

帯域幅比率に応じたネットワーク負荷分散による複数経路伝送手法

西村 閣晋^{1,a)} 佐藤 健哉^{1,b)}

概要：近年、モバイルでのインターネットを利用したトラフィック量が増加している。その背景として、スマートフォンの普及によるアプリケーションゲームや動画といった大容量サービスの成長が著しいことが考えられる。このトラフィック量の増加に伴い、ISP(Internet Service Provider)内の通信負荷増大による通信障害が懸念される。そのため膨大な量のトラフィックがある状況下でもアプリケーションが利用できるようなネットワークの輻輳を抑制する仕組みが必要となっている。対策として挙げられるのがISP内のルーティングプロトコルの改善である。既存のルーティングプロトコルは、最適経路を用いた単一経路制御であるため、リンクやノードにトラフィックが集中し、輻輳のリスクが高まってしまうといった欠点がある。この問題に対する解決策として、本研究では、パケット転送時、最適経路以外の冗長経路に対して帯域幅比率に応じた分散転送による複数経路制御を行う。これによりネットワーク全体の利用率向上を見込め、輻輳の抑制が期待できる。

キーワード：経路制御, OpenFlow, 帯域幅

Multipath Transmission Method by Network Load Balancing According to Bandwidth Ratio

Abstract: In recent years, the amount of traffic using the mobile Internet has increased. As the background, the growth of large-capacity services such as application games and videos due to the spread of smartphones is considered. With this increase in traffic volume, there is concern about communication failure due to increased communication load in ISP (Internet Service Provider). Therefore, there is a need for a mechanism that suppresses network congestion so that applications can be used even in situations where there is a large amount of traffic. One countermeasure is the improvement of the routing protocol in the ISP. The existing routing protocol is a single route control using the optimum route, so traffic is concentrated on links and nodes, and the risk of congestion increases. As a solution to this problem, in this study, multiple routes are controlled by distributed transfer according to the bandwidth ratio for redundant routes other than the optimal route during packet transfer. As a result, the utilization rate of the entire network can be expected and congestion can be suppressed.

Keywords: Routing, OpenFlow, Bandwidth

1. はじめに

近年、モバイルでのインターネットを利用したトラフィック量が増加している。その背景として、スマートフォンの

普及によるアプリケーションゲームや動画といった大容量サービスの成長が著しいことが考えられる。世界のモバイルデータ量は2020年までに現在のモバイルデータ量と比較すると約1.5倍になると予想されている[1]。このトラフィック量の増加に伴い、ISP(Internet Service Provider)内の通信負荷増大による通信障害が懸念される。実際に、国内のISPにおいて遅延やパケット損失による通信障害の事例が報告されている[2][3]。そのため膨大な量のトラ

¹ 同志社大学大学院理工学研究科
Doshisha University Graduate School of Science and Engineering

a) kakushin.nishimura@nislabs.doshisha.ac.jp

b) ksato@mail.doshisha.ac.jp

フィックがある状況下でも、アプリケーションが利用できるようなネットワークの輻輳を抑制する仕組みが必要となっている。特に、交通や決済関連のアプリケーションをスムーズに利用できることが重要であると考え、上記問題の対策として挙げられるのがISP内におけるルーティングプロトコルの改善である。既存のルーティングプロトコルは、最適経路を用いた単一経路制御であるため、リンクやノードにトラフィックが集中し、輻輳のリスクが高まってしまうといった欠点がある。

この問題に対する解決策として、最適経路以外の宛先ノードに通じる冗長経路を活用し、複数の経路を伝送に使用する複数経路制御が考えられる [4]。これによりネットワーク全体の使用率向上を見込め、輻輳の抑制が期待できる。

2. 関連研究

2.1 OSPF Optimized Multipath

Curtis Villamizar の研究では、OSPF(Open Shortest Path First)[5] を拡張した OSPF-OMP(Optimized Multipath) というリアルタイムトラフィックを考慮した複数経路制御を提案している [6]。各ルータが定期的にネットワーク全体のリンクの負荷状態を収集しながら、最大流問題を解くようにして経路制御を行なっている。各ルータは OSPF により等コスト複数経路を算出し、各経路のリンク負荷状態が均一になるようにパケット伝送確率を調整することで、動的な複数経路制御を実現している [7]。しかし、ネットワーク上に各ルータのリンクごとの負荷情報パケットが混在するため、帯域を逼迫する。さらに、負荷が均一となるようパケット伝送確率の調整を行うため、経路収束が極めて遅く、パケットのスループットが低下してしまうといった問題点がある [8]。

2.2 OpenFlow を用いた TCP トラフィック動的経路分散ネットワークの提案

吉村らの研究ではダイクストラ法を改良し、最適経路からプラス 3 ホップまでの複数の経路を通信路として使用している。また、ルーティング手法としては、現在の各リンクの負荷情報を元に経路の再選択を OpenFlow[9] を用いて行っており、動的な複数経路制御を実現している [10]。

しかし、負荷状況に応じた動的経路制御を行うために、フローエントリを消去し、逐次経路の再計算を行っている。このため経路探索における OpenFlow コントローラ上での計算コストが大きく、パケットのスループットが低下してしまうといった問題点がある。

3. 提案手法

3.1 概要

本研究では、リンクの帯域幅比率に応じて複数経路にパ

ケットを分散転送することで、計算量を抑えた複数経路伝送手法を提案する。さらに、OpenFlow を用いて、リアルタイムトラフィックを考慮した適切なタイミングでの複数経路制御を行うことで、輻輳発生時でない状況下におけるスループットの最大化も検討する。

システムの概要を図 1 に示す。まず、OpenFlow コントローラがネットワークトポロジの把握と各ルータのポートごとの帯域幅を取得する。続いてルーティング手法は、単一経路制御による最適経路を伝送路として採用する。ただし、上記で記述したように、適切なタイミングで本提案手法での複数経路制御を行う。適切なタイミングとは、単一経路制御における最適経路上のボトルネックリンクの帯域幅使用率が閾値以上の状況下を指す。ここでの帯域幅使用率は、OpenFlow コントローラが一定時間毎に各ルータに問い合わせることで情報を取得する。

続いて、本提案手法における複数経路制御の方法について説明する。ボトルネックリンクの帯域使用率が閾値以上となると、OpenFlow コントローラは該当する最適経路のフローエントリをルータに破棄させる。その後、宛先までの複数経路を算出し、複数フローエントリを作成する。この際、フローエントリに対して各経路のボトルネックリンクの帯域幅比率に応じた重み付けを行う。これにより、帯域幅比率に応じた複数経路制御を行う。

3.2 OpenFlow コントローラ

本研究における、OpenFlow コントローラの機能について以下に示す。

- 経路の格納
最適経路制御と複数経路制御で計算した経路情報を、経路切り替え時に再計算を行わないために OpenFlow コントローラにキャッシュする。
- 最適経路制御
ルータからのパケット転送先の問い合わせに対して、宛先までの経路にかかるコストがもっとも最小となる最適経路を探索し、フローエントリを返す。
- 帯域使用率の問い合わせ
各ルータに対して、一定時間毎にリンクの帯域使用率を問い合わせる。ボトルネックリンクの帯域使用率が閾値を超えていた場合、複数経路制御機能を呼び出す。
- 複数経路制御
複数経路の算出を行う。その後、各経路のボトルネックリンクの帯域幅比率に応じて重み付けを行なったグループテーブル、およびフローエントリを返す。

3.3 動作手順

以下に提案手法の動作手順を述べる。また図 2 にシーケンス図を示す。なお、シーケンス図中の番号と、以下の動作手順の番号の処理は対応している。

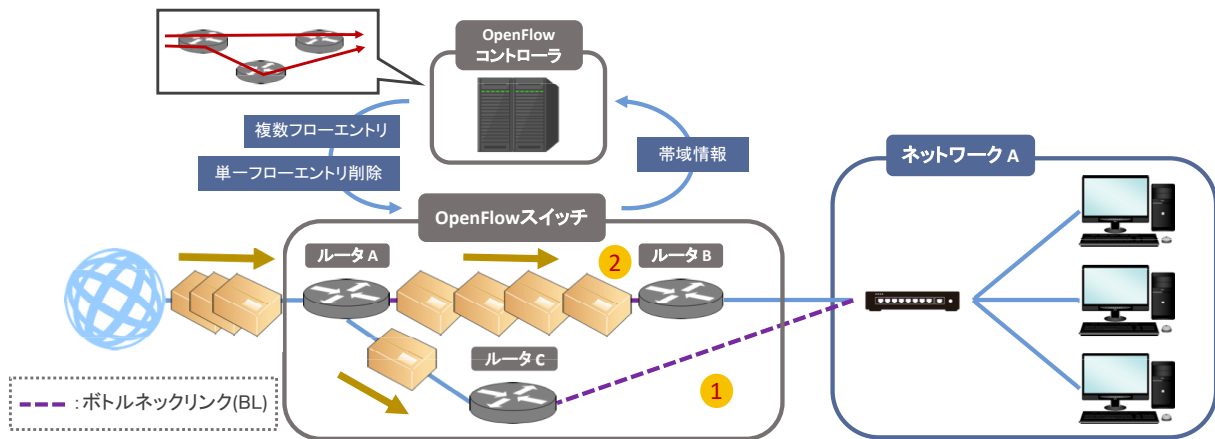


図 1 システム概要

Fig. 1 System overview

- (1) ルータは OpenFlow コントローラにパケット転送先を要求
- (2) OpenFlow コントローラは最適経路を探索し、ルータにフローエントリを送信
- (3) OpenFlow コントローラはルータに各リンクの帯域使用率を要求
- (4) ルータは OpenFlow コントローラに各リンクの帯域使用率を送信
- (5) 帯域使用率が閾値を超えていた場合、OpenFlow コントローラは複数経路を探索
- (6) OpenFlow コントローラはルータにフローエントリの削除とグループテーブルおよび、フローエントリを送信
- (7) ルータは最適経路のフローエントリを削除し、受信したグループテーブル、フローエントリを適用

4. 実装

4.1 概要

本研究の実装は、離散事象ネットワークシミュレータである ns-3.29[11] を用いて行う。また、帯域幅比率に応じたネットワーク負荷分散には、OpenFlow1.1 から採用されたフローエントリのグループ化の概念を利用する。グループエントリ中にある、複数アクションから select タイプを用いることで1つのアクションの決定を行う。アクションとは、ルータの複数経路制御における各送出ポートへの送信処理のことを指す。このアクション選択時、重み (weight) として複数経路の各ボトルネックリンクの帯域幅比率を設定することで、提案手法のネットワーク負荷分散を実現する。

4.2 OFSWITCH13

ns-3 の拡張ライブラリとしてオープンソースで提供されている OFSWITCH13[12] を用いる。このライブラリは、

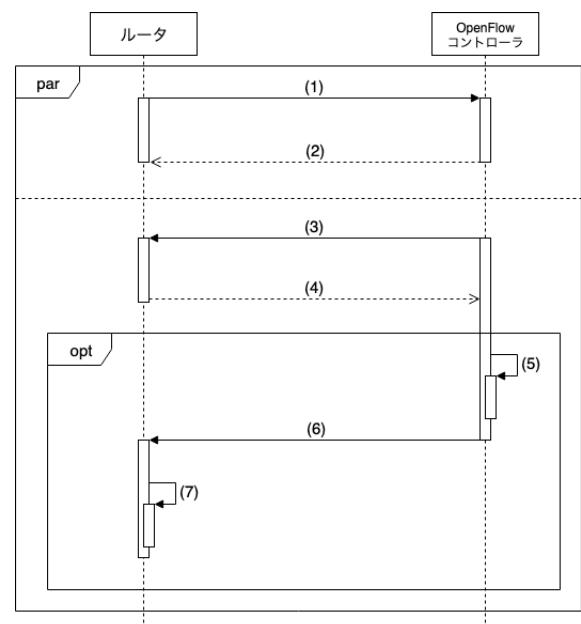


図 2 シーケンス図

Fig. 2 Sequence diagram

OpenFlow1.3 プロトコルを ns-3 上で実現するためのコントローラとスイッチを提供する。以下に、帯域幅取得とパケット負荷分散時に用いたモジュールを記載する。

- 帯域幅取得

OpenFlow1.3 から各スイッチのポート情報の取得には、マルチパートフレームワークを用いる。OpenFlow コントローラがマルチパートリクエストとして OF-PMP_PORT_DESC タイプを要求すると、OpenFlow スイッチに属するポート情報がレスポンスとして返される。これにより、OFSWITCH13 の HandleMultipartReply ハンドラーにより、マルチパートリプライパケット中のポート情報を取得し、スイッチの各ポートの帯域幅情報を取得する。

- ネットワーク負荷分散

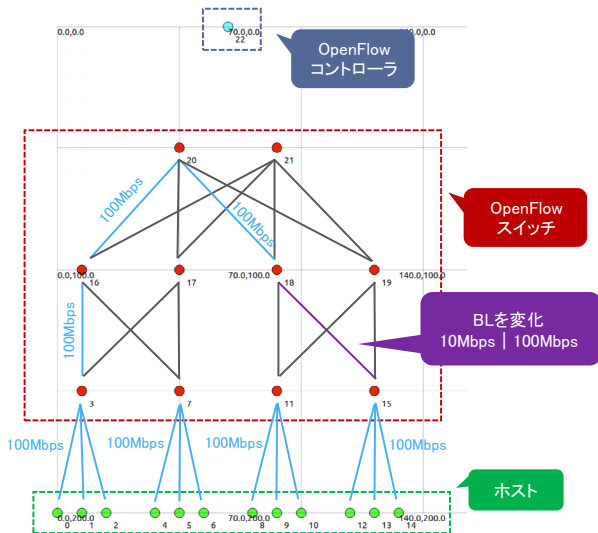


図 3 評価シナリオ
Fig. 3 Evaluation scenario

送信元、宛先 IP アドレスから複数経路をダイクストラ法によって算出する．続いて，OpenFlow コントローラへ経路のキャッシュを行った上で，OFSWITCH13 の CreateGroupMod メソッドによりグループエントリの作成を行う．この際，グループエントリのタイプは select とする．最後に，先ほど作成したグループエントリのグループ ID を設定したフローエントリを OFSWITCH13 の CreateFlowMod メソッドにより作成する．

4.3 シナリオ環境

本実験では大容量データ通信を想定する．図 3 に評価シナリオを示し，その構成要素を表 1 に示す．評価シナリオのトポロジ形態は，現在多くの ISP のトポロジ構成として採用されている Fat-Tree ネットワーク構成を用いる [13]．また，リンク帯域幅について，OSPF のコスト計算上 100Mbps 以上の帯域幅におけるリンクのコスト値は 1 となるため，10Mbps と 100Mbps の 2 つの帯域幅を採用した．

ノード 0, 2 番ホストから，ノード 13 番ホストへ，ノード 8, 10 番ホストから，ノード 5 番ホストへそれぞれトラフィックを発生させる．トラフィック内容は，1 パケットあたりのペイロードが MSS(Maximum Segment Size) の最大値である 1460Byte の UDP(User Datagram Protocol) パケットを用いる．さらに，トポロジ上の一部リンクの帯域幅を 10Mbps と 100Mbps のそれぞれに変化させる．

5. 評価

5.1 概要

比較対象には，現在も広く利用されている OSPF を選定する．なお，本稿における OSPF とは OSPFv2 のこと

表 1 シナリオ構成要素
Table 1 Scenario components

OpenFlow コントローラ	1 台
OpenFlow スイッチ	10 台
ホスト	12 台
リンク帯域幅	10Mbps, 100Mbps
シミュレーション時間	10s

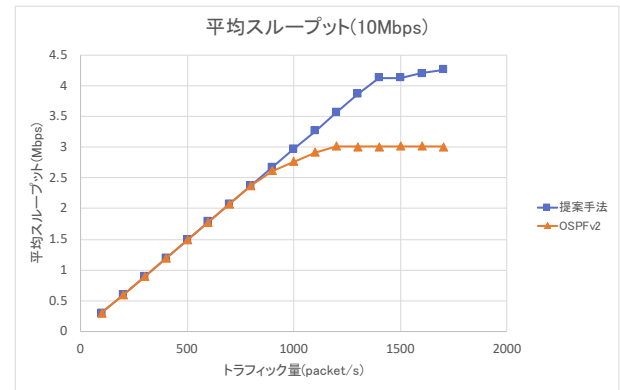


図 4 平均スループット (10Mbps)
Fig. 4 Average throughput(10Mbps)

を指し，OSPF-ECMP(Equal-Cost Multipath)[14]を用いたルーティングを行う．比較項目は，単位時間あたりのトラフィック量に対する平均スループットと平均パケット損失率とする．パケット生成に関しては，1 秒間に生成するパケットを 100～1700 の間で変化させる．また，「平均スループット」と「平均パケット損失率」とは，4.3 節で説明した各通信におけるスループットとパケット損失率の平均値のことである．さらに，最適経路のボトルネックリンクの帯域幅を 10Mbps と 100Mbps それぞれに変化させて，2 パターンの評価を行う．これは，ボトルネックリンクの帯域幅比率に応じた負荷分散を行う際，帯域幅比率に差があると評価結果に影響を及ぼすのかを調べるためである．

5.2 性能評価

● スループット

ボトルネックリンクが 10Mbps 時の平均スループットのグラフを図 4 に示す．提案手法は，OSPF と比較して最大スループットが約 1.3 倍となっている．このことから，提案手法は OSPF に比べてネットワーク全体の性能を有効に活用できていることがわかる．次に，ボトルネックリンクが 100Mbps 時の平均スループットのグラフを図 5 に示す．ボトルネックリンクが 10Mbps 時の平均スループットとほとんど変わらないことがわかる．

● パケット損失率

ボトルネックリンクが 10Mbps 時の平均パケット損失率のグラフを図 6 に示す．提案手法は，OSPF と比較

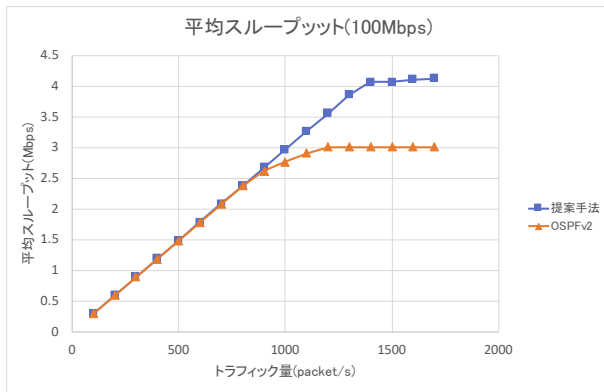


図 5 平均スループット (100Mbps)
Fig. 5 Average throughput(100Mbps)

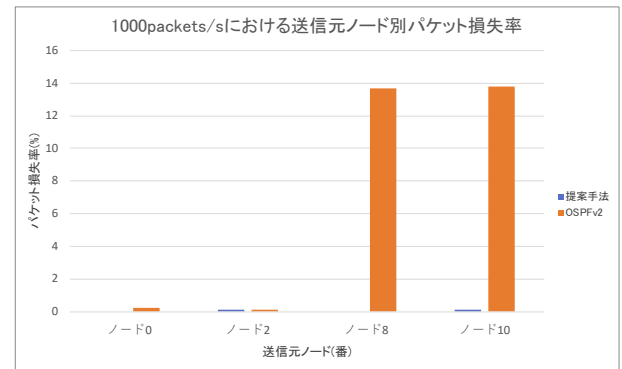


図 8 送信ノード別パケット損失率 (1000packet/s)
Fig. 8 Packet loss rate by sending node(1000packet/s)

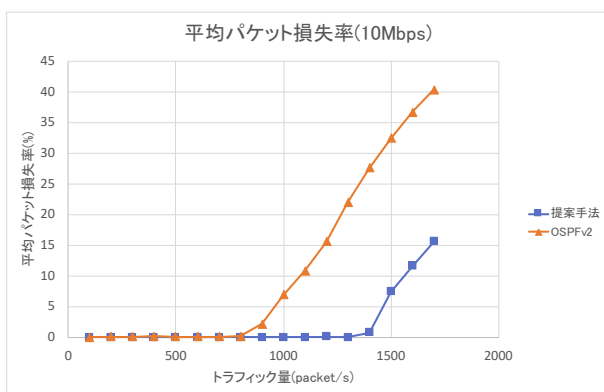


図 6 平均パケット損失率 (10Mbps)
Fig. 6 Average packet loss rate(10Mbps)

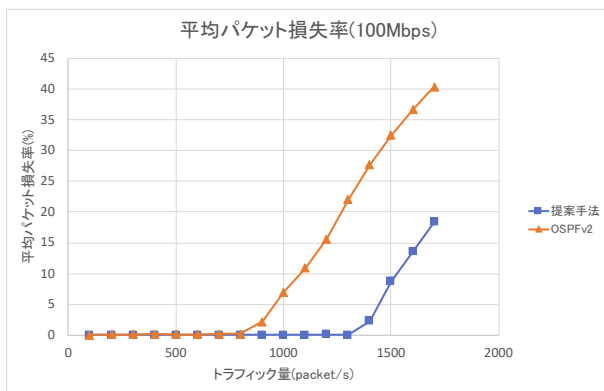


図 7 平均パケット損失率 (100Mbps)
Fig. 7 Average packet loss rate(100Mbps)

していずれのパケット生成速度においても低い損失率を記録している。これは提案手法では冗長経路を活用しているため、ネットワーク内で保持できるパケット数が増加したためだと考えられる。次に、ボトルネックリンクが 100Mbps 時の平均パケット損失率のグラフを図 7 に示す。ボトルネックリンクが 10Mbps 時の平均パケット損失率とほとんど変わらないことがわかる。

6. 考察

提案手法では冗長経路を用いたネットワーク負荷分散を行っている。一方、OSPF では宛先ノードまでのコスト値が等しい経路では負荷分散を行うが、基本的には最適経路での単一経路制御を行う。つまり、既存手法と比べると活用しているルータ数が多いことがわかる。本実験では、提案手法がノード 17, 19, 21 番のルータを効率的に利用したルーティングを行っていることを観測した。これにより、ネットワーク全体でパケットのバッファ可能な容量が既存手法と比べて多いと推測できる。したがって、パケット損失率が低下する。さらに、これによりルータが効率良く稼働し、スループットが向上したと考えられる。さらに、最適経路におけるボトルネックリンクの帯域幅を 10 倍に変更しても、提案手法の変更前の性能と比較して、変化はほとんどない。このことから、帯域幅比率に応じた静的負荷分散により、効率的にネットワーク性能を引き出せていると考えられる。また、図 8 にトラフィック量が 1000packets/s における各ノードの通信時のパケット損失率を示す。OSPF ではノード 8, 10 番ホストの送信パケットの損失率が高い。他の送信元ノードのパケット損失率と比較すると、ノード 8, 10 番ホストの送信パケットの到達性が低いことがわかる。これは、ボトルネックとなるルータに遠いノードの送信パケットほど損失する可能性が高いためである。本実験では、ノード 20 番のルータがボトルネックとなる。ノード 0, 2 番ホストからノード 20 番までのコスト値は 3 であるのに対し、ノード 8, 10 番ホストからノード 20 番までのコスト値は 12 である。つまり、ノード 0, 2 番ホストからの送信パケットの方が、ノード 8, 10 番ホストからの送信パケットに比べてノード 20 番のルータにバッファされる可能性が高い。したがって、ノード 0, 2 番ホストの送信パケットの損失率は低くなり、ノード 8, 10 番ホストに比べてパケットの到達率が高いと考えられる。

さらに、図 9 にトラフィック量が 1700packets/s にお

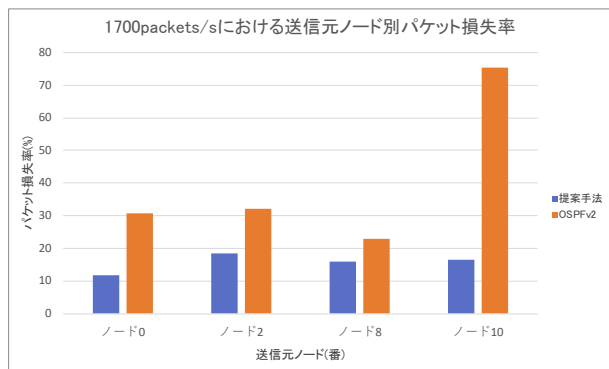


図 9 送信ノード別パケット損失率

Fig. 9 Packet loss rate by sending node(1700packet/s)

る各ノードの通信時のパケット損失率を示す。提案手法では、各ノードの送信パケットの損失率に大きな差はなく、どのノードの送信パケットも宛先ノードへ均一に到達していることがわかる。

7. まとめ

近年のネットワークトラフィック量の増加に伴い、既存のルーティングプロトコルにおける輻輳のリスクについて言及した。既存研究では、複数経路を用いた動的経路制御手法が考えられているが計算コストの面で実用的ではない。本研究では、複数経路を用いた負荷分散を動的ではなく、リンク帯域幅比率に応じて静的に行うことで計算コストを抑え、スループットの向上を目指した。提案手法と OSPF を比較した結果、スループットとパケット損失率において提案手法が優位であることを示した。さらに、提案手法によるルーティングでは、ホストごとのパケット到達率に偏りが生じにくいと考えられる。

参考文献

- [1] 総務省, "総務省 | 平成 29 年版 情報通信白書 | データ流通量の爆発的拡大", <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/html/nc121210.html>, (accessed 2019-10-23).
- [2] SAKURA internet, "メンテナンス・障害情報・機能追加 | さくらインターネット公式サポートサイト", <https://help.sakura.ad.jp/maint/24961/>, (accessed 2019-10-26).
- [3] BANDAI NAMCO, "10月5日 ネットワーク障害のお知らせ - 機動戦士ガンダムオンライン - バンダイナムコオンライン", <https://msgo.bandainamco-ol.jp/member/information/article.php?no=2252&tag=Trouble>, (accessed 2019-10-26).
- [4] Jiayue He, Jennifer Rexford, "Toward internet-wide multipath routing", IEEE, vol.22, pp.16-21, (2008).
- [5] Ascend Communications, Inc., "RFC 2328 - OSPF Version 2", IETF, <https://tools.ietf.org/html/rfc2328>, (accessed 2019-10-22).
- [6] Curtis Villamizar, "draft-ietf-ospf-omp-02 - OSPF Optimized Multipath (OSPF-OMP)", IETF, <https://tools.ietf.org/html/draft-ietf-ospf-omp-02>, (accessed 2019-10-22).

- [7] G.Michael Schneider, Tamas Nemeth, "A simulation study of the OSPF-OMP routing algorithm", Computer Networks, vol.39, pp.457-468, (2002).
- [8] 可児友邦, "通信状況に応じて柔軟に経路再選択を行うマルチパスルーティングの提案", 北陸先端科学技術大学院大学, <https://dspace.jaist.ac.jp/dspace/bitstream/10119/15216/7/paper.pdf>, (accessed 2019-10-22).
- [9] E.Haleplidis, Ed, "RFC 7426 - Software-Defined Networking (SDN)", IETF, <https://tools.ietf.org/html/rfc7426>, (accessed 2019-10-22).
- [10] 吉村悠, 佐藤健哉, "OpenFlow を用いた TCP トラフィック動的経路分散ネットワークの提案", マルチメディア, 分散協調とモバイルシンポジウム 2016 論文集, vol.2016, pp.1403-1409, (2016).
- [11] ns-3 Project, "ns-3", <https://www.nsnam.org/>, (accessed 2019-10-22).
- [12] Luciano Jerez Chaves, "OpenFlow 1.3 module for ns-3 -Irc - Unicamp", the Computer Networks Laboratory of the University of Campinas (Unicamp), <http://www.lrc.ic.unicamp.br/ofswitch13/>, (accessed 2019-10-22).
- [13] Y.Li, D.Pan, "OpenFlow based load balancing for Fat-Tree networks with multipath support", Proc. 12th IEEE International Conference on Communications (ICC' 13), pp.1-5, (2013).
- [14] D.Thaler, "RFC 2991 - Multipath Issues in Unicast and Multicast Next-Hop Selection", IETF, <https://tools.ietf.org/html/rfc2991>, (accessed 2019-10-22).