A Method for Assessing Quality of Service in Broadband Networks

Tomasz Bujlow, Tahir Riaz, Jens Myrup Pedersen
Section for Networking and Security, Department of Electronic Systems
Aalborg University, DK-9220, Aalborg East, Denmark
Email: {tbu, tahir, jens}@es.aau.dk

***Abstract*—Monitoring of the Quality of Service (QoS) in high-speed Internet infrastructures is a challenging task. However, precise assessments must take into account the fact that the requirements for the given quality level are service-dependent. The backbone QoS monitoring and analysis requires processing of large amounts of data and the knowledge about the kinds of applications, which generate the traffic. To overcome the drawbacks of existing methods for traffic classification, we proposed and evaluated a centralized solution based on the C5.0 Machine Learning Algorithm (MLA) and decision rules. The first task was to collect and to provide to C5.0 high-quality training data divided into groups, which correspond to different types of applications. It was found that the currently existing means of collecting data (classification by ports, Deep Packet Inspection, statistical classification, public data sources) are not sufficient and they do not comply with the required standards. We developed a new system to collect the training data, in which the major role is performed by volunteers. Client applications installed on volunteers' computers collect the detailed data about each flow passing through the network interface, together with the application name taken from the description of system sockets. This paper proposes a new method for measuring the level of Quality of Service in broadband networks. It is based on our Volunteer-Based System to collect the training data, Machine Learning Algorithms to generate the classification rules and the application-specific rules for assessing the QoS level. We combine both passive and active monitoring technologies. The paper evaluates different possibilities of the implementation, presents the current implementation of the particular parts of the system, their initial runs and the obtained results, highlighting parts relevant from the QoS point of view.**

***Index Terms*—broadband networks, data collecting, Machine Learning Algorithms, performance monitoring, Quality of Service, traffic classification, volunteer-based system.**

I. Introduction

One of the most interesting challenges in today's world is how to measure the performance of computer network infrastructures, when different types of networks are merged together. In the last few years, the data-oriented networks evolved into converged structures, in which the real-time traffic, like voice calls or video conferences, is more and more important. The structure is composed of traditional data cable or more modern fiber links, existing Plain Old Telephone Service (POTS) lines used to provide analog services (voice telephony), or digital services (ADSL, PBX, ISDN), and nowadays also of mobile and wireless networks. There are numerous methods for the measurement of Quality of Service (QoS) in the current use, which provide the measurements both on the user side and in the core of the network. Internet Service Providers are interested in centralized measurements and detecting problems with particular customers before the customers start complaining about the problems, and if possible, before the problems are even noticed by the customers.

Each network carries data for numerous different kinds of applications. QoS requirements are dependent on the service. The main service-specific parameters are bandwidth, delay, jitter, and packet loss. Regarding delay, we can distinguish strict real time constraints for voice and video conferences, and interactive services from delivery in relaxed time frame. In a conversation, delay of about 100 ms is hardly noticeable, but 250 ms of delay means an essential degradation of the transmission quality, and more than 400 ms is considered as severely disturbing [1].

Therefore, in order to provide detailed information about the quality level for the given service in the core of the network, we need to know, what kinds of data are flowing in the network at the present time. Processing all the packets flowing in a high-speed network and examining their payload to get the application name is a very hard task, involving large amounts of processing power and storage capacity. Furthermore, numerous privacy and confidentiality issues can arise. A solution for this problem can be the use of Machine Learning Algorithms (MLAs), which use previously generated decision rules, which are based on some statistical information about the traffic. In our research, we used one of the newest MLAs – C5.0. MLAs need very precise training sets to learn how to accurately classify the data, so the first issue to be solved was to find a way to collect high-quality training statistics.

In order to collect the necessary statistics and generate the training sets for C5.0, a new system was developed, in which the major role is performed by volunteers. Client applications installed on their computers collect the detailed information about each flow passing through the network interface, together with the application name taken from the description of the system sockets. The information about each packet belonging to the flow is also collected. Our volunteer-based system guarantees precise and detailed data sets about the network traffic. These data sets can be successfully used to generate statistics used as the input to train MLAs and to generate accurate decision rules.

The knowledge about the kind of application to which the traffic belongs obtained from MLAs can be used together with traffic requirements for the given application to assess the QoS level in the core of the real network. The real traffic needs to be sampled to obtain the necessary raw statistics. Parameters like jitter, burstiness, download and upload speed (and delay and packet loss for TCP traffic) can be assessed directly on the basis of the information obtained from the captured traffic. To assess the delay and packet loss for UDP traffic, active measurement techniques must be involved (like ping measurements in both directions).

The remainder of this document is splitted into several sections describing in detail the system architecture and some parts of the implementation. Section II contains an overview of current methods of assessing the network QoS level. Both passive and active methods are described along with their advantages and weaknesses. Section III gives an overview of our methods, so the reader is able to understand how the particular components are built and connected with each other. Sections IV, V, VI, and VII demonstrate the design and implementation of the system, while Section VIII summarizes the most important points.

II. Related work

During the last 20 years we have been witnesses to the subsequent and increasing growth of the global Internet and the network technology in general. The broadband and mobile broadband performance today is mainly measured and monitored by speed. However, there are several other parameters, which are important for critical business and real-time applications, such as voice and video applications or first-person shooter games. These parameters include round trip time, jitter, packet loss, and availability [2], [3].

The lack of the centralized administration makes it difficult to impose a common measurement infrastructure or protocol. For example, the deployment of active testing devices throughout the Internet would require a separate arrangement with each service provider [2]. This state of affairs led to some attempts to make simulation systems representing real characteristics of the traffic in the network. Routers and traffic analyzers provide passive single-point measurements. They do not measure the performance directly, but the traffic characteristics are strongly correlated with the performance. Routers and switches usually feature a capability to mirror incoming traffic to a specific port, where a traffic meter can be attached. The main difficulty in passive traffic monitoring is the steadily increasing rate of transmission links (10 or 100 GB/s), which can simply overwhelm routers or traffic analyzers, which try to process packets. It forces the introduction of packet sampling techniques and, therefore, it also introduces the possibility of inaccuracies. Even at 1 Gbit/s, the measurements can result in enormous amounts of data to process and store within the monitoring period [2].

To overcome the heavy load in the backbone and to not introduce inaccuracies, a smart monitoring algorithm was needed. There are several approaches to estimate which traffic flows need to be sampled. A path anomaly detection algorithm was proposed in [4]. The objective was to identify the paths, whose delay exceeds their threshold, without calculating delays for all paths. Path anomalies are typically rare events, and for the most part, the system operates normally, so there is no need to continuously compute delays for all the paths, wasting processor, memory, and storage resources [4]. Authors propose a sampling-based heuristic to compute a small set of paths to monitor, reducing monitoring overhead by nearly 50 % comparing to monitoring all the existing paths.

The next proposals on how to sample network traffic in an efficient way were made on the basis of adaptive statistical sampling techniques, and they are presented in [5] and [6].

If a congestion is detected, from user's perspective it is very important to know, if the congestion is located on the local or on the remote side. If the link experiences a local congestion, the user may be able to perform certain actions, e.g. shut down an application, which consumes a lot of bandwidth, to ease the congestion. On the other hand, if the congested link is a remote link, either in the Internet core or at the server side, the back-off of the low-priority applications on the user's side is unnecessary. It only benefits the high-priority flows from other users, which compete for that link. Since this altruistic behavior is not desirable, the low priority TCP only needs to back off, when the congested link is local [7].

Detecting the location of the congestion is a challenging problem due to several reasons. First of all, we cannot send many probing packets, because it causes too much overhead, and it even expands the congestion. Secondly, without a router support, the only related signals to the end applications are packet losses and delays. If the packet losses were completely synchronized (packets were dropped from all the flows), the problem would be trivial. In the reality, the packet loss pattern is only partially synchronized [7]. Authors of [7] attempted to solve the problem of detecting the location of the congestion by using the synchronization of the behavior of loss and delay across multiple TCP sessions in the area controlled by the same local gateway. If many flows see a synchronized congestion, the local link is the congested link. If the congested link is remote, it is less likely that many flows from the same host pass the same congested link at the same time. If there is only a small number of flows which see the congestion, the authors performed an algorithm based on queuing delay patterns. If the local link is congested, most flows typically experience high delays at a similar level. Otherwise, the congestion is remote [7].

The traffic can be profiled according to the protocol composition. Usually, the predominance of the TCP traffic is observed (around 95 % of the traffic mix). When a congestion occurs, TCP sources respond by reducing their offered load, whereas UDP sources do not. It results in the higher ratio of UDP to TCP traffic. If the proportion becomes high and the bandwidth available to TCP connections becomes too low to maintain a reasonable transmission window, the packet loss increases dramatically (and TCP flows become dominated by retransmission timeouts) [2]. Packet sizes provide insight into

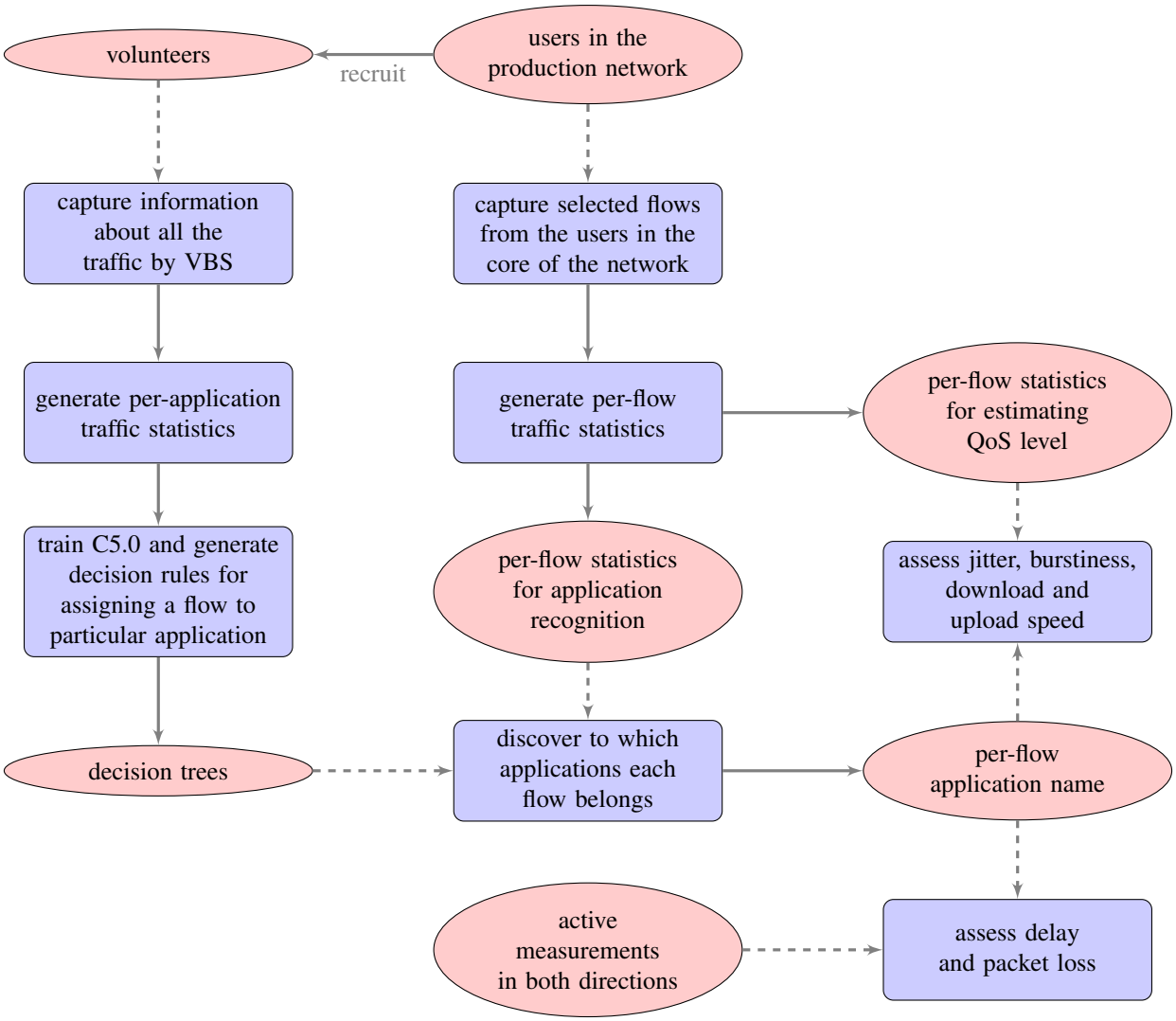

Figure 1. Overview of the system architecture

the types of packets, e.g. short 40-44 bytes packets are usually TCP acknowledgments or TCP control segments (SYN, FIN or RST) [2].

Active methods for QoS monitoring raise three major concerns. First, the introduction of the test traffic will increase the network load, which can be viewed as an overhead cost for active methods. Second, the test traffic can affect measurements. Third, the traffic entering an ISP can be considered as invasive and discarded or assigned to a low-priority class [2].

Within an administrative domain (but not across the entire Internet), the performance can be actively monitored using the data-link layer protocol below IP, as the Operations, Administration and Maintenance (OAM) procedure in ATM and MPLS networks. As a result, at the IP layer it is often desirable to measure performance using the IP/ICMP protocol. So far, most tools or methods are based on ping (ICMP echo request and echo reply messages) or traceroute (which exploits the TTL field in the header of the IP packet) [2].

Although the round-trip times measured by ping are important, ping is unable to measure the one-way delay without additional means like GPS to synchronize clocks at the source and destination hosts. Another difficulty is that pings are often discarded or low-prioritized by many ISP in their networks. Traceroute will not encounter this problem because UDP packets are used. However, traceroute has known limitations. For example, successive UDP packets sent by traceroute are not guaranteed to follow the same path. Also, the returned ICMP message may not follow the same path as the UDP packet that triggered it [2].

Although the end-to-end performance measurements can be carried out at the IP layer or the transport/application layer, the latest is capable of measurements closer to user's perspective. The basic idea is to run a program emulating a particular application that will send traffic through the Internet. All the parameters (delay, packet loss, throughput, etc) are measured

on the test traffic. This approach has one major drawback - a custom software needs to be installed on the measurement hosts [2].

On the basis of the mentioned work, we found out that the existing solutions are not sufficient for precise QoS measurements. This state of affairs motivated us to create a new system which combines both passive and active measurement technologies.

III. Overview of the Methods

The flow chart of our system is shown in Figure 1. The following paragraphs contain the detailed description of our methods. At first, the volunteers must be recruited from the network users. The volunteers install on their computer a client program, which captures the relevant information about the traffic and submits the data to the server. On the server, these data are used to generate per-application traffic statistics. The C5.0 Machine Learning Algorithm uses these statistics to learn how to distinguish between different types of applications and, later, it generates the classification rules (decision trees).

In order to assess the network QoS level in the core of the network for particular users, we needed to find a method to capture the relevant traffic. The challenging task is to process significant amounts of traffic in high-speed networks. When the relevant flows are captured, per-flow statistics need to be generated. There are two kinds of statistics generated at this step: one used for determining the kind of application associated with that flow, and one used for assessing the QoS level in the passive way. The system uses previously generated classification rules together with the first type of statistics to find out which application the flow belongs to. Then, on the basis of the kind of the application, the system determines the ranges of values of the relevant QoS parameters. The last step is to check if the current values (obtained from the flow statistics or in the active way) match the expected ones. If not, the quality of the given service is considered as degraded.

IV. Volunteer-Based System

There are many possible methods for collecting data, but not all the methods are feasible to deliver data required for obtaining accurate statistics to train the MLAs:

- running one application per host at a time and capturing all the data by Pcap, Wireshark, or a similar tool [8]. It requires installing and running every application of which we would like to capture the traffic. It is slow and not scalable. Background traffic can easily influence the obtained results
- port-based classification [9], [10]. It is fast and supported on almost all the network layer-3 devices, but it is not possible to classify protocols using dynamic port numbers, like P2P and Skype [8], [11], [12]
- Deep Packet Inspection (DPI) [13] is slow, requires a lot of processing power [8], [11], and privacy and confidentiality issues can appear [8]. It is not possible to use DPI to recognize encrypted traffic

Therefore, we decided to develop a system based on volunteers, which captures the traffic from their network interfaces, and groups the traffic into flows associated with the application name taken from Windows or Linux sockets. The architecture and the prototype were described and analyzed in [14] and [15], and the first version of our current implementation was presented in [16]. This cross-platform solution consists of clients installed on users' computers (Microsoft Windows XP and newer and Linux are supported), and of a server responsible for storing the collected data. The client registers information about each flow passing the Network Interface Card (NIC), with the exception of the traffic to and from the local network. It collects also information about all the packets associated with each flow. Collected information is then transmitted to the server, which stores all the data in a MySQL database for further analysis. The system was shown in [17] to be feasible and capable of providing detailed per-application information about the network traffic.

V. Obtaining Per-Application Statistics

The next step was to obtain the statistical profiles of flows for different applications. Therefore, we developed a tool for calculating statistics of several traffic attributes for each flow in the database, which fulfills our requirements. In our small-scale prototype for tests, we decided to limit the number of applications and take into account Skype, FTP, torrent, web traffic, web radio traffic, interactive game traffic, and SSH [18]. The statistics include 32 attributes based on sizes and 10 protocol-dependent attributes [18]. We suspect that the attributes based on sizes are independent of the current conditions in the network (like for example congestions). All the protocol-dependent attributes are very general – they contain transport protocol, local and remote port information, number of TCP flags in the traffic (in both directions), proportion of inbound / outbound / both directions packets without payload to the number of all packets. Precise port numbers are not used, but only the information about whether the port is well-known or dynamic. This way, we avoid constructing a port-based classifier, but we can retain the information if the application model is more like client-server or peer-to-peer.

VI. Machine Learning Algorithms

In the recent literature, we can find numerous approaches to use Machine Learning Algorithms to classify the traffic in computer networks. The most widely used MLA classifiers are C4.5 [8] and its modified Java implementation called J48 [11], [19]. Based on statistical analysis, MLAs have the ability to assign a particular class (like P2P) even to traffic generated by unknown applications [8]. It was also proved in [19] that the statistical parameters for the encrypted and unencrypted traffic produced by the same application are similar and, therefore, the encrypted payload does not influence results of the training or the classification. The accuracy of the classification by MLAs was claimed to be over 95 % [8]–[10], [12], [13], [20]–[22]. The analysis of the related work can be found in [18].

It was found in [10] that the results of the classification are most accurate when the classifier was trained in the same network as the classification process was performed. This may be due to different parameters, which are constant in the particular network, but which differ among various networks. A good example is the Maximum Transmission Unit, which can easily influence statistics based on sizes. Therefore, in our design, we decided to train the classifier by volunteers in the same network as the classifier will be installed. This allows us to make a self-learning system, where a group of volunteers in the network delivers the data used for training the classifier constantly improving its accuracy, while all the users can be monitored in the core using the generated decision rules. The next advantage of the design is that even if some network users cannot participate in the data collecting process because of using other operating systems or devices than supported (like MacOS, Apple or Android smartphones), they will still be able to be monitored in the core of the network because of rules created on the basis of the data collected from the other users.

Our system uses the C5.0 MLA, which is a successor of C4.5. It is proven to have many advantages over its predecessor, such as higher accuracy, possibilities to use boosting, pruning, weighting and winnowing attributes. Furthermore, the time needed to generate the decision tree or rules drastically decreased [23]. In order to test the efficiency of C5.0, we performed a set of tests during which we used various training and classification options. The training statistics were obtained from the data provided by our VBS. During our research, we found relevant the set of arguments and discovered that the best results were obtained using the boosted classifier. The average accuracy fluctuated between 99.3 % and 99.9 %, depending on the number of training and test cases and the amount of data from each case. It is worth mentioning that in our experiment we considered only 7 different groups of applications and only flows longer than 15 packets. In our small-scale prototype for tests, we decided to limit the number of applications and take into account Skype, FTP, torrent, web traffic, web radio traffic, interactive game traffic, and SSH [18]. The limitation of the flow length was done because we needed to have at least 5 packets to generate the statistics (the first 10 packets of each flow were skipped as their behavior is different than the behavior of the rest of the flow). The detailed description of our methods and results can be found in [18]. The decision tree generated in this step can be used to classify the traffic in the real network.

VII. Centralized monitoring solution

This paragraph presents the proposed design of the centralized monitoring solution, which can be placed in any point in the network to examine the network QoS.

Because of the heavy load in the high-speed networks, it is not possible to monitor all the flows passing the central point at the same time. Therefore, only the statistics from selected flows can be captured and passed to C5.0. The selection of such flows can be based on two methods: capturing one flow per user and intelligent switching between the flows. From the QoS point of view, it is important to discover the problems with a particular user or to inform the user that the problems experienced by him are the result of problems in the remote network. If it is the user who has the problem, then the problem usually influences all the user's network activity.

Each application has some special requirements regarding the network parameters. When a small congestion occurs, the service level can still be sufficient for P2P file downloads, but Skype communication may be not possible because of big jitter and delays. It is, therefore, not sufficient to monitor one random flow at a time, but we need to monitor a flow which have high quality requirements. Our solution should be built based on the following assumptions:

- Only one flow per user at a time is consistently monitored for QoS.
- Statistics for another random flow per user at a time are passed to C5.0 to discover the application.
- If the application has higher QoS requirements than the currently monitored, switch monitoring to the new flow; if not, stick to the current one.
- If the monitoring of the selected flow discovers problems, start monitoring a few flows at a time to check if this problem lay on the user's side or on the remote side.

Because of the dynamic switching between the flows when determining the application, it is most probable that the system will not be able to capture flows from their beginning. The classifier designed by us, which uses C5.0, is able to determine the application on the basis of the given number of packets from any point in the flow [18].

Monitoring of the QoS can be done in a passive or an active mode. The passive mode relies mostly on time-based statistics, which are obtained directly from the flow passing the measurement point. This way, we can assess the jitter, the burstiness and the transmission speed (both download and upload). Unfortunately, it is not possible to receive the information about the packet loss or the delay for other than TCP streams while using this method. For that reason, additional tools performing active measurements must be involved in the process of estimating the QoS. One option is to use the ping-based approach, as it can measure both the delay and packet loss. Unfortunately, other issues can arise. Ping requests and responses are often blocked by network administrators, or their priority is modified (decreased to save the bandwidth or increased to cheat the users about the quality of the connection). Other options include sending IP packets with various TTL and awaiting *Time Exceeded* ICMP messages, which are usually allowed to be transmitted in all the networks and their priority is not changed. Active measurements must be done in both directions (from the user and from the remote side). The total packet loss and the delay can be calculated as the sum of the delays and the packet losses from both directions of the flow. Furthermore, the knowledge of the direction that causes the problems can be used to assess if the problems are located in the local network or somewhere

outside.

VIII. Conclusion

This paper shows a novel method for assessing the Quality of Service in computer networks. Our approach involves a group of volunteers from the target network to participate in the initial training of the system, and later in the self-learning process. The accurate data obtained from the volunteers are used by the C5.0 MLA to create the per-application profiles of the network traffic as classification decision trees. The centralized measurement system uses the decision trees to determine the applications associated with the flows passing through the measurement point. This knowledge allows us to precisely define the QoS requirements for each particular flow. To assess the QoS level two methods are proposed: the passive and the active one. Two elements of the system are already built and tested: volunteer-based system for obtaining the training data and the classification system based on C5.0. Further research could focus on the design and implementation of the other parts of the system.

References

[1] Gerhard Haßlinger. Implications of Traffic Characteristics on Quality of Service in Broadband Multi Service Networks. In *Proceedings of the 30th EUROMICRO Conference (EUROMICRO'04)*, pages 196–204. IEEE, Rennes, France, September 2004. DOI: 10.1109/EURMIC.2004.1333372.

[2] Thomas M. Chen and Lucia Hu. Internet Performance Monitoring. *Proceedings of the IEEE*, 90(9):1592–1603, September 2002. DOI: 10.1109/JPROC.2002.802006.

[3] LIRNEasia Broadband QoSE Benchmarking project, 2008. [Online]. Available: http://lirneasia.net/projects/2008-2010/indicators-continued/broadband-benchmarking-qos-20/.

[4] K. v. M. Naidu, Debmalya Panigrahi, and Rajeev Rastogi. Detecting Anomalies Using End-to-End Path Measurements. In *Proceedings of the 27th Conference on Computer Communications IEEE INFOCOM 2008*, pages 16–20. IEEE, Phoenix, Arizona, USA, April 2008. DOI: 10.1109/INFOCOM.2008.248.

[5] Aboagela Dogman, Reza Saatchi, and Samir Al-Khayatt. An Adaptive Statistical Sampling Technique for Computer Network Traffic. In *Proceedings of the 7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP 2010)*, pages 479–483. IEEE, Newcastle upon Tyne, England, United Kingdom, July 2010.

[6] Baek-Young Choi, Jaesung Park, and Zhi-Li Zhang. Adaptive Random Sampling for Traffic Load Measurement. In *Proceedings of the IEEE International Conference on Communications (ICC'03)*, volume 3, pages 1552–1556. IEEE, Anchorage, Alaska, USA, May 2003. DOI: 10.1109/ICC.2003.1203863.

[7] Shao Liu, Mung Chiang, Mathias Jourdain, and Jin Li. Congestion Location Detection: Methodology, Algorithm, and Performance. In *Proceedings of the 17th International Workshop on Quality of Service (IWQoS 2009)*, pages 1–9. IEEE, Charleston, South Carolina, USA, July 2009. DOI: 10.1109/IWQoS.2009.5201404.

[8] Jun Li, Shunyi Zhang, Yanqing Lu, and Junrong Yan. Real-time P2P traffic identification. In *Proceedings of the IEEE Global Telecommunications Conference (IEEE GLOBECOM 2008)*, pages 1–5. IEEE, New Orleans, Louisiana, USA, December 2008. DOI: 10.1109/GLOCOM.2008.ECP.475.

[9] Riyad Alshammari and A. Nur Zincir-Heywood. Machine Learning based encrypted traffic classification: identifying SSH and Skype. In *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA 2009)*, pages 1–8. IEEE, Ottawa, Ontario, Canada, July 2009. DOI: 10.1109/CISDA.2009.5356534.

[10] Sven Ubik and Petr Žejdl. Evaluating application-layer classification using a Machine Learning technique over different high speed networks. In *Proceedings of the Fifth International Conference on Systems and Networks Communications (ICSNC)*, pages 387–391. IEEE, Nice, France, August 2010. DOI: 10.1109/ICSNC.2010.66.

[11] Ying Zhang, Hongbo Wang, and Shiduan Cheng. A Method for Real-Time Peer-to-Peer Traffic Classification Based on C4.5. In *Proceedings of the 12th IEEE International Conference on Communication Technology (ICCT)*, pages 1192–1195. IEEE, Nanjing, China, November 2010. DOI: 10.1109/ICCT.2010.5689126.

[12] Jing Cai, Zhibin Zhang, and Xinbo Song. An analysis of UDP traffic classification. In *Proceedings of the 12th IEEE International Conference on Communication Technology (ICCT)*, pages 116–119. IEEE, Nanjing, China, November 2010. DOI: 10.1109/ICCT.2010.5689203.

[13] Riyad Alshammari and A. Nur Zincir-Heywood. Unveiling Skype encrypted tunnels using GP. In *Proceedings of the 2010 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, Barcelona, Spain, July 2010. DOI: 10.1109/CEC.2010.5586288.

[14] Kartheepan Balachandran, Jacob Honoré Broberg, Kasper Revsbech, and Jens Myrup Pedersen. Volunteer-Based Distributed Traffic Data Collection System. In *Proceedings of the 12th International Conference on Advanced Communication Technology (ICACT 2010)*, volume 2, pages 1147–1152. IEEE, Phoenix Park, PyeongChang, Korea, February 2010.

[15] Kartheepan Balachandran and Jacob Honoré Broberg. Volunteer-Based Distributed Traffic Data Collection System. Master's thesis, Aalborg University, Department of Electronic Systems, Denmark, June 2010.

[16] Tomasz Bujlow, Kartheepan Balachandran, Tahir Riaz, and Jens Myrup Pedersen. Volunteer-Based System for classification of traffic in computer networks. In *Proceedings of the 19th Telecommunications Forum TELFOR 2011*, pages 210–213. IEEE, Belgrade, Serbia, November 2011. DOI: 10.1109/TELFOR.2011.6143528.

[17] Tomasz Bujlow, Kartheepan Balachandran, Sara Ligaard Nørgaard Hald, Tahir Riaz, and Jens Myrup Pedersen. Volunteer-Based System for research on the Internet traffic. *TELFOR Journal*, 4(1):2–7, September 2012. Accessible: http://journal.telfor.rs/Published/Vol4No1/Vol4No1.aspx.

[18] Tomasz Bujlow, Tahir Riaz, and Jens Myrup Pedersen. A method for classification of network traffic based on C5.0 Machine Learning Algorithm. In *Proceedings of ICNC'12: 2012 International Conference on Computing, Networking and Communications (ICNC): Workshop on Computing, Networking and Communications*, pages 244–248. IEEE, Maui, Hawaii, USA, February 2012. DOI: 10.1109/ICCNC.2012.6167418.

[19] Jason But, Philip Branch, and Tung Le. Rapid identification of BitTorrent Traffic. In *Proceedings of the 35th Annual IEEE Conference on Local Computer Networks (LCN)*, pages 536–543. IEEE, Denver, Colorado, USA, October 2010. DOI: 10.1109/LCN.2010.5735770.

[20] Li Jun, Zhang Shunyi, Lu Yanqing, and Zhang Zailong. Internet Traffic Classification Using Machine Learning. In *Proceedings of the Second International Conference on Communications and Networking in China (CHINACOM '07)*, pages 239–243. IEEE, Shanghai, China, August 2007. DOI: 10.1109/CHINACOM.2007.4469372.

[21] Yongli Ma, Zongjue Qian, Guochu Shou, and Yihong Hu. Study of Information Network Traffic Identification Based on C4.5 Algorithm. In *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, pages 1–5. IEEE, Dalian, China, October 2008. DOI: 10.1109/WiCom.2008.2678.

[22] Wei Li and Andrew W. Moore. A Machine Learning Approach for Efficient Traffic Classification. In *Proceedings of the Fifteenth IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS'07)*, pages 310–317. IEEE, Istanbul, Turkey, October 2007. DOI: 10.1109/MASCOTS.2007.2.

[23] Is See5/C5.0 Better Than C4.5, 2009. [Online]. Available: http://www.rulequest.com/see5-comparison.html.