에뮬레이션형 테스트베드 활용기술 지원

자: 2014년 11월 25일

부　서: 첨단연구망센터/첨단연구망개발팀

제출자: 석 우 진

KiSTi www.kisti.re.kr
한국과학기술정보연구원
Korea Institute of Science and Technology Information
305-806 대전광역시 유성구 어은동 52번지
TEL (042)869-0676 / FAX (042)869-0679
www.kisti.re.kr

에뮬레이션형 테스트베드 활용 지원

작성자: 석우진

첨단연구망개발팀, 한국과학기술정보연구원

wjseok@kisti.re.kr

최종수정일: 2014년 11월 25일

Abstract

미래의 네트워크는 보다 혁신적이고 진보적이어서, 이러한 네트워크를 연구하기 위하여 고도화된 테스트베드의 필요성이 제기되고 있다. 본 문서에서는 이러한 필요를 충족하는 사용자 기반의 네트워크 테스트베드를 살펴보고, 네트워크를 기반으로 하는 보안 연구 및 네트워크 프로토콜 분야를 중심으로 분산/병렬 분야의 활용에 대해서 살펴보고자 한다.

Topics

Ⅰ. 서 론

미래의 인터넷은 현재의 인터넷보다 훨씬 더 복잡하고 다양하게 발전하게 될 것이다. 인터넷의 기술은 더 정교하고 복잡한 프로토콜로 진화할 것이며 다양한 단말을 수용하여야 할 것이다. 그래서 네트워크 기술을 연구하고 개발하는 것이 점점 더 어려워질 수밖에 없고, 또한 비용도 점점 더 증가할 수밖에 없다. 예를 들면, 디도스(DDoS) 공격을 방어하는 알고리즘을 개발하기 위해서는, 수백 혹은 수천대의 PC 환경을 구축하는 충분한 비용을 필요로 하게 될 것이다.

이러한 행위가 하나의 대규모 테스트베드에서 네트워킹 관련 연구 및 개발을 목적으로 한군데서 이루어진다면, 각각의 연구 및 개발자들은 이제까지 경험하지 못했던 규모의 개발환경 및 검증을 할 수 있게 될 것이다. 특히, 네트워크와 분산 시스템 연구에 있어서 유효 적절한 검증도구는 매우 중요하다. 필요에 따라 다양한 검증도구가 사용될 수 있는데, 검증도구가 얼마나 사용하기 쉬운가에 대한 사용성, 검증도구를 검증 전 또는 검증 중에 얼마나 제어가 가능한가에 대한 제어성, 마지막으로 검증도구가 얼마나 실제 상황과 일치하는가에 대한 실제성에 따라 적절한 검증도구를 선택하게 된다. NS-2이나 OPNET와 같은 시뮬레이션 기반 검증도구는 사용성과 제어성이 높아, 재실험 및 제어가 언제든지 필요한 연구에서 주로 사용한다. 하지만 시스템에서 운영되는 운영체제의 동작 및 인터럽트 등으로 인한 오버헤드 등 시스템 내부의 역학 관계가 전혀 반영되지 않으므로 실제성 부분에 근본적인 문제점을 내포하고 있다. 한편 실제 시스템 및 네트워크를 사용하는 방법은 실제성은 높은 반면에 사용성 및 제어성이 매우 낮아 검증에 필요한 환경 구성 및 재실험 등이 어렵다. 이런 문제를 해결할 수 있는 검증방법론이 에뮬레이션 방법론이라고 할 수 있다.

에뮬레이션 방법을 통하여 테스트가 진행될 수 있는 에뮬레이션형 테스트베드로 Emulab이 대표적이며 또한 여러 가지 변형에 대한 구축도 연구 중이다. 실제 PC, 운영체제, 및 응용 어플리케이션, 실제 네트워크를 사용하여 실제성을 높이고 동시에 언제든지 제어가 가능한 프레임워크를 구축하여 제어성을 확보하는 한편, 원격 사용자가 손쉽게 시스템 및 네트워크 환경을 구축하고 이용할 수 있는 인터페이스를 제공함으로써 사용성을 높이고자 하였다. 본 문서에서는 이러한 에뮬레이션형 테스트베드를 살펴보고 주요 활용을 분석하고자 한다.

II. 에뮬레이션형 테스트베드

Emulab은 연구자에게 시스템을 개발, 디버그와 평가를 할 수 있는 광범위한 환경을 제공하는 에뮬레이션형 테스트베드이다. Emulab이란 이름은 하드웨어, 소프트웨어 시스템 모두를 뜻하는 것으로서 네트워킹 분야와 분산 시스템 분야 등 다양한 컴퓨터 과학 연구자들에 의해 사용될 뿐만 아니라 교육부분에서도 널리 활용된다. 이는 유타 대학의 전산학과 내 Flux 연구 그룹에서 운영되고 있는 것이 대표적이나 전 세계의 다양한 그룹에서 수백 개의 노드를 운영 및 사용하는 것으로 정책에 따라서 사용권한을 요청한 연구자들에게 무료로 제공된다. 실제 여러 연구 분야에서 프로토콜의 검증 및 성능평가 등을 위해서 Emulab을 사용하고 있다. 이는 실제 시스템을 운영하여 결과를 도출하기 때문에 실제 테스트와 같은 성과를 보일 뿐만 아니라 전 세계의 다양한 그룹에서 수백 개의 노드를 운영하여 사용할 수 있기 때문에 확장성이 좋다.

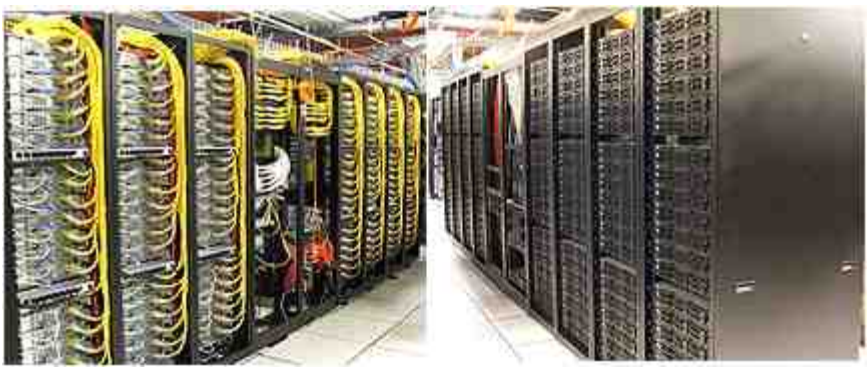

그림 1. EMULAB

DETERLab는 차세대 사이버 보안과 관련된 연구와 개발을 지원하는 다목적 실험에 의거하는 테스트베드이다. DETERLab은 악의적인 코드를 이용한 실험을 포함하는 네트워크 보안 프로젝트 분야에서 예상되는 각종 중간 규모의 인터넷 에뮬레이션 실험을 허락한다. DETERLab은 Utah 대학에 의해 개발된 Emulab 테스트베드를 기반으로 한다. 그러므로 각각의 실험에서 발생하는 요구사항을 충족한다. 임의의 토폴로지에서의 고속 링크와 함께 상호 연결 및 로드, 원격 모니터링 등을 할당할 수 있는 실험 노드인 PC의 Pool을 제어할 수도 있다. 사용자는 자신의 프로젝트와 관련된 실험을 원격으로 정의하거나 로드, 제어, 모니터링하기 위해

DETERLab의 웹 인터페이스를 사용한다. DETERLab은 DDos, 웜 바이러스 그리고 다른 악성코드, 경로 지정 간 외부 공격으로부터 방어를 하는 다양한 컴퓨터 보안 실험의 효과적인 결과를 제공하기 위해 Emulab에서 확장되었다. 사용자는 DETERLab에 등록하게 되면 프로젝트와 관련된 개발, 구성 및 구성한 네트워크 토폴로지에서 노드와 링크로부터 수집된 정보를 제어하도록 원격으로 접근할 수 있다. Testbed 노드의 Pool은 일반적으로 서로 각각 분리되어 있지만 동시에 발생하는 여러 개의 실험 간에 공유된다. 노드 Pool은 현재 USC ISI 및 UC Berkely에 위치하는 약 400여개의 PC를 포함하고 있지만 하나의 테스트베드로서 각종 관련 작업을 수행한다. 지원 OS로는 Linux, FreeBSD, Windows와 Click Modular Router를 포함한다.

PRObE는 대규모 시스템 연구를 위한 NSF가 후원하는 프로젝트로 규모가 큰 컴퓨팅 리소스의 hands-on 오퍼레이션을 수행할 수 있는 에뮬레이션형 테스트베드로써 Emulab을 기반으로 하고 있다. 높은 재구성, 원격 접근 및 제어 환경을 제공하며 연구자들은 작은 규모에서는 불가능한 실험을 수행하기 위해 사용할 수 있다. PRObE는 전 세계에서 유일한 대규모, 저수준(low-level), 높은 장비활용의 연구시설물을 커뮤니티에 제공하는 것을 목적으로 하고 있으며, 로스알라모스국립연구소(NANL)의 퇴역된 슈퍼컴퓨터를 재사용하여, 연구자들에게 제공할 수 있도록 구성함으로써 수행하게 된다. PRObE에서는 대규모 스케일의 실험이 주로 이루어지고 있다. 예를 들면, “ExAlt: Robust Exa-byte Storage”과 같은 것은, 임의 오류에 대한 EXA-bytes까지의 확장과 강력한 스토리지 시스템을 구축하는 것으로 이러한 규모와 시스템은 내구성 및 예약 가능 여부 10~20 디스크, 지리적 복제된 데이터 센터와 머신의 수만큼 구성할 수 있다. 따라서 본 프로젝트는 디스크의 손상, 메모리의 손상, 또는 CPU오류와 같은 임의의 고장에 대하여 시스템에 End-to-End 정확성 보증을 제공하기 위한 솔루션 연구를 진행하고 있다.

일본의 정보 통신 기술 국립 연구소(NICT)는 대규모의 복합 네트워크 시스템의 성능을 구현하고, 유비쿼터스 시스템의 신뢰성과 차세대 ICT 시스템의 처리를 위해 대규모 에뮬레이션형 테스트베드인 StarBED의 건설과 운영을 2002년부터 시작하였다. StarBED은 세 가지 종류의 주요 그룹으로 구성되어 있으며, 첫 번째 그룹으로는 실험 노드 그룹으로 1,100대의

PC 서버로 구성되어 있고, 이 그룹은 모델과 연도별로 7개의 그룹으로 구분되어 있음. 두 번째 그룹은 관리용 네트워크 스위치 그룹으로 실험 노드를 제어하는데 사용되며, 세 번째 그룹은 실험용 네트워크 스위치 그룹으로 최대 속도 200Gbps까지 네트워크 백본을 통해 실험 노드 사이의 양방향 통신을 제공한다.

그림 2. StarBED Servers and Network Switches in NICT

StarBED는 대규모의 실험노드를 사용하여, 일반적 개발 사이트 에서 이룰 수 없는 실험규모를 지원하는 비분산 환경을 제공하고 일반적 실행 및 운영 조건 내에서 실제 장비를 이용하여 실험을 지원한다.

III. 활용분석

에뮬레이션형 테스트베드에서 주요 활용 분야는 아래 표와 같다. 본 문서에서는 보안 분야, 네트워크 프로토콜 분야, 분산/병렬 분야, 교육 분야에서의 활용 사례를 살펴보고자 한다. 이러한 활용 사례는 국가적 테스트베드 구축 및 연구개발 분야에 적용에 있어서 좋은 사례가 될 것이다.

표 1. 에뮬레이션형 테스트베드 활용 사례

분 야	주요 내용	비 고
미래인터넷	▪ 가상화 네트워킹 실험실은 가상화 기술 및 미래인터넷이 범용의 새로운 네트워크 구조로 발전할 가능성에 대한 실제적 연구 개발 및 검증을 위한 테스트베드로 사용	▪ GENI/미국 ▪ Starbed/일본
정보 보안	▪ 대학, 보안업체, 정부연구기관과 함께 네트워크 인프라 마비공격에 대한 감시 및 추적기술을 연구하는 프로젝트로서 네트워크 실험실(Emulab)을 기반으로 하여 사이버 공격 및 방어에 대한 연구를 수행	▪ DETER/미국
교 육 및 과학기술	▪ 대학/대학원 강의 및 연구수행이 네트워크 실험실(Emulab) 기반 프로젝트 446개 진행 중, Emulab을 활용한 논문 실적 269건 등재 15개 대학 21개 클래스에서 활용 중(2009년 기준) ▪ 유선/무선/센서 네트워크, 분산컴퓨팅, 보안, 클라우드컴퓨팅 등 대부분 IT 분야에 대해서 폭넓게 활용	▪ 유타대학/미국 ▪ TWISC/대만 ▪ Britsh Columbia 대학/영국 등 다수
대규모 프로젝트	▪ 클라우드 컴퓨팅/고성능 컴퓨팅/대용량 데이터 기반의 프로젝트인 PRObE 프로젝트에서 Emulab을 플랫폼으로 활용 - 대규모/대용량의 시스템 (1,000대의 클러스터 이상)을 제공하고 있으며, 미국 NSF에서 10억 달러 규모의 예산이 지원됨 ▪ 미래인터넷의 대표적인 프로젝트인 ProtoGENI 프로젝트에서미래인터넷을 위한 네트워크 인프라로 활용	▪ 로스알라모스 연구소/미국 ▪ GENI Project Office/미국

1. 보안 분야

Emulab에서 가장 많이 활용된 사이버 보안기술 연구 분야 중에 하나는 보안 분야이며 그중에서도 DDoS 연구 분야다. 수십에서 수백대의 테스트 노드와 네트워크를 원하는 대로 쉽게 구성할 수 있을 뿐 아니라, 다양한 DDoS 공격 툴을 실제 시스템에서 작동시킬 수 있으므로 DDoS 연구에 장점이 있다. Yu Chen 등은 다수의 네트워크 도메인에서 동시에 일어나는 DDoS 공격을 플로우 레벨에서 효과적으로 탐지하는 협업 탐지 방안을 제시하였다[7]. 이를 검증하기 위해 USC ISI Emulab에서 실험을 진행하였고, 4개의 AS 도메인 상의 32개 라우터를 Emulab에서 테스트를 진행하였다. Maitreya Natu 등은 클라이언트 평판도 기반으로 DDoS를 효과적으로 탐지하고 차단하는 방안을 제시하였다[8]. 이를 검증하기 위해 Emulab을 사용하였고 하나의 피해자 노드 (V), 두 개의 정상 노드 (L1, L2), 그리고 7개의 공격 노드 (A1~A7)을 아래와 같이 구성하여 테스트 하였다.

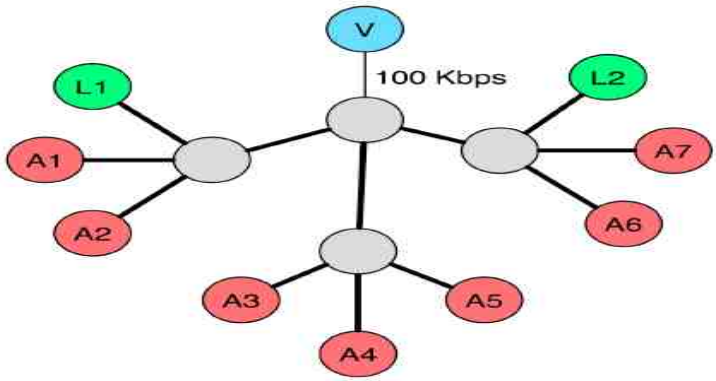

그림 3. 클라이언트 평판도 기반 DDoS 탐지 연구를 위한 Emulab 실험 환경[8]

Emulab이 왕성히 활용되고 있는 또 다른 분야는 웜 및 봇넷 관련 연구다. DDoS와 비슷하게 Emulab에서 실제 시스템과 네트워크에서 웜, 봇넷 등의 악성코드를 안전하게 테스트 해 볼 수 있어서 Emulab이 많이 활용되고 있다. Cliffork Neuman 등은 DeterLab에서 자가유포를 하는 악성코드를 테스트를 하는 방법을 설명하고 비슷한 테스트를 위한 유의사항을 제시했다[9]. Steve Hanna 등은 전파속도가 매우 빠른 하나의 flash

worm을 구현하고 그 웜이 얼마나 빨리 전파될 수 있는지를 Emulab에 테스트 했으며 사용한 토폴로지는 위의 그림과 같다[10]. DDOS 와 웜 및 봇넷 연구 분야 외에도 BGP 보안, IDS 및 방화벽 분야 등에서도 이러한 에뮬레이션형 테스트베드의 활용이 가능하다.

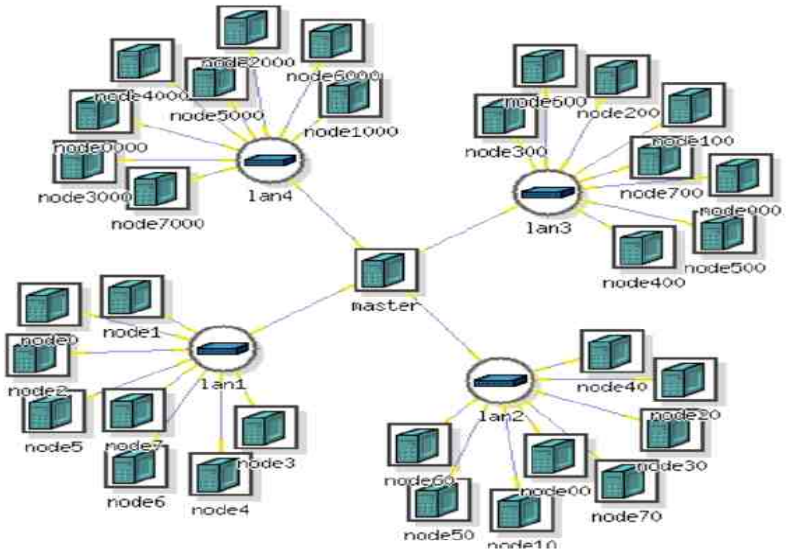

그림 4. Flash-worm 구현을 위한 Emulab 실험 환경[10]

특히 이중에서, Slowloris를 이용하여 DDoS 공격을 재연하고 이로 인한 웹서버 반응 성능이 얼마나 저하되는지, 또한 웹이 제 기능을 상실하는지를 보여주는 것을 살펴보고자 한다. Slowloris 공격원리는 다음과 같다.

1) TCP 세션을 맺은 뒤 GET 요청을 보낼 때 http 헤더가 완전하지 않은 패킷을 보냄
2) 일반적인 http request 메시지는 일반적으로 헤더는 두번의 CR + LF로 마무리
3) Slowloris 공격은 한번의 CR+LF로 끝을 냄
4) Apache 서버는 불완전한 http GET 메시지를 받으면 새로운 연결을 처리하기 위해 thread를 생성하고 바로 response를 보내지 않고 약 300초간 기다림
5) 이 timer가 만료되기 전에 불완전한 http GET 메시지를 다시 보내면 이전에 생성된 thread 계속 유지
6) 서버 종류별로 최대로 수용할 수 있는 thread 도는 연결의 개수 넘으면 웹서비스 다운

7) 이전 공격들에 비해 상당히 적은 개수의 패킷으로도 효과적인 DoS 공격 가능

이의 테스트 환경 구축을 위하여 아래같이 구성하였으며, 이는 에뮬레이션형 테스트베드를 통하여 이루어진다.

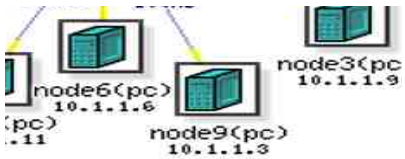

그림 5. 테스트 토폴로지 구성

구성된 토폴로지를 바탕으로, ① 노드 10개에 Slowloris 공격도구를 설치하고 노드 1개를 공격할 예정이며, ② 추가 1개 노드에서 공격대상 노드로 웹서버 연결 시도 및 시간 측정한다. 또한 해당 노드는 아래와 같은 패키지가 설치되며, 설치된 이후에 wget 을 통하여 시간을 측정하여 공격대상 서버의 성능을 측정할 수 있다.

(1) (공격수행@node1-10) Net-SSLeay 및 IO-Socket-SSL 설치
#yum install perl-Net-SSLeay
#yum install perl-IO-Socket-SSL.noarch (위의것 설치이후)
Slowloris.pl 스크립트 파일 다운로드 (/usr/local/src/test 에 복사)

(2) (공격대상용@node0, 웹서버) 웹서버 설치
yum install httpd
chkconfig httpd on (공격대상용@node0, 웹서버)
/etc/init.d/httpd start (공격대상용@node0, 웹서버)

(3) (웹서버 접근@node11) 웹서버 접근 도구
wget

2. 네트워크 프로토콜 분야

IPv6는 차세대 인터넷프로토콜로써, 이미 주소할당이 중단된 IPv4를 대체할 프로토콜이다. 주소고갈로 인한 많은 기업과 관공서에서는 이미 IPv4 네트워크에서 IPv6 네트우크로 전환사업을 진행하고 있으며 IPv6 네트워크를 사용하는 곳이 늘어나고 있다. IPv6 네트워크 환경에서는 IP 주소는 Link Local Address 와 Global Unicast Address를 통하여 통신을 하게 되며, Link Local Address를 통하여 동일 네트워크에 있는 호스트들 간의 통신이 가능하고 Global Unicast Address를 통하여 다른 네트워크에 있는 호스트들과 통신이 가능하다. 이러한 IPv6 주소체계에서의 연구개발을 위하여 독립적인 테트워크의 구성이 필요하다. 이러한 목적으로 에뮬레이션형 테스트베드가 활용가능하다.

IPv6 노드를 설정하기 위하여, 본 테스트에는 IPv6 환경을 구축하고 간단한 테스트 환경을 갖추기 위하여 총 3대의 노드를 이용하여 구성하였다. 노드0에는 PC 라우터를 구성하기 위한 FreeBSD 노드, 노드1과 노드2에는 IPv6 노드들을 위한 CentOS 노드가 구성된다.

Reserved Nodes

Node ID	Name	Type	Default OSID	Node Status	Hours Idle[1]	Startup Status[2]	SSH	Console	Log
pc2	node0	dellR710	FBSD82-STD	up	1.01	none			
pc3	node2	dellR710	CENTOS55-STD	up	0	none			
pc14	node1	dellR710	CENTOS55-STD	up	1.07	none			

그림 6. 네트워크 구성 1

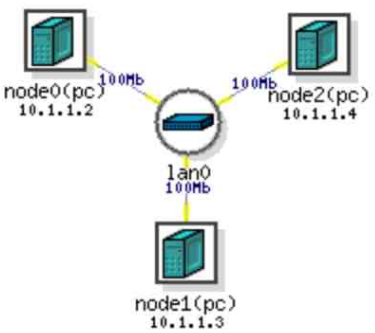

그림 7. 네트워크 구성 2

두 노드 모두 IPv6를 설정하기 위하여 IPv6를 설치하고 각각에 TCP dump 와 Wireshark 프로그램을 설치한다. CentOS 5.5 버전에서는 기본적으로 IPv6가 구성되지만, IPv6가 설치되지 않았으면 다음과 같은 명령으로 IPv6를 설치할 수 있다.

```
[root@node2 /]# modprobe ipv6
[root@node2 /]# lsmod
Module                  Size  Used by
nfs                   234433  3
lockd                  63337  2 nfs
fscache                20321  1 nfs
nfs_acl                 7617  1 nfs
ipv6                  270433  34
xfrm_nalgo             13381  1 ipv6
crypto_api             12609  1 xfrm_nalgo
sunrpc                146685  10 nfs,lockd,nfs_acl
tg3                   123721  0
e1000e                117025  0
```

그림 8. 노드에 IPv6 설치

ifconfig 명령을 이용하여 IPv6 링크로컬 주소를 확인하며, 각각의 노드에 다음과 같은 명령을 이용하여 wireshark 및 tcpdump를 설치한다.

<노드1> # yum install tcpdump	<노드2> # yum install wireshark # yum install wireshark-gnome # yum install libsmi

IPv6 PC 라우터를 구성하기 위하여 Kernel Network Information Table에 IPv6 기능에 필요한 정보를 추가하여야 하며, /etc/rc.conf(네트워크 설정 파일)에 다음의 엔트리를 포함한다.

```
ipv6_enable="YES" # Set to YES to set up for IPv6.
ipv6_network_interfaces="bce0 bce1 lo0" # List of network interfaces (or "auto").
ipv6_prefix_bce1="2001:db8:1:1" # Prefix for router advertisementi
ipv6_ifconfig_bce1="2001:db8:1:1::1 prefixlen 64" # Alias entry
ipv6_gateway_enable="YES" # Set to YES if this host will be a gateway.
rtadvd_enable="YES"
# Set to YES to enable an IPv6 router advertisement daemon. If set to YES, this rou
ter becomes a possible candidate IPv6 default router for local subnets.
gif_interfaces="gif0" # List of GIF tunnels (or "NO").
ipv6_firewall_enable="NO" # Set to YES to enable IPv6 firewall functionality
```

그림 9. PC 라우터의 IPv6 설정

그리고, PC 라우터를 재시작 하며, ifconfig 명령을 이용하여 IPv6가 정상적으로 작동하는지 확인한다.

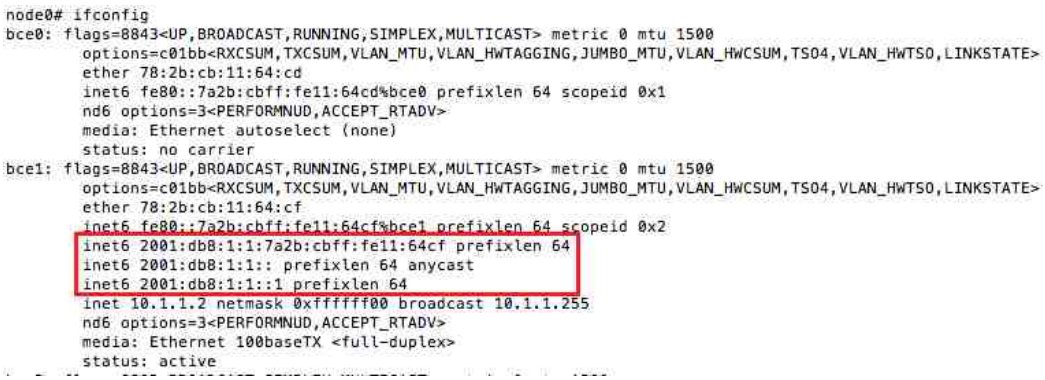

그림 10. IPv6 글로벌 주소 확인

3. 분산/병렬 연구 분야 : Hadoop

Hadoop은 MapReduce 기반 오픈 소스 소프트웨어 기반의 미들웨어로서 Yahoo, Facebook, Amazon, IBM, NexR 등 많은 기업들에서 클라우드 컴퓨팅 플랫폼으로 활용되고 있다. Hadoop은 크게 분산 파일 시스템인 HDFS와 분산 프로그래밍 모델인 MapReduce의 두 가지 구성요소로 이루어진다. Emulab에서의 하둡 클러스터 구성에서는 node0을 마스터 노드로 node1를 슬레이브 노드로 구성하여 테스트한다. 2개 노드로 구성된 토폴로지에서, Hadoop을 다음과 같은 절차에 의하여 설치한다.

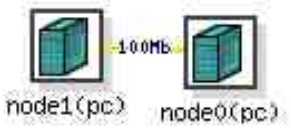

그림 11. 네트워크 구성

- JDK 설치

Hadoop 설치 디렉토리를 위한 디렉토리는 /hadoop/hadoop-1.0.2 으로 지정한다. Hadoop 응용 구동을 위한 서버 및 응용은 Java 언어로 작성되어 있으므로 Hadoop 설치를 위해서는 JDK 설치가 선행되어야 한다. 여기서는 OpenJDK 1.6.0_22 버전을 설치하여 사용한다. 설치되어진 버전의 확인은 다음 명령어를 사용한다.

- SH 키생성 및 공개키 복사

Hadoop 응용은 다수의 작업이 병렬 수행하는 과정을 통해 수행되므로 Hadoop 클러스터는 클러스터에 접근을 통제하기 위해 ssh을 기반으로 신뢰관계를 구성하여야 하며 이를 위해 공개키(public key)와 사적키(private key)를 구성한 후 시스템 접근을 통제한다. 여기서는 원격로그인을 위한 ssh 의 패스워크 없는 접근을 설정하도록 한다.

- node0 에서는 키쌍을 생성한 후 공개키를 node1에 복사한다.
- node0에서 ssh 를 사용하여 node1로 로그인이 되는지 확인한다.

```
ssh node1
```

- Apache 사이트에서 hadoop-1.2.0.tar.gz를 다운로드 한 후에 아래 명령을 통해 압축을해제한다

```
wget http://apache.mirror.cdnetworks.com/hadoop/common/hadoop-1.2.0/hadoop-1.2.0.tar.gz
tar xzf hadoop-1.2.0.tar.gz
```

- 압축 해제후 Hadoop 디렉토리는 /hadoop/hadoop-1.2.0 에 복사한다.

- 네임 노드 환경 구축

Hadoop 홈을 /hadoop/hadoop-1.2.0 으로 지정한 후 Hadoop 응용 수행을 위한 name 노드을 구동한 후 HDFS를 초기화한다. 이를 위해 다음 명령을 수행한다.

```
/hadoop/hadoop-1.2.0/conf/hadoop-env.sh 화일에 JAVA_HOME 셋업
bin/hadoop namenode -format
```

/local/hadoop-yhkang/dfs/name 에는 namde 노드에서 관리하는 데이터의 메타데이터에 대한 정보를 유지한다.

- 주요 데몬 구동

Hadoop 홈에서 다음의 명령을 수행하여 node0에서 데몬을 구동한다. (path 설정필요)

```
bin/start-all.sh
```

데몬 구동 후에는 Hadoop 클러스터를 구성하는 node0과 노드 9에서 데몬의 성공적으로 동작하는지 체크한다.
마스터 노드인 node0에는 NameNode와 JobTracker가 구동되어야 하며 NameNode의 자료의 체크포인트를 위해 SecondaryNameNode 가 구동되어야 한다.

```
[yhkang@node0 hadoop-1.2.0]$ jps
23480 JobTracker
23383 SecondaryNameNode
23156 NameNode
23589 Jps
```

슬레이브 노드인 node1에서는 DataNode 와 TaskTrack가 구동되고 있는지 확인이 필수적이다.

```
[yhkang@node1 ~]$ jps
26675 DataNode
26926 Jps
26806 TaskTracker
```

Hadoop 프레임워크는 웹기반으로 구성된 Hadoop 클러스터의 NameNode의 상태를 체크할 수 있도록 기능을 제공한다.

설치된 구성을 바탕으로 수행 테스트를 위하여, 주어진 문장속의 단어 개수를 카운트(count)하는 응용을 다음과 같이 수행한다.

Hadoop 응용의 수행에 필요한 자료는 하둡분산 파일시스템 HDFS에 위치하여야 한

다. 이를 위해 단어의 수를 얻기 위해 자료를 다음 명령어을 사용하여 HDFS 에 복사한다.

```
[yhkang@node0 hadoop-1.2.0]$bin/hadoop dfs -copyFromLocal ./dft dft
```

다음 명령어를 사용하여 해당 자료가 HDFS 에 복사되었음을 확인한다.

```
[yhkang@node0 hadoop-1.2.0]$ bin/hadoop dfs -ls
Found 1 items
-rw-r--r--   1 yhkang supergroup      13366 2013-06-20 17:51 /user/yhkang/dft
```

다음 명령어를 사용하여 Hadoop 응용인 wordcount 를 실행한다. wordcount 의 실행결과는 HDFS 의 dft-output 디렉토리에 저장된다.

```
bin/hadoop jar hadoop-examples-1.2.0.jar wordcount dft dft-output
```

다음 명령어를 사용하여 수행결과가 HDFS 의 디렉토리 dft-output에 저장되었음을 확인한다. 해당 디렉토리 dft-output 는 크기가 0인 _SUCCESS 이 생성되는 경우 작업이 성공됨을 의미하며 결과는 part-r-00000 에 저장된다.

```
bin/hadoop dfs -ls dft-output
Found 3 items
-rw-r--r--   1 yhkang supergroup          0 2013-06-20 17:57
/user/yhkang/dft-output/_SUCCESS
drwxr-xr-x   - yhkang supergroup          0 2013-06-20 17:56
/user/yhkang/dft-output/_logs
-rw-r--r--   1 yhkang supergroup       7376 2013-06-20 17:57
/user/yhkang/dft-output/part-r-00000
```

Hadoop 응용의 처리결과인 단어의 수를 얻기 위해 다음의 명령어를 사용한다.

```
[yhkang@node0 hadoop-1.2.0]$ bin/hadoop dfs -cat dft-output/part-r-00000
"AS     3
"Contribution"  1
"Contributor"   1
"Derivative     1
"Legal  1
"License"       1
"License");     1
"Licensor"      1
"NOTICE"        1
"Not    1
```

4. 교육 분야

공과대학의 학부수업에서는 학생들이 수업내용을 충실히 이해하고 수업내용의 활용을 극대화하여 교육의 성취도를 높이는 중요한 방법으로 이론적인 수업이외에도 컴퓨터를 활용한 실험 및 실습을 제공하고 있다. 이러한 실험 및 실습이 배우는 학생에게는 지식과 기술을 적용하는데 있어 매우 중요하지만, 실험실을 운영하는데 드는 비용의 문제와 한 번의 수업이 2-3시간이 소요되기에 일정의 조정이 어려울 수 있으며, 또한 실험조교가 수업 중에 생기는 질문에 답하고 도움을 주는 역할을 담당해야하는 어려움도 있다[11].

지난 수십 년간 기술의 발달은 이전보다 더 정밀하고 정확한 측정을 가능하게 하였고, 개인용 컴퓨터의 급속한 보급은 실험장비의 다양화와 이에 따른 활용을 통해 우수한 공학교육을 가능하게 하였다. 실험 실습의 중요한 목적은 이론수업을 현실세계에 적용 혹은 연결하여 학생들의 이해를 돕고 공학 수업에 대한 지속적으로 동기부여를 하는 것이라 할 수 있다.

1993년 미국의 IEEE 교육학회는 “전기(전자)공학교육에서의 계산과 컴퓨터”에 관한 특별 호를 통해 당시로는 최첨단의 시뮬레이션을 제공하는 PSPICE를 소개하였고 이를 이용하여 히스테리시스 효과를 모형 화하고 회로분석 및 회로 시뮬레이션을 수행하였다. 실험을 컨트롤하고 자료를 수집하며, 분석, 연관 짓고, 그 결과를 발표하는데 컴퓨터가 사용되었고 이렇게 바뀐 학부실험실에서 컴퓨터는 이론수업과 실험실습을 연결하는 가장 중요한 요소라 할 수 있다.

2006년부터 2011년까지 우리나라 정부는 IT분야에서 국내대학의 교육품질이 글로벌 수준에 부합하여 현장 적응력과 국제 경쟁력을 갖춘 IT전문인력을 양성하고자 대학IT전공역량강화(NEXT)사업을 실시하였고 이를 통해 교육여건 개선을 위한 실험실습 장비지원이 진행되었다. 하지만 여전히 컴퓨터 네트워크나 분산시스템과 같이 시스템중심의 실험 실습에서는 학생들이 네트워크계층이나 시스템커널에 접근해야하는 인프라 테스트의 복잡성으로 인해 많은 어려움을 가지고 있다. 이러한 어려움을 해결하기 위하여 에뮬레이션형 테스트베드를 활용한 IT 교육의 사례를 보여주고자 한다.

에뮬레이션형 테스트베드에서 생성되는 모든 새로운 프로젝트는 프로젝트 리더에 의해 시작이 되며 네트워크 테스트베드 운영위원회에서 승인이 되면 리더는 프로젝트에 참가하는 다른 참여자를 임명할 수 있는 권한을 위임받는다. 이러한 계층구조는 특히 수업에 활용하기에 적합하며, 지도교수의 지도하에 임명된 조교가 권한을 위임받아 프로젝트(실습)를 운영하게 된다.

각각의 사용자는 웹 인터페이스를 통해 네트워크 테스트베드 포탈에 접속을 하게 되며 이를 통해 노드 속성을 구성하거나, 가상의 토폴로지에 따른 실험을 시작 혹은 종료할 수 있다. 일단 실험이 시작되면, 사용자는 할당된 노드에 직접 접근할 수 있다. 노드구성은 Unix 동적 링크와 로딩으로 이루어지며, 노드의 메모리나 로컬 디스크는 지워질 수 있는, 변경이 가능한 상태로 실험이 “swap out(스왑아웃)"되어 자원이 다른 실험이 이용되는 것을 가능하게 한다. 사용자가 수정된 로컬 디스크를 저장하고 싶다면, 디스크 이미지를 데이터베이스에 저장할 수도 있다. 실험이 스왑아웃 되면 테스트베드에서는 데이터베이스안의 가상 토폴로지, 호스트 이름 그리고 일반적인 설정을 저장하며 다시 "swap in"을 하면, 실제 자원을 저장된 스크립트로 복원하여 네트워크를 구성한 후 해당 노드를 부팅함으로서 이전의 상태를 가져온다. 이는 하나의 실험이 너무 길어져서 자원이 독점적으로 활용되는 문제점을 방지하게 한다.

또한 동적인 실험제어를 지원하기위해 네트워크 시뮬레이터의 이벤트시스템을 사용하며, 사용자는 링크 속성을 제어할 수 있고, 실험중인 링크의 지연, 대역폭 그리고 손실률을 동적으로 변화시킬 수 있다. 테스트베드는 여러 사용자가 자원을 공유하기 때문에 효율적인 자원 활용 역시 중요하다 할 수 있다. 로컬 노드에서 트래픽, 콘솔 작업 여부, CPU 사용량 등으로 유휴(idle)상태를 판별하며, 유휴상태의 실험노드를 스왑아웃시켜, 다른 실험을 할 수 있는 노드로 변경할 수 있다.

가상토폴로지를 실제 장비로 자동 맵핑시키는 작업은 네트워크 관리자나 숙련된 학생연구원이 수행할 때 평균 3.25시간정도 걸리는 작업으로 테스트베드를 통해 간단하게 3분 미만에 설치된다는 것은 사용자가 실험을 수행할 때의 큰 어려움을 제거해 주는 것이라 할 수 있다[12].

테스트베드 제어에서의 중요한 기능중의 하나는 노드의 로컬디스크 내용이 자동으로 다시 로딩 되는 것이다. 현재 FreeBSD와 여러 버전의 리눅스 이미지가 제공 되고 있으며, 사용자 맞춤의 디스크이미지는 테스트베드에서 지원되지 않는 OS를 부팅하는데 사용될 수 있다. 지도교수가 원하는 프로젝트를 이미지로 관리하여 실습에 활용할 수도 있다.

실험을 주도하는 프로젝트리더는 에뮬레이션 테스트베드 웹사이트[13]를 통해 프로젝트계정을 개설하고 실험을 시작할 수 있으며, 실험의 모든 관리는 이 웹사이트를 통해 이루어진다. 에뮬레이션 테스트베드가 수업에 활용될 때에는 학생 하나하나가 계정을 신청하는 것이 아니라 지도교수가 지정한 조교에 의해 학생들의 계정이 관리될 수 있다.

다음은 컴퓨터공학 수업의 TCP/IP 네트워크 활용 예시를 보여준다. 에뮬레이션 테스트베드에서 네트워크 설정을 해보고 iperf를 사용하여 네트워크 성능을 측정해 보는 실습을 수행할 수 있다.

- 그림2와 같이 노드0에서 노드3까지 4개의 노드로 구성된 네트워크를 확인하고, 노드간 TCP 처리량(throughput)을 계산하기위해 iperf를 사용한다. iperf를 사용 시 하나의 노드에는 -s를 다른 한쪽의 노드에는 -c 옵션을 사용한다.

- iperf의 UDP옵션을 사용하여 UDP패킷을 보내고 패킷로스가 생길 때까지 -b옵션을 써서 대역폭을 조절한다.

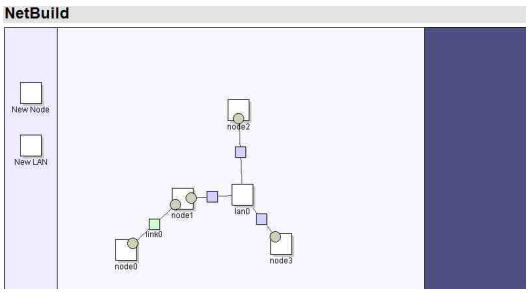

그림 12. 네트워크 구성도

- 노드 사이의 RTT가 큰 네트워크의 TCP 대역폭을 조절하고 -w옵션을 사용하여 서버와 클라이언트의 윈도우 크기를 조절할 수 있으며 -t 옵션으로 소요시간을 조절하여 링크의 상태를 조사한다.
- 각 노드사이의 가능한 TCP 대역폭을 찾아내고 각 링크에서 병목구간이 어디인지 조

사한다.

본 TCP/IP 예시에서와 같이 에뮬레이션 기반의 테스트베드는 시스템 및 네트워크를 다루는 컴퓨터공학의 모든 분야에서 다양한 용도로 사용될 수 있을 것으로 기대한다.

Ⅳ. 결론

미래의 네트워크는 보다 혁신적이고 진보된 고도화 네트워크로 진화하고 있다. 따라서 이러한 네트워크 기반의 선도적인 연구를 위하여 고도화된 테스트베드의 필요성이 제기되고 있다. 에뮬레이션형 테스트베드는 이러한 요구사항을 충족시켜줄 것이라 예상한다. 본 문서에서는 이러한 에뮬레이션형 테스트베드와 특정 분야에 대하여 적용하는 성과를 보여주었다.

참 고 문 헌

[1] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, “An integrated experimental environment for distributed systems and networks,” in USENIX OSDI, Boston, MA, Dec. 2002, pp. 255–270.

[2] M. Ott, I. Seskar, R. Siraccusa, and M. Singh, “ORBIT testbed software architecture: Supporting experiments as a service,” in IEEE Tridentcom, Trento, Italy, Feb. 2005, pp. 136–145.

[3] M. Hibler, L. Stoller, J. Lepreau, R. Ricci, and C. Barb, “Fast, scalable disk imaging with Frisbee,” in Proc. of the 2003 USENIX Annual Technical Conf., San Antonio, TX, Jun. 2003, pp. 283–296.

[4] Flux Research Group, University of Utah, “The Emulab Web site,” http://www.emulab.net/.

[5] A. Bavier, N. Feamster, M. Huang, J. Rexford, and L. Peterson, “In VINI veritas: Realistic and controlled network experimentation,” in ACM SIGCOMM, Pisa, Italy, Aug. 2006, pp. 3–14.

[6] T. Miyachi, A. Basuki, S. Mikawa, S. Miwa, K. Chinen, and Y. Shinoda, “Educational environment on StarBED: case study of SOI Asia 2008 spring global E-Workshop,” in ACM Asian Conference on Internet Engineering, Bangkok, Thailand, Nov. 2008, pp. 27–36.

[7] Y. Chen, K. Hwang, W. Ku, "Collaborative Detection of DDoS Attacks over Multiple Network Domains," IEEE Transactions on Parallel and Distributed Systems, vol.18, no.12, pp.1649-1662, Dec. 2007

[8] M. Natu, J. Mirkovic, "Fine-grained Capabilities for Flooding DDoS Defense Using Client Reputations," Proceedings of the 2007 Workshop on Large Scale Attack Defense, LSAD '07, pp. 105-112, 2007

[9] C. Neuman, C. Shah, K. Lahey, "Running Live Self-Propagating Malware on the DETER Testbed," DETER Community Workshop on Cyber Security Experimentation and Test, Jun 2006

[10] S. Hanna, D. Nicol, "Implementation and Instrumentation of a Flash-Worm," DETER Community Workshop on Cyber Security

Experimentation and Test, Jun 2006
[11] J. Campbell, J. Bourne, P. Mosterman, and A. Brodersen, "The effectiveness of learning simulations for electronic laboratories," Journal of Engineering Education, Vol. 91, No. 1, pp. 81-87, 2002.
[12] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. "An integrated Experimental Environment for distributed systems and networks," OSDI 2002, Dec. 2002.
[13] KREONET-Emulab, https://www.emulab.kreonet.net.