

통신시스템의 내부연동망을 위한 LAN 스위치 성능 시험기 개발

김동원*, 조동욱*, 박병준**, 허재두***

*충북도립대학 정보통신과

**한양대학교 전자공학과

***한국전자통신연구원 서비스네트워크연구부

e-mail : dwkim@occ.ac.kr

LAN Switch Performance Tester for the Interconnection Network of a Communication System

Dong-Won Kim*, Dong-Uk Cho*, Byung-Joon Park**, Jae-Doo Huh***

*Dept. of Information Communications, Okchon Provincial University

**Dept. of Electronics Engineering, Han-Yang University

***Dept. of Service Network, ETRI

요 약

본 논문에서는 대용량 통신처리시스템의 트래픽 처리 능력 예측을 위해 실제 네트워크 구축 상황과 서비스 상황을 모의로 만들어 줄 수 있는 테스터 소프트웨어 개발을 목적으로, 리눅스환경에서 Network Traffic Generator, Remote 노드에 사용될 패킷 귀한 프로그램, Statistical Analyzer, GTK+ 라이브러리를 이용한 GUI, Network Traffic Generator 와 Statistical Analyzer 를 하나의 프로그램 안에서 동시에 동작하도록 하는 Thread 사용 기법이 연구되었다.

이를 이용하여 대용량 통신 처리시스템의 성능 규격의 정확한 확인을 통해 가용 가능한 가입자 노드의 용량을 결정할 수 있으며, 상용현장시험이 가능하고 또한 설치 규격을 제공할 수 있다.

1. 서론

대용량 통신 처리 시스템^[1]은 PSTN, ISDN, Frame Relay, X.25, LAN, ATM 등 다양한 통신망들이 내부의 백본망인 Ethernet switch 를 통해 상호 연결접속이 되고, 사용자 트래픽, 제어 트래픽 등이 이 Ethernet switch 에 의해 교환되고 수송된다. 따라서 백본망의 성능에 따라 대용량 통신 처리 시스템의 성능이 결정된다. 그러므로 대용량 통신 처리 시스템을 구축함에 있어서 백본망인 Ethernet switch 의 성능을 정확하게 측정할 필요가 있다.

본 연구에서는 이러한 대용량 통신 처리 시스템의 성능 측정을 위해 실제 네트워크 구축 상황과 서비스 상황을 모의로 만들어 줄 수 있는 테스터 소프트웨어 개발을 목적으로 한다. 이와 같이 대용량 통신 처리 시스템의 성능 규격의 정확한 확인을 통해 가용가능한 가입자 노드의 용량을 결정할 수 있으며, 상용현장시

험이 가능하고 또한 설치 규격을 제공할 수 있다.

최종적으로 본 연구에서는 다음과 같은 내용이 개발되었다. 사용자가 패킷의 크기와 패킷 전송 시간 간격 및 전송 패킷을 결정하여 모의 패킷을 전송할 수 있는 Network Traffic Generator, Remote 노드에 사용될 패킷 귀한 프로그램, 패킷을 수신하고 수신된 패킷으로부터 전송지연시간과 데이터 전송율을 계산 분석하는 Statistical Analyzer, GTK+ 라이브러리를 이용한 GUI, Network Traffic Generator 와 Statistical Analyzer 를 하나의 프로그램 안에서 동시에 동작하도록 하는 Thread 사용 기법 등이 개발되었다.

2. 통신 처리시스템의 백본망 시험기

본 연구의 궁극적인 목표는 Ethernet 백본망의 성능을 테스트 할 수 있는 소프트웨어 도구의 개발이다. GUI 환경에서 모의로 패킷을 만들고, 이것을 전송후

귀환(loopback)된 패킷을 다시 수신하는 실험을 통해 일정한 시간동안 백본망이 처리할 수 있는 패킷의 수와 전체 데이터의 전송율을 측정할 수 있는 소프트웨어를 개발한다.

본 연구에서는 위와 같은 궁극적인 목표를 달성하기 위하여 다음과 같은 기능이 필요하다.

- 모의 실험에 사용할 패킷의 크기와 내용을 사용자가 지정하고 수정할 수 있는 기능.
- 패킷을 전송할 때, 패킷들의 전송간의 시간 간격(time interval)의 유형과 평균 시간 간격을 사용자가 선택할 수 있는 기능.
- Remote 노드에 사용될 패킷 귀환 프로그램.
- Remote 노드에서 귀환된 패킷을 수신하는 기능.
- 패킷의 전송지연시간과 데이터 전송율 계산을 위한 분석기능.

2.1 시스템의 전체 구조

통신 처리시스템의 시험을 위해서는 실제 통신 처리시스템을 중심으로 각각의 노드에 가상의 Traffic 발생장치를 부착하고, 임의의 Traffic 패턴에 대해 통신처리시스템이 어떻게 처리하는가에 대한 결과를 얻어야 한다. 이를 위하여 전체 시험용 시스템을 구성하면 그림 1 과 같다.

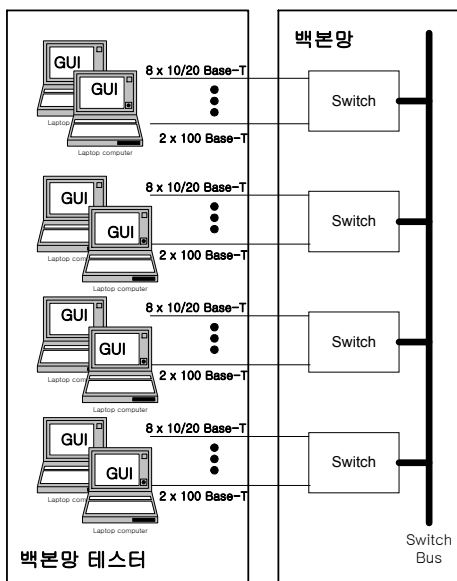


그림 1. 대용량 통신 처리 시스템의 블록도

이 때 Master 노드에서는 PC 를 이용하여, 임의의 Traffic 발생을 할 수 있도록 하고 Slave 노드에서는 다른 PC 를 이용하여 자신에게 전송되는 packet 을 원래의 Master 노드에게로 돌려 준다. 그림 2 는 이와 같이 각 노드의 역할을 보여준다.

Master 노드와 Slave 노드는 기본적으로 소켓에 패킷데이터를 써주고, 소켓으로부터 데이터를 캡처하는 면에서 같은 중복된 기능으로 사용할 수 있다. Master 노드의 중요 기능별 블록도를 보면 그림 3 과 같다. Slave 노드의 기능별 블록도는 그림 4 와 같다.

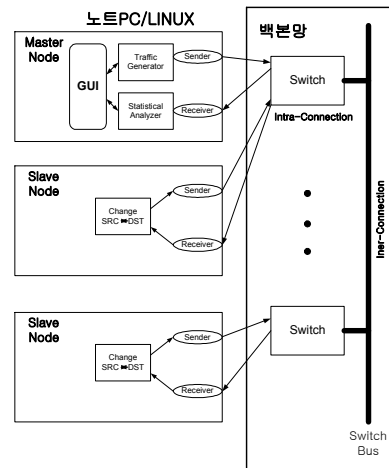


그림 2. 각 노드의 내부가 보이는 전체 블록도

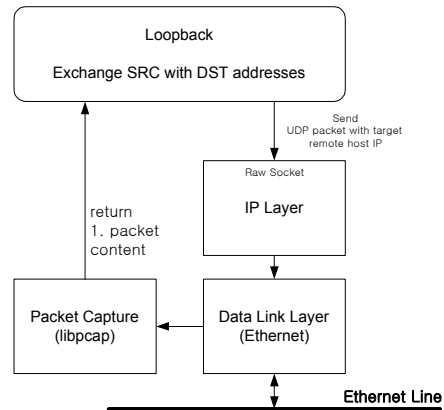


그림 3. 마스터 노드의 블록도

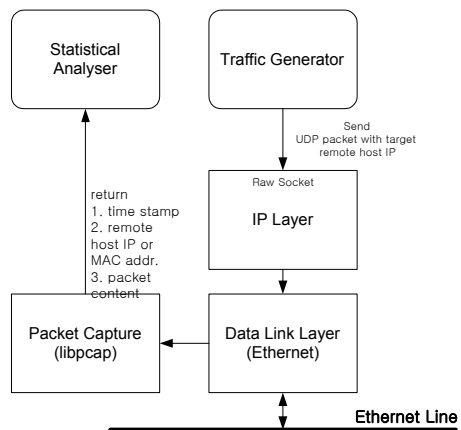


그림 4. 리모트 노드의 블록도

이렇게 구성한 테스트 프로그램의 중요 세가지 부분은 Packet Generator 와 Statistical Analyser 그리고, Remote Loopback 으로 나눌 수 있다. 2.2 절 부터는 이 세가지 부분에 대한 자세한 설명이 있다.

2.2. Packet Generator

Packet Generator 에서는 packet 을 생성하고 이를 socket 에 쓰는 일련의 과정을 포함하고 있다. 이를

흐름도로 나타내면 다음과 같다.

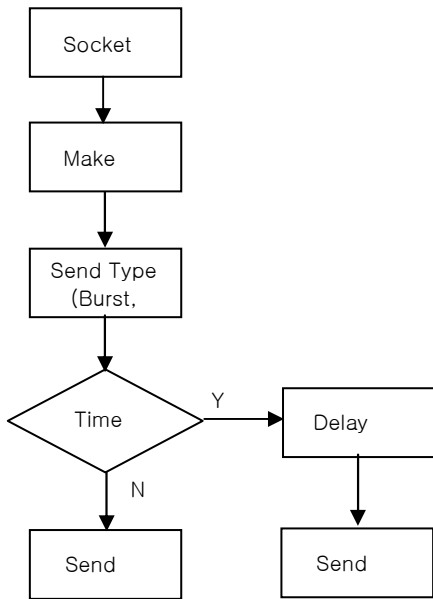


그림 5. Generator 의 흐름도

Packet 의 내용을 목적지 주소로 보내기 위해서는 소켓을 생성하여 이를 통해 전달하게 된다. 여기서는 소켓에서 packet 을 읽고 쓸 수 있도록 rawsocket (rawsocket)을 생성하게 된다. 이는 데이터 링크층에서 직접 packet 을 쓰고 읽을 수 있는 기능을 갖게 된다.

Packet 은 UDP 헤더와 IP 헤더, 그리고 사용자 데이터로 이루어져 있다. UDP 헤더는 8 바이트 크기이며 IP 헤더는 20 바이트의 크기를 갖는다. UDP 프로토콜은 TCP 프로토콜과는 다르게 연결을 맺고 끊는 과정이 필요 없고 단지 요청과 응답을 교환하는 두개의 packet 만을 이용하게 된다. 또한 UDP 는 포트 번호를 제공하고 체크섬을 계산하기 위해 사용하게 되고 이를 위해서 체크섬이 계산되기 전에 가상헤더를 만들게 된다. 이런 packet 을 분석하기 위해서는 기본적인 정보 외에 추가된 정보가 필요하게 된다. 목적지 주소가 하나가 아닌 여러 개가 존재하게 되면 목적지 주소에 따라서 buffer 에 각 packet 을 저장하게 되고 분석을 용이하게 하기 위해서 packet 의 ID 를 포함하게 된다. 또한 시간적인 측면에서 분석하고자 packet 에 timestamp 를 추가하게 된다. 따라서 여기서 사용하는 packet 의 구조는 UDP 헤더 8 바이트와 IP 헤더 20 바이트, 그리고 추가된 ID 1 바이트, second 와 microsecond 정보를 포함하는 timestamp 8 바이트, 사용자 데이터로 구성되어 있다.

Packet 을 보내는 형태는 두 가지로 나눌 수 있는데 사용자가 packet 의 보내는 회수를 정하는 회수 지정 형태와, 그리고 몇 분 동안 packet 을 보내는 시간 지정 형태가 있다. 이 때 각 패킷을 전송하는 시간간격을 세 가지의 형태 중에 선택할 수 있다. 시간 간격이 없이 연속해서 전송하는 경우와 사용자가 임의로 정한 일정한 시간동안 기다린후 다음 패킷을 전송

하는 uniform 형태, 그리고 패킷의 Transmission 패턴이 Poisson 분포를 갖도록 time 을 정하는 형태를 지원한다. 이는 랜덤값 추출과 시스템의 시간을 이용하여 second 와 microsecond 단위의 시간으로 패킷간의 시간간격을 정하게 된다.

생성된 packet 을 소켓을 통하여 원하는 목적지로 보내게 된다. 이 때 packet 을 분석하기 위해 timestamp 를 포함하게 되는데 이는 delay 에서와 마찬가지로 시스템의 시간을 second, microsecond 단위로 생성하게 된다. 또한 목적지 주소가 여러 개가 있을 경우는 목적지 주소 개수만큼 packet 을 순차적으로 보내게 된다.

2.3. Statistical Analyzer

통신처리 시스템의 성능을 측정하기 위해서 많이 사용되는 통계 값으로는 Packet 이 전달되는 데 걸린 시간을 측정하는 지연시간(Delay Time)과 통신시스템이 처리할 수 있는 최대 데이터 처리율(Maximum Transfer rate)이 있다.

본 연구에서 개발한 시스템에서는 packet 을 만들어 보낼 때, timestamp 를 삽입하고, 상대방의 remote 호스트에서 귀환(loopback)된 packet 으로부터 timestamp 를 추출한 후 현재의 시스템 시간과 비교하는 방법을 이용하여 통신처리 시스템상의 지연시간을 측정한다. 이 때에는 각각 개별 packet 에 대해서 구한 값을 바로 사용하지 않고, 단위시간(1 초)동안 각 packet 의 delay 값을 누적한 후, 평균과 최대, 최소 값을 구하는 방법으로 보다 정확한 지연시간 수치를 구할 수 있다.

또한 전체 전달된 단위시간당 데이터의 양을 바이트 단위로 계산하면 이 통신 시스템이 처리할 수 있는 최대 데이터 전송율을 구할 수 있다. 데이터 전송율은 보통 BPS (Bits Per Second) 단위로 표시하는데, 단위시간당 전체 데이터 송신량과 전체 데이터 수신량을 더하면, 전체 데이터 전송율이 된다. 이와 같은 통계값을 구하기 위한 Statistical Analyzer 의 흐름도는 그림 6 과 같다.

2.4. Remote Loopback

Slave 노드에서는 Master 노드로부터 받은 패킷을 원래의 Master 노드에게로 돌려 보내는 역할을 한다. 이러한 역할을 하기 위해서 먼저 패킷을 캡처한 후 해당 패킷이 자신에게 온 패킷인지 확인한다. 그리고 나서 자신에게 온 테스트 프로그램의 패킷임을 확인하면, 주소영역에서 패킷의 목적지와 근원지 항목을 서로 바꾼 후, 다시 소켓에 써 넣는다. 이렇게 하여 패킷을 아무런 변경없이 원래의 Master 노드에게로 돌려보낸다. 그림 7 에서는 이와 같은 과정을 흐름도로 표시하였다.

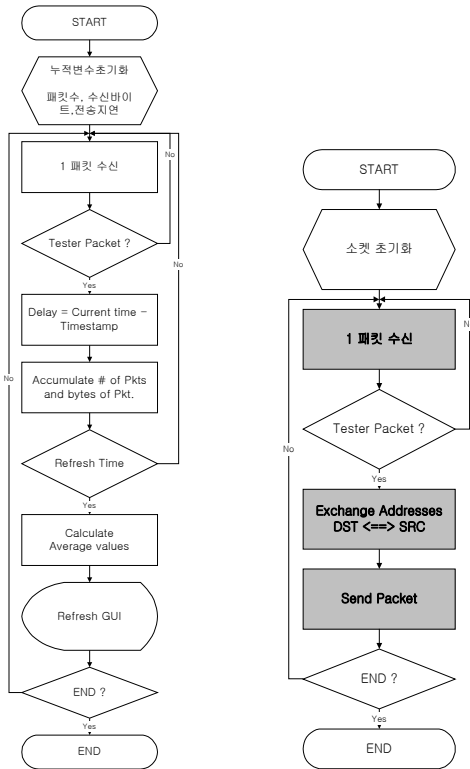


그림 6. Statistical Analyzer 그림 7. Remote Loopback
의 흐름도 흐름도 3

3. 시험 프로그램의 개발환경^[2,5,6]

3.1 Packet Creation 과 Transmit

소켓을 생성은 socket 함수를 호출함으로 이루어진다.

Packet 생성은 UDP 헤더와 IP 헤더, 그리고 사용자 데이터를 생성하는 것이다. 또한 packet 의 식별위해서 ID 를 포함한다. 그리고 여러 목적지 주소를 처리하기 위해서 배열을 만들어서 목적지에 따라서 packet 을 생성시키게 된다.

Packet 을 보내는 형태가 3 가지로 나타낼 수 있는데 delay 없이 보내는 burst 형태와 packet 의 보내는 회수(times)를 정하는 형태가 있다. 그리고 몇 분 동안(time) packet 을 보내는 형태가 존재한다. burst 형태는 아무 조건없이 소켓에 packet 을 바로 쓰는 경우이고 times 형태는 사용자가 임의로 회수를 정하여 그 만큼의 packet 을 보내게 된다. 그리고 time 형태는 사용자가 몇분 동안 packet 을 보낼 건지 임의로 정하여 그 시간동안 packet 을 보내게 되는데 이를 구현하기 서 gettimeofday 함수를 이용하여 시스템의 시간을 사용하게 된다.

Time delay 는 times 형태와 time 형태에서 적용되는 것으로서 packet 을 보낼 때 delay 를 포함하여 packet 을 보내게 된다. Time delay 형태는 세 가지가 있는데 첫째는 delay 가 없는 경우, 두 번째는 일정한 시간을 delay 로 정하는 경우이다. 세 번째는 다음 버전에서 구현될 Poisson 분포를 갖는 time delay 형태이다. 이의 구현은 gettimeofday 함수를 이용하여 second 와

microsecond 형태의 시스템 시간을 이용하여 원하는 시간동안 while 문을 돌면서 소켓에 packet 을 쓰게 된다.

모든 패킷이 만들어 졌으면 이것을 데이터링크층에 써넣는 일만 남았다. 이를 위하여 sendto 함수를 사용한다.

sendto(rawfd, buf, userlen, 0, dest, destlen)

또한 packet 을 분석할 때 시간개념에서 분석하기 위해서 packet 을 보낼 때 packet 의 timestamp 를 기록하는 과정을 거치게 된다. 이는 time delay 부분에서 마찬가지로 gettimeofday 함수를 이용하여 second, microsecond 단위의 시간을 packet 의 기록하게 된다.

3.3. Packet Capture

네트워크상의 패킷 정보를 Data Link Layer 수준에서 읽어 들이기 위해서는 몇 가지 방법이 있는데, 그 중 한가지는 하드웨어적인 메모리 내용을 직접 읽어내는 방법이고, 다른 하나는 Data Link Layer 수준의 소켓을 생성하여 읽는 방법이 있다. libpcap 은 버클리대학에서 만든 공개한 것으로, 후자에 속하는 Data Link Layer 수준의 소켓을 생성하고 패킷을 읽어내는 범용의 라이브러리이다.

이 라이브러리는 네트워크상의 모든 패킷을 잡아내는 packet capturing 기능 뿐만 아니라, 사용자가 원하는 packet 만을 선택할 수 있는 packet filtering 기능도 제공한다. 가장 먼저 pcap_open_live()함수를 이용하여, 소켓을 생성한다. 생성된 소켓에 필터를 설정할 때는, 먼저 pcap_compile()함수를 이용하여, 필터를 만들어 준 다음, pcap_setfilter()함수를 이용하여 만들어진 필터를 설정해 준다. 이렇게 일단 설정을 해 놓으면, 두 가지 방법으로 패킷을 캡처할 수 있는데, 첫번째는 pcap_next() 함수를 이용하여 무조건 한 개씩 패킷을 받아오는 방법이고, 두번째는 pcap_loop()나 pcap_dispatch()함수를 이용하여 패킷이 일정한 크기만큼 받아지면 callback 으로 지정된 함수를 자동 호출하는 방식이다.

본 연구에서는 패킷의 수와 크기가 모두 필요하므로, pcap_next()함수를 이용하여 한 개씩의 패킷을 가져오는 방식으로 구현하였다.

3.3. POSIX Thread 활용^[3]

본 연구에서는 그림 8 과 같이 Traffic Generator 와 Statistical Analyzer 등, Concurrent 를 유지하여야 하는 모든 동작을 thread 로 처리하여 효율적인 동작과 각 동작단위 간의 빠른 데이터 전송을 구현하였다. GUI 상에서 패킷테스트를 시작하게 되면, 패킷을 전송을 담당하는 Thread 와 패킷의 수신 후, 분석 값을 저장하는 두 개의 Thread 를 수행하고, 원래의 Main thread 에서는 매 초마다 수신 thread 에 의해 저장된 자료값을 바탕으로 통계치를 화면에 출력하는 역할을 맡아서 수행한다.

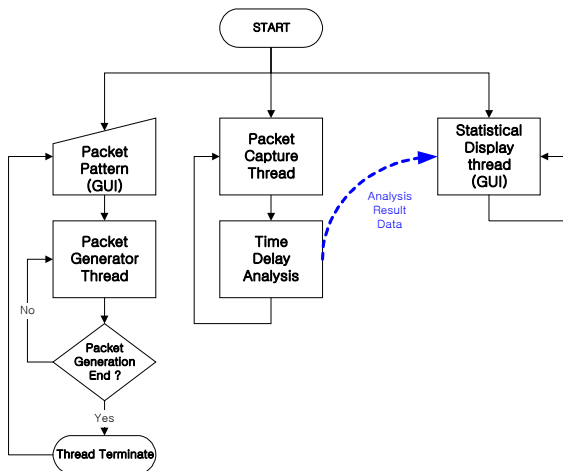


그림 8. Thread 를 이용한 Traffic Generator 와 Statistical Analyzer, 그리고 GUI 의 동작 관계

4. 프로그램 사용방법

본 연구에서는 리눅스 상에서 GUI 를 구현하기 위하여, GNU 에 의해 공개된 라이브러리인 GTK+^[4]를 사용하여 개발하였다. 이 장에서는 각 GUI 구성요소에 대해서 설명하고, 각각의 사용법을 기술한다.

그림 9 는 미지리눅스 1.0 버전에서 실행한 테스트 프로그램의 주화면이다.

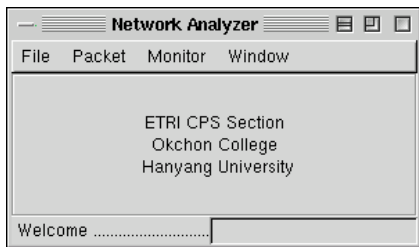


그림 9. 테스트 프로그램의 주화면

4.1. Packet Generator Control Panel

패킷을 만들어 송신할 때는, 먼저 송신을 종료할 조건에 대해서 선택을 한다. 그림 10 의 왼쪽영역이 바로 송신 조건을 지정하는 부분이다. 몇 개의 패킷을 보내도록 선택하거나, 일정한 시간동안 패킷을 보내도록 지정할 수 있다.

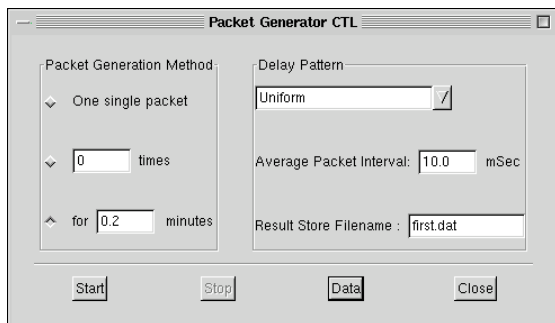


그림 10. Packet Generator Control Panel 화면

Control 창에는 네 개의 단추가 있는데, 각각의 단추는

Start : 테스트를 시작한다.

Stop : 현재 진행중이 테스트를 중단시킨다.

Data : 테스트를 할 때, 사용할 Slave 의 IP 주소를 설정하거나, 테스트에 사용할 패킷의 내용물을 생성하는 창을 띄운다.

Close : 테스트를 종료하고, Control 창을 닫는다. 기능을 갖고 있다.

그림 11 에서는 각 패킷을 전송할 때, 전송 패킷들 사이의 시간 간격을 지정하는 것을 보여준다. 시간 간격이 없이 연속적으로 보내는 “Burst”방법과 모든 패킷을 일정한 시간간격으로 전송하는 “Uniform” 방법이 있다.

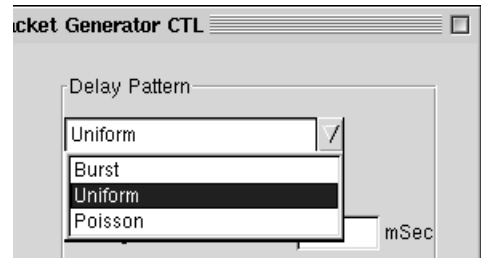


그림 11. Packet 간 Interval 패턴의 설정

시간 간격을 정하는 방법을 결정하고, 시간 간격을 mSec 단위로 결정해서 써 준 다음, 분석결과를 저장할 텍스트 파일의 이름을 Result filename 항목에 써준다.

4.2. Packet Generator Data Setup Dialog

그림 12 에서 보여주고 있는 Data Setup 창은 메뉴에서 “Packet Data”를 선택하거나 Control 창에서 “Data” 단추를 누르면, 나타난다. 왼쪽에 보이는 부분은 테스트에 사용할 Slave 노드의 IP 주소를 적어준다. 여러 개의 Slave 가 있으면 각각의 IP 주소를 순서대로 적으면 된다.

오른쪽에는 테스트에 사용하는 임의의 패킷을 생성할 수 있는 창을 가지고 있다. 실제 테스트에서는 패킷의 헤더와 테스트에 사용되는 정보로 인하여 이 창에서 설정한 패킷 내용크기보다 54 바이트가 더 큰 패킷이 생성된다. 즉, 20 바이트의 패킷 내용을 생성한다면 실제 테스트에 사용되는 한 개의 패킷의 크기는 74 바이트가 된다. 패킷 내용의 크기를 결정하고 “Fill” 단추를 누르면 임의의 난수를 발생시켜 필요한 크기만큼의 패킷 내용을 자동으로 채워준다.

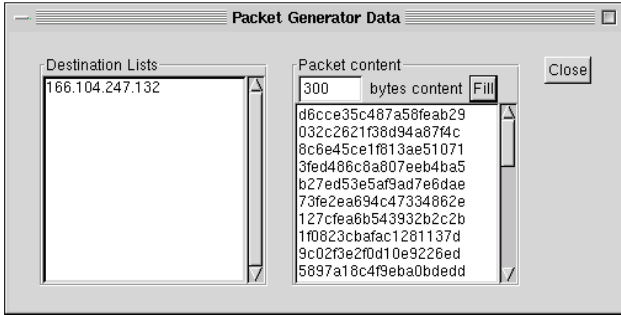


그림 12. Packet content data 설정 창

이상의 내용을 모두 기록해서 채워주면, 실험을 위한 준비를 다 마친 것이다. 이제 Control 창에서 "Start" 버튼을 누르면, 실험을 시작한다.

4.3. Dashboard

그림 13 에서는 분석결과를 보여주고 있다. 맨 첫 번째 줄은 현재 실험에서 누적된 패킷의 전송 및 수신량을 보여준다. 그 다음 줄은 전송 및 수신한 데이터의 바이트 수를 표시해 준다.

세 번째 줄부터는 1 초 간격의 분석 값을 보여주고 있는데, Throughput 은 매 초마다의 Kbps 단위의 전송률을 표시하여 준다. 세 번째 줄 때 오른쪽의 Max 항목은 매 초마다 갱신되는 전송률 중 최대값을 표시하여 준다.

네 번째 줄에서는 각 1 초 간격으로 패킷의 반송되기까지의 시간을 마이크로초 단위로 계산한 다음 이것의 평균을 낸 것인데, 여기서 Min 과 Max 항목은 각각 모든 패킷 중에서 가장 짧은 반송시간과 가장 긴 반송시간에 대해서 값을 표시한 것이다. 각 1 초 간격의 반송시간에 대한 분산값을 표시하였다.

Packet Generator Analyzer				
	Generated	Received	Sum	Loss
Total pkts	427	427	854	Not Yet Avail
Total bytes	35868	35868	71736	Not Yet Avail
	Generated	Received	Sum	Max
Throughput(bps)	108.	108.19	216.38	216.38
	Min	Avg	Max	Variance
Return time(usec)	767.000	787.559	13064.000	2931050.992

그림 13. 패킷의 분석결과 표시창

모든 시험이 끝나면, Control 창에서 지정한 파일에 매초 간격으로 갱신된 Throughput 과 Return time 의 값들이 저장되어 있다. GUI 에서는 순간순간의 테스트 결과를 확인하고, 최종 분석이나 보고서 작성을 위해서는 저장된 텍스트 파일을 사용한다.

5. 결론

본 연구에서는 리눅스 OS 를 기반으로 하여 PC 를 활용한 Lan switch 장비를 테스트할 수 있는 소프트웨어를 개발하였다. 본 프로그램을 활용하여 Lan switch

장비의 효과적인 성능분석 및 확인이 가능하다.

현재의 테스트 소프트웨어는 소켓에 데이터를 쓸 때, IP 계층에서 패킷을 작성하여, Data Link Layer 에 패킷을 써준다. 그러나 시스템 내부에서 커널작업에 소요되는 시간을 더 줄여 보다 정확한 시간 측정값을 얻기 위해서는 직접 Ethernet 패킷을 작성하여 물리계층에 패킷을 기록해 줄 필요가 있다.

또한 패킷을 받을 때도 범용의 패킷캡처용 라이브러리인 libpcap 을 사용하였으나 직접 Ethernet 계층의 소켓을 생성하여 읽어내면 더 좋은 성능을 얻을 수 있을 것이다.

참고문헌

- [1] 김도영, 김대웅, 김동원, "고속 이더넷 기반의 통신처리시스템 백본망의 설계 및 구현", 한국통신학회 추계 학술발표, pp.246~249, 1998
- [2] W. Richard Stevens, 김치하 이재용 공역, UNIX Network Programming: 2nd ed. Volume1 Networking APIs: Sockets and XTI, 교보문고 1999.
- [3] David R. Butenhof, Programming with POSIX Threads, Addison-Wesley, 1997.
- [4] Eric Harlow, Developing Linux Applications with GTK+ and GDK, New Riders, 1999.
- [5] Matt Welsh and Lar Kaufman, Running Linux: 2nd ed., O'Reilly, 1999
- [6] Uday O. Pabrai, UNIX Internetworking: 2nd ed., Artech House Inc. 1996.