# Indian Institute of Technology Mandi

---

## A Simulation Study of Minimal Input Based AQM Deployment in Internet Routers

---

### Project Report


***Supervised By:*** Dr.Sreelakshmi Manjunath

***Submitted By:*** Mahima Gupta(T22055)

# Contents

# 1 Abstract


Internet assistance has become an integral part of our day-to-day lives from using web applications to streaming videos. In this era of perpetually growing Internet consumers, the network also needs to evolve to comply with the increased users. Transmission Control Protocol is being used for reliable end-to-end delivery of packets and managing congestion as it responds to congestion feedback(packet loss or an estimate of queueing delay). Various TCP variants have been developed and upgraded over time for faster recovery mechanism. For comparative analysis of TCP variants, refer [20].

However, simply relying on just TCP feedback mechanism does not suffice for an effective congestion control mechanism that could prevent congestion collapse. Efficient Queue Management Techniques are also required for handling bottleneck links within the network to manage traffic. Currently, the Internet is deployed with the classic droptail queues, which do not work well for high-speed networks[21].

It has been shown that implementing AQM on a single router often has a positive impact on other, non-AQM routers [22]. In this study, we propose the appropriate centrality measure for selecting router for AQM deployment. We have studied the behavior of TCPNewReno and TCPLinuxReno with droptail queues; Along with droptail queues that are implemented with threshold based queue policy. Threshold policy deterministically drops packets if the queue size is greater than the threshold($q_{th}$). To simulate the implementation, we are using ns3 simulator.

In this study, we started simple, with simulating simpler topologies like single bottleneck Topology, Parking Lot topology. Later, studying more complex topologies like multibottleneck topology, a representation of an autonomous system within the Internet. For Analysis, we are tracing Queue Size, Throughput, Packet Loss, Utilization and Latency. We are comparing DropTail Queues against Queues implemented with threshold policy. Threshold, $q_{th}$, has been set to 100 and 15.

When routers are implemented with threshold based queue policy, Queue size goes up only to the defined threshold due to which the selected router deployed with threshold policy (R6) experiences higher packet loss. However, the reduced latency makes up for the packet loss. Therefore, **throughput and link utilization remains uncompromised**

# 2 Objective

---

Objective of the project is to implement threshold based queueing policy in minimal number of AQM capable Internet Routers for congestion control. A multiple bottleneck network has been chosen which represents an autonomous system within the Internet. The project has been strategically divided in two segments -

***Segment 1*** : Learning ns3 tool. Build different network topology including single bottleneck topology, parking lot topology and multiple bottleneck Topology. Tracing system oriented and user oriented metrics. Writing master script to run the simulations

***Segment 2*** : Implementing source-based routing for flow dynamics in Multiple bottleneck Topology. Writing scripts to calculate latency and compute mean & standard deviations for all metrics. Determine router for AQM Deployment using Katz Centrality. Implementing Threshold Policy. Include Heterogeneous traffic in multi bottleneck and Obtain Results.

Segment 1 has been completed - July'23 to Nov'23. Manuscript Writing is in progress. Segment 2 has been completed - Dec'23 to May'24.

Timeline and along with task description has been specified below in the gantt chart.

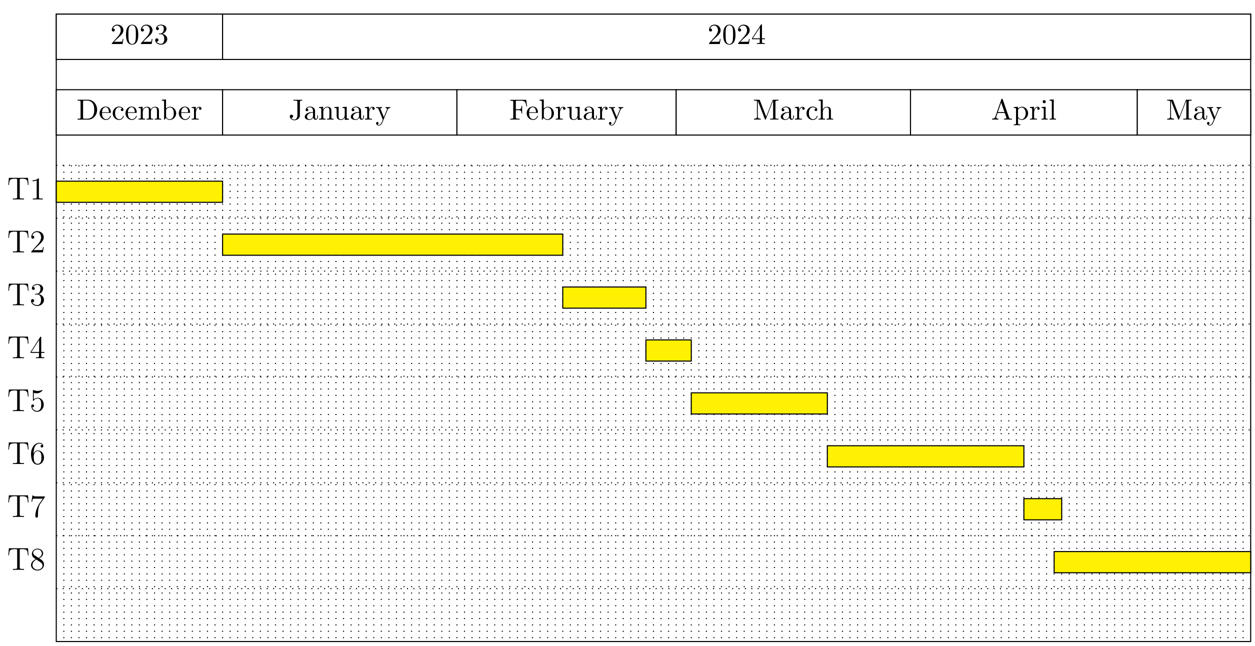



Figure 1: Segment 02 Work Timeline Gantt Chart

***Task 1:*** Implement Multiple Bottleneck Topology as per the given flow matrix.

***Task 2:*** Implement Source based routing for flow dynamics

***Task 3:*** Writing scripts to calculate latency, compute mean & standard deviations for all metrics.

***Task 4:*** Determine router for AQM Deployment using Katz Centrality. Implement Threshold Policy with $q_{th}$=100 & $q_{th}$=15.

***Task 5:*** Collecting Results and Analysis.
***Task 6:*** Implement heterogeneous traffic within the multiple bottleneck topology.
***Task 7:*** Collecting Results and Analysis.
***Task 8:*** Manuscript Writing.

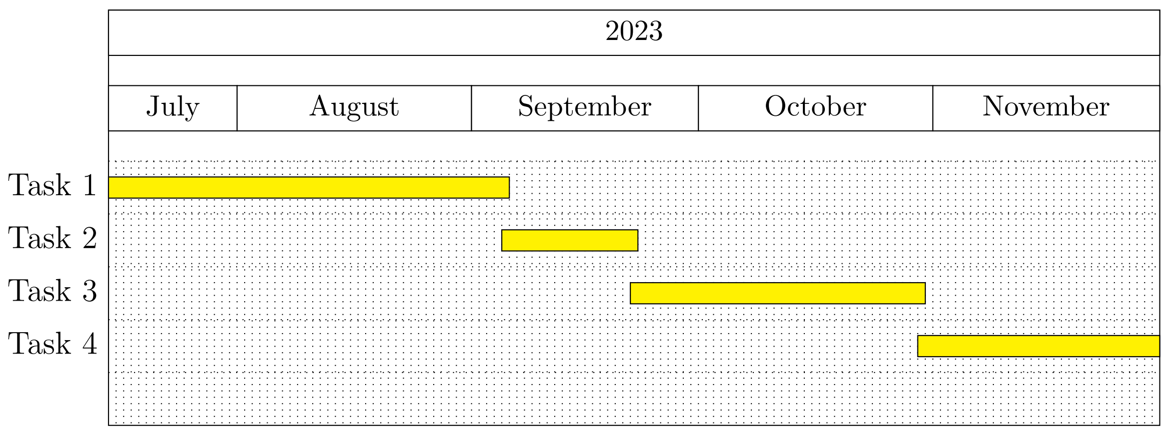



Figure 2: Segment 01 Work Timeline Gantt Chart

***Task 1:*** Understanding Ns3 and Simulating a Single Bottleneck Topology

***Task 2:*** Simulating Parking Lot Topology

***Task 3:*** Simulating a Multi Bottleneck Topology from a given Adjacency Matrix. Write scripts for running the full factorial simulations on both topologies.

***Task 4:*** Collecting Results and Analysis. Complete Pending tasks.

# 3 Introduction

Lagging webpages and loading buffer signs on streaming videos/audios has become a common problem. This increasingly pervasive issue occurs due to congestion in the network. Congestion is responsible for degradation of performance of the network, causing disruption in streaming/surfing experience. To know congestion, the first step is to understand the concept of bufferbloat. Bufferbloat is caused by general over-buffering in router queues that hold traffic that cannot be immediately forwarded. These queues are deep but not actively managed.[1] The term 'bufferbloat' refers to buffers being bloated or becoming enlarged beyond expectation. Bufferbloat leads to high queuing delays and delay variation which in turn leads to congestion in a network.[2]

All the packets that will arrive in the network after bufferbloat will be rejected as there is no space in buffers to store these packets. Incoming packets will be dropped until some of the packets currently residing in buffer are serviced. This leads to congestion on that link. Router will have to take some corrective measures to handle the situation. It will redistribute its loads to neghboring routers and redirect the incoming packets to other routers. This will ultimately lead to congestion in nearby links too as they will be receiving influx more than their serving capacity. Furthermore, **this cascaded redirection of load results in congestion collapse of the network**. The network succumbs to a state of inefficiency and reduced throughput.

In 1985, John Nagle identified the problem of congestion collapse and described it as [4] -

> *When a new bulk transfer begins and starts filling a large window. Should the round-trip time exceed the maximum retransmission interval for any host, that host will begin to introduce more and more copies of the same datagrams into the net. Eventually all available buffers in the switching nodes will be full and packets must be dropped . The round-trip time is now at its maximum . Hosts are sending each packet several times. This is congestion collapse.*

AQM was developed for effective end-to-end congestion control. AQM improves network performance in terms of delay, utilization and system fairness. The basic idea behind AQM is that it will notify the network about impending congestion, allowing sources to reduce the incoming data rate before the router queue overflows.

To understand congestion control and AQM, we need to delve into TCP dominance. Behavior of AQM is dependent on the behavior of TCP Variant. TCP is responsible for reliable communication over the network but to ensure reliability, it often leads to increased delay and reduced network throughput. Internet traffic has been dominated by TCP [5] that responds to network congestion in an **Additive Increase Multiplicative Decrease** (AIMD) manner. Newer variants like TcP BBR (Bottleneck Bandwidth and Round trip time) work with MIMD (Multiplicative Increase Multiplicative Decrease). We are going to implement threshold policy for TCP flavors that exhibit AIMD behavior. Since TCP is a closed feedback mechanism, it responds to packet loss and delay. Using these properties, a mathematical model has been developed that discusses the local stability of the network. *It is well known that for droptail queues,local stability of the network fluctuates when RTT is increased*. However, it has been proved that **local stability of the networks remains unchanged even when RTT is increased for queues implemented with threshold policy**[17].

In this study, we are analyzing the impact of droptail queues in a network versus threshold policy deployed within AQM routers in a network. With the current results, it has been observed that after implementing threshold policy, packet loss increases at the selected router. Latency decreases for routes taken by destinations attached at the selected router. Selected router also experience maximum utilization as it is serving some packet all the time. However, droptail has been still dominating the internet routers. As deploying AQM algrorithm is a heavy duty task in itself. Tuning the network parameters for AQM algorithm is a tedious task. Currently, there is very limited research on implementation of AQM Policies in multi-bottleneck networks. In this paper, we are presenting an approach to how minimal number of AQM routers need to be deployed with threshold policy in order to avert congestion in *multi bottleneck network*.

In the following section, various AQM Algorithms have been explained along with their drawbacks. We will also be exploring reasons why these algorithms have not been deployed yet. Section V elaborates on the methodology used for building various network topologies in ns3 and approach to the problem statement. All the graph plots and results have been added in Section VI. Analysis for Results along with inferences & conclusions from the study in Section VII. Future work that needs to be done has been described in Section VIII.

# 4 Literature Review

In this section, we will be discussing studies which collectively unveil the underlying reasons why despite the array of different AQM Algorithms, droptail has been still dominating the internet routers. Fejes[11] argues that when dimensioning router buffers in networks where connections with different Congestion Control Algorithms coexist, the right choice of AQM is more important than the size of the buffer itself. Let's focus on the various algorithms that fall under the umbrella of AQM.

*Random Early Detection* (RED), one of the most well known AQM algorithms, family1 router queue management mechanisms use the average queue length to compute drop probability[6]. RED decides whether the incoming packet should be enqueued or dropped. It marks packets with probability proportional to the current average queue length. At each arriving packet, if average queue size is larger than maximum threshold packet is marked. And if the average queue size is in between minimum and maximum threshold packet is marked with a probability. RED depends on calculating the average queue size using the Exponential Weight Moving Average (EWMA) and calculating the packet-marking probability.[7] RED was proposed in 1993, even though the term "Bufferbloat" was coined in 2011, solves the problem of bufferbloat.

*Random Exponential Marking*(REM), decouples congestion measure from performance measure [8]. The congestion measure indicates excess demand for bandwidth whereas the performance measure indicates queue length and delay. Calculations for congestion measure and marking probability are done differently in REM, as compared to RED. REM maintains a variable called price which is used to determine the marking probability. *Price* is a weighted sum of two parameters, namely, rate mismatch (difference between the input rate and link capacity) and queue mismatch (difference between the current queue length and target queue length). Price will be positive if the input rate exceeds the available bandwidth or the buffer occupancy is above the target queue length. To measure the amount of congestion along a path, REM uses the sum of link prices along that path. [8]

*Proportional Integral Controller*(PI) computes the packet dropping probability based on the deviations of the current queue length from a reference value and a queue length history. Recently, an alternate AQM scheme, Proportional Integral controller Enhanced (PIE) uses a probability to drop packets at enqueue, based on the estimated level of congestion. The congestion is used to increase or decrease p, based on the deviation of the current queuing delay from a target delay. Queuing delay is calculated from the queue length and the packet departure rate.

*Controlled Delay* (CoDel) drops a departing packet if the queuing delay exceeds a predefined target delay for a time period called interval. [2] CoDel design goals introduce the notion of "good queue" and "bad queue". CoDel monitors the queuing delay of packets in the router.Once CoDel enters to the dropping state it maintains a count that is increased by one after every drop. CoDel autotunes the dropping distance by dividing the interval by square root of the count. The next drop is scheduled to occur when the current time is past the dropping distance. When the delay falls below target, CoDel leaves the dropping state.[10]

Now, that we understand the basic idea behind the implementation of above mentioned AQM algorithms, we need to understand the drawbacks of them to realize why these still haven't been deployed in Internet Routers. The major drawback of RED is that it is difficult to configure the network dependent parameters. It is also insensitive to busty traffic.

Studies[9] show that REM has low throughout for web traffic and shows inconsistency with TCP Sender Mechanism. Even though REM can outperform Droptail schemes, it need careful configuration of parameters, especially target queue size. CoDel performance will not scale when the number of flows increases.[10]. It can be said that **major challenge with AQM algorithms is network parameter tuning**. Too small a value can increase loss for applications with a low RTT. However, increasing the delay target may significantly impair performance for interactive applications with a high path RTT. Since most routers are unaware of the actual path RTT between end hosts this requires a trade-off that depends on the traffic dynamics at the queue.

The work in [12] presents a tool to detect AQMs deployed in Internet routers. There are results to show that DropTail is still widely used. [13] and [14] show that researchers are working on developing tools to detect AQMs. Therefore, we are trying to devise approach to strategically apply AQM to minimal routers to avert congestion.

# 5 Methodology

---

Packet level Simulations have been implemented using NS3 (Network Simulator **version 3.36**)In initial stages, single bottleneck topology and parking lot topology was studied.

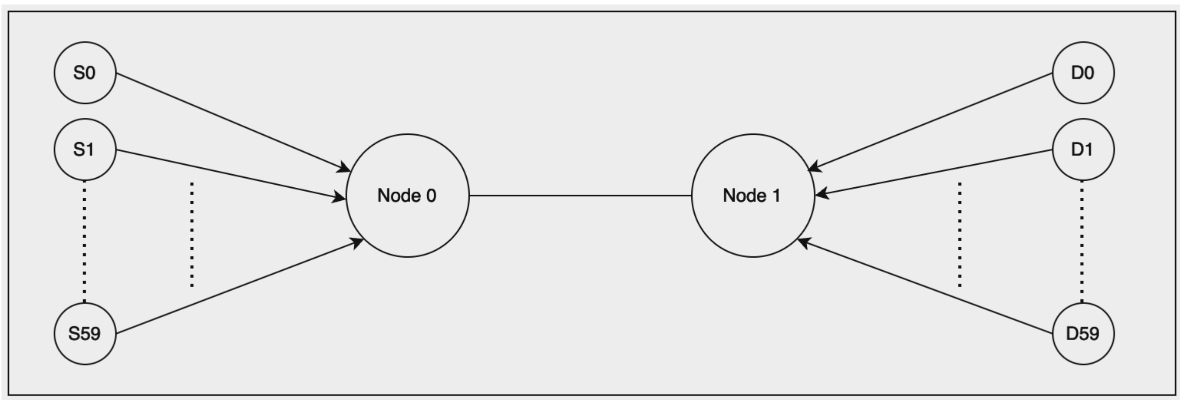



Figure 3: Single Bottleneck Topology

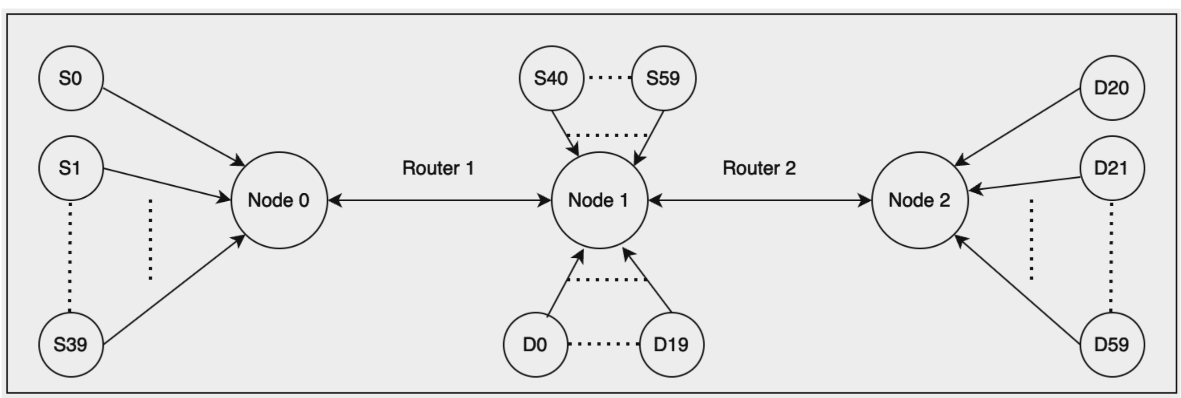



Figure 4: Single Bottleneck Topology

Following metrics were traced for above mentioned topology -

- Queue size
- Packet Loss
- Link Utilization
- Throughput

In networks, an autonomous system contains edge router, core router, source nodes, destination nodes, some outflows from the systems and a few intermediary nodes. Edge routers are the entry points for hosts connecting to their respective destinations. Core Routers are routers that operate on high bandwidth and forward the traffic to the edge routers at the other end. For multiple bottleneck topology,

- $N_S$ = Number of Sources = $N_S$ = Number of Destinations = 60
- $N_R$ = Number of Routers = 6

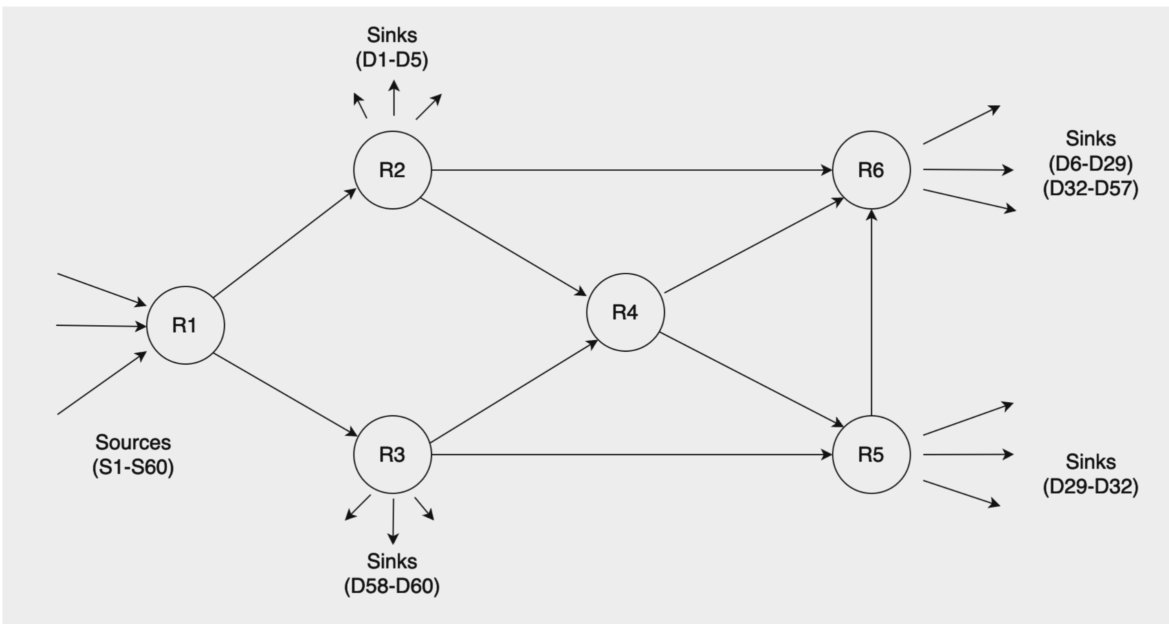



Figure 5: Multiple Bottleneck Topology

In this topology, R1 is the edge router. R2 and R3 are intermediary routers. R4, R5 and R6 are the core routers. For designing the multiple bottleneck topology, following flow matrix was considered -

$$Flow_{N_R \times N_R} = \begin{bmatrix} 0 & 0.4 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2 & 0 & 0.6 \\ 0 & 0 & 0 & 0.8 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.7 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0.9 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In the above matrix, each cell represent the percentage of traffic that flows from $R_i$ to $R_j$ where i=row number and j=column number. For less complexity, **single edge router with bandwidth higher than the core router has been considered**.

Capacities and Queue sizes for Routers are given below -

- Capacity of R1 = 100 Mbps, Queue Size of R1 = 2084 pkts
- Capacity of R2 = 40 Mbps, Queue Size of R2 = (2084 * 0.4)   834 pkts
- Capacity of R3 = 40 Mbps, Queue Size of R3 = (2084 * 0.4)   834 pkts
- Capacity of R4 = 60 Mbps, Queue Size of R4 = (2084 * 0.6)   1250 pkts
- Capacity of R5 = 60 Mbps, Queue Size of R5 = (2084 * 0.6)   1250 pkts
- Capacity of R6 = 60 Mbps, Queue Size of R6 = (2084 * 0.6)   1250 pkts

Source based routing has been implemented for following routes-

**Route 01**
S0 - S4: R1 − > R2 : D0 - D4

**Route 02**
S5 - S9: R1 − > R2 − > R4 − > R6 : D5 - D9

**Route 03**
S10 - S23: R1 − > R2 − > R6 : D10 - D23

**Route 04**
S24 - S28: R1 − > R3 − > R4 − > R6 : D24 - D28

**Route 05**
S29 - S31: R1 − > R3 − > R4 − > R5 : D29 - D31

**Route 06**
S32 - S52: R1 − > R3 − > R4 − > R5 − > R6 : D32 - D52

**Route 07**
S53 - S56: R1 − > R3 − > R5 − > R6 : D53 - D56

**Route 08**
S57 - S59: R1 − > R3 : D57− > D59.

For all network topology, following fixed parameters were considered-

- Simulation Duration = 250 seconds
- Access Link Bandwidth = 2 Mbps, Bottleneck Delay = 1ms
- Packet Size = 1446

**Choice of Router for AQM Deployment -**

Katz Centrality has been used for determining the most significant node. Katz Centrality measure for each router node is -

| | Router 2 | Router 3 | Router 4 | Router 5 | Router 6 |
|---|---|---|---|---|---|
| ***Katz Centrality*** | 0.14367 | 0.05410 | 0.09314 | 0.08970 | 0.38096 |

Since, **Router 6** has highest Katz centrality Measure. It also aligns with the intution that Router which serves maximum number of TCP flows should be most significant node. Therefore, Router 6 has been selected as router for AQM activation.

For the multiple bottleneck topology(refer to Fig.5), two types of traffic have been implemented -

- Homogeneous Traffic : In this traffic type, all the 60 source nodes have TCP Sockets which are sending FTP traffic. Data flows as per the flow matrix.
- Heterogeneous Traffic : In this traffic type, different types of sources have been used. There are 52 sources which have TCP sockets sending FTP Traffic over the network. Two different TCP variants have been implemented on these 52 TCP sources. 50% TCP sources are TCPNewReno and 50% TCP sources are TCPCubic. On the other hand, there are 8 sources which have UDP sockets sending HTTP Traffic over the network.

| **Variable Parameters** |
|---|
| ***Round Trip Time:*** |
| 10ms - 300ms with step of 05 |
| ***Queue Type:*** |
| DropTail Queues |
| Threshold Policy where $q_{th}$= 100 |
| Threshold Policy where $q_{th}$= 15 |

| **Metrics** |
|---|
| ***System Oriented Metrics*** |
| Queue Size |
| Packet Loss |
| Link Utilization |
| ***User Oriented Metrics*** |
| Throughput |
| Latency |

For calculating the latency, queuing delay has to be calculated.

$$QueuingDelay = \frac{QueueSize * PacketSize * 8}{Capacity * 10^6}$$

The above formula would give Queuing Delay in seconds, it has to be converted to Latency(ms). To Calculate Latency,

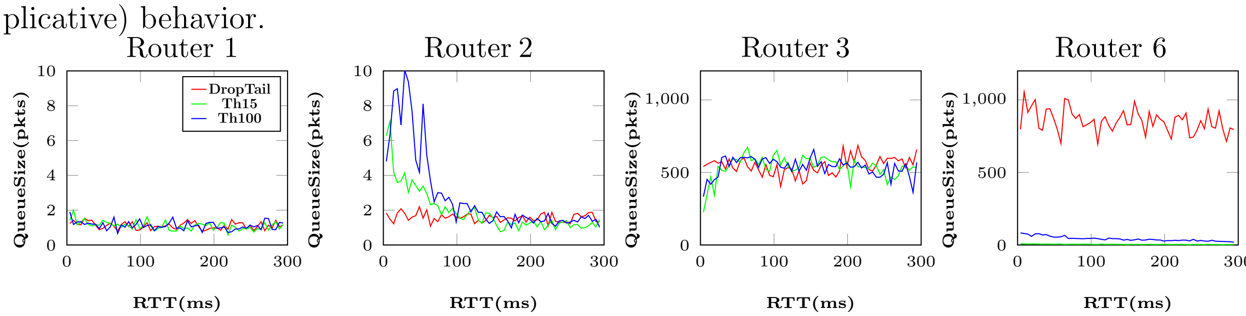
$$Latency = RTT + QueuingDelay$$

# 6 Results

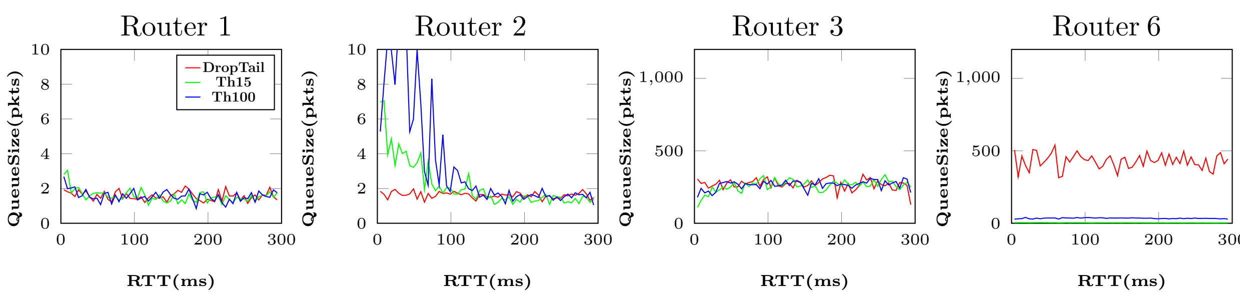
In this section, results for homogeneous traffic and heterogeneous traffic have been shown. Here, Round trip time refers to the constant portions of RTT i.e., propagation delay and transmission delay.

## 6.1 Heterogeneous Traffic

In Heterogeneous traffic, 13% of traffic are UDP sources and remaining 87% traffic is TCP sources. Amongst the 52 TCP sources, 50% sources are TCPNewReno and 50% sources TCPCubic. Both the TCP variants exhibit AIMD(Additive Increase Multiplicative) behavior.



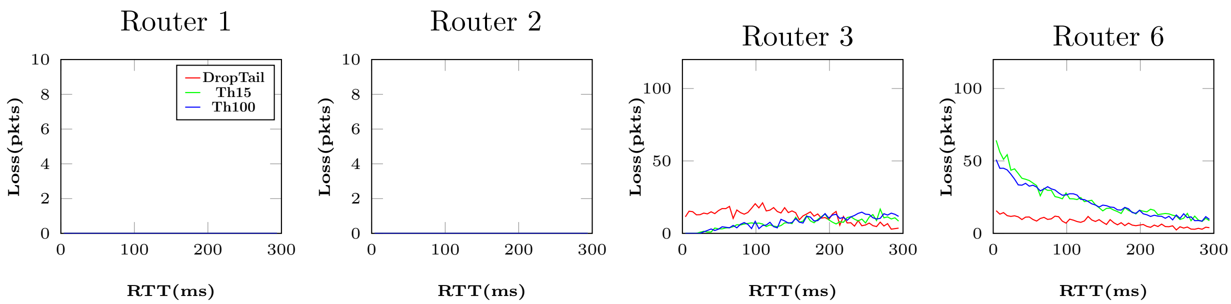

Results for Mean Queue Size



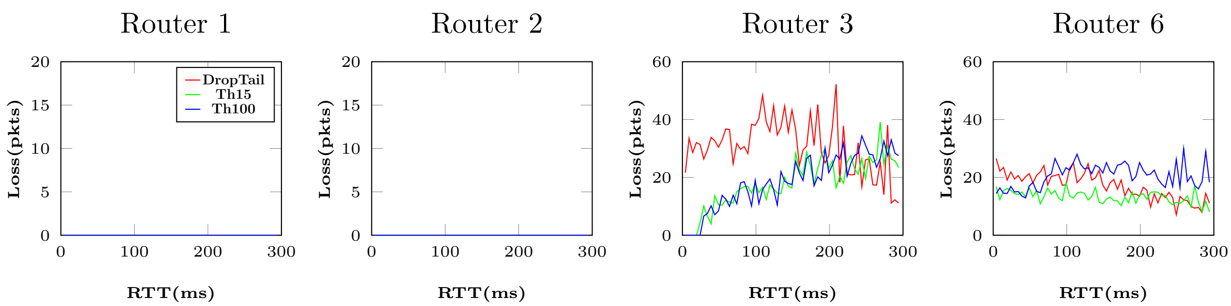

Results for Standard Deviation of Queue Sizes

It can be observed that -

- Mean Queue size for droptail queues is higher than queues implemented with threshold based queue policy. It indicates that when R6 was implemented with queueing policy, there was lesser congestion within the network.
- Queue does not build for Router 4 and Router 5 as their serving capacities are sufficient for handling incoming traffic.



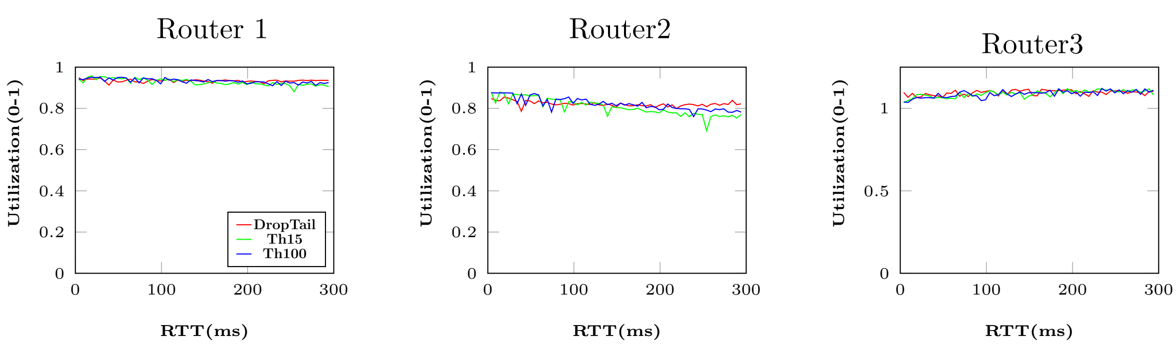
Results for Mean Packet Loss

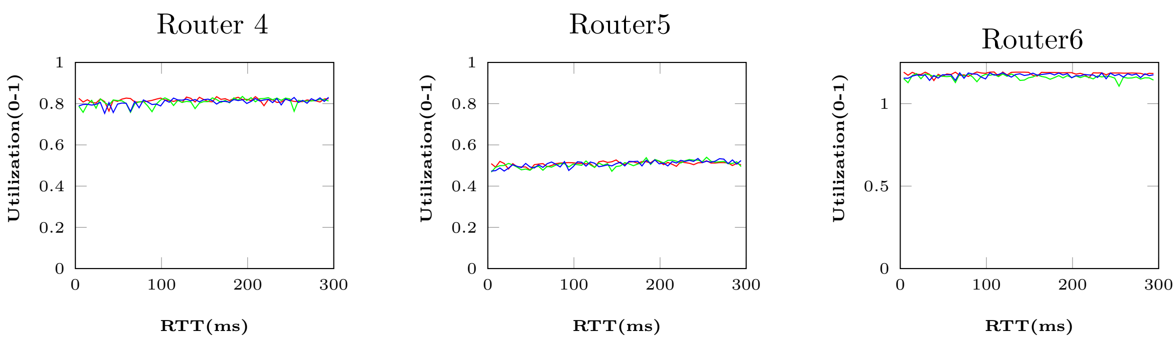


Results for Standard Deviation of Packet Loss

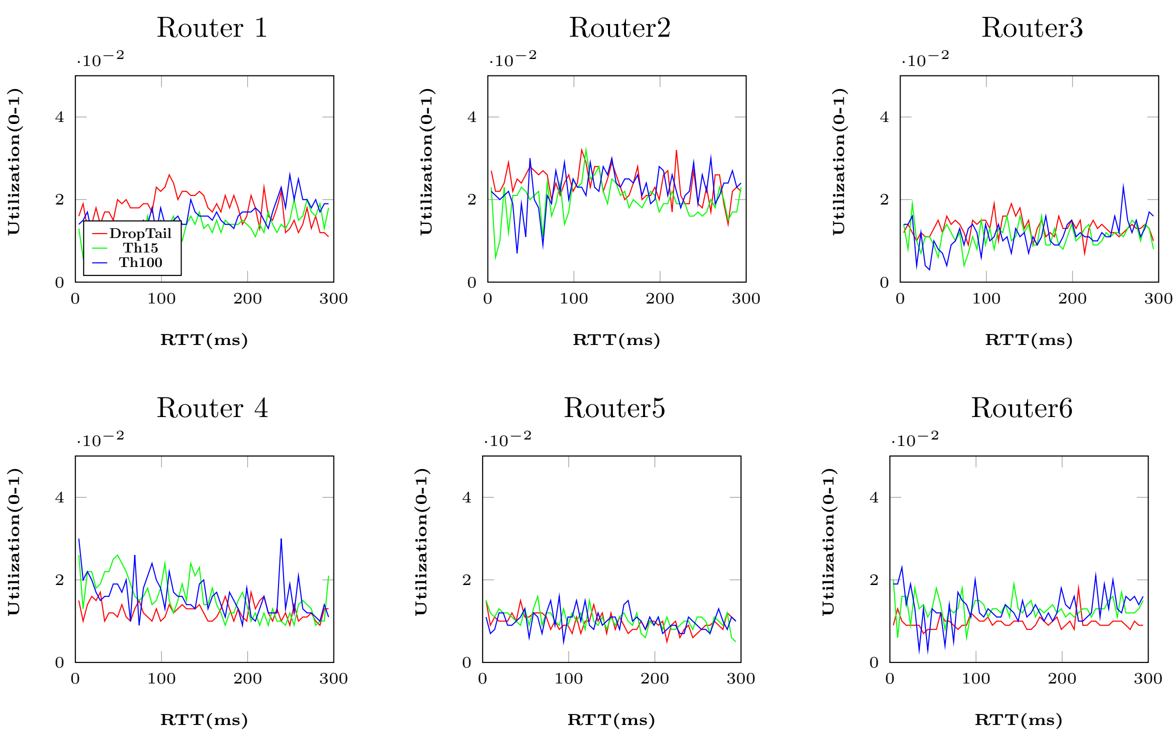
It can be observed that -

- Since the Queue sizes for Router 1 and Router 2 minimally, no packet loss has been observed for these routers.
- No packet loss for Router 4 and Router 5.
- Even though threshold policy has been implemented on Router 6, it does not undergo significantly high packet loss compared to droptail queues.





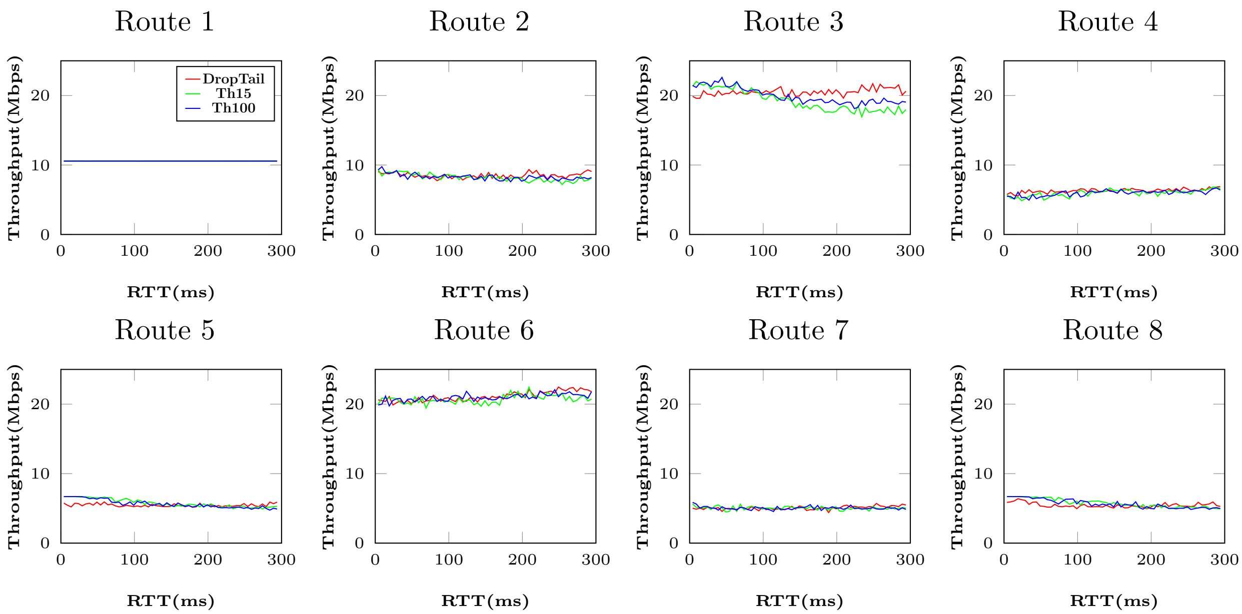

Results for Mean Link Utilization



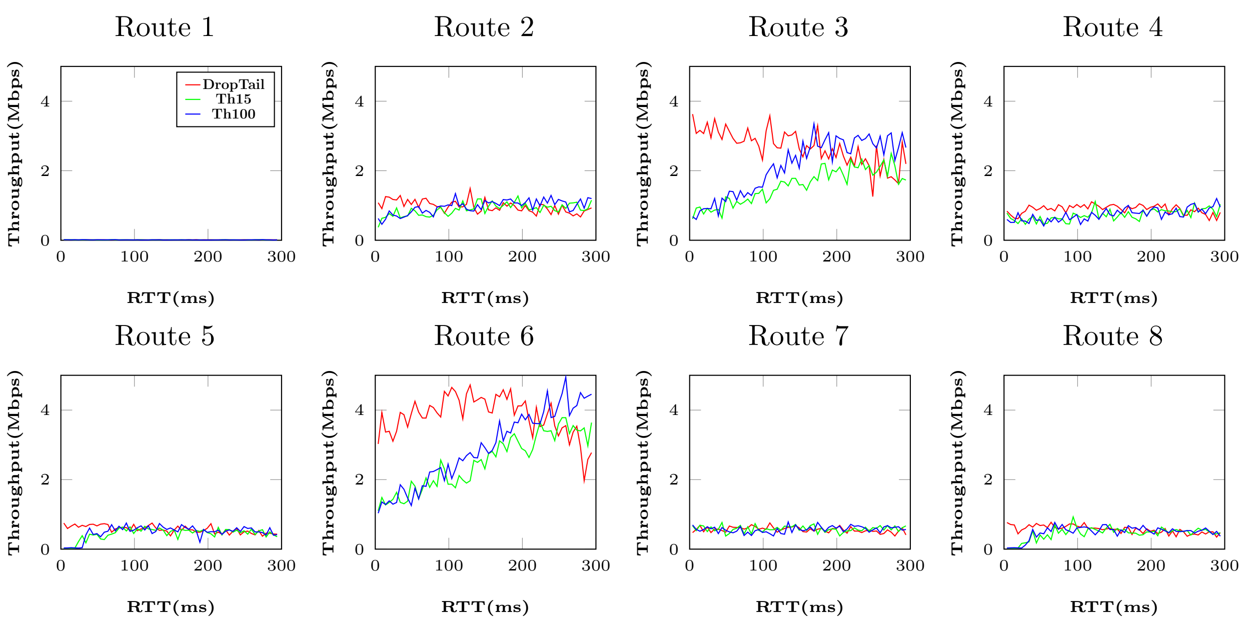

Results for Standard Deviation for Link Utilization

It can be observed that -

- Link Utilization for droptail queues is similar to the queues implemented with threshold policy.
- Even though there is comparatively higher packet loss in Router6, the link utilization is not impacted. Due to decreased queueing delays, Routers still stay busy serving packets.



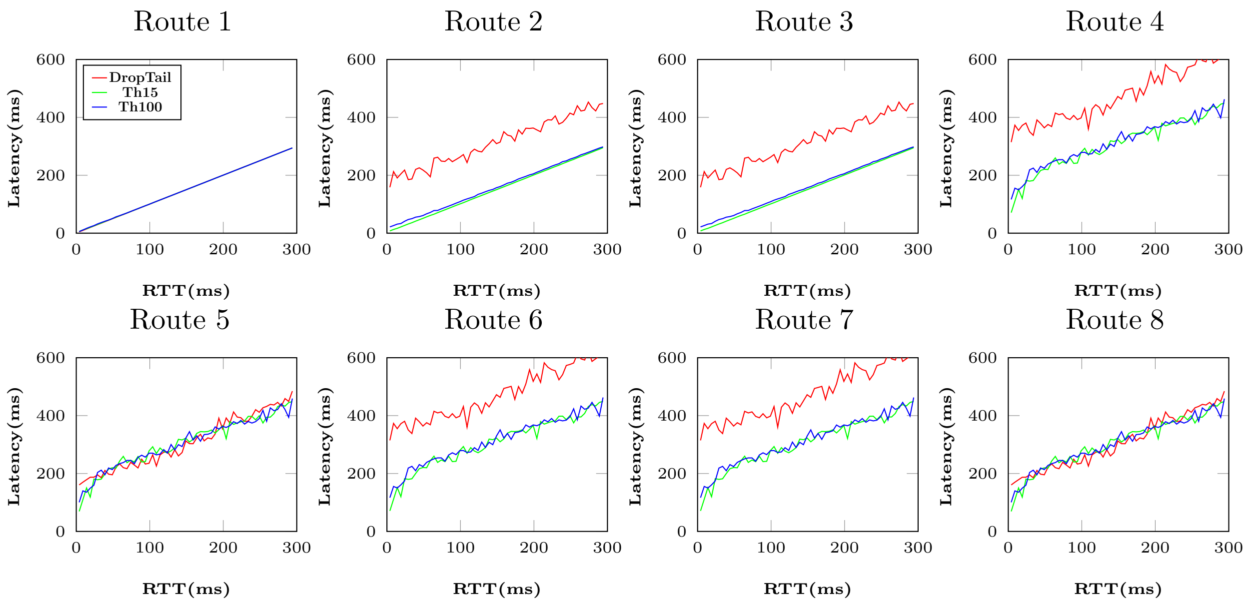

Results for Mean Throughput



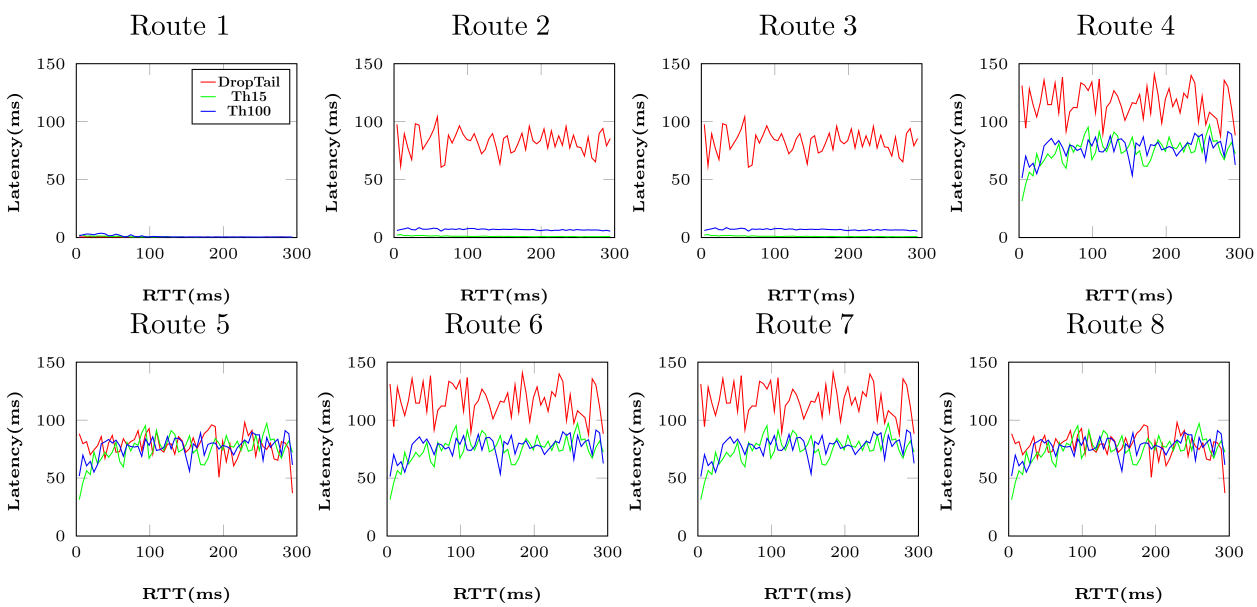

Results for Standard Deviation for Throughput

It can be observed that -

- Throughput for droptail queues is similar to the queues implemented with threshold policy. Lesser queueing delay makes up for the high packet loss.




Results for Mean Latency



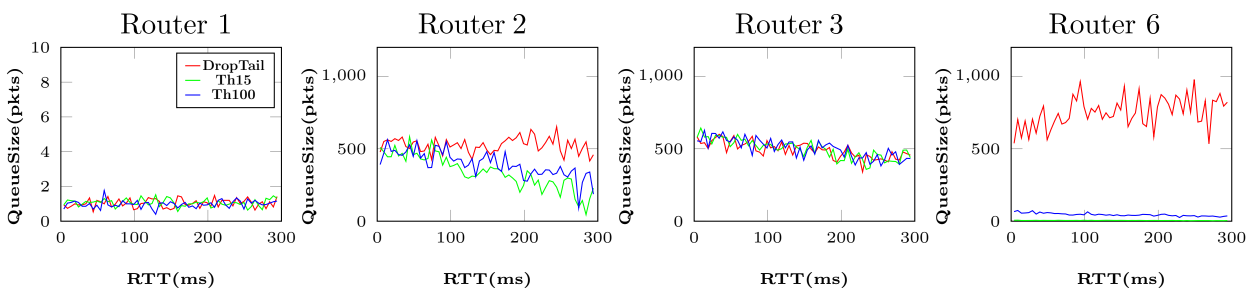

Results for Standard Deviation for Latency

It can be observed that -

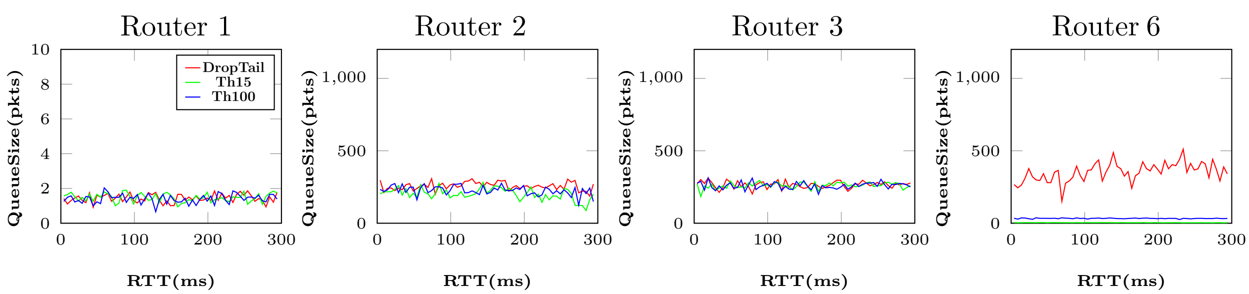
- Mean Latency for Queues implemented with threshold policy is significantly lower than the Mean latency for Drop Tail Queues.
- Routes which pass through Router 6 have lower mean latency for queues implemented with threshold policy. Whereas routers which do not pass through R6 are not impacted.

## 6.2 Homogeneous Traffic - TCPLinuxReno Variant

All the 60 sources have TCP sockets which sends FTP Traffic. TCP Variant - TCPLinuxReno follows AIMD behavior (Additive Increase and Multiplicative Decrease).



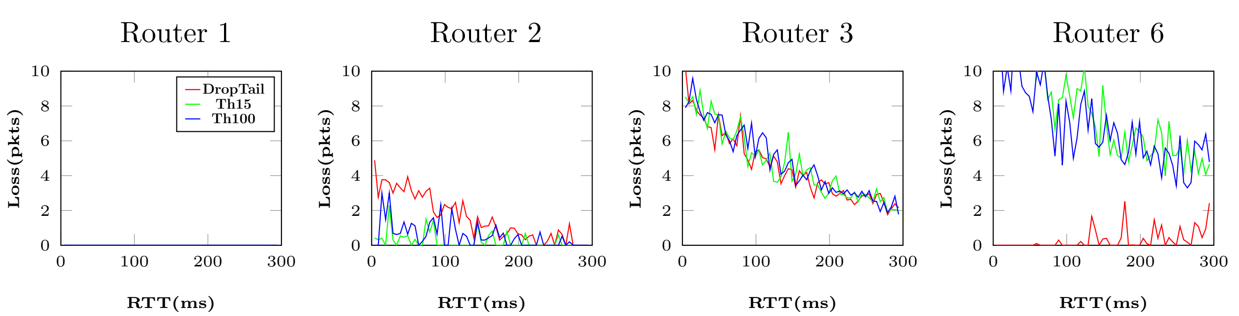

Results for Mean Queue Size



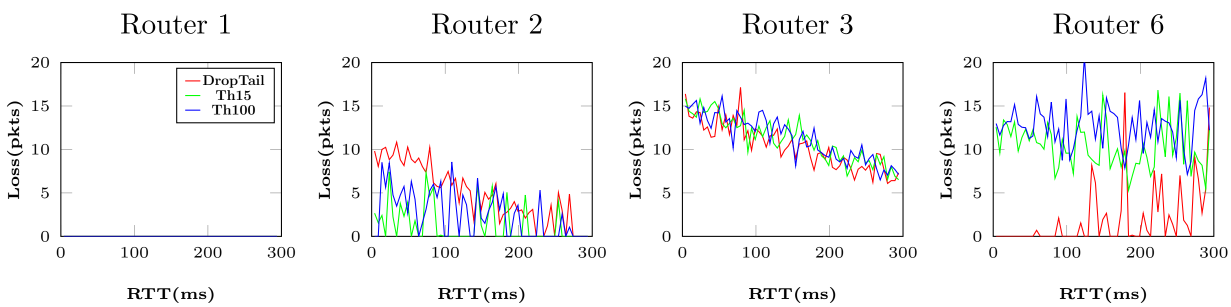

Results for Standard Deviation of Queue Sizes

It can be observed that -

- Compared to Heterogenous traffic where queue for R2 was building upto 10 packets, in homogeneous traffic purely TCP sources build queue for R2 upto 100 packets.
- Since Router 6 has been implemented with threshold policy, Queue for R6 builds only to few tens of packets unlike droptail queues.



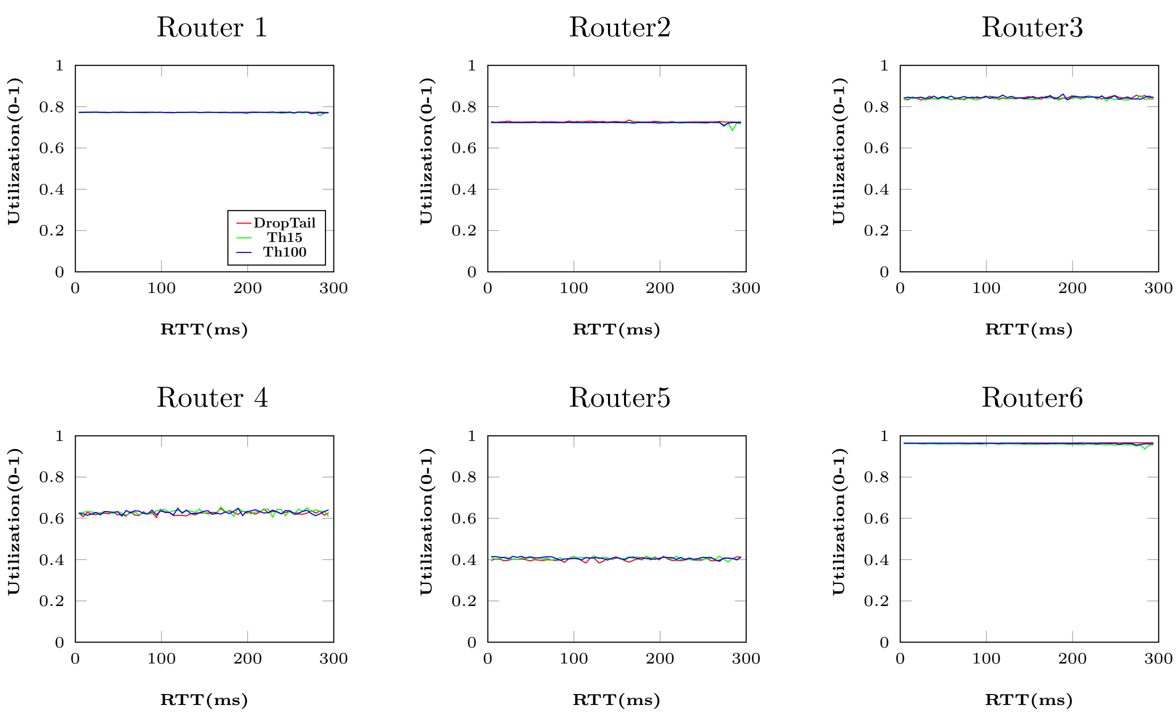

Results for Mean Packet Loss



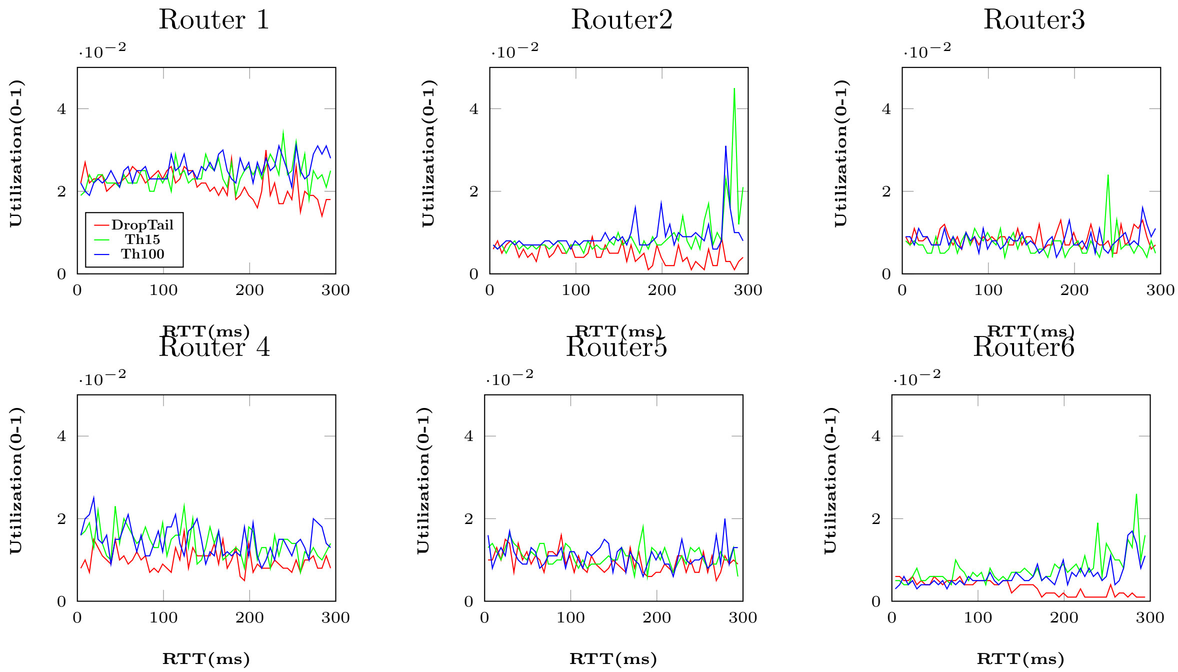

Results for Standard Deviation of Packet Loss

It can be observed that -

- Compared to Heterogeneous traffic, in homogeneous traffic setup Router 6 experiences higher packet loss.
- Since, in homogeneous traffic setup, queue is building for R2, it experiences higher packet loss compared to heterogeneous traffic.



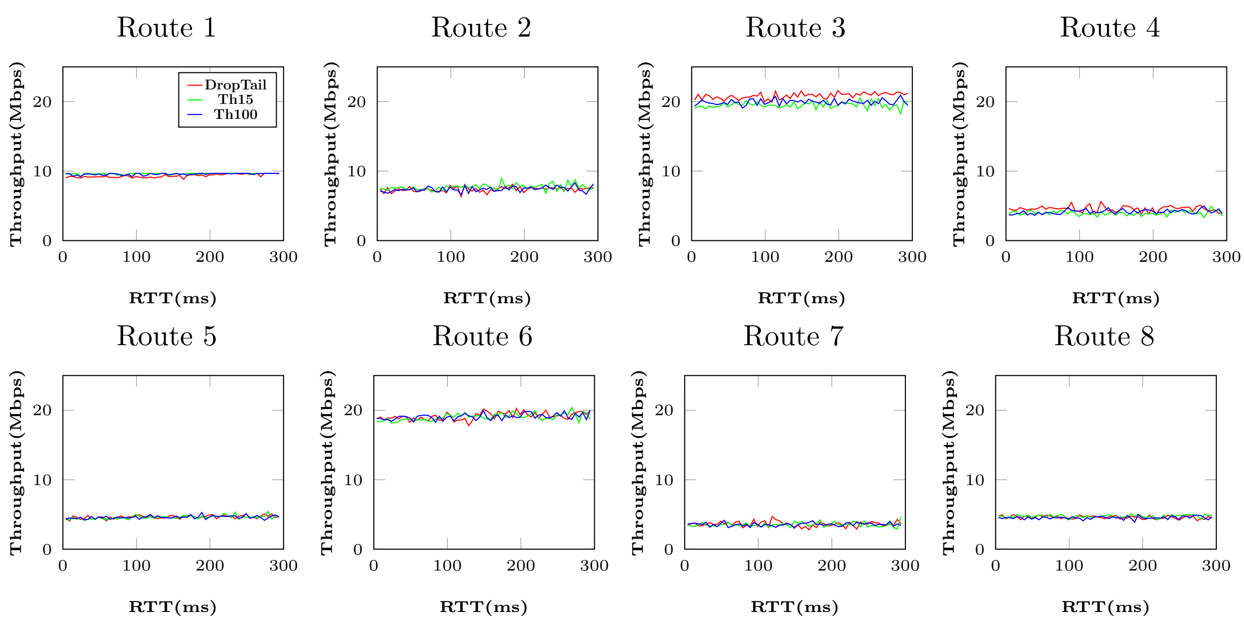

Results for Mean Link Utilization



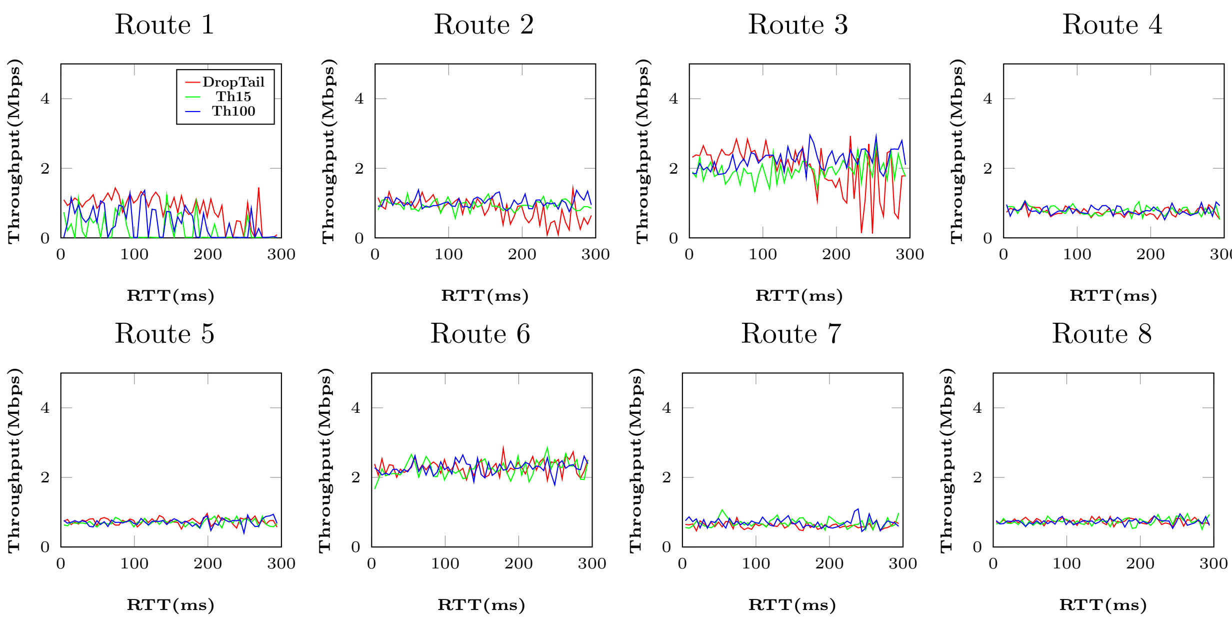

Results for Standard Deviation for Link Utilization

It can be observed that -

- Link Utilization for Queues implemented with threshold Policy is similar to the utilization for droptail Queues.



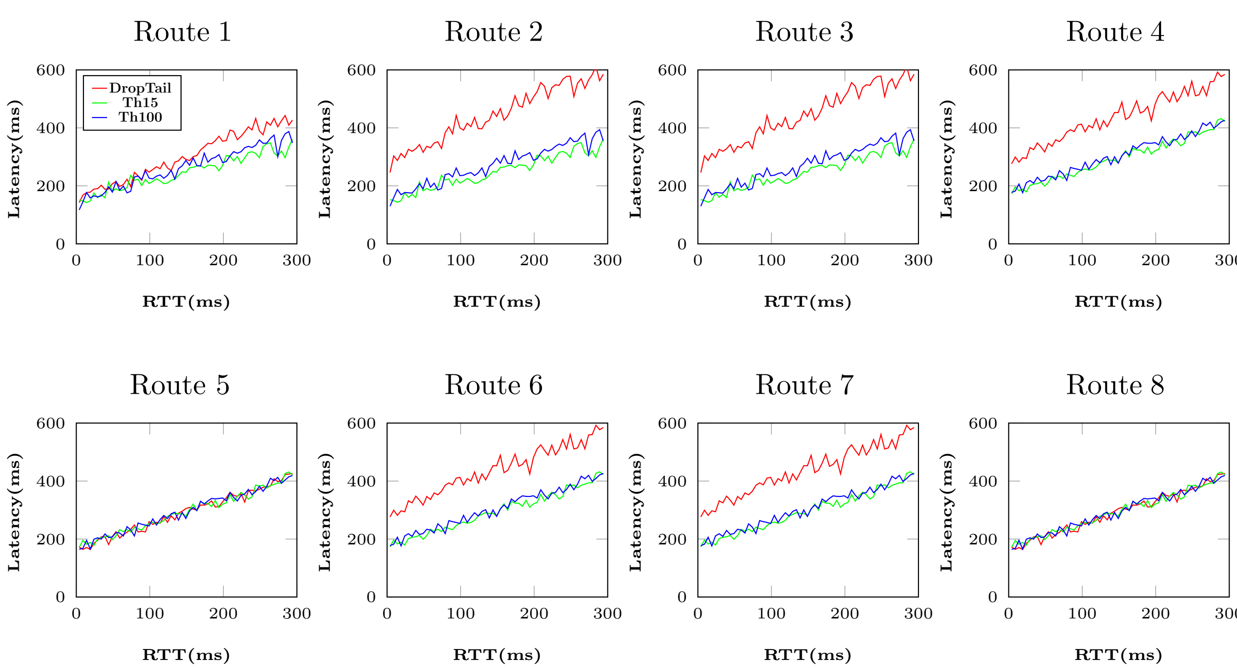

Results for Mean Throughput



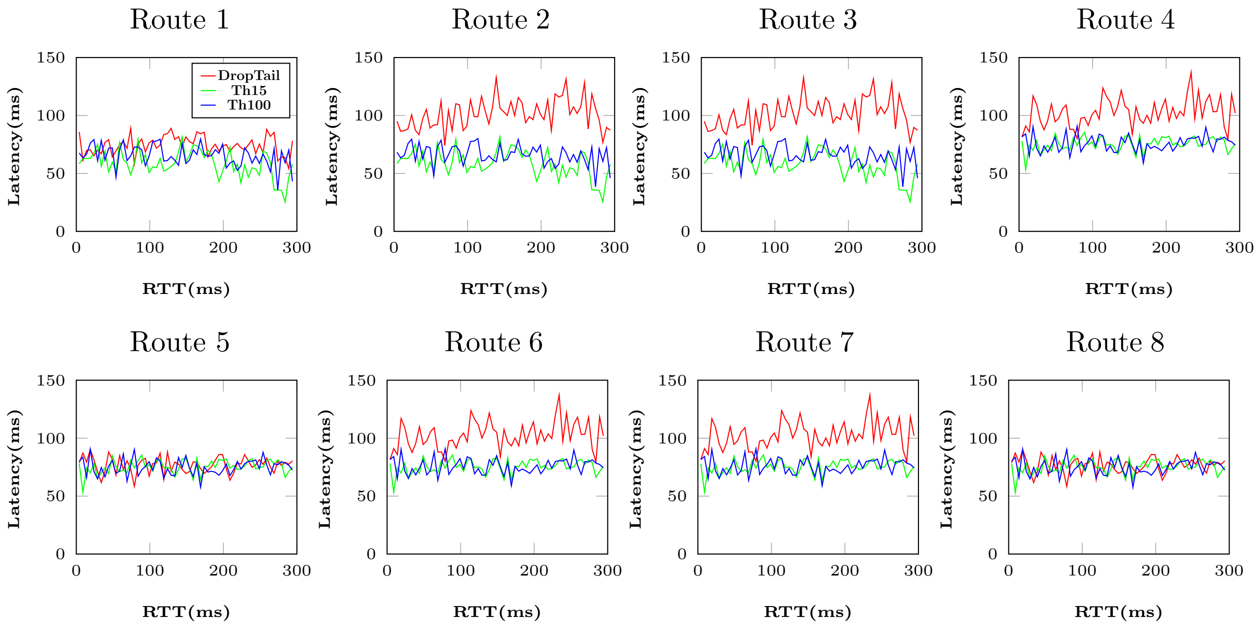

Results for Standard Deviation for Throughput

It can be observed that -

- Throughput for Queues implemented with threshold Policy is similar to the throughput for droptail Queues.




Results for Mean Latency



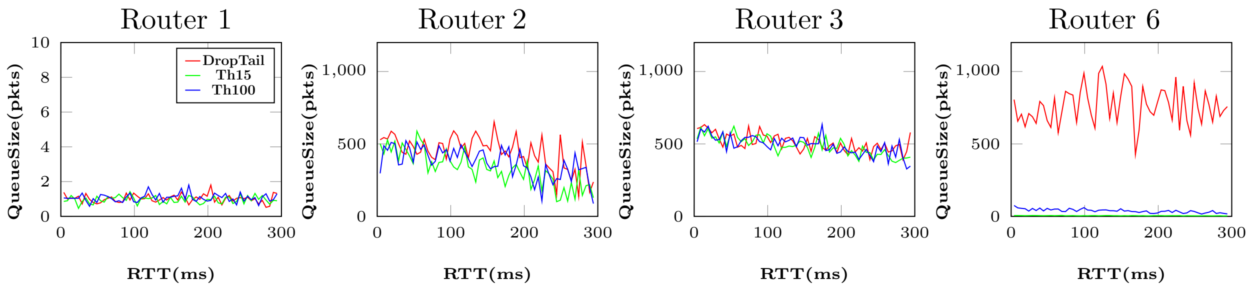

Results for Standard Deviation for Latency

It can be observed that -

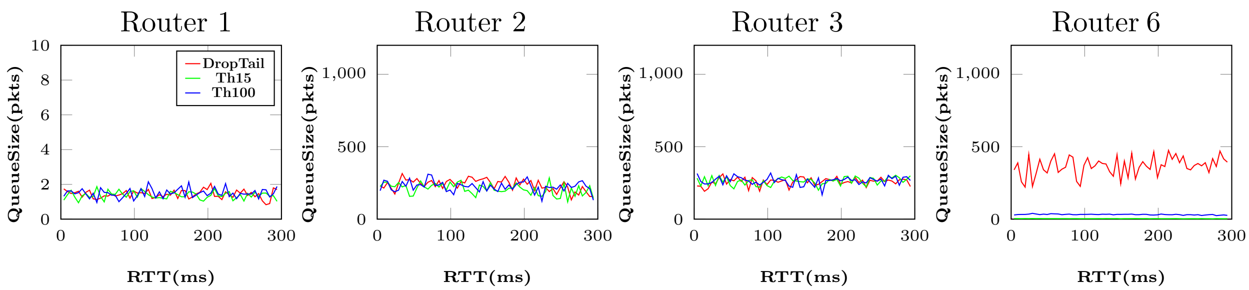
- Mean Latency for queues implemented with threshold policy is lower than the droptail queues.
- As the RTT increases, the Latency also increases as RTT is one of the components used in calculating latency.
- For homogeneous traffic, as the RTT is increases, the latency gap between threshold and droptail also increases.

## 6.3 Homogeneous Traffic - TCPNewReno Variant

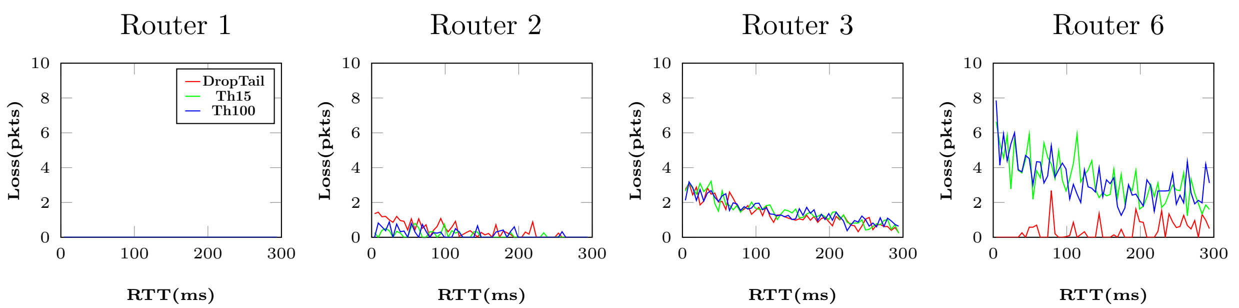
**Note** : Results for TCPLinuxReno are similar to that of TCPNewReno as both the TCP variants exhibit AIMD behavior for congestion control. Therefore, all observations made for TCPLinuxReno mention in previous subsection are relevant for this subsection as well. To avoid repetition, those observations have not been highlighted here.



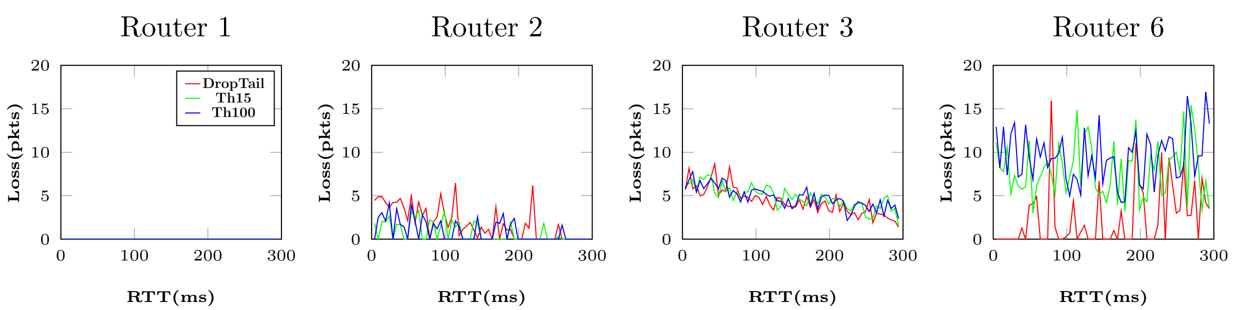
Results for Mean Queue Size



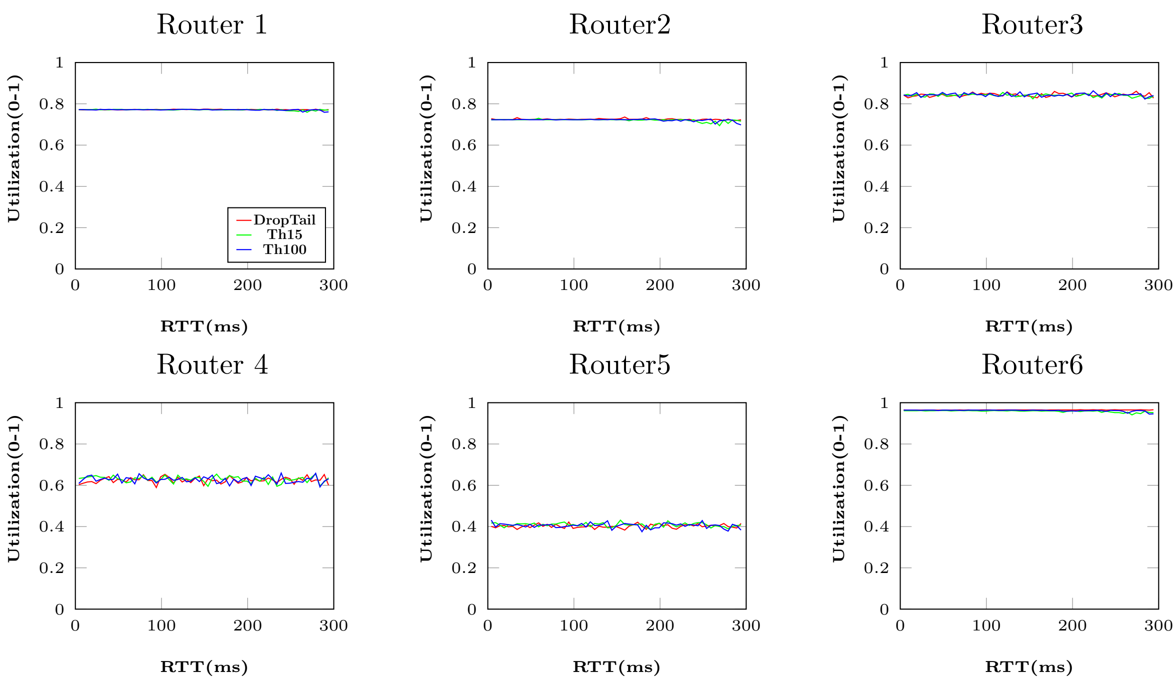
Results for Standard Deviation of Queue Sizes



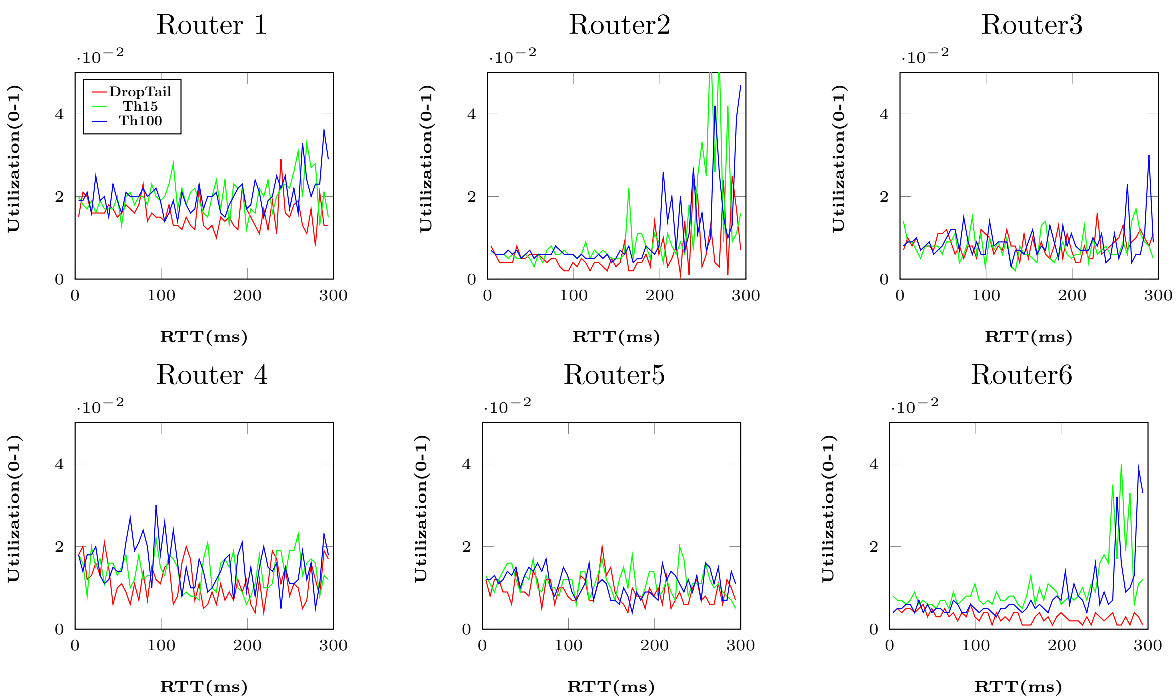
Results for Mean Packet Loss



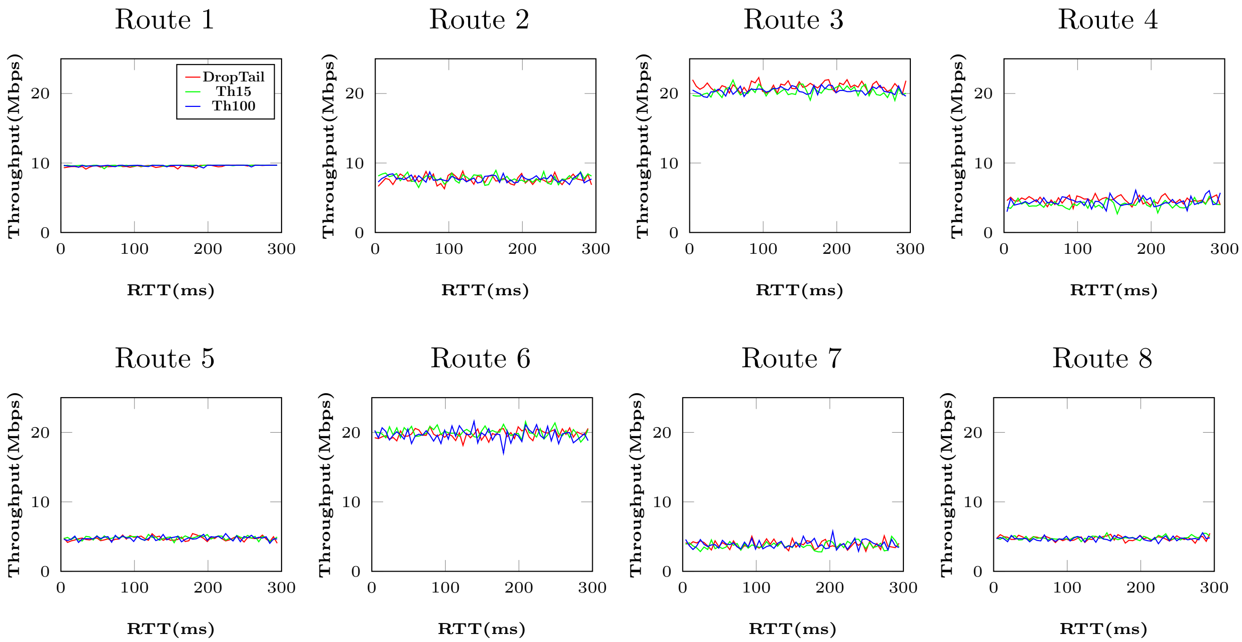
Results for Standard Deviation of Packet Loss



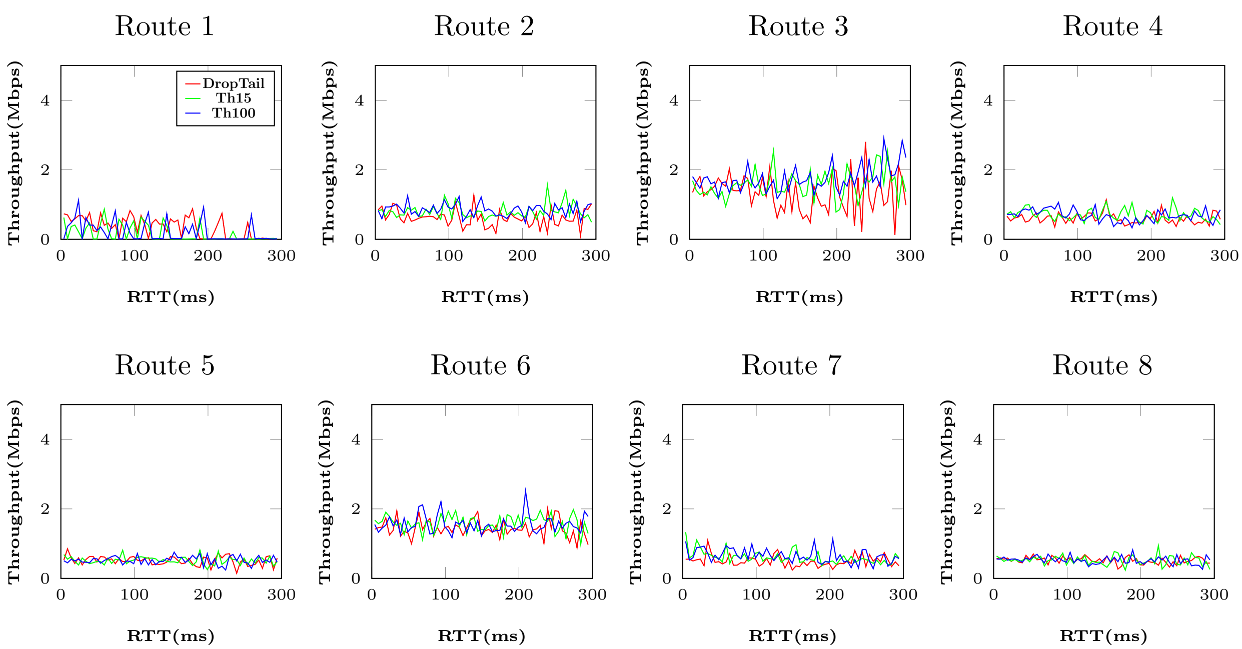

Results for Mean Link Utilization



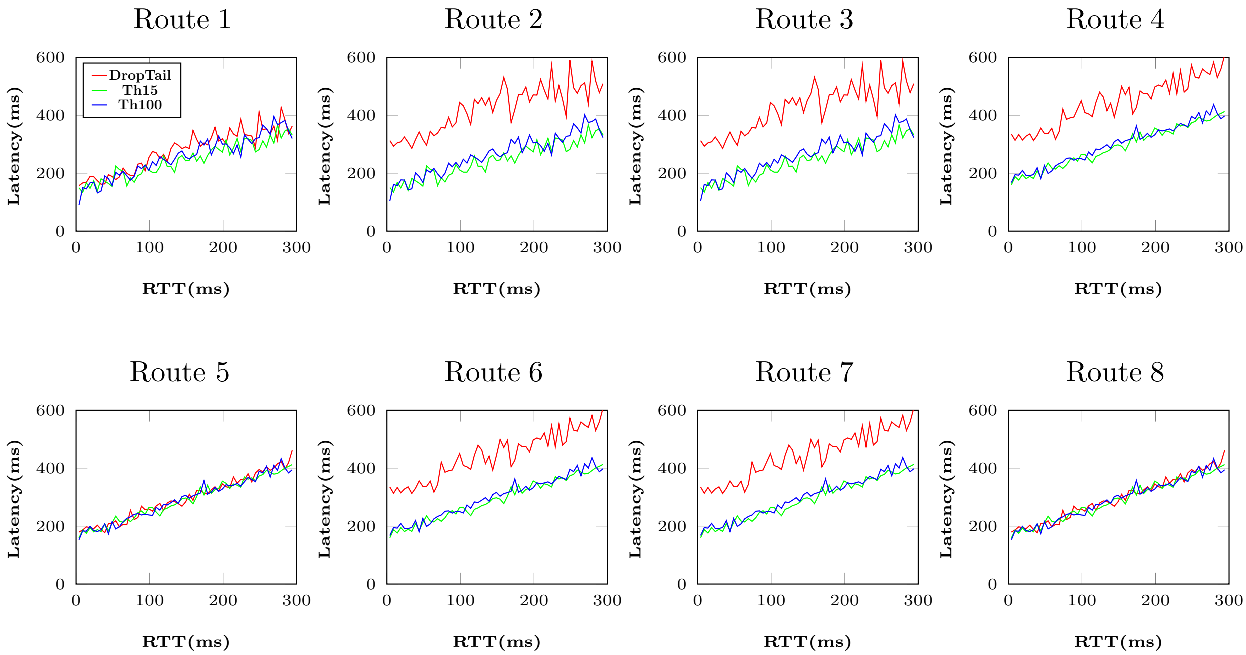

Results for Standard Deviation for Link Utilization



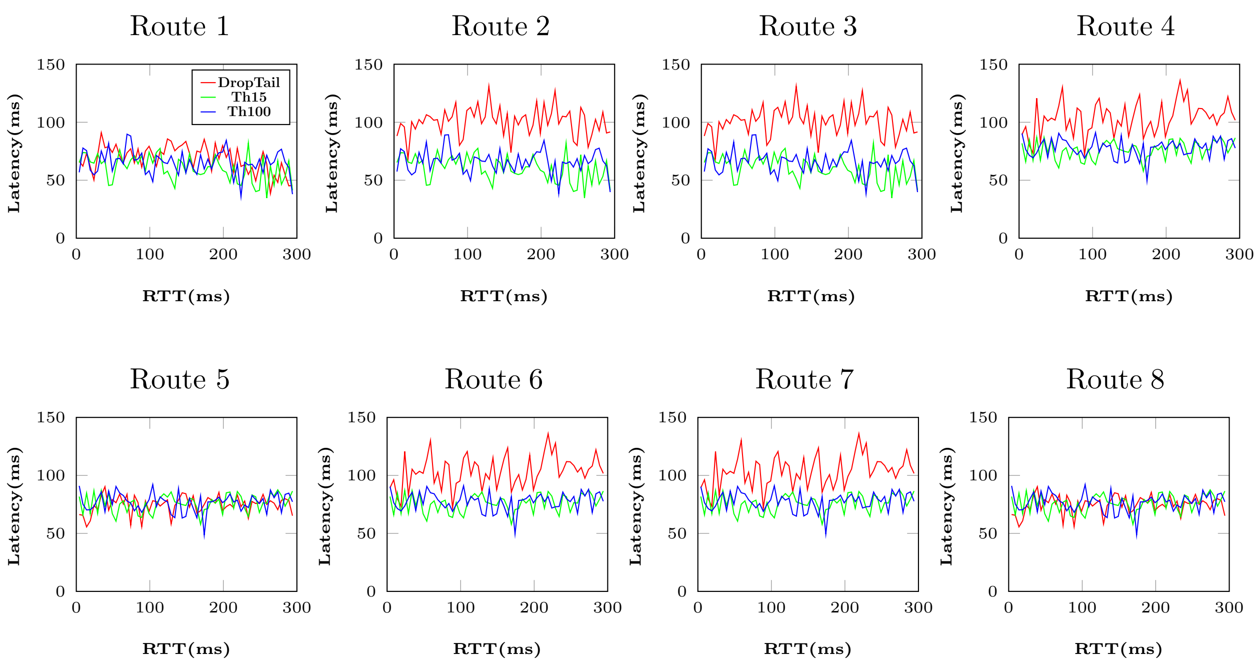

Results for Mean Throughput




Results for Standard Deviation for Throughput




Results for Mean Latency




Results for Standard Deviation for Latency

# 7 Conclusion

In initial stages, behavior of various TCP variants with reference to RTT and Droptail queues was studied. Different Network Topology (Single Bottleneck and Parking Lot Topology) were studied. It was found that as the number of nodes within the network increases, congestion also increases. For TCP Variants with AIMD behavior, queue sizes follows a saw tooth pattern with respect to time. TCP Variants following MIMD behavior such as TCP BBR was also studied. It was found that TCP Bbr does not follow a saw tooth pattern, rather queue size fluctuates with greater amplitude and goes up to zero as well. Therefore, it can be said that the *stability of the network varies with TCP Variant*.

In this study, we have worked with TCP variants exhibiting AIMD behavior as most routers in today's internet implement TCP variants with AIMD behavior. It has also been concluded that with these greater fluctuations in queue size causing instability, droptail queues must be replaced with a more efficient queueing policy. In this paper, we have provided the design guidelines for implementing one of the AQM policies i.e., Threshold based queue Policy.

Threshold based queue policy has been implemented in multiple bottleneck network. This policy has been implemented on only one router node, router 6, to cause minimal disturbance to the network. This node has been chosen using Katz centrality. We know that TCP is a closed feeback mechanism, it responds to packet loss or delay(dependent on variant). When Router 1 observes that queue size is increasing significantly for R2, R3 and R6 leading to greater queueing delays and packet loss, it reduces the incoming rate and therefore *Queue in R1 is not filling up much*. Higher packet loss at R6 when implemented with threshold policy as it deterministically starts dropping packets once the queue size reaches the defined threshold.

In a nutshell, we can say that when threshold based queue policy is implemented within a network, latency of the network reduces as compared to droptail queues. Even though, we observe higher packet loss, the reduced queueing delay makes up for it. Therefore, the performance of the network remains unaffected. Performance of the network has been measured in the terms of throughput and link utilization.

# 8 Manuscripts(under Preparation)

1. Mahima Gupta, Dipesh Sharma, Rijul Jain and Sreelakshmi Manjunath, "Centrality-based design guidelines for AQM deployment for congestion control in multi-bottleneck networks", to be submitted to ACM Transactions on Modeling and Performance Evaluation of Computing Systems.
2. Mahima Gupta, Gaurav Shukla, Dipesh Sharma, Rijul Jain and Sreelakshmi Manjunath, "Design guidelines of activation of AQM in RED-based Internet routers", under preparation, to be submitted to ACM Transactions on Modeling and Performance Evaluation of Computing Systems.

# References


[1] M. Allman, “Comments on bufferbloat,” *ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 30–37, 2013.

[2] Jiancheng Ye, Ka-Cheong Leung, Steven H. Low ”Combating Bufferbloat in Multi-Bottleneck Networks: Theory and Algorithms” *IEEE/ACM Transactions on Networking* Vol. 29, No. 4, August 2021

[3] V. G. Cerf, “Bufferbloat and other Internet challenges,” *IEEE Internet Comput.*, vol. 18, no. 5, pp. 79–80, Sep./Oct. 2014

[4] John Nagle ”Congestion Control in TCP/IP Networks” *Ford Aerospace and Communications Corporation Palo Alto, California*

[5] K. Thompson, G. Miller, and R. Wilder. Wide-Area Internet Traffic Patterns and Characteristics. *IEEE Network*,11(6):10–23, November/December 200

[6] V. Firoiu and M. Borden. ”A Study of Active Queue Management for Congestion Control” In *Proceedings of IEEE INFOCOM*, Mar. 2000

[7] Ayman EL-Sayed and Zeiad Elsaghir ”Enhanced Random Early Detection” *International Journal of Computer Applications* Vol.92 – No.9, April 2014

[8] Isha Tarte, Aparna R. Joshi, Navya RS and Mohit P. Tahiliani ”Implementation and Validation of Random Exponential Marking (REM) in ns-3” *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)* doi: 10.1109/ANTS.2017.8384131

[9] Uma R. Pujeri, V. Palaniswamy, P. Ramanathan, Ramachandra Pujeri ”Comparative Analysis and Comparison of Various AQM Algorithm for High Speed” *Indian Journal of Science and Technology* Vol.8 - No.35 December 2015

[10] Ilpo Jarvinen and Markku Kojo, ”Evaluating CoDel, PIE, and HRED AQM Techniques with Load Transients” *39th Annual IEEE Conference on Local Computer Networks* Edmonton, AB, Canada, 2014, pp. 159-167, doi: 10.1109/LCN.2014.6925768.

[11] Eduard Grigorescu, Chamil Kulatunga, Gorry Fairhurst ”Evaluation of the Impact of Packet Drops due to AQM over Capacity Limited Paths” *2013 21st IEEE International Conference on Network Protocols (ICNP)*, Goettingen, Germany, 2013, pp. 1-6, doi: 10.1109/ICNP.2013.6733658.

[12] Fejes, Ferenc, et al. ”Who will save the internet from the congestion control revolution?.” *Proceedings of the 2019 Workshop on Buffer Sizing* 2019.

[13] Bideh, Minoo Kargar, et al. "Tada: An active measurement tool for automatic detection of AQM." *EAI Endorsed Transactions on Self-Adaptive Systems* 2.8 (2016): 55-60.

[14] Baykal, Cenk, Wilko Schwarting, and Alex Wallar. "Detection of AQM on Paths using Machine Learning Methods." *arXiv preprint arXiv:1707.02386 (2017).*

[15] Bachl, Maximilian. *Machine learning methods for communication networks: characterization and analysis of selected use cases.* Diss. Wien, 2021.

[16] Hollot, Christopher V., et al. "Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Transactions on Automatic Control*, Vol.47, No.6, June 2002.

[17] Sreelakshmi Manjunath, Gaurav Raina, "Stability and performance of compound TCP with a proportional integral queue policy" *IEEE Transactions on Control Systems Technology*, 27(5), 2139-2155, 2018.

[18] Barczyk, Marek, and Andrzej Chydzinski. "AQM based on the queue length: A real-network study." Plos one 17.2 (2022): e0263407.

[19] El Fezazi, Nabil, et al. "AQM congestion controller for TCP/IP networks: Multiclass traffic." Journal of Control, Automation and Electrical Systems 31 (2020): 948-958.

[20] S. Patel, Y. Shukla, N. Kumar, T. Sharma and K. Singh, "A Comparative Performance Analysis of TCP Congestion Control Algorithms: Newreno, Westwood, Veno, BIC, and Cubic," 2020 6th International Conference on Signal Processing and Communication (ICSC), Noida, India, 2020, pp. 23-28, doi: 10.1109/ICSC48311.2020.9182733

[21] Shan Chen, Brahim Bensaou, Can high-speed networks survive with DropTail queues management, Computer Networks, Volume 51, Issue 7, 2007, Pages 1763-1776, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2006.11.004.

[22] P. Mrozowski, A. Chydzinski "On the deployment of aqm algorithms in the internet". In Proceedings of the 11th WSEAS international conference on Mathematical methods and computational techniques in electrical engineering 2009 Sep 28 (pp. 276-281)