



(51) International Patent Classification:
H04L 12/717 (2013.01)

(21) International Application Number:
PCT/JP2013/001913

(22) International Filing Date:
21 March 2013 (21.03.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2012-068286 23 March 2012 (23.03.2012) JP

(71) Applicant (for all designated States except US): **NEC CORPORATION** [JP/JP]; 7-1, Shiba 5-chome, Minato-ku, Tokyo, 1088001 (JP).

(72) Inventors; and

(71) Applicants (for US only): **ASHIDA, Yuta** [JP/JP]; c/o NEC Corporation, 7-1, Shiba 5-chome, Minato-ku, Tokyo, 1088001 (JP). **KOIDE, Toshio** [JP/JP]; c/o NEC Corporation, 7-1, Shiba 5-chome, Minato-ku, Tokyo, 1088001 (JP).

(74) Agent: **KUDOH, Minoru**; 6F, KADOYA BLDG., 24-10, Minamiooi 6-chome, Shinagawa-ku, Tokyo, 1400013 (JP).

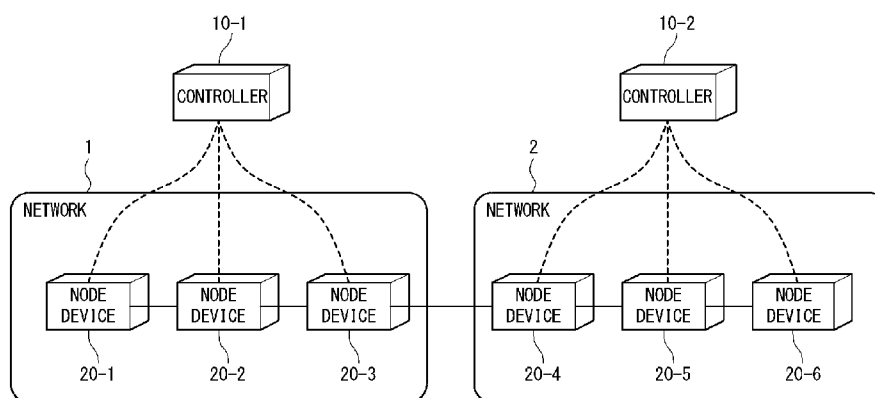
(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, 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, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

(54) Title: SYSTEM AND METHOD FOR COMMUNICATION



(57) Abstract: A communications system includes: a first node device provided in a first network; a first controller controlling the first node device; a second node device provided in a second network and connected to the first node device; and a second controller controlling the second node device. The first controller sets the first node device with a processing rule according to which packets transferred between the first and second controllers are processed. The second controller sets the second node device with a processing rule according to which the packets are processed. The first and second controllers exchanges the packets each other through at least the first and second node devices.

Description

Title of Invention: SYSTEM AND METHOD FOR COMMUNICATION

Technical Field

[0001] The present invention relates to a communications system and more particularly relates to a communications system managed by a centralized management, in which controllers control packet transfers among network devices.

Background Art

[0002] One problem of conventional network devices is that flexible load control, such as load distribution and load concentration, cannot be achieved by an external control. This makes it difficult to monitor and improve the system behavior in a large-scale network, causing a problem that a modification in the system design and/or configuration requires a large cost.

[0003] One proposed approach for solving this problem is separation of the packet transfer function and the route control function which are both conventionally implemented by a network device. For example, in a system in which the packet transfer function is assigned to network devices and the control function is assigned to a controller provided separately from the network devices, the controller can perform centralized management of packet transfer, which allows establishing a network with high flexibility.

[0004] (CD-Separated Network)

One proposed function-separated network is a CD-separated network (where C stands for control plane and d stands for data plane), in which a controller operating on the control plane controls node devices operating on the data plane.

[0005] One example of the CD-separated network is an OpenFlow network which is based on the OpenFlow technique, in which the route control in the network is achieved by controlling switches by a controller. Details of the OpenFlow technique are disclosed in non-patent documents 1 and 2. Note that the OpenFlow network should be construed as one example.

[0006] (OpenFlow Network)

In an OpenFlow network, a controller, such as an OpenFlow controller (OFC), controls the behavior of node devices, such as OpenFlow switches (OFSs), by processing route control information (or a flow table) which describes the route control of the node devices.

[0007] Controllers and node devices are connected via control channels (communication channels for control) called "secure channels", which are communication paths

protected by using dedicated lines or an SSL (Secure Socket Layer) technique. Controllers and node devices exchange OpenFlow messages defined in the OpenFlow protocol via control channels.

- [0008] In the OpenFlow network, each node device is provided in the OpenFlow network and controlled by the controller; each node device may be an edge switch or a core switch. A series of packet transfers from the receipt of a packet at an ingress edge switch to the transmission at an egress edge switch in the OpenFlow network is referred to as "flow". In the OpenFlow network, communications are each regarded as an end-to-end flow, and the route control, the trouble recovery, and the load distribution and optimization are carried out in units of flows.
- [0009] It should be noted that the frame should be regarded as an alternative of the packet. The difference between the packet and the frame only lies in the difference in the protocol data unit (PDU). The packet is the PDU of the TCP/IP (transmission control protocol / internet protocol). On the other hand, the frame is the PDU of the "Ethernet" (registered trademark).
- [0010] The route control information (or the flow table) includes a set of: identifying conditions (identifying rules) to identify packets to be treated as a flow; statistical information that indicates the number of times in which packets comply or match with the identifying conditions (or the identifying rules); and processing rules (or flow entries) which define a group of contents of processing (or actions) to be performed on packets.
- [0011] The identifying conditions (or the identifying rules) of each processing rule (or each flow entry) are defined by various combinations of any or all of data of respective protocol hierarchies included in the header region (or the header field) of the packet, and distinguishable one another by using these data. One example of data of the respective protocol hierarchies may include the destination address, the source address, the destination port, the source port. Note that the above-described addresses may be defined by a MAC (media access control) address or an IP address. Also, data of the ingress port may be also used as the identifying conditions (or the identifying rules) of the processing rules (or the flow entries). Furthermore, the identifying conditions (or the identifying rules) of the processing rules (or the flow entries) may be set in the form of a representation in which some or all of the values of the header region of the packet to be treated as the flow are represented by using a normal representation, a wildcard character "*" or the like.
- [0012] The contents of processing (or the action) of a processing rule (or a flow entry) indicates an operations such as "output to a particular port", "discarding" and "rewriting of header". For example, if the contents of processing (or the action) of a processing rule (or a flow entry) indicates identification information of an output port

(an output port number and the like), the node device outputs the packet to the indicated port, and if not, the node device discards the packet. Instead, if the contents of processing (or the action) of the processing rule (or the flow entry) indicates the header information, the node device rewrites the header of the packet on the basis of the header information.

[0013] A node device in the OpenFlow network performs the contents of processing (or the action) of a processing rule (or a flow entry) on a group of packets (or a series of packets) that complies with the identifying conditions (or the identifying rules) of the processing rule (or the flow entry).

[0014] For example, when receiving a packet, an OpenFlow switch (OFS), which corresponds to a node device in the OpenFlow network, retrieves the processing rule (or the flow entry) associated with the identifying conditions (or the identifying rule) complying with the header information of the received packet from the route control information (of the flow table). If the processing rule (flow entry) complying with the received packet is found out as a result of the retrieval, the contents of processing (or the action) described in the action field of the processing rule (or the flow entry) is performed on the received packet. If no processing rule (or the flow entry) complying with the received packet is found as a result of the retrieval, on the other hand, the received packet is judged as the first packet. In this case, inquiry information of the received packet is transmitted to an OpenFlow controller (OFC), which corresponds to the controller in the OpenFlow network, via a control channel to request for determining the route of the packet based on the source and destination of the received packet; this is followed by receiving a processing rule (or a flow entry) for attaining the packet transfer along the determined route and then updating the route control information (or the flow table).

[0015] It should be noted that an initial state processing rule (or a default entry) is registered in the route control information (or the flow table), where the initial state processing rule describes identifying conditions (or identifying rules) of a low priority which are defined so as to comply with the header information of any packets. If no other processing rule (or flow entry) complying with the received packet is found, the received packet complies with the initial state processing rule (or the default entry). The contents of processing (or the action) of the initial state processing rule (or the default entry) is defined to instruct to transmit inquiry information of the received packet to the OpenFlow controller (OFC).

[0016] As mentioned above, the OpenFlow switch (OFS) determines processing to be done on a packet in accordance with the setting of processing rules (or flow entries) set by the OpenFlow controller (OFC). In particular, the "output" processing, in which a packet is outputted to a specified interface, is often used as the processing. It should be

noted that the specified interface is not limited to a physical interface; the specified interface is not limited to the physical port and may be a virtual port.

[0017] As thus described, control of packets is attained by centralized control of OpenFlow switches (OFS) by an OpenFlow controller (OFC) in the OpenFlow network. One issue is that one OpenFlow controller (OFC) can control only a limited number of OpenFlow switches (OFS). Accordingly, an increase in the scale of the network, which causes an increase in the number of the OpenFlow switches (OFS), may result in that calculation of processing rules (flow entries) in the OpenFlow controller (OFC) and the like becomes a bottleneck of the network quality.

[0018] One approach to address this may be connecting a plurality of OpenFlow networks, each including one Openflow controller (OFC) and a plurality of OpenFlow switches (OFS) controlled by the OpenFlow controller, when the scale of the network is increased.

Citation List

Non Patent Literature

[0019] [NPL 1] Nick Mckeown and Other Seven Persons, "OpenFlow: Enabling Innovation in Campus Networks", online, Retrieval on January 23, 2012, Internet (URL: <http://www.openflow.org/documents/openflow-wp-latest.pdf>)

[NPL 2] "OpenFlow Switch Specification, Version 1.1.0 Implemented", online, Retrieval on February 28, 2012, Internet (URL: <http://www.openflowswitch.org/documents/openflow-spec-v1.1.0.pdf>)

Summary of Invention

[0020] An interconnection of a plurality of networks each managed by a centralized management, such as OpenFlow networks, requires exchanging and sharing of information related to route control among the controllers to control the traffic over the networks. One approach to achieve this may be using an existing routing protocol, such as the OSPF (open shortest path first) protocol and the BGP (border gateway protocol) or other data sharing protocols in order to exchange and share information related to the route control among the controllers.

[0021] The use of these protocols to exchange information related to the route control among the controllers, however, requires establishing a connection between adjacent controllers.

[0022] In an aspect of the present invention, a communications system includes: a first node device provided in a first network; a first controller controlling the first node device; a second node device provided in a second network and connected to the first node device; and a second controller controlling the second node device. The first controller sets the first node device with a processing rule according to which packets transferred

between the first and second controllers are processed. The second controller sets the second node device with a processing rule according to which the packets are processed. The first and second controllers exchange the packets each other through at least the first and second node devices.

- [0023] In another aspect of the present invention, a communication method includes:
- setting a first node device provided in a first network by a first controller controlling the first node device with a processing rule according to which packets transferred between the first controller and a second controller controlling a second node provided in a second network are processed;
 - setting the second node device by the second controller with a processing rule according to which the packets are processed;
 - establishing a connection between the first and second node devices; and
 - exchanging the packets between the first and second controllers through at least the first and second node devices.
- [0024] In another aspect of the present invention, a program is provided for causing a computer or a network device to perform the operations of respective devices in the above-described communication method. The program may be stored in a storage device or a non-transitory recording medium.
- [0025] The present invention enables establishing a communication connection between controllers of adjacent networks.

Brief Description of Drawings

- [0026] [fig.1]Fig. 1 shows an exemplary configuration of a communications system according to the present invention.
- [fig.2]Fig. 2 shows an exemplary system configuration in a first embodiment.
- [fig.3]Fig. 3 shows an exemplary configuration of a controller.
- [fig.4]Fig. 4 shows an exemplary configuration example of a node device.
- [fig.5]Fig. 5 shows an exemplary configuration of an LLDP frame in the first embodiment.
- [fig.6A]Fig. 6A shows exemplary contents of a processing rule defined for a first network.
- [fig.6B]Fig. 6B shows exemplary contents of a processing rule defined for a second network.
- [fig.7]Fig. 7 shows an exemplary configuration of an LLDP frame in a second embodiment.
- [fig.8]Fig. 8 shows an exemplary configuration of a controller in a third embodiment.
- [fig.9A]Fig. 9A shows exemplary contents of a sorting rule defined for the controller of the first network.

[fig.9B]Fig. 9B shows exemplary contents of a sorting rule defined for the controller of the second network.

[fig.10A]Fig. 10A shows an exemplary configuration of the first network of the communications system in a fourth embodiment.

[fig.10B]Fig. 10B shows an exemplary configuration of the second network of the communications system in the fourth embodiment.

[fig.11A]Fig. 11A shows exemplary contents of a sorting rule defined for the controller of the first network.

[fig.11B]Fig. 11B shows exemplary contents of a sorting rule defined for the controller of the second network.

[fig.12]Fig. 12 shows an exemplary hardware configuration of a controller according to the present invention.

[fig.13]Fig. 13 shows an exemplary hardware configuration of a node device according to the present invention.

Description of Exemplary Embodiments

[0027] Embodiments of the present invention are described in the following with an example in which OpenFlow networks, which are a sort of networks managed by a centralized management, are used. It should be noted, however, that the present invention is not limited to a communication system in which OpenFlow networks are used.

[0028] <Embodiment>

Embodiments of the present invention are described below in details with reference to the attached drawings.

[0029] <System Configuration>

A description is first given of an exemplary configuration of a communications system in one embodiment of the present invention with reference to Fig. 1.

[0030] In one embodiment, a communications system includes controllers 10 and node devices 20.

[0031] The controllers 10 are information processing apparatuses which control the node devices 20.

[0032] The node devices 20 are communication devices provided in networks.

[0033] The controllers 10 and the node devices 20 are connected through control channels. The controllers 10 and the node devices 20 communicate with each other by using OpenFlow messages defined in an OpenFlow protocol via the control channels.

[0034] Each node device 20 is connected to the adjacent node device 20 via a data communication link such as a LAN (local area network) and the like. Moreover, a node device 20 which operates as an edge switch is adapted to have a connection with a host (a client, a server or the like) or an external network device which does not belong to the

network in which the node device 20 is provided.

[0035] It should be noted that the controllers 10 and the node devices 20 are not limited to physical machines; the controllers 10 and the node devices 20 may be a virtual machine (VM).

[0036] Fig. 1 shows controllers 10-1 and 10-2, as examples of the controllers 10. Fig. 1 also shows node devices 20-1 to 20-6 as examples of the node devices 20.

[0037] The controller 10-1 is connected via a control channel to each of the node devices 20-1 to 20-3.

[0038] The controller 10-2 is connected via a control channel to each of the node devices 20-4 to 20-6.

[0039] The node devices 20-1 to 20-6 are each connected to one or more adjacent node devices via one or more data communication links, such as LANs.

[0040] In this embodiment, the node device 20-1 is connected to the node device 20-2 via a data communication link. The node device 20-2 is connected to the node device 20-3 via a data communication link. The node device 20-3 is connected to the node device 20-4 via a data communication link. The node device 20-4 is connected to the node device 20-5 via a data communication link. The node device 20-5 is connected to the node device 20-6 via a data communication link.

[0041] The node devices 20-1 to 20-3 are provided in a network 1. That is, each of the node devices 20-1 to 20-3 is arranged in the network 1.

[0042] The node devices 20-4 to 20-6 are arranged in the network 2. That is, each of the node devices 20-4 to 20-6 is arranged in the network 2.

[0043] Accordingly, that the networks 1 and 2 are connected to each other via the data communication link connecting the node device 20-3 and the node device 20-4.

[0044] It should be noted that the control channels and the data communication links may be a wired communication link or a wireless communication link.

[0045] (Identification Information of Controllers and Node Devices)

In this embodiment, the controller 10-1 is assigned with a controller ID "CPID1" as its own identification information. The controller 10-2 is assigned with a controller ID "CPID2" as its own identification information. The node device 20-1 is assigned with a node device ID "DPID1" as its own identification information. The node device 20-2 is assigned with a node device ID "DPID2" as its own identification information. The node device 20-3 is assigned with a node device ID "DPID3" as its own identification information. The node device 20-4 is assigned with a node device ID "DPID4" as its own identification information. The node device 20-5 is assigned with a node device ID "DPID5" as its own identification information. The node device 20-6 is assigned with a node device ID "DPID6" as its own identification information.

[0046] <First Embodiment>

A first embodiment of the present invention is described below.

[0047] In this embodiment, the controllers 10 are each connected to one of the node devices 20 via a data communication link, similarly to the connections between the node devices 20. That is, the controllers 10 may also operate as a "host" connected to any of the node devices 20.

[0048] (System Configuration in First Embodiment)

An exemplary configuration of a communications system in this embodiment is described below with reference to Fig. 2.

[0049] In this embodiment, the controller 10-1 is further connected to the node device 20-1 through a data communication link. It should be noted that the controller 10-1 may be logically connected to the node device 20-1 by tunneling or the like in an actual implementation.

[0050] Similarly, the controller 10-2 is further connected to the node device 20-6 via a data communication link. It should be noted that the controller 10-2 may be logically connected to the node device 20-2 by tunneling or the like in an actual implementation.

[0051] It should be noted that the system configuration is not limited to the above-described embodiment.

[0052] (Configuration of Controllers)

An exemplary configuration of each controller 10 is described with reference to Fig. 3. Note that this exemplary configuration is common to both of the controllers 10-1 and 10-2.

[0053] Each controller 10 includes a node device control section 11 and an interface management section 15.

[0054] The node device control section 11 controls node devices 20 via control channels. For example, the node device control section 11 executes a software program which allows operating as an OpenFlow controller (OFC) in an OpenFlow network. Here, the node device control section 11 monitors and manages interface units contained in each of the node devices 20 via the control channels and specifies processing rules (flow entries) of packets transmitted and received by the interface units for the node devices 20. One example of the contents of a processing rule (a flow entry) of the packets may include an instruction of outputting to an interface unit or a node device control section 11 specified on the basis of the feature of each packet received by the interface unit, an instruction of outputting to an interface unit specified when the packet prepared by the node device control section 11 is received, and the like.

[0055] The interface management section 15 manages a network interface provided in the controller 10. In this embodiment, the interface management section 15 is adapted to have a connection to a network interface within any of the node devices 20 via data a communication link connected to the network interface within the controller 10.

- [0056] In this embodiment, the node device control section 11 is connected to the interface management section 15. Thus, the node device control section 11 can communicate with any of the node devices 20 via the interface management section 15.
- [0057] (Configuration of Node device Control Unit)
- An exemplary configuration of the node device control section 11 is described below.
- [0058] The node device control section 11 includes a topology management section 111, a control message processing section 112, a node communication section 113, an adjacency discovery section 114, an identifying condition calculation section 115, a route calculation section 116, a processing rule calculation section 117, a processing rule management section 118 and a processing rule storage section 119.
- [0059] The topology management section 111 manages topology information and boundary information of its own network. The topology management section 111 is connected to the interface management section 15.
- [0060] The control message processing section 112 prepares control messages in accordance with controls to be performed for the node devices 20. The control message processing section 112 also analyzes control messages received from the node device 20.
- [0061] The node communication section 113 is connected to node devices 20 via control channels to communicate with the node devices 20. The node communication section 113 transmits control messages to the node devices 20 in response to control message transmission requests from the control message processing section 112. The node communication section 113 also transfers control messages received from the node devices 20 to the control message processing section 112.
- [0062] The adjacency discovery section 114 discovers a node device 20 located on the boundary with a different network and the controller 10 in the different network.
- [0063] The identifying condition calculation section 115 calculates identifying conditions (or the identifying rules) of packets. In this embodiment, the identifying condition calculation section 115 calculates identifying conditions (or identifying rules) of packets transferred between the controllers 10 on the basis of the topology information and the boundary information of its own network. The identifying condition calculation section 115 is also adapted to, when receiving inquiry information of the first packet from a node device 20, calculate identifying conditions (or an identifying rule) of the packet on the basis of the header information of the packet.
- [0064] The route calculation section 116 calculates transfer routes of packets. Here, the route calculation section 116 determines the end points of the route to be used in the packet transfer between the controllers 10 on the basis of the topology information and the boundary information of the networks, and calculates the route to connect the determined end points.

- [0065] The processing rule calculation section 117 calculates processing rules (or flow entries) on the basis of the identifying conditions calculated by the identifying condition calculation section 115, and the routes calculated by the route calculation section 116.
- [0066] The processing rule management section 118 manages the processing rules (or the flow entries). The processing rule management section 118 registers information related to the processing rules (or the flow entries) in the processing rule storage section 119, correlating information related to the processing rules with identification information (processing rule IDs) of the processing rules (or the flow entries). The processing rule management section 118 also requests the control message processing section 112 to set the node devices 20 with processing rules (or flow entries).
- [0067] The processing rule storage section 119 stores: the information with regard to the processing rules (or the flow entries) set to the respective node devices 20 under the management of the controller 10; and the identification information (the processing rules ID) of the processing rules (or the flow entries). In an actual implementation, the processing rule storage section 119 may store copies or master tables of the route control information (or the flow table) of the respective node devices 20.
- [0068] (Configuration of Node Devices)
- An exemplary configuration of each node device 20 is described with reference to Fig. 4. Note that this exemplary configuration is common to all of the node devices 20-1 to 20-6.
- [0069] Each node device 20 includes a communication unit 21 and one or more interface units 22 (one shown).
- [0070] The communication unit 21 is connected to a controller 10 via a control channel to exchange control messages. In one example, the communication unit 21 executes a software program which allows operating as an OpenFlow switch (OFS) in an OpenFlow network. The communication unit 21 also processes a packet inputted from a interface unit 22 in accordance with processing rules (or flow entries) and processing commands (output instructions or the like), which are specified by the node device control section 11 in the controller 10. It should be noted that, when neither processing rule (flow entry) nor processing command (output instructions or the like) is specified for an inputted packet, the communication unit 21 judges the packet as the first packet and transmits inquiry information of this packet to the controller 10 through the control channel.
- [0071] The interface unit 22 is a network interface provided within the node device 20. The interface unit 22 may be a network interface having physical ports or a network interface having virtual ports. The interface unit 22 is used for establishing a connection to a connection destination, such as an adjacent node device and a host and

adapted to exchange packets through a data communication link. The interface unit 22 may be used for establish a connection with a controller 10. When receiving a packet from a connection destination, such as an adjacent node device and a host, the interface unit 22 outputs the packet to the communication unit 21.

[0072] (Topology Retrieving Process in First Embodiment)

An exemplary topology retrieving process in the first embodiment is described below.

[0073] Each of the controllers 10-1 and 10-2 prepares topology retrieval packets in order to retrieve the topology of its own network. In one embodiment, LLDP (Link Layer Discovery Protocol) frames may be used as topology retrieval packets. It should be noted that topology retrieval packets are not limited to LLDP frames and packets other than LLDP frames may be used as topology retrieval packets in an actual implementation.

[0074] In this embodiment, LLDP frames are used as exemplary topology retrieval packets.

[0075] (LLDP Frame Format in First Embodiment)

An exemplary format of LLDP frames in this embodiment is described with reference to Fig. 5.

[0076] In this embodiment, each LLDP frame includes an "LLDP header" region (or field), and an "optional TLVs" region (or field). Note that "TLV" stands for type, length and value.

[0077] The "LLDP header" region includes a "source MAC address" region, a "destination MAC address" region and an "ether type" region, which is denoted by the legend "eth_type".

[0078] The "source MAC address" region contains the source MAC address of the LLDP frame.

[0079] The "destination MAC address" contains the destination source MAC address of the LLDP frame.

[0080] The "ether type" region contains information that indicates the ether type of the LLDP frame. Usually, "0x88cc" is specified as the ether type of the LLDP frame.

[0081] The "optional TLVs" region includes an "identification information" region (or a "controller ID" region).

[0082] The "identification information" region contains the identification information (or the controller ID) of the controller that prepares the LLDP header.

[0083] Various types of identification information (or the controller ID) may be used if they are unique in the networks. For example, a MAC address of the interface of the controller 10, an IP address, a VLAN tag information (VLAN ID) and the like may be used as the identification information. It should be noted that identification information actually-used is not limited to these example.

[0084] (Procedure of Topology Retrieval in First Embodiment)

Next, an exemplary procedure of the topology retrieval in the first embodiment is described below.

[0085] (1) Operation for Transmitting LLDP Frame

First, a description is given of an exemplary operation in which LLDP frames are transmitted from the controllers 10 to the node devices 20.

[0086] The topology management section 111 in each of the controllers 10-1 and 10-2 prepares LLDP frames as topology retrieval packets.

[0087] At this time, the topology management unit 111 in each of the controllers 10-1 and 10-2 incorporates its own identification information (or controller ID) into each LLDP frame to identify the controller; the identification information (or the controller ID) in each LLDP frame is defined to be unique among the networks.

[0088] In this embodiment, the topology management section 111 in each of the controllers 10-1 and 10-2 incorporates identification information (or the controller ID) of the controller 10-1 or 10-2 into the "identification information" region of the "optional TLVs" region within each LLDP frame as shown in Fig. 5.

[0089] The topology management section 111 in each of the controllers 10-1 and 10-2 requests the control message processing section 112 to instruct to transmit the LLDP frames to the node devices 20. In an actual implementation, the topology management section 111 may only output the LLDP frames to the control message processing section 112.

[0090] The control message processing section 112 in each of the controllers 10-1 and 10-2 prepares packet-out messages each incorporating an LLDP frame and a transmission instruction thereof in response to the request of the transmission instruction of the LLDP frames to the node devices 20, and requests the node communication section 113 to transmit the packet-out messages to all of the node devices 20 provided in its own network. The packet-out message is a sort of control messages. The packet-out messages include transmission instructions which instruct to transmit the LLDP frame from all the interface units 22 in each node devices 20.

[0091] The node communication section 113 in each of the controllers 10-1 and 10-2 transmits the packet-out messages from the control message processing section 112 to all of the node devices 20 provided in its own network through the control channels.

[0092] The communication unit 21 in each node device 20 transmits the LLDP frame to the connection destinations from all the interface units 22 in its own node device 20, in response to the above-described packet-out message. That is, the communication unit 21 in each node device 20 transmits the LLDP frame in the form of broadcasting. Also, the communication unit 21 in each node device 20 receives the LLDP frames from the connection destinations via the interface units 22.

[0093] (2) Operation for Transmitting Inquiry Information of LLDP Frame via Control Channel

In the following, a description is given of an exemplary operation of transmitting inquiry information of an LLDP frame from each node device 20 to the controller 10 via the control channel.

[0094] When receiving an LLDP frame from a connection destination via an interface unit 22, the communication unit 21 in each node device 20 transmits a packet-in message as inquiry information to the controller 10 that controls the node device 20, through the control channel. The packet-in message is a sort of control messages.

[0095] The packet-in message includes an "LLDP frame" region, a "node device ID" region and an "interface ID" region.

[0096] The "LLDP frame" region contains the received LLDP frame.

[0097] The "node device ID" region contains the identification information (or the node device ID) of the node device 20.

[0098] The "interface ID" region contains identification information (or the interface ID) of the interface unit 22 that receives the LLDP frame.

[0099] Usually, the communication unit 21 in each node device 20 receives LLDP frames containing identification information (the controller ID) of the controller 10 of the same network from the node device 20 of the same network, and transmits packet-in messages respectively incorporating the LLDP frames as the inquiry information to the controller 10.

[0100] When the source is the node device 20-3 and the destination is the node device 20-4, however, an LLDP frame containing identification information (or the controller ID) of the controller 10-1 is transmitted to the controller 10-2. That is, when receiving the LLDP frame having the identification information (controller ID) of the controller 10-1 from the node device 20-3, the communication unit 21 in the node device 20-4 transmits a packet-in message containing the LLDP frame as the inquiry information to the controller 10-2.

[0101] When the source is the node device 20-4 and the destination is the node device 20-3, on the other hand, the LLDP frame which has the identification information (or the controller ID) of the controller 10-2 is transmitted to the controller 10-1. That is, when receiving the LLDP frame having the identification information (or the controller ID) of the controller 10-2 from the node device 20-4, the communication unit 21 in the node device 20-3 transmits a packet-in message incorporating the LLDP frame as the inquiry information to the controller 10-1.

[0102] The node communication section 113 in each of the controllers 10-1 and 10-2 receives the above packet-in messages from the respective node devices 20 connected thereto via the control channels and outputs the packet-in messages to the control

message processing section 112.

- [0103] The control message processing section 112 in each of the controllers 10-1 and 10-2 analyzes the above packet-in messages and extracts the LLDP frames incorporated in the packet-in messages. The control message processing section 112 outputs the extracted LLDP frames to the adjacency discovery section 114.
- [0104] The adjacency discovery section 114 in each of the controllers 10-1 and 10-2 determines whether the identification information (or the controller ID) described in each LLDP frame is identical to the identification information (controller ID) of its own controller 10.
- [0105] In this embodiment, the adjacency discovery section 114 in each of the controllers 10-1 and 10-2 refers to the "identification information" region of the "optional TLVs" region of each LLDP frame (see Fig. 5) and determines whether the identification information (or the controller ID) incorporated in the "identification information" region is identical to the identification information (or the controller ID) of its own controller 10.
- [0106] If the identification information (controller ID) incorporated in an LLDP frame is identical to the identification information (or the controller ID) of its own controller 10, the adjacency discovery section 114 in each of the controllers 10-1 and 10-2 can dynamically detect and monitor the node device 20 adjacent to another node device only in its own network and the interface units 22 of the node device 20.
- [0107] If the identification information (controller ID) in an LLDP frame is different from the identification information (or the controller ID) of its own controller 10, the adjacency discovery section 114 in each of the controllers 10-1 and 10-2 can detect the existence of a controller 10 in a different network. Moreover, the adjacency discovery section 114 can dynamically detect and monitor a node device 20 at the boundary with the different network and the interface units 22 of the node device 20.
- [0108] (3) Operation for transmitting LLDP Frame via Data Communication Link
An exemplary operation for transmitting an LLDP frame from each node device 20 to the controller 10 via a data communication link is described below.
- [0109] A communication unit 21 in a node device 20 connected to the controller 10 via a data communication link, for example, the node device 20-1, transmits the LLDP frame also from the interface unit 22 connected to the controller 10. For example, the communication unit 21 in the node device 20-1 transmits the LLDP frame to the controller 10-1 from the interface unit 22 connected to the controller 10-1 via the data communication link. Similarly, the node device 20-6 transmits the LLDP frame to the controller 10-2 from the interface unit 22 connected to the controller 10-2 via the data communication link.
- [0110] The interface management section 15 in each of the controllers 10-1 and 10-2

receives the LLDP frame from each node device 20 of the connection destination, via the data communication link, and outputs the LLDP frame to the adjacency discovery section 114.

[0111] The adjacency discovery section 114 in each of the controllers 10-1 and 10-2 determines whether the identification information (or the controller ID) described in the LLDP frame is identical to the identification information (controller ID) of its own controller 10.

[0112] If the identification information (controller ID) incorporated in the LLDP frame is identical to the identification information (or the controller ID) of the controller 10, the adjacency discovery section 114 in each of the controllers 10-1 and 10-2 can dynamically detect and monitor the node device 20 connected to the controller 10 and the interface units thereof.

[0113] (4) Operation for Preparing Topology Information

An exemplary operation for preparing topology information in the controller 10 is described below.

[0114] The adjacency discovery section 114 in each of the controllers 10-1 and 10-2 sends information detected or obtained by the above-described operations, to the topology management section 111.

[0115] The topology management section 111 in each of the controllers 10-1 and 10-2 prepares and stores topology information and boundary information of the corresponding network on the basis of the information received from the adjacency discovery section 114.

[0116] (Note)

The detecting method of the node devices 20 connected to the controller 10 and the interface units 22 thereof is not limited to this method in an actual implementation. For example, information related to each node device 20 connected to the controller 10 and the interface units 22 thereof may be notified and set in advance by a manual operation by an operator or the like.

[0117] (Operation for Mutual Notification of Judging Conditions of Packets between Controllers in First Embodiment)

An exemplary operation for mutual notification of identifying conditions (or identifying rules) of packets between the controllers 10 in the first embodiment is described below.

[0118] The identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 calculates identifying conditions (or identifying rules) of packets used to communicate with the controller 10 in a different network on the basis of the topology information and the boundary information of its own network, which are stored in the topology management section 111, and prepares a notification packet that stores the

identifying conditions (or the identifying rules). Moreover, the identifying condition calculation section 115 requests and instructs the control message processing section 112 to transmit the notification packet to a node device 20 at the boundary with a different network. The identifying conditions (or the identifying rules) are attached to the packet header, which may include the MAC address, the IP address or the protocol type. It should be noted that in an actual implementation, the identifying condition calculation section 115 may only output the notification packet to the control message processing section 112.

- [0119] The control message processing section 112 in each of the controllers 10-1 and 10-2 prepares a packet-out message incorporating the notification packet and an instruction to output the notification packet to a node device 20 in a different network in response to a request for instructing transmission of the notification packet to a node device 20 at the boundary with the different network. Moreover, the control message processing section 112 requests the node communication section 113 to transmit the packet-out message to the node device 20 at the boundary with the different network.
- [0120] The node communication section 113 in each of the controllers 10-1 and 10-2 transmits the above packet-out message to the node device 20 at the boundary with the different network via the control channel, in response to the transmission request of the packet-out message from the control message processing section 112.
- [0121] When receiving the above-described packet-out message via the control channel, the communication unit 21 in the node device 20 at the boundary with the different network transmits the notification packet to the node device 20 in the different network through the data communication link from the interface unit 22 connected to the node device 20 in the different network, on the basis of the instruction of outputting the notification packet to the node device 20 in the different network.
- [0122] When a notification packet is inputted from the interface unit 22 connected to the node device 20 in the different network, the communication unit 21 in the node device 20 at the boundary with the different network transmits a packet-in message incorporating the notification packet as inquiry information to the controller 10 that controls its own node device 20.
- [0123] The node communication section 113 in each of the controllers 10-1 and 10-2 receives the above-described packet-in message from each node device 20 connected thereto via the control channel and outputs the packet-in message to the control message processing section 112.
- [0124] The control message processing section 112 in each of the controllers 10-1 and 10-2 analyzes the above-described packet-in message and extracts the notification packet incorporated in the packet-in message. The control message processing section 112 then outputs the notification packet to the adjacency discovery section 114.

- [0125] The identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 recognizes the identifying conditions (or the identifying rules) of packets used to communicate with the controller 10 in the different network, on the basis of the notification packet.
- [0126] As a result, the controllers 10-1 and 10-2 can recognize the identifying conditions (or the identifying rules) of packets used in the mutual communications.
- [0127] (1) Notification to Controller 10-2 from Controller 10-1
An exemplary operation of notifying identifying conditions (or identifying rules) of packets from the controller 10-1 to the controller 10-2 is described.
- [0128] The controller 10-1 prepares a notification packet incorporating identifying conditions (or identifying rules) of packets used to communicate with the controller 10-2 and transmits a packet-out message incorporating the notification packet and an instruction of outputting the notification packet to the node device 20-4 to the node device 20-3 at the boundary with the different network via the control channel.
- [0129] When receiving the packet-out message from the controller 10-1, the node device 20-3 transmits the notification packet from the interface unit 22 connected to the node device 20-4 in accordance with the instruction of outputting the notification packet to the node device 20-4.
- [0130] When receiving the notification packet from the node device 20-3, the node device 20-4 transmits a packet-in message incorporating the notification packet as inquiry information to the controller 10-2, which controls the node device 20-4, via the control channel.
- [0131] The controller 10-2 receives the packet-in message from the node device 20-3 and obtains the notification packet incorporated in the packet-in message. This allows the controller 10-2 to recognize the identifying conditions (or the identifying rules) of packets used to communicate with the controller 10-1.
- [0132] (2) Notification to Controller 10-1 from Controller 10-2
An exemplary operation of notification of identifying conditions (or identifying rules) of packets from the controller 10-2 to the controller 10-1 is described.
- [0133] The controller 10-2 prepares a notification packet incorporating identifying conditions (or identifying rules) of packets used to communicate with the controller 10-1, and transmits a packet-out message incorporating the notification packet and an instruction of outputting the notification packet to the node device 20-3, to the node device 20-4 at the boundary with the different network via the control channel.
- [0134] When receiving the packet-out message from the controller 10-2, the node device 20-4 transmits the notification packet from the interface 22 connected to the node device 20-3 in accordance with the instruction of outputting the notification packet to the node device 20-3.

- [0135] When receiving the notification packet from the node device 20-4, the node device 20-3 transmits a packet-in message incorporating the notification packet as inquiry information to the controller 10-1, which controls the node device 20-3 via the control channel.
- [0136] The controller 10-1 receives the packet-in message from the node device 20-4 and obtains the notification packet incorporated in the packet-in message. This allows the controller 10-1 to recognize the identifying conditions (or the identifying rule) of packets used to communicate with the controller 10-2.
- [0137] (Operation for Setting Processing Rules in First Embodiment)
An exemplary operation for setting processing rules (or flow entries) in the first embodiment is described below.
- [0138] The identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 calculates identifying conditions (or identifying rules) of packets used to communicate with the controller 10 in the different network, on the basis of the topology information and the boundary information of its own network which are stored in the topology management section 111. Also, the identifying condition calculation section 115 recognizes identifying conditions (or an identifying rule) of packets notified by the controller 10 in the different network.
- [0139] The route calculation section 116 in each of the controllers 10-1 and 10-2 determines the end points of a route to be used in the communication, on the basis of the topology information and the boundary information of the network stored in the topology management section 111.
- [0140] At this time, one of the end points of the route is defined as the node device at the boundary of the different network. In the network 1, the node device 20-3 is defined as one of the end points. In the network 2, the node device 20-4 is defined as one of the end points.
- [0141] Also, the other end point of the route is defined as the node device connected to the controller via a data communication link. In the network 1, the node device 20-1 is defined as the other end point. In the network 2, the node device 20-6 is defined as the other end point.
- [0142] The route calculation section 116 in each of the controllers 10-1 and 10-2 calculates a route which connects the end points thus-determined. Specifically, the route calculation section 116 determines a bi-directional transfer route which connects the interfaces of the determined node devices 20.
- [0143] In this embodiment, the route calculation section 116 in the controller 10-1 calculates the route which connects the interface unit used for connection to controller 10-1 in the node device 20-1 in the network 1 to the interface unit used for connection to the node device 20-4 in the network 2 in the node device 20-3.

- [0144] Similarly, the route calculation section 116 in the controller 10-2 calculates the route which connects the interface unit used for connection to controller 10-2 in the node device 20-6 in the network 2 and the interface unit used for connection to the node device 20-3 in the network 1 in the node device 20-4.
- [0145] The processing rule calculation section 117 in each of the controllers 10-1 and 10-2 obtains the route to be used for transfer from the route calculation section 116 and obtains the identifying conditions (or the identifying rules) of packets transferred on the route from the identifying condition calculation section 115.
- [0146] The processing rule calculation section 117 in each of the controllers 10-1 and 10-2 uses the obtained information to calculate processing required to transfer the packets in each node device 20 on the route, and calculates the processing rules (or the flow entries) in which the packet identifying conditions (or the packet identifying rules) are defined as conditions under which the processing is to be performed. At this time, the processing rule calculation section 117 incorporates the identification information (processing rule ID) into each calculated processing rule (or flow entry). Specifically, the processing rule ID is incorporated in a cookie region (a region of 64 bits in which any information can be stored) in each processing rule (or flow entry).
- [0147] The processing rule management section 118 in each of the controllers 10-1 and 10-2 registers information related to the processing rules (or the flow entries) calculated by the processing rule calculation section 117 into the processing rule storage section 119, and requests the control message processing section 112 to set processing rules (or the flow entries) for each node devices 20 on the route.
- [0148] When receiving the setting request of processing rules (or flow entries) for the node devices 20, the control message processing section 112 in each of the controllers 10-1 and 10-2 prepares flow modification messages (or FlowMod-Add) in order to register the processing rules (or the flow entries) in the node devices 20. After that, the control message processing section 112 requests the node communication section 113 to transmit the flow modification messages (or FlowMod-Add) to the node devices 20 on the route. The flow modification messages (or FlowMod-Add) are a sort of control messages.
- [0149] When receiving the transmission request of the flow modification messages (FlowMod-Add) to the node devices 20, the node communication section 113 in each of the controllers 10-1 and 10-2 transmits the flow modification messages (FlowMod-Add) to the node devices 20 on the route via the control channels.
- [0150] When receiving a flow modification message (FlowMod-Add) from the controller 10 via the control channel, the communication unit 21 in each node device 20 registers the processing rules (or the flow entries) on the basis of the contents of the flow modification message (or FlowMod-Add).

- [0151] When then receiving a packet through a interface unit 22, the communication unit 21 in each node device 20 processes the received packet in accordance with the contents of the registered processing rules (or the flow entries).
- [0152] (Summary of First Embodiment)
- In this way, the above-described operation for setting the processing rules (or the flow entries) is carried out independently in the networks 1 and 2. As result, a bidirectional transfer route is established between the controllers 10.
- [0153] Each of the controllers 10-1 and 10-2 determines the identifying conditions (or the identifying rules) of packets used in the mutual communications.
- [0154] Each of the controllers 10-1 and 10-2 calculates the bidirectional route within its own network and specifies the node devices on the route and the interface units thereof.
- [0155] Each of the controllers 10-1 and 10-2 prepares processing rules (or flow entries) that defines a set of identifying conditions (or identifying rules) and contents of processing (or actions) which involves outputting packets to the interface unit connected to the adjacent node device, for the node devices on the route.
- [0156] In this embodiment, each of the controllers 10-1 and 10-2 sets the node devices 20 under the control of the each controller 10 with processing rules (or the flow entries) which allows transferring packets from the node device at the boundary with the adjacent network to the node device connected to the controller 10 via the data communication link. For example, the controller 10-1 sets each of the node devices 20-1 to 20-3 with processing rules (or flow entries) which allows transfer of packets from the node device 20-3 to the node device 20-1. The controller 10-2 sets each of the node devices 20-4 to 20-6 with processing rules (or flow entries) which allows transfer of packets from the node device 20-4 to the node device 20-6.
- [0157] Also, each of the controllers 10-1 and 10-2 sets the node devices 20 under its control with processing rules (or flow entries) which allows transfers of packets from the node device connected via the data communication link to the interface unit of the node device at the boundary with the adjacent network. For example, the controller 10-1 sets each of the node devices 20-1 to 20-3 with processing rules (or flow entries) which allows transfer of packets from the node device 20-1 to the node device 20-3. The controller 10-2 sets each of the node devices 20-4 to 20-6 with processing rules (or flow entries) which allows transfer of packets from the node device 20-6 to the node device 20-4.
- [0158] Since the above-described operations are carried out in each of the adjacent networks, the transfer route is established in each of the networks, this enables communications between the controllers without direct transactions between adjacent networks.
- [0159] Note that the link connection implies connection through a data communication link.

[0160] (Format of Processing Rule)

An exemplary format of the processing rules (or the flow entries) is described with reference to Fig. 6A and Fig. 6B.

[0161] A processing rule (or a flow entry) includes an identifying condition region (or an identifying rule region) and a contents-of-processing region (or an action region).

[0162] The identifying condition region (or the identifying rule region) includes the "source MAC address" region, a "destination MAC address" region and an "ether type" region (or an "eth_type" region).

[0163] The "source MAC address" region contains the MAC address of the network interface of the source of the packet.

[0164] The "destination MAC address" region contains the MAC address of the network interface of the destination of the packet.

[0165] The "ether type" region contains information indicating the ether type of the packet.

[0166] The "contents-of-processing" region include an "action" region.

[0167] The "action" region contains an instruction of outputting the packet to a predetermined output destination.

[0168] In this embodiment, each controller is allowed to flexibly use various types of packets for each counterpart controller.

[0169] (Processing Rule in Network 1)

Fig. 6A shows an example of processing rules (or flow entries) respectively set to the node devices 20-1 to 20-3 from the controller 10-1 in the network 1.

[0170] (1) Processing Rule (Flow Entry) from Node Device 20-1 to Controller 10-1

Source MAC address:

MAC address of controller 10-2

Destination MAC address:

MAC address of controller 10-1

Ether Type (eth_type): 0x8800

Action: output to the controller 10-1

[0171] (2) Processing Rule (Flow Entry) from Node Device 20-1 to Node Device 20-2

Source MAC Address:

MAC address of controller 10-1

Destination MAC address:

MAC address of controller 10-2

Ether Type (eth_type): 0x8800

Action: output to node device 20-2

[0172] (3) Processing Rule (Flow Entry) from Node Device 20-2 to Node Device 20-1

Source MAC address:

MAC address of controller 10-2

Destination MAC address:

MAC address of controller 10-1

Ether Type (eth_type): 0x8800

Action: output to node device 20-1

[0173] (4) Processing Rule (Flow Entry) from Node Device 20-2 to Node Device 20-3

Source MAC address:

MAC address of controller 10-1

Destination MAC address:

MAC address of controller 10-2

Ether type (eth_type): 0x8800

Action: output to node device 20-3

[0174] (5) Processing Rule (Flow Entry) from Node Device 20-3 to Node Device 20-2

Source MAC address:

MAC address of controller 10-2

Destination MAC address: MAC Address of controller 10-1

Ether type (eth_type): 0x8800

Action: output to node device 20-2

[0175] (6) Processing Rule (Flow Entry) from Node Device 20-3 to Node Device 20-4

Source MAC address]:

MAC address of controller 10-1

Destination MAC address:

MAC Address of controller 10-2

Ether type (eth_type): 0x8800

Action: output to node device 20-4

[0176] The above-described processing rules (1) and (2) are processing rules (or flow entries) used in the node device 20-1.

[0177] The above-described processing rules (3) and (4) are processing rules (or flow entries) used in the node device 20-2.

[0178] The above-described processing rules (5) and (6) are processing rules (or flow entries) used in the node device 20-3.

[0179] (Processing Rule in Network 2)

Fig. 6B shows an example of the processing rule (or the flow entry) set to each of the node devices 20-4 to 20-6 from the controller 10-2 in the network 2.

[0180] (1) Processing Rule (Flow Entry) from Node Device 20-4 to Node Device 20-3

Source MAC address:

MAC address of controller 10-2

Destination MAC Address:

MAC address of controller 10-1

Ether Type (eth_type): 0x8800

Action: output to node device 20-3

[0181] (2) Processing Rule (Flow Entry) from Node Device 20-4 to Node Device 20-5

Source MAC address:

MAC Address of controller 10-1

Destination MAC address:

MAC Address of controller 10-2

Ether type (eth_type): 0x8800

Action: output to node device 20-5

[0182] (3) Processing Rule (Flow Entry) from node device 20-5 to node device 20-4

Source MAC address:

MAC address of controller 10-2

Destination MAC address:

MAC Address of controller 10-1

Ether Type (eth_type): 0x8800

Action: output to node device 20-4

[0183] (4) Processing Rule (Flow Entry) from node device 20-5 to node device 20-6

Source MAC address]:

MAC address of controller 10-1

Destination MAC Address:

MAC address of controller 10-2

Ether Type (eth_type): 0x8800

Action: output to node device 20-6

[0184] (5) Processing Rule (Flow Entry) from node device 20-6 to node device 20-5

Source MAC address:

MAC address of controller 10-2

Destination MAC address:

MAC Address of controller 10-1

Ether type (eth_type): 0x8800

Action: output to node device 20-5

[0185] (6) Processing Rule (Flow Entry) from node device 20-6 to controller 10-2

Source MAC address]:

MAC Address of controller 10-1

Destination MAC address:

MAC Address of controller 10-2

Ether type (eth_type): 0x8800

Action: output to controller 10-2

[0186] The above-described processing rules (1) and (2) are processing rules (or flow

entries) used in the node device 20-4.

[0187] The above-described processing rules (3) and (4) are processing rules (or flow entries) used in the node device 20-5.

[0188] The above-described processing rules (5) and (6) are processing rules (or flow entries) used in the node device 20-6.

[0189] (Supplement)

The MAC address of the controller 10-1 means the MAC address of the network interface in the controller 10-1. Also, the MAC address of the controller 10-2 means the MAC address of the network interface in the controller 10-2.

[0190] It should be noted that the MAC address is merely one exemplary identifier used to identify the source and the destination. The IP address or the like may be used in place of the MAC address.

[0191] <Second Embodiment>

A description is given of a second embodiment of the present invention in the following.

[0192] In this embodiment, identifying conditions (or identifying rules) are also determined in the topology retrieval by using LLDP frames. Specifically, identifying conditions (or identifying rules) of packets are embedded in LLDP frames in addition to identification information (or the controller ID) of the controller when LLDP frames are prepared in the controller.

[0193] (Format of LLDP Frame in Second Embodiment)

An exemplary format of the LLDP frame in this embodiment is described with reference to Fig. 7.

[0194] In this embodiment, each LLDP frame includes an "LLDP header" region (or field) and an "optional TLVs" region.

[0195] The contents of the "LLDP header" region in the second embodiment are same as those in the first embodiment.

[0196] The "optional TLVs" region includes an "identification information" region (or a "controller ID" region, a "MAC address" region and an "ether type" region (or an "eth_type" region).

[0197] The "identification information" region contains identification information (or the controller ID) of the controller which prepares the LLDP frame.

[0198] The "MAC address" region contains a MAC address which is one of identifying conditions (or identifying rules) of packets.

[0199] The "ether type (eth_type)" region contains information that indicates an ether type which is one of identifying conditions (or identifying rules) of packets.

[0200] It should be noted that the MAC address is merely one exemplary identifier used to identify the source and the destination. The IP address or the like may be used in place

of the MAC address.

[0201] That is, in this embodiment, identifying conditions (or identifying rules) of packets are stored in the "optional TLVs" region in each LLDP frame.

[0202] (System Configuration)

This embodiment is described on the basis of the system configuration shown in Fig. 2, similarly to the first embodiment.

[0203] The configurations of the controllers 10 and the node devices 20 are basically similar to those of the first embodiment.

[0204] (Operation of Topology Retrieval in Second Embodiment)

An exemplary operation for topology retrieval in the second embodiment is described below.

[0205] First, the topology management section 111 in each of the controllers 10-1 and 10-2 prepares LLDP frames as topology retrieval packets, similarly to the first embodiment.

[0206] At this moment, the topology management section 111 in each of the controllers 10-1 and 10-2 incorporates identification information (or the controller ID) which indicates its own controller in each LLDP frame, the identification information being unique among the networks.

[0207] Here, the topology management section 111 in each of the controllers 10-1 and 10-2 incorporates the identification information (or the controller ID) of its own controller 10 into the "identification information" region in the "optional TLVs" region in each LLDP frame as shown in Fig. 7.

[0208] The topology management section 111 in each of the controllers 10-1 and 10-2 outputs the prepared LLDP frames to the identifying condition calculation section 115.

[0209] The identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 calculates identifying conditions (or identifying rules) of packets used to communicate with the controller 10 in the different network, on the basis of the topology information and the boundary information of the networks which are stored in the topology management section 111.

[0210] In this embodiment, the identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 incorporates identifying conditions (or identifying rules) of packets used to communicate with the adjacent node device 20 by each node device 20 into each LLDP frame.

[0211] In this embodiment, the identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 incorporates the MAC address of the network interface within its own controller 10 in the "MAC address" region of the "optional TLVs" region in each LLDP frame as shown in Fig. 7.

[0212] Also, the identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 incorporates information indicating the ether type of packets used to com-

municate with the controller 10 in the different network in the "ether type" region of the "optional TLVs" region in each LLDP frame as shown in Fig. 7.

[0213] At this moment, the identification information (or the controller ID) of the controller and identifying conditions (or rules) of packets are contained in each LLDP frame.

[0214] The identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 requests the control message processing section 112 to instruct transmission of the LLDP frame to each node device 20. In an actual implementation, the identifying condition calculation section 115 may only output the LLDP frames to the control message processing section 112. It should be noted that, in an actual implementation, the identifying condition calculation section 115 may return the LLDP frames in which identifying conditions (or rules) of packets are embedded to the topology management section 111. In this case, similarly to the first embodiment, the topology management section 111 requests the control message processing section 112 to instruct transmission of the LLDP frame to each node device 20.

[0215] The control message processing section 112 in each of the controllers 10-1 and 10-2 prepares the packet-out message incorporating the LLDP frame and the transmission instruction, in response to the request for instructing the transmission of the LLDP frame to each node device 20, and requests the node communication section 113 to transmit the packet-out message to all of the node devices 20 in its own network. The packet-out message includes transmission instructions which instruct to transmit the LLDP frame from all of the interface units 22 in each node device 20.

[0216] The node communication section 113 in each of the controllers 10-1 and 10-2 transmits the packet-out message from the control message processing section 112 to all of the node devices 20 in its own network.

[0217] The communication unit 21 in each node device 20 transmits the LLDP frame to the connection destinations from all of the interface units 22 in the node device 20. Also, the communication unit 21 receives the LLDP frames from the connection destinations via the interface units 22.

[0218] The communication unit 21 in each node device 20 transmits the packet-in message as inquiry information to the controller 10 that controls the node device 20 via the control channel. At this time, the communication unit 21 in each node device 20 incorporates the LLDP frame, interface information of the interface unit that receives the LLDP frame (or (the interface unit ID), and identification information of the node device (or the node device ID) in the packet-in message.

[0219] Usually, the communication unit 21 in each node device 20 receives an LLDP frame having the identification information (or the controller ID) of the controller 10 in the same network from the node device 20 in the same network, and transmits a packet-in message incorporating the LLDP frame as inquiry information to the controller 10.

- [0220] When the source is the node device 20-3 and the destination is the node device 20-4, however, an LLDP frame having identification information (the controller ID) of the controller 10-1 is transmitted to the controller 10-2. That is, when receiving an LLDP frame having identification information (or the controller ID) of the controller 10-1 from the node device 20-3, the communication unit 21 in the node device 20-4 transmits a packet-in message incorporating the LLDP frame as inquiry information to the controller 10-2.
- [0221] When the source is the node device 20-4 and the destination is the node device 20-3, on the other hand, an LLDP frame having identification information (or the controller ID) of the controller 10-2 is transmitted to the controller 10-1. That is, when receiving the LLDP frame having identification information (or the controller ID) of the controller 10-2 from the node device 20-4, the communication unit 21 in the node device 20-3 transmits a packet-in message incorporating the LLDP frame as inquiry information to the controller 10-1.
- [0222] The node communication section 113 in each of the controllers 10-1 and 10-2 receives the above-described packet-in message from each node device 20 of the connection destination through the control channel and outputs the packet-in message to the control message processing section 112.
- [0223] The control message processing section 112 in each of the controllers 10-1 and 10-2 analyzes the received packet-in messages and extracts the LLDP frames incorporated in the packet-in messages. The control message processing section 112 outputs the extracted LLDP frames to the adjacency discovery section 114 and the identifying condition calculation section 115.
- [0224] The adjacency discovery section 114 in each of the controllers 10-1 and 10-2 operates similarly to the first embodiment.
- [0225] In this embodiment, the identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 determines whether the identification information (controller ID) contained in each LLDP frame is identical to the identification information (controller ID) of its own controller 10.
- [0226] Here, the identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 refers to the "identification information" region (or the "controller ID" region) of the "optional TLVs" region of the LLDP frame (see Fig. 7), and determines whether the identification information (or the controller ID) contained in the "identification information" region (or the "controller ID" region) is identical to the identification information (or the controller ID) of its own controller 10.
- [0227] If the identification information (controller ID) contained in an LLDP frame differs from the identification information (controller ID) of its own controller 10, the identifying condition calculation section 115 in each of the controllers 10-1 and 10-2

determines that the LLDP frame results from the controller 10 in the different network. In this case, the identifying condition calculation section 115 determines that the identifying conditions (or the rules) contained in the LLDP frame are the identifying conditions (or the identifying rules) of packets used to communicate with the controller 10 in the different network.

[0228] In this embodiment, the identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 refers to the "MAC address" region of the "optional TLVs" region in the LLDP frame (see Fig. 7) and obtains the MAC address of the network interface in the controller 10 in the different network.

[0229] Also, the identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 refers to the "ether type" region (or the "eth_type" region) of the "optional TLVs" region in the LLDP frame shown in Fig. 7, and obtains information that indicates the ether type of packets used to communicate with the controller 10 in the different network.

[0230] In this way, an LLDP frame prepared by each of the controllers 10-1 and 10-2 is transmitted from the interface units of each node device and this allows discovering of the network adjacency and recognizing the identifying conditions (or the rules) in the topology retrieval at the same time.

[0231] <Third Embodiment>

A third embodiment of the present invention is described below.

[0232] The system configuration of this embodiment is identical to that shown in Fig. 1.

[0233] The system configuration of this embodiment differs from the system configuration of the first embodiment shown in Fig. 2 in that the controllers 10 and the node devices 20 are logically connected via control channels; the interfaces are not physically connected to each other via data communication links.

[0234] Each controller 10 communicates with the node device 20 at the boundary with a different network. That is, the controller 10-1 communicates with the node device 20-3 via a control channel. The controller 10-2 communicates with the node device 20-4 via a control channel.

[0235] Each node device 20 at the boundary with a different network communicates with the node device 20 in the different network through a data communication link. In this embodiment, the node device 20-3 communicates with the node device 20-4 through a data communication link. The node device 20-4 communicates with the node device 20-3 through the data communication link.

[0236] In this embodiment, each controller 10 sets the node device 20 located on the boundary with a different network with processing rules (or flow entries) which instruct to, when receiving a packet addressed to the controller 10 in its own network from a different network, transfer the packet via the control channel to the controller

10 in its own network. That is, the controller 10-1 sets the node device 20-3 with processing rules (or flow entries) which instruct the node device 20-3 to transfer a packet addressed to the controller 10-1 when receiving the packet from the controller 10-2. The controller 10-2 sets the node device 20-4 with processing rules (or flow entries) which instruct the node device 20-3 to transfer a packet addressed to the controller 10-2 when receiving the packet from the controller 10-1.

[0237] This allows communications between the controllers 10-1 and 10-2.

[0238] (Configuration of Controller in Third Embodiment)

An exemplary configuration of the controller 10 in this embodiment is described with reference to Fig. 8. Note that this exemplary configuration is common to the controllers 10-1 and 10-2.

[0239] The controller 10 includes a node device control section 11, a sorting section 12, a sorting rule storage section 13, one or more virtual ports 14 (one shown) and an interface management section 15.

[0240] The node device control section 11 and the interface management section 15 basically have same functions as those in the first embodiment,

[0241] The node device control section 11 controls the node devices 20 connected thereto via the control channels. For example, the node device control section 11 executes a software program which allows operating as an OpenFlow controller (OFC) in an OpenFlow network. Here, the node device control section 11 monitors and manages each interface unit contained in each node device 20 via the control channel, and sets each node device 20 with processing rules (or flow entries) for packets transmitted and received in its interface units. One example of the contents of a processing rule (a flow entry) of a packet may include an instruction to output to a interface unit or a node device control section 11 specified on the basis of the feature of the packet received by the interface unit, an instruction to, when a packet prepared by the node device control section 11 is received, output the packet to a specified interface unit, and the like.

[0242] In this embodiment, when receiving a packet from a node device 20, the node device control section 11 outputs the packet to the sorting section 12 after attaching information that identifies the node device 20 and the interface unit 22 of the source of the packet. Also, when receiving a packet from the sorting section 11, the node device control section 11 analyzes information attached to the packet that indicates the node device 20 and the interface unit 22 of the source of the packet and performs control to output the packet from a specified interface unit 22 in a specified node device 20 by issuing an output instruction specifying a proper control channel.

[0243] The sorting section 12, upon receiving a packet from a virtual port 14, refers to sorting rules stored in the sorting rule storage section 13 to specify a node device 20 and an interface unit 22 which are suitable as the output destination, on the basis of

sorting conditions, which may include the feature of the packet, and identification information of the virtual port of the source of the packet, and outputs a message incorporating the packet and an output instruction to the specified node device 20 and interface unit 22 to the node device control section 11. Also, when receiving a packet from the node device control section 11, the sorting section 12 refers to the sorting rules stored in the sorting rule storage section 13 to specify a virtual port suitable for the output destination, on the basis of sorting conditions, which may include the feature of the packet, identification information of the node device of the source, and identification information of the interface section and the like, and outputs the packet to the specified virtual port. In an actual implementation, the sorting section 12 may be attained by installing an existing virtual machine monitor (VMM), a hypervisor or the like with the above-described functions.

[0244] The sorting rule storage section 13 stores the sorting rules of packets exchanged between the virtual ports in the controller 10 and the interface units in the respective node devices 20. Here, the sorting rule storage section 13 stores information required to sort packets as the sorting rules. The sorting rule storage section 13 returns and provides suitable sorting rules in response to a reference request from the node device control section 11 and the sorting section 12. For example, the sorting rule storage section 13 stores information which correlates virtual ports with interface units in a one-to-one relationship; and information which correlates features of packets (source and destination addresses, the kind of packets and the like) with virtual ports and interface units in a one-to-one relationship. The sorting rule storage section 13 may be attained by using a RDB (relational data base). It should be noted that the relations described in the sorting rules may be arbitrarily modified in accordance with the OS (Operation System), software and the like used in the computer which operates as the controller 10, or may be manually modified in accordance with a user operation. For example, the relations described in the sorting rules may be dynamically modified as a part of the QoS control. It should be noted, however, these are merely examples and an actual implementation is not limited to these examples.

[0245] The virtual ports 14 are each a virtual network interface provided in the controller 10. The virtual ports 14 are recognized as an entity equivalent to a physical network interface or treated in the same way as a physical network interface, by the OS (operating system) of the computer operating as the controller 10 and the like. This implies that the virtual ports 14 are each capable of transmitting and receiving packets. For example, the virtual ports 14 may be implemented as a virtual device, such as a TUN/TAP installed in the OS (Operating System) or other software. Also, individual virtual machines (VM) operating on the controller 10 may each contain an OS (Operating System) and a virtual port 14. It should be noted, however, an actual imple-

mentation is not limited to these examples

[0246] The interface management section 15 manages the network interface in the controller 10. Here, the interface management section 15 is connected to the virtual ports 14 and capable of communicating with the virtual ports 14.

[0247] In this embodiment, the node device control section 11 is connected to the sorting section 12, the sorting rule storage section 13 and the interface management section 15.

[0248] It should be noted the interface management section 15 may be connected to any of the node devices 20 via a data communication link, similarly to the first embodiment.

[0249] (Configuration of Node Device Control Section)

An exemplary configuration of the node device control section 11 of this embodiment will be described below.

[0250] In this embodiment, the configuration of the node device control section 11 is basically identical to that of the first embodiment.

[0251] The node device control section 11 includes a topology management section 111, a control message processing section 112, a node communication section 113, an adjacency discovery section 114, an identifying condition calculation section 115, a route calculation section 116, a processing rule calculation section 117, a processing rule management section 118 and a processing rule storage section 119, similarly to that shown in Fig. 3.

[0252] The topology management section 111, the control message processing section 112, the node communication section 113, the adjacency discovery section 114, the identifying condition calculation section 115, the route calculation section 116, the processing rule calculation section 117, the processing rule management section 118 and the processing rule storage section 119 basically operate similarly to those of the first embodiment.

[0253] It should be noted, however that, in this embodiment, the control message processing section 112 is further connected to the sorting section 12 and the sorting rule storage section 13.

[0254] (Operation of Topology Retrieval in Third Embodiment)

In one operation example, the topology retrieval in this embodiment is basically achieved in the procedure similar to that of the first embodiment or the second embodiment.

[0255] For example, the operation of the topology retrieval of this embodiment may involve the operations described in the sections entitled "(1) Operation for Transmitting LLDP Frame", "(2) Operation for Transmitting Inquiry Information of LLDP Frame via Control Channel", and "(4) Operation for Preparing Topology Information"; these sections describe the exemplary operation of the topology retrieval in the first embodiment.

- [0256] This allows each controller 10 to dynamically recognize and monitor existences of a controller 10 in a different network, a node device 20 located on the boundary with the different network, and its interface unit 22.
- [0257] (Operation for Setting Processing Rule in Third Embodiment)
An exemplary operation of setting processing rules (or flow entries) in the third embodiment is described below.
- [0258] The identifying condition calculation section 115 in each of the controllers 10-1 and 10-2 calculates identifying conditions (or rules) of packets used to communicate with the controller 10 in a different network, on the basis of the topology information and the boundary information of the network, which are stored in the topology management section 111.
- [0259] The route calculation section 116 in each of the controllers 10-1 and 10-2 determines the end points of a route used in the communication, on the basis of the topology information and the boundary information of the network, which are stored in the topology management section 111.
- [0260] Here, each end point of the route is defined as a node device located on the boundary with a different network. In the network 1, the node device 20-3 is defined as one end point. In the network 2, the node device 20-4 is defined as the other end point.
- [0261] The route calculation section 116 in each of the controllers 10-1 and 10-2 calculates the route which connects the controller 10 to the node device 20 located on the boundary with a different network and through the control channel.
- [0262] The processing rule calculation section 117 in each of the controllers 10-1 and 10-2 obtains the route used for packet transfer from the route calculation section 116 and obtains the identifying conditions (or the identifying rules) of packets transferred on the route from the identifying condition calculation section 115.
- [0263] The processing rule calculation section 117 in each of the controllers 10-1 and 10-2 uses the obtained information and calculates processing rules (or flow entries) to be set to the node device 20 located on the boundary with the different network. At this time, the processing rule calculation section 117 incorporates identification information (or the processing rule ID) of each processing rule (flow entry), into each processing rule (or flow entry). Specifically, the processing rule ID is stored in a cookie region in each processing rule (or flow entry). Also, contents of processing (or the actions) of the processing rules (or the flow entries) are defined to instruct an operation in which, when receiving a packet addressed to the controller 10 in its own network from the controller 10 in a different network, a packet-in message incorporating the packet and the processing rule ID is transmitted to the controller 10 in its own network through the control channel.
- [0264] For example, the processing rule calculation section 117 in the controller 10-1

calculates processing rules (or flow entries) which instruct to, when receiving a packet addressed to the controller 10-1 from the controller 10-2, transmit a packet-in message incorporating the packet and a processing rule ID "A" to the controller 10-1 via the control channel, as the processing rules (or the flow entries) to be set to the node device 20-3.

- [0265] Similarly, the processing rule calculation section 117 in the controller 10-2 calculates processing rules (or flow entries) which instruct to, when receiving a packet addressed to the controller 10-2 from the controller 10-1, transmit a packet-in message incorporating the packet and a processing rule ID "B" to the controller 10-2 via the control channel, as the processing rules (or the flow entries) to be set to the node device 20-4.
- [0266] The processing rule management section 118 in each of the controllers 10-1 and 10-2 registers information related to the processing rules (or the flow entries) calculated by the processing rule calculation section 117 into the processing rule storage section 119, and requests the control message processing section 112 to set the node device 20 located on the boundary with the different network with the processing rules (or the flow entries).
- [0267] The control message processing section 112 in each of the controllers 10-1 and 10-2, upon receiving the request for setting the node device 20 with the processing rules (or the flow entries), prepares a flow modification message (FlowMod-Add) to register the processing rules (or the flow entries) in the node device 20. The control message processing section 112 then requests the node communication section 113 to transmit the flow modification message (FlowMod-Add) to the node device 20 located on the boundary with the different network.
- [0268] The node communication section 113 in each of the controllers 10-1 and 10-2, upon receiving the transmission request of the flow modification message (FlowMod-Add) to the node device 20 located on the boundary with the different network, transmits the flow modification message (FlowMod-Add) to the node device 20 via the control channel.
- [0269] The communication unit 21 in each node device 20, upon receiving the flow modification message (FlowMod-Add) from the controller 10 via the control channel, registers therein the processing rules (or the flow entries) on the basis of the contents of the flow modification message (FlowMod-Add).
- [0270] Hereafter, the communication unit 21, upon receiving a packet via the interface unit 22, processes the packet in accordance with the contents of the registered processing rules (or the flow entries).
- [0271] (Operation for Notifying Processing Rules between Controllers in Third Embodiment)

An exemplary operation for notifying processing rules (flow entries) between the

controllers 10 in the third embodiment is described below.

- [0272] The control message processing section 112 in each of the controllers 10-1 and 10-2 obtains the processing rules (the flow entries) with regard to the communications between the controllers 10 from the processing rule management section 118, and obtains the processing rules ID embedded in the processing rules (the flow entries). The control message processing section 112 then prepares a notification packet incorporating the processing rules ID. For example, the control message processing section 112 in each of the controllers 10-1 and 10-2 incorporates the processing rules ID in the cookie region or data region of the notification packet. When preparing the notification packet, the control message processing section 112 defines the destination address as the address of the controller 10 in the different network, and defines the source address as the address of the controller 10 in its own network. It should be noted that, the control message processing section 112, may carry out this process at the same time, when preparing the flow modification message (FlowMod-Add).
- [0273] The control message processing section 112 in each of the controllers 10-1 and 10-2 prepares a packet-out message which incorporates the above-described notification packet and a transmission instruction thereof.
- [0274] The control message processing section 112 in each of the controllers 10-1 and 10-2 requests the node communication section 113 to transmit the packet-out message to the node device 20 located on the boundary with the different network.
- [0275] The node communication section 113 in each of the controllers 10-1 and 10-2, when receiving the transmission request of the packet-out message to the node device 20 located on the boundary with the different network, transmits the packet-out message to the node device 20 via the control channel. For example, the node communication section 113 in the controller 10-1 transmits the packet-out message to the node device 20-3. Similarly, the node communication section 113 in the controller 10-2 transmits the packet-out message to the node device 20-4.
- [0276] When receiving the packet-out message from the controller 10 via the control channel, the communication unit 21 in the node device 20 located on the boundary with the different network transmits the notification packet from the interface unit 22 to the node device 20 in the different network on the basis of the contents of the packet-out message. For example, the communication unit 21 in the node device 20-3 transmits the notification packet to the node device 20-4 via the data communication link. Similarly, the communication unit 21 in the node device 20-4 transmits the notification packet to the node device 20-3 via the data communication link.
- [0277] Also, the communication unit 21 in the node device 20 located on the boundary with the different network, when receiving a notification packet from the node device 20 in the different network through the interface unit 22, transmits a packet-in message

which incorporates the notification packet as inquiry information to the controller 10 that controls the node device 20 via the control channel. When receiving the notification packet from the node device 20-4, for example, the communication unit 21 in the node device 20-3 transmits a packet-in message which incorporates the notification packet to the controller 10-1 via the control channel. Similarly, when receiving a notification packet from the node device 20-3, the communication unit 21 in the node device 20-4 transmits a packet-in message which incorporates the notification packet to the controller 10-2 via the control channel.

[0278] The node communication section 113 in each of the controllers 10-1 and 10-2 receives the above-described packet-in message from the node device 20 located on the boundary with the different network via the control channel, and outputs the packet-in message to the control message processing section 112.

[0279] The control message processing section 112 in each of the controllers 10-1 and 10-2 extracts the notification packet stored in the above packet-in message, and obtains the processing rules ID that indicates the processing rules (flow entries) of the controller 10 in the different network from the contents of the notification packet.

[0280] For example, the controller 10-1 recognizes that a processing rule identified by a processing rule ID "B" is used in the controller 10-2 when transmitting a packet in which the destination address is defined as the address of the controller 10-2 and the source address is defined as the address of the controller 10-1. Similarly, the controller 10-2 recognizes that a processing rule identified by a processing rule ID "A" is used in the controller 10-1, when transmitting a packet in which the destination address is defined as the address of the controller 10-1 and the source address is defined as the controller 10-2.

[0281] (Operation for Setting Sorting Rule in Third Embodiment)

An exemplary operation for setting the sorting rules in the third embodiment is described below.

[0282] When receiving a setting request of processing rules (flow entries) to the node device 20 located on the boundary with the different network from the processing rule management section 118, the control message processing section 112 in each of the controllers 10-1 and 10-2 prepares a sorting rule and stores the prepared sorting rule in the sorting rule storage section 13.

[0283] At this time, the control message processing section 112 in each of the controllers 10-1 and 10-2 specifies identification information (the node device ID) of the node device 20 located on the boundary with the different network as information to be stored in a "node device ID" region in the sorting rule. The control message processing section 112 also specifies identification information (or the interface ID) of the interface unit 22 used for the connection to the node device 20 in the different network

as information to be stored in an "interface ID" region of the sorting rule. The control message processing section 112 also specifies the address of the controller 10 in its own network as information to be stored in an "own address" region of the sorting rule. The control message processing section 112 also specifies the address of the controller 10 in the different network as information of a "counterpart address" region of the sorting rule. The control message processing section 112 specifies the processing rules ID embedded in the processing rule (or the flow entry) as information to be stored in a "processing rule ID" region of the sorting rule.

[0284] Also, the sorting section 12 in each of the controllers 10-1 and 10-2 recognizes a virtual port 14 used to communicate with the controller 10 in the different network, and specifies identification information of the virtual port (or the virtual port ID) as information to be stored in a "virtual port ID" region for the above-described sorting rule.

[0285] It should be noted that the setting method of the sorting rules is not limited to this method. For example, sorting rules may be manually stored in the sorting rule storage section 13 in advance by an operator's input or the like.

[0286] (Configuration of Sorting Rule)

An exemplary format of the sorting rules stored in the sorting rule storage section 13 is described with reference to Figs. 9A and 9B.

[0287] A sorting rule includes a "virtual port ID" region, a "node device ID" region, an "interface ID" region, an "own address" region, a "counterpart address" region and a "processing rule ID" region.

[0288] The "virtual port ID" region contains identification information (or the virtual port ID) of the virtual port 14 used to communicate with the controller 10 in a different network.

[0289] The "node device ID" region contains identification information (or the node device ID) of the node device 20 located on the boundary with the different network.

[0290] The "interface ID" region contains identification information (or the interface ID) of the interface unit 22 used for the connection to the node device 20 in the different network.

[0291] The "own address" region contains the address of its own controller 10.

[0292] The "counterpart address" region contains the address of a communication counterpart.

[0293] The "processing rule ID" region contains identification information (or the processing rule ID) of a processing rule (or a flow entry).

[0294] (Supplement)

The above-described addresses may be a MAC address, an IP address, or other identification information.

[0295] Each of the addresses described in the "own address" region and the "counterpart

address" region may be a destination address or a source address. When a case that the source address of a packet transferred between the controllers 10 is the address described in the "own address" region of a sorting rule and the destination address is the address described in the "counterpart address" region of the sorting rule, for example, this implies that the packet matches with the sorting rule. Also, when a case that the source address of a packet transferred between the controllers 10 is the address described in the "counterpart address" region of a sorting rule and the destination address is the address described in the "own address" region of the sorting rule, this implies that the packet matches with the sorting rule.

[0296] In this embodiment, it is assumed that an "own address" region contains the address of the controller 10 in its own network, and a "counterpart address" region contains the address of a controller 10 in a different network. Moreover, it is assumed that the virtual ports 14 are assigned with addresses in each controller 10. For example, a typical virtual machine (VM) may be assigned with a virtual MAC address, a virtual IP address or the like.

[0297] (Identification Information of Virtual Port and Interface Unit in Third Embodiment)

Here, a virtual port 14 in the controller 10-1 is assigned with a virtual port ID "VP1". A virtual port 14 in the controller 10-2 is assigned with a virtual port ID "VP2". The interface unit 22 used for the connection to the node device 20-4 in the node device 20-3 is assigned with an interface ID "IF3". The interface unit 22 used for the connection to the node device 20-3 in the node device 20-4 is assigned with an interface ID "IF4".

[0298] (Sorting Rule in Controller in Network 1)

Fig. 9A describes an example of the sorting rule in the controller 10-1 in the network 1.

[0299] (1) Sorting Rule That Uses Control Channel

Virtual port ID: VP1

Node device ID: DPID3

Interface ID: IF3

Own address: address of controller 10-1

Counterpart address: address of controller 10-2

Processing rule ID: A

[0300] (Sorting Rule in Controller in Network 2)

Fig. 9B describes an example of a sorting rule in the controller 10-2 in the network 2.

[0301] (1) Sorting Rule That Uses Control Channel

Virtual port ID: VP2

Node device ID: DPID4

Interface ID: IF4

Own address: address of controller 10-2

Counterpart address: address of controller 10-1

Processing rule ID: B

[0302] (Operation of Packet Transfer between Controllers in Third Embodiment)

An exemplary operation for packet transfer between the controllers 10 in the third embodiment is described below.

[0303] The sorting section 12 in each of the controllers 10-1 and 10-2, when receiving a packet from a virtual port 14, determines identification information (or the virtual port ID) of the virtual port 14, the destination address and the source address of the received packet.

[0304] The sorting section 12 in each of the controllers 10-1 and 10-2 searches the sorting rules stored in the sorting rule storage section 13, using the identification information (virtual port ID) of the virtual port 14, the destination address and the source address as search keys, to specify a node device 20 and an interface unit 22 which are suitable as the output destination of the packet.

[0305] For example, the sorting section 12 in the controller 10-1 searches the sorting rules stored in the sorting rule storage section 13, using the virtual port ID "VP1" of the virtual port 14, the destination address (that is, the address of the controller 10-2), the source address (that is, the address of the controller 10-1) as search keys, and consequently specifies the node device ID "DPID3" and the interface ID "IF3".

[0306] Similarly, the sorting section 12 in the controller 10-2 searches the sorting rules stored in the sorting rule storage section 13 using the virtual port ID "VP2" of the virtual port 14, the destination address (that is, the address of the controller 10-1) and the source address (that is, the address of the controller 10-2) as search keys, and consequently specifies the node device ID "DPID4" and the interface ID "IF4".

[0307] The sorting section 12 in each of the controllers 10-1 and 10-2 outputs a message incorporating the packet and an instruction to output the packet to the specified node device 20 and interface unit 22, to the control message processing section 112 in the node device control section 11.

[0308] For example, the sorting section 12 in the controller 10-1 outputs a message incorporating the packet and an instruction to output the packet to the node device 20-3 and the specified interface unit 22 thereof, to the control message processing section 112 in the node device control section 11.

[0309] Similarly, the sorting section 12 in the controller 10-2 outputs a message incorporating the packet and an instruction to output the packet to the node device 20-4 and the specified interface unit 22 thereof, to the control message processing section 112 in the node device control section 11.

[0310] The control message processing section 112 in each of the controllers 10-1 and 10-2

refers to the sorting rules stored in the sorting rule storage section 13 to prepare a packet-out message incorporating the packet and an instruction to transmit the packet, and requests the node communication section 113 to transmit the packet-out message to the node device 20 located on the boundary with the different network.

- [0311] The node communication section 113 in each of the controllers 10-1 and 10-2 transmits the packet-out message received from the control message processing section 112 to the node device 20 located on the boundary with the different network.
- [0312] For example, the node communication section 113 in the controller 10-1 transmits the packet-out message from the control message processing section 112 to the node device 20-3.
- [0313] Similarly, the node communication section 113 in the controller 10-2 transmits the packet-out message from the control message processing section 112 to the node device 20-4.
- [0314] The communication unit 21 in the node device 20 located on the boundary with the different network transmits the packet incorporated in the above-described packet-out message from the specified interface unit 22 to the node device 20 in the different network.
- [0315] For example, the communication unit 21 in the node device 20-3 transmits the packet incorporated in the above-described packet-out message from the specified interface unit 22 to the node device 20-4.
- [0316] Similarly, the communication unit 21 in the node device 20-4 transmits the packet incorporated in the above packet-out message from the specified interface unit 22 to the node device 20-4.
- [0317] Also, the communication unit 21 in the node device 20 located on the boundary with the different network, when receiving the packet from the node device 20 in the different network through the interface unit 22, transmits a packet-in message as inquiry information to the controller 10 that controls the its own node device 20 via the control channel in accordance with the processing rules (or the flow entries). At this time, the communication unit 21 incorporates the packet, the node device ID, the interface ID and the processing rule ID into this packet-in message.
- [0318] For example, the communication unit 21 in the node device 20-3, when receiving a packet from the node device 20-4 via the interface unit 22, transmits a packet-in message as inquiry information to the controller 10-1 that controls the node device 20-3 via the control channel in accordance with the processing rules (or the flow entries). At this time, the communication unit 21 in the node device 20-3 incorporates the packet, the node device ID "DPID3", the interface ID "IF3" and the processing rule ID "A", into this packet-in message.
- [0319] Similarly, the communication unit 21 in the node device 20-4, when receiving a

packet from the node device 20-3 via the interface unit 22, transmits a packet-in message as inquiry information to the controller 10-2 that controls the node device 20-4 via the control channel in accordance with the processing rules (or the flow entries). At this time, the communication unit 21 in the node device 20-4 incorporates the packet, the node device ID "DPID4", the interface ID "IF4" and the processing rule ID "B", into this packet-in message.

- [0320] The node communication section 113 in each of the controllers 10-1 and 10-2 receives the above-described packet-in message from the node device 20 connected thereto via the control channel, and outputs the packet-in message to the control message processing section 112.
- [0321] The control message processing section 112 in each of the controllers 10-1 and 10-2 analyzes the above-described packet-in message and obtains the packet, the node device ID, the interface ID and the processing rule ID. At this time, the control message processing section 112 can obtain the destination address and the source address from the header information of the packet and the like.
- [0322] For example, the control message processing section 112 in the controller 10-1 analyzes the above-described packet-in message and obtains the packet, the node device ID "DPID3", the interface ID "IF3" and the processing rule ID "A". At this time, the control message processing section 112 can obtain the address of the controller 10-1 as the destination address and the address of the controller 10-2 as the source address from the header information of the packet, and the like.
- [0323] Similarly, the control message processing section 112 in the controller 10-2 analyzes the above-described packet-in message and obtains the packet, the node device ID "DPID4", the interface ID "IF4" and the processing rule ID "B". At this time, the control message processing section 112 can obtain the address of the controller 10-2 as the destination address and the address of the controller 10-1 as the source address from the header information of the packet, and the like.
- [0324] The control message processing section 112 in each of the controller 10-1 and 10-2 searches the sorting rules stored in the sorting rule storage section 13 by using at least one of the node device ID, the interface ID, the destination address, the source address and the processing rule ID as a search key(s), and recognizes that the packet is addressed to a virtual port 14 in its own network. The control message processing section 112 then outputs the packet to the sorting section 12 together with the information used as the search key(s). At this time, the control message processing section 112 may output any or all of the node device ID, the interface ID, the destination address, the source address and the processing rule ID together with the packet, to the sorting section 12.
- [0325] The sorting section 12 in each of the controllers 10-1 and 10-2 searches the sorting

rules stored in the sorting rule storage section 13 using at least one of the node device ID, the interface ID, the destination address, the source address and the processing rule ID as a search key(s), and thereby recognizes that the packet is addressed to a virtual port 14 in the controller 10 in its own network, and then outputs the packet to the virtual port 14.

[0326] For example, the sorting section 12 in the controller 10-1 searches the sorting rules stored in the sorting rule storage section 13 using the processing rule ID "A" as a search key, and thereby specifies the virtual port ID "VP1". The sorting section 12 in the controller 10-1 then outputs the packet to the virtual port 14 identified by the virtual port ID "VP1".

[0327] Similarly, the sorting section 12 in the controller 10-2 searches the sorting rules stored in the sorting rule storage section 13 by using the processing rule ID "B" as a search key, and thereby specifies the virtual port ID "VP2". The sorting section 12 in the controller 10-2 then outputs the packet to the virtual port 14 identified by the virtual port ID "VP2".

[0328] (Supplement)

When searching the sorting rules stored in the sorting rule storage section 13 using at least one of the node device ID, the interface ID, the destination address, the source address and the processing rule ID as a search key(s), the control message processing section 112 and the sorting section 12 may use any one of the node device ID, the interface ID, the destination address, the source address and the processing rule ID, or any combination of them, or all of them as the search key(s).

[0329] It should be noted that the sorting section 12 may unconditionally specify the interface unit 22 in the node device 20 located on the boundary with the different network as the output destination of the packet inputted from a predetermined virtual port 14. Similarly, the sorting section 12 may unconditionally specify a predetermined virtual port 14, as the output destination of the packet from the interface unit 22 in the node device 20 located on the boundary with the different network.

[0330] For example, when receiving a packet from a virtual port 14, the sorting section 12 may search the sorting rules stored in the sorting rule storage section 13 using only the virtual port ID as the search key, irrespectively of the destination address and the source address of the inputted packet and specify a node device ID and an interface ID which indicate the output destination. When receiving a packet from the control message processing section 112, on the other hand, the sorting section 12 may search the sorting rules stored in the sorting rule storage section 13 using only the node device ID and the interface ID as the search keys and specify a virtual port ID which indicates the output destination.

[0331] This embodiment eliminates the need for physically connecting interfaces of a

controller and a node device via a data communication link, effectively improving the flexibility in actual implementations.

[0332] <Fourth Embodiment>

A fourth embodiment of the present invention is described below.

[0333] In this embodiment, which is based on the third embodiment, client processes are executed in one of the two controllers, and server processes are executed in the other controller. At this time, the two controllers communicate with each other via the node devices under the control thereof.

[0334] (Configuration of Communication System in Fourth Embodiment)

An exemplary configuration of the communications system of this embodiment is described with reference to Figs. 10A and 10B. It should be noted that Fig. 10A shows an exemplary network configuration of the network 1 and Fig. 10B shows an exemplary network configuration of the network 2.

[0335] The communications system of this embodiment includes a controller 10-1, a controller 10-2, node devices 20-1 to 20-6.

[0336] The controller 10-1 controls the node devices 20-1 to 20-3 and the controller 10-2 controls the node devices 20-4 to 20-6.

[0337] The node devices 20-1 to 20-3 are arranged in the network 1 and the node devices 20-4 to 20-6 are arranged in the network 2. At least one of the node devices 20-1 to 20-3 is connected to at least one of the node devices 20-4 to 20-6 via a data communication link.

[0338] (Configuration of Controller according to Fourth Embodiment)

Exemplary configurations of the controller 10-1 and the controller 10-2 are described below.

[0339] The controllers 10-1 and 10-2 each correspond to the controller 10 shown in Fig. 8.

[0340] The controller 10-1 includes a node device control section 11-1, a sorting section 12-1, a sorting rule storage section 13-1, virtual ports 14-11 and 14-12, an interface management section 15-1 and a client section 16.

[0341] The controller 10-2 contains a node device control section 11-2, a sorting section 12-2, a sorting rule storage section 13-2, virtual ports 14-21 and 14-22, an interface management section 15-2 and a server section 17.

[0342] The node device control section 11-1 and the node device control section 11-2 each correspond to the node device control section 11 shown in Fig. 8.

[0343] Here, the node device control section 11-1 controls each of the node devices 20-1 to 20-3 via a control channel.

[0344] Also, the node device control section 11-2 controls each of the node devices 20-4 to 20-6 via a control channel.

[0345] The sorting sections 12-1 and 12-2 each correspond to the sorting section 12 shown

in Fig. 8.

- [0346] The sorting rule storage sections 13-1 and 13-2 each correspond to the sorting rule storage section 13 shown in Fig. 8.
- [0347] The virtual ports 14-11, 14-12, 14-21 and 14-22 each correspond to the virtual port 14 shown in Fig. 8.
- [0348] The interface management sections 15-1 and 15-2 each correspond to the interface management section 15 shown in Fig. 8. It should be noted that the interface management sections 15-1 and 15-2 may have the same function as the interface management section 15 shown in Fig. 3.
- [0349] The client section 16 executes processes in the seventh layer (the application layer or layer 7) out of the seven layers defined in the OSI reference model. The client section 16 may be realized by a client processing function provided by the OS (operating system) or software of the computer operating as the controller 10. Note that actual implementations of the client section 16 are not limited to those examples.
- [0350] The server section 17 executes processes in the seventh layer (the application layer, layer 7) out of the seven layers defined in the OSI reference model. The server section 17 may be realized by a server processing function provided by the OS (operating system) or software of the computer operating as the controller 10. However, actual implementations of the server section 17 are not limited to those examples.
- [0351] It should be noted that each of the controllers 10-1 and 10-2 may incorporate both of the client section 16 and the server section 17.
- [0352] (Connection Example of Virtual Port in Fourth Embodiment)
 In this embodiment, the virtual port 14-11 in the controller 10-1 is assigned with an address "AD11". The virtual port 14-11 is provided between the sorting section 12-1 and the client section 16, allowing packet transfer between the sorting section 12-1 and the client section 16. Also, the virtual port 14-12 in the controller 10-1 is assigned with an address "AD12". The virtual port 14-11 and the client section 16 may be installed in the same virtual machine (VM). The virtual port 14-12 is provided between the sorting section 12-1 and the interface management section 15-1, allowing packet transfer between the sorting section 12-1 and the interface management section 15-1. In an actual implementation, the sorting section 12-1, the virtual ports 14-11, 14-12 and the interface management section 15-1 may be realized as the functions installed in the same OS (operating system) and software.
- [0353] Similarly, the virtual port 14-21 in the controller 10-2 is assigned with an address "AD21". The virtual port 14-21 is arranged between the sorting section 12-2 and the server section 17, allowing packet transfer between the sorting section 12-2 and the server section 17. The virtual port 14-21 and the server section 17 may be installed in the same virtual machine (VM). Also, the virtual port 14-22 in the controller 10-2 is

assigned with an address "AD22". The virtual port 14-22 is provided between the sorting section 12-2 and the interface management section 15-2, allowing packet transfer between the sorting section 12-2 and the interface management section 15-2. In an actual implementation, the sorting section 12-2, the virtual ports 14-21, 14-22 and the interface management section 15-2 may be realized as the functions installed in the same OS (Operating System) and software.

[0354] (Supplement)

Specifically, the above-described addresses "AD11" and "AD21" may be defined as any region (field) which can be identified in the OpenFlow technique, such as the MAC address, the IP address, the TCP/UDP (User Datagram Protocol) port number and the like, or a combination of them. It should be noted, however, actual implementations are not limited to these examples.

[0355] (Identification Information of Virtual Port in Fourth Embodiment)

The virtual port 14-11 in the controller 10-1 is assigned with a virtual port ID "VP11". The virtual port 14-12 in the controller 10-1 is assigned with a virtual port ID "VP12". The virtual port 14-21 in the controller 10-2 is assigned with a virtual port ID "VP21". The virtual port 14-22 in the controller 10-2 is assigned with a virtual port ID "VP22".

[0356] (Configuration of Node Device in Fourth Embodiment)

Exemplary configurations of the node devices 20-1 to 20-6 are described below.

[0357] The node devices 20-1 to 20-6 each correspond to the node device 20 shown in Fig. 4.

[0358] The node device 20-1 includes a communication unit 21-1, an interface unit 22-11 and an interface unit 22-12. The node device 20-2 includes a communication unit 21-2, an interface unit 22-21 and an interface unit 22-22. The node device 20-3 includes a communication unit 21-3, an interface unit 22-31 and an interface unit 22-32. The node device 20-4 includes a communication unit 21-4, an interface unit 22-41 and an interface unit 22-42. The node device 20-5 includes a communication unit 21-5, an interface unit 22-51 and an interface unit 22-52. The node device 20-6 includes a communication unit 21-6, an interface unit 22-61 and an interface unit 22-62.

[0359] The communication unit 21-1 to 21-6 each correspond to the communication unit 21 shown in Fig. 4.

[0360] The interface units 22-11, 22-12, 22-21, 22-22, 22-31, 22-32, 22-41, 22-42, 22-51, 22-52, 22-61 and 22-62 each correspond to the interface units 22 shown in Fig. 4.

[0361] (Connection Example of Interface Unit in Fourth Embodiment)

The interface unit 22-11 in the node device 20-1 is connected to the interface management section 15-1 in the controller 10-1. The interface unit 22-12 in the node device 20-1 is connected to the interface unit 22-21 in the node device 20-2. The

interface unit 22-22 in the node device 20-2 is connected to the interface unit 22-31 in the node device 20-3. The interface unit 22-32 in the node device 20-3 is connected to the interface unit 22-41 in the node device 20-4. The interface unit 22-42 in the node device 20-4 is connected to the interface unit 22-51 in the node device 20-5. The interface unit 22-52 in the node device 20-5 is connected to the interface unit 22-61 in the node device 20-6. The interface unit 22-62 in the node device 20-6 is connected to the interface management section 15-2 in the controller 10-2.

[0362] The node device 20-3 is arranged in the network 1, and the node device 20-4 is arranged in the network 2. This implies that the interface unit 22-32 in the node device 20-3 and the interface unit 22-41 in the node device 20-4 each serve as an interface unit 22 located on the boundary with a different network.

[0363] (Identification Information of Interface Unit according to Fourth Embodiment)

The interface unit 22-11 is assigned with an interface ID "IF11". The interface unit 22-12 is assigned with an interface ID "IF12". The interface unit 22-21 is assigned with an interface ID "IF21". The interface unit 22-22 is assigned with an interface ID "IF22". The interface unit 22-31 is assigned with an interface ID "IF31". The interface unit 22-32 is assigned with an interface ID "IF32". The interface unit 22-41 is assigned with an interface ID "IF41". The interface unit 22-42 is assigned with an interface ID "IF42". The interface unit 22-51 is assigned with an interface ID "IF51". The interface unit 22-52 is assigned with an interface ID "IF52". The interface unit 22-61 is assigned with an interface ID "IF61". The interface unit 22-62 is assigned with an interface ID "IF62".

[0364] (Sorting Rule in Controller in Network 1)

Fig. 11A shows an example of the sorting rules in the controller 10-1 in the network 1.

[0365] (1) Sorting Rule which Uses Control Channel

Virtual port ID: VP11

Node device ID: DPID3

Interface ID: IF32

Own address: AD11

Counterpart address: AD21

Processing rule ID: A1

[0366] (2) Sorting Rule which Uses Data Communication link

Virtual port ID: VP12

Node device ID: DPID1

Interface ID: IF11

Own address: AD12

Counterpart address: AD22

Processing rule ID: A2

[0367] (Sorting Rule in Controller in Network 2)

Fig. 11B shows an example of the sorting rules in the controller 10-2 in the network 2.

[0368] (1) Sorting Rule which Uses Control Channel

Virtual port ID: VP21

Node device ID: DPID4

Interface ID: IF41

Own address: AD21

Counterpart address: AD11

Processing rule ID: B1

[0369] (2) Sorting Rule which Uses Data Communication Link

Virtual port ID: VP22

Node device ID: DPID6

Interface ID: IF62

Own address: AD22

Counterpart address: AD12

Processing rule ID: A2

[0370] (Registration of Processing Rule (Flow Entry) to Node Device 20-3)

The node device control section 11-1 in the controller 10-1 searches the sorting rules stored in the sorting rule storage section 13-1 and outputs a flow modification message (FlowMod-Add) to set the node device 20-3 with a processing rule (or a flow entry). Here, the destination address "AD11" and the source address "AD21" are specified as the identifying conditions (or the identifying rule) of the processing rule (or the flow entry). Also, the contents of processing (or the action) of the processing rule (flow entry) are specified to perform an operation of outputting a packet-in message incorporates the packet and the processing rule ID "A1" to the controller 10-1 (or the output port connected to the controller 10-1). That is, the node device control section 11-1 outputs to the node device 20-3 a flow modification message (FlowMod-Add) to instruct the node device 20-3 to carry out a process of outputting the packet-in message which incorporates the packet and the processing rule ID "A1", when the interface unit 22-32 receives the packet in which the destination address is "AD11" and the source address is "AD21".

[0371] (Registration of Processing Rule (Flow Entry) to Node Device 20-4)

The node device control section 11-2 in the controller 10-2 searches the sorting rules stored in the sorting rule storage section 13-2, and outputs a flow modification message (FlowMod-Add) to set the node device 20-4 with a processing rule (or a flow entry). Here, the destination address "AD21" and the source address "AD11" are specified as

identifying conditions (or an identifying rule) of the processing rule (or the flow entry). Also, the contents of processing (or the action) of the processing rule (or the flow entry) are specified to perform an operation of outputting a packet-in message which incorporates the packet and the processing rule ID "B1" to the controller 10-2 (or the output port connected to the controller 10-2). That is, the node device control section 11-2 outputs to the node device 20-4 a flow modification message (FlowMod-Add) to instruct the node device 20-4 to carry out a process of outputting the packet-in message which incorporates the packet and the processing rule ID "B1", when the interface unit 22-23 receives the packet in which the destination address is "AD21" and the transmission source address is "AD11",.

[0372] (Packet Output Instruction to Node device 20-3)

When receiving a packet from the virtual port 14-11, the sorting section 12-1 in the controller 10-1, searches the sorting rules stored in the sorting rule storage section 13-1, and if the destination address of the packet is "AD21" and the source address of the packet is "AD11", outputs a message of requesting the node device control section 11-1 to output the packet from the interface unit 22-32 in the node device 20-3.

When receiving the above-described message from the sorting section 12-1, the node device control section 11-1 transmits a packet-out message which incorporates the packet and a instruction of outputting the packet to the interface unit 22-12, to the node device 20-3 via the control channel.

[0373] (Packet Output Instruction to Node device 20-4)

When receiving a packet from the virtual port 14-21, the sorting section 12-2 in the controller 10-2, searches the sorting rules stored in the sorting rule storage section 13-2, and if the destination address of the packet is "AD11" and the source address of the packet is "AD21", outputs a message of requesting the node device control section 11-2 to output the packet from the interface unit 22-41 in the node device 20-4.

[0374] (Operation for Transferring Packet to Server Section from Client Section)

An exemplary operation for transferring a packet to the server section 17 from the client section 16 is described below.

[0375] The client section 16 in the controller 10-1 outputs a packet in which the destination address is defined as "AD21" and the source address is defined as "AD11", to the virtual port 14-11.

[0376] When receiving the packet from the client section 16, the virtual port 14-11 outputs the packet to the sorting section 12-1.

[0377] When receiving the packet from the virtual port 14-11, the sorting section 12-1 searches the sorting rules stored in the sorting rule storage section 13-1 and outputs a message of requesting the node device control section 11-1 to output the packet from the interface unit 22-32 in the node device 20-3, since the destination address of the

packet is "AD21" and the source address of the packet is "AD11".

- [0378] When receiving the above-described message from the sorting section 12-1, the node device control section 11-1 transmits a packet-out message which incorporates the packet and an instruction of outputting the packet to the interface unit 22-32, to the communication unit 21-3 in the node device 20-3.
- [0379] When receiving the packet-out message from the node device control section 11-1 in the controller 10-1, the communication unit 21-3 in the node device 20-3 outputs the packet incorporated in the packet-out message to the interface unit 22-32.
- [0380] When receiving the packet from the communication unit 21-3, the interface unit 22-32 transfers the packet to the interface unit 22-41 in the node device 20-4, which is the connection destination, via the data communication link.
- [0381] When receiving the packet from the interface unit 22-32 in the node device 20-3, the interface unit 22-41 in the node device 20-4 outputs the packet to the communication unit 21-4.
- [0382] When receiving the packet from the interface unit 22-41, the communication unit 21-4 processes the inputted packet in accordance with the processing rules (the flow entries).
- [0383] In this embodiment, the communication unit 21-4 is set with a processing rule (or a flow entry) by the node device control section 11-2 in the controller 10-2, wherein the processing rule instructs to, when receiving a packet in which the destination address is "AD21" and the source address is ""AD11" in the interface unit 22-41, output a packet-in message which incorporates the packet and the processing rule ID "B1" to the controller 10-2.
- [0384] Accordingly, the communication unit 21-4 outputs the packet-in message incorporating the packet, which complies with the identifying conditions (or the identifying rule) of the processing rule (or the flow entry), and the processing rule ID "B1", to the node device control section 11-2 in the controller 10-2, via the control channel.
- [0385] When receiving the above-described packet-in message from the communication unit 21-4 in the node device 20-4, the node device control section 11-2 in the controller 10-2 outputs a message which incorporates the above-described packet and the processing rule ID "B1" to the sorting section 12-2.
- [0386] When receiving the above message from the node device control section 11-2, the sorting section 12-2 searches the sorting rules stored in the sorting rule storage section 13-2, and determines that the packet is to be outputted to the virtual port 14-21 on the basis of the processing rule ID "B1" included in the above message, and outputs the packet incorporated in the above message to the virtual port 14-21.
- [0387] When receiving the packet from the sorting section 12-2, the virtual port 14-21 outputs the packet to the server section 17, which is the connection destination thereof.

- [0388] The server section 17 processes the packet when receiving the packet from the virtual port 14-21.
- [0389] (Operation for Transferring Packet to Client Section from Server Section)
Next, an exemplary operation of transferring a packet from the server section 17 to the client section 16 is described below.
- [0390] The server section 17 in the controller 10-2 outputs a packet in which the destination address is defined as "AD11" and the source address is defined as "AD21" to the virtual port 14-21.
- [0391] When receiving the packet from the server section 17, the virtual port 14-21 outputs the packet to the sorting section 12-1.
- [0392] When receiving the packet from the virtual port 14-21, the sorting section 12-1 searches the sorting rules stored in the sorting rule storage section 13-1. Since the destination address of the packet is "AD11" and the source address of the packet is "AD21", the sorting section 12-1 outputs a message of requesting the node device control section 11-2 to output the packet from the interface unit 22-41 in the node device 20-4.
- [0393] When receiving the above message from the sorting section 12-2, the node device control section 11-2 transmits a packet-out message which incorporates the packet and an instruction of outputting the packet to the interface unit 22-41, to the communication unit 21-4 in the node device 20-4 via the control channel.
- [0394] When receiving the packet-out message from the node device control section 11-2 in the controller 10-2, the communication unit 21-4 in the node device 20-4 outputs the packet incorporated in the packet-out message to the interface unit 22-41.
- [0395] When receiving the packet from the communication unit 21-4, the interface unit 22-41 transfers the packet to the interface unit 22-32 in the node device 20-3, which is the connection destination, through the data communication link.
- [0396] When receiving the packet from the interface unit 22-41 in the node device 20-4, the interface unit 22-32 in the node device 20-3 outputs the packet to the communication unit 21-3.
- [0397] When receiving the packet from the interface unit 22-32, the communication unit 21-3 processes the received packet on the basis of the processing rules (or the flow entries).
- [0398] Here, the communication unit 21-3 is set with a processing rule (or a flow entry) from the node device control section 11-1 in the controller 10-1, wherein the processing rule instructs to, when receiving a packet in which the destination address is defined as "AD11" and the source address is defined as "AD21" in the interface unit 22-32, output a packet-in message which incorporates the packet and the processing rule ID "A1" to the controller 10-1.

- [0399] As a result, the communication unit 21-3 outputs the packet-in message which incorporates the packet complying with the identifying conditions (or the identifying rule) of the processing rule (or the flow entry) and the processing rule ID "A1", to the node device control section 11-1 in the controller 10-1 via the control channel.
- [0400] When receiving the above packet-in message from the communication unit 21-3 in the node device 20-3, the node device control section 11-1 in the controller 10-1 outputs a message which incorporates the above packet and the processing rule ID "A1" to the sorting section 12-1.
- [0401] When receiving the above message from the node device control section 11-1, the sorting section 12-1 searches the sorting rules stored in the sorting rule storage section 13-1, and determines that the packet is to be outputted to the virtual port 14-11 on the basis of the processing rule ID "A1" incorporated in the above message, and then outputs the packet incorporated in the above message to the virtual port 14-11.
- [0402] When receiving the packet from the sorting section 12-1, the virtual port 14-11 outputs the packet to the client section 16, which is the connection destination.
- [0403] The client section 16 processes the packet when receiving the packet from the virtual port 14-11.
- [0404] (Communications via Data Communication Link)
In the fourth embodiment, communications between the controllers 10 can be carried out by packet transfer via a data communication link, similarly to the first embodiment.
- [0405] (1) Setting of Processing Rules (Flow Entries) to Each Node Device 20
Initially, the node device control section 11 in each of the controllers 10-1 and 10-2 calculates a route to connect the end points in each network and sets each node device 20 on the route with processing rules (or flow entries) for the packet transfer.
- [0406] For example, the node device control section 11-1 in the controller 10-1 calculates a route to connect the interface unit 22-11 in the node device 20-1 with the interface unit 22-32 in the node device 20-3 in the network 1 and sets each node device 20 on the route with processing rules (flow entries) for the packet transfer.
- [0407] Similarly, the node device control section 11-2 in the controller 10-2 calculates a route to connect the interface unit 22-41 in the node device 20-4 to the interface unit 22-62 in the node device 20-6 in the network 2 and sets each node device 20 on the route with processing rules (flow entries) for the packet transfer.
- [0408] (2) Transmission and Reception of Packet through Link for Data Communication
The sorting section 12 in each of the controller 10-1 and the controller 10-2 then outputs a packet which is to be transferred through the data communication link to the virtual port 14 connected to the interface management section 15. Consequently, the packet is outputted from the interface management section 15. At this time, the sorting section 12 may refer to the sorting rule stored in the sorting rule storage section 13 and

determine the virtual port 14 of the output destination.

- [0409] For example, the sorting section 12-1 in the controller 10-1 outputs a packet in which the destination address is "AD22" and the source address is "AD12" to the virtual port 14-12. As a result the packet is outputted from the interface management section 15-1. In this case, the client section 16 in the controller 10-1 may output the packet in which the destination address is "AD22" and the transmission source address is AD12" via the virtual port 14-11 to the sorting section 12-1.
- [0410] Similarly, the sorting section 12-2 in the controller 10-1 outputs a packet in which the destination address is "AD12" and the source address is "AD22" to the virtual port 14-22. As a result, the packet is outputted from the interface management section 15-2. In this case, the server section 17 in the controller 10-1 may output the packet in which the destination address is "AD12" and the source address is "AD22" via the virtual port 14-21 to the sorting section 12-2.
- [0411] The communication unit 21 in each node device 20 transfers packets in accordance with the processing rules (or the flow entry).
- [0412] The interface management section 15 in each of the controller 10-1 and the controller 10-2 outputs packets which are received from a node device 20 via a data communication link to the sorting section 12 via a virtual port 14.
- [0413] For example, the interface management section 15-1 in the controller 10-1 outputs a packet received from the node device 20-1 to the sorting section 12-1 via the data communication link through the virtual port 14-12. In this case, the sorting section 12-1 in the controller 10-1 may output the packet received from the virtual port 14-12 to the client section 16, through the virtual port 14-11.
- [0414] Similarly, the interface management section 15-2 in the controller 10-2 outputs a packet received from the node device 20-6 to the sorting section 12-2 via the data communication link through the virtual port 14-22. In this case, the sorting section 12-2 in the controller 10-2 may output the packet received from the virtual port 14-22 to the server section 17 through the virtual port 14-21.
- [0415] (Supplement)
The sorting section 12 in each controller 10 may perform packet transfer between the virtual ports 14 in the controller 10 without referring to the sorting rules stored in the sorting rule storage section 13.
- [0416] It should be noted that the sorting rule storage section 13 may store the sorting rules of packets exchanged among the virtual ports 14 in the controller 10.
- [0417] When one of the controllers is used as a terminal device connected to the node device, this embodiment allows communications that uses a conventional communication method, such as TCP/IP and the like, between the terminal apparatus and the controller.

[0418] <Advantage of Communications Systems disclosed in Above-Described Embodiments>

Conventionally, there is a difficulty in achieving communications among controllers by using a conventional communication method such as TCP/IP in a CD-separated type network, such as OpenFlow networks. The use of the systems disclosed in these embodiments of the present invention enables communications based on a conventional communication method such as TCP/IP between or among a plurality of controllers.

[0419] This allows achieving a distributed control of the entire system by a plurality of controllers by reusing distributed-control applications based on a conventional communication architecture such as TCP/IP, making it easy to establish a large-scale system.

[0420] It should the above-mentioned respective embodiments can be carried out by combining them.

[0421] <Examples of Hardware Configuration>

Examples of hardware devices which may be used in the communications system according the present invention is described below.

[0422] Examples of devices which may be used as the controllers include a computer such as PC (personal computer), an appliance, a thin client server, a workstation, a main frame, a super computer. Note that the controllers may be a relaying device or a peripheral device, not limited to a terminal device or a server. Also, an expansion board installed in a computer or the like may be used as the controller, or a virtual machine (VM) established on a physical machine may be used as the controller.

[0423] Examples of devices which may be used as the node devices include a network switch, a router, a proxy, a gateway, a firewall, a load balancer, a packet shaper, a SCADA (supervisory control and data acquisition) security monitoring controller, a gatekeeper, a base station, an access point (AP), a communication satellite (CS), and a computer having a plurality of communication ports. Also, a virtual switch operating on a virtual machine (VM) established on a physical machine may be used as the node device. The controllers and the node devices may be installed on movable bodies such as vehicles, ships and airplanes.

[0424] In one example, as shown in Fig. 12, each controller 10 may each include a storage device (or memory) 31, a processor 32, and an interface 33. The storage device 31 stores a software program 31a which includes codes describing the above-described operations of the controller 10. The storage device 31 is also used to store various data used and generated in the operations of the controller 10. The processor 32 executes the software program 31a to perform the above-described operations of the controller 10. The interface 33 is used to communicate with the node devices 20. A non-transitory recording medium 50 may be used to install the software program 31a onto

the storage device 31.

- [0425] Similarly, as shown in Fig. 13, each node device 20 may each include a storage device (or memory) 41, a processor 42, and interfaces 43 and 44. The storage device 41 stores a software program 41a which includes codes describing the above-described operations of the node device 20. The storage device 41 is also used to store various data used and generated in the operations of the node device 20. The processor 32 executes the software program 41a to perform the above-described operations of the node device 10. The interface 43 is used to communicate with another node device 20, and interface 44 is used to communicate with a controller 10. A non-transitory recording medium 60 may be used to install the software program 41a onto the storage device 41.
- [0426] Examples of the processors 32 and 42 include a CPU (central processing unit), a network processor (NP), a microprocessor, a microcontroller, and a large scale integrated circuit (LSI) having a dedicated function and the like.
- [0427] Examples of the storage devices (or memories) 31 and 41 include a semiconductor storage device such as a RAM (Random Access Memory), a ROM (Read Only Memory), an EEPROM (Electrically Erasable and Programmable Read Only Memory), a flash memory, an auxiliary storage device such as an HDD (Hard Disk Drive), an SSD (Solid State Drive), a removable disk such as a DVD (Digital Versatile Disk), a storage medium such as an SD memory card (Secure Digital memory card), and the like. Also, a buffer, a register and the like may be used as the storage devices (or memories) 31 and 41. In one embodiment, a storage device that uses a DAS (direct attached storage), an FC-SAN (fiber channel - storage area network), an NAS (network attached storage), IP-SAN (IP - storage area network) and the like may be used the storage devices (or memories) 31 and 41.
- [0428] The processor 31 and the storage device 32 may be monolithically integrated and the processor 41 and the storage device 42 may be monolithically integrated. In recent years, one-chip microcomputers have been made popular. In one embodiment, a one-chip microcomputer installed in an electronic appliance or the like may monolithically integrate the above-described processor and storage device.
- [0429] Examples of the above-described interface includes a circuit board (a mother board, an I/O board) and a semiconductor integrated circuit which are adapted to a network communication, a network adaptor such as an NIC (network interface card), a similar expansion card, a communication apparatus such as an antenna, and a communication port such as a connection port (connector).
- [0430] Also, examples of the network include the Internet, an LAN (local area network), a wireless LAN, a WAN (wide area network), a backbone, a cable television (CATV) line, a fixed telephone network, a mobile telephone network, WiMAX (IEEE 802.16a),

a 3G (3rd generation) communication system, a dedicated line (lease line), an IrDA (infrared data association), Bluetooth (Registered Trademark), a serial communication line, a data bus and the like.

[0431] Configuration elements included in each of the controllers and the node devices may be modules and components, or dedicated devices, or starting (calling) programs for them.

[0432] It should be noted that actual implementations are not limited to these examples.

[0433] <Summary>

As discussed above, the communications system in exemplary embodiments of the present invention includes controllers and node devices. The controllers control packet processing in the respective node devices.

[0434] The node devices outputs packets from their own physical or logical interfaces under control of the controllers.

[0435] Each controller each obtains identifying conditions (identifying rules) of packets used to communicate with a network that is not under control of the controller (that is, a network provided outside its own network).

[0436] Also, each controller calculates the transfer route which connects an interface of one node device and another interface of a boundary node device located on the boundary with a different network, which is used for establishing a connection to a device provided outside its own network.

[0437] In one embodiment, the controllers may each calculate a transfer route in which a start point is defined as an interface of a node device connected to the controller via a data communication link and an end point is defined as an interface of a boundary node device which is used for establishing a connection to a device provided outside its own network. Alternatively, the controllers may each calculates a transfer route in which a start point is defined as an interface of a boundary node device which is used for establishing a connection to a device provided outside its own network and an end point is defined as an interface of a node device connected to the controller via a data communication link.

[0438] Also, the controllers each sets the node devices with processing rules (or flow entries) so that packets complying with the identifying conditions (or the identifying rules) are transferred on the calculated transfer route.

[0439] <Supplementary note>

Some or all of the above-mentioned embodiments may be represented as described in the following supplementary notes. It should be noted that actual implementations are not limited to the following supplementary nodes.

[0440] (Supplementary note 1)

A communications system in which controllers control packet processing in each of

node devices, and the node devices each output packets any interfaces thereof under control of a controller connected thereto,

wherein each of the controllers includes:

a node communication section which sets a control channel to control each of the node devices and transmits and receives control messages;

a network interface connected to one of node devices via a data communication link;

an adjacency discovery section which discovers a boundary node device from the node devices, the boundary node being located on the boundary with a different network that is controlled by a different controller;

an identifying condition calculating section calculating identifying conditions (or identifying rules) of packets used to communicate with a controller in the network adjacent thereto;

a route calculating section that calculates a transfer route having a start point determined as a node device connected to an interface of the controller, through which a packet is outputted to an interface of the boundary node device, the interface being connected to a different network outside its own network, and a transfer route having a start point determined as the boundary node device, through which a packet is outputted to an interface of the node device connected to the interface of the controller; and

a processing rule calculating section that sets the node devices connected to the each controller with processing rules (flow entries) so as to transfer packets complying with the identifying conditions (the identifying rules) on the transfer route.

[0441] (Supplementary note 2)

The communications system set forth in supplementary note 1, wherein the adjacency discovery section embeds unique identification information of each controller into retrieval packets used to retrieve a connection relation among the node devices inside each network.

[0442] (Supplementary note 3)

The communications system set forth in the supplementary note 1, wherein the controller instructs the boundary node device to output a packet which incorporates identifying conditions (identifying rules) from the interface connected to the different network.

[0443] (Supplementary note 4)

The communications system described in the supplementary note 1, wherein the identifying condition calculating section incorporates identifying conditions used in the communication between the controllers into a retrieval packet used to retrieve a connection relation among the node devices inside each network.

[0444] (Supplementary note 5)

The communications system set forth in any one of supplementary notes 1 to 4, wherein the route calculation section refers to identification information incorporated in the retrieval packet transmitted to the controller through an interface by a node device, and, if it is equal to identification information indicative of its own controller, determines the interface of the node device connected to the controller as an end point in the route calculation.

[0445] (Supplementary note 6)

The communications system set forth in any one of supplementary notes 1 to 5, wherein the network interface is physically connected through a network link connection cable.

[0446] (Supplementary note 7)

The communications system set forth in any one of supplementary notes 1 to 5, wherein the controller contains one or more virtual ports for transmitting and receiving packets, a sorting rule storage section storing one or more sorting rules of packets and a sorting section for specifying a sorting destination of packets,

wherein the sorting rule storage section retrieves and return selected one of the sorting rules in response to a reference request, and

wherein the sorting section specifies a transfer destination of packets transmitted and received between the interface of the boundary node device and the virtual ports, in accordance with the sorting rule selected by referring to the sorting rule storage section.

[0447] (Supplementary note 8)

The communications system set forth in any one of supplementary notes 1 to 5, wherein the communications system includes a plurality of controllers, an interface section in a node device controlled by one of the controllers is connected via a communication line to an interface section in a node device controlled by a different one of the controllers,

wherein each of the controllers contains an wide area control section that communicates with a different one of the controllers,

wherein the wide area control sections are each connected to one or more of the virtual ports, and

wherein the wide area control sections communicate with each other through the virtual ports.

[0448] (Supplementary note 9)

A communicating method in which controllers control packet processing in each of node devices, and the node devices each output packets any interfaces thereof under control of a controller connected thereto,

wherein each of the controllers is connected via a network interface which achieves a data transfer link connection to one or more of the node devices, and

wherein the communication method includes:

discovering a boundary node device from the node devices, located on a boundary with a different network that is controlled by a different controller;

calculating identifying conditions (or identifying rules) of packets used to communicate with the different controller in the different network;

calculating a transfer route having a start point determined as a node device connected to an interface of the controller, through which a packet is outputted to an interface of the boundary node device, the interface being connected to a different network outside its own network, and a transfer route having a start point determined as the boundary node device, through which a packet is outputted to an interface of the node device connected to the interface of the controller; and

setting the node devices connected to the each controller with processing rules (flow entries) so as to transfer packets complying with the identifying conditions (the identifying rules) on the transfer route, and

processing a packet complying with the identifying conditions (of the identifying rules) of a processing rule in accordance with the processing rule (flow entry).

[0449] (Supplementary note 10)

The communicating method system described in supplementary note 9, wherein each controller embeds unique identification information of the each controller into retrieval packets used to retrieve a connection relation among the node devices inside each network,

wherein the controller refers to identification information incorporated in the retrieval packet, and compares the identification information incorporated in the packet with its own identification information and consequently determines whether the retrieval packet comes from a different controller.

[0450] (Supplementary note 11)

The communicating method set forth in supplementary note 9, wherein the controller instructs the boundary node device to output a packet incorporating identifying conditions (or the identifying rules) from the interface connected to the different network.

[0451] (Supplementary note 12)

The communicating method set forth in supplementary note 9, wherein the controller incorporates packet judgment conditions used in the communication between the controllers into a retrieval packet used to retrieve the connection relation between the node devices in its own network.

[0452] (Supplementary note 13)

The communicating method set forth in any one of supplementary notes 9 to 12, wherein each controller refers to identification information incorporated in the retrieval

packet transmitted to the controller through the interface from the node device, and if the identification information is equal to its own identification information, determines the interface of the node device connected to the controller as an end point in the route calculation.

[0453] (Supplementary note 14)

The communicating method described in one of the supplementary notes 9 to 13, wherein the controller is physically connected through a cable for a network link connection to the node device.

[0454] (Supplementary note 15)

A controller for controlling packet processing of node devices, including:
a network interface for establishing a connection to one of the node devices via a data transfer link;

an adjacency discovery section for discovering a boundary node device from the node devices, the boundary node device being located on the boundary with a different network that is controlled by a different controller;

an identifying condition calculating section calculating identifying conditions (or identifying rules) of packets used to communicate with a controller in the network adjacent thereto;

a route calculating section that calculates a transfer route having a start point determined as a node device connected to an interface of the controller, through which a packet is outputted to an interface of the boundary node device, the interface being connected to a different network outside its own network, and a transfer route having a start point determined as the boundary node device, through which a packet is outputted to an interface of the node device connected to the interface of the controller; and

a processing rule calculating section that sets the node devices connected to the each controller with processing rules (flow entries) so as to transfer packets complying with the identifying conditions (the identifying rules) on the transfer route.

[0455] (Supplementary note 16)

The controller described in the supplementary note 15, wherein the adjacency discovery section embeds unique identification information of each controller into retrieval packets used to retrieve a connection relation among the node devices inside each network.

[0456] (Supplementary note 17)

The controller described in the supplementary note 15, wherein the identifying condition calculating section of the controller instructs the boundary node device to output a packet which incorporates identifying conditions (identifying rules) from the interface connected to the different network.

[0457] (Supplementary note 18)

The controller described in the supplementary note 15, wherein the identifying condition calculating section incorporates identifying conditions used in the communication between the controllers into a retrieval packet used to retrieve a connection relation among the node devices inside each network.

[0458] (Supplementary note 19)

The controller described in one of the supplementary notes 15 to 18, wherein the route calculation section refers to identification information incorporated in the retrieval packet transmitted to the controller through an interface by a node device, and, if it is equal to identification information indicative of its own controller, determines the interface of the node device connected to the controller as an end point in the route calculation.

[0459] (Supplementary note 20)

The controller described in one of the supplementary notes 15 to 20, wherein the network interface is physically connected through a network link connection cable.

[0460] It should be noted that an information processing apparatus may be used as the above-described controller. Also, a communicating apparatus may be used as the above-described node device.

[0461] <Remark>

While the invention has been particularly shown and described with reference to exemplary embodiments thereof, the invention is not limited to these examples. It will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the present invention as defined by the claims.

[0462] This application is based upon claims the benefit of priority from Japanese patent application No. 2012-068286, filed on March 23, 2012, the disclosure of which is incorporated herein in its entirety by reference.

Claims

- [Claim 1] A communications system, comprising:
a first node device provided in a first network;
a first controller controlling said first node device;
a second node device provided in a second network and connected to said first node device; and
a second controller controlling said second node device,
wherein said first controller sets said first node device with a processing rule according to which packets transferred between said first and second controllers are be processed,
wherein said second controller sets said second node device with a processing rule according to which said packets are processed, and
wherein said first and second controllers exchanges said packets each other through at least said first and second node devices.
- [Claim 2] The communications system according to claim 1, wherein said first node device requests said first controller for a processing rule for performing processing on a packet from said second controller,
wherein said first controller determines said first node device as a node device located on a boundary with a different network based on said request of said first node device, wherein said second node device requests said second controller for a processing rule for performing processing on a packet from said first controller, and wherein said second node device requests said second controller for a processing rule for performing processing on a packet from said first controller, and wherein said second controller determines said second node device as a node device located on a boundary with a different network based on said request of said second node device.
- [Claim 3] The communications system according to claim 2, wherein said first controller determines that said first node device is a node device located on a boundary with a network managed by said second controller, based on a packet incorporating an identifier of said second controller and received from said second controller, and wherein said second controller determines that said second node device is a node device located on a boundary with a network managed by said first controller, based on a packet incorporating an identifier of said first controller and received from said first controller.
- [Claim 4] The communications system according to claim 2 or 3, wherein said

first controller transmits a packet for collecting information related to connections of node devices controlled by said first controller and receives from said second controller a packet for said second controller to collect connections of node devices controlled by said second controller, wherein said second controller transmits a packet for collecting information related to connections of node devices controlled by said second controller and receives from said first controller a packet for said first controller to collect connections of node devices controlled by said first controller.

[Claim 5] The communications system according to any one of claims 1 to 4, wherein said first and second controllers exchange information related to packets used for communications between said first and second controllers via at least said first and second node devices.

[Claim 6] The communications system according to any one of claims 1 to 5, wherein said first controller is further connected to said first node device via a first data communication link, wherein said first controller transmits and receives packets to be exchanged between said first and second controllers to and from said first node device via said first data communication link, wherein said second controller is further connected to said second node device via a second data communication link, and wherein said second controller transmits and receives packets to be exchanged between said first and second controllers, to and from said second node device via said second data communication link.

[Claim 7] The communications system according to claim any one of claims 1 to 6, wherein said first controller transmits and receives packets to be exchanged between said first and second controllers to and from said first node device via a control channel established between said first controller and said first node device, and wherein said second controller transmits and receives packets to be exchanged between said first and second controllers to and from said second node device via a control channel established between said second controller and said second node device.

[Claim 8] The communications system according to any one of claims 1 to 7, wherein said first controller sets said first node device with a processing rule for transferring a packet to be transmitted from said first controller to said second controller, wherein said first node device transfers to said second node device a packet to be transmitted from said first controller to said second controller in accordance with the

processing rule set by said first controller, wherein said second controller sets said second node device with a processing rule for transferring a packet to be transmitted from said second controller to said first controller, wherein said second node device transfers to said first node device a packet to be transmitted from said second controller to said first controller in accordance with the processing rule set by said second controller.

[Claim 9]

The communications system according to any one of claims 1 to 8, wherein said first controller sets said first node device with a processing rule for transferring to said first controller a packet to be transmitted from said second controller to said first controller, wherein, when a first received packet which said first node device receives from said second node device is a packet to be transferred from said second controller to said first controller, said first node device transfers said first received packet to said first controller in accordance with the processing rule set by said first controller, wherein said second controller sets said second node device with a processing rule for transferring to said second controller a packet to be transmitted from said first controller to said second controller, and wherein, when a second received packet which said second node device receives from said first node device is a packet to be transferred from said first controller to said second controller, said second node device transfers said second received packet to said second controller in accordance with the processing rule set by said second controller.

[Claim 10]

The communications system according to any one of claims 1 to 9, wherein said first controller stores first sorting rules which correlate virtual ports of said first controller and interfaces of said first node and sorts inputted packets to any of the virtual ports of said first controller and the interfaces of said first node device in accordance with said first sorting rules, and wherein said second controller stores second sorting rules which correlate virtual ports of said second controller and interfaces of said second node and sorts inputted packets to any of the virtual ports of said second controller and the interfaces of said second node device in accordance with said second sorting rules.

[Claim 11]

The communications system according to claim 10, wherein, when a first input packet is inputted from a virtual port of said first controller, said first controller outputs to said first node said first input packet and an instruction to output said first input packet from an interface of said

first node in accordance with said first sorting rules, wherein, when a second input packet is inputted from an interface of said first node device, said first controller outputs said second input packet to a virtual port of said first controller in accordance with said first sorting rules, wherein, when a third input packet is inputted from a virtual port of said second controller, said second controller outputs to said second node said third input packet and an instruction to output said third input packet from an interface of said second node device in accordance with said second sorting rules, wherein, when a fourth input packet is inputted from an interface of said second node device, said second controller outputs said fourth input packet to a virtual port of said second controller in accordance with said second sorting rules.

[Claim 12] An information processing apparatus used in any one of the first and second controllers in the communications system according to any one of claims 1 to 11.

[Claim 13] A communication device used in any one of the first and second node devices in the communications system according to any one of claims 1 to 11.

[Claim 14] A communication method, comprising:
by a first controller controlling a first node device provided in a first network, setting said first node device with a processing rule according to which packets transferred between said first controller and a second controller controlling a second node provided in a second network are processed;
by said second controller, setting said second node device with a processing rule according to which said packets are processed;
establishing a connection between said first and second node devices;
and
exchanging said packets between the first and second controllers through at least the first and second node devices.

[Claim 15] The communication method according to claim 14, further comprising:
said first node device requesting said first controller for a processing rule for performing processing on a packet from said second controller;
said first controller determining said first node device as a node device located on a boundary with a different network based on said request of said first node device, said second node device requesting said second controller for a processing rule for performing processing on a packet from said first controller; and said second controller determining said

- second node device as a node device located on a boundary with a different network based on said request of said second node device.
- [Claim 16] The communication method according to claim 15, further comprising: said first controller determining that said first node device is a node device located on a boundary with a network managed by said second controller, based on a packet incorporating an identifier of said second controller and received from said second controller; and said second controller determining that said second node device is a node device located on a boundary with a network managed by said first controller, based on a packet incorporating an identifier of said first controller and received from said first controller.
- [Claim 17] The communication method according to claim 15 or 16, further comprising: said first controller transmitting a packet for collecting information related to connections of node devices controlled by said first controller; said second controller transmitting a packet for collecting information related to connections of node devices controlled by said second controller; said first controller receiving from said second controller a packet for said second controller to collect connections of node devices controlled by said second controller; and said second controller receiving from said first controller a packet for said first controller to collect connections of node devices controlled by said first controller.
- [Claim 18] The communication method according to any one of claims 14 to 17, wherein said first and second controllers exchange information related to packets used for communications between said first and second controllers via at least said first and second node devices.
- [Claim 19] The communication method according to any one of claims 14 to 18, wherein said first controller is further connected to said first node device via a first data communication link, wherein said second controller is further connected to said second node device via a second data communication link, and wherein said method further comprises: said first controller transmitting and receiving packets to be exchanged between said first and second controllers to and from said first node device via said first data communication link, and said second controller transmitting and receiving packets to be exchanged between said first and second controllers, to and from said second node device via said second data communication link.
- [Claim 20] The communication method according to claim any one of claims 14 to

19, further comprising: said first controller transmitting and receiving packets to be exchanged between said first and second controllers to and from said first node device via a control channel established between said first controller and said first node device; and said second controller transmitting and receiving packets to be exchanged between said first and second controllers to and from said second node device via a control channel established between said second controller and said second node device.

[Claim 21]

The communication method according to any one of claims 14 to 20, further comprising: said first controller setting said first node device with a processing rule for transferring a packet to be transmitted from said first controller to said second controller; said first node device transferring to said second node device a packet to be transmitted from said first controller to said second controller in accordance with the processing rule set by said first controller, said second controller setting said second node device with a processing rule for transferring a packet to be transmitted from said second controller to said first controller; and said second node device transferring to said first node device a packet to be transmitted from said second controller to said first controller in accordance with the processing rule set by said second controller.

[Claim 22]

The communication method according to any one of claims 14 to 21, further comprising: said first controller setting said first node device with a processing rule for transferring to said first controller a packet to be transmitted from said second controller to said first controller; when a first received packet which said first node device receives from said second node device is a packet to be transferred from said second controller to said first controller, transferring said first received packet to said first controller by said first node device in accordance with the processing rule set by said first controller, said second controller setting said second node device with a processing rule for transferring to said second controller a packet to be transmitted from said first controller to said second controller; and when a second received packet which said second node device receives from said first node device is a packet to be transferred from said first controller to said second controller, transferring said second received packet to said second controller by said second node device in accordance with the processing rule set by said second controller.

[Claim 23]

The communication method according to any one of claims 14 to 22,

further comprising: said first controller storing first sorting rules which correlate virtual ports of said first controller and interfaces of said first node; said first controller sorting inputted packets to any of the virtual ports of said first controller and the interfaces of said first node device in accordance with said first sorting rules; said second controller storing second sorting rules which correlate virtual ports of said second controller and interfaces of said second node; and said second controller sorting inputted packets to any of the virtual ports of said second controller and the interfaces of said second node device in accordance with said second sorting rules.

[Claim 24]

The communication method according to claim 23, further comprising: when a first input packet is inputted from a virtual port of said first controller, outputting to said first node said first input packet and an instruction to output said first input packet from an interface of said first node in accordance with said first sorting rules by said first controller; when a second input packet is inputted from an interface of said first node device, outputting said second input packet to a virtual port of said first controller in accordance with said first sorting rules by said first controller; when a third input packet is inputted from a virtual port of said second controller, outputting to said second node said third input packet and an instruction to output said third input packet from an interface of said second node in accordance with said second sorting rules by said second controller; and when a fourth input packet is inputted from an interface of said second node device, outputting said fourth input packet to a virtual port of said second controller in accordance with said second sorting rules by said second controller.

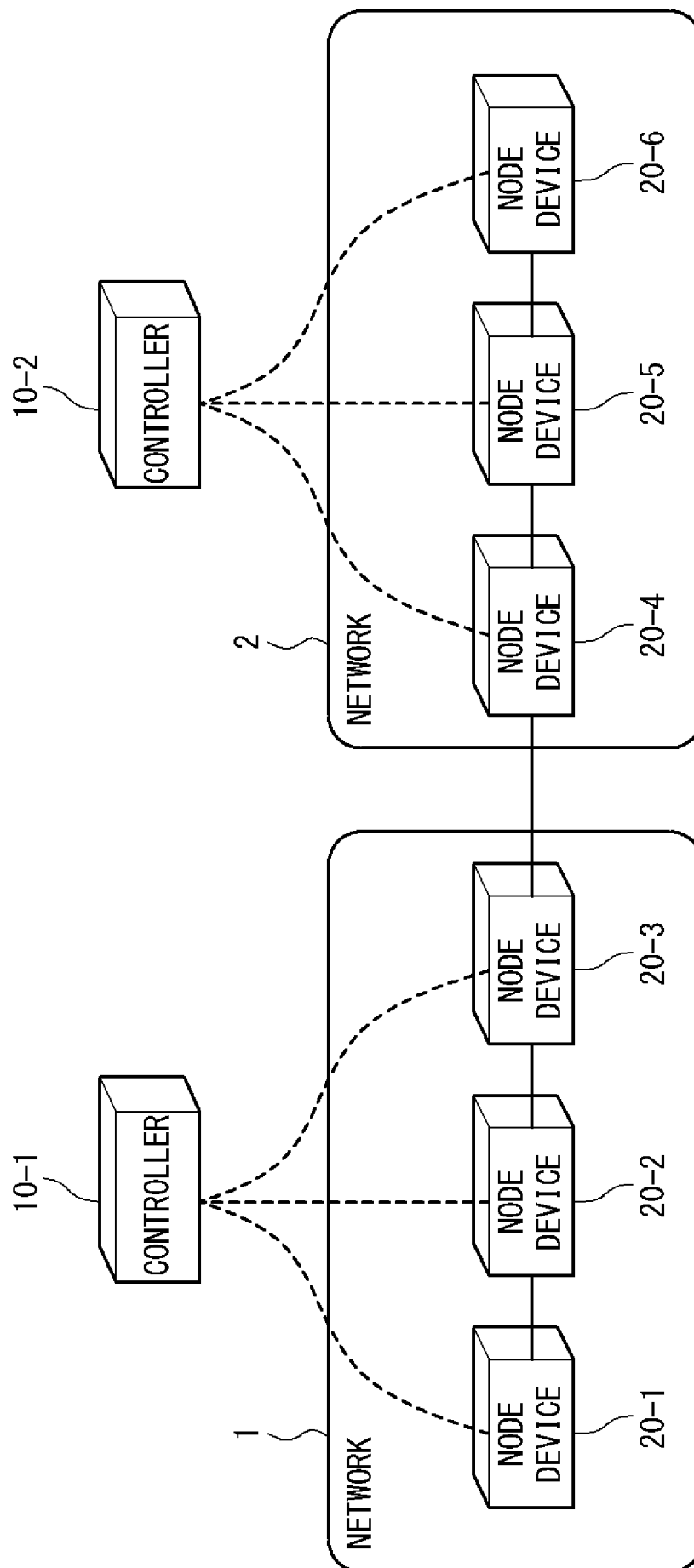
[Claim 25]

A non-transitory recording medium storing a program which when executed causes an information processing apparatus to perform an operation of one of the first and second controllers according to any one of claims 14 to 24.

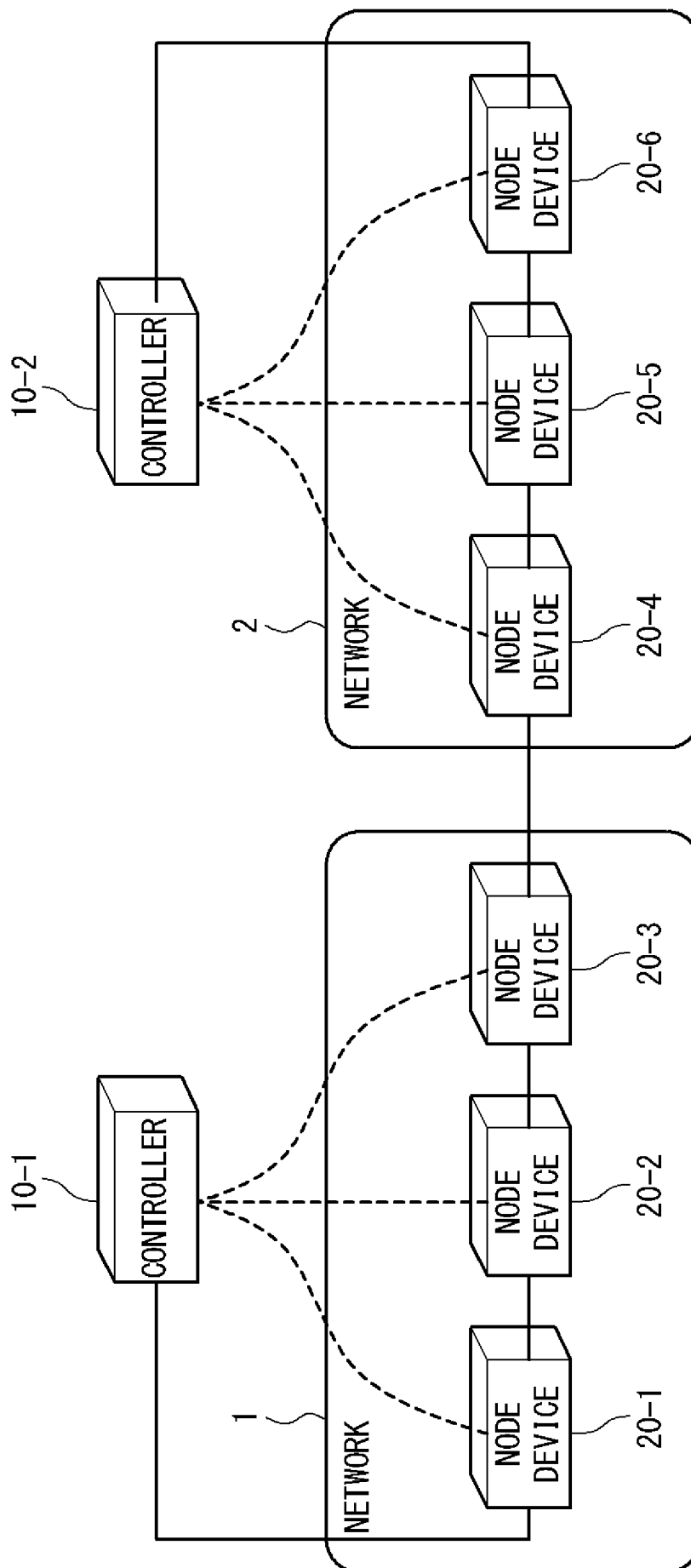
[Claim 26]

A non-transitory recording medium storing a program which when executed causes a communication apparatus to perform an operation of one of the first and second node devices according to any one of claims 14 to 24.

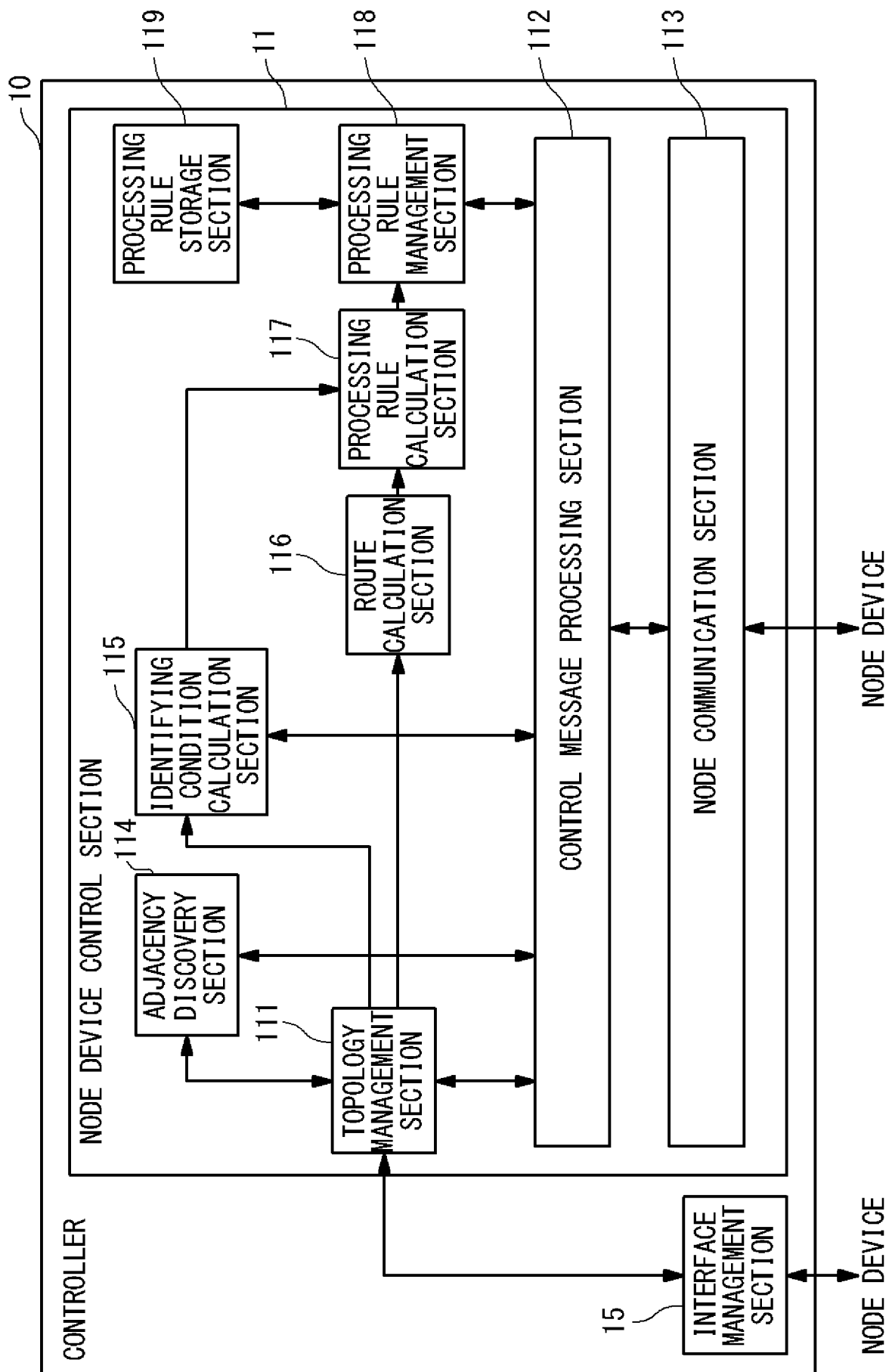
[Fig. 1]



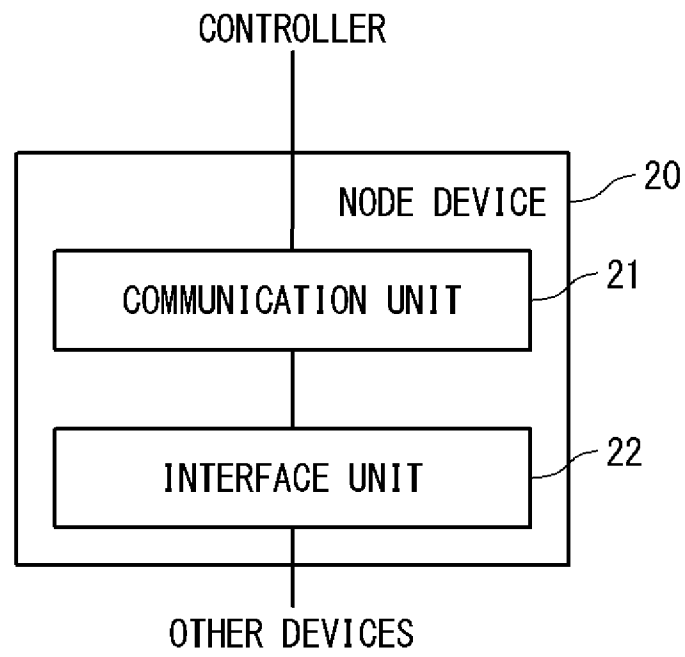
[Fig. 2]



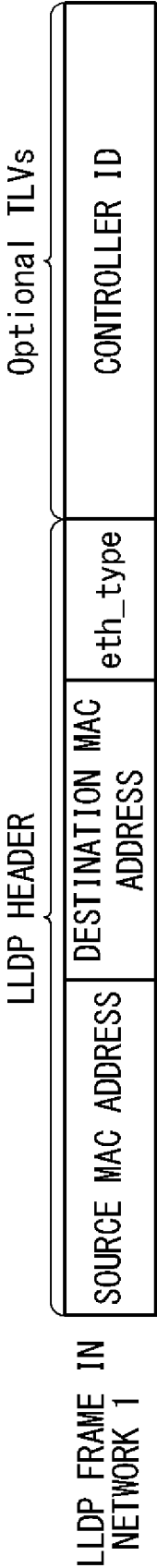
[Fig. 3]



[Fig. 4]



[Fig. 5]



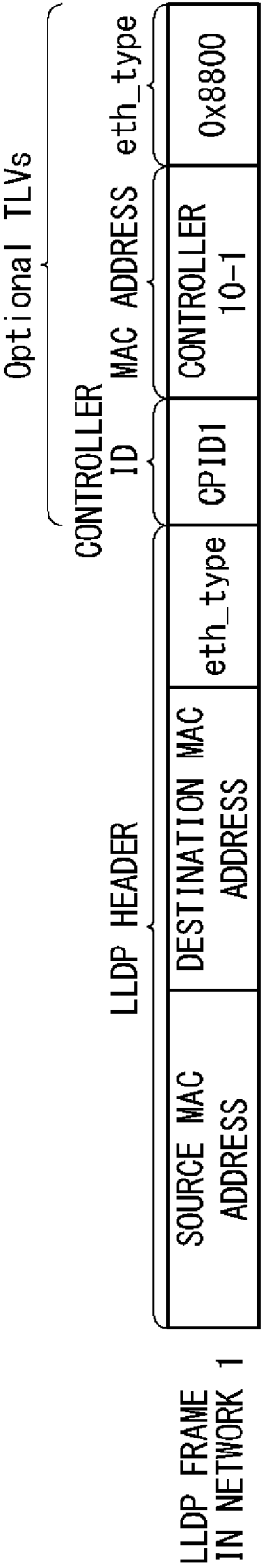
[Fig. 6A]

NETWORK 1	IDENTIFYING CONDITIONS			CONTENTS OF PROCESSING	
	SOURCE MAC ADDRESS		DESTINATION MAC ADDRESS	eth_type	ACTION
	CONTROLLER 10-2	CONTROLLER 10-1		0x8800	output to CONTROLLER 10-1
PROCESSING RULE IN NODE 20-1	(1)				
	(2)				
PROCESSING RULE IN NODE 20-2	(3)				
	(4)				
PROCESSING RULE IN NODE 20-3	(5)				
	(6)				

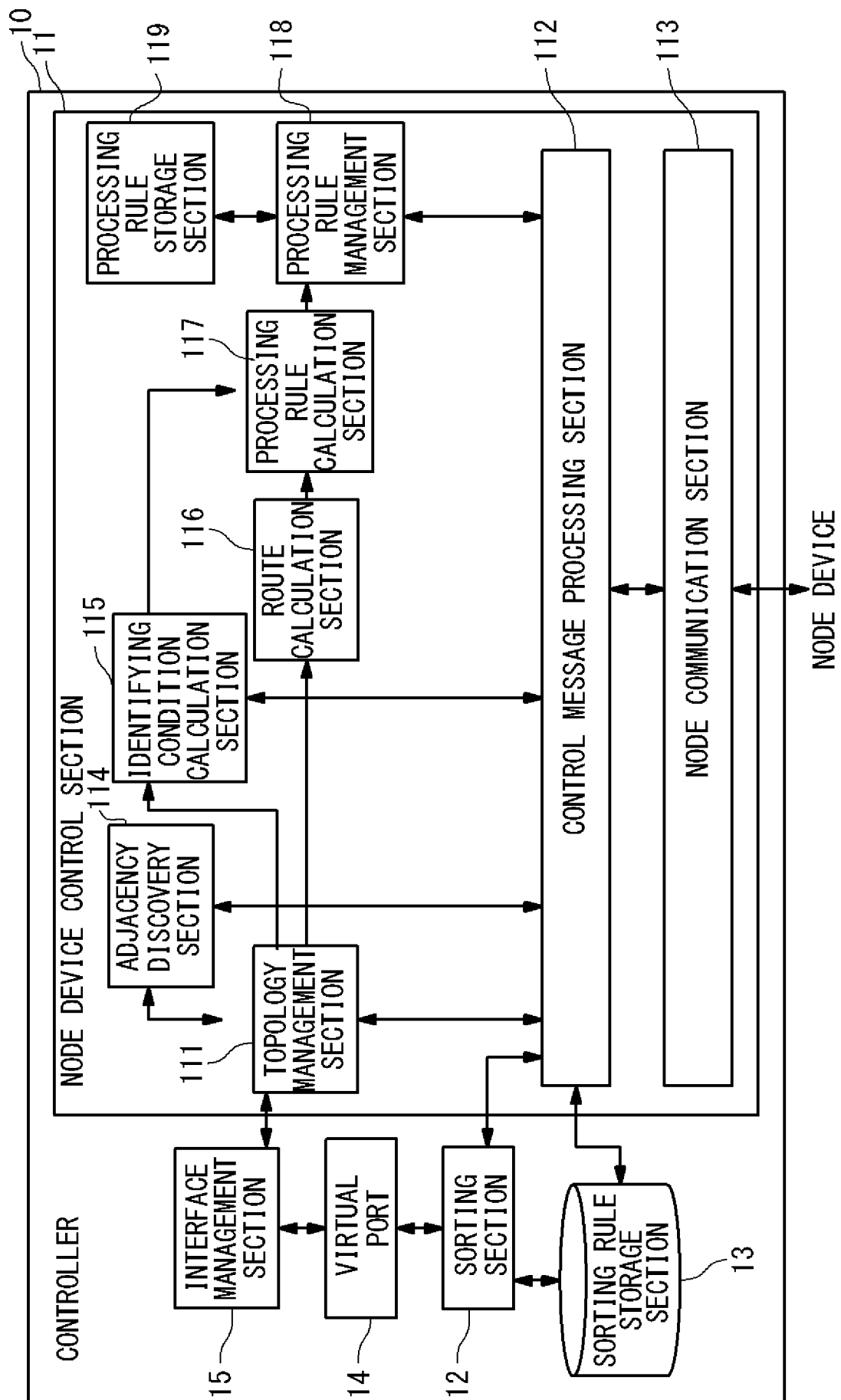
[Fig. 6B]

NETWORK 2		IDENTIFYING CONDITIONS			CONTENTS OF PROCESSING	
		SOURCE MAC ADDRESS		DESTINATION MAC ADDRESS	eth_type	ACTION
PROCESSING RULE IN NODE 20-4	(1)	CONTROLLER 10-2	CONTROLLER 10-1		0x8800	output to NODE DEVICE 20-3
	(2)	CONTROLLER 10-1	CONTROLLER 10-2		0x8800	output to NODE DEVICE 20-5
PROCESSING RULE IN NODE 20-5	(3)	CONTROLLER 10-2	CONTROLLER 10-1		0x8800	output to NODE DEVICE 20-4
	(4)	CONTROLLER 10-1	CONTROLLER 10-2		0x8800	output to NODE DEVICE 20-6
PROCESSING RULE IN NODE 20-6	(5)	CONTROLLER 10-2	CONTROLLER 10-1		0x8800	output to NODE DEVICE 20-5
	(6)	CONTROLLER 10-1	CONTROLLER 10-2		0x8800	output to CONTROLLER 10-2

[Fig. 7]



[Fig. 8]



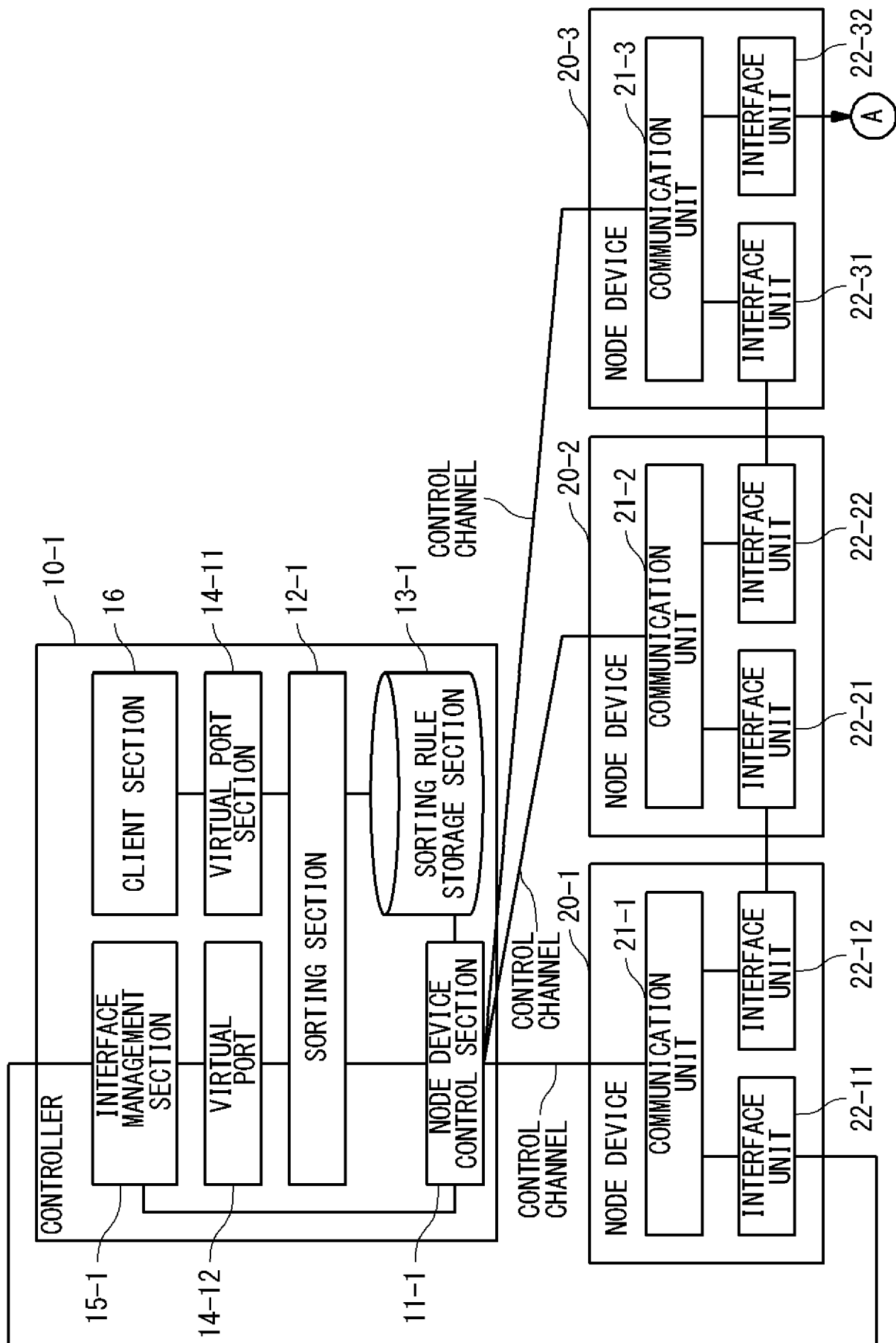
[Fig. 9A]

SORTING RULE IN CONTROLLER 10-1	VIRTUAL PORT ID	NODE DEVICE ID	INTERFACE ID	OWN ADDRESS	COUNTERPART ADDRESS	PROCESSING RULE ID
	VP1	DP1D3	1F3	CONTROLLER 10-1	CONTROLLER 10-2	A

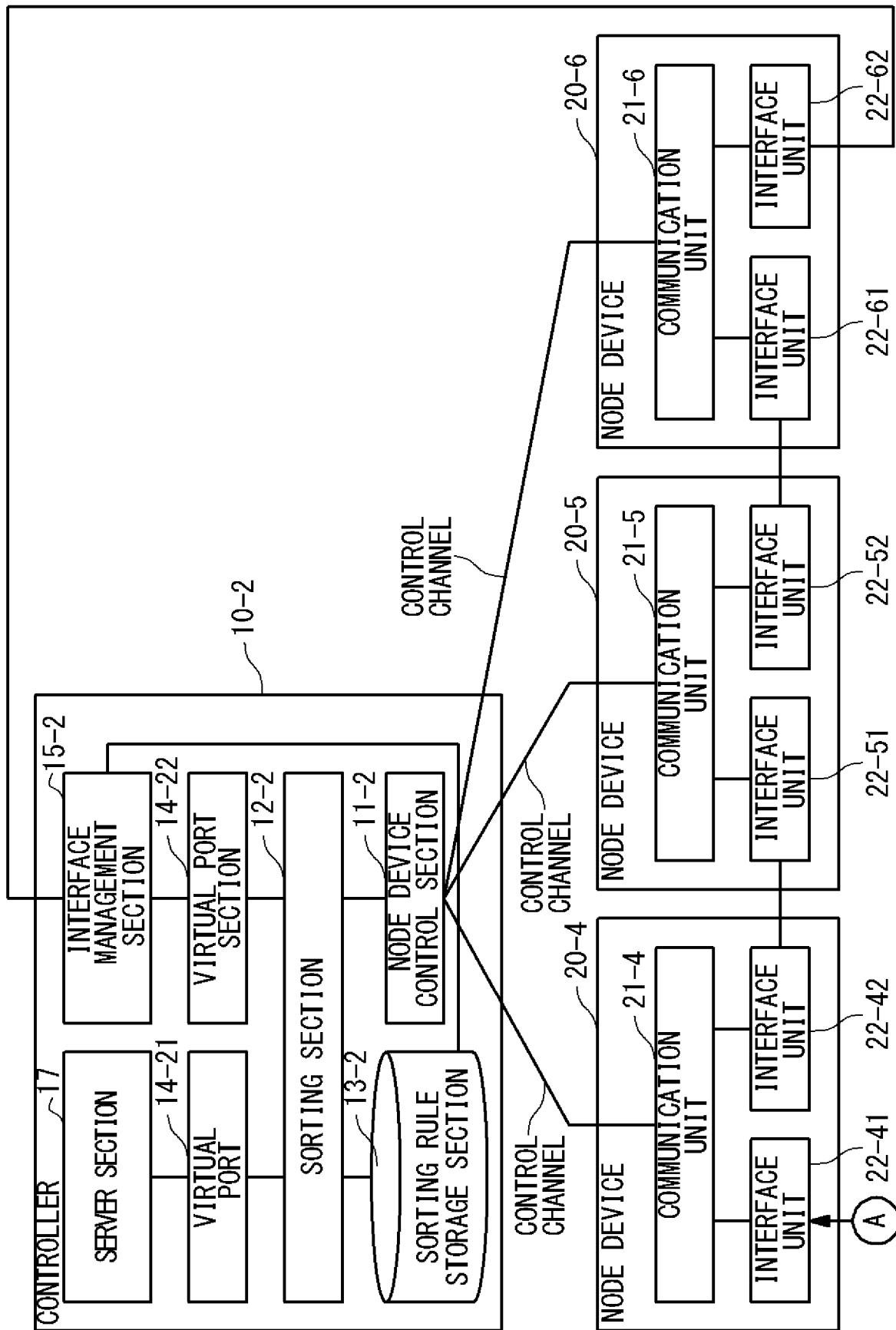
[Fig. 9B]

SORTING RULE IN CONTROLLER 10-2	VIRTUAL PORT ID	NODE DEVICE ID	INTERFACE ID	OWN ADDRESS	COUNTERPART ADDRESS	PROCESSING RULE ID
	VP2	DPID4	IF4	CONTROLLER 10-2	CONTROLLER 10-1	B

[Fig. 10A]



[Fig. 10B]



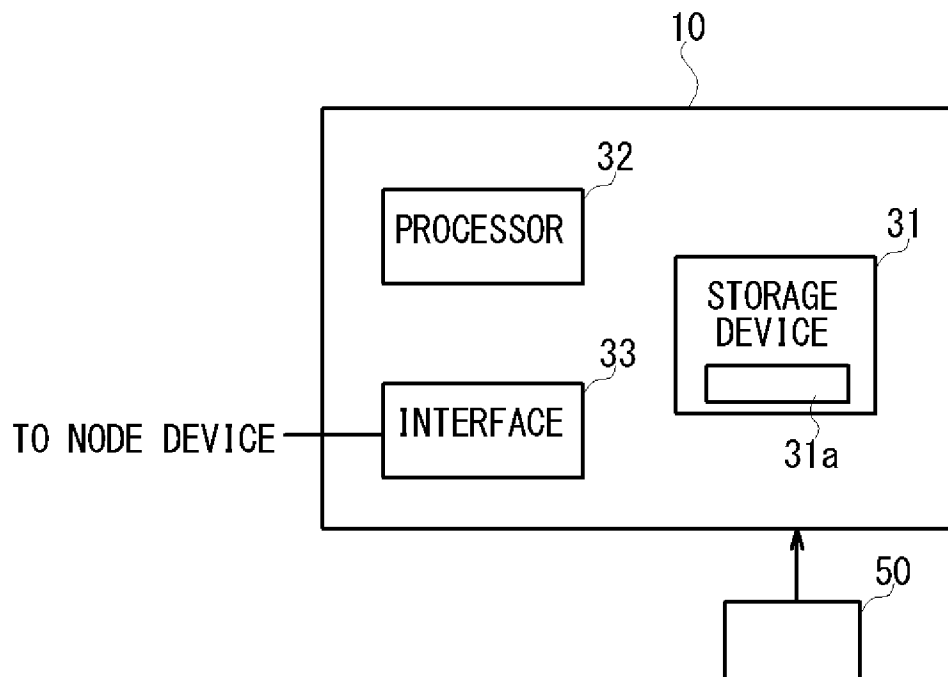
[Fig. 11A]

VIRTUAL PORT ID	NODE DEVICE ID	INTERFACE ID	OWN ADDRESS	COUNTERPART ADDRESS	PROCESSING RULE ID
VP11	DP1D3	IF32	AD11	AD21	A1
VP12	DP1D1	IF11	AD12	AD22	A2
...

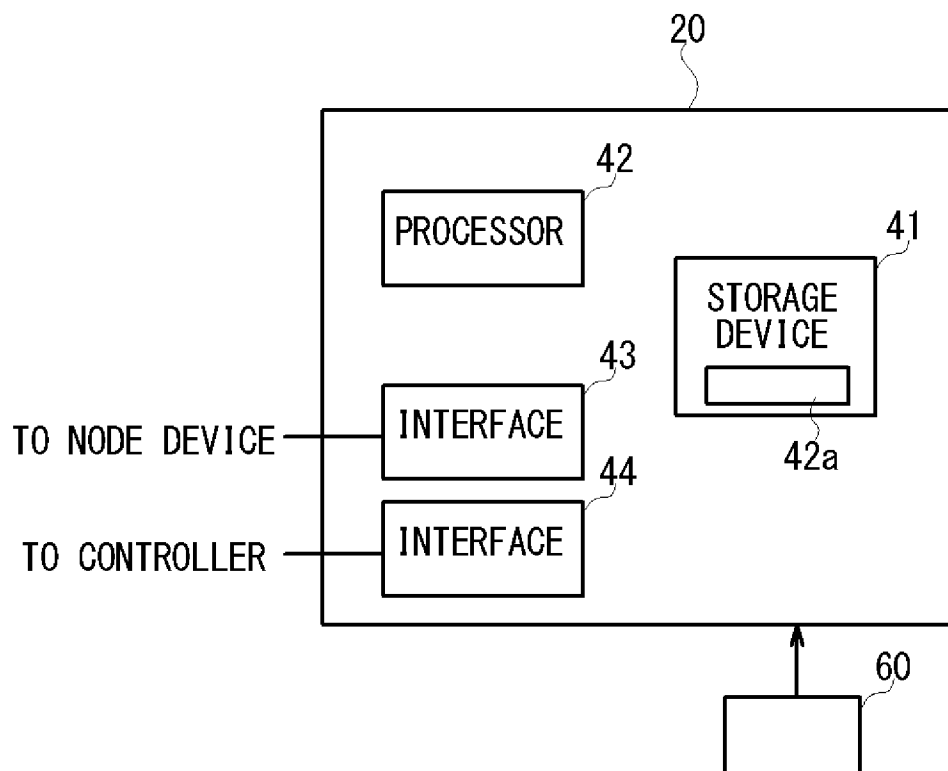
[Fig. 11B]

VIRTUAL PORT ID	NODE DEVICE ID	INTERFACE ID	OWN ADDRESS	COUNTERPART ADDRESS	PROCESSING RULE ID
VP21	DPID4	IF41	AD21	AD11	B1
VP22	DPID6	IF62	AD22	AD12	B2
...

[Fig. 12]



[Fig. 13]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2013/001913

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. H04L12/717 (2013.01) i

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)

Int.Cl. H04L12/717

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

Published examined utility model applications of Japan 1922-1996
 Published unexamined utility model applications of Japan 1971-2013
 Registered utility model specifications of Japan 1996-2013
 Published registered utility model applications of Japan 1994-2013

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

JSTPlus/JMEDPlus/JST/580 (JDreamIII)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	JP 2011-155508 A (NIPPON TELEGRAPH AND TELEPHONE CO.) 2011.08.11, 【0048】, Fig.1 & US 2012/0294617 A1 & EP 2530887 A1 & WO 2011/093396 A1 & CN 102726011 A	1, 6-9, 12-14, 19-22, 25, 26 2-5, 10, 11, 15-18, 23, 24
Y A	KOIDE, Toshio. A study on the automatic construction mechanism of control network in OpenFlow-based network, IEICE Technical Report, 2010.02.25, Vol.109 No.448, pp.19-24	1, 6-9, 12-14, 19-22, 25, 26 2-5, 10, 11, 15-18, 23, 24
Y	JP 2012-49674 A (NEC CO.) 2012.03.08, 【0003】 (Family : none)	1, 6-9, 12-14, 19-22, 25, 26



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 application or patent 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

27.05.2013

Date of mailing of the international search report

04.06.2013

Name and mailing address of the ISA/JP

Japan Patent Office

3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan

Authorized officer

KOMIYA, Shinji

Telephone No. +81-3-3581-1101 Ext. 3556

5K

9567

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2013/001913

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2012-39188 A (NIPPON TELEGRAPH AND TELEPHONE CO.) 2012.02.23, 【0003】 (Family : none)	1, 6-9, 12-14, 19-22, 25, 26