Real-time traffic analysis in Ethernet

T. Kováčik, I. Kotuliak, P. Podhradský
Slovak University of Technology
Ilkovičova 3, 812 19 Bratislava, Slovakia
Phone: (421) 2-68279 410 Fax: (421) 2-68279 601 E-mail: Ivan.Kotuliak@stuba.sk

Keywords: Ethernet, traffic traces, network performance, pcap

Abstract – This article presents our framework intended for measurement of the Ethernet traffic, its evaluation and finally graphical representation. Whole project consists of three main parts network measurement, trace analysis and user interface.

1. INTRODUCTION

Considering the fact that the most of today's computers are interconnected and the fact that network resources and connection to the Internet are part of business expenses, the task to measure and analyze type and amount of the network traffic becomes more and more relevant. In smaller companies, the evaluation of network load can be used to monitor the activity of employees, efficiency of using a connection to the Internet and the investigation of possible security issues. In this way, we can monitor whether users do not use unauthorized software (e.g. instant messaging software), do not download high amount of data or do not use the network for harming the interests of the company [1]. For security reasons, we can monitor the network to check whether it was not affected by some attack and misused for example for distribution of spam emails. Another reason to monitor the load on the network is to save network resources used by application which we develop. This way we can optimize our application and save some bandwidth for more important usage [2]. The monitoring can be also part of network audit and administrators can use it while planning network upgrade.

We have implemented new network analyzer called KaTaLyzer. The focus of the KataLyzer is given on monitoring of network traffic depending on IP and MAC addresses, quantity of traffic and frame counts from respective addresses and protocol analysis. Obtaining all these information, we can present data in the form of the graph depending on the network address or protocol. Even more, it is obvious to identify who consumes most of the bandwidth.

Our analyzer is programmed and compiled under Linux operating system which was chosen for its security, stability, well developed implementation of C programming language, ease of configuration of web server and last but not least for its zero price.

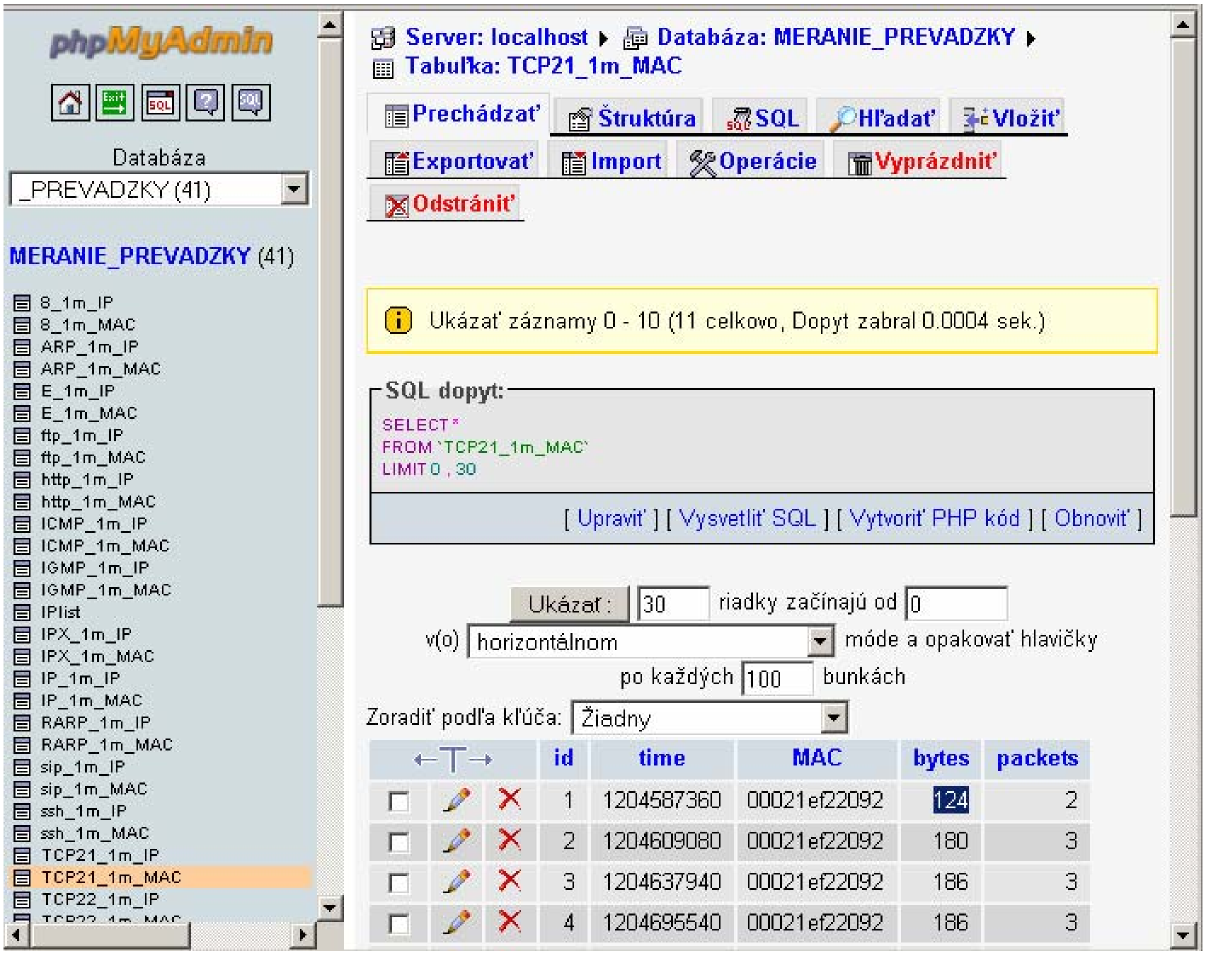
There are several network monitoring projects (for windows and linux), e.g. NetFlow Analyzer [3], NetFlow Traffic Analyzer [4], ntop [5], iptraf [6], IP Traffic Monitor [7], Zabbix [8] and many others. They are either oriented on very small part of functionality we provide (e.g. drawing only summary in/out graphs, not traffic for each address, showing only non-graphical information, no support for some protocols,…). Other program are rather complex or are distributed under proprietary licenses. Up to our knowledge, there is no application, which would show graphs for freely configurable protocols for each IP/ MAC address, which would allow rewieving data from history, which would be available from any part of the world thanks to its web interface, which would support graphs export and had so many features that our project does have.

In this article, firstly we will describe functionality of the KaTaLyzer. Third Section focuses on more detailed information about functionality. In the Section 4, we present achieved results while presenting graphical web page with traffic graphs. Concluding remarks and open questions for the future work are given in the last Section.

2. KaTaLyzer framework

KaTaLyzer is a continuation of project started in 2004. Anyway there is not much left from original analyzator as it used to have many functionalities, e.g. capturing traffic into dump files, offline analysis of captured data, detail information mode to show and inspect each frame details, use packet filters in case we are interested only in some type of traffic, put brief information into simple statistics html file and online analysis mode in which we are able to analyze each frame, adjust statistics according to captured information and evaluate it in graphical form. The last mentioned functionality is now the core of new analyzer. We decided to divide original project into two independent parts - the first one which is focused on capturing traffic information into dump files, offline analysis and statistical evaluation of captured data and the second one on which this article is focused. It is oriented on online traffic analysis and graphical interpretation of measured data.

As we have already mentioned, we are focused on speed because online analysis of each frame is very difficult and therefore consumes a lot of computer's performance. As



Server: localhost ▸ Databáza: MERANIE_PREVADZKY ▸
Tabuľka: TCP21_1m_MAC

Ukázať záznamy 0 - 10 (11 celkovo, Dopyt zabral 0.0004 sek.)

SQL dopyt:

```
SELECT *
FROM `TCP21_1m_MAC`
LIMIT 0 , 30
```

Ukázať : 30 riadky začínajú od 0

v(o) horizontálnom móde a opakovať hlavičky

po každých 100 bunkách

Zoradiť podľa kľúča: Žiadny

←T→			id	time	MAC	bytes	packets
□	✎	✕	1	1204587360	00021ef22092	124	2
□	✎	✕	2	1204609080	00021ef22092	180	3
□	✎	✕	3	1204637940	00021ef22092	186	3
□	✎	✕	4	1204695540	00021ef22092	186	3

Fig. 1. Table example containing statistics of TCP port 21

C language is simple and fast we decided to implement capture, analysis and statistical part in it.

Main idea of analyzer is to capture a frame, analyze it Byte-by-Byte, adjust statistics and after chosen amount of time send summary information into MySQL database. Capturing a frame is provided by pcap library [8] which has several functionalities according to network traffic analysis already implemented and is distributed under BSD license. The frame analysis is based on comparing particular bytes of the frame to values corresponding to various protocols. We can simply choose which protocols will be analyzed by setting measurement and analysis configuration in config file named my_config.conf. Here we can choose interface which we want to monitor, database information like name of the server, name of the database, user and password, number of seconds after which summary information is written into database and finally we can choose protocols which we are interested in and which we want to monitor. As a data storage we chose MySQL database as it is widely used, has great support, its installation and configuration is pretty easy and again, it is free to use and distribute. Another reason to choose MySQL is that web page which uses data from the database to display monitoring results is written in PHP and it has very good support for MySQL.

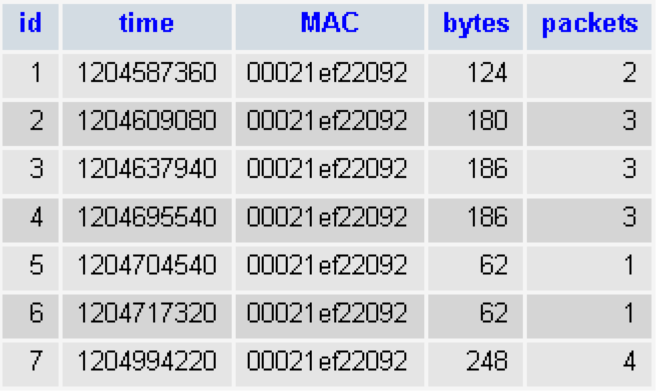
2.1 Analyzer

Start of the analyzer is possible only by user root. If this condition is not met analyzer is not able to open network adapter specified in configuration file as only user root can do that and it ends with error message: ”Error opening adapter”. After opening the adapter analyzer waits until new minute begins according to system time. It means that it does not start measuring traffic for example in the middle of actual minute in order to measure real values and not only partial minute traffic.
The main task of analyzer is to make a loop in program which checks specified network interface for received or sent data and runs pcap_handler callback function. In this function we do our analysis and statistics adjustment. It is simple process which checks which protocols were defined to be monitored in configuration file. We can chose which protocols we do not want to monitor - ETHERNET II, IEEE802.3, ARP, RARP, IP, IGMP, ICMP, IPX, TCP and UDP(default setting is to monitor all implemented protocols, setting appropriate value to 0 will disable monitoring the protocol). We can also specify TCP and UDP port numbers which we want to monitor.

After specified amount of time (which is read from configuration file) statistic data is sent to MySQL database. Each monitored protocol has its own database table which is checked to exist before sending data into it. If it does not exist it is created. Table name consists from protocol identificator (read from configuration time), amount of time between two database updates (also read from configuration time) and identificator MAC or IP which indicates whether the table consists information about MAC or IP addresses (RM OSI layer 2 vs. RM OSI layer 3 address). Example of table name is `TCP21_60_MAC` which means that we are interested in TCP port with number 21, time interval between two database updates is 60 seconds and the table stores MAC addresses which sent data through TCP port with number 21. Data is sent into the database in multiple updates as follows:

```
INSERT INTO TCP21_60_MAC (time, MAC,
bytes, packets) VALUES
('1204587360','00021ef22092','124','22'),
('1204587360','00021ef22093','114','12'),
('1204587360','00021ef22094','104','8');
```

This command inserts 3 rows into table `TCP21_60_MAC` with information captured in minute `1204587360` (which is unix time format) which says that mac address `00021ef22092` sent 124 Bytes in 22 frames, mac address `00021ef22093` sent 114 Bytes in 12 frames and mac address `00021ef22094` sent 104 Bytes in 8 frames.



id	time	MAC	bytes	packets
1	1204587360	00021ef22092	124	2
2	1204609080	00021ef22092	180	3
3	1204637940	00021ef22092	186	3
4	1204695540	00021ef22092	186	3
5	1204704540	00021ef22092	62	1
6	1204717320	00021ef22092	62	1
7	1204994220	00021ef22092	248	4

Fig. 2. Detail information from table collecting data about TCP protocol – port 21 traffic from particular MAC addresses

After inserting values into protocol tables we put MAC and IP addresses into pairs and send them into table named IPlist. If the table already contains such IP address we update particular table entry with new values. This table is used for obtaining additional information about status of the network which we monitor.

3. Project results

Analyzer is only one part of the project – a computing part. As we need to display results of capturing, analysis and statistic information, we decided to prepare simple web page which shows data depending on criteria which we choose. This page uses same configuration file as computing part. Depending on it the web page displays only protocols which we decided to monitor.

After making our choice about protocol we want to display we need to choose which addresses we are interested in – IP or MAC ones, than we choose whether we want to display information about number of frames or information about amount of traffic (in kibit/s, Mibit/s…) in graph and than we choose date and time we want traffic to be displayed in. We can also choose whether we want to display information about all traffic which passed the interface or we can choose specific IP/MAC address which generated the traffic. When we submit our request we are given online generated graph as following example.

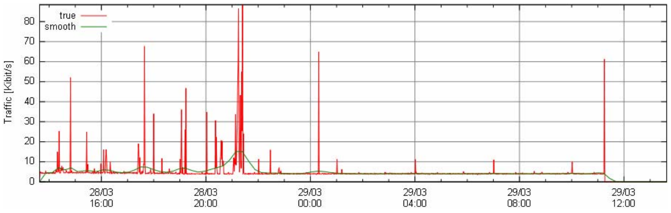


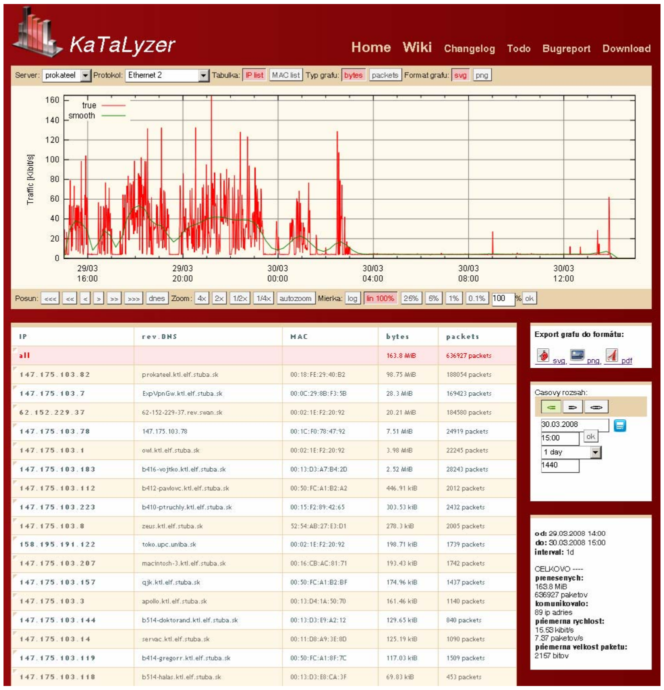
Fig. 3. Example of graph of traffic (IP traffic)

If we want to zoom some part of the graph we can do it in both axes – in time and also in value.

There are some more information displayed on the web page e.g. how many MiB were transferred in how many frames, how many IP addresses communicated, what was average speed – in kibit/s and frames/s and how many bits had average frame. In the lower part of the web page there are displayed individual IP addresses with their DNS equivalents, MAC address from which it was captured, amount of bytes and number of frames that came from this particular address. Clicking on single IP address we can display graph of the traffic coming from this address and investigate it more deeply.

In upper part of the web page we can find shortcuts to WIKI pages: Changelog, Todo, Bugreport and Download. These subpages serve for project development and to display more information about it. In lower part we can export displayed graph into one of following formats: SVG, PNG and PDF. These we can later use e.g. for already mentioned network audits or monitoring of users and proving their activities.

Actual testing page of our project is on [11], where anybody can find results of running measuring on one of our faculty servers.



Server: prokateel Protokol: Ethernet 2 Tabulka: IP list MAC list Typ grafu: bytes packets Format grafu: svg png

100 %

IP	rev.DNS	MAC	bytes	packets
all			163.8 MiB	636927 packets
147.175.103.82	prokateel.ktl.elf.stuba.sk	00:18:FE:29:40:B2	98.75 MiB	188054 packets
147.175.103.7	ExpVpnGw.ktl.elf.stuba.sk	00:0C:29:8B:F3:5B	28.3 MiB	169423 packets
62.152.229.37	62-152-229-37.rev.swan.sk	00:02:1E:F2:20:92	20.21 MiB	184580 packets
147.175.103.78	147.175.103.78	00:1C:F0:78:47:92	7.51 MiB	24919 packets
147.175.103.1	owl.ktl.elf.stuba.sk	00:02:1E:F2:20:92	3.98 MiB	22245 packets
147.175.103.183	b416-vojtko.ktl.elf.stuba.sk	00:13:D3:A7:B4:2D	2.52 MiB	28243 packets
147.175.103.112	b412-pavlovc.ktl.elf.stuba.sk	00:50:FC:A1:B2:A2	446.91 kiB	2012 packets
147.175.103.223	b410-ptruchly.ktl.elf.stuba.sk	00:15:F2:89:42:65	303.53 kiB	2432 packets
147.175.103.8	zeus.ktl.elf.stuba.sk	52:54:AB:27:E3:D1	278.3 kiB	2005 packets
158.195.191.122	toko.upc.uniba.sk	00:02:1E:F2:20:92	198.71 kiB	1739 packets
147.175.103.207	macintosh-3.ktl.elf.stuba.sk	00:16:CB:AC:81:71	193.43 kiB	1742 packets
147.175.103.157	qjk.ktl.elf.stuba.sk	00:50:FC:A1:B2:BF	174.96 kiB	1437 packets
147.175.103.3	apollo.ktl.elf.stuba.sk	00:13:D4:1A:50:70	161.46 kiB	1140 packets
147.175.103.144	b514-doktorand.ktl.elf.stuba.sk	00:13:D3:E9:A2:12	129.65 kiB	840 packets
147.175.103.14	servac.ktl.elf.stuba.sk	00:11:D8:A9:3E:8D	125.19 kiB	1090 packets
147.175.103.119	b414-gregorr.ktl.elf.stuba.sk	00:50:FC:A1:8F:7C	117.03 kiB	1509 packets
147.175.103.118	b514-halas.ktl.elf.stuba.sk	00:13:D3:E8:CA:3F	69.83 kiB	453 packets

Export grafu do formátu:

Casovy rozsah:
30.03.2008
15:00
1 day
1440

od: 29.03.2008 14:00
do: 30.03.2008 15:00
interval: 1d

CELKOVO ----
prenesenych:
163.8 MiB
636927 paketov
komunikovalo:
89 ip adries
priemerna rychlost:
15.53 kibit/s
7.37 paketov/s
priemerna velkost paketu:
2157 bitov

Fig. 4. Web page layout

4. Conclusion

The goal of our work was to create LAN traffic analysis software which would allow us to monitor used bandwidth on specified interface of the monitoring server. Such a project running e.g. on software bridge or router on the edge of the network would inform us about bandwidth utilization, who creates most of the traffic, which protocols eat our network resources, who breaks company's security rules for example for downloading from P2P networks or whether our network infrastructure is not attacked from outside the network and misused for spam attacks. We implemented simple analysis program and web page for displaying results of the measuring. The project is fully configurable to meet specific user's needs.

To improve the project we will implement some more features like VoIP SIP protocol support, SNMP and other network monitoring protocols support to monitor events in network and multithreading for more efficient capture and analysis of frames.

ACKNOWLEDGEMENT

This paper presents the research results and acquired experience of the authors obtained within the following research projects:

Convergence of ICT networks and services in the Slovak communication infrastructure (national research project),

VEGA No. 1/3094/06 (national basic research project) and VEGA 1/4084/07.

Applied research project grant in cooperation with Sitronics Telecom Solutions Slovakia under no.AV 4/0019/07

References

[1] Juraj Kacur, Jan Korosi, An Accuracy Optimization of a Dialog ASR System Utilizing Evolutional Strategies, ISPA 2007, Istanbul, Turkye, Sept. 2007.

[2] Trúchly, P., Urbanovič, M.: MPLS Throughput over GEO satellites, ELMAR-2006, 48th International Symposium on Multimedia Signal Processing and Communications, 7-9 June 2006, Zadar, Croatia, pp. 305-308

[3] AdventNet NetFlow Analyzer: http://www.adventnet.com/news/netflow.html, March 2008

[4] Solarwinds Network Traffic Analyzer: http://www.solarwinds.com/products/orion/nta, March 2008

[5] ntop: http://www.ntop.org/, March 2008

[6] IP Network Monitoring Software: http://iptraf.seul.org/, March 2008

[7] Skyward IP Traffic Monitor: http://www.skyward-soft.com/mambo/index.php?option=content&task=view&id=17&Itemid=38, March 2008

[8] ZABBIX: http://www.zabbix.com/, March 2008

[9] Roman Benkovič: Meranie a vyhodnotenie prevádzky v sieťach LAN a IP, Diploma thesis 2006, KTL, FEI STU Bratislava

[10] pcap library: http://www.tcpdump.org/pcap3_man.html, March 2008

[11] http://ngnlab.eu/katalyzer Project page, May 2008.