2009-IOT-4（5）
2009 / 3 / 5

リンク状態型経路制御における局所更新手法

鈴木　一哉[†,††]　地引　昌弘[†]　　吉田　健一[††]

† 日本電気株式会社 システムプラットフォーム研究所
〒 211–8666 神奈川県川崎市中原区下沼部 1753
†† 筑波大学 ビジネス科学研究科
〒 112–0012 東京都文京区大塚 3–29–1

あらまし　リンク状態型経路制御においてトポロジー変化時に、経路計算の実施を経路更新を必要とするノードのみに限定することで、ネットワーク全体の処理負荷を軽減する手法の提案を行う。リンク状態型経路制御では、各ノードがネットワークトポロジー情報を共有した上で経路の決定が行われる。このためリンクやノードの故障によりトポロジーに変化が生じた場合、ネットワーク中の全ノードは新たなトポロジーに基づいた経路計算を行う必要がある。しかし故障箇所から離れているノードでは、経路変更の有無がパケット転送に影響を与えない可能性が高い。このため、経路更新の有無がパケット転送に影響を与えるノードにのみ経路計算を実施させることで、ネットワークの処理負荷の軽減が実現可能である。本論文では、故障を検知したノードが、故障前後の最短パスツリーを比較することで他のノードにおける経路更新の必要性を判断する手法の提案を行い、その効果について評価を行う。

キーワード　リンク状態型経路制御、IP ルーティング、ネットワークの安定性

A Local Update Method for Link-State Routing

Kazuya SUZUKI[†,††], Masahiro JIBIKI[†], and Kenichi YOSHIDA[††]

† System Platforms Research Laboratories, NEC,
1753, Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa, 211–8666, Japan
†† Graduate School of Business Sciences, University of Tsukuba,
3–29–1 Otsuka, Bunkyo-ku, Tokyo, 112–0012, Japan

Abstract　We propose a local update method, which limits calculating routes to nodes which need to update their routes when a failure has occurred. In link-state routing, all nodes need to share the topology information of network in order to calculate their routes. When a failure has occurred in the network, all nodes therefore need to recalculate their routes. But the failure is not likely to influence packet forwarding by nodes which is far from the failure, regardless of whether the nodes change thier routes or not. Our proposal therefore has nodes which need to update routes calculate routes in order to reduce loads of each nodes.

Key words　Link-State Routing, IP Routing, Network Stability

1. ま え が き

近年インターネットに代表される Internet Protocol を用いたネットワークは、我々の生活に欠かせない社会インフラとして存在意義を増している。その規模は年々拡大を続けている一方、従来の IP における経路制御では、一度障害が発生するとその影響範囲は非常に大きいという問題がある。

我々はドメイン内の経路制御方式として、広く用いられているリンク状態型方式に着目し、経路計算を故障箇所近辺のみで行う事により、ネットワーク全体としての負荷を軽減する方式の提案を行う。文献 [1] には、故障箇所から遠いノードでは経路表に影響を与える可能性が低くなる事が示されている。つまり、故障発生時には経路表を更新する必要のあるノードのみに経路計算を実行させることができれば、ネットワーク全体の負荷を軽減し、その安定性向上を実現できると考えられる。そのためには、リンク故障時に経路更新を必要とするノードの特定を行う必要がある。リンク故障により各ノードの経路に変更があるかを調べるためには、リンク故障後のトポロジーを元に、各ノードを起点とした経路計算を実施し、リンク故障前の経路と比較すればよい。しかし、本来各ノードで実施していた経路計算を故障検知ノードで代わりに実行しているだけであるため、この方法では経路計算の負荷を減らすという当初の目的を実現する事はできない。そのために本報告では、リンク故障発生時にネットワーク中のノードが経路更新を必要とするかを少ない計算量で判定するアルゴリズムを提案する。提案アルゴリズムはリンク故障発生時に故障リンクに接続しているノードにより実行され、実行ノードにおけるリンク故障前後の最短パスツリーを用いる事で、少ない計算量での判定を実現している。

本報告では、まず局所的な経路制御を実現するための従来技

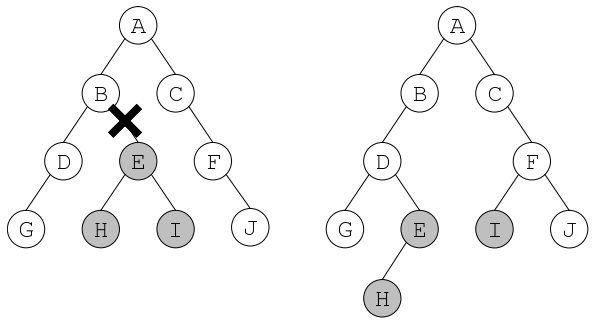


図 1　Incremental SPF によるツリー再構築
Fig. 1　Recalculation by Incremental SPF

術について述べ、その問題点を指摘する。次に、指摘の問題点を解決するための局所更新手法について述べる。局所更新手法を実現するための、リンク故障時に経路更新を必要とするノードを判定する条件およびその判定アルゴリズムについて述べた後、提案手法についてシミュレーションを用いた評価を行う。最後に今後の課題について述べる。

2. 従来技術とその問題点

本章では、経路計算の局所化を行うための従来手法について説明を行い、それぞれの手法の問題点について述べる。

2.1 OSPF における階層化

代表的なリンク状態型経路制御プロトコルである OSPF [2] では、経路計算の局所化を実現するために階層化の手法が用いられている。これは、ネットワークをいくつかのエリアに分割し、エリア内外で異なる経路計算を行う手法である。他のエリアに対しては、詳細なトポロジー情報に代わりサマリー情報を通知する事により、エリア内に流れるトポロジー情報の量を減らし、経路計算負荷の軽減を実現している。しかし、OSPF における階層化の手法には、以下の問題がある。

（1） 階層化レベルに関する問題
OSPF は、その仕様上二階層までしか扱う事ができない。

（2） 境界の設定に関する問題
運用中にエリアの境界を変更する事は、非常に困難である。そのため、運用前に適切にエリア分割を行う必要がある。

（3） 詳細情報の隠蔽に関する問題
エリアの境界をまたいでの最適経路の計算ができない。

(1) の問題は OSPF 特有の問題であるため、OSPF の仕様を拡張し多階層を扱えるようにするというアプローチも考えられる。しかしそれでも、(2), (3) の問題は残るため、運用者の負担を増加させない手法の確立が望まれる。

2.2 Incremental SPF

OSPF における経路計算の負荷軽減を目的とした Incremental SPF [3] と呼ばれる手法が存在する。この手法では、経路計算の過程で生成される最短パスツリーを保持しておき、トポロジー変更時に最短パスツリー上で影響のある部分のみ再構築を行う。Incremental SPF による最短パスツリーの再構成について、図 1 を用いて説明を行う。図 1 (a) はリンク故障が生じる前にノード A が保有する最短パスツリーを表している。ノード B とノード E を結ぶリンクに故障が生じた場合、影響を受けるノード E, H, I のみを対象とし、ツリーの再構築を行う。再構築の結果得られる最短パスツリーが図 1 (b) となる。

本手法を用いることで、トポロジー変更後の新たな最短パスツリーを、ツリー全体の再構築を行うことなく、差分の計算のみで算出することが可能である。このことによりトポロジー変更時における経路計算の負荷を軽減することができる。

経路計算の局所化の観点から本手法を見てみると、現在のトポロジーを元に計算された最短パスツリー上に故障リンクが含まれていない場合、故障の有無に関わらず最短パスツリーに変更は生じない。このため最短パスツリー上に故障リンクが含まれていない場合には経路計算を実施しなければ、経路計算の局所化を実現できる。しかし、この場合経路更新が必ずしも必要ないにも関わらず、経路計算を実施してしまう場合がある。図 1 中のノード A にとって、宛先ノード E, H に対する次転送先は、故障の有無に関わらずノード B である。このように最短パスツリー中に故障リンクが存在しても、経路変更が行われない場合がある。このため、経路変更を必要とするノード数よりも多くのノードが経路計算を実施することになる。これがどの程度のオーバーヘッドとなるかについては、4.2 で提案手法との比較評価を行う。

2.3 高速迂回

故障を検知したノードが、即座にパケット転送を、予め準備された代替経路、パスへと切り替える高速迂回 (Fast Reroute) と呼ばれる手法が提案されている [4]〜[7]。これらの手法における利点はサービス停止時間の短縮を実現できる点だけでなく、故障箇所の周辺のノードのみで対処が可能な点も挙げられる。このため、本手法を適用することでネットワーク全体における経路制御負荷の低減も期待できる。しかし、文献 [4]〜[6] の手法は IP とは異なるフォワーディング方式を前提としており、一般の IP ネットワークにそのまま適用することはできない。また、文献 [7] の手法は、ループが生じない代替経路に対してのみ高速迂回を適用する手法であるため、ループが生じる経路への対策が別途必要となる。

3. 局所更新手法

提案手法では、経路計算の実施を経路更新が必要なノードに限定することで、ネットワーク全体の負荷を軽減する。本手法はネットワークの階層化を必要とせず、大規模ネットワークに対しても適用可能である。そのため事前に階層化設計を行うといった運用者の手間を必要としない。また、一般の IP フォワーディング方式の変更も必要としない。以下、本章では経路更新が必要かを判別するための条件について示し、その条件を判定するためのアルゴリズムを示す。

3.1 経路更新が必要となる条件

リンクに故障が発生したとき、ネットワーク中のどのノードに経路更新が必要になるにかについて、図 2 に示すネットワークを用いて説明する。図 2 の (a), (b) は、それぞれノード B に接続するリンクに故障が発生する前後の状態を表しており、また図中の矢印は各ノードから宛先ノード D への経路を表している。

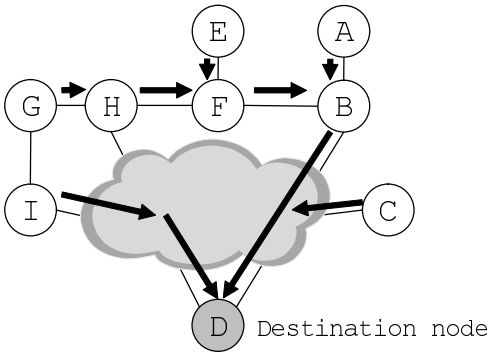
故障前後に経路が変わる可能性があるノードは、故障リンクの上流にあるノード A, B, E, F, G, H である。故障発生前後を比較すると、ノード B, F, G, H において経路が変わっている。ノード B, F, H に関しては、故障前の最短パスはノード B → ノード F → ノード H であるが、故障後の最短パスはノード H → ノード F → ノード B と向きが逆になっている。つまり、これらのノードのいずれかが経路更新を行わなかった場合、ループが発生してしまう。一方ノード G に関しては経路が変わっているが、経路の変更を行わなくてもループとなる事がないため、必ずしも経路更新を必要としない。

つまり、以下の条件を満たすノードを見つければよい。

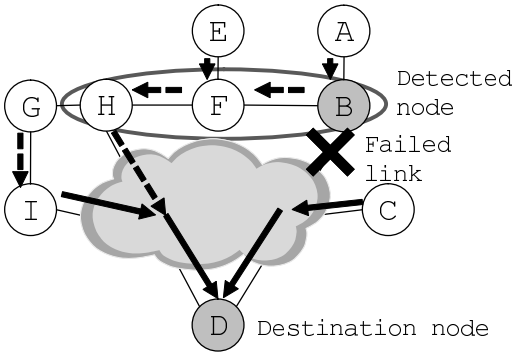
[条件 1]　故障発生前、自身の最短パスツリー中における宛先までのパス上に故障リンクが存在する。

[条件 2]　故障発生後、故障リンクに接続するノード (検知ノード) の最短パスツリーにおいて宛先までのパス上に自身が存在する。

[定理 3.1]　ネットワーク中の単一リンク故障に対し、リンク故障後のトポロジーを元にすべてのノードが経路更新を実施した場合、すべてのノード間において IP のパケット転送が可能であると仮定する。このとき、ネットワーク中のいずれかの宛先ノードに対し条件 1, 2 を満たすノードのみを経路更新を行



(a) Before failure



(b) After failure

図 2　経路更新が必要となるノード
Fig. 2　Nodes required route update

えば、すべてのノード間において IP のパケット転送が可能である。

［証明 3.1］ 仮定より、故障後のトポロジーにおいてすべてのノード間には必ずパスが存在する。このため、IP のパケット転送においてある宛先に対するパケットがその宛先に到達できないのは、宛先までのパス中にループが生じる場合のみである。

ここで、ある宛先ノード n_d に対し、ネットワーク中のノードを以下の三つの集合に分けて考える。

- ネットワーク中のいずれかの宛先ノードに対して、条件 1, 2 共に満たすノードの集合 $\mathcal{V}_u$
- $\mathcal{V}_u$ に含まれず、宛先ノード n_d に対し条件 1 を満たすノードの集合 $\mathcal{V}_1$
- $\mathcal{V}_u, \mathcal{V}_1$ いずれにもに含まれないノードの集合 $\mathcal{V}_2$

ここで集合 $\mathcal{V}_1, \mathcal{V}_2$ 中のノードは故障前のトポロジーに基づいて計算された経路表を持ち、集合 $\mathcal{V}_u$ 中のノードは故障後のトポロジーに基づいて計算された経路表を持つ。同一のトポロジーを元に計算された経路表を持つノード間におけるパケット転送ではループは生じない。このためループが生じる可能性があるのは、$\mathcal{V}_u$ 中のノードから $\mathcal{V}_1, \mathcal{V}_2$ 中のノードに転送された場合、もしくはその逆の場合のいずれかのみである。

次に宛先ノード n_d が上記のいずれの集合に含まれるか考える。宛先ノード n_d が条件 1 を満たさないのは自明であるため、n_d は集合 $\mathcal{V}_1$ の要素ではありえない。そのため $n_d \in \mathcal{V}_2$ もしくは $n_d \in \mathcal{V}_u$ のいずれかである。

まず $\mathcal{V}_1$ 中のノードが宛先 n_d へのパケットを他のノードから受信した場合、もしくはパケットの送信元であった場合について考える。$n_d \notin \mathcal{V}_1$ であるため、宛先 n_d へのパケットはいずれ $\mathcal{V}_1$ 中のノードから $\mathcal{V}_u$ もしくは $\mathcal{V}_2$ 中のいずれかのノードへと転送される。

次に $\mathcal{V}_u$ 中のノードについて考える。もし $n_d \in \mathcal{V}_u$ である場合、ノード n_d 宛てのパケットは、$\mathcal{V}_u$ 中のノードのみを経由して、宛先 n_d へと到達する。一方 $n_d \notin \mathcal{V}_u$ つまり $n_d \in \mathcal{V}_2$ である場合、n_d 宛てのパケットは、$\mathcal{V}_u$ 中のノードから $\mathcal{V}_1$ もしくは $\mathcal{V}_2$ 中のいずれかのノードへと転送される。ここでノード $n_u(n_u \in \mathcal{V}_u)$ からノード $n_1(n_1 \in \mathcal{V}_1)$ へとパケットが転送されると仮定する。このとき n_u における n_d への最短パス上に n_1 が存在することとなる。しかし、このことは n_1 が条件 2 を満たさないという仮定と矛盾する。つまり $n_d \in \mathcal{V}_2$ である場合のみ、n_d 宛てのパケットは、$\mathcal{V}_u$ 中のノードから $\mathcal{V}_2$ 中のノードへ転送される。

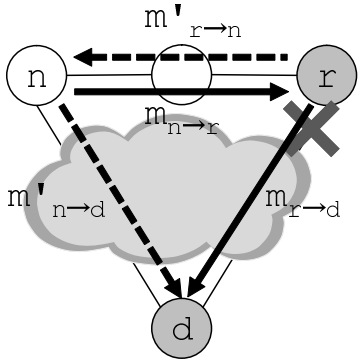


図 3　経路更新が必要となる条件
Fig. 3　Condition for requirement of route update

最後に $\mathcal{V}_2$ に含まれるノードについて考える。まず $n_d \in \mathcal{V}_2$ である場合、この集合中のノードが受信したもしくは送信元である n_d 宛てのパケットは、$\mathcal{V}_2$ 中のノードのみを経由して宛先 n_d まで到達する。次に $n_d \notin \mathcal{V}_2$ つまり $n_d \in \mathcal{V}_u$ である場合について考える。$\mathcal{V}_2$ のノードは仮定より、故障発生前におけるノード n_d までの最短パス中に故障リンクを含まない。また、この最短パス中のノードは $\mathcal{V}_1$ 中のノードではない。以上より $n_d \in \mathcal{V}_u$ である場合のみ n_d 宛てのパケットは、$\mathcal{V}_2$ 中のノードから $\mathcal{V}_u$ 中のノードへ転送される。

以上により、ノード n_d を宛先とするパケットに対し上記の異なる集合に含まれるノード間でパケット転送が行われるのは、以下の三つのいずれかの場合のみである。

（1） $n_1(n_1 \in \mathcal{V}_1)$ から $n_u(n_u \in \mathcal{V}_u)$ もしくは $n_2(n_2 \in \mathcal{V}_2)$ へ

（2） $n_d \in \mathcal{V}_u$ のとき、$n_2(n_2 \in \mathcal{V}_2)$ から $n_u(n_u \in \mathcal{V}_u)$ へ

（3） $n_d \notin \mathcal{V}_u$ のとき、$n_u(n_u \in \mathcal{V}_u)$ から $n_2(n_2 \in \mathcal{V}_2)$ へ

上記における (2) の場合、$\mathcal{V}_u$ 中に n_d が存在するため、$\mathcal{V}_u$ 中のノードに至った宛先 n_d のパケットは、$\mathcal{V}_u$ 中のノードのみを経由して宛先へと到達する。また (3) の場合についても同様に $\mathcal{V}_2$ 中のノードに至ったパケットは、$\mathcal{V}_2$ 中のノードのみを経由して宛先へと到達する。このため、異なる集合に含まれるノード間のパケット転送において、ループは生じない。

以上から、リンク故障に対する経路更新の実施を集合 $\mathcal{V}_u$ に含まれるノードのみしか行わなかった場合でも、全ノード間において IP のパケット転送は可能である。 □

3.2 判別手法

ここでは、前章で述べた条件を各ノードが満たすかを判定する方法について述べる。条件 2 を満たすノードは、故障リンクに接続しているノードにおける故障後の最短パスツリーを用いて、宛先まで順にたどれば見つける事が可能である。このためノード n が、条件 2 を満たすと仮定したときに、条件 1 を満たすかどうかを調べる方法について検討を行う。

図 3 中のノード r は故障リンクに接続しているノード、ノード d は宛先ノードとする。ここでノード i からノード j までの故障前後の最短パスにおけるメトリックをそれぞれ $m_{i\to j}, m'_{i\to j}$ と表記する。

ノード n が条件 1 を満たすということは、宛先ノード d までの最短パスが故障前後で変わるということである。つまり、式 (1) が成り立てば、条件 1 が成り立つという事ができる。

$$m'_{n\to d} \geqq m_{n\to r} + m_{r\to d} \quad (1)$$

リンク故障後のトポロジーにおけるノード n からノード d へのメトリック $m'_{n\to d}$ は、ノード r からノード d へのメトリック $m'_{r\to d}$ からノード r からノード n へのメトリック $m'_{r\to n}$ を引いた値と等しい。つまり式 (1) は、式 (2) へと変形できる。

$$m'_{r\to d} - m'_{r\to n} \geqq m_{n\to r} + m_{r\to d} \quad (2)$$

ここでノード r からノード n へのメトリックはリンクの故障前後で変わらないため、式 (2) は最終的に式 (3) となる。

$$m'_{r\to d} - m_{r\to d} \geqq m_{r\to n} + m_{n\to r} \quad (3)$$

式 (3) の左辺の値は、故障前後におけるノード r からノード d へのメトリックの差である。つまりノード r において、故障前後の最短パスツリーからから求める事が可能である。また、式 (3) の右辺の第一項は、やはりノード r における故障前の最短パスツリーから求める事が可能である。式 (3) の右辺の第二項の値を求めるためには、ノード r において自身を起点とする逆方向の最短パスツリーを計算した上で求める必要がある。式 (3) 中で用いられている各メトリックの意味とその算出方法を表 1 に示す。

記号	説明	算出方法
$m'_{r\to d}$	故障後の経路表におけるノード r からノード d へのメトリック	最短パスツリー (故障後) を使用
$m_{r\to d}$	故障前の経路表におけるノード r からノード d へのメトリック	最短パスツリー (故障前) を使用
$m_{r\to n}$	故障前の経路表におけるノード r からノード n へのメトリック	最短パスツリー (故障前) を使用
$m_{n\to r}$	故障前の経路表におけるノード n からノード r へのメトリック	逆方向最短パスツリー (故障前) を使用

表 1　メトリックと算出方法
Table 1　Metrics and their calculation

3.3　提案アルゴリズム

図 4 は、前章で述べた判別手法をまとめたアルゴリズムである。表 2 に、アルゴリズム中で用いられている記号の説明を示す。

```
1:  T ⇐ SPF(r, V, E)
2:  T_i ⇐ SPF(r, V, E − {e_i})
3:  T^R ⇐ rSPF(r, V, E)
4:  V_u ⇐ φ
5:  for ∀d ∈ V do
6:    t_d ⇐ m(r, d, T_i) − m(r, d, T)
7:    n ⇐ Child(r, d, T_i)
8:    while m(r, n, T) + m(n, r, T^R) ≦ t_d do
9:      V_u ⇐ V_u ∪ {n}
10:     n ⇐ Child(n, d, T_i)
11:   end
12: end
```

図 4　提案アルゴリズム
Fig. 4　Proposal algorithm

このアルゴリズムはリンク e_i に故障が発生したときにノード r において実行され、その結果経路更新が必要なノードの集合として V_u が得られる。図 4 中の 1〜3 行目は、それぞれ故障前後の最短パスツリー $\mathcal{T}, \mathcal{T}_i$ および逆方向最短パスツリー $\mathcal{T}^R$ を求める処理である。次にネットワーク中のすべての宛先ノードを対象として、6〜11 行目の処理が繰り返し実行される。6 行目は、式 (3) の左辺の値を求める処理である。7 行目および 10 行目は、条件 1 を満たすノードについて処理を行うために、故障後の最短パスツリー $\mathcal{T}_i$ に沿って自ノード r から宛先ノード d へのパスに沿って順に判定処理を行うためのノード n を決定している。ここで決定されたノード n は、8 行目により式 (3) の不等式を満たしているかの判定が行われる。このとき不等式を満たしていれば、ノード n を V_u の要素として加え (9 行目)、次の宛先ノード d へのパス上におけるノード n の隣接ノードを新たなノード n とし (10 行目)、8 行目の判定処理が繰り返される。

記号	説明
$\mathcal{V}$	ネットワーク中のノードの集合
$\mathcal{E}$	ネットワーク中のリンクの集合
$SPF(r, \mathcal{V}, \mathcal{E})$	ノード r を起点とした $\{\mathcal{V}, \mathcal{E}\}$ 上の最短パスツリー
$rSPF(r, \mathcal{V}, \mathcal{E})$	ノード r を終点とした $\{\mathcal{V}, \mathcal{E}\}$ 上の逆方向最短パスツリー
$Child(i, d, \mathcal{T})$	起点からノード d に至る最短パスツリー $\mathcal{T}$ 上のノードにおいて、ノード i の次ホップとなるノード
$m(r, d, \mathcal{T})$	ノード r からノード d までの $\mathcal{T}$ 上でのメトリック

表 2　記号の説明
Table 2　Notation

このアルゴリズムツリー $\mathcal{T}, \mathcal{T}_i$ は経路計算を行う過程で得られる。リンク故障前後の経路計算は、本提案手法を適用していなくても行われる計算である。図 4 における、逆方向の最短パスツリーの算出 (3 行目) と経路更新必要性の判定 (4 行目以降) の処理が、本提案アルゴリズムを実行するノードにおける追加の計算コストとなる。

3.4　提案アルゴリズムの改良

図 4 に示す提案アルゴリズムでは、5 行目以降の処理についてネットワーク中の全ノードをそれぞれ宛先ノードとして処理を行っている。しかし、必ずしも全ノードに対して適用する必要はない。以下に示す宛先ノードに対しては、判定処理を行わなくてもよい。

- 故障前の経路表における次転送先が、故障リンクの対向ノードとなっていない宛先ノードに対しては、適用する必要はない。上記の宛先ノードに対するパスはリンク故障により変化しないため、式 (3) の左辺が必ず 0 になるためである。
- ある宛先ノード d' に至るパスが、別の宛先ノード d へと至るパスの一部であるとき、その宛先ノード d' に対しては判定処理を行う必要はない。宛先ノード d' に対するパス上のノード n に対して判定を行い、経路更新が必要であると判断された場合、そのノード n は宛先ノード d に対する判定でも必ず経路更新が必要であると判断されるためである。

以上のことを考慮する事で、図 4 における繰り返し処理の回数を削減する事ができる。

4.　評　　価

ここではシミュレーションを用いて、提案手法の効果について評価を行う。シミュレーションでは、実際のネットワークのモデルとして用いられる Barabashi-Albert(BA) モデルを用いる。このモデルに従ったネットワークトポロジーの生成には、BRITE [8] というツールを用いた。

4.1　経路更新を必要とするノードの割合

ネットワーク中のあるリンクに単一故障が生じたときにどの程度の割合のノードが、提案アルゴリズムにより経路更新が必要であると判定されるかを調べた。リンクメトリックが対称なネットワークについて、ネットワーク中のノード数を 100〜500 とし、ノードの平均度数を 4〜10 として、それぞれ 10 回ずつシミュレーションを行った。ネットワーク中のすべてのリンク

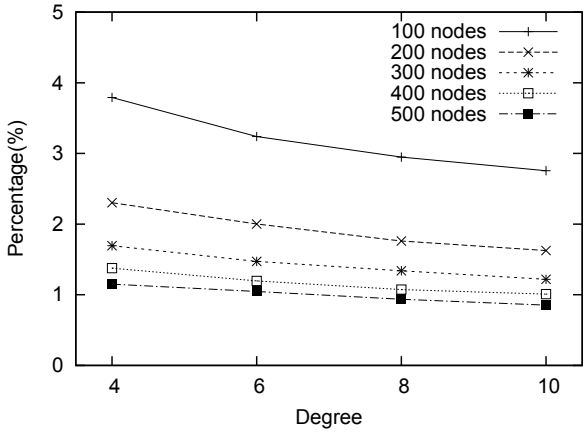


図 5　経路更新必要ノードの割合 (平均)
Fig. 5　A rate of nodes required route update(mean)

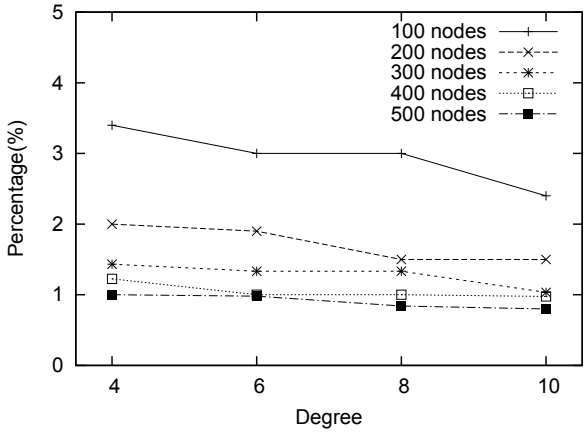


図 6　経路更新必要ノードの割合 (中央値)
Fig. 6　A rate of nodes required route update(median)

の単一故障について、経路更新が必要となるノード数の全体に対する割合を調べ、その平均値および中央値をまとめたのが、それぞれ図 5 および図 6 である。

図 5, 6 は、提案手法を用いた場合それぞれ全体に対して 1% から 4% 程度のノードのみが経路更新を行えばよいことを示している。度数が大きくなるにつれて経路更新が必要となるノード数が少なくなっているのは、度数が大きくなることはすなわちネットワーク中のリンク数が多くなることであり、一つ一つのリンクの重要度が相対的に下がるためであると考えられる。

図 7 は、あるリンクの故障が発生したとき経路更新を必要とするノードの数を、ネットワーク中のすべてのリンクに対してそれぞれ調べ、ノード数を横軸とした度数分布にまとめた図である。ネットワークの規模が 100,300,1000 の場合における最頻値は、それぞれ 3,4,5 となっている。また故障時に経路更新が必要とするノード数が 10 以内であるリンクが、ネットワーク規模が 1000 の場合において全体の約 88% となっている。このようにネットワーク中大半のリンクにおける単一故障では、たかだか 10 程度のノードが経路更新を行えばよいということを示している。一方、故障により多くのノードの経路更新を必要とするリンクも少数ながら存在する。しかし、それぞれ最大値は 11,31,121 であり、ネットワーク全体に対してたかだか 10% ～ 12% 程度の値である。

4.2　他手法との比較

次に以下の三つの場合について、それぞれ比較を行った。

（1） 提案手法を適用した際に、経路更新が必要であると判断されるノード数 (Proposal method)

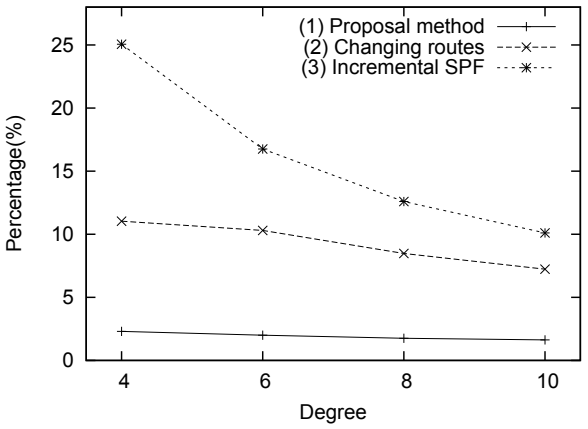


図 8　他手法との比較 (割合)
Fig. 8　Comparison of other methods

（2） 局所更新を行わなわず全ノードが経路計算を実施した場合に、実際に経路に変更が生じたノード数 (Changing routes)

（3） Incremental SPF を適用した際に、経路更新が必要であると判断されるノード数 (Incremental SPF)

ネットワーク規模は 200、平均度数は 4～10 として、それぞれ 10 回ずつシミュレーションを行い、その結果をまとめたのが図 8 である。また度数 4 の場合について、単一リンク故障発生時における前記 (1) ～ (3) のノード数を全リンクに対してそれぞれ調べ、まとめた度数分布を図 9 に示す。

図 8, 9 を見ると (3) Incremental SPF は、(2) Changing routes と比べ、大きな値となっている。つまり Incremental SPF を適用した場合経路更新が必要と判断されるノードの数は、実際に経路が変更になるノードの数よりも多いことを示している。その理由は、図 1 におけるノード E, H のように最短パスツリー中に含まれるが、経路計算を行っても経路に変更が生じない場合が存在するためである。

一方 (1) Proposal method は、(2) Changing routes の値と比べて、小さい値となっている。その理由は、提案手法では条件 1 を満たすが、条件 2 を満たさないノードに対して経路計算を行う必要があると判断しないためである。これらのノードは、実際に経路計算を実施した場合最短パスが変わる可能性があるが、定理 3.1 の証明で示したように、経路計算実施の有無がパケットの到達性には影響がないため、提案手法では経路計算が必要であると判断されないためである。

(1) Proposal method は、(3) Incremental SPF を用いた場合の約 1/11 ～ 1/6 のノードの経路計算しか必要としないことが、図 8 から分かる。また図 9 においても (3) Incremental SPF では故障発生時に経路計算が必要と判断されるノード数の最大値が 199 であるのに対し、(1) Proposal method では最大でもたかだか 24 である。このように提案手法では、Incremental SPF を使う場合と比較し、より少数のノードの経路計算のみを行えばよいことを示している。

5.　経路制御への提案手法適用に関する考察

本章では、3. において提案したアルゴリズムを実際の経路制御にどのように適用すべきかについて検討を行う。

提案アルゴリズムは故障リンクに接続する両端のノードにおいて実行される。この時、経路計算が必要であると判断された各ノードに対してのみリンク故障を通知し、他のノードには通知を行わないといった方法が考えられる。リンク故障を通知されたノードのみが経路計算を実施し、経路の更新を行う。しかしこの方法では、あるリンク e_1 の故障に対し前述の局所更新を実施した後に、さらに別のリンク e_2 に故障が生じた場合に問題が生じる可能性がある。リンク e_2 の故障に対し経路計算

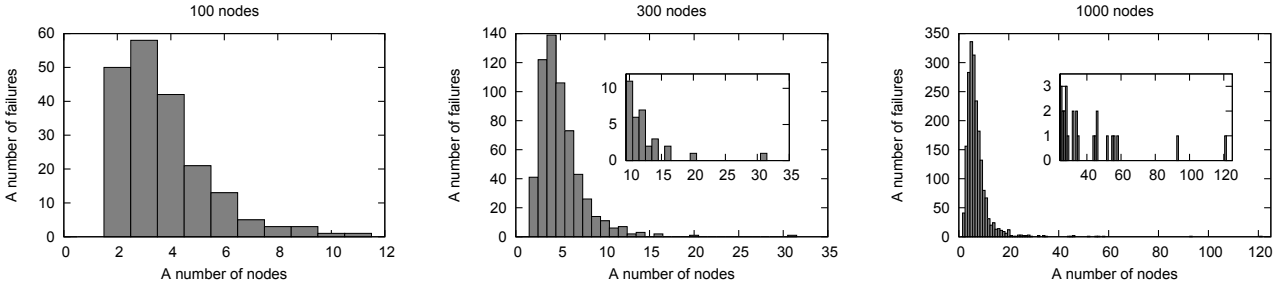


図 7　単一リンク故障で経路更新を必要とするノード数の度数分布
Fig. 7　Frequency distribution for nodes requiring route update by a link failure

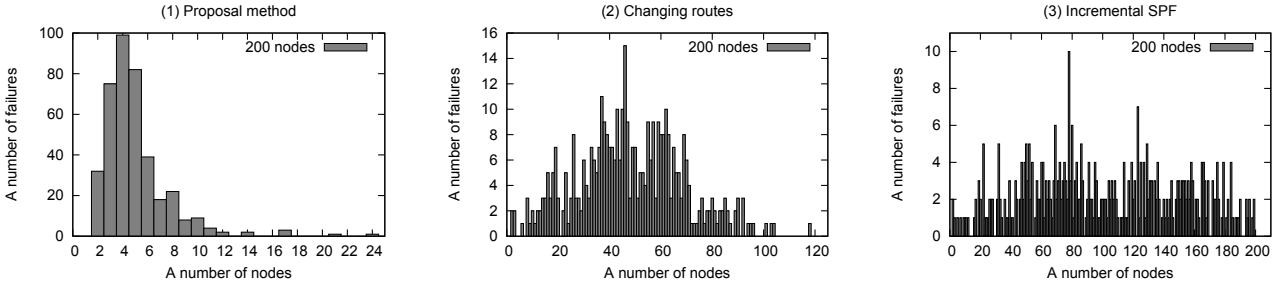


図 9　他手法との比較 (度数分布)
Fig. 9　Frequency distribution by each methods

が必要とされるノードの中には、リンク e_1 の故障を通知されたノードと通知されていないノードの両方が存在する可能性がある。異なるトポロジー情報を元に経路計算した結果は、お互いに矛盾し、経路ループが生じる可能性がある。このため局所更新実現のためには、リンク故障の通知はネットワーク全体に行うこととし、経路計算の実行のみを提案アルゴリズムにより得られるノードに限定する方法がよいと考えられる。

4.2 では Incremental SPF をリンク故障時に各ノードが経路更新を必要とするかどうかを判別するための手法の一つとして、提案手法との比較を行った。しかし Incremental SPF を用いる場合の大きな利点は、経路計算の際に用いられる最短パスツリーの再構築をより少ない計算コストで実施できる点である。つまり、経路計算の必要性を判断するためには提案手法を用い、実際の経路計算を行う際には Incremental SPF を用いるということで、両者の利点をそれぞれ生かすことが可能である。提案手法と Incremental SPF を組み合わせて使用することで、それぞれ単独で用いる場合と比べ、経路計算を実施するノード数をより少なく出来、また実施される経路計算の負荷もより少なくすることが出来る。

6. あとがき

本報告では、まず局所的な経路制御を実現するための従来技術について述べ、その問題点を指摘した。次に、指摘の問題点を解決するための局所更新手法の提案を行った。さらにシミュレーションにより、提案手法を適用した場合、リンク故障時ネットワーク中のどの程度の割合のノードが経路更新を必要とするかについて調べた。その結果規模 100〜500、平均度数 4〜10 のネットワークにおいてリンクの単一故障発生時に、全ノードのうち平均で 1〜4 % 程度、また最大でも 10 数 % 程度のノードのみが経路更新を行えばよいことを分かった。また提案手法を用いることで、経路更新を必要とするノード数を従来手法と比較し、約 1/11 〜 1/6 に抑えることが可能であること示した。さらに、提案手法を実際の経路制御に適用する場合について、考察を行った。

しかし、経路制御の局所更新を実現するためには、まだ以下の課題がある。

- 本報告ではポイントトゥポイント型のネットワークにおけるリンク故障のみを対象とした検討であったため、ブロードキャスト型ネットワークやノード故障に対する検討も必要である。
- 提案方式における計算量に関する検証が十分行われていない。

今後これらの課題の解決に向けた検討を行っていく予定である。

謝辞

本研究の一部は、総務省の委託研究「次世代バックボーンに関する研究開発」プロジェクトの成果である。

文　献

[1] K. Suzuki and M. Jibiki : “Formalization and Analysis of Routing Loops by Inconsistencies in IP Forwarding Tables”, IEICE Trans. Communications, vol.E90-B, no.10, pp.2755–2763, 2007.

[2] J. Moy : “OSPF Version 2”, RFC2328, IETF, 1998.

[3] J. M. McQuillan, I. Richer and E. C. Rosen: “The New Routing Algorithm for the ARPANET”, IEEE Trans. Communications, vol.28, No.5, pp.711–719, 1980.

[4] S. Nelakuditi, S. Lee, Y. Yu, Z. Zhang and C. Chuah : “Fast Local Rerouting for Handling Transient Link Failures”, IEEE/ACM Trans. Networking, vol.15, No.2, pp.359–372, 2007.

[5] J. Wang and S. Nelakuditi : “IP Fast Reroute with Failure Inferencing”, Proceeding of ACM INM 2007.

[6] P. Pan, G. Swallow and A. Atlas : “Fast Reroute Extensions to RSVP-TE for LSP Tunnels”, RFC 4090, IETF, 2005.

[7] 鈴木一哉, 地引昌弘 : “IP 高速迂回実現のための迂回可能経路の判別手法の提案”, 信学技報, vol.107, TM2007-59, 2008.

[8] A. Medina, A. Lakhina, I. Matta and J. Byers : “BRITE: An Approach to Universal Topology Generation”, In Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems - MASCOTS ’01, 2001.