

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

牛込 翔平^{1,a)} 園生 遥¹ 國宗 永佳² 新村 正明¹

概要：筆者らは、情報系大学院生を対象に Software Defined Network(以下、SDN) というネットワーク技術の授業を展開し、ネットワーク制御プログラム開発実習を実施している。実習の際、講師側で学習状況を把握するためには、学習者が利用する実習用仮想マシンから情報を参照する必要がある。しかし、実習用仮想マシンを各学習者で用意させる場合、実習用仮想マシンは各学習者 PC 内で動作し、ホストマシン以外の外部ネットワークと通信できない場合がある。そこで本稿は、各学習者ごとに分散している実習用仮想マシンに対し、外部ネットワークと通信させるインターフェースを開発し、開発したインターフェースを利用することで講師側から SDN 実習時の各学習者の学習状況の把握が可能となるシステムを開発した。

1. はじめに

近年、Software Defined Network (以下、SDN) というネットワーク機能や構成をソフトウェアで制御する技術が普及してきている。SDN の普及に伴い、技術者育成も同時に必要となり、大学等の教育機関でも SDN に関する授業が行われている [1][2]。

SDN の授業では、ルーティング機能やロードバランシング機能等の講師側で提示したネットワーク機能を学習者に開発させる実習を行う場合がある。実習では SDN を実現するプロトコルとして代表的な OpenFlow[3] を採用することが多い。OpenFlow では、各ネットワーク機器の制御機能と転送機能を分離したアーキテクチャをとり、複数のネットワークスイッチ (OpenFlow スイッチ) をネットワーク制御プログラム (OpenFlow コントローラ) により一括制御することが可能となる。実習では OpenFlow コントローラのソースコードを学習者に開発させる内容のものが多い。そのような実習を行う際の実習環境として OpenFlow コントローラ開発用フレームワーク、OpenFlow ネットワークエミュレートソフトを導入してある仮想マシンを用いることが多い。

また、プログラム作成実習やネットワーク構築実習において、学習者の作業進度に合わせた指導を行うため、講師側で学習者の作業進度、作業結果等の学習状況を把握可能

にするシステムが開発されており、一定の成果を挙げている [4][5][6][7]。これにより、SDN の実習においても講師側で学習者の学習状況を把握することは有効であると考えられる。

本稿では、SDN の実習時における各学習者の学習情報を取得し、講師側で実習時の学習状況の把握を可能にするシステムについて報告する。

2. 研究目的

プログラミング作成実習やネットワーク構築実習における学習状況の把握には、学習者が実習時に作成するプログラムやネットワークの動作テスト等の結果を集計することから成されている。それらのことから、SDN の実習における学習状況の把握には、各学習者の学習情報として、実習時に作成する OpenFlow コントローラのソースコードやホスト間のパケット疎通結果等を取得し集計することとする。これにより、講師側で学習状況を把握することができると考えられる。

また、プログラム作成実習やネットワーク構築実習において、各学習者ごとに実習用の仮想マシンが必要となる場合、実習環境構築方法として、講師側で仮想マシンを一括で用意する場合と各学習者側でそれぞれ用意する場合の二種類がある。これらのうち、講師側で必要なりソースが少なくすむことや講師側で大量の仮想マシンを管理する必要がない等のメリットから、学習者側で実習用仮想マシンを用意させる場合がある。

学習者側で実習用仮想マシンを用意させる場合、各学習者は実習用仮想マシンをクラウド上や自身の PC 内で動作

¹ 信州大学大学院理工学系研究科
Division of Science and Technology, Shinshu University

² 信州大学工学部
Faculty of Engineering, Shinshu University

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

させるのが一般的である。しかし、そのような環境で仮想マシンを動作させている場合、仮想マシンが動作している仮想システムの設定や、実習用仮想マシン自身の設定により、外部ネットワークに設置してある学習状況把握システムから実習用仮想マシンに対し学習状況の取得が困難な場合がある。

そこで、本研究ではSDN実習において講師側で学習者の学習状況を把握することを目的とし、まず外部ネットワークから実習用仮想マシンに対し、実習時の情報を取得可能な通信インターフェースを開発した。そして、開発したインターフェースを利用しSDN実習における各学習者の学習状況を把握するシステムを開発した。

3. 関連研究

ネットワーク実習における学習状況把握を可能にするシステムが開発されている。井口らは、User Mode Linuxを利用したネットワーク構築演習システムを開発している[8][9]。井口らのシステムを利用することで、学習者はスイッチやルータ等の設定を行うネットワーク実習を実施でき、各機器の設定内容の自動評価も可能となっている。更に、各学習者の評価結果を講師側で取得することもできる。しかし、井口らのシステムは、使用する仮想マシンを講師側サーバで一括で管理するため、本研究で想定する学習者側で仮想マシンを用意させる方式と異なる。さらに、実習内容においてもルータ等のネットワーク機器設定を対象としているため、本研究で対象としているSDN実習が実施できない。

また、プログラミング演習における学習状況の把握を可能にするシステムの研究も行われている。石川らは、プログラミング演習時における学習状況把握システムを提案、開発している[10]。石川らのシステムは、学習者にwebブラウザからjavaプログラムをコンパイル、実行させることで、それらの情報を取得し、講師側でクラス全体における学習進度等の学習状況を把握でき、プログラムのコンパイルエラー分類や作業が遅れている学習者の特定等も可能となっている。しかし、これらのシステムにおいて、プログラムを動作させるのは講師側システムであり、SDNの実習に必要な学習者ごとの仮想マシン環境を必要としない。そのため、本研究で想定している実習用仮想マシンから各学習者の学習情報が取得できない。

4. 学習状況取得用通信インターフェース

4.1 想定する実習環境

2章で述べた通り各実習用仮想マシンをクラウド上や学習者PC内で動作させる分散方式を採用した場合においても、学習状況把握の為に実習用仮想マシンから学習者の実習時の情報を取得する必要がある。

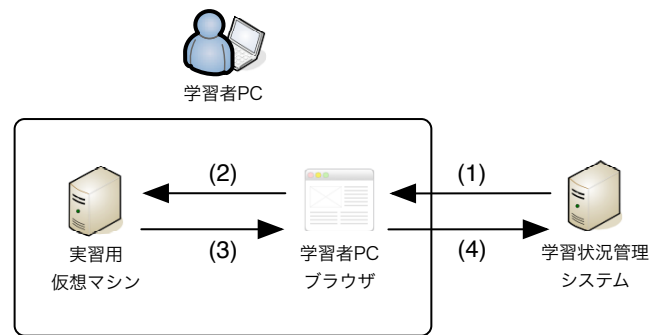


図1 学習状況取得インターフェース通信手順

Fig. 1 Learning status acquisition interface communication procedure

多くの学習状況管理システムでは学習者にブラウザからシステムにアクセスさせ、ブラウザ上で実習に必要な情報の参照やプログラムの動作テストを行わせている。そして、その際に学習者から入力された内容等をシステム側で収集、分析し講師に提供している。

また、実習の際に学習者は自身のPCから実習用仮想マシンにSSH等でログインし作業を行うこととなる。そのため、学習者PCから実習用仮想マシンへの通信は保証されていると考えられる。

そこで本研究では、実習の際に学習者にブラウザから学習状況管理システムへアクセスさせることと、学習者PCから実習用仮想マシンへの通信が可能なことを想定し、学習状況管理システムと実習用仮想マシン間の通信インターフェース開発を行った。

開発したインターフェースにより実習用仮想マシンと学習状況管理システム間の通信路が確立されることから、学習情報を取得するプログラム（学習者のソースコードをテストする検証用プログラムやそのテストケース等）を用意することで講師側で学習情報を取得できる。

4.2 ブラウザを介した socket.io による通信の実現

本インターフェースでは、実習用仮想マシンと学習状況管理システム間の通信路を確立する方法として、学習者PCのブラウザを介した socket.io[12] を利用する方法を考案した。socket.io とはリアルタイム web 通信を実現する javascript ライブラリセットであり、socket.io を利用することで幅広いブラウザでリアルタイム双方向通信が可能となる。

本インターフェースの利用には、学習状況管理システム側と実習用仮想マシン側の両方で socket.io サーバを動作させる。更に、実習時に学習者が利用する web ページに socket.io による通信を実現する javascript を埋め込んでおく。このような状態にすることで、各サーバからブラウザに対し情報の受信、送信が可能となる。

4.3 通信インターフェースにおける通信手順

例として、学習者 PC 内で動作している実習用仮想マシンと学習状況管理システム間における、本インターフェースを利用した具体的な通信手順について図 1 を用いて説明する。

- (1) 学習者 PC のブラウザが学習状況管理システムから情報取得用プログラムを受信する。
- (2) 学習者 PC のブラウザが情報取得用プログラムを実習用仮想マシンへ送信する。
- (3) 実習用仮想マシンは受信した情報取得用プログラムを実行し、標準出力で得た結果をブラウザに送信する。
- (4) 学習者 PC のブラウザは受信した結果を学習状況管理システムに送信する。

以上の様に、通信させる情報を一度学習者 PC のブラウザを経由させることで、外部ネットワークに設置してある学習状況管理システムから実習用仮想マシンに対しての通信が可能となる。また、本インターフェースは学習状況管理システムから学習状況取得プログラムを登録し、そのテスト結果を標準出力として受け取る。そのため、講師側で用意する学習状況取得プログラムを実習内容に合わせて変更することで、様々な情報を得ることができ、様々な実習に応用できる。

5. SDN 実習時における学習状況管理システム

4 章で述べた学習状況取得用通信インターフェースを利用することにより、SDN 実習時における各学習者の学習情報を取得し、講師側で学習状況を把握可能にするシステムを開発した。

5.1 信州大学大学院における SDN 授業

本システムは信州大学大学院（以下、本学）で開講されている「情報基礎特論第 2」で実施される実習を想定して開発することとした。この授業では、最新技術の学習の一環として、2012 年度より SDN に関する授業 [11] が実施され、授業では SDN の概念や利用法を学習し、1 コマ 90 分の授業が全 15 回実施される。

本学の SDN 授業においても OpenFlow を用いた SDN の実習を実施している。実習内容は各スイッチのリンクアップ・ダウンによる段階的なネットワークトポロジ変化に対応し、宛先ホストにパケットをルーティングさせる OpenFlow コントローラを開発させている。実習の評価基準は、トポロジ変化を起こした際にホスト間のパケット疎通確認を総当たりで行い、その際に疎通確認された回数を点数としている。

2013 年度に開講された本学の授業では、実習用仮想マシンを講師側で一台の仮想プラットフォーム上で動作させ、学習者に提供する方式を採用した。しかし、実習では学習

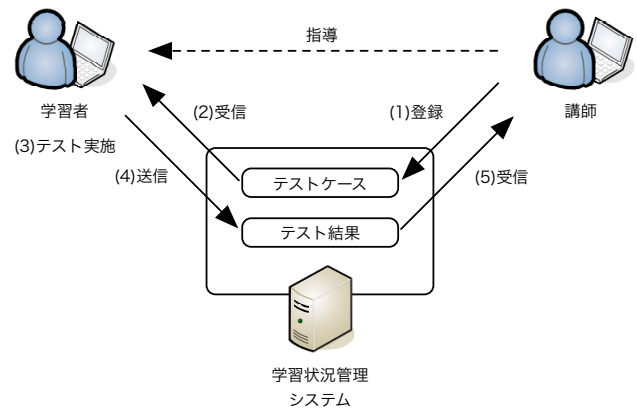


図 2 SDN 実習用学習状況管理システム構成図

Fig. 2 SDN practice for learning management system configuration diagram

者がトラフィック制御を誤り、仮想ネットワーク内でブロードキャストストームを発生させ、実習用仮想マシンに大きな負荷をかけてしまっていた。この際、一台の仮想マシンに高負荷がかかると他仮想マシン群のパフォーマンスにも影響を与えてしまい、他学習者の仮想マシンがうまく動作せず、実習困難となる事象が発生した。そのため、来年度以降の実習では実習用仮想マシンを各学習者に用意させることとしている。

実習の際、学習者は自身の実習用仮想マシンにログインし OpenFlow コントローラのソースコードを記述する。ソースコード記述後に正しく OpenFlow ネットワークを制御可能か動作テストを行う。動作テストでは、段階的にネットワークトポロジの変化を起こし、その際の各ホスト間のパケット疎通確認を行う。本システムでは試験的に、学習情報として動作テストを行う際の情報を取得することとした。

5.2 学習情報取得機能

本学の 2013 年度で実施した実習内容を想定し、学習者が行う OpenFlow コントローラ動作テスト結果とその際のソースコード等を講師側で取得する機能を実装した。開発した学習状況管理システムで SDN の実習時に取得する情報を以下に示す。

- テストを行った学習者の学籍番号
- テストを行った OpenFlow コントローラのソースコード
- 動作テストを行ったイベントリスト
- 動作テスト結果
- テスト日時

これらの状況を講師側で取得することで、クラス全体についての課題に対する作業進度、学習者のプログラム履歴を把握することができる。

本システムの構成を図 2 に示す。本システムは講師側

で学習状況を管理する学習状況管理システムと各学習者に分散させた実習用仮想マシンから構成される。システムは web アプリケーションとして開発し、システムと実習用仮想マシンでは socket.io サーバが動作している。システムは学習者の実習時の情報を取得し、学習状況格納用 DB に格納する。それらの情報から学習者の学習状況を講師側に提供する。

5.3 学習情報取得例

本システムで学習情報を取得する手順を図 2 を用いて説明する。以下の手順における学習状況管理システムと実習用仮想マシン間の通信は 4 章で述べた学習状況取得用通信インターフェースで実現している。

5.3.1 テストケース登録

本システムの利用にあたり、まず講師側で OpenFlow コントローラ検証用プログラム、テストケースを事前に本システムに登録しておく。テストケースでは、どのスイッチのどのポートをどの順序でアップ、ダウンさせるかが設定される。また、今回作成した検証用プログラムは、登録されたテストケースに従い段階的に仮想ネットワークのトポロジを変化させ、その都度仮想ネットワーク上の全ホスト間のパケット疎通確認を行い、試行回数と成功回数を出力するものとなっている。

5.3.2 テストケース受信

学習者はソースコードを記述後、ブラウザ上から OpenFlow コントローラの動作テストを行う。動作テストを行う際、講師側で事前に登録されたテストケースとテストを行うソースコードのファイル名を指定する。テストケースが選択されると、選択されたテストケース情報と検証用プログラムが実習用仮想マシンへ送信される。

また本システムでは学習情報取得機能に加え、学習者の実習を支援する機能として、自身の仮想マシンで動作している OpenFlow ネットワークの情報の参照 (図 3) を行うことができる。この際、システム側では現在システムにアクセスしている学習者の学籍番号を取得し、現時点で何人の学習者が実習中なのかを把握している。

5.3.3 実習用仮想マシンでのテスト実施

実習用仮想マシンでは、ブラウザから検証用プログラムとテストケースを受信すると学習者が作成したソースコードの動作テストを行う。この際、動作テスト結果は標準出力により得られる。

5.3.4 システムへテスト結果送信

動作テスト完了後、実習用仮想マシンは動作テスト結果、テストを行ったソースコードをブラウザへ送信する。ブラウザはテスト結果を実習用仮想マシンから受信すると、それらの情報をシステムに送信する。システムは受信した情報を学習状況格納データベースに格納する。

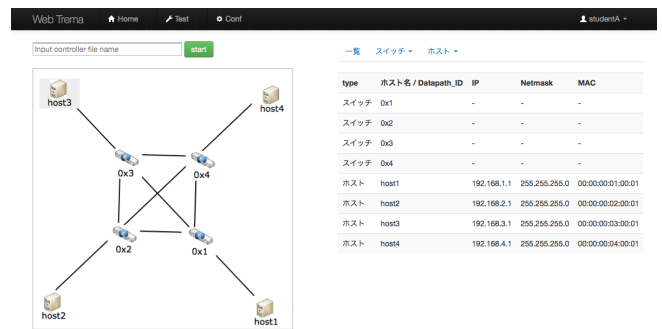


図 3 実習支援用 Web インターフェース
Fig. 3 Training support web interface

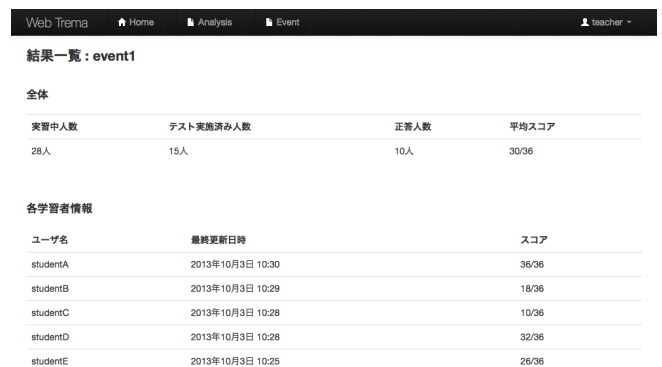


図 4 講師用 Web インターフェース
Fig. 4 Instructor web interface

5.3.5 講師側への学習状況提供

講師側は格納された学習情報を web ページ上で確認することができ (図 4)、学習者の動作テスト履歴、ソースコード履歴の詳細も確認できる。更に、これらの収集した情報からクラス全体での課題に対する作業進捗や学習者のプログラム履歴を把握することができ、学習者の作業進捗に合わせた実習運営が可能になると考えられる。

6. おわりに

本稿では、分散状態にある各学習者の実習用仮想マシンに対し、外部ネットワークに設置する学習状況管理システム側から学習情報の取得が可能な学習状況取得インターフェースを開発した。そして、開発した学習状況取得インターフェースを利用することで SDN の実習における学習状況の取得が可能となるシステムを開発した。

本システムを利用することにより、実習用仮想マシンが分散している状況でも、SDN の実習時に講師側で学習者の学習状況を管理しながら実習を実施できる。また、開発した学習状況取得インターフェースは検証用プログラムやテストケース等を変更することで、SDN 実習以外のプログラミング演習等にも利用できる汎用性も持ち合わせている。

しかしながら、開発した学習状況管理システムで取得可

能な学習状況は SDN の実習内容に十分に則しているとは言えない。今後はより SDN 実習に則した学習状況の把握を可能にする機能を開発するとともに、本システムを実際に SDN の授業に導入し、本システムの有効性を評価する予定である。

参考文献

- [1] 長谷川剛：情報ネットワーク学演習 II(online), 入手先 https://koan.osaka-u.ac.jp/syllabus_ex/ (2013.11.12).
- [2] Eric Keller：ECEN 5013-Software Defined Networking(online), 入手先 http://ecee.colorado.edu/ekeller/classes/fall2012_sdn/ (2013.11.12).
- [3] Open Networking Foundation：Open-Flow Switch Specification(online), 入手先 <https://www.opennetworking.org/sdn-resources/onf-specifications/openflow> (2013.11.12).
- [4] 倉澤邦美, 鈴木恵介, 飯島正也, 横山節雄, 宮寺庸造：プログラミング演習における一斉指導のための学習状況把握支援システムの開発, 電子情報通信学会技術研究報告, ET, 教育工学, vol.104, no.703, pp.19-24 (2005).
- [5] 宮地恵佑, 高橋直久：構造誤り検出機能を有するアセンブラプログラミング演習支援システムの実現と評価, 電子情報通信学会論文誌, vol.J91-D, no.2, pp.280-292 (2008).
- [6] 井垣宏, 齊藤俊, 井上亮文, 中村亮太, 楠本真二：プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案, 情報処理学会論文誌, vol.54, no.1, pp.330-339 (2013).
- [7] 中松智昭, 立岩佑一郎, 片山喜章, 高橋直久：仮想マシンネットワークを用いた初学者向け IP ネットワーク構築演習の自動評価システムの実現, 電子情報通信学会技術研究報告, ET, 教育工学, vol.110, no.453, pp.169-174 (2011).
- [8] 北澤友基, 井口信和：IP ネットワーク構築演習支援システムの自動採点機能の開発, 電子情報通信学会技術研究報告, ET, 教育工学, vol.113, no.229, pp.69-74 (2013).
- [9] 北澤友基, 越智洋司, 溝淵昭二, 井口信和：クラウド環境を利用した協調演習を可能とする IP ネットワーク構築演習支援システムの検討, 電子情報通信学会技術研究報告, ET, 教育工学, vol.112, no.66, pp.19-24 (2012).
- [10] 加藤利康, 石川孝：プログラミング演習支援システムにおける学習状況把握機能の提案, 情報処理学会研究報告, コンピュータと教育 (CE), 2013-CE-120(2), pp.1-8 (2013).
- [11] 新村正明：情報情報基礎特論第 2(online), 入手先 <http://campus-2.shinshu-u.ac.jp/syllabus/syllabus.dll/top> (2013.11.12).
- [12] Socket.Io(online), 入手先 <http://socket.io/> (2013.11.12).