

2T-07 大規模災害時における仮想化ノード FLARE によるアプリケーション毎の SDN 制御

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 [‡]東京大学 ^{††}工学院大学

1. はじめに

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、センサーのみを使用したネットワーク状況の迅速な把握が困難である。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。

本研究では自動制御を実現するプラットフォームとして FLARE[1] を使用し、SNS 情報に基づく SDN による自動経路制御実験や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。

2. 提案システム

2.1 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。

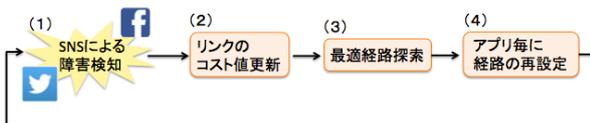


図 1: ソーシャル情報に基づく経路制御システム

- 1) Twitter による障害検知として、リアルタイムにツイートを監視し、通信障害に関係のあるツイート数と障害が発生した地名を取得することにより、どこで障害が起きているのかを検知する。
- 2) スイッチ間のコスト値の更新として、スイッチ間のコスト値のデフォルト値を 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 する。コスト値の更新は 60 秒間隔で行う。
- 3) 最適経路探索として、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、経路を決定する。
- 4) 経路の再設定として、OpenFlow の REST-API を使用し、コントローラから経路を自動で切り替える。

2.2 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。

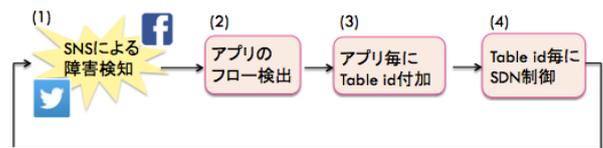


図 2: アプリケーション毎の SDN 制御システム

- 1) Twitter による障害検知として、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きているのかを検知する。
- 2) アプリのフロー検出として、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リを識別する。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 付加として、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 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) 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。

3. ソーシャル情報に基づく経路制御システム実験

3.1 実験

広域ネットワークテストベッド JGN-X 上の FLARE スイッチネットワーク上で、東日本大震災時 (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し本システムを動作させた。

図 3 に障害検知前と障害検知後のネットワークを示す。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。システムは、

SDN Control for Each Application by Virtualization Node FLARE in Large-Scale Disasters

[†] Tsumugi Tairaku, Masato Oguchi

[‡] Akihiro Nakao, Shu Yamamoto

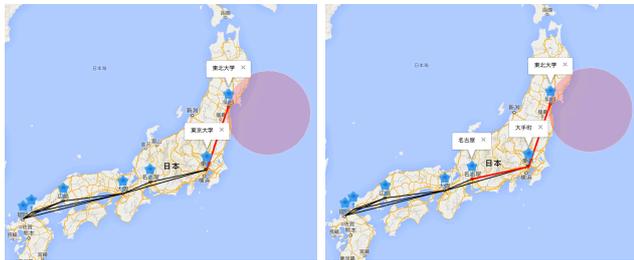
^{††} Saneyasu Yamaguchi

Ochanomizu University ([†])

University of Tokyo([‡])

Kogakuin University (^{††})

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確認できる。また、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ことを確認することができた。



障害検知前 障害検知後

図 3: 実験結果

3.2 性能評価

JGN-Xを模擬したFLAREノードネットワーク上で、本システムの性能評価を行う。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こるとされている。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、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。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の東京と宮城間にパケットロスを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。

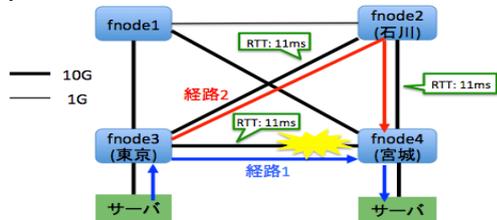


図 4: 性能評価に使用するFLAREノード

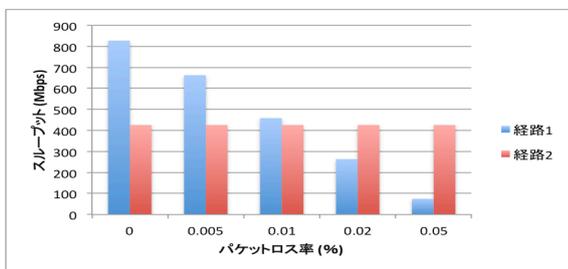


図 5: 経路1と経路2のスループット

パケットロスが0%の場合、経路1のスループットは経路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。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性能が良くなるのが分かる。

4. アプリケーション毎の制御システム実験

アプリケーション毎のSDN制御システム実験を、実際のアプリケーションを使用し行った。FLAREスイッチを経由して、外部のwebサーバ等に接続出来るようにProxyサーバを構築した。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い、また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、600Mbpsに帯域の制限を行った。

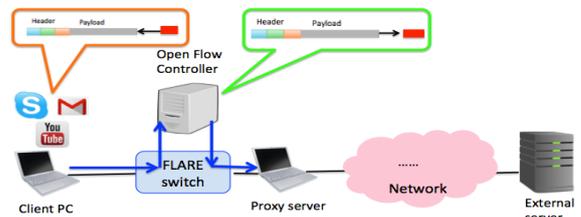


図 6: 実験用ネットワーク

帯域を制限していないSkypeは800Mbpsであるのに対し、帯域が制限されているYouTubeは400Mbps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行われていることが確認できる。

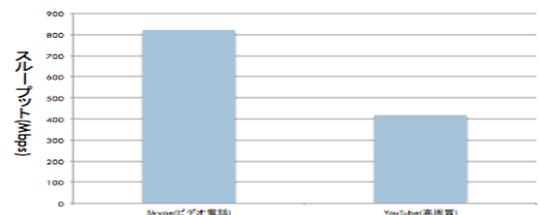


図 7: YouTubeとSkypeのスループット

5. まとめと今後の課題

Twitterの障害情報をもとに、障害を回避する経路に自動切替える仕組みを実験的に確認し、本システムの性能評価を行った。また、アプリケーション毎のSDN制御システムを実際のアプリケーションを使用し実験的に確認した。今後の課題としては、災害時のネットワークにさらに近いシナリオで実験を行う。

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。

参考文献

[1] 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