A Method for Assessing Quality of Service in Broadband Networks

Bujlow, Tomasz; Riaz, M. Tahir; Pedersen, Jens Myrup

Published in:
2012 14th International Conference on Advanced Communication Technology (ICACT)

Publication date:
2012

Document Version
Accepted author manuscript, peer reviewed version

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 Quality of Service (QoS) in high-speed Internet infrastructure is a challenging task. However, precise assessments must take into account the fact that the requirements for the given quality level are service-dependent. Backbone QoS monitoring and analysis requires processing of large amount of the data and knowledge of which kind of application the traffic belongs to. To overcome the drawbacks of existing methods for traffic classification we proposed and evaluated a centralized solution based on C5.0 Machine Learning Algorithm (MLA) and decision rules. The first task was to collect and provide C5.0 high-quality training data, divided into groups corresponding to different types of applications. It was found that 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. To collect training data a new system was developed, in which the major role is performed by volunteers. Client applications installed on their 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 Quality of Service (QoS) level in broadband networks, based on our Volunteer-Based System for collecting the training data, Machine Learning Algorithms for generating the classification rules and application-specific rules for assessing the QoS level. We combine both passive and active monitoring technologies. The paper evaluates different implementation possibilities, presents the current implementation of particular parts of the system, their initial runs and 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 regarding computer networks is how to measure the performance, when different types of networks are merged together. In the last few years the data-oriented networks evolved into converged structures, in which 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 POTS (Plain Old Telephone Service) lines used to provide analog (voice telephony), or digital (ADSL, PBX, ISDN) services – and nowadays also of mobile and wireless networks. There are numerous methods for Quality of Service (QoS) measurement in current use, providing the measurements both on 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 conversation, a delay of about 0.1 s is hardly noticeable, but 0.25 s delay means an essential degradation of transmission quality, and more than 0.4 s is considered as severely disturbing [1].

Therefore, in order to provide detailed information about quality level for the given service in the core of the network, we need to know, what kind of data is 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 memory. Furthermore, numerous privacy and confidentiality issues can arise. A solution to this problem can be the Machine Learning Algorithms (MLAs), which use previously generated decision rules based on some statistical information about the traffic. High efficiency in using available resources is associated with accuracy of above 95 %. 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 solve was finding a way to collect high-quality training statistics.

In order to collect the necessary statistics and generate 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 system sockets. Information about each packet belonging to the flow is also collected. Our volunteer-based system guarantees obtaining of precise and detailed data sets about the network traffic. These data sets can be successfully used to generate statistics used as an input to train MLAs and generate accurate decision rules.

The knowledge about kind of application to which the traffic belongs obtained from MLAs can be used together with traffic

requirements for the 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 can be assessed directly on the basis of information from the captured traffic. To assess delay and packet loss 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 design and implementation of the system, while Section VIII summarizes the most important points.

II. Related work

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

The lack of centralized administration makes it difficult to impose a common measurement infrastructure or protocol. For example, 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 [4]. Routers and traffic analyzers provide passive single-point measurements. They do not measure performance directly, but traffic characteristics are strongly correlated with 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 trying to process packets. It forces to introduce packet sampling techniques and therefore also the possibility of inaccuracies. Even at 1 Gbit/s, the measurement can result in an enormous amount of data to process and store within a 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. Path anomaly detection algorithm was proposed in [5]. 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 [5]. 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.

Next proposals how to sample network traffic in efficient way were made on the basis of adaptive statistical sampling techniques and they are presented in [6] and [7].

If a congestion is detected, from users' perspective it is very important to know, if it is local or remote. If the link experiences local congestion, the user may be able to perform certain actions, e.g. shut down a bandwidth heavy local application 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 only benefits high-priority flows competing for that link, which are most probably flows from other users. Since this altruistic behavior is not desirable, the low priority TCP only needs to back off, when the congested link is local [8].

Detecting the location of congestion is a challenging problem due to several reasons. First of all we cannot send many probing packets, causing too much overhead and even expanding the congestion. Secondly, without router support, the only related signals to end applications are packet losses and delays. If packet losses were completely synchronized (packet drops from all the flows), then the problem would be trivial. In the reality, packet loss pattern is partially synchronized [8]. Authors of [8] attempt to solve congestion location detection problem using synchronization of loss and delay behaviors across multiple TCP sessions in the area controlled by the same local gateway. If many flows see synchronized congestion, then the local link is the congested link. This is because 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 seeing congestion, authors perform algorithm based on queuing delay patterns. If the local link is congested, typically most flows will experience high delays at a similar level. Otherwise the congestion is remote [8].

Traffic can be profiled according to protocol composition. Usually predominance of TCP traffic is observed (around 95 % of the traffic mix). When congestion occurs, TCP sources respond by reducing their offered load, whereas UDP sources do not. It results in a 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, packet loss increases dramatically (and TCP flows become dominated by retransmission timeouts) [2]. Packet sizes provide insight into the type of packet, e.g. short 40-44 bytes packets are usually TCP acknowledgment or TCP control segments (SYN, FIN or RST) [2].

Active methods for QoS monitoring raise three major concerns. First, the introduction of test traffic will increase the network load, which can be viewed as an overhead cost for active methods. Second, test traffic can affect measurements. Third, traffic entering ISP can be considered as invasive and

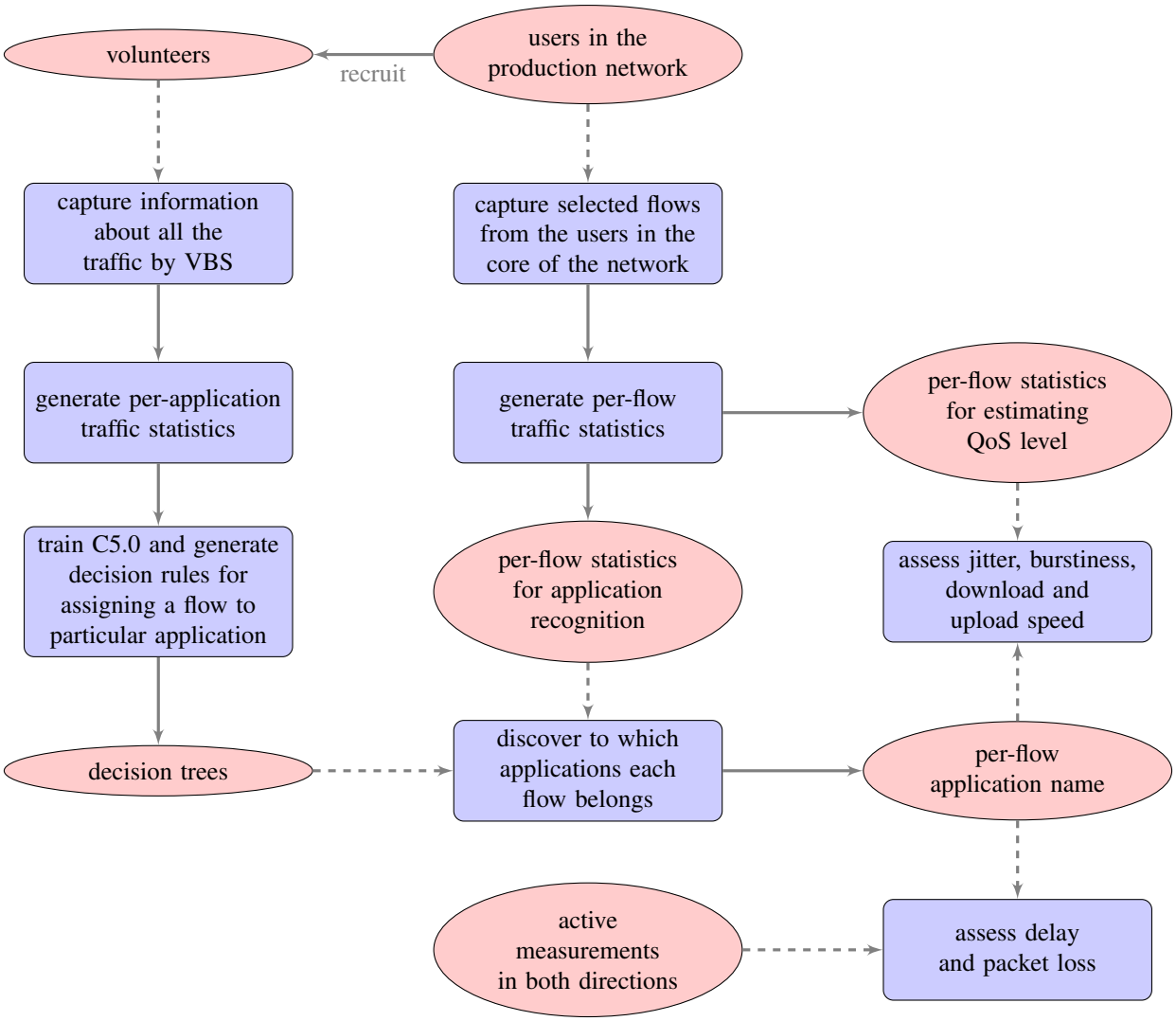

Figure 1. Overview of the system architecture

discarded or assigned to a low-priority class [2].

Within an administrative domain performance can be actively monitored (but not across the entire Internet) using the data-link layer protocol below IP, like operations, administration and maintenance (OAM) procedure in ATM and MPLS networks. As a result it is often desirable to measure performance at the IP layer using 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 IP packet header) [2].

Although round-trip times measured by ping are important, ping is unable to measure 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 in many ISP 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, a returned ICMP message may not follow the same path as the UDP packet that triggered it [2].

Although 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 or TCP, that will send traffic through the Internet. All the parameters (delay, loss, throughput, etc) are measured from the test traffic. This approach has one major drawback - custom software needs to be installed at 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

Overview of the system architecture is shown on Figure 1. Subsequent paragraphs contain 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 relevant traffic information and submits the data to the server. On the server these data is used to generate per-application traffic statistics. C5.0 Machine Learning Algorithm uses these statistics to learn how to distinguish between different types of applications and generate classification rules (decision trees).

In order to assess network QoS level in the center of the network for particular users we needed to find a method to capture the relevant traffic. The challenging task is to process significant amount of traffic in the high-speed networks. When the relevant flows are captured, per-flow statistics need to be generated. There are two kind of statistics generated at this step: used for determining the kind of application associated to that flow, and 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 to which application the flow belongs. Then, on the basis of the kind of the application the system determines ranges of values of the relevant QoS parameters. The last step is to check if the current values (obtained from flow statistics or in the active way) match the expected ones. If not, 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 [9]. 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 [10], [11]. 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 [9], [12], [13]
- Deep Packet Inspection (DPI) [14] is slow, requires a lot of processing power [9], [12] and privacy and confidentiality issues can appear [9]. It is not possible to use DPI to recognize encrypted traffic

Therefore we decided to develop a system based on volunteers, which captures the flows together with the application name taken from Windows or Linux sockets. Architecture and prototype were described and analyzed in [15] and [16], and our current implementation in [17]. 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 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 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 statistical profiles of flows for different applications. Therefore we developed a tool for calculating statistics for several traffic attributes for each flow record in the database fulfilling our requirements. In our small-scale prototype for tests we decided to limit 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 on the current conditions in the network (like for example congestion). 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 information, if the port is well-known or dynamic. This way we avoid to construct 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 traffic in computer networks. The most widely used MLA classifiers are C4.5 [9] and its modified Java implementation called J48 [12], [19]. Based on statistical analysis MLAs have the ability to assign a particular class (like P2P) even to traffic generated by unknown applications [9]. It was also proven in [19] that the statistical parameters for encrypted and unencrypted traffic produced by the same application are similar and therefore encrypted payload does not influence results of training or classification. Accuracy of classification by MLAs was claimed to be over 95 % [9]–[11], [13], [14], [20]–[22]. Analysis of the related work can be found in [18].

It was found in [11] that results of the classification are most accurate when the classifier was trained in the same network as the classification process was performed. This can be due to different parameters, which are constant in the particular network, but differ among various networks. A good example can be 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 deliver data used for training the classifier constantly improving its accuracy, while all the users can be

monitored in the core using accurate 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 another operating system or devices than supported (like MacOS, Apple or Android smartphones), he will still be able to be monitored in the core of the network because of rules created on the basis of data collected from the other users.

Our system uses C5.0 MLA, which is a successor of C4.5. It is proven to has many advantages over its ancestor, like higher, accuracy, possibilities to use boosting, pruning, weighting and winnowing attributes. Furthermore, the time to generate the decision tree or rules rapidly decreased [23]. In order to test efficiency of C5.0 we performed a set of tests during which we used different training and classification options. The training statistics were obtained from data provided by our VBS. During our research we found relevant set of arguments and discovered that the best results were obtained using boosted classifier. Average accuracy fluctuated between 99.3 % and 99.9 % depending on number of training and test cases and amount of data from each case was made. It is worth to mention that in our experiment we considered only 7 different group of applications and only flows longer than 15 packets. Flow length limitation was done because we needed to have at least 5 packets to generate the statistics (first 10 packets of each flow were skipped as their behavior is different than the rest of the flow). Detailed description of our methods and results can be found in [18]. Decision tree generated in this step can be used to classify the traffic in the real network.

VII. A centralized monitoring solution

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

Because of 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 statistics from selected flows can be captured and passed to the C5.0. 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 problems with a particular user or to inform the user that problems experienced by him are results of problems in the remote network. If it is the user who has the problem, then the problem usually influences all user's network activity.

Each application has some special requirements regarding network parameters. When a small congestion occurs the service level can be still 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 a flow having 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 currently monitored, switch monitoring to the new flow; if not, stick to the current
- if monitoring of the selected flow discover problems, start monitoring few flows at a time to check if this problem lay on the user's side or on the remote side

Because of 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. Designed by us classifier using C5.0 is able to determine the application on the basis of given number of packets from any point of the flow [18].

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

VIII. Conclusion

The paper shows a novel method for assessing Quality of Service in computer networks. Our approach involves a group of volunteers from the target network to participate in initial training of the system, and later in the system self-learning process. Accurate data obtained from the volunteers are used by C5.0 MLA to create per-application profile of the network traffic as classification decision tree. The centralized measurement system uses the decision tree to determine application associated with flows passing through the measurement point. This knowledge allows to define precisely QoS requirements for the particular flow. To assess the QoS level two methods are proposed: passive and active. 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 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*, Proceedings of the 30th EUROMICRO Conference (EUROMICRO'04), IEEE Computer Society 2004.

[2] Thomas M. Chen, *Internet Performance Monitoring*, Proceedings of the IEEE, vol. 90, no. 9, September 2002, pp. 1592–1603.

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

[4] Kartheepan Balachandran, Jacob Honore Broberg, Kasper Revsbech, Jens Myrup Pedersen, *Volunteer-Based Distributed Traffic Data Collection System*, Feb. 7-10, 2010 ICACT 2010, pp. 1147–1152.

[5] K. v. M. Naidu, Rajeev Rastogi, Bell Labs Research India, Bangalore, *Detecting Anomalies Using End-to-End Path Measurements*, IEEE INFOCOM 2008 proceedings, 2008, pp. 16–20.

[6] A. Dogman, R. Saatchi, S. Al-Khayatt, *An Adaptive Statistical Sampling Technique for Computer Network Traffic*, IEEE CSNDSP 2010, pp. 479–483.

[7] Baek-Young Choi, Jaesung Park, Zhi-Li Zhang, *Adaptive Random Sampling for Traffic Load Measurement*, IEEE International Conference on Communications, IEEE 2003, pp. 1552–1556.

[8] Shao Liu, Mung Chiang, Mathias Jourdain, Jin Li, *Congestion Location Detection: Methodology, Algorithm, and Performance*, 17th International Workshop on Quality of Service, IEEE 2009.

[9] Jun Li, Shunyi Zhang, Yanqing Lu, Junrong Yan, *Real-time P2P traffic identification*, IEEE GLOBECOM 2008 PROCEEDINGS.

[10] Riyad Alshammari, A. Nur Zincir-Heywood, *Machine Learning based encrypted traffic classification: identifying SSH and Skype*, Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009).

[11] Sven Ubik, Petr Žejdl, *Evaluating application-layer classification using a Machine Learning technique over different high speed networks*, 2010 Fifth International Conference on Systems and Networks Communications, IEEE 2010, pp. 387–391.

[12] Ying Zhang, Hongbo Wang, Shiduan Cheng, *A Method for Real-Time Peer-to-Peer Traffic Classification Based on C4.5*, 12th IEEE International Conference on Communication Technology, IEEE 2010, pp. 1192–1195.

[13] Jing Cai, Zhibin Zhang, Xinbo Song, *An analysis of UDP traffic classification*, 12th IEEE International Conference on Communication Technology, IEEE 2010, pp. 116–119.

[14] Riyad Alshammari, A. Nur Zincir-Heywood, *Unveiling Skype encrypted tunnels using GP*, IEEE Congress on Evolutionary Computation (CEC), IEEE 2010.

[15] Kartheepan Balachandran, Jacob Honoré Broberg, Kasper Revsbech, Jens Myrup Pedersen, *Volunteer-based distributed traffic data collection system*, Feb. 7-10, 2010 ICACT 2010, pp. 1147–1152.

[16] Kartheepan Balachandran, Jacob Honoré Broberg, *Volunteer-based distributed traffic data collection system*, Master Thesis at Aalborg University, Department of Electronic Systems, June 2010.

[17] Tomasz Bujlow, Kartheepan Balachandran, Tahir Riaz, Jens Myrup Pedersen, *Volunteer-Based System for classification of traffic in computer networks*, 19th Telecommunications Forum TELFOR 2011, IEEE 2011, pp. 210–213.

[18] Tomasz Bujlow, Tahir Riaz, Jens Myrup Pedersen, *A method for classification of network traffic based on C5.0 Machine Learning Algorithm*, to appear in International Conference on Computing, Networking and Communications (ICNC 2012).

[19] Jason But, Philip Branch, Tung Le, *Rapid identification of BitTorrent Traffic*, 35th Annual IEEE Conference on Local Computer Networks, IEEE 2010, pp. 536–543.

[20] Jun Li, Shunyi Zhang, Yanqing Lu, Zailong Zhang, *Internet Traffic Classification Using Machine Learning*, Second International Conference on Communications and Networking in China, CHINACOM '07, 2007, pp. 239–243.

[21] Yongli Ma, Zongjue Qian, Guochu Shou, Yihong Hu, *Study of Information Network Traffic Identification Based on C4.5 Algorithm*, 4th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE 2008.

[22] Wei Li, Andrew W. Moore, *A Machine Learning Approach for Efficient Traffic Classification*, Proceedings of the Fifteenth IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS'07), IEEE 2008, pp. 310–317.

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