Adaptive Control of Traffic Congestion with Neuro-Fuzzy based Weighted Random Early Detection

Irina Topalova
Department of Information Technology
University of Telecommunications and Post
Technical University Sofia
Sofia, Bulgaria
email:itopalova@abv.bg

Pavlinka Radoyska
Department of Information Technology
University of Telecommunications and Post
Sofia, Bulgaria
email:pradoiska@abv.bg

***Abstract*— Differentiating class-based traffic and class-based queue management is the most advanced approach for queue management in routers and switches, controlling and preventing the congestion. The combination of a mechanism for prioritizing the Internet Protocol traffic and the way to dynamically modify the parameters of the packet rejection algorithm is essential for achieving efficient and reliable traffic. In this study, a method is proposed, exploring the automatic adaptation of new users added to the backbone of the network, to the already defined weighted random early detection parameters. A neuro-fuzzy-logic network is trained to automatically adapt new end users to the quality of service policy already set in the backbone area. This network is trained with the quality of service parameters of the backbone area and serves to adapt these parameters in the newly-added routers. The results obtained are compared with those from the study of this problem by the authors, when a multilayer neural network is used.**

Keywords-traffic congestion; Quality of Service; Weighted Random Early Detection; fuzzy logic; neuro-fuzzy system.

I. Introduction

In modern Information Technology (IT) communications, the Quality of Services (QoS) is essential for traffic efficiency. The creation of queues and their inadequate management results in substantial packet delays. The QoS aims to guarantee the quality of message delivering by congestion management and congestion avoidance. Various methods are currently applied to reduce the negative effect of the problem. But more and more experimental methods of artificial intelligence are being explored, hoping for better results as given by K. Markov et al. [1], B. Deaire et al. [2].

In this study, a method is proposed, to investigate the automatic adaptation of new users added to the backbone of the network, to previously defined weighted random early detection parameters. The neuro-fuzzy logic network is trained to automatically adapt new end-users to the service quality policy already in place. This network is trained with the service quality parameters of the main zone and serves to adapt these parameters to the newly added routers. The results obtained are compared with those, from the study of this problem by the authors, when a multilayer neural network is used, as well as with results from other similar researches. The novelty of the proposed method consists in the possibility of automated adaptation of the newly added QoS parameters to an already existing communication structure without having to reconfigure these network devices. This is made possible by the proposed adaptive neuro-fuzzy system, which approximates the parameters of these devices in order to bring them closer to the one already set.

The rest of this paper is organized as follows. Section II describes the related to the research works. Section III describes methods for congestion avoidance and Weighted Random Early Detection methods. In Section IV, the proposed method for weighted random early detection parameter adjustment is presented. Section V gives the experimental results. Section VI closes the article.

II. Related Work

In the recent years, various methods have been proposed to implement fuzzy logic to optimize traffic or to create predictive models. E. Jamhoura et al. [3] propose a method for building a fuzzy predictor to model a differentiated services (DiffServ) node with two queues - for Voice over IP (VoIP) traffic and self-similar data traffic. They define the fuzzy membership functions on the base of extending the existing queue models and apply a fuzzy model to build network traffic controllers. M. Yaghmaee et al. [4] proposed a fuzzy based controller for traffic differentiated services. Their fuzzy scheduler is based on the waited fair queue mechanism, in which the significance of each queue is adjusted by the fuzzy controller. To dynamically tune the committed interface rate, the authors use a two input one output fuzzy controller. The presented results show better performance than non-fuzzy mechanisms. The researchers S. Shalinie et al. [5] describe the input and output of a queue size regulation system, by a fuzzy set. In the proposed model, they use two inputs - Traffic Intensity and Available Link Bandwidth. Output of this model is the Queue size parameter. But fuzzy logic-based Adaptive Drop Tail shows significant improvement in controlling congestion without any need for special parameterization or tuning as given by A. Mishra [7]. The outcome shows that their proposed Adaptive Drop Tail Fuzzy Logic controller has reduced packet loss when compared to traditional Drop Tail mechanisms. The simulation is designed to maintain adaptive buffer space when a sudden change in overloading occurs, which prevents Internet router buffers from becoming full when overloading occurs.

The above-mentioned methods that use fuzzy logic offer ways to reduce congestion through non-traditional queue management in network device buffers, rather than addressing the problem of adapting the new device's QoS parameters to those already set in the primary communications area.

In our study, we use fuzzy logic combined with neural network training to offer a simplified method of adjusting the QoS parameters of newly-added routers to the backbone area. The difference in the approach of this method is that the need to create an analytical model of traffic queues is eliminated. At the same time, the use of fuzzy logic in conjunction with a Nero Fuzzy System (NFS) allows the uncertainty of the average queue, to be transformed into a specific value of the Mark Denominator parameter. This parameter, obtained as a NFS solution, is fed to the newly-added routers to match this QoS parameter to those already set in the backbone area.

III. Congestion Avoidance and Weighted Random Early Detection

Network congestion occurs in two cases: when data arrive on a big pipe and get sent out a smaller pipe and when multiple input streams arrive at a router whose output capacity is less than the sum of the inputs. Congestion avoidance in network communications has two significant components: congestion management in end devices based on Transmission Control Protocol (TCP) algorithm and Active Queue Management in routers.

A. TCP congestion avoidance

The main purpose of congestion management in end devices is to adapt TCP window size to the bottleneck throughput while maintaining an optimal exchange rate. The basic congestion control algorithms, focused on end devices are implemented in TCP protocol (RFC 5681) and include: slow start, congestion avoidance, fast retransmit (TCP Tahoe) and fast recovery (TCP Reno). Different variants are compared by authors H. Kaur and G. Singh [6], A. Mishra [7], N. Parvez et al. [8].

The TCP window size is measured in bytes. The communication starts with the slow start phase (RFC5681). The sender doubles the widow size on every received TCP acknowledgement (ACK). Slow start stops when the window size reaches slow start threshold (*ssthresh*) or at the first missing ACK. The congestion avoidance phase starts after the widow size reaches *ssthresh*. The window size increments by one full size segment on any Round Trip Time (RTT). The congestion avoidance phase stops at the first missing ACK. According to the traditional TCP algorithm, the slow start phase is activated after any missing ACK. Congestion avoidance defines how to deal with lost packets. There are two indications of packet loss: (1) waiting time has expired and (2) receipt of duplicate ACKs. Retransmit Time Out (RFC 6298) is a parameter which determines the wait time for acknowledgment. The Receiver returns to the sender an ACK after every arrived segment. But it acknowledges the latest ordered segment data. The segments that arrive out of order (there is a missing segment) are buffered, but not acknowledged. This mechanism follows more than one ACK for a segment. According to the Selective Acknowledge TCP algorithm (SACK), the receiver acknowledges the last ordered segment and all buffered segments. SACKs with the same last ordered segment acknowledged, independently of acknowledged buffered segments, are considered as duplicate acknowledgements.

B. Congestion avoidance mechanisms in the routers

Random Early Detection (RED) was proposed by C. Ghazel and C. Saidane [9] in the early 1990s to address network congestion in a responsive rather than reactive manner. It aims to trigger TCP congestion avoidance in end devices before traffic congestion has occurred. As a result, the data transmission speed is reduced and congestion in the router is avoided. RED controls the average queue size in the router and compares it with the predefined threshold for the minimum (*minq*) and maximum queue (*maxq*) size. RED runs in *minq* – *maxq* range – shown in Figure 1. At an average queue size less than the *minq*, the packets are sent in pure FIFO mode. At an average queue size greater than the *maxq*, all packets are dropped. RED decides which packages to drop using probability calculations based on the minimum, maximum and average queue size, the ratio of the current packet size to the maximum one and the number of packets in the queue as is given by S. Rajput [10]. MPD (Mark Probability Denominator) is used to limit the dropped packets, according to the average queue size during the RED phase. MPD defines the number of dropped packets when average queue size is equal to *maxq*, just before full drop phase.

Some authors give a review in IEEE Transactions [11] and also X. Jiang et al. [12] consider the main problems of the RED algorithm as: 1) unpredictable queuing delay and 2) a sharp decrease in the throughput with high traffic load. Unpredictable queuing delay provokes to instability of RTT. RTT may become larger than the Recovery Time Objective (RTO) and causes retransmission of packets already received, and hence overload the network. Other authors, Cisco IOS Quality of Service Solutions [13], consider this behavior to be reasonable.

The traffic flow describes the communication between two sockets. The packets marked for dropping are selected based on the probability theory rather than on full statistics. This can cause more frequent dropping of packets from some flows than packets of other flows. Thus, the mechanism of congestion avoidance in some flows is triggered more often than others, and the speed of communication between two sockets can be drastically slow while others remain high. Furthermore, some packets carry no TCP traffic and are therefore not sensitive to the TCP congestion avoidance mechanism. These packets will not reduce their transmission rate and it is very likely, that the queue will be filled with their packets only. As a result, the router will become “impassable” for TCP communication. On the third hand, in the presence of DoS (Denial of Service) attacks, the attackers turn off the slow start and congestion avoidance mechanisms and send their packages with maximum windows size. Of course, there are serious defenses against such attacks, but with low-rate DoS attacks the most of them do not work as shown by M. Al-shaw and A. Laurent [14].

The WRED algorithm given by A. Custura et al. [15] has been developed to achieve better fairness and is implemented into the operating systems of two of the leading companies in the communications industry – Cisco and Juniper. The packages are split into flows. Flows are merged into queues. Each queue gets a specific portion of the outgoing bandwidth. Within each queue, streams get their weighting priority. Priorities are defined as: high, medium, and low. The priority determines the probability of the packets dropping. Each package is marked to determine which queue it belongs to and what its priority is. The DifServ fields in the IPv4 (RFC) and Type of Service (ToS) header in the IPv6 header (RFC) are used to classify traffic. These fields are 8 bits long. The first 6 of them are used for Differentiated Service Code Point (DSCP) (RFC2474), (RFC2475) and the last 2 bits are for experimental use. RFC5865 describes service classes, according to the traffic types. When constructing the DSCP classes, it is recommended to apply Per-Hop Behaviors (PHBs) and Active Queue Management (AQM) mechanisms. Service class applies to applications with similar characteristics and performance requirements, such as specific delay, loss and jitter. DSCPs to Service Class Mapping is shown in Table 1. The network administrator may choose to implement different service classes, or to implement different behaviors for service classes, or to aggregate different kinds of traffic into one class. Only the Default Forwarding (DF) "Standard" service class is required. All other service classes are optional. Three types of queues can be defined: priority queue, rate queue and AQM. Each defined queue gets the portion of outbound bandwidth. Cisco developments recommended by Geib, R. and D. Black [16] a bandwidth distribution by types of traffic.

Only AQM queues are based on packet dropping and RED/WRED. AQM queues define only Assured Forwarding (AF) classes and DF. Any package that is not explicitly marked belongs to DF class. DSCP bits for AF classes are depicted in Figure 2. The first 3 bits define the class number, the next two - the priority, and the last one must be 0.

Collections of packets with the same DSCP setting that are sent in a particular direction can be grouped into a Behavior Aggregate (BA). Packets from multiple sources or applications can belong to the same BA. DSCP is used to select the Per-Hop Behavior (PHB) at each interface.

PHB (RFC2475) is a mechanism that allows independent management of DSCP classes in each router. DSCP classes are queueing in the router in a locally defined manner. A portion of the output bandwidth is allocated to each queue. For example, class traffic with DSCP Af11, Af12 and Af13 is incorporated in one AQM queue with name gold (statement 1). For this queue, 35% of outbound bandwidth (statement 2) is allocated. *Minq, maxq*, and *MPD* are defined for DSCP classes in the queue. Let configure *minq* =20, *maxq* =40, *MPD* =10 for DCSP class *af11* (statement3).

```
class-map match-all gold match ip dscp af11 af12 af13
class gold bandwidth percent 35
random-detect dscp af11 20 40 10
```

Based on these analyzes and research, as well as our observations, we have come to the conclusion that the most efficient action, would be the implementation of a WRED mechanism, but with the ability to transform the queue uncertainty into specific values of the rejected packets. This leads to the idea of using fuzzy logic.

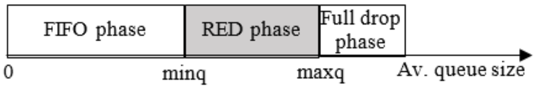

Figure 1. Queue management phases.

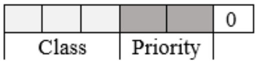

Figure 2. DSCP bits for AF classes.

TABLE I. DSCP TO SERVICE CLASS MAPPING

Service Class Name	DSCP Name	DSCP Value	Application Examples
Network Control	CS6	110000	Network routing
Telephony	EF	101110	IP Telephony bearer
Signaling	CS5	101000	IP Telephony signaling
Multimedia Conferencing	AF41, AF42, AF43	100010, 100100, 100110	H.323/V2 video conferencing (adaptive
Real-Time Interactive	CS4	100000	Video conferencing and Interactive gaming
Multimedia Streaming	AF31, AF32, AF33	011010, 011100, 011110	Streaming video and audio on demand
Broadcast Video	CS3	011000	Broadcast TV & live events
Low-Latency, Data	AF21, AF22, AF23	010010, 010100, 010110	Client/server transactions Web-based ordering
OAM	CS2	010000	OAM&P
High-Throughput Data	AF11, AF12, AF13	001010, 001100, 001110	Store and forward applications
Standard	DF (CS0)	000000	Undifferentiated applications

IV. PROPOSED METHOD FOR QOS PARAMETER ADJUSTMENT

The proposed method is based on the functional scheme shown in Figure 3. The assumption is that in the end (Remote site) routers, as well as in the Central router, the AF classes with related traffic types are already defined. We assume that WRED and differentiated services are configured in the end routers. The DSCP values and the minimum and maximum threshold range, considered for managing the average queue depth in the central router, are both configured. The Neuro-Fuzzy Device Manager

(NFDM) is trained with these two parameters and prepares in its output the calculated Mark Denominator (MD). MD defines the fraction of packets dropped when the average queue depth is at the maximum threshold. Thus, the NFDM system consists of two inputs and one output variables. As newly-added routers are connected to the Central router area, their also configured DSCP values are submitted to the already trained NFDM. According to the defined linguistic rules in the inference phase, the NFDM sends the calculated MD to the added routers. This action seems to be reasonable, because the following baseline markings with DSCP Assured Forwarding PHB are typically recommended by Cisco Systems, represented by B. Hedlund [17]:

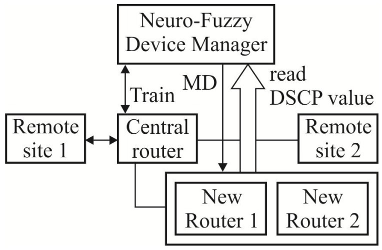

Figure 3. Functional scheme of the proposed method

- Interactive Video - AF41
- Mission Critical Data (locally defined) - AF31
- Transactional Data (dlsw, sql, sap): AF21
- Bulk Data (email, ftp, backups): AF11.

Thus, it is assumed that these classes are set by following these recommendations in the newly-added routers.

V. Experimental Results

The NFDM was trained with two input variables. The first one is represented in Figure 4 and defines the membership functions of the DSCP values - combination of traffic class and its priority. The upper angles of the trapezoidal membership functions are set to point to the exact DSCP values of the respective range of the standard AF classes. Seven ranges are defined - 10-12 and 12-14 respectively for class 1; 18-20 and 20-22 for Class 2; 26-28 and 28-30 for Class 3; 34-38 for Class 4. The overlapping areas of the trapezoidal shapes are so set, that the values of the membership functions are negligible, as no packets with DSCP values beyond the standard are expected. The second input variable is represented in Figure 5 and defines the four ranges are chosen as typically recommended [17]. membership function of the Min-Max threshold values. Here The defined MD as output result of the Inference and Defuzzifiction phase is shown in Figure 6. Because of its neural network structure, the system is capable of learning (an advantage of neural systems) and because of its fuzzy-like topology it is possible to recreate the processing steps. To design a system that takes advantage of neural networks and fuzzy systems in one project, we need a system that processes the fuzzy membership functions and fuzzy model rules.

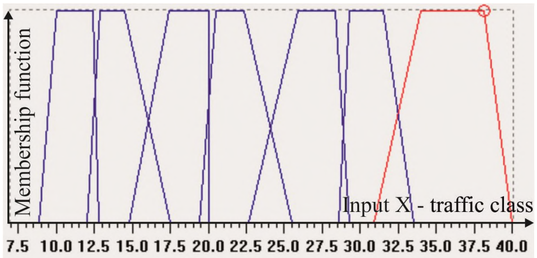

Figure 4. Membership function of DSCP values (combination of traffic class and its priority)

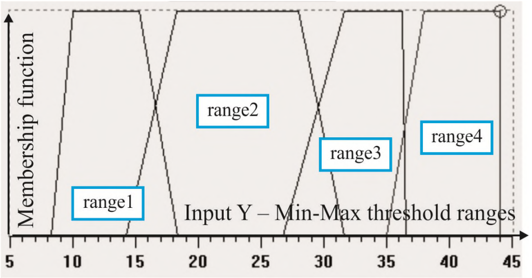

Figure 5. Membership function of the Min-Max threshold values

Thus, we can determine knowledge from the sampling data using neurons capable of learning. You can solve this problem by using a special neural network called NFN. It consists of three neural subnets (NSNs) that emulate the three sub-sequences - fuzzification, inference and defuzzification - of the fuzzy system [18].

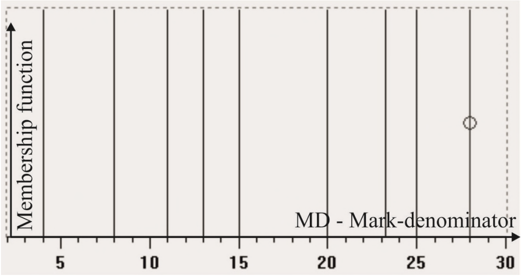

Figure 6. Membership function of MD

Thus, the fuzzification of the non-fuzzy input variables is implemented by a layer of neurons with activation functions of the both above described inputs. One neuron is assigned to each membership function of the input variables. In the second layer of the neuro-system, the rule base of the fuzzy system is applied and one neuron is assigned to each rule. The IF part of a fuzzy rules is implemented by the first neuron layer and the 2nd layer implements the THEN part of a fuzzy rule. The number of neurons in the second layer is equivalent to the number of membership functions of the output variables. The output values of the system in the 3rd neuron layer are implemented by the standard defuzzification method with MAX-PROD inference followed by centroid calculation as given in NeuroSystem, User Manual [18].

Table II demonstrates the chosen linguistic rules, which are inputs to the second layer neurons. For example, the first to third columns of Table II, set the following linguistic rules:

IF *DSCP is 10 to 12* and *Range 1* THEN MD=8;
IF *DSCP is 12 to 14* and *Range 1* THEN MD=4;
IF *DSCP is 18 to 20* and *Range 2* THEN MD=8; etc.

TABLE II. LINGUISTIC RULES OF THE NFDM

Input X: traffic class with drop preferenc	AF11-AF12 (DSCP values 10 to 12)	AF12-AF13 (DSCP values 12 to 14)	AF21-AF22 (DSCP values 18 to 20)	AF22-AF23 (DSCP values 20 to 22)	AF31-AF32 (DSCP values 26 to 28)	AF32-AF33 (DSCP values 28 to 30)	AF41-AF43 (DSCP values 34 to 38)
Input Y: Min-Max Threshold range	range 1	range 1	range 2	range 2	range 3	range 3	range 4
Output Z: MD	8	4	15	11	25	20	28

Figure 7 shows the obtained 3D surface, which illustrates the obtained dependencies between inputs X and Y and the resulting value of MD (axis Z) at the output of the NFDM. But the 3D presentation is not informative enough, when the number of input and output variables for NFDM is higher. In this case, it is better to represent the variables as it is shown in Figure 8 and Figure 9.

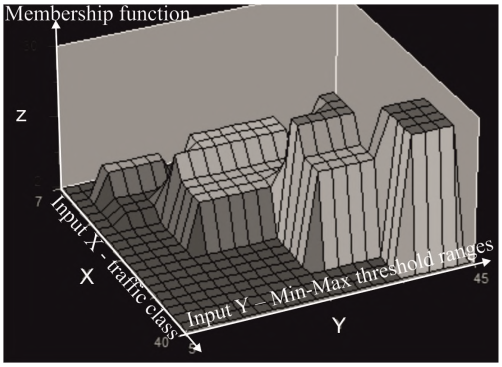

Figure 7. Membership function of MD (axis Z) according to Input X (DSCP) and Input Y (Min-Max threshold ranges)

Figure 8 shows the variations of MD (green line) according to AF12 (DSCP=12; yellow line) and all threshold ranges 1 to 4 (red line). Figure 9 shows the variations of MD (green line) according to AF32 (DSCP=28; yellow line) and all threshold ranges 1 to 4 (red line). Both figures show very well the change in the value of MD, that is submitted to the newly-added router, when its DSCP value is brought to the input of the already trained NFDM system.

VI. CONCLUSION AND FUTURE WORK

The use of fuzzy logic in conjunction with a neural network NFS in the proposed method, allows the uncertainty of the average queue to be transformed into a specific value of the Mark Denominator parameter. This parameter, obtained as a NFS solution, is fed to the newly-added routers to match this QoS parameter to those already set in the backbone area. An advantage of the method is that no any analytical model is designed, but only NFDM training is required.

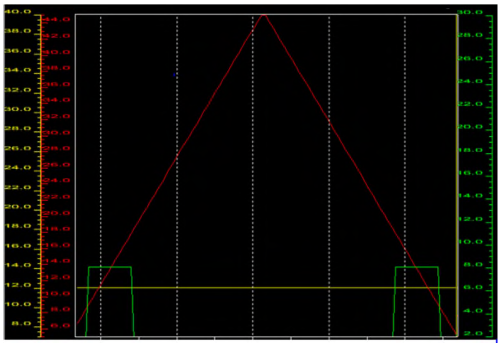

Figure 8. Variations of MD (green line) according to AF12 (DSCP=12; yellow line) and all threshold ranges 1 to 4 (red line)

The advantage is the ability of the network to learn, i.e. its ability to adapt to changed behavior and new situations. To exploit the benefits of both - the easy understandability of fuzzy systems and the ability to train neural networks - the two techniques are combined. Compared to the results of the authors' study given by I. Topalova and P. Radoyska [19], where only neural network is used to match MD, the NFDM method does not require a large volume of trained samples, as the initial uncertainty of the selected input variables. The methods discussed above use fuzzy logic and offer ways to reduce congestion through non-traditional queue management in network device buffers, rather than addressing the problem of adapting the new device's QoS parameters to those already set in the primary communications area, applying automated adaptation. In this sense, we consider the proposed method of no analogue in the scientific registers

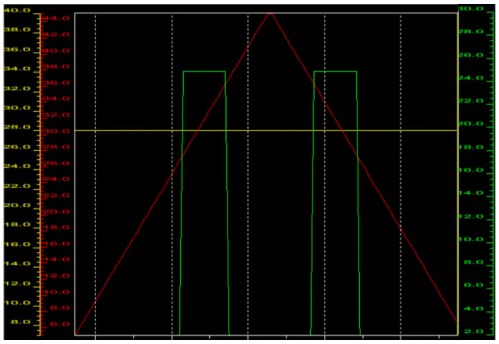

Figure 9. Variations of MD (green line) according to AF32 (DSCP=28; yellow line) and all threshold ranges 1 to 4 (red line)

As a further continuation of the study, we are testing the NFMM system with more MD values for tracking and validating the adjusted values for the newly-added routers. Different forms (for example triangular) of the Membership functions of the Min-Max threshold values and of the DSCP values will also be attempted, aiming to investigate the degree of uncertainty.

REFERENCES

[1] K. Markov, K. Ivanova, K. Vanhoof, I. Mitov, B. Depaire, V. Velychko, and V. Gladun, “Intelligent Data Processing Based on Multi-Dimensional Numbered Memory Structures,” Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems, IGI Global, 2013, pp. 156-184, doi: 10.4018/978-1-4666-1900-5.ch007, ISBN: 978 1-4666-1900-5, EISBN: 978-1-4666-1901 2.

[2] B. Deaire, K. Ivanova, K. Markov, I. Mitov, K. Vanhoof, and V. Velychko,“Multi-dimensional Information Spaces as Memory Structures for Intelligent Data Processing in GMES,” pp 347-370 In: Kr. Markov et al. Intelligent Data Processing in Global Monitoring for Environment and Security, ITHEA, 2011, Kiev, Ukraine - Sofia, Bulgaria. ISBN: 978-954-16-0045-0 (printed), ISBN: 978-954-16-0046-7 (CD/DVD), ISBN: 978-954-16-0047-4 (online). ITHEA® IBS ISC No.: 21. 410 p.

[3] E. Jamhoura, M. Pennaa, R. Nabhenb, and G. Pujolleb, ” Modeling a multi-queue network node with a fuzzy predictor,” Fuzzy Sets and Systems 160 (2009) 1902–1928, © 2008 Elsevier B.V., doi:10.1016/j.fss.2008.12.004.

[4] M. Yaghmaee, M. Menhaj, and H. Amintoosi, “Design and performance evaluation of a fuzzy based traffic controller for differentiated services,” Computer Networks 47 (2005) 847-869, available online at www.sciencedirect.com, access date april, 2019.

[5] S. Shalinie, G. Preetha, S. Nidhya, and B. Devi, “Fuzzy Adaptive Tuning of Router Buffers for Congestion Control,” International Journal of Advancements in Technology, http://ijict.org, Vol 1, No 1, © IJoAT ISSN 0976-4860, June 2010.

[6] H. Kaur and G. Singh, “TCP Congestion Control and Its Variants,” Advances in Computational Sciences and Technology ISSN 0973-6107 Volume 10, Number 6 (2017) pp. 1715-1723.

[7] A. Mishra, “Performance Analysis of TCP Tahoe, Reno and New Reno for Scalable IoT Network Clusters in QualNet” ® Network Simulator, International Jpurnal of Computer Sciences and Engineering 6(8), pp:347-355, August 2018 DOI: 10.26438/ijcse/v6i8.347355.

[8] N. Parvez, A. Mahanti, and C. Williamson, "An Analytic Throughput Model for TCP NewReno," in IEEE/ACM Transactions on Networking, vol. 18, no. 2, pp. 448-461, April 2010. doi: 10.1109/TNET.2009.2030889

[9] C. Ghazel and C. Saidane, “Next generation networks dimensioning for improving and guaranteeing quality of service,” The International Journal of Networks (JNW) 5 (7), pp.782-791, 2018.

[10] S. Rajput, V. Kumar, and S. Paul, "Comparative analysis of random early detection (RED) and virtual output queue (VOQ) algorithms in differentiated services network," 2014 International Conference on Signal Processing and Integrated Networks (SPIN), Noida, 2014, pp. 237-240. doi: 10.1109/SPIN.2014.6776954

[11] Learning-Automata-Like Solution, in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 40, no. 1, pp. 66-76, Feb. 2010. doi: 10.1109/TSMCB.2009.2032363

[12] X. Jiang, J. Yang, G. Jin, and W. Wei, RED-FT: A Scalable Random Early Detection Scheme with Flow Trust against DoS Attacks, IEEE COMMUNICATIONS LETTERS, Vol. 17, No. 5, pp. 1032-1035, May 2013.

[13] Cisco IOS Quality of Service Solutions Configuration Guide, [Online] Available: https://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/configuration/guide/fqos_c/qcfintro.html, access date march, 2019.

[14] M. Al-shaw and A. Laurent, QoS Design Principles and Best Practices, Cisco Press, Jan 1, 2018, [Online] Available:http://www.ciscopress.com/articles/printerfriendly/2756478, access date march, 2019.

[15] A. Custura, A. Venne, and G. Fairhurst, "Exploring DSCP modification pathologies in mobile edge networks," 2017 Network Traffic Measurement and Analysis Conference (TMA), Dublin, 2017, pp. 1-6.

doi: 10.23919/TMA.2017.8002923

[16] R. Geib and D. Black, "Diffserv-Interconnection Classes and Practice", RFC 8100, DOI 10.17487/RFC8100, March 2017, [RFC8100].

[17] B. Hedlund, “Enterprise QoS Solution Reference Network Design Guide,” Cisco Systems, 2017.

[18] NeuroSystem, User Manual, Copyright © Siemens AG, 2006.

[19] I. Topalova and P. Radoyska, “Control of Traffic Congestion with Weighted Random Early Detection and Neural Network Implementation”, ICAS 2018, The Fourteenth International Conference on Autonomic and Autonomous Systems, pp. 8-12, Nice, France, 20-24 May 2018.