

## 仮想環境ソフトウェアに基づくネットワーク処理可視化教育システムの開発

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

書籍や講義による TCP/IP の学習を支援するためのシステムを作成した。ネットワークを仮想的かつ自由に構築でき、通信のシミュレーション結果に基づいてネットワーク処理を可視化表示できる。仮想的なネットワークは仮想 OS UserModeLinux の活用により実現している。可視化表示用のデータは、仮想 OS 内のネットワーク処理内容と通信データである。UserModeLinux のソースコードの改変と、パケットダンププログラムにより、これらを収集している。

## Development of a system for education by visualizing network processing based on virtual environment software

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

We have developed a system to support learning of TCP/IP through books or lectures. This system has two functions: One is that it can construct a network freely and virtually. The other is that it enables users to visualize network processing based on a telecommunication simulation result. This system employs the Linux Virtual OS User Mode to create a virtual network. Data for visualization comprise network processing records and telecommunication data in the Virtual OS. This system collects these data by changing the source code of Linux User Mode and utilizing a packet dumping program.

## 1. はじめに

インターネットや LAN の普及に伴い、ネットワーク技術者の効率的な養成が求められてきている。特にネットワーク技術の基礎知識である TCP/IP は、大学や専門学校において講義が開かれ、書店では多くの入門書が販売されている。これらにより TCP/IP の習得には一定の成果が見られるが、内容が複雑で難解な分野では、不完全な理解のまま終わってしまう場合もある。そのような分野の中で、特に重要なものとしてインターネット層とトランスポート層の動作の仕組みがあげられる。

この分野に対し、従来の書籍や講義による学習では静的な情報を基にした机上での解説となるため「学習者が立てた仮説を検証しづらい」、「実感がわかない」という原因により十分な学習効果が得られていない。また、書籍や講義の補助として、通信実験による学習が行われている。この方法は、実際のネットワークで発生した通信パケットを閲覧することにより学習を行うものである。この方法では、ネットワーク設定の変更や実際の通信を行えるため、書籍や講義による方法の問題点に対しある程度の効果はあるが、「実験条件設定の自由度」「学習情報のわかりやすさ」に問題がある。

このような問題点に対し我々は、ネットワークシミュレーションに基づく可視化アニメーション表示システムを提案してきた[1][2][3]。本システムは、通信実験を仮想的なネットワークにて行い、学習のための情報を直感的でわかりやすい可視化アニメーションという形で表示する。これにより、実際のネットワークを使用した通信実験の利点を継承しつつ、その問題点を補うシステムとして、書籍や教科書による学習を支援することができる。

## 2. 関連研究

まず、TCP/IP の学習を支援するという点で本研究に類似した研究について述べる。荒井らは、ネットワークを流れるデータを取得し可視化表示することで、データ構造と通信手順を学習するツールと、TCP の制御方法を学習するツールを開発し[4][5]、その学習効果の検証を行っている[6]。また、山根らは、ネットワークを流れるデータを取得し、TCP のセッション単位に表示するパケットモニタリングツールの開発を行っている[7]。これら方法による学習用情報は、ネットワークシミュレーションに基づく本システムの学習用情報

と大きく異なる。

次に、仮想的にネットワークを構築できるシステムという点で本システムに類似した研究について述べる。早川らは、コンピュータ上に ICMP 機能をシミュレートできるネットワークを仮想的に構築することによって、ネットワークの構築方法の学習を目的とするシステムの開発を行っている[8]。また、南カリフォルニア大学では、仮想的なネットワークの構築を行えるネットワークシミュレータの開発が行われている[9]。このシステムは、ボトルネックの計測といった専門的な利用を目的としている。また、精槿らはコンピュータ上に仮想的なネットワークを構築することによって、ネットワークの構築方法を学習するためのツールの開発を行っている[10]。これらの研究は、仮想的にネットワークを構築するという点では本研究に類似しているが、TCP/IP の学習にとって有効な情報が少ないので、TCP/IP の学習には向いていない。

## 3. 従来法の問題点と提案法

### 3.1 TCP/IP 学習における問題分野と従来法の問題点

本研究では TCP/IP の学習範囲のうち、インターネット層とトランスポート層の動作の仕組みを対象範囲とする。これは、情報系大学院生を対象とした簡単なヒアリングにおいて、特に理解しづらい分野に挙げたことと、TCP/IP の学習分野における重要性を考慮した結果である。具体的な学習対象を表 1 に示す。

この分野における書籍や講義による学習方法は、以下のような問題点が存在する。

#### 1. 学習者の立てた仮説の検証がしづらい

TCP/IP の中でも抽象的な分野であり、習得には理解を必要とするため、学習者は理論を元に自ら仮説を立て、その仮説の正誤によって徐々に理解していく方法が有効である。しかし、書籍や講義による説明は主に静的な情報であるため、仮説に適合した条件とならない場合が多く、検証することは難しい。

#### 2. 実感がわかない

書籍や講義では、机上の説明となってしまう実感がわきにくい。そのため、感覚的にイメージとして捉えることができず、知識が定着しづらくなる。

ところで、教科書や講義での学習方法の補助として、通信実験におけるパケット閲覧による学習が行われている。この方法は、まず、実際のネットワークにおいて、任意のクライアントマシンにパケットダンププログラムを実行させておく。そして、そのクライアント

表 1 : 学習分野と表示対象

表示対象	インターネット層				トランスポート層			
	再構築	分割・ HOPメッセージ	ルーティング	IP アドレス解 理	コネクション管 理	シーケンス制御	ウィンドウ制御	フロー制御
パケットデータ	○	○	○	○	○	○	○	○
パケット経路	—	○	◎	○	○	○	○	○
パケット書込・読込処理	◎	◎	◎	◎	◎	◎	◎	◎
ルーティングテーブル	◎	◎	◎	—	—	—	—	—
ARP テーブル	—	—	—	◎	—	—	—	—
タイマー	—	—	—	○	○	—	—	—
キュー、バッファ	◎	—	—	—	—	◎	◎	◎
ソケット	—	—	—	—	◎	—	—	—

非常に有効:◎, 有効:○, 無関係:—

マシンから他のマシンへ通信を行い、通信の際に送受信したパケットの内容を閲覧する方法である。

この方法は1, 2の問題点に対しある程度の効果を出している。1に関しては、通信実験であれば、ネットワークを自由に構築し、通信結果を検証することによって仮説の検証をすることができる。2に関しては、ウェブの閲覧やメールの送受信といった身近な通信を行い、その結果を題材にした学習をすることができるので、イメージしやすく実感もわく。

しかし、実際に行うにはいくつかの問題点も存在する。実験環境の面から見た場合、「実験条件の設定の自由度」と「経済的な負担」が問題となる。構内 LAN などの既存のネットワークを実験環境とする場合、ネットワーク管理の関係やセキュリティ面でのリスクから、設定を変更することは難しい。そのため、ネットワーク形態の組み換えや IP アドレスの変更など、実験に重要な条件の設定を行うことができない。また、新たに LAN を作成して実験環境とする場合、機材をそろえることによる経済的負担が大きい。

一方、学習効果の面から見た場合、「学習情報のわかりやすさ」が問題となる。パケットダンププログラムによる表示は、取得したパケットの内容が文字で表示されているだけであり、イラストや解説など理解を支援する情報が乏しい。また、各マシン間で得た学習用の情報の同期がとりにくいことも問題である。この方法では、情報は時刻順に整理することになるが、各マシン間での時刻の同期の精度が、通信時間の記録に

必要となる精度より低いためである。各マシンで取得した通信パケットを送受信順に見ることは通信の理解に重要なことであるが、通信パケットを時刻順に整理した表示では、例えば、送信前のパケットが受信されているといった現象が発生してしまい、学習に支障が出てしまう。

### 3. 2 システムの提案

本研究では、書籍や講義による学習を支援するためのシステムとして、ネットワークシミュレーションに基づく学習用アニメーション表示システムを提案する。

#### 3.2.1 ネットワークシミュレーションによる通信実験

学習者が実感を持つことができ、イメージすることができるようにするには、身近な通信を題材とした学習情報を提示することが大切である。本システムでは、「ウェブ閲覧」、「メール送信」、「導通確認」などのネットワーク通信を実行できるようにする。

ネットワークシミュレーションに使用する機器は、学習に必要な動作や仕組みを実装しており、学習者の身近にあるものであることが望ましい。本システムでは、ネットワーク通信における、クライアント役、またはサーバ役として「コンピュータ」を、インターネット層の経路制御の動作を行わせるために「ルータ」を、それらを接続するために「スイッチングハブ」を使用可能とする。

### 3.2.2 学習に必要なとなる情報

通信の結果は学習に重要な情報であるため、学習者の行った通信結果を表示する必要がある。例えば、ウェブ閲覧を行った場合、成功すればサーバ側にあるコンテンツがクライアントのブラウザに表示され、失敗すればエラーページがクライアントのブラウザに表示されるようにする。学習者はこの結果を見ることにより実感をわかせることができたり、結果を通信の成功理由や失敗理由の推定などに使用することができたりする。

学習対象を理解するために必要となる情報を表1に示す。パケットの内容とルーティングテーブル、ARPテーブルは実機を用いた通信実験においても表示できるものである。ソケット、タイマー、キュー、パケット処理は、ネットワーク機器内部の情報であり、本システムにおいて表示可能となるものである。

### 3.2.3 直感的でわかりやすい表示方法

TCP/IP 学習のためには、通信全体からみた表示対象の位置付けや関連性がわかるように表示することが大切である。ネットワーク通信は個々の表示対象が関連しあうことによって成立している。従って、表示対象単独での表示では、位置づけや関連性を学習することができない。本システムでは、すべての表示を通信の進捗に同期させ、構築したネットワーク全体の表示を起点とし、ネットワーク機器の内部へ向かって、学習者よりの視点から機器内部への視点へ順に辿ることができるようにする。

表示対象は書籍や講義で使用された書式に近い形で表示する。本システムの使用される場面は、書籍や講義にて学習した後である。従って、学習者が書籍や講義にて現れた表示と本システムの表示が同じ書式であれば、抵抗なく学習に入ることができる。書式の例としては、パケットデータの書式や、TCP/IP 階層モデルに基づいたネットワーク機器内部の表示などがあげられる。

表示対象が通信の進捗状況によって変化していく様子を表現するために、アニメーション表示により直感的に理解できるようにする。例えば、ルーティングテーブルの情報が追加される様子やパケットの位置の変化を、ルーティングテーブルを表すアイコンの更新や、パケットを表すアイコンの移動によって表現する。

## 4. 提案法の実現方法

### 4. 1 システム全体の構成

本システムの構成を図1に示す。本システムでは、仮想的なネットワークと通信シミュレーションを実現するために、仮想環境ソフトウェア User-mode Linux[11] (以下 UML) を活用している。仮想環境ソフトウェアを実行している OS は Linux である。また、学習者が UML を操作するため、学習情報を表示するための GUI のコントローラ画面を用意した。このコントローラ画面と UML の接続には擬似端末という技術を用いている。

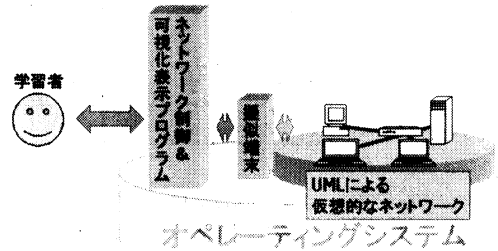


図 1 : システム構成図

### 4. 2 仮想環境ソフトウェア UML による仮想的なネットワークの実現方法

#### 4.2.1 仮想環境ソフトウェアに必要な条件

本研究では、仮想的なネットワークを仮想環境ソフトウェアによって実現している。仮想的なネットワーク機器 1 台に対して、仮想環境ソフトウェア 1 つを実行する。このような仮想的なネットワーク機器を複数実現し機器間でデータ交換を行えるように接続することで、仮想的なネットワークをシミュレートしている。

本研究の目的とする学習実験を行うためには、仮想環境ソフトウェアが下記の条件を満たす必要がある。

1. 軽快な動作
2. ソースコード公開による拡張性

仮想環境ソフトウェアは一般的な性能を持つコンピュータ上で複数動作できなければならない。そのため、CPU やメモリといったリソースの消費が少ないものが望ましい。また、本研究で表示対象としているものの中には、ネットワーク機器内部の処理記録を必要とするものがある。通常の仮想環境ソフトウェアにはこのような機能は存在しないため、ソースコードを改変することにより新たに機能を付け加えられるようなソフトウェアでなければならない。

#### 4.2.2 UML による仮想ネットワーク機器の実現

本研究では、4.2.1 で述べた条件を満たす仮想環境ソフトウェアとして User-mode Linux(UML)を選択した。UMLはLinux上で動作する仮想的なLinuxである。コンピュータとルータはUMLを用いて実現し、スイッチングハブはUML 付属のプログラムuml\_switchにより実現する。各ネットワーク機器間のデータ転送はuml\_switchと接続することにより可能となる。

uml\_switchは単体で動作するが、UMLをコンピュータやルータとして動作させるには、Linux同様、ディストリビューションが必要となる。コンピュータとルータにて実現したい機能を実装したプログラムを搭載したディストリビューションが各々必要である。

しかし、公開されているディストリビューションでは搭載しているプログラム数が多すぎるためメモリを多量に消費したり、少なすぎて実現したい機能がなかったりする。本研究で使用するディストリビューションは、複数台動作させることと、起動時間を考慮すれば、必要最小限のプログラムを搭載するものが望ましい。そこで、OSとして機能するための必要最低限のプログラムを持つディストリビューションに、通信の実行に必要なサーバプログラムの搭載と、新たに実装した軽量なクライアントプログラムの搭載をすることで、本システムにとって望ましいディストリビューションを使用している。

#### 4.2.3 擬似端末による GUI コントローラからの UML の制御

本システムの使用者はTCP/IPの学習を目的としており、システムの操作に関する負担は極力減らすことが望まれる。従って、ネットワークの構築は、マウス操作中心となるGUI(Graphical User Interface)ベースのコントローラにて行うことが望ましい。ネットワーク構築にて使用するUMLはコンソールアプリケーションであるため、コマンド入力により制御することになる。コマンド入力による制御では、様々なコマンドを覚えなければならず、学習者に負担が大きい。

本研究では擬似端末という技術によって、作成したGUIコントローラをUMLと接続することで、UMLをGUIベースのコントローラから制御できるようにした。これは、UML自身は別途作成されたGUIベースのプログラムと接続できる機能を持っていないためである。擬似端末はLinux特有の技術で、GUIとコンソールアプリケーションを接続する機能を持つ。GUI

コントローラが発信したメッセージは擬似端末を經由してUMLへ伝達される。UMLはコンソールからメッセージが送られてきたと解釈し、メッセージを実行する。実行結果としてコンソールに表示されるメッセージは擬似端末に渡されGUIへ伝達される。

### 4. 3 表示に必要なデータと取得方法の提案

#### 4.3.1 表示対象に必要なデータの取得方法

本システムでの表示対象と、表示に必要なデータを表2に示す。必要となるデータを取得するには、UML内部の処理を記録し取得する必要がある。ネットワーク通信が行われている過程で、パケットやルーティングテーブルなどに対して行われた処理の記録が必要となる。

表 2 : 表示対象と必要データ

表示対象	必要データ
パケット	時刻、内容、位置
パケット書込み・読み込み処理	時刻、パケットヘッダ構築、パケット分割、など
ルーティングテーブル	時刻、追加・削除・検索結果
ARPテーブル	時刻、追加・削除・検索結果
タイマー	時刻、実行情報
キュー・バッファ	時刻、送信待機パケット、受信待機パケットなど
ソケット	時刻、待機アドレス、待機ポート番号など

しかし、UMLには内部処理を記録し取得するための機能は存在しないため、本システムでは図2に示すような仕組みを開発した。UMLにおいてネットワーク処理を行っている部分は、ネットワーク処理手続きとネットワークインタフェースである。この部分に関するデータを、図右下の可視化表示部に伝達することがこの仕組みの目的である。ネットワーク処理手続きでは、ルーティングテーブルやソケットに関する処理とパケットへのデータ処理を行い、ネットワークインタフェースでは外部ネットワークとのパケットの送受を行っている。

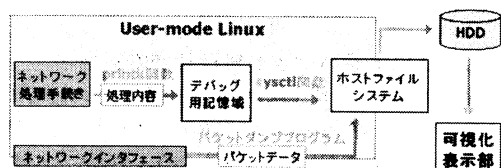


図 2 : UML 内部データの取得方法

UML 内のデータを、UML を実行している OS、Linux に伝達する方法としてホストファイルシステムを利用した。ホストファイルシステムは Linux の持つハードディスク領域の一部を UML の持つ仮想的なハードディスク領域に結びつける UML の機能である。UML 内のアプリケーションがホストファイルシステムに記録した内容は、Linux のアプリケーションから読み取ることができる。

しかし、ネットワーク処理手続きの処理データとネットワークインタフェースのデータを、通常のアプリケーションが読み込むことは困難である。この原因は UML のデータ空間構成にある。UML はカーネル空間とユーザ空間の 2 つのデータ空間を持っており、お互いのデータ空間は通常の方法ではアクセスできない仕組みになっている。カーネル空間に属するネットワーク処理手続きとネットワークインタフェースのデータは、ユーザ空間に属するアプリケーションからアクセスできず、ホストファイルシステムへ書き込むこともできない。

本研究ではネットワーク処理手続き内でのデータの記録は `printk` 関数とデバッグ用記憶域と `sysctl` 関数を実装したアプリケーションによってホストファイルシステムへ伝達する仕組みを開発した。`printk` 関数は UML のデバッグに用いられる関数で、UML の内部処理をデバッグ用記憶域に記録することができる機能を持つ。デバッグ用記憶域はカーネル空間とユーザ空間の両方からアクセス可能な特殊な記憶空間である。この空間からユーザ空間側からデータを読み出すための関数が `sysctl` 関数である。この関数を実装したアプリケーションによってデバッグ用記憶域に出力された処理記録をホストファイルシステムへ渡している。

ネットワークインタフェースのデータは、`libpcap[12]` という特殊なライブラリを利用することで取得している。`libpcap` はパケットダンププログラムに使用されているライブラリで、ネットワークインタフェースを通過したデータを取得することができる機能を持つ。本システムでは、プログラムの軽量化も考慮し、一般的なパケットダンププログラムではなく、必要最低限の機能を持った軽量のプログラムを実装し、使用している。

#### 4.3.2 ID によるパケット処理管理の実現

ネットワーク処理手続き内におけるパケットに対する処理を記録する際には、記録データとパケットの対応ができる形で記録する必要がある。手続き内にはパケットが複数存在し、様々な箇所において処理が行

われるためである。

ここで、「パケットデータ+処理情報」という記録方法が考えられる。この方法では、処理情報とパケットデータの対応付けはできるが、記録できる数がわずかとなってしまふ。パケットデータのサイズは約 1500 バイトであるのに対し、デバッグ用記憶域が約 16000 バイトであるためである。

そこで、本システムでは個々のパケットに 16 バイトの ID を付加し、「ID+処理情報」という記録形式によって、パケットと処理情報の対応を維持したまま多くの処理情報を記録可能にした(図 3)。ID はパケットの格納変数に付加している。そのため、パケットデータそのものにはまったく影響を与えることないので、通信への影響もない。

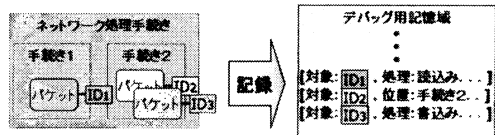


図 3 : ID による処理情報管理

## 5. 実行画面

ネットワークの仮想的な構築についての実行画面を図 4 に示す。(1)に表示されているネットワーク機器を示すアイコンをマウスで選択し、(2)の領域をクリックすることで簡単に機器を設置できる。設置した機器は図 5 に示すような画面によって、通信実験に必要な設定を行うことができる。この例では、ネットワークインタフェース `eth0` に対する IP アドレスなどネットワークの基本的設定を行っている。構築したネットワークにおいて行える通信のうち、ウェブページ取得の実行画面(図 6)と、導通確認(図 7)の実行画面を示す。図 6 では、ウェブサーバに該当ページが存在しないため、エラーページが表示されていることがわかる。図 7 では、`ping` プログラムを実行した結果、目的の IP へ到達できることを示している。以上のように本システムでは、マウス操作を中心とした GUI ベースのコントローラにより、簡単かつ自由に仮想的なネットワークを構築でき、通信を行うことができる。

本システムでは、TCP/IP の学習用情報として、これらの通信結果を基に生成した情報を表示する。図 8 ではパケットの通った経路を、パケットを表すアイコン(青四角)のアニメーション表示によって示している。このアイコンをマウスでクリックすると図 9 のように、パケットデータが表示される。図 10 ではネッ

トワーク機器間の通信概要を表示している。ネットワーク機器を TCP/IP 階層モデルによって表し、階層間において行っている通信の説明を表示している。図 11 ではルータの内部処理を表示している。TCP/IP 階層モデルのネットワーク層における IP 処理である。(1) は処理対象となるパケット、(2)(3)は IP 処理に必要となる機器内部の情報である。(1)のパケットに対し、ヘッダの整合性チェックを行い(2)、ルーティングテーブル(3)にてパケットの転送先を検索した場面である。

## 6. まとめ

本稿では、書籍や講義による TCP/IP を補助する方法として、ネットワークシミュレーション結果を可視化表示するシステムについて述べた。1 台のコンピュータ上にてネットワークを仮想的に構築し、様々な通信をシミュレーションし、その結果を基にした学習用情報をアニメーションによってわかりやすく表示する機能を持つ。

今後の課題は、学習用情報の改良と実証実験である。現在の学習用情報はネットワーク通信の動作を表示できているが、個々の動作の意味についての説明が乏しい。TCP/IP の学習支援のためには、書籍や講義で説明された用語との対応付けを考慮した説明を表示できることが望ましい。また、実証実験により学習効果を確認するとともに、ユーザインタフェースの改良などシステムの使いやすさを向上させたい。

## 7. 謝辞

本研究の一部は、文部科学省 21 世紀 COE プログラム「社会情報基盤のための音声映像の知的統合(IMI)」、科研費基盤研究 C(2)16500056 の研究助成による。

### 参考文献

- [1] 立岩佑一郎, 安田孝美, 横井茂樹: TCP/IP 学習のための可視化シミュレータの研究, 第 3 回情報科学技術フォーラム, 情報科学技術レターズ pp.355-357(2004).
- [2] 立岩佑一郎, 安田孝美, 横井茂樹: TCP/IP 学習支援システムにおける仮想的なネットワークの構築手法について, 平成 16 年度 電気関係学会東海支部連合大会, 講演論文集 O-388(2004).
- [3] 立岩佑一郎, 安田孝美, 横井茂樹: TCP/IP 学習支援のための可視化シミュレータの開発, 第 3 回情報科学技術フォーラム, 講演論文集(2005).
- [4] 荒井正之, 田村尚也, 渡辺博芳, 小木曾千秋, 武井恵雄: TCP/IP プロトコル学習ツールの開発, 情報技術レターズ,

Vol.1, pp.243-244(2002).

[5] Arai, M., Tamura, N., Watanabe, H., Ogiso, C. and Takei, S.: Design and Implementation of a Learning Tool for TCP/IP Protocols, Proc. International Conference on Computers in Education, Vol.2, pp.1010-1015(2001).

[6] 荒井正之, 田村尚也, 渡辺博芳, 小木曾千秋, 武井恵雄: TCP/IP プロトコル学習ツールの開発と評価, 情報処理学会論文誌, Vol.44, pp.3242-3251(2003).

[7] 山根健一, 矢吹道郎: TCP セッションを考慮したパケットモニタリングツール, 情報処理学会 47 回全国大会, 5E-6, pp.197-198(1994).

[8] 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 26 回全国大会, E5-4, pp367-368(2001).

[9] The Network Simulator ns2.:

<http://www.isi.edu/nsnam/ns/>

[10] 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 65 回全国大会, 2D-2, pp273-274(2003).

[11] The User-mode Linux Kernel Home Page:

<http://user-mode-linux.sourceforge.net/>

[12] The libpcap project:

<http://sourceforge.net/projects/libpcap/>

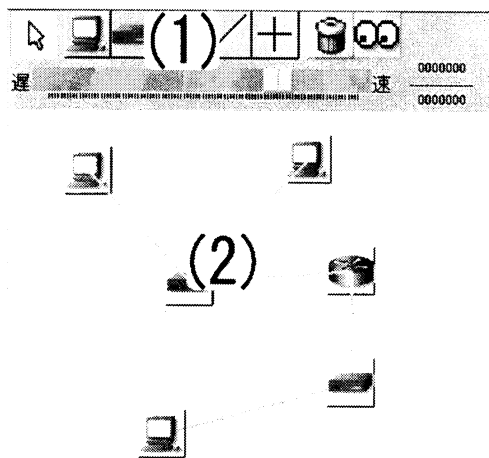


図 4 : ネットワーク構築画面

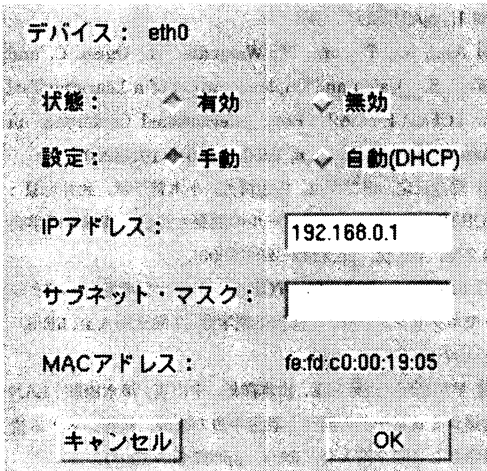


図 5 : ネットワーク設定画面

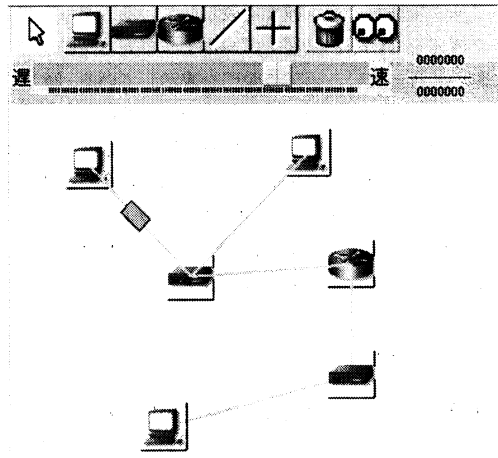


図 8 : パケット移動表示画面

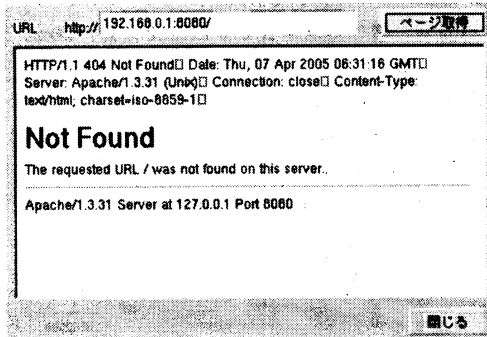


図 6 : ウェブページ閲覧通信画面

Ethernet	IP	TCP
送信データ	パケット	送信ポート
受信データ	ヘッダ	宛て先ポート
タイプフィルド	タイプ番号	シーケンス番号
	フラグ	ウィンドウサイズ
	オプション	チェックサム
	フラグメントオフセット	優先順位
	総長	タイムアウト
	ポート番号	送信バッファ
	ヘッダチェックサム	送信ウィンドウ
	宛て先IP	送信ウィンドウ
	送信IP	送信ウィンドウ

図 9 : パケット内容表示画面

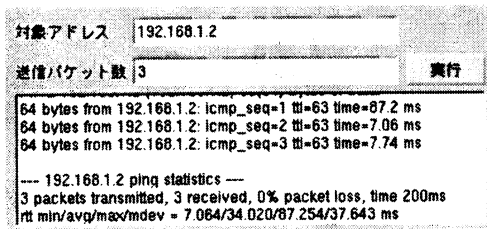


図 7 : 導通確認通信画面

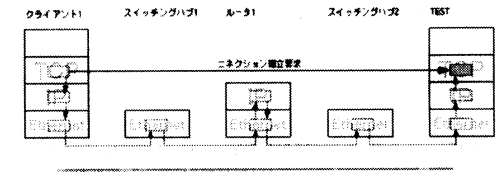


図 10 : ネットワーク機器間通信概要表示画面

A	B	IP	送信条件	送信ポート	宛先ポート	送信ウィンドウ	送信バッファ
0	0	内部	(1)	C	C	0	0
0	1	内部	(2)	C	C	0	0
192.168.0.1	192.168.0.2	内部	(3)	C	C	0	0
0	0	内部		C	C	0	0
0	1	内部		C	C	0	0
192.168.0.1	192.168.0.2	内部		C	C	0	0
192.168.0.1	192.168.0.2	内部		C	C	0	0
192.168.0.1	192.168.0.2	内部		C	C	0	0

図 11 : 機器内部処理表示画面