

# 仮想化ノード FLARE によるアプリケーション毎の SDN 制御手法の実装と評価

平久 紬<sup>1</sup> 中尾 彰宏<sup>2</sup> 山本周<sup>2</sup> 山口 実靖<sup>3</sup> 小口 正人<sup>1</sup>

**概要:** 現在のインターネットは、多種多様なアプリケーションのトラフィックが混在しているため、大地震などの緊急災害時のネットワーク輻輳では、すべてトラフィックが同一に扱われ緊急情報に中々アクセスできないという問題が生じる。また、トラフィック監視のみのネットワーク運用では、外部要因に基づく対局的緊急障害対応が難しい。そこで、ソーシャル・データから実社会の急激な状況変化を検出し、自動でユーザが必要とする情報を提供するアプリケーションを優先的に提供する仕組みが必要と考える。本研究では自動制御を実現するプラットフォームとして FLARE を使用し、SNS 情報に基づく実際のアプリケーションを用いたアプリケーション毎の SDN 制御実験を行うことにより、本システムの有効性を示す。

## Implementation and Evaluation of Application-specific SDN Control Method Using Virtualization node FLARE

TSUMUGI TAIRAKU<sup>1</sup> AKIHIRO NAKAO<sup>2</sup> SHU YAMAMOTO<sup>2</sup> SANEYASU YAMAGUCHI<sup>3</sup>  
MASATO OGUCHI<sup>1</sup>

### 1. はじめに

近年、モバイル端末の高機能化やクラウドコンピューティングの発達により、現在のインターネットは、多種多様なアプリケーションのトラフィックが混在している。このようなネットワークでは、トラフィックは実社会の状況により大きく変動する。このことから、大地震などの災害時のネットワーク輻輳では、すべてのトラフィックが同一に扱われユーザが必要とする緊急情報に中々アクセスできないという問題が生じる。例えば、電話など情報交換をリアルタイムに行えるアプリケーションが災害時にはユーザにとっての重要度が高いが、帯域を多く使用する YouTube などの動画やゲームなどの娯楽目的のアプリケーションのトラフィックは緊急時は優先度を低くすべきと考える。

また、通常ではネットワークの状態はトラフィックモニターにより監視されているが、2011 年 3 月 11 日に東日本

大震災が発生した際には、トラフィック量は通常の数倍以上にも増えた [1]。これにより、経路制御に必要なトラフィック状態に関する情報が膨大となり、トラフィックモニターのみでネットワーク全体の状態を迅速に把握することは困難であった。そのため、長時間ネットワーク障害により、電話が繋がらないなどといった問題が生じた。その一方で、地震発生から 1 時間以内に、通信障害の被害の少ない東京において毎分 1,200 件以上のツイートの投稿では、地震に関するリアルタイムでの情報のやりとりが非常に活発に行われたことが明らかにされた [2]。このことから、Twitter [3] などのソーシャル情報は、災害時に有用なものであると言える。

そこで、トラフィックの種別を判定して、アプリケーションごとに制御を行う方式と、Twitter などのソーシャル・データから実社会の急激な状況変化を検出し、自動でユーザが必要とするアプリケーション情報を安定して提供できる仕組みの実装が必要と考えられる。

本研究では、自動制御を実現するプラットフォームとしてネットワーク仮想化ノード FLARE スイッチを使用し、Twitter などのソーシャル情報に基づき SDN によりアプ

<sup>1</sup> お茶の水女子大学  
Ochanomizu University, Bunkyo, Tokyo 112-8610, Japan

<sup>2</sup> 東京大学  
Tokyo University, Bunkyo, Tokyo, 113-8654, Japan

<sup>3</sup> 工学院大学  
Kogakuin University, Shinjuku, Tokyo, 163-8677, Japan

リケーション種別でトラフィック制御する手順の提案を行い、FLARE スイッチを使用したアプリケーション識別ネットワーク実験によりその可能性を示す。

本稿の構成は以下の通りである。2. 章で関連研究について述べ、3. 章で SDN と DPN について紹介する。4. 章ではアプリケーション毎の SDN 制御システム概要、5. 章ではアプリケーション毎の SDN 制御システム実験について説明し、6. 章でアプリケーション毎のスライス切り替え実験を説明する。最後に、7. 章で本稿をまとめる。

## 2. 関連研究

SDN や OpenFlow ?の技術を用いた、トラフィックの自動制御は既の実現している ???。ネットワークのトラフィック量の変動に応じて、ネットワークの構成などを動的に制御している。これによりネットワークの自動制御を実現しているが、トラフィック制御自体はトラフィックモニターに基づくものである。一方、Twitter 上で発信している緊急地震速報や Yahoo!ニュースの地震カテゴリの記事による情報を基にしたトラフィック制御も検討されている ??。

これらの研究は、アプリケーションの種別を直接識別し、アプリケーション毎のトラフィック制御あるいは経路制御を行っていない。例えば、災害時には重要である電話などのリアルタイムの情報交換を優先し、緊急時には輻輳の要因となる動画トラフィックや娯楽目的のアプリケーションのトラフィックを抑圧することは難しい。すなわち、ユーザが緊急災害時に必要とするアプリケーション情報を安定に提供することを目標としている本研究とは異なる。

そこで、本研究ではアプリケーションの種別を判断し、アプリケーション毎に経路を切り替えることが可能な FLARE スイッチ ?????を使用し、トラフィック制御実験を行っていく。

## 3. SDN と DPN

SDN (Software Defined Network) はネットワークの構成や機能をソフトウェアによって動的に設定可能にすること、およびそのアーキテクチャの概念のことである。SDN の実装技術の1つである OpenFlow は、従来1つのネットワーク機器の中で一体化されていた経路制御を行うコントロールプレーンとデータ転送を行うデータプレーンを分離している。

SDN (Software Defined Network) は、コントロールプレーンとデータプレーンの分離に加え、コントロールプレーンがプログラマブルであるという点が重要である。DPN (Deeply Programmable Network) は、そのプログラム性を強調したものであり、ネットワークの制御を担うコントロールプレーンだけでなく、パケットを転送するデー

タプレーンをプログラマブルにした点である。この DPN のコンセプトを実装したスイッチが、FLARE スイッチである。FLARE はネットワーク仮想化技術も提供する。ネットワーク仮想化技術によりスライスと呼ばれるトラフィック処理領域を複数生成することができる。さらに複数のスライスに対しトラフィックを分離し各スライスに割り当てる機能も提供する。スライス同士はメニーコアプロセッサ使用により物理的に処理リソースのアイソレーションを行っている。

したがって、複数のスライスに対し、それぞれ異なる処理をプログラマブルに実行できる。したがって、FLARE でアプリケーション識別機能を実装し、識別されたトラフィックをスライスに振り分けて個別にネットワーク制御することが可能になる。FLARE でのトラフィック処理は、Click module router ?でプログラミング可能であり、OpenFlow 機能も実装できる。

本研究では、この FLARE を使用することにより、SDN では不可能であった。緊急時に特定のアプリケーションのトラフィックを優先的に流すような、きめこまやかな制御が可能にする。

## 4. アプリケーション毎の SDN 制御システム概要

災害時におけるソーシャル・データに基づいた、アプリケーション毎における SDN 制御システムの概要は以下の通りである。図1はシステムの概要図を示している。本システムは6. 章で使用する。(1)-(4)の流れで、自動でトラフィック制御を行っていく。

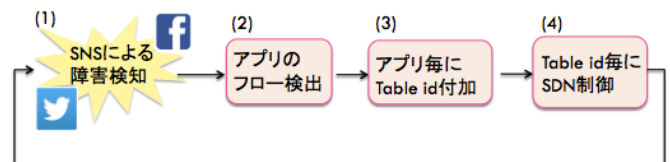


図1 アプリケーション毎の SDN 制御システムの概要

### (1) Twitter による障害検知 ?

リアルタイムにツイートを監視し、地理的に通信障害エリアを特定する。通信障害に関係あるキーワードを含むツイートを取得し、取得したツイートを同じ地名名詞が含まれるツイートごとにまとめる。機械学習を用い、ツイートに出現する地名が「その地域で障害が起きている」かを判断し、都道府県ごとにツイート数を数える。

### (2) アプリのフロー検出

ユーザ端末にインストールしたクライアントでユーザ

が使用しているアプリケーションのプロセスのポート番号からアプリケーションとそのフローとの紐づけを行いアプリケーションが生成するパケットの最初の SYN パケットをフィルタしアプリケーション識別用に SYN パケットの後ろにアプリケーション情報として、アプリケーション名とアプリケーション名の長さを付ける。

(3) アプリケーション毎に Table id 付加

アプリケーション毎に Table id を付加するため、少なくとも 1 つの OpenFlow を実装した FLARE スイッチにフローを収容し。FLARE スイッチに収容したフローのうち、SYN パケットのみ OpenFlow コントローラに packet in する。コントローラ上で、クライアント PC 上で付加したアプリケーション名からアプリケーションを認識し、そのアプリケーションが「災害時にユーザにとって重要なアプリケーションであるかどうか」を判別し、アプリケーション毎に SDN 制御を行うために必要な Table id を付加する。最後に、クライアント PC 上でパケットの後ろに付加したアプリケーション情報を取り除く。

(4) Table id 毎に SDN 制御

REST-API を使用し、Table id 毎に帯域制限等を行うことを記述したシェルスクリプトを用意しておく。コントローラから自動で再設定を行う。

## 5. アプリケーション毎の制御システム実験

### 5.1 実験概要

アプリケーション毎の制御システム概要を図 2 に示す。クライアント PC 上でアプリケーション識別子を付加し、FLARE スイッチ上でアプリケーション識別子を外す。トラフィックを FLARE スイッチに通すために、Proxy サーバを構築する。クライアント PC の各アプリケーション上で Proxy サーバの設定を行うことにより、FLARE スイッチと Proxy サーバを経由して、外部の Web サーバ等に接続させる。

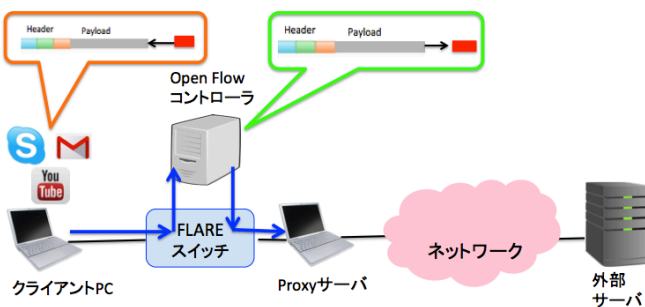


図 2 アプリケーション毎の制御システム概要

ユーザ端末を模擬したクライアント PC 上で YouTube と Skype を起動させ、アプリケーション毎の制御システムを動作させた。本実験では、YouTube は災害時に帯域を多く使用してしまうが、Skype などのアプリケーションに比べ、災害時には重要度が低いと判断し、300Mbps に帯域の制限を行った。帯域制御の効果を評価するため、それぞれのアプリケーショントラフィックを想定したフローに対し iperf を使用し、帯域を制限しないパケットと帯域を 300Mbps に制限したパケットのスループットをそれぞれ測定した。図 3 に実験ネットワークを示す。FLARE スイッチを 2 台使用し、クライアント PC と Proxy サーバ間のスループットをそれぞれ測定した。

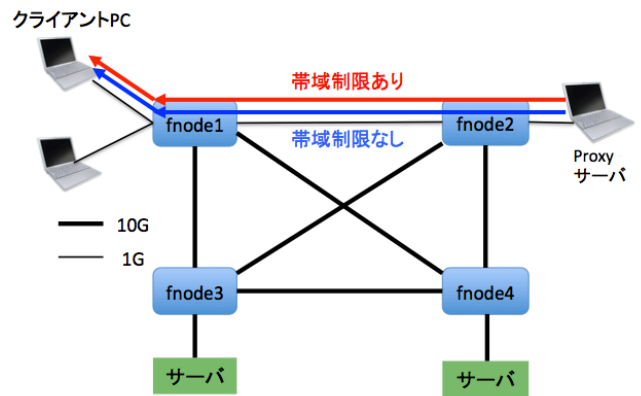


図 3 実験ネットワーク

### 5.2 実験結果

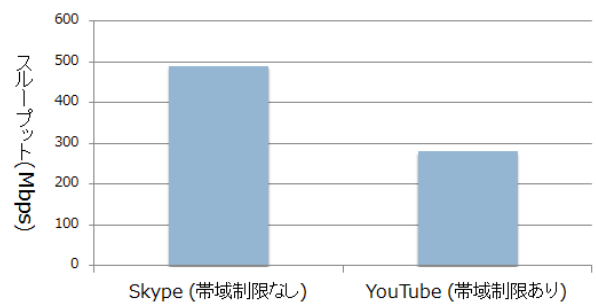


図 4 スループット測定結果

帯域制限したパケットと帯域制限を行わないパケットそれぞれのスループットを測定した。SDN 制御時のそれぞれのスループットを図 4 に示す。帯域を制限していないパケットは約 500Mbps であるのに対し、帯域が制限されているパケットは 280Mbps に抑えられている。この実験結果より、実際にアプリケーション毎に帯域制限が出来され

ていることを確認することができた。実験では、一スライスを使用し DRPOP 制御による帯域制御を行っているため、ネットワークのリンク帯域に余裕があっても、一スライス内では、最大帯域はフロー数で割った以上にならない。すなわち、2 フローの場合、最大帯域制限は 500Mbps である。

## 6. アプリケーション毎のスライス切り替え

### 6.1 実験概要

5. 章で使用した実験室内の環境でのネットワークを使用し、FLARE スイッチ上のスライス数を増やし合計 2 スライスで実験を行った。実験ネットワークを図 5 に示す。FLARE のトラフィック・スライス振分け機能でアプリケーション識別を行い、各スライスにトラフィックを収容し、スライス 1 のパケットにのみ帯域を制限する。iperf を使用し、帯域を制限しないフローと帯域を 300Mbps に制限したフローのスループットをそれぞれ測定した。

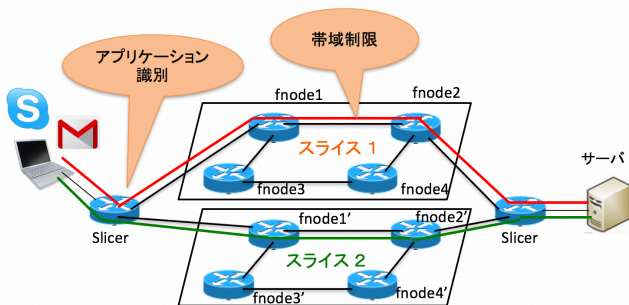


図 5 実験ネットワーク

### 6.2 実験結果

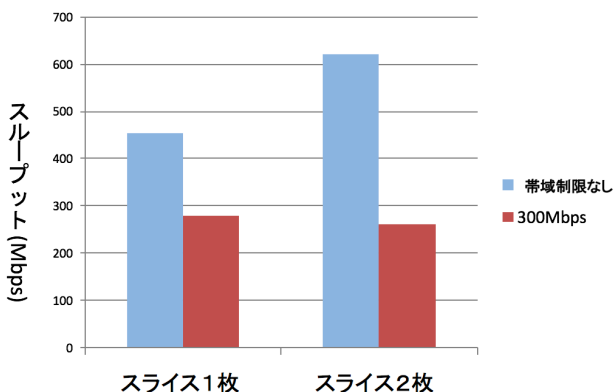


図 6 スループット測定結果

帯域制限したパケットと帯域制限を行わないパケットそれぞれのスループットをスライス一つの場合と 2 スライスの場合について、それぞれ測定した。SDN 制御時のそれぞれのスループットを図 6 に示す。一スライスの場合には帯域制限なしのパケットのスループットは 470Mbps に対して、二スライスの場合では帯域制限なしのフローのスループットは 500Mbps 以上が得られた。本実験結果より、スライスを複数使用することにより帯域制御を行った場合に、ネットワークリンク帯域をより効率的に使用することが可能となる。

## 7. まとめと今後の課題

東日本大震災のような大規模災害時には、トラフィック種別を判定しアプリケーションごとに制御を行う方式と、ソーシャル・データから実社会の急激な状況変化を検出し、自動でユーザが必要とするアプリケーション情報を安定に提供できる仕組みが必要である。先行研究として、東日本大震災などの大規模災害時における通信障害時に必要とされる、Twitter などのソーシャル情報に基づく経路制御システムの実装と評価を行った。そこで本研究では、アプリケーション毎に制御を行うシステムを実装し、実験室環境のネットワーク上で実験を行った。実際のアプリケーションを使用し、アプリケーション識別の SDN 制御システムを動作させ、実験を行った。実験では、実際にアプリケーション識別によりトラフィック制御できることを確認した。さらにアプリケーションに対応するフローを iperf で生成し実験を行った。フローに対し個別に帯域制御できることを確認し、帯域制限したトラフィックと帯域制限をしないトラフィックのスループットを測定することにより、本システムが動作していることを、実験的に確認した。また、スライスを複数使用し、帯域制御を行った。その結果、スライスを複数使用することによって、帯域を上手く使えることが確認できた。

今後の課題としては、災害時のトラフィックが膨大である輻輳状態に近い状態を再現し、本システムの実験を行い、Skype の遅延時間やジッタの測定を行う。帯域制限と経路切り替えを組み合わせたシステム実験を予定している。

## 謝辞

本研究は一部、総務省戦略的情報通信研究開発推進事業 (SCOPE) 先進的通信アプリケーション開発推進型研究開発および JST CREST JPMJCR1503 の支援を受けたものである。

## 参考文献

- [1] 総務省:「東日本大震災における通信の被災・輻輳状況」,<http://www.soumu.go.jp/maincontent/000136962.pdf>
- [2] 吉次由美:「東日本大震災に見る大災害時のソーシャルメディアの役割」,  
[https://www.nhk.or.jp/bunken/summary/research/report/2011\\_07/20110702.pdf](https://www.nhk.or.jp/bunken/summary/research/report/2011_07/20110702.pdf)
- [3] Twitter, <http://twitter.com/>
- [4] OpenFlow コンソーシアム: OpenFlow 仕様, Open Networking Foundation: <http://www.openflow.org/050819.pdf>, 2005 年 8 月 19 日
- [5] 飯島明夫:「OpenFlow/SDN のキャリアネットワークへの適用について」 電子情報通信学会技術研究報告. NS, ネットワークシステム 112.231 (2012): 85-87.
- [6] NEC:「OpenFlow の活用により災害時における移動通信ネットワークの動的なソフトウェア制御を実現」,  
<http://jpn.nec.com/rd/innovation/feature/2013/11-openflow.html?>, コンセンサス, 2013 年 11-12 月号
- [7] 橋本直樹, 廣津登志夫:「冗長経路を利用した OpenFlow 制御ネットワークの自動構築」,  
日本ソフトウェア科学会 第 15 回 インターネットテクノロジーワークショップ
- [8] 原瑠理子, 長谷川友香, 小口正人:「モニタリング情報に基づく OpenFlow を用いたネットワークトラフィック制御モデル」, DEIM2014, C9-6, 2014 年 3 月
- [9] 高橋裕, 秋山友理愛, 神津智樹, 山口実靖:「バースト的負荷変動を考慮した OpenFlow を用いた動的資源割り当て (SDN/OpenFlow).」,  
電子情報通信学会技術研究報告. NS, ネットワークシステム 113.472 (2014): 225-229.
- [10] Akihiro Nakao, "FLARE: Open Deeply Programmable Network Node Architecture," ,  
Stanford Univ. Networking Seminar, October 2012.  
[http://netseminar.stanford.edu/10\\_18\\_12.html](http://netseminar.stanford.edu/10_18_12.html)
- [11] A. Nakao. Software-defined data plane enhancing sdn and nfv. Special Section on Quality of Diversifying Communication Networks and Services,  
IEICE Transactions on Communications, E98-B(1): 12-19, 2015.
- [12] A. Nakao, P. Du, T. Iwai: "Application Specific Slicing for MVNO through Software-Defined Data Plane Enhancing SDN" , IEICE TRANSACTIONS on Communications, Vol.E98-B, No.11, pp.2111-2120, 2015/11/01
- [13] Akihiro Nakao, Ping Du, "Application and Device Specific Slicing for MVNO" ,  
2014 International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC), 2014, October
- [14] UCLA.edu, The Click Module Router Project  
<http://read.cs.ucla.edu/click/click>
- [15] Chihiro Maru, Miki Enoki, Akihiro Nakao, Shu Yamamoto, Saneyasu Yamaguchi, and Masato Oguchi, "Network Failure Detection System for Traffic Control using Social Information in Large-Scale Disasters,"  
ITU Kaleidoscope Conference 2015: Trust in the Information Society, S5.3, 9-11 December 2015.