탐색공간 최적화를 통한 시그니처기반 트래픽 분석 시스템 성능 향상☆

Performance Improvement of Signature-based Traffic Classification System by Optimizing the Search Space

박 준 상* 윤 성 호** 김 명 섭***
Jun-Sang Park Sung-Ho Yoon Myung-Sup Kim

요 약

인터넷에 기반한 응용 프로그램의 종류와 네트워크 대역폭이 증가하면서 페이로드 시그니처 기반 트래픽 분류 시스템에서 처리하는 데이터의 양이 급격하게 증가하고 있다. 대용량 트래픽 데이터에 대한 처리 속도를 향상시키기 위한 방법으로 다양한 패턴 매칭 알고리즘이 제안되고 있다. 하지만 비약적으로 늘어나는 시그니처의 수와 트래픽 양에 비해 패턴 매칭 알고리즘의 성능 향상 속도는 한정적이고, 입력데이터의 특성에 의존적인 성능을 나타낸다. 따라서 본 논문에서는 분류 시스템의 입력 데이터로 제공되는 트래픽 데이터와 시그니처의 탐색 공간을 최적화할 수 있는 분류 시스템 구조를 제안한다. 또한 제안하는 분류 시스템을 학내 망에서 발생하는 대용량의 트래픽에 실시간으로 적용하여 그 타당성을 증명한다.

ABSTRACT

The payload signature-based traffic classification system has to deal with large amount of traffic data, as the number of internet-based applications and network traffic continue to grow. While a number of pattern-matching algorithms have been proposed to improve processing speedin the literature, the performance of pattern matching algorithms is restrictive and depends on the features of its input data. In this paper, we studied how to optimize the search space in order to improve the processing speed of the payload signature-based traffic classification system. Also, the feasibility of our design choices was proved via experimental evaluation on our campus traffic trace.

☞ keyword : Application-Level Traffic Classification, Payload-Signature, Real-time classification, 응용 레벨 트래픽 분류, 시그니처 기반 분석, 실시간 분석

1. 서 론

네트워크의 고속화와 더불어 다양한 서비스와 응용프로그램이 개발됨에 따라 기업이나 개인들은 인터넷으로 대표되는 네트워크에 대한 의존성이 상당히 커져가고 있다. 이와 같은 현실 속에서 네트워크의 효율적 운용과 관리를 위한 응용 레벨의 트래픽의 모니터링과 분석은 네트워크 사용 현황 파악과 확장계획 수립 등의 다양한 분야에서 필요성이 커져가고 있다. 예를 들어 종량제 과금, CRM, SLA, 보안 분석 등 트래픽 모니터링 및 분석에 대한 필요성은 지금뿐만 아니라 앞으로 더욱더 크게 증가할 것이다. 이를 위해서는 다양한 종류의 응용 레벨 트래픽을 정확하게 분류할 수 있는 방법과 고속 링크에서 발생하는 대용량의 트래픽을 실시간으로 처리하는 방법이 요구된다.

* 정 회 원 : 고려대학교 대학원 컴퓨터정보학과 박사과정
junsang_park@korea.ac.kr
** 정 회 원 : 고려대학교 대학원 컴퓨터정보학과 박사과정
sungho_yoon@korea.ac.kr
*** 정 회 원 : 고려대학교 컴퓨터정보학과 부교수
tmskim@korea.ac.kr
[2010/12/18 투고 - 2011/01/04 심사(2011/02/28 2차 - 2011/04/18 3차) - 2011/04/27 심사완료]
☆ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단-미래기반기술개발사업(첨단융복합분야)의 지원을 받아 수행된 연구임(20100020728).

응용 레벨 트래픽 분류 방법에 있어 페이로드 시그니처 기반 분석 방법은 패킷의 헤더 정보나 통계 정보를 이용하는 다른 분석 방법들에 비해 상대적으로 높은 분류 정확성과 분석률을 보인다.[1-3,6,7] 하지만 전송 기술의 발달로 10G Ethernet이 일반화되고 상황에서 페이로드 시그니처 기반 분류 시스템의 처리 속도는 고속 네트워크 상에서 발생하는 대용량 트래픽을 실시간으로 처리하기에 부적합한 방법이다. 또한 응용의 수와 대용량의 트래픽을 발생시키는 응용의 사용이 증가하고 있는 추세를 고려했을 때 페이로드 기반 분석 방법의 처리 속도 문제는 반드시 해결되어야 하는 과제이다. 이러한 문제점을 해결하기 위하여 기존 연구에서는 패턴 매칭 알고리즘의 처리 속도 향상에 관한 연구가 주를 이루었다.[4,5,8] 하지만 패턴 매칭 알고리즘의 처리 속도는 입력 데이터의 크기와 특성에 의존적인 결과를 보이며, 제한적인 성능 향상을 나타낸다. 따라서 본 논문에서는 페이로드 시그니처 기반 분류 시스템의 처리 속도에 영향을 미치는 요소를 입력 데이터의 관점에서 정의하고, 처리 속도 향상을 위한 분류 시스템의 구조를 제안한다.

본 논문의 구성은 다음과 같다. 본 장의 서론에 이어, 2장에서는 관련 연구에 대해 기술하며, 3장에서는 제안하는 방법론의 성능 평가에 사용되는 기준 시스템의 구성과 트래픽 데이터에 대하여 설명한다. 4장에서는 처리 속도에 영향을 미치는 요소를 정의하고, 실험 결과를 바탕으로 탐색 공간을 최적화하기 위한 방법을 제안한다. 5장에서는 4장의 결과를 바탕으로 분류 시스템의 구조를 기술한다. 6장에서는 제안하는 방법을 학내 망의 트래픽에 적용하여 그 타당성을 증명한다. 마지막으로 7장에서는 결론 및 향후 연구에 대해 기술한다.

2. 관련 연구

네트워크 기반의 응용의 증가로 인해 응용 프로그램을 분류하기 위한 시그니처의 개수가 증가하고 있다. 시그니처의 복잡도가 커지고, 개수가 증가하면서 페이로드 시그니처 기반 분류 시스템의 처리 속도는 트래픽 분류 시스템의 성능을 결정하는 중요한 요소로 작용하게 되었다.

분류 시스템의 처리 속도 향상을 위해 패턴 매칭 알고리즘의 성능 향상을 위한 방법을 제안하고 있다.

(표 1) 시그니처 기반 분석 시스템의 성능 평가

Tool	Signature Format	# of Signature	bps	Matching Algorithm
L7-Filter	RE	About 70	Less than 10Mbps	NFA
Snort	ES + RE	About 5000	Less than 100Mbps	DFA

(표 1)은 페이로드 시그니처 기반 분석 시스템으로 많이 알려진 L7-filter와 Snort의 시그니처 구성과 처리 속도를 나타낸다. 응용 레벨 트래픽 분류를 위한 도구로 많이 사용되고 있는 L7-filter는 시그니처를 정규 표현식으로 기술하고 패턴 매칭 알고리즘으로 NFA(Nondeterministic Finite Automata)를 적용한다. 하지만 70여 개의 시그니처를 적용하였을 때 3.5Mbps 이하의 처리 속도를 보인다.[4] NFA의 처리 속도를 향상 시키기 위해 DFA(Deterministic Finite Automaton) 기반의 분석이 제안되고 활용되고 있지만 Snort의 경우 100Mbps 이하의 처리속도를 갖는다.[4,5,9]

모든 입력 데이터에 대하여 최적화되어 있는 매칭 알고리즘은 존재하지 않으며, 입력 데이터의 구성에 의존적이다.

(표 2)는 매칭 알고리즘의 처리 속도에 영향을 미치는 요소를 기반으로 4가지 유형의 시그니처로 구분하고, 각 시그니처의 유형에 따른 분류 알고리즘의 처리 속도를 비교한 결과이다. 4가지 스트링 매칭 알고리즘은 시그니처 유형에 따라서 다른 처리 속도를 보이는 것을 알 수 있다. 이와 같이 모든 입력 데이터에 대하여 항상 최적의 성

(표 2) 시그니처 유형 별 알고리즘의 성능 비교

Matching Algo. Signature Type		Robin-Karp	DFA Full	NFA Partial	NFA Full
Explicit String	Fixed offset	***0.03***	0.05	0.08	0.08
	Variable offset	1.28	***0.32***	0.90	0.42
Regular Expression	".*" <= 2	3.45	0.19	***0.08***	0.16
	".*" > 3	1.35	0.06	***0.05***	0.55

능을 보장할 수 있는 알고리즘은 존재하지 않는다.

따라서 본 논문에서는 시그니처 매칭 속도를 향상 시키기 위하여 입력 데이터를 처리하기 위한 매칭 알고리즘의 탐색 공간을 최적화할 수 있는 방법을 제시한다.

3. 실험 환경 및 트래픽 데이터

본 장에서는 제안하는 트래픽 분류 시스템의 성능 비교를 위한 기준 시스템과 실험에 사용된 트래픽 데이터에 대하여 기술한다.

3.1 기준 트래픽 분류 시스템

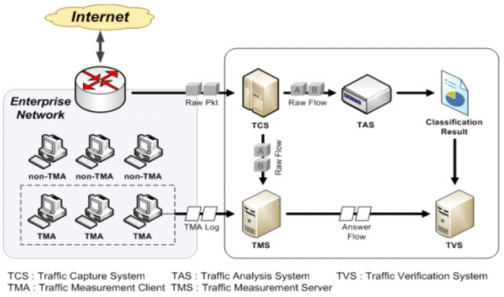
우리는 선행 연구를 통해 학내 망에서 발생하는 모든 트래픽을 페이로드 시그니처 기반으로 분류하는 시스템을 구축하였다.[1] 기존 시스템은 트래픽을 수집과 분석 시스템으로 구분된다. 트래픽 수집 시스템은 학내 망에서 발생하는 모든 패킷을 손실없이 양방향의 플로우로 생성한다. 트래픽 분석 시스템은 분 단위로 수집된 모든 플로우에 대해 882개의 페이로드 시그니처를 전수 조사하여 트래픽을 분류하며, 1분 평균 160Mbps의 처리량을 나타낸다.

트래픽 분석 시스템은 Core2 Duo E7200 2.53GHz의 CPU와 3GByte 메모리로 구성된 범용 컴퓨터 수준의 사양을 갖추고 있다.

(표 3)은 분석 시스템의 모듈별 소비 시간 비율을 측정한 결과이다. 85% 이상은 시그니처 매칭 모듈에 의해서 소비됨을 알 수 있다. 본 논문에서는 시그니쳐를 매칭 시간을 최소화할 수 있는 방법을 제시한다.

(표 3) 트래픽 분석 시스템 모듈별 소비 시간 비율

Module	Load Sig.	Load Traffic	Matching	Etc
Execute Time Ratio	0.10%	1.23%	85.15%	14.52%



(그림 1) 분류 시스템 및 검증 네트워크 구성

3.2 트래픽 트래이스

본 절에서는 실험에 사용한 트래픽에 대한 설명과 정확성을 검증하기 위한 환경에 대하여 기술한다.

(그림 1)은 학내 망에 구축되어 있는 트래픽 분석 시스템과 검증 네트워크의 구성을 나타내고 있다. TCS에 의해서 학내 망에서 발생하는 모든 트래픽을 수집하고, TAS에 의해 응용 계층 트래픽이 분석된다. 분류된 결과의 정확성을 증명하기 위하여 TMA[1]를 기반으로 정답지 트래픽을 수집한다. TMA는 종단 호스트에 설치되며, 프로세스 이름을 포함한 소켓 정보를 TMS에 전송하고, TMS에서는 TMA로부터 전송받은 데이터를 통합하여 정답지 데이터를 생성한다. TVS는 정답지를 기반으로 분류의 정확성을 측정하고 보고한다.

(표 4) 실험에 사용된 트래픽 트래이스 구성

Unit	TCP	UDP	Total
Flow	3,972,069	2,462,886	6,434,955
Packet(K)	218,013	113,094	331,107
Byte(MB)	188,415	100,332	288,747

(표 4)는 실험에 사용되는 트래픽 트래이스의 구성을 나타내고 있다. 트래픽 데이터는 300Mbps 링크 크기를 갖는 학내 망과 인터넷의 연결 지점에서 수집되었다. 3000여대의 호스트에서 발생하는 다양한 응용 트래픽으로 구성되어 있다.

4. 입력 데이터 최적화 방법

본 장에서는 페이로드 시그니처 기반 분석 시스템에서처리하는 입력 데이터의 탐색 공간을 최적화하는 방법을 제시한다. 처리 시간에 영향을 미치는 요소를 트래픽과 페이로드 시그니처로 구분하여 정의하며 실험적 결과를 바탕으로 기술한다.

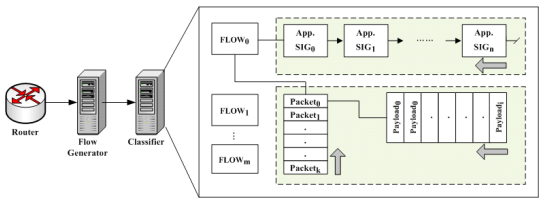
(그림 2)는 분류 시스템의 탐색 공간 최적화를 위한 입력 데이터의 구조를 도식화하여 보여주고 있다.

4.1 트래픽 탐색 공간 최적화

매칭 모듈의 입력 데이터로 제공되는 모든 트래픽 데이터에 대해 시그니처 매칭 과정을 수행하면 불필요한 탐색 공간의 비교로 인한 오버헤드가 발생한다. 따라서 본 절에서는 트래픽 데이터의 탐색 공간을 최적화하기 위한 방법을 제시한다.

4.1.1 PBMS vs. SBMS

응용프로그램의 시그니쳐 매칭 방법은 매칭 단위에 따라서 PBMS(Packet-based Matching System)과 SBMS(Stream-based Matching System)으로 구분할 수 있다. PBMS는 패킷 단위에서 추출된 페



(그림 2) 분류 시스템 입력 데이터 구조

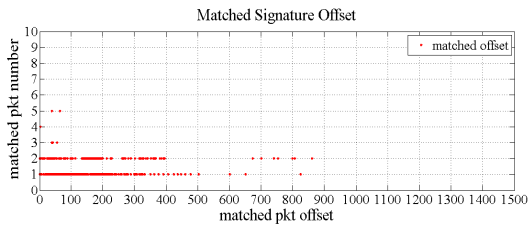
이로드 시그니처를 플로우의 초기 패킷부터 순차적으로 매칭하며 매칭이 발생하면 다음 패킷은 검사하지 않는다. SBMS는 시그니처의 매칭을 위해 1개 이상의 패킷을 스트림 형태로 재조합하여 시그니처 매칭을 수행하기 때문에 PBMS에 비해 상대적으로 많은 수의 패킷을 조사해야 한다. 또한 SBMS는 패킷의 재조합을 요구하기 때문에 분류 시스템의 부가적인 프로세싱 과정과 저장 공간이 필요하며, 패킷의 손실, 비대칭 라우팅으로 인해 플로우의 스트림이 완전하게 구성되지 못하면 분류가 불가능하게 되는 문제점이 발생한다. 따라서 패킷 단위의 시그니처 매칭이 필요하다.

4.1.2 조사하는 패킷 개수 최적화

플로우 내의 조사하는 패킷의 개수를 제한함으로써 트래픽에 대한 탐색 시간을 감소 시킬 수 있다. (표 5)는 하루 동안 학내망에서 발생하는 실제 트래픽을 기반으로 조사되는 플로우의 초기 패킷 개수를 증가 시키면서 평균 분석률과 정확도를 측정한 결과이다. 이 때 첫 번째 패킷은 TCP연결 설정 이후 페이로드가 존재하는 최초의 패킷으로 정의한다. 패킷 개수의 제한에 따른 분석 결과 분류의 정확성과 분석률은 5번째 패킷 이후에는 동일한 것을 알 수 있다. 이는 페이로드 시그니처가 서버-클라이트 사이에서 컨트롤 패킷을 전송하는 과정에서 추출되기 때문이다. 대부분의 플로우는 콘텐츠를 전송하기 전에 응용 레벨 프로토콜에 의해서 약속된 컨트롤 패킷을 전송한다. 시그니처는 컨트롤 패킷을 보내는 단계에서 추출된다. 따라서 조사하는 패킷의 개

(표 5) 조사하는 패킷 개수에 따른 분석률과 정확도

		Pkt1	Pkt2	Pkt3	Pkt4	Pkt5	Pkt6	Pkt7
Complete-ness	*Flow*	92.3	93.1	93.1	93.2	93.2	93.2	93.2
	Pkt	82.4	85.2	86.5	86.7	86.8	86.8	86.8
	Byte	77.5	81.1	82.2	82.5	82.7	82.7	82.7
Overall Accuracy	*Flow*	96.2	96.3	96.3	96.3	96.3	96.3	96.3
	Pkt	98.6	98.7	98.7	98.7	98.7	98.7	98.7
	Byte	97.4	97.6	97.6	97.6	97.6	97.6	97.6



(그림 3) 시그니처가 매칭되는 위치

수를 5개로 제한함으로서 분류 시간을 감소시킬 수 있는 효과가 있다.

4.1.3 조사하는 패킷 크기 최적화

1개의 패킷 내에서 조사하는 페이로드 데이터 크기를 제한함으로써 트래픽 데이터의 탐색 공간을 감소시킬 수 있다.

(그림 3)은 기존 분류 시스템[1]을 활용하여 하루 동안 학내망에서 발생하는 트래픽을 대상으로 HTTP시그니처를 제외한 시그니처의 매칭 위치를 나타내고 있다. 기존 분류 시스템은 플로우의 모든 패킷과 패킷 전체 페이로드를 조사한다. 대부분의 시그니처는 패킷의 500byte, 플로우의 2번째 이하에서 매칭됨을 알 수 있고 최대 5번째 패킷과 1000bytes 이하에서 매칭이 발생하는 것을 알 수 있다. HTTP프로토콜의 시그니처는 1000bytes 이상에서 매칭되는 경우가 발생하는데 이는 HTTP 트래픽의 요청 패킷을 구성하는 페이로드 데이터에 파일에 대한 경로정보가 1000 bytes 이상으로 구성되는 경우에 시그니처가 1000bytes 이상에서 발생한다.

(표 6) 시그니처 재기술 결과

Feature	Before rewriting	After rewriting
% of patterns start with '^', '$'	45.45%	50.78%
"." > 3*	6.49%	3.48%
% of patterns with "[]"	0%	0%

본 논문에서는 (그림 3)의 실험 결과에 의해서 HTTP 시그니처가 아닌 경우 5번째 패킷과 1000bytes 이하의 페이로드 데이터만을 조사한다. HTTP 시그니처는 5번째 이하의 패킷과 패킷의 모든 페이로드 데이터를 적용한다.

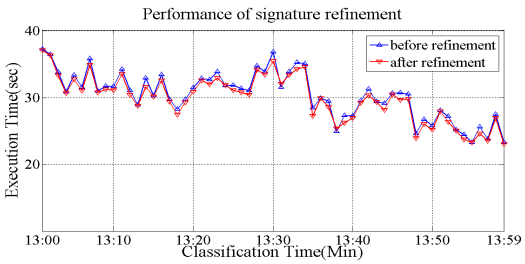
4.2 시그니처 탐색 공간 최적화

본 절에서는 시그니처에 대한 불필요한 탐색 공간을 최적화하여 분류 시스템의 처리 속도를 향상 시키는 방법을 제시한다.

4.2.1 매칭 알고리즘에 최적화된 시그니처 기술

시그니처를 명시적인 단일 스트링 형태로 표현하는 방법에서 정규 표현식 형태로 기술하는 방법이 늘어나고 있다. 정규 표현식을 기술하는 방법 중 '^', '$'의 사용 여부와 '.', '*', '?', "[]"등의 와일드 카드와 클래스의 사용 빈도는 매칭 알고리즘의 처리 속도에 영향을 미치는 요소이다. '^'과 '$'의 사용은 시그니처의 시작과 마지막 바이트를 명시적으로 표현해 주어 빠른 시점에 Miss-match를 결정할 수 있으며, 시그니처의 탐색 공간을 감소시킬 수 있다. 와일드 카드와 클래스의 사용은 유한 오토마타를 구성하는 상태의 수를 급격하게 증가시키기 때문에 탐색 공간이 많아지고 처리 속도가 감소된다. '^', '$'의 사용 빈도를 높이고, 와일드 카드와 클래스의 사용을 자제하여 표현해야 한다.

(표 6)은 882개의 시그니처에 대해 재기술 전후의 각 특징에 대한 시그니처 비율을 나타내고 있다.



(그림 4) 시그니처 재기술에 따른 처리 시간 비교

'^'과 '$'를 포함하는 표현은 증가하였고, ".*"를 3개 이상 사용하는 비율은 감소하였다. 또한 클래스에 의한 시그니처 표현은 사용하지 않았다.

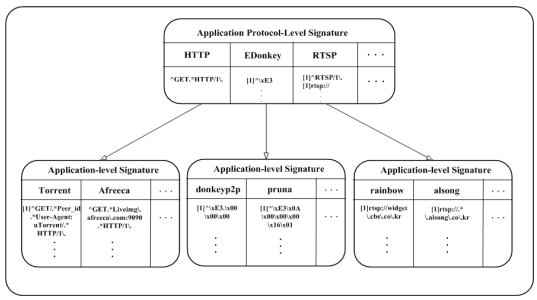
(그림 4)는 매칭 알고리즘의 시그니처 탐색 공간을 최적화할 수 있도록 시그니처를 재기술하기 전과 후의 분석 시간을 표현한 결과이다. 재기술하기 전보다 1-2초 정도의 분류 속도가 향상되었음을 알 수 있다. 이는 와일드 카드의 사용 빈도를 줄이고, '^'과 '$'를 이용한 Fixed offset 형태의 시그니처 표현이 증가했기 때문이다.

4.2.2 시그니처 계층 구조 기반 분석

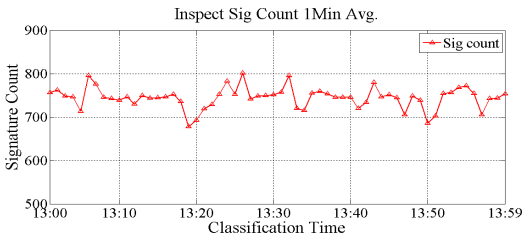
페이로드 시그니처는 시그니처 사이의 포함 관계를 기반으로 계층 구조로 표현할 수 있다. 이러한 특징에 기반하여 시그니처 매칭을 위한 메모리 구조를 계층적으로 구성하여 시그니처의 탐색 공간을 최적화하는 방법을 제시한다.

응용 프로그램 시그니처는 (그림 5)와 같이 응용 레벨 프로토콜 시그니처와 응용 프로그램 시그니처의 2단계 계층 구조로 표현 할 수 있다. 시그니처 S_x에 의해 분류되는 트래픽이 시그니처 S_y에 의해 모두 분류되면 S_y는 S_x를 포함한다. 이때 S_y는 S_x의 응용 프로토콜 레벨 시그니처가 되며, S_x는 S_y의 응용 프로그램 레벨 시그니처가 된다.

이와 같은 계층 구조는 응용 레벨 프로토콜 단위로 분석 후 해당 응용 레벨 프로토콜에 포함되는 응용 프로그램 시그니처만을 탐색하여 분류 시스템의 시그니처 탐색 공간을 줄여 처리 시간을 단축할 수 있다.



(그림 5) 페이로드 시그니처의 계층 구조



(그림 6) 요구되는 평균 시그니처 개수

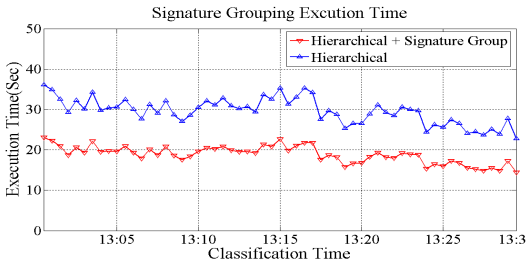
(그림 6)은 1개의 플로우를 분석하기 위해 요구되는 평균 시그니처 개수를 나타내고 있다. 기존의 분석 방법은 1개의 플로우를 분석하기 위해 모든 시그니처에 대해 비교하는 구조를 갖지만 제안하는 계층 구조 기반 분석 방법은 (그림 6)과 같이 100개 이상의 시그니처에 대한 탐색 공간을 감소할 수 있어 분류 시스템의 처리 속도를 향상시킬 수 있다.

4.2.3 MSSA(Multiple Signature Single Automata)

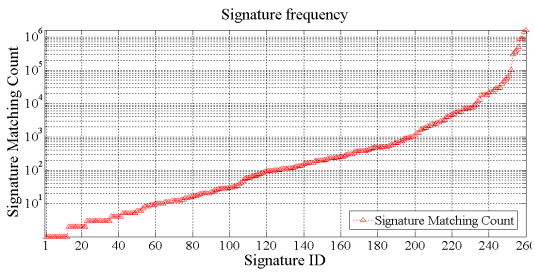
분류 시스템은 정규 표현식으로 기술된 시그니처를 유한 오토마타로 구성한 후 매칭 과정을 수행한다. 본 논문에서는 시그니처의 탐색 공간을 줄이는 방법으로 단일 시그니처를 단일 오토마타로 구성하는 형태에서 1개 이상의 시그니처를 오토마타로 구성하는 방법을 제시한다. 하지만 1개 이상의 시그니처를 유한 오토마타로 구성했을 경우 오토마타를 구성하는 상태의 개수가 급격하게 증가하여 오히려 매칭 시간이 증가되

(표 7) 시그니처 그룹핑 전후 처리 시간 비교

	Signature	DFA Full	NFA Partial
SSSA	*^GET.*NateOn.**	0.23sec	0.05 sec
	*^GET.*nateon\.nate\.com.**	0.22 sec	0.06 sec
	*^GET.*adimg\.nate\.com.**	0.18 sec	0.11 sec
	*^GET.*cyad\.nate\.com.**	0.25 sec	0.15 sec
	*^GET.*nateonipml\.nate\.com.**	0.21 sec	0.10 sec
MSSA	*^GET.*(NateOn)|(nateon\.nate\.com)| (adimg\.nate\.com)|(cyad\.nate\.com)| (nateonipml\.nate\.com).**	0.22 sec	9.24 sec



(그림 7) 시그니처 그룹핑 전후 성능 비교



(그림 8) 시그니처 발생 빈도

는 문제점이 발생할 수 있다. 따라서 이러한 방법은 단일 시그니처를 단일 오토마타로 구성하여 분류하는 시간의 합보다 그룹 시그니처의 오토마타로 분류하는 시간이 짧을 때 분석 시간의 단축을 기대할 수 있다. (표 7)은 국내에서 많이 사용되는 네이트온 메신저의 시그니처에 대해 그룹 전 후의 DFA와 NFA의 분석 시간을 측정한 결과이다.

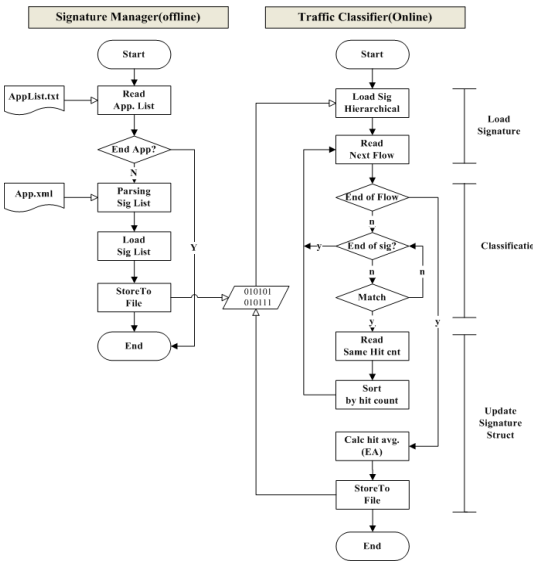
MSSA(Multiple Signature Single Automata) 형태의 시그니처는 SSSA(Single Signature Single Automata) 형태의 5개의 시그니처를 그룹한 시그니처이다. DFA의 경우 그룹 전후의 시그니처에 대해 절대적인 시간을 소비하지만 NFA의 경우 그룹 후 급격하게 분류 시간이 증가하는 것을 알 수 있다. (표 7)의 결과를 통해 시그니처를 그룹 후 MSSA로 구성하여 분류함으로서 분류 시간을 단축할 수 있음을 알 수 있다.

(그림 7)은 시그니처 그룹핑에 의한 처리 속도 향상 결과를 나타낸다. 그룹핑 전의 처리 속도보다 20초 이상의 처리 속도를 감소 시킬 수 있었다. 이는 시그니처의 그룹을 통해 시그니처 개수를 감소시켜 시그니처에 대한 탐색 공간을 줄일 수 있었기 때문이다.

4.2.4 트래픽 발생 패턴에 기반한 분석

응용 프로그램 트래픽의 발생 특성을 네트워크를 구성하는 단위로 분석하였을 때 네트워크마다 사용하는 응용의 개수가 제한적이다. 또한 트래픽이 발생하는 시간 추이로 분석하였을 때 응용 별로 트래픽의 지역성이 존재한다. 하지만 기존의 분석 방법은 이러한 트래픽의 발생 패턴을 분류 시스템에 반영하지 못 한다.

(그림 8)은 기존의 분류 시스템을 기반으로 5시간 동안의 트래픽에 대해 시그니처 별로 Hit Count를 누적한 결과이다. 총 882개의 시그니처 중 260개의 시그니처만이 Hit가 발생하였고, 1000번 이상의 Hit Count를 갖는 시그니처는 60여개

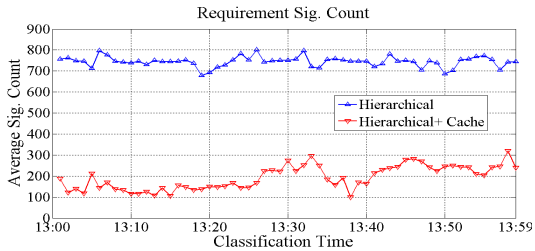


(그림 9) 분류 시스템의 흐름도

의 시그니처로 나타났다. 즉 특정 시간에 발생되는 응용의 개수는 제한적이고, 응용 프로그램 내에서도 소수의 시그니처에 의해서 대부분의 트래픽이 분류됨을 알 수 있다.

(그림 8)에서 알 수 있듯이 조사 대상 네트워크에서 사용되는 시그니처의 개수는 제한적이고, 나타나는 응용 프로그램의 트래픽은 시간적인 지역성 갖는다. 이와 같은 응용 트래픽의 지역성을 분류 시스템에 반영하여 시그니처의 탐색 공간을 감소 시킬 수 있다. 본 논문에서는 시그니처 메모리 구조를 Hit Count를 기준으로 동적으로 변화시켜 발생 빈도가 높은 시그니처를 선행적으로 검사하는 방법으로 시그니처의 탐색 공간을 최적화한다. 또한 Hit Count(HC) 값을 아래의 수식과 같이 Exponential Average(A) 를 적용하여 지속적으로 갱신하면서 시간의 변화에 따른 응용 트래픽의 변화에 유연하게 대처할 수 있다. 이때 값은 현재의 Hit Count를 높게 반영하는 0.6~0.9까지 변경하면서 실험하였고, 가장 높은 Cache Hit Ratio를 나타내는 0.6으로 설정하였다.

$$A_n = \alpha \times HC_n + (1-\alpha) \times A_{n-1}$$



(그림 10) 요구되는 시그니처의 개수

(표 8) 분류 시스템의 구조

Coverage	*Process*	*Application*	*Signature*
	260	126	882
Classification Criteria	Application(Set of Process)		
Classification Unit	Bidirectional Flow		
Matching Unit	Packet		
Inspected Packet offset	1st~5th packet in flow, 1000byte in packet		
Automata Type	MSSA		
Classification Cycle	Every 1minute		

(그림 9)는 Hit Count와 Exponential Average를 이용하여 시그니처를 동적으로 갱신하는 분류 시스템의 흐름도를 나타내고 있다.

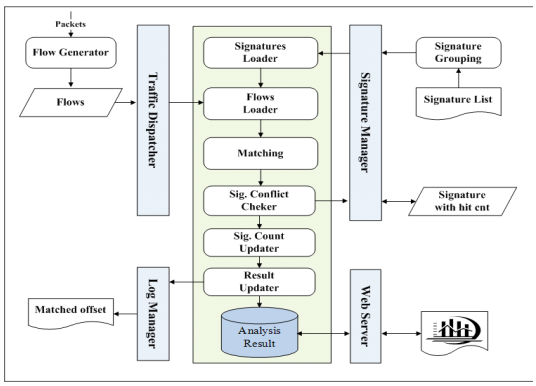
(그림 10)은 1개의 플로우를 분류하기 위해 요구되는 시그니처의 평균 개수를 나타낸다. 시그니처의 매칭 횟수를 기반으로 시그니처 충돌 관계를 반영하기 때문에 전수 조사 과정이 불필요하고, 발생 빈도에 따라 시그니처를 검사하기 때문에 기존의 방법에 비해 4~5배의 시그니처 탐색 공간을 감소 시킬 수 있다.

5. 분류 시스템 구현

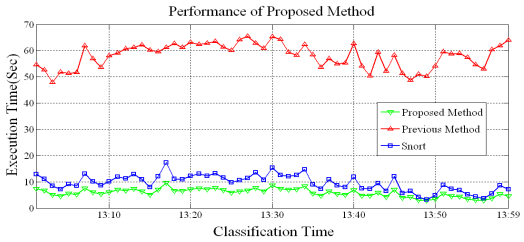
본 장에서는 4장에서 제안하는 디자인 선택 사항을 기반으로 분류 시스템을 설계하고 구현한다.

(표 8)은 4장에서 제안하는 방법을 종합적으로 적용한 결과와 분류 시스템의 구조를 보여주고 있다.

(그림 11)은 분류 시스템의 구조를 나타내고 있



(그림 11) 분류 시스템 구조



(그림 12) 제안하는 시스템의 성능 평가

다. 분류 시스템은 Traffic Dispatcher를 통해 1분 동안 수집된 플로우 데이터와 MSSA를 구성하기 위한 그룹핑되어진 시그니처를 입력데이터로 받는다. 분류된 결과는 웹 페이지를 통하여 관리자에게 분류 결과를 제공하며 Log Manager는 시그니처의 매칭 위치에 대한 정보를 저장한다.

매칭 결과를 바탕으로 시그니처의 Hit Count정보는 는 1분 주기로 Exponential Average 값을 계산하여 갱신한다. Exponential Average를 적용함으로써 시간의 흐름에 따라 적응적으로 시그니처 Hit Count를 조절할 수 있다.

6. 실험 및 결과 분석

본 장에서는 5장에서 기술한 분류 시스템을 기반으로 제안하는 분류 시스템의 성능을 평가한다.

(그림 12)는 제안하는 트래픽 분류 시스템의 처리 속도에 대한 결과를 나타내고 있다. 실험 결과는 1분 동안의 트래픽 데이터를 처리하는 분류 시스템의 처리 시간을 비교한다.

제안하는 분류 시스템의 성능은 기존의 방법[1]과 비교해 5배 이상의 처리 속도 향상을 나타낸다. 또한 Snort 기반의 분석 방법에 비해 1.5정도의 처리 속도 향상을 나타내고 있다. 기존의 방법은 트래픽에 따라서 분석 시간이 결정되지만 제안하는 방법은 각 분석 시점의 트래픽 양이 다르지만 유사한 분석 시간을 나타내고 있다. 이는 발생 빈도가 잦은 시그니처를 우선적으로 검사하여 시그니처의 탐색 공간이 줄이고, 조사하는 패킷의 크기와 개수를 줄일 수 있었기 때문이다.

7. 결론 및 향후 과제

본 논문에서는 페이로드 기반 응용 레벨 트래픽 분류 시스템의 처리 속도에 영향을 미치는 요소들을 입력 데이터의 탐색 공간을 기준으로 정의하였다. 각 요소를 실험적으로 평가하고 효율적인 분류 시스템을 구성하기 위한 방법을 제안하였다. 제안한 방법은 분류 시스템의 처리 속도를 기존의 분류 방법에 비해 5배 이상 향상시킬 수 있었다.

제안하는 방법은 범용 컴퓨터 환경에서 소프트웨어 적으로 분류 시스템의 처리 속도를 향상시킬 수 있는 방법이다. 향후 연구로 제안하는 방법을 하드웨어와 접목하여 대용량의 링크에서 실시간으로 분석 가능한 분류 시스템을 디자인하는 연구를 진행할 계획이다.

참 고 문 헌

[1] Jun-Sang Park, Jin-Wan Park, Sung-Ho Yoon, Young-Suk Oh, Myung-Sub Kim.: Development of signature Generation system and verification network for application-level traffic classification. In: Conference of Korea Information Processing

Society, Busan, Apr. 23-24, 2009, Vol.16, No. 1, pp. 1288-1291.

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

[3] F. Risso, M. Baldi, O. Morandi, A. Baldini, and P. Monclus.: Lightweight, Payload-Based Traffic Classification An Experimental Evaluation. In : IEEE International Conference on Communications, Beijing, China, May. 19-23, 2008, pp. 5869-5875.

[4] Fnag Yu, Zhifeng Chen, Yanlei Dino, T. V. Lakshman, Randy H. Katz.: Fast and memory Efficient Regular Expression Matching for Deep Packet Inspection. In : ANCS 2006, December, 2006, San jose, California USA.

[5] Christopher L. Hayes, Yan Luo.: DPICO: a high speed deep packet inspection engine using compact finite automata. In : ACM/IEEE Symposium on Architecture for networking and communications systems, December 03-04, 2007, Orlando, Florida, USA.

[6] Liu, Hui Feng, Wenfeng Huang, Yongfeng Li, Xing.: Accurate Traffic Classification. In : Networking, Architecture, and Storage, NAS 2007. International Conference.

[7] Byung-Chul Park, Young Won, Mi-Jung Choi, Myung-Sup Kim, and James W. Hong.: Empirical Analysis of Application-Level Traffic Classification Using Supervised Machine Learning. In : Proc. of the Asia-Pacific Network Operations and Management Symposium (APNOMS) 2008, LNCS5297, Beijing, China, Oct. 22-24, 2008, pp. 474-477.

[8] G. Vasiliadis, M. Polychronakis, S. Antonatos, E. P. Markatos, and S. Ioannidis.: Regular expression matching on graphics hardware for intrusion detection. In : RAID, 2009, pp. 265 – 283.

[9] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.: Introduction to Algorithms, Second Edition. In : MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Chapter 32: String Matching, pp.906 – 932.

◐ 저 자 소 개 ◑

박 준 상

2008년 고려대학교 컴퓨터정보학과 졸업(학사)
2010년 고려대학교 대학원 컴퓨터정보학과 졸업(석사)
2011년~현재 고려대학교 대학원 컴퓨터정보학과 박사과정
관심분야 : 네트워크 관리 및 보안, 트래픽 모니터링 및 분석.
E-mail : junsang_park@korea.ac.kr

윤 성 호

2009년 고려대학교 컴퓨터정보학과 졸업(학사)
2011년 고려대학교 대학원 컴퓨터정보학과 졸업(석사)
2011년~현재 고려대학교 대학원 컴퓨터정보학과 박사과정
관심분야 : 네트워크 관리 및 보안, 트래픽 모니터링 및 분석.
E-mail : sungho_yoon@korea.ac.kr

김 명 섭

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