

仮想環境ソフトウェアに基づく Linux ネットワークトラブルシューティング実習環境提供システムの開発

立岩 佑一郎† 安田 孝美† 横井 茂樹†
†名古屋大学大学院情報科学研究科

アブスト：大学や専門学校でのネットワーク管理者育成の一環として、ネットワークトラブルシューティングの実習は非常に大切である。しかしながら、壊れたスイッチングハブや設定ミスのある Linux サーバなどから構成されるトラブルのあるネットワークを生徒毎に用意するのは非常に困難であるため、現実的に実習を行うことはできない。そこで、我々はネットワークトラブルシューティングの実習環境を現実的に提供するためのシステムを新たに開発した。システムは、仮想環境ソフトウェア User-mode Linux の活用により実装されている。生徒は Linux が動作する一台の PC 上にて、問題のある仮想的なネットワークのトラブルシューティングを実習できる。このシステムの負荷実験として、CPU 使用率の計測を行った結果、システムのレスポンスにおいて実習者が大きなストレスを感じることはないことがわかった。また、大学生 13 人による評価実験によると、ネットワークトラブルシューティングの実習に有効であることが示された。しかし、ユーザインタフェースに問題を持っていることも示された。

Development of the system for providing a training environment for Linux network troubleshooting based on virtual environment software

Yuichiro TATEIWA† Takami YASUDA† Shigeki YOKOI†
† Graduate School of Information Science, Nagoya University

Abstract: As a part of fostering network administrators in universities and vocational schools, training of network troubleshooting is very important. However, since preparing networks with problems which consist of broken switching hubs and Linux servers with failure settings is very difficult, their educational institutions cannot undertake the training realistically. Consequently, we newly developed a system for realistically providing training environment for network troubleshooting. Our system is implemented by making use of virtual environment software User-mode Linux. On one Linux PC, a student can train troubleshooting in virtual networks with problems. Based on our measurement of CPU use rate during troubleshooting, it was concluded that students hardly feel stressed in the system response. And, based on questionnaire results answered by thirteen students, it was concluded that our system was highly effective for the training. However it was concluded that our system had problems with user interfaces, too.

1. はじめに

インターネットや LAN の普及により、ネットワーク管理者の育成は必要不可欠である。ネットワーク管理者育成の場である大学や専門学校の多くでは、ネットワーク管理者育成の一環として Linux サーバ構築を中心とした LAN 構築の実習と、TCP/IP 理論の講義が行われている。

ところで、大学や専門学校でのネットワーク管理者育成の一環として、初歩的なネットワークトラブルシューティングの実習を行うことも大切であると我々は考えている。実習により、トラブルの原因の絞り込み方法や、ネットワーク診断ツールの使い方を学習できるのである。また、トラブルの種類によっては、LAN 構築技能、TCP/IP 理論の「間違い探し」という角度からの復習にもなる。このように、トラブルシューティングの実習は、ネットワーク管理者を目指す者にとって非常に役に立つと言える。

しかし、大学や専門学校においてネットワーク

トラブルシューティングの実習を実際に行うのは困難である。なぜなら、壊れた機器の用意や、設定ミスのあるネットワークを用意したりすることが、現実的ではないからである。例えば、パケットロスが適切に発生するような壊れた機器を用意したり、実習者毎に提示するネットワークの機器一つ一つに設定ミスを施したりすることは非常に困難である。

これを受け、我々は大学や専門学校でのネットワークトラブルシューティングの実習を現実的に行うための環境を提供するシステム LiNeS (Linux Network Simulator) を開発してきた[1]。本システムは、一台の PC 上で動作するアプリケーションソフトウェアで、トラブルのある仮想的なネットワークを提示する機能を持つ。提示されるネットワークは、設定ミスのある Linux サーバや壊れたスイッチングハブなどの仮想機器から構成され、これらの設定や配置などを XML で記述することによって作成される。したがって、

ネットワークトラブルシューティングの実習を手軽に行うことができるという利点がある。このシステムを、我々は仮想環境ソフトウェア **User-mode Linux**[2] (以下 **UML**) の活用により実現したのである。今回、これに加えユーザインタフェースの拡張と、システムの評価実験を行ったので、本稿で併せてこれを報告する。

これまでに行われてきている研究でネットワークトラブルシューティングの実習を支援するものは見あたらなかった。本研究により大学・専門学校でのネットワーク管理者育成の質の向上が期待される。

2. 関連システム

様々なシステムを幅広く探したが、我々の目的に適合するシステムは見あたらなかった。そこで、ネットワーク構築実習に有用なシステムをあげ、それらが我々の目的に適切かどうかを検討する。なぜなら、ネットワーク構築の実習の延長上にネットワークトラブルシューティングの実習が位置するからである。

非常に精巧なネットワークシミュレーションを行えるシステムとして **OPNET**[3]、**ns2**[4]、**MAADNET**[5]、**GNS3**[6]が有名である。これらは、サーバオブジェクトやケーブルの配置を行って、ネットワークを仮想的に構築することができる。また、ネットワークのトポロジー構築の学習に対象を特化したシステムとして、早川らのシステム[7]や精廬らのシステム[8]がある。これらは、GUI上でネットワークトポロジーの構築、および各機器の **TCP/IP** の設定を学習させることを目的としている。しかしながら、以上のシステムでは本研究の実習対象とする **Linux** サーバをシミュレーションできないという問題がある。

仮想環境ソフトウェアによりネットワーク構築の実習を行うシステムとして、中川らのシステム[9]があげられる。これは、**VMware**[10]により仮想的なネットワークを実現しているため、実習者は **Linux** や **FreeBSD** を含んだネットワーク構築の実習を行うことができる。しかし、**VMware** の複数を同時に動作させるための専用のシステム環境が必要となることと、トラブルのある機器、例えば壊れたスイッチングハブなど、を実現することが困難であるという問題点を持つ。また、上田らは、我々と同様に **User-mode Linux** を活用し、手軽に使用できるネットワーク管理者教育のためのシステムを開発しているが、我々の研究とは研究対象が異なったものである[11]。我々のシステムは、サーバソフトウェアやファイアウォールの設定などアプリケーション層を中心とした学習を目的としている。一方、上田らの研究は、

Cisco Systems のシスコネットワークキングアカデミーの講義[12]の支援を目的とするものであり、ルータやスイッチといったトランスポート層・ネットワーク層を中心とした学習を目的としているシステムである。

以上により、従来のシステムでは我々の目的に優れた効果を期待できない。このため、我々はネットワークトラブルシューティングの実習環境を提供するためのシステムを新たに開発したのである。

3. システムの要件

本章では、最初に本システムの目的を整理し、その結果より本システムに必要な要件を導き出す。

本システムの目的は、基本的なネットワークトラブルシューティングの実習を行うための環境を提供することである。大学生や専門学校生が **LAN** 構築実習や **TCP/IP** 講義を終えた後で、本システムで実習することを想定している。この実習により、実習者がネットワークトラブルシューティングの基礎的な手順と、ネットワーク診断ツールの使い方を習得することが、本システムの期待する効果である。対象となる診断ツールは、**Linux** のトラブルシューティング用の代表的なツールである **ping**、**tracert**、**ifconfig**、**route**、**ps**、**netstat**、**telnet**、**arp**、**nslookup**、**tcpdump** である。これらのツールはネットワーク管理において重要で基本的なものである。

このような実習を効率的に行うには、**LAN** 構築の実習と **TCP/IP** 理論の講義で学習した知識に基づいて行えることが重要である。したがって、以下のようなシステム要件となる。

要件 A. 基本的な構成のネットワーク：**LAN** 構築実習後に使用するシステムであるので、**LAN** 構築の実習において使用したネットワークが望ましい。したがって、ネットワーク機材としては、クライアント、サーバ、スイッチ、ルータ、ネットワークケーブルを使用し、ネットワーク規模は、15 台程度のものまでを提供できるようにする。

要件 B. ネットワークトラブルの 3 つの原因：実習者に提示するネットワークトラブルの原因は、「機材故障」、「設定ミス」、および「接続ミス」とする。「機材故障」は要件 A で述べたネットワーク機材の故障で、パケットロスやパケット配送遅延といったトラブルを引き起こす原因となるものである。「設定ミス」は **IP** アドレスなどの **TCP/IP** 設定やウェブサーバ **Apache** などのサーバソフトウェア設定のミスである。これらにより、「ウェブページが閲覧できない」、「メールを受信できない」といったトラブルが引き起こされる。

「接続ミス」はネットワークケーブルの接続ミスで、初学者がよく遭遇するミスの一つである。例えば、接続ポートを間違えて接続してしまったり、コネクタが外れていたりするものである。これにより、「目標外のサーバへ接続されてしまったり」、「まったく通信できなかつたりする」といったトラブルが引き起こされる。これらの原因を複数組み合わせることで、バリエーションを増やし、難易度も上げることが出来る。

要件 C. 実習者に提示するネットワークは、実習者の管理する領域と、実習者の管理していない領域から構成されるものとする。この概念は、実際のネットワーク形態を抽象化したものであり、ネットワークトラブルの難易度をアレンジ可能にするものである。すべてのネットワーク機器が管理領域内に設置されたケースと、一部のネットワーク機器が管理領域外に設置されたケースとでは難易度が大きく異なる。例えば、後者のケースにおいて、管理内のネットワークにトラブルの原因が無いことを証明することは非常に難易度が高い実習である。

要件 D. システムの現実的な運用：実習毎の準備の手間と機材への投資が現実的な範囲内であることが、実際に実習を行おうとする場合には必須の要件となる。実習者一人一人に対してトラブルのあるネットワークを課題毎に教授者が準備する手間は大変なものである。また、故障した機材を実現するために正常な機材を意図的に破壊することは経済的ではなく、狙い通りの故障を実現させるのは困難である。したがって、教授者への負担はできるだけ減らし、故障した機材の実現も効率良く行わなければならない。

4. システムの実装

3章で述べた要件 A~D を満たすシステムを、本研究では仮想環境ソフトウェア UML によって実現した。UML は Linux 上で動作する仮想的な Linux である。ネットワーク構成機能を持ち、サーバソフトウェアなどを実行できる。加えて、Linux 上での Linux のシミュレートに特化しているためにより軽快に動作するため、一般的な性能のコンピュータで複数起動することができる。本システムでは、UML により仮想的なネットワーク機材を実現し、それらを複数使用することで構築した仮想的なネットワークを実習者へ提供している。

本システムの構成を図 1 に示す。本システムは一台の Linux 端末上で動作するアプリケーションである。使用可能な仮想的なネットワーク機材は、クライアント、スイッチングハブ、サーバ、ルータとそれらを接続するネットワークケーブル

ルである (要件 A)。これらの機材を実現している UML のソースコードを改造することにより、トラブルのあるネットワークを実現している (要件 B)。実習者は UML を直接制御するのではなく、アイコンによって提示されたインタフェースを通して、ネットワーク機材を配置したりプログラムを実行したりすることができる (要件 C)。提示されるトラブルを持つネットワークは、あらかじめ教授者が定義したデータを基に構築される (要件 D)。この定義したデータは Network database に格納されている。

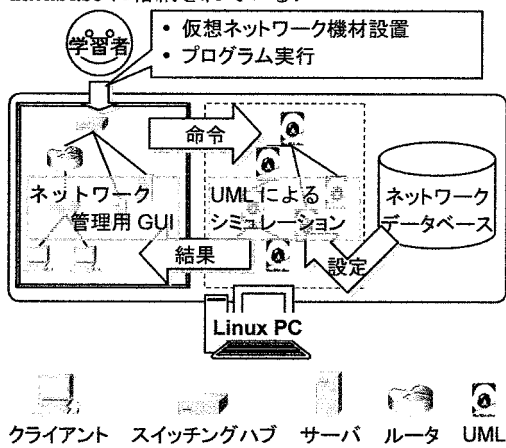


図 1 システム構成

4.1. XML によるトラブルのあるネットワークの設定

本システムは、XML ファイルにネットワークの情報を記述するだけで、自由に簡単にトラブルのあるネットワークを作り出せる。表 1 に本システムで定義した XML タグの一部を示す。init タグにはサーバ、クライアント、ルータの初期設定を、conf タグにはスイッチングハブの初期設定を設定する。各機器はここに記述された通りの初期化設定で起動された後、実習者に提示される。gift タグには仮想機器に渡したいファイル名を設定する。サーバ、クライアント、ルータは起動時に、ここに記述されたファイルを自らのデータ領域にコピーする。このようなタグを使用して、トラブルのあるネットワークの情報をファイルに定義することで、教授者は手間なく自由なネットワークを実習者に提示することができるのである。

表 1 仮想ネットワーク構築用タグの一部

タグ	説明
client	クライアントを示すタグ
init	client, router, server の初期化用シエルク립トを指定するタグ
gift	client, router, server に渡すファイ

	ルを指定するタグ
conf	switchinghub の初期化設定を指定するタグ
属性	説明
area	管理領域を示す属性
x	x 方向の設置座標を示す属性
srcid	cable の接続先機材 1
srdport	cable の接続先ポート 1
guestpath	<gift> で指定されたファイルの仮想機器内におけるファイル名

4.2. ソースコード追加による故障した機器の実現

本システムでは、UML にソースコードを追加することで、故障した機器を実現した。通常の UML には故障した機器を実現する機能はない。そこで、UML がオープンソースソフトウェアであり、そのソースコードの変更が認められているという特性を利用した。例えば、故障したスイッチングハブの実現においては、受信したパケットをロストしたり、一定時間保持した後に配送したりするようなソースコードを追加した。前節で述べた XML タグでパケットロストの確率などを指定することで、トラブルのあるネットワークに組み込むことができる。

4.3. UML の改造による設定ミスのある機器の実現

4.1 においてクライアント、サーバ、およびルータを対象としたタグ `init` により定義された初期設定を、自動的にこれらの機器に適用するための機能を実現した。これは、トラブルのあるネットワークを実習者に提示するために必要な機能である。この機能により、ネットワーク機器に特殊な設定を施した状態や、新たな設定ファイルを追加した状態で実習者に提示することができるようになる。

我々は図 3 に示すような仕組みによってこの機能を実装した。本システムのクライアント、サーバ、およびルータは UML により実装されているので、UML の従来の初期設定処理に追加する形で、UML が Network database にある初期設定を自身に適用すればよい。従って、UML の改造により、UML が `init` 処理の最後に Host File System にあるファイルを読み、その内容を自身に適用するようにした。また、そのファイルには Network database から抽出された UML 用の設定情報が記述されているようにした。

この仕組みでは UML が `init` 処理の最後で、追加の設定を自身に適用している。UML は、通常の Linux と同様、起動後にハードウェア認識やモジュールの読み込みといったカーネル処理 (Kernel) を行い、IP アドレス設定や常駐ソフト

ウェア起動などの初期設定を自身に適用し (`init`)、Login 画面の表示などのログイン処理 (Login) を行う。したがって、Login 処理により実習者に UML の制御を渡す前の段階で、追加の設定を施すのである。

そして、UML は Host File System を経由して、教授者により定義された設定情報を得ている。UML と UML を起動する OS (Host OS と呼ぶ) のデータ空間は、隔離されているためお互いのファイルを直接参照することができない。そこで、両者の間に共有のデータ空間を構築するための UML の機能の一つである Host File System を利用したのである。システムが Network database から初期設定を抽出し、それを Host File System 上のファイルに記述する。そのファイルを UML が参照することで、Network database に定義されていた初期設定を適用できる。

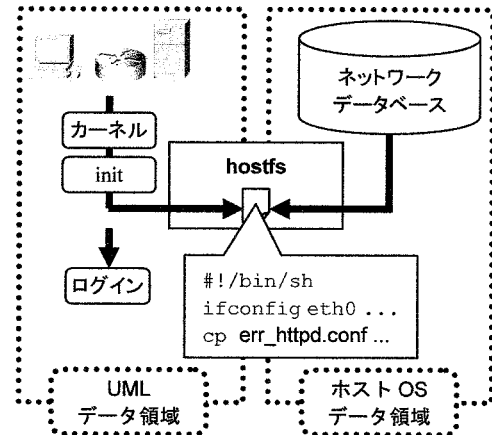


図 2 初期設定の自動的な適用の仕組み

4.4. 管理領域とケーブルの接続性の表現

実習者の制御可能な管理内領域と、制御できない管理外領域をユーザインタフェースの工夫によって実装した。制御可能な管理内領域では、実習者がすべてのネットワーク機器を確認でき、それらを自由に操作・設置・撤去できる必要がある。制御できない管理外領域では、実習者がネットワーク機器を見ることができず、機器の操作・設置・撤去ができてはならない必要がある。そこで、図 4 に示すように、ネットワークを表示する画面の背景により管理内および管理外の領域を表現し、管理内の機器はアイコンを表示しておき、管理外の機器は基本的にアイコンを表示しないようにした。しかし、トラブルシューティングの作業の便宜を考慮し、管理外の領域の機器であっても、設定記述によってはアイコンを表示可能にもしてある。

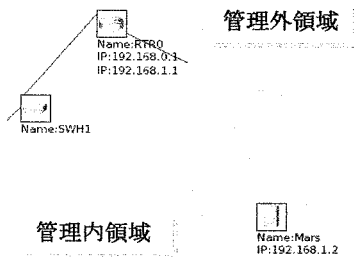


図 3 管理内領域と管理外領域の表現

ケーブルの接続性の表現もユーザインタフェースの工夫によって実装した。ケーブルの接続性の表現は、

ケーブルの一端が、

- ・機器 X のポート Y に接続されている
- ・機器 X に接続されていると錯覚する
- ・どの機器にも明らかに接続されていない

を表現することである。そこで、表 2 に示すように、ポートを表すアイコンを作成し、各機器の裏にそれらを配置し、ケーブルの両端に接続状態を示すアイコンを表示することによって実装した。これにより、「一見接続されているように見えるが、実際には未接続となっているケーブル」などをトラブルの原因として使用できるようになる。

表 2 ケーブルの接続性の表現

表現項目	アイコン
仮想機器表面 (スイッチングハブの例)	
仮想機器裏面 (スイッチングハブの例)	
未接続状態のコネクタ	
仮想機器へ接続された状態のコネクタ	
接続されていると錯覚するケーブル配線 (スイッチングハブの例)	

5. 本システムのメリットと実行例

5.1. 本システムのメリット

本システムの主なメリットは 2 つある。

一つは、教授者が簡単にトラブルのあるネットワークを実習者に提供できることである。実機を使用して、トラブルのあるネットワークを提供するのは大変なことである。故障した機材の実現や、実習者毎に課題毎のネットワークを構築し、機器一つ一つに必要な設定を施すことは、教授者にとって非常に負担となる。しかし、本システムでは、教授者は XML ファイルに提供したいネットワーク情報を記述するだけでよいのである。

もう一つは、実習者が安心して自由な発想で、ネットワークトラブルシューティングの作業を行えることである。試行錯誤を伴うトラブルシューティングの作業は、各機器のシステム設定を破壊してしまう可能性がある。最悪の場合、再インストールをしなければならないため、実習にとって相応しい環境ではない。本システムでは、仮想環境ソフトウェアによる仮想機器を使用することで、システムを再起動するだけで各機器をリカバリーできるため、実機より効率よく実習できる。例えば、Apache の設定ファイルなどを再起不能にしてしまった場合や、最初からトラブルシューティングを手早くやり直したい場合などに、非常に有効であると言える。

5.2. 本システムの実行例

本システムでは、まず、実習者が実習したいネットワークトラブルを選択する。そうすると、システムが該当するネットワークを構築し図 4 のように実習者へ提示する。その後、実習者は症状の確認やネットワーク診断ツールの実行を行い、トラブルシューティング作業を行う (図 5~8)。

図 4 に提示されたネットワークは、「クライアント Sun からサーバ Mars のウェブページ (<http://www.mars.com/>) の閲覧ができない」というトラブルを持つものである。このトラブルは、サーバソフトウェア Apache の設定とルータのファイアウォールの設定が適切でないことに起因する。

実習者は図 4 のネットワーク表示ウィンドウ上部のボタンによって仮想機器を追加したり削除したりできる。また、クライアント制御ウィンドウでは、ウェブブラウザやメーラーなどを起動できる。ルータ制御ウィンドウでは、IP アドレスや経路制御の設定を行うことができる。サーバ制御ウィンドウでは、実際のサーバ制御と同じようにコマンド入力によりサーバを制御できる。

トラブルシューティング作業において、まず、実習者は web ブラウザを使用しサーバ Mars のウェブページが閲覧できないことを確認する (図 5(1))。次に、ps によりサーバソフトウェア Apache が稼働していることを確認し (図 6(1))、netstat によりサーバソフトウェア Apache の不適切な動作を発見し (図 6(2))、Apache の設定ファイル httpd.conf をエディタで編集し、Apache を再起動して設定の変更を有効にする。この作業を行った結果、クライアント Venus からは閲覧できるが、依然としてクライアント Sun や Saturn から閲覧することができない。そこで、ルータ RTR0 の設定を調査しファイアウォールの設定が不適切であることを突き止め (図 7(1))、これを修正する。以上の作業によってクライアント

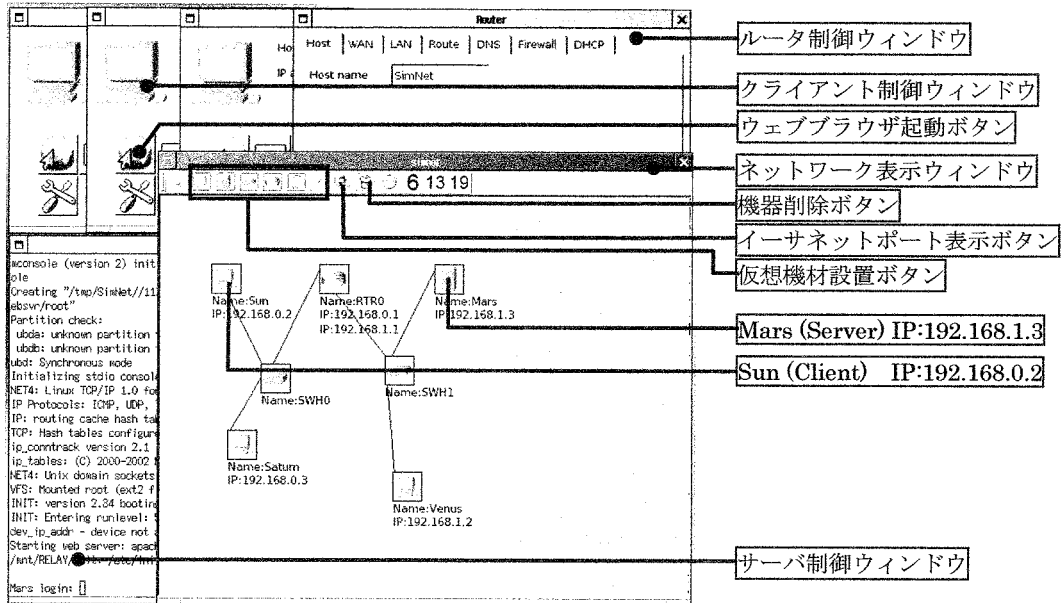


図 4 トラブルのあるネットワーク

ト Sun からサーバ Mars のウェブページを閲覧できるようになり (図 8), トラブルを解決できたことになる。

実習者はこのような作業を通じて、ネットワーク診断ツールの実習や、Apache の設定などの LAN 構築技能の復習、ファイアウォールに設定したポート番号の役割などの TCP/IP 理論の復習を行える。また、様々な機器からアプローチすることによって原因を絞り込むという、基本的なトラブルシューティングの手順を実習できるのである。

本システムでは、このような複雑なネットワークであっても、XML ファイルの記述により、簡単に実習者に提供することができる。個々の原因を複合させることで、より難易度の高いネットワークトラブルシューティングを実習させることができるのである。

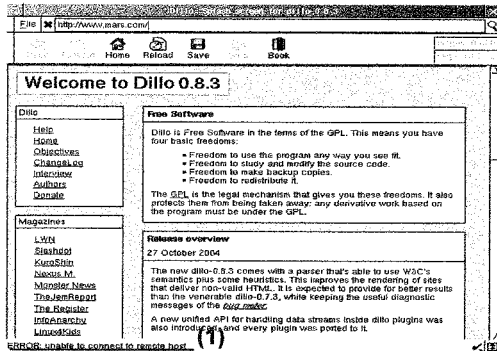


図 5 Mars 上のウェブページを取得失敗

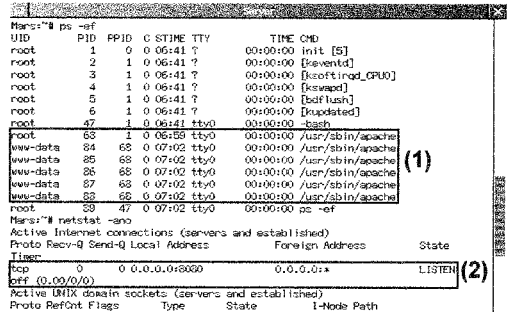


図 6 サーバでのネットワーク診断ツール ps, netstat の実行結果

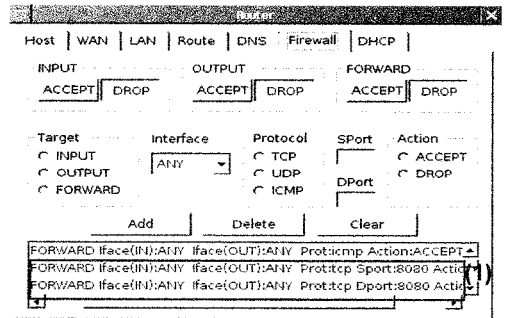


図 7 ルータのファイアウォールの設定

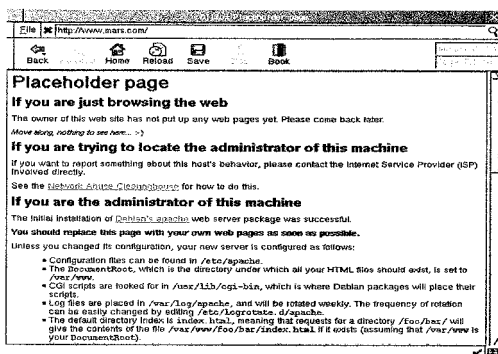


図 8 Mars 上のウェブページを取得成功

6. 実証実験

ネットワークトラブルシューティングのための実習環境の提供という目的に対する本システムの有効性を測るために、システムの性能評価とアンケートを行った。実験は、教育現場での使用を考慮した性能の PC (CPU: PentiumM1.6GHz, メモリ: 512MB) を使用した。この PC で動作すれば、教育現場においても十分に本システムを使用できると考えられる。

6.1. システムの性能評価

トラブルのある 2 つのネットワークにおいて、システムの性能評価を行った。一つは 6 台 (クライアント 2 台, スイッチングハブ 2 台, ルータ 1 台, サーバ 1 台) から構成され、スイッチングハブの 1 台が故障しているもの (以下 Network A と呼ぶ) で、もう一つは 15 台 (クライアント 4 台, スイッチングハブ 5 台, ルータ 4 台, サーバ 2 台) から構成され、いくつかの機器に設定ミスのあるもの (以下 Network B と呼ぶ) である。前者は、本研究が想定している平均的な規模のネットワークで、後者は本研究が想定している最大の規模のネットワークである。両者においてネットワークトラブルシューティングを行った際の CPU 負荷を計測することによって、システムが実用に耐えるのかの推定を試みた。

図 9, 図 10 にネットワークトラブルシューティングの作業過程の CPU 使用率を示す。図中 (1) は、システムが各機器を起動し設定を施している期間である。本システムが、実習者へのトラブルのあるネットワークの提示完了までに要した時間は、Network A は 33 秒, Network B は 98 秒であった。その後のトラブルシューティングの作業の期間は図中 (2) である。Network A では ping による原因の特定や故障機器の交換などを行った。ping 実行のための端末の起動には約 1 秒であった。Network B ではウェブ閲覧(ブ

ラウザの起動時間は約 3 秒, Web ページ取得時間は約 1 秒) や web サーバ Apache の再起動 (再起動時間は約 2 秒) などを行った。

ネットワークの準備時間, 各アプリケーションの起動時間・実行時間, 加えて作業中の CPU 使用率から考えると, システムのレスポンスにおいて実習者がストレスを感じることは少ないと思われる。15 台と機器の台数が増えるとシステム負荷も大きくなるが, 実習者は作業を一つずつ個別に実行するので, 大きな問題ではないと考えている。

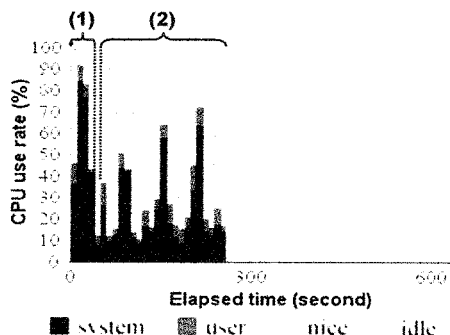


図 9 6 台のネットワークでの CPU 使用率

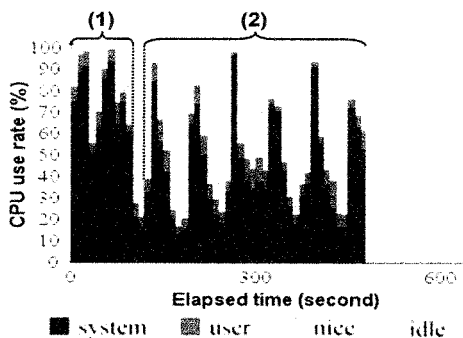


図 10 15 台のネットワークでの CPU 使用率

6.2. アンケート結果

表 3 に示すような質問項目, および自由記述欄から構成されるアンケートを行った。アンケート項目に対して {5 そう思う | 4 どちらかと言えばそう思う | 3 どちらとも言えない | 2 あまりそう思わない | 1 そうは思わない} の 5 段階評価と自由記述で評価を実施した。被験者は, 情報系大学生・大学院生 13 名で, TCP/IP を学んだことがあり, 10 台程度の小規模な LAN を構築できる者である。彼らは, 本研究で想定している対象者とほぼ同じステータスである。このアンケートの狙いは, 学習効果の見込みを推測することと, システムの操作感を明らかにすることにより, システムの有効性の検証と, 今後の課題を明らかに

することである。

表 3 質問項目とその平均点, および自由記述 (一部要約)

質問	評価
Q1. ネットワークトラブルシューティングの実習に役立ちましたか?	4.5
Q2. 診断ツールの実習に役立ちましたか?	4.5
Q3. TCP/IP 理論の復習に役立ちましたか?	4.4
Q4. LAN 構築技能の復習に役立ちましたか?	4.5
Q5. システムのユーザインタフェースは使いやすかったですか?	3.6
Q6. トラブルのあるネットワークの準備時間 (自動構築) は許容範囲内でしたか?	4.1
Q7. システムの反応は許容範囲内でありましたか?	4.1
Q8. 操作ミスが致命傷にならないことは, 実習に役立ちましたか?	4.9
自由記述	
A. トラブルの可能性を示したりわからない場合に怪しいポイントを点滅したりといった拡張をされると良い B. システムの操作に慣れが必要だと思う C. ウィンドウの数が増えてくると扱いにくくなる D. 仮想機器は壊してしまっても問題ないので良い	

アンケート結果より, 学習に関する評価は概ね良好であったが, 操作感に関する評価は芳しくなかった。特に, ユーザインタフェースに関して課題が残っているといえる。

学習に関する評価は, Q1~4 および自由記述 A である。Q1,2 が本システムの主目的であるトラブルシューティングの実習に関する評価で, Q3,4 が副目的である LAN 構築技能と TCP/IP 理論の復習に関する評価である。これらの評価結果が概ね良好であることにより, 本システムを講義に組み込んで使用した場合でも, 高い学習効果を期待できると推定できる。自由記述 A に述べられているヘルプ機能に関しては, 更に高い実習効果への課題と考えている。

システム操作に関する評価は, Q5~8 および自由記述 B,C,D である。ユーザインタフェースに関する評価(Q.5, 自由記述 B,C)が良好ではないので, ユーザインタフェースの改良は今後の課題である。システム反応に関する評価(Q6,7)は, 平均的な評価であった。5.1 節で示した結果と併せて考察すると, 本システムは実用に耐えるものであると思われる。Q8 の評価が非常に高かったことと自由記述 D により, トラブルシューティングのような試行錯誤を伴う作業の実習において, 本システムのような仮想機器を使用することの優位性が示された。

7. まとめ

大学・専門学校でのネットワークトラブルシューティングの実習を現実的に行うための環境を提供するシステムの開発を目的とした研究を行ってきた。今回, 研究の第一ステップとして, トラブルのある仮想的なネットワークを提示する機能をもったシステムを開発した。本システムの有効性を測るために, システムの性能評価とアンケートを行った。性能評価として CPU 使用率を計測した結果, システムのレスポンスにおいて実習者がストレスを感じる可能性は低いことがわかった。また, 大学生・大学院生 13 人の評価アンケートにより, トラブルシューティングの実習, TCP/IP 理論および LAN 構築技能の復習に高い効果を期待できることがわかった。一方で, ユーザインタフェースに改善の余地があることもわかった。研究の次のステップとして, システムの実践導入を目標に, システムの使いやすさを念頭においた改善と, 教育コースの構築を行うことを考えている。

謝辞: 本研究の一部は, 科研費および (財) 電気通信普及財団の研究助成による。

8. 参考文献

- [1] 立岩佑一郎, 安田孝美, 横井茂樹: 仮想環境ソフトウェアに基づくネットワークトラブルシューティング実習環境提供システムの評価, 第 6 回情報科学技術フォーラム, 情報科学技術レターズ pp.469-470, 2007.
- [2] The User-mode Linux Kernel Home Page: <http://user-mode-linux.sourceforge.net/>.
- [3] OPNET: <http://www.opnet.com/>.
- [4] The Network Simulator ns2: <http://www.isi.edu/nsnam/ns/>.
- [5] MAADNET: <http://www.maadnet.net/>.
- [6] Graphical Network Simulator: <http://www.gns3.net/>.
- [7] 早川正昭, 丹野克彦, 山本洋雄, 中山実, 清水康敬: LAN 構築シミュレータの開発と教育手法の改善, 教育システム情報学会 2 6 回全国大会講演論文集, E5-4, pp.367-368 (2001) .
- [8] 精廬幹人, 木村昌史: 教育向けネットワークシミュレータの開発, 情報処理学会 6 5 回全国大会講演論文集, 2D-2, pp.273-274 (2003) .
- [9] 中川泰宏, 須田宇宙, 三井田惇郎, 浮貝雅裕: VMware を利用した学習用 LAN 構築支援システムの開発, 教育システム学会誌, Vol24, 教育システム情報学会, pp.126-136 (2007) .
- [10] VMware - Virtualization Software: <http://www.vmware.com/>.
- [11] 上田拓実, 井口信和: 仮想 Linux 環境を用いたネットワーク教育システムのための仮想ルータと GUI の実装, 第 6 回情報科学技術フォーラム講演論文集, J-026, pp.447-448 (2007) .
- [12] Cisco Networking Academy: <http://www.cisco.com/web/learning/netacad/>.