経路制御における
ネットワーク可用性向上に関する研究

筑波大学審査学位論文 (博士)

2011

鈴木 一哉

筑波大学大学院
ビジネス科学研究科 企業科学専攻

論文概要

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、広帯域化、低価格化が進み、その上で多くの人による様々なアプリケーションの利用が可能となっている。そのため、既存の電話網 (PSTN : Public Switched Telephone Network) が従来果たしていた社会インフラとしての役割は、IP を用いたネットワークによって取って代わられようとしている。

電子メールに代表される IP ネットワーク上で旧来用いられていたアプリケーションは、ネットワークの一時的な停止などに、比較的寛容であった。仮に、ネットワークが一時的に停止しても、復旧後に再送を行うことで、利用者には停止の影響を感じさせることはなかった。しかし、今日 IP ネットワーク上では、ネットワークの停止が利用者に大きな影響を与えるアプリケーションの利用が拡大している。例えば、VoIP に代表されるリアルタイム系のアプリケーションは、短時間であってもネットワークが停止すると、利用者に多大な影響を与える。また金融取引などのビジネスアプリケーションでは、ネットワークの停止が金銭的な損失につながる可能性が高い。IP ネットワークには、そのサービスを停止・中断することなく提供し続ける能力、つまり可用性が、従来以上に高いことが求められるようになってきている。

現状の IP ネットワークを用いた電気通信サービスにおいては、長時間のサービス停止を伴う障害が毎年多数発生している。日本では電気通信事業法において、電気通信役務の提供の停止又は品質低下を受けた利用者数が 3 万人以上、もしくは停止時間もしくは品質低下の時間が 2 時間以上に及ぶ重大事故は、総務省への報告が義務付けられている。総務省により毎年公表されている重大事故の件数は、2003 年度の 7 件から 2009 年度には 18 件に増加している。さらに 2009 年度には 1000 万人以上に影響を与える大きな事故も発生している。

このような状況に対し、総務省の諮問を受けた情報通信審議会において、ネッ

トワークの IP 化に対応した安全・信頼性対策に関する答申がまとめられている。その中でネットワーク設備の運用・管理面についての課題の一つとして、「IP ネットワークの早期異常検知機能等の設備監視技術と予備系装置への自律切替などの研究開発」が挙げられており、この分野の技術開発が望まれている。

上記背景を踏まえ、本研究は IP ネットワークの可用性を向上させることを目的とした。IP ネットワークにおいては、予め冗長に構成されたノードやリンクを、故障発生時に切り替えて使用するという手段で、障害復旧が行われている。可用性向上のためには、この障害からの復旧時間をより短くする必要がある。しかし、これまでに提案されてきた手法ではネットワーク中の各ノードへの故障通知や経路の再計算にはネットワークの規模に応じた時間がかかるため、大規模なネットワークでは切替時間が長くなるという問題があった。

本研究では、故障により影響を受ける範囲の局所性に着目し、この問題の解決を目指した。あるノードが、故障によって影響を受けるか否かは、故障箇所からの距離に関係すると推測される。たとえば、故障箇所から遠いノードは、故障による経路更新を行わなくても、パケットの到達性に影響を与えない可能性が高い。一方で故障箇所に近いノードが経路更新を行わなかった場合、パケットは正しく転送されず、その宛先まで到達しない可能性が高くなると考えられる。この局所性をうまく活用することで、ネットワークの規模に関わらず、ほぼ一定の時間で故障発生時における切替時間を短縮することを考えた。

具体的には、まず第 3 章において、故障発生時における経路更新の有無がパケット到達性に与える影響を調べた。まず故障によるトポロジー変化に起因して到達不能ノードが生じる条件が、以下の四つであることを明らかにした。

- 宛先ノード自体の故障
- 宛先ノードまでのパスの喪失
- 次転送先ノードの故障
- ルーティングループの発生

これらの条件のいずれかに当てはまることが到達不能ノードの発生の必要十分条件であることを示した後、パケット転送動作復旧のためには、それぞれの条件に

対してどのように対処すべきかの考察を行った。上記条件のうち、次転送先ノードの故障とルーティングループ発生は、パケット到達性の復旧のために各ノードにおける適切な経路更新が必要となるため、経路制御において対処を行うべき問題である。隣接ノードである次転送先の故障は、高速検知技術を用いて故障を検知後、即座に対処を行えばよい。一方、ルーティングループ発生は、故障箇所に隣接していない場合に到達不能ノードが生じるケースである。さらに、このケースは故障箇所からの距離が遠いほど発生する可能性が低いことを、シミュレーションを用いて示した。

次に、第 4 章において、この解析結果を元にした経路変更箇所の局所化アルゴリズムの提案を行った。故障発生時におけるパケット到達性を高速に復旧させるためには、経路更新を必要とするノードを特定することが重要である。これらのノードを特定するための条件として、以下の二つの条件を示した。

- 経路更新を必要とするノードは、故障前のトポロジーにおいて、自身から宛先ノードへの最短パス中に故障リンクを含む。
- 経路更新を必要とするノードは、故障後のトポロジーにおいて、故障リンクに接続しているノードから宛先ノードへの最短パス上に存在する。

さらにこれらの条件を共に満たすノードのみが経路更新を行うことにより、パケット到達性を復旧できることを証明した。

次に、これらの条件の判定処理を、ネットワーク中の各ノードにより分散判定を行う方法を示した。この分散判定は、一箇所で集中して判定を行う一括処理と比較し、計算量の面でオーバーヘッドが存在しない。また、この分散判定を行うための手順として局所化アルゴリズムを示した。このアルゴリズムを用いることで、高速迂回を実現する際に必要となる、故障発生時に経路更新を必要とするノードの特定が可能である。

そして、このアルゴリズムを使った二つの高速復旧手法を第 5 章、第 6 章において、それぞれ提案した。一つ目の事前計算型経路更新手法では、故障発生時に使用する代替経路表を事前に計算しておくことにより、復旧時間の短縮を実現する。一般に、代替経路表は故障箇所毎に異なるため、大規模なネットワークでは非常に多くの代替経路表を事前に計算する必要がある。これに対して、事前計算

型経路更新手法では、局所化アルゴリズムを使用することでこの代替経路表数を実装可能な数まで抑える。このことにより、必要とする代替経路表数を従来と比べ、平均で 1/100 に削減できる。さらに、代替経路表の計算コストと格納メモリ量は、実際の装置に実装を行う際に問題とならない程度に小さいことを示した。

局所化アルゴリズムを使った二つ目の高速復旧手法である故障箇所高速迂回手法は、故障を検知したノードが、故障の影響を受けないノードまでトンネリングを用いることで、パケットを迂回させる。この手法は、検知ノードを除く各ノードは自身の経路表を更新する必要がないため、検知ノードからの故障通知を必要としない。そのため、高速迂回手法は、事前計算型経路更新手法において故障通知にかかっていた時間の短縮を実現する。また、この手法におけるパケットの迂回動作は、故障検知したノードによるトンネルへの迂回処理と、終端ノードにおけるトンネルパケットのカプセル化解除処理のみから構成される。後者のトンネル終端処理を行う機能は、一般的なノード装置に備わっているため、この手法の配備には、前者のトンネルへの迂回機能の導入さえ行えば良い。それゆえ、この手法は事前計算型経路更新手法よりも、既存のネットワークへの導入は容易である。また、代替経路表を格納するために必要なメモリ量およびその計算コストを、事前計算型経路更新手法と比較して、より少なくできることを示した。

故障発生時における復旧時間短縮のために課題となっていた、経路表更新時間の削減を事前計算型経路更新手法で、また故障通知時間の削減を故障箇所迂回手法で、それぞれ実現した。これらの成果を活用することにより、IP ネットワークの可用性を従来よりも飛躍的に向上させることが可能であると考えている。

目次

図目次

表目次

第1章　序論

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、広帯域化、低価格化が進み、その上で多くの人による様々なアプリケーションの利用が可能となっている。そのため、既存の電話網 (PSTN : Public Switched Telephone Network) が従来果たしていた社会インフラとしての役割は、IP を用いたネットワークによって取って代わられようとしている [総務 05]。

電子メールに代表される IP ネットワーク上で旧来用いられていたアプリケーションは、ネットワークの一時的な停止などに、比較的寛容であった。仮に、ネットワークが一時的に停止しても、復旧後に再送を行うことで、利用者には停止の影響を感じさせることはなかった。しかし、今日 IP ネットワーク上では、ネットワークの停止が利用者に大きな影響を与えるアプリケーションの利用が拡大している。例えば、VoIP に代表されるリアルタイム系のアプリケーションは、短時間であってもネットワークが停止すると、利用者に多大な影響を与える。また金融取引などのビジネスアプリケーションでは、ネットワークの停止が金銭的な損失につながる可能性が高い。IP ネットワークには、そのサービスを停止・中断することなく提供し続ける能力、つまり可用性が、従来以上に高いことが求められるようになってきている。

現状の IP ネットワークを用いた電気通信サービスにおいては、長時間のサービス停止を伴う障害が毎年多数発生している。日本では電気通信事業法において、電気通信役務の提供の停止又は品質低下を受けた利用者数が 3 万人以上、もしくは停止時間もしくは品質低下の時間が 2 時間以上に及ぶ重大事故は、総務省への報告が義務付けられている。総務省により毎年公表されている重大事故の件数は、2003 年度の 7 件から 2009 年度には 18 件に増加している [総務 10]。さらに 2009 年度には 1000 万人以上に影響を与える大きな事故も発生している。

このような状況に対し、総務省の諮問を受けた情報通信審議会において、ネッ

トワークの IP 化に対応した安全・信頼性対策に関する答申 [情報 07] がまとめられている。その中でネットワーク設備の運用・管理面についての課題の一つとして、「IP ネットワークの早期異常検知機能等の設備監視技術と予備系装置への自律切替などの研究開発」が挙げられており、この分野の技術開発が望まれている。

上記の背景を踏まえ、本研究は IP ネットワークの可用性を向上させることを目的とする。IP ネットワークにおいては、予め冗長に構成されたノードやリンクを、故障発生時に切り替えて使用するという手段で、障害復旧が行われている。可用性向上のためには、この障害からの復旧時間をより短くする必要がある。しかし、これまでに提案されてきた手法ではネットワーク中の各ノードへの故障通知や経路の再計算にはネットワークの規模に応じた時間がかかるため、大規模なネットワークでは切替時間が長くなるという問題があった。

本研究では、故障により影響を受ける範囲の局所性に着目し、この問題の解決を目指す。あるノードが、故障によって影響を受けるか否かは、故障箇所からの距離に関係すると推測される。たとえば、故障箇所から遠いノードは、故障による経路更新を行わなくても、パケットの到達性に影響を与えない可能性が高い。一方で故障箇所に近いノードが経路更新を行わなかった場合、パケットは正しく転送されず、その宛先まで到達しない可能性が高くなると考えられる。この局所性をうまく活用することで、ネットワークの規模に関わらず、ほぼ一定の時間で故障発生時における切替時間を短縮することを考える。

具体的には、まず故障がパケット到達性に影響を与える条件について形式化を行う。さらに、故障箇所との距離がパケット転送に与える影響を調べる。

次に、この解析結果を元にした経路変更箇所の局所化アルゴリズムの提案を行う。このアルゴリズムは、故障箇所に対して、パケット到達性を維持するために経路更新を行うべきノードを特定する。各ノードが自律分散的に動作する IP ネットワークでは、各ノードがそれぞれ分散計算を行うことが求められる。このアルゴリズムは自身を基点とする三種類の最短パスツリーのみを用いることで計算可能であり、各ノードにおける分散計算を可能としている。

さらに、局所化アルゴリズムを使った高速復旧手法を二つ提案する。一つ目の事前計算型経路更新手法では、故障発生時に使用する代替経路表を事前に計算しておくことにより、復旧時間の短縮を実現する。一般に、代替経路表は故障箇所毎

に異なるため、大規模なネットワークでは非常に多くの代替経路表を事前に計算する必要がある。これに対して、提案手法では、局所化アルゴリズムを使用することでこの代替経路表数を実装可能な数まで抑える。また二つ目の故障箇所高速迂回手法は、故障を検知したノードが、故障の影響を受けないノードまでトンネリングを用いることで、パケットを迂回させる。この手法では、検知ノードを除く各ノードは自身の経路表を更新する必要がないため、検知ノードからの故障通知を必要としない。そのため、高速迂回手法は、事前計算型経路更新手法において故障通知にかかっていた時間の短縮を実現する。この高速迂回手法では、局所化アルゴリズムを用いることで故障の影響を受けないノードを特定し、そのノードをトンネルの終点として用いる。

本論文の構成は以下の通りである。まず第 2 章では、IP ネットワークの可用性向上に関する従来技術について述べる。次に第 3 章では、故障箇所とパケット到達性の関係について解析を行う。この解析結果を元にして、第 4 章では、経路変更箇所の局所化アルゴリズムの提案を行う。この局所化アルゴリズムを用いた高速復旧手法である事前計算型経路更新手法を第 5 章で、また故障箇所高速迂回手法を第 6 章で、それぞれ提案する。最後に第 7 章で本論文のまとめを行う。

第2章　IP ネットワークの可用性向上に関する従来研究

2.1　背景

従来既存の電話網が果たしてきた社会インフラとしての役割は、IP ネットワークに取って変わられようとしている。このため IP ネットワークには、そのサービスを停止・中断することなく提供し続けることが求められるようになってきている。しかしながら、現在の IP ネットワークを用いた電気通信サービスにおいては、長時間のサービス停止を伴う障害が毎年多数発生している。我が国では、電気通信事業法において、電気通信役務の提供の停止又は品質低下を受けた利用者数が 3 万人以上、もしくは停止時間もしくは品質低下の時間が 2 時間以上に該当する事故は重大事故と位置づけられており、その発生時には総務省への報告が義務付けられている。表 2.1 は、総務省により毎年公表 [総務 10] されている電気通信事業者から報告された事故件数をまとめた表である。表 2.1 を見ると、重大な事故の発生件数は年々増大している。2009 年度における重大な事故に関しては、発生件数は前年度と同等であるが、その中には 1000 万人以上に影響を与える大きな事故も含まれている。

このような状況に対し、総務省の諮問を受けた情報通信審議会において、ネッ

表 2.1: 日本における電気通信事故発生件数の推移

年度	2003	2004	2005	2006	2007	2008	2009
重大な事故	7	7	14	13	11	18	18
その他の事故	32	13	52	56	140	171	177
合計	39	20	66	69	151	189	195

トワークの IP 化に対応した安全・信頼性対策に関する答申 [情報 07] がまとめられている。その中でネットワーク設備の運用・管理面についての課題の一つとして、「IP ネットワークの早期異常検知機能等の設備監視技術と予備系装置への自律切替などの研究開発」が挙げられており、可用性向上を目的とした技術開発が望まれている状況である。

可用性とは、コンピュータシステムがそのサービスを停止・中断することなく提供し続ける能力を示す指標である。本研究対象である IP ネットワークも、指定の宛先までパケットを届けるサービスを提供するシステムと捉えることで、同様に扱うことが可能である。システムの平均故障間隔を $MTBF$ (Mean Time Between Failure) とし、故障からの平均復旧時間を $MTTR$ (Mean Time To Repair) とした場合、システムの可用性は以下の式で定義される [Lal84]。

$$\text{システムの可用性} = \frac{MTBF}{MTBF + MTTR} \tag{2.1}$$

可用性の値を大きくするためには、$MTBF$ の値を大きくするか、もしくは $MTTR$ の値を小さくするかの二つのアプローチが考えられる。

前者の実現のためには、IP ネットワークの構成要素であるリンクやノードの故障率をより低くする必要がある。このため IP 層におけるリンク、ノードを高信頼化するための各種技術が提案、実用化されている。一方で後者に関しては、各宛先までのパケット転送動作を維持するために、予めノードやリンクを冗長に構成しておき、故障発生時にはこれらを切り替えて使用するというアプローチで、経路制御技術と呼ばれる技術が確立されている。図 2.1 に、ネットワーク可用性向上技術の分類を示す。

以下 2.2 で IP ネットワークの概要について説明を行った後、2.3 において、リンクやノードの高信頼化技術について述べ、これらの技術のみを用いての可用性向上には限界があることを説明する。2.4 において、経路制御による復旧技術の詳細について述べ、さらなる可用性向上を実現するためには、この技術による故障発生時における復旧時間の短縮が重要であることを示す。この復旧時間のうち故障通知および経路表更新にかかる時間の削減を目的とした、高速迂回と呼ばれる技術が多数提案されている。この高速迂回技術は、経路制御による復旧技術の一種であるが、別途 2.5 においてその詳細を述べる。最後に 2.6 で研究課題のまとめを行い、本研究の位置づけを明らかにする。

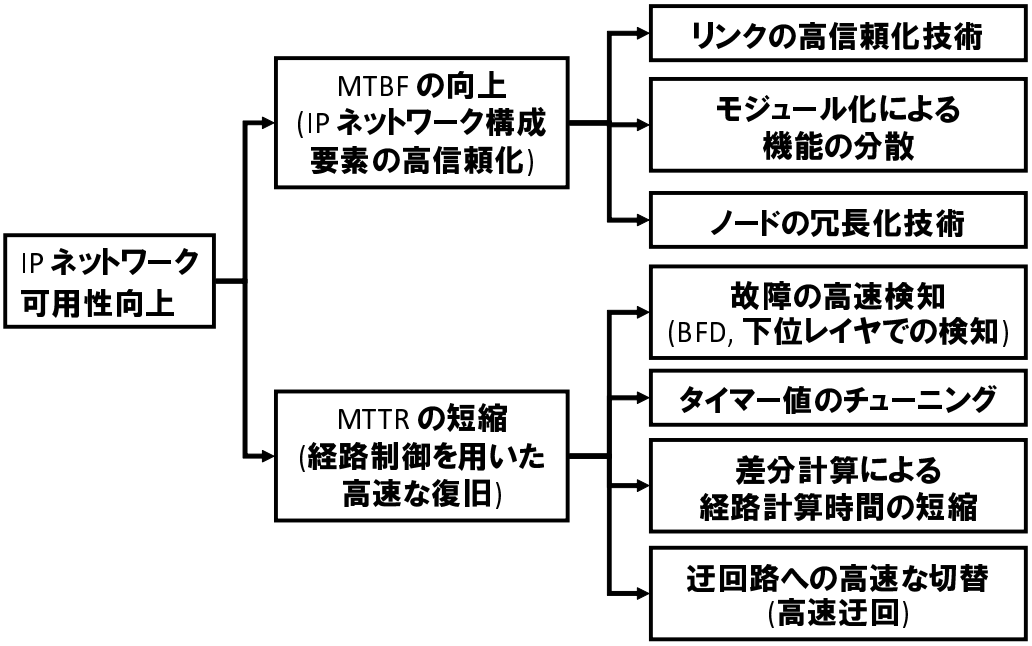

図 2.1: ネットワーク可用性向上技術の分類

2.2 IP ネットワーク概要

IP ネットワーク技術とは、様々な伝送技術により構成される下位層のネットワークを相互に接続する技術である [Tan02]。これらの下位層ネットワークは、IP ネットワーク上においてはリンクとして扱われる。IP ネットワークは、リンクおよびそれらを相互に接続するノードによって構成される。

下位層ネットワークで用いられている伝送技術の違いにより、IP ネットワークにおけるリンクには、以下の二つのタイプが存在する [Tan02]。

ブロードキャスト型： 複数のノードが単一の通信路を共有するネットワーク。このネットワークに対して送信されたパケットは、すべてのノードにて受信される。このときパケット中の宛先アドレスが受信したノード自身宛であれば、そのパケットは処理が行われ、そうでなければ破棄される。

ポイントトゥポイント型： 二つのノードが一対一で接続するネットワーク。

IP ネットワークを構成するノードは、パケット中の宛先アドレスに応じて次転

送先ノードを決定し、決定された次転送先ノードにパケットを転送する。この次転送先ノードの決定に用いられる情報は、経路と呼ばれ、宛先アドレスと次転送先の組として構成されている [Hun02]。各ノードは、ネットワーク中の各宛先毎の経路を、経路表と呼ばれるテーブルに格納している。各宛先へのパケット転送を正しく行うためには、各ノードにそれぞれ経路が正しく登録されている必要がある。IP ネットワークにおけるノードが果たす機能は、以下の二つに整理可能である [Tan02]。

IP パケット転送 (Forwarding) : 経路表に格納された経路に従って、パケット転送を行う機能

経路制御 (Routing) : 各ノード同士がネットワークの状態に関する情報を互いに交換しあうことで、有効な経路を自律的に構築する機能[1]

経路制御により各ノードの経路表が正しく設定されており、宛先までパケットが正しく転送可能であるとき、その宛先までパケット到達性があると言う。また、送信元から宛先までパケットが通過するノード、リンクの並びをパスと呼ぶ。

経路制御を行うためのプロトコルは、目的に応じて各種提案、実用化されている。RIP [RFC2080, RFC2453] は、比較的小規模な IP ネットワークでの運用を目的として開発された経路制御プロトコルである。このプロトコルでは、宛先とその宛先に到るまでの距離 (経由するノード数) の組が記載されたメッセージを交換し、メッセージの受信方向から宛先に至る経路を決定する、距離ベクトル型と呼ばれるアルゴリズムを採用している。ただし、交換すべき情報が多いため、大規模なネットワークへの適用には向いていない。

より規模の大きな IP ネットワークへの適用のために、リンク状態型アルゴリズムを採用した経路制御プロトコルである OSPF [RFC2328, RFC2740] および IS-IS [RFC1195] が提案されている。IS-IS は、国際標準化団体 ISO において策定された OSI ネットワーク層に用いるために開発された経路制御プロトコルであり、後に IP ネットワークへ適用するための拡張が行われた。一方 OSPF は IS-IS を

[1]管理者が人手で有効な経路を設定することを静的経路制御、各ノードの情報交換により自律的に有効な経路を構築することを動的経路制御と、それぞれを区別して呼ぶ場合もある。多くのネットワークでは動的経路制御が用いられているため、本論文ではこの動的経路制御を単に経路制御と呼ぶこととする。

元に、IP ネットワークへの適用を目的として開発された経路制御プロトコルである。共に、リンク状態型アルゴリズムを採用している点など多くの共通点を持ち、現在多くの IP ネットワークで使われている [Per00]。

リンク状態型アルゴリズムで用いられるリンク状態 (Link-state) とは、ネットワーク中の各ノード、リンク間の接続状態を表す情報である。またリンク状態中には、管理者によってリンク毎に定められたメトリックと呼ばれる値 [2] が含まれており、この値が最短パスを選択する基準として用いられる。このアルゴリズムでは、以下の手順により経路を決定する。

1. 各ノードが、自身に接続するリンクに基づいてリンク状態を生成する。
2. 隣接ノード間において、相手が保有しないリンク状態を相互に交換する。
3. 各ノードが、リンク状態を用いて、自身からネットワーク中の各ノードに対する最短パスを計算し、各ノードに対する経路を決定する。

手順 2 が繰り返し行われることで、ネットワーク中の全ノードにおいてすべてのリンク状態が共有される。上記の手順 3 における最短パスとは、ある二ノード間のパスのうち含まれるリンクのメトリックの合計が最小となるパスであり、ダイクストラ法 [Dij59] により計算される。ある宛先までの最短パス上における次のノードが、その宛先への経路における次転送先となる。さらに、故障などによりリンク状態に変化があった場合にも、ネットワーク中の全ノードにより上記の手順が実施される。

ある単一の管理組織によって運用されているネットワークをドメインと呼び、RIP, OSPF, IS-IS はドメイン内で用いられる経路制御プロトコルである。本研究では単一ドメイン内の IP ネットワークにおける可用性向上を目的とする。特に、OSPF もしくは IS-IS などのリンク状態型経路制御プロトコルを用いて運用されているネットワークを研究の対象とする。

[2] コストと呼ばれることもある。

2.3 IP ネットワーク構成要素の高信頼化技術

高信頼化技術の目的は、所定の機能を継続して提供する能力を向上させ、高い MTBF 値を実現することにある。IP ネットワークの構成要素であるリンクやノードもハードウェアである以上、故障発生の可能性を排除することは不可能である。そのため高い信頼性を確保するためには、故障に対する対策が必要になる。

[南谷 91] には、耐故障性向上のための基本原理として、「冗長性」と「分散」の二つが挙げられている。冗長性が確保されていなければ、故障によって損なわれる機能を提供し続けることは不可能である。また、例え冗長性が存在しても、その冗長性がある故障により同時に影響を受けてしまうのでは意味が無い。それゆえ、故障の影響を受ける範囲を限定するために、モジュール化による機能分散も重要となる。

IP ネットワークにおけるノードに対する高信頼化技術が各種提案されている。これら技術もまた耐故障性向上の基本原理に基づいており、冗長化およびモジュール化の二つに分類可能である。ノードを冗長化するための技術については、2.3.1 において説明を行う。また、ノードを構成する各機能要素をモジュール化する技術については、2.3.2 において詳細に説明を行う。

IP ネットワークにおけるリンクに対しては、下位層ネットワークを構成する伝送技術毎にそれぞれ高信頼化技術が提案されている。本研究では IP ネットワークを対象としているため、これらの技術については、ここでは扱わない。

2.3.1 ノードの冗長化技術

IP ネットワークにおけるノード装置を冗長化することによりパケット転送サービスの中断を防ぐ技術として、Hot Standby Router Protocol (HSRP) [RFC2281] や Virtual Router Redundancy Protocol (VRRP) [RFC3768, RFC5798] などの技術が提案され、実用化されている。HSRP は Cisco Systems 社により開発された技術であり、同社製品に実装されている。一方、HSRP をベースに、標準化団体である IETF において策定された技術が VRRP である。これらの技術には、キープアライブパケットのフォーマット、送信方法に違いが存在するが、冗長化実現のための動作に関して大きな違いはない。

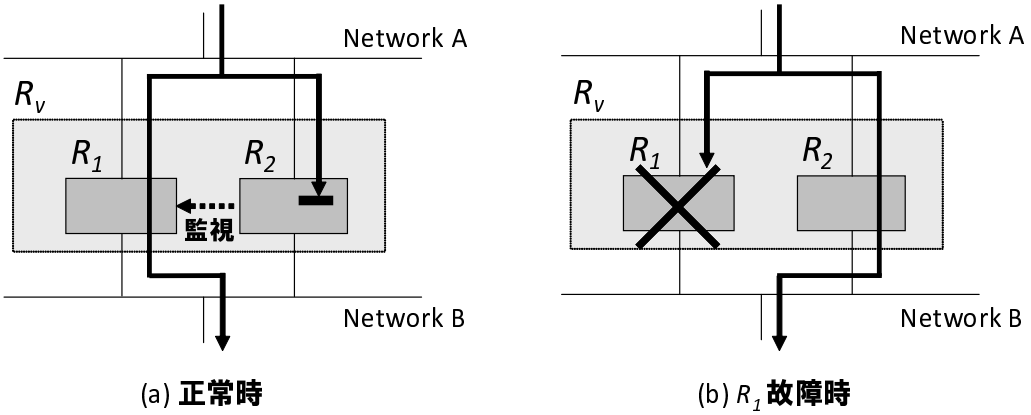

図 2.2: ノード冗長化

これらの技術では、ノード装置が冗長に配置され、一方の故障時に他方がその動作を引き継ぐ。図 2.2 にノード冗長化技術における動作例を示す。ネットワーク A, B の間には、二台のノード R_1, R_2 が配置されており、R_1 がパケット転送を行い、R_2 は予備系として R_1 の動作を監視する。R_1 に故障が発生した場合、R_1 が行っていたパケット転送の動作を R_2 が即座に引き継ぐ。

これらの技術は、パケット転送動作を引き継ぐために、R_1, R_2 が同一の内部状態を保持していることが前提となっている。しかし、ノード上で動的な経路制御を行っている場合や、ファイアウォール機能を動作させている場合には、内部状態はパケット転送動作中に変化する。例えば、出口が二つ以上あるネットワークでは、故障時には有効な出口へパケットが向かうように経路を切り替える必要がある。このために、経路制御プロトコルにより取得した情報を元に、パケット転送に用いる経路をネットワークの状態に応じて生成するという動作が行われている。またファイアウォール機能は、通過するパケットがどのフローに属するのかを識別し、フローを構成する正規のパケットのみを通過させるという動作を行っており、通過するフロー状態を内部状態として持っている。これらの機能を使用する場合、冗長ノードを構成する各ノード間で内部状態の同期を行う必要がある。

しかし、内部状態の同期を行うための仕組みは HSRP や VRRP には用意されていない。これらの目的に応じて、内部情報を同期する仕組みが各種提案されている。[鈴木 04] では、冗長ノードを構成する二つのノード間で経路情報を共有す

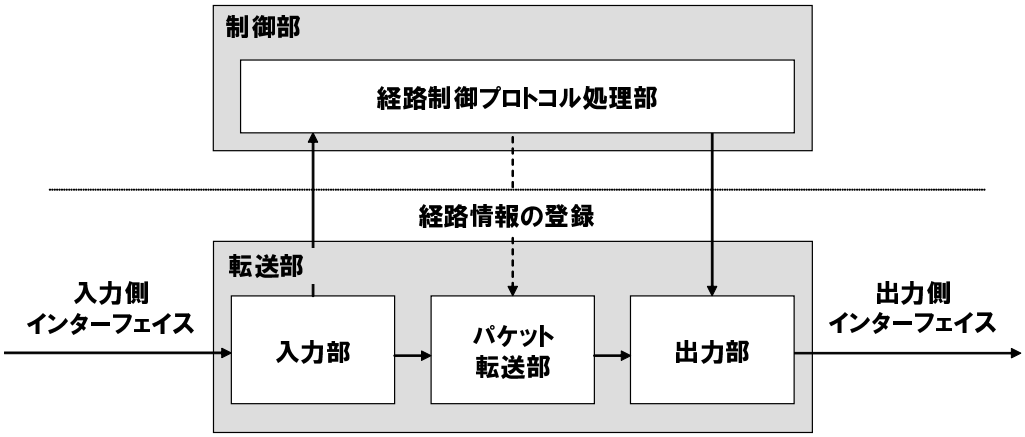

図 2.3: 制御/転送分離型アーキテクチャ

る機構の提案が行われている。また [狩野 05] では、冗長ノードを構成する複数のノード間でフロー状態を共有する機構が提案されている。

2.3.2 モジュール化による機能の分散

近年のノードには、トラフィックの増大に対して、パケット転送能力の向上が求められてきた。パケット転送機能を高速化するために、専用のハードウェアが開発され、使用されている [Cro00, Dix06]。また、経路制御機能および装置自体の管理機能など、パケット転送以外の機能を実現するために、別途汎用のプロセッサが搭載されている。図 2.3 に示すように、転送部と制御部をそれぞれ独立したモジュールとして構成することが、ノードのアーキテクチャとして現在主流となっている [RFC3746]。

各機能要素毎にモジュール化することは、ノードの高信頼化にも貢献している。一般の運用では、設定変更や機能追加に伴うファームウェアのアップデートなどの理由により、ノードの再起動を必要とするケースが存在する。このとき、ノードが機能要素毎にモジュール化されていれば、転送部を動作させたまま、制御部だけ再起動を行うことが可能となる。しかし、死活監視のために送られるキープアライブパケットに対する応答は制御部で行われるのが一般的である。このため、制御部が停止すると、隣接ノードからはノード全体がダウンしたと判断されてしまう。このことを防ぐた

めには、制御部が一時的に停止するが転送部は継続動作することを、隣接ノードに予め伝えておけばよい。このための手順は一般にグレースフルリスタート手順と呼ばれ、各経路制御プロトコル毎にそれぞれ標準化 [RFC3623, RFC4724, RFC5306] がなされている。またモジュール化された制御部を同一ノード内で複数持ち、それらの間で現用、予備の冗長系を構成することで、ノードの信頼性を向上させる技術の提案もなされている [沖田 04, 関根 05]。

2.3.3 高信頼化による可用性向上の限界

高信頼化技術を活用することで、ネットワークの可用性をある程度向上することが可能であるが、それには限界がある。

ノード冗長化技術において待機系のノードは、正常系ノードの動作を引き継ぐために、ネットワーク内におけるリンク接続状態は正常系ノードと同一である必要がある。このことから一般的に、正常系、待機系のノードは、物理的に比較的近い場所に配置されている。それゆえ、両ノードが配置されているビル設備の障害により停電などが発生した場合には、両ノードが同時に停止することとなり、その目的を果たすことができない。

以上の理由から、高い可用性を必要とする IP ネットワークの運用では、本節で説明した高信頼化技術と次節以降に説明を行う経路制御技術を組み合わせて使用することが一般的である。

2.4 経路制御技術を用いた故障からの復旧

ネットワーク内のノードやリンクに故障が発生した場合には、経路制御技術により、故障箇所を使用しないように経路が再構成されることで、パケット転送動作が継続される。故障からの復旧とは、故障発生後各ノードが経路表更新などの処理を行い、ネットワーク中の任意の二ノード間におけるパケット到達性が全て復元された状態を指すこととする。

経路制御による復旧動作は、具体的には以下の各フェーズから構成される。

故障の検知 ノードが、自身に接続するリンクや隣接するノードの故障を検知する。

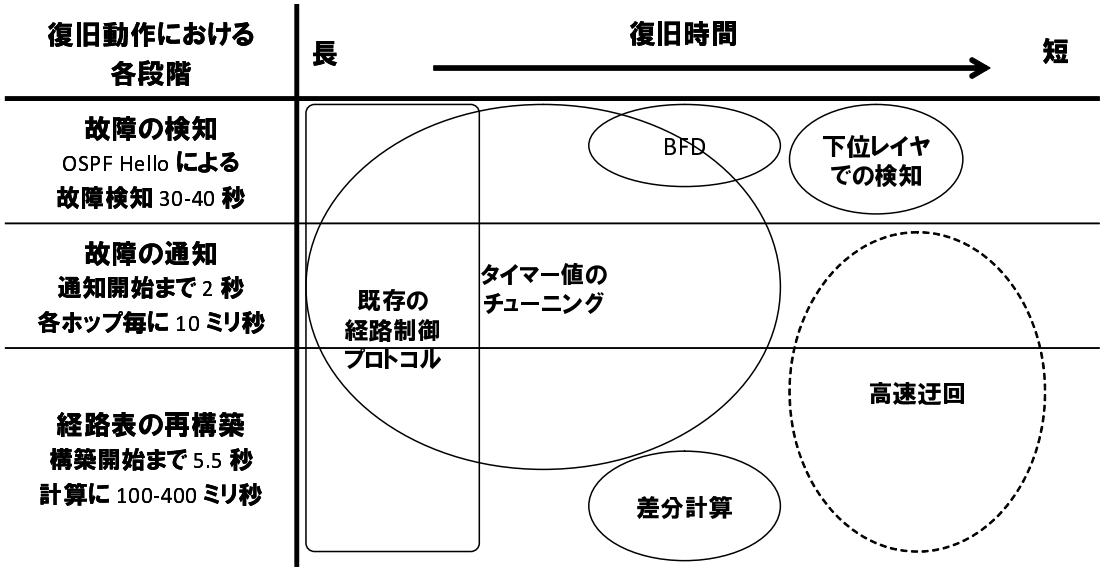

図 2.4: 切替動作各フェーズと切替時間短縮技術

故障の通知 検知した故障を、ネットワーク中の全ノードへ通知する。

経路表の再構築 検知もしくは通知された故障に基づき、新たな経路の計算を行い、経路表を更新する。

これらの各フェーズに費やす時間の合計が MTTR であるため、IP ネットワークの可用性を向上させるためには、これらの時間をより短くすることが重要である。[Ian04] では、ある商用ネットワークにおける MTTR についての調査結果が報告されている。この文献によれば、故障の通知にはホップ毎に 10 ミリ秒程度、経路計算には 100 ミリ秒から 400 ミリ秒程度の時間をそれぞれ要している。また、経路表の更新には 1 経路あたり 20 ミリ秒の時間がかかることが報告されている。

図 2.4 に復旧動作における各フェーズと、それぞれのフェーズが要する時間を短縮するための各種技術の関係を示す。これらの復旧時間短縮技術の詳細については、以降の節において述べる。

2.4.1 故障の高速検知

故障の検知技術には主に以下の二つのメカニズムがあり、商用ネットワークではこの双方が利用されている [Ian04]。

- 下位層での検知
- キープアライブメッセージの使用

ノード同士が直接ポイントトゥポイント型のリンクで接続している場合には、下位層における接続情報を用いることで、短時間での故障検知を実現できる。例えば IP ネットワークの下位層ネットワークの一つとして用いられている SONET/SDH [Gor00] では、リンクの故障により隣接ノードとの接続状態が失われたときに 10–20 ミリ秒で上位層にアラームを上げることができ [Ian04, Fra05b]、これらの情報を用いることで、高速な故障検知を実現できる。

下位層の検知技術とは別に、各経路制御プロトコルはそれぞれ、自身に接続するリンクや隣接ノードの故障を検知するメカニズムをそれぞれ持っている。具体的には隣接するノード同士で定期的にキープアライブパケットを相互に送りあい、ある一定時間相手からのキープアライブパケットが届かなかった場合、隣接ノードが故障したと判断する。各経路制御プロトコルにおいて規定されているキープアライブパケットの送信間隔は、数十秒と比較的長い値となっている。例えば、OSPF では、Hello と呼ばれるキープアライブパケットが用いられており、10 秒間隔で送られる Hello が 4 回連続受信されなかった場合、隣接ノードが故障したと判断される。つまり、実際に隣接ノードが故障してから、故障を検知するまでに 30-40 秒の時間を要する。より短時間での故障検知を実現するために、経路制御プロトコルとは独立し、キープアライブに特化した専用のプロトコル Bidirectional Forwarding Detection (BFD) [RFC5880] が提案され、実用化されている。このプロトコルは、数十ミリ秒程度の短い時間でのキープアライブメッセージの送信が可能なように仕様が設計されている。各経路制御プロトコル毎の故障検知メカニズムの代わりに、この BFD を用いることで、より高速な故障検知が実現可能である。

表 2.2: 故障復旧フェーズにおける主なタイマー [Ian04]

	説明	値	
		デフォルト	最小
Notification timer	故障検知後、通知を開始するまでのタイマー	2 s	10 ms
LSP generation timer	故障検知後もしくは通知受信後、他ノードへの通知のためメッセージである LSP (Link-State Packet) の作成を開始するまでのタイマー	50 ms	1 ms
Computiation timer	通知受信後、経路計算を開始するまでのタイマー	5.5 s	1 ms

2.4.2 タイマー値のチューニング

前述の故障復旧における各フェーズ間には、動作を安定化させるための各種タイマーが用いられている [Ian04]。例えば、リンクのアップ、ダウンが短期間に繰り返されるようなリンク故障に対してその都度通知を行った場合、通知を処理する各ノードの負荷が高くなってしまう。このことを防ぐ目的で、設定された時間までの残り時間を管理するための機能であるタイマーを、次のように用いる。リンク故障検知時にタイマー (Notification timer) の動作を開始させ、タイマーの残り時間が無くなるまでにそのリンクが復旧しなかった場合にのみ、検知した故障を通知する。この他に用いられている主なタイマーを、表 2.2 に示す。タイマー値の大きさは、復旧時間に影響を与えるが、必要以上に小さいと動作を安定化をさせるという本来の目的を損なうこととなる。しかし、適切なタイマー値を決めることは難しいため、実運用では保守的な値であるデフォルト値がそのまま用いられることが多かった。このことが全体の復旧時間を必要以上に長くする原因となっていた。

Francois らは、これらタイマー値のチューニングを行うことで、復旧時間の短縮を実現している [Fra05b]。タイマー値のチューニングは無駄な待ち時間の削減

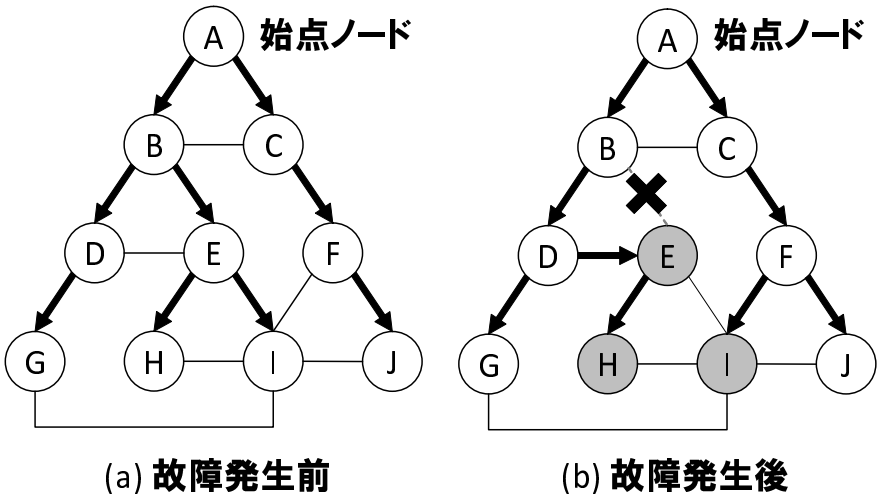

図 2.5: Incremantal SPF

を実現できるが、依然として故障の通知時間や経路の再計算時間にはある程度の時間を必要とする。

2.4.3 差分計算による経路計算時間の短縮

OSPF や IS-IS などのリンク状態型経路制御プロトコルでは、ダイクストラ法 [Dij59] によって得られる最短パスツリーを元に、各ノード宛の経路を決定している。故障発生時には、故障後の使用する経路決定のために、新たなトポロジーを元に再度最短パスツリーを構築する必要がある。ダイクストラ法の計算コストは、ノード数、リンク数をそれぞれ N, L としたとき、$O((N+L) \times logN)$ のオーダーで表される [Bar98]。つまり、ノード数とリンク数で表されるネットワークの規模に応じて、計算コストが大きくなり、故障時の復旧時間が長くなる原因となっていた。しかし、故障前の最短パスツリーに対して、故障後の差分のみを再構築するという計算法 (Incremental SPF) が提案されており [McQ80]、この手法を用いることで故障時における経路再計算時間を短縮することができる。

例えば図 2.5 のネットワークにおいて、ノード B–E 間のリンクに故障が発生した場合について考える。このとき、影響をうけるノードは、このリンクを経由して始点ノード A と接続しているノード E, H およびノード I のみである (図 2.5 (a)

参照)。これらのノードを除く他のノードへの、始点ノード A からの最短パスは故障前後で変わらない。そのため、Incremental SPF ではノード E, H およびノード I に関わる部分だけ、最短パスツリーの再構築を行う。

図 2.5 (b) の故障発生後のネットワークはノード、リンクの総数がそれぞれ 10, 14 である。一方、Incremental SPF を用いた場合の計算対象となるノード数は 3 であり、これらのノードに接続するリンク数が 7 である。この場合、最短パスツリーの構築を最初から行った場合と比べ、ノード数、リンク数はそれぞれ 3/10, 1/2 となり、Incremental SPF の計算コストは約 19 パーセントにまで削減できる。

2.5 高速迂回を用いた復旧手法

前節では故障時における経路の復旧時間を短縮する各種技術について説明を行ったが、故障の通知や経路表の再構築にかかる時間の短縮という課題が依然として残っていた。予め設定したパスに沿ってパケットの転送を行うネットワーク技術である MPLS [RFC3031] では、この課題を解決する手法として、高速迂回 (Fast Reroute) と呼ばれる手法 [RFC4090] が実用化されている。この手法では、故障を検知したノードが即座に、パケットを代替パスへ迂回させる。この代替パスを事前に設定しておくことで、故障通知やパスの再計算などの動作を故障後に行うことなく、パケットの到達性を復旧させることが可能となる。

予め設定したパスに沿ってパケット転送を行うネットワーク技術である MPLS では、パケット転送を開始するノードがパスを選択することで、通常時と故障発生時で異なるパスでパケット転送を行うことが可能である。一方で、IP 転送ではパケットの宛先のみを用いて転送を行っているため、複数のパスを用意しておいて使い分けるといった手法を用いることはできない。つまり、 MPLS とはパケット転送方式が異なる IP ネットワークに対しては、上記の高速迂回手法をそのまま適用することは出来ないしかし、この手法は故障発生時の短時間での復旧のためには非常に有効であるため、この手法を IP ネットワークに適用するための研究が進められている [Wan07, Kva06, Sha10, Li 09]。

このような状況に伴い、IP ネットワーク分野における標準化団体である IETF では、研究開発のためのフレームワークの策定が行われた [RFC5714]。このフレー

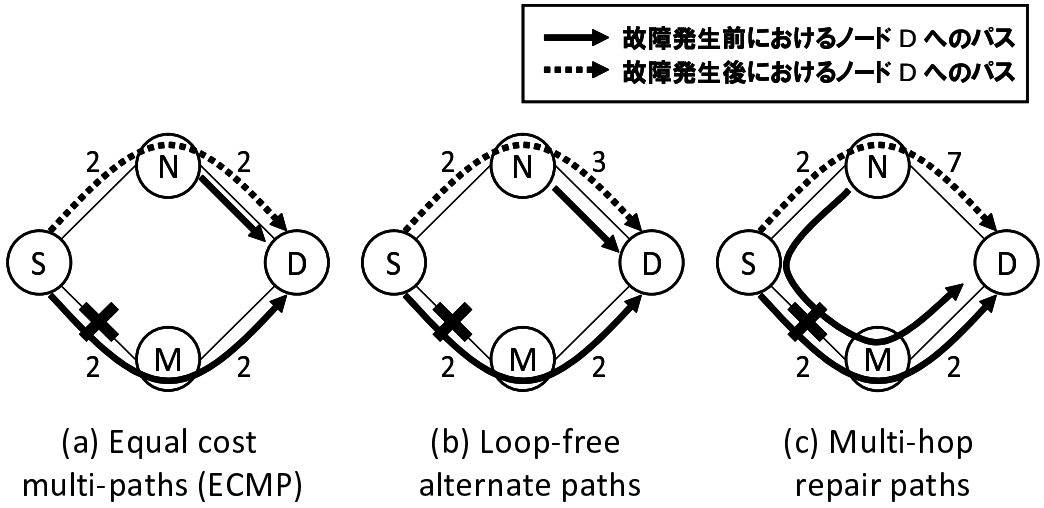

図 2.6: 代替パスの状態

ムワーク中では、各種用語の定義や解決すべき問題の分析などが行われている。故障発生時におけるパケットの迂回方法は、宛先まで至る代替パスの状態に応じて異なるため、このフレームワーク中では代替パスの状態を以下の三つにケースに分類している。

(a). Equal cost multi-paths(ECMP) : 宛先までの最短パスが複数あり、その中に故障の影響を受けない代替パスが存在する場合

(b). Loop-free alternate paths : 代替パスにおける次転送先が経路更新を行わなくても、ループが発生しない場合

(c). Multi-hop repair paths : 上記のいずれにも該当しない場合

それぞれのケースにおける代替パスの例を、図 2.6 に示す。いずれのケースもノード N を経由するパスが代替パスとなる。図 2.6 (a) における故障発生後のノード D へのパスは、そのメトリックが故障前のパスと同一であるため、代替パスとしてそのまま使用することが可能である。このパスは通常の最短パスツリーの計算時に算出することが可能である。

一方でケース (a) に相当する代替パスが存在しない場合には、別途計算により代替パスを求め、その代替パスが他ノードの経路更新なしに使用可能であるかを

判定する必要がある [RFC5286]。図 2.6 (b) の例では、ノード D 宛のパケットは、ノード S から代替パス上の次転送先であるノード N に転送された場合、ノード N によりノード D に転送されるため、宛先に到達できる。一方、図 2.6 (c) の場合、ノード N からノード S, M を経由するパスが、故障前におけるノード D への最短パスである。つまり、ノード S–M 間のリンク故障に伴う経路更新をノード N が行わなければ、ノード S とノード N との間でループが発生する。そのため、ノード N を経由する代替パスを使用するためには、代替パス上のノードの経路更新が必要となる。ここで、代替パス上のノードにおける経路更新を行わなくてもループが発生しない条件は、以下のように一般化することができる。

- 迂回を行うノード S がパケットを次転送先ノード N に迂回させたとき、N から宛先ノード D への最短パス上に S が存在しないこと。

ノード A, B 間の最短パスにおけるメトリックを $Metric(A, B)$ と表す。このとき、上記の条件は次の式で表される。

$$Metric(N, D) < Metric(N, S) + Metric(S, D) \tag{2.2}$$

この式を満たす代替パスは、ケース (b) に相当し、代替パス上のノードが経路更新を行わなくても利用することができる。[鈴木 08a] では、代替パスの計算と同時に行うことで、式 (2.2) の判定を効率的に行うためのアルゴリズムが提案されている。

ケース (c) については、代替パス上の各ノードもなんらかの対処を行う必要があるため、通常であればそれらのノードへ故障の通知を行う必要がある。この故障通知にかかる時間の短縮を目的として、FIR 手法, MRC 手法および Not-Via アドレスを用いた手法 [Sha10] などが提案されている。これらの各手法については、次節以降で説明する。また 4 章では、式 (2.2) をケース (c) に適用するための拡張、および拡張条件を分散判定するアルゴリズムの提案が行われている。

2.5.1 Failure Insensitive Routing (FIR)

Failure Insensitive Routing (FIR) 手法 [Lee04, Nel07, Wan07] は、パケットがどのインターフェイス経由で到達したかに基づいて、次転送先を決定するという手

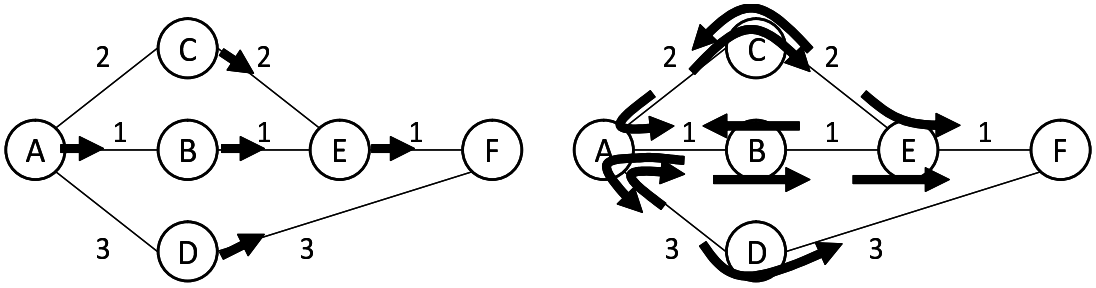

図 2.7: 宛先ノード F への経路の違い

故障前経路												
node	A			B		C		D		E		
prev	B	C	D	A	E	A	E	A	F	B	C	F
next	D	B	B	E	A	E	A	F	-	F	F	-
ノード B–E 間リンク故障後経路												
node	A			B		C		D		E		
prev	B	C	D	A	E	A	E	A	F	B	C	F
next	D	B	B	**A**	-	E	A	F	-	F	F	-

図 2.8: FIR における宛先ノード F への経路表

法である。一般の IP における転送ではパケットの宛先のみで次転送先ノードを決定しているのに対し、この方式では宛先とパケットの入力インターフェイス毎の組で次転送先を決定している。図 2.7 に、ネットワーク中の各ノードにおける宛先ノード F への経路に関して、通常の IP 転送と FIR 手法との違いを示す。この図中の数値は、各リンクのメトリックを表す。また、図 2.7 (b) における各ノードが保有する経路表中の宛先ノード F への各経路を図 2.8 に示す。

通常の IP 転送においてノード B は、ノード F 宛のパケットをノード E に転送する (図 2.7 (a) 参照)。このネットワークにおいて、ノード B–E 間のリンクに故障が発生した場合、故障を検知したノード B は、ノード F 宛のパケットをノード A に転送する。しかし、ノード A は、故障に伴う経路更新を行わなかった場

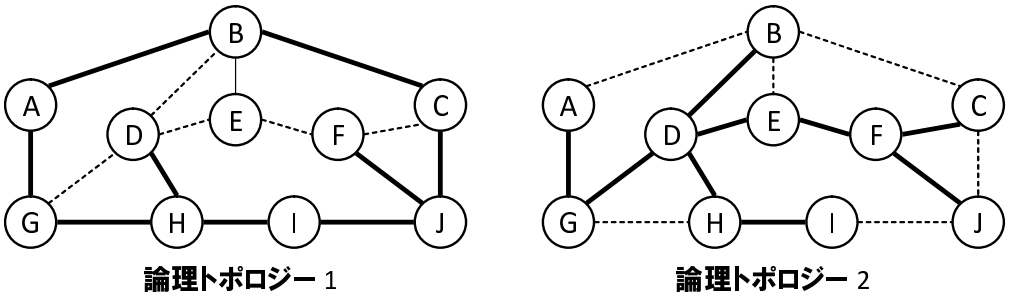

図 2.9: MRC における論理トポロジー

合、ノード F 宛のパケットをノード B に転送する。このため、通常の IP 転送において、故障発生後のノード B からノード F へのパケット到達性復旧には、ノード A の経路更新が完了するのを待たなければならない。

次に、FIR 手法を用いた場合について、図 2.7 (b) を用いて説明を行う。ノード B–E 間のリンクに故障が発生した場合、検知したノード B は、ノード F 宛のパケットをノード A に転送する。ノード F 宛のパケットをノード B から受信したノード A は、ノード B–E 間のリンクもしくはノード E–F 間のリンクが故障していると推測できる。図 2.8 を参照すると、ノード A がノード B から受信した時の次転送先はノード D となっているが、この経路はこれらの推測に基づいて事前に計算されている。

この手法の利点は、明示的な故障通知を行う必要がないという点にある。しかし、通常の IP とは異なる転送方式を採用しており、ネットワークの全ノードがこの方式に対応している必要がある。つまり、既存のネットワークへの段階的な導入が難しいことが、この方式の欠点である。

2.5.2 Multiple Router Configurations (MRC)

Multiple Router Configurations (MRC) 手法 [Kva06, Kva09] は、物理トポロジー上に複数の論理トポロジーを構成し、故障発生などの状況に応じてパケット転送に使用するトポロジーを変える手法である。

図 2.9 は、物理トポロジー上に二つの論理トポロジーが構成された例を示している。この二つの論理トポロジーは、なるべく共通となるリンクを持たず、かつ

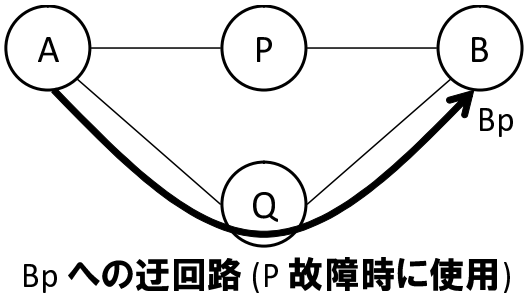

図 2.10: Not-Via アドレスを用いた迂回路

それぞれが連結グラフとなるように構成される。例えば、図 2.9 において、ノード D–E 間のリンクに故障が発生した場合、このリンクを含まない論理トポロジー 1 を用いてパケット転送が行われる。故障検知したノード D は、論理トポロジー 1 を用いるよう、パケットにマーキングを行う。以降、マーキングされたパケットを受け取った各ノードは、論理トポロジー 1 に基づいて計算された経路表を用いてパケット転送を行う。

この手法を有効に活用するためには、論理トポロジーをいかに構成するかが鍵となる。そのため、論理トポロジーを生成するアルゴリズムに関して各種提案が行われている。[Han07] では複数の故障への対応を考慮したトポロジーを作成するためのアルゴリズムの提案が行われている。[Kva07] では迂回パス長の削減を、また [Cic08b, Cic09, Kam09, Kam10b] では論理トポロジー数の削減を、それぞれ目的とした論理トポロジー生成アルゴリズムの提案が行われている。

この手法ではパケットへのマーキングを行うことで、明示的な故障通知を行うことなく、他ノードへ暗に故障の存在を通知している。本手法は、パケットのマーキング状況に応じて使用するトポロジーを変えており、やはり通常の IP とは異なる転送方式が用いられている。

2.5.3 Not-Via アドレスを用いた IP 高速迂回

Not-Via アドレスを用いた手法 [Sha10] は、故障箇所の迂回パスを構成するために、専用のアドレス (Not-Via アドレス) を導入する手法である。

図 2.10 中において、ノード A からノード B 宛のパケットの最短パス上のノードであるノード P が故障した場合を考える。ノード B には通常のアドレスの他に、” ノード P を経由せずにノード B へと至るパス” を構築する目的のみで使用する Not-Via アドレス Bp を割り当てておく。各ノードは、このアドレス Bp 宛の経路を、ノード P が故障により存在しないという前提で、予め計算しておく。ノード A はノード P の故障を検知した場合、次転送先がノード B であるパケットに対しカプセル化を行ない、そのパケットの宛先をノード Bp として送信する。アドレス Bp 宛のカプセル化されたパケットは、ノード P を迂回してノード B に至る。

この手法は、IP の転送方式を変える必要がなく、また原理的に全ての単一故障に対応することが可能である。しかし、ノード故障と Not-Via アドレスとの対応や、Not-Via アドレスに対してどのノードが経路を用意すべきかなど、事前に設定を行うべき項目が多く、この点が自律運用を基本とする IP ネットワークへの適用を困難としている。

2.5.4 各手法の課題

前節までで IP 高速迂回を実現するための三手法について述べてきたが、これらの手法は実用化され広く使われるまでには至っていない。現状広く使われている IP ネットワークに適用するためには、以下の二つの条件を満たすことが必要である。

条件 1. 転送方式の変更を必要としないこと： すでに広く用いられている IP ネットワークに適用するためには、転送方式の変更を伴わない手法が求められる。転送方式が異なる機器同士が同一ネットワーク中に共存することはできないため、独自の転送方式を採用する手法は既存のネットワークへの段階的な導入が行えない。

条件 2. 分散処理が可能であること： IP ネットワークは、ノードやリンクの増設、故障などのネットワークの状態の変化に追随するために、各ノードの自律動作により運用可能なように設計されており、各ノードがそれぞれネットワークの状態を把握し、自身の経路表を各自計算を行っている。仮に管理サーバな

表 2.3: 各手法の比較

	FIR 手法	MRC 手法	Not-Via 手法
条件 1. (転送方式)	× (独自方式)	× (独自方式)	○ (IP 転送)
条件 2. (分散処理)	○ (可能)	× (不可)	× (不可)

どにより集中的な計算を行い、その結果をネットワーク中の全ノードに配布する場合、管理サーバと各ノードの間の通信路の確保が別途必要となる。これらのことにより IP 高速迂回手法も分散処理が可能であることが望ましい。

前述の三手法の上記条件への適合状況を、表 2.3 にまとめた。パケットの次転送先の決定に、宛先アドレスに加え、受信インターフェイスを用いる FIR 手法や、パケット中のフラグを用いる MRC 手法は、条件 1 を満たしていない。迂回用経路表作成が分散処理に対応出来ていない MRC 手法や Not-Via 手法は、条件 2 を満たしていない。このように既存の三手法のうち、二つの条件を共に満たす手法は存在しない。

2.6 本研究の位置づけ

今後 IP ネットワークの可用性をさらに向上させるためには、故障発生時における復旧時間を短縮することが不可欠となる。故障の検知、通知および経路表の再構築の三つのフェーズからなる復旧動作のうち故障検知は、2.4.1 に記載の技術により数十ミリ秒での検知が実現されている。このため、復旧時間短縮のためには、故障の通知および経路表の再構築それぞれにかかる時間を短縮する技術の確立が重要となる。この目的のために各種高速迂回手法が提案されているが、前節で示したように、これらの手法をそのまま現状の IP ネットワークに広く適用するには課題があった。よって、本研究では現状の IP ネットワークに広く適用可能な新たな高速迂回手法について検討を行った。

提案にあたり、転送方式の変更を必要としない高速迂回手法を実現するためには、故障発生時の経路更新とパケット到達性の関係を調べる必要があるため、ま

ず故障がパケット到達性に影響を与える条件について調べ、故障箇所との距離がパケット転送にどのように影響を与えるかについて解析した (第 3 章)。

次に、先の解析結果を元に経路更新必要箇所の局所化アルゴリズムを実現した。故障発生時におけるパケット到達性を高速に復旧させるためには、経路更新を必要とするノードを特定することが重要である。これらのノードを特定するための条件を示し、この条件の判定を各ノードにおいて分散処理するためのアルゴリズムを提示した (第 4 章)。

そして、このアルゴリズムを用いた高速復旧手法として、まず事前計算型経路更新手法の提案を行った。この手法では、故障発生時に使用する代替経路表を事前に計算しておくことにより、復旧時間の短縮を実現する。しかし、代替経路表は故障箇所ごとに異なるため、大規模なネットワークでは非常に多くの代替経路表を事前に計算する必要がある。このため本手法では、局所化アルゴリズムを使用することで、この代替経路表数を実装可能な数まで抑える (第 5 章)。

さらに事前計算型経路更新手法では未着手であった、故障通知時間を短縮を実現する復旧手法として、故障箇所高速迂回手法を提案した。この手法では、故障を検知したノードが、故障の影響を受けないノードまでトンネリングを用いることで、パケットを迂回させる。この手法では、検知ノードを除く各ノードは自身の経路表を更新する必要がないため、検知ノードからの故障通知を必要としない。そのため、本手法は従来故障通知にかかっていた時間の短縮を実現する。本手法では、局所化アルゴリズムを用いることで故障の影響を受けないノードを特定し、そのノードをトンネルの終点として用いる (第 6 章)。

関連研究と本論文における各章の内容との関係を、図 2.11 にまとめた。本研究における手法により、故障発生時における高速復旧実現に対し課題となっていた故障通知時間および経路表の再構築時間の短縮を実現した。特にこれらの手法は、IP の転送方式を変更する必要がなく、かつ各ノードによる分散処理が可能であるので、従来の IP ネットワークに広く適用可能である。

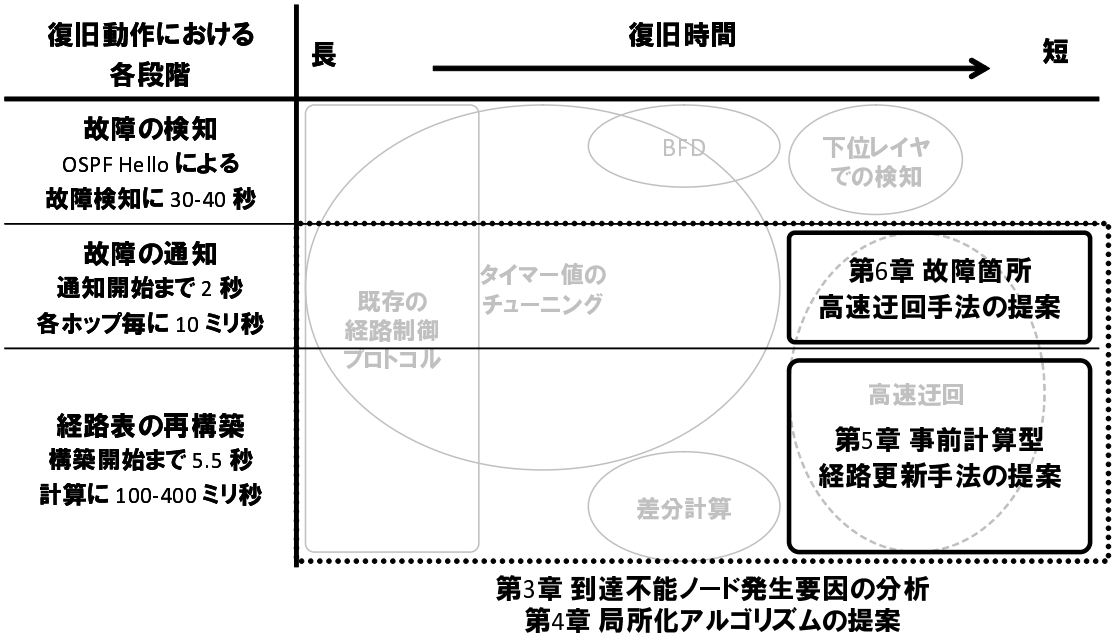

図 2.11: 関連研究と本研究の位置づけ

第3章　経路表矛盾により到達不能ノードが生じる要因の分析

3.1　緒言

現在社会インフラの一部としてその重要性を増している IP ネットワークには、高い可用性が必要とされている。高い可用性を実現するためには故障発生時より短い時間で復旧することが必要であり、高速復旧を実現するための手法が各種提案されている。しかし、これらの手法の多くは、転送方式の変更を必要とするため、既存の IP ネットワークに対してそのまま適用できなかった。そのため、転送方式の変更を必要としない高速復旧手法が求められている。転送方式の変更を伴わない高速復旧手法を実現するためには、故障発生時における既存の転送方式の動作を明らかにすることが重要である。特に、故障に伴う各ノードの経路更新動作にはタイムラグがあるため、この時経路表に矛盾が生じることが知られている。このため、経路表の矛盾がパケットの到達性に影響を与える条件について、明らかにする必要がある。

そこで本章では、ネットワーク中のあるノードのみが故障前の経路表を用い、他のノードは故障後の経路表を用いている状況を想定する。この状況下で、パケットが到達できない宛先ノード (到達不能ノード) が生じる場合、どのような条件により到達不能になるかを明らかにする。さらに、シミュレーションを用いて、各条件毎に到達不能ノード数を調べる。また、故障箇所から遠いノードが故障後に作成した新たな経路表は、その中身が故障前と同一である場合がある。この場合、経路表の再構築を行わなくても到達不能ノードが生じることはない。一般に到達不能ノードの発生は、経路表の矛盾が生じている箇所と故障箇所との距離に関係していると考えられる。故障箇所からの距離と到達不能ノード数との関係についてもシミュレーションを用いて調査を行う。

本章の構成は以下の通りである。まず 3.2 においてネットワークシステムの形式化を行った上で、3.3 において各ノードの経路表間に矛盾が生じた場合に到達不能ノードが生じるための条件を示す。さらに 3.4 では、シミュレーションを用いて、経路表の矛盾が引き起こす到達不能ノード数について定量評価を行う。3.5 では、パケット到達性復旧のためには、到達不能ノード発生条件毎に、どのような対処を行えばよいか考察を行う。最後に 3.6 で本章の内容のまとめを行う。

3.2 ネットワークシステムの形式化

本節では、現実の IP ネットワークにおけるパケット転送を抽象化して議論を行うために、IP ネットワークをネットワークシステムとして形式化する。

定義 3.1 (ネットワークシステム) ネットワークシステム $\mathcal{N} = (\mathbb{V}, \mathbb{E}, \mu)$ は、以下の三項組によって定義される。

1. ノードの集合 $\mathbb{V} = \{n_1, n_2, \ldots, n_p\}$.
2. リンクの集合 $\mathbb{E} = \{e_1, e_2, \ldots, e_q\}$.
3. 関数 $\mu : \mathbb{E} \to \mathbb{N}$.

ここで $\mathbb{N}$ は非負整数の集合である。n_i と n_j 間のリンクはたかだか一つであるとする。n_i と n_j 間のリンクを $e_{i\to j}$ と表記することとする。 □

定義 3.2 (双方向ネットワークシステム) ネットワークシステム $\mathcal{N}$ 中すべてのリンク $e_{j\to i}$ に対して $e_{i\to j}$ が存在し、かつ $\mu(e_{i\to j}) = \mu(e_{j\to i})$ であるとき、$\mathcal{N}$ を双方向ネットワークシステムと呼ぶ。双方向ネットワークシステムを $\widetilde{\mathcal{N}}$ と表記することとする。 □

定義 3.3 (リンク識別子) ネットワーク中のすべてのリンクには、リンク識別子として一意的な番号が割り当てられているものとする。リンク e のリンク識別子は、$\nu(e)$ と表記する。 □

上記で定義されるネットワークシステムとは IP ネットワークを、ノードとは IP ネットワークを構成するルータを、それぞれ表している。また $\mu(e_{i\to j})$ は、リンク $e_{i\to j}$ 毎に管理者によって割り当てられてるメトリックを意味する。リンクのメトリックは、二ノード間の複数パスから実際のパケット転送に用いるベストパスを選択する際に用いられる。具体的には、パス中に含まれるリンクのメトリックの合計が最小となるパスが、ベストパスとして選択される。ベストパスの定義については、後ほど行う。また、リンク識別子は、各リンク (IP 層におけるネットワーク) に割り当てられたネットワークアドレスを表している。議論が複雑になることを避けるために、すべてのネットワークはポイントトゥポイント型であると仮定する。

次に、あるノードが故障した場合のネットワークシステムを表現するために、部分システムを定義する。

定義 3.4 (部分システム) $n_i\,(n_i \in \mathbb{V})$ に接続するリンクの集合を $\mathbb{E}_i = \{e_{i_1}, \ldots, e_{i_j}\}$ $(\mathbb{E}_i \subset \mathbb{E})$ とする。この時、

$$\mathcal{N}_i = (\mathbb{V} - \{n_i\}, \mathbb{E} - \mathbb{E}_i, \mu)$$

は $\mathcal{N} = (\mathbb{V}, \mathbb{E}, \mu)$ の部分システムである。部分システム $\mathcal{N}_i$ を

$$\mathcal{N}_i = \mathcal{N} - \{n_i\}$$

と表記することとする。 □

次に、ネットワークシステムにおけるパケット転送を形式化するために、二ノード間のパスを定義する。

定義 3.5 (隣接ノード) n_i と n_j 間にリンク $e_{i\to j}$ が $\mathcal{N}$ 中に存在するとき, n_i と n_j は互いの隣接ノードであるとする。 □

定義 3.6 (パス) $e_{i_j\to i_{j+1}}$ $(j = 1, 2, \ldots, t-1)$ が存在するとき、

$$P = \langle n_{i_1}, n_{i_2}, \ldots, n_{i_t} \rangle$$

を n_{i_1} から n_{i_t} へのパスと呼ぶ。さらに、n_{i_1} と n_{i_t} をそれぞれ始点ノード、終点ノードと呼ぶ。またこの時 P のパス長を $t-1$ とする。 □

定義 3.7 (パス長) $P = \langle n_{i_1}, \ldots, n_{i_t} \rangle$ としたとき、

$$\delta(P) = t - 1$$

を P のパス長と定義する。 □

定義 3.8 (部分パス) $P = \langle n_{i_1}, n_{i_2}, \ldots, n_{i_t} \rangle$ が存在するとき、

$$Q = \langle n_{i_j}, n_{i_{j+1}}, \ldots, n_{i_k} \rangle \quad (1 \leq j < k \leq t)$$

を P の部分パスと定義する。 □

定義 3.9 (パスのメトリック) $P = \langle n_{i_1}, n_{i_2}, \ldots, n_{i_t} \rangle$ が存在するとき、

$$|P| = \sum_{j=1}^{t-1} \mu(e_{i_j \to i_{j+1}})$$

を P のメトリックと定義する。 □

定義 3.10 (逆方向パス) $P = \langle n_{i_1}, n_{i_2}, \ldots, n_{i_t} \rangle$ が存在するとき、

$$\overline{P} = \langle n_{i_t}, n_{i_{t-1}}, \ldots, n_{i_1} \rangle$$

を P の逆方向パスと定義する。 □

ネットワークシステムが双方向である場合、パス P とその逆方向パスである $\overline{P}$ それぞれのメトリックは、定義 3.2 および 3.9 により、同一の値となる。

定義 3.11 (ベストパス) 以下の手続きにより決定される P を、n_i から n_j へのベストパスと定義する。

Step 1. $\mathbb{P}'$ を、n_i から n_j への全てのパスのうち、最小のメトリックを持つパスの集合とする。また $\mathbb{P}'$ の要素数を $|\mathbb{P}'|$ と表記する。$|\mathbb{P}'| = 1$ であるとき、$\mathbb{P}'$ 中の唯一の要素である P を n_i から n_j へのベストパスとする。

Step 2. $|\mathbb{P}'| \neq 1$ であるとき、$\mathbb{P}'$ 中で $\delta(P)$ が最短となるパスの集合を $\mathbb{P}''$ とする。$|\mathbb{P}''| = 1$ であるとき、$\mathbb{P}''$ 中の唯一の要素である P を n_i から n_j へのベストパスとする。

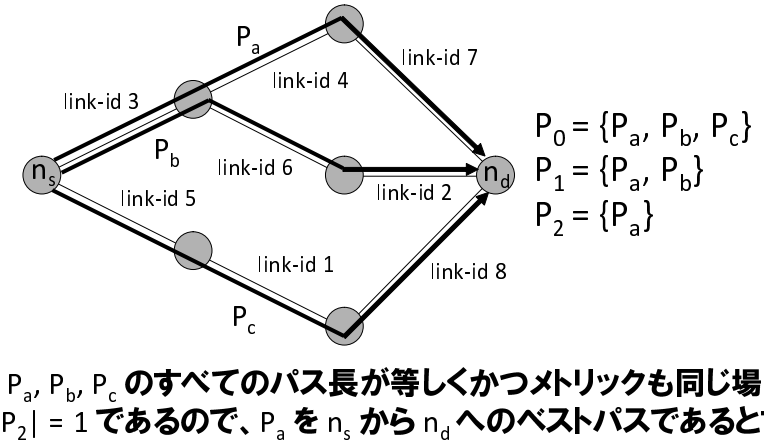

図 3.1: リンク識別子の比較手順

Step 3. $|\mathbb{P}''| \neq 1$ であるとき、以下の手続き *(*図. *3.1* 参照*)* において決定される $\mathbb{P}$ 中の唯一の要素 P を n_i から n_j へのベストパスとする。

```
1:  ℙ ⇐ ℙ''
2:  k ⇐ 1
3:  while |ℙ| ≠ 1 do
4:    ℙ ⇐ MIN(k, ℙ)
5:    k ⇐ k + 1
6:  end
```

$MIN(k, \mathbb{P})$ は、 $\mathbb{P}$ 中のパスのうち、k 番目のリンクを比較し、最小のリンク識別子をもつパスを返す関数であるとする。

□

すべてのリンクは一意に決まる識別子をもっているので、上記の手続きにおいて必ずベストパスが一意に定まる。ネットワークシステムにおいて、各パケットはベストパスを通って、宛先まで転送されるとする。

定理 3.1 ベストパス $P = \langle n_{i_1}, \ldots, n_{i_t} \rangle$ が存在し、その部分パスが

$$P_1 = \langle n_{i_1}, n_{i_2}, \ldots, n_{i_j} \rangle \quad (1 < j \leq t-1)$$

であるとする。このとき P_1 は n_{i_1} から n_{i_j} へのベストパスである。

証明 3.1 $P_1' \neq P_1$ かつ n_{i_1} から n_{i_j} へのベストパスとなるような P_1' が存在すると仮定する。このとき P がベストパスであるという前提との矛盾を導く。

$j = 2$ であるとき、P_1' は $P_1' = \langle n_{i_1}, n_{i_2} \rangle$ と表せる。これは、n_{i_1} から n_{i_2} へのリンクは高々一つしか存在しないという定義 *3.1* から、$P_1 \neq P_1'$ と矛盾する。つまり P_1' が存在する可能性があるのは、$j > 3$ の場合のみである。

$P_2 = \langle n_{i_j}, \cdots, n_{i_t} \rangle$ となる P_2 を P の部分パスであるとする。定義 *3.8* と *3.9* から以下の式が成り立つ。

$$|P| = |P_1| + |P_2| \tag{3.1}$$

$|P_1| \geq |P_1'|$ であるため、式 *(3.1)* から以下の式が導かれる。

$$|P| \geq |P_1'| + |P_2| \tag{3.2}$$

P はベストパスであるとの前提から、式 *(3.2)* は、$|P| = |P_1'| + |P_2|$ であるときのみ成り立つ。このことから以下の式が成り立つ。

$$|P_1'| = |P_1| \tag{3.3}$$

定義 *3.7* と *3.8* から、以下の式が導かれる。

$$\delta(P) = \delta(P_1) + \delta(P_2) \tag{3.4}$$

P_1' がベストパスであり、また式 *(3.3)* が成り立つため、定義 *3.11* から $\delta(P_1) \geq \delta(P_1')$ が導かれる。それゆえ、次の式が成り立つ。

$$\delta(P) \geq \delta(P_1') + \delta(P_2) \tag{3.5}$$

P はベストパスであるとの前提から、式 *(3.5)* は、$\delta(P) = \delta(P_1') + \delta(P_2)$ であるときのみ成り立つ。故に、以下の式が成り立つ。

$$\delta(P_1') = \delta(P_1) \tag{3.6}$$

$j > 3$ であるので、 P_1 と P_1' はそれぞれ次のように表記される。

$$\begin{aligned} P_1 &= \langle n_{i_1}, \ldots, n_{i_{k-1}}, n_{i_k}, \ldots, n_{i_j} \rangle \\ P_1' &= \langle n_{i_1}, \ldots, n_{i_{k-1}}, n_{i'_k}, \ldots, n_{i_j} \rangle \quad (n_{i_k} \neq n_{i'_k}) \end{aligned}$$

式 *(3.3)*、式 *(3.6)* および定義 *3.11* の *Step 3* から、以下の式が成り立つ。

$$\nu(e_{i_{k-1} \to i'_k}) < \nu(e_{i_{k-1} \to i_k}) \tag{3.7}$$

P の部分パス P_1 を P_1' に置き換えたパスである P' は以下のように表される。

$$P' = \langle n_{i_1}, \ldots, n_{i_{k-1}}, n_{i'_k}, \ldots, n_{i_j}, \ldots, n_{i_t} \rangle$$

式 *(3.3)* と式 *(3.6)* から、以下の式が導かれる。

$$\begin{aligned} |P'| &= |P_1'| + |P_2| = |P_1| + |P_2| = |P| \\ \delta(P') &= \delta(P_1') + \delta(P_2) = \delta(P_1) + \delta(P_2) = \delta(P) \end{aligned}$$

上記および式 *(3.7)* から、P' は n_{i_1} から n_{i_t} へのベストパスである。この結果は、P がベストパスであるのと前提と矛盾する。

つまり、$P_1' \neq P_1$ かつ n_{i_1} から n_{i_j} へのベストパスとなるような P_1' は存在しない。つまり P_1 が n_{i_1} から n_{i_j} へのベストパスである。 ***Q.E.D.***

定義 3.12 $\mathcal{N}$ 中において、$P = \langle n_{i_1}, n_{i_2}, \ldots, n_{i_t} \rangle$ を n_{i_1} から n_{i_t} へのベストパスであるとする。このとき関数 f および f^j を以下のように定義する。

$$f(n_{i_k}, n_{i_t}, \mathcal{N}) = \begin{cases} n_{i_{k+1}} & (1 \leq k \leq t-1) \\ n_{i_t} & (k = t) \end{cases} \tag{3.8}$$

$$f^j(n_{i_k}, n_{i_t}, \mathcal{N}) = \begin{cases} f(n_{i_k}, n_{i_t}, \mathcal{N}) & (j = 1) \\ f(f^{j-1}(n_{i_k}, n_{i_t}, \mathcal{N}), n_{i_t}, \mathcal{N}) & (j > 1) \end{cases} \tag{3.9}$$

□

特に $f(n_s, n_d, \mathcal{N})$ は、$\mathcal{N}$ 中のノード n_s のフォワーディングテーブルにおける宛先ノード n_d への次転送先ノードを表している。

上記の $\mathcal{N}$ に関する定義から、次の定理が導かれる。

定理 3.2 $P = \langle n_{i_1}, n_{i_2}, \ldots, n_{i_t} \rangle$ が $\mathcal{N}$ 中 における n_{i_1} から n_{i_t} へのベストパスであるとしたとき、

$$f(n_{i_1}, n_{i_2}, \mathcal{N}) = n_{i_2} \tag{3.10}$$

が成り立つ。

証明 3.2 $P_1 = \langle n_{i_1}, n_{i_2} \rangle$ を P の部分パスであるとする。定理 *3.1* によれば、P_1 は n_{i_1} から n_{i_2} へのベストパスである。定義 *3.1* から二ノード間のリンクはたかだか一つであるため、P_1 はまた n_{i_1} から n_{i_2} へのベストパスでもある。$k = 1$ 及び $t = 1$ を、定義 *3.12* 中の式 *(3.9)* に適用することで、式 *(3.10)* を導くことができる。 ***Q.E.D.***

定理 3.3 $P = \langle n_{i_1}, n_{i_2}, \ldots, n_{i_t} \rangle$ が n_{i_1} から n_{i_t} へのベストパスであるとしたとき、

$$n_{i_{j+1}} = f^j(n_{i_1}, n_{i_t}, \mathcal{N}) \quad (1 \leq j \leq t-1) \tag{3.11}$$

が成り立つ。

証明 3.3 帰納法を用いて証明を行う。$j = 1$ のとき、定義 *3.12* から $n_{i_2} = f(n_{i_1}, n_{i_t}, \mathcal{N})$ を導くことができる。故に式 *(3.11)* は、$j = 1$ のときに成り立つ。

次に、式 *(3.11)* が $j = k-1$ のときに成り立つと仮定する。式 *(3.11)* に $j = k-1$ を代入すると、以下の式が導かれる。

$$n_{i_k} = f^{k-1}(n_{i_1}, n_{i_t}, \mathcal{N}). \tag{3.12}$$

定義 *3.12* により、n_{i_k+1} に関する次の式が成り立つ。

$$n_{i_{k+1}} = f(n_{i_k}, n_{i_t}, \mathcal{N}). \tag{3.13}$$

式 *(3.13)* 中の n_{i_k} に式 *(3.12)* を代入すると、次の式が得られる。

$$n_{i_{k+1}} = f(f^{k-1}(n_{i_1}, n_{i_t}, \mathcal{N}), n_{i_t}, \mathcal{N}). \tag{3.14}$$

定義 *3.12* より、式 *(3.14)* は次の式に変形可能である。

$$n_{i_{k+1}} = f^k(n_{i_1}, n_{i_t}, \mathcal{N}). \tag{3.15}$$

この式 *3.15* は、式 *3.11* において $j = k$ である場合に相当する。

最終的に式 *(3.11)* は、$j = 1$ であるときに成り立ち、かつ $j = k - 1$ のときに成り立てば $j = k$ のときも成り立つ。それゆえ任意の j において式 *(3.11)* は成り立つ。 ***Q.E.D.***

定理 3.3 は、$f^j(n_s, n_d, \mathcal{N})$ が n_s から n_d へのベストパス中のノードのうち、j 番目に到達するノードであることを示している。

3.3 到達不能条件の導出

本節では、単一ノード故障発生時に、ネットワークシステム中のある一つのノードを除いてすべてのノードが経路更新を行った場合、経路更新を行わなかったノードが保有する経路が到達不能経路となる条件を、前節で形式化を行ったネットワークシステムを用いて導く。ここでは、議論を単純化するために、経路更新を行わないノードはネットワーク中に一つのみとしている[1]。さらにこれらの条件は、到達不能ノードが生じる必要条件となっていることを証明する。

ネットワークシステム $\mathcal{N}$ 中のノード n_s を、故障発生時における経路更新を行わないノードであるとする。また $\mathcal{N}$ 中のノード n_f に故障が発生した後のネットワークシステムを $\mathcal{N}_f = \mathcal{N} - \{n_f\}$ とする。ここで、パケットが始点ノード n_s から宛先ノード n_d へ到達するための条件について考える。宛先ノード n_d 自体に故障が発生した場合、n_d 宛てのパケットは宛先まで到達できない。つまり、以下の条件が、パケットが宛先に到達するための必要条件の一つとなる。

- n_d が故障ノードでない。

また、故障前のネットワークシステム $\mathcal{N}$ 中において n_s から n_d への唯一のパスが故障により失われた場合、n_s から n_d へのパケットが宛先に到達できないのは自明である。つまり、以下の条件も必要条件の一つとなる。

[1] 4 章で議論する局所化アルゴリズムでは、単一故障に対して複数のノードで経路更新が必要になる場合も扱うが、本章では経路更新を行わないノードを一つに限定し、その影響を分析する。

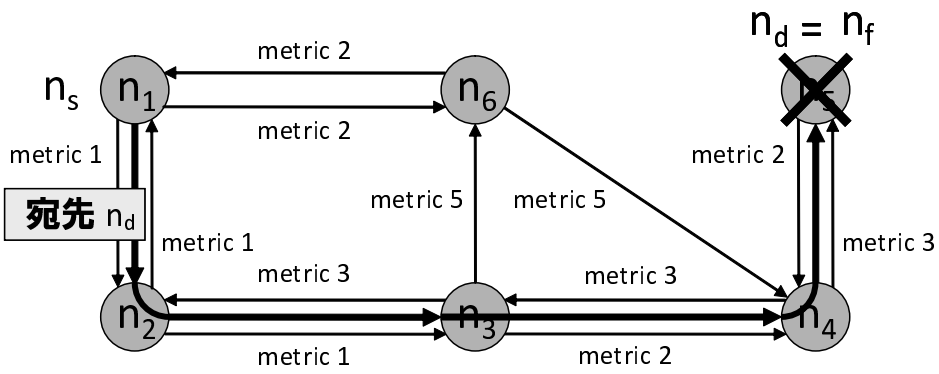

図 3.2: 宛先ノード自体の故障

- 故障後のネットワークシステム $\mathcal{N}_f$ 中において、n_s から n_d へのパスが少なくとも一つ以上存在している。

また、上記のいずれの条件を満たしていても、各ノードがパケット転送を正しく行なわなければ、パケットは宛先に到達しない。そのため、以下が必要条件の一つとなる。

- 各ノードが正しい次転送先ノードを選択する。

これらの条件を、以下にまとめる。

(a). 宛先ノードは、故障ノードではない。

(b). $\mathcal{N}_f$ 中に、宛先までのパスが少なくとも一つ存在する。

(c). 各ノードは正しい次転送先を選択する。

上記の条件をいずれかひとつでも満たさないとき、n_s から n_d へのパケットは宛先まで到達できない。これらの条件 (a)–(c) を用いて、ネットワークシステム中に故障が発生したときにあるノードが経路更新を行わなかった場合における到達不能ノードが生じる条件について形式化を行う。

まず、条件 (a) および条件 (b) が成り立たない場合について、それぞれ形式化を行う。

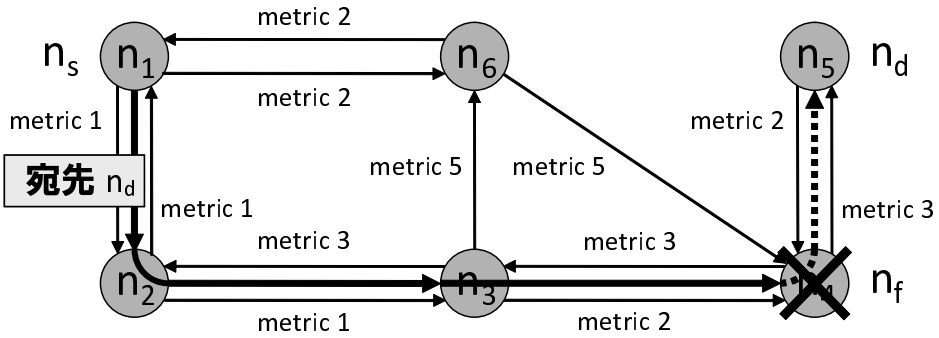

図 3.3: 宛先ノードまでのパスの喪失

条件 3.1 $n_d = n_f$. □

条件 3.2 $\mathcal{N}_f$ 中に n_s から n_d へのパスが存在しない。 □

これらの条件をそれぞれ、図 3.2 および図 3.3 に示す。次に、条件 (c) について考える。条件 (c) を満たさないということは、つまり $\mathcal{N}_f$ 中のいずれかのノードが誤った次転送先を選択するということである。n_s 以外のノードは、n_f の故障時に経路更新を行っている。それゆえ、誤った次転送先を選択する可能性があるのは、経路更新を行わない n_s だけである。

この条件は、さらに以下の二つのケースに分けることができる。n_s の次転送先ノードが故障ノードである場合とそうでない場合である。n_s の次転送先ノードが故障ノード n_f である場合 (図 3.4 参照)、n_s から n_d へのパスは、$P = \langle n_s, n_f, \cdots, n_d \rangle$ と表せる。しかし、故障後のネットワークシステムである $\mathcal{N}_f$ 中に、n_f は存在しないため、P もまた存在しない。それゆえこのケースは以下のように記述できる。

条件 3.3 $f(n_s, n_d, \mathcal{N}) = n_f$. □

次に、n_f が n_s の次転送先ノードでない場合について考える。n_s は、故障発生に伴う経路更新を行っていないので、$\mathcal{N}$ に基づいた経路表を用いている。一方 n_s 以外のノードは、故障発生に伴い自身の経路表を更新している。このとき n_s における n_d 宛てのパケットの次転送先ノードは n_h であるとする。$\mathcal{N}_f$ 中において n_h から n_d へのパス上に n_s が存在しなかった場合、パケットは n_s から n_d に到達す

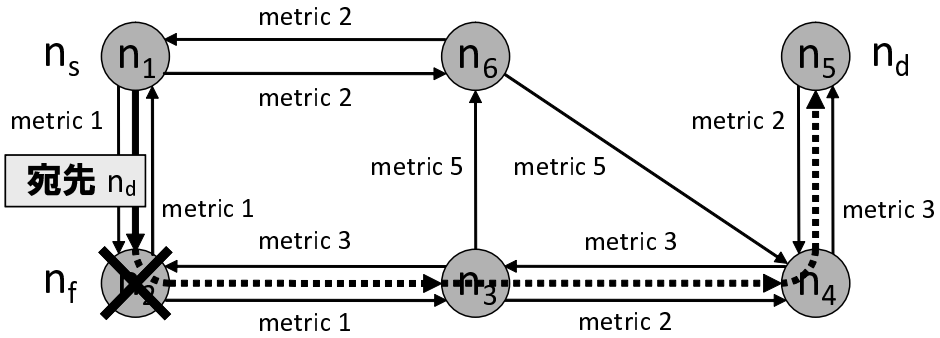

図 3.4: 次転送先ノードの故障

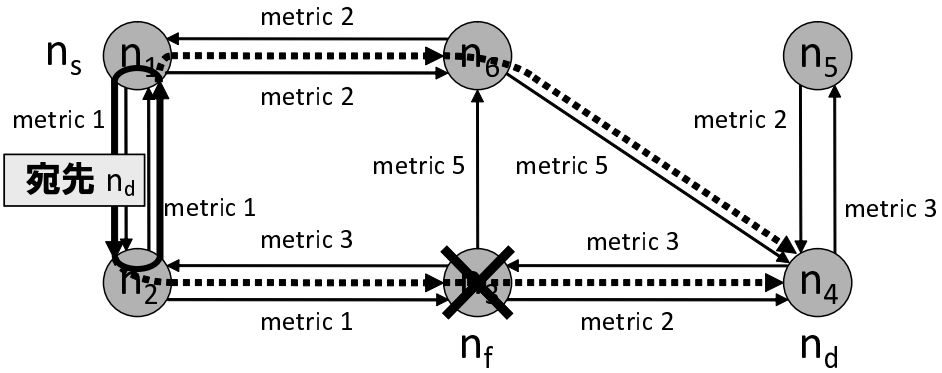

図 3.5: ルーティングループ

ることができる。なぜなら n_h から n_d 中の各ノードは、故障後のトポロジーに基づいた経路表を持っており、宛先 n_d に向けた正しい次転送先にパケットを転送するからである。一方で、n_h から n_d へのパス上に n_s が存在する場合、n_s から送られたパケットが再び n_s に到達することとなる。つまり、ルーティングループが発生する (図 3.5 参照)。この条件は、以下のように形式化される。

条件 3.4 $n_h = f(n_s, n_d, \mathcal{N})$ である時、$f^i(n_h, n_d, \mathcal{N}_f) = n_s$ を満たす i が存在する。□

ここで挙げた条件は、一般的なネットワークシステムにおけるルーティングループが生じる条件である。ネットワークシステム $\mathcal{N}$ は双方向であるとの仮定によ

り、条件 3.4 は以下の定理を用いることで簡素化できる。

定理 3.4 n_s, n_h, n_f および n_d はそれぞれ、双方向ネットワークシステム $\widetilde{\mathcal{N}}$ 中の異なるノードであるとする。ただし n_s は n_f に隣接していないものとする。また $\widetilde{\mathcal{N}}_f = \widetilde{\mathcal{N}} - \{n_f\}$ であるとする。また $n_h = f(n_s, n_d, \widetilde{\mathcal{N}})$ であるとしたとき、以下の式が成り立つのはたかだか $i = 1$ 時のみである。

$$f^i(n_h, n_d, \widetilde{\mathcal{N}}_f) = n_s \tag{3.16}$$

証明 3.4 式 *(3.16)* が $i > 1$ のときに成り立つと仮定し、矛盾を導く。

定理 *3.3* によれば、n_s は $\widetilde{\mathcal{N}}_f$ 中において n_h から n_d へのベストパス P 上における i 番目のノードである。ベストパス P は次のように表記できる。

$$P = \langle n_h, \ldots, n_s, \ldots, n_d \rangle$$

パス P の部分パスであり、n_h から n_s へのパスを $P_1 = \langle n_h, \ldots, n_s \rangle$ とする。また互いに隣接する n_h から n_s へのパスを $P_2 = \langle n_h, n_s \rangle$ とする。定理 *3.1* により、P_1 は n_h から n_s へのベストパスである。このとき、以下の式が成り立つ。

$$|P_1| \leq |P_2| \tag{3.17}$$

定義 *3.6* によれば、$i > 1$ なので P_1 の長さは *1* 以上であり、P_2 の長さは *1* である。$|P_1| = |P_2|$ であれば、定義 *3.11* によりベストパスは P_2 であり、P_1 ではない。このことは P_1 がベストパスであるとの仮定と矛盾する。

それゆえ、式 *(3.17)* から以下の式が得られる。

$$|P_1| < |P_2| \tag{3.18}$$

定理 *3.2* と $n_h = f(n_s, n_d, \widetilde{\mathcal{N}})$ から、次の式が導かれる。

$$n_h = f(n_s, n_h, \widetilde{\mathcal{N}}) \tag{3.19}$$

式 *(3.19)* は、$\widetilde{\mathcal{N}}$ 中における n_s から n_h へのベストパスが P_2 の逆方向パスである $\overline{P_2} = \langle n_s, n_h \rangle$ であることを示している。n_f は $\overline{P_2}$ 上には存在しないため、

$\widetilde{\mathcal{N}_f}$ 中において $\overline{P_2}$ よりも小さなメトリックを持つパスは存在しない。式 *(3.19)* と $n_h \neq n_f$ から $\widetilde{\mathcal{N}_f}$ 中における以下の式が導かれる。

$$n_h = f(n_s, n_h, \widetilde{\mathcal{N}_f}) \tag{3.20}$$

ここで P_1 の逆パスを $\overline{P_1} = \langle n_s, \cdots, n_h \rangle$ とする。式 *(3.20)* から $\overline{P_2}$ は n_s から n_h へのベストパスであり、$\overline{P_1}$ の長さは *1* 以上、$\overline{P_2}$ の長さは *1* であることから、次の不等式が成り立つ。

$$|\overline{P_1}| \geq |\overline{P_2}| \tag{3.21}$$

双方向ネットワークシステムでは、あるパスとその逆パスのメトリックは同一であるため、式 *(3.18)* と式 *(3.21)* はお互いに矛盾する。それゆえ式 *(3.16)* が成り立つのはたかだか $i = 1$ であるときのみである。 ***Q.E.D.***

定理 3.4 から、双方向ネットワークシステムにおいては、ルーティングループは始点ノード n_s とその隣接ノード n_h との間でのみ発生する。それゆえ、条件 3.4 は、定理 3.4 により以下の条件に置き換えることができる。

条件 3.4' $\widetilde{\mathcal{N}}$ を双方向ネットワークシステムとしたとき、$n_h = f(n_s, n_d, \widetilde{\mathcal{N}})$ とする。このとき、以下の式が成り立つ。

$$f(n_h, n_d, \widetilde{\mathcal{N}_f}) = n_s \tag{3.22}$$

□

最後に、条件 3.1 から条件 3.4 のいずれかを満たすことが、到達不能経路が生じるための必要条件であることを示す。

定理 3.5 ネットワークシステム $\mathcal{N}$ 中において、n_s を経路更新を行わないノードであるとする。また n_f は故障ノードであるとする。$\mathcal{N}_f$ は $\mathcal{N}$ の部分ネットワークシステムであり、$\mathcal{N}_f = \mathcal{N} - \{n_f\}$ であるとする。

n_d が n_s から到達可能であるならば、条件 *3.1* から条件 *3.4* のうち、いずれかの条件が成り立つ。

証明 3.5 n_d が n_s に到達できなくなる条件が、条件 *3.1* から条件 *3.4* を除いて他にないことを示す。n_d が故障ノードではなく、かつ n_d までのパスが存在している、つまり条件 *3.1* および条件 *3.2* を満たしていないケースについて考える。

このケースは、n_s の経路表中の次転送先ノード n_h としたとき、以下の二つのケースに分けて考えることができる。

(i). n_s から n_h へ到達不能である。

(ii). n_h から n_d へ到達不能である。

上記の条件がそれぞれ成り立つ時、条件 *3.3* および条件 *3.4* もそれぞれ成り立つことを示す。

始めに前者のケースについては、対偶の証明を行う。$\mathcal{N}_f$ 中において n_h が故障ノードでなければ、n_h は n_s からのパケットを受け取ることができる。その理由は n_h は n_s の隣接ノードであり、n_f の存在は n_s から n_h へのパケットの到達性に影響を与えないからである。それゆえ、条件 *3.3* が成り立たなければ、ケース *(i)* も成り立たない。

次に、条件 *3.4* が成り立たない時には、ケース *(ii)* もまた成り立たないことを示す。$\mathcal{N}_f$ 中において、P は n_h から n_d へのベストパスであるとする。P 上のノードのうち誤った次転送先ノードを選択する可能性があるのは n_s のみである。しかし、条件 *3.4* が成り立っていなければ、n_s は P 上には存在しない。それゆえ P 上のすべてのノードは、正しい次転送先ノードを選択可能である。すなわち n_h から n_d は到達可能である。従って、条件 *3.4* が成り立たない時には、ケース *(ii)* も成り立たない。

以上より、n_d が n_s から到達不能であれば、条件 *3.1* から条件 *3.4* のいずれかが成り立つ。 ***Q.E.D.***

3.4 到達不能ノード数の実験的評価

双方向ネットワークシステムにおいて、前節で述べた条件を満たす到達不能経路がどの程度生じるかを調べるために、シミュレーションを行った。シミュレー

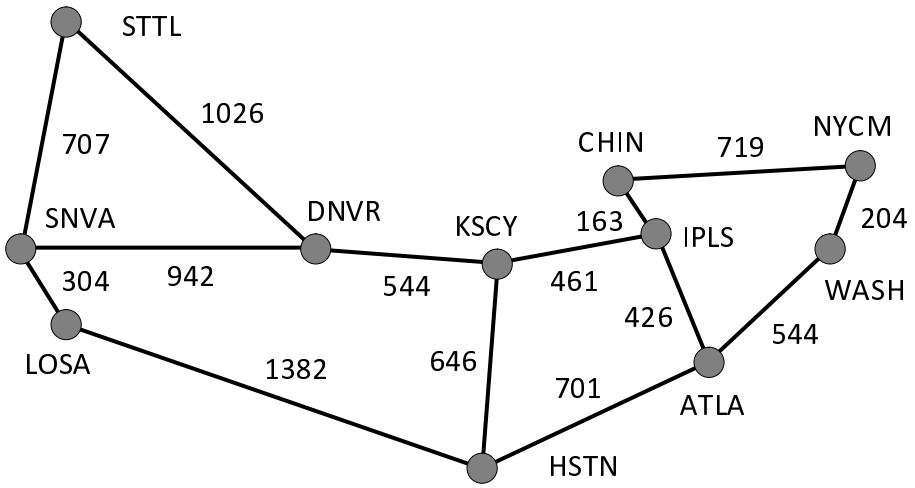

図 3.6: Internet2 バックボーン

ションでは、二つのトポロジーモデルを使用した。一つは、図 3.6 に示した Internet2 [Internet2] のバックボーンネットワークを用いた。この図中の各ノード毎に記載されているラベルはノードが配置されている拠点名を、また各リンク毎に記載されている数値はリンクのメトリックを、それぞれ表している。またもう一方は、ネットワークのモデル化によく用いられているランダムネットワーク [Wax88, Doa93] を用いた。ランダムネットワークでは、N 個のノードがユークリッド空間上に配置され、ノードペア (u, v) 間のリンクは以下の確率で生成される。

$$P(u, v) = \beta \exp \frac{-d(u, v)}{\alpha L},$$

ここで $d(u, v)$ はノード u, v 間のユークリッド距離を、L は二ノード間の距離の最大値をそれぞれ表している。α は u, v 間の距離とリンク生成確率の関係を制御するパラメータであり、また β は生成されるリンク数に影響を与えるパラメータである。また $d(u, v)$ を、二ノード u, v 間のリンクのメトリックとして用いる。

3.4.1 各条件毎の到達不能ノード数

図 3.6 に示すトポロジーにおいて、単一ノード故障が発生した場合に条件 3.1 ～ 3.3 および 3.4’ によりどの程度のノードへ到達不能となるかを調べた。双方向ネッ

表 3.1: 条件別到達不能ノード数 (Internet2 バックボーン)

距離	条件			
$(n_s \rightarrow n_f)$	3.1	3.2	3.3	3.4'
1	1.00	-	2.92	-
2	1.00	0.00	-	0.59
3	1.00	0.00	-	0.09
4	1.00	0.00	-	0.06
5	1.00	0.00	-	0.00

トワークシステムを対象としているため、条件 3.4 の代わりに条件 3.4' を用いている。始点ノードと故障ノードとの距離毎に、到達不能となる経路数の平均を求めた結果を、表 3.1 に示す。

表 3.1 の結果から以下のことがわかる。

- 条件 3.1 は宛先ノード自体が故障する場合であるため、その故障ノードのみが到達不能となる。
- 条件 3.2 は宛先ノードに至るまでの唯一のパスが故障により失われる場合である。図 3.6 のトポロジーでは任意の二ノード間には少なくとも二つ以上のパスが存在すため、条件 3.2 の元に到達不能となる経路は存在しない。
- 条件 3.3 は始点ノードに隣接するノードに故障が生じた場合を意味している。始点ノードから送られるすべてのパケットは隣接ノードのいずれかを通過するので、隣接ノードの故障が多数の経路に影響を与えるのは明らかである。
- 条件 3.4' は、ルーティングループにより、到達不能なノードが生じるケースである。表 3.1 の結果は、始点ノードから故障ノードまでの距離が増加するにつれて、到達不能経路数が減ることを示している。

次に、さらに大規模なネットワークシステムの場合について検証するために、1000 ノード、2000 リンクのランダムネットワークを用いたシミュレーションを行った。その結果を表 3.2 に示す。

表 3.2: 条件別到達不能ノード数 (1000 ノード、2000 リンクのランダムネットワーク)

距離	条件			
$(n_s \to n_f)$	3.1	3.2	3.3	3.4'
1	1.00	-	244	-
2	1.00	0.09	-	17.1
3	1.00	0.11	-	1.93
4	1.00	0.05	-	0.60
5	1.00	0.07	-	0.26
6	1.00	0.10	-	0.12
7	1.00	0.09	-	0.04
8	1.00	0.09	-	0.00
9	1.00	0.05	-	0.00
10	1.00	0.00	-	0.00

- 条件 3.2 に該当する到達不能経路数は、Internet2 バックボーンの場合と結果が異なっている。その理由は、あるノード故障により二ノード間のパスが失われた場合、ランダムネットワークでは代替となるパスが存在しないケースがあるためである。

- 条件 3.3 及び条件 3.4' による結果は、ネットワークシステム中のノード数の増加に伴い、それぞれの条件による到達不能経路の数が増えることを示している。

条件 3.4' に関する二つのシミュレーションの結果はそれぞれ、到達不能経路数と始点から故障ノードまでの距離との間には、逆相関の関係があることをしめしている。つまり、始点ノードが故障ノードから遠ければ遠いほど、ルーティングループが発生する可能性は低くなる。この結果の理由は、以下のとおりであると考えられる。

- 始点ノードから故障ノードへの距離が遠い場合、故障ノードが始点ノードから宛先ノードまでの最短パス上のノードである可能性が低くなる (図 3.7 参照)。これは、始点から近いノードは、遠くのノードより、多くパス上に含まれるためである。

- 一般に代替パスは、故障ノードの周辺に構成される (図 3.8 参照)。そのため、故障ノードが始点ノードから遠くなるにつれて、始点ノードの次転送先ノードが変わる可能性が低くなる。始点ノードからの次転送先ノードが変わらなければ、ルーティングループは発生しない。つまり、故障ノードと始点ノードの距離が大きくなるにつれて、ルーティングループが発生する可能性は低くなる。

3.4.2 ルーティングループの詳細評価

3.4.1 節で、条件 3.4' による二つの結果は、到達不能経路数と始点ノードから故障ノードまでの距離との関係に関して同じ傾向を示していたが、その値には違いが見られた。そこでこの違いが、ネットワークシステム中のリンク数の違いに起因

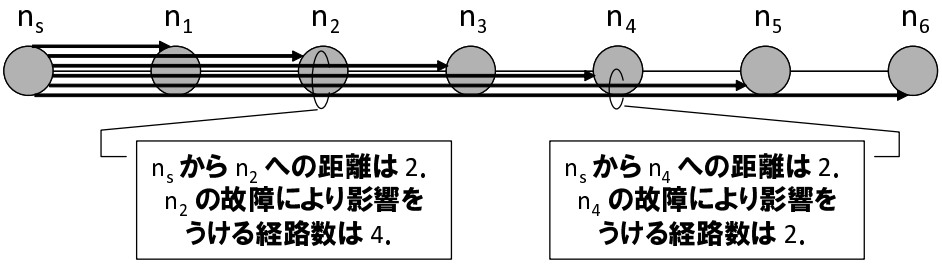

図 3.7: 故障ノードまでの距離と影響を受けるノード数の関係

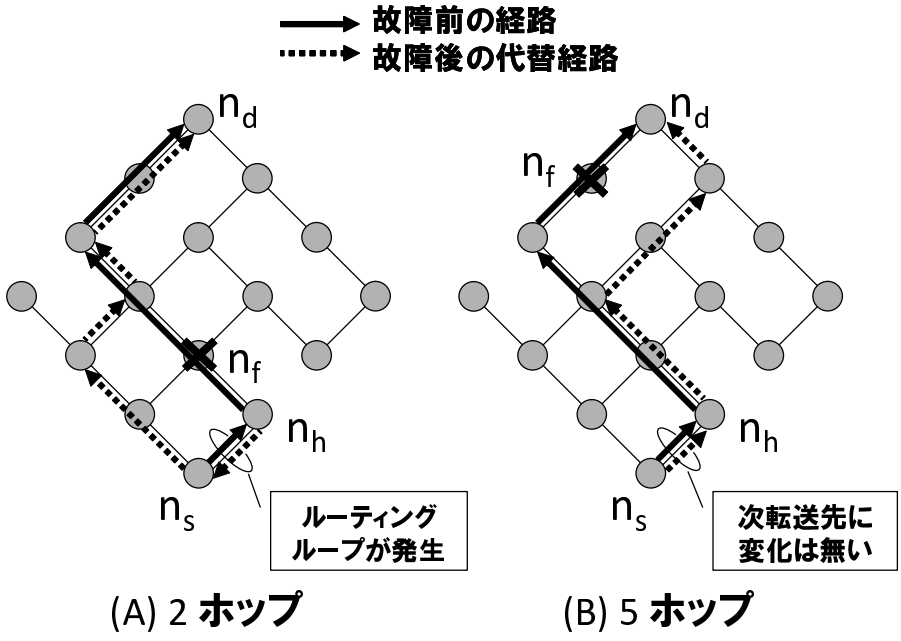

図 3.8: 故障ノード周辺における代替パスの形成

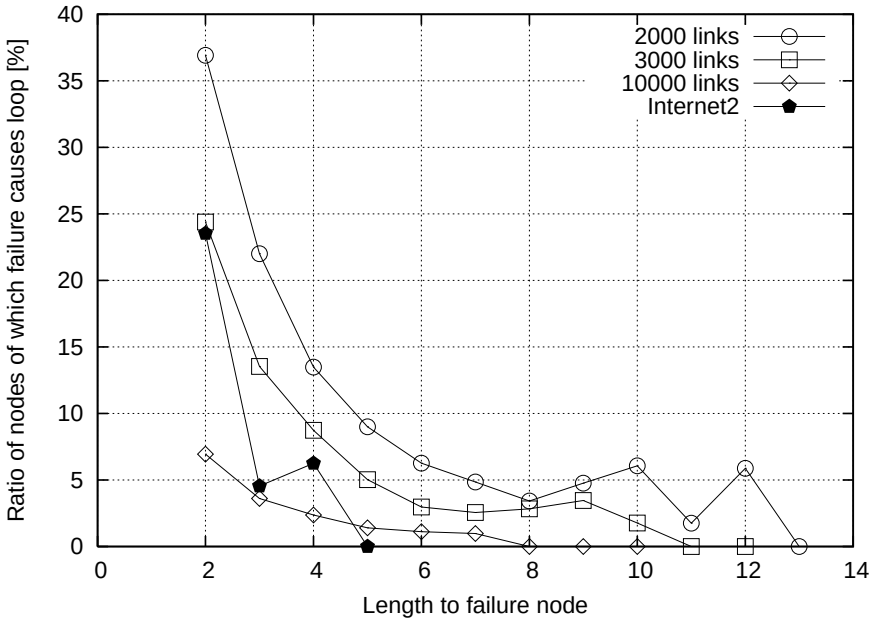

図 3.9: リンク数とループの関係

しているのか否かを確認する。リンク数がそれぞれ 2000, 3000 および 10000 と異なる 1000 ノードのランダムネットワークを複数用意し、それぞれに対してシミュレーションを実施した。それぞれのリンク数のランダムネットワークを 10 回ずつ生成し、条件 3.4' に相当する経路数の平均を求めた。図 3.9 にその結果を示す。

この結果から、ルーティングループが発生する可能性は、ネットワークシステム中のリンク数の増加と共に減少していることが分かる。条件 3.4' で示したように、ルーティングループは経路更新を行わないノードとその隣接ノードとの間で発生する。ネットワークシステム中のリンク数が多くなると、隣接ノードから宛先までの代替パスの候補の数が多くなるため、隣接ノードがある宛先への経路として経路更新を行わないノードを選ぶ可能性が、相対的に低くなる。そのため、ルーティングループの発生する可能性が減少すると考えられる。

3.5 考察

3.3 で、故障発生時における到達不能ノードが生じる四つの条件を挙げた。経路

制御による高速な復旧を実現するにあたり、それぞれのケースに対して、どのように対処すべきかの考察を行う。

条件 3.1 はパケット自体の宛先ノード自体が故障するケースであり、条件 3.2 は宛先までの代替パスが存在しないケースである。これらの故障に伴い、各ノードが経路更新を行った場合、経路表からこれらの条件に該当する宛先エントリが存在しなくなるため、この宛先のパケットは廃棄される。経路更新を行わなかった場合、故障箇所の直前のノードまでは配送されたパケットは、それ以上の転送ができないため廃棄される。つまり、これらのケースは、いずれも経路制御により解決可能な問題ではない。条件 3.1 への対処法としては、IP ネットワークを利用する上位のアプリケーションが通信を行う宛先ノードを変更するという方法が一般的である。例えば、同等のサービスを提供するサーバを二台用意しておき、アプリケーションは、一方からの応答がない場合には、もう一方へ通信を切り替える。IP ネットワークは、それぞれのノードへの到達性を独立に確保しておけばよい。また条件 3.2 に関しては、ネットワーク中へのリンク追加など、ネットワークトポロジーの設計により解決すべき問題である。

一方で条件 3.3 と条件 3.4 は、適切な経路更新により到達不能状態を解消行うべきケースである。条件 3.3 は、隣接ノードの故障であるため、2.4.1 で説明を行った技術を用いて故障検知を行い、検知した故障に応じて経路表更新等の対処を行えばよい。一方、条件 3.4 は、故障箇所に隣接していない場合に到達不能ノードが生じるケースである。この条件により到達不能ノードが生じる否かを判定することは、高速な復旧を実現するためには重要である。3.4.1 では、ルーティングループ発生の有無は、故障箇所からの距離に関係することを示したが、その具体的な判定手段については示していない。この判定手段については、4 章において議論を行う。

3.6 結言

本研究では、故障発生時に一つのノードが経路更新を行わなかった場合、パケット到達性にどのような影響が生じるかについて調査を行い、次の事を明らかにした。

まず故障によるトポロジー変化に起因して到達不能ノードが生じる条件として

以下の四つを示した。

- 宛先ノード自体の故障
- 宛先ノードまでのパスの喪失
- 次転送先ノードの故障
- ルーティングループの発生

これらの条件のいずれかに当てはまることが、到達不能ノードの発生の必要十分条件である。

上記条件のうち、次転送先ノードの故障とルーティングループ発生は、経路制御において対処を行うべき問題ある。特に次転送先の故障は、2.4.1 で説明を行った技術を用いて故障検知を行い、経路表の更新等の対処を行えばよい。一方、ルーティングループ発生は、故障箇所に隣接していない場合に到達不能ノードが生じるケースである。このケースは、故障箇所からの距離が遠いほど発生する可能性が低いことをシミュレーションを用いて示した。

故障発生時におけるパケット到達性を高速に復旧させるためには、経路更新を必要とするノードを特定することが重要である。経路更新を必要とするノードのみが特定出来れば、それらのノードだけ他のノードよりも先に、なんらかの故障に対する対処を実施させることが可能となる。このことにより、従来と比べ、より高速な復旧が見込める。4 章において、これらのノードを特定する手法について議論を行う。

第4章　ループ回避条件を用いた経路変更箇所の局所化アルゴリズム

4.1　緒言

IP ネットワークの可用性向上のために、故障からの高速復旧を実現する手法の実現が望まれている。しかし、従来提案されている高速復旧手法の多くは、転送方式の変更や代替経路の集中的な計算が必要であり、このことが既存の IP ネットワークに対する適用への妨げとなっていた。このため、転送方式の変更を必要とせず、分散処理が可能である高速復旧手法が求められている。

前章では、既存の IP 転送方式においては、故障発生時におけるパケット到達性復旧のために、ネットワーク中のノードが経路更新を行う必要があることを明らかにした。しかし、故障箇所から遠いノードが経路更新を行わなくても、パケット到達性に影響はないことから、経路更新を行うべきノードはネットワーク中の全ノードである必要はない。また、経路更新を行うべきノードを限定することは、故障発生におけるパケット到達性を復旧を高速に行うために、有用であると考えられる。

そこで本章では、前章における結果に基づいて、経路更新を必要とするノードを特定する局所化アルゴリズムを提案する。ネットワーク中のあるリンクに対し、本アルゴリズムを適用すると、そのリンクの故障発生時、経路更新を行わないとループを引き起こすノード (影響ノード) を特定することができる。また、本アルゴリズムはネットワーク中の各ノードにより分散して実行することが可能であるため、既存の IP ネットワークへ適用可能な高速復旧手法の実現に役立つと考えられる。

本章の構成は以下の通りである。4.2 で、本章で提案するアルゴリズムにおける局所化の実現について、基本アイデアの説明を行う。4.3 では、あるリンク故障に対してノードが影響ノードか否かを判定するための条件であるループ回避条件を示し、パケット到達性を復旧させるためには、これらの条件を共に満たすノードのみが経路更新を行えばよいことを証明する。また 4.4 では、4.3 で示したループ回避条件の判定を、ネットワーク中の各ノードにより分散して行うためのアルゴリズムを提示し、最後に 4.5 において本章のまとめを行う。

4.2 局所化の基本アイデア

図 4.1 (a) 中のネットワークにおいて、ノード D, F 間のリンクが故障した場合について考える。ドメイン内経路制御プロトコルの一つである OSPF [RFC2338] では、故障発生時には、ネットワーク中の全ノードが経路表の再計算を行う。しかし Incremental SPF アルゴリズム [McQ80] における差分計算の考え方を流用することで、経路表を計算するノードの数を最小限に抑えることが可能である。図 4.1 (b) において、ノード A からノード F, G へ送られるパケットは、ノード D, F 間のリンクを通過する。Incremental SPF を用いた場合、ノード D, F 間のリンク故障時には、ノード A はノード F, G への経路を再計算する。一方で、図 4.1 (c) においては、ノード E から送られるパケットは、ノード D, F 間のリンクを使用しない。Incremental SPF を用いた場合、ノード E は、このリンクの故障時に自身の経路を計算する必要がない。

Incremental SPF アルゴリズムにおける計算量の削減は、故障リンクが最短パスツリー上にあるか否かというアイデアに基づいている。例えば図 4.1 中のネットワークは 8 本のリンクを持っているため、各ノードは故障後に使用すべき代替経路表がそれぞれのリンクに対して必要であった。しかし、最短パスツリー上にあるか否かという判断基準を用いることで、ノード A が必要とする代替経路表は 8 から 6 に削減することができる (図 4.1 (b) 参照)。これは、ノード C–E 間のリンクおよびノード E–G 間のリンクに関しては、ノード A を始点とする最短パスツリー上になく、その故障時にノード A の経路に影響を与えないためである。以降、この手法を最短パスツリーアプローチと呼ぶこととする。

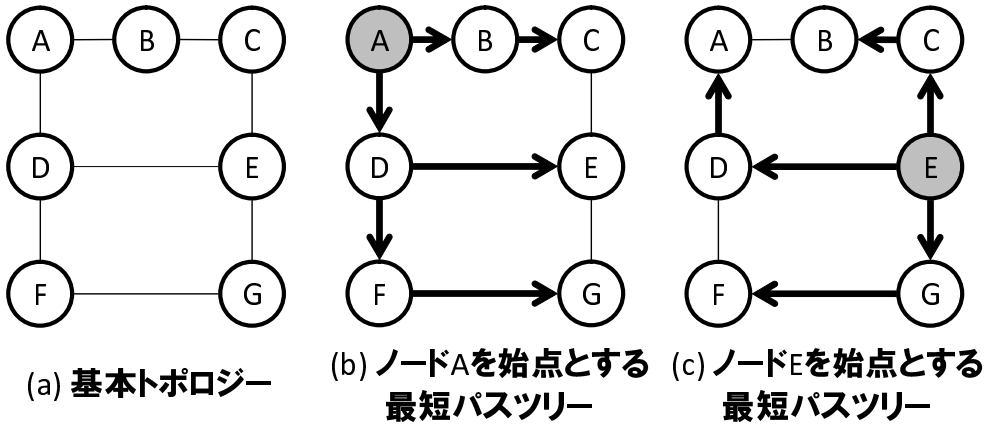

図 4.1: ネットワークトポロジーと最短パスツリー

ネットワーク中のすべてのリンクに対して故障時に使用する代替経路表を用意するアプローチ (以降、網羅的アプローチと呼ぶ) に対して、最短パスツリーアプローチは代替経路表数を削減することができるが、それでもまだ多くの代替経路表を用意する必要がある。ネットワーク中のノード数を N とした時、このネットワーク上におけるあるノードを始点とする最短パスツリーに含まれるリンク数は $N-1$ である。それゆえ、必要となる代替経路表数は、ネットワークの規模に応じて大きくなる。そこで必要とする代替経路表数を大幅に削減するための手法を提案する。本手法では、代替パスが最短パスとなることを諦める代わりに、パケット到達性を保証するのに必要最小限となるノードのみが代替経路表を用意することとする。

例えば、ノード D–F 間のリンク故障発生時におけるノード A を始点とする最短パスツリーを作るためには、ノード A は代替経路表を用意する必要がある。しかし、図 4.2 (a) におけるノード A からノード G へのパケット到達性のみを考えた場合、ノード D がノード D–F 間のリンク故障に使用する代替経路表を持っているので、ノード A は故障前の経路表をそのまま用いても問題ない。同じ理由で、ノード A はノード D–E 間、ノード E–G 間およびノード F–G 間の各リンクの故障時にも、故障前に使用していた経路表をそのまま使用することが可能である。一方で、ノード B–C 間のリンク故障時、ノード A が故障前に使用していた経路表をそのまま利用した場合、ノード A からノード C 宛のパケットの到達性に

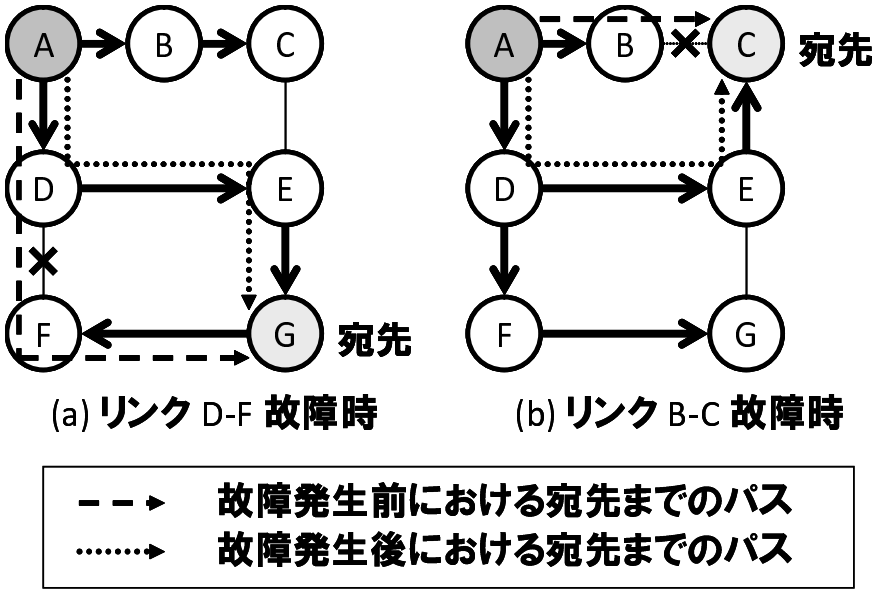

図 4.2: リンク故障と経路表の更新

問題が生じる。この場合、このパケットはノード A からノード B に転送される。ここで、ノード B の経路更新が行われなかった場合、ノード B–C 間のリンク故障のため、このパケットはノード C に到達できない。一方、ノード B が経路更新を行った場合、ノード A–B 間においてルーティングループが発生する (図 4.2 (b) 参照)。このため、ノード B–C 間のリンク故障時に、ノード A からノード C へパケットを到達させるためには、ノード A の経路更新が必要となる。ノード A–B 間およびノード A–D 間のリンクについても同様に、その故障時にノード A は経路更新を行う必要がある。つまり、パケットの到達性を保証するには、ノード A はノード A–B 間、ノード A–D 間およびノード B–C 間の三つのリンクに対してのみ、故障時に使用する代替経路表を用意すればよい。一方で、網羅的アプローチでは、図 4.2 中のネットワークにおける八つのリンクすべてに対して、代替経路表を用意する必要がある。このため、本手法を用いることで五つの代替経路表を削減することができる。

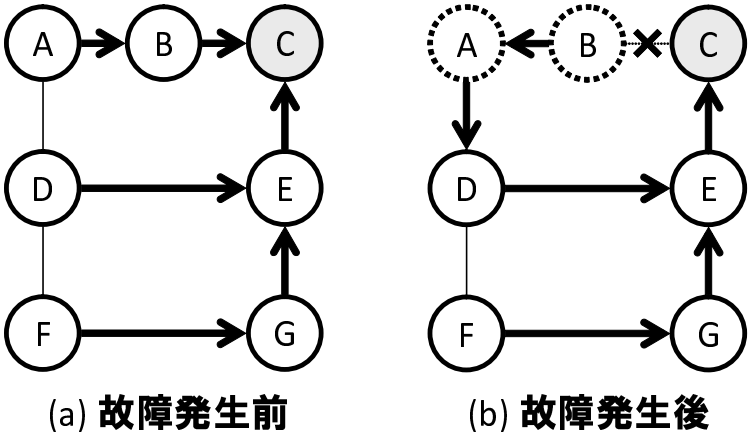

図 4.3: 逆方向最短パスツリー

4.3 ループ回避条件

4.3.1 ループ発生回避条件の導出

本節では、あるリンクの故障に対し、経路更新を行うべきノードを特定するため条件を示す。

IP パケットは、宛先までの最短パスに沿って転送される。そのため、経路更新を行うべきノードを特定するためには、故障前後における宛先へ至る最短パスの変化を捉えることが重要である。この変化は、図 4.3 に示すように、逆方向最短パスツリーを用いることで捉えることが出来る。OSPF の経路計算などで用いられる順方向の最短パスツリーは特定の始点ノードから他の全ノードへの最短パスを表すのに対し、逆方向最短パスツリーはネットワーク中の全ノードから特定の宛先ノードへ至る最短パスを表す。

図 4.3 に示すように、ノード B, C 間のリンクが故障したとき、ノード A および B 間はパスの向きが変わっている。したがって、ノード A, B いずれかのノードが経路表を更新しなかった場合、これらのノード間でルーティングループが発生する。それゆえ、これらのノードにおける経路更新は必須となる。これらのノードは、以下の二つの特徴をともに持つ。まず一つ目の特徴は以下のとおりである。

- 故障前のトポロジーにおいて、これらのノードから宛先ノード C への最短

パスは故障リンクを含んでいる (図 4.3 (a))。

故障前のトポロジーにおいて宛先までの最短パス中に故障リンクを含まない場合、故障後においてもその最短パスに変化はないため、経路更新を行う必要はない。図 4.3 (a) において、ノード A, B がこの特徴を持つ。次に、二つ目の特徴を示す。

- 故障後のトポロジーにおいて、これらのノードは、故障リンクに接続しているノード B から宛先ノード C への最短パス上に存在する (図 4.3 (b))。

この条件を満たすノードが経路更新を行うことにより、故障後のトポロジーにおいて、故障リンクに接続しているノード B からノード C に至るパスが構成され、ネットワーク中の各ノードからノード C へのパケット到達性が復旧する。図 4.3 (b) においてノード B, A, D, E がこの特徴を持つ。このうち一つ目の特徴を持たないノード D, E は、図 4.3 (a), (b) を比較すると、故障前後でノード C 宛の経路に変化がない。このため、故障時にこれらのノードの経路更新は不要である。つまり、二つの特徴を共に持つノード A, B のみが経路更新を行えば、ネットワーク中の各ノードからノード C へのパケット到達性を復旧することができる。以上から、リンク故障発生時に経路更新が必要となるノードを特定するためには、これらに二つの特徴を満たすノードを発見すればよい。

これらの二つの特徴をループ回避条件として、一般化を行う。e は故障リンク、n_r は e に接続するリンク、n_d は宛先ノードであるとした時に、もし、n_i が以下の条件をともに満たせば、n_i は e の故障時に n_d 宛への経路を更新する必要がある。

条件 4.1 e の故障前のトポロジーにおいて、n_i から n_d への最短パス上に e が存在する。

条件 4.2 e の故障後のトポロジーにおいて、n_r から n_d への最短パス上に n_i が存在する。

条件 4.1 は、経路更新が必要なノードの候補を選定するために用いる一般的な基準である。条件 4.2 は、ルーティングループを避けつつ、影響ノードを最小限に抑えるために用いる。最短パスツリーアプローチは、本質的に条件 4.1 のみを用いている。つまり、条件 4.2 を用いていない分、余分な経路表が必要となる。

しかし我々の手法では、宛先ノード n_d に対して条件 4.1 と条件 4.2 の両方を満たす場合のみ、e の故障時に n_d への経路を更新する。

4.3.2 証明

ネットワーク中のあるリンク e に故障が発生したと仮定する。ただし、リンク e の故障後においても、ネットワーク中の任意の二点間には必ずパスが存在すると仮定する。このときネットワーク中のすべてのノードからノード n_d へのパケット到達性を確保するためには、条件 4.1 および条件 4.2 の両方を満たしたノードのみが宛先ノード n_d への経路の更新を行うことで、十分であることを証明する。

$\mathcal{V}$ はネットワーク中のすべてのノードの集合であるとする。$\mathcal{V}$ を、互いに排他となる以下の三つの部分集合に分割する。

- $\mathcal{V}_u^{(d)}$: n_d に対し、条件 4.1 および条件 4.2 を共に満たすノードの集合
- $\mathcal{V}_a^{(d)}$: n_d に対し、条件 4.1 を満たすが、条件 4.2 を満たさないノードの集合
- $\mathcal{V}_b^{(d)}$: $\mathcal{V}_u^{(d)}$ と $\mathcal{V}_a^{(d)}$ のどちらにも含まれないノードの集合 (条件 4.1 を満たさないノードの集合)

ここでは、リンク e の故障時に n_d への経路更新を、$\mathcal{V}_u^{(d)}$ 中のノードのみが行い、他のノードは行わないと仮定する。これらの各集合間の関係を、図 4.4 に示す。

ここで、後の定理の証明に用いるためにいくつかの補題とその証明を示す。

補題 4.1 n_d は $\mathcal{V}_b^{(d)}$ 中のノードである。

証明 4.1 ノード n_d 自体は、n_d に対する条件 *4.1* を満たさないことは自明である。よって、n_d は $\mathcal{V}_b^{(d)}$ の要素である。 □

補題 4.2 リンク e に接続し、かつ e の故障前における n_d までの最短パスに e が含まれるノードを n_r とする。このとき、n_r は集合 $\mathcal{V}_u^{(d)}$ の要素である。

証明 4.2 n_r に関する定義から、n_r は、n_d に対して条件 *4.1* を満たす。n_r が、n_d に対して条件 *4.2* を満たすことは自明である。それゆえ、n_r は集合 $\mathcal{V}_u^{(d)}$ 中の要素である。 □

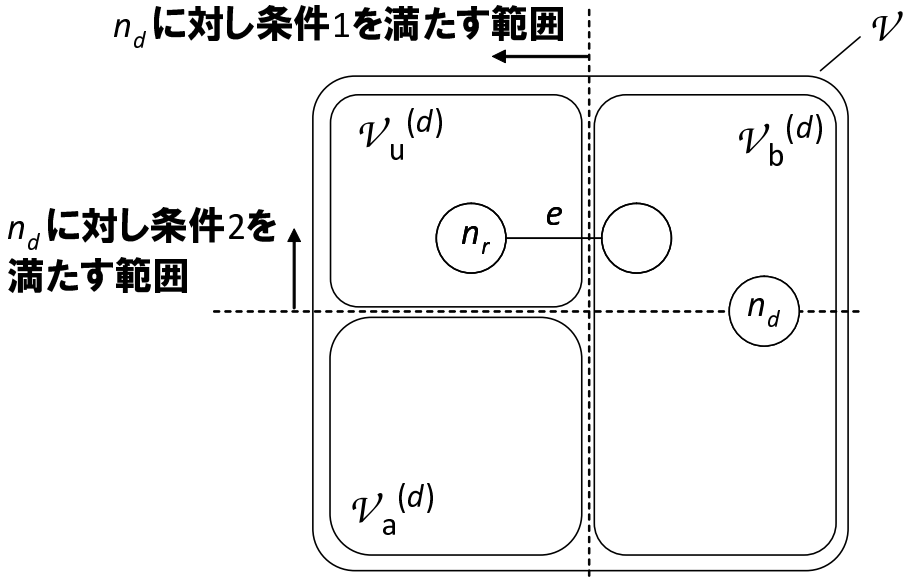

図 4.4: 各集合間の関係

次に、部分集合 $\mathcal{V}_u^{(d)}$、$\mathcal{V}_a^{(d)}$ および $\mathcal{V}_b^{(d)}$ のいずれかのノードから送信されるパケットに対して、それぞれ以下の三つの補題の証明を行う。

補題 4.3 $\mathcal{V}_b^{(d)}$ 中のいずれかのノードから転送を開始された n_d 宛のパケットは、その宛先である n_d まで到達することができる。

証明 4.3 補題 *4.1* によって、n_d は $\mathcal{V}_b^{(d)}$ 中の要素である。$\mathcal{V}_b^{(d)}$ 中の全てのノードは n_d に対して条件 *4.1* を満たしていないことから、$\mathcal{V}_b^{(d)}$ 中の任意のノードから n_d までの最短パスは e を含まない。つまり e が故障しているか否かに関わらず、この最短パスに変化はない。それゆえ $\mathcal{V}_b^{(d)}$ 中のノードから転送を開始されたパケットは、宛先まで到達可能である。 □

補題 4.4 $\mathcal{V}_u^{(d)}$ 中のいずれかのノードから転送を開始された n_d 宛のパケットは、$\mathcal{V}_b^{(d)}$ 中のいずれかのノードへ到達する。

証明 4.4 リンク e の故障発生後における $\mathcal{V}_u^{(d)}$ 中の任意のノードから n_d への最短パスについて考える。$\mathcal{V}_u^{(d)}$ 中のすべてのノードは n_d への経路を更新するので、

$\mathcal{V}_a^{(d)}$ 中のノードは n_d 宛のパケットをこの最短パスに沿って転送する。それゆえ、$\mathcal{V}_u^{(d)}$ 中の各ノード間でルーティングループは発生しない。

次に、この最短パスは $\mathcal{V}_a^{(d)}$ 中のいかなるノードも含まないことを示す。補題 *4.2* によれば、n_r は $\mathcal{V}_u^{(d)}$ 中のノードである。$\mathcal{V}_u^{(d)}$ 中のノードはすべて、n_r から n_d への最短パス上のノードであるため、$\mathcal{V}_u^{(d)}$ 中の任意のノードから n_d への最短パスは、n_r から n_d への最短パスの一部である。$\mathcal{V}_a^{(d)}$ の定義から、$\mathcal{V}_a^{(d)}$ 中のすべてのノードは条件 *4.2* を満たしていない。もしノードが n_r から n_d への最短パス上のノードであれば、条件 *4.2* を満たす。それゆえ、$\mathcal{V}_u^{(d)}$ 中の任意のノードから n_d までの最短パス上のすべてのノードは $\mathcal{V}_u^{(d)}$ もしくは $\mathcal{V}_b^{(d)}$ 中のノードであり、$\mathcal{V}_a^{(d)}$ 中のノードではない。

それゆえ、$\mathcal{V}_u^{(d)}$ 中の任意のノードから転送が開始されたパケットは、$\mathcal{V}_b^{(d)}$ 中のいずれかのノードへ到達する。 □

補題 4.5 $\mathcal{V}_a^{(d)}$ 中のいずれかのノードから転送を開始された n_d 宛のパケットは、$\mathcal{V}_u^{(d)}$ もしくは $\mathcal{V}_b^{(d)}$ 中のいずれかのノードへ到達する。

証明 4.5 $\mathcal{V}_a^{(d)}$ 中のすべてのノードは n_d への経路を更新していないので、n_d 宛のパケットは故障前の n_d までの最短パスに沿って転送される。それゆえ、$\mathcal{V}_a^{(d)}$ 中のノード間ではルーティングループは生じない。また、補題 *4.1* により $\mathcal{V}_a^{(d)}$ は n_d を含んでいない。

それゆえ $\mathcal{V}_a^{(d)}$ 中のいずれかのノードから転送を開始されたパケットは $\mathcal{V}_u^{(d)}$ または $\mathcal{V}_b^{(d)}$ のいずれかのノードに到達する。 □

これらの補題を用いると、次の定理が証明できる。

定理 4.1 ネットワーク中においてリンク e に故障が発生したとき、$\mathcal{V}_u^{(d)}$ 中のノードのみ n_d への経路を更新すれば、ネットワーク中の任意のノードからノード n_d へのパケット到達性を保証できる。

証明 4.6 補題 *4.5* により、$\mathcal{V}_a^{(d)}$ 中の任意のノードから転送を開始された n_d 宛のパケットは、$\mathcal{V}_u^{(d)}$ もしくは $\mathcal{V}_b^{(d)}$ 中のいずれかのノードに到達する。補題 *4.4* により、$\mathcal{V}_u^{(d)}$ 中の任意のノードから転送を開始された n_d 宛のパケットは、$\mathcal{V}_b^{(d)}$ 中の

いずれかのノードに到達する。補題 *4.3* により、$\mathcal{V}_b^{(d)}$ 中の任意のノードから転送を開始された n_d 宛のパケットは、宛先ノード n_d に到達可能である。それゆえ、ネットワーク中においてリンク e に故障が発生したとき、$\mathcal{V}_u^{(d)}$ 中のノードのみ n_d への経路を更新すれば、ネットワーク中の任意のノードから転送を開始された宛先 n_d のパケットは、宛先ノード n_d に到達可能である。 □

4.3.3 条件判定に必要な計算量

ここでネットワーク中のすべてのノード中から、上記の条件をノードを見つけるのに必要な計算量について、他の手法と比較を行う (表 4.1 参照)。計算量には、時間計算量と領域計算量の二つが存在するが、本論文では時間計算量について扱う。ネットワーク中のノード数、リンク数をそれぞれ N, L とする。網羅的アプローチでは、すべてのノードがすべてのリンクに対して代替経路表を用意する必要がある。この代替経路表の用意のために、始点ノード n_i と故障リンク e のすべての組み合わせに対して、つまり $N \times L$ 回の最短パスツリーの計算を必要とする。最短パスツリーアプローチは、条件 4.1 のみ判定を行うため、 N 回の最短パスツリーの計算が必要になり、最短パスツリー上のリンク数は $N-1$ であるため、ネットワーク全体で $N(N-1)$ 回の最短パスツリー計算が必要となる。これらに対し、提案手法では、条件 4.1 および条件 4.2 の両条件に対して判定を行う必要がある。条件 4.1 の判定には、ネットワーク全てのノードそれぞれを始点ノードとした最短パスツリーを計算する必要があり、その計算回数は N 回となる。また、条件 4.2 の判定には、ネットワーク全てのリンクに対し、リンク両端のノードを始点としたリンク故障時における最短パスツリーを計算する必要があり、その計算回数は $2L$ 回となる。

4.4 分散判定を実現する局所化アルゴリズム

4.4.1 ループ回避条件の分散判定

ここでは、前章で示した条件を満たすノードを判定する処理を、各ノードで分散して行う方法について述べる。

表 4.1: 計算コストの比較

	条件の判定	代替経路表数	
		ノード毎	合計
網羅的アプローチ	0	L	$N \times L$
最短パスツリーアプローチ	N	$N-1$	$N(N-1)$
提案手法	$N+2L$	別途評価 (5.3.2 参照)	

まず単純に条件 4.2 を各ノードごとに並行に行う方法を考える。D_r を n_r に接続するリンクの数とした場合、以下に示すように、この方法は各ノードにおいて $N + D_r$ 回の最短パスツリー計算を必要とする。e に接続する各ノードが n_r として条件の判定を行う場合を考えると、すべてのノードは条件 4.1 をそれぞれ独立に調べなければならない。それゆえ、各ノードが必要とする最短パスツリーの計算回数はそれぞれ、条件 4.1 を判定するために N 回、条件 4.2 を判定するために D_r 回となる。

一方、もし条件 4.1 の判定を並行して行う場合、各ノードは、$1 + 2L$ 回の最短パスツリー計算が必要になる。この場合、すべてのノードは条件 4.2 を独立に判定を行う必要がある。一般に $L > N$ かつ $L > D_r$ であるため、前者が選択される。つまり各ノードにおいて $N + D_r$ 回の最短パスツリーの計算が必要となる。

最短パスツリーの計算コストは高いため、この計算回数をすべてのノードに行わせることは多大なコスト増につながる。そこで、両条件の判定をより少ない計算コストで行う方法を提案する。この方法では、最短パスにおけるパス長の比較を用いることで、より少ない最短パスツリーでの条件判定を可能とする。この方法の説明に用いる表記を表 4.2 に示す。この表中の最後の七つは本方法において重要なメトリックを表しており、それぞれの関係を図 4.5 に示す。

まず初めに、条件 4.1 を満たすための必要条件を示す。リンク e に接続するノード n_r は n_i から n_d への最短パス上のノードであるため、以下の式が成り立つ。

$$m_{i\to d} = m_{i\to r} + m_{r\to d}. \tag{4.1}$$

次に条件 4.2 が成り立つための必要十分条件を示す。ノード n_r が条件 4.2 を満た

表 4.2: 記号の説明

$\mathcal{V}$	=	ネットワーク中のノードの集合
$\mathcal{E}$	=	ネットワーク中にリンクの集合
e	=	故障リンク
$SPF(n, \mathcal{V}, \mathcal{E})$	=	ネットワーク $\{\mathcal{V}, \mathcal{E}\}$ における 始点ノード n からの最短パスツリー
$RevSPF(n, \mathcal{V}, \mathcal{E})$	=	ネットワーク $\{\mathcal{V}, \mathcal{E}\}$ における 始点ノード n からの逆方向最短パスツリー
$Metric(n_s, n_d, \mathcal{T})$	=	最短パスツリー $\mathcal{T}$ から求まる n_s から n_d へのメトリック
$Nexthop(n_s, n_d, \mathcal{T})$	=	最短パスツリー $\mathcal{T}$ から計算される n_s 上の経路表において、宛先ノード n_d に対する次転送先ノード
$Link(n_i, n_j)$	=	n_i, n_j 間のリンク
$\mathcal{T}_r$	=	$SPF(n_r, \mathcal{V}, \mathcal{E})$
$\mathcal{T}_i$	=	$SPF(n_i, \mathcal{V}, \mathcal{E})$
$\mathcal{T}'_r$	=	$SPF(n_r, \mathcal{V}, \mathcal{E} - \{e\})$
$\widetilde{\mathcal{T}}_r$	=	$RevSPF(n_r, \mathcal{V}, \mathcal{E})$
$m_{i \to d}$	=	$Metric(n_i, n_d, \mathcal{T}_i)$
$m_{i \to r}$	=	$Metric(n_i, n_r, \mathcal{T}_i)$
$m_{r \to d}$	=	$Metric(n_r, n_d, \mathcal{T}_r)$
$m'_{r \to d}$	=	$Metric(n_r, n_d, \mathcal{T}'_r)$
$m'_{r \to i}$	=	$Metric(n_r, n_i, \mathcal{T}'_r)$
$m'_{i \to d}$	=	$Metric(n_i, n_d, \mathcal{T}'_i)$
$\widetilde{m}_{i \to r}$	=	$Metric(n_i, n_r, \widetilde{\mathcal{T}}_r)$

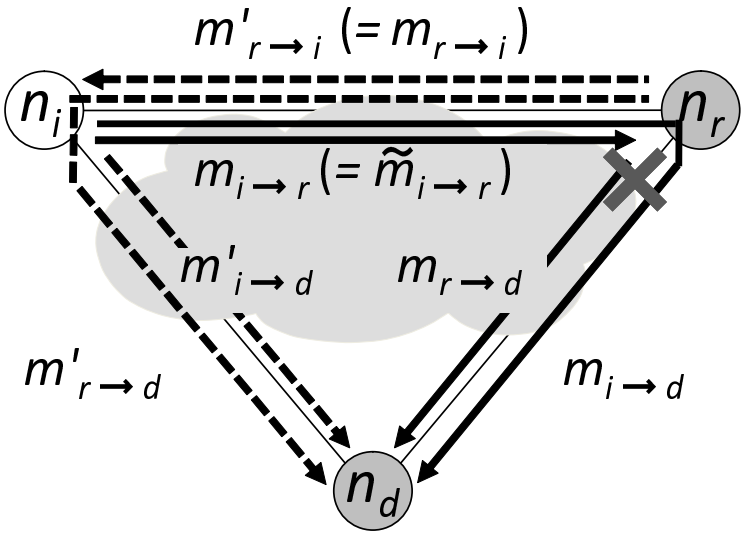

図 4.5: 各ノード間メトリックの関係

すとき、以下の式が成り立つ。逆もまた真である。

$$m'_{r\to d} = m'_{r\to i} + m'_{i\to d}. \tag{4.2}$$

式 (4.1) と式 (4.2) のあいだの関係を示す次の不等式を用いる。始点ノードから終点ノードへの最短パスにおけるメトリックは、故障前の方が故障後よりも小さいかもしくは等しいことは自明なので、以下の不等式が成り立つ。

$$m_{i\to d} \le m'_{i\to d}. \tag{4.3}$$

式 (4.1) と式 (4.2) を式 (4.3) に代入した結果、以下の不等式が得られる。

$$m_{i\to r} + m_{r\to d} \le m'_{r\to d} - m'_{r\to i}. \tag{4.4}$$

この式を変形すると、以下のようになる。

$$m_{i\to r} + m'_{r\to i} \le m'_{r\to d} - m_{r\to d}. \tag{4.5}$$

次に不等式 (4.5) を以下の式を用いて変形を行う。リンク e の故障は n_r から n_i へのパスには影響を与えないので、以下の式が成り立つ。

$$m'_{r\to i} = m_{r\to i}. \tag{4.6}$$

n_i を始点とした最短パスツリー $\mathcal{T}_i$ から計算される $m_{i\to r}$ は、n_r を宛先とする逆方向最短パスツリー $\widetilde{\mathcal{T}}_r$ から得られる $\widetilde{m}_{i\to r}$ と同じ値となる。

$$m_{i\to r} = \widetilde{m}_{i\to r}. \tag{4.7}$$

式 (4.6) と式 (4.7) を式 (4.5) に代入すると、最終的に以下の式が得られる。

$$\widetilde{m}_{i\to r} + m_{r\to i} \leq m'_{r\to d} - m_{r\to d}. \tag{4.8}$$

リンク e の故障により、ノード n_r から n_d への最短パスツリーが変化し、これらのノード間のメトリックが変化する。式 (4.8) の右辺はこのメトリックの差分を表している。左辺の第二項の計算には、故障前の最短パスツリーを用いればよい。また左辺の第一項は、n_r を終点とした逆方向最短パスツリーから求めることができる。このように、式 (4.8) は、三つの最短パスツリー ($\mathcal{T}_r$, $\mathcal{T}'_r$ および $\widetilde{\mathcal{T}}_r$) から計算することができる。これらの三つのツリーはすべて、ノード n_r を始点もしくは終点とした最短パスツリーと逆方向最短パスツリーとなっている。

経路表を更新すべきすべてのノードは、不等式 (4.8) を満たす。ノードに接続しているリンクの数を D_r としたとき、この不等式の評価には $2 + D_r$ 回の最短パスツリー計算を必要とする。これは $\mathcal{T}_r$ および $\widetilde{\mathcal{T}}_r$ の計算と、さらに n_r が持つ各リンク故障時における $\mathcal{T}'_r$ の計算が D_r 回、それぞれ必要となる。一般的に各ノードは、通常時に使用する経路表を作成するために、$\mathcal{T}_r$ を計算する。これを差し引くと各ノードは、条件の判定、すなわち 4.8 の評価のために、$1 + D_r$ 回の最短パスツリー計算を余分に必要とする。つまり、不等式 (4.8) を使用することで、$N + D_r$ 回必要だった最短パスツリー計算の回数を分散処理時には $1 + D_r$ 回まで削減することができる。

以上を考慮すると、各ノードは $1 + D_r$ 回の最短パスツリー計算が必要であるため、ネットワーク中の全ノードにおける計算量の合計は次の式で表される。

$$\sum_{i\in\mathcal{N}}(1 + D_i) = N + \sum_{i\in\mathcal{N}} D_i = N + 2L \tag{4.9}$$

この式は、最短パスツリー計算の必要回数が分散処理と一括処理 (表 4.1) で、等しいことを示している。つまり、不等式 (4.8) を用いた分散判定には、一箇所で集中して行う場合と比べて、オーバーヘッドは存在しない。

Enumerating algorithm(n_r, e)

1: $\mathcal{T}_r \Leftarrow SPF(n_r, \mathcal{V}, \mathcal{E})$
2: $\mathcal{T}'_r \Leftarrow SPF(n_r, \mathcal{V}, \mathcal{E} - \{e\})$
3: $\widetilde{\mathcal{T}}_r \Leftarrow RevSPF(n_r, \mathcal{V}, \mathcal{E})$
4: **for** $\forall n_d \in \mathcal{V}$ **do**
5: $n_k \Leftarrow Nexthop(n_r, n_d, \mathcal{T}_r)$
6: **next if** $e \neq Link(n_r, n_k)$
7: $\mathcal{V}_u^{(d)} \Leftarrow \phi$
8: $n_i \Leftarrow n_r$
9: $t_d \Leftarrow Metric(n_r, n_d, \mathcal{T}'_r) - Metric(n_r, n_d, \mathcal{T}_r)$
10: **repeat**
11: $\mathcal{V}_u^{(d)} \Leftarrow \mathcal{V}_u^{(d)} \cup \{n_i\}$
12: $n_i \Leftarrow Nexthop(n_i, n_d, \mathcal{T}'_r)$
13: **until** $Metric(n_i, n_r, \widetilde{\mathcal{T}}_r) + Metric(n_r, n_i, \mathcal{T}_r) > t_d$
14: **end**
15: **return**

図 4.6: 局所化アルゴリズム

4.4.2 局所化アルゴリズム

4.3.1 で示した条件 4.1 および条件 4.2 の両条件を判定するためのアルゴリズムを、図 4.6 に示す。このアルゴリズムは、ノード n_r および n_r に接続されるリンク e に対して、条件 4.1 および条件 4.2 の両条件を満たすノードを求め、その結果を宛先毎にノードの集合 $\mathcal{V}_u^{(d)}$ に格納して返す。

ステップ 5, 6 は、式 (4.1) を満たすが条件 4.1 を満たさないケースを排除するための処理である。式 (4.1) を満たすことは、条件 4.1 が成り立つための必要条件であるが、十分条件ではない。このため、式 (4.1) を満たすケースの中には、n_i から n_d への最短パスが n_r を含むが、e を含まないケースが存在する。ステップ 5 では、故障前における n_r を始点とする最短パスツリー上での n_d への次転送先ノードを計算する。もし n_r とステップ 5 で求めた次転送先ノードとの間のリンク

が e でない場合、n_r, n_d および e の組み合わせにおいて条件 4.1 を満たすノードは存在しない。このとき、この n_d に対するステップ 7 から 13 は行わない事で、式 (4.1) を満たすが条件 4.1 を満たさない場合に、以降の処理実施を防いでいる。

4.5 結言

本章では、リンク故障時において、経路更新を必要とするノードを特定する局所化アルゴリズムの提案を行った。本アルゴリズムは、3 章で示した四つ到達不能要因のうち、ルーティングループ発生を防ぐために経路更新が必要となるノードを特定することができる。

本章ではまず、経路更新を必要とするノードを特定するための条件として、以下の二つの条件を提示した。

- 経路更新を必要とするノードは、故障前のトポロジーにおいて、自身から宛先ノードへの最短パス中に故障リンクを含む。
- 経路更新を必要とするノードは、故障後のトポロジーにおいて、故障リンクに接続しているノードから宛先ノードへの最短パス上に存在する。

次に、これらの条件を共に満たすノードのみが経路更新を行うことにより、パケット到達性を復旧できることを証明した。さらに、これらの条件の判定処理を、ネットワーク中の各ノードにより分散判定を行う方法を示した。そして、この分散判定は、一箇所で集中して判定を行う一括処理と比較し、計算量の面でオーバーヘッドが存在しないことを示した。最後に、この分散判定を行うための手順として局所化アルゴリズムを示した。

本アルゴリズムは、IP 転送方式の変更を必要とせず、かつ分散処理可能な高速復旧手法を実現するために、有用であると考えられる。本アルゴリズムを用いた高速復旧手法として、5 章で事前計算型経路更新手法を、また 6 章で故障箇所高速迂回手法を提案する。

第5章　局所化アルゴリズムを用いた事前計算型経路更新手法

5.1　緒言

IP ネットワークの可用性向上を実現するためには、故障発生時における復旧時間の短縮が重要である。故障からの復旧動作において、ネットワーク中の各ノードは、故障発生後のトポロジーに基づいて、経路表の再構築を行う。しかし、経路表再構築にかかる時間は、ある商用ネットワークにおいて 100 ミリ秒から 400 ミリ秒もの時間がかかることが報告されている [Ian04]。この時間が、故障からの復旧時間に対して大きな影響を与えている。

本章では、故障後における経路表再構築時間の削減を実現する事前計算型経路更新手法の提案を行う。提案手法では、リンク故障後に使用する代替経路表を予め計算しておき、故障発生直後即座に切り替えを行うことで、故障からの復旧時間を短縮する。しかし、故障毎に使用できる代替経路表は異なるため、ネットワーク中の全てのリンク故障へ対応する場合、大量の代替経路表を用意しなければならない。そこで、提案手法では、前章で提案した局所化アルゴリズムを用いて、用意すべき代替経路表数を削減する。

本章の構成は以下の通りである。5.2 において、事前計算型経路更新手法の提案を行い、5.3 において評価を行う。さらに 5.4 において、提案手法の適用範囲をノード故障にまで広げるための議論を行ったうえで、最後に 5.5 においてまとめを行う。

5.2 事前計算型経路更新手法の提案

本節では、事前計算型経路更新手法の提案を行う。提案手法では、リンク故障時に用いる代替経路表を事前に計算しておくことで、故障時の復旧時間短縮を実現する。しかし、使用できる代替経路表は故障毎に異なるため、ネットワーク中の全てのリンク故障へ対応する場合、大量の代替経路表が必要になる。その数を削減するために、提案手法では、用意すべき代替経路表をパケット到達性を復旧させるのに必要なもののみに限定する。提案手法では、4 章で提案した局所化アルゴリズムを用いて、あるリンク故障発生時においてパケット到達性を復旧させるために経路更新が必要となるノードを特定し、それらのノードのみが該当リンクの故障時に使用する代替経路表を準備する。このことにより、全てのリンク故障に対してそれぞれ代替経路表を用意する場合と比べ、用意すべき代替経路表の数を大幅に削減できる。

提案手法では、まず以下の手順により、局所化アルゴリズムを用いて故障後に経路更新が必要となるノードを特定し、それらのノードに対して代替経路表の準備を指示する通知を行う。

1. 正常時に使用する経路表を計算する。
2. 自身に接続するリンク e に対して図 4.6 に示された局所化アルゴリズムを適用し、宛先 n_d 毎に $\mathcal{V}_u^{(d)}$ の数え上げを行う。
3. $\mathcal{V}_u^{(d)}$ 中の全ての要素 n_u に対して、リンク e 故障時における宛先 n_d への代替経路の準備を行うよう通知を行う。この処理を、全ての宛先に対して繰り返す。
4. ステップ 2, 3 の手順を、自身に接続するすべてのリンクに対して繰り返す。

次に、上記ステップ 3 で通知を受けた各ノード n_u は、以下の手順で代替経路表の作成を行う。

5. 各ノード n_r からの受信メッセージを、故障リンク e 毎にまとめる。
6. e を取り除いたトポロジーをベースに最短パスツリーを作成し、n_d への代替経路を計算する。

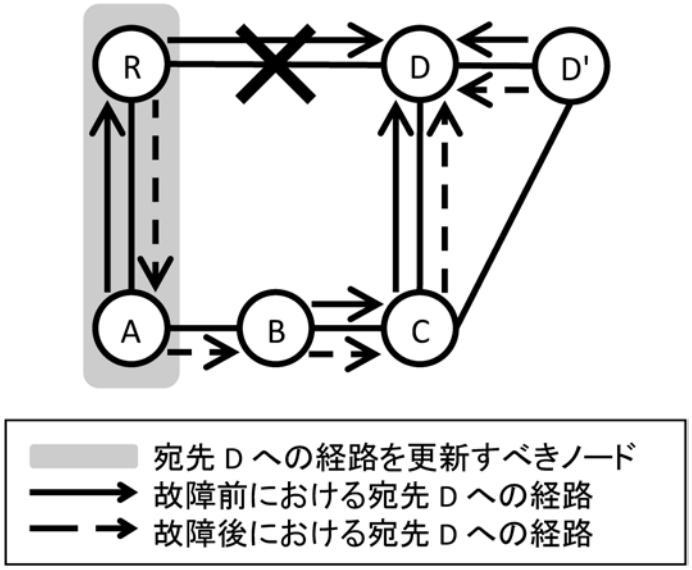

図 5.1: 提案手法の動作例

7. 計算によって得られた代替経路を、リンク e の故障時に使用する代替経路表に格納する。

8. ステップ 6, 7 を、すべての e に対して実施する。

ノード n_r が自身に接続するリンク e の故障を検知したとき、n_r はネットワーク中のノードへ、適切な代替経路表への切り替えを促す故障通知を送る。現在運用されている IP ネットワークでは、一般に OSPF などの経路制御プロトコルが用いられており、提案手法はこれらと併用する。そのため、代替経路表への切替を促す故障通知には、OSPF におけるリンク状態広告など既存の経路制御プロトコルにおける故障通知機能が利用できる。

図 5.1 中の R–D 間のリンクが故障した場合における提案手法の動作例を説明する。R–D 間のリンクに接続するノード R は正常時に使用する経路表の計算を行った後、このリンクの故障に対して代替経路を用意する必要があるノードの特定を行う (上記のステップ 1, 2)。このとき、宛先ノード D に対して経路更新が必要となるのはノード R, A であるため、ノード R はノード A に対して、宛先ノード D への代替経路を用意するように通知する (上記のステップ 3)。通知を受けたノード A は、R–D 間のリンクを取り除いたトポロジーをベースに最短パスツリー

を計算し、ノード D への代替経路を計算し、R–D 間のリンク故障時に使用する代替経路表に格納する (上記ステップ 6, 7)。実際に R–D 間のリンクが故障した場合、故障を検知したノード R が、ノード A に故障通知を行う。故障通知を受信したノード A は、パケット転送に用いる経路表を、事前に用意した代替経路表に切り替える。R–D 間のリンクは、ノード R からノード D' への最短パスにおいても使用されている。このため、ノード R は宛先ノード D' に対しても同様に、上記の処理を実施する。

5.3 提案手法の評価

提案手法を採用する場合のコストを評価するために、作成が必要となる代替経路表の数と、これらの代替経路表を計算するのに必要な計算リソースの見積もりを行う。

5.3.1 評価で用いるネットワークトポロジー

ノード数や平均次数などさまざまなパラメータによる違いが提案方式の効果に与える影響を調べるため、BRITE [Med01] により生成されるネットワークトポロジーを用いて評価を行った。BRITE とは、シミュレーションを行うためのトポロジーを生成するために広く用いられているソフトウェアである。使用するネットワークモデルとしては、ネットワークの評価に広く用いられている Barabasi-Albert (BA) モデル [Bar99] を使用した。

議論を単純化するために、ネットワークトポロジーの生成に関して以下の三つの仮定を置いた。

- すべてのリンクはポイントトゥポイント型であるとする。
- すべてのリンクは双方向通信可能であり、そのコストは対称であるとする。
- ネットワーク中のすべてのノードはそれぞれアドレスを一つ持ち、リンクはアドレスを持っていないとする。つまり、各ノードが持つ経路表には、宛先としてリンクではなく各ノードが登録される。

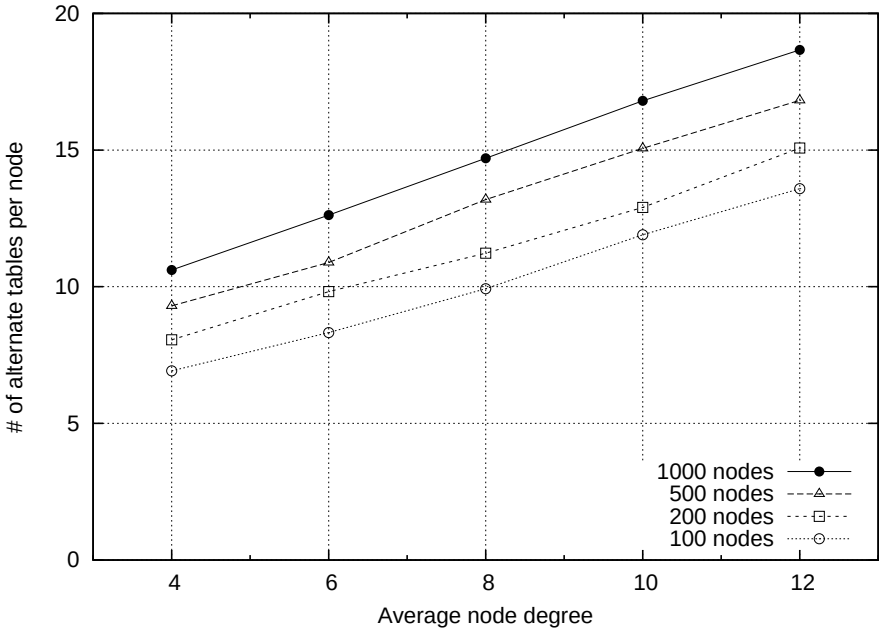

図 5.2: 一ノードあたりの代替経路表数 (平均)

以下、5.3.2 から 5.3.5 では、BA モデルにおけるさまざまなパラメータが我々の提案方式にどのように影響を与えるかについて調べる。5.3.6 では、GÉANT2 [GEANT2], Internet2(Abilene) [Internet2] および SINET3 [SINET3] と言った実際のネットワークトポロジーを用いた評価を行う。ただし、次数が 1 であるノードであるスタブノードは、評価を行う際にトポロジー中からは取り除いている。その理由は、スタブノードに接続するリンクはただ一つのみであり、ネットワーク中の任意のリンク故障時はスタブノードの経路表に影響を及ぼさないためである。

5.3.2 事前計算で必要となる代替経路数

提案手法において、各ノードが用意すべき代替経路表数を図 5.2 から図 5.4 に示す。評価にはノード数を 100 から 1000、平均次数を 4 から 12 とする BA モデルのトポロジーを使用した。図 5.2 は一ノードあたりの代替経路表数の平均、図 5.4 はその最大値を示す。また図 5.3 は代替経路表数が少ないノードから順に全体の 95% の範囲内における最大値を示す。

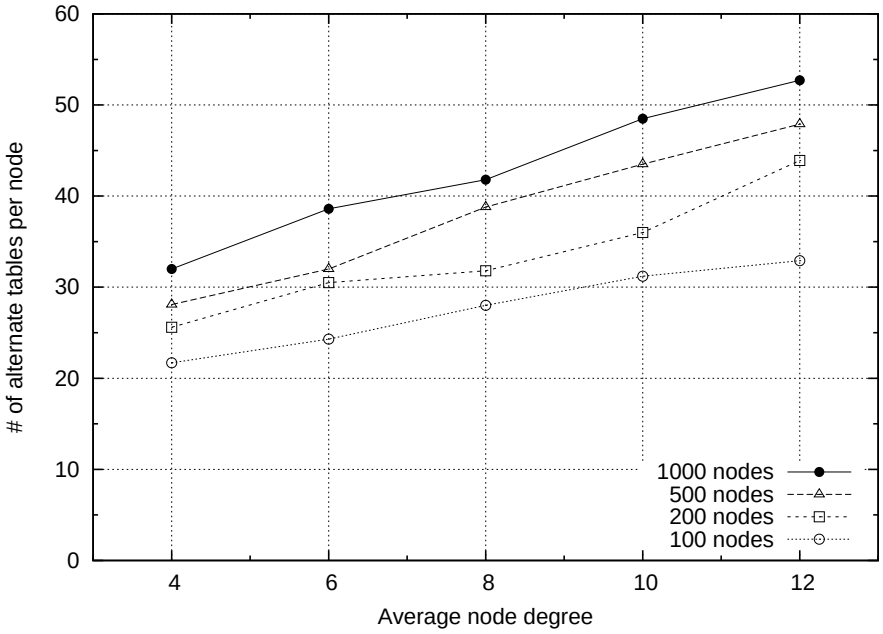

図 5.3: 一ノードあたりの代替経路表数 (下位 95% 内での最大値)

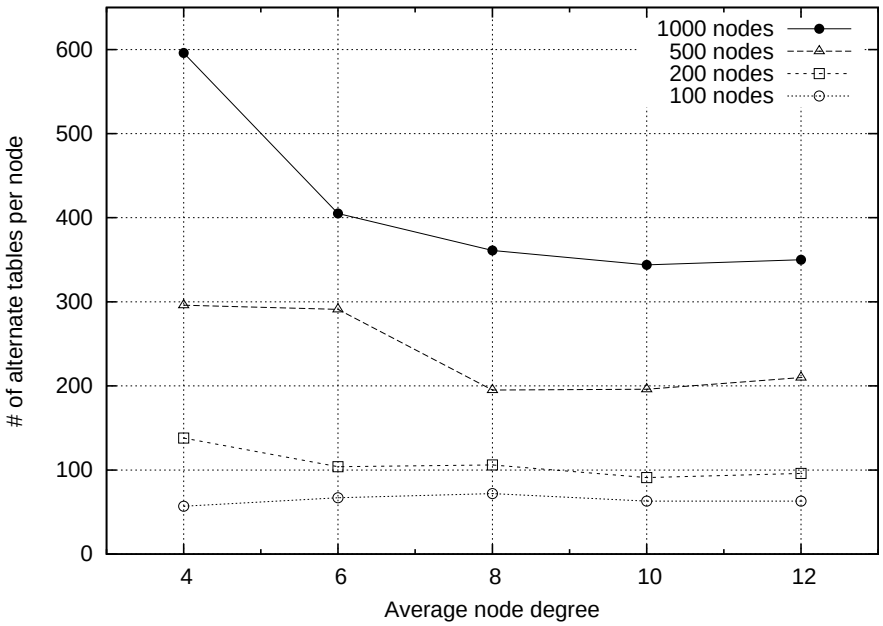

図 5.4: 一ノードあたりの代替経路表数 (最大値)

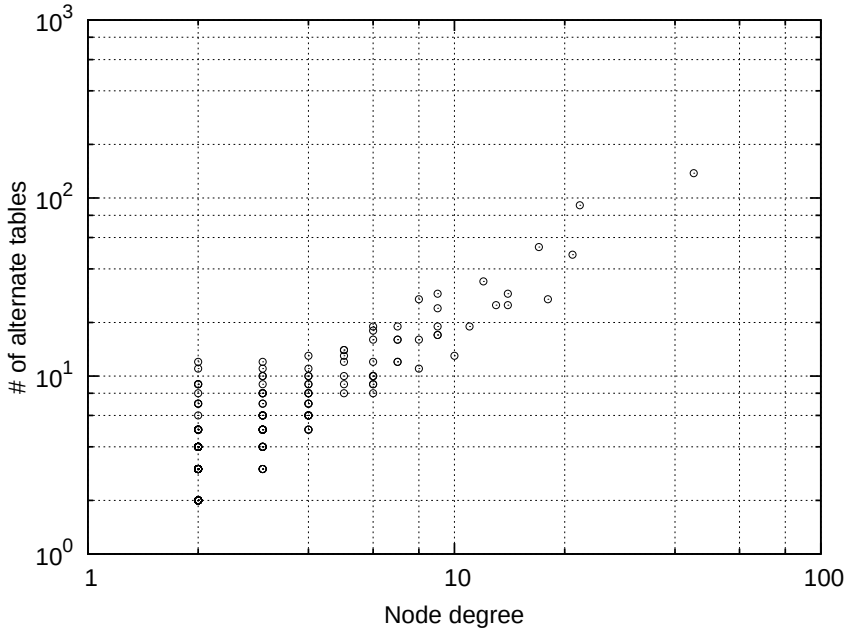

図 5.5: ノード次数と代替経路表数の関係 (200 ノード、平均次数 4)

図 5.2 からは、ノード数および平均次数の増加に伴って必要代替経路表数の平均は増加しているが、その値は比較的小さいことが分かる。例えば、1000 ノード、平均次数 12 のネットワークにおいて、提案手法における一ノードあたりの平均代替経路表数は、たかだか 19 程度に抑えられている。実際のネットワークの平均次数は 4 程度 (5.3.6 参照) であるため、この値よりさらに小さくなることが予想される。

しかし、図 5.4 を見ると、1000 ノード、平均次数 12 のネットワークにおいて、最大 350 の代替経路表を必要とするノードが存在することが分かる。一方で、図 5.3 を見ると、全体の 95% のノードは代替経路表数が 50 未満で収まっていることも分かる。つまり、ごく一部のノードのみが多数の代替経路表を必要としていると言える。

次に、平均次数が 4 であるネットワーク中のそれぞれのノードにおける次数と代替経路表数の関係を調べた。ノード数が 200, 500, 1000 である各ネットワークにおける結果を、図 5.5 から図 5.7 に示す。これらの図を見ると、やはりノードの次数増加と共に、代替経路表数も増加している。実際のネットワークにおいて

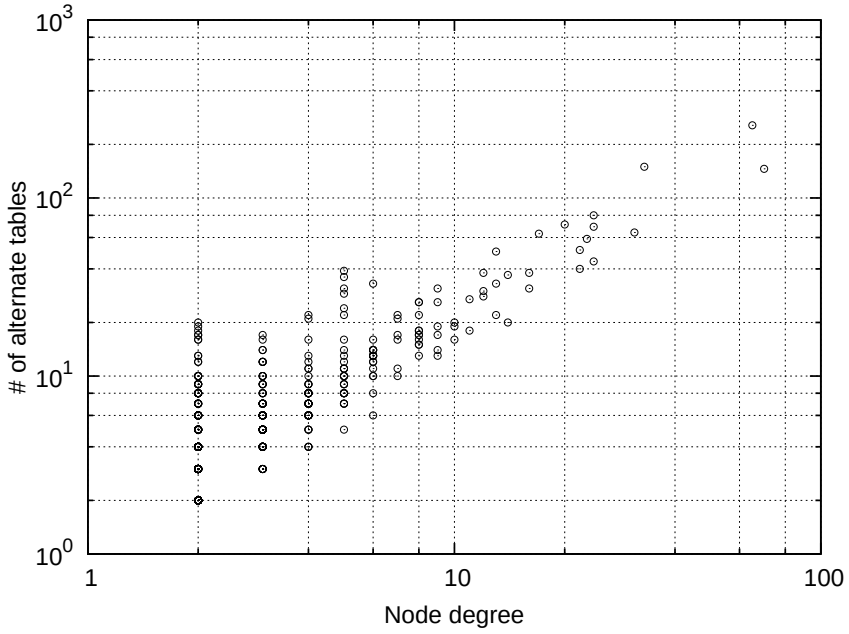

図 5.6: ノード次数と代替経路表数の関係 (500 ノード、平均次数 4)

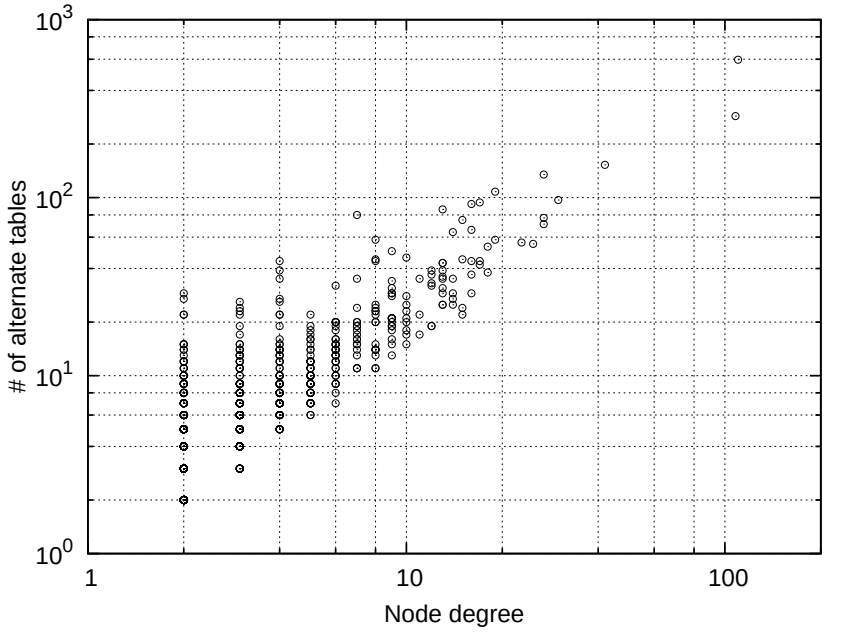

図 5.7: ノード次数と代替経路表数の関係 (1000 ノード、平均次数 4)

次数の大きなノードとは、インターフェイスを多く持つルータすなわちコアルータに相当し、次数の少ないノードはエッジルータに相当する。コアルータの方がより多くの代替経路表を用意する必要があることを示している。

図 5.7 は、次数 110 のコアルータが必要とする代替経路表数が 596 であることを示している。この数の代替経路表の計算、格納のために必要となるリソースが、実際のルータ装置において現実的であるかについては、次章以降で議論を行う。

5.3.3 代替経路表格納のためのメモリ量

ルータの持つメモリの主な用途は、経路表の格納である。経路表を格納するのに必要なメモリ量は経路表中の経路数に依存するため、ネットワーク中の各ルータは運用するネットワークの規模に応じた量のメモリが搭載されている。提案手法を採用する場合、代替経路表を格納するために、ルータにはさらに多くのメモリを搭載する必要がある。そこで、通常の経路表を格納するのに必要なメモリ量と比較して、代替経路表を格納するためはどの程度のモリ量が必要になるかを調べた。

図 5.8 から図 5.10 は、各ノードの次数と必要とするメモリ量の関係を示している。これらの図の縦軸は、通常の経路表を格納するのに必要なメモリ量を一単位として、その何倍のメモリ量を必要とするかを表している。

あるリンク故障により変化がある経路は全経路のうち一部のみであるため、変化のある経路のみを格納することで、全体のメモリ量を削減することができる。このように差分のみを格納することで必要なメモリ量を、通常の経路表のサイズと比べて最大でも約 5.3 倍程度にまで削減することができる。例えば 1000 ノードのネットワーク中のルータが全代替経路表を格納するのに必要とするメモリ量は、最大でも 184 キロバイト[1] 程度である。代替経路表をメモリに格納することを考えた場合、本方式実現に際して、これは問題となる値ではない。

[1]IPv6 の経路表における 1 エントリを、それぞれ 16 バイトの宛先プレフィックスおよび次転送先アドレス、1 バイトのプレフィックス長、2 バイトのインターフェイス識別子から構成されるとした場合

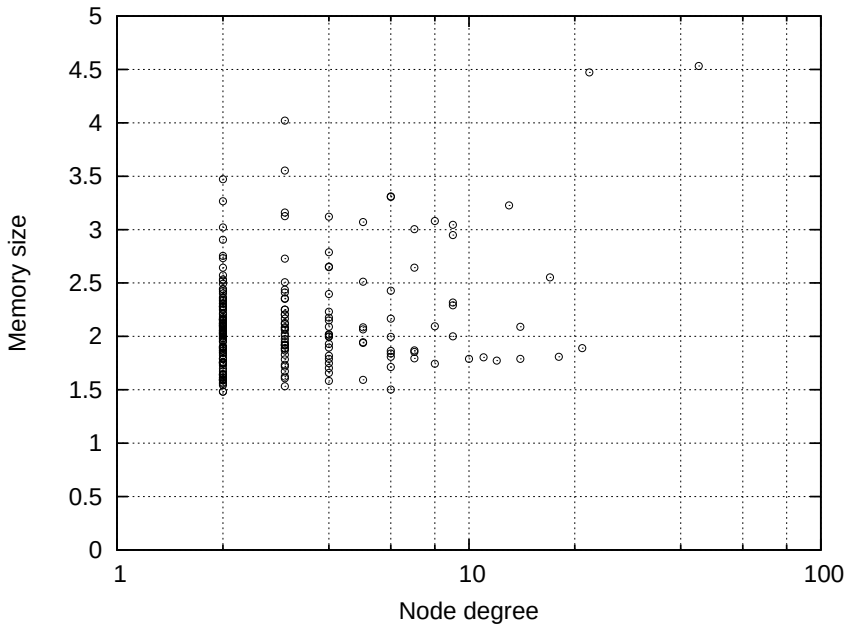

図 5.8: 代替経路表格納に必要なメモリ量 (200 ノード、平均次数 4)

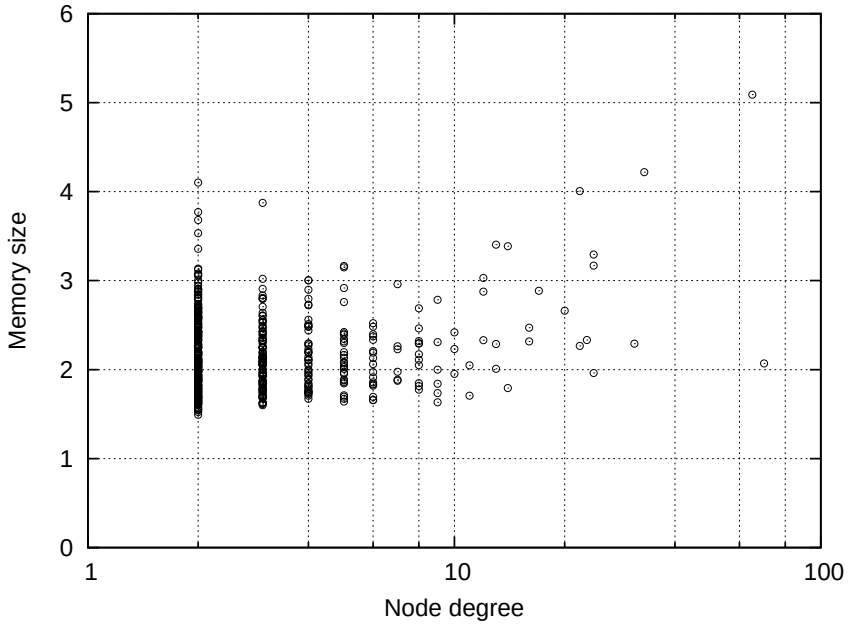

図 5.9: 代替経路表格納に必要なメモリ量 (500 ノード、平均次数 4)

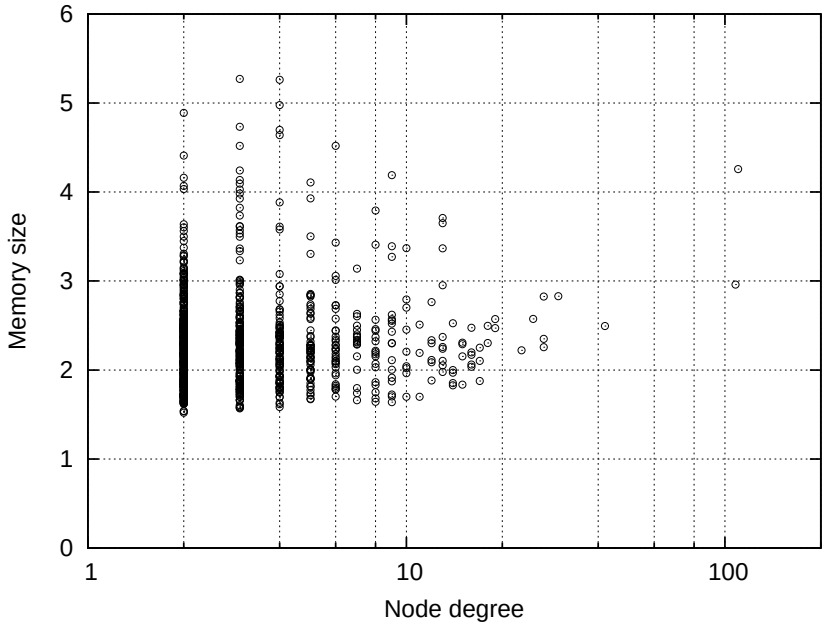

図 5.10: 代替経路表格納に必要なメモリ量 (1000 ノード、平均次数 4)

5.3.4 代替経路表算出のための計算コスト

代替経路表計算に用いるトポロジーと、通常の経路表計算に用いたトポロジーとの差は、ネットワーク中のあるリンクにおける故障の有無のみである。このため、通常の経路表計算に用いた最短パスツリーに対して、2.4.3 で述べた Incremental SPF [McQ80] による差分計算を行うことで、代替経路表の計算コストを削減することが可能である。

前節の結果における代替経路表すべてを Incremental SPF を用いて計算した場合のコストの合計を求め、通常の経路表を作成時に必要なコストと比較を行った結果を図 5.11 から図 5.13 に示す。比較にあたり、ネットワーク中のノード数は 200, 500, 1000、平均次数は 4 を仮定した。

ダイクストラ法による最短パスツリー算出計算は、代替経路表の数と同じだけ行う必要がある。しかし、Incremental SPF を使用しているため、一回一回の計算コストは全体を計算する場合と比べて大幅に削減されている。全ノード数、全リンク数がそれぞれ N, L であるとき、ダイクストラ法の計算量は $O((N+L) \times \log N)$ であることが知られている [Bar98]。Incremental SPF を用いて、元の最短パスツ

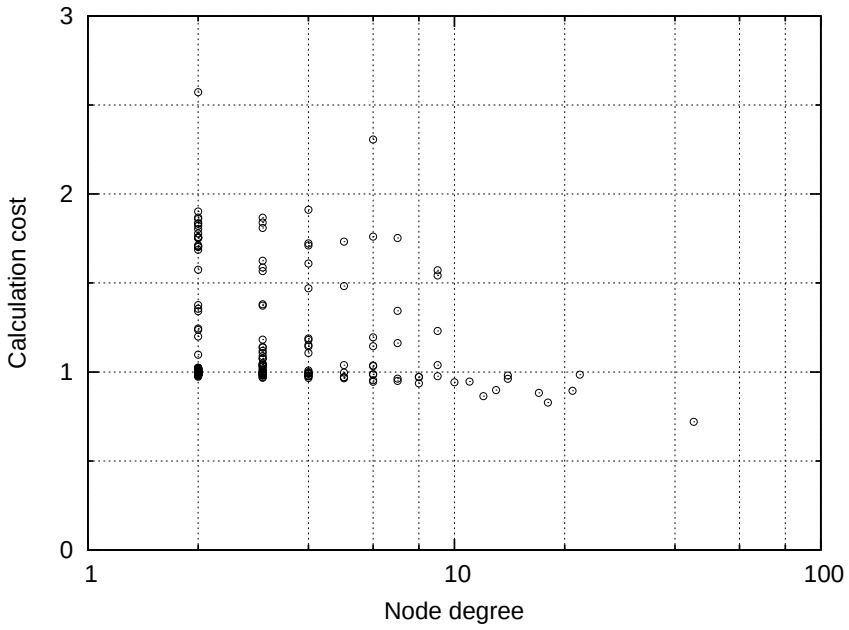

図 5.11: 代替経路表の計算コスト (200 ノード、平均次数 4)

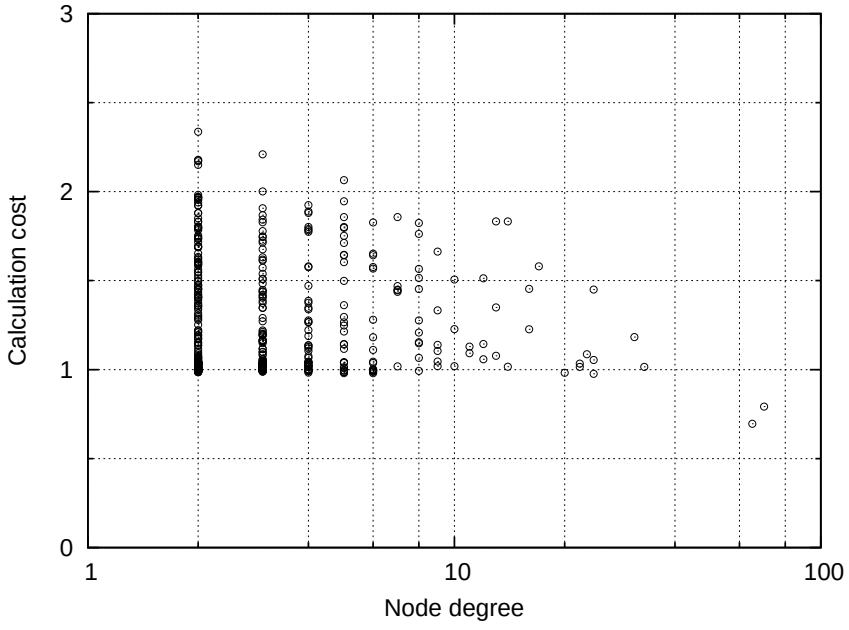

図 5.12: 代替経路表の計算コスト (500 ノード、平均次数 4)

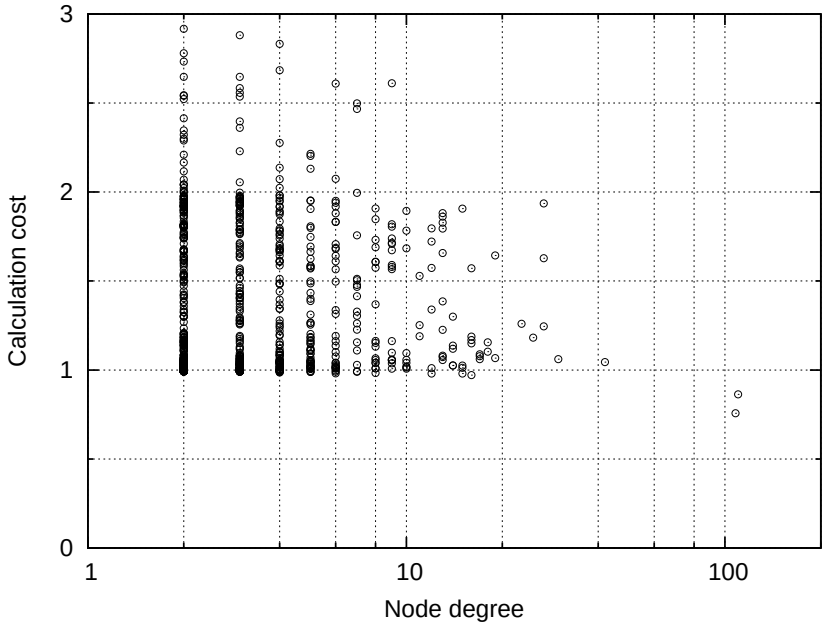

図 5.13: 代替経路表の計算コスト (1000 ノード、平均次数 4)

リーに対して、故障により影響を受けるノードに関する部分だけ再構築を行うことで、計算コストを大幅に削減することができる。

図 5.13 を見ると、最大の次数を持つノードにおける計算コストは、1 以下となっており、通常の経路表を一回計算するのに必要なコストよりも低い。最大の次数を持つノードは実際のネットワークではコアルータに相当し、多くのリンクと接続する。そのため、個々のリンク単一故障に対して影響を受けるノードの数が比較的限定される。その結果、Incremental SPF を使うことで、大幅に計算コストが削減出来ていると考えられる。

一方で次数の小さなノードにおいては、個々のリンク単一故障で影響を受けるノード、リンク数が比較的多くなるため、次数が大きなノードと比べて計算コストが比較的大きくなっている。しかし、それでも通常の計算におけるコストに対して、たかだか 3 倍程度に抑えられている。

また、図 5.11 から図 5.13 における最大値を比較してみると、ネットワークの規模にかかわらずその値が大きく変わっていないことが分かる。以上から、本方式において準備が必要となる代替経路表の計算コストは、大きな問題とならない

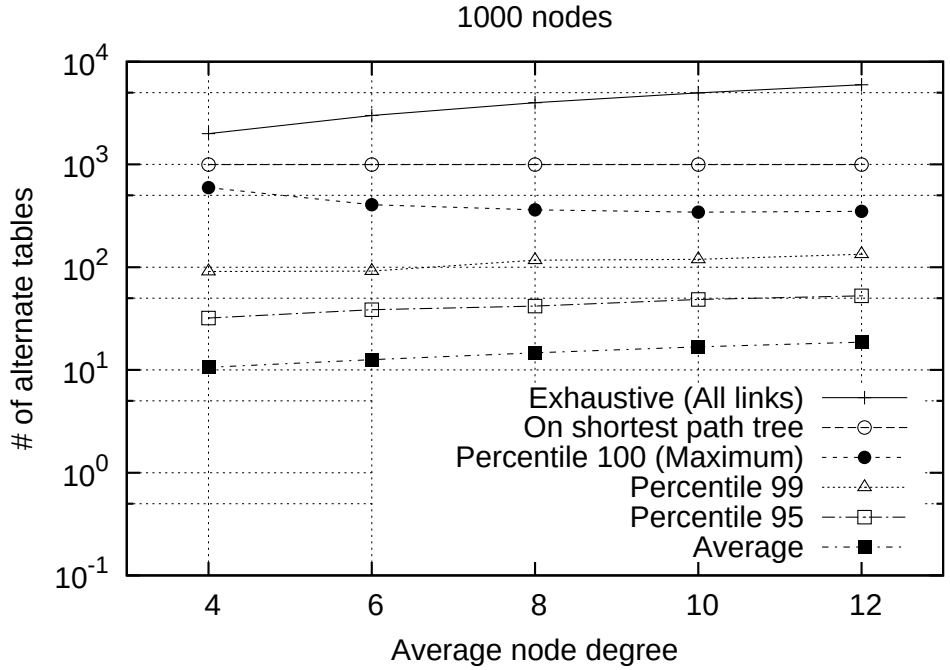

図 5.14: 従来手法との代替経路表数の比較

ものと考えられる。

5.3.5 従来手法との比較

図 5.14 に、1000 ノードのネットワークにおいて必要とされる代替経路表数を、提案手法と従来手法との間で比較した結果を示す。この図において、縦軸は代替経路表数、横軸はネットワークの平均次数をそれぞれ表している。従来手法として、4.2 において説明を行った網羅的アプローチおよび最短パスツリーアプローチを用いた。図中の Exhaustive は網羅的アプローチを、On shortest path tree は最短パスツリーアプローチを、それぞれ表している。

最短パスツリーアプローチは、最短パスツリー上の各リンクに対し、それぞれ故障時に使用する代替経路表を用意する。最短パスツリー上のリンク数は、ネットワーク中の全ノード数から 1 引いた値となる。つまり最短パスツリーアプローチにおける代替経路表数は、ネットワークの平均次数にかかわらず、999 となって

表 5.1: 実ネットワークにおける代替経路表数

	ネットワークのサイズ		ノード毎の代替経路表数 (全リンクに対する割合)		
	ノード数	平均次数	最小	平均	最大
GÉANT2	32	3.3	0 (0%)	3.8 (7.3%)	12 (23%)
Internet2	11	2.5	2 (14%)	4.5 (32%)	7 (50%)
SINET3	12	2.3	4 (29%)	6.3 (45%)	10 (71%)
BA(10)	10	3.4	2 (12%)	3.7 (22%)	7 (41%)
BA(30)	30	3.8	2 (3.5%)	4.8 (8.4%)	19 (33%)

いる。

図 5.14 で示した結果によれば、最短パスツリーアプローチと比較した場合、提案手法は必要とする代替経路表数の平均が 1/50 から 1/100 となっている。また提案手法における最大値でも、最短パスツリーアプローチにおける値の 60% 未満である。このように、提案手法は、最短パスツリーアプローチと比較して、必要とする代替経路表数が少ない。

5.3.6 実ネットワークへの適用結果

前節までの評価結果と同等な結果が実ネットワークに対しても得られるかを調べるために、実ネットワークである GÉANT2, Internet2, SINET3 と、同等のノード次数を持つ BA モデルのトポロジーとの比較を行った。提案手法をそれぞれのトポロジーに対して適用した場合に必要となる代替経路表数を、表 5.1 に示す。この表中において括弧で囲まれた数値は、代替経路経路表を必要とするリンクが、ネットワーク中の全リンクに対して占める割合を示す。

この表の結果を参照すると、実際のネットワークと BA モデルにおけるネットワークとでは大きな差がないことが分かる。つまり、提案手法は実際のネットワークにおいても、必要とする代替経路表数を大きく削減できることを示している。

5.4 ノード故障への対応に関する議論

ここまでは、ノード故障ではなく、リンク故障への対応に焦点をあてて議論を行ってきた。しかし、下位レイヤの支援による特別な故障検知メカニズムなしには、ノード故障かリンク故障かを区別することは一般に難しい。特に一般に広く用いられている L2 ネットワーク技術である Ethernet はこのような特別なメカニズムを持っていないので、L3 におけるキープアライブを用いた故障検知メカニズム、例えば OSPF における Hello や BFD [RFC5880] などが使用される。この場合、リンク故障とノード故障の区別がつかないため、すべての故障はノード故障として取り扱われる。提案手法においては、局所化アルゴリズムを以下のように変更することで、ノード故障を取り扱うことができる。

> n_f を故障ノード、 n_r をその隣接ノード、また $\mathcal{E}_f$ を n_f に隣接するすべてのリンクの集合であるとする。このとき、図 4.6 の 2 行目を、以下の式に置き換える。
>
> $$\mathcal{T}_r' \Leftarrow SPF(n_r, \mathcal{V} - \{n_f\}, \mathcal{E} - \mathcal{E}_f)$$

5.5 結言

IP ネットワーク中での故障発生時における復旧時間を短くするためには、故障後に使用する代替経路表の計算を事前に行っておくという手法が有効である。しかし、異なる故障箇所に対してそれぞれ代替経路表を用意しておく必要があるため、用意すべき代替経路表の数が膨大になってしまうという課題があった。そこで我々は、第 4 章において示した局所化アルゴリズムを使い、用意すべき代替経路表を必要最小限に限定する手法を提案した。この手法は、従来手法と比較して、以下の特徴がある。

- 従来手法と比べて、必要とする代替経路表数を平均で 1/100 に削減できる。
- 代替経路表の計算コストと格納メモリ量は、実際の IP ルータに実装を行う際の問題とならない程度に小さい。

- 提案手法は、IP 転送のメカニズムを変更していないため、従来の IP ネットワークへの適用が容易である。

この手法では、故障発生時の復旧時間のうち、経路表再計算時間の短縮を実現している。しかし、故障の通知時間に関しては考慮されていない。故障通知時間の短縮を実現する高速迂回手法については、次章で検討する。

第6章　局所化アルゴリズムを用いた故障箇所高速迂回手法

6.1　緒言

IP ネットワークの可用性を向上させるためには、故障発生時における復旧時間を短縮することが重要である。しかし、従来提案されている高速復旧手法の多くは、転送方式の変更もしくは集中計算が必要であり、実際の IP ネットワークに対してそのまま適用できなかった。そこで、前章では、転送方式を変更せずに、かつ分散処理が可能である高速復旧方式である事前計算型経路更新手法の提案を行った。この手法では、故障発生時に使用する代替経路表を事前に計算しておくことにより、復旧時間中の経路表更新時間の短縮を実現する。しかし、より高速な復旧を実現するためには、故障発生時にその故障をネットワーク中の各ノードに通知する故障通知時間の短縮が重要である。そこで本章では、トンネルを用いることで、故障通知を必要とせずにパケットの迂回を行う故障箇所高速迂回手法の提案を行う。

トンネル [RFC1853, RFC2784] とは、直接接続されていない二ノード間に仮想的に構築される通信路であり、一般的な IP ルータの大半はトンネルを構成する機能を有している。トンネルの始点となるノードは、受信したパケットに対して、トンネルの終点となるノードのアドレスを宛先とする IP ヘッダを付与する。この動作をカプセル化と呼び、トンネルの終点となるノードを終端ノードと呼ぶ。カプセル化されたパケットは、新たに付与された IP ヘッダが参照され、その宛先である終端ノードまで転送される。終端ノードに到達したパケットは、付与された IP ヘッダが取り除かれ、本来の宛先に転送される。

例えば、図 6.1 において、ノード R からノード D に至るパス中のリンクに故障が発生した場合、ノード D 宛のパケットは、ノード E を終端とするトンネル

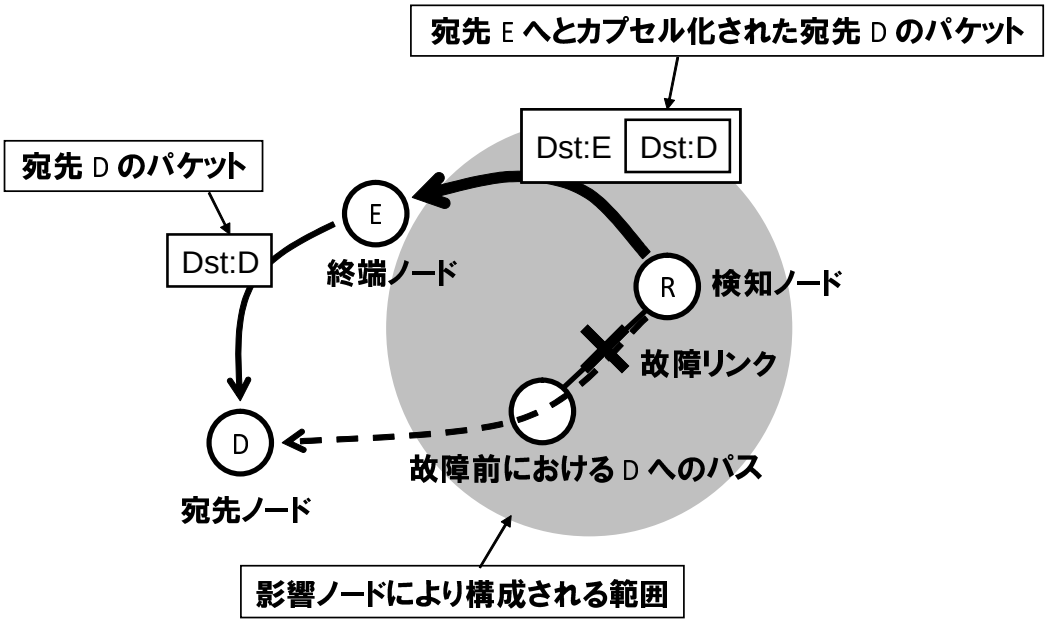

図 6.1: トンネリングを用いた迂回動作

を経由することで、故障リンクを迂回して、宛先まで到達することができる。このように提案手法は、リンク故障への対処は故障を検知したノード R のみが行うため、故障通知を必要とせず、前章の手法と比べてより短い時間でパケット到達性を復旧できる。

本章の構成は次の通りである。6.2 において故障箇所高速迂回手法の提案を行った後、6.3 においてシミュレーションを用いた提案手法の評価結果を示す。さらに 6.4 において、提案手法で復旧できないケースについての議論を行った後、6.5 においてまとめを行う。

6.2 故障箇所高速迂回手法の提案

本節では、故障箇所高速迂回手法の提案を行う。提案手法では、故障を検知したノードが、トンネルにパケットを送ることで、故障リンクを迂回させる (図 6.1)。提案手法では、検知ノードにおいてトンネルを事前に設定しておけば、他ノードの経路表更新は不要となる。このため、従来手法において必要であった経路表更

新のための故障通知は、提案手法では必要ない。このため、故障通知にかかっていた時間を削減でき、故障からの復旧時間を従来よりも短縮できる。

しかし、トンネルを用いた迂回を実現するためには、トンネルから出たパケットはその後ループを起こさず宛先ノードまで到達する必要があり、トンネルの終端ノードを適切に選ぶ必要がある。ここで、故障に伴い経路更新を行わなければループを引き起こすノードを影響ノードと呼ぶ。提案手法では、4 章において示した局所化アルゴリズムを用いて、影響ノードを特定し、影響ノードではないノードをトンネルの終端ノードとして選択する。

図 6.2 に七つのリンクを持つネットワークの例を示す。ネットワーク中のリンクのうち一つ (R–D 間のリンク) に故障が発生したとき、始点がノード S 、宛先がノード D であるパケットは、通常のルーティングでノード R に転送される。しかし、このパケットを転送するために、ノード R は R–D 間のリンクを使用することはできない。もし、ノード R がこのパケットをノード A に転送すると、ノード A が自身の経路表を更新しないことを前提としているため、ノード R と A の間でルーティングループが生じ、パケットはその宛先まで到達することができない。提案手法では、ノード R が、ノード D 宛てのパケットに対しノード B を宛先としたトンネリングパケットにカプセル化を行い、カプセル化されたパケットをノード B に転送する。カプセル化されたパケットは、ノード A の経路表を更新を行わなくても、ノード B に到達できる。ノード B に到達性したパケットは、カプセル化を解かれ、通常のルーティングでノード C に転送された後、その最終宛先であるノード D に到達する。

このように、提案手法では、故障リンクに接続するノードのみが、その故障に対処するための動作を行う。他のノードは普段どおり、通常の経路表を用いたパケット転送を行なっていればよい。

6.2.1 提案手法における迂回手順

ここでは、提案手法における故障発生時のパケット迂回手順を具体的に示す。各ノード n_r は以下の手順により、パケット転送を行う。

1. 通常経路表を参照し、パケットヘッダ中の宛先ノード n_d に対する次転送先

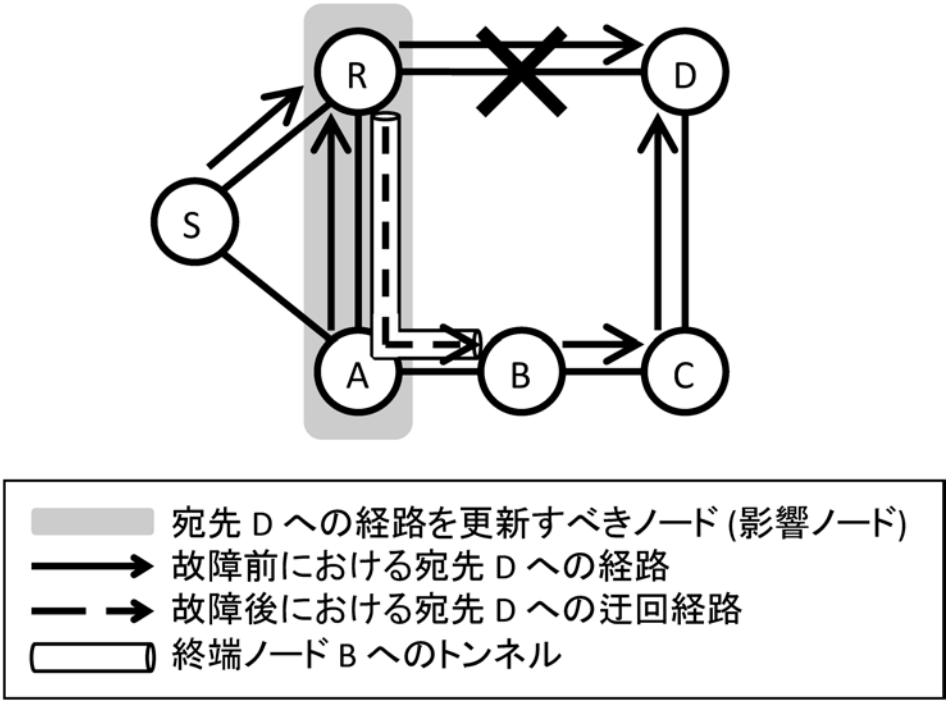

図 6.2: トンネルを用いた高速迂回

n_n を決定する。

2. n_r, n_n 間のリンク e が故障リンクでない場合、パケットを n_n に転送する。

3. n_r, n_n 間のリンク e が故障リンクである場合、後述の手順を用いて終端ノード n_i を決定する。

4. 終端ノード n_i が n_r の隣接ノードである場合、パケットを n_i に転送する。

5. 終端ノード n_i が n_r の隣接ノードでない場合、パケットに対し宛先を n_i とするカプセル化を行う。カプセル化したパケットに対し、本手順を再度適用して、次転送先を決定する。

上記ステップ 1, 2 は通常の IP ルーティングと同様の動作である。リンク故障により次転送先 n_n へのリンクが故障している場合には、提案手法の適用を行う。ステップ 3 で決定した終端ノード n_i が n_r の隣接ノードである場合には、n_r, n_i 間に影響ノードが存在しないため、直接パケットを n_i に転送すればよい。これに対し n_i が n_r の隣接ノードでない場合は、n_r, n_i 間に存在する影響ノードを通過させる必要がある。そのために、カプセル化の宛先を n_i と設定したうえで、ステップ 1 からの手順を再度適用し、カプセル化したパケットの転送を行う。

次に、パケット転送手順中のステップ 3 における終端ノード決定手順を以下に示す。

1. リンク e 故障後のトポロジーにおけるノード n_r を始点とする最短パスツリー $\mathcal{T}_r'$ を計算する。

2. 4 章で示した局所化アルゴリズムを用いて、故障リンク e および宛先ノード n_d の組に対する影響ノードの集合 $\mathcal{V}_u^{(d)}$ を求める。

3. 図 6.3 のアルゴリズムを用いて宛先ノード n_d に対するトンネルの終端ノード n_i を決定する。

図 6.3 中のステップ 3 では、リンク e 故障後のトポロジーにおける最短パスツリー $\mathcal{T}_r'$ に沿って、始点ノード n_r から宛先ノード n_d まで辿る処理である。ステッ

Find end-point node(n_r, $\mathcal{T}_r'$, $\mathcal{V}_u^{(d)}$)

1: $n_i \Leftarrow n_r$
2: **repeat**
3: $n_i \Leftarrow Nexthop(n_i, n_d, \mathcal{T}_r')$
4: **until** $n_i \neq \mathcal{V}_u^{(d)}$
5: n_i is the end-point node for n_d.
6: **return**

図 6.3: 終端ノードの決定アルゴリズム

プ 3 におけるノード n_i が $\mathcal{V}_u^{(d)}$ 中のノードではない、つまり影響ノードではないときに、n_i を宛先 n_d に対する終端ノードとする。

ここでは、終端ノード決定手順を、パケット転送手順中のステップ 3 において、故障発生時に行うよう説明しているが、実際には故障発生前に行っておく。この手順の事前実行により、パケット転送手順はテーブルの検索処理とカプセル化処理のみとなるため、故障からの復旧時間を削減できる。

6.2.2 提案手法の利点および欠点

故障箇所高速迂回手法の利点は、次のとおりである。

- 故障通知を必要としないため、前章の事前計算型経路更新手法と比べ、より故障からの復旧時間を短縮できる。
- 提案手法により用意すべき代替経路表数とそれらを計算するためのリソースはそれぞれ、前章の事前計算型経路更新手法と比べ、少ない。これらの定量的評価については、6.3 で述べる。
- 検知ノードによる復旧動作のみで、パケットの到達性を復旧させることができる。このことにより、前章の手法と比べ、既存ネットワークへの提案手法の配備を段階的に行うことが容易である。

ネットワーク中のノードに提案手法を実現する機能を追加する、もしくはネットワーク中のノードをこの機能を持ったノードと入れ替えることを提案手法の配備と呼ぶ。実際のネットワークにおいて、提案手法による故障発生時の復旧時間短縮を実現するためには、ネットワーク中に提案手法の配備を行う必要がある。前章の手法は、故障発生時に影響ノードが事前に用意した代替経路表に切替を行うため、その効果を得るために、検知ノードと影響ノード双方に配備を行う必要があった。一方、提案手法では検知ノードがトンネルの事前設定を行っておくだけでよい。つまり、あるリンクの故障に対し、そのリンクの両端のノードのみが、故障検知時に即座にパケットをトンネルに迂回させる機能を有していれば、提案手法の効果を得ることができる。このため、提案手法は段階的な配備がより容易であり、故障発生時におけるパケット到達性の改善を段階的に行うことができる。

提案手法の欠点は、トンネルを用いることで全体のパス長の増大を招く可能性がある点である。前章の事前計算型経路更新手法において形成される迂回パスも通常の経路更新と比較し、パス長が長くなる可能性があった。しかし、提案手法による迂回パスはそれよりも長くなる可能性がある。図 6.4 を用いて、ノード R–D 間のリンク故障に対する迂回パスのパス長の比較を行う。通常の経路更新では、ノード S から転送が開始されたノード D 宛のパケットは、ノード B、ノード C によって転送される (図 6.4 (a) 参照)。前章の事前計算型経路更新手法では、このパケットはノード A、ノード B、ノード C の順に転送される (図 6.4 (b) 参照)。これに対し、提案手法においては、ノード A を経由してノード R に到達したパケットは、トンネリングによりノード A およびノード B を経由してノード C に至る。その後、ノード C によりカプセル化を解かれたパケットは、ノード D に転送される (図 6.4 (c) 参照)。通常の経路更新では、故障後のトポロジーに基づいた最短パスを形成するので、故障箇所を迂回するパスはノード S からノード D への最短パスであり、そのパス長は 5 となる。それに対し、事前計算型経路更新手法および故障箇所高速迂回手法における迂回パスはともに最短パスとはならず、パス長はそれぞれ 6 および 8 となる。図 6.4 (c) を参照すると、故障箇所高速迂回手法においては、ノード A–R 間を接続するリンク上を通ってノード R に至るパケットは、カプセル化された後、再度このリンクを通過する。このように故障箇所高速迂回手法では、パケットが同一リンク上を往復する可能性があるため、一般に

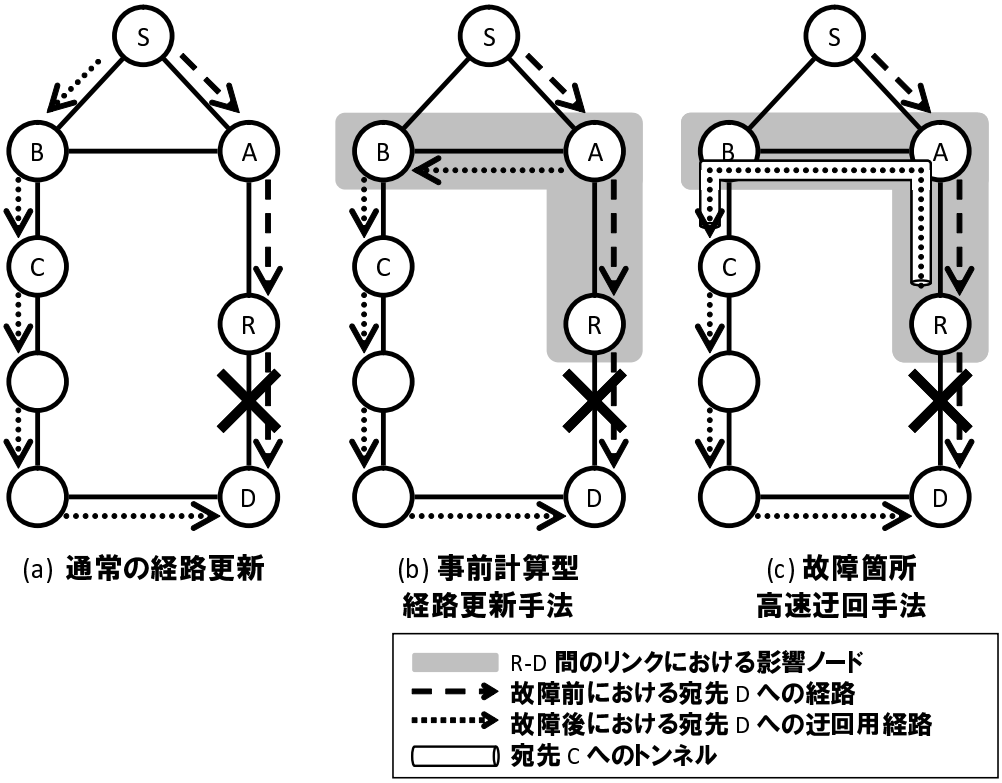

図 6.4: 各手法において生成される迂回パス

事前計算型経路更新手法と比べオーバーヘッドが大きくなる。しかし、実際のネットワークにおいてはこれらのオーバーヘッドが小さいことを、6.3.3 において定量的に示す。

6.3 提案手法の評価

本節では、シミュレーションを用いて、前章の事前計算型経路更新手法と故障箇所高速迂回手法の定量的な比較を行う。初めに 6.3.1 でシミュレーションで用いるトポロジーの説明を行う。次に、両手法により用意すべき代替経路表の数およびその計算リソースについて 6.3.2 において評価を行う。さらに故障箇所を回避する迂回パスのパス長に関して、通常の経路更新と比較した場合の両手法のオーバーヘッドの評価を 6.3.3 で行い、6.3.4 で段階的配備におけるカバー率について評価

する。

6.3.1 評価で用いるネットワークトポロジー

本節では、5.3.1 と同様に BRITE [Med01] を用いて、ノード数や平均次数がことなる各種トポロジーを生成し評価を行う。使用するモデルは、5.3.1 と同様に、Barabasi-Albert (BA) モデル [Bar99] を用いる。

議論を単純化するために、ネットワークトポロジーの生成に関して以下の三つの仮定を置く。

- すべてのリンクはポイントトゥポイント型であるとする。
- すべてのリンクは双方向通信可能であり、そのコストは対称であるとする。
- ネットワーク中のすべてのノードはそれぞれアドレスを一つのみ持ち、リンクはアドレスを持っていないとする。つまり、各ノードが持つ経路表には、宛先としてリンクではなく各ノードが登録される。

これらは、事前計算型経路更新手法における結果との比較を行うために、5.3.1 と同一の仮定となっている。

6.3.2 必要となる代替経路表数とリソース

リンクの単一故障に対して用意すべき代替経路表数を評価するために、リンク数、ノード数の異なる様々なネットワークを用意した。これらの各ネットワーク毎に、そのネットワーク内の各ノードが用意すべき代替経路表数の平均および最大値を求め、それぞれ図 6.5 および図 6.6 にまとめた。これ以降の図中では、事前計算型経路更新手法を “Update” と、故障箇所高速迂回手法を “Reroute” と表記する。

図 6.5 を参照すると、事前計算型経路更新手法において用意すべき代替経路表数は、ノード数および次数の増加に対して、それぞれ増加していることが分かる。これに対し、故障箇所高速迂回手法では、各ノードが用意すべき代替経路表は、自

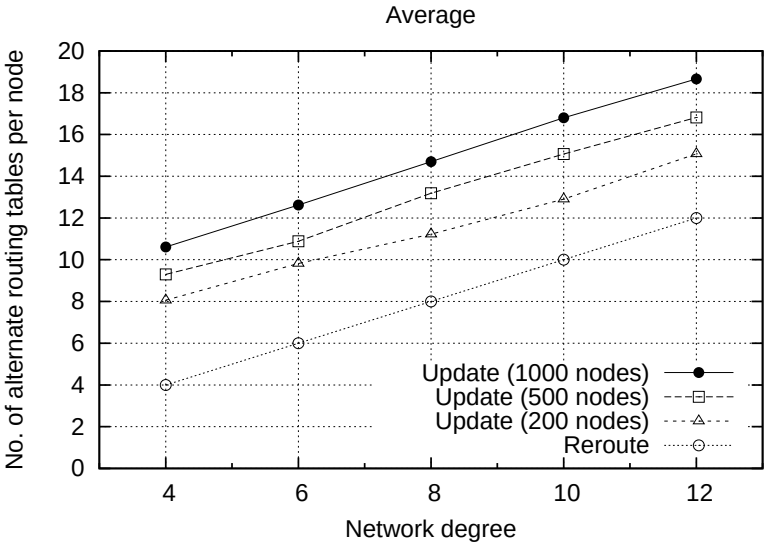

図 6.5: ノード毎に用意すべき代替経路表数 (平均値)

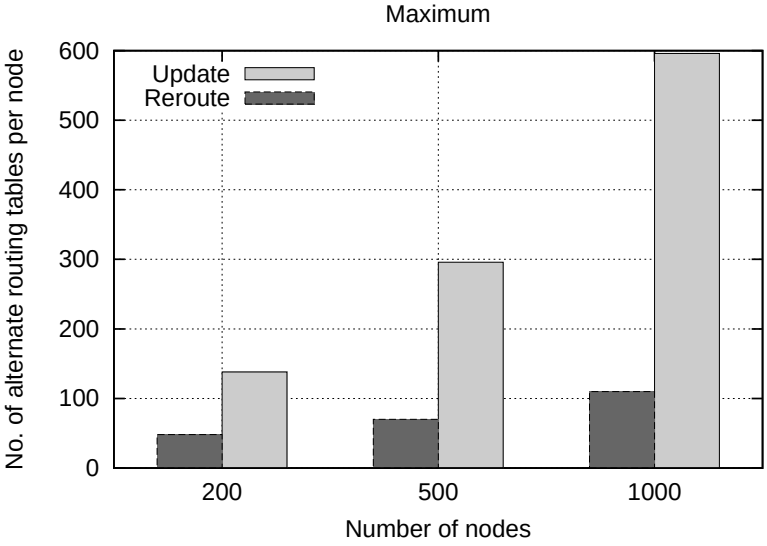

図 6.6: ノード毎に用意すべき代替経路表数 (最大値)

身に接続するリンク毎に用意すればよい。つまり、各ノード毎の代替経路表数の平均は、ネットワーク中のノード数に関わらず、ネットワークの次数と等しくなる。両手法における代替経路表数はともに比較的少ないが、故障箇所高速迂回手法がより少なくて済む。例えば、ネットワークのノード数が 1000 であっても、平均次数 12 のネットワークにおいては各ノードが必要とする代替経路表数は故障箇所高速迂回手法では平均で 12 となる。しかし、事前計算型経路更新手法は同じ条件で平均 19 の代替経路表を必要とする。

両手法における必要となる代替経路表数の平均は、それほど大きな差ではなかった。しかし、両手法における最大値には大きな差が見られる。図 6.6 中の 1000 ノードのネットワークでの結果によると、事前計算型経路更新手法においては、最大で 600 もの代替経路表を必要とするノードが存在する。これに対し、故障箇所高速迂回手法では、同じ条件で最大でも 110 の代替経路表で済むことが分かる。

平均次数 4 のネットワークにおける各ノード毎の次数と代替経路表の数の関係を、図 6.7 から図 6.9 に示す。ノード次数の増加につれて、両手法とも代替経路表の数が増加していることがわかる。実際のネットワークでは、高い次数を持つノードはコアルータに相当し、低い次数を持つノードはエッジルータに相当する。本節の結果によれば、コアルータはより多くの代替経路表を必要とすることとなる。図 6.9 によれば、110 リンクを持つコアルータは、事前計算型経路更新手法および故障箇所高速迂回手法においてそれぞれ 596、110 の代替経路表を必要とする。

代替経路表は通常の経路表との差分だけを記録することで、格納するメモリサイズを削減できる。すべての代替経路表を差分だけ記録することとした場合、通常の経路表のみを記録する場合の何倍のメモリサイズを必要とするかを、シミュレーションにより求めた。表 6.1 を参照すると、事前計算型経路更新手法における代替経路表を格納するためのメモリ量は、通常の経路表を格納する場合の 5.27 倍となっている。これに対し、故障箇所高速迂回手法では、通常の経路表と同じメモリ量で十分である。

例えば、次数が 110 であるノードが、事前計算型経路更新手法および故障箇所高速迂回手法それぞれで格納すべき代替経路表のサイズは、1000 ノードのネットワークにおいて、それぞれ 184 キロバイトおよび 35 キロバイト程度である (通常

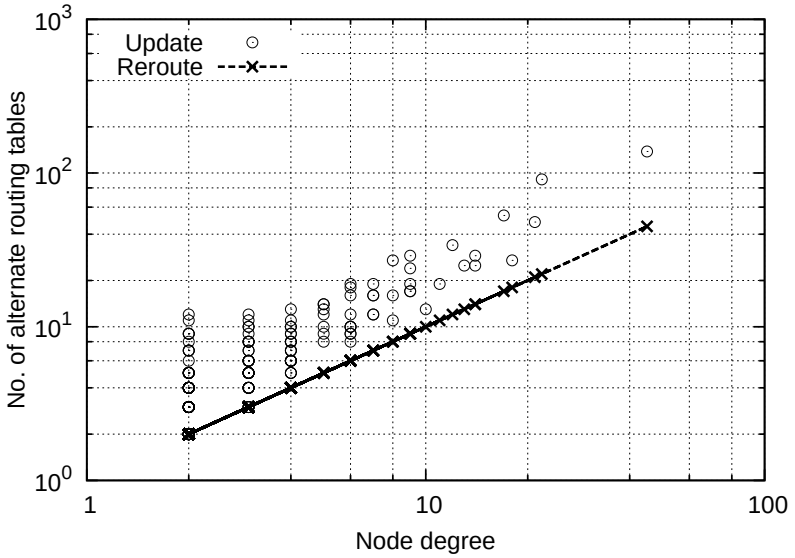

図 6.7: ノード次数と代替経路表数の関係 (200 ノード、397 リンク)

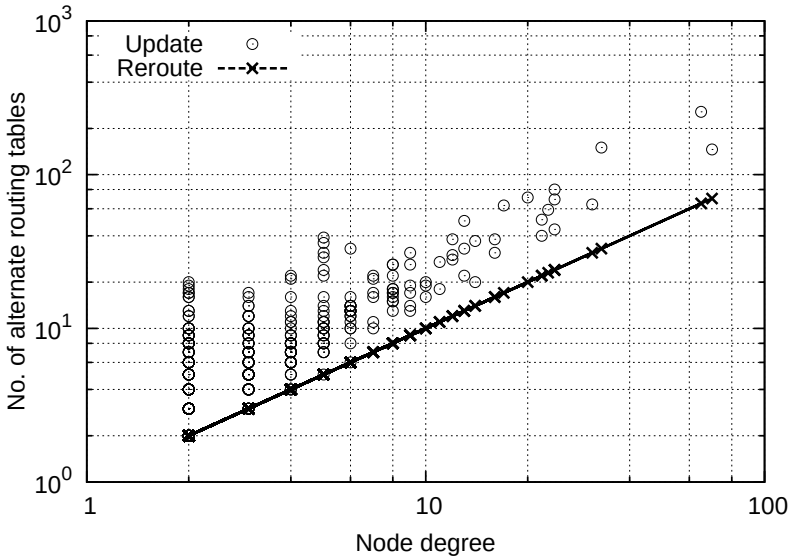

図 6.8: ノード次数と代替経路表数の関係 (500 ノード、997 リンク)

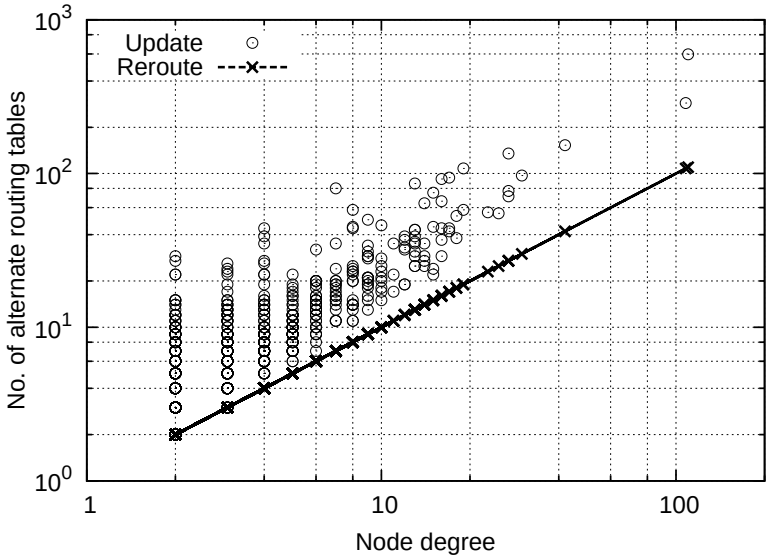

図 6.9: ノード次数と代替経路表数の関係 (1000 ノード、1997 リンク)

の経路表のサイズは 35 キロバイト[1]であり、その 5.27 倍および 1.0 倍)。これらの代替経路表を格納するのに DRAM を使用した場合、実装上の問題となるサイズではない。

両手法における代替経路表の計算コストもまた小さい。各代替経路表を算出するために、最短パスツリーを計算する必要がある。それぞれの最短パスツリー計算

表 6.1: 代替経路を格納するのに必要となるメモリ量 (平均次数 4)

	事前計算型経路更新手法			故障箇所高速迂回手法
ノード数	最小	平均	最大	
200	1.48	2.15	4.53	1.0
500	1.49	2.19	5.09	1.0
1000	1.52	2.31	5.27	1.0

[1]IPv6 の経路表における 1 エントリを、それぞれ 16 バイトの宛先プレフィックスおよび次転送先アドレス、1 バイトのプレフィックス長、2 バイトのインターフェイス識別子から構成されるとした場合

表 6.2: 代替経路表の総計算コスト (平均次数 4)

	事前計算型経路更新手法			故障箇所高速迂回手法		
ノード数	最小	平均	最大	最小	平均	最大
200	0.72	1.16	2.57	0.57	0.97	1.04
500	0.70	1.24	2.34	0.64	1.00	1.05
1000	0.76	1.29	2.92	0.64	1.01	1.06

は、Incremental SPF アルゴリズム [McQ80] を用いることで、計算コストを削減できる。最短パスツリーの計算に用いられるダイクストラ法のコストは、ノード数を N、リンク数 L とした場合、$O((N+L)\log N)$ で表せる。Incremental SPF は、オリジナルの最短パスツリーに対し、差分のみを計算することで、計算コストの削減を実現している。ノード数が 200, 500 および 1000、平均次数が 4 のネットワークに対して、Incremental SPF を用いて両手法における代替経路表を計算した場合の総計算コストについて調査を行った。通常の最短パスツリー計算を基準にした結果を、表 6.2 に示す。

事前計算型経路更新手法と故障箇所高速迂回手法を比較した場合、代替経路表数が少ない故障箇所高速迂回手法の方が、総計算コストも少ない。このため、総計算コストの面でも、故障箇所高速迂回手法は事前計算型経路更新手法よりも優れている。

6.3.3 パス長に関するオーバーヘッド

故障箇所高速迂回手法は、故障箇所を回避するよう構成される迂回パスのパス長が、事前計算型経路更新手法に対して長くなるという弱点を持つ。6.2.2 で議論したように、事前計算型経路更新手法および故障箇所高速迂回手法における迂回パスのパス長それぞれは、通常の経路更新におけるそれよりも長い。ネットワーク中のあるリンクが故障した場合、それぞれの手法で形成される迂回パス長を調べ、通常の経路更新におけるパス長との比較を行った。この比較では、その最短パス中に故障リンクが含まれる二ノードペアのみを対象とした。ノード数がそれ

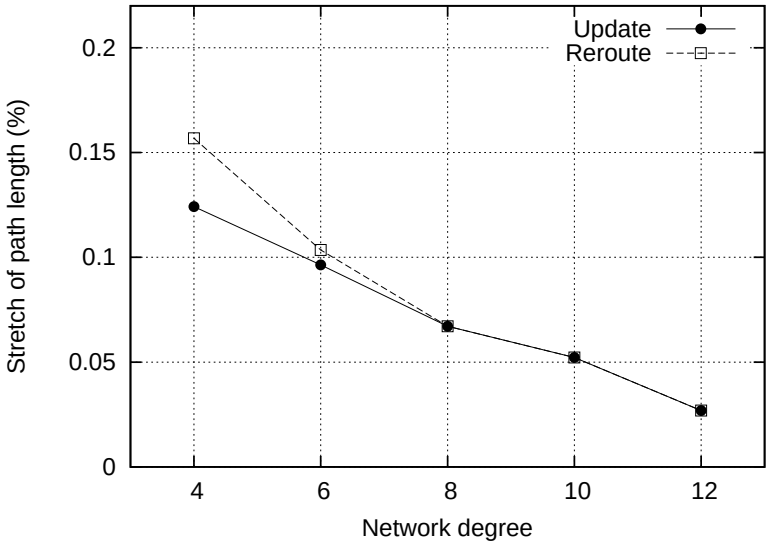

図 6.10: パス長のオーバーヘッド (200 ノード)

ぞれ 200, 500, 1000 であり、平均ノード次数が 4 から 12 までのネットワークそれぞれに対して両手法の迂回パス長の計算を行い、通常の経路更新における迂回パス長と比較した結果を図 6.10–6.12 にそれぞれ示す。

ノードの平均次数が 4 および 6 であるネットワークにおいては、故障箇所高速迂回手法において構成される迂回パスのパス長は、事前計算型経路更新手法におけるそれより長くなっている。これに対し、平均次数が 8 より大きなネットワークにおいては、事前計算型経路更新手法および故障箇所高速迂回手法によって構成される迂回パス長は同一となっている。次数が高くなるにつれて、故障箇所高速迂回手法によって構成される迂回パス中において、同一リンク上 (図 6.4 (c) 中におけるノード A–R 間のリンク) を往復する可能性が低くなるためである。

迂回パス長に関しては両手法間に大きな差がない。また両手法とも通常の経路更新に対してオーバーヘッドが存在するが、その割合は 0.2 % 以下であり、実用上大きな問題にはならないと考えられる。

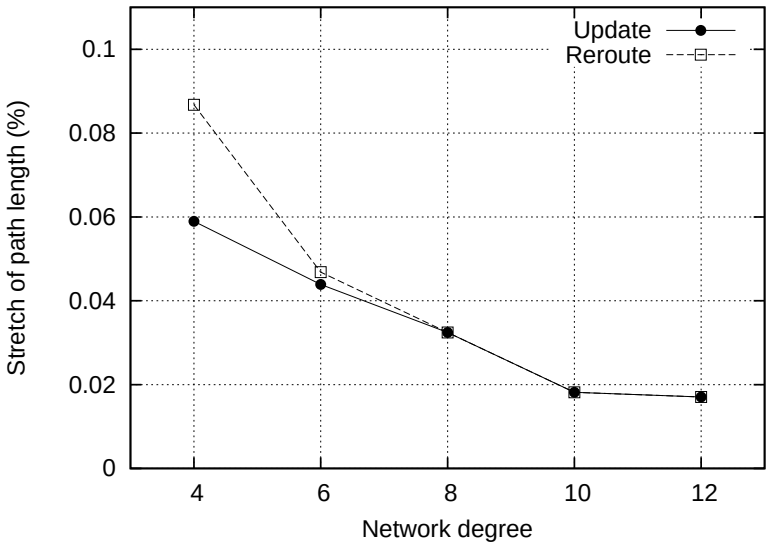

図 6.11: パス長のオーバーヘッド (500 ノード)

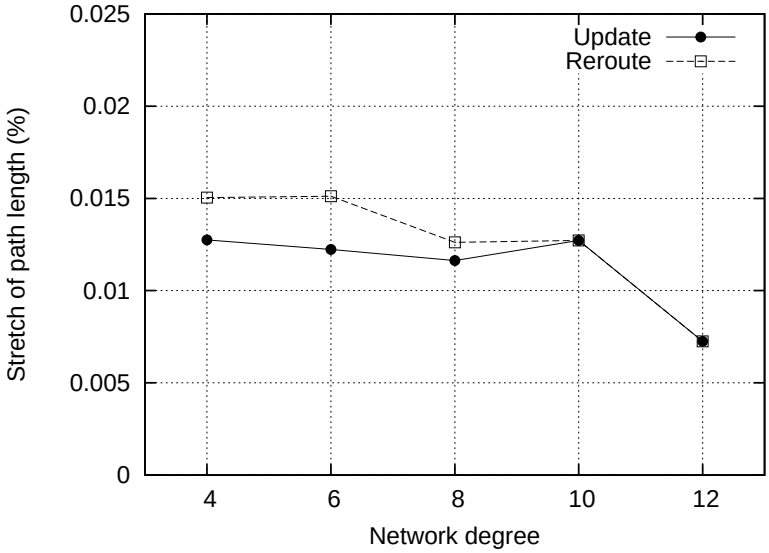

図 6.12: パス長のオーバーヘッド (1000 ノード)

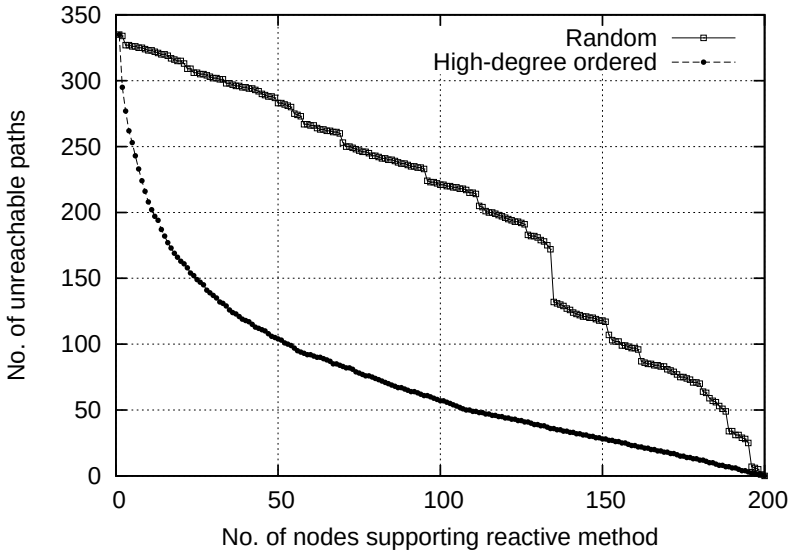

図 6.13: 提案手法を採用したノードの数と、単一故障時に到達不能になるパス数の関係 (200 ノード、397 リンク)

6.3.4 段階的配備時におけるカバー率

故障箇所高速迂回手法の重要な利点のひとつに、事前計算型経路更新手法では不可能であった段階的な配備が可能である点が挙げられる。Barabasi-Albert モデルで 200, 500, 1000 ノードのネットワークに対し、提案手法を適用したノードの数を徐々に増やしていったとき、到達不能パス数がどのように変化するかについて調べた結果を図 6.13–6.15 に示す。各ノードに対する提案手法の適用を、ランダム (Random) に行った場合と次数が大きい順 (High-degree ordered) に行った場合の二通りについて、調査を行った。ノード数が N であるネットワーク中の任意の二ノード間のパスの数は、$N \times (N-1)$ で表される。そのため、200, 500, 1000 ノードのネットワークにおける任意の二ノード間の全パス数は、39800, 249500, 999000 になる。これらのパスのうち、単一リンク故障が発生したとき、提案手法で復旧できず到達不能となるパスの数を、ここでは到達不能パス数とする。図 6.13–6.15 において、縦軸は到達不能パス数を表し、横軸は提案手法を適用したノードの数を表している。

これらの図から、以下のことが分かる。

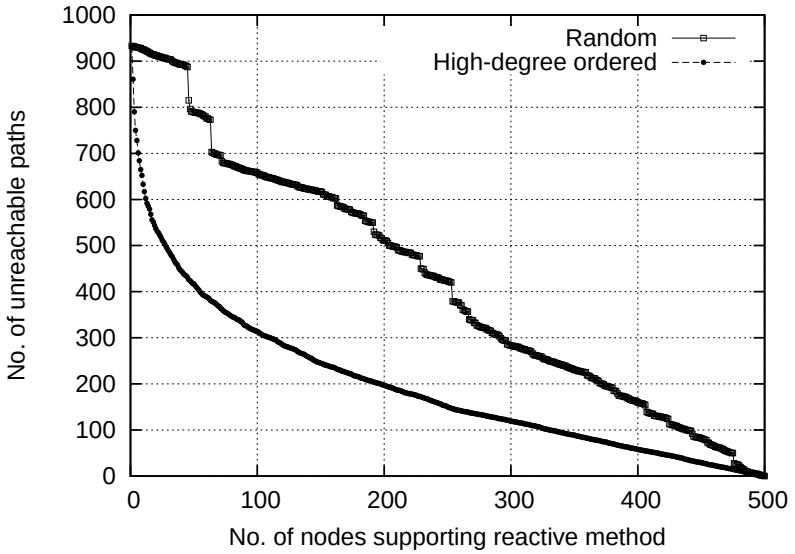

図 6.14: 提案手法を採用したノードの数と、単一故障時に到達不能になるパス数の関係 (500 ノード、997 リンク)

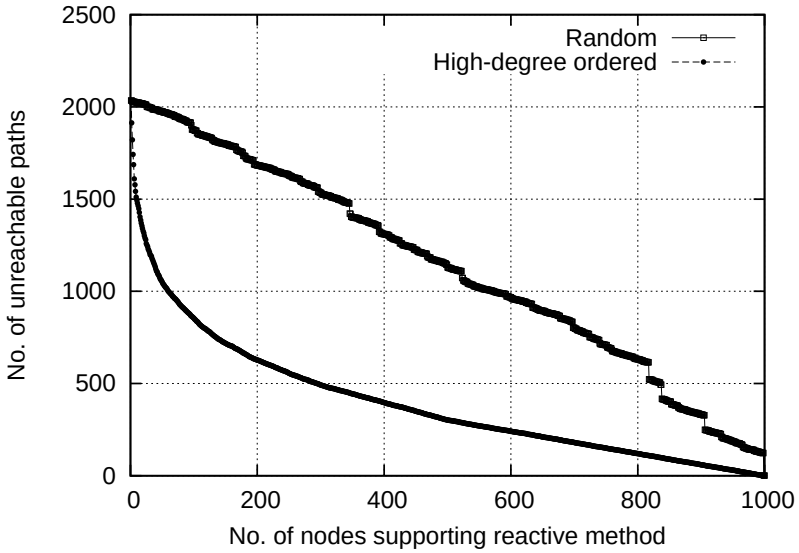

図 6.15: 提案手法を採用したノードの数と、単一故障時に到達不能になるパス数の関係 (1000 ノード、1997 リンク)

- ネットワーク中に提案手法を適用するノードが存在しないとき、200, 500 および 1000 ノードの BA ネットワークにおいては、単一リンク故障に時に、それぞれ平均で 335, 933 および 2035 の到達不能パスが生じる。これに対し、全ノードが提案手法を適用した場合、到達不能パス数は当然 0 となる。

- ノードの次数が大きい順に提案手法を適用したほうが、ランダムに適用した場合と比べて、すみやかに到達不能ノード数を少なくすることができる。つまり、コアルータに相当する次数が大きいノードに対し先に提案手法を適用したほうが、エッジルータに相当する次数の小さいノードに対するより、効果が大きい。

6.4 提案手法で復旧できないケースについての議論

前節では、故障箇所高速迂回手法の評価のために、BA モデルにおいて生成したネットワークトポロジーを用いた。6.3.4 では、これらのトポロジーに対して全ノードに提案手法を適用することで、すべてのケースに対応可能であることを示した。しかし、図 6.16 に示すケースでは、提案手法を適用しても復旧できない。ノード A–D 間のリンクのコストは他と比べて非常に高いので、ノード D 宛てのパケットは、ノード A における経路表の有無にかかわらず、このリンクを使っての転送は行われない。このケースにおいてノード A は影響ノードであり、ノード D がトンネルの端点となるノードである。カプセル化がなされたとしても、カプセル化されたパケットがトンネル端点のノードにも到達できないため、提案手法ではこのパケットを宛先まで届けることができない。

前節の評価では、すべてのリンクは同じコストを持つネットワークを使用していたため、図 6.16 のケースは生じていなかった。しかし、現実のネットワークには異なる帯域のリンクが混在しており、広帯域を持つリンクには、より多くのトラフィックを流すために、低いコスト値を割り当てるのが一般的である。反対に、ノード A–D 間のリンクのように高いコストを持つリンクは、その帯域が狭いことが多い。一般に帯域の狭いリンクは必要性の高い通信に限定して使用されることが通常であり、提案手法がターゲットとする無条件での迂回に使用しないのが普通である。それゆえ、これは提案手法にとって大きな問題とはならない。

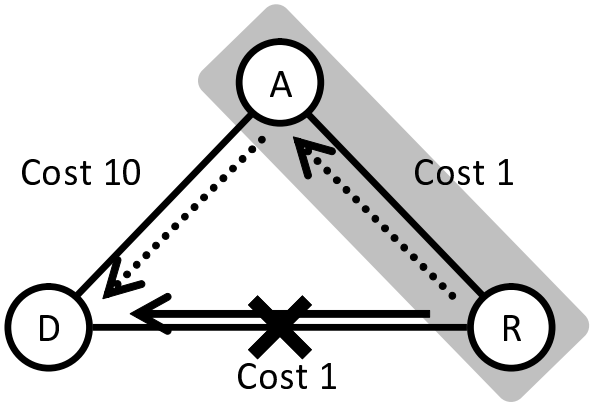

図 6.16: 不均一なコストを持つネットワーク

6.5 結言

本章では、経路切り替えの際に故障通知を不要とする故障箇所高速迂回手法の提案を行い、前章で提案した事前計算型経路更新手法との比較を行った。

本章で提案した故障箇所高速迂回手法では、故障を検知したノードがトンネルを用いて、パケットを迂回させる。トンネルから出たパケットはその後宛先までループを起こさずに宛先ノードまで到達する必要があり、そのためトンネルの終端ノードを適切に選ばなければならない。提案手法では、4 章において示した局所化アルゴリズムを用いることで適切な終端ノードを決定する方法を示した。提案手法では、故障に対する対処は検知ノードのみが行えばよく、他のノードはトンネル経由で送られてきたパケットのカプセル化を解除するだけでよい。このため、提案手法には以下のような利点がある。

- 故障箇所高速迂回手法は、事前計算型経路更新手法と比較して、より少ないリソース (代替経路表を格納するのに必要なメモリ量およびその計算コスト) で実装を行うことが可能である。
- 検知ノード以外のノードは自身の経路表を更新する必要がないため、事前計算型経路更新手法と比較して、より短い時間でパケット到達性を復旧することができる。

- 一般的な IP ルータの大半はトンネルを構成する機能を有するので、提案手法の配備は段階的に行うことが可能である。このため、事前計算型経路更新手法よりも既存のネットワークへの配備は容易である。

第7章　結論

IP ネットワークは、今後様々な分野への適用が見込まれ、その利用が拡大すると考えられている。本論文では、その可用性を向上させるための手法を取り上げている。

第 2 章では、ネットワークの可用性向上技術に関する従来技術について述べ、本研究の位置づけを明らかにした。可用性向上のためには、平均故障間隔の向上および平均復旧時間の短縮のいずれかが必要である。前者を実現するために IP 層におけるノード装置であるルータの高信頼化技術が各種提案されているが、これらの技術のみでネットワークの可用性を向上させることには限界がある。そのため、故障発生時に経路の切替を行う経路制御技術と組み合わせて使用することが一般的である。経路制御における故障発生時の復旧動作は、以下の各フェーズから構成される。

- 故障の検知
- 故障の通知
- 経路表の再構築

このうち、故障検知フェーズに関しては、数十ミリ秒で故障を検知する技術が確立されている。故障の通知と経路表の再構築にかかる時間を短縮するために、高速迂回手法と呼ばれる技術が各種提案されている。既存の IP ネットワークに広く適用可能であるためには、転送方式の変更を必要としないことおよび分散処理が可能であることという二つの条件が必要であるが、これらを共に満たす従来方式は存在しない。そのため、本研究ではこれらの条件を満たす高速迂回方式を実現し、故障通知と経路表再構築にかかる時間の短縮を目標とした。

第 3 章では、転送方式の変更を必要としない高速迂回手法を実現するために、故障発生時における経路更新の有無がパケット到達性に与える影響を調べた。ま

ず故障によるトポロジー変化に起因して到達不能ノードが生じる条件として以下の四つを示した。

- 宛先ノード自体の故障
- 宛先ノードまでのパスの喪失
- 次転送先ノードの故障
- ルーティングループの発生

これらの条件のいずれかに当てはまることが、到達不能ノードの発生の必要十分条件であることを示し、パケット転送動作復旧のためには、それぞれの条件に対して、どのように対処すべきかの考察を行った。

上記条件のうち、次転送先ノードの故障とルーティングループ発生は、各ノードが適切に経路を更新することでパケット到達性を復旧させられるので、経路制御において対処を行う方法が有効である。隣接ノードである次転送先の故障は、高速検知技術を行いて故障を検知後、即座に対処を行えばよい。一方、ルーティングループ発生は、故障箇所に隣接していない場合に到達不能ノードが生じるケースである。このケースは、故障箇所からの距離が遠いほど発生する可能性が低いことをシミュレーションを用いて示した。

第 4 章では、先の解析結果を元に経路更新箇所の局所化アルゴリズムを提案した。故障発生時におけるパケット到達性を高速に復旧させるためには、経路更新を必要とするノードを特定することが重要である。これらのノードを特定するための条件が、以下の二つであることを明らかにした。

- 経路更新を必要とするノードは、故障前のトポロジーにおいて、自身から宛先ノードへの最短パス中に故障リンクを含む。
- 経路更新を必要とするノードは、故障後のトポロジーにおいて、故障リンクに接続しているノードから宛先ノードへの最短パス上に存在する。

さらにこれらの条件を共に満たすノードのみが経路更新を行うことにより、パケット到達性を復旧できることを証明した。次に、これらの条件の判定処理を、ネットワーク中の各ノードにより分散判定を行う方法を示した。この分散判定は、一箇

所で集中して判定を行う一括処理と比較し、計算量の面でオーバーヘッドが存在しない。また、この分散判定を行うための手順として局所化アルゴリズムを示した。このアルゴリズムを用いることで、高速迂回を実現する際に必要となる、故障発生時に経路更新を必要とするノードの特定が可能である。

第 5 章では、事前計算型経路更新手法を提案した。この手法では、故障発生時に使用する代替経路表を事前に計算しておくことにより、復旧時間の短縮を実現できる。しかし、代替経路表は故障箇所毎に用意する必要があるため、大規模なネットワークでは非常に多くの代替経路表を事前に計算する必要があった。この手法では局所化アルゴリズムを用いることで、代替経路表の数が必要最小限になるよう絞り込みを行う。このことにより、必要とする代替経路表数を従来と比べ、平均で 1/100 に削減できることを示した。また、代替経路表の計算コストと格納メモリ量は、実際の IP ルータに実装を行う際に問題とならない程度に小さいことを示した。

第 6 章では、故障箇所高速迂回手法を提案した。この手法では、故障を検知したノードが、故障の影響を受けないノードまでトンネリングを用いて、パケットを迂回させる。故障の影響を受けないノードの特定に、局所化アルゴリズムを用いる。この手法では、検知ノードを除く各ノードは自身の経路表を更新する必要がないため、検知ノードからの故障通知を必要としない。そのため、この手法は従来故障通知にかかっていた時間の短縮を実現している。また、この手法におけるパケットの迂回動作は、故障検知したノードによるトンネルへの迂回処理と、終端ノードにおけるトンネルパケットのカプセル化解除処理のみから構成される。後者のトンネル終端処理を行う機能は、一般的な IP ルータ装置に備わっているため、この手法の配備には、前者のトンネルへの迂回機能の導入さえ行えば良い。それゆえ、この手法は事前計算型経路更新手法よりも、既存のネットワークへの導入は容易である。また、代替経路表を格納するために必要なメモリ量およびその計算コストを、事前計算型経路更新手法と比較して、より少なくできることを示した。

全体として、故障発生時における復旧時間短縮のために課題となっていた、経路表更新時間の削減を事前計算型経路更新手法で、また故障通知時間の削減を故障箇所迂回手法で、それぞれ実現した。これらの成果を用いた故障発生時における復旧時間短縮により、IP ネットワークの可用性を従来よりも飛躍的に向上させ

ることが可能であると考えている。今後 IP ネットワークの適用範囲が従来以上にさまざまな分野へ広がり、多くの人がいつでもその恩恵を受けられるようになれば幸いである。

謝辞

本研究を遂行し、学位論文としてまとめるにあたり、多くの方々にお世話になりました。ここに深く感謝の意を表します。

筑波大学大学院ビジネス科学研究科吉田健一教授には、研究活動全般に渡り、様々なご支援およびご指導を頂きました。研究の進め方、論文のまとめ方、研究発表でのアピールの仕方などご指導いただいたことはいずれも、本論文をまとめる上で欠かすことができないものでした。ここに深く感謝します。

筑波大学大学院ビジネス科学研究科久野靖教授、大木敦雄准教授、倉橋節也准教授には、本研究を進める上で貴重なご助言を頂きました。ここに深く感謝します。発表会の場などで、様々なコメント、アドバイスを下さった筑波大学ビジネス科学研究科教員の方々に感謝します。また、本研究を進める上で様々なご支援を頂いた吉田研究室の皆様にも深くお礼を申し上げます。

元日本電気株式会社システムプラットフォーム研究所主任研究員地引昌弘博士 (現情報通信研究機構研究員) には、本学への入学のきっかけを頂き、また研究と仕事の両面からサポートいただいたことに深く感謝いたします。仕事と通学の両面からサポートしていただいた日本電気株式会社システムプラットフォーム研究所部長塩尻浩久氏 (当時)、矢野由紀子氏には心よりお礼を申し上げます。

最後に、これまで私を暖かく応援していただいた両親に心から感謝します。

参考文献

[Bar98] M. Barbehenn: A Note on the Complexity of Dijkstra's Algorithm for Graphs with Weighted Vertices, *IEEE Transactions on Computers*, Vol. 47, No. 2, p. 263, 1998.

[Bar99] A. L. Barabasi and R. Albert: Emergence of Scaling in Random Networks, *Science*, Vol. 286, pp. 509–512, 1999.

[Bar08] F. Barreto, E. Wille, and L. Nacamura: A Fast Rerouting Approach to Reduce Packet Loss during IP Routing Protocols Convergence, in *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications (AINA)*, pp. 675–682, 2008.

[Bas01] A. Basu and J. Riecke: Stability issues in OSPF routing, *ACM SIGCOMM Computer Communication Review*, Vol. 31, No. 4, pp. 225–236, 2001.

[Bon07] O. Bonaventure, C. Filsfils, and P. Francois: Achieving sub-50 milliseconds recovery upon BGP peering link failures, *IEEE/ACM Transactions on Networking*, Vol. 15, No. 5, pp. 1123–1135, 2007.

[Bou02] C. Boutremans, G. Iannaccone, and C. Diot: Impact of link failures on VoIP performance, in *Proceedings of the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 63–71, 2002.

[Bu02] T. Bu and D. Towsley: On distinguishing between Internet power law topology generators, in *Proceedings of the IEEE INFOCOM 2002*, Vol. 2, pp. 638–647, 2002.

[Chi93] B. Chinoy: Dynamics of internet routing information, *ACM SIGCOMM Computer Communication Review*, Vol. 23, No. 4, pp. 45–52, 1993.

[Chi00] T. Chiueh and P. Pradhan: Suez: A cluster-based scalable real-time packet router, in *Proceedings of the IEEE International Conference on Distributed Computing Systems*, p. 136, 2000.

[Cic08a] T. Cicic: On basic properties of fault-tolerant multi-topology routing, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol. 52, No. 18, pp. 3325–3341, 2008.

[Cic08b] T. Cicic, A. Hansen, A. Kvalbein, M. Hartman, R. Martin, and M. Menth: Relaxed multiple routing configurations for IP fast reroute, in *Proceedings of the IEEE Network Operations and Management Symposium (NOMS)*, p. 457, 2008.

[Cic09] T. Cicic, A. Hansen, A. Kvalbein, M. Hartmann, R. Martin, M. Menth, S. Gjessing, and O. Lysne: Relaxed multiple routing configurations: IP fast reroute for single and correlated failures, *IEEE Transactions on Network and Service Management*, Vol. 6, No. 1, pp. 1–14, 2009.

[Cro00] P. Crowley, M. E. Fluczynski, J.-L. Baer, and B. N. Bershad: Characterizing processor architectures for programmable network interfaces, in *Proceedings of the 14th international conference on Supercomputing*, pp. 54–65, 2000.

[Dav06] M. David: Generating Synthetic VoIP Traffic for Analyzing Redundant OpenBSD-Firewalls, 2006.

[Dij59] E. W. Dijkstra: A note on two problems in connexion with graphs, *Numerische mathematik*, pp. 269–271, 1959.

[Dix06] A. Dixit: Networking Applications for Xtensa Configurable Processors, *Linley Tech 2006*, 2006.

[Doa93] M. Doar and I. Leslie: How Bad is Naive Multicast Routing?, in *Proceedings of IEEE INFOCOM 1993*, Vol. 93, pp. 82–89, 1993.

[Eny09] G. Enyedi, P. Szilágyi, G. Rétvári, and A. Császár: IP fast reroute: Lightweight not-via without additional addresses, in *Proceedings of the IEEE INFOCOM 2009*, pp. 2771–2775, 2009.

[Fal99] M. Faloutsos, P. Faloutsos, and C. Faloutsos: On power-law relationships of the Internet topology, in *Proceedings of the ACM SIGCOMM 1999*, pp. 251–262, 1999.

[Fra05a] P. Francois and O. Bonaventure: An evaluation of IP-based fast reroute techniques, in *Proceedings of the ACM conference on Emerging network experiment and technology (CoNEXT)*, pp. 244–245, 2005.

[Fra05b] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure: Achieving subsecond IGP convergence in large IP networks, *ACM SIGCOMM Computer Communication Review*, Vol. 35, No. 3, pp. 35–44, 2005.

[Fra07] P. Francois, M. Shand, and O. Bonaventure: Disruption free topology reconfiguration in OSPF networks, in *Proceedings of the IEEE INFOCOM 2007*, pp. 89–97, 2007.

[Fum00] A. Fumagalli and L. Valcarenghi: IP restoration vs. WDM protection: Is there an optimal choice?, *IEEE Network*, Vol. 14, No. 6, pp. 34–41, 2000.

[Gar97] J. J. Garcia-Luna-Aceves and S. Murthy: A Path-Finding Algorithm for Loop-Free Routing, *IEEE/ACM Transactions on Networks*, Vol. 5, No. 1, pp. 148–160, 1997.

[GEANT2] GÉANT2 Website, http://www.geant2.net/.

[Gjo07] M. Gjoka, V. Ram, and X. Yang: Evaluation of IP Fast Reroute Proposals, in *Proceedings of the IEEE Communication Systems Software and Middleware (Comsware)*, 2007.

[Gor00] W. Goralski: *SONET*, McGraw-Hill Companies, 2000.

[Han06] A. Hansen, T. Cicic, and S. Gjessing: Alternative schemes for proactive IP recovery, in *Proceedings of the 2nd Conference on Next Generation Internet Design and Engineering (NGI)*, p. 8, 2006.

[Han07] A. Hansen, O. Lysne, T. Cicic, and S. Gjessing: Fast proactive recovery from concurrent failures, in *Proceedings of the IEEE ICC 2007*, pp. 115–122, 2007.

[Hay08] A. Hay, P. Giannoulis, and K. Hay: *Nokia firewall, VPN, and IPSO configuration guide*, Syngress Publishing, 2008.

[Hen02] U. Hengartner, S. Moon, R. Mortier, and C. Diot: Detection and analysis of routing loops in packet traces, in *Proceedings of the ACM SIGCOMM Workshop on Internet Measurment*, pp. 107–112, 2002.

[Hui95] C. Huitema: *Routing in the Internet*, Prentice Hall, 1995.

[Hun02] C. Hunt: *TCP/IP Network Administration*, O'Reilly Media, 2002.

[Ian02] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot: Analysis of link failures in an IP backbone, in *Proceedings of the ACM SIGCOMM Workshop on Internet measurement*, pp. 237–242, 2002.

[Ian04] G. Iannaccone, C. Chuah, S. Bhattacharyya, and C. Diot: Feasibility of IP restoration in a Tier 1 Backbone, *IEEE Network*, Vol. 18, No. 2, pp. 13–19, 2004.

[Internet2] The Internet2 Network, http://www.internet2.edu/.

[ITU06] ITU-T: Network Performance Objectives for IP-Based Services, *Rec. Y.1541, International Telecommunication Union*, 2006.

[Kam09] S. Kamamura, T. Miyamura, C. Pelsser, I. Inoue, and K. Shiomoto: Scalable backup configurations creation for IP fast reroute, in *Proceedings of*

the 7th International Workshop on Design of Reliable Communication Networks(DRCN), pp. 312–318, 2009.

[Kam10a] S. Kamamura, T. Miyamura, C. Pelsser, I. Inoue, and K. Shiomoto: Minimum backup configuration-creation method for IP fast reroute, in *Proceedings of the IEEE Globecom 2009*, pp. 1–6, 2010.

[Kam10b] S. Kamamura, T. Miyamura, Y. Uematsu, and K. Shiomoto: Scalable Backup Configurations Creation for IP Fast Reroute, *IEICE Transactions on Communications*, Vol. E94-B, No. 1, pp. 109–117, 2010.

[Kva06] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne: Fast IP Network Recovery Using Multiple Routing Configurations, in *Proceedings of the IEEE INFOCOM 2006*, pp. 1–11, 2006.

[Kva07] A. Kvalbein, T. Cicic, and S. Gjessing: Post-failure routing performance with multiple routing configurations, in *Proceedings of the IEEE INFOCOM 2007*, pp. 98–106, 2007.

[Kva09] A. Kvalbein, A. Hansen, T. Cicic, S. Gjessing, and O. Lysne: Multiple Routing Configurations for Fast IP Network Recovery, *IEEE/ACM Transactions on Networking*, Vol. 17, No. 2, pp. 473–486, 2009.

[Lab98] C. Labovitz, G. Malan, and F. Jahanian: Internet routing instability, *IEEE/ACM Transactions on Networking*, Vol. 6, No. 5, pp. 515–528, 1998.

[Lal84] P. Lala: *Fault tolerant and fault testable hardware design*, Prentice-Hall, Inc., 1984, (当麻 喜弘, 玉本 英夫, 古屋 清 訳 : フォールト・トレランス入門, オーム社, 1988).

[Lee04] S. Lee, Y. Yu, S. Nelakuditi, Z. Zhang, and C. Chuah: Proactive vs reactive approaches to failure resilient routing, in *Proceedings of the IEEE INFOCOM 2004*, pp. 176–186, 2004.

[Li 09] A. Li, X. Yang, and D. Wetherall: SafeGuard: Safe Forwarding during Route Changes, in *Proceedings of the ACM conference on Emerging network experiment and technology (CoNEXT)*, pp. 301–312, 2009.

[Li04] L. Li, D. Alderson, W. Willinger, and J. Doyle: A first-principles approach to understanding the internet's router-level topology, in *Proceedings of the ACM SIGCOMM 2004*, pp. 3–14, 2004.

[Li07] A. Li, P. Francois, and X. Yang: On improving the efficiency and manageability of NotVia, in *Proceedings of the ACM conference on Emerging network experiment and technology (CoNEXT)*, 2007.

[Man04] J. Manchester, D. Saha, and S. Tripathi: Protection, restoration, and disaster recovery, *IEEE Network*, Vol. 18, No. 2, pp. 3–4, 2004.

[Mao02] Z. Mao, R. Govindan, G. Varghese, and R. Katz: Route flap damping exacerbates internet routing convergence, in *Proceedings of the ACM SIGCOMM 2002*, Vol. 32, pp. 221–233, 2002.

[Mar04] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot: Characterization of failures in an IP backbone, in *Proceedings of the IEEE INFOCOM 2004*, Vol. 4, pp. 2307–2317, 2004.

[McQ80] J. M. McQuillan, I. Richer, and E. C. Rosen: The New Routing Algorithm for the ARPANET, *IEEE Transactions on Communications*, Vol. 28, No. 5, pp. 711–719, 1980.

[Med01] A. Medina, A. Lakhina, I. Matta, and J. Byers: BRITE: An approach to universal topology generation, in *Proceedings of Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp. 346–353, 2001.

[Nel07] S. Nelakuditi, S. Lee, Y. Yu, Z. Zhang, and C. Chuah: Fast Local Rerouting for Handling Transient Link Failures, *IEEE/ACM Transaction on Networking*, Vol. 15, No. 2, pp. 359–372, 2007.

[Pax06] V. Paxson: End-to-end routing behavior in the Internet, *ACM SIGCOMM Computer Communication Review*, Vol. 36, No. 5, p. 56, 2006.

[Pei03] D. Pei, L. Wang, D. Massey, S. Wu, and L. Zhang: A Study of Packet Delivery Performance during Routing Convergence, in *Proceedings of IEEE International Conference on Dependable Systems and Networks (DSN)*, pp. 183–192, 2003.

[Per00] R. Perlman: *Interconnections, 2nd edition*, 2000.

[Pra02] P. Pradhan and T. Chiueh: Implementation and evaluation of a QoS-capable cluster-based IP router, Proceedings of the IEEE/ACM SC2002 Conference (SC '02), 2002.

[RFC1142] D. Oran: OSI IS-IS Intra-domain Routing Protocol, *RFC 1142, Internet Engineering Task Force*, 1990.

[RFC1195] R. Callon: Use of OSI IS-IS for Routing in TCP/IP and Dual Environments, *RFC 1195, Internet Engineering Task Force*, 1990.

[RFC1853] W. Simpson: IP in IP Tunneling, *RFC 1853, Internet Engineering Task Force*, 1995.

[RFC2080] G. Malkin and R. Mannear: RIPng for IPv6, *RFC 2080, Internet Engineering Task Force*, 1997.

[RFC2281] T. Li, B. Cole, P. Morton, and D. Li: Cisco Hot Standby Router Protocol (HSRP), *RFC 2281, Internet Engineering Task Force*, 1998.

[RFC2328] J. Moy: OSPF Version 2, *RFC 2328, Internet Engineering Task Force*, 1998.

[RFC2338] S. Knight, D. Weaver, D. Whipple, R. Hinden, D. Mitzel, P. Hunt, P. Higginson, M. Shand, and A. Lindem: Virtual router redundancy protocol, *RFC 2338, Internet Engineering Task Force*, 1998.

[RFC2439] C. Villamizar, R. Chandra, and R. Govindan: BGP Route Flap Damping, *RFC 2439, Internet Engineering Task Force*, 1998.

[RFC2453] G. Malkin: RIP Version 2, *RFC 2443, Internet Engineering Task Force*, 1998.

[RFC2740] R. Coltun, D. Ferguson, and J. Moy: OSPF for IPv6, *RFC 2740, Internet Engineering Task Force*, 1999.

[RFC2784] D. Farinacci, S. Hanks, D. Meyer, and P. Traina: Generic Routing Encapsulation (GRE), *RFC 2784, Internet Engineering Task Force*, 2000.

[RFC2991] D. Thaler and C. Hopps: Multipath issues in unicast and multicast, *RFC 2991, Internet Engineering Task Force*, 2000.

[RFC3031] E. Rosen, A. Viswanathan, and R. Callon: Multiprotocol Label Switching Architecture, *RFC3031, Internet Engineering Task Force*, 2001.

[RFC3623] J. Moy, P. Pillay-Esnault, and A. Lindem: Graceful OSPF restart, *RFC3623, Internet Engineering Task Force*, 2003.

[RFC3746] L. Yang, R. Anderson, and R. Gopal: Forwarding and Control Element Separation Framework, *RFC 3746, Internet Engineering Task Force*, 2004.

[RFC3768] R. Hinden: Virtual Router Redundancy Protocol (VRRP), *RFC 3768, Internet Engineering Task Force*, 2004.

[RFC4090] P. Pan, G. Swallow, and A. Atlas: Fast Reroute Extensions to RSVP-TE for LSP Tunnels, *RFC 4090, Internet Engineering Task Force*, 2005.

[RFC4271] Y. Rekhter, T. Li, and S. Hares: A Border Gateway Protocol 4 (BGP-4), *RFC 4271, Internet Engineering Task Force*, 2006.

[RFC4724] S. Sangli, E. Chen, R. Fernando, J. Scudder, and Y. Rekhter: Graceful Restart Mechanism for BGP, *RFC 4724, Internet Engineering Task Force*, 2007.

[RFC5286] A. Atlas and A. Zinin: Basic Specification for IP Fast Reroute: Loop-Free Alternates, *RFC 5286, Internet Engineering Task Force*, 2008.

[RFC5306] M. Shand and L. Ginsberg: Restart Signaling for IS-IS, *RFC 5306, Internet Engineering Task Force*, 2008.

[RFC5714] M. Shand and S. Bryand: IP Fast Reroute Framework, *RFC 5714, Internet Engineering Task Force*, 2010.

[RFC5715] M. Shand and S. Bryand: A Framework for Loop-Free Convergence, *RFC 5715, Internet Engineering Task Force*, 2010.

[RFC5798] S. Nadas: Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6, *RFC 5798, Internet Engineering Task Force*, 2010.

[RFC5880] D. Katz and D. Ward: Bidirectional Forwarding Detection, *RFC 5880, Internet Engineering Task Force*, 2010.

[Sha10] M. Shand, S. Bryant, and S. Previdi: IP Fast Reroute Using Not-via Addresses, *draft-ietf-rtgwg-ipfrr-notvia-addresses-05, Internet Engineering Task Force*, 2010.

[SINET3] SINET3 (Science Information NETwork 3), http://www.sinet.ad.jp/.

[Ste94] R. Stevens: *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley Professional, 1994.

[Suz07] K. Suzuki and M. Jibiki: Formalization and Analysis of Routing Loops by Inconsistencies in IP Forwarding Tables, *IEICE Transactions on Communications*, Vol. E90-B, No. 10, pp. 2755–2763, 2007.

[Suz10a] K. Suzuki, M. Jibiki, and K. Yoshida: Comparison of Proactive and Reactive Methods for IP Fast Restoration using Localization Algorithm, in *Proceedings of 4th International Conference on Signal Processing and Communication Systems (ICSPCS)*, IEEE, 2010.

[Suz10b] K. Suzuki, M. Jibiki, and K. Yoshida: Selective Precomputation of Alternate Routes using Link-State Information for IP Fast Restoration, *IEICE Transactions on Communications*, Vol. E93-B, No. 5, pp. 1085–1094, 2010.

[Tam10] A. Tam, K. Xi, and H. Chao: A fast reroute scheme for IP multicast, in *Proceedings of the IEEE Globecom 2009*, pp. 1–7, 2010.

[Tan02] A. Tanenbaum: *Computer Networks, Fourth Edition*, Prentice Hall, 2002.

[Tos04] E. Tosaya, S. Ouimet, R. Martel, R. Lord, P. Inc, and C. Milpitas: Router flip chip packaging solution and reliability, in *Proceeding of the Electronic Components and Technology Conference 2004*, Vol. 1, pp. 1153–1160, 2004.

[Vas04] J. Vasseur, M. Pickavet, and P. Demeester: *Network recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*, Morgan Kaufmann, 2004.

[Vup97] V. Vuppala and L. Ni: Design of A Scalable IP Router, in *Proceedings of the IEEE Hot Interconnects*, 1997.

[Wan07] J. Wang and S. Nelakuditi: IP fast reroute with failure inferencing, in *Proceedings of the ACM SIGCOMM Workshop on Internet Network Management*, pp. 268–273, 2007.

[Wax88] B. Waxman: Routing of multipoint connections, *IEEE journal on selected areas in communications*, Vol. 6, No. 9, pp. 1617–1622, 1988.

[Ye06] Q. Ye and M. MacGregor: Cluster-based IP router: Implementation and evaluation, in *Proceedings of the IEEE International Conference on Cluster Computing*, pp. 1–10, 2006.

[阿留 05] 阿留多伎 明良, 加納 敏行, 江川 尚志, 菊地 芳秀, 桐葉 佳明, 岩田淳：ディペンダブルネットワーク技術による情報通信インフラ構築, NEC 技報, Vol. 58, No. 5, pp. 79–85, 2005.

[沖田 03] 沖田 英樹, 柘植 宗俊, 滝広 眞利：経路制御部の冗長化によるルータ高可用化方式, 電子情報通信学会ソサイエティ大会, Vol. B-6, No. 2, p. 84, 2003.

[沖田 04] 沖田 英樹, 拓殖 宗俊, 滝広 眞利, 平田哲彦：経路制御部の冗長化によるルータ高可用化方式, 信学技報ネットワークシステム研究会, Vol. 103, No. 624, pp. 43–46, 2004.

[関根 05] 関根 賢郎, 沖田 英樹, 長谷川 千絵, 鈴木 敏明：経路制御部冗長化ルータの特性評価, 電子情報通信学会総合大会, Vol. B-6, No. 2, p. 147, 2005.

[狩野 04] 狩野 秀一, 鈴木 一哉, 地引昌弘：ルータクラスタにおける二重パケット処理冗長方式, 情処学研報高品質インターネット研究会, Vol. 2004, No. 104, pp. 21–26, 2004.

[狩野 05] 狩野 秀一, 地引 昌弘：ルータクラスタにおける二重パケット処理冗長方式, 電子情報通信学会論文誌 B 分冊, Vol. 88, No. 10, pp. 1956–1967, 2005.

[小池 05] 小池 友岳, 白鳥 毅, 鈴木 俊範, 吉本 正明, 安藤 智和, 水上 貴司, 尾崎裕二：キャリアグレード・サービス・プラットフォーム, 沖テクニカルレビュー, Vol. 72, No. 1, pp. 24–29, 2005.

[情報 07] 情報通信審議会　情報通信技術分科会　IP　ネットワーク設備委員会：ネットワークの IP　化に対応した　安全・信頼性対策, http://www.soumu.go.jp/menu_news/s-news/2007/070418_3.html, 2007.

[総務 05] 総務省 次世代 IP インフラ研究会 IP ネットワーク WG：第三次報告書 電話網から IP 網への円滑な移行を目指して, http://www.soumu.go.jp/s-news/2005/050811_5.html, 2005.

[総務 10] 総務省：電気通信事故発生状況　(平成　21　年度), http://www.soumu.go.jp/main_content/000066638.pdf, 2010.

[大矢 10] 大矢 貴文, 三好 潤：IPsecGW 冗長構成アーキテクチャに関する一検討, 電子情報通信学会総合大会, Vol. B-6, No. 64, p. 64, 2010.

[田村 08] 田村 藤嗣彦, 土井 俊介, 市川 恭之, 本野 智治, 天野 祥行：IPsec-VPN の信頼性向上に関する一検討, 電子情報通信学会総合大会, Vol. B-6, No. 75, p. 75, 2008.

[当麻 90] 当麻 喜弘：フォールトトレラントシステム論, 電子情報通信学会, 1990.

[南谷 91] 南谷 崇：フォールトトレラントコンピュータ, オーム社, 1991.

[福田 08] 福田 亜紀, 橋本 仁, 行松 健一, 鎌村 星平, 宮村 崇, 塩本 公平：ノード次数と必要 MRC 数の関係性についての検討, 信学技報 ネットワークシステム研究会, Vol. 108, No. 203, pp. 131–134, 2008.

[福田 09] 福田 亜紀, 橋本 仁, 行松 健一, 鎌村 星平, 宮村 崇, 塩本 公平：MRC 網における OSPF 最適リンクコスト設定法適用の検討と評価, 信学技報ネットワークシステム研究会, Vol. 108, No. 457, pp. 269–272, 2009.

[北御 09] 北御門 靖宏, 大矢 貴文, 高橋 良, 市川 恭之, 三好潤：MOBIKE を応用した IPsecGW 冗長化方式の提案, 電子情報通信学会ソサイエティ大会, Vol. B-6, No. 27, p. 27, 2009.

[鈴木 04] 鈴木 一哉, 地引 昌弘：ルータクラスタ構成における経路情報共有機構の提案, 信学技報 テレコミュニケーションマネジメント研究会, Vol. 104, No. 36, pp. 47–52, 2004.

[鈴木 08a] 鈴木 一哉, 地引 昌弘：IP 高速迂回を実現する迂回可能経路の判別手法, 信学技報テレコミュニケーションマネジメント研究会, Vol. 107, No. 545, pp. 37–42, 2008.

[鈴木 08b] 鈴木 孝明, 鈴木 一哉, 柳生 智彦, 地引 昌弘：AS 間経路制御安定性向上のためのローカルリルート方式の提案, 信学技報テレコミュニケーションマネジメント研究会, Vol. 107, No. 545, pp. 31–36, 2008.

[鈴木 09] 鈴木 一哉, 地引 昌弘, 吉田 健一：リンク状態型経路制御における局所更新手法, 信学技報 インターネットアーキテクチャ研究会, Vol. 108, No. 459, pp. 25–30, 2009.

関連業績リスト

学術論文

1. K. Suzuki and M. Jibiki: Formalization and Analysis of Routing Loops by Inconsistencies in IP Forwarding Tables, *IEICE Transactions on Communications*, Vol. E90-B, No. 10, pp. 2755–2763, 2007.
2. K. Suzuki, M. Jibiki, and K. Yoshida: Selective Precomputation of Alternate Routes using Link-State Information for IP Fast Restoration, *IEICE Transactions on Communications*, Vol. E93-B, No. 5, pp. 1085–1094, 2010.

国際会議論文

1. K. Suzuki, M. Jibiki, and K. Yoshida: Comparison of Proactive and Reactive Methods for IP Fast Restoration using Localization Algorithm, in *Proceedings of 4th International Conference on Signal Processing and Communication Systems (ICSPCS)*, IEEE, 2010.

学会発表

1. 鈴木 一哉, 地引 昌弘：ルータクラスタ構成における経路情報共有機構の提案, 信学技報 テレコミュニケーションマネジメント研究会, Vol. 104, No. 36, pp. 47–52, 2004.
2. 狩野 秀一, 鈴木 一哉, 地引 昌弘：ルータクラスタにおける二重パケット処理冗長方式, 情処学研報高品質インターネット研究会, Vol. 2004, No. 104, pp. 21–26, 2004.

3. 鈴木 一哉, 地引 昌弘：IP 高速迂回を実現する迂回可能経路の判別手法, 信学技報テレコミュニケーションマネジメント研究会, Vol. 107, No. 545, pp. 37–42, 2008.

4. 鈴木 孝明, 鈴木 一哉, 柳生 智彦, 地引 昌弘：AS 間経路制御安定性向上のためのローカルリルート方式の提案, 信学技報テレコミュニケーションマネジメント研究会, Vol. 107, No. 545, pp. 31–36, 2008.

5. 鈴木 一哉, 地引 昌弘, 吉田 健一：リンク状態型経路制御における局所更新手法, 信学技報 インターネットアーキテクチャ研究会, Vol. 108, No. 459, pp. 25–30, 2009.