

OpenFlow を用いたネットワーク監視とトポロジーの可視化

秋田 海人^{†1} 長田 智和^{†2} 谷口 祐治^{†3}^{†1} 琉球大学工学部情報工学科 ^{†2} 琉球大学工学部工学科知能情報コース^{†3} 琉球大学総合情報処理センター

1 はじめに

大規模で複雑なネットワークをもつ組織では、ネットワーク全体の状態を詳細に把握することは容易ではない。例えば、障害発生時に障害発生箇所の特定や復旧に時間を要してしまう。このため、大規模なネットワークほど、運用管理の負担が増加する問題がある。

本研究では、ネットワークをソフトウェアで動的制御することができる OpenFlow 技術を用いて、GUI 操作によるネットワーク経路の変更や、Web アプリケーション上でのネットワークの状態監視及び論理トポロジーの可視化を実現するシステムを開発する。これによって、前述の問題を解決するとともに、OpenFlow の特徴を生かした柔軟なネットワーク構成の変更を可能とすることを目的とする。

2 関連研究と課題

コントローラの設定情報と論理トポロジーの視覚的な関連付けを可能とするネットワーク運用管理支援システムの研究として、藤田らの研究 [1] がある。この研究では、クライアントアプリケーションとコントローラから構成され、REST API を用いてコントローラの設定情報をクライアントアプリケーションが取得することで論理トポロジーを視覚的に把握することができる。しかし、ネットワーク運用管理支援システムに必要な障害発生時に対応する機能が存在しない問題点がある。

3 研究概要

本研究で提案するシステム構成図は図 1 の通りである。Web アプリケーションとコントローラを用い REST API によって通信を行う。また、本システムでは、データベースを用いて、論理トポロジーの情報、通信可能な経路情報、VLAN 設定情報を保持する。

Network Monitoring and Topology Visualization Using OpenFlow

Kaito AKITA^{†1} Tomokazu NAGATA^{†2}Yuji TANIGUCHI^{†3}^{†1}Faculty of Engineering and Information Engineering, University of the Ryukyus^{†2}Faculty of Engineering Computer Science and Intelligent Systems Program, University of the Ryukyus^{†3}Information Processing Center,

本システムの Web アプリケーションとコントローラの詳細について述べる。

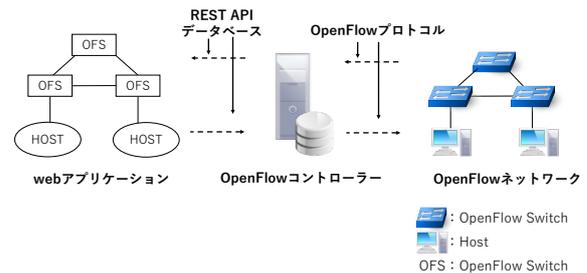


図 1: システム構成図

3.1 Web アプリケーション

Web アプリケーションは、利用者が設定を行う操作をした際にコントローラの設定に対応した API が動作する。その後、コントローラによりデータベースの情報を追加、削除、更新のいずれかを行い、設定変更後のデータベースの情報を Web アプリケーションは取得し、VLAN ごとの通信経路表示や通信経路の削除を可能にする。

Web アプリケーションでは、一定時間ごとにデータベースに格納されている論理トポロジーの情報を取得する。その後、Web アプリケーション上に取得した情報を基に論理トポロジーを表示する。さらに、経路制御を設定する。設定可能な項目として、VLAN 手動通信経路の選択と削除、自動通信経路設定、VLAN ごとの帯域幅制御、VLAN ごとのプロトコル指定及びその優先制御の 4 種類がある。また、VLAN ごとの通信経路表示やスイッチ間の遅延量の表示といった機能も有する。

3.2 コントローラ

本システムは、論理トポロジーの検出、スイッチ間の遅延量の測定、スイッチ間のリンク切断時の通信経路の変更に LLDP を用いる。論理トポロジーの検出は、LLDP パケットによって論理トポロジーの情報を取得し、データベースに書き込み情報を保持する。スイッチ間の遅延量の測定には、LLDP パケットに現在時刻のタイム

スタンプをつける、隣接したスイッチが発見された際に Packet-in の処理が発生した時刻と LLDP パケットに付加されていたタイムスタンプの時刻の差分を求めることによって遅延量を測定している。スイッチ間のリンク切断時の通信経路の変更は、一定時間ごとに接続の確認を行っており、LLDP パケットによって取得できなかったリンクがあった場合に通信可能な経路にダイクストラ法を用いて切り替えを行う。

4 評価実験

本システムの動作検証及び性能評価実験を行う。

4.1 動作検証

動作検証では、表 1 を基に正常に動作するか確認する。検証に用いる環境は、KVM と OpenFlow Switch で構築した仮想 OpenFlow ネットワークである。検証では、2 つの PC 上で行い、1 つは Web アプリケーションのみを動作させ、もう 1 つはコントローラーと OpenFlow ネットワークを動作させる。フルメッシュ型のトポロジを仮想 OpenFlow ネットワークとして構築し、動作検証を行なった。まず、LLDP パケットで取得した、OpenFlow ネットワークの論理トポロジを Web アプリケーション上に表示されることを確認した。そして、Web アプリケーション上から設定を行い、コントローラーから OpenFlow ネットワークに設定が正常に反映されることを確認した。動作検証の結果、本システムの全ての機能が動作することを確認した。

表 1: 設定した項目及び確認方法

設定	操作	確認事項
VLAN 手動通信経路の選択及び削除	まず、VLAN の番号を決定する その後、通信させたい 2 個のホストのうち 1 つを選択し、経路を選択後残っているホストを選択する それにより、ホスト間の通信が可能になる また、削除を行うことにより、通信が不可能になる	同一 VLAN 上のホスト同士で ping が到達することに加えて、経路情報が存在するホスト間で ping が到達すること 異なる VLAN 上のホスト同士で ping が到達不可であることに加えて、経路情報が存在しないホスト間で ping が到達不可能であること
自動通信経路設定	まず、VLAN の番号を決定する その後通信させたい 2 個のホストを入力をする	同一 VLAN 上のホスト同士で ping が到達することに加えて、経路情報が存在するホスト間で ping が到達すること また、その通信経路が最短経路であること
プロトコル指定による優先制御	2 個のプロトコルに優先度の差をつけ、両方のプロトコルのパケットを同時に送す	優先度が高いプロトコルのパケットが優先して処理・転送されること
VLAN ごとの帯域幅制限	VLAN ごとに帯域幅の制限値を設定する	設定を施した経路上に制限値を超えるパケットを流した際、パケット流量が制限値を超えないこと

4.2 性能評価

性能評価実験では、コントローラーが制御する OpenFlow スwitch の台数が増加した時、遅延量の変化、REST API の応答時間、論理トポロジの検出の 3 種類について及ぼす影響とフルメッシュ型の論理トポロジを用いてリンク切断時の経路切替時間の計測の検証をする。

3 種類の設定項目について、OpenFlow スwitch の台数を増加させ、最大 50 台まで繰り返し計測した。結

果、遅延量は 50ms の状態から約 1000ms の状態になり、正常な値を表示することができなくなった。REST API の応答時間と論理トポロジの検出は、1 台の時と 50 台の時の差は約 0.05 秒以下となり、ほとんど遅延が発生しなかった。リンク切断時の経路切替は、あらかじめ通信経路を設定し、ホスト間で通信ができる状態にしておく。そして、通信可能な経路を 1 つ切断し通信の切替までの時間を計測を行なった。実験では、10 秒ごとに断線検知を行う条件にしていた。そのため、断線検知までに約 10 秒程度かかり、その間は通信不可能な状態になる (図 4)。これらより、OpenFlow スwitch の台数の増加が本システムに及ぼす影響はわずかであることがわかった。また、リンク切断時の経路切替では、通信不可能な時パケットロスが発生したが、通信経路を切替られることが確認できた。

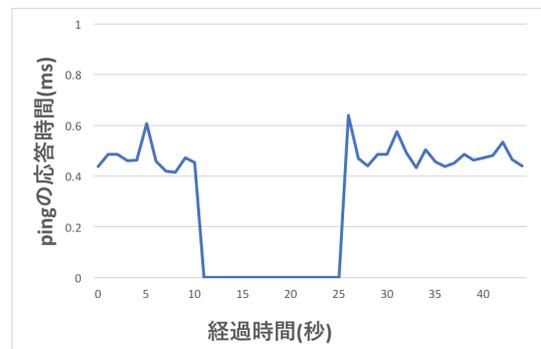


図 2: リンク切断時による経路切替時間

5 まとめ

本研究では、ネットワークをソフトウェアで動的制御することができる OpenFlow 技術を用いて、GUI 操作によるネットワーク経路の変更や、Web アプリケーション上でのネットワークの状態監視及び論理トポロジの可視化を実現するシステムを開発した。本システムは、遅延量の表示や断線検知をすることで障害発生時の障害発生箇所特定や一時的に通信経路を確保することで運用管理者の支援を行う機能を実現した。さらに、GUI 操作によるネットワーク経路の変更や、Web アプリケーション上の論理トポロジの可視化により容易に通信経路の変更やネットワークの構成を把握することができた。

参考文献

[1] 藤田紘生 他, “コントローラの設定情報と論理トポロジの視覚的な関連付けを可能とする OpenFlow ネットワーク運用管理支援システム”, 第 79 回情報処理学会全国大会論文集, pp.168-169, 2017