

大規模災害時における 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[†] 中尾 彰宏^{††} 山本 周^{††} 山口 実靖^{†††} 小口 正人[†]

[†] お茶の水女子大学

〒 112-8610 東京都文京区大塚 2-1-1

^{††} 東京大学

〒 113-8654 東京都文京区本郷 7-3-1

^{†††} 工学院大学

〒 163-8677 東京都新宿区西新宿 1-24-2

E-mail: [†]{g1220526,oguchi}@is.ocha.ac.jp, ^{††}{nakao,shu}@iii.u-tokyo.ac.jp, ^{†††}sane@cc.kogakuin.ac.jp

あらまし 現在のインターネットは、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、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。そこで、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と考える。本研究では、ソーシャル・データから通信障害を検知し、トラフィックからアプリケーションの種類を識別を行うことにより、仮想化プログラマブルノード FLARE を用いてアプリケーション毎に帯域制御や経路制御を行う手法を提案し、実装と評価を行う。

キーワード SDN, OpenFlow, DPN, FLARE, 通信障害, ネットワーク制御

Implementation and Evaluation of SDN Control Method for Each Application by FLARE in Large-scale Disaster

Tsumugi TAIRAKU[†], Akihiro NAKAO^{††}, Shu YAMAMOTO^{††}, Saneyasu YAMAGUCHI^{†††}, and

Masato OGUCHI[†]

[†] Ochanomizu University

2-1-1 Otsuka, Bunkyo-ku, Tokyo, 112-8610 JAPAN

^{††} Tokyo University

7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8654 JAPAN

^{†††} Kogakuin University

1-24-2 Nishi-Shinjuku, Shinjuku-ku, Tokyo, 163-8677 JAPAN

E-mail: [†]{g1220526,oguchi}@is.ocha.ac.jp, ^{††}{nakao,shu}@iii.u-tokyo.ac.jp, ^{†††}sane@cc.kogakuin.ac.jp

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、電話などの情報通信を行うアプリケーションは災害時にはユーザにとっての重要度が高いが、YouTube やゲームなどの娯楽目的のアプリケーションは重要度が低いと思われる。また、通常ではネットワークの状態はセンサを使って管理されているが、2011 年 3 月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, センサ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, 長い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, 東京からだけで毎分 1,200 件以上のツイートが投稿され, 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示されている [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現が必要と考えられる.

そこで本研究では, 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く SDN による自動経路制御実験とアプリケーション毎の SDN 制御実験を行い, その結果より本システムの有効性を示す.

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. 章で提案システムについて紹介する. 4. 章では JGN-X 上の実験を行い, 5. 章では経路制御システムの性能評価を行い, また, 6. 章ではアプリケーション毎の制御システムの実験を行い, 本システムの動作を確認する. 最後に, 7. 章で本稿をまとめる.

2. 関連研究

SDN や OpenFlow [4] の技術を用いた, トラフィックの自動制御は既に実現されている [5][6][7][8]. これらの手法では, 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. 通常のトラフィック操作のみで制御ネットワークをスイッチ間接続によって構成されるデータネットワーク上に自動構築することを可能としている. また, 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存在する [9]. これらの研究は, アプリケーションの種別を判定し, 電話などの情報通信を行うアプリケーションは重要と判断して優先的にトラフィックを流し, YouTube やゲームなどの娯楽目的のアプリケーションはトラフィックを後回しにするような, アプリケーション毎の経路制御を行っていない. そのため, 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的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別し, 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 [10][11][12][13] を使用し, 実験を行っていく.

3. 提案システム

3.1 ソーシャル情報に基づく経路制御システム

災害時におけるソーシャル・データに基づいた, トラフィックの自動経路切り替え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. (1)-(4) の流れで, 自動で経路設定を行っていく. 本システムは 4. 章と 5. 章で使用される.

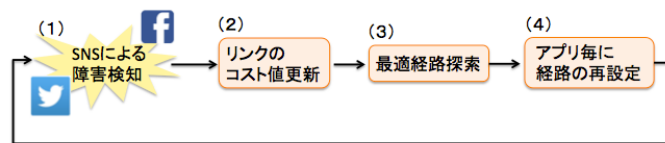


図 1: 自動経路制御システムの概要

(1) Twitter による障害検知 [14]

リアルタイムにツイートを監視し, 通信障害がどこで起こっているかを検知する. 通信障害に関係のあるツイートを取得し, その中から障害が起きている地域に関する情報を含むツイートを抽出し, 地域ごとにツイート数を数える.

(2) スイッチ間のコスト値の更新

スイッチ間のリンクのコスト値を, 60 秒間隔で更新する. コスト値の初期値を 1 とし, 障害ツイート中にスイッチと対応させた地名を含むツイートが 20 件以上あったら+1 する.

(3) 最適経路探索

グラフ上の 2 頂点間の最短経路を求めるアルゴリズムである, ダイクストラ法を用い, コスト値が最小になるような最適経路探索を行い, トラフィックの経路を決定する.

(4) 経路の再設定

OpenFlow の REST-API を使用し, あらかじめ経路を記述したシェルスクリプトを用意しておく. (3) で決定された経路に, コントローラから自動で再設定を行い, 経路を切り替える.

3.2 アプリケーション毎の SDN 制御システム

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 2 はシステムの概要図を示している. 本システムは 6. 章で使用される.

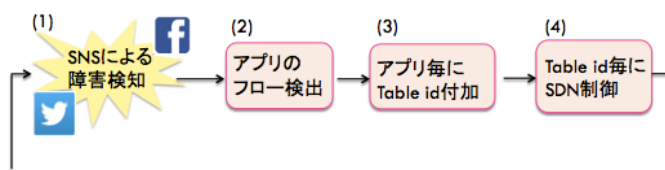


図 2: アプリケーション毎の SDN 制御システムの概要

(1) Twitter による障害検知

経路制御システムと同様に, ソーシャルデータから通信障害がどこで起こっているのかを検知する.

(2) アプリケーションのフロー検出

クライアント PC 上でアプリケーションの識別とアプリケーション識別のためのタグを付ける. パケットの量が膨大であるため, コネクション確立時に生成される SYN パケットのみを packet in し, プロセステーブルと宛先パケット番号からアプ

りを識別する。また、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 FLARE スイッチを通るようにする。FLARE スイッチに流れてきたパケットのうち、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。

4. JGN-X 上の実験

4.1 実験概要

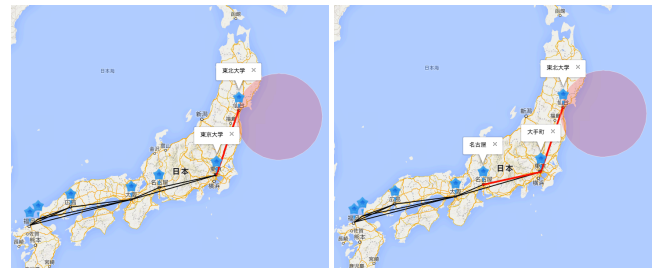
本研究では、国立研究開発法人情報通信研究機構 (NICT) が運用する研究用ネットワークである JGN-X [15] を使用する。全国 8 カ所の JGN-X の拠点および大学に設置した DPN の FLARE ノードを使用し、実験を実施した。JGN-X での実験では、東日本大震災時の 2011 年 3 月 11 日 14 時から 15 時の実際のツイートを基に、東北大から東大への通信を想定する。ソーシャル情報に基づく経路切り替えシステムを動作させ、トラフィックの流れを各スイッチ上で tcpdump を用いて確認した。

4.2 実験結果

システムは、東北大-東大サーバ間で通信障害を検知し、自動的に経路が切り替わったことを各スイッチ上で tcpdump を用いて確認した。図 3 に、障害検知前と障害検知後の経路を示す。(a) は障害検知前を、(b) は障害検知後のネットワークを示したものである。それぞれ、JGN-X 上の FLARE を Google Map 上に表示している。青いピンは各 FLARE スイッチ、黒い線は FLARE スイッチ間のリンク、赤い円は震源地を表している。赤い線がパケットが実際に流れた経路を表している。(a) 障害検知前は東北大から東大へパケットが流れており、(b) 障害検知後は東北大、大手町、名古屋の経路を通っている。

最初は東北大と東大間の経路を通る通信を行っていたが、東北大と東大の間で障害を検知した後、東北大、大手町、名古屋の経路を通り通信が行われた。経路切替として、アクセスするサーバを東大のサーバのデータをレプリケーションしている名古屋に設置したサーバへのアクセスに変更する経路切り替えが行われた。

また、経路が切り替わったことを Ping の応答遅延時間を測定することにより確認した。現時点では、システムが障害を検知してから、約 3 秒ほどでパケットが流れる経路が切り替わる



障害検知前 障害検知後

図 3: 実験結果

ことを確認できた。

この結果より、Twitter の情報から通信障害を検知し、パケットの流れる経路を自動的に障害を回避する経路に切り替える実験に成功した。このことから、本システムが有効に動作していると言える。

5. 経路制御システムの性能評価

5.1 実験概要

JGN-X を模擬した FLARE スイッチノードネットワークを使用し、性能評価を行う。RTP を使用する VoIP のアプリケーションでは、パケットロスが発生するとサービス品質の劣化が起るとされている [16]。また、通信事業者の IP 通信サービスが満たすべき国際標準規格 (Y.1541) では、パケットロス率の上限は 0.1% と規定されている。そこで、2 点間の通話におけるパケット損失評価を行う。FLARE ノードをそれぞれ、東京、石川、宮城にあると想定し、図 4 に示すようにローカルな環境において実験を行う。各 FLARE スイッチ間は 10Gbps の LAN で接続されている。東日本大震災時のツイートを基にシステムを動作させると、東京と宮城の間で通信障害を検知し、経路 1 から石川を経由する経路 2 に経路が切り替わる。

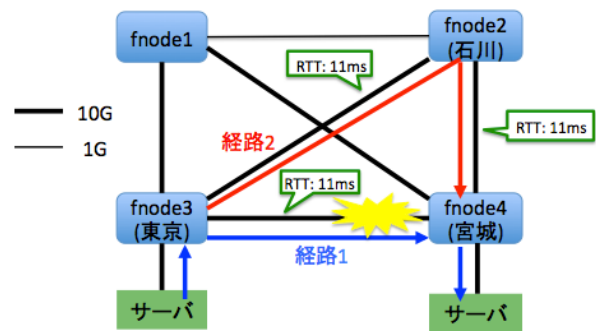


図 4: 性能評価に使用する FLARE ノード

各 FLARE ノード間に、距離に基づいた遅延を発生させ、障害検知前の経路である経路 1 の東京に対応するノードと宮城に対応するノードの間のみパケットロス率をコマンドで 0~0.05% 発生させる。経路 1 のパケットロス率に対する経路 1 と経路 2 のスループットを測定する。ここでは、各都道府県間の直線距離から計算して、各ノード間の RTT を 11ms と仮定し、それぞれ遅延を挿入する。

5.2 実験結果

図5は東京と宮城間のパケットロス率を変化発生させた場合の経路1と経路2のスループットである。

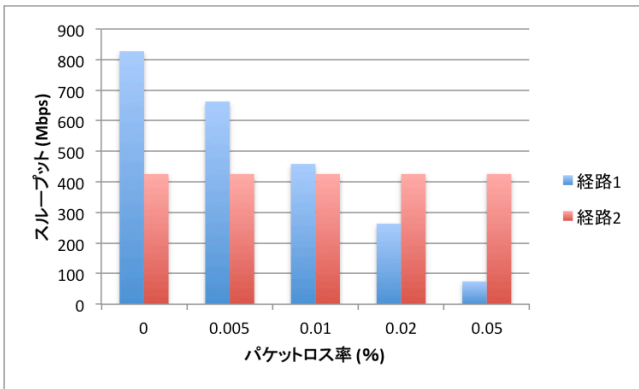


図5: 東京と宮城間のパケットロス率を変化発生させた場合の経路1と経路2のスループット

パケットロス率が0%の場合、経路1のスループットは石川に対応するfnode3のFLAREスイッチを経由している経路2のスループットに比べ、2倍ほどスループットの値が高い。経路2の方がスループットが低いのは、経由するFLAREノードが多いことと、距離に応じた遅延時間が挿入されていることが原因であると考えられる。0.01%のパケットロスが起こると、経路2とほぼ同じ性能になり、0.02%のパケットロス率になると、経路2の方が性能が良くなるのが分かる。0.05%のパケットロス率であると、格段に経路2の方が性能が良いことが確認できる。これにより、0.01%を超えるパケットロスが起こった場合には、石川を経由する経路に切り替える方がスループットの値が大きくなる、つまり性能が良くなるのが分かる。

6. アプリケーション毎の制御システム実験

6.1 実験概要

アプリケーション毎の制御システム概要を図6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。

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。そこで、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7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する。

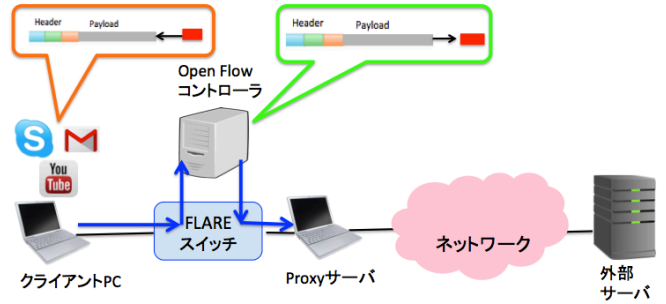


図6: アプリケーション毎の制御システム概要

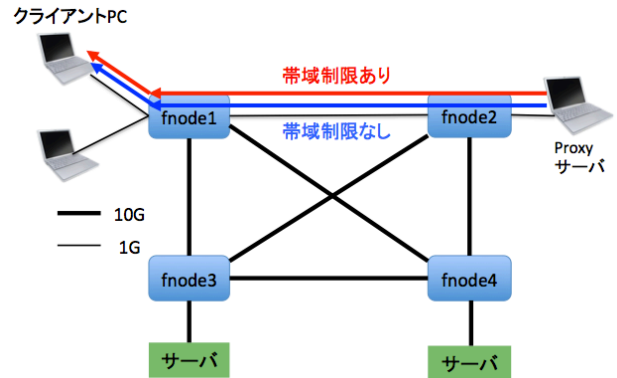


図7: 実験ネットワーク

6.2 実験結果

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8に示す。帯域を制限していないパケットは500Mbpsほどであるのに対し、帯域が制限されているパケットは280Mbpsほどに抑えられている。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ていることを確認することが出来る。

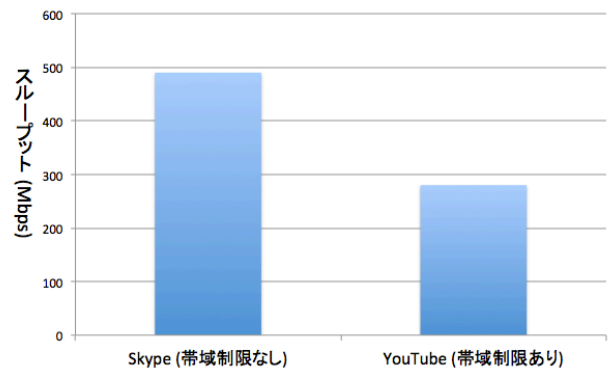


図8: スループット測定結果

7. まとめと今後の課題

東日本大震災などの大規模災害時における通信障害発生時に

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とアプリケーション毎のSDN制御システムの実装を行った。そして, 実際の東日本大震災時のツイート情報を基に, 広域ネットワークテストベッドJGN-X上のFLAREスイッチノードネットワーク上で, ソーシャル情報に基づく経路切り替えシステムを動作させ, 自動的に障害を回避する最適な経路に切り替わることを, 実験的に確認した。ソーシャル情報に基づく経路制御システムの性能評価をJGN-Xを模擬したFLAREノードネットワーク上で行った。また, 実際のアプリケーションを動作させることによって, アプリケーション毎のSDN制御システムを動作させ, 実験を行った。そこで, 実際にアプリケーション毎に帯域が制限されていることを確認し, 帯域制限したトラフィックと帯域制限を行わ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, 本システムが正常に動作していることを, 実験的に確認した。

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。また, 帯域制限と経路切り替えを組み合わせたシステムを考え, 実験を行う。

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科学技術振興機構戦略的創造研究推進事業(CREST)によるものである。

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/main_content/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/201107/20110702.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/>
- [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] 橋本直樹, 廣津登志夫:「OpenFlowネットワークにおける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14回インターネットテクノロジーワークショップ
- [9] 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
- [10] Akihiro Nakao, "FLARE: Open Deeply Programmable Network Node Architecture", Stanford Univ. Networking Seminar, October 2012. http://netseminar.stanford.edu/10_18_12.html
- [11] Akihiro Nakao, "FLARE: Open Deeply Programmable Network Node Architecture", Stanford Univ. Networking Seminar, October 2012, http://netseminar.stanford.edu/10_18_12.html
- [12] A. Nakao: "Software-defined data plane enhancing sdn and nfv. Special Section on Quality of Diversifying Communication Net-

- works and Services", IEICE Transactions on Communications, E98-B(1): 12-19, 2015
- [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] 丸 千尋, 榎 美紀, 中尾 彰宏, 山本 周, 山口 実靖, 小口 正人:「大規模災害時におけるTwitterを用いたネットワークシステム制御に有用な情報の抽出」第7回データ工学と情報マネジメントに関するフォーラム (DEIM2015), C7-3, 磐梯熱海ホテル華の湯, 2015年3月.
- [15] NICT, Network Testbed JGN-X <http://www.jgn.nict.go.jp>
- [16] 「次世代IPネットワーク推進フォーラム技術基準検討WG検討報告」, 次世代IPネットワーク推進フォーラム技術基準検討WG