

(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2010-4387

(P2010-4387A)

(43) 公開日 平成22年1月7日(2010.1.7)

(51) Int.Cl.
H04L 12/56 (2006.01)F I
H04L 12/56 I O O Zテーマコード (参考)
5 K O 3 O

審査請求 未請求 請求項の数 6 O L (全 13 頁)

(21) 出願番号 特願2008-162249 (P2008-162249)
(22) 出願日 平成20年6月20日 (2008.6.20)(71) 出願人 000208891
K D D I 株式会社
東京都新宿区西新宿二丁目3番2号
(74) 代理人 100106909
弁理士 棚井 澄雄
(74) 代理人 100064908
弁理士 志賀 正武
(74) 代理人 100146835
弁理士 佐伯 義文
(74) 代理人 100138759
弁理士 大房 直樹
(72) 発明者 荻野 長生
埼玉県ふじみ野市大原2丁目1番15号
株式会社K D D I 研究所内

最終頁に続く

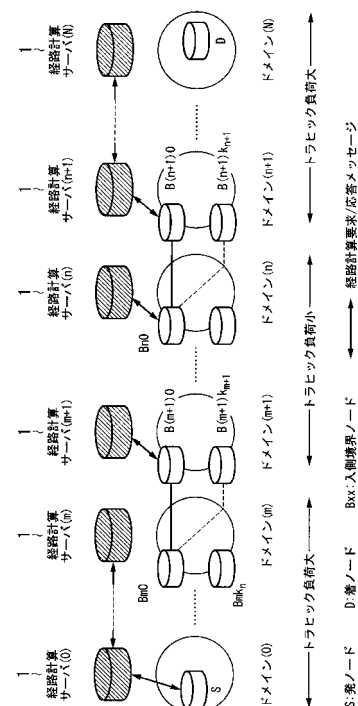
(54) 【発明の名称】 経路計算サーバ、経路計算方法及びプログラム

(57) 【要約】

【課題】各ドメインのトラヒック負荷の状況に応じたドメイン間パス設定方式を用いることにより、ドメイン間パス設定に要する時間を短縮する。

【解決手段】各ドメインが境界ノードによって互いに接続されるネットワークシステムにおいて、各ドメインに対応して設置され、発ノードから着ノードまでの経路を計算する経路計算サーバであって、ドメイン内のトラヒック負荷の大きさを判定する判定手段と、前記経路を計算する経路計算手段と、経路計算要求を行ったノードに、前記経路計算手段により計算された経路計算結果を通知する通知手段と、を備え、前記経路計算手段は、ドメインのトラヒック負荷が大きい場合にはエンドツーエンド経路計算方式で経路を計算し、ドメインのトラヒック負荷が小さい場合にはドメイン毎経路計算方式で経路を計算する。

【選択図】図1



【特許請求の範囲】**【請求項 1】**

各ドメインが境界ノードによって互いに接続されるネットワークシステムにおいて、各ドメインに対応して設置され、発ノードから着ノードまでの経路を計算する経路計算サーバであって、

ドメイン内のトラヒック負荷の大きさを判定する判定手段と、

前記経路を計算する経路計算手段と、

経路計算要求を行ったノードに、前記経路計算手段により計算された経路計算結果を通知する通知手段と、

を備え、

10

前記経路計算手段は、

着ノードのドメインでは、

自ドメインのノードから経路計算要求を受信するとドメイン毎経路計算方式で着ノードまでの経路を計算し、

前ドメインの経路計算サーバから経路計算結果を受信するとエンドツーエンド経路計算方式で着ノードまでの経路を計算し、

着ノードのドメインでないドメインでは、

自ドメインのノードから経路計算要求を受信すると、自ドメイン及び次ドメインのトラヒック負荷がともに大きい場合には、エンドツーエンド経路計算方式でトラヒック負荷が小さいドメインまでの経路を計算し、そうでない場合には、ドメイン毎経路計算方式で次ドメインまでの経路を計算し、

20

前ドメインの経路計算サーバから経路計算結果を受信すると、次ドメインのトラヒック負荷が大きい場合には、エンドツーエンド経路計算方式でトラヒック負荷が小さいドメインまでの経路を計算し、そうでない場合には、エンドツーエンド経路計算方式で次ドメインまでの経路を計算する

ことを特徴とする経路計算サーバ。

【請求項 2】

前記経路計算手段は、エンドツーエンド経路計算方式で経路を計算する場合には、次ドメインの最も多くの境界ノードに至ることができる境界ノードを経由する経路を選択することを特徴とする請求項 1 に記載の経路計算サーバ。

30

【請求項 3】

各ドメインが境界ノードによって互いに接続されるネットワークシステムにおいて、各ドメインに対応して設置される経路計算サーバにおける発ノードから着ノードまでの経路を計算する経路計算方法であって、

ドメイン内のトラヒック負荷の大きさを判定するステップと、

前記経路を計算するステップと、

経路計算要求を行ったノードに、前記経路計算手段により計算された経路計算結果を通知するステップと、

を有し、

40

着ノードのドメインでは、

自ドメインのノードから経路計算要求を受信するとドメイン毎経路計算方式で着ノードまでの経路を計算し、

前ドメインの経路計算サーバから経路計算結果を受信するとエンドツーエンド経路計算方式で着ノードまでの経路を計算し、

着ノードのドメインでないドメインでは、

自ドメインのノードから経路計算要求を受信すると、自ドメイン及び次ドメインのトラヒック負荷がともに大きい場合には、エンドツーエンド経路計算方式でトラヒック負荷が小さいドメインまでの経路を計算し、そうでない場合には、ドメイン毎経路計算方式で次ドメインまでの経路を計算し、

前ドメインの経路計算サーバから経路計算結果を受信すると、次ドメインのトラヒック

50

負荷が大きい場合には、エンドツーエンド経路計算方式でトラヒック負荷が小さいドメインまでの経路を計算し、そうでない場合には、エンドツーエンド経路計算方式で次ドメインまでの経路を計算する

ことを特徴とする経路計算方法。

【請求項 4】

エンドツーエンド経路計算方式で経路を計算する場合には、次ドメインの最も多くの境界ノードに至ることができる境界ノードを経由する経路を選択することを特徴とする請求項 3 に記載の経路計算方法。

【請求項 5】

各ドメインが境界ノードによって互いに接続されるネットワークシステムにおいて、各ドメインに対応して設置され、発ノードから着ノードまでの経路を計算するためのコンピュータに、

ドメイン内のトラヒック負荷の大きさを判定するステップと、

前記経路を計算するステップと、

経路計算要求を行ったノードに、前記経路計算手段により計算された経路計算結果を通知するステップと、

を実行させるものであり、

着ノードのドメインでは、

自ドメインのノードから経路計算要求を受信するとドメイン毎経路計算方式で着ノードまでの経路を計算し、

前ドメインの経路計算サーバから経路計算結果を受信するとエンドツーエンド経路計算方式で着ノードまでの経路を計算し、

着ノードのドメインでないドメインでは、

自ドメインのノードから経路計算要求を受信すると、自ドメイン及び次ドメインのトラヒック負荷がともに大きい場合には、エンドツーエンド経路計算方式でトラヒック負荷が小さいドメインまでの経路を計算し、そうでない場合には、ドメイン毎経路計算方式で次ドメインまでの経路を計算し、

前ドメインの経路計算サーバから経路計算結果を受信すると、次ドメインのトラヒック負荷が大きい場合には、エンドツーエンド経路計算方式でトラヒック負荷が小さいドメインまでの経路を計算し、そうでない場合には、エンドツーエンド経路計算方式で次ドメインまでの経路を計算する

ことを特徴とするプログラム。

【請求項 6】

エンドツーエンド経路計算方式で経路を計算する場合には、次ドメインの最も多くの境界ノードに至ることができる境界ノードを経由する経路を選択することを特徴とする請求項 5 に記載のプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、通信ネットワークにおける経路計算サーバ、経路計算方法及びプログラムに関する。

【背景技術】

【0002】

従来、異なる管理領域に属する複数のドメインが、それぞれが所有する境界ノードによって互いに接続されているようなマルチドメインネットワークにおいて、これら複数のドメインを経由するドメイン間パスを設定する際、以下に示す 2 つの方式が用いられていた。

まず、非特許文献 1 に記載されたエンドツーエンド経路計算方式について図 5 を参照して説明する。

エンドツーエンド経路計算方式では、パスの始点である発ノード S 0 が属する発ドメイ

10

20

30

40

50

ン(ドメイン(0))からパスの終点である着ノードD0が属する着ドメイン(ドメイン(N))までの各ドメインに対応する経路計算サーバが連携して、発ノードS0から着ノードD0までの最適経路を計算する。そして、発ノードS0から着ノードD0までの最適経路が計算された後に、最適経路に沿って発ノードS0から着ノードD0までのドメイン間パスが設定される。

次に、非特許文献2に記載されたドメイン毎経路計算方式について図6を参照して説明する。

ドメイン毎経路計算方式では、まず、発ノードS0が属する発ドメイン(ドメイン(0))の経路計算サーバが、発ノードS0からドメイン(1)の入側境界ノードまでの最適経路を計算する。そして、計算結果として得られた最適経路に沿ってドメイン(0)からドメイン(1)のドメイン間パスセグメントを設定する。同様に、ドメイン(1)の経路計算サーバは、ドメイン(0)において選択された入側境界ノードからドメイン(2)の入側境界ノードまでの最適経路を計算する。そして、計算結果として得られた最適経路に沿ってドメイン(1)からドメイン(2)のドメイン間パスセグメントを設定する。この処理を着ノードD0が属する着ドメイン(ドメイン(N))まで逐次行うことにより、発ノードS0から着ノードD0までのドメイン間パスが設定される。

また、非特許文献3に記載されたクランクバック処理では、ドメイン毎経路計算方式において、前ドメインにより選択された入側境界ノードからの空きの経路が存在しない場合は、前ドメインに空き経路がない旨を通知して前ドメインとのドメイン間パスを解放する。そして、前ドメインにより異なる入側境界ノードまでのドメイン間パスが新たに設定されると、新たに選択された入側境界ノードから経路計算をやり直す。クランクバック処理に回数制限を設けないことにより、パス設定損失率はエンドツーエンド方式と同程度になる。

【非特許文献1】JP. Vasseur, ed., "A backward recursive PCE-based computation (BRPC) procedure to compute shortest inter-domain traffic engineering label switched paths," IETF Draft, draft-ietf-pce-brpc-07.txt, Feb. 2008.

【非特許文献2】JP. Vasseur and A. Ayyangar, ed., "A per-domain path computation method for establishing inter-domain traffic engineering (TE) label switched paths (LSPs)," IETF Draft, draft-ietf-ccamp-inter-domain-pd-path-comp-06.txt, Nov. 2007.

【非特許文献3】A Farrel, ed., "Crankback signaling extensions for MPLS and GMPLS RSVP-TE," IETF RFC 4920, July 2007.

【発明の開示】

【発明が解決しようとする課題】

【0003】

しかしながら、非特許文献1に記載されたエンドツーエンド経路計算方式では、経路計算量が比較的多いため、各ドメインのトラフィック負荷が小さい場合には、ドメイン毎経路計算方式と比較して、ドメイン間パス設定に要する時間が長くなる、という問題がある。

一方、非特許文献2に記載されたドメイン毎経路計算方式では、各ドメインのトラフィック負荷が大きい場合には、クランクバック処理の回数が増加するため、ドメイン間パス設定に要する時間が長くなる、という問題がある。

本発明は上記の点に鑑みてなされたものであり、その目的は、各ドメインのトラフィック負荷の状況に応じたドメイン間パス設定方式を用いることにより、ドメイン間パス設定に要する時間を短縮することができる経路計算サーバ、経路計算方法及びプログラムを提供することにある。

【課題を解決するための手段】

【0004】

本発明は上記の課題を解決するためになされたものであり、本発明の一態様は、各ドメインが境界ノードによって互いに接続されるネットワークシステムにおいて、各ドメインに対応して設置され、発ノードから着ノードまでの経路を計算する経路計算サーバであっ

て、ドメイン内のトラヒック負荷の大きさを判定する判定手段と、前記経路を計算する経路計算手段と、経路計算要求を行ったノードに、前記経路計算手段により計算された経路計算結果を通知する通知手段と、を備え、前記経路計算手段は、着ノードのドメインでは、自ドメインのノードから経路計算要求を受信するとドメイン毎経路計算方式で着ノードまでの経路を計算し、前ドメインの経路計算サーバから経路計算結果を受信するとエンドツーエンド経路計算方式で着ノードまでの経路を計算し、着ノードのドメインでないドメインでは、自ドメインのノードから経路計算要求を受信すると、自ドメイン及び次ドメインのトラヒック負荷がともに大きい場合には、エンドツーエンド経路計算方式でトラヒック負荷が小さいドメインまでの経路を計算し、そうでない場合には、ドメイン毎経路計算方式で次ドメインまでの経路を計算し、前ドメインの経路計算サーバから経路計算結果を受信すると、次ドメインのトラヒック負荷が大きい場合には、エンドツーエンド経路計算方式でトラヒック負荷が小さいドメインまでの経路を計算し、そうでない場合には、エンドツーエンド経路計算方式で次ドメインまでの経路を計算することを特徴とする経路計算サーバである。

10

【0005】

また、本発明の一態様は、上記の経路計算サーバにおいて、前記経路計算手段は、エンドツーエンド経路計算方式で経路を計算する場合には、次ドメインの最も多くの境界ノードに至ることができる境界ノードを経由する経路を選択することを特徴とする。

【0006】

また、本発明の一態様は、各ドメインが境界ノードによって互いに接続されるネットワークシステムにおいて、各ドメインに対応して設置される経路計算サーバにおける発ノードから着ノードまでの経路を計算する経路計算方法であって、ドメイン内のトラヒック負荷の大きさを判定するステップと、前記経路を計算するステップと、経路計算要求を行ったノードに、前記経路計算手段により計算された経路計算結果を通知するステップと、を有し、着ノードのドメインでは、自ドメインのノードから経路計算要求を受信するとドメイン毎経路計算方式で着ノードまでの経路を計算し、前ドメインの経路計算サーバから経路計算結果を受信するとエンドツーエンド経路計算方式で着ノードまでの経路を計算し、着ノードのドメインでないドメインでは、自ドメインのノードから経路計算要求を受信すると、自ドメイン及び次ドメインのトラヒック負荷がともに大きい場合には、エンドツーエンド経路計算方式でトラヒック負荷が小さいドメインまでの経路を計算し、そうでない場合には、ドメイン毎経路計算方式で次ドメインまでの経路を計算し、前ドメインの経路計算サーバから経路計算結果を受信すると、次ドメインのトラヒック負荷が大きい場合には、エンドツーエンド経路計算方式でトラヒック負荷が小さいドメインまでの経路を計算し、そうでない場合には、エンドツーエンド経路計算方式で次ドメインまでの経路を計算することを特徴とする経路計算方法である。

20

30

【0007】

また、本発明の一態様は、上記の経路計算方法において、エンドツーエンド経路計算方式で経路を計算する場合には、次ドメインの最も多くの境界ノードに至ることができる境界ノードを経由する経路を選択することを特徴とする。

【0008】

また、本発明の一態様は、各ドメインが境界ノードによって互いに接続されるネットワークシステムにおいて、各ドメインに対応して設置され、発ノードから着ノードまでの経路を計算するためのコンピュータに、ドメイン内のトラヒック負荷の大きさを判定するステップと、前記経路を計算するステップと、経路計算要求を行ったノードに、前記経路計算手段により計算された経路計算結果を通知するステップと、を実行させるものであり、着ノードのドメインでは、自ドメインのノードから経路計算要求を受信するとドメイン毎経路計算方式で着ノードまでの経路を計算し、前ドメインの経路計算サーバから経路計算結果を受信するとエンドツーエンド経路計算方式で着ノードまでの経路を計算し、着ノードのドメインでないドメインでは、自ドメインのノードから経路計算要求を受信すると、自ドメイン及び次ドメインのトラヒック負荷がともに大きい場合には、エンドツーエンド

40

50

経路計算方式でトラヒック負荷が小さいドメインまでの経路を計算し、そうでない場合には、ドメイン毎経路計算方式で次ドメインまでの経路を計算し、前ドメインの経路計算サーバから経路計算結果を受信すると、次ドメインのトラヒック負荷が大きい場合には、エンドツーエンド経路計算方式でトラヒック負荷が小さいドメインまでの経路を計算し、そうでない場合には、エンドツーエンド経路計算方式で次ドメインまでの経路を計算することを特徴とするプログラムである。

【 0 0 0 9 】

また、本発明の一態様は、上記のプログラムにおいて、エンドツーエンド経路計算方式で経路を計算する場合には、次ドメインの最も多くの境界ノードに至ることができる境界ノードを経由する経路を選択することを特徴とする。

【 発明の効果 】

【 0 0 1 0 】

本発明によれば、ドメインのトラヒック負荷が大きい場合には、エンドツーエンド経路計算方式で経路を計算し、そうでない場合には、ドメイン毎経路計算方式で経路を計算する。これにより、各ドメインのトラヒック負荷の状況に適したドメイン間パス設定ができるため、ドメイン間パス設定に要する時間を短縮することができる。

【 発明を実施するための最良の形態 】

【 0 0 1 1 】

以下、図面を参照しながら本発明の実施形態について詳しく説明する。

図 1 は、本発明の一実施形態によるドメイン間パス設定処理を説明するための概略図である。

この図に示す例では、パスの始点である発ノード S からパスの終点である着ノード D までの経路設定を行う。発ノード S が属するドメイン (0) から着ノード D が属するドメイン (N) までの経路には複数のドメインが存在する。各ドメインには、ドメイン間パスの経路決定を行う経路計算サーバ 1 が設置されている。また、各ドメインは、出側境界ノードと入側境界ノードによって接続される。出側境界ノードとは、他のドメインのノードに接続可能なノードである。入側境界ノードとは、他のドメインのノードから接続可能なノードである。なお、本実施形態では、ドメイン (i) の j 番目の入側境界ノードを B_{ij} と表わす。また、ドメイン (i) の入側境界ノード数を k_i と表わす。

【 0 0 1 2 】

エンドツーエンド経路計算方式は、発ノード S から着ノード D までの経路を一旦全て計算してからドメイン間パス設定を行う。このため、トラヒック負荷が大きい場合であっても、空いていない経路に沿ってドメイン間パスを設定することがない。しかしながら、計算量が多いため、トラヒック負荷が小さい場合には、ドメイン毎経路計算方式に比してドメイン間パス設定までの時間が長くなる。一方、ドメイン毎経路計算方式は、ドメイン毎にドメイン間パス設定を行う。このため、計算量が少なく短い時間でドメイン間パスの設定ができる。しかしながら、トラヒック負荷が大きい場合には、次ドメインでは空いていない経路に沿ってドメイン間パスを設定することがあるためクランクバック処理が頻繁に発生し、ドメイン間パスを再設定するための余計な時間がかかる。よって、トラヒック負荷が大きいドメインではエンドツーエンド経路計算方式が適しており、トラヒック負荷が小さいドメインではドメイン毎経路計算方式が適しているといえる。

【 0 0 1 3 】

このため、本実施形態では、トラヒック負荷が大きいドメインでは、エンドツーエンド経路計算方式を用いてドメイン間パスを設定する。一方、トラヒック負荷が小さいドメインでは、ドメイン毎経路計算方式を用いてドメイン間パスを設定する。

本例では、発ノード S が属するドメイン (0) からドメイン (m) までのトラヒック負荷が大きい。また、ドメイン (m + 1) からドメイン (n) までのトラヒック負荷は小さい。また、ドメイン (n + 1) から着ドメイン D の属するドメイン (N) までのトラヒック負荷は大きい。よって、ドメイン (0) からドメイン (m) まで及びドメイン (n + 1) からドメイン (N) まではエンドツーエンド経路計算方式でドメイン間パスを設定する

。一方、ドメイン ($m + 1$) からドメイン (n) まではドメイン毎経路計算方式を用いてドメイン間パスを設定する。

【 0 0 1 4 】

次に、本実施形態におけるドメイン間パス設定手順について説明する。

まず、発ノード S が、ドメイン (0) の経路計算サーバ 1 に経路計算要求を送信する。経路計算要求は、ドメイン間パスの経路計算を要求するメッセージである。

次に、ドメイン (0) からドメイン (m) までの経路計算サーバ 1 が、連携してドメイン (0) からドメイン ($m + 1$) までの経路を決定する。そして、決定した経路は、ドメイン (0) の経路計算サーバ 1 から発ノード S に通知される。これにより、発ノード S からドメイン ($m + 1$) の入側境界ノード B ($m + 1$) 0 までのドメイン間パスが設定される。

次に、入側境界ノード B ($m + 1$) 0 が、ドメイン ($m + 1$) の経路計算サーバ 1 に経路計算要求を送信する。すると、ドメイン ($m + 1$) の経路計算サーバ 1 は、次ドメインの入側境界ノードまでの経路を計算して入側境界ノード B ($m + 1$) 0 に通知する。これによりドメイン ($m + 1$) からドメイン ($m + 2$) までのドメイン間パスが設定される。この処理をドメイン (n) まで繰り返す。

次に、ドメイン ($n + 1$) では、入側境界ノード B ($n + 1$) 0 が、経路計算要求を経路計算サーバ 1 に送信する。次に、ドメイン ($n + 1$) からドメイン (N) までの経路計算サーバ 1 が、連携して入側境界ノード B ($n + 1$) 0 から着ノード D までの経路を決定する。そして、決定した経路は、ドメイン ($n + 1$) の経路計算サーバ 1 から入側境界ノード B ($n + 1$) 0 に通知される。これにより、入側境界ノード B ($n + 1$) 0 から着ノード D までのドメイン間パスが設定される。

このような手順により、発ノード S から着ノード D までのドメイン間パスが設定される。

【 0 0 1 5 】

図 2 は、本実施形態における経路計算サーバ 1 の構成を示すブロック図である。

経路計算サーバ 1 は、通信部 1 1 と、トラヒック測定部 1 2 と、トラヒック取得部 1 3 と、経路計算処理部 1 4 と、を含んで構成される。

通信部 1 1 は、自身が管理するドメイン (以下、自ドメインとする) のノード及び他のドメインの経路計算サーバ 1 と通信する。

【 0 0 1 6 】

トラヒック測定部 1 2 は、自ドメインのトラヒック負荷が大きい小さいかを判定する。具体的には、まず、トラヒック測定部 1 2 は、平均パス保留時間あたりに到着する経路計算要求の数を計算する。平均パス保留時間は、実験などにより測定された実測値に基づいて予め決定された定数である。そして、トラヒック測定部 1 2 は、計算した結果が予め設定された閾値より大きい場合に、トラヒック負荷が大きいと判定する。一方、トラヒック測定部 1 2 は、閾値より小さい場合には、トラヒック負荷が小さいと判定する。なお、本実施形態では、平均パス保留時間を用いてトラヒック負荷を算出したが、パス同時接続数を用いて算出してもよい。パス同時接続数は、ドメイン間パスの設定数とドメイン間パスの解放数の差である。また、トラヒック測定部 1 2 は、トラヒック負荷の判定結果を隣接するドメインの経路計算サーバ 1 へ通知する。

【 0 0 1 7 】

トラヒック取得部 1 3 は、隣接するドメインの経路計算サーバ 1 からトラヒック負荷を取得する。

経路計算処理部 1 4 は、自ドメインとパスの経路上で次にあるドメイン (以下、次ドメインとする) のトラヒック負荷に応じて、経路計算要求のあったドメイン間パスの経路を計算する。

【 0 0 1 8 】

以下、経路計算処理部 1 4 の処理の流れについて説明する。

図 3 は、本実施形態における経路計算処理部 1 4 の処理の手順を示すフローチャートで

10

20

30

40

50

ある。この図に示す処理は、自ドメインのノードから経路計算要求を受信した際にスタートする。

まず、ステップ S 1 では、自ドメインが着ドメインであるか否かを判定する。着ドメインとは、着ノード D が属するドメインである。図 1 に示す例ではドメイン (N) が着ドメインである。着ドメインである場合には、ステップ S 2 へ進む。一方、着ドメインでない場合には、ステップ S 6 へ進む。

【 0 0 1 9 】

ステップ S 2 では、経路計算要求のあったノード (入側境界ノード又は発ノード) から着ノードまでの最もコストの低い空き経路を計算する。

次に、ステップ S 3 では、ステップ S 2 の経路計算の結果、経路計算要求のあったノードから着ノードまで空き経路があるか否かを判定する。空き経路がある場合には、ステップ S 4 へ進む。一方、空き経路がない場合には、ステップ S 5 へ進む。

ステップ S 4 では、経路計算要求のあったノードにステップ S 2 で計算した経路を通知する。これにより、経路計算要求のあったノードから着ノードまでのパスが設定される。

一方、ステップ S 5 では、経路計算要求のあったノードに空き経路がない旨を通知する。これにより、経路計算要求のあったノードが入側境界ノードである場合には、パスの経路上で前にあるドメイン (以下、前ドメインとする) にてクランクバック処理が行われる。

【 0 0 2 0 】

一方、ステップ S 6 では、自ドメインのトラヒック負荷が大きいか否かを判定する。トラヒックが小さい場合には、ステップ S 7 へ進む。一方、トラヒック負荷が大きい場合には、ステップ S 10 へ進む。

【 0 0 2 1 】

ステップ S 7 では、経路計算要求のあったノード (入側境界ノード又は発ノード) から次ドメインの全ての入側境界ノードまでの経路計算を行う。図 1 に示すドメイン (n) を例に具体的に説明する。まず、経路計算サーバ 1 は、経路計算要求のあったノード B n 0 からノード B (n + 1) 0 までの経路を計算する。この処理をノード B (n + 1) 0 からノード B (n + 1) k (n + 1) まで繰り返す。これにより、ノード B n 0 からドメイン (n + 1) の全ての入側境界ノード (ノード B (n + 1) 0 ~ ノード B (n + 1) k (n + 1)) までの経路が計算される。

【 0 0 2 2 】

次に、ステップ S 8 では、ステップ S 7 の経路計算の結果、経路計算要求のあったノードからの空き経路があるか否かを判定する。空き経路がある場合には、ステップ S 9 へ進む。一方、空き経路が無い場合には、ステップ S 5 へ進む。

ステップ S 9 では、経路計算要求のあったノードに、経路計算要求のあったノードから次ドメインの全ての入側境界ノードまでの経路の内、最もコストの小さい経路を通知する。これにより、自ドメインから次ドメインまでのドメイン間パスが設定される。

【 0 0 2 3 】

一方、ステップ S 10 では、次ドメインのトラヒック負荷が大きいか否かを判定する。次ドメインのトラヒック負荷が大きい場合にはステップ S 11 へ進む。一方、次ドメインのトラヒック負荷が小さい場合には、ステップ S 7 へ進む。

ステップ S 11 では、経路計算要求のあったノードから次ドメインの全ての入側境界ノードまでの経路計算を行う。次に、ステップ S 12 では、ステップ S 11 の結果、経路計算要求のあったノードからの空き経路があるか否かを判定する。空き経路がある場合には、ステップ S 13 へ進む。一方、空き経路がない場合には、ステップ S 5 へ進む。

次に、ステップ S 13 では、次ドメインの経路計算サーバ 1 に、経路計算要求のあったノードから次ドメインの空き経路が存在する全ての入側境界ノードまでの各空き経路の最小コストを通知する。

【 0 0 2 4 】

図 4 は、本実施形態における経路計算処理部 14 の処理の手順を示すフローチャートで

10

20

30

40

50

ある。この図に示す処理は、前ドメインの経路計算サーバ1から空き経路が存在する全ての入側境界ノードまでの各空き経路の最小コストを受信した際にスタートする。

まず、ステップS101では、自ドメインが着ドメインであるか否かを判定する。着ドメインである場合には、ステップS102へ進む。一方、着ドメインでない場合には、ステップS104へ進む。

【0025】

ステップS102では、着ノードまでの経路計算を行う。図1に示すドメイン(N)を例に具体的に説明する。ドメイン(N)の経路計算サーバ1は、全ての入側境界ノード($B_{N0} \sim B_{Nk_N}$)までの各空き経路の最小コストを前ドメイン(N-1)の経路計算サーバから受信している。ドメイン(N)の経路計算サーバ1は、各入側境界ノードから着ノードDまでの最もコストの小さい空き経路をそれぞれ計算する。そして、それぞれの入側境界ノードにおいて、受信した入側境界ノードまでの最小コストを着ノードDまでの空き経路の最小コストに加算する。

ステップS103では、ステップS102で加算した結果、最もコストの小さくなる経路を前ドメインの経路計算サーバ1に通知する。当該経路は、前ドメインの経路計算サーバ1を介して経路計算要求を行ったノードに通知される。これにより、経路計算要求を行ったノードから着ドメインまでのドメイン間パスが設定される。

【0026】

一方、ステップS104では、次ドメインのトラヒック負荷が大きいかな否かを判定する。次ドメインのトラヒック負荷が大きい場合には、ステップS107へ進む。一方、次ドメインのトラヒック負荷が小さい場合には、ステップS105へ進む。

【0027】

ステップS105では、空き経路が存在する全ての入側境界ノードから次ドメインの全ての入側境界ノードまでの最もコストの小さい空き経路を計算する。図1に示すドメイン(m)を例に具体的に説明する。ドメイン(m)の経路計算サーバ1は、全ての入側境界ノード($B_{m0} \sim B_{mk_m}$)までの各空き経路の最小コストをドメイン(m-1)の経路計算サーバ1から受信している。ドメイン(m)の経路計算サーバ1は、各入側境界ノードからドメイン(m+1)の入側境界ノード $B_{(m+1)0}$ までの最もコストの小さい空き経路をそれぞれ計算する。この処理をドメイン(m+1)の全ての入側境界ノード($B_{(m+1)0} \sim B_{(m+1)k_{m+1}}$)に対して行う。そして、各入側境界ノードにおいて、受信した入側境界ノードまでの経路の最小コストを算出した入側境界ノード($B_{(m+1)0} \sim B_{(m+1)k_{m+1}}$)までの各空き経路の最小コストに加算する。

次に、計算した経路から最適な経路を選択する。最適な経路とは、(1)最も多くの次ドメインの入側境界ノードに至ることができる入側境界ノードを経由する経路であって、(2)経路計算の結果得られたコスト最小の経路である。

これは、クランクバック処理が発生した場合に対応するためである。例えば、図1に示すドメイン(m+1)において、経路計算要求を行う入側境界ノード $B_{(m+1)0}$ からの空き経路がない場合、ドメイン(0)からドメイン(m+1)までドメイン間パスの再設定をしなければならない。これを防ぐために、ドメイン(m)では、最も多くの次ドメインの入側境界ノードに接続できる入側境界ノードを経由する経路に沿ってドメイン間パスを設定しておく。これにより、ドメイン(m-1)以前のドメインでパスを再設定する確率が低くなる。このため、ドメイン(m+1)において、経路計算要求を行う入側境界ノード $B_{(m+1)0}$ からの空き経路がない場合であっても、ドメイン(m)からドメイン(m+1)までのドメイン間パスの再設定のみ行えば、クランクバック処理が完了する確率が高くなる。

【0028】

次に、ステップS106では、前ドメインの経路計算サーバ1に、選択された経路を通知する。当該経路は、前ドメインの経路計算サーバ1を経由して経路計算要求を行ったノードに通知される。これにより、経路計算要求を行ったノードから次ドメインまでのドメイン間パスが設定される。

10

20

30

40

50

【 0 0 2 9 】

一方、ステップ S 1 0 7 では、次ドメインの全ての入側境界ノードまでの最もコストの小さい空き経路を計算する。ドメイン (N - 1) を例に具体的に説明する。ドメイン (N - 1) の経路計算サーバ 1 は、全ての入側境界ノード (B (N - 1) 0 ~ B (N - 1) k_{N-1}) までの各空き経路の最小コストをドメイン (N - 2) の経路計算サーバ 1 から受信している。ドメイン (N - 1) の経路計算サーバ 1 は、各入側境界ノードからドメイン (N) の入側境界ノード B N 0 までの最もコストの小さい空き経路をそれぞれ計算する。この処理をドメイン (N) の全ての入側境界ノード (B N 0 ~ B N k_N) について行う。そして、それぞれの入側境界ノードにおいて、受信した入側境界ノードまでの最小コストを入側境界ノード (B N 0 ~ B N k_N) までの各空き経路の最小コストに加算する。最後に、ドメイン (N) の各入側境界ノード (B N 0 ~ B N k_N) までの最適な経路を選択する。最適な経路とは、加算したコストが最も小さい経路である。これにより、ドメイン (N) の全ての入側境界ノード (B N 0 ~ B N k_N) までの各経路の最小コストが決定する。

10

【 0 0 3 0 】

ステップ S 1 0 8 では、次ドメインの経路計算サーバ 1 に、ステップ S 1 0 7 で計算した次ドメインの全ての入側境界ノードまでの各空き経路の最小コストを通知する。次ドメインの経路計算サーバ 1 は、全ての入側境界ノードまでの各空き経路の最小コストを受信すると、上述したステップ S 1 0 1 からの処理を行う。

【 0 0 3 1 】

このように、本実施形態によれば、トラフィック負荷が大きいドメイン群ではエンドツーエンド経路計算方式を用いてドメイン間パスを設定し、トラフィック負荷が小さいドメイン群ではドメイン毎経路計算方式を用いてドメイン間パスを設定している。これにより、各ドメインのトラフィック負荷の状況に適したドメイン間パス設定ができるため、ドメイン間パス設定に要する時間を短縮することができる。

20

【 0 0 3 2 】

また、図 2 に示す経路計算サーバの機能を実現するためのプログラムをコンピュータ読み取り可能な記録媒体に記録して、この記録媒体に記録されたプログラムをコンピュータシステムに読み込ませ、実行することにより、経路計算処理を行ってもよい。なお、ここでいう「コンピュータシステム」とは、OS や周辺機器等のハードウェアを含むものであってもよい。

30

また、「コンピュータシステム」は、WWWシステムを利用している場合であれば、ホームページ提供環境（あるいは表示環境）も含むものとする。

また、「コンピュータ読み取り可能な記録媒体」とは、フレキシブルディスク、光磁気ディスク、ROM、フラッシュメモリ等の書き込み可能な不揮発性メモリ、CD-ROM等の可搬媒体、コンピュータシステムに内蔵されるハードディスク等の記憶装置のことをいう。

【 0 0 3 3 】

さらに「コンピュータ読み取り可能な記録媒体」とは、インターネット等のネットワークや電話回線等の通信回線を介してプログラムが送信された場合のサーバやクライアントとなるコンピュータシステム内部の揮発性メモリ（例えばDRAM（Dynamic Random Access Memory））のように、一定時間プログラムを保持しているものも含むものとする。

40

また、上記プログラムは、このプログラムを記憶装置等に格納したコンピュータシステムから、伝送媒体を介して、あるいは、伝送媒体中の伝送波により他のコンピュータシステムに伝送されてもよい。ここで、プログラムを伝送する「伝送媒体」は、インターネット等のネットワーク（通信網）や電話回線等の通信回線（通信線）のように情報を伝送する機能を有する媒体のことをいう。

また、上記プログラムは、前述した機能の一部を実現するためのものであっても良い。さらに、前述した機能をコンピュータシステムにすでに記録されているプログラムとの組み合わせで実現できるもの、いわゆる差分ファイル（差分プログラム）であっても良い。

50

【 0 0 3 4 】

以上、図面を参照してこの発明の一実施形態について詳しく説明してきたが、具体的な構成は上述のものに限られることはなく、この発明の要旨を逸脱しない範囲内において様々な設計変更等を行うことが可能である。

【図面の簡単な説明】

【 0 0 3 5 】

【図 1】本発明の一実施形態によるドメイン間パス設定処理を説明するための概略図である。

【図 2】本実施形態における経路計算サーバの構成を示すブロック図である。

【図 3】本実施形態における経路計算処理部の処理の手順を示すフローチャートである。

【図 4】本実施形態における経路計算処理部の処理の手順を示すフローチャートである。

【図 5】エンドツーエンド経路計算方式を用いたドメイン間パス設定処理を説明するための概略図である。

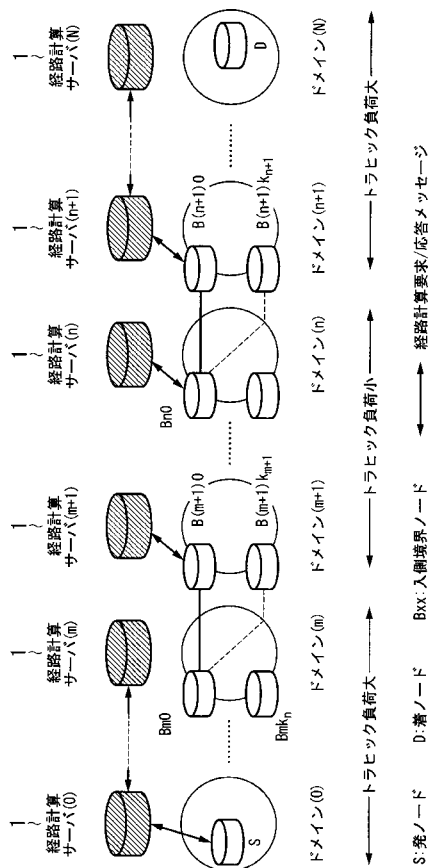
【図 6】ドメイン毎経路計算方式を用いたドメイン間パス設定処理を説明するための概略図である。

【符号の説明】

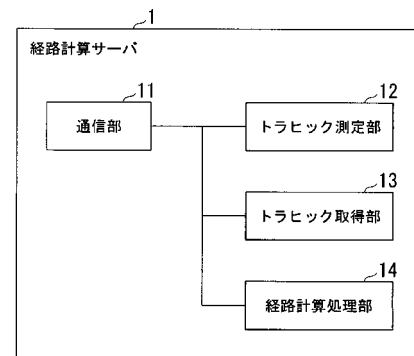
【 0 0 3 6 】

1 ... 経路計算サーバ 1 1 ... 通信部 1 2 ... トラヒック測定部 1 3 ... トラヒック取得部
部 1 4 ... 経路計算処理部

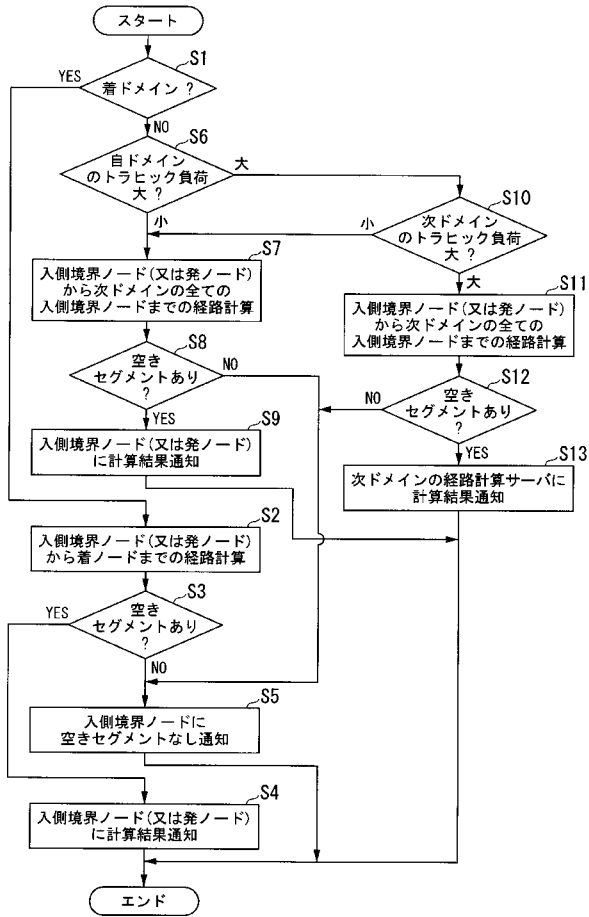
【図 1】



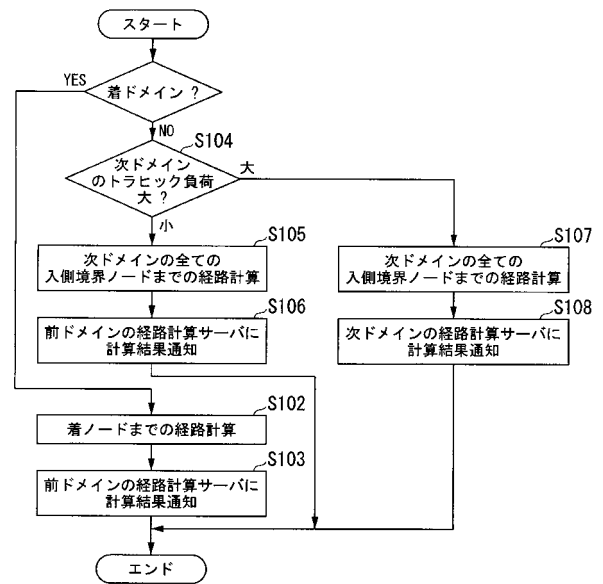
【図 2】



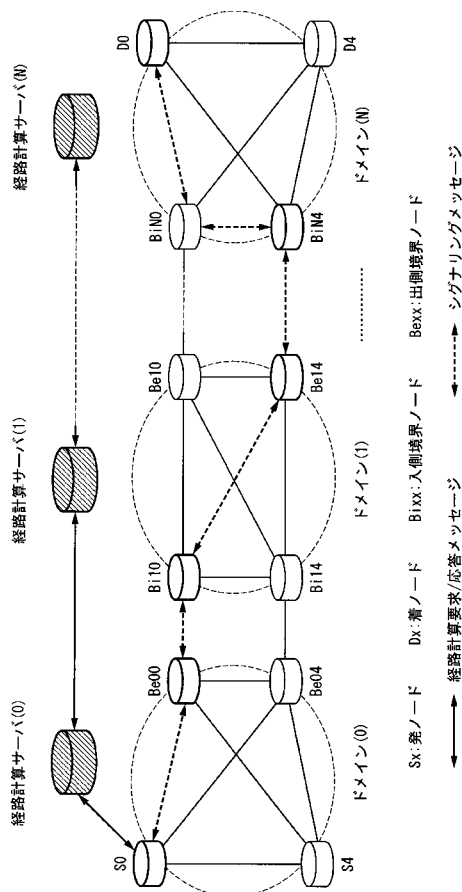
【図 3】



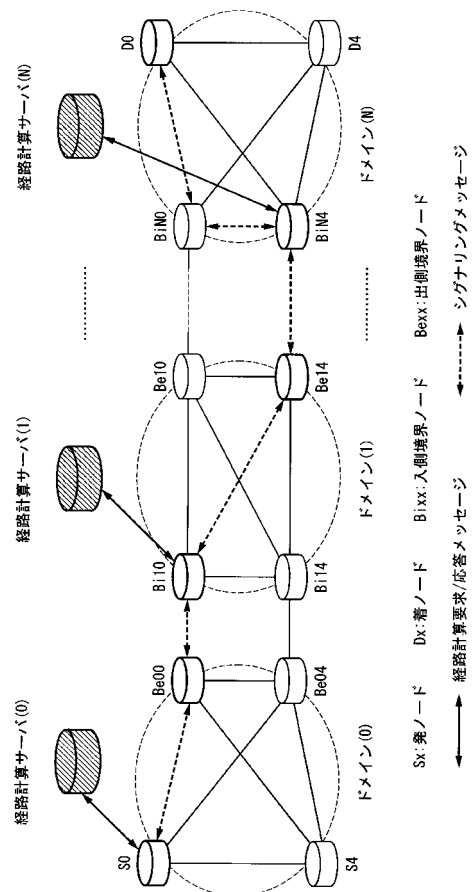
【図 4】



【図 5】



【図 6】



フロントページの続き

(72)発明者 中村 元

埼玉県ふじみ野市大原 2 丁目 1 番 1 5 号 株式会社 K D D I 研究所内

F ターム(参考) 5K030 GA01 HA08 JA07 LB05