Appeal decision

Appeal No. 2018-9853

Republic of China
Appellant Chunghwa Telecom Co., Ltd.

Patent Attorney MURAI, Koji

Patent Attorney ISHIKAWA, Takayuki

The case of appeal against the examiner's decision of refusal of Japanese Patent Application No. 2016-206949, entitled "REPORT CALCULATION METHOD OF PATH STATE BASED ON CENTRALIZED CONTROL PLANE" (the application published on May 25, 2017, Japanese Unexamined Patent Application Publication No. 2017-92957, 6 Claims) has resulted in the following appeal decision.

Conclusion

The examiner's decision is revoked.

The invention of the present application shall be granted a patent.

Reason

No. 1 History of the procedures

The present application is an application filed on October 21, 2016 (priority claim under the Paris Convention: November 5, 2015, TW), and the history of the procedures is as follows.

dated August 14, 2017	Notification of reasons for refusal
November 21, 2017	Submission of Written opinion and Written amendment
dated March 13, 2018	Examiner's decision of refusal
July 19, 2018	Submission of Request for appeal and Written amendment

No. 2 Outline of the examiner's decision

The outline of the examiner's decision (the examiner's decision of refusal dated March 13, 2018) is as follows.

The inventions according to Claims 1 to 7 of the application could have been easily made before the priority date of the application by a person ordinarily skilled in the art in the Technical field to which the invention belongs based on an invention described in the following publication distributed, or inventions that were made publicly available through an electric telecommunication line, in Japan or a foreign country before the priority date of the application. Thus, the appellant should not be granted a patent for the inventions under the provisions of Article 29(2) of the Patent Act.

<List of Cited Documents, etc.>
1. International Publication No. 2015/119611
2. Mukesh Hira, et. al., Improving Network Monitoring and Management with Programmable Data Planes, [online], The P4 Language Consortium, September 25, 2015, [retrieved on 2017-08-14]. Retrieved from the Internet: <URL: http://p4.org/p4/inband-network-telemetry/>

No. 3 Regarding the amendment as of the request for appeal

It cannot be said that the amendment as of the request for appeal violates the requirements in Article 17-2(3) to 17-2(6) of the Patent Act.

The amendment to add the matter, "regarding generating a test packet by simulating an actual packet, the test packet can be distinguished from other packets having the same header by writing a dedicated keyword in a payload of the test packet basically", to Claim 1 by the amendment as of the request for appeal is to add limitation, "the test packet can be distinguished from other packets having the same header by writing a dedicated keyword in a payload of the test packet basically", regarding "generating a test packet by simulating an actual packet" in Claim 1 before the amendment, which is intended for restriction of scope of claims and described in [Claim 4] of the original specification and paragraph [0022]. Thus, the amendment is not an addition of new matter.

As indicated in "No. 4 the Invention" to "No. 6 Comparison / Judgment", the inventions according to Claims 1 to 6 after the amendment satisfy Independent requirements for patentability.

No. 4 The Invention

The inventions according to Claims 1 to 6 of the application (hereinafter referred to as "Invention 1" to "Invention 6", respectively) are inventions specified by the matters described in Claims 1 to 6 of the scope of claims amended by the written amendment as of July 19, 2018. The inventions are as follows.

"[Claim 1]

A report calculation method of path state based on a centralized control plane,

which is executed by a network management tool of a network administrator, and

includes basically:

generating, as a generation step, a test packet by simulating an actual packet in accordance with data flow information to be tested,

transmitting, as a test step, the test packet to an exchanger where a path to be tested passes, for path test,

and reporting, as a report step, an actual path where the data flow passes and the path state,

the test step including a procedure of transmitting a test packet to an exchanger located in a path start point to be tested, receiving a report immediately made by the exchanger, a judgment made by an exchanger located in a path end point to be tested about completion of report, and a judgment as to whether or not a time limit has been reached, after standby, wherein a judgment by an end-point exchanger about completion of report is basically a judgment by the end-point exchanger as to whether or not the report has been made, entering the report step when the report has been made, or entering a judgment that the time limit has been reached when the report has not been made, the procedure of judging as to whether or not the time limit has been reached is basically a judgment as to whether or not test time limit has been reached, entering the report step when the time limit has been reached, or returning to the step of receiving, after standby, the report immediately made by the exchanger when the time limit has not been reached, and

regarding generating a test packet by simulating an actual packet, the test packet can be distinguished from other packets having the same header by writing a dedicated keyword in a payload of the test packet basically.

[Claim 2]

The report calculation method of the path state based on a centralized control plane described in Claim 1, which is configured,

regarding generating a test packet by simulating an actual packet in accordance with data flow information to be tested, to secure a state where the test packet has the same routing action as a packet of a data flow to be tested by generating a header of the test packet in accordance with the data flow information to be tested.

[Claim 3]

The report calculation method of the path state based on a centralized control plane described in Claim 1 or Claim 2,

wherein the data flow information to be tested includes information to be used for identifying a packet of a source MAC, a destination MAC, a source IP, or a destination IP.

[Claim 4]

The report calculation method of the path state based on a centralized control plane described in Claim 1, configured,

regarding transmitting a test packet to an exchanger located in a path start point to be tested, to transmit a test packet generated in preparation step to an exchanger located in a start point basically.

[Claim 5]

The report calculation method of the path state based on a centralized control plane described in Claim 1, configured,

regarding receiving a report immediately made by an exchanger after standby, to receive a test packet reported by an exchanger located in an end point after standby basically.

[Claim 6]

The report calculation method of the path state based on a centralized control plane described in Claim 1, wherein the report step reports to a network administrator an actual path of a data flow aggregated in accordance with the state reported by network elements during the test step, and the path state."

No. 5 Cited documents and Cited inventions

1 Regarding Cited Document 1

In Cited Document 1 cited in the reasons for refusal of the examiner's decision, the following matters are described with drawings.

"Background

[0001] In a software defined network (SDN), applications can control the way that packets traverse a network. An SDN is a network where the data plane (the underlying systems that forward traffic) is separated from the control plane (the system that makes decisions about how traffic traverses the network). These applications can control packet flows by programming flow tables on network devices such as switches, routers, bridges, etc. Examples of such applications include load balancers, network address translation (NAT), security applications, and routing applications such as open shortest path first (OSPF), etc."

"[0017] According to a number of examples of the present disclosure, a network administrator can use the network controller 102 to generate a trace packet including a source address matching a source address of the existing flow 108 (e.g., the address 10.0.0.6 of the computing device 104-2), a destination address matching the destination address of the existing flow 108 (e.g., the address 10.0.0.5 of the computing device 104-1), and a unique identifier (UID) for the trace packet. The network controller 102 can encipher the trace packet to generate the UID so that the network controller 102 can identify it uniquely for efficient and proper analysis (e.g., as opposed to reflecting other traffic associated with the existing flow 108 to the network controller 102). Enciphering the trace packet can include converting the trace packet to a coded form (e.g., by operation of an algorithm such as a hashing algorithm, where operating the algorithm using the trace packet as an input generates the UID). The UID can be registered in the network controller 102. Registering the UID in the network controller 102 and/or embedding the UID in the trace packet that may reach the observation post 107 can facilitate unique identification of the trace packet by the observation post 107 and/or the network controller 102. The observation post 107 is described in more detail below. Registering the UID can allow the network controller 102 to encipher a received packet to generate a UID, compare it to the registered UID, and in response to the UIDs matching, identify that the received packet matches the trace packet. In some examples, the UID can be added to the trace packet (e.g., during generation of the trace packet and enciphering thereof). Trace packets can be generated for protocols that carry a data payload in some form such as protocols in the network layer or above in the Open Systems Interconnection (OSI) model. Examples of such protocols can include the transmission control protocol (TCP), user datagram protocol (UDP), dynamic host configuration protocol (DHCP), and ICMP, among others.

[0018] The network controller 102 can be used to assign an observation post 107 (e.g., network device 106-1). The observation post 107 is assigned to any of the network devices 106 on the programmed path 108. The network controller 102 can instruct the observation post 107 to send any packet that includes the matching criteria specified by the trace packet to the network controller 102 for path analysis. Once the observation post 107 has been assigned, a trace packet can be generated with the UID to test the programmed path 108. If the observation post 107 receives a packet with the specified matching criteria, it can send a copy of the packet to the network controller 102. The network controller 102 can decode the packet to get a fingerprint of the packet (e.g., the path history of the packet, including addresses and/or ports of network devices 106 through which it has traversed) and/or the UID. In response to the received fingerprint matching the matching criteria of the trace packet (e.g., matching the programmed path 108), the network controller 102 can update its record (e.g., memory structure) that it received the trace packet from the observation post 107. The network controller 102 can set a success status for the observation post 107 in response to the trace packet being received from the observation post 107. The network administrator can use the network controller 102 to query whether the trace packet was received from the observation post 107. If the status is set to success, it means that the programmed path is correct with respect to the network device 106 assigned as the observation post 107. If the status is not set to success, it means that either the trace packet is being dropped or that it is not yet received by the observation post.

[0019] The network administrator can use the network controller 102 to reassign the observation post 107 to a different network device (e.g., network device 106-2) along the programmed path 108 of the trace packet (e.g., in response to the success status being set for the previous observation post). Moving the observation post 107 can facilitate a determination of the point of failure along the programmed path 108.

[0020] Figure 2 is a diagram illustrating an example of a network 200 according to the present disclosure. The network 200 can be analogous to the network 100 illustrated in Figure 1. The network 200 can be an SDN and includes an SDN controller 202. The network 200 is illustrated including a number of computing devices 204-1, 204-2 (referred to generally herein as computing devices 204) and a number of network devices 206-1, 206-2, 206-3 (referred to generally herein as network devices 206). A first computing device 204-1 is connected to a second computing device 204-2 via a number of physical links 210 through a first network device 206-1, a second network device 206-2, and a third network device 206-3. In the example illustrated in Figure 2, the physical links 210 (e.g., expected path) are the same as the programmed path 208 (e.g., existing flow). Figure 2 also includes a number of path injector and path analysis actions 214 as described herein (e.g., implemented in the control plane of the network 200).

[0021] Figure 2 includes a flow table for each network device (e.g., a flow table 215 for the network device 206-2). The flow tables can include flow rules for existing flows (e.g., an existing flow 225 defined in a flow table 215). By way of example, the network controller 202 can assign an observation post to the second network device 206-2 to test the programmed path 208 between the first computing device 204-1 and the second computing device 204-2. In some examples, a network administrator can assign the observation post via the network controller 202. The network controller 202 can generate a trace packet 213, and register the trace packet 213 (as used herein, registering the trace packet can include registering the trace packet itself, registering a fingerprint of the trace packet, and/or registering a UID of the trace packet) to an existing flow 225 based on a source address 216-2, a destination address 218-2, and a protocol of the trace packet 213. In some examples, a network administrator can define the trace packet via the network controller 202. As illustrated, the existing flow 225 defined on the second network device 206-2 includes a source address 216-1 and a destination address 218-1 that match the source address 216-2 and the destination address 218-2 of the trace packet 213. In some examples, the incoming port 220-2 of a particular network device of the trace packet 213 can match an incoming port 220-1 of the existing flow 225. Accordingly, a flow rule 226 can be added to the existing flow 225 on the second network device 206-2, where the flow rule 226 includes the source address 216-2, the destination address 218-2, an incoming port 220-2, a priority 222-2, and an outgoing port / action 224-2 for the trace packet 213.

[0022] The flow rule 226 can be added for the trace packet 213 with a higher priority 222-2 (e.g., 1001) than a priority 222-1 (e.g., 1000) of the existing flow 225 corresponding to the trace packet 213 on the second network device 206-2. The higher priority for the flow rule 226 for the trace packet 213 can cause the second network device 206-2 to take the defined action 224-2 for the existing flow rule 226 before and/or instead of taking the defined action for the existing flow 225. In the example illustrated in Figure 2, this means that the corresponding packet (trace packet 213) would be sent out of the "controller port" (e.g., be sent to the network controller 202).

[0023] Although not specifically illustrated in the flow rule 226 in Figure 2, the flow rule 226 can specify a more specific flow based on flow table capability in order to send to the network controller 202 only one class of traffic consisting of the trace packet (e.g., to reduce the volume of traffic being sent to the controller 202).

[0024] Although not specifically illustrated in the flow rule 226 in Figure 2, the flow rule 226 can also specify a protocol for the trace packet 213. Thus, the trace packet 213 can be application specific. Specifying a protocol can prevent the second network device 206-2 from sending to the network controller 202 any packet (e.g., other traffic that is associated with a flow that is not experiencing an error because the flow for that protocol is working properly) that matched the source address 216-2 and destination address 218-2 of the trace packet 213.

[0025] The network controller 202 can inject the trace packet 213 into the network 200 on behalf of a source of the existing path (e.g., computing device 204-1) via a specified port of a network device (e.g., port 2 of network device 206-1) along a programmed path 208 of the existing flow. As illustrated, the network controller 202 can inject the trace packet 213 via a port of a network device 206-1 that is upstream, with respect to the programmed path 208, of the network device 206-2 that is assigned an observation post and downstream of the computing device 204-1 that is the source of the flow.

[0026] If the programmed path 208 is programmed correctly on the first network device 206-1 where the trace packet 213 is injected, then the first network device 206-1 should correctly forward the trace packet 213 to the second network device 206-2 (e.g., out of port 3 of the first network device 206- 1 and into port 4 of the second network device 206-2). Because the second network device 206-2 has been assigned the observation post with the accompanying flow rule 226, the second network device 206-2 will send the trace packet 213 to the network controller 202. In some examples, the network controller 202 sets a success status for the second network device 206-2 (observation post) merely because the trace packet 213 was received therefrom. In some examples, the network controller 202 can first examine a fingerprint of the trace packet received from the second network device 206-2 and compare it to expected matching criteria for the trace packet 213 based on the programmed path 208. In this example, the fingerprint and/or matching criteria could include information such as one or more of the following: the source address of the computing device 204-1 (00:1 E:08:AE:D3:BE), the outgoing port (1) of the computing device 204-1, the incoming port (2) of the first network device 206-1, the outgoing port (3) of the first network device 206-1, the incoming port (4) of the second network device 206-2, and/or the destination address (00:0c:29:03:7f:48) of the second computing device 204-2. If the fingerprint of the trace packet received from the second network device 206-2 matches the expected matching criteria for the trace packet 213, then the network controller 202 sets the success status for the second network device 206-2.

[0027] After the network controller 202 sets the success status for the second network device 206-2, a network administrator can query a status for the second network device 206-2 (e.g., via the network controller 202). If the status is set to success, then the network administrator may wish to continue testing the programmed path 208. The network controller 202 injects the trace packet 213 back into the network 200 via a defined network device (e.g., via the second network device 206-2) and instructs a different observation post (e.g., a different network device 206-3 downstream of the previous observation post with respect to the existing flow) to send any packet that includes the UID to the network controller 202. This testing process can continue until the test packet 213 is not returned to the network controller 202, which would indicate that the observation post that is assigned when the test packet 213 is not returned to the network controller 202 is a point of failure."

According to the above description, it is acknowledged that Cited Document 1 describes the following invention (hereinafter referred to as "Cited Invention").

"A method of a network with a control plane implemented thereon including

using a network controller to generate a trace packet including a source address matching a source address of an existing flow, a destination address matching a destination address of the existing flow, and a unique identifier (UID) for the trace packet, wherein

the network controller injects the trace packet into a network on behalf of a source of the existing path via a specified port of a first network device along a programmed path of the existing path,

if the programmed path is programmed correctly on the first network device, the first network device correctly forwards the trace packet to a second network device,

the second network device which has been assigned an observation post with a path rule sends the trace packet to the network controller,

the network controller first examines a fingerprint including a source address of a first computing device of the trace packet received from the second network device, an outgoing port of the first computing device, an incoming port of the first network device, an outgoing port of the first network device, and an incoming port of the second network device, and/or a destination address of the second computing device, and sets a success status for the second network device when the fingerprint matches an expected matching criteria,

after the network controller sets the success status for the second network device, a network administrator can query a status for the second network device via the network controller,

if the status is set to success, then the network administrator may wish to continue testing the programmed path, the network controller injects the trace packet back into the network via the second network device and instructs a different network device located downstream of the previous observation post with respect to the existing path to send any packet that includes the UID to the network controller."

2 Regarding Cited Document 2

In Cited Document 2 cited in the reasons for refusal of the examiner's decision, the following matters are described with drawings.

"An illustrative example of INT is described here. Source vSwitch (vSwitch 1) embeds instructions for each network element to report the latency that the packet encounters at the network element (delta between local egress timestamp and local ingress timestamp). The receiving vSwitch (vSwitch2) can compute the end-to-end latency as a sum of the per-hop latencies (Under the assumption that switching and queueing latencies dominate and propagation delays are minimal, which is typically true in today's networks). Per-hop latencies in the packet received at the destination vSwitch can also be used to determine which network element(s) contributed most to the end-to-end latency."

According to the above description, Cited Document 2 describes the following matters (hereinafter referred to as "Matters described in Cited Document 2").

"The receiving vSwitch (vSwitch2) can compute the end-to-end latency as a sum of the per-hop latencies, and per-hop latencies in the packet received at the destination vSwitch can also be used to determine which network element(s) contributed most to the end-to-end latency."

3 Regarding Cited Document 3

On pp. 145-150 of "SDN Traceroute: Tracing SDN Forwarding without Changing Network Behavior, ACM SIGCOMM'14" (hereinafter referred to as "Cited Document 3"), which is a publication distributed or made publicly available through an electric telecommunication line before the priority date of the application, written by Kanak Agarwl and three others, published on August 22, 2014, the following matters are described with drawings.

"3.4 Conducting the Trace Route

Once the network is configured in the manner discussed above, it is ready to accept Ethernet probe frames for route tracing. The process is best explained via an example, shown in Figure 2.

SDN traceroute begins by identifying the injection point. This is either identified in the API call or it is assumed to be the attachment point of the source host, which is looked up by source MAC or IP address. Once SDN traceroute has the injection switch identifier and port, it looks up the color of the ingress switch and inserts the color into the header tag bits of the probe frame.

SDN traceroute then sends the probe to the ingress switch as a PACKET_OUT message with the input port set to the injection point. The action for the PACKET_OUT is set to TABLE, indicating that the switch should treat the packet as though it had been received on the input port (step 1).

On receiving the PACKET_OUT, the ingress switch processes the packet in its flow table. Since the header tag bits in the packet are set to the color of the switch itself, the packet does not encounter a match on any of the high-priority rules SDN traceroute has installed. Consequently, the packet is forwarded to the next hop as though it were a regular, default-tagged packet (step 2). This ensures that the actual forwarding rules in the switch are used to route the packet even though it is a probe and not production traffic.

The packet arrives at the second switch while still carrying the header tag bits set to the color of the first switch. Since each switch is configured with high-priority rules that trap all packets matching the neighboring switches' colors, the packet at the second switch results in a match and is sent to the controller as a PACKET_IN (step 3). SDN traceroute receives the packet at the controller and logs the switch-id and port information of the switch that forwarded the packet to the controller as the next hop in the path.

Once SDN traceroute records the current hop, it modifies the received probe frame by rewriting the reserved tag field to the bits corresponding to the color of the current switch. It then sends the modified probe back as a PACKET_ OUT to the same switch that had sent the PACKET_IN message. The input port in the PACKET_OUT is set to the input port where the packet was received at the switch. The action field is once again set to TABLE (step 4). The switch receives the modified probe from the controller and applies its flow-table action on the probe. Since the reserved tag bits in the modified probe are set to the color of the switch, the tag based rules do not match and the packet is forwarded along the next hop as a regular frame (step 5).

This process (steps 3-5) repeats for each hop in the path. The process terminates when a time-out occurs between consecutive PACKET_IN events, indicating that the packet has left the network or been consumed by a host, or when a predetermined <switch-id, port>is repeated in the route, indicating the presence of a loop.

Lastly, note that in step 3 the trace route application only handles probe PACKET_IN messages that do not match the color of the switch sending the PACKET_IN. This allows PACKET_IN messages matching the input switch color to be forwarded to other modules in the controller for processing. This allows for scenarios where regular packet processing at a switch may itself initiate a PACKET_IN to the controller, such as in reactive rule installation."

4 Regarding Cited Document 4

In Japanese Unexamined Patent Application Publication No. 2006-319973 (hereinafter referred to as "Cited Document 4"), which is a publication distributed or made publicly available through an electric telecommunication line before the priority date of the application, the following matters are described with drawings.

"[Technical field]
[0001]

This invention relates to protocol-generic network eavesdropping."

"[Means for solving the problem]
[0008]

Embodiments of the present invention provide a protocol-generic (or "protocol-unaware") eavesdropping device that is capable of monitoring traffic communicated over a communication network and identifying packets within such traffic that are of interest to the eavesdropping device. As described further herein, techniques are provided that enable packet identification and packet authentication. As described further herein, in certain embodiments, such "packet authentication" does not authenticate the entire packet, but instead authenticates the ID and content that is intended for the eavesdropping device. Thus, an eavesdropping device can identify packets that are of interest and can authenticate the identifying information and the information that is of interest to the eavesdropping device. Embodiments of the present invention are protocol-generic and thus do not require that the eavesdropping device have a priori knowledge about the communication protocol being used in order for the eavesdropping device to be able to identify packets that are of interest and authenticate such packets. Thus, the protocol-generic eavesdropping device can be employed and dynamically adapt to any communication protocol that may be utilized on the communication network without requiring any modification to the eavesdropping device.

[0009]

As described further below, rather than including information in a packet in a protocol-specific way (such as in the header of packets) for identifying the packet as one that is of interest to the eavesdropping device, embodiments of the present invention include an identifier in a protocol-generic manner. For instance, an identifier may be included in the payload of the packet. In certain embodiments, the eavesdropping device can scan the payload of a captured packet for an identifier, and upon recognizing the identifier may use techniques for authenticating the packet.

[0010]

According to one embodiment, a method comprises capturing, by an eavesdropping device, a packet communicated over a communication network. The eavesdropping device scans the packet's payload, and determines if an identifier is included in the packet's payload that identifies the packet as containing content of interest to the eavesdropping device. Based at least in part on determining that the packet's payload includes such identifier, the eavesdropping device uses the content of interest contained in the packet's payload.

[0011]

According to one embodiment, a method comprises forming a packet containing content intended for an eavesdropping device, wherein the packet includes a header portion and a payload portion. The payload portion comprises a) a predefined identifier that identifies the packet as containing the content intended for the eavesdropping device, b) the content intended for the eavesdropping device, and c) an authentication token for authenticating the predefined identifier and the content intended for the eavesdropping device. The method further comprises directing the packet via a communication network to a destination. In certain embodiments, the destination to which the packet is directed is a destination other than the eavesdropping device, wherein the eavesdropping device intercepts such packet and recognizes it as containing the content intended for the eavesdropping device.

[0012]

The foregoing has outlined rather broadly the features of the present invention in order that the detailed description of the invention that follows may be better understood. Additional features and advantages of the invention will be described hereinafter which form the subject of the claims of the invention. It should be appreciated that the conception and specific embodiment disclosed may be readily utilized as a basis for modifying or designing other structures for carrying out the same purposes of the present invention. It should also be realized that such equivalent constructions do not depart from the invention as set forth in the appended claims. The novel features which are believed to be characteristic of the invention, both as to its organization and method of operation, together with further objects and advantages, will be better understood from the following description when considered in connection with the accompanying figures. It is to be expressly understood, however, that each of the figures is provided for the purpose of illustration and description only and is not intended as a definition of the limits of the present invention."

No. 6 Comparison / Judgment

1 Regarding Invention 1

(1) Comparison

The following matters are found by comparing Invention 1 and the Cited Invention.

A The method of the Cited Invention is configured so that "a network administrator can query a status for the second network device via the network controller" and "if the status is set to success, then the network administrator may wish to continue testing the programmed path, the network controller injects the trace packet back into the network via the second network device and instructs a different network device located downstream of the previous observation post with respect to the existing path to send any packet that includes the UID to the network controller", which means a tool for responding to a query or the like via a network controller.

Thus, the tool of the Cited Invention corresponds to "a network management tool of a network administrator" in Invention 1.

B The Cited Invention is, as examined in A, a tool for responding to a query or the like via a network controller, and responds to a query of a network administrator about network path. It is obvious that a certain calculation is executed for the response, and the target network includes a control plane implemented thereon.

Therefore, the Cited Invention and the "report calculation method of the path state based on a centralized control plane" of Invention 1 are identical in point of "a report calculation method of the path state based on a control plane".

C The description in the Cited Invention, "using a network controller to generate a trace packet including a source address matching a source address of an existing flow, a destination address matching a destination address of the existing flow, and a unique identifier (UID) for the trace packet", corresponds to the description in Invention 1, "generating, as generation step, a test packet by simulating an actual packet in accordance with data flow information to be tested".

D The description in the Cited Invention, "the network controller injects the trace packet into a network on behalf of a source of the existing path via a specified port of a first network device along a programmed path of the existing path", corresponds to the description in Invention 1, "transmitting, as test step, the test packet to an exchanger where a path to be tested passes, for path test". As examined in A, the Cited Invention, which manages a path state of the trace packet by a status of the network device, is acknowledged to conduct a test corresponding to the "path test" in Invention 1.

In addition, the description, "the network controller injects the trace packet into a network on behalf of a source of the existing path via a specified port of a first network device along a programmed path of the existing path", corresponds to the description in Invention 1, "the test step including a procedure of transmitting a test packet to an exchanger located in a path start point to be tested".

E As examined in A, since the Cited Invention manages path state of the trace packet by a status of the network device and the network administrator can query the status via the network controller, the response to the query and the description in Invention 1, "reporting, as a report step, an actual path where the data flow passes and the path state" are identical in point of "reporting, as a report step, an actual path where the data flow passes".

F Accordingly, Invention 1 and the Cited Invention are identical and different in the following points.

[Corresponding Feature]

"A report calculation method of the path state based on a control plane,

which is executed by a network management tool of a network administrator, and

includes basically:

generating, as a generation step, a test packet by simulating an actual packet in accordance with data flow information to be tested,

transmitting, as a test step, the test packet to an exchanger where a path to be tested passes, for path test,

and reporting, as a report step, an actual path where the data flow passes, the test step including a procedure of transmitting a test packet to an exchanger located in a path start point to be tested."

[Different Feature 1]

Regarding the "report step", Invention 1 "reports, as a report step, an actual path where the data flow passes and the path state", while the Cited Invention does not report the path state of the path where the data flow passes.

[Different Feature 2]

Regarding the "test step", in Invention 1, "the test step including a procedure of transmitting a test packet to an exchanger located in a path start point to be tested, receiving a report immediately made by the exchanger, a judgment made by an exchanger located in a path end point to be tested about completion of report, and a judgment as to whether or not a time limit has been reached, after standby, wherein a judgment by an end-point exchanger about completion of report is basically a judgment by the end-point exchanger as to whether or not the report has been made, entering the report step when the report has been made, or entering a judgment that the time limit has been reached when the report has not been made, the procedure of judging as to whether or not the time limit has been reached is basically a judgment as to whether or not a test time limit has been reached, entering the report step when the time limit has been reached, or returning to the step of receiving, after standby, the report immediately made by the exchanger when the time limit has not been reached", while the Cited Invention includes "transmitting a test packet to an exchanger located in a path start point to be tested", but does not execute the "procedure of receiving a report immediately made by the exchanger, a judgment made by an exchanger located in a path end point to be tested about completion of report, and a judgment as to whether or not a time limit has been reached, after standby, wherein a judgment by an end-point exchanger about completion of report is basically a judgment by the end-point exchanger as to whether or not the report has been made, entering the report step when the report has been made, or entering a judgment that the time limit has been reached when the report has not been made, the procedure of judging as to whether or not the time limit has been reached is basically a judgment as to whether or not test time limit has been reached, entering the report step when the time limit has been reached, or returning to the step of receiving, after standby, the report immediately made by the exchanger when the time limit has not been reached".

[Different Feature 3]

Invention 1 describes that "regarding generating a test packet by simulating an actual packet, the test packet can be distinguished from other packets having the same header by writing a dedicated keyword in a payload of the test packet basically". In the Cited Invention, the location of "a unique identifier (UID) for the trace packet" included in a packet is unclear.

[Different Feature 4]

The "control plane" in Invention 1 is "a centralized control plane", while it is unclear whether the "control plane" in the Cited Invention is a "centralized" one or not.

(2) Judgment on the different features

The "Different Feature 1" is examined below.

Cited Invention 1 does not include any description or indication about reporting a state corresponding to the "path state" of Invention 1.

It cannot be said that it is obvious to report the "path state" in Cited Invention 1.

The Matters described in Cited Document 2 describe reporting "per-hop latency" corresponding to the "path state" in Invention 1. The Cited Invention includes the following matters:

A after "the first network device correctly forwards the trace packet to a second network device",

B "the second network device which has been assigned an observation post with a path rule sends the trace packet to the network controller",

C "the network controller first examines a fingerprint including a source address of a first computing device of the trace packet received from the second network device, an outgoing port of the first computing device, an incoming port of the first network device, an outgoing port of the first network device, and an incoming port of the second network device, and/or a destination address of the second computing device, and sets a success status for the second network device when the fingerprint matches expected matching criteria", and

D after "a network administrator" "queries a status for the second network device via the network controller", "if "the network administrator" "may wish to continue testing the programmed path", "the network controller injects the trace packet back into the network via the second network device and instructs a different network device located downstream of the previous observation post with respect to the existing path to send any packet that includes the UID to the network controller".

The second network device transmits a trace packet to the network controller after receiving the trace packet before injecting the trace packet back into the network, and, after a status is set, the network controller injects the trace packet back into the network via the second network device.

Accordingly, since the Cited Invention cannot measure the "per-hop latency" in the Matters described in Cited Document 2, it can be said that there is a disincentive to employ the Matters described in Cited Document 2 in the Cited Invention.

Cited Documents 3 and 4 do not include any description or indication about reporting a state corresponding to the "path state" in Invention 1.

Therefore, it cannot be said that Invention 1 could have been easily made by a person skilled in the art based on the Cited Invention and the matters described in Cited Documents 2 to 4, without examining other different features.

2 Regarding Inventions 2 to 6

Inventions 2 to 6 also include "reporting, as a report step, an actual path where the data flow passes and the path state". For the same reasons as those of Invention 1, it cannot be said that Inventions 2 to 6 could have been easily made by a person skilled in the art based on the Cited Invention and the matters described in Cited Documents 2 to 4.

No. 7 Regarding the examiner's decision

It cannot be said that Inventions 1 to 6, which include "reporting, as a report step, an actual path where the data flow passes and the path state", could have been easily made by a person skilled in the art on the basis of Cited Documents 1 and 2 cited in the examiner's decision of refusal. Therefore, the examiner's decision cannot be maintained.

No. 8 Closing

As described above, the application cannot be rejected due to the reasons of the examiner's decision.

No other reasons for refusing the application were found.

Therefore, the appeal decision shall be made as described in the conclusion.

August 19, 2019

Chief administrative judge:	YOSHIDA, Koichi
Administrative judge:	ODA, Hiroshi
Administrative judge:	KAJIO, Seiya