

# 汎用学習管理システムと連動する IP 技術自己学習機能の提案

河野歩磨 鈴木大雅 永濱裕太 藤本衡

ネットワーク技術を効果的に学習するため、講義資料や演習問題などに加えて仮想ネットワークの構築演習を実現するためのシステムを提案する。講義資料等の提供は汎用のオンライン学習管理システム Moodle を用い、仮想ネットワークの構築と動作確認はネットワークシミュレータ ns-3 を元に改良を行う。この学習システムが多数の学習者の同時アクセスに耐えうるものかを検証するため、同時リクエスト数およびシミュレータの並行実行数を変化させて性能評価実験を行う。並行実行数がコア数とほぼ比例したことがわかった。

## 1 はじめに

インターネットの普及に伴い、情報技術者にとって TCP/IP を中心としたネットワーク技術の理解は必須となっている。ネットワーク技術の知識を定着させるためには座学や筆記演習だけではなく、実際にネットワーク環境を構築・設定し動作を確認させる演習を組み合わせることが効果的であると考えられる。しかし、演習に必要なネットワーク機器を学習者自身が準備することは困難であり、また専門教育機関においても演習環境を学習者ひとりずつに提供することは（コストや設置場所・時間などの問題から）難しいのが現状である。

一方でコンピューティング資源の増大により、シミュレータソフトウェアや仮想ネットワーク技術による演習環境は、比較的 low コストで利用可能となっている。これらの演習環境を学習者に分かりやすい形で提供することにより、学習者それぞれが必要な規模のネットワーク構築演習を実施し、知識の定着を促進することが期待される。

こうした自己学習用のシステムは実験的なものが

Online self-learning system in conjunction with the general-purpose learning management system.

Ayuma Kono, Daiga Suzuki, Yuta Nagahama, Kou Fujimoto, 東京電機大学理工学部, School of Science and Engineering, Tokyo Denki University.

提案されているが、その多くは演習環境のみの実装にとどまっている。そこで本研究では、講義資料や演習問題などと一体となった学習を可能にするため、汎用のオンライン学習管理システムに組み込む形で演習環境を実装する。

また、既存の演習環境は仮想マシンおよび仮想ネットワークを利用したシステムを前提としており、多数の学習者が学習管理システムに同時アクセスした場合、スケーラビリティの面で課題が残る。本研究では、仮想マシンではなくオープンソースのシミュレータソフトウェアを利用することで、スケーラビリティを確保する。

## 2 関連研究

2000 年代以降、IP ネットワークの構築演習支援システムについての研究が複数行われている。

北澤ら [1] は、User Mode Linux を用いてサーバ上に複数の仮想マシンを生成し、ネットワークシミュレーションを行うシステムを開発した。このシステムは Web アプリケーションとして動作し、学習者はアプリケーションを操作することによって仮想ネットワークを構築する。仮想マシンを多数生成しメモリ消費量を確認する実験においては、おおよそ 1 台あたり 27MB 程度を消費し、台数に対して線形に増加することが示された。このことから、多くの学習者が同

時にアクセスし複雑なネットワークを構築する場合に、スケーラビリティの問題が生じるおそれがある。さらに、利用にはある程度の知識が必要との記述がみられ、ネットワークの学習途上にある学習者には必ずしも適切とは言えない。

舛賀ら [2] は [1] に加えて操作履歴の共有機能および疑似学習者の機能を追加した。

また孫ら [3] は、ネットワーク構成を図示することが学習者の助けになると論じ、実際に機器の設定情報などを収集し自動的に図示するシステムを提案した。こうした議論からも、ネットワークを視覚的に表現したハンズオン学習環境を構築することは、ネットワーク技術の習得に重要な意味を持つことがわかる。

### 3 システム概要

本研究で提案するシステムは、クライアント（汎用 Web ブラウザ）、フロントエンド（汎用学習管理システム+独自プラグイン）、バックエンド（ネットワークシミュレータソフトウェア）からなる。本システムの基本設計を図 1 に示す。

クライアントはネットワークシミュレータを操作するためのユーザインタフェース (UI) を学習者に提供する。UI 上で構築されたネットワークを表現するデータは JSON 形式 [6] で記述され、フロントエンドに送信される。UI 部分の設計については、3.1 節で詳しく述べる。

本研究では汎用学習システムとして Moodle 3.0 [4] を、またネットワークシミュレータソフトウェアとして ns-3 [5] を使用する。Moodle は PHP で記述された Web アプリケーションであり、プラグインと呼ばれる PHP スクリプトを追加することで機能拡張を実現する。ns-3 は C++ で記述されたネットワークシミュレータで、シミュレーション環境および動作を表現するシナリオファイルもまた C++ で記述される。

本研究で実装する独自プラグインは、おもに 2 つの機能を持つ。第 1 に、クライアントに対して UI 用の HTML5+JavaScript コードを提供する。第 2 に、クライアントから送信された JSON 形式を ns-3 が解釈可能な C++ 形式のシナリオファイルに変換する。そして、このシナリオファイルを ns-3 に処理させ、UI

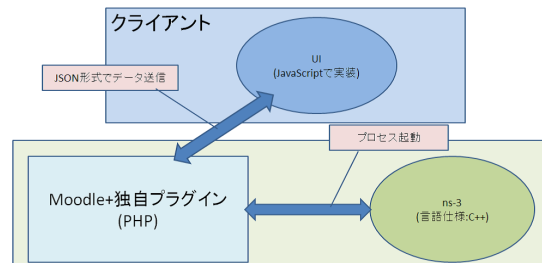


図 1 基本設計

で生成された仮想ネットワークにおける通信の可否などを確認する。ns-3 の出力結果はこの独自プラグインを介して、クライアントに送信される。これにより、ネットワークシミュレーションが実現できる。Moodle および独自プラグインについては、3.2 節で述べる。

#### 3.1 UI の構成

本節では提案するシステムの UI の概要を説明する。UI のデザインは HTML と CSS で記述され、JavaScript のライブラリである jQuery で UI の動的な機能を表現する。UI のデザインを図 2 に示す。

図 2 の画面は大きく分けて 5 つに分かれており、それぞれ、ナビゲーション部・機器部・ネットワーク部・詳細設定部・コンソール部を表示する。

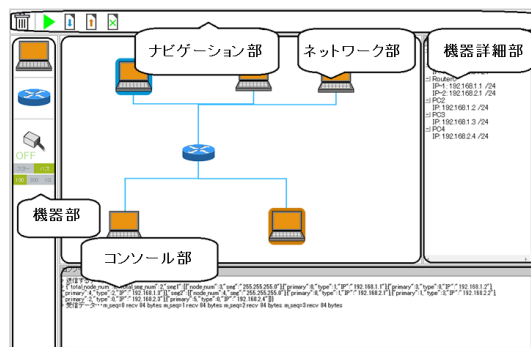


図 2 ネットワークシミュレータ UI

機器部では、上から「Host」「Router」「LINK」のアイコンを表示している。「Host」と「Router」は、ネットワーク部にドラッグ&ドロップすることで配置することができる。「LINK」はクリックすることでONとOFFを切り替えることができ、ONの状態ではネットワーク部に配置された機器の間にリンクを生成することができる。その他に「LINK」の設定を変更するボタンがあり、リンクの属性を変えることができる。

詳細設定部では、各機器の名称と設定内容が表示される。設定内容は各機器に対するIPアドレスとサブネットワークマスクを表示する。本システムではIPアドレスとサブネットワークマスクはCIDR表記で表示する。詳細設定を変更するには、設置した機器を右クリックすることで図3のコンテキストメニューを表示する。そして、IPアドレスとサブネットワークマスクを設定する。

ナビゲーション部では、左から「全削除」「実行」「受信パケット付与」「送信パケット付与」「パケット削除」のアイコンを表示している。「受信パケット付与」はネットワーク部に配置されたホストをクリックすることで、そのホストを宛先として設定する。宛先ホストには青色の枠が付加される。同様に「送信パケット付与」は送信元を設定し、送信元ホストには赤色の枠が付加される。それらの設定を解除したい場合は「パケット削除」を選択して当該ホストをクリックする。実行アイコンをクリックすることで、ネットワーク部に構築されたネットワークを表すJSON形式のデータが生成され、サーバ側の独自プラグインに送信される。

コンソール部では、UI上で起こったエラーやns-3から独自プラグインを介して受信したデータを表示する。

### 3.2 Moodleと独自プラグイン

本システムではeラーニングプラットフォームであるMoodleを採用した。Moodle[4]は学習管理システム(Learning Management System)であり、GNU GPL (General Public License)の下で配布されているオープンソースソフトウェア(OSS)である。

標準機能の活動モジュールを用いることで、構築演



図3 機器のコンテキストメニュー

習以外の座学や筆記演習を実現することが可能であり、本研究で提案する構築演習システムと組み合わせることで一貫したネットワーク学習課程を実施できると期待される。

MoodleはPHPで記述したプラグインによって機能拡張が可能であることから、本研究で提案するシステムをPHPで記述し、Moodleのプラグインの一種である活動モジュールとして実装することとした。活動モジュールとして実装された構築演習システムのスクリーンショットを図4に示す。これによって、Moodle上でネットワークシミュレータを用いて仮想的にネットワークを構築できるようにした。

独自プラグイン内部では、クライアントからJSON形式で記述された仮想ネットワークの情報を受け取る。これをns-3が解釈可能なC++形式のシナリオファイルに変換する。このファイルをns-3に受け渡

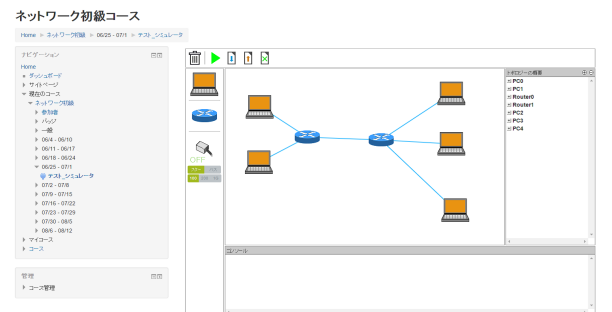


図4 Moodleのプラグイン

し、ns-3 を子プロセスとして実行する。このとき、ns-3 は元々複数のプロセスとして並行実行されることを想定していないが、独自プラグインはクライアントからのリクエストに応じて個別のプロセスとして実行される。このため複数の ns-3 プロセスが実行可能のように改変を加えた。具体的には ns-3 ではシナリオファイルを置くディレクトリが固定であるため、プロセスごとに異なるディレクトリを参照するように変更した。

また、ns-3 はシナリオファイルに対してコンパイルを実行するため、あまりに多くの ns-3 プロセスが同時に実行されると、著しい性能劣化を引き起こす恐れがある。そこでセマフォを用いて ns-3 の並行実行数に制限を加える機能を付与した。

### 3.3 ns-3

ns-3 [5] とは、分散イベント駆動型のネットワークシミュレータであり、主に研究や教育向けに開発された OSS である。

本研究では、構築された仮想ネットワークのシミュレーションを ns-3 が担当する。

ns-3 におけるネットワークシミュレーションでは、シナリオファイルを記述してモジュールファイルとリンクしてネットワークシミュレーションを行う。

モジュールファイルではプロトコルやネットワーク機器などのネットワーク機能を構成する要素が定義されている。各モジュールはヘルパークラスと呼ばれるクラスを用いて設定できる。

シナリオファイルは、これらのモジュールを組み合わせてネットワーク構成を記述する部分である。ns-3 のシナリオファイルは、ノード、チャンネル、ネットワークデバイス、アプリケーションの 4 つに分けて構成されている。ノードは ns-3 における基本的なコンピューティングデバイスを指し、このノードにネットワークデバイス機能、プロトコルスタックやアプリケーションを乗せることによって、ネットワーク上のルータやスイッチ、コンピュータなどの振る舞いを分けることが可能になる。

本研究ではネットワーク層の学習も目的の 1 つとしているが、ns-3 のデフォルトの IP 割り当て用のヘ

ルパークラスである Ipv4AddressHelper では自動で同セグメント内の IP 割り当てを行ってしまうため、そのままでは学習者に IP の設定を問うような問題が作成できない。そこで、個々の機器に IP を静的に割り当てられるヘルパークラスを新たに作成した。

## 4 性能評価実験

3 節で提案したシステムが、多数の学習者が同時接続した場合においても十分に機能するかどうかを検証するため、以下の実験を行った。提案システムを実装するサーバのスペックは、CPU が Intel Core i7-2600(クロック 3.40GHz, 4core), 16GByte メモリを搭載している。OS は Ubuntu 16.04 LTS (64bit) を、Web サーバは Apache2.4.18 を用いた。PHP スクリプトはバージョン 7.0 を前提に記述された。

### 4.1 実験手順

シミュレーションの対象となる仮想ネットワークの構成を図 5 に示す。図 5 は、パソコン 6 台とルータ 1 台で構成されたネットワークである。各ノードには IP アドレスとサブネットマスクを設定して、P0 を宛先、P4 を送信元に設定する。この仮想ネットワークは通信に成功する前提で構成されている。

多数の同時接続を行うため、汎用のブラウザの代わ

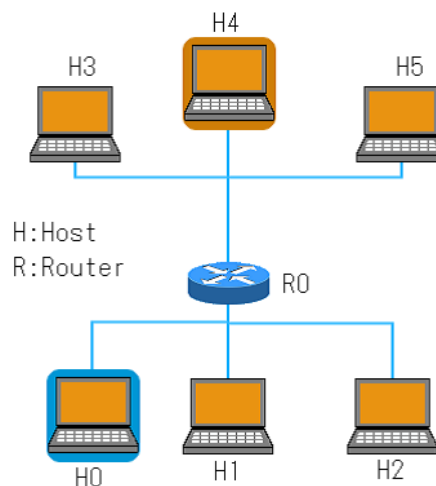


図 5 実験で用いたトポロジー図

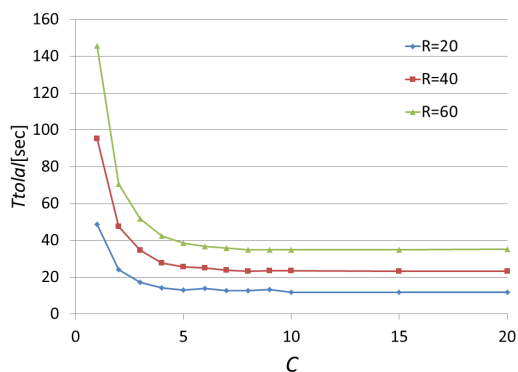


図6 経過時間

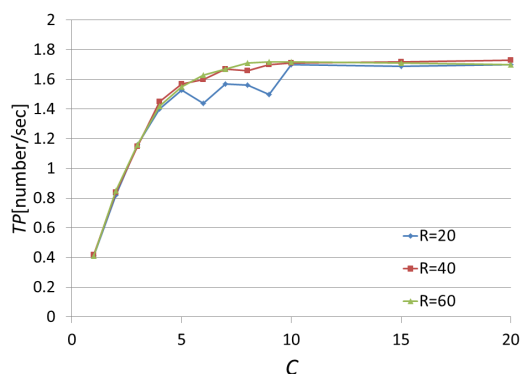


図7 一秒間に処理したリクエスト数の平均値

りに Web ベンチマークツールである ApacheBench を用いた。ApacheBench では、複数のリクエストを 1 つもしくは複数のコネクションでサーバに送信することが可能であるが、本節の実験では各学習者が一斉にサーバにアクセスすることを想定し、リクエスト数とコネクション数を同じ値に設定する。

#### 4.2 実験結果

リクエスト数  $R$  は 20・40・60 の 3 種類を想定する。また、ns-3 の同時実行数  $C$  は 1~10 まで 1 ずつ増加させ、さらに  $C = 15, 20$  も実験した。以上のパラメータについて、リクエストを開始してから最後の応答が返るまでの時間  $T_{total}$ 、および 1 秒あたりに処理したリクエスト数  $TP$  を計測した。

$C$  の増加に対して  $T_{total}$  がどのように変化するかを図 6 に示す。 $T_{total}$  は実質的に最も長く応答を待つ学習者の待ち時間を表しており、リクエスト数が増えると場合によっては 1 分近く待つ必要があることがわかる。このことから、待機状態であることを UI 上で表すことが必要と考えられる。

$C$  の増加に対して  $TP$  がどのように変化するかを図 7 に示す。 $C = 1, 2, 3$  においては、ほぼ  $C$  の増加に対して線形に性能が向上していることがわかる。これは CPU のコア数が 4 であるため、並行実行の効果が高いためと考えられる。 $C = 4$  のときには  $C = 1$  の 3.5 倍程度の性能であり、ns-3 以外のプロセスがあることから線形には性能向上していない。さらに、 $C \geq 5$  ではほとんど性能向上が見られず、ns-3 の CPU 利用率が極めて高いことを示唆している。以上のことから、本システムの性能向上にはコア数の増加が大きく寄与すると考えられる。

#### 5 まとめと今後の課題

本研究では、Web ブラウザ上で動作する IP 技術自己学習機能を提案した。また、この機能を汎用の学習管理システムである Moodle のプラグインとして実装した。このシステムに対して同時リクエスト数を変化させて性能実験を行い、CPU コア数を増加させることで性能向上を見込めることがわかった。

現状では ns-3 の機能上の制約により、ルーティングテーブルを静的に設定するといった機能が実現できていない。この点は、3.3 節で説明した IP アドレスの静的割り振りと同様に、ヘルパーを新たに定義することで対応可能と考えられる。また現在はネットワーク層の部分に関する構築演習しか実装していない。今後、データリンク層やアプリケーション層の機能追加などを行うことが期待される。

#### 参考文献

- [1] 北澤友基, 井口信和, “クラウド環境を利用した IP ネットワーク構築演習支援システムの開発”, 情報処理学会第 74 回全国大会講演論文集, pp.891-892, Mar. 2012.
- [2] 舩賀計彦, 早川諒, 井口 信和, “IP ネットワーク構築演習支援システムにおける擬似学習者機能の開発”, 2015 年度情報処理学会関西支部 支部大会.
- [3] 孫一, 清光英成, 柏木治美, 康敏, 大月一弘, “ネット

- ワーク技術学習のためのネットワーク図自動作成機構の試作”, pp.79–83, Jan.2014.
- [4] “Moodle — Open-source learning platform”, <https://moodle.org/>, 2016-Aug-18.
- [5] “ns-3”, <https://www.nsnam.org/>, 2016-Aug-18.
- [6] D. Crockford, “The application/json Media Type for JavaScript Object Notation (JSON)”, RFC 4627, 2006.