G-23

# トポロジの可視化と直感的なタッチ操作による素早いネットワーク制御を可能とするネットワーク運用管理支援システム

# A System for Supporting Management and Operation of Network by Visualization of Topology and Touch Operation

藤田 紘生†　　井口 信和‡
Hiroki Fujita　　Nobukazu Iguchi

## 1. はじめに

仮想化技術やクラウドサービスの普及に伴い，ネットワークに対する要件が変化している．例えば仮想化技術により，仮想サーバなどの仮想機器の追加，移動，削除が容易に行えるようになった．このような環境の変化により，ネットワークの構成や設定にも変更が必要となる場合がある．従来型のネットワークでは，設定の変更が必要となるたびに，ルータやスイッチといったネットワーク機器ごとに設定を施す必要がある．そのため，仮想機器の追加，移動，削除の回数が増加するにつれて，ネットワーク機器の設定変更に多大な時間が必要となり，対応することが困難となる．そこで，より柔軟にネットワークを制御できる仕組みが求められている．

以上の背景から，ネットワークを柔軟に制御できるコンセプトとして，Software-Defined Networking(SDN)が注目されている[1-5]．従来型のネットワーク機器には，経路制御部に相当する OS やソフトウェアと，パケット転送部に相当するハードウェアの両方が搭載されている．OS やソフトウェアの実装はネットワーク機器のベンダに依存するため，提供される機能もベンダによって異なる．またネットワーク機器はそれぞれ独立した OS により動作しており，設定も機器ごとに施す必要がある．一方 SDN では，ネットワーク機器から経路制御部を切り離すことでベンダ依存を解消している．また，コントロールプレーンが一元的にネットワークを管理するため，機器ごとに設定を施す必要がなくなる．ネットワーク管理者はコントロールプレーンを自由に開発できるため，ネットワーク要件に応じた柔軟なネットワークを構築できる．

一般的に，ネットワークの障害には迅速な対応が求められる．SDN によりネットワークに柔軟性を持たせることで，ある程度の障害を予測し，自動的に対応できるネットワークを構築することが可能である．しかし，想定外の障害が発生した場合，被害を最小限に抑えるため，迅速な対応が必要である．対応が遅れると，被害が急速に拡大し，長時間にわたり可用性に問題が生じる可能性がある．SDN によりネットワークに柔軟性を持たせることが可能となった一方で，ネットワークアーキテクチャは従来と比較して複雑化している．例えば，物理ネットワークの上に仮想的にネットワークを構築するオーバーレイネットワークでは，ネットワークのトポロジが物理トポロジと全く異なる場合もある．このとき，ネットワークの把握や設定変更に時間を要する場合があり，障害への対応が遅れる可能性がある．

そこで本研究では，トポロジの可視化と直感的な操作による素早いネットワーク制御を可能とするネットワーク運用管理支援システム（以下，本システム）を開発する．本システムは，タッチディスプレイ上にネットワークのトポロジを表示し，ネットワーク管理者がネットワークのトポロジを容易に把握することを可能にする．また，トポロジが表示されているタッチディスプレイをタッチ操作することで，直感的な操作で素早くネットワークの構築や設定変更ができる．本システムにより，ネットワークに想定外の障害が発生したとき，ネットワークのトポロジを容易に把握でき，また障害への対応を素早く行える．本稿では，本システムを実装し，実験方法と評価方法の検討を行う．

本稿の構成は次の通りである．まず，2 章で本研究と関連する研究について述べ，本研究との比較を行う．次に，3 章で本研究に関連する技術ついて述べる．4 章，5 章，6 章で本システムの詳細を述べる．7 章で実験および評価方法の検討について述べる．8 章でまとめと今後の課題を述べる．

## 2. 関連研究

本章では，本研究と関連する研究やシステムを挙げ，本システムとの比較を行う．

大阪大学の渡場氏らのシステム[6]は，本システムと同様に OpenFlow ネットワークの可視化を行う．また，OpenFlow スイッチ間の結線のトラフィック量を表示する機能や，経路情報の直接入力によりネットワークを制御する機能を備えている．渡場氏らのシステムでは，経路情報を入力するとき，システムの使用者がルーティングの条件を詳細に入力する必要がある．これに対して，本システムでは直感的なタッチ操作により，短い手順で経路の追加や削除ができる．そのため，予想外のネットワーク障害などが発生した場合に，より素早い対応が可能となる．

琉球大学の秋田氏らのシステム[7]は，本システムと同様に，OpenFlow ネットワークの可視化を行う．秋田氏らのシステムでは，Web アプリケーションがネットワークのトポロジを可視化する．また，Web アプリケーションから，VLAN 設定などの 4 種類のネットワーク設定を施すことができる．秋田氏らのシステムは，VXLAN を使用しないため，ネットワークを論理的に分割できる数は VLAN の限界である約 4000 である．これに対し本システムは VXLAN に対応しているため，約 1600 万までネットワークを論理的に分割できる．また秋田氏らのシステムは，物理トポロジ上のネットワーク機器に対して設定を施すため，ネットワークの構成は物理トポロジに依存する．これに対し，本シス

† 近畿大学大学院 総合理工学研究科, Graduate School of Science and Engineering Research, Kindai University

‡ 近畿大学 理工学部 情報学科，Faculty of Science and Engineering, Kindai University

テムは VXLAN によるオーバーレイネットワークを構築するため，物理トポロジに依存しない柔軟なネットワークの構築ができる．

## 3. 関連技術

本章では，本システムを使用する上で前提となる技術について述べる．

### 3.1 OpenFlow

OpenFlow[8]は，SDN を実現する技術のひとつである．OpenFlow ネットワークでは，コントロールプレーンとデータプレーンはそれぞれ OpenFlow コントローラと OpenFlow スイッチに相当する．OpenFlow コントローラは OpenFlow ネットワークの経路制御を一元的に管理するソフトウェアである．OpenFlow スイッチは，OpenFlow における経路情報であるフローエントリに従いパケットを転送するネットワーク機器である．ネットワーク管理者は，ソフトウェアである OpenFlow コントローラを自由に実装できるため，ネットワークの要件に応じた柔軟な経路制御が可能となる．

本システムでは，タッチディスプレイからネットワークを一元的に管理するために OpenFlow を使用する．

### 3.2 VXLAN

VXLAN[9]は，トンネリングにより Layer 3 ネットワーク上に Layer 2 ネットワークを構築するプロトコルである．VXLAN は RFC7348 で標準化されている．VXLAN で構築されたネットワークの一例を図 1 に示す．VXLAN では，L3 ネットワークを通過するパケットをカプセル化する．L2 ネットワークと L3 ネットワークの境界のネットワーク機器は VXLAN Tunnel End Point(VTEP)と呼び，VTEP がパケットのカプセル化，非カプセル化を行う．VXLAN では，このカプセル化により L3 ネットワーク上にオーバーレイ型の L2 ネットワークを構築する．VXLAN は，VLAN と同様にネットワークを論理的に分割できる．各 VXLAN セグメントは 24 ビットの VXLAN Network ID(VNI)により識別される．VLAN のネットワーク分割数の上限が約 4000 であることに対し，VXLAN では約 1600 万のセグメントにネットワークを分割できる．これにより，仮想化技術の普及による L2 ネットワークの増加に対応できる．

本システムでは，物理トポロジに依存しない柔軟なネットワークの構築を可能とするため，VXLAN を使用する．

## 4. システム概要

本システムの構成を図 2 に示す．本システムは，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る．

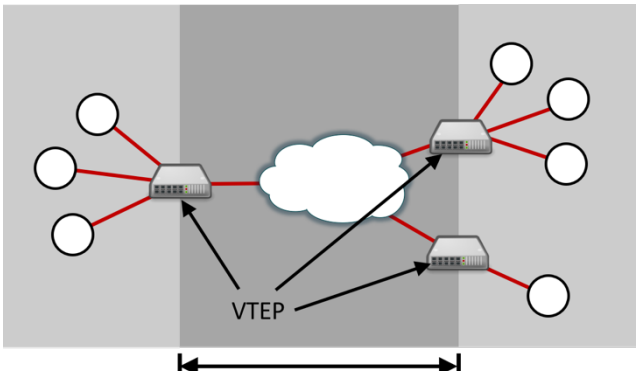


図 1：VXLAN の一例

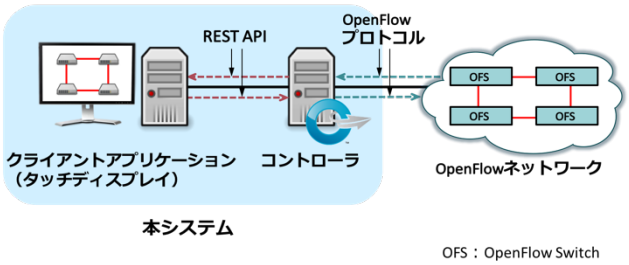


図 2：システム構成

クライアントアプリケーションはタッチディスプレイ上で動作し，OpenFlow ネットワークのトポロジの表示と直感的なタッチ操作によるネットワーク制御を可能とする．コントローラは VXLAN によるオーバーレイ型の OpenFlow ネットワークを制御する．次に，クライアントアプリケーション，コントローラ，そして本システムの管理対象となる OpenFlow ネットワークの詳細を述べる．

### 4.1 クライアントアプリケーション

クライアントアプリケーションはタッチディスプレイ上で動作し，ネットワークを管理するための GUI をネットワーク管理者に提供する．クライアントアプリケーションは，タッチディスプレイにトポロジを表示するとき，OpenFlow ネットワークの情報をコントローラへ要求し，取得する．取得した OpenFlow ネットワークの情報を基に，タッチディスプレイ上に OpenFlow ネットワークのトポロジを表示する．また，トポロジに対するタッチ操作を読み取って解釈し，ネットワーク設定の変更要求としてコントローラへ送信する．

### 4.2 コントローラ

本システムのコントローラは，OpenFlow における OpenFlow コントローラの役割を果たす．コントローラは，クライアントアプリケーションから受信した要求に従い，OpenFlow ネットワークの経路制御を行う．クライアントアプリケーションから OpenFlow ネットワークの情報の要求を受信すると，コントローラは自身が保持している OpenFlow ネットワークの情報をクライアントアプリケーションへ送信する．また，OpenFlow ネットワークの設定変更の要求を受信すると，要求に従い OpenFlow スイッチの設定を書き換える．

### 4.3 OpenFlow ネットワーク

本研究で管理する OpenFlow ネットワークは，VXLAN を用いたオーバーレイネットワークである．VXLAN により，物理トポロジに依存しない柔軟なネットワークを構築することが可能となる．

## 5. トポロジ表示機能

本章では，本システムが持つ機能のうち，タッチディスプレイに OpenFlow ネットワークのトポロジを表示する機能について述べる．

本機能によりタッチディスプレイにトポロジを表示するときの動作の流れを図 3 に示す．前提として，コントローラは，制御する OpenFlow ネットワークの情報を Link Layer Discovery Protocol(LLDP)により常に収集しているものとする．クライアントアプリケーションが起動すると，コントローラに対して OpenFlow ネットワークの情報を要求する．コントローラはこの要求を受け取ると，自身が LLDP により収集している OpenFlow ネットワークに関する情報のう

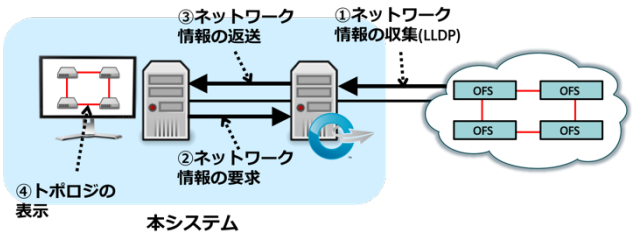


図 3：トポロジ表示の流れ

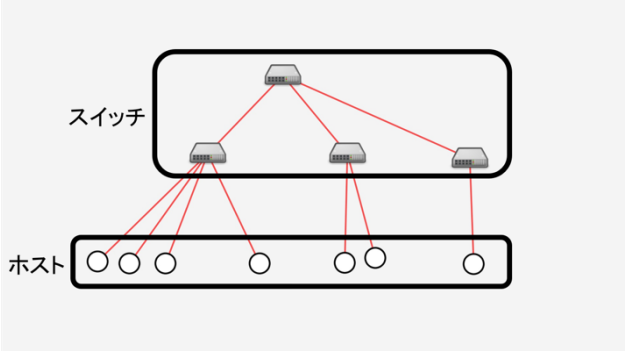


図 4：トポロジが表示されている様子

ち，トポロジを表示するために必要なものをクライアントアプリケーションへ返送する．クライアントアプリケーションは受け取った情報を基に，タッチディスプレイ上に OpenFlow ネットワークのトポトジを表示する．トポロジがタッチディスプレイ上に表示されている様子を図 4 に示す．図の上部に表示されている台形のネットワーク機器が OpenFlow スイッチを表し，下部に表示されている円がネットワークのホストを表している．また，ノード間の実線は論理経路を表しており，実線が存在するノード間は通信が可能である．クライアントアプリケーションは，タッチディスプレイ上にトポロジが表示された後も，一定時間ごとにコントローラへ OpenFlow ネットワーク情報の要求を繰り返すことで，トポロジを更新し続ける．このときの要求の間隔は，管理するネットワークやコントローラの負荷に応じて，ネットワークの管理者が自由に設定できる．

本機能により，管理するネットワークのトポロジが可視化されるため，ネットワーク管理者は管理するネットワークを容易に把握できる．

## 6. ネットワーク設定機能

本章では，本システムが持つ機能のうち，タッチ操作によりネットワーク設定を行う機能について述べる．

本機能によりネットワークの設定を書き換えるときの動作の流れを図 5 に示す．ネットワーク管理者が設定を変更するタッチ操作をすると，クライアントアプリケーションはタッチ操作を解釈し，ネットワーク設定を変更する要求をコントローラへ送信する．コントローラは設定を変更する要求を受け取ると，管理する OpenFlow ネットワークの設定を書き換える．書き換えが問題なく完了すると，コントローラはクライアントアプリケーションへ設定完了の通知を返す．クライアントアプリケーションは，この通知を受け取ると，タッチディスプレイ上のトポロジを更新する．

本機能により，ネットワーク管理者は直感的なタッチ操作による素早いネットワークの構築や設定変更を行うことが可能となる．次に，本機能が対応している設定と，設定するときのタッチ操作の方法について述べる．

### 6.1 設定モードの切り替え

本システムでは，タッチ操作のみで論理経路の追加・削除や，VXLAN セグメントの作成などのネットワーク設定を施すことができる．単純なタッチ操作のみで多様な設定をできるようにするため，設定モードを変更しながら操作を行う．各設定モードで施すことができる設定は 1 種類もしくは 2 種類となっており，施したい設定に合わせて設定モードを変更する．設定モードを変更するには，まずディスプレイを 2 本の指でタッチする．すると図 6 に示すように，タッチした点を中心に円が表示される．円周上には各モードの名称が並んでおり，変更したいモードの上まで指を動かし，ディスプレイから指を離すことで，設定モードの変更ができる．

### 6.2 ノードの移動

ノードの移動は Arrange モードで行う．ノードの移動では，トポロジ上に表示されたスイッチやホストなどのノードを並び替えることで，トポロジを理解しやすい形に整列させることができる．ノードの移動をする様子を図 7 に示す．ノードを移動させるには，移動させたいノードに対して 1 本の指でタッチし，そのままスライドさせる．そのまま画面上で指をスライドすることにより，ノードを画面上

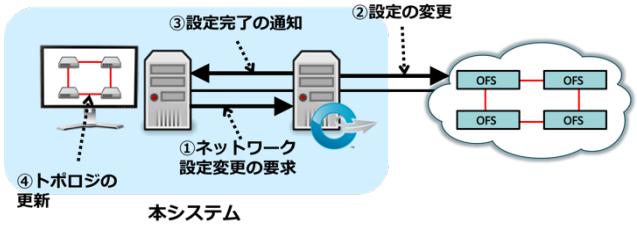


図 5：設定変更の流れ

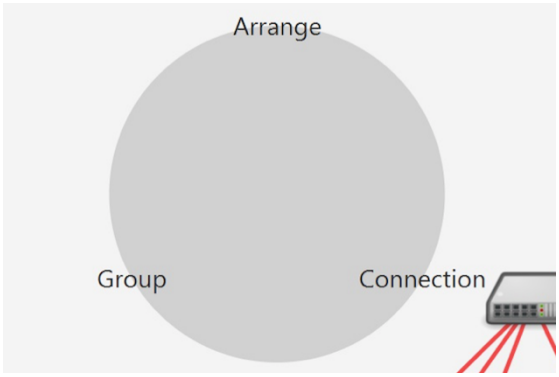


図 6：モードの変更


図 7：ノードの移動


図 8：スイッチの追加

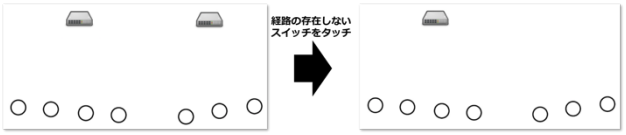


図 9：スイッチの削除

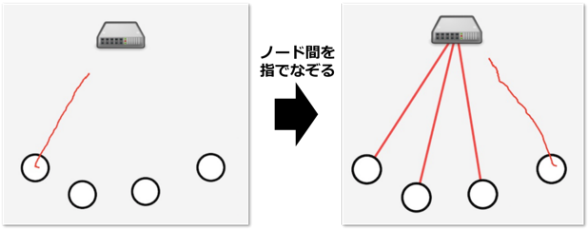


図 10：論理経路の追加

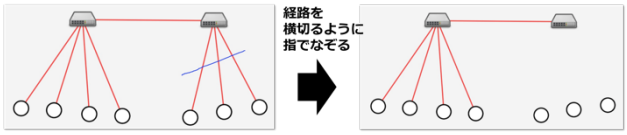


図 11：論理経路の削除


図 12：VXLAN セグメントの作成

の任意の位置に移動させられる．これにより，ネットワーク管理者はノードを理解しやすい形に自由に整列できる．

### 6.3 仮想スイッチの追加・削除

仮想スイッチの追加は，Arrange モードで行う．仮想スイッチの追加をする様子を図 8，削除する様子を図 9 に示す．仮想スイッチを追加するには，画面上の空白部分に 1 本の指でタッチをする．仮想スイッチを削除するには，論理経路が存在しない仮想スイッチに対してタッチする．

### 6.4 論理経路の追加・削除

論理経路の追加・削除は，Connection モードで行う．このモードでは，任意の 2 つのノード間において，論理経路の追加・削除ができる．

論理経路を追加する様子を図 10 に示す．論理経路を追加する場合は，まず，論理経路を追加する 2 つのノードのうち一方を 1 本の指でタッチする．そのまま指をスライドさせ，他方のノードに指を運んだ後に，画面から指を離す．以上のタッチ操作により，2 つのノード間に論理経路が追加され，そのノード間での通信が可能となる．

論理経路を削除する様子を図 11 に示す．論理経路を削除する場合は，まずトポロジ上の空白を 1 本の指でタッチする．そのまま削除したい経路を横切るように指でなぞった後，画面から指を離す．以上のタッチ操作により，指が横切った論理経路を全て削除できる．削除された論理経路は使用できなくなり，その経路を使用する通信が全て遮断される．

### 6.5 VXLAN セグメントの作成

VXLAN セグメントの作成は Group モードで行う．このモードでは，1 つの OpenFlow スイッチを起点として，VXLAN セグメントを作成できる．起点となるノードは，その VXLAN におけるゲートウェイの役割を果たす．VXLAN セグメントを作成する様子を図 12 に示す．まず，VXLAN の起点となるノードに 1 本の指でタッチする．そのまま，VXLAN の中に含めたいノードを囲うように，指で閉路を描く．指が再び起点となるノードと重なったら，指を離す．以上の操作により，指で描いた閉路に囲まれたノードが全て含まれる VXLAN セグメントが生成される．

## 7. 実験および評価の検討

本章では，本システムを評価するために予定している動作検証，性能評価実験，利用評価実験について述べる．

### 7.1 前提

前提として，全ての実験に用いる共通の環境を述べる．実験は全て単一の PC 上で行うため，今回の実験では，クライアントアプリケーション，コントローラ，OpenFlow ネットワークの各間の物理的距離による応答時間等の影響は考慮しない．実験の環境を図 13 に示す．ホスト PC(CPU:Intel(R) Core(TM) i5 3.2GHz*2， RAM:16.0GB，OS:Windows10 Home Edition 64bit)上に，VirtualBox を用いてゲスト OS(Ubuntu 14.04 64bit)を立ち上げる．ホスト PC 上でクライアントアプリケーションを動作させ，ゲスト OS 上でコントローラおよび仮想 OpenFlow ネットワークを動作させる．OpenFlow スイッチには，仮想スイッチである Open vSwitch[10]を用いる．以上の環境で，クライアントアプリケーションとコントローラ，コントローラと仮想 OpenFlow ネットワークをそれぞれ接続する．

### 7.2 動作検証

動作検証では，本システムが有する全ての機能が仕様通りに動作することを検証する．

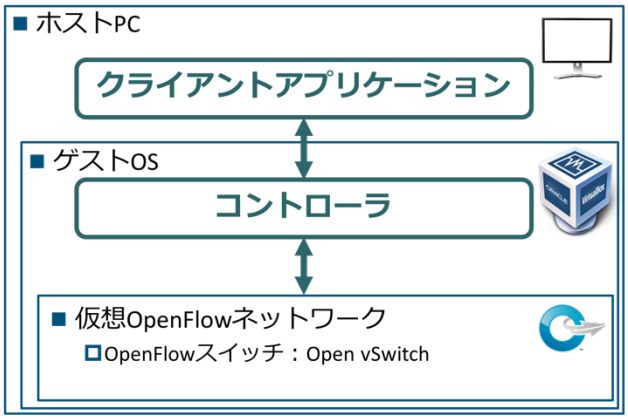


図 13：実験環境

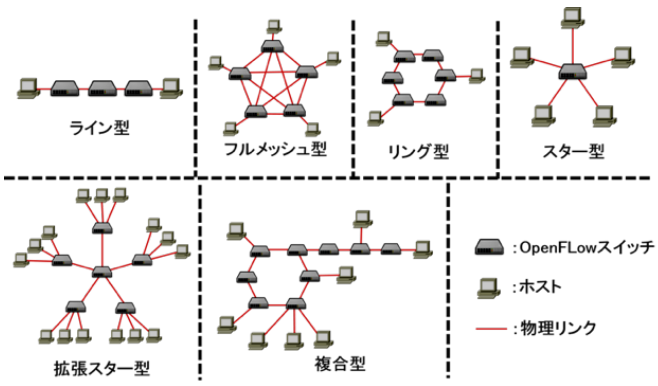


図 14：動作検証を行う物理トポロジ

#### 7.1.1 実験環境

検証に用いるネットワークとして，図 14 に示す 6 種類の物理トポロジのネットワークを用意する．これらのネットワーク上に本システムによるオーバーレイ型の OpenFlow ネットワークを作成し，その上で検証を行う．

#### 7.1.2 手順

まず，トポロジ表示機能の動作検証を行う．クライアントアプリケーションが起動したとき，ネットワーク上に存在するホストが全てタッチディスプレイに表示されることを確認する．続いて，ネットワーク設定機能の動作検証を行う．本システムを用いてネットワーク設定を行い，その結果がネットワークに正しく反映されていることを確認する．ホスト間で ping の疎通確認を用いて，論理経路が存在する経路のみで通信が可能であることと，同一の VXLAN セグメント内でのみ通信が可能であることを確認する．以上の手順を，用意した全ての物理トポロジ上で同様に行う．

#### 7.1.3 評価

検証の結果，用意した全ての物理トポロジで本システムの全ての機能が仕様通りに動作することを示す．これにより，物理トポロジによらず，本システムが仕様通りに動作することを示す．

### 7.3 性能評価実験

性能評価実験では，本システムの応答時間の計測や，本システムが制御できるネットワークの最大規模の推定を行う．

#### 7.2.1 実験環境

実験には，スター型トポロジを用いる．また，トポロジ表示機能におけるネットワーク情報の要求の間隔は 5 秒とする．ホストの数は 10 台から開始し，1 回の実験が完了するごとに 10 台ずつ増加させながら，ホストの数が 100 台になるまで実験を繰り返し行う．

#### 7.2.2 手順

まず，クライアントアプリケーションが起動してからネットワーク上のホストが表示されるまでの時間を計測する．その後，本システムを用いてネットワーク設定を行い，反映されるまでにかかる時間を計測する．ここでは，論理スイッチの追加，論理経路の追加，VXLAN セグメントの作成の 3 種類の設定について計測を行う．

#### 7.2.3 評価

計測した時間をグラフに表し，ホストの増加に対する応答時間の変化の割合を算出する．これにより，本システムがどの程度のホスト数に対応できるかを推定する．

### 7.4 利用評価実験

利用評価実験では，被験者を用意し，本システムを用いた設定変更に要する時間を計測する．実験では，ネットワークに予期せぬ障害が発生した場合を想定し，障害が発生した場所をネットワークから遮断するまでに要する時間を計測する．

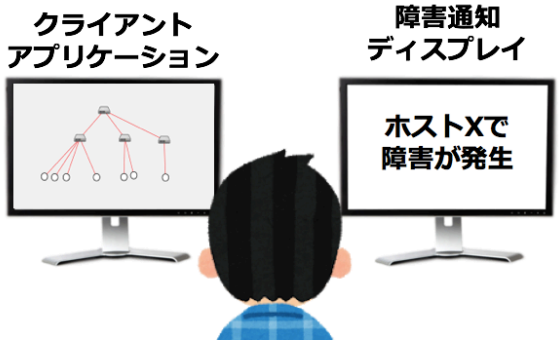


図 15：利用評価実験

#### 7.3.1 実験環境

本実験は，本研究室の学生 10 名を対象に行う．本実験には，あらかじめ本システムにより設定された OpenFlow ネットワークを使用する．実験環境の様子を図 15 に示す．実験の前に，被験者はネットワークのトポロジを確認しておく．また，トポロジを表示するタッチディスプレイとは別に，ネットワークに障害が発生したことを通知するためのディスプレイ（以下，障害通知ディスプレイ）を横に並べて設置する．

#### 7.3.2 手順

被験者は，タッチディスプレイと，障害通知ディスプレイの両方を確認できる位置に待機する．この状態で，ネットワークに障害が発生したことと，障害発生地点を知らせる文章を予告なしに障害通知ディスプレイに表示する．被験者は通知された障害に対して，本システムの設定変更機能を用いて，障害発生箇所とそうでない箇所の切り分けを行う．この時，障害発生ディスプレイに障害は発生した瞬間から，障害発生地点の切り分けが完了するまでの時間を計測する．

#### 7.3.3 評価

計測した時間の平均と標準偏差を求め，想定外の障害が発生してから，本システムにより障害発生部をネットワークから遮断するまでにどの程度の時間を要するかを推定する．また，他のシステムを使用した場合や，システムを利用しない場合に同様の実験を行った場合，どの程度の時間を要するかを計測し，本システムを用いた場合との比較を行う．比較により，本システムがネットワーク障害への素早い対応が可能であるかどうかを評価する．

## 8. おわりに

本稿では，トポロジの可視化とネットワーク障害への素早い対応を目的とし，トポロジの可視化と直感的な操作による素早いネットワーク制御を可能とするネットワーク運用管理支援システムを開発した．本システムは，OpenFlow と VXLAN のふたつの技術を用いることにより，物理トポロジに依存しない柔軟なネットワークを構築できる．また，直感的なタッチ操作により，ネットワーク設定の素早く変更できる．

今後は，本稿で検討した実験について，より詳細に実験環境や条件の設定を行い，実施する．その後，得られた実験結果から本システムの評価と考察を行う．

## 参考文献

[1] B. A. A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, “A Survery of Software-Defined Networking: Past, Present, and Future of Programmable Networks,” IEEE Communications Surveys & Turtorials, vol.16, no.3, pp.1617-1634, 2014.

[2] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-Defined Networking: A Comprehensive Survey,” Proceedings of the IEEE, vol.103, no.1, pp.14–76, 2015.

[3] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, “Survey on Network Virtualization Hypervisors for Software

Defined Networking," IEEE Communications Surveys & Tutorials, vol.18, no.1, pp.655–685, 2016.

[4] Y. Li and M. Chen, "Software-Defined Network Function Virtualization, A Survey," IEEE Access, vol.3, pp.2542–2553, 2015.

[5] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software-Defined Networking," IEEE Communications Surveys & Tutorials, vol.17, no.1, pp.27–51, 2015.

[6] Y. Watashiba, S. Hirabara, S. Date, H, Abe, K. Ichikawa, Y. Kido, S. Shimojo, and H. Takemura, "OpenFlow Network Visualization Software with Flow Control Interface," 2013 IEEE 37th Annual Computer Software and Applications Conference, Kyoto, pp. 475-477, 2013.

[7] 秋田 海人, 長田 智和, 谷口 祐治, "OpenFlow を用いたネットワーク監視とトポロジーの可視化, " 第 80 回情報処理学会全国大会論文集, pp.197-198, 2018

[8] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," ACM SIGCOMM Computer Communication Review, vol.38, no.2, pp.69-74, 2008.

[9] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks," RFC 7348, 2014, https://tools.ietf.org/html/rfc7348 (参照 2018/7/27)

[10] J. Pettit, J. Gross, B. Pfaff, M. Casado, S. Crosby, "Virtual Switching in an Era of Advanced Edges," 2nd Workshop on Data Center - Converged and Virtual Ethernet Switching, ITC 22, Sep. 6, 2010.