인터넷상에서 트래픽 관리를 위한 효율적인 RTP 패킷 분류 방법☆

An Efficient Online RTP Packet Classification Method for Traffic Management In the Internet

노 병 희*
Byeong-hee Roh

요 약

RTP (real-time transport protocol) 는 인터넷상에서 실시간 멀티미디어 트래픽을 전송하기 위한 유력한 프로토콜로서 간주되고 있다. 망내에서 실시간 멀티미디어 트래픽을 제어하고 관리하기 위하여는 망 관리자가 망을 통하여 전달되는 실시간 멀티미디어 트래픽들을 감시하고 분석해내는 것이 필요하지만, 기존의 트래픽 분석 도구들은 RTP 패킷들을 비실시간 뿐만 아니라 실시간으로도 정확히 분류, 분석해 내지 못하고 있다. 본 논문에서는 인터넷에서 RTP를 사용하는 실시간 멀티미디어 트래픽을 실시간으로 분류해 내기 위한 방법을 제안한다. 한국전산원의 국제망 연동을 위한 게이트웨이 라우터에서 직접 수집한 데이터를 사용하여, 제안 방법의 정확성과 신속성을 보였다.

Abstract

For transporting real-time multimedia traffic, RTP is considered as one of the most promising protocols operating at application layer. In order to manage and control the real-time multimedia traffic within networks, network managers need to monitor and analyze the traffic delivering through their networks. However, conventional traffic analyzing tools can not exactly classify and analyze the real-time multimedia traffic using RTP on the basis of real-time as well as non-real-time operations. In this paper, we propose an efficient online classification method of RTP packets, which can be used on high-speed network links. The accuracy and efficiency of the proposed methodhave been tested using captured data from a KIX node with 100 Mbps links, which interconnects between domestic and overseas Internet networks and is operated by NCA.

Keyword : Traffic Measurement, Internet Management, RTP, Packet Classification

1. 서 론

인터넷이 대표적인 통신수단으로서의 위치를 갖게 됨에 따라, 전자 우편, 파일 전송, 원격 접속 또는 웹 검색등과 같은 전통적인 데이터기반의 서비스 이외에 영상, 음성을 포함하는 다양한 실시간 멀티미디어 서비스가 개발, 보급되고 있다. RTP (real-time transport protocol)[1]는 인터넷상에서 실시간 멀티미디어 트래픽을 전송하기 위한 유력한 프로토콜로서 간주되고 있으며, 인터넷 전화, 화상 회의등과 같이 표준화 기구에서 제시되는 응용들을 포함하는 실시간 멀티미디어 서비스 제공을 위한 다양한 응용 분야를 갖는다[2, 3]. 인터넷이 확장됨에 따라 RTP를 사용하는 실시간 멀티미디어 트래픽의 양도 점차 증가할 것으로 기대된다[4].

그러나, 인터넷 트래픽의 지속적인 증가는 인터넷상에서의 자원 부족과 폭주를 유발하게 되어, 지연 또는 시간에 있어서 엄격한 품질 (Quality of Services, QoS) 보장이 요구되는 실시간 멀티

* 정 회 원 : 아주대학교 정보통신전문대학원 부교수
bhroh@ajou.ac.kr(제 1저자)
☆ 본 논문은 과학기술부 목적기초연구 (R05-2002-000-00829-0) 지원으로 수행되었음.

미디어 관련 응용 서비스 제공에 심각한 영향을 주게 되었다. 실시간 멀티미디어 서비스의 품질 향상을 위한 단말 차원에서의 제어 방법들이 제시되어 있지만[4], 망차원에서의 지원 없이는 품질 향상에 한계가 있다. 망내에서 실시간 멀티미디어 트래픽을 제어하고 관리하기 위하여는 망 관리자가 망을 통하여 전달되는 실시간 멀티미디어 트래픽들을 감시하고 분석해내는 것이 필요하다. 예를 들어, H.323[2]에 기반을 둔 인터넷 전화의 경우에 있어서, H.323 존(zone)의 설계, 게이트웨이(gateway) 설치 위치 및 게이트웨이간 라우팅 경로의 선정, 게이트키퍼(gatekeeper)에서의 연결 수락 제어(admission control)등과 같은 연결 관리를 위하여는 인터넷 전화 서비스 도메인내에서의 인터넷 전화 응용 프로토콜들의 트래픽에 대한 정보가 필요하다.

인터넷에서 트래픽을 감시하고 분석하기 위한 많은 도구들이 제공되고 있으나[5-9], 이들은 RTP를 사용하는 실시간 멀티미디어 트래픽을 다른 패킷들과 실시간 또는 비실시간으로도 정확히 분류해 내지 못하고 있다. 이것은 이들 트래픽 분석 도구들이 응용 계층의 프로토콜 구분을 위하여 트랜스포트 계층 데이터그램 헤더내의 포트 번호를 사용하는데, FTP, Telnet등과 같이 고유한 포트번호가 부여된 이른바 well_known_port 번호를 사용하는 경우에는 분류가 가능하나, RTP와 같이 임의의 포트 번호를 사용하는 경우에는 정확한 분류를 해내지 못하기 때문이다. 따라서, 이들 도구들로 부터의 정보를 활용하여 시간 또는 지연등에 있어서 엄격한 품질 요구를 갖는 실시간 멀티미디어 서비스의 품질 향상을 위한 실시간 인터넷 자원 관리에 적용하는 것은 적절하지가 못하다. 이것은well_known_port로 지정된 응용 프로토콜들은 대부분이 시간 또는 지연등의 품질에 민감하지 않은 것들이고, 실시간 멀티미디어 응용 서비스 제공에 많이 활용되는 RTP는 well_known_port를 사용하지 않으므로[1] 이를 구분해 내지 못하기 때문이다.

본 논문에서는 인터넷에서 RTP를 사용하여 전달되는 패킷들을 실시간으로 분류해 내기 위한 방법을 제안하고, 이의 구현에 대하여 설명한다. 제안 방법의 정확성과 신속성을 보이기 위하여 한국전산원의 국제망 연동을 위한 게이트웨이 라우터에서 직접 수집한 데이터를 사용하였으며, 실제 인터넷 환경에서도 적용하여 보았다. 그리고, 제안된 방법의 적용 예로서 인터넷 전화 서비스의 망 관리에의 적용 가능성을 보여준다.

본 논문의 구성은 다음과 같다. 제2장에서는 제안하는 실시간 RTP 패킷 분류를 위한 기본 방법을 설명하고, 제3장에서는 제안 방법의 성능 실험 결과를 보인다. 그리고, 제4장에서는 본 논문의 결론을 기술한다.

2. RTP 패킷의 실시간 분류 방법

RFC 1889[1]는 RTP 패킷 수신측에서 다른 응용들과의 혼동을 피하기 위하여 수신 패킷들의 RTP 헤더 유효성을 검사하여 RTP 패킷들을 분류해 내기 위한 지침을 제공한다. 이 헤더 유효성 검사는 극히 적은 수의 연결을 구성하는 최종 수신 사용자(receiving end-user side)에 초점이 맞추어져 있으므로, 경유하는 패킷들과 연결들의 수가 상대적으로 무한히 큰 망내의 중간 라우터들에 직접 적용하는 것은 한계가 있다. 본 절에서는, RFC 1889에서의 헤더 유효성 검사를 확장한 RTP 패킷의 실시간 분류 방법에 대하여 설명한다.

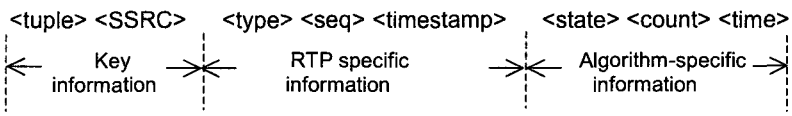
네트워크 장치에 도착하는 패킷들은 IP와 TCP/UDP 헤더내에 있는 프로토콜 필드와 포트 번호들을 조사하여 이들 패킷들이 알려진 응용 프로토콜을 사용하는지를 검사한다. 이때 이들이 알려진 응용 프로토콜을 사용하지 않는 경우, 기존 트래픽 분석 도구들은 이들을 unknown 프로토콜 종류로 구분한다. 대신에, 본 논문에서 제안하는 방법에서는 이러한 unknown으로 분류된 패킷들이 RTP를 사용하는 실시간 멀티미디어 응용 프로토콜을 사용하는지를 검사하여 구분해 낸다. 이

러한 RTP 패킷의 분류를 위하여, (그림 1) 에 나타낸 바와 같은 필드들을 갖는 RTP 패킷 분류 (RPC, RTP Packet Classification) 테이블을 사용한다. RPC 테이블에는 RTP를 사용하는 연결들에 대한 정보를 관리한다.

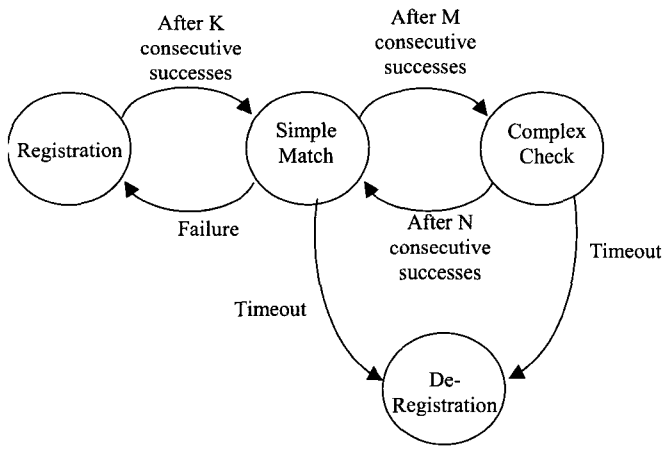
그림 1에서 <tuple> 과 <SSRC>는 RPC 테이블을 검색하기 위한 주요 정보들이다. <tuple>은 송신측과 수신측의 IP 주소, 송신측과 수신측 포트번호들로 구성되며, <SSRC>는 RTP 세션 연결에 대한 SSRC(synchronization source) identifier를 나타낸다. <tuple> 과<SSRC>는 RTP 세션이 유지되는 동안은 변하지 않는 요소들임에 주의한다. <type>, <seq> 그리고 <timestamp>들은 한 RTP 세션내에서 전달되는 패킷들에서 변화되는 부분들로서 각각 RTP 헤더내의 PT(payload type), sequence number, timestamp 필드들에 해당한다. 마지막으로, RTP 패킷 분류 알고리즘에 사용되는 정보로서 <state>, <count>, <time>들이 있으며, 이들중 <state>와 <count>의 역할에 대하여는 다음에서 (그림 2)를 설명하는 과정에서 자세히 표현되고, <time>은 RTP 세션상에서 RTP 패킷들이 도착시마다 이에 대한 도착시간이 기록된다.

그림 2는 각 RTP 세션을 대상으로하여 본 논문에서 제안하는 RTP 패킷 분류를 위한 상태 다이어그램을 보여준다.

well_known 포트 번호가 아닌 짝수의 포트 번호를 갖고 도착한 UDP 패킷에 대하여, 이 패킷에서 대한 주요 정보인 <tuple> 과 <SSRC>에 해당하는 부분을 추출하여 RPC 테이블에 해당 정보가 존재하는지를 검사한다. 이 패킷이 RTP 세션으로 부터의 첫번째 패킷이라면 RPC 테이블 내에는 주요 정보가 존재하지 않음에 주목한다. 이러한 경우에 이 패킷에 대하여 RFC 1889에 규정된 약한 유효성 검사(weak validity check)를 수행하여, 검사 결과가 성공적으로 판단되면 이 패킷에 대한 주요 정보를 RPC 테이블에 새로 삽입하고, 이 연결에 대한 <state>는 Registration



〈그림 1〉 RPC 테이블 구성 필드들



〈그림 2〉 각 RTP 세션에 대한 RTP 패킷 분류를 위한 상태 변화 다이어그램

상태로 기록한다. 검사 결과가 실패한 경우에는 이 패킷은 unknown protocol을 사용하는 패킷으로 처리한다. 이를 수행하는 과정에 대한 가상 코드를 다음에 나타내었다. 이 과정에서 이루어지는 유효성 검사 함수를 weak_validity_check( )로 이름지었고, 수행 내용은 다음과 같다. 즉, RFC1889에서와 같이 우선 UDP 헤더 다음의 데이터에서 처음 2비트 값이 현재 RTP 버전값인 2인지의 여부와 데이터의 길이가 RTP CC 필드 부분을 고려하였을 때의 길이 보다 큰지의 여부를 판단하게 된다.

```
check whether the information asso-
ciated with packet's key
information is kept in the RPC table.

if ( it does not exist in the RPC table )
   do weak_validity_check().
   if ( succeed in weak_validity_
        check() )
      register the packet's <tuple>,
       <SSRC>, <seq>, <type> and
      <timestamp> with <state>=
       Registration, <count>=1 and
      <time>=receiving time in the
       RPC table.
   else
      classify the packet into un
       known protocol.
   endif
endif
```

도착한 패킷의 주요 정보가 RPC 테이블에 존재하는 경우에는 이에 해당하는 <state>에 따라 적절한 유효성 검사를 수행한다. 그림 2에 보인 바와 같이, <state>에는 Registration, Complex Check, Simple Match들중의 하나가 기록된다.

첫째로, <state> 필드가 Registration인 경우에는 다음과 같은 방법이 적용된다.

```
do complex_validity_check_in_reg().
if ( succeed in complex_validity_check_
  in_reg() )
   <count> = <count> + 1.
   if ( <count> > K )
      <state> = Simple_Match.
      <count> = 1.
   endif
   update <seq>, <timestamp>, <type>
    and <time>.
else
   classify packets pending in
    REGISTRATION into unknown class.
   reset the connection, and restart
    in REGISTRATION.
endif
```

함수 complex_validity_check_in_reg()는 도착한 패킷의 UDP 헤더 다음의 데이터들에 대하여 RTP 헤더의 각 필드에 해당하는 비트들이 RPC 테이블에 기록되어 있는 상황과 일관성있게 일치하는지를 검사한다. 즉, 함수 complex_validity_check_in_reg()는 함수 weak_validity_check()에서 수행되는 검사와 함께 이전 패킷으로부터 구하여 기록된 <SSRC>, <type>, <seq>, <timestamp>에 대한 일관성을 검사한다. 좀더 정확히 설명하면, 이전의 패킷과 비교하였을 때 <SSRC>는 변하지 않아야 하고, <seq>는 증가하는 값이어야 하며, <type>에 따라 <seq>와 <timestamp>는 일관성있게 증가하여야 한다. RTP 세션으로부터 처음 도착한 패킷에서 K개의 연속해서 도착한 패킷들이 이들 유효성 검사에 성공한 경우에 이후부터 해당 <tuple>과 <SSRC>를 갖고 도착하는 패킷들은 모두 RTP 패킷들로 분류된다. 즉, RTP 패킷으로 판단되기 위하여 해당 세션에 대하여는 최소 K개의 패킷들이 검사되어야 한다.

Registration 상태에서의 유효성 검사들에 의하여 해당 세션이 RTP를 사용함이 판단된 후에는 그림 2에서와 같이 Simple_Match와 Complex_Check의 두개의 상태들이 교대로 적용된다. 이때, 이들 상태들의 지속되는 횟수를 M과 N의 파라메터로서 나타내기로 한다.

Complex_Check 상태에서는 다음의 가상 코드

로 적은 과정이 적용된다. 이 과정에서 적용되는 유효성 검사를 수행하는 함수 detailed_validity _check() 에 대하여는 다음절에서 설명하도록 한다.

```
do detailed_validity_check().
if ( succeed in detailed_validity_
  check() )
    classify the packet into RTP.
    <count> = <count> + 1.
    if ( <count> > N )
       <state> = Simple_Match.
       <count> = 1.
    endif
    update <seq>, <timestamp>, <type>
      and <time>.
else
    classify the receiving packet into
      unknown.
    remove the packet's key information
      from the table.
endif
```

Simple_Match 상태에서는, 다른 유효성 검사 없이 해당 <tuple>과 <SSRC>를 갖는 패킷들은 모두 RTP로 분류하고, RPC 테이블내의 각 필드들에 대한 정보만을 갱신한다. Complex_Check와 Simple_Match 상태에서는 각각 N과 M회의 성공이 있은 후에는 다른 상태로 천이하면서 <count>값은 모두 1로 재설정하게 된다.

RTP 세션의 종료에 대한 상황을 중간 라우터들에서 알수는 없으므로, 이에 대한 처리를 위하여 RPC 테이블을 주기적으로 정리하는 시간 초과 방식(time-out mechanism)을 적용한다. 즉, 매 RPC 테이블 정리 시간마다 매우 오래된 <time> 정보를 갖는 엔트리들, 즉, 정리시의 시간과 해당 엔트리의 <time> 의 차이가 특정 임계값(Th)을 초과하는 엔트리들은 RPC 테이블에서 제거된다.

RTP 세션이 분류되면, 해당 RTCP 패킷들도 쉽게 분류가능하다. 이것은 RTCP 세션은 RTP 세션에서 사용되는 포트 번호에 1을 더한 값이 사용되기 때문이다[1].

3. 실험 결과

실험은 Windows 2000 OS를 사용하는 500 MHz Pentium III 컴퓨터에 본 논문에서 제안한 방법을 구현하여 수행하였다. 실험을 위하여 국내와 국제 인터넷 망을 연결하여 주는 KIX (Korea Internet eXchange) 라우터에서 실제로 수집한 트래픽 데이터를 사용하였다[13]. KIX 라우터들은 한국전산원[8]에서 운용하고 있다. 패킷 캡쳐는 24시간동안 3회에 걸쳐서 측정되었고, 패킷량은 평균적으로 1.2 Mpackest/sec 이고, 트래픽양은 평균 90 Mbits/sec이다[13]. 실험을 위하여 전체 캡쳐된 데이터중에서 UDP를 사용하면서, 포트 번호가 1023보다 큰 패킷들만을 선별하여 실험 데이터를 새로 구성하였다. 선별하여 새로 구성된 패킷들의 순서는 변하지 않도록 하였다.

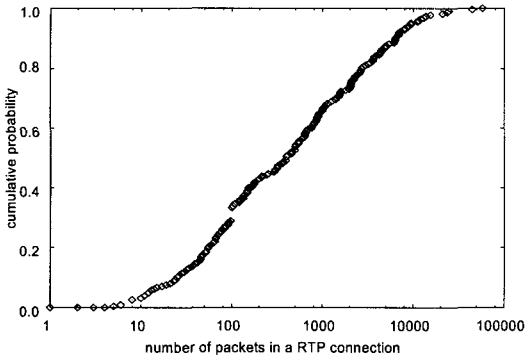
본 논문에서 제안하는 방식의 정확성과 효율성을 보이기 위하여, 실험 데이터에서 RFC 1889에서 정의한 정밀 유효성 검사와 일부 수작업에 의한 경험적 실험에 의하여 RTP를 사용하는 것으로 추정되는 세션과 패킷들에 대한 정보를 추출하여, 이들에 대한 정보를 사전에 저장하였다.

3.1. 파라메터의 선정

여기에서는 본 논문에서 제안하는 방식에서 사용되는 (K, N, M)과 함수 detailed_validity_ check() 에서 적용될 RTP 필드들을 결정하는 것에 대하여 설명한다.

우선적으로, 파라메터 K에 대한 적절한 값을 선정하기 위하여, 그림 3과 같은 사전에 정밀 추출한 정보로부터 각 RTP 세션 연결당 전달된 패킷들의 수에 대한 누적 확률 분포를 구하였다. 그림 3으로부터, 캡쳐한 데이터들에서 나타나는 대부분의 RTP 세션 연결들은 5개 이상의 패킷들을 전송하고 있음을 알수 있다. 또한, 트래픽 제어의 관점에서, 한 세션내에서 다수의 패킷들의 전달이 이루어지는 이른바 장구간(long-lived) 트래픽이

더 고려할 가치가 있다. 이러한 사항들로부터, 파라메터 K에 대한 적정값으로서 5를 선정하였다.

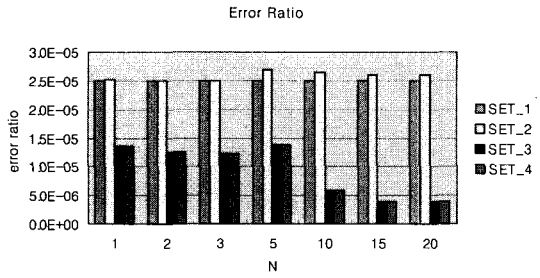


〈그림 3〉 RTP 세션내의 패킷수에 대한 누적 확률 분포

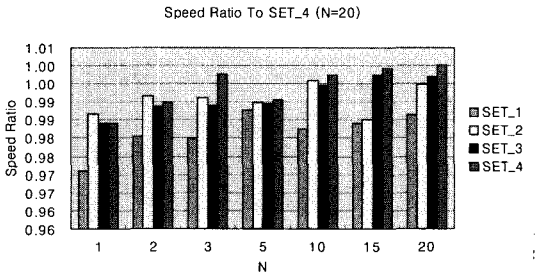
다른 파라메터들인 (N, M)과 detailed_validity _check() 검사에 사용할 RTP 필드들의 선정을 위하여 다음과 같은 사항들을 고려하였다. 즉, 통신망에서의 실시간 트래픽 제어를 위하여 사용되는 방법들은 수십 Mbps에서 Gbps이상까지의 매우 큰 트래픽을 실시간적으로 처리하여야만 한다. 따라서, 본 논문에서 제안하는 방법도 이러한 요구를 만족시킬 정도로 고속으로 RTP 패킷들을 분류해내어야만 한다. 이러한 관점에서 수용할수 있는 정확성을 갖으면서 분류의 속도를 최대한 빠르게 하는 것이 실시간 망관리를 위하여 요구되는 가장 중요한 요소들중의 하나가 된다.

특히 파라메터 N은 Complex_Check 상태에서 RTP 패킷들의 각 필드들의 일관성 여부를 판단하므로 검사에 적용할 필드들의 수에 따라 정확도와 속도가 달라질 수 있다. 따라서, N값의 결정은 검사에 적용할 필드들에 대한 선택과 함께 고려되어져야 한다. RTP 세션내에서 <tuple> 과 <SSRC>의 정보는 변하지 않으므로, Registration 과정을 통하여 RTP 세션으로 분류되면, RTP 패킷들을 구분하는 것이 Registration 과정에서의 complex_validity_check_in_reg()에서 보다 훨씬 더 단순하게 하여 분류 속도를 더욱 빠르게 할수 있다. 즉, Registration 이후의 Detail_Check 상태에서 적용되는 분류를 위한 함수인 detailed_ validity_check()에서는 RTP 패킷 분류를 위하여 RTP 헤더내의 모든 필드들을 검사할 필요가 없게 된다. 이때 어떠한 필드들을 사용하는 것이 정확성과 신속성면에서 유리한지를 비교하기 위하여 <표 1>과 같은 4개의 파라메터 집합들에 대하여 실험을 수행하였다. SET_4는 Registration 상태에서의 함수인 complex_validity_check_ in_ reg()에서 사용되는 파라메터들과 동일하다.

그림 4는 표 1의 각 SET들에 대하여 파라메터 N과 M을 변화시켜 가면서 실험한 제안 방법의 정확성과 신속성에 대한 결과를 보여준다. 정확성은 사전에 정밀 추출한 결과와 본 논문에서 제안한 방법을 사용하였을 경우에 대한 결과를 비교한 것이다.



(a) 오류율



(b) 상대 소요시간

〈그림 4〉 신속성 및 정확성 실험 결과

〈표 1〉 Detail Check 상태에서 적용을 위한 파라메터 실험 집합

구분	검사에 적용할 파라메터들
SET_1	VER
SET_2	VER, PT
SET_3	VER, PT, SEQ, TS
SET_4	VER, PT, SEQ, TS, P, CC, X

(VER: version, P:padding, CC: CSRC count, PT: payload type, SEQ:sequence number, TS:time stamp, X:extension)

그림 4(a)는 N 값이 1, 2, 3, 5, 10, 15, 20과 같을 때, 각 N값에 대하여 M값을 10, 15, 20, 25, 30, 35, 40, 45, 50로 변화시켜가면서 구한 평균 오류율에 대한 결과를 보여준다. 오류율은 다음과 같이 구해진다. 모든 캡쳐한 패킷들에 일련 번호를 부여하고, 사전에 정밀하게 추출한 정보를 사용하여 각 패킷들이 RTP 에 해당하는지를 기록하였다. 그리고, 제안한 방법을 적용하였을때도 마찬가지로 모든 캡쳐한 패킷들이 RTP로 분류되었는지의 여부를 기록하였다. 이로부터, 다음과 같이 오류율을 계산하였다.

$$\text{오류율} = \frac{(\text{Type1 오류 패킷수}) + (\text{Type2 오류 패킷수})}{\text{총 RTP 패킷수}}$$

여기에서 Type1 오류 패킷수는 RTP 패킷을 RTP 패킷이 아닌것으로 분류한 경우에 대한 것이고 Type2 오류 패킷수는 반대의 경우에 대한 것이다. 그림 4(a)에 보인바와 같이 N이 증가할수록, SET_4로 갈수록 오류율이 약간씩 줄어들고 있다. 그러나, N이 증가함에 따른 오류율의 감소는 매우 미미하게 나타나고 있으며, 모든 집합들에 대하여 정확성이 99%가 넘는 성능을 보여준다.

그림 4 (b)는 그림 4 (a)의 실험에서 소요된 평균 시간을 보여준다. 그림 4 (b)에서의 시간 값들은 N=20이고 SET_4를 사용하였을 때 걸린 시간을 1로 하였을때의 상대값으로 나타내었다. N이 증가할수록 Detailed_Check 상태가 많아지므로 시간이 더 소요되고, SET_1보다는 SET_4가 더 많은 파라메터를 검사하므로 시간이 더 소요됨을 볼 수 있다.

그림 4로부터, SET1과 작은 값의 N을 사용하여도 다른 SET들과 N값들에 비하여 많은 차이가 나지 않고, 수용가능한 정확성을 갖게 됨을 알수 있다. 이러한 사실로부터, 파라메터 (N,M)값으로서는 (1,15)를 선정하였고, 검사할 RTP 필드로서는 SET1에 정의된 필드를 사용하였다. 더 나은 정확성을 얻기 위하여는, N 값을 증가시킬수 있으나, 그림 4에서 보듯이 이에 따른 정확성의 향상은 그리 크지 않다.

파라메터들 (K,N,M)을 (1,∞,0)로 하는 것은, unknown 포트 번호를 사용하는 연결로부터 처음 도착한 패킷이 Registration 과정을 통하여 RPC 테이블에 등록된후, 모든 패킷들에 대하여 함수 detailed_validity_check()만을 사용하여 검사하는 것과 동일하다. 이러한 방법이 매우 단순하고 고려가능한 것과 같이 보이나, 이것은 RTP로 분류되는 패킷의 개수가 SET4를 적용하였을 때의 경우에 비하여 3.7배이상 더 많게 되었고, 이에 따라 오류율도 매우 커지게 되는 결과를 얻을 수 있었다.

4.2. RTP 트래픽 분류 실험 결과

표 2는 파라메터 (K,N,M)을 (5,1,15)로 하여 적용한 경우에 대한 실험 결과로서, 모든 실험 데이터를 완료하였을때의 소요된 시간을 포함하는 주요 통계 값들을 보여준다.

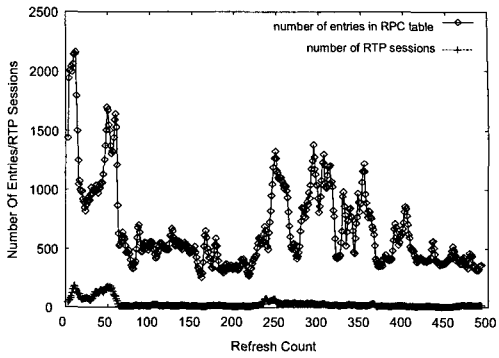
〈표 2〉 실험 데이터에 대한 적용 실험 결과

항 목	결 과
총 패킷 수	11.493 x 106 패킷
RTP로 분류된 패킷 수	1.097 x 106 패킷
분류 정확성	98.4 %
패킷당 평균 알고리즘 소요 시간	1.07 x 10-6 초/패킷 (= 9.33x105 패킷/초)
RTP 가능성 있는 패킷당 평균 알고리즘 소요 시간	8.58 x 10-6 초/패킷 (= 1.16x105 패킷/초)

표 2에서 분류 정확성값은 RTP로 분류하기 이전 상태인 Registration 과정동안의 K개의 패킷들이 제외된 값이다. 그림4 (a)에서는 이들 Registration 상태동안의 K개의 패킷들도 RTP로 분류된 것으로 하여 계산되었다.

그림 5는 RPC 테이블 갱신 시간 간격을 180초로 하였을 때, 매 RPC 테이블 갱신 시간에서 RPC 테이블내의 엔트리들의 수와 RTP 세션의 수의 변화를 보여준다. 평균적으로 존재하는 RPC 테이블내의 엔트리수는 약 626.4개이고, RTP 세션의 수는 약 27.8 개가 되었다. 본 논문에서의 RPC 테이블내의 엔트리 검색을 위하여 이진 탐색(binary search) 방법을 사용하였다. 앞에서도 서술한 바와 같이, 캡쳐한 데이터내에서 RTP를 사용하는 패킷들의 비율은 매우 작고, 이에 따라 RPC 테이블내에 저장되는 엔트리의 수와 RTP 세션 수도 매우 작고, RTP를 사용하는 트래픽이 많아질 경우 그림 5에 나타낸 값들은 매우 커지게 될 것이고, 이에 따라 처리 속도가 표 2에서 보인 결과에 비하여 다소 감소할 것으로 예상할 수 있으나, 이것은 본 논문에서 사용한 이진 탐색 방법대신에 다른 탐색 방법들을 사용함으로써 해결이 가능할 것으로 판단된다. 그러나, 다른 탐색 알고리즘을 적용하여 비교하여 보는 것은 탐색 알고리즘의 성능에 관한 문제가 될 뿐, 본 논문에서의 RTP 분류 방법의 정확성과 신속성에는 영향을 주지는 않게 되어, 본 논문에서의 초점과 벗어나므로 여기에서는 다루지 않는다.

표 2에서의 정확성 98.4 %는 국제 관문 라우터에서의 데이터 수집에 의한 것이므로, 매우 다양한 형태의 데이터가 존재하여 이상적인 RTP 형태를 정확히 매칭시키지 않는 경우가 존재하는데서 나타나고 있다. 실제로 RTP 패킷들을 생성하는 PC들과 RTP 이외의 다른 TCP 또는UDP 패킷들을 생성 시키는 PC들로 내부적으로 구성한 실험망에서 실험을 하였을때는 RTP 각 세션당 Registration 과정에서의 5개 패킷을 제외한 모든 RTP 패킷들을 100% 정확히 분류해 냄을 확인하였다.



〈그림 5〉 RPC 테이블내 엔트리수 및 RTP 세션수의 변화 (refresh interval=180초)

본 논문에서 제안하는 방법은 패킷의 헤더만을 대상으로 하므로 패킷의 길이와 무관하게 동일한 성능을 얻을수 있음에 주의한다. 실험으로 사용한 데이터에서 1023보다 큰 포트 번호를 사용하는 UDP 패킷들의 평균 크기는 329.4바이트이었다. 이로부터, 순수하게 1023보다 큰 포트 번호를 사용하는 UDP 패킷들만을 대상으로 할 경우에 본 논문에서의 방법은 약 2.46Gbits/sec의 고속 링크에 대하여도 적용 가능함을 알 수 있다. 또한, RTP로 분류된 패킷들의 평균 크기는 149.8 바이트 이었고, 모든 대상 패킷들이 RTP인 경우에도 140 Mbits/sec의 링크에 대하여 적용 가능하게 된다. 현재 대부분의 인터넷 패킷들중에서 TCP를 사용하는 패킷들의 빈도가 94%정도 되고, UDP는 6% 정도이며, 이들 UDP 패킷들중에서 RTP 패킷의 비율은 약 4%정도이다[13]. 따라서, 모든 트래픽을 대상으로 할 경우에 본 논문에서의 제안 방법은 2.5Gbps 보다 더 고속의 링크에서도 충분히 적용 가능할 것으로 기대된다. 또한, 본 실험을 수행한 컴퓨터가 500MHz급의 Pentium-III 컴퓨터인 점을 감안할 때 더 빠른 CPU를 갖는 컴퓨터에서 수행시 더 빠른 고속 링크에서도 적용이 가능할 것이다.

4. 결 론

본 논문에서는 RTP 패킷을 실시간으로 분류하

는 방법을 제안하였다. 제안된 방법의 효율성과 정확성은 한국전산원에서 운영하는 국내와 국외 인터넷망을 연결하는 100Mbps의 링크 속도를 갖는 KIX의 노드에서 캡쳐한 데이터를 사용하여 검증하였다. 제안된 방법은 장기간 생존형의 스트림 형식의 알려지지 않은 포트 번호를 사용하는 프로토콜들을 분류하는데 확장 적용 가능할 것으로 판단된다.

RTP는 유선 인터넷 뿐만 아니라 무선 인터넷 환경에서 실시간 멀티미디어 데이터를 전달하기 위한 프로토콜로서 여겨진다. 따라서, RTP 트래픽을 실시간으로 분류, 분석해 내는 것이 인터넷상에서 실시간 멀티미디어 서비스 트래픽을 제어하고 관리하는데 있어서 가장 중요한 문제들중의 하나가 될 것으로 생각한다. 예를 들어, 본 논문에서 제안한 방법을 기존의 망관리 체계에 부가적인 정보로서 제공하게 함으로써, H.323 기반의 서비스 구조 설계, 실시간 멀티미디어 스트리밍 서비스를 위한 프록시 서버들의 설치와 같은 서비스 구조 설계시 활용 가능하고, 특정 라우터에서 RTP를 사용하는 실시간 멀티미디어 트래픽의 품질 향상을 위하여 이들 패킷들을 추출하여 우선 순위 제어와 같은 트래픽 제어를 적용하는데 활용 가능할 것으로 판단된다. 이와 같은 본 논문에서 제안한 방법을 망관리와 트래픽 제어에 활용하는 것은 더 연구가 되어야 할 것이다.

참 고 문 헌

[1] H.Schulzrinne, S.Casner, R.Frederick, V. Jacobson, "RTP : A Transport Protocol for Real-Time Applications", IETF RFC1889, Jan. 1996.

[2] ITU-T Recommendation H.323, "Packet-based Multimedia Communication Systems"

[3] J.Rosenberg, H. Schulzrinne, G.Vamarillo, A.Johnston, J.Peterson, R. Sparks, M.Handley, E.Schooler, "SIP: Session Initiation Protocol," IETF RFC 3261, June 2002

[4] D.Wu, Y.T.Hou, and Y.Q.Zhang, "Transporting real-time video over the Internet: Challenges and approaches," Proc. IEEE, vol.88, pp.1855-1875, Dec. 2000

[5] The Computer Network and Network Intelligence Group of Politecnico di Torino, "WinPcap: a Packet Capture Architecture for Windows," http://netgroup-serv.polito.it/winpcap/

[6] Tobias Oetiker and D. Rand, "MRTG: Multi Router Traffic Grapher," http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/

[7] The Ethereal Network Analyzer, http://www.ethereal.com/

[8] Sniffer Technologies, http://www.sniffer.com

[9] network top (ntop), http://www.ntop.org

[10] National Computerization Agency, http://www.nca.or.kr

[11] IANA, "Port Numbers", http://www.isi.edu/in-notes/iana/assignments/port-numbers

[12] W. Stallings, "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2," 3rd Edition, Addison-Wesley Pub Co, January 1999

[13] 유승화, 노병희 외, 차세대 인터넷으로의 전환에 대비한 데이터/음성/영상 트래픽 측정 및 분석에 관한 연구, 최종보고서, 한국전산원, 2001년 11월

◐ 저 자 소 개 ◑

노 병 희

1987년 한양대학교 전자공학과 졸업(학사)
1989년 한국과학기술원 전기및전자공학과 졸업(석사)
1998년 한국과학기술원 전기및전자공학과 졸업(박사)
1989년~1994년 한국통신 통신망연구소
1998년~2000년 삼성전자
2000년~현재 아주대학교 정보통신전문대학원 부교수
관심분야 : 유/무선 인터넷 멀티미디어 통신및 응용, 트래픽 제어, 유비쿼터스 네트워킹, RFID 네트워킹, 인터넷보안
E-mail : bhroh@ajou.ac.kr