(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6193473号
(P6193473)
(45)発行日　平成29年9月6日(2017.9.6)　(24)登録日　平成29年8月18日(2017.8.18)

(51)Int.Cl.　F I
H04L 12/741 (2013.01)　H04L 12/741

請求項の数 13　(全 38 頁)

(21)出願番号	特願2016-508179 (P2016-508179)	(73)特許権者	515289598
(86)(22)出願日	平成26年4月17日 (2014.4.17)		エンチュイティ　リミテッド
(65)公表番号	特表2016-519911 (P2016-519911A)		ＥＮＴＵＩＴＹ　ＬＩＭＩＴＥＤ
(43)公表日	平成28年7月7日 (2016.7.7)		イギリス、イーシー２エム　４ワイエル
(86)国際出願番号	PCT/EP2014/057962		ロンドン、デボンシャー　スクエア　９エー
(87)国際公開番号	WO2014/170457		
(87)国際公開日	平成26年10月23日 (2014.10.23)	(74)代理人	100129425
審査請求日	平成28年1月21日 (2016.1.21)		弁理士　小川　護晃
(31)優先権主張番号	1307131.1	(74)代理人	100099623
(32)優先日	平成25年4月19日 (2013.4.19)		弁理士　奥山　尚一
(33)優先権主張国	英国 (GB)	(74)代理人	100087505
(31)優先権主張番号	1406568.4		弁理士　西山　春之
(32)優先日	平成26年4月11日 (2014.4.11)	(74)代理人	100168642
(33)優先権主張国	英国 (GB)		弁理士　関谷　充司

最終頁に続く

(54)【発明の名称】コンピュータ実施方法、コンピュータプログラム製品及びコンピュータ

(57)【特許請求の範囲】
【請求項１】
コンピュータネットワーク内で接続されたフォーカスデバイスの出口ポートを識別するコンピュータ実施方法であって、
前記コンピュータネットワークに接続されたモニタコンピュータにおいて実施され、
前記フォーカスデバイスに対するクエリメッセージを発生し、前記クエリメッセージは、前記フォーカスデバイスを識別するアドレスと、宛先識別子に基づいて構築されたクエリキーとを含み、前記クエリメッセージが前記フォーカスデバイスで受信された場合には、前記宛先識別子において識別される宛先にアドレス指定されたメッセージ用の出口ポートの識別を更に含む結果メッセージを戻すために前記フォーカスデバイスで読み取り可能な命令を含むことと、
前記モニタコンピュータにおいて結果メッセージを受信することと、
前記結果メッセージを読み取ることと、
前記結果メッセージが出口ポートを識別しない場合に、同じフォーカスデバイスを識別するデバイスアドレスと、前記モニタコンピュータにより選択された異なるクエリキーと、前記フォーカスデバイスが前記出口ポートの識別を含む次の結果メッセージを戻すために前記フォーカスデバイスで読み取り可能な命令とを含む、少なくとも１つの次のクエリメッセージを自律的に発生することと
を行い、
ある次の結果メッセージによって前記フォーカスデバイスの前記出口ポートが識別され

るまで複数のクエリメッセージを発生させる、コンピュータ実施方法。
【請求項２】
前記異なるクエリキーは、前記フォーカスデバイスで異なる転送テーブルにアクセスするために同一の宛先識別子に基づいて構築されている、請求項１に記載のコンピュータ実施方法。
【請求項３】
前記異なるクエリキーは、前記フォーカスデバイスにアクセスするために異なる宛先識別子に基づいて構築されている、請求項１に記載のコンピュータ実施方法。
【請求項４】
前記フォーカスデバイスがルーティングデバイスであり、前記クエリメッセージが前記ルーティングデバイスのルーティングテーブルに向けて送られ、前記結果メッセージが前記宛先識別子に対するルートタイプが間接的であることの指示を含み、前記結果メッセージを読み取るステップが、間接タイプを検出することと、異なる宛先識別子に基づいて構築されたクエリキーによって次のクエリメッセージを発生することとを含む、請求項３に記載のコンピュータ実施方法。
【請求項５】
前記結果メッセージが、前記コンピュータネットワークにおける前記フォーカスデバイスからのネクストホップのためのルーティングアドレスを含み、前記次のクエリメッセージが前記異なる宛先識別子としての前記ネクストホップのためのルーティングアドレスに基づいて構築されている、請求項３に記載のコンピュータ実施方法。
【請求項６】
前記次のクエリメッセージが、異なるクエリキーによって前記フォーカスデバイスを識別するアドレスを含む、請求項５に記載のコンピュータ実施方法。
【請求項７】
前記次のクエリメッセージが前記ネクストホップのためのルーティングアドレスによって識別される前記フォーカスデバイスにおけるマッピングテーブルに向けて送られ、前記マッピングテーブルが、ルーティングアドレスプロトコルに従った宛先識別子を、スイッチングアドレスプロトコルに従った宛先識別子に割り振る、請求項５又は６に記載のコンピュータ実施方法。
【請求項８】
前記クエリメッセージにおける前記宛先識別子がルーティングアドレスプロトコルに従ったものであり、前記異なるクエリキーがスイッチングアドレスプロトコルに従ったものである異なる宛先識別子に基づいて構築されている、請求項１に記載のコンピュータ実施方法。
【請求項９】
前記少なくとも１つの次のクエリメッセージが、前記マッピングテーブルにクエリを実行して、前記フォーカスデバイス上のどのインタフェースから前記ネクストホップアドレスのためのルーティングアドレスとスイッチングアドレスとの間のマッピングを導出したかを確認するように構築されている、請求項７に記載のコンピュータ実施方法。
【請求項１０】
前記結果メッセージが出口ポートを識別する場合、ネットワークトポロジに基づいて前記出口ポートに接続されたデバイスを識別しようとする、請求項１に記載のコンピュータ実施方法。
【請求項１１】
前記結果メッセージにおいて戻された前記出口ポートが接続されたデバイスを一意に識別しないと判定される場合、前記少なくとも１つの次のクエリメッセージが前記フォーカスデバイスの上位層及び／又は下位層のポートの関連付けを要求する、請求項１０に記載のコンピュータ実施方法。
【請求項１２】
出口ポート識別ユーティリティを実施するコンピュータプログラムを備えるコンピュー

タプログラム製品であって、
コンピュータにより実行された場合に請求項１から１１の何れか１項に記載のコンピュータ実施方法を実施する、コンピュータプログラム製品。
【請求項１３】
コンピュータであって、
少なくとも１つのフォーカスデバイスを含むネットワークと接続するためのネットワークインタフェースと、
コンピュータプログラムを実行して、請求項１から１１の何れか１項に記載のコンピュータ実施方法を実施するように構成されたプロセッサと、
を有する、コンピュータ。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、コンピュータネットワークにおいてデバイスにおける出口ポートを識別することに関する。
【背景技術】
【０００２】
コンピュータネットワークは、多種多様な状況において、ＩＴ（情報技術）インフラストラクチャの基盤を形成する。このようなコンピュータネットワークは、相互接続された様々なタイプのデバイスを含んでいる。ネットワークの目的は、ネットワーク上で情報、アプリケーション、及びサービス等を配信するように、それらのデバイス間のメッセージフローをサポートすることである。ネットワークを管理するために複数の技法を利用することができる。
この状況において、ネットワークの管理は、ネットワークを監視して障害点及びホットスポット等の他の問題のある領域を識別すること、並びに、その問題を修正できるようにネットワークの管理者及びユーザに情報を提供することを含む。ネットワークトポロジを提供するために利用可能な複数のツールがある。ネットワークトポロジは、ネットワーク内のデバイスがどのように物理的又は論理的に相互接続されるかを識別する。
このため、ある特定の単一のデバイスは、隣接デバイスに対して１つ以上の接続を有し得る。ネットワークを「発見する」コンピュータ化ツールが利用可能であり、それらは、ネットワーク内のデバイスの相互接続及びそれらのデバイスの性質を規定するネットワークトポロジを生成する。
【発明の概要】
【課題を解決するための手段】
【０００３】
本発明者らは、（ルーティングプロトコルの詳細についてのクエリ（ｑｕｅｒｙ）実行とは対照的に）フォーカスデバイスにクエリを実行して、それが仮想パケットとどのような関係を持つかを判定することによって、デバイスからの出口ポートを識別するための手法を開発した。第１のクエリが使用可能な答えを戻さない場合、モニタコンピュータにおいて実行されるユーティリティによって次（第２）のクエリを自動的に構築する。
【０００４】
本発明の一態様によれば、コンピュータネットワーク内で接続されたフォーカスデバイスの出口ポートを識別するコンピュータ実施方法が提供される。このコンピュータ実施方法は、コンピュータネットワーク内で接続されたフォーカスデバイスの出口ポートを識別するコンピュータ実施方法であって、上記コンピュータネットワークに接続されたモニタコンピュータにおいて実施され、上記フォーカスデバイスに対するクエリメッセージを発生し、上記クエリメッセージは、上記フォーカスデバイスを識別するアドレスと、宛先識別子に基づいて構築されたクエリキーとを含み、上記クエリメッセージが上記フォーカスデバイスで受信された場合には、上記宛先識別子において識別される宛先にアドレス指定されたメッセージ用の出口ポートの識別を更に含む結果メッセージを戻すために上記フォ

ーカスデバイスで読み取り可能な命令を含むことと、上記モニタコンピュータにおいて結果メッセージを受信することと、上記結果メッセージを読み取ることと、上記結果メッセージが出口ポートを識別しない場合に、（ｉ）上記コンピュータネットワーク内で接続された異なるデバイスのための異なるアドレスと（ii）上記モニタコンピュータにより選択された異なるクエリキーとの少なくとも１つを含む、少なくとも１つの次のクエリメッセージを自律的に発生することと、を行い、上記フォーカスデバイスの上記出口ポートを識別するために充分なクエリメッセージを発生させる。

【０００５】

このコンピュータ実施方法は、モニタコンピュータにインストールされたコンピュータプログラムの形態の出口ポート識別ユーティリティを実行することにより実施可能である。また、本発明は、コンピュータにインストールされた場合に上記で規定した方法を実施するコンピュータプログラム製品も提供する。本発明は、更に、少なくとも１つのフォーカスデバイスを含むネットワークに接続するためのインタフェースと、上記で規定した方法を実施するコンピュータプログラムを実行するように構成されたプロセッサと、を有するモニタコンピュータを提供する。

【０００６】

上記異なるクエリキーは、上記フォーカスデバイスで異なる転送テーブルにアクセスするために同一の宛先識別子に基づいて構築することができる。あるいは、上記異なるクエリキーは、上記フォーカスデバイスにアクセスするために異なる宛先識別子に基づいて構築することができる。

【０００７】

上記フォーカスデバイスがルーティングデバイスである場合、上記クエリメッセージは、上記ルーティングデバイスのルーティングテーブルに向けて送ることができる。その場合、結果メッセージは、宛先識別子に対するルートタイプが間接的であることの指示を含むことができ、結果メッセージを読み取るステップは、間接タイプを検出することと、異なる宛先識別子に基づいて構築されたクエリキーによって次のクエリメッセージを発生することとを含む。

【０００８】

結果メッセージが、上記コンピュータネットワークにおける上記フォーカスデバイスからのネクストホップ（ｎｅｘｔ　ｈｏｐ）のためのルーティングアドレスを含む場合、上記次のクエリメッセージは、上記異なる宛先識別子としてのネクストホップのためのルーティングアドレスに基づいて構築することができる。上記次のクエリメッセージは、異なるクエリキーによって上記フォーカスデバイスを識別するアドレスを含むことができる。あるいは、上記次のクエリメッセージは、同一のクエリキーによって異なるデバイスを識別する異なるアドレスを含むことができる。

【０００９】

上記次のクエリメッセージは、上記ネクストホップのためのルーティングアドレスによって識別される上記フォーカスデバイスにおけるマッピングテーブルに向けて送ることができ、マッピングテーブルは、ルーティングアドレスプロトコルに従った宛先識別子を、スイッチングアドレスプロトコルに従った宛先識別子に割り振る（マッピング）。

【００１０】

上記クエリメッセージにおける宛先識別子がルーティングアドレスプロトコルに従ったものである場合、上記異なるクエリキーがスイッチングアドレスプロトコルに従ったものである異なる宛先識別子に基づいて構築することができる。これによって、いわゆるレイヤ２／レイヤ３調査（ｌａｙｅｒ２／ｌａｙｅｒ　３　ｉｎｖｅｓｔｉｇａｔｉｏｎ）をモニタコンピュータで自律的に実行することができる。

【００１１】

上記少なくとも１つの次のクエリメッセージが、マッピングテーブルにクエリを実行するように構築されている場合、これは、上記フォーカスデバイス上のどのインタフェースから上記フォーカスデバイスがネクストホップアドレスのためのルーティングアドレスと

スイッチングアドレスとの間のマッピングを導出したかを確認するようにセットアップすることができる。

【００１２】

上記結果メッセージが出口ポートを識別する場合、このコンピュータ実施方法は、結果メッセージにおいて戻された出口ポートが接続されたデバイスを一意に識別しないと判定するように構成することができる。その場合、自律的に発生された次のクエリメッセージは、上記フォーカスデバイスの上位層及び／又は下位層のポートの関連付けを要求することができる。

【００１３】

各デバイスに送信されるクエリは、各デバイスにクエリを実行して、上記宛先識別子により識別される宛先にアドレス指定された仮想メッセージのためにデバイスが用いる出口ポートを表す１つの出口ポートの識別子を明らかにするように構成されている。クエリが実行されるデバイスのネットワーク内の位置に応じて、所定のクエリについての宛先識別子は、ターミナルデバイスの宛先識別子か又はそれ以外とすることができる。これは、デバイスがルータである場合に、クエリを受信した時にそのアクティブなルーティングテーブル内に何があるかを問い合わせることによって実行可能である。上記宛先識別子は、例えばＩＰ（インターネットプロトコル）アドレス等、ルーティングテーブル又はＡＲＰテーブルに用いられる転送アドレスである。

【００１４】

クエリ自体は、モニタコンピュータからクエリ対象のデバイス（フォーカスデバイス）に送信されるメッセージ又は信号内に収容することができる。クエリメッセージ又は信号は、経路が決定されるメッセージフローを構成しない。各クエリは、宛先識別子（転送アドレス）を含み、これは、フォーカスデバイスの転送テーブルにクエリを実行して、クエリの受信時にフォーカスデバイスが決定する必要がある場合にその宛先にアドレス指定された仮想メッセージをどのように処理するかを明らかにする。

したがって、フォーカスデバイスは、その宛先にアドレス指定された実メッセージにその時使われるはずの直接（ｉｍｍｅｄｉａｔｅ）出口ポートを識別する結果を戻す。クエリは、ネットワークがアクティブでありメッセージフローが所定の位置にある状態で送信することができる。

しかしながら、メッセージフロー自体がアクティブでない場合にクエリを送信することも可能である。つまり、この技法は、どちらの状況でも使用可能である。

【００１５】

クエリがメッセージ又はパケットの形態である場合、例えばメッセージが宛先ＩＰアドレスを有するＳＮＭＰメッセージであり得る場合、クエリは、それ自体の宛先アドレスを搬送し、モニタコンピュータからフォーカスデバイスまでネットワーク上で配信することができる。その場合、クエリメッセージの宛先アドレスはフォーカスデバイスのものである。これは、クエリ自体に含まれる宛先識別子（転送アドレス）と同じものではない。代替的な構成では、１つ又は複数のクエリ信号を、モニタコンピュータから、ＣＬＩ又はＸＭＬ　ＡＰＩ機構等の直接接続を介してフォーカスデバイスに送信することができる。

【００１６】

ユーティリティは、本明細書において「推論的なキーイング（ｓｐｅｃｕｌａｔｉｖｅ　ｋｅｙｉｎｇ）」と定義する技法を用いて向上させることができる。これによって、推論的なキーの有界リスト（ｂｏｕｎｄｅｄ　ｌｉｓｔ）を発生して、トラフィック転送テーブルのクエリ実行を簡略化すると共に、そのようなキーに必要なコンピュータネットワーク上のトラフィックを軽減することが可能となる。推論的なキーの小さい有界リストを発生させることによって、全ての要求を順次でなく並列に発行することができる（すなわち、一般的なクエリメッセージに複数のクエリキーを含ませることができる）。

【００１７】

転送アドレスを表すビットシーケンスを、埋め込みインデックスを表すビットシーケンスと論理的に組み合わせることによって、宛先識別子（転送アドレス）を各埋め込みイン

デックスと組み合わせることができる。

【００１８】

転送テーブルがルーティングテーブルである場合、ルーティングテーブル内の各インデックスは、ネットマスクである。クエリメッセージ内で用いるキーは、転送アドレスを表すビットシーケンスを、ネットマスクを表すビットシーケンスと論理的に組み合わせることで発生させた一意のキーのみから選択することができる。転送アドレスは、ＩＰ（インターネットプロトコル）アドレスとすればよく、転送テーブルは、レイヤ３ルーティングデバイス用のものとすればよい。

【００１９】

推論的なキーイング技法は、埋め込みインデックスがルータ等のトラフィック転送デバイスにおけるＡＲＰテーブルのインタフェースインデックスである場合にも適用可能である。

【００２０】

ユーティリティが特に有用であるのは、特定のメッセージフローについて相互接続デバイスのネットワークに採用される経路を識別するために本発明者等が開発した新規の手法において用いられる場合である。この技法は、「前もって」収集された最少量のデータ、すなわちスタティックなネットワークトポロジとエンドホスト位置（どのクライアント及びサーバがどのアクセススイッチ／エッジスイッチに接続されているか）だけに頼っており、そのような極めてダイナミックなデータの必要に応じて、要求されるものは実行中に極めて選択的に収集する。

最新のダイナミック環境では、直ちにすなわちリアルタイムでエンドツーエンド経路を計算する能力は応用範囲が広い。大規模な実世界ネットワークと共に用いる場合、データ収集及びその処理は、アルゴリズムが実用的な価値を持つために非常に迅速でなければならない。

【００２１】

特定のデバイスにおける挙動を、「ホップごとの振舞い（ＰＨＢ：ｐｅｒ　ｈｏｐ　ｂｅｈａｖｉｏｕｒ）」と呼ぶ。これは、この出口ポート識別ユーティリティの特に有益な点である。ＰＨＢは単独でエンドツーエンド経路を提供することはできない。しかしながら、パケットがデバイスを出ていく特定のインタフェースを知ることは、そのインタフェースにどのデバイス及びインタフェースが接続されているかわからない場合に有用であり得る。ＰＨＢと組み合わせてネットワークトポロジを用いることで、アプリケーションフローについてネットワークを通るエンドツーエンド経路を直接計算することが可能となる。

【００２２】

ネットワークトポロジの決定は、多くの方法で実行することができる。ネットワーク接続性の優れた表現を与えるために個別に又は組み合わせて利用可能な技法は、例えば以下を含む。

・シスコ検出プロトコル（ＣＤＰ：Ｃｉｓｃｏ　Ｄｉｓｃｏｖｅｒｙ　Ｐｒｏｔｏｃｏｌ）

・リンクレイヤ検出プロトコル（ＬＬＤＰ：Ｌｉｎｋ　Ｌａｙｅｒ　Ｄｉｓｃｏｖｅｒｙ　Ｐｒｏｔｏｃｏｌ）

・ＳｏＮＭＰ（ＳｙｎＯｐｔｉｃｓ　Ｎｅｔｗｏｒｋ　Ｍａｎａｇｅｍｅｎｔ　Ｐｒｏｔｏｃｏｌ）

・スパニングツリープロトコル（ＳＴＰ：Ｓｐａｎｎｉｎｇ　Ｔｒｅｅ　Ｐｒｏｔｏｃｏｌ）

・ＩＰ　Ｔｒａｃｅｒｏｕｔｅ

・ＩＰｖ６近隣探索（ＩＰｖ６　Ｎｅｉｇｈｂｏｕｒ　Ｄｉｓｃｏｖｅｒｙ）

・ユーザによる追加／変更／削除

【００２３】

ネットワークのトポロジを知ることは、極めて有用であるが、起こり得る全ての問題に

対する解決策を提供するわけではない。ネットワークは、遠い地理的位置間で長距離にわたって又は複数の相互接続デバイスを用いた極めて複雑なネットワークにおいて、アプリケーション及びサービスの配信をサポートするインフラストラクチャを提供するためにますます用いられるようになっている。ネットワーク管理者及びユーザは、必ずしもネットワークの全ての詳細を知ることでなく、ネットワーク上でのアプリケーション及びサービスの配信の性能の理解に関心を持つようになっている。したがって、いわゆる「エンドツーエンド」モニタリングはいっそう一般的なものとなっている。「エンドツーエンド」モニタリングでは、ソースデバイスから宛先デバイスへのメッセージフローを伴うアプリケーションは、そのソース及び宛先デバイス間で配信される際に性能が監視される。性能パラメータを用いてネットワークでの起こり得る障害について推定又は推測することができるが、パラメータはそういった障害の位置についての具体的な情報を与えないので、直接に解決策を示すわけではない。

【００２４】

多くの場合、ソースデバイスは特定のサービスを提供するサーバであり、宛先デバイスは、ネットワークを介してサーバに接続されてそのサービスを用いる必要があるクライアント端末である。本明細書において用いる場合、「デバイス」という言葉は、ネットワークにおいて接続可能ないかなるデバイスも包含することを意図している。「サーバ」という言葉は、サービス又はアプリケーションの配信を担当するデバイスを示すために用いる。「クライアント」という言葉は、そのアプリケーション又はサービスに依存するデバイス（ユーザベースの又は別の依存したマシンもしくはサーバ）を示すために用いる。

【００２５】

アプリケーションの性能が低下していることがわかった場合に問題がどこにあるか推測する際の大きな困難は、そのアプリケーションのためのメッセージフローにより採用された可能性のあるネットワーク内の経路についての理解の不足である。ネットワークは、エンドポイントデバイスを接続するために多くのタイプのネットワークデバイス（例えばルータ、スイッチ、ファイアウォール、サーバ負荷分散装置（ロードバランサ）等）に依存するので、いかなる所定のソースエンドポイントについても、そのエンドポイントからのメッセージがどのようにネットワーク内を所定の宛先エンドポイントまでルーティングされるかを示すことは極めて難しい。このような経路決定の複雑さは、複数の代替経路、冗長経路、ロードバランシング等を用いることで更に悪化する。

【００２６】

特定のパケットがネットワーク内をどのようにルーティングされるかを予測する試みが行われてきた。このような予測は、ネットワークトポロジの複雑なモデルと、特定のデバイスがネットワーク内でどのように振舞うかに関するデバイスごとの指示に基づく。ネットワークデバイスは極めて高性能である場合があり、特定のデバイスにおけるルーティング戦略を決定するために複数の複雑なアルゴリズムが開発されてきた。更に、ルーティング戦略は、ネットワークに影響を与えるトラフィック及びその他の環境的な要素（他のデバイスの障害等）に依存し得る。ルーティング戦略を決定するためデバイスによって利用可能である複雑なアルゴリズムは、例えば以下を含み得る。

・入口インタフェース及び入口インタフェース技術
・パケットヘッダ（Ｌ２、Ｌ３、ＭＰＬＳ、ＡＴＭ等）
・スタティック及び直接接続ルート
・共有ルーティングテーブル（ＢＧＰ、ＯＳＰＦ、ＲＩＰ、ＥＩＧＲＰ等の完全な知識。アクティブなネイバー、リンク状態、ルートコスト、ルート重量等）
・学習されたＭＡＣ転送テーブル
・アクセス制御リスト
・ネットワークオーバーレイ技術（例えばＭＰＬＳ、８０２．１ｑＶＬＡＮ）等
・ループ回避技術　例えばＰＶＳＴＰ
・トンネリングプロトコル（ＭＰＬＳ、ＩＰＳＥＣ、ＳＳＬ、ＧＲＥ）
・ロードバランス／冗長リンク

・デフォルトゲートウェイ

【００２７】

しかしながら、過去には原則として所定のパケットが次にどの特定デバイスへ転送されるか予測できたとしても、これには収集に時間がかかる膨大な量のデータが必要であり、ルーティングデバイス動作の即時性によってすぐに時代遅れになり得る。更に、このデータを獲得するだけでも、ネットワークデバイス及びネットワークの双方に著しい負荷がかかっていた。

【００２８】

本明細書に記載するユーティリティは、複数の有用なネットワーク解析技法を可能とする。これによってオンデマンドの経路決定を行えるので、特定のアプリケーションについて経路を決定しようとする管理者は、ある程度瞬時にモニタコンピュータに質問し、経路の結果を受信することができる。

【００２９】

これは、複数の経路の発見を可能とする。すなわち、ネットワーク内に環境変化があるので、ルーティングデバイスはそういった変化に応じて異なるようにメッセージフローをルーティングし得る。したがって、経路を識別するための第１のクエリセットが第１の経路を記録するのに対し、第２のクエリセットは、第２の経路を識別することがあり、これは第１及び第２のクエリセットが時間的に極めて近い場合であっても当てはまる。共通のエンドポイント（すなわち同一のソースデバイス及び同一の宛先デバイス）間の複数の経路についての情報をグラフ又は画像によって提示して、各経路を特定のメッセージフローのために採用した場合の経路の性質だけでなく時間の割合をユーザに示すことができる。これが容易に達成可能であるのは、クエリ自体がネットワークの著しいオーバーヘッドとならないので、性能に大きな影響を及ぼすことなく複数のクエリセットのディスパッチ（発信）が可能だからである。

【００３０】

本方法では、迅速な合理的（ｌｅｇｉｔｉｍａｔｅ）経路変更の検出が可能である。すなわち、ネットワークを調節して経路を変更することができ、これを検出して視覚的なグラフィカルユーザインタフェースでユーザに対してフラグ表示することができる。

【００３１】

共通のソースデバイスと宛先デバイスとの間に複数の経路がある場合、経路は異なる待ち時間を有する可能性がある。時として、インテリジェントなルーティングを実行するルーティングデバイスは、特定のメッセージが経路から経路に頻繁にスイッチングする「ルートフラップ」として知られる現象を起こすことがある。例えば、そのような経路変更がエンドツーエンド待ち時間に及ぼす影響のため、及びそのような「ジッター」がＶｏＩＰ（Ｖｏｉｃｅ　ｏｖｅｒ　ＩＰ）電話の会話に及ぼす影響のため、ネットワーク管理者は、そのような発生を識別することが有用であり得る。

【００３２】

本方法を用いて経路障害の位置を特定することができる。すなわち、この方法の好適な実施形態においては、デバイスが宛先デバイスとして識別されるまで、クエリをディスパッチし、結果を受信及び解析して、次のデバイスを識別する。しかしながら時として、ネットワークに障害があるために、ネットワークは、メッセージフローを宛先デバイスまで配信しない。この方法は、エンドツーエンド経路を横断できなくなるまで経路に沿って動作することによってその状況を識別することができ、続いて、このネットワーク位置を管理者に通知することができる。

【００３３】

更に、本方法は、ネットワークトポロジに基づいた推定を用いて、その経路における次のデバイスで再出発する可能性を与えることができる。次に、その障害点から宛先デバイスに到達するまで再び経路発見（経路識別）法を採用すればよい。このように、モニタリングコンピュータが認知性を持たないネットワーク部分（例えば適切な管理インタフェースを持たないデバイス、又は異なる組織に属するデバイス）は迂回することができ、経路

解析を継続する。

【００３４】

また、本方法は、非対称ルーティング識別を可能とする。ソースデバイスと宛先デバイスとの間のメッセージフローが、その方向に応じて異なる経路を採用することは珍しくない。すなわち、ソースデバイスから宛先デバイスへのメッセージフローにはフォワード経路を用い、宛先デバイスからソースデバイスまでは異なる戻り経路を用いることができる。

【００３５】

経路は、メモリに記録するか、モニタコンピュータに記憶するか、又はモニタコンピュータによりアクセス可能である。経路記録は接続デバイスとインタフェースのセットを含む。これは、２つのエンドポイント間に配列したデバイス（ネットワークコンポーネント）の一覧の形態で提示することができる。これにより、イベント通知、報告、ＳＬＡ（サービスレベル合意）を含むネットワーク経路可用性モニタリング、故障デバイスの報告、ハイデバイスＣＰＵ、ローデバイスメモリ、ポート輻輳等、及びインパクト分析（キャパシティプランニング、「ｗｈａｔ－ｉｆ」分析）を含む事前対応型ネットワーク管理が可能となる。

【００３６】

ネットワークにより配信されるアプリケーション又はサービスとネットワークデバイス又はコンポーネント自体とのマッピングを経路の識別によって確認可能であることは顕著な利点である。これは、ネットワークの管理において大きな前進となる。

【００３７】

本発明をよりいっそう理解し、これをどのように実行し得るかを示すため、ここで一例として添付図面を参照する。

【図面の簡単な説明】

【００３８】

【図１】ネットワークの概略図である。

【図２ａ】経路発見アルゴリズムの説明図である。

【図２ｂ】経路発見アルゴリズムの説明図である。

【図２ｃ】経路発見アルゴリズムの説明図である。

【図３】経路発見アルゴリズムのフローチャートである。

【図４】１つの発見された経路を示す説明図である。

【図５】線形ルーティングテーブルの構造を示す説明図である。

【図６】宛先アドレスを複数のルートマスクと組み合わせることから得られる結果セットを示す説明図である。

【図７】ＡＲＰテーブルの構造を示す説明図である。

【図８】モニタコンピュータの概略図である。

【図９】レイヤ３ルータの概略図である。

【図１０】レイヤ２スイッチの概略図である。

【図１１ａ】モニタコンピュータで実行されるユーティリティのフローチャートである。

【図１１ｂ】モニタコンピュータで実行されるユーティリティのフローチャートである。

【図１１ｃ】モニタコンピュータで実行されるユーティリティのフローチャートである。

【図１１ｄ】モニタコンピュータで実行されるユーティリティのフローチャートである。

【図１２】ループ実行プログラムのフローチャートである。

【図１３】ループ実行プログラムの終端プロセスを示すフローチャートである。

【図１４】ループ実行プログラムのオプションＣを示すフローチャートである。

【図１５】ループ実行プログラムのオプションＳを示すフローチャートである。

【図１６】オプションＳのプロセスの続きを示すフローチャートである。

【図１７】ループ実行プログラムのオプションのプロセスを示すフローチャートである。

【図１８】図１７のプロセスの続きを示すフローチャートである。

【図１９】ループ実行プログラムのオプションｃのプロセスを示すフローチャートである

。
【図２０】オプションＡのプロセスを示すフローチャートである。
【図２１】ＶＬＡＮヒントを取得するためのプロセスを示す説明図である。
【図２２ａ】接続ポート及び接続デバイスを取得して経路記録に記憶するためのプロセスを示す説明図である。
【図２２ｂ】接続ポート及び接続デバイスを取得して経路記録に記憶するためのプロセスを示す説明図である。
【図２３】前出のオプションのいくつかで利用される、ルートを見つけるための繰り返しプロセスを示すフローチャートである。
【図２４】プライミングプロセス（ｐｒｉｍｉｎｇ　ｐｒｏｃｅｓｓ）を示す説明図である。
【図２５】プライミングプロセスを示す説明図である。
【図２６】プライミングプロセスを示す説明図である。
【図２７】繰り返しルート探索プロセスで用いられるルート探索プロセスを示すフローチャートである。
【図２８】フォーカスデバイスの転送データベースエントリを検索するためのプロセスを示す説明図である。
【発明を実施するための形態】
【００３９】
図１は、ネットワークの概略図である。このネットワークは、複数の異なる地理的位置にわたっている。各端部の地理的位置には、エンドポイントデバイス及びネットワークデバイス又はノードがある。ネットワークデバイスは、ルータ及びスイッチを含む。ネットワークのコアは、複数のネットワークデバイスを備える。ロンドンと表記した地理的位置について考えると、クライアント端末２は、エンドポイントデバイスとして機能することができる。同様に、サーバ４は、エンドポイントデバイスとして機能することができ、プリンタ６は、エンドポイントデバイスと見なすことができる。
同様のデバイスが、異なるレイアウトのパリ及びニューヨークの地理的位置に示されている（ニューヨークにはサーバファーム又はデータセンタが示されている）。ニューヨークの位置では、複数のサーバ８が重要なアプリケーション又はサーバエンドポイントデバイスを表すことに留意すべきである。
【００４０】
図１に示すネットワークは、一例として与えていることを理解すべきである。使用可能なネットワークが多種多様に存在し、本発明は、いかなる相互接続デバイスのネットワークにおいても使用可能である。特に、エンドポイントデバイス及び特定のネットワークデバイス又はノードの性質は様々に変動し得る。開示されている特定のネットワークでは、ネットワークデバイスは、レイヤ３又はレイヤ２のデバイスとすることができる。
【００４１】
開放型システム間相互接続（ＯＳＩ：Ｏｐｅｎ　Ｓｙｓｔｅｍｓ　Ｉｎｔｅｒｃｏｎｎｅｃｔｉｏｎ）モデルは、通信システムのプロトコルを特徴付けることができる７つのレイヤを定義している。ここで記載される経路発見アルゴリズム（ｐａｔｈ　ｆｉｎｄｉｎｇ　ａｌｇｏｒｉｔｈｍ）は、レイヤ２及びレイヤ３において利用可能な情報を用いてネットワーク経路を計算する。
【００４２】
レイヤ２（データリンクレイヤ）で動作するデバイスは、直接隣接したデバイスの情報を有し、あるレイヤ２デバイスから次のレイヤ２デバイスへのパケットの取得（レイヤ２ＭＡＣ（メディアアクセス制御）アドレスに基づいて）を担っている。
【００４３】
レイヤ３（ネットワークレイヤ）で動作するデバイスは、ネットワーク内のあるポイントからネットワーク内の別のポイント（多くの場合、数十又は数百のデバイスを隔てている）へのパケット伝搬を担っている。

所定のレイヤ３経路にどのデバイスが参加するべきか（本明細書では「レイヤ３ホップ」と称する）を計算するため、レイヤ３デバイスは、ルーティング情報を交換し、ルーティングプロトコルを用いて最も望ましい経路（複数の経路）を計算する。経路内の連続したレイヤ３デバイス間でパケットを渡すために、レイヤ２で動作しているデバイスが用いられる。多くの場合、各レイヤ３デバイス間には複数のレイヤ２デバイスが存在する（本明細書では「レイヤ２ホップ」と称する）。

【００４４】

したがって、大きいネットワークは、効果的に複数のセグメントに分割され、その各々は通常、レイヤ３デバイスにより接続された複数のレイヤ２デバイスを含んでいる。

【００４５】

図９は、レイヤ３ルーティングデバイスの概略図である。このデバイスは、例えば制御コード、ファームウェアを実行するマイクロプロセッサ、又は他の何れかの適切な実施の形態のコントローラ９０を備えている。コントローラ９０は、後に図５を参照して更に詳しく検討するルーティングテーブル９２にアクセスすることができる。

レイヤ３ルーティングデバイスは、ポートＰｉ／Ｐｏを有する。各ポートは、図１のネットワークに示すように、物理リンクに接続されている。この表記において、Ｐｉは「入口」ポートを示し、Ｐｏは「出口」ポートを示す。これは表記の便宜上のものであり、実際には、通常デバイスは入口ポート又は出口ポートとして専用のポートを有するわけではなく、それらが入口であるか出口であるかは、その時に転送しているデータに依存する。多くのポートはいつでも出口及び入口として機能する。

【００４６】

入口ポートＰｉに到達するパケットは、例えばバス９４を介してコントローラ９０により読み取られるＩＰ（インターネットプロトコルアドレス）のような宛先識別子を有し得る。コントローラ９０は、ルーティングテーブル９２にアクセスし、そこから導出された情報に基づいて、着信パケットを送り出すルーティングスイッチ９６を制御する。

次に、ルーティングスイッチ９６は、ルーティングテーブル内の情報に応じて、着信パケットを適切な出口ポートＰｏにルーティングする。このルーティングデバイスは、以降のルーティングのためにレイヤ３アドレスをレイヤ２のアドレスに割り振るマッピングテーブル９１を含む。

このようなルーティングデバイスの動作は、当技術分野において既知であるので、本明細書でこれ以上は説明しない。この状況では、モニタコンピュータからリンクを介して入口ポートＰｉに到達したパケットをコントローラ９０で傍受（ｉｎｔｅｒｃｅｐｔ）することによって、そのようなパケットによりルーティングテーブルのクエリが実行され得ることに留意すべきである。そのようなクエリパケットは、更なるルーティングのためルーティングスイッチ９６に供給されるのではなく応答を発生し、この応答がルーティングデバイスから出力され、出口ポートからネットワークを介して問い合わせ側の対象物に戻される。

この場合、その問い合わせ側の対象物は、モニタコンピュータ１６である。ネットワークを介して伝達される全てのパケット（クエリパケットを含む）は、ソース及び宛先アドレスを含む。すなわちクエリパケットは、モニタコンピュータに対応したソースアドレスと、クエリが実行されるデバイスに対応した宛先アドレスと、を有する。応答を送信する必要がある場合、ソースアドレス及び宛先アドレスを交換して、ソースアドレスをクエリ実行対象デバイスとすると共に、宛先アドレスをモニタコンピュータとする。

【００４７】

図１０は、レイヤ２スイッチをより図式化した説明図である。レイヤ３ルーティングデバイスと同様に、レイヤ２スイッチは、ポートＰｉ／Ｐｏを有し、その各々は、例えば図１のネットワークに示すように、物理リンクに接続されている。上述のように、ポートは通常、入力専用又は出力専用ではない。入口ポートＰｉの着信パケットは、スイッチ１００に送り出される。スイッチ１００は、パケット内の宛先識別子（通常はヘッダ）に基づいてどのようにパケットをルーティングするかを決定するため、レイヤ２転送データベー

ス（ＦＤＢ）１０２にアクセスすることができる。レイヤ２転送データベースは、着信パケットの識別子を、このパケットを転送すべき出口ポートに割り振る。

既に説明したように、ＯＳＩモデルによれば、レイヤ３ルーティングデバイスの識別子は、ＩＰアドレスであり、レイヤ２デバイスの識別子は、ＭＡＣアドレスである。

【００４８】

レイヤ３デバイスの場合と同様に、レイヤ２は当技術分野において既知であり、したがって、本明細書でこれ以上は検討しない。しかしながら、レイヤ３デバイスと同様に、更にまた、それらは入口ポートＰｉでパケット内のクエリを受信し、出口ポートＰｏにおけるレイヤ２スイッチからの出力に、そのクエリに対する応答を発生できることに留意すべきである。したがって、クエリパケット自体は、スイッチにおいてルーティングされるのではなく、クエリ側のデバイス（この場合はモニタコンピュータ１６）に返される応答を発生する。

【００４９】

スイッチにおけるスイッチコントローラ１０１は、トラフィックの転送及び応答の発生を担っている。

【００５０】

更に最新のデバイスの中には、レイヤ３及びレイヤ２の機能を実行可能なものがある。

【００５１】

以下に記載する本発明の実施形態は、所定のソースデバイスと所定の宛先デバイスとの間でメッセージフローが進む経路を識別する方法を提供する。例えばエンドポイントＸをソースデバイスと考え、エンドポイントＹを宛先デバイスと考えることができる。図１のネットワークを見ると、既に述べたように、何れかの所定時間かつ所定の環境条件のセットのもとで、それらのエンドポイント間でネットワーク内のどの経路を採用するかを確定することは決して些細なタスクではない。

図１は、そのような経路を発見し記録することができる経路発見プログラムを実行するモニタコンピュータ１６を示す。図８は、モニタコンピュータ１６をより図式化した説明である。コンピュータ１６は、マイクロプロセッサ８０を備え、これは、このプロセッサにより実行されるコードが記憶されたメモリ８２にアクセスすることができる。この場合、そのコードは、経路発見プログラムを含む。メモリ８２は、経路発見プログラムにより生成された経路記録８１も記憶している。コンピュータは、ユーザインタフェース（ＵＩ）８４を有し、これは、マウス又はキーボード等のユーザ入力デバイスと、ユーザに対して情報を表示するためのディスプレイと、を含むことができる。特に、本明細書において更に詳しく検討するように、ユーザインタフェース８４において、経路発見プログラムの後の警告（アラート）又は経路発見プログラムに関する情報をユーザに表示することができる。図２ａから図２ｃは、経路のステップを示す。次にこれらについて説明する。

【００５２】

高水準のアルゴリズムは「フォーカスデバイス」の概念を用いる。フォーカスデバイスは、仮想パケットを次にどこに送信するか（すなわち、どのインタフェースから仮想パケットを送信するか）に関して現在クエリが実行されているデバイスである。アルゴリズムは、ソースデバイスから開始し、各フォーカスデバイスを順番に評価することによって、ターミナルデバイス（すなわちパケットの最終的な宛先）へと進む。

デバイスがレイヤ３で動作している場合、レイヤ３のネクストホップ（ＮＨＬ３）行きのパケットを送信するためにどのインタフェース（出口ポート）を用いるかについてクエリが実行される。デバイスがレイヤ２で動作している場合、次レイヤ３ホップのレイヤ２（ＭＡＣ）アドレス（ＮＨＬ２）行きのパケットを送信するためにどのインタフェース（出口ポート）を用いるかについてクエリが実行される。フォーカスデバイスの応答をネットワークトポロジと関連付けて用いて、経路内の次のデバイスを決定することができる。このようにして、アルゴリズムは、レイヤ２デバイスを用いながらレイヤ３経路に沿って動作して、連続したレイヤ３ノード間をナビゲートする。

【００５３】

主要なアルゴリズムを開始する前に、ソースデバイス及びターミナルデバイスの位置を特定する。これは単純でない場合があり、これを達成するための技法について後に検討する。

【００５４】

主要なアルゴリズムに従って第１ホップの位置を特定する。経路をシードし（ｓｅｅｄ）、ループカウントをゼロに設定する。ループリミットが、経路識別ループ（後に検討する）を実行する回数を定める。

【００５５】

[レイヤ３における第１ホップの探索]

レイヤ３における最初のネクストホップ（ソースデバイスからのネクストホップ）（ＮＨＬ３）を見つけることで、第１ホップの位置を特定する。以下の説明では「クエリ」という言葉を頻繁に用いる。クエリは、後で更に詳しく説明するように発生され構築される。クエリの目的は、クエリがアドレス指定されるフォーカスデバイスから、ネクストホップアドレス及び出口ポートの位置を特定することである。

最初のＮＨＬ３アドレスは、まず宛先ＩＰアドレスを用いてソースデバイスＸにクエリを実行することで決定できる。すなわち、ソースデバイスのルーティングテーブルにＮＨＬ３及び出口ポートについてクエリを実行する試みが行われる。

ルートが見つからず、かつソースデバイスがレイヤ３アクセススイッチを有する場合、宛先ＩＰアドレスを用いて、このレイヤ３アクセススイッチにＮＨＬ３についてのクエリを実行する。これが成功しない場合、ＮＨＬ３を確認するためにソースデバイスのデフォルトゲートウェイにクエリを実行する。これが成功しない場合、宛先ＩＰアドレスを用いてアクセススイッチにデフォルトゲートウェイについてのクエリを実行する。ＮＨＬ３アドレスが見つからない場合、これは障害と見なされるが、アルゴリズムが失敗したことを意味するのではなく、経路内の障害点がこのポイントで識別されたことを意味する。あるいは、ＮＨＬ３が見つからなかった他の理由がある場合もある。

【００５６】

[経路のシード（ｓｅｅｄ）]

経路をシードするため、ソースデバイスの位置が特定されたら、これを経路に追加する。すなわち、ソースデバイスの出口インタフェースの位置が特定され経路に追加される。ソースデバイスのルーティングテーブルからＮＨＬ３が見つかった場合、このＮＨＬ３アドレスに対するソースデバイス出口インタフェースは、経路に追加される。後に説明するように、レイヤ３アドレス（ＮＨＬ３）に対応したレイヤ２アドレス（ＮＨＬ２）を確認することができる。ソースデバイスのルーティングテーブルからＮＨＬ３の出口ポートが見つからない場合、ＮＨＬ２についてのソースデバイスのレイヤ２転送テーブルは、出口ポートを見つけるために用いられる。出口ポートが見つかったら、その出口ポートを経路に追加する。

【００５７】

[経路発見アルゴリズムの概要]

図２ａにおいて、モニタコンピュータ１６からソースデバイスＸにディスパッチされたクエリは、直接の矢印として示されるが、実際にこれを実施するには、図１のネットワークにおいてモニタコンピュータ１６がソースデバイスＸにアドレス指定されたメッセージ又はパケットを発行すればよい。

上述したように、クエリは、宛先ポイントＹのレイヤ３アドレスであるターミナルＩＰ（宛先ＩＰ）のためのネクストホップＩＰ（及び出口ポート）についてソースデバイスに質問する。その目的は、ソースデバイスＸに、ＮＨＬ３及びＮＨＬ３のための出口ポート（ターミナルＩＰアドレス）を含む応答を提供させることである（図３のステップＳ１及び図２ａを参照）。

【００５８】

上述したように、ソースデバイスが必要な情報を供給できない状況があり得る。第１の「フォーカス」デバイスを取得するための上述した他の可能性は、接続されたアクセススス

イッチに対してレイヤ３ルーティング情報についてのクエリを実行すること（アクセススイッチがレイヤ３スイッチである場合）を含み、これが失敗した場合、アルゴリズムは、接続されたアクセススイッチに対して、デフォルトゲートウェイ及び第１のＮＨＬ３として用いられるデフォルトゲートウェイのＩＰアドレスについてのクエリを実行する。

【００５９】

ステップＳ２では、ＮＨＬ３アドレスからネクストホップレイヤ２（ＭＡＣ）アドレスを決定し、ＮＨＬ２をこのＭＡＣアドレスに設定する。これは、Ｌ３アドレスを Ｌ２アドレスに割り振るマッピングテーブル９１にクエリを実行することで達成可能である。そのようなマッピングテーブルの１つは、ＡＲＰテーブルである（他には「直接マッピング」及び近隣探索が含まれる）。これは、後述のＡＲＰクエリを用いた、ソースデバイスＡＲＰ、次Ｌ３ホップデバイスＡＲＰ、又はグローバルキャッシュＡＲＰであり得る。

ステップＳ１では、識別した出口ポートを経路記録に追加する（Ｓ１Ａ）。ステップＳ３では、キャッシュしたエンドホスト位置（スイッチＣＡＭクエリからの）を用いて開始ネットワークスイッチ（及びポート）を見つけ、フォーカスデバイスとして設定する。

ステップＳ４では、キャッシュしたエンドホスト位置（スイッチＣＡＭクエリからの）を用いてターミナルネットワークスイッチを見つける。開始スイッチは、経路記録に追加される。

【００６０】

本方法は、現時点で、経路識別ループに入る準備ができた状態にある。ステップＳ５では、ＮＨＬ２が既知であるか否かを判定する。既知である場合、ループは、ステップＳ５Ａに進む。既知でない場合、プロセスは、ステップＳ５Ｂを実行して、フォーカスデバイス又はＮＨＬ３デバイスにおけるＡＲＰクエリによってＮＨＬ２を決定する。

図７を参照して、レイヤ３アドレスをレイヤ２アドレスに相関付けるためのクエリの発生について更に詳しく検討する。簡単に言うと、クエリを実行するデバイスについて、ネットワークトポロジから、又はデバイス自体のインタフェーステーブルからインタフェースインデックス（ｉｆＩｎｄｅｘ）をウォークする（ｗａｌｋ）ことによって、インデックスのリストが取得される。デバイスの各ｉｆＩｎｄｅｘは、デバイスへのクエリに含めるためのキーのセットを生成するために、ＮＨＬ３アドレスと組み合わされる。これにより、これらのキーを含むクエリが構築され、フォーカスデバイスに送信される。フォーカスデバイスは、ゼロ又は成功応答（１）を生成する。

【００６１】

ＮＨＬ２を決定するための上述の２つの技法が失敗した場合、グローバルＡＲＰは、アクセスされる。ステップＳ５Ａでは、アドレスＮＨＬ３が現在のフォーカスデバイス上にあるか否かを判定する。

【００６２】

ＮＨＬ３が現在のデバイス上にない場合、ステップＳ６において、プロセスは、出口ポートを得るためＮＨＬ２についてのレイヤ２ＦＤＢエントリを見つけるためのクエリをディスパッチする。レイヤ２でのクエリの発生については、後述する。これが成功した場合、出口ポートを経路記録に追加し（Ｓ６Ａ）、キャッシュしたトポロジ３を用いてリンク終端のポート及びデバイスを見つけ（Ｓ７）、デバイスを経路に追加し（Ｓ７Ａ）、リンク端部で位置特定したばかりのデバイスをフォーカスデバイスに設定する（Ｌ２ホップ）。ステップＳ６Ａ、Ｓ７、及びＳ７ＡをＬ２ホップと称することがあり得る。この点については図２ｂを参照のこと。

ステップＳ５Ａでは、フォーカスデバイスはデバイスＡである。これは、レイヤ２ＦＤＢエントリを見つけるためのクエリを受信し、出口ポートを戻す。そのリンク端部にあると判定されたデバイスは、デバイスＢであり（図２ｃ）、これは、依然として宛先ＩＰアドレスに設定されているＮＨＬ３のクエリを受信する。

【００６３】

レイヤ２ＦＤＢエントリが見つからなかった場合、又はＳ５ＡにおいてＮＨＬ３がフォーカスデバイス上でホストされていると判定された場合、ステップＳ８でルートクエリを

実行して、フォーカスデバイス上で宛先ＩＰアドレスへのＬ３ルートが見つかったか否かを判定する。ルートクエリは、単一ルート又は再帰的ルートクエリであり得る。これらについては後に説明する。これは、ネクストホップＩＰ及び出口インタフェースを確定する。Ｌ３ルートが見つからない場合、壊れた経路が示され、プロセスは停止する（Ｓ８Ａ）。

【００６４】

ステップＳ９（Ｌ３ホップ）では、ルーティングテーブル出口インタフェースを経路に追加し、ＮＨＬ３を新しいネクストホップＩＰアドレスに設定し、プロセスはデバイスに対しクエリを実行してＮＨＬ３のレイヤ２アドレスを確認する。ＮＨＬ２を決定できない場合、ＮＨＬ２を「未知」に設定する。

ステップＳ１０では、現在のＮＨＬ３アドレスを宛先ＩＰアドレスと比較する。ＮＨＬ３が宛先ＩＰでない（すなわち経路発見アルゴリズムがまだ最終Ｌ２セグメント上にない）場合、ステップＳ１１では、キャッシュしたトポロジを用いてリンク端部のポート及びデバイスを見つけ、デバイスを経路記録に追加し、このデバイスをフォーカスに設定する。次に、プロセスは、フォーカスデバイスがターミナルデバイスであるか問い合わせる（Ｓ１２）。フォーカスデバイスがターミナルデバイスでない場合、プロセスはステップＳ５に戻るが、ステップ９で設定したＮＨＬ３及びＮＨＬ２を用いる。

【００６５】

[終端（ｔｅｒｍｉｎａｔｉｏｎ）]

ターミナルデバイスに到達し、ターミナルポート及び宛先サーバを経路に追加すると、アルゴリズムは終了する。他の終端条件により、アルゴリズムのループ処理を無期限に阻止する。経路の各繰り返しでは、スイッチフラグを偽に、ルートフラグを偽に設定することで繰り返しが開始する。Ｌ２ホップが行われると（Ｓ７）、スイッチフラグは真に設定される。Ｌ３ホップが行われると（Ｓ９）、ルートフラグは真に設定される。

既に述べたように、出口ポートをフォーカスデバイスから決定し、ネットワークトポロジを用いて結合されたデバイス（ａｔｔａｃｈｅｄ　ｄｅｖｉｃｅ）及び結合されたデバイスの入口ポートを見つける。各繰り返しについて、「フォーカスデバイス、ＮＨＬ２、ＮＨＬ３」の組み合わせを記憶する。

【００６６】

フォーカスデバイス、ＮＨＬ２、又はＮＨＬ３が変化し、「フォーカスデバイス、ＮＨＬ２、ＮＨＬ３」の新しい組み合わせが見つかった場合、ループ検出イベントを生成し、ループを停止する。ループ限度に到達しておらず、ルーティング又はスイッチングの何れかが実行され（すなわちルートフラグ又はスイッチフラグが真である）、更に、フォーカスデバイスがターミナルデバイスに等しくない場合、再び繰り返しを行う。繰り返しのたびに、繰り返しループ限度に到達したか否かを査定する。到達した場合、アルゴリズムは終端する。

【００６７】

繰り返しが終了すると、フォーカスデバイスがターミナルデバイスである場合は、ターミナルデバイスは、経路に追加される。フォーカスデバイスがターミナルデバイスでないがアルゴリズムが停止した場合、経路発見アルゴリズムは予想外の位置で終端するのでエラーが報告される。ターミナルデバイスがアクセススイッチである場合、アクセススイッチ出口ポートを「宛先の位置特定」（Ｓ４）から経路に追加し、アクセススイッチ出口ポートから導出された宛先デバイスを経路に追加する。そして、アルゴリズムは終了する。ターミナルデバイスが宛先デバイスと等しい場合、アルゴリズムは終了する。次に、アルゴリズムの詳細について更に詳しく検討する。

【００６８】

[具体例]

図４は、経路発見アルゴリズムの動作の１つの結果を示す。すなわちこれは、経路発見アルゴリズムがネットワークにクエリを実行した時に、宛先デバイスＹにアドレス指定されたソースデバイスＸからのデータパケットがネットワーク上で進むルートを提供する。

経路は、経路記録の一部を形成するデバイスＡ～Ｊを含むように示されている。経路記録は、それらのデバイスの各々からの入口ポート及び出口ポートを含む。

【００６９】

再び、図１の元のネットワークを見ると、図４に示す経路記録の第１の部分は図１のネットワークから導出されたと考えられる。対応する文字を用いて、前のスイッチ又はルーティングデバイスにより選択されるデバイスを示す。経路発見アルゴリズムが動作した時、ルーティングデバイスＢは、パケットをスイッチＣに送信すると決定している。

しかしながら、本発明を用いない場合、これをリアルタイムで達成することは極めて難しい。ルーティングデバイスＢには同様に、パケットをコアネットワーク内のルータＦにルーティングするオプションがあった。宛先Ｙにアドレス指定された仮想パケットに基づいて、リアルタイムで（又はある程度はリアルタイムで）ルーティングデバイスＢにクエリを実行することにより、ルーティングデバイスＢは、そのアドレスを有する実パケットが到達した場合に行われていたはずの決定を戻す。

ルーティングデバイスＢがパケットをスイッチＣにディスパッチすると確認し、それから、スイッチＣがその出口ポートの遠端でルーティングデバイスＤに接続していると確定することにより、Ｃ及びＤは経路記録８１に追加されている。

このように、パケット識別アルゴリズムは、ネットワーク内のデバイスにクエリを実行した時に仮想パケットが採用する経路内を１つずつ進んでいく。ルーティングデバイスＤに隣接した囲み枠は、ＮＨＬ３及びＮＨＬ２の設定を示す。

すなわち、ＮＨＬ３は、現在ルーティングデバイスＤでアクティブであるルーティングテーブルに基づいてデバイスＤの遠端デバイスとして確定されているデバイスＥのＩＰアドレスに設定されている。ＮＨＬ２は、デバイスＥのＡＲＰエントリについてデバイスＤにクエリを実行することにより、デバイスＥのＭＡＣアドレスとして確定されている。

【００７０】

[ネットワークトポロジ]

上述したように、ネットワークトポロジは、ネットワークデバイス相互接続性及びエンドホスト位置の双方を含む。ネットワークトポロジ３は、ポートツーポート（ｐｏｒｔ－ｔｏ－ｐｏｒｔ）接続の詳細を与えるトポロジサーバによって提供することができる。このため、あるデバイスにおいて出口ポートが識別されると、トポロジで識別されるポートツーポート接続を用いて、接続されたデバイスの入口ポートを確認することができる。

出口ポート及び入口ポートの双方を経路記録に追加することができる。トポロジサーバは、グローバルＣＡＭ、グローバルＡＲＰ、及びデバイス認証情報も提供する。更に、トポロジに記録された各デバイスについて、インタフェースインデックス（ＩｆＩｎｄｅｘ）リスト及びＶＬＡＮ（仮想ローカルエリアネットワーク）リストがあることが好ましい。ＶＬＡＮデバイスについては、まだ検討していない。それらについては本明細書で更に検討する。

モニタコンピュータ１６に応答が戻されると、モニタコンピュータは、処理レイヤ２が応答した場合に以下の順序でトポロジ３にクエリを実行する。この状況では、レイヤ２応答は、レイヤ２スイッチデバイスからの出口ポートを識別した応答である。クエリ実行の順序は、ＣＤＰ、ＬＬＤＰ、ＳＴＰ、及びＳＯＮＭＰ、ＩＰｖ６　ＮＤである。

【００７１】

[ソースデバイスの位置特定]

上述したように、経路内の第１のデバイス（ソースデバイスに接続されたデバイス）の位置特定は、必ずしも単純ではない。一実施形態では、モニタコンピュータ１６は、アルゴリズムを実施して、まず接続されたホストとしてソースを見つけようとし、これが失敗するとネットワークデバイスとしてソースを見つけようとする。

接続されたホストとしてソースを見つけようとする場合、ソースＩＰのレイヤ２（ＭＡＣ）アドレスについて、ソースデバイスにクエリを実行する。これは、ステップＳ５Ｂで上述したようなフォーカスデバイスに対するクエリと同じ方法で達成可能である。すなわち、プロセスは、ソースＩＰアドレスについてのＡＲＰエントリを見つけるためのクエリ

をディスパッチする。

【００７２】

ソースデバイスからのレイヤ２アドレスがない場合、トポロジサーバにおけるグローバルキャッシュＡＲＰテーブルにクエリを実行する。上述の実施形態では、これらはＡＲＰテーブルとされるが、レイヤ３アドレスをレイヤ２アドレスに割り振るいかなるテーブルも利用可能である。

ソースＩＰアドレスに対応するＭＡＣアドレスが見つかった場合、このＭＡＣアドレスからのトラフィックを認識したポートを見つけるためにトポロジサーバのグローバルキャッシュレイヤ２転送テーブルにクエリを実行することで、ソースＩＰのＭＡＣ位置についてトポロジサーバにクエリを実行する。

トポロジサーバは、複数の一致（ソースＭＡＣが複数のポート上で認識された）を除去すること、トランクとフラグされたポート、過剰な数のＭＡＣを有するポート（アクセススイッチポートのＦＤＢエントリは典型的に単一の「認識された（ｓｅｅｎ）」ＭＡＣアドレスを有する）、ネットワーク間トポロジを有するポート（例えばポートがＣＤＰ隣接情報を有する場合、これはアクセススイッチ上のポートではあり得ない）等を取り除くことによって、一意のソースＭＡＣ位置を戻すことが予想される。

【００７３】

接続されたホストとしてソースが見つからない場合、ネットワークデバイスとしてソースを見つける試みを行う。これは、全ての管理されたネットワークデバイス上で見つかる全てのＩＰアドレスについてトポロジサーバにクエリを実行して、ＩＰアドレスがネットワークデバイス上にあるか否かを調べることにより達成可能である。そうである場合、そのネットワークデバイスをフォーカスデバイスとして設定する。

【００７４】

[宛先デバイスの位置特定]

同様の考察が宛先デバイスの位置特定にも当てはまる。まず、接続されたホストとして宛先デバイスを見つける試みを行う。これが失敗すると、ネットワークデバイスとして宛先を見つける試みを行う。接続されたホストとして宛先デバイスを見つけるため、そのレイヤ２アドレスについて宛先デバイスにクエリを実行し、又はトポロジサーバにおいてグローバルキャッシュレイヤ３－レイヤ２マッピングテーブルにクエリを実行する（上述のソースデバイスと同様）。次に、トポロジサーバ上のグローバルキャッシュレイヤ２転送テーブルにクエリを実行して、このＭＡＣからのトラフィックを認識したポートを見つける（これもソースデバイス位置に関しての説明と同様）。

【００７５】

上記が失敗した場合にネットワークデバイスとして宛先を見つけるため、全ての管理されたデバイス上で見つかる全てのＩＰアドレスについてトポロジサーバにクエリを実行して、ＩＰアドレスがネットワークデバイス上にあるか否かを調べればよい。次に、ネットワークをターミナルデバイスとして設定することができる。

【００７６】

[ホップごとのユーティリティ]

経路発見アルゴリズムを実施するため、モニタコンピュータ１６は、上述したようなコンピュータプログラムを実行する。このコンピュータプログラムは、「ホップごとの（ｐｅｒ　ｈｏｐ）」クエリを処理するユーティリティを提供する。すなわち、経路発見アルゴリズムは、モニタコンピュータからフォーカスデバイスにクエリをディスパッチし、トポロジへのアクセスに使用可能な出口ポートをフォーカスデバイスから受信することに頼っている。これは、必ずしも単一のクエリによって達成することはできない。

上述のように、アルゴリズムは、レイヤ３における最初のネクストホップを必要とする（ＮＨＬ３）。ユーティリティは、宛先ＩＰアドレスを用いて、ＮＨＬ３及び出口ポートについてソースデバイス上のルーティングテーブルに対するクエリの実行を試みる。ルートが見つからなければ、アクセススイッチがレイヤ３スイッチである場合、このアクセススイッチ上のルーティングテーブルにクエリを実行する（これはＮＨＬ３についてソース

デバイスに接続された第１のデバイスである）。
ここで、ルートが見つからない場合、ＮＨＬ３のデフォルトゲートウェイについてソースデバイスにクエリを実行する。ルートが見つからない場合、デフォルトゲートウェイについて第１のデバイスにクエリを実行する。
（上述したように）ルーティングテーブルにクエリを実行してＮＨＬ３を見つけるため、後に説明する推論的なキーイング技法を用いてルーティングデバイスにクエリを実行することによって、当該ＩＰアドレス（「探索対象の」ＩＰアドレス）のためのルートを見つける。
ルートが見つかったが、出口ポートが特定されない場合、ネクストホップＩＰアドレスを戻し、ＮＨＬ３として用いる。ルートが見つかり、出口インタフェースｉｆＩｎｄｅｘがゼロより大きい場合、出口ポートをＮＨＬ３アドレスと共に戻し、出口ポートは、経路に追加される。ルートが見つかり、出口インタフェースｉｆＩｎｄｅｘがゼロに等しい場合、ユーティリティは、探索対象のＩＰを（前のクエリからの）ネクストホップＩＰに設定し、（後に説明するような）推論的なキーイングを用いてデバイスにクエリを実行して探索対象のＩＰのためのルートを見つけることによって、何度も繰り返す。これは、戻されるｉｆＩｎｄｅｘがゼロ以外の値になるまで繰り返される。
【００７７】
探索対象のＩＰのためのルートを見つけるステップは、推論的なキーイング技法を用いてルートエントリを戻す。ルートエントリが見つかると、ユーティリティは、ｉｐＲｏｕｔｅＮｅｘｔＨｏｐ．ＮｅｔｗｏｒｋＡｄｄｒｅｓｓからのネクストホップアドレスに対してポーリング（例えば、通信回線を共有する各端末に順次問い合わせて端末を特定すること）する。ユーティリティは、ｉｐＲｏｕｔｅＩｆｉｎｄｅｘ．ＮｅｔｗｏｒｋＡｄｄｒｅｓｓからの出口インタフェースに対して、及びｉｐＲｏｕｔｅＴｙｐｅ．ＮｅｔｗｏｒｋＡｄｄｒｅｓｓに対してもポーリングする。ｉｐＲｏｕｔｅＴｙｐｅが「直接（ｄｉｒｅｃｔ）」である場合、探索対象のＩＰをネクストホップに設定する。ＩＰルートタイプ「直接」は、これがネットワークセグメントに直接接続されていることを示すからである。
【００７８】
デバイス上のルーティングテーブルから複数の一致が戻される可能性がある。この場合、例えばデバイスがトラフィックのロードバランシングを担っている場合、複数のルートが用いられているか否かを判定することが適切である。単一のルートのみがアクティブに用いられている場合、このアクティブなルートが決定されるはずである。
複数のルートが用いられている場合、この時点で経路を分割することができ、経路記録は、この時点以降に見つかる全てのルートに適用される経路発見アルゴリズムの結果を含むことができる。
多くの場合、デバイスにおけるルーティングのための複数のオプションは、デバイスが様々な測定基準（メトリクス）に基づいてインテリジェントにルーティングされることを示している。これらの測定基準にもクエリを実行し、モニタコンピュータでの記録のために戻すことができる。
【００７９】
また、ユーティリティは、フォーカスデバイスにおいてレイヤ３－レイヤ２マッピングテーブル９１にクエリを実行して、レイヤ２における最初のネクストホップ探索を担っている。
レイヤ２アドレスが見つからず、フォーカスデバイスがソースデバイスである場合、ユーティリティは、アクセススイッチにクエリを実行する（これがレイヤ３スイッチである場合、レイヤ３からレイヤ２へのマッピングを提供するはずである）。レイヤ２アドレスが見つからない場合、ユーティリティは、トポロジサーバ３上のグローバルキャッシュＡＲＰテーブルにクエリを実行する。デバイス上のレイヤ２アドレスについてのクエリは、ステップＳ５Ｂを参照して上述したように実行される。
【００８０】

ＮＨＬ３アドレスがフォーカスデバイス上にない場合、ユーティリティは、レイヤ２アドレスＮＨＬ２のための出口ポートについてフォーカスデバイスに対してポーリングする。出口ポートＮＨＬ２についてフォーカスデバイスに対してポーリングするステップは、ＶＬＡＮ（仮想ローカルエリアネットワーク）に特定的なポーリングを含む。

すなわち、これは、デバイスが、トポロジ３に従って、デバイスにおける記録どおりに、どのＶＬＡＮに参加しているかを確定するステップを含む。これらのＶＬＡＮは、特定のＶＬＡＮについての転送テーブルのエントリの探索に役立てるために用いられる（ＦＤＢは多くの場合、どのＶＬＡＮに関連しているかに従って分割される。例えば、ＶＬＡＮごとのスパニングツリープロトコル（ＰＶＳＴＰ：Ｐｅｒ　ＶＬＡＮ　Ｓｐａｎｎｉｎｇ　Ｔｒｅｅ　Ｐｒｏｔｏｃｏｌ）では、各ＶＬＡＮが一致を見つけようとする状況でＦＤＢクエリを実行する必要がある）。

【００８１】

（特定的なＶＬＡＮ又はネイティブＶＬＡＮを用いて）出口ポートがレイヤ２ＦＤＢから見つからない場合、ユーティリティは、ｉｐＮｅｔＴｏＭｅｄｉａＰｈｙｓＡｄｄｒｅｓｓ７１に対するポーリングにより、どのインタフェースがＡＲＰ記録からＮＨＬ２に向かうかを見つけようとする（図７）。すなわち、ユーティリティは、どのインタフェースからレイヤ２とレイヤ３との関係が学習されたかを学ぼうとする。

【００８２】

ユーティリティは、レイヤ２アドレスを用いて出口ポートを見つけたら、この出口ポートを経路記録に追加し、トポロジサーバ３を用いて出口ポートに結合されたリモートポートを見つける。このリモートポートは、次のデバイス上の入口ポートとして記録される。

【００８３】

[ポートチャネル／多重化ポート]

リモートポートが見つからない場合、又は出口ポート名が上位層もしくは下位層レイヤのポートの使用を命令する場合、ユーティリティは、下位層レイヤのポート又は上位層レイヤのポートをチェックする。すなわち、仮想経路出力が物理ポートにマッピングされる状況があり得る。経路発見アルゴリズムを成功させるため、トポロジサーバにアクセスするための物理出口ポートを識別する必要がある。

下位層レイヤのポートのチェックによって下位層レイヤのポートの存在が明らかになった場合、これらの下位層レイヤのポートを出口ポートとして用いることができる。トポロジサーバにアクセスして、出口ポートに結合されたリモートポート（次のデバイスの入口ポート）を見つける。この時点で、経路は複数の別個の経路に分割され、それらの各々はこの時点から別々にトレースされる。

【００８４】

上位層レイヤのポートが識別された場合、上位層レイヤのポートを出口ポートに用いる。トポロジサーバを用いて、この上位層レイヤの出口ポートに結合されたリモートポートを見つける。

【００８５】

[ネクストホップ]

ルートフラグ及びスイッチフラグフラッグを偽に設定する。トポロジサーバ又はフォーカスデバイスに対する直接クエリを用いて、フォーカスデバイスがそのポートの何れかでＮＨＬ３　ＩＰアドレスをホストするか否かを確認する。これがＮＨＬ３　ＩＰアドレスをホストする場合、ユーティリティは続いて、推論的なキーイング技法を用いることにより、宛先ＩＰへのルートについてフォーカスデバイスルーティングテーブルにクエリを実行する。

ユーティリティが候補ルートの位置を特定したら、レイヤ３からレイヤ２へのマッピングについてフォーカスデバイス（又はグローバルキャッシュＡＲＰテーブル）にクエリを実行することで、次のレイヤ２アドレスＮＨＬ２を設定し、ルートフラグを真に設定する。ＮＨＬ３が宛先ＩＰと等しい場合、これは、ユーティリティが宛先に最も近い最後のレイヤ３デバイスに到達したことを示すので、ネクストホップはレイヤ２ホップであり、も

うこのデバイスを進める必要はない。
したがって、ユーティリティは候補ルートの出口ポートを経路に追加する。ＮＨＬ３が宛先ＩＰと等しくない場合、これが最終的なレイヤ２セグメントでないことが示され、候補ルートの出口ポートを経路に追加する。
【００８６】
この繰り返しの間にルーティングが行われなかった（ルートフラグがまだ偽のままである）場合、ユーティリティは、レイヤ２アドレスＮＨＬ２のための出口ポートについてフォーカスデバイスに対してポーリングする。出口ポートＮＨＬ２についてフォーカスデバイスに対してポーリングするステップは、ＶＬＡＮ（仮想ローカルエリアネットワーク）に特定的なポーリング（上述のような）を含む。
（特定的なＶＬＡＮ又はネイティブのＶＡＬＮを用いて）レイヤ２ＦＤＢから出口ポートが見つからない場合、ユーティリティは、ｉｐＮｅｔＴｏＭｅｄｉａＰｈｙｓＡｄｄｒｅｓｓ７１に対してポーリングすることによりＡＲＰ記録からどのインタフェースがＮＨＬ２に向かうかを見つけようとする。
すなわち、ユーティリティは、どのインタフェースからレイヤ２とレイヤ３の関係が学習されたかを学ぼうとする。
ユーティリティは、レイヤ２アドレスを用いて出口ポートを見つけたら、この出口ポートを経路記録に追加し、トポロジサーバ３を用いて出口ポートに結合されたリモートポートを見つける。このリモートポートは、次のデバイス上の入口ポートとして記録される。ＦＤＢクエリ又はＡＲＰクエリを用いて出口ポートが見つかったら、スイッチフラグを真に設定する。
【００８７】
トポロジサーバにクエリを実行した場合にリモートポートが見つからない場合、又は出口ポート名が上位層もしくは下位層レイヤのポートの使用を命令する場合、上述のように下位層レイヤ又は上位層レイヤのポートをチェックする。出口ポートが見つかった場合、これを経路に追加し、ポートを含むデバイスを経路に追加し、フォーカスデバイスをリモートデバイスに設定する。
【００８８】
「ネクストホップ」ステップは、繰り返し数の既定限度に達するまで、又は経路が終了に達する（すなわちスイッチングもルーティングも行われない）まで、繰り返される。
【００８９】
プロセスが以前に識別したターミナルデバイスにおいて終了し、そのデバイスがアクセススイッチである場合、出口ポートは、「宛先の位置特定」から経路記録に追加され、宛先デバイスは経路記録に追加される。ターミナルデバイスが宛先デバイス自体である場合、ユーティリティは終了する。
【００９０】
図１１ａから図１１ｄは、モニタコンピュータで実行されるユーティリティの動作のフローチャートを示す。
【００９１】
[ロードバランサ（サーバ負荷分散装置）]
上述したように、フォーカスデバイスがターミナルデバイスである場合、ターミナルデバイスを宛先と共に経路記録に追加する。ターミナルデバイスがロードバランサである場合、ロードバランサのための仮想ＩＰ－サーバプールマッピングを取得する。これによって、ロードバランサのためのサーバ－物理サーバのマッピング（割り振り）を識別することができる。経路は、「ルート」経路である限り（ロードバランサデバイスまで）維持される。次に、各物理サーバＩＰアドレスについて、ロードバランサから物理サーバＩＰアドレスまで追加の経路発見ユーティリティを実行し、各追加経路の先頭に「ルート」経路を付け加える。
【００９２】
[ルーティングテーブルクエリ]

経路アルゴリズムを特に効率的なものとする要因の１つは、ルーティングデバイスに対するクエリを効率的に発生する、すなわち、著しいオーバーヘッドなく短時間でルーティングデバイスが応答することができるクエリを発生する能力である。

図５は、ＳＮＭＰによってアドレス指定可能な線形ルートテーブルの構造を示す。特定の宛先へのルートを確定するため、ｉｐＲｏｕｔｅＤｅｓｔがルートテーブル内への必要なインデックスである。

これは図５において４８で示されている。テーブル内の対象となるエントリは、出口インタフェースを規定するｉｐＲｏｕｔｅＩｆｉｎｄｅｘ５０、ネクストホップのＩＰアドレス（ネクストホップＩＰ）を規定するｉｐＲｏｕｔｅＮｅｘｔＨｏｐ５２、及びルーティングエントリのタイプ（無効／直接／間接）を規定するｉｐＲｏｕｔｅＴｙｐｅ５４である。通常、テーブル内へのアクセスには、ｉｐＲｏｕｔｅＭａｓｋ５６の知識が必要である。これによって特定のネットワークＩＰの位置を特定することができる。

しかしながら、図５に見られるように、ｉｐＲｏｕｔｅＭａｓｋ自体は、ｉｐＲｏｕｔｅＥｎｔｒｙに埋め込まれているので、クエリ内でセットアップされることは知られていない。必要なのは、テーブルへのインデックスを表すＩｐＲｏｕｔｅＤｅｓｔキー４８を見つけるため、以下に対する一致を見つけることである。

＜対象のＩＰ＞＆＜ｉｐＲｏｕｔｅＭａｓｋ．Ｘ＞　＝＝　＜ｉｐＲｏｕｔｅＤｅｓｔ．Ｘ＞

【００９３】

図２７は、プロセスを示す。

【００９４】

本発明者等により観察されたように、ＩｐＲｏｕｔｅＭａｓｋには、３３通りのみの可能性がある（／３２．．．／０）。すなわち、２５５．２５５．２５５．２５５、２５５．２５５．２５５．２５４、２５５．２５５．２５５．２５２、．．．０．０．０．０．である。

ＩＰアドレス内のゼロの数のために、これら複数により同一のＩＰアドレスについて複製ネットワークＩＤが生成される。全ての可能な３３通りのネットマスクのリストを生成し（Ｚ２）、ＩＰアドレスに適用する（Ｚ３）。

図６は、ＩＰアドレス１０．４４．１．２１３＝ＯＡ．２Ｃ．０１．Ｄ５＝００００　１０１０　００１０　１１００　０００Ｏ　０００１　１１０１　０１０１に対する３３の全てのネットマスクの適用を示す。

【００９５】

これは、１２個の固有値（標識された３２、３１、２９、２７、２５、２４、２３、１３、１２、１０、６、４）を生成する。したがって、ここでは、ルートを見つけるために１２個の固有値のＳＮＭＰクエリ（これは単一クエリパケットに提示することができる）を生成するだけでよい。

ステップＺ４からＺ５において、デフォルトルートが許可されているかを判定し、これに応じてネットワークを除去した後、１２個の固有値の結果をフォーカスデバイスのルートテーブルと照合して、一致が見つかったら、必要な要素ｉｐＲｏｕｔｅＩｆＩｎｄｅｘ（ｅｇｒｅｓｓＩｆｉｎｄｅｘ）、ｉｐＲｏｕｔｅＮｅｘｔＨｏｐ、及びｉｐＲｏｕｔｅＴｙｐｅを検索し（Ｚ１２）、モニタコンピュータ１６に対する応答で戻す。

【００９６】

この結果のインタフェースをｅｇｒｅｓｓＩｎｔｅｒｆａｃｅに設定する（Ｚ１３）。

【００９７】

ルートを見つけるために必要なクエリ数の削減を、本明細書において「推論的なキーイング」と称する。これによって、極めて効率的なリアルタイムのルートテーブルクエリ処理の実行が可能となる。

【００９８】

実ルーティングテーブルを調べる場合、所定のＩＰアドレスについて見つかったルートが有効な出口インタフェースを持たずにネクストホップアドレスを与えるのみであること

は珍しくない。これらの場合、ネクストホップアドレスをルーティングテーブルの次のクエリで用いて、そのネクストホップアドレスのための出口インタフェースを取得する試みを行う。このネクストホップアドレスの再使用を、出口インタフェースが取得されるまで繰り返す。

【００９９】

この手法によれば、第１のステップにおいて単一ルート探索クエリは、概説したように推論的なキーイングを用いて、指定されたＩＰアドレス（IP_x）のためのルーティングエントリを見つける。関連するｉｐＲｏｕｔｅＴｙｐｅが「直接」である場合、IP_x（及び$ipRouteIfindex_x$）は、ネクストホップとしてモニタコンピュータへの応答で返される。すなわち、これは直接接続されているので、レイヤ３ネクストホップを有しない。

【０１００】

関連するｉｐＲｏｕｔｅＴｙｐｅが直接でない場合、ｉｐＲｏｕｔｅＮｅｘｔＨｏｐ及びｉｐＲｏｕｔｅＩｆＩｎｄｅｘは、モニタコンピュータへの応答で返される。

【０１０１】

ルート探索（経路発見）プロセスでは、クエリの実行が更に難しいＩＰクラスレスドメイン間ルーティング（ＩＰ　Ｃｌａｓｓｌｅｓｓ　Ｉｎｔｅｒ－Ｄｏｍａｉｎ　Ｒｏｕｔｉｎｇ）テーブルも考慮している。

この場合、ステップＺ１０の結果としてＩＰアドレスが得られないならば、プロセスは、ステップＺ１４に移り、ＩＰｃｉｄｒＲｏｕｔｅＤｅｓｔ＋ネットワークアドレス＋ネットマスクを用いて、ＳＮＭＰクエリ（Ｇｅｔ　Ｎｅｘｔ（次を取得））をデバイスに発行する。

結果がＩＰアドレスでない場合、プロセスは、ループを戻ってステップＺ７に移り、再びステップＺ８、Ｚ９、Ｚ１０と進んでいく。結果がＩＰアドレスである場合、戻されたＯＩＤからネットワークアドレスを抽出する。

次に、ＯＩＤのネットワークアドレスがクエリのネットワークアドレスに一致するか否かを判定する。一致しない場合、プロセスはステップＺ７に戻る。一致する場合、見つかったルートを真に設定し、戻されたクエリのＯＩＤにＣＩＤＲキーを設定し、ＩＰｃｉｄｒＲｏｕｔｅＤｅｓｔすなわちＣＩＤＲルートテーブルに対するインデックスを除去する。次に、プロセスは、ＳＮＭＰクエリが、ネクストホップ、出口ｉｆＩｎｄｅｘ、及びルートタイプを取得することを可能とする。

【０１０２】

図に示すように、ＦｉｎｄＲｏｕｔｅＩｔｅｒａｔｉｖｅ（ルート探索繰り返し）プロセスにおいて、必要なＩＰアドレス（IP_x）についてＦｉｎｄＲｏｕｔｅ（ルート探索）ステップＦ１を行う。ルートが見つからない場合、失敗を戻す。ルートが見つかったが出口インタフェースがない場合、ｉｐＲｏｕｔｅＮｅｘｔＨｏｐを戻す。ルートが見つかると共にｉｐＲｏｕｔｅＩｆｉｎｄｅｘがゼロに等しい場合、ｉｐＲｏｕｔｅＮｅｘｔＨｏｐのＩＰアドレスについてＦｉｎｄＲｏｕｔｅＩｔｅｒａｔｉｖｅステップを行う。これには同一の４つの結果があり得る。

【０１０３】

推論的なキーイングは、大きいデータセットの効率的なクエリ処理のための特に良好な技法であるが、その主な適用範囲は、部分的なキーが既知である導出キーによってインデックスが付されたデータにクエリを実行する場合である。これは、ＳＮＭＰルートテーブル解析及びＳＮＭＰ　ＡＲＰテーブルクエリ処理の関係で特に有用となるからである。しかしながら、例えばＣＬＩアクセス及びＸＭＬ　ＡＰＩのような他のクエリ処理技法を用いて、迅速なネットワークデバイスごとの転送挙動も確認することができる。

【０１０４】

[ＡＲＰクエリ]

ここで図７を参照して、推論的なキーイングを用いてＡＲＰテーブルにクエリを実行するための効率的な技法について説明する。クエリの発生については、後で図７を参照して

更に詳しく検討する。クエリを実行するデバイスについて、ネットワークトポロジから、又はデバイス自体のインタフェースインデックス（ＩｆＩｎｄｅｘ）を移動することで、Ｉｆｉｎｄｅｘのリストを取得する。

デバイスの各ｉｆＩｎｄｅｘをＮＨＬ３アドレスと組み合わせて、デバイスへのクエリに含ませるキーのセットを発生する。これにより、これらのキーを含むクエリが構築され、フォーカスデバイスに送信される。フォーカスデバイスは、ゼロ又は１の成功応答を発生する。

図７は、ｉｐＮｅｔＴｏＭｅｄｉａＥｎｔｒｙテーブルフォーマットを示し、これは原則としていかなる所定のＩＰアドレスについてもＭＡＣアドレスを決定することができる。どのインタフェースからＡＲＰエントリが学習されたかが既知でない限り、特定のＩＰアドレスについて、一意のエントリを見つけることはできないので、各ＩＰアドレスをデバイス上の全てのｉｆＩｎｄｅｘと組み合わせることによって推論的なキーイングを用いる。

つまり、ＩＰアドレスをｉｆＩｎｄｅｘと組み合わせることで各クエリキーを生成することができる。このように、ＳＮＭＰクエリの数はデバイス上のインタフェースの数であり、これは典型的にデバイス上のＡＲＰエントリ数よりはるかに少なく、したがって、極めて高効率である。

【０１０５】

推論的なキーイングでは、単一のクエリメッセージ内に複数のクエリキーを含ませることができる。

【０１０６】

次に、経路識別のための代替的なアルゴリズムについて説明する。図１２を参照して、ループ繰り返しプロセスを説明する。メインループへのエントリは、図１２の上部でエントリ矢印４により示されている。エントリ矢印４は、後で説明するプライミングプロセスの終了時の状態を示す。エントリ状態は以下を含む。

＜オプション、フォーカスデバイス、ＮＨＬ３、ＮＨＬ２、ＶＬＡＮ＞

【０１０７】

これらのアイテムは、後で説明するプライミングプロセスによって設定される。これらを以下では「状態変数」と称する。「オプション」と呼ぶ状態変数は、配列したループオプションのシーケンスを有する。本例では、配列シーケンスはＣＳＲＡｃｒを備える。

【０１０８】

「ＶＬＡＮ」と呼ぶ状態変数は、経路のこのポイントで仮想パケットに現在タグ付けしているＶＬＡＮの仮想ローカルエリアネットワーク識別子（数字）である。

【０１０９】

ループのステップＬ０１では、第１のオプション（リストの先頭）を選択して実行する。これらのオプションについては後述する。オプションの実行後、リストオプションの先頭により実施される処理ステップの後に、プロセスは、リターンポイントＬ０２に戻って新しい状態を生成する（Ｌ０３）。この状態が以前に発生したか否かを判定し（Ｌ０４）、発生していない場合は状態を記憶する（Ｌ０５）。状態が以前に発生している場合は「ループ発見」報告を発生し、ループ限度をゼロに設定する。これはループを終了させる結果となる。

【０１１０】

ステップＬ０７では、ループ限度をデクリメントする。ループ限度がゼロ以下である場合、又はオプションシーケンス内にもう利用可能なオプションがない場合、又はフォーカスデバイスがターミナルデバイスと等しい場合、「終端は真に等しい」条件を設定する。

ステップＬ０８では、終端条件のチェックを実行し、終端条件が真である場合、メインループは終端する。他の場合、ステップＬ０３で生成した新しい状態を用いてメインループエントリポイントに戻る。

【０１１１】

ループ繰り返しにおける各オプションの実行において、オプション実行の第１のステッ

プは、オプション状態変数において配列シーケンスからそのオプションを除去することであることに留意すべきである。

【０１１２】

オプションの処理ステップの終了時、オプション及び処理ステップの結果に応じて、そのオプションは、シーケンス内にリセットされるか又は永久に除去される可能性がある。

【０１１３】

また、オプションの処理ステップ実行において、他の状態変数（フォーカスデバイス、ＮＨＬ２、ＮＨＬ３、ＶＬＡＮ）を個別に又は全体的に変更可能であることにも留意すべきである。何れかの状態変数を変更すると新しい状態が生じ、これは次のループ繰り返しのための新しいエントリ状態を構成することができる。

【０１１４】

これより図１３を参照してメインループプロセスの第２の部分について説明する。ステップＬ０９では、予想されたターミナルデバイスに到達したか否かを判定する。到達した場合、ステップＬ１０で、ターミナルデバイスがサーバＩＰに接続されたアクセススイッチであるか否かを判定する。そうでない場合、経路発見の完了成功を示した後にプロセスは終端し、経路完了を戻す。ターミナルデバイスがサーバＩＰに接続されたアクセススイッチである場合、アクセスポート接続及びサーバＩＰを経路に追加し、プロセスは、次に進んで経路発見成功で完了し、経路完了を戻した後に停止ステップで終了する。

【０１１５】

ステップＬ０９で予想されたターミナルデバイスに到達していないと判定した場合、ステップＬ１４においてＮＨＬ３がサーバＩＰであるか否かを質問する。そうでない場合、経路発見の完了が不成功であったと判定し、一部の経路を戻した後に停止する。ＮＨＬがサーバＩＰであった場合、これは、プロセスが最終ＬＳセグメントにあることを示すので、プロセスは、スキップセグメントカウンタをインクリメントすると共にフォーカスデバイスをＮＨＬ３に設定することで、宛先へとスキップすることができる。

【０１１６】

図１４は、オプションＣを示す。第１のステップＣ１では、オプション変数のシーケンスからオプションを除去する。ステップＣ２では、ネットワークアドレスが宛先（サーバＩＰ）と一致する単一のインタフェースがあるかについてチェックする。

単一のインタフェースがある場合、これは、アルゴリズムが最終スイッチネットワーク部分に到達したことを示すので、ＮＨＬ３をサーバＩＰに設定する。

ステップＣ３では、フォーカスデバイスにクエリを実行してサーバＩＰのためのＡＲＰエントリを見つけ、その結果をＮＨＬ２に設定する。次に、プロセスは、メインループ（Ｃ４）に戻り、別のオプションによって出口インタフェースを決定することができる。フォーカスデバイス上のＳＮＭＰ－ＡＲＰテーブルを用いてフォーカスデバイスへのクエリを行うか、又は見つからない場合は、ネットワーク管理システムＡＲＰキャッシュにクエリを行う。これらのクエリは、後で更に充分に説明する技法に従ったものである。

【０１１７】

ステップＣ２において、宛先サーバＩＰの単一のインタフェースのチェックが失敗した場合、単一のインタフェースは識別されない。状態変数は全く更新されない。この場合、オプション「Ｃ」を評価（及び廃棄）し、これが非生産的であることを認識するだけである。

【０１１８】

図１５は、オプションＳを示す。ステップＳ１０１に従って、オプションＳを配列シーケンスから除去する。ステップＳ１０２では、ネットワーク管理システムにクエリを実行し、又はフォーカスデバイスへのＳＮＭＰクエリを用いて、フォーカスデバイスがＮＨＬ３をホストするか否かを見出す。

ＮＨＬ３がフォーカスデバイス上にある場合、プロセスは、メインループリターンポイントに戻る（Ｓ１０４）。ルーティングアドレスＮＨＬ３がフォーカスデバイス上にない場合、スイッチングアドレスＮＨＬ２が設定されているか否かを判定し、ＮＨＬ３が与え

られたＮＨＬ２についてマッピングテーブル（ＡＲＰ）でフォーカスデバイスにクエリを実行する。

クエリについては後で更に充分に説明する。ステップＳ１０６では、ＶＬＡＮヒントが設定されているか否かを判定する。ＶＬＡＮヒントについては後述する。設定されていない場合、フォーカスデバイスから又はネットワーク管理システムからフォーカスデバイス上のＶＬＡＮリストを求める。リストからＶＬＡＮを選択し、フォーカスデバイス、ＮＨＬ２、及びＶＬＡＮを用いて、転送データベースエントリの検索を実行する。ステップＳ１０９に示す転送データベースエントリの検索については、図Ｘのフローチャートに示す。

ステップＳ１０６に戻ると、ＶＬＡＮヒントが設定されている場合、プロセスは直接ステップＳ１１０に進む。これは、ステップＳ１０９と同様のＦＤＢの検索であるが、ＦＤＢ内でクエリ実行したＶＬＡＮとしてＶＬＡＮヒントを用いる。

ステップＳ１１２（及びＳ１１１）では、転送データベース内でエントリが見つかったか否かを判定する。見つかった場合、プロセスは、図１６に示すオプションＳの第２の部分（エントリ矢印５）に進む。ステップＳ１１１でＦＤＢエントリが見つからない場合、ＦＤＢエントリがあるか否か判定されるまで、又はプロセスがメインループに戻ると判定されるまで、ＶＬＡＮループに入る。オプションＳの第２の部分に対するエントリポイントを図１５の下部に矢印５で示す。

これは、図１６の上部にも示されている。上述したように、転送データベースエントリが見つかった場合、これは経路の出口ポートを示している（Ｓ１１５）。これを用いて、ネットワークトポロジから次の接続デバイスを得ることができる。これについては、ステップＳ１１７に示し、後で更に充分に説明する。ステップＳ１１６は、図２に示し後述するＶＬＡＮヒントを得るステップである。

【０１１９】

ステップＳ１１７では、接続されたポートを取得するための処理を示すフローチャートが図２２ａ、ｂに示されており、それゆえ、出力ポートに基づいて、ネットワーク管理システムから後続の接続デバイスは、接続ポートが転送データベースから返される。

ステップＳ１１８で接続ポートが見つかった場合、接続ポート及び接続デバイスを識別した経路に追加し（ステップＳ１１９）、ステップＳ１２０では、フォーカスデバイスを変更して接続デバイスとする。ステップＳ１２１では、ループオプションをＣＳＲＡｃｒにリセットする。

次に、プロセッサは、ステップＳ１２２でメインループに戻る。ステップＳ１１８に戻ると、接続ポートが見つからない場合、プロセスは、スキップセグメントカウンタ８９（図８）をインクリメントすると共にフォーカスデバイスをＮＨＬ３に変更することで、ＮＨＬ３へとスキップする。スキップセグメントカウンタは、管理コンピュータにおいてハードウェア、ファームウェア、又はソフトウェアで実施され、以前のプロセスステップから次の接続デバイスが容易に確認できないことが明白である場合に経路のセグメントをスキップすることを表すことができる。

ステップＳ１２４では、フォーカスデバイスが宛先（サーバＩＰ）でないか否かを判定する。宛先でない場合、ステップＳ１２５でループオプションをＣＳＲＡｃｒにリセットする。フォーカスデバイスが宛先サーバである場合、ステップＳ１２６では、宛先が既知のアクセススイッチ上にあるか否かを判定し、ある場合、ＮＨＬ３をサーバアクセススイッチアドレスに設定する。

ステップＳ１２５でループオプションを設定した後、ステップＳ１２８では、フォーカスデバイスのＡＲＰテーブル又はＮＭＳにクエリを行うことで、ステップＳ１２７で設定したＮＨＬ３アドレスを用いてＮＨＬ２についてフォーカスデバイスにクエリを実行する。

【０１２０】

オプションＳは、ステップＳ１０１で利用可能オプションからこのオプションを除去すること、更に処理ステップの結果に基づいてステップＳ１２１及びステップＳ１２５で利

用可能オプションにこのオプションをリセットすることを含むことに留意すべきである。
【０１２１】
これより図１７を参照してオプションＲ及びオプションｒについて説明する。これらのオプションの各々は、ステップＲ１、ｒ１において配列したオプション変数のシーケンスからそのオプションを除去することで開始する。
ステップＲ２、ｒ２では、プロセスに、図に示す繰り返しルート探索プロセスを用いて宛先ＩＰ（サーバＩＰ）へのルートを探させる。オプションＲでは、プロセスは、デフォルトのルートが許可されていない場合に動作する（デフォルトルート許可は偽に等しい）。オプションｒでは、プロセスはデフォルトのルートを許可する（デフォルトルート許可は真に等しい）。ルートが見つからない場合、ステップＲ３においてプロセッサはメインループに戻る。ルートが見つかった場合、ルートが見つかったルーティングテーブルから求めた候補ＮＨＬ３を用いて、フォーカスデバイスにクエリをディスパッチする。このクエリは、候補ＮＨＬ３に対応した候補ＮＨＬ２を求めるためにデバイスのＡＲＰテーブルに対して行われる。
候補ＮＨＬ２が見つからない場合、プロセスは、オプションＲ／ｒの第２の部分のエントリポイント６に移る。ＡＲＰクエリから候補ＮＨＬ２が見つかった場合、ステップＲ８において、候補ＮＨＬ２が、プロセスＲ／ｒへのエントリ状態で状態変数として記録されているＮＨＬ２と同じであるか否かを判定するためにチェックを行う。それらが同じである場合、プロセスは、エントリポイント６に進む。
それらが同じでない場合、ステップＲ８の後、候補ＮＨＬ２がエントリ状態ＮＨＬ２と同じでない場合、ＮＨＬ３を候補ルートのネクストホップＩＰに設定し、ＮＨＬ２を候補ＮＨＬ２に設定する。ステップＲ１０では、Ｒ２及びｒ２におけるルーティングテーブルのクエリによって出口ポートが与えられたか否かを判定する。与えられた場合、プロセスはエントリポイント６に進む。与えられなかった場合、ステップＲ１１でオプションをＣＳＲＡｃｒにリセットする。
図１８では、図の上部にエントリポイント６を示す。次のステップＲ１２では、ルーティングテーブルにより出口ポートが与えられるか否かを判定する。与えられない場合、プロセスはメインループに戻る。ステップＲ１３では、図２１に従って、かつ後述するように、ＶＬＡＮヒント取得プロセスを行う。
【０１２２】
次に、図２２ａ、ｂに示すように接続ポート取得プロセスを行う。ステップＲ１５では、接続ポートが見つかったか否かを判定する。見つかった場合、出口ポート、接続ポート、及び接続デバイスを経路に追加する。フォーカスデバイスを接続デバイスに変更し、ループオプションをＣＳＲＡｃｒにリセットする。接続ポートが見つからない場合、この段階では何も行わない。プロセスはステップＳ２０に進み、ＮＨＬ３が更新されているか否かを判定する。更新されていない場合、プロセスはメインループに戻る。更新されている場合、ルーティングテーブルからの候補ＮＨＬ３があるとして、候補ＮＨＬ２について古いフォーカスデバイスにクエリを実行する。そのステップの後、ＮＨＬ２を決定した場合、プロセスはメインループに戻る。そうでない場合は、候補ＮＨＬ３があるとして、候補ＮＨＬ２について新しいフォーカスデバイスにクエリを実行する。
【０１２３】
次に図１９を参照してオプションｃについて説明する。第１のステップｃ１では、状態変数のオプションから「ｃ」を除去する。ステップｃ２では、ネットワークアドレスがＮＨＬ３ネットワークアドレスと一致する単一のインタフェースについてチェックを行う。一意のインタフェースが見つからない場合、プロセスはメインループに戻る。一意のインタフェースが見つかり、出口インタフェース名がある場合、図２１を参照して記載するＶＬＡＮヒントプロセスを実行する。ステップＣ４の後、図２２ａ、ｂに示す接続ポート取得プロセスを用いて接続ポートを得る。
【０１２４】
ステップｃ７では、ピアポート（ｐｅｅｒｅｄ　ｐｏｒｔ）が見つかったか否かを判定

する。見つかった場合、出口ポート、ピアポート、及びピアデバイスを経路に追加し、フォーカスデバイスをピアデバイスに設定する。ｃ８において、利用可能オプションをＣＳＲＡｃｒにリセットする。ステップｃ７でピアポートが見つからない場合、プロセスはメインループに戻る。

【０１２５】

次に、図２０を参照してオプションＡについて説明する。ステップＡ１では、状態変数のオプションシーケンスからオプションＡを除去する。ステップＡ２では、ＮＨＬ３からＮＨＬ２へのマッピングのためのＳＮＭＰ　ＡＲＰエントリを見つける。マッピングが見つからない場合、プロセスはメインループに戻る。

【０１２６】

マッピングが見つかった場合、プロセスはＳＮＭＰを用いて、関係が学習されたインタフェースのｉｆＩｎｄｅｘを見つける。ステップＡ５では、一意のインタフェースが見つかったか否かを判定する。見つからなかった場合、プロセスはメインループに戻る。見つかった場合、プロセスはステップＡ６に進んで、利用可能な出口インタフェース名が存在するか否かを判定する。存在する場合、図２１を参照して説明するようにＶＬＡＮヒントプロセスを実行させる。次に、Ａ８において、図２２ａ、ｂに示すように接続ポート取得プロセスを実行させる。

【０１２７】

接続ポート取得プロセスの結果としてピアポートが見つかった場合、出口ポートを経路に追加し、ピアポート及びピアデバイスを経路に追加し、フォーカスデバイスをピアデバイスに設定する。更に、利用可能オプションをＣＳＲＡｃｒにリセットする。ピアポートが見つからない場合、プロセスはメインループに戻る。

【０１２８】

次に、図２１を参照してＶＬＡＮヒントプロセスについて説明する。このプロセスは、オプションＡのステップＡ７、オプションｃのステップｃ５、オプションＲ／ｒのステップＲ１３、及びオプションＳのステップＳ１１６で用いられた。更に、これは、まだ検討していないプライミングプロセスの１つにおいても用いられる。プロセスは、ステップＶＬ１でアクセスポート名から開始する。ステップＶＬ２では、名前が「ＶＬ＋数字」の形態であるか否かを判定し、そうである場合、ステップＶＬ３でこの数字を抽出してＶＬＡＮヒントとして記憶する。

【０１２９】

そうでない場合、ネットワーク管理システムでは、インタフェース上の全てのＶＬＡＮのリストが要求される。ステップＶＬ５は、競合するＶＬＡＮが存在するかについてチェックする。存在しない場合、ステップＶＬ６で一意のＶＬＡＮをＶＬＡＮヒントとして記憶する。競合するＶＬＡＮが存在する場合、既に記憶されているＶＬＡＮヒントを変更せずに残す。

【０１３０】

ＶＬＡＮヒントは、ネットワークのスイッチ部分（レイヤ２）で論理ループを防止する（トラフィックが無限にループするのを回避する）ために用いられるスパニングツリープロトコル（ＳＴＰ）と呼ばれるスイッチング技法を用いたネットワークで生じ得る問題に対処する。このプロトコルは、スイッチングデバイスが所定のパケットをどこに転送するべきかを決定するために用いられる。

【０１３１】

すなわち、デバイスがスイッチングしている場合、スイッチはレイヤ２（ＭＡＣ／イーサネット（登録商標））ヘッダを調べると共に宛先レイヤ２アドレス（ＮＨＬ２）を調べてから、内部データベース（ＦＤＢすなわち転送データベース）に照会して、どのポートからパケットを送信するかを確認する。多くの企業は、ＰＶＳＴＰ（ＶＬＡＮごとのスパニングツリープロトコル）と呼ばれるＳＴＰの拡張版を使用し、これによって各パケットにＶＬＡＮ識別子も付ける。スイッチは別個のＦＤＢをＶＬＡＮごとに１つ維持する。

【０１３２】

これは、一部には効率のため、一部にはいっそう複雑な仮想トポロジを可能とするために実行される。したがって、同一のレイヤ２宛先を有する２つのパケットが異なるポートによって配信することは完全に可能である（と共に、珍しいことではない）。それらの宛先が同一デバイス／ポートであっても、異なるＶＬＡＮにタグ付けされているからである。

【０１３３】

この結果、一致が見つかるまで、このプロセスは、ＶＬＡＮごとのＦＤＢの全てを徹底的に探す（ｔｒａｗｌ）ことができない。パケットがどのＶＬＡＮのメンバであるとタグ付けされているかを即座（アプリオリ）に知ることは重要である。

【０１３４】

このＶＬＡＮタグ付けはネットワーク内の様々な場所で行われ得る。例えばこれは、ソースアクセスポートで、すなわちソースデバイスが物理的に接続されている箇所で、又はネットワーク内の別の場所で行われ得る。あるＶＬＡＮタグが別のもので置換されることは珍しいことではない（これはＶＬＡＮ間ルーティングと呼ばれる）。

【０１３５】

例えば、パケットがネットワークデバイスＤに到着した場合（経路Ａ→Ｂ→Ｃ→Ｄ）、ＡからのパケットがＤに到達した時点で位置するＶＬＡＮが既知である場合、Ｄには現在の出口インタフェースについてクエリを実行するだけでよい。例えば、ＡがＶＬＡＮ１００にパケットを配置し、Ｂが（ＶＬＡＮ１００を用いて）これを送り、次に、Ｃが１００を２００に変更し、次に、ＤがＶＬＡＮ２００を用いてこれをスイッチングするという場合がある。

【０１３６】

このため、ソースデバイスから宛先デバイスまでネットワーク中で、追跡する仮想パケットの一部としてＶＬＡＮヒントを「運ぶ」必要がある。したがって、適用可能な場合にＶＬＡＮヒントの使用、取り消し、リセット、又は更新が行われる。

【０１３７】

既に述べたように、図２３は、オプションＲ及びオプションｒで用いられるｆｉｎｄＲｏｕｔｅＩｔｅｒａｔｉｖｅプロセスを示す。このプロセスは、ステップＦ１で開始してルート限度チェックＦ２で終端するルート探索ループを含む。次に、プロセスｆｉｎｄＲｏｕｔｅＩｔｅｒａｔｉｖｅは、ルートが見つかったか否かを判定し、ルートに関連した出口インデックスの位置を特定することができる。

【０１３８】

メインループを開始する前に、ループの第１の繰り返しのエントリ状態を設定するために実施する３つのプライミングプロセスがある。図２４に第１のプライミングプロセスを示す。これは、（図２４でクライアント側と称される）ソースデバイスについて識別されるアクセススイッチ又はネットワークデバイスを開始ポイントとして設定する。同様に、図２４でサーバ側と称される宛先デバイスに基づいて、アクセススイッチ又はネットワークデバイスを停止ポイントとして記憶する。図２４では、クライアントＩＰ及びサーバＩＰはそれぞれソース及び宛先のアドレスである。

【０１３９】

図２５は、初期エントリ状態ＮＨＬ３及びＮＨＬ２アドレスをセットアップするためのプライミングプロセスである。

【０１４０】

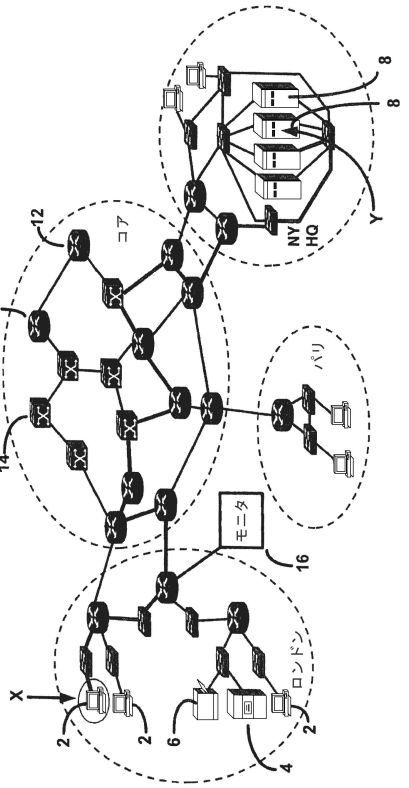
図２６は、初期エントリ状態についてフォーカスデバイスを設定する第３のプライミングプロセスを示す。

【０１４１】

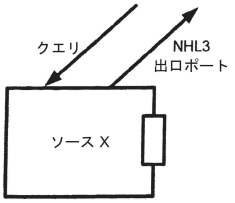
図２４の第１のプライミングプロセスは、図２５の第２のプライミングプロセスに至り、図２５の第２のプライミングプロセスは、図２６の第３のプライミングプロセスに至ることに留意すべきである。第３のプライミングプロセスは、図Ａに示すメインループのメインエントリポイント４に至る。

【０１４２】
[追加の技術／プロトコル]
経路発見アルゴリズムは、上述のように利用された場合、一般的に既知のネットワークプロトコルに従って動作している相互接続デバイスのネットワーク内で特定のパケット又はメッセージが取り得る特定の経路を識別する効果的な方法を提供する。何らかの理由で、経路発見アルゴリズムが特定の問題に直面する状況が生じる。これらの問題のいくつかについて以下で検討する。
【０１４３】
場合によっては、アルゴリズムにおいて実行されるユーティリティは、マルチプロトコルラベルスイッチング（ＭＰＬＳ：Ｍｕｌｔｉ－Ｐｒｏｔｏｃｏｌ Ｌａｂｅｌ Ｓｗｉｔｃｈｅｄ）ネットワークセグメントを横断しなければならない。これを達成するため、（トラフィックがＭＰＬＳセグメントに入るポイントで）初期ラベル割り当てを見つけ、ラベルのポッピング、プッシング、及び転送のホップごとの詳細を用いたホッピングによってＭＰＬＳネットワーク中で追跡する。これを、トラフィックがその最終ラベルをポッピングしてＭＰＬＳセグメントから出るまで続ける。
【０１４４】
別の問題は、ＮＡＴデバイスのＮＡＴテーブルのポーリングによって達成され得るＮＡＴ境界の横断である。これは、ダイナミックＮＡＴについてリアルタイムの推論的なポーリング（ｓｐｅｃｕｌａｔｉｖｅ ｐｏｌｌｉｎｇ）を必要とする場合があるが、スタティックＮＡＴではバックグラウンドポーリングを用いることが可能であり得る。
【０１４５】
ＩＰＳＥＣ／ＧＲＥ／ＳＳＬ等のトンネルプロトコルにおいて、ユーティリティは、トンネルの一端から他端までの直接ルートについてチェックする（典型的に、それらの間の全ノードを表す１つの未知のレイヤ３ホップによる）。ユーティリティは、更に、プロトコルに特有のトポロジ情報をチェックし、ルーティングテーブル／インタフェースにおいて暗号トンネル（ｃｒｙｐｔｏ／ｔｕｎｎｅｌｌｉｎｇ）ホップが存在するかチェックする。
【０１４６】
別の問題は仮想化である。アルゴリズムが物理的な出口ポートを識別して、出口ポートに接続された物理デバイスにトポロジからアクセス可能とすることは重要である。多くのネットワークは、様々な異なる仮想化レイヤで動作する。追加のＡＰＩを用いて仮想スイッチにクエリを実行することができる。トポロジサーバがエンドホスト位置に関するタイムリーな情報を有することを確実とするため、トポロジサーバは仮想化管理プラットフォームと一体化して仮想マシンの再配置に関する更新を実行し、影響を受ける仮想スイッチ上のエンドホスト位置の事前対応型ポーリングを可能とすることが必要であり得る。
【０１４７】
ユーティリティは、特定のＶＲＦ（ｖｉｒｔｕａｌｉｚｅｄ ｒｏｕｔｉｎｇ ａｎｄ ｆｏｒｗａｒｄｉｎｇ）識別子に必要である適切なＩＰ転送（ルーティングテーブル）にクエリを実行することで、仮想化ルーティング及び転送テーブル（ＶＲＦ）をネゴシエートする。例えばＳＮＭＰにおいて、これはＶＲＦコンテキスト化コミュニティストリングを用いて実行可能である。

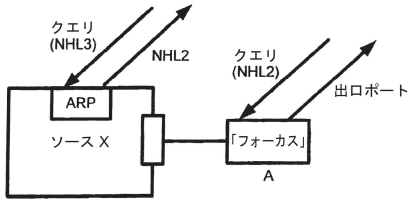
【図１】



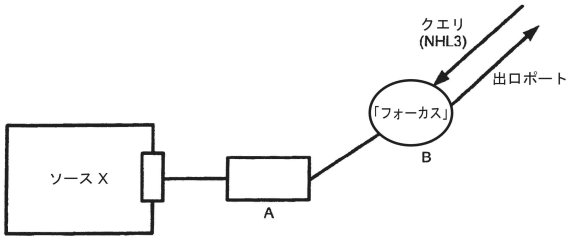
【図２ａ】



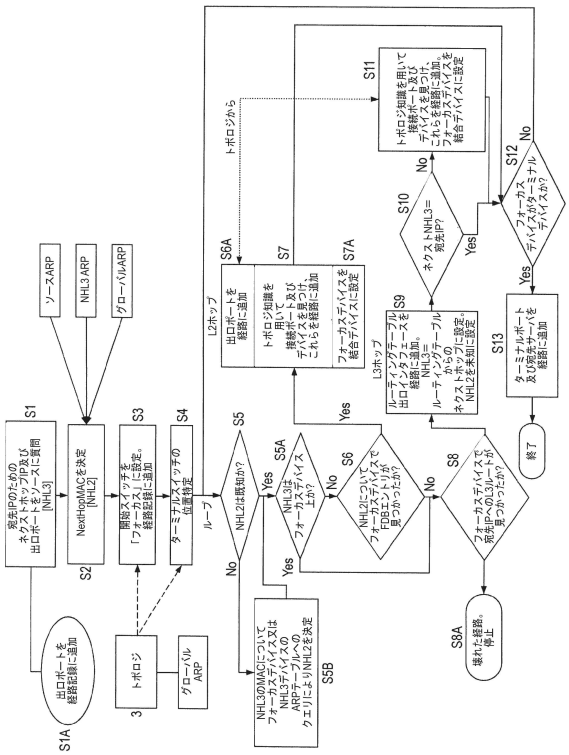
【図２ｂ】



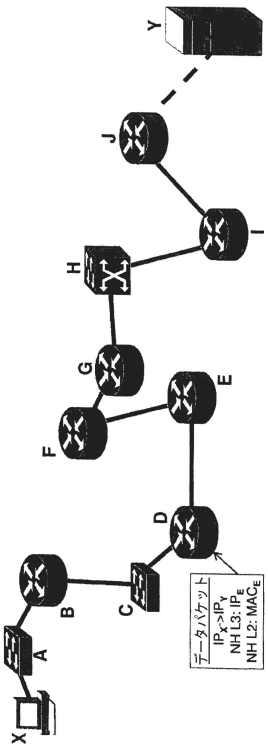
【図２ｃ】



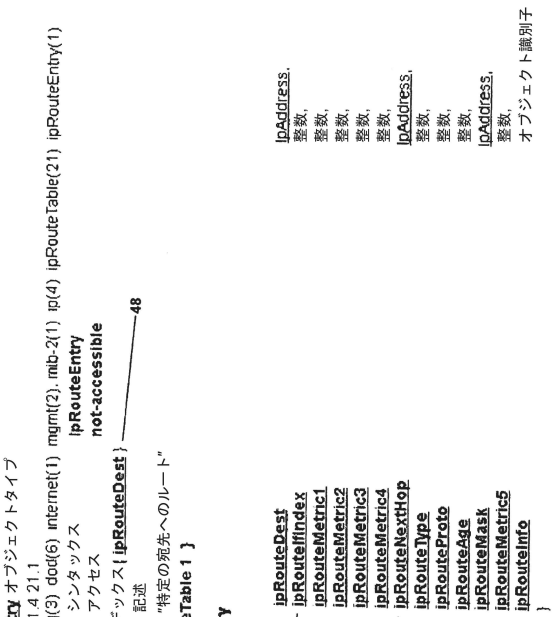
【図３】



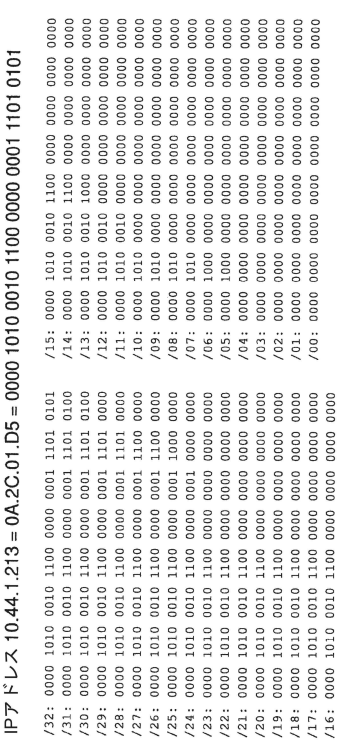
【図４】



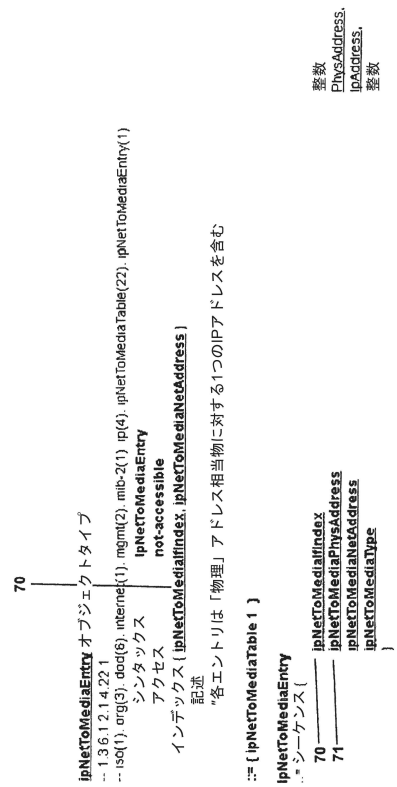
【図５】


ipRouteEntry オブジェクトタイプ
-- 1.3.6.1.2.1.4.21.1
-- iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) ip(4) ipRouteTable(21) ipRouteEntry(1)
シンタックス ipRouteEntry
アクセス not-accessible
インデックス { ipRouteDest } 48
記述
"特定の宛先へのルート"
::= { ipRouteTable 1 }
ipRouteEntry
::=
シーケンス{
ipRouteDest IpAddress,
50 ipRouteIfIndex 整数,
ipRouteMetric1 整数,
ipRouteMetric2 整数,
ipRouteMetric3 整数,
ipRouteMetric4 整数,
52 ipRouteNextHop IpAddress,
54 ipRouteType 整数,
ipRouteProto 整数,
ipRouteAge 整数,
56 ipRouteMask IpAddress,
ipRouteMetric5 整数,
ipRouteInfo オブジェクト識別子
}

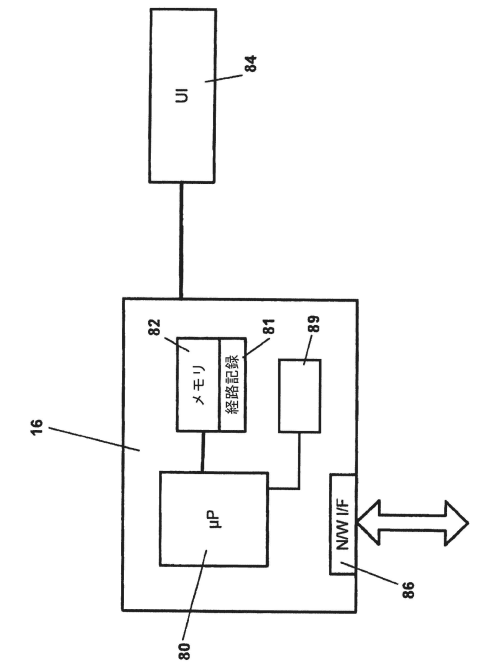
【図６】


IPアドレス 10.44.1.213 = 0A.2C.01.D5 = 0000 1010 0010 1100 0000 0001 1101 0101
/32: 0000 1010 0010 1100 0000 0001 1101 0101 /15: 0000 1010 0010 1100 0000 0000 0000 0000
/31: 0000 1010 0010 1100 0000 0001 1101 0100 /14: 0000 1010 0010 1100 0000 0000 0000 0000
/30: 0000 1010 0010 1100 0000 0001 1101 0100 /13: 0000 1010 0010 1000 0000 0000 0000 0000
/29: 0000 1010 0010 1100 0000 0001 1101 0000 /12: 0000 1010 0010 0000 0000 0000 0000 0000
/28: 0000 1010 0010 1100 0000 0001 1101 0000 /11: 0000 1010 0010 0000 0000 0000 0000 0000
/27: 0000 1010 0010 1100 0000 0001 1100 0000 /10: 0000 1010 0000 0000 0000 0000 0000 0000
/26: 0000 1010 0010 1100 0000 0001 1100 0000 /09: 0000 1010 0000 0000 0000 0000 0000 0000
/25: 0000 1010 0010 1100 0000 0001 1000 0000 /08: 0000 1010 0000 0000 0000 0000 0000 0000
/24: 0000 1010 0010 1100 0000 0001 0000 0000 /07: 0000 1010 0000 0000 0000 0000 0000 0000
/23: 0000 1010 0010 1100 0000 0000 0000 0000 /06: 0000 1000 0000 0000 0000 0000 0000 0000
/22: 0000 1010 0010 1100 0000 0000 0000 0000 /05: 0000 1000 0000 0000 0000 0000 0000 0000
/21: 0000 1010 0010 1100 0000 0000 0000 0000 /04: 0000 0000 0000 0000 0000 0000 0000 0000
/20: 0000 1010 0010 1100 0000 0000 0000 0000 /03: 0000 0000 0000 0000 0000 0000 0000 0000
/19: 0000 1010 0010 1100 0000 0000 0000 0000 /02: 0000 0000 0000 0000 0000 0000 0000 0000
/18: 0000 1010 0010 1100 0000 0000 0000 0000 /01: 0000 0000 0000 0000 0000 0000 0000 0000
/17: 0000 1010 0010 1100 0000 0000 0000 0000 /00: 0000 0000 0000 0000 0000 0000 0000 0000
/16: 0000 1010 0010 1100 0000 0000 0000 0000

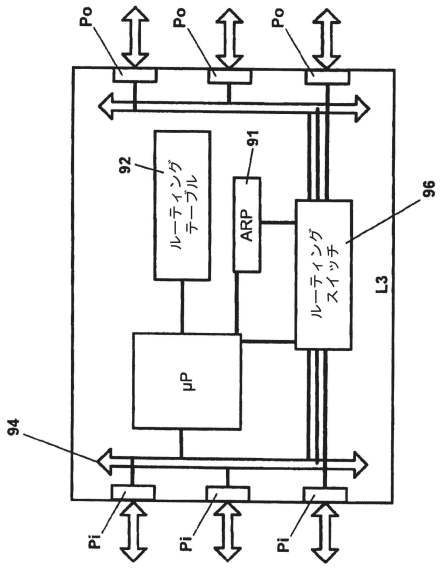
【図７】


70
ipNetToMediaEntry オブジェクトタイプ
-- 1.3.6.1.2.1.4.22.1
-- iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) ip(4) ipNetToMediaTable(22) ipNetToMediaEntry(1)
シンタックス ipNetToMediaEntry
アクセス not-accessible
インデックス { ipNetToMediaIfIndex, ipNetToMediaNetAddress }
記述
"各エントリは「物理」アドレス相当物に対する1つのIPアドレスを含む
::= { ipNetToMediaTable 1 }
ipNetToMediaEntry
::=
シーケンス {
70 ipNetToMediaIfIndex 整数
71 ipNetToMediaPhysAddress PhysAddress,
ipNetToMediaNetAddress IpAddress,
ipNetToMediaType 整数
}

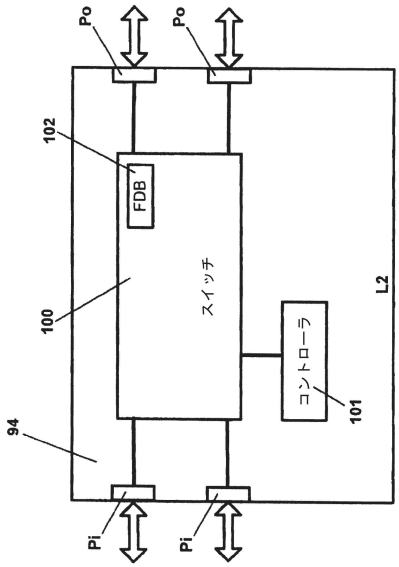
【図８】


16
UI
84
82
メモリ
経路記録
81
89
μP
NW I/F
80
86

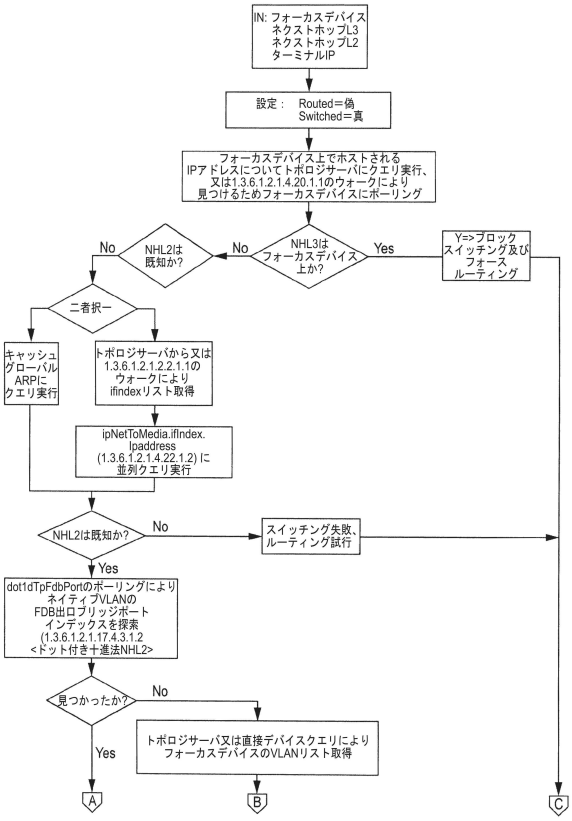
【図９】



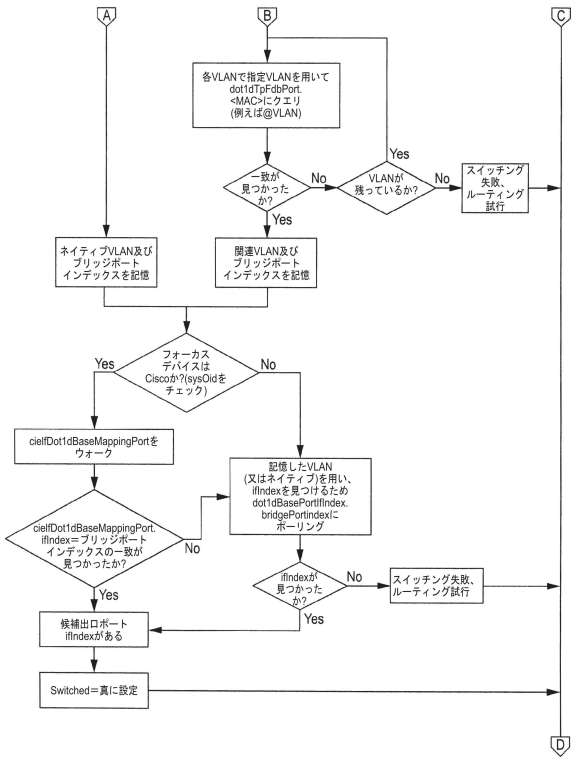
【図１０】



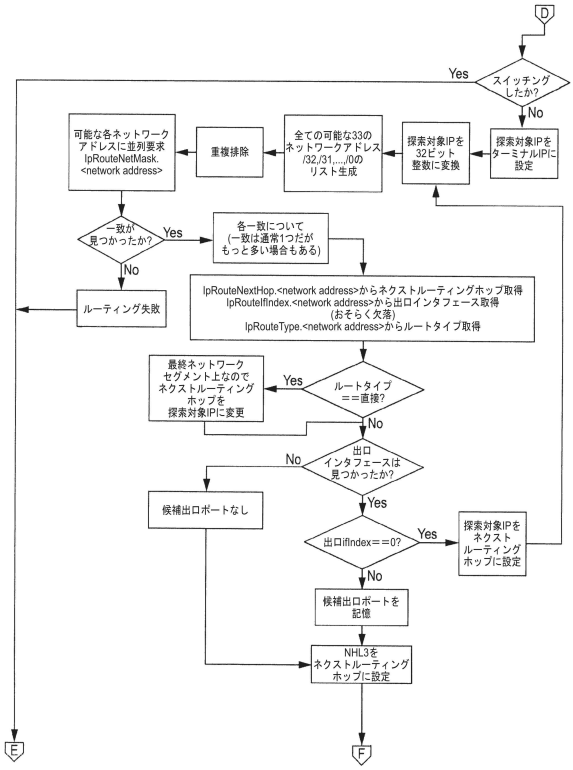
【図１１ａ】



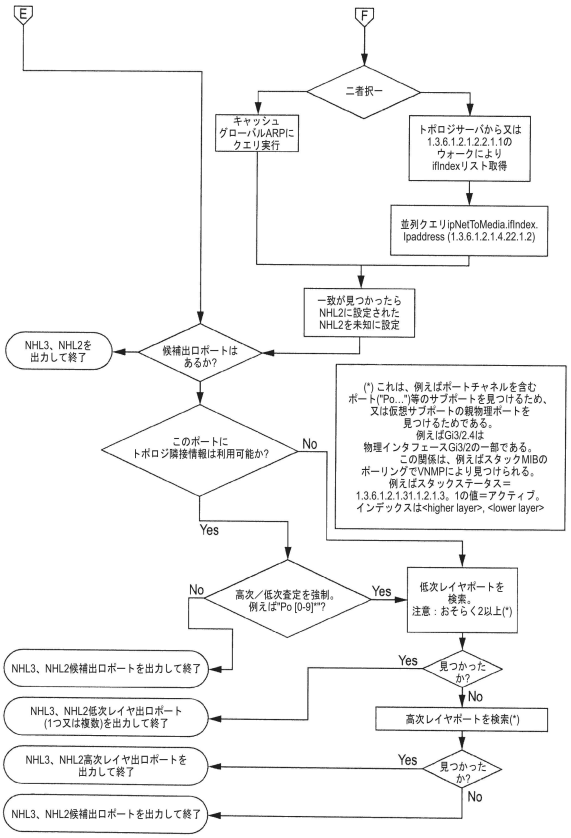
【図１１ｂ】



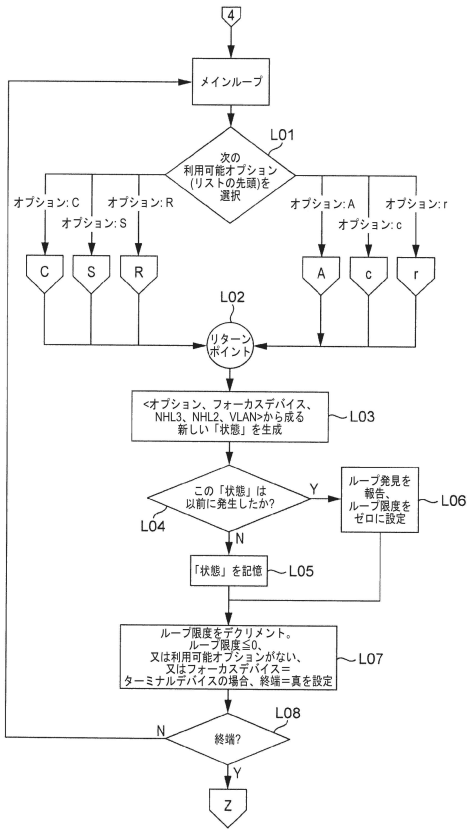
【図１１ｃ】



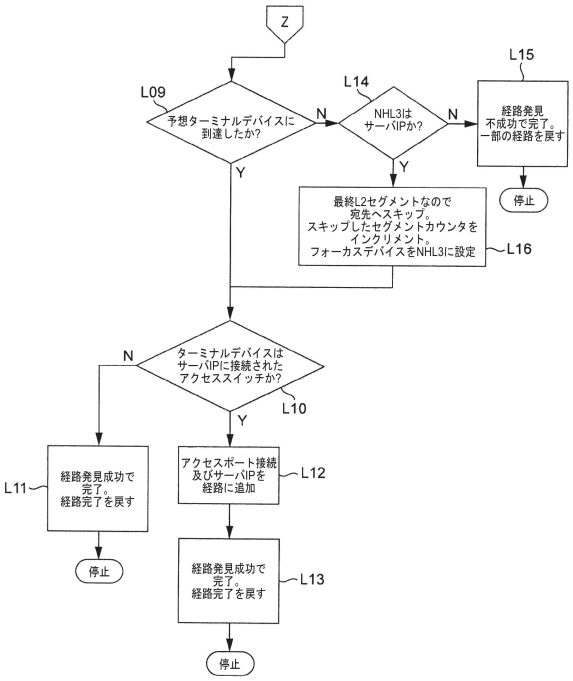
【図１１ｄ】



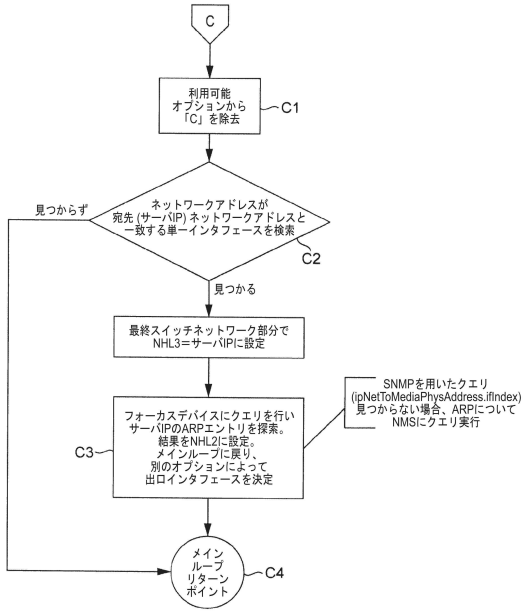
【図１２】



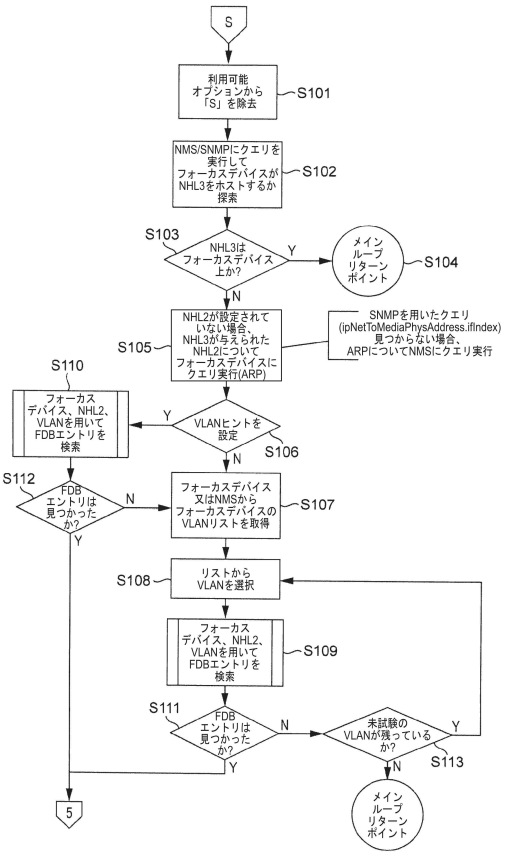
【図１３】



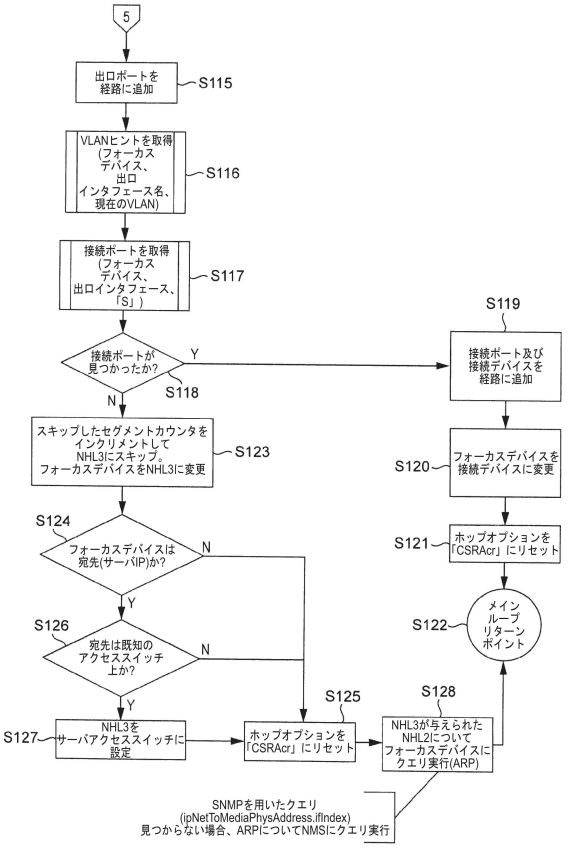
【図１４】



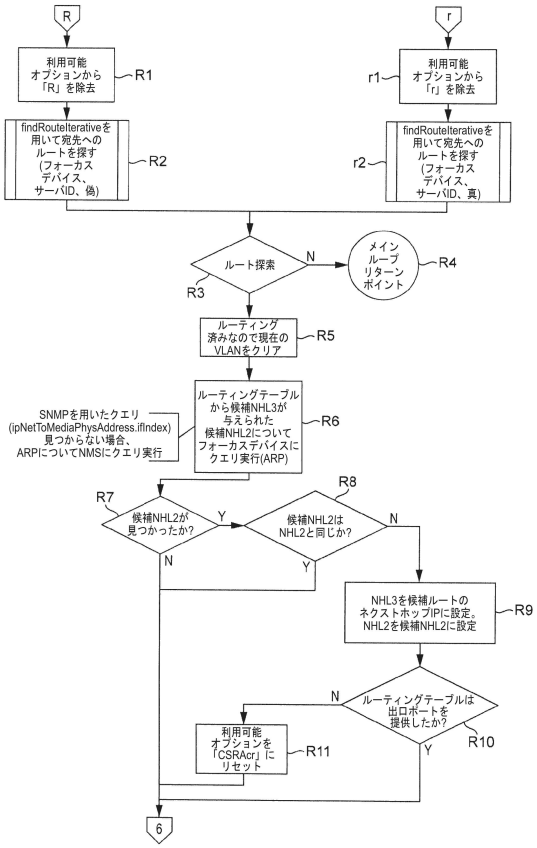
【図１５】



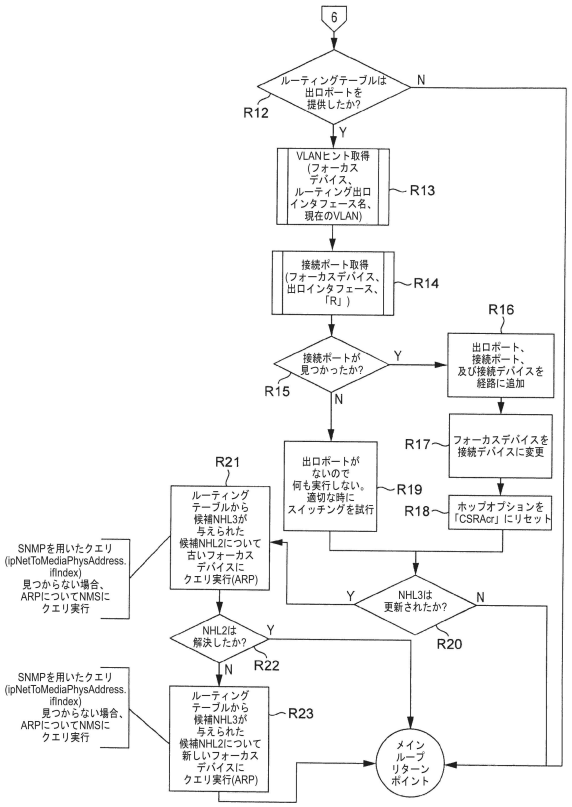
【図１６】



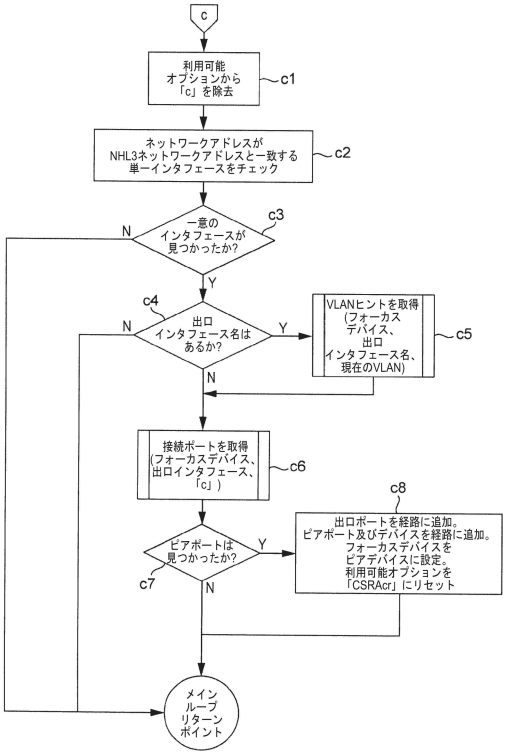
【図１７】



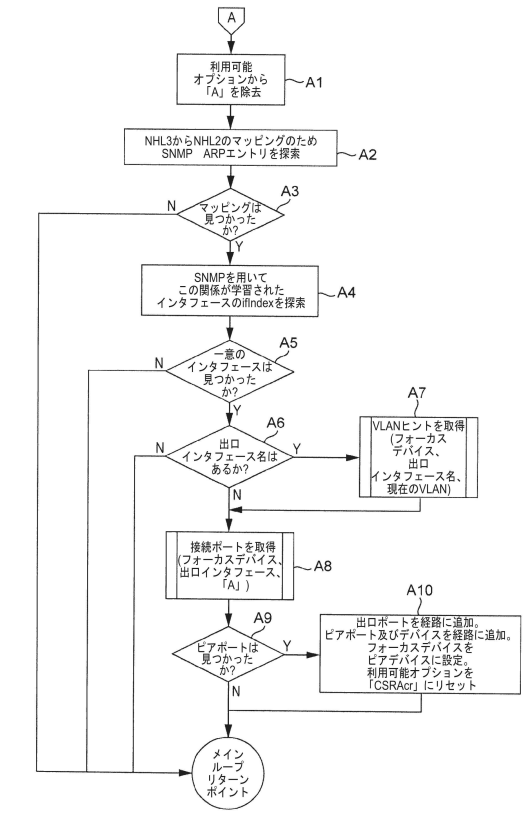
【図１８】



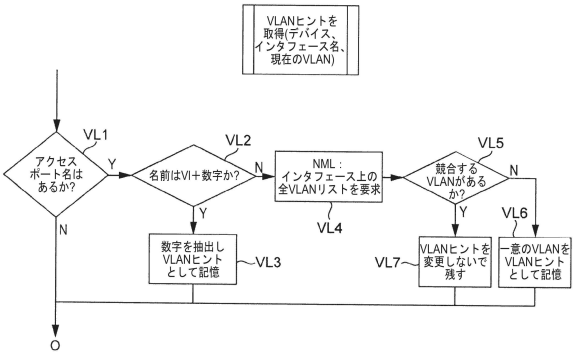
【図１９】



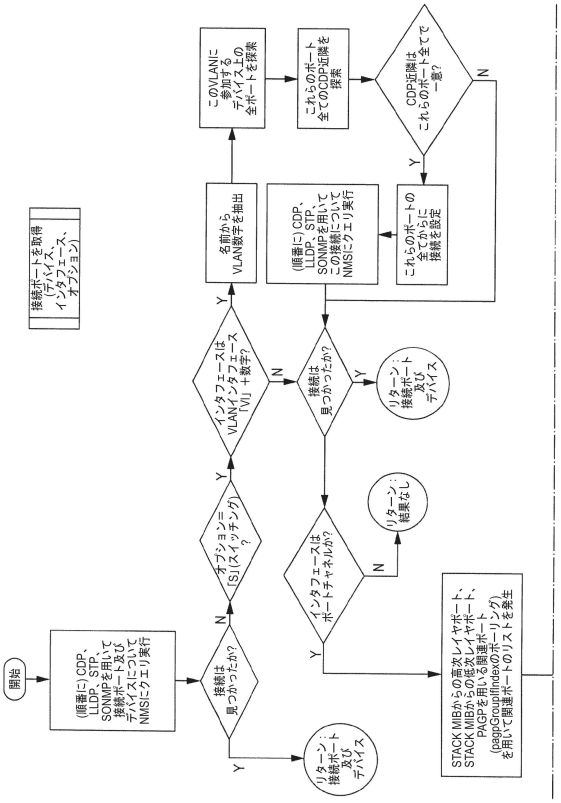
【図２０】



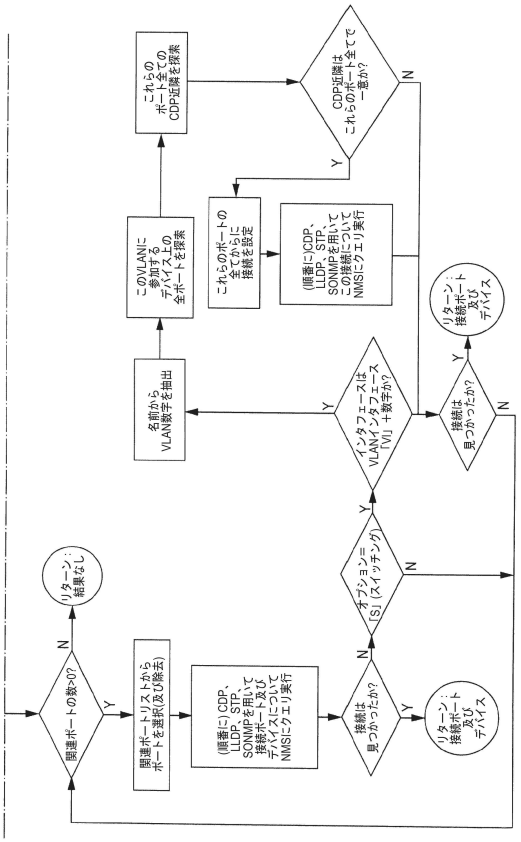
【図２１】



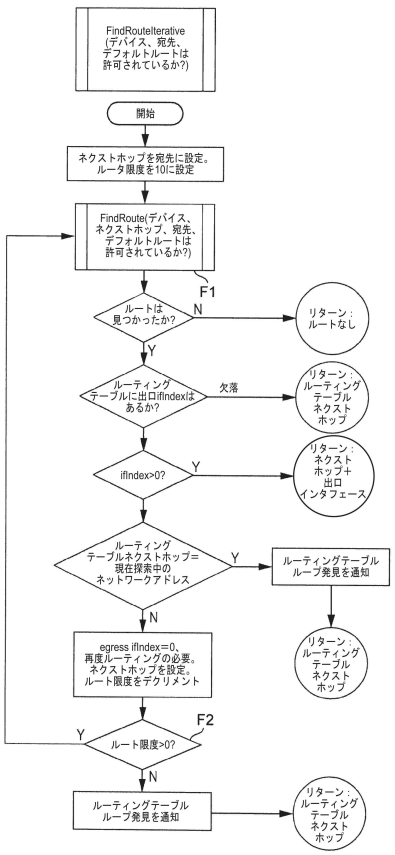
【図２２ａ】



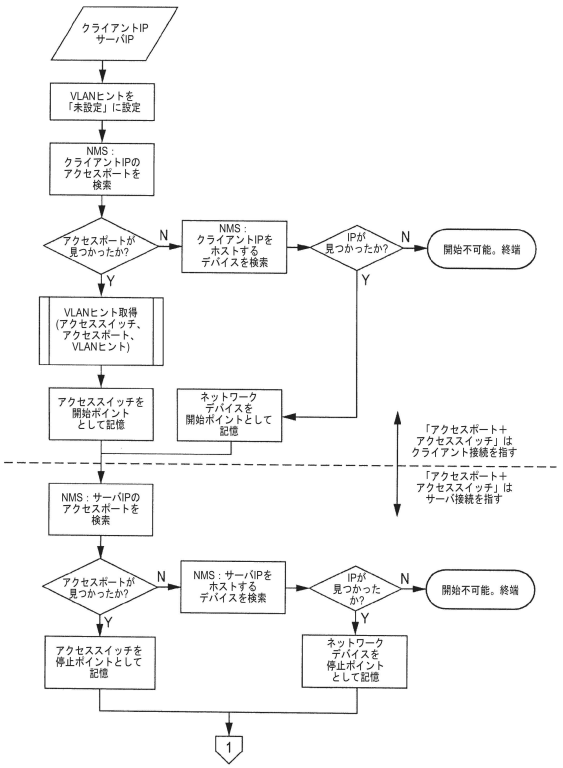
【図２２ｂ】



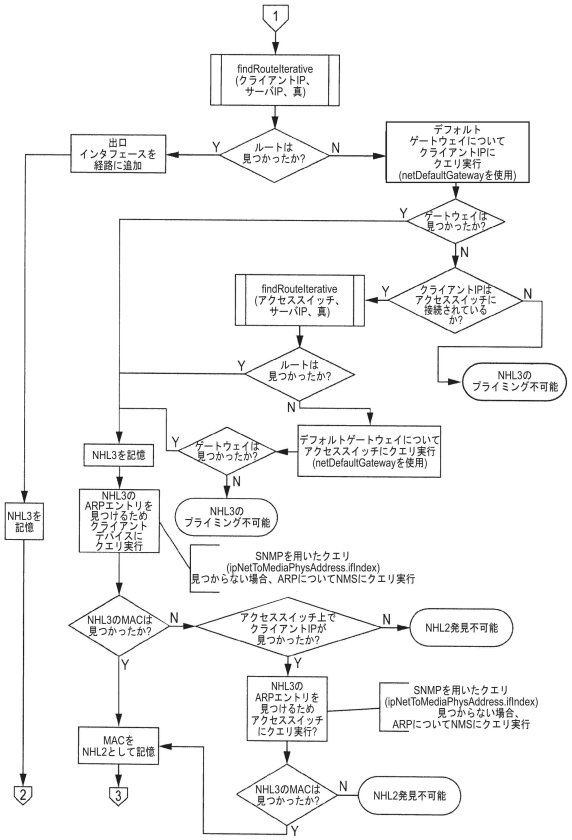
【図２３】



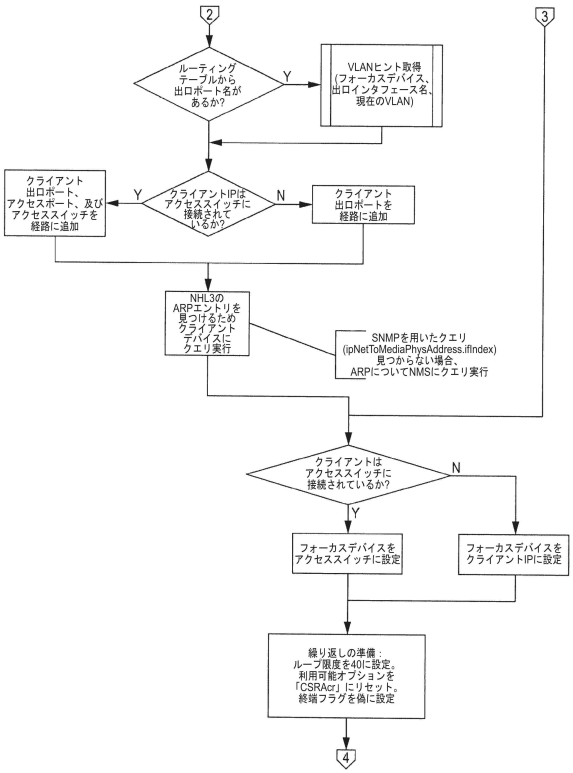
【図２４】



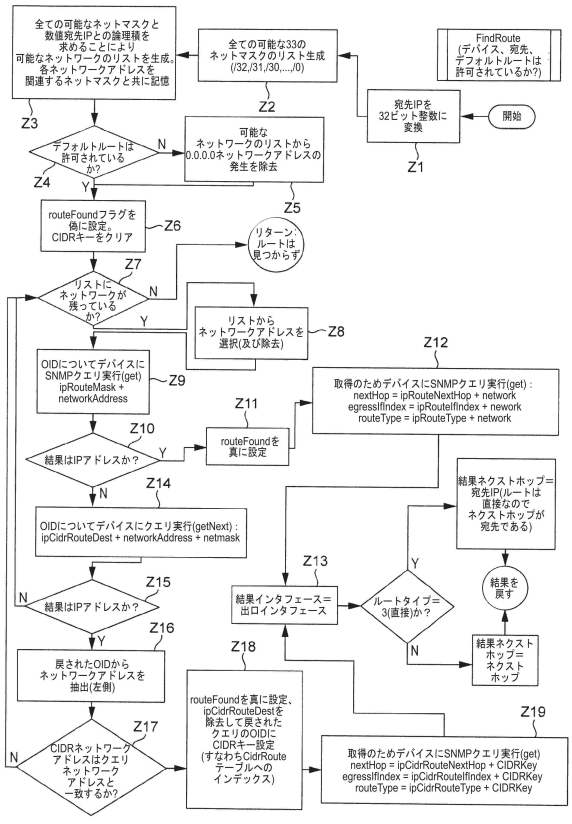
【図２５】



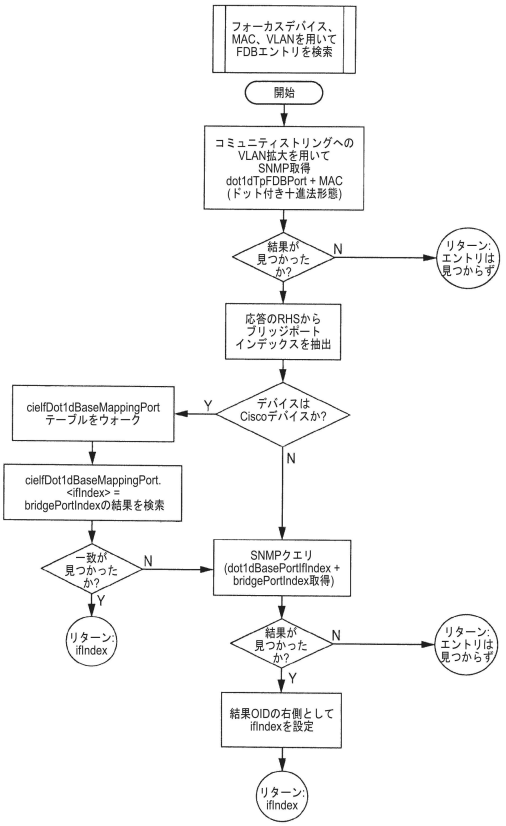
【図２６】



【図２７】



【図２８】



フロントページの続き

(74)代理人 100096769
弁理士 有原 幸一
(74)代理人 100107319
弁理士 松島 鉄男
(74)代理人 100114591
弁理士 河村 英文
(72)発明者 ローパー，ジェフリー，ジョン
イギリス、ディーエー３ ７キューエヌ ロングフィールド ケント、ノースダウン ロード ５１

審査官 宮島 郁美

(56)参考文献 米国特許第０５６７５７４１（ＵＳ，Ａ）
米国特許出願公開第２００７／０１７１８４４（ＵＳ，Ａ１）
特開２０１０－０６３０５８（ＪＰ，Ａ）
特表２００４－５３７８８１（ＪＰ，Ａ）
特開２００５－０１２２９０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ１２／００－１２／２８，１２／４４－１２／９５５