제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

비쥬얼 이더넷 트래픽 발생기의 설계 및 구현

정인환, 김진환
한성대학교 컴퓨터공학부
e-mail : ihjung@hansung.ac.kr, kimjh@ice.hansung.ac.kr

Design and Implementation of Visual Ethernet Traffic Generator

Inhwan Jung, Jinhwan Kim
School of Computer Engineering
Hansung University, Seoul 136-792, Korea

요 약

본 논문에서는 이더넷(Ehternet)의 네트워크 진단과 분석을 위해 필수적인 트래픽 발생기(traffic generator)를 설계하고 구현한다. 이더넷 트래픽은 패킷의 양과 프로토콜의 종류 그리고 패킷의 길이 등에 영향을 받는다. 본 연구에서 구현하고자 하는 트래픽 발생기는 이 세가지 항목을 편리한 사용자 인터페이스를 통해 선택하여 다양한 네트워크 트래픽을 발생하는 것을 목표로 한다. 구현된 트래픽 발생기 VTG 는 허브, 라우터 등 네트워크 장비의 성능 평가와 패킷을 모니터링하는 소프트웨어들인 네트워크 진단 시스템과 침입탐지 시스템 등의 성능 평가 및 검증을 위해 사용될 수 있다.

1. 서론

현재 이더넷(Ehternet)은 가장 널리 사용되는 근거리 통신망의 형태이다. 이더넷은 네트워크의 물리계층(physical layer)을 많은 종류의 네트워크 장비와 컴퓨터들이 접속되어 사용되며 각각의 장비들에서는 다양한 데이터가 오고 가고 있으며 물리적으로 하나의 네트워크를 공유하기 때문에 네트워크의 트래픽은 예측되기 어렵다. 이렇게 예측되기 힘든 네트워크 트래픽으로인해 이더넷에 접속되어 있는 네트워크 장비등의 성능 평가와 검증을 위한 별도의 트래픽 발생기(traffic generator)가 요구된다.

본 논문에서는 이더넷 네트워크의 진단과 분석을 위해 필수적인 트래픽 발생기를 설계하고 구현하였다. 구현된 트래픽 발생기 VTG(Visual Ethernet Traffic Generator)는 윈도우 환경하에서 수행되는 프로그램으로 편리한 사용자 화면을 제공하며, 다양한 방법으로 이더넷 패킷을 발생함으로써 네트워크의 성능을 다양한 각도에서 검사할 수 있게 해 준다.

본 논문의 구성은 다음과 같다. 2 장 관련연구에서는 이더넷 패킷을 직접 처리할 수 있는 환경과 이더넷의 트래픽 모델링에 대하여 설명한다. 3 장에서는 VTG 의 설계 및 구현에 대하여 설명한다. 4 장에서는 VTG 를 이용한 트래픽 발생 결과와 성능에 대하여 설명한다. 마지막으로 5 장에서는 결론 및 향후 연구에 대하여 기술한다.

2. 관련연구

2.1 저수준 패킷 처리 환경

본 연구에서는 이더넷에 패킷을 직접 읽고 쓸 수 있는 환경인 Winpcap[2]을 사용하여 트래픽 발생기를 구현하였다.

이더넷에서 네트워크 인터페이스를 통해 패킷을 직접 읽거나 쓰는 연구는 대표적으로 BPF[1]와 WinPcap[2]이 있다. BPF 는 Unix 환경하에서 패킷 감시 라이브러리인 libpcap 의 형태로 제공되며 Unix 또는 Linux 환경하에서 효과적으로 패킷을 감시하는 기능을 제공하였다. BPF 는 사용자가 정의한 조건(filter)를 운영체제 안에서 검사하여 조건에 맞는 패킷만 사용자 버퍼에 복사하는 구조를 갖고 있다. WinPcap 은 기존의 BPF 와 libpcap 의 호환성을 유지하면서 윈도우에서 패킷을 감시할 수 있는 환경을 제공한다. WinPcap 은 BPF 가 2 단계 버퍼를 사용하는 것과 달리 원형 버퍼를 사용함으로써 커널에서 사용자 버퍼로 데이터가 복사되는 도중에도 패킷을 수집하여 저장할 수 있다. 따라서 WinPcap 이 BPF 에 비하여 높은 성능

을 보였다[2].

WinPcap 의 또다른 특징은 저 수준 패킷 전송 기능을 제공한다는 것이다. BPF 와 WinPcap 모두 저 수준의 패킷 전송 기능을 제공한다. 그러나 libpcap 은 BPF 의 이런 기능을 사용하지 않는다. Windows2000 에서도 저 수준 소켓을 제공하지만 제한적인 기능만 가능하다. 따라서 본 연구에서는 트래픽 발생기를 구현하기 위하여 WinPcap 의 패킷 전송 기능을 이용하였다.

2.2 이더넷 트래픽 모델링

이더넷 트래픽에대한 분석과 모델링은 많은 연구가 있어왔다[3][4[5]. 기존의 연구에서 이더넷의 트래픽은 자기 유사성(self-similarity)를 갖는 것으로 알려져왔다. 즉, 짧은 시간동안 발생하는 트래픽의 형태는 긴 시간을 두고 보았을 때 유사성을 갖는다는 것이다. 이러한 배경하에 [3]에서는 실제 이더넷 트래픽과 유사한 트래픽을 발생하는 모델을 제시하였다. 그리고 인공적으로 합성된 이더넷 트래픽이 실제 트래픽과 같이 자기 유사성을 갖는 것을 보였다. 자기 유사성을 판단하는 기준은 단위 시간당 발생되는 패킷의 수이다.

기존 연구에서 트래픽 발생을 단위 시간당 패킷의 수에 둔 것에 반하여 본 연구에서는 전체 패킷의 수, 총 전송 시간, 전송속도 및 전송량 등 다양한 조건으로 트래픽을 발생시키는 것을 고려하였다.

3. 설계 및 구현

이더넷 패킷은 그림 1 과 같이 이더넷 헤더, IP 헤더, TCP 헤더, 그리고 응용데이터로 구성된다.

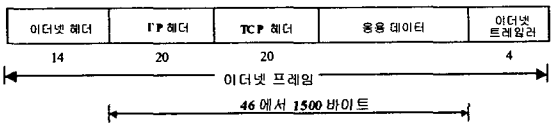

그림 1: 이더넷 프레임의 구조

이더넷 헤더는 목적지주소(6 바이트), 출발지주소(6), 프레임형태(2) 모두 14 바이트를 차지한다. 일반적인 네트워크 프로그래밍 환경하에서는 이 이더넷 헤더를 직접 변경할 수 없다. 그러나 본 논문에서 구현된 VTG 는 이더넷 헤더를 포함한 모든 데이터를 직접 설정하여 패킷을 생성할 수 있다.

본 연구에서 구현한 VTG 는 기본적으로 4 가지 방법으로 트래픽을 발생하도록 설계되었다.

- 패킷수
- 전송시간
- 전송속도
- 전송량

패킷수, 전송시간, 전송량은 절대적인 값이기 때문에 이들 방법으로 패킷을 발생시키는 것은 문제가 되지 않는다. 그러나 주어진 전송속도에 맞게 트래픽을 발생시키기 위해서는 실제 전송되는 속도를 측정하여 목표로 하는 전송속도와의 차이를 시간 지연으로 동기화 하여야 한다. 다음은 전송속도를 동기화 하는 알고리즘이다.

```
do {
   very_last_time = clock(); // 현재 시간
   do {
      process_other_event(); // 속도 동기화를 위한 지연
   } while (clock() < (very_last_time + delay_time));
   total_senet_bytes += SendPacket(); // 1 개의 패킷을 전송
   cur_time = clock(); // 현재 시간
   run_time = cur_time – start_time; // 총 실행 시간 계산
   cur_speed = (total_sent_bytes*8) / run_time // 속도
   if (cur_speed > requested_speed)// 목표 속도와 비교
         delay_time ++; // 속도가 빠르면 지연시간 증가
   else if (cur_speed <requested_speed)
         delay_time--; // 속도가 느리면 지연시간 감소
} while (run_time < requested_time_duration)
```

그 밖에 VTG 를 설계하면서 고려된 사항은 다음과 같다.

- 네트워크 어댑터의 선택: 때에 따라서는 한 컴퓨터에 2 개 이상의 네트워크 어댑터가 존재할 수 있다. 이런 경우에는 사용자가 패킷 발생을 원하는 네트워크 어댑터를 설정할 수 있어야 한다.
- 패킷크기의 설정: VTG 에서는 발생 시키고자 하는 패킷의 크기를 60 바이트에서 1514 바이트 사이에서 고정적으로 또는 무작위(random)로 설정할 수 있도록 하였다.
- 이더넷 주소의 설정: 목적지와 출발지 이더넷 주소를 설정할 수 있도록 함으로써 특정 컴퓨터나 라우터에게 트래픽이 집중될 수 있도록 하였다.
- 인터넷 주소의 설정: 이더넷에서 가장 많이 사용되는 프로토콜인 인터넷프로토콜(IP)을 발생할 수 있도록 인터넷 주소를 설정할 수 있다.
- TCP 또는 UDP 포트의 설정: TCP/IP 패킷을 발생할 때 출발지와 목적지 포트 번호를 설정할 수 있게 함으로써 특정 응용 프로그램에 해당되는 프로토콜을 집중적으로 발생시킬 수 있도록 했다.
- 트래픽 발생 통계의 표시: 트래픽 발생이 진행되는 동안의 경과시간, 패킷수, 총발생된 바이트수 그리고 현재 발생된 트래픽의 속도를 표시할 수 있도록 하였다.

VTG 는 Windows 환경하에서 Visual C++ 와 WinPcap 에서 제공되는 API(application programming interface)를 이용하여 구현되었다. 그림 2 에 VTG 의 구성도를 보여주고 있다.

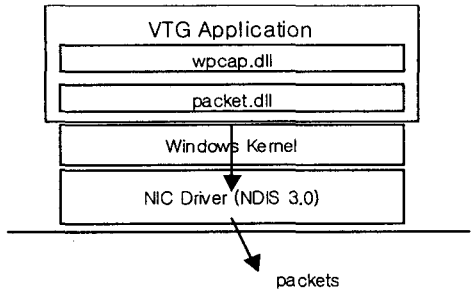

그림 2: VTG 프로그램의 구성

그림 2 에서 VTG 프로그램은 WinPcap[2]에서 제공하는 wpcap.dll 과 packet.dll 을 사용한다. 이들 두 모듈은 다시 Windows 의 Kernel 을 이용하여 네트워크 어댑터에 직접 패킷을 쓰거나 패킷을 모니터링하기 위해 읽어올 수 있다.

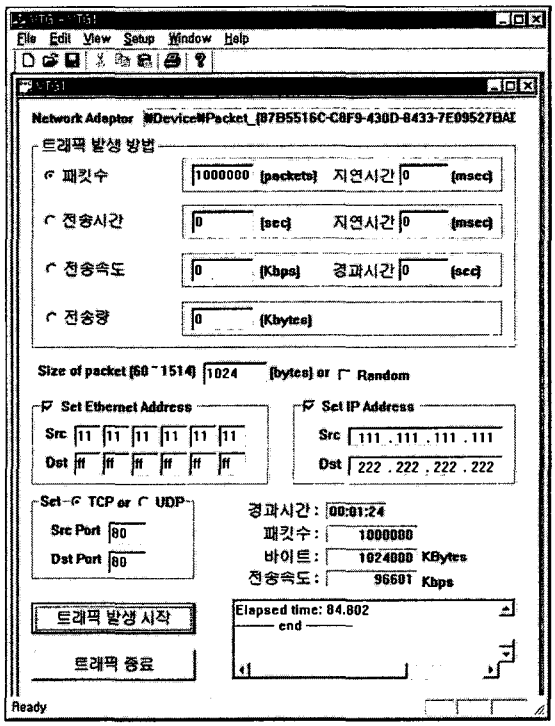

그림 3: VTG 실행 화면

그림 3 은 VTG 의 실행 화면을 보여주고 있다. 모든 조건을 입력한 뒤 [트래픽 발생 시작]버튼을 누르면 트래픽이 발생되며 실행 도중 [트래픽 종료]버튼을 누르면 트래픽 발생이 중단된다. 트래픽이 발생되는 동안 경과시간과 패킷수 그리고 전송 속도들을 표시한다.

4. 실험 및 성능평가

4.1 실험 환경

실험을 위하여 사용된 컴퓨터와 구성도는 그림 4 와 같다.

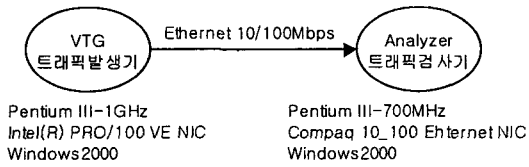

그림 4: VTG 실험 환경

그림 4 에서 트래픽 검사기는 WinPcap 을 이용한 프로토콜 분석기인 Analyzer[6]을 사용하였다. 실험은 VTG 가 수행되는 PC 와 Analyzer 가 실행되는 PC 를 직접 연결하고 통신 속도를 10Mbps 와 100Mbps 로 변경하면서 VTG 의 각각의 항목에 대하여 행하였다.

4.2 실험 1: 패킷 크기의 변경

그림 5 는 패킷 크기를 변화시키면서 1,000,000 개의 패킷을 발생시켰을 때의 평균 속도를 보여준다.

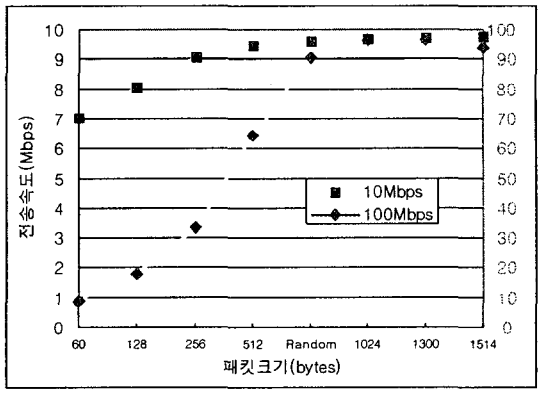

그림 5: 패킷 크기 변화에 따른 전송 속도

그림 5 에서 좌측 눈금은 LAN 환경이 10Mbps 인 경우이고 우측 눈금은 100Mbps 인 경우이다. 패킷크기 중 Random 이란 패킷의 크기를 고정시키지 않고 60 에서 1514 사이의 무작위로 패킷 크기를 지정한 경우이다. 이 경우, 통신 속도는 60~1514 의 평균 값인 약 790 인 경우와 유사한 결과가 나왔다. LAN 속도가 10Mbps 환경인 경우 통신 속도는 60 일때 약 7Mbps 에서 시작하여 최대 9.787Mbps 까지 점진적으로 증가하는 것을 보여준다.

반면에 LAN 속도가 100Mbps 인 경우에는 크기가 60 바이트일 때 최소 8.618Mbps 에서 시작하여 약 800 바이트 까지는 패킷 크기에 비례하여 속도가 같이 증가하는 것을 보여준다. 그러나 패킷 크기가 1024 보다 큰 경우에는 전송 속도의 변화가 크지 않았으며 오히려 1300 바이트를 초과하는 경우에는 속도가 감소하는

현상을 보였다. 그 이유는 1300 바이트보다 큰 패킷을 처리하기 위해서 VTG 프로그램과 Windows 사이에 발생하는 처리부담(버퍼와 버퍼 사이에 복사하는)이 커지고, 프로그램 구조상 윈도우 이벤트 처리를 위한 시간 지연으로 전송 라인을 최대한 점유하지 못하기 때문인 것으로 판단된다.

4.3 실험 2: 지연시간 변경

그림 6 은 패킷의 크기를 1514 최대값으로 정한 뒤 패킷 간 지연시간에 변화를 주었을 때의 발생된 속도 변화를 보여준다.

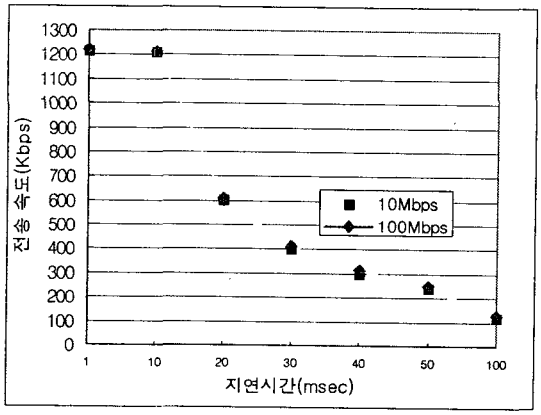

그림 6: 지연시간 변화와 속도(패킷크기=1514)

그림 6 에서 특이한 점은 LAN 속도가 10Mbps 인 경우와 100Mbps 인 경우에 거의 같은 결과를 보여준다는 것이다. 그것은 패킷간 지연시간이 msec 단위로 주어졌기 때문에 LAN 속도에 관계 없이 트래픽 발생기의 시간 지연에만 의존되기 때문인 것으로 해석된다. 즉, 10Mbps 를 가정하였을 때 1msec 안에 처리할 수 있는 데이타의 양은 10Kbit 이고 이값은 약 1.2Kbyte 가 되므로 패킷 크기가 1514 인 경우에 약 1.3msec 에 송신이 이루어지므로 VTG 에서 지연시간을 주는 것이 LAN 의 속도와는 무관하게 일정한 속도로 패킷이 발생된다는 것이다. 이런 이유로 10Mbps 인 경우와 100Mbps 인 경우 모두 같은 속도가 나오는 것이다.

4.4 실험 3: 전송속도 변경

그림 7 은 VTG 에서 전송 속도를 지정했을 때 실제로 발생된 트래픽의 전송속도를 보여준다. 이 실험에서 사용된 패킷의 크기는 최대 크기인 1514 이고 주어진 시간은 각각의 속도에 대하여 10 초이다.

그림은 100Mbps LAN 에서 10Mbps 부터 100Mbps 까지 10Mbps 간격으로 10 초간 발생된 트래픽을 보여준다. 90Mbps 까지는 사용자가 원하는 전송 속도가 얻어졌으나 90Mbps 이상의 전송속도에 대해서는 주어진 조건보다 약간 작은 전송 속도가 얻어질 수 있었다.

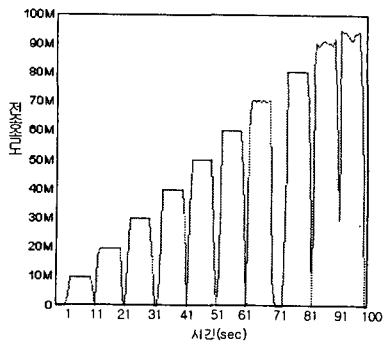

그림 6: 전송속도 변화

5. 결론 및 향후 연구

본 논문에서는 다양한 방법으로 트래픽을 발생할 수 있는 비쥬얼 이더넷 트래픽 발생기 VTG 를 설계하고 구현하였다. 구현된 트래픽 발생기를 이용하여 실제 LAN 상에서 얼마나 효율적으로 트래픽을 발생시킬 수 있는지 검사하였으며 또 이더넷 주소와 인터넷 주소를 지정함으로써 특정 통신장비에 집중적으로 트래픽을 발생시킬 수 있었다.

실험을 통해 얻은 결과 중 특이할만한 점은, 패킷의 크기를 변화하면서 속도를 측정하였을 때, 일정한 크기 이하의 패킷 크기로는 주어진 LAN 의 최대 속도를 낼 수 없다는 것이었다.

본 논문에 이은 향후 연구로는, 기존 연구들에서 제안된 자기 유사성을 갖는 트래픽이 발생될 수 있도록 트래픽 모델링 방법을 보완하고, TCP/IP 가 아닌 다른 프로토콜 및 상위 계층의 프로토콜에 대한 트래픽 발생 방법과 모델링 방법을 연구하는 것이다.

참고문헌

[1] S. McCanne and V. Jacobson, The BSD Packet Filter: A New Architecture for User-level Packet Capture., Proceedings of the 1993 Winter USENIX Technical Conference (San Diego, CA, Jan. 1993), USENIX.

[2] F. Risso and L. Degioanni, An Architecture for High Performance Network Analysis, Proceedings of the Sixth IEEE Symposium on Computers and Communications, pp. 686 – 693., 2001.

[3] J. F. Shoch and J. A. Hupp. Measured performance of an Ehternet local area network, Communications of the ACM, 23(12), pp. 711 – 721, 1980.

[4] W. E. Leland and D. V. Wilson, High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection. In Proceedings of the Infocom 91, pp. 1360 – 1366, 1991.

[5] Walter Willinger, Murad S. Taqqu, Robert Sherman, Daniel V. Wilson, Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level., IEEE/ACM Transactions on Networking 5(1),71-86 (1997)

[6] F. Risso and L. Degioanni, Analyzer: a public domain protocol analyzer, http://anlayzer.polito.it/