

A SELF ORGANIZING MAP BASED APPROACH FOR CONGESTION AVOIDANCE IN AUTONOMOUS IP NETWORKS

Özen Yelbaşı*, Emin Germen†

Abstract: This work presents a Self Organizing Map (SOM) based queue management approach against congestion in autonomous Internet Protocol (IP) networks. The new queue management approach is proposed with consideration to the pros and cons of two well-known queue management algorithms: Random Early Detection (RED) and Drop Tail (DT). At the beginning of this study, RED and DT are compared by observing their effects on two important indicators of congestion: end-to-end delay and delay variation. This comparison reveals that the performances of RED and DT vary according to the level of global congestion: under low congestion conditions, when packet losses caused by congestion are unlikely, DT outperforms RED; while under high congestion, RED is superior to DT. The SOM based approach takes into account the variations in the global congestion levels and makes decisions to optimise congestion avoidance. A centralized observation unit is designed for monitoring global congestion levels in autonomous IP networks. A traffic flow is generated between each router and the observation unit so as to follow the changes in the global congestion level. For this purpose, IP routers are specialized to send packets carrying queue length information to the observation unit. A SOM based decision mechanism is used by the observation unit, to make predictions on the future congestion behavior of the network and inform the routers. Routers use this information to update their congestion avoidance behavior, as their ability to update their RED parameters is enhanced by the congestion notifications sent by the observation unit. In this work, multiple simulations are undertaken in order to test the performance of the proposed SOM-based method. A considerable improvement is observed from the point of view of end-to-end delays and delay variations, by comparison with DT and RED as used in recent IP networks.

Key words: *Self Organizing Map, congestion avoidance, random early detection, drop tail*

Received: November 16, 2012

DOI: 10.14311/NNW.2015.25.007

Revised and accepted: April 1, 2015

*Ö. Yelbaşı – Corresponding Author, Electrical and Electronics Engineering Department, Anadolu University, Eskişehir (26555), Turkey, Tel.: +90 222 3350580, E-mail: oyelbasi@anadolu.edu.tr

†E. Germen, Electrical and Electronics Engineering Department, Anadolu University, Eskişehir (26555), Turkey, Tel.: +90 222 3350580, E-mail: egermen@anadolu.edu.tr

1. Introduction

Congestion occurs in a network when the capacity of resources (such as buffer space in a router and link bandwidth) become insufficient, generally as a result of overload, to meet overall demand [8, 25]. Persistence of congestion increases the quantity of packet losses, end-to-end delays and delay variations (jitters), on which most Quality of Service (QoS) requirements have constraints. In order to satisfy QoS requirements, the problem of congestion demands serious attention. As described in [25], congestion control and congestion avoidance are two different approaches used for dealing with congestion: congestion control is a reactive approach that is applied to already congested networks, while congestion avoidance is used as a prevention against congestion. In this work, congestion avoidance is examined under the scope of queue management algorithms and global congestion behavior of multi-bottleneck IP networks is considered. As explained in [27], main considerations of this study may be summarized through the following steps:

- presentation of a centralized observation method for monitoring global congestion
- design of an observation unit that collects queue length information from router output interfaces
- training of a Self Organizing Map (SOM) with the collected queue length information
- utilization of the SOM to visualize global congestion and to make predictions on the future of congestion
- generation of a congestion notification traffic to inform routers about the global congestion level

The new queue management approach, which is proposed as a part of this work, aims for the treatment of end-to-end delays/delay variations, by controlling the number of packets queued in router output interfaces. As a result, necessary updates are performed on the new queue management technique, upon receiving the congestion notification sent by the observation unit. In this work, a centralized solution to congestion avoidance problem in the multi-bottleneck environment is proposed. However it may also be adopted to solve each router's individual congestion avoidance problem. In fact, providing decentralized solutions for congestion arising at routers is the main aim of the SOM based approach. Performance of the SOM based approach is verified by the Opnet Modeler simulation program. Simulation results are used to make comparisons with DT and RED algorithms, which, in the existing network infrastructure, would be largely used against congestion. The results of the centralized approach presented in this study, show the potential of the SOM based approach also for solving the decentralized congestion avoidance problem.

This paper is organized as follows: Section 2 is a motivation for the reader. Principles of passive and active queue management are explained here by referencing to well-known algorithms such as DT and RED. Main problem statement, utilization of SOMs in monitoring tendency of global congestion, definition of some

parameters used in statistical data analysis and development of the new queue management approach are all presented in Section 3. In the Section 4, details are given of simulations investigating the performance of the new queue management approach and results are discussed. Last section presents concluding remarks.

2. Bottlenecks and queue management

Bottleneck is a general name given to nodes (such as router output interfaces) or transmission links shared by multiple traffic flows. It is well-known that congestion problems arise at bottlenecks as a result of heavy traffic load and limitations in link/queue capacities. For example, congestion is likely to occur at bottleneck links when the demand of multiple traffic flows sharing the same link goes beyond the available transmission capacity. Each output interface of a router is also a potential bottleneck for the network and local congestions may occur at these nodes. Packets are either queued or dropped at bottlenecks, therefore the management of queues at bottlenecks is an important challenge. Congestion avoidance, that is to say, depends on queue management algorithms being employed at bottlenecks. Queue management may be studied under two main types, as the Passive Queue Management (PQM) [2] and the Active Queue Management (AQM), which are explained below with example approaches.

Queue management first originated with the development of PQM algorithms. The simplest and most widely used of which is DT, whose operation principles almost characterize PQM. A DT applied queue accepts arriving packets until the maximum queue length is reached; at which point all arriving packets are rejected (dropped) pending sufficient space being gained with transmissions [14].

As defined in [3], AQM is a preventive queue management technique that predicts congestion before it occurs and informs end users to decrease the amount of traffic and thus prevents packet losses related with buffer overflow. For the purpose of improving queue management and congestion avoidance in the Internet, the utilization of the RED algorithm is suggested for providing AQM in routers [14]. Maintaining an upper bound on the average queue length, avoiding biases against bursty traffic and providing a solution for global synchronization are the achievements of RED that is presented in [7]. In the RED algorithm, marking (dropping) probability for the new arrival, p_a , is calculated as follows [7]:

$$p_a = p_b / (1 - C \cdot p_b),$$

where p_b is the marking probability and C is the number of arrivals since the last marking (dropping).

The partial expression of p_b is given below:

$$p_b = \begin{cases} 0, & q_{ave} < t_{min} \\ \frac{p_{max}(q_{ave} - t_{min})}{t_{max} - t_{min}}, & t_{min} \leq q_{ave} \leq t_{max} \\ 1, & t_{max} < q_{ave} \end{cases} ,$$

where q_{ave} is the average queue length and p_{max} is the value of packet marking probability for $q_{ave} = t_{max}$. The value of q_{ave} is updated at packet arrival times by the following equation:

$$q_{ave} = q_{ave_p} (1 - w_q) + w_q q,$$

where q_{ave_p} is the previous value of the average queue length, q is the current value of the queue length and w_q is the queue weight. Rules for assigning values to t_{min} , t_{max} and w_q are also given in [7].

There are some difficulties/drawbacks related with RED implementations. It is stated in [6] that the performance of RED varies according to the variations in congestion status, owing to static parametrization. Such variations in the queue management performance cause an instability observable as oscillations in queue lengths [3]. In [4], attention is drawn to long jitter (delay variation) characteristic of RED when traffic is non bursty, and in [1], to its inability to provide high link utilization and low delay simultaneously during congestion. These weaknesses of RED have attracted attention among network researchers and various RED variants are proposed to improve queue management quality. In [19] an adaptive nonlinear active queue management method has been proposed. Here it has been argued that the RED performance is greatly affected by the shape of the incoming traffic and the parameter setting has a substantial influence on the effectiveness of the RED algorithm. In this work a dynamic parameter estimation model is offered in the queue management in order to reduce the effect of sensitivity of RED parameters. In another recent work [24] a control based approach is proposed to adapt the RED parameters. This work proposes an adaptive predictive congestion controller and neural based controller in order to predict the dynamics of the queue management algorithm and claims promising results. In [11] a novel autonomous adjustment controller is introduced for eliminating the shortcomings of predefined RED parameters.

The importance of AQM algorithms in TCP/IP networks is explained in [2]. In literature, it is also possible to find the applications of robust control principles on AQM algorithms. For example, stability of queue lengths is considered in studies [12] and [13] where controller designs using Proportional (P) control and Proportional-Integral (PI) control techniques are presented. A robust H_∞ controller is developed and its advantages over RED and PI schemes are presented in [23].

3. SOM and the SOM based approach

In this work, a SOM based AQM approach is proposed by considering the global congestion behavior of multi-bottleneck networks. The main motivation of the proposed method is to prevent potential deteriorations in the performance of multipath routing schemes by predicting the possible congestion in the routers. Here it focuses on the respective characteristics of congested or uncongested nodes in autonomous systems to deduce the likelihood of near-immediate congestion, and on modifying the RED parameters to increase total throughput. When routers are thus informed (since their interface queues accommodate queue management algorithms), they interpret any variation in global congestion level as a sign of future

congestion status. Being focused on the global congestion problem in IP networks has led us to follow the changes in the queues of router output interfaces. The information gathered from different interfaces is used by a SOM and, in this way, it becomes possible to deduce the future behavior of global congestion. At this point, some information about SOMs will be helpful.

3.1 SOM

SOM is a special method of artificial neural networks, which allows unsupervised learning scheme to implement a transformation from a high-dimensional space of input vectors onto a one/two dimensional array of neurons [15–17]. Fig. 1 shows a SOM configuration and a mapping example. The two dimensional SOM in Fig. 1 is composed of n neurons and each hexagonal neuron is surrounded by at most 6 neighbouring neurons. For preventing visual complexity in the figure, only the connections for the first neuron are drawn. As seen in Fig. 1, a weight vector $\mathbf{m}_i = [m_{i1}, m_{i2}, \dots, m_{il}]^T$ is associated with neuron i and an input vector $\mathbf{x} = [x_1, x_2, \dots, x_l]^T$ is accepted to be virtually connected to all neurons in parallel via weights m_{ij} ($i = 1, 2, \dots, n$ and $j = 1, 2, \dots, l$), where T stands for the transpose operation [15]. At the outset, it is necessary to have a sufficiently large collection of input vectors, with which to produce a SOM to represent this multi-dimensional data (with a limited number of neurons) and to make classifications on it, as desired. An initial weight vector is assigned to each neuron and a sample \mathbf{x}_s is randomly selected from the collection of input vectors. Then, to find out the neuron which best represents the selected sample, \mathbf{x}_s is compared with every \mathbf{m}_i value. Euclidean distance, $\|\mathbf{x}_s - \mathbf{m}_i\|$, is an effective measure of similarity between an input vector and a neuron. As a result of the comparison based on euclidean distances, the neuron with the smallest Euclidean distance to the input vector is obtained. We dub this “winner” neuron as the “Best Matching Unit (BMU)” [15]. After finding the BMU, weight vectors are updated due to the following formula [17]:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t) [\mathbf{x}(t) - \mathbf{m}_i(t)],$$

where $\mathbf{m}_i(t)$ is the weight vector \mathbf{m}_i at time t , $\mathbf{x}(t)$ is the input vector at time t and $h_{ci}(t)$ is the neighborhood function at time t . The update operation for $h_{ci}(t)$ is performed by considering the neighborhood around the BMU [17]:

$$h_{ci}(t) = \alpha(t) \exp\left(-\frac{\|\mathbf{r}_c - \mathbf{r}_i\|^2}{2\sigma^2(t)}\right),$$

where

$\alpha(t)$: learning rate factor, $0 < \alpha(t) < 1$,

$\mathbf{r}_c, \mathbf{r}_i$: location vectors of neurons c and i , respectively, where $c, i = 1, 2, \dots, n$,

$\sigma(t)$: width of the neighbourhood function.

The procedures of BMU detection and weight update are repeated for other randomly selected inputs, until a desired number of iterations are completed. Each one of the publications [15] and [10] includes a detailed explanation of building/training a SOM.

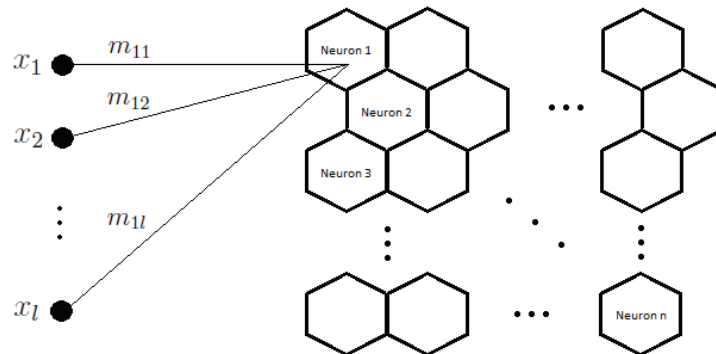


Fig. 1 Configuration of a SOM.

The SOM algorithm is used in many research areas, such as signal processing, speech analysis, data communication, etc. For example, in [27], a new approach has been proposed to define RED parameters in IP networks. [20, 21] are studies where dynamical properties of an IP network traffic is analyzed with the aid of the SOM. In [20], real environment measurements of QoS-related parameters, such as end-to-end delay and packet loss rate are used to train the map; by this way QoS level of the communication could be estimated. Another work [18], introduces an adaptive RED algorithm, KRED, which uses Kohonen neural network model to solve the stability problem in queue lengths. L-RED, which is proposed in [5], is an example study for the usage of neural networks in AQM: a variant of RED with prediction ability is introduced for increasing network utilization rate.

In this work, a SOM based AQM approach is proposed and simulations are performed for testing its performance. This approach depends on the adaptation of RED to the changes in the congestion level by considering the general effects of local congestions on the global congestion. SOM architecture is used to study the tendency of global congestion level and to make future predictions.

3.2 The SOM based approach

Finding a trade off between delay and throughput has been a favorite research area since both schemes are emphasized in determining QoS requirements for IP networks. Minimizing delays/delay variations without compromising throughput is one of the major aims of current queue management algorithms, which basically amounts to finding a trade off between the pros and cons of PQM and AQM methods. [4] presents simulation results for the comparison of two queue management algorithms: DT and RED; [22] explains the advantages of each and foresees positive results in a combination of the two. Our approach, with the addition of the global congestion data determined through the SOM algorithm, is to harness the advantages of both DT and RED. In the routers of an uncongested/low congested network, DT is the better option for prevention of unnecessary early drops, effective use network resources, and maximising throughput and link utilization. Conversely, in cases of high or increasing congestion levels of the network, DT queues suffer

from full queue and lock out problems, and delays/delay variations increase. The SOM based approach, uses the advantages of both schemes to improve RED. We opted to make variations on RED, rather than switching between DT and RED, to make RED sensitive to global congestion notifications.

At the beginning of this study, a centralized observation unit is designed to provide communication with all the routers in an autonomous IP network. IP routers are then specialized for communicating with the observation unit and in order to make necessary updates in their queue management schemes with respect to the congestion notifications. The management of queues in router output interfaces are provided by the SOM based approach. At the beginning, a SOM structure is produced by following the congestion status of all router queues in a traffic scenario case. Equipped with this congestion information, the trained SOM presents the future tendency of global congestion. The information provided by the SOM is then interpreted for making variations on the congestion avoidance algorithms for router queues. An example autonomous network model featuring our SOM based approach takes place is shown in Fig. 2. There are a number of routers providing communication between various source-destination pairs. All routers are directly connected to an observation unit via a centralized router. The details of this network model are given in Section 4. The development of the SOM based approach may be summarized in the following steps (details of Steps 1–6 are explained in [27]):

Step 1: In order to follow the global congestion status of the network, the sum of queue lengths on output interfaces of all routers is observed. A list of sums is produced by using the sum values collected during a period of time,

$$S_k = q_{1k} + q_{2k} + \dots + q_{fk},$$

where S_k is the k -th sample among the list of observed sums, q_{mk} is the k -th sample of the queue length at output interface m ($m = 1, 2, \dots, f$, where f is the total number of output interfaces).

Step 2: Groups of sums are obtained by taking Z consecutive sum values from the list produced in Step 1. A window of size Z is shifted by P ($P < Z$) values each time a group is produced, such that each group is made to have $Z - P$ common values with the previous group.

$$g_i = \{S_{(i-1).P+1}, S_{(i-1).P+2}, \dots, S_{(i-1).P+Z}\},$$

where g_i is the i -th group of sums. For example, if $Z = 4$ and $P = 2$, then the groups will be formed as follows:

$$g_1 = \{S_1, S_2, S_3, S_4\},$$

$$g_2 = \{S_3, S_4, S_5, S_6\},$$

⋮

and so on.

Step 3: Mean, variance, skewness, kurtosis values of each group are obtained to perform a statistical analysis.

Mean of g_i is obtained by using the following equation, where x_{ij} is the j -th element of g_i (where $i = 1, 2, \dots, m$):

$$\mu_i = \frac{1}{Z} \sum_{j=1}^Z x_{ij}.$$

Variance of g_i is obtained by the following equation:

$$\vartheta_i = \frac{1}{Z-1} \sum_{j=1}^Z (x_{ij} - \mu_i)^2.$$

In [9], skewness is described as “an indication of the extent of deviation from symmetry”. Skewness of g_i is obtained by using the following equation:

$$\delta_i = \frac{\frac{1}{Z} \sum_{j=1}^Z (x_{ij} - \mu_i)^3}{\left[\sqrt{\frac{1}{Z} \sum_{j=1}^Z (x_{ij} - \mu_i)^2} \right]^3}.$$

“Kurtosis is the degree of peakedness of a distribution” [9]. Kurtosis of g_i is obtained by the following equation:

$$\kappa_i = \frac{\frac{1}{Z} \sum_{j=1}^Z (x_{ij} - \mu_i)^4}{\left[\sqrt{\frac{1}{Z} \sum_{j=1}^Z (x_{ij} - \mu_i)^2} \right]^4}.$$

After normalization, mean, variance, skewness and kurtosis of a group will be used as the components of an input vector to the SOM, during the training process.

Step 4: A ‘label’ is appended to each input vector, in order to classify the input with respect to the congestion level it represents. In labelling, the mean value is considered as an indicator of global congestion level. Tab. I shows the intervals of $mean_i$ and the corresponding label values utilized in this study.

Step 5: SOM structure is defined and training is performed to obtain the final form of the map.

Step 6: The BMU neurons are investigated to describe the congestion behavior of the network. A neuron is said to be hit when it is chosen as the BMU for an input vector. The BMU neurons hit by input vectors with the same ‘label’s are grouped to build regions, such that each region corresponds to a different congestion status.

Step 7: After training, SOM structure is embedded in the centralized observation unit and it is used to follow the tendency of global congestion. The queue length information received by the observation unit is used to produce input vectors to the SOM. The BMU corresponding to the applied input is identified, and the information about the BMU environment is sent encapsulated in reply packets, from the observation unit to routers. In addition, an active queue management scheme is implemented at each router output interface, so that RED parameters (t_{\max} and $p_{\max_{\text{den}}}$, where $p_{\max_{\text{den}}} = 1/p_{\max}$) are updated with respect to the congestion status of the network. The regions of current and previous BMUs are investigated in each router and a congestion alarm number is generated. If the global congestion

μ_i interval	label
$100 > \mu_i$	0
$200 > \mu_i \geq 100$	1
$300 > \mu_i \geq 200$	2
$400 > \mu_i \geq 300$	3
$500 > \mu_i \geq 400$	4
$600 > \mu_i \geq 500$	5
$700 > \mu_i \geq 600$	6
$800 > \mu_i \geq 700$	7
$900 > \mu_i \geq 800$	8
$\mu_i \geq 900$	9

Tab. I Labels assigned to input vectors with different mean values.

status is low, congestion alarm number is 1 and RED parameters (t_{\max} and $p_{\max_{\text{den}}}$) should be changed for preventing unnecessary early drops. In that case, the queue management scheme will be more like DT. As the global congestion status increases, congestion alarm number moves from 1 to 4 which provides early drops.

What is outlined in these seven steps may be considered as either a preventive approach for possible congestion in future or a curative treatment to eliminate current congestion. The steps, in short, represent an important purpose: avoiding the propagation of congestion, that is to say, minimising the overloaded-queue and packet-loss problems, occurring at a network node, before effecting the neighbouring nodes.

4. Simulations and results

In order to train the SOM deployed in the central observation unit, it is necessary to track the global congestion behavior of an autonomous IP network. The proposed method is described in the following subsection.

4.1 Observation of the variations in the global congestion level

To obtain data on the congestion behavior of a network, it is necessary to refer to the queue lengths on router output interfaces. The network model seen in Fig. 2, is designed with the aid of the OPNET Modeler. The network has five IP routers (node_8, node_9, ..., node_12), four client-server pairs (node_0-node_6, node_1-node_7, node_2-node_4, node_3-node_5) and a centralized observation unit. Clients, servers and the observation unit are connected to the edge routers by '10BaseT' links with a transmission rate of 10Mbps. The links between routers are 'PPP_E1' with a transmission rate of 2.048 Mbps. An ftp traffic is generated between workstations and servers. In OPNET Modeler, FTP Traffic is provided by means of 10 FTP Profiles each of which defines an FTP application (FTP services are supported by the servers named node_4, node_5, node_6, node_7 as

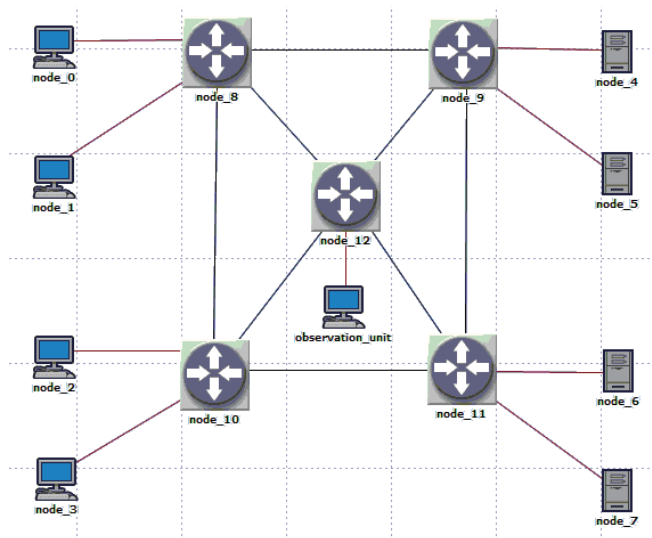


Fig. 2 Network model for simulation scenarios explained in subsections 4.1 and 4.4.

seen in Fig. 2). Tabs. II–IV present the details of the FTP profiles and the FTP applications.

In addition to the ftp traffic, there exists a traffic between routers and the observation unit: IP packets carrying the queue length values are sent to the centralized observation unit. Each of these special packets originates in one of the routers and carries information about the output interface and its queue length. In the observation unit, these packets are received and the queue length values are collected

Name of the profile	Client-Server pair	Start time of the profile		Duration of the profile	
		Distribution type	Mean outcome (seconds)	Distribution type	Mean outcome (seconds)
P1	node_0-node_6	constant	10	constant	20
P5			80		30
newp1			31		48
P2	node_1-node_7	constant	10	constant	20
P6			80		30
newp2			31		48
P7	node_2-node_4	constant	80	constant	30
newp3			31		48
P8	node_3-node_5	constant	80	constant	30
newp4			31		48

Tab. II FTP Profiles used in simulations.

Name of the profile	FTP Application executed within the profile	Repeatability of FTP application in the profile	
		Distribution type of inter-repetition time	Mean outcome for inter-repetition time (seconds)
P1	A1	exponential	300
P5	A5		
newp1	newapp1		
P2	A2		
P6	A6		
newp2	newapp2		
P7	A7		
newp3	newapp3		
P8	A8		
newp4	newapp4		

Tab. III FTP Applications executed within FTP Profiles.

Name of the application	Inter-request time for FTP file transfer		FTP file size	
	Distribution type	Mean outcome (seconds)	Distribution type	Mean outcome (bytes)
A1	constant	0.01	constant	1000
A2		0.01		1000
A5	constant	0.003	constant	1000
A6		0.003		1000
A7		0.003		1000
A8		0.003		1000
newapp1	constant	0.1	constant	1
newapp2		0.1		1
newapp3		0.1		1
newapp4		0.1		1

Tab. IV Properties of FTP file transfer.

(together with the receipt time). DT (FIFO mechanism with disabled RED criteria and a buffer size of 300 packets) is used as the queue management scheme in the routers. Finally, the information received by the observation unit is used to train a SOM. In our network model, there are 25 router output interfaces. Data groups are produced by taking 50 consecutive sum values (each of these values is the sum of queue lengths at 25 interfaces). A window of size 50 ($Z = 50$) values is shifted by 5 ($P = 5$) values each time a group is produced, therefore each group is made to have 45 common values with the previous group. Mean, variance, skewness, kurtosis of

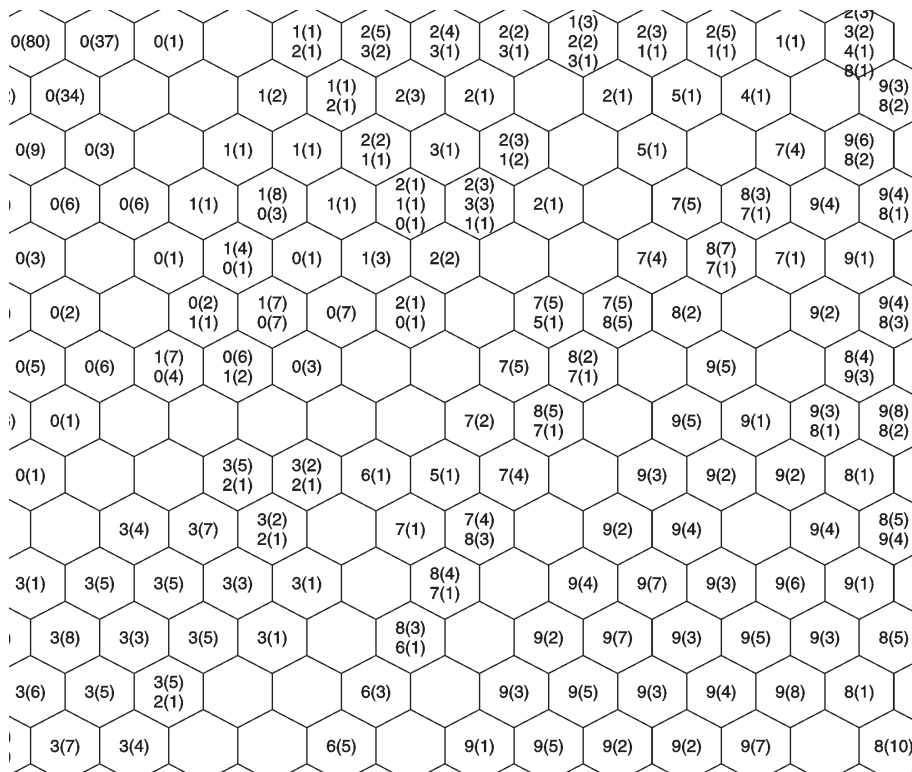


Fig. 3 A magnified part of the trained map with hit and frequency values.

each group are found and normalized due to variance criteria. Normalized values of $\mu_i, \vartheta_i, \delta_i, \kappa_i$ are used to form an input vector, which will be used to train SOM. 'label's are assigned to input vectors, in order to label them with respect to the congestion level they represent.

By the end of this step, data vectors (1×4) are produced to be used as inputs for the SOM. Each input vector has an appended label value which represents the level of global congestion. Input vectors are used during the training process.

Training is performed by the help of the SOM Toolbox for MATLAB [26]. The end-product of the training is a SOM and a codebook matrix. In this work, the map size of the trained SOM is chosen as 20×20 with 400 neurons; the codebook matrix is composed of codebook vectors of size 1×4 . A neuron is said to be hit when it is chosen as the BMU for an input vector. In our proposed method, not only the hit values but also the frequencies of the hits are taken into consideration while studying the congestion behavior. The trained map is shown in Fig. 3, where the information carried by a neuron is presented as follows:

$$\begin{aligned} & \text{hit}_1(f_1) \\ & \text{hit}_2(f_2) \\ & \vdots \\ & \text{hit}_n(f_n). \end{aligned}$$

This representation shows that the neuron is hit for n different labels ($hit_1, hit_2, \dots, hit_n$). The frequency values in paranthesis, f_1, f_2, \dots, f_n , show the number of hits for that label. A weighted averaging technique is used to represent each neuron by a single label value:

$$\frac{(hit_1) \cdot f_1 + (hit_2) \cdot f_2 + \dots + (hit_n) \cdot f_n}{f_1 + f_2 + \dots + f_n}$$

The result of the above equation is rounded up to obtain the average label value of each neuron. The SOM obtained in this subsection will be used in Scenario 3.

The performance of our new queue management approach is verified through a number of simulations. Under each simulation case, three different scenarios are generated: in Scenario 1 and Scenario 2, the performances of DT and RED are tested, respectively; in Scenario 3, the SOM based approach is used for queue management and simulation results are obtained in order to make comparisons with Scenario 1 and 2. The details about these scenarios are explained in the following subsections.

4.2 Scenario 1 (DT)

In this scenario, DT (FIFO mechanism with disabled RED criteria and a buffer size of 300 packets) is used as the queue management scheme in the routers. The network model in Fig. 4 is used for simulation. Unlike Fig. 2, the network model in Fig. 4 does not have an observation unit to collect queue length information from the routers.

4.3 Scenario 2 (RED)

In this scenario, RED is used as the queue management scheme in the routers. The values of RED parameters are as follows:

- $t_{min} = 100$ packets,
- $t_{max} = 200$ packets,
- $p_{max_{den}} = \frac{1}{p_{max}} = 10$,
- $w_q = 2^{-9}$.

As in Scenario 1, the network model in Fig. 4, is used for simulation of Scenario 2.

4.4 Scenario 3 (SOM based approach)

The resulting SOM structure described in subsection 4.1 is used in the simulation of Scenario 3. At specific times, the vector of mean, variance, skewness, kurtosis values is applied as an input to the previously obtained SOM. Euclidean distance of this vector to each SOM codebook vector is calculated. The neuron whose codebook vector is the most similar to the input vector -with the smallest euclidean distance- is the BMU for this input. Fig. 3 shows the hit and frequency values of the neurons on the SOM. A weighted averaging technique is used to represent each neuron by a single label value. Fig. 5 shows the SOM with new label values. For estimating future congestion status, the congestion regions (R1, R2 and R3) shown

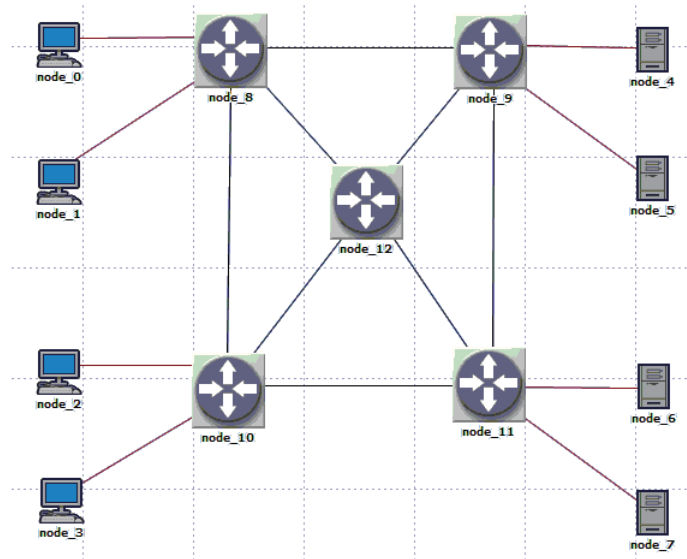


Fig. 4 Network model for Scenarios 1–2.

in Fig. 5 are carefully examined. These regions differ from one another by the level of congestion they represent:

- Neurons with average label values 0 and 1 (dark gray neurons in Fig. 5) are grouped to form region R1. R1 represents the network status with no global congestion problem.
- Neurons with average label values between 2 and 5 (white neurons with label values in Fig. 5) represent R2, which is the region between R1 and R3.
- Neurons with average label values greater than or equal to 6 (light gray neurons in Fig. 5) are grouped to form region R3. R3 represents the status, where the congestion problem is in need of serious attention.
- Neurons with no hit values (empty white neurons in Fig. 5) are ineffective in updating RED parameters. If one of these neurons is chosen as the BMU, RED parameters keep their previous values.

As seen in Fig. 3, there are a few neurons which represent a special case: they are hit not only by inputs with labels less than 6, but also by inputs with labels greater than 6, which means that they represent characteristics of more than one region. When one of these neurons are chosen as the BMU, mean value in the input vector is taken into consideration for updating RED parameters. Scenario 3 provides the estimation of RED parameters. The queue management scheme used in this scenario is a RED variant whose parameters are given in Tab. V, where K stands for *congestion alarm number* and it takes integer values between 1 to 4. The value of K varies proportionally to the global congestion level. If the network

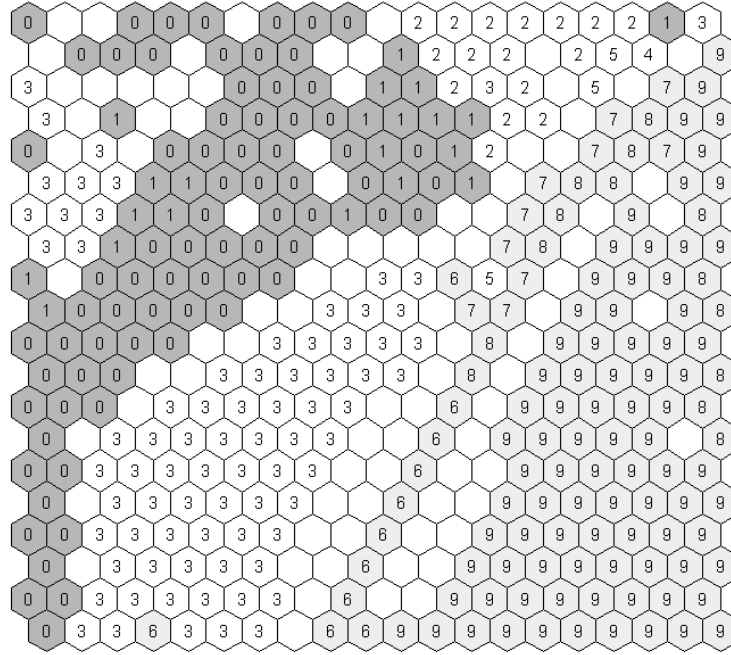


Fig. 5 Self organizing map with regions.

is highly congested (or tends to be so), K becomes 4; as congestion level decreases (or tends to decrease), K also decreases. In Scenario 3, a movement towards region R2 is considered as a sign of increasing/decreasing congestion. For example, a movement from R1 to R2 indicates an increase tendency of the congestion level and a movement from R3 to R2 shows the tendency to decrease. RED parameters are updated by estimating the future global congestion level. As seen in Tab. V, the value of t_{\min} is kept constant while t_{\max} and $p_{\max_{\text{den}}}$ values are changed with

Region of previous BMU	Region of current BMU	K	t_{\min}	t_{\max}	$p_{\max_{\text{den}}}$
R1	R1	1	100	295	1000
R1	R2	2	100	290	1000
R1	R3	4	100	200	10
R2	R1	1	100	295	1000
R2	R2	previous value	100	previous value	previous value
R2	R3	4	100	200	10
R3	R1	1	100	295	1000
R3	R2	3	100	205	10
R3	R3	4	100	200	10

Tab. V BMU regions and RED parameters for Scenario 4.

respect to BMU movements in the SOM. t_{\max} and $p_{\max\text{den}}$ are evaluated due to the following formula:

$$t_{\max} = \begin{cases} 200 + (K_{\max} - K) \cdot s_i, & K > K_{\max}/2 \\ 295 - (K - K_{\min}) \cdot s_d, & K \leq K_{\max}/2 \end{cases},$$

$$p_{\max\text{den}} = \begin{cases} 10, & K > K_{\max}/2 \\ 1000, & K \leq K_{\max}/2 \end{cases},$$

where K_{\max} is the maximum value of the congestion alarm number (as seen in Tab. V, $K_{\max} = 4$); K_{\min} is the minimum value of the congestion alarm number (as seen in Tab. V, $K_{\min} = 1$); s_i is the amount of an increment step in t_{\max} (here, $s_i = 5$); s_d is the amount of a decrement step in t_{\max} (here, $s_d = 5$).

The network model which is used for the simulation of Scenario 3 is shown in Fig. 2. As seen in Tab. V, the greatest values assigned to t_{\max} and $p_{\max\text{den}}$ are 295 and 1000, respectively. These values are used when the BMU exists in R1, meaning low/decreasing congestion, and the queue management scheme is forced to operate like DT. When the current BMU is in region R2, region of the previous BMU is considered in order to determine the tendency of congestion: if the previous region was R1, the congestion level tends to increase, so the value of t_{\max} is decreased to 290 from 295 to encourage packet drops; conversely for R3, where the congestion is likely to decrease, t_{\max} is increased to 205 from 200 to improve the number of queued packets since the congestion is likely to decrease. When the current BMU exists in R3, the congestion level is high/increasing and the smallest values are assigned to t_{\max} and $p_{\max\text{den}}$. These are the values used for the RED parameters in Scenario 2.

A number of simulations have been performed as a part of this study. In this section, the average of the simulation results are presented in Fig. 6–Fig. 11. Each figure has 2 graphs and each graph is composed of 3 curves: the curves obtained for Scenario 1: DT (thin solid line), for Scenario 2: RED (dashed line) and for Scenario 3: SOM (thick solid line). The graphs present the results of various statistics: ftp traffic sent, ftp traffic received, throughput, end-to-end delay, end-to-end delay variation. The reason for plotting the graphs of two global statistics, sent and received traffic, is to observe throughput. In single bottleneck networks, it is easier to capture the throughput of the link. However, the throughput in multi-bottleneck networks, like the one which is modeled in this work, is calculated by the ratio of traffic received to traffic sent through the network. In this work, the figure for throughput results, Fig. 7, is plotted by using the periodically calculated ratio values. Duration of each simulation is 110 sec, where [0–80] sec. is the time interval when the global congestion level is low, while [80–110] sec. is the interval of highly congested traffic. When the performances of RED and DT are compared, it is seen that DT performs better (with less delay/delay variation) under low congestion; while RED is preferable to DT under high congestion.

The results of the SOM based approach show that it is possible to minimize delay/delay variation, under varying congestion conditions. In the interval [0–80] sec., the SOM based approach performs comparably with DT while in the interval [80–110] sec., its performance is as good as RED's. The summation results for end-to-end delays are shown in Tab. VI while the improvement percentages of RED

and the SOM based approach against DT are seen in Tab. VII. The results for delay variations are given in Tab. VIII and Tab. IX. These tables show clearly how the SOM based approach combines the advantages of RED and DT in order to provide a solution for QoS requirements related with delays. Additionally, the improvements in minimizing delays and delay variations do not cause a decrease in throughput, as seen in Fig. 7.

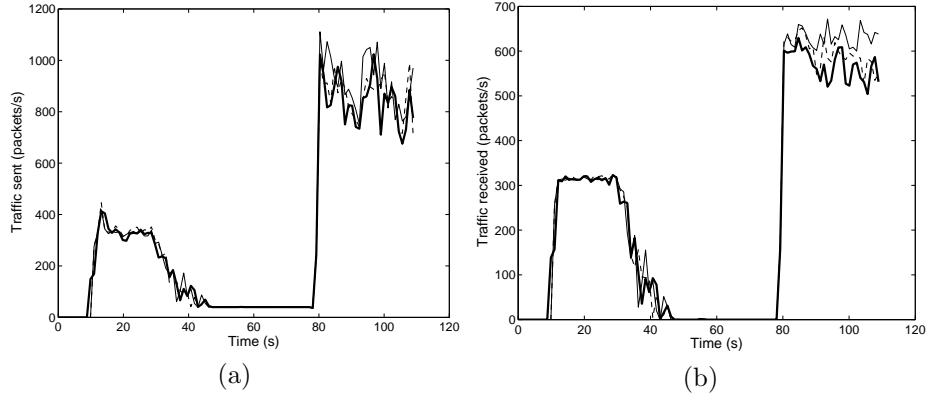


Fig. 6 (a) FTP Traffic sent in the network, (b) FTP Traffic received in the network (Scenario 1 (thin solid line), Scenario 2 (dashed line), Scenario 3 (thick solid line)).

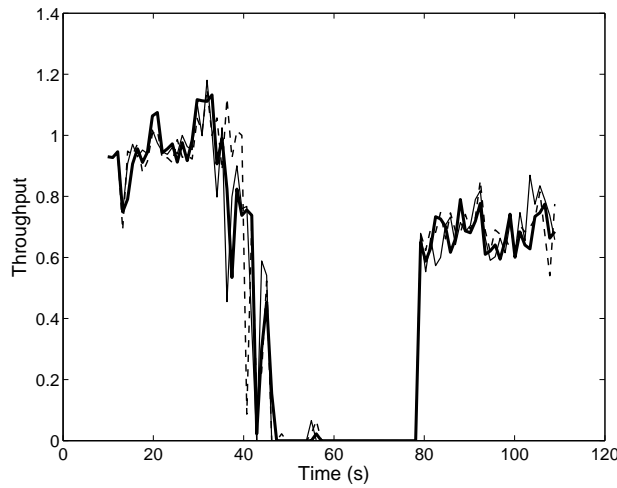


Fig. 7 Throughput (found by calculating the ratio of FTP Traffic received in the network to FTP Traffic sent in the network), (Scenario 1 (thin solid line), Scenario 2 (dashed line), Scenario 3 (thick solid line)).

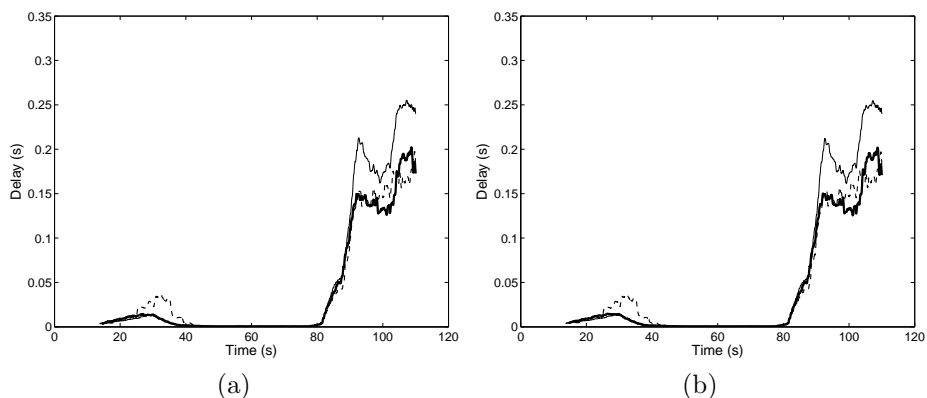


Fig. 8 Moving average of end-to-end delay measured between (a) node0 and node6, (b) node1 and node7 (Scenario 1 (thin solid line), Scenario 2 (dashed line), Scenario 3 (thick solid line)).

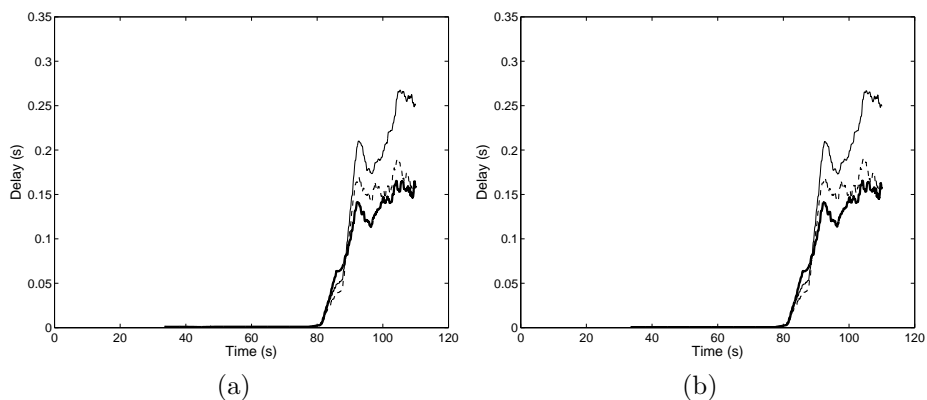


Fig. 9 Moving average of end-to-end delay measured between (a) node2 and node4, (b) node3 and node5 (Scenario 1 (thin solid line), Scenario 2 (dashed line), Scenario 3 (thick solid line)).

Client-Server	Scenario 1 (DT)	Scenario 2 (RED)	Scenario 3 (SOM)
node0-node6	471.6366	387.0474	369.4892
node1-node7	471.5646	388.1737	370.9732
node2-node4	473.3798	359.7726	330.4746
node3-node5	472.2996	360.5875	328.5377
TOTAL	1888.8806	1495.5812	1399.4747

Tab. VI Sum of end-to-end delay values (in seconds) measured between client-server pairs and the overall sum.

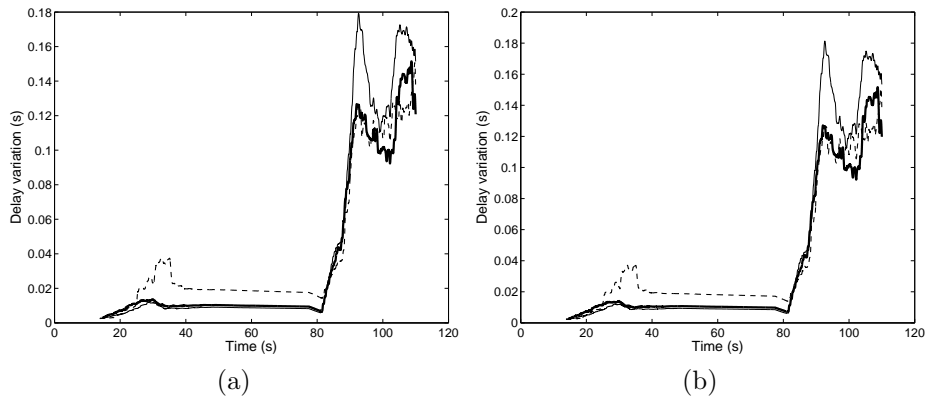


Fig. 10 Moving average of end-to-end delay variation measured between (a) node0 and node6, (b) node1 and node7 (Scenario 1 (thin solid line), Scenario 2 (dashed line), Scenario 3 (thick solid line)).

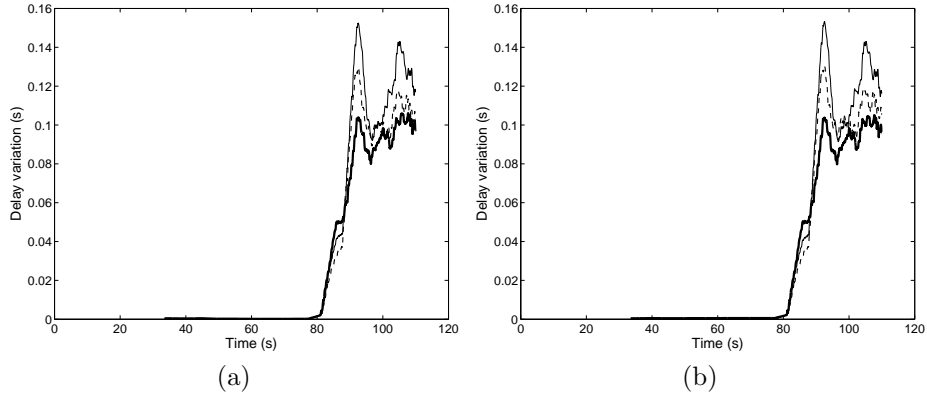


Fig. 11 Moving average of end-to-end delay measured between (a) node2 and node4, (b) node3 and node5 (Scenario 1 (thin solid line), Scenario 2 (dashed line), Scenario 3 (thick solid line)).

Congestion avoidance method	Improvement percentage [%]
RED	20.82
SOM based approach	25.91

Tab. VII Improvement percentages of RED and the SOM based approach, against DT, in end-to-end delays.

5. Conclusion

In the literature of data communication networks, there exist numerous studies on developing congestion avoidance algorithms and queue management schemes. RED and DT are two of the most famous and widely cited queue management ap-

Client-Server	Scenario 1 (DT)	Scenario 2 (RED)	Scenario 3 (SOM)
node0-node6	384.7935	386.6607	333.6449
node1-node7	390.6685	385.4089	337.1229
node2-node4	275.4269	244.7331	226.5750
node3-node5	275.4535	247.7059	227.1142
TOTAL	1326.3424	1264.5086	1124.4570

Tab. VIII Sum of end-to-end delay variation values (in seconds) measured between client/server pairs and the overall sum.

Congestion avoidance method	Improvement percentage [%]
RED	4.66
SOM based approach	15.22

Tab. IX Improvement percentages of RED and the SOM based approach, against DT, in end-to-end delay variations.

proaches. At the beginning of this study, the performances of RED and DT are investigated under dynamically changing congestion conditions in a multi-bottleneck network. From the perspective of minimizing delays/delay variations, the results of comparison show how hard it was to prefer one approach to the other. The performance of RED was inferior to DT, in low/decreasing congestion conditions, owing to its early drop mechanism and static threshold values. Conversely, full queue and lock out problems made DT less effective than RED at minimising delays/delay variations in high/increasing congestion conditions. Development of a new queue management approach is inspired by the advantages/drawbacks of RED/DT implementations in multi-bottleneck networks. In this study, the new queue management scheme is offered by using SOM based approach in order to combine the advantageous features of both RED and DT methods.

In order to follow the variations in the global congestion level of an IP network, a centralized observation unit is presented. Since the queue lengths at router output interfaces are important indicators of congestion, IP routers are specialized to inform the observation unit about the changes in their queue lengths by sending information packets. The information collected by the observation unit is used to train a SOM. After training, SOM structure is embedded in the observation unit, ready to determine the tendency of global congestion level. Besides monitoring global congestion level, the observation unit also notifies the routers about the tendency of congestion behavior. IP routers are improved to accept the notifications from the observation unit and manage their queue lengths with respect to the SOM based approach. OPNET Modeler is used not only for building network models with improved IP routers and the centralized observation unit but also for making simulations. Simulations play an important role in testing the performance of SOM based approach and making comparisons with RED and DT. The performance of the SOM based approach is verified through a number of simulations and the aver-

age of results is presented in this paper. Simulation results show that, SOM based approach presents a considerably improved performance on minimizing end-to-end delays and delay variations. Drawbacks of RED/DT are successfully eliminated without causing a decrease in throughput. Hence, our approach merits attention for providing congestion avoidance in multi-bottleneck IP networks.

The focus of this particular study has been the development and utilization of the SOM based approach in a centralized manner. This is not to imply, however, that the area of interest for this approach is limited to centralized congestion avoidance. The approach may also be adopted for decentralized congestion avoidance, whereby, each router considering the tendency of global congestion may manage its interface queues independently. Decentralized congestion avoidance with the SOM based approach is the topic of a future study, towards which this work is the first and essential step, and its results promising for the success of the SOM based approach.

References

- [1] ATHURALIYA S., LOW S.H., LI V.H., YIN Q. *REM: Active queue management*. IEEE Network Magazine. 2001, 15, pp. 48-53, doi: 10.1109/65.923940.
- [2] ATITUZZAMAN M., ZHENG B. Active Queue Management in TCP/IP Networks. In: M. HASSAN, R. JAIN, ed. *High Performance TCP/IP Networking: Concepts, Issues, and Solutions*. U.S.A.: Pearson Prentice Hall, 2004, pp. 281-307.
- [3] AWEYA J., OUELLETTE M., MONTUNO D.Y. *A control theoretic approach to active queue management*. Computer Networks. 2001, 36, pp. 203-235.
- [4] BONALD T., MAY M., BOLOT J.-C. Analytic Evaluation of RED performance. In: *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM 2000)*, Tel Aviv, Israel: IEEE, 2000, pp. 1415-1424, doi: 10.1109/INFCOM.2000.832539.
- [5] ENHAI L., YAN L., RUIMIN P. An improved random early detection algorithm based on flow prediction. In: *Proceedings of Second International Conference on Intelligent Networks and Intelligent Systems (ICINIS 2009)*, Tianjin, China: IEEE, 2009, pp. 425-428, doi: 10.1109/ICINIS.2009.115.
- [6] FENG W.-C., KANDLUR D.D., SAHA D., SHIN K.G. A self-configuring RED gateway. In: *Proceedings of the Conference on Computer Communications (IEEE INFOCOM 1999)*, New York, U.S.A.: IEEE, 1999, 3, pp. 1320-1328, doi: 10.1109/INFCOM.1999.752150.
- [7] FLOYD S., JACOBSON V. *Random early detection gateways for congestion avoidance*. IEEE/ACM Transactions on Networking. 1993, 1(4), pp. 397-413, doi: 10.1109/90.251892.
- [8] GEVROS P., CROWCROFT J., KIRSTEIN P., BHATTI S. *Congestion control mechanisms and the best effort service model*. IEEE Network Magazine. 2001, 15, pp. 16-26, doi: 10.1109/65.923937.
- [9] HAWKINS C.A., WEBER J.E. *Statistical Analysis: Applications to Business and Economics*. New York: Harper & Row Publishers, 1980.
- [10] HAYKIN S. *Neural Networks: A comprehensive foundation*. U.S.A.: Prentice Hall, 1999.
- [11] HO H.-J., LIN W.-M. AURED – Autonomous random early detection for TCP congestion control. In: *Proceedings of the Third International Conference on Systems and Networks Communications (ICSNC 2008)*, Sliema, Malta: IEEE, 2008, pp. 79-84, doi: 10.1109/ICSNC.2008.22.
- [12] HOLLOT C.V., MISRA V., TOWSLEY D., GONG W.-B. *Analysis and design of controllers for AQM routers supporting TCP flows*. IEEE Transactions on Automatic Control. 2002, 47, pp. 945-959, doi: 10.1109/TAC.2002.1008360.

- [13] HOLLOT C.V., MISRA V., TOWSLEY D., GONG W.-B. On designing improved controllers for AQM routers supporting TCP flows. In: *Proceeding of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM 2001)*, Anchorage (Alaska), U.S.A.: IEEE, 2001, 3, pp. 1726-1734, doi: 10.1109/INFCOM.2001.916670.
- [14] INTERNET ENGINEERING TASK FORCE (IETF). RFC 2309: Recommendations on queue management and congestion avoidance in the Internet [online]. Written by BRADEN B., et al. April 1998. Available from: <https://tools.ietf.org/html/rfc2309>.
- [15] KOHONEN T. The Self-Organizing Map. In: *Proceedings of the IEEE*, 1990, 78(9), pp. 1464-1480, doi: 10.1109/5.58325.
- [16] KOHONEN T. Things you haven't heard about the self-organizing map. In: *Proceedings of the IEEE International Conference on Neural Networks 1993*, San Francisco, CA: IEEE, 1993, 3, pp. 1147-1156, doi: 10.1109/ICNN.1993.298719.
- [17] KOHONEN T. *The self-organizing map*. Neurocomputing. 1998, 21, pp. 1-6.
- [18] LOCHIN E., TALAVERA B. Managing network congestion with a Kohonen-based RED queue. In: *Proceedings of IEEE International Conference on Communications (ICC 2008)*, Beijing, China: IEEE, 2008, pp. 5586-5590, doi: 10.1109/ICC.2008.1047.
- [19] MANASA S. *ANLRED: A Robust AQM Mechanism for Congestion Avoidance*. International Journal of Computer Applications. 2013, 81(15), pp. 1-9, doi: 10.5120/14196-2259.
- [20] MASUGI M. QoS mapping of VoIP communication using self-organizing neural network. In: *IEEE workshop on IP operations and management (IPOM 2002)*, 2002, Dallas, Texas, U.S.A.: IEEE, pp. 13-17, doi: 10.1109/IPOM.2002.1045749.
- [21] MASUGI M., TAKUMA T. *Multi-fractal analysis of IP network traffic for assessing time variations in scaling properties*. Physica D. 2007, 225, pp. 119-126, doi: 10.1016/j.physd.2006.10.015.
- [22] PATEL S., GUPTA P., SINGH G. Performance Measure of Drop Tail and RED Algorithm. In: *Proceedings of International Conference on Electronic Computer Technology (ICECT 2010)*, Kuala Lumpur, Malaysia: IEEE, 2010, pp. 35-38, doi: 10.1109/ICECTECH.2010.5479996.
- [23] QUET P.-F., ÖZBAY H. *On the design of AQM supporting TCP flows using robust control theory*. IEEE Transactions on Automatic Control. 2004, 49(6), pp. 1031-1036, doi: 10.1109/TAC.2004.829643.
- [24] ROUHANI M., TANHATALAB M.R., SHOKOHI-ROSTAMI A. Nonlinear Neural Network Congestion Control Based on Genetic Algorithm for TCP/IP Networks. In: *Proceedings of Second International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN 2010)*, Liverpool, United Kingdom: IEEE, 2010, pp. 1-6, doi: 10.1109/CICSyN.2010.21.
- [25] RYU S., RUMP C., QIAO C. *Advances in Internet congestion control*. IEEE Communications Surveys. 2003, 5(1), pp. 28-39, doi: 10.1109/COMST.2003.5342228.
- [26] SOM TOOLBOX TEAM. SOM Toolbox 2.0 [software]. [accessed 2002-10-1]. Available from: <http://www.cis.hut.fi/projects/somtoolbox>.
- [27] YELBAŞI Ö. *Self Organizing Map Based RED Parameter Estimation for Congestion Avoidance*. Türkiye, 2011. PhD thesis, Anadolu University. Available from: <https://tez.yok.gov.tr/UlusalTezMerkezi/tarama.jsp>.