페이로드 시그니쳐 기반 인터넷 트래픽 분류

박준상, 윤성호, 안현민, 김명섭

고려대학교 컴퓨터정보학과

{junsang_park, sungho_yoon, queen26, tmskim}korea.ac.kr

요 약

응용 레벨 트래픽 분류는 안정적인 네트워크 운영과 자원 관리를 위해서 필수적으로 요구된다. 트래픽 분류 방법에 있어서 페이로드 시그니쳐 기반 트래픽 분류는 패킷 헤더 기반, 통계 기반 분석 등의 다양한 분류 방법의 평가 기준으로 활용될 만큼 높은 분류 정확도와 분석률을 보장하고 있는 방법이다. 하지만 네트워크 고속화, 응용 프로그램의 다양성으로 인해 분류 정확도, 처리 속도가 감소되고 있다. 이러한 문제점을 해결하기 위해서 페이로드 시그니쳐 모델, 입력 데이터 최적화 모듈, 패턴 매칭 모듈 등의 분류 시스템의 세부 구성 모듈에 대한 부분적인 성능 향상을 위한 다양한 연구가 수행되고 있다. 하지만 분류 시스템의 성능 향상을 위해서는 응용의 분류 기준 정의와 각 응용 프로토콜의 특징을 고려한 시그니쳐 모델 정의가 선행된 후에 분류 기준 및 시그니쳐 모델에 최적화된 세부 모듈 구성 요구된다. 본 논문에서는 응용의 분류 기준, 페이로드 시그니쳐 모델, 입력 데이터 필터, 패턴 매칭 측면에서 최근의 연구 동향을 조사하고, 페이로드 시그니쳐 기반 분류 시스템의 분류 정확도와 처리 속도 향상을 위한 최적의 분류 시스템의 구조를 제안한다.

1. 서론

네트워크의 고속화와 더불어 다양한 서비스와 응용프로그램이 개발됨에 따라 개인 또는 기업은 인터넷으로 대표되는 네트워크에 대한 의존이 상당히 커져가고 있다. 이와 같은 현실 속에서 네트워크의 효율적 운용과 관리를 위한 응용 레벨의 트래픽의 모니터링과 분석은 네트워크 사용현황 파악과 확장계획 수립 등의 다양한 분야에서 필요성이 증가하였다. 예를 들어 종량제 과금, CRM, SLA, 보안 분석 등 트래픽 모니터링 및 분석에 대한 필요성은 지금뿐만 아니라 앞으로 더욱더 크게 증가할 것이다.

응용 레벨 트래픽 분류 방법에 있어 페이로드 시그니쳐 기반 분석 방법은 헤더 정보 기반, 통계 기반 분류 방법론에 비해 상대적으로 높은 분류 정확성과 분석률을 보장할 있는 방법이다[1-4]. 하지만 네트워크 링크의 고속화, 응용 프로그램을 구성하는 프로토콜의 복잡성, 응용 프로그램 다양성으로 인해 처리 속도 및 분류 정확도가 감소되고 있다. 이러한 문제를 해결하기 위해서 분류 시스템을 구성하는 세부 모듈의 부분적인 성능 개선안들이 제안되고 있다[5-7]. 하지만 분류 시스템의 세부 모듈은 트래픽의 분류 기준과 분류 대상 트래픽의 특징에 따라 다른 구성이 요구된다. 예를 들어, HTTP 트래픽을 응용 레벨 프로토콜 기준으로 분석하기 위해서는 정규 표현식보다는 HTTP 프로토콜의 요청 method(e.g. “GET”, “POST”, “HEAD”)를 단순한 스트링 형태의 시그니쳐로 구성하여 매칭하는 방법이 처리 속도와 메모리 사용 측면에서 효과적이다. 반면에 HTTP 트래픽을 세부 서비스 단위로 분석하기 위해서는 시그니쳐가 나타나는 HTTP 프로토콜의 주요 필드(e.g. URI, Host, User-agent)에서 시그니쳐를 추출하고, 필드 단위 매칭을 수행하면 분류 정확도와 처리 속도를 향상 시킬 수 있다[8]. 하지만 기존의 연구에서는 트래픽의 분류 기준, 분석 대상 응용의 프로토콜 특징을 고려하지 않고, 분석 대상 링크에서 발생하는 모든 트래픽을 동일한 분석 구조를 통해 분석하기 때문에 분류 속도, 분류 정확도가 감소되고 있다.

본 논문에서는 최근 페이로드 시그니쳐 기반 분석 방법의 성능 향상에 관한 다양한 연구를 분류 기준, 시그니쳐 모델, 입력 데이터 필터, 매칭 알고리즘 관점에서 조사하고, 분류 시스템 구현을 위한 요구 사항을 도출한다. 도출된 요구 사항을 기반으로 분류 정확도, 처리 속도에 최적화된 분석 시스템의 구조를 제안한다.

본 장의 서론에 이어, 2 장에서는 페이로드 시그니쳐 기반 분석 방법의 범위와 성능 향상을 위한

이 논문은 2012 년 정부(교육과학기술부)의 재원으로 한국연구재단(2012R1A1A2007483) 및 2013 년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보.컴퓨팅기술개발사업(2010-0020728)의 지원을 받아 수행된 연구임.

기존의 연구를 조사하고 요구사항을 도출한다. 3 장에서는 요구사항을 기반으로 분류 시스템의 구조를 제안한다. 마지막으로 4 장에서는 결론 및 향후 연구에 대해 기술한다.

2. 페이로드 시그니쳐 기반 분석 방법

시그니쳐 기반 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. 시그니쳐의 추출 위치와 가공 방법에 따라 Header[9], Payload[3], Statistic[10], Behavior 시그니쳐[11]로 구분할 수 있다. 본 논문에서는 페이로드 시그니쳐 기반 응용 레벨 트래픽 분류 방법으로 조사 범위를 제안한다.

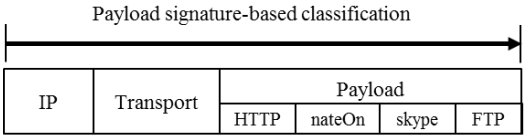

Figure 1. Inspection range

그림 1 은 페이로드 시그니쳐 기반 트래픽 분류 방법의 탐색 범위를 정의하고 있다. 본 논문에서는 Transport 이후의 데이터를 페이로드로 구분한다. 페이로드 시그니쳐 기반 분석 방법은 IP, Transport, 페이로드의 조합으로 응용 트래픽 분류하는 것으로 정의한다. IP 와 Port 를 이용한 트래픽 분류 방법이 CDN, 다이나믹 포트 사용으로 인해 분류 정확도가 감소하였지만 [12]에 의하면 30%-70%이상의 트래픽을 정확하게 분석할 수 있는 방법으로 조사되었다. 또한 IP, Port, 페이로드 시그니쳐의 조합으로 시그니쳐를 구성함으로써 분류 정확도를 향상 시킬 수 있으며, 페이로드 시그니쳐의 탐색 공간을 줄여서 분류 속도를 향상 시킬 수 있다.

2.1 페이로드 시그니쳐 기반 분석 모듈

페이로드 시그니쳐 기반 분류 시스템은 트래픽과 페이로드 시그니쳐를 입력으로 받아서 입력데이터 필터 모듈을 통해 시그니쳐 및 패킷 데이터의 탐색 공간을 최소화하는 과정을 거친 후 패턴 매칭을 통해 분류 결과를 도출한다. 그림 2 는 페이로드 시그니쳐 기반 분석 시스템의 입/출력 데이터와 주요 모듈을 나타내고 있다.

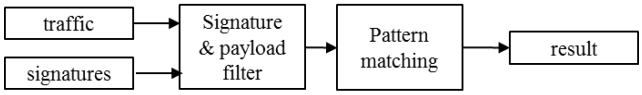

Figure 2. Diagram of payload signature-based classifier

트래픽은 패킷 또는 패킷의 집합인 플로우로 구성되며, 분류 결과 또한 이와 같은 단위로 도출된다. 시그니쳐는 패킷 또는 스트림 단위로 추출되며 패킷 단위에서 추출된 시그니쳐는 패킷 단위로 매칭을 수행하며, 스트림 단위로 추출된 시그니쳐는 패킷의 페이로드를 재조합하여 스트림으로 구성한 후 트래픽을 분석하게 된다. 시그니쳐 및 페이로드 필터는 패턴 매칭의 부하를 최소화하기 위해서 패턴 매칭 과정에서 불필요한 데이터를 제거하는 역할을 수행한다. 패턴 매칭 모듈은 패턴 매칭 알고리즘을 적용하여 페이로드와 시그니쳐의 매칭 여부를 결정한다. 패턴 매칭 결과는 시그니쳐에 정의된 분류 기준에 따라 분석 결과로 저장된다.

2.2 분류 기준

기존의 페이로드 시그니쳐 기반 트래픽 분류 방법은 QoS, SLA 등과 같은 분류 결과의 활용 방안을 고려하지 않고, 시그니쳐 구성에 의존적으로 분류 기준을 정의하고 트래픽을 분류하고 있다. 표 1 은 기존의 페이로드 시그니쳐 기반 트래픽 분류 방법에서 사용하고 있는 분류 기준을 보여 주고 있다.

Table I. Classification criteria

Type	Classes	Ref.
Application	afreeca, rainbow, alsong, nateon, gomplayer, donkeyp2p	[1]
Protocol	Web(HTTP, HTTPS), FTP, BT, SSH, SMTP	[13]
Function	Bittorrent(Downloading, DHT management, Tracker access)	[14]
Service	google, naver, daum, yahoo…	[8]
Traffic type	Interactive, Game, Bulk, Multimedia, Mail, Web, P2P, Conferencing…	[4]
Mixture	HTTP, BitTorrent, SMTP, MSN, Kugoo, SSL, Gnutella	[15]

표 1 과 같이 기존의 페이로드 시그니쳐 기반 분석 방법은 통일된 기준 없이 시그니쳐 구성에 따른 분류 결과를 제시하고 있다. 이와 같은 분류 결과는 QoS 나 방화벽에 모호한 정책으로 반영됨에 따라 효과적인 네트워크 자원 관리가 불가하다. 따라서 네트워크 관리 정책을 반영할 수 있는 구체적인 분류 기준을 확립하고 이를 기반으로 트래픽을 분류 할 수 있는 방안이 요구된다.

2.3 시그니쳐 모델

페이로드 시그니쳐는 기술 방법에 따라 Field-based simple string, simple string, regular expression 으로 구분할 수 있다. 표 2 는 BitTorrent 의 페이로드 시그니쳐를 시그니쳐의 기술 방법에 따라 구분한 결과를 나타내고 있다.

Table II. Signature example for each type

Model	Example	Ref.
Field-based simple string	User-Agent : BTWebClient, Domain : utorrent, Host : update	[8]
Simple string	\x13BitTorrent protocol	[16]
regular expression	^\x13bittorrent protocol.*	[21]

기존의 페이로드 시그니쳐 기반 분석 방법은 동일한 시그니쳐에 대해 서로 다른 형태의 시그니쳐 기술 방법을 사용하고 있다. 이는 프로토콜의 특징과 분석 시스템의 성능을 고려하지 않고, 시그니쳐 모델을 정의하였기 때문이다. 프로토콜의 특징을 고려한 시그니쳐 기술 방법이 요구된다.

HTTP 와 같은 프로토콜은 HTTP 의 특정 필드에서 시그니쳐가 추출된다. 이와 같은 트래픽은 필드 단위로 시그니쳐를 추출하고 매칭하는 것이 분류 정확도를 향상 시킬 수 있을 뿐만 아니라 필드 단위 매칭을 통해 분류 시스템의 처리 속도를 향상 시킬 수 있다.

정확도에 영향을 미치지 않는다면 regular expression 보다는 simple string 형태의 기술 방법이 분류 시스템의 부하를 줄일 수 있다. 동일한 페이로드 시그니쳐를 regular expression 또는 simple string 형태로 기술할 수 있다. 시그니쳐의 기술 방법은 패턴 매칭 알고리즘을 결정하기 때문에 분류 시스템의 처리 속도에 영향을 미치게 된다. Regular expression 은 simple string 형태의 기술 방법보다 폭넓은 표현력을 갖는다. 하지만 모든 시그니쳐를 regular expression 형태로 기술하는 것은 automata 에 기반한 패턴 매칭을 수행하기 때문에 계산 복잡도와 메모리 사용량 측면에서 simple string 보다 많은 부하를 요구하게 된다.

2.4 입력 데이터 필터 및 패턴 매칭 알고리즘

대용량의 트래픽과 많은 양의 시그니쳐는 분류 시스템의 처리 속도와 메모리 사용량을 증가 시키기 때문에 입력 데이터를 최소화할 수 있는 방법과 시그니쳐 형태에 적합한 분류 알고리즘이 요구된다.

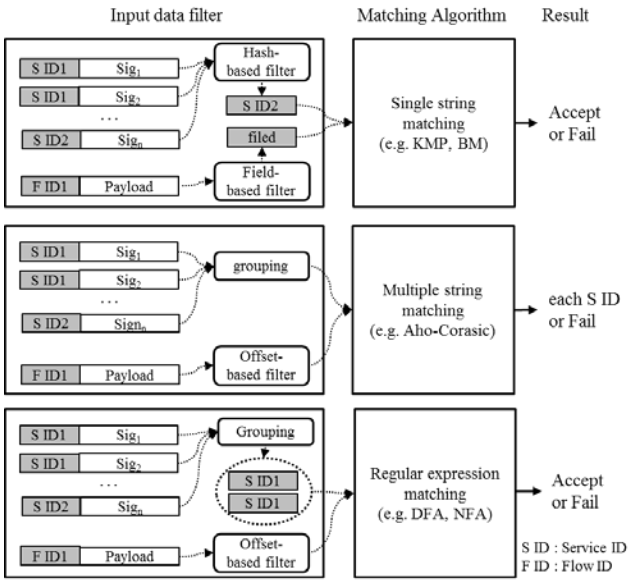

Figure 3. input data filter and pattern matching algorithm

시그니쳐의 탐색 공간을 최소화할 수 있는 방법은 Hash-based filter, Offset-based filter 으로 구분할 수 있다. 또한 트래픽의 탐색 공간을 최소화할 수 있는 방법으로 Field-based filter 가 있다.

[8]에서는 HTTP 트래픽을 응용프로그램, 서비스, 기능으로 분류하기 위해서 HTTP 프로토콜의 필드를 Domain, Host, URI 로 구분하여 각 필드의 시그니쳐를 자동으로 추출하여 Hash 기반으로 분석하는 방법론을 제안하였다. Hash 를 이용하여 시그니쳐를 필터링하여 시그니쳐의 탐색 공간을 줄이고, 페이로드를 필드 단위로 매칭하기 때문에 전체 페이로드를 검사하는 부하를 줄일 수 있었다. 하지만 해시키의 비교를 위해서는 시그니쳐의 추출 시점에서 시그니쳐 추출 방법과 분류 시점에서 페이로드에서 필드를 추출하는 방법이 동일해야 적용이 가능한 방법이다.

[1]은 패킷의 페이로드 데이터에서 응용 트래픽 분류를 위한 시그니쳐가 매칭되는 offset 은 프로우의 첫 5 번째 패킷이며, 패킷 페이로드의 1,000Byte 이하 임을 증명하였다. 또한 Snort 에서는 패킷의 페이로드 내에서 시그니쳐가 나타나는 범위를 제한하기 위해서 offset, depth, within 과 같은 옵션을 제공하고 있다[22]. 이와 같이 패킷의 페이로드 내에서 시그니쳐의 존재 유무를 검사하는 범위를 offset 을 기준으로 제한함으로써 분류 시스템의 처리 속도를 향상 시킬 수 있다.

패턴 매칭 알고리즘은 Single string matching 과 Multiple string matching 알고리즘으로 구분된다. Single string matching 은 연속된 문자열로 구성된 단일 스트링을 페이로드와 1:1 로 매칭하여 성공 또는 실패의 결과를 반환한다. 대표적인 알고리즘으로 KMP(Knuth Morris Pratt), BM(Boyer Moore)이 사용된다[17]. Multiple string matching 알고리즘은 그 결과값을 수락 또는 실패로 제공하는 알고리즘과 여러 개의 서비스 ID 를 제공할 수 있는 알고리즘으로 구분할 수 있다. DFA 와 NFA 는 수락 또는 실패로 결과를 제공하기 때문에 동일한 분류 ID 를 갖는 시그니쳐들에 대해서만 하나의 오토마타로 구성할 수 있다[18]. 반면에 Aho-Corasick 은 복수 개의 서비스의 시그니쳐를 단일 트리로 구성하고 한번의 트리 탐색으로 각 분류 ID 값을 반환한다[17].

스트링 매칭 알고리즘의 시간/공간 복잡도를 최소화하기 위한 다양한 연구가 수행되고 있다[18, 19]. 하지만 모든 시그니쳐에 대해 최적의 성능을 보장할 수 없으며 시그니쳐 구성에 의존적인 결과로 나타난다. 따라서 시그니쳐의 기술 방법에 적합한 매칭 알고리즘을 적용해야 한다.

3. 제안하는 분류 방법론

본 장에서는 2 장에서 기술한 분석 시스템 설계 시 요구 사항을 기반으로 제안하는 페이로드 시그니쳐 기반 분석 시스템의 구현 방안을 제안한다. 그림 4 는 본 논문에서 제안하는 분류 시스템의 구성도를 보여주고 있다.

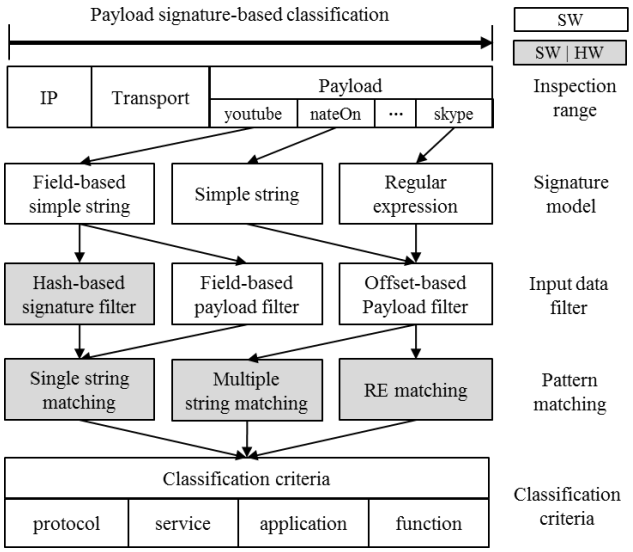

Figure 4. Diagram of payload signature-based classification system

페이로드 시그니쳐 기반 분류 시스템을 구성하기 위해서는 분류 기준을 정의하는 것이 우선이 되어야 한다. 본 논문에서는 프로토콜, 서비스, 응용 프로그램, 기능의 4 차원의 분류 기준을 제안한다. 4 차원의 분류 기준은 1 개의 플로우를 4 가지 관점으로 분류함으로써 QoS, SLA, 방화벽과 같은 분류 결과를 활용하는 단계에서 유연성을 제공할 수 있다.

분류 기준이 정의되면 분석 대상 응용의 트래픽의 특성을 분석하여 시그니쳐 모델을 정의해야 한다. 본 논문에서는 Field-based simple string, Simple string, Regular expression 3 가지 페이로드 시그니쳐 모델으로 구분한다. 각 응용 레벨 프로토콜의 특징을 고려하여 시그니쳐 모델을 선정한다. 표 3 은 각 시그니쳐 모델에 적합한 특징을 갖는 응용과 해당 응용의 시그니쳐를 보여주고 있다.

Table III. Signature example for each type

Model	Example	App.
Field-based simple string	User-agent : Mozilla; Host : www; Domain : youtube; Uri : login	youtube
Simple string	content : NCPT; depth : 4;	nateon
regular expression	RE : ^(\x01.? .? .? .? .? .? .? .?\x01| \x02.? .? .? .? .? .? .? .?\x02| … \xff.? .? .? .? .? .? .? .?\x0ff)	skype

youtube 와 같이 HTTP 를 이용하여 트래픽을 발생하는 응용은 Field-based simple string 형태의 시그니쳐 기술 방법이 적합하다. nateon 과 같은 응용은 페이로드의 특정 위치에서 나타나는 연속된 문자를 시그니쳐로 사용하는 Simple string 형태로 기술한다. skype 는 시그니쳐가 고정된 문자로 나타나지 않고, 특정 범위 내의 문자가 랜덤하게 발생하기 때문에 regular expression 형태의 기술 방법이 적합하다.

Input data filter 는 시그니쳐와 페이로드의 탐색 범위를 최소화할 수 있는 모듈이다. Hash-based signature filter, Field-based payload filter, Offset-based payload filter 로 구성된다. 각 모듈은 시그니쳐 모델과 시그니쳐 자동 추출 여부에 따라서 결정된다. 시그니쳐를 특정 응용의 프로토콜의 필드에서 자동으로 추출할 수 있다면 Hash-based signature filter 에 적용한다. 하지만 시그니쳐 추출 과정에서 관리자의 개입이 요구되면 시그니쳐와 페이로드에서 동일한 문자열을 추출할 수 없기 때문에 field-based payload filter 를 적용한다. 프로토콜의 특정 필드 단위로 구성된 시그니쳐가 아닌 경우에는 offset-based payload filter 가 적용된다.

패턴 매칭 알고리즘은 Single string, Multiple string, Regular expression matching 으로 구성된다. Single string matching 알고리즘은 Hash-based signature filter 에 의해서 매칭된 결과를 검증하기 위한 목적으로 패턴 매칭을 수행한다. 또한 Field-based payload filter 에 의해서 추출된 필드를 매칭하기 위해 적용된다. Multiple string matching 알고리즘은 Simple string 형태로 기술된 복수개의 시그니쳐를 단일 트리로 구성하고 트리에 대한 한번의 탐색으로 시그니쳐의 탐색 공간을 줄일 수 있는 패턴 매칭 알고리즘으로 Aho-Corasic 을 사용한다. Regular expression 형태로 기술된 시그니쳐는 Automata 를 이용하여 패턴 매칭을 수행한다.

페이로드 시그니쳐 기반 분석 시스템의 분류 정확성, 처리 속도를 향상 시키기 위해서는 분석 시스템의 세부 모듈의 부분적인 성능 향상과 함께 분류 기준과 시그니쳐 모델에 적합한 분석 시스템 구조를 설계해야 한다.

4. 결론 및 향후 과제

본 논문에서는 응용 레벨 트래픽 분류 분야의 페이로드 시그니쳐 기반 분류 방법의 성능 향상을 위한 기존의 연구를 분류 기준, 시그니쳐 모델, 입력 데이터 최소화 방법, 패턴 매칭 알고리즘 측면에서 조사하였다. 조사 결과를 바탕으로 분류 정확도, 처리 속도 향상을 위한 분류 시스템의 요구 사항을 도출하였다. 이를 기반으로 분류 결과의 활용도, 분류 정확도, 처리 속도 향상을 위한 페이로드 시그니쳐 기반 분류 시스템의 구조를 제안하였다.

본 논문에서는 범용 컴퓨터 환경에서 소프트웨어적으로 접근할 수 있는 성능 향상 기법을 제안하였다. 최근 GPU, FPGA 와 같은 하드웨어를 기반으로 분류 시스템의 성능을 향상 시킬 수 있는 방법이 제시되고 있는데, 본 논문에서 제안하는 소프트웨어적인 성능 향상 기법을 하드웨어와 접목하여 성능을 극대화할 수 있는 방안에 대한 연구를 수행할 계획이다.

참고 문헌

[1] J. S. Park, S. H. Yoon, M. S. Kim, “Software Architecture for a Lightweight Payload Signature-based Traffic Classification System”, Proc. Traffic Monitoring and Analysis Workshop, Vienna, Austria, pp. 136-149, Apr. 2011.

[2] A. Dainotti, A. Pescape, K. Claffy, “Issues and future directions in traffic classification”, IEEE Network: The Magazine of Global Internetworking, Vol. 26, No. 1, pp. 35-40, Jan. 2012.

[3] R. Antonello, S. Fernandes, D. Sadok, J. Kelner, “Characterizing Signature Sets for Testing DPI Systems”, Proc. IEEE GLOBECOM Management of Emerging Networks and Services Workshop, Houston, TX, USA, pp. 678-683, Dec. 2011.

[4] Aceto, G.., Dainotti, A., de Donato, W., Pescape, A., “PortLoad: taking the best of two worlds in traffic classification”, Proc. IEEE INFOCOM Conference on Computer Communications Workshops, San Diego, CA, USA, pp. 1-5, Mar. 2010.

[5] Huang, N. F., Jai, G. Y., Chao, H. C., Tzang, Y. J., Chang, H. Y., “Application traffic classification at the early stage by characterizing application rounds”, Information Sciences, Vol. 232, pp. 130-142, May 2013.

[6] T. Ban, S. Guo, M. Eto, D. Inoue, K. Nakao, “Towards Cost-Effective P2P Traffic Classification in Cloud Environment”, IEICE Transactions on Information and Systems, Vol. E95-D, No. 12, pp. 2888-2897, Dec. 2012.

[7] Khalife, J. M., Hajjar, A., Díaz-Verdejo, J., “Performance of OpenDPI in Identifying Sampled Network Traffic”, Journal of Networks, Vol. 8, No. 1, pp. 71-81, Jan. 2013.

[8] Ji-Hyeok Choi, Myung-Sup Kim, "Processing Speed Improvement of Traffic Classification based on Payload Signature Hierarchy," Proc. of the Asia-Pacific Network Operations and Management Symposium (APNOMS) 2013, Hiroshima, Japan, Sep. 25-27, 2013.

[9] Sung-Ho Yoon, Jun-Sang Park, and Myung-Sup Kim, "Signature Maintenance for Internet Application Traffic Identification using Header Signatures," Proc. of the 4th IEEE/IFIP International Workshop of the Management of the Future Internet (ManFI 2012), Hwaii, USA, Apr. 16, 2012.

[10] 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 Transactions on Knowledge Discovery from Data, vol. 6, no. 1, pp. 1-34, March, 2012.

[11] Sung-Ho Yoon, Myung-Sup Kim, "Behavior Signature for Big Data Traffic Identification ," Proc. of the International Conference on Big Data and Smart Computing (BigComp) 2014, Bangkok, Thailand, Jan. 15-17, 2014, pp. 261-266.

[12] A. Moore and K. Papagiannaki, “Toward the Accurate Identification of Network Applications,” in Passive and Active Network Measurement, ser. Lecture Notes in Computer Science, C. Dovrolis, Ed. Springer Berlin Heidelberg, 2005, vol. 3431, pp. 41–54.

[13]YANG, Baohua, et al. SMILER: towards practical online traffic classification. In:Proceedings of the 2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems. IEEE Computer Society, 2011. pp. 178-188.

[14] PARK, Byungchul; HONG, JW-K.; WON, Young J. Toward fine-grained traffic classification. Communications Magazine, IEEE, 2011, Vol. 49, No.7, pp. 104-111.

[15] LIU, Tingwen, et al. Improving matching performance of DPI traffic classifier. In:Proceedings of the 2011 ACM Symposium on Applied Computing. ACM, 2011. pp. 514-519.

[16] MU, Cheng, et al. Automatic traffic signature extraction based on fixed bit offset algorithm for traffic classification. The Journal of China Universities of Posts and Telecommunications, 2011, Vol. 18, pp. 79-85.

[17] CHAUDHARY, Ajay; SARDANA, Anjali. Software based implementation methodologies for deep packet inspection. In: Information Science and Applications (ICISA), 2011 International Conference on. IEEE, 2011. pp. 1-10.

[18] QI, Yaxuan, et al. Towards high-performance pattern matching on multi-core network processing platforms. In: Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE. IEEE, 2010. pp. 1-5.

[19] ANTONELLO, Rafael, et al. Deterministic finite automaton for scalable traffic identification: the power of compressing by range. In: Network Operations and Management Symposium (NOMS), 2012 IEEE. IEEE, 2012. pp. 155-162.

[20] LIU, Tingwen; SUN, Yong; GUO, Li. Fast and memory-efficient traffic classification with deep packet inspection in CMP architecture. In: Networking, Architecture and Storage (NAS), 2010 IEEE Fifth International Conference on. IEEE, 2010. pp. 208-217.

[21]L7-filter, http:// http://l7-filter.sourceforge.net/, accessed Aprl. 2014.

[22]Snort, http://www.snort.org, accessed Aprl. 2014.