시그니처 기반의 실시간 트래픽 분류 알고리즘의 성능 향상

오영석*, 윤성호*, 박준상*, 김명섭*
*고려대학교 컴퓨터정보학과
e-mail : {840105, hiption, runtoyou, tmskim}@korea.com

Performance Improvement of Real-time Traffic Classification Algorithm based on Application Signature

Young-Seok Oh*, Sung-Ho Yoon*, and Myung-Sup Kim*
*Dept. of Computer and Science, Korea University

요 약

현재 다량의 네트워크 대역폭을 소모하는 응용 프로그램 트래픽을 확인하고 분류하는데 많은 방법들이 사용되고 있지만 정통적인 트래픽 분류 방법론인, 포트 번호, ip 등 등의 헤더 정보만으로는 응용 프로그램의 트랙픽을 정확하게 분류하지 못한다. 최근 동적인 포트 번호를 사용하는 새로운 트래픽 응용의 등장과 방화벽을 통과하기 위한 포트번호 변경으로 인하여 전통적인 TCP/UDP 헤더 기반의 트랙픽 분류 방법은 부정확해지고 있다. 이러한 트래픽을 정확하게 식별하고 분류하기 위해서는 패킷의 페이로드 내용에 대한 조사도 병행되어야 하고 시그니처 기반의 식별 방법을 사용하여야 한다. 하지만 이 방법은 정확도가 높은 반면 시그니처의 목록을 매번 최신 상태로 유지하여야 하는 단점과 길어지는 탐색 시간에 따른 시스템 부하의 문제를 가지고 있다. 본 연구에서는 이러한 단점을 향상시키는 목적으로 새로운 시그니처 기반의 해쉬 테이블에 캐시를 이용한 방법론인 효율적인 알고리즘을 제안하고 시그니처의 자료구조와 실제 패킷과 시그니처의 비교 방식을 수정함으로써 효율성을 높이는데 목적을 두고 있다.

1. 서론

최근 인터넷이 급속도로 발전하면서 이메일, 온라인 쇼핑, 온라인 뱅킹, 채팅, 온라인 게임 및 멀티미디어 스트리밍, p2p / 웹 디스크를 이용한 파일 공유, 웹디스크를 이용한 파일 공유 등이 사용되고 있다. 하지만 일반적인 인터넷 응용들과 다른 인터넷 웜, 바이러스, 특정 네트워크나 호스트를 공격하는 트래픽과 같은 부정적인 사용 또한 증가하고 있다.

일반적인 인터넷 트래픽 모니터링 도구들은 TCP/UDP 포트 번호를 이용하여 응용 트래픽을 분류한다. 하지만 네트워크 관리를 위한 트래픽 분석에 있어서 헤더 정보만을 이용하여 분류를 수행하는 것은 응용 트래픽을 정확하게 구분하기 어렵다. 예를 들면, 파일구리와 같은 응용의 기본 포트 번호는 9493 을 사용하는데 그 포트 번호를 변경하여 사용하는 경우 분류하는 것이 불가능하다. 그리고 80 번 포트 번호로 통신하는 패킷은 일반적으로 HTTP 를 이용하는 웹 응용으로 분류된다. 그러나 최근의 P2P 응용은 방화벽을 통과하기 위하여 80 번 포트 번호로 통신이 가능하다. 포트 번호나 IP 등의 헤더 정보로 응용 트래픽을 분류하는 방법보다 시그니처 기반의 분석 방법은 패킷의 페이로드 내용을 검사하기 때문에 보다 정확성이 높을 수 있다. 하지만 시그니처 기반의 분석 방법의 단점은 실시간 분석일 경우 시스템의 부하가 커지고 scalability 의 저하가 증대 된다.

본 연구에서는 모든 시그니처를 linked list 로 연결시키는 방법과 해쉬 테이블과 캐시를 이용하는 방법을 제안하여 각각의 실행 시간을 비교해본다. 키 값은 각 service code 를 이용하여 해쉬 테이블을 구현하고 각 service code 에 해당하는 시그니처들을 그 service code 에 linked list 로 연결시킨다. 즉, 하나의 서비스에는 그에 해당하는 시그니처들만이 linked list 로 연결된다. 여기에서 service code 란 각 어플리케이션들에게 주어진 고유한 번호이다.

즉, 패킷들을 검사할 경우에 하나의 패킷이 어떤 특정한 서비스의 패킷이라고 한다면 그 다음의 패킷도 이전 패킷과 동일할 수 있다고 가정할 수 있다.

본 연구의 핵심은 캐시라는 5 개의 배열을 생성한 후 매칭된 시그니처의 service code 를 캐시에 넣고 다음 패킷을 검사할 때에는 캐시 안의 service code 의 시그니처만 검사하도록 패킷의 페이로드와 시그니처와의 비교 검색 시간을 최소화시키고 또한 시그니처 기반의 분석의 단점인 시스템 부하를 최소화 시키는

이 논문은 2007 년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2007-331-D00387)

데 있다.

본 논문은 다음과 같이 구성된다. 2 장에서는 본 연구와 관련된 연구를 조사하고 3 장에서는 본 연구에서 구현한 시그니처 기반 분석 시스템에 대한 구조와 설계 및 구현에 대해 설명한다. 4 장에서는 두 가지의 실험 결과에 대해서 설명하고 5 장에서 결론을 맺는다.

2. 관련 연구

대부분 트래픽 차단 시스템, 그리고 새롭게 제안되고 있는 트래픽 모니터링 툴에는 시그니처에 기반을 둔 트래픽 감지 및 분류기능이 포함되어 있다. 시그니처는 패킷의 페이로드에서 추출된 패턴을 의미하는 것으로, 특정한 응용에 의해 생성된 트래픽 플로우나 패킷을 해당 응용에 속한 것으로 판정하기에 충분할 만큼 빈번히 관측되어야 한다.

페이로드 기반의 시그니처는 IDS/IPS 와 같은 내용 기반 트래픽 식별/차단 시스템에 의해 주로 사용된다. 여기서 IDS 란 침입탐지시스템으로 방화벽과 함께 활용되는 네트워크 보안 solution 을 말한다. [1,5]에 제안된 방법은 페이로드 시그니처를 사용하여 인터넷 트래픽을 어떻게 분류할 수 있는지를 보여주는 좋은 사례이다.

[2]에서는 'graphlet' 이라는 이름의 통신 패턴 시그니처를 제안하고 있다. graphlet 을 사용하면 트래픽을 분류하는 작업이 패킷의 페이로드를 검사하지도 않고도 이루어질 수 있다는 장점이 있지만 새로운 graphlet 을 만들어 내는 비용이 페이로드 기반 시그니처에 비해 낮다고 볼 수 없기 때문에 통신 패턴만을 고려할 경우 완벽한 트래픽 분류가 이루어 질 수 없는 경우가 있어 graphlet 과 같은 형태의 시그니처는 인터넷 트래픽에 온라인으로 적용하기에 적합하지 않다.

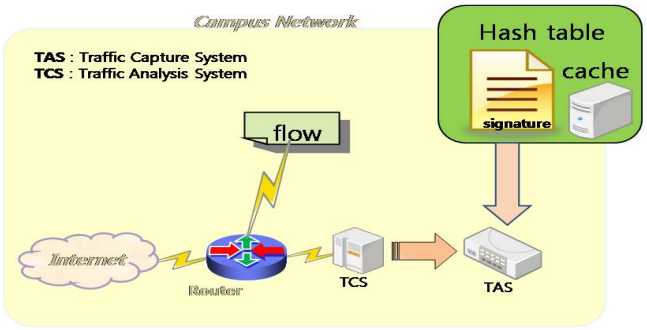
인터넷 트래픽을 분류하는데 여러 가지의 방법들이 있다. 그 중 페이로드 기반의 시그니처 해싱에 기반한 고성능 침입 방지 시스템에서도 패킷의 페이로드와 시그니처의 패턴 매칭의 수행 성능을 향상시키기 위해서 필요한 가장 중요한 부분은 들어온 패킷과 시그니처와 비교해야 하는 룰의 개수를 줄이거나 경우의 수를 줄임으로써 매칭 수행 성능을 향상시킬 수 있다.

하지만 단순히 분류를 잘 해서 매칭을 수행해야 하는 시그니처의 수를 줄이는 것은 시그니처의 개수가 점점 많아지고 있는 현실을 감안하면 그다지 효과적이라고 할 수 없는 방법이다. 이를 보장하기 위해서는 항상 예측 가능하고, 성능이 보장되는 새로운 알고리즘이 필요하다.[4]

따라서 본 논문에서는 해쉬테이블과 캐시를 이용한 시그니처 기반 시스템을 구현하여 보다 효율적인 알고리즘을 제안한다.

3. 시그니처 기반 분석 알고리즘

3.1 시그니처 기반 트래픽 분류 시스템



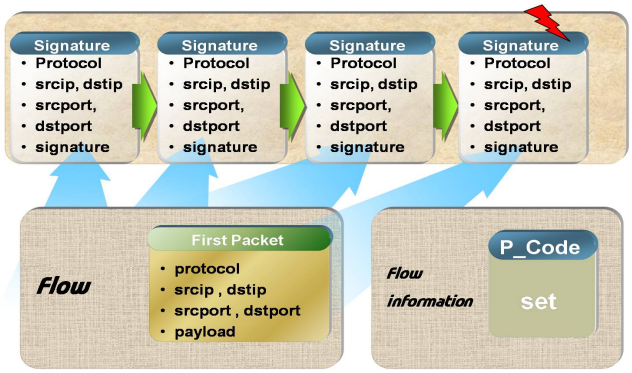
(그림 1) 시그니처 기반 트래픽 분류 시스템

그림 1 은 본 논문에서 사용하는 시그니처 기반 트래픽 분류 시스템의 구성을 나타낸 것이다. 시그니처를 기반으로 인터넷 트래픽을 분류하는 것은 실제 패킷의 페이로드 부분과 시그니처로 정의된 일정한 패턴으로 비교 검사하여 매칭을 시키므로 헤더 정보로 트래픽을 분류하는 것보다는 월등히 정확성이 높다.

그림 1 에서 볼 수 있듯이 TCS(Traffic Capture System)에서 실제 캠퍼스 네트워크 망에서 발생하는 인터넷 트래픽들을 수집하고 그것들을 해쉬 테이블과 캐시로 구현되어 있는 시그니처 시스템과의 매칭을 통하여 시그니처를 분류하게 된다. Traffic Analysis System (TAS)은 실제 패킷과 시그니처들을 분류하는 역할을 한다.

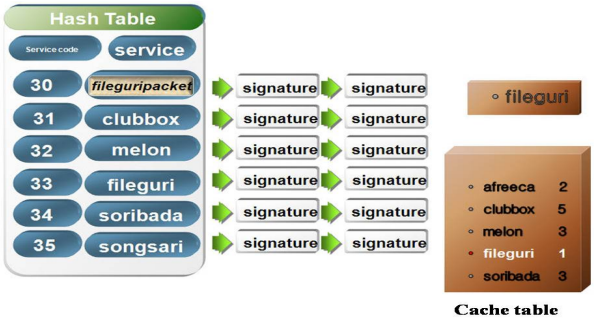
시그니처를 기반으로 인터넷 트래픽을 분류하는 시스템 중 본 연구에서는 탐색 시간을 최소화할 수 있는 해쉬 테이블과 캐시로 구현하였다.기존의 linked list 로 구현되어 있는 시그니처 기반 분류 시스템보다는 매칭이 될 때까지의 탐색 시간을 최소화하고 또한 그에 따른 시스템의 부하도 줄일 수 있다.

3.2 linked list 구조와 해쉬 테이블의 구조 비교



(그림 2) linked list 의 시그니처 구조

그림 2 는 시그니처의 자료구조를 linked list 의 구조로 나타낸 것이다. 그림 2 에서 보듯이 첫 번째 flow 의 첫 번째 패킷을 메모리 상에 서비스와 프로세스의 구분 없이 linked list 로 연결된 모든 시그니처들과 검사하여 매칭이 되면 프로세스 코드가 삽입된다. 최악의 경우 매칭이 되는 시그니처가 맨 마지막에 위치해 있다면 첫 시그니처부터 마지막 시그니처까지 탐색하여야 하므로 그 탐색 시간은 매우 길고 비효율적이다.



(그림 3) 해쉬 테이블 의 시그니처 구조

그림 3 은 시그니처의 자료구조를 해쉬 테이블로 구현한 것이다. 그림 3 의 해쉬테이블을 보면 키 값은 서비스 코드 번호로 지정하고 각 각의 서비스 코드에 대한 시그니처들을 linked list 로 연결한다. 또한 5 개의 서비스 코드를 저장할 수 있는 캐시가 있기 때문에 어떤 하나의 패킷이 어느 하나의 응용으로 분류가 되면 그 서비스 코드를 캐시에 저장한다. 그 이후에 패킷은 전체 시그니처를 검색하는 것이 아니라 캐시에 저장되어 있는 서비스 코드에 대한 서비스의 시그니처들을 먼저 매칭될 확률을 향상시킴으로써 탐색 시간을 현저히 줄일 수 있다. 이것은 하나의 flow 가 발생하면 같은 서비스의 flow 가 연속적으로 발생한다는 가정을 둔 것이다. 그 정책의 수단으로 캐시 안의 들어가는 서비스의 코드는 hit_count 수를 가지게 되어 매칭이 될 때마다 하나씩 증가하게 된다.

3.2 설계

시그니처 기반의 해쉬 테이블과 캐시를 구현하기 위해 본 논문에서는 그림 4 와 같은 알고리즘을 제안한다. 먼저 가장 첫 번째 패킷이 들어오면 가장 먼저 확인해야 할 것은 캐시가 비어있는지 .

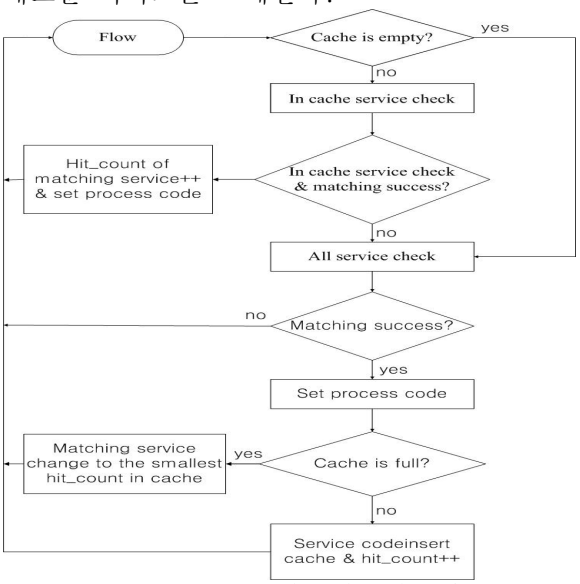
캐시가 비어있을 경우에는 패킷이 모든 서비스의 시그니처들을 검사하여야 한다. 만약 매칭이 되었다면 매칭된 서비스의 코드를 캐시에 삽입하고 hit_count 를 하나 증가시킨다. 하지만 캐시가 비어있지 않을 경우에는 두 가지 경우로 나누어 볼 수 있다.

즉 캐시에 하나 이상 네 개 이하의 서비스 코드가 들어있거나 캐시가 꽉 차있는 경우이다.

먼저 캐시에 하나 이상 네 개 이하로 차있는 경우에 해당 패킷에 대하여 캐시에 있는 서비스 코드의 서비스의 시그니처부터 우선적으로 검사한다. 만약 캐시에 존재하는 서비스 코드의 서비스의 시그니처로 매칭되었다면 캐시 안의 매칭된 서비스 코드의 hit_count 를 하나 증가시키고 프로세스 코드를 삽입한다.

반대로 캐시 안의 서비스 코드의 시그니처로 매칭이 되지 않았다면 다시 모든 서비스의 시그니처를 검사한다. 만약 매칭이 되었다면 캐시에 삽입될 공간이 있는지 없는지 다시 캐시를 검사한 후 삽입될 공간이 있으면 그 서비스 코드를 캐시에 삽입하고 hit_count 를 증가시킨다.

두 번째로 캐시가 꽉 차있을 경우에는 캐시 안에 들어있는 각 서비스의 hit_count 가 가장 작은 서비스와 새로운 서비스를 교체한다.



(그림 4) 알고리즘의순서도

3.3 구현

3.2 의 설계를 바탕으로 시그니처 기반의 알고리즘을 캐시를 이용하여 구현하였다. 구현 방식을 시그니처를 해쉬 테이블과 리스트 방식으로 검색할 수 있다.

<표 1>캐시를 이용한 시그니처 생성 시스템 개발 환경

OS	Linux Fedora 6
Language	C,C++
CPU	Core2 3.00GHz
Memory	2GByte

본 논문에서 개발한 해쉬 테이블과 캐시를 이용한 시그니처 기반의 시스템은 표 1 과 같은 개발환경에서 개발되었다.

4. 실험

4.1 실험 데이터

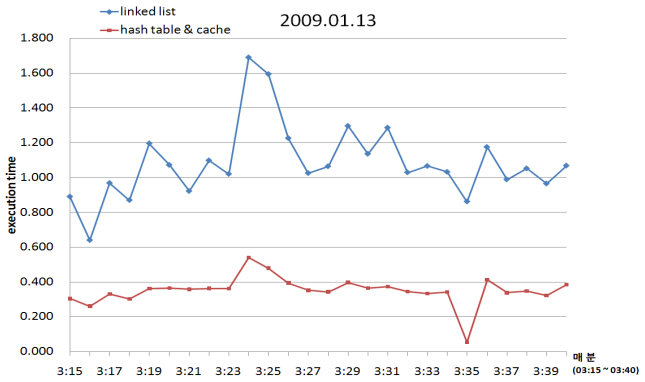
본 연구에서 사용하는 플로우 데이터 셋은 실제로 학내 망과 인터넷을 연결하는 백본에서 실시간으로 수집한 데이터들이다. 단 몇 가지 조건으로 같은 시그니처의 개수(675 개)와 같은 시간으로 모든 시그니처들을 통일시켰고 프로그램 상의 출력 문들은 모두 주석 처리를 하였다. 실제 데이터들을 실시간으로 두 시스템의 시그니처들과 비교시켰고 그에 따른 실행 시간을 비교하였다.

4.2 시그니처 기반 분석 결과 비교

여기서의 응용은 p2p, 웹디스크, 스트리밍 서비스 등의 응용프로그램을 나타낸다.

<표 2> input data information

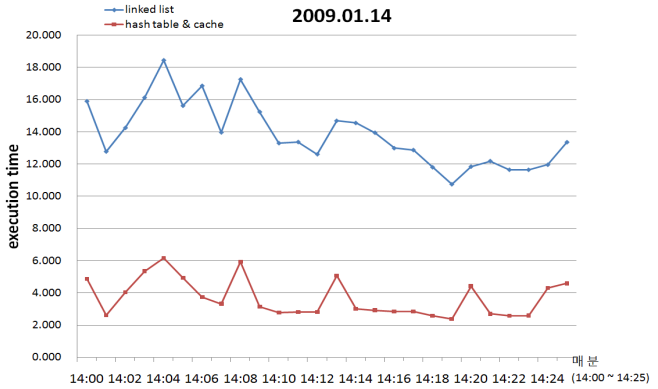
Application	Process	Signature
82 개	156 개	675 개



(그림 5) 실행 시간의 결과(오전)

그림 5 는 실험의 조건이 같다는 전제 하에 비록 트래픽이 적은 시간대에 2009 년 1 월 13 일 03 시 15 분부터 03 시 40 분까지 25 분 간 실시간으로 실제 패킷과 시그니처들의 매칭되는 시간을 나타낸 그래프이다.

그림 5 의 결과를 보고 알 수 있듯이 캐시를 이용하여 실제 패킷들이 시그니처와 매칭되는데 걸리는 시간은 linked list 의 구조보다 확연히 빠르다는 것을 알 수 있다. 그림 5 의 실험은 같은 조건 하에 x 축은 매 분을 나타내고 y 축은 그 시간에 걸린 실행 시간을 나타낸다.



(그림 6) 실행 시간의 결과(오후)

그림 6 는 2009 년 1 월 14 일 오후 14 시부터 14 시 5 분까지 25 분 간 측정한 결과이다. 위의 그래프는 linked list 이기 때문에 모든 시그니처를 검사해야 하기 때문에 그래프가 들쑥날쑥 하지만, 아래의 캐시를 이용한 그래프는 거의 대부분에서 일정한 수평선으로 나타나는 것을 볼 수 있다. 그 이유는 캐시 안의 서비스들이 hit 가 많이 나기 때문이다. 즉, 같은 시간에 패킷이 연속적으로 나타난다는 것을 나타낸다. 하나의 패킷이 나타나면 그 다음의 패킷도 같은 플로우의 패킷일 가능성이 높다는 결과를 얻어냈다. 또한 급격히 변하는 이유는 캐시 안의 서비스의 시그니처와 매칭이 되지 않아서 모든 시그니처를 검사하기 때문에 시간이 오래 걸렸다고 볼 수 있다.

5. 결론

본 연구에서는 시그니처 기반의 트래픽을 분류하는 데에 캐시를 사용하여 효율을 높이는 새로운 방법을 제안하였다. 앞서 말했듯이 시그니처 기반의 트래픽 분류 방법은 TCP/IP 헤더 필드들을 이용한 분류 방법보다 정확성이 높지만 시스템의 부하가 크다. 그러한 단점을 보완하기 위해 패킷과 시그니처의 검색 시간을 줄임으로써 더 빠르게 분류가 가능하게 하였다.

먼저 본 논문에서 제안한 시그니처의 해쉬테이블과 캐시를 이용한 시그니처 기반 분류 시스템의 효율성을 알아보았고 일반적인 시그니처의 자료구조 보다 본 논문에서 제안한 알고리즘이 효율적이라는 사실도 실험을 통해 알아보았다. 또한 다음에 발생할 플로우도 같은 응용의 플로우일 것이라는 처음 가정한 가설도 확인되었다.

즉, 같은 시간대에는 일정한 응용들이 사용된다는 것을 알 수 있다. 시그니처 기반의 분석 시스템에서 캐시를 이용하는 방법은 현재 응용의 다양성과 방대한 트래픽의 양이 많아질수록 효율적이라고 할 수 있다.

패킷을 깊은 수준에서 분석하고 확인하는데 있어 매칭 알고리즘은 네트워크 성능이 증가하고 그 처리 속도가 중요시되는 현실에 비추어 볼 때 매우 중요한 부분임에 분명하다.

이러한 새로운 기법을 이용하면 기존에 비해 훨씬 효율적이고 안정적인 네트워크 보안 구축이 가능할 것이다. 본 연구에서 제안한 매칭 알고리즘은 탐색 시간을 확연히 줄일 수 있지만, hit_count 를 이용하여 캐시가 꽉 차있는 상태에서 교체 정책을 여러 가지 방법으로 실험하여 더 효율적인 정책이 무엇인지에 대하여 연구해야 할 것이고, 또한 캐시에 과거 패턴을 이용한다든지 또한 캐시에 시간 정보를 추가하여 더욱 더 효율적인 알고리즘으로 발전시킬 수 있다.

또한 캐시에 저장되는 최적의 응용의 수를 변경하고 실험하여 캐시에 저장되는 최적의 응용의 수를 찾는 것도 앞으로 해야 할 과제이다.

또한 p2p 와 web disk 는 플로우의 발생량이 다르기 때문에 응용의 특성(발생되는 플로우의 양)에 따라 캐시 내의 응용 변경 정책을 달리 적용하는 것도 향후 과제이다.

참고 문헌

[1] S. Sen, O. Spatscheck, and D. Wang, "Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures," WWW, 2004.

[2] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel Traffic Classification in the Dark, " ACM SIGCOMM, 2005.

[3] Jeongs eok WangO Huiung Kwon Yun jae Jung Hu keun Kwak Kyusik Chung “A High Performance IPS Based on Signature Hashing” School of Electronics Engineering, Soongsil University.

[4] A.Silberschartz and P. Galvin, "Operating System Concepts," Addison Wesley, 1997.

[5] T. Choi, S. Yoon, H. Chung, J. Park, B. Lee, S.Yoon, and T. Jeong, "Flow-based Applicationaware Internet Traffic Monitoring and Field Trial Experiences," APNOMS,2005