

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2011-103613

(P2011-103613A)

(43) 公開日 平成23年5月26日(2011.5.26)

(51) Int.Cl. F I テーマコード(参考)  
 H04L 12/56 (2006.01) H04L 12/56 100Z 5K030

審査請求 未請求 請求項の数 9 O L (全 18 頁)

(21) 出願番号 特願2009-258484 (P2009-258484)  
 (22) 出願日 平成21年11月12日(2009.11.12)

(71) 出願人 000004226  
 日本電信電話株式会社  
 東京都千代田区大手町二丁目3番1号  
 (74) 代理人 100064414  
 弁理士 磯野 道造  
 (74) 代理人 100127720  
 弁理士 大石 恵  
 (72) 発明者 杉園 幸司  
 東京都千代田区大手町二丁目3番1号 日  
 本電信電話株式会社内  
 (72) 発明者 山崎 裕史  
 東京都千代田区大手町二丁目3番1号 日  
 本電信電話株式会社内  
 Fターム(参考) 5K030 GA01 HA08 HD09 KA05 LB05  
 MD07

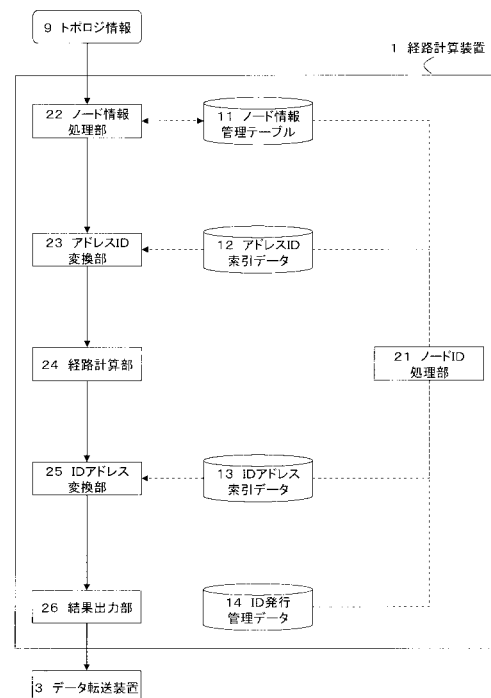
(54) 【発明の名称】 経路計算装置、経路計算方法、および、経路計算プログラム

(57) 【要約】

【課題】経路計算処理におけるトポロジのノードアクセス処理において、計算コスト削減と、使用メモリ量削減とをバランスよく実現すること。

【解決手段】経路計算装置1のノードID処理部21が、トポロジ情報9のノードアドレスごとに、ノードの識別情報としてのノードIDをノードアドレスと1:1対応するように発行し、アドレスID変換部23が、トポロジ情報9のノードごとに、ノードアドレスを、アドレスID索引データ12を参照してノードID処理部21が発行したノードIDへと変換し、経路計算部24が、アドレスID変換部23が変換したトポロジ情報9をもとに、変換したノードIDをノードの識別情報としてトポロジ情報9の経路を計算し、IDアドレス変換部25が、経路計算部24の計算結果の各ノードIDをノードアドレスへと変換する。

【選択図】図3



**【特許請求の範囲】****【請求項 1】**

ノードアドレスで識別されるノードがリンクで接続されるトポロジ情報から、ノードごとに経路計算を行う経路計算装置であって、

前記経路計算装置は、アドレスID索引データおよびIDアドレス索引データを記憶する記憶手段と、ノードID処理部と、アドレスID変換部と、経路計算部と、IDアドレス変換部とを有し、

前記ノードID処理部は、前記トポロジ情報のノードアドレスごとに、ノードの識別情報としてのノードIDをノードアドレスと1:1対応するように発行し、その発行した結果を、ノードIDからノードアドレスを検索するための前記IDアドレス索引データと、ノードアドレスからノードIDを検索するための前記アドレスID索引データとにそれぞれ格納し、

前記アドレスID変換部は、前記トポロジ情報のノードごとに、そのノードに関する経路計算に使用される情報のうちのノードアドレスを、前記アドレスID索引データを参照して前記ノードID処理部が発行したノードIDへと変換し、

前記経路計算部は、前記アドレスID変換部が変換した前記トポロジ情報をもとに、変換したノードIDをノードの識別情報として前記トポロジ情報の経路を計算し、

前記IDアドレス変換部は、前記経路計算部の計算結果を読み取り、その計算結果に含まれる各ノードIDについて、前記IDアドレス索引データを参照してノードアドレスへと変換することで、ルーティングテーブルを作成することを特徴とする

経路計算装置。

**【請求項 2】**

前記経路計算装置は、さらに、ノード情報処理部を有し、

前記ノード情報処理部は、前記ノードID処理部が既に発行したノードIDについて、その発行後に前記トポロジ情報の変更が通知されたときには、発行されたノードIDのうちの前記トポロジ情報の変更差分に該当するノードIDを差分更新することを特徴とする請求項1に記載の経路計算装置。

**【請求項 3】**

前記ノードID処理部は、ノードIDを発行するときに、そのノードIDと、そのノードに関する経路計算に使用される情報の格納先となるメモリアドレスとを対応づけて前記記憶手段に記憶し、

前記経路計算部は、前記ノードID処理部が対応づけたデータを参照して、ノードIDから、そのノードに関する経路計算に使用される情報を取得することを特徴とする

請求項1または請求項2に記載の経路計算装置。

**【請求項 4】**

前記経路計算装置は、ノードIDを引数とする所定関数の出力値であるメモリアドレスに、ノードに関する経路計算に使用される情報をノードごとに前記記憶手段に記憶することを特徴とする

請求項1または請求項2に記載の経路計算装置。

**【請求項 5】**

ノードアドレスで識別されるノードがリンクで接続されるトポロジ情報から、ノードごとに経路計算を行う経路計算装置による経路計算方法であって、

前記経路計算装置は、アドレスID索引データおよびIDアドレス索引データを記憶する記憶手段と、ノードID処理部と、アドレスID変換部と、経路計算部と、IDアドレス変換部とを有し、

前記ノードID処理部は、前記トポロジ情報のノードアドレスごとに、ノードの識別情報としてのノードIDをノードアドレスと1:1対応するように発行し、その発行した結果を、ノードIDからノードアドレスを検索するための前記IDアドレス索引データと、ノードアドレスからノードIDを検索するための前記アドレスID索引データとにそれぞれ格納し、

10

20

30

40

50

前記アドレスID変換部は、前記トポロジ情報のノードごとに、そのノードに関する経路計算に使用される情報のうちのノードアドレスを、前記アドレスID索引データを参照して前記ノードID処理部が発行したノードIDへと変換し、

前記経路計算部は、前記アドレスID変換部が変換した前記トポロジ情報をもとに、変換したノードIDをノードの識別情報として前記トポロジ情報の経路を計算し、

前記IDアドレス変換部は、前記経路計算部の計算結果を読み取り、その計算結果に含まれる各ノードIDについて、前記IDアドレス索引データを参照してノードアドレスへと変換することで、ルーティングテーブルを作成することを特徴とする

経路計算方法。

【請求項6】

10

前記経路計算装置は、さらに、ノード情報処理部を有し、

前記ノード情報処理部は、前記ノードID処理部が既に発行したノードIDについて、その発行後に前記トポロジ情報の変更が通知されたときには、発行されたノードIDのうちの前記トポロジ情報の変更差分に該当するノードIDを差分更新することを特徴とする

請求項5に記載の経路計算方法。

【請求項7】

前記ノードID処理部は、ノードIDを発行するときに、そのノードIDと、そのノードに関する経路計算に使用される情報の格納先となるメモリアドレスとを対応づけて前記記憶手段に記憶し、

前記経路計算部は、前記ノードID処理部が対応づけたデータを参照して、ノードIDから、そのノードに関する経路計算に使用される情報を取得することを特徴とする

20

請求項5または請求項6に記載の経路計算方法。

【請求項8】

前記経路計算装置は、ノードIDを引数とする所定関数の出力値であるメモリアドレスに、ノードに関する経路計算に使用される情報をノードごとに前記記憶手段に記憶することを特徴とする

請求項5または請求項6に記載の経路計算方法。

【請求項9】

請求項5ないし請求項8のいずれか1項に記載の経路計算方法を、コンピュータである前記経路計算装置に実行させるための経路計算プログラム。

30

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、経路計算装置、経路計算方法、および、経路計算プログラムに関する。

【背景技術】

【0002】

ルータなどが実行するデータ転送処理において、パケットの宛先に向かって正しい転送先を選択するために、ルーティングテーブルが参照される。ルーティングテーブルは、ノードをリンクで接続した集合であるネットワークトポロジの情報から、ルータごとに計算される。ネットワークトポロジは、時間の経過とともに変動する情報であるので、一度計算すればよいものではなく、ネットワークトポロジの更新を契機に、何度も再計算される。よって、経路計算処理は、高性能化（処理の高速化、使用メモリの省メモリ化）が求められる処理である。

40

【0003】

ネットワークトポロジを構成する各ノードには、IPアドレスなどのノードを特定するためのノードアドレスが割り当てられている。そして、経路計算処理において各ノードを参照するための検索キーとして、このノードアドレスが使用される。

【0004】

ノードアドレスの取りうる値が非常に広範囲であるので、ノードアドレスを管理するデータ構造として、非特許文献1などに記載されている平衡二分木を利用した方法などが、

50

利用される。平衡二分木の各ノードにはノードアドレスが割り当てられているので、平衡二分木を辿ることによって、検索キーの値から検索対象を絞り込み、絞り込んだ検索対象の中から検索キーと合致するキーを持つ情報を探し出すことで情報を検索する。

【先行技術文献】

【非特許文献】

【0005】

【非特許文献1】microsoft、“An Extensive Examination of Data Structures Using C # 2.0”、[online]、[平成21年11月4日検索]、インターネット<URL:http://msdn.microsoft.com/en-us/library/ms379573%28VS.80%29.aspx>

【発明の概要】

10

【発明が解決しようとする課題】

【0006】

平衡二分木などのデータ構造を用いるときには、検索キーの値から検索対象を求める処理が、1回の検索処理では収まらないので、検索処理の計算コストが大きくなってしまふ。この計算コストは、そのままノード検索を呼び出す経路計算処理のコストへと影響するため、経路計算処理の計算量の増大を招いてしまふ。

【0007】

一方、検索キーから1回の検索で情報を検索する方法として、一次元配列を用いることにより、検索キーの値から検索対象を求める処理を、1回の検索処理(配列の添え字を入力すると、その添え字に対応する配列要素の値を出力する)に抑えることができる。しかし、検索キーにノードアドレスを使用する場合、検索キーとなりうる値が非常に広範囲の値になるため、確保すべきメモリ領域が膨大なものとなるという問題が生じる。つまり、計算コストの削減への代償として、使用メモリ量が増大してしまふ。

20

【0008】

よって、経路計算処理におけるトポロジのノード検索処理は、計算コストを削減するとともに、使用メモリ量も削減する必要がある。以上説明した従来の技術では、計算コスト削減と、使用メモリ量削減とをバランスよく行う手法は、提案されていない。

【0009】

そこで、本発明は、前記した問題を解決し、経路計算処理におけるトポロジのノードアクセス処理において、計算コスト削減と、使用メモリ量削減とをバランスよく実現することを、主な目的とする。

30

【課題を解決するための手段】

【0010】

前記課題を解決するために、本発明は、ノードアドレスで識別されるノードがリンクで接続されるトポロジ情報から、ノードごとに経路計算を行う経路計算装置であって、前記経路計算装置が、アドレスID索引データと、IDアドレス索引データとを記憶手段に記憶し、ノードID処理部と、アドレスID変換部と、経路計算部と、IDアドレス変換部とを有し、前記ノードID処理部が、前記トポロジ情報のノードアドレスごとに、ノードの識別情報としてのノードIDをノードアドレスと1:1対応するように発行し、その発行した結果を、ノードIDからノードアドレスを検索するための前記IDアドレス索引データと、ノードアドレスからノードIDを検索するための前記アドレスID索引データとにそれぞれ格納し、前記アドレスID変換部が、前記トポロジ情報のノードごとに、そのノードに関する経路計算に使用される情報のうちのノードアドレスを、前記アドレスID索引データを参照して前記ノードID処理部が発行したノードIDへと変換し、前記経路計算部が、前記アドレスID変換部が変換した前記トポロジ情報をもとに、変換したノードIDをノードの識別情報として前記トポロジ情報の経路を計算し、前記IDアドレス変換部が、前記経路計算部の計算結果を読み取り、その計算結果に含まれる各ノードIDについて、前記IDアドレス索引データを参照してノードアドレスへと変換することで、ルーティングテーブルを作成することを特徴とする。

40

さらに、本発明は、前記経路計算装置に、前記各処理を実行させることを特徴とする経

50

路計算方法である。

さらに、本発明は、前記経路計算装置に、前記各処理を実行させるための経路計算プログラムである。

【0011】

これにより、経路計算時にノード情報を検索する処理がノードアドレスからでなくノードIDでできるようになり、ノード情報の検索時間が短縮するとともに、検索空間が狭いノードIDを用いることで、省メモリ化が実現できる。

【0012】

本発明は、前記経路計算装置が、さらに、ノード情報処理部を有し、前記ノード情報処理部が、前記ノードID処理部が既に発行したノードIDについて、その発行後に前記トポロジ情報の変更が通知されたときには、発行されたノードIDのうちの前記トポロジ情報の変更差分に該当するノードIDを差分更新することを特徴とする。

10

【0013】

これにより、トポロジ変更のたびに全ノードアドレスに対しノードIDを割り当てる処理を省くことができ、ノードIDの割り当て時間を短縮することが可能になる。

【0014】

本発明は、前記ノードID処理部が、ノードIDを発行するときに、そのノードIDと、そのノードに関する経路計算に使用される情報の格納先となるメモリアドレスとを対応づけて前記記憶手段に記憶し、前記経路計算部が、前記ノードID処理部が対応づけたデータを参照して、ノードIDから、そのノードに関する経路計算に使用される情報を取得

20

【0015】

これにより、ノードIDをもとにノード情報のメモリアドレスを特定できるため、ノード情報を検索する処理時間を短縮することが可能となる。

【0016】

本発明は、前記経路計算装置が、ノードIDを引数とする所定関数の出力値であるメモリアドレスに、ノードに関する経路計算に使用される情報をノードごとに前記記憶手段に記憶することを特徴とする。

【0017】

これにより、ノードIDをもとにノード情報のメモリアドレスを計算できるため、ノード情報を検索する処理時間を短縮することが可能となる。

30

【発明の効果】

【0018】

本発明によれば、経路計算処理におけるトポロジのノードアクセス処理において、計算コスト削減と、使用メモリ量削減とをバランスよく実現することができる。

【図面の簡単な説明】

【0019】

【図1】本発明の一実施形態に関する通信システムを示す構成図である。

【図2】本発明の一実施形態に関する経路計算装置による経路計算対象であるトポロジ情報を示す説明図である。

40

【図3】本発明の一実施形態に関する通信システムの経路計算装置を示す構成図である。

【図4】本発明の一実施形態に関するヒープを用いるID発行管理データを示す説明図である。

【図5】本発明の一実施形態に関するアドレスID索引データを示す説明図である。

【図6】本発明の一実施形態に関する経路計算部による経路の計算結果の一例を示す説明図である。

【発明を実施するための形態】

【0020】

以下、本発明の一実施形態を、図面を参照して詳細に説明する。

【0021】

50

図 1 は、通信システムを示す構成図である。

通信システムは、経路計算装置 1 と、データ転送装置 3 とがネットワークで接続されて構成される。1 台の経路計算装置 1 は、1 台以上の（図 1 では 3 台の）データ転送装置 3 のルーティングテーブル 3 2 を計算して、各データ転送装置 3 に配布する。

なお、通信システムの各装置（経路計算装置 1、データ転送装置 3）は、それぞれ CPU と記憶手段（メモリなど）とを備えるコンピュータとして構成される。ここで、経路計算装置 1 とデータ転送装置 3 とは、同一筐体として構成してもよいし、別々の筐体として構成してもよい。

経路計算装置 1 のインタフェース 1 9 は、経路計算装置 1 内の各処理部と、経路計算装置 1 とは別の外部装置（データ転送装置 3 など）との間でデータの入出力を行う。

データ転送装置 3 のインタフェース 3 9 は、データ転送装置 3 内の各処理部と、データ転送装置 3 とは別の外部装置（経路計算装置 1 など）との間でデータの入出力を行う。

#### 【 0 0 2 2 】

各データ転送装置 3 は、経路管理部 3 1 と、ルーティングテーブル 3 2 と、データ転送部 3 3 とを備える通信装置（ルータ、スイッチなど）として構成される。

経路管理部 3 1 は、経路計算装置 1 から受信した自装置の経路情報を受け取り、ルーティングテーブル 3 2 として記憶手段に記憶する。

ルーティングテーブル 3 2 は、データ転送部 3 3 のデータ転送処理時に参照され、到着したパケットの宛先に到達するためのネクストホップの情報を含む（詳細は、表 7 で後記する）。

データ転送部 3 3 は、ルーティングテーブル 3 2 を参照して、受信したパケットを転送する。

#### 【 0 0 2 3 】

図 2 は、経路計算装置 1 による経路計算対象であるトポロジ情報 9（図 3 参照）を示す説明図である。

#### 【 0 0 2 4 】

図 2（a）に示すように、トポロジ情報 9 は、楕円で示す各ノードと、各ノード間を接続するリンクとで構成されるグラフである。

各ノード内には、「192.168.1.0」のようなノードアドレスを表記する。なお、1 つのノードは、1 つ以上のネットワークデバイスに対応する。ネットワークデバイスとは、経路の分岐点となりうるデータ転送装置 3 という 1 台の装置や、データ転送装置 3 内の 1 つのインタフェース 3 9（図 1 参照、IP アドレスの割り当て対象）や、データ転送装置 3 間を接続する 1 つの下位レイヤネットワーク（ネットワークケーブルなど）として構成される。つまり、トポロジ情報 9 におけるノードとは、経路計算アルゴリズムにおける経路が分岐する 1 つの構成単位に対応する。

各リンクには、そのリンクをデータ通信で利用するときのコストを併記する。このコストは、ノード間の接続関係や経路への選ばれやすさを示す指標であり、コストの数値が高いほど経路として選ばれにくいように、経路計算される。

#### 【 0 0 2 5 】

図 2（b）は、図 2（a）のトポロジ情報 9 を構成するノードごとのノード情報を、表形式で示す。本実施形態では、ノードごとのそのノードに関する情報を総称して、「ノード情報」とする。

なお、ノード情報は、ノードごとの識別情報としてのノードアドレスと、そのノードから 1 回のリンクで辿ることができる隣接ノードのノードアドレスと、その隣接ノードへのリンクのリンクコストとを対応づけて構成する。

例えば、1 行目のノードは、自身に割り当てられているノードアドレスが「192.168.1.0」であり、1 つめの隣接ノードとしてリンクコスト「1」で到達する隣接ノード「192.168.2.0」と、2 つめの隣接ノードとしてリンクコスト「3」で到達する隣接ノード「192.168.3.0」と、それぞれ接続されている。

#### 【 0 0 2 6 】

ノードアドレスは、ネットワーク全体でユニークに割り当てられるので、あるノードと別のノードとを区別するためのノードの識別情報として作用する。

しかし、ノードアドレスが取りうる値であるアドレス空間は、ノードアドレスとしてIPv4アドレスを使用するときには32ビット長（「0.0.0.0」から「255.255.255.255」まで）であるので、2の32乗通り約43億と非常に大きな範囲になってしまう。

よって、ノードアドレスをノードの識別情報として用いると、そのノードアドレスの検索処理は、その検索対象の空間が非常に広大であることにより、CPU使用率および使用メモリ効率の両面から、非効率な検索処理となってしまう。

#### 【0027】

そこで、本実施形態では、ノード情報へアクセス（リードアクセスだけでなく、ライトアクセスも）するためのノードの識別情報として、ノードアドレスをそのまま用いるのではなく、ノードアドレスと1:1対応であるノードIDを用いることにする。

ノードIDは、ノードアドレスと1:1対応であるため、ノードの識別情報として作用する。さらに、ノードIDには、整数連番値などの効率的な値（割り当てられた値の分布の密度が高く、連続する割り当てられた値どうしの間隔が狭い値）が割り当てられるので、効率的なノードアクセス処理を実現することができる。

そして、ノードIDを割り当てるノードの数の上限値が、利用可能なメモリ空間値（物理メモリでもよいし、仮想メモリでもよい）を超えないようにノードIDを発行することにより、メモリを節約することができる。

#### 【0028】

図3は、通信システムの経路計算装置1を示す構成図である。

経路計算装置1は、経路計算装置1の記憶手段に格納される各データ構造（ノード情報管理テーブル11、アドレスID索引データ12、IDアドレス索引データ13、ID発行管理データ14）と、経路計算装置1の記憶手段から読み出されるプログラムがCPUによって実行されることで構成される各処理部（ノードID処理部21、ノード情報処理部22、アドレスID変換部23、経路計算部24、IDアドレス変換部25、結果出力部26）とを有する。

#### 【0029】

ノードID処理部21は、1つのノードアドレスに対して、1つのノードIDを割り当てるとともに、その割り当てたノードアドレスを各データ構造（ノード情報管理テーブル11、アドレスID索引データ12、IDアドレス索引データ13、ID発行管理データ14）で管理する。

#### 【0030】

##### 【表1】

#### 14 ID発行管理データ

発行順序	手法1 (整数連番値)	手法2 (整数連番値/2)	手法3 (整数飛び値)
1番目	1	0.5	1
2番目	2	1	2
3番目	3	1.5	3
4番目	4	2	5
:	:	:	:

#### 【0031】

表1は、ID発行管理データ14におけるノードIDの新規発行の手法を示す表である。

手法1（表の第2列）は、整数連番値になるように割り当てる方法である。これは1から順番に空いている番号を割り当てていく方法である。ノードIDを1、2、3、...と割り当てていく。

10

20

30

40

50

手法 2 (表の第 3 列) は、定数倍 (例えば 2 倍) すると整数連番値になる値である。0 . 5、1、1 . 5、... と番号を割り当てていく。この番号は 2 倍すると 1、2、3、... となり整数連番値になる。

手法 3 (表の第 4 列) は、整数連番値ではあるが、一部の値が飛んでいるような割り当て方である。1、2、3、... と連番で並んでいるが、4 がとんで 5 からまた連番値になるよう割り当てられる方法である。

【 0 0 3 2 】

また、明示的にメモリ容量を考慮した割り当て方としては、トライアンドエラーを繰り返す方法がある。トライアンドエラー方式では、まずノード ID を割り当て、割り当てたノード ID から検索されたメモリアドレスが参照不可なものであれば、再度割り当て直し、参照可能であれば、その値を使用するというものである。なお、ここで挙げた例の他にも前記のようにノード ID から参照可能なメモリアドレスを検索可能なものであれば、割り当てを限定しない。

10

【 0 0 3 3 】

トポロジ情報が変更されたとき、特にノードの追加や削除が起こった場合、割り当てたノード ID を更新する必要がある。このとき、トポロジ変更が発生するたびに、変更箇所だけでなく、全てノード ID を再割り当てする方法があるが、この方法では、トポロジ変更対象でないノードについてもノード ID を割り当て直すことになり、変更に伴う負荷が大きくなってしまう。

そこで、トポロジの変更箇所 (追加箇所、削除箇所) のみを対象として、ノード ID の変更処理を行うという方法がある。この方法を行うには、ノード ID の割り当てを記録し、トポロジ変更が生じるとトポロジの差分情報をチェックする。ノードが追加されたときには、空き番号となっている番号をノード ID として割り当てる。そして、ノードが削除されたときには割り当てられていたノード ID を削除する。

20

【 0 0 3 4 】

このような、ノード情報処理部 2 2 が実行するノード ID の差分変更処理は、線形サーチなどの手段で行うと時間がかかることが予想される。この処理の高速化を行う方法として、ノード ID の割り当て状況や空き番号の管理を、ヒープを用いることで効率化できる。

【 0 0 3 5 】

図 4 は、ヒープを用いる ID 発行管理データ 1 4 を示す説明図である。ヒープはノード ID の割り当て候補値の集合のうち、最小値 (図では、ノード ID = 4) を出力するとき使用されるデータ構造である。このデータ構造を用いて、空き番号のうちの最小値を出力することで、できるだけノード ID が連番になるよう割り当てることが可能になる。そして、ノードが削除されたとき、このヒープに削除されたノードのノード ID を入れることで空き番号を管理する。このように、ヒープを用いることにより、ノード ID が飛び飛びの値になることを防ぐことができる。

30

【 0 0 3 6 】

図 3 のノード情報処理部 2 2 は、トポロジ情報 9 を構成するノードの情報であるノード情報 (新規登録分、追加分、削除分など) を受信すると、それらをもとにノード情報管理テーブル 1 1 へと更新する。

40

ノード情報管理テーブル 1 1 は、ノード ID と、そのノード ID が示すノード情報 (またはそのノード情報の格納先へのポインタ) とを対応づけるテーブルである。以下、ノード情報管理テーブル 1 1 の構成例として、2 つの例を説明する。

【 0 0 3 7 】



## 【表 2】

## 11 ノード情報管理テーブル

ノードID	ノード情報
1	(隣接ノードID=2, リンクコスト=1) (隣接ノードID=3, リンクコスト=3)
2	(隣接ノードID=1, リンクコスト=1) (隣接ノードID=4, リンクコスト=5) (隣接ノードID=5, リンクコスト=1)
3	(隣接ノードID=1, リンクコスト=3) (隣接ノードID=6, リンクコスト=10)
4	(隣接ノードID=2, リンクコスト=5)
5	(隣接ノードID=2, リンクコスト=1)
6	(隣接ノードID=3, リンクコスト=10)
:	:

10

## 【0038】

表 2 は、ノード ID と、そのノード ID が示すノード情報とを配列で格納するノード情報管理テーブル 11 を示す。

ノード情報管理テーブル 11 の左列のノード ID は、検索キーとなる要素であり、ノード情報管理テーブル 11 を配列としたときの添え字（引数）となる。なお、表 2 では、ノード ID として 1 から始まる整数連番値を用いているが、ノード ID として任意の数字や文字列を用いてもよい。そのときには、配列の添え字には整数値以外の数字や文字列が代入されるので、ノード情報管理テーブル 11 の配列は、それらの添え字を許容する連想配列として構成される。

20

ノード情報管理テーブル 11 の右列のノード情報は、対応する左列で示されるノード ID のノード情報である。表 2 では、ノード情報として、隣接ノード ID と、その隣接ノードへのリンクコストとの組が、隣接ノードごとに記憶されている。

## 【0039】

## 【表 3】

## 11 ノード情報管理テーブル

ノードID	メモリアドレス
1	1000
2	1005
3	1010
4	1015
5	1020
6	1025
:	:

10

メモリアドレス	ノード情報
1000	ノードID=1の情報 (隣接ノードID=2, リンクコスト=1) (隣接ノードID=3, リンクコスト=3)
1005	ノードID=2の情報 (隣接ノードID=1, リンクコスト=1) (隣接ノードID=4, リンクコスト=5) (隣接ノードID=5, リンクコスト=1)
1010	ノードID=3の情報 (隣接ノードID=1, リンクコスト=3) (隣接ノードID=6, リンクコスト=10)
1015	ノードID=4の情報 (隣接ノードID=2, リンクコスト=5)
1020	ノードID=5の情報 (隣接ノードID=2, リンクコスト=1)
1025	ノードID=6の情報 (隣接ノードID=3, リンクコスト=10)
:	:

20

30

## 【0040】

表3は、ノードIDと、そのノードIDが示すノード情報の格納先を示すメモリアドレスとを配列で格納するノード情報管理テーブル11(表3の上部)と、そのメモリアドレスの位置に格納されているノード情報(表3の下部)とを示す。この方式では、ノードIDを検索キーとして、ノード情報へのポインタとなるメモリアドレスを検索するため、その検索されたメモリアドレスに格納されているノード情報の格納容量は、固定長でも可変長でもよい。

## 【0041】

以上、表2および表3を参照して、ノード情報管理テーブル11について、説明した。さらに、ノード情報が固定長であり、ノードIDとして整数連番値が割り当てられるときなど、所定の規則性を有するときには、ノードIDごとのレコードを有するノード情報管理テーブル11を用意する代わりに、ノードIDの値を入力パラメータとして、所定の計算式を計算することにより、ノード情報の格納先を示すメモリアドレスを取得してもよい。

40

例えば、表3に示すノード情報管理テーブル11は、計算式「(メモリアドレス) = 1000 + (ノードID - 1) × 5」によって、ノードIDの値に応じたメモリアドレスを取得することができる。なお、1000とは、ノード情報を格納するメモリ空間(表3の下部)の先頭位置である。

50

## 【 0 0 4 2 】

図 3 のアドレス ID 変換部 2 3 は、アドレス ID 索引データ 1 2 をもとに、ノードアドレスからノード ID への変換を行う。つまり、アドレス ID 変換部 2 3 は、ノード情報に記載されている情報のうち、ノードアドレスが記載されている箇所を対応するノード ID に書き換える。

## 【 0 0 4 3 】

アドレス ID 索引データ 1 2 は、ノードアドレスを検索キーとして、対応するノード ID を取得するためのデータ構造である。以下、アドレス ID 索引データ 1 2 の例を 2 つ挙げる。

## 【 0 0 4 4 】

## 【表 4】

## 12 アドレスID索引データ

IPアドレス (検索キー)	ノードID (検索結果)
192.168.1.0	1
192.168.2.0	2
192.168.3.0	3
192.168.4.0	4
192.168.5.0	5
192.168.6.0	6
:	:

10

20

## 【 0 0 4 5 】

表 4 は、IP アドレスを 8 ビット 4 区切り表記としたときの、アドレス ID 索引データ 1 2 を示す表である。アドレス ID 索引データ 1 2 は、ノードアドレスである IP アドレスを検索キーとして、対応するノード ID を検索するためのデータ構造である。検索キーが IP アドレスという文字列であるので、アドレス ID 索引データ 1 2 は、IP アドレスを添え字とする連想配列として実現される。連想配列は、添え字である検索キーの値と、その検索キーに対する情報が格納されているメモリアドレスの関係が状況に応じて動的に最適化される（詳細は、後記する図 5 の説明を参照）。

30

アドレス ID 変換部 2 3 は、IP アドレス「192.168.1.0」が検索キーとして入力されると、アドレス ID 索引データ 1 2 の 1 行目のレコードを参照して、対応するノード ID 「1」を得る。同様に、アドレス ID 変換部 2 3 は、IP アドレス「192.168.3.0」が検索キーとして入力されると、アドレス ID 索引データ 1 2 の 3 行目のレコードを参照して、対応するノード ID 「3」を得る。

## 【 0 0 4 6 】

## 【表 5】

## 12 アドレスID索引データ

IPアドレス(10進数表記) (検索キー)	ノードID (検索結果)
3232235776	1
3232236032	2
3232236288	3
3232236544	4
3232236800	5
3232237056	6
:	:

10

## IPアドレスの表記変換

IPアドレス (8ビット4区切り表記)	IPアドレス (10進数表記)
192.168.1.0	3232235776
192.168.2.0	3232236032
192.168.3.0	3232236288
192.168.4.0	3232236544
192.168.5.0	3232236800
192.168.6.0	3232237056
:	:

20

## 【0047】

表5は、IPアドレスを10進数表記(情報キー)に変換したときの、アドレスID索引データ12を示す表(表5の上部)と、その10進数変換の計算結果の表(表5の下部)である。

30

10進数の計算方法であるが、ノードアドレス「a.b.c.d」に対し、「 $a*256^3+b*256^2+c*256^1+d$ 」という計算(演算子「\*」は乗算、演算子「^」はべき乗を示す)を行った結果がノードアドレス「a.b.c.d」を10進化した値である。

このように、IPアドレスを10進数の情報キーとして表記することにより、アドレスID索引データ12を連想配列ではなく通常の配列(添え字が整数値である配列)として実現できるうえに、IPアドレス同士の大小比較が可能となる。

## 【0048】

図5は、アドレスID索引データを示す説明図である。表4, 5で説明したアドレスID索引データ12は、例えば、以下の2種類の実装方法のいずれかとして実現される。

## 【0049】

40

図5(a)に記載されているアドレスID索引データ12のデータ構造は、非特許文献1などに記載されている平衡二分木と呼ばれる実装方法である。この実装方法で管理する方法では、ノードごとに、そのノードのIPアドレスから計算した情報キー(10進数表記)と、ノードIDとの組の情報を保持する。そして、アドレスID索引データ12は、ノード間のリンクを、上部が親側、下部が子側とするツリー構造として構成する。

ここで、所定のノードの情報キーと、その所定のノードの子ノードの情報キーとの関係について、「(所定のノードの左側のリンクから辿る子ノードの情報キー) < (所定のノードの情報キー) < (所定のノードの右側のリンクから辿る子ノードの情報キー)」という大小関係を有する。

## 【0050】

50

例えば、検索キー「3232236288」に該当するノードを検索するときには、以下の手順により、目的のノードへと順に辿る。

手順(1)：検索キー「3232236288」が、ルートノードの情報キー「3232236544」と一致せず、かつ、それより小さいので、左側のリンクを辿り、ルートノードの子ノードへと進む(図ではリンクを辿るところは、実線の矢印で表記し、辿らないところは、点線の矢印で表記した)。

手順(2)：検索キー「3232236288」が、子ノードの情報キー「3232236032」と一致せず、かつ、それより大きいので、右側のリンクを辿り、ルートノードの孫ノードへと進む。

手順(3)：検索キー「3232236288」が、孫ノードの情報キー「3232236288」と一致するので、ここで検索を完了する。

10

#### 【0051】

図5(b)に記載されているアドレスID索引データ12のデータ構造は、ハッシュと呼ばれる実装方法である。この方法では、検索キーとして入力された、ノードのIPアドレスから計算した情報キー(10進数表記)に対して、ハッシュ関数と呼ばれる関数を計算し、その計算結果(ハッシュ値)をもとにノードを検索する手法である。なお、ハッシュ値が同じであるノードが複数存在するときには、それらのノードをリストで接続することで、所望のノードを検索できる。例えば、図5(b)では、ハッシュ関数として、情報キーを1000で割ったときの余りの値を用いている。

#### 【0052】

20

例えば、検索キー「3232236288」に該当するノードを検索するときには、以下の手順により、目的のノードへと順に辿る。

手順(1)：検索キー「3232236288」のハッシュ値「288」に該当する検索結果として、リストの先頭要素である情報キー「3729340288」のエントリを得る。しかし、このエントリの情報キーは、検索キー「3232236288」と一致しないので、リストの次の要素へと進む。

手順(2)：ハッシュ値「288」に該当するリストの2番目の要素として、情報キー「1029383288」のエントリを得る。しかし、このエントリの情報キーは、検索キー「3232236288」と一致しないので、リストの次の要素へと進む。

手順(3)：ハッシュ値「288」に該当するリストの3番目の要素として、情報キー「3232236288」のエントリを得る。このエントリの情報キーは、検索キー「3232236288」と一致するので、ここで検索を完了する。

30

#### 【0053】

図6(a)は、図2(a)で示したIPアドレス表記のトポロジ情報9に対して、アドレスID変換部23がアドレスID索引データ12を参照して、ノードアドレスをノードIDへとノードごとに変換した結果を示す。図6(a)では、ノードの識別情報を書き換えただけなので、トポロジ情報9のノードやリンクの構成は、変更していない。

#### 【0054】

そして、図3の経路計算部24は、図6(a)で示したような、アドレスID変換部23がノードIDに書き換えたノード情報を含めたトポロジ情報9を用いて、経路を計算する。

40

図6(b)は、経路計算部24による経路の計算結果の一例として、ノードID=1のノードを起点として、他の各ノードへの経路情報を示す。

#### 【0055】

ここで、経路計算部24は、経路計算処理において、ノード情報ごとに割り当てられるノードの識別情報として、低効率なノードアドレスではなく高効率なノードIDを参照することができるため、ノード情報管理テーブル11から高速にノード情報を検索して取得することができる。

さらに、経路計算部24によるノード情報の検索動作が多いほど、計算時間は短縮される。検索動作が多くなる状況として、複数のデータ転送装置3の経路情報を一台で計算す

50

る場合がある。経路計算が終了すると、ノード間の経路情報が得られる。なお、経路計算部 24 による経路計算アルゴリズムは、例えば、ダイクストラアルゴリズムを用いることができる。

【0056】

図 3 の ID アドレス変換部 25 は、ID アドレス索引データ 13 をもとに、経路計算部 24 による経路計算結果 (図 6 (b)) において、ノードの識別情報をノード ID からノードアドレスへの変換を行う。ID アドレス変換部 25 によるノード ID からノードアドレスへの変換処理は、アドレス ID 変換部 23 によるノードアドレスからノード ID への変換処理に対する逆変換処理である。このように、逆変換処理が可能なのは、ノード ID とノードアドレスとが 1 : 1 対応しているためである。

これにより、データ転送装置 3 は、従来のデータ転送処理と同様に、ノードアドレスを元にしたフォワーディングが実行できるので、既に多くの台数が設置および稼動しているデータ転送装置 3 への変更や停止を行わずに済む。

【0057】

【表 6】

13 IDアドレス索引データ

ノードID (検索キー)	IPアドレス (検索結果)
1	192.168.1.0
2	192.168.2.0
3	192.168.3.0
4	192.168.4.0
5	192.168.5.0
6	192.168.6.0
:	:

【0058】

表 6 は、ID アドレス変換部 25 が参照する ID アドレス索引データ 13 を示す表である。ID アドレス索引データ 13 は、アドレス ID 索引データ 12 と等価なデータではあるが、ID アドレス索引データ 13 の検索キーは、ノード ID であるので、検索キーが IP アドレスであるアドレス ID 索引データ 12 よりも、高速な検索処理を実現することができる。

例えば、ID アドレス索引データ 13 を参照すると、ノード ID 「1」のノードアドレスは「192.168.1.0」であり、ノード ID 「2」のノードアドレスは「192.168.2.0」であり、ノード ID 「3」のノードアドレスは「192.168.3.0」であることが、それぞれ取得できる。このように変換した経路情報をデータ転送装置 3 における次の転送先の設定に使用する。

【0059】

10

20

30

## 【表 7】

## 32 ルーティングテーブル

宛先 IPアドレス	ネクストホップ IPアドレス	出力先 インタフェース
192.168.1.0	(自装置)	(自装置)
192.168.2.0	192.168.2.0	A
192.168.3.0	192.168.3.0	B
192.168.4.0	192.168.2.0	A
192.168.5.0	192.168.2.0	A
192.168.6.0	192.168.3.0	B
:	:	:

10

## 【0060】

表 7 は、ID アドレス変換部 25 による図 6 ( b ) からの変換結果である、ルーティングテーブル 32 を示す表である。図 6 ( b ) に記載された各経路について、ノード ID から IP アドレスへの変換が行われている。

そして、図 3 の結果出力部 26 は、表 7 に示すようなルーティングテーブル 32 を、経路計算結果として各データ転送装置 3 に出力する。これにより、データ転送装置 3 は、ノード ID が経路計算に用いられていることを全く意識せず、従来と同じようにパケットのデータ転送を実行することができる。

20

## 【0061】

以上説明した本実施形態は、コンピュータネットワークにおける経路計算のときに、経路計算アルゴリズムから参照されるトポロジ情報 9 のノード情報のアクセスを高速化する技術である。

従来では、ノードの識別情報として IP アドレスを使用するため、検索空間が非常に広範囲となってしまう、確保すべきメモリ領域が膨大なものとなるという問題があった。

これに対し、本実施形態では、経路計算部 24 の経路計算の前処理として、アドレス ID 変換部 23 がノードの識別情報を IP アドレスからノード ID に変換し、経路計算の後処理として、ID アドレス変換部 25 がノードの識別情報をノード ID から IP アドレスに変換する。これにより、経路計算部 24 は、経路計算に伴うノード情報のアクセス処理において、検索空間が狭いノード ID の検索空間だけを検索すればよいため、確保すべきメモリ領域を大幅に節約することができる。

30

## 【0062】

## 【表 8】

## 12 アドレスID索引データ(CIDR対応)

IPアドレス/マスク長 (検索キー)	ノードID (検索結果)
192.168.1.0/24	1
192.168.1.0/28	2
192.168.1.0/30	3
192.168.2.0/24	4
192.168.2.0/28	5
192.168.3.0/24	6
:	:

10

## 13 IDアドレス索引データ(CIDR対応)

ノードID (検索キー)	IPアドレス/マスク長 (検索結果)
1	192.168.1.0/24
2	192.168.1.0/28
3	192.168.1.0/30
4	192.168.2.0/24
5	192.168.2.0/28
6	192.168.3.0/24
:	:

20

## 【0063】

表 8 は、ノードアドレスとして、C I D R (Classless Inter-Domain Routing) に対応したときの、各索引データを示す表である。本実施形態では、ノードアドレスとして、I P v 4 アドレスを用いたが、割り当て可能なネットワークの識別情報なら、I P v 6 アドレスや、アドレスとそのアドレスの付加情報(マスク長)とを組とした情報を用いてもよい。C I D R は、I P アドレスとそのマスク長との組の情報をネットワークの識別情報として用いる方式である。

30

よって、表 8 に示すように、アドレス I D 索引データ 1 2、I D アドレス索引データ 1 3 などの各データ構造を、I P v 4 アドレスから I P アドレスとそのマスク長との組の情報へと置き換えることで、経路計算部 2 4 による任意の経路情報の計算処理に、ノード I D を用いることができる。

## 【符号の説明】

## 【0064】

- 1 経路計算装置
- 3 データ転送装置
- 1 1 ノード情報管理テーブル
- 1 2 アドレス I D 索引データ
- 1 3 I D アドレス索引データ
- 1 4 I D 発行管理データ
- 2 1 ノード I D 処理部
- 2 2 ノード情報処理部
- 2 3 アドレス I D 変換部
- 2 4 経路計算部
- 2 5 I D アドレス変換部

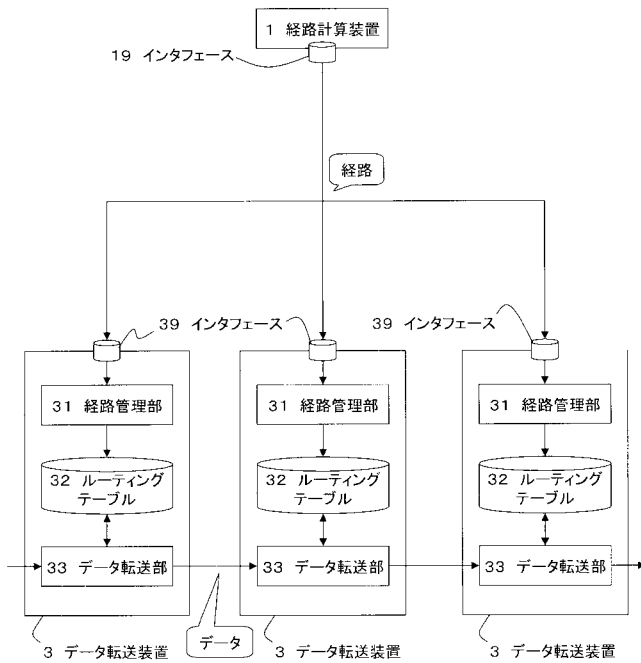
40

50



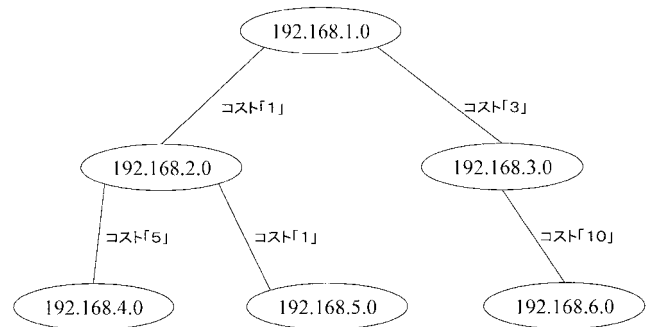
- 2 6 結果出力部
- 3 1 経路管理部
- 3 2 ルーティングテーブル
- 3 3 データ転送部

【 図 1 】



【 図 2 】

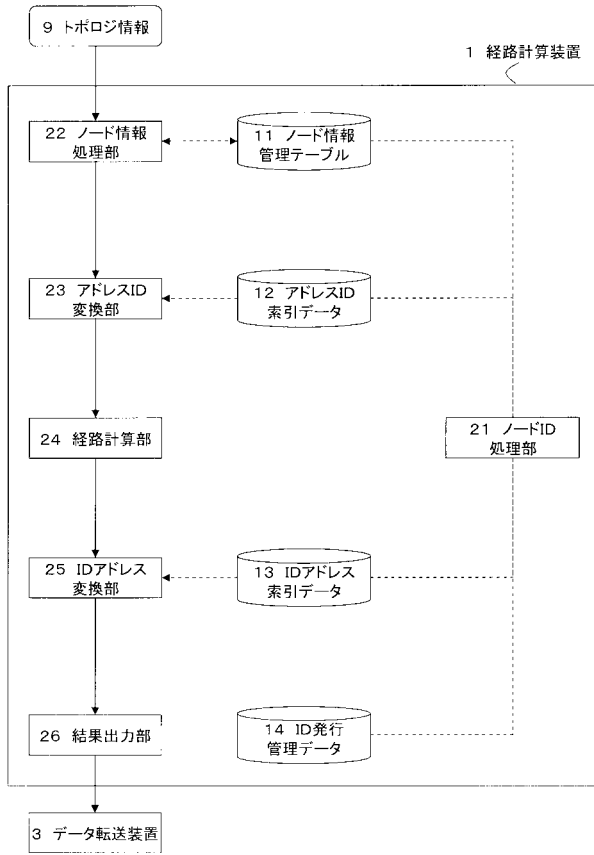
(a)トポロジ情報(IPアドレス表記)



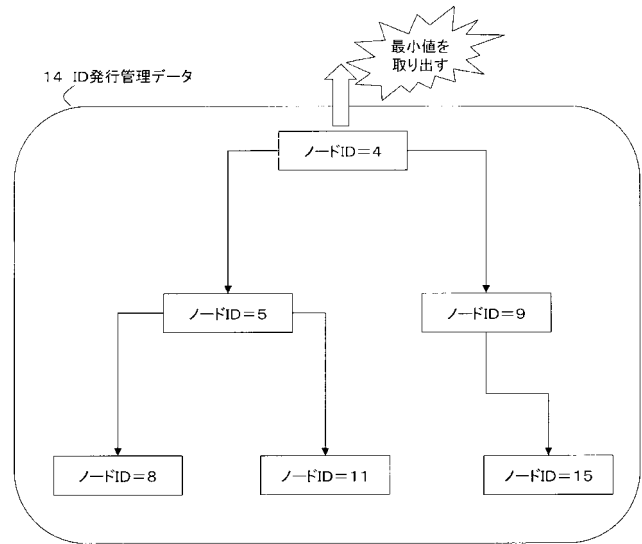
(b)トポロジ情報(テーブル形式)

ノード アドレス	隣接ノード アドレス	隣接ノードへの リンクコスト
192.168.1.0	192.168.2.0	1
	192.168.3.0	3
192.168.2.0	192.168.1.0	1
	192.168.4.0	5
	192.168.5.0	1
192.168.3.0	192.168.1.0	3
	192.168.6.0	10
192.168.4.0	192.168.2.0	5
192.168.5.0	192.168.2.0	1
192.168.6.0	192.168.3.0	10
:	:	:

【 図 3 】

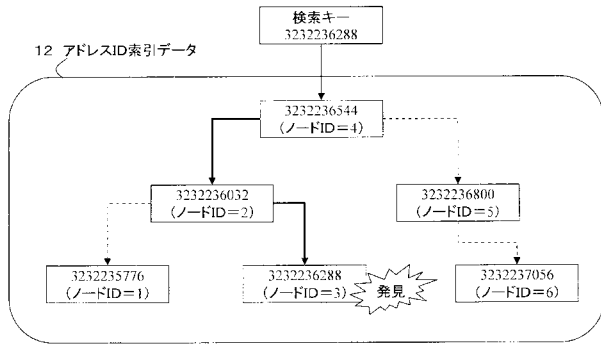


【 図 4 】

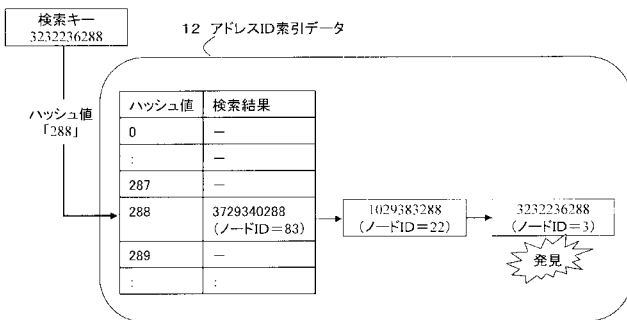


【 図 5 】

(a) 二分木実装

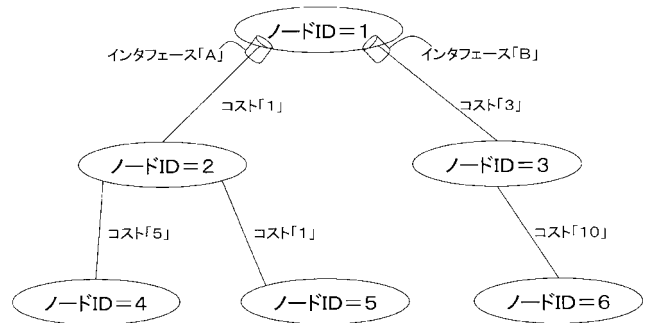


(b) ハッシュ実装



【 図 6 】

(a) トポロジ情報(ノードID表記)



(b) ルーティングテーブル(ノードID表記)

宛先ノードID	ネクストホップノードID	出力先インタフェース
1	(自装置)	(自装置)
2	2	A
3	3	B
4	2	A
5	2	A
6	3	B
:	:	: