FNAS Journal of Mathematical Modeling and Numerical Simulation
Print ISSN: 3027-1282
www.fnasjournals.com
Volume 2; Issue 1; September 2024; Page No. 1-13.

Packet Management System in Single-Server Queue Networks with Poisson Arrivals Using a Treap-Based Model

[1]Adeleke I. A., *[1]Gbadebo, A. D., & [2]Oyebode, E., O.
[1]Department of Computer Science Education, Lagos State University of Education, Oto / Ijanikin, Lagos State, Nigeria
[2]Department of Computer Sciences, Ajayi Crowther University, Oyo, Oyo State, Nigeria.

***Corresponding author email:** gbadeboad@lasued.edu.ng

Abstract
There are enormous studies on packets' congestion control in queue networks. However, not enough research work had been done on the management of packets being dropped in queue networks such that these packets could be re-transmitted without accruing unnecessary costs to the system. Since packets' losses amount to a waste of network resources, this study proposes a model with which packets dropped are kept in a tree structure and later re-transmitted to the server once it becomes available. A Treap-Model Congestion Control System (TMCCS) was proposed to prevent network congestion by ensuring that packets arrivingin the system when the queue network is saturated are managed to avoid congestion. The model was benchmarked with Random Early Detection with Reconfigurable Maximum Dropping Probability (RRMDP). OMNeT++ was used as a simulation framework while datasets were generated randomly. Simulation results indicated that while the average throughput for RRMDP was 93.8mbs, that of TMCCS was 103.6mbs. Similarly, while the packets' average queue size for RRMDP was 55.1($x10^{-3}$)mbs, that of TMCCS was 54.3($x10^{-3}$)mbs respectively. Consequently, it was concluded that TMCCS is more efficient in the management of packets in queue networks with regard to network throughput and average queue size.

Keywords: Packets' transmission, Tree manager, Nodes, Network traffic, Tree structure

Introduction
Advances in queue network applications are enormous and are fast gaining applications in industry, education, transport, engineering, etc. With these advancements, traffic patterns in communications networks are being directed at minimizing network congestion. In general terms, congestion control methods are modelled such that transmission rates increase linearly when there are no congestion signals (Wang et al., 2021). The implication of this is that when congestion is discovered, packets' transmission rate is decreased by a multiplicative factor as applicable in the Transmission Control Protocol (TCP) of the Internet. Hamdi et al. (2020) opined that the TCP mechanism is applied to prevent congestion collapse in the Internet while Active Queue Management (AQM) methods had been suggested to complement TCP network congestion control. Using the AQM system, the performance of two network thresholds overasingle threshold is known and can always be modified to derive a lower delay for thesame throughput (Yazdani et al. 2023). However, there are AQM schemes applicable to priority systems which can give aconsiderable queue service system, provide better quality of service (QoS), and managetraffic congestion as well as customer delays (Li et al., 2023).

Altman et al. (2019) investigated the application of controlled delay needed to manage congestion and adaptationto data weight to avoid queue overflow and minimise the occurrence of queue delayat evolved-NodeB (eNodeB) in Long-Term Evolution (LTE) networks using a feedback mechanism. Simulation results indicated that the proposed method performed better than the Random Early Detection (RED) gateway. Similarly, Adesh and Renuka (2019) in a related study suggested adopting a fuzzy-queueing approach to congestion control in queue networks. In this case, the researchers proposed a modification to service quality using the queue method in Internet topology. This involved the application of First-In, First-Out (FIFO), Random RED and Per-Connection Queue (PCQ) methods. Simulation results indicated that RED outperformed PCQ and FIFO when several users were simultaneously downloading data in the queue.

Cite this article as:
Adeleke I. A., Gbadebo, A. D., & Oyebode, E., O., (2024). Packet management system in single-server queue networks with Poisson arrivals using a Treap-based model. *FNAS Journal of Mathematical Modeling and Numerical Simulation, 2*(1), 1-13.

The performance of two adaptive TCP models was compared with RED and fixed-parameter Proportional Integral (PI) by Miao (2019). Simulation results indicated that the two proposed models performed better than the fixed-parameter PI and RED controllers. In a related study by Okokpujie et al. (2018) on the relationship between buffer characteristics and network threshold, experimental results indicated that the rate of change of buffer capacity lies between minimum and maximum thresholds. This implied that the model wasable to reduce the loss of customers to droppings by managing the difference between minimum and maximum thresholds.

Queues are every day's events as people wait in cinemas, banks, hotels, hospitals, supermarkets, airports, hospitals and so on. Queues in this category are visible in nature. Of course, there are also queues of data packets, voice calls, etc in communication channels (Pathan et al., 2024). These are also common but invisible. Most often, queues are not desirable as they cost valuable resources such as money and time (Saini et al., 2023). Generally, queues exist because available service resources are insufficient to satisfy demand. Thiscould be attributed to several factors which might include unavailability of servers as a result of space orcost limitations. In other cases, it may not always pay to provide the level of service necessary to prevent waiting (James et al. 2019).

Hoiland-Jorgensen et al. (2018) defined queue theory as a branch of mathematics which deals with waiting lines. Queue theory uses mathematical tools to predict the behaviour of queueing systems. Predictions deal with the probability of having *n customers* in the system,mean length of queues, mean waiting time, throughput and so on (Floyd et al., 2021). Generally, a queue is formed when customers arrive at a service location expecting to be served with limited resources. If the server is not immediately available, the customers need to join a waiting line. The use of queue theory allows the study of different processes associated with queues including arrivals, waiting and service (Chen et al., 2023). The applications of queue theory in traffic flow, telecommunications and facility design, provide a clear usage of the method in solving a wide range of industrial and domestic problems (Abubakar et al. 2022). A queue system consists of a stream of arriving customers, a queue and a service process. To model such a system, Nagori and Thanker (2018) identified the following basic elements as required:

1. A stochastic process which describes the arrival of packets. This refers to the inter-arrival times of packets and may be any of the following distributions: Poisson, deterministic or general in nature. Generally, inter-arrival times are often assumed to be independent and memoryless which is characteristic of a Poisson distribution;
2. A stochastic process which describes the service system for arriving packets. Thiscould be exponential, constant, hyper-exponential, hypo-exponential or general in nature.
3. The number of available servers is an important factor in any queue system. The queue and service systems depend on whether there is a single server or multipleservers in the network. The size of the queueing network is one of the determining factors for the number of available servers in most cases (Golkar et al. 2022);
4. The capacity of the system. The number of customers, i.e. packets in a queue network can be from *1* to *n*, including the packets awaiting service turns as well as those in service, if any;
4. The size of customers, i.e. packets' population is another important factor. In this case, the queue network could be modeled to have infinite or finite queue length; and
5. The queue discipline describes queue and service patterns. In this case, there are several possibilities in terms of the sequence of customers, i.e. packets to be served, including first-come first-served, last-come first-served, service-in-random order as well as priority discipline (Abdali et al., 2023).

Although numerous researches have been done on packets' congestion control in queue networks, this study focuses on the avoidance of packet losses to droppings due to congestion in queue networks using a Treap structure. This control system is applicable irrespective of the transmission rate of such packets. This is achieved by creating a Treap structure for ensuring network stability with tree operations of insertions, rotations and removals. The process is the same when packets are removed from the tree for transmission to the server. In other words, this study proposes a method of preventing packets losses from dropping in cases of queue network congestion using the Treap structure.

Problem definition

Packets' droppings in queue networks result in the wastage of network resources. This is the reason it is absolutely necessary to control congestion in queue networks. Themajor cause of packets' dropping is an insufficient number of servers which results in longqueues of packets awaiting service turns, consequently

Cite this article as:
Adeleke I. A., Gbadebo, A. D., & Oyebode, E., O., (2024). Packet management system in single-server queue networks with Poisson arrivals using a Treap-based model. *FNAS Journal of Mathematical Modeling and Numerical Simulation, 2*(1), 1-13.

resulting in congestion. In order tocontrol congestion, the system automatically drops some of the packets awaiting service turns to ensure queue network stability. This is why a direct relationship often exists between throughput and latency in queue networks. Figure 1 shows the relationship between throughput and latency.

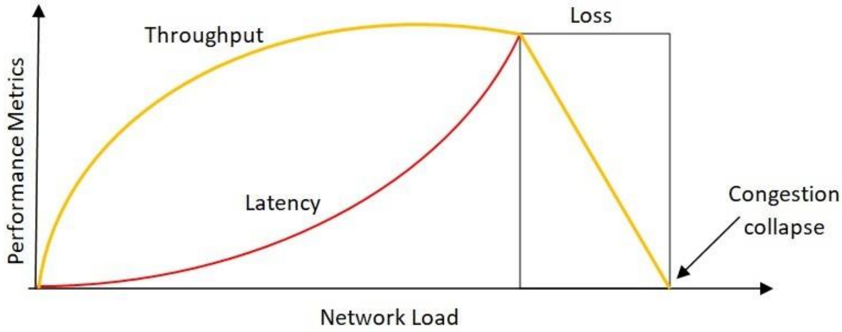

Figure 1: Relationship between network throughput and latency

Methodology
This section describes the methods used in the study.

Queue network stability
A stable queue network is one that has a proper limiting distribution of the queuesize such that for every n=0, 1, …, ∞, the following limit exists:

$$\lim_{t\to\alpha} P(X(t)-n) = p_n \quad \text{and} \quad \sum_{n=0}^{\alpha} Pn = 1$$

From the above, it is possible to regard a queue network without a droppingfunction to be in a stable state, if and only if, $\rho < 1$. Conversely, a queue network witha dropping function *d(n)* and load ρ is regarded to be in a stable state, if and only if:

$$S = \sum_{k=1}^{\alpha} \rho^k (1-d(0))(1-d(1))\dots(1\text{-}\mathrm{d}(\mathrm{k}\text{-}1)) < \alpha$$

In essence, at time t_0, the queue size is *n* while the service process is suspended. This refersto the time the first packet is admitted with an exponential distribution density of:

$$f_n(t) = (1-d(n))\lambda e^{-1(1-d(n))\lambda t}$$

This implies that until the first packet is admitted, the Poisson arrival process has an intensity λ and probability $d(n)$ at rate $(1-d(n))\lambda$. In essence, if the queue size is $n > 0$ at time t_0, while packets are been serviced, based on the order of event, the queue size either reduces after an exponential service time with parameter μ or increases after an exponential service time with parameter $(1-d(n))\lambda$. This means that if the queue size is $n = 0$ at time t_0, the initial packet arrival parameter is $(1-d(0))\lambda$. This allows an increment after an exponential service time. The queue size can be either $n-1$ or $\mathrm{n}+1$ after the first increment. The followingintensity matrix is possible considering the continuous-time Markov chain such that:

$$Q = [q_i, j]_{i=0,1,\dots\, j-0,1,\dots} \quad \text{where}$$

$$q_{i,j} = 0, \quad i > j+1 \quad \text{or} \quad \mathrm{j} > \mathrm{i}+1,$$

$$q_{i,i+1} = (1-d(i))\lambda, \quad \mathrm{i} = 0,1,\dots,$$

$$q_{i+1}, \quad \mathrm{i} = \mu, \quad \mathrm{i} = 0,1,\dots,$$

Cite this article as:
Adeleke I. A., Gbadebo, A. D., & Oyebode, E., O., (2024). Packet management system in single-server queue networks with Poisson arrivals using a Treap-based model. *FNAS Journal of Mathematical Modeling and Numerical Simulation, 2*(1), 1-13.

$$q_{i,i} = -1(-d(i))\lambda - \mu, \quad \mathrm{i} = 1,2,...$$

$$q_{0,0} = -(1-d(0))\lambda$$

From the foregoing, it is possible to stabilize a queue network in a continuous-timeMarkov chain, if and only if,

$$\Sigma_{k=1}^{\alpha}(q_{0,1}, q_{1,2,...,}q_{k-1},_k \ / \ \mathrm{q}_{1,0,}\mathrm{q}_{2,1,...,}\mathrm{q}_{k,k\text{-}1}) < \alpha$$

Proposed Model

The model proposed for the control of packets' dropping is shown in figure 2.

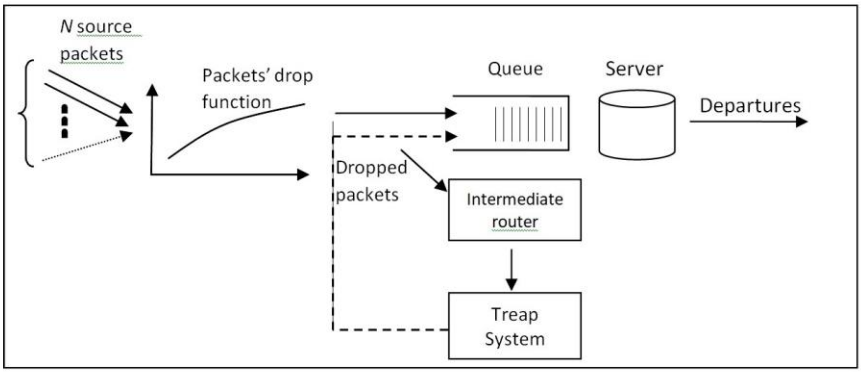

Figure 2: Proposed model

Packets are transmitted through *N* source into a buffer. When the transmission rate exceeds the capacity of the server, the system begins to transmit packets awaiting service into the Treap-based system through the intermediate router in order to avoid undue droppings and consequent losses. Generally, packets are transmitted from source to destination when the queue system is in a steady state. However, whenthe system begins to function below expectation or stops functioning, possibly as a result of the server's partial or complete breakdown, arriving packets are not served as expected. When the server is partially faulty, for instance, it operates below capacity or with compromised results. Similarly, when the server is completely faulty, it stop rendering service. In either of these cases, arriving packets are under-served. This couldresult in a situation whereby these packets are dropped. In the proposed model, such packets are transmitted to the Treap-based model through the intermediate router in this case.

Generic tree structure

A tree is an abstract data type which can be used in the simulation of a hierarchical structure with a root value and sub-trees of children with a parent node, represented as a set of linked nodes. A tree is a non-linear and hierarchical data structure consistingof a collection of nodes such that each node of the tree stores a value which is a listof references to the nodes i.e. the children (Dordal, 2021).

A tree, *T* is a finite set of one or more nodes such that there is one designated node *R*, called the root of *T*. If the set $(T - \{R\})$ is not empty, these nodes are partitioned into $n \geq 0$ disjoint subsets $T_0, T_1, ..., T_{n-1},$, each of which is a tree, and whose roots $R_1, \mathrm{R}_2, ..., R_n,$, respectively, are children of *R*. The subsets $T_i(0 \leq i < n)$ are subtrees of *T* which are ordered such that T_i comes before T_j provided $i < j$. Conventionally,the subtrees are arranged from left to right with subtree T_0 referred to as the leftmostchild of *R*. Figure 3 shows the basics of a tree structure.

Cite this article as:
Adeleke I. A., Gbadebo, A. D., & Oyebode, E., O., (2024). Packet management system in single-server queue networks with Poisson arrivals using a Treap-based model. *FNAS Journal of Mathematical Modeling and Numerical Simulation, 2*(1), 1-13.

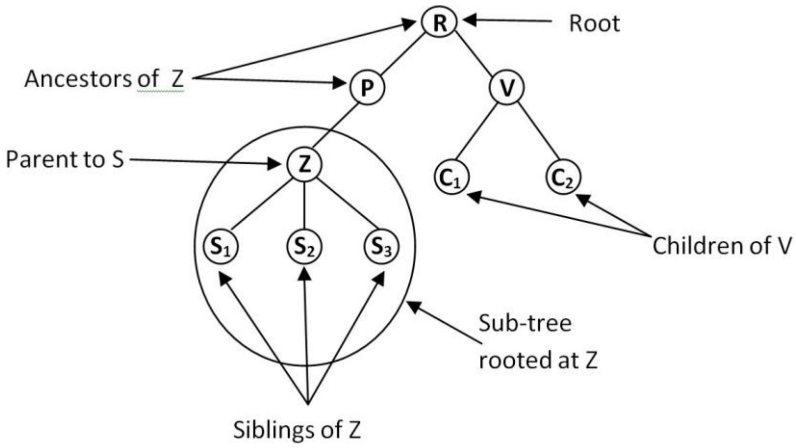

Figure 3: Generic tree structure

In Figure 3, node R is the root and parent of nodes P and V**.** While C_1 and C_2 are the children of V, the siblings of Z are S_1, S_2 and S_3. Nodes R and P are ancestors of node Z. The oval surrounds the subtree having Z as its root.

A tree is made up of a root and zero or more subtrees. Consequently, there is an edge from the root to each subtree in any tree. In a tree, X is a set of n items such that each n item has a key and a priority which is drawn from an ordered universe. ATreap for X is a rooted binary tree with node set X that is arranged in in-order (for the keys) and heap-order (for the priorities). This implies that a Treap is a binary search tree (BST) with two parameters: a *key* and a *priority*. A Treap structure of N nodes is represented mathematically as:

$$\text{Treap (T)} = \{(a_1, b_1), (a_2, b_2), ..., (a_n, b_n)\}$$

where a is the S_{on} and b is T_{on}.

A Treap-Model Congestion Control System (TMCCS) was proposed. In the design of the proposed model, the structure adopted is a tree with several nodes which represent network packets. Since packets are treated as nodes in the tree, the packet's information considered in the design are "size of node" i.e. S_{on} and "time of node" i.e. T_{on}. In essence, every node in the tree has two parameters i.e. S_{on} and T_{on} on the basis of which node selection, removal from the tree and consequent transmission to the server are made.

The Treap structure adopted in the study is presented in Figure 4.

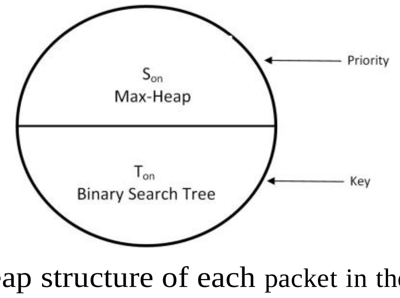

Figure 4: Treap structure of each packet in the tree

Cite this article as:
Adeleke I. A., Gbadebo, A. D., & Oyebode, E., O., (2024). Packet management system in single-server queue networks with Poisson arrivals using a Treap-based model. *FNAS Journal of Mathematical Modeling and Numerical Simulation, 2*(1), 1-13.

Table 1 shows the first four packets in the tree.

Table 1 Sample packets treated as nodes in the tree

Nodes	S_{on} (bytes)	T_{on} $(x10^{-1})$ ms (microseconds)
1	31	16
2	14	12
3	23	18
4	17	9
.	.	.
.	.	.
.	.	.
N_n	S_{onn}	T_{onn}

These packets are arranged as nodes in the tree given in figure 5.

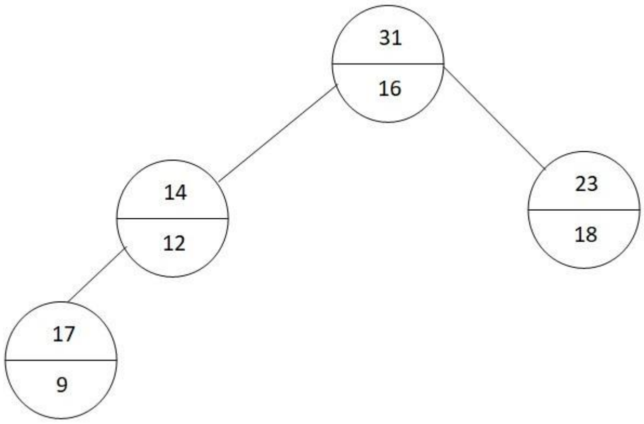

Figure 5: A tree showing S_{on} and T_{on} values arranged in max-heap and BST order respectively

Node insertion operation

Insertion in a Treap takes cognizance of the entire properties regarding the heap andBST of all nodes contained in the tree. The node insertion algorithm is applicable in a Treap is presented below.

Insertion of node in the max-heap tree

```
void insert (int node) struct
node*t, node*p:
t = (node*) malloc (size of (struct node));
t*data = d;
t→left = NULL;
t→right = NULL;
p = root;
if (root = NULL)
root = t;
else struct node* current;current = root;
while (current) parent = current;
if (t → data > current → data)
current = current → right;
else  current = current → left;
if (t → data > parent → data)
parent → right = t;
else parentleft = t
```

In order to insert a node into the Treap, consideration is given to both the BSTand max-heap properties of the tree. For instance, if node $S_{on}21{:}T_{on}20$ arrives the sub-destination to join the tree in Figure 5, it is attached as indicated in Figure 6.

Cite this article as:
Adeleke I. A., Gbadebo, A. D., & Oyebode, E., O., (2024). Packet management system in single-server queue networks with Poisson arrivals using a Treap-based model. *FNAS Journal of Mathematical Modeling and Numerical Simulation, 2*(1), 1-13.

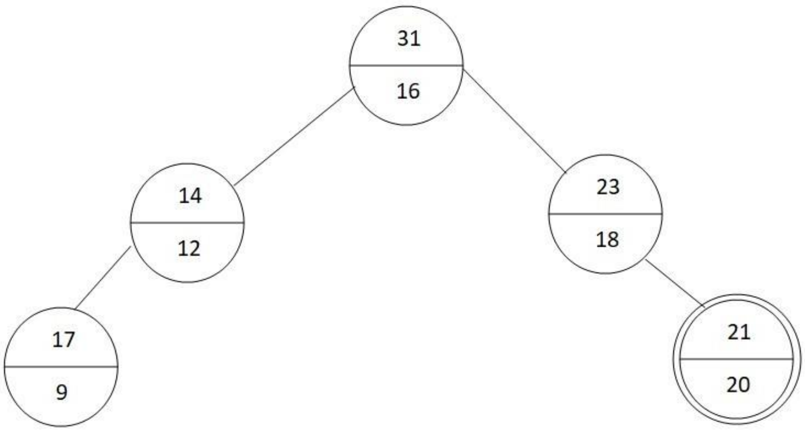

Figure 6: Placement of node $S_{on}21{:}T_{\bar{o}n}20$ based on BST and max-heap properties

The process of node insertion adopted in the study is given in the following algorithm.
Node insertion in the proposed modelInput:
Node_Insertion (self, root, key)
if not root
return Tree_Node(key)
else if key < root.value
root l = self insert(*root l*, key)
else root r = self_ insert(*root r*, key)
If T = tnull *then*
T ← *New_ Node*()
T → [T_{on}, S_{on}, lchild, rchild] ← [T_{on}, S_{on}, tnull, tnull, tnull]
else *if* S_{on} < T → T_{on} then
Node Insert ((S_{on}, T_{on}), T → lchild)
if T → lchild→ packet length > T→ packet length *then*
Insert_Right(T)
else if S_{on} > T → T_{on} *then*
Node_Insert ((S_{on}, T_{on}), T→ rchild)
if T → rchild → S_{on} > T → S_{on} *then*
Insert_Left(T)
root h = 1 + *max*(*self getHeight*(*rootl*) *r*)
else T_{on} is already in Tree T.

Packets' removal from tree and consequent transmission
It is important to note that removal of node in a Treap also follows the considerationof both the heap and BST properties of the entire tree. The root node removal module in a max-heap tree is as presented below.

Removal of root node in a max-heap tree (tree, n, item)
item = tree[i];
last = tree[n], n = n-1;
loc=1, left=2, right=3;
while (right ≤n);
if (last ≥ tree [left] && last ≥ tree [right])
tree[loc]=last;
return
if (tree[right] ≤ tree[left])
tree[loc]= tree[left], loc=left;else
tree[loc] = tree[right], loc=right;
left = 2*loc;right = 2*loc + 1;

if (*left* == *n* && *last* < *tree* [left])
tree[loc]=tree[left], loc=left;}
tree[loc]=item.

In order for a node to be transmitted to a server as a packet, it has to be removed from the tree. For the purpose of this study, the following algorithm is used for the removal of a root node in the tree.

Node Removal (S_{on}:T_{on}, Node:tree)
If tnull → WT ← NS Node
Removal (S_{on}, Node)
If S_{on} < Node → T_{on} *then*
Node Removal (T_{on}, Node → lchild) *else*
if S_{on} > Node → T_{on} *then*
Node_Removal(T_{on}, Node→ *rchild*) *else*
Root_Removal(T)
Procedure Root Removal (Node : Tree)
if Is Leaf Or Null(Node) *then*
Node← tnull
else if T→ *lchild*→ S_{on} > T→ *rchild*→ S_{on} *then*
Remove_Left (Node→ *rchild*)
else Remove_Right (Node→ *lchild*).

In figure 6, if a node is to be removed as transmitted to the server after approximately 2*ms*, then the node to be removed is S_{on} 31: T_{on} 16 being a max-heap tree. The resulting tree is given in Figure 7.

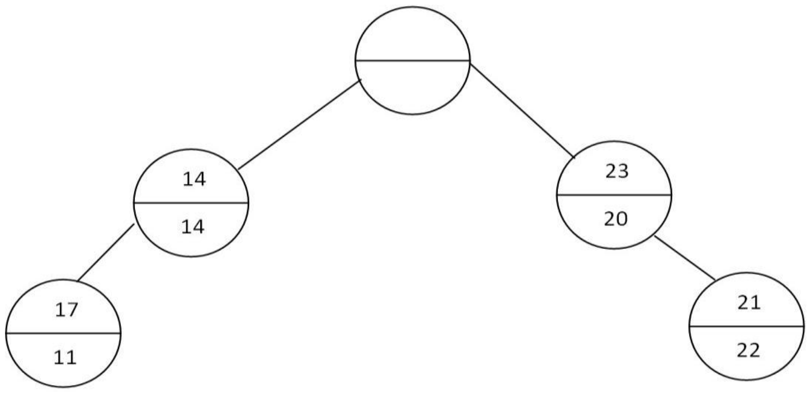

Figure 7: Resulting tree after the removal of root node S_{on}31:T_{on}16 for transmission to server

In order to replace the node that was removed, the tree is rotated and the resulting tree is indicated in figure 8.

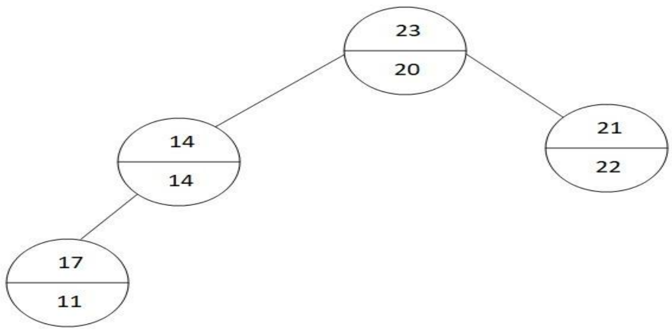

Figure 8: The resulting tree after rotation of the tree

In Figure 8, if a node is to be removed as transmitted to the server after ≈1*ms*, then the node to be removed is S_{on} 23:T_{on} 20 being a max-heap tree. The resulting tree is given in Figure 9.

Cite this article as:
Adeleke I. A., Gbadebo, A. D., & Oyebode, E., O., (2024). Packet management system in single-server queue networks with Poisson arrivals using a Treap-based model. *FNAS Journal of Mathematical Modeling and Numerical Simulation*, *2*(1), 1-13.

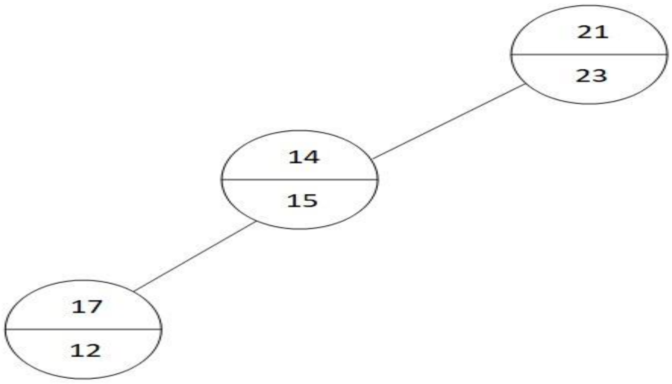

Figure 9: Resulting tree after the deletion of the root node

The processes of node insertions and deletions continue in this order until all nodes are eventually removed from the tree for transmission to the server.

Implementation and simulation

OMNet++ was used as simulation testbed. Consideration is given to the sizes of packets as well as their corresponding "wait time" in the tree.There were eight traffic sources as well as traffic destination points, a source router, abottleneck router as well as a destination router. The bottleneck router serves as the link between the source and destination routers. Connections to the traffic sources had a router with 25mbs while the link between the source and the bottleneck routeris 3mbs. Similarly the link between the bottleneck and destination router is 25mbs. The delay between the source and the bottleneck router is 20ms while that of the bottleneck and destination router was generated randomly.
A packet generator (PktGen) was used to generate network traffic with various sizes and patterns. Both the minimum (min_{th}) and maximum threshold (max_{th}) for both models were set as $min_{th} = 75$ while $max_{th} = 225$ (i.e. the max_{th} is triple of thevalue of min_{th}). This ratio had been suggested Su, et al (2023). The traffic generating sources and destination have packets' and acknowledgment sizes of 1448 and 40 bytes respectively.

Results

The proposed model, i.e TMCCS was benchmarked with Random Early Detection with Reconfigurable Maximum Dropping Probability (RRMDP) as proposed by Al-Allaf and Jabbar (2019). The performance of both models was compared using the following metrics: throughput, packets' average queue size, packets' average wait time and latency.

System throughput

The performance of the queue network is such that network throughput increases linearly as the network load. This is evident in figure 10.

Cite this article as:
Adeleke I. A., Gbadebo, A. D., & Oyebode, E., O., (2024). Packet management system in single-server queue networks with Poisson arrivals using a Treap-based model. *FNAS Journal of Mathematical Modeling and Numerical Simulation, 2*(1), 1-13.

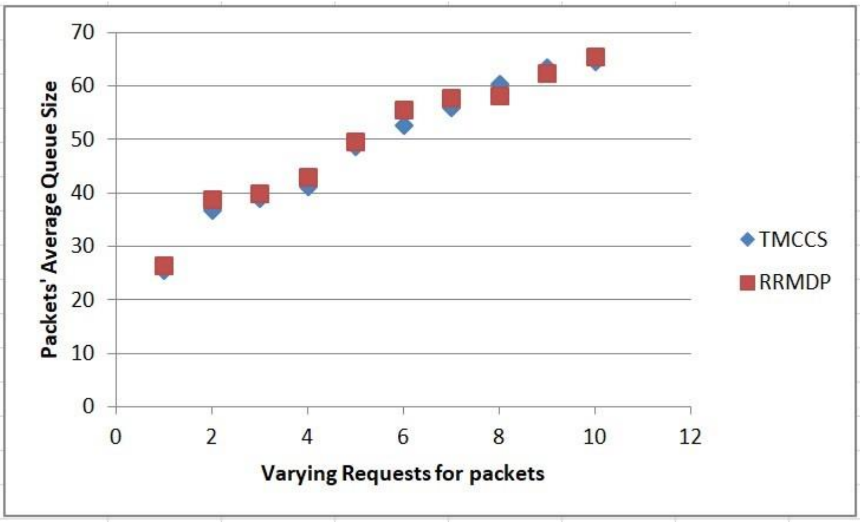

Figure 10: Throughput performance of TMCCS and RRMDP

From Figure 10, the performance of TMCCS is reasonably higher than RRMDP. While RRMDP has throughput of 42.5mbs, 51.8mbs, 66.9mbs, 68.2mbs and 69.8mbs, TMCCS has 49.5mbs, 56.8mbs, 67.9mbs, 71.2mbs and 72.8mbs respectively for the first five iterations. Similarly, the average throughput for RRMDP and TMCCS was 93.8mbs and 103.6mbs respectively. This indicates a significant improvement in the performance of TMCCS over RRMDP.

Packets' Average Queue Size

Packets' average queue size refers to the average size of packets waiting for service turns in the queue network. The size of packets awaiting service turns continue to increase as more packers arrive and the service capacity of the server remains the same. In most cases involving a static queue network, the average queue size has a direct relationship with packets' arrivals. Figure 11 shows the performance of TMCCS and RRMDP with regards to average queue size.

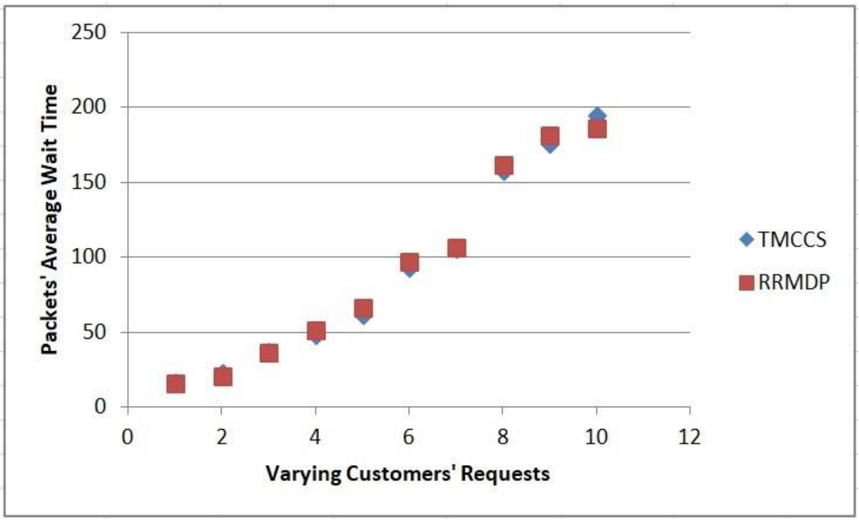

Figure 11: Average queue size performance of TMCCS and RRMDP

It is obvious from Figure 11 that TMCCS and RRMDP have the following average queue size for the third to the seventh iterations as 38.9mbs, 41.2mbs, 48.8mbs, 52.6mbs and 55.9mbs ($x10^{-3}$) and 39.9mbs, 43.2mbs, 49.8mbs, 55.6mbs and 57.9mbs ($x10^{-3}$) respectively. In a similar way, the packets' average queue size for RRMDPand TMCCS was 55.1 ($x10^{-3}$)mbs and 54.3 ($x10^{-3}$)mbs respectively. This indicates asignificant improvement in the performance of TMCCS over RRMDP.

Cite this article as:
Adeleke I. A., Gbadebo, A. D., & Oyebode, E., O., (2024). Packet management system in single-server queue networks with Poisson arrivals using a Treap-based model. *FNAS Journal of Mathematical Modeling and Numerical Simulation, 2*(1), 1-13.

Packets' Average Wait Time

Packets' average wait time refers to the average time packets wait in the queue network before being served as well as in service. Generally, average wait time of packets awaiting service turns continue to increase as more packers arrive and the service capacity of the server remains unchanged being a single server queue network. In essence, the average wait time of packets has direct relationship with packets' arrivals. Figure 12 shows the network performance of TMCCS and RRMDP with regards to average queue size.

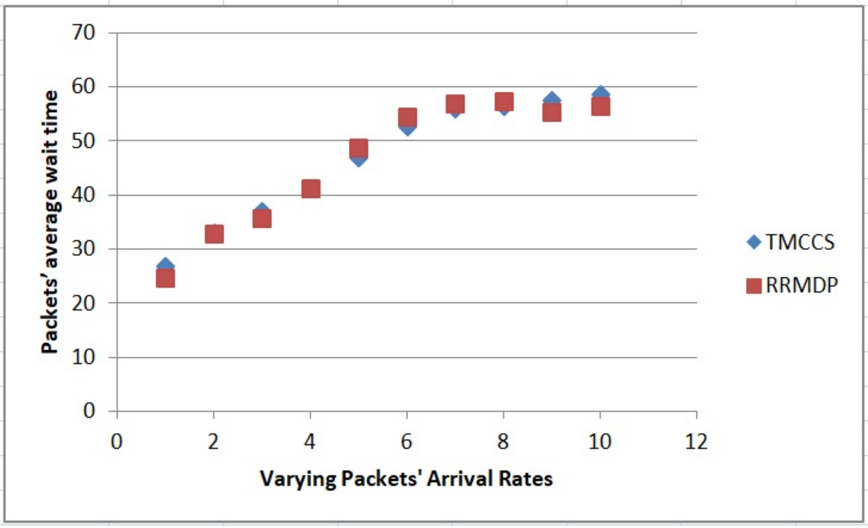

Figure 12: Packets' average wait time performance of TMCCS and RRMDP

It is obvious from Figure 12 that while TMCCS has the following averagewait time of packets for the first five iterations: 16.8s, 22.9s, 36.9s, 48.2s and 61.8($x10^{-1}$)s, RRMDP has 15.5s, 20.3s, 35.9s, 50.9s and 66s ($x10^{-1}$). Consequently, the averagewait time of packets for RRMDP and TMCCS as shown in Figure 12 are 102.6($x10^{-1}$)s and 103.1($x10^{-1}$)s respectively. This indicates a seemingly equal performance for both methods.

Latency

Latency refers to the delay in network communication. It is the time data takes to travel across a network. In other words, latency is the amount of time taken for a packet of data to travel through multiple devices and then be received. The performance of both methods regarding latency isdepicted in Figure 13.

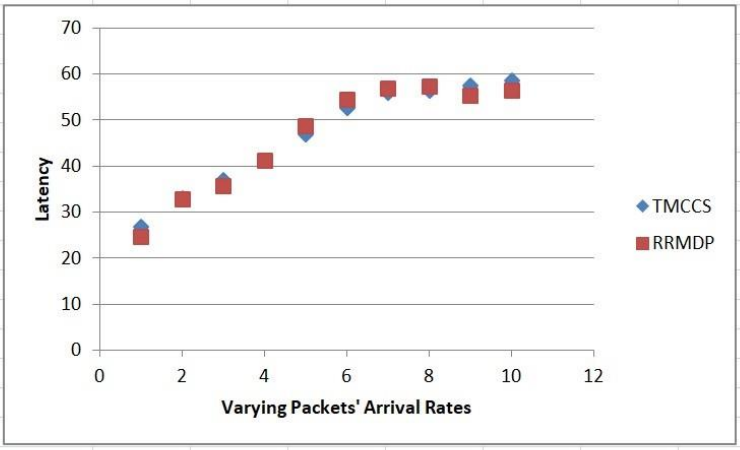

Figure 13: Latency of TMCCS and RRMDP

The latency for the sixth to the tenth iterations in Figure 13 is 52.6s, 55.9s, 56.4s, 57.4s and 58.6s

Cite this article as:
Adeleke I. A., Gbadebo, A. D., & Oyebode, E., O., (2024). Packet management system in single-server queue networks with Poisson arrivals using a Treap-based model. *FNAS Journal of Mathematical Modeling and Numerical Simulation, 2*(1), 1-13.

$(x10^{-1})$s for TMCCS and 54.6s, 56.9s, 57.4s, 55.5s and 56.6s $(x10^{-1})$s for RRMDP respectively. The average latency for TMCCS and RRMDP are 59.4 $(x10^{-1})$s and 57.6 $(x10^{-1})$s respectively. The performance of both methods as indicated in Figure 13 gave a performance of > 0.35% for RRMDP over TMCCS.

In summary, the study provides an insight into the management of packets in queue networks with single server using a Treap-based model. The proposed model is TMCCS and was benchmarked with RRMDP. Simulation results indicated that the proposed model, i.e. TMCCS out-performed RRMDP with regards to network throughput and average queue size. However, it was discovered that both methods are equal in performance with regards to packets' average wait time while RRMDP performed fairly better than TMCCS with regards to network latency.

Conclusion

This study proposes a Tree-Based Congestion Control Model for a single server queue network with Poisson arrivals. Congestion is controlled in the proposed model by using a tree structure in which nodes are treated as packets taking cognizance of their sizes and waiting times. Simulation results show a considerable improvement in network throughput, average queue size and average wait time. However, a negligible performance of <0.35% was discovered for latency of RRMDP over TMCCS. From these results, it is concluded that TMCCS is effective in the packets' management in single server queue networks.

References

Abdali, N., Heidari, S., Alipour-Vaezi, M., Jolai, F., & Aghsami, A. (2023). A priority queueing-inventory approach for inventory management in multi-channel service retailing using machine learning algorithms. *Kybernetes, ahead-of-print*(ahead-of-print). https://doi.org/10.1108/K-07-2023-1281

Abubakar, I. A., Arora, G., Kumar, B., & Danjuma, M. (2022). The optimal number of servers in a many server queuing system. *Journal of Physics: Conference Series, 2267*(1), 012105. https://doi.org/10.1088/1742-6596/2267/1/012105

Adesh, N., & Renuka, A. (2019). Avoiding queue and reducing queueing delay at eNodeB in LTE networks using congestion feedback mechanism. *Computer Communications, 14*(6), 131–143.

Al-Allaf, A. F., & Jabbar, A. (2019). RED with reconfigurable maximum dropping probability. *International Journal of Computing and Digital Systems*, *8*(1), 61–72.

Altman, E., Avrachenkov, K., & Ayesta, U. (2019). A survey on discriminatory processor sharing. *Queuing System Theory and Applications*, *5*(3), 53–63.

Chen, H., Duenyai, S., & Iravani, S. (2023). Admission and routing control of multiple queues with multiple types of customers. *IEEE/ACM Transactions on Networking*, *44*(3), 998–1011.

Dordal, P. L. (2021). *An introduction to computer networks* (pp. 537). Department of Computer Science, Loyola University, Chicago.

Floyd, S., Gummadi, R., & Shenker, S. (2021). Adaptive RED: An algorithm for increasing the robustness of REDs active queue management [Technical report]. ICSI. http://www.icir.org/oyd

Golkar, A., Malekhosseini, R., RahimiZadeh, K., Yazdani, A., & Beheshti, A. (2022). A priority queue-based tele-monitoring system for automatic diagnosis of heart diseases in integrated fog computing environment. *Health Informatics Journal*, *28*(4). https://doi.org/10.1177/14604582221137453

Hamdi, M. M., Mahdi, H. F., Abood, M. S., Mohammed, R. Q., Abbas, A. D., & Mohammed, A. H. (2020). A review on queue management algorithms in large networks. *2nd International Scientific Conference of Engineering Sciences (ISCES)*, 110. https://doi.org/10.1088/1757-899X/1076/1/012034

Hoiland-Jorgensen, T., T'aht, D., & Morton, J. (2018). Piece of CAKE: A comprehensive queue management solution for home gateways. *Proceedings of IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN 2018)* (pp. 37–42). https://doi.org/10.1109/LANMAN.2018.8475045

James, S., Prakash, P., & Nandakumar, R. (2019). The tree list: Introducing a data structure. *International Journal of Recent Technology and Engineering (IJRTE)*, *7*(6), 1093–1095.

Li, B., Tan, K., Luo, L. L., Peng, Y., Luo, R., Xu, N., Xiong, Y., Cheng, P., & Chen, E. (2023). Clicknp: Highly flexible and high performance network processing with reconfigurable hardware. *Proceedings of the ACM SIGCOMM Conference* (pp. 1–14).

Miao, W. (2019). Stochastic performance analysis of network function virtualization in future internet. *IEEE Journal on Selected Areas in Communications*, *37*(3), 613–626.

Nagori, V., & Thanker, S. (2018). Analysis of fuzzification process in fuzzy expert system. *International*

Conference on Computational Intelligence and Data Science (ICCIDS). Retrieved from www.sciencedirect.com

Okokpujie, K. O., Chukwu, E. C., Noma-Osaghae, E., & Okokpujie, I. P. (2018). Novel active queue management scheme for routers in wireless networks. *International Journal on Communications Antenna and Propagation, 8*(1), 53–61.

Pathan, N., Muntaha, M., Sharmin, S., Saha, S., Uddin, A., Nur, F. N., & Aryal, S. (2024). Priority-based energy and load-aware routing algorithms for SDN-enabled data centre network. *Computer Networks, 240*, 110166. https://doi.org/10.1016/j.comnet.2023.110166

Saini, A., Gupta, D., & Tripathi, A. K. (2023). Performance analysis of priority queue network consisting biserial and parallel channel. *AIP Conference Proceedings, 2916*, 060002. https://doi.org/10.1063/5.0177469

Su, X., Shi, P., Wu, L., & Song, Y. D. (2023). A novel approach to filter design for fuzzy discrete-time systems with time-varying delay. *IEEE Transactions on Fuzzy Systems, 20*(6), 114–119.

Wang, Z., Yang, L., Cui, S., & Wang, J. (2021). In-queue priority purchase: A dynamic game approach. *Queueing System, 97*(3), 343–381.

Yazdani, A., Dashti, S. F., & Safdari, Y. (2023). A fog-assisted information model based on priority queue and clinical decision support system. *Health Informatics Journal, 1–21*. https://doi.org/10.1177/14604582231152792

Cite this article as:
Adeleke I. A., Gbadebo, A. D., & Oyebode, E., O., (2024). Packet management system in single-server queue networks with Poisson arrivals using a Treap-based model. *FNAS Journal of Mathematical Modeling and Numerical Simulation, 2*(1), 1-13.