컴퓨터 네트워크 가상 실습을 위한 컴포넌트 기반의 시뮬레이터 설계 및 구현

임명식** 김기형**

Design and Implementation of a Component-based Simulator for the Virtual Laboratory of Computer Networks

Myungsik Lim, Kihyung Kim

Abstract

Recently, various network protocols have been developed to keep step with the rapid growth of Internet users. For the education of students in the computer networks classes, there have been many attempts to allow students experiment routers setting, operations and management of networks by themselves, in addition to the classroom lectures based on textbooks. One of the attempts is to install expensive real router experiment sets in laboratories for students, and one another is using router and network simulators for virtual experiments. This paper presents the design and implementation of NetSim, a scalable, component-based simulator environment for the network experimentation. NetSim expands the existing component-based JavaSim simulation tool for the education purpose, and it allows the design and experiment of various networks and protocols. For the evaluation of NetSim, it is shown that the network design and RIP-based router setting process is possible in NetSim.

Key Words: 가상실험실, 컴포넌트 기반 시뮬레이션, 네트워크 시뮬레이션

1. 서론

최근 들어 인터넷 사용자의 급속한 확산과 더불어 초고속 인터넷의 기술적 사회적 요구가 증대되고 있다. 이와 더불어 다양한 컴퓨터 통신 및 네트워크 기술들이 개발되고 있으며 또한 인터넷 프로토콜들도 많은 연구와 더불어 급속한 발전을 보이고 있다. 이와 같이 다양하고 복잡해지는 컴퓨터 네트워크 프로토콜들을 학생들에게 보다 쉽게, 그리고 변화에 발맞추어 교육시키기 위한 노력도 다각도로 시도되고 있다. 암기위주의 교과서교육이 아니라 실습을 통해 프로토콜의 동작원리 및 운용을 직접 익히고자 하는 시도도 네트워크 교육에서는 많이 시도되고 있다.

인터넷 프로토콜의 실습을 위한 기존의 노력 중에는 실제로 실습용 라우터세트를 구비하고 학생들에게 실제로 인터넷프로토콜들을 운용 관리시켜 봄으로써 프로토콜의 이해를 좀더 쉽게 하고자하는 시도가 있다[1]. 학생들은 직접 라우터세트 앞에서 터미널을 통해 라우터에 접속하고 CLI(Command Line Interface)를 통해 라우터를 셋팅해가면서 네트워크를 구성해가는 방법을 배우는 것이다. 그러나 이러한 라우터 세트 구입은 비용이 많이 들고 또한 제한된 실습실에 제한된 인원만 실습을 할 수 있다는 어려움이 있다. 또 다른 시도로는 네트워크 시뮬레이터를 이용하는 방법이 있다. eSim[2]은 플래쉬를 사용하여 라우터 시뮬레이터를 구축하였다. 사용자는 실제 라우터에서 작업하듯이 라우터시뮬레이터 상에서 라우터 셋팅 명령어를 입력하게 된다. 이 방법의 단점은 플래쉬를 하여 작성하였기 때문에 정해진 네트워크 토폴로지와 정해진 셋팅만을 사용해야만 한다. 만약 다양한 네트워크 토폴로지에 대해서 실험을 하려면 eSim을 사용할 수는 없다. SimDraw[3]는 원격교육을 위해 가상 실험실을 구축하였다. 클라이언트/서버구조로 되어 있으며 학생들은 씬(thin) 클라이언트 프로그램을 통하여 시뮬레이션 및 애니메이션을 할 수 있으며, 가상실험을 수행할 수 있다. VisuSIM[4]은 컴퓨터구조에 대한 웹 기반 시뮬레이션 툴로서 Java를 이용하여 CPU 시뮬레이터를 구현하여 실험용 툴로서 효과적으로 사용될 수 있음을 보였다.

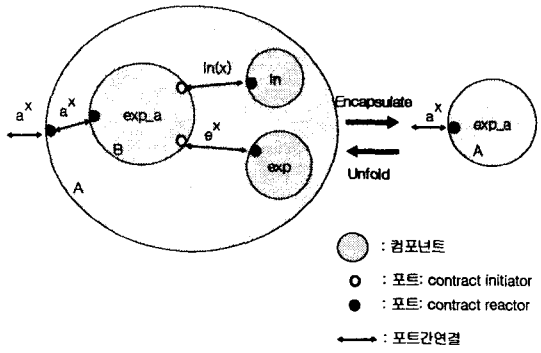
본 논문에서는 시뮬레이터를 통하여 라우터와 네트워크교육을 위한 가상 실습 환경을 제공하는 NetSim을 제안한다. NetSim은 컴포넌트 기반의 JavaSim[5]을 확장하여 설계 및 구현되었으며, 다양한 토플로지 생성, 가상 터미널을 통한 라우터 명령어입력(CLI) 및 라우터 셋팅 등의 가상 실습환경 기능을 제공한다. 또한 NetSim의 사용자 인터페이스는 모델 설계를 쉽게 하기 위해 GUI에 기반하여 시각적 모델링이 가능하도록 설계되었다.

논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 NetSim의 기본이 되는 컴포넌트 기반의 네트워크 시뮬레이터를 소개하고, 3장에서는 NetSim의 설계 및 구현에 대해 다룬다. 4장에서는 실험결과로서 NetSim을 이용하여 시뮬레이션 모델을 생성, 실행하는 방법에 대해 다룬다. 5장에서는 NetSim과 기존의 네트워크 시뮬레이터들과 비교를 한다. 끝으로 6장에서 결론을 맺는다.

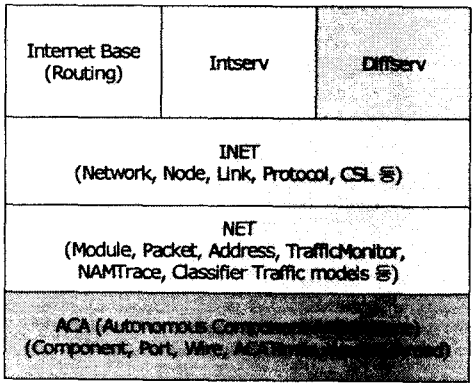
2. 컴포넌트기반의 네트워크 시뮬레이터

본 절에서는 본 논문에서 설계 및 구현할 가상 실습용 시뮬레이터인 NetSim의 기반이 되는 JavaSim[5]의 특징을 기술한다. JavaSim은 컴포넌트기반의 네트워크 시뮬레이터로서 ACA (Autonomous Component Architecture) 에 바탕을 두고 있다[6]. <그림 1>과 같이 컴포넌트간의 연결은 포트를 통해서만 이루어지므로, 컴포넌트의 캡슐화, 모듈화가 가능해진다. <그림 1>에서 exp_a 컴포넌트는 외부적으로 A 컴포넌트에 의해 완전히 캡슐화 되어 A 컴포넌트 외부에서는 정해진 포트만을 이용하여 연결하게 된다. 이와 같은 ACA의 개념은 소프트웨어 개발 방법론에

서는 일반화된 개념이지만, 네트워크 시뮬레이터 분야로 확장하였다고 볼 수 있다. 즉, 시뮬레이터가 컴포넌트로 구성되어 확장성, 재사용성의 장점을 가질 수 있게 된다.

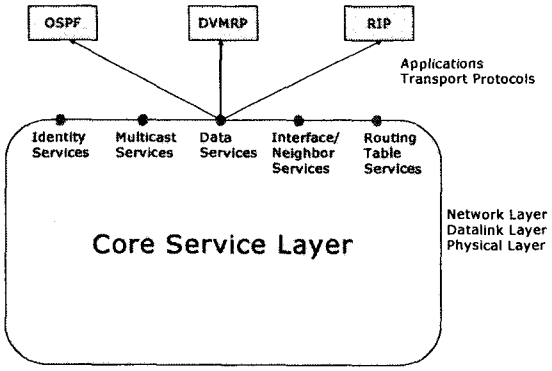


<그림 1> 컴포넌트 간 연결과 컴포넌트의 캡슐화



<그림 2> JavaSim의 컴포넌트 계층구조

JavaSim의 구조는 <그림 2>와 같이 ACA (Autonomous Component Architecture) 기본 계층과 그 상위계층인 NET계층, 그리고 INET (internetworking framework) 계층으로 구성된다. INET계층은 ACA 및 NET의 상위 계층으로서, 패킷 스위칭 기반의 네트워킹 기능을 제공하기 위한 기본 컴포넌트들 즉, 네트워크 노드 컴포넌트, NIC (네트워크 인터페이스 카드) 컴포넌트, 링크 컴포넌트, 프로토콜 컴포넌트, 네트워크 컴포넌트 등으로 구성된다. 또한 INET 계층의 컴포넌트를 확장하여 라우팅 모듈[7], Integrated Services[8] 모듈, Differentiated Services[9], OSPF [10], DVMRP[11], MPLS[12] 모듈 등의 새로운 프로토콜들을 추가할 수 있다.



<그림 3> 네트워크 노드 컴포넌트의 구조

네트워크 노드 컴포넌트는 <그림 3>과 같이 CSL(Core Service Layer)과 CSL의 서비스를 이용하는 트랜스포트 레이어 및 애플리케이션 레이어의 다양한 프로토콜 모듈로 구성되어진다. 네트워크 노드 컴포넌트는 CSL을 통하여 다른 컴포넌트의 CSL연결되며, IP패킷의 구성 및 데이터링크레이어의 모델링을 표현하게 된다.

JavaSim의 모델 구성, 연결 관리, 그리고 모니터링은 실행 시간 가상 시스템(Run Time Virtual System)과 Tcl 언어를 통해서 이루어진다. RUV는 Tcl의 확장으로서 마치 유닉스 명령어를 사용하듯이 동적으로 모델구성, 연결, 관리가 가능하도록 해준다. RUV는 쉘/터미널 컴포넌트와 출력 컴포넌트를 기반으로 동작하며, 쉘/터미널 컴포넌트는 Tcl과 RUV 명령어를 실시간으로 처리 하고, 출력 컴포넌트는 실시간으로 시스템을 모니터링 하고 출력하기 위한 것이다.

3. NetSim의 설계 및 구현

본 장에서는 컴퓨터 네트워크의 설계 및 다양한 프로토콜을 실습해 볼 수 있는 가상 실습용 시뮬레이터인 NetSim을 설계하고 구현한다.

3.1 설계 요구사항(Design Requirements)

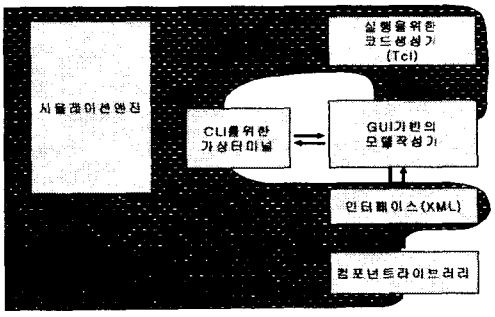
NetSim은 학생들이 시뮬레이터를 이용하여 다양한 네트워크 위상(topology)를 설계하고 라우터를 운용관리 (OAM: Operations and Management)를 할 수 있도록 해주는 것이 주요한 목표이다. 다음은 네트워크 교육용 시뮬레이터인 NetSim의 설계 시 고려해야만 했던 요구사항들이다.

1) 라우터 운용관리를 위한 CLI (Command line interface)를 제공한다.
2) 다양한 네트워크 위상을 초보자가 쉽게 설계할 수 있도록 GUI기반의 모델 저작도구를 설계한다.
3) TCP/IP기반의 다양한 프로토콜을 시뮬레이션하기 위해 확장성이 있도록 설계한다.
4) 개발된 라이브러리의 유지보수가 쉬워야 한다.

컴포넌트 기반으로 이식성과 확장성이 뛰어난 JavaSim을 이용하여 GUI 환경을 이용해 간단한 작업으로 다양한 시뮬레이션 모델을 할 수 있다. 컴포넌트 기반의 JavaSim을 확장한 NetSim의 구조와 NetSim을 이용한 모델 설계 그리고 시뮬레이션 수행 및 결과에 대해서 알아본다.

3.2 NetSim의 구조 설계

NetSim의 전체 구조는 <그림 4>와 같이 시뮬레이션 엔진, 컴포넌트 라이브러리, 인터페이스, 모델 작성기, Tcl 코드 생성기, 그리고 가상 터미널로 구성된다.



<그림 4> NetSim의 전체구조

1) 시뮬레이션 엔진

시뮬레이션 엔진은 JavaSim를 이용한 시뮬레이션 로직과 모델 작성기, 인터페이스를 제어 하는 부분이며, 실행 시에 모델 작성기로 설정된 정보를 이용하여 시뮬레이션을 수행한다.

2) 컴포넌트 라이브러리

JavaSim의 다양한 컴포넌트기반의 라이브러리를 이용하여 확장 가능한 가상 실습환경을 구축할 수 있다.

3) 인터페이스

XML언어로 기술되며, 자바 코드로 구현된 컴포넌트를 시각적으로 보여 주기 위해서 XML로 기술된 컴포넌트 정보를 읽어서 GUI 형태로 볼 수 있게 변환된다. 기존의 설계된 파일을 읽어오거나, 모델 작성기를 이용해 새로운 모델을 만들 때 해당 컴포넌트의 정보를 해석하여 GUI화면으로 나타낸다.

4) 모델 작성기

시각적 모델링이 가능하도록 설계되었으며, 네트워크의 토폴로지, 컴포넌트 속성, 프로토콜 정보 등을 쉽게 편집할 수 있다.

5) 코드 생성기

모델 작성기로 작성한 시뮬레이션정보를 이용하여 Tcl코드를 생성한다.

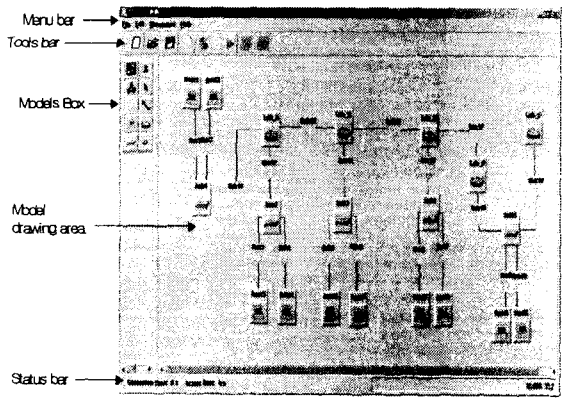
6) 가상 터미널(Virtual Terminal)

가상터미널은 라우터 명령어(CLI)를 입력하기 위한 화면이다. 라우터 컴포넌트마다 하나의 가상 터미널이 연결되며 이 터미널을 통

하여 라우터 명령어를 입력하고 라우터를 셋팅하게 된다.

3.3 NetSim 모델 작성기의 화면구성

NetSim 모델 작성기의 화면 구성은 <그림 5>와 같다. 모델 박스(Model box)는 각 네트워크 컴포넌트들의 아이콘을 모아놓은 툴박스이다. 예를 들어 사용자가 모델 박스에서 라우터 아이콘을 선택하고 모델 작성 영역(Model drawing area)의 임의의 지역에 마우스를 클릭하면, 그곳에 라우터 컴포넌트가 생성된다. 또한 라우터 컴포넌트 두개를 연결하기 위해 포트를 생성하고 링크를 생성하게 된다.



<그림 5> NetSim의 화면 구성

3.4 NetSim의 구현

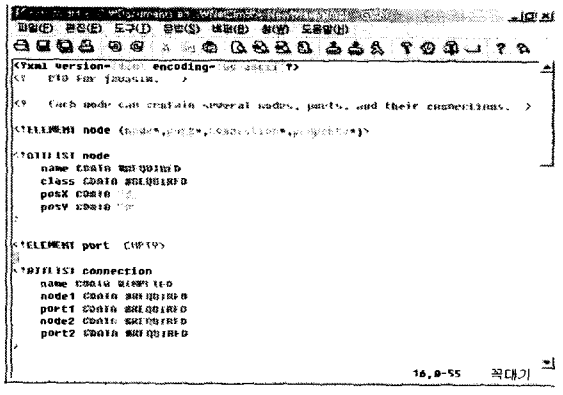
NetSim의 구현은 기본적으로 JavaSim의 확장을 통해 이루어졌다. JavaSim은 컴포넌트기반의 네트워크 시뮬레이터이므로 확장성이 매우 뛰어나고, 따라서 자연스럽게 JavaSim에서 개발된 컴포넌트들을 확장함으로써 비교적 빠르게 JavaSim을 구현할 수 있다. <그림 4>에서 빗금 친 영역에 속하는 컴포넌트들은 JavaSim이 가지고 있던 컴포넌트를 활용하여 구현된 것들이다. 일부만 빗금 친 영역에 속하는 컴포넌트들(즉, 모델작성기, 네트워크 컴포넌트 라이브러리, 그리고 가상터미널 컴포넌트들)은 JavaSim 컴포넌트를 확장하여 본 연구에서 새롭게 구현하였음을 의미한다. 다음은 새롭게 구현된 기능을 설명한다.

1) 컴포넌트 라이브러리 추가

GUI환경의 시뮬레이션 수행을 위해 NetSim에서 호스트 컴포넌트, 라우터 컴포넌트, 허브 컴포넌트, ping 컴포넌트, 명령 컴포넌트를 추가하였다. 호스트 컴포넌트는 모델 작성기에서 마우스를 이용하여 각 호스트의 호스트 네임, 인터넷 주소, 서브넷 마스크 주소를 변경할 수 있도록 제작되었다. Router 컴포넌트는 모델 작성기에서 라우터의 콘솔 화면을 통해 라우터 명령어의 입력, 처리기능을 제공한다. 핑 컴포넌트는 구성된 네트워크가 잘 동작하는지 확인하기 위해 호스트와 호스트 간 연결을 확인하는 기능을 한다.

2) 인터페이스와 코드 발생기

NetSim은 모델의 맵핑 및 변환을 위한 모델 기술 언어로서 XML DTD 와 XML파서를 이용하여 XML 파일을 생성한다. 따라서 모델작성기의 저장 파일 형식은 XML형식이며 모델작성기는 XML파서를 통해 모델 컴포넌트의 정보를 해성하여 GUI화면에 표시하게 된다. <그림 6>은 모델 정보를 위한 DTD 파일을 나타낸다.



<그림 6> XML 코드 생성의 위한 DTD 파일

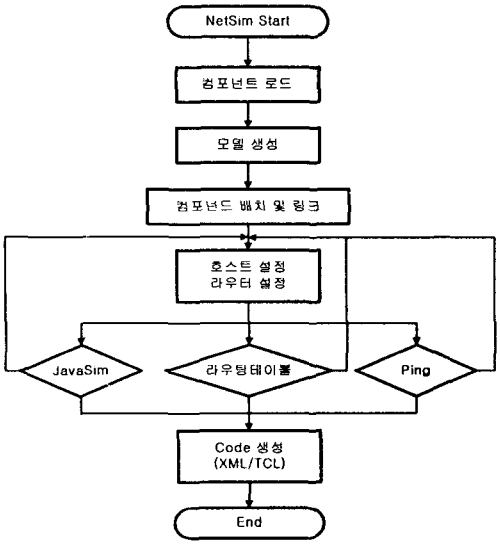
3) 가상 터미널

라우터시뮬레이터의 가상터미널은 기존 JavaSim의 RUV(Run Time Virtual System)를 확장하여 구현되었다. 각 라우터마다 가상터미널이 팝업되고 사용자는 가상터미널을 통하여 라우터의 여러 속성들을 변경시킬 수 있다. 가상터미널은 라우터 명령어(CLI)를 입력하기 위한 화면이다. 즉, 라우터에 하이퍼터미널과 같은 환경을 제공하여 이 터미널을 이용하여 라우터 명령어를 입력하여 라우터를 셋팅하게 된다. 터미널 화면에서 라우터 명령어를 입력받으면 가상터미널 컴포넌트는 해당 명령어를 처리하여 라우터를 설정하게 된다.

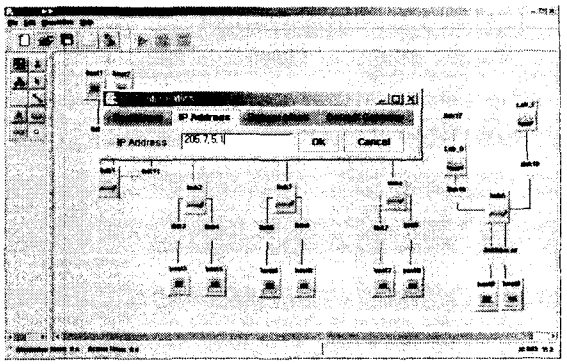
4. 실험 결과

4.1 NetSim을 이용한 가상실습의 수행

<그림 7>은 NetSim을 이용하여 네트워크 모델의 설계 및 시뮬레이션을 하는 전체 과정이다. 네트워크 모델 설계 하는 과정은 다음과 같은 단계로 실행되어 진다.



<그림 7> 네트워크 모델 설계 절차



<그림 8> 호스트 설정

1) 모델 생성

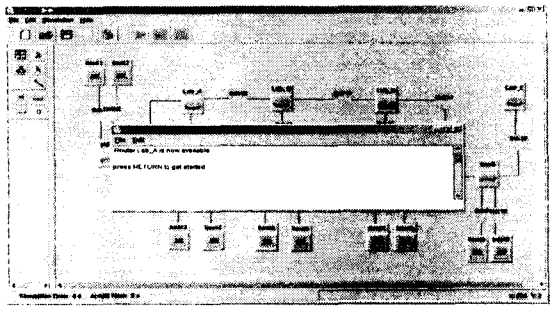
사용자는 NetSim 모델작성기를 통해서 새롭게 네트워크 모델을 작성하거나 기존에 작성된 모델정보 파일로부터 읽어서 모델을 구성하게 된다. 모델작성기 내에서 컴포넌트들은 적절한 위치에 배치되고 또한, 포트를 통해 서로 연결되는데, 이 때 컴포넌트의 좌표와 연결정보는 컴포넌트 리스트(벡터)에 저장된다.

2) 호스트 설정

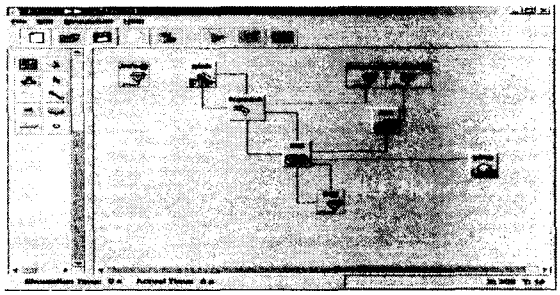
네트워크 컴포넌트들의 물리적 연결이 끝나면, 각 호스트(커넥션의 끝점)마다 팝업메뉴를 이용하여 호스트이름, IP, 서브넷 마스크 등을 설정한다. <그림 8>은 호스트의 설정화면을 보여준다.

3) 라우터 설정

라우터의 설정을 위해서는 라우터 명령어를 이용하여 설정하게 되는데, <그림 9>와 같이 라우터의 콘솔 화면이 팝업되며, 이 콘솔창을 통해 라우터 명령어를 입력하여 라우터를 설정하게 된다. <그림 10>은 라우터 컴포넌트의 구성 요소로서, 설정된 라우터의 정보를 포함하게 된다.



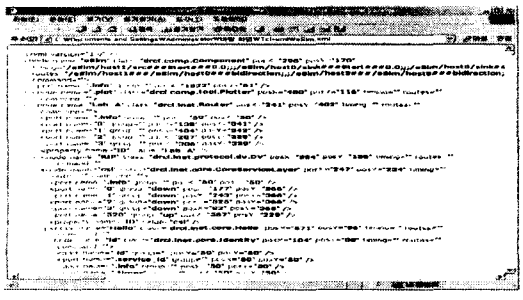
<그림 9> 라우터 설정을 위한 콘솔 화면



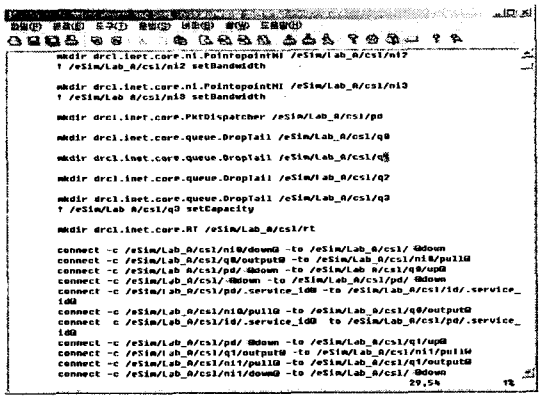
<그림 10> 라우터 컴포넌트의 구성요소

4) 코드 저장

모델 작성기를 통해 생성된 모델 토폴로지와 라우터 및 호스트 셋팅 정보는 두 가지 방식으로 저장이 되는데 첫 번째 방법은 XML 파서를 통해 XML 트리 구조로 표현되어 저장하는 것이고, 두 번째 방법은 JavaSim에서 실행 가능한 포맷인 Tcl 포맷으로 저장하는 것이다. <그림 11>과 <그림 12>는 각각 XML코드와 Tcl코드를 나타낸 것이다.



<그림 11> XML 코드 생성



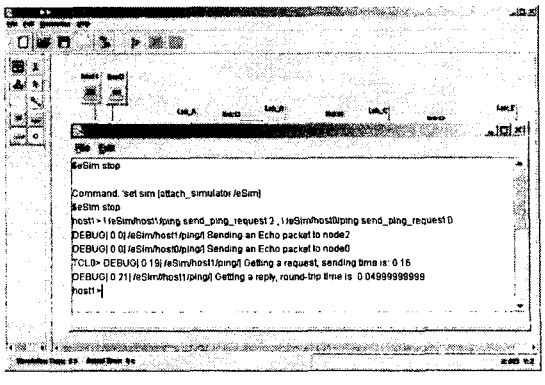
<그림 12> Tcl 코드 생성

4.2 실습 결과의 분석

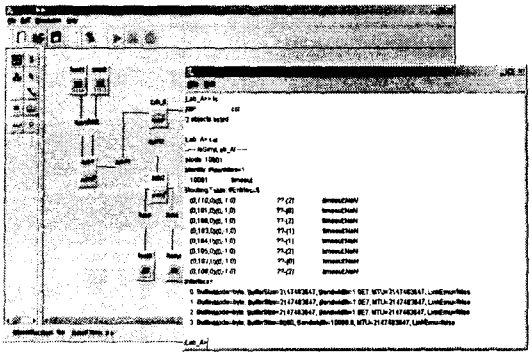
라우터와 호스트 설정이 끝나면 시뮬레이션을 수행하게 되고, 모델의 구성이 잘 되었는지를 확인하게 된다. 실습결과의 분석은 다음과 같이 세 가지의 과정을 통해 이루어진다.

1) ping 명령어 실행

<그림 13>과 같이 ping을 이용하여 호스트 및 라우터의 설정이 제대로 이루어지고 통신이 잘고 있는지 확인한다. ping 명령어는 간단하면서도 실제 인터넷이나, 네트워크에서 가장 많이 사용하는 명령어이다. 네트워크문제 발생시 에러 체크를 위해서 사용된다.



<그림 13> ping을 이용한 테스트



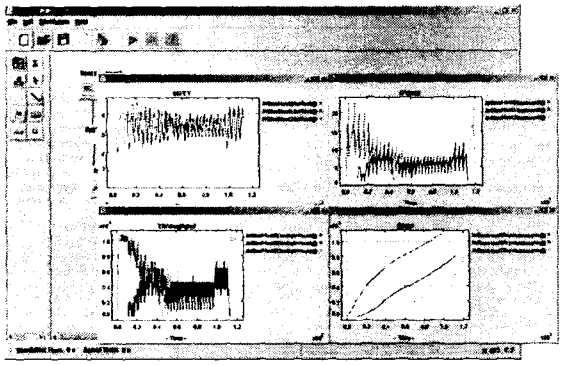
<그림 14> 라우팅 테이블 이용한 테스트

2) 라우팅 테이블을 확인

<그림 14>와 같이 라우팅 테이블을 확인하여 라우팅 프로토콜이 제대로 동작하는지 확인 할 수 있다. 본 논문에서는 RIP[13]를 이용하여 라우터를 구성하였다.

3) 실험결과 분석

<그림 15>와 같이 각 호스트에서 TCP/UDP 프로토콜을 사용하여, 통신 성능을 분석 할 수 있다.



<그림 15> JavaSim을 이용한 실험결과 분석

5. NetSim과 기존의 네트워크시뮬레이터들과의 비교 분석

컴퓨터 네트워크의 시뮬레이션을 수행하는 다양한 시뮬레이터들을 사용 목적으로 분류해 보면 크게 교육용과 연구용으로 나눌 수 있다. 교육용 툴은 학생들에게 다양한 프로토콜 및 네트워크 설계를 시뮬레이터 상에서 수행해볼 수 있도록 설계된 것이고, 연구용 툴은 새로운 프로토콜들을 설계하고 성능평가해보기 위해 만들어진 툴들이다.

eSim[2]은 컴퓨터 네트워크 가상실습을 위해 만들어진 교육용 시뮬레이터이다. 플래쉬(flash)로 제작되었으며 웹 기반으로 라우터 운용 관리 및 네트워크 설계를 해볼 수 있도록 설계되어 있다. 하지만 플래쉬로 제작되었기 때문에 특정한 네트워크 토폴로지만을 대상으로 가상실습을 할 수 있고 다양한 네트워크 토폴로지에의 적용 및 새로운 프로토콜에 적용하기에는 기본적인 한계를 가지고 있다.

SimDraw[3]는 웹 기반 클라이언트/서버 환경을 기반으로 원격 사이버교육에서 컴퓨터 네트워크 가상실습을 할 수 있는 교육용 시뮬레이터이다. 원격지의 학생들은 웹 브라우저 상의 클라이언트 환경에서 GUI를 이용하여 컴퓨터 네트워크 시뮬레이션 코드 생성하고 이를 서버 측으로 전달하여 서버에서 라이브러리와의 컴파일을 하게 된다. 컴파일된 코드는 다시 클라이언트로 전달되어 클라이언트 환경에서 시뮬레이션수행 및 애니메이션을 하게 된다.

연구용으로 주로 사용되어지는 컴퓨터 네트워크 시뮬레이터로서는 JavaSim[5], Opnet[14], NS2[15], GloMoSim[16] 등이 있다. NS2와 GloMoSim은 연구용으로 많이 사용되는 시뮬레이션 툴이지만, 초보자가 사용하기에는 어렵다는 단점이 있다. Opnet은 상업용 툴로서 학생들이 쉽게 접근하기 어려운 단점이 있다. JavaSim은 컴포넌트 기반으로 시

뮬레이터가 구현되어 졌기 때문에, 확장성이 뛰어나며, 자바 언어로 구현 되어 이식성이 뛰어나다. <표 1>은 기존의 시뮬레이터들과 NetSim을 용도와 특징 측면에서 비교 한 것이다.

<표 1> 기존 시뮬레이터들과 NetSim의 비교

시뮬레이터	개발언어	용도	특징	단점
eSim	플래쉬	교육용	사용이 간편	확장성 없음
Opnet	C++	상업/연구용	다양한 라이브러리 상업용	높은 가격
NS2	C++, OTcl	연구용	많은 사용자 다양한 라이브러리	초보자의 사용이 어려움
SimJava	Java	연구용	애니메이션지원	라이브러리 부족
SimDraw	Java	가상실험용	클라이언트/서버구조 가상실험에 적합 애니메이션지원	라이브러리 부족
JavaSim	Java, Tcl, XML	연구용	컴포넌트기반	사용이 어려움
NetSim	Java Tcl XML	교육용	JavaSim의 확장 라우터의 CLI지원	애니메이션 지원 안 됨

6. 결론

컴퓨터 네트워크의 교육에서 기존의 암기 위주의 교과서 교육이 아니고 실습을 통해서 프로토콜의 동작원리 및 운용을 직접 익히고자 하는 시도가 계속 이루어지고 있다. 실습용 라우터 세트가 따로 실험실에 구성되어 학생들이 직접 라우터 셋팅과 운용 및 네트워크 설계를 해볼 수 있도록 하는 시도도 있고, 실제 라우터와 유사한 시뮬레이터를 이용하여 가상 라우터실험을 해볼 수 있도록 하는 노력도 있다. 본 논문에서는 확장가능하고 다양한 네트워크 실습이 가능한 시뮬레이터환경 NetSim을 설계 및 구현하였다. NetSim은 기존의 컴포넌트 기반으로 구현된 JavaSim을 확장하여 라우터 터미널(CLI)을 통한 라우터 셋팅, 다양한 네트워크 토폴로지구성, 확장성 있는 프로토콜 실습 등이 가능하도록 설계되었다. NetSim의 효용성을 보이기 위해 NetSim을 이용하여 네트워크 설계 및 RIP를 이용한 라우터 셋팅 등의 실습 과정이 가능함을 보였다.

참고 문헌

[1] Cisco Networking Academy, URL: http://cisco.netacad.net/public/index.html.

[2] eSim, URL: http://cisco.netacad.net/public/index.html.

[3] Hyungon Seo, Bong Sagong, and Kihyung Kim, "Web-based Modeling, Simulation and Animation of Routing Protocols", Proceedings of IASTED Conference on Internet and Multimedia Systems Applications, pp313-382, 2000

[4] Yoshiro Imai, Shinji Tomita, "Design and Implementation of Web-based Education Tool", Proceedings of the 2002 Symposium on Applications and the Internet(SAINT'02w), IEEE, 2002.

[5] H. Tyan, "Design, Realization and Evaluation of a Component-based Compositional Software Architecture for Network Simulation", Ph. D. Dissertation, Ohio State University, 2002.

[6] H. Tyan, "The Autonomous Component Architecture(ACA)", URL: http://www.javasim.org/whitepapers/aca.html.

[7] J. Doyle, "CCIE Professional Development : Routing TCP/IP", Volume 1, CISCO Press, 1998.

[8] S. Shenker, J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements" IETF RFC 2215, Sep. 1997.

[9] M. Carlson, W. Weiss, S. Blake, Z. Wang, D. Black, E. Dabies, "An Architecture for Differentiated services", IETF RFC2475, Dec. 1998.
[10] J. Moy, "OSPF Version 2, IETF RFC2328, Apr. 1998.
[11] D. Waitzman and C. Partridge and S. Deering, "Distance Vector Multicast Routing Protocol", IETF RFC1075, Aug 1996.
[12] E. Rosen, A. Viswanathan and R. Callon, "Multi Protocol Label Switching Architecture", IETF RFC3031, Jan. 2001.
[13] G. Malkin, "RIP Version 2", IETF RFC2453, Nov. 1998.
[14] Opnet modeler, URL: http://www.opnet.com.
[15] K. Fall and K. Vardhan, "NS notes and documentation", URL:http://www.isi.edu/nsname/ns.
[16] X. Zeng, R. Bagrodia and M. Gerla, "GloMoSim: a Library for the Parallel Simulation of Large-scale Wireless Networks", pp. 154--161, Proceedings of PADS, 1998.

● 저자소개 ●

임명식

2000 영남대학교 전자공학과 학사
2002 영남대학교 컴퓨터공학과 석사
관심분야: 컴퓨터네트워크, 컴퓨터 시뮬레이션

김기형

1990 한양대학교 전자통신공학과 학사
1992 한국과학기술원 전자공학과 석사
1996 한국과학기술원 전자공학과 박사
2001 AdForce, Inc. Senior Engineer
1997~현재 영남대학교 전자정보공학부 조교수
관심분야: Ad Hoc Networks, Multicasting, Simulation, Embedded Systems