CONGESTION CONTROL IN COMPUTER NETWORKS USING HYBRID INVASIVE WEEDS OPTIMIZATION ALGORITHM AND K-NEAREST NEIGHBOR

Reza Nadi[1], Amir Massoud Bidgoli[2*]

[1]Islamic Azad University-Tehran North Branch
[2]Assistant Professor, B.Sc., M.Sc., Ph.D. (Manchester University), MIEEE, Islamic Azad University-Tehran North Branch

ABSTRACT

One of the main components of TCP protocol is its congestion control structure. TCP protocol is the most commonly used and popular transfer protocol. When the total amount of network data exceeds the network capacity, there will be congestion problem in this protocol. In computer networks, especially in networks with many nodes, there is a potential problem of packet sending which sometimes results in data loss due to the exchange of large volumes of data. Queue overflow can be used to detect congestion. If the number of packets in a queue is more than the threshold value, there would be the probability of congestion. The more close packets are to the threshold, the more probable is the congestion. In this paper, the hybrid invasive weeds optimization algorithm (IWO) and K-Nearest Neighbor (KNN) have been used to control the congestion. The IWO algorithm is one of the meta-innovative algorithms used to form vectors and each vector contains the packet's size. Also, KNN model is used to classify the packets and to prioritize them. The results of simulation in the NS2 environment have shown that in comparison to Cuckoo model, the proposed model implementing at different times has been able to send more packets.

KEYWORDS:

Congestion Control, Invasive Weeds Optimization Algorithm, K-Nearest Neighbor

1. INTRODUCTION

Congestion in computer networks increases the queue length in nodes, packet loss, and end to end latency, and in general, reduces productivity. Congestion control in computer networks means to prevent accumulation of excessive data in the waiting queue and then packets are not lost. Congestion occurs when the amount of input data in data bus is more than its tolerance. As a result, lack of coordination and proportion between the rate of packet sending in the sender and network resources (the capacity of middle rows and speed of the middle nodes and the receiver in transmitting and processing the data) is one reason for congestion. Congestion in the network increases the queue length in middle nodes, and the result is packet loss, end-to-end latency, and low productivity. To control congestion, sending rate must be controlled or the resources should be more available. In secured communications, services such as congestion control, active buffer monitoring, and retrieving the lost packets are necessary to ensure secured packet delivery.

One of the main issues in computer networks is congestion. Congestion in the network occurs when the number of packets sent to the network exceeds its capacity. A set of mechanisms to inhibit congestion and to keep network load on a value less than network capacity is called congestion control [1]. Congestion control in computer networks means preventing excessive accumulation of data in the waiting queue and then, there will not be packet loss [2]. The reason for congestion in the network is the presence of queue in routers and intermediate switches of the network. The switching tools in the network have queues that store the packets before and after processing. For example, a router has an input/output queue for each of the other interfaces. If the packet arrival rate is higher than processing rate, the input queues will be longer. Also, if the exit rate is lower than processing rate, the output queues will be longer. In both cases, there will be congestion in the network.

When a large number of packets are sent to a part of the network, its efficiency will be reduced. If the number of packets sent by the host machine to the subnet is proportional to the network's capacity, all the packets will be delivered to their destination [3]. But as traffic increases excessively, the routers will not be able to rout and the packets will start to be lost. The reasons for the congestion are as follow:

- **Transmission of multiple input packets from an outgoing line:** if there are multiple lines of packets entering into one outgoing line concurrently, there will be a queue of information. When there is not enough memory to store this information, the packets will be eliminated. It may seem useful to add memory, but it is not because the packets arrive when their delivery time to the destination has been expired and their duplicate copies have been sent. According to their task, all these duplicate packets are diverted to the next router and the network load increases in the whole path to the destination. In other words, the congestion will be propagated in the whole of the path.
- **Slow processor:** slow processors cause congestion; despite the high carrying capacity of lines, if routing processors are slow in running tasks like buffering queues, rebuilding tables and etc., there will be queues due to the low bandwidth that results in congestion. Congestion control is carried out in three phases: congestion discovery, congestion announcement and adjusting the sending rate.
- **Lines with low bandwidth:** lines with low bandwidth almost lead to congestion. The lines' bandwidth improvement without any change in processors and vice versa often do little and just changes the bottleneck and origin of the problem. Also, improving a small part (for example, changing a router and upgrading the capacity of its lines) without changing the whole system, only change the location of problem and bottleneck and will not have any significant effect on improvement of the whole system efficiency.

These solutions are applied at different time scales to prevent congestion or to react to it. The increase in packet loss and latency are the disadvantages of congestion. One of the important techniques to provide service quality and to prevent congestion occurrence in computer networks is to use queuing and scheduling management techniques in routers. These mechanisms can be useful to Control the congestion and to improve the efficiency of the network [4].

In a computer network, the packets that are transmitting in the network nodes will be placed in queues in which each of them would be processed or sent according to packet message. Each node queues the packets that cannot be processed currently and takes and processes them in different ways. In computer networks, congestion impacts the reliability and quality of the network service. For example, congestion can cause a queue overflow that increases queuing delays and packet loss. Removing packets not only reduces the reliability and quality of service but also distorts paths in the nodes. Also, congestion reduces the productivity of vehicles connectivity. Congestion usually occurs on network paths and reduces network efficiency. So, it must be properly controlled or prevented in order to increase the network performance. There are mechanisms like congestion detection, congestion announcement and congestion prevention or reduction to be used in congestion control [5].

Increasing growth of the Internet over the last two decades has been accompanied by the arrival of users and new technologies such as mobile devices and wireless media. Today, the major part of traffics uses the transferable protocol, i.e. TCP. Congestion control is very important in TCP-based computer networks. In order to control the flow of data, the buffer is used in TCP protocol, and data is buffered before being sent to the higher layer of application and then is delivered in batches [6]. Sometimes the application cannot receive its buffered data within the deadline, so the buffer will be filled. In this case, the TCP software will not be able to receive and to store the data in its buffer. As a result, in the popup window of each TCP packet which is sent to the other party, the volume of free space of buffer will be declared so that the TCP software could coordinate its sending data with existing buffer`s free space. This means that it does not send a packet larger than declared buffer space. Otherwise, this packet will not be accepted. It can be said that the flow control operation is used to prevent a receiver from being drowning under TCP packets which are sent from a fast sender. In the hybrid model, we use the IWO [7] and KNN [8] algorithms to adjust and to optimize the queuing operation. We form the packets vector by the IWO algorithm and make the classification of the packet to be processed by KNN. In packets` classification, we classify the packets based on size or priority and assign a specific category to each queue. The IWO algorithm [7] is one of the new and powerful meta-innovative optimization algorithms which is inspired by compatibility and randomness of colonic weeds. As defined, weed, as a plant produced and grew in unwanted places, dependent on the conditions, is a serious pest of useful arable crops and inhibits their growth. Despite the simplicity, this algorithm is very effective and fast in finding optimal points. It works on the basis of the original and natural properties of weeds, such as seed production, growth, and competition for survival in a colony.

The KNN [8] assigns the data according to the rate of closeness (similarity) to the categories; therefore, data of each category is different from the other categories. By repeating the same procedure, it is possible to calculate new centers for each data by averaging them, and again to assign data to the new categories. This process continues until there is no change in the data. The KNN is one of the best and most usable learning algorithms and is used widely in many applications.

One of the methods to control congestion in computer networks is to add a packet counter in the packet headings that begin to be numbered from the first packet. In this way, it can be controlled whether packets are lost in the path or not. In this case, it is the receiver task to inform whether packets are received or not by sending specific messages. Usually, receivers use three types of messages to notify the sender, which are named ACK, NACK, and INACK, respectively. They use the ACK to confirm packet arrival and NACK to inform that packet is not received. INACK is like NACK; the difference is that INACK is implicitly installed on another packet. Retrieving the packets lost due to congestion is carried out by two general methods: end to end and step by step method [9]. In the first case, only the packet sender response to resend the data, but in a second way, the middle routers also do it. In this paper, we propose a method to control congestion in computer networks and, we perform the following operations by using the combination of IWO and KNN:

- First, we place the packets generated in the network in IWO algorithm vectors.
- Second, using KNN, we classify packets based on packet priority or size.
- Third, we compare the results of the proposed model with the fuzzy hybrid model and Cuckoo search algorithm [10].

The overall structure of this paper is organized as follow: literature preview is presented in Section 2. The proposed model will be explained in Section 3. In Section 4, the results of proposed model will be assessed and compared with Cuckoo model, and finally, the conclusion and future works will be presented in Section 5.

2. RELATED WORKS

Congestion control in computer networks is a widespread issue which is the research topic of many researchers of the network industry. Today, due to the widespread use of the internet and the simultaneous requests of users from existing resources, congestion is considered to be natural. In line with this problem, a field, called congestion control, is developed. Looking at the research and works done in this area, we will notice the expanse of this issue. Given that today the issue of congestion control in computer networks is an active and widespread field and has become one of the important issues of the day, many researchers of the world work in this field and each of them have different views on this issue.

The Exponential Increase/Multiplicative Decrease (EIMD) model [11] has been proposed with an emphasis on increasing the congestion window. The TCP protocol has a mechanism that reduces the window’s size when there is congestion and increases the window’s size when the congestion problem is removed. The purpose of slider window mechanism is to maintain the data channel filled and to minimize the waiting delay. In the EIMD model, the Round-Trip Time (RTT) the factor is divided into two parts which allow the packets to be sent and to be received faster. The

RTT factor represents the time length it takes to send a packet to the network and to verify it. The results have shown that EIMD model has a higher packet send rate than standard TCP.
TUROWSKA has proposed a fuzzy controller based model to adjust Active Queue Management (AQM) queening [12]. Active queue management is a technique that involves dropping the packets by marking them before the router is filled up. Generally, a router holds a set of queues in each interface which include packets planned to outgo from each interface. The queues use the Drop-Tail method to regularize: a packet is placed on the queue, if the queue (the queue size is measured by packet or byte) is smaller than its maximum capacity, the packet will be delivered and otherwise, it will be discarded. The active queue drops packets or marks them before the queue is filled up. Based on the fuzzy model, the AQM queue capacity is divided into five classifications of very small, small, optimal, large and very large capacity. The results show that the fuzzy-based model of AQM queuing services more packets.

A model based on Fuzzy-Multiple Queue Management (Fuzzy-MQM)[13] is proposed to control congestion in computer networks. In this model, fuzzy logic is used to classify the packets. The fuzzy logic aims to model the systems that cannot be modeled by classical and mathematical methods. Packets are classified into three categories of high, medium and low priority. The type of distribution rate of packets is Poisson in Fuzzy-MQM. The results showed that, in comparison to the MQM model, the Fuzzy-MQM model drops fewer packets. In a total comparison, with 10000 produced packets, the MQM model and the Fuzzy-MQM model had dropped 70 and 120 packets, respectively. So, the results show that the Fuzzy-MQM model has higher performance.
The Quick Transport Control Protocol (QTCP) model [14] is a model for congestion control in computer networks. In this model, the main purpose is to control the congestion window. If the congestion window is controlled properly, fewer packets are discarded in the TCP model. Also, the RTT factor is optimized and less time is wasted to send and to receive packets. The QTCP, with two flows and 500 seconds run, is simulated in the NS2 environment. The results showed that, in comparison to HTCP and CUBIC models, the QTCP model controlled the congestion window of lower size and removed fewer packets.

The Optimization Based Active Queue management (OBQ) queuing based on fair mode AQM is suggested to control congestion[15]. In the OBQ model, the packets are classified into three categories of high, medium and low. The range of Poisson distribution rate of packets is [0.5-1.5]. Active AQM queue management with dropping/marking of packets in routers' queue is proposed as a new suggestion to control the congestion of TCP/IP networks with the end-to-end protocol. The active queue management in TCP/IP network routers takes place to exploit the existing bandwidth and to reduce the sending latency. The results show that, in comparison to RED queue, the OBQ queue drop fewer packets and is more efficient in sending the packets. Active queue management such as RED, requires accurate adjustment of parameters. While, the OBQ queue is adjusted previously and is implemented with its own contractual parameters or with all the restrictions.

Particle Swarm Optimization (PSO) algorithm is proposed to control congestion[16]. One of the important issues in congestion control is the application of queue theory. For this purpose, there are various methods to place the packets in the proposed queue which can be used under various conditions. PSO has been proposed in order to limit packets loss rate which is generated due to the exponential increase of network traffic. With the help of these mechanisms, congestion occurrence is controlled and the network efficiency is prevented to be reduced. The results of the PSO show that the congestion control rate varies in different time intervals and is less than the numbers of sent packets.

A multi-routing model is proposed to control congestion[17]. The purpose of multiple routing is to send data through multiple paths to the destination system. In the multi-routing model, the other systems are used as routing interfaces. The Poisson rate has been used to distribute the packets. Results in the NS2 environment have shown that packet delivery rates in multiple routing are higher than single-rout routing.

A model is proposed for routing and predicting congestion based on the artificial neural network[18]. First, the relationship between links, the number of packets and the time it takes to send the packets is determined. Then, the nodes are given as inputs to the artificial neural network and the training is done on them. In the training phase, the time and links` relation are detected and the optimal route is determined to send the packet. The results have shown that the error rate is very low and the best routes without congestion are selected for packet sending.

Comparison of the proposed models for congestion control based on advantages and disadvantages has been investigated in Table (1).

Table1.Comparison of the Proposed Models for Congestion Control

Refs	Model	Advantages	Disadvantages
[11]	EIMD	●Reducing data delay ●Informing the packet authentication	●Increasing the time length of packets authentication
[12]	Fuzzy Inference System	●Adjusting the AQM queuing ●Classifying the packets based on the size	●Unfair turn rating of packets ●Increasing the packet deletion
[13]	Fuzzy-MQM	●Classifying the packets based on priority ●decreasing the delay ●Reducing the number of packets deletion	●Unfair turn rating of packets
[14]	QTCP	●Controlling the congestion window in TCP ●Reducing the number of packets	●Increasing the traffic

		deletion	
[15]	OBQ model	●Categorizing the classes based on priority ●Reducing the delay	●Unfair turn rating of packets
[16]	Particle Swarm Optimization	●Reducing the number of packets deletion	●Increasing the calculation time ●increasing the length of entry queue
[17]	Multiple routing	●Reducing the number of packets deletion ●Selecting the optimal paths ●decreasing the length of entry queue	●Increasing the network traffic
[18]	Artificial Neural Network	●Detecting the communicating links between the nodes ●Selecting the optimal path	●Increasing the calculation time

3. PROPOSED MODEL

The proposed method, performed in a combined and optimal way to deal with congestion in network, means the elimination of a precursor (prediction of congestion) with an IWO and KNN algorithm. The proposed model is a combination of IWO and KNN algorithm. In this model, the packets are generated in network environments and then based on the packets' size and time-to-live (TTL) [19]field, are placed in vectors of IWO algorithm. The specifications of Packets' fields are as follows:
Size field: this filed specifies the size of packets.
TTL field: TTL field, as a counter, is a 8-byte field that specifies the packet`s lifetime. The packet's lifetime refers to the time that a packet can wander on the network. The maximum lifetime of a packet will be 255 and this amount will be reduced by 1 unit for each router. When a packet delays due to buffering in the memory of a router, 1 unit of this field will be reduced for each second. As soon as the value of this field reaches zero, the packet will be deleted at any point in the path and its route to the destination will be prevented. In figure 1, the flowchart of proposed model is shown.

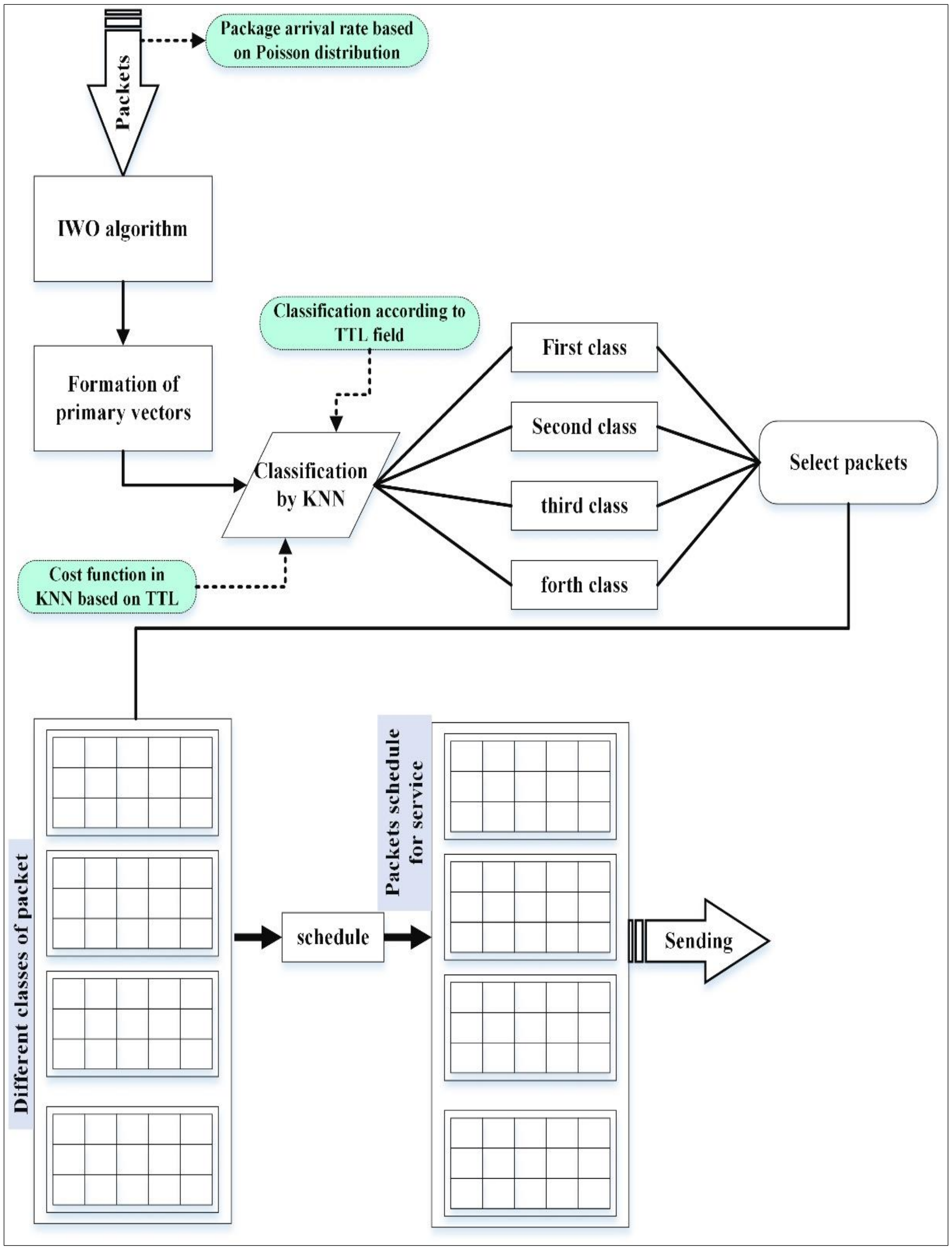

Figure 1.The flowchart f proposed model.

In the proposed model, primary vectors are formed on the basis of the packet size and the TTL field and service to the vectors begins. The total size of produced packets is calculated according to Eq. (1); where, n is the number of packets and P is the packet size.

$$SP = \sum_{i=1}^{n} P_i \qquad (1)$$

In the proposed model, the position of elements in each vector is the same due to the constant size of the packets. The only important issue is the number of elements of each vector. Because the number of each vector is directly related to the value of TTL field. This means that if the number of a vector's elements is high and TTL field waits for a long time in the router line, in this case all the packets will be deleted or the service time of packets will be high and therefore congestion will occur. So, in the proposed model, a duplication technique is used which is based on IWO algorithm. In this technique, vectors that have delivered the packets to recipients without any fault, in their own fitness, generate vectors that use vectors of first stage as a sample to generate vectors of other stages. In fact, the vectors are developed and created to avoid congestion in the network. The following conditions should be considered to create optimal vectors: 1) the number of packets in each vector, and 2) the remained TTL value of each vector. When the packets of each vector arrived at destination, total packets and TTL value are calculated to find the optimal vectors. Eq. (2) is used to create optimal vectors in later stages. In Eq. (2), v is the value of each vector elements, n is the number of vector elements and TTL is the remainder of step number.

$$OV = \frac{\sum_{i=1}^{n} v_i}{TTL\_R} \qquad (2)$$

The process of generating vectors is based on the total number of elements of each vector. The threshold for each vector is 20000 bytes. In the stage of optimality, vectors are created with lower OV value. If the OV value is lower, it means that the greater TTL value remains after the service of packets, and so the number of vector elements is in optimal mode.

3.1. K-NEAREST NEIGHBOR ALGORITHM

In the KNN [8] model, the most important criterion to detect similar samples is using the distance criterion. If a characteristic vector is defined as $< a_1(x), a_2(x), ..., a_n(x) >$, according to Eq. (3), the Euclidean distance will be used to obtain the distance between the two attributes x_j and x_i:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \qquad (3)$$

According to the KNN model, experimental samples are first evaluated and then a model is created to simulate the similarity between the samples. The samples are assigned to categories with respect to distance and similarity. First, a sample, with the largest number of related samples in its neighbor, should be selected for classification. Then, the Euclidean distance between the points is calculated, and by sorting the elements according to Euclidean distance, among neighbor-K, the label with the maximum value is selected for the unknown sample.

In the proposed model, we first put the generated packets in vectors of IWO algorithm. Then we classify the packets based on KNN model. In order to classify the vectors, we give them to KNN model. The KNN model considers 4 levels of classification based on TTL field in each vector. In KNN algorithm, the classification is carried out at 3 levels of low, average, high and very high on the basis of TTL field.

In the KNN algorithm, 4 classes are defined based on TTL field for the cost of each class. Each class holds its own resources and according to specified cost, the generated packets are placed in one of the four classes according to specified cost. The TTL value reduces in each router by 1 unit. For example, if 4 routers are between the origin and destination and the original TTL generated by the destination is 255, the TTL value equals to 255-(4*1), i.e. 251. In addition to the number of routers, one unit of TTL value will be reduced for waiting based on 1 second. Packets classification for the 4 classes are as follow: range of {0-65} for low class, range of {66-132} for average class, range of {133-198} for high class and range of {199-255} for very high class. In order to reduce the number of removed packets in the system, first the low class packets are serviced; because their TTL value is decreasing and they are more likely to be lost than other packets.

3.2. PACKETS SERVICING

The arrival of packets to physical queues according to specified classes and the possibility of removing packets is carried out in classification section. After determining the percentage of packet transmission by KNN algorithm, the packets are transmitted from classification section to router section and the scheduling is done to service or to remove them. Hence, servicing the various queues is done in Round Robin manner; because if packets processing is done in the order of their entry and in a queue, an intruder sender can take up more router capacity and the quality of other services will be reduced. When there is not congestion in the network, any type of packets with any deletion priority can be entered into the queues in the routing section and will receive the service by being transferred to the scheduling section. During the congestion period, entering into the queues in the scheduling section by the KNN algorithm decreases and packets with a high deletion priority will be allowed to enter into the router scheduling section periodically.

The way of vectors forming on the basis of packet size and TTL field is shown in figure 2. The TTL field value reduces by passing the routers and at the KNN stage, each vector is classified based on the TTL field.

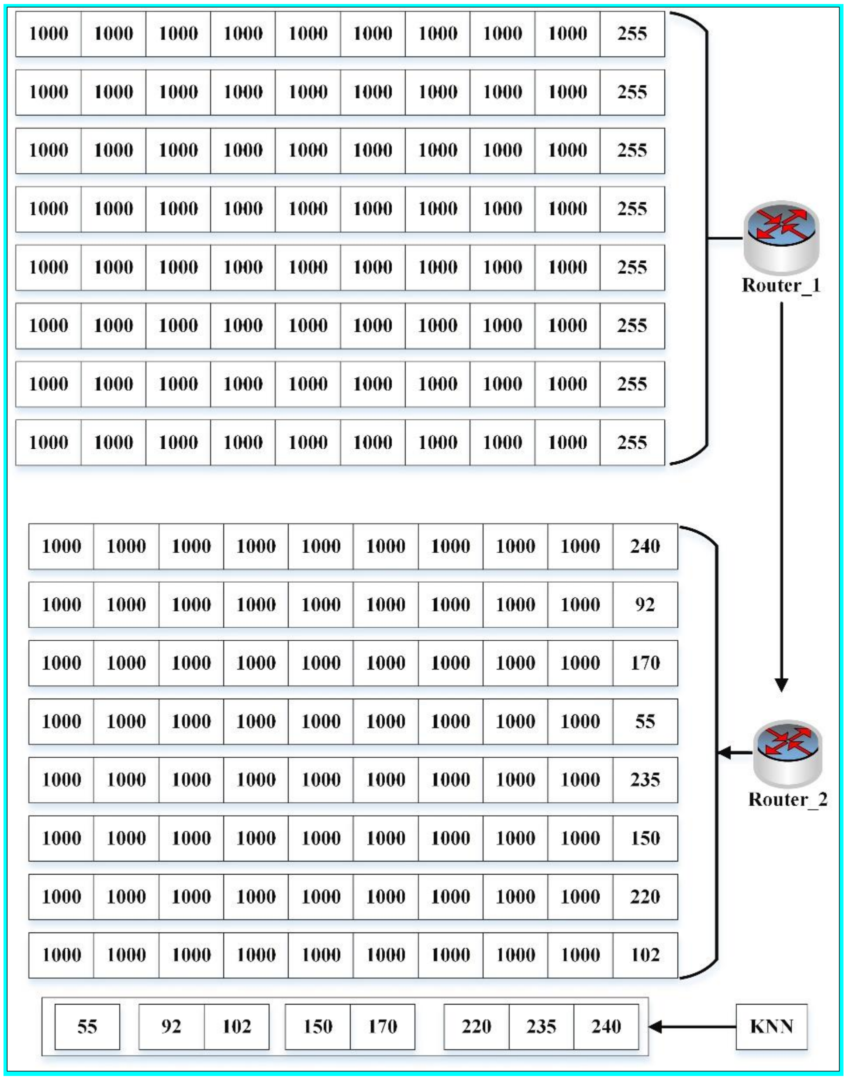

Figure2. Formation of vectors based on packet size and TTL field.

With the packets arrival to physical queues, the possibility of congestion increases. Because the packets are prioritized and enter into the physical queues directly. Therefore, the possibility of packets deletion is calculated by Eq. (4) due to preventing the congestion. In Eq. (4), T_{min} and T_{max} are respectively the minimum and the maximum queue lengths. The parameter S is the queue length in the current state. The parameter ρ_{max} is equal to s/T_{max}. If the pd value is close to 1, the possibility of congestion occurrence is high and if it is less than 1, the congestion will not occur; so, a number of packets should be deleted. If $s \leq Tmax$, the congestion is less and the possibility of packets deletion is low. But, if $s \geq Tmax$, the congestion is more and then the possibility of packets deletion is high.If $Tmin \leq s \leq Tmax$, the queue length is in the mean state, the packets are serviced and there is not any possibility of congestion to be occurred.

$$pd = \frac{p_{\max}(s - T_{\min})}{T_{\max} - T_{\min}},\ T_{\min} \leq s \leq T_{\max}, 0 < p_{\max} \leq 1 \qquad (4)$$

After classifying the packets on the sender side, they are entered into the router, and at this point, the buffer is classified into 4 classes and a physical queue is assigned to each class. The classified packets enter into the router with a specified rate and these packets are placed in the physical and virtual queues of the router. Then, the packets will be shipped from the router securely. Therefore, all the packets are directed to one of the quadruple physical queues that are specified from 1 to 4 and one of the triple queues that are specified from zero to 2, which if not deleted, these packets will arrive to the destination. For example, by queue 42, we mean that the number 4 physical queue and the number 3 virtual queue.

For example, if the queue is empty and the waiting time in the queue is low, it is concluded that the number of packets directed to the queue should be increased. It means that a queue with low service time will result in more packets being sent. The best situation is when queue length is empty and waiting time is low; in this case, each packet can be entered into the queues without any limitation. The waiting time is stored at each stage and accordingly, the best waiting time is resulted. If the queue is full and its waiting time is long, then less packets should be sent to that queue.

3.3. CHARACTERISTICS OF PROPOSED MODEL

The queuing method in the proposed method has the following characteristics:
The packets arrive at the Poisson distribution rate [20, 21]. The Poisson distribution is used to model the number of occurrences within a given time period; i.e. the number of produced packets in a given time range. In the proposed model, the packets arrival rate is calculated according to Eq. (5).

$$f(N = n)\frac{e^{-a}a^{n}}{n!}\ for \quad n = 0,1,2,3,\ldots \qquad (5)$$

In Eq. (5), the parameter a is the arrival time rate of packets per unit time (sec) and the parameter n is the possibility of packets number occurrence.

2) The capacity of the physical queues available in the router classification section is 10 packets and the total system capacity is 80 packets.

3) The router output bandwidth for sending the packets is 7Mb/s and the input packets are from 2 different links, and each of them are entered into the routing queues with 1Mb bandwidth.

4) The size of all packets is constant, 1000 bytes, and the number of packets produced by the source is about 100000 packets.

5) Each queue in the router classification section has a threshold which is equal to 10 packets. So, there are 2 possible situation in this system:

a) If the number of packets in a queue is less than the threshold, the packets with any deletion priority will be entered into the corresponding queue.

b) If the number of packets in a queue is more than the threshold, only the packets with deletion priority will enter into the queue until the queue overflows, in this case, the Tail Drop policy is used.

4. EVALUATION AND RESULTS

The NS2 simulator software is used to evaluate the proposed model. This software is a powerful tool for simulating computer and telecommunication networks that can support various network protocols. This super-simulator is a branch of the REAL Network Simulator project which began in 1989 and has been completed and developed over the last few years. This NS2 software is designed based on an incidental simulation technique and covers many applications, protocols, network types, network components and network models. The NS2 software provides protocols such as TCP, UDP, queuing management mechanisms in routers, such as RED, Drop Tail, CBQ and queuing algorithms. The NS2 software has been designed and implemented using C++ and OTCL programming language. The OTCL programming language is a scripting language with the script structure of the TCL programming language and also features and capabilities of objects. We have implemented the network structure of Figure (3) to compare the performance of rules.

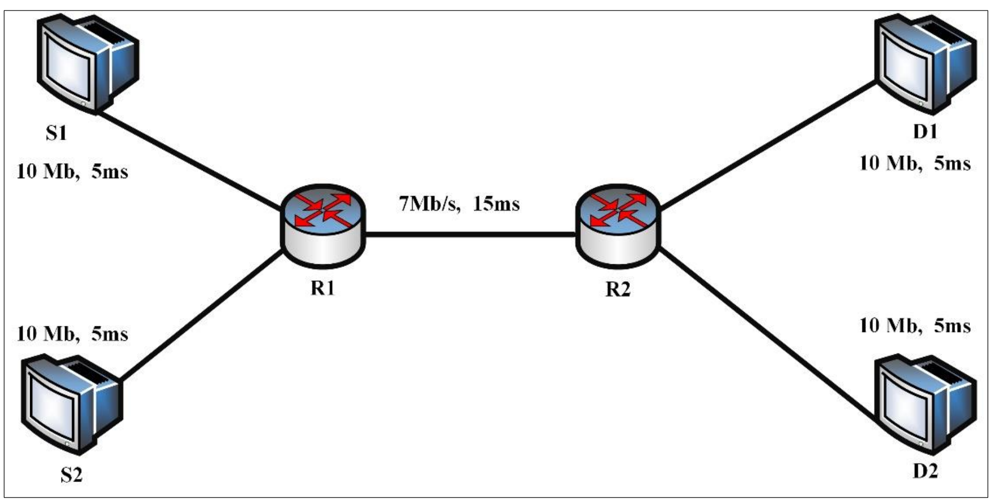

Figure 3. Experimental Network Implemented to Evaluate the Proposed Model [10]

Figure (3) shows the congestion in computer networks in which the communication link between R1 and R2 routers is a bottleneck due to its low bandwidth, and if the packets are sent simultaneously from S1 and S2 nodes, because the input rate to R1 router is higher than the sending rate of the R1 output, the R1 router buffer will gradually filled up. Now, if this process continues for a long time, packets will gradually disappear. When the destination observes the packets loss, it asks the source about packets resending, and as a result, this leads to reduces operational power. According to the scenario executed in seconds 20, 40, 60 and 80, the queue function was sampled and the total number of packets that reached the queue and the number of guided and deleted packets have been determined. It should be noted that the headings of the columns of the report are as follow [10]:

1) TotPkts: the packets produced in system
2) EnPkts: the packets guided to queues
3) TxPkts: sent packets
4) Ldrops: deleted packets

All of the generated packets are guided to one of the four physical queues which are marked from 1 to 4 and one of the three virtual queues which are marked from 0 to 2. These packets, if not removed, have arrived at the destination. The report has been generated with separate virtual and real queues. For example, the 43 queue means the physical queue of number 4 and a virtual queue of number 3. The simulation time is 80 seconds.

4.1. The Results of Proposed Model Run with 20 Seconds

The results of Cuckoo model in the 20s are shown in the Table (2). These results show that, compared to Cuckoo model, the proposed model run with 20 seconds has been able to reduce the number of removed packets and to increase the number of sent packets.

Table 2. Comparison of the Proposed Model with Cuckoo Model run with 20 Seconds

CP	Cuckoo Model [10]				Proposed Model			
	TotPkts	EnPkts	TxPkts	ldrops	TotPkts	EnPkts	TxPkts	ldrops
All	24994	24994	17613	7381	24994	24994	18040	6954
10	2015	2015	2014	1	2015	2015	2015	0
11	2014	2014	1158	856	2014	2014	1482	532
12	2095	2095	1227	868	2095	2095	1327	768
20	2088	2088	2088	0	2088	2088	2087	1
21	2112	2112	1172	940	2112	2112	1418	694
22	2142	2142	1144	998	2142	2142	1221	921

30	2143	2143	2143	0	2143	2143	2142	1
31	2040	2040	1136	904	2040	2040	1284	756
32	2064	2064	1127	937	2064	2064	1218	846
40	2125	2125	2122	3	2125	2125	2123	2
41	2014	2014	1098	916	2014	2014	1283	731
42	2142	2142	1184	958	2142	2142	1218	924

4.2. RESULTS OF THE PROPOSED MODEL RUN WITH 40 SECONDS

The results of the Cuckoo model in the 40s are shown in the Table (3). These results show that, compared to Cuckoo model, the proposed model run with 40 seconds has been able to reduce the removed packets and to increase the number of sent packets.

Table 3. Comparison of the Proposed Model with Cuckoo Model run with 40 Seconds

CP	Cuckoo Model [10]				Proposed Model			
	TotPkts	EnPkts	TxPkts	ldrops	TotPkts	EnPkts	TxPkts	ldrops
All	49994	49994	35093	14901	49994	49994	35589	14405
10	4113	4113	4112	1	4113	4113	4113	0
11	4148	4148	2321	1827	4148	4148	2386	1762
12	4115	4115	2340	1775	4115	4115	2417	1698
20	4166	4166	4166	0	4166	4166	4166	0
21	4156	4156	2322	1834	4156	4156	2360	1796
22	4160	4160	2278	1882	4160	4160	2339	1821
30	4274	4274	4274	2	4274	4274	4273	1
31	4134	4134	2275	1859	4134	4134	2392	1742
32	4159	4159	2224	1935	4159	4159	2263	1896
40	4253	4253	4246	7	4253	4253	4248	5
41	4068	4068	2216	1852	4068	4068	2264	1804
42	4248	4248	2321	1927	4248	4248	2486	1762

4.3. RESULTS OF THE PROPOSED MODEL RUN WITH 60 SECONDS

The results of the Cuckoo model [10] in the 60s are shown in the Table (4). These results show that, compared to Cuckoo model, the proposed model run with 60 seconds has been able to reduce the removed packets and to increase the number of sent packets.

Table 4. Comparison of the Proposed Model with Cuckoo Model Run with 60 Seconds

CP	Cuckoo Model [10]				Proposed Model			
	TotPkts	EnPkts	TxPkts	ldrops	TotPkts	EnPkts	TxPkts	ldrops
All	74994	74994	52603	22391	74994	74994	52950	22044
10	6263	6263	6261	2	6263	6263	6262	1
11	6190	6190	3407	2783	6190	6190	3543	2647
12	6220	6220	3486	2734	6220	6220	3494	2726
20	6233	6233	6231	2	6233	6233	6233	0
21	6244	6244	3465	2779	6244	6244	3623	2621
22	6221	6221	3445	2776	6221	6221	3524	2697
30	6308	6308	6305	3	6308	6308	3607	1
31	6214	6214	3455	2759	6214	6214	3673	2541
32	6256	6256	3395	2861	6256	6256	3492	2764
40	6357	6357	6349	8	6357	6357	6354	3
41	6126	6126	3300	2826	6126	6126	3415	2711
42	6362	6362	3504	2858	6362	6362	3535	2827

4.4. RESULTS OF THE PROPOSED MODEL RUN WITH 80 SECONDS

The results of the Cuckoo model [10] in the 80s are shown in the Table (5). These results show that, compared to Cuckoo model [10], the proposed model run with 80 seconds has been able to reduce the removed packets and to increase the number of sent packets.

Table 5. Comparison of the Proposed Model with Cuckoo Model run with 80 Seconds

CP	Cuckoo Model [10]				Proposed Model			
	TotPkts	EnPkts	TxPkts	ldrops	TotPkts	EnPkts	TxPkts	ldrops
All	99994	99994	70103	29891	99994	99994	70860	29134

10	8372	8372	8370	2	8372	8372	8370	2
11	8249	8249	4514	3735	8249	8249	4570	3679
12	8386	8386	4642	3744	8386	8386	4676	3710
20	8310	8310	8307	3	8310	8310	8309	1
21	8287	8287	4593	3694	8287	8287	4663	3624
22	8295	8295	4627	3668	8295	8295	4650	3645
30	8427	8427	8424	3	8427	8427	8426	1
31	8297	8297	4571	3726	8297	8297	4606	3691
32	8305	8305	4524	3781	8305	8305	4549	3756
40	8429	8429	8421	8	8429	8429	8423	6
41	8221	8221	4461	3760	8221	8221	4489	3732
42	8416	8416	4649	3767	8416	8416	4669	3747

4.5. COMPARISON AND EVALUATION

The results of the current and previous simulation are summarized below in the Table (6), as the total number of sent packets, and Table (7), as the total number of dropped packets. To better understand, the statistics are set in two tables to make the comparison easier in both cases. These tables are arranged in four columns, and their titles and descriptions are as follow:

1) Time(Sec): simulation time in seconds [10]
2) TotPkts: total number of packets produced in the system at any time interval [10]
3) FGenR-TxPkts: sent packets with basic rules [10]
4) FOpR-TxPkts: sent packets with optimized rules [10]
5) FGenR-ldrops: dropped packets with basic rules [10]
6) FOpR-ldrops: dropped packets with optimized rules [10]
7) FdsRedR-TxPkts: sent packets withdsRED traditional method rules [10]
8) FdsRedR-drops: dropped packets with dsRED traditional method rules [10]

The total number of sent packets is shown in Table (6). The results of Table (6) show that the proposed model is more efficient than other models, and also the total number of sent packets are more in the proposed model.

Table 6. Comparison of Total Number of Sent Packets

Time (Sec)	FdsRedR-TxPkts [10]	FGenR-TxPkts[10]	FOpR-TxPkts [10]	Proposed Model

0	0	0	0	0
20	12526	15128	17613	17867
40	25030	30087	35093	35503
60	37526	44962	52603	52890
80	50025	59941	70103	70683

Figure (4) shows the comparison chart of the proposed model with other models based on a total number of sent packets.

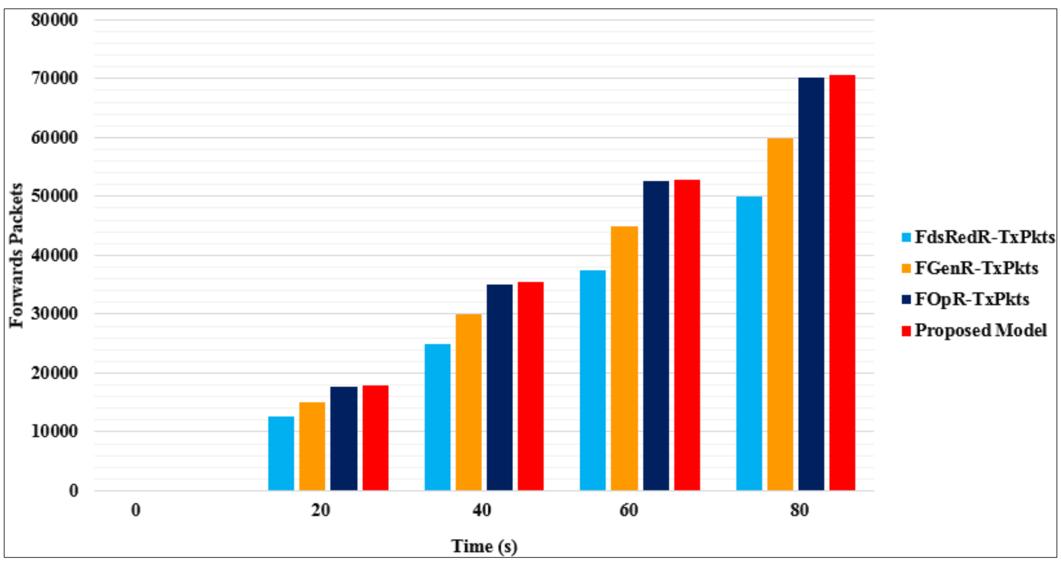

Figure 4. Comparison of the Proposed Model with other Models based on the Total Number of Sent Packets

In Figure (5), the graph diagram of the models is shown based on the total number of sent packets. This diagram shows that the proposed model is more efficient than other models

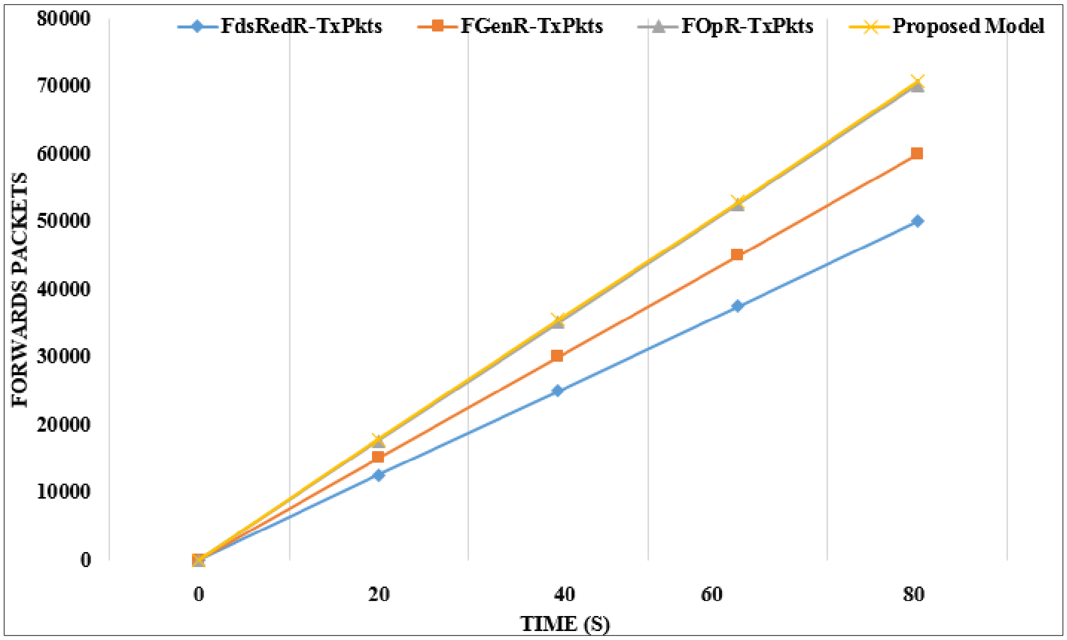

Figure 5. Graph Diagram of the Models based on the Total Number of Sent Packets

The total number of dropped packets is shown in Table (7). The results of Table (7) show that the total number of dropped packets in the proposed model is lower than the other models.

Table7. Comparison of the Total Number of Dropped Packets

Time (Sec)	FdsRedR-ldrops [10]	FGenR-ldrops [10]	FOpR-ldrops [10]	Proposed Model
0	0	0	0	0
20	11992	9459	7381	7127
40	24050	19534	14901	14491
60	36084	29641	22391	22204
80	48118	39684	29891	29412

The comparison of the proposed model with other models based on the total number of dropped packets is shown in Figure (6).

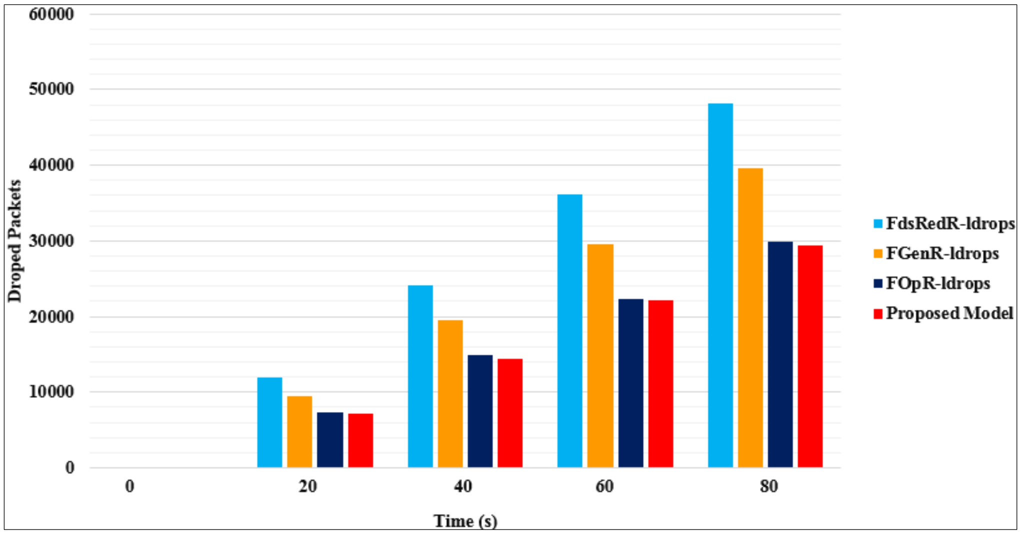

Figure 6. Comparison of the Proposed Model with other Models based on the Total Number of Dropped Packets

In Figure (7), the graph diagram of the models is shown based on the total number of dropped packets. This diagram shows that the number of dropped packets in the proposed model is less than other models.

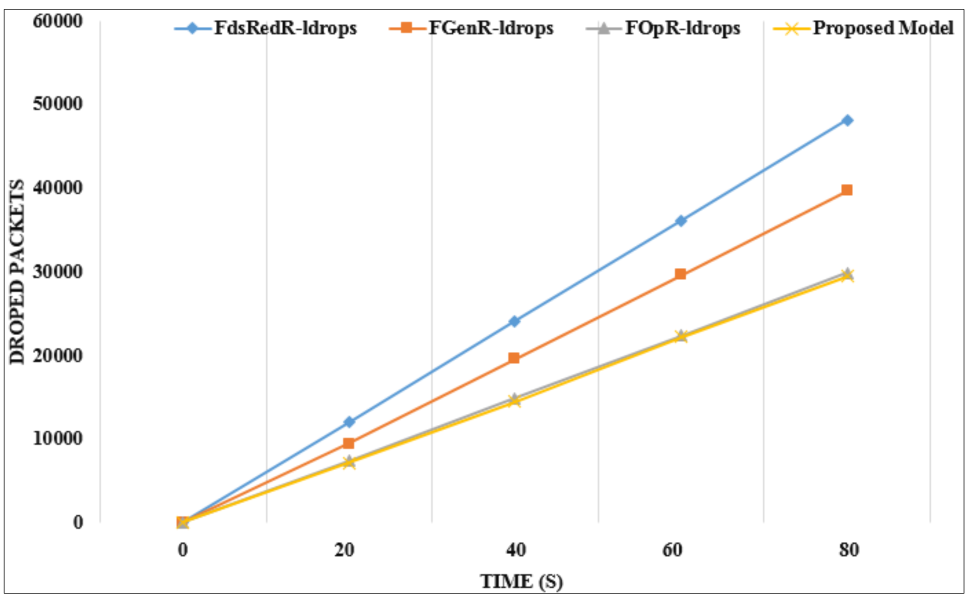

Figure 7. Graph Diagram of the Models based on the Total Number of Dropped Packets

In this section, the proposed model was evaluated and compared to the NS2 environment. The evaluation was carried out based on the number of different packets or times of simulation. The evaluations and comparisons showed that the proposed model has sent more and dropped fewer packets than other models. The success rate of the sent packets in the proposed model was above 70%.

5. CONCLUSION AND THE FUTURE WORKS

The capacity of each part of the network is proportional to the type and conditions of its hardware and software. When each part of the network is loaded more than its capacity, there will be network congestion. If several packets enter from three or four input lines suddenly and then want to exit from one output line, there will be a queue. If there is not enough memory to store them, the packets will be dropped. When the router services the related processes slowly, such as queuing the buffers and upgrading the tables, there will be queues even if the capacity of lines' carrying is more than the current rate. In this paper, we propose a model based on the IWO and the KNN algorithms to control congestion. The main goal of the IWO algorithm was to create packet vectors and then, based on the KNN model, the priority setting was performed. The results showed that the proposed model, in comparison to the other models, sent more packets and dropped fewer packets.

One of the important methods to detect the congestion is to use the queue in order to inform senders about the congestion before the router's buffer are filled up. The routers can play a key role in improving the quality of the service by the proper management of the packets queue. The fair division of the queue's space among different flows and the way of dealing with malicious traffic are effective parameters in the performance of routing queue management algorithm. The number of packets for the future status of the routing buffer can be specified by monitoring the average length of the routing buffer queue.

REFERENCES

[1] M. Casoni, C.A. Grazia, M. Klapez, N. Patriciello, How to avoid TCP congestion without dropping packets: An effective AQM called PINK, Computer Communications, Vol. 103, pp. 49-60, 2017.

[2] S. Grzyb, P. Orlowski, Congestion feedback control for computer networks with bandwidth estimation, 2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR), IEEE, pp. 1151-1156, 2015.

[3] S.B. Alaoui, E.H. Tissir, N. Chaibi, Active queue management based feedback control for TCP with successive delays in single and multiple bottleneck topology, Computer Communications, Vol. 117, pp. 58-70, 2018.

[4] P. Ignaciuk, A. Bartoszewicz, Congestion Control in Data Transmission Networks, Communications and Control Engineering, 2013.

[5] S. V. Uma, K. S. Gurumurthy, Congestion Control by Heuristics in High-Speed Networks Using ANN, Emerging Research in Electronics, Computer Science and Technology, Vol. 248, pp. 323-333, 2013.

[6] M. Fukuhara, F. Hirose, T. Hatano, H. Shigeno, Ken-ichi Okada, SRF TCP: A TCP Friendly and Fair Congestion Control Method for High-Speed Networks, Principles of Distributed Systems, Vol. 3544, pp. 169-183, 2005

[7] A.R. Mehrabian, C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, Ecol. Inform. 1(4): 355-366, 2006

[8] Martin, Instance-Based Learning: Nearest Neighbour with Generalisation, Doctoral dissertation, University of Waikato, 1995.

[9] O. Tickoo, V. Subramanian, S. Kalyanaraman, K.K. Ramakrishnan, LT-TCP: End-to-End Framework to Improve TCP Performance over Networks with Lossy Channels, Lecture Notes in Computer Science, Vol. 3552, pp. 81-93, 2005

[10] F. Moosakhah, A.M. Bidgoli, Congestion Control in Computer Networks with a New Hybrid Intelligent Algorithm, INTERNATION JOURNAL OF COMPUTERS AND TECHNOLOGY, Vol. 13, No. 8, pp. 4688-4706, 2014

[11] E. Yang, Seong-il Ham, S. Cho, Chong-kwon Kim, and P. Lee, EIMD: A New Congestion Control for Fast Long-Distance Networks, Springer-Verlag Berlin Heidelberg, ICOIN 2006, LNCS 3961, pp. 379-388, 2006.

[12] M. Turowska, Adaptive congestion control in computer networks, 19th International Conference on Systems Engineering, IEEE, pp. 205-210, 2008.

[13] B. Safaiezadeh, A.M. Rahmani, E. Mahdipour, A New Fuzzy Congestion Control Algorithm in Computer Networks, 2009 International Conference on Future Computer and Communication, IEEE, pp. 314-318, 2009

[14] B. Qureshi, M. Othman, S. Sabraminiam, and N.A. Wati, QTCP: An Optimized and Improved Congestion Control Algorithm of High-Speed TCP Networks, Springer-Verlag Berlin Heidelberg, ICSECS 2011, Part I, CCIS 179, pp. 56-67, 2011.

[15] N. Farzaneh, R. Monsefi, M.H. Yaghmaee, A.H. Mohajerzadeh, A Novel Congestion Control Protocol with AQM Support for IP-based Networks, Telecommun System, 52:229-244, 2013.

[16] S. Grzyb, P. Orlowski, Congestion control in computer networks: Application of piece-wise affine controller and particle swarm optimization, IEEE, pp. 834-838, 1995

[17] G. Carofiglio, M. Gallo, L. Muscariello, Optimal multipath congestion control and request forwarding in information-centric networks: Protocol design and experimentation, Computer Networks, Vol. 110, pp. 104-117, 2016.

[18] A.Y. Abdalla, T.Y. Abdalla, K.A. Nasar, Routing with Congestion Control in Computer Network using Neural Networks, nternational Journal of Computer Applications, Vol. 57, No.2, pp. 34-41, 2012.

[19] H. Wu, F. Ren, D. Mu, W. Pan, Utilizing TTL to Enhance TCP Fairness, 2007 Second International Conference on Communications and Networking in China, pp. 208- 212, 2007.

[20] W. Wang, X. Chen, C. Wang, H.R. He, an improved preference method for the FAST TCP protocol parameter, Cluster Computing, pp. 1-9, 2018.

[21] S.K. Bisoy, P.K. Pattnaik, Design of feedback controller for TCP/AQM networks, Engineering Science and Technology, an International Journal, Vol. 20, Issue 1, pp. 116-132, 2017.