審決

不服２０１８－９８５３

（省略）
請求人　中華電信股▲分▼有限公司

（省略）
代理人弁理士　村井 康司

（省略）
代理人弁理士　石川 貴之

特願２０１６－２０６９４９「集中式コントロールプレーンに基づくパス状態の報告演算方法」拒絶査定不服審判事件〔平成２９年５月２５日出願公開、特開２０１７－９２９５７、請求項の数（６）〕について、次のとおり審決する。

結論
原査定を取り消す。
本願の発明は、特許すべきものとする。

理由
第１　手続の経緯
本願は，平成２８年１０月２１日（パリ条約による優先権主張２０１５年１１月５日，台湾）の出願であって，その手続の経緯は以下のとおりである。
平成２９年　８月１４日付け　　拒絶理由通知書
平成２９年１１月２１日　　意見書，手続補正書の提出
平成３０年　３月１３日付け　　拒絶査定
平成３０年　７月１９日　　審判請求書，手続補正書の提出

第２　原査定の概要
原査定（平成３０年３月１３日付け拒絶査定）の概要は以下のとおりである。

この出願の請求項１ないし７に係る発明は，その優先権主張日前に日本国内又は外国において，頒布された下記の刊行物に記載された発明又は電気通信回線を通じて公衆に利用可能となった発明に基いて，その優先権主張日前にその発明の属する技術の分野における通常の知識を有する者が容易に発明をすることができたものであるから，特許法第２９条第２項の規定により特許を受けることができない。

＜引用文献等一覧＞

１.国際公開第２０１５／１１９６１１号
２.Mukesh Hira, et. al.，Improving Network Monitoring and Management with Programmable Data Planes，[online]，The P4 Language Consortium，２０１５年９月２５日，[retrived on 2017-08-14]. Retrived from the Internet: <URL: http://p4.org/p4/inband-network-telemetry/>

第３　審判請求時の補正について
　審判請求時の補正は，特許法第１７条の２第３項から第６項までの要件に違反しているものとはいえない。
　審判請求時の補正によって請求項１に「実際のパケットをシミュレーションしたテストパケットを生成することについて、主にテストパケットの payload に専用のキーワードを記入することで、テストパケットが同じ header を有する他のパケットと区別できるものである」という事項を追加する補正は，補正前の請求項１の「実際のパケットをシミュレーションしたテストパケットを生成」することについて，「主にテストパケットの payload に専用のキーワードを記入することで、テストパケットが同じ header を有する他のパケットと区別できるものである」との限定を加えるものであるから，特許請求の範囲の減縮を目的とするものであり，当初明細書の【請求項４】や段落【００２２】に記載されているから，当該補正は新規事項を追加するものではない。
　そして，「第４　本願発明」から「第６　対比・判断」までに示すように，補正後の請求項１ないし６に係る発明は，独立特許要件を満たすものである。

第４　本願発明
　本願請求項１ないし６に係る発明（以下，それぞれ「本願発明１」ないし「本願発明６」という。）は，平成３０年７月１９日付けの手続補正書で補正された特許請求の範囲の請求項１ないし６に記載された事項により特定される発明であり，以下のとおりの発明である。
「【請求項１】
　ネットワーク管理者のネットワーク管理ツールによって実行され、
　生成段階として、テストを行いたいデータフロー情報に従い実際のパケットをシミュレーションしたテストパケットを生成し、
　テスト段階として、テストパケットをテストを行いたいパスが通過する交換機に送信し、パステストを行い、
　報告段階として、データフローが通過する実際のパスとパス状態を報告する、ことを主に含み、
　テスト段階が、テストパケットをテストを行いたいパス始点に位置する交換機に送信し、交換機が直ちに行う報告、テストを行いたいパス終点に位置する交換機による報告済かの判断と時間制限に到達済かの判断を待機後に受信し、終点交換機による報告済かの判断が主に終点交換機による報告済か否かの判断であり、報告済ならば報告段階に入り、未報告ならば時間制限に到達済の判断に入る手順であり、時間制限に到達済かの判断の手順が主にテストの時間制限

が到達済か否かの判断であり、到達済ならば報告段階に入り、未到達ならば交換機が直ちに報告するのを待機後に受信するに戻る手順を含み、
　実際のパケットをシミュレーションしたテストパケットを生成することについて、主にテストパケットの payload に専用のキーワードを記入することで、テストパケットが同じ header を有する他のパケットと区別できるものである、集中式コントロールプレーンに基づくパス状態の報告演算方法。
【請求項２】
　テストを行いたいデータフロー情報に従い実際のパケットをシミュレーションしたテストパケットを生成することについて、テストを行いたいデータフロー情報に従い、テストパケットの header を生成することで、テストパケットがテストを行いたいデータフローのパケットと同様のルーティングアクションを備える状態を確保するものである、
請求項 1 に記載の集中式コントロールプレーンに基づくパス状態の報告演算方法。
【請求項３】
　テストを行いたいデータフロー情報が送信元 MAC、宛先 MAC、送信元 IP 又は宛先 IP のパケットの識別用に使える情報を含む、
請求項 1 又は 2 に記載の集中式コントロールプレーンに基づくパス状態の報告演算方法。
【請求項４】
　テストパケットをテストを行いたいパス始点に位置する交換機に送信することについて、主に始点の交換機に準備段階で生成したテストパケットを送信する、
請求項 1 に記載の集中式コントロールプレーンに基づくパス状態の報告演算方法。
【請求項５】
　交換機が直ちに報告するのを待機後に受信することについて、主に終点の交換機が報告するテストパケットを待機後に受信するものである、
請求項 1 に記載の集中式コントロールプレーンに基づくパス状態の報告演算方法。
【請求項６】
　報告段階が、ネットワーク管理者にテスト段階時にネットワークエレメントが報告した状況に従い集合させたデータフローの実際のパスとパス状態を報告するものである、請求項 1 に記載の集中式コントロールプレーンに基づくパス状態の報告演算方法。」

第５　引用文献，引用発明等
１　引用文献１について
　原査定の拒絶の理由された引用文献１には，図面とともに次の事項が記載されている。

「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 can include load balancers, network address translation (NAT), security applications, and routing applications such as open shortest path first (OSPF), etc. 」

（当審訳：背景技術

［０００１］　ソフトウェア定義ネットワーク(SDN)では，アプリケーションは，パケットがネットワークを通過する方法を制御することができる。SDN は，データプレーン（トラフィックを転送する基盤となるシステム）がコントロールプレーン（トラフィックがネットワークをどのように通過するかを決定するシステム）から分離されたネットワークである。これらのアプリケーションは，スイッチ，ルータ，ブリッジ等のネットワークデバイス上の経路テーブルをプログラミングすることによって，パケット経路を制御する。そのようなアプリケーションの例には，ロードバランサー，ネットワークアドレス変換(NAT)，セキュリティアプリケーション，およびｏｐｅｎ　ｓｈｏｒｔｅｓｔ　ｐａｔｈ　ｆｉｒｓｔ(OSPF)などのルーティングアプリケーションが含まれる。）

「[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 on the network controller 102. Registering the UID on 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 can be 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 packetcan 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 include 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., flow table 215 for network device 206-2). The flow tables can include flow rules for existing flows (e.g., existing flow 225 defined in 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, an 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 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 only one class of traffic consisting of the trace packet to the network controller 202 (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 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 to the network controller 202.

[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 2026-2). 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 can set 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:0B: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 can set 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 can inject the trace packet 213 back into the network 200 via a defined network device (e.g., via the second network device 206-2) and instruct 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 can be a point of failure. 」

（当審訳：［００１７］　本開示のいくつかの例によれば，ネットワーク管理者は，ネットワークコントローラ１０２を使用して，既存の経路１０８のソースアドレスと一致するソースアドレス（例えば，コンピューティングデバイス１０４－２のアドレス 10.0.0.6）と，既存の経路１０８の宛先アドレスに一致する宛先アドレス（例えば，コンピューティングデバイス１０４－１のアドレス 10.0.0.5）と，トレースパケットのための一意の識別子（ＵＩＤ）を含むトレースパケットを生成することができる。ネットワークコントローラ１０２は，（例えば，既存の経路１０８に関連する他のトラフィックをネットワークコントローラ１０２に反映することとは対照的に）効率的且つ適切な分析のために，一意に識別できるようにトレースパケットを暗号化して，ＵＩＤを生成することができる。トレースパケットを暗号化することは，トレースパケットを符号化形式に変換することを含むことができる（例えば，ハッシングアルゴリズムなどのアルゴリズムの動作により，トレースパケットを入力としてアルゴリズムを動作させることより，ＵＩＤを生成する。）。ＵＩＤは，ネットワークコントローラ１０２に登録することができる。ネットワークコントローラ１０２上のＵＩＤの登録，および／又は，観測点１０７に到達する可能性のあるトレースパケットのＵＩＤを埋め込むことは，観測点１０７および／又はネットワークコントローラ１０２によるトレースパケットの一意の識別を容易にする。観測点１０７は，以下でより詳細に説明される。ＵＩＤを登録することにより，

ネットワークコントローラ１０２は，受信したパケットを暗号化してＵＩＤを生成し，それを登録されたＵＩＤと比較し，ＵＩＤの一致に応答して，受信したパケットを一致したトレースパケットと認識する。いくつかの例では，ＵＩＤをトレースパケットに追加することができる（例えば，トレースパケットの生成中および暗号化中に）。トレースパケットは，ネットワークレイヤ内のプロトコル，または開放型システム間相互接続（ＯＳＩ）モデル内のプロトコルなど，何らかの形式でデータペイロードを搬送するプロトコルに対して生成することができる。そのようなプロトコルの例は，伝送制御プロトコル（ＴＣＰ），ユーザデータグラムプロトコル（ＵＤＰ），ダイナミックホストコンフィギュレーションプロトコル（ＤＨＣＰ）およびＩＣＭＰなどを含むことができる。［００１８］　ネットワークコントローラ１０２は，観測点１０７（例えば，ネットワークデバイス１０６－１）を割り当てるために使用される。観測点１０７は，プログラムされた経路１０８上の任意のネットワークデバイス１０６に割り当てる。ネットワークコントローラ１０２は，トレースパケットによって指定されたマッチング基準を含む任意のパケットを，経路解析のために，ネットワークコントローラ１０２に送信するように観測点１０７に指示する。観測点１０７が割り当てられると，プログラムされた経路１０８をテストするために，ＵＩＤを有するトレースパケットを生成する。観測点１０７が指定された一致基準を有するパケットを受信すると，パケットのコピーをネットワークコントローラ１０２に送信する。ネットワークコントローラ１０２は，パケット及び／又はＵＩＤの指紋（例えば，通過したネットワークデバイス１０６のアドレス及び／もしくはポートを含む，パケットの経路履歴）を得るために，パケットを復号する。トレースパケットの受信した指紋がマッチング基準と一致する（例えば，プログラムされた経路と一致する）ことに応答して，ネットワークコントローラ１０２は，観測点１０７からのトレースパケットを受信したという記録（例えば，メモリ構造）を更新する。ネットワークコントローラ１０２は，観測点１０７のトレースパケットを受信したことに応じて，観測点１０７に成功のステータスを設定する。ネットワーク管理者は，ネットワークコントローラ１０２を使用して，観測点１０７がトレースパケットを受信したどうかを問い合わせることができる。ステータスが成功に設定されていれば，観測点１０７として割り当てられたネットワークデバイス１０６に関してプログラムされた経路が正しいことを意味する。ステータスが成功に設定されていない場合，トレースパケットがドロップされているか，観測点で未だ受信されていないことを意味する。

［００１９］　ネットワーク管理者は，ネットワークコントローラ１０２を使用して，トレースパケットのプログラムされた経路１０８に沿って（例えば，以前の観測点に設定された成功状態に応じて），別のネットワークデバイス（例えば，ネットワークデバイス１０６－２）を観測点１０７として再割り当てを行う。観測点１０７を変更することは，プログラムされた経路１０８に沿って，障害発生箇所の判断を容易にすることができる。

［００２０］　図２は，本開示におけるネットワーク２００の一例を示す図である。ネットワーク２００は，図１に示したネットワーク１００に類似する。

ネットワーク２００はＳＤＮであり，ＳＤＮコントローラ２０２を含む。ネットワーク２００は，コンピューティングデバイス２０４－１，２０４－２（本明細書では，一般にコンピューティングデバイス２０４と呼ぶ。）と，いくつかのネットワークデバイス２０６－１，２０６－２，２０６－３（本明細書では，一般にネットワークデバイス２０６と呼ぶ。）を含んでいる。第１のコンピューティングデバイス２０４－１は，第１のネットワークデバイス２０６－１，第２のネットワークデバイス２０６－２および第３のネットワークデバイス２０６－３を介し，いくつかの物理的リンクを通じて，第２のコンピューティングデバイス２０４－２に接続する。図２に示す例では，物理リンク２１０（例えば，予測経路）は，プログラムされた経路２０８（例えば，既存の経路）と同じである。図２はまた，本明細書に記載されているように，いくつかの経路挿入及び分析動作２１４を含む（例えば，ネットワーク２００には，コントロールプレーンが実装される）。

［００２１］　図２は，各ネットワークデバイスの経路テーブルを含む（例えば，ネットワークデバイス２０６－２のための経路テーブル２１５）。経路テーブルは，既存の経路（例えば，経路テーブル２１５に定義されている既存の経路２２５）に対する経路ルールを含んでいる。一例として，ネットワークコントローラ２０２は，第１のコンピューティングデバイス２０４－１と第２のコンピューティングデバイス２０４－２との間のプログラム経路２０８を試験するために，第２のネットワークデバイス２０６－２に観測点を割り当てることができる。いくつかの例では，ネットワーク管理者は，ネットワークコントローラ２０２を介して観測点を割り当てることができる。ネットワークコントローラ２０２は，トレースパケット２１３を生成し，トレースパケット２１３を登録し（本明細書で使用されるように，トレースパケットを登録することは，トレースパケットを登録すること自体，トレースパケットの指紋を登録する，及び／又は，トレースパケットのＵＩＤを登録することを含む。），既存の経路２２５にソースアドレス２１６－２，宛先アドレス２１８－２，トレースパケット２１３のプロトコルで送信することができる。いくつかの例において，ネットワーク管理者は，ネットワークコントローラ２０２を介して，トレースパケットを定義することができる。図示のように，第２のネットワークデバイス２０６－２に定義されている既存の経路２２５は，トレースパケット２１３のソースアドレス２１６－２と宛先アドレス２１８－２と一致するソースアドレス２１６－１と宛先アドレス２１８－１を含む。いくつかの例において，トレースパケット２１３の特定のネットワークデバイスの入力ポート２２０－２は，既存の経路２２５の入力ポート２２０－１と対応する。したがって，経路ルール２２６は，第２のネットワークデバイス２０６－２上の既存の経路２２５に加えることができ，ここで，経路ルール２２６がトレースパケット２１３のソースアドレス２１６－２，宛先アドレス２１８－２，着信ポート２２０－２，優先度２２２－２，出力ポート／動作２２４－２を含む。

［００２２］　経路ルール２２６は，第２のネットワークデバイス２０６－２に，トレースパケット２１３に対応する既存の経路２２５の優先度２２２－１（例えば，１０００）よりも高い優先度（例えば，１００１）を有するトレー

スパケット２１３を追加することができる。トレースパケット２１３のためのより優先度の高い経路ルール２２６は，第２のネットワークデバイス２０６－２の既存の経路ルール２２６に対して，既存の経路のための定義された動作を行う前もしくは代わりに，定義された動作２２４－２を起こさせることができる。図２に示された例は，これを意味し，対応するパケット（トレースパケット２１３）は，「コントローラポート」から送信される（例えば，ネットワークコントローラ２０２に送信される）。

［００２３］　図２の経路ルール２２６には具体的に示されていないが，経路ルール２２６は，トレースパケットからなるトラフィックの１つのクラスのみをネットワークコントローラ２０２に送信するために経路テーブルの機能に基づいて，具体的な経路を指定することができる（例えば，コントローラ２０２に送信されるトラフィックをの量を減らすため。）。

［００２４］　図２の経路ルール２２６には具体的に示されていないが，経路ルール２２６はまた，トレースパケット２１３のためのプロトコルを指定することができる。これにより，トレースパケット２１３は，アプリケーション固有とすることができる。プロトコルを指定することは，第２のネットワークデバイス２０６－２が，ソースアドレス２１６－２及び宛先アドレス２１８－２に一致したパケット（例えば，そのプロトコルの経路が適切に機能しているためにエラーを経験していない経路と関連している他のトラフィック）を送信することを防ぐ。

［００２５］　ネットワークコントローラ２０２は，既存の経路のプログラムされた経路２０８に沿って，ネットワークデバイスの指定されたポート（例えば，ネットワークデバイス２０６－１のポート２）を介して既存の経路のソース（例えば，コンピュータティングデバイス２０４－１）の代わりに，トレースパケット１３をネットワーク２００に注入することができる。

［００２６］　トレースパケット２１３が注入された，プログラムされた経路２０８は，第１のネットワークデバイス２０６－１に正しくプログラムされている場合，第１のネットワークデバイス２０６－１はトレースパケット２１３を第２のネットワークデバイス２０６－２に正しく転送しなければならない（例えば，第１のネットワークデバイス２０６－１のポート３から出て，第２のネットワークデバイス２０６－２のポート４に入る。）。第２のネットワークデバイス２０６－２には，経路ルール２２６により観測点が割り当てられているので，第２のネットワークデバイス２０６－２はトレースパケット２１３をネットワークコントローラ２０２に送る。いくつかの例において，トレースパケット２１３が受信されたという理由だけで，ネットワークコントローラ２０２は，第２のネットワークデバイス２０６－２（観測点）について成功のステータスを設定する。いくつかの例において，ネットワークコントローラ２０２は，第２のネットワークデバイス２０６－２から受信したトレースパケットの指紋を最初に調べ，それをプログラムされた経路２０８に基づいてトレースパケット２１３の予想一致基準と比較することができる。この例では，指紋及び／又は一致基準が以下の１つまたは複数の情報を含むことができる。：コンピューティングデバイス２０４－１のソースアドレス（00:1E:08:AE:D3:BE），

コンピューティングデバイス２０４－１の出力ポート（１），第１のネットワークデバイス２０６－１の入力ポート（２），第１のネットワークデバイス２０６－１の出力ポート（３），第２のネットワークデバイス２０６－２の入力ポート（４），そして／もしくは，第２のコンピューティングデバイス２０４－２の宛先アドレス(00:0c:29:03:7f:48)。第２のネットワークデバイス２０６－２から受信したトレースパケットの指紋がトレースパケット２１３の予想一致基準と一致する場合，ネットワークコントローラ２０２は第２のネットワークデバイス２０６－２に成功のステータスを設定する。

［００２７］　ネットワークコントローラ２０２が第２のネットワークデバイス２０６－２の成功のステータスに設定した後，ネットワーク管理者は（例えば，ネットワークコントローラ２０２を介して），第２のネットワークデバイス２０６－２のステータスを問い合わせることができる。ステータスが成功に設定されている場合，ネットワーク管理者はプログラムされた経路２０８のテストの継続を行いたい場合がある。ネットワークコントローラ２０２は，定義されたネットワークデバイスを介して（例えば，第２のネットワークデバイス２０６－２を介して），トレースパケットをネットワーク２００に戻し，異なる観測点（例えば，既存の経路において，前の観測点の下流に存在する異なるネットワークデバイス）に，ＵＩＤを含むパケットをネットワークコントローラ２０２に送信することを指示する。このテストプロセスは，テストパケット２１３がネットワークコントローラ２０２に戻されなくなるまで続けることができ，これは，テストパケット２１３がネットワークコントローラ２０２に戻されないときに，割り当てられた観測点が故障となっていることを示す。）

上記記載から，引用文献１には，以下の発明（以下，「引用発明」という。）が記載されていると認められる。

「ネットワークコントローラを使用して，既存の経路のソースアドレスと一致するソースアドレスと，既存の経路の宛先アドレスに一致する宛先アドレスと，トレースパケットのための一意の識別子（ＵＩＤ）を含むトレースパケットを生成し，

ネットワークコントローラは，既存の経路のプログラムされた経路に沿って，第１のネットワークデバイスの指定されたポートを介して既存の経路のソースの代わりに，トレースパケットをネットワークに注入し，

プログラムされた経路は，第１のネットワークデバイスに正しくプログラムされている場合，第１のネットワークデバイスはトレースパケットを第２のネットワークデバイスに正しく転送し，

経路ルールにより観測点が割り当てられている第２のネットワークデバイスは，トレースパケットをネットワークコントローラに送り，

ネットワークコントローラは，第２のネットワークデバイスから受信したトレースパケットの，第１のコンピューティングデバイスのソースアドレス，第１のコンピューティングデバイスの出力ポート，第１のネットワークデバイスの入力ポート，第１のネットワークデバイスの出力ポート，第２のネットワー

クデバイスの入力ポート，そして／もしくは，第２のコンピューティングデバイスの宛先アドレスからなる指紋を最初に調べ，予想一致基準と一致する場合，第２のネットワークデバイスに成功のステータスに設定し，
　ネットワークコントローラが第２のネットワークデバイスに成功のステータスに設定した後，ネットワーク管理者はネットワークコントローラを介して第２のネットワークデバイスのステータスを問い合わせることができ，
　ステータスが成功に設定され，プログラムされた経路のテストの継続を行いたい場合，ネットワークコントローラは，第２のネットワークデバイスを介して，トレースパケットをネットワークに戻すとともに，既存の経路において，前の観測点の下流に存在する異なるネットワークデバイスに，ＵＩＤを含むパケットをネットワークコントローラに送信することを指示する，
　コントロールプレーンが実装されたネットワークの方法。」

２　引用文献２について
　原査定の拒絶の理由された引用文献２には，図面とともに次の事項が記載されている。

「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 computer 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.」
（当審訳：ＩＮＴの実施例をここで説明する。ソースｖＳｗｉｔｃｈ（ｖＳｗｉｔｃｈ１）は，パケットがネットワーク要素で遭遇する遅延（ローカル出力タイムスタンプとローカル入力タイムスタンプとの間の差）を報告するために，各ネットワーク要素に対する命令を埋め込む。受信側のｖＳｗｉｔｃｈ（ｖＳｗｉｔｃｈ２）は，ホップ毎の遅延の合計としてエンドツーエンドの遅延を計算できる（今日のネットワークは，スイッチング及びキューイングの遅延が支配的であり，伝搬遅延は最小であるとしている）。宛先ｖＳｗｉｔｃｈで受信されたパケットのホップ毎の遅延も，どのネットワーク要素がエンドツーエンドの待ち時間に最も寄与したかを判断するために使用する。）

　上記記載から，引用文献２には，以下の事項（以下，「引用文献２記載事項」という。）

「受信側のｖＳｗｉｔｃｈ（ｖＳｗｉｔｃｈ２）は，ホップ毎の遅延の合計としてエンドツーエンドの遅延を計算でき，また，宛先ｖＳｗｉｔｃｈで受信されたパケットのホップ毎の遅延も，どのネットワーク要素がエンドツーエンドの待ち時間に最も寄与したかを判断するために使用すること。」

３　引用文献３について

本願の優先権主張日前に頒布された又は電気通信回線を通じて公衆に利用可能となった刊行物である，Ｋａｎａｋ Ａｇａｒｗｌ 他３名，ＳＤＮ ｔｒａｃｅｒｏｕｔｅ：Ｔｒａｃｉｎｇ ＳＤＮ Ｆｏｒｗａｒｄｉｎｇ ｗｉｔｈｏｕｔ Ｃｈａｎｇｉｎｇ Ｎｅｔｗｏｒｋ Ｂｅｈａｖｉｏｒ，ＡＣＭ ＳＩＧＣＯＭＭ’１４，２０１４年８月２２日，Ｐ．１４５－１５０（以下，「引用文献３」という。）には，図面とともに以下の事項が記載されている。

「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 given <switch-id, port>is repeated in the route, indicating the presence of a loop. Lastly, notice 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.」

（当審訳：３．４　トレースルートの実行

ネットワークが上述のように構成されると，ルートトレース用のイーサネットプローブフレームを受け入れる準備が整う。このプロセスは，図２に示す例によって最もよく説明される。

SDN tracerute は，注入ポイントを特定することから開始される。これはＡＰＩ呼び出しで識別されるか，又は送信元ＭＡＣまたはＩＰアドレスによって検索される送信元ホストの接続点にあるとみなされる。SDN tracerute は，注入スイッチ識別子とポートを取得すると，入力スイッチのカラーを調べ，そのカラーをプローブフレームのヘッダータグビットに挿入する。

SDN tracerute は，入力ポートを注入ポイントに設定し，PACKET_OUT をメッセージとしてプローブを入力スイッチに送信する。PACKET_OUT の動作は，TABLE に設定され，スイッチがあたかも入力ポートで受信したかのように扱う必要があることを示す（step1）。

PACKET_OUT を受信すると，入力スイッチはその経路テーブル内のパケットを処理する。パケット内のヘッダータグビットは，スイッチ自体のカラーに設定されているので，パケットは SDN tracerute がインストールした優先順位の高いルールのいずれにも一致しない。その結果，パケットは，通常のデフォルトタグ付きパケットであるかのように，次のホップに転送される（step2）。これにより，本当のトラフィックでないプローブであっても，スイッチ内の実際の転送ルールがパケットのルーティングに使用される。

パケットは，２番目のスイッチに到着する一方で，第１のスイッチのカラーに設定されたヘッダータグビットを依然として保持する。各スイッチは，隣接するスイッチのカラーと一致する全てのパケットをトラップする優先度の高いルールが設定されているため，２番目のスイッチのパケットは一致し，PACKET_IN としてコントローラに送信する（step3）。SDN tracerute は，コントローラでパケットを受信し，経路内の次のホップとしてパケットをコントローラに転送したスイッチの，スイッチＩＤとポート情報を記録する。

SDN tracerute は，現在のホップを記録すると，予約タグフィールドを現在のスイッチのカラーに対応するビットに書き換えることで，受信したプローブフレームを修正する。次に，PACKET_IN メッセージを送った同じスイッチに，PACKET_OUT として，修正されたプローブを戻す。PACKET_OUT の入力ポートは，パケットがスイッチで受信した入力ポートに設定される。アクションフィールドは，再び TABLE に設定される（step4）。スイッチは，コントローラにより修正されたプローブを受信し，その経路テーブルの動作をプローブに適用する。修正されたプローブの予約済みタグビットは，スイッチのカラーに設定されているため，タグベースのルールは一致せず，パケットは次のホップに沿って，通常のフレームとして転送される（step5）。

このプロセス（step3～5）は，経路内の各ホップに対して繰り返される。連続する PACKET_IN イベントの間に，パケットがネットワークを離れたか，ホストによって消費された，あるいは，経路内で所定の <switch-id, port> が反復されるループの存在を示す，タイムアウトが発生しとき，プロセスは終了する。

最後に，step3 で，トレースルートアプリケーションは，PACKET_IN を送信するスイッチのカラーと一致しないプローブ PACKET_IN メッセージのみを処理することに注意されたい。これにより，入力スイッチのカラーと一致する PACKET_IN メッセージをコントローラ内のモジュールに転送して処理させることができる。これにより，リアクティブルールのインストールなど，スイッチでの通常のパケット処理自体がコントローラに対して PACKET_IN を開始する可能性があるシナリオが可能となる。）

４　引用文献４について

本願の優先権主張日前に頒布された又は電気通信回線を通じて公衆に利用可能となった刊行物である，特開２００６－３１９９７３号公報（以下，「引用文献４」という。）には，図面とともに以下の事項が記載されている。

「【技術分野】
【０００１】
本発明は、プロトコル汎用のネットワーク傍受に関する。」

「【課題を解決するための手段】
【０００８】
本発明の実施形態は、通信網上で通信されるトラフィックをモニタリングし、該トラフィック内で、傍受装置が目的とするパケットを識別することができる、プロトコル汎用（すなわち「プロトコルを意識しない」）の傍受装置を提供する。さらに、本明細書に説明するように、パケット識別およびパケット認証を可能にする手法も提供される。さらに、本明細書に説明するように、所定の実施形態では、このような「パケット認証」は、パケット全体を認証するのではなく、傍受装置のためのＩＤおよび内容（コンテンツ）だけを認証する。こうして、傍受装置は、目的とする（of interest）パケットを識別して、識別情報および該傍受装置が目的とする情報を認証することができる。本発明の実施形態は、プロトコル汎用であり、傍受装置が目的とするパケットを識別して認証するために、傍受装置は、使用されている通信プロトコルに関して事前の知識を有する必要がない。したがって、傍受装置を変更することなく、プロトコル汎用の該傍受装置を使用して、該傍受装置を、通信網上で使用されている任意の通信プロトコルに対して動的に適応させることができる。
【０００９】
さらに詳述するように、パケットを、傍受装置が目的とするパケットとして識別するために、プロトコルに固有の方式でパケット内に情報（たとえばパケットのヘッダ）を含めるのではなく、本発明の実施形態は、プロトコル汎用の方式で、識別子を含める。たとえば、識別子は、パケットのペイロード

(payload)に含めることができる。一部の実施形態では、傍受装置は、捕捉したパケットのペイロードを識別子に関してスキャンし、該識別子を認識する際に、該パケットを認証するための技術を使用することができる。

【００１０】

一実施形態では、本方法は、通信網上で通信されたパケットを傍受装置が捕捉することを含む。傍受装置は、パケットのペイロードをスキャンし、傍受装置が目的とする内容をパケットが含むことを識別する識別子を、該パケットのペイロードが含むかどうかを判断する。少なくとも一部の手法では、パケットのペイロードがこのような識別子を含むとの判断に基づいて、傍受装置は、該パケットのペイロード内に含まれる内容を使用する。

【００１１】

一実施形態では、本方法は、傍受装置のための内容（コンテンツ）を含むパケットを形成することを含む。ここで、パケットは、ヘッダ部分およびペイロード部分を含む。ペイロード部分は、ａ）傍受装置のための内容（コンテンツ）をパケットが含むことを識別する所定の識別子と、ｂ）該傍受装置のための該内容と、ｃ）該所定の識別子および該傍受装置のための該内容を認証する認証トークンと、を含む。本方法はさらに、通信網を介してパケットを宛先に転送することを含む。一部の実施形態では、パケットを転送する宛先は、傍受装置以外の宛先である。ここで、傍受装置は、このパケットを傍受し、このパケットが該傍受装置のための内容を含むことを認識する。

【００１２】

上記は、次の本発明の詳細な説明をよりよく理解するために、本発明の特徴の概要を示したものである。次に、本発明の請求項の主題を形成する、本発明の別の特徴および利点を記述する。開示された概念および特定の実施形態は、本発明と同じ目的を実行するように他の構成を修正または設計する基礎としても容易に使用できることが理解されるであろう。また、このような等価の構成も、請求項に示す本発明から離れるものではないことを理解されたい。本発明の構成と方法の両方について、本発明に特徴的と考えられる新規な特徴、および、別の目的と利点は、図面と共に次の説明を読めばよりよく理解されるであろう。各図は図示と説明のために提供されたものに過ぎず、本発明の限定を定義することを目的としているわけではなことを理解されたい。」

第６　対比・判断

１　本願発明１について

（１）対比

本願発明１と引用発明とを対比すると，次のことがいえる。

ア　引用発明の方法は，「ネットワーク管理者はネットワークコントローラを介して第２のネットワークデバイスのステータスを問い合わせることができ，」又，「ステータスが成功に設定され，プログラムされた経路のテストの継続を行いたい場合，ネットワークコントローラは，第２のネットワークデバイスを介して，トレースパケットをネットワークに戻すとともに，既存の経路において，前の観測点の下流に存在する異なるネットワークデバイスに，ＵＩＤを含

むパケットをネットワークコントローラに送信することを指示」しているから，このことは，ネットワークコントローラを介した問い合せ等に答えるツールであるといえる。
　そうすると，引用発明のツールは，本願発明１の「ネットワーク管理者によるネットワークの管理ツール」に相当する。

イ　また，引用発明は，上記アで検討したように，ネットワークコントローラを介した問合せ等に答えるツールであって，ネットワーク管理者のネットワークの経路に関する問い合せに対して回答を行っており，この回答のために，何らかの演算を行っていることは明らかであり，さらに，対象とするネットワークはコントロールプレーンが実装されたものである。
　してみると，引用発明と，本願発明１の「集中式コントロールプレーンに基づくパス状態の報告演算方法」は，「コントロールプレーンに基づくパス状態の報告演算方法」である点で共通する。

ウ　引用発明の「ネットワークコントローラを使用して，既存の経路のソースアドレスと一致するソースアドレスと，既存の経路の宛先アドレスに一致する宛先アドレスと，トレースパケットのための一意の識別子（ＵＩＤ）を含むトレースパケットを生成」することは，本願発明１の「生成段階として、テストを行いたいデータフロー情報に従い実際のパケットをシミュレーションしたテストパケットを生成」することに相当する。

エ　引用発明の「ネットワークコントローラは，既存の経路のプログラムされた経路に沿って，第１のネットワークデバイスの指定されたポートを介して既存の経路のソースの代わりに，トレースパケットをネットワークに注入」することは，本願発明１の「テスト段階として、テストパケットをテストを行いたいパスが通過する交換機に送信」することに相当し，また，上記アで検討したように，引用発明は，トレースパケットの経路の状態をネットワークデバイスのステータスで管理しているから，本願発明１の「パステスト」に相当するテストを行っていると認められる。
　さらに，該「ネットワークコントローラは，既存の経路のプログラムされた経路に沿って，第１のネットワークデバイスの指定されたポートを介して既存の経路のソースの代わりに，トレースパケットをネットワークに注入」することは，本願発明１の「テスト段階が、テストパケットをテストを行いたいパス始点に位置する交換機に送信」することに相当する。

オ　また，上記アで検討したように，引用発明は，トレースパケットの経路の状態をネットワークデバイスのステータスで管理しており，このステータスをネットワーク管理者はネットワークコントローラを介して問い合わせることができるから，この問合せに対する回答と，本願発明１の「報告段階として、データフローが通過する実際のパスとパス状態を報告する」こととは，「報告段階として、データフローが通過する実際のパスを報告する」点で共通する。

カ　そうすると，本願発明１と引用発明とは以下の点で一致し，又，相違する。

［一致点］
「ネットワーク管理者のネットワーク管理ツールによって実行され，
　生成段階として，テストを行いたいデータフロー情報に従い実際のパケットをシミュレーションしたテストパケットを生成し，
　テスト段階として，テストパケットをテストを行いたいパスが通過する交換機に送信し，パステストを行い，
　報告段階として，データフローが通過する実際のパスを報告する，ことを主に含み，
　テスト段階が，テストパケットをテストを行いたいパス始点に位置する交換機に送信し，
　コントロールプレーンに基づくパス状態の報告演算方法。」

［相違点１］
　「報告段階」について，本願発明１の「報告段階として、データフローが通過する実際のパスとパス状態を報告」しているのに対して，引用発明は，データフローが通過するパス状態を報告していない点。

［相違点２］
　「テスト段階」について，本願発明１は「テスト段階が、テストパケットをテストを行いたいパス始点に位置する交換機に送信し、交換機が直ちに行う報告、テストを行いたいパス終点に位置する交換機による報告済かの判断と時間制限に到達済かの判断を待機後に受信し、終点交換機による報告済かの判断が主に終点交換機による報告済か否かの判断であり、報告済ならば報告段階に入り、未報告ならば時間制限に到達済の判断に入る手順であり、時間制限に到達済かの判断の手順が主にテストの時間制限が到達済か否かの判断であり、到達済ならば報告段階に入り、未到達ならば交換機が直ちに報告するのを待機後に受信するに戻る手順を含」んでいるのに対して，引用発明は「テストパケットをテストを行いたいパス始点に位置する交換機に送信し」ているもの，「交換機が直ちに行う報告、テストを行いたいパス終点に位置する交換機による報告済かの判断と時間制限に到達済かの判断を待機後に受信し、終点交換機による報告済かの判断が主に終点交換機による報告済か否かの判断であり、報告済ならば報告段階に入り、未報告ならば時間制限に到達済の判断に入る手順であり、時間制限に到達済かの判断の手順が主にテストの時間制限が到達済か否かの判断であり、到達済ならば報告段階に入り、未到達ならば交換機が直ちに報告するのを待機後に受信するに戻る手順」は行っていない点。

［相違点３］
　本願発明１は「実際のパケットをシミュレーションしたテストパケットを生成することについて、主にテストパケットの payload に専用のキーワードを記

入することで、テストパケットが同じ header を有する他のパケットと区別できるものである」のに対して，引用発明の「トレースパケットのための一意の識別子（ＵＩＤ）」がパットのどの部分に含まれているのか不明である点。

［相違点４］
「コントロールプレーン」について，本願発明１は「集中式コントロールプレーン」であるのに対して，引用発明の「コントロールプレーン」は「集中式」であるのか否か不明である点。

（２）相違点についての判断
以下，［相違点１］について検討する。
引用文献１には，本願発明１の「パス状態」に相当する状態を報告することについて記載も示唆もない。
また，引用文献１において「パス状態」を報告することが自明の事項であるともいえない。
さらに，引用文献２記載事項には，本願発明１の「パス状態」に相当する「ホップ毎の遅延」を報告することが記載されているが，引用発明は，

ア　「第１のネットワークデバイスはトレースパケットを第２のネットワークデバイスに正しく転送し」た後，

イ　「経路ルールにより観測点が割り当てられている第２のネットワークデバイスは，トレースパケットをネットワークコントローラに送り」，

ウ　「ネットワークコントローラは，第２のネットワークデバイスから受信したトレースパケットの，第１のコンピューティングデバイスのソースアドレス，第１のコンピューティングデバイスの出力ポート，第１のネットワークデバイスの入力ポート，第１のネットワークデバイスの出力ポート，第２のネットワークデバイスの入力ポート，そして／もしくは，第２のコンピューティングデバイスの宛先アドレスからなる指紋を最初に調べ，予想一致基準と一致する場合，第２のネットワークデバイスに成功のステータスに設定し」，

エ　「ネットワーク管理者」が「ネットワークコントローラを介して第２のネットワークデバイスのステータスを問い合わせ」た後，「ネットワーク管理者」が「プログラムされた経路のテストの継続を行いたい場合」に，「ネットワークコントローラは，第２のネットワークデバイスを介して，トレースパケットをネットワークに戻すとともに，既存の経路において，前の観測点の下流に存在する異なるネットワークデバイスに，ＵＩＤを含むパケットをネットワークコントローラに送信することを指示」しており，

第２のネットワークデバイスがトレースパケットを受け取ってから，トレースパケットをネットワークに戻す間に，ネットワークコントローラにトレースパケットを送り，ステータスの設定をした後に，ネットワークコントローラが

第２のネットワークデバイスを介して，トレースパケットをネットワークに戻すことを行っている。
　そうすると，引用発明では，上記引用文献２記載事項の「ホップ毎の遅延」を測ることができないから，引用発明に，引用発明２記載事項を採用することには，阻害要因があるといえる。
　また，引用文献３及び４には，本願発明１の「パス状態」に相当する状態を報告することについて記載も示唆もない。
　そうすると，他の相違点について検討するまでもなく，本願発明１は，引用発明及び引用文献２ないし４に記載された事項から，当業者が容易に発明できたものであるとはいえない。

２　本願発明２ないし６について
　本願発明２ないし６も，本願発明１の「報告段階として、データフローが通過する実際のパスとパス状態を報告する」ことを行っているから，本願発明１と同じ理由により，本願発明２ないし６は，引用発明及び引用文献２ないし４に記載された事項から，当業者が容易に発明できたものであるとはいえない。

第７　原査定について
　本願発明１ないし６は，「報告段階として、データフローが通過する実際のパスとパス状態を報告する」ことを行っているから，当業者であっても，拒絶査定において引用された引用文献１及び２に基づいて，容易に発明できたものとはいえない。したがって，原査定を維持することはできない。

第８　むすび
　以上のとおり，原査定の理由によって，本願を拒絶することはできない。
　また，他に本願を拒絶すべき理由を発見しない。
　よって，結論のとおり審決する。

令和１年８月１９日

審判長　特許庁　審判官　▲吉▼田 耕一
特許庁　審判官　小田 浩
特許庁　審判官　梶尾 誠哉

〔審決分類〕Ｐ１８．１２１－ＷＹ（Ｈ０４Ｌ）

審判長　特許庁　審判官　▲吉▼田　耕一　9194
特許庁　審判官　梶尾　誠哉　9370
特許庁　審判官　小田　浩 9188