서버 캐쉬 기반 트래픽 분류 방법에 관한 연구

박준상, 박태영, 윤성호, 김명섭

고려대학 컴퓨터정보학과

{ junsang_park, no14on, sungho_yoon, tmskim}korea.ac.kr

Study on Server Cache-based Traffic Classification Method

요 약

응용 레벨 트래픽 분류는 안정적인 네트워크 운영과 자원 관리를 위해서 필수적으로 요구된다. 트래픽 분류 분야에 있어서 페이로드 시그니쳐 기반 응용 레벨 트래픽 분류 방법은 고속 링크의 트래픽을 실시간으로 처리하는 과정에서 헤더 정보 및 통계 정보 이용 방법론에 비해 상대적으로 높은 부하를 발생시키며 처리 속도가 느린 단점을 갖는다. 본 논문에서는 페이로드 시그니쳐 기반 트래픽 분류 시스템의 처리 속도 향상 위하여 응용의 서버 IP, Port 정보를 캐쉬에 등록하고 분석하는 방법을 제안한다. 또한 분석 시스템의 처리 속도와 분석률을 향상시킬 수 있는 캐쉬 관리 기법을 제안한다. 제안하는 방법을 학내 망의 실제 트래픽 분석에 적용하여 페이로드 시그니쳐 기반 분석 방법 대비 최대 10 배 이상 처리 속도를 감소시킬 수 있었고, 플로우 분석률을 10% 이상 향상시킬 수 있었다.

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 확장 계획 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. 이를 위해서는 다양한 종류의 응용 레벨 트래픽을 정확하게 분류할 수 있는 방법과 고속 링크에서 발생하는 대용량의 트래픽을 실시간으로 처리하는 방법이 요구된다.

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, 3,4,8,9] 하지만 페이로드 시그니쳐 기반 분석 방법은 분류 시스템의 처리 속도에 있어 현재의 고속 네트워크 상에서 발생하는 대용량 트래픽을 실시간으로 처리하기에 부적합한 방법이다. 응용의 수와 대용량의 트래픽을 발생시키는 응용의 사용이 증가하고 있는 추세를 고려했을 때 페이로드 기반 분석 방법의 처리 속도 문제는 반드시 해결되어야 하는 과제이다. 이를 해결하기 위해 기존의 다양한 연구에서는 패턴 매칭 알고리즘의 성능 개선 기법에 대한 연구가 주를 이룬다[6,7,10,11] 하지만 매칭 알고리즘의 성능 개선은 제한적이며 현재의 고속 링크의 대용량 트래픽을 수용할 수 없는 것이 현실이다.

본 논문에서 응용 트래픽의 발생 특징을 분석 시스템에 반영하여 트래픽 분류 시스템의 성능을 향상 시킬 수 있는 방법을 제안한다. 분석 대상 네트워크에서 발생하는 응용의 종류는 다양하지만 트래픽의 발생량 측면에서 소수의 응용에 의해서 대부분의 트래픽이 발생한다. 또한 특정 응용에서 접속하는 서버 IP, Port 는 제한적이다. 학내망에서 발생하는 트래픽을 대상으로 조사한 결과, 전체 TCP 플로우의 80%가 10,000 개 이하의 서버 IP, Port 로 접속하는 것을 확인할 수 있었다. 이러한 현상을 본 논문에서는 응용 트래픽의 지역성으로 정의하고, 이를 이용하여 페이로드 시그니쳐 기반 분류 시스템의 처리 속도를 향상 시킬 수 있는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 본 장의 서론에 이어, 2 장에서는 관련연구에 대해 기술하고, 3 장에서는 제안하는 방법의 배경이 되는 응용 트래픽의 지역성에 대해 설명한다. 4 장에서는 실험 결과를 바탕으로 제안하는 방법을 기술한다. 5 장에서는 제안하는 방법을 분류 시스템에 적용하여 그 타당성을 증명한다. 마지막으로 6 장에서는 결론 및 향후 연구에 대해 기술한다.

* 이 논문은 정부(교육과학기술부)의 재원으로 2010 년도 한국연구재단-차세대정보컴퓨팅기술개발사업(20100020728) 및 2012 년도 한국연구재단(2012R1A1A2007483)의 지원을 받아 수행된 연구임.

2. 관련 연구

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

응용 프로그램 트래픽 분류를 위한 도구로 많이 사용되고 있는 L7-filter 는 시그니쳐를 정규표현식으로 표현하고 패턴 매칭 알고리즘으로 NFA(Nondeterministic Finite Automata)를 적용한다. 하지만 70 여 개의 시그니쳐를 적용하였을 때 3.5Mbps 이하의 처리 속도를 보인다.[6] NFA 의 처리 속도를 향상 시키기 위해 DFA(Deterministic Finite Automaton) 기반의 분석이 제안되고 활용되고 있지만 100Mbps 이하의 처리속도를 갖는다.[6,7,10]

Christopher L. Hayes 와 Yan Luo 는 트래픽 분류 시스템의 처리 속도 향상을 위해 패턴 매칭 알고리즘의 성능 향상을 위한 방법을 제안하지만 매칭 알고리즘의 성능은 입력 데이터의 구성에 의존적이며, 제한적인 성능 향상을 나타낸다[7]. 패턴 매칭 알고리즘으로 오토마타에 기반한 NFA 와 DFA 알고리즘의 성능 향상을 위한 방법론들이 제시되고 있지만 오토마타를 이용한 방법은 '.*'와 같은 와일드 카드의 사용 빈도에 따라 시간 및 공간 복잡도 급격하게 증가하여 성능이 저하되는 문제점이 있다[10,11].

Abhishek Mitra 외[12]는 NFA 에 기반한 패턴 매칭 알고리즘을 FPGA 로 구현한 하드웨어 기반 방법론을 제시하였다. 하지만 하드웨어 기반 분석 방법은 고가의 비용이 요구되며, 현재와 같이 응용 프로그램의 출현, 소멸, 갱신 등 변화가 잦은 환경에는 적합하지 않은 방법이다.

응용 레벨 트래픽 분류 방법으로 포트 기반 트래픽 분석 기법이 많이 활용되었다[5]. 포트 기반 트래픽 분석 방법은 다른 분석 방법에 비해 분석 속도가 빠르고 높은 정확도를 보장하는 방법이다. 하지만 동적 포트를 사용하는 응용의 증가와 알려지지 않은 포트의 사용으로 분류 정확도를 신뢰할 수 없는 문제점이 발생하여 그 활용도 감소되었다. 동적 포트를 사용하는 응용의 트래픽을 페이로드 시그니쳐 기반으로 분석하여 그 정보를 분석 결과에 활용한다면 포트 기반 분석의 장점을 유지하면서 페이로드 기반 분석 방법의 단점인 처리 속도 문제를 보완할 수 있다. 기존 연구는 페이로드 시그니쳐 기반 트래픽 분류 시스템의 처리 속도 향상을 위해서 패턴 매칭 기법을 소프트웨어 또는 하드웨어적으로 개선하려는 노력이 주를 이루었다. 하지만 이러한 방법은 네트워크 대역폭 증가에 비해 상대적으로 제한적인 성능 향상을 보인다.

3. 응용 트래픽의 지역성

본 장에서는 제안하는 방법론의 배경이 되는 응용 트래픽의 지역성을 트래픽의 발생 패턴을 이용하여 정의하고, 학내망의 실제 트래픽을 통해서 트래픽의 지역성을 실험적으로 확인한다.

3.1 트래픽 트레이스 구성

본 절에서는 응용 트래픽의 지역성을 확인하고, 제안하는 방법의 타당성을 증명하기 위해 사용된 트래픽의 구성에 대하여 기술한다.

표 1 은 실험에 사용된 트래픽 트래이스와 실험 기간을 보여주고 있다. 학내 망과 인터넷의 연결 지점에서 하루동안 3,000 여대의 호스트에서 발생한 트래픽을 플로우와 페이로드 데이터를 포함한 패킷 형태로 수집하였다.

표 1. 트래픽 트레이스

measure	Flows	Packets	Bytes
Volume	51,477K	2,012M	1,578GB
Duration	2012.09.12 00:00 ~23:59		

표 2 는 본 연구진에서 수행한 선행 연구[1]를 통해 추출한 페이로드 시그니쳐를 기반으로 분류한 결과에 대해 응용의 유형 별로 Top 10 의 트래픽양을 보여주고 있다.

학내 망의 전체 인터넷 트래픽에 대해 실시간으로 적용한 결과 flow/byte/packet 단위로 99%이상의 정확도와 85% 이상의 분석률을 보였다.

플로우를 기준으로 P2P 파일공유 응용들이 50% 이상을 차지하고 있고, 웹 브라우져 트래픽이 다음으로 많은 양의 트래픽을 발생하고 있는 것을 알 수 있다.

이와 같이 분석 대상 네트워크에서 발생하는 트래픽은 소수의 응용에 의해서 대부분의 트래픽이 발생한다. 이러한 응용에 대한 분석 시간을 단축 시킬 수 있다면 분류 시스템의 처리 속도를 향상 시킬 수 있다.

표 2. Top 10 응용 트래픽 발생량

Top 10	App. Type	Flows	Packets	Bytes
1	p2pfilesharing	25,395K	561,709K	356GB
2	web browser	17,043K	1,037,368K	864GB
3	im	1,086K	3,0381K	16GB
4	utility	973K	4,1905K	32GB
5	multimedia	759K	20,5534K	172GB
6	game	691K	31,088K	18GB
7	sns	449K	9820K	6GB
8	security	248K	6,283K	4GB
9	vaccine	230K	28,970K	28GB
10	commercial	126K	22,358K	20GB

3.2 서버 IP, Port 의 분포

그림 1 은 표 1 의 트래픽에 대해 서버/클라이언트를 결정할 수 있는 TCP 플로우의 개수와 TCP 플로우를 서버의 IP, Port, L4 프로토콜로 그룹하였을 때 그룹의 개수를 나타내고 있다. 서버의 IP, Port, L4 프로토콜이 동일한 플로우 그룹을 SSIP(Same Server IP Port)라고 명명하도록 하겠다.

전체 TCP 플로우는 최소 45.79%, 최대 16.34%, 평균 29.59%의 동일한 서버 IP, Port 로 접속하는 것을 확인할 수 있다. 이와 같이 특정 분석 대상 망에서 발생하는 응용 트래픽은 접속하는 서버에 대한 지역성을 갖는다.

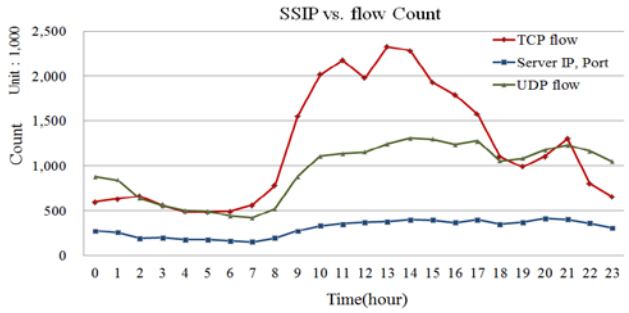

그림 1. SSIP vs. Flow

그림 2 는 TCP 플로우에 대하여 서버 IP, Port 별로 플로우 개수에 대한 PDF 를 보여주고 있다.

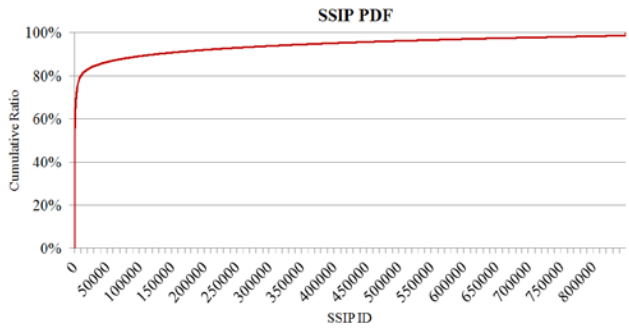

그림 2. SSIP PDF

그림에서 알 수 있듯이 TCP 전체 플로우의 80%가 10,000 개 이하의 SSIP 로 접속하는 것을 알 수 있다. 서버의 IP, Port 는 응용 프로그램의 특정 서비스를 연결하는 주소로 사용된다. 따라서 10,000 개 이하의 서버 IP, Port 가 제공하는 응용의 분류 결과를 분석 시스템에서 유지할 수 있으면 동일한 서버 IP, Port 로 접속하는 플로우를 페이로드 시그니쳐 기반 분석없이 식별할 수 있게 된다.

4. 제안하는 방법

본 장에서는 논문에서 제안하는 SSIP 캐쉬 기반 페이로드 시그니쳐 분석 방법론을 기술한다.

그림 3 은 제안하는 트래픽 분류 방법론의 개념도를 보여준다. 클라이언트 A 와 B 가 같은 응용을 사용하여 동일한 서버의 IP 에 동일한 Port 로 접속한다. 이때 클라이언트 A 에서 발생한 플로우가 페이로드 시그니쳐 기반으로 분석되면, 클라이언트 B 에서 발생한 플로우는 페이로드 시그니쳐로 분석하지 않고 Server IP, Port 만을 비교하여 분석할 수 있다.

페이로드 시그니쳐 기반 분석기에 의해서 특정 플로우가 분석되면 해당 플로우의 서버 IP, Port 정보가 SSIP 캐쉬 테이블에 업데이트된다. SSIP 캐쉬에 정보가 저장되어 있으면 트래픽 분류 시스템은 SSIP 캐쉬를 통해 플로우를 분석하고, 분석되지 않은 플로우에 대해서 페이로드 시그니쳐 기반 분석을 수행한다.

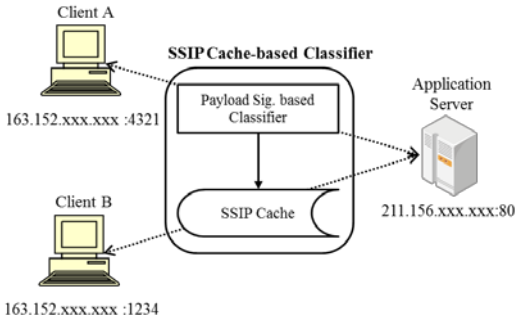

그림 3. 제안하는 방법의 개념도

4.1 SSIP 캐쉬 기반 트래픽 분류 방법

그림 4 는 제안하는 SSIP 캐쉬 기반 트래픽 분석 방법론의 흐름도를 나타내고 있다.

분류 시스템은 페이로드 시그니쳐와 분석 대상 트래픽을 입력으로 받아 최종적으로 응용이 이름이 식별된 트래픽 데이터를 결과로 제공한다.

분류 시스템은 크게 시그니쳐를 매칭하는 부분과 SSIP 캐쉬를 관리하는 부분으로 구성된다.

시그니쳐를 매칭하는 부분은 SSIP 캐쉬 기반 매칭 모듈과 페이로드 시그니쳐 기반 매칭 모듈로 구성된다. 시그니쳐 매칭 모듈은 SSIP 캐쉬를 선 매칭하고, 분류되지 않은 트래픽에 대해서는 페이로드 시그니쳐 기반 매칭을 수행한다. SSIP 캐쉬를 관리는 부분은 SSIP 의 삽입, 갱신, 삭제, 재배열의 기능을 통해서 분석 시간을 최소화하고, 분석률을 향상시킬 수 있다.

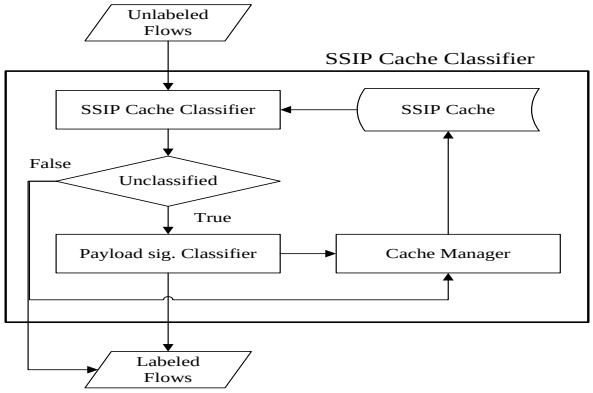

그림 4. 분류 시스템 흐름도

4.2 SSIP 캐쉬 메모리 구조

트래픽 분류 시스템은 매 분단위로 실시간으로 운영되기 때문에 SSIP 캐쉬에는 대량의 정보가 누적된다. 대량의 SSIP 정보가 캐쉬에 누적 저장되면 캐쉬 기반의 트래픽 분류 속도가 오히려 감소되는 문제가 발생한다. 따라서 캐쉬 메모리를 효과적으로 관리할 수 있는 메모리 구조가 요구된다.

본 논문에서는 트래픽 분석을 위해 일반적으로 사용되는 싱글 해쉬 구조를 보완한 듀얼 해쉬 구조를 제안한다.

그림 5 는 싱글 해쉬와 듀얼 해쉬 구조를 도식화한 것이다. 싱글 해쉬는 대용량의 데이터 탐색 시 동일한 키를 갖는 노드의 개수가 증가하여 트래픽 분석의 부하로 작용한다. 이러한 문제를 해결하기 위해 해쉬 테이블의 크기를 증가시키는 방안을 적용하지만 분석 시스템에서 사용할 수 있는 메모리의 크기는 제한되어 있기 때문에 무한정 해쉬 테이블의 크기를 증가시키는 방법은 효과적인 해결책이 될 수 없다.

듀얼 해쉬 방식은 2 단계 해쉬 키 생성 방법을 이용하여 동일한 키 값을 갖는 노드의 개수를 최소화함으로써 탐색 공간을 감소 시키는 방법이다. 이 때 사용되는 해쉬 테이블의 크기는 학내망의 트래픽 양을 고려하여 2^3*2^{16} 으로 정의하였다. 1 단계 해쉬 함수는 서버 IP, Port, L4 Protocol 에 대한 마지막 8 비트의 합하여 3 비트만을 키로 사용한다. 2 단계 해쉬 키는 서버 IP 의 앞, 뒤를 Folding 방법으로 2byte 씩 더하고, 서버 port 와 L4 Protocol 의 2byte 를 더한 값에 0xFFFF 값과 AND 연산을 함으로써 키 값 (0 ~ 65,535)을 얻는다.

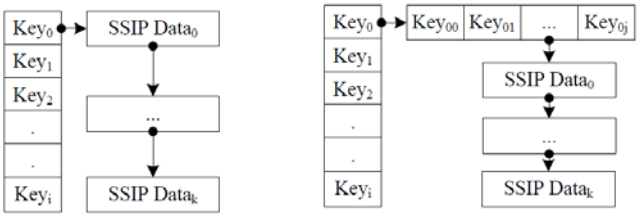

그림 5. 캐쉬 메모리 구조

그림 6 은 1 시간 수집한 동일한 트래픽에 대해서 싱글 해쉬와 듀얼 해쉬를 적용하여 해쉬 테이블에 저장된 모든 트래픽 데이터를 1 회씩 비교 검색할 때 소요되는 총 비교 횟수를 나타내고 있다.

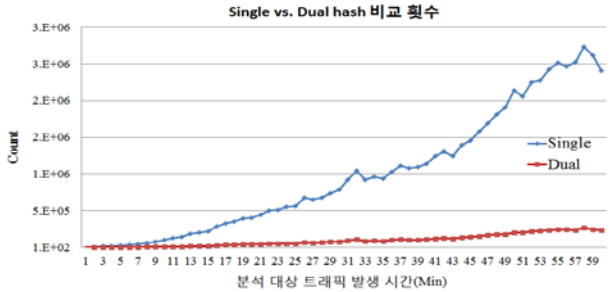

그림 6. Single 과 Dual 해쉬의 매칭 시도 횟수

듀얼 해쉬는 싱글 해쉬보다 평균적으로 10 배 이상의 비교 횟수를 감소 시킬 수 있었다.

4.3 SSIP 캐쉬 메니저

SSIP 캐쉬를 통해 처리 시간과 분석률을 향상 시킬 수 있지만 캐쉬에 영구적으로 정보를 저장하고 분석하면 탐색하는 정보의 양이 증가하여 처리 속도가 늦어지는 문제점이 발생한다. 또한 P2P 응용의 서버 호스트 정보를 캐쉬에 유지하면 분류 정확도가 감소되는 문제점이 발생한다. 따라서 캐쉬의 정보를 관리하는 정책이 반드시 요구된다.

그림 7 은 캐쉬 메니저의 기능을 도식화한 것이다. 캐쉬 메니저는 삽입, 갱신, 삭제, 재배열의 기능을 수행한다.

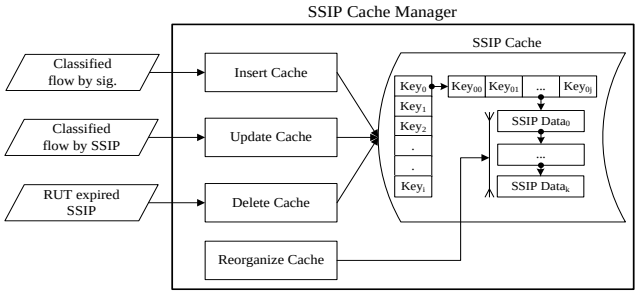

그림 7. 캐쉬 관리 구조

캐쉬 메니저의 삽입 기능은 페이로드 시그니쳐를 기반으로 분석된 플로우의 서버 IP, Port, 응용의 정보를 캐쉬에 저장하는 역할이다. 갱신은 SSIP 캐쉬가 마지막으로 분석에 사용된 시간과 분류에 사용된 횟수를 갱신하는 역할을 수행한다. 이 정보는 캐쉬의 정보를 제거하고, 메모리 구조를 재배열하는 기능에 사용된다. 캐쉬 데이터 삭제 모듈은 오래된 항목을 삭제하는 기능을 수행한다. 캐쉬 메모리 재배열 모듈은 캐쉬에 대량의 정보가 저장되면 캐쉬의 검색 속도가 감소되기 때문에 자주 사용되는 SSIP 정보를 우선 검사하여 검색 속도를 향상 시키기 위한 목적으로 사용된다.

4.3.1 캐쉬 데이터 제거

P2P 와 같은 응용은 다양의 플로우를 발생 시키며 Peer 의 서버 IP, Port 정보가 캐쉬에 저장되기 때문에 캐쉬 검색 양을 증가 시켜서 분석 속도를 저하 시키기 때문에 이와 같은 정보를 제거하기 위한 방안이 요구된다. 캐쉬 데이터 제거 방안으로는 캐쉬에 데이터가 등록되는 시간과 마지막으로 분류에 사용된 시간을 기준으로 적용하는 방법을 실험적으로 평가하였다.

그림 8 은 등록 시간 기준 방법과 최근 사용 시간 기준 방법을 적용하여 플로우 단위 분석률을 측정한 결과이다.

최근 사용 시간 기준 삭제 방법이 등록 시간 기준 삭제 방법 보다 더 높은 분석률을 나타내는 것

을 알 수 있다. 이는 등록 시간 기준 기반 삭제 방법은 최근 사용 유무를 반영하지 않고 캐쉬에서 제거하고, 페이로드 시그니쳐로 다시 분석되면 캐쉬에 다시 등록되기 때문에 분석률을 저하시키는 원인이 된다. 또한 캐쉬 데이터의 교체가 빈번하게 발생하기 때문에 최근 사용 시간을 기준으로 하는 캐쉬 데이터 제거 방안이 더 효과적이다.

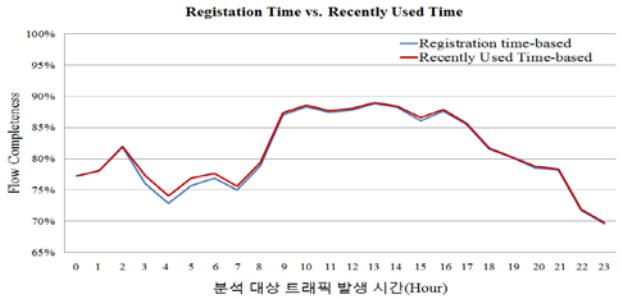

그림 8. RT vs. RUT 분석률 비교

캐쉬 기반의 분석 시스템의 분석 시간을 단축하기 위해서는 캐쉬에서 데이터를 유지하는 시간을 결정해야 한다. 캐쉬에 장시간을 유지하면 분석률을 향상 시킬 수 있지만 캐쉬에서 탐색하는 시간이 부하로 작용할 수 있기 때문이다.

그림 9 는 표 1 의 트래픽에 대해 순수하게 페이로드 시그니쳐 기반 분류 방법으로 분석한 분류 시간을 기준(100%)으로 하여 SSIP 캐쉬의 LT(life-time)를 1 시간씩 증가 시키면서 분석 시간의 비율을 측정한 결과이다. 이 때 LT 는 서버 IP, Port 정보가 SSIP 캐쉬에 유지되는 기간을 의미하며, 각각의 서버 IP, Port 별로 마지막으로 분류에 사용된 시각을 기준으로 업데이트된다. 따라서 분석에 지속적으로 사용되는 서버 IP, Port 정보는 SSIP 캐쉬에 유지되며, 그렇지 않은 경우에는 캐쉬에서 제거된다.

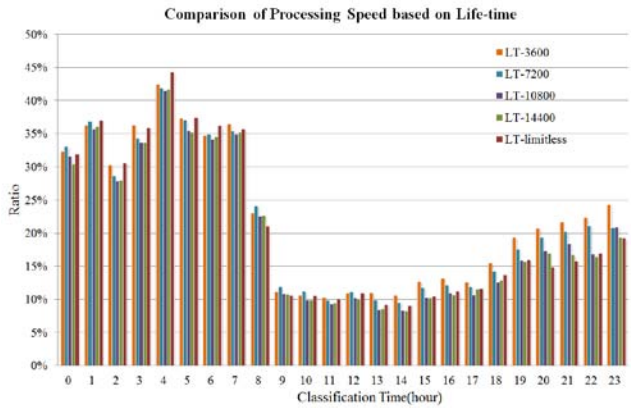

그림 8. RUT 의 변화에 따른 처리 시간 비교

서버 캐쉬의 LT 를 1 시간 단위로 증가시킨 결과, 3 시간까지 증가 시키면 캐쉬로 분석 가능한 플로우의 양이 증가하여 분석 시간이 감소된다. 반면에 LT 를 4 시간 이상으로 적용하면 분류 속도가 오히려 저하된다. 또한 LT 에 대한 제한이 없는 경우에도 분석 시간이 감소되는 것을 알 수 있다. 이는 SSIP 캐쉬에 저장되는 서버 IP, Port 정보의 양이 증가하여 분류 시스템의 처리 속도를 저하시키기 때문이다. 이러한 결과를 바탕으로 SSIP 캐쉬에 저장되는 정보는 최종적으로 분석에 사용된 시점부터 3 시간 동안 저장하는 정책으로 캐쉬를 관리한다.

4.3.2 캐쉬 데이터 재배치

4.2 절에서 듀얼 해쉬 기반 메모리 구조를 제안하여 해쉬에 동일한 키 값으로 저장되는 정보의 양을 감소 시켰지만 대용량의 트래픽에서는 이러한 충돌이 빈번하게 나타나기 때문에 이를 해결할 수 있는 방안이 요구된다.

본 논문에서는 이러한 동일한 키를 갖는 캐쉬 데이터를 분류에 사용된 빈도를 기준으로 메모리를 재배열하여 사용 빈도가 많은 정보를 우선 매칭하는 방법을 제안한다.

그림 10 은 1 시간 수집한 동일한 트래픽에 대해서 캐쉬의 키 값이 동일한 데이터를 주기적인 정렬하여 메모리를 재배열하는 기법을 적용한 것과 적용하지 않은 경우에 캐쉬 검색 시 소요되는 총 비교 횟수를 나타내고 있다.

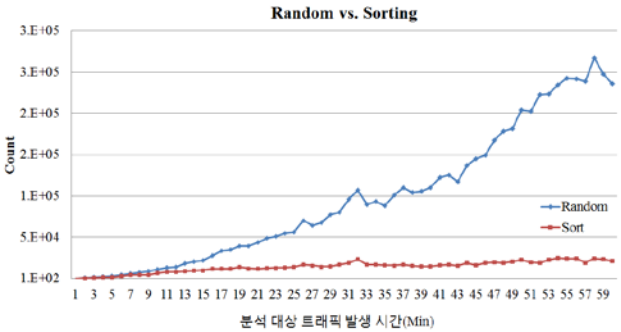

그림 10. 데이터 재배치에 따른 매칭 시도 횟수

정렬 기법을 적용하였을 최대 캐쉬에서의 매칭 시도 횟수가 13 배 이상 감소하는 것을 알 수 있다.

5. 성능 평가

본 장에서는 4 장에서 기술한 SSIP 캐쉬 기반 트래픽 분류 방법론을 적용하여 분류 시스템의 처리 속도와 분석률을 평가한다.

그림 11 은 페이로드 시그니쳐 기반 분류 방법과 SSIP 캐쉬 기반 분석 방법론의 분류 시간을 비교한 그래프이다. 제안하는 방법론은 페이로드 기반 분석 방법에 비해 최대 10 배 이상의 처리 속도가 향상되는 것을 알 수 있다. 페이로드 시그니쳐 기반 분석 방법은 트래픽의 발생량이 많은 09 시~21 시에 처리 시간이 급격하게 증가하는 반면에 제안하는 방법의 처리 시간은 크게 증가하지 않음을 알 수 있다.

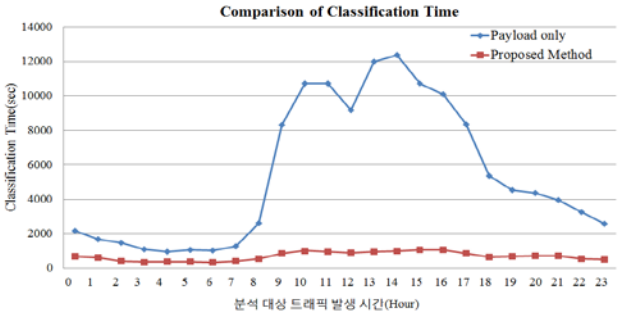

그림 11. 분석 시간 비교

표 3 은 페이로드 시그니쳐 기반 분석 방법과 제안하는 방법의 분석률을 보여주고 있다.

표 3. 분석률 비교

Method \ Measure		Flow		Packet		Byte	
		# of flow	Ratio	# of pkt	Ratio	# of Byte	Ratio
Total		51,477K	100.00%	2,012M	100.00%	1,578GB	100.00%
Payload	Classified	35,988K	69.91%	1,281M	63.69%	1,025GB	64.99%
	Unclassified	15,488K	30.09%	730M	36.31%	552GB	35.01%
Proposed Method	Classified by payload	22,032K	42.80%	686M	34.10%	583GB	36.98%
	Classified by SSIP Cache	19,553K	37.98%	662M	32.93%	466GB	29.58%
	Unclassified	9,891K	19.22%	663M	32.97%	527GB	33.43%

제안하는 방법은 페이로드 기반 분석 방법에 비해 10% 이상의 플로우를 추가적으로 분석하는 것을 알 수 있다. 이는 1 개의 서버 IP, Port 에서 제공하는 여러 가지의 응용 프로그램의 기능 중 페이로드 시그니쳐가 추출되지 않은 기능에 대해서도 서버 IP, Port 로 분석되기 때문이다.

6. 결론 및 향후 과제

본 논문에서는 페이로드 시그니쳐 기반 응용 레벨 트래픽 분류 시스템의 처리 속도 향상을 위해서 SSIP 캐쉬 기반 분석 방법론과 캐쉬 관리 기법을 제안하였다. 제안하는 분류 방법론은 페이로드 시그니쳐 기반 분석 방법과 비교해 최대 10 배 이상의 처리 속도를 향상 시킬 수 있었다. 또한 페이로드 시그니쳐로 분석하지 못한 트래픽을 추가적으로 분석하여 10% 이상의 플로우 분석률을 향상 시킬 수 있었다.

본 논문에서는 서버 Life-time 을 기반으로 모든 응용 트래픽에 대하여 일괄적인 캐쉬 관리 정책을 세웠다. 하지만 응용의 트래픽 발생 패턴은 다양하기 때문에 이를 고려한 관리 정책이 요구된다. 향후 연구로 응용의 트래픽 발생 패턴에 기반한 캐쉬 관리 방법에 대한 연구를 수행할 계획이다.

참고 문헌

[1] J. S. Park, J. W. Park, S. H. Yoon, Y. S. Oh, M. S. Kim, "Development of signature Generation system and Verification Network for Application Level Traffic Classification", in Proc. KIPS conf. Apr. 23-24, 2009, pp. 1288-1291, PuSan, Korea.

[2] S. H. Yoon, H. G. Roh, M. S. Kim, "Internet Application Traffic Classification using Traffic Measurement Agent ", in Proc. KICS Jul. 2-4, 2008, pp.618. Jeju Island, Korea.

[3] 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.

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

[5] 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.

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

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

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

[9] 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," Proc. of the Asia-Pacific Network Operations and Management Symposium (APNOMS) 2008, LNCS5297, Beijing, China, Oct. 22-24, 2008, pp. 474-477.

[10]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.

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

[12]Abhishek Mitra , Walid Najjar , Laxmi Bhuyan, Compiling PCRE to FPGA for accelerating SNORT IDS, Proceedings of the 3rd ACM/IEEE Symposium on Architecture for networking and communications systems, December 03-04, 2007, Orlando, Florida, USA