

DPN 環境におけるコンテキストに基づいたネットワークトラフィック制御方式

柳田 晴香 (指導教員：小口 正人)

1 はじめに

震災のような緊急時には、被災地域で通常の何倍にも及ぶ通信が集中することで輻輳が発生し、ネットワークの機能が著しく低下してしまうといった問題が生じる。緊急災害時には、ネットワークによる情報提供が最も重要であるにも関わらず、これが機能しなくなる事は致命的である。現状のネットワーク制御では、経路を二重化し、一方の経路で障害が発生したら、もう一方の経路に切り替えるといった単純な障害対策の経路制御や、トラフィックの急増が予想される場合に、回線やサーバを事前に増強するといった制御しか行われていない。これは予測が極めて困難な緊急災害時においては対応が難しい。さらに、膨大で多種多様なデータが飛び交うようになった現代のネットワークでは、人が状況を目で見て判断する静的な制御では限界となってきた。

そこで、本研究では細かい条件に応じた複雑で動的な制御を迅速に行うため、SDN(Software-Defined Networking) や OpenFlow プロトコルを用いて、ソフトウェアによる自動ネットワーク制御を行う。また、アプリケーションごとに緊急度の優先順位をつけるなどといった、より細かい制御を行うため、DPN(Deeply Programmable Network) の概念をも適用し、研究の新規性を保証する。

本研究は外部情報より障害を検知したと仮定し、それをトリガとして、トラフィックの最適化をアプリケーション別に自動で行う、高度なネットワークトラフィック制御手法の提案を行う。これにより、緊急災害時等でもユーザが安定して情報へアクセスできるシステムの構築を目指す。

2 提案システムの概要

本研究の提案システムの概要を図1に示す。動作は以下の通りである。

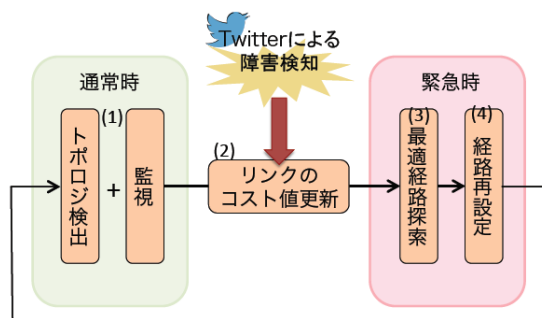


図1: 本研究の提案システムの概要

- (1) トポロジ検出と監視
スイッチのポートの接続状況を検出するアプリケーションを動かす、トポロジ情報の検出を行う。また、通常時において、トラフィックモニタによりネットワークに異常がないかの監視を行う。
- (2) リンクのコスト値更新
Twitter で検知した障害情報に従い、スイッチ間を結ぶリンクのコスト値を更新する。

- (3) 最適経路探索
更新されたコスト値を用いて、ダイクストラ法で最適経路を探索する。
- (4) 経路再設定
決定された最適経路に経路を再設定する。

また、これら一連の動作の自動化を行った。以上の提案システムをSDNやDPN環境で構築する。2.1節で具体的な例を交えつつ、その工程を説明する。

2.1 研究方針

上で提案したシステムを以下の環境において検証する。

- ①PC上のMiminet仮想環境(SDNの検証)
- ②ローカルのFLARE実機環境(DPNの検証)

まず、①のSDN環境における提案システムを、スイッチが3つのメッシュトポロジの場合(図2)で説明する。各スイッチの一つずつホストが接続されているが、この図では省略している。また、ホスト1からホスト2へとパケットが送信されているとする。

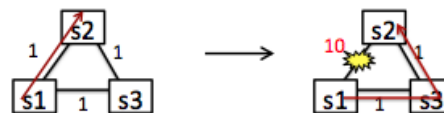


図2: SDN環境における提案システムの例

通常時のトポロジ検出によりコントローラはスイッチのトポロジ情報を保持している。この場合スイッチ3つが各スイッチ全てに繋がっており、それぞれのリンクのコスト値はデフォルトで1と設定される。ここで、s1-s2の経路で障害を検知したと仮定すると、s1-s2間のリンクのコスト値をトラフィック量に基づき10に更新する。そして、ダイクストラ法を実行すると、コスト最小の経路を探索するため、最適経路はコスト値2であるs1-s3-s2の経路と決定され、経路が再設定される。ここで、コスト値が小さいということは帯域が大きいということを示す。

②のDPN環境では、①で可能な制御に加え、図3や図4のような制御も可能になる。FLAREスイッチを用いたDPN環境では、仮想スイッチの組み合わせでスライスと呼ばれる仮想ネットワークを作成することができる。よって、図3のように、複数の仮想ネットワークをスライスとして構築しておき、特定の条件が満たされた場合に、スライスの切り換えを行うことによって最適な経路を選択することができる。

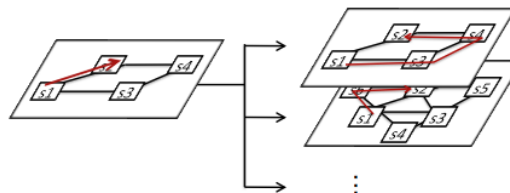


図3: DPN環境におけるスライスの切替

さらに、図4のように、緊急災害時において、現場の状況を伝えたり情報交換を行ったりするためのアプリケーションについては優先的にパケットを通し、エン

ターテインメント目的のトラフィックには制約をかけるなどといった制御も実現可能になる。

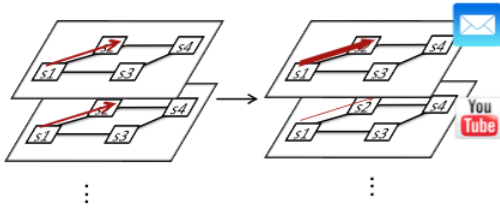


図 4: DPN 環境におけるアプリケーション毎の QoS の実現

3 Mininet を用いた制御プログラムの開発

今回、OpenFlow フレームワークの一つである Mininet エミュレータを使用し、PC 上の仮想環境で、実機実験の前に検証実験を行った。この実験の環境を表 1 に示す。

表 1: 開発環境

OS	ubuntu14.04 64bit
フレームワーク	Mininet 2.1.0p1
コントローラ	Ryu-manager 3.15
スイッチ	Open vSwitch 2.0.2

3.1 Mininet における実験

上記の環境で、3OVS ノードのメッシュトポロジを作成し、以下の 5 つの動作を確認した。

- (1) REST-API を使った手動経路切替え
REST-API を用いて、h1-s1-s2-h2 の経路と、h1-s1-s3-s2-h2 の 2 つの経路の切替スクリプトを作成し実行した。
- (2) トポロジ検出
コントローラがトポロジを検出できるように、REST-API でトポロジ検出アプリケーションを作成した。
- (3) トポロジ情報にコスト値を付加
(2) で検出したトポロジ情報に、コスト値を付加するスクリプトを作成した。
- (4) 最適経路探索
コスト値をもとに、ダイクストラ法による最適経路探索を行うアプリケーションを作成した。
- (5) 障害イベントによる自動経路切替え
以上により、例えば h1-s1-s2-h2 の経路に何か障害イベントが発生したことを仮定してコスト値を変更すると、自動的に h1-s1-s3-s2-h2 の s3 を経由する経路に切替えることを確認した。

動作イメージは図 5 の通りである。

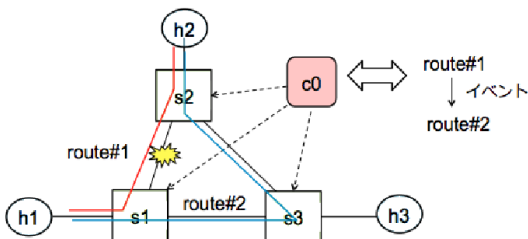


図 5: 障害イベントによる自動経路切替え

4 FLARE 実機実験

本研究では、図 6 のような物理構成で実機実験を行う。4 台の FLARE スイッチ各々に、端末がぶら下がる形である。FLARE Central は FLARE 管理用のサーバである。この FLARE Central サーバ上にコントローラを置いた。コントローラで 4 台の FLARE スイッチを制御し、様々なトラフィック制御モデルを検討する。

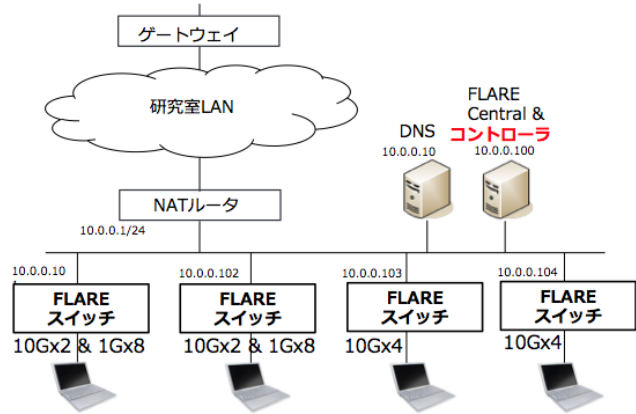


図 6: FLARE 実機実験環境

現段階では、Mininet で開発した OpenFlow のスクリプトをそのまま乗せて、3.1 節に示した実験と同等な実験をこの FLARE 実機環境で行うことができた。

5 まとめと今後の課題

緊急災害時に外部情報より障害を検知したと仮定し、その情報をトリガとして、ネットワークトラフィックの最適化をアプリケーション毎に自動で行う、高度なネットワークトラフィック制御システムを提案した。

現時点では、Mininet エミュレータとローカルの FLARE 実機上で、OpenFlow を用いた経路切替制御を自動で行うスクリプトを作成し、動作を確認した。今後は、FLARE 実機上で DPN を生かした制御の検証を行った後、広域なネットワークテストベッドである JGN-X (Japan Gigabit Network 第 4 世代) 上で実際に運用されているシステムに近い状況で検証していく。

今後の課題としては、さらに複雑なトポロジになった際に、この提案システムが正確に挙動するのか検証すると共に、より良い最適経路探索モデルを模索し、評価を行いたい。

謝辞

本研究を進めるにあたって、東京大学の中尾彰宏先生、山本周氏、工学院大学の山口実靖先生より大変有用なアドバイスをいただきました。深く感謝いたします。

参考文献

- [1] 柳田晴香, 中尾彰宏, 山本周, 山口実靖, 小口正人: 「DPN 環境におけるコンテキストに基づいたネットワーク制御方式」, DEIM2015, E1-3, 2015 年 3 月 発表予定