

# SDNによるネットワーク構築演習における 学習者の行動履歴収集

横山 貴志<sup>1,a)</sup> 百瀬 拓也<sup>1,b)</sup> 新村 正明<sup>1,c)</sup> 國宗 永佳<sup>1,d)</sup>

**概要:** 近年, Software Defined Networking(SDN) と呼ばれる技術が注目を集めている. SDN が普及するにつれ, さまざまな教育機関で SDN によるネットワーク構築演習が行われている. 演習形式の講義において, 学習者の状況把握を行なうことは有用とされ, プログラミング演習や従来のネットワーク構築で多くの研究が行われている. しかし SDN によるネットワーク構築演習において, 学習状況把握に関する研究は例が少ない. 本稿ではつまづいている学習者の発見を容易とすることを目的とし, 学習状況把握に関する関連研究から SDN におけるネットワーク構築演習において有用である学習履歴について精査し, それらの学習履歴の収集を行なうシステムを試作した.

**キーワード:** SDN, 演習, 学習履歴可視化

## SDN

TAKASHI YOKOYAMA<sup>1,a)</sup> TAKUYA MOMOSE<sup>1,b)</sup> MASAOKI NIIMURA<sup>1,c)</sup> HISAYOSHI KUNIMUNE<sup>1,d)</sup>

**Abstract:** Recently, much attention has been paid to the Software Defined Networking (SDN). As spread of SDN, network construction practices using SDN is carried out in some educational institutions. Since it is considered that learning states gathering is useful, and there are many studies about this in programming practice and legacy network construction practice, but few studies in SDN. In this study, we focus to detect students who having setback. We select some informations to detect these students from relevant studies, and conduct an investigation for the useful learning state in network construction practice using SDN. Subsequently, we produced the system which gathered them experimentally.

**Keywords:** SDN, practice, learning states visualization

## 1. はじめに

近年, クラウドサービスの普及に伴い, ネットワークトラフィックが増大している. そのため, Software Defined Networking(SDN)[1] と呼ばれる技術によってネットワーク機器の振る舞いを制御し, ネットワーク構成を最適化する手法が取られている. SDN の普及に伴って技術者の需要が増えるため, SDN によるネットワーク構築演習が必要

となる. プログラミング演習や従来のネットワーク構築演習では, 学習状況把握が有用であるとされ多数の研究が存在する. 本研究では, SDN によるネットワーク構築演習を対象とした学習状況把握を行う.

## 2. 研究背景

### 2.1 Software Defined Networking

近年, ネットワークに対する要求が変化している. クラウドサービスの普及に伴って仮想化技術も進歩し, サーバの生成や起動, 停止が動的に行われるようになった. そのため, ネットワーク機器の設定をサーバの変化に合わせて随時行なうことが求められている. また, センサやスマー

<sup>1</sup> 信州大学

Shinshu University

a) yokoyama@seclab.shinshu-u.ac.jp

b) momose@seclab.shinshu-u.ac.jp

c) niimura@cs.shinshu-u.ac.jp

d) kunimune@cs.shinshu-u.ac.jp

トフォンからの多種多様なデータによるネットワークトラフィックの増大に対して、効率的にデータを転送する必要がある。これらの問題に対して、従来のネットワーク構造では対応が困難になり、動的で迅速なネットワーク構築を行なう技術が求められている。その要求を満たす様々な手段が考案されているが、その一つに Software Defined Networking(SDN) が存在する。

SDN はソフトウェアによってネットワークを制御する技術である。SDN の利点として以下の点が挙げられる。

- プログラミングによるネットワーク機器の動作設定
- 複数ネットワーク機器の一括制御

従来はネットワークデバイスベンダが作成した機能を指定して通信の制御を行っていた。しかし SDN では、制御部分を使用者がプログラミングする事が可能である。よって従来のネットワークと比べ、柔軟なネットワークを構築可能となった。

また従来のネットワークは、それぞれのネットワークデバイスが制御部と転送部を有していた。SDN ではネットワークデバイスは転送部のみを有し、制御部は他のコンピュータが担う手法を採る。これにより、複数のネットワークデバイスを単一のソフトウェアで制御する事が可能となった。

従来のネットワーク機器では、ネットワーク機器が対応していないと、時間別トラフィックの統計情報や IDS など他システムとの連携ができなかった。その対応を行うのはベンダであり、使用者は対応しているネットワーク機器を使用する他なかった。しかし、先述したように SDN の制御部はアプリケーションであるため、使用者が設定を行って通信制御を行なう際に外部アプリケーションと連携することが可能である。これにより、QoS 向上やネットワーク構成の最適化などを、より動的に行なう事ができる。

Google がデータセンター間ネットワークを 100%SDN に置き換えるなど [2][3]、通信事業者やクラウドサービスを提供している企業が、サービス競争力の向上やコストの削減を目的として SDN を導入する事例も増えている。

## 2.2 OpenFlow

SDN を実装する通信プロトコルとして OpenFlow がある。OpenFlow は通信ルールを「フロー」と規定し、OpenFlow に則ったスイッチはフローを複数保持する「フローテーブル」を各々持つ。それぞれのスイッチのフローテーブルは、SDN の制御部に当たる「OpenFlow コントローラ」によって設定される。OpenFlow コントローラをプログラミングし、それぞれのスイッチのフローテーブルを個別に設定することで、SDN を実現することが可能である。

尚、OpenFlow を実装するためのフレームワークは複数あるが、本研究では trema[4] を使用する。

一般的に OpenFlow コントローラ開発は以下の手順で行

われる

- (1) コントローラコード記述
- (2) コントローラ実行
- (3) コマンド実行 (デバッグ)
- (4) コード修正後、(2) と (3) を繰り返す

通常のプログラミングと手順は似通っているが、デバッグ部が異なる。作成したコントローラコードは、用意したネットワーク環境のコントローラとして実行を行う。そして各ホストやスイッチに対してコマンドを実行し、ホスト間の導通やスイッチの状態を確認する。これがプログラミングにおけるデバッグに相当する。

このことから、OpenFlow コントローラ開発はプログラミングと従来のネットワーク構築の 2 つの側面を持つと言える。

## 3. 本研究の目的

### 3.1 SDN 学習の問題点

SDN が普及するにつれ、教育機関でも SDN に関する講義が行われつつあり、演習形式を採る場合がある。演習形式の講義では各学習者がそれぞれ課題に取り組むが、講師はそれぞれの学習者の進捗状況に合わせた指導が必要となる。よって、プログラミング演習や従来のネットワーク構築演習では、学習状況把握に関する研究が多数実施されている。しかし SDN はそれらと学習方法が異なるため、既存の学習状況把握システムを使用することができない。

SDN によるネットワーク構築演習における学習状況把握に関する先行研究として、我々は学習成果物の収集について研究を行った [5]。先行研究は演習環境の設置場所に依存せず、学習成果物を収集可能とするシステムを開発した。講師は演習に対するテストケースを用意し、学習者は完成したコントローラに対してテストケースを実行し、その結果とコントローラコードをシステムに蓄積する。先行研究のシステムでは演習の成否に関わらず、動作することが保証されているコントローラコードの収集を行なう。しかし、一般的にプログラミング学習者は構文エラーなどを多く起こすと想定されるため、先行研究のシステムではそのようなつまづいている学習者を認識できないと考えられる。

### 3.2 本研究の目的

本研究は SDN によるネットワーク構築学習において、コントローラの実行に至らない、意図した動作を実装することができないといった、つまづいている学習者の発見を容易とすることを目的とする。目的達成の手段として、SDN によるネットワーク構築学習においても有用である学習履歴の収集と、収集した学習履歴の可視化によって視覚的なフィードバックを行なう。プログラミング演習や、従来のネットワーク構築演習の学習状況把握に関する既存の研究

から,SDN によるネットワーク構築演習においてつまづいている学習者を発見可能とする学習履歴を検討する.

## 4. 学習履歴の収集

### 4.1 関連研究

2.2 で示した OpenFlow の開発手順において,それぞれの段階で学習者はなんらかの行動履歴を生ずる.またそれらの行動履歴は,既存の学習状況把握に関する研究で使用されている学習履歴と似通っているため,SDN のネットワーク構築演習において関連研究で使用された学習履歴は有用であると思われる.

プログラミング演習で市村らは [6],つまづいている学習者の早期発見と,多くの学習者が共通に抱える問題の発見を容易にする事を目的として,エラーに着目した分析を行っている.SDN によるネットワーク構築においても学習者はコントローラ実行段階で,構文エラーなどを発生させることが多いと思われる.よって SDN によるネットワーク構築学習で,学習者がコントローラ実行時に起こしたエラーの内容を収集することは有用である.

次に従来のネットワーク構築演習で能見らは [7],学習者の構築したネットワークの設定と,学習者の行動を確認することを目標として,トポロジの編集履歴とコマンド実行履歴を収集している.SDN によるネットワーク構築において,ネットワークの設定はコントローラコードに当たるため,コントローラコードを収集することで,学習者のネットワーク構築状態を伺うことが可能となる.そして,コマンドによってネットワークの動作を確認することは従来のネットワーク構築と同じであるため,学習者が実行したコマンドを収集することは SDN においても有用であると思われる.また能見らは,学習者が操作を行っていない時間に着目し,つまづいている学習者の発見に使用しているため,SDN でも学習者の行動時の時刻は学習履歴として有用である.

3.1 で挙げたように,動作可能なコントローラコードを収集するのみでは,つまづいている学習者を認識できないと思われる.よって本研究では実行時のコントローラコードのみでなく,コントローラコードの編集履歴として保存時のコントローラコードも収集する.

### 4.2 収集する学習履歴

4.1 で述べたように,学習者のつまづきを発見するため,本研究では以下の学習履歴を収集する.

- 保存時のコントローラコード
- 実行時のコントローラコード
- 実行時に起きたエラーの内容
- 実行したコマンド
- 実行したコマンドの結果

保存時のコントローラコードは,学習者の仮想環境に保

存されているコントローラコードを1分に1度取得し,前回取得時と差分があった場合のみ記録を行う.実行時のコントローラコードは,コマンドで OpenFlow コントローラを実行する際に指定したコードを記録する.また,学習者が実行しようとしたコードを収集するため,実際には構文エラーなどにより実行できない可能性がある.構文エラーが起きた場合はそのエラー内容を記録する.そしてデバッグなどを目的として実行したコマンドと,コマンドの結果を記録する.

学習者の行動時の時刻を保存するため,それぞれの学習履歴は記録された時間が付属する.

## 5. 試作及び評価

### 5.1 データ収集システムの実装

本研究では図1に示すように,1つの物理マシン内で複数の仮想マシンを稼働させ,その仮想マシンで学習者が演習を行なう形式を採った.また図1に示すように,それぞれの仮想マシンには Web ベース統合開発環境である Orion を導入している [8].それによって学習者が OpenFlow コントローラを作成し,テストを行なう.Orion は OpenFlow コントローラを記述するためのエディタと,実行とデバッグを行えるコンソールを持ち,以下のコマンドを実行できる.

```
ping 送信元ホスト名と宛先 IP アドレスを指定して ping を打つ
show 指定されたスイッチの情報を出力する
ifconfig 指定されたホストの情報を出力する
trema run 指定したコントローラコードを実行する
trema stop 稼働しているコントローラコードを停止する
trema dump_flows 指定したスイッチのフローテーブルを表示する
```

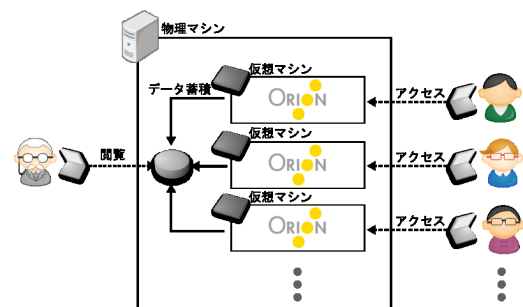


図1 学習履歴収集システムの概要

Fig. 1 Architecture of learning activity data gather system

### 5.2 試験運用結果

実際の講義で先述したシステムの試験運用を行った.対象は長野県工科大学短期大学校情報技術科の講義である,データ通信工学を受講している SDN 初学者である学生 20 名で

ある。講義では、講師が図2に示す仮想ネットワーク環境を各学習者の仮想マシンに作成し、学習者はその仮想ネットワーク上で動作するOpenFlowコントローラを作成する。コントローラ記述後、学習者はコマンドによって各ホストの導通を確認する。

システムを試験運用した結果、表1と図3,4,5に示すような学習者の行動履歴を収集できた。図3は学習状況を学習者ごとに時系列に沿って表示している。また、図4に学習者のコントローラ実行状況、図5にpingの実行状況を示す。

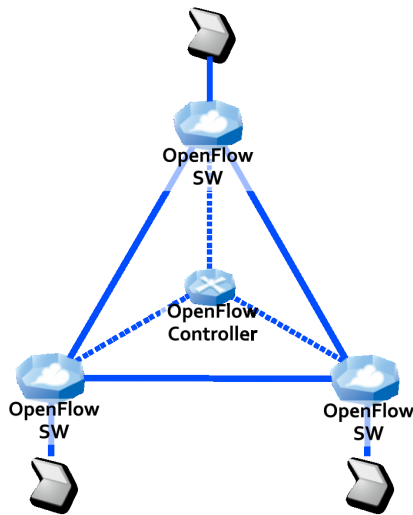


図2 学習用仮想ネットワーク

Fig. 2 Virtual network for practices

保存時のコントローラコード	118 個
実行時のコントローラコード	93 個
- 成功回数	60 回
- 失敗回数	33 回
実行されたコマンド	204 個

表1 全学習者の行動履歴個数

Table 1 The number of learning activity histories of students

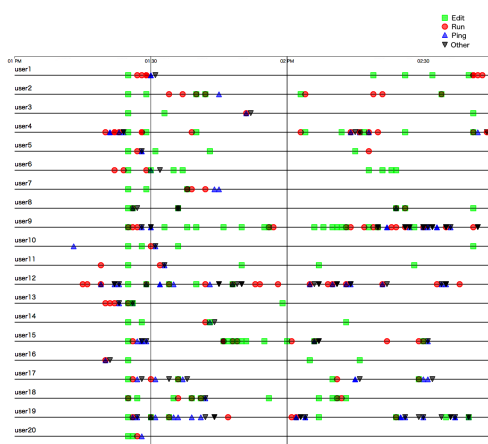


図3 学習状況

Fig. 3 Learning status of students

### 5.3 評価

コントローラ実行回数が2回以下である学習者が半数以上となったため、初学者はコントローラプログラムの作成過程でのつまづきが多いと推測される。コントローラ実行回数が少ない学習者であっても、ファイルの編集履歴を収集したことで学習履歴は多く得られた。またpingの実行状況を見てみると、全ての学習者が失敗しているため、多くの学生が共通してつまづいている部分が存在する可能性があると思われる。

そして学習者の行動時の時刻を記録したことにより、図3を生成できたため、学習者の行動過程を把握することが容易となった。一部を拡大した図が6である。

収集した学習履歴によって、コントローラ実行状況やコマンドの状況がフィードバックできるため、分析により学習者がつまづいている点を判断できると思われる。また、行動が途切れている学生やコマンドの失敗が頻発している学生に対しては、何らかの支援が必要だと見て取れるため、本研究で収集した学習履歴により教授者に有益な情報が提供できると示唆された。

よって、SDNによるネットワーク構築演習において本研究で考案した学習履歴は有用であるといえる。

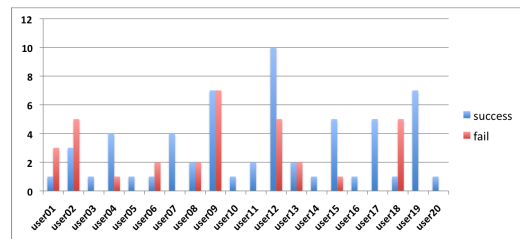


図4 コントローラ実行状況

Fig. 4 Graph about running controller program

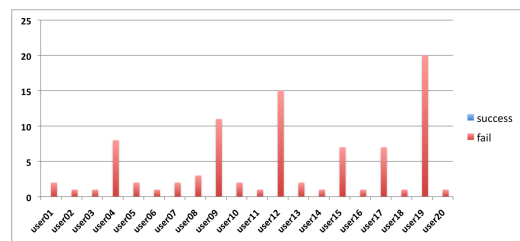


図5 ping 実行状況

Fig. 5 Graph about using ping command

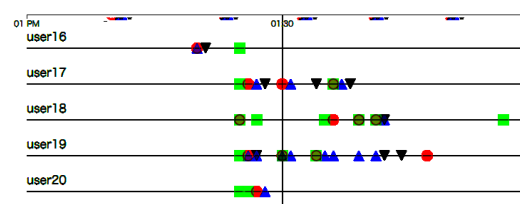


図6 学習状況図一部

Fig. 6 Part of learning status of students

## 6. 今後の展望

本研究では収集する学習履歴の精査のみで留まっているため、コントローラコードや実行時エラー内容、コマンド履歴の分析には至っていない。これらを分析することで、より的確なフィードバックを行えると思われる。

また学習者のデータが収集済みであるならば本システムによって可視化が可能であるが、リアルタイムでの可視化には至っていない。学習履歴収集部と可視化部は実装済みであるため、それらのデータの受け渡しを適切に行うことでリアルタイムで学習状況が把握でき、つまづいている学生に講義中に指導を行うことが可能である。

本研究で対象とした講義は学習者が少ないため、図3のような時系列表示で全員の学習状況を容易に確認可能である。しかし学習者がより多数になると、時系列表示のみでは学習者全体の把握が難しくなる。よって多数の学習者を分類し、より指導が必要な学生を検出するなどのアプローチが必要となる。

謝辞 本研究は科研費 26540180 の助成を受けたものである。

### 参考文献

- [1] : Software Defined Networking, (online), available from <https://www.opennetworking.org> (accessed 2016/04/13).
- [2] : How Google is using OpenFlow to lower its network costs, (online), available from <https://gigaom.com/2012/04/09/how-google-is-using-openflow-to-lower-its-network-costs/> (accessed 2016/04/13).
- [3] : Google's next OpenFlow challenge: taking SDNs to the consumer, (online), available from <https://gigaom.com/2012/04/17/googles-next-openflow-challenge-taking-sdns-to-the-consumer/> (accessed 2016/04/13).
- [4] : Trema Full-Stack OpenFlow Framework in Ruby and C, (online), available from <https://trema.github.io/trema/> (accessed 2016/04/13).
- [5] 牛込翔平, 園生遥, 國宗永佳, 新村正明: SDN によるネットワーク構築実習における分散型実習環境管理システムの開発, 研究報告教育学習支援情報システム (CLE), Vol. 2013, No. 4, pp. 1-5 (2013).
- [6] 哲 市村, 知記梶並, 洋行平野: プログラミング演習授業における学習状況把握支援の試み, 情報処理学会論文誌, Vol. 54, No. 12, pp. 2518-2527 (2013).
- [7] 北澤友基, 井口信和: 演習履歴の管理を可能とする IP ネットワーク構築演習システムの開発, 第 76 回全国大会講演論文集, Vol. 2014, No. 1, pp. 781-783 (2014).
- [8] 百瀬拓也, 横山貴志, 國宗永佳, 新村正明: Web ベース統合開発環境による SDN 実習支援システムの提案, 電子情報通信学会技術研究報告, Vol. 115, No. 492, pp. 7-10 (2016).