

# 大規模災害における SNS 情報をトリガとしたアプリケーション毎のトラフィック制御の一検討

柳田 晴香<sup>†</sup> 中尾 彰宏<sup>‡</sup> 山本 周<sup>‡</sup> 山口 実靖<sup>††</sup> 小口 正人<sup>†</sup>  
<sup>†</sup>お茶の水女子大学 <sup>‡</sup>東京大学 <sup>††</sup>工学院大学

## 1. はじめに

東日本大震災時、電話やインターネットが使えない事態が生じたことから、大規模災害時には、従来のネットワーク機器監視による制御だけではなく新たな情報を利用した制御が期待されている。我々は、多くの人間の目や口コミによるネットワーク障害に対する集合知が災害の状況把握に有益であるという仮説に基づいて、SNS 情報に基づいたネットワーク制御システムを提案している [1]。

従来のネットワーク制御システムには、制御プレーンをプログラム可能とする SDN (Software Defined Networking) による柔軟な集中制御が期待されているが、現在の SDN では依然として、ヘッダ情報に基づく経路制御など従来と同様な制御の効率化に限定される。そこで我々は、データプレーン処理とその API (Southbound Interface) をプログラム可能とする SDN を拡張する DPN (Deeply Programmable Network) の概念を適用し、より高度で柔軟な制御を実現する。災害時においては、ネットワークリソースの不足が予測され、情報伝達のためのアプリケーションの優先制限などアプリケーション毎の QoS 制御が効果的だと判断できる。DPN を実現する FLARE[2] では、このような制御もプログラマブルに実装可能である。

本稿では、この FLARE スイッチを用いて、SNS 情報を活用した、アプリケーションの種類に基づくプログラマブルな QoS 制御システムを実装し、本提案システムが災害時に有効であることを示す。

## 2. 提案システム

本研究で提案するシステムの動作は以下の通りである。

### (1) アプリケーションの識別

ユーザ端末で、プロセステーブルの参照によりアプリケーションを識別し、アプリケーションの名前をトレーラとして SYN パケットの後ろに付加する。FLARE スイッチが、この SYN パケットをコントローラに渡すと、コントローラでトレーラを読み込み、アプリケーションの識別を行う。

### (2) Twitter による障害検知

文献 [1] に従って、コントローラでリアルタイムにツイートの解析を行い、障害ツイート数と障害が発生した地名を取得。取得した地名とスイッチを対応させる。

### (3) リンクのコスト値の更新

スイッチ間のリンクのコスト値を、初期値を 1 とし、障害ツイート中に対応させた地名を含むツイートが

20 件以上あったら + 1 と更新する。コスト値の更新は 60 秒間隔で行う。

### (4) 最適経路探索

ダイクストラ法を用いて、スタートからゴールへコスト値最小の経路をリストで出力。

### (5) 経路の再設定

決定された経路に、REST-API でフローエントリをスイッチ側に設定する。

### (6) アプリケーション毎の QoS 制御

メールや電話などの情報伝達のアプリケーションには帯域を確保し、逆に帯域を沢山取ってしまう YouTube などの娯楽目的の動画視聴のアプリケーションには制約をかけるアプリケーション毎の QoS 制御を行う。

## 3. FLARE 実機実験

### 3.1 実験環境

本研究では、図 1 に示す物理構成で実機実験を行う。1G は 1Gbps, 10G は 10Gbps のネットワーク接続を示す。コントローラから、4 台の FLARE スイッチを制御し、様々な制御モデルを検討する。また、h4 はプロキシサーバ、h6 はウェブサーバとして利用する。

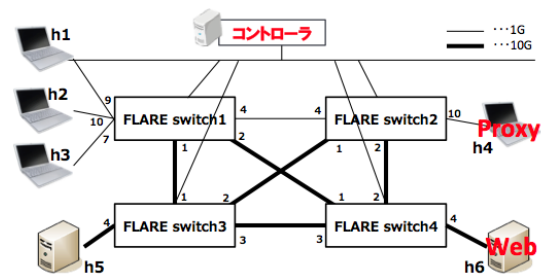


図 1: FLARE 物理構成

### 3.2 比較実験

本実験では、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用いる。図 2 のように FLARE スイッチを 1 から順に、岩手、東京、京都、福岡の地域名に対応づけ、岩手付近の h1, h2, h3 が、東京付近の h4 のプロキシサーバを通して通信する。h1 は Skype のトラフィック、h2, h3 は YouTube を想定したファイルダウンロードのトラフィックを流す。前提として、障害検知後は Skype を優先し、YouTube は迂回経路を通す。実験は、提案システムなしの場合と提案システムあり (図 2) の場合で比較して行う。提案システムなしの場合、障害の前後でも Skype と YouTube は同一経路を流れる。

提案システムありの場合、まず Skype と YouTube のアプリケーションの識別が行われる。また、全てのリンク間のコスト値はデフォルトで 1 なので、コスト値最小の Route1 の経路がデフォルトで設定される。コスト値の更新が 60 秒間隔で行われ、Twitter の解析システムより岩手

東京間で障害に関するツイートが 20 件以上検知されたことによって、2 回目の 120 秒後の更新の際に岩手と東京間のコスト値が + 1 更新される。その後も 60 秒間隔で岩手東京間で障害が検知され続けたことにより、コスト値が + 1 され続ける。3 回目の 180 秒後の更新の時点で、岩手と東京間のコスト値が 3 になるため、コスト値最小の 2 である京都を通る Route2 の経路が選択される。ダイクストラ法によってこの Route2 の迂回経路が選択されると、経路を設定する REST-API が呼び出される。これにより、フローエントリがスイッチ側に投げられて、YouTube のトラフィックにおいて、Route2 の迂回経路が再設定される。本実験により、災害時に Twitter 情報に基づいて、障害地区を迂回する経路切り替えを確認できる。

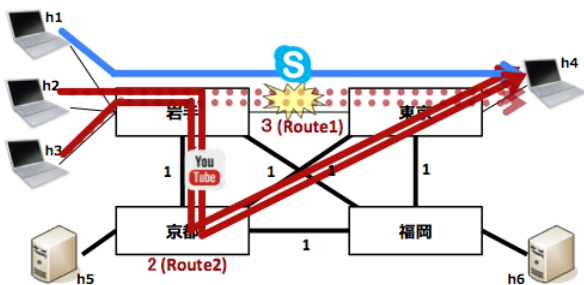


図 2: 提案システムありの場合のトラフィックの流れ

### 3.3 スループット測定

3.2 節の実験において、スループットの測定を行う。結果は図 3、図 4 の通りである。図 3 が提案システムなしの場合、図 4 が提案システムありの場合である。

図より、提案システムなしに比べて、提案システムありの場合、障害検知の後に Skype のスループット値が上がっていることが分かる。また、迂回経路を通した YouTube に関して、提案システムなしに比べてスループット値が大きくなっていることがわかる。よって、災害時における情報伝達のためのアプリケーションの優先制御に加え、全体のスループット値の向上が達成されたと判断できる。

## 4. まとめと今後の課題

SNS 情報から障害検知を行う、災害時に最適なアプリケーション毎の QoS 制御システムの提案を行った。本稿では、提案システムを実装することで、情報伝達のためのアプリケーション以外のトラフィックを迂回経路に通す経路制御を達成した。さらに、提案手法の評価として、RTT を 20-30ms 程度に抑えつつ特定の情報伝達のアプリケーションのスループット値を 800Kbits/sec 程上げることに成功し、災害時において本提案システムが有効であることを示した。本論文の貢献は以下の通りである。

第一に、SNS 情報を利用したネットワーク障害検知システムをネットワーク経路制御に統合することで、SNS 情報に基づいて、ネットワークを自動的かつ自律的に制御できることを示した。

第二に、FLARE の実験環境で、アプリケーションの識別を行うことで、特定のアプリケーションに対して優先制

御をする、より高度な QoS 制御が行えることを示した。今後の課題としては、更に複雑なシナリオやトポロジでの実証実験である。

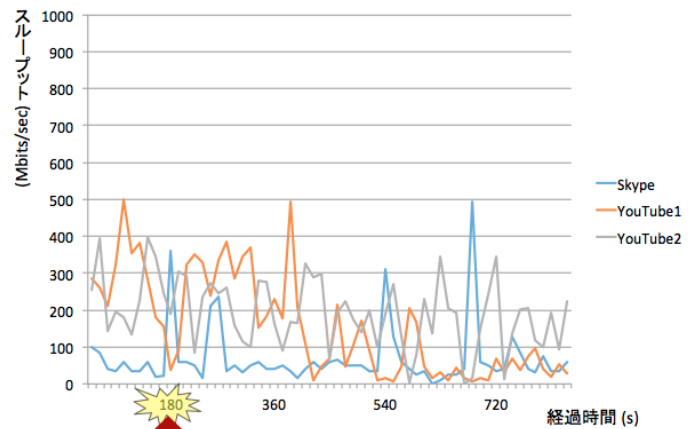


図 3: 提案システムなしの場合のスループット値

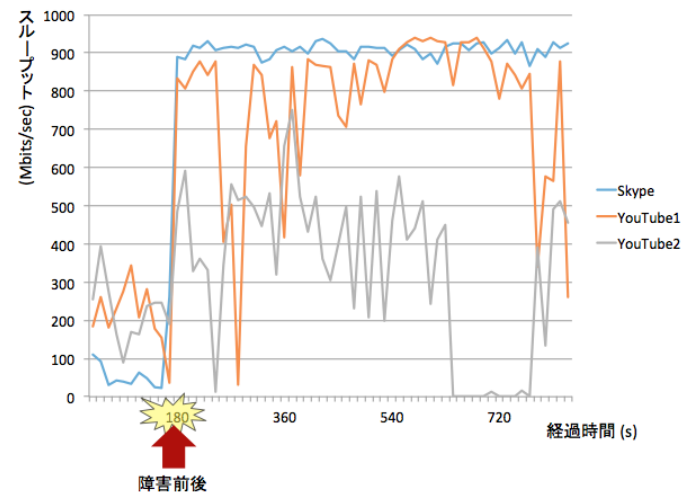


図 4: 提案システムありの場合のスループット値

## 参考文献

- [1] Chihiro Maru, Miki Enoki, Akihiro Nakao, Shu Yamamoto, Saneyasu Yamaguchi, and Masato Oguchi: "Development of Failure Detection System for Network Control using Collective Intelligence of Social Networking Service in Large-Scale Disaster" In Proc. the 27th ACM Conference on HT2016, pp.267-272, July 2016.
- [2] 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)