大規模災害時における SNS 情報を用いたアプリケーション毎の QoS 制御手法の実装と評価

柳田　晴香† 　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: †{g1120541,oguchi}@is.ocha.ac.jp, ††{nakao,shu}@iii.u-tokyo.ac.jp, †††sane@cc.kogakuin.ac.jp

あらまし　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 [2] 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 [3] では，この様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．

キーワード　SNS, SDN, OpenFlow, DPN, FLARE

1. はじめに

2011 年 3 月に発生した東日本大震災において，一部ネットワークが断絶し，電話やメールなどの通信手段が利用できない事態が発生した．この際，復旧に時間が掛かった要因として以下の 2 点が挙げられた．経路切り替え等の設定が一部手作業であったこと，また，ネットワーク機器からの監視アラート情報が膨大になり，どこが深刻なダメージを受けているのか迅速に把握できなかったことである [4].

これらの背景に対し，本研究では，大きく 2 つのアプローチをとる．第一に，従来のネットワーク機器監視に加え，多くの人の目に依る SNS 情報を利用することで，大震災のような大規模な災害時にネットワーク状況を迅速に把握することである．我々は主要な SNS のひとつである Twitter [5] からリアルタイムにツイートを解析することで，高い正確性でネットワーク通信障害を迅速に検知するシステムを構築した [1]．この手法により，大規模な災害時に，ネットワーク全体でどこが深刻なダメージを受けているのか速やかに特定することができる．

第二に，ネットワーク経路の柔軟な自動集中制御である．災害時において経路切替を自動で柔軟に制御することは重要である．近年では制御プレーンをプログラム可能とする SDN(Software Defined Networking) や OpenFlow [6] による柔軟な制御が期待されている．しかし現在の SDN では，依然として，ヘッダ情報に基づく SPI(Stateful Packet Inspection) 制御など従来と同様な制御の効率化に限定され，DPI(Deep Packet Inspection) のように IP パケットのデータペイロードの情報を基にフィルタリングなどの処理方法を決める，高度なパケット処理は考えられていない．インターネットの利用が一般化し，その上で動くサービスやアプリケーションが複雑になった現代のネットワークでは，SPI では不十分になってきた [7]．そこで最近，SDN を拡張し，データプレーン処理とその API(Southbound Interface) をプログラマブルにすることで DPI 処理を可能にする，DPN (Deeply Programmable Network) の概念に基づいた，ネットワーク機器の開発がなされている [3] [2] [8]．大規模な災害時には，ネットワークリソースの不足が予測され，特定のアプリケーションの優先制限など，DPI を活用する制御が効果的であると判断できる．よって本研究では，DPN 環境を構築し，より高度で柔軟なネットワーク制御を検討する．

本稿では，OpenFlow の機能とアプリケーションの中身に基づく制御機能の両方を実装可能な FLARE スイッチ [3] [2] を用いて DPN 環境を構築し，災害時に最適なアプリケーション毎の QoS 制御を，SNS による障害検知と合わせて達成する．

図 1 は提案システムを適用したネットワーク環境を示している．この図に示すように，SNS 情報をコントローラと呼ばれるサーバ上に集めて解析し，ネットワークの障害地点の推定を行う．この情報を基に，コントローラで判断を行い，ネットワーク上のスイッチに指示を送る．スイッチはアプリケーションを振り分け，アプリケーションの種類に応じた経路制御や帯域制

御を行うことで，アプリケーション毎の QoS 制御を実現する．例えば，災害時に重要なメールや Skype などの情報伝達のための アプリケーションの優先制御である．

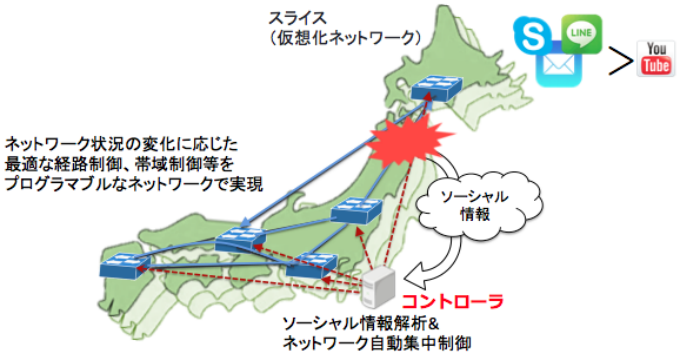

図 1　提案システムにより実現されるネットワーク

本論文の貢献は，以下の 2 点に集約される．

i　SNS 情報を利用したネットワーク障害検知システム [1] をネットワーク経路制御に統合する．これにより，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す．

ii　DPN 環境の適用により，ネットワークを流れるパケットをアプリケーション別のトラフィックに振り分け，種類別に適切な帯域を設定するような，より高度なネットワーク制御が行えることを示す．

本論文の構成は以下の通りである．まず 2. 章で関連研究について述べ，3. 章で SDN と DPN について説明する．次に，4. 章で提案システムの概要を紹介する．そして，5. 章で実験環境を示し，6. 章で各実験について述べる．最後に，7. 章で本稿をまとめる．

2. 関 連 研 究

SDN や OpenFlow 技術を利用して，災害時に適切なネットワーク制御を自動的に行う障害対策手法が数多く存在する [9] [10] [11] [12] [13] [14] [15]．NTT ドコモら [9] は，OpenFlow 技術を用いて，低優先トラフィックに対する抑制を行うことで通信サービスの優先度に応じた制御を実現している．具体的には，DiffServ (Differentiated Services) を用いたクラスベースの CoS 制御で，受信パケットの通信フローとクラスのマーキングを行うことで，相対的に優先度を識別している．しかし，この優先制御は 1 人のユーザが複数の通信サービスを使用する状況下では，正常な識別は可能でない．よって，1 ユーザが複数のアプリケーションを使用する場合でも正常に識別でき，各アプリケーションに対して各々に適切な制御を絶対的に行う本研究手法とは異なる．また，小川ら [10] [11] は，IP アドレスと位置情報をマッピングして DB で管理し，DB と OpenFlow コントローラとの連携をとる手法を提案している．加えて，被災地の通信パケットに対して，IP ヘッダ中の ToS フィールドに災害 ID を付与することで被災地の通信パケットを識別し，QoS を実現する提案をしている．さらに OpenFlow コントローラに，緊急地震速報や気象情報などの外部災害情報を収集する機能を付加する点でも本研究と似ている．しかし本研究は，SNS による集合知を用いる点と，アプリケーションの種類毎に QoS 制御を行うという点で新規性がある．[12] [13] [14] については，無線アドホックネットワークに OpenFlow を適用するもので，ユーザが端末上で評価基準の重みを事前に登録し，それを基にした QoS 制御を提案している．江戸ら [15] は，災害リスクのモデル化を行い，モデル化された災害シナリオを基に経路制御を行っている．しかし，これらの研究はネットワークのトラフィックデータなど内部情報を基に通信制御を行おうとしており，SNS データという外部情報を基に通信制御を行う本研究とは異なる．

加えて，既存の研究はどれも，コントローラによるネットワーク機器の集中管理という SDN の概念に留まっている．すなわちこの場合，DPI などのより高度で柔軟な制御は考えられていない．

3. SDN と DPN

3.1 SDN/OpenFlow による制御

OpenFlow [6] は SDN を実現するためのプロトコルのひとつで，ネットワーク全体の集中管理・集中制御を可能にする．この技術は既に実用化段階で，主にクラウドを提供するデータセンターや企業内 LAN などの狭域ネットワークでの導入が進められている．

OpenFlow の仕組みは，コントロールプレーンとデータプレーンの機能の分離である．コントロールプレーンとはデータの転送経路を決定する経路制御の頭脳であり，データプレーンはコントロールプレーンの指示に従ってデータを転送する．これら二つの機能は，従来の物理スイッチではどちらも同じ機器内に組み込まれていたが，OpenFlow ではコントロールプレーンをネットワーク機器外部のサーバ上にソフトウェアとして分離し，スイッチではデータ転送機能のみを実行する．OpenFlow プロトコルはこれらを接続する標準的なインタフェースで，インタフェース経由でのデータプレーンの制御を可能にする．

3.2 DPN/FLARE による制御

3.1 節より，SDN ではデータプレーンは，データ転送という固定的な動きしか実行しないことがわかる．このデータプレーンをもプログラム可能にするのが，DPN の概念である．つまり，DPN ではネットワークをフルにプログラム可能にする事を目的とする．

そして DPN を実現するために開発されたアーキテクチャの一つが，FLARE である [3]．FLARE では，データプレーンを Click [16] という言語で実装する．Click の特徴は，「フレームを受け取る」「フレームを転送する」といった様々な基本機能が，モジュールとして用意されている点である．モジュールを組み合わせることで，簡単にスクリプトでネットワーク機器の動作を記述できる．また，独自のモジュールを作成することも可能で，ユニークな動作をするネットワーク機器を作成することが可能である．東京大学では，OpenFlow1.3 スイッチを Click のモジュールで独自に開発しており，FLARE の OpenFlow による操作を可能にしている．

図 2 に FLARE のメカニズムを示す．スリバーと呼ばれるコ

ンテナの中に，Click を用いて作成された，プログラム可能な仮想スイッチが実装される．これらスリバーの集合で，スライスと呼ぶ仮想ネットワーク空間を作成する．

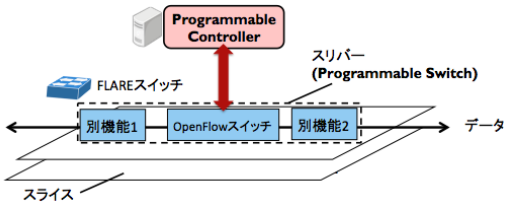

図 2　FLARE の仕組み

4.　提案システムの概要

本研究では，Twitter の解析から得られた障害情報をトリガとして，トラフィックの最適化をアプリケーション毎に自動的・自律的に行う．提案システムの概要を図 3 に示す．

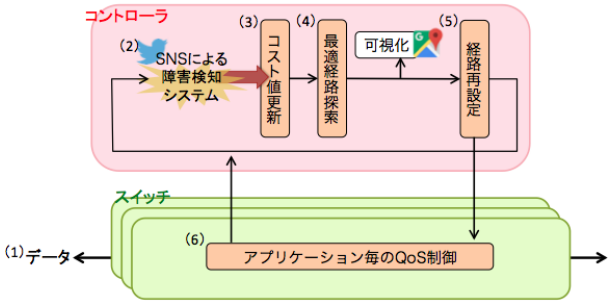

図 3　提案システム概要

(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．

4.1　アプリケーションの識別と帯域制御

図 3 の (1) について動作を確認する（図 4）．システム上でアプリケーションの識別を行うために，ユーザ端末でアプリケーションの識別子を付加する．具体的には，まずユーザ空間で SYN パケットをフックし，プロセステーブルと宛先ポート番号でアプリケーションの識別を行う．次に識別したアプリケーションの名前と名前の長さをトレーラとして SYN パケットの後ろに付加する．後はチェックサムの再計算をしたところでアプリケーションの情報を付加した SYN パケットを戻す．暗号化された通信でも，トレーラは暗号化されないので，アプリケーション情報は自由にアクセス可能となる [17] [18].

スイッチ上では，アプリケーション識別子を付加された SYN パケットを，コントローラに渡す．コントローラでは，トレーラの中身を見て，アプリケーションの識別を行う．更に，識別したアプリケーション毎に ID を与え，トレーラを外す．ID を指定してキューの設定を行うことで，帯域制御が可能になる．

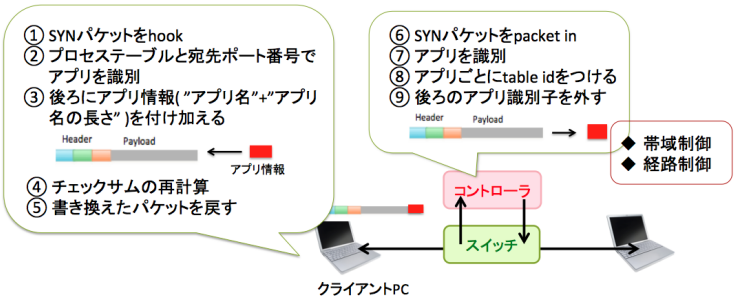

図 4　アプリケーションの識別とアプリケーション毎の制御の流れ

5.　FLARE 実機環境

5.1　物 理 構 成

本研究では，図 5 に示す物理構成で実機実験を行う．図中で 1G は 1Gbps，10G は 10Gbps のネットワーク接続を示す．コントローラは FLARE 管理用のサーバでありスライスの作成等ができる．図 3 に示す提案システムのコントローラ部を，このサーバ上に構築する．このコントローラから，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．各マシンの仕様は表 1 の通りである．

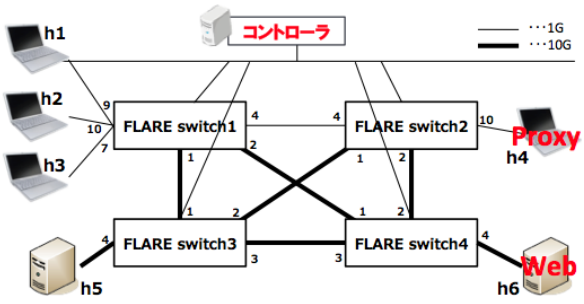

図 5　FLARE 物理構成

表 1　マシンの仕様

FLARE switch1 ～ FLARE switch4	Model	Sunwaytech SWS (barebone)
	CPU	Core i7-3612QE Mobile 2.1GHz
	Memory	8GB
	OS	CentOS 6.4
h1～h4	Model	Toshiba dynabook R734
	CPU	Core i5-4210 M 2.6GHz
	Memory	8GB
	HDD	SATA 500GB 5400RPM
	OS	Ubuntu14.04
h5, h6	Model	Dell PowerEdge R220
	CPU	Xeon E3-1241 v3 3.5GHz
	Memory	8GB
	HDD	SATA 1TB 7200RPM
	OS	Ubuntu14.04

5.2　基礎性能測定

次に，前節で説明した FLARE 実験環境の動作確認と特性を把握するため，各経路のスループットを iPerf により測定した（図 5 に示していないトポロジを含む）．結果を表 2 に示す．1G と 10G が混合している経路は，1G のみや 10G のみの経路に比べてややスループットの低下が見られる．

表 2　各経路のスループット測定結果

	経路	スループット
1G10G 混合	h1-s1-s3-h5	500(Mbits/sec)
	h1-s1-s2-s3-h5	590(Mbits/sec)
	h1-s1-s4-s3-h5	500(Mbits/sec)
	h1-s1-s2-s4-s3-h5	500(Mbits/sec)
1G のみ	h1-s1-h2	930(Mbits/sec)
	h1-s1-s2-h4	930(Mbits/sec)
10G のみ	h5-s3-h6	2.1(Gbits/sec)
	h5-s3-s4-h6	1.7(Gbits/sec)

6.　FLARE 実機実験

6.1　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．下図のように 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って通信する．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（図 6）と提案システムあり（図 7，図 8）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（図 6）．

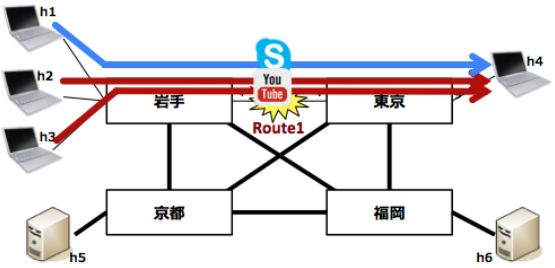

図 6　提案システムなしの場合のトラフィックの流れ

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 7）．

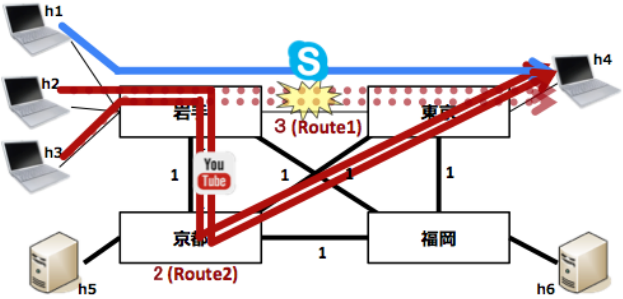

図 7　提案システムあり（経路制御時）の場合のトラフィックの流れ

また，提案システムありの場合で，アプリケーション毎に帯域を設定する場合が図 8 である．Skype と YouTube のアプリケーションの識別が行われ，Skype は災害検知後に最低でも 800 Mbps の帯域確保をし，YouTube には最大でも 100 Mbps までとなる帯域制限をかける．次節にスループット値とともに，結果を示す．

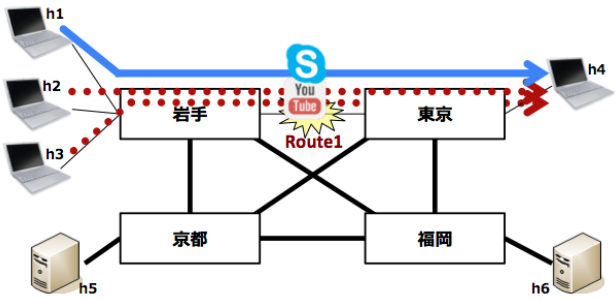

図 8　提案システムあり (アプリケーション毎の帯域制御時) の場合のトラフィックの流れ

6.2　スループット測定

6.1 節の実験における，スループットの測定結果を示す．図 9 が提案システムなしの場合，図 10 が提案システムによる経路制御をする場合，図 11 がさらにアプリケーション毎の QoS 制限をする場合である．

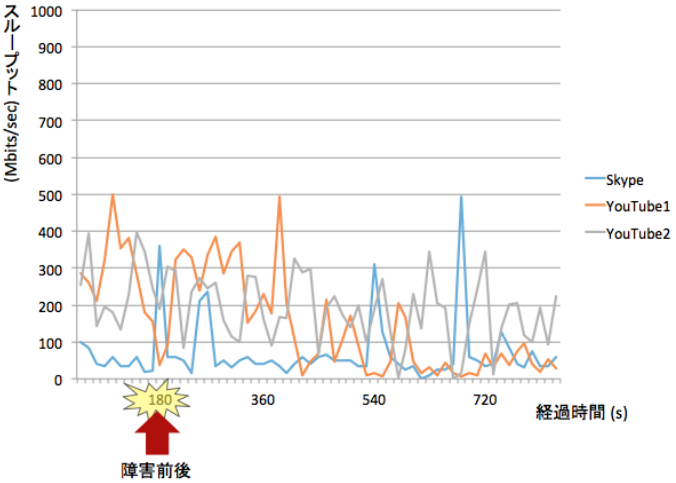

図 9　提案システムなしの場合のスループット値

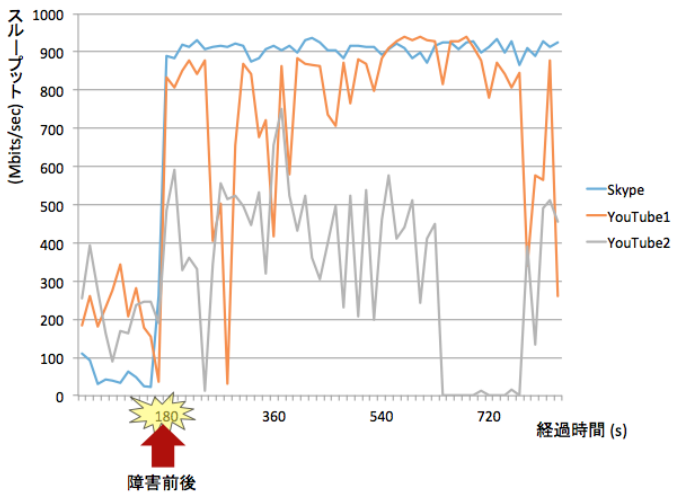

図 10　提案システムあり（経路制御時）の場合のスループット値

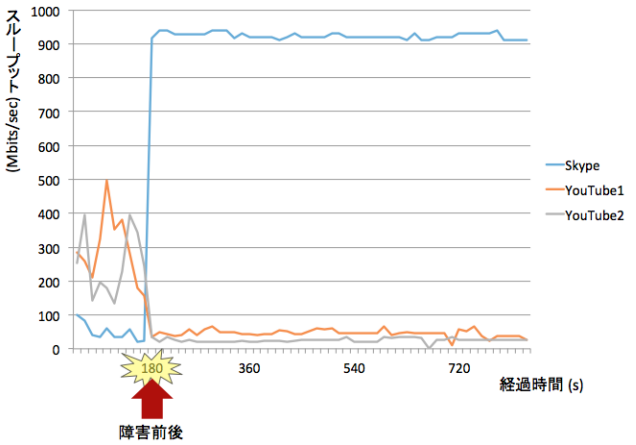

図 11　提案システムあり (アプリケーション毎の帯域制御時) のスループット値

図より，提案システムなし（図 9）に比べて，提案システムあり（図 10，図 11）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．図 11 においては，Skype のトラフィックは 800 Mbps の帯域保証が行われ，YouTube のトラフィックは 100 Mbps 以下にシェービングされるような，特定のアプリケーションに対するきめ細やかな制御が見て取れる．よって，本提案システムに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とアプリケーション毎の QoS 制御が達成されたと判断できる．また，図 10 において，迂回経路を通した YouTube に関してもスループット値の向上が見られることから，本提案システムによりネットワーク全体のスループット値の向上にも成功したと言える．

6.3　遅延時間の測定

次に，提案システムの経路切替にかかる時間を把握するため，Ping を 1ms 秒毎に 1 つ飛ばし遅延時間を測定する．結果を図 12 に示す．経路切替時に 20-30ms 程度の遅延が発生しているのは，コントローラにフローエントリを問い合わせるためである．

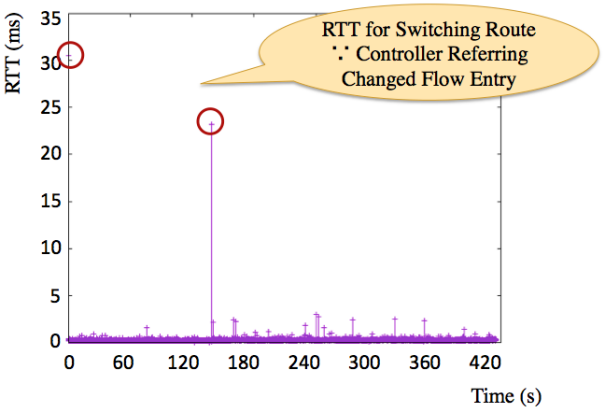

図 12　経路切替時の往復遅延時間（RTT）

7.　まとめと今後の課題

本研究では，災害時において情報伝達のためのアプリケーションを優先すべきであると考え，SNS でリアルタイムに検知した障害情報に基づいて，情報伝達のためのアプリケーションを優先する制御を高度で柔軟なプログラマブルな制御手法により実現することを研究目的とした．

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と，アプリケーション毎に帯域を設定するアプリケーション毎の QoS 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 800 Mbps 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，より詳細な提案手法の評価として，更に複雑なトポロジでの実証実験の評価を行いたい．

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] 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 Hypertext and Social Media (HT2016), pp.267-272, Halifax, Canada, July 2016.

[2] Akihiro Nakao, ”Software-Defined Data Plane Enhancing SDN and NFV”, Special Section on Quality of Diversifying Communication Networks and Services, IEICE Transactions on Communications, vol.E98-B, No.1, pp.12-19, Jan. 2015.

[3] Akihiro Nakao, ”FLARE: Open Deeply Programmable Network Node Architecture,” Stanford Univ. Networking Seminar, October 2012. http://netseminar.stanford.edu/10_18_12.html

[4] NTT DOCOMO, ”Improvement of Credibility for Operation System in the Case of Large Disaster”, https://www.nttdocomo.co.jp/binary/pdf/corporate/technology/rd/technical_journal/bn/vol20_4/vol20_4_026jp.pdf, Technical Journal Vol. 20 No. 4, 2013.

[5] Twitter, http://twitter.com/

[6] N.McKeown, T.Anderson, H.Balakrishnan, G.Parulkar, L.Peterson, J.Lexford, S.Shenker, and J.Turner. ”OpenFlow: enabling innovation in campus networks.” ACM SIGCOMM Computer Communication Review, Vol.38, No.2, pp.69-74, 2008.

[7] 日経ネットワーク, http://itpro.nikkeibp.co.jp/article/Keyword/20121112/436523/?rt=nocnt

[8] Barefoot Networks, http://www.barefootnetworks.com

[9] NTT ドコモ，東北大学，日本電気株式会社，富士通株式会社，株式会社日立ソリューションズ東日本：「大規模災害時における移動通信ネットワーク動的通信制御技術の研究開発」総務省平成 23-24 年度研究開発.

[10] 小川康一，吉浦紀晃：「災害 ID 付与方式による災害時のネットワーク優先配送 OpenFlow による実装と評価」, 電子情報通信学会技術報告, vol.2014-IOT-24, no.23, pp.1-6, 2014-02-20

[11] 小川康一，吉浦紀晃：「通信の信頼性確保を考慮した位置情報に基づくネットワーク運用手法」,DICOMO 2015, 8B-2, pp.1646-1652, 2015

[12] 熊谷友来，関野雄人，内田法彦，柴田義孝：「OpenFlow をベースとした災害時における End-to-End 通信路の選択方法の実現」, 情報処理学会第 75 回全国大会, pp.3-337-338，2013 年 3 月.

[13] 関野雄人，柴田義孝，内田法彦，白鳥則郎：「OpenFlow をベースとした災害情報ネットワークにおけるリンク切り替え技法の実現に関する研究」, 情報処理学会 SIG, Vol.2013-DPS-154 No.49, 2013 年 3 月.

[14] 佐藤剛至，柴田義孝，内田法彦：「SDN によるコグニティブ無線技術を基盤とした災害に強いネバー・ダイ・ネットワークに関する研究」, マルチメディア通信と分散処理, IPSJ-DPSWS2013006, pp.46-52，2013 年 12 月.

[15] 江戸麻人，和泉諭，阿部亨，菅沼拓夫：「災害リスクを考慮したスマートルーティングの設計と実装」, DICOMO 2015, 7E-3, pp.1520-1524，2015 年 7 月.

[16] The Click Modular Router Project:http://www.read.cs.ucla.edu/click/

[17] Akihiro Nakao, Ping Du. ”Application and Device Specific Slicing for MVNO”, 2014 International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC), Oct. 2014.

[18] Akihiro Nakao, Ping Du, Takamitsu 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.