IP Traffic Classifiers Applied to DiffServ Networks

Michael Taynnan Barros*†,Reinaldo Gomes‡, Marcelo Sampaio de Alencar‡§ and Anderson F. B. F. da Costa ¶

*Telecommunication Software and Systems Group (TSSG), Waterford, Ireland
†Waterford Institute of Technology (WIT), Waterford, Ireland
‡Federal University of Campina Grande (UFCG) , Campina Grande, Brazil
§Institute for Advanced Studies in Communications (Iecom), Campina Grande, Brazil
¶Federal Institute of Paraiba (IFPB), Campina Grande, Brazil
Email: mbarros@tssg.org, reinaldo@dsc.ufcg.edu.br, malencar@iecom.org.br, anderson@ifpb.edu.br

***Abstract*—The future Internet scenario consists of a higher number of users and applications, which demand more resources from the communication infrastructure. Techniques for providing performance and scalability, such as Traffic Engineering (TE), will always be necessary even if the transmission rate is very high, because of such demands. Quality of Service is one of the solutions that can be used to improve the traffic engineering in the Internet, with the most referenced architecture: DiffServ. In general, TE needs traffic classification to accurately identify the input traffic and manage it properly. However, the current DiffServ port traffic classifier is considered outdated. This paper presents a performance evaluation of machine learning traffic classification solutions applied to DiffServ, and investigates their benefits on network performance. For a backbone network with 40 nodes, the performance of the network can increase up to 15% for both data and voice traffic.**

I. Introduction

Nowadays, the world has a communication infrastructure called Internet, which has produced economic and social effects, and whose importance is unquestionable [1]. However, the Internet faces technical problems, related to the establishment and guarantee of connections. Studies indicate a possible change of the current Internet communication paradigm in the near future [2]. The increase in the number of users is a relevant concern, reaching six billion users in 2020, plus more 50 billion entities (including sensors, security and control software) [3]. This scenario requires techniques to handle a high number of users and maintain the performance of the data traffic applications, such as Traffic Engineering (TE) and Quality of Service (QoS).

In a nutshell, TE establishes the parameter and the operational point for the three aspects of the network: the system of demands (traffic), the system of restrictions (interconnected network elements) and the system of response (network protocols and processes), in an operational context [4]. The most common TE technique is QoS, which guarantees reliability of information transmission according to the application requirements, as well as adequate performance levels [5]. Many obstacles have prevented a large scale implementation of QoS [6], and its maturity to handle commercial requirements. Nevertheless, there is still some research effort to improve QoS, specifically for mobile networks [7] [8].

In the DiffServ QoS architecture, the traffic classifier influences directly the performance of the architecture. It is responsible to identify the type of service of the current input packet and forward the results to the traffic conditioner, in which the packet should be managed to obtain a better performance in the network.

The RFC 4594 indicates that the packet must be marked by the application with some identification. If not, the packet should be identified by any DiffServ router, task which is performed by a traffic classifier. The current DiffServ classification approach is based on port, a technique which has an average performance of 70% of accuracy, not efficient enough to support all types of traffic, such as P2P, FTP, streaming, games, chat and security [9] [10] [11] [12]. This accuracy problem causes a performance degradation of the application data traffic as well as the network, causing a higher packet drop probability. The problem can be solved applying machine learning classifiers (MLC), whose performance is 20% superior to the port classification.

The MLC techniques require flow analysis, instead of packet processing, and is considered a scalable approach because it is independent of the packets, but depends on the traffic characteristics. This approach was mentioned by the DiffServ task force, before it closed down in 2006 [13]. This paper shows how the updated DiffServ, with a more accurate classifier, can improve both the data and voice applications performance in terms of packet delivery ratio, end-to-end delay and jitter.

The use of machine learning increased the performance of the network in 15%. This approach forces the increase of the packet delivery ratio. Also, with more accurate classifiers, the queue works faster, providing a low end-to-end delay. The paper presents enough evidence encouraging new experiments in test-beds, and the results indicate a powerful and easy implementation approach to deal with current problems in the Internet.

This paper is organized as follows. Section II discusses the DiffServ accuracy problem. Section III presents an evaluation of the studied IP classifiers based on machine learning. Section IV discusses the experimentation process of a improved DiffServ network, and is followed by the results in Section V. Finally, the conclusions are presented in Section VI.

978-1-4799-3755-4/13/$31.00 ©2013 IEEE

II. Problem Statement

TCP streams are divided into groups according to each traffic profile. A set of TCP stream groups is denoted by Υ_{TCP}, and the set of UDP stream groups by Υ_{UDP}. Consider $\Upsilon = \Upsilon_{TCP} \bigcup \Upsilon_{UDP}$ denoting the set of all flow groups $l \in \Upsilon = \{1, \cdots, L\}$, in which each flow group consists of n_l similar flows.

In this model, the TCP congestion control follows a differential equation, as a function of the aggregate streams, as described in [14]. This results in a stable point for the average TCP transmission rate (λ), which depends on the *Round Trip Time* (RTT) and on the packet loss probability p

$$\lambda = \frac{1}{RTT}\sqrt{\frac{2(1-p)}{p}}. \tag{1}$$

Formula 1 relates the congestion control with the Per Hop Hehavior (PHB), by means of the packet loss probability, and calculates the average transmission rate for TCP connections. The packet loss probability depends of the stream group l, because each PHB group (whether BE, AF or EF) has a different setup and a different drop of precedence level. The formulas must be formulated separately for each TCP traffic profile [15]. For the sake of brevity, the model extension of different TCP traffic profile is omitted in this paper.

A. Classifier Model

The classifier model is implemented with the traffic conditioner. The classifier has an accuracy probability p_a.

Consider a stream in a group $l \in \Upsilon$. Consider that $v(l)$ denotes the traffic flow l intensity. Each stream is divided into three different drop in precedence level $i \in X = \{1, 2, 3\}$ with a traffic conditioner mechanism. The traffic profile for a flow is determined by two traffic intensity thresholds $v_l^{limit,1}$ and $v_l^{limit,2}$, $0 < v_l^{limit,1} < v_l^{limit,2}$. Consider that $\lambda_l(i)$ denotes the traffic intensity of a portion from stream l, which is marked with a drop in precedence level i. The profile and intensity of traffic flow defines $\lambda_l(i)$ using 2, 3 and 4.

$$\lambda_l(1) = \begin{cases} v(l) & , \quad 0 \leq v(l) \leq v_l^{limit,1} \\ v_l^{limit,1} & , \quad v_l^{limit,1} < v(l) \leq v_l^{limit,2} \\ v_l^{limit,1} & , \quad v(l) > v_l^{limit,2} \end{cases} \tag{2}$$

$$\lambda_l(2) = \begin{cases} 0 & , \quad 0 \leq v(l) \leq v_l^{limit,1} \\ v(l) - v_l^{limit,1} & , \quad v_l^{limit,1} < v(l) \leq v_l^{limit,2} \\ v_l^{limit,2} - v_l^{limit,1} & , \quad v(l) > v_l^{limit,2} \end{cases} \tag{3}$$

$$\lambda_l(3) = \begin{cases} 0 & , \quad 0 \leq v(l) \leq v_l^{limit,1} \\ 0 & , \quad v_l^{limit,1} < v(l) \leq v_l^{limit,2} \\ v(l) - v_l^{limit,2} & , \quad v(l) > v_l^{limit,2} \end{cases} \tag{4}$$

For a given input stream, one must compute its traffic intensity. In this regard, a probability is assigned to each $i \in X$. For the right $\lambda_l(i)$ the probability is p_a, and for the other the probability is $\frac{1-p_a}{2}$. One can model this process as a roulette wheel selection (RWS), as in Equation 5, in which each function $f(\cdot)$ returns the assigned probability of each i. The value MAX$\{P_{sel}(i)\}$ is the chosen one.

$$P_{sel}(i) = \frac{f(i)}{\sum_{i=1}^{|X|} f(i)}, \tag{5}$$

in which $|X|$ is the cardinality of the set X.

The traffic conditioner divides the stream into precedence levels

$$v(l) = \sum_{i=1}^{|X|} \lambda_l(i).$$

Therefore, the distribution of the flow, as the precedence levels drop, can be expressed as

$$\sigma_l(i) = \frac{\lambda_l(i)}{v(l)}, \forall i \in X. \tag{6}$$

B. System Model

The system model is presented in Figure 1. Consider that $v(l)$ denotes the transmission rate of an associated stream with group l. The flows are classified and after they are marked as class b_l, $b_l \in B$ and, thereafter, stored in a buffer with a queue in the PHB node. Using the definition from Equation 6, the total traffic intensity with drop in precedence level i in a buffer b is

$$\lambda^b(i) = \sum_{l:b_l=b} n_l \sigma_l(i) v(l). \tag{7}$$

The packet drop probability $p^b(i)$ can be determined for each PHB behavior (AF, BE and EF) with different drop in precedence levels, as discussed in the previous section. The packet loss probability $q(l)$ of a flow is defined as

$$q(l) = \sum_{i=1}^{|X|} \sigma_l(i) p^{b_l}(i). \tag{8}$$

For each TCP group the transmission rate is defined by Formula 1, which relates the packet drop probability to the average transmission rate of a TCP flow.

$$v(l) = \frac{1}{RTT}\sqrt{\frac{2(1-q(l))}{q(l)}}, l \in L_{TCP}. \tag{9}$$

The transmission rate $v(l)$ for an UDP flow is fixed and independent of the packet loss probability, and can be defined as

$$v_{eff}(l) = v(l)(1 - q(l)), l \in L. \tag{10}$$

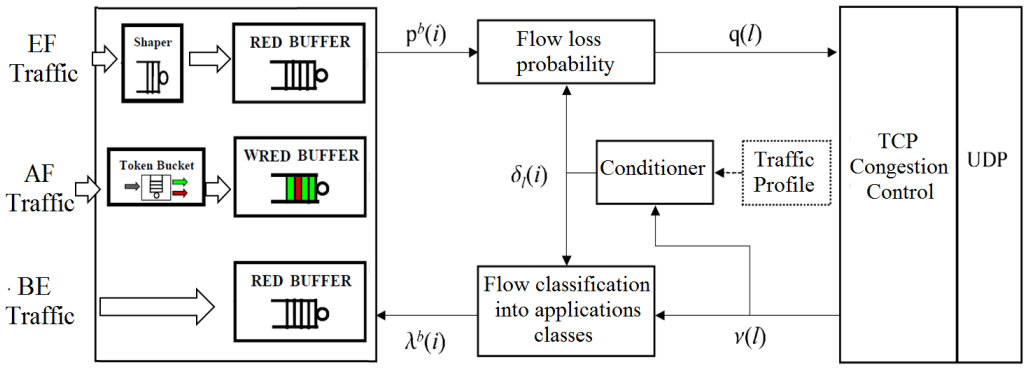

Fig. 1. System Model.

C. Accuracy Problem

The accuracy problem is referred to the estimated value of p_a. With the current classification methods, in the DiffServ, this value is around 0.7 [9]. This number has been updated to the range 0.9 to 0.95, with the usage of traffic classifiers based in machine learning [10] [16] [11] [9] [12]. This improvement indicates that 25% of the flow that passes through an edge router is correctly classified, which increases the network QoS level. The objective of this paper is to present evidences confirming this statement, as well as, its importance in the DiffServ performance analysis.

III. Evaluation of IP Traffic Classifiers

Several classifiers are identified and investigated in this paper, based on machine learning that satisfy the restrictions imposed by the DiffServ architecture. Such constraints include time requirements for training and classification time for flows, variables which are related to the classification accuracy, and the classifier qualitatively reports the efficiency of classification. Works like [12], concluded that the classifiers Decision Trees and Bayesian Networks satisfy the restrictions already mentioned. Given that a calibration study was conducted and presented in [17], an evaluation is presented again and the conclusions are obtained.

The investigated classifiers include: Decision Trees, Naive Bayes, Bayesian Networks and Neural Networks. The results are analyzed in terms of accuracy, precision, recall, F-measure, training time and time sorting by flow.

Five samples of Internet IP traffic, which are part of this evaluation, were collected in the edge of backbone networks obtained by [18] and [19]. They have temporal and geographic variety, within the period 2002 to 2011 located at the United States and Japan. The samples are: Link CAIDA OC48 USA in the years 2002 and 2003, CAIDA Chicago Backbone Network-USA, in 2010, San Francisco Backbone Network CAIDA-USA in 2011, and JP-WIDE backbone network (M) in 2011. Figure 2 shows all flows in quantity, mapping their respective applications and categories by the colors, being the ground-truth of the samples used.

Figure 3 presents the values of accuracy, precision, recall and F-measure for all the evaluated techniques in this

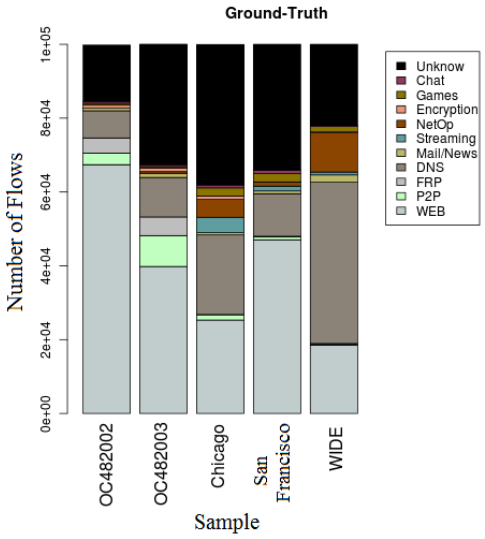

Fig. 2. Ground Truth mapping the number of flows, samples and application.

experiment. Techniques such Decision Trees and Bayesian Networks have higher values for all the four metrics. In terms of performance, these techniques are considered the best in the experiments, followed by Naive Bayes and Neural Networks.

Previously, Naive Bayes was rejected as a viable classifier to DiffServ specifically because of its low performance, as shown in [12]. The increase in performance of the technique Naive Bayes shows the benefits of calibration, since it presents results very close to both Decision Trees and Bayesian Networks techniques. In the case of Neural Networks, the calibration process did not bring any benefit. However, a certain gain exists with the use of filters, as shown in the first study presented in this chapter.

Figure 3 presents the results of the average training time for classifiers studied. The classifier Neural Networks presents the greatest training time compared to other techniques, which can make usage of neural networks for some applications of traffic classification.

The time of classification per flow is important because high processing time of packets degrades the performance of appli-

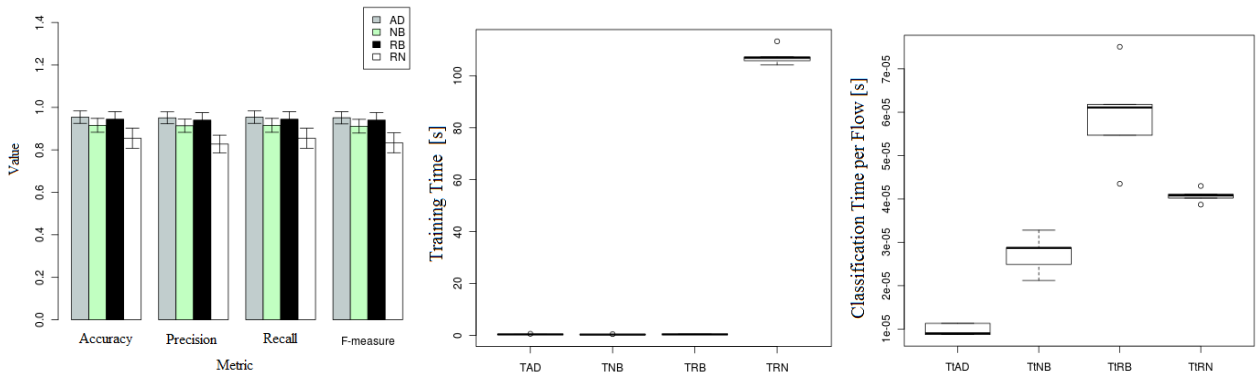

Fig. 3. Results of accuracy, precision, recall and F-measure for IP classifiers evaluation.

cation in terms of end-to-end delay.. The performance analysis is based on the flows that have least possible delay, e.g. video conferencing applications, categorized here as streaming. In [20], the authors present a maximum value of $100ms$ end-to-end delay for this specific type of application. Therefore, the classification process should identify the flow in a much shorter time than the value mentioned, since routers between source and destination nodes add extra propagation and processing times. Figure 3 shows time values of classification time per flow for each evaluated classifier. All classifiers have values below $100ms$, about 1000 times smaller than the value established by non-elastic applications, such as streaming. The classification time per flow for all the classifiers is not high enough to interfere in the performance of the application in terms of end-to-end delay.

It was concluded that with the addition of the filtering and calibration techniques, all evaluated classifiers exhibited satisfactory performance satisfying the DiffServ requirements. All classifiers are applicable to the problem described in Section II.

IV. Experiments

There is a need to provide evidences to the statement that the machine learning classifiers can increase the performance of QoS networks. This is the objective of the evaluation, which intends to measure this possible increase. A simulation model is presented with an experimental design.

In view of a simple analysis for a quick visualization and understanding of the conclusions, the experiments have only one independent variable, which is the number of connections.

The presented analysis is simple although it is a complete one. The following dependent variables were chosen for the performance evaluation: delay, jitter and packet delivery ratio.

The experiment population is voice and data traffic. For the data, the HTTP (Hypertext Transfer Protocol) application is used, because it is the most used application protocol over the Internet. For voice, the VoIP (Voice Over IP) application is used, because it is the most popular one for voice communication over packet switched networks.

Figure 4 presents the used topology, in which a 24 node distribution is used. It also shows the information on connections

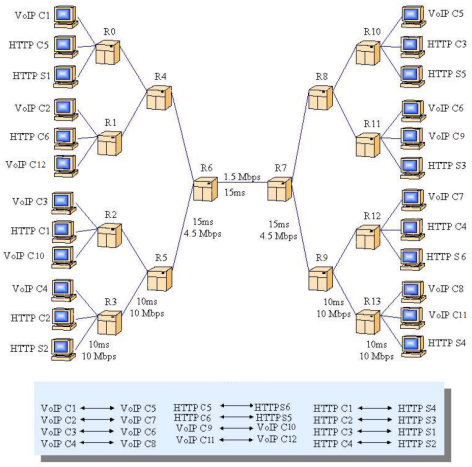

Fig. 4. Used Topology in the experiments with the transmission rate supported in each channel.

and the supported transmission rate for each channel. This topology was designed to fully stress the network equipment and create a bottleneck, which brings the R6 and R7 queue constantly full, affecting directly the applications, principally the delay intolerant ones.

For a higher validation of the results, a well known simulation (Network Simulator 2 – NS2) is used. The simulation scripts are more detailed in [21] and available in [22]. To keep statistical differences low, the number of runs in the simulations was 30 (changing the seed after each run), based in the Central Limit Theorem. The graphs show the results for 95% of statistical confidence intervals.

V. Results and Discussion

A. Packet Delivery Ratio

This subsection presents the Packet Delivery Ratio (PDR) measurement in the experiments. The graphs are presented in Figure 5 (a) for mixed traffic, b) for voice and c) for data), which shows the Packet Delivery Ration (in percentage) versus the number of connections.

The graphs present an increase of the number of delivered packet to their destinies, due the a change in the current

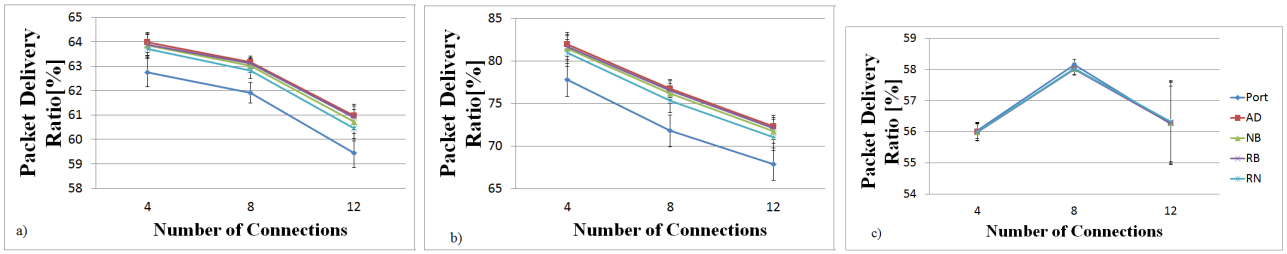

Fig. 5. Packet Delivery Ratio versus the variation of connection number with Port, AD (Decision Tree), RB (Bayesian Nets), NB (Naive Bayes) and RN (Neural Networks) classifiers.

classifier (Port), with the machine learning classifiers (AD, RB, NB and RN). There is a 2% difference for the mixed traffic and 8% for the voice application. For data application there is no significant statistical difference, and one can argue that there is no advantage in changing the current classifier (machine learning) with data traffic relating to PDR.

This difference can be explained as: 1) the TCP has traffic regulatory mechanisms of flow control and congestion control and 2) because of the classification in the routers queue. The Slow Start mechanism decreases or increases the packet transmission rate if there is loss, or not, of packets. This makes a difference in the input rate, and the routers queue becomes more congested. Sequentially, the classifier has an important role in this event because packets with lower priority are dropped. If the classifier wrongly identifies voice packets, they will have lower priority and will be dropped. This explains the fact that more accurate classifiers are more unlikely to make this mistake, in other words, the voice packets will be brought to the beginning of the queue, and will not dropped anymore. With data traffic the importance is different, because data packets have lower priority level.

It can be considered that, if the PDR is high it means that the amount of information is high too. This argument is interesting for voice application, because the audio quality of a call can be increased. And similarly, the packet loss is decreased, presenting evidences for the machine learning classifiers efficiency. The highest difference between data and voice traffic is that voice packets with wrong PHBs are more likely to be dropped.

It is noteworthy that the PDR decreases with an increase of the connections number. This is because the number of packets transmitted in the same media is higher, considering also a bottleneck with a maximum transmission rate of 1.5M*bps*. Thus, the queue will be more frequently full, and the input packets will be dropped.

B. End to end delay

The graphs of end to end delay are shown in Figure 6 (a) for mixed traffic, b) for voice and c) for data). End to end delay is presented in terms of microseconds (ms) as a function of the number of connections. For voice application, the end to end delay must be below of the 100ms threshold. Therefore, the client is unable to notice the delay in a voice stream. The graphs present a satisfactory decrease in the end to end delay, either for the mixed traffic or the voice application, which was an expected result. For the mixed traffic the decrease is around 15% and for voice the gain is even higher, reaching 20% of decrease in delay. Delay values stayed below of 100ms for four connections and eight connections, and slightly higher for 12 connections. While with the use of the current classifier the results indicate that this threshold is extrapolated even for the smallest number of connections. Thus, one can argue that the machine learning classifiers directly influence the results of end to end delay. For data application, the delay remains the same, however this type of application is a delay tolerant traffic.

The explanation for such performance resides in the fact that more packets are marked with right PHBs, which leads to a more efficient treatment of the traffic. Thus, the voice packets that have higher priority will pass through the router queue and reach their destiny as fast as possible.

A decrease in the end to end delay with an increase in data application is verified, which may seen incoherent, but there is an explanation. Based on the increase of drop rate, which was pointed by the PDR, the queue becomes full and starts to drop more packets. This is not computed as an end to end delay event, because the packet was dropped half-way to the destination.

C. Jitter

The graphs of jitter are presented in Figure 7 (a) for mixed traffic, b) for voice and c) for data). Jitter is plotted in ms versus the number of connections. The graphs present a slight difference of jitter with the use of machine learning classifiers. The current network setup influences the values of this metric.

For voice applications, the jitter tends to be lower with an increase in the number of connections. This variation is due to the queue, because when it is full the packets needs to wait longer to be delivered to their destination.

This behaviour is not observed with the data traffic. Because these applications are controlled by TCP and by its flow and congestion control mechanisms, implying a constant change of the packet transmission rate with higher packet loss for higher values of connections. Therefore, if this mechanism changes directly the delay values, the jitter will attain higher values.

VI. Conclusion and Future Work

This paper presents a performance evaluation of the Diff-Serv backbone network with new traffic classification tech-

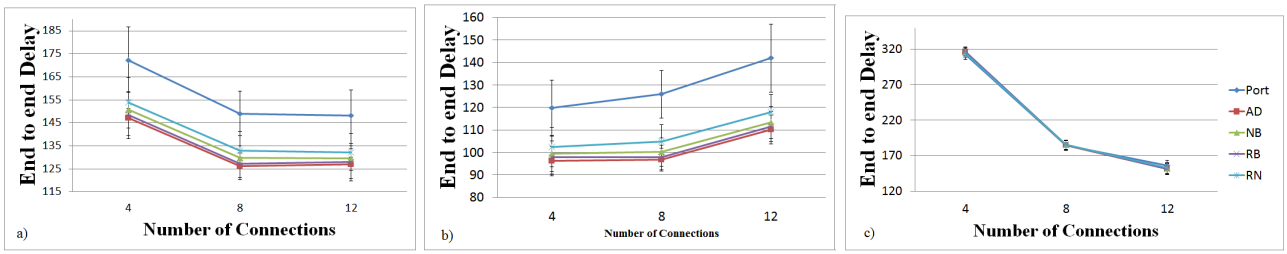

Fig. 6. End to end delay versus the variation of connection number with Port, AD (Decision Tree), RB (Bayesian Nets), NB (Naive Bayes) and RN (Neural Networks) classifiers.

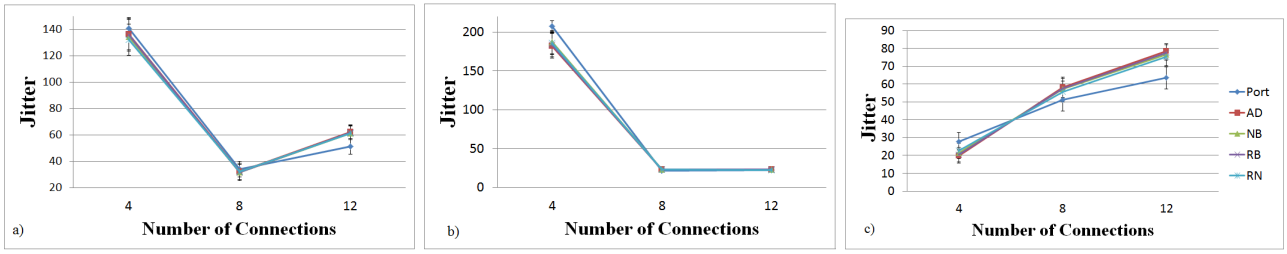

Fig. 7. Jitter versus variation of the connections number versus the variation of connection number with Port, AD (Decision Tree), RB (Bayesian Nets), NB (Naive Bayes) and RN (Neural Networks) classifiers.

niques based on machine learning algorithms. The evaluation considers three types of traffic and four machine learning techniques with an All-IP network varying the number of connections and providing a sensitive analysis of the results. A difference of performance is found in the results, between the current classifier of the DiffServ and the proposed machine learning classifiers.

Based on the results, one may argue that the machine learning classifier increases the network performance by increasing the performance of the voice applications, providing more quality to the call and more reliable transmissions, because it leads to a higher packet delivery ratio and a lower delay.

In the near future the authors plan to investigate an optical infrastructure, the evaluation of a mobile network with DiffServ and the proposed traffic classifiers, and the verification of the traffic classification in the dark for DiffServ networks.

REFERENCES

[1] S. Richards, *FutureNet: The Past, Present, and Future of the Internet as Told by Its Creators and Visionaries*, Wiley, Ed. Wiley, 2002.

[2] J. Zittrain, *The Future of the Internet–And How to Stop It*, Caravan, Ed. Caravan, 2008.

[3] T. Tronco, *New Network Architectures: The path to the future Internet*, Springer, Ed. Springer, 2010.

[4] D. O. Awduche, "MPLS and traffic engineering in IP networks," *IEEE Communications Magazine*, vol. 37, pp. 42–47, 1999.

[5] C. A. K. e Djamel Sadok, "Service definition and negotiation in the chameleon architecture," in *IEEE International Symposium on Telecommunications (IST 2001)*, 2001.

[6] A. Meddeb, "Internet QoS: pieces of the puzzle," *Comm. Mag.*, vol. 48, pp. 86–94, January 2010.

[7] K. Kim, "A distributed channel assignment control for QoS support in mobile ad hoc networks," *J. Parallel Distrib. Comput.*, vol. 71, pp. 335–342, March 2011.

[8] M. Natkaniec, K. Kosek-Szott, S. Szott, J. Gozdecki, A. Glowacz, and S. Sargento, "Supporting QoS in integrated ad-hoc networks," *Wirel. Pers. Commun.*, vol. 56, pp. 183–206, January 2011.

[9] H. Kim, D. Barman, M. Faloutsos, M. Fomenkov, and K. Lee, "Internet traffic classification demystified: The myths, caveats and best practices," in *Proc. ACM CoNEXT*, 2008.

[10] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A survey on Internet traffic identification," *IEEE Communications Surveys Tutorials*, vol. 11, no. 3, pp. 37 –52, 2009.

[11] T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Communications Surveys Tutorials*, vol. 10, no. 4, pp. 56 –76, 2008.

[12] M. Barros, R. de Morais Gomes, M. Alencar, P. Júnior, and A. Costa, "Avaliação de classificação de tráfego ip baseado em aprendizagem de máquina restrita à arquitetura de serviços diferenciados," *Revista de Tecnologia da Informacao e Comunicacao (RTIC)*, vol. 1, pp. 10–20, 2012.

[13] IETF, "IETF DiffServ Task Force," http://datatracker.ietf.org/wg/diffserv/charter/, 2011, accessed in November 2011.

[14] F. Kelly, "Mathematical modelling of the Internet," in *Fourth International Congress on Industrial and Applied Mathematics*, 1999.

[15] E. Kuumola, "Analytical model of AF PHB node in diffserv network," Networking Laboratory, Helsinki University of Technology,, Tech. Rep., 2001.

[16] S. Li and Y. Luo, "High performance flow feature extraction with multicore processors," in *IEEE Fifth International Conference on Networking, Architecture and Storage*, july 2010, pp. 193 –201.

[17] P. R. L. Jnior, "Roteamento adaptativo com agregao de trfego em redes pticas dinmicas," Master's thesis, UFCG, 2008.

[18] CAIDA, http://www.caida.org/home/., Accessed in August 2011.

[19] WIDE, http://mawi.wide.ad.jp/mawi/., Accessed in August 2011.

[20] M. Baldi and Y. Ofek, "End-to-end delay analysis of videoconferencing over packet switched networks," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 479–492, 1998.

[21] T. Viipuri, A Case Study of Simulating DiffServ in NS-2. http://www.netlab.tkk.fi/opetus/s383180/2005/harj/H3/diffnet.pdf, Accessed in 2012.

[22] ——, DiffServ scritps for NS2. http://www.netlab.tkk.fi/opetus/s383180/2005/harj/H3/ex3_scripts.tar.gz, Accessed in 2012.