

ネットワークリソース利用効率化のための
トラフィック制御に関する研究

2023年 3月

松井 健一

ネットワークリソース利用効率化のための

トラフィック制御に関する研究

松井 健一

理工情報生命学院

システム情報工学研究群

筑波大学

2023年 3月

概要

社会の急速な DX 化に伴い、通信キャリアのバックボーンネットワークが生活に欠かせないインフラとなっているため、日本のインターネットトラフィックは 2 割～5 割／年のペースで増加を続けている[1]。通信キャリアは、爆発的に増加するトラフィックを収容できるネットワークを提供する必要があるが、経済的にネットワークを提供する必要もあるため、トラフィックの増大に伴う設備投資や運用コストを抑えるためにネットワークリソース利用の効率化や安定した運用が必要である。

ネットワークリソースは、ネットワーク装置やリンクの転送能力である。通信キャリアのバックボーンにおいて、ネットワーク装置は経路の冗長性を保つためリンクにより複雑に接続されており、都市部や地方の利用者を収容する大規模から小規模にわたる多種のネットワーク装置が全国に配置され転送能力にも違いがある。リソース利用を効率化するには、これらの多様なネットワーク装置の属性やリンクの接続状況を把握し、トラフィックの配置を適切に計算し制御することで、ネットワークリソースを最大限活用することが求められる。

近年は、Web 閲覧など一定のトラフィック量・通信品質で 2 拠点間の通信を行うだけでなく、映像配信、テレビ会議やメタバース空間の提供など、大容量・高品質な通信品質や他拠点へトラフィックの配信が必要な通信サービスが一般化している。このような通信品質のニーズの多様化および他拠点配信ニーズの増加に対応して、利用者が求める通信品質を提供できるネットワークが求められる。

コミュニケーション通信を支えるバックボーンネットワークにおいて増大を続けるトラフィックを効率的に収容・制御するためには、バックボーンネットワークのトラフィックの効率的な配置制御、多地点への効率的な配信制御、必要な通信品質を提供する通信制御を行い、トータルでのネットワークリソース利用及び運用効率化の実現が必

要となっている。

本論文では、2章においてトラフィック制御によるリソース利用効率化、3章においてマルチキャスト経路管理による他拠点配信の運用効率化、4章においてオンデマンドフロー管理による通信品質管理の運用効率化を実現する方式の提案と評価を行う。5章において現在の通信キャリアのバックボーンネットワークへの適用について述べる。6章において本研究のまとめと今後の課題を述べる。

目次

概要	i
目次	iii
図目次	vi
表目次	viii
1. 序論	1
1.1. 本研究の背景	1
1.2. 本論文の構成	4
2. トラヒック制御によるリソース利用効率化	6
2.1. 背景	6
2.2. トラヒックエンジニアリング技術の概要	7
2.3. 大規模ネットワークにおけるトラヒックエンジニアリングに対する要求条件	8
2.4. 従来技術の問題点	9
2.4.1. トラヒック観測	9
2.4.2. 経路計算	9
2.4.3. 経路設定	10
2.4.4. 先行研究	10
2.5. 大規模ネットワークにおけるトラヒックエンジニアリング制御方式の検討	13
2.5.1. トラヒック観測	14
2.5.2. 経路計算	15
2.5.3. 経路計算アルゴリズム	16
2.5.4. 経路設定	17
2.6. 提案方式	17

2.7.	経路計算アルゴリズム	21
2.7.1.	帯域ボトルネックの回避	21
2.7.2.	経路計算アルゴリズムの提案	22
2.8.	経路アルゴリズムの評価	28
2.8.1.	一般ユーザによるトラフィック交流を想定した評価	28
2.8.2.	データセンタへのトラフィックを想定した評価	32
2.9.	経路設定方法の比較評価	35
2.9.1.	検証項目 1：ホップ数と設定時間	37
2.9.2.	検証項目 2：LSP 数と設定時間	43
2.9.3.	検証項目 3：故障回復時間	50
2.9.4.	検証項目 4：CPU 負荷	54
2.10.	まとめ	56
3.	マルチキャスト経路管理によるリソース利用効率化	58
3.1.	背景	58
3.2.	従来方式の課題	59
3.3.	提案方式	60
3.4.	提案方式の課題	64
3.5.	提案方式の設備設計	65
3.6.	まとめ	68
4.	オンデマンドフロー管理によるリソース利用効率化	69
4.1.	背景	69
4.2.	問題点と課題	70
4.3.	提案方式	72
4.4.	制御動作	73
4.5.	システム構成	74

4.6.	特徴	75
4.7.	効果	79
4.8.	提案方式の設備設計	81
4.9.	まとめ	86
5.	現在のネットワークへの適用について	87
5.1.	バックボーンネットワーク制御への適用	87
5.2.	インテントベースのアプリケーション・ネットワーク協調制御への適用	88
5.3.	インテントベースのクラウドサーバ・アプリケーション・ネットワーク協調 制御への適用	89
6.	結論	91
6.1.	本論文のまとめ	91
6.2.	今後の課題	92
	謝辞	94
	参考文献	95
	発表論文	99

目次

図 1-1. 日本の固定通信トラフィックと移動通信トラフィック	1
図 2-1. トラフィックエンジニアリングモデル	8
図 2-2. 自律分散型トラフィックエンジニアリング制御アーキテクチャ	14
図 2-3. 集中制御型トラフィックエンジニアリング制御アーキテクチャ	14
図 2-4. トラフィックエンジニアリングサーバのシステム構成	18
図 2-5. トラフィックトランク	19
図 2-6. トラフィックエンジニアリングサーバ機能構成	20
図 2-7. 従来の経路計算の問題点	22
図 2-8. ボトルネックを回避する経路計算アルゴリズム	24
図 2-9. 提案アルゴリズムによる経路計算例	28
図 2-10. 日本全国の一般ユーザによるトラフィック交流を想定した評価に用いた網構成	30
図 2-11. リンク帯域の標準偏差	31
図 2-12. リンク帯域の標準偏差	32
図 2-13. 変形リングネットワーク	33
図 2-14. 2軸Xツリーネットワーク	34
図 2-15. 検証環境のネットワーク構成	36
図 2-16. 検証項目 1 のトポロジ構成	37
図 2-17. ネットワーク構成 (検証項目 1)	38
図 2-18. RSVP-TE 方式における試験結果 (検証項目 1)	40
図 2-19. トラフィックエンジニアリングサーバ方式における試験結果 (検証項目 1)	41
図 2-20. ホップ数に対する設定時間	42
図 2-21. 検証項目 2 のトポロジ構成	43
図 2-22. ネットワーク構成 (検証項目 2)	43
図 2-23. RSVP-TE 方式における試験結果 (検証項目 2)	46
図 2-24. トラフィックエンジニアリングサーバ方式における試験結果 (検証項目 2)	49
図 2-25. 経路数に対する設定時間	49

図 2-26. 検証項目 3 のトポロジ構成	50
図 2-27. ネットワーク構成 (検証項目 3)	51
図 2-28. 故障回復時間の試験結果 (検証項目 3)	54
図 2-29. RSVP-TE 方式における MPLS WORKBENCH の CPU 使用率 (検証項目 4)	55
図 2-30. トラフィックエンジニアリングサーバ方式における MPLS WORKBENCH の CPU 使用率 (検証項目 4)	56
図 3-1. 分散型マルチキャストシステム	59
図 3-2. マルチキャスト領域の分類	60
図 3-3. サーバ型マルチキャストシステム	61
図 3-4. サーバ型マルチキャストシステムの動作	62
図 3-5. MCRS の分割	65
図 3-6. ユーザの参加離脱の状態遷移モデル	65
図 3-7. ユーザのマルチキャスト受信要求頻度	66
図 4-1. 通信品質の多様化	70
図 4-2. オンデマンドフロー管理システム	73
図 4-3. オンデマンドフロー管理システム構成	74
図 4-4. マルチユーザ管理	76
図 4-5. ポリシの階層化	79
図 4-6. 解決すべき課題と提案システムの特徴との対応	81
図 4-7. 本システムの動作のモデル化	82
図 4-8. 一般ユーザのフロー要求動作モデル	82
図 5-1. 現在の電気・フォトニックによるバックボーンネットワーク	87

表目次

表 2-1. 日本全国からデータセンタへのトラフィックを想定した評価結果	34
表 2-2. 検証に用いた端末一覧	35
表 2-3. 端末スペック一覧	36
表 2-4. RSVP-TE 方式による試験結果	39
表 2-5. トラフィックエンジニアリングサーバ方式による試験結果	41
表 2-6. RSVP-TE 方式による試験結果	45
表 2-7. トラフィックエンジニアリングサーバ方式による試験結果	48
表 2-8. 試験概要一覧（検証項目 3）	53
表 2-9. 検証結果（方式別故障迂回時間一覧）	53
表 2-10. CPU 使用率測定対象プロセス一覧	54
表 3-1. MCRS-USER1 台が収容可能なユーザ数	67
表 4-1. ユーザ一覧	77
表 4-2. ポリシー一覧	78
表 4-3. パタン毎のフロー要求数とフロー設定数	85

1.序論

1.1. 本研究の背景

社会の急速な DX 化に伴い、通信キャリアのバックボーンネットワークが生活に欠かせないインフラとなっている。このため、図 1-1 に示すように、近年の日本のインターネットトラフィックは、2019 年以降、1 割～5 割/年のペースで増加を続けている [1]。

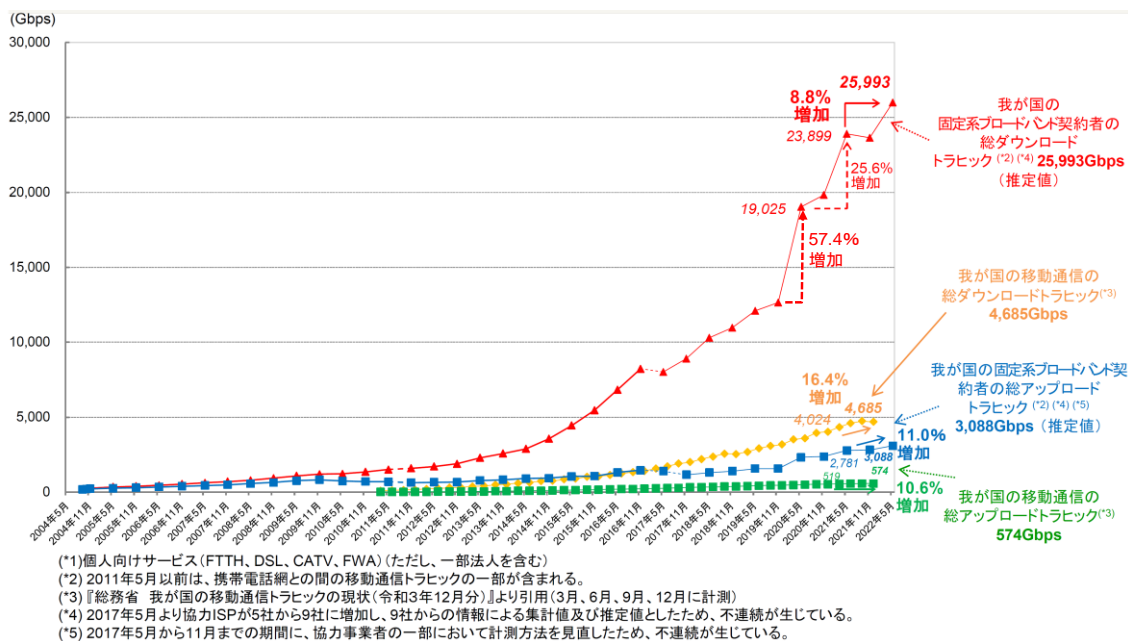


図 1-1. 日本の固定通信トラフィックと移動通信トラフィック

総務省 総合通信基盤局，我が国のインターネットにおけるトラフィックの集計結果 (2022 年 5 月分) より引用。2019 年の増加率は著者により追記。

国内通信キャリアは、爆発的に増加するトラフィックを収容できるネットワークを提

供する必要があるが、経済的にネットワークを提供する必要もあるため、トラヒックの増大に伴う設備投資や運用コストを抑えるためにネットワークリソース利用の効率化や安定した運用が必要である。

ネットワークリソースは、ネットワーク装置やリンクの転送能力である。通信キャリアのバックボーンにおいて、ネットワーク装置は経路の冗長性を保つためリンクにより複雑に接続されており、都市部や地方の利用者を収容する大規模から小規模にわたる多種のネットワーク装置が全国に配置され転送能力にも違いがある。リソース利用を効率化するには、これらの多様なネットワーク装置の属性やリンクの接続状況を把握し、トラヒックの配置を適切に計算し制御することで、ネットワークリソースを最大限活用することが求められる。これを解決するため、トラヒック情報を収集し、トラヒック経路を制御・分散配置することで、リソース利用を抑えつつ高容量のトラヒックを収容し、転送品質の低下を防ぐトラヒックエンジニアリング技術をバックボーンネットワークへ適用することが求められている。適用に向けては、トラヒックを考慮して経路制御を行う場合、トラヒック情報の遅延や差異などにより、ルータ間のトラヒック経路制御のオーケストレーションが不十分となり、通信が不安定になる可能性がある。また、トラヒックが特定のネットワーク個所に集中しないよう効率的に分散配置する経路計算を行う必要がある。

バックボーンネットワークで利用される IP アプリケーションについては、多数のユーザに大容量の映像を配信するストリームアプリケーションの利用が増大している。しかし、ユーザごとにストリームを配信するユニキャスト通信は、ユーザ数に比例してトラヒックが増大するため、リソースを圧迫する問題がある。これを解決するため、ネットワーク内の最適な箇所でパケットをコピーし、トラヒックを抑制して情報を配信するマルチキャスト通信が求められている。マルチキャスト通信を放送や映像配信などのビジネス用途に活用する場合、想定外の利用者による配信による乗っ取りやトラヒック圧迫を防ぐため、従来の IP 通信にはない、送信および受信するユーザを把握・認証する技術が必要となる。これらの課題を解決し、バックボーンネットワー

クを放送や映像配信に活用することのできるビジネス向けマルチキャスト技術が求められている。

近年は、Web 閲覧など一定のトラフィック量・通信品質で 2 拠点間の通信を行うだけでなく、映像配信、テレビ会議やメタバース空間の提供など、大容量・高品質な通信品質や他拠点へトラフィックの配信が必要な通信サービスが一般化している。また、リモートワークが急速に普及しており、バックボーンネットワークにおける IP ストリーミングアプリケーションにおいて、ビデオ会議システムなど双方向通信が拡大している。双方向のリアルタイムストリーミングは、音声や動画の品質劣化がユーザに意識されやすいため、体感品質を保つには通信遅延や揺らぎを一定以下にする必要がある。一方で、バックボーンネットワークには電子メールやウェブなどのリアルタイム性を要求しない通信など様々な種類のトラフィックが混在しており、一律の遅延制御は難しい。近年、フロー毎の優先／非優先制御を行う通信品質制御装置が活用されているが、企業網内など特定のネットワーク範囲において、あらかじめ設定された品質制御を行うものであり、幅広いユーザ通信需要をオンデマンドに対応していない。このような通信品質のニーズの多様化および他拠点配信ニーズの増加に対応して、利用者が求める通信品質を提供できるネットワークが求められている。バックボーンネットワークにおける通信品質の確保やネットワークリソースの利用を効率化するためには、一般ユーザの通信要求を把握し、通信品質制御を一元的に管理して、リアルタイムストリーミングを選択的に優先するフロー管理技術が必要となる。

これらのサーバ型トラフィックエンジニアリングシステムによるバックボーンネットワーク制御と、マルチキャスト管理と、オンデマンドフロー管理による IP アプリケーション管理を連携させることで、IP アプリケーションのリソース利用状況を把握してトラフィック分散制御を行うことができるため、本研究の成果を活用することで、バックボーンネットワークの効率的・安定的な制御を可能とすることが出来る。これらの研究は通信キャリアのバックボーンにて有効活用が可能である。

1.2. 本論文の構成

本論文の構成は以下の通りである。

第 2 章ではトラフィック制御によるリソース利用効率化について述べる。トラフィック情報と経路制御を一元管理することで、トラフィック情報の時間変動に対する収束性を向上させ、経路の安定性を向上させるサーバ型トラフィックエンジニアリング技術、トラフィック流量を最大にする経路計算方式を提案し、シミュレーションでトラフィック収容効率の向上を確認したこと、エミュレータで経路設定時間の向上を確認したことについて述べる。

第 3 章ではマルチキャスト経路管理による他拠点配信の運用効率化について述べる。ユーザの認証を行い、マルチキャスト配信を管理するサーバ型マルチキャスト管理システムを提案する。標準的なマルチキャストプロトコルを終端する機能や Web ベースの設定インタフェースを持ち、ユーザ契約やリソース利用状況に基づいたマルチキャスト経路配信を一元的に行うため、既存のプロトコルを改変することなく、ユーザやオペレータの設定により、マルチキャスト送受信者の配信管理が可能となることについて述べる。また、設備設計の指針となる処理負荷の見積もりの計算方法について述べる。

第 4 章ではオンデマンドフロー管理による通信品質管理の運用効率化について述べる。ユーザの通信品質要求をオンデマンドで受け付け、要求に基づいてバックボーンネットワーク上に配置された複数の品質制御装置を一元的に管理、設定するサーバ型オンデマンドフロー管理システムの提案について述べる。一般ユーザから管理ユーザやキャリアオペレータまで階層化したユーザ構造、および利用から契約や運用まで階層化したフロー管理ポリシーを持ち、一般ユーザのフロー管理要求の受け付けを行いネットワーク運用ポリシーと整合し管理するため、バックボーンネットワークにおけるフロー管理が可能となることについて述べる。また、設備設計の指針となる処理負荷の見積もりの計算方法について述べる。

第 5 章では現在の通信キャリアのバックボーンネットワークへの研究成果の適用について述べる。本研究は、2000 年代を中心に行ったバックボーンネットワークのリソース利用を効率化する研究開発であるが、現在のバックボーンネットワークにおける IP 通信と光通信の最適な組み合わせ、ユーザのインテントベースのクラウド・アプリケーション・ネットワークの連携制御に活用できることを述べる。

第 6 章において本研究のまとめを行う。

2. トラフィック制御によるリソース利用 効率化

2.1. 背景

インターネットの普及および社会の急速な DX 化等により，IP 通信が社会を支える重要な基盤となっている．それに伴い，通信キャリアが提供するネットワークの信頼性の向上や，爆発的に増加し続けるトラフィックを効率良く処理する技術が求められている．そこで，耐障害性を向上させ，トラフィックを制御し，効率的にネットワークリソースを利用するトラフィックエンジニアリング技術が提案されている．

トラフィックを制御するためには，パケットにラベリングを行い，明示的に経路を指定する必要がある．ラベル転送ネットワークとしては Multiprotocol Label Switching (MPLS) が活用されている[2]．MPLS に対するトラフィックエンジニアリング機構としては，Internet Engineering Task Force (IETF) の Internet Traffic Engineering Working Group[3]において標準化に向けた RFC が策定されている[4][5]．それらの標準化においては自律分散型トラフィックエンジニアリング機構が規定されている．IETF Application-Layer Traffic Optimization Working Group[6]では，その機構を活用し，アプリケーションレイヤにおけるトラフィック最適配置を目指して，ネットワークトポロジ，帯域，ドメイン間トラフィック交流およびコストの制御に関する標準化作業が進められている．

本研究を行った 2000 年代においては，自律分散型トラフィックエンジニアリング機構は，各転送ノードが flooding を用いてトラフィック情報を交換し，経路計算を行い，Signaling を用いて経路設定を行っていた．そのため，網が大規模化すると，flooding

によるトラヒック情報伝播遅延が大きくなり、経路計算、Signaling 処理に伴うプロセス負荷が増加し、経路収束の不安定化や転送性能の劣化が発生する可能性がある。

そこで、本研究では、大規模バックボーンネットワークにおいて、経路制御を一元管理し集中設定を行うことで経路設定時間を短縮しネットワークの安定性を向上させる方式、およびトラヒック流量を最大にするような迂回経路計算アルゴリズムを提案し、これらを実装したトラヒックエンジニアリングサーバを実装した。本章では提案技術とその実装および評価について述べる。

2.2. トラヒックエンジニアリング技術の概要

トラヒックエンジニアリングとは、網内のトラヒックを観測し、その結果を下にトラヒックを効率良く配置するための制御を行う技術であり、トラヒック測定、分析・モデル化、経路計算、制御のフェーズから構成される。

ネットワーク装置の属性やリンクの接続状況などの制御条件を把握し、トラヒック情報を計測・収集し、トラヒックの状況を分析・モデル化し、最適な経路を計算・制御するトラヒックエンジニアリング技術（図 2-1）が必要となる。トラヒックの状況に応じた経路の分散配置を行うことで、有限のネットワークリソースを最大限活用できる。また、トラヒック制御条件を満足する経路を自律的・継続的に測定、計算、設定するため、多層化し複雑化するネットワークのオペレーションコストを削減することも可能となる。

これらのフェーズからなるサイクルを繰り返すことにより、ネットワークの送信元から宛先へのトラヒックからなるトラヒック交流状況の変化に適応したトラヒック制御を行う。

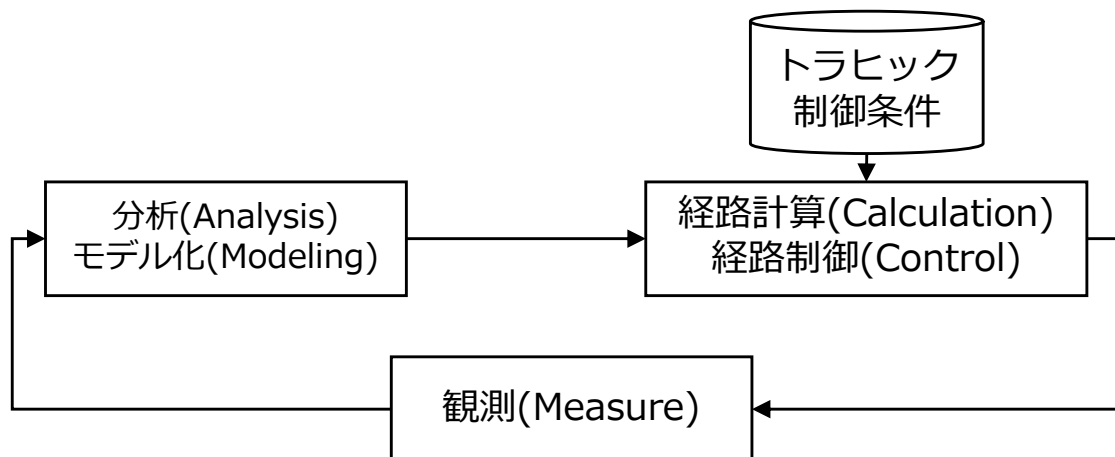


図 2-1. トラヒックエンジニアリングモデル

トラヒックエンジニアリングを行うことにより、トラヒック交流状況に応じた経路の分散が可能となり、大量のトラヒックを効率よく転送することが可能となる。また、トラヒック情報のみならず、オペレータの入力した多様なルーティングポリシーをも考慮に入れて経路計算を行うことにより、トラヒックの QoS 特性やサービスグレードに応じた QoS ベースの経路を設定することが可能になる。また、トラヒック制御条件を満足する経路を自律的・継続的に測定、計算、設定するため、多層化し複雑化するネットワークのオペレーションコストを削減することも可能となる。

2.3. 大規模ネットワークにおけるトラヒックエンジニアリングに対する要求条件

ネットワーク内のトラヒック交流状況の変化に迅速に対応したトラヒックエンジニアリングを実現するためには、まず、トラヒック情報の測定から経路制御までの各フェーズがそれぞれ迅速に行える必要がある。次に、トラヒックエンジニアリングの効率を高める観点から、経路計算で導出される経路がより最適である必要がある。そして、トラヒックエンジニアリング制御に伴う転送ノードの転送性能劣化を最小化するため、プロセッサ負荷やトラヒックエンジニアリング制御に伴うトラヒックの量の増

加を最小限にする必要がある。

2.4. 従来技術の問題点

トラヒックエンジニアリングのフェーズである，トラヒック観測，経路計算，経路制御に関して，2000年代におけるトラヒックエンジニアリング技術の概要とその問題点について述べる。

2.4.1. トラヒック観測

従来（2000年代以前，以下同様）のトラヒックエンジニアリング技術では，各転送ノードが Traffic Engineering Extensions to OSPF (OSPF-TE)[7]や Traffic Engineering Extensions to IS-IS (IS-IS-TE) [8]等のトラヒックエンジニアリング向けに拡張されたルーティングプロトコルを用いてトラヒック情報を収集する．いずれの方式もトラヒック情報伝播手段として flooding を用いるため，各転送ノードの持つトラヒック情報が完全に同期するまでの遅延が大きくなり，帯域の圧迫に伴う転送性能の劣化も発生する．

2.4.2. 経路計算

従来のトラヒックエンジニアリング技術では，各転送ノードが収集したトラヒック情報を基に経路計算を行うため，プロセッサ負荷が増大し，転送性能が劣化する．また，OSPFやIS-IS等における経路選択アルゴリズムは，各経路のリンク容量とホップ数を基にした経路計算を行うため，トラヒックが特定リンクに集中する可能性がある．

OSPF-TE や IS-IS-TE 等のトラヒックエンジニアリング拡張が施されたアルゴリズムは，リンク毎の残余帯域と要求帯域の比較を行い，要求帯域より残余帯域のほうが大きければその経路を採用するという拡張が行われているだけであり，特定リンクのリンク帯域付近までトラヒックが集中してしまうという問題点がある．

2.4.3.経路設定

従来のトラヒックエンジニアリング技術では、バックボーンネットワークへトラヒックを入力する転送ノードが、Signaling (RSVP-TE[9], CR-LDP[10]) を用いて、ホップバイホップに経路を設定していく。そのため、各転送ノードには、Signaling 処理に伴うプロセッサ負荷が発生し、転送性能に悪影響を及ぼす。また、ホップバイホップで経路設定を行っていくため、バックボーンネットワーク内でのホップ数に比例して経路設定時間が増大する。さらに、複数の転送ノードが同時に経路設定しようとするリソースの競合が発生する可能性があり、その結果、当初の想定どおりの経路設定が出来ない可能性がある。

2.4.4.先行研究

分散型経路制御の安定性に関する研究として、大規模ネットワークにおける OSPF-TE プロトコル安定性のシミュレーション評価が挙げられる[11]。本研究では、評価モデルとして、米国の都市を結ぶ大規模 Internet Service Provider (ISP) のバックボーンネットワークを模擬し、292 のノードと 765 のリンクを持つハブアンドスポーク型のネットワークトポロジを作成し、ノードプロセッサにおけるルーチング処理負荷、経路変動数、経路収束時間を評価している。ノードが故障した際の安定性を評価するため、1, 5, 10 ノードが同時に故障する 3 通りの故障シナリオをシミュレートし、故障ノードが多くなるほど、経路変動数が多くなることを示している。また、ノード故障が発生した後の経路収束時間としては約 80 秒を要することを示している。

シミュレーション評価から、分散型である OSPF-TE では、各ノードへの情報伝搬や経路設定に時間を要することが示された。本研究結果に対する考察として、分散型である OSPF-TE によるバックボーンネットワークの経路制御は長い時間をかけることで可能と考えられるが、ネットワーク規模が大きくなるにつれて経路制御時間がより長くなることが想定される。通信キャリアのバックボーンネットワークにおいては、

故障や輻輳が発生しても、短い時間で迅速に経路制御を行うことによって通信の安定性を保つことが求められるため、安定した通信には分散型が必要とする経路制御時間は十分でないと考えられる。

その他の分散型経路制御の経路安定性の研究としては、初期の研究として Khanna と Zinky による研究[12]が挙げられる。この研究は ARPANET における経路の振動を調べ、安定性をもたらすためのルーティングメトリクスの変更を提案し、ネットワークの制御理論としてフィードバックループモデルを提案している。Boel による研究[13]は、短期および長期の経路制御の性能を向上させるためには、故障情報を迅速に、修復情報をゆっくりと広めることが必要であることを示している。

インターネットから収集した統計情報を用いて、経路の安定性に関する定量的な情報を導く研究も行われている。Chinoy によるインターネット経路情報の変動についての研究[14]では、経路情報アップデートの内容（何パーセントが実際に新しい経路につながる情報を提供したか）、経路情報アップデート時間、経路変動の大きさと分布、到達不能時間など、インターネット経路情報の変動を分析している。Labovitz らによる研究[15]では、インターネットにおけるネットワーク故障が経路の安定性に及ぼす影響について、いくつかのインターネット故障の際に収集されたデータを使用し、その原因と思われるものを分析し、経路変動の発生頻度や、経路変動の原因となるネットワーク故障の平均発生時間について調査している。Labovits らによる研究[16]では、ドメイン間ルーティングにおけるネットワーク故障と経路情報アップデート、回復時間の関係を調べている。Rybowski らによる研究[17]では、OSPF によりリンク故障時に経路情報がネットワーク内のルータに送信される状況をシミュレーションで評価している。シミュレーション評価によって、各ルータへ経路情報のアップデートが行われる過程で、ルータが経路を非同期で更新するため過渡的なマイクロループやブラックホールが発生しパケットがロスするなど、経路が収束するまで転送が不安定になることを示している。

このように、経路の安定性については、ネットワークの制御理論としてフィードバックループが提案されて以降、経路の安定性に影響するパラメータの抽出や、実際のインターネット挙動を分析した経路安定性の調査、分散型制御による非同期なルータの経路設定による転送の不安定性の評価は行われたが、経路の安定性を根本的に高めるアーキテクチャの研究は行われていない。

また、トラフィックエンジニアリングにおける QoS ベースの経路制御の研究としては、Apostolopoulos らによる研究[18]では、QoS ベースの経路制御時間に影響を与えるパラメータとして、経路情報をアップデートする条件、条件を発動させる感度、アップデートの制限タイマーを示し、これらのパラメータを変化させた場合の経路情報のアップデートと QoS ベースの経路設定の成功数との関係性を評価している。Shaikh らによる研究[19]も経路情報のアップデート条件と QoS ベースの経路設定の関係を評価している。Hao らの研究[20]では、ネットワークを複数のドメインに分割し、ドメイン内の経路情報を集約した後にドメイン間の経路情報として交換することで、経路情報のアップデートを減少させる方式を示している。Shaikh らの研究[21]では、ネットワークが過負荷の状態において経路情報が伝達されない場合、経路変動の回数が増加することを示した。Németh らの研究[22]では、OSPF における Equal Cost Multipath (ECMP) を用いて、ノード間のリンクを仮想的に複数経路に分割し、リンク容量に対しトラフィック量を分散収容できるように経路を分割する手法を提案している。リンク間の収容効率は向上するが、複数リンク間のトラフィック割り当て競合が発生するため、ネットワーク全体での最適化が困難となる課題がある。

このように、QoS ベースの経路制御については、分散制御における経路情報のアップデートと経路制御の関係性の評価、経路情報がアップデートされない場合の経路変動への影響、ネットワーク分割による経路情報の集約の工夫、リンク単位での経路分散が提案されていたが、大規模ネットワーク全体のノードやリンクを一元的に管理する状況を想定しておらず、通信キャリアが必要とする大規模なバックボーンネットワ

ークのリソースを一元的に管理する経路制御の提案は行われていない。

2.5. 大規模ネットワークにおけるトラヒックエンジニアリング制御方式の検討

本節では、従来技術の問題を解決し、大規模ネットワークにも適用可能なトラヒックエンジニアリング制御方式を検討する。

トラヒックエンジニアリング制御アーキテクチャとしては、自律分散型と集中制御型の2種類がある。

自律分散型とは、各転送ノードにトラヒックエンジニアリングの各フェーズの機能を配備し、転送ノードが他の転送ノードと協調的に動作することによって、トラヒック制御の目的を達成しようとするアーキテクチャである（図 2-2）。本研究を行った2000年代のトラヒックエンジニアリングは、バックボーンネットワークに配置された転送ノードである Edge Node (EN), Core Node (CN) が、ルーチングプロトコルを用いて、トラヒック情報を交換し、それぞれの転送ノードが経路を計算し、設定を行うなど、自律分散的に転送ノードが経路の計算と設定を行う方式が主流であった。

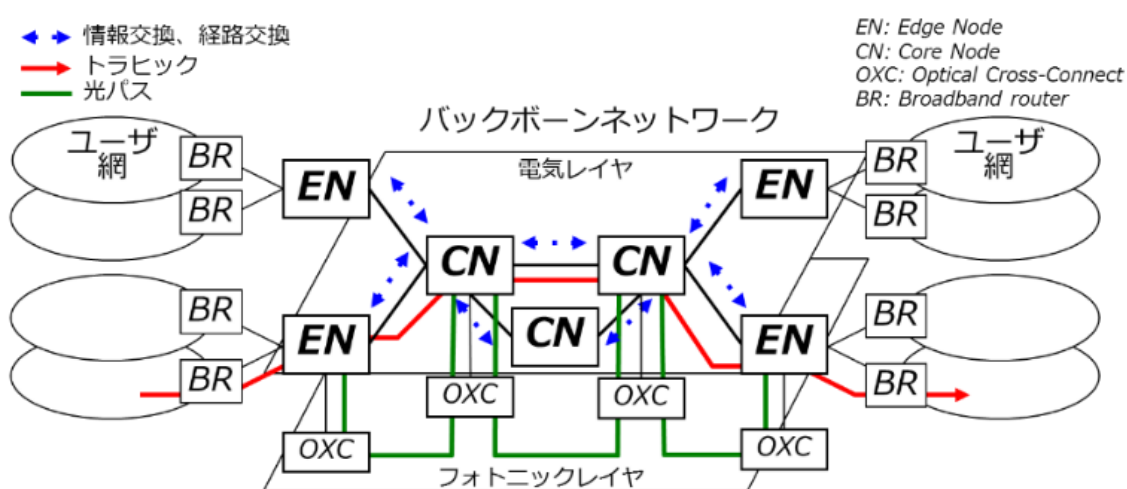


図 2-2. 自律分散型トラヒックエンジニアリング制御アーキテクチャ

一方，集中制御型とは，トラヒックエンジニアリング機能をサーバに配置し，転送ノードはサーバからの制御によりトラヒック制御の目的を達成しようとするアーキテクチャである（図 2-3）．電話網の交換機や広域ネットワークサービスプラットフォームは集中制御型のアーキテクチャを採用している．

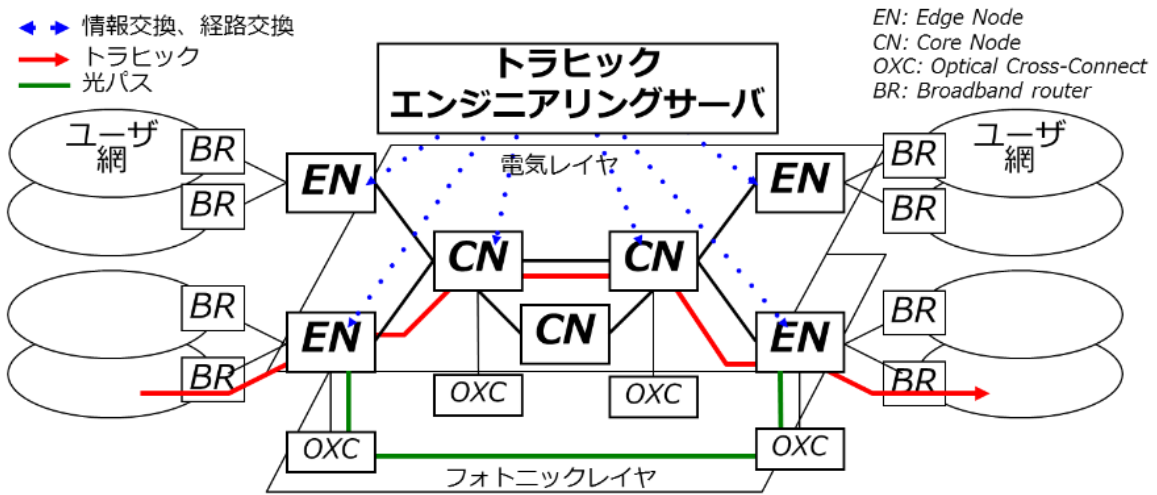


図 2-3. 集中制御型トラヒックエンジニアリング制御アーキテクチャ

以下では，トラヒックエンジニアリングのフェーズ毎に比較を行い，最適な機能配備を検討する．

2.5.1. トラヒック観測

トラヒック観測のフェーズは，自ノードにおけるトラヒックを観測するトラヒック観測機能と，他ノードにおけるトラヒック観測結果を収集するトラヒック情報収集機能の 2 つに分けられる．ネットワーク内のトラヒック交流状況を把握するためには，トラヒック観測ポイントは複数必要であり，トラヒック観測機能は，各転送ノードが機能配備する必要がある．トラヒック収集機能に関しては，自律分散型と集中制御型

のアプローチが考えられる。

自律分散型によるトラフィックエンジニアリング制御では、**flooding** を用いてトラフィック情報を収集するため、ネットワーク規模が大きくなるほどトラフィック情報の収集に時間がかかり、**flooding** が帯域を圧迫する。具体的には、ある転送ノードから他の転送ノードに対しネットワーク情報を伝搬し、ネットワーク内の全転送ノードで情報の同期を取る **flooding** は従来の経路制御においても行われていたが、トラフィックエンジニアリングの場合、ネットワークのリソース情報（リンクの利用可能帯域など）も伝搬する必要があるため、**flooding** 情報量が増加する。リンクの利用可能帯域などのリソース属性情報は変化するため、定常的に **flooding** が発生する。ネットワークが大規模化するに従って **flooding** 情報の変動が多くなり、結果として、通信の安定性が損なわれる可能性がある。

一方、集中制御型によるトラフィックエンジニアリング制御では、トラフィックエンジニアリングサーバが各転送ノードから直接トラフィック情報を収集するため、トラフィック収集が迅速であり、トラフィック収集に伴う帯域の圧迫が最小限となる。さらに、サーバは転送ノードから独立しているため、様々な種別のルータから構成されるネットワークに対する適用性も高いと考えられる。

そのため、集中制御型のほうが有利である。

2.5.2.経路計算

自律分散型における最適経路の計算は、転送ノードが個別に行うため、転送ノードにおけるプロセッサ負荷が増大し、転送性能の劣化に影響する。また、複数ルータによる経路設定の競合が懸念される。トラフィックの分散を考慮して経路を設定する場合、リソース割り当てを考慮する必要がある。複数のルータが同時に同一リンクに対して経路を設定しようとした場合、当該リンクの利用可能帯域の確保に競合が発生し、結果として、トラフィックと利用可能帯域にずれが生じ、通信の安定性が損なわれる可能

性がある。

集中制御型によるトラヒックエンジニアリング制御では、トラヒックエンジニアリングサーバのみが経路計算を行い、転送ノードでは経路計算を行わないため、転送ノードにプロセッサ負荷はかからず、転送性能を劣化させることは無い。また、サーバはネットワーク全体のリソース情報、経路、トラヒック交流を把握し、ネットワーク内の全ての経路の設定状況を踏まえたリソース割り当て、経路計算が可能である。ネットワーク情報はルータからサーバ間に伝搬されるため、ルータ間を flooding する必要がなく、通信の不安定性を回避できる。さらに、サーバはルータから独立しているため、様々な種別のルータから構成されるネットワークに対する適用性も高いと考えられる。

そのため、集中制御型のほうが有利である。

2.5.3.経路計算アルゴリズム

トラヒックエンジニアリングによる経路分散の効率の観点から、最適な経路計算アルゴリズムを検討する。

経路計算アルゴリズムとしては、OSPF-TE、IS-IS-TE 等の Dijkstra 法をベースとしてリンク帯域の利用の最小化を考慮した従来のアルゴリズムと、Dijkstra 法をベースとしながらリンク帯域の利用の平準化に重きを置いた提案アルゴリズムがある。

OSPF-TE や IS-IS-TE 等のアルゴリズムは、各経路のリンク容量とホップ数を基にして、リンク毎の残余帯域と要求帯域の比較を行い、要求帯域よりも残余帯域の方が大きければその経路を採用するだけであり、ネットワーク内で利用するリンク帯域の合計値の最小化を目的としているため、特定リンクのリンク帯域の上限近くまでトラヒックが集中してしまう可能性がある。

提案アルゴリズムは、経路内のリンク毎の残余帯域を比較し、もっとも残余帯域の

大きい経路を選択する。リンク帯域の利用の平準化を行うため、トラヒックが集中するボトルネックリンクを回避することができるなど、従来のアルゴリズムよりも効率的のトラヒックを配置することが可能であると考えられる。

2.5.4.経路設定

自律分散型による経路設定では、まず、**Signaling** 処理に伴い転送ノードのプロセッサ負荷が増大し、転送性能の劣化を引き起こす。次に、経路上の1つ1つの転送ノードに対して順番に、ホップバイホップに経路設定を行うため、経路上の転送ノード数すなわちホップ数に比例して設定時間が増大する。その結果、複数の転送ノード間でリソースの競合が発生し、正しく経路設定がされない可能性がある。

集中制御型による経路設定では、トラヒックエンジニアリングサーバが各転送ノードに直接経路設定を行う。そのため、転送ノードにかかるプロセッサ負荷を最小限に抑えることが可能である。また、**Signaling** のようにホップバイホップで設定を行わないため、ネットワークが大規模化しても高速に経路設定が可能である。さらに、トラヒックエンジニアリングサーバが全経路の設定を行うため、リソース競合が発生しない。

そのため、集中制御型のほうが有利である。

2.6. 提案方式

以上の考察により、本研究ではサーバ型トラヒックエンジニアリング方式（図 2-4）を提案する[23][24][25][26].

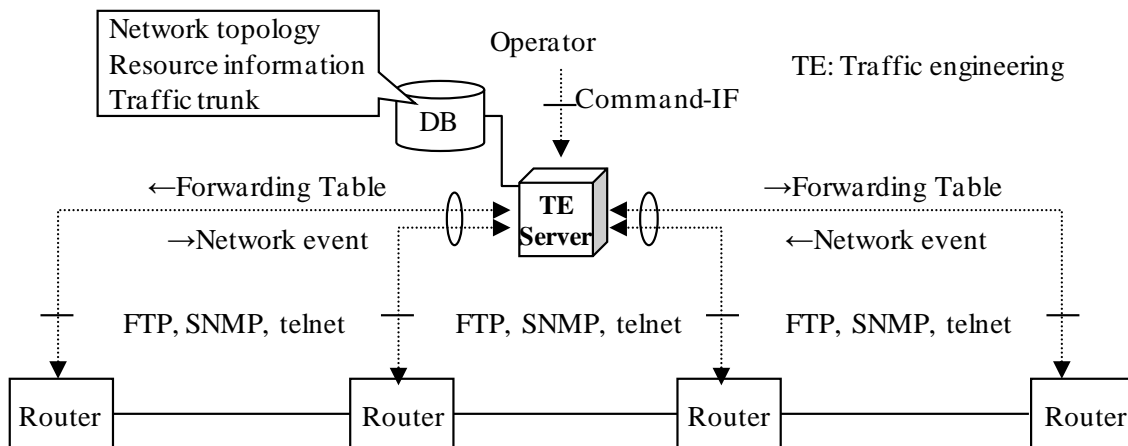


図 2-4. トラフィックエンジニアリングサーバのシステム構成

トラフィックエンジニアリングサーバは、ルーチングポリシーとネットワーク情報に従ってネットワークのルーチングを制御するため、次の順序でルーチング経路を設定する。まず、ネットワーク運用者が設定したトラフィックトランク（トラフィックフローの集合体）に指定された帯域を、ネットワークリソースから得た情報をもとに割り当てる。次に、その属性を満たす経路を計算する。最後に、各ネットワークノード固有のルーチングテーブルを構築し、関連するノードにルーチングするためのテーブルをセットアップする。MPLS ネットワークでは、サーバは FTP, telnet, SNMP のインタフェースを持ち、ネットワークノードに接続して転送テーブルを設定したり、ネットワークリソースやイベントに関する情報を受け取ったりすることができる。また、ネットワークトポロジやトラフィックトランクの仕様などの運用情報を受信するためのオペレータとのコマンドインターフェイスも備えている。

ここで、トラフィックトランク属性（図 2-5）は、トラフィックエンジニアリングにおける制御単位であり、同一の属性を適用する（図に記載している推定使用帯域、優先順位などの全ての属性が一致する）複数のフローを集約したものである[27]。

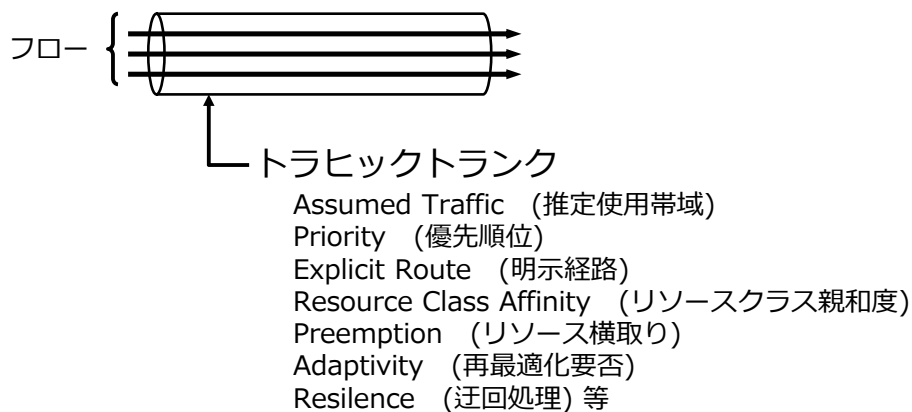


図 2-5. トラヒックトランク

経路計算・管理は、標準で規定されたネットワークリソース属性とトラヒックトランク属性を元に、帯域の最大化、明示経路の包含、等の各種属性を満たす経路を計算する。経路設定は、経路計算後、経路上の各ネットワーク装置の転送テーブルに、FTP, SNMP, telnet 等により、並列に設定を行う。ネットワーク装置に対する設定情報は複数必要となるが、設定時間短縮のため 1 つのファイルにまとめて送信し、ネットワーク装置が展開し指定された順に設定を行う。

これにより、従来の分散型トラヒックエンジニアリングではネットワーク装置が順々に経路を設定することに比べて高速に経路を設定できるため、遅延によるネットワークリソース割当競合が発生せず効率的にネットワークリソースを利用できる。

実装したトラヒックエンジニアリングサーバの機能構成を図 2-6 に示す。

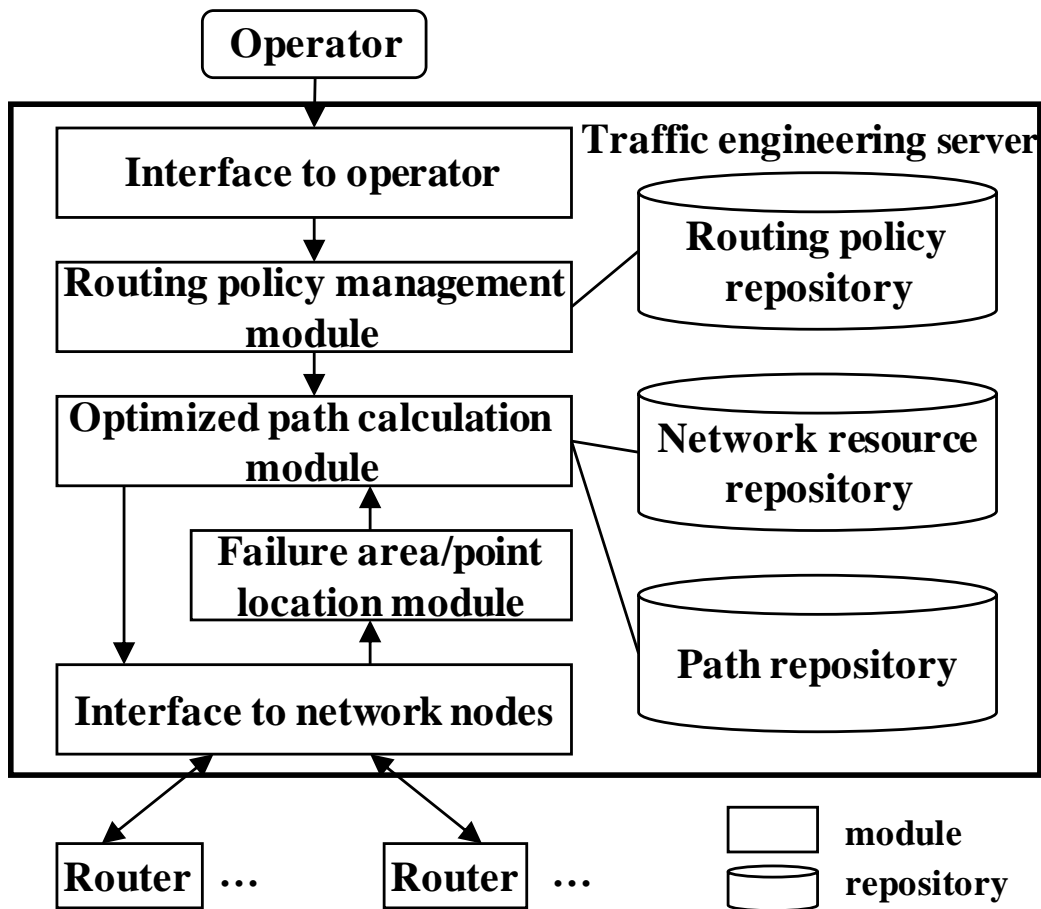


図 2-6. トラフィックエンジニアリングサーバ機能構成

1. 管理機能 (Interface to operator)

トラフィックエンジニアリングサーバにネットワーク固有情報 (ルータ種別, ネットワークトポロジ, IP アドレス等のネットワーク構成情報) を設定するインタフェースを提供する. ネットワークオペレータが本機能を利用してトラフィックエンジニアリングサーバの動作に必要な設定を行う.

2. ルーティングポリシー管理機能 (Routing policy management module)

ネットワークのトラフィック交流状況, リンクの利用可能帯域の管理を行う.

3. 経路計算機能 (Optimized path calculation module)

トラフィック交流に基づき、ネットワーク構成情報、リンクの利用可能帯域を満たしつつ、リンク帯域の利用効率を最大化する経路を計算する。

4. 故障判定機能 (Failure area/point location module)

転送ノードからの故障情報に基づき、故障個所を特定し、ネットワーク構成情報を更新する。

5. 転送ノード制御機能 (Interface to network nodes)

転送ノードへ転送テーブルを設定するインタフェース機能を提供する。転送方式としては、FTP, SNMP, telnet に対応する。

6. リポジトリ (Routing policy, Network resource, Path repository)

リポジトリは、TE サーバが保持すべき情報を格納する。TE サーバは、ルータ間のネットワーク接続帯域、接続状況であるネットワーク構成情報を格納するリポジトリ、リンクの利用可能帯域などのネットワーク情報を格納するリポジトリ、計算した経路情報を格納するリポジトリを持つ。

2.7. 経路計算アルゴリズム

2.7.1. 帯域ボトルネックの回避

トラフィックをスムーズに転送するには、帯域ボトルネックを回避し、転送に必要な帯域を確保した経路を計算する必要がある。そのような経路は、経路上のリンクにおける残帯域(帯域 - トラフィック)の最小値を比較し、これが最大である経路を選択することで求められる。

一方、本研究を行った 2000 年代の経路計算方式 (Dijkstra 法) は、始点からの確定

済の経路のリンク使用帯域の合計値を保持して計算を行い、経路上のリンク使用帯域の合計値が最小の経路を選択するが、各リンクの使用帯域の偏りを考慮しないため、経路が集中し残容量が少なくなるボトルネックリンクが発生する可能性がある。このため他のリンクの帯域が十分あるにもかかわらずトラフィック量が制限されることで、リソース利用効率が低下する可能性がある。このようにボトルネックが発生した状況を図 2-7 に示す。図において各リンクの帯域は同一とする。

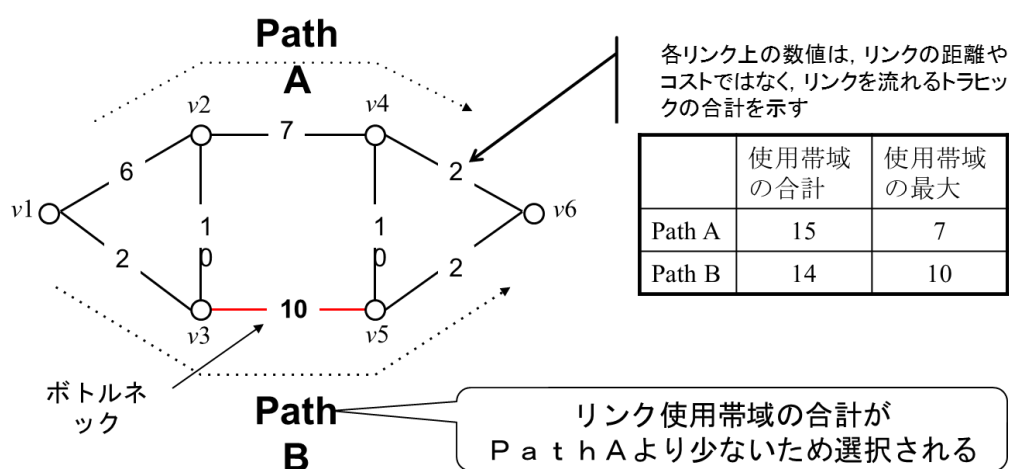


図 2-7. 従来の経路計算の問題点

経路 A は、経路 B よりも残帯域の最小値が大きい。これは経路 A には経路 B にあるボトルネックとなりうるリンクが少ないことを示す。帯域ボトルネックを回避するためには、path A のようにコストの最小値が最大となる経路を選択する必要がある。しかしながら、従来の経路計算アルゴリズムは、残帯域の合計が最小となる経路を選択する[28]。このため、経路 B を選択することとなる。

2.7.2.経路計算アルゴリズムの提案

この課題を解決するため、経路上の各リンク使用帯域の最大値を求めそれが最小の経路を選択する経路計算方式を提案する。ボトルネックリンクを回避する経路を計算

するため、リソース利用効率を向上させることが可能となる。

経路計算は次のステップで行う。

1. Dijkstra 法における経路探索において、始点からの確定済の経路のリンク使用帯域の合計値を保持するのではなく、始点から確定済の経路のリンク使用帯域のうち最も大きい値を保持する。
2. 保持した値が最も小さいノードについて、隣接ノードへのリンク使用帯域が最も小さいリンクを経路に選択する。
3. これを繰り返し、複数の経路候補のリンク使用帯域の最大値のうち、その値が最も小さいリンク使用帯域を持つ経路を選択できる。これによりボトルネックを回避する経路を計算する。

図 2-8 に経路計算アルゴリズムを示す。

<定義>

G:計算対象となるネットワーク

V:すべてのノードの集合

U: v_1 からの最適経路が求まったノードの集合

v_1 : 経路計算を開始するノード

v_{adj} : ノード v の隣接ノード

$c(v)$: v_1 からノード v までの経路上のリンクコストの最小値

$adj(v)$: v_1 から v までのリンクコストの最小値を持つ v の隣接ノード

$w(v_1, v_2)$: リンク (v_1, v_2) のコスト

<経路計算アルゴリズム>

```

begin

  V :=  $\forall v \in G$ ;

  U :=  $\varphi$ ;

  c_(v1) := 0;

  c_(v) := infinity, for  $\forall v \in \{G; \text{except } v1\}$ ;

  while U  $\neq$  V do

    begin

      select v, which has min c_(v):  $v \in \{V-U\}$ ;

      U := U  $\cup$  {v};

      for  $\forall v\_adj \in \{V-U\}$ , which has a link to v do

        begin

          c_(v_adj) := min{c_(v_adj), max{c_(v), w(v, v_adj)}};

          if c_(v_adj) < max{c_(v), w(v, v_adj)}

            adj_(v_adj) := v;

          endif

        end

      endfor

    end

  endfor

end

```

図 2-8. ボトルネックを回避する経路計算アルゴリズム

図 2-8 を用いた具体的な経路計算のステップの例を図 2-9 に示す。この例では、ノード v1 から v6 までの帯域ボトルネックを回避した経路を計算している。図中の表

は計算中のパラメータ値を示している。

図 2-9 の計算のステップを説明する。アルゴリズムに基づき、Step1 ではネットワーク上のノードを V に格納し、 $v1$ からノード v までの経路上のリンクコストの最小値 $c(v)$ には、発ノードである $v1$ の値を 0、他のノードは無限大と設定する。

Step2 では、 $v1$ を計算対象に選定し、 $v1$ の隣接ノードを $v2$ と $v3$ であると求め、 $c(v2)$ を $v1-v2$ 間のリンクコストとしてリンクの使用帯域 6 と設定し、 $c(v3)$ には同様に $v1-v3$ 間のリンク使用帯域 2 と設定する。 $v1$ から v までのリンクコストの最小値を持つ v の隣接ルータ $adj(v)$ に、 $adj(v2)=v1$ 、 $adj(v3)=v1$ と設定する。 $v1$ からの最適経路が求まったノードの集合 U に、計算対象として選定した $v1$ を設定する。

Step3 では、 U に含まれておらず V のみに含まれているノードのうち、 $c(v)$ が最小である $v3$ を選択し、 $v3$ の隣接ノードを $v2$ と $v5$ であると求め、 $c(v2)$ を $v3-v2$ 間のリンクコストとしてリンクの使用帯域 10 と現在の設定の 6 を比較して少ない 6 のままで設定し、 $c(v5)$ には $v3-v5$ 間のリンク使用帯域 10 を設定する。 $v1$ から v までのリンクコストの最小値を持つ v の隣接ノード $adj(v)$ には $adj(v2)=v1$ のままで設定し、 $adj(v5)=v3$ と設定する。 $v1$ からの最適経路が求まったルータの集合 U に、計算対象として選定した $v3$ を加える。

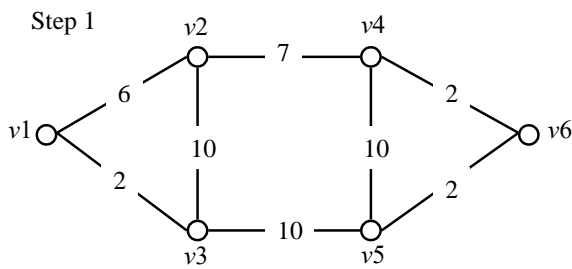
Step4 では、 U に含まれておらず V のみに含まれているノードのうち、 $c(v)$ がもっとも大きいノード $v2$ を選択する。 $v2$ の隣接ノードを $v1$ と $v4$ であると求め、 $c(v2)$ を $v2-v4$ 間のリンクコストとしてリンクの使用帯域 7 と現在の設定の 6 を比較して少ない 6 のままで設定し、 $c(v4)$ には $v2-v4$ 間のリンク使用帯域 7 を設定する。 $v1$ から v までのリンクコストの最小値を持つ v の隣接ノード $adj(v)$ に、 $adj(v2)=v1$ のままで設定する。計算対象として選定した $v2$ を、 $v1$ からの最適経路が求まったルータの集合 U に加える。

Step5 では、 U に含まれておらず V のみに含まれているノードのうち、 $c(v)$ がもっ

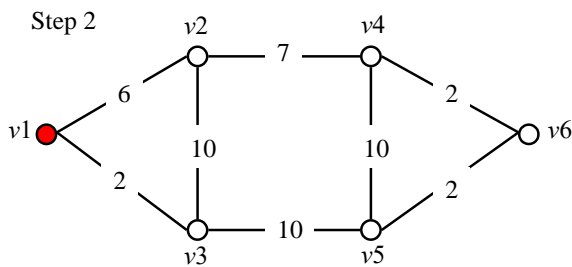
とも大きいノード v_4 を選択する。 v_4 の隣接ノードを v_2 と v_5 であると求め、 $c(v_4)$ を v_4 - v_5 間のリンクコストとしてリンクの使用帯域 7 と現在の設定の 10 を比較して少ない 7 のままで設定する。 v_1 から v までのリンクコストの最小値を持つ v の隣接ノード $\text{adj}(v)$ に、 $\text{adj}(v_4)=v_2$ のままで設定する。 計算対象として選定した v_4 を v_1 からの最適経路が求まったルータの集合 U に加える。

Step6 では、 U に含まれておらず V のみに含まれているノードのうち、 $c(v)$ がもっとも大きいノード v_5 を選択する。 v_5 の隣接ノードを v_4 と v_3 であると求め、 $c(v_4)$ を v_3 - v_5 間のリンクコストとしてリンクの使用帯域 7 と現在の設定の 7 を比較して同数のため 7 のままで設定する。 v_1 から v までのリンクコストの最小値を持つ v の隣接ノード $\text{adj}(v)$ に、 $\text{adj}(v_4)=v_2$ のままで設定する。 計算対象として選定した v_5 を、 v_1 からの最適経路が求まったルータの集合 U に加える。

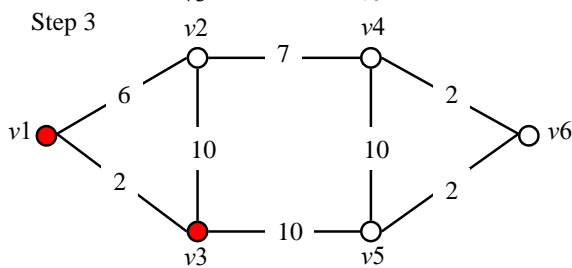
Step7 では、 U に含まれておらず V のみに含まれているノードがなくなったため、 計算を終了する。 求める経路は、 $\text{adj}(v)$ をたどっていくことで求めることができる。 今回は v_6 - v_1 間のボトルネックを回避した経路選択のため、 $\text{adj}(v_6)=v_4$ 、 $\text{adj}(v_4)=v_2$ 、 $\text{adj}(v_2)=v_1$ から、 v_6 - v_4 - v_2 - v_1 の経路が選択された。 この経路は pathA であり、 pathB よりもボトルネックが回避されている。



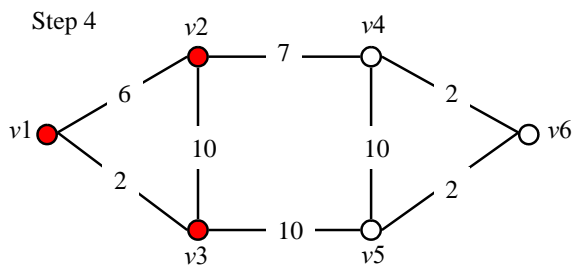
c_(v1)	c_(v2)	c_(v3)	c_(v4)	c_(v5)	c_(v6)
0	∞	∞	∞	∞	∞
adj_(v1)	adj_(v2)	adj_(v3)	adj_(v4)	adj_(v5)	adj_(v6)
V			U		
v1, v2, v3, v4, v5, v6					



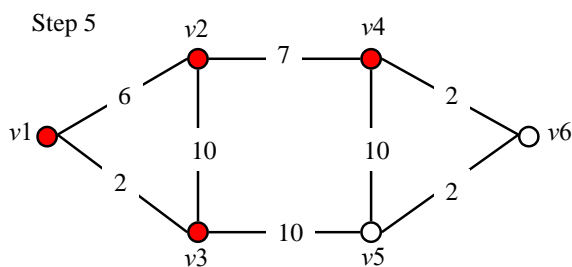
c_(v1)	c_(v2)	c_(v3)	c_(v4)	c_(v5)	c_(v6)
0	6	2	∞	∞	∞
adj_(v1)	adj_(v2)	adj_(v3)	adj_(v4)	adj_(v5)	adj_(v6)
	v1	v1			
V			U		
v1, v2, v3, v4, v5, v6					
v1					



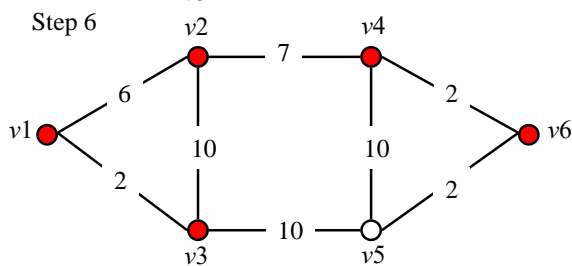
c_(v1)	c_(v2)	c_(v3)	c_(v4)	c_(v5)	c_(v6)
0	6	2	∞	10	∞
adj_(v1)	adj_(v2)	adj_(v3)	adj_(v4)	adj_(v5)	adj_(v6)
	v1	v1		v3	
V			U		
v1, v2, v3, v4, v5, v6					
v1, v3					



c_(v1)	c_(v2)	c_(v3)	c_(v4)	c_(v5)	c_(v6)
0	6	2	7	10	∞
adj_(v1)	adj_(v2)	adj_(v3)	adj_(v4)	adj_(v5)	adj_(v6)
	v1	v1	v2	v3	
V			U		
v1, v2, v3, v4, v5, v6					
v1, v3, v2					



c_(v1)	c_(v2)	c_(v3)	c_(v4)	c_(v5)	c_(v6)
0	6	2	7	10	7
adj_(v1)	adj_(v2)	adj_(v3)	adj_(v4)	adj_(v5)	adj_(v6)
	v1	v1	v2	v3	v4
V			U		
v1, v2, v3, v4, v5, v6					
v1, v3, v2, v4					



c_(v1)	c_(v2)	c_(v3)	c_(v4)	c_(v5)	c_(v6)
0	6	2	7	7	7
adj_(v1)	adj_(v2)	adj_(v3)	adj_(v4)	adj_(v5)	adj_(v6)
	v1	v1	v2	v6	v4
V			U		
v1, v2, v3, v4, v5, v6					
v1, v3, v2, v4, v6					

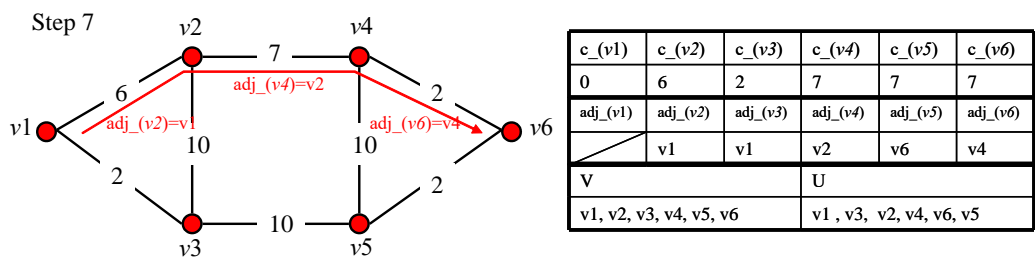


図 2-9. 提案アルゴリズムによる経路計算例

2.8. 経路アルゴリズムの評価

一般的な経路計算アルゴリズムである Dijkstra と帯域ボトルネック回避アルゴリズムそれぞれを用いて経路を計算し、トラヒック収容効率を比較評価した。

比較においては、以下の通り、一般ユーザのトラヒック収容の効率性の評価、および、企業ユーザのデータセンタトラヒック収容の効率性の評価、の2つの観点で評価を行った。

- (1) 日本全国の一般ユーザによるトラヒック交流を想定した評価
- (2) 日本全国からデータセンタへのトラヒックを想定した評価

2.8.1. 一般ユーザによるトラヒック交流を想定した評価

大規模ネットワークを想定した評価を行うため、日本全国の一般ユーザを収容する通信キャリアのバックボーンネットワークを想定した評価ネットワークを作成した [29].

<ネットワーク構成>

ネットワーク作成にあたっては、商用ネットワークに必要とされる以下の条件を満たすよう、作成した。

- ・保守性のためノードは階層構成とする

- ・ 経済性のため遠隔のノード間はリンクで接続しない
- ・ 故障対策のためノード間は複数経路を設定可能とする
- ・ 利用者数に応じてノードを配置する

ネットワーク装置の接続形態は、トラヒックの効率的な収容を目的として階層構成を採用し、ユーザを収容するネットワーク装置である Edge Node (EN)、EN間のトラヒックを中継する Core Node (CN)、CN間のトラヒックを集約して中継する Significant Core Node (SCN)の3種類のネットワーク装置を日本全国に配置し、トラヒックの集約を行うことでリンク数を抑えたバックボーンネットワークを構築した。

SCNは、地形を元に日本の東西南北にそれぞれ1台ずつ設置した。SCN間のトラヒックは、他のルータ間の Traffic よりも混雑する事が予想されるため、なるべくフルメッシュにすることにした。また、各SCNは、その下に複数のCNをもつものとした。

CNは人口分布に従って配置した。CN-SCN間のリンク故障、及びSCN間の経路冗長化を考慮し、同一SCN間の子CN同士はリンクで接続し、また、異なるSCNの子CN同士でも、地域的に隣接しているCNは相互に接続した。

ENは、人口分布に基づき、各CNに4つまで配置した。また、地理的に分散したリンクでノード間の経路を複数設定し、災害などによりリンクが切断されても通信が継続できるネットワーク構成とした。

以上の考えに基づき、図 2-10 の通り、評価ネットワークを構成した。簡略化のためENの記載は省略している。

日本全国をカバーするネットワークモデルを作成

- ユーザを収容するEN(Edge Node)
- EN間のトラフィックを中継するCN(Core Node)
- CN間のトラフィックを集約して中継するSCN(Significant Core Node)

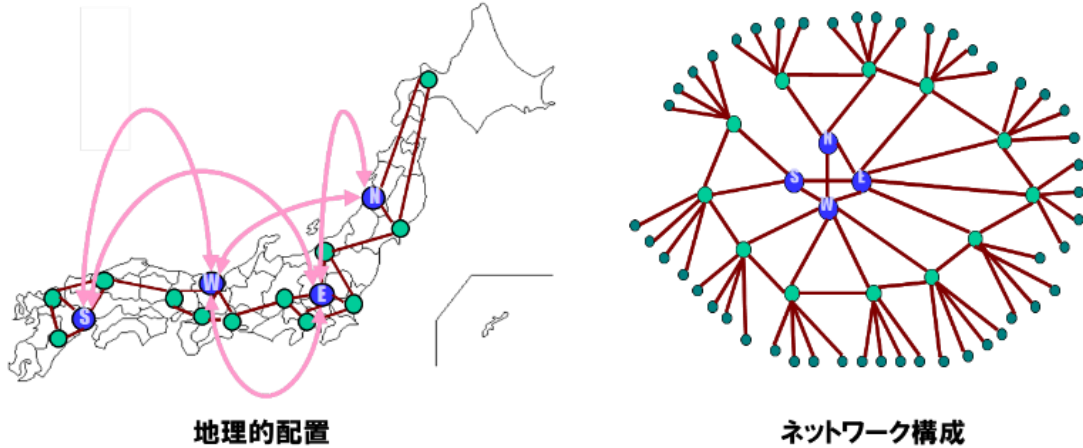


図 2-10. 日本全国の一般ユーザによるトラフィック交流を想定した評価に用いた網構成

<トラフィック交流>

EN 間のトラフィック交流モデルは、日本国内の電話網における通話量データ [30] を発着信比率に用い、式(2-1)のように EN に接続するユーザのアクセスネットワーク帯域からトラフィック量を算出した。このため、トラフィック量としては地域間の交流特性を表している。

$$B(EN_1, EN_2) = K \times \sum_{i=EN_2 \text{ 接続ユーザ}} S_i \times WD_2 \times \frac{WS_1 \times \sum_{j=EN_1 \text{ 接続ユーザ}} S_j}{\sum_{k=EN_2 \text{ 以外のEN接続ユーザ}} S_k} \quad (2-1)$$

- B (EN₁, EN₂): ED₁ から ED₂ へのトラフィック量
- S_i: user i のアクセスネットワーク帯域
- K: 重み係数
- WD₂: EN₂ への着信率
- WS₁: EN₁ からの発信率

<評価結果>

各 EN 間を交流するトラフィックを印加し，従来方式（Dijkstra）と提案方式における経路を計算し，リンク帯域の標準偏差と平均を評価した．

1. リンク帯域の標準偏差

リンク帯域の標準偏差はトラフィック量の分散度合いを表しており，少ない値であるほど効果的にネットワーク全体にトラフィックが分散されていることを表す．図 2-11 に各経路計算アルゴリズムが使用したリンク帯域の標準偏差を示す．

評価の結果，Dijkstra が使用されたリンク帯域の標準偏差は，帯域ボトルネック回避アルゴリズムの標準偏差と比べて，57.7Mbps のトラフィック量を印加した時点でリンク帯域の利用上限に達し，それ以上の経路計算/帯域割り当てが出来なくなった．一方，帯域ボトルネック回避アルゴリズムは，130Mbps のトラフィック量を印加した場合でもリンク帯域の標準偏差は小さく，十分なリンク帯域があった．この結果から，帯域ボトルネック回避アルゴリズムは Dijkstra よりもトラフィックを分散してネットワーク上に配置する経路を計算している．

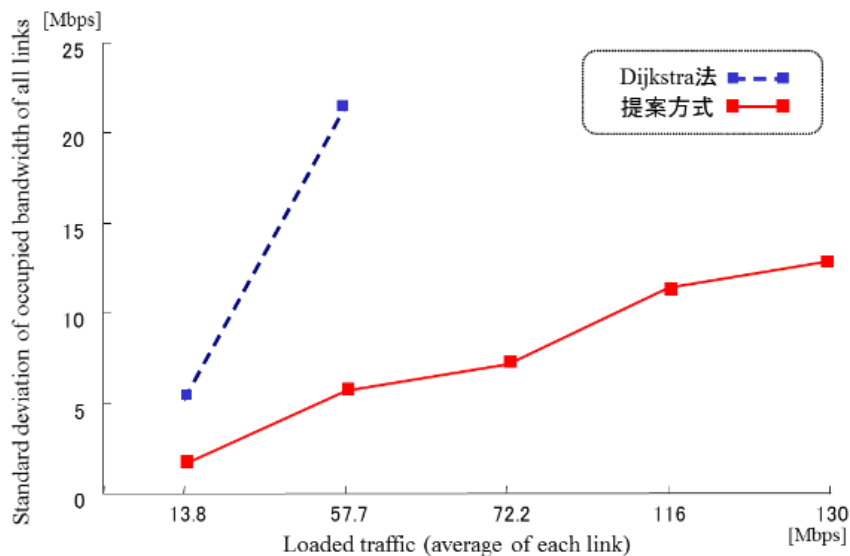


図 2-11. リンク帯域の標準偏差

2. リンク帯域の平均

リンク帯域はネットワーク全体のリンク帯域の利用効率の指標であり、少ない値であるほど、効果的にトラフィックがリンクに割り当てられていることを表す。図 2-12 に各経路計算アルゴリズムが使用したリンク帯域の平均を示す。

評価の結果、Dijkstra と帯域ボトルネック回避アルゴリズムが使用したリンク帯域の平均は、トラフィック量が 57.7Mbps に達するまでは同一であったが、それ以上のトラフィック量においては、リンク帯域の利用上限に達し、それ以上の経路計算/帯域割り当てが出来なくなった。一方、帯域ボトルネック回避アルゴリズムは、130Mbps のトラフィック量を印加した場合でも経路を計算することが出来た。この結果から、帯域ボトルネック回避アルゴリズムは Dijkstra と同等かそれ以上のリンク利用を行っている。

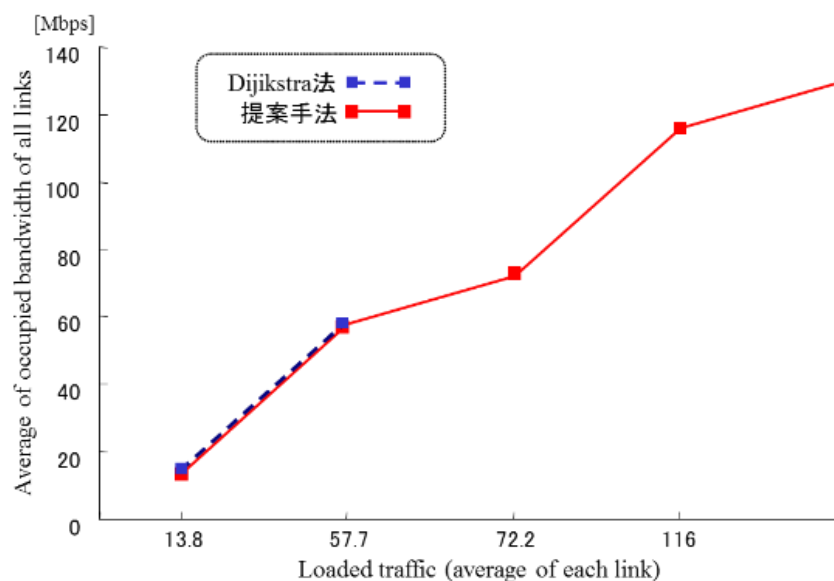


図 2-12. リンク帯域の標準偏差

2.8.2. データセンタへのトラフィックを想定した評価

日本全国から 1ヶ所のデータセンタに対するトラフィックを想定した評価を行った [31].

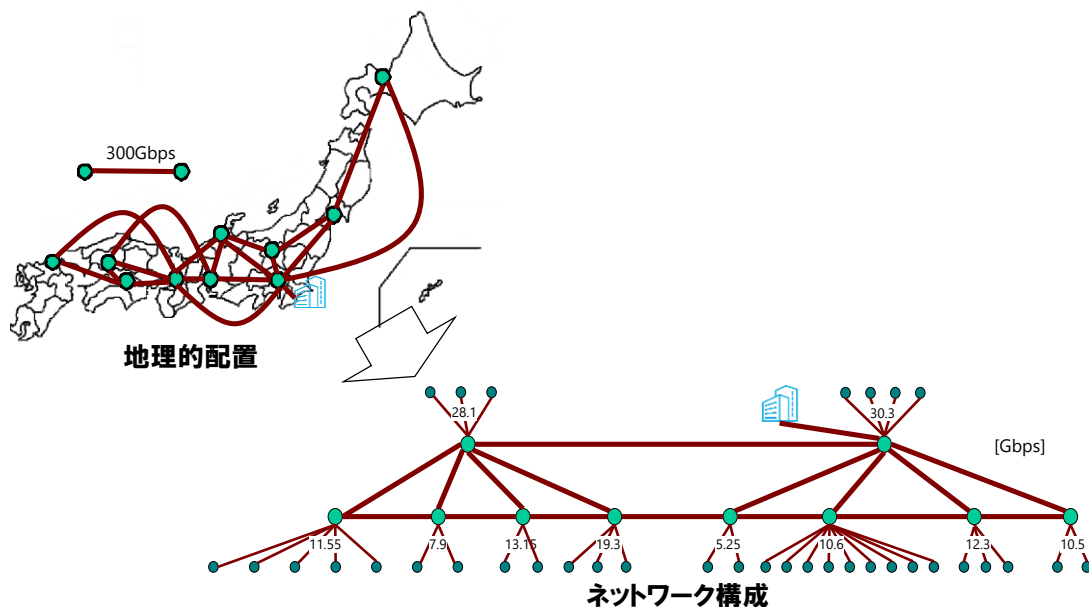


図 2-14. 2 軸 X ツリーネットワーク

<評価結果>

シミュレーション結果を表 2-1 に示す。トラフィックの単位は Gbps である

表 2-1. 日本全国からデータセンタへのトラフィックを想定した評価結果

	変形リング		2 軸 X ツリー	
	提案手法	Dijkstra	提案手法	Dijkstra
最大使用帯域	135	242	121	184
合計使用帯域	1153	795	991	564
平均使用帯域	50	35	30	17
標準偏差	56	65	36	39

Dijkstra 法と比較して提案手法の方が最大使用帯域や標準偏差が小さい。これは、提案手法がトラフィックをネットワーク内に分散しているためである。また、合計使用帯域や平均使用帯域が大きいのは、提案手法が経路を分散した結果、Dijkstra 法よりも遠回りの経路を選択しているためである。

シミュレーションの結果から、Dijkstra 法よりも提案手法の方が経路を分散し、ネットワーク内の各リンクにおける平均使用帯域を抑えることが確認できた。

2.9. 経路設定方法の比較評価

ネットワーク装置の経路制御機能を備えた MPLS エミュレータ (MPLS Workbench) を用いて、従来方式 (RSVP-TE) とサーバ方式による経路設定時間を、ホップ数および経路数の観点から比較評価した。

<検証環境>

検証に当たって用いた装置は表 2-2 以下の通り。

表 2-2. 検証に用いた端末一覧

No.	装置名	IP アドレス	概要
1	トラフィックエンジニアリングサーバ	200.1.1.210	開発したトラフィックエンジニアリングサーバ。
2	MPLS Workbench	200.1.1.1 ~ 40	MPLS ルータの制御機構から構成される MPLS エミュレータ。
3	XebeoMS	200.1.1.201-202	MPLS エミュレータ制御用の Management Server
4	telnet 用端末	200.1.1.xxx	トラフィックエンジニアリングサーバおよび XebeoMS にリモートログインする端末。

本検証に用いた端末のスペック一覧は以下の表 2-3 の通り。

表 2-3. 端末スペック一覧

No.	装置名	OS	CPU	メモリ
1	トラヒックエンジニアリングサーバ	RedHat Linux 7.2	Pentium4 3.06GHz	1GB
2	MPLS Workbench	XOS(R32821)	PentiumIII 866MHz	30MB
3	XebeoMS	FreeBSD 4.1.1-RELEASE	PentiumII 450MHz	64MB
4	telnet 用端末	WindowsXP Professional	PentiumIII 933MHz	256MB

検証環境のネットワーク構成は以下の図 2-15 の通り。

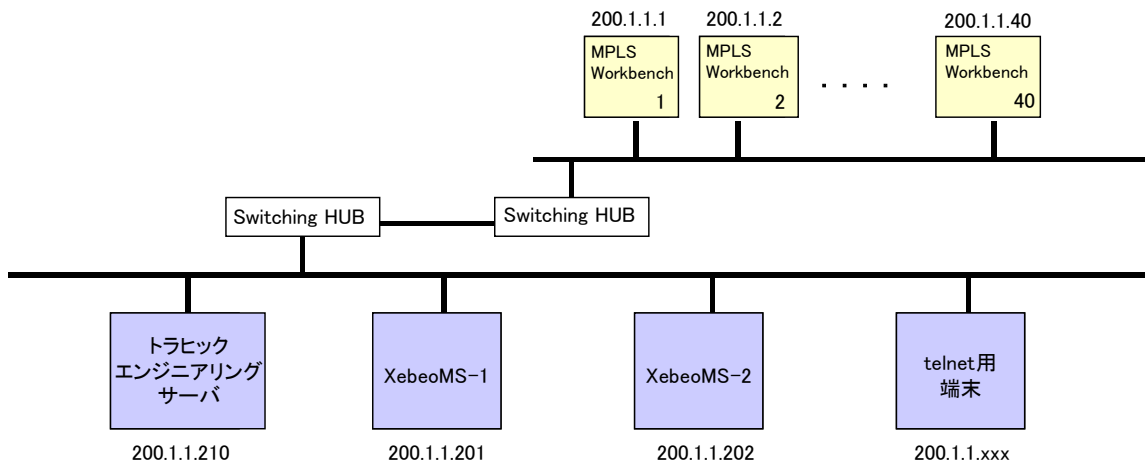


図 2-15. 検証環境のネットワーク構成

<検証項目>

以下に検証項目概要を示す。

検証項目 1 : ホップ数と設定時間

LSR を直列に接続した網を構成し、ホップ数をパラメータとして RSVP-TE とトラヒックエンジニアリングサーバによる 1 LSP 設定時間を評価する。

検証項目 2 : LSP 数と設定時間

LSR 2 台で構成された小規模な網において、LSP 数をパラメータとして RSVP-TE とトラヒックエンジニアリングサーバによる LSP 設定時間を評価する。

検証項目 3：故障回復時間

リンクダウンを意図的に発生させ、RSVP-TE とトラヒックエンジニアリングサーバ方式の故障回復時間を測定する。

検証項目 4：CPU 負荷

MPLS Workbench の CPU 負荷を計測し、LSP 設定に要する処理負荷について検証する。

上記の項目について、RSVP-TE 方式、トラヒックエンジニアリングサーバ方式による測定を行った。

2.9.1. 検証項目 1：ホップ数と設定時間

ホップ数の増加による LSP 設定時間への影響を明確化するため、MPLS ルータを直線的に配置した直列網を構成し、ホップ数をパラメータとして LSP 設定時間を評価する。

試験構成を図 2-16 に示す。

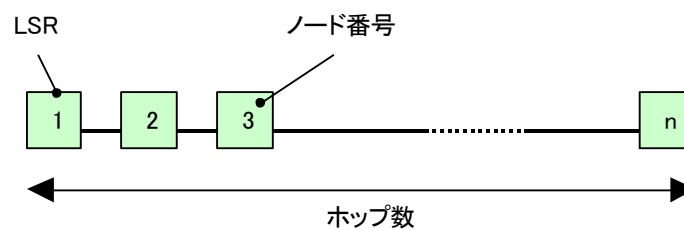


図 2-16. 検証項目 1 のトポロジ構成

検証項目 1 では、ホップ数をパラメータとして試験をおこなうため、ホップ数変更ごとにトポロジ構成を変更して試験をおこなう。最小 2 台、最大 40 台の MPLS ルータを直線的に接続した網構成とする。

設定時間は、トラヒックジェネレータ (SmartBits) とトラヒックモニタリング装

置 (Sniffer) を使用し, LSP 設定時間を測定する. LSP 設定前にアクセス網の外側から SmartBits によりデータを流し続け, 他方のアクセス網でこのデータが到着するのを Sniffer で観測し,最初にデータが到着した時間を導通時間とする.

以下に Sniffer と SmartBits を含めた網構成図を図 2-17 に示す.

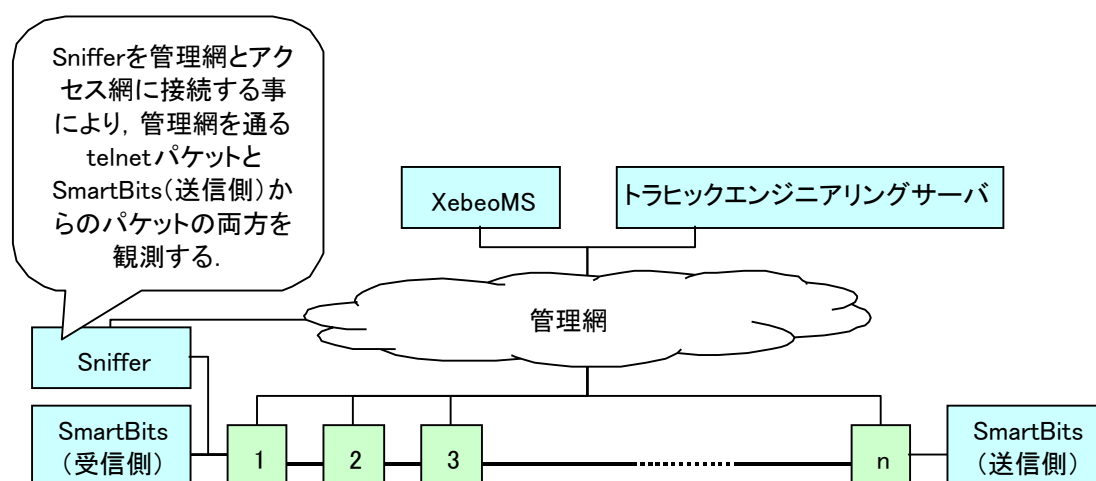


図 2-17. ネットワーク構成 (検証項目 1)

(1) 試験条件

- ・ パラメータとなるホップ数の最小は 1, 最大は 39 とする.
- ・ 最小ホップから最大ホップまで 5 ホップ間隔で試験を実施する
- ・ 1 試験あたり複数回の計測を行い, 精度の高い測定値を採取する.

(2) 試験手順

試験手順は以下のとおりである.

- (1) 事前環境整備 (MPLS Workbench への初期設定情報投入)
- (2) SmartBits によるデータ送信開始.
- (3) Sniffer によるキャプチャ開始.
- (4) 管理サーバからの LSP 設定開始.
- (5) Sniffer と SmartBits (受信側) により, SmartBits (送信側) から送信されて

きたデータを受信（パケットを受信することで LSP の導通性を確認する）。

(3) 試験結果

RSVP-TE 方式における試験結果を表 2-4 に示す。

表 2-4. RSVP-TE 方式による試験結果

イベント ホップ数	telnet 開始時間(ms)	commit 完了時間(ms)	導通時間(ms)
ホップ数 1	0	1013	2658
ホップ数 5	0	877	4221
ホップ数 10	0	877	6106
ホップ数 15	0	907	8182
ホップ数 20	0	878	10053
ホップ数 25	0	891	12180
ホップ数 30	0	709	13930
ホップ数 35	0	797	16019
ホップ数 39	0	815	17647

*試験結果は、試験計測回数 10 回の平均値。

表 2-4 におけるイベントの意味は以下のとおりである。

・ telnet 開始時間

XebeoMS から MPLS ルータへの telnet 開始時間。本試験では、双方向に LSP を設定するため、両端の MPLS Workbench に対して telnet 接続をおこなう。このうち、早い方の telnet 開始時間を 0 秒（基準時間）とする。

・ commit 完了時間

XebeoMS から MPLS ルータへの commit コマンド完了時間。両端の MPLS Workbench のうち、遅い方の値を採用する。

・ 導通時間

SmartBits(送信側)が送信したデータを Sniffer が最初に観測した時間とする。

表 2-4 をグラフ化したものを以下図 2-18 に示す。

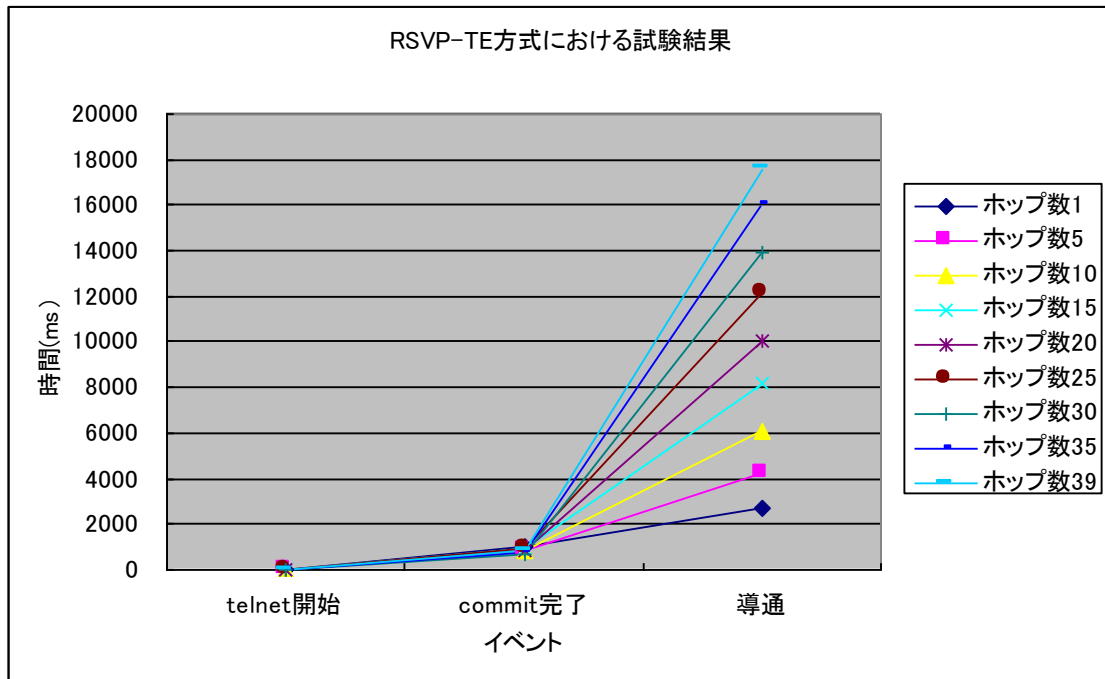


図 2-18. RSVP-TE 方式における試験結果 (検証項目 1)

試験により以下の結果を得られた。

(1)telnet 開始から commit 完了までは一定

RSVP-TE 方式では, XebeoMS から MPLS Workbench に投入するコマンド数, 制御する MPLS ルータ数はホップ数に依存することなく一定であるため, telnet 開始から commit 完了までは一定であった。

(2)commit 完了から 導通まではホップ数の増加に比例して上昇

ホップ数の増加にともない 導通までの時間が比例した。

トラヒックエンジニアリングサーバにおける試験結果を表 2-5 に示す。

表 2-5. トラヒックエンジニアリングサーバ方式による試験結果

イベント ホップ数	telnet 開始 時間(ms)	ftp 開始時間 (ms)	導通時間(ms)
ホップ数 1	0	1401	3148
ホップ数 5	0	1327	3178
ホップ数 10	0	1298	3375
ホップ数 15	0	1334	4001
ホップ数 20	0	1321	4207
ホップ数 25	0	1314	4527
ホップ数 30	0	1342	4735
ホップ数 35	0	1349	4608
ホップ数 39	0	1318	5275

*試験結果は、試験計測回数 3 回の平均値。

上記の表をグラフ化したものを図 2-19 に示す。

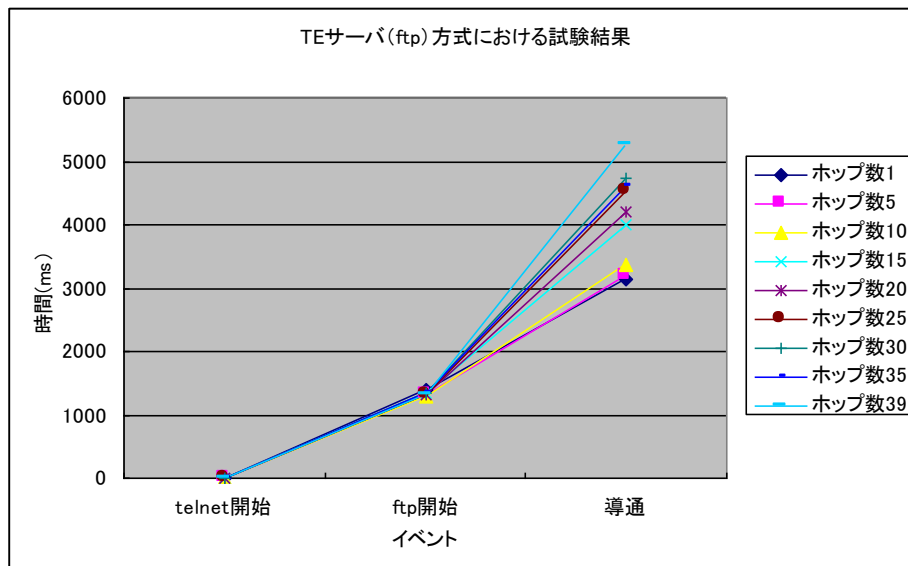


図 2-19. トラヒックエンジニアリングサーバ方式における試験結果 (検証項目 1)

試験により以下の結果を得られた。

(1) telnet 開始から ftp 開始までは一定

ホップ数の増加とともに、トラヒックエンジニアリングサーバが制御する MPLS Workbench 数も増加するため、当初は telnet 開始から ftp 開始までの時間がホップ数に比例して増加すると考えていたが、実際には全試験をとおしてほぼ同じ値であった。これは、telnet 開始から ftp 開始までの処理が非常に軽く、ノード数が増えることによる並列処理に影響を与えないためと考えられる。

(2) ホップ数に従い ftp 開始から導通までの時間が上昇する傾向にある

ホップ数の上昇に従い ftp 開始から導通までの時間が上昇する傾向にある。Sniffer で取得したキャプチャファイルを解析した結果、並列処理により ftp 転送にかかる時間が増えていることがわかったが、試験手順上、全 MPLS Workbench が同一 ftp サーバに接続するため、これによる ftp 接続の負荷が原因と考えられる。

<試験結果の考察>

RSVP-TE およびトラヒックエンジニアリングサーバによるホップ数に対する設定時間の試験結果をまとめた図を図 2-20 に示す。

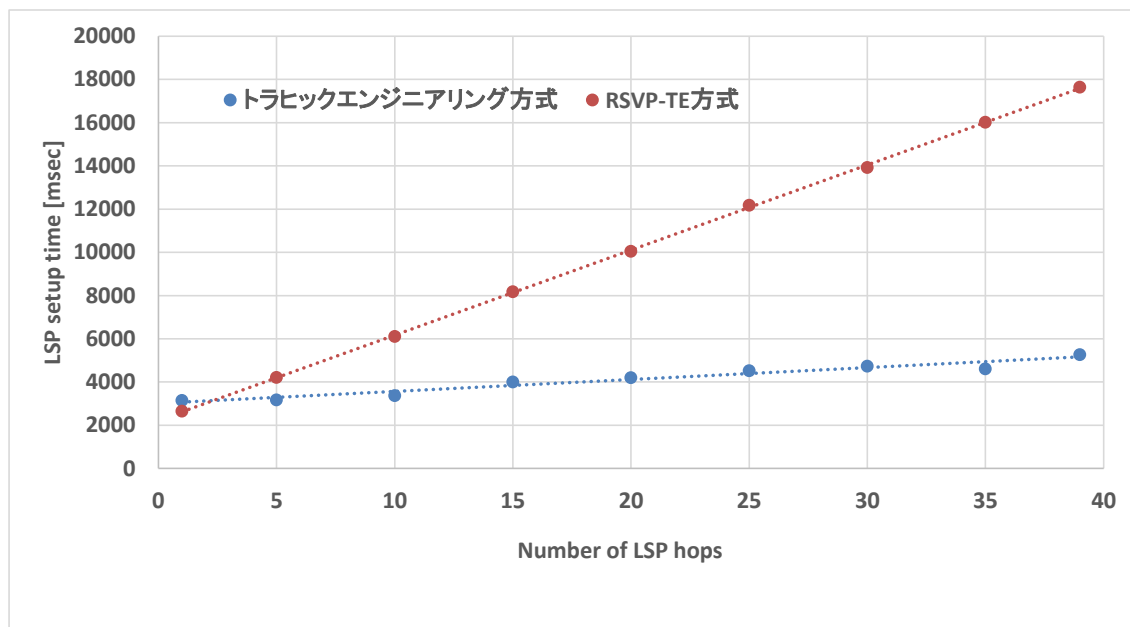


図 2-20. ホップ数に対する設定時間

試験結果より、トラヒックエンジニアリングサーバ方式は、RSVP-TE方式と比較して、ホップ数増加の影響を受けにくいことが分かった。ホップ数が増えるにつれて、RSVP-TE方式に比べて、トラヒックエンジニアリングサーバ方式の設定時間が短いことがわかる。

2.9.2. 検証項目 2 : LSP 数と設定時間

設定する LSP 数の増加による全 LSP 設定時間への影響を明確化するために、RSVP-TE方式とトラヒックエンジニアリングサーバ方式の双方で検証を行い、比較検討する。試験構成を図 2-21 に示す。

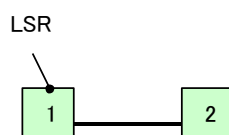


図 2-21. 検証項目 2 のトポロジ構成

LSP 設定時間の測定方法は、検証項目 1 と同様である。

以下に Sniffer と SmartBits を含めた網構成図を図 2-22 に示す。

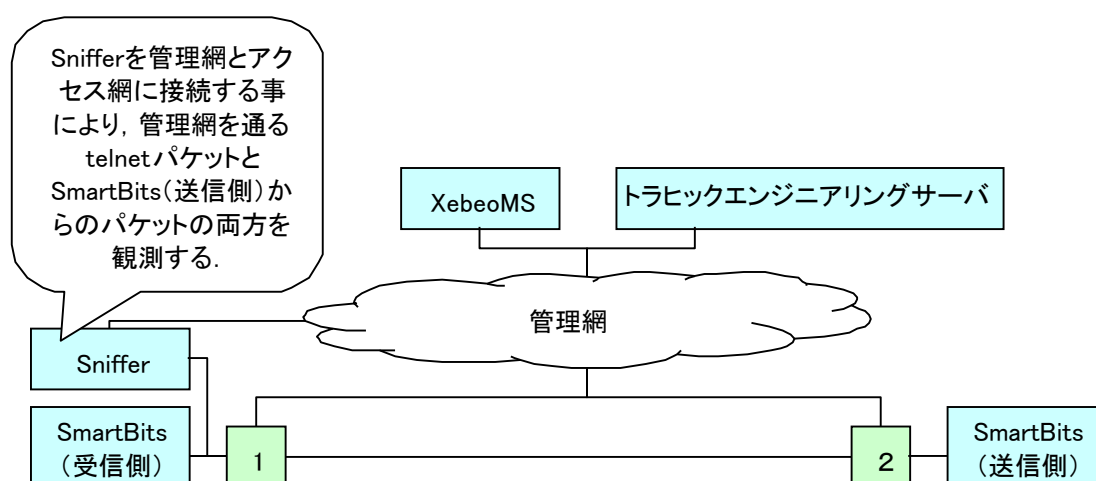


図 2-22. ネットワーク構成 (検証項目 2)

(1) 試験条件

- ・ パラメータとなる設定 LSP 数の最小は 1, 最大は 496 とする.
- ・ 最小 LSP 数から最大 LSP までは 30LSP 間隔で試験を実施する.
- ・ 1 試験あたり複数回の計測を行い, 精度の高い測定値を採取する.

(2) 試験手順

試験手順は検証項目 1 と同様である.

(3) 試験結果

RSVP-TE 方式における試験結果を示す (表 2-6).

表 2-6 におけるイベントの意味は以下のとおりである.

- ・ telnet 開始時間, commit 完了時間, 導通時間について
試験項目 1 と同一である.

- ・ rsvp-msg 送信開始時間

RSVP Path Message パケット (以下, rsvp-msg と表記) の送信開始時間. 図 2-22 の MPLS Workbench2 から MPLS Workbench1 へ流れる複数の rsvp-msg のうちの, 最初の時間を採用する.

- ・ rsvp-resv 受信完了時間

RSVP Reservation パケット (以下, rsvp-resv と表記) の受信完了時間. rsvp-resv パケットの受信完了時間を採用する.

表 2-6. RSVP-TE 方式による試験結果

イベント LSP 数	telnet 開始 時間(ms)	commit 完了 時間(ms)	rsvp-msg 送 信開始時間 (ms)	rsvp-resv 受信 完了時間(ms)	導通時間(ms)
LSP 数 1	0	772	2055	2253	2553
LSP 数 30	0	3689	4733	5103	5403
LSP 数 60	0	6789	7752	8236	8537
LSP 数 90	0	9809	10863	11646	11946
LSP 数 120	0	12829	13799	15072	15372
LSP 数 150	0	1583	16807	18307	18606
LSP 数 180	0	18840	19847	21568	21862
LSP 数 210	0	21870	22826	25088	25389
LSP 数 240	0	24861	25716	28303	28603
LSP 数 270	0	27909	28899	31646	31866
LSP 数 300	0	30980	31928	34814	35113
LSP 数 330	0	33528	34992	38393	38693
LSP 数 360	0	36087	38295	41838	42176
LSP 数 390	0	39074	40998	44810	45109
LSP 数 420	0	42021	44022	48343	48643
LSP 数 450	0	45065	47031	51594	51894
LSP 数 480	0	48133	49532	54804	55104
LSP 数 496	0	50148	51657	56510	56850

*試験結果は、試験計測回数 10 回の平均値 (LSP 数 1~300 の試験)。

*試験結果は、試験計測回数 5 回の平均値 (LSP 数 330~496 の試験)。

表 2-6 をグラフ化したものを図 2-23 に示す。

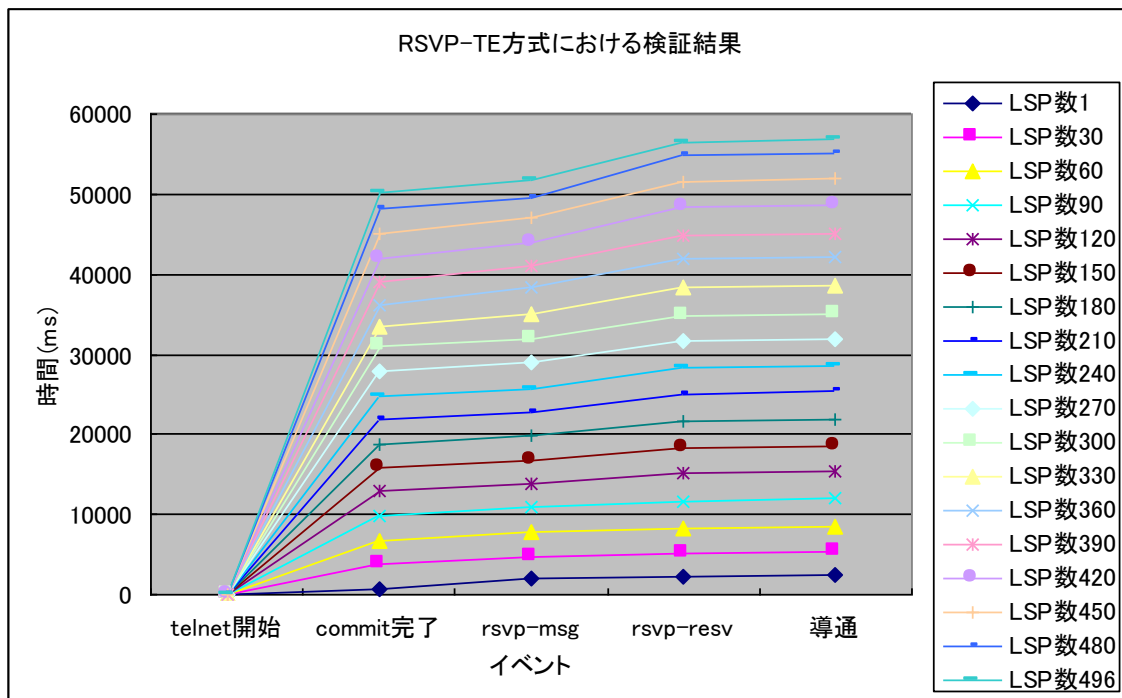


図 2-23. RSVP-TE 方式における試験結果 (検証項目 2)

試験により以下の結果を得られた。

(1)telnet 開始から commit 完了までは LSP 設定数に比例して上昇

試験で採取したパケットを解析した結果、これは XebeoMS から MPLS Workbench への、telnet による LSP 設定コマンド数に依存していることがわかった。RSVP-TE 方式において LSP を 1 本設定する場合、XebeoMS から Ingress となる MPLS Workbench に対して 1 コマンド投入する。よって LSP 設定数が 496 の場合、投入するコマンド数は 496 コマンドとなる。パケット解析の結果、telnet ログインに必要な時間が約 500ms、XebeoMS による 1 コマンド実行時間が約 100ms であることが分かり、実行処理時間に比例していることが分かった。

(2)commit 完了から rsvp-msg 送信開始までは一定

commit 完了から約 1000ms 後に rsvp-msg の送信を始めていることが分かった。

(3)rsvp-msg 送信開始から rsvp-resv 受信完了までは LSP 数に比例して上昇

試験結果より、設定する LSP 数に比例して rsvp パケットの交換に必要な時間が上昇していることが分かった。

(4)rsvp-resv から導通までは一定

rsvp-resv 受信完了から導通までは、全試験において一律約 300ms であることが分かった。

トラヒックエンジニアリングサーバにおける試験結果を表 2-7 に示す。

試験により以下の結果を得られた。

(1)ftp 開始より ftp 完了までの時間は LSP 数に比例して上昇

LSP 数が増える事により、MPLS Workbench が ftp による取得するコンフィグレーションファイルサイズが増えるため、それに伴ってファイル転送時間が増えていると考えられる。

(2)ftp 完了時間から導通までの時間も LSP 数に比例して上昇

これも(1)と同様に、受信したコンフィグレーションファイルのサイズの増大に比例して、受信したコンフィグレーションファイルの構文解析、設定にかかる時間も増加していると考えられる。

表 2-7. トラヒックエンジニアリングサーバ方式による試験結果

イベント LSP 数	telnet 開始 (ms)	ftp 開始 (ms)	ftp 完了 (ms)	導通 (ms)
LSP 数 1	0	1408	1496	3154
LSP 数 30	0	1413	1913	3401
LSP 数 60	0	1409	2234	3848
LSP 数 90	0	1414	2515	4274
LSP 数 120	0	1415	2808	5255
LSP 数 150	0	1409	3093	5638
LSP 数 180	0	1408	3342	5952
LSP 数 210	0	1416	3573	7101
LSP 数 240	0	1411	3965	7461
LSP 数 270	0	1409	4229	7879
LSP 数 300	0	1414	4597	9021
LSP 数 330	0	1409	4830	9204
LSP 数 360	0	1408	5129	9596
LSP 数 390	0	1411	5464	9998
LSP 数 420	0	1416	5778	11209
LSP 数 450	0	1406	6068	11524
LSP 数 480	0	1414	6360	11955
LSP 数 496	0	1407	6488	12067

*試験結果は、試験計測回数 3 回の平均値.

表 2-7 をグラフ化したものを図 2-24 に示す。

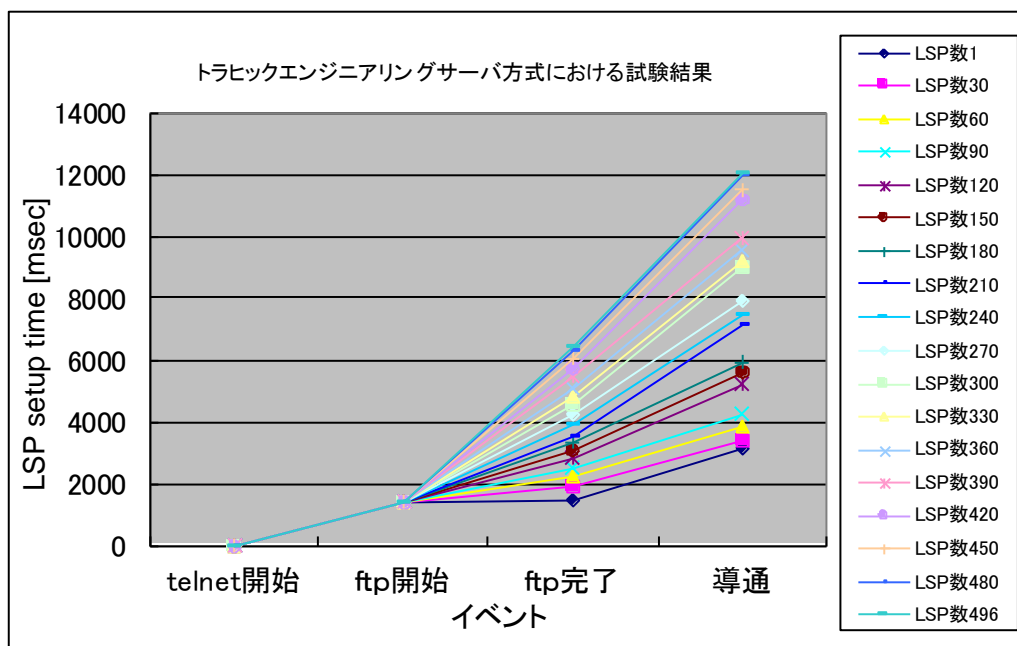


図 2-24. トラヒックエンジニアリングサーバ方式における試験結果（検証項目 2）

<試験結果の考察>

RSVP-TE およびトラヒックエンジニアリングサーバによる LSP 導通時間の試験結果を図 2-25 に示す。

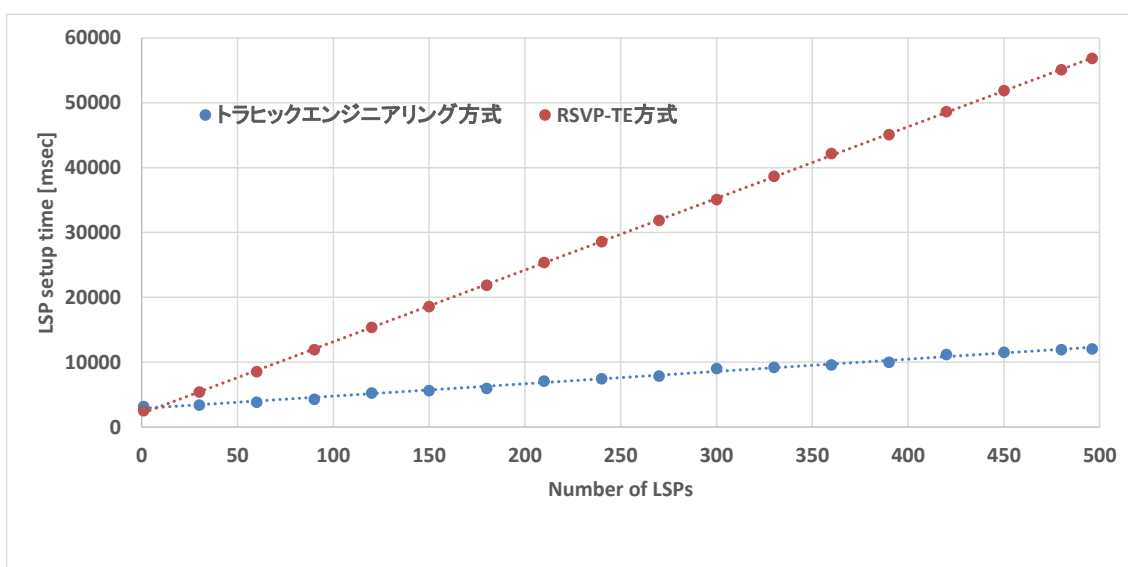


図 2-25. 経路数に対する設定時間

上記の結果より，RSVP-TE 方式よりトラヒックエンジニアリングサーバ方式の設定時間が短いことが分かった．サーバ方式は，経路上の各ネットワーク装置へ並列に設定を行うため，順に設定を行う従来方式よりも経路設定時間が短いことが分かった．

2.9.3. 検証項目 3 : 故障回復時間

RSVP-TE 方式，トラヒックエンジニアリングサーバ方式における故障迂回時間を測定する．リストラクションによる迂回，プロテクションによる迂回，故障させるリンク箇所の変更など，さまざまな条件下で試験をおこなうことで，各々の特性について考察する．

試験構成を図 2-26 に示す．

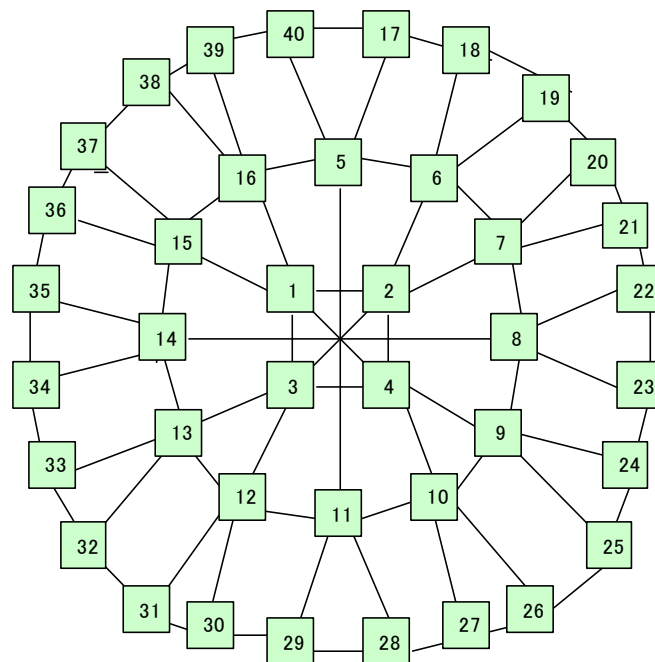


図 2-26. 検証項目 3 のトポロジ構成

図 2-26 に示す正円網を使用した。図を簡略化しているため、28 台の MPLS Workbench により構成された図となっているが、実際には 40 台の MPLS Workbench により構成された正円網を使用した。

検証項目 3 では、SmartBits と Sniffer を使用し、故障時のパケット損失時間を測定する。このパケット損失時間を故障時における迂回時間と判断する。

Sniffer と SmartBits を含めた網構成図を図 2-27 に示す。

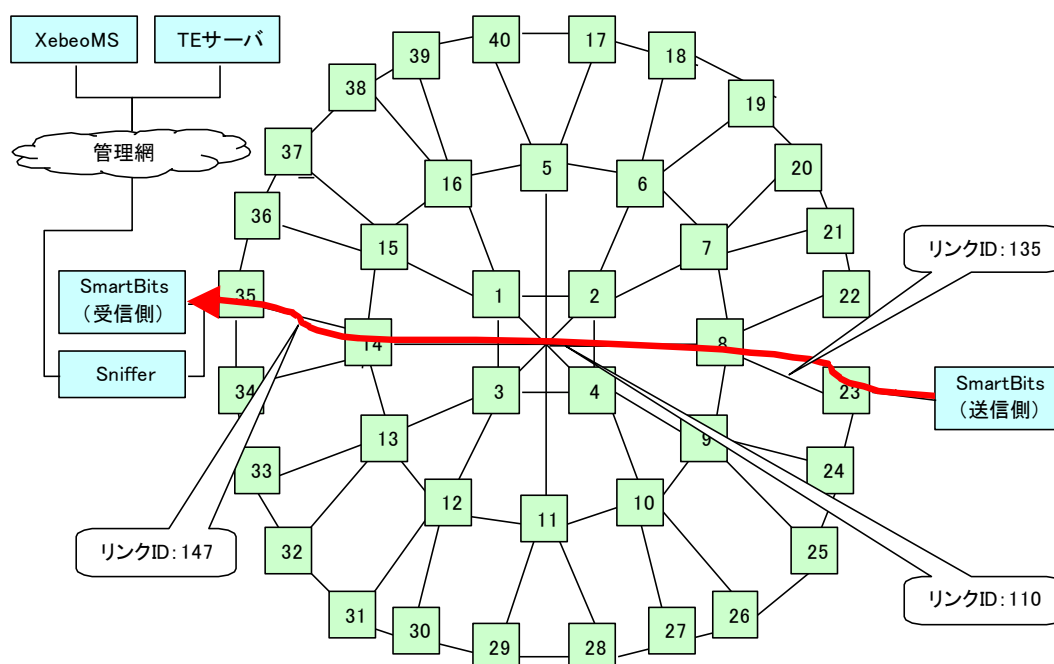


図 2-27. ネットワーク構成 (検証項目 3)

(1) 試験条件

- ・ リストレーションによる迂回，プロテクションによる迂回の両方について実施する。
- ・ 故障箇所は 1 箇所限定せず，複数の故障箇所を用意する。
- ・ 1 試験あたり複数回の計測を行い，精度の高い測定値を採取する。

(2) 試験手順

試験手順は以下の通り。

- (1) 事前環境整備 (MPLS Workbench への初期設定情報投入)。
- (2) 管理サーバより、事前に LSP を設定。
- (3) SmartBits によるデータ送信開始。
- (4) Sniffer によるキャプチャ開始。
- (5) 特定のリンクケーブルを MPLS Workbench のインタフェースから抜き、意図的にリンクダウンを発生。
- (6) Sniffer により、データの再受信を確認 (このデータ受信により、故障迂回が終了したと判断)。

(3) 試験結果

今回の試験では、4 パタンについて検証をおこなった。試験概要、実施試験回数について表 2-8 に示す。試験結果を表 2-9 に示す。表 2-9 をグラフ化したものを図 2-28 に示す。試験により以下の結果を得られた。

トラヒックエンジニアリングサーバ方式は、最小経路、中間経路の時間差がなく、安定した時間で故障切り替えを行うことが出来る。あらかじめ用意したセカンダリパスに切り替えるため、設定変更が必要なノード数、設定量が経路によって変わらないことが理由であると考えられる。

一方、RSVP-TE 方式のリストレーションは故障切り替えに要する時間がリンク ID により異なり、トラヒックエンジニアリングサーバ方式よりも長い。リンク 147, 110, 135 の迂回後のホップ数はそれぞれ 5, 8, 5 であり、ホップ数が長い分 LSP 導通時間に時間が必要であると考えられる。RSVP-TE 方式のプロテクションでは、リンク 147 の迂回についてトラヒックエンジニアリングサーバ方式よりも長い時間がかかっている。リンク 147 については受信側に近いリンク設定に要する時間が大きいことが原因と考えられる。

表 2-8. 試験概要一覧（検証項目 3）

試験方式	試験概要	試験回数
RSVP-TE 方式 (プロテクション)	セカンダリ経路をあらかじめ設定し，故障検出時に RSVP-TE プロトコルにより自律的に故障迂回をおこなう方式.	10
RSVP-TE 方式 (リストレーション)	故障検出時に RSVP-TE により代替経路の再計算をおこない，動的に経路を切り替える方式.	10
トラヒックエンジニアリング方式 (プロテクション：最小経路)	故障想定箇所を局所的に迂回するような最小迂回経路をセカンダリパスとして設定し，トラヒックエンジニアリングサーバが当該ルータに直接セカンダリパスへの切り替えを指示する方式.	5
トラヒックエンジニアリング方式 (プロテクション：中間経路)	遠回りとなる経路をセカンダリパスとして設定し，トラヒックエンジニアリングサーバが当該ルータに直接セカンダリパスへの切り替えを指示する方式.	5

表 2-9. 検証結果（方式別故障迂回時間一覧）

リンク ID	故障迂回時間 (ms)			
	RSVP-TE 方式 (プロテクション)	RSVP-TE 方式 (リストレーション)	トラヒックエンジニアリング方式 (プロテクション，最小経路)	トラヒックエンジニアリング方式 (プロテクション，中間経路)
147	1958	3946	1610	1539
110	972	4845	1638	1476
135	1109	3744	1641	1568

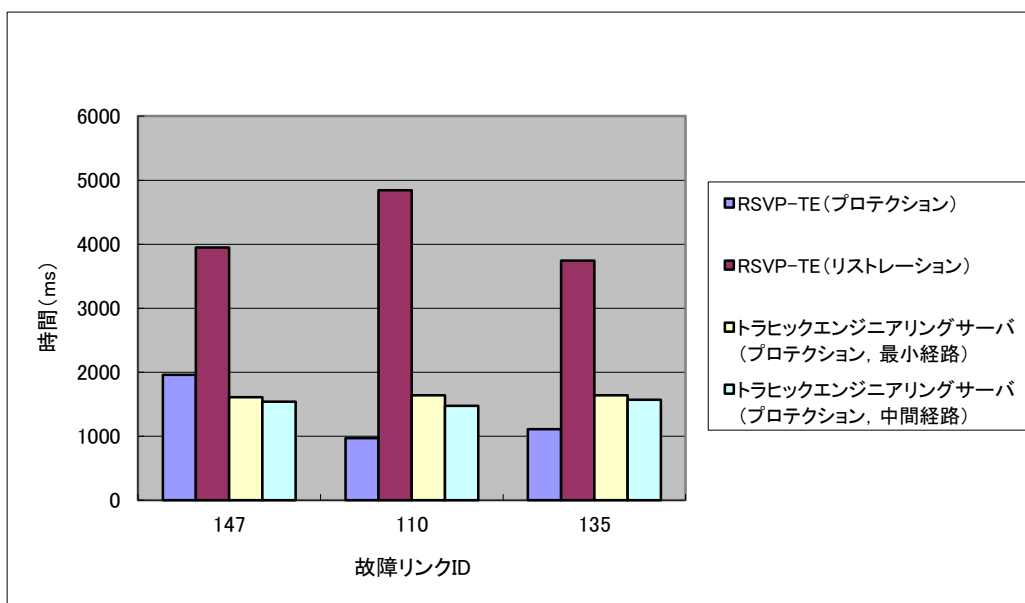


図 2-28. 故障回復時間の試験結果 (検証項目 3)

2.9.4. 検証項目 4 : CPU 負荷

RSVP-TE, トラヒックエンジニアリングサーバのそれぞれにおいて, LSP 設定時の MPLS Workbench の CPU 使用率を測定し, MPLS ネットワークの LPS 設定における CPU 負荷について検証する. CPU 使用率測定対象プロセス一覧を表 2-10 に示す.

表 2-10. CPU 使用率測定対象プロセス一覧

対象プロセス	概要
xsh	telnet ログインセッションごとに起動するプロセス.
cfgdb	コンフィグレーションデータなどを MPLS Workbench のバッファ領域に書き込む際の処理プロセス.
netctl	OSPF, LDP, RSVP-TE などのプロトコル実装プロセス.
hwsim	ソフトウェア処理によるパケットフォワーディングを実現するプロセス. パケットフォワーディングのエミュレーションプロセス.

測定結果を図 2-29, 図 2-30 に示す。CPU 負荷について、RSVP-TE 方式, トラヒックエンジニアリングサーバ方式ともに, hwsim 以外の CPU 負荷が非常に小さい事がわかった。hwsim による処理は本検証項目で使用した MPLS Workbench 独自の処理であり, エミュレータでは無い実際の MPLS ルータでは行われたい処理であるため, この値が高いことに問題はない。一方, その他のプロセスの CPU 使用率は非常に小さく, RSVP-TE 方式, トラヒックエンジニアリングサーバ方式ともに 1%程度の CPU 使用率である。このことから, 双方の方式において MPLS Workbench の CPU への負荷はほとんどないと考えられる。本検証は 2000 年代に行っており, 現在は MPLS ルータの性能が向上しているが, LSP 設定処理における CPU 負荷への影響は少ないことから, CPU 処理性能の向上による処理時間への影響は少なく, 当時の検証結果は現在でも同様になると想定される。

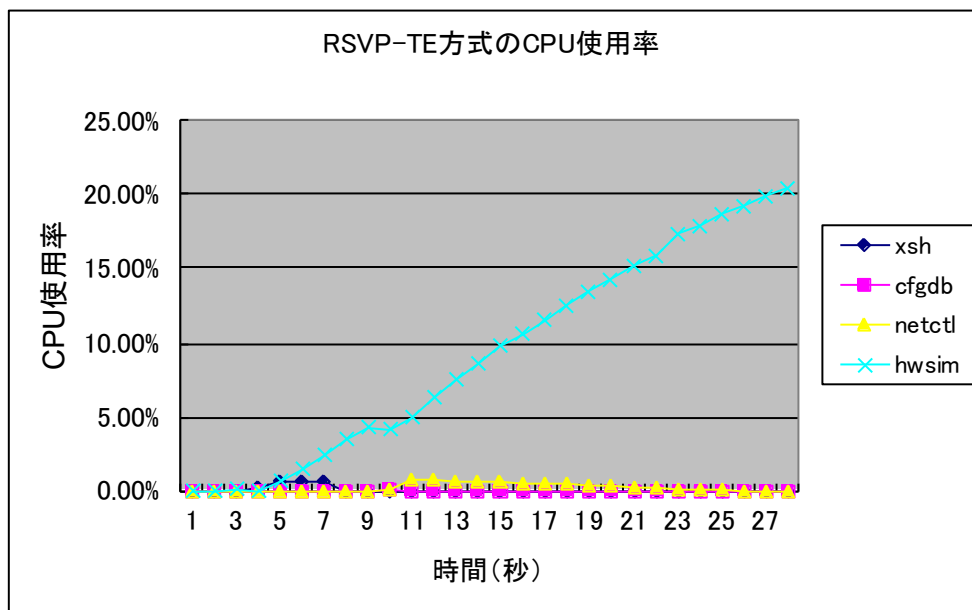


図 2-29. RSVP-TE 方式における MPLS Workbench の CPU 使用率 (検証項目 4)

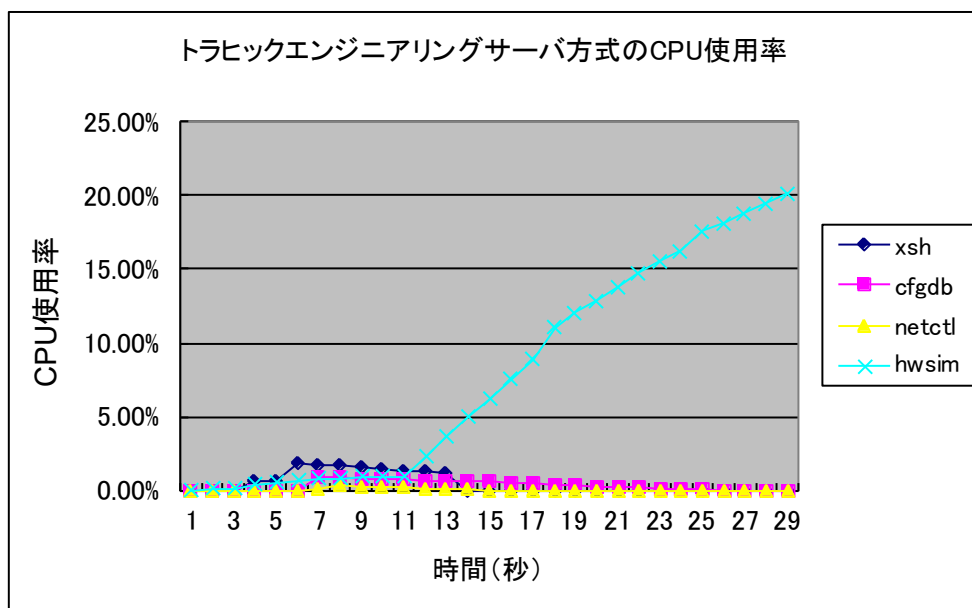


図 2-30. トラヒックエンジニアリングサーバ方式における MPLS Workbench の CPU 使用率 (検証項目 4)

2.10. まとめ

本章では，サーバ型トラヒックエンジニアリング方式を提案した．ネットワーク全体の帯域とトラヒック配置を一元管理し，効率的な計算アルゴリズムを採用することにより，従来の分散方式に比べ，トラヒック収容効率の向上，経路設定時間の短縮化を可能とした．

経路計算方式をシミュレーションにより評価した結果，提案する帯域ボトルネックを回避する経路計算アルゴリズムは，従来の計算アルゴリズムよりもリンクに割り当てた帯域の標準偏差，平均が少ないため，ネットワーク帯域を効率的に使用できた．経路設定時間をエミュレータにより評価した結果，サーバ型による経路設定時間は，分散型の経路設定時間よりも短く，故障回復時間も短く，ホップ数および経路数にあまり依存しない経路が多い大規模ネットワークにおいても高速な経路設定が可能であった．

また，本検証は 2000 年代に行ったものであり，現在は MPLS ルータの性能が向上しているが，LSP 設定処理における CPU 負荷への影響は少ないことから，CPU 処理性能の向上による処理時間への影響は少なく，当時の検証結果は現在でも同様になると想定される。

これらにより，提案するサーバ型トラフィックエンジニアリング方式は，現在においても大規模なネットワークにおいてネットワークリソースを効率よく制御するために有効な方法であり，通信キャリアのバックボーンネットワークにおいて有効に活用できると考える。

3. マルチキャスト経路管理によるリソース利用効率化

3.1. 背景

多数のユーザに対し効率的に情報を配信する技術として、IP マルチキャストが活用されている。映像や音声のコーディング技術の発展と端末でのエンコード・デコード処理技術の進展により、テレビ会議やインターネット放送など、多対多や1対多のリアルタイム通信をIPネットワーク上で行うことが一般的になっている。ユニキャスト通信を用いて多地点へトラフィックを配信する場合、サーバもしくは端末にて配信先数分のトラフィックを生成し、ネットワーク上の各転送ノードがトラフィックを転送しなければならない。利用数が増えるにつれてバックボーンネットワークの帯域を圧迫する。一方、マルチキャスト通信を用いて多地点へトラフィックを配信する場合は、1つのトラフィックを受信端末になるべく近い転送ノードにてコピーすることで、多数の端末へ配信を行う技術であるため、送信サーバ・端末や転送ノードに負荷をかけず、スケーラビリティが高いメリットがある。

しかし、標準化されたIPマルチキャストアーキテクチャは、マルチキャストグループに対して自由にデータを送信/受信できるため、ネットワークはユーザに対し、無関係の第三者による送信や受信を防いだマルチキャストサービスを提供できない。このため、スケーラブルに他地点配信を行うことの出来るマルチキャストをビジネスに用いることが出来ない問題がある。

3.2. 従来方式の課題

現在使用されている IP マルチキャストルーティングプロトコルは、受信要求を行う IGMP[32]とマルチキャストパケットのルーティングを行う PIM-SM[33]等がある。これらのプロトコルにより、各転送ノードが受信要求情報とマルチキャストルーティング情報を交換し、分散的にマルチキャストパケットの転送を行う (図 3-1)。

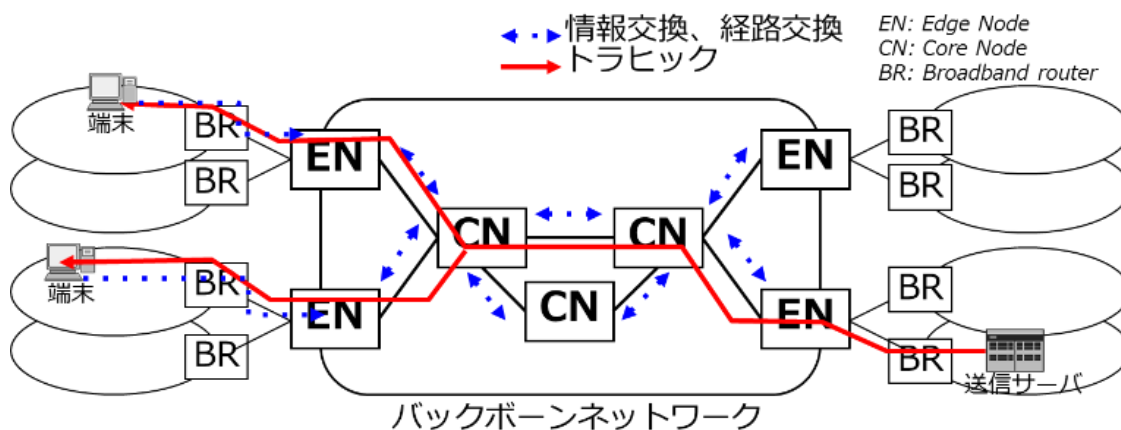


図 3-1. 分散型マルチキャストシステム

しかしながら、いずれのプロトコルにも、ユーザを認証する機構は存在せず、任意のユーザが任意のマルチキャストグループに対し送信/受信が可能である。しかし、ユーザがビジネスとしてマルチキャスト通信を利用するには、マルチキャストグループのジャック (不正送信) や、盗み聞き (不正受信) が発生しないマルチキャスト通信が必要となる。ゆえに、送信者保護 (送信者を特定し第三者からの送信を防ぐ)、受信者制限 (受信者を特定し第三者の受信を防ぐ) を提供するマルチキャストサービスが必要となる (図 3-2)。

		受信者	
		特定	不特定
送信者	特定	テレビ会議	放送
	不特定	提案サービス領域	現在のIPマルチキャスト領域

図 3-2. マルチキャスト領域の分類

図 3-2 にユーザの種別（送信者／受信者，特定／不特定）によりマルチキャストサービス領域を分類し，マルチキャストアプリケーションをビジネス用途で利用するために要求される領域を示す．要求領域をカバーする新たなマルチキャストサービスの提供が必要である．

IP マルチキャストにおいて送信者保護/受信者制限を行う方法として，

(1) マルチキャストグループの送受信者をネットワークに登録し特定のユーザ間でマルチキャスト通信を行うよう経路制御する方法

(2) パケットを暗号化し特定のユーザのみが解読する方法

が考えられる．(2)の方法は，ユーザにとっては端末の処理能力が要求され，一方，ネットワークにとっては不要トラフィックによる混雑は解消されないという問題がある．そこで本研究では(1)の方式を検討する．

3.3. 提案方式

送信者保護／受信者制限を行うには，(1) 送信者保護／受信者制限に必要な情報の把握，と，この情報に基づいた(2) 受信要求の認証，不正送信の防止，の2つの機

能を実装する必要がある。本研究では、ユーザの認証を行い、マルチキャスト配信を管理するサーバ型マルチキャスト管理システムを提案した[34][35]。本研究が対象とするネットワークは、自律分散型でルーチングを行うインターネットなどのネットワークではなく、サーバ型で集中管理を行うキャリアのバックボーンネットワークであり、マルチキャスト通信を用いた他地点配信サービスをビジネス展開する場合を対象としている。提案システムの構成図を図 3-3 に示す。

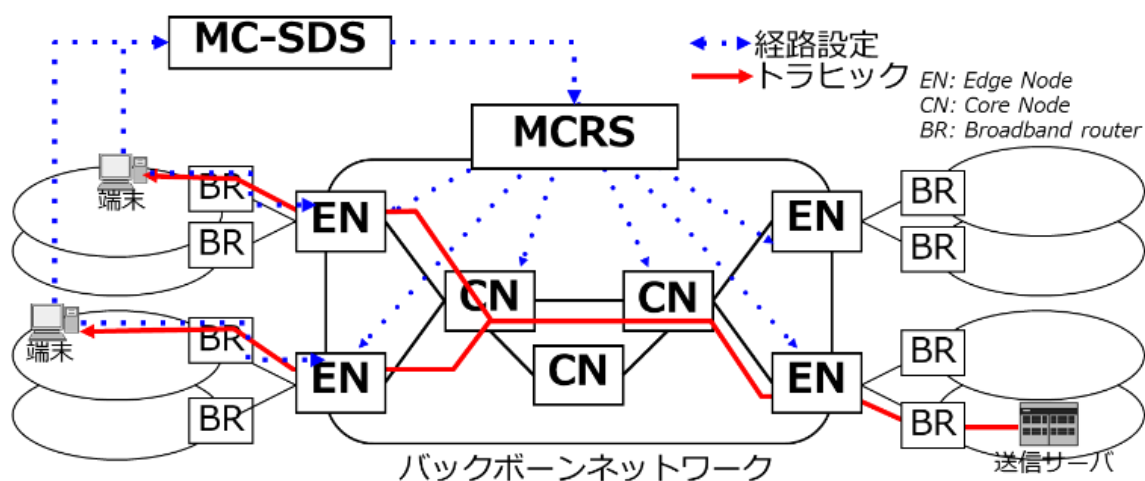


図 3-3. サーバ型マルチキャストシステム

マルチキャストルートサーバ (MCRS) が、標準的なマルチキャストプロトコルである IGMP を終端する機能を持ち、ユーザ認証、トラフィック情報、経路情報を管理し、ユーザ契約やリソース利用状況に基づいたバックボーンネットワークの経路制御を実施する。MCRS が IGMP による受信要求に含まれる IP アドレスや回線 ID によってユーザを識別し、許可されたユーザのみがマルチキャストグループへ送信できるように制御を行う。これにより、未許可のユーザがマルチキャストグループへ配信することを防止できる。また、ユーザ契約に基づく送信可否の管理や課金が可能となる。マル

マルチキャストサービスディレクトリサーバ (MC-SDS) は、ユーザの要求を受け付けて、MCRS と連携し、マルチキャストグループと送受信可能なユーザの管理を行うことが可能となる。既存のプロトコルを改変することなく、ユーザ契約やリソース利用状況に基づいたマルチキャスト経路配信管理を一元的に行うことが可能となる。

具体的な動作の流れを図 3-4 に示す。

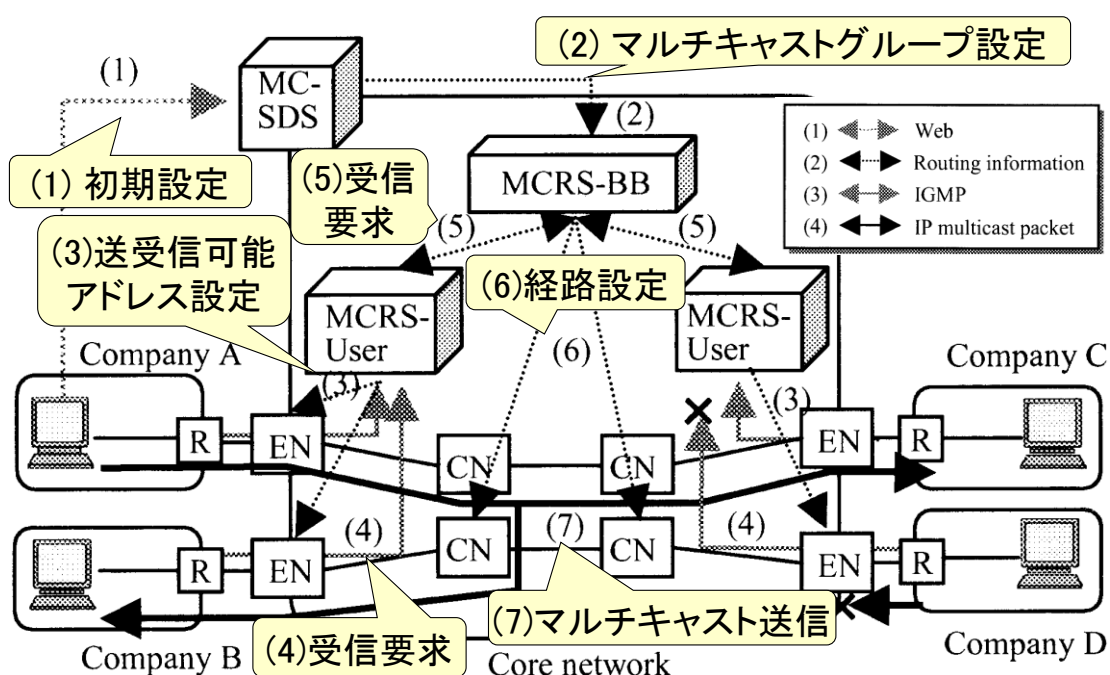


図 3-4. サーバ型マルチキャストシステムの動作

図 3-4 では、企業 A, B, C, D のネットワークが異なるエッジノード (EN) に接続されている。企業 A, B, C は送受信者制限が行われているマルチキャストグループに参加し、企業 D は参加権限を持たず送受信が出来ないものとする。各企業ネットワークのボーダルータ (R) は IGMP Proxy 機能を備えており、各企業ネットワーク内の端末は IGMP を用いて受信要求を MCRS-User へ送信することが可能である。この状況において、企業 A, B, C のネットワーク間でテレビ会議を行う場合の動作は以下の通り。

1. 初期設定として、TV会議の管理者は、MC-SDSに、マルチキャストグループへ送受信できる企業 A, B, C の IP アドレスを登録する。
2. マルチキャストグループ設定として、MC-SDS は MCRS-BB へマルチキャストグループに送受信できる IP アドレスを通知する。MCRS-BB はマルチキャストグループが利用する IP マルチキャストアドレスを割り当て、マルチキャストグループと送受信可能な IP アドレスの組み合わせを MCRS-User へ設定する。
3. 送受信可能アドレス設定として、MCRS-User は EN へ送受信可能な IP アドレスの組み合わせを設定する。
4. 受信要求として、企業 A, B, C のネットワークの端末からボーダールータに対し、IGMP を用いて受信要求を行う。ボーダールータは IGMP Proxy 機能を用いて受信要求を EN へ送信する。EN は登録情報に基づいて受信要求に含まれる Source IP アドレスを認証し、MCRS-User から設定された送受信可能な IP アドレスを照合する。一致している場合は、受信要求を MCRS-User へ送信する。
5. 受信要求として、MCRS-User は、EN から送信された受信要求に含まれるマルチキャストアドレスと IP アドレスを、登録されている組み合わせと照合する。登録されていれば、マルチキャストアドレスに対する初めての受信要求であれば、受信要求を MCRS-BB へ送信する。2回目以降の受信要求であれば、MCRS-BB へ受信要求を送信せず、マルチキャストアドレスに対する受信要求数をインクリメントする。
6. 経路設定として、MCRS-BB は、受信要求を受け取った後、ネットワークにおけるマルチキャスト経路を計算し、企業 A, B, C を収容する EN とそれらを接続する CN ヘルパーテーブルを設定する。
7. マルチキャスト送信として、企業 A のネットワークからマルチキャストグループに対し、マルチキャストパケットを送信すると、MCRS-BB からの設定に基づき、

EN, CN はマルチキャストパケットを受信, コピーして企業 B, C のネットワークへ送信する. なお, 企業 D のネットワークから受信要求があったとしても, EN にて受信要求は拒否され, ルーティングテーブルが設定されないため, 送信者制限がかかることとなる. マルチキャストグループからの離脱時は, 同様に IGMP メッセージを EN へ送信し, MCRS-BB がマルチキャスト経路を再計算し設定を行う. なお, 企業 D のユーザが企業 B や C の IP アドレスを詐称しても, マルチキャストパケットを送信できない.

本研究が対象とするキャリアのバックボーンネットワークにおいては, キャリアからユーザに IP アドレスが払い出されており, キャリアにおいては回線 ID と IP アドレスの紐付け管理を行っている. そのため, EN にて払い出し以外の IP アドレスからの送信は拒否することが可能である. そのため, 企業 D のユーザが企業 B や C の IP アドレスを詐称してマルチキャストパケットを送信したとしても, EN にて回線 ID と IP アドレス違いを判断することが出来るため, 送信者制限が可能である.

3.4. 提案方式の課題

MCRS はユーザ受信要求を集中管理するため, 処理負荷が集中する課題がある. イベントなど受信要求が時間的に集中し処理負荷が増大したり, バックボーンネットワークの拡大に応じて処理負荷が増大したりする. 処理負荷の集中を改善するため, MCRS を, 受信要求を集約する MCRS-User と全体制御を行う MCRS-BB に分割する (図 3-5). MCRS-User が EN 単位に受信要求を集約して MCRS-BB へ送信する. これにより, MCRS が直接ユーザからの要求を受け付ける方式に比べて, MCRS-BB への受信要求を削減することができるため, 処理負荷に応じて適切に MCRS-BB と MCRS-User を配置することで, 大規模なバックボーンネットワークにおいても, 提案するビジネス向けマルチキャスト技術を適用可能となる.

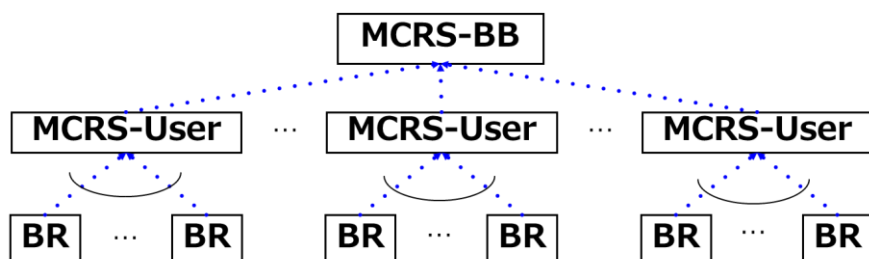


図 3-5. MCRS の分割

3.5. 提案方式の設備設計

提案方式を大規模なネットワークに適用するには、ユーザの利用状況における処理負荷を見積もり、適切な設備設計を行って、MCRS-BB と MCRS-User の配置を行う必要がある。設備設計を行うために処理負荷となるユーザ受信要求の頻度を計算する。

ユーザを収容するボーダールータ（BR）からのマルチキャストグループへの参加と離脱が図 3-6 の確率で状態遷移するとモデル化する。

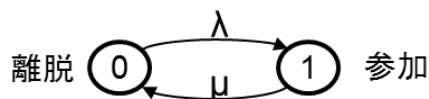
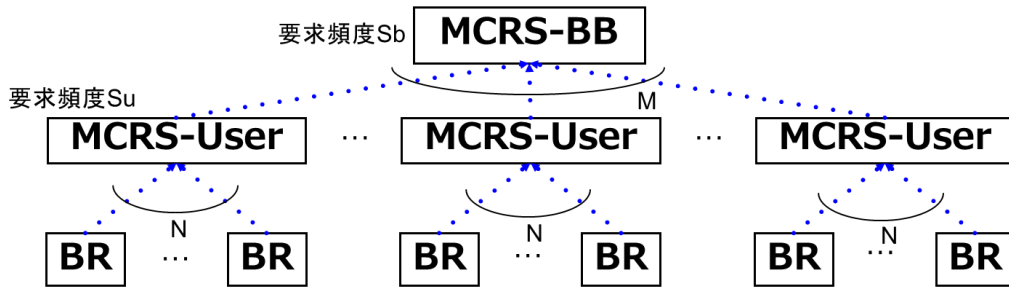


図 3-6. ユーザの参加離脱の状態遷移モデル

ユーザから MCRS-User へのマルチキャスト受信要求頻度、MCRS-User から MCRS-BB への受信要求頻度を図 3-7 に定義する。



S_u : MCRS-User への BR からの受信要求頻度

S_b : MCRS-BB への MCRS-User からの受信要求頻度

N : MCRS-User に接続する BR 数

M : MCRS-BB に接続する MCRS-User 数

図 3-7. ユーザのマルチキャスト受信要求頻度

マルチキャストグループにおける平均視聴時間を $L=1/\mu$, 視聴率を $V=\lambda/(\lambda+\mu)$ とすると, MCRS-User および MCRS-BB へのマルチキャスト受信要求頻度は式(3-1)のように計算できる.

$$S_u = \frac{2NV}{L}, S_b = \frac{2MNV}{L} (1 - V)^{N-1} \quad (3-1)$$

実際の利用環境では, 各マルチキャストグループはその特性ごとに平均視聴率は異なる. マルチキャストグループが G 種類ある際の各マルチキャストグループ i の視聴率 V_i が Zip の法則に従うと仮定し, マルチキャストグループにおいて実際に視聴するユーザの割合を α ($0 \leq \alpha \leq 1$) とすると式(3-2)のように計算できる.

$$V_i = \frac{C\alpha}{i}, \text{ where } C = 1/\sum_{i=1}^G \frac{1}{i} \quad (3-2)$$

各マルチキャストグループの視聴率が独立していると仮定すると、式(3-1)と式(3-2)より、G種類ある全てのマルチキャストグループからの MCRS-User への BR からの受信要求頻度 $S_u total$ は式(3-3)のように計算できる。

$$S_u total = \sum_{i=1}^G \frac{2NV_i}{L} = \frac{2N}{L} \sum_{i=1}^G V_i = \frac{2N}{L} \alpha \quad (3-3)$$

式(3-3)より、ユーザ数およびマルチキャスト視聴率/視聴時間に応じた適切な設備設計が可能となる。

計算例を示す。短時間と長時間の視聴が混在する場合として、平均視聴時間 $L=10$, $L=1000$ の 2 つのマルチキャストグループが 1:99 の割合で混在する環境における $S_u total$ は式(3-4)のように計算できる。

$$\begin{aligned} S_u total &= (S_u total\{L = 10\} \times 0.01 + S_u total\{L = 1000\} \times 0.99) \\ &= 3.98 \times 10^{-3} N \alpha \end{aligned} \quad (3-4)$$

2000年代のサーバ能力では、MCRS-User1台の受信要求処理能力は20Hzと想定していた。この場合1台のMCRS-Userで処理できるユーザ数と実利用者の比率は $N\alpha = 20/3.98 \times 10^{-3} = 5025$ となる。実利用者の比率ごとに1台当たり処理可能なユーザ数を

表 3-1 に示す。これにより利用状況に応じて MCRS-User の設備数を決定可能となる。

表 3-1. MCRS-User1 台が収容可能なユーザ数

実利用者比率(α)	ユーザ数(N)	設備設計例
0.05	100,480	10万規模の中小都市に1台
0.1	50,240	10万規模の中小都市に2台
0.2	25,120	10万規模の中小都市に4台

本論文執筆時では、2000年代と比較してCPUクロック数の増加は10倍以上となっており[36]、処理可能なユーザ数も10倍以上と見積もられる。現在はコンピューティングの仮想化によりマルチキャスト利用状況に応じたコンピューティングリソースの動的な割当が可能である。本研究の成果を活用し、ユーザ利用状況を基に自動的にリソース割当を行うオペレーションも可能である。アプリケーションとネットワークを連動した品質制御が求められているため、利便性がより高まることにより、利用者数が増加することが想定される。本論文の想定よりも利用者が増加する場合は、 α の値を多くするように見直して、設備設計に活用することが必要となる。

3.6. まとめ

本節では、サーバ型マルチキャスト管理システムを提案した。マルチキャストルーティングサーバ(MCRS)が、ユーザ認証、トラフィック情報、経路情報を管理し、バックボーンネットワークの経路制御を実施する。ユーザを識別して送信制限を行うため、通信の乗っ取りを防止し、ユーザ契約に基づく送信可否の管理や課金が可能となる。設備設計に必要な値の導出を行い、利用状況に応じたシステムリソースの導出を可能とし、効率的な運用を可能とした。

4. オンデマンドフロー管理によるリソース利用効率化

4.1. 背景

イントラネット、エクストラネットの普及に伴い、企業における IP アプリケーションの利用が進んでいる。これらのネットワークにおいて、一般的な IP アプリケーションである電子メールやウェブブラウザばかりでなく、グループウェアや社内システム等の業務アプリケーション、内線電話や社内放送などの VoIP/ストリームアプリケーションも同時に用いられる傾向にある。また、リモートワークが急速に普及しており、バックボーンネットワークにおける IP ストリーミングアプリケーションにおいて、ビデオ会議システムなど双方向通信が拡大している。

双方向のリアルタイムストリーミングは、音声や動画の品質劣化がユーザに意識されやすいため、体感品質を保つには通信遅延や揺らぎを一定以下にする必要がある。一方で、バックボーンネットワークには電子メールやウェブなどのリアルタイム性を要求しない通信など様々な種類のトラフィックが混在しており、一律の遅延制御は難しい。

様々なトラフィックが混在するネットワークにおいて、上記の重要アプリケーションフローの通信品質を選択的に優先制御するため、アプリケーションフロー毎の優先/非優先制御を行う品質制御装置(Packet Shaper[37]等)の導入が進んでいる。これらの品質制御装置は、アプリケーションフローを認識し、フロー毎にあらかじめ設定された品質制御を行う。これにより、重要アプリケーションのレスポンス向上や、むやみな通信回線増強の回避を図ることが出来る。

しかし、これらの装置を運用する場合、設定の容易性、設定の連携性、設定の即応性、が求められる。本研究を行った 2000 年代は、フロー毎の優先／非優先制御を行う通信品質制御装置が活用されていたが、企業網内など特定のネットワーク範囲において、あらかじめ設定された品質制御を行うものであり、幅広いユーザ通信需要をオンデマンドに対応していなかった。バックボーンネットワークにおける通信品質の確保やネットワークリソースの利用を効率化するためには、一般ユーザの通信要求を把握し、通信品質制御を一元的に管理して、リアルタイムストリーミングを選択的に優先するフロー管理技術が必要となる（図 4-1）。

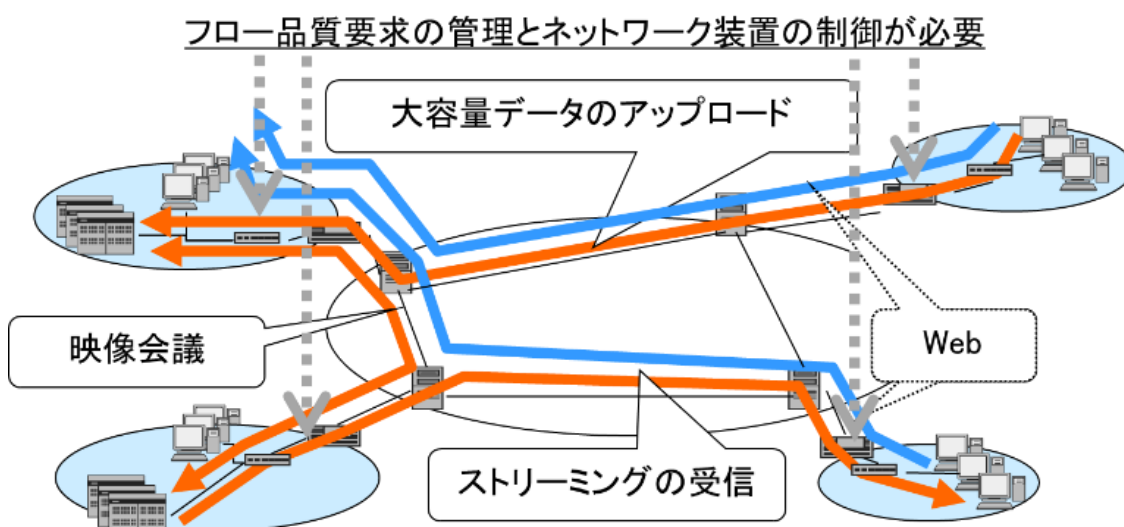


図 4-1. 通信品質の多様化

4.2. 問題点と課題

本研究を行った 2000 年代においては、市販の品質制御装置の設定や運用は、内蔵のコマンドライン/Web インタフェースを用いて行っていた。また、専用オペレーティングソフトウェアを介して装置を設定する場合もあった。これらの方法には、次の問題がある。

1. 設定の容易性

市販の品質制御装置の設定は、実トラヒック測定値を元に行っている。そのため、トラヒックの変動によって設定値が不適切になる可能性がある。このため、市販の品質制御装置あるいは専用オペレーティングソフトウェアには、トラヒックレポートやネットワーク利用率を出力する機能が設けられており、このデータを基にオペレータがフロー利用状況を推定し、設定と測定の繰り返しによって、最適な設定値を割り出すことも行われていた。

しかし、ユーザの利用状況により変動するトラヒックの予測は困難であり、短期間での設定値の割り出しは非常に困難である。また、トラヒック状況の分析にはデータの蓄積が必要であるが、ネットワークの利用状況は刻々と変化するため、蓄積によっても最適な設定値を見出すことが困難である。さらに、設定値を最適にするには、設定と測定を繰り返してチューニングする必要があるが、オペレータの運用稼働が大きく、かつ利用状況変動時の即応性に欠ける、という問題がある。

これらの問題を解決するには、各ユーザの利用要求に基づいた設定を行う仕組みが必要である。

2. 設定の連携性

同一の運用ポリシーに基づき管理されるネットワークに複数の品質制御装置を設置する場合、1つのフローが複数の装置を通過する可能性がある。そのため、装置間の設定を連携させないと、意図しないフロー廃棄が発生する問題がある。フロー廃棄を防ぐために、フローを意識した連携設定が必要である。

しかし、市販の品質制御装置の設定は、個々の装置が持つコマンドラインインタフェースあるいは Web インタフェースを用いて、オペレータが個々の装置毎に行う。また、市販の専用オペレーティングソフトウェアを用いた場合、1つのソフトウェアで同時に複数の品質制御装置を管理することも可能であるが、設定は装置単位であり、

フロー単位ではない。このため、1つのフローが通過する複数の品質制御装置を連携して設定するには、オペレータが、フローの通過する品質制御装置を手作業でリストアップして設定しなければならず、運用稼働の増大や設定ミスの可能性を排除できないという問題がある。

これらの問題を解決するには、ネットワーク構成、経路、フロー属性を一元的に管理し、フロー単位で複数の品質制御装置を管理、設定する仕組みが必要である。

3. 設定の即応性

品質制御装置の設定は、内蔵のコマンドライン/Web インタフェースもしくは専用オペレーティングソフトウェアを介して行っている。いずれの方法も、全てのフロー設定を同一ユーザが行えるといったシングルユーザの概念で設計されているため、セキュリティの観点から、個々の一般ユーザが品質制御装置の設定を行うことはできない。このため、一般ユーザのフロー設定要求はオペレータが受け付け、帯域の割り振りをを行い、必要な装置に設定することとなり、オペレータの運用稼働が大きいという問題がある。さらに、ユーザの要求にオンデマンドで対応できないため、即応性にかけ、効率的な帯域利用が出来なくなる問題がある。

これらの問題を解決するには、オペレーティングにマルチユーザの概念を導入し、個々の一般ユーザが、自分が設定できるフローや帯域の範囲内で、必要なときに帯域を増大/減少させる仕組みが必要である。

4.3. 提案方式

この課題を解決するため、本研究では、ユーザの通信品質要求をオンデマンドで受け付け、要求に基づいてバックボーンネットワーク上に配置された複数の品質制御装置を一元的に管理、設定するポリシ階層型オンデマンドフロー管理システムを提案する[38][39]。提案方式は、ネットワークの各拠点に設置された品質制御装置 (Quality

control device) と、バックボーンネットワークに設置したオンデマンドフロー管理システム (On-demand flow management system) により構成される (図 4-2)。

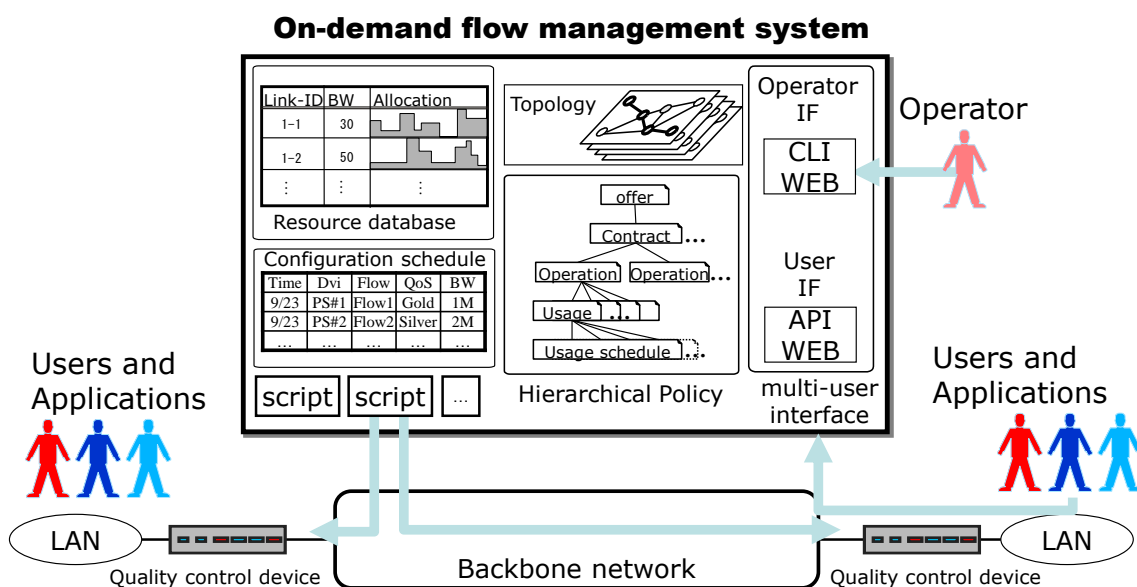


図 4-2. オンデマンドフロー管理システム

4.4. 制御動作

本方式において、フローは次のように制御される。

1. 初期設定

オペレータは、サービス開始前に、サーバに対し、品質制御装置の配置情報などを含むネットワーク構成情報、経路情報、帯域情報、を入力する。また、ネットワーク回線の帯域情報などを含むリンク帯域情報を入力する。さらに、各ユーザのポリシー情報 (契約情報など) を入力する。

2. フロー設定要求

ユーザは、フローが必要なとき、あるいは今後必要になると判断したときに、サー

バへフロー情報と要求品質を入力し、フロー設定要求を行う。サーバは、ユーザの要求情報を聞き、リンク帯域情報とユーザの要求帯域を確認して超過しなければ、ユーザのフロー設定要求を許可できると判断した場合、サーバは要求を受け入れて、フロー情報を設定する。

3. フロー設定

サーバは、フロー開始時間になると、当該フローが経由する品質制御装置に対し、ユーザが入力したフロー情報を設定し、フロー制御が可能となる。

4.5. システム構成

本研究において、オンデマンドフロー管理システムを実装した。実装においては、図 4-3 のように、複数の機能部とリポジトリから構成した。



図 4-3. オンデマンドフロー管理システム構成

システム動作の流れを、機能部とリポジトリの観点から述べる。

オペレータがオンデマンドフロー管理システムに対し、オペレーション受付部を介して、ユーザ情報、リソース情報、ポリシー情報を入力し、オンデマンドフロー管理シ

システムはそれらの情報をそれぞれのリポジトリに格納する。次に、ユーザがオンデマンドフロー管理システムに対し、フロー設定受付部を介して、フロー要求を行う。オンデマンドフロー管理システムは、ユーザ情報、ポリシー情報を参照して、当該ユーザが当該フローを要求できるかどうか判断する。さらに、リソース情報を参照して、通過するリンクが当該フローの帯域を収容できるかどうか判断する。フロー要求が受け付け可能な場合、オンデマンドフロー管理システムはユーザに利用可能と伝え、リソース情報、フロー要求情報に当該フローの情報を追加する。オンデマンドフロー管理システムのスケジューラはフロー要求情報を参照し、利用開始時間を迎えたフロー設定を、通過する品質制御装置に行う。品質制御装置は、フロー設定に従い、フローの帯域を制御する。

4.6. 特徴

提案するオンデマンドフロー管理システムは次の特徴を持つ。

1. ユーザ要求に基づくフロー管理

本研究を行った 2000 年代においては、オペレーティングソフトウェア等では、実トラヒック情報やオフラインでの一般ユーザの要求を元にオペレータがフロー設定を行っていた。オンデマンドフロー管理システムは、一般ユーザのフロー設定要求を Web インタフェースによりオンデマンドで受け付け、設定する。このため、要求をリアルタイムに品質制御装置の設定に反映できるほか、一般ユーザのフロー設定要求を蓄積し、ネットワークの利用傾向を把握して設備設計や自動的な制御に活用することが可能となる。

2. フロー単位のオペレーション

本研究を行った 2000 年代において行われていた品質制御装置単位のオペレーションと異なり、フロー単位のオペレーションシステムでは、ネットワークのリソースを一

元管理し，フロー単位でオペレーションを行う．ユーザのフロー設定要求を基に，フロー経路上の各リンクおよび品質制御装置の帯域を論理的に管理・予約し，フロー設定開始時間に，経路上の複数の品質制御装置を同時に設定する．これにより，フロー単位で帯域の確保を行い，設定の同期により経路上の帯域を確保し，品質低下を抑止することが出来る．

3. マルチユーザ管理

本研究を行った 2000 年代において行われていたシングルユーザモデルのオペレーティングソフトウェアと異なり，本システムはオペレータから一般ユーザまで，権限の異なる 4 種類のユーザを設ける ()．ユーザを複数種類設置し，権限を分割することで，運用稼働の分散が可能となる．

マルチユーザ管理機能により，キャリアのネットワークオペレータが，オンデマンドフロー管理システムを通して，これまで出来なかった一般ユーザの品質要求を受け付けることが可能となっている (表 4-1)．ユーザ要求を直接受け付けてフロー管理が可能となるため，オペレータ経由での設定に比べて，よりリアルタイムにユーザ要求を受け付けて品質制御装置へ設定できる．

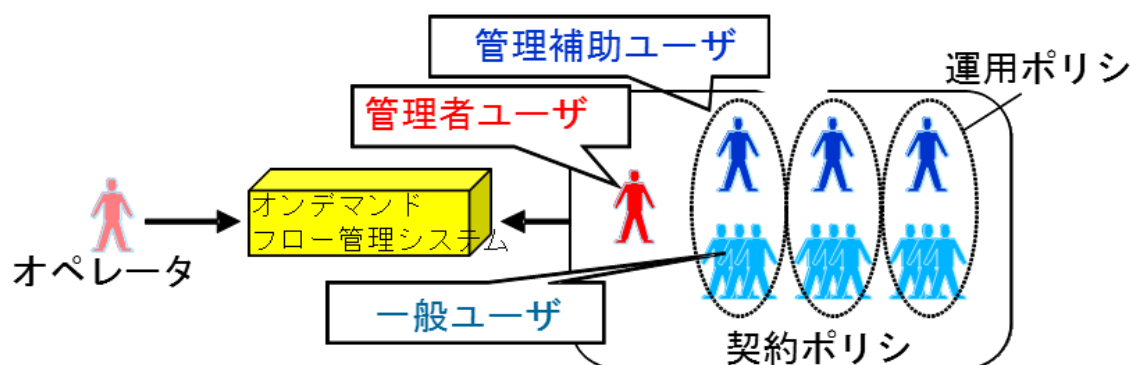


図 4-4. マルチユーザ管理

表 4-1. ユーザー一覧

	ユーザ種別	ユーザ概要
1	オペレータ	本システムの運用を行う。ネットワーク構成情報、経路情報、リンク情報などの基本的ネットワーク構成情報の管理設定を行う。通信キャリアオペレータが務める。
2	管理者ユーザ	本システムにおけるフロー利用形態やポリシーの設定などを行う。ユーザのネットワーク管理者が務める。
3	特別ユーザ	管理者ユーザが務める運用の一部を代行する。管理者ユーザが設定した制限の下、フロー利用形態の設定や一般ユーザの生成削除を行う。
4	一般ユーザ	本システムへフロー設定要求を行い、フロー品質を利用する。利用したいフローの発着 IP アドレス、ポート番号、帯域、必要日時等を入力し、フロー設定要求を行う。

4. ポリシの階層化

本システムでは、各ユーザ種別にて実行可能なフロー管理設定を階層化し、4種類のポリシーを設ける（

表 4-2）。それぞれのポリシーはユーザの役割毎に分類されており、下位ポリシーが上位ポリシーに違反しないように管理する。また、上位のポリシーから下位のポリシーへ移行するほど、より詳細な設定を行うように、ポリシーを階層化している。オンデマンドフロー管理システムは下位ポリシーが上位ポリシーに違反しないよう管理する。このようなポリシーの階層化により、設定項目数の分割による運用稼働の分散、ユーザの権限に応じた

設定項目の配置による設定の容易さ，を提供できる。

表 4-2. ポリシー一覧

	ポリシー種別	ポリシー概要
1	提供ポリシー	ネットワーク構成を下に定めるポリシー。品質制御装置の種別，性能，運用可能な品質グレードなどを規定する。
2	契約ポリシー	ユーザが利用できるフローや帯域，ユーザネットワーク構成などを規定する。
3	運用ポリシー	提供可能なフローの枠組を定める。発着ネットワークアドレス，品質グレード，帯域，方向，利用可能ユーザなどを規定する。
4	利用ポリシー	利用可能なフローを定める。運用ポリシーの枠に基づき，利用可能時間やフロー情報を規定する。

また，ポリシーを階層化しているため，各ユーザの役割毎にポリシーを設定し，ポリシー設定情報を分割することで運用稼働の分散を可能とした。図 4-5 に示すとおり，一般ユーザは利用ポリシーのみを，管理者ユーザは運用ポリシーのみを設定するだけでよい。提供されるサービス以上の設定を行わないように，オペレータが設定する提供ポリシー，契約ポリシーとの整合性は，オンデマンドフロー管理システムが担保する。これにより，一般・管理者・管理補助ユーザは，ポリシー全体を把握した設定を行うことが不要となり，設定が容易となる。

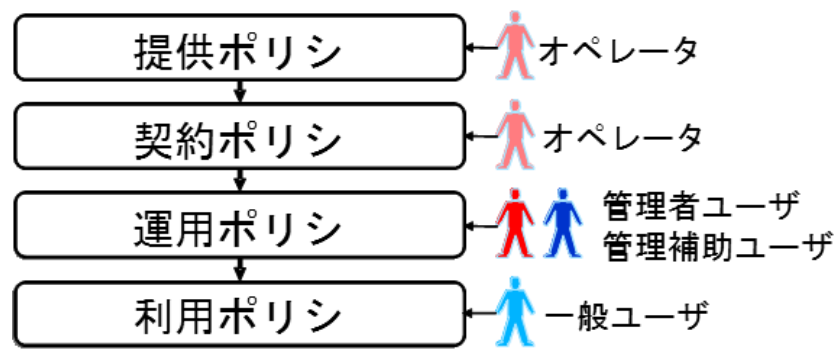


図 4-5. ポリシの階層化

5. 論理的な帯域管理

本システムはフロー設定要求に基づき、ネットワーク全体の帯域要求を時間単位、ユーザ単位、経路単位で論理的にデータベース管理している。一般ユーザからのフロー設定要求をデータベースに反映することで、各リンクで現在使用されている帯域を論理的に管理できる。リンク毎に物理帯域と最大割り当て可能率を指定することが出来るため、オペレータに対し、特定リンクへのフロー集中による輻輳を未然に回避する手段を提供することが可能になる。さらに、物理的な帯域および経路制御を行うトラヒックエンジニアリングシステムと連携し、フローの経路における必要な帯域を設定し、ユーザ要求に応じた通信品質を提供する。

4.7. 効果

4.2 節で示した 3 つの課題に対するオンデマンドフロー管理システムの効果を示す。

1. 設定の容易性

設定の容易性を解決するには、4.2 節で示したように各ユーザの利用要求に基づいた設定を行う仕組みが必要であるが、提案システムは一般ユーザからのフロー設定要求を直接受け付け、各ユーザが利用する帯域を論理的に管理する。また、フロー設定の予約も行い、現時点だけではなく未来にわたって必要とされるフロー設定、帯域も管

理する。これにより、ユーザのフロー利用状況を幅広く把握し、適切な装置に適切な設定を行うことが可能になる。また、ポリシーを階層化して各レベルのユーザが行う設定を簡易化しているため、一般ユーザは簡単な項目を入力するだけでフロー設定が可能である。

2. 設定の連携性

設定の連携性を解決するには、ネットワーク構成、経路、フロー属性を一元的に管理し、フロー単位で複数の品質制御装置を管理、設定する仕組みが必要であるが、提案システムはネットワーク上の複数の品質制御装置を一元的に管理し、フロー単位で必要な装置に設定を行う。これにより、オペレータやユーザは、フローがどの品質制御装置を通過するか意識することなく、フロー設定を行うことが可能である。

3. 設定の即応性

設定の即応性を提供するには、個々の一般ユーザが、自分が設定できるフローや帯域の範囲内で、必要なときに帯域を増大／減少させる仕組みが必要であるが、提案システムはオペレーティングにマルチユーザの概念を導入し、一般ユーザが直接フロー設定要求を行える。これにより、ユーザのフロー設定要求に対し、オンデマンドで品質制御装置の設定を行うことが可能である。

解決すべき課題と提案システムの特徴との対応関係を図 4-6 に示す。本システムにより、課題を網羅的に解決している。

以上のように、本システムは、一般ユーザから管理ユーザやキャリアオペレータまで階層化したユーザ構造、および利用から契約や運用まで階層化したフロー管理ポリシーを持ち、一般ユーザのフロー管理要求の受け付けを行いネットワーク運用ポリシーと整合し管理するため、バックボーンネットワークにおいて、ユーザ単位で要求を把握し、フロー単位で通信品質を制御が可能となる。

	設定の容易性	設定の連携性	設定の即応性
ユーザ要求に基づくフロー管理	■		■
フロー単位オペレーション		■	
マルチユーザ管理	■		■
ポリシーの階層化	■		
予約によるフロー設定	■		
論理的帯域管理	■		

図 4-6. 解決すべき課題と提案システムの特徴との対応

4.8. 提案方式の設備設計

提案方式を大規模なネットワークに適用するには、ユーザの利用状況を踏まえた設備設計を行い、フロー管理システムの必要台数などの設備設計を行う必要がある。設備設計に必要な処理負荷の計算を行う。

本システムの動作を図 4-7 のようにモデル化する。このモデルでは、 N 人の一般ユーザが 1 台のフロー管理システムに対し、 K 通りのフローのいずれかを要求する。システムは M 台の品質制御装置に対し設定を行う。各一般ユーザは独立してフロー要求を行う。システムは 1 台の品質制御装置に対し、1 フローあたり 1 回の設定と削除を行う。

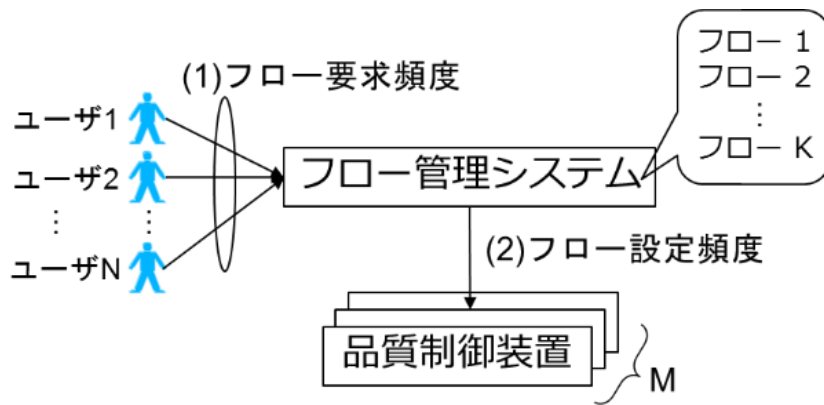


図 4-7. 本システムの動作のモデル化

1 ユーザのオンデマンドフロー管理システムに対する 1 フローの即時利用要求動作に着目すると、「フローを利用している」「フローを利用していない」の 2 状態の間を遷移していると考えられる。そこで、一般ユーザのフロー管理システムへのフロー要求動作を図 4-8 のようにモデル化する。

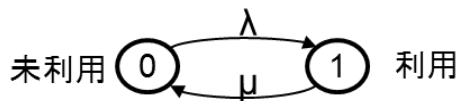


図 4-8. 一般ユーザのフロー要求動作モデル

フローを利用していない状態を P_0 、利用している状態を P_1 とし、状態間は λ 、 μ の確率で遷移する。ユーザは P_0 、 P_1 のいずれかの状態に属している。システムは定常状態にあると仮定する。

このモデルにおいて、式(4-1)、(4-2)が成立する。

$$\lambda P_0 = \mu P_1 \tag{4-1}$$

$$P_0 + P_1 = 1 \tag{4-2}$$

式(4-1), (4-2)より, 式(4-3)のように P_0, P_1 が導出できる.

$$P_0 = \frac{\mu}{\lambda + \mu}, \quad P_1 = \frac{\lambda}{\lambda + \mu}. \quad (4-3)$$

(1) フロー要求頻度

全てのユーザが Zip のモデルに従ってフロー設定要求を行うと仮定すると, 1 フローに対する全ユーザの単位時間当たりのフロー要求数 F_{user} は式(4-4)となる.

$$F_{user} = \text{ユーザ数} \times P(\text{未利用} \rightarrow \text{利用}) = N\lambda P_0 \quad (4-4)$$

また, Zip のモデルにおいて平均利用率を V , 平均利用時間を L とする. 一般ユーザの要求フローの平均利用時間を $L=1/\mu$, フロー要求の平均利用率を $V=\lambda/(\lambda+\mu)$ とすると, 1 フローに対する全ユーザのフロー要求数 F_{user} は式(4-5)のように計算できる.

$$F_{user} = \frac{NV}{L} \quad (4-5)$$

フローが K 種類ある際の各フロー i の平均利用率 V_i が Zip の法則に従うと仮定し, 実際に利用要求を行うフローの割合を $\alpha (0 \leq \alpha \leq 1)$ とすると, V_i は式(4-6)のように計算できる.

$$V_i = \frac{C\alpha}{i}, \text{ where } C = 1 / \sum_{i=1}^K \frac{1}{i} \quad (4-6)$$

ユーザ間のフロー要求が独立していると仮定すると, 式(4-5)と式(4-6)より, K 種類ある全てのフローに対する全ユーザのフロー要求数 $F_{usertotal}$ は式(4-7)のように計算できる.

$$F_{user\ total} = \sum_{i=1}^K \frac{NV_i}{L} = \frac{N}{L} \sum_{i=1}^K V_i = \frac{N}{L} \alpha \quad (4-7)$$

(2) フロー設定頻度

フロー管理システムから品質制御装置へのフロー設定、削除動作は、フロー利用状態への遷移、未利用の状態への遷移の際に行われるため、図 4-8 のモデルを用いる。フローは 2 拠点間で生成され 2 つの品質制御装置を経由することから、1 つのフローが設定される際に、ある品質制御装置が設定を受ける確率を $2/M$ とする。この場合、フロー管理システムが 1 つの品質制御装置に対して行うフロー設定頻度 F_{system} は式(4-8)のように計算できる。

$$F_{system} = N(\lambda P_0 + \mu P_1) \frac{2}{M} = \frac{4NV}{LM} \quad (4-8)$$

式(4-6)と式(4-8)より、全フローについてフロー管理システムが 1 品質制御装置に行うフロー設定数 $F_{system\ total}$ は式(4-9)のように計算できる。

$$F_{system\ total} = \frac{4N}{KM} \sum_{i=1}^K V_i = \frac{4N\alpha}{LM} \quad (4-9)$$

式(4-7)および式(4-9)によりオンデマンドフロー管理システムの必要台数などの設備設計が可能となる。代表的な事例を基にした計算例を表 4-3 に示す。

表 4-3. パタン毎のフロー要求数とフロー設定数

	中規模企業	大規模企業	CDN
ユーザ数 N	1,000	10,000	100,000
平均利用時間 L	3,600	3,600	600
品質制御装置数 M	10	20	100
利用要求フロー割合 α	0.2	0.2	0.05
ユーザのフロー要求数 $F_{user\ total}$	0.06	0.56	8.33
システムのフロー設定数 $F_{system\ total}$	0.02	0.11	0.33

中規模企業の社内ネットワークを想定した場合、業務に関するフロー設定が多いため、平均利用時間は比較的高いと考えられる。大規模企業の社内ネットワークを想定した場合、中規模企業と同様に平均利用時間は高く、ユーザ数や品質制御装置も多いと考えられる。コンテンツデリバリネットワーク（CDN）を想定した場合、ザッピング視聴が多いと想定されるため、平均利用時間は比較的に短いと考えられる。

フロー管理システム 1 台の要求処理能力を 2000 年代のサーバ能力を想定して 20Hz と仮定すると、いずれの事例においても十分処理できると考えられる。本計算は定常状態を仮定しているため、突発的な負荷集中なども踏まえた設備設計が必要となる。

現在はコンピューティングの仮想化によりマルチキャスト利用状況に応じたコンピューティングリソースの動的な割当てが可能である。本研究の成果を活用し、把握したユーザ要求を、ネットワーク制御だけでなく、コンピューティングリソース制御にも活用し、サービス品質を向上することが可能である。例えば Web 会議を提供するクラウドのコンピューティングリソース制御を連動して行い、ユーザが求める品質に必要な

なコンピューティングリソースをオンデマンドで提供することも可能と考えられる。ユーザの要求を把握・管理する特徴は、IT サービス提供において幅広く活用が可能と考えられる。

4.9. まとめ

本章では、ポリシ階層型オンデマンドフロー管理システムを提案した。設定の容易性、連携性、即応性を提供することが可能となり、個人・サービス単位で必要なリソースを割り当てられるため、様々な通信品質を利用可能としサービスの利便性を向上した。処理負荷となるフロー要求頻度/設定頻度の計算式を作成し、フロー利用時間/視聴時間に応じた設備設計を可能とした。ネットワーク規模による設備設計が可能となり、バックボーンネットワークへの適用が可能となった。

5. 現在のネットワークへの適用について

5.1. バックボーンネットワーク制御への適用

2章で記載した研究は、2000年代を中心に行ったバックボーンネットワークのリソース利用を効率化する研究開発である。当時から光通信とIP通信の組み合わせがあったように、現在も大容量・低遅延の光通信装置がさらに導入されており、光波長でのルーチングを行うオールフォトニックネットワーク[40]に向けて、IP通信と光通信の効率的な組み合わせ利用の必要性は引き続き高い(図5-1)。

本研究におけるトラヒック配置・通信要求の管理システムは、様々なリソース種別を定義できるため、フォトニックレイヤの様々な組み合わせに対応できる経路計算手法を導入すれば、現在のバックボーンネットワーク制御にも適用できると考える。

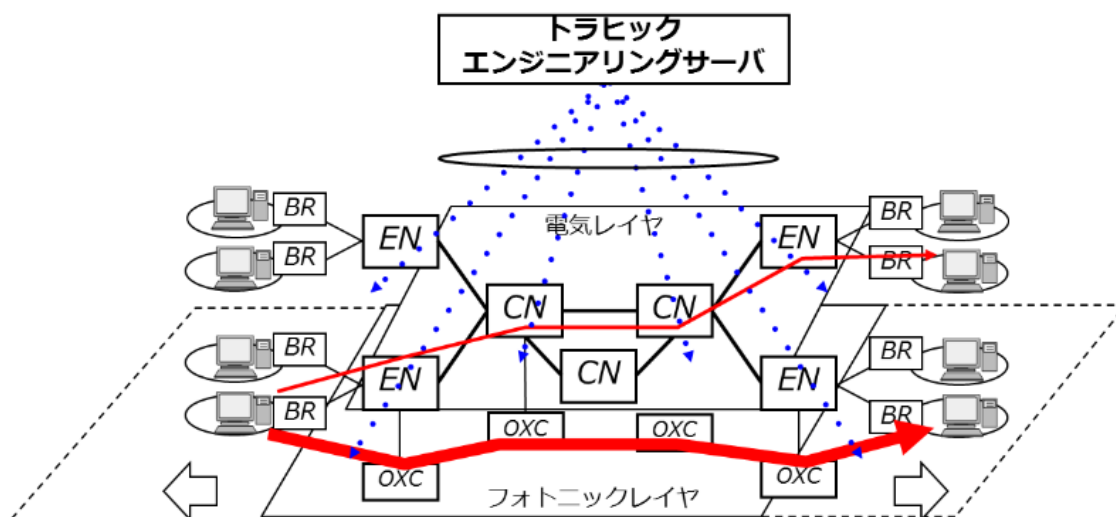


図 5-1. 現在の電気・フォトニックによるバックボーンネットワーク

5.2. インテントベースのアプリケーション・ネットワーク ワーク協調制御への適用

2章で記載した研究は、2000年代を中心に行ったバックボーンネットワークのリソース利用を効率化する研究開発であるが、それ以降、SDN（Software-Defined Networking）やNFV（Network Functions Virtualization）等のネットワーク仮想化技術の発展により、ネットワークは動的かつ柔軟に制御することが可能になりつつある。ネットワークを介したサービスは多様化しており、超高速・大容量、超低遅延、超低消費電力・低コスト化などネットワークに求められる要件も複雑化している。

現在、ユーザが快適にサービスを楽しむためには、多様化するサービスに対するユーザの要求（インテント）を正しく把握したうえでインテントを満たすように要件を策定し、策定した要件に基づいて制御を行う研究開発が進んでいる[41]。

ユーザ端末で用いられているテレビ会議やストリーミングなどの通信品質を必要とするアプリケーションにおいては、サービスに対するユーザの満足度を向上させるために、ネットワーク品質を考慮したアプリケーションデータの配信レートの制御がされている。ネットワークにおいては、ネットワーク内で取得したトラフィック情報を用い、トラフィックエンジニアリングにより、与えられたネットワーク品質要件を満たす経路の制御が行われている。しかし、現状では、アプリケーション制御とネットワーク制御は独立しているため、ユーザの満足度やネットワークリソースの利用効率の観点で十分な効果が得られない課題がある。そのため、インテントに基づいてアプリケーション制御とネットワーク制御を協調させ、ユーザの満足度を高めるアプローチの研究開発が進んでいる。

4章で記載した研究が提案するオンデマンドフロー管理システムは、ユーザのインテントを把握しポリシーに基づく品質制御を行うことが可能である。ユーザのインテントを把握する方法としては、これまでのWebインタフェースに加えて、アプリケーション連動やAIによる振る舞いからの予測などを活用することが出来る。これらインテン

ト情報を API から入力し、本研究のポリシー管理と組み合わせることにより、サービス条件に基づくユーザインテントの最適化を図ることが可能であると考える。さらに、トラヒック配置・通信要求の管理システムと組み合わせることにより、トラヒック情報に加えて、ユーザのインテントを踏まえたトラヒックエンジニアリングによるトラヒックの効率制御が可能となる。本研究成果は、将来的なアプリケーション・ネットワーク協調制御に向けて活用できると考える。

5.3. インテントベースのクラウドサーバ・アプリケーション・ネットワーク協調制御への適用

4章で記載した研究は、2000年代を中心に行ったユーザ要求に基づくオンデマンド品質制御の研究開発であるが、それ以降、リモートワークの急増やデジタルトランスフォーメーション（DX）推進など、予測不能な社会情勢の変化により、サービスに対する需要の急変やユーザニーズの多様化が加速している。

それら環境の変化や多様性に追従する手段として各種システムのクラウドネイティブ化が進み、クラウドサーバ上でさまざまなサービスやネットワーク機能が迅速かつ多様な粒度で提供されるようになってきている。そのため、サーバリソース制御の重要性が増しており、これまでの経験頼りのリソース設計やシステムパフォーマンスベースでのリソース制御では、ユーザ品質要件や利用状況に則したサービス提供が困難である課題がある。

そのため、システム視点に加えて、多様なユーザ要件を考慮した新たなリソース制御技術の研究開発が進められている[42]。新たなリソース制御技術では、ユーザ体感品質を維持しつつ、さまざまなサービス利用状況下における最適リソースを自動算出することで、リソース設計や運用稼働の削減やリソースコストの削減を目指している。

4章で記載した研究が提案するオンデマンドフロー管理システムは、ユーザのイン

テントを把握しポリシーに基づく品質制御を行うことが可能である。また、階層化されたポリシーは属性の追加が容易に可能である。このため、把握したインテントを、ネットワーク制御だけでなく、コンピューティングリソース制御に活用することが可能である。例えば Web 会議サービスの実施において、ユーザのインテントを把握し、ネットワークの品質制御だけでなく、Web 会議を提供するクラウドのコンピューティングリソース制御を連動して行い、ユーザが求める品質をオンデマンドで提供することが考えられる。本研究成果は、将来的なクラウドサーバ・アプリケーション・ネットワーク協調制御に向けて活用できると考える。

6. 結論

6.1. 本論文のまとめ

本論文では、サーバ型トラフィックエンジニアリングシステムによるバックボーンネットワーク制御、マルチキャスト管理およびオンデマンドフロー管理による IP アプリケーション管理を連携させることで、IP アプリケーションのリソース利用状況を把握してトラフィック分散制御を行う研究および実装評価を行った。本研究は、効率的に大容量トラフィックを制御するアーキテクチャ・技術を確立し、大規模化するバックボーンネットワークの効率的・安定的な制御を可能とした。

第 2 章では、トラフィック制御によるリソース利用効率化について論じた。トラフィック情報と経路制御を一元管理することで、トラフィック情報の時間変動に対する収束性を向上させ、経路の安定性を向上させるサーバ型トラフィックエンジニアリング技術、トラフィック流量を最大にする経路計算方式を提案した。日本国内における通信キャリアのバックボーンネットワークでの利用を想定し、現実に近い通信キャリアネットワークを想定したシミュレーションモデルを構築し、トラフィック収容効率の向上を確認した。MPLS エミュレータを用いて大規模な MPLS バックボーンネットワーク環境を模擬した検証環境を構築し、様々なネットワーク構成を生成して、通常運用の LSP 設定や故障時の LSP 設定を評価し、従来方式よりも短い時間で設定できることを示し、提案するサーバ型トラフィックエンジニアリングの有効性を示した。

第 3 章では、マルチキャスト経路管理による他拠点配信の運用効率化について論じた。ユーザの認証を行い、マルチキャスト配信を管理するサーバ型マルチキャスト管理システムを提案した。標準的なマルチキャストプロトコルを終端する機能を持ち、ユーザ契約やリソース利用状況に基づいたマルチキャスト経路配信を一元的に行うため、既存のプロトコルを改変することなく、配信管理が可能となるため、通信キャリ

アがビジネス用途としてマルチキャスト通信を活用できる道を切り開いた。システム性能の評価モデルを構築し、企業内の会議からネットワークワイドの配信まで、さまざまな他地点配信を想定して、サーバ型マルチキャスト管理システムに必要な設備設計に資する性能要件を計算した。

第 4 章では、オンデマンドフロー管理による通信品質管理の運用効率化について論じた。ユーザの通信品質要求をオンデマンドで受け付け、要求に基づいてバックボーンネットワーク上に配置された複数の品質制御装置を一元的に管理、設定するサーバ型オンデマンドフロー管理システムを提案した。一般ユーザから管理ユーザやキャリアオペレータまで階層化したユーザ構造、および利用から契約や運用まで階層化したフロー管理ポリシーを持ち、一般ユーザのフロー管理要求の受け付けを行いネットワーク運用ポリシーと整合し管理するため、バックボーンネットワークにおけるフロー管理を可能とし、ユーザ単位の要求に基づくネットワーク品質制御サービスを提供可能とした。システム性能の評価モデルを構築し、大規模から中小規模までを想定して、サーバ型オンデマンドフロー管理システムに必要な設備設計に資する性能要件を計算した。

第 5 章では、現在の通信キャリアのバックボーンネットワークへの研究成果の適用について論じた。本研究は、2000 年代を中心に行ったバックボーンネットワークのリソース利用を効率化する研究開発であるが、現在のバックボーンネットワークにおける IP 通信と光通信の最適な組み合わせや、ユーザのインテントを把握し、インテントベースのクラウド・アプリケーション・ネットワークの連携制御に活用できることを述べた。

6.2. 今後の課題

本研究は、2000 年代に行ったものであるが、現在の通信キャリアが抱える課題に適用可能である。

2 章で論じたサーバ型トラヒックエンジニアリングシステムは、バックボーンネットワークのリソース利用を効率化する研究開発である。当時から光通信と IP 通信の組み合わせがあったように、現在も大容量・低遅延の光通信装置がさらに導入されており、光波長でのルーチングを行うオールフォトニックネットワークに向けて、IP 通信と光通信の効率的な組み合わせ利用の必要性は引き続き高い。本研究におけるトラヒック配置・通信要求の管理システムは、様々なリソース種別を定義できるため、フォトニックレイヤの様々な組み合わせに対応できる経路計算手法を導入すれば、現在のネットワーク制御にも適用できると考える。

4 章で論じたオンデマンドフロー管理システムは、ユーザの要求を把握・管理し、ポリシーに基づく品質制御を行うことが可能である。現在、ユーザの要求（インテント）を把握し、インテントを満たすように要件を策定し、策定した要件に基づいて制御を行うインテントベースのリソース制御の研究開発が注目されている。3GPP などモバイル系の標準化団体で議論が活発に行われており、サービスに対するユーザの満足度を向上させるために、現状では独立しているアプリケーション制御とネットワーク制御を協調させ、インテントに基づいてユーザの満足度を高めるようにキャリアのリソースを制御する研究開発が進められている。オンデマンドフロー管理システムの成果を活用し、把握したユーザ要求を、ネットワーク制御だけでなく、コンピューティングリソース制御にも活用し、サービス品質を向上できると考える。例えば Web 会議を提供するクラウドのコンピューティングリソース制御を連動して行い、ユーザが求める品質に必要なコンピューティングリソースをオンデマンドで提供するなど、ユーザの要求を把握・管理する特徴は、IT サービス提供において幅広く活用が可能と考える。

筆者は、通信キャリアにおいてネットワークを中心とした研究開発に携わっているが、筑波大学大学院博士課程の営みで得られた、これまでの研究開発成果と今後のネットワークサービスへの適用に向けた検討を、今後の研究開発において活かしていきたい。

謝辞

多くのご指導と助言を頂きました木村成伴先生に深謝いたします。社会人学生として入学を希望したときから、現在に至るまで、親身になってご指導を頂き、多くの貴重なアドバイスやご意見、ご指導を賜りました。誠にありがとうございます。

学位審査にあたり、ご多忙な中、審査をお引き受けくださいました亀山啓輔先生、張勇兵先生、建部修見先生、片岸一起先生に深謝いたします。

これまでの研究開発活動の中で、様々な形でご協力いただいた歴代の上司、同僚、後輩の皆様、家族の協力に感謝いたします。

参考文献

- [1] 総務省 総合通信基盤局, 我が国のインターネットにおけるトラフィックの集計結果 (2022年5月分), 情報通信統計データベース, 2022
- [2] IETF, Multiprotocol Label Switching (mpls) working group, <https://data-tracker.ietf.org/wg/mpls/charter/>
- [3] IETF, Internet Traffic Engineering (tewg) working group, <https://data-tracker.ietf.org/wg/tewg/charter/>
- [4] IETF, RFC3272, “Overview and Principles of Internet Traffic Engineering “, May 2002.
- [5] IETF, RFC3346, “Applicability Statement for Traffic Engineering with MPLS”, August 2002.
- [6] IETF, Application-Layer Traffic Optimization (alto) working group, <https://data-tracker.ietf.org/wg/alto/charter/>
- [7] IETF, RFC3630, “Traffic Engineering (TE) Extensions to OSPF Version2”, September 2003.
- [8] IETF, RFC5305, “IS-IS Extensions for Traffic Engineering”, October 2008.
- [9] IETF, RFC4875, “IETF, RFC3212, “Constraint-Based LSP Setup using LDP”, May 2007.
- [10] IETF, RFC3212, “Constraint-Based LSP Setup using LDP”, January 2002.
- [11] A. Basu and J. Riecke: “Stability issues in OSPF routing”, 2001 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '01), pp. 225–236, San Diego, California, USA , August, 2001.
- [12] A. Khanna and J. Zinky: ”The Revised ARPANET Routing Metric”, 1989 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '89), pp. 45–56, Austin, TX, September 1989.

- [13] R. K. Boel: “Dynamic Routing in Delay Networks: Ergodicity, Transients and Large Excursions”, The 30th Conference on Decision and Control, Brighton, England, December 1991.
- [14] B. Chinoy: “Dynamics of Internet Routing Information”, 1993 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '93), pages 45–52, San Francisco, CA, September 1993.
- [15] C. Labovitz, A. Ahuja, and F. Jahanian: “Experimental Study of Internet Stability and Wide-Area Backbone Failures”, the Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing, Madison, WI, June 1999.
- [16] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian: “Delayed Internet Routing Convergence”, 2000 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '00), pages 175–187, Stockholm, Sweden, August–September 2000.
- [17] N. Rybowski and O. Bonaventure: “Evaluating OSPF Convergence with ns-3 DCE”, 2022 Workshop on ns-3 (WNS3 '22), pp. 120–126, New York, NY, USA, June 2022.
- [18] G. Apostolopoulos, R. Guérin, S. Kamat, and S. K. Tripathi: “Quality of Service Based Routing: A Performance Perspective”, 1998 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '98), pages 29–40, Vancouver, Canada, August–September 1998.
- [19] A. Shaikh, J. Rexford, and K. G. Shin: “Evaluating the Overheads of Source-Directed Quality-of-Service Routing”, the 6th IEEE International Conference on Network, Protocols, Austin, TX, October 1998.
- [20] F. Hao and E. Zegura: “On Scalable QoS Routing: Performance Evaluation of Topology Aggregation”, IEEE Infocom 2000, Tel Aviv, Israel, March 2000.
- [21] A. Shaikh, A. Varma, L. Kalampoukas, and R. Dube: “Routing Stability in Congested Networks: Experimentation and Analysis”, 2000 conference on Applications, technologies, architectures, and protocols for computer communications

- (SIGCOMM '00), pages 163–174, Stockholm, Sweden, August–September 2000.
- [22] K. Németh, A. Kőrösit and G. Rétvári: "Enriching the poor man's traffic engineering: Virtual link provisioning for optimal OSPF TE," 16th International Telecommunications Network Strategy and Planning Symposium (Networks), pp.1-7, Funchal, Portugal, September 2014.
- [23] K. Matsui, M. Kaneda, H. Takenaka, H. Ishii, M. Imase :TRES: Traffic and Reliability Engineering Server for MPLS, IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM) 2001, TP5.3, Vol.1 pp.583-586, Victoria, Canada, August (2001)
- [24] K. Matsui, T. Sakurai, M. Kaneda, J. Murayama, H. Ishii: A Multi-layered Traffic Engineering Architecture for the Electronic/Optical Hybrid Network, IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM) 2003, Vol.1 pp.293-296, Victoria, Canada, August (2003)
- [25] K. Matsui, T. Yagi, Y. Naruse, J. Murayama, K. Matsuda: Dynamic Multi-Layered Traffic Engineering Architecture for Electrical/Optical Hybrid Networks, World Telecommunications Congress/International Switching Symposium (WTC/ISS) 2004, TS09-3 (6 pages), Seoul, Korea, September (2004)
- [26] K. Matsui, M. Kaneda, K. Matsuda: Evaluation of a Server-based Traffic Engineering Architecture suitable for Large-scale MPLS Networks, Asia-Pacific Symposium on Information and Tele-communication Technologies (APSITT) 2010, B-6-1 (6pages), Kuching, Malaysia, July (2010)
- [27] IETF, RFC2702, "Requirements for Traffic Engineering Over MPLS", September 1999.
- [28] Mark Ziegelmann, "Constrained Shortest Path and Related Problems: Constrained Network Optimization", VDM Verlag, December 2007
- [29] 松井 健一, 金田 昌樹, 松田 和浩 : 大規模 MPLS ネットワーク向けサーバ型トラフィックエンジニアリング方式の評価, 電気学会 技術研究報告, CMN-11, pp.47-52, 2011 年 1 月.

- [30] 一般社団法人 電気通信事業者協会, “Telecommunications Carriers Association Annual Report”, Telecommunications Carriers Association, 2008.
- [31] 櫻井 俊之, 松井 健一, 金田 昌樹, 石井 啓之: 大規模 MPLS 網における集中制御型 TE サーバ, 電子情報通信学会技術研究報告, IN2002-284, NS2002-311, 2003 年 3 月.
- [32] IETF, RFC3376, “Internet Group Management Protocol, Version 3”, October 2002.
- [33] IETF, RFC7761, “Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)”, March 2016.
- [34] K. Matsui, M. Kaneda, H. Takenaka, H. Ichikawa: Managed IP Multicast Platform Suitable for Business Usage, IEICE Transactions on Information and Systems, Vol.E84-D No. 5, pp. 560-569, (2001)
- [35] K. Matsui, J. Aramomi, H. Ichikawa: A Managed IP Multicast Platform Suitable for Business Usage, International Conference on Computer Communication '99 (ICCC '99), T-09-03 (8 pages), Tokyo, Japan, September (1999)
- [36] 国立研究開発法人 科学技術振興機構低炭素社会戦略センター, 情報化社会の進展がエネルギー消費に与える影響 (Vol.4), 2022 年 2 月
- [37] Broadcom, Packet Shaper, <https://techdocs.broadcom.com/us/en/symantec-security-software/web-and-network-security/packetshaper/11-10.html>
- [38] K. Matsui, , M. Kaneda, M. Suzuki, H. Ishii: On-Demand QoS Management System, World Telecommunications Congress / International Switching Symposium (WTC/ISS) 2002, TS04 NM-1 (5 pages), Paris, France, September (2002)
- [39] K. Matsui, M. Kaneda, H. Ishii: Hierarchical Policy-Based Multi-User Bandwidth Broker System Suitable for Large-Scale Bandwidth On-Demand Services, Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT) 2012, SL-1-3 (6 pages), Santiago and Valparaiso, Chile, December 2012.
- [40] 川端 明夫, 青柳 雄二: APN で実現するネットワークサービス技術, NTT 技術ジ

ジャーナル, Vol.33 No.8, pp.10-14, 2021年8月.

- [41]河野 太一, 小林 正裕:映像配信サービスにおけるインテントに基づくアプリケーション・ネットワーク協調制御技術, NTT 技術ジャーナル, Vol.34 No.7, pp.44-48, 2022年7月.
- [42]菊島 宏明, 呉 超:クラウドサーバリソース最適化による快適な Web 会議サービスの実現, NTT 技術ジャーナル, Vol.34 No.7, pp.39-43, 2022年7月.

発表論文

査読付き学術論文

- [1] K. Matsui, M. Kaneda, H. Takenaka, H. Ichikawa:Managed IP Multicast Platform Suitable for Business Usage, IEICE Transactions on Information and Systems, Vol.E84-D No. 5, pp. 560-569, 2001

査読付き国際会議

- [1] K. Matsui, M. Kaneda, H. Takenaka, H. Ishii, M. Imase :TRES: Traffic and Reliability Engineering Server for MPLS, IEEE Pacific Rim Conference on Communications, Computers and Signal Processing 2001, TP5.3, Vol.1 pp.583-586, Victoria, Canada, August 2001
- [2] K. Matsui, T. Sakurai, M. Kaneda, J. Murayama, H. Ishii: A Multi-layered Traffic Engineering Architecture for the Electronic/Optical Hybrid Network, IEEE Pacific Rim Conference on Communications, Computers and Signal Processing 2003, Vol.1 pp.293-296, Victoria, Canada, August 2003
- [3] K. Matsui, T. Yagi, Y. Naruse, J. Murayama, K. Matsuda: Dynamic Multi-Layered Traffic Engineering Architecture for Electrical/Optical Hybrid Networks, World Telecommunications Congress/International Switching Symposium 2004,

TS09-3 (6 pages), Seoul, Korea, September 2004

- [4] K. Matsui, M. Kaneda, K. Matsuda: Evaluation of a Server-based Traffic Engineering Architecture suitable for Large-scale MPLS Networks, Asia-Pacific Symposium on Information and Telecommunication Technologies 2010, B-6-1 (6pages), Kuching, Malaysia, July 2010
- [5] K. Matsui, J. Aramomi, H. Ichikawa: A Managed IP Multicast Platform Suitable for Business Usage, International Conference on Computer Communication, T-09-03 (8 pages), Tokyo, Japan, September 1999
- [6] K. Matsui, , M. Kaneda, M. Suzuki, H. Ishii: On-Demand QoS Management System, World Telecommunications Congress / International Switching Symposium 2002, TS04 NM-1 (5 pages), Paris, France, September 2002
- [7] K. Matsui, M. Kaneda, H. Ishii: Hierarchical Policy-Based Multi-User Bandwidth Broker System Suitable for Large-Scale Bandwidth On-Demand Services, Asia-Pacific Symposium on Information and Telecommunication Technologies 2012, SL-1-3 (6 pages), Santiago and Valparaiso, Chile, December 2012

紀要

- [1] K. Matsui, T. Yagi, Y. Naruse, and J. Murayama: Cut-through Optical Path Control Technology for a Terabit-class Super-network, NTT Technical Review, Vol.2 No.3, pp.32-40, March (2004)
- [2] J. Murayama, K. Matsui, K. Matsuda, and M. Makino: Conceptual Design for a Terabit-class Super-networking Architecture, NTT Technical Review, Vol.2 No.3, pp.12-20, March (2004).
- [3] Y. Naruse, T. Yagi, K. Matsui, and J. Murayama: IP-in-IPv6 Overlay Networking Technology for a Terabit-class Super-network, NTT Technical Review, Vol.2 No.3, pp.21-31, March (2004).

- [4] T. Yagi, K. Matsui, Y. Naruse, and J. Murayama: Cut-through IP Forwarding Technology for a Terabit-class Super-network, NTT Technical Review, Vol.2 No.3, pp.41-49, March (2004).
- [5] 成瀬 勇一, 八木 毅, 松井 健一, 村山 純一, 松田 和浩, 今宿 亘, 滝川 好比郎: 光パス制御技術の動向, NTT 技術ジャーナル, Vol.13 No.12, pp.38-41, 2004年10月.

国内会議発表

- [1] 松井 健一, 橋本 貴志, 新籾 純, 市川 弘幸: 大規模 IP マルチキャスト通信における共有木コストの一考察, 電子情報通信学会 1998年 総合大会, B-7-129, 1998年3月.
- [2] 松井 健一, 田中 延昭, 丹羽 恵吾, 新籾 純, 市川 弘幸: 送信者保護・受信者制限つきマルチキャストサービス提供方式の検討, 電子情報通信学会 1999年 総合大会, B-7-149, 1999年3月.
- [3] 松井 健一, 渡瀬 順平, 金田 昌樹, 田中 延昭, 市川 弘幸: コネクションレス通信におけるトラヒックを考慮した迂回経路制御方式, 電子情報通信学会 2000年 総合大会, B-7-103, 2000年3月.
- [4] 渡瀬 順平, 羽柴 正治, 松井 健一, 金田 昌樹, 市川 弘幸: 高付加価値サービス提供型 CL-ATM 網における迂回経路制御方式, 電子情報通信学会 2000年 総合大会, B-7-91, 2000年3月.
- [5] 松井 健一, 金田 昌樹, 竹中 光, 市川 弘幸: 大規模 IP ネットワークにおけるマルチキャストルーティングプロトコル適用の検討, 情報処理学会 2000年 全国大会, 1G-12, 3-317, 2000年10月.
- [6] 松井 健一, 金田 昌樹, 竹中 光: トラヒックエンジニアリングにおける経路計算方式の検討, 電子情報通信学会 2001年 総合大会, B-7-155, 2001年3月.

- [7] 田島 佳武, 飯盛 可織, 松井 健一, 村山 純一: 広域ネットワークサービスプラットフォームにおけるトラヒックエンジニアリング方式, 電子情報通信学会 技術研究報告, IN2000-224, SSE2000-268, 2001 年 3 月.
- [8] 松井 健一, 金田 昌樹, 石井 啓之: トラヒックエンジニアリング技術の実装について, 電子情報通信学会 技術研究報告, TM2001-22, 2001 年 7 月.
- [9] 松井 健一, 金田 昌樹, 飯盛 可織, 田島 佳武, 石井 啓之: 品質制御方式とトラヒックエンジニアリング方式の連携, 電子情報通信学会 2001 年 ソサイエティ大会, B-7-113, 2001 年 9 月.
- [10] 田島 佳武, 松井 健一, 村山 純一: GNSP における二階層型トラヒックエンジニアリング方式, 電子情報通信学会2001 年 ソサイエティ大会, B-7-35, 2001 年 9 月.
- [11] 飯盛 可織, 松井 健一, 金田 昌樹, 石井 啓之: GNSP における分散サーバ型帯域管理方式, 電子情報通信学会 2001 年 ソサイエティ大会, B-7-37, 2001 年 9 月.
- [12] 松井 健一, 金田 昌樹, 鈴木 宗良, 石井 啓之: オンデマンドフロー管理システム, 電子情報通信学会技術研究報告, NS2001-239, IN2001-195, 2002 年 3 月.
- [13] 松井 健一, 金田 昌樹, 鈴木 宗良, 石井 啓之: オンデマンドフロー管理システムの検討, 電子情報通信学会 2002 年 総合大会, B-7-115, 2002 年 3 月.
- [14] 村山 純一, 鈴木 耕世, 岡 大祐, 松井 健一, 長津 尚英: VNSP におけるフォトニック-IPv6 連携トラヒックエンジニアリング方式, 電子情報通信学会 2002 年 総合大会, B-7-123, 2002 年 3 月.
- [15] 松井 健一, 櫻井 俊之, 金田 昌樹, 村山 純一, 石井 啓之: テラビット級スーパーネットワークにおけるマルチレイヤトラヒックエンジニアリングの検討, 電子情報通信学会 技術研究報告, IN2003-102, 2003 年 2 月.
- [16] 松井 健一, 櫻井 俊之, 金田 昌樹, 村山 純一, 石井 啓之: TSN におけるカットス

ルー光パス割当方式の設計，電子情報通信学会 2003 年 総合大会，B-7-84，2003 年 3 月．

[17] 村山 純一，八木 毅，辻元 孝博，櫻井 俊之，松井 健一，住本 順一，金田 昌樹，松田 和浩，石井 啓之：テラビット級スーパーネットワーク(TSN)の研究開発，電子情報通信学会 2003 年 総合大会，B-7-81，2003 年 3 月．

[18] 辻元 孝博，八木 毅，松井 健一，村山 純一，松田 和浩，石井 啓之：TSN における光カットスルー方式の評価，電子情報通信学会 2003 年 総合大会，B-7-82，2003 年 3 月．

[19] 櫻井 俊之，松井 健一，金田 昌樹，村山 純一，石井 啓之：TSN における光パス制御方式の評価，電子情報通信学会 2003 年 総合大会，B-7-83，2003 年 3 月．

[20] 松井 健一，櫻井 俊之，金田 昌樹，村山 純一，石井 啓之：テラビット級スーパーネットワークにおけるマルチレイヤトラヒックエンジニアリングの検討，電子情報通信学会 技術研究報告，IN2002-289，NS2002-316，2003 年 3 月．

[21] 櫻井 俊之，松井 健一，金田 昌樹，石井 啓之：大規模 MPLS 網における集中制御型 TE サーバ，電子情報通信学会技術研究報告，IN2002-284，NS2002-311，2003 年 3 月．

[22] 櫻井 俊之，松井 健一，金田 昌樹，石井 啓之：アプリケーション連動型フロー管理システムの検討，電子情報通信学会 2002 年 ソサイエティ大会，B-7-36，2003 年 8 月．

[23] 松井 健一，八木 毅，金田 昌樹，成瀬 勇一，村山 純一：TSN におけるカットスルー光パス配置アルゴリズムの検討，電子情報通信学会 2003 年 ソサイエティ大会，B-7-64，2003 年 9 月．

[24] 八木 毅，松井 健一，村山 純一，松田 和浩，牧野 将哉：TSN におけるレイヤ間連

携カッタスルー経路制御方式, 電子情報通信学会 2003 年 ソサイエティ大会,
B-7-62, 2003 年 9 月.

[25] 松井 健一, 八木 毅, 金田, 成瀬 勇一, 村山 純一: テラビット級スーパーネットワークにおけるカッタスルー光パス方式の検討, 電子情報通信学会 技術研究報告 IN2003-82, NS2003-116, CS2003-91, 2003 年 9 月. s

[26] 八木 毅, 松井 健一, 村山 純一, 松田 和浩 テラビット級スーパーネットワークにおけるカッタスルーIP 経路制御方式の評価, 電子情報通信学会 技術研究報告, N2003-83, NS2003-117, CS2003-92, 2003 年 9 月.

[27] 松井 健一, 成瀬 勇一, 八木 毅, 村山 純一, 松田 和浩: TSN におけるネットワーク制御サーバ分散方式の検討, 電子情報通信学会 2004 年 総合大会, B-7-91, 2004 年 3 月.

[28] 八木 毅, 松井 健一, 成瀬 勇一, 村山 純一, 松田 和浩: TSN におけるユーザ要求を考慮したカッタスルー光パス制御方式, 電子情報通信学会 2004 年 総合大会, B-7-93, 2004 年 3 月.

[29] 成瀬 勇一, 松井 健一, 八木 毅, 村山 純一 TSN における光パス割当ポリシーのフレームワーク設計, 電子情報通信学会 2004 年 総合大会, B-7-94, 2004 年 3 月.

[30] 村山 純一, 八木 毅, 松井 健一, 成瀬 勇一, 松田 和浩, 牧野 将哉: テラビット級スーパーネットワークのアーキテクチャ設計, 電気学会技術研究報告, CMN-04-17, 2004 年 6 月.

[31] 松井 健一, 八木 毅, 成瀬 勇一, 村山 純一, 松田 和浩: TSN におけるコスト更新頻度に基づく経路計算方式, 電子情報通信学会 2004 年 ソサイエティ大会, B-7-17, 2004 年 9 月.

[32] 八木 毅, 松井 健一, 村山 純一, 松田 和浩, 牧野 将哉: TSN におけるパケットの

到着順序逆転を防止した経路制御方式, 電子情報通信学会 2004 年 ソサイエティ大会, B-7-16, 2004 年 9 月.

[33] 八木 毅, 松井 健一, 成瀬 勇一, 村山 純一, 松田 和浩: トラヒックエンジニアリングにおける動的フローリダイレクション方式, 電子情報通信学会 技術研究報告 IN2004-180, 2005 年 2 月.

[34] 八木 毅, 松井 健一, 成瀬 勇一, 村山 純一, 松田 和浩: TSN におけるリアルタイムトラヒックエンジニアリング技術, 電子情報通信学会 2005 年 総合大会, B-7-98, 2005 年 3 月.

[35] 成瀬 勇一, 松井 健一, 八木 毅, 村山 純一, 松田 和浩: TSN における IP/光レイヤ間連携トラヒックエンジニアリング, 電子情報通信学会 2005 年 総合大会, B-7-99, 2005 年 3 月.

[36] 八木 毅, 松井 健一, 成瀬 勇一, 村山 純一, 松田 和浩: テラビット級スーパーネットワークにおけるデマンドベースカットスルー光パス制御方式, 電子情報通信学会 技術研究報告, IN2004-286, NS2004-286, 2005 年 3 月.

[37] 松井 健一, 金田 昌樹, 松田 和浩: 大規模 MPLS ネットワーク向けサーバ型トラヒックエンジニアリング方式の評価, 電気学会 技術研究報告, CMN-11, pp.47-52, 2011 年 1 月.