DOI: 10.3745/KIPSTC.2010.17C.1.099

응용 레벨 트래픽 분류를 위한 시그니쳐 생성 및 갱신 시스템 개발

박 준 상[†] · 박 진 완[†] · 윤 성 호[†] · 이 현 신[†] · 김 명 섭[††]

요 약

네트워크 트래픽 모니터링과 분석은 엔터프라이즈 네트워크의 효율적인 운영과 안정적 서비스를 제공하기 위한 필수적인 요소이다. 응용 레벨 트래픽의 분석을 위한 다양한 방법이 존재하지만 분류의 정확성, 분석률, 실용성을 고려했을 때 페이로드 시그니쳐 기반의 분석 방법은 가장 높은 성능을 보인다. 하지만 페이로드 시그니쳐를 수작업으로 추출하는 과정은 응용프로그램 및 응용 프로토콜에 대한 선행적인 분석이 필요하기 때문에 많은 시간과 인력이 요구된다. 또한 응용프로그램의 통합, 변경, 출현은 시그니쳐의 유지 및 관리에 대한 복잡성을 증대시킨다. 따라서 본 논문에서는 응용프로그램의 페이로드 시그니쳐 생성 과정의 단점을 보완할 수 있는 시그니쳐 자동 생성 시스템을 제안하여 시그니쳐 생성 효율을 향상시키며, 응용프로그램의 변화, 출현에 유연하게 대처할 수 있는 페이로드 시그니쳐 갱신 시스템을 구축하여 지속적으로 시그니쳐 유지, 관리가 가능하도록 하였다. 또한 학내망에 적용하여 제안한 시스템의 실용성을 증명하였다.

키워드 : 응용 레벨 트래픽 분류, 시그니쳐, 자동 생성시스템, 시그니쳐 갱신 시스템

Development of Signature Generation and Update System for Application-level Traffic Classification

Jun-Sang Park[†] · Jin-Wan Park[†] · Sung-Ho Yoon[†] · Hyun-Shin Lee[†] · Myung-Sup Kim[††]

ABSTRACT

The traffic classification is a preliminary but essentialstep for stable network service provision and efficient network resource management. While various classification methods have been introduced in literature, the payload signature-based classification is accepted to give the highest performance in terms of accuracy, completeness, and practicality. However, the collection and maintenance of up-to-date signatures is very difficult and time consuming process to cope with the dynamics of Internet traffic over time. In this paper, We propose an automatic payload signature generation mechanism which reduces the time for signature generation and increases the granularity of signatures. Furthermore, We describe a signature update system to keep the latest signatures over time. By experiments with our campus network traffic we proved the feasibility of our mechanism.

Keywords : Application-Level Traffic Classification, Signature, Automatic Signature Generation, Signature Update System

1. 서 론

과거의 인터넷은 잘 알려진 포트 기반의 HTTP, Telnet, E-mail, FTP, NNTP의 응용들이 대부분의 인터넷 트래픽을 차지하고 있었기 때문에 IANA[1]에 정의된 포트 정보 기반의 분석으로 신뢰성이 높은 분석 결과를 도출할 수 있었다. 하지만 스트리밍 응용프로그램 및 passive FTP와 같이 하나의 응용 프로그램이 둘 이상의 세션을 형성하고, 이들 중 데이터 세션의 포트가 동적으로 생성됨에 따라 포트 기반의 분석은 더 이상 높은 신뢰성을 제공할 수 없게 되었다. 이를 보완하기 위한 방법으로 응용계층프로토콜 내용을 참조하여 동적으로 생성되는 포트 정보를 얻어내어 분석하는 방법인 Mmdump [2], SM-MON [3]에서 소개되었다. 그러나 이 방법은 RTSP, MMS, SIP와 같이 응용 프로토콜이 공개되었거나 알려진 응용 트래픽의 분석에만 사용 가능하고 비공개 응용 프로토콜이 대다수 포함된 전체 인터넷 트래픽에 적용할 수 없다는 문제점이 있다. 이와 같은 문제점을 해결하기 위해 시그니쳐 기반 분석 방법[4]이 제시되었다. 이 방

※ 이 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단(KRF-2007-331-D00387)과 한국전자통신연구원의 위탁연구과제 지원을 받아 수행된 연구임.
† 준 회 원 : 고려대학교 컴퓨터정보학과 석사과정
†† 종신회원 : 고려대학교 컴퓨터정보학과 조교수
논문접수 : 2009월 8월 25일
수 정 일 : 1차 2009년 10월 15일, 2차 2009년 10월 26일
심사완료 : 2009년 10월 26일

법은 시그니쳐가 확인된 응용에 대해서는 정확한 분석이 가능하다는 장점을 갖지만, 모든 응용별로 시그니쳐를 수작업으로 찾아야만 하고, 찾아진 시그니쳐가 응용프로그램의 변화에 유연하게 대처하지 못하는 문제점을 보인다. 또한 다양한 연구[6, 8, 11]에서 90%이상의 높은 분석률을 주장하지만 검증을 위한 데이터의 범위가 제한적이며, 가정에 기반한 검증 결과를 보이고 있어 신뢰성을 보장하지 못하는 문제점을 보인다.

따라서 본 논문에서는 페이로드 시그니쳐 생성 과정의 단점을 보완하여효율적으로 시그니쳐 추출이 가능한 페이로드 시그니쳐 자동 생성 시스템을 제안한다. 또한 에이전트를 기반으로 검증 네트워크를 구축하여 생성 시스템의 실효성과 생성된 시그니쳐의 정확성을 객관적으로 증명하고, 검증된 결과를 바탕으로 응용프로그램의 변화에 유연하게 대처할 수 있는 시그니쳐 갱신 시스템을 제안한다.

본 논문은 다음과 같은 순서로 구성된다. 2장에서는 응용 레벨 시그니쳐 생성에 관한 기존 연구에 대한 문제점을 살펴보고, 3장에서는 응용프로그램 시그니쳐 생성을 위한 고려사항에 대해 설명한다. 4장에서는 페이로드 시그니쳐 생성 방법 및 시스템의 구성에 대해 기술한다. 5장에서는 제안한 생성 시스템에 의해서 추출한 시그니쳐와 기존 연구에서 보고되는 시그니쳐를 비교함으로써 생성 시스템의 실효성을 증명한다. 6장에서는 검증 시스템에 기반한 시그니쳐 갱신시스템에 대해 기술하고, 분류된 결과를 분석한다. 마지막으로 7장에서는 결론을 맺고 향후 연구에 대하여 기술한다.

2. 관련 연구

페이로드 시그니쳐의 자동 생성 방법에 관한 기존의 연구[5-8, 10] 에서는페이로드 데이터로부터 동일한 스트링을 추출하는 알고리즘을 제시하고 있지만, 수동적인 데이터의 수집 방법을 사용하고, 추출되는 시그니쳐의 정보를 스트링으로 제한한다. 또한다른 응용프로그램 시그니쳐와의 충돌 관계를 고려하지 않고 생성된다.

[5]는 LCS알고리즘에 기반한 시그니쳐 자동 생성 시스템을 제시하고 있다. 하지만 LCS의 입력 데이터에 대한 제약사항으로 패킷 크기만을 비교하여 적용하고 있다. 이러한 방법은 패킷 크기에 대한 임계값 설정에 어려움이 발생하고, 다른기능을 수행하는 트래픽이 동일한 패킷 크기를 갖는 경우 시그니쳐 생성이 불가하거나, 잘 못된 시그니쳐의 추출 가능성이 높다. 응용프로그램은 다양한 형태의 트래픽을 발생시키기 때문에 LCS 입력 데이터를 생성하기 위해 유사한 동작을 수행하는 트래픽의 그룹핑 과정이 반드시 요구된다. 또한 시그니쳐로서 추출하는 정보가 스트링으로 제한되어 있다. 페이로드 시그니쳐의 정확성을 향상 시키고, 분류 시스템의 부하를 줄이기 위해서는 패킷의 순서, 페이로드 offset 정보에 대한 추가적인 정보의 추출이 요구된다.

[7]은 기계 학습에 기반한 시그니쳐 자동 생성 시스템을 제시하고 있다. 기계 학습에 기반한 방법은 학습을 위한 Ground Truth 데이터의 정확성에 따라 그 성능이 좌우되기 때문에 정확한 데이터 수집 방법이 전제되어야 한다. 하지만 수작업으로 시그니쳐 추출을 위한 데이터를 수집하기 때문에 정확성을 보장하지 못한다. 또한 제안된 방법은 트래픽을 응용 레벨 프로토콜을 기준으로 분류하는 것을 목적으로 한다. 하지만 응용 레벨 프로토콜에 의한 분류 기준은 HTTP Tunneling과 같이트래픽을 은닉하여 정보를 전달하는 응용프로그램의 트래픽의 분석이 어렵고, 응용프로그램의 통합화로 하나의 응용프로그램이 다양한 종류의 프로토콜에 기반하여 서비스되기 때문에 프로토콜 기준의 분류는 네트워크 관리자에게 효과적인 정보를 제공하기 못한다.

본 논문에서는 시그니쳐 자동 생성 과정을 데이터수집, 시그니쳐 추출, 시그니쳐 유효성 검사 단계로 나누어 각 단계에 대한 시스템을 구축한다. 시그니쳐 생성을 위한 데이터 수집 단계에서는 응용프로그램을 기준으로 트래픽을 자동으로 수집하고, 응용에서 발생한 모든 트래픽에 대한 시그니쳐를 추출하기 위해 응용의 동작별로 트래픽을 그룹한다. 시그니쳐 추출 시스템은 LCS 알고리즘을 기반으로 데이터 수집 단계의 모든 그룹에 대해 시그니쳐를 추출한다. 최종적으로 추출한 시그니쳐를 기존의 유효한 시그니쳐 목록과 충돌 관계를 검사하여 시그니쳐로 등록된다.

3. 응용 레벨 시그니쳐 정의 및 고려사항

본 장에서는 응용 레벨 시그니쳐에 대해 정의하며, 시그니쳐의 추출 및 분류시 요구되는 다양한 고려사항에 대한 기준을 제시한다.

3.1 응용 프로그램 시그니쳐의 정의

응용프로그램 시그니쳐란 응용프로그램 별로 그들만이 사용하여 다른 응용들과 구분되는, 전체 응용프로그램의 트래픽으로부터 해당 응용프로그램을 분류할 수 있는패킷 페이로드내의 고유한 패턴으로 정의된다. 시그니쳐의 패턴은 해당 응용프로그램의 플로우 내의 패킷페이로드에서 순서를 갖는바이트의 나열이다. 시그니쳐는 생성된 위치와 연속되는 스트링의 개수에 따라 다음과 같이 네 가지형태로 구분된다.

- One String with Fixed Offset: 패킷의 고정된 위치에서 하나의 공통된 문자열이 나타나는 형태
- One String with Variable Offset: 패킷의 유동적 위치에서 하나의 공통된 문자열이 나타나는 형태
- Sequence of Strings with Fixed Offset: 패킷의 고정된 위치에서 하나 이상의 문자열들이 순서를 갖고 나타나는 형태
- Sequence of Strings with Variable Offset: 패킷의 유동적 위치에서 두 개 이상의 공통된 문자열들이 순서를 갖고 나타나는 형태

시그니쳐의 형태는 시그니쳐 추출 알고리즘을 결정하는 중요한 요인으로 작용된다. 대부분의 응용프로그램의 시그니쳐는 Sequence of Strings with Variable Offset 형태를 갖는다. LCS(Longest Common Subsequence) 알고리즘은 Sequence of Strings with Variable Offset 형태의 시그니쳐 추출에 적합한 알고리즘이며, One String with Fixed Offset, One String with Variable Offset, Sequence of Strings with Fixed Offset 형태의 시그니쳐의 추출이 가능하기 때문에 본 논문에서는 응용프로그램의 시그니쳐 추출을 위해 LCS 알고리즘을 적용하였다.

시그니쳐의 발생 형태는 분류 시스템의 복잡도에 영향을 미친다. Fixed Offset 형태를 갖는 시그니쳐의 경우 패킷 페이로드에서 고정된 위치의 바이트의 매칭을 통해 시그니쳐 매칭의 복잡도를 감소시킨다. 반면 Variable Offset 형태를 갖는 경우 패킷 페이로드의 모든 바이트와 매칭이 요구되어 복잡도가 증가된다.

3.2 응용 프로그램 시그니쳐 고려사항

3.2.1 시그니쳐의 생성 범위

응용프로그램을 기준으로 분류하는 기존의 다양한 연구[5, 7, 9]에서는 해당 응용프로그램이 발생하는 주요 기능에서 발생하는 트래픽에 대한 분류를 목적으로 시그니쳐를 생성하였다. 하지만 응용프로그램이 통합화되고 복잡해지면서 응용프로그램의 설치, 갱신을 위한 트래픽과 주요 기능을 제공하기 위한 보조적인 트래픽의 양이 증가하고 있기 때문에 이에 대한 분석이 필수적으로 요구되고 있다. 따라서 본 논문에서는 응용프로그램의 발생하는 모든 트래픽을 대상으로 트래픽을 분류하는 시그니쳐를 생성한다.

3.2.2 응용프로그램 vs.응용 레벨 프로토콜

응용 레벨 트래픽 분석의 분류 기준은 그 목적과 활용도에 따라 결정되어야 한다. 기존의 다양한 연구[10-12]에서는 분석 방법에 의존적인 분류 기준을 제시하고 정확성을 측정하고 있다. 하지만 이러한 분류 기준은 Application Monitoring, QoS, SLA, Link Provisioning 등 네트워크 관리 목적에 적합하지 않는 기준을 제공한다. 페이로드 시그니쳐의 특성상 응용 레벨 프로토콜 기준의 분석이 용이하다. 하지만 응용 레벨 프로토콜에 의한 분류는 네트워크에 미치는 위험도가 다르고, 성격이 다른 응용 프로그램이 하나의 프로토콜로 통합되어 분류되는 문제점이 발생한다. 이러한 트래픽 분류 결과는 서비스의 통합화, 응용 레벨 트래픽의 복잡성을 고려했을 때 네트워크 관리자에게 효과적인 정보를 제공하기 어렵다. 따라서 본 논문에서는 객관적이고 구체적인 정보를 제공하기 위해 응용프로그램의 프로세스로 분류 가능한 시그니쳐를 추출하여1차적으로 분류하며, 각 프로세스의 집합으로 구성되는 응용프로그램으로 2차 분류한다.

3.2.3 패킷vs. 스트림

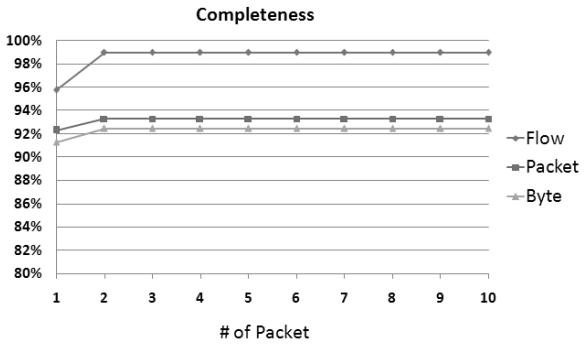
응용프로그램의 시그니쳐는 패킷을 기준으로 생성하는 방법과 플로우 전체의 스트림에서 생성하는 방법으로 나눌 수 있다. 플로우의 스트림에서 생성된 시그니쳐에 기반한 분류 시스템은 트래픽의 분류를 위해 모든 패킷을 스트림 형태로 재조합하는 과정이 요구된다. 때문에 분류 시스템의 부가적인 프로세싱 과정과 저장 공간이 필요하다. 또한 패킷의 손실, 비대칭 라우팅으로 인해 플로우의 스트림이 완전하게 구성되지 못하면 분류가 불가능하게 되는 문제점이 발생한다. 따라서 본 연구에서는 응용 프로그램의 분석의 정확성과 분류시스템의 부하를 고려하여 패킷을 기준으로 시그니쳐를 생성하는 방법으로 응용프로그램 시그니쳐를 생성한다.

3.2.4 단방향 vs. 양방향

단 방향의 시그니쳐를 적용하는 분류 시스템에서 시그니쳐를 포함하고 있는 패킷이 손실되거나 비대칭 라우팅에 의한 패킷 전송이 발생하는 경우 미 분류의 원인으로 작용할 수 있다. 따라서 분류의 정확도를 향상시키고 견고한 분류 시스템을 구축하기 위해서는 양 방향에서 시그니쳐를 생성해야 한다.

3.2.5 시그니쳐 생성을 위한 플로우의 초기 패킷 수

페이로드 시그니쳐 생성 시스템에서 시그니쳐를 추출하기 위해 조사하는 패킷에 개수를 제한한다. (그림 1)은 분류한 플로우에 대한 정확도가 100%인 파일구리의 시그니쳐를 기반으로 동일한 트레이스에 대해 조사하는 패킷의 개수를 증가시키면서 분석률을 측정한 결과이다. 플로우의 두 번째 패킷 이후로 동일한 분석률을 보이는 것을 확인할 수 있다. 이러한 결과를 바탕으로 시그니쳐를 추출하기 위한 플로우의 초기 패킷 개수를 두 개로 제한하였다. 시그니쳐를 추출하기 위한 패킷의 개수에 대한 제한은 시그니쳐 추출 시스템의생성 효율을 높이고, 페이로드 시그니쳐 기반 분류 시스템의 실시간 분석률을 향상시킨다.



(그림 1) 조사하는 패킷 개수에 따른 Completeness

4. 시그니쳐 자동 추출 방법

본 장에서는 3장에서 정의한 시그니쳐와 시그니쳐생성 시 고려사항을 바탕으로 시그니쳐 생성 시스템을 설계하고, 시그니쳐 생성 시스템의 구성을 단계 별로 설명한다.

4.1 시그니쳐 생성 시스템의 자동화 범위

본 논문에서 제안하는 시그니쳐 생성 시스템은 시그니쳐 생성을 위한 데이터 수집과 추출로 구분된다.

기존의 데이터 수집 방법은 시그니쳐 생성을 위한 응용프로그램의 특정 기능을 독립적으로 반복 수행하여 데이터를 수집한다. 이와 같은 방법은 분류 기준에 적합한 정확한 데이터 수집이 어렵기 때문에 시그니쳐의 신뢰도를 보장할 수 없다. 따라서 본 논문에서는 사용자의 컴퓨터에서 발생하는 모든 응용프로그램의 트래픽을 발생 시킨 프로세스를 기준 패킷의 페이로드를 포함한 플로우 단위로 수집한다.

시그니쳐를 추출하기 위해 네트워크 관리자는 수집된 데이터를 바탕으로 데이터를 의미적으로 판단하여 시그니쳐를 추출한다. 이러한 방법은 해당 응용프로그램에서 사용하는 응용 레벨 프로토콜에 대한 사전 지식을 요구하게 된다. 공개되지 않은 응용 레벨 프로토콜을 사용하는 응용프로그램의 수가 증가하고, 응용 프로그램의 변화가 잦은 추세를 고려했을 때 많은 시간이 소비되는 비효율적인 방법이다. 이러한 문제를 해결하기 위한 시그니쳐 추출 시스템은 LCS (longest Common Subsequence) 알고리즘을 기반으로 패킷의 페이로드로부터 동일한 시퀀스를 추출한다.

(그림 2)는 시그니쳐 생성 시스템의 구성과 패킷 수집부터 시그니쳐 생성까지의 과정을 나타내고 있다.

4.1.1 플로우 데이터수집 시스템

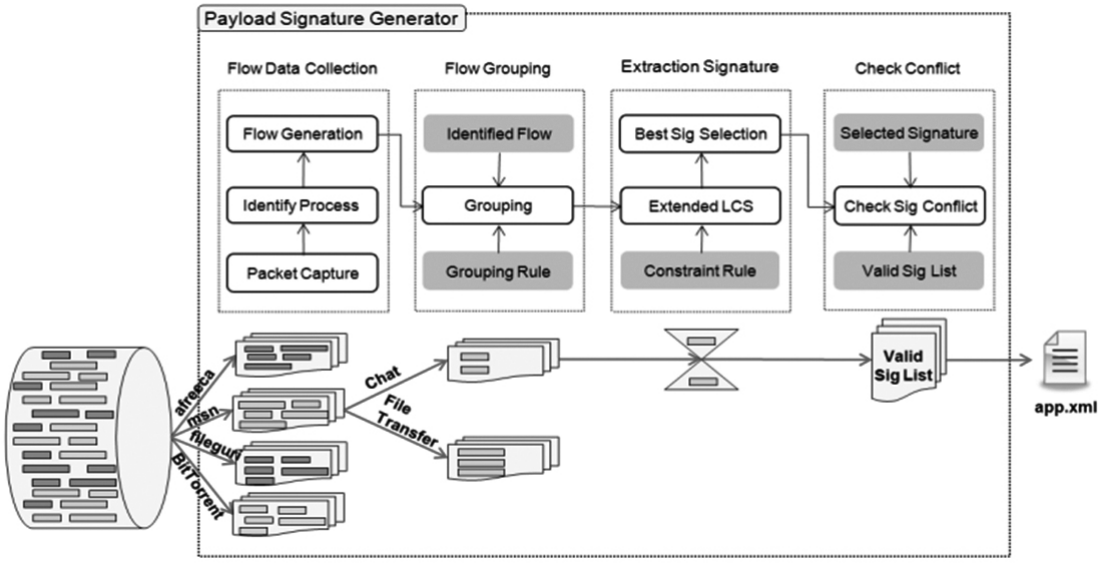
플로우 데이터 수집 시스템은 생성하고자 하는 응용프로그램의 트래픽을 포함한 사용자의 컴퓨터에서 발생한 모든 트래픽을 수집한다. 플로우는 5-tuple(Source IP, Source Port, Destination IP, Destination Port, Transport Layer Protocol)정보를 기준으로 정의되며 해당 플로우를 발생시킨 소켓 정보와 프로세스를 매핑하여 프로세스를 기준으로 페이로드를 포함하는 플로우 데이터가 저장된다. 플로우는 패킷의 집합으로 구성되는데 모든 패킷을 대상으로 시그니쳐를 추출할 필요가 없는 것은 (그림 1)을 통해 증명하였다. 조사하는 패킷의 개수는 LCS 기반의 시그니쳐 추출 시스템의 효율성에 영향을 미치기 때문에 플로우의 초기 두개의 패킷으로 제한한다. 또한 시그니쳐 생성을 위한 불필요한 데이터 수집을 줄이고 수집 시스템의 부하를 줄여 시그니쳐의 정확성과 생성 효율을 향상 시키기 위해 IP, Port, Protocol, Process를 기준으로 트래픽을 수집 가능하도록 구축되었다.

4.1.2 플로우 그룹핑 시스템

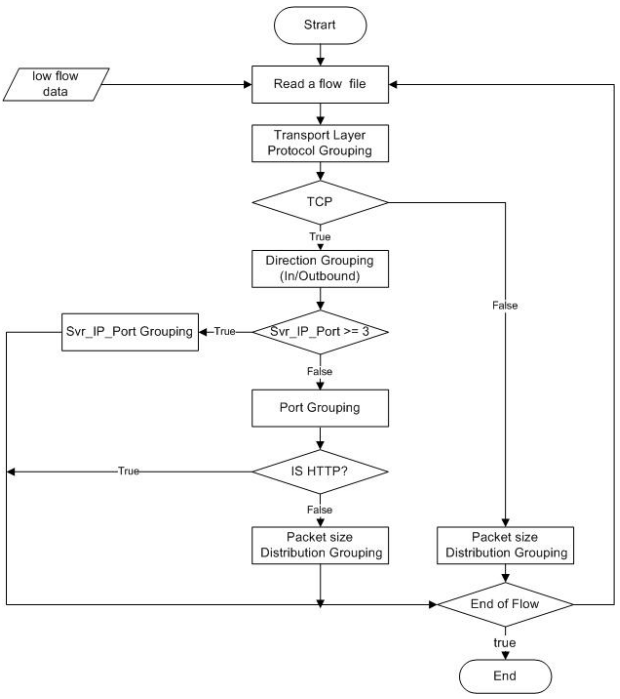
응용프로그램의 트래픽은 다양한 동작 형태로 발생하기 때문에 플로우를 동작별로 그룹하는 과정을 거치지 않고 LCS 기반의 시그니쳐 추출 시스템의 입력으로 사용한다면 부정확한 시그니쳐가 생성되거나 시그니쳐가 생성되지않는다. 따라서 플로우 그룹핑을 통해서 동일한 동작을 수행하는 플로우를 그룹하여 LCS 기반의 시그니쳐 추출 시스템의 입력데이터를 생성하는 플로우 그룹핑이 요구된다. (그림 3)은 플로우 데이터 수집 시스템의 결과를 입력으로 받아 플로우 그룹핑을 수행하는 과정을 나타낸다. 플로우 데이터 수집 시스템의 결과는 그룹핑 규칙에 따라 유사한 동작을 수행하는 플로우 그룹을 생성하고 LCS 기반의 시그니쳐 추출 시스템의 입력 데이터로 적용된다.

포트 기준의 그룹 후 플로우의 첫 번째, 두 번째 패킷의 크기를 기준으로 그룹한다. 포트는 특정 동작을 수행하는 단위로 사용되지만 하나의 포트를 통해 다양한 기능을 제공하는 응용이 존재하기 때문에 패킷 크기를 통한 세분화된 그룹이 요구된다. (그림 4)는 AfreecaTV 플로우에대해 첫 번째, 두 번째 패킷 크기가 동일한 분포를 갖는 플로우의 개수를 나타낸다.

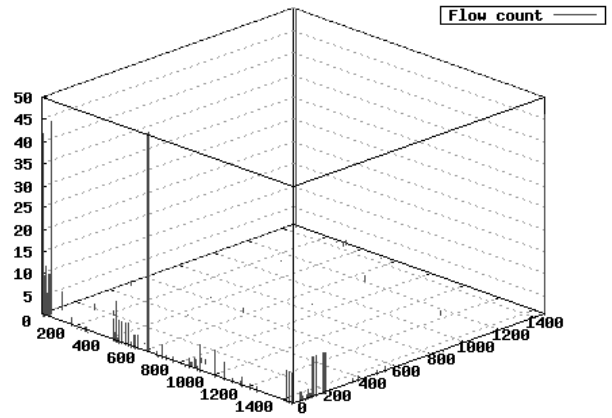
(그림 4)에서 알 수 있듯이 플로우의 첫 번째 패킷과 두



(그림 2) 시그니쳐 생성 시스템 구조



(그림 3) 플로우 그룹핑 순서도



(그림 4) afreecaplayer 패킷 크기 분포

번째 패킷 크기에 대한 분포는 플로우 사이의 유사도를 측정하기 위한 적합한 특징을 갖는다. 클러스터링 문제 해결에 가장 일반적으로 적용되는 Euclidean Distance를 이용하여, 패킷 크기의 분포를 특징으로 플로우 사이의 유사도를 측정하고 유사한 형태의 플로우 그룹을 생성한다.[8] x, y 플로우 사이의 Euclidean Distance는 다음과 같은 식에 의해서 구해진다.

$$Dist(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

n은 패킷의 크기를 적용하는 플로우 내의 패킷 순서를 나타내며, x와 y는 각각의 플로우를 대표한다.

4.1.3 LCS 기반 시그니쳐 추출 시스템

LCS(Longest Common Subsequence)는 3장에서 정의한 페이로드 시그니쳐를 추출하기 위해 적합한 알고리즘이다. LCS 문제를 해결하기 위한방법 중 Brute force 알고리즘은 ($n2^m$) 시간복잡도와 추가적인 저장 공간을 요구하는 알고리즘으로 입력 데이터가 크고 복잡한 응용프로그램의 트래픽에서 LCS를 기반으로 시그니쳐를 추출하기에 부적합한 알고리즘이다. 따라서 본 논문에서는 LCS 문제를 해결하기 위해 추가적인 저장 공간만을 요하고 상수 시간에 문제를 해결할 수 있는 Dynamic program 방법을 선택하였다.

(그림 5)는 LCS 기반 시그니쳐 추출 시스템의 흐름도를 나타내고 있다.

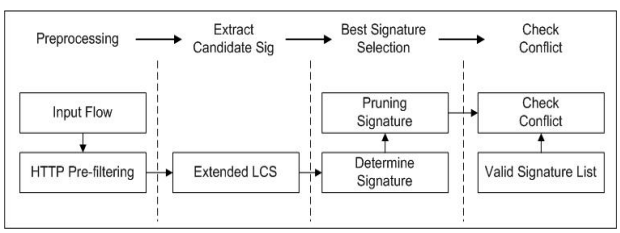
HTTP Pre-filtering 단계는 HTTP플로우에 대한 시그니쳐의 정확성을 높이고, 추출 시스템의 부하를 줄이기 위한 단계이다. 시그니쳐 추출 결과 HTTP를 사용하는 응용프로그램의 시그니쳐는 요청하는 파일에 대한Path정보, 서버의 URL 정보에서 추출되었다. 이 단계에서는 파일 경로, URL 정보를 포함하는 스트링을 제외한 컨텐츠를 제거하는 역할을 수행한다.

Extract Candidate Signature 단계는 LCS알고리즘을 기반으로 가능한 모든 후보 시그니쳐를 추출하는 단계이다. 기본적인 LCS에 의해서 추출되는 시그니쳐는 하나의 결과로 나타나지만, 실제 LCS에 의해서 추출 가능한 결과는 다양한 경우의 수로 나타난다. 따라서 추출 가능한 시그니쳐에 대한 모든 경우의 수를 고려해야 정확한 응용프로그램 시그니쳐의 추출이 가능하다. 아래의 예는 이와 같은 문제점을 간단한 입력 데이터를 통해 보여 주고 있다.

X = <*A, B, C, B, D, A, B*>
Y = <*B, D, C, A, B, A*>

X와 Y 두 개의 입력 스트링으로부터 추출 가능한 LCS는 <*B,C,B,A*>, <*B,C,A,B*>, <*B,D,A,B*>으로 다양한 경우의 수로 나타난다. (그림 6)은 기본적인 LCS 알고리즘을 수정하여 모든 가능한 경우에 대한 경로를 추출하기 위한 개선된 LCS알고리즘의 pseudo code를 보여주고 있다.

Best Signature Selection 단계는 Extract Candidate Signature 단계를 통해 생성된 후보 시그니쳐에서 최적의 시그니쳐를 찾는 과정이다. 이 단계는 두 가지 구성된다. 첫 번째 단계는 후보 시그니쳐로부터최적의 시그니쳐를 추출하는 과정이



(그림 5) LCS 기반 시그니쳐 추출 시스템 흐름도

```
1:    Extended-LCS(X,Y)
2:
3:      m ← length[X]
4:      n ← length[Y]
5:      for i ← 1 to m
6:           do C[i,0] ← 0
7:      for j ← 0 to n
8:           do C[0,j] ← 0
9:      for i ← 1 to m
10:          do for j ← 1 to n
11:               do if Xi = Yj
12:                    if C[i-1,j]=C[i-1, j-1]
13:                         B[i,j] = “ “ & “ “
14:                    if else C[i, j-1]=C[i-1, j-1]
15:                         B[i,j] = “ “ & “ “
16:                    If else C[i, j-1]=C[i-1, j-1]=C[i-1, j]
17:                         B[i,j] = “ “ & “ “ & “ “
18:               else if C[i-1, j] >= C[I, J-1]
19:                    then C[I, j] = C[i-1,j]
20:                         B[I,j] = “ “&“ “
21:               else C[I, j] = C[I, j-1]
22:                    B[I, j] = “ “
23:     return C and B
```

(그림 6) 수정된 LCS 알고리즘의 pseudo code

다. 다음과 규칙을 통해 최적의 시그니쳐를 선택한다.

- 길이가 가장 긴 substring이 존재하는 경우
- 페이로드 시작 부분을 포함하는 경우

두 번째 단계는 첫 번째 단계에서 선택된 시그니쳐에서 시그니쳐로서 변별력을 갖지 못하는 substring을 제거하는 과정이다. 이때 적용되는 규칙은 다음과 같다.

- Substring의 최소 길이는 2바이트로 제한
- Common substring 제거

시그니쳐가 결정되면 check conflict 단계에서는 기존의 유효한 시그니쳐 목록과 충돌 관계를 검사하여 최종적으로 시그니쳐 기반 분석 시스템의 입력데이터로 제공된다.

5. 시그니쳐의 생성 결과

본 장에서는 기존 연구에서 제시하고 있는 응용 레벨 프로토콜의 문법에 기초하여 수작업으로 생성한 페이로드 시그니쳐와 생성 시스템에 의한 페이로드 시그니쳐를 비교함으로써 그 타당성을 증명한다.

<표 1>은 utorrent의 프로토콜 분석에 기반하여 추출한 시그니쳐와 제안한 페이로드 시그니쳐 추출 시스템에 기반한 시그니쳐 생성 결과를 나타낸다.

<표 1>에서 알 수 있듯이 페이로드 시그니쳐 생성 시스템에 기반하여 추출한 시그니쳐는 프로토콜 분석에 의해 추출한 시그니쳐와 응용프로그램이 발생하는 보조 트래픽에 대한 시그니쳐를 포함한다. 또한 패킷의 순서와 페이로드내의 offset 정보를 포함하여 시그니쳐의 정확성을 향상 시킬 수 있었다.

〈표 1〉 시그니쳐 생성 시스템에 기반한 시그니쳐

App	Method	Payload Signature
utorrent	Protocol Analysis	\x13BitTorrent protocol
	Payload Signature Generator	[1]^\x13BitTorrent protocol
		[1]^GET/@peer_ip@User-Agent: uTorrent/@HTTP1\.1
		[1]utorrent\.com
		uTorrent
		[1]^\x64\x31\x3A.\x64\x32\x3A\x69\x64\x32\x30\x3A@\x3A@\x31\x3A\x79\x31\x3A.\x65

6. 시그니쳐 검증 및 갱신 시스템

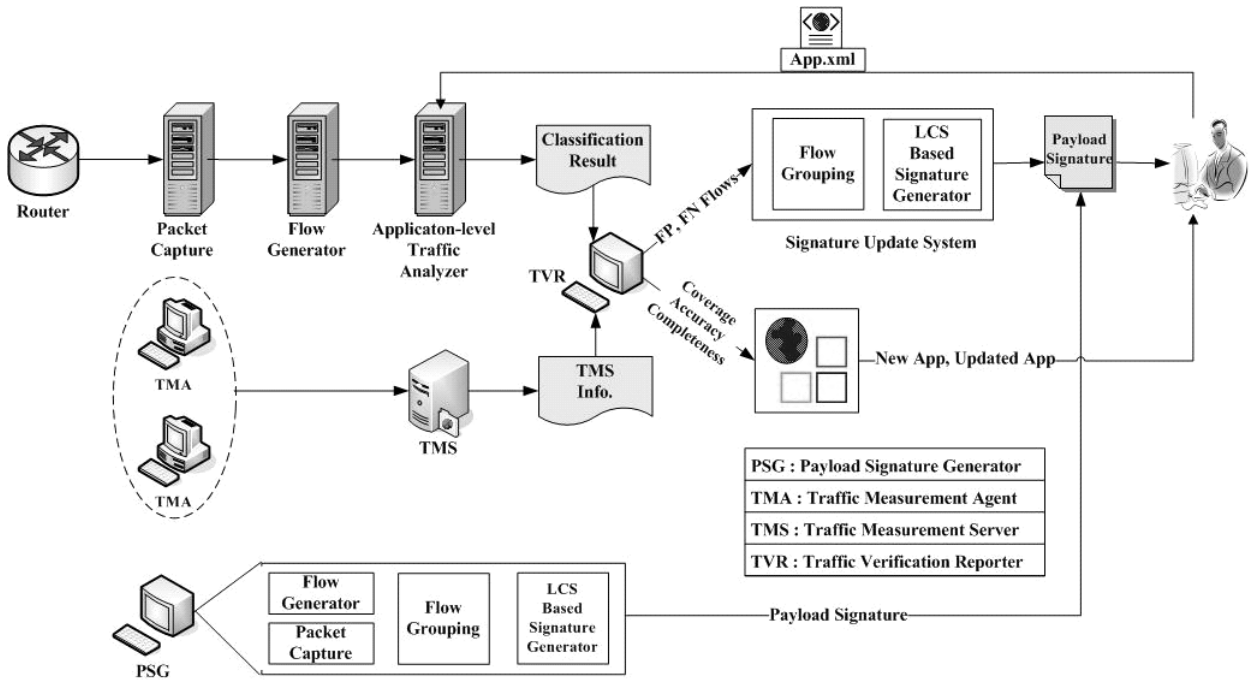
기존의 다양한 연구에서 트래픽 분류 방법론을 제안하고 분류 결과를 검증하고 있지만, 분류 기준이 불명확하고 가정에 기반한 트래이스를 대상으로 결과를 검증하고 있다. 이러한 검증 방법은 정확한 Ground Truth를 제공하지 못하기 때문에 검증 결과의 신뢰성을 보장할 수 없다. 이러한 문제점을 해결하고 시그니쳐 생성 시스템의 타당성을 증명하기 위해 학내망에서 발생하는 모든 트래픽을 대상으로 실시간 검증 시스템을 구축하였다. 또한 검증된 결과를 바탕으로 오분류되거나 미분류되어진 트래픽을 대상으로 시그니쳐를 추출하여 응용의 변화에 유연하게 대처할 수 있는 시그니쳐 갱신 시스템을 구축하였다.

6.1 검증 시스템 및 갱신 시스템 구조

검증 및 갱신 시스템의 구조는 (그림 7)과 같이 트래픽 수집 시스템, 페이로드 기반 트래픽 분류 시스템, TMA (Traffic Measurement Agent)[14], TMS(Traffic Measurement Server), TVR(Traffic Verification Reporter), 시그니쳐 갱신 시스템으로 구성된다. 검증 시스템은 고속 링크의 대용량 트래픽을 실시간으로 처리하고, 다양한 트래픽 분류 방법의 동시 적용을 위해 분산 시스템 환경으로 구성된다.

생성된 시그니쳐를 기반으로 분류된 결과는 관리자에게 Accuracy, Completeness, Coverage를 제공하며, 또한 오 분류되거나 미 분류된 플로우를 수집하여 오 분류된 플로우에 대해서는 분류 로그 정보를 제공하고, 미 분류된 플로우는 시그니쳐 갱신 시스템의 입력 데이터로 제공하여 플로우 그룹핑 단계, LCS 기반의 시그니쳐 추출 단계, 시그니쳐 유효성 평가 단계를 거쳐 응용프로그램의 시그니쳐를 갱신한다. 또한 TMA를 기반으로 수집된 응용의 정보에 기초하여 새로운 응용의 출현에 대한 리포팅을 통해 시그니쳐 추출 대상 응용프로그램을 선정하고 시그니쳐 생성 시스템을 기반으로 시그니쳐 생성하고등록하게 된다. 현재의 응용프로그램 변화 주기를 고려했을 때 갱신 시스템의 필요성은 더욱 커지고 있다.

결과의 정확성을 검증하기 위해 TMA 기반으로 Ground Truth 트래픽을 수집한다. TMA는 학내망의 단말 호스트에 설치되며 소켓 정보를 기반으로 하여 Process name, IP, port,



(그림 7) 검증 및 갱신 시스템 구조

〈표 2〉 분류 알고리즘 검증 요소

검증 항목	검증 시간	검증 단위	검증 요소					
Accuracy	minhourday	flowbyte-packet	TP	TN	FP	FN	Precision	Recall
Comple-teness			Amount					

〈표 3〉 Coverage 결과

분류 단위	분류 대상 개수	
	07.28 00:00	08.03 11:59
응용프로그램	126	126

protocol, path 등의 정보를 생성한다. TMA가 설치된 호스트에서 열려진 소켓을 주기적으로 검사하여 TMS로 TMA 정보를 전송하고 TMS는 각 호스트로부터 전달받은TMA 정보를 통합하여 분류 시스템의 분류 결과의 Ground Truth를 제공한다.

TVR은 시그니쳐 기반 분류 시스템의 분류 결과와 TMS 정보를 비교하여 전체 트래픽, 응용프로그램 단위 검증 결과를 웹을 통해 실시간으로 제공한다. TVR에서는 제공하는 정보는 선행 연구[13]에서 정의하였으며 <표 2>와 같다.

6.2 트래픽 분류 결과

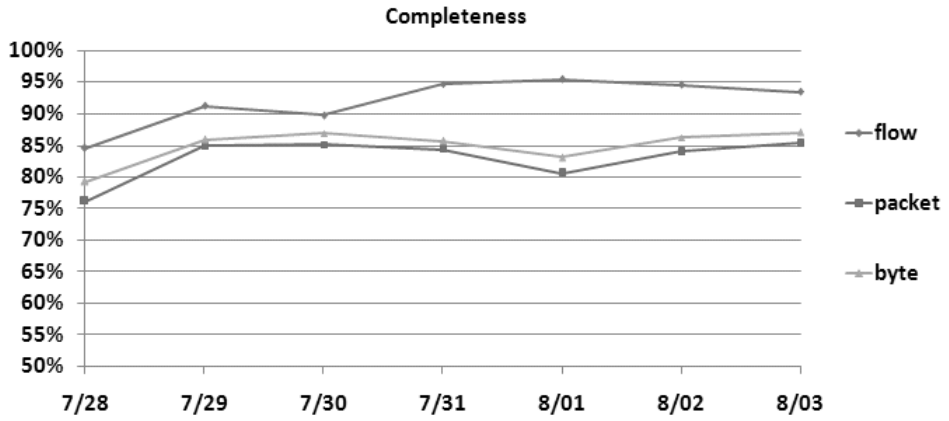
Coverage, Accuracy, Completeness는 시그니쳐의 실효성 및 정확성을 평가하기 위한 Metric으로 사용된다. 실험 결과는 2009년 07월 28일 00시 00분부터 2009년 08월 03일 11시 59분까지의 연속적인 트래픽 트레이스를 바탕으로 결과를 보여주고 있다.

ⅰ Coverage : 분류 가능한 범위

시그니쳐 생성 시스템을 이용하여 학내 망에서 발생하는 응용프로그램들을 대상으로 <표 3>과 같은 분류 범위를 갖는 시그니쳐를 추출하였다. 갱신 시스템의 미 분류 트래픽에 대한 시그니쳐 추출 결과 실험 종료 시점의 Coverage가 증가하였다.

ⅱ. Completeness: 전체 트래픽 중 분류된 양

Completeness는 분류 가능한 범위(Coverage) 내에서 분류된 트래픽의 양을 나타낸다. (그림 8)은 실험 기간 동안 학내망 전체에서 발생된 트래픽의 Completeness를 나타내고 있다. 미 분류 트래픽은 시그니쳐의 높은 신뢰도를 고려했을 때 대부분 Coverage에 포함되지 않는 응용프로그램에 의해서 발생되는 트래픽으로 분석되었다. 다른 원인으로는 응용 프로그램이 변화하면서 Coverage내의 응용이 분석되지 않는 경우이다. 이러한 트래픽은 갱신 시스템에서 미분류 트래픽으로 수집되고, 시그니쳐 추출 후 등록되기 때문에 Completeness는 다시 향상되었다. 분류 결과를 살펴보면 플로우, 패킷, 바이트의 증감의 차가 다른 것을 알 수 있다. 이는 발생하는 응용의 종류에 따라서 트래픽의 발생 형태가 다르기 때문이다. 검색 트래픽이 많이 발생하는 P2P 트래픽의 경우 패킷과 바이트에 비해 상대적으로 많은 양의 플로우를 발생시키며, 웹하드의 경우 적은 수의 플로우로 대용량의 트래픽을 발생한다. 실험 시작일보다 종료시점에 22개의 시그니쳐가 증가하였는데 추가된 시그니쳐는 생성 당시 발생하지 않았던 응용의 업데이트 트래픽이 대부분이었다. 최근에는 응용의 업데이트가 잦은 점을 고려했을 때 시그니쳐의 개수는 지속적으로 증가할 것으로 예상된다.



(그림 8) Completeness 결과

iii. Accuracy: 분류의 정확도

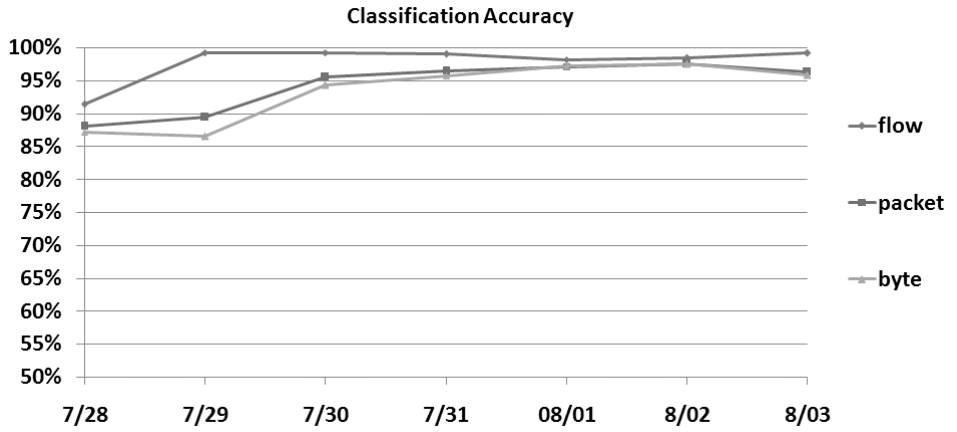
(그림 9)는 시그니쳐의 정확성을 평가하는 Acccuracy에 대한 측정 결과를 나타낸다.

Accuracy의 측정 결과를 통해서 시그니쳐 갱신 시스템의 중요성을 확인할 수 있다. 07월 28일 시그니쳐 생성 시스템에 기반한 시그니쳐를 기반으로 학내망 전체 트래픽을 분류한 후 오분류, 미분류된 플로우를 대상으로 갱신시스템을 통해 시그니쳐를 변경하였다. 그 결과 시그니쳐의 정확도가 향상되었고 95%이상의 높은 정확도를 유지하고 있는 것을 확인할 수 있었다. 오 분류의 발생 원인은 특정 응용프로그램의 시그니쳐에 의해서 Internet Explorer의 트래픽을 분류하는 것으로 분석되었다. 이는 특정 응용프로그램에 내장되어 부가적으로 발생하는 Internet Explorer의 트래픽과의 충돌이 발생하기 때문이다. 미 분류의 발생 원인은 세가지 원인으로 분석된다. 첫째, 패킷의 페이로드가 존재하지 않은 트래픽의 경우 페이로드 시그니쳐를 적용할 수 없기 때문에 미분류 되었다. 둘째, 페이로드의 암S호화로 인해 시그니쳐 추출이 불가능한 트래픽이 존재하였다. 셋째, Coverage 외의 응용프로그램에 의해서 발생된 트래픽을 분류하지 못 하였다.

7. 결론 및 향후 과제

본 논문에서는 페이로드 시그니쳐 생성 과정의 문제점을 보완하고 시그니쳐 생성의 효율을 높이기 위해 LCS에 기반한 시그니쳐 생성 시스템을 제안하였다. 또한 응용프로그램의 변화에 따른 시그니쳐가 변경되는 문제에 대처하기 위해 검증 네트워크에 기반한 시그니쳐 갱신 시스템을 구축하였다. 시그니쳐 생성 시스템을 기반으로 추출한 시그니쳐를 학내망 전체 트래픽에 실시간으로 적용하여 95%이상의 정확도와 80%이상의 분석률을 보이는 시그니쳐를 추출함으로써 시그니쳐 생성 시스템의 실효성과 타당성을 증명하였다.

응용프로그램의 변화와 출현에 대해 네트워크 관리자가 수동적으로 인식하고 대처하기에는 많은 인력과 시간이 요



(그림 9) Accuracy 결과

구된다. 이러한 문제점을 해결하기 위해 검증 네트워크에 데이터 수집, 시그니쳐 추출, 시그니쳐 갱신 시스템을 통합하고 시그니쳐를 데이터베이스화 하여 주기적으로 시그니쳐의 갱신이 이루어지는 시스템을 구축할 계획이다.

참 고 문 헌

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

[2] Jacobus van der Merwe, Ramon Caceres, Yang-hua Chu, and Cormac Sreenan "mmdump- A Tool for Monitoring Internet Multimedia Traffic," ACM Computer Communication Review, 30(4), October, 2000.

[3] Hun-Jeong Kang, Myung-Sup Kim, and James Won-Ki Hong, "Streaming Media and Multimedia Conferencing Traffic Analysis Using Payload Examination," ETRI Journal, Vol.26, No.3, Jun., 2004, pp.203-217.

[4] TS Choi, CH Kim, SH Yoon, JS Park, HS Chung, BJ Lee, HH Kim, and TS Jeong, "Rate-based Internet Accounting System Using Application-aware Traffic Measurement," APNOMS 2003, Fukuoka, Japan, October 1-3, 2003, pp.404-415.

[5] Byung-Chul Park, Young J. Won, Myung-Sup Kim, James W. Hong, "Towards Automated Application Signature Generation for Traffic Identification", NOMS 2008, Salvador, Bahia, Brazil, April, 7-11, 2008, 160-167.

[6] Subhabrata Sen , Oliver Spatscheck , Dongmei Wang, "Accurate, scalable in-network identification of p2p traffic using application signatures" World Wide Web 2004, May 17-20, 2004, New York, USA.

[7] Patrick Haffner, Subhabrata Sen, Oliver Spatscheck, Dongmei Wang, "ACAS: automated construction of application signatures," ACM SIGCOMM, August 26-26, 2005, Philadelphia, Pennsylvania, USA.

[8] Jeffrey Erman, Martin Arlitt , Anirban Mahanti, "Traffic classification using clustering algorithms," SIGCOMM 2006, September 11-15 2006, Pisa, Italy pp.281-286.

[9] Xiao, F., Hu, H. "ASG - Automated signature generation for worm-like P2P traffic patterns," WAIM 2008. July 20-22 2008. pp.645-660.

[10] Hui Liu, Wenfeng Feng, Yongfeng Huang, and Xing Li, "A peer-to-peer traffic identification method using machine learning" ProcGuilin , China, July, 29-31, 2007, pp.155-160.

[11] TS Choi, SH Yoon, HS Chung, JS Park, BJ Lee, SS Yoon, and TS Jeong, "Flow-based Application-aware Internet Traffic Monitoring and Field Trial Experiences" APNOMS, Okinawa, Japan, Sep., 27-30, 2005, pp.214-225.

[12] Andrew W. Moore, Denis Zuev, "Internet traffic classification using bayesian analysis techniques", ACM 2005, Banff, Alberta, Canada, June, 06-10, 2005.

[13] Sung-Ho Yoon, Jin-Wan Park, Young-Seok Oh, Jun-Sang Park, and Myung-Sup Kim, "Internet Application Traffic Classification Using Fixed IP-port," APNOMS 2009, LNCS, Jeju, Korea, Sep., 23-25, 2009, pp.21-30.

[14] 윤 성호, 노현구, 김명섭, "TMA(Traffic Measurement Agent)를 이용한 인터넷 응용 트래픽 분류", 2008년 제29회 정보처리학회 춘계학술발표대회, 대구, 경일대학교, May, 17, 2008, 제15권 제1호, pp.946-949.

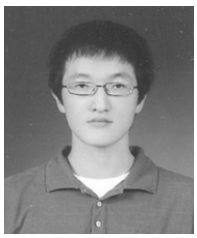
박 준 상

e-mail : junsang_park@korea.ac.kr

2008년 고려대학교 컴퓨터정보학과(학사)

2008년～현 재 고려대학교 컴퓨터정보학과 석사과정

관심분야 : 네트워크 관리 및 보안, 트래픽 모니터링 및 분석



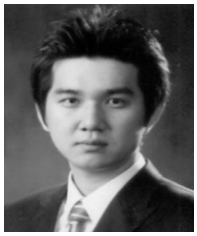
박 진 완

e-mail : jinwan_park@korea.ac.kr

2009년 고려대학교 컴퓨터정보학과(학사)

2009년～현 재 고려대학교 컴퓨터정보학과 석사과정

관심분야 : 네트워크 관리 및 보안, 트래픽 모니터링 및 분석



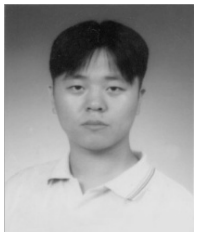
윤 성 호

e-mail : sungho_yoon@korea.ac.kr

2009년 고려대학교 컴퓨터정보학과(학사)

2009년～현 재 고려대학교 컴퓨터정보학과 석사과정

관심분야 : 네트워크 관리 및 보안, 트래픽 모니터링 및 분석



이 현 신

e-mail : hyunshin_lee@korea.ac.kr

2009년 고려대학교 정보수학과(학사)

2009년～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 조교수

관심분야 : 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 멀티미디어 네트워크