http://dx.doi.org/10.3745/KTCCS.2013.2.8.327

Web Application Awareness using HTTP Host

Choi Ji Hyeok† · Kim Myung Sup††

ABSTRACT

Today's network traffic has become extremely complex and diverse since the speed of network became faster and a variety of application services appear. Moreover, many applications appear and disappear fast and continuously. However, the current traffic classification system does not give much attention to this dynamic change of applications. In this paper, we propose an application awareness system in order to solve this problem. The application awareness system can provide the information, such as the usage trend of conventional applications and the emergence of new applications by recognizing the application name in a rapidly changing network environment. In order to recognize the application name, the Host field of HTTP protocol has been utilized. The proposed mechanism consists of two steps. First, the system generates the candidates of application name by extracting the domain name from the Host field in HTTP packet. Second, the administrator confirms the name afterward. The validity of the proposed system has been proved through the experiments in campus network.

Keywords : Traffic Classification, Application Awareness, HTTP Host Field, Domain Name

HTTP Host를 이용한 웹 어플리케이션 인식에 관한 연구

최 지 혁† · 김 명 섭††

요 약

네트워크의 고속화와 다양한 응용 서비스의 등장으로 오늘날의 네트워크 트래픽은 복잡해지고 다양해졌다. 지금 이 순간에도 수 많은 응용들이 나타나고 사라지기를 반복하고 있는데, 이러한 다양한 트래픽의 변화에 현재의 트래픽 분류 시스템은 빠르게 대처 하지 못하고 있다. 본 논문에서는 이러한 문제점을 해결하기 위해 새롭게 출현하는 응용에 빠르게 대처할 수 있는 응용 인식 시스템을 제안한다. 응용 인식 시스템은 빠르게 변화하는 네트워크 환경에서 응용프로그램들의 이름을 인식하여 새로운 응용의 출현과 기존 응용의 변화 추이 등의 정보를 제공한다. 본 논문에서 빠르고 정확한 응용 인식을 위해 HTTP 프로토콜의 Host 필드를 이용한다. Host 필드의 domain정보를 추출하여 응용의 이름을 임시로 정하고 추후 관리자의 개입을 통해 응용의 이름을 확정 짓는 구조이다. 단순히 응용의 이름만을 알아내는데 그치지 않고 응용마다 고유의 Client IP를 카운팅하여 분석 대상 망에서 많이 사용된 응용들을 알아 낼 수 있다. 또한 응용 인식을 통해 나온 응용들을 트래픽 분류 시스템에 등록하여 기존에 분석 되지 않았던 새로운 응용들에 대한 분석도 가능하게 된다. 제안한 방법은 학내 망에서의 실험을 통해 결과를 도출하고 시나리오 별로 결과를 나눠서 분석함으로써 타당성을 증명하였다.

키워드 : 트래픽 분류, 응용 인식, HTTP Host 필드, 도메인 이름

1. 서 론

네트워크의 고속화와 다양한 응용 서비스의 등장으로 오늘날의 네트워크 트래픽은 복잡해지고 다양해졌다. 지금 이 순간에도 수많은 응용들이 나타나고 사라지기를 반복하고 있는데, 이러한 복잡 다양한 트래픽들을 현재의 트래픽 분류 시스템은 빠르게 대처를 하지 못하고 있다. 지금까지의 트래픽 분류 방법은 관리자가 원하는 응용을 선정하고 해당 응용에 한해서만 시그니처를 만들고 트래픽을 분류하고 있는데 이런 방식으로는 빠르게 변화하는 응용들에 적용하기 어려울 뿐만 아니라 트래픽을 분류하는데 있어서 분석률 또한 떨어질 수 밖에 없다[1, 13, 17].

본 논문에서는 응용들의 이름을 신속하게 파악 하여 분류 시스템에 적용 할 수 있도록 도움을 주는 응용 인식 시스템을 제안한다. 응용 인식 시스템이 응용의 이름을 파악하기

※ 이 논문은 2012년 정부(교육과학기술부)의 재원으로 한국연구재단(2012R1A1A2007483) 및 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보.컴퓨팅기술개발사업(2010-0020728)의 지원을 받아 수행된 연구임.
† 준 회 원 : 고려대학교 컴퓨터정보학과 석사과정
†† 종신회원 : 고려대학교 컴퓨터정보학과 부교수
논문접수 : 2013년 2월 13일
수 정 일 : 1차 2013년 4월 26일, 2차 2013년 5월 31일
심사완료 : 2013년 5월 31일
* Corresponding Author : Kim Myung Sup(tmskim@korea.ac.kr)

위해 사용한 방법은 HTTP 프로토콜의 특징을 이용하는 것이다[18]. 현재 사용되고 있는 다수의 응용들은 인터넷에 기반하여 동작하기 때문에 HTTP 프로토콜을 많이 사용하게 된다. 이러한 HTTP 프로토콜에는 응용의 이름과 가장 유사한 domain 이름을 가진 host 필드가 존재하는데, 응용 인식 시스템은 이러한 domain 이름을 기반으로 응용의 이름을 유추할 수 있다. 또한 응용의 이름을 알아내는데 그치지 않고 하루 동안 사용된 응용의 빈도 수를 체크하여 가장 많이 사용된 응용 순서대로 관리자에게 리포팅 한다. 관리자는 이러한 결과를 보고 하루 동안 분석 대상 망에서 사용 된 응용들에 대해서 파악 할 수 있고, 가장 많이 사용된 응용과 그렇지 못한 응용들 또한 알 수 있다. 단순히 응용의 이름과 사용 빈도수만 파악 하는 것이 아니라 추가적으로 다양한 분석이 이루질 수 있다. 또한 새로운 응용이나 많이 사용 된 응용들에 대해서는 시그니처를 만들어 트래픽 분류 시스템에 적용하여 분석률을 높이는 작업을 수행할 수 있다.

본 논문은 다음과 같이 구성된다. 서론에 이어 2장에서는 관련 연구로서 기존 트래픽 분류 방법에 대해 간략히 설명하고 응용 인식의 필요성을 기술한다. 3장에서는 응용 인식 시스템의 전체적인 구조와 응용 인식 알고리즘에 대해 설명한다. 4장에서는 응용 인식 시스템을 통해 나온 결과로 실험을 진행하고 실험 결과에 대해 기술한다. 마지막으로 5장에서는 결론 및 향후 연구에 대해 기술한다.

2. 관련 연구

본 장에서는 현재 트래픽 분류 시스템의 트래픽 분류 과정과 트래픽들을 분류하기 위해 지금까지 연구된 트래픽 분류 방법들에 대해 간단히 설명하고, 응용 인식 시스템의 필요성에 대해 설명한다.

현재의 트래픽 분류 시스템은 시그니처 기반 분류 방법을 주로 사용하고 있다[2, 11]. 시그니처를 만들기 위해서는 먼저 분석하고자 하는 응용을 선정하고 해당 응용에 맞는 트래픽을 수집하게 된다. 그 다음에는 다양한 트래픽 분류 방법론을 통해 시그니처를 생성하고 마지막으로 시그니처를 분류 시스템에 적용 하게 된다[13, 17]. 이러한 트래픽 분류 방법에는 크게 헤더 기반[3, 4, 5, 6, 12], 페이로드[7, 11]기반, 머신 러닝[8, 9, 12]기반, 트래픽 상관관계[1, 4, 10]기반, 통계 기반 분석[14, 15, 16]으로 나눌 수 있다.

헤더 기반 트래픽 분류는 잘 알려진 포트 번호(well-known Port Number)를 사용하는 HTTP, FTP, e-mail, SMTP 등 IANA[5]에서 지정한 포트 정보를 이용하는 방법이고, 페이로드 기반 트래픽 분류 방법은 각 응용 트래픽별로 그들만이 사용하는 다른 응용들과 구분되는 페이로드 내의 공통분모를 찾아내어 그것을 이용해 트래픽을 분류하는 방법이다. 그리고 머신 러닝 기반 트래픽 분류 방법은 응용 별 트래픽의 특징이 될 수 있는 요소(port, inter-arrival time, packet size)를 머신 러닝 알고리즘으로 학습을 시킨 후에 분류하는 방법이다. 트래픽 상관관계 기반 분류 방법은 인터넷 트래픽의 3레벨 주소체계(IP address, port number, protocol)과 트래픽 발생 형태 등의 고유한 특성을 바탕으로 연관성을 가중치로 표현하고 그 임계값을 설정하여 트래픽을 분류하는 방법이다. 마지막으로 통계 시그니처 기반 분류 방법은 패킷의 헤더 정보(패킷 크기, 윈도우 크기 등)나 캡처 정보(패킷 캡처 시간 등)를 기반으로 하여 다른 응용 프로그램과 구별할 수 있는 응용 프로그램의 고유한 통계적 특징을 찾아서 분류하는 방법이다.

이러한 트래픽 분류 방법론들은 단순히 트래픽을 다양한 관점에서 바라보고 분석하는 방법론일 뿐이다. 결국 이러한 분류 방법들이 필요한 이유는 특정 응용을 좀 더 정확하고 다양한 각도에서 분석하기 위함이다. 그렇다면 결국 중요한 것은 어떤 응용을 분석 할 것인가에 대한 응용 선정의 문제이다[17]. 응용 선정을 잘 하기 위해서는 현재 가장 많이 사용되고 인기가 있는 응용들이 어떤 것들이 있는지에 대한 분석이 먼저 선행 되어야 한다.

그러기 위해 본 논문은 빠르게 변화하는 응용들을 신속하게 알아 낼 수 있는 응용 인식 시스템을 제안하고 응용 인식 결과로 나온 결과들을 바탕으로 다양한 실험을 진행하여 응용 인식 시스템의 필요성과 타당성을 증명한다.

3. 응용 인식 시스템

본 장에서는 응용 인식 시스템의 전체 구성과 응용 인식 알고리즘에 대해서 기술한다. 먼저 시스템의 전체 구성도는 다음과 같다.

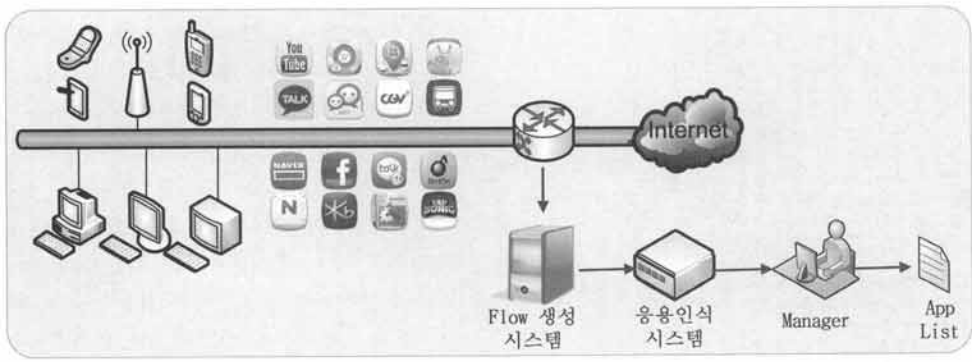

Fig. 1. System Structure

Fig. 1과 같이 응용 인식 시스템은 크게 세 모듈로 구성된다. 첫 번째 모듈은 대상 망에서 발생되는 모든 유무선 트래픽을 수집하여 플로우 형태로 만들어주는 Flow Generator이다. 두 번째 모듈은Flow Generator에서 생성한 플로우를 입력으로 받아 HTTP 플로우만을 추출하는 HTTP 플로우 수집 모듈이다. 세 번째 모듈은 추출된 HTTP 플로우를 입력으로 받아 응용의 이름을 인식하는 응용 인식 모듈이다. 본 논문에서는Flow Generator를 제외한 HTTP Flow 수집 모듈과 응용 인식 모듈에 대하여 기술한다.

3.1 HTTP Flow 수집 모듈

본 절에서는 HTTP 플로우 수집 모듈의 구조와 기능에 대해 기술한다. Fig. 2는 HTTP 플로우 수집 모듈의 전체적인 구조를 보여준다.

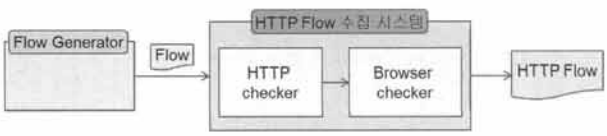

Fig. 2. HTTP flow collect system structure

HTTP 플로우 수집 시스템은 분석 대상 망에서 발생하는 유무선을 포함한 모든 트래픽들을 패킷 캡처 프로그램을 통해 수집한 후, Flow Generator에 의해 플로우 형태로 변환된다. 생성된 플로우는 HTTP 플로우 수집 시스템의 입력으로 들어 가게 된다. HTTP 플로우 수집 시스템은 크게 두 가지의 모듈로 구성되어 있는데, 해당 플로우가 HTTP 프로토콜을 사용한 플로우인지 검사하는 HTTP checker모듈과 해당 플로우가 IE, chrome, firefox에서 발생 된 플로우인지 검사하는 Browser checker모듈로 나누어져 있다.

먼저 첫 번째 HTTP checker 모듈은 해당 플로우가 HTTP 프로토콜을 사용한 플로우인지 아닌지를 검사하는 모듈이다. HTTP checker에서 사용되는 HTTP 플로우 판별법은 HTTP method 필드를 이용하는 것이다. HTTP 프로토콜 규정에서 정의된 method는 Table 1과 같이 총 8가지이다.

Table 1. HTTP Method

GET	POST	HEAD	PUT
TRACE	CONNECT	DELETE	OPTIONS

Method 필드는 항상 HTTP 플로우 첫 번째 패킷 맨 앞부분에 정의 되어 있기 때문에, 실제 패킷의 맨 앞부분과 HTTP method 필드를 스트링 매칭 하여 일치하게 되면 해당 플로우가 HTTP 프로토콜을 사용하는 플로우라는 것을 알 수 있다. HTTP 플로우로 판별된 플로우는 두 번째 모듈인 Browser checker의 입력이 된다.

Browser checker에서 하는 역할은 User-Agent가 IE, chrome, firefox와 같이 웹 브라우저에서 발생 한 플로우일 경우 해당 플로우는 저장하지 않고 제거하는 역할을 한다. 일반적으로 사용자가 웹 브라우저를 이용하여 발생하는 트래픽들은 대부분 인터넷 서핑이나 뉴스 보기 아니면 인터넷 쇼핑 등 이다. 이러한 트래픽들은 특정 응용 프로그램을 사용 했을 때 발생한 트래픽이라고 보기 어렵기 때문에 웹 브라우저에서 발생 한 플로우들을 제거하는 과정을 거친다.

HTTP 플로우 수집 시스템의 두 가지 모듈을 통해 나온 플로우는 응용 인식 모듈의 입력이 된다.

3.2 응용 인식 모듈

본 절에서는 응용 인식 모듈의 구조와 응용을 인식하는 방법에 대한 알고리즘을 기술한다.

Fig. 3은 응용 인식 모듈의 Flow chart이다. 응용 인식 시스템은 HTTP 수집 시스템에서 추출된 하루 치 플로우 전체가 입력으로 들어가게 된다. 빠르게 변하는 응용들의 변화를 살피는 기준으로 하루가 적당하다고 판단 되었기 때문에 시스템의 인풋을 하루로 정하였다. 하루치의 입력이 들어 오면 응용 인식 모듈에서는 하나의 플로우를 읽어 들여 각각의 플로우마다 세 가지 정보를 추출 한다. 세 가지 정보에는 Host, User-Agent, Client IP가 있는데 이 세 가지의 정보가 필요한 이유는, 먼저 host 필드에는 응용의 이름과 가장 가까운 domain정보를 추출할 수가 있다. Table 2를 보면 domain 정보가 어느 정도 응용의 이름과 일치하는 것을 알 수가 있다.

하지만 Host 정보만으로는 응용의 이름을 판단하기 어려운 응용들이 있다. v3나 facebook 같은 경우는 host 정보만

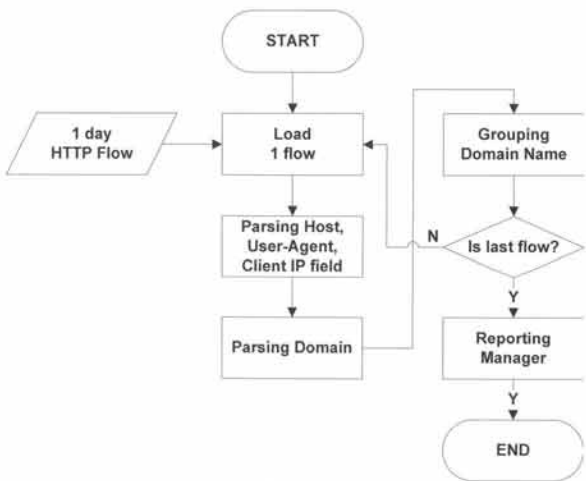

Fig. 3. Application Awareness Module Flow Chart

Table 2. Host, User-Agent information

응용 이름	Host	User-Agent
Gomtv	trlog.gomtv.com	Htttpgetfile
Google	tools.google.com	Google
V3	gms.ahnlab.com	V3inet2
Facebook	Profile.ak.fbcdn.net	facebook

으로는 응용의 이름을 판별 하기 어렵기 때문에 User-Agent 정보를 보면 응용을 판단하는데 큰 도움이 된다.

마지막으로 Client IP 정보가 필요한 이유는 하루 동안 가장 많이 쓰인 응용을 알아보기 위함이다. 먼저, domain 이름이 같은 플로우들을 체크하고 그 중에서도 다른 Client IP와 중복되지 않는 Client IP만 카운팅 한다. 즉, 하나의 응용을 동일한 사용자가 하루에 두 번 사용 하더라도 Client IP는 하나가 되는 것이다. 최종적으로는 Client IP 수가 domain 이름과 함께 응용 인식의 결과로 나오게 된다.

Host, User-Agent, Client IP 정보를 추출 한 후에는 위에서도 언급 했듯이 Host 필드에서 domain 정보를 추출하는 과정을 거치게 된다. Table 3은 다양한 host 필드에서의 domain 위치를 보여준다.

Table 3에서 나온 host들은 모두 서로 다른 형태를 지니고 있지만 공통점이 하나 있는데 이는 응용의 이름과 유사한 domain이 항상 2계층 혹은 3계층에 있다는 것이다. Bigtorrent.org:2710 같은 경우는 뒤에 포트번호만 제거하면 2 계층에 이름과 가까운 domain 정보가 있다는 것을 알 수 있다. 그리고 마지막에 있는 175.178.17.38같은 host는 예외처리를 통해 제거하는 작업을 수행 한다.

Table 4는 domain의 위치가 2계층에 있는지 3계층에 있는지를 알아낼 수 있는 psedo code 이다. 1계층의 글자 수가 .com이나 .info 처럼3글자 이상일 때는 무조건 domain의 위치는 2계층이다. 하지만 .kr 같은 경우는 torrent.kr 처럼 domain이 2계층에 위치 할 수도 있고 altools.co.kr 처럼 3계층에 올 수도 있다.

Table 3. Domain location in host field

Host	Domain 위치
www.naver.com	2계층
Aldn.altools.co.kr	3계층
o-o.preferred.hkgo3305.v16.cache.c.pack.google.com	2 계층
Bigtorrent.org:2710	2 계층
175.178.17.38	X

Table 4. Find Domain Psedo Code

```
Procedure Find application name (flow)
    if(1 계층 domain 글자 수 = 2)
        if(2 계층 domain 글자 수 = 2)
        응용이름 = 3 계층
        else
        응용이름 = 2 계층
    else if(1 계층 domain 글자 수 >= 3)
        응용이름 = 2 계층
    else
        응용이름 = 찾을 수 없음
end procedure
```

Table 4과 같은 알고리즘을 통해 domain을 찾은 후, 같은 이름을 가진 domain들끼리 그룹핑하는 작업을 수행 하게 된다. 그룹핑이 진행되는 과정과 동시에 Clinet IP를 카운팅하는 작업을 진행 하게 되는데, 이를 통하여 나온 Client IP는 식(1)과 같이 해당 응용의 score가 된다.

$$Score = C_{ip} \tag{1}$$

Score 시스템을 도입한 이유는 응용의 사용 빈도를 점수로 표현하여 하루가 다르게 변하는 응용의 변화를 점수의 변동을 통해 쉽게 파악하고 분석할 수 있기 때문이다. 특정 응용이 얼마나 많은 client들에게 사용 되었는지를 파악해야 현재 분석 대상 망에서 사용 되는 응용들의 경향을 쉽게 파악 할 수 있다.

하루 동안 발생된 모든 플로우들이 그룹핑을 마치면 Table 5와 같이 하루 동안 발생한 모든 플로우에 대한 응용 이름과 score가 관리자에게 리포팅 된다.

Table 5. Application score

응용 이름	score
Microsoft	1444
Windowupdate	1380
Naver	1133
Ahnlab	1044
Thawte	1044
Verisign	952
Google	829
Daum	749
Altools	566
Nateon	477
...	...

Table 5의 결과는 4장에서 진행된 하루의 실험 데이터에서 가장 높은 score를 기록한 상위 10개의 응용을 나타낸 것이다. 결과에 대한 자세한 내용은 4장에서 기술한다.

응용의 이름과 score가 관리자에게 전달 되면 관리자는 크게 두 가지의 실험을 통해 결과를 정리하게 된다. 먼저 실험 기간 동안 사용된 응용들의 score 변동폭을 살펴 보고, 특정 응용에 대한 사용자 수의 증가와 감소를 파악한다. 그리고 이전에는 없던 새로운 응용이 나타났는지에 대해 알아보게 된다. 두 번째 실험에서는 응용의 score가 높음에도 불구하고 트래픽 분류 시스템에 등록 되지 않은 응용들에 한해서 시그니처를 만들고 분류 시스템에 적용하는 실험을 진행 한다.

4. 실험 및 실험 결과

이번 장은 학교 망에서 응용 인식 시스템을 구축하고, 실

Table 6. Traffic Trace

Total trace	Test trace
총 77일(11 주)	총 44일(11 주)

험한 결과에 대해 기술한다. Table 6는 학교 망에서 실험한 트래픽 트레이스를 보여준다.

수집된 총 traffic과 실제로 쓰인 실험 traffic이 다른 이유는 학교 망의 특성상 트래픽이 적은 금요일, 토요일, 일요일을 제외한 주 4일에 대한 트래픽만을 실험 트래픽으로 선정하였기 때문이다.

4.1 실험 1

본 절에서는 총 44(11 주)일의 응용 인식 결과를 통계적으로 분석하고, 그 결과에 대해 정리해 보았다.

보통 하루치 플로우를 응용 인식 시스템의 입력으로 넣게 되면 수천 개의 서로 다른 응용들이 나오게 된다. 하지만 수천 개의 응용들이 모두 다 중요하진 않기 때문에 score 시스템을 이용하여 일주일 단위로 의미 있는 응용들을 선정하였다.

$$1\,day\,score \geq 200 \text{ and } other\,day\,score \geq 100 \quad (2)$$

위의 식(2)는 의미 있는 응용 선정 기준을 수식으로 정리한 부분으로 위 식에서 말하는 1day score는 하루 동안 특정 응용을 사용한 client수를, other day score는 1day score를 제외한 일주일의 나머지 날 동안 꾸준히 사용한 client수를 의미한다. 즉, 일주일 중에 적어도 하루만큼은 200명 이상의 client가 사용을 해야 하고, 일주일의 나머지 날들은 100명 이상의 client가 사용을 해야 의미가 있는 응용이라 정하였다. 수천 개의 응용 중에 의미 있는 응용으로 선정된 응용의 수는 실험 기간 동안 일주일 평균 26개 정도이고 본 논문에서는 의미 있는 응용에 대해서 분석한 내용만을 기술한다.

Table 7. Selection application result

응용이름 (첫주)	score	응용이름 (마지막 주)	score
microsoft	1454	microsoft	1518
Windowsupdate	1409	Windowsupdate	1396
ahnlab	1081	naver	1198
thawte	995	ahnlab	1122
verisign	936	google	887
naver	932	verisign	879
google	816	thawte	846
daum	662	daum	799
altools	580	altools	567
msftncsi	483	msftncsi	496
usertrust	450	adobe	457
nateon	416	nateon	450
…		…	
altoolbar	176	youtube	223

위와 같은 기준으로 총 11주의 응용 인식 결과를 정리해 보면 Table 7과 같다. 전체를 보여주기엔 너무 양이 많기 때문에 11주의 첫 주와 마지막 주의 결과를 정리하였다.

첫 주와 마지막 주의 응용 list와 점수를 살펴보면 커다란 변화는 없는 것처럼 보이지만 그 안에서 작은 변화들이 존재 하였다. 이러한 변화들을 크게 세 가지의 시나리오로 나누어 분석하였다.

시나리오 1은 실험 기간 동안 점수 변화가 200 점 이하인 응용들을 보여준다.

$$Score_{vary} \leq 200 \quad (3)$$

시나리오 2는 실험 기간 동안 점수 변화가 200점 이상인 응용들을 보여준다.

$$Score_{vary} > 200 \quad (4)$$

시나리오 3은 실험 기간 동안 새롭게 출현한 응용들에 대해서 보여준다.

대부분의 응용들이 시나리오 1에 해당 되었는데, 그 중에서도 중요한 몇 가지 응용들에 대해 분석한 결과는 Table 8과 같다.

시나리오 1에 해당 되는 응용들은 대체적으로 인기 있는 응용들이 많이 있다. naver, daum, google과 같은 대형 포탈 사이트와youtube나 gomtv같은 멀티미디어 응용들 그리고 그 외에 nateon, v3, altools등 일반적으로 많이 사용하는 응용들이 있다는 것을 확인할 수 있다. 그리고 thawte나

Table 8. Scenario 1 applications

응용 이름	score
microsoft	1450 ~ 1639
Windowupdate	1380 ~ 1568
ahnlab	1040 ~ 1180
Naver	830 ~ 1010
Thawte	812 ~ 996
Verisign	805 ~ 957
Google	794 ~ 940
Daum	662 ~ 849
Altools	589 ~ 772
Nateon	386 ~ 491
Youtube	139 ~ 233
Gomtv	143 ~ 216

verisign처럼 보안 인증서 쪽으로 유명한 응용들도 존재 한다. 그리고 최상위권에 위치한 windowupdate와 microsoft는 11주 동안 1위와 2위를 차지하고 있는데 그 이유는 학교 망의 특성상 모든 컴퓨터에 설치된 하드 보안관 때문이라고 추측된다. 하드 보안관은 컴퓨터를 재 부팅 할 때마다 윈도우 업데이트를 실행 하게 되는데 이로 인해 두 개의 응용이 가장 많은 트래픽을 발생한 것으로 판단된다.

시나리오 2에 해당 되는 응용들을 정리한 결과 총 2개(facebook, public-trust)의 응용이 시나리오 2에 해당 되었다.

Table 9. Scenario 2 applications

응용 이름	score
Facebook	234 ~ 442
Public-trust	101 ~ 403

public-trust같은 경우는 보안 인증서쪽 응용으로 파악이 되고, 특정 한 주에서만 높은 score를 기록하고 다시 원래의 score를 유지하였다. facebook의 경우는 평소에는 높은 score를 유지하다가 시험기간 때 상대적으로 낮은 score를 기록 하였기 때문에 시나리오 2에 해당되었다. 시험이 끝난 후 에는 본래의 score를 유지 하였다.

시나리오 3에 해당 되는 응용은 Table 10과 같이 단 두 개의 응용뿐이었다.

Table 10. Scenario 3 applications

응용 이름	score	출현 빈도수
Gameframe	약 1615	2번
odnoklassniki	약 350	6번

gameframe 같은 경우는 11주의 실험 기간 동안 이틀만 나왔는데 그 양이 windowupdate 보다 많았다. 그래서 추측하기로는 windowupdate처럼 하드 보안관과 상관이 있는 응용이라고 파악 된다. 그렇지 않다면 1600개나 되는 ip에서 발생되지 않았을 것이다. odnoklassniki는 러시아에서 사용 하는 SNS라고만 파악이 되었다. 두 개의 새로 나온 응용의 공통점은 실험 기간 동안 새롭게 나타나서 꾸준히 score를 유지하는 것이 아니라 잠깐 나타났다가 사라져 버렸기 때문에 사용자들이 많이 사용해서 발생된 응용이라고 보기 어렵다.

실험 1에서는 하루 단위의 응용인식 결과를 주 단위의 의미 있는 응용으로 선정하고 score의 변동에 따라 크게 세 가지 관점으로 나누어서 분석을 진행 하였다. 이를 통해 분석 대상 망에서 어떤 응용들이 많이 사용되고 적게 사용 되는지에 대해 파악 할 수 있었고, 변화폭이 컸던 응용들과 새롭게 나온 응용들에 대해서도 알 수 있었다.

4.2 실험 2

본 절에서는 응용 인식 시스템을 통해 알게 된 응용들을 바탕으로 시그니쳐를 생성하여 트래픽 분류 시스템에 적용하는 실험을 진행하였다. 현재 학내 망에 설치된 분류 시스템은 분석하지 못하는 HTTP 기반 응용들이 다수 존재 한다. 그 이유는 해당 응용에 대한 시그니쳐를 가지고 있지 못하기 때문이다. 응용 인식 결과로 나온 의미 있는 응용들 중에서 현재 분석기가 가지고 있지 않은 응용을 정리한 내용이 Table 11이다. 그리고 Table 12은 기존의 분류 시스템이 가지고 있는 응용이지만 응용 인식결과를 통해 시그니쳐가 업데이트 된 경우이다.

Table 11. additional application List

thawte	nefficient	msftncsi	usertrust
verisign	globalsign	entrust	digicert
msecnd	comodoca	4shared	x-cdn
adlocal	macromedia	geotrust	sun
nprotect			

Table 12. Update application List

Altools	Google	Svchost	Facebook	V3

Table 11과 Table 12에 나와있는 응용들의 시그니쳐를 만들고 분석기에 적용하면 Table 13과 같은 결과를 얻을 수 있었다.

시그니쳐 추가 전의 분석기는 하루치 데이터를 분석 하였을 때, flow 단위로는 20,522,400개의 flow를 분석하였다. 하지만 응용 인식 결과로 인해 시그니쳐를 추가한 후에는 그보다 103,982개가 많은 20,626,382개를 flow를 분석할 수 있게 되었다. 패킷 수와 바이트 양도 마찬가지로 이전의 분석기 보다 더 많은 양을 분석할 수 있게 되었다.

Table 13. Completeness compare

	적용 전	적용 후	결과
Flow	20,522,400	20,626,382	103,982 flow 증가
Pkt	568,922,849	574,953,966	6,031,117 pkt 증가
Byte	436,913,261,076	441,721,075,266	4,807,814,190 byte 증가

5. 결론 및 향후 과제

응용 인식 시스템은 빠르게 변화하고 있는 다양한 응용들을 신속하고 정확하게 알아 낼 수 있는 시스템이다. 본 논문에서 제안하는 응용 인식 방법은 HTTP host 필드에 존재하는 응용의 이름과 가장 유사한 domain 이름을 추출하고 추후에 관리자의 개입으로 응용의 이름을 확정 짓는 시스템 구조이다. 간단한 방법으로 응용의 이름을 알아냄으로

써 새롭게 출현 하는 응용에 대한 신속한 파악이 가능하고, 응용들의 사용 빈도수를 체크하여 분석 대상 망에서 가장 많이 쓰인 응용들을 파악 할 수 있다. 또한 응용 인식 시스템에서 나온 결과는 관리자의 추가적인 작업으로 더욱 다양한 분석이 가능해질 뿐만 아니라 트래픽 분류 시스템에서 응용의 이름을 몰라 분석하지 못하였던 응용들의 시그니쳐를 생성함으로써 트래픽 분석이 가능해진다.

향후 연구로는 HTTP 프로토콜을 제외한 다른 프로토콜 중에 응용의 이름과 가까운 정보가 있는지에 대한 확인이 필요하다. 그리고 모바일 트래픽 같은 경우는 지금 사용하는 domain 추출 알고리즘으로 추출이 불가능한 경우가 있기 때문에 모바일 트래픽에 적합한 알고리즘을 개발하는 연구가 필요하다.

참 고 문 헌

[1] Myung-Sup Kim, Young J. Won, and James Won-Ki Hong, "Application-Level Traffic Monitoring and an Analysis on IP Networks," ETRI Journal, Vol.27, No.1, Feb., 2005, pp.22-42.

[2] Jun-Sang Park, Jin-Wan Park, Sung-Ho Yoon, Young-Seok Oh, Myung-Sup Kim, "Development of signature Generation system and Verification Network for Application Level Traffic classification", Conference of Korea Information Communication Society, Apr. 23-24, 2009, pp.1288-1291.

[3] W. Li et al."Efficient application identificationand the temporal and spatial stability of classification schema", Computer Networks, 2009.doi:10.1016/ j.comnet.2008.11.016.

[4] Thomas Karagiannis, Konstantina Papagiannaki, Michalis Faloutsos. "BLINC: Multilevel Traffic Classification in the Dark", Proc. of SIGCOMM 2005, Philadelphia, PA, Aug. 22-26, 2005.

[5] IANA port number list, IANA, http://www.iana.org/assignments/port-numbers.

[6] Jian Zhang and Andrew Moore, "Traffic Trace Artifacts due to Monitoring Via Port Mirroring," Proc. of the IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services (E2EMON) 2007, Munich, Germany, May 21, 2007.

[7] Risso, F. Baldi, M. Morandi, O. Baldini, A.Monclus, P. "Lightweight, Payload-Based Traffic Classification:An Experimental Evaluation," Proc. of the Communications, 2008. ICC '08. IEEE International Conference, 2008.

[8] Jeffrey Erman, Martin Arlitt, Anirban Mahanti,"Traffic Classification Using Clustering Algorithms," Proc. of SIGCOMM Workshop on Mining network data, Pisa, Italy, Sep., 2006, pp.281-286.

[9] Andrew W. Moore and Denis Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques," Proc. of the ACM SIGMETRICS, Banff, Canada, Jun., 2005.

[10] Thomas Karagiannis, Konstantina Papagiannaki, and Michalis Faloutsos. "BLINC: Multilevel Traffic Classification in the Dark," Proc. of SIGCOMM 2005, Philadelphia, PA, Aug. 22-26, 2005.

[11] Liu, Hui Feng, Wenfeng Huang, Yongfeng Li, Xing "Accurate Traffic Classification", Networking, Architecture, and Storage, 2007. International Conference

[12] Hyun-chul Kim, kc claffy, Marina Fomenkov, Dhiman Barman, Michalis Faloutsos, Ki-young Lee, "Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices" Proc. of ACM SIGCOMM CoNEXT, Madrid, Spain, Dec., 2008.

[13] Young-suk Oh, Jun-sang Park, Sung-ho Yoon, Jin-wan Park, Sang-woo Lee, Myung-sup Kim, "Multi-Level basd Application Traffic Classification Method", The Korean Institute of Communications and Information Science, Vol.35, No.8, pp.1170-1178.

[14] Jin-Wan Park, Myung-Sup Kim, "Performance Improvement of the Statistic Signature based Traffic Identification System", Conference of Korea Information Communication Society, Aug., 2011.

[15] Hyun-Min An, ji-hyeok Choi, Myung-Sup Kim, "A Method to resolve the Limit of Traffic Classification caused by Abnormal TCP Session", KNOM Review, Vol.15, No.1, Dec., 2012, pp.31-39.

[16] Hyun-Min An, Min Hur, Myung-Sup Kim, "A Study on the Limit of Traffic Classification using Payload Size Distribution caused by Abnormal TCP Session", The Korean Institute of Communications and Information Science, Jun. 20-22, 2012, pp.347-348.

[17] Ji-hyeok Choi, Sung-Ho Yoon, Myung-Sup Kim, "A study on signature extraction method for application-level traffic classification", The Korean Institute of Communications and Information Science, Feb. 8-10, 2012.

[18] Ji-Hyeok Choi, Jun-Sang Park, Myung-Sup Kim, "A Study on Awareness of Application using HTTP Traffic", The Korean Institute of Communications and Information Science, Jun. 20-22, 2012, pp.1000-1001.

최 지 혁

e-mail : jihyeok_choi@korea.ac.kr
2012년 고려대학교 컴퓨터정보학과(학사)
2012년~현 재 고려대학교 컴퓨터정보학과 석사과정
관심분야 : 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

김 명 섭

e-mail : tmskim@korea.ac.kr
1998년 포항공과대학교 전자계산학과(학사)
1998년~2000년 포항공과대학교 컴퓨터공학과(석사)
2000년~2004년 포항공과대학교 컴퓨터공학과(박사)
2004년~2006년 Post-Doc., Dept. of ECE, Univ. of Toronto, Canada
2006년~현 재 고려대학교 컴퓨터정보학과 부교수
관심분야 : 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 멀티미디어 네트워크