

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5756049号  
(P5756049)

(45) 発行日 平成27年7月29日(2015.7.29)

(24) 登録日 平成27年6月5日(2015.6.5)

(51) Int.Cl. F I  
H O 4 L 12/761 (2013.01) H O 4 L 12/761

請求項の数 8 (全 14 頁)

<p>(21) 出願番号 特願2012-85942 (P2012-85942)                  (22) 出願日 平成24年4月4日(2012.4.4)                  (65) 公開番号 特開2013-219436 (P2013-219436A)                  (43) 公開日 平成25年10月24日(2013.10.24)                  審査請求日 平成26年7月28日(2014.7.28)</p>	<p>(73) 特許権者 000004226                  日本電信電話株式会社                  東京都千代田区大手町一丁目5番1号                  (74) 代理人 100107766                  弁理士 伊東 忠重                  (74) 代理人 100070150                  弁理士 伊東 忠彦                  (74) 代理人 100124844                  弁理士 石原 隆治                  (72) 発明者 松浦 洋                  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                  審査官 松崎 孝大</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

最終頁に続く

(54) 【発明の名称】 マルチキャスト経路計算方法及び装置

(57) 【特許請求の範囲】

【請求項1】

複数ノードを通信リンクを介して相互に接続可能としたネットワーク上で、ある始点ノードから他のすべてのノード内の部分集合である複数ノードまでのマルチキャスト経路を計算するためのマルチキャスト経路計算方法であって、

最初はマルチキャスト始点ノードのみからなるマルチキャストツリーを定義し、当該マルチキャストツリーから、まだ経路が決まっていないマルチキャスト終点ノード集合の1終点ノードにたどり着く最短経路を新ブランチとしてマルチキャストツリーに加え、この処理を最終的にすべてのマルチキャストツリーブランチが決まるまで繰り返すマルチキャストツリー生成アルゴリズムにおいて、

アルゴリズムの第1回目のルーチンはマルチキャスト始点ノードから始まり、そのノードから発するリンクを求め、そのすべてのリンクをブランチ候補としてブランチ候補記憶手段に格納し、当該ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出し、そのブランチ候補の終点がマルチキャスト終点の場合はマルチキャストを最終的に構成するブランチとしてブランチ記憶手段に格納し、

アルゴリズムの第2回目のルーチン以降は、直前のルーチンで前記ブランチ候補記憶手段から取り出した前記最短経路ブランチ候補の終点がマルチキャスト終点の場合は、当該最短経路ブランチ候補の始点を除いたすべてのノードから発するリンクを求め、そのすべてのリンクをブランチ候補として該ブランチ候補記憶手段に格納し、当該ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出し、そのブランチ候補

の終点がマルチキャスト終点の場合はマルチキャストを最終的に構成するブランチとしてブランチ記憶手段に格納し、

また、アルゴリズムの第2回目のルーチン以降で、直前のルーチンで取り出した前記最短経路ブランチ候補の終点がマルチキャスト終点でない場合は、当該最短経路ブランチ候補の終点から発するすべてのリンクを求め、当該最短経路ブランチ候補の後に個々のリンクを加えた結果できるすべてのブランチを前記ブランチ候補記憶手段に格納し、当該ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出し、そのブランチ候補の終点がマルチキャスト終点の場合はマルチキャストを最終的に構成するブランチとしてブランチ記憶手段に格納することを特徴とするマルチキャスト経路計算方法。

10

【請求項2】

前記ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出す手順において、

前記最短経路ブランチ候補の終点がマルチキャスト終点の場合は、当該ブランチ候補上の始点を除いたすべてのノードにマルチキャストツリーの構成要素になったという印("reached属性"=yes)を付与する

請求項1記載のマルチキャスト経路計算方法。

【請求項3】

前記ブランチ候補を前記ブランチ候補記憶手段に格納する手順において、

新規ブランチ候補の最終リンクとして選択されたリンクの終点ノードに前記印("reached属性"=yes)が付いている場合は、当該ブランチ候補を前記ブランチ候補記憶手段に格納しない

20

請求項2記載のマルチキャスト経路計算方法。

【請求項4】

ネットワーク上の各ノードに、"当該ノードまでの最短ブランチ候補とそのコスト"を属性として保持させ、両者のアルゴリズム開始時の初期値はブランチ候補 = 空集合、経路コスト = とするが、アルゴリズム走行後は当該属性にアルゴリズムの開始からその時点までにアルゴリズムが扱った当該ノードを終点に持つブランチ候補の中で最もコストが小さいブランチ候補とその経路コストを保持する

請求項1記載のマルチキャスト経路計算方法。

30

【請求項5】

前記ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出す手順において、

当該ブランチ候補の終点の"当該ノードまでのブランチ候補とそのコスト"属性からブランチ候補を抽出し、ブランチ候補 = 空集合にする

請求項4記載のマルチキャスト経路計算方法。

【請求項6】

新規ブランチ候補の最終リンクとして選択されたリンクの終点ノードの上記"当該ノードまでのブランチ候補とそのコスト"属性にすでにブランチ候補が格納されている場合は、当該属性に格納されている経路コストと新規ブランチ候補の経路コストを比較して新規ブランチ候補の当該ノードまでの経路コストが既存ブランチ候補の経路コストより小さい場合にのみ、前記ブランチ候補記憶手段から当該既存ブランチ候補を抽出し、新規ブランチ候補を前記ブランチ候補記憶手段に格納し、当該終点ノードの当該属性の既存ブランチ候補とその経路コストを新規ブランチとその経路コストに置き換える

40

請求項5記載のマルチキャスト経路計算方法。

【請求項7】

新規ブランチ候補の最終リンクとして選択されたリンクの終点ノードの上記"当該ノードまでのブランチ候補とそのコスト"属性にブランチ候補が格納されていない場合は、当該属性に格納されている経路コストと新規ブランチ候補の経路コストを比較して、当該ノードまでの新規ブランチ候補の経路コストが当該属性内の経路コストより小さい場合は、

50

新規ブランチ候補を前記ブランチ候補記憶手段に格納し、当該終点ノードの当該属性に新規ブランチとその経路コストを格納する  
請求項6記載のマルチキャスト経路計算方法。

【請求項8】

複数ノードを通信リンクを介して相互に接続可能としたネットワーク上で、ある始点ノードから他のすべてのノード内の部分集合である複数ノードまでのマルチキャスト経路を計算するためのマルチキャスト経路計算装置であって、

ブランチ候補を格納するブランチ候補記憶手段と、  
ブランチを格納するブランチ記憶手段と、

最初はマルチキャスト始点ノードのみからなるマルチキャストツリーを定義し、当該マルチキャストツリーから、まだ経路が決まっていないマルチキャスト終点ノード集合の1終点ノードにたどり着く最短経路を新ブランチとしてマルチキャストツリーに加え、この処理を最終的にすべてのマルチキャストツリーブランチが決まるまで繰り返すマルチキャストツリー生成手段と、

を有し、

前記マルチキャストツリー生成手段は、

第1回目の処理として、マルチキャスト始点ノードから始まり、そのノードから発するリンクを求め、そのすべてのリンクをブランチ候補として前記ブランチ候補記憶手段に格納し、当該ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出し、そのブランチ候補の終点がマルチキャスト終点の場合はマルチキャストを最終的に構成するブランチとして前記ブランチ記憶手段に格納する手段と、

第2回目以降の処理として、直前のルーチンで前記ブランチ候補記憶手段から取り出した前記最短経路ブランチ候補の終点がマルチキャスト終点の場合は、当該最短経路ブランチ候補の始点を除いたすべてのノードから発するリンクを求め、そのすべてのリンクをブランチ候補として該ブランチ候補記憶手段に格納し、当該ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出し、そのブランチ候補の終点がマルチキャスト終点の場合はマルチキャストを最終的に構成するブランチとしてブランチ記憶手段に格納する手段と、

2回目以降の処理として、直前のルーチンで取り出した前記最短経路ブランチ候補の終点がマルチキャスト終点でない場合は、当該最短経路ブランチ候補の終点から発するすべてのリンクを求め、当該最短経路ブランチ候補の後に個々のリンクを加えた結果できるすべてのブランチを前記ブランチ候補記憶手段に格納し、当該ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出し、そのブランチ候補の終点がマルチキャスト終点の場合はマルチキャストを最終的に構成するブランチとしてブランチ記憶手段に格納する手段と、

を有することを特徴とするマルチキャスト経路計算装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、マルチキャスト経路計算方法及び装置に係り、特に、マルチキャストツリーを生成するための経路計算（ルーティング）装置におけるマルチキャスト経路計算方法及び装置に関する。

【背景技術】

【0002】

IP(Internet Protocol)ネットワークでは、IPTVのような放送型のサービス要求が高まっており、このIPストリームを(G)MPLS(Multi Protocol Label Switching)で明示的に張った土管に通すことにより、帯域を確保して放送型サービスのQoSを保証する必要がある。これらの要求を満たすためには、効率よくマルチキャストツリーを作成する必要がある。

【0003】

10

20

30

40

50

複数のノードとその複数ノード間を繋ぐ複数の方向性リンク（リンクの方向によりリンクコストが異なるリンク）で構成されるネットワーク上で、アルゴリズムの最初はマルチキャストの始点ノードのみからなるマルチキャストツリーを定義し、当該マルチキャストツリーから、まだ経路が決まっていないマルチキャスト終点ノード集合に辿り着く経路の中で最も経路コストが小さい経路をマルチキャストツリーに加え、この処理を最終的に全てのマルチキャストツリー終点ノードまでの経路が決まるまで繰り返すマルチキャストツリー生成アルゴリズムがある。

【 0 0 0 4 】

そのマルチキャストツリー生成アルゴリズムとして、MPH(Minimum-cost Path Heuristics)アルゴリズム（例えば、非特許文献 1 参照）がある。

10

【 0 0 0 5 】

以下にMPHアルゴリズムについて説明する。

【 0 0 0 6 】

図 1 にMPHアルゴリズムフローを示す。

【 0 0 0 7 】

ステップ 1 ) 当該ステップでは、アルゴリズムの初期条件と、アルゴリズムが使う集合の定義を示す。集合Eはマルチキャストの終点ノードのうち、まだその終点までの経路が決定していない終点ノードの集合である。集合Eはアルゴリズムの最初は全てのマルチキャストの終点ノードを保持するが、終点ノードまでの経路が決定した終点ノードは集合Eから削除される。集合Vはアルゴリズムの過程で作成される既存ツリー上のすべてのノード(集合Rに登録されているすべてのブランチ上のノード)が格納されている。但し、アルゴリズムの最初はマルチキャストツリーの始点ノードのみがVに格納される。集合Rには、ステップ 2 で選択されたV-E間最短経路が新ブランチとして格納されていき、最終的にこのブランチの集合がマルチキャストツリーの全ての構成ブランチとなる。

20

【 0 0 0 8 】

ステップ 2 ) 当該ステップでは既存ツリー上のノード集合であるVから、まだノードまでのブランチが決まっていない終点ノード集合Eまでの最短経路を求めて、その最短経路を新ブランチとして集合Rに登録する。また、当該新ブランチの終点ノードは集合Eから取り除かれ、当該新ブランチの始点以外のノードは集合Vに追加される。

【 0 0 0 9 】

ステップ 3 ) 集合Eが空集合かどうかの判断を行い、もしEが空集合であればすべての終点までのブランチが決定したことになるので、アルゴリズムの終了となる。もし集合Eが空集合ではない場合はステップ 2 に戻る。

30

【 0 0 1 0 】

ステップ 2 のV-E間の最短経路算出のためのダイクストラのアルゴリズム（非特許文献 2 ）は、現在最も計算量を小さくできるFibonacciヒープ（非特許文献 3、4 ）を使ったデータ構造で、 $O(l+n \log n)$ の計算量である。なお、ここでlはネットワーク上のリンク数を示し、nはノード数を示す。図 1 のステップ 2 ~ 3 のループはマルチキャスト終点数であるm回走行されるので、MPHの計算量は最悪時間計算量、平均時間計算量ともに $O(m(l+n \log n))$ となる。

40

【 先行技術文献 】

【 非特許文献 】

【 0 0 1 1 】

【 非特許文献 1 】 Hiromitsu Takahashi and Akira Matsuyama, "AN APPROXIMATE SOLUTION FOR THE STEINER PROBLEM IN GRAPHS," Math. Japonica, vol. 24, pp. 573-577, 1980

【 非特許文献 2 】 <http://www.deqnotes.net/acmicpc/dijkstra/>

【 非特許文献 3 】 M.L. Fredman and R.E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," Journal of the ACM (JACM), vol. 34, no. 3, pp. 596-615, 1987.

50

【非特許文献4】<http://www.lab2.kuis.kyoto-u.ac.jp/~itohiro/lecture/heap-GT.pdf>

【発明の概要】

【発明が解決しようとする課題】

【0012】

しかしながら、上記のアルゴリズムMPHは、マルチキャスト終点数回のダイクストラアルゴリズム走行が必要となるため、計算量が大きく処理時間が大きくなる問題がある。

【0013】

本発明は、上記の点に鑑みなされたもので、従来のMPHアルゴリズムで必要であったノード間の最短経路を求めるダイクストラアルゴリズムの走行を行わず、短時間でMPHアルゴリズムと同じマルチキャストツリーを作成することが可能なマルチキャスト経路計算方法及び装置を提供することを目的とする。

【課題を解決するための手段】

【0014】

上記の課題を解決するため、本発明は、複数ノードを通信リンクを介して相互に接続可能としたネットワーク上で、ある始点ノードから他のすべてのノード内の部分集合である複数ノードまでのマルチキャスト経路を計算するためのマルチキャスト経路計算方法であって、

最初はマルチキャスト始点ノードのみからなるマルチキャストツリーを定義し、当該マルチキャストツリーから、まだ経路が決まっていないマルチキャスト終点ノード集合の1終点ノードにたどり着く最短経路を新ブランチとしてマルチキャストツリーに加え、この処理を最終的にすべてのマルチキャストツリーブランチが決まるまで繰り返すマルチキャストツリー生成アルゴリズムを用い、

アルゴリズムの第1回目のルーチンはマルチキャスト始点ノードから始まり、そのノードから発するリンクを求め、そのすべてのリンクをブランチ候補としてブランチ候補記憶手段に格納し、当該ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出し、そのブランチ候補の終点がマルチキャスト終点の場合はマルチキャストを最終的に構成するブランチとしてブランチ記憶手段に格納し、

アルゴリズムの第2回目のルーチン以降は、直前のルーチンで前記ブランチ候補記憶手段から取り出した前記最短経路ブランチ候補の終点がマルチキャスト終点の場合は、当該最短経路ブランチ候補の始点を除いたすべてのノードから発するリンクを求め、そのすべてのリンクをブランチ候補として該ブランチ候補記憶手段に格納し、当該ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出し、そのブランチ候補の終点がマルチキャスト終点の場合はマルチキャストを最終的に構成するブランチとしてブランチ記憶手段に格納し、

また、アルゴリズムの第2回目のルーチン以降で、直前のルーチンで取り出した前記最短経路ブランチ候補の終点がマルチキャスト終点でない場合は、当該最短経路ブランチ候補の終点から発するすべてのリンクを求め、当該最短経路ブランチ候補の後に個々のリンクを加えた結果できるすべてのブランチを前記ブランチ候補記憶手段に格納し、当該ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出し、そのブランチ候補の終点がマルチキャスト終点の場合はマルチキャストを最終的に構成するブランチとしてブランチ記憶手段に格納する。

【0015】

また、本発明は、前記ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出す手順において、

前記最短経路ブランチ候補の終点がマルチキャスト終点の場合は、当該ブランチ候補上の始点を除いたすべてのノードにマルチキャストツリーの構成要素になったという印("reached属性"=yes)を付与する。

【0016】

また、本発明は、前記ブランチ候補を前記ブランチ候補記憶手段に格納する手順において、

10

20

30

40

50

新規ブランチ候補の最終リンクとして選択されたリンクの終点ノードに前記印("reached属性"=yes)が付いている場合は、当該ブランチ候補を前記ブランチ候補記憶手段に格納しない。

【0017】

また、本発明は、ネットワーク上の各ノードに、"当該ノードまでの最短ブランチ候補とそのコスト"を属性として保持させ、両者のアルゴリズム開始時の初期値はブランチ候補 = 空集合、経路コスト = とするが、アルゴリズム走行後は当該属性にアルゴリズムの開始からその時点までにアルゴリズムが扱った当該ノードを終点に持つブランチ候補の中で最もコストが小さいブランチ候補とその経路コストを保持する。

【0018】

また、本発明は、前記ブランチ候補記憶手段から最も経路コストが小さい最短経路ブランチ候補を取り出す手順において、

当該ブランチ候補の終点の"当該ノードまでのブランチ候補とそのコスト"属性からブランチ候補を取出し、ブランチ候補 = 空集合にする。

【0019】

また、本発明は、新規ブランチ候補の最終リンクとして選択されたリンクの終点ノードの上記"当該ノードまでのブランチ候補とそのコスト"属性にすでにブランチ候補が格納されている場合は、当該属性に格納されている経路コストと新規ブランチ候補の経路コストを比較して新規ブランチ候補の当該ノードまでの経路コストが既存ブランチ候補の経路コストより小さい場合にのみ、前記ブランチ候補記憶手段から当該既存ブランチ候補を取出し、新規ブランチ候補を前記ブランチ候補記憶手段に格納し、当該終点ノードの当該属性の既存ブランチ候補とその経路コストを新規ブランチ候補とその経路コストに置き換える。

【0020】

また、本発明は、新規ブランチ候補の最終リンクとして選択されたリンクの終点ノードの上記"当該ノードまでのブランチ候補とそのコスト"属性にブランチ候補が格納されていない場合は、当該属性に格納されている経路コストと新規ブランチ候補の経路コストを比較して、当該ノードまでの新規ブランチ候補の経路コストが当該属性内の経路コストより小さい場合は、新規ブランチ候補を前記ブランチ候補記憶手段に格納し、当該終点ノードの当該属性に新規ブランチ候補とその経路コストを格納する。

【発明の効果】

【0021】

本発明によれば、従来のMPHアルゴリズムで必要であったノード間の最短経路を求めるダイクストラアルゴリズムの走行を行う必要がなく、ネットワーク上のリンクをたどることにより、最終的にMPHアルゴリズムと同じマルチキャストツリーを作成することを可能としている。

【0022】

そのため、本発明のアルゴリズム平均時間計算量は $(\log m)(1 + n \log n)$ になる。この時間計算量はMPHの平均時間計算量 $m(1 + n \log n)$ よりも明確に小さい。

【0023】

実際にランダムに配置した5,070のノード間に方向性リンク57,174本を設定し、リンクのコストは1-100間でランダムに平均41.95のコストで設定し、最もノード次数が高いノード(ノード次数:28)をマルチキャストの始点とし、ノード次数が1以上で次数が低いものから順に200-2,000の終点ノードを200間隔で設定し、その設定された終点ノードに対してMPHと本発明のアルゴリズムを使ってマルチキャストツリーを計算機(Linux(登録商標)マシン)により生成した場合の処理速度を図2に示す。図2より、MPHは終点数が多いとダイクストラアルゴリズムの走行回数が多くなり、Fibonacciヒープへのアクセス回数が大きくなるため、処理速度が落ちる。しかし、本発明のアルゴリズムはFibonacciヒープへのアクセスが終点数 $m$ にかかわらずMPHに比べて大幅に少ないため、 $m$ が大きくなればなるほど処理時間がMPHに対して速くなる。

## 【図面の簡単な説明】

【0024】

【図1】MPH(minimum-cost path heuristic)フローである。

【図2】本発明のアルゴリズムの効果を示す図である。

【図3】本発明の一実施の形態におけるマルチキャスト経路計算装置の構成図である。

【図4】本発明の一実施の形態におけるマルチキャスト経路計算装置の動作のフローチャートである。

【図5】本発明の一実施の形態におけるアルゴリズム適用動作例(その1)である。

【図6】本発明の一実施の形態におけるアルゴリズム適用動作例(その2)である。

【図7】本発明の一実施の形態におけるアルゴリズム適用動作例(その3)である。

【図8】本発明の一実施の形態におけるアルゴリズム適用動作例(その4)である。

【図9】本発明の一実施の形態におけるアルゴリズム適用動作例(その5)である。

【図10】本発明の一実施の形態におけるアルゴリズム適用動作例(その6)である。

## 【発明を実施するための形態】

【0025】

以下、図面と共に本発明の実施の形態を説明する。

【0026】

図3は、本発明の一実施の形態におけるマルチキャスト計算装置の構成を示す。

【0027】

同図に示す計算装置は、ネットワークポロジ管理部100、ブランチ候補管理部(PQ)200、最小木作成部300から構成される。

【0028】

ネットワークポロジ管理部100は、ネットワークポロジ管理処理部11、リンク情報記憶部21、ノード情報記憶部22から構成され、ブランチ候補管理部200は、ブランチ候補処理部12、ブランチ候補記憶部23から構成され、最小木作成部300は、最小木作成処理部13とブランチ記憶部24から構成される。

【0029】

ネットワークポロジ管理部100のネットワークポロジ管理処理部11は、最小木作成処理部13との通信と、当該ネットワークポロジ管理部100内のリンク情報記憶部21及びノード情報記憶部22へのアクセスを行う。

【0030】

ネットワークポロジ管理部100のリンク情報記憶部21はネットワーク上の各方向性リンク毎にその方向毎のリンクコストを保持する。ノード情報記憶部22はネットワーク上の各ノード毎に情報を保持する。ノード情報記憶部22は、接続リンク情報を保持する。接続リンク情報は当該ノードに接続しているリンク情報である。またノード情報記憶部22は、後述するアルゴリズム中で利用する当該ノードまでの"最短ブランチ候補"と、その"最短ブランチ候補コスト"を格納する"当該ノードまでのブランチ候補とそのコスト"属性、また当該ノードがマルチキャストツリー構成ノードかどうかを判別する"reached属性"を保持する。

【0031】

最小木作成部300は、最小木作成処理部13で、図4に後述するアルゴリズムを実行する。最小木作成処理部13は、図4のステップ102ではネットワークポロジ管理処理部100を通して、ノード情報記憶部22から集合Vに属する各ノードの接続リンクを取得し、当該リンクの終点ノードの"reached属性"= no、且つ"最短ブランチ候補コスト"よりも経路コストが小さいブランチをブランチ候補としてブランチ候補記憶部23に格納する。ブランチ記憶部24は図4の集合Rに相当し、ステップ103で選択されたブランチ候補記憶部(PQ)23内の最小ブランチで終点が集合Eに属するブランチが格納される。

【0032】

ブランチ候補管理部(PQ)200は、ブランチ候補処理部12が最小木作成処理部13か

10

20

30

40

50

らの要望に応じて、ブランチ候補記憶部(PQ)23へのブランチ候補の格納、削除、及び最短ブランチ候補の取得を行う。ブランチ候補記憶部(PQ)23にはFibonacciヒープ等の構造が用いられ、ブランチ候補がその経路コストに応じてソートされ、ブランチ候補へのアクセス速度を速めることができる。このブランチ候補記憶部(PQ)23への新ブランチ候補登録は図4のステップ102であり、入れ替えられた古いブランチ候補はブランチ候補記憶部PQ23から削除される。またステップ103で取得されるブランチ候補記憶部PQの最短ブランチ候補はブランチ候補記憶部(PQ)23内で最小の経路コストを持つブランチ候補である。

【0033】

以下に、上記の構成における動作を説明する。

10

【0034】

図4は、本発明の一実施の形態におけるマルチキャスト経路計算方法のフローチャートである。

【0035】

ステップ101) アルゴリズムの初期条件として、メモリ(図示せず)上の集合Eはマルチキャストのすべての終点ノードを保持する。メモリ(図示せず)上の集合Vはステップ102で選択されるリンクの始点ノード集合が格納され、アルゴリズムの最初はマルチキャスト始点ノードだけが格納される。

【0036】

ブランチ候補記憶部23の集合PQはブランチ候補を格納する集合であり、空集合( )で設定される。

20

【0037】

ブランチ記憶部24の集合Rは最終的なツリーを構成するブランチの集合であり、最初は空集合( )で設定される。

【0038】

マルチキャストツリーを作成するネットワークの各ノードは本発明アルゴリズムのための属性を保持するが、"reached属性"は当該ノードがマルチキャストツリーの構成要素になったかを示す属性であり、yes/noの値をとる(請求項2に対応)。アルゴリズムの最初では始点ノードのみがマルチキャストツリーに含まれるので、yesに設定されるが、他ノードはすべてnoに設定される。

30

【0039】

ノード情報記憶部22の"当該ノードまでの最短ブランチ候補とそのコスト"はアルゴリズム開始時の初期値は"最短ブランチ候補" = 空集合、"最短ブランチ候補コスト" = と設定される。

【0040】

ステップ102) ブランチ候補処理部12は、集合V内の各ノードから発するリンクを選択し、当該リンクからブランチ候補を作成する。その際、直前のステップ104でブランチ終点が集合Eに属するマルチキャスト終点の場合(ステップ104 = yes)は当該リンクがそのままブランチ候補になるが、マルチキャスト終点でない場合(ステップ104 = no)は当該リンクを直前のステップ103で選択された最短ブランチ候補の後に繋げた経路がブランチ候補になる(請求項1に対応)。アルゴリズムの第1回目のルーチンでは、リンクがそのまま新規ブランチ候補になる。この新規ブランチ候補作成処理はV内のノードから発するすべてのリンクについて行われるが、ブランチ候補は"reached属性"=noのノード毎に最大1つ存在する。つまり、新規ブランチ候補の終点ノードが保持する"当該ノードまでの最短ブランチ候補"が空集合ではなく、既存の候補リストである"最短ブランチ候補コスト"より新規ブランチの経路コストが小さい場合のみ、候補の入れ替えがおきる。また、候補の入れ替えが生じた場合は、新規ブランチ候補の終点ノードの"当該ノードまでの最短ブランチ候補"に新規ブランチ候補が格納され、"最短ブランチ候補コスト"に新規ブランチ候補の経路コストが格納される。リンクの終点ノードの"reached属性"がyesの場合(請求項2に対応)は、そのノードはすでにマルチキャストツリーに含まれると

40

50

ということなので、ブランチ候補とならない（請求項3に対応）。"reached属性"がnoの場合で終点ノード属性の"最短ブランチ候補コスト"に格納されている経路コストが新規ブランチ候補の経路コストより大きい場合は新規ブランチ候補がブランチ候補記憶部（PQ）23に登録される。もし、"最短ブランチ候補"が空集合の場合、ブランチ候補記憶部（PQ）23から削除されるブランチ候補はないが（請求項7に対応）、"最短ブランチ候補"が存在する場合は、そのブランチ候補がブランチ候補記憶部（PQ）23から削除される（請求項6に対応）。また、"最短ブランチ候補"と"最短ブランチ候補コスト"の値は新規ブランチ候補とその経路コストの値に置き換えられる（請求項6,7に対応）。"最短ブランチ候補コスト"に格納されている経路コストが新規ブランチ候補の経路コスト以下の場合は新規ブランチ候補はブランチ候補記憶部（PQ）23に格納されず、ブランチ候補記憶部（PQ）23から削除されるブランチ候補も存在しない。また当該ノードの"最短ブランチ候補"と"最短ブランチ候補コスト"の値に変化はない。

10

## 【0041】

ステップ103) ブランチ候補記憶部（PQ）23内の最短ブランチ候補を取り出す。この際に、当該最短ブランチ候補の終点ノードの属性中の"当該ノードまでの最短ブランチ候補"を空集合に設定する。また、集合V内ノードを空集合にした後、当該最短ブランチ候補の終点ノードをVに格納する。

## 【0042】

ステップ104) ステップ104で選択した最短ブランチ候補の終点ノードが集合E内のマルチキャスト終点ノードとして存在するかどうかを判断し、Yesの場合はステップ105へ進み、Noの場合はステップ102へ進む。

20

## 【0043】

ステップ105) ステップ103で選択された最短ブランチ候補を新ブランチとしてブランチ記憶部24の集合Rに格納するとともに、当該最短ブランチ候補の終点ノードを集合Eから削除する。

## 【0044】

ステップ106) 集合Eが空集合かどうかを判断し、空集合であればすべてのマルチキャスト終点までのブランチが得られたことと同値であるため、アルゴリズムの終了となる。空集合でない場合は、ステップ107へ進む。

## 【0045】

ステップ107) 現在新ブランチの終点ノードが格納されている集合V内のノード集合を新ブランチの始点を除くすべてのノード集合に置き換える（請求項1に対応）。また、集合V内のすべてのノードの"reached属性"をyesに設定する。

30

## 【0046】

以下、図5～図10に上記のアルゴリズム適用動作例を、小規模ネットワークを用いて示す。これらの図において、左上のPQは、ブランチ候補記憶部23にブランチ候補が格納される様子を示す。また、これらの図において、ノードAはマルチキャスト始点ノード、ノードB-Dはマルチキャスト終点ノード、ノードE、Fはその他のノードを示す。

## 【0047】

図5において、各ノードでは、アルゴリズムの最初は"reached属性"=no, "最短ブランチ候補"=、"最短ブランチ候補コスト"= に設定されている。

40

## 【0048】

図6において、アルゴリズムの第1回目のルーチンのステップ102ではノードAからのリンクが最短ブランチ候補としてブランチ候補記憶部（PQ）23に登録される。また各ブランチ候補の終点ノードの"最短ブランチ候補"、"最短ブランチ候補コスト"も、当該ブランチ候補の値に更新される。第1回目のルーチンのステップ103では、ブランチ候補記憶部（PQ）23中の最短ブランチ候補であるノードA-E（コスト=1）が選択され、ブランチ候補記憶部（PQ）23から取り出される。ノードEは終点ノードではないので、ノードEの"reached属性"はnoのままである。

## 【0049】

50

図7において、アルゴリズムの第2回目のルーチンの処理を示すが、ステップ102ではノードEから出ているリンクであるL5, L7をL1に加えて新規ブランチ候補としてノードB, Dまでの既存のブランチ候補とのコスト比較を行う。その際、ブランチ候補A-E-B(コスト=3)は既存のA-B(コスト=4)よりもコストが小さいため、ブランチ候補記憶部(PQ)23内のブランチ候補の入れ替えが起きると共に、ノードBの"最短ブランチ候補"、"最短ブランチ候補コスト"を新規ブランチ候補の値に入れ替える。A-E-D(コスト=4)はA-D(コスト=3)よりもコストが大きいためブランチ候補の入れ替えはおきない。ステップ103ではブランチ候補記憶部(PQ)23内の最短ブランチ候補であるA-F(コスト=2)が選択され、ブランチ候補記憶部(PQ)23から取り出される。

【0050】

10

図8において、アルゴリズムの第4回目のルーチンの処理を示すが、ステップ102ではリンクL3にノードFから出ているリンクであるL6を加え、ブランチ候補記憶部(PQ)23にブランチ候補A-F-Cを加える。ステップ103ではブランチ候補記憶部(PQ)23内の最短ブランチ候補をブランチ候補記憶部(PQ)23から取り出すが、ノードBがマルチキャスト終点のため、ステップ105でノードA-E-Bを集合Rに加える。第4回目のルーチンのステップ107では、集合V内のノード集合にノードA-E-Bの始点を除いたノードであるE, Bが入れられるとともに、E, Bの"reached属性"がyesに設定される。reached属性をyesに設定することにより、E, Bを終点とする新規ブランチ候補は作られなくなる。

【0051】

20

図9では、アルゴリズムの第5から第6回目のルーチンのステップ102までの処理を示すが、第5回目のルーチンのステップ102ではノードEとBから出ているリンクを新規ブランチ候補としてブランチ候補記憶部(PQ)23への登録を試みる。ノードBから出ているリンクはないので登録されない。ノードE-BはすでにノードEの"reached属性"がyesになっているため登録されない。ノードE-Dはコストが3であるが、ノードDが持つ最短ブランチ候補A-Dのコストも3であるため、登録されない。ステップ103ではブランチ候補記憶部(PQ)23内の最小のブランチ候補A-Dが選択され取り出され、ノードDはマルチキャスト終点のためA-Dがブランチ記憶部24の集合Rに格納される。第5回目のルーチンのステップ107で終点Dが集合Vに格納され、第6回目のルーチンのステップ102でノードDから出ているリンクL8(D-C)が新規ブランチ候補となる。このブランチ候補D-Cのコスト(コスト=2)はCまでの既存のブランチ候補A-F-Cのコスト6よりも小さいため、ブランチ候補記憶部(PQ)23内のブランチ候補の入れ替えが起き、ブランチ候補D-Cがブランチ候補記憶部(PQ)23に入る。

【0052】

30

図10は、第6回目のルーチンのステップ105が終わったときの状態を示す。ブランチ候補記憶部(PQ)23内にはノードD-Cのみが第6回目のルーチンのステップ103時には存在するので、ブランチ候補記憶部(PQ)23からブランチ候補D-Cが取り出されるが、ノードCが終点のため、ブランチ候補D-Cがブランチ記憶部24の集合Rに格納される。第6回目のルーチンのステップ106でノードEが空集合になるのでアルゴリズムの終了となる。

【0053】

40

上記のように、本発明の図4に示すアルゴリズムによる処理により、ダイクストラアルゴリズムの走行を終点数だけ繰り返すMPHに比べてPQのFibonacciヒープに対するアクセス数が大幅に減少するため、計算量が少なくなり処理速度の大幅な向上が得られる。

【0054】

上記の実施の形態における図4のアルゴリズムをプログラムとして構築し、コンピュータにインストールして実行させる、または、ネットワークを介して流通させることが可能である。

【0055】

本発明は、上記の実施の形態に限定されることなく、特許請求の範囲内において種々変

50

更・応用が可能である。

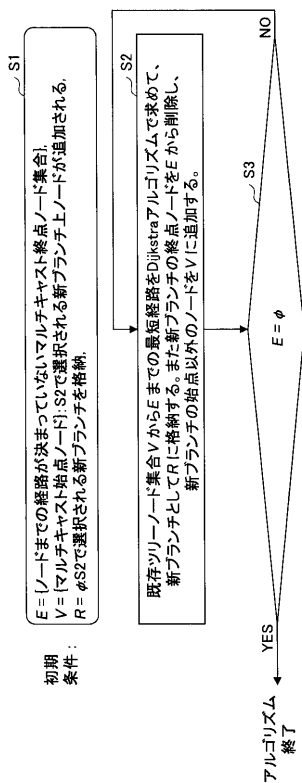
【符号の説明】

【0056】

- 1 1 ネットワークトポロジ管理処理部
- 1 2 ブランチ候補処理部
- 1 3 最小気作成処理部
- 2 1 リンク情報記憶部
- 2 2 ノード情報記憶部
- 2 3 ブランチ候補記憶部
- 2 4 ブランチ記憶部
- 1 0 0 ネットワークトポロジ管理部
- 2 0 0 ブランチ候補管理部
- 3 0 0 最小木作成部

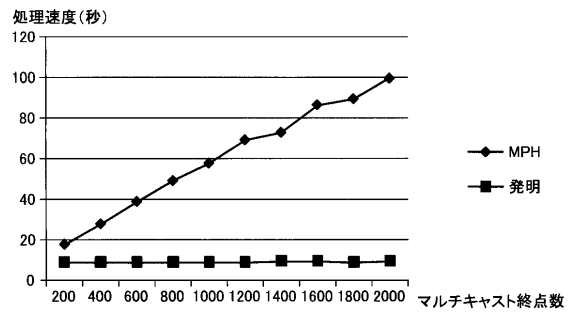
【図1】

MPH (minimum-cost path heuristic) のフローチャート



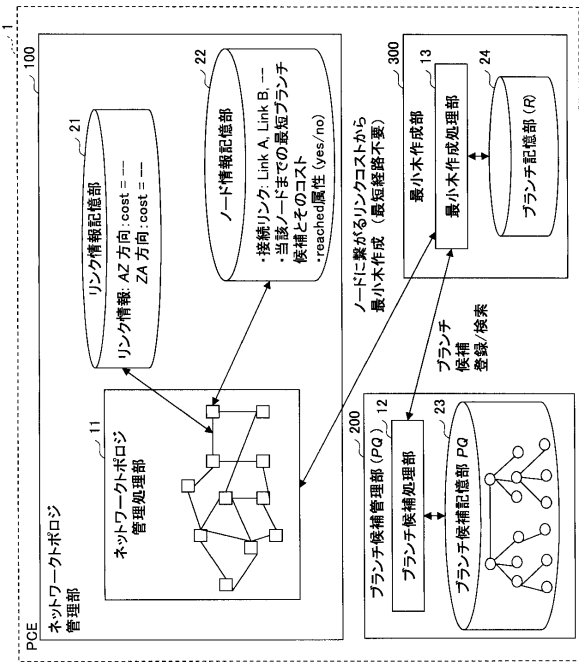
【図2】

本発明のアルゴリズムの効果を示す図



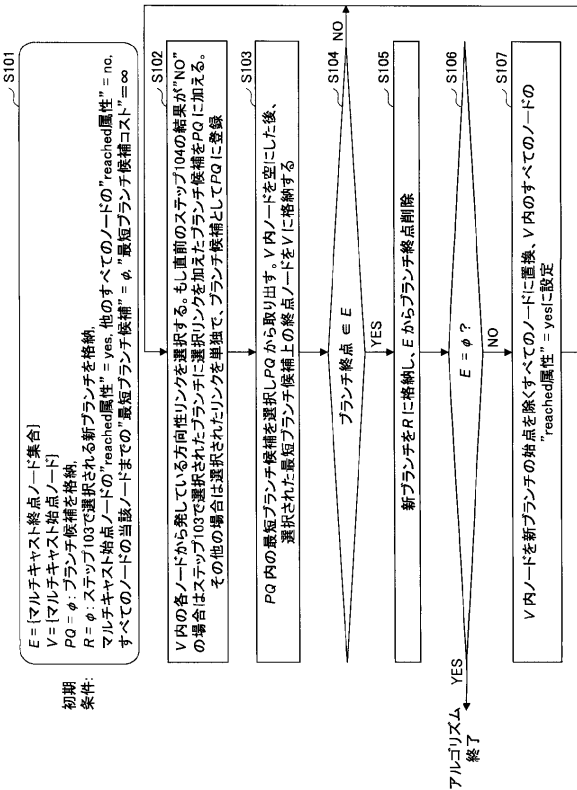
【図3】

本発明の一実施の形態におけるマルチキャスト経路計算装置の構成図



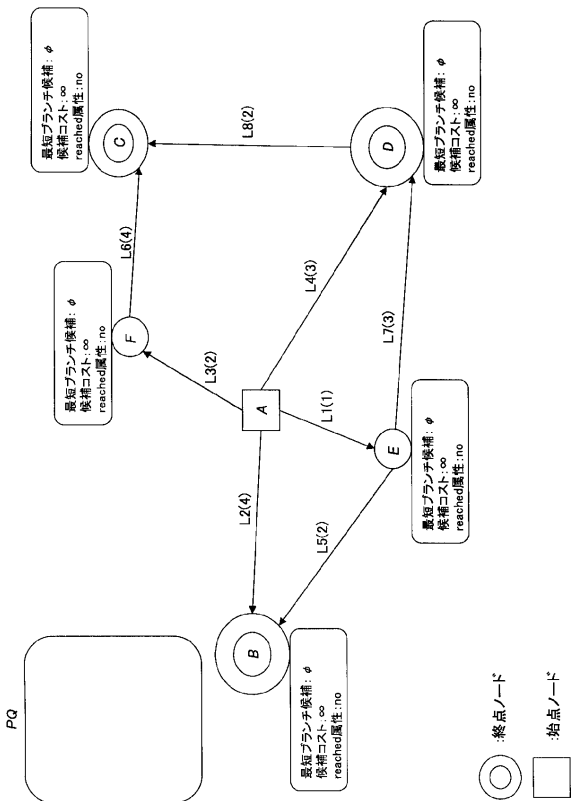
【図4】

本発明の一実施の形態におけるマルチキャスト経路計算装置の動作のフローチャート



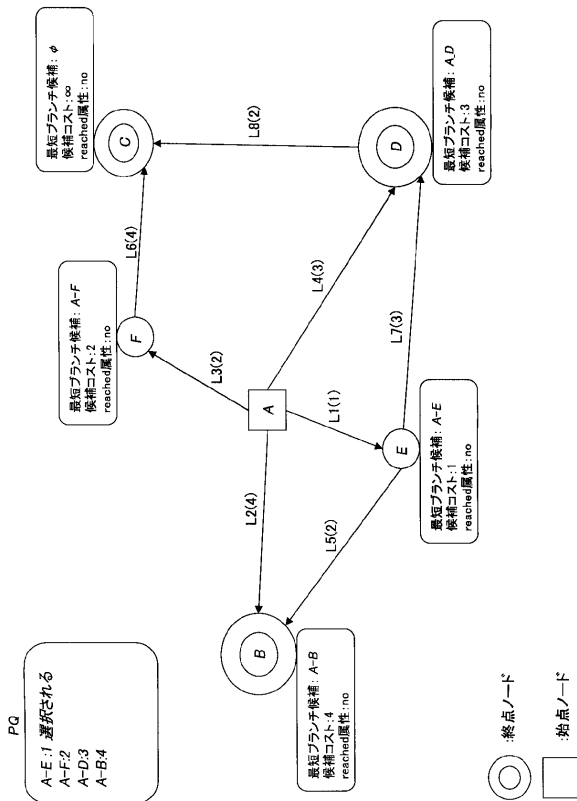
【図5】

本発明の一実施の形態におけるアルゴリズム適用動作例(その1)



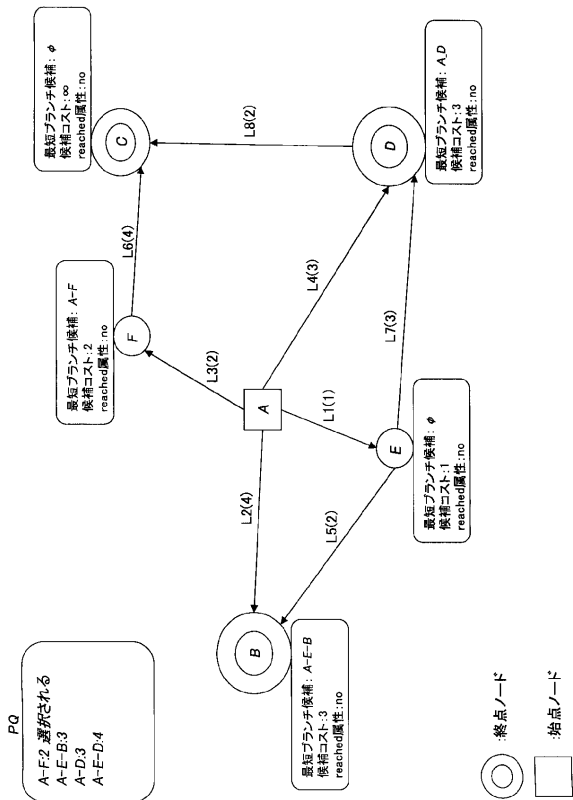
【図6】

本発明の一実施の形態におけるアルゴリズム適用動作例(その2)



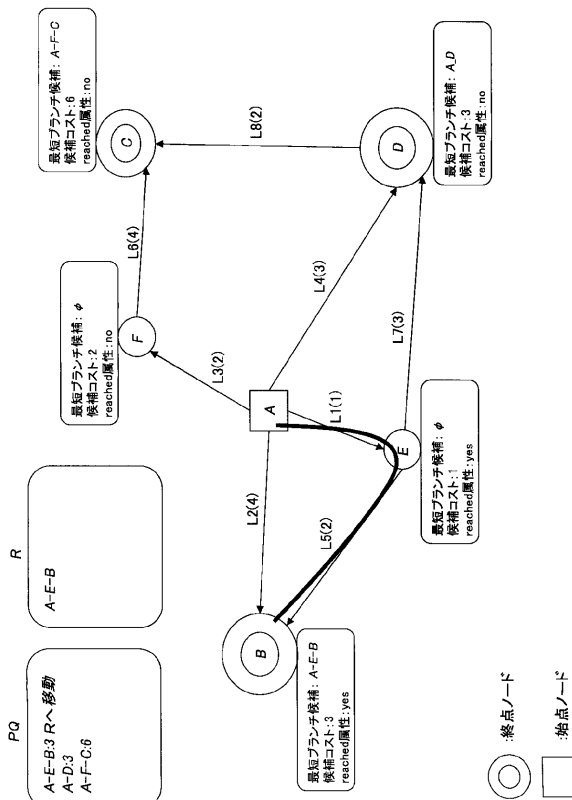
【図7】

本発明の一実施の形態におけるアルゴリズム適用動作例(その3)



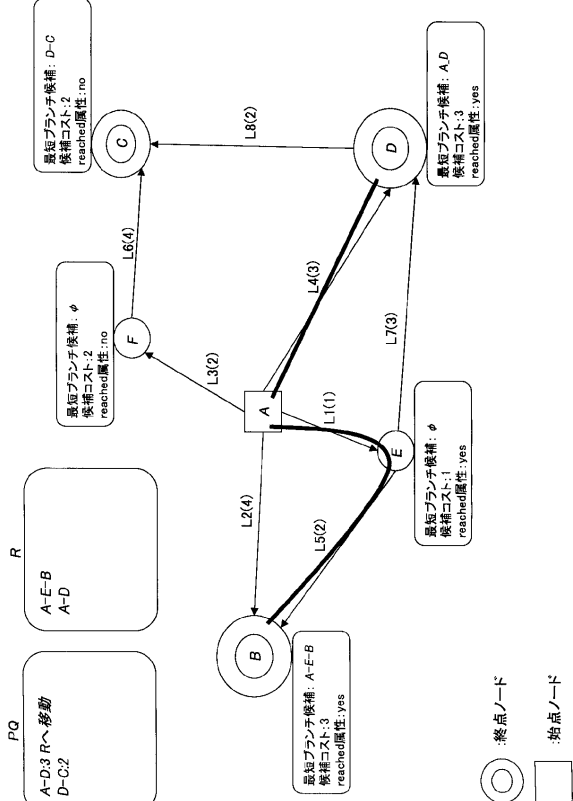
【図8】

本発明の一実施の形態におけるアルゴリズム適用動作例(その4)



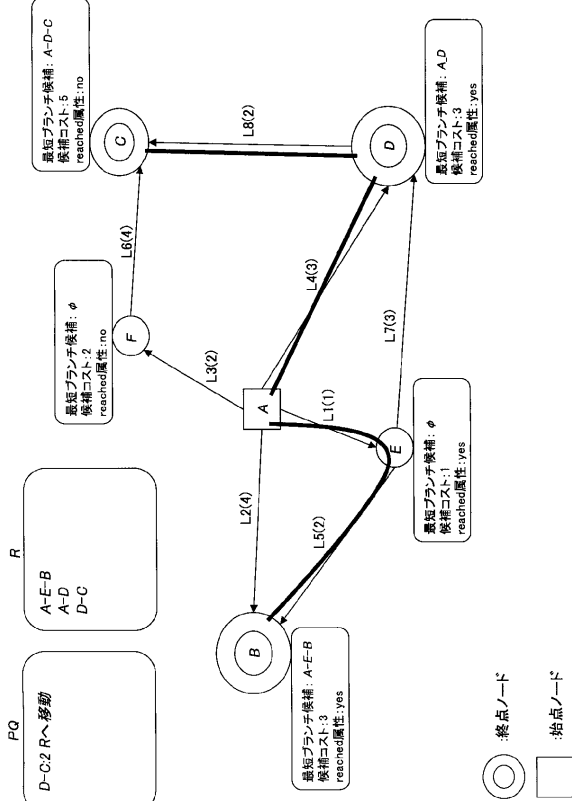
【図9】

本発明の一実施の形態におけるアルゴリズム適用動作例(その5)



【図10】

本発明の一実施の形態におけるアルゴリズム適用動作例(その6)



---

フロントページの続き

- (56)参考文献 特開2002-261804(JP,A)  
特開2008-005066(JP,A)  
松浦 洋、森田 直孝、高見 一正、P2MP TEに適用するSteiner Treeアル  
ゴリズムの検討、電子情報通信学会技術研究報告 Vol.109 No.275、社団法人電  
子情報通信学会、2009年11月 5日、pp.35-40

- (58)調査した分野(Int.Cl., DB名)

H04L 12/761