Libpcap를 이용한 Cacti기반 네트워크 트래픽 모니터링 시스템

이성옥* · 주강** · 정회경***

Cacti-based Network Traffic Monitoring System Using Libpcap

Sung-Ock Lee* · Zhu Jiang** · Hoe-Kyung Jung***

요 약

네트워크 기술이 빠르게 성장하고 있어서 네트워크 환경이 복잡해지고 있다. 이에 따라, 네트워크 트래픽이나 정보를 이용하여 실시간으로 자원을 모니터링 하는 기술들이 발전하고 있다. 대표적인 모니터링 툴은 Cacti이다. Cacti는 RRDTool(Round Robin Database tool), SNMP(Simple Network Management Protocol)를 기반으로 한 모니터링 툴 이고, Libpcap는 네트워크 카드에서 패킷 캡쳐를 용이하게 해주는 라이브러리이다. 본 논문에서는 Cacti와 Libpcap 기반으로 시스템을 개발하여 실시간으로 대상을 모니터링 할 수 있고 스마트폰으로 실시간 알림 이메일을 받을 수 있다. 본 시스템은 Libpcap으로 포착한 네트워크 트래픽 패킷을 분석하고 그래프 형식으로 Cacti에서 표현되어 모니터링 할 수 있다. 이는 높은 효율성을 가지며 관리가 간편하고 정확성을 가지므로, 향후 널리 활용될 것으로 보인다.

ABSTRACT

For network is growing at a rapid rate, network environment is more complex. The technology of using network traffic to monitor our network in real-time is developed. Cacti is a representative monitoring tool which based on RRDTool(Round Robin Database tool), SNMP(Simple Network Management Protocol). In this paper, it show you how to develop a system which based on Cacti and Libpcap to monitor our monitored objects. At this system, using Libpcap to capture network traffic packets, analyze these packets and then turn out in Cacti in graphical form. So as to achieve monitoring system. This system's execution is efficient and the management is easy and the results are accurate, so it can be widely utilized in the future.

키워드

Cacti, 네트워크 트래픽, Libpcap, 트래픽, 모니터링, 실시간, RRDTool

Key Word

Cacti, Network traffic, Libpcap, Traffic, Monitoring, Real-time, RRDTool

* 정회원 : 배재대학교
** 준회원 : 배재대학교
*** 종신회원 : 배재대학교 (교신저자, hkjung@pcu.ac.kr)

접수일자 : 2012. 06. 28
심사완료일자 : 2012. 07. 17

Open Access http://dx.doi.org/10.6109/jkiice.2012.16.8.1613

Ⅰ. 서 론

정보화가 발전하고 네트워크 영역이 점점 더 넓어지고 네트워크 기술이 발전함에 따라, 네트워크 구조도 복잡해지고 있다. 네트워크에서 트래픽은 작은 변화에도 네트워크 안전과 애플리케이션에 큰 영향을 끼친다. 이에 따라, 네트워크의 동작 상태를 전체적으로 분석하여 네트워크의 운영 안전, 운영 상태 파악, 미리 예측, 관리 기능을 함으로써, 큰 도움을 줄 것이다[1].

본 논문에서는 1분마다 네트워크 카드에서 패킷을 캡처하고 캡처된 데이터가 쉘(shell) 프로그램을 통하여 패킷들의 정보들을 통계 및 분석할 수 있는 시스템을 제안한다. 이는 수집된 데이터를 이용하여 Cacti에서 그래프로 표현한다. 만약 어떤 패킷의 수량이 미리 설정된 값보다 크면 시스템에서 자동적으로 관리자에게 실시간으로 알림 이메일을 보낸다.

이 시스템은 설치가 간단하고 결과가 정확하고 고효율적이며, 실시간성이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 Cacti와 Libpcap를 기술한다. 3장에서는 시스템 환경 및 구조 설계에 대해 설명한다. 4장에서는 실험에서 나타난 결과를 분석하고 결론 및 향후 연구과제는 5장에 기술한다.

Ⅱ. 관련 연구

2.1. Cacti 개요

본 절은 Cacti의 개념과 운영 원리를 설명하고 어떻게 설치하는지 설명한다.

2.1.1. RRD(Round Robin Database)

RRD는 네트워크 대역폭, 기계실 온도, 서버의 평균 부하 등과 같은 시간대별 데이터를 저장하고 표시하기 위한 시스템이다. RRD는 매우 간결한 방법으로 데이터를 저장하므로, 시간의 흐름에 따라 데이터 양이 그리 크게 늘어나지 않는다.

RRD는 항상 일정한 데이터 밀도를 강제로 유지하기 위해 데이터를 처리함으로써, 유용한 그래프를 제공한다. 이를 위해서는 쉘이나 펄(perl) 등으로 만들어진 단순한 래퍼 스크립트나, 또는 네트워크 장비에 주기적으로 질의를 던지고 편리한 인터페이스를 제공하는 프론트엔드의 이용 등 어떤 방식이라도 이용할 수 있다[2].

2.1.2. Cacti

Cacti는 RRDTool의 데이터베이스를 이용한 웹 그래픽 생성엔진이다. Cacti는 각종 데이터를 MySQL 등의 데이터베이스에 시스템 상황 등의 데이터 값을 저장하고 시간단위로 저장된 데이터를 분석하여 웹상에서 그래프를 생성하여 보여주는 매우 유용한 시스템 모니터링 도구이다[3]. 그림 1은 Cacti의 작업 원리를 표현한다.

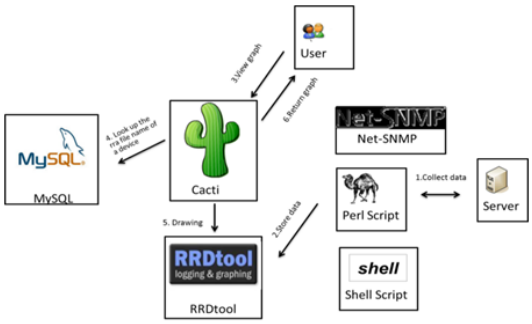

그림 1. Cacti 실행 원리
Fig. 1 Cacti Work Principle

2.1.3. PIA(Plugin Architecture) 추가

기본 Cacti는 그다지 기능이 많지 않으나 플러그인 하여 PIA 패키지를 사용 할 수 있다.

먼저 PIA 파일을 다운로드 받아 SQL 파일을 수집한다. 다음에 global.php 파일을 수정한다. 이 파일은 var/www/html/include에 있으며, $plugins=array()를 찾고 그 뒤에 “$plugins[]=’setting’;”과 “$plugins[]=’thold’;”를 추가하면 된다. 그 후 Cacti 메인 페이지에서 “console”를 클릭하고 “plugin management” 에서 플러그인들을 설치하면 된다.

2.2. Libpcap

패킷 캡처란 네트워크 상의 패킷을 볼 수 있다는 것을 의미한다. 일반적인 인터넷 환경에서 라우터는 내부 네트워크로 향하는 패킷을 브로드캐스팅하게 되고 각 컴퓨터들은 자신의 인터페이스에 들어오는 패킷 중 목적지가 자신인 경우에만 받아들여 이를 운영체제가 처

리한다. 패킷 캡처는 이처럼 자신에게 전달되는 패킷을 받아들여 패킷의 내용을 확인할 수 있음을 의미한다. 이러한 패킷 캡처는 네트워크의 사용에 대한 통계나 보안을 목적으로 하는 모니터링 하고, 네트워크를 디버깅하기 위한 것, 스니퍼링 하는 것 등 다양한 형태로 응용이 가능하다[4,5].

Libpcap은 이러한 패킷 캡처를 용이하게 해주는 라이브러리이다. 각 운영체제 벤더들이 각각의 패킷 캡처 도구들을 제공하고 있어 개발이나 포팅 등에 어려움이 있기 때문에 각 도구들의 기능을 포함하면서 시스템에 독립적인 Libpcap이 등장하게 되었다.

그림 2는 Libpcap를 이용하고 네트워크 카드가 패킷을 포착하는 과정이다.

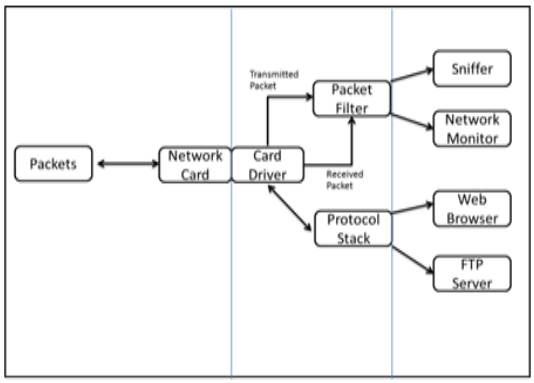

그림 2. 패킷 처리 과정
Fig. 2 The Capture Process

Ⅲ. 시스템 설계

3.1. 시스템 설계 및 환경

본 시스템의 설계 원리는 여러 가지 호스트가 모니터링 대상을 방문하고 1분마다 이 방문 수를 기록하고 이 방문 수는 임의 처리 방식을 이용하여 데이터 정보를 얻을 수 있다. 이 데이터 정보를 통계하고 이 데이터를 이용하여 그래픽 형태로 Cacti에서 표현한다. 만약 어떤 패킷의 수가 미리 설정된 값보다 크면 시스템은 관리자에게 이메일로 보내고 경고를 나타내어 실시간으로 목적 대상을 모니터링 할 수 있다. 이 설계 원리는 그림 3과 같다. 표 1은 시스템 설계 환경을 기술한다.

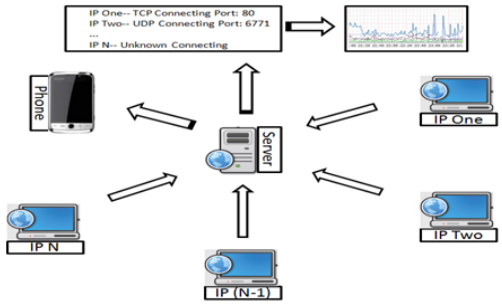

그림 3. 시스템 설계
Fig. 3 Design of System

표 1. 시스템 설계 환경
Table. 1 Environment of System Design

개발 환경	비고
개발 시스템	CentOS 5.4
개발 언어	C, Perl, Shell
개발 소프트웨어	Cacti 0.8.7.C, PHP 5.1.6, MySQL 5.0.82SP1, RRDTool 1.2.23, Apache 2.0.63

3.2. 시스템 구조

본 시스템은 그림 4와 같은 다섯 부분으로 구성되어 있다. 첫 번째 부분은 패킷을 캡처한 부분이며, 두 번째 부분은 패킷의 통계 부분이며, 세 번째 부분은 데이터 쿼리 부분이고, 네 번째 부분은 Cacti를 연결하는 부분이며, 마지막으로 경고하는 부분이다. 패킷을 캡처하는 부분은 네트워크 카드에 지나는 모든 패킷을 캡처 하는 과정으로, 이 과정에서 생긴 데이터 흐름은 두 번째 부분에서 통계를 구하고, 세 번째 부분에서 지난 데이터 흐름을 하루 동안 파일에 백업한다. Cacti로의 연결 부분은 펄 스크립트로 만들고, Cacti에서 data input method를 설치한 후에 모니터닝 대상을 추가하면 된다. 이메일 알림부분은 Cacti의 'Alerting/Thold'을 이용하고 이메일을 보낸다.

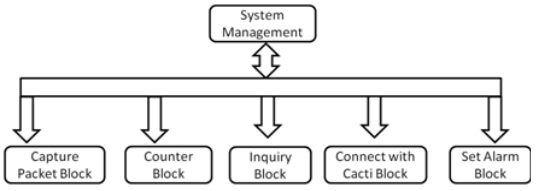

그림 4. 시스템 구조
Fig. 4 Structure of System

3.2.1. 패킷 블록 캡처

패킷 캡처 부분으로 여기에서는 네트워크 스니퍼를 만드는데, 이것을 만들기 위해서는 다섯 가지를 해야 한다. 첫 번째 디바이스를 얻고, 두 번째는 이 디바이스를 연다. 세 번째는 데이터 링크 유형을 설정하고, 넷 번째는 패킷 캡처를 시작한다. 다섯 번째는 캡처한 패킷을 분석한다. 본 시스템에서 캡처한 패킷의 정보를 이용하여 특정한 함수를 만든다.

이 함수를 통하여 IP(Internet Protocol), ARP(Address Resolution Protocol), RARP(Reverse Address Resolution Protocol), SNMP(Simple Network Management Protocol)로 유형을 구분하고, 그 외의 패킷 정보는 “UNKNOWN” 유형으로 구분한다. 이를 통해 IP와 ARP 패킷의 수신과 송신을 알 수 있다[6]. IP 패킷에서는 TCP(Transport Control protocol) 패킷, UDP(User Datagram Protocol) 패킷, ICMP(Internet Control Message Protocol)패킷, IGMP(Internet Group Management Protocol) 패킷들을 포함하는데 남은 것은 “UNKNOWN” 유형으로 한다. 그림 5은 함수 처리 유형이다[6].

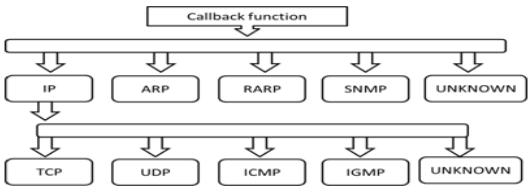

그림 5. 함수 처리 유형
Fig. 5 Type of Function Process

모두 설치한 후 스니퍼를 실행하면 그림 6처럼 결과가 나타난다. 이 결과를 보면 캡처 시간, 패킷 크기, 유형, 원 주소, 목표 주소, 통신 포트 등이 나타난다.

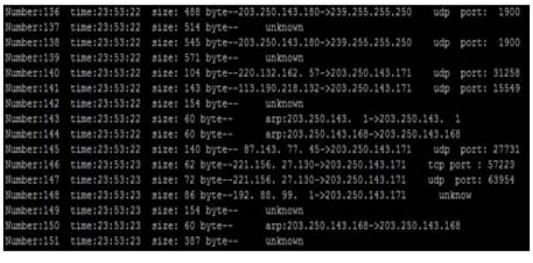

그림 6. 캡처한 패킷
Fig. 6 Captured Packets

3.2.2. 카운터 블록

이 부분은 통계 부분이다. 패킷 캡처의 시간은 1분마다 수행하고, 이 시간 후에 캡처한 패킷을 Start_File에 기록한다. 그림 7은 쉘 프로그램이 캡처 한 패킷을 처리하는 과정이다. 먼저 쉘 프로그램은 Start_File의 내용을 Copy_File, Log_File에 복사하고, 다음에 Copy_File에 임의 유형 패킷을 통계하고 그 결과는 몇 개 있는지 Result.h 파일에 저장한다.

예를 들면, ARP, RARP, UDP, TCP, ICMP, IGMP등 패킷이 몇 개가 있는지 그 결과를 Result.h에 저장한다. 이 과정을 수행하고 1분 후에는 Copy_File 내용을 토대로 Result.h 파일 내용을 다시 수정한다. 이 과정을 하루 동안 계속 반복한다.

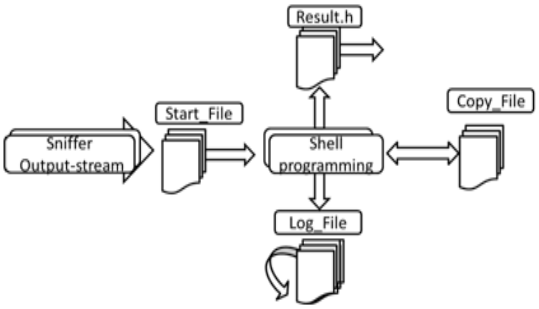

그림 7. 쉘 프로그램 처리
Fig. 7 Process of Shell Program

3.2.3. 질의 불록

이 부분은 조회 부분이다. 만약 나타난 결과가 이상이 있다면 그 결과를 재조회해야 한다. 앞서 언급한 Log_File이 조회대상이 된다. 이 Log_File은 시간에 의하여 조회 할 수 있다. 단, 이 시간은 하루 안에 조회 할 수 있고 하루 지나면 Log_File이 삭제되기 때문에 조회할 수 없다.

3.2.4. Cacti 연결

이 부분은 Cacti로의 연결 부분으로 테이블을 만든 후, 앞서 말한 Result.h 파일을 프로그램 상에서 사용한다. 이를 통해 그림 8과 같은 테이블 구조에 저장한다. 본 시스템을 통해 펄 프로그램을 이용하여 DB 내에 있는 데이터를 얻고, 다음에 Cacti에 전달한다. 마지막은 Cacti에서 “data input method”를 만들고 설치하면 된다.

```
mysql> desc monitoring;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| interface | varchar(10) | YES  |     | NULL    |       |
| arp       | int(20)     | YES  |     | NULL    |       |
| rarp      | int(20)     | YES  |     | NULL    |       |
| udp       | int(20)     | YES  |     | NULL    |       |
| tcp       | int(20)     | YES  |     | NULL    |       |
| icmp      | int(20)     | YES  |     | NULL    |       |
| igmp      | int(20)     | YES  |     | NULL    |       |
| unknown   | int(20)     | YES  |     | NULL    |       |
| snmp      | int(20)     | YES  |     | NULL    |       |
| total     | int(20)     | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

그림 8. DB 테이블 구조
Fig. 8 Structure of DB Table

3.2.5. 알람 설정

이 부분은 알림부분이다. 만약 얻은 데이터 값이 설정된 값보다 더 크거나 적으면 이 시스템은 알림 이메일을 보낸다. 특정 값과 이메일 엔진은 Cacti에서 설정할 수 있다.

Ⅳ. 실험 결과 및 분석

4.1. 실험

PuTTy는 telnet, rlogin, SSH(Secure SHell) 프로토콜을 이용하며 윈도우에서 리눅스 서버로 로그인 할 수 있는 원격 터미널 프로그램이다[7]. 그림 9는 Cacti가 15초마다 DB에서 데이터를 얻어 나타내는 그래프의 한 과정의 화면이다. 그림 10은 Log_File 내에 있는 내용이고, 그림 11은 알림 이메일, 그림 12는 Cacti에 나타낸 그래프이다.

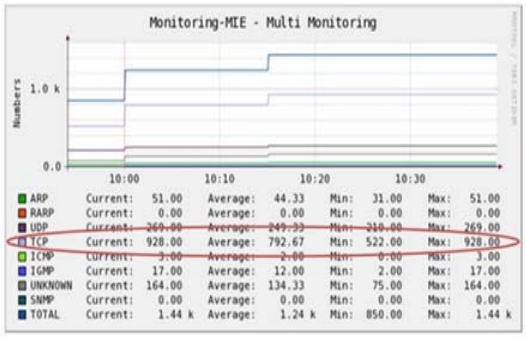

그림 9. 실시간 그래프
Fig. 9 Real Time Graph

4.2. 실험 분석

PuTTy는 신뢰할 수 있는 TCP 프로토콜 기반인 클라이언트 터미널 프로그램이고, SSH 프로토콜의 통신 포트는 22이다[8]. 그리고, 실험 결과 정보들을 검토해보고 그 결과를 통하여 본 시스템의 신뢰성, 정확성 및 실시간성을 확인 할 수 있었다.

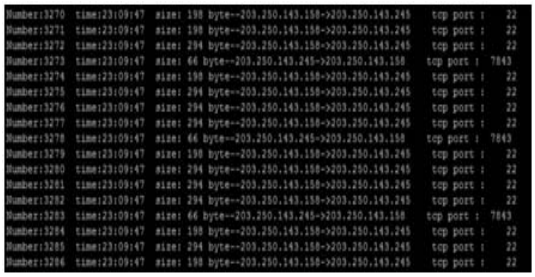

그림 10. Log_File 내용
Fig. 10 Contents of Log_File

From: **admin** <xiaodongyixin@qq.com>
Date: 2011/5/19
Subject: Monitoring-MIE - Multi Monitoring [TCP] went above threshold of 100 with 109
To: xiao58588395@gmail.com

An alert has been issued that requires your attention.

Host: Monitoring-MIE (203.250.143.158)
203.250.143.158: http://203.250.143.158///graph.php?local_graph_id=101&rra_id=1
Message: Monitoring-MIE - Multi Monitoring [TCP] went above threshold of 800 with 928

그림 11. 알람 이메일
Fig. 11 Alarming Email

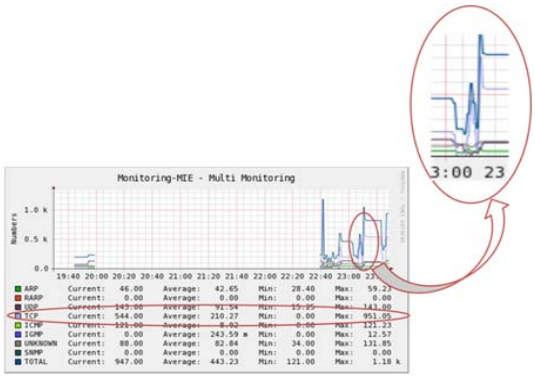

그림 12. TCP에 대한 그래프
Fig. 12 Graph for TCP

V. 결론 및 향후 연구 과제

본 논문의 시스템은 Cacti 기반으로 설계되었다. 본 시스템은 사용이 쉽고 또한, 데이터 저장이 RRD 방식을 사용한다. 그리고, 결과의 정확성, 그래픽의 직관성, 안전의 실시간성을 가지고 있다. 따라서, 네트워크 환경 모니터링을 위해 널리 활용될 것으로 판단된다.

향후, 연구 과제로는 로그 파일 확인을 용이하게 하고, 이더넷 유형 이외도 고려한 복잡한 네트워크 환경을 고려하여 연구되어야 할 것이다.

참고문헌

[1] Myung-Sup Kim, Yong J.Won, and James Won-Ki Hong, “Application-Level Traffic Monitoring and an Analysis on IP Networks”, ETRI Journal, Volume 27, Number 1, February 2005

[2] http://www.mrtg.org/rrdtool, 2011.05

[3] http://www.cacti.net/features.php, 2011.05

[4] 노광민, “리눅스에서 pcap library를 사용하여 패킷을 잡아보기 v0.3”, 2000.09.14

[5] 강승일, “Packet Capture using Libpcap”, 2006.03.10

[6] Luis Martin Garcia, “Programming with Libpcap” http://undergraduate.csse.uwa.edu.au/units/CITS3231/reading/libpcap-programming.pdf

[7] en.wikipedia.org/wiki/PuTTY, 2011.05

[8] en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers, 2011.05

저자소개

이성옥(Sung-Ock Lee)

2008 한양대 영미언어문화학 학사
2012 고려대 영어교육학 석사
2012～현재: 배재대학교 컴퓨터공학 박사과정

※관심분야 : 멀티미디어정보처리, USN, Ubiquitous Computing

주강(Zhu Jiang)

2012년 배재대학교 컴퓨터공학 학사
2012년～현재 배재대학교 컴퓨터공학 석사과정

※관심분야 : 웹 서비스, 유저인터페이스, 모바일 컴퓨팅, HTML5

정회경(Hoe-Kyung Jung)

1985년 광운대학교 컴퓨터공학과(공학사)
1987년 광운대학교 컴퓨터공학과(공학석사)
1993년 광운대학교 컴퓨터공학과(공학박사)
1994년～현재 배재대학교 컴퓨터공학과 교수

※관심분야 : 멀티미디어 문서정보처리, XML, SVG, Web Services, Semantic Web, MPEG-21, Ubiquitous Computing, USN