

分散ルーティングアーキテクチャにおける絞込みエージェントの性能評価

福岡龍一†, 池田 誠†, バロリ レオナルド‡,
デマルコ ジュゼッペ†‡, 小山 明夫†‡

† 福岡工業大学大学院工学研究科

‡ 福岡工業大学情報工学部情報通信工学科

†‡ 豊田工業大学

‡‡ 山形大学工学部情報工学科

近年、トラフィックの増加によりその制御が難しくなっている。そこで我々はこれまでに分散ネットワークアーキテクチャを提案してきた。今回はその分散ルーティングアルゴリズムにおける絞込みエージェントの性能評価を行った。本稿では従来手法の性能評価を行いその問題点を解決するために新たな手法を提案する。シミュレーションを行った結果、提案手法は従来手法より出力経路数や処理時間を改善することができた。

Performance Evaluation of a Network Search Space Reduction Agent in a Distributed Routing Architecture

Ryuichi Fukuoka †, Makoto Ikeda †, Leonard Barolli ‡,
Giuseppe De Marco ‡, Akio Koyama †‡

† Graduate School of Engineering, Fukuoka Institute of Technology

E-mail: {mgm06006, bd07001}@ws.ipc.fit.ac.jp

‡ Faculty of Information Engineering, Fukuoka Institute of Technology

E-mail: barolli@fit.ac.jp

†‡ Toyota Technological Institute

E-mail: demarco@toyota-ti.ac.jp

‡‡ Faculty of Engineering, Yamagata University

Recently, the network control is becoming difficult because of the increase of network traffic. To deal with this problem, we have proposed a distributed network architecture by using different agents. In this paper we evaluate the performance of a network Search Space Reduction Agent (SSRA). We first evaluate the performance of a previous SSRA and point out its drawbacks. Then, we propose a new SSRA to improve the previous SSRA performance. The performance evaluation shows that the proposed SSRA has better behavior than the previous SSRA considering the delay time and the number of output routes.

1. はじめに

今日、インターネットの発展に伴いインターネットの人口普及率は68.5%となり、その利用者数は8754万人になっている[1]。また、インターネットサービスの発展によりその情報量が増加している。このため、効率的なトラフィック制御が必要となっている。それにより、クライアントが求めるサービス品質(QoS)を考慮したルーティングが重要となっている[2]。更にクライアントが求めるQoSは単一とは限らず、複

数存在する場合もある。それらのQoSにはロードオフ関係にある場合も多い。以上の背景からクライアントが求める複数のQoSを満たすルーティング方法[3]の研究を行った。

本研究の目的として、ネットワークの規模拡大やトラフィックの増加に伴い、効率の良い通信を行うためには、輻輳状態にある経路を回避し、より遅延時間の少ない経路を選択することによって、通信回線にかかる負荷を分散させるルーティングをめざした。通信経路を固定化するの

ではなく、常に変化しているネットワーク環境に応じて動的に通信経路を決定する。そこで、遺伝的アルゴリズム、複数のエージェント[4]、キャッシュエンジンを用いて早期により効率的な経路を探索するアルゴリズムが提案された。

本研究では複数あるエージェントの1つである SSRA (Search Space Reduction Agent) と TMNA について、新たな機能の追加を提案した。

本論文では、2 章では、本手法における分散型ネットワークアーキテクチャーの構成、各エージェントの特徴を示す。3 章では、従来の問題点と提案手法を示し、4 章では提案手法のシミュレーションについて環境及び条件を述べ、5 章では結果と考察を示す。そして、最後 6 章でまとめを述べる。

2. 分散型ネットワークアーキテクチャー

図 1 に分散型ネットワークアーキテクチャーを示す。ネットワークを複数のドメインに分割し [5]、各ドメインで複数のエージェント (図 2 参照) を用いルーティングを行う。各エージェントについて説明していく。まず、Resource Management Agent (RMA) で Fuzzy を用いた接続制御を行う。Precomputation Agent (PA) は、SSRA と、Tree Model Network Agent (TMNA) で構成される。SSRA はしきい値を用いて輻輳状態にあるリンクの絞り込みを行う。そして TMNA でループを統合したツリーに変換する。Destination Discovery Agent (DDA) は宛先ノードがどこにあるか探索するエージェント、Routing Agent (RA) は階層的エージェントで、Inter Domain Agent (InterD) その内側に Intra Domain Agent (IntraD) と Connectivity Management Agent (CMA) で構成される。CMA はドメイン間の接続制御を行う。IntraD Agent 内で GA を用いている。複数のドメインに分割し経路制御を行うことにより、ネットワーク規模の拡大にも柔軟に適応することが可能となる。

本論文の提案手法では、適応型アルゴリズム [6,7] の分散型ルーティングと始点制御ルーティングを用いている。分散・適応型ルーティングは、ネットワークの大域情報と局所情報を併用し、より詳細な情報を用いている。そして、ネットワーク環境に応じたルーティングを行う。始点制御ルーティングとは、パケットを送信するノードにおいて最終目的地までの完全な経路を決定するルーティング手法である。つまり、通信パケットの送信元ですべての経路情報を管理する。利点は、大域的なトラフィックの把握ができるため、効率の良い経路選択が可能である。一方、リソース情報を各ノードが分散して

持っている場合と比較したとき、通信経路を動的に変化させるのは非常に難しくなる。始点制御のデメリットとして、ネットワーク規模が大きい場合、到達可能な経路の数が膨大になり、データの扱いが大変になるという欠点もある。

2.1 RMA

リソース管理エージェントでは Fuzzy 理論を用いた接続制御を行う [8]。FAC: (Fuzzy Admission Control) スキームを図 3 に示す。

2.2 PA

2.2.1 SSRA

SSRA は、RMA から入力されたトポロジーをもとにノードやリンクのパラメータに閾をかけて縮小したトポロジーを出力するエージェントである。SSRA を用いるメリットとして、大規模で帯域幅にムラがあるネットワークを、規模を縮小し、各ノード間の帯域を均一化することが可能になり輻輳も回避することが出来る。図 4 は SSRA のフローチャートである。

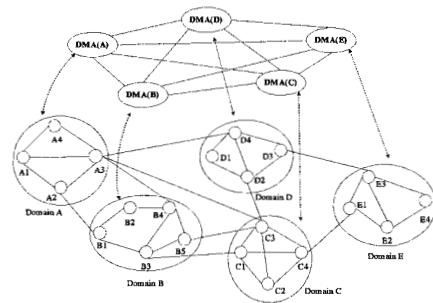


図 1 分散型ネットワークアーキテクチャー

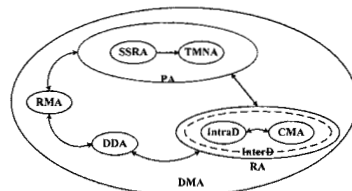


図 2 ドメイン管理エージェント

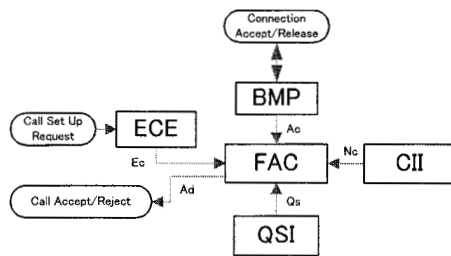


図 3 FAC スキーム

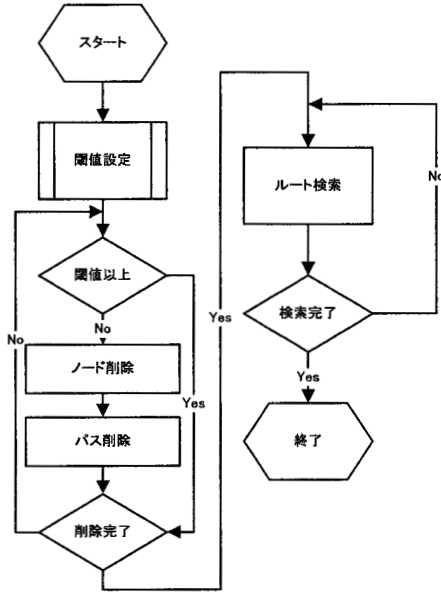


図4 SSRA flowchart.

2.2.2 TMNA

TMNA は SSRA から送られてきた効率的なトポロジーを木構造に変換するエージェントである。図5はAからHまでの8つのノードで構成された小規模なネットワークです。ノードAをソースノードとし、ノードHを宛先ノードとする。図6を見ると、ノードCからHまで同じ経路を示す場所がある。そこで同じ経路を、ひとつにまとめることを行うことにより(図7参照), 木を縮小することが可能となり、ループを避けることが可能となる。

2.3 DDA

DDA は目的地ノードと始点ノードが同じドメインにあるか判断し、その情報を伝達するエージェントである。

2.4 IntraD Agent

IntraD Agent は GA に基づく、遺伝的アルゴリズム(Genetic Algorithm: GA)は生物の進化を工学的にモデル化したアルゴリズムである。図8に遺伝的アルゴリズムのサイクルを示す。GAは、選択や交叉、突然変異といった遺伝的操作を各個体に適用し、遺伝的要素の組み換えを行うことにより次世代の個体を作り、個体集団を進化させる。

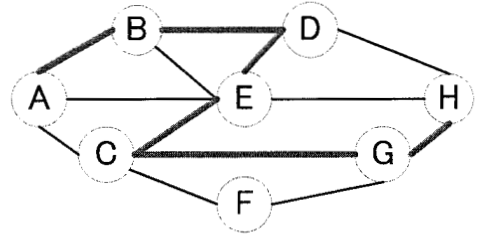


図5 A network example with 8 nodes.

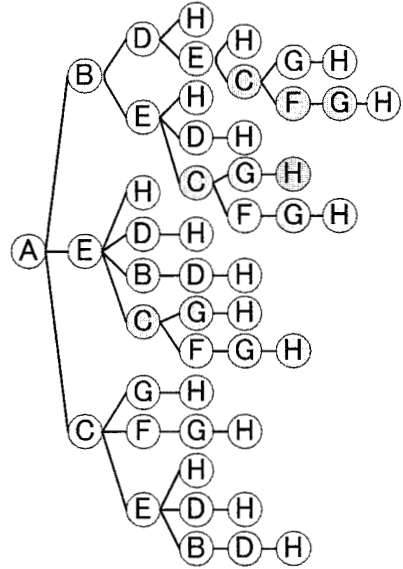


図6 Network tree model.

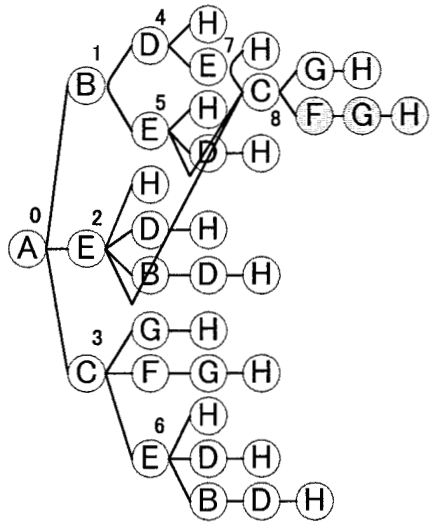


図7 Reduced network tree model.

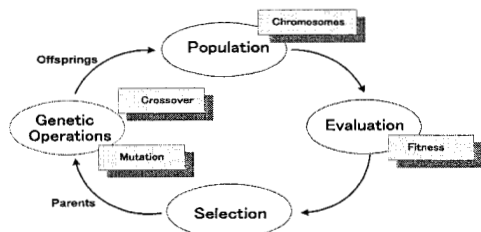


図8 遺伝的アルゴリズムのサイクル

0	1	2	3	4	5	6	7	8				
B	E	C	D	E	H	D	C	H	B	C	G	F
↓	↓			↓				↓	↓			
B	D			E				C	F			
0	1	2	3	4	5	6	7	8				
B	D	-	-	E	-	-	C	F				
1	1	0	0	1	0	0	1	1				

図9 遺伝子マッピング

IntraD Agent は遺伝子コーディングを行うために各分岐点を遺伝子として、その並びによって示される染色体から経路が決定される。ここでの遺伝子とは図9にあるような配列を取ったときの各配列番号で取りうる値(1番目においては D, E)を指し、染色体とはその並びによって表される1つの配列である。

2.5 InterD Agent

InterD Agent は IntraD Agent と CMA から構成される。それらは段階的に相互ドメインに拡大してルーティングを行う。段階的に行うことで、情報交換はドメイン領域だけで済む。全てのドメインで情報をフローディングさせる必要なく、ネットワークのリソースを効率的に使うことが可能となる。

3. 従来手法と提案手法

この章では絞り込みエージェントである SSRA 及びトポロジーを木構造に変換する TMNA の従来手法と提案手法について説明する。

3.1 従来手法

従来手法は、はじめにノード数を決定しそれぞれのノードを一定の確率で連結させる。リンクのパラメータは遅延時間、パケット損失、帯域幅の3つを用い、それぞれ一定の範囲でランダムに値をとる。その後、それぞれのパラメータごとに閾をとりトポロジーを縮小する。閾の

かけ方はノード数が多くなるごとに強くかけるようにした。次にトポロジーを木構造に変換しすべての経路情報を出力する。出力された経路情報をもとに遅延時間及びパケット損失ではそれぞれの経路でのパラメータ値の合計で閾をとる。帯域幅は経路中のパラメータ値がすべて閾値より良いものだけを出力する。最後にそれぞれのパラメータの出力結果を合わせて最終的な出力とする。

図10はSSRA前のトポロジーを図にした全経路を表すもので、図11はSSRAで閾をかけて絞込みを行った後のトポロジーを表したものである。

この従来手法は次の問題点がある。

1. 処理が重く、数時間かかる場合がある。
2. 出力される経路数が膨大になる。
3. 閾を強くかけすぎると経路がまったくでない。
4. 出力される結果に大きなばらつきがある。

3.2 提案手法

本稿では新たに以下のような操作を行った。

1. ソースとディスティネーションにつながるリンクを閾にかけないようにする。
2. 経路の絞込みは目標値より大きいときのみかけるようにする。
3. 閾を平均値+補正值とする。

従来手法ではすべてのリンクに均一に閾をかけていた。そのため閾を強くかけると経路が出ないといった問題点があった。そこでソースとディスティネーションにつながるリンクを閾にかけないようにしたことにより経路数が0になる確率が低くなりより強く閾をかけられるようになった。

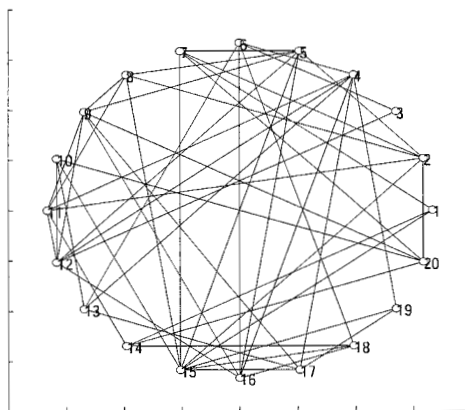


図10 全経路

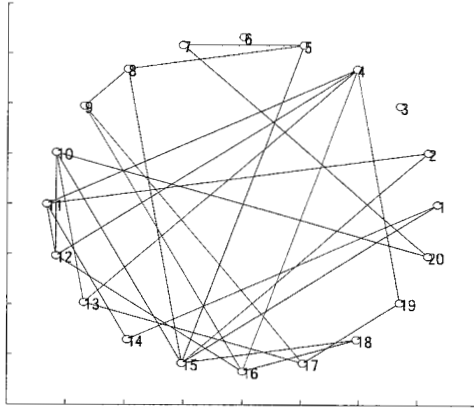


図 11 出力経路

表 1 補正值

ノード数	補正值
100	17
90	15
80	13
70	11
60	9
50	5
40	0

また、経路の絞込みは常に行うようにしていた。しかし、それでは出力経路が少ない場合に経路が出力されない場合がある。そこで、経路の絞込みは目標値より大きいときのみ操作を行うようにしたため、必要以上に経路数を絞り込まないため、処理操作が減り時間も軽減できる。

最後に、これまでは閾値を一定にしていたため、パラメータがランダムに変化すると出力結果に大きなばらつきが出ていた。そこで閾を平均値+補正值にすることにより、閾値がパラメータ値の変化を考慮することができるようになり、出力のばらつきを抑える。補正值は経験則によって決定し表 1 のようになった。補正值は 0 のとき補正なしとなり、数値が大きくなるにしたがって強く閾をかけるようになる。

4 シミュレーション環境

シミュレーション環境は以下のとおりである。
 CPU: Athlon64+3200, Memory: 1024MB,
 OS: Windows XP Pro, IDE: MATLAB 6.5.

シミュレーションは、従来手法でノード数 20, 30, 40, 提案手法で 40, 50, 60, 70, 80, 90,

100 を用いて行った。従来手法でノード数が少ないのは、40 を超えるノード数では処理が重くデータ取りが困難だったためである。また提案手法が 100 ノードまでなのはドメイン内のノード数を 100 と想定しているためである。これは、遺伝的アルゴリズムを用いたルーティングが小規模ネットワークに向いているためである。

パラメータは遅延時間、パケット損失、帯域幅の 3 つのパラメータを用い、それぞれ一定の範囲でランダムに値をとる。

結果は 30 回シミュレーションを行った平均とする。出力される結果は、SSRA を用いた後に出力される全経路数、TMNA を用いた後に出力される出力経路数及び処理時間である。

5. シミュレーション結果と考察

シミュレーションを行った結果を図 12, 13, 表 2, 3 に示す。

図 12, 13 を見ると従来手法に比べて提案手法では、ノード数が増えていっても全経路数が増加しないことがわかった。値がやや上下しているのは、パラメータがランダムであることと補正值が最適でないことが考えられる。出力経路数では、平均の場合において目標値内に収めることができた。

提案手法では処理時間はノード数の増加とともに増えていったが、全経路数及び出力経路数においてはノード数が増えていってもそれほど大きな変動は見られなかった。また、表 2, 3 を見ると従来手法で見られた経路数のばらつきもある程度抑えられた。

提案手法では従来手法よりさらに多くのノードでシミュレーションができるようになった。また全経路数、出力経路数、処理時間や経路数のばらつきを減少させ、出力経路数を目標値に抑えることができた。

以上のことにより提案手法 1 の問題点を改善することができた。

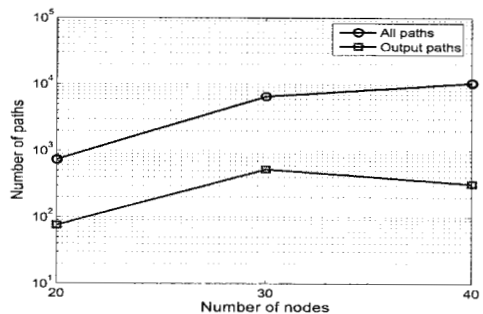


図 12 従来手法のシミュレーション結果

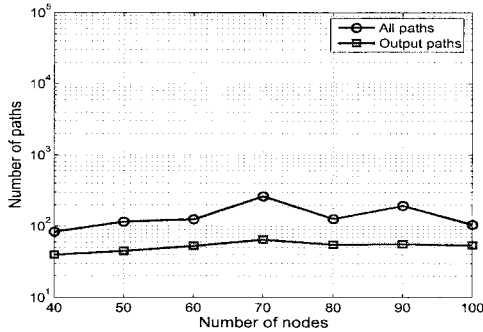


図 13 提案手法のシミュレーション結果

表 2 従来手法のノード数 30 の場合

全経路数	出力経路数	処理時間
13825	455	20.891
1206	137	2.812
11893	1225	36.219
13825	455	20.735
1206	137	2.828
11893	1225	33.984
13825	455	20.641
1206	137	2.844
11837	1225	35.328
13825	455	20.282

表 3 提案手法のノード数 100 の場合

全経路	出力経路	処理時間
19	19	1.938
31	31	1.968
63	63	1.985
95	95	2.109
96	96	2.203
48	48	2.047
78	78	2.078
134	52	2.219
72	72	2.156
132	94	2.281

6. まとめ

本稿では、経路数及び処理時間の減少、また出力される経路数のばらつきを抑えるために、SSRA と TMNA に新たな機能の追加を提案し、その性能の比較、評価を行った。

シミュレーションの結果より従来手法の問題点であった処理に時間が掛かる、経路数が膨大になるなどを提案手法では改善することができた。今後の課題として処理の並列化などを行い、処理時間を減少させていきたい。

参考文献

- [1] 総務省 通信利用動向調査, 10 May 2007.
- [2] C. Baransel, W. Dobosiewicz and P. Gburzyuski, "Routing in Multihop Packet Switching Networks: Gb/s Challenge", IEEE Network, Vol.9, No.3, pp.38-60, 1995.
- [3] A. Koyama, L. Barolli, K. Matsumoto, B.O.Apduhan, "A GA-based Multipurpose Optimization Algorithm for QoS Routing", Proc. of IEEE AINA-2004, Vol.1, pp. 3-28, March 2004.
- [4] M. Ikeda, L. Barolli, S. Ohba, G. Capi, A. Koyama, M. Duresi, "A CAC and Routing Framework for Multimedia Applications in Broadband Networks Using Fuzzy Logic and Genetic Algorithm", Proc. of IEEE ICPADS-2005, Vol.1, pp. 648-654, July 2005.
- [5] 池田 誠, "ヒューリスティック手法を用いた次世代ネットワークのアーキテクチャとアルゴリズムに関する研究", 福岡工業大学 大学院工学研究科 修士論文 February 2007.
- [6] 棟朝雅晴, 高井昌彰, 佐藤義治, "遺伝的アルゴリズムによる負荷分散機能を有する適応型ルーティング", 情報処理学会論文誌, Vol. 39, No. 2, pp. 219-226, 1998.
- [7] M. Munetomo, Y. Takai and Y. Sato, "An Adaptive Routing Algorithm with Load Balancing by a Genetic Algorithm", Trans. of IPSJ, Vol.39, No.2, pp.219-227, 1998.
- [8] D. Dubois, H. Prade, and R. Yager, (Eds.), "Fuzzy Sets for Intelligent Systems", Morgan Kaufman Publishers, 1993.