

2017 情報システムデザイン学系卒業論文

2017 年度 情報システムデザイン学系卒業論文

論文番号 fm2017-01

応答性を向上した IP ネットワーク個人学習システム

魚本 悠太 (14RD019)、大須賀 旭 (14RD024)、中村 優 (14RD128)

指導：藤本 衡 准教授

提出日：平成 30 年 2 月 28 日

概 要

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要不可欠である。ネットワークの構築演習として実機を使用した演習があるが、学習者一人ひとりに実機を提供することは現実的ではない。学習者がネットワーク技術を効果的に習得するため、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独自の判定システムを作成し使用する。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コネクション数を変化させて性能評価実験を行う。経過時間がリクエスト数と比例していることがわかった。また、作成したシステムの問題点や使用感を調査するため、アンケート評価を実施した。作成したシステムがネットワーク学習の支援になっているという意見が多かったが、問題点としてシステムの使用方法がわかりにくいことが挙げられた。

執筆分担について、魚本が1節「はじめに」、3.3節「Moodle と独自プラグイン」、4節「性能評価実験」、4.1節「実験手順」、4.2節「実験結果」、6節「まとめと課題」について担当した。大須賀が2節「関連研究」、3.2節「独自の判定システム」、5節「アンケート評価」、5.1節「アンケート手順」、5.2節「アンケート結果」について担当した。中村が3節「システム概要」、3.1節「UI の構成」について担当した。

目 次

1	はじめに	4
2	関連研究	6
3	システム概要	8
3.1	UI の構成	10
3.2	判定システム	19
3.3	Moodle と独自プラグイン	21
4	性能評価実験	23
4.1	実験手順	24
4.2	実験結果	25
5	アンケート評価	32
5.1	アンケート手順	32
5.2	アンケート結果	33
6	まとめと課題	35

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要不可欠である。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に加えて、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現実的ではない。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）困難である。

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を用いた演習環境を構築することは比較的容易となっている。これらの演習環境をユーザビリティを考慮した形で学習者に提供することにより、学習者それぞれがネットワーク構築演習を実施し、知識の定着を促すことができると考える。

こうした自己学習用のシステムは実験的なものが提案されている。しかし、今まで提案されてきた多くの自己学習システム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、システム全体を汎用のオンライン学習管理システムに組み込むことで、講義資料や演習問題など一体となった学習システムを実現する。また、既存の研究で応答性を妨げる要因として、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の場合では、多数の学習者が学習管理システムに同時アクセスした時に処理しきれないことが挙げられる。さらに、オープンソースのシミュレータソフトウェアを利用したシステムの場合では、汎用的なネットワークのシミュレーショ

ンを行うための実装が取り入れられているため、TCP/IP を学習する上で必要のない機能が多く存在していることが挙げられる。そこで本研究では、ルーティングに関係する処理のみを実装した独自の判定プログラムを利用することで、スケーラビリティを確保する。

また、実装した学習システムが学習者にとって分かりやすく、使いやすいかは知識の定着への貢献に大きく関係するものである。そこで、本システムがネットワーク技術の理解の助けとして機能するかを調査するために、実際にネットワーク技術を学習している人物を対象に本システムを使用してもらい、アンケート評価を実施する。

以下、2 節では、本研究に関連する類似研究について説明する。3 節では、本研究で提案するルーティングに関する学習システムについて説明する。4 節では、複数の同時リクエストがあった場合に本システムがどのぐらいの性能を発揮するのか実験した結果を説明する。5 節では、システムを実際にユーザーに触れてもらった結果、判明した問題点を説明する。

執筆分担について、魚本が 1 節「はじめに」、3.3 節「Moodle と独自プラグイン」、4 節「性能評価実験」、4.1 節「実験手順」、4.2 節「実験結果」、6 節「まとめと課題」について担当した。大須賀が 2 節「関連研究」、3.2 節「独自の判定システム」、5 節「アンケート評価」、5.1 節「アンケート手順」、5.2 節「アンケート結果」について担当した。中村が 3 節「システム概要」、3.1 節「UI の構成」について担当した。

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。

孫ら [1] は、理論学習とハンズオン学習などの実践学習をうまく結びつけることが重要であることを指摘し、その上で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た。そこで、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。

北澤ら [2] は、仮想 Linux である User Mode Linux を利用してサーバ上に複数の仮想マシンを生成し、それらを仮想ネットワーク機器として扱うことでネットワークシミュレート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。このアプリケーションの性能評価実験として、仮想マシンを多数生成しメモリ使用量を確認する実験を行った。その結果、1 台あたりおおよそ 32MB 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て仮想のネットワークを構築する場合、スケーラビリティの問題が生じるおそれがあると考えられる。

舩賀ら [3] は北澤らの研究に加えて操作履歴の共有、疑似学習者、問題演習・自動採点の機能を追加した。自動採点機能で学習者の習熟度レベルを判定し、それぞれに応じた演習モードを切り替えることで学習者のモチベーションの維持を提案した。

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。

また河野ら [4] は、上記の各先行研究では演習環境のみの実装にとどまっている点を指摘し、講義資料や演習問題など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した。更に、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いたため、大人数の学習に向いていないという指摘もした。そこで、ネットワークシミュレートを行う部分を、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ことを提案した。しかし、リクエスト毎にシナリオファイルをコンパイルしてシミュレーションを行うため、同時実行数が 60 を越えると 1 分以上待たなくてはならないという問題が発生した。そのため、大人数に提供するシステムとしては適していないと考えられる。

3 システム概要

本研究のシステムは、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（判定システム）で構成されている。本システムの構成図を図 1 に示す。

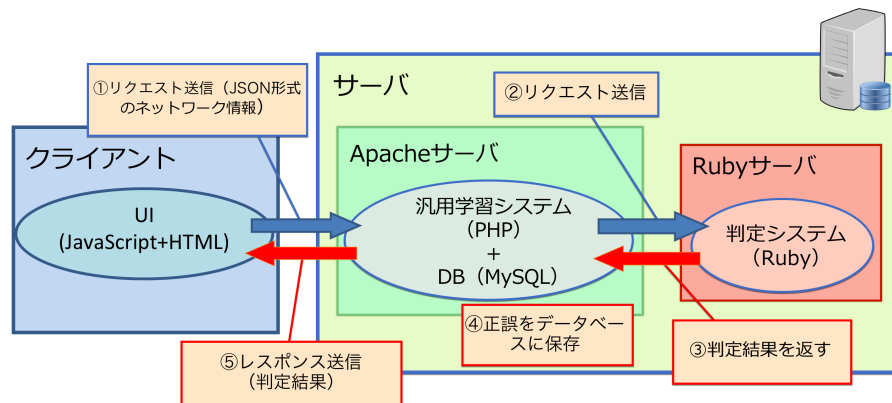


図 1: システム構成

クライアントは判定システムを操作するためのユーザーインターフェース（UI）を学習者に表示する。独自プラグインは、HTML と JavaScript で記述されている。また、UI 上で構築されたネットワークの情報を JSON 形式 [7] のデータに変換し、フロントエンドに送信する機能を持つ。

本研究では、河野らと同様に汎用学習管理システムとして、Moodle を用いているが、判定システムは Ruby で書かれた独自のスクリプトを使用する。Moodle は PHP で記述された Web アプリケーションであり、PHP スクリプトをプラグインとして追加する。本研究では UI を独自プラグインとして追加している。判定システムは Ruby で記述されており、フロントエンドから JSON 形式で記述されたネットワーク

の情報を送信することで、ネットワークの正誤を出力する。出力された情報は、独自プラグインを介して、ブラウザ上に表示される。

3.1 節では、河野らの研究からどのように UI が変化したかを詳しく述べる。また、3.2 節では判定システムについて、3.3 節では汎用学習型システムと独自プラグインについて詳しく述べる。

3.1 UIの構成

本節ではシステムの UI の概要を説明する。UI のデザインは、河野らと同様に、HTML5 と CSS3 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。河野らとの大きな違いは、学習者がネットワークを自身で構成し、機器情報を設定する自由描画モードと、学習者に既に作成されたネットワーク構成を問題として与え、機器情報を設定させる問題演習モードの2つのモードがあることである。最初に、自由描画モードの UI のデザインを図 2 に示す。

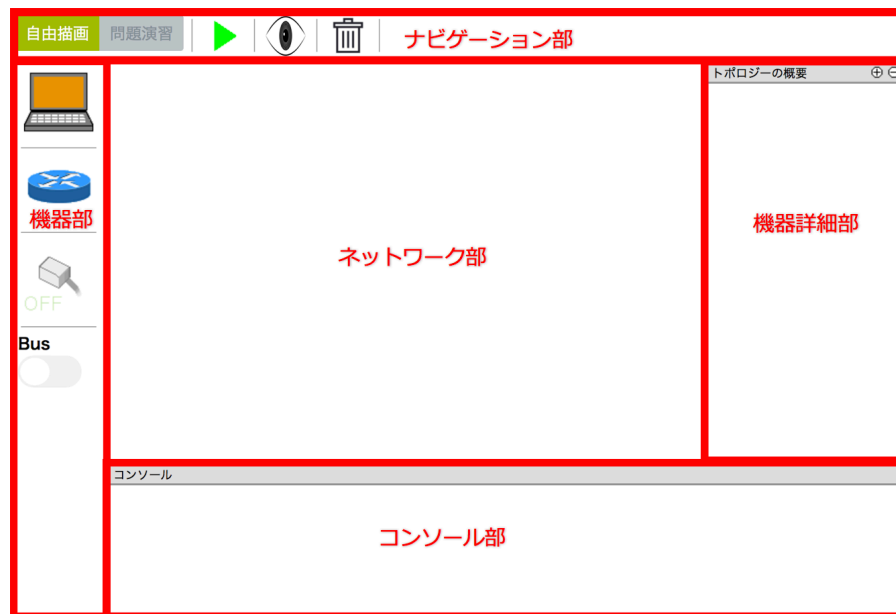


図 2: ネットワークシミュレータ UI 自由描画モード

河野らと同様に、図 2 の画面は大きく 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・コンソール部を表している。

次に、問題演習モードの UI デザインを図 3 に示す。問題演習モードでは、ナビゲーション部に新しいアイコン、UI 下部に問題表示部が追加されている。



図 3: ネットワークシミュレータ UI 問題演習モード

ネットワーク部では、機器部からドラッグ&ドロップされた「PC」と「Router」を配置することができる。本研究では新たに、ノードの周囲に名前やインターフェースの番号を表示している。

詳細設定部では、河野らと同様に、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マスクに加え、新たにルーティングテーブルの情報が表示されている。ネットワーク層の学習において、ルーティン

グテーブルの学習は必須であると考えられるが、河野らが使用していた ns-3 は、ソフトウェアが自動的にルーティングテーブルを作成してしまうため、学習者が自分で設定することができなかった。本研究では ns-3 を使用しないため、学習者にルーティングテーブルを設定させることが可能になっている。本システムでは IP アドレスとサブネットマスクは CIDR 表記で入力する。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 4 のコンテキストメニューを開き、IP アドレスとサブネットマスクを入力する。また、ルーティングテーブルには、宛先ルートとネクストホップアドレス、インターフェースを入力する。+ ボタンで新たにルーティングテーブルを一行追加することができ、- ボタンでその行を削除することができる。

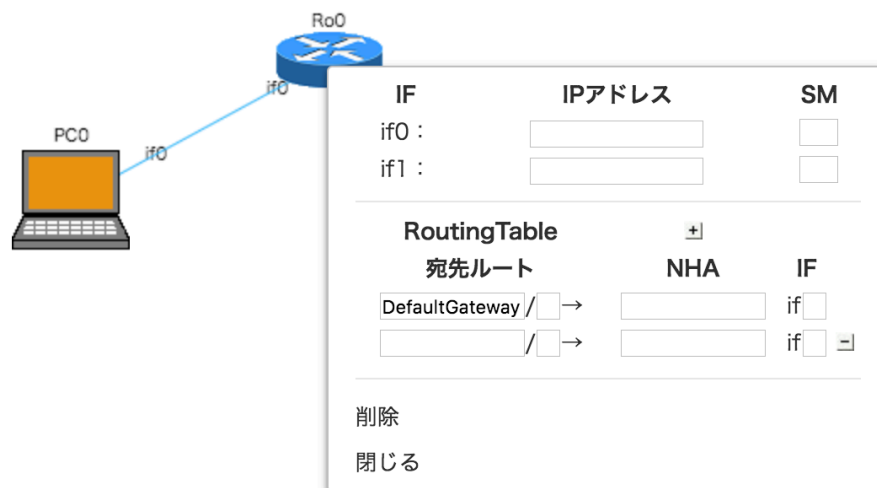


図 4: ネットワークシミュレータ UI 機器のコンテキストメニュー

自由描画モードのとき、ナビゲーション部は左から、「モード選択」「実行」「確認」「全削除」のアイコンである。「モード選択」は自由描画モードと問題演習モードを切り替えるボタンである。「実行」はクリックすることで、ネットワーク部に構成されたネットワークの情報とコンテキストメニューに入力されたデータを元にJSON形式のデータを生成し、サーバに送信する。その後、構成したネットワークの正誤を表示する。「確認」はクリックすることで、図5のように各ノードの情報を表示する。「全削除」はクリックすることで、ネットワーク部に配置された機器やその情報、機器詳細部とコンソール部のテキストを削除する。

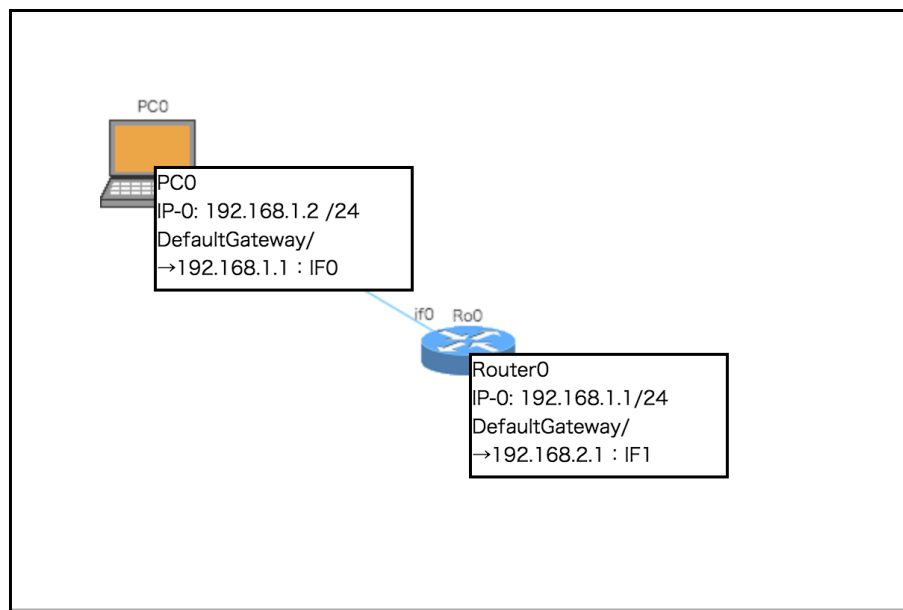


図 5: ネットワークシミュレータ UI 確認

問題演習モードのとき、ナビゲーション部には「全削除」の右から「練習問題」「セーブ」「ロード」「終了」のアイコンが追加されている。「練習問題」はクリックすることで、図6がUI上に表示される。表示されたボタンをクリックすることで、問題表示部に問題文、ネットワーク部にネットワーク構成が表示される。ボタンはそれぞれ色分けされており、灰色は無解答の問題、緑色は過去に正解したことがある問題、赤色は過去に不正解だった問題を示している。「セーブ」はクリックすることで、学習者が入力したネットワークの構成情報をJSON形式のデータに変換し、サーバのデータベースに保存する。「ロード」はクリックすることで、「セーブ」によって保存されたネットワーク構成を呼び出すことができる。「セーブ」「ロード」によって、解答途中の問題を保存して、別の問題に移ることができる。「終了」はクリックすることで、問題演習モードを終了し、図7のように成績を表示することができる。

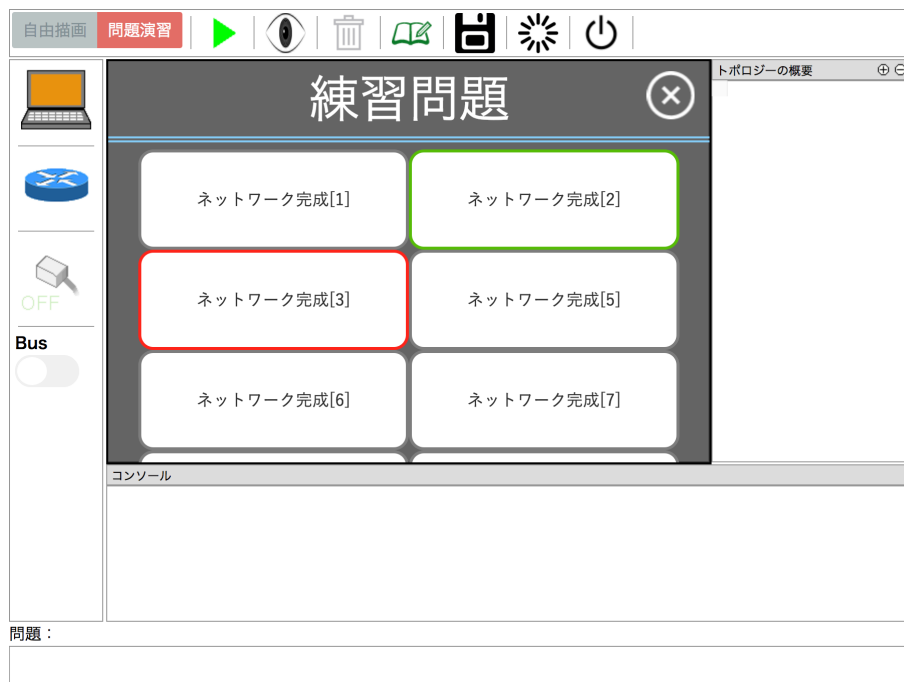


図 6: ネットワークシミュレータ UI 練習問題

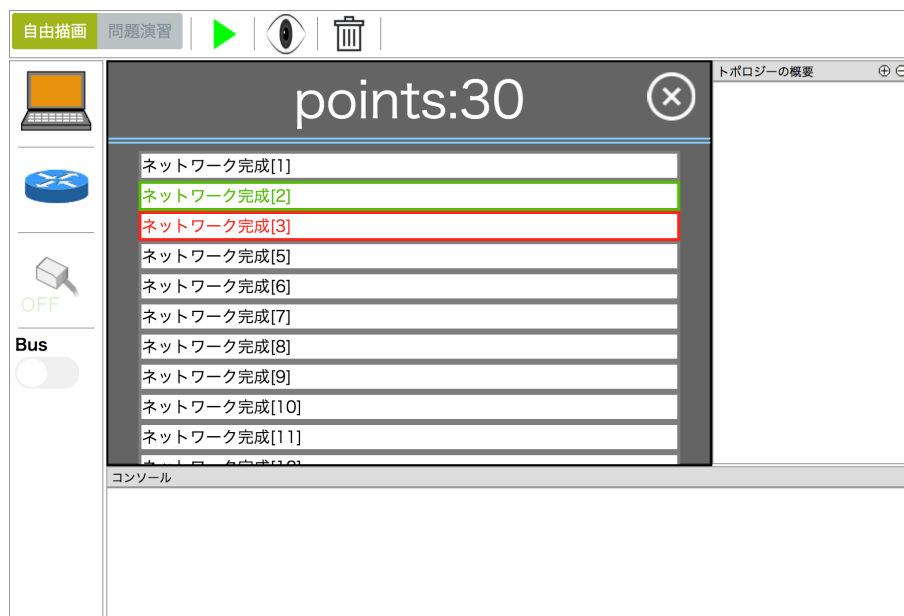


図 7: ネットワークシミュレータ UI 終了

機器部では、上から「PC」「Router」「LINK」「Bus」のアイコンを表示している。「Bus」はスイッチを ON にすることで、図 8 のようなバス型ネットワークを構成する際の横線を描画することができる。また、河野らは、「LINK」の設定を変更するボタンがあったが、本研究では使用しないため削除した。

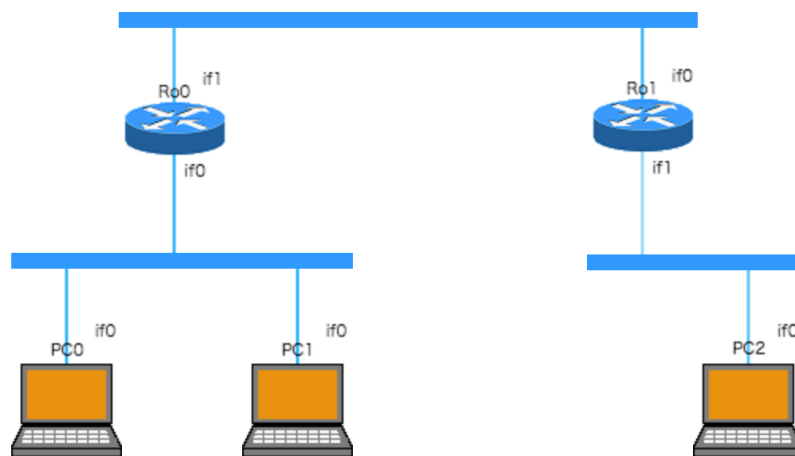


図 8: ネットワークシミュレータ UI バス型ネットワーク

コンソール部では、UI 上で起こったイベントやエラー、サーバ側から受信した情報を表示する。本研究では新たに、コンソールに出力された結果をクリックすることで図 9、図 10 のようなアニメーションが表示される機能を実装した。この機能によって、学習者はどのノードで間違えたかを確認することができる。

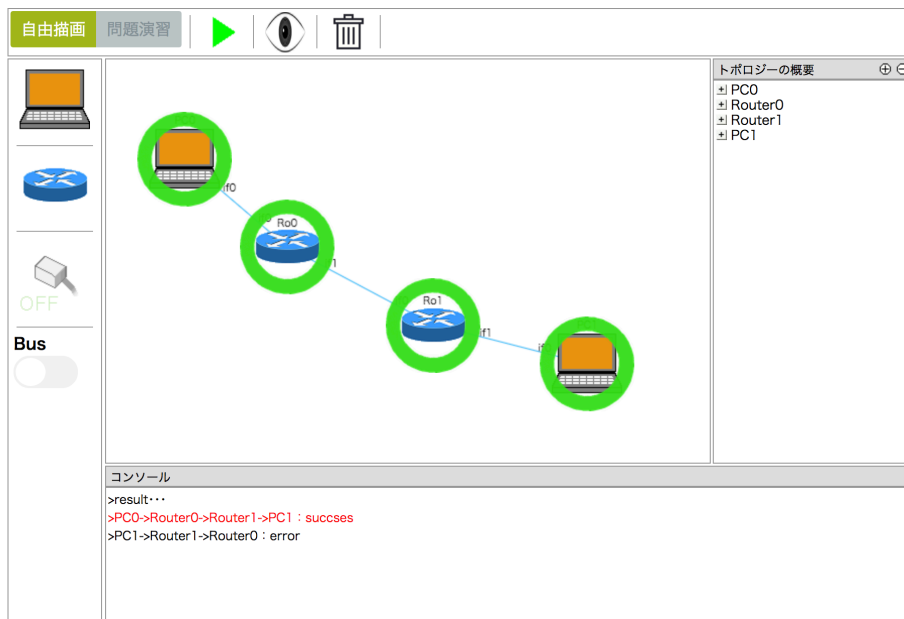


図 9: ネットワークシミュレータ UI 正解アニメーション

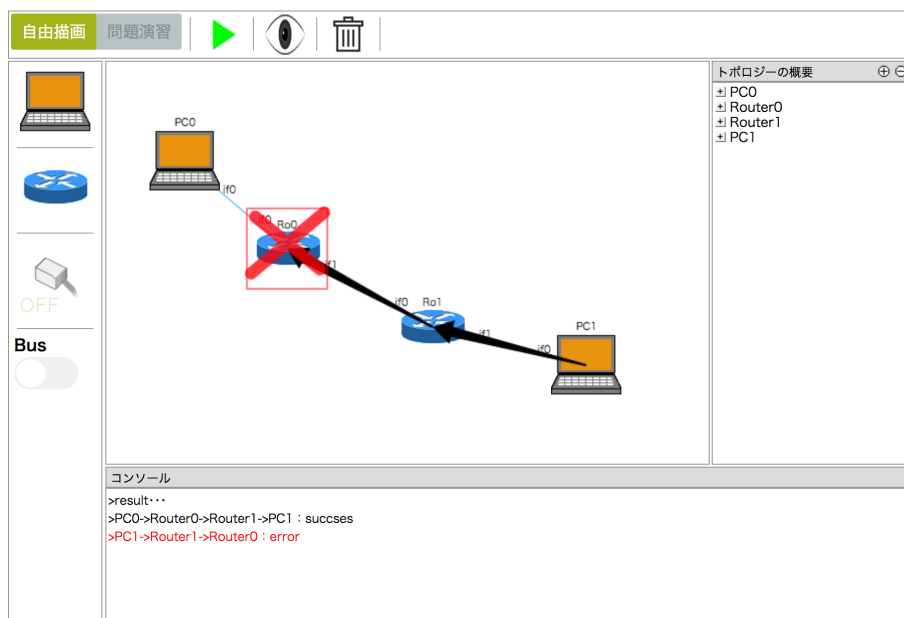


図 10: ネットワークシミュレータ UI 不正解アニメーション

本研究では、河野らの研究の課題になっていた描画のバグが改善されている。描画バグの要因として、画面をスクロールした状態での、ブラウザのウィンドウのリサイズが挙げられる。この時、可変するウィンドウに合わせて再描画されるが、初期位置（読み込み時の位置）から画面をスクロールした分だけ描画がずれていた。そこでウィンドウのリサイズ時に画面を初期位置に戻すことで描画のズレをなくしている。

3.2 判定システム

本研究では、既存研究で見られる仮想環境あるいはソフトウェア的なシミュレータによる解答正誤判定に代わり、設定情報のみから妥当性を判定するソフトウェアを作成する。言語は Ruby で記述している。

このシステムは簡易的なサーバーを作成し、常時 Listen する。シミュレートが行われる度、UI 側との情報の交換をプロセス間通信にて行う。この時の情報は受講者が設定したネットワークの構成図であり、この情報は JSON 形式で送信されている。そして、構成されたネットワークが正しく接続されているかどうかを Ruby で記述された関数で判定を行い、レスポンスとして結果を JSON 形式で送信する。

UI から受け取る JSON にはすべてのノードの情報と使用されているバスの情報が記述されている。構造として、大きく `pc_num`、`nodeInfo`、`busInfo` に分かれている。`pc_num` には描画されているパソコンの数が記述されている。`nodeInfo` にはノードの情報が記されており、`name`(ノードの名前)、`ip`(IP アドレス)、`sm`(サブネットマスク)、`routingtable`(ルーティングテーブル)、`link`(接続されているノードの名前) で構成されている。更にルーティングテーブルには `ip`(受信先の IP アドレス)、`sm`(サブネットマスク)、`nha`(ネクストホップアドレス)、`if`(インターフェース番号) が記述されている。`busInfo` にはバスの情報が記述されている。バスの情報には、一つのバスに接続されているノードが `nodes` として記述されていて、使用したバスの個数分の情報が記述されている。そして、結果として送信する JSON には各ノード同士が辿った経路と正誤を記述している。正解していた場合は辿った経路と正解であるということ、不正解の場合は誤った部分までと不正解であるという結果を記述して

いる。

判定プログラムは、以下の動作をすべての PC 同士の組み合わせにおいて行う。

1. pc0,...,pcX から任意の 2 つを送信元・受信先に指定
2. 送信元のルーティングテーブルを参照し、次にどのノードへ向かうかを検索
3. 次のノードと現在の送信元が正しく設定されていれば、次のノードが新たに送信元となり、手順 2 に戻る
4. もし、次のノードが送り主であれば正解を返す
5. 次のノードへたどり着かなかった場合、止まった場所と不正解を返す

[4] では、ネットワークのシミュレーションを行う際に ns-3 という離散イベント駆動型のネットワークシミュレータを用いている。ns-3 はシナリオファイルとモジュールファイルを元にコンパイルし、実行ファイルを元にネットワークのシミュレーションを行う。この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。また、ネットワーク機器やプロトコルなどのパラメータを設定することもできるシナリオファイルでは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り、ノード、チャネル、ネットワークデバイス、アプリケーションの 4 つに別れて構成されているこのノードにネットワークデバイス機能などを乗せることによってネットワーク上のルータやコンピュータなどの振る舞いを分けることが可能となっている。

本研究では初学者向けのネットワーク学習のオンラインの学習機能の実装を目標としていて、ns-3 の情報量やコンパイルにかかる時間を考慮すると、大人数向けの

システムとしては実行に時間がかかってしまうという問題が発生した。この問題を解決するため、ネットワーク構成のみを判定するプログラムを作成することで、大幅の実行時間の短縮を行うことに成功した。

3.3 Moodle と独自プラグイン

本システムはeラーニングプラットフォームである Moodle を使用して実装を行った。Moodle は学習管理システム (Learning Management System) である。これは、GNU GPL(General Public License) の下で配布されており、オープンソースソフトウェア (OSS) となっている。標準機能である活動モジュールを用いることで、構築演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 の機能拡張は PHP で記述されたプラグインによって可能であるため、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り、これを判定システムに HTTP 通信で送信する。

自由描画モードのときは、判定された結果を受け取り、クライアントに結果を渡して処理は終了となる。問題演習モードのときは、判定された結果を受け取り、クライアントに結果を渡すと同時に、正誤を確認してデータベースに保存する。このとき、データベースに保存されるデータはユーザー ID、問題 ID、アイテム ID、正誤となる。アイテム ID は Moodle 内でそれぞれの活動モジュールに割り振られる ID で

あり、活動モジュールを判別する際に用いられる。また、このときに保存されるデータは問題 1 問ごとの形となっており、最終的な成績は終了ボタンが押されたときに問題 1 問ごとの正誤を確認し、正解の場合は問題に割り振られた点数を合計していくことで成績を算出する。これを Moodle が管理する成績データベースの finalgrade レコードに保存することで Moodle の成績管理システムと連携した。

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 Xeon E5-2620 v2(クロック 2.10GHz, 12core)、32GByte メモリを搭載している。このサーバに VMware ESXi 6.5.0 standard をインストールした。その上で仮想マシンを作成し、提案システムを実装するサーバとした。この仮想マシンのスペックは OS が Ubuntu Server 16.04.2(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。また、CPU とメモリはそれぞれ 2core、2GByte と 8core、16GByte の 2 パターンで実験を行った。さらに、性能評価を行うクライアントとして仮想マシンを作成した。クライアントのスペックはサーバとして作成した仮想マシンと同様である。

4.1 実験手順

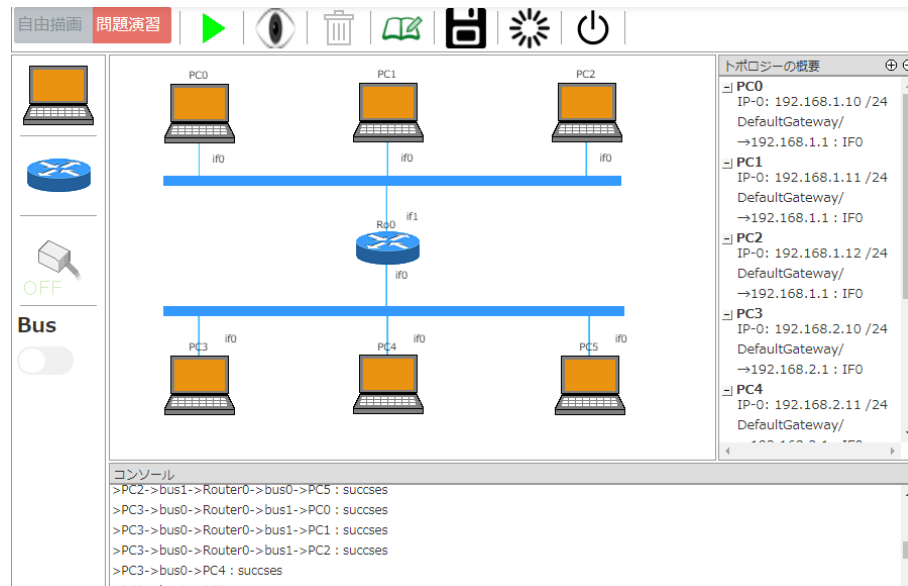


図 11: 実験で用いたトポロジー図

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 11 に示す。図 11 は、[4] と同様にパソコン 6 台とルータ 1 台で構成されたネットワークである。全てのノード間で通信が行えるように IP アドレスとサブネットマスク、ルーティングテーブルを設定する。多数の同時接続を行うため、汎用のブラウザの代わりに Web ベンチマークツールである ApacheBench を用いた。

4.2 実験結果

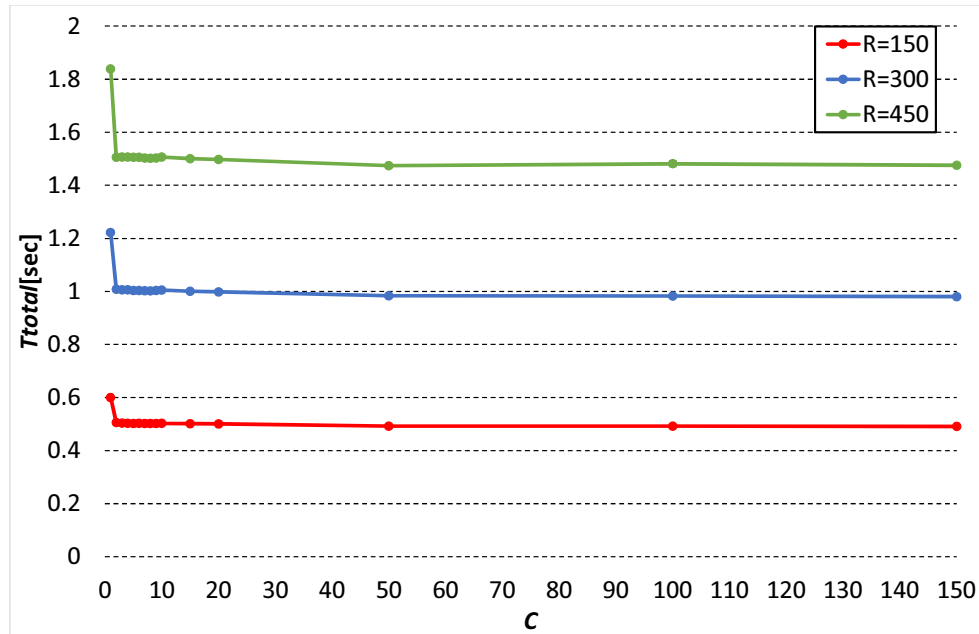


図 12: サーバ、クライアント高スペック時の経過時間

リクエスト数 R は 150・300・450 の 3 種類を想定する。また、ApacheBench のコネクション数 C は 1～10 まで 1 ずつ増加させ、さらに $C=15, 20, 50, 100, 150$ 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 100 回計測し、平均を取った。また、このときのサーバ、クライアントのスペックは CPU が 8core、メモリが 16GByte で計測を行った。

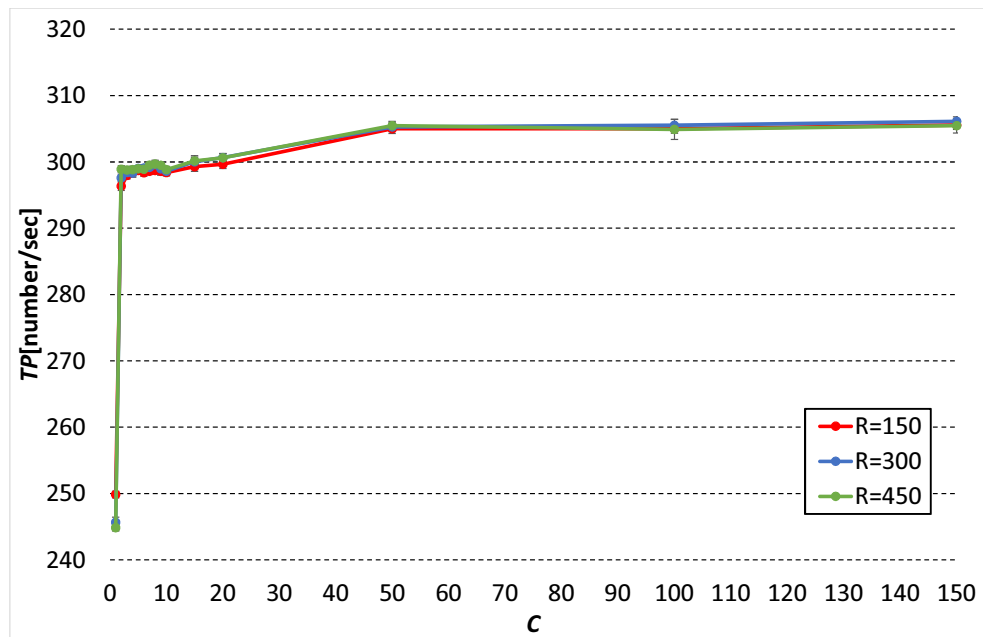


図 13: サーバ、クライアント高スペック時のスループット

C の増加に対して T_{total} がどのように変化するかを図 12 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、150 人の学習者が 3 回のリクエストをほぼ同時に行っても 2 秒以下で返答できていることがわかる。また、経過時間がリクエスト数と比例してると考えられる。 C の増加に対して TP がどのように変化するかを図 13 に示す。 $C=1$ において極端に応答性が落ちていることがわかる。

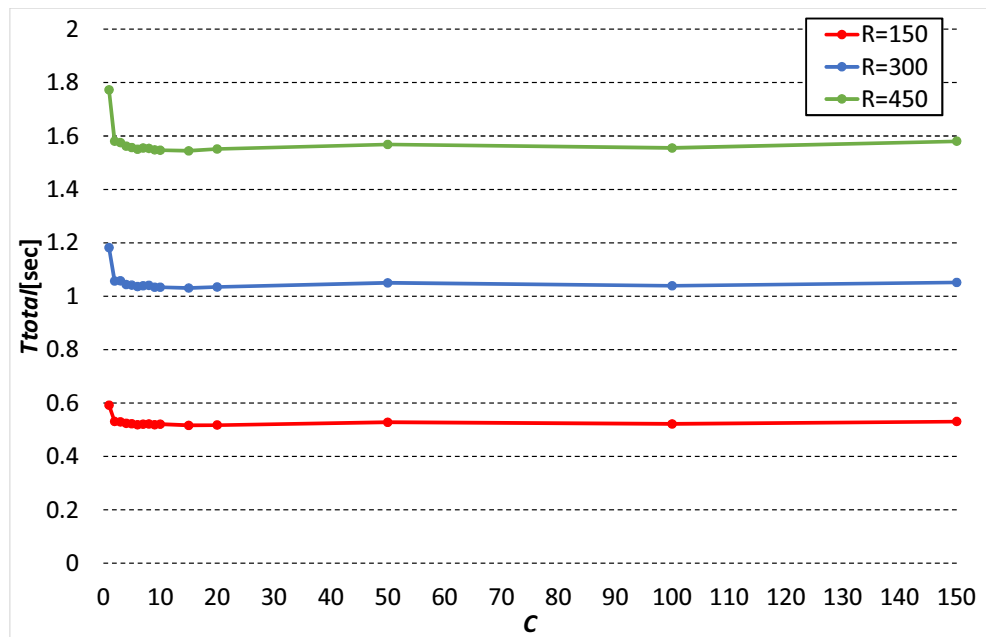


図 14: サーバ、クライアント低スペック時の経過時間

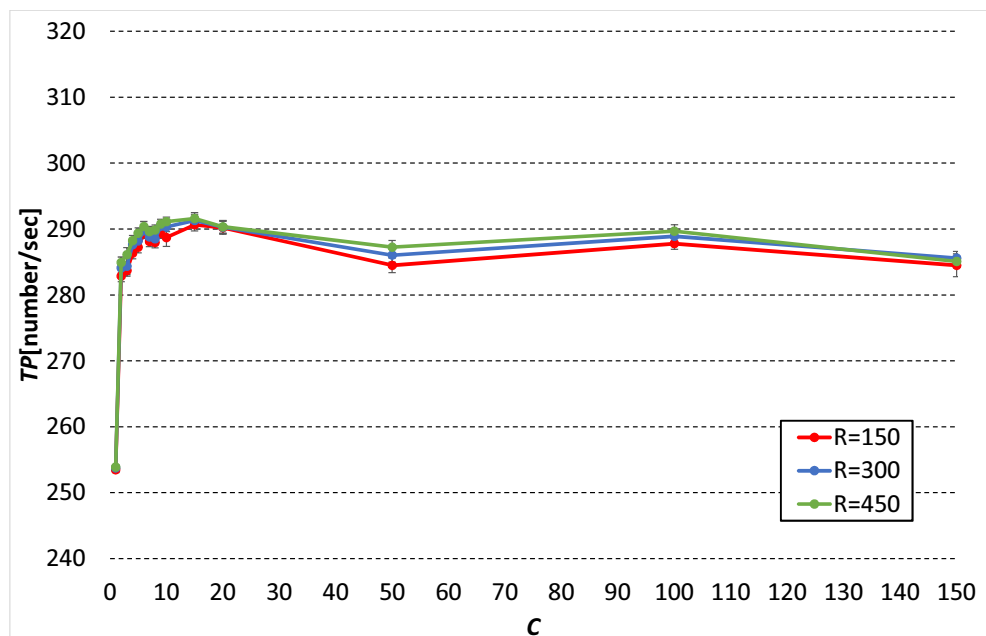


図 15: サーバ、クライアント低スペック時のスループット

そこでサーバ、クライアントのスペックと判定システムのパフォーマンスの関係を調査するために、サーバ、クライアントのスペックを CPU が 2core、メモリが 2GByte に変更して実験を行った。その結果が図 14、図 15 である。この結果より、サーバ、クライアントの性能が一定以上であれば本システムの応答性にほぼ影響がないことがわかる。そのため、 $C=1$ のとき極端に経過時間が高くなってしまう原因は判定システムが 1 つのインスタンスで処理を行っているためであると考えられる。

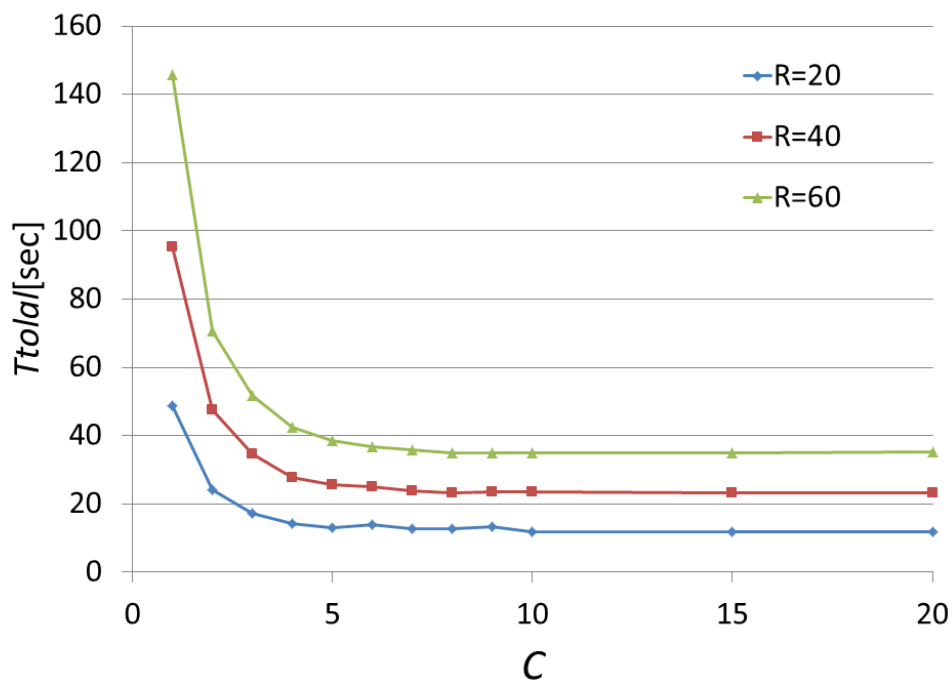
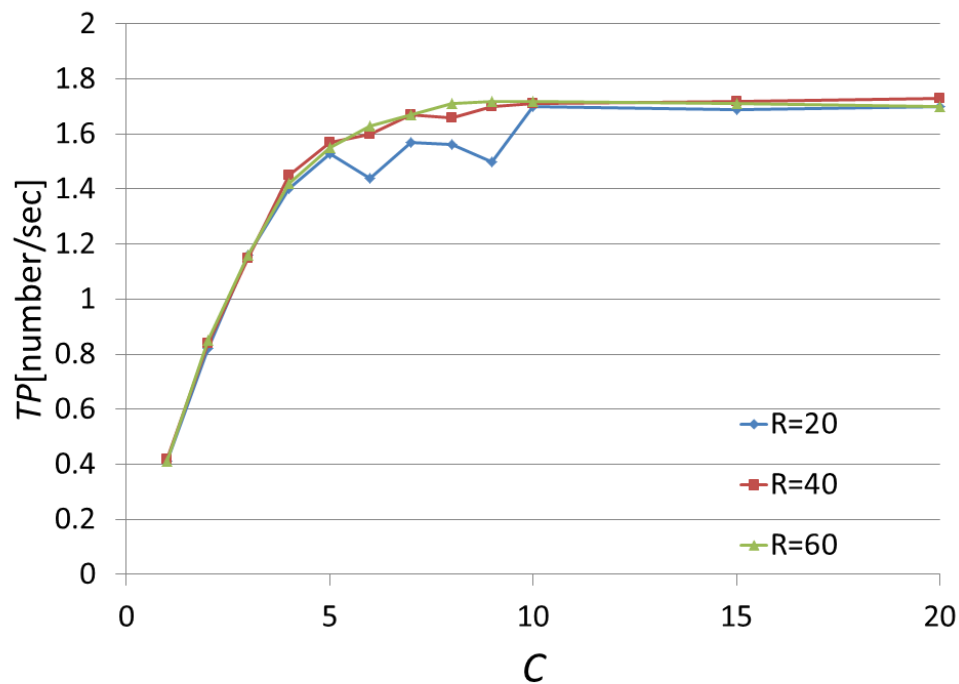


図 16: [4] で提案されたシステムの経過時間¹

¹出典：河野歩磨, 鈴木大雅, 永濱裕太
 “汎用学習管理システムと連動する IP 技術学習支援機能の提案”
 2016 年, 17 ページ 図 8

図 17: [4] で提案されたシステムのスループット²

[4] で提案されたシステムの性能評価結果を図 16、図 17 に示す。このときに用いた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た。次に、比較用に本システムでの実験を図 16、図 17 と同じリクエスト数で行った場合の結果を図 18、図 19 に示す。この時のサーバ、クライアントのスペックは図 12、図 13 と同様である。

²出典：河野歩磨, 鈴木大雅, 永濱裕太
 “汎用学習管理システムと連動する IP 技術学習支援機能の提案”
 2016 年, 18 ページ 図 9

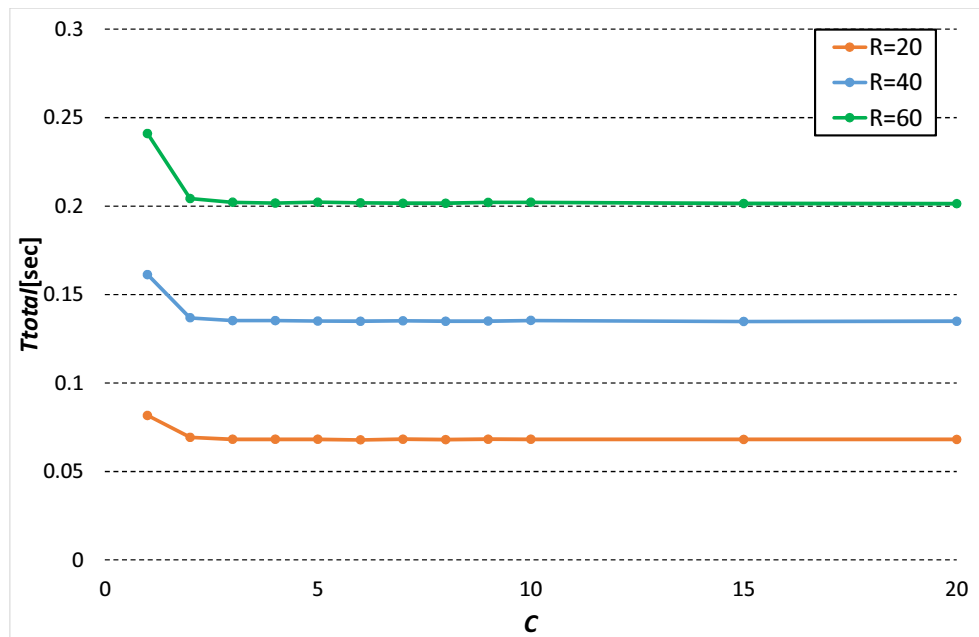


図 18: 比較用のシステムの経過時間

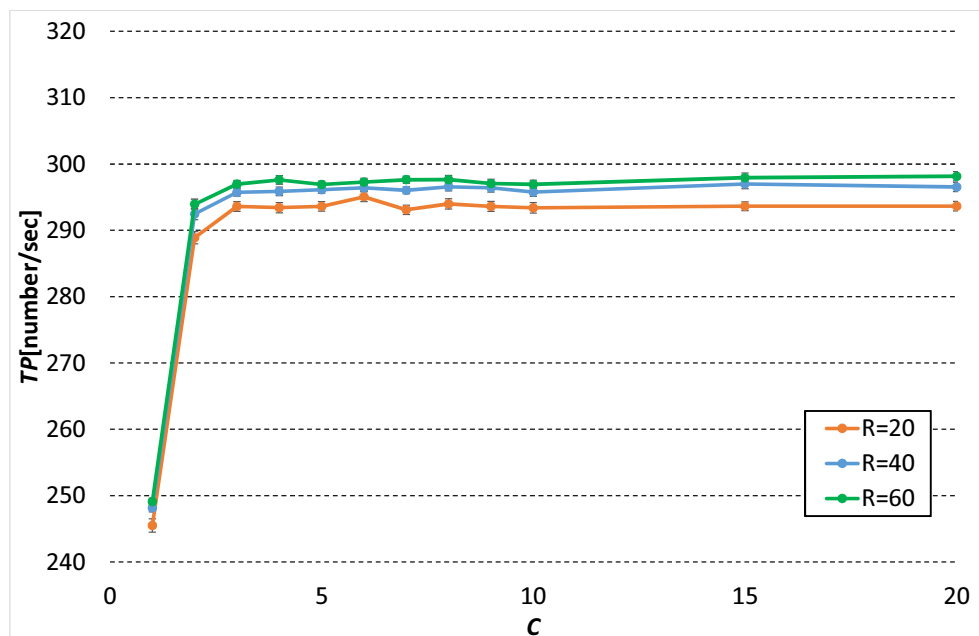


図 19: 比較用のシステムのスループット

これらの結果より、ns-3 の仕様上リクエストごとにシナリオファイルをコンパイルする必要があるため、60 人の学習者が 1 回のリクエストを同時に行っただけで最大で 2 分間近く待機状態になることがあった。そのため、独自の判定システムを用いることで大幅に応答性が向上したと言える。

5 アンケート評価

[4] では実際に学生にシステムを扱ってもらえていないため、学習者にデザインなどの評価をしてもらうことができなかった。そこで、本システムの評価実験として、実際にネットワーク層の学習を終えた情報システムデザイン学系の専門科目である情報ネットワーク概論を受講している学生 107 名に本システムを利用してもらい、私たちが作成したアンケートにより、評価を実施した。

5.1 アンケート手順

実際に利用してもらう前に、Moodle のアカウントを受講者分作成した。初期 ID を 8 文字の英数字で与え、初回ログイン時に各自でパスワードを再設定するようにして機密性を確保した。

そして Moodle にログインし、使い方などを記載したチュートリアルと本システムを、それぞれ事前説明をほとんどせずに利用してもらった。チュートリアルはパワーポイント形式のものを作成し、Moodle 上にアップロードした。アンケートの内容は以下の通りである。

- チュートリアルの説明はわかりやすかったか
- 難易度はどうだったか
- システムがみやすかったか
- システムが使いやすかったか

以上のアンケートを四択で回答してもらい、集計した。また、デザインについての意見と使用した感想をそれぞれ記述式で回答してもらい、集計をした。

5.2 アンケート結果

アンケートの有効回答数は31件で、回答率は29.0%となった。アンケートの結果、67.7%の学生がネットワーク学習の役に立つという意見であった。チュートリアルの説明はわかりやすかったと答えた学生は64.5%であった。また、本システムが扱いやすかったと答えた学生は58.1%であった。以上の結果より、本システムがネットワーク学習の支援をできていると考えられる。

しかし、事前説明なしでいきなり本システムを開くと使い方がわからず困惑したという意見や、チュートリアルを開きながらシステムを扱うことができずに不便であるという意見が挙がった。また、画面上に説明が少ないため分かり辛いという意見も挙がった。これは、システムの使い方の説明をチュートリアルに依存した結果であると考えられる。学習者がチュートリアルを必ず見るとは限らない。また、チュートリアルを一度見ただけで完全に操作方法を覚えるということは難しく、使用している途中で使い方がわからなくなった場合に一度学習を中断してチュートリアルを見る必要があるので、チュートリアルのあり方やアイコンのデザイン、ヘルプなどを工夫する必要があると分かった。この問題を解決することで、扱いやすいと答える学生が増えると考えられる。

また、自由描画モードと問題演習モードを切り替える際に、本システムではポップアップ等の確認なしに切り替わる。その際に、それまで描画していた設定をすべ

て削除する。この機能を知らずに、間違えて切り替えてしまって今までの設定をすべて消去してしまったという意見が挙がった。この意見からも、チュートリアルの説明が不足していたと分かった。

デザインについて、見やすいと答えた学生は九割程度であった。この結果から、アイコンの画像を変える必要はなく、ヘルプ機能を充実させることでシステムの扱いやすさが向上するのではないかと考えられる。

6 まとめと課題

本研究では、Web ブラウザ上で動作する独自シミュレーションソフトウェアを使用した 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、採点機能と連携を行った。このシステムに対してリクエスト数、コネクション数を変化させて性能実験を行った。その結果、本システムのパフォーマンスはサーバ、クライアントのマシンスペックにほぼ左右されず、1 人の学習者が極端に多くのリクエストを行うときに複数人でリクエストを行うときより待機時間が長くなってしまったことがわかった。

このことを改善するためには、シミュレータソフトウェアのマルチスレッド化することで解決できると考えられる。また、本研究では Moodle の採点機能との連携を実装したが、それ以外のロール機能やグループ機能などの多くの機能に対応できていない。これらの機能との連携を実装することで、問題作成の簡略化や複数人で問題に取り組むことにより知識の定着への貢献が期待できる。さらに現在はネットワーク層のルーティングに関する構築演習しか実装していないため、ネットワーク層で動作するファイアウォールの実装やデータリンク層、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。

謝辞

本研究においてアンケートにご協力いただいた東京電機大学理工学部情報システムデザイン学系の情報ネットワーク概論受講者の皆様、情報学ゼミ学部3年の皆様に謝意を表します。さらに本研究を引き継ぐ際に様々な情報を教えていただいた河野歩磨氏、鈴木大雅氏、永濱裕太氏に感謝したいと思います。また、本研究の御指導や実験への協力をして下さいました藤本准教授とシステム評価研究室の皆様に対し、ここに心より深く御礼申し上げます。

参考文献

- [1] 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”, 信学技報 113(377), pp.79–83, 2014.
- [2] 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, 2012.
- [3] 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疑似学習者機能の開発”, 情報処理学会関西支部 支部大会 講演論文集, 2015.
- [4] 河野歩磨, 鈴木大雅, 永濱裕太, “汎用学習管理システムと連動する IP 技術学習支援機能の提案”, 日本ソフトウェア科学会 第 33 回大会講演論文集, 2016.
- [5] “ns-3”, <https://www.nsnam.org/>, 2017/12/23.
- [6] “Moodle - Open-source learning platform”, <https://moodle.org/>, 2017/12/23.
- [7] D. Crockford, “The application/json Media Type for JavaScript Object Notation (JSON)”, RFC 4627, 2006.