(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 3 703 314 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
02.09.2020 Bulletin 2020/36

(51) Int Cl.:
H04L 12/24 *(2006.01)*

(21) Application number: **19315011.7**

(22) Date of filing: **28.02.2019**

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
KH MA MD TN

(71) Applicant: **OVH**
59100 Roubaix (FR)

(72) Inventors:
- **Chereau, Romain Françis Daniel**
31650 Saint-Orens-de-Gameville (FR)
- **Guerin, Benjamin André Jean**
29870 Lannilis (FR)
- **Havard, Thibaut Bernard Dominique**
29870 Lannilis (FR)
- **Bedouet, Olivier Jacques Henri**
26120 Malissard (FR)

(74) Representative: **BCF Global**
Centre d'Entreprise et d'Innovation
56, Bd Niels Bohr
CS 52132
69603 Villeurbanne Cedex (FR)

Remarks:
Amended claims in accordance with Rule 137(2) EPC.

(54) METHOD OF DEPLOYING A NETWORK CONFIGURATION IN A DATACENTER HAVING A POINT OF PRESENCE

(57) A method is devised of deploying a network configuration in a datacenter having a point of presence (POP). It involves the steps of :
- modelling the network configuration through objects, comprising a node object (813, 814, 826, 827) representing a point of interconnection in the network, an interface object (821, 822, 846, 847), being hierarchally inferior to the node object, and representing a connection to a point of interconnection, an evpnEdge object (823, 824, 816, 817) representing the transport to/from of a point of interconnection in the network, and a layer object (811, 822, 836, 837), being hierarchally inferior to the evpnEdge object, and representing the characteristics of the transported protocol to/from a point of interconnection ;
- setting up the network configuration by a succession of commands on the objects ; and
- pushing the set up network configuration to the POP and the datacenter.

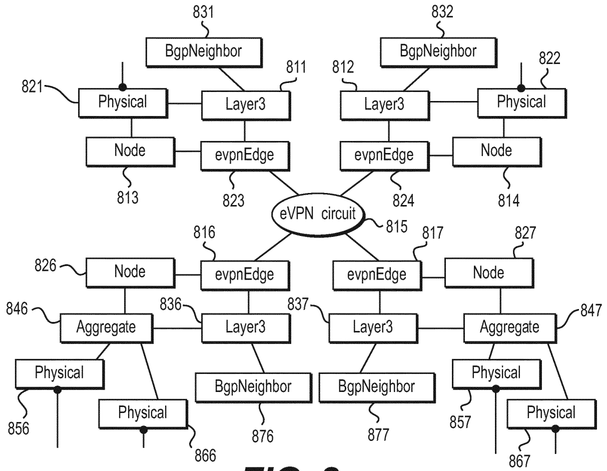


FIG. 8

Printed by Jouve, 75001 PARIS (FR)

Description

FIELD

[0001] The present technology relates to deploying network configurations in particular in a datacenter having a point of presence.

BACKGROUND

[0002] Today, connectivity of a customer's network to an infrastructure provider's datacenter is at the core of the cloud-related business. There is a need for efficiently deploying in a datacenter resources that integrate well within an existing customer network. Efficiency in that context points to :

- automation and speed of the deployment;
- ability to handle complex connectivity scenarios ; and
- deploying only valid and secure configurations in the point(s) of presence and the datacenter(s).

[0003] Various attempts have been made at tackling wholly or in part this quest for efficiency by infrastructure providers. However these attempts have failed to contain the problem in particular as it relates to full automation of a deployment, including :

- pushing Layer 2 configuration to the point(s) of presence and the datacenter(s); and
- managing Layer 3 configuration so as enable complex connectivity and flexible topology.

SUMMARY

[0004] The proposed invention aims at providing a method for deploying a network configuration by creating a model for network configurations involving objects, following a sequence of instantiating steps so as to set up the configuration when applied on the objects, then pushing the network configuration to physical resources.

[0005] Various implementations of the present technology provide a method of deploying a network configuration in a datacenter having a point of presence, comprising the steps of:

- modelling the network configuration through objects, comprising :
    - a node object representing a point of interconnection in the network ;
    - an interface object, being hierarchally inferior to the node object, and representing a connection to a point of interconnection ;
    - an evpnEdge object representing the transport to/from of a point of interconnection in the network ; and
    - a layer object, being hierarchally inferior to the evpnEdge object, and representing the characterististics of the transported protocol to/from a point of interconnection ;
- setting up the network configuration by a succession of commands on the objects ; and
- pushing the set up network configuration to the POP and the datacenter.

[0006] In one embodiment of the present technology, the method involves REST API calls applied on the objects as commands.

[0007] In one embodiment of the present technology, the pushing step is performed after a number of commands that is less than the total number of commands involved in the succession of commands on the objects.

[0008] In one embodiment of the present technology, the layer object is adapted for a Layer 2 protocol to be the transported protocol.

[0009] In one embodiment of the present technology, the layer object is adapted for a Layer 3 protocol to be the transported protocol.

[0010] In one embodiment of the present technology, the transported protocol is VxLAN.

[0011] In one embodiment of the present technology, the transported protocol is MPLS.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] For a better understanding of the present technology, as well as other aspects and further features thereof, reference is made to the following description which is to be used in conjunction with the accompanying drawings, where:

Fig.1 is an operational view of a first examplary customer network configuration ;
Fig.2 is a high level model for the first examplary customer network configuration ;
Fig.3 illustrates the model objects and classes applied to the first examplary customer network configuration ;
Fig.4 illustrates the classes as applied to the customer connection;
Fig.5 is an operational view of a second examplary customer network configuration;
Fig.6 is a logical view of the second examplary customer network configuration;
Fig.7 is a high level model for the second examplary customer network configuration ;
Fig.8 illustrates the model objects and classes applied to the second examplary customer network configuration ;
Fig.9 illustrates the generic ressources available in a provider datacenter(s) and point(s) of presence ;
Fig.10-A and 10.B describe a process for implementing the first examplary customer network configuration in the provider datacenter(s) and point(s) of presence ;
Fig.11 illustrates the effect of the process of Fig.10-A-B applied to the generic ressources available in the provider datacenter(s) and point(s) of presence ;
Fig.12-A to 12-D describe a process for implementing the second examplary customer network configuration in the provider datacenter(s) and point(s) of presence ;
Fig.13 illustrates the effect of the process of Fig.12-A-B applied to the generic ressources available in the provider datacenter(s) and point(s) of presence ;
Fig.14 illustrates the effect of the process of Fig.12-C applied to the generic ressources available in the provider datacenter(s) and point(s) of presence ;
Fig.15 illustrates the effect of the process of Fig.12-D applied to the generic ressources available in the provider datacenter(s) and point(s) of presence ; and
Figure 16 illustrates an example of a computing device that may be used to implement any of the methods described herein.

[0013] It should be noted that, unless otherwise explicitly specified herein, the drawings are not to scale. Finally, elements that are identical from one drawing to the other bear the same numeral reference across drawings.

DETAILED DESCRIPTION

[0014] The examples and conditional language recited herein are principally intended to aid the reader in understanding the principles of the present technology and not to limit its scope to such specifically recited examples and conditions. It will be appreciated that those skilled in the art may devise various arrangements which, although not explicitly described or shown herein, nonetheless embody the principles of the present technology and are included within its spirit and scope.

[0015] Furthermore, as an aid to understanding, the following description may describe relatively simplified implementations of the present technology. As persons skilled in the art would understand, various implementations of the present technology may be of a greater complexity.

[0016] In some cases, what are believed to be helpful examples of modifications to the present technology may also be set forth. This is done merely as an aid to understanding, and, again, not to define the scope or set forth the bounds of the present technology. These modifications are not an exhaustive list, and a person skilled in the art may make other modifications while nonetheless remaining within the scope of the present technology. Further, where no examples of modifications have been set forth, it should not be interpreted that no modifications are possible and/or that what is described is the sole manner of implementing that element of the present technology.

[0017] Moreover, all statements herein reciting principles, aspects, and implementations of the present technology, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof, whether they are currently known or developed in the future. Thus, for example, it will be appreciated by those skilled in the art that any block diagrams herein represent conceptual views of illustrative circuitry embodying the principles of the present technology. Similarly, it will be appreciated that any flowcharts, flow diagrams, state transition diagrams, pseudo-code, and the like represent various processes which may be substantially represented in computer-readable media and so executed by a computer or processor, whether or not such computer or processor is explicitly shown.

[0018] The functions of the various elements shown in the figures, including any functional block, may be provided through the use of dedicated hardware as well as hardware capable of executing software in association with appropriate software. When provided by a processor, the functions may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. In some embodiments of the present technology, the processor may be a general purpose processor, such as a central processing unit (CPU) or a processor dedicated to a specific purpose, such as a digital signal processor (DSP). In the aforementioned, explicit use of the term a « processor » should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include, without limitation, application specific integrated circuit (ASIC), field programmable gate array (FPGA), read-only memory (ROM) for storing software, random access memory (RAM), and non-volatile storage. Other hardware, conventional and/or custom, may also be included.

[0019] Software modules, or simply modules which are implied to be software, may be represented herein as any combination of flowchart elements or other elements indicating performance of process steps and/or textual description. Such modules may be executed by hardware that is expressly or implicitly shown. Moreover, it should be understood that « module » may include for example, but without being limitative, computer program logic, computer program instructions, software, stack, firmware, hardware circuitry or a combination thereof which provides the required capabilities.

[0020] With these fundamentals in place, we will now consider some non-limiting examples to illustrate various implementations of aspects of the present technology.

[0021] Fig.1 is an operational view of a first examplary customer network configuration. A customer operates a customer network 101, to which are interconnected merely as illustration, machines such as servers 102, 103, 104. Customer network 101 is connected through link 105 to a customer switch 106. Two physical connections 107 and 108, connect customer switch 106 to provider switch 109, and are configured by the customer in aggregate mode, for example using Link Aggregation Control Protocol (LACP). Provider switch 109 constitutes a Point Of Presence (POP) for, and connects to, provider network 111 through link 110. Provider network 111 is operated by an infrastructure provider, and could for example be a virtual resource dedicated to the customer, and for example reside in a provider's datacenter. Merely as illustration, several components could be connected to provider network 111, such as a private cloud 114, a public cloud 112 or a Software Defined Networking (SDN) component 113, so as to facilitate network management for provider network 111.

[0022] Connectivity at Layer 2 level between customer network 101 and provider network 111 according to Fig.1 may be represented using a high level model of the provider network and POP's configuration such as shown Fig.2.

Customer connection 201 enters, through an Aggregate class 202, an EntryNode 203 for provider network 111 (physically : the provider switch 109). An EndNode 205 in provider network 111 is connected to EntryNode 203 through a Layer 2 tunnel 204. Through an Aggregate class 206, connection 207 is given to the customer's dedicated virtual resource in provider network 111.

[0023] Connectivity at Layer 2 level between customer network 101 and provider network 111 through the POP according to Fig.1 and Fig.2, is automatically provided by the deployment of a configuration according to Fig.3. Implementation of the configuration is divided into objects and classes, with each object or class being associated with an implementation action. Each of EntryNode 203 and EndNode 205 is instantiated as the association of respectively evpnEdge objects 303 and 305, with respectively Node objects 313 and 315. Connection to customer's dedicated virtual resource 207 is instantiated as the association of Layer 2 object 307, with Aggregate class 317, and Physical classes 327 and 337. Customer connection 201 is instantiated as the association of Layer 2 object 301, with Aggregate class 311, and Physical classes 321 and 331. L2 tunnel 204 is instantiated as an evpn circuit object 304.

[0024] Objects and classes have thus been defined to cover all information required to deploy a configuration, and for a Representational State Transfer (REST) Application Programming Interface (API) to provide calls necessary to cover all networking situations.

Fig.4 illustrates the classes as applied to the customer connection 201. Physical classes 421 and 431 provide information regarding respectively switch and interface. Aggregate class 411 provides information regarding portChannelId. Class 401 is a Layer2 class. EvpnEdge Class 403 provides information regarding vlanId and the BGP Community. EvpnEdge class 404 provides information regarding vniId.

[0025] Fig.5 is an operational view of a second examplary customer network configuration.

A customer operates a customer network 101, as in Fig.1. Customer network 101 is connected through link 505 to a customer router 506. Two connections 507 and 508, connect customer router 506 to respectively provider routers 509 and 510. On each connection a Border Gateway Protocol (BGP) session is established. Provider routers 509 and 510 both constitute a POP for, and connect to, provider network 513 through links 511 and 512. Provider network 513 is operated by an infrastructure provider. Merely as illustration, several components could be connected to provider network 513, providing virtual dedicated resources to the customer, such as a private cloud 516, a public cloud 514 or a SDN component 515, for example physically residing in a first datacenter of the provider, and a private cloud 519, a public cloud 517 or a SDN component 518, for example physically residing in a second datacenter of the provider.

[0026] Fig.6 is a logical view of the second examplary customer network configuration. Customer operates an IP Network C 601, for example over customer network 101.

Customer router 506 is connected to IP Network C 601 through two Layer 3 connections. Each such connection is routed by customer router 506 to respectively IP Network B1 602 and IP Network B2 603.

Provider router 510 is BGP peered with customer router 506, and connects IP Network B1 602 to a Mesh 604. Provider router 509 is BGP peered with customer router 506, and connects IP Network B2 603 to Mesh 604. Provider routers 509 and 510 may be considered POPs for the provider network 513.

To Mesh 604 are also connected respectively provider routers 605 and 606.

Provider router 605 may be connected to IP Network D1 607 through a Layer 3 connection.

A provider router 609 may also be connected to IP Network D1 607, be BGP peered with provider router 605, and be connecting IP Network D1 607 to IP Network A1 611. To IP Network A1 611 may then be connected private cloud 519, public cloud 517 and SDN component 518.

Provider router 606 may be connected to IP Network D2 608 through a Layer 3 connection.

A provider router 610 may also be connected to IP Network D2 608, be BGP peered with provider router 606, and be connecting IP Network D2 608 to IP Network A2 612. To IP Network A2 612 may then be connected private cloud 516, public cloud 514 and SDN component 515.

[0027] Connectivity at Layer 3 level between customer network 101 and provider network 513 according to Fig.1 may be represented using a high level model of the configuration for provider network 513's and associated POPs, such as shown Fig.7.

Customer connection 701 enters an EntryNode 703 for provider network 513 (physically : the provider router 509, or POP). Customer connection 702 enters an EntryNode 704 for provider network 513 (physically : the provider router 510, or other POP).

EndNodes 706 and 707 in provider network 513 are connected in the Layer 3 domain to EntryNodes 703 and 704 through the Mesh 604. Connections 708 and 709 give access to the customer's dedicated virtual resource in provider network 513.

[0028] Connectivity at Layer 3 level between customer network 101 and provider network 111 according to Fig.5-7, is automatically provided by the deployment of a configuration according to Fig.8.

Implementation of the configuration is divided into objects and classes, which each object or class being associated with an implementation action. Each of EntryNode 703 and 704, is instantiated as the association of respectively evpnEdge objects 823 and 824, with respectively Node objects 813 and 814. Each of customer connections 701 and 702, is instantiated as the association of respectively Layer 3 objects 811 and 812, with respectively Physical classes 821 and 822, and BGP Neighbor objects 831 and 832. Mesh 604 is instantiated as evpn Circuit object 815. Each of EndNode 706 and 707, is instantiated as the association of respectively evpnEdge objects 816 and 817, with respectively Node objects 826 and 827. Connections to customer's dedicated virtual resource 708, 709 are instantiated as the association of respectively Layer 3 objects 836 and 837, Aggregate classes 846 and 847, Physical classes 856, 866, 857 and 867, and BGP Neighbor objects 876 and 877.

The hierarchy between objects is apparent, with the interface objects 821, 822, 846, 847, being hierarchally inferior to the node objects 813, 814, 826, 827, and layer objects 811, 822, 836, 837, being hierarchally inferior to the evpnEdge objects 823, 824, 816, 817.

[0029] A set of calls, syntax and associated parameters is defined as the REST API which, applied to four categories of objects modelling a network configuration: Node, Interface, evpnEdge, and Layer, allows to set up a valid and secure actual configuration, then push it in the provider datacenter(s) and POP(s). The model and the REST API make it possible to address all networking configurations beyond the first and second examplary customer network configurations above.

[0030] As regard the Node objects, the set of calls, syntax and associated parameters, is as follows:

Table 1

TYPE	URL	PARAMETERS [optional]	DESCRIPTION / RESPONSE
POST	/node/{id}/ interface	descr : 'Agregat' bandwidth: 1000 switchMode: 'trunk', 'access', 'routed' isMLAG : true lacpMode: 'normal'	Create an aggregate interface *computed: PortChannelId*
PUT	/node/{id}/ interface/{id}	descr :'eth1' bandwidth: 1000 switchMode: 'trunk', 'access', 'routed' speed: 1000	Configure a physical interface
PUT	/node/{id}/ interface/{id}	PortChannelId : 20	Attach the physical interface to an aggregate
PUT	/node/{id}/ interface/{id}	AdminState : up / down	Administrate physical interface state

(continued)

TYPE	URL	PARAMETERS [optional]	DESCRIPTION / RESPONSE
PUT	/node/{id}/ interface/{id}	isBlocked : true / false	Block / unblock the interface
DELETE	/node/{id}/ interface/{id}		Unconfigure physical interface or delete aggregate interface
GET	/node	[driver] [nodeOrigin] [nodeType] [hostname] [hostnamePeer] [ip] [ipPeer] [status]	List node
GET	/node/{id}		Get node details
GET	/node/{id}/ interface	[interfaceType] **if interfaceType='phy'** [descr] [adminStatus] [interfaceName] [peerNode] **IfinterfaceType='aggr'** [portChannelId]	Get interface list
GET	/node/{id}/ interface/{id}		Get interface details

[0031] As regard the Interface objects, they are instantiated through the REST API with URL: /nodes/{id}/interface

[0032] As regard the evpn objects, the set of calls, syntax and associated parameters, is as follows:

Table 2

TYPE	URL	PARAMETERS [optional]	DESCRIPTION / RESPONSE
POST	/evpn	evpnType : 'L2' or 'L3'	Create a L2 or L3 evpn *computed: vni* **L3 context** *computed: vrf*
POST	/evpn/{id}/evpn_edge	nodeId : nodeId import_policy : 'open' or 'limit' export_policy : 'open' or 'limit' **L2 context** vlanId : vlanId **L3 context** customerBgpAsn	Create an evpnNode *calculated: bgpComm*
POST	/evpn/{id}/evpn_edge/{id}/layer_ 2	InterfaceId : interface/{id} encapMode : 'tagged', 'untagged' allowedMac : 300 (100 by default)	Apply a Layer 2 on the interfaceId

(continued)

TYPE	URL	PARAMETERS [optional]	DESCRIPTION / RESPONSE
POST	/evpn/{id}/evpn_edge/{id}/layer_3	encapMode : 'tagged', 'untagged' interfaceId : interface/{id} subnet: '10.123.50.0/30' **Tagged context** vlanid : 11	Apply a new Layer 3 on the interfaceId
POST	/evpn/{id}/evpn_edge/{id}/layer_3/{id}/ bgp_neighbor	bgpAsn : 65501 Ip: 192.168.1.4	Add an IPV4 BGP neighbor in VRF
POST	/evpn/{id}/evpn_edge/{id}/layer_3/{id}/ static_routes	ipRoute : 192.168.20.0/16 ipGateway: 192.168.1.4	Add an IP static route
POST	/evpn/{id}/evpn_edge/{id}/layer_3/{id}/ vrrps	ip	Add an IP VRRP *calculated vrrpId calculated ip*
PUT	/evpn/{id}/evpn_edge/{id}/layer_3/{id}	customerBgpAsn	Modify the customer BGP ASN
PUT	/evpn/{id}/evpn_edge/{id}/layer_3/{id}	ip	Modify IP on Layer 3
PUT	/evpn/{id}/evpn_edge/{id}/layer_3/{id}/ bgp_neighbor/{id}	enable : true / false	Enable / disable a IPV4 BGP neighbor in VRF
DELETE	/evpn/{id}		Delete the evpn
DELETE	/evpn/{id}/evpn_edge/{id}		Delete the evpnEdge
DELETE	/evpn/{id}/evpn_edge/{id}/layer_2/{id}		Delete the Layer 2

Table 3

TYPE	URL	PARAMETERS [optional]	DESCRIPTION / RESPONSE
DELETE	/evpn/{id}/evpn_edge/{id}/layer_3/ {id}		Delete the Layer 3
DELETE	/evpn/{id}/evpn_edge/{id}layer_3/ {id} /bgp_neighbor/{id}		Delete the BGP neigbhor
DELETE	/evpn/{id}/evpn_edge/{id}/layer_3/ {id}/ static_routes/{id}		Delete the IP static route
DELETE	/evpn/{id}/evpn_edge/{id}/layer_3/ {id}/ vrrps/{id}		Delete the IP VRRP
GET	/evpn	[evpnType] [vni] [vrf]	List evpn
GET	/evpn/{id}		Get evpn details
GET	/evpn/{id}/evpn_edge	[nodeId] [evpnId]	List attached evpnedge
GET	/evpn/{id}/evpn_edge/{id}		Get evpnEdge details
GET	/evpn/{id}/evpn_edge/{id}/layer_2		Get the Layer 2 declared on the Edge
GET	/evpn/{id}/evpn_edge/{id}/layer_2/ {id}		Get Layer 2 details

(continued)

TYPE	URL	PARAMETERS [optional]	DESCRIPTION / RESPONSE
GET	/evpn/{id}/evpn_edge/{id}/layer_3	[encapMode] [subnet] [vlanId] [interfaceId]	Get the Layer 3 declared on the Edge
GET	/evpn/{id}/evpn_edge/{id}/layer_3/ {id}		Get Layer 3 details

[0033] As regard the Layer objects, they are instantiated through the REST API with URLs:

/evpn/{id}/evpn_edge/{id}/layer_3, or

/evpn/{id}/evpn_edge/{id}/layer-2

depending on the context of the customer configuration.

[0034] Aggregate and Physical classes are not visible in the above described REST API, but are still used and instantiated internally as the case may be, through a parameter in:
/node/{id}/interface
of a POST call in Table 1.

[0035] We will now show for the first examplary customer network configuration, how the REST API according to the invention allows to efficiently deploy the configuration into a provider datacenter(s) and POP(s).
The provider datacenter(s) and POP(s) are assumed to have the generic switch fabric and resources as described in relation to Fig.9. EntryNodel 901 and EntryNode2 902 each have two switches A and B. Each switch A or B has a number of EtherChannels as inputs, in this case, and merely as an example, switch B of EntryNodel has only one input Et1/1, while all other switches have two inputs Et1/1 and Et2/1. EntryNodel 901 and EntryNode2 902 represent two POPs for the provider infrastructure. Both EntryNodel 901 and EntryNode2 902 are connected to a provider backbone 903. To the same backbone 903 are connected EndNode1 904, EndNode2 905 and EndNode3 906. Each of these EndNodes has two switches A and B. Each switch A or B has provisioned a number of PortChannels as outputs, in this case, and merely as an example, respectively PortChannels Po101 and Po102 for Layer 3 connectivity, and a PortChannel Po100 common between switch A and switch B for Layer 2 connectivity. The choices made herein for generic resources are arbitrary, even if incidentally they provide a redundancy improving the robustness of the network. It will be apparent to the person skilled in the art that the generic switch fabric and resources as described in relation to Fig.9 are provided merely as an example of implementation, and that the invention could operate in other generic environments. For example:

- EntryNodes and EndNodes, or certain of them, could feature a different number of switches: 1, or more than 2;
- there could be any number of EntryNodes and EndNodes; etc.

[0036] The examplary parameter values are assumed to be had in the right hand side column of Table 4, as follows:

Table 4

	Value
EntryNode 1 (nodeId)	4
EntryNode 2 (nodeId)	5
BGP Area	65100
endNode A (nodeId)	1
Po100 (Id)	20
Po101 (Id)	21
Po102 (Id)	22
endNode B (nodeId)	2
Po100 (Id)	23

(continued)

	Value
Po101 (Id)	24
Po102 (Id)	25
endNode C (nodeId)	3
Po100 (Id)	26
Po101 (Id)	27
Po102 (Id)	28

[0037] Fig.10-A and 10.B describe a process for implementing the first examplary customer network configuration in the provider datacenter(s) and POP(s). At step 1001, an evpn object is created of the type Layer 2. At step 1002, a POP is created at EntryNodel with a vlanId=100. At step 1003, a Multi-chassis Link Aggregation (MLAG) is created. A step 1004, access to the physical interface on switch A of EntryNodel and more particularly EtherChannel 1/1, is obtained. At step 1005, the physical interface on switch A is configured. At step 1006, the physical interface on switch A is activated. At step 1007, the physical interface on switch A is attached to the MLAG. At step 1008, access to the physical interface on switch B of EntryNodel and more particularly EtherChannel 1/1, is obtained. At step 1009, the physical interface on switch B is configured. At step 1010, the physical interface on switch B is activated. At step 1011, the physical interface on switch B is attached to the MLAG. At step 1012, the vlan is associated to the PortChannel Po100. At step 1013, endNode1 is created. At step 1014, access to the PortChannel Po100 is obtained. At step 1015, Vlan is associated to PortChannel Po100.

[0038] The effect of applying the foregoing process on the generic switch fabric resources of the provider datacenter(s) and POP(s) is to set up the first examplary customer network configuration, according to Fig.11. Cross-reference is made between the physical elements of Fig.11 and the process steps of Fig.10A-B: MLAG 1101 has been set up at step 1003. 1102 and 1103 are connections to the EtherChannels respectively Et1/1 of switch A of EntryNodel 901, and Et1/1 of switch B of EntryNodel 901. These connections have been set up respectively at steps 1004 and 1008. Connection 1104 has been set up between EntryNodel 901 and provider backbone 903 at step 1002, and between provider backbone 903 and EndNode1 904 at step 1013. Connection 1105 has been set up at step 1015.

[0039] We will now show for the second examplary customer network configuration, how the REST API according to the invention allows to efficiently deploy the configuration in the provider datacenter(s) and point(s) of presence. The provider datacenter(s) and POP(s) are assumed to have the generic switch fabric and resources as described in relation to Fig.9, together with the parameter values of Table 4.

[0040] Fig.12-A to 12-D describe a process for implementing the second examplary customer network configuration in the provider datacenter(s) and POP(s). At step 1201, an evpn object is created of the type Layer 3. At step 1202, a POP is created at EntryNodel in a BGP area 65100, and Virtual Routing and Forwarding (VRF) configured on EntryNodel. A step 1203, access to the physical interface on switch A of EntryNodel and more particularly EtherChannel 2/1, is obtained. At step 1204, the physical interface (Et 2/1) on switch A is configured for routed mode. At step 1205, the physical interface on switch A is activated. At step 1206, an IP address is set up on EtherChannel 2/1. At step 1207, VRF is configured on EndNode1. At step 1208, an IP address is configured on PortChannel Po101 of EndNode1. At step 1209, an IP address is configured on PortChannel Po102 of EndNode1. At step 1210, Virtual Router Redundancy Protocol (VRRP) protocol is configured on switch A of EndNode1. At step 1211, VRRP protocol is configured on switch B of EndNode1. At step 1212, a BGP neighbor is configured. At step 1213, the BGP neighbor is activated.

[0041] The effect of applying the foregoing process on the generic switch fabric resources of the provider datacenter(s) and POP(s) is to set up a configuration as shown Fig.13, after at step 1213, a Layer 3 virtual connection is implemented. Cross-reference is made between the physical elements of Fig.13 and the process steps of Fig.12A-B: a customer router 1301 is connected via connection 1307 to EtherChannel 2/1 of switch A of EntryNodel 901. This is done in BGP domain 65501. The route 1302, 1304 through evpn network 1303 in BGP domain 65001 has been set up, from EntryNodel to EndNode1. Connections 1305 and 1306 have been set up at steps 1208 and 1209.

[0042] Starting from step 1214 on Fig.12-C, the process allows to extend the previous Layer 3 virtual connection to a second POP. At step 1214, a second POP is created at EntryNode2 in a BGP area 65100, and VRF configured on EntryNode2. At step 1215, access to the physical interface on switch A of EntryNode2 902 and more particularly EtherChannel 1/1, is obtained. At step 1216, the physical interface (Et1/1) on switch A is configured for routed mode. At step 1217, the physical interface (Et1/1) on switch A is activated. At step 1218, an IP address is set up on EtherChannel 1/1. At step 1219, a BGP neighbor is configured. At step 1220, the BGP neighbor is activated.

[0043] The effect of applying the foregoing process on the generic switch fabric resources of the provider datacenter(s) and POP(s) is to set up a configuration as shown Fig.14, after at step 1220, a Layer 3 virtual connection is extended to a second POP. Cross-reference is made between the physical elements of Fig.14 and the process steps of Fig.12-C: a customer router 1401 is connected via connection 1407 to EtherChannel 1/1 of switch A of EntryNodel 902. This is done in BGP domain 65501. The route 1403, 1402 through evpn network 1303 in BGP domain 65001 has been set up, from EntryNode2 to EndNode1.

[0044] Starting from step 1221 on Fig.12-D, the process allows to extend the previous Layer 3 virtual connection to a second datacenter. At step 1221, VRF is configured on EndNode2. At step 1222, an IP address is configured on PortChannel Po101 of EndNode2. At step 1223, an IP address is configured on PortChannel Po102 of EndNode2. At step 1224, VRRP protocol is configured on switch A of EndNode2. At step 1225, VRRP protocol is configured on switch B of EndNode2.

[0045] The effect of applying the foregoing process on the generic switch fabric resources of the provider datacenter(s) and POP(s) is to set up a configuration as shown Fig.15, after at step 1225, a Layer 3 virtual connection is extended to a second datacenter. Cross-reference is made between the physical elements of Fig.15 and the process steps of Fig.12-D : connections 1505 and 1506 have been set up at steps 1222 and 1223.

[0046] The process according to the invention allows to ensure only valid and secure customer configurations are pushed to the provider datacenter(s) and POP(s). Indeed, the object modelling for the configuration, as well as the set of calls, syntax and associated parameters in the REST API, result in breaking down the configuration into a sequence of calls on objects, with each object being valid because its parents in hierarchy are also valid.

[0047] For example, to create a Layer 3 object, the call is:

```
POST /evpn/<id>/evpnEdge/<id>/layer3
```

[0048] However, for this call to not return in error, it is required that both evpn et evpnEdge objects be instantiated before the call.

[0049] First, evpn would have to be instantiated through the call:

```
POST /evpn
```

[0050] Then, evpnEdge:

```
POST /evpn/<id>/evpnEdge
```

[0051] Finally the above call POST /evpn/<id>/evpnEdge/<id>/layer3 would allow a Layer 3 object to be instantiated.

[0052] In the same token, a DELETE call applied on an evpnEdge object may not succeed if not preceded by a DELETE call applied on a Layer 3 object.

[0053] Incomplete and temporary configurations may thus be pushed to the provider datacenter(s) and POP(s), but not configurations that are not valid or not secure.

[0054] The process according to the invention further allows multi-tenancy in the provider datacenter(s) and POP(s). Indeed, each network configuration pushed to the datacenter(s) and POP(s) is fully independent of other network configurations, for example for other customers.

[0055] While the invention has been described in relation to Virtual Extensible LAN (Vxlan) protocol, it will be apparent to the person skilled in the art that the teachings herein equally apply to other protocols, such as the Multiprotocol Label Switching (MPLS) protocol.

[0056] While the invention has been described in relation to a number of POPs and of datacenters, it will be apparent to the person skilled in the art that the teachings herein equally apply to any number of POPs and datacenters, and of combinations thereof: for example one POP and two datacenters, or two POPs and one datacenter.

[0057] While the invention has been described in relation to Layer 2 and/or 3 protocols from one evpnEdge to the other, it will be apparent to the person skilled in the art that the teachings herein equally apply to other and even future protocols such as Layer 3v6: a child Layer class only needs to be added for the deployment method to apply in the context of other protocols without other changes to the method involved.

[0058] Figure 16 illustrates an example of a computing device that may be used to implement any of the methods described herein. 1600 represents a computing environment in accordance with an embodiment of the present technology is shown. In some embodiments, the computing environment 1600 may be implemented by any of a conventional personal computer, a computer dedicated to operating and/or monitoring systems relating to a data center, a controller and/or an electronic device (such as, but not limited to, a mobile device, a tablet device, a server, a controller unit, a control device, a monitoring device etc.) and/or any combination thereof appropriate to the relevant task at hand. In some embodiments, the computing environment 1600 comprises various hardware components including one or more single or multi-core processors collectively represented by a processor 1601, a solid-state drive 1602, a random access memory 1603 and an input/output interface 1604.

[0059] In some embodiments, the computing environment 1600 may also be a sub-system of one of the above-listed systems. In some other embodiments, the computing environment 1600 may be an "off the shelf' generic computer system. In some embodiments, the computing environment 1600 may also be distributed amongst multiple systems. The computing environment 1600 may also be specifically dedicated to the implementation of the present technology. As a person in the art of the present technology may appreciate, multiple variations as to how the computing environment 1600 is implemented may be envisioned without departing from the scope of the present technology.

[0060] Communication between the various components of the computing environment 1600 may be enabled by one or more internal and/or external buses 1605 (e.g. a PCI bus, universal serial bus, IEEE 1394 "Firewire" bus, SCSI bus, Serial-ATA bus, ARINC bus, etc.), to which the various hardware components are electronically coupled.

[0061] The input/output interface 1604 may allow enabling networking capabilities such as wire or wireless access. As an example, the input/output interface 1604 may comprise a networking interface such as, but not limited to, a network port, a network socket, a network interface controller and the like. Multiple examples of how the networking interface may be implemented will become apparent to the person skilled in the art of the present technology. For example, but without being limitative, the networking interface may implement specific physical layer and data link layer standard such as Ethernet, Fibre Channel, Wi-Fi or Token Ring. The specific physical layer and the data link layer may provide a base for a full network protocol stack, allowing communication among small groups of computers on the same local area network (LAN) and large-scale network communications through routable protocols, such as Internet Protocol (IP). According to implementations of the present technology, the solid-state drive 1602 stores program instructions suitable for being loaded into the random access memory 1603 and executed by the processor 1601 for executing operating data centers based on a generated machine learning pipeline. For example, the program instructions may be part of a library or an application.

Claims

1. A method of deploying a network configuration in a datacenter (904, 905, 906) having a point of presence (POP, 901, 902), comprising the steps of :

    - modelling the network configuration through objects, comprising :

        - a node object (813, 814, 826, 827) representing a point of interconnection in the network ;
        - an interface object (821, 822, 846, 847), being hierarchally inferior to the node object, and representing a connection to a point of interconnection ;
        - an evpnEdge object (823, 824, 816, 817) representing the transport to/from of a point of interconnection in the network ; and
        - a layer object (811, 822, 836, 837), being hierarchally inferior to the evpnEdge object, and representing the characterististics of the transported protocol to/from a point of interconnection ;

    - setting up the network configuration by a succession of commands on the objects ; and
    - pushing the set up network configuration to the POP and the datacenter.

2. The method of claim 1 in which the commands on the objects are REST API calls applied on the objects.

3. The method of claim 1 or 2 in which the pushing step is performed after a number of commands that is less than the total number of commands involved in the succession of commands on the objects.

4. The method of any of the claims 1 to 3 in which the layer object is adapted for a Layer 2 protocol to be the transported protocol.

5. The method of any of the claims 1 to 3 in which the layer object is adapted for a Layer 3 protocol to be the transported protocol.

6. The method of any of the claims 1 to 3 in which the transported protocol is VxLAN.

7. The method of any of the claims 1 to 3 in which the transported protocol is MPLS.

8. A computer-implemented system configured to perform the method of any one of claims 1 to 7.

9. A computer-readable medium comprising instructions causing to perform the method of any one of claims 1 to 7.

Amended claims in accordance with Rule 137(2) EPC.

1. A method of deploying a network configuration in a datacenter (904, 905, 906) having a point of presence (POP, 901, 902), comprising the steps of :

- modelling the network configuration through objects, comprising :

    - a node object (813, 814, 826, 827) representing a point of interconnection in the network, wherein the POP is a point of interconnection in the network;
    - an interface object (821, 822, 846, 847), being hierarchally inferior to the node object, and representing a connection to a point of interconnection ;
    - an evpnEdge object (823, 824, 816, 817) representing the transport to/from of a point of interconnection in the network ; and
    - a layer object (811, 822, 836, 837), being hierarchally inferior to the evpnEdge object, and representing the characteristics of the protocol of transport to/from a point of interconnection ;

- setting up the network configuration by a succession of commands on the objects ; and
- pushing the set up network configuration to the datacenter.

2. The method of claim 1 in which the commands on the objects are REST API calls applied on the objects.

3. The method of claim 1 or 2 in which the pushing step is performed after a number of commands that is less than the total number of commands involved in the succession of commands on the objects.

4. The method of any of the claims 1 to 3 in which the layer object is adapted for a Layer 2 protocol to be the transported protocol.

5. The method of any of the claims 1 to 3 in which the layer object is adapted for a Layer 3 protocol to be the transported protocol.

6. The method of any of the claims 1 to 3 in which the transported protocol is VxLAN.

7. The method of any of the claims 1 to 3 in which the transported protocol is MPLS.

8. A computer-implemented system configured to perform the method of any one of claims 1 to 7.

9. A computer-readable medium comprising instructions causing to perform the method of any one of claims 1 to 7.

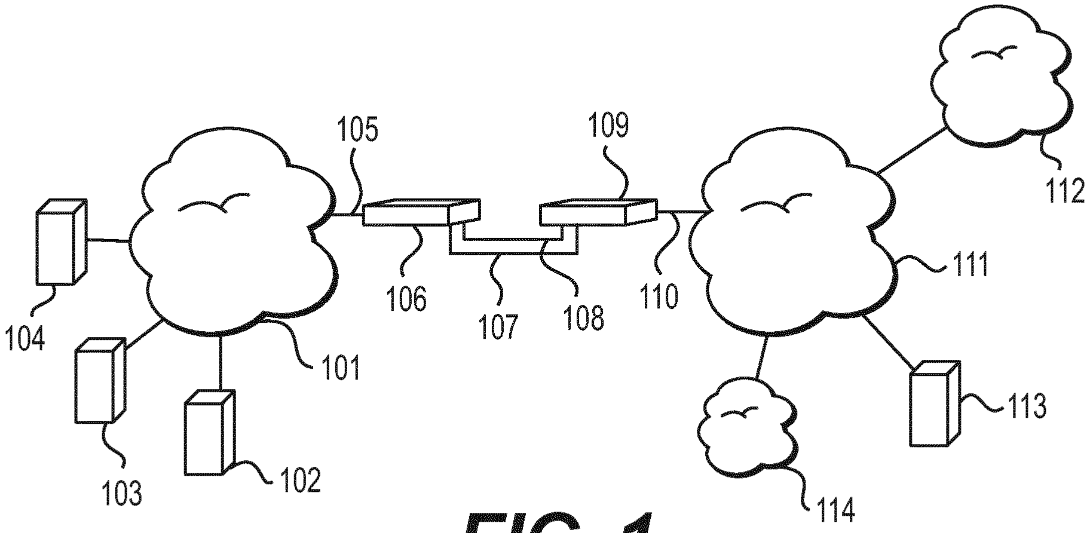


FIG. 1

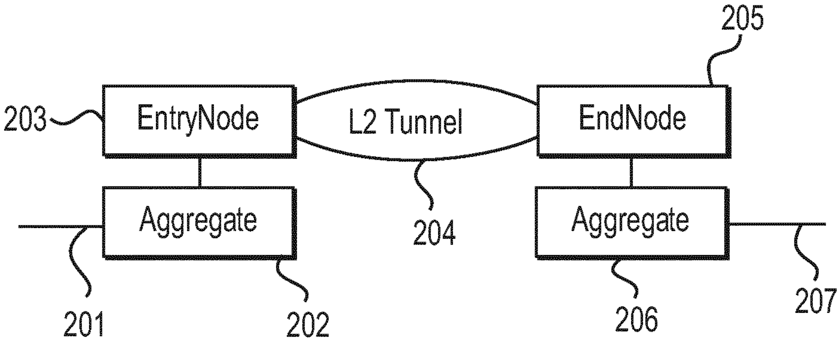


FIG. 2

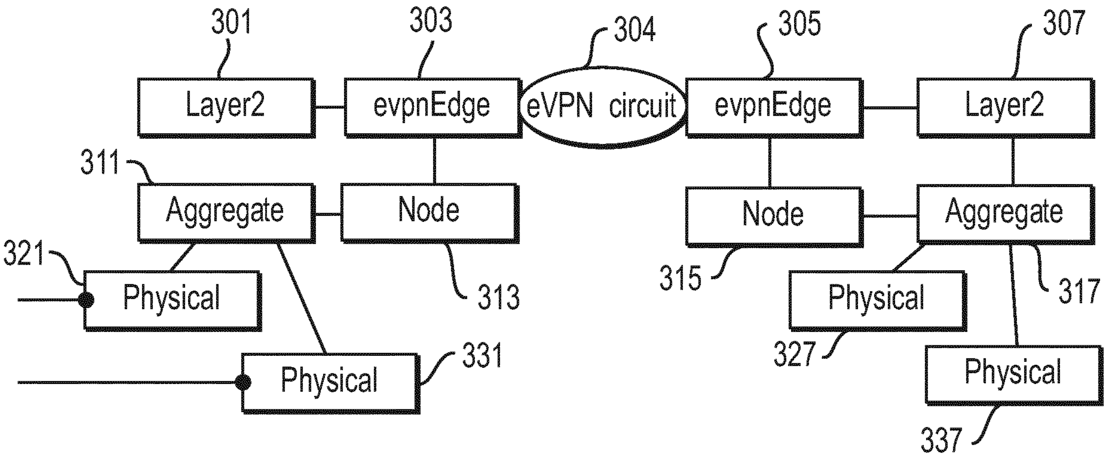


FIG. 3

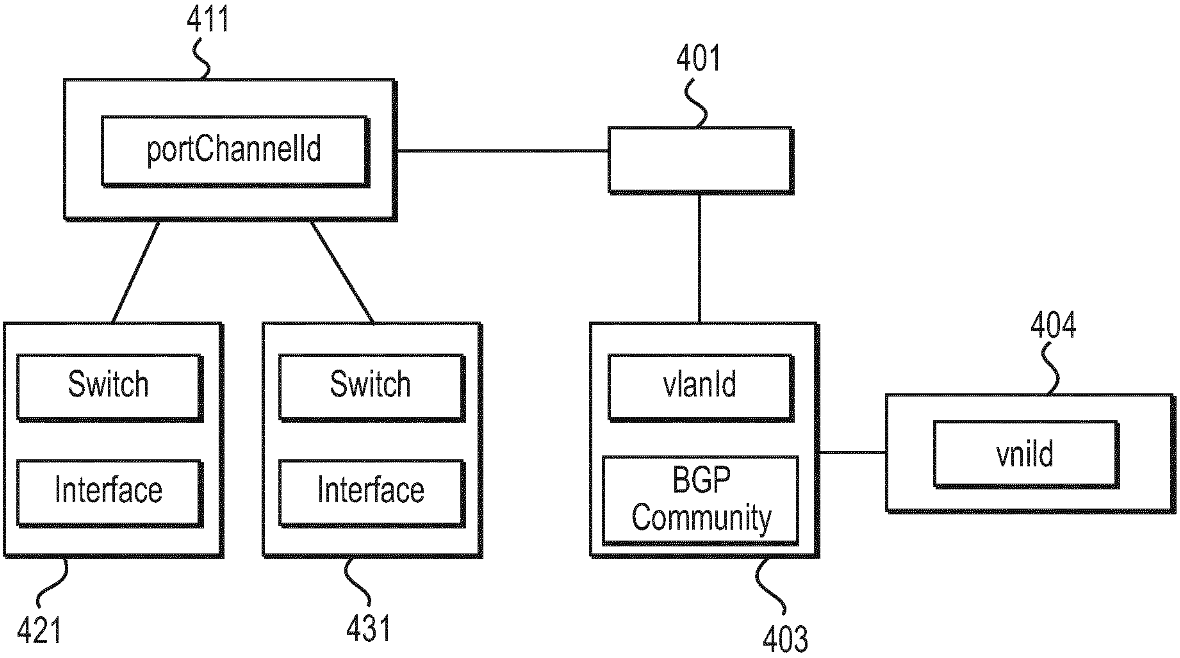

411
portChannelId
401
404
Switch
Interface
Switch
Interface
vlanId
BGP Community
vniId
421
431
403

FIG. 4

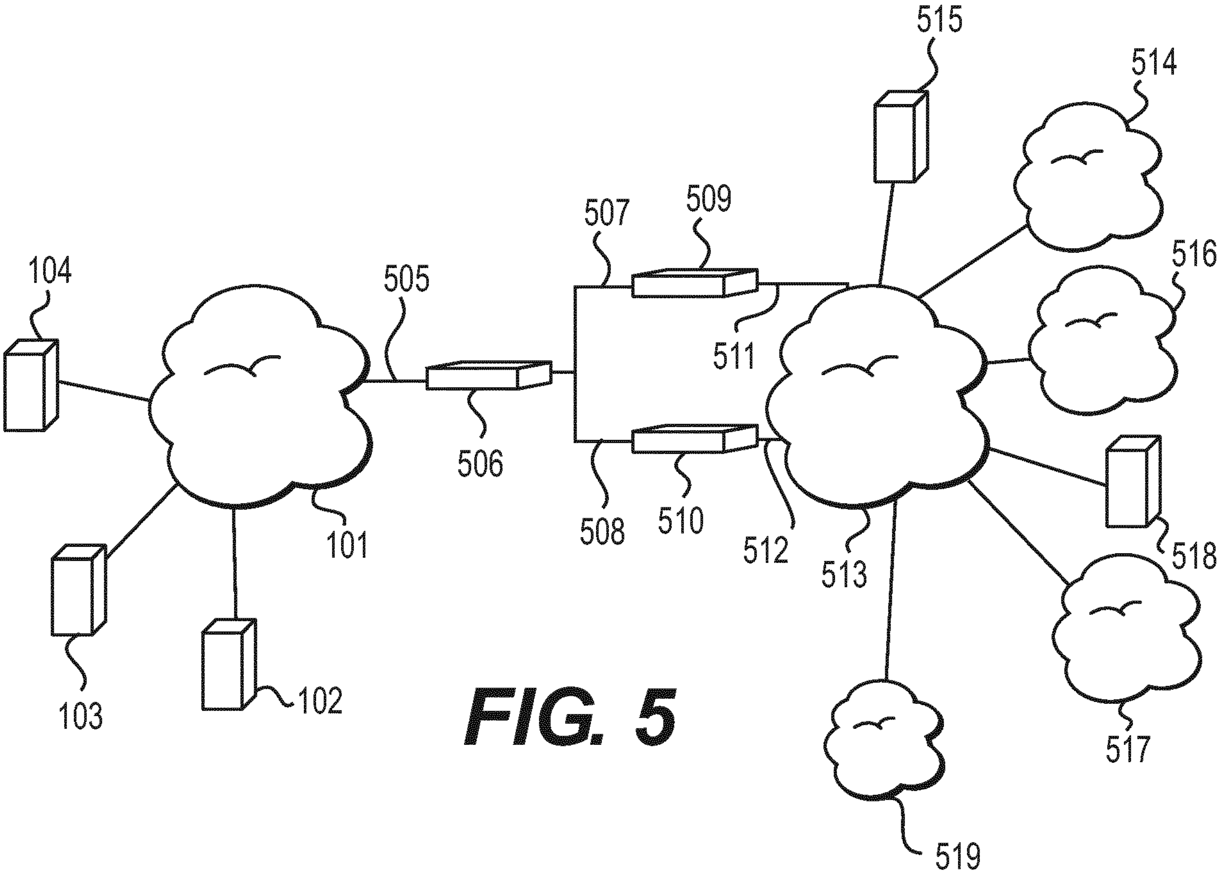

515
514
509
507
516
104
505
511
506
101
508
510
512
513
518
103
102
517
519

FIG. 5

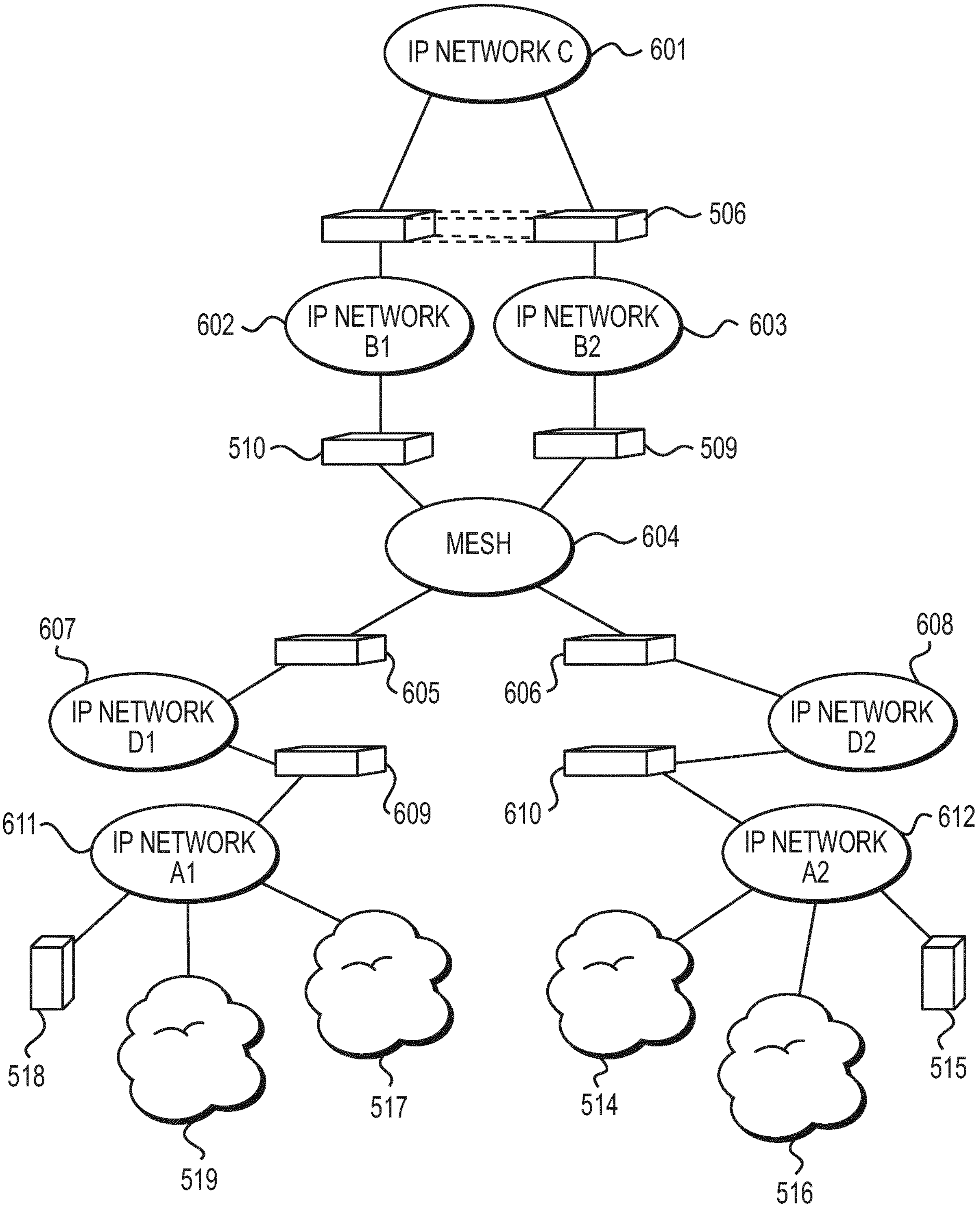

IP NETWORK C
601
506
IP NETWORK B1
602
IP NETWORK B2
603
510
509
MESH
604
607
605
606
608
IP NETWORK D1
IP NETWORK D2
609
610
611
IP NETWORK A1
IP NETWORK A2
612
518
519
517
514
516
515

FIG. 6

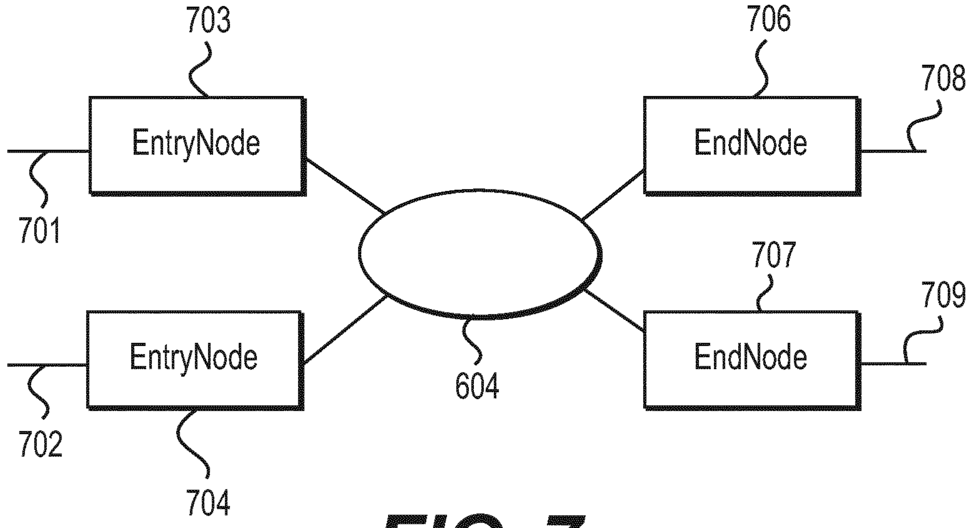

703
706
708
EntryNode
EndNode
701
707
709
EntryNode
EndNode
702
604
704

FIG. 7

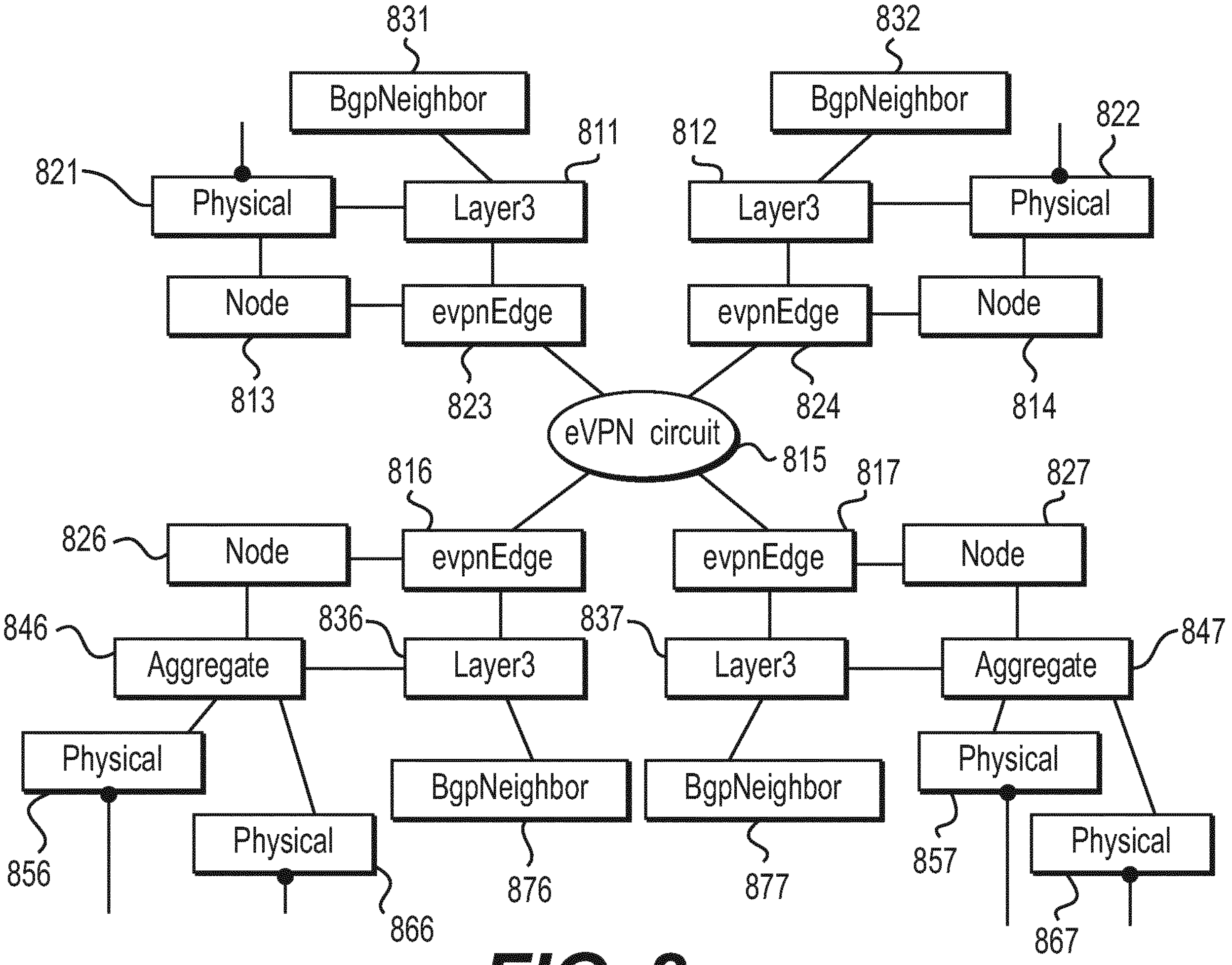

831
832
BgpNeighbor
BgpNeighbor
811
812
822
821
Physical
Layer3
Layer3
Physical
Node
evpnEdge
evpnEdge
Node
813
823
824
814
eVPN circuit
815
816
817
827
826
Node
evpnEdge
evpnEdge
Node
846
836
837
847
Aggregate
Layer3
Layer3
Aggregate
Physical
BgpNeighbor
BgpNeighbor
Physical
Physical
Physical
856
866
876
877
857
867

FIG. 8

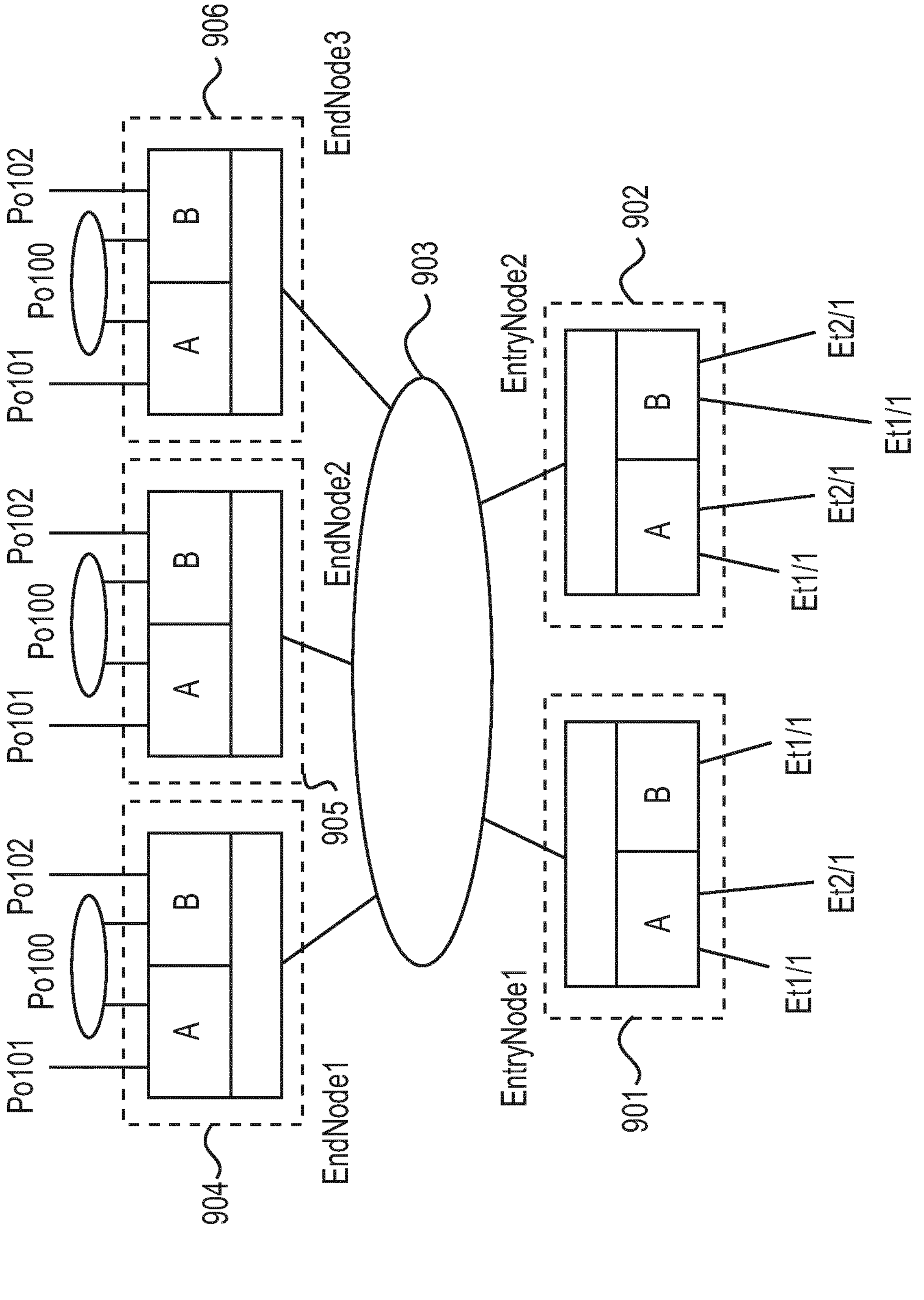

EndNode1
904
Po101
Po100
Po102
A
B
EndNode2
905
Po101
Po100
Po102
A
B
EndNode3
906
Po101
Po100
Po102
A
B
903
EntryNode1
901
A
B
Et1/1
Et2/1
Et1/1
EntryNode2
902
A
B
Et1/1
Et2/1
Et1/1
Et2/1

FIG. 9

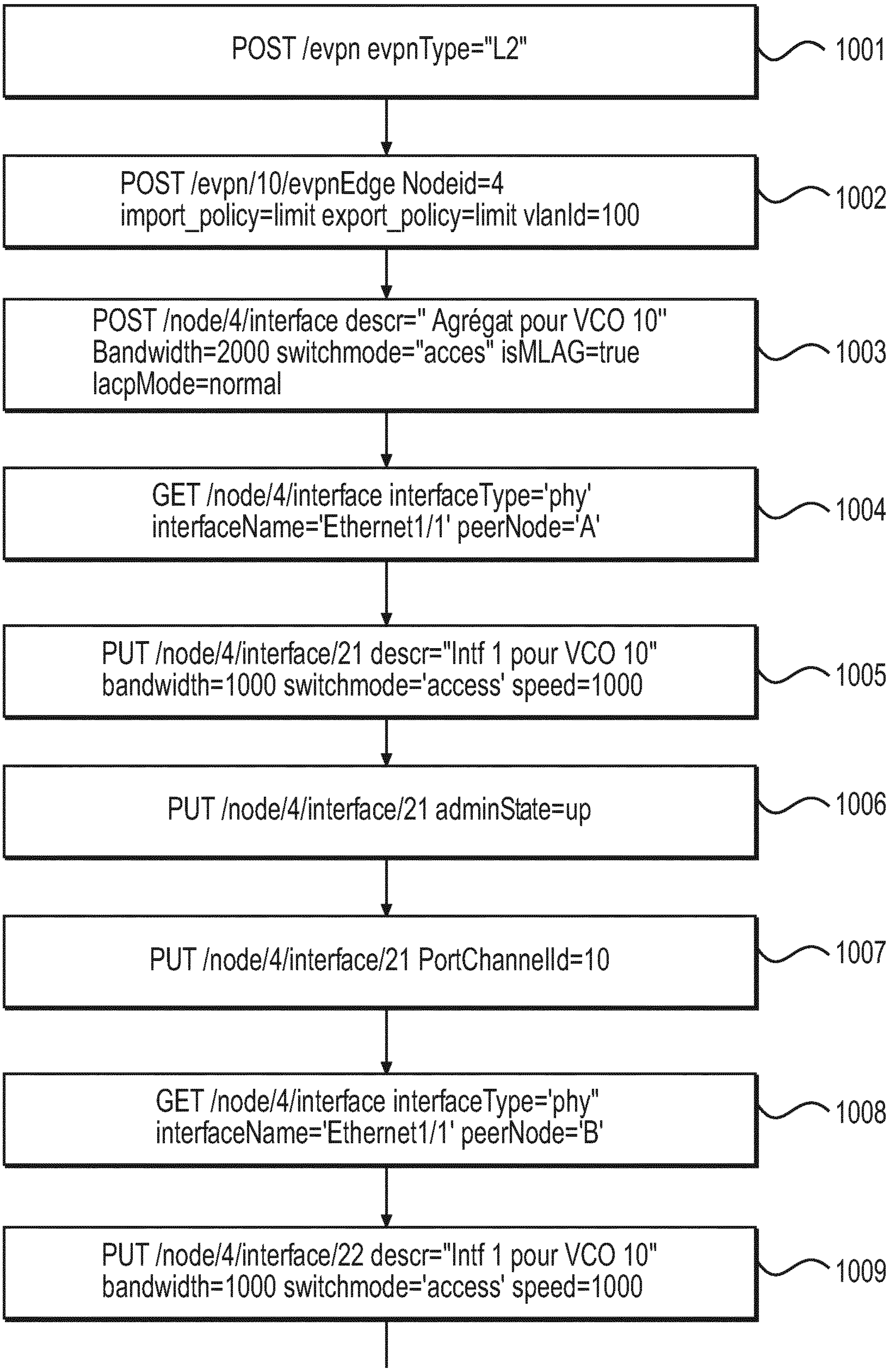

POST /evpn evpnType="L2"
1001
POST /evpn/10/evpnEdge Nodeid=4
import_policy=limit export_policy=limit vlanId=100
1002
POST /node/4/interface descr=" Agrégat pour VCO 10"
Bandwidth=2000 switchmode="acces" isMLAG=true
lacpMode=normal
1003
GET /node/4/interface interfaceType='phy'
interfaceName='Ethernet1/1' peerNode='A'
1004
PUT /node/4/interface/21 descr="Intf 1 pour VCO 10"
bandwidth=1000 switchmode='access' speed=1000
1005
PUT /node/4/interface/21 adminState=up
1006
PUT /node/4/interface/21 PortChannelId=10
1007
GET /node/4/interface interfaceType='phy"
interfaceName='Ethernet1/1' peerNode='B'
1008
PUT /node/4/interface/22 descr="Intf 1 pour VCO 10"
bandwidth=1000 switchmode='access' speed=1000
1009

FIG. 10A

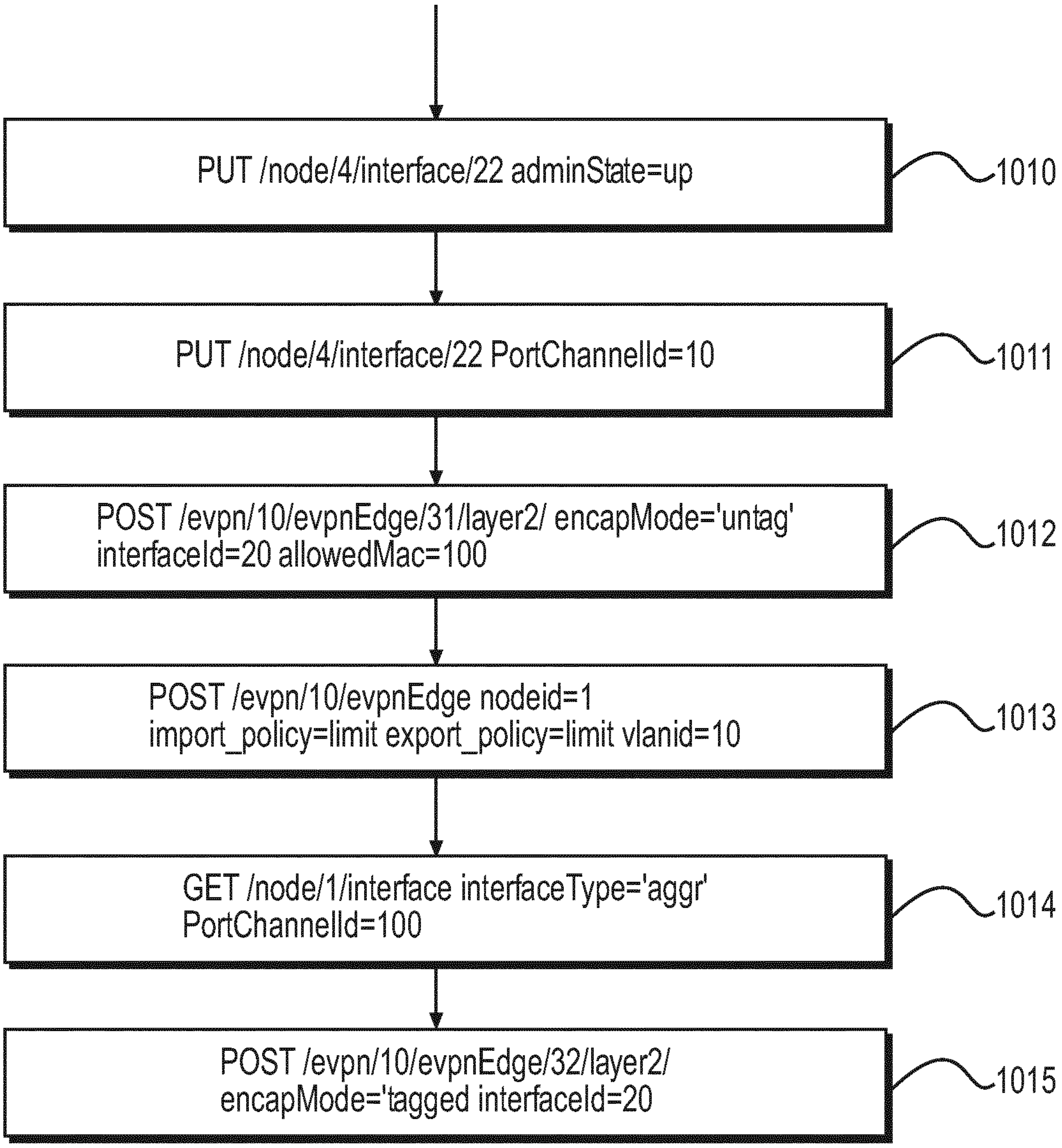

PUT /node/4/interface/22 adminState=up
1010
PUT /node/4/interface/22 PortChannelId=10
1011
POST /evpn/10/evpnEdge/31/layer2/ encapMode='untag' interfaceId=20 allowedMac=100
1012
POST /evpn/10/evpnEdge nodeid=1 import_policy=limit export_policy=limit vlanid=10
1013
GET /node/1/interface interfaceType='aggr' PortChannelId=100
1014
POST /evpn/10/evpnEdge/32/layer2/ encapMode='tagged interfaceId=20
1015

FIG. 10B

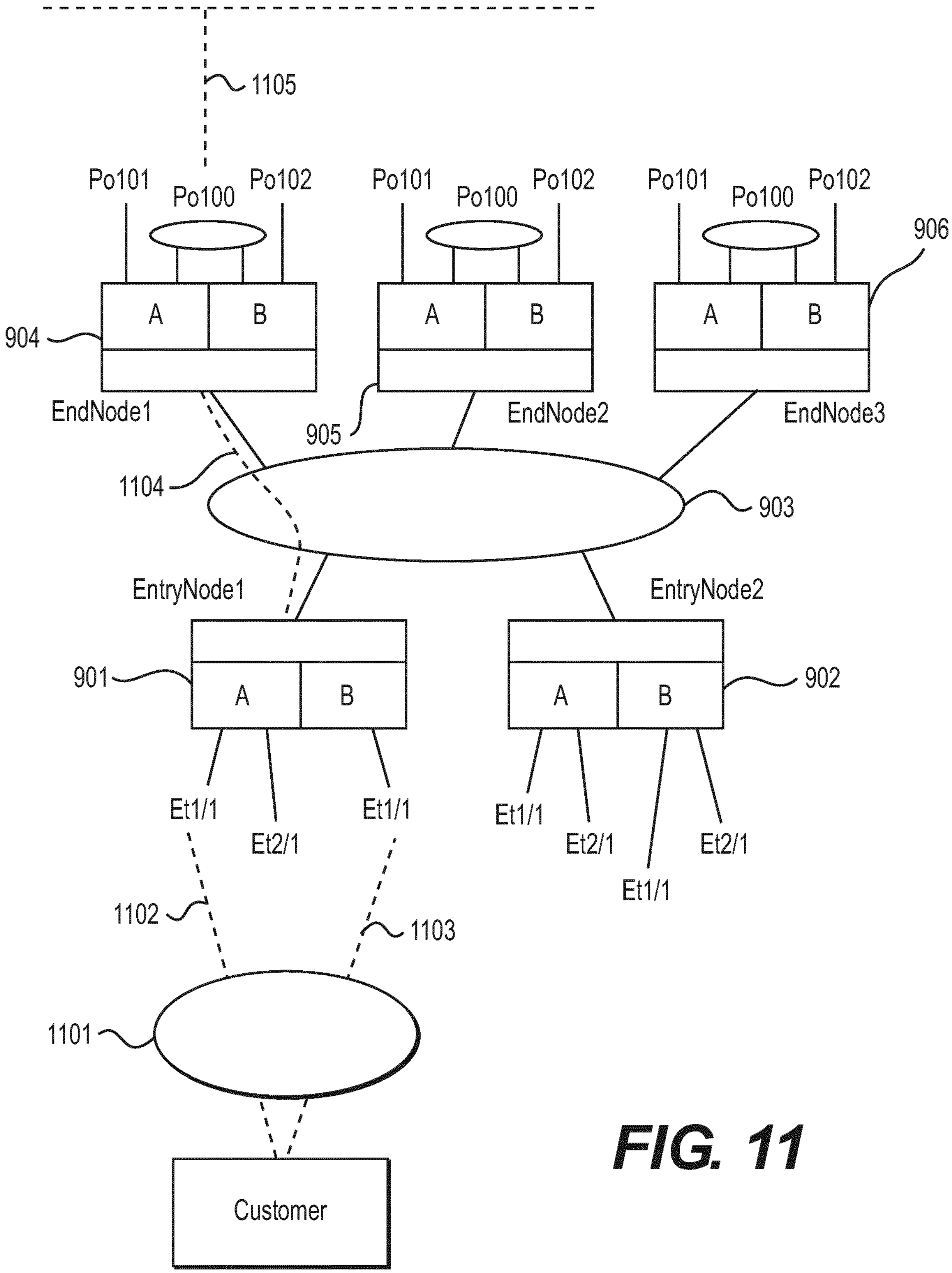

1105
Po101
Po100
Po102
Po101
Po100
Po102
Po101
Po100
Po102
906
A
B
A
B
A
B
904
EndNode1
905
EndNode2
EndNode3
1104
903
EntryNode1
EntryNode2
901
A
B
A
B
902
Et1/1
Et2/1
Et1/1
Et1/1
Et2/1
Et1/1
Et2/1
1102
1103
1101
Customer

FIG. 11

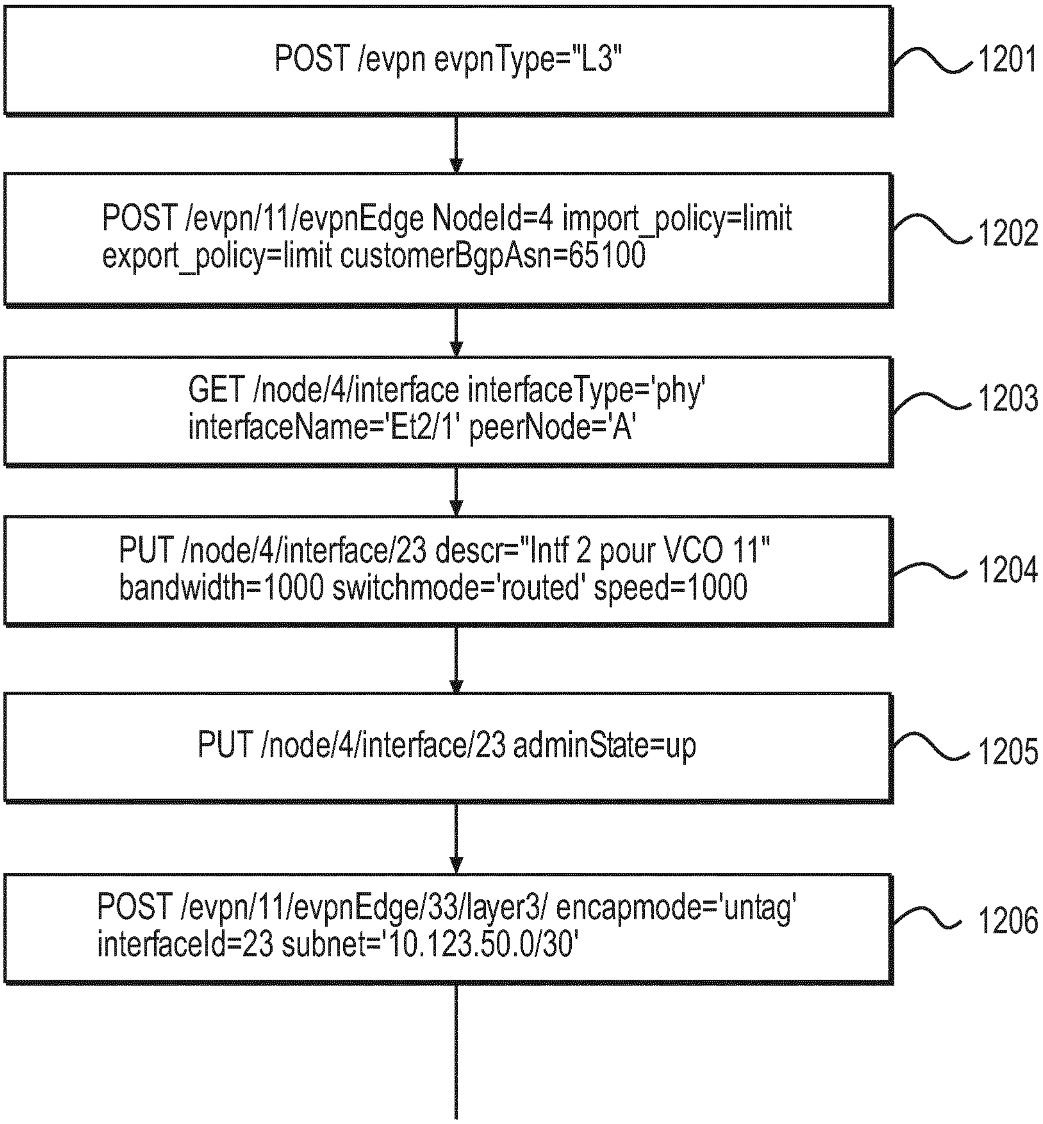
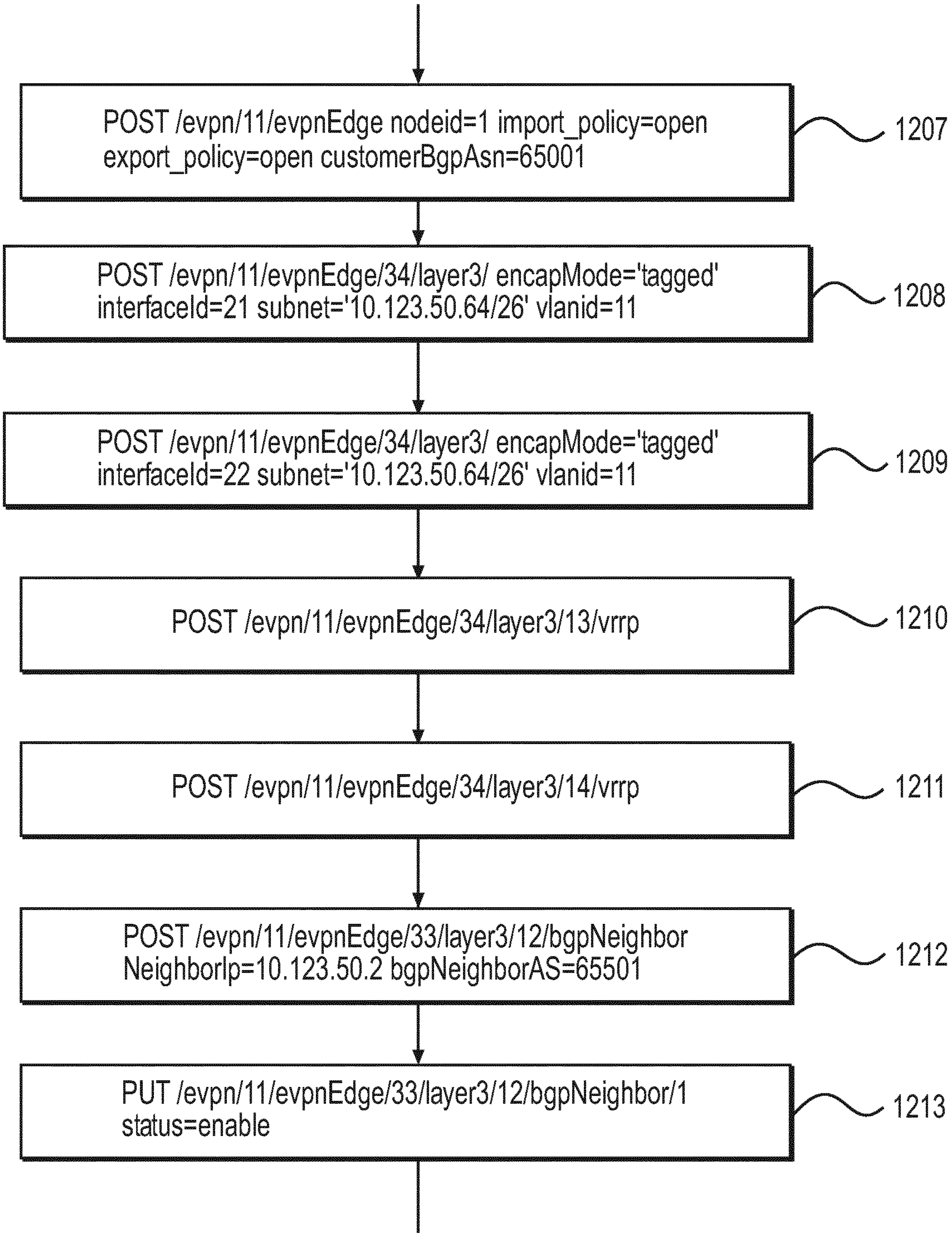

POST /evpn evpnType="L3"
1201
POST /evpn/11/evpnEdge NodeId=4 import_policy=limit export_policy=limit customerBgpAsn=65100
1202
GET /node/4/interface interfaceType='phy' interfaceName='Et2/1' peerNode='A'
1203
PUT /node/4/interface/23 descr="Intf 2 pour VCO 11" bandwidth=1000 switchmode='routed' speed=1000
1204
PUT /node/4/interface/23 adminState=up
1205
POST /evpn/11/evpnEdge/33/layer3/ encapmode='untag' interfaceId=23 subnet='10.123.50.0/30'
1206

FIG. 12A



```
POST /evpn/11/evpnEdge nodeid=1 import_policy=open
export_policy=open customerBgpAsn=65001
```

```
POST /evpn/11/evpnEdge/34/layer3/ encapMode='tagged'
interfaceId=21 subnet='10.123.50.64/26' vlanid=11
```

```
POST /evpn/11/evpnEdge/34/layer3/ encapMode='tagged'
interfaceId=22 subnet='10.123.50.64/26' vlanid=11
```

```
POST /evpn/11/evpnEdge/34/layer3/13/vrrp
```

```
POST /evpn/11/evpnEdge/34/layer3/14/vrrp
```

```
POST /evpn/11/evpnEdge/33/layer3/12/bgpNeighbor
NeighborIp=10.123.50.2 bgpNeighborAS=65501
```

```
PUT /evpn/11/evpnEdge/33/layer3/12/bgpNeighbor/1
status=enable
```

FIG. 12B

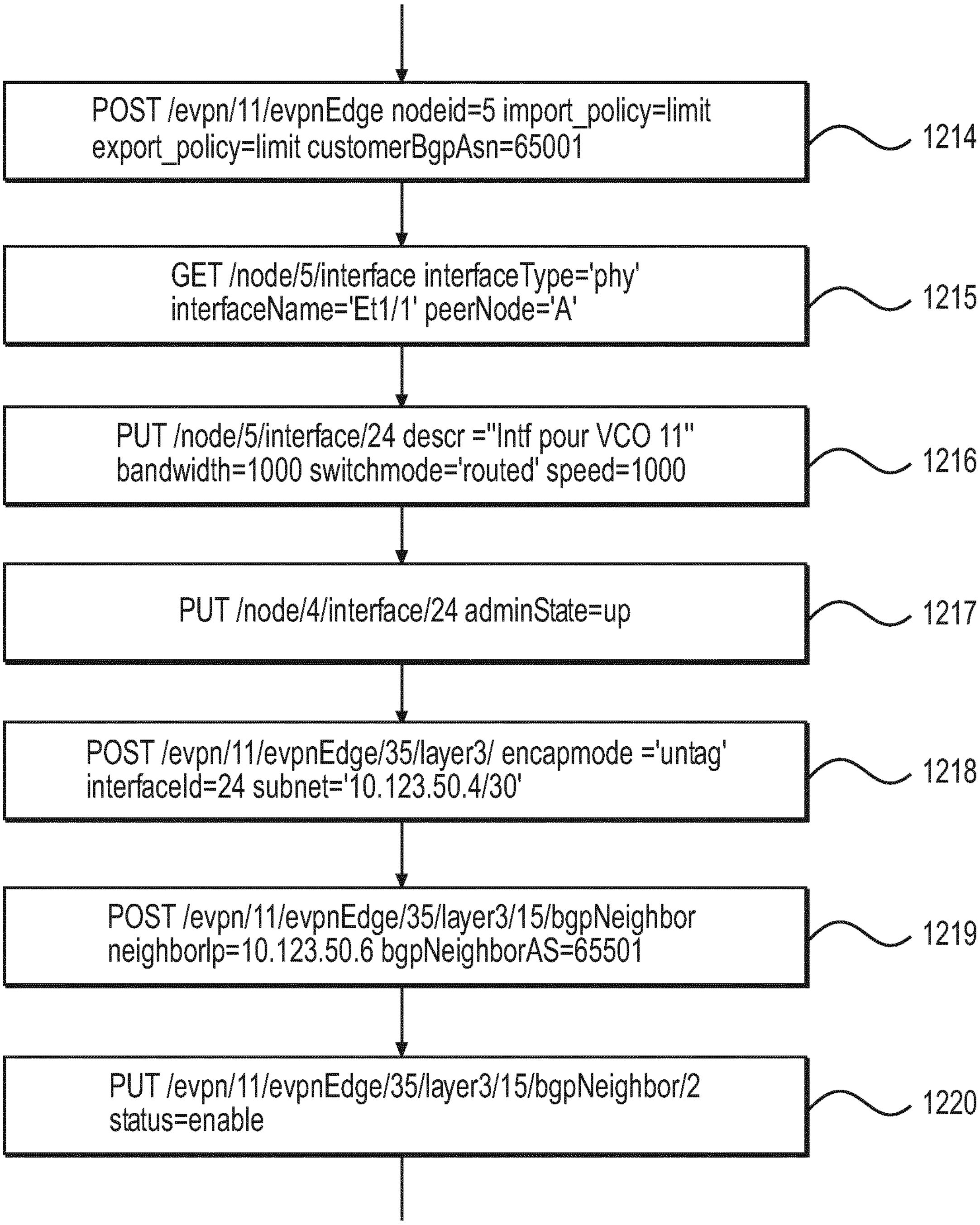

POST /evpn/11/evpnEdge nodeid=5 import_policy=limit export_policy=limit customerBgpAsn=65001
1214
GET /node/5/interface interfaceType='phy' interfaceName='Et1/1' peerNode='A'
1215
PUT /node/5/interface/24 descr ="Intf pour VCO 11" bandwidth=1000 switchmode='routed' speed=1000
1216
PUT /node/4/interface/24 adminState=up
1217
POST /evpn/11/evpnEdge/35/layer3/ encapmode ='untag' interfaceId=24 subnet='10.123.50.4/30'
1218
POST /evpn/11/evpnEdge/35/layer3/15/bgpNeighbor neighborIp=10.123.50.6 bgpNeighborAS=65501
1219
PUT /evpn/11/evpnEdge/35/layer3/15/bgpNeighbor/2 status=enable
1220

FIG. 12C

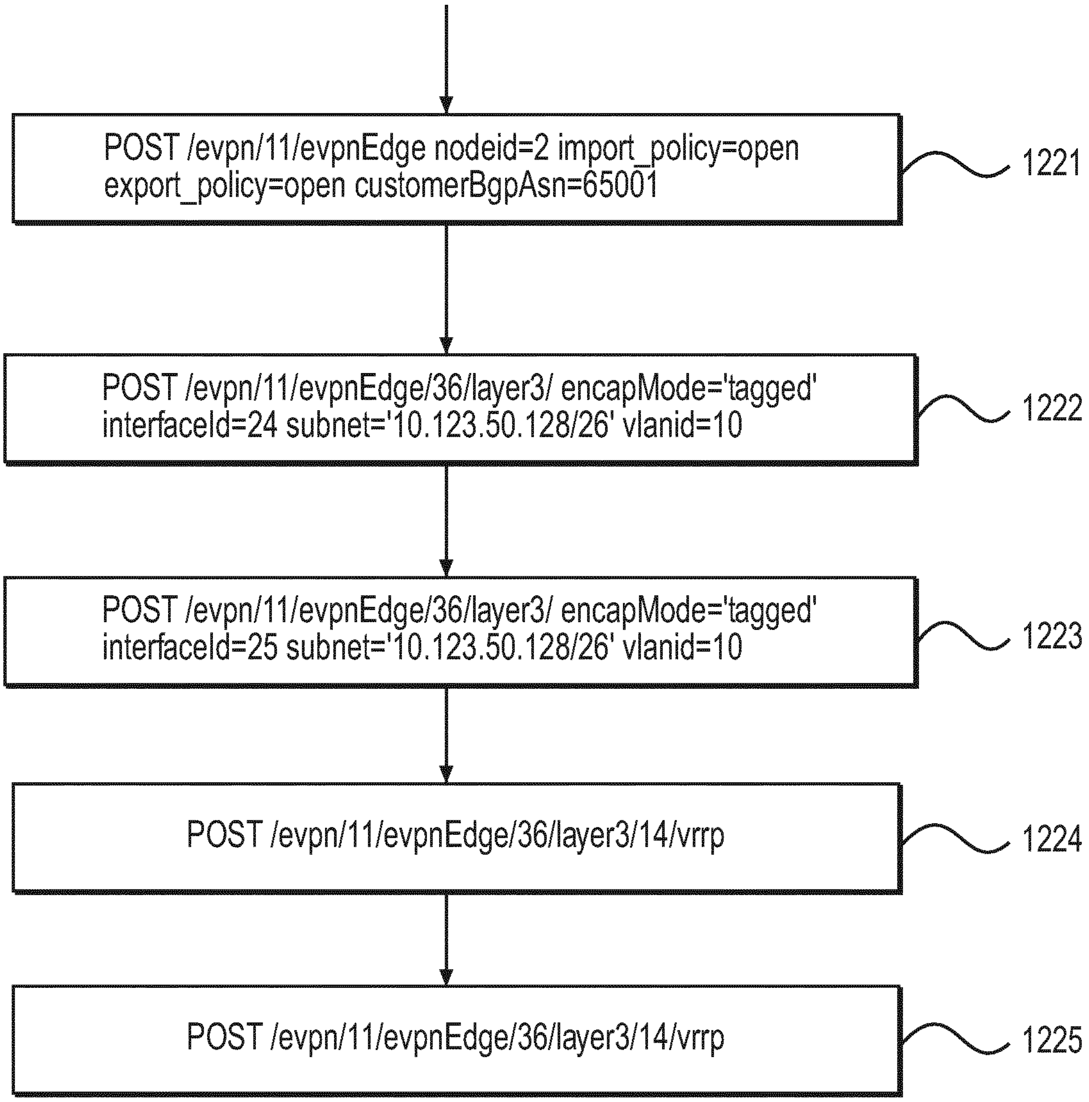

POST /evpn/11/evpnEdge nodeid=2 import_policy=open export_policy=open customerBgpAsn=65001
1221
POST /evpn/11/evpnEdge/36/layer3/ encapMode='tagged' interfaceId=24 subnet='10.123.50.128/26' vlanid=10
1222
POST /evpn/11/evpnEdge/36/layer3/ encapMode='tagged' interfaceId=25 subnet='10.123.50.128/26' vlanid=10
1223
POST /evpn/11/evpnEdge/36/layer3/14/vrrp
1224
POST /evpn/11/evpnEdge/36/layer3/14/vrrp
1225

FIG. 12D

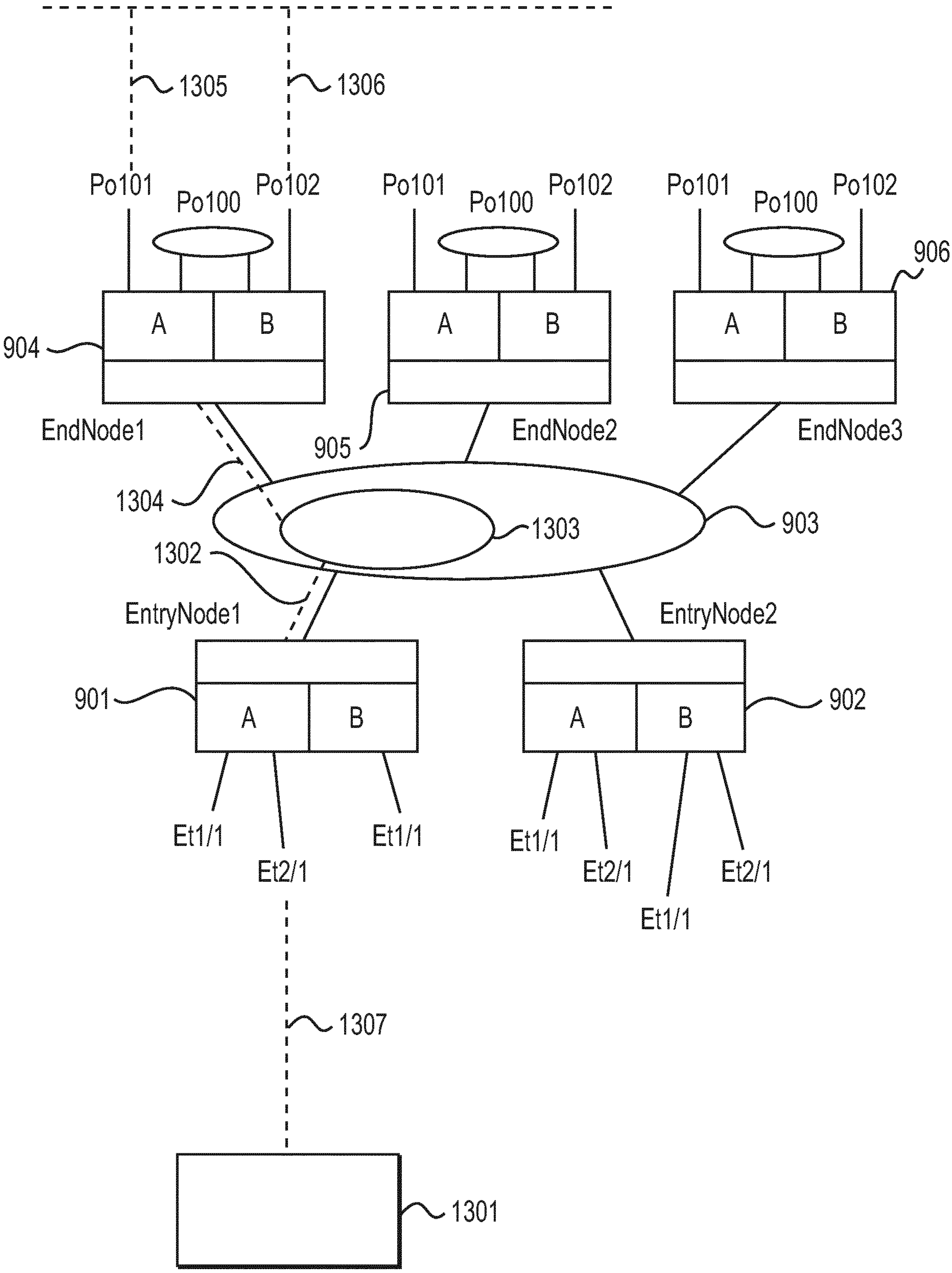

1305
1306
Po101
Po100
Po102
Po101
Po100
Po102
Po101
Po100
Po102
906
904
A
B
A
B
A
B
EndNode1
905
EndNode2
EndNode3
1304
1303
903
1302
EntryNode1
EntryNode2
901
A
B
A
B
902
Et1/1
Et2/1
Et1/1
Et1/1
Et2/1
Et2/1
Et1/1
1307
1301

FIG. 13

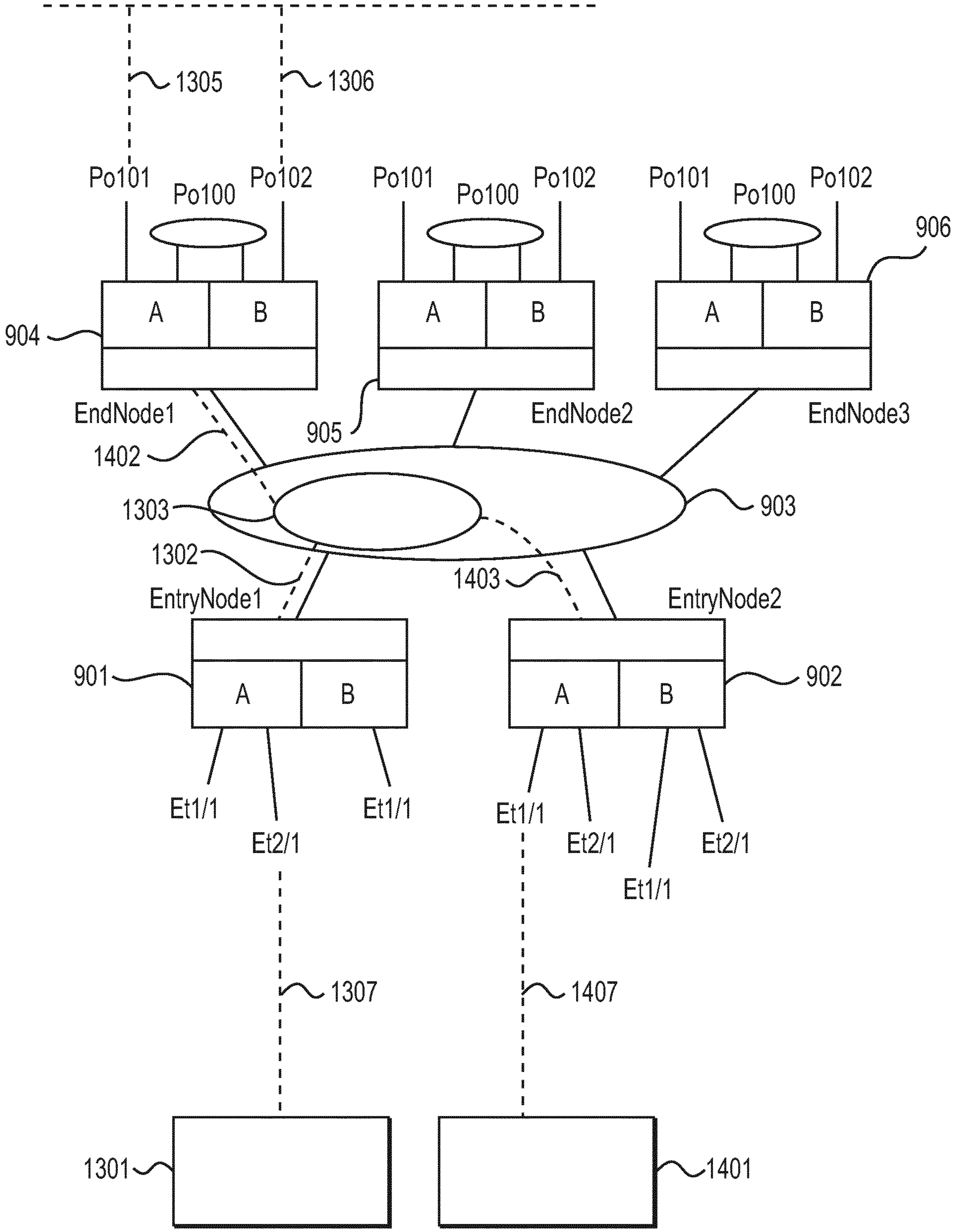

1305
1306
Po101
Po100
Po102
Po101
Po100
Po102
Po101
Po100
Po102
906
904
A
B
A
B
A
B
EndNode1
905
EndNode2
EndNode3
1402
1303
903
1302
1403
EntryNode1
EntryNode2
901
A
B
A
B
902
Et1/1
Et2/1
Et1/1
Et1/1
Et2/1
Et2/1
Et1/1
1307
1407
1301
1401

FIG. 14

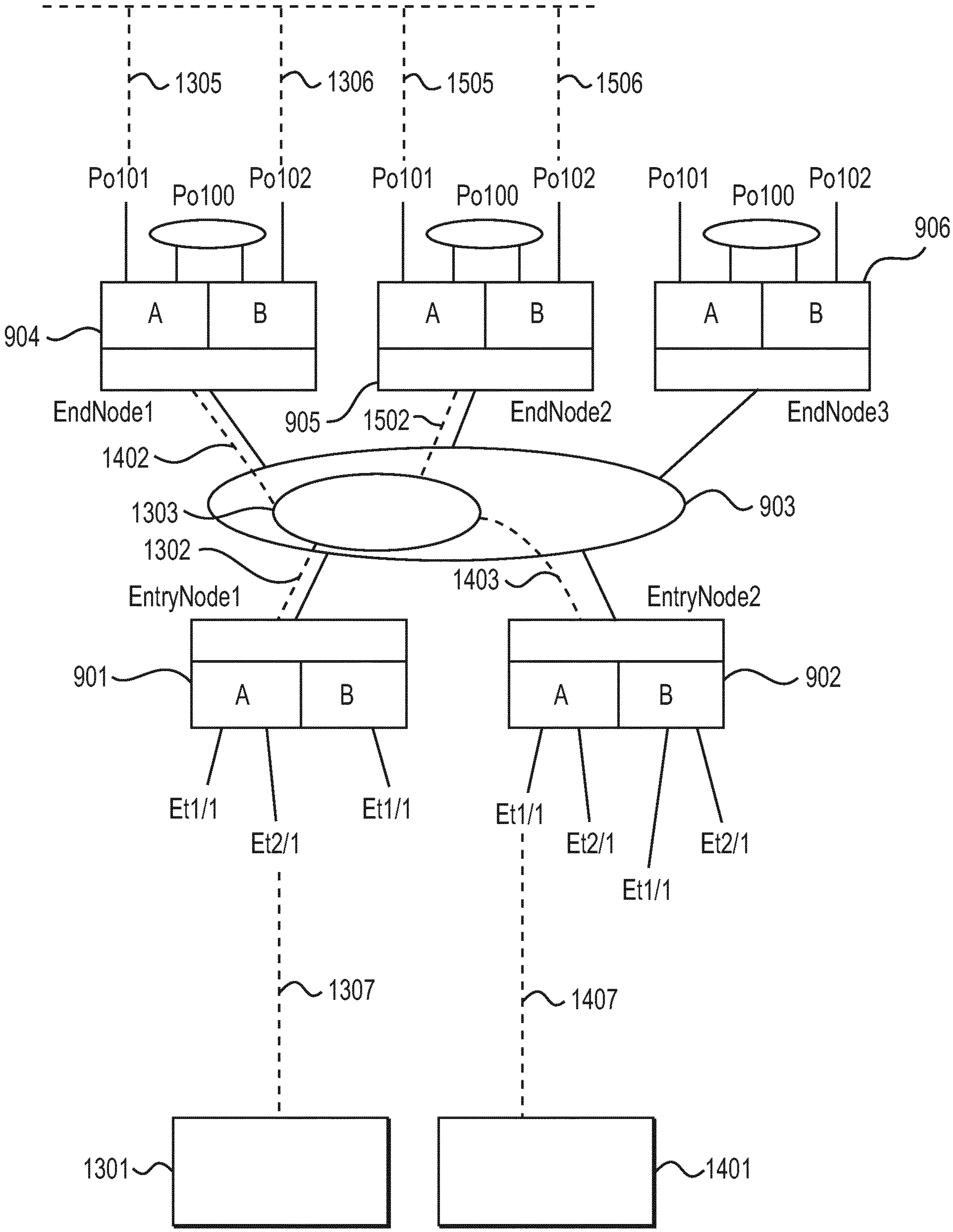

1305
1306
1505
1506
Po101
Po100
Po102
Po101
Po100
Po102
Po101
Po100
Po102
906
A
B
A
B
A
B
904
EndNode1
905
1502
EndNode2
EndNode3
1402
1303
903
1302
EntryNode1
1403
EntryNode2
901
A
B
A
B
902
Et1/1
Et2/1
Et1/1
Et1/1
Et2/1
Et1/1
Et2/1
1307
1407
1301
1401

FIG. 15

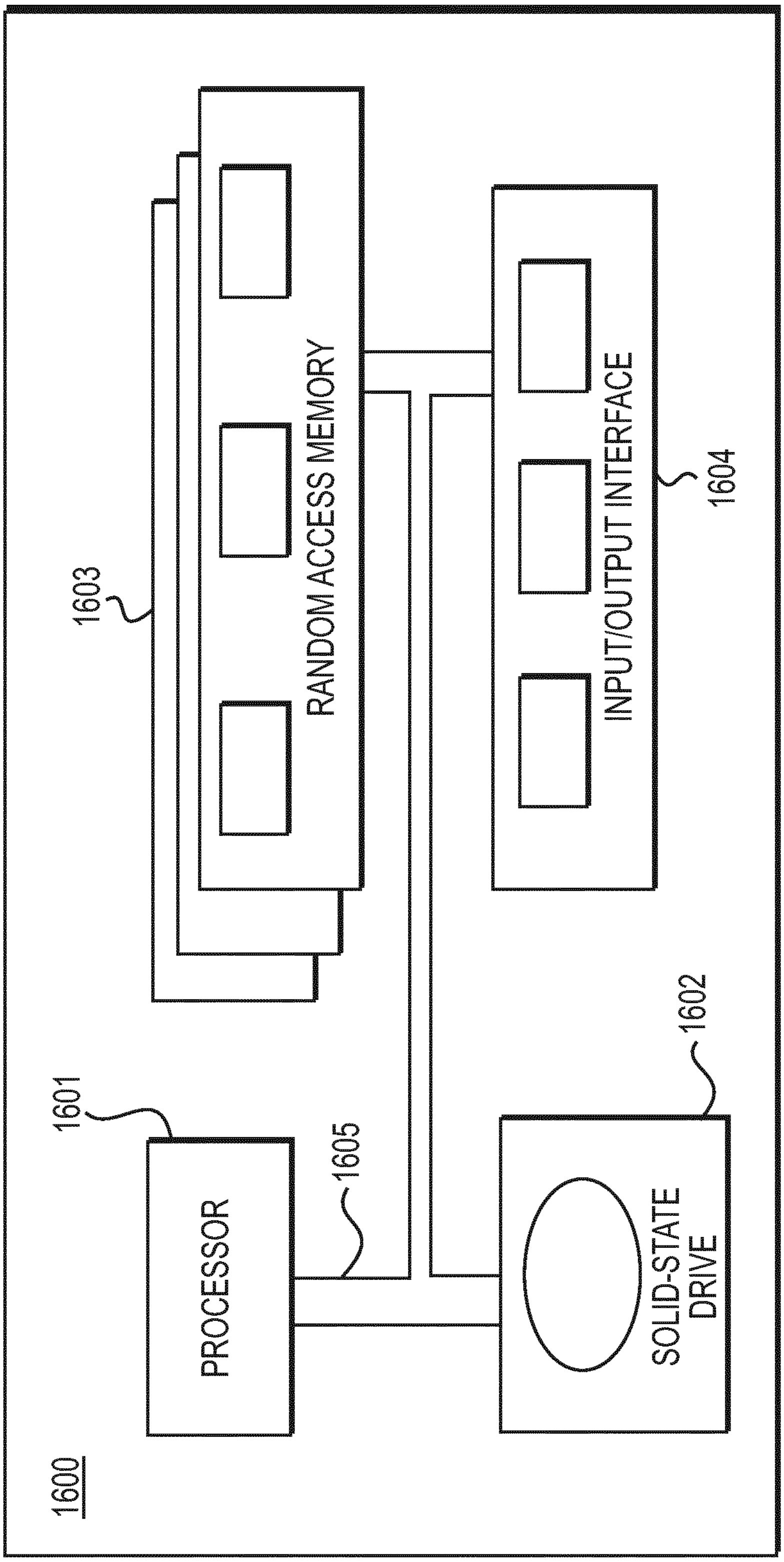

1600
1601
PROCESSOR
1605
1603
RANDOM ACCESS MEMORY
INPUT/OUTPUT INTERFACE
1604
1602
SOLID-STATE DRIVE

FIG. 16

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 19 31 5011

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	ISRAEL R ET AL: "Configuration Management of Large IP Telephony Networks", 20000410; 20000410 - 20000414, 10 April 2000 (2000-04-10), pages 435-446, XP010376700, * abstract * * Chapter 2. Network Architecture; page 437; figure 1 * * page 438; figure 3 * * page 440 - page 443; figures 4,6 * -----	1-9	INV. H04L12/24
A	US 2016/308762 A1 (TENG JUXIANG [US] ET AL) 20 October 2016 (2016-10-20) * abstract * * column 9, line 25 - column 15, line 47; figures 3A, 3B,4-6 * -----	1-9	
			TECHNICAL FIELDS SEARCHED (IPC)
			H04L

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	15 July 2019	Von Der Straten, G

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons
....
& : member of the same patent family, corresponding document

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 19 31 5011

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

15-07-2019

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2016308762 A1	20-10-2016	AU 2016248307 A1	22-12-2016
		BR 112016029187 A2	22-08-2017
		CA 2951940 A1	20-10-2016
		CN 106464592 A	22-02-2017
		EP 3155765 A1	19-04-2017
		JP 6491241 B2	27-03-2019
		JP 2017524290 A	24-08-2017
		SG 11201610056R A	29-11-2017
		US 2016308762 A1	20-10-2016
		US 2017093702 A1	30-03-2017
		WO 2016168577 A1	20-10-2016

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82