http://dx.doi.org/10.7840/kics.2015.40.3.575

페이로드 시그니처 매칭 순서 최적화를 통한 응용 트래픽 분류 속도 향상

이 성 호◆, 박 준 상*, 김 명 섭°, 석 우 진°°

Application Traffic Identification Speed Improvement by Optimizing Payload Signature Matching Sequence

Sung-ho Lee◆, Jun-sang Park*, Myung-sup Kim°, Woojin Seok°°

요 약

응용 레벨 트래픽 분류는 안정적인 네트워크 운영과 자원 관리를 위해서 필수적으로 요구된다. 트래픽분류에 있어 페이로드 시그니처 기반 응용 레벨 트래픽 분류 방법은 고속 링크의 트래픽을 실시간으로 처리하는 과정에서 헤더 정보 및 통계 정보 이용 방법론에 비해 상대적으로 높은 부하를 발생시키고 시그니처 개수가 증가 할수록 트래픽의 발생 특징과 각 시그니처의 가치를 반영하지 않은 매칭 방법 때문에 분석 속도가 감소하는 단점이 있다. 본 논문에서는 페이로드 시그니처 기반 응용 트래픽 분석 속도의 향상을 위하여 각 페이로드 시그니처 별 트래픽 분석 효율을 고려하여 리스트에 존재하는 시그니처 순서를 재정렬 하는 방법을 제안한다. 제안하는 방법은 재정렬되지 않은 시그니처 리스트를 적용했을 때 보다 평균 30%정도의 분석 속도 향상을 얻을 수 있었다.

Key Words : traffic analysis, signature matching, Torrent, Identification, network management

ABSTRACT

The traffic classification is a preliminary and essential step for stable network service provision and efficient network resource management. However, the payload signature-based method has significant drawbacks in high-speed network environment that the processing speed is much slower than other methods such as header-based and statistical methods. In addition, as signature numbers are increasing, traffic analysis speed also declines because of signature matching method that does not consider analytic efficiency of each signature and traffic occurrence feature. In this paper, we propose a signature list reordering method in order by analytic value of each signature. When we reordered the signature list by the proposed method, we achieved about 30% improvement in speed of the traffic analysis compared with random signature list.

※ 이 논문은 2012년 정부(교육과학기술부)의 재원으로 한국연구재단(2012R1A1A2007483) 및 2015년 한국과학기술정보연구원 연구과제(첨단연구망 기반 협업플랫폼 서비스 기술 개발 및 적용, K15-L01-C004-S03) 지원으로 수행되었습니다.
◆ First Author : Korea University Department of Computer and Information Science, gaek5@korea.ac.kr, 학생회원
° Corresponding Author : Korea University Department of Computer and Information Science, tmskim@korea.ac.kr, 종신회원
°° Corresponding Author : University of Science & Technology, wjseok@gmail.com, 정회원
* Korea University Department of Computer and Information Science, junsang_park@korea.ac.kr, 학생회원
논문번호 : KICS2014-11-450, Received November 3, 2014; Revised Feburary 3, 2015; Accepted March 16, 2015

Ⅰ. 서 론

네트워크의 고속화와 더불어 다양한 서비스와 응용 프로그램이 개발됨에 따라 기업이나 개인들은 인터넷으로 대표되는 네트워크에 대한 의존이 상당히 커져가고 있다. 이와 같은 현실 속에서 네트워크의 효율적 운용과 관리를 위한 응용 레벨 트래픽의 모니터링과 분석은 네트워크 사용현황 파악과 확장계획 수립 등의 다양한 분야에서 필요성이 커져가고 있다. 이를 위해서는 다양한 종류의 응용 레벨 트래픽을 정확하게 그리고 빠른 시간 안에 분류할 수 있는 방법이 요구된다.

빠르고 정확한 응용 레벨 트래픽 분류를 위해 시그니쳐 분석 방법은 전통적으로 보안, 응용 트래픽 분류 등의 다양한 분야에서 활용되어 왔다. 시그니쳐 기반 분석 방법은 패킷 또는 플로우의 특정 위치에서 응용을 식별하기 위한 고유한 정보를 추출하여 이를 기반으로 분류하는 방법이다. 시그니쳐는 그 추출 위치와 가공 방법에 따라 Header[3], Payload[4], Statistic[5], Behavior[6]시그니쳐로 구분할 수 있다. 본 논문에서는 페이로드 시그니쳐 기반 분석 방법으로 적용 범위를 제한한다.

응용 레벨 트래픽 분류 방법에 있어 페이로드 시그니쳐 기반 분석 방법은 패킷의 헤더 정보나 통계 정보를 이용하는 다른 분석 방법들에 비해 상대적으로 높은 분류 정확성과 분석률을 보인다[1,2].하지만 시그니쳐를 찾는 작업이 수작업으로 이루어져 응용 프로그램의 변화에 적절히 대처하지 못한다는 단점을 가지고 있다. 또한, 암호화된 트래픽과 같이 시그니쳐를 확인하기 힘든 응용 프로그램들에 대해서는 많은 제약이 따른다. 따라서 현재의 고속 네트워크상에서 발생하는 대용량 트래픽을 원활히 처리하기위해서는 보다 빠른 방법이 요구 된다. 대용량의 트래픽을 발생시키는 응용수의 증가를 고려했을 때, 페이로드 기반 분석 방법의 처리 속도 향상은 언제나 중요한 과제이다. 이를 해결하기 위해 기존의 다양한 연구에서는 패턴 매칭 알고리즘의 성능 개선 기법에 대한 연구가 주를 이뤘다[7-10,11]. 하지만 매칭 알고리즘의 성능 개선은 제한적이며 현재의 고속 링크의 대용량 트래픽을 수용하는 부분에서 한계를 보이고 있다.

본 논문에서 응용 트래픽 분석 시스템 내에 존재하는 페이로드 시그니쳐 리스트를 재정렬 시켜 분석 성능을 향상 시킬 수 있는 방법을 제안한다. 페이로드 시그니쳐의 개수가 증가 할수록 응용 트래픽 분석률은 증가했지만, 각 시그니쳐들의 분석 효율을 고려하지 않은 순차적인 시그니쳐 리스트 탐색, 매칭 방법 때문에 분석 시간 증가해 결과적으로 분석 속도가 떨어지게 되는 문제가 발생했다. 본 논문에서 이러한 현상을 시그니쳐 리스트의 비효율성이라 정의하고, 이를 해결하기 위한 효율적인 시그니쳐 리스트 재정렬 방법을 제안한다.

본 논문의 구성은 다음과 같다. 본 장의 서론에 이어, 2장에서는 관련연구에 대해 기술하고, 3장에서는 제안하는 방법의 배경이 되는 기존의 시그니쳐 매칭 방법의 문제점들에 대해 정의한다. 4장에서는 시그니쳐 리스트 재정렬을 위한 방법론에 대해 기술한다. 5장에서는 제안하는 방법을 통해 제작된 재정렬 시그니쳐 리스트를 응용 트래픽 분석 시스템에 적용하고 성능 평가를 통해 그 효율성을 증명한다. 마지막으로 6장에서는 결론 및 향후 연구에 대해 기술한다.

Ⅱ. 관련 연구

응용 프로그램 서비스 제공자는 방화벽을 우회하여 사용자에게 원활한 서비스를 제공하기 위해 복잡한 구조의 응용 레벨 프로토콜 구성하기 때문에 시그니쳐 또한 복잡하고 다양한 형태로 나타난다. 또한 인터넷에 기반한 응용의 증가로 인해 시그니쳐의 개수가 증가하고 그 가치 또한 높아지고 있다. 시그니쳐의 복잡도가 커지고, 개수가 증가하면서 페이로드 시그니쳐 기반 분류 시스템의 처리 속도는 트래픽 분류 시스템의 성능을 결정하는 중요한 요소로 작용하게 되었다.

분류 시스템의 속도 향상을 위한 다양한 패턴 매칭 알고리즘들이 제안되었지만 패턴 매칭 알고리즘의 성능은 입력 데이터의 구성에 의존적이며, 제한적인 성능 향상을 나타낸다[8]. 오토마타에 기반한 NFA(Non-deter-ministic Finite Automata)와 DFA(Deterministic Finite Automata) 알고리즘의 성능 개선을 위한 방법론들이 제시되고 있지만 오토마타를 이용한 방법은 '.*'와 같은 와일드 카드의 사용빈도에 따라 시간 및 공간 복잡도 급격하게 증가하여 성능이 저하되는 문제점이 있다[9,10].

페이로드 시그니쳐 매칭 과정의 복잡도를 줄이기 위해 HTTP 프로토콜의 필드를 Domain, Host, URI로 구분하여 각 필드의 시그니쳐를 자동으로 추출하여 Hash 기반으로 분석하는 Field-based-filter 방법 또한 제시되었다[11]. Hash를 이용하여 시그니쳐를 필터링하여 시그니쳐의 탐색 공간을 줄이고, 페이로드를 필드 단위로 매칭하기 때문에 전체 페이로드를 검사하는 부하를 줄일수있다. 하지만 Hash key의 비교를 위해서는 시그니쳐의 추출 시점에서 시그니쳐 추출

방법과 분류 시점에서 페이로드 필드 추출 방법이 동일해야만 적용이 가능하다는 제약이있다.

패턴이 매칭되는 offset에 초점을 맞춘 연구에서는[15] 패킷의 페이로드 내에서 시그니쳐의 존재 유무를 검사하는 범위를 특정 offset을 기준으로 제한함으로써 분류 시스템의 처리 속도를 향상 시킬 수 있었다. 그러나 빠르게 변화하는 응용 트래픽에 대해 고정적인 offset을 적용할 경우 다양한 트래픽에 유연하게 대처할 수 없어 분석률이 떨어지는 문제점이 존재한다.

기존의 매칭 알고리즘 성능 개선의 한계적 문제점을 해결하기 위해 응용 레벨 트래픽의 발생 패턴을 분석 시스템에 반영하여 시그니쳐의 탐색 공간을 최소화하는 방법이 제안되었다[16]. 트래픽의 발생 패턴이나 특징을 반영해 분석 효율을 높인다는 점에서 본 논문에서 제안하는 방법과 유사하다. 그러나 기존의 연구에서는 각 시그니쳐의 HC(Hit Count)만을 고려했고 실질적으로 시그니쳐 리스트를 발생 패턴에 따라 재정렬 하는 방법에 대해서는 제시하지 못하였다. 결과적으로 탐색 공간은 최소화 했지만 페이로드 시그니쳐 매칭 과정에 있어서는 여전히 비효율적이다.

멀티코어 CPU의 장점인 병렬처리를 이용한 방법과 하드웨어적인 측면에서 NFA기반 패턴 매칭 알고리즘을 FPGA(Field Programmable Gate Array)로 구현한 방법 또한 제시되었다.[12,13] 하지만 하드웨어 기반 분석 방법은 고가의 비용을 요구되며, 현재와 같이 변화가 잦고 복잡한 환경에서 실시간으로 동작하기 위해서는 소프트웨어 적인 방법에 비해 고려해야 할 사항들이 많기 때문에 다양한 환경에 적용하기 힘들다는 단점이 있다.

시그니쳐 패턴 매칭 알고리즘에 의존하지 않고 트래픽 간의 지역적인 연관성을 이용한 분석 방법도 제시되었다.[14]하지만 CDN(Content Delivery Network) 트래픽과 같은 상호 연관성은 존재하지만, 서로 다른 응용을 나타내는 트래픽을 분석하는 데에는 그 한계점이 존재한다.

기존의 연구는 페이로드 시그니쳐 기반 트래픽 분류 시스템의 처리 속도 향상을 위해서 패턴 매칭 기법을 소프트웨어 또는 하드웨어적으로 개선하려는 노력과 트래픽의 특징을 정의하고 그룹화해서 시그니쳐 탐색 범위를 최소화 하는 방법이 주를 이루었다. 하지만 이러한 방법은 네트워크 대역폭 증가에 비해 상대적으로 제한적인 성능 향상을 보이거나 현재의 네트워크 환경에 적용하기에는 난해하기 때문에 활용도가 떨어진다.

Ⅲ. 기존 시그니쳐 매칭 방법의 문제점

본 장에서는 기존의 시그니쳐 매칭 방법이 갖고 있는 특징들과 문제점에 대해 정의하고 시그니쳐 리스트 재정렬이 필요한 이유를 제시한다.

그림 1은 일반적인 플로우 단위 응용 트래픽 분석 시스템의 전체적인 구성을 나타내고 있다. 시스템의 개요는 수집된 트래픽(Traffic trace)에 시그니쳐 리스트(Sig list)를 매칭 시킨 뒤 매칭된 결과를 바탕으로 분석률을 평가하고 추가적인 시그니쳐를 추출해 리스트에 추가하는 과정이다.

일반적인 트래픽 응용 분류 시스템에서는 단위 플로우에 대해 응용 시그니쳐가 매칭되면 매칭 시퀀스를 끝내는 부분 매칭(Partial-matching)방식을 사용한다. 반면 그림 1의 분석 시스템에서는 단위 플로우에 모든 시그니쳐를 매칭 시키는 풀 매칭(Full-mathcing) 방식을 사용한다. 풀 매칭 방식을 사용 할 경우 플로우에 다수의 시그니쳐가 매칭되는 중복과 충돌 매칭 현상이 불가피하게 발생하게 된다.

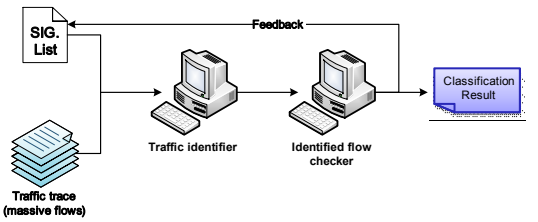


그림 1. 트래픽 분석 시스템
Fig. 1. Traffic analysis system

3.1 시그니쳐 간 중복과 충돌

본 절에서는 시그니쳐 매칭 과정에서 발생하는 시그니쳐 간 중복과 충돌에 대해 기술한다. 그림 2와 그림 3은 그림 1의 시그니쳐 리스트(Sig List)가 트래픽 플로우와 매칭되는 과정을 보여준다.

시그니쳐 중복 매칭은 그림 2와 같이 같은 응용(Torrent)의 시그니쳐가 하나의 플로우에 두 번 이상 매칭 되는 현상이다.

시그니쳐 충돌은 그림 3과 같이 다른 응용의 시그니쳐가 하나의 플로우에 동시에 매칭 되는 현상이다. 시그니쳐 충돌 매칭이 발생하면 응용 분석이 올바르게 되지 않기 때문에 시그니쳐 자체의 오류로 판단하고 시그니쳐 리스트를 최적화 재정렬 방법을 제안하는 본 논문에서는 고려하지 않기로 한다.

그림 1의 분석 시스템에서 풀 매칭 방식을 사용하는 이유는 풀 매칭 분석과정에서 발생하는 시그니쳐

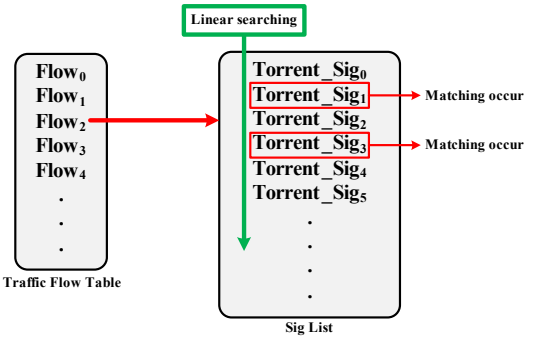


그림 2. 시그니쳐 중복
Fig. 2. Signature duplication

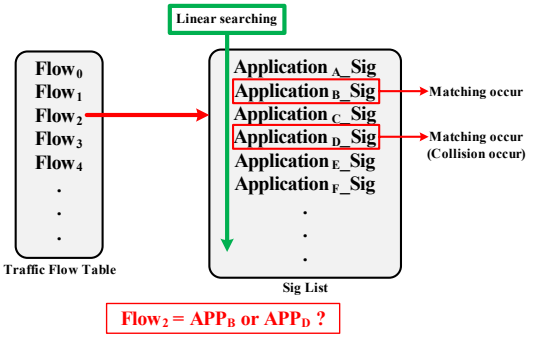


그림 3. 시그니쳐 충돌
Fig. 3. Signature collision

중복 현상을 통해 모든 시그니쳐의 분석 활용도를 파악 할 수 있기 때문이다. 중복 매칭 현상을 분석하면 부분 매칭 방식으로는 알 수 없었던 모든 시그니쳐의 응용 분석 효율을 알 수 있고 이를 바탕으로 시그니쳐 리스트 재정렬이 가능하게 된다.

3.2 시그니쳐 리스트 재정렬의 필요성

본 논문에서 제안하는 시그니쳐 리스트 최적화 재정렬의 목적은 비효율적인 매칭 방법을 개선해 불필요한 매칭 횟수를 줄이고 응용 분석 시간을 단축시켜 시그니쳐 리스트의 비효율성을 낮춤으로써 분석 속도를 높이는 것이다.

기존의 시그니쳐 리스트처럼 시그니쳐들이 무작위로 정렬된 리스트를 랜덤 리스트(Random list)로 정의하고 본 논문에서 제안하는 방법을 통해 재정렬 된 리스트를 재정렬 리스트(Reordered list)라고 정의한다. 이 때 랜덤 리스트의 문제점은 트래픽의 발생 특징과 리스트 내에 존재하는 각 시그니쳐의 분석 효율을 고려하지 않은 시그니쳐 정렬 방식이다. 그 결과 리스트 탐색 과정에서 불필요한 추가적 매칭이 발생하고 분석 속도가 감소하게 되는 원인이 된다. 그림 4를 통해

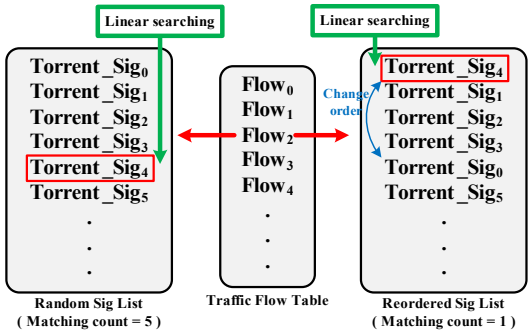


그림 4. 매칭 횟수 비교
Fig. 4. Matching count comparison

두 리스트 간의 응용 분석 효율을 비교 할 수 있다.

그림 4와 같이 랜덤 리스트는 플로우에 시그니쳐를 매칭하는 과정에서 재정렬 리스트에 비해 시그니쳐 리스트를 탐색, 매칭하는 과정(matching count)에 불필요한 시간이 추가적으로 소비 된다. 랜덤 리스트는 첫 번째 시그니쳐 매칭까지 매칭 횟수가 5번인 반면 재정렬 리스트는 1번의 매칭만으로 플로우 분석이 가능하다. 리스트 탐색에 시간이 추가적으로 소비되면 결국 분석 속도가 떨어지게 된다.

결과적으로 보다 빠른 응용 트래픽 분석 속도를 위해 중요한 것은 매칭 횟수를 낮춰 시그니쳐 매칭에 소비되는 시간을 줄이는 것이다.

Ⅳ. 시그니쳐 리스트 최적화 방법

본 장에서는 논문에서 제안하는 효율적 시그니쳐 리스트 재정렬 방법에 대해 기술한다.

재정렬 방법의 개요는 시그니쳐 리스트에 존재하는 각 시그니쳐들의 특징과 분석 효율을 평가해 수치화하고 수치화 된 값을 기준으로 높은 값을 갖는 시그니쳐부터 내림차순으로 정렬하는 과정이다.

4.1 Signature value

본 절에서는 시그니쳐 리스트 재정렬을 위한 각 시그니쳐의 가치 값(Signature value)을 계산하는 방법에 대해 기술한다.

시그니쳐 가치 값(Signature value)는 1에서부터 100까지의 값을 갖고, 값이 높을수록 더 가치가 높은 시그니쳐이다. Signature value(이하 SV)는 시그니쳐 풀 매칭 분석과정에서 발생하는 모든 시그니쳐 매칭 상황을 고려한 3가지 변수를 통해 계산 된다.

첫 번째 변수는 Frequency value이다. Frequency value(이하 FV)는 수식 1과 같이 전체 flow 중 특정

시그니쳐에 매칭된 flow의 비율을 나타내는 값으로, 리스트에 존재하는 시그니쳐들이 각각 flow를 얼마나 분석 할 수 있는지를 수치화 한 변수이다. FV는 1에서부터 100까지의 값을 갖고 높은 값을 갖을수록 더 가치가 높은 변수이다.

$$Frequency\ Value = \frac{Sig_x\ Matched\ Flows}{Total\ Flows} * 100 \quad (1)$$

두 번째 변수로 Offset value가 있다. Offset value (이하 OV)는 각 시그니쳐가 패킷의 페이로드에 매칭되는 offset을 나타낸다. 앞쪽에서 매칭되는 시그니쳐 일수록 더 높은 OV값을 갖는다. 패킷 페이로드의 앞부분에서 매칭 될수록 더 빠르게 매칭 된다는 것을 의미하고 빠르게 매칭 된다는 것은 결국 응용 분석 속도 측면에서 또한 빠르다는 것을 의미한다.

OV는 수식 2와 같이 계산 된다. 값은 FV와 마찬가지로 1에서부터 100까지의 값을 갖고, 높은 값을 갖을수록 더 빠르게 매칭 된다는 것을 의미한다.

예로 패킷의 50번째 오프셋에서 매칭되는 시그니쳐 X와 200번째 오프셋에서 매칭되는 시그니쳐 Y가 있을 때 시그니쳐 X는 96의 OV를 시그니쳐 Y는 86의 OV를 갖는다.

$$Offset\ Value = \frac{(\ Payload\_MAX\_Length\ ) - Sig_x offset}{Payload\_MAX\_Length} * 100 \quad (2)$$

세 번째 변수는 Type value로 4.2절에서 정의 할 시그니쳐의 유형과 연관되는 값을 갖는다. Type value(이하 TV)는 시그니쳐의 유형에 따라 고정된 값을 갖거나 유연하게 상황에 맞춰 변화하는 값을 갖는다. TV는 정확한 SV값을 위한 보정치로써 시그니쳐의 유형에 따라 각 시그니쳐별 우선순위가 존재할때 순위가 역전되는 현상을 방지해준다.

본 논문에서는 시그니쳐들을 크게 두 가지 유형으로 나누고 각 유형별로 수식 3과 같이 TVET 와 TVLT로 나누어 정의했다.

$$\begin{aligned} Essential\ Signature\ Type &: TV_{ET} = 0.5 \\ Low\ Value\ Signature\ Type &: TV_{LT} = Flexible \end{aligned} \quad (3)$$

시그니쳐의 가치 값(Signature value)은 최종적으로 수식 4와 같이 계산된다. 계산된 각 시그니쳐의 SV값을 바탕으로 시그니쳐 리스트를 재정렬한다.

$$Signauture\ Value = TV * (FV + OV) \quad (4)$$

4.2 시그니쳐 유형

본 절에서는 시그니쳐 유형을 각 리스트에 존재하는 시그니쳐들이 가지고 있는 고유한 특징을 바탕으로 그림 5과 같이 크게 2가지 세부적으로 4가지로 유형으로 정의한다. 시그니쳐 유형을 정의하는 첫 번째 이유는 각 시그니쳐의 직접적인 매칭 효율성과는 별개로 실제 분석 시스템에서 시그니쳐들이 갖는 고유한 특징이나 중요도를 SV값에 적용시키기 위해서이고 두 번째 이유는 이러한 각 시그니쳐의 특성에 따른 중요도를 TV값을 통해 조절하기 위해서이다.

Essential signature type은 분석률 유지를 위해 반드시 필요한 시그니쳐 유형들로 유형 1, 2, 3의 시그니쳐들은 높은 우선순위를 갖는다. 반면 Low Value signature type은 분석률 측면에서 제외시켜도 되는 시그니쳐 유형이다.

이 중 유형 3은 상대적으로 특수한 유형이다. 하나의 플로우에 시그니쳐 중복 매칭이 발생 할 경우 해당 시그니쳐들 간의 FV+OV 값을 비교 후 더 높은 값을 갖는 시그니쳐는 유형 3으로 낮은 값을 갖는 시그니쳐는 유형 4로 정의한다.

예시로 그림 6에서 6개의 flow에 매칭 된 6개의 시그니쳐들(0, 9, 11, 16, 17, 19번)에 대해 유형을 정의한다.

그림 6에서 주목 할 점은 시그니쳐 0번과 11번, 19번만 있다면 나머지 9, 16, 17번 없이도 7개의 flow를 모두 분석 할 수 있다는 것이다. 따라서 그림 5의 유형 정의에 의해 시그니쳐 0번과 19번은 유형 1, 11번은 유형 3으로 정의되고 모두 Essential signature type

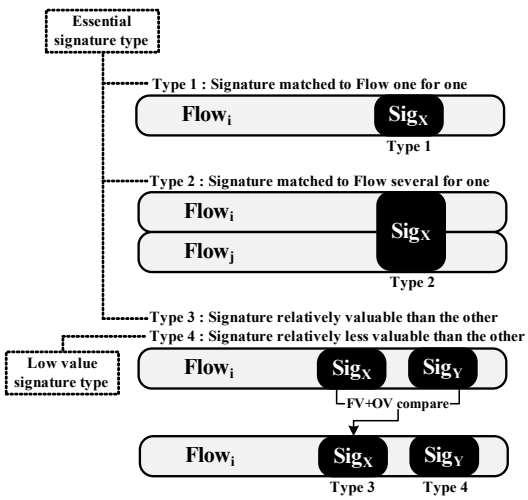


그림 5. 시그니쳐 유형
Fig. 5. Signature type

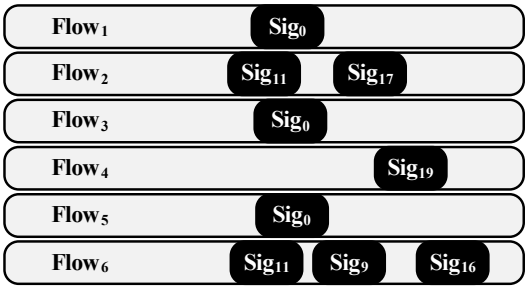


그림 6. 유형 결정 예
Fig. 6. Type decision example

표 1. 시그니쳐 유형 결정 알고리즘
Table. 1. Signature type decision algorithm

```
f : Classified flow
S : Matched signature
T : Signature type
S(f) = { S | S0, S1, S2 .... Sn }
T(S) ∈ { T1, T2, T3, T4 }

1   Give type to each flow matched signature
2:  Input : Flow info Container
3:  for each fi do // Type 1 decision
4:     if ( |S(fi)|== 1 )
5:        T(S0) = T1;
6:  done
7:  for each fi  do // Type 2 , 4 decicison
8:     if( |S(fi)| > 1 )
9:     {
10:       for each S(fi) do
11:          if ( T(Sn) == T1 )
12:             T(Sn) = T2;
13:          else if ( T(Sn) == None )
14:             T(Sn) = T4;
15:       done
16:    }
17: done
18: for each fi do // Type 3 decision
19:    if ( T(Sn) is all T4 )
20:       find Sk which has maximum FV+OV value among S(fi)
21:       T(Sk) = T3;
22: done
```

이다. 나머지 9, 16, 17번은 모두 유형 4이고 Low value type으로 정의된다.

시그니쳐 유형 결정 방법은 표 1의 알고리즘의 의사코드를 바탕으로 구현되었다.

4.3. Type value

각 시그니쳐의 유형이 정의되면 유형 별로 TV값을 부여한다. 수식 4를 참고하였을 때 FV와 OV값이 합의 최대치가 200이기 때문에 SV의 최대치인 100으로 조정하기 위해서 Essential signature type들(유형 1,2,3)의 시그니쳐들의 TV값(TVET)은 수식 3과 같이 고정적으로 0.5를 부여한다. 반면 Low value signature type(유형 4)의 시그니쳐들에 대해서는 수식 5와 같이 계산한다.

Essential signature type 시그니쳐들이 갖는 SV값들 중 최소값을 SVET_min이라 정의하고 Low value signature type 시그니쳐가 갖는 SV값들 중 최대값을 을 SVLT_max라 정의했을 때, SVET_min 값은 SVLT_max 값보다 항상 더 커야한다. 따라서 TVLT 값은 수식 5와 같이 계산 될 수 있다.

$$TV_{LT} = (SV_{ET\_min} / (FV_{LT} + OV_{LT})) - 0.01$$
$$\because SV_{LT\_max} < SV_{ET\_min}$$
$$TV_{LT} < SV_{ET\_min} / (FV_{LT} + OV_{LT}) \quad (5)$$

모든 시그니쳐의 TV값이 설정되면 SV값을 계산하기 위한 모든 변수들이 정의 되었다. 각 시그니쳐들의 SV값을 계산해 높은 값부터 내림차순으로 시그니쳐 리스트를 재정렬 한다.

V. 재정렬 시그니쳐 리스트 성능 평가

본 장에서는 4장에서 기술한 재정렬 시그니쳐 리스트를 실제 트래픽 분석에 적용하여 기존의 시그니쳐 리스트와의 응용 트래픽 분석 속도를 비교하고 실제로 더 효율적인지 평가한다.

그림 7은 그림 1의 분석 시스템을 수정해 재정렬 리스트와 다른 알고리즘을 적용한 리스트 간의 응용 분석 속도를 비교 가능하도록 했다.

실험은 다량의 트래픽 분석을 통해 제작된 경험적인 재정렬 시그니쳐 리스트를 새로운 트래픽 분석에 적용 시켰을 때 다른 알고리즘의 리스트에 비해 얼마나 더 효율적인지 평가 한다.

사용되는 트래픽은 재정렬 리스트 제작을 위해 사용되는 Trainig set 트래픽과 각 리스트들의 성능 평가를 위해 사용되는 Testing set 트래픽이 있다. 시그니쳐 리스트로는 3.2절에서 정의한 Random list와 본 논문에서 제안하는 Reordered list 그리고 Ideal list가 있다. Ideal list는 Testing set 트래픽에 대한 응용 분석률이 가장 높은 시그니쳐부터 정렬 시킨 리스트이다. Ideal list는 Testing set 트래픽을 분석하기에 가장 이상적인 리스트로 재정렬 리스트와의 상대적 성능

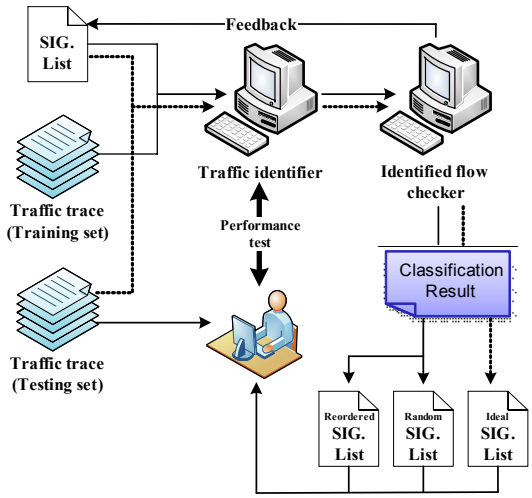


그림 7. 성능 테스트를 위한 트래픽 분석 시스템
Fig. 7. Traffic analysis system for performance test

비교를 위해 제작되었다.

그림 7과 같이 실험의 첫 단계는 Training set 트래픽에서 4장에서 정의했던 방법들을 통해 Reordered list를 제작한다. 최종적으로 완성된 재정렬 리스트와 Random list, Ideal list를 각각 Testing set 트래픽에 매칭 시켜 본 후 성능을 평가한다. 성능 평가를 할 때는 일반적인 응용 분석 시스템과 같이 부분 매칭 방식을 사용한다.

각 리스트들의 성능 평가는 분석시간 과 총 시그니쳐 매칭 횟수 두 가지 척도를 기준으로 판단한다. 첫 번째 척도인 분석 시간은 분석기에서 시그니쳐를 매칭하는 과정에서 CPU가 실제로 작업을 수행하는데 걸린 User-time을 측정한다. 두 번째 척도인 매칭 횟수는 시그니쳐가 플로우에 매칭된 실질적인 매칭 횟수를 측정한다. 매칭 횟수가 성능 평가 척도로서 의미

표 2. 트레이닝 세트 정보
Table. 2. Training set information

Traffic Trace ID	Content Size (MB)	Duration (min.)	Flow	Pkt	Byte
008_UT_FP06	20	1	278	4.8E+04	3.5E+07
008_UT_FP07	961	21	2,505	1.1E+06	1.2E+09
008_BT_FP08	899	5	1,760	5.3E+05	4.5E+08
008_BT_FP09	1,490	3	2,262	6.2E+05	6.8E+08
008_UT_FP10	1,520	1	1,363	1.0E+06	9.6E+08
008_UT_FP11	25	1	330	3.3E+04	2.8E+07
008_UT_FP13	954	5	3,204	1.0E+06	8.5E+08
012_FD_FP01	890	13	758	9.0E+05	9.0E+08
012_QB_FP01	1,020	18	2,632	1.0E+06	9.0E+08
Total	7,780	58	15,092	6.3E+06	6.0E+09

표 3. 테스트 세트 정보
Table. 3. Testing set information

Traffic Trace ID	Content Size (MB)	Duration (min.)	Flow	Pkt	Byte
006_UT_FP02	119	24	10,874	2.1E+06	1.8E+09
006_UT_FP03	42.1	6	3,988	1.0E+06	8.4E+08
006_UT_FP04	355	7	2,996	1.5E+06	1.3E+09
006_UT_FP05	46.1	9	2,961	1.1E+06	9.7E+07
006_BT_FP02	119	18	4,946	2.0E+06	1.8E+09
006_BT_FP03	42.1	8	3,329	9.4E+05	8.4E+08
006_BT_FP04	355	10	3,603	1.5E+06	1.3E+09
006_BT_FP05	46.1	5	2,619	1.3E+05	1.2E+09
010_VZ_FP01	119	6	806	1.2E+06	9.2E+08
013_UT_FP01_1	788	9	3,198	8.2E+05	7.8E+08
013_UT_FP01_2	788	7	2,599	9.2E+05	8.4E+08
013_UT_FP01_3	788	9	3,124	7.9E+05	7.4E+08
013_UT_FP01_4	788	3	126	8.8E+05	8.4E+08
013_UT_FP01_5	788	3	131	8.5E+05	8.7E+08
013_UT_FP01_6	788	3	127	8.5E+05	8.7E+08
013_UT_FP01_7	788	3	131	7.3E+05	7.4E+08
013_UT_FP01_8	788	4	109	6.8E+05	7.0E+08
013_UT_FP01_9	788	4	102	6.7E+05	6.8E+08
013_UT_FP01_10	788	5	106	7.1E+05	7.2E+08
Total	9120	143	45,875	1.9E+07	1.8E+10

를 갖는 이유는 분석 시간으로 구분 할 수 없는 보다 직접적인 비교 분석을 가능하게 해주기 때문이다.

5.1 실험 환경

본 절에서는 재정렬 시그니쳐 리스트의 성능 평가 실험을 위한 실험 환경에 대해 기술한다.

실험을 위해 수집한 트래픽은 모두 토렌트(Torrent) 트래픽으로 WireShark와 Net Monitor를 통해 트래픽을 수집한 뒤 다른 응용의 트래픽은 모두 제외 시키고 토렌트 응용의 트래픽만을 저장했다. 모아진 트래픽 파일은 총 28개 파일로 이 중 9개의 파일은 재정렬 시그니쳐 리스트 제작을 위한 Training set으로 사용 되었고 나머지 19개의 파일은 각 리스트의 성능 평가를 위한 Testing set으로 사용 되었다. 각 Set의 정보는 표 2, 표 3과 같다.

트래픽 파일을 분석하기 위한 페이로드 시그니쳐의 개수는 총 32개로 표 4와 같다. 32개의 시그니쳐 모두 토렌트 응용 트래픽을 분석하기 위해 작성 되었다.

5.2 실험 결과

5.1 절에서 나타낸 실험 환경에서 그림 7의 분석 시스템을 적용시켜 실험을 진행 한 결과 재정렬 시그니쳐 리스트는 표 5와 같이 제작되었다.

표 4. 토렌트 페로드 시그니처
Table. 4. Torrent payload signatures

Sig ID	Payload signature	Protocol	Port
0	.*BitTorrent protocol.*	TCP	N/A
1	.*BitTorrent protocol.*	UDP	N/A
2	.*d1:ad2:id20.*	TCP	N/A
3	^d1:ad2:id20.*	UDP	N/A
4	.*d1:rd2:id20.*	TCP	N/A
5	^d1:rd2:id20.*	UDP	N/A
6	.*find_node.*UT.*	UDP	N/A
7	.*info_hash.*get_peer.*	UDP	N/A
8	.*5:peers.*	TCP	N/A
9	.*User-Agent:.*uTorrent.*	TCP	N/A
10	.*Host: com-utorrent.*	TCP	N/A
11	.*User-Agent: BTWebClient.*	TCP	N/A
12	.* //announce.info_hash=.*peer_id=.*	TCP	N/A
13	.*//scrape.info_hash=.*	TCP	N/A
14	.*d5:added.*	TCP	N/A
15	.*d5:added.*	UDP	N/A
16	.*Host:.*utorrent\.com*."	TCP	N/A
17	.*Host:.*bittorrent\.com*."	TCP	N/A
18	.*bittorrent.com.	UDP	N/A
19	.*tracker.*	UDP	N/A
20	*BT.*announce	UDP	N/A
21	.*UT.*announce.*	UDP	N/A
22	.*d2:ip6:.*1:rd2:id20.*	UDP	N/A
23	^d2:ip6:.*1:rd2:id20.*	UDP	N/A
24	^.....BT.....*	UDP	N/A
25	^.....UT.....*	UDP	N/A
26	.*\x00\x00\x04\x17\x27\x10\x19\x80\x00\x00\x00.*	UDP	N/A
27	^BT-SEARCH.*	UDP	N/A
28	.*BitTorrent.*	UDP	1900
29	.*uTorrent.*	UDP	1900
30	.*Host:.*deluge-torrent\.org.*	TCP	N/A
31	.*User-Agent: Deluge.*	TCP	N/A

표 5. 재정렬된 시그니처 리스트
Table. 5. Reordered signature list

Sig ID	FV	OV	Type	TV	SV	Rank
3	83.63	99.93	2	0.5	91.78	1
23	32.14	99.93	2	0.5	66.04	2
5	10.04	99.93	2	0.5	54.99	3
0	8.17	99.93	2	0.5	54.05	4
1	3.31	98.87	2	0.5	51.09	5
12	0.52	99.67	2	0.5	50.09	6
27	0.02	99.93	1	0.5	49.98	7
19	0.11	99.67	2	0.5	49.89	8
13	0.02	99.67	3	0.5	49.84	9
15	0.01	98.73	1	0.5	49.37	10
⋮						

표 5에서 나타내는 FV, OV, TV, SV 값은 4장에서 정의했던 변수들이고 SV값을 기준으로 높은 순으로 내림차순 되었다.

실험 결과는 Testing set에 포함된 트레이스들 중 두 개의 단일 트래픽 트레이스 기준으로 각각 그림 8, 그림 9과 같고 모든 Testing set 트래픽 트레이스 기준으로 했을 때는 표 6, 그림 10와 같다.

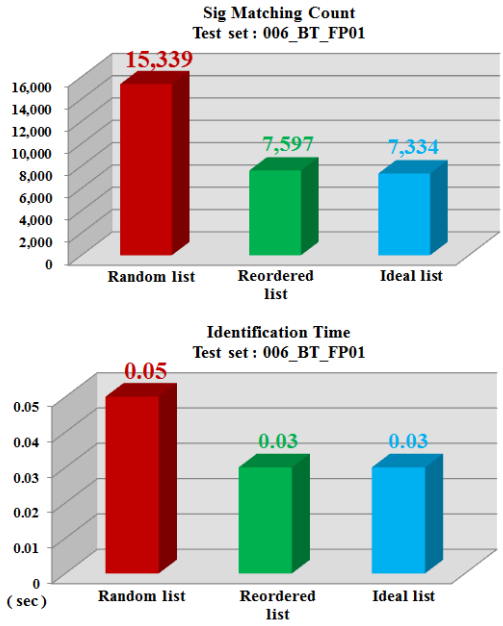


그림 8. 테스트 세트(006_UT_FP01) 결과
Fig. 8. Testing set (006_UT_FP01) result

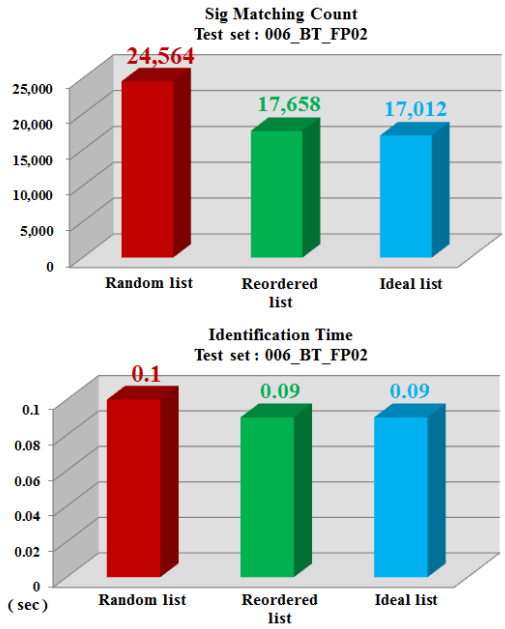


그림 9. 테스트 세트(006_UT_FP02) 결과
Fig. 9. Testing set (006_UT_FP02) result

단일 트래픽 트레이스(ID 006_BT_FP01)를 기준으로 했을 때 Reordered list는 Random list보다 시그니처 매칭 횟수 측면에서 약 50% 분석 속도 측면에서

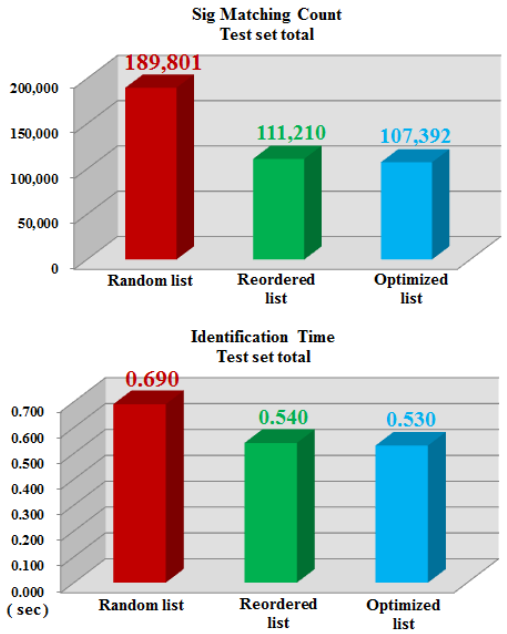


그림 10. 테스트 세트(Total) 결과
Fig. 10. Testing set (Total) result

표 6. 테스트 세트(Total) 결과-2
Table. 6. Testing set (Total) result-2

Category	Random List	Reordered List	Ideal List
Classification rate	98%	98%	98%
Matching time(sec)	0.69	0.54	0.53
Matching count	189,801	111,210	107,392
Rank	3	2	1

약 40% 정도의 분석 속도 향상을 보였다. 또한 Ideal list와 비교했을 때는 상대적으로 약간 저조하지만 매우 미미한 성능 차이를 보인다.

그림 9(ID 006_BT_FP02)에서도 마찬가지로 기존의 Random list에 비해 매칭 횟수에서는 30%, 분석 시간에서는 10%의 분석 속도 향상을 보였고, Ideal list와 비교 했을 때는 매칭 횟수 측면에서 약간의 성능 차이가 있었다.

최종적으로 모든 Testing set 트래픽 트레이스를 기준으로 했을 때 결과는 표 6, 그림 10과 같다. 마찬가지로 3개의 리스트와 2개의 판단 척도를 기준으로 평가 했을 때, 전체적으로 Reordered list는 기존의 Random list보다 분석 시간 측면에서는 약 20%의 분석 시간 단축률을 보였고 Ideal list와는 거의 동일한 분석 시간을 보였다. 시그니쳐 매칭 회수 측면에서는 Random list 보다 약 40%의 추가 분석 효율을 보였고 Ideal list 보다는 약 3%정도 분석 효율이 떨어지는 것으로 나타났다.

결과적으로 Random list보다 Reordered list의 분석 성능이 높은 것은 당연한 결과라고 판단 할 수 있다. 또한 Testing set 트래픽을 분석하기에 가장 이상적인 Ideal list가 실험에서는 가장 효율적인 것으로 나왔지만 표 6를 참조하였을 때 Reordered list와 크게 차이가 없는 것으로 나타났다. 결국 본 논문에서 정의하고 제안하는 시그니쳐 최적화 재정렬 방법이 매우 효과적이라는 것이 실험을 통해 증명되었다.

Ⅵ. 결론 및 향후 과제

본 논문에서는 페이로드 시그니쳐 기반 응용 트래픽 분석 속도 향상을 위해 페이로드 시그니쳐 별 분석 효율을 고려하여 리스트에 존재하는 시그니쳐 순서를 최적화해 재정렬 하는 방법을 제안하였다. 제안하는 시그니쳐 리스트 재정렬 방법은 트래픽 응용 분석률은 유지하면서 기존의 시그니쳐 리스트(Random list)에 비해 전체적으로 약 30%의 분석 성능 향상을 보였고 Ideal list 와도 비슷한 수준의 분석 성능을 보였다.

본 논문에서는 오프라인 환경에서 이미 수집이 완료된 트래픽에 대해서만 시그니쳐 리스트 재정렬 방법을 적용시켰다. 향후 연구로 오프라인 환경 뿐만 아니라 실시간 응용 트래픽 분석 환경에서도 시그니쳐 리스트 재정렬 방법을 적용 시킬 수 있도록 연구를 수행할 계획이다.

References

[1] J. S. Park, J. W. Park, S. H. Yoon, Y. S. Oh, and M. S. Kim, “Development of signature generation system and verification network for application level traffic classification,” in *Proc. KIPS Conf.*, pp. 1288-1291, Pusan, Korea, Apr. 2009.

[2] S. H. Yoon, H. G. Roh, and M. S. Kim, “Internet application traffic classification using traffic measurement agent,” in *Proc. KICS Summer Conf.*, pp. 1747-1750, Jeju Island, Korea, Jul. 2008.

[3] S.-H. Yoon and M.-S. Kim, “Research on signature maintenance method for internet application traffic identification using header

signatures," *J. KICS,* vol. 36 no. 6, pp. 600-607, Jun. 2011.

[4] R. Antonello, S. Fernandes, D. Sadok, and J. Kelner, "Characterizing signature sets for testing DPI systems," in *Proc. IEEE GLOBECOM Management of Emerging Networks and Services Workshop*, pp. 678-683, Houston, TX, USA, Dec. 2011.

[5] Y. Jin, N. Duffield, J. Erman, P. Haffner, S. Sen, and Z.-L. Zhang, "A modular machine learning system for flow-level traffic classification in large networks," *ACM Trans. Knowledge Discovery from Data,* vol. 6, no. 1, pp. 1-34, Mar. 2012.

[6] S.-H. Yoon and M.-S. Kim, "Behavior signature for big data traffic identification," in *Proc. Int. Conf. Big Data and Smart Comput. (BigComp),* pp. 261-266, Bangkok, Thailand, Jan. 2014.

[7] F. Yu, Z. Chen, Y. Dino, T. V. Lakshman, and R. H. Katz, "Fast and memory efficient regular expression matching for deep packet inspection," in *Proc. ACM/IEEE Symp. Architecture Netw. Commun. Syst. (ANCS '06)*, pp. 93-102, San Jose, USA, Dec. 2006.

[8] C. L. Hayes and Y. Luo, "DPICO: A high speed deep packet inspection engine using compact finite automata," in *Proc. ACM/IEEE Symp. Architecture Netw. Commun. Syst. (ANCS '07)*, pp. 195-203, Orlando, USA, Dec. 2007.

[9] G. Vasiliadis, M. Polychronakis, S. Antonatos, E. P. Markatos, and S. Ioannidis, "Regular expression matching on graphics hardware for intrusion detection," in *Proc. 12th Int. Symp. Recent Advances Intrusion Detection (RAID '09)*, pp. 265-283, Saint-Malo, France, Sept. 2009.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd Ed., MIT Press and McGraw-Hill, 2001.

[11] J.-H. Choi and M.-S. Kim, "Processing speed improvement of traffic classification based on payload signature hierarchy," in *Proc. Asia-Pacific Network Operations and Management Symp.(APNOMS)*, Hiroshima, Japan, Sept. 2013.

[12] S.-H. Yoon and M.-S. Kim, "Performance improvement of a real-time traffic identification system on a multi-core CPU environment," *J. KICS,* vol. 37B, no. 5, pp. 348-356, May 2012.

[13] A. Mitra, W. Najjar, and L. Bhuyan, "Compiling PCRE to FPGA for accelerating SNORT IDS," in *Proc. 3rd ACM/IEEE Symp. Architecture Netw. Commun. Syst. (ANCS '07)*, pp. 127-136, Orlando, USA, Dec. 2007.

[14] J.-S. Park, S.-H. Yoon, and M.-S. Kim, "Performance improvement of the payload signature based traffic classification system using application traffic locality," *J. KICS,* vol. 38B, no. 7, pp. 519-525, Jul. 2013.

[15] J. S. Park, S. H. Yoon, M. S. Kim, "Software architecture for a lightweight payload signature-based traffic classification system," in *Proc. Traffic Monitoring and Anal. Workshop,* pp. 136-149, Vienna, Austria, Apr. 2011.

[16] J. S. Park and M. S. Kim, "Performance improvement of application-level traffic classification system using application traffic pattern," in *Proc. KICS Summer Conf.*, pp. 3-7, Jeju, Korea, Jun. 2011.

이 성 호 (Sung-ho Lee)

2010년~현재 : 고려대학교 컴퓨터 정보학과

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 트래픽 분류

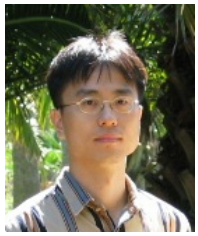
박 준 상 (Jun-sang Park)

2008년 : 고려대학교 컴퓨터 정보학과 졸업
2010년 : 고려대학교 컴퓨터 정보학과 석사
2014년 : 고려대학교 컴퓨터 정보학과 박사
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 트래픽 분류

김 명 섭 (Myung-sup Kim)

1998년 : 포항공과대학교 전자계산학과 졸업
2000년 : 포항공과대학교 컴퓨터 공학과 석사
2004년 : 포항공과대학교 컴퓨터 공학과 박사
2006년 : Post-Doc. Dept. of ECE, Univ. of Toronto, Canada
2006년~현재 : 고려대학교 컴퓨터정보학과 부교수
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 멀티미디어 네트워크

석 우 진 (Woojin Seok)



1996년 경북대학교 컴퓨터 공학과 졸업
2003년 Univ. of North Carolina, Computer Science 석사
2008년 충남대학교 컴퓨터 공학과 박사
2010년~2011년 : 교환연구원, School of Computing, Univ. of Utah, USA
2003년~현재 : 한국과학기술정보연구원 책임연구원
2012년~현재 : 연합대학원 겸임교수
<관심분야> 미래 인터넷, 네트워크 관리, SDN, 클라우드 컴퓨팅, TCP 성능분석, 네트워크 QoS