

複数メトリックを用いたルーティング方法の提案と その充足すべき条件に関する一考察

森山 敦文[†] 石西 洋[†] 中村 勝一[†] 堀 良彰^{††}

[†] 株式会社ネットワーク応用技術研究所 (NAL)

〒 812-0011 福岡県福岡市博多区博多駅前 1-4-4 JPR 博多ビル 6 階

^{††} 九州大学

〒 819-0395 福岡県福岡市西区大字元岡 744 番地

E-mail: †{moriyama,ishinishi,nakamura}@nalab.jp, ††hori@inf.kyushu-u.ac.jp

あらまし 今日の経路制御は、ダイクストラ法を使った OSPF やベルマンフォード法を使った RIP が一般的だが、複数のメトリックに対応していない。又、複数メトリックの場合、加重平均や各種のオペレーションズリサーチの手法を使うこともあるが、妥当性や簡便性の観点からして、経路制御には不適切である。本研究では、I. A. Almerhag と M. E. Woodward 両氏が提案したセキュリティメトリック 3 種類に対して、ルーティングポリシーを設定して経路制御を行なう方法を提案する。更に、提案アルゴリズムの動作条件として、メトリックやポリシーに対する制約条件についても考察した。

キーワード 経路制御, セキュリティメトリック, ポリシールーティング。

A Proposal of Routing Algorithm Dealing with Multiple Metrics and A Study on the Precondition

Atsufumi MORIYAMA[†], Hiroshi ISHINISHI[†], Katsuichi NAKAMURA[†], and Yoshiaki HORI^{††}

[†] Network Application Engineering Laboratories LTD.

JPR Hakata Bldg. 6F, Hakata-ekimae 1-4-4, Hakata-ku, Fukuoka-shi, Fukuoka, 812-0011 Japan.

^{††} Kyushu University

Ooaza-Motooka 744, Nishi-ku, Fukuoka-shi, Fukuoka, 819-0395 Japan.

E-mail: †{moriyama,ishinishi,nakamura}@nalab.jp, ††hori@inf.kyushu-u.ac.jp

Abstract In routing, we usually use OSPF with Dijkstra or RIP with Bellman-Ford, but we cannot deal with multiple metrics with them. With multiple metrics, we would use a weighted average of the metrics or techniques from operations research, but they are not suitable for routing because they are lacking in validity and simpleness. In this paper, we propose a routing algorithm, with an example, giving a routing policy, to deal with the three security metrics proposed by I. A. Almerhag and M. E. Woodward. Besides, we make a study on the constraints of the metrics and the policy as the preconditions of the routing algorithm.

Key words routing, security metric, policy routing.

1. はじめに

今日のインターネットにおける経路制御は、ダイクストラ法を使った OSPF やベルマンフォード法を使った RIP がよく使われているが、その何れも複数のメトリックに対応していない。

今後のネットワークは、従来のメトリックであるホップ数や通信速度などに加えて、セキュリティ強度や電力消費量などの新しいメトリックも含めて、様々な性質を複合的に考慮した経路制御をすることになると思われる。

具体的に、著者らが考えている応用事例として、スマートグリッドを構成するセンサーノードがスマートメーターの自動検針データを中継する場合を考える。今日のインターネットでは、データ早期到達性に重点を置くため、図 1 のように限られた電力量しか保持していないセンサーノードが中継ポイントとして存在してもデータを送付してしまう。又、検針データという個人情報を含むようなデータをセキュアでないセンサーノードに渡すことで個人情報が盗まれる恐れがあってもデータを送付してしまう。この場合、経路選択としては電力量がより多く残っ

ていて、かつ認証機構によりセキュアと認定されたセンサーノードを経由するようにならなければならない。又、更なる応用として、中継ノードがネットワークコーディングなどを利用して、受信したデータをマージして同報することなどによって、電力消費を抑えることも考えられるので、それが可能となるように経路制御を行なうことも考えられる。

が見られる。これらの手法は、重みづけの妥当性を付与したものは評価できるが、内部で重みの決定や解の収束など時間のかかる複雑な計算が用いられているため、逆にアルゴリズムの簡便性が損なわれ、経路制御には不適切である。

そこで、本研究では、重みづけの妥当性の問題を克服するため、加重平均のような恣意的な重みづけをせずに、経路の優劣比較方法、つまりルーティングポリシーを自然にかつ柔軟に設定できるように、又、ダイクストラ法やベルマンフォード法などのアルゴリズムに準じた簡便性をもった、経路制御のアルゴリズムを提案する。

そして、提案アルゴリズムの動作例として、従来の経路制御のメトリックに、I. A. Almerhag と M. E. Woodward 両氏が [3] で提案したセキュリティメトリック 3 種類を付加して、ルーティングポリシーを適宜に設定して、具体的に示す。

更に、今後のために、メトリックやルーティングポリシーが充足すべき条件について考察して、別のメトリックとルーティングポリシーに対しても動作できるよう方向性を与える。

以下、第 2 節で従来の経路制御と第 3 節で先行研究を簡単に紹介した上で、第 4 節でそれらの問題点を述べる。第 5 節で提案アルゴリズムの説明をして、最後にまとめと今後の課題を述べる。

2. 従来の経路制御

一般的に、経路制御問題とは、ネットワークの一ノードから他ノードまでの、何らかの条件、又はポリシーを満たす経路を決定する問題である。

従来の経路制御問題は、各リンクにメトリックとして単一数値を与えて、パス上の各リンクのメトリック和を最小にする最短経路を決定する問題である。

このような従来の経路制御問題に対する経路制御方法として、提案アルゴリズムと深く関連する、ダイクストラ法が使われている OSPF とベルマンフォード法が使われている RIP を紹介する。

2.1 OSPF(ダイクストラ法)

OSPF は、小規模から大規模までのネットワーク向けのリンクステート型の経路制御手法である。OSPF では、ネットワーク規模が大きくなると、スケーラビリティを保つために、エリアという区域に分割して経路制御する。

OSPF を使っているネットワークにおける各ノードは、隣接ノードとのリンク状態をフラッディングを用いて情報交換し、受け取ったリンク状態を収集し、ネットワークトポロジーを把握した上で、ダイクストラ法による最小コストパスの算出をすると共にルーティングテーブルへその結果を反映する。なお、ダイクストラ法ではメトリックを慣習的にコストと呼び、その最小化を考えるので、ここもそれに従う。

ダイクストラ法のアルゴリズムは以下の通りである。

(0) 各ノードは、それまでに判明した最短経路長を保持する。最初に、自分までの最短経路長は 0 で自明なので確定し、他ノードまでの最短経路長は不明なので ∞ に設定する。

(1) 全ノードが確定されるまでループする。

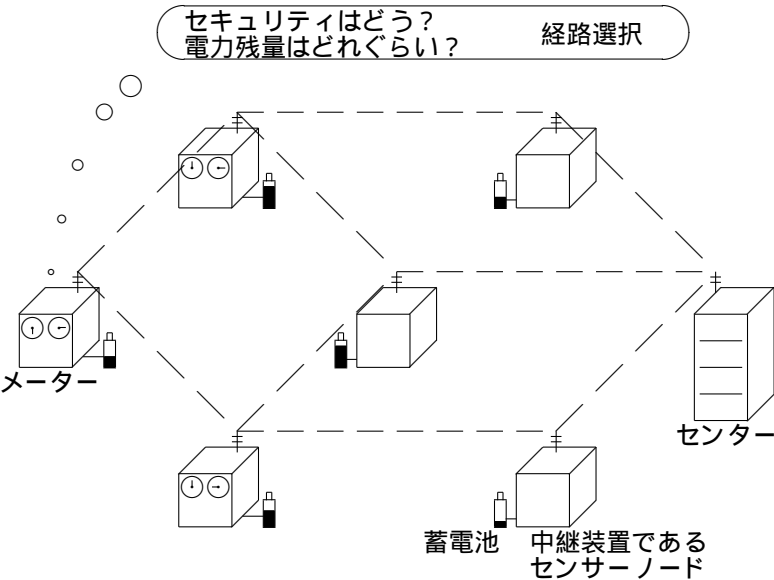


図 1 想定している応用事例

このように、従来の経路制御で扱っていた単一数値のメトリックに加えて、非数値など新型のメトリックも含めた、複数メトリックに対応した経路制御は、今後のネットワークにおいて、非常に重要となることが予想され、これに対して、様々な手法が研究されている。

一般的に採られる方法としては、非数値のメトリックはファジー理論などで数値化して、複数のメトリックを加重平均して、強制的に単一数値のメトリックにすることで、従来の経路制御方法へ帰着させる手法がよく用いられる。この手法は簡便だが、例えば、上述の応用事例において、電力残量よりもセキュリティ強度を重視したい場合に電力残量に 0.2 の重みとセキュリティ強度に 0.8 の重みなど、理論的な裏づけがないまま、重要度を高めたい項目に適当に大きな重みを付与することになる。しかし、実際、ユーザが設定したい重みは、それほど大きくなかったり、もっと大きくしたかったり、又は、経路に含まれるノードは電池切れ寸前であるが、セキュリティが非常に強固なので、どうしても利用したいなどと、状況に応じて臨機応変的に変動したりするかもしれない。このように、重みづけの理論的な裏づけがないと、ユーザが希望する経路と異なる経路になってしまい、重みづけの妥当性が危惧される。

一方、近年の研究では、費用便益分析や階層分析法 [1] など意思決定問題に対するオペレーションズリサーチの手法や、最新の動向としては、遺伝的アルゴリズムによる解空間の多点探索で多目的最適化問題に対するパレート最適解を求める手法 [2]

i. 新たに確定したノードからリンク1本でたどりつける隣接ノードまでの最短経路長を更新して、その隣接ノードが存在しなければ確定候補ノード集合へ追加する。

ii. 確定候補ノード集合にあるノードのうち、最短経路長が最短となっているノードを確定する。最短となっているノードが複数の場合、その何れを確定してもよい。

(2) 各ノードまでの最短経路長とその経路が求まったら、ルーティングテーブルへ結果を反映する。

なお、ダイクストラ法は、最も忠実な実装で計算量はノード数の2乗オーダーとなる。

又、ダイクストラ法が動作する条件としては、リンクコストは必ず正値でなければならない。これは、負のコストをもったリンクが存在して、上記のアルゴリズムで最短経路が見つからなくなる可能性があるのと、負のコストをもったループが存在して、そこを廻って最短経路長をいくらでも小さくできる可能性があるのを、防ぐためである。

2.2 RIP(ベルマンフォード法)

RIPはディスタンスベクター型の経路制御方法である。RIPを使っているネットワークにおける各ノードは、他のノードまでのホップ数をメトリックとして、隣接ノードと経路情報を動的に交換することで、最短経路を決定する。その根底にあるのがベルマンフォード法で、RIPはその分散版に相当する。

ベルマンフォード法は、ダイクストラ法と同様、最短経路を求めるアルゴリズムだが、負のコストをもったリンクにも対応し、負のコストをもったループを検出できる。その代わりに、ダイクストラ法よりも実行時間がかかり、計算量はノード数とリンク数をかけたオーダーとなる。ベルマンフォード法は、スケラビリティがよくないことと、ネットワーク構成の変更を反映するまで時間がかかることで、主に小規模のネットワークでしか使わない。

3. 先行研究

従来の経路制御問題では、メトリックが単一数値であったのに対して、そのような単一数値メトリックに加えて、非数値など新型のメトリックも含めた、複数メトリックに対応した経路制御問題を考える。

このような経路制御問題に対して、様々な手法が研究されている。一般的に考えられるのは加重平均である。近年、意思決定問題に対するオペレーションズリサーチの手法が研究されていて、これを流用すれば、可能な全経路より最善と思われる経路を一つ決定できる。別途、I. A. Almerhag と M. E. Woodward 両氏は [3] でセキュリティメトリック3種類を提案したが、その具体的なシステムの動作までは言及していない。これを受けて、D. Han らは [2] でこのセキュリティメトリック3種類を含む複数メトリックを考慮した、NP 困難である多目的最適化問題に対して、遺伝的アルゴリズムの複数個体発生による解空間の多点探索でパレート最適解を求める手法を提案している。

3.1 加重平均値による方法

一般的に採られる方法である加重平均などを用いた手法では、非数値のメトリックなどがあれば、ファジー理論などの手法を

用いて、非数値メトリックを数値化し、複数のメトリックを加重平均することで、従来の経路制御方法へ帰着させている。

確かに従来の経路制御方法へ帰着させるのは非常に簡便な方法ではある。しかし、そもそもメトリックは、具体的な基準がないまま、OSPF や RIP などの preference で適当に1とか100とかの数値を設定しているような、粗い精度をもったものであるのか。又、加重平均の際にも、理論的な裏づけがないまま、重要度を高めたい項目に適当に0.6や0.8など大きな重みをつけているような、恣意的にならざるをえない重みづけの仕方では、果たしてユーザの好みを反映した設定になるのか。このように様々な問題点があるので、その妥当性は疑わしい。

3.2 近年の研究

近年、意思決定問題に対するオペレーションズリサーチの手法が多く研究されている。

費用便益分析は、システムを導入すべきかを検討する際、又は幾つかのシステムを比較する際に、システムの全リソースと、そのもたらす効果を、すべて金額へ換算して評価する手法である。費用には、人件費や原材料費など実際にかかる実際費用と、そのシステムを選択しなかったために見逃された機会に相当する価値である機会費用がある。便益には、直接的な一次的な直接便益と、附随的な二次的な間接便益がある。その評価基準として、費用一定の条件下で便益の最大化と、便益一定の条件下で費用の最小化と、便益対費用の比率の最大化がある。又、拡張として、費用有効度分析という手法もあり、システムの効果を金額で評価できない場合に、便益に代わりて目的の達成度を有効度という数値で評価したものである。

階層分析法 [1] は、幾つかの代替案を比較して選択する際に、各評価項目の重要度を決定した上で、各代替案の優先度を決定して、最優先のものを選択する手法である。まず、評価項目を二個ずつ対にして重要度を相対的に比較する一対比較法を用いて、相対的な重要度の間の整合性も玩味しながら、重要度を決定する。そして、代替案も一対比較する相対評価法と代替案を一対比較しない絶対評価法があるが、代替案は、各評価項目の値を重要度で加重平均して、総合的な優先度が算出される。こちらも様々な拡張が考えられている。

これらは何れも、複数パラメータを考慮して複数選択肢より取捨選択して一つ決定する方法である。これを流用すれば、可能な全経路より最善と思われる経路を一つ決定することが可能ではあるが、重みや総合評価を決定するまでの計算は非常に複雑である。

3.3 セキュリティメトリック3種類

I. A. Almerhag と M. E. Woodward 両氏は [3] でセキュリティメトリック3種類を提案している。この後、これを使って提案アルゴリズムの具体例を示すので、ここで説明しておく。

両氏は、通信データのセキュリティレベルを計測する標準的な手法が確立されていないゆえ、経路制御用のセキュリティメトリックも定義されないと指摘した。そもそもメトリックは明確かつ具体的で、計測可能、入手可能、再現可能で、時刻に依存すべきである。そのため、実際、相互接続網において、防御の

第一線を形成する隣接ルータ間の認証と、ネットワークのデータ通信における機密性を保証する暗号化方法と、情報システムにおけるサービスの利用可能性を保障するアクセス制御システムの、セキュリティの三つの目的における達成度を管理できる、合理的で実用的なセキュリティメトリック 3 種類を定義できる。その際、利用可能なパスより選択したパスのメトリックが最良、つまり脆弱度が最小でセキュリティレベルが最大となるようにした。なお、これは、非数値で扱いにくかったセキュリティメトリックを、ファジー理論などに依らない方法で数値化できた一例である。

以下、単一リンクにおける各メトリック $M^{[*]}$ の定義と、リンク e_1, e_2, e_3, \dots からなるパス p における各メトリック $M_p^{[*]}$ の、各リンクにおけるそのメトリック $M_{e_i}^{[*]}$ による計算方法を示す。

(1) 隣接ルータ間の認証における暗号化の有無 $M^{[1]}$
このメトリックは真理値で、隣接ルータ間の認証における鍵交換にセキュアな MD5 を使っていれば真つまり 1 とし、それ以外は偽つまり 0 とする。パスにおけるメトリックに関しては、途中の一箇所でも鍵交換にセキュアな MD5 を使っていなければ、パス全体で使っていないのと同じになるので、論理積を取る形となる。つまり、 $M_p^{[1]} = \text{AND}_i (M_{e_i}^{[1]})$ で計算される。これをバイナリ合成ルールと呼ぶ。

(2) データ送受信におけるセキュリティ強度 $M^{[2]}$
このメトリックは 0~1 間の実数値で、データ通信における暗号化方法の、暗号のキー長が十分に長く、今後 30 年間解読されないなら 1 とし、暗号のキー長が推奨サイズよりも短いなら 0 とし、その中間の場合は、暗号が見破られる年数 Y に応じて $0.01 + 0.033 \times Y$ とする。パスにおけるメトリックに関しては、途中のセキュリティが最も弱い箇所が全体に影響する基準となるので、最小値を取る形となる。つまり、 $M_p^{[2]} = \text{MIN}_i (M_{e_i}^{[2]})$ で計算される。これを凹型合成ルールと呼ぶ。

(3) アクセス制御におけるトラフィックフィルタリング技術 $M^{[3]}$
このメトリックも 0~1 間の実数値で、あるノードを通過して、その出リンクにおける、攻撃や侵入をファイアウォールが防御するか IDS/IPS が検出するかの確率で定義する。パスにおけるメトリックに関しては、途中の何れかの箇所が防御が検出されればよいので、パス全体で攻撃や侵入を防御検出できない確率は、途中の各リンクの防御検出できない確率の積となるので、パス全体の攻撃や侵入を防御検出する確率は、途中の各リンクの防御検出確率を用いて $M_p^{[3]} = 1 - \prod_i (1 - M_{e_i}^{[3]})$ で計算される。これを乗算的合成ルールと呼ぶ。

3.4 最新の動向

最新の動向として、以下のような研究がある。

D. Han らは [2] で、このセキュリティメトリック 3 種類も含めて、複数メトリックを考慮した NP 困難である多目的最適化問題に対して、遺伝的アルゴリズムの複数個体発生による解空間の多点探索でパレート最適解を求める手法を提案している。

最小化する他のメトリックに合わせて、最大化すべきセキュリティメトリック 3 種類を以下のように定義しなおしている。

- 隣接ルータ間の認証における暗号化方式を示すメトリックは、真理値ではなく、鍵交換にセキュアな MD5 を使っていれば 4, AES なら 5, SHA なら 6, RSA なら 7, DES なら 8 というように数値を設定する。又、パスにおけるメトリックは、各リンクにおけるメトリックの加算和として定義する。

- アクセス制御におけるトラフィックフィルタリング技術を示すメトリックは、1 より減じた形で考える。更に、乗算の形は、パスにおけるメトリックを計算する上で不便を来たすので、メトリックは対数を取って符号を反転した形で定義する。

- データ送受信におけるセキュリティ強度を示すメトリックも、1 より減じた形で考える。又、パスにおけるメトリックは、各リンクにおけるメトリックの最大値として定義する。

この手法で多目的最適化問題を解く際、解空間に複数の点を配置し、遺伝的アルゴリズムを用いることで、評価を改善する方向を探索・改善を繰り返すことで最終的にパレート最適解に落ち着く。このようにすれば、絡み合う条件下で同時に複数の最適と思われる解を算出できる。この手法を利用すれば、可能な全経路より最適と思われる経路の複数候補を見出すことが可能ではあるが、パレート最適解に収束するまで時間がかかってしまい、実用上、問題がある。

このように、第 3.2 節や第 3.4 節の手法を用いれば経路選択は可能ではあるが、その何れの手法も複雑で、時間がかかってしまうため、実際のネットワークでの経路制御には不適切である。

4. 問題提起

従来の経路制御問題に対しては、ダイクストラ法を使った OSPF とベルマンフォード法を使った RIP があるが、何れも単一数値のメトリックにしか対応していない。又、非数値の新型メトリックも含む複数メトリックに対応した経路制御問題に対して、解法に繋がる様々な手法が提案されているが、何れも経路制御には不適切だと考える。例えば、現在も具体的な基準がないまま適当な数値を設定しているメトリックを、ユーザの好みを適切に反映しているか疑わしい恣意的な重みづけで加重平均をするのは、妥当性に欠ける。又、意思決定問題に対する費用便益分析や階層分析法や、多目的最適化に対する遺伝的アルゴリズムでパレート最適解を求める手法は、経路制御への適用も考えられるが、アルゴリズムの簡便性が損なわれてしまい、現実的な経路制御には使えない。

本研究では、重みづけの妥当性の問題を克服するため、加重平均のような恣意的な重みづけをせずに、経路の優劣比較方法、つまりルーティングポリシーを自然でかつ柔軟に設定できるように、又、ダイクストラ法やベルマンフォード法などのアルゴリズムに準じた簡便性をもった、非数値など新型のメトリックも含む、複数メトリックに対応した経路制御方法を考える。

そして、提案アルゴリズムの動作例として、I. A. Almerhag と M. E. Woodward 両氏が [3] で提案したセキュリティメトリック 3 種類を使った、具体的なシステムの動作を示す。

従来の経路制御問題と比較して、以下の箇所が変化している。

- 従来のメトリックは最小化であったのに対して、セキュ

リティメトリック 3 種類は最大化である。

- 各リンクにおけるメトリックは、距離やホップ数、又は様々な条件を考慮して設定すると思われる適当な単一数値であったのに対して、数値の組、又は非数値を含む場合もあるので、メトリック値の組となる。

- パスにおけるメトリックは、パスに含まれる各リンクにおけるメトリックの加算和であったのに対して、そうでない場合も出てくる。実際、セキュリティメトリック 3 種類とも加算和でなく、各メトリックの集計方法に従って算出されたパスにおけるメトリック値の組となる。

- 経路の優劣判定は、単一数値の大小比較であったのに対して、メトリック値の組の比較方法を別途設定する必要がある。このメトリック値の組の比較方法、つまり経路の優劣判定方法は、経路に対する好みが含まれているので、これをルーティングポリシーと呼ぶ。

5. 提案アルゴリズム

提案アルゴリズムは、ダイキストラ法を、簡便性を損なわずに以下のように拡張して、複数メトリックの経路制御問題に対して最良経路を算出できるようにした。

- 複数メトリックをそのままメトリック値の組として扱う。パスにおけるメトリックは、そのパス上の各リンクにおけるメトリックを、各メトリックの集計方法に従ってそれぞれ集計する。

- 経路選択においては、予め決定しておいたルーティングポリシーに従って、優劣判定をする。

しかし、このままでは、各ノードは隣接ノードを認証などで知るが、ネットワークポロジ全体は知りえない。もし OSPF のようにフラディングで全ノードがネットワークポロジを把握するようにすると、単純に見積って、通常のダイキストラ法の、メトリック種類だけの倍数のトラフィックが発生するので、問題である。そこで、解決策として、ベルマンフォード法の分散版である RIP を見倣って、この提案アルゴリズムを、各ノードが情報交換しながら分散並列的に計算する形にすることを考えた。

以下、提案アルゴリズムの詳細を、具体的なシステム動作例と共に、示す。

5.1 前提条件

提案アルゴリズムが前提としている入力などの条件で必要なのは、以下の二つである。

- メトリックとその集計方法

ダイキストラ法などが適用される従来のメトリックは通信可能性を保証するものなので、ここではそれに [3] のセキュリティメトリック 3 種類を追加した具体例で示すが、一般的に、メトリックとその集計方法は第 5.4 節で述べる動作条件を満たす必要がある。

- ルーティングポリシー

経路の優劣比較の方法として、ここでは、下記のルーティングポリシーを用いた具体例で示す。その他には、ルーティングポリシーがメトリックの単純な数式で表現できる場合は勿論のこ

と、複数の経路を選択して同時に保持するようなポリシーや、好みを適切に反映した、一対比較などを用いた複雑なポリシー、又は、辞書式順序など数式で表現しづらいポリシーを設定することもできる。しかし、一般的に、ルーティングポリシーは第 5.4 節で述べる動作条件を満たす必要がある。

ここで具体例として挙げるルーティングポリシーは、以下のようにより、経路を 2 ステップで考慮して選択するようなものである。

(1) ルータ間の認証がセキュアであるリンクを優先して比較して利用する。

具体的には $M^{[1]} = 1$ となっているリンクがあれば、 $M^{[1]} = 0$ となっているリンクをすべて除外して考慮する。

(2) 従来のメトリック $M^{[0]}$ に見合ったセキュリティをもったリンクを優先する。

具体的には $(M^{[2]} + M^{[3]}) / M^{[0]}$ が最大となるリンクを選択する。

5.2 アルゴリズムの動作

提案アルゴリズムの動作は以下の通りで、具体的な動作例を、第 3.3 節で述べたメトリックと、前節で述べたルーティングポリシーを用いて図 2 に示す。なお、現時点では、各ノードがネットワークポロジ全体や各リンクにおけるメトリックを全て把握しているものとする。

(0) 各ノードは、判明済みの最良経路のメトリックを保持する。複数の最良経路を求める場合は複数スロットを用意する。最初に、自分までの最良経路は自明なので確定する。メトリックはリンクに対して付与されているので特に保持する必要はない。保持しようとするなら、各メトリックの集計演算の単位元を設定する。他ノードまでの最良経路は不明と設定する。

(1) 全ノードが確定されるまでループする。

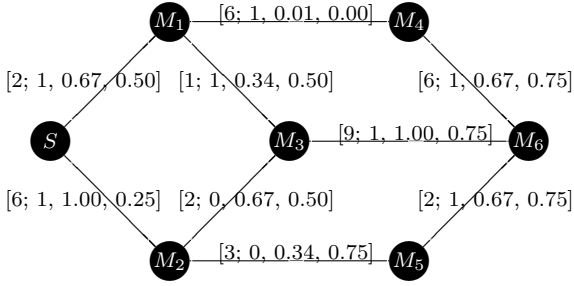
i. 新たに確定したノードからリンク 1 本でたどりつける隣接ノードまでのメトリックをそれぞれの集計方法で集計して、メトリック値の組として、ルーティングポリシーでより良い評価になるようなら更新して、その隣接ノードを確定候補ノード集合に、存在しなければ、追加する。

ii. 確定候補ノード集合にあるノードのうち、ルーティングポリシーに従って、メトリック値の組が最良となっている経路をもつノードを確定する。ルーティングポリシーで同一評価となるノードが複数の場合、その何れを確定してもよい。

(2) 各ノードまでの最良経路とそのメトリックが求まったことになるので、ルーティングテーブルへ反映する。

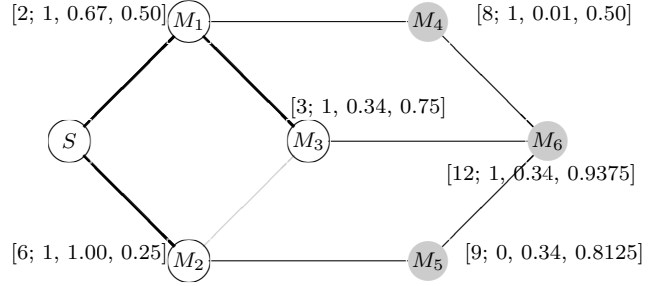
各ノードが、ネットワークポロジ全体や各リンクにおけるメトリックをすべて把握しているものと仮定していたが、実際は、その情報を交換したりして収集に取りかからなければいけない。しかし、フラディングで全ノードがネットワークポロジ全体を把握しようとする、単純に見積って、メトリック種類だけの倍数のトラフィックが発生する。そこで、ベルマンフォードとその分散版である RIP を参考として、グラフの接続行列から到達可能頂点を算出していく手法を用いた。具体的には、以下のように各ノードが情報交換しながら分散並列的に計算していくアルゴリズムとなる。

このネットワークポロジにおいて、
ノード S から他ノードまでの最良経路を算出したい。



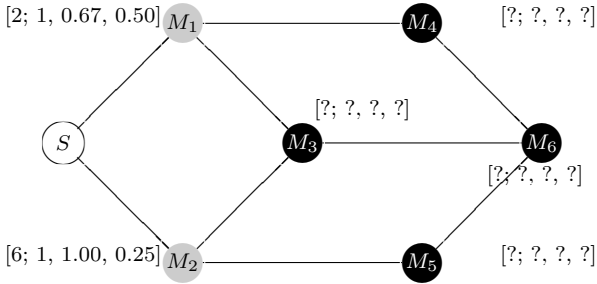
(0) 問題設定

メトリック最良の M_2 を確定して、 M_5 までのメトリックを更新する。



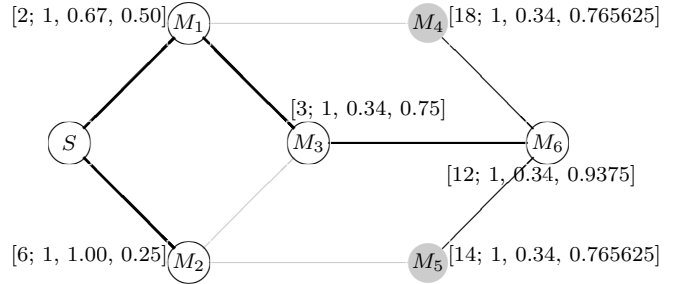
(4) 第 4 ステップ

自分を確定して、隣接ノードまでのメトリックを更新する。
他のノードまでのメトリックは不明のまま。



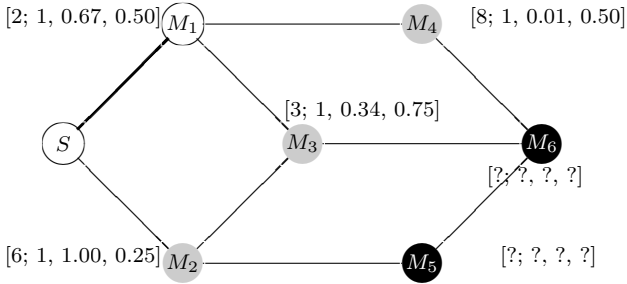
(1) 第 1 ステップ

メトリック最良の M_6 を確定して、 M_4 と M_5 までのメトリックを更新する。



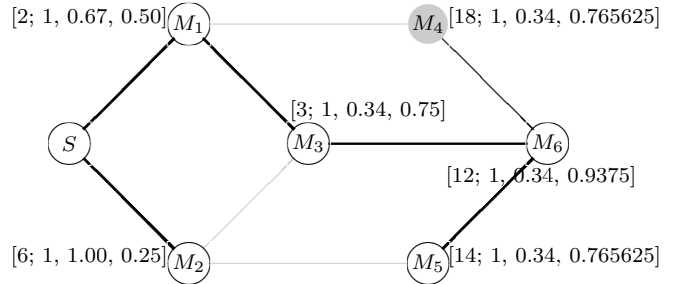
(5) 第 5 ステップ

$[2; 1, 0.67, 0.50] > [6; 1, 1.00, 0.25]$ なので M_1 を確定して、
 M_3 と M_4 までのメトリックを更新する。



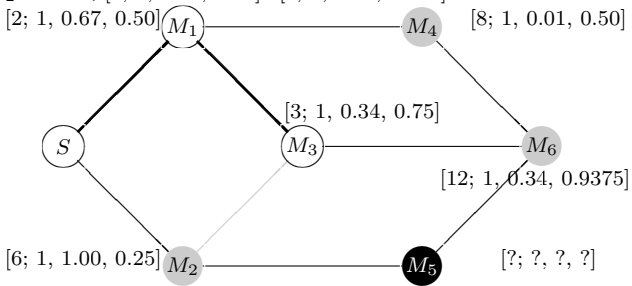
(2) 第 2 ステップ

$[14; 1, 0.34, 0.765625] > [18; 1, 0.34, 0.765625]$ なので M_5 を
先に確定するものの、もう更新すべきメトリックはない。



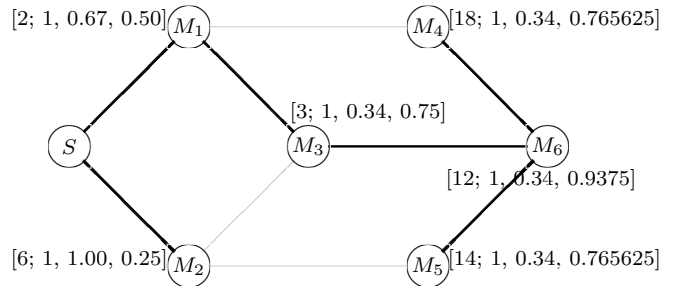
(6) 第 6 ステップ

メトリック最良の M_3 を確定して、 M_6 までのメトリックを更新する。
 M_2 までは、 $[6; 1, 1.00, 0.25] > [5; 0, 0.34, 0.875]$ なので更新しない。



(3) 第 3 ステップ

最後に M_4 を確定して、全ノードが確定できたので、
最良経路とメトリックをルーティングテーブルへ反映する。



(7) 第 7 ステップ

図 2 提案アルゴリズムの動作

提案アルゴリズムの分散並列版の動作は以下の通りで、具体的な動作例を図3に示した。

(0) 設定

各ノードは、判明した最良経路、実はネクストホップのみでよいのでネクストホップと、そのメトリック値の組を、ルーティングテーブルに保持する。

(1) 接続行列の構築

最初は、各ノードは、自分から隣接ノードまでのメトリック値の組のみを設定する。ネットワークポロジ全体で眺めると、これは各ノードが隣接ノードまでしか見えていない状態に相当する。

(2) ループ

以下の処理を概して高々ノード数だけ繰り返す。非同期の分散並列計算なので、隣接ノードのルーティングテーブルの更新が済んでいれば、その分だけ収束が早まったり、逆に、ネットワークポロジに変動があった場合であれば、収束が遅くなったりすることがある。又、収束後も変動があった場合は、再び以下の処理を繰り返して、最良経路を見出す。

i. 情報交換と分散並列計算

隣接ノードのみと情報交換して、隣接ノード経由で他ノードへ行く別のパスが存在するなら、比較して、より良いほうを利用するようにルーティングテーブルを更新する。これは、接続行列の冪乗を計算して、隣接ノードを経由して到達可能ノードまでの経路情報を算出することに相当する。

(3) 各ノードでのルーティングテーブルには最良経路、又はネクストホップと、その経路を使った場合のメトリック値の組が求まっている。

ダイクストラ法と、それを拡張した提案アルゴリズムの相違点、或いは、ベルマンフォード法、又はその分散並列版であるRIPと、この提案アルゴリズムの分散並列版の相違点も実は同じだが、これを以下に示す。

- メトリックを、単一数値から、非数値も含めて複数個に拡張した。

- メトリックの集計演算は、加算から、一般的な集計演算に拡張した。各メトリックはそれぞれの集計演算に従って集計される。

- 経路の優劣比較は、単一数値の大小比較から、一般的な優劣比較に拡張した。メトリック値の組をもって、ルーティングポリシーを反映した比較方法が定められる。

これ以外はほぼ全く同様の動作をする。

5.3 利点と欠点

提案アルゴリズムを用いた場合の、主だった利点と欠点を以下に示す。

● 利点

- メトリックは、複数個を自由に扱えて、同じ枠組みで動作させることができる。

- 各メトリックの集計方法は、一般的な集計関数を用いる場合、その定義域や値域に注意する必要があるが、第5.4節の条件を満たしていれば、自由に設定できる。

- メトリック値の組に対する優劣比較方法であるルーティ

ングポリシーも、第5.4節の条件を満たしていれば、自由に設定できる。

● 欠点

- メトリックの種類数にも依るが、ネットワーク規模が増大すると、OSPFのエリアよりも細かい単位に区切って処理する必要がある。但し、これは各ノードでの処理を分散並列化することで幾分か解決できている。

- ルーティングポリシーとして一対比較など複雑な方法を組み込んだ場合、計算量や処理時間がかかってしまう。これはルーティングポリシー、つまり各回の経路優劣比較の方法に依存する。

- メトリックやルーティングポリシーは、後述の制約条件を満たすものでなければ、提案アルゴリズム自体が動作しない可能性がある。

5.4 動作条件

提案アルゴリズムが動作するための条件を考察して、パスにおけるメトリックの計算と、その比較であるルーティングポリシーは、以下の条件を満たす必要があることがわかった。

● 一意性

パスにおけるメトリックの値がその計算順序に依存せず、一意的であるためには、各メトリック M に対して、その集計方法 \oplus が結合則 $(M_{A \rightarrow B} \oplus M_{B \rightarrow C}) \oplus M_{C \rightarrow D} = M_{A \rightarrow B} \oplus (M_{B \rightarrow C} \oplus M_{C \rightarrow D})$ を満たさなければいけない。これを満たす場合、単に $M_{A \rightarrow B} \oplus M_{B \rightarrow C} \oplus M_{C \rightarrow D}$ のように、括弧を外して書くことができるようになる。

● 単調性

他の経路に比べて優れている最良経路は、他と同じリンクやパスを前後へ追加しても、他に比べてやはり優れている、要は、経路の優劣順序が、その前後への経路追加に対して、保存される、つまり、パスにおけるメトリックの値が単調的であるためには、他の経路を p' 、最良経路を p 、経路が優れていることを示す n 個のメトリック値の組 $\vec{M}_* = (M_*^{[1]}, M_*^{[2]}, \dots, M_*^{[n]})$ の関係を \succ で表わすこととして、 p と同じ始点と終点をもつパス p' や、その前後それぞれに追加する経路 a と b に対して、 $\forall p'; \vec{M}_p \succ \vec{M}_{p'} \Rightarrow \forall a, b; \vec{M}_{a \cdot p \cdot b} \succ \vec{M}_{a \cdot p' \cdot b}$ が成立しなければいけない。又、便宜的に各メトリックの集計演算を組にして \oplus と表記すると、 $\vec{M}_{a \cdot p \cdot b} = \vec{M}_a \oplus \vec{M}_p \oplus \vec{M}_b$ となるので、各メトリックの集計演算に逆演算が存在しているなら、左から \vec{M}_a 、右から \vec{M}_b 、演算 $\overleftarrow{\oplus}$ を施すことによって、この条件は、優劣関係 \succ が演算 \oplus やその逆演算 $\overleftarrow{\oplus}$ に対して順序を保つ、つまり順序が逆転しないと言い換えることができる。

5.5 動作条件の検証

具体例として、ダイクストラ法の場合と提案アルゴリズムの場合をそれぞれ確認する。

ダイクストラ法では、従来のメトリックは、加算の結合則によって、その一意性が保証され、加算やその逆演算である減算が、優劣比較としての大小比較の向きを変えないことによって、その単調性が保証されていることが、すぐに確認できる。

提案アルゴリズムでは、従来のメトリックは同じく、加算の結合則によって、その一意性が保証される。他のメトリックは

同じ経路制御問題を考える． S が隣接ノードと情報交換してルーティングテーブルを初期化する．

S の経路表		
行き先	メトリック	経由
S	[ループバック]	S
M_1	[2; 1, 0.67, 0.50]	M_1
M_2	[6; 1, 1.00, 0.25]	M_2
M_3	[?; ?, ?, ?]	?
M_4	[?; ?, ?, ?]	?
M_5	[?; ?, ?, ?]	?
M_6	[?; ?, ?, ?]	?

(1) 第 1 ステップ

この時点で，各ノードが保持するルーティングテーブルは接続行列の各列に相当する．
因みに，無向グラフなので接続行列は対称行列となる．

から まで	S	M_1	M_2	M_3	M_4	M_5	M_6
S	[ループバック]	[2; 1, 0.67, 0.50]	[6; 1, 1.00, 0.25]	[?; ?, ?, ?]	[?; ?, ?, ?]	[?; ?, ?, ?]	[?; ?, ?, ?]
M_1	[2; 1, 0.67, 0.50]	[ループバック]	[?; ?, ?, ?]	[1; 1, 0.34, 0.50]	[6; 1, 0.01, 0.00]	[?; ?, ?, ?]	[?; ?, ?, ?]
M_2	[6; 1, 1.00, 0.25]	[?; ?, ?, ?]	[ループバック]	[2; 0, 0.67, 0.50]	[?; ?, ?, ?]	[3; 0, 0.34, 0.75]	[?; ?, ?, ?]
M_3	[?; ?, ?, ?]	[1; 1, 0.34, 0.50]	[2; 0, 0.67, 0.50]	[ループバック]	[?; ?, ?, ?]	[?; ?, ?, ?]	[9; 1, 1.00, 0.75]
M_4	[?; ?, ?, ?]	[6; 1, 0.01, 0.00]	[?; ?, ?, ?]	[?; ?, ?, ?]	[ループバック]	[?; ?, ?, ?]	[6; 1, 0.67, 0.75]
M_5	[?; ?, ?, ?]	[?; ?, ?, ?]	[3; 0, 0.34, 0.75]	[?; ?, ?, ?]	[?; ?, ?, ?]	[ループバック]	[2; 1, 0.67, 0.75]
M_6	[?; ?, ?, ?]	[?; ?, ?, ?]	[?; ?, ?, ?]	[9; 1, 1.00, 0.75]	[6; 1, 0.67, 0.75]	[2; 1, 0.67, 0.75]	[ループバック]

(0) 接続行列

S が M_1 より M_3 と M_4 までの情報を入手してルーティングテーブルを更新する．

S の経路表			M_1 の経路表			S の経路表		
行き先	メトリック	経由	行き先	メトリック	経由	行き先	メトリック	経由
S	[ループバック]	S	S	[2; 1, 0.67, 0.50]	S	S	[ループバック]	S
M_1	[2; 1, 0.67, 0.50]	M_1	M_1	[ループバック]	M_1	M_1	[2; 1, 0.67, 0.50]	M_1
M_2	[6; 1, 1.00, 0.25]	M_2	M_2	[?; ?, ?, ?]	?	M_2	[6; 1, 1.00, 0.25]	M_2
M_3	[?; ?, ?, ?]	?	\oplus M_3	[1; 1, 0.34, 0.50]	M_3	\Rightarrow M_3	[3; 1, 0.34, 0.75]	M_1
M_4	[?; ?, ?, ?]	?	M_4	[6; 1, 0.01, 0.00]	M_4	M_4	[8; 1, 0.01, 0.50]	M_1
M_5	[?; ?, ?, ?]	?	M_5	[?; ?, ?, ?]	?	M_5	[?; ?, ?, ?]	?
M_6	[?; ?, ?, ?]	?	M_6	[?; ?, ?, ?]	?	M_6	[?; ?, ?, ?]	?

(2) 第 2 ステップの 1

同様に M_2 より M_3 と M_5 までの情報を入手してルーティングテーブルを更新する．

S の経路表			M_2 の経路表			S の経路表		
行き先	メトリック	経由	行き先	メトリック	経由	行き先	メトリック	経由
S	[ループバック]	S	S	[6; 1, 1.00, 0.25]	S	S	[ループバック]	S
M_1	[2; 1, 0.67, 0.50]	M_1	M_1	[?; ?, ?, ?]	?	M_1	[2; 1, 0.67, 0.50]	M_1
M_2	[6; 1, 1.00, 0.25]	M_2	M_2	[ループバック]	M_2	M_2	[6; 1, 1.00, 0.25]	M_2
M_3	[3; 1, 0.34, 0.75]	M_1	\oplus M_3	[2; 0, 0.67, 0.50]	M_3	\Rightarrow M_3	[3; 1, 0.34, 0.75]	M_1
M_4	[8; 1, 0.01, 0.50]	M_1	M_4	[?; ?, ?, ?]	?	M_4	[8; 1, 0.01, 0.50]	M_1
M_5	[?; ?, ?, ?]	?	M_5	[3; 0, 0.34, 0.75]	M_5	M_5	[?; ?, ?, ?]	?
M_6	[?; ?, ?, ?]	?	M_6	[?; ?, ?, ?]	?	M_6	[?; ?, ?, ?]	?

(3) 第 2 ステップの 2

同様に M_1 と M_2 より経路情報を入手してルーティングテーブルを更新しつづける．

その時点では最良経路とは限らないが，取りあえず経路の一つとして使えて，更新していくうちに最良経路となる．

(4) 第 3 ステップ以降

図 3 分散並列版提案アルゴリズムの動作

以下のように、同様に一意性が確認できる。

$$\begin{aligned}
 (1) \quad & M^{[1]} \text{ に関しては AND の結合則によって保証される。} \\
 (2) \quad & M^{[2]} \text{ に関しては MIN の結合則によって保証される。} \\
 (3) \quad & M^{[3]} \text{ は以下のように、乗算の結合則に帰着できる。} \\
 & (M_{A \rightarrow B}^{[3]} \oplus^{[3]} M_{B \rightarrow C}^{[3]}) \oplus^{[3]} M_{C \rightarrow D}^{[3]} \\
 & = \left(1 - \left(1 - M_{A \rightarrow B}^{[3]}\right) \left(1 - M_{B \rightarrow C}^{[3]}\right)\right) \oplus^{[3]} M_{C \rightarrow D}^{[3]} \\
 & = 1 - \left(\left(1 - M_{A \rightarrow B}^{[3]}\right) \left(1 - M_{B \rightarrow C}^{[3]}\right)\right) \left(1 - M_{C \rightarrow D}^{[3]}\right) \\
 & = 1 - \left(1 - M_{A \rightarrow B}^{[3]}\right) \left(\left(1 - M_{B \rightarrow C}^{[3]}\right) \left(1 - M_{C \rightarrow D}^{[3]}\right)\right) \\
 & = M_{A \rightarrow B}^{[3]} \oplus^{[3]} \left(1 - \left(1 - M_{B \rightarrow C}^{[3]}\right) \left(1 - M_{C \rightarrow D}^{[3]}\right)\right) \\
 & = M_{A \rightarrow B}^{[3]} \oplus^{[3]} \left(M_{B \rightarrow C}^{[3]} \oplus^{[3]} M_{C \rightarrow D}^{[3]}\right).
 \end{aligned}$$

単調性に関しては、以下の 2 段階に分けて確認する。

(1) ポリシー前半の $M^{[1]}$ の比較について確認する。 $M^{[1]}$ の集計方法 AND が、通常の意味での大小比較を反転しないことを確認すればよい。1 と AND を取れば、両辺がそのままなので、大小比較を反転しない。0 と AND を取れば、両辺が同じならよいが、異なるなら両辺とも 0 になるので、順序は保存されるとは限らない。しかしながら、これは最後の比較ではないので、等しくなることがあっても大小関係が反転することはないので、次の比較に確認を委ねることになる。

(2) ポリシー後半の $(M^{[2]} + M^{[3]})/M^{[0]}$ の比較については、計算が複雑で、確認できなかった。

一般に、各メトリックの集計方法が異なっているため、メトリックを総合的に考慮した演算やその逆演算は考えられず、上のように段階を踏んでも確認するが、それでも確認できないこともある。

5.6 動作条件の検証 (ルーティングポリシーの別例として辞書式順序)

ここでルーティングポリシーの別例として、メトリックに優先順位をつけて、個別に、そして順に優劣比較をする、辞書式順序を、考えてみた。

一意性は、従来のメトリックと、メトリック 3 種類とも、上で確認できている。単調性に関しては、それぞれ以下のように確認できる。

(0) 従来のメトリックに関して単調性は上で確認しているので、これで辞書式順序を組んでも問題はない。

(1) $M^{[1]}$ の比較は上で確認している。最後の比較でなければ、等しくなることがあっても AND で大小関係が反転することはないので、次の比較に確認を委ねることになる。

(2) $M^{[2]}$ の比較も同様に確認すると、同じく、最後の比較でなければ、等しくなることがあっても MIN で大小関係が反転することがないと判明したので、次の比較に確認を委ねることになる。

(3) $M^{[3]}$ の比較は、 $A:B$ を正順とすると、 $1-A:1-B$ は逆順となって、 $(1-A)(1-C):(1-B)(1-C)$ は、余程、ファイアウォールや IDS/IRS で全侵入を防御・検出できない、つまりメトリックの値 C が 1 にならない限り、等号が成り立つことなく逆順で、最後に、 $1-(1-A)(1-C):1-(1-B)(1-C)$ は正順に戻り、順序が保存される。

このように、ルーティングポリシーとして、従来のメトリック、又は完全なファイアウォールや IDS/IRS が存在しない仮定の下では、 $M^{[3]}$ を最後の比較とした、辞書式順序が使えることがわかった。

メトリックやルーティングポリシーに対する制約条件として、一意性と単調性がなければ、提案アルゴリズムが動作しないと判明したが、逆に、これらの条件を満たすメトリックやルーティングポリシーはどのように構成できるかは興味深い問題である。もしこの問題が解決できないなら、経路をすべて保持して、その最良となっているものを見出すことになるが、このような方法では指数的爆発が明らかである。どのようなものがメトリックとして使えるか、どのようなものがルーティングポリシーとして定められるか、その指針を明白に示すことを今後、目指していきたいと思う。

6. まとめと今後の課題

今日のインターネットにおいて広く使われている OSPF や RIP 等の経路制御方式では、単一数値のメトリックによる経路計算を行っており、複数メトリック値を用いた計算には対応していない。非数値など新型のメトリックを含めた、複数メトリックの場合の経路制御に対して、加重平均や、オペレーションズリサーチで使われる手法、遺伝的アルゴリズムを用いた手法などが研究されているが、重みづけの妥当性やアルゴリズムの簡便性において、経路制御には不適切だと思われる。

本研究では、非数値など新型のメトリックを含めた、複数メトリックのある経路制御問題に対して、妥当性や簡便性に配慮した経路制御のアルゴリズムを提案した。又、提案アルゴリズムの具体的な動作を I. A. Almerhag と M. E. Woodward 両氏が提案したセキュリティメトリック 3 種類を使って例示して、更に、メトリックやルーティングポリシーに対する制約条件に関して考察した。

その結果、提案アルゴリズムはダイクストラ法やベルマンフォード法に準じた簡便性をもつ上、[3] より引用したメトリック 3 種類も非常に良い性質を示すが、残念ながら、ルーティングポリシーの制約条件が厳しく、具体例として挙げたものは満たすことができなかった。

しかしながら、提案アルゴリズムは非常に簡便なので、今後は [3] より引用したメトリック 3 種類が共通してもつ良い性質の数学的な背景を確認しながら、使用できるメトリックとそれを用いて構成できるポリシーを検討していきたいと考えている。

文 献

- [1] Thomas L. Saaty, "Decision Making with the Analytic Hierarchy Process," International Journal Services Sciences, vol.1, no.1, pp.83-98, 2008.
- [2] D. Han, G. M. Hu, and C. Lu, "Multi-Object Optimal Secure Routing Algorithm Based on Differentiated Service Model," Computer Engineering and Applications, vol.44, no.19, pp.104-108, 2008.
- [3] I. A. Almerhag and M. E. Woodward, "Security as a Quality of Service Routing Problem," CoNEXT '05: Proceedings of the 2005 ACM Conference on Emerging Network Experiment and Technology, pp.222-223, ACM, New York, NY, USA, October 24-27, 2005. Toulouse, France.