Fuzzy Logic Queue Discipline Processing Over Bottleneck Link

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

Abstract

Traffic congestion occurs inevitably in aggregated channel when access networks connect to transport network over a bottleneck data transmission link. The reason of congestion is a lack of bandwidth and continuously customer requirements growth in transmission rate to network content. In this article is investigated influence of data transmission link parameters, as an available bandwidth and packet transmission delay, to queue discipline processing in access router. Fuzzy logic controller (FLC) is considered as an active queue management discipline processing with explicit congestion notification (ECN). Traffic congestion is simulated in a bottleneck link between two routers in Network simulator-2 (NS-2) software. Queue discipline processing was estimated by calculation the quality of service indicators: packets loss, average packet delay and packet delay variation. Statistical analysis of simulation result have given the possibility to define dependence between the quality of service indicators and the data link parameters applicable to considered queue discipline.

Index Terms: Active queue management, congestion avoidance, fuzzy logic, FLC, network simulator, NS-2, TCP/IP.

I. INTRODUCTION

The studied subject is the router, one of which tasks is to prevent overloads in the data transmission channel. Only directly connected to the bottleneck link router has the necessary information on the occurrence of congestion in the link or the state close to it. The router should estimate the current degree of congestion of the output queue and the current increase or decrease the intensity of the load, and report to TCP transmitters on the need to reduce the congestion window. Protocol ECN (Explicit Congestion Notification) is used by appropriate markings of passing packets, as described in the document RFC-3168 [1]. In case of the threat of congestion rather than marking, the packets can be dropped in random order if there is no support for ECN. Otherwise, the queue overflow leads to multiple packet losses, the phenomenon of synchronization of TCP sessions and periodic alternation of moments of buffer overflow and underflow. Emptying the queue is also undesirable, since this affects the quality parameter as the link utilization.

There is traffic with 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). We consider the case where competing connections have the same class of service and follow through the same queue. In today routers, the most common methods of active queue management mechanism are based on Random Early Detection (RED) [2]. RED mechanism controls the weighted moving average queue length, and therefore allows for oscillations in the instantaneous queue length. Also, this mechanism uses a linear law for control the drop/marking 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]. In such systems are typically available qualitative information about the current state of the system (for example, the relative packet arrival rate of the incoming traffic and the degree of the output queue filling), but the system state change law is unknown, or it is very difficult to describe. The data for qualitative estimates of the system state can be formulated as a set of rules to control the queue length. That is, for example, under highly filled queue in combination with high intensity of the input traffic, in order to reduce the current queue length, the drop/marking probability should be significantly increased.

A comparison between different methods of active queue management for fixed parameters of the channel is shown in [4]. This paper presented the results of simulations of the network performance with suggested method of queue management based on fuzzy logic controller. The simulation is done for the different parameters of the bottleneck link, such as available bandwidth and delay of packets in the link.

II. MAIN PART

A. Queue Management Module

Network Simulator-2 (NS-2) [5] software was used for simulations. A new module written in C++ programming language was added to NS-2 and the system compiled with the new module. The new module implements the method of active queue management based on a fuzzy controller [4] and has the following structure which is shown in Fig. 1.

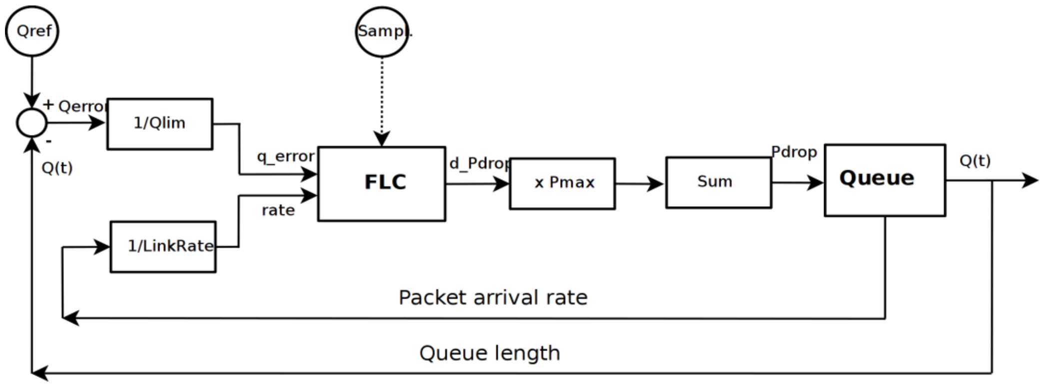


Fig. 1. The structure of the queue management module

The software module gets the current queue length *Q(t)* and calculates the value of error *Qerror* as a difference between the current value and the specified reference *Qref*. The error value is normalized by the maximum value of the queue size *Qlim*. The normalized error value *q_error* enters to the first input of controller FLC. The second measured parameter is the rate at which packets arrived in the queue. It enters to the second input of FLC after normalizing by maximum link rate *LinkRate* (Fig. 2).

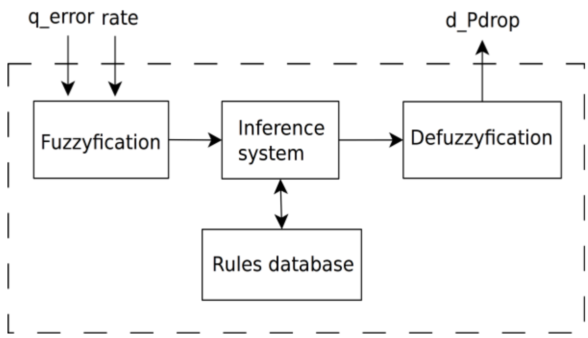


Fig. 2. The structure of the fuzzy logic controller (FLC)

These two input variables *q_error* and *rate*, and an appropriate set of rules, using expert estimation, are the data on which the FLC controller decides to change the current value of the increment of the drop/marking probability *d_Pdrop*. The “Fuzzification” part of FLC calculates the values of input variables membership functions μ, i.e. degree of certainty that selected input variable belongs to fuzzy set (for example, fuzzy sets "Large", "Low" and "Average"). Actually 7 fuzzy sets per each value in the model were used. Membership functions are chosen so that the sum of the all functions of the input variable equal to one. Triangular and trapezoidal functions used for simplicity of calculations. One of the widely used algorithms for fuzzy inference is Mamdani algorithm. According to the algorithm first computes the values of truth conditions for each rule (the operation minimum). All fuzzy sets derived from the rules are merged together and formed a single aggregated fuzzy set *C* (the operation maximum). Bringing to the definition, that is, computation of the actual value of the output variable is performed using a discrete method of center of gravity:

$$P_{drop} = \frac{\sum_{j=1}^{m} y_j \cdot \mu_C(y_j)}{\sum_{j=1}^{m} \mu_C(y_j)} \quad (1)$$

where – y is the discrete set of output variable values, m - number of values, and - μ is the aggregated membership function of fuzzy set *C*.

FLC calculates a new value of the drop probability increment *d_Pdrop* with a sampling frequency *Sampl*. The function of drop probability increment is shown on Fig. 3. As shown on Fig. 3 the output variable of drop probability is increasing when increasing the input variables of the queue error and packet arrival rate. Then the calculated increment value *d_Pdrop* is multiplied by the constant *Pmax,* which is the maximum drop probability change during time interval *Sampl*. Finally the function *Sum* adds increment to the current drop/mark probability *Pdrop,* which is then used for queue management.

B. Simulation modeling

NS-2 software simulates congestion in the bottleneck link between two routers in order to estimate the performance of queue management. The link bandwidth was set to following values: 10, 20, 35 and 50 Mbit/s for the different experiments. The packet delay in the link was set to 5, 10, 20, 50 and 100 msec. So, totally 20 experiments with a duration of 600 seconds continuously of modeling time for each one were carried out.

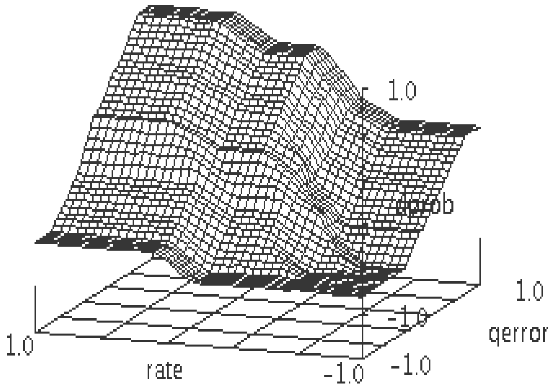


Fig. 3. Normalized drop probability increment *dprob* function of packet arrival rate and queue error

Multiservice traffic is transmitted over the link with 1000 bytes packets size. The traffic consists of 3 types: long-live TCP sessions (created by 100 simultaneous FTP applications), short-live TCP sessions (created by HTTP applications with an intensity of 50 new connections per second) and unmanageable UDP traffic with constant rate of 128 kbit/s full duplex (Fig. 4).

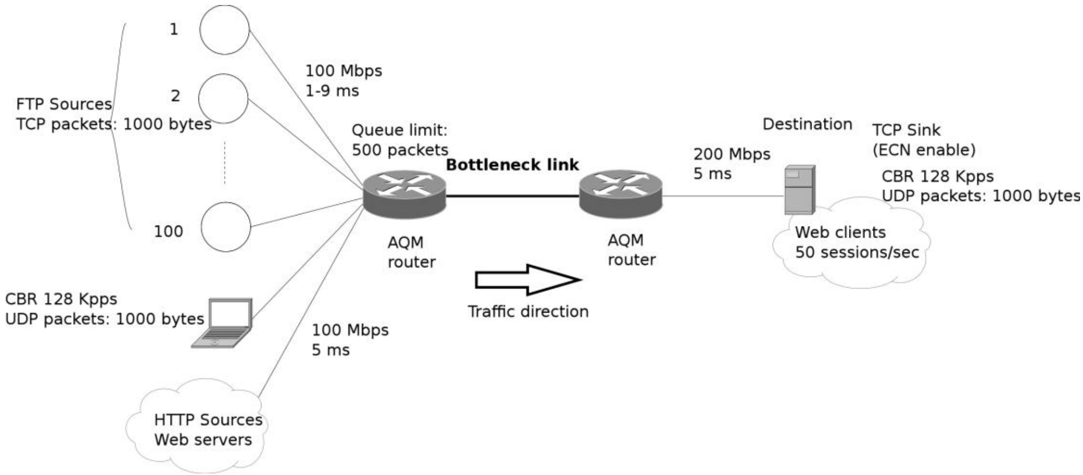
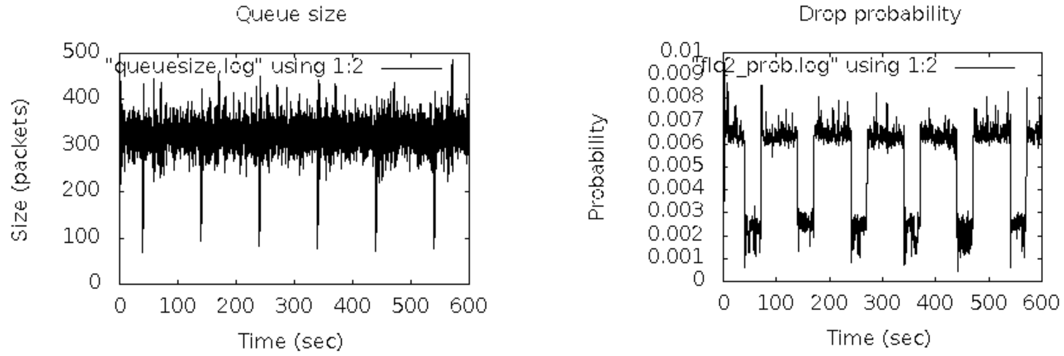


Fig. 4. Network diagram for simulation in NS-2

The "NewReno" implementation of TCP protocol was used with enabled ECN packet marking, to indicate the router to the transport layer of the possible congestion in the link. Packets were randomly dropped in the case of congestion of UDP traffic. FTP sources had different time delays in the link to the router. The delays are uniformly distributed on the interval from 1 to 9 ms. FTP sources star simultaneously transmission at the beginning and then follow the next scenario in order to simulate the network dynamics. 50 FTP sources stop the transfer at the moment of 40 seconds and resume the transfer at the moment of 70 seconds at each 100 seconds interval of total 600 seconds of modeling time.

The frequency of measurements of FLC was set to 6 ms and the maximum value of the drop probability increment to 8E-5. The desired queue length set to 300 packages, with a maximum queue size of 500 packets.

The utilization of the bottleneck link in the all experiments was close to unity. The queue length and drop/mark probability evaluation graphs (for the experiment at a rate in the channel of 50 Mbit/s and a delay of 5 ms) are shown in Fig. 5a and Fig. 5b, respectively:



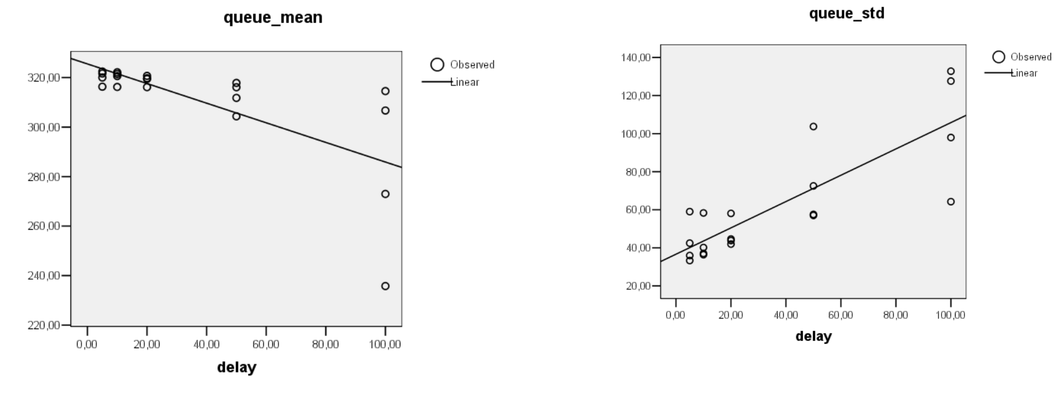
a) Queue length (in packets) b) Drop probability

Fig. 5. Evaluation of queue length and drop probability during the simulation

The queue management method demonstrates an automatically adjusting of the value of the drop/mark probability. FLC keeps the current queue length around the given value of 300 packages in depending on the traffic intensity. The average queue length was 322 packets with a confidence interval of ± 1 packet, and standard deviation was 33 packets with a confidence interval of ± 1 packet. The confidence intervals were calculated with 95% accuracy using the five successive time intervals of 100 seconds. The steady-state process was confirmed by stability of the average queue length value.

C. The analysis of the simulation results

Regression analysis was performed to determine the dependence of the quality indicators of queue management method from the bottleneck link parameters. The found dependencies are shown in Fig. 6:

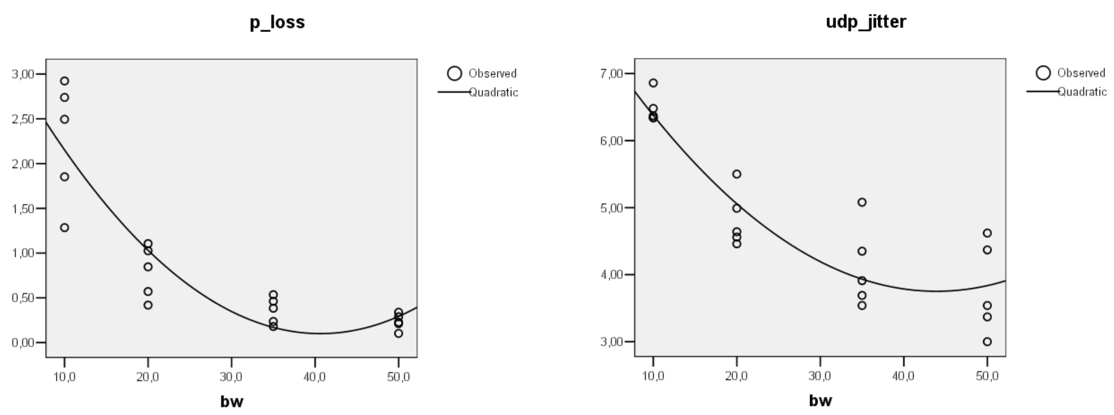


a) Dependence of the average queue length *queue_mean* (packets) from the link delay (ms) b) Dependence of the standard deviation of the queue length *queue_std* from the link delay (ms)

Fig. 6. Dependencies of average queue length and standard deviation from the link delay

Fig. 6a shows the dependence of the average queue length on the delay in the link. The link average queue length decreases with increasing delay. The standard deviation of queue average increases linearly in the same case (Fig. 6b). So, we could see that queue length is more stable for the smaller values of the delay. The smallest value of standard deviation was obtained in the experiments with link delay of 5 ms. This behavior of the queue management mechanism could be accounted by the fact that for large delays TCP transmitter has to wait for the congestion notification. The information about congestion comes too late and it turn the affects to quality of queue management.

Total packets losses mainly depend on the available bandwidth in the link (Fig. 7a). The packets loss is reduced to a quadratic law with increasing of the bandwidth. Indeed, the higher link bandwidth means a less service time of packets in the queue. The probability of queue overflow and the packet loss are respectively less. The packet loss basically corresponds to uncontrolled UDP traffic. The losses are minimal for long-live TCP connections and FTP traffic. Even in the worst case it did not exceed 0.2% loss due to implementation of ECN marking instead of packets drop.



a) Dependence of the percentage of packets loss *p_loss* (%) from the link bandwidth *bw* (Mbit/s)

b) Dependence of the jitter of UDP packets *udp_jitter* (ms) from the link bandwidth *bw* (Mbit/s)

Fig. 7. Dependencies of packets loss percentage and UDP packets jitter from the link bandwidth

The value of the jitter of UDP packets also decreases quadratically with increasing of the link bandwidth (Fig. 7b). This is also shows an increase of the quality of the queue managing method in case of reducing the service time at the same number of TCP connections.

III. Conclusion

The simulation of the developed method of queue management based on FLC in NS-2 software demonstrated the following dependencies on wide range of bottleneck link parameters:

- the stability of the queue length increases linearly with the decrease of link delay;
- the percentage of lost packets and the average packet delay variation decreases with increasing the bandwidth for a quadratic law.

A queue management mechanism based on fuzzy logic controller could effectively keep the queue length around a given value in a complex traffic condition with non-linear dynamics. The suggested queue management method in conjunction with ECN marking minimized the packets losses of long-live TCP connections.

In further work is planned to implement the developed queue management method on a Linux based router with open source software that will allow to test the method in a real network.

REFERENCES

[1] K. Ramakrishnan, S. Floyd, D. Black, The Addition of Explicit Congestion Notification (ECN) to IP, RFC-3168, Sep. 2001.

[2] S. Floyd, V. Jacobson, Random Early Detection gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, p. 397-413.

[3] C. Chrysostomou, A. Pitsillides, Y.A. Sekercioglu, Fuzzy explicit marking: A unified congestion controller for Best-Effort and Diff-Serv networks, Computer Networks 53 (2009), p. 650-667.

[4] A.G. Maslennikov, Active queue management techniques for routers / Network-Journal. Theory and practice, No.2 (19) 2011. http://network-journal.mpei.ac.ru.

[5] The Network Simulator, NS-2, http://nsnam.isi.edu/nsnam/.