A Hybrid Neural Network Approach for Congestion Control in TCP/IP Networks

Samira Razmara

Electrical Engineering Department

Islamic Azad University Tehran Central Branch

Tehran,Iran

razmara.samira63@gmail.com

Roohollah Barzamini[†]

Electrical Engineering Department

Islamic Azad University Tehran Central Branch

Tehran,Iran

Barzamini@aut.ac.ir

AlirezaIzadi

Electrical Engineering Department

Islamic Azad University Tehran Central Branch

Tehran,Iran

Ali.izadi.eng@iauctb.ac.ir

Negar Janpors

Computer Engineering Department

Islamic Azad University South Tehran Branch

Tehran,Iran

janpors@alborz.kntu.ac.ir

ABSTRACT

In this article, a hybrid approach and model for congestion control in TCP/IP networks using improved neural networks and genetic algorithms is presented. In fact, the primary network traffic flow obtained by monitoring the router buffer has been investigated and the effective parameters have been identified and selected from the remote point of view with the help of the Arma model time series models. The selected model has been used to detect the threshold to increase and decrease the router buffer, Integrated data free of noise and redundancy was presented as input to the neural network algorithm, and at the same time, the genetic algorithm was used in case of crowding, The genetic algorithm has been used to improve the cold start challenge and the transmission rate and the congestion rate, and the neural network algorithm has been developed by relying on the four components of the input rate, the service rate, the percentage of the empty queue and its total in two steps of learning and testing. The proposed method has been investigated from the point of view of root mean square error, the average absolute value of error percentage, coefficient of determination in diagnosis, and correlation error with the help of an applicable and dynamic model. The results obtained with the help of simulations performed in MATLAB and Rapidminer tools show the improvement process compared to RED and DRL-AQM methods.

Keywords:-Congestion control, genetic algorithm, neural network, TCP/IP network

1 Introduction

Communication networks use two types of Transmission control protocol (TCP) and User datagram protocol (UDP) in their transmission layer, due to the increasing use of communication networks and the expansion of information transmitted through the Internet, congestion control in communication networks. It is necessary, that the occurrence of congestion in the network causes the loss of some packets and undesirable transmission delay, as well as the loss of bandwidth, and as a result, reduces the efficiency of the network [4]. When the total input rate to the link exceeds its capacity, the packets are stored in a buffer. If the buffer length is limited, the continuation of such a process will cause insufficient space to store the packets and therefore the incoming packets will be discarded. It is in such a situation that congestion occurs due to buffer saturation. Congestion control on the Internet includes TCP algorithms and Active Queue Management (AQM) algorithms. TCP algorithms are implemented in end hosts and AQM algorithms are implemented in routers to increase network efficiency and reduce queue delay. Since the pattern governing the traffic of computer networks is a non-linear pattern. Neural networks, as an advanced and widely used tool that has the ability to process massive amounts of data in parallel, have been widely used in the identification, prediction and optimization of devices in various fields of science [1,5].

In a communication network with TCP protocol, before the data transfer takes place, a TCP connection is established to ensure the correct data transfer. To achieve this goal, a confirmation message is used. If this confirmation message is returned to the source late or has an error, the packet is retransmitted. If the communication channel has noise or disturbance, it may take several transmissions for a packet to be transmitted correctly. Also, to ensure the correctness of the destination, the TCP protocol uses the Header for the sent packets, which contains both the source address and the destination address. A time-out mechanism is used to control congestion in TCP [2]. Therefore, the TCP protocol has a congestion control mechanism, but congestion is detected after its occurrence. Various works have been done by researchers regarding the investigation of congestion control [4-9], and it has also been proven that congestion management algorithms are different from the aspects of diagnosis and control [3]. Congestion detection is an important initial step in any congestion control algorithm [2]. Modern computer networks, including the Internet, are designed for the rapid transfer of a large amount of information in such a way that congestion control algorithms are of great importance to them. Some of the algorithms used by researchers are: Drop Tail (DT) and DEC bit. Algorithm and Random Early Detection (RED), as well as some of the controllers used are robust control, optimal control, adaptive control and Fuzzy control, as well as congestion control methods in terms of the type of feedback received are [10]: Timeout-Based Congestion Control, ECN-Based Congestion Avoidance (ECN stands for: Explicit Congestion Notification) and Delay-Based Congestion Avoidance.

In this article, Multi-Layer Perceptron (MLP) neural network has been used to extract the governing pattern of network traffic and predict future traffic values, so that the buffer of network routers is monitored first to extract network traffic as a time series. Then, the time series is used to train the neural network with properly selected parameters so that the neural network implements it in the network router.

Neural network can be used to predict the behavior of time series and learn the pattern governing computer network traffic. In fact, to increase the speed and accuracy of detecting and predicting congestion and implementing the DBRED mechanism, we have used a neural network. DBRED relies on active queue management mechanisms, the most common of which is RED. The suggestion of this article is that instead of DBRED making a decision based on a fixed value of the threshold limit, it should estimate it dynamically with the help of a neural network. With this, the BRED mechanism will be replaced with a mechanism that has a wider horizon and will be more complex at the same time. In fact, our attempt in this article is to implement a suitable neural network to estimate threshold limits for queue management algorithms. For this purpose, first we have designed the appropriate neural network and trained it by the time series of traffic data, we can implement the trained neural network in the input of the routers so that based on the current value and the previous values of the traffic rate of the input to the router, about Determine the appropriate threshold for receiving or not receiving packages.

The objectives of this article:

1) Congestion control to increase the quality of service and efficiency of the TCP/IP model

2) Time series extraction based on router buffer data

3) Congestion control and management with the benefit of neural network algorithm

4) Solving the cold start problem by covering the initial time delay with the help of genetic algorithm

5) Providing an applicable and dynamic model by reducing cost, response time and avoiding other expenses

Figure 1 shows an example of a congestion control structure. The strategy of avoiding crowding is based on the detection of early signs, we prevent the collapse of the crowd by taking preventive measures. For example, TCP-Vegas tries to detect congestion in the early stages and by comparing the measured power with the expected power in the transmitter, it detects congestion [1].

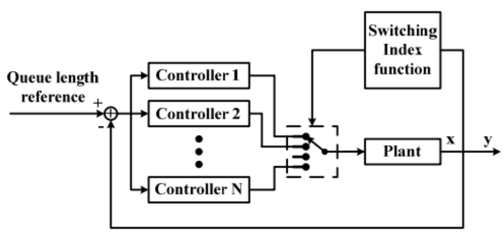


Figure 1:An example of a congestion control structure.

2 Model

In this method, first the network traffic flow is given to the model as input and primary data, these raw and primary data from the network flow are converted into a time series with the help of linear equations and the packet bypasses are also removed. In the next step, to allocate packets that include network flows to processing resources, the buffer status of routers and important indicators such as input rate, service rate, percentage of the empty queue and percentage of the entire queue are checked. First, we use neural network techniques and genetic algorithms in parallel for dynamic allocation and transfer of data to processing resources and queue balance. In this way, we give the time series flow as an input to the neural network and we use the genetic algorithm according to the time delay in training and solving the cold start problem. After detecting and applying the reduction or increase of the threshold limit, the dynamic balance queue is performed and the routers perform the requested allocations with the help of this resulting dynamic state. This process is performed for all resources individually and in a local approach. Accept and in a higher view, all local situations are centrally controlled and managed to avoid congestion, in fact, network traffic management is done dynamically with optimal efficiency.

In the BRED algorithm, parameters L1, L2, P1 and P2 are updated only when a new flow is added to the network, while in the modified algorithm, a default value is first considered for these parameters, then with Paying attention to the behavior of the currents, the values are updated. Flow behavior means the average rate of sending packets and the ratio of service rate to data input rate. In BRED algorithm, relations 1 to 5 are used to calculate the probability of discarding packets.

$$P_2 = \frac{\sqrt{N}}{\sqrt{N}+10} \quad (1)$$

$$P_1 = \frac{P_2}{10} \quad (2)$$

$$L_u^i = \alpha W_m \quad (3)$$

$$L_I^i = \beta L_u^i \quad (4)$$

$$P_{drop}^i = \begin{cases} 0 & qlen_i < L_1^i \\ P_I^i = \frac{\alpha}{L_u^i - L_I^i}(qlen_i - L_I^i) & \\ P_u^i = \frac{1-\alpha}{W_m - L_u^i}(qlen_i - L_u^i) & \\ 1 & W_{m<}qlen_i \end{cases} \quad (5)$$

By default, α=0.7 and β=0.6, but over time (with the help of time series or the use of Tapped Delayed Line (TDL) neural network, α values are dynamically calculated).

In the used perceptron model, 200 data were used as a dataset for training, which was collected from the 10th to the 14th moment of networks with different conditions and topologies and normalized based on the relation (6). Also, the hyperbolic tangent function is used as the activation function for the middle layer and the linear function is used as the output activation function. The structure of the delayed inputs of the neural network used can be seen in Figure2.

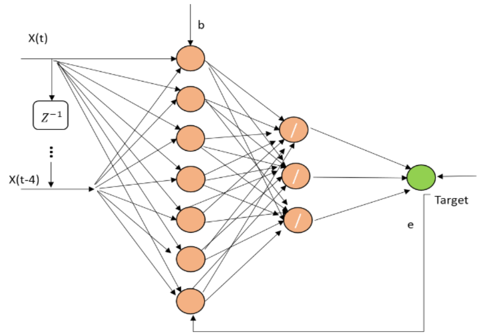


Figure 2:Structure of neural network delayed inputs.

$$norm_{data} = \frac{data - min_{data}}{max_{data} - min_{data}} \quad (6)$$

To convert the output of the neural network to the value of 1, 0 and -1 according to the equation (7), a step function is used at the end of the work:

$$output = \begin{cases} 1 & 0.2 < out_{NN} \\ 0 & -0.2 \leq out_{NN} \leq 0.2 \\ -1 & out_{NN} < 0.2 \end{cases} \quad (7)$$

• If the output value of the neural network is greater than 0.2, the output of the step function becomes one, which means that the input rate of the corresponding stream is lower than the service rate or the service rate and the input rate is equal to each other. near and the empty space of the entire buffer is more than 80%, so the data reception threshold can be increased by one step (0.1).

• If the output value is between 0.2 and -0.2, the output of the step function will be zero, which means that it is not necessary to change the value of the thresholds.

• If the output is less than -0.2, the step function returns the value -1, which means that according to the input data, the service ratio is lower than the input rate or there is little space in the queue. It is empty, so the probability of congestion is higher and the thresholds should be reduced by one step.

In this model, we can identify bad behavior flows that send data aggressively and by limiting them, we can have a suitable mechanism to prevent and control congestion. Finally, for the last layer, we have used three nodes, each of which represents one of the outputs (-1, 0, 1), the output that has the highest probability is selected as the desired output. Figure 3 shows the general architecture of an active queue management system based on a delayed neural network. We have used the RR algorithm to send packets, and the packets related to different streams are served in a rotating manner. The number of packets that the router is able to serve in each time cycle is obtained from the relationship C=U/T, where U is the queue processing rate in bits per second and T is the average size of each packet. It is by a bit. Therefore, the service rate for each flow is equal to μ=C/N.

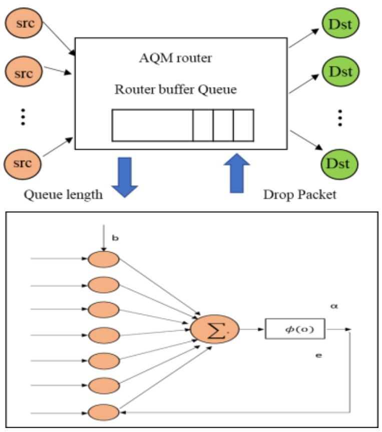


Figure 3:AQM system architecture

3 Methods

In order to solve the cold start challenge, each gene refers to a request number, and the value of each gene is assigned to the source number for the request. The primary population is a set of randomly generated chromosomes. Each gene of the chromosome with a value of 1 means the source and a value of zero means the normal state and -1 is an inactive node [12]. This population was presented as a dataset to the genetic algorithm in order to start the work. In this method, the average length of the queues is calculated first, and then, randomly, a number of requests will be selected as bypasses from the queues that are more crowded than the average. Requests will be selected as bypass candidates that have higher queue congestion than the average queue of all requests. By using this method, the genetic algorithm moves toward the global optimal solution and the selection of the appropriate resource with a higher speed.

$$I_j = \begin{cases} 1\ ifE\left(Req_j\right) > 0\ andreq_j = resource \\ 0\ ifE\left(Req_j\right) > 0\ andReq_j = non-resource\forall_j \in \{1.2.3.\ldots.N\} \\ -1\ ifE\left(Req_j\right) = 0 \end{cases} \quad (8)$$

First, each router guesses a random number between zero and one. If the value of the guessed random number is less than $TH_i$and the requested request is also a member of the candidate set of CanResource, the gene related to that request will be equal to one in the chromosome of random answers, otherwise, the requested gene Comment is considered equal to zero. How to calculate $TH_i$is shown in relation 9.

$$TH_i = \begin{cases} \frac{E_i}{\sum_{j=1}^{N} E_j} if i \in CanResource \\ 0 \quad otherwise \end{cases} \quad (9)$$

The CanResource set refers to the set of resources whose current queue length is greater than the average of the crowd of all living resources. E_i refers to the current congestion of the i-th queue and N refers to the total number of live sources in the current stage.

Since the data is sent in the data transmission phase, there is a possibility of congestion in this phase. The proposed model controls the congestion by controlling the data transmission rate of the transmitters. Equation (10) shows the optimization problem for sending rate control:

$$MinF = \alpha\, [\sum_{i=1}^{n}(1 - S_i/1 + S_i)\ .\ P_i] + (1-\alpha)S_c \quad (10)$$

$$S_1 + S_2 + \cdots + S_n + S_c = 1 , \forall i\, .\, 0 \le S_i \le 1 , 0 \le S_c \le 1 , 0 \le \alpha \le 1$$

The variables $S_1 + S_2 + \cdots + S_n$ are the share of resources of an intermediate router with n neighbors. Each of the si's that are calculated in the current router and sent to the neighbors are known as S_C^{NH} in the neighboring router [11]. The optimization function in relation 11 specifies the degree of congestion in the current router and QL/n indicates the maximum length of the virtual queue for the ith neighbor.

$$vq_1(t) + vq_2(t) + \cdots + vq_n(t) = QL \quad (11)$$

$$\sum_{i=1}^{n} S_i \cdot \left(q_i \cdot n /_{QL}\right) < S_C^{NH} \quad (12)$$

$$S_i^{new} = YS_i^{new} - (1-y)S_i^{old} \quad (13)$$

The contribution amount is in the interval $0 \le S_c \le 1$. The proposed model uses a flexible procedure for queue management. The rows are virtual and the border between them is not fixed; It means that if one of the virtual queues has an empty space, the flows related to other neighbors whose virtual queue is filled can use the empty virtual queue. We have used the roulette cycle method to select chromosomes. In this relationship, N is equal to the number of the initial population of chromosomes and F_i is equal to the value of the fitting function of each chromosome. The first chromosome whose random number value is less than the cumulative probability of that chromosome is known as the target chromosome.

$$P_i = \frac{{}^{1}/_{F_i}}{\sum_{j=1}^{N} {}^{1}/_{F_i}} \qquad (14)$$

$$C_i = \sum_{j=1}^{i} P_J \qquad (15)$$

The role of mutation in the genetic algorithm is to restore the lost or unfound genetic material inside the population, to prevent the premature convergence of the algorithm to local optimal solutions. In the genetic algorithm, we perform mutation with a very low probability (less than 0.05). The mutation used in the proposed algorithm randomly selects the target gene and performs zero to one mutation and vice versa. In this algorithm, the router starts sending data by receiving side information such as input rate, service rate, percentage of empty queue, empty percentage of the entire queue from the monitoring done. Then, the role of each router is determined and the router informs the source through a control packet. Based on the TDMA protocol, routers activate the radio part only when sending data, and the hardware part of the router is turned off at other times. Using the TDMA protocol, in addition to reducing the energy consumption, also helps to reduce the collision of data packets.

4 Simulation

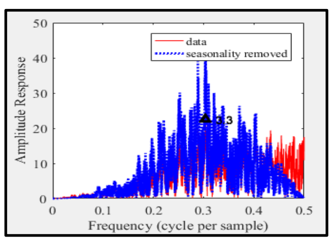
In order to evaluate the used model, we have used an evaluation approach from the point of view of accuracy, correctness and recovery to manage congestion in the network and the results of the investigations showed that the presented model leads to cost reduction and prevents time wastage. It is possible to spend the least possible cost and time on reforms in order to improve and increase efficiency. The evaluation relationships are based on the defined parameters as described in Table 1.

Table 1: Evaluation relationships based on defined parameters

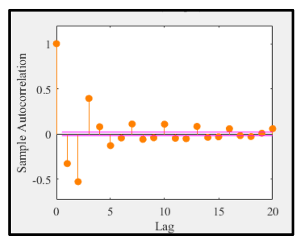
How to evaluate	Description	Name of the component
$Accuracy = \frac{\text{TP} + \text{TN}}{N}$	Ratio of correct detections to total data TP+TN	accuracy
$Recall = \frac{\text{TP}}{TP + FN}$	Number of identified positive samples to total positive samples	sensitivity
$Precision = \frac{\text{TP}}{TP + FP}$	The number of true positive diagnostic samples to the total number of declared positive samples	correctness

After preparing the initial dataset according to time series techniques, we select a model to perform control learning and prediction with the help of neural network. In the presented model, in order to select the model to calculate the closed bypass and provide the input data to the neural network, various situations are examined and evaluated. In this section, to select the model, measure specific values of the situations defined as (Moving average) MA (1), MA (2), (Seasonal Autoregressive) SAR (1), SAR (2), (Autoregressive moving average) model) ARMA (1,1) ARMA(1,2), ARMA(2,1), ARMA(2,2), (Autoregressive integrated moving average) ARIMA (1,1,1), ARIMA(1,2, 1), ARIMA(2,1,1), ARIMA(2,2,1), ARIMA(1,1,2), ARIMA(1,2,2) is performed. The presented model has been developed with the help of RapidMiner and MATLAB tools.

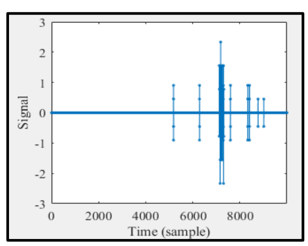
Among the analyzed models, ARMA(2,1) is more suitable for data generation and is chosen as the final model. The results of this model are shown in Figure 4.



According to the frequency and amplitude of the response



In terms of partial correlation error



according to time and signal change

Figure 4:The results of ARMA(2,1)

The basic assumptions and data of the genetic algorithm simulation for the cold start challenge can be seen in Table 2.

Table 2: Assumptions and initial data of genetic algorithm simulation for the cold start challenge

value	Description
Based on the remaining capacity of the queues	Initial population generation algorithm
Uniform and two points	link algorithm
100npop	number of routes (population)
Roulette cycle	Selection Algorithm (link)
2 (sending rate and congestion rate)	Number of invoices
0/3	Closed bypass percentage
pc=0.7	Link percentage
ncross=2*round(npop*pc/2)	Number of children for transplant
pm=0.3	Composition percentage
nmut=round(npop*pm)	Number of children to combine
maxiter	The number of iterations of the algorithm

We used the following three methods for the final evaluation of our model:

1- The number of sensors is equal to 50 and the number of epochs is equal to 100

2- The number of sensors is equal to 100 and the number of epochs is equal to 150

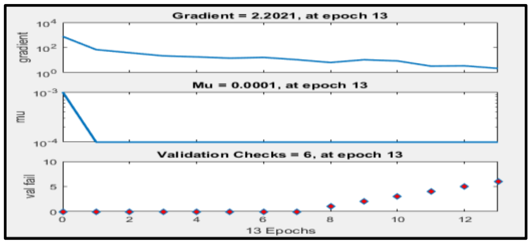
3- The number of sensors is equal to 150 and the number of epochs is equal to 200

The average results can be seen in table 3.

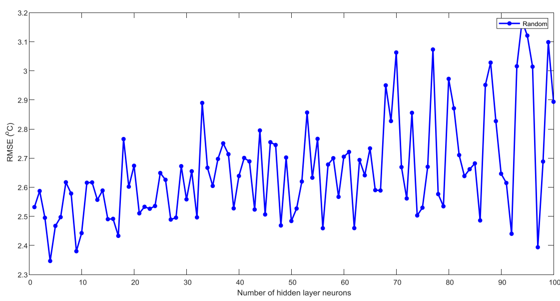
Table 3: Average total results of three different scenarios

Improved congestion rate	Improved delivery rate	epochs
0.6071813	0.0001840077	100
0.529785667	0.0000451082	150
0.52766965	0.0000222910	200

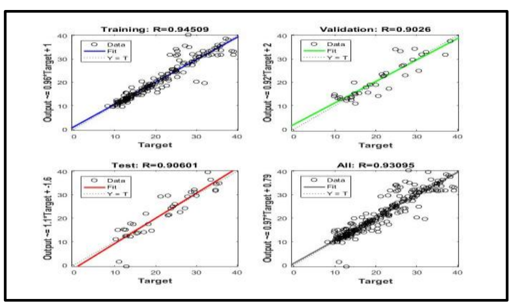
It can be said that as the number of repetitions increased, the sending rate has been improved, in other words, according to the congestion rate, this optimization and balance has been done, and the congestion rate has decreased. Figure 5 shows the results of the error and quality check.



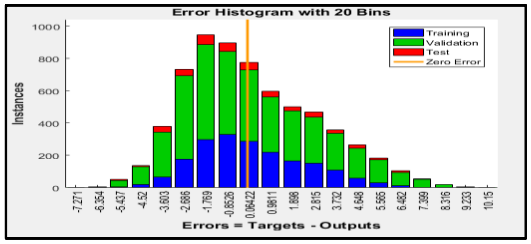
The diagram of the amount of errors according to the stopping conditions in optimization



The sum of the square roots of the error of each diagnosis according to the number of layers in the neural network



Comparison of the quality of neural network and real with adaptation on the target



The amount of belonging of each category of data for different errors

Figure 5:The results of error and quality check in the results

In order to check the performance of the model of this article, the results obtained were compared with RED, DRL-AQM methods, the results can be seen in Figure 6. From the point of view of the delay in the queue according to the number of epochs, it can be said that the accuracy of the allocation in the proposed model is optimal and the performance is favorable compared to similar methods.

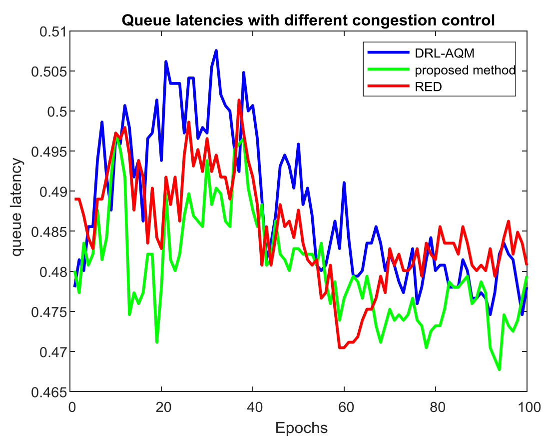


Figure 6:Comparison of three models of queuing observation according to the number of epochs.

The comparative results of the recommended model with the RED and DRL-AQM methods can be seen in Table 4.

Table 4: Comparing the results of the proposed model with other similar methods.

Algorithm	Assigned	Accuracy	sensitivity	Precision
RED	k-fold =5	71.63	78.56	76.42
DRL-AQM		89.45	85.81	84.97
proposed model		95.4	85.77	96.68

5 Conclusion

In the control design presented in this article, with the help of a hybrid approach for congestion management, dynamic decision-making was used to reduce or increase the threshold in such a way that it was monitored in the buffer of the routers and the network traffic flow was obtained, which was beneficial. An initial network traffic data was collected from the previous datasets, and in the next step, pre-processing was done by analyzing and checking these data, and in the next step, with the help of time series models, packet bypassing and extraction of Necessary knowledge were carried out so that a uniform and integrated dataset can be used as a training model in the neural network algorithm. At this stage, in parallel, if there is crowding and estimation, the two techniques of genetics and neural network are executed simultaneously until the time Training and coverage of the cold start problem using genetic algorithm is finally done by controlling the threshold and assigning the right path for the requests for processing to the congestion management resources in a dynamic way. Finally, the presented model was evaluated and measured from the point of view of precision, accuracy and recovery parameters with some other similar methods in the form of an applicable model. The obtained results also indicate the desirability of the presented solution compared to other similar methods.

REFERENCES

[1] MaXiaoping, L., Deng, L., (2020), “Congestion tracking control for multi-router TCP/AQM network based on integral backstepping”, Computer Networks, Vol 175, pp. 231-248.

[2]LiXiangyong, Z., Qiu,C., (2020), “TCP/AWM network congestion algorithm with funnel control and arbitrary setting time”, Applied Mathematics and Computation, Vol 385, pp. 504-518.

[3] Demeke, N., Lakew, D., Cho, S., (2019), "Congestion control vs. link failure: TCP behavior in mmWave connected vehicular networks", Future Generation Computer Systems, Vol 101, pp. 1213-1222.

[4] Bansal, D., Balakrishnan, H., (2016), "Binomial Congestion Control Algorithms",in IEEE INFOCOM, Vol 5, pp. 504-512.

[5] Beivens, J.Alan, Szymanski, B., Embrechts, M., (2019), "Network Congestion Arbitration and Source Problem Prediction Using Neural Networks", Smart Engineering System Design, Vol 7, pp. 112-126.

[6] Barzamini, R., Shafiee, M., Dadlani, A., (2012), "Adaptive generalized minimum variance congestion controller for dynamic TCP/AQM networks", Computer Communications, Vol 2, pp.170-178.

[7] Barzamini, R., Shafiee, M., (2011), "A new sliding mode controller for TCP congestion control",201119 thTelecommunications Forum (TELFOR) Proceedings of Papers, Vol 5, pp.214-217.

[8] Barzamini, R., Shafiee, M., (2010), "LMI based switching congestion controller for multiple bottleneck packet switching networks", Journal of American Science, Vol 7, pp.254-261.

[9] Barzamini, R., Shafiee, M., (2011), "Congestion control of differentiated services networks by sliding mode control", Elektronika ir Elektrotechnika, Vol 19, pp. 83-86.

[10] Balador, A., Cinque, E., Mohammadi, M., (2021), "Survey on decentralized congestion control methods for vehicular communication", Vehicular Communications, Vol 33, Article 100394.

[11] Rosberg, Z., Matthews, J., Zukerman, M., (2010), "A network rate management protocol with TCP congestion control and fairness for all", Computer Networks, Vol 54, Issue 9, pp. 1358-1374.

[12] Kumar, R., Singh, N., Piyush, P., (2020), "Genetic CoCoA++: Genetic Algorithm based Congestion Control in CoAP", International Conference on Intelligent Computing and Control Systems (ICICCS), Vol 15 ,pp. 321-335.