service within an IB subnet;

allowing the set of subnet manager instances to negotiate with each other and elect a master subnet manager; and

configuring and managing the middleware machine environment using the private secure key that is associated with the master subnet manager.

38. A subnet manager within a subnet in a middleware machine environment, comprising:

an associating module configured to associate a different private secure key with the subnet manager; and

a negotiating module configured to allow the subnet manager to negotiate with other subnet managers within the subnet and elect a master subnet manager, which is responsible for configuring and managing the middleware machine environment using the private secure key that is associated with the master subnet manager.

39. The subnet manager according to Claim 38, wherein:

the subnet is an Infiniband (IB) subnet.

40. The subnet manager according to Claim 38, further comprising:

a communication module configured to allow the subnet manager to communicate with other subnet managers within the subnet using an in-band communication protocol.

41. The subnet manager according to Claim 38, wherein:

the subnet can be divided into a dynamic set of resource domains implemented by subnet partitions.

42. The subnet manager according to Claim 38, further comprising:

an initialization module configured to, responsive to the subnet manager being elected as the master subnet manager, initialize the master subnet manager using a default patitioning policy when no partitioning policy is specified.

43. The subnet manager according to Claim 38, wherein:

the private secure key is a M_key that is a 64 bit secret value that is known only to authorized entities in the subnet.

44. The subnet manager according to Claim 43, wherein:

when a subnet management agent (SMA) associated with a port in the subnet is configured with a M_Key value, an in-band request needs to specify the M_Key value in order to change a state associated with the port.

45. The subnet manager according to Claim 38, wherein:

the subnet is reconfigured to be associated with a different private secure key, when a different subnet manager is elected as the master subnet manager.

46. The subnet manager according to Claim 38, wherein:

each different private secure key is defined in a different range that is known to the other subnet manager within the subnet.

47. The subnet manager according to Claim 38, further comprising:

a determining module configured to dynamically determine which private secure key in a defined range is in use depending on which subnet manager is currently the master subnet manager.

48. The subnet manager according to Claim 38, wherein:

an old master subnet manager is automatically prevented from resuming normal operations as a master subnet manager after a new master subnet manager is elected in order to prevent a split brain scenario.

49. The subnet manager according to Claim 38, wherein:

the master subnet manager can determine that a connection to a new subnet manager is unintentional since the new subnet manager can not recognize the private secure key used in the subnet, and the new subnet manager is prevented to change any state in the subnet.

50. A network switch comprising:

one subnet manager according to any one of claims 38-49;

one or more external ports that are used to connect with an external network; and

one or more internal ports that are used to connect with a plurality of host servers in the middleware machine environment.

51. A system for supporting a middleware machine environment, comprising one or more network switches according to claim 50.

52. The system according to Claim 51, further comprising:

a separate storage system that connects with the plurality of host servers through said one or more network switches.

53. The system according to Claim 51, further comprising:

one or more gateway that can be accessed by a guest.

54. The system according to Claim 51, wherein:

one or more updated subnet configuration policies can be applied in the subnet through one or more subnet managers that can recognize a private secure key associated with the updated subnet configuration policies, and other subnet managers within the subnet that can not recognize the private secure key can be left unaffected by the one or more updated subnet configuration policies.

55. A system for supporting a middleware machine environment, comprising:

a set of subnet manager that cooperate to provide a highly available subnet manager service within a subnet, wherein each said subnet manager is associated with a different private secure key, and

wherein the set of subnet manager can negotiate with each other and elect a master subnet manager, which is responsible for configuring and managing the middleware machine environment using the private secure key that is associated with the master subnet manager.

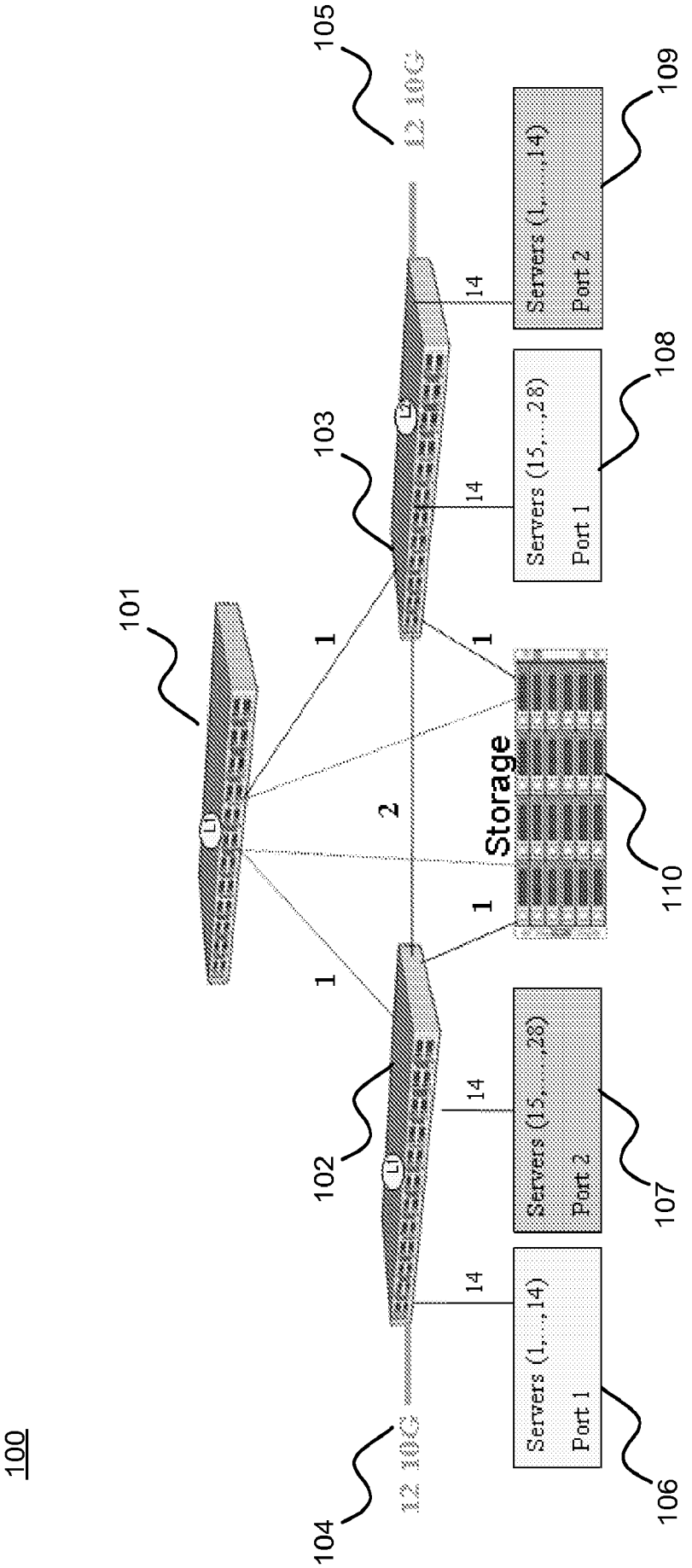
100
101
102
103
104
105
106
107
108
109
110
12 10G
12 10G
Storage
Servers (1,...,14)
Port 1
Servers (15,...,28)
Port 2
Servers (15,...,28)
Port 1
Servers (1,...,14)
Port 2
1
2
14

FIGURE 1

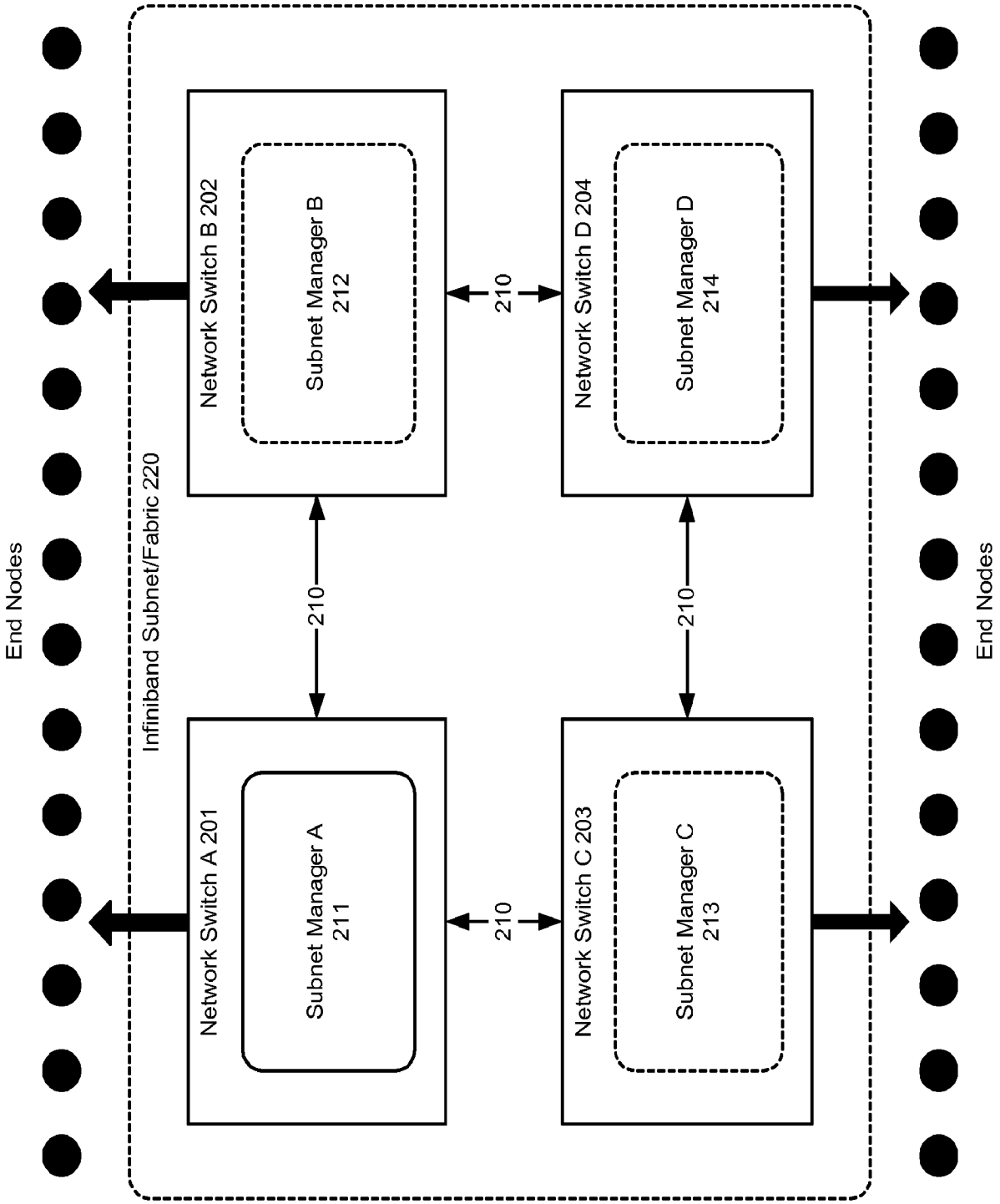
End Nodes
Infiniband Subnet/Fabric 220
Network Switch A 201
Subnet Manager A
211
Network Switch B 202
Subnet Manager B
212
Network Switch C 203
Subnet Manager C
213
Network Switch D 204
Subnet Manager D
214
210
210
210
210
End Nodes
200

FIGURE 2

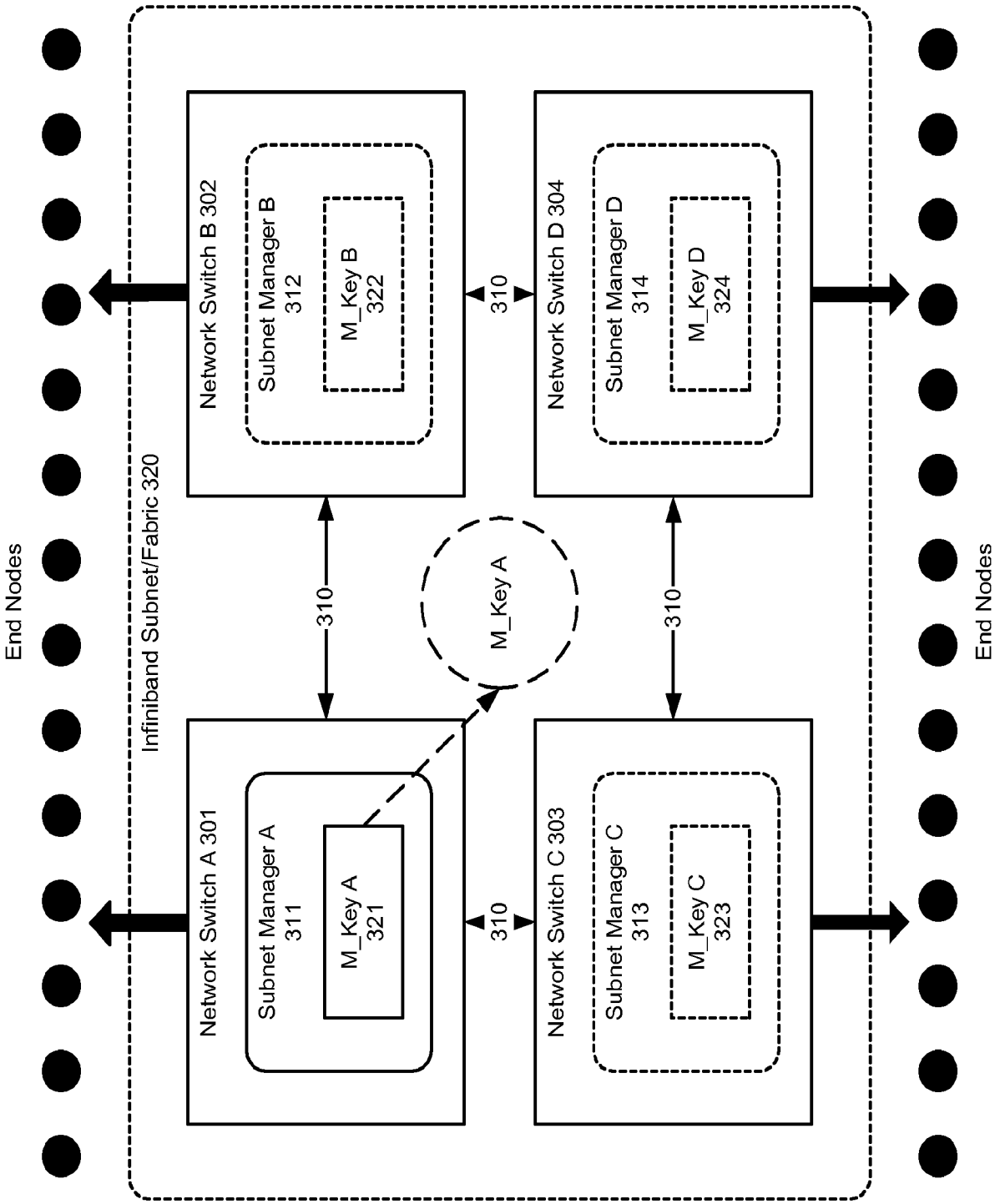
300
End Nodes
Infiniband Subnet/Fabric 320
Network Switch A 301
Subnet Manager A
311
M_Key A
321
Network Switch B 302
Subnet Manager B
312
M_Key B
322
Network Switch C 303
Subnet Manager C
313
M_Key C
323
Network Switch D 304
Subnet Manager D
314
M_Key D
324
310
310
310
310
M_Key A
End Nodes

FIGURE 3

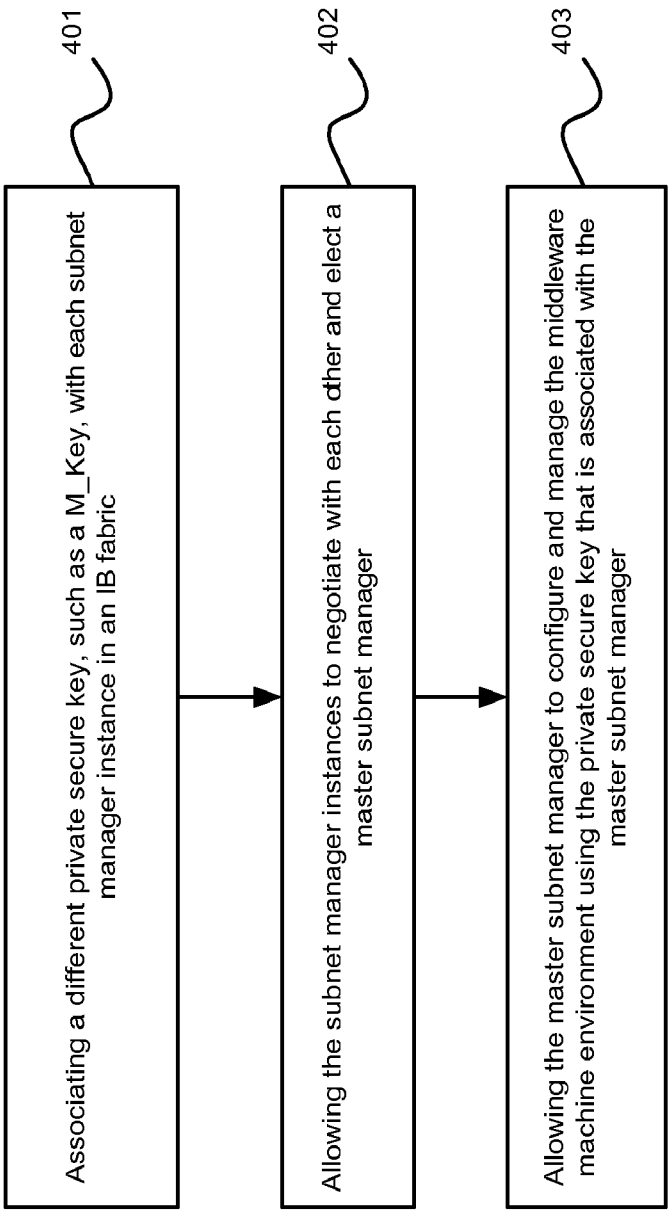
401
Associating a different private secure key, such as a M_Key, with each subnet manager instance in an IB fabric
402
Allowing the subnet manager instances to negotiate with each other and elect a master subnet manager
403
Allowing the master subnet manager to configure and manage the middleware machine environment using the private secure key that is associated with the master subnet manager

FIGURE 4

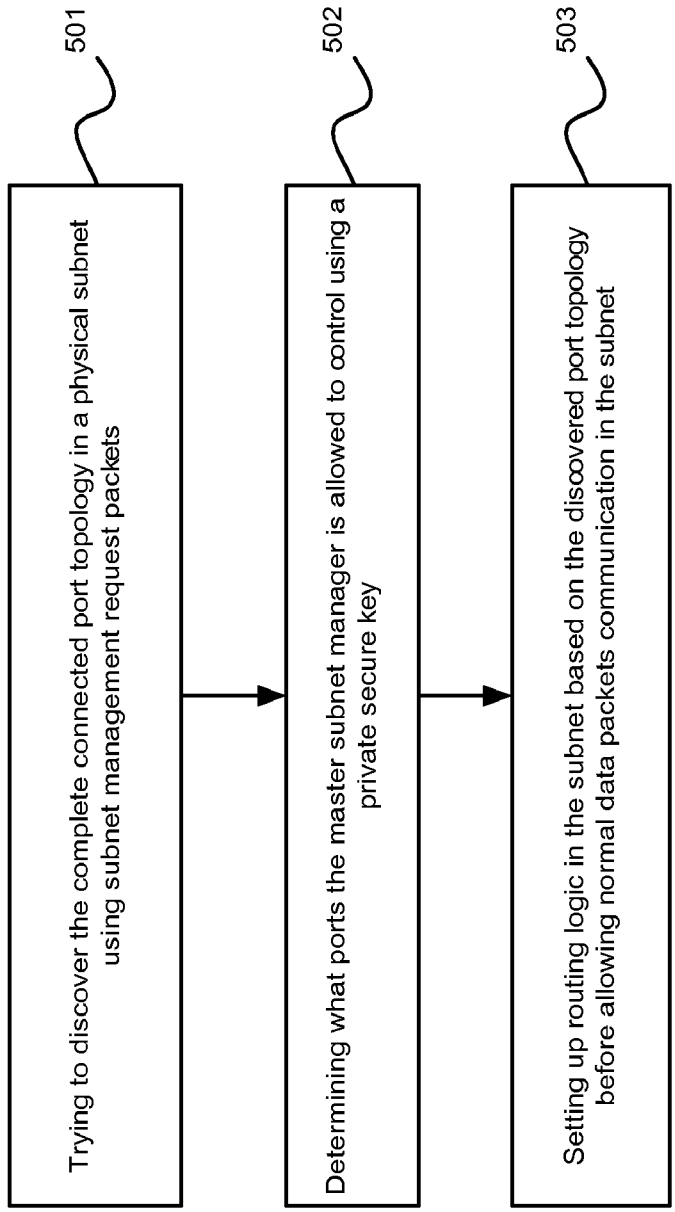
501
Trying to discover the complete connected port topology in a physical subnet using subnet management request packets
502
Determining what ports the master subnet manager is allowed to control using a private secure key
503
Setting up routing logic in the subnet based on the discovered port topology before allowing normal data packets communication in the subnet

FIGURE 5

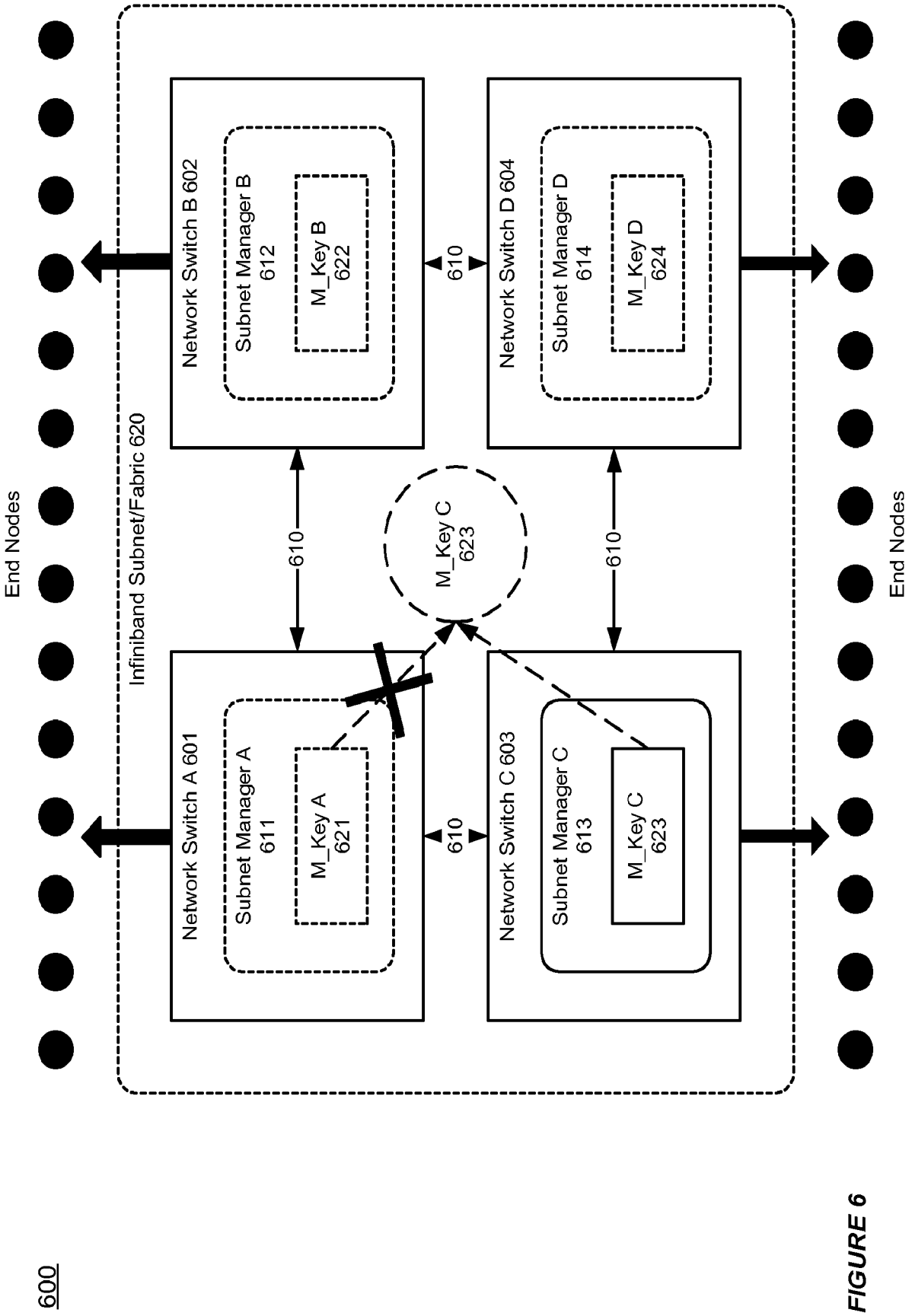
600
End Nodes
Infiniband Subnet/Fabric 620
Network Switch A 601
Subnet Manager A
611
M_Key A
621
Network Switch B 602
Subnet Manager B
612
M_Key B
622
Network Switch C 603
Subnet Manager C
613
M_Key C
623
Network Switch D 604
Subnet Manager D
614
M_Key D
624
610
610
610
610
M_Key C
623
End Nodes

FIGURE 6

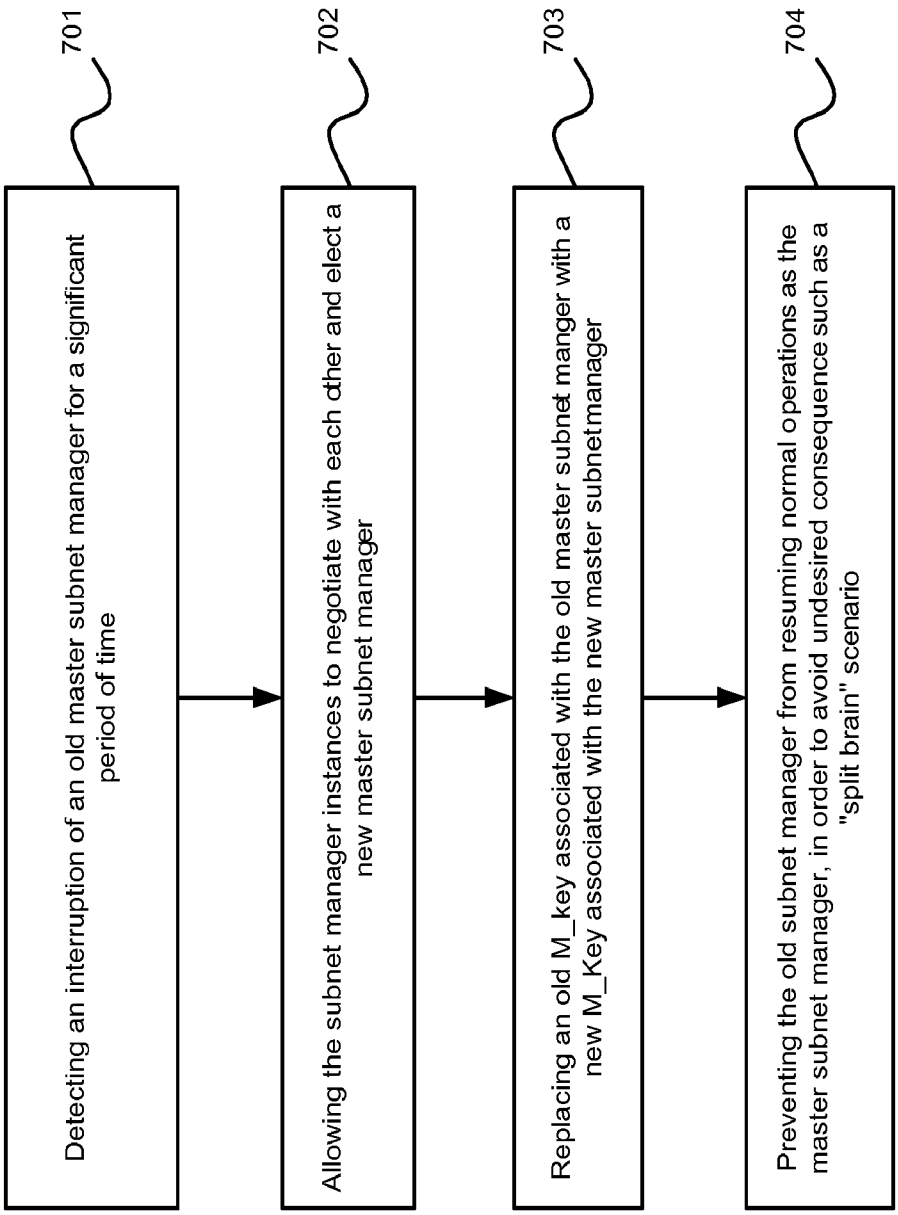
701
Detecting an interruption of an old master subnet manager for a significant period of time
702
Allowing the subnet manager instances to negotiate with each other and elect a new master subnet manager
703
Replacing an old M_key associated with the old master subnet manger with a new M_Key associated with the new master subnetmanager
704
Preventing the old subnet manager from resuming normal operations as the master subnet manager, in order to avoid undesired consequence such as a "split brain" scenario

FIGURE 7

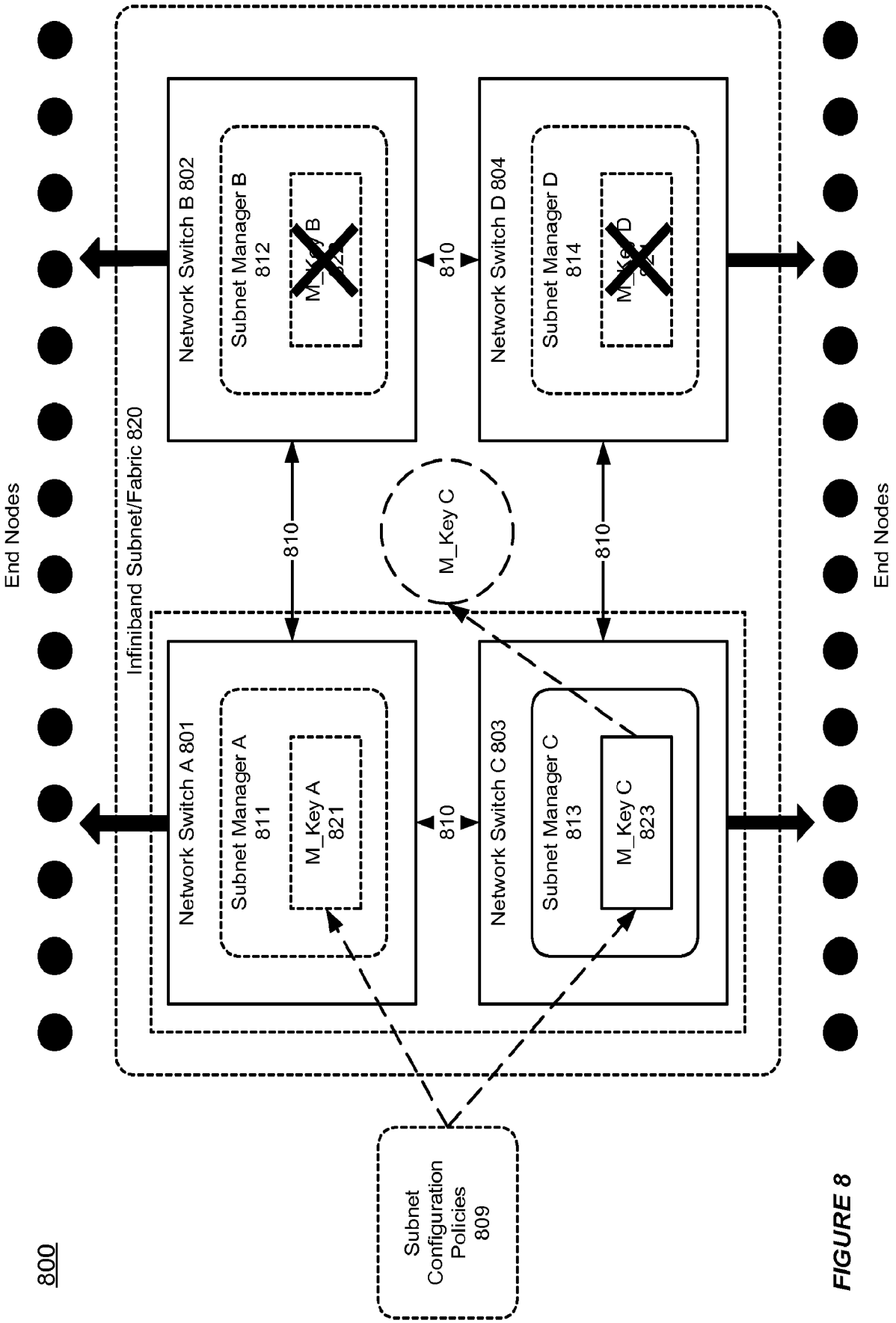
800
End Nodes
Infiniband Subnet/Fabric 820
Network Switch A 801
Subnet Manager A
811
M_Key A
821
Network Switch B 802
Subnet Manager B
812
Network Switch C 803
Subnet Manager C
813
M_Key C
823
Network Switch D 804
Subnet Manager D
814
810
M_Key C
Subnet
Configuration
Policies
809
End Nodes

FIGURE 8

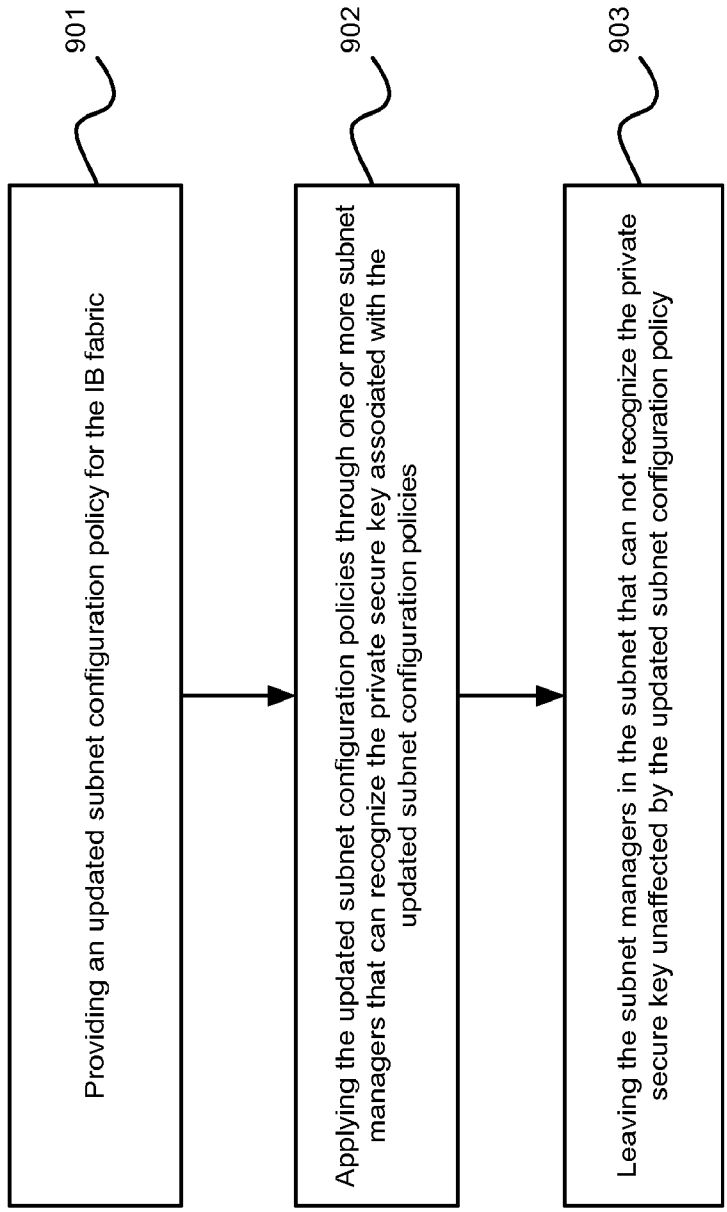
901
Providing an updated subnet configuration policy for the IB fabric
902
Applying the updated subnet configuration policies through one or more subnet managers that can recognize the private secure key associated with the updated subnet configuration policies
903
Leaving the subnet managers in the subnet that can not recognize the private secure key unaffected by the updated subnet configuration policy

FIGURE 9

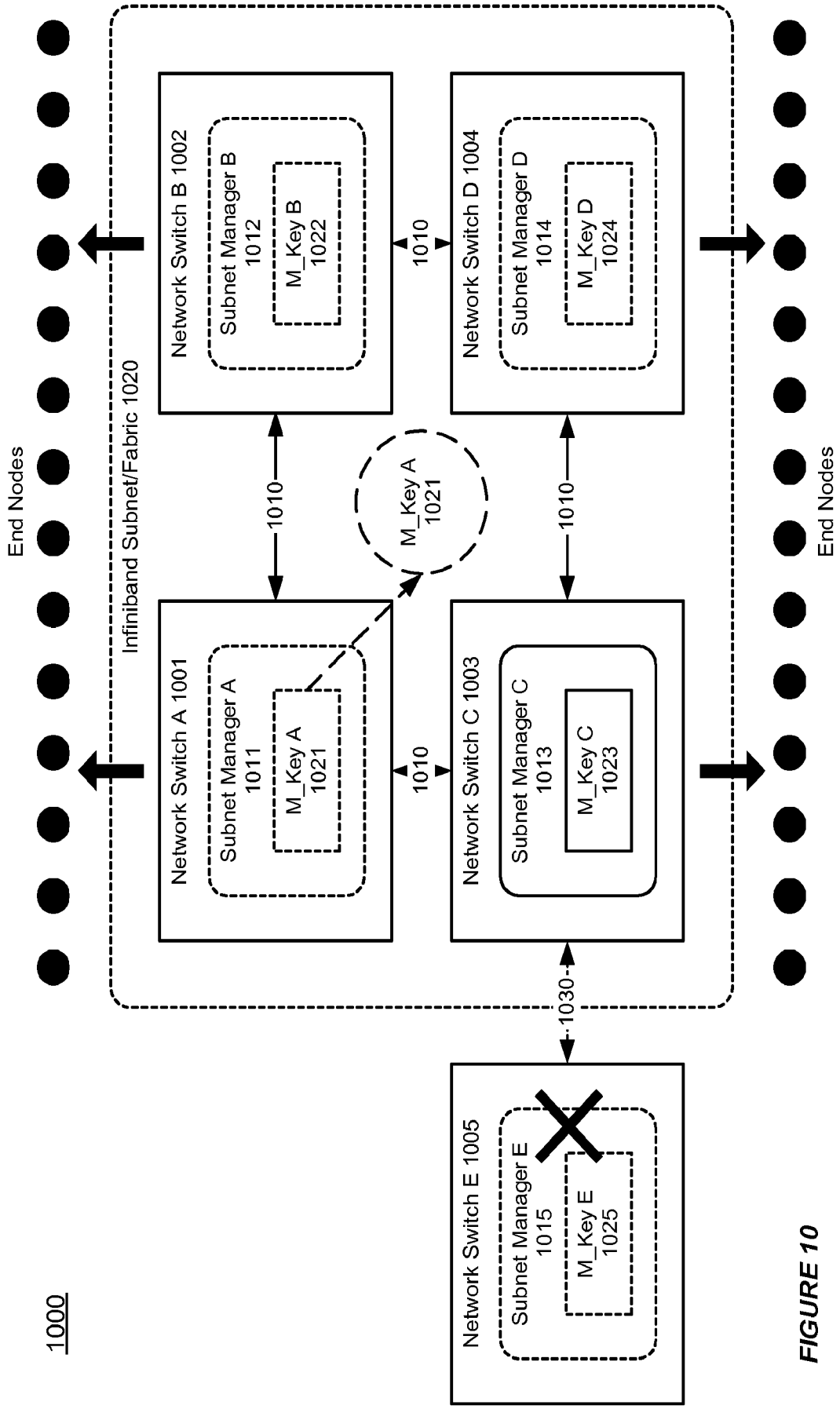
1000
End Nodes
Infiniband Subnet/Fabric 1020
Network Switch A 1001
Subnet Manager A
1011
M_Key A
1021
Network Switch B 1002
Subnet Manager B
1012
M_Key B
1022
Network Switch C 1003
Subnet Manager C
1013
M_Key C
1023
Network Switch D 1004
Subnet Manager D
1014
M_Key D
1024
Network Switch E 1005
Subnet Manager E
1015
M_Key E
1025
M_Key A
1021
1010
1010
1010
1030
End Nodes

FIGURE 10

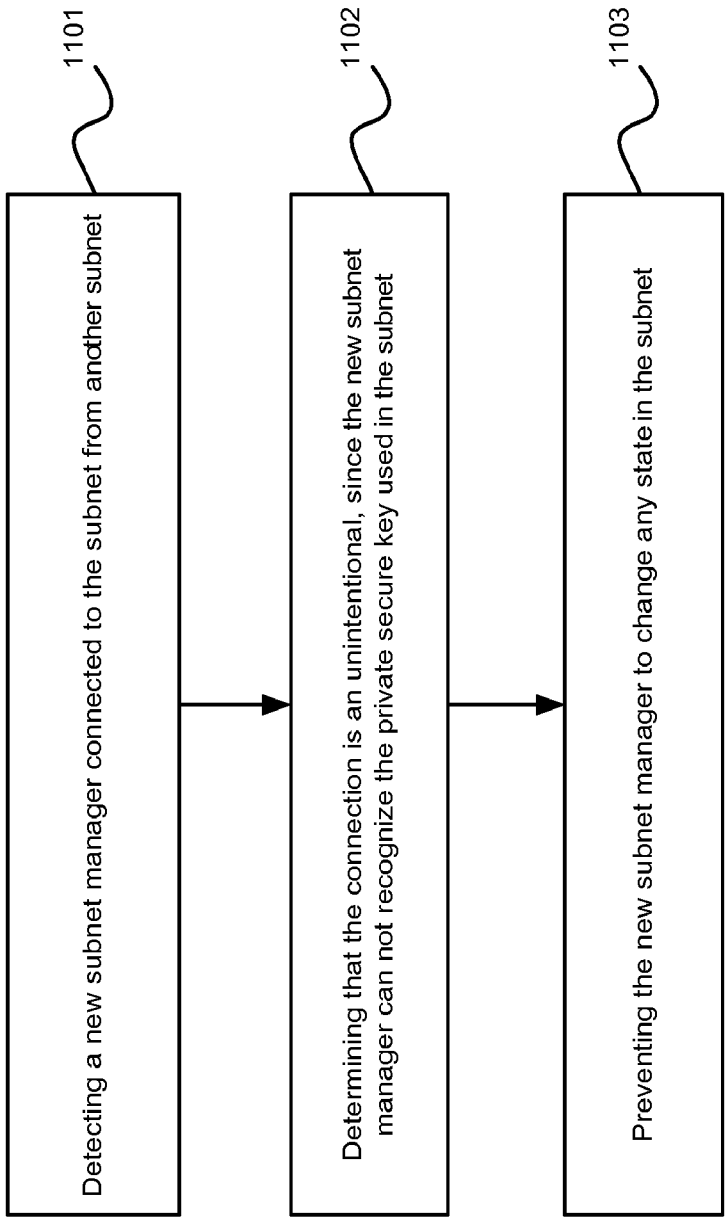
1101
Detecting a new subnet manager connected to the subnet from another subnet
1102
Determining that the connection is an unintentional, since the new subnet manager can not recognize the private secure key used in the subnet
1103
Preventing the new subnet manager to change any state in the subnet

FIGURE 11

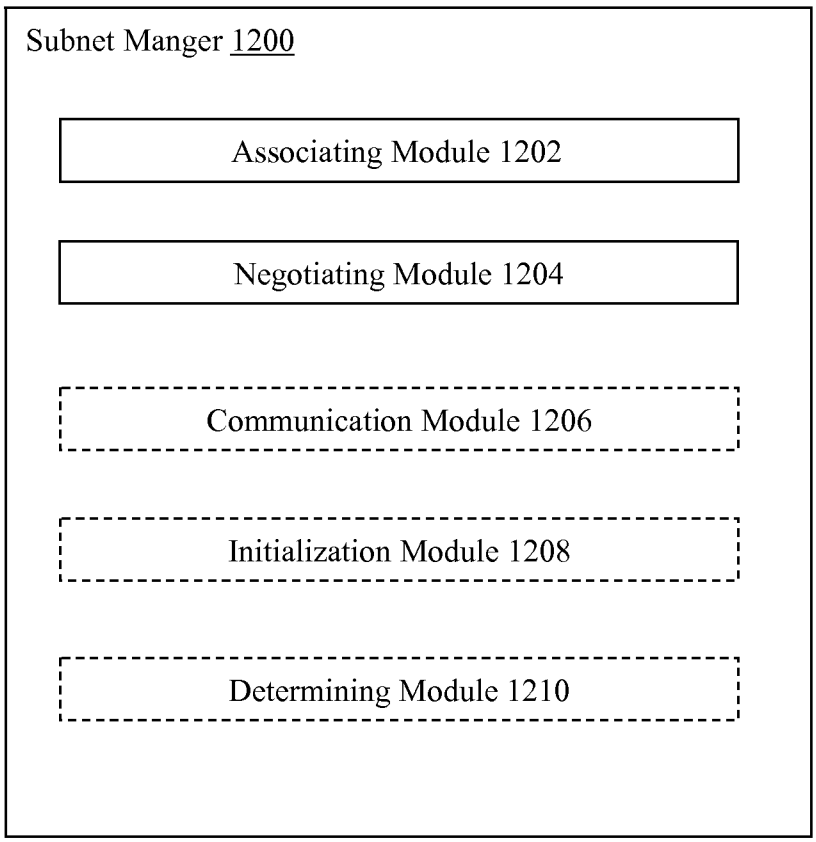
Subnet Manger 1200
Associating Module 1202
Negotiating Module 1204
Communication Module 1206
Initialization Module 1208
Determining Module 1210

FIGURE 12

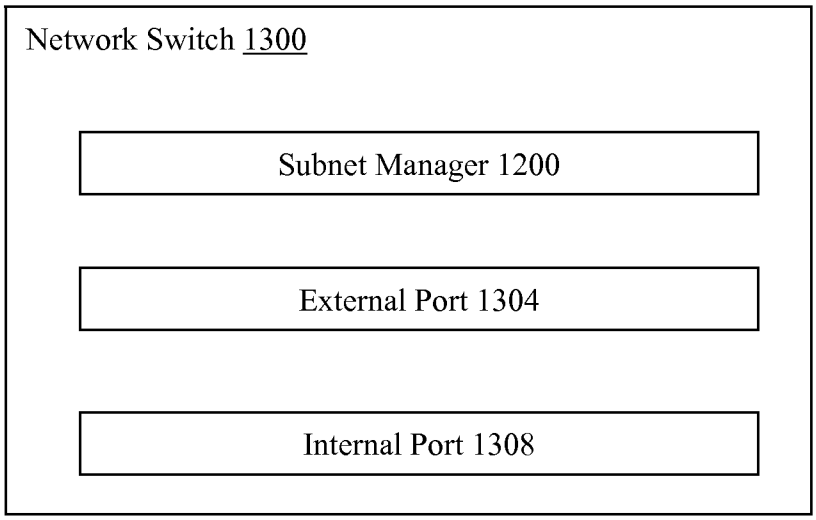

FIGURE 13

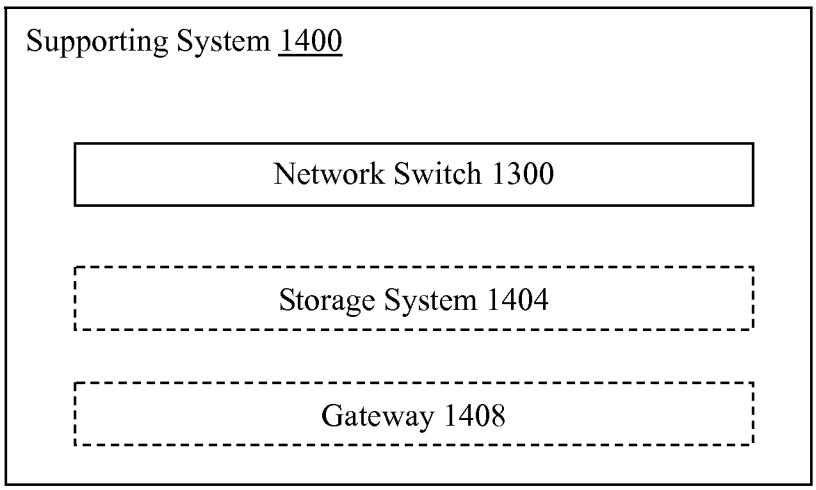

FIGURE 14

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2011/052029

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/24 H04L29/08 H04L12/56
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7 113 995 B1 (BEUKEMA BRUCE LEROY [US] ET AL) 26 September 2006 (2006-09-26) paragraph [0055] paragraph [0060] - paragraph [0065] -----	1-55

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
3 January 2012	11/01/2012

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Brichau, Gert

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2011/052029

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7113995 B1	26-09-2006	NONE	