트래픽 분산 그래프를 이용한 이상 호스트 탐지 기법

(An Anomalous Host Detection Technique using Traffic Dispersion Graphs)

김 정 현 † 원 유 집 †† 안 수 한 †††
(Junghyun Kim) (Youjip Won) (Soohan Ahn)

요 약 오늘날의 인터넷은 일상생활에서 필수적인 요소가 되었으며, 인터넷의 이상 현상은 사회적 문제가 되고 있다. 이 때문에 인터넷 트래픽의 특성을 연구하기 위한 인터넷 측정 연구가 주목을 받고 있다. 특히 최근에는 트래픽 분산 그래프(TDG, Traffic Dispersion Graph)라는 새로운 트래픽 분석 기법이 제안되었다. TDG는 기존의 트래픽의 통계적 표현과 분석이 아닌, 그래프를 이용하여 네트워크 요소들의 상호작용을 표현하는 기법이다. 본 연구에서는 새로운 이상 탐지 기법의 패러다임과 TDG를 활용한 이상 탐지 기법을 제시한다. 기존의 이상 탐지 패러다임은 "비정상 패킷이나 플로우(Abnormal Packets or Flows)를 탐지하여 제거하자"는 것이지만, 본 연구에서는 "이상 트래픽의 근원이 되는 이상 호스트(Anomalous Hosts)를 탐지하여 이상 현상에 대응할 것"을 제안한다. 이를 위해서 TDG 클러스터링 기법(TDG Clustering Technique)을 고안하였다. 제안한 기법에 대한 실험에서 짧은 시간 안에 웜 바이러스(Worm Virus)에 감염된 호스트들을 찾아낼 수 있었고, 그 호스트들이 발생하는 이상 트래픽을 제거하여 정상적인 트래픽을 얻을 수 있었다. TDG 클러스터링 기법은 연산 속도가 빠르므로 실시간 이상 탐지에 적용될 수 있을 것으로 기대된다.

키워드 : 트래픽 분산 그래프, 인터넷 측정, 이상 탐지, 웜 바이러스

Abstract Today's Internet is one of the necessaries of our life. Anomalies of the Internet provoke social problems. For that reason, Internet Measurement which studies characteristics on Internet traffic attracts pubic attention. Recently, Traffic Dispersion Graph (TDG), a novel traffic analysis method, was proposed. The TDG is not a statistical analysis method but a graphical visualization method on interactions among network components. In this paper, we propose a new anomaly detection paradigm and its technique using TDG. The existing studies have focused on detecting anomalous packets of flows. On the other hand, we focus on detecting the sources of anomalous traffic. To realize our paradigm, we designed the TDG Clustering method. Through this method, we could classify anomalous hosts infected by various worm viruses. We obtained normal traffic through dropping traffic of the anomalous hosts. Especially, we expect that the TDG clustering method can be applied to real-time anomaly detection because calculations of the method are fast.

Key words : Traffic Dispersion Graphs, Internet Measurement, Anomaly Detection, Worm Virus

· 본 연구는 한국학술진흥재단 문제해결형인력양성지원사업(KRF-2006-511-D00370)과 2006년도 2단계 BK21사업의 지원으로 수행되었습니다.

† 비 회 원 : 한양대학교 전자컴퓨터통신공학과
junghyun@ece.hanyang.ac.kr
†† 종신회원 : 한양대학교 전자컴퓨터통신공학과 교수
yjwon@ece.hanyang.ac.kr
††† 비 회 원 : 서울시립대학교 통계학과 교수
sahn@uos.ac.kr

논문접수 : 2008년 6월 9일
심사완료 : 2009년 2월 5일

정보과학회논문지: 정보통신 제36권 제2호(2009.4)

1. 서 론

월드와이드웹(WWW, World Wide Web)이 일반화되면서 인터넷은 일상생활에서 필수적인 요소가 되었다. 그러나 여러 가지 예상치 못했던 이상 현상들이 인터넷에서 나타나고 있다[1]. 이러한 이상 현상은 사회적 문제를 일으키기도 하므로 세심하게 관찰하고 대응할 필요가 있다. 최근에 인터넷의 특성을 분석하여 인터넷의 개선 방향을 제시하는 인터넷 측정(Internet Measurement) 연구가 우리나라를 비롯한 세계 각국에서 주목을 받고 있다[2]. 특히 우리나라는 전 세계에서 인터넷 인프라가 가장 잘 갖추어진 국가다. 그러나 인터넷에서 발생되는 이상 현상을 탐지하고 대응하는 수준은 높지 못하다. 일반적으로 DoS(Denial of Service)[3] 또는 DDoS(Distributed DoS)[4] 공격이나 웜 바이러스(Worm Virus)[1,6-8]에 의한 이상 트래픽(Anomalous Traffic)이 이상 현상의 원인이 되고 있다. 예를 들어, 2000년에는 DoS(Denial of Service) 공격에 의해 Yahoo, eBay, E*trade가 피해를 입었고, 2001년에는 유사한 DoS 공격으로 인해 MS의 네임서버(Name Server)가 다운되었다[3]. 또 다른 예로, 2003년에는 슬래머웜(Slammer Worm)[6,8]과 블래스터웜(Blaster Worm)[7]이 우리나라를 비롯한 전 세계에 큰 피해를 주었다. Snort나 Bro와 같은 NIDS(Network Intrusion Detection System)나 방화벽(Firewall)이 설치되어 이상 현상의 대응에 사용되고 있으나 아직까지는 완벽한 보안은 기대하기 힘들다[9]. 따라서 지능화되고 있는 DDoS 공격 도구나 웜 바이러스들에 대한 많은 연구가 앞으로도 필요하다. 이러한 이상 현상이 자주 발생하는 핵심적인 원인은, 인터넷이 처음 만들어질 때 다양한 예외 상황을 감안하고 있지 않았으며, 인터넷의 각 구성 요소들이 자체적인 버그를 포함하고 있기 때문이다[2,4]. 이러한 허점을 미리 찾아내고 대응할 수 있도록 하는 것은 물론, 프로토콜 개발과 개량에 필요한 정보를 제공하는 것이 인터넷 측정의 핵심적인 목적이다[2].

본 연구에서는 2007년에 발표된 새로운 인터넷 측정 방식인 TDGs(Traffic Dispersion Graphs)[5]를 이용하여 인터넷 구성 요소 간의 상호작용을 분석한다. 본 연구에서 실행한 TDG 분석 항목에는 응용층 프로토콜(Application-Layer Protocol)인 HTTP, DNS, P2P가 있다. 또한 본 연구에서는 TDG의 특성을 활용하여 이상 호스트(Anomalous Hosts)를 탐지하는 TDG 클러스터링(TDG Clustering)을 제안한다. 기존의 이상 탐지 연구는 패킷(Packet) 또는 플로우(Flow) 단위 분석을 통하여 비정상 트래픽(Anomalous Traffic)을 찾는 것에 초점이 맞추어져 있다[3,9,16]. 그러나 비정상 트래픽이 한 번 제거되더라도, 비정상 트래픽을 발생시키는 이상 호스트는 계속 활동하고 있으므로, 기존에 제안된 대응책들은 근본적인 해결책이 될 수 없다. 본 연구에서 제안하는 기법은 매우 짧은 시간(5초 정도)의 트래픽을 TDG 클러스터링으로 분석하여 비정상적 상호작용을 보이는 이상 호스트(Anomalous Host)를 찾아내는 기법이다. 현재까지 이상 호스트를 찾아 인터넷의 이상 현상에 대응하는 이상 탐지 패러다임은 발표되지 않았으므로, 본 연구에서 제안하는 기법과 동일하게 비교할 수 있는 관련 연구는 없는 것으로 판단된다. 이상 호스트가 탐지될 경우, 탐지된 호스트에 설치된 악성 코드를 제거하도록 관리자에게 권고하거나, 네트워크 사용을 제안하는 방식으로 근본적으로 이상 트래픽을 막을 수 있다. 특히 제안한 기법의 연산 속도가 빠르므로 실시간 이상 탐지에도 활용할 수 있을 것으로 기대된다. 본 연구의 실험에서는, 웜 바이러스에 감염되어 호스트 스캐닝을 하고 있는 이상 호스트들을 탐지할 수 있었다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 TDG가 무엇이고, TDG를 어떻게 생성하는지 설명한다. 3장에서는 TDG를 이용한 인터넷 측정의 간단한 예를 살펴보고, 4장에서는 TDG 클러스터링 기법을 설명하고, 이상 호스트를 탐지한 예를 살펴본다.

2. TDGs(Traffic Dispersion Graphs)

2.1 정의(Definition)

TDG(Traffic Dispersion Graph)는 새로운 트래픽 시각화 기법(Traffic Visualization Method)이다[5]. 기존의 패킷(Packet), 플로우(Flow) 등의 시각화 방법에 비하여, 호스트(Host) 간의 상호작용을 효과적으로 시각화할 수 있다. 즉, TDG는 “누가 누구와 통신하는가?(Who talks to whom?)”를 시각화하는 기법이다. TDG는 방향성 그래프(Directed Graph)의 일종으로, 노드(Node)는 특정한 IP 주소(Distinct IP Address)를, 에지(Edge)는 패킷을 이용한 노드 간의 상호작용(Interaction)을 나타낸다. 특히, TDG는 시간 및 공간적으로 다르게 나타난다. 어떤 시간에, 얼마 동안, 어떤 링크의 트래픽을 표현하는지에 따라 다양한 TDG를 얻을 수 있다.

2.2 에지 필터(Edge Filter)

TDG에서 에지는 노드 간의 상호작용이 있음을 의미한다. 에지가 어떤 기준으로 구성되느냐에 따라 동일한 트래픽에서도 다른 TDG를 얻는다. 따라서 에지를 결정하는 에지 필터는 TDG의 생성에서 매우 중요한 부분이다. 에지는 구성하는 가장 기본적인 방법은 어떤 노드에서 또다른 노드로 1개 이상의 패킷이 전달되면 상호작용이 있는 것으로 간주하고 에지는 생성하는 것이다.

또는 SYN 패킷이 발생하면 통신이 시작되는 것이므로 에지를 생성한다. 이외에도 TCP 3단계 패킷 교환(TCP 3-way handshake)이 발생할 경우 에지는 생성하는 방법도 좋은 선택이 될 수 있다. 포트(Port)를 에지 필터의 조건에 포함 시킬 경우, 네트워크 어플리케이션의 종류에 따라 호스트들이 어떻게 상호작용하는지 알 수 있다. 기존 연구[5]에 사용된 대표적인 에지 필터로 EFP 필터와 EFSP 필터를 들 수 있다.

- EFP 필터(The Edge on First Packet Filter): 어떤 노드에서 또다른 노드로 1개 이상의 패킷이 발생하게 되면, 노드 간의 상호작용이 이루어진 것으로 간주하여 에지를 생성한다. 주로 UDP 트래픽에 적합하다.
- EFSP 필터(The Edge on First SYN Packet Filter): 어떤 노드에서 또다른 노드로 1개 이상의 SYN 패킷이 발생하면, 노드 간의 상호작용이 이루어진 것으로 간주하여 에지를 생성한다. 주로 TCP 트래픽에 적합하다.

2.3 TDG의 생성(Generating TDGs)

(1) 캡쳐된 트래픽에서 분석 지점(Observation Point)을 선택한다. (2) 에지 필터를 선택한다. 예를 들어, HTTP 트래픽의 TDG를 얻기 위해서는 EFSP 필터와 80 포트를 선택한다. (3) 패킷이 에지 필터를 만족할 경우 에지를 생성한다. (4) IP 주소를 기준으로 노드를 생성하고, 에지들을 노드와 연결하여 TDG를 생성한다.

생성된 논리적 TDG를 효과적으로 시각화하기 위해서는 적절한 도구가 필요하다. 본 연구에서는 기존 연구[5]와 같이 GraphViz[10]를 시각화 도구로 사용했다.

3. TDG와 인터넷 측정

3.1 데이터(Data)

본 연구에서는, 국내외 국외를 연결하는 백본망의 155 Mbps인 한 링크에서 일주일간(2004년 10월 29일부터 11월 4일까지) 수집한 트래픽 데이터를 사용하였다. 대부분의 인터넷 측정 연구에서는 대학 캠퍼스에서 수집된 트래픽 데이터를 주로 사용하다. 본 연구에서 사용한 데이터는 우리나라의 백본망에서 수집된 흔치 않은 데이터다. 캡쳐된 원본 트래픽은 pcap[11] 포맷이며, 샘플링되지 않은 약 1.5TB(Terabytes) 용량의 데이터다.

본 연구에서 사용한 트래픽 관리 유틸리티는 직접 개발한 것들이며, 그 집합은 “DMC_Traff_Mon”이라고 부른다. “DMC_Traffic_Mon”은 libpcap[11]과 MySQL[12] 기반으로 개발되었으며, 연구의 목적에 맞게 최적화 되어 빠른 속도로 트래픽 데이터를 관리한다. IP 주소(IP address)나 페이로드(Payload)와 같은 민감한 데이터를 암호화하는 기능도 포함하고 있다. 그림 1은 본 연구에서 사용한 일주일간의 트래픽 데이터를 packets/hour

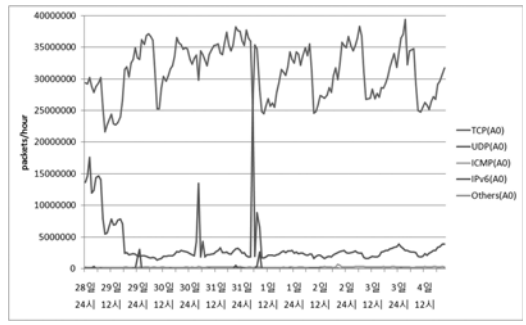


그림 1 일주일간의 트래픽 데이터

표 1 대상 트래픽(30분 동안)의 정보

	Pkt Count	Pkt/sec	Avg Pkt Size
TCP	16,024,808	8902.67	324.99 Byte
UDP	1,207,874	671.04	93.47 Byte

그래프로 나타낸 것이다.

주로 사용한 데이터는 11월 2일의 30분간의 트래픽이다. 또한 국외에서 국내로 유입되는 Ingress 트래픽이다. 표 1은 대상 트래픽에 대해 TCP와 UDP로 나누어 전체 패킷 개수(Packet Count), 초당 패킷 개수(Packets/Sec), 평균 패킷 크기(Average Packet Size)를 나타내고 있다.

3.2 TDG를 이용한 트래픽 시각화

TDG를 이용하여 HTTP(Hypertext Transfer Protocol), DNS(Domain Name System), P2P(Peer to Peer) 트래픽을 시각화한 예를 살펴본다.

그림 2는 120초 동안의 HTTP 트래픽을 TDG로 표현한 것이다. HTTP는 WWW의 기반이 되는 응용층 프로토콜(Application Layer Protocol)이며, TCP 트래픽의 약 20%를 차지한다. HTTP 트래픽을 TDG로 시각화하기 위해서 EFSP 필터를 사용하였고, 포트는 80과 8080을 선택하였다. 일반적으로 알려져 있는 것과 같이, HTTP는 몇몇 웹 서버들(Web Servers)을 중심으로 클라이언트들(Clients)이 클러스터(Cluster)를 이루고 있다. 웹 서버에 해당하는 노드의 경우에는 Indegree가 높게 나타나고 있다. 대상 트래픽은 국외에서 국내로 유입되는 Ingress 트래픽이므로, 그림 2는 국외 사용자가 국내의 웹 서버를 사용하고 있는 형태라고 볼 수 있다.

그림 3은 60초 동안의 DNS 트래픽을 TDG로 표현한 것이다. DNS 트래픽은 UDP 트래픽의 대부분을 차지하는 응용층 프로토콜이다. DNS 트래픽을 시각화하기 위해서 EFP 필터를 사용하였고, 포트는 53을 선택하였다. 인터넷에서 DNS는 매우 중요한 역할을 한다. 번호로 설정된 IP 주소 자체를 기억하는 것은 쉽지 않은 일인

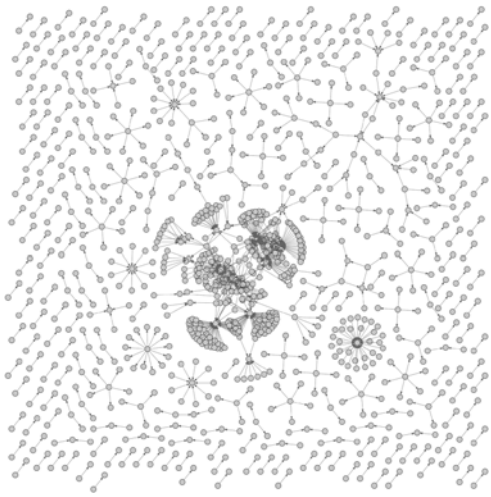


그림 2 HTTP의 TDG(120초 동안 1135개 노드)

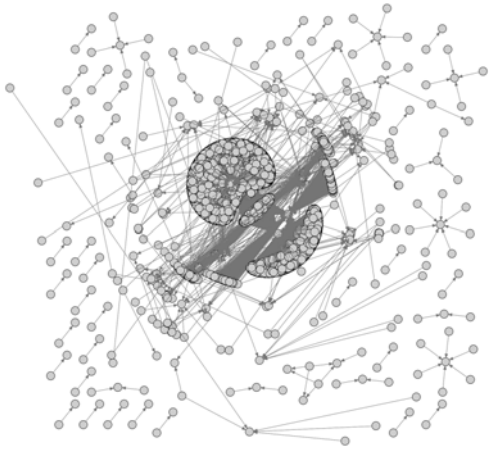


그림 3 DNS의 TDG(60초 동안 12877개 노드)

데, IP 주소를 문자로 된 도메인 네임으로 사용할 수 있도록 한다. 그림 3에서 볼 수 있듯이 60초 동안 매우 많은 노드가 발생함을 알 수 있다. 즉, 소수의 DNS 서버가 많은 인터넷 사용자에게 서비스를 제공하고 있음을 확인할 수 있다.

본 연구에서는 국내에서 많이 사용되는 소리바다를 P2P의 예로 선택하였다. 그림 4는 소리바다의 트래픽을 TDG를 표현한 것이다. 소리바다는 TCP의 경우에는 7675, 7676, 7677, 22322 포트를 사용하며, UDP의 경우에는 7674, 22321을 사용한다. 소리바다의 TDG 표현을 위해서, TCP 트래픽에 대해서 EFSP 필터를 사용했다.

그림 4는 전형적인 P2P 트래픽의 TDG를 보여주고 있다. 전체적인 모든 노드가 하나의 연결된 형태를 보이고 있기 때문이다. 대상 트래픽이 국외에서 국내로 유입

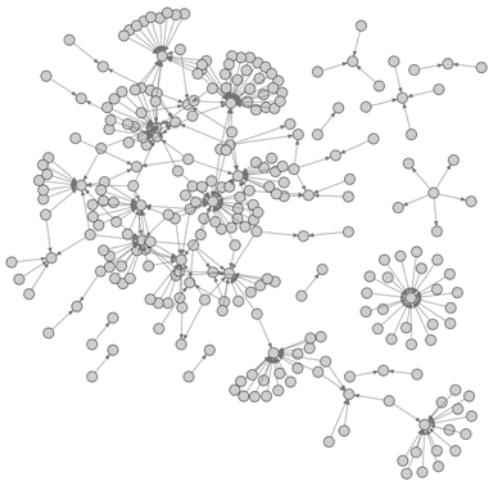


그림 4 소리바다의 TDG(600초 동안 271개 노드)

되는 Ingress 트래픽이므로, 600초 동안 271개의 노드만이 나타나고 있다. 이것은 국외의 사용자가 주로 국내 음악의 MP3 파일을 공유하는 소리바다를 많이 사용하지 않기 때문이다.

4. TDG 클러스터링: 이상 호스트 탐지 기법

4.1 TDG 클러스터링 기법

인터넷 트래픽을 TDG로 표현할 경우, 매우 다양한 Indegree와 Outdegree를 가진 노드들이 나타나고 있음을 확인할 수 있었다. 이러한 TDG의 특성을 기반으로 K-means 클러스터링을 할 경우, TDG의 노드들을 분류할 수 있다. 그림 5는 TDG 클러스터링을 이용한 이상 탐지 기법을 나타내고 있다.

TDG 클러스터링의 첫 번째 단계는 TDG 생성하기(Generating A TDG)이다. TDG 생성과 관련된 내용은 2장에서 언급하였다.

두 번째 단계는 TDG 특징 추출(Extracting TDG Features)이다. TDG의 특성을 활용하기 위해서 본 연

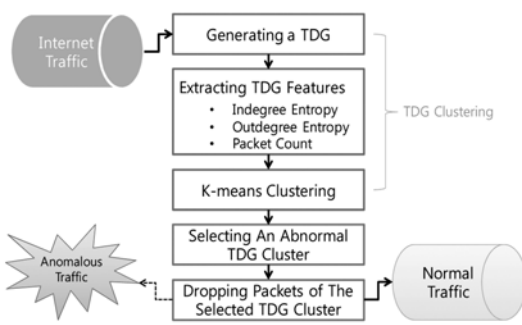


그림 5 TDG 클러스터링의 이상 탐지 활용

구에서는 각각의 노드에 대해서 “Indegree 엔트로피(Indegree Entropy)”, “Outdegree 엔트로피(Outdegree Entropy)”, “패킷 개수(Packet Count)”와 같은 3개의 변수를 선택하였다. 노드의 Indegree와 Outdegree는 노드간의 상호작용이 비정상인지를 판단하는 중요한 기준이 된다. 노드 간의 상호작용에서 얼마나 많은 트래픽을 발생했는지 판단하는 것 또한 매우 중요하므로 “패킷 개수”를 포함시켰다. 3개의 변수를 선택한 또다른 이유는 연산 시간 때문이다. 더 많은 변수의 조합은 좋은 결과를 기대할 수 있으나, 많은 연산시간을 요구하므로 실제 활용에 있어서 문제가 있다. 본 연구에서는 가능한 빠르게 비정상 노드를 분류할 수 있도록 하기 위해서 TDG의 특성을 고려하여 3개의 변수만을 선택했다.

엔트로피(Entropy)는 대상 데이터들의 복잡성을 분석할 때 효과적이며, 네트워크의 이상 탐지에 많이 활용되고 있다[9,16,18]. 엔트로피는 다음의 식 (1)과 같이 나타낼 수 있다[17].

$$H_{in/out}(X) = -\sum_{i=0}^{n} p_i(x)\log_2 p_i(x) \quad (1)$$

X는 하나의 노드를 의미하며, $H_{in/out}(X)$는 하나의 노드에 대한 Indegree 엔트로피 또는 Outdegree 엔트로피가 된다.

엔트로피 분석에서 $p_i(x)$를 정의하는 것은 매우 중요하다. 본 연구에서는 $p_i(x)$를 식 (2)와 같이 정의하여 사용한다.

$$p_i(x) = \frac{\alpha_i}{\eta} \quad (2)$$

Indegree 및 Outdegree 엔트로피는 각각의 노드에 대해서 구하게 된다. Indegree 엔트로피($H_{in}(X)$)를 구할 경우, α_i는 다른 노드로부터의 특정한(Distinct) Indegree를 의미하며, η는 모든(Total) Indegree를 의미한다. Outgdegree 엔트로피($H_{out}(X)$)를 구할 경우, α_i는 다른 노드로부터의 특정한(Distinct) Outdegree를 의미하며, η는 모든(Total) Outdegree를 의미한다. 각각의 다른 노드로부터의 특정한 Indegree에는 i로 번호가 매겨지며 (i = 0,1,2,…,n), i마다 식 (2)와 같은 $p_i(x)$를 구하여 식 (1)의 $H_{in/out}(X)$를 구하면 노드의 Indegree 엔트로피 또는 Outdegree 엔트로피를 얻게 된다. 이상 호스트가 발생시키는 호스트 스캔이나 포트 스캔을 탐지하기 위하여, 가중치(Weighted Value)를 적용하였다. 웜 바이러스에 감염된 호스트의 경우, 무작위로 IP 주소를 선택하여 스캔을 시도하며, 포트 스캔의 경우에도 특정 호스트에게 여러 포트에 대한 스캔을 시도하므로 Outdegree에 많은 가중치를 두었다. 식 (3)은 가중치 적용을 나타낸 식이다.

$$v_i = (w_x x_i, w_y y_i, w_z z_i) \quad (3)$$

v_i는 벡터 공간(Vector Space) 상의 한 점을 나타내며, 하나의 노드와 동일하다. i는 전체 노드의 번호이며 (i = 0,1,2,…,n), x, y, z는 각각 Indgree 엔트로피, Outdegree 엔트로피, 패킷 개수를 의미한다. w_x, w_y, w_z는 가중치이며, 각각 17, 34, 1을 선택하였다. “Indegree 엔트로피”의 평균은 0.928, 최대값은 129.620이었고, “Outdegree 엔트로피”의 평균은 5.598, 최대값은 325.23이었다. “패킷 개수”의 평균은 4.117, 최대값은 510개였다. 각 변수의 특징에서 볼 수 있듯이, “패킷 개수” 값이 크기 때문에 가중치를 통해 적절히 보정할 필요가 있었다. 본 연구에서 클러스터링에 가장 영향이 큰 “패킷 개수”의 가중치를 줄이고, “Indegree 엔트로피”와 “Outdegree 엔트로피”의 가중치를 높였다. 가장 관심이 있는 것은 degree로 상호작용의 정상 및 비정상을 판단하는 것이고, 패킷 개수는 발생된 트래픽의 양을 감안하기 위한 것이다. 가중치는 실험을 통해 이상 탐지에 적당한 값을 선택한 것이다. 즉, 연구 대상 트래픽의 여러 구간(Period)을 선택하여, 이상 탐지의 정확도가 가장 높은 것을 선택하였다. 가중치의 선택은 실제 우리나라에서 발생된 트래픽을 기반으로 했으므로, 선택된 가중치는 국내망의 다른 트래픽에서도 그대로 적용 가능할 것으로 예상된다. 이와 관련한 심도 있는 연구는 또다른 실제 트래픽의 확보 및 분석과 관련된 문제이므로 본 논문의 범위를 넘어선다.

세 번째 단계는 K-means 클러스터링(K-means Clustering)[19]이다. 비지도 학습(Unsupervised Learning) 방식의 대표적 기법인 K-means 클러스터링(K-means Clustering)을 비정상 노드들의 분류에 사용하였다. K-means 클러스터링은 비교적 단순하고, 실제 활용에서 매우 빠른 속도를 보인다[19]. 따라서 실시간 이상 탐지에도 적절하다. 알고리즘은 그림 6과 같다.

네 번째 단계는 비정상 TDG 클러스터를 선택하는 것(Selecting An Abnormal TDG Cluster)이다. 짧은 시간 동안 너무 많은 Outdegree 엔트로피와 가장 작은 Indegree 엔트로피를 가진 노드들을 많이 포함한 TDG 클러스터는 포트 스캔 또는 호스트 스캔 중인 호스트일

```
1: Select K points as initial centroids
2: Repeat
3:     From K clusters by assigning each point to its
       closest centroid
4:     Recompute the centroid each cluster
5: Until centroids do not change
```

그림 6 K-means 클러스터링 알고리즘

가능성이 매우 높기 때문이다. 따라서 K개의 TDG 클러스터 중에서 중심(Centroid)인 $w_y y_c$가 가장 크고, $w_x x_c$가 0에 가까운 TDG 클러스터를 비정상 TDG 클러스터로 선택한다. x_c와 y_c는 각각 중심의 x와 y를 의미한다.

마지막, 다섯 번째 단계는 네 번째 단계에서 선택된 비정상 TDG 클러스터와 관련된 패킷을 제거하는 것(Dropping Packets of The Selected TDG Cluster)이다. 또한 탐지된 이상 노드들은 실제 이상 호스트과 일치하므로, 해당 호스트의 관리자에게 악성 코드에 대한 제거를 요구하거나, 네트워크 사용을 잠시 제한함으로써, 이상 현상의 원인을 근본적으로 제거할 수 있게 된다.

TDG 클러스터링은 JDK(Java Development Kit) 1.6과 Netbeans 6.0(Java Integrated Development Environment)을 기반으로 구현했다.

4.2 분석 대상 구간의 TDG

본 연구에서는 짧은 시간 안에 이상 현상을 탐지할 수 있도록 하기 위해 대상 구간의 길이를 5초로 선택했다. 더 짧은 시간을 선택할 경우, TDG를 구성하는 노드들의 특성 판단에 필요한 정보가 너무 작아지게 된다.

그림 7은 5초 동안의 TCP 트래픽을 TDG로 표현한 것이다. 전체 노드는 3109개, 에지는 2435개이다. 그림 7을 살펴보면 5초의 짧은 시간 동안에 다양한 Indegree와 Outdegree를 가진 노드들이 나타나고 있다. 그림 7에 표현된 5초 동안의 TCP 트래픽에서는 특별한 크기 이상(Volume Anomaly)가 발견되지 않았다. 따라서 Indegree가 어느정도 나타나는 노드들은 대부분 웹 서버나 P2P 등에 관련된 노드들로 판단된다. 분석 결과 실제로도 웹 서버와 P2P 등에 관련된 것들이었다. 한편,

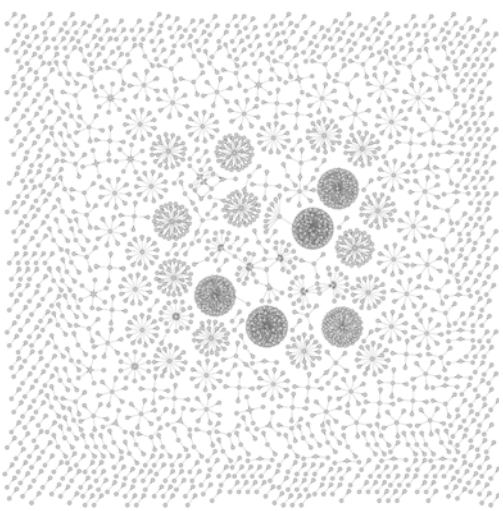


그림 7 5초 동안의 TCP 트래픽에 대한 TDG

그림 9에서 명확히 Outdegree가 매우 큰 노드들이 여러 덩어리 형태로 나타나고 있다. 이러한 모양의 TDG가 나타난 이유는, 짧은 시간 안에 너무 많은 Outdegree가 나타나고 있기 때문이다. 따라서 웜 바이러스나 공격 도구들의 포트 또는 호스트 스캔일 가능성이 높다. 더 자세한 분석을 위해서 그림 8의 노드들의 Degree 분포를 나타내었다. 편의상 Degree Count(x축)에 따른 1-P(X<x)(y축)를 log 스케일로 나타내었다. 그림 8에서 알 수 있는 것은 5초 동안의 트래픽에서 최대 15개의 Indegree를 가진 노드가 나타났으며 정상적인 수준의 상호작용을 보였다는 점이다. 반면, 최대 200여개의 Outdegree를 가진 노드가 나타나기도 했다는데, Outdegree 분포가 비정상 적임을 확인할 수 있다. 5초 동안 200개가 넘는 호스트들과 상호작용하는 호스트가 있는 것이다. 이것은 전형적인 웜 바이러스에 감염된 호스트의 동작과 일치한다. 웜 바이러스에 감염된 호스트는 취약한 호스트를 찾기 위해 랜덤 스캔을 하기 때문이다[1,6-8]. 본 연구에서는 분석 대상 구간의 TDG를 기반으로, 이상 현상을 발생시키는 비정상 노드들을 분류해 낼 것이다. 또한 분류된 비정상 노드들의 이상 트래픽을 제거하여 정상 트래픽을 얻을 수 있음을 보일 것이다.

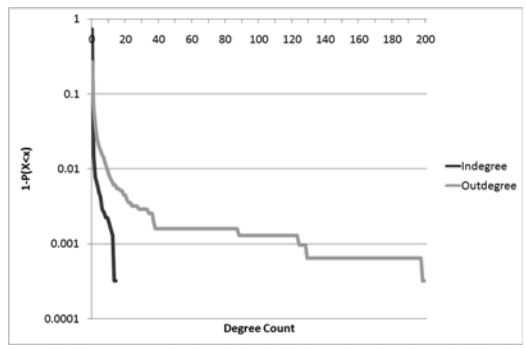


그림 8 대상 구간의 Degree 분포

4.3 TDG 클러스터링의 결과

본 절에서는 4.1절에서 소개했던 TDG 클러스터링 기법을 적용한 결과를 살펴본다. 4.1절에서 설명했듯이, TDG 클러스터링 과정에서 K-means 클러스터링을 활용한다. 여기서 K를 3으로 선택했다. K의 선택에 대한 자세한 내용은 다음절에서 살펴본다.

그림 9는 TDG 클러스터링을 수행한 결과다. SAS[15]의 CANDISE Procedure를 이용해서 클러스터링된 3차원 벡터들을 2차원으로 표현했다. 그림 9에서 볼 수 있듯이 각 클러스터를 “TDG 클러스터0”, “TDG 클러스터1”, “TDG 클러스터2”로 이름을 붙였으며, 번호는 클러스터에 포함된 노드의 개수가 작은 것부터 정렬된

것이다. 참고로, 분류된 TDG클러스터들의 중심(Centriod of TDG Cluster)을 식 (3)과 같이 나타내면 다음과 같다.

$$ctc_0 = (0,\ 240.805,\ 170)$$

$$ctc_1 = (0,\ 73.732,\ 10.568)$$

$$ctc_2 = (0.994,\ 0.008,\ 2.896)$$

표 2는 분류된 TDG 클러스터들의 특징을 분류한 것이다. 표 2로부터 본 연구에서 제안한 TDG 클러스터링 기법이 호스트들은 비정상 호스트, 정상 서버, 정상 클라이언트들로 분류함을 볼 수 있다. TDG 클러스터링 단계(그림 5 및 4.1절)의 네 번째 단계에 의해서 비정상 TDG 클러스터는 TDG 클러스터1이 선택된다.

그림 10은 TDG 클러스터0과 1을 TDG로 표현한 것이다. 앞에서 설명했듯이, TDG 클러스터0의 TDG에는 주로 Indegree가 많은 노드들이 분류되어 있으며 정상적인 상호작용을 보이고 있다(그림 10(a)). TDG 클러스터1의 TDG에서는 Outdegree가 많은 노드들이 분류되어 있으며 웜 바이러스의 전형적인 행태를 보이고 있다.

표 3은 비정상 노드들의 집합인 TDG 클러스터1의 포트 특성을 표로 나타낸 것이다. 가장 많이 나타나는 5개의 포트(135, 445, 17300, 9898, 5554)는 여러 웜 바이러스들이 자주 사용하는 포트이다. Blaster 웜과 Welchia웜은 135, 445 포트, Sasser웜은 445와 555포트,

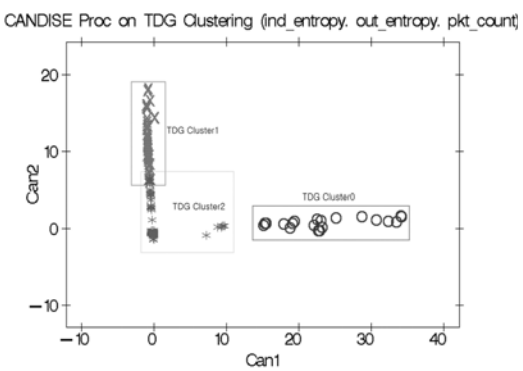
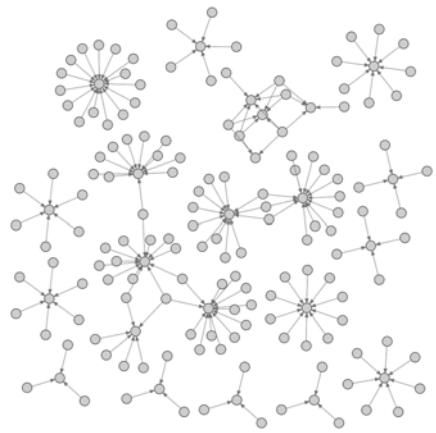


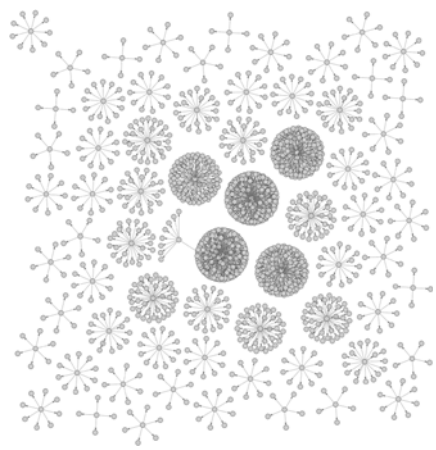
그림 9 TDG 클러스터링으로 분류된 노드들

표 2 TDG 클러스터들의 특징

클러스터	특징
TDG 클러스터0	Indegree가 큰 노드들이며, 주로 서버들
TDG 클러스터1	Indegree가 거의 없고, 비정상적으로 Outdegree가 큰 노드들이며, 주로 웜 바이러스와 같은 악성 코드에 감염된 호스트들
TDG 클러스터2	Indegree와 Outdegree가 작거나 없는 노드들



(a) TDG 클러스터0 (23개 정상 노드)



(b) TDG 클러스터1 (69개 비정상 노드)

그림 10 TDG 클러스터0과 1의 TDG

표 3 TDG 클러스터1의 포트 특성

순위	포트	개수	비율	누적
1	135	1079	62.98%	62.98%
2	445	245	14.30%	77.29%
3	9898	145	8.46%	85.75%
4	17300	129	7.53%	93.28%
5	5554	109	6.36%	99.64%
6	기타	6	0.36%	100%

Dabber 웜은 5554와 9898포트을 사용한다. 표 3에서 나타난 결과대로, TDG 클러스터1에 포함된 호스트 중 99.64%가 웜 바이러스에 의한 이상 트래픽을 발생시키는 것으로 확인되었고, 0.36%만이 정상일 가능성이 있었다. TDG 클러스터1의 노드들은 5초 동안 평균 21.39개의 패킷을 발생시켰다. 최소 4개부터 최대 200개까지 다양하게 나타났다.

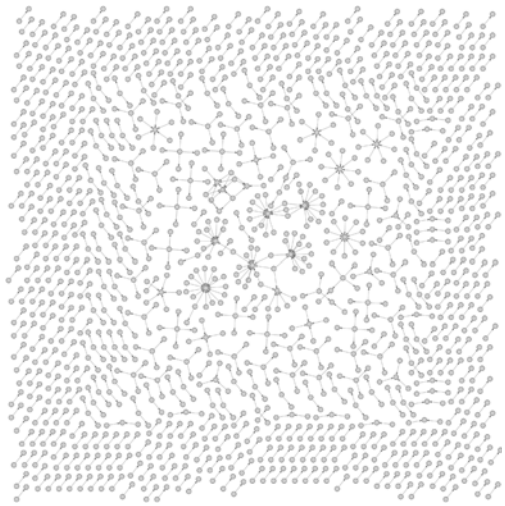


그림 11 TDG 클러스터1을 제거한 TDG

그림 11은 비정상 노드들의 클러스터인, TDG 클러스터1에 해당하는 노드들의 트래픽을 제거한 것이다. 즉, TDG 클러스터0과 2만이 TDG로 표현된 것이다. 그림 11은 비정상적인 행태를 보이는 노드들이 제거되었으므로 정상적인 노드 간의 상호작용을 볼 수 있다.

표 4는 대상 트래픽(그림 7)의 포트 특성을 보이고 있다. 표 5는 TDG 클러스터1을 제거한 TDG(그림 11)의 포트 특성을 나타내었다. 주로 웜 바이러스에 의해 사용되는 포트의 트래픽은 대상 트래픽에서 5.06%(표 4)였고, TDG 클러스터1을 제거한 후에는 1.51%(표 5)로 줄었다. 나머지 1.51%의 의심되는 트래픽을 분석한 결과, 5초 동안 최소 1개부터, 최대 4개의 패킷이 나타났고 평균 1.6개의 패킷이 발생했다. 5초간 평균 1.6개의 패킷을 발생시킨 노드들을 비정상적 상호작용을 한다고 판단할 수는 없다. 이러한 패킷들까지 제거한다면 정상 서비스가 방해를 받을 수 있다. 본 논문에서 제안한 기법은 5초마다 이상 현상을 탐지한다. 만약 표 4의 1.51%의 트래픽을 발생시키는 노드 중에서 웜 바이러스에 감염된 노드들(이번 5초에서 단지 랜덤 스캔의 패킷 1~2개만 발생시킨 것들)이 있다면, 다음 번 5초 간의 트래픽에 대한 이상 탐지에서 정확히 탐지될 것이다.

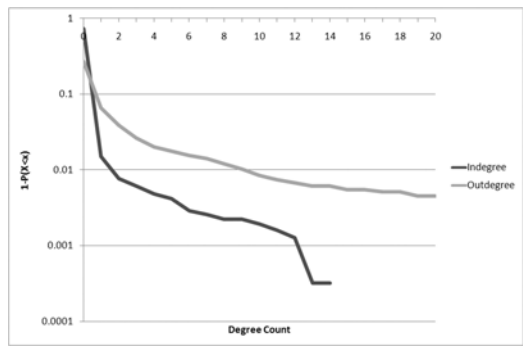
그림 12는 그림 7와 11의 TDG에 대한 Degree 분포를 나타낸 것이다. 그림 12는 그림 8과 같이 Degree Count(x축)에 따른 1-P(X<x)(y축)를 log 스케일로 나타내었다. 그림 12(a)는 그림 7의 TDG의 Degree 분포를 나타낸 것으로 그림 8과 동일하지만, 그림 12(b)와 비교를 위해 X축(Degree Count)의 스케일(Scale)을 최대 20으로 조정한 것이다. 그림 12(b)는 그림 11의 Degree 분포를 나타낸 것이다. 그림 12(a)와 12(b)를

표 4 대상 트래픽의 비정상 포트 특성

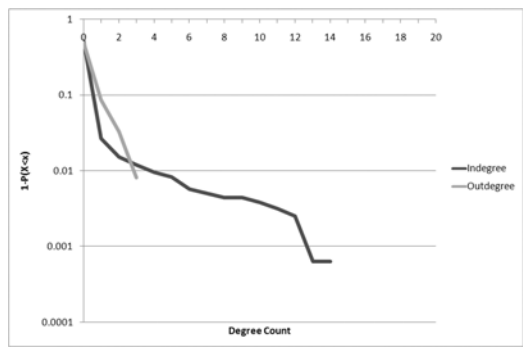
순위	포트	개수	비율	누적
1	135	1593	3.35%	3.35%
2	445	434	0.91%	4.26%
3	9898	145	0.30%	4.56%
4	17300	129	0.27%	4.83%
5	5554	113	0.23%	5.06%
6	기타	45100	94.94	100%

표 5 TDG 클러스터0과 2의 TDG의 비정상 포트 특성

순위	포트	개수	비율	누적
1	135	514	1.10%	1.10%
2	445	189	0.41%	1.51%
3	5554	4	0.0%	1.51%
4	9898	0	0.0%	1.51%
5	17300	0	0.0%	1.51%
6	기타	45066	98.49%	100%



(a) 대상 구간의 Degree 분포(X축의 스케일 조정)



(b) TDG 클러스터1을 제거한 TDG의 Degree 분포

그림 12 비정상 TDG 클러스터의 삭제 전과 삭제 후의 Degree 분포

비교하면 대량의 Outdegree 분포가 제거된 것을 확인할 수 있다. 이것은 포트 또는 호스트 스캔을 수행하는 비정상 노드들(즉, TDG 클러스터1)을 제거했기 때문이다.

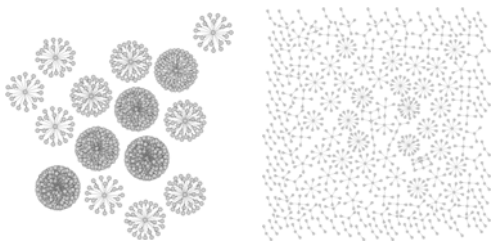
실험결과, 제안한 기법은 다른 구간에서도 평균적으로 약 98~99%의 정확도를 보였다. 기타 1~2% 정도의 노드들은 정상일 수도 있지만, 알려지지 않은 새로운 형태의 웜 바이러스일 가능성이 있다. 본 연구에서 제안한 기법은 기존에 알려진 이상 현상 이외에도 새로운 이상 현상을 탐지 할 수 있기 때문이다.

본 연구에서 제안한 TDG 클러스터링 기법을 사용할 경우, 네트워크 관리자의 정책에 따라 기타 1~2%는 통과시키거나 막는 방식을 사용하면 매우 효과적으로 다양한 이상 현상에 대응할 수 있을 것이다. 단지 이상 현상을 일으키는 트래픽만을 제거하는 것뿐만이 아니라, 이상 트래픽을 계속 발생시키는 호스트들에 대한 대응을 감안하고 있기 때문이다.

특히 TDG 클러스터링으로 분류된 결과를 기반으로 NN분류기(Nearest-Neighbor Classfier)[19]를 적용하면 매우 빠른 시간 안에 비정상 노드를 분류해 낼 수 있을 것으로 기대된다. 즉, 분석 대상 트래픽을 미리 TDG 클러스터링하여 비정상 호스트에 대한 학습(Learning)을 수행한 후에는, NN 분류기로 매우 빠르게 비정상 호스트를 탐지할 수 있다. 단, NN 분류기를 사용할 경우, 미리 TDG 클러스터링으로 탐지되었던 이상(anomaly)과 유사한 경우만 탐지할 수 있다는 한계가 있다.

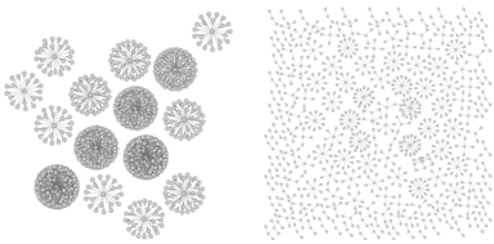
4.4 K의 선택

TDG 클러스터링에서 가장 중요한 요소 중의 하나는, “K-means 클러스터링 단계에서 K를 어떤 값으로 선택할 것인가?”이다. K-means 클러스터링에서 K를 어떤 값으로 선택하는 것이 적절한지는 명확히 밝혀져 있지 않다[19]. 결국 각각의 경우에 따라 적정한 K를 선택해야 한다. 본 연구에서 K를 3, 4, 5 중에서 어떤 것으로 선택할지 판단할 결과 최종적으로 K를 3으로 선택하게 되었다. K를 4나 5로 선택했을 때, 정상일 가능성이 높은 노드들까지도 비정상 TDG 클러스터로 분류되었기 때문이다. 그림 13과 14는 K=4일 때와 K=5일 때 탐지된 비정상 TDG 클러스터를 나타낸 것이다. K=4일 때와 K=5일 때, 탐지된 결과는 동일하게 나타났다. 그림 13(a)와 그림 14(a)에서는 15개의 비정상 노드가 탐지되었는데, 실제로 웜 바이러스에 감염된 호스트의 호스트 스캔 트래픽이 가장 강하게 나타나는 것들이었다. 그러나 그림 13(b)와 14(b)에 나타난 TDG 클러스터의 경우에는 정상 노드까지 포함하고 있다. 표 6을 보면 이러한 문제점을 명확하게 확인할 수 있다. 비정상으로 의심되는 경우는 전체의 54.23%에 그쳤으며, 45.77%를 차지하는 기타의 경우, 80, 8080 등과 같은 정상 포트가 많이 나타나고 있었다.



(a) TDG 클러스터0 (15개 노드) (b) TDG 클러스터2 (189개 노드)

그림 13 K=4일 때 탐지된 비정상 TDG 클러스터



(a) TDG 클러스터0 (15개 노드) (b) TDG 클러스터3 (189개 노드)

그림 14 K=5일 때 탐지된 비정상 TDG 클러스터

표 6 K=4일 때 TDG 클러스터2와 K=5일 때 TDG 클러스터3의 비정상 포트 특성

순위	포트	개수	비율	누적
1	135	757	50.94%	50.94%
2	445	49	3.29%	54.23%
3	5554	0	0.0%	54.23%
4	9898	0	0.0%	54.23%
5	17300	0	0.0%	54.23%
6	기타	680	45.77%	100%

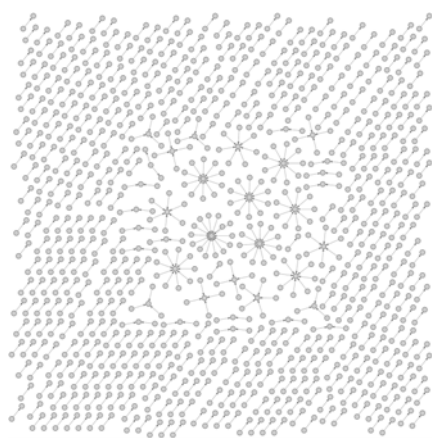


그림 15 K=4 또는 K=5일 때 비정상 TDG 클러스터를 삭제한 TDG

그림 15는 K=4나 K=5일 때, 비정상 TDG 클러스터를 제거한 TDG이다. 그림 15와 그림 11(K=3일 때, 비정상 TDG 클러스터를 제거한 TDG)을 비교하면, 그림 15에서 더 많은 노드들이 제거된 것을 확인할 수 있다. 문제는 제거된 노드의 약 45.77%가 정상일 가능성이 높다는 것이다.

식 (3)의 가중치를 조절할 경우, 표 6과 그림 15에 나타난 결과보다는 좋은 결과를 얻을 수는 있으나, 결국 K=3일 때가 가장 적당하다고 판단하게 되었다. K=3일 때 가장 좋은 결과를 얻을 수 있었기 때문이다.

5. 결 론

인터넷의 이상 현상을 탐지하는 것은 앞으로도 끝없이 연구되어야 할 과제다. 인터넷은 태생적 한계(의도적인 악용을 감안하지 않고 설계되었던 문제)를 가지고 있고, 공격자들은 그러한 약점을 파고들기 때문이다. 이에 대응하기 위해, 인터넷 측정 분야의 연구자들도 심도있는 연구를 통해 새로운 분석 및 탐지 기법을 제시하고 있다. 최근에는 트래픽 분산 그래프(TDG, Traffic Dispersion Graph)를 이용한 새로운 방식의 인터넷 측정을 제안한 연구가 발표되었다. 본 연구에서는 새롭게 제안된 TDG 기법을 우리나라에서 수집된 트래픽을 대상으로, 응용층 프로토콜(HTTP, DNS, P2P)이 어떤 형태로 동작하는지 적용해 보았다. 특히 본 연구에서는 이상 탐지에 대한 새로운 패러다임을 제시하였다. 이상 트래픽 자체를 찾는 것에 관심을 두기 보다는, 비정상적 상호작용을 보이는 이상 호스트는 찾는 방식을 제안한 것이다. 이를 실체화하기 위해 TDG를 활용한 TDG 클러스터링 기법을 제안하였다. 제안한 기법은, 다양한 웜 바이러스에 감염된 호스트를 정확히 분류해 내었다. TDG 클러스터링 기법은 TDG로 표현된 데이터에 대한 3개의 변수(Indegree 엔트로피, Outdegree 엔트로피, 패킷 개수)를 추출하여 K-means 클러스터링을 활용한 기법이다. 본 연구에서 제안한 기법은 98~99%정도의 이상 호스트 탐지 성능을 보였다. 또한 5초간의 트래픽에 대한 연산 속도가 1~2초 내외로 나타났다. 이는 실시간 이상 탐지에서도 활용 가능할 정도로 매우 빠른 성능이다. 본 연구에서 제안한 TDG 클러스터링 기법의 활용은 아직 시작일 뿐, 더욱 심도있는 연구를 통해 앞으로도 다양한 이상 탐지에 응용할 수 있을 것으로 기대한다.

참 고 문 헌

[1] R. R. Panko, "Corporate Computer and Network Security," Prentice Hall, 2004.

[2] M. Crovella and B. Krishnamurthy, "Internet Measurement: Infrastructure, Traffic, and Applications," John Wiley & Sons, Ltd, 2006.

[3] D. Moore and G. M. Voelker and S. Savage, "Inferring Internet Denial-of-Service Activity," In Proc. of Usenix Security Symposium, pp. 9-22 Washington, DC, August 2001.

[4] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS attack and DDoS defense Mechanisms," ACM SIGCOMM Computer Communication Review, Vol. 34, Issue 2, pp. 39-53, April 2004.

[5] Marios Iliofotou, et al., "Network Monitoring using Traffic Dispersion Graphs (TDGs)," In Proc. ACM Internet Measurement Conference, pp. 315-320, San Diego, California, USA, October 2007.

[6] CERT Advisory MS-SQL Server Worm, "http://www.cert.org/advisories/CA-2003-04.html," January 2003.

[7] CERT Advisory W32/Blaster worm, "http://www.cert.org/advisories/CA-2003-20.html," August 2003.

[8] D. Moore and V. Paxson and S. Savage and S. Staniford and N. Weaver, "Inside the Slammer worm," IEEE Security & Privacy, Vol. 1 issue 4, pp. 33-39, August 2003.

[9] D. Brauckhoff and B. Tellenbach and A. Wagner and M. May and A. Lakhina, "Impact of Packet Sampling on Anomaly Detection Metrics," In Proc. ACM Internet Measurement Conference, pp. 159-164, Rio de Janeriro Brazil, October 2006.

[10] Graphviz - Graph Visualization Software, "www.graphviz.org,"

[11] TCPDUMP/LIBPCAP public repository, "http://tcpdump.org"

[12] MySQL: The world's most popular open source database, "http://www.mysql.com"

[13] Nick Duffield, "Sampling for Passive Internet Measurement: A Review," Statistical Science Vol. 19, No. 3, pp. 472-498, 2004.

[14] Juniper Traffic Sampling, "http://www.juniper.net/techpubs/software/junos/junos60/swconfig60-policy/html/sampling-overview.html"

[15] SAS - Business Intelligence Software and Predictive Analytics, "http://www.sas.com"

[16] J. Kim and S. Ahn and Y. Won, "Mining An Anomaly: On The Small Time Scale Behavior of The Traffic Anomaly," In Proc. of IADIS International Conference WWW/Internet, Murcia, Spain, pp. 552-559, October 2006.

[17] T. M. Cover and J. A. Thomas, "Elements of Information Theory," Wiley Interscience, 1991.

[18] A. Lakhina and M. Crovella and C. Diot, "Mining Anomalies using Traffic Feature Distributions," ACM SIGCOMM Computer Communication Review, Vol. 35, Issue 4, pp. 217-228, October 2005.

[19] P. Tan and M. Steinbach and V. Kumar, "Introduction to Data Mining," Addison Wesley, 2006.

김 정 현

2004년 명지대학교 컴퓨터공학과 학사 졸업. 2008년 한양대학교 전자컴퓨터통신 공학과 박사 수료. 관심분야는 인터넷 측정, 이상탐지, 운영체제

원 유 집

1990년 서울대학교 계산통계학과 학사 졸업. 1992년 서울대학교 계산통계학과 석사 졸업. 1997년 University of Minnesota 박사 졸업. 1997년~1999년 Intel 연구원. 1999년~현재 한양대학교 전자컴퓨터통신공학과 부교수. 관심분야는 운영체제, 컴퓨터네트워크, 성능평가

안 수 한

1992년 서울대학교 계산통계학과 학사 졸업. 1994년 서울대학교 계산통계학과 석사 졸업. 2000년 서울대학교 통계학과 박사 졸업. 2001년~2003년 AT&T Labs-Research, Post-Doc, Consultant 2004년~현재 서울시립대학교 통계학과 부교수. 관심분야는 Fluid Flow Model, Queueing, Telecommunication Network