연속된 플로우 정보를 이용한 CDN 서비스 트래픽 분류 방법 연구
Study on CDN Traffic Classification using Cascade Flow Information

이수강 O, 심규석, 정우석, 김명섭

고려대학교 컴퓨터정보학과

{sukanglee O, kusuk007, hary5832, tmskim}@korea.ac.kr

요 약

인터넷 속도의 증가에 힘입어 인터넷을 이용하는 다양한 웹 서비스가 개발됨에 따라 이들이 발생시키는 인터넷 트래픽의 양이 급격히 증가하고 있다. 이러한 웹 서비스의 트래픽에서 제공하는 컨텐츠는 대부분 CDN(Contents Delivery Network)를 통해 전송된다. 기존 트래픽 분류 방법은 단일 플로우를 대상으로 트래픽을 분류하기 때문에 CDN 트래픽을 정확히 분류할 수 없다. 본 논문에서는 CDN 트래픽 분류의 한계를 해결하기 위해 연속된 플로우 연관관계를 이용하여 트래픽을 분류하는 방법을 제안한다. 제안하는 방법은 특정 서비스를 사용할 때 발생하는 연속된 플로우들 간 나타나는 연관관계를 추출하여 시그니쳐로 사용하는 것이다. 제안하는 방법을 이용하여 시그니쳐를 만들고, 실제 트래픽에 적용한 결과 기존 트래픽 분류 방법으로는 unknown 트래픽으로 분류되던 CDN 트래픽이 각 웹 서비스로 정확히 분류 할 수 있었다.

Keyword : Cascade flow information, Traffic association rule mining, CDN traffic classification

1. 서론

인터넷 속도의 증가에 힘입어 인터넷을 이용하는 다양한 웹 서비스가 개발됨에 따라 서비스를 이용하는 사용자와 이들이 발생시키는 인터넷 트래픽의 양이 급격히 증가하고 있다. 이에 따라 2000 년대 초반부터 급격히 늘어나는 트래픽에 대응하기 위한 트래픽 분산 기술이 연구되어 왔으며, CDN (Content Delivery Network)은 트래픽 분산 기술 중 가장 보편적으로 사용되고 있는 기술이다.

CDN[1]은 기존의 컨텐츠 전송 과정에서 빈번하게 발생하는 트래픽 집중 및 병목현상, 지연, 끊김과 같은 문제를 해결하기 위해 등장한 개념으로 사진, 동영상과 같은 대용량의 컨텐츠를 사용자 근처에 미리 옮겨놓고 그곳에서 콘텐츠를 사용자들에게 신속하게 배달하는 것이다. 국내에서는 KT, SK Broadband, LG U+와 같은 ISP 사업자들이 CDN 서비스를 제공하고 있다. 실제 국내 포털 사이트들이 제공하는 대부분의 컨텐츠는 컨텐츠를 제공하는 주체와 물리적으로 다른 곳에 위치한 CDN 서버에서 전송되고 있다.

CDN 트래픽은 그림 1 과 같이 발생한다. 사용자는 웹 포털에서 동영상과 같은 특정 컨텐츠를 요청하면 해당 포털에서 사용하는 CDN 업체의 서버에서 컨텐츠를 제공받는다. 이러한 상황에서 네트워크 관리자는 사용자가 컨텐츠를 요청할 때 발생하는 플로우(A)와 실제 컨텐츠가 사용자에게 제공될 때 발생하는 플로우(B)를 모두 분류할 수 있어야 한다. 기존의 트래픽 분류 방법으로는 플로우(A)가 어떤 포털인지 구분할 수 있지만 플로우(B)가 어떤 포털의 어떤 컨텐츠를 가지고 있는지를 정확하게 알 수 없기 때문에 특정 응용이나 서비스로 분류하기가 까다롭다. 따라서 플로우(A)와 플로우(B)가 같은 응용이나 서비스에 의해 발생한 트래픽이지만 두 플로우를 같은 응용이나 서비스로 분류할 수 없다.

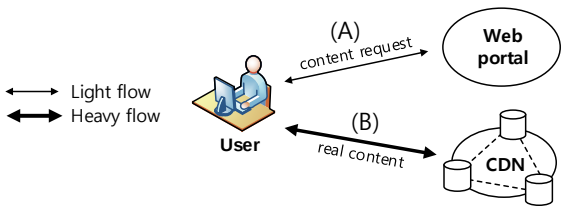

그림 1. CDN 트래픽 발생 예

본 논문에서는 연속되어 발생하는 플로우의 연관관계를 분석하여 CDN 트래픽이 어떤 컨텐츠 제공자에 의해 발생하는지를 분석하고, 이를 규칙화 하여 해당 컨텐츠 제공자가 제공하는 서비스와 함께 분류하는 방법을 제안한다.

본 논문은 다음과 같은 순서로 기술한다. 2 장에

이 논문은 2015 년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No.2015R1D1-A3A01018057).

서는 기존에 제시된 트래픽 분류 방법 설명하고 3 장에서는 CDN 트래픽 분류 방법을 제안한다. 4 장에서는 본 논문의 결론과 향후 연구를 제시한다.

2. 관련 연구

인터넷 트래픽 분류는 그 중요성이 증가함에 따라 다양한 분류방법이 제시되고 있다. 가장 원시적인 포트 기반 분석은 Internet Assigned Number Authority(IANA)[2] 에 등록된 포트 정보를 이용하여 트래픽을 분류한다. 초기 인터넷에서는 고정적인 포트 번호와 대응하는 서비스(HTTP(80), SSH(22), FTP(20, 21), e-mail(25,110))가 트래픽 대부분을 차지하였기 때문에 이를 기준으로 정확한 트래픽 분석이 가능하였다. 하지만 시간이 지남에 따라 인터넷을 사용하는 서비스가 다양해지고, 사용되는 포트 번호가 임의의 포트가 사용되므로 포트번호로는 정확한 트래픽 분류를 할 수 없게 된다. 포트 기반 분석의 한계점을 극복하기 위해 제시된 시그니쳐 기반의 인터넷 트래픽 분류 방법은 현재까지 보안, 응용 트래픽 분류 등의 다양한 분야에서 활용되고 있다.

시그니쳐 기반 트래픽 분류 방법은 패킷 또는 플로우의 특정 위치에서 응용을 식별하기 위한 고유한 정보를 추출하고 이를 기반으로 분류하는 방법이다. 시그니쳐는 시그니쳐 추출에 사용되는 정보에 따라 Header[3], Payload[4,5,6], Statistic[7,8], Behavior[9,10] 시그니쳐로 구분할 수 있다.

3. CDN 트래픽 분류 방법

지금까지 연구되었던 트래픽 분류 방법들은 단일 플로우를 대상으로 트래픽 분류를 하기 때문에 CDN 서비스를 이용할 때 발생하는 트래픽을 분류하기 위해서는 플로우 간 연관관계를 고려하여 트래픽을 분류해야 한다. 이를 위해 본 장에서는 연속된 플로우 정보를 이용한 트래픽 분류 방법을 제안한다.

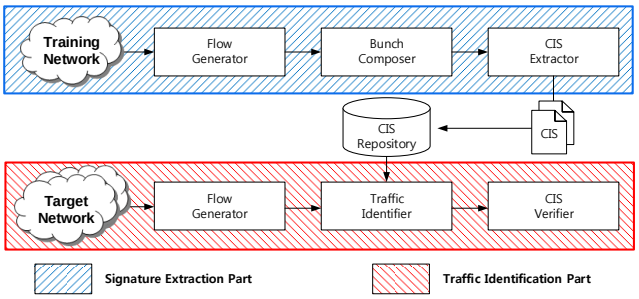

그림 2. Entire structure of identification system

그림 2 는 본 논문에서 제안하는 연속된 플로우 정보를 이용한 트래픽 분류 시스템의 구조를 나타낸다. 해당 시스템은 크게 2 개의 파트로 구성되며, 시그니쳐 추출부(Signature Extraction Part)에서는 트래픽 데이터를 플로우 데이터 형태로 변환하고, 플로우를 Bunch 로 구성한다. Bunch 로 구성한 후 Bunch 별 헤더 정보(IP, Port, Protocol)를 추출하여 CIS(Cascade Information Signature) 형태로 시그니쳐를 생성한다. 트래픽 분석부(Traffic Identification Part)에서는 트래픽 데이터를 플로우 데이터 형태로 변환하고, 생성된 CIS 와 Flow 데이터를 매칭하여 트래픽을 분류한다. 최종적으로 본 논문에서 제안하는 분류 방법의 결과를 기존의 트래픽 분류방법으로 분류한 결과와 비교하게 된다.

CIS(Cascade Information Signature) 추출 단계에서는 플로우의 헤더 정보를 사용한다. 플로우의 헤더 정보로 목적지 IP, 목적지 포트 번호, 프로토콜 번호를 사용한다.

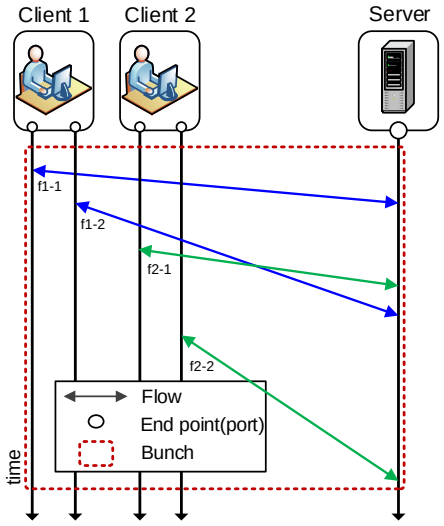

그림 3. Bunch 의 구조

일반적으로 사용자가 특정 서버에 접속하여 인터넷을 사용할 때, 단일 호스트와 단일 서버사이에서 발생하는 트래픽에서 나타나는 포트 번호와 여러 호스트와 서버와의 관계를 N:1 관계로 표현할 수 있다. 즉 여러 호스트가 여러 개의 포트번호를 사용하여 고정된 서버 포트에 접속한 다는 것이다. 이러한 서버와 다수의 호스트에서 발생하는 여러 플로우를 서버 정보가 동일한 것들을 모아서 Bunch 라는 플로우 집합을 정의하였다. Bunch 는 목적지(서버) IP, 포트 번호, 프로토콜이 같은 플로우들로 구성된다.

그림 3 은 Bunch 의 구조를 설명하기 위한 그림으로 Client1 과 Client2 가 특정 서버(IP, 포트, 프로토콜 번호)로 접근할 때 발생하는 플로우(f1-1, f1-2, f2-1, f2-2)는 각각 파란색(Client1), 녹색(Client2)으로 표현하였다. 그림 2 의 경우 Client1, 2 가 발생시킨 플로우들은 하나의 Bunch 로 구성되며, 이것은 목적지(서버)의 IP, 포트 번호, 프로토콜이 서로 같다는 것이다.

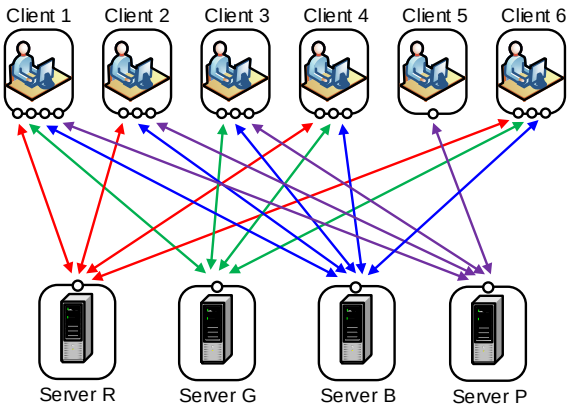

그림 4. 특정 사이트의 서비스에서 발생하는 플로우 예

그림 4 는 특정 사이트의 서비스 이용 시 클라이언트(1 ~ 6)과 서버(R, G, B, P) 사이에서 발생하는 플로우를 나타낸 그림이다. 발생한 플로우는 화살표로 표현되어 있다. 그림 4 의 경우 각 플로우들은 서버 기준으로 Bunch 가 구성되며 CIS 생성단계에서는 발생한 플로우를 바탕으로 구성된 Bunch 간 연관관계를 분석하여 규칙을 찾아낸다. Bunch 간 연관관계 분석을 통해 각 Bunch 의 Support(지지도), Confidence(신뢰도), 그리고 최종적으로 Lift(향상도) 값을 계산하여 특정 서비스를 사용할 때 발생하는 연속적인 플로우를 찾아낼 수 있다. 아래 수식은 Support, Confidence, Lift 를 계산하는 수식이다.

$$Support(A \Rightarrow B) = P(A \cap B) \quad (1)$$

$$Confidence(A \Rightarrow B) = P(B|A) = \frac{P(A \cap B)}{P(A)} \quad (2)$$

$$Lift(R \Rightarrow B) = \frac{P(B|R)}{P(B)} = \frac{P(R \cap B)}{P(B) \cdot P(R)} \quad (3)$$

Support(R⇒B)는 전체 트래픽에서 서버 R 의 플로우와 서버 B 의 플로우가 얼마나 같이 나타나는지를 의미하며 수식 (1)로 정의된다. Confidence(R⇒B)는 전체 트래픽에서 서버 R 의 플로우가 나타났을 때 해당 트레이스에서 서버 B 의 플로우도 나타나는 조건부 확률을 의미하며 수식(2)로 정의된다. Lift(R⇒B)는 서버 R 의 플로우 출현에 상관없이 서버 B 의 플로우가 나타나는 확률에 비해 서버 R 의 플로우가 나타날 경우 서버 B 의 플로우가 나타나는 확률의 증가 비율이며 수식(3)으로 정의된다. 즉, 두 서버의 플로우가 출현하는 것이 서로 관련이 없는 경우에는 Lift ≈1 이며, Lift > 1이면 두 서버 플로우의 출현은 서로 연관관계가 크다는 것이다. Lift < 1 인 경우 두 서버 플로우의 출현은 서로 반대의 관계가 있다는 것을 의미한다.

표 1 은 그림 4 의 플로우를 발생현황을 나타낸 것이다. 서버 R 과 B 의 Support 는 0.66 이며 Confidence 는 1 이 된다. 최종적으로 두 서버 R 과 B 의 Lift 는 1.2 가 되며 Lift > 1이므로 두 서버 플로우의 출현은 서로 연관관계가 크다는 것이다. 이와 같이 각 Bunch 의 모든 경우의 수를 계산하고 난 후 Lift > 1 을 만족하는 경우는 (R⇒B), (R⇒G), (B⇒G), ({R,B}⇒G)이다. Lift > 1 을 만족하는 경우를 시그니쳐로 표현하기 위해 DAG(Directed Acyclic Graph)를 사용하며, 위상정렬 알고리즘을 이용하여 연관관계 규칙 추출 결과를 그래프 형태의 그림 5 처럼 나타낼 수 있다.

표 1. 그림 4 의 플로우 발생 현황

Server	R	B	G	P
Client 1	O	O	O	O
Client 2	O	O		O
Client 3		O	O	O
Client 4	O	O	O	
Client 5				O
Client 6	O	O	O	

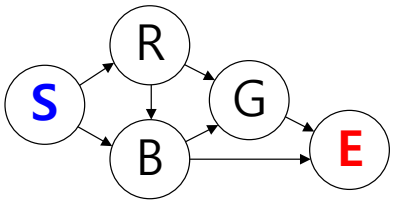

그림 5. 추출된 연관관계 규칙

표 2 는 그림 5 와 같이 추출된 연관관계 규칙을 DAG 형태의 시그니쳐로 나타내기 위한 위상정렬 알고리즘의 코드이다.

표 2. CIS 생성을 위한 위상정렬 알고리즘

```
1:  topologicalSort(R)
2:  Input : R is rule about cascade flow information
3:  For(i=1; i=R.count(); i++)
4:      Find node its indegree is zero → b
5:      B[i] = b
6:      Delete "b" node and out-coming edge of "b"
7:  End For;
8:  Return B[];
```

표 2 에서는 추출된 연관규칙을 입력으로 사용하며, 연관규칙의 모든 노드를 검사하여 DAG 형태의 그래프를 생성한다. 표 2 의 4 번째 줄은 해당 노드가 진입간선의 개수(incoming edge count, indegree)가 0 인 검색하고 선택하는 부분이며, 5 번째줄은 선택된 노드 b 를 B 에 저장하는 것이다. 6 번째 줄은 선택된 노드 b 의 모든 진출간선(outcoming edge)과 노드 b 를 삭제하는 것이다. 이와 같은 방법으로 모든 노드를 탐사하여 위상정렬이 완료된 그래프가 생성되며 이를 CIS 로 트래픽 분류에 사용한다.

5. 실험 및 결과

본 장에서는 제안하는 방법의 타당성 및 성능을 검증하기 위해 본 논문에서 제안하는 방법으로 CDN 서비스를 이용하는 각 웹서비스들의 CIS 를 생성하였다. 그리고 생성된 CIS 를 실제 학내 망에서 발생하는 트래픽 분류에 적용해보았다. 실험 대상으로 선정한 4 개 서비스는 Naver Sports, pooq, Daum Sports, Daum TV 이다. 최종적으로 CIS 의 CDN 트래픽 분류 성능을 확인하기 위해 CIS 를 이용한 트래픽 분류 결과와 페이로드 시그니쳐를 이용한 트래픽 분류 결과를 비교한다.

표 3. CIS 생성에 사용된 트래픽 정보

	Packet	Flow	Byte(MB)	CIS
Naver Sports	29,564,624	8,529	800	246
pooq	4,298,174	3,574	340	128
Daum Sports	6,438,894	5,321	533	157
Daum TV	5,165,127	4,268	412	201

표 3 은 각 웹서비스의 시그니쳐를 생성하기 위한 트래픽의 정량적 수치와 생성된 CIS 의 개수이다. 해당 트래픽을 이용하여 시그니쳐를 생성한 결과 각 응용별 246, 128, 157, 201 개의 연관관계 규칙이 추출되었고, 이를 이용하여 CIS 를 생성하였다.

생성된 CIS 시그니쳐와 페이로드 시그니쳐를 실제 학내망 트래픽 분류에 적용한 결과 각 웹 서비스에서 발생하는 CDN 서비스 트래픽 탐지에서 많은 차이가 있었다. 그림 6 은 CIS 와 페이로드 시그니쳐의 분류 결과를 그래프로 나타낸 것이다. 실험 결과 기존 페이로드 시그니쳐로 트래픽을 분류할 때, 각 웹 서비스에서 발생된 CDN 트래픽이 대부분 Unknown 트래픽으로 분류되었다. 그러나 본 논문에서 제안한 CIS 를 이용하면 Unknown 으로 분류된 트래픽이 각각 웹 서비스 별로 분류가 된다. 결론적으로 CDN 서비스를 이용하는 웹 서비스의 트래픽에서 실제 컨텐츠를 전송하는 CDN 트래픽을 각 웹 서비스 별로 분류할 수 있다는 것이다.

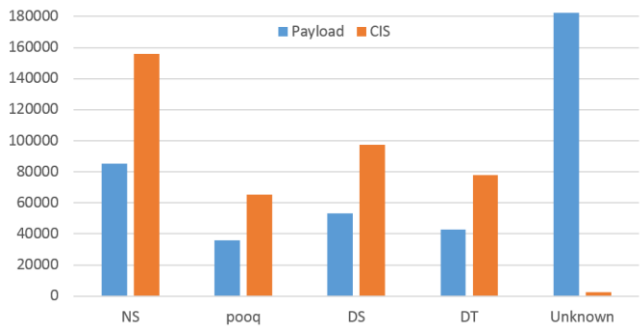

그림 7. CDN 트래픽 분석 결과 비교

6. 결론 및 향후 연구

본 논문에서는 CDN 트래픽을 분류할 때 기존의 트래픽 분류 방법의 한계점을 극복하기 연속된 플로우의 특징을 이용한 트래픽 분류 방법을 제안하였다. 연속되어 나타나는 플로우들의 특징을 시그니쳐로 추출하기 위한 방법을 연구하였으며 해당 시그니쳐를 CIS 로 정의하였다. 본 논문에서 제안한 CIS 를 사용한 결과 기존 방법으로는 Unknown 으로 분류된 대부분의 CDN 트래픽이 각 웹 서비스로 분류되었다. 결론적으로 본 논문에서 제안한 연속된 플로우 정보를 이용한 트래픽 분류 방법은 그동안 분류하지 못했던 CDN 트래픽을 각 웹 서비스 별로 분류 할 수 있다는 것이다.

향후 연구 계획으로는 CIS 를 이용한 트래픽 분류에 정확도 측면까지 고려한 방법을 연구할 계획이다.

참고 문헌

[1] https://en.wikipedia.org/wiki/Content_delivery_network

[2] H. Kim, K. C. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, “Internet traffic classification demystified: myths, caveats, and the best practices,” in Proceedings of the 2008 ACM CoNEXT conference, 2008, p.11

[3] 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.

[4] 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.

[5] Y. Wang, Y. Xiang, W. L. Zhou, and S. Z. Yu, "Generating regular expression signatures for network traffic classification in trusted network management," Journal of Network and Computer Applications, vol. 35, pp. 992-1000, May 2012.

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

[7] 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.

[8] An, H. M., Lee, S. K., Ham, J. H., & Kim, M. S. (2015). Traffic Identification Based on Applications using Statistical Signature Free from Abnormal TCP Behavior. JOURNAL OF INFORMATION SCIENCE AND ENGINEERING,31(5), 1669-1692.

[9] 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.

[10] Sung-Ho Yoon, Jun-Sang Park, and Myung-Sup Kim, "Behavior Signature for Fine-grained Traffic Identification," Applied Mathematics & Information Sciences Vol. 9, No. 2L, , Apr. 2015, pp. 523-534.