Traffic Management in Rural Networks

Rodrigo Emiliano[1], Fernando Silva[1], Luís Frazão[1], João Barroso[2], António Pereira[1,3]

[1] School of Technology and Management, Computer Science and Communication Research Centre, Polytechnic Institute of Leiria, Leiria, Portugal
rodrigoemiliano@hotmail.com, {fernando.silva, luis.frazao, antonio.pereira}@ipleiria.pt
[2] INESC TEC (formerly INESC Porto) and Universidade de Trás-os-Montes e Alto Douro, Quinta de Prados, 5000-801 Vila Real, Portugal
jbarroso@utad.pt
[3] Information and Communications Technologies Unit, INOV INESC Innovation-Delegation Office at Leiria, Leiria,Portugal

Abstract. The internet is increasingly present in people's lives, being used in diverse tasks, such as checking e-mail up to online gaming and streaming. The so-called "killer applications" are applications that, when not properly identified and prevented, have more impact on the network, making it slow. When these applications are used on networks with limited resources, as happens in rural networks, they cause a large load on the network, making it difficult its use for work purposes. It is important then to recognize and characterize this traffic to take action so that it does not cause network problems. With that in mind, the work presented in this paper describes the research and identification of cost free traffic analysis solutions that can help to overcome such problems. For that, we perform preliminary testing and a performance comparison of those tools, focusing on testing particular types of network traffic. After that, we describe the analysis and subsequent modification of the source code for storing important traffic data for the tests, as well as the test scenarios in laboratory and real-life environments. These tasks are aimed on collecting information that assists in taking action to improve the allocation of network resources to priority traffic.

Keywords: Internet, Network Traffic, Rural Networks, Traffic Analysis, Deep Packet Inspection.

1 Introduction

The number of people with access to the Internet has been growing very quickly over the years. It is estimated that 1 in 3.5 people in the world have internet access [1]. These figures suggest the global character of this service, resulting from its increasing availability and from its decreasing access prices.

Nevertheless the Internet is still not available equally to all people. Unfortunately, the more remote and distant areas of the cities are the most affected, due to the cost and non-profitability of implementing an Internet service with good quality in those

areas [2]. Most of those zones are rural areas that by not having a large population density are set aside for not presenting a benefit for the implementation of such service.

Generally, if an access service to the Internet exists in rural areas, it is fairly limited, slow and with a low bandwidth. Therefore, when a large number of people accesses this service at the same time, a huge load is caused on the network. This load can be caused by the use of "killer applications" such as BitTorrent clients, peer-to-peer (P2P) programs, or even services that require lots of bandwidth, such as video and movies streaming. This compromises the proper functioning of the network and the quality of the service for other users that needs the internet for work purposes [3].

This is a difficult problem to tackle due to the weak network resources and the lack of control regarding the use of these "killer applications". One possible solution to this problem is the use of network traffic analysis tools to identify and control the traffic generated by these applications. Deep Packet Inspection (DPI) is a form of network packet filtering that allows identifying this type of traffic by checking the data packet origin application, as well as allows identifying the users that are responsible for its generation. The identification of this traffic and the implementation of measures that modify the allocation of resources to these applications, assist to a better management of the network resources, allowing a fair differentiation of applications and thus, preventing overloading the network with "unwanted" traffic and, consequently, its malfunction.

The research, testing and implementation of cost-free applications that perform this type of analysis, is the way forward to prevent the application of large load in rural networks.

This paper is organized as follows. In the next Section we introduce concepts underlying our research on traffic analysis technologies. Sections 3 and 4 describe our proposed solution and its application to a rural network, respectively. In Section 5 we describe our test scenarios on the selected application and in Section 6 we describe the adaptations made to its source code and the tests performed. Section 7 presents the results that we were able to observe, both in a laboratory and in a rural network. Finally, the ending Section presents final considerations on our work and topics for future work.

2 Traffic Analysis and Deep Packet Inspection

The traffic analysis on a network is increasingly important, contributing not only to understand the type of traffic flowing in the network, but also to prevent its congestion and ensure information security. This section presents DPI as a valuable traffic analysis approach.

2.1 Traffic Analysis

The traffic analysis on a network is widely used in the detection and resolution of network problems, thus improving the network in order to improve its operation in the

future [3]. The traffic analysis makes the examination of packets passing a network card interface into promiscuous mode, or through Port Mirroring or Network Taps, and it generally allows deducing information from patterns in this communication.

2.2 Deep Packet Inspection

The Deep Packet Inspection (DPI) is a form of filtering/analysis of data packets, which examines more than the destination and source addresses. This parses the packet from layer 2 to 7 of the OSI model. This type of packet analysis allows a more complete monitoring and management of the network, by collecting information from headers, data, and protocol structures, in order to identify and classify the type of package or communication [4]. However, although this type of analysis is already being used for some time now in advertising or even in preventing spam emails, it creates many conflicts among the defenders of neutrality on the Internet due to its intrusive nature. Nevertheless, despite this ethical issue, DPI helps on several network tasks [4].

3 Proposed Solution

To collect and store the information in the network, we propose the model for the collection and processing of information illustrated in Fig. 1. This model consists of the following modules: gathering, filtering, storage, parsing and action.

Fig. 1. Proposed model for traffic analysis

Collection: The first step in processing information flowing in the network is the collection of such information. This collection must gather all of the data circulating on the network, such as the actors involved in the communication, the application that is transmitting the information, among others.

Filtering: The data collection should be filtered to display only the information relevant for the traffic analysis to subsequently proceed to its storage.

Storage: The filtered information should be stored in a media where it can be accessed. This storage will allow building the network traffic history.

Parsing: When necessary, the filtered information should be processed in order to easily understand what it concerns. This step also allows the reorganization of information in order to be saved in a format that can be imported by different programs.

Action: In the future, this information will provide the basis for the imposition of rules of distribution of resources and improvement of the network.

4 Solution Applied to a Rural Network

In order to improve resource allocation in networks with limited resources, a rural network was chosen to test the architecture proposed in Section 4.

Due to not having a large population density, many rural areas have very limited access to internet services, since they do not present a clear benefit to the operators of this service. Even when an internet service exists, it is rather poor in its quality. It is important then, to improve the allocation of resources in these networks, as the management of broadband networks in rural areas is of great importance. Fig. 2 shows how our solution can be applied to a rural network for management and traffic analysis purposes.

The proposed architecture has the following main objectives:

- Gathering all the information that enters/exits the network: This objective is illustrated through the use of a Central Server, where all traffic entering or leaving the network flows. Since the majority of the network traffic flows through this point it is ideal for collecting all of the necessary information. Therefore, this server will run the tool that collects information about network traffic. Using a Central Server simplifies the data collection and analysis processes, since it is not necessary to place the traffic analysis tool in many different parts of the network.
- Remote Administration: This allows non-presence maintenance of the solution, which represents a clear advantage. Thus, software maintenance and configuration is easily achieved, as well as the collection and observation of the data.
- Modelling of the network: After collecting and storing information, the network must be modelled so that all available resources are allocated according to the current needs. This modelling will avoid an excessive load on the network and, consequently, will result in its proper functioning.

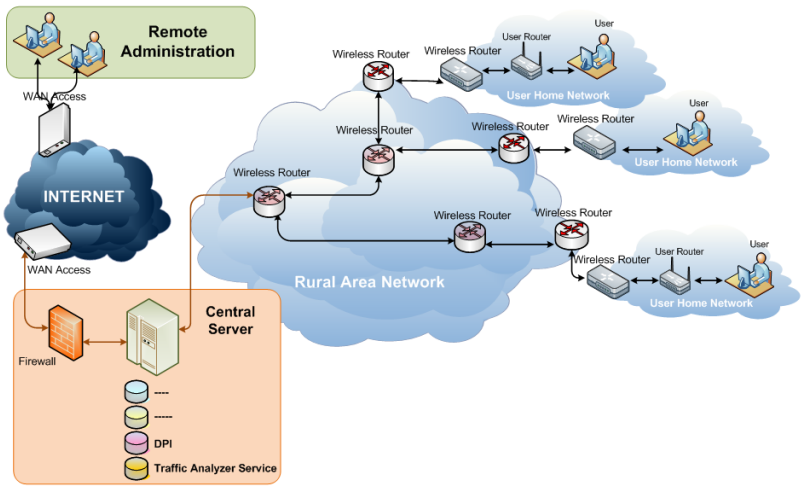

Fig. 2. Proposed solution architecture

5 Testing Settings

As one of the major objectives of this work is the detection of harmful traffic to the network, it is necessary to identify its different intervenient in order to perform the management that allows the proper functioning of the network.

One possible way to discover the application that is responsible for sending traffic is through the identification of its operating port [5]. However, due to technology advancements, there is increasing difficulty in identifying an application through the packet headers. Hence the use of DPI, which identifies those applications and all of the traffic present on the network [4].

During the initial stage of this study we have investigated several network traffic analysis programs, in order to find the tool that best fits our goals. After this extensive research, and taking into account different criteria, ntop [6] was chosen as the most appropriate tool. Ntop is very easy to install and to setup. Also, it has an advanced DPI library and it allows gathering relevant information that will help on managing the network load [7]. Being an open source tool, it is possible to edit its source code in order to perform additional functions as required, such as automation of traffic analysis and data storage. It is also possible to perform its configuration remotely, which represents an advantage since it is not necessary to travel to the site to perform tasks.

Using ntop, several test sessions were held, each with the duration of one hour, for the identification of applications and application protocols. These tests consisted in using controlled traffic with the aim of identifying it through the DPI. The traffic generated by applications such as Twitter, YouTube, Skype, among others, was tested. The results were then registered for each one of the tests performed.

After knowing the features of ntop and finishing the preliminary testing, the tests with "killer applications" were performed. Several P2P programs were tested and compared.

5.1 Peer to Peer (P2P)

P2P networks are applications with distributed architecture that distribute tasks among peers. All of the peers have the same permissions, capabilities and responsibilities to each other, in which they differ from the client/server architecture, where some of the actors are dedicated to serving others. This type of architecture is quite simple and it does not require coordination by a central authority.

P2P is often incorrectly used for setting up file sharing, such as music, video or game files via the Internet. However, its usefulness is much wider.

5.2 Unknown Traffic

Although DPI allows a very rich data analysis, there are still many applications and types of traffic to be identified. The traffic that ntop is unable to identify is classified as "unknown". During some sessions, the initial traffic was detected as "unknown". This happened when the web interface was started, suggesting that some kind of connection exists.

However, it was observed that there are more cases in which traffic is marked as “unknown”. These cases are mostly related to P2P traffic. By enabling protocol encryption, the BitTorrent clients prevent their traffic identification, which is then identified as “unknown”. The traffic generated by these clients is quite bulky and causes a large load on the network, therefore, although not identified, it does not go unnoticed. Even when identified as “unknown” there is some traffic that is correctly identified as “bittorrent”. Knowing that there is BitTorrent traffic flowing in the network and that there is also a large volume of unknown traffic, it is safe to infer that these two types are related. Then, it is important to identify, not only those involved in BitTorrent traffic, but also the ones involved in the unknown traffic.

5.3 Protocol Encryption and DPI

Protocol encryption is widely used by BitTorrent clients. This happens because many Internet Server Providers (ISPs), block P2P traffic in order to prevent the network bandwidth overload [3]. Due to protocol encryption, BitTorrent traffic remains undetected by ISPs. The DPI analysis also fails on its identification, marking it as “unknown” traffic. Therefore, we infer that the majority of the unknown traffic is possibly caused by BitTorrent clients' protocol encryption. In order to detect and register the perceivable changes that this traffic causes on the network, we have tested different BitTorrent clients.

KTorrent. It is fairly easy to install on UNIX operating systems. When performing the tests, it was found that all of the captured traffic was recognized by P2P filtering. However, this BitTorrent client supports encryption protocol, making it unrecognizable to the DPI.

OpenBitTorrent. This is a tool that usually comes included with Ubuntu 12.04 operating system. It is quite simple; however, in the tests performed, the DPI did not identify its traffic as “bittorrent”. Instead, this traffic was identified as “unknown”, which suggests that this client uses protocol encryption.

UTorrent. It is one of the most widely used BitTorrent clients. It can be identified by ntop's DPI. Due to its importance, our tests are mostly focused in this tool.

In order to know how traffic changes with the different possible configurations, we started by using Utorrent default settings. We observed that near 90% of the traffic was correctly identified as “bittorrent”. However, UTorrent supports protocol encryption, which makes its traffic unrecognizable to ntop's DPI.

Relationship between P2P, Hosts, and Active Sessions. When a new actor is detected in the network traffic, it is added to a ntop list of hosts, storing several information about its data, such as name, IP address and amount of sent and received traffic, among others. As the P2P connection uses multiple peers, it detects much traffic with

different origins [8]. Thus, during the tests with BitTorrent clients, multiple hosts from different countries, with high amounts of traffic sent/received, were detected.

6 Source Code Analysis, Implementation and Testing

Because many BitTorrent clients are configured for protocol encryption and much of the information that is detected cannot be extracted through the available mechanisms, we have proceeded with the study of ntop's source code. Thus, we sought to understand the packet capture functioning and its respective analysis and processing, for subsequent amendments in the code for collecting the required information.

After confirming the architecture of ntop, we have identified the code snippets responsible for capturing and processing packages: *startSniffer()*, *processPacket()*, *isP2P()*. However, these snippets do not distinguish relevant traffic for the data collection. For this, we have deepened our study to the DPI code.

Within the protocols folder of the DPI code, we find different files for identifying the different protocols, as illustrated in Fig. 3. For example, for the discovery of BitTorrent traffic, there is a file named "bittorrent.c". However, for HTTP traffic, such as YouTube and Twitter, there is only one file: "http.c".

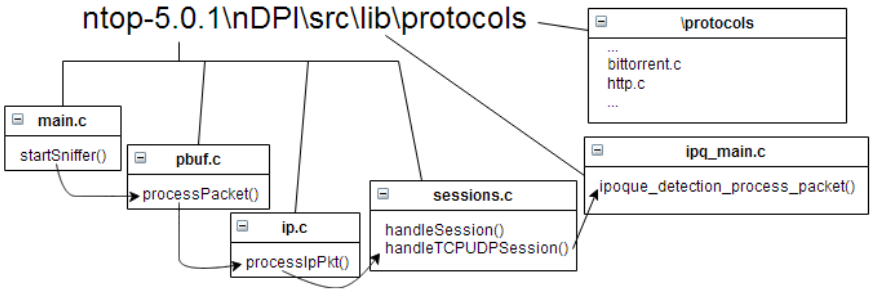

Fig. 3. Ntop's architecture

The study of the DPI code and the tests performed on BitTorrent clients, allowed us finding areas in the code where it is possible to collect information concerning hosts that carry harmful network traffic. For this purpose, it is necessary to perform the information collection within each file responsible for the different protocols.

6.1 Bitorrent.c

This file's functions allow determining if a data packet is originated by BitTorrent traffic. Thus, it is possible to find those responsible for this traffic by simply performing a *printf* of the data packet. In the code, the packet is located within an *ipoque_packet_struct* structure, which in turn is within the *ipoque_struct* structure.

The collected information was not in a suitable format, so a parser was made to treat this information. The data taken from the "bittorrent.c" file were the destination

IP address, the source IP address, the Terms of Service (TOS) and the source and destination ports.

6.2 Extension of Data Collection to the "unknown" Data Type

Capturing traffic identified as "unknown" is of major importance, because a great majority of P2P traffic is marked as such. However, there is no source file to treat unknown traffic, therefore we have climbed in the file hierarchy and we have analysed the "ipq_main.c" code file. This file has the *ipoque_detection_process_packet()* function, which is responsible for discovering the application that is sending a given data packet. This function discovers the application protocol of the data packet. We assume that whenever it is not possible identifying the application, it should be identified as "unknown". Therefore, in these cases, we collect this type of information by making the function return the value "unknown".

6.3 Collecting Information on the Size and Timestamp of the Packet

After collecting the information on the different intervenient and on the applications of a given data package, we have captured more information which help us on building statistics about the data. This information is the size and the timestamp of the package. This information also helps on identifying network traffic delays.

6.4 Using Wireshark to Compare Data

Throughout the testing of our solution, all collected data were compared with data from the Wireshark tool [9], which also allows the collection and analysis of traffic. These comparisons were made using samples of traffic used in the tests performed with ntop. As these samples can also be read by Wireshark, it was possible to check the accuracy of the proposed solution.

6.5 The Parser

A parser, written in PERL, was created to handle data. This parser is designed to perform its conversion to a suitable and perceivable format that the *printf* function does not always allows achieving.

The parser converts the IP and port address from the decimal format, to dotted decimal and port, respectively. Then, it converts the protocol number into a string, which can present the values "UDP" or "TCP". After that, it performs the conversion of application number into the corresponding application, through an additional file containing the correspondence "application → value". Finally, it stores the converted information into a file, using the Comma Separated Value format (.csv), as illustrated in Fig. 4.

	A	B	C	D	E	F	G	H
1	Application	Protocol	Source IP	Port	Destination IP	Port	TOS	Size
2	[UNKNOWN]	[TCP]	77.243.180.38	80	10.0.1.2	49596	0	1480
3	[UNKNOWN]	[TCP]	77.243.180.38	80	10.0.1.2	49596	0	1480
4	[UNKNOWN]	[TCP]	10.0.1.2	49596	77.243.180.38	80	0	52
5	[UNKNOWN]	[TCP]	77.243.180.38	80	10.0.1.2	49596	0	1480

Fig. 4. Sample of parser's processed file

6.6 Automation

In order to reduce the human interaction to a minimum, we opted for using automation for the data collection and its processing procedures. The main objective of this automation is to perform an action that depends on the current results. For that purpose we use the cron program, which allows the automatic execution of a given program [10]. Thus, a simple change via cron enables the automatic collection and processing of data, continuously or through sample packets.

7 Results

The laboratory results show that it is possible to discover the interveners in heavy traffic in the network and collect proper information on that same traffic. However, in order to ensure that these results are also observed in a real environment, we have implemented our model to Memória Network.

The Memória Network was the result of a project entitled Memoria Online [11], which consisted on the implementation of a network in this rural location, granting to its inhabitants access to the internet. However, due to the number of clients using this network, its resources must be well distributed, in order to maintain its quality and speed. So, our proposed solution was implemented in this network in order to collect and analyze traffic, for helping to perform a better modeling of the network.

After applying our model in Memória Network, data collection was performed over a period of 3 hours. This collection has captured about 10 million packages. After analyzing the collected data, we conclude that it is possible to identify camouflaged P2P traffic in the network. This traffic is then queued in the low priority Quality if Service (QOS) list in order to not disturb the other traffic on the network. We also observe that the traffic is being well identified, allowing to detect if that same traffic is being correctly classified and marked by the QOS mechanisms.

In sum, the proposed model allows unmasking the camouflaged traffic and that affects network performance, as well as provides efficient mechanisms that help optimizing the network.

8 Conclusion

The aim of this project was to create a model that would allow improving resource allocation in networks with scarce resources, such as rural networks. This would require using cost-free solutions of network traffic analysis and the ability to perform data filtering through DPI.

Ntop is a network traffic analysis tool that complies with the requirements of the project. It was preliminarily tested in controlled environments for identifying applications generating traffic considered harmful and not harmful. Then, its source code has been modified to allow storing the information relating to the most harmful network traffic. Solutions that enable automation of the model and evaluating the classification accuracy of the collected data were also implemented.

The results show that our model allows unmasking the camouflaged traffic that negatively affects scarce resource networks, simultaneously providing relevant information for improving network management.

We conclude that ntop is a very good tool for the identification of network traffic, especially for identifying harmful network applications. Being an open-source tool and having a good capacity for analysis and identification of traffic through the DPI, it is a viable option for implementing in resource-limited networks, or even in networks that seek an intelligent management of their traffic.

For future work, it is important to perform additional tests on other real scenarios, as well as to collect additional information that enables a richer statistical analysis. It is also important to investigate other tools for analysing traffic on large networks such as ntop next-generation [12], which allows network analysis with great resources and with high speed traffic exchange.

References

1. Internet World stats, “Usage and population statistics”, http://www.internetworldstats.com/stats.htm. Date accessed, January 2014.
2. Liew, J. H., Yeo, A. W., Hamid, K. A., Othman, A. K.: Implementation of wireless networks in rural areas. Computer Science and Information Tech, Malaysia Univ. (Sarawak) (2004)
3. Feitosa, E., Souto, E., Sadok, D. H.: An orchestration approach for unwanted Internet traffic identification. Computer Networks, 56(12), 2805-2831 (2012)
4. Mueller, M.L.: Convergence of control? Deep packet inspection and the future of the internet. Communications & Convergence Review, 2(2), 92-103 (2010)
5. IANA, “Protocol Numbers”, http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml. Date accessed, January 2014].
6. ntop, “ntop”, http://www.ntop.org/. Date accessed, January 2014.
7. Bujlow, T., Carela-Español, V., Barlet-Ros, P.: Comparison of Deep Packet Inspection (DPI) Tools for Traffic Classification. Technical Report, Universitat Politècnica de Catalunya (2013)
8. Gomes, J. V., Inácio, P. R. M., Pereira, M., Freire, M. M., Monteiro, P. P.: Detection and Classification of Peer-to-Peer Traffic: A Survey. ACM Computing Surveys 45(3), 1-40 (2013)

9. Wireshark, “Wireshark”, http://www.wireshark.org/. Date accessed, January 2014
10. LinuxQuestions, “Tcpdump with cron,”. http://www.linuxquestions.org/questions/linux-software-2/tcpdump-with-cron-121727/. Date accessed, January 2014
11. Salvador, N.; Filipe, V.; Rabadão, C.; Pereira, A.: Management Model for Wireless Broadband Networks. In: 3rd International Conference on Systems and Networks Communications, pp. 38-43, ICSNC (2008)
12. ntop, “ntopng”, http://www.ntop.org/. Date accessed, January 2014