

A Hysteretic Model of Queuing System with Fuzzy Logic Active Queue Management

Vladimir Deart, Andrey Maslennikov
 Moscow Technical University of Communications and Informatics
 Moscow, Russia
 vdeart@mail.ru, amasl2048@gmail.com

Yuliya Gaidamaka
 Peoples Friendship University of Russia
 Moscow, Russia
 ygaidamaka@mail.ru

Abstract—We consider a data transmitting system with an active queue management designed to prevent overloading, where fuzzy logic controller is used. We developed a mathematical model that takes into account the features of the data transfer system with an active queue management, which keeps the queue length in the range of values close to a given reference value of the queue length. The method of hysteretic control for incoming load with two thresholds was used as a basis of the model. The mathematical model is a queuing system with a threshold control, which is designed for the analysis of the possibility of hysteresis in modeling of systems with active queue management. The model was described by a Markov process, for which the numerical solution of the equilibrium equations was obtained, steady state probabilities were calculated. The main probabilistic measures are the following: the mean value and the standard deviation of a queue length, and the probability for the queue length of being within the specified limits from the reference value. The numerical analysis in the load range, which includes a system overload, indicated the adequacy of the constructed mathematical model with hysteretic control and system with an active queue management based on fuzzy logic controller. The proposed fuzzy logic method was implemented for Linux kernel and the test results show better quality of service parameters than other tested methods.

I. INTRODUCTION

Despite of a steady increase data transmission rates in TCP/IP networks, a problem of traffic congestion is still remains relevant. Providing total connection speed for subscribers greater than available on aggregation site is economically justified in packet networks. Therefore, there is a possibility of router output buffer overloading and degrading a quality of service values, such as a percentage of packets lost, delay and jitter. Using the active queue control based on fuzzy logic controller (FLC) allows to effectively manage the nonlinear dynamics of the load and maintain queue length at a predetermined level, which provides stability of packets delay [1].

II. DATA TRANSMITTING SYSTEM

A directly connected to the bottleneck link router (Fig. 1) has the necessary information about congestion occurrence in the link or about the state close to it. The router could estimate the current degree of the output queue load and the current traffic intensity, and report to TCP transmitters about the needs to reduce the congestion window.

The queue overflow leads to multiple packet losses, the phenomenon of synchronization of TCP sessions and periodic

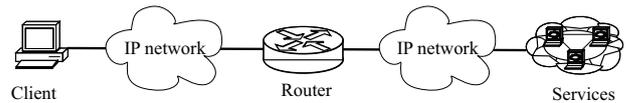


Fig. 1. IP network diagram

alternation of moments of buffer overflow and underflow. The queue emptying is also undesirable, since this affects the traffic quality of service parameter like a link utilization. The traffic intensity has a complex nonlinear dynamics on the input port of an aggregation router because of competition for the bandwidth of a large number of concurrent TCP sessions belonging to different applications with different behavior (e.g., short-time HTTP/web connection, long-live FTP connections and unmanageable UDP traffic). In today's routers, the most usable method of active queue management mechanism is a Random Early Detection (RED) [2]. The RED mechanism controls the weighted moving average queue length, and therefore allows oscillations in the instantaneous queue length. Also, RED uses a linear law for control the drop/marketing probability that the nonlinear change in the intensity of traffic will lead to inefficient management of the queue. Fuzzy logic controllers (FLC) are proven for manage systems with nonlinear dynamics [3].

A transmitting data system with buffer size B , for which is determined a value Q_{ref} ($0 < Q_{ref} < B$) of the reference queue length. The system receives a stream of packets, which are transmitted to the channel in the order "first come – first served" (FCFS) at a constant rate. In moments of t_i measurement module "Monitor" (Fig.2) checks two parameters of the system – current queue length q_i and a packets incoming intensity r_i for the most recently ended the observation interval $\Delta t_i = t_i - t_{i-1}$, where $i \geq 1$.

Module "Monitor" transfers to the module "Control Function" two values. First, the normalized value of $q_{errornorm}^i$ – deviation of the current queue length and the reference queue value, and the second, the value of the normalized packets arrive intensity in the interval Δt_i , which are calculated by the formulas:

$$q_{errornorm}^i = \begin{cases} \frac{q_i - Q_{ref}}{B - Q_{ref}}, & q_i \geq Q_{ref}, \\ \frac{q_i - Q_{ref}}{Q_{ref}}, & q_i < Q_{ref}, \end{cases} \quad (1)$$

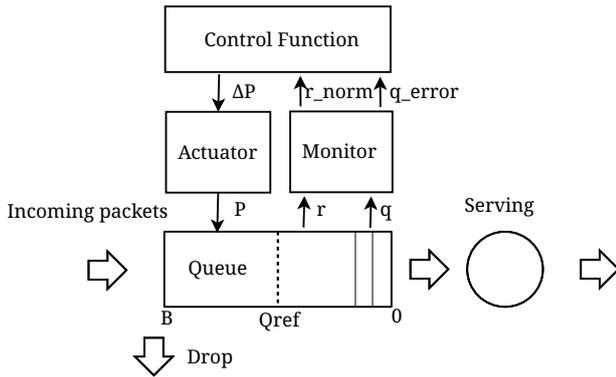


Fig. 2. A data transmission system with active queue management

$$r_{\text{norm}}^i = \begin{cases} \frac{r^i - \mu}{\mu}, & r_i \leq 2 \\ 1, & r_i > 2, \end{cases} \quad (2)$$

where μ – is an intensity of serving packets.

The module "Control Function" uses the values of the input parameters $q_{\text{errornorm}}^i$ and r_{norm}^i for the calculation of the output parameter – increment of probability of packets drop ΔP^i at the next observation interval Δt_{i+1} , $i \geq 1$. We are considering that the system control function is obtained by the method of fuzzy logic [4], [5], [6], and its shape is shown in Fig. 3.

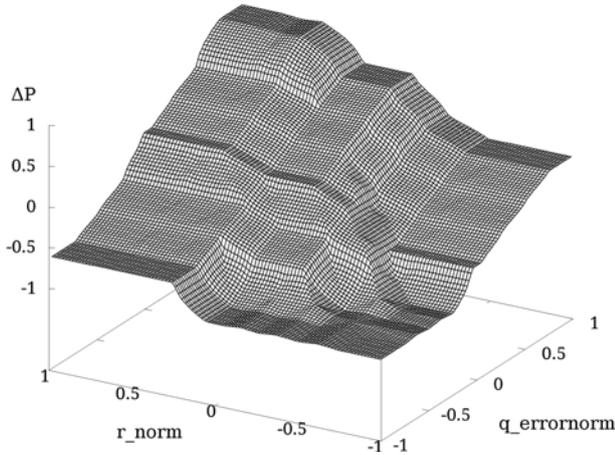


Fig. 3. Control function of the active queue management

The drop probability increment $\Delta P^i \in [-1, 1]$ is used by the "Actuator" module for calculating the probability of packets drop by the formula (3)

$$P^{i+1} = P^i + \Delta P^i \cdot P_{\text{max}}, \quad i \geq 1, \quad (3)$$

where $P_{\text{max}} \in [0, 1]$ – maximum drop probability increment on any observation interval. Thus, module "Actuator" drop packets on the interval of Δt_i with probability P^i on the input of router buffer.

The objective of this paper is to construct a mathematical model especially taking into account the operation of a data transmission system with active queue management, and research opportunities to use a hysteresis load control system.

In section III we construct the model as a Queuing System (QS) with hysteresis control load similarly [7], [8], Section IV shows the system of equations Markov process describing the operation of the QS, and in section V we give an example of a numerical analysis of its basic probability-time characteristics.

III. QUEUEING SYSTEM MODEL WITH HYSTERETIC CONTROL

Lets make a discrete system of queue control function parameters by introducing a parameter $r \in \{0, 1, 2, 3, 4\}$, which characterizes the level of the intensity of traffic load, and the parameter $s \in \{0, 1, 2, 3, 4\}$ overload status, which determines the level of system load, i.e. degree of the data buffer fulfilment. In this case, the states with the same level of the traffic intensity r can match to different system overload status s .

Table 1 shows the parameter values r corresponding to the ranges of intensity r_{norm}^i , and Table 2 shows compliance of overload status values.

TABLE I. THE LEVEL OF THE TRAFFIC LOAD INTENSITY

Load level	"r" value	" r_{norm}^i " range
Small	0	[-1; -0,6)
Average	1	[-0,6; -0,2)
Normal	2	[-0,2; 0,2)
High	3	[0,2; 0,6)
Overload	4	[0,6; 1]

TABLE II. SYSTEM OVERLOAD STATUS

Overload status	"s" value
Small load	0
Normal load	1
Overload start	2
Overload	3
Load drop	4

Two thresholds was introduced in order to control the intensity of the proposed load in the queue system. The lower threshold L and an upper threshold of H was selected in correspondence to relation $L < Q_{\text{ref}} < H$. While the total number of requests in the system exceeds $H - 1$, the system operates normal mode (small and normal load states). If the number of requests in the system has exceeded the value of $H - 1$, the system goes into overload mode (start of overload and overload states). The system remains in overload mode until the number of requests in the queue q reaches values $L - 1$ or $B - 1$. Upon reaching the queue length value $L - 1$, system returns to the normal operation mode, and when the queue length reaches value $B - 1$ the system goes into load drop mode, which remains as long as the queue length exceeds threshold of H , and returns to the overload mode when the number of requests in the system becomes equal to H . Fig. 4 shows the queuing system with thresholds.

The queuing system with a buffer storage capacity B , the lower threshold of L , the upper threshold of H and reference value of the queue length Q_{ref} receives a flow of request with Poisson distribution with intensity $\lambda(s, q, r)$, depending on the system's states. The requests are serviced in the order received

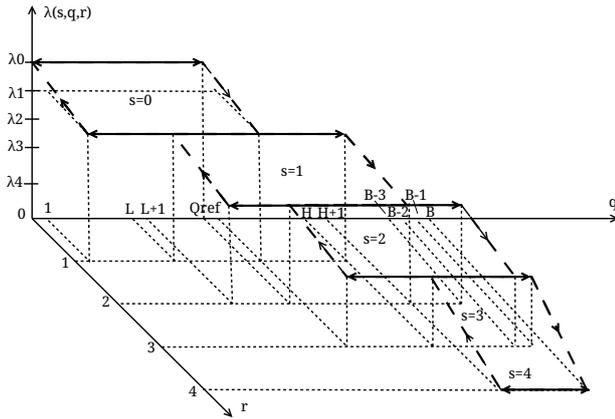


Fig. 5. Hysteresis control load with active queue management

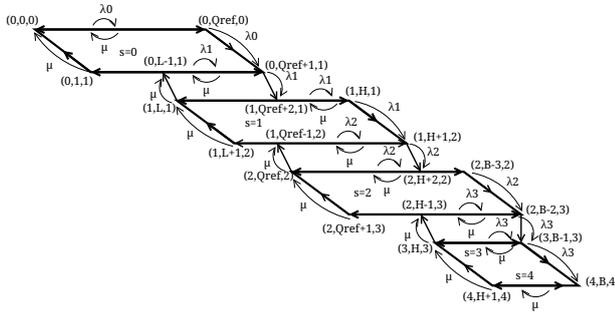


Fig. 6. The diagram of the Markov process transition intensities

The system of equilibrium equations in a matrix view with normalization condition could be shown as follows:

$$\begin{cases} \vec{p} \cdot \mathbf{A} = 0 \\ \vec{p} \cdot \mathbf{1} = 1 \end{cases}, \quad (9)$$

where \vec{p} – is a vector of stationary probability, \mathbf{A} – matrix of transition intensity.

The numerical analysis of the probability characteristics of the studied QS was done by solving of equilibrium equations by using the numerical method of LU-decomposition.

V. NUMERICAL ANALYSIS

The set of states Y denote the QS states, in which the length of the queue is stay in the range of $q \in [L, H]$, and could be represented in the follow form:

$$Y = Y_0 + Y_1 + Y_2, \quad (10)$$

where

$$Y_0 = \{(s, q, r) : s = 0, r = 0, L \leq q \leq Q_{ref}\} \cup \{(s, q, r) : s = 0, r = 1, L \leq q \leq Q_{ref} + 1\};$$

$$Y_1 = \{(s, q, r) : s = 1, r = 1, L \leq q \leq H\} \cup \{(s, q, r) : s = 1, r = 2, L + 1 \leq q \leq H\};$$

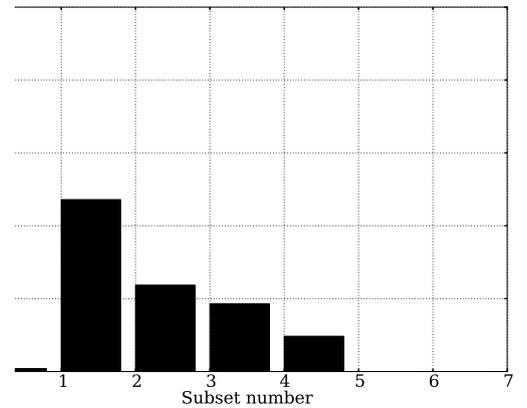
$$Y_2 = \{(s, q, r) : s = 2, r = 2, Q_{ref} \leq q \leq H\} \cup \{(s, q, r) : s = 2, r = 3, Q_{ref} + 1 \leq q \leq H\}.$$

 TABLE III. STRUCTURE OF TRANSITION INTENSITY MATRIX \mathbf{A}

	$X_{0,0}$	$X_{0,1}$	$X_{1,1}$	$X_{1,2}$	$X_{2,2}$	$X_{2,3}$	$X_{3,3}$	$X_{4,4}$
$X_{0,0}$	D	U	0	0	0	0	0	0
$X_{0,1}$	L	D	U	0	0	0	0	0
$X_{1,1}$	0	L	D	U	0	0	0	0
$X_{1,2}$	0	0	L	D	U	0	0	0
$X_{2,2}$	0	0	0	L	D	U	0	0
$X_{2,3}$	0	0	0	0	L	D	U	0
$X_{3,3}$	0	0	0	0	0	L	D	U
$X_{4,4}$	0	0	0	0	0	0	L	D

The system should be optimized in order to achieve a maximum of probability $P(Y)$ which represents the states when queue length deviation from the reference value is within the $L \leq q \leq H$ thresholds.

We set buffer storage capacity $B = 50$ in order to carry out the numerical analysis. The value of the reference queue length was set to $Q_{ref} = 25$ and the thresholds are $L = 20$ and $H = 30$. Note that cardinality of the set of QS states data values, and, therefore, dimension of the system of equilibrium equations is equal to 160. We choose the values of intensities of the proposed load λ_r as that the probability of $P(Y)$ has reached its maximum value. In this our example the values of the intensities were $\lambda_0 = 1.95$, $\lambda_1 = 1.2$, $\lambda_2 = 0.47$, $\lambda_3 = 0.43$, $\lambda_4 = 0$ and service intensity was $\mu = 1$. This values had allowed to reach probability value $P(Y) = 0,68$. The stationary probabilities of the system being in subsets of states Markov process $X(t)$ are shown in Fig. 7.


 Fig. 7. The probabilities of queuing systems subsets $X_{0,0}, X_{0,1}, X_{1,1}, X_{1,2}, X_{2,2}, X_{2,3}, X_{3,3}, X_{4,4}$

On Fig. 8 are shown the calculated for our numerical example dependencies of the average queue length from traffic load coefficient $\rho = \lambda/\mu$ in the range including system overloading $\rho \in [0, 2]$.

From the graphs in Fig. 8 follows that with the traffic load increases the system goes to overload mode $\rho > 1$ and the average queue length tends to the reference value of the queue length $Q_{ref} = 25$.

The calculated data for well known model M/M/1/n with $n = 50$ and $\rho < 1$, and the simulation data have obtained in Network Simulator-2 (NS-2) software are also shown on Fig. 8 for comparison.

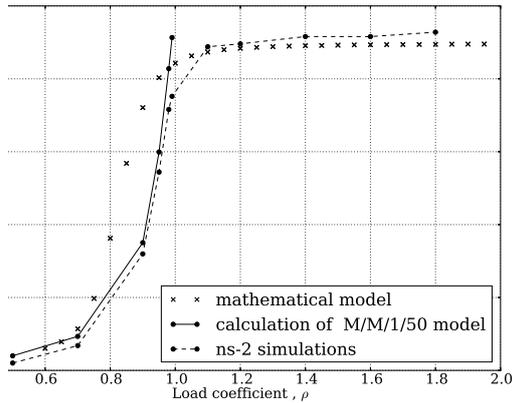


Fig. 8. The dependence of average queue length from the load coefficient ρ

VI. IMPLEMENTATION IN LINUX KERNEL

The developed fuzzy logic queue management method is implemented in a Linux-kernel module as a queue discipline for Linux-router application. For testing purpose FLC module was loaded in Linux kernel space on PC for egress traffic treatment. The scheme of packets treatment in Linux kernel is shown on Fig. 9 [9].

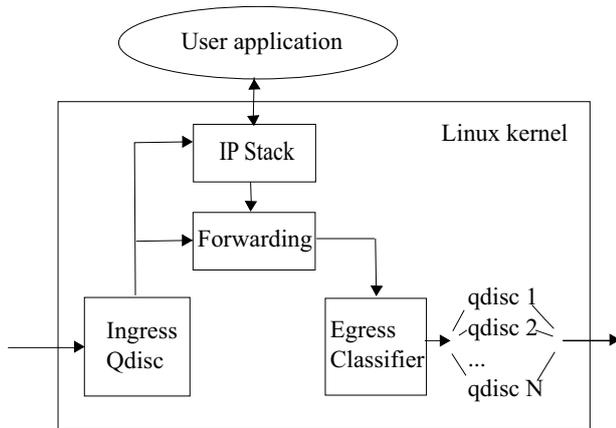


Fig. 9. Packets treatment in Linux kernel

Modified Linux utility *tc* (Traffic Control) is used for queue discipline configuration in kernel from user space. The control function described on Fig. 3 was designated as a pre-calculated table in order to avoid float point calculation prohibition in Linux kernel space. In our experiment user application generates TCP traffic which are passed through egress classifier and FLC egress queue discipline to the network. Additionally we use egress classifier rules to separate test traffic from PC management traffic and limit rate of test traffic to 50 Mbit/s. The traffic classification scheme is show on Fig. 10.

Hierarchical Token Bucket (htb) is used as a root queue discipline for rate limiting traffic. The management traffic passed over default class 1:10, but test traffic passed to class 1:50 with ceil rate 50 Mbit/s and was assigned to FLC queue discipline. Linux utility *iptables* is used for pre-marking test

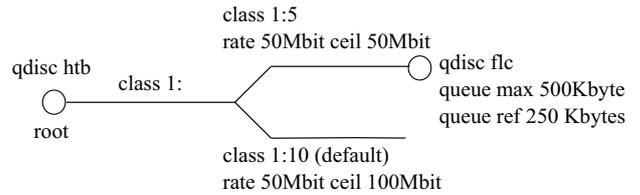


Fig. 10. Egress traffic classification scheme

traffic in order to classifier could separate it to the designated class. Explicit Congestion Notification (ECN) was enabled in Linux kernel in order to use packet marking instead of packet drop [10]. A 10 msec delay was emulated between TCP transmitter and receiver, because both were running on the same PC in virtual environment. The experiment scheme is presented on Fig. 11.

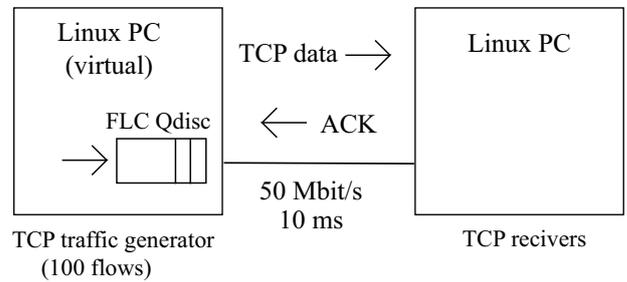


Fig. 11. Experiment scheme

We set up queue target length to 250 Kbyte and the maximal 500 Kbyte. The FLC module check actual queue length and traffic rate every 6 msec and could change drop probability maximal on $8E-5$ at each 6 msec interval. From the experiment beginning 100 TCP flows start to send traffic with 964 TCP segment size. At the moment of 100 sec from the beginning 50 TCP flows stop to transmit the traffic and after 100 sec of interruption again continue to transmit in order to emulate real network traffic dynamics. At the moment of 300 sec from the beginning all flows stop and the experiment is finished. Additionally UDP traffic was sending at 128 kbit/s rate with 980 bytes datagram size during the test. The queue length evolution is show on Fig. 12.

The Fig. 12 displays that implemented FLC method could keep queue length on target level 250 Kbyte despite of traffic dynamics and under overload conditions. We could also see that PC could not establish 100 new TCP connection simultaneously, it takes about 25 sec for start and transmission did not finished in 300 seconds. It was sent about 2 GBytes (or 2 millions of packets) of traffic over the FLC queue discipline during the test and it utilized whole available bandwidth. Packets drop rate has low level of hundreds packets owing to ECN support despite of link overload. The UDP flow shows low level jitter 3 msec and 3% packets loss. Higher UDP loss in comparison to TCP traffic is explained that FLC queue discipline randomly mark TCP packets with ENC support, drop UDP packets without ECN support.

The same test scenario was used for queue discipline RED and TailDrop in order to compare data transmission quality of service parameters between different disciplines. We repeated

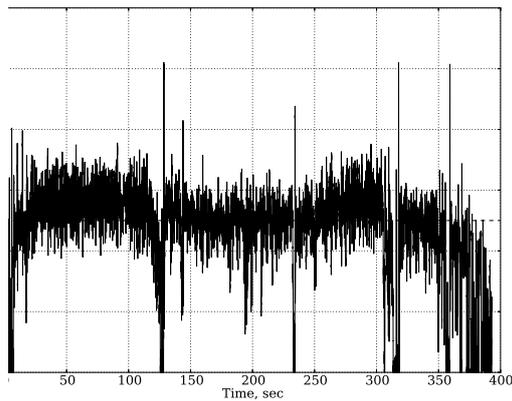


Fig. 12. Queue length evolution in Linux-router buffer with FLC discipline

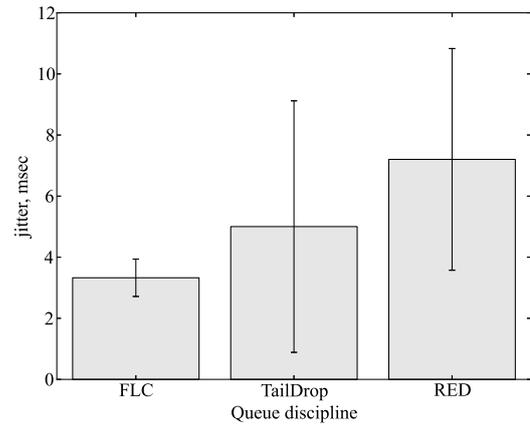


Fig. 14. Mean values for jitter measurements for UDP flow

the experiment 5 times for each discipline in order to calculate mean values and confidence intervals. For RED queue discipline the minimal threshold was set to 125 Kbyte, the maximal threshold to 375 Kbyte and the queue limit 500 Kbyte. In according to RED parameters setting recommendation the maximal threshold is 3 times higher than minimal. Average queue length is equal to FLC queue target value 250 Kbyte. The drop probability for RED algorithm was set to 0.02. The ECN support was enables as for FLC queue discipline test. For TailDrop test the queue limit was set to 500 Kbyte and ECN is not supported. The mean values for packets loss for three tested queue disciplines: FLC, TailDrop and RED for whole traffic are shown on Fig. 13.

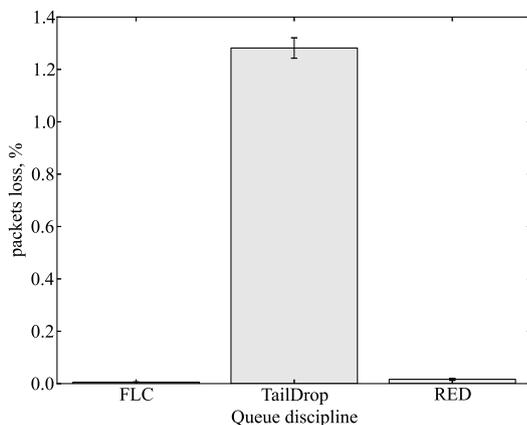


Fig. 13. Full packets loss mean values

Owing to ECN support the FLC and RED show very low packets loss rate because they mark packets in advance for congestion avoidance instead of drop. The TailDrop discipline shows much higher packets loss due to queue limit overflow.

The mean values of jitter levels for UDP constant rate flow are shown on Fig. 14. The vertical lines on the figure displays limits for confidence intervals at 95% trustworthy level.

The minimal mean value of the jitter for FLC queue

discipline and smaller confidence interval confirm that FLC method could better stabilize queue length than other tested methods. The RED discipline displays higher jitter values because RED method could control only average queue length and permit instantaneous queue oscillations between the configured thresholds. The TailDrop discipline displays unstable queue length behavior.

VII. CONCLUSION

In this paper a mathematical model of the queuing system with active queue management was designed. The model was constructed to investigate the possibility of using hysteretic policy under active management queue and qualitative analysis of its probability-time characteristics. The proposed FLC queue management method was implemented in Linux kernel as a queue discipline to prove the concept against other well-known approaches.

The comparison of results of the model numerical analysis with the data obtained from the simulations shows the adequacy of the constructed mathematical model with hysteretic control system with active queue management based on fuzzy logic controller. The Linux kernel implementation confirms the FLC method possibility to keep a queue length at target level. The comparison of testing results obtained on different queue disciplines shows that FLC method allows to achieve better quality of service parameters as a packets loss and jitter. Our future work may include the application of the proposed model to the analysis of congestion in NGN networks.

ACKNOWLEDGMENTS

The reported study was partially supported by the RFBR, research projects No. 12-07-00108, 13-07-00953.

REFERENCES

- [1] R. Fengyuan, R. Yong and S. Xiuming, "Design of fuzzy logic controller for active queue management", *Computer Communications*, no. 25, 2002, pp. 874-883.
- [2] S. Floyd, V. Jacobson, "Random Early Detection gateways for Congestion Avoidance", *IEEEACM Transactionso and Networking*, vol. 1, no. 4, August 1993, p. 397-413.

- [3] C. Chrysostomou, A. Pitsillides and Y.A Sekercioglu, "Fuzzy explicit marking: A unified congestion controller for Best-Effort and Diff-Serv networks", *Computer Networks*, no. 53, 2009, pp. 650-667.
- [4] A. Piegat, *Fuzzy Modeling and Control*. Moscow: Binom. Laboratory of knowledge, 2009.
- [5] V. Deart and A. Maslennikov, "Investigation of influence of data transmission link parameters to queue management procedures", *T-Comm*, no. 7, 2012, pp. 77-81.
- [6] V. Deart and A. Maslennikov, "Fuzzy logic queue discipline processing over bottleneck link", *Proceedings of 11th conference of Open Innovations Association FRUCT*, St-Petersburg, April, 2012, pp. 40-46.
- [7] P.O. Abaev, Yu.V. Gaidamaka and K.E. Samouylov, "Signaling load hysteretic control in the SIP-servers network", *Bulletin of Peoples' Friendship University of Russia. Series "Mathematics. Informatics. Physics"*, no. 4, 2011, pp. 54-71.
- [8] Yu.V. Gaidamaka, "Model with threshold control for analyzing a server with an SIP protocol in the overload mode", *Automatic Control and Computer Sciences*, vol. 49, no. 2, 2013, pp. 65-76.
- [9] *Linux Advanced Routing & Traffic Control HOWTO*, Web: <http://lartc.org/howto/>
- [10] K. Ramakrishnan, S. Floyd, D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", *RFC-3168*, Sep. 2001.