International Journal of Computer Applications (0975 – 8887)
Volume 186 – No.73, March 2025

Performance Evaluation of Diffserv in various Scenario with Machine Learning Classification Effect

Neji Kouka
MARS laboratory, ISITCom hammam Sousse, University of Sousse, Tunisia

Jawaher Ben Khalfa
MARS laboratory, ISITCom hammam Sousse, University of Sousse, Tunisia

Jalel Eddine Hajlaoui
MARS laboratory, ISITCom hammam Sousse University of Sousse, Tunisia

ABSTRACT

Diffserv was introduced by the IETF as a standard model to offer QoS across core networks. Diffserv supports a QoS feature based on differentiation traffic. So far, little interest has been granted on machine learning feature in QoS . In this paper, we evaluate the effectiveness of these model in QoS in various scenarios. We show that under very heavy network load, Diffserv with machine learning classification (MLC) have a limited effect on the QoS parameter.

Keywords

Diffserv, QoS , Machine learning, NS2

1. INTRODUCTION

In the early days of the Internet and internet of things (IoT), the main concern was to be able to route packets from a source to a destination, regardless of the transit time. The traffic that circulated on the networks was not yet as diverse as to encounter the problems that we perceive today: network congestion, slow downloads loss of information in the network overload, etc. We are therefore increasingly demanding in terms of quality of reception of information: quality of service intervenes accordingly, to obtain user satisfaction: Quality of Experience (QoE). Several quality of service architectures have been implemented for quality of experience. We distinguish between data link level architectures and network level architectures. RFC 2990 highlights the most insightful problems in terms of QoS architectures in the Internet. A fundamental question raised in RFC 2990 is "what is the precise nature of the problem that QoS is trying to solve?".

From an application perspective, QoS is intended to control the network so that it generates a response to a specific service user in such a way that the response is consistent, predictable, and provides a minimum level of service guarantee. From a network perspective, QoS is intended to control network resources such that a service user obtains a level of service that is fair and superior to the service level of other users based on that user's request. According to RFC 2990, the network should actively participate in the tasks of resource allocation and service provision rather than passively requiring end devices to adapt to the resources available in the network. This raises a number of important QoS architectural questions including whether QoS should be offered at the application, transport, network or data link level or some/all of these levels and how, where and when to perform QoS routing and signaling ? This paper demonstrated that in realistic scenarios where the network is reasonably loaded, it is difficult to apply QoS based MLC.

.

2. RELATED WORKS

QoS refers to an assurance by the Internet to provide a set of measureable services attributes to the end-to-end users in terms of delay, jitter, available bandwidth and packet loss. QoS support in the Internet can generally be obtained by means of over-provisioning of resources and/or traffic engineering. IntServ model and DiffServ model [1, 2] are the typical QoS models employed in the Internet, which employs reservation-based and reservation-less approach, respectively. Machine learning adopted in traffic classification has giving a lot of attention today, In [3], which is chosen as a benchmark papers, authors proposed an automated Class of Service (CoS) mapping. 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 [4] [5] [6] [7]. 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 . This number has been updated to the range 0.9 to 0.95, with the usage of traffic classifiers based in machine learning [8] [9] [10] [11] . 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. In [12] authors propose a dynamic classification procedure, referred to as Learning-powered DiffServ (L-DiffServ), able to detect the distinctive characteristics of traffic and to dynamically assign service classes to IP packets.

3. QUALITY OF SERVICE

QoS refers to the overall performance of a network service, particularly in terms of its ability to prioritize certain types of traffic to ensure reliable and efficient data transmission. QoS is commonly used in networking to manage bandwidth, reduce latency, minimize packet loss, and improve the user experience for critical applications.

The Key Aspects of QoS are :

1. **Traffic Prioritization**: Assigning higher priority to critical data (e.g., VoIP calls, video conferencing) over less time-sensitive traffic (e.g., file downloads).
2. **Bandwidth Management**: Allocating network resources to ensure sufficient bandwidth for high-priority applications.
3. **Latency Control**: Reducing delays in data transmission, especially for real-time applications like gaming or streaming.

4. **Packet Loss Prevention**: Minimizing the loss of data packets during transmission to maintain data integrity.
5. **Jitter Reduction**: Stabilizing the variability in packet arrival times, which is crucial for real-time communication.

3.1 The Intserv/RSVP model

The RSVP model refers to the Resource Reservation Protocol, a network protocol used to reserve resources across a network for quality of service (QoS) purposes. RSVP is primarily used in IP networks to ensure that bandwidth and other resources are allocated for specific data flows, particularly for real-time applications like video conferencing, VoIP, and streaming..

3.2 The Diffserv Model

The DiffServ is based on packet marking with which a priority value is assigned to them in the queuing process. Service differentiation consists in a congestion situation of transferring packet loss to certain classes of traffic, to protect others. There is therefore no guarantee on flows because there is no dynamic intake control to avoid congestion. Admission control is done a priori by defining a contract for each traffic class and by sizing the resources to be able to guarantee this contract. DiffServ packets are marked at the entrance to the network, and routers decide based on this label which queue the packets are going to be placed in. This architecture is suitable for networks where it is not reasonable to consider flow-by-flow signaling. It therefore only considers flow aggregates for which signaling with resource reservation can be considered. In fact, a core router does not maintain a state for a given stream or aggregate, but treats all packets of a given class in the same way. The data is identified by marking in the ToS (Type of Service) field, which sets the priorities. Service differentiation has the following advantages: Signaling is done in each packet by assigning a different meaning to the bits in the service type field. There is no longer a need to keep a context in the router linking the signaling flow to the data flow. This also allows for a natural

aggregation of flows, so for an operator, packets that are marked for a certain class can belong to multiple sources. The complexity of processing is concentrated in routers at the edges of the network. They carry out the "complex" operations of checking the validity of the contract for the different classes of services.

4. DEFINITIONS AND TERMINOLOGIES

4.1 SLA (Service Level Agreement)

An SLA is a contract between the network operator and the customer, the purpose of which is to specify the permissible limits on the expected levels of service quality. This contract is negotiated statically or dynamically (on demand). An SLA is the formalisation of an agreement negotiated between two parties. It sets out in writing the expectation of the parties in terms of services, priorities, responsibilities, guarantees and, ultimately what we might define as the 'service level'. For example, it can be used to specify levels of availability, service, performance, operation or any other attribute of the service in question, such as billing, or even penalties in the event of failure to meet the SLA. An SLA is generally commercial and does not deal with technical aspects. The technical specifications of an SLA are generally described either by means of an SLS (Service Level Specification) and an SLO (Service Level Objectives). The SLS .is a technical specifying certain criteria that the service must meet. The SLO, on the other hand, provides the necessary to measure the quality of service as described in the SLS. The SLS contains parameters such as transmission capacity, burst size and peak data rate. As implemented by ISPs to date, SLAs for residential customers are usually referred to as 'terms of service'. In general, there is no traffic prioritisation (CoS) and little assurance of quality of service. In particular, there are no guarantees on latency, jitter or packet loss. the SLA serves more to limit the ISP's liability than to protect residential users.

4.2 Contractual QoS vs. on-demand

With contractual QoS, a fixed network resource (mainly part of the bandwidth) is assigned to each user. In this approach, users pay for the level of service even if they do not actually use all the allocated resources. In the on-demand QoS approach, instead of negotiating a fixed resource for each user, resources are allocated to each user. resources are allocated according to the user's actual needs over a specific period, With this approach, users only pay for the resources they actually use. It should also be noted that on-demand QoS can also be delivered by adjusting the speed of the client modem (CPE), known as 'bandwidth on demand'.

4.3 Soft QoS vs hard QoS

In the 'soft' QoS model, there is no QoS assurance during abnormal network conditions, such as large-scale attacks or failures of multiple network elements. Generally, a penalty system applies when the SLA is not met during normal operating conditions and the service is free of charge during abnormal conditions. The soft QoS model is not appropriate for applications requiring predictable quality of service. However, it can be useful for to sell QoS on demand. In addition, soft QoS does not offer any significant compared to the traditional Best Effort service. ISPs often use this model to to protect themselves rather than to provide QoS to customers. For Therefore, it may not be attractive enough for users to pay extra for quality of service. additional charges for a quality of service that they may or may not get. On the other hand, in the hard QoS model, ISPs try to provide a predictable QoS. It is more appropriate to sell contractual QoS.

Compared to the soft QoS model, it becomes easier to convince users to pay for QoS. However, during abnormal network operating conditions, it is difficult for an ISP to maintain the service at a well defined level. Consequently, network reliability becomes very important in the context of hard QoS. In addition, hard QoS can be commercially risky for ISPs because usually the penalties are too high.

4.4 Explicit QoS vs Implicit QoS

Explicit QoS means that the customer explicitly requests a specific level of service. Implicit QoS, on the other hand, means that the customer does not specifically request a specific level of QoS. In general, quality of service is built into the latter and there is no charge associated with this quality of service.

4.5 QoS planning

In a nutshell, network planning is the process where various network elements such as topology , capacity, traffic matrices, routing methods, and control methods are considered together in order to optimise network performance according to objectives, criteria and constraints. QoS is generally approached as a constraints, although it can also be an objective (minimizing delay, for example). According to the IBM Information Centre, 'planning is the most important step in achieving quality of service'. ISPs can also consider outages when planning and/or deploying their networks. This would make it possible to analyse the effect of a network element on the perception of the service by a given user and on the overall performance of the network. ISPs should also predict the most catastrophic events. Network auditing is also important because it can uncover weak points in the network. In fact, the QoS obtained is generally that corresponding to the worst-performing part of the network. Network administrators also need to check for configuration errors, strange behaviour and security vulnerabilities.

4.6 Reliability

Reliability can be defined as the availability of end-to-end functionality for customers and the ability of the network to survive failures or attacks, without affecting service or operation of the network. Network reliability includes survivability, fault tolerance and maintenance. While reliability is not part of our study, it is very important and plays a major role in the perception of the quality of the network. In fact, reliability and QoS can be addressed separately but this can lead to a complexity/command since complexity makes the network more vulnerable and triggers the need for additional control mechanisms which, in turn, increase the complexity of the network and so on.

4.7 Traffic Engineering, Management and Control

According to RFC 3272, Internet traffic engineering deals with the performance evaluation and optimisation of operational Internet networks. In addition, according to RFC 2702, Internet traffic engineering covers the application of scientific principles for the evaluation, characterisation, modelling, and control of Internet traffic. Generally, traffic engineering (TE) is applied during the network planning phase, Traffic Management (TM) is applied during the network operation phase, while traffic control (TC) specifies the means and actions that must be implemented in order to enforce the decisions taken during the engineering and traffic management phases. TM and TC mechanisms can be used by service providers to DiffServ's classes of service (the PHBs) as well as by hardware manufacturers in their in routing (routers) and switching (switches) devices. This includes packet classification, scheduling, marking, shaping, monitoring and forwarding. In addition, TM and TC give ISPs the opportunity to offer differentiated services and charge users accordingly. For example, VoIP, videoconferencing, online distributed gaming are delay-sensitive but can tolerate minor losses. As a result control mechanisms should be designed so that this type of traffic is dispatched immediately. On the other hand, email, web browsing, and file transfer can tolerate reasonable delays but can be severely affected by packet loss. Thus, traffic control must be designed such that traffic is not lost and is prioritised for storage in a congested situation. In a congested situation. TM and TC can also be used as a means of eliminating unwanted traffic such as spam and traffic created by worms and botnets. Some ISPs may even consider VoIP traffic that competes with local service offerings as undesirable. The unwanted traffic can be blocked or hindered entirely. Some ISPs may also wish to block Peer-to-Peer (P2P) traffic that consumes a large amount of bandwidth to the detriment of 'legitimate' customers. However, P2P traffic is generally 'agnostic' to QoS in the sense that it hides the identity of the protocols it is carrying; this prevents P2P traffic from being classified appropriately. Nevertheless, current trends indicate a continued growth in P2P traffic as it is becoming one of the main motivations for the proliferation of Internet services, which could dissuade ISPs from systematically blocking it.

5. SIMULATIONS

The network to be simulated consists of six nodes (FIG 1).

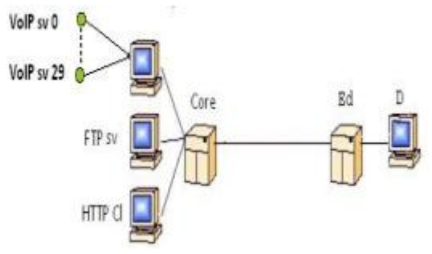

FIG 1: Simulation architectures 1.

5.1 Simulation without Machine learning

For this study, we simulated Diffserv with three types of traffic. It is important for the study to choose different types of traffic whose transport is not treated with the same priority. In our case, we chose Voice over IP (VoIP), FTP and HTTP traffic.

The PHB for VoIP traffic is Expedited Forwarding (EF), that of FTP traffic is Assured Forwarding (AF1x) while the class of service for HTTP traffic is Class Selector (CS1). It should be noted that the same scenario is used for simulations with and without Machine learning classification. A VoIP server generates voice traffic to client D at a bit rate of 64 Kbit/s, an FTP server implemented in the FTP sv node generates FTP-type traffic to client D with a bit rate of 800 Kbit/s, and finally an http client named HTTP Cl receives http traffic with a bit rate of 1Mb/s from the web server implemented in node D. The VoIP application layer is created by an agent constant packet generator, i.e. a 160 byte packet generated every 20 ms, which is equivalent to a data rate of 64 Kbits/s. This agent is implemented in the VoIP sv node. At transport layer level, this agent is attached to a UDP connection. We use this protocol because VoIP traffic is in unconnected mode. The FTP application layer is created by an agent that generates packets at constant speed, i.e. packets of 200 bytes, generated every 2 ms. This agent is implemented in the FTP sv node on top of the TCP protocol. And finally, the HTTP application layer is created by a constant speed packet generator agent, which generates packets of 200 bytes, generated every 2 ms. This agent is implemented in node D, because web traffic is generally downstream.

a. Marking of flows

DiffServ differentiation is applied at the entrance to the network, in two stages. When a packet enters the network, it is 'marked' with an identifier (DSCP) calculated from its protocol, source and destination. based on its protocol, source and destination. The routers that perform the marking at the entrance to the network are called 'Edge Routers'. Inside the network, each time a packet is routed, the DiffServ mechanism consults the type of packet, and, depending on the probability and the queue parameters, decides whether to transfer or destroy the packet. The routers carrying out the routing within the network are called 'core routers'.An incoming packet is placed in the queue corresponding to its marker. The packet is then routed or dropped according to the queue parameters. In our simulation, we want to give priority to VoIP packets over other packets in order to guarantee a certain quality of service. packets in order to guarantee a certain quality in the use of this technology. Other packets such as FTP or HTTP can be retransmitted because they are implemented under a TCP layer. In NS2 simulator, the DiffServ algorithm is placed on the links, not on the routers.

b. The three types of queues

In NS2, DiffServ works on the principle of using virtual queues within physical queues. Within physical queues. Different priorities and characteristics are defined for these queues in order to treat them in a more or less privileged way. In order to treat them in a more or less privileged way. In particular, it is possible to vary the probability of dropping a packet. In our simulation, we decided to define three types of packet, since the aim is to give priority to VoIP flows over other flows. To do this, we have defined three types of PHBs: 1. Assured Forwarding AF4x for the highest priority packets (VoIP). 2. Assured Forwarding AF1x for FTP packets. 3. Class Selector CS1 for HTTP packets. In addition, we have three physical queues, one for packets corresponding to the type of VoIP traffic, a second for FTP traffic and a third for HTTP traffic.

c. Creating edge links

Edge' links are responsible for marking packets so that the "Core" links can sort them. So we configure an Edge link by defining the three types of flow and the marking policy we want to apply to them. For our simulation, we identified the type of traffic depending on the source, since each source has a different type of traffic. each source has a different type of traffic. In reality, we have to identify the protocol of each packet in addition to the source.

Using the DiffServ mechanism, it is also possible to identify the destination of the packet, but we can't use this functionality, which would make the easier and more readable. We have opted for a 'TSW2CM' policing model, which uses two parameters CIR (Committed Information Rate). and CBS (Comitted Burst Size). CIR is a bit rate expressed in bits/s, while the CBS is expressed in bytes.CIR is a bit rate expressed in bits/s, while the CBS is expressed in bytes.

```
 $qCEd configQ 0 0 20 60 0.01
 $qCEd configQ 0 1 20 60 0.01
 $qCEd configQ 0 0 15 40 0.02
$qCEd configQ 0 1 15 40 0.02
 $qCEd configQ 0 0 10 25 0.05
 $qCEd configQ 0 1 10 25 0.05
```

So a packet will be marked with minimum priority if it is wider than the 'TSW2CM' sot. This model is added to the traffic going from the source to the destination. Several other policing techniques can be used instead of TSW2CM. are : TSW3CM, Token Bucket, srTCM (Single Rate Three Color Marker), trTCM (Two Rate Three Color Marker). These techniques are parameterised as follows: TSW2CM Initial code point CIR. TSW3CM Initial code point CIR PIR,. TokenBucket Initial code point CIR CBS. srTCM Initial code point CIR CBS EBS. trTCM Initial code point CIR CBS PIR PBS. CIR and PIR data rates are expressed in bit/s; CBS, EBS and PBS buckets are expressed in bytes. Note that only one policing policy can be used at a time for a pair of source destination. Once the 'dsRED/Core' and 'dsRED/Edge' queues have been defined, we need to distinguish between flows and link them with their sources and destinations using a 'policing' table and a 'PHBTable'. In other words, for each source-destination pair, the values of the marking and policing parameters must be specified. of the marking and policing parameters used. The addPolicyEntry command establishes three policies for edge nodes: one between node S1 and destination D, another between S2 and the same destination and finally one between source S3 and node D. and node D. The [$s1 id] command returns an identity value for which will be used by the addPolicyEntry. The addPHBEntry command maps each code point to the appropriate queue.This command adds an entry to the PHB table; for example, code point 36 is mapped to physical queue 0 and CP 37 to logical queue, for NS2, the default point code is zero.

5.2. Simulations with machine learning

a. Proposed Machine Learning Model for Dynamic Traffic classification

To overcome the limitations of traditional DiffServ, 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[3]. In this work we adopt model that dynamically adjusts QoS settings based on real-time traffic conditions as in [3]. The dataset was used as a list of features values used in ns2 simulations.

1. Data Collection: The system collects real-time data on network parameters such as:

- Throughput (kbit/s)
- Jitter (ms)
- Packetloss (%)

2. Feature Extraction and Model Training: Historical network performance data is used to train a machine learning model based neural network. Key features for the model include:

- Number of active servers (traffic load)
- Type of traffic (VoIP, FTP, HTTP)
- CurrentQoS configuration

3. Real-Time Traffic Prediction and QoS Adjustment: Once trained, the model is deployed in real-time to predict traffic patterns. It adjusts DiffServ classification based in [4]. The machine learning classification algorithm ensures that VoIP traffic maintains high priority and optimal performance, while adapting the QoS parameters for lower-priority traffic like FTP and HTTP.

6 Results

6.1 Influence of ML Diffserv on throughput

We need to check whether, when the number of priority flows is increased, Machine learning classification have an effect on quality of service? To simulate this scenario, we keep the same server and gradually increase the number of VoIP servers, up to 30. These are all attached to the S1 source, using the same UDP protocol. This is illustrated in FIG1.

For each simulation (for a given number of servers), we measure the throughput of the VoIP source sv0 as perceived by the destination D. We want to study the impact of the increase in priority flows on the quality of Service with the case with ML and without ML. The server generates VoIP traffic at a rate of 64 kbit/s.

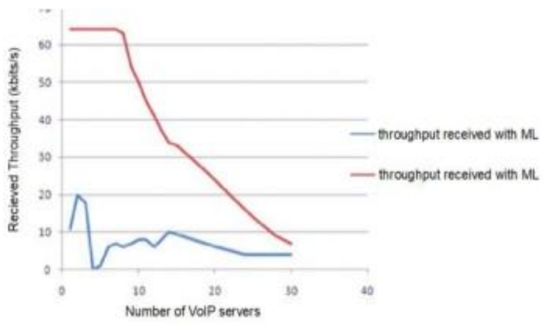

FIG 2: Throughput received vs number of VoIP server

the blue curve shows the VoIP server throughput sv0 perceived by destination D. This is taken as a function of the number of servers. In other words, each time we increase the number of servers by one, we measure the throughput for a given flow (all flows are identical). Note that the throughput drops from 11 kbit/s when using a single server to 4 kbit/s when 30 servers are used. It is clear that the quality of this stream is poor because the loss rate varies from 84% (one server) to 90% (30 servers). the red curve shows the throughput generated by the VoIP server sv0 and perceived by the destination D. It can be seen that MLC improves quality of service for up to 7 servers. In other words, although priority servers in addition to VoIP server sv0, the loss for the latter is zero, which shows that the MLC approach improve quality of service. However, as soon as we exceed 10 servers, the VoIP throughput sv0 perceived by destination D starts to decrease, reaching 7 kbit/s with 30 servers. It can also be seen that with 30 number of servers, the throughput is almost the same for the scenarios with and without DiffServ MLC. These results show that MLC improves quality of service if there isn't too much priority traffic As shown, the machine learning model was able to maintain higher throughput levels for VoIP traffic (with minimal packet loss) even as the number of servers increased beyond 10. , that the effect of MLC is no longer perceptible if all users want to be their traffic have priority.

6. 2 Influence of MLC Diffserv on jitter

Similarly, we measure the jitter of the VoIP stream for the VoIP server sv0 as a function of the number of VoIP servers. VoIP sv0 jitter as a function of the number of servers. Illustrates the results obtained. It can be seen that without MLC DiffServ the jitter is 4s for the case of a single VoIP server. After that decreases enormously to 0.3s with two servers. This decrease is mainly due to the increase in VoIP packets.

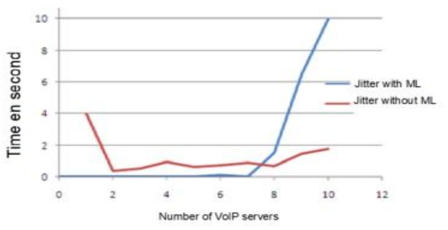

FIG 3: Gitter with and without MLC Diffserv

for a single VoIP server, the packet generation rate is lower than the FTP and http server rates. In other words, it is very likely that VoIP packets will be interspersed with an unknown and variable number of FTP packets. As VoIP flows increase the VoIP server jitter sv0 will become lower and this is mainly due to the evolution of the VoIP flow, i.e. it is very likely that VoIP packets a lower transfer delay. As soon as the number of servers reaches 10, this jitter is 1.8s. On the other hand, with MLC the jitter is almost zero for all cases with less than seven servers. then it increases rapidly to 1.5s when using 8 servers, then, as soon as another server is added, it increases sharply to reach 6.5s , then it increases exponentially for more than eight servers. So, we can see that with MLC, jitter is also improved when there is a little of priority traffic (here less than seven servers). With the machine learning-based approach, jitter was significantly reduced under high traffic conditions .

6.3 Influence of MLC Diffserv

On loss rate we keep the same simulation architecture and each time we measure the total loss in kbit/s of VoIP traffic for the 30 scenarios.

For each scenario, the number of VoIP servers is incremented by one. The results are shown in Figure 4 .

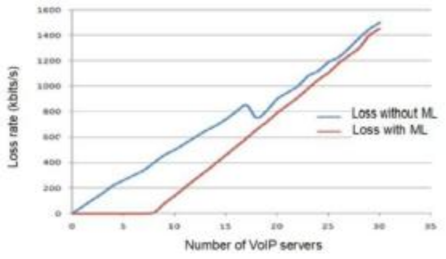

FIG 4: loss rate vs number of VoIP server

We can see that the loss curve without MLC shows three phases. In the first phase, until 17 servers are used, the loss rate increases by almost 60 kbit/s for each server added. Then, as soon as another server is added, the loss rate decreases by almost 50 kbit/s to 770 kbit/s. This decrease is mainly due to to the non-stability of the queue, in fact it may happen that there was congestion during the previous measurement. Then, in the third phase, the loss increases uniformly by almost 80 Kbits/s for each each server added until it reaches 1,500 kbit/s when 30 VoIP servers are used. VoIP servers, which have a total transmission rate of 1920 kbit/s, giving a loss rate of almost 78%. On the other hand, when MLC is implemented, the loss is zero for up to 7 servers. It is 10 kbit/s for 8 VoIP servers. Thereafter, it increases almost linearly from almost 100 kbit/s until it reaches 1460 kbit/s for 30 servers, which is almost equal to the loss rate obtained without Machine learning classifier.

7. CONCLUSION AND FUTURE WORKS

Our goal was to evaluate the effect of MLC Diffsev on QoS depending on the traffic. By integrating this method into our simulatin, we highlight a novel approach to overcoming the limitations of traditional Diffserv under heavy network loads. We could obervse that In low loaded networks MLC Diffserv has an impact on QoS but in heavy loaded conditions this technique is still scarce.

Future research could use a deep reinforcement learning, to further enhance QoS management in increasingly complex network environments. Additionally, we will integrating this approach with other QoS architectures to offer a heterogeneous QoS .

8. REFERENCES

[1] Wroclawski J (1997) The Use of RSVP with IETF Integrated Services.RFC 2210 pp: 1-33

[2] Blake S, Black D, Carlson M, Davies M, Wang Z, et al. (1998) An Architecture for Differentiated Services. RFC 2475 pp: 1-36.

[3] J. Wang, “Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication),” IEEE J. Quantum Electron., submitted for publication.

[4] 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.

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

[6]] 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.

[7] M. Barros, R. de Morais Gomes, M. Alencar, P. J´ unior, and A. Costa, “Avaliac ¸˜ ao de classificac ¸˜ao de tr´afego ip baseado em aprendizagem de m´ aquina restrita `a arquitetura de servic¸os diferenciados,” Revista de Tecnologia da Informacao e Comunicacao (RTIC), vol. 1, pp. 10–20, 2012.

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

[9] S. Li and Y. Luo, “High performance flow feature extraction with multi core processors,” in IEEE Fifth International Conference on Networking, Architecture and Storage, july 2010, pp. 193–201

[10] 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.

[11] 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.

[12] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, and A.Rezaee, “Load BalancingMechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of theLiterature,” IEEE Access, vol. 6, no. c, pp. 14159–14178, 2018