Passive Monitoring 기반의 P2P 트래픽 탐지 및 QoS 제어기법

김희준O 한민규 성백동 홍진표
한국외국어대학교
{ icemichyO, hufs96mk, iceboy98, jphong }@hufs.ac.kr

P2P traffic Detecion and QoS Control Algorithm based Passive Monitoring

Hee Joon KimO Minkyu Han, Baek Dong Sung, Jinpyo Hong
Hankuk University of Foreign Studies

요 약

최근 다양한 P2P 프로그램을 많이 사용함에 따라 네트워크에서 생겨나는 트래픽의 상당 부분이 P2P가 발생시키는 트래픽으로 이미 HTTP, FTP의 양을 훨씬 뛰어넘고 있다. 현재 인터넷 환경에서 방화벽을 통과하기 위해 포트번호를 변경하여 통신을 하는 새로운 P2P응용들의 행동들은 전통적인 well-known port 기반의 응용프로그램을 구분하는 단순한 분석 방법만으로 신뢰하기가 어렵다. 새로운 P2P 응용들과 같은 트래픽 모니터링의 정확도를 높이기 위해서는 TCP/IP 헤더만이 아니라 패킷이 담고 있는 페이로드 내용에 대한 조사 차원의 모니터링 방법이 필요하다. 본 논문에서는 TCP/IP 헤더 정보와 더불어 패킷의 페이로드 내용을 조사하여 P2P 트래픽을 탐지하는 모니터링 기법을 제안한다. 이어 탐지되는 P2P 트래픽에 대하여 Linux Netfilter Framework의 Queuing Discipline에서 제공하는 계층적인 우선순위 큐를 사용하여 일정한 양의 대역폭을 할당하는 정책을 적용함으로써 안정적이면서 효율적인 네트워크 운용 방안을 제시한다.

1. 서 론

인터넷이 발달하고 이를 사용하는 사용자 수가 증가함에 따라 많은 네트워크 서비스 또한 개발되고 있다. 이러한 네트워크 서비스 중 P2P 응용 분야가 대표적인 예라고 할 수 있다. 대표적인 인터넷 트래픽 모니터링 회사인 CacheLogic회사의 통계를 보면, 2005년 P2P 트래픽이 차지하는 비중이 작게는 60%에서 약 80%까지 이르고 있는 추세이고 현재는 그 비중이 더해가고 있다.[2][3] 이러한 P2P는 파일-스와핑과 같은 기술발전과 결부되어 제한적/혼잡한 포트를 회피함으로써 대용량 파일과 동적 기술을 공유하게 하여, P2P를 인터넷상의 주도적 프로토콜이 되게 하였고, 이것은 인터넷 서비스공급자 네트워크 총 트래픽의 대부분을 차지하는 결과를 낳았다. P2P는 광대역사용의 수송량을 증가시킴으로써 서비스 공급자에 영향을 미치고, 또 엄청난 트래픽과 P2P 의 대칭적 성격으로 인하여 Edge와 Core 및 Access 네트워크상에 혼잡한 상황을 유발한다. 이것은 가입자 서비스품질의 수준(QoS)에 영향을 미칠 수 있고, 또 연관된 잠재적 가입자 위험과 함께 만약 해결되지 않으면, 고비용의 하부구조 업그레이드 및 네트워크 재분할의 필요성이 대두되는 문제점이 발생한다. 최근 등장하는 새로운 P2P는 동적인 포트번호 사용과 방화벽을 피하기 위한 일반적인 HTTP 80번 포트를 사용하는 기법으로 인해 패킷의 내용을 살펴보지 않고서는 관련된 세션들을 제대로 관리할 수 없을 뿐 아니라 애드웨어나 스파이웨어 등이 Data에 잠식해 있을 경우 이 또한 감지를 할 수 있는 애플리케이션 수준에서 패킷 검사가 필요하다. 애플리케이션 수준에서의 패킷 검사는 특정 P2P 응용이 사용하는 패턴인 시그니처를 탐색해야 한다. 시그니처는 응용의 제어나 패킷 데이터의 종류를 나타내기 위해 사용되는 일정한 패턴이다.[6][7]

본 논문에서는 TCP/IP 헤더 정보만이 아닌 패킷의 페이로드에 포함되어 있는 특정 P2P 응용에 대한 시그니처를 탐색하여 P2P 트래픽 모니터링을 수행할 수 있는 방안을 제안한다. 제안하는 트래픽 모니터링 방법으로 Linux Netfilter Framework 환경[1]의 conntrack 정보를 가지고 흐름 기반의 탐색이 가능하게 설계를 하였다. 또 탐지되는 P2P 트래픽에 Linux Netfilter Framework의 Queuing Discipline에서 제공하는 계층적인 큐를 이용하여 대역폭 조절을 통한 QoS 기법을 제시함으로써 네트워크 관리차원에서의 대응방안을 마련하였다.

2. 관련연구 동향

2.1 Passive / Active Monitoring

패킷에 대한 분석을 하는 모니터링 기법에는 Passive 방식과 Active 방식이 있다. Active 모니터링은 네트워크상의 주기적으로 시험 패킷을 보내어 시험 패킷 또는 그 응답에 대한 퍼포먼스를 측정하는 방식으로 주로 네트워크 성능 평가 측면에서 사용된다. Passive 모니터링은 라우터나 스위치기반의 네트워크나 링크에서 관찰되

는 패킷에 대하여 Splitter 또는 포트 미러링, SPAN 등의 기법으로 모니터링을 하는 방식이다.
본 논문에서는 실제 라우터나 스위치가 위치한 네트워크상의 P2P 패킷을 탐지하기 위하여 네트워크상의 패킷을 관찰하여 수집하는 방법인 Passive 모니터링 기법을 도입하였다.[4][5]

2.2 P2P 트래픽 탐지 동향

2.2.1 국내 동향

- nofree의 CPS(Copyright Protection System)
- Raonet의 SER series router
- 탑레이어네트웍스 (탑인스펙트 딥 패킷 인스펙션)

2.2.2 국외 동향

- IPP2P project
- SourceForget.net의 L7-filter
- Juniper Networks의 IDP(Intrusion Detection and Prevention)
- IETF IPFIX(IP Flow Information Export) WG

2.3 P2P 프로토콜의 특성

2.3.1 P2P 응용 프로그램의 특징

P2P 응용 프로그램은 다음과 같은 특징을 지니고 있다.

- 웹브라우저를 사용하여 일반적인 통신 트래픽 활용
- 유저의 컴퓨터가 클라이언트와 서버의 역할을 모두 수행
- 시스템은 개인이 컨텐츠를 제작하거나 기능을 추가할 수 있는 도구를 제공함
- 범용 프로토콜인 SOAP(Simple Object Access Protocol)이나 XML-RPC등을 지원

표 1 P2P 프로그램별 프로토콜, 포트번호 현황

프로그램	프로토콜	포트번호	L7-patten	Port change
Bittorrent	TCP	6881-6889	유	yes
EDonkey	TCP (UDP)	4661,4662,4223,4242,3000,4771, 4881,2121,7777,6886,5661,4040	유	yes
eMule	TCP (UDP)	4662 (4672)	유	
Gnutella	TCP	6346-6352	유	
Napster	TCP	8888, 7777, 6699	유	yes
vshare	TCP	8401-8404		
프루나 당나귀	TCP (UDP)	4662 (4672)		
구루구루	TCP	9292,9293		
KaZaA	TCP (UDP)	1214, 1285 (1285)		

2.3.2 P2P 트래픽 특징

- Dynamic Ports 사용으로 탐지 곤란
  - 통신을 위하여 통계적으로 정의된 port들을 사용하는 경우(예: KaZaA 2.02버전 이하로는 1214pots로 UDP 및 TCP 프로토콜 사용)
  - Default Port로 80번 port 사용하는 경우 firewall에선 그냥 통과
  - Random Port를 선택하여 사용하는 경우
- 일부 프로그램은 Proxy Server를 경유하여 80번 포트로 터널링하여 방화벽을 통과하기 때문에 탐지 곤란
- P2P응용이 사용자에게 다양한 형태의 암호화 제공

따라서 P2P 애플리케이션은 패킷의 내용을 살펴보지 않고서는 관련된 세션들을 제대로 관리할 수 없을 뿐 아니라 애드웨어나 스파이웨어 등이 Data에 잠식해 있을 경우 이 또한 감지를 할 수 있는 애플리케이션 프로토콜 수준에서 패킷 검사가 필요하다.[6][7]

2.4 P2P 프로토콜의 탐지 방식

P2P application 등 Well-known port를 사용하지 않는 응용 계층 프로토콜은 패킷 헤더 분석을 통해 알아 낼 수 없기 때문에, 패킷 단위로 트래픽 흐름에 대한 식별자(source IP, source port, destination IP, destination port, transport protocol)를 통해 세션이 있는지를 알아내고, 세션 별로 IP defragmentation, reassembly를 수행하여 payload 부분에 있는 응용 데이터를 추출해서 탐지할 수 있어야 한다. 시그니처 기반 탐지는 프로토콜 이례 탐지의 일부로서 수행되는 stateful inspection과 프로토콜 분석을 이용하여 공격 패턴 식별하여 공격에 의해 손상 가능한 통신 상태의 공격만을 조사하여 성능 개선을 도모하고, 규칙기반 탐지는 세션 기반으로 응용 데이터를 추적하여 규칙에 매칭시켜 탐지하는 기법이나 전체 패킷의 흐름을 조사하는 것으로 인해 성능 저하 우려가 있으며, 다량의 데이터 조사 시 오탐율이 발생할 가능성이 증가된다는 단점이 있다. 프로토콜 이례탐지 기법은 공격이 프로토콜 표준으로부터 벗어난다는 사실(잘 알려진 프로토콜(예: http)에 터널링 함으로써 악의적인 공격을 시도)에 기초하여, 알려지지 않은 새로운 공격을 탐지한다. [6][7]
이에 본 논문에서는 P2P 트래픽 탐지는 시그니처 기반, 규칙기반, 프로토콜 이례 탐지와 같은 여러 가지 방법의 탐지에 기초한 하이브리드 탐지 방식을 채택하여 오탐율을 감소시킨다는 의의를 둔다. 또 탐지된 세션에 대해 악의적인 세션(Malicious Session)을 차단하거나 대역폭을 제한함으로써 탐지뿐만이 아닌 트래픽 제어를 통한 성능의 경쟁력을 확보한다.

2.5 Linux Netfilter Framework

2.5.1 Netfilter

Netfilter는 리눅스 커널의 네트워크 스택 내부에 있는 훅(hook)의 집합으로써 패킷이 임의의 훅을 지나갈 때

패킷을 처리하기 위하여 호출되어야 하는 함수를 모듈의 형태로 등록할 수 있도록 구현된 프레임워크이다.

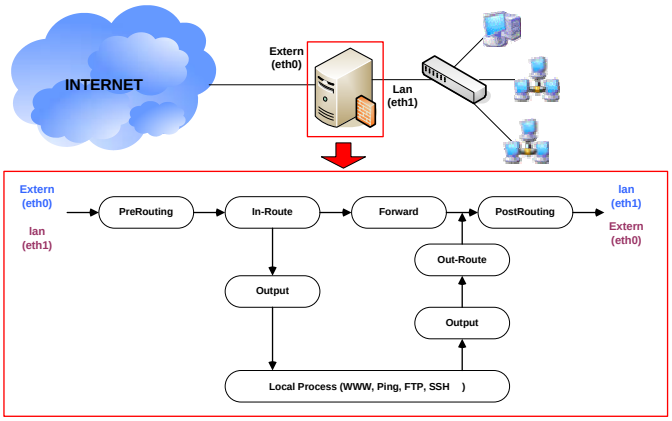

그림 1 Netfilter/Iptables 구조

Netfilter는 IP layer에 전달되는 모든 패킷을 검사하기 위해 다섯 개의 훅으로 구성되면 각 훅의 종류와 기능은 표 2와 같다. [1][8]

표 2 Netfilter 훅의 종류와 기능

훅의 종류	기능
PRE_ROUTING	라우팅되기 전에 패킷을 처리
LOCAL_IN	로컬시스템이 받을 패킷을 처리
FORWARD	포워딩되는 패킷을 처리
LOCAL_OUT	로컬시스템이 보내는 패킷을 처리
POST_ROUTING	보내기 위해 라우팅 되어진 패킷을 처리

2.5.2 Iptables

iptables는 규칙의 집합(rule set)을 정의하기 위한 일반적인 테이블 구조이다. iptables는 패킷을 처리하는 방법에 따라 세 개의 다른 테이블(filter, nat, mangle)로 나눈다. filter는 패킷을 필터링하기 위한 테이블을 의미하며, nat은 패킷의 IP 주소를 수정하기 위한 테이블, mangle은 패킷의 옵션을 변경하기 위한 테이블을 의미한다. 또한, 각 테이블들은 다수의 체인(chain)을 포함한다. 체인의 내부에는 규칙(rule)이 존재하며, 각각의 규칙(rule)은 여러 개의 match와 하나의 target으로 구성된다. match는 규칙의 조건의 나타내고, target은 조건이 성립할 경우 실행하게 되는 행동을 의미한다. [1][8]

2.5.3 Connection Tracking

Connection Tracking은 packet의 connection에 대한 상태 정보를 유지하는 능력을 말한다. 시스템에 iptable이 동작하고 있으면 packet이 통과할 때 tracking을 하는데 이를 위하여 protocol type, source / destination IP address, port number 등의 정보를 이용한다. Linux 2.4.18이후 버전부터 Connection Tracking은 conntrack이라 불리는 framework에서 수행된다. [1][8]

3. P2P traffic 탐지 구조

3.1 P2P traffic 탐지 모델링

그림 2는 전체적인 시스템의 모델을 나타낸 것이다. 모니터링 서버에서 패킷 캡쳐 및 제어를 담당하게 하여 기능을 분리함에 따라 단지 라우터에서는 패킷을 룰에 따라 처리하도록 하여 패킷 캡처에 따른 부담을 없애고자 효율성 측면을 강조한 구성이다.

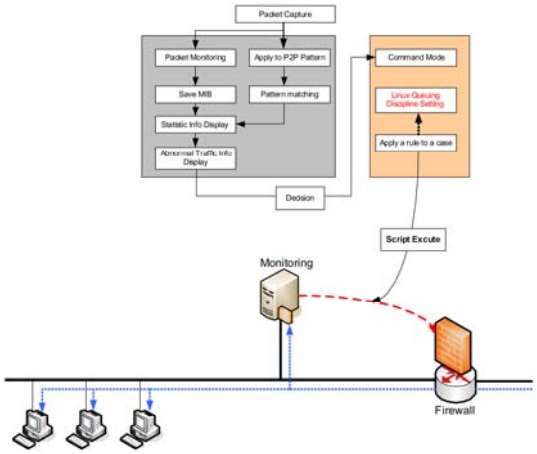

그림 2 전체 P2P traffic 탐지 시스템 구조

3.2 P2P 프로토콜에 대한 시그니처 도출 및 규칙 정의

공개된 정보(protocolinfo.org)를 활용하여 사용하는 transport protocol과 port 번호, application data를 식별하기 위한 regular expression, 기타 식별할 수 있는 정보를 분석하여 P2P 트래픽의 signature의 도출을 하였다. 프로토콜 분석기를 이용한 모니터링 및 분석을 통해 국산 P2P 응용에 대한 프로토콜 분석 및 시그니처 도출을 하였다.[7] …

3.3 Netfilter의 conntrack을 이용한 packet 정보 분석

Netfilter 프레임워크에서 지원하는 기능 중 패킷이 통과할 때 tracking을 하는데 이를 위하여 흐름을 식별하여 패킷의 연결에 대한 상태 정보를 유지하는 conntrack 모듈을 확장하여 커널 모듈 및 시스템콜 기법을 이용한 흐름 기반의 패킷 탐지가 가능하다.

3.4 Netfilter/Iptables 기능 확장

응용계층 수준에서의 프로토콜을 탐지하기 위한 목적으로 regular expression이 사용 가능하게 규칙을 확장하였다. 여러 패킷 흐름이 어떤 P2P 프로토콜을 사용하고 있는지를 탐지하기 위해 응용계층의 데이터를 미리 정의해둔 regular expression으로 매칭을 한다. 매 패킷에 대해 패턴 매칭을 수행하면 자체 장치의 성능을 저하시키기 때문에 앞서 언급한 conntrack 기능을 통해 식별된 connection에 대해서 커널확장모듈로 만든

시스템콜을 호출하여 초기 10개의 패킷만을 regular expression을 적용하게 하였다.[8][13]

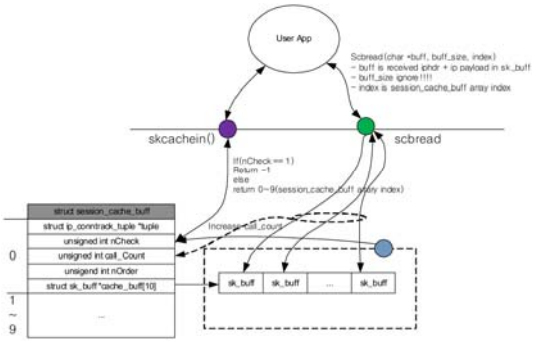

그림 3 conntrack을 이용한 패킷 캡쳐링 모듈 구조

regular expression으로 탐지 불가능한 경우를 위해 "IPP2P"[6]의 접근 방식을 분석하여 채택하여 오탐율을 감소 시켰으며, 세션 관리 모듈을 첨가하여 세션기반의 정보요소(source ip/port, destination ip/port)를 바탕으로 규칙들을 설정한 후 리눅스 Netfilter 프레임워크에서 제공하는 패킷필터링 설정 응용 프로그램인 iptables[8]를 직접 장치 내에 적용 시키는 방식을 도입하여 세션 기반에서의 패킷 제어를 할 수 있는 방안을 마련하였다.

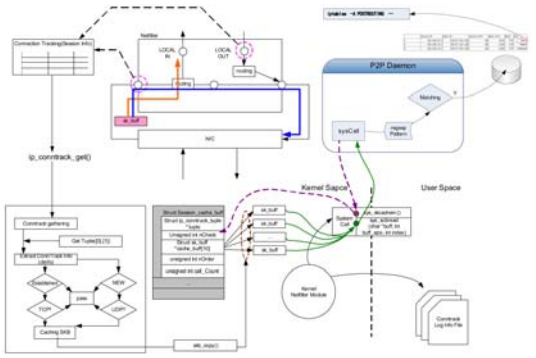

그림 4 Iptables/Netfilter를 확장한 시스템 구조

4. Linux Netfilter Framework의 Queuing Discipline을 이용한 P2P 트래픽 흐름 제어

4.1 계층적인 queue 설정 및 응용에 따른 대역폭을 할당

본 논문에서는 P2P 트래픽 제어를 위한 QoS기법을 좀 더 단순하고 유용성 있게 수행하기 위해 iptables CLASSIFY option과 트래픽 제어 프로그램인 tc를 이용한다. QoS rule을 전송방향에 맞춰 설정을 함으로써 router에 P2P 트래픽이 도착하면 해당 트래픽 클래스에 설정된 룰에 따른 큐를 따라 흐르게 함으로써 P2P 트래픽 전송을 제한한다. 그림3에서 나와있듯이 우선 디바이스에 대해 계층적인 큐(Hierarchical token bucket: HTB)[10]를 적용하는 설정을 하고, 전체 큐를 제어하는 루트 큐에 대한 클래스 ID를 부여한 후 루트 큐를 부모로 하는 여러 큐들을 생성할 수 있다.

```
#Set up the Bandwidth Management for eth 0
#(traffic transmitted TO users coming FROM the WAN )
tc qdisc add dev eth0 root handle 1: htb

# Setup the root class for all Traffic classes
# rate is set to 100 Mbps.
#The 'ceil' ceiling value is omitted,
#and will be set to the same value as the rate (100 Mbps)
tc class add dev eth0 parent 1: classid 1:1 htb rate 100mbit

# The class for limited traffic (P2P traffic)
# rate is set to 1 Mbps. 'ceil' is omitted here also.
tc class add dev eth0 parent 1:1 classid 1:17 htb rate 1mbit
# Alternate example: Provide a minimum of 512 Kbps and set a
maximum of 1 Mbps
#tc class add dev eth0 parent 1:1 classid 1:17 htb rate 512kbit ceil 1mbit
```

그림 5 계층적인 큐를 설정하기 위한 tc 스크립트

그림 6에서 나와있듯이 우선 디바이스에 대해 계층적인 큐(Hierarchical token bucket: HTB)[10]를 적용하는 설정을 하고, 전체 큐를 제어하는 루트 큐에 대한 클래스 ID를 부여한 후 루트 큐를 부모로 하는 여러 큐들을 생성할 수 있다.

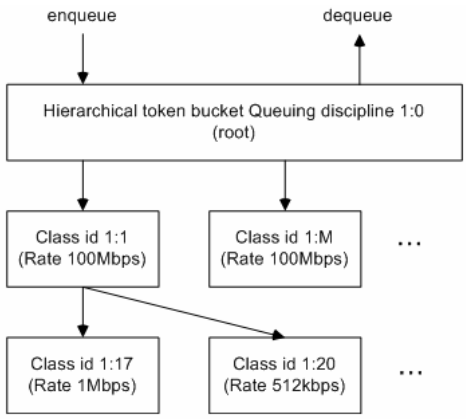

그림 6 HTB 큐 구조

각 큐에는 해당 큐에 대한 ID설정과 전송비율을 설정하여 흐름 별 또는 서비스 및 응용에 따른 클래스 id를 적용시켜 클래스 ID와 매칭하는 큐로 패킷을 흐르게 하는 원리이다. 따라서 서비스 및 응용에 따른 차별적인 대역폭의 할당을 이용한 QoS 메커니즘 적용이 가능하게 하였다.

4.2 규칙 및 정책을 통한 트래픽 흐름 제어

오류! 참조 원본을 찾을 수 없습니다.에서와 같이 P2P 트래픽을 감지를 하고, 감지된 트래픽을 해당 클래스 분류에 따른 계층적인 큐로 흐르게 하기 위해선 앞서 언급한 iptables CLASSIFY 설정이 수반되어야 한다. Iptables에는 규칙의 집합을 정의하기 위한 테이블 구조가 구성되어 있어 규칙의 조건(match)이나 조건이 성립되었을 경우(target) 실행하게 되는 행동을 설정할 수 있다. 테이블은 filter table(패킷을 필터링), nat table(패킷의 IP 주소를 수정), mangle table(패킷의 옵션을 변경)이 있으며 기본적인 명령어 형태는 다음과 같다.

```
iptables [-t table] command [match] [target/jump]
```

Netfilter의 conntrack을 이용한 응용계층 수준의 패킷 페이로드를 regular expression을 검사하여 P2P 트래픽에 대해 규칙을 적용하기 위해서는 해당 세션에 대한 정보를 분석하여 iptables를 이용한 규칙을 적용한다. 이로 인해 동적으로 P2P 세션에 대한 클래스 분류를 하고 이와 매칭되는 큐로 이어지는 트래픽 종류별 흐름 제어가 가능하다.[11]

```
#전송률을 제한하고자 하는 traffic class에 대한 iptables classify option setting

iptables ?A POSTROUTING ?t mangle ?o eth1 -s 220.67.124.138 ?d
192.168.10.2 ?j CLASSIFY --set-class 1:17

#FTP 프로토콜처리 예제

iptables -A POSTROUTING -t mangle -o eth0 -p tcp --sport 20:21 -j
CLASSIFY --set-class 1:11
```

그림 7 iptables CLASSIFY option 적용 스크립트

5. P2P 트래픽 제어 시스템 구조

처음 패킷이 도착했을 시 전체적인 트래픽 모니터링을 위한 정보와 P2P 패턴 매칭을 통한 정보를 보여주고 사용자는 이에 따라 P2P 트래픽을 처리하는 결정을 해주어 라우터에 그 룰을 적용하도록 하는 구조를 설계하였다.

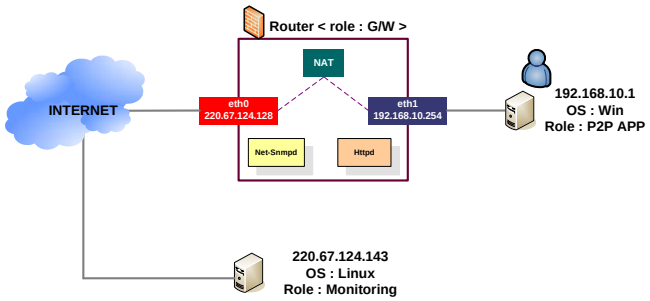

그림 8 테스트베드 환경

그림 8은 모델링을 통한 실제 테스트 베드 환경을 나타낸 것이다. 실제 외부 인터넷과 내부 네트워크를 연결해 주는 라우터에서 룰을 적용하여 패킷을 처리하는 기능을 탑재하였고 내부 네트워크에서는 P2P 응용을 사용하는 사용자를 두었다. 모니터링 서버는 라우터 내부에서 일어나는 패킷 트래픽 상황과 P2P 매치현황을 Net-SNMP[12]와 데이터베이스를 통하여 나타나게 하였다. 라우터에서 일어나는 패킷 트래픽 현황을 모니터링 서버에서 나타낸 결과는 그림 9과 같이 나타내었다. 라우터안의 두 인터페이스에서 발생하는 전체 트래픽양을 Inbound 방향과 Outbound 방향으로 나타내도록 하여 패킷의 트래픽 흐름에 대한 방향을 통해 내부에 트래픽 서버의 존재여부와 P2P 트래픽 발생여부 등의 판단을 사용자가 쉽게 인지할 수 있게 하였다.

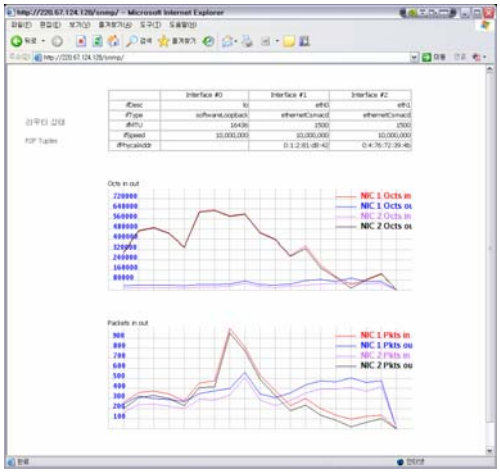

그림 9 Monitoring 화면

또 P2P tuple 항목에서는 그림 10와 같이 라우터내에 흐르는 트래픽 흐름 중 시그니처를 미리 정의하여 매칭된 P2P 응용에 대한 현황을 나타내게 하였다. 현황에는 IP와 Port기반의 정보를 보여 주는것과 동시에 사용자가 선택기능으로 해당 트래픽의 제어여부를 설정할 수 있게 기능을 두었다.

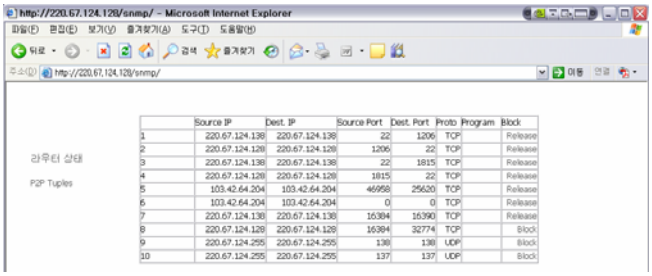

그림 10 P2P Traffic Detection 현황

6. 결론 및 향후과제

본 논문에서는 라우터 기반에서의 P2P 트래픽을 캡쳐하고 이를 미리 정의한 시그니처 기반으로 탐지를 하여 제어를 하는 기법을 제시하였다. 이 방법은 Passive 모니터링을 통한 실제 트래픽 분석 및 제어를 하는 방법에 기반하여 모델을 설계하였고, 로컬 및 일반 네트워크 망에서 인터넷으로 나가거나 들어오는 길목의 트래픽을 모니터링하여 분석 및 탐지가 이루어진다. 또한 이 방법은 실제 트래픽 제어와 분석을 하는 기능을 라우터와 모니터링 서버로 분배하여 Passive 모니터링 방안에서 발생할 수 있는 트래픽 분석을 통해 발생되는 분담을 줄이고자 하였다.

그리고 P2P 응용을 사용하였을 당시의 실제 데이터를 수집하고 분석함으로써 P2P 트래픽을 regular expression을 통해 매칭하는 방안에 대한 적절성도 검증할 수 있었다. 실제 트래픽을 세션 기반의 정보를 바탕으로 QoS 기법의 제어기능을 수행하도록 하여 의미있는 트래픽 제어 솔루션으로써의 제공도 가능함을 보였다.

추가적인 관점에서 보면 유저인터페이스 측면에서 지원할 수 있는 기능이 더 많이 제시되어야 사용자 측면에서 고려할 수 있는 유용성을 더 극대화 할 수 있을 것이다. 또한 보다 다양한 측면에서의 접근하여 무분별하게 증가하는 P2P 트래픽을 탐지하고 제어하는 방안의 기능성을 더 높여야 할 것으로 판단된다.

참고문헌

[1]Klaus Wehrle and Frank Pahlke and Hartmut Ritter and Daniel Muller and Marc Bechler, “The Linux Networking Architecture”, Pearson Prentice Hall, 2004.

[2] 박호진, 박광로, “P2P 기술 동향 및 홈네트워크 응용”, 전자통신동향분석 제21권, 2006.

[3] P2P Detection, http://www.cachelogic.com/

[4] 심원태, “인터넷망 트래픽 분석을 통한 조기예경보”, 제9회 정보보호 심포지움 SIS 2004, 2004

[5] 홍원기, “Internet Traffic Monitoring and Analysis”, 삼성전자세미나, 2003.

[6] IPP2P, http://www.ipp2p.org/

[7] L7-filter, http://l7-filter.sourceforge.net/

[8] Netfilter and Iptables , http://www.netfilter.org/

[9] Nicolas Bouliane, “Writing your own netfilter match”, LinuxFocus article, 2005.

[10] Martin Devera aka devik, “HTB Linux queuing discipline manual”, 2002.

[11] Limiting / Classifying Peer to Peer Traffic, http://www.imagestream.com/

[12] Net-snmp, http://sourceforge.net/projects/net-snmp/

[13] 한동훈. “정규표현식”, 2004