

Bangladesh University of Engineering and Technology

Department of Computer Science and Engineering

CSE 322 - Computer Networks Sessional

Report Of NS2-Project

TCP-FIT: An improved TCP congestion control algorithm and its performance

Submitted by:

Abdur Rafi

1805008

Supervisor:

Dr. Md. Shohrab Hossain

Professor

Department of Computer Science and

Engineering, BUET.

Date of Submission

27 February, 2023

# Table of contents

[Chosen Paper](#_knhgmptc8c2h)

[Paper link](#_wdbsod9cwogv)

[Authors](#_yhckng10ipjb)

[Intuition behind the TCP-Fit algorithm](#_6nduywacz01q)

[Network Topologies Under Simulation](#_jvwjhnjhtls0)

[Topology used in paper](#_jsyzrba4ydxm)

[Topology simulated](#_wh1v20vduyjp)

[Wired:](#_fnhm6u7mi0sf)

[Wired and Wireless:](#_r83h1q9qptxp)

[Wireless:](#_jfetdbapj2yh)

[Parameters Under Variation](#_w37ykcu122tu)

[Modifications Made In The Simulator](#_s04dzs2zz6v8)

[Equations implemented](#_wciwrt4tsw07)

[Change in code](#_mjobhtdr5ieb)

[Default values of parameters](#_7bzdy5ss2jkf)

[Results with graph](#_bgf7xin44da3)

[Wired Network](#_kxa92lrnapy0)

[Varying flows](#_vui828yv2b3e)

[Observations](#_lb7t6jjm96h1)

[Varying nodes](#_wf36fyposomy)

[Observations](#_ux9xbyvu1fb1)

[Varying packet rate](#_qskr2y4ce9yt)

[Observations](#_jki02b47g4ob)

[Wired-Wireless](#_asabh5fza1l0)

[Varying flows](#_rs5d3hykm0h9)

[Observation](#_qgi8fshf0c2v)

[Varying nodes](#_4wmafci97mhd)

[Observations](#_tleboysi2trj)

[Varying packet rate](#_bkxtoeus13fg)

[Observations](#_9frosoxlao4w)

[Varying speed](#_3gon2kuay5qm)

[Observations](#_wq3phcu9fw0r)

[Wireless-Bottleneck](#_w8qxu1u6ubam)

[Varying flows](#_o0p6f4eoh9ug)

[Observations](#_skvltefdqezi)

[Varying nodes](#_a4ixqzao445t)

[Observations](#_y8f2qsi4lvhq)

[Varying packet rate](#_rilx9s6pt7ao)

[Observations](#_q35vi61ektks)

[Varying speed](#_usv1nmaqevcr)

[Observations](#_n1h8f4m3l01d)

[Wireless Random](#_muodw8tajo16)

[Varying flows](#_fodbk4cknkx6)

[Observations](#_ti9sd1h6c96y)

[Varying nodes](#_4xp5xuusx1xx)

[Observations](#_xgrj9fuwsy6n)

[Varying packet rate](#_9svgux29sy62)

[Observations](#_6y8se8s5fn9k)

[Varying speed](#_bzwpb4lyw7im)

[Observations](#_4fg6ygpyeb61)

[Wireless-Grid](#_2f78udx0y85r)

[Varying flows](#_nvz8aef3xdhy)

[Observations](#_7148wp4mbyxk)

[Varying nodes](#_qaaop74jxg35)

[Observations](#_rconjueumdri)

[Varying packet rate](#_408uhw5c6ny0)

[Observations](#_guaf346h3mu5)

[Varying speed](#_uob8zckgsu1)

[Observations](#_2m90uz83o2hr)

[Summary Findings](#_kdvmgfiotqmi)

# 

# 

# 

# Chosen Paper

### Paper link

[TCP-FIT: An improved TCP congestion control algorithm and its performance](https://ieeexplore.ieee.org/document/5935128)

### Authors

[Jingyuan Wang](https://ieeexplore.ieee.org/author/37716384100)

[Jiangtao Wen](https://ieeexplore.ieee.org/author/37291696100)

[Jun Zhang](https://ieeexplore.ieee.org/author/37717249400)

[Yuxing Han](https://ieeexplore.ieee.org/author/37406874100)

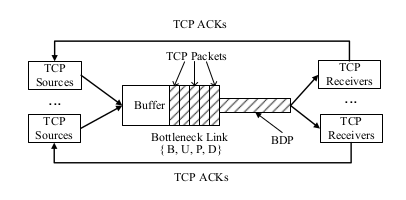
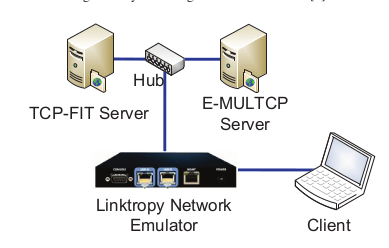
### Intuition behind the TCP-Fit algorithm

Tcp algorithms in general fail to maintain performance in wireless high BDP networks. In such networks there are packet drops and high delays. TCP algorithms may think that the network is congested and send less packets. As a result, bandwidth utilization decreases.

The TCP-Fit algorithm takes a different approach to solving this issue. It simulates N parallel connections through the congestion window. N is dynamically changed based on network condition. The idea is similar to parallelism in computer hardware. When we failed to increase the performance of a single cpu by a good margin, we employed parallel cores to increase performance. I understood similar kind of intuition behind the TCP-Fit algorithm

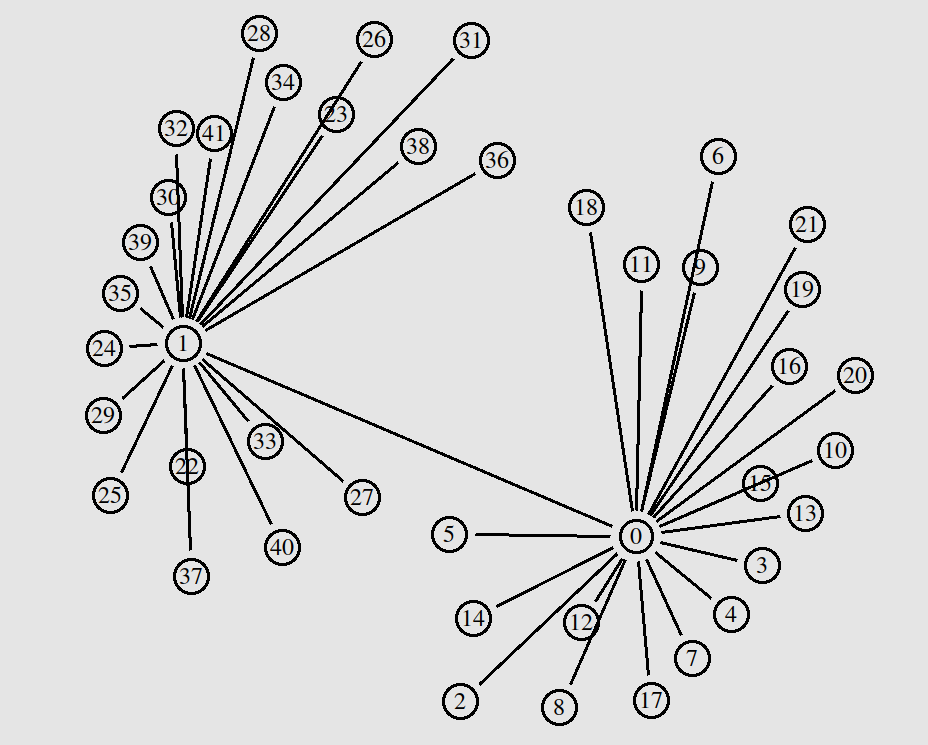
# Network Topologies Under Simulation

## Topology used in paper



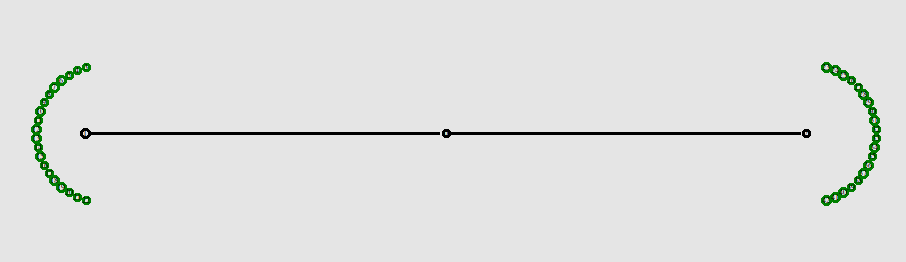
## Topology simulated

#### Wired:



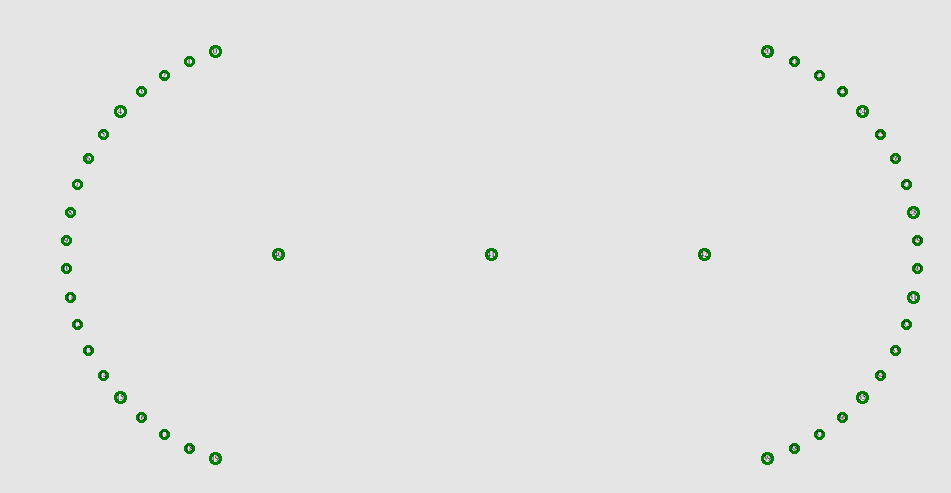
Single High BDP Bottleneck Link

### Wired and Wireless:

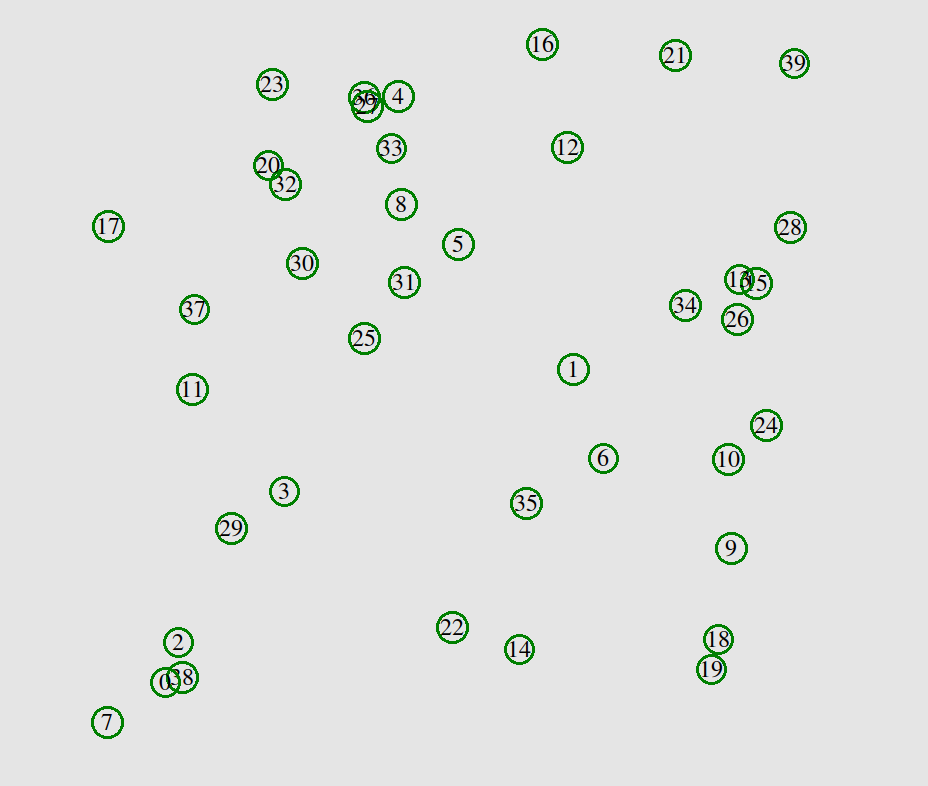


Single High BDP Bottleneck Link

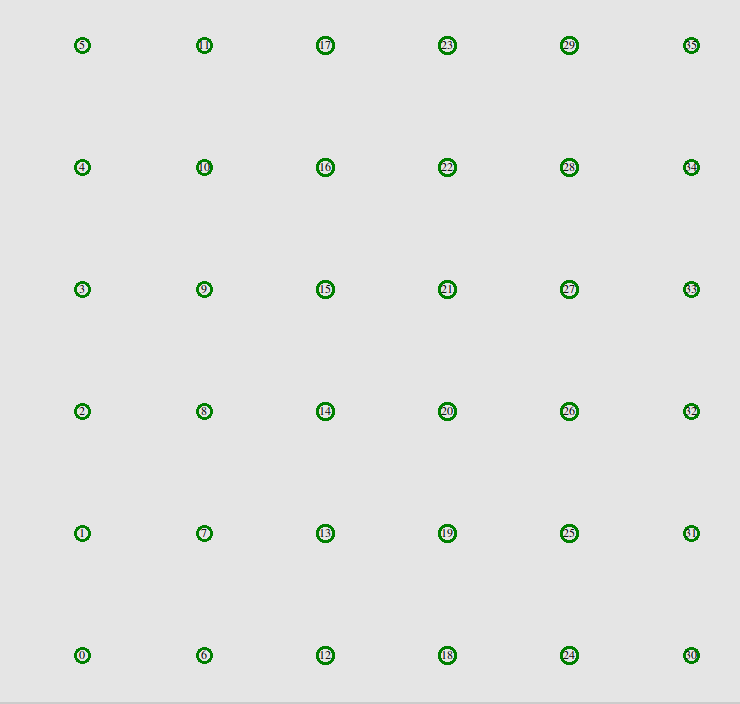
### Wireless:



High BDP Link Simulated in Wireless Network



Wireless Random Nodes



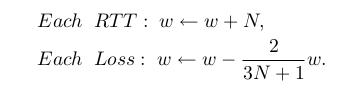
Wireless Grid Network

# Parameters Under Variation

1. Number of Nodes (20, 40, 60, 80, 100)
2. Number of flows (10, 20, 30, 40, 50)
3. Number of packets sent ( 100, 200, 300, 400, 500)
4. Speed (for wireless nodes) ( 5 m/s, 10 m/s, 15 m/s, 20 m/s, 25 m/s)

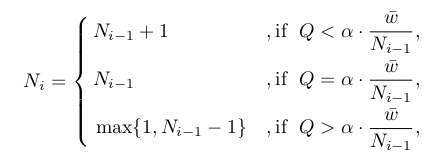
# Modifications Made In The Simulator

### Equations implemented



Where w is the congestion window

N is a dynamic parameter which is updated after each packet loss



Here Q is an estimate of the number of in-flight session that are queued in the network buffer



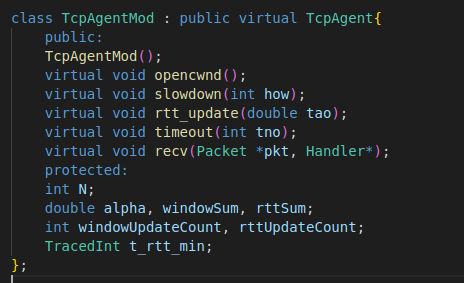
### 

### 

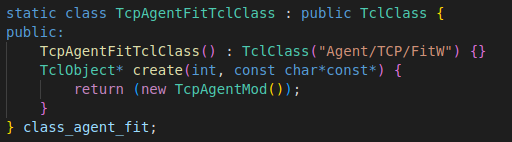
### Change in code

A new agent for tcp fit added in tcp.h

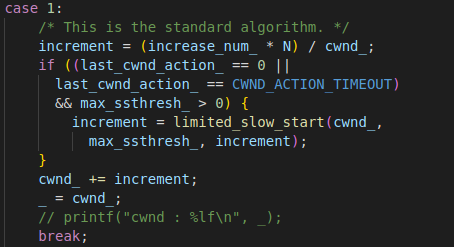
A new file tcpMod.cc added



Corresponding TCL class

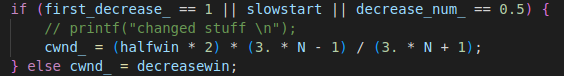


In **opencwnd** method



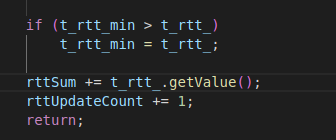
Updating congestion window at each RTT

In **slowdown** method



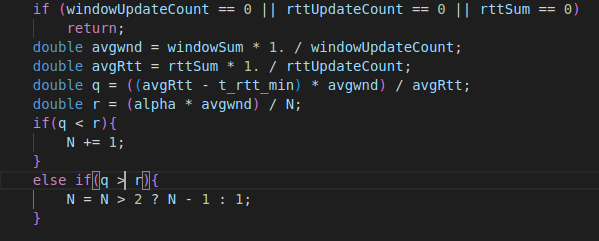
Decreasing congestion window at packet loss

In **rtt\_update** Method



Keeping track of minimum rtt value and average rtt

In **timeout** method



Updating N

In recv method



Keeping track of average window size

# Default values of parameters

Number of nodes = 40

Number of flows = 40

Area = 500

Speed = 15 m/s

Routing protocol for wireless networks = DSR

Queue size = 50

Queue Type = CMUPriQueue

Traffic = Exponential

Packet Rate of traffic source = 300

Packet size = 200 bytes

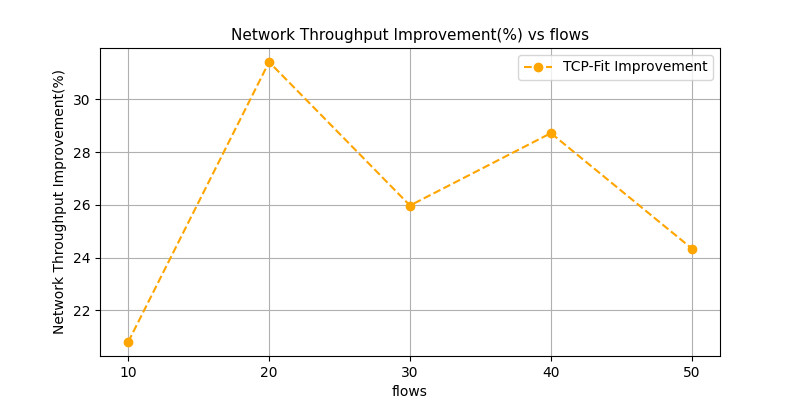
# Results with graph

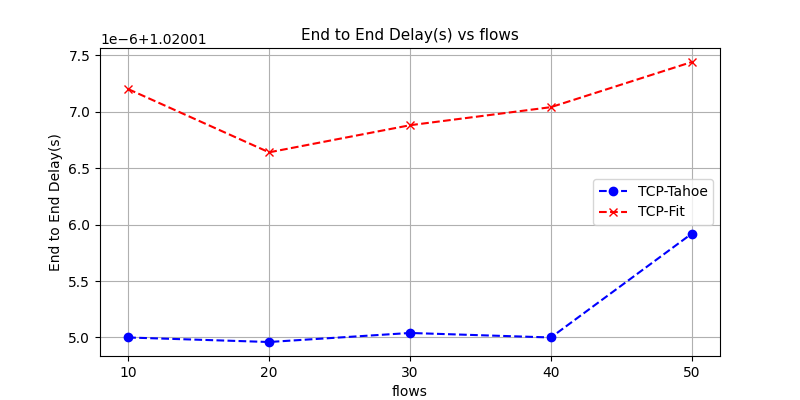
## Wired Network

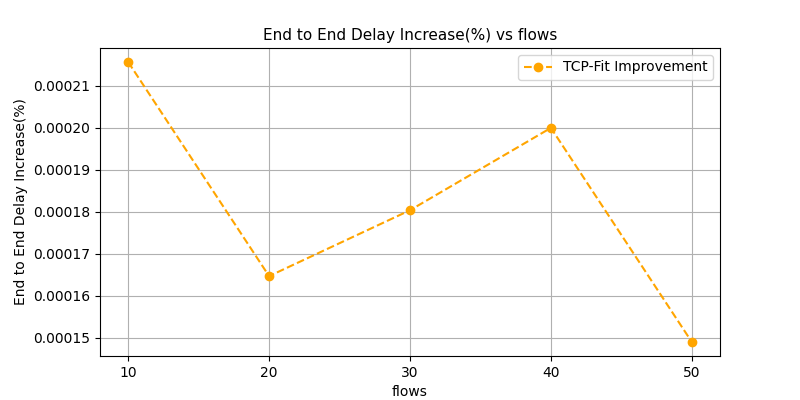
Wired network with a single high BDP link. The link introduces 5% random uniform packet drops.

### Varying flows

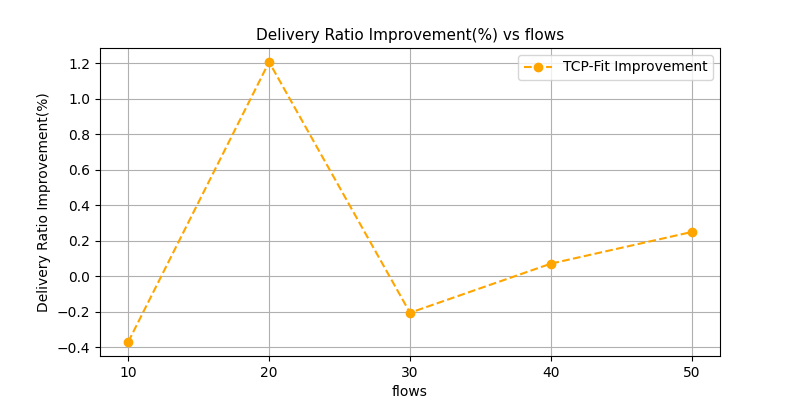


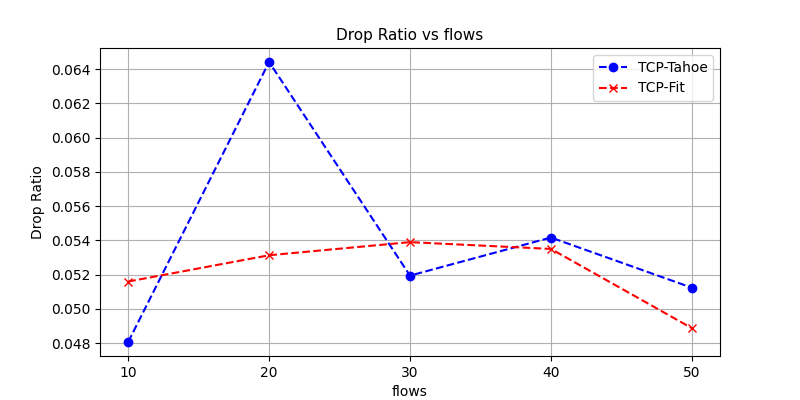


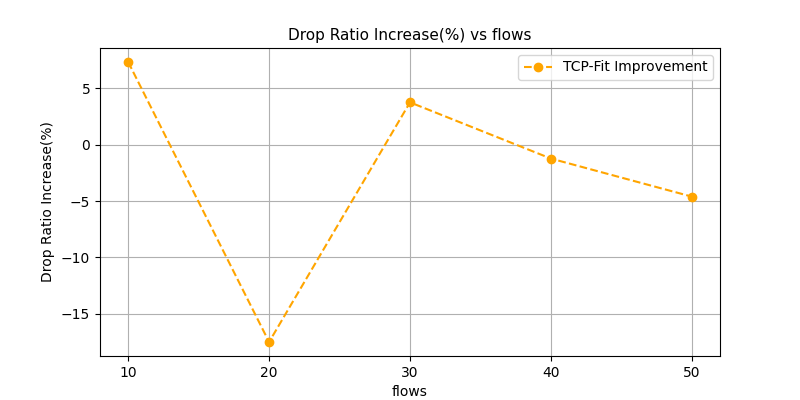








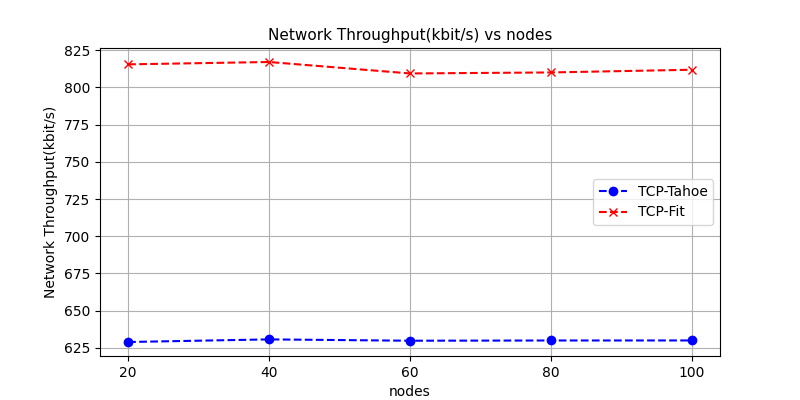


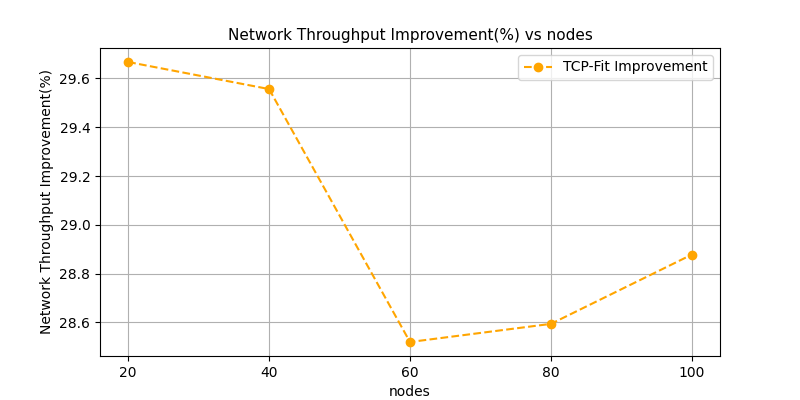


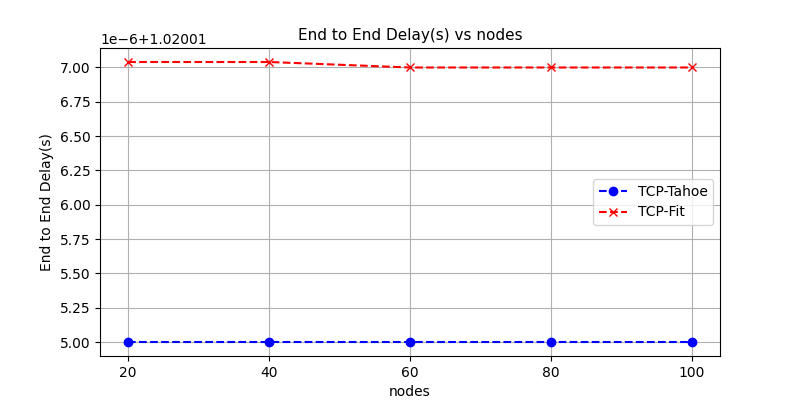
#### Observations

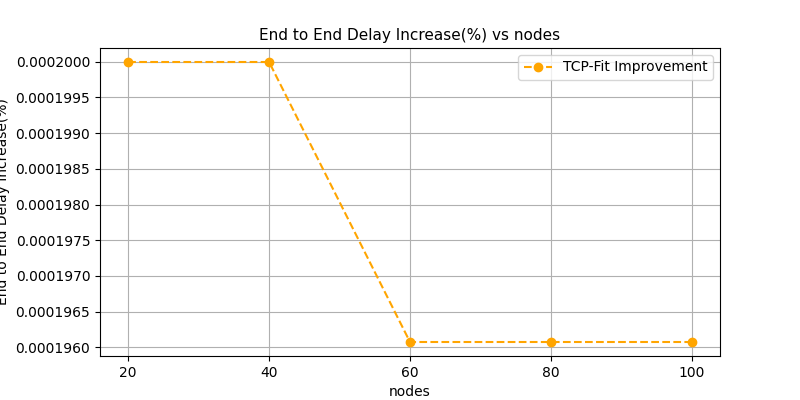
1. Throughput is seen to increase with increasing number of flows. This aligns with intuition. More flows means more packets sent at a unit time and so more throughput. Tcp fit shows significant improvement in network throughput compared to TCP Tahoe
2. End to end delay pretty much stays the same for varying flows, which is somewhat expected for wired networks. End to end delay is very slightly higher for TCP Fit.
3. 5% random packet drop was added to the bottleneck link to simulate random packet drops in a real life network. The delivery ratio and drop ratio are in accordance with that random packet drop.

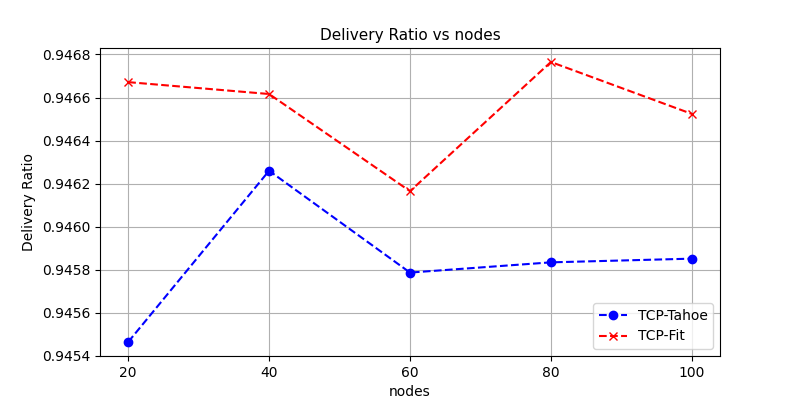
### Varying nodes

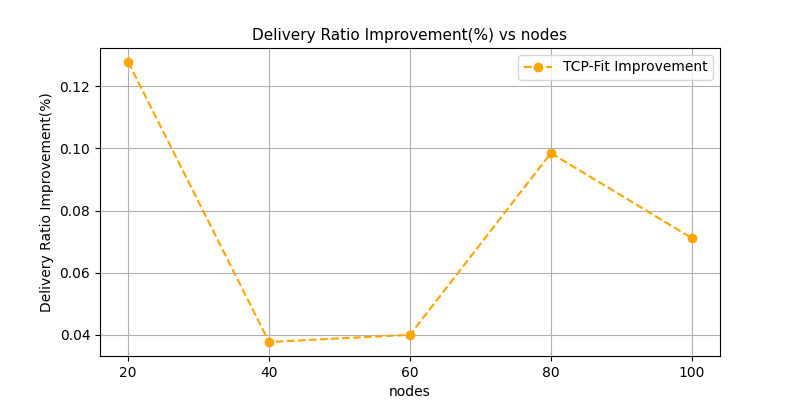


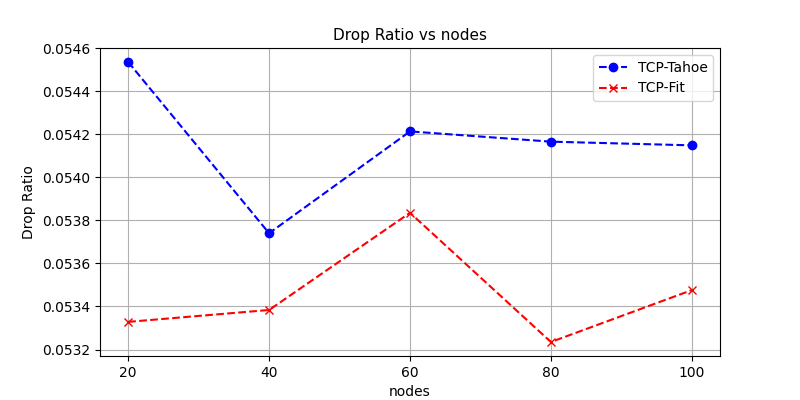


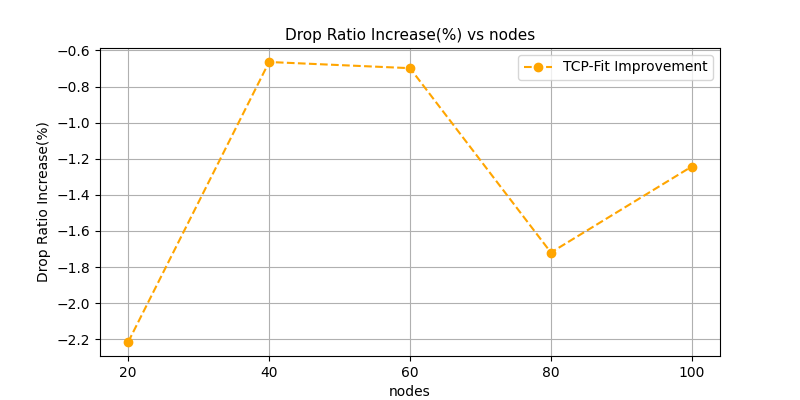








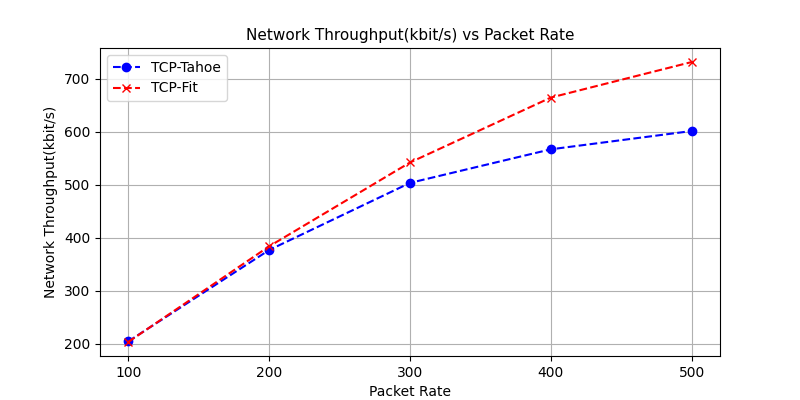


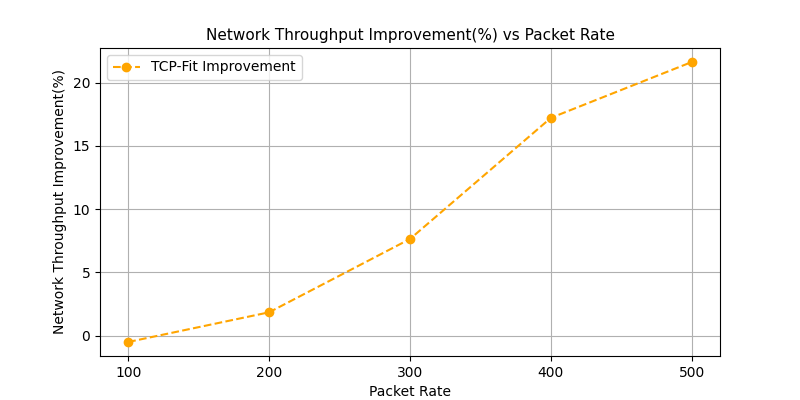


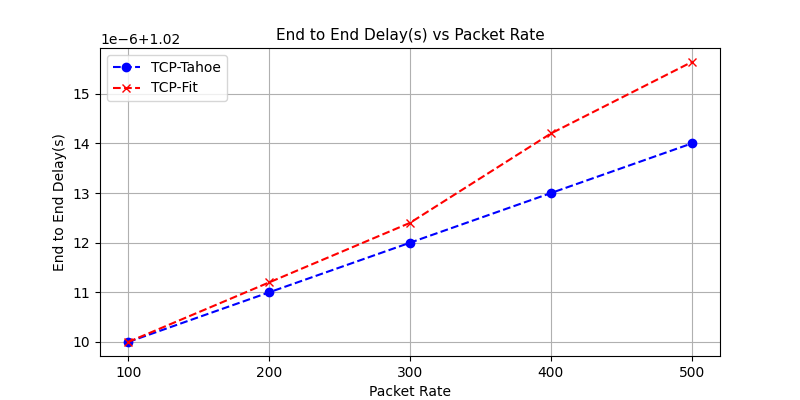
#### Observations

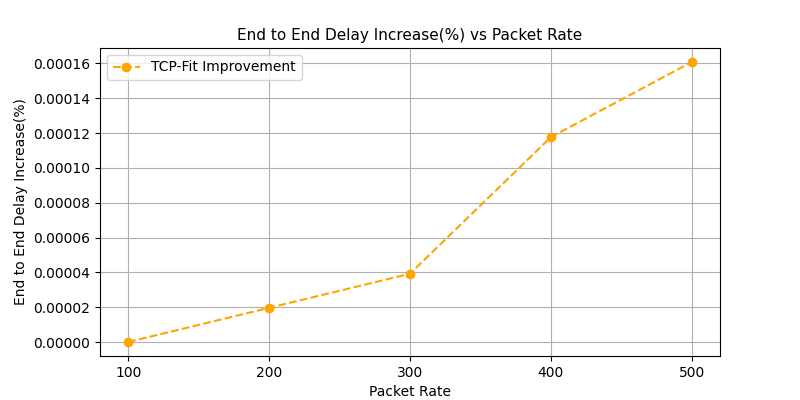
Varying number of nodes in wired networks with the considered topology does not really have any considerable effect on any of the metrics as they do not affect the path of the packets.

### Varying packet rate



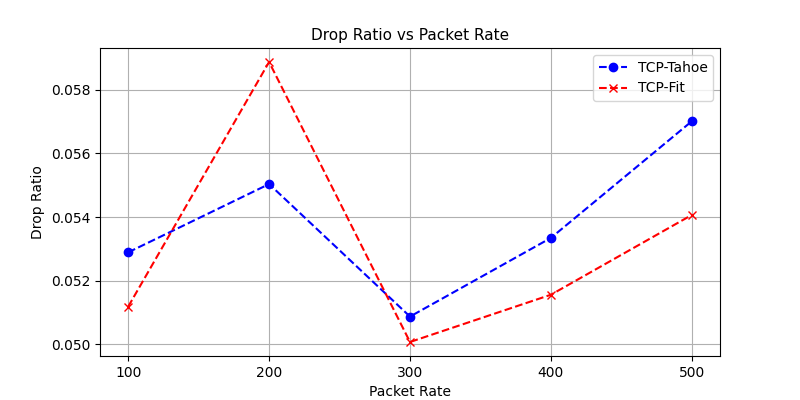


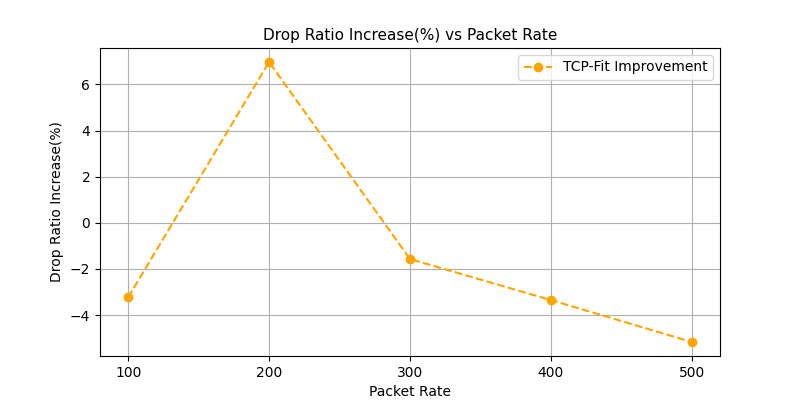












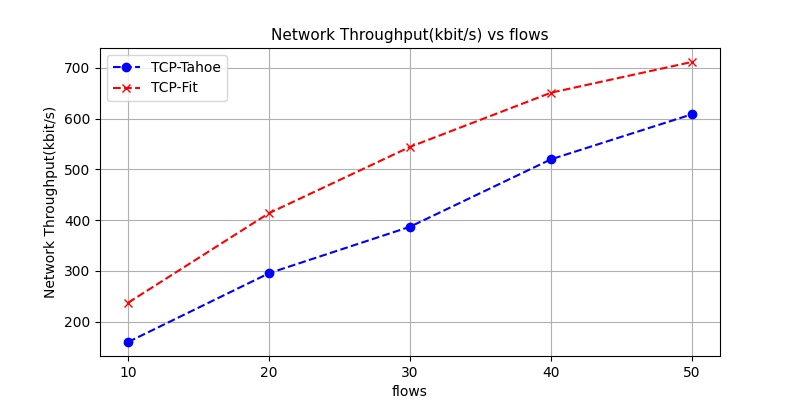
#### Observations

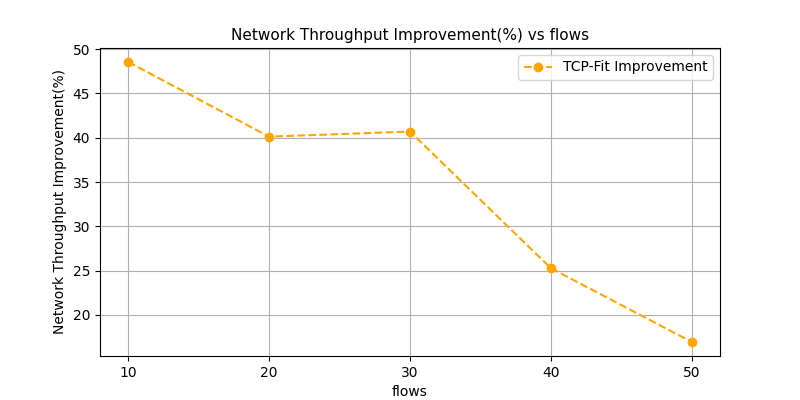
1. Throughput increases with increasing packet rate. The reasoning is similar to increasing the number of flows. As packet rate increases, TCP Fit seems to perform increasingly better compared to TCP Tahoe.
2. End to end delay is seen to increase very slightly with increasing packet rate. This is intuitive as higher packet rate should mean higher queuing delay. However, this delay is evidently very small as seen from the step size of the y axis. Again, end to end delay is very slightly higher for TCP-Fit

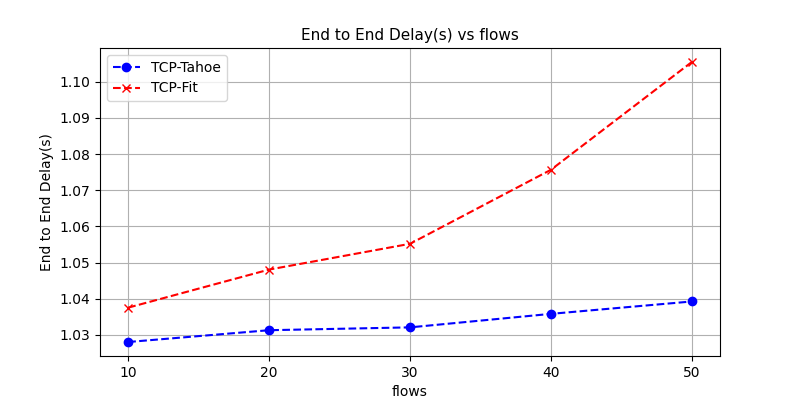
## Wired-Wireless

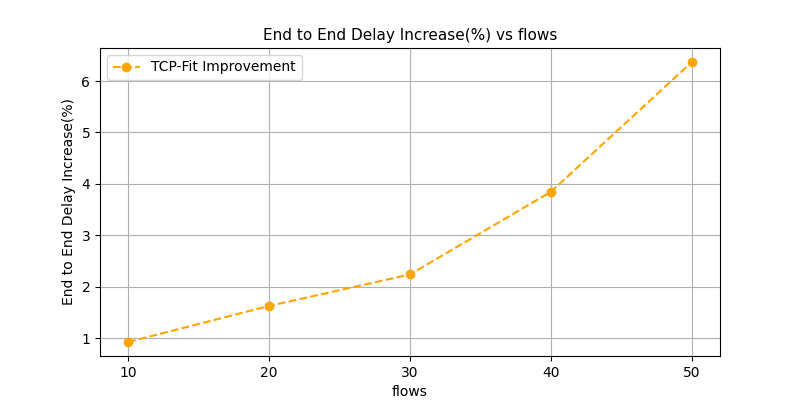
The bottleneck link was given 7% random packet drop and 1s delay

### Varying flows

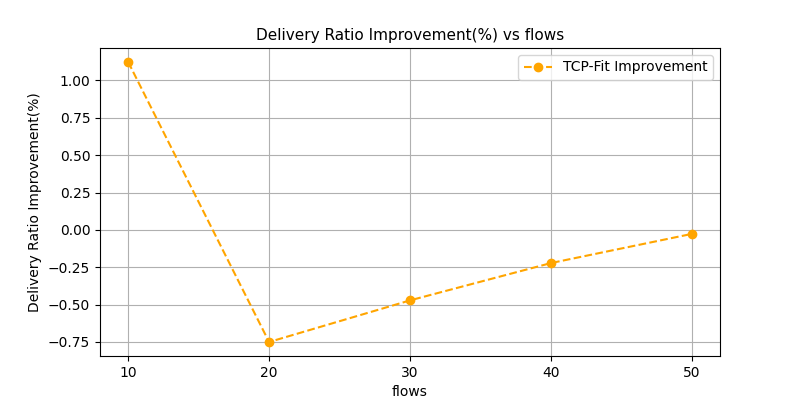


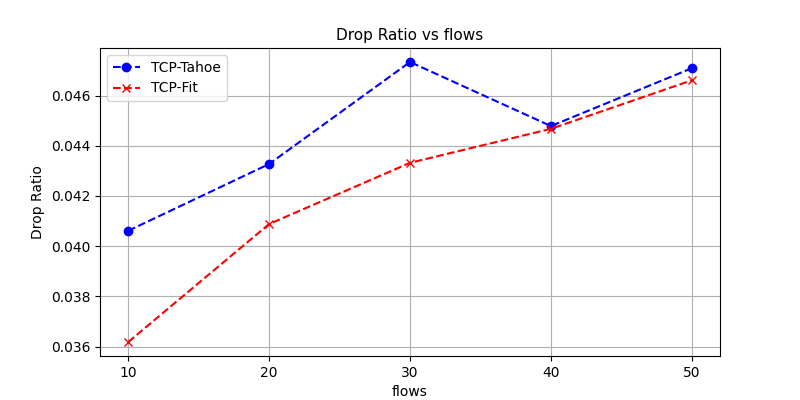


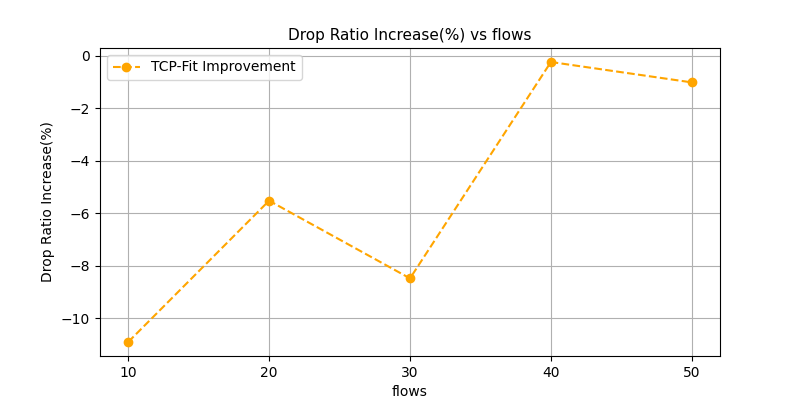








****

****

#### 

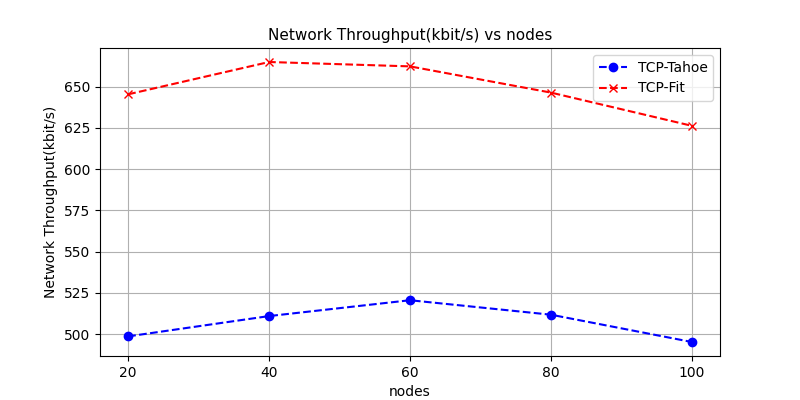
#### 

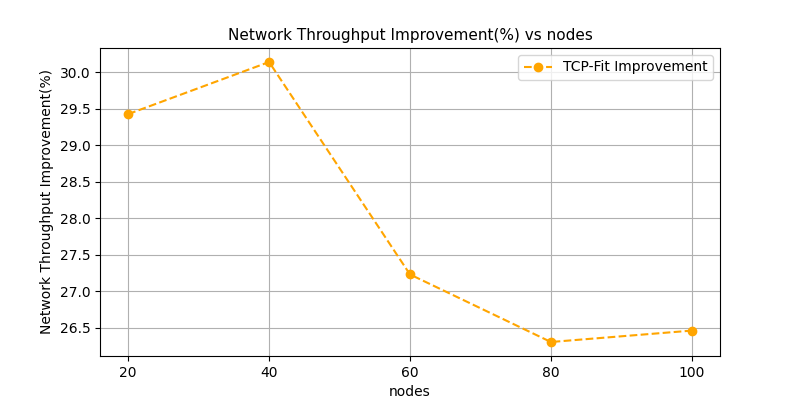
#### 

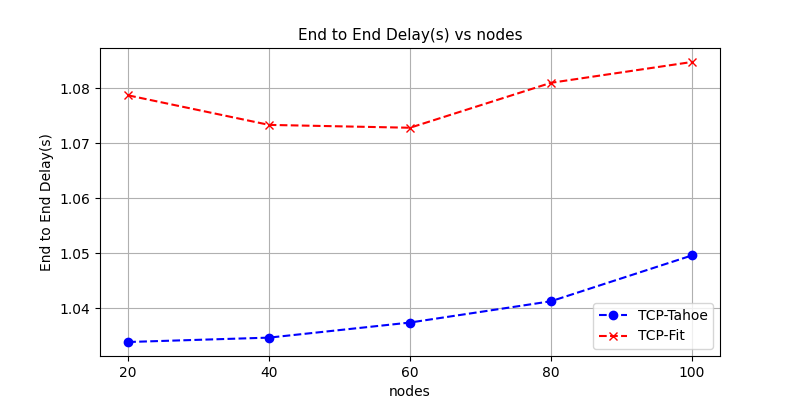
#### Observation

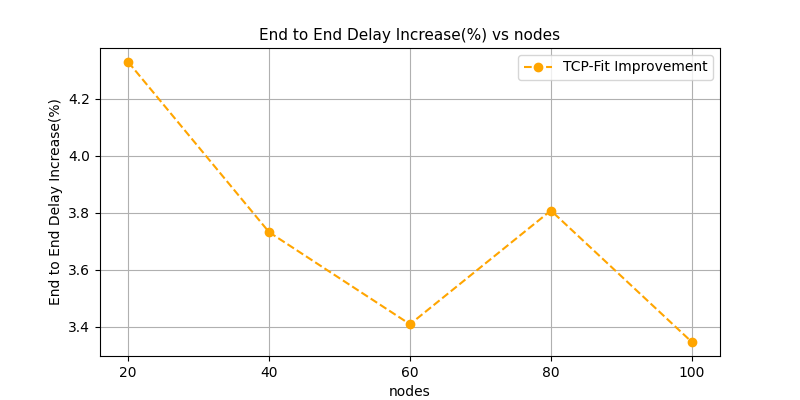
1. Throughput increases with the number of flows due to similar reasoning as wired networks. TCP-Fit gives significantly higher throughput than TCP-Tahoe
2. End to end delay is seen to increase with increasing flows. Notably end to end delay in TCP-Fit is higher and the difference seem to increase with more flows
3. It is seen that the delivery ratio drops and so the drop ratio increases slightly with increasing number of flows. This is most likely because of the packet drops due to limited buffer size. TCP-Fit does not seem to have any advantage or disadvantage against TCP-Tahoe in this regard.

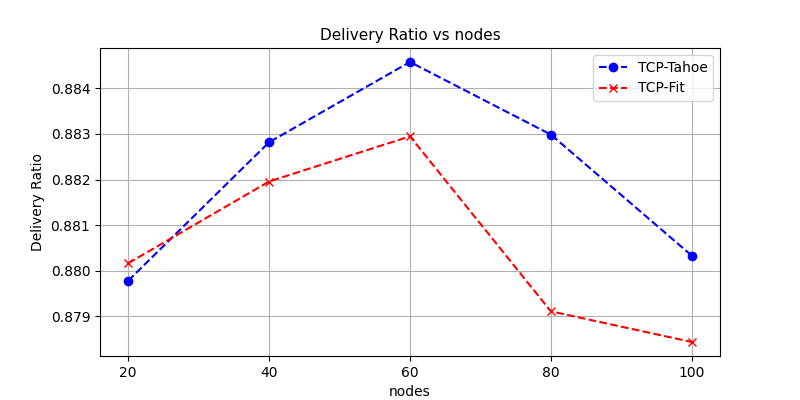
### Varying nodes

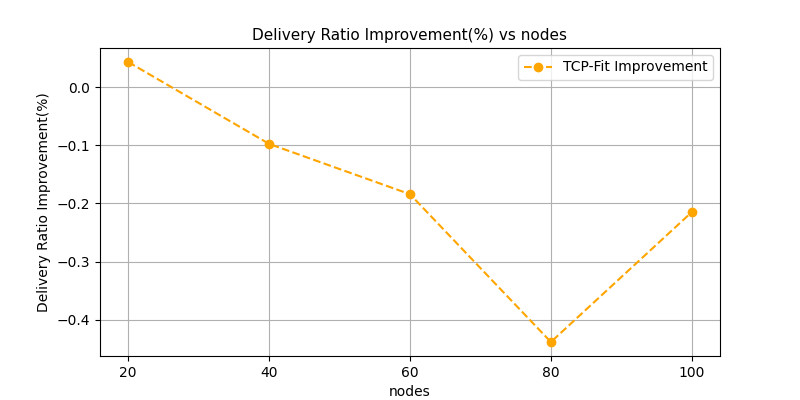
****

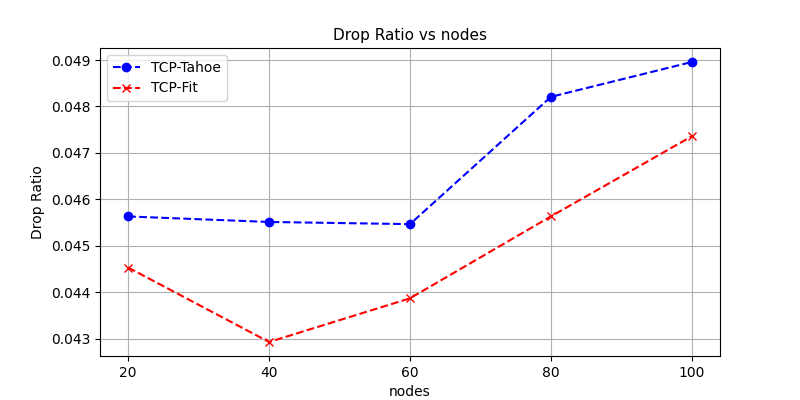
****

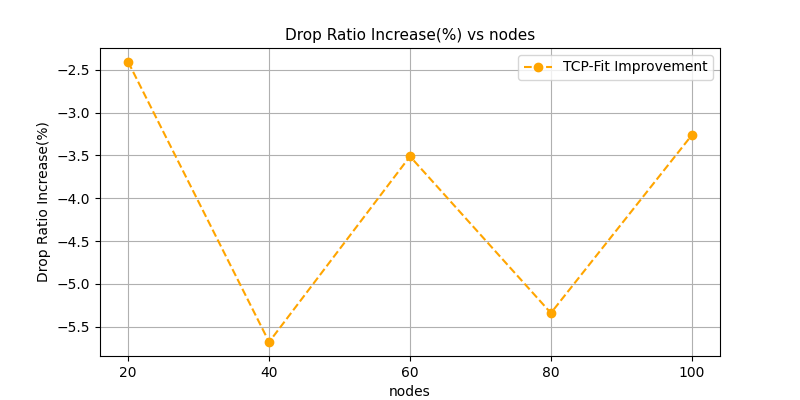
****

****

****

****

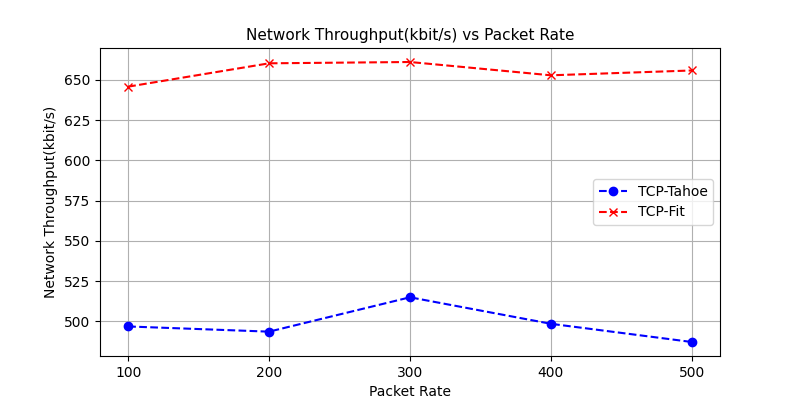
****

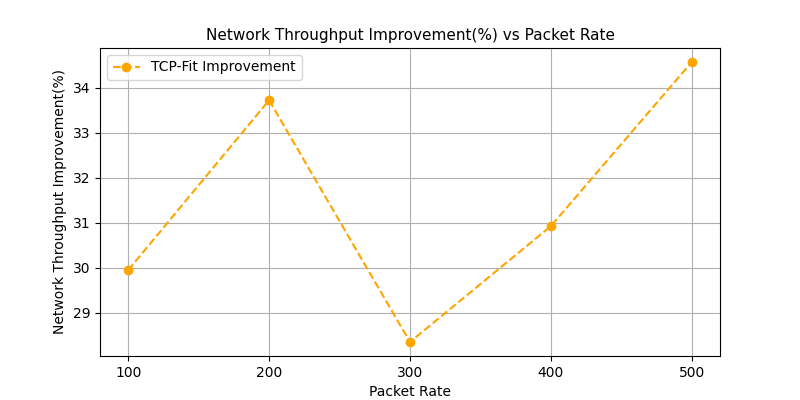
****

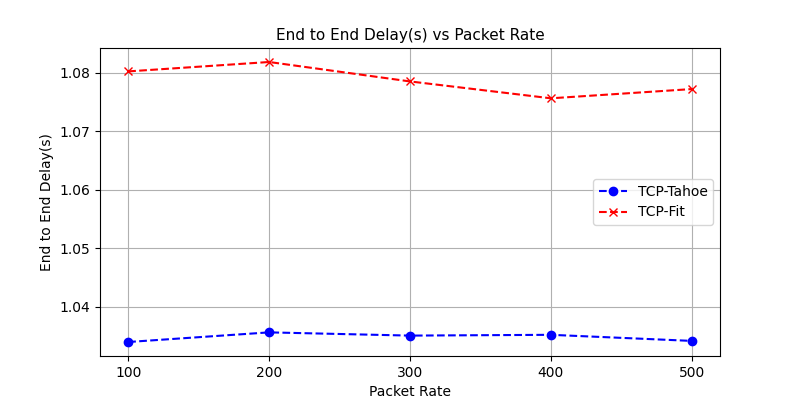
#### Observations

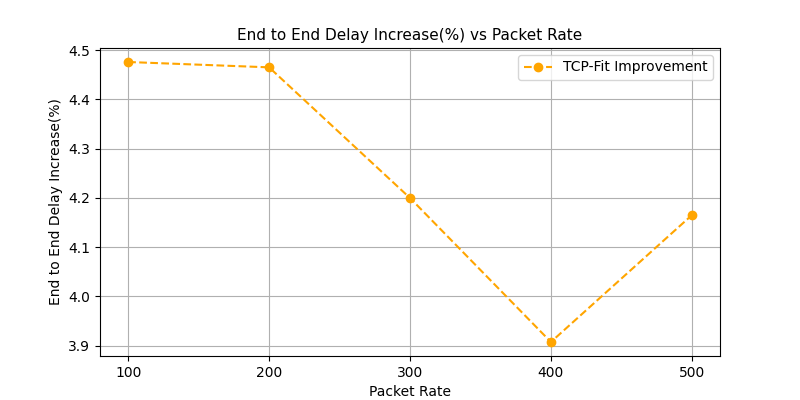
1. Throughput seems to increase with increasing number of nodes up to some point, then decrease. As the number of flows remains the same, increasing node count at first most likely distributes the flows. As a result throughput increases. Then at some point ,increase of interference takes over, and causes the throughput to decrease.
2. End to end delay is slightly affected. With the current topology, end to end delay is not much affected since most packets directly goes to the base station, only some packets take multiple hops before reaching base station or before reaching destination
3. No notable trend in drop ratio or delivery ratio is noticed
4. TCP-Fit continues to give large improvement in network throughput with small changes in other metrics

### Varying packet rate

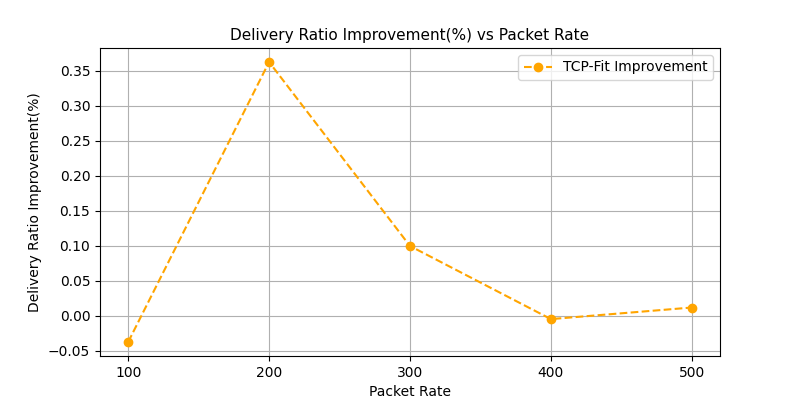
****

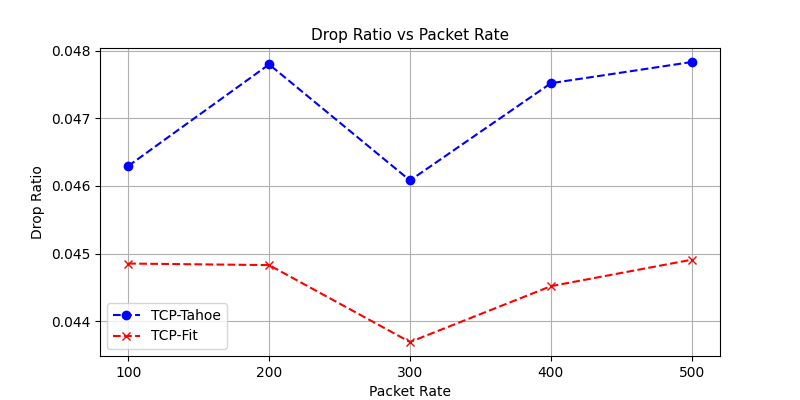
****

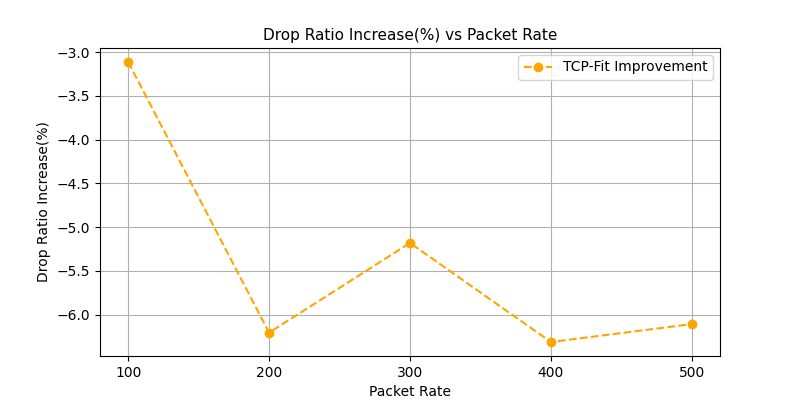
****

****

****

****

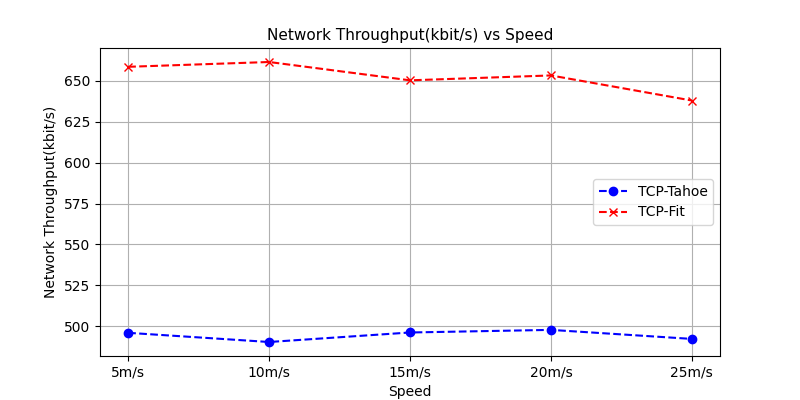
****

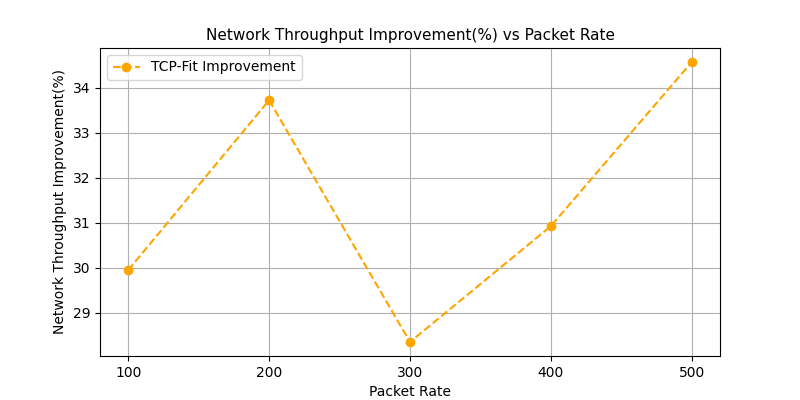
****

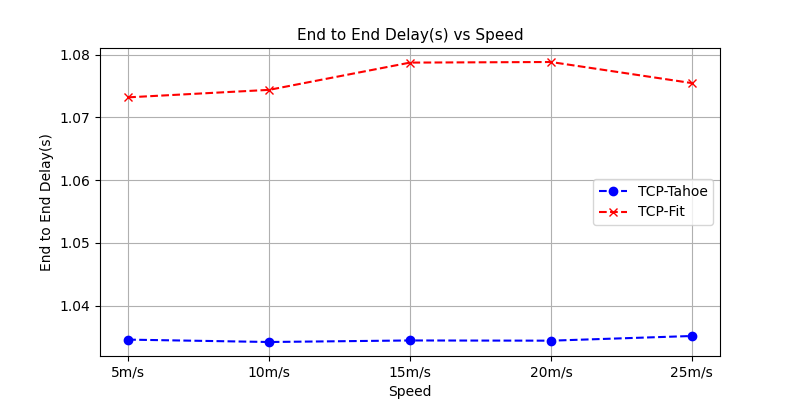
#### Observations

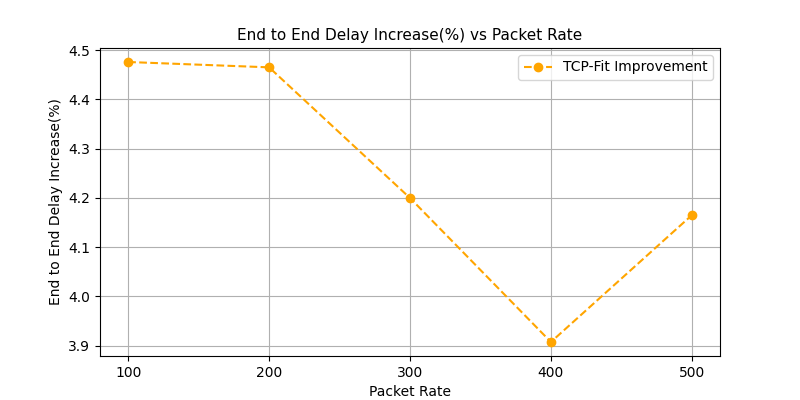
1. The metrics does not seem much affected by the packet rate indicating that bandwidth utilization for both algorithms were were at their maximum capacity for all the packet rate
2. TCP Fit continues to give large improvement in network throughput
3. TCP Fit seems to have slightly higher end to end delay and slightly better delivery ratio and so slightly smaller drop ratio

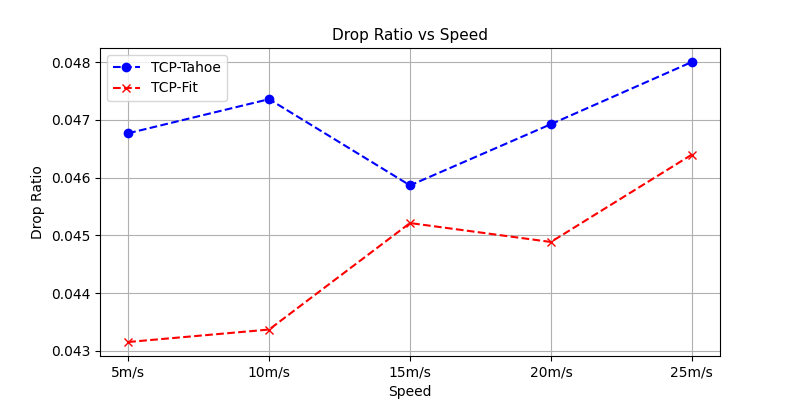
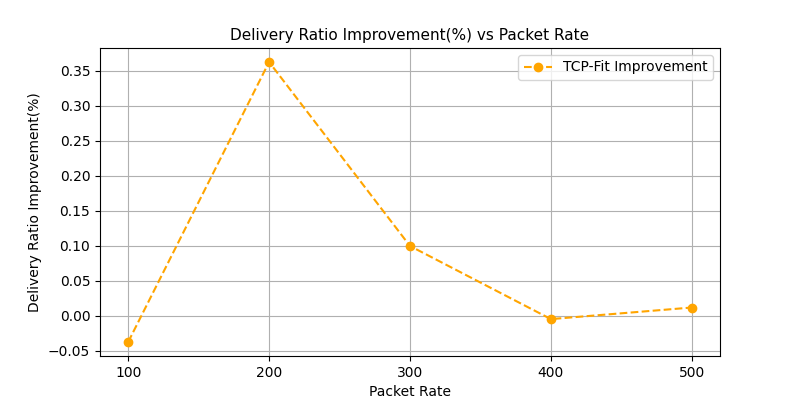
### Varying speed

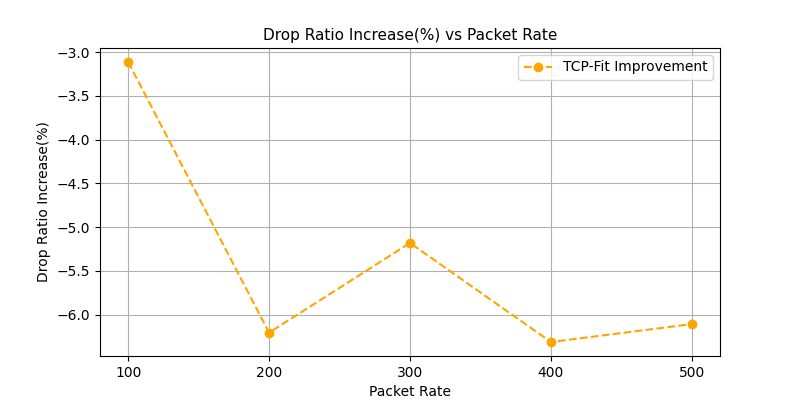
****

****

****

****

****

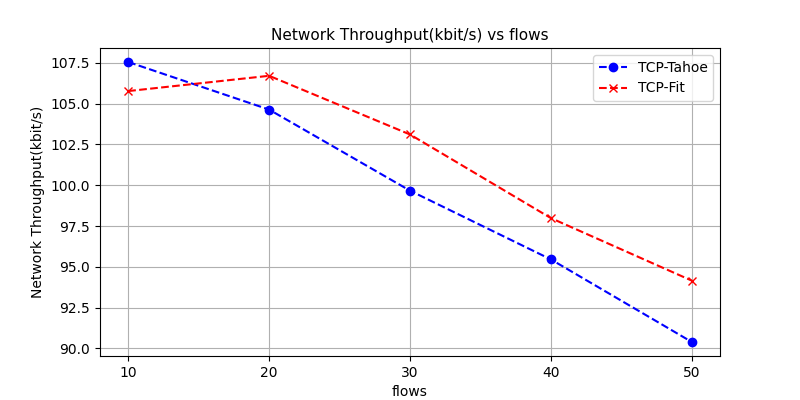
****

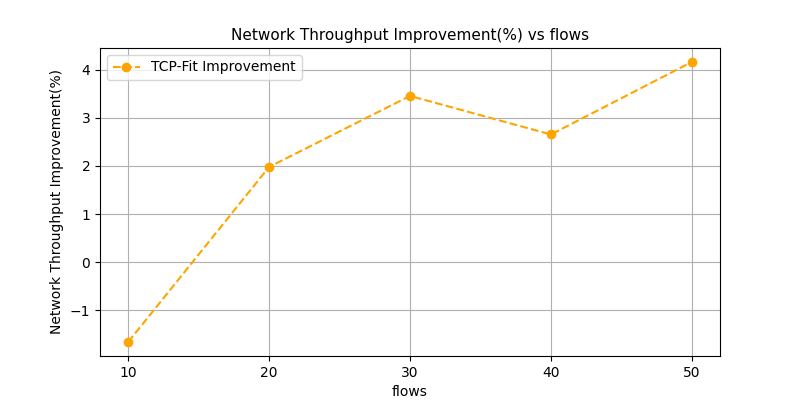
#### Observations

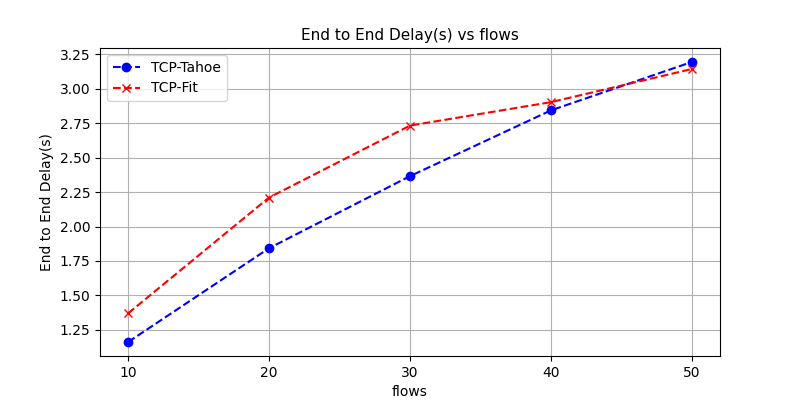
1. Varying speed does not seem to have a notable effect. This could be due to the nature of the motion. The motion was random. The nodes mostly moved about their initial position and so, there probably was not much change in routing paths throughout the simulation due to node movement.
2. TCP Fit continues to give large improvement in network bandwidth with marginal difference in other metrics

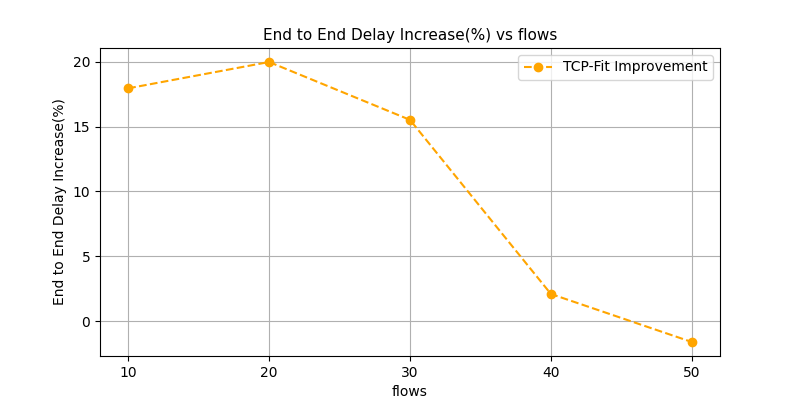
## Wireless-Bottleneck

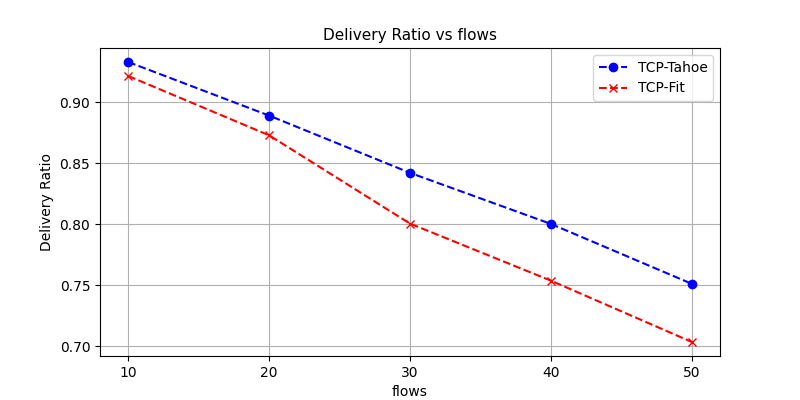
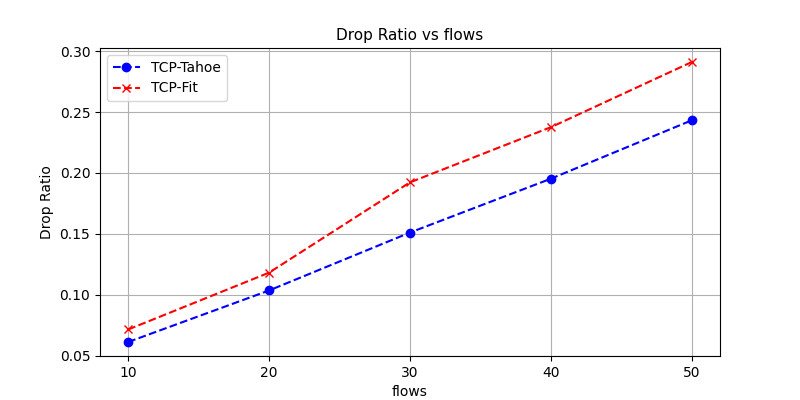
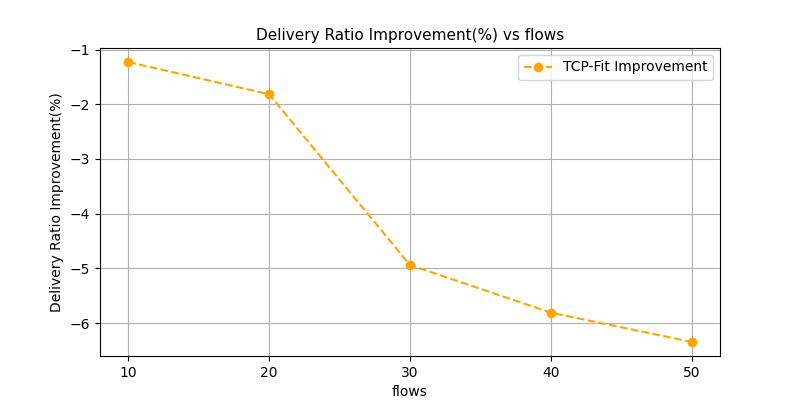
### Varying flows

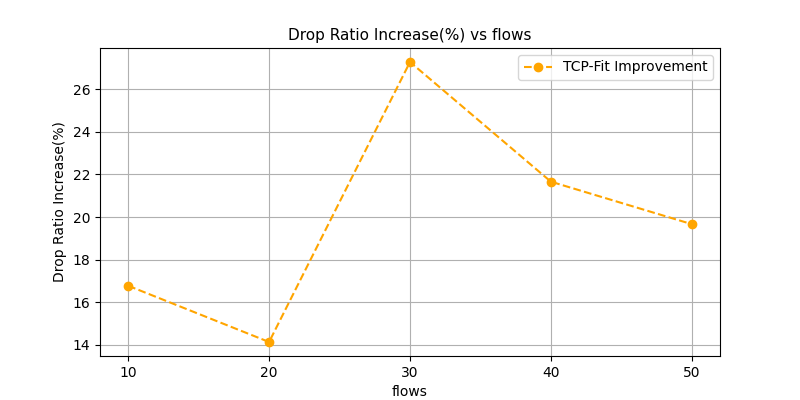


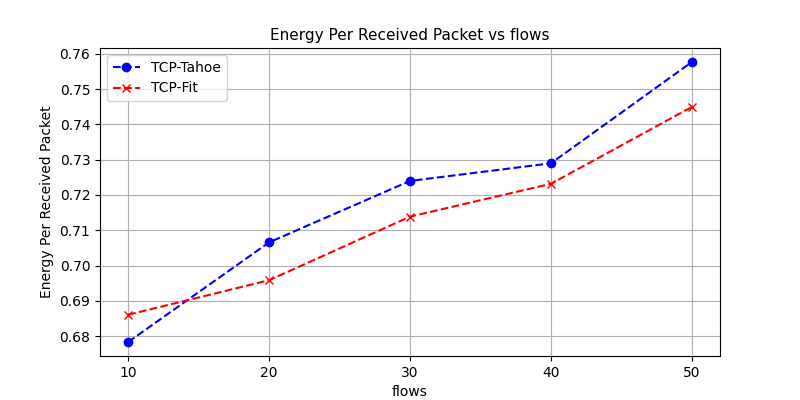












****

#### Observations

1. Throughput is seen to decrease with the number of flows, indicating that the network is congested. TCP-Fit only performs slightly better than TCP-Tahoe
2. End to end delay increases with more congestion. TCP Fit shows higher end to end delay
3. Delivery ratio drops and drop ratio increases with increasing flow, as expected from the throughput graph.
4. Higher amount of energy required for each received packet due to the higher packet loss. TCP-Fit shows lower energy requirements. This is due to the increased throughput.
5. TCP-Fit shows quite poor performance on the congested network. The throughput improvement was lower than 5%, however end to end delay increased as much as 20%, delivery ratio decreased by as much as 6% and drop ratio increased by as much as 26%

### 

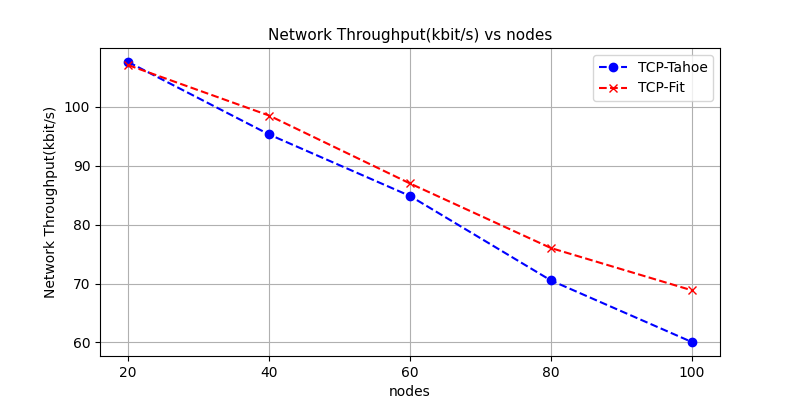
### 

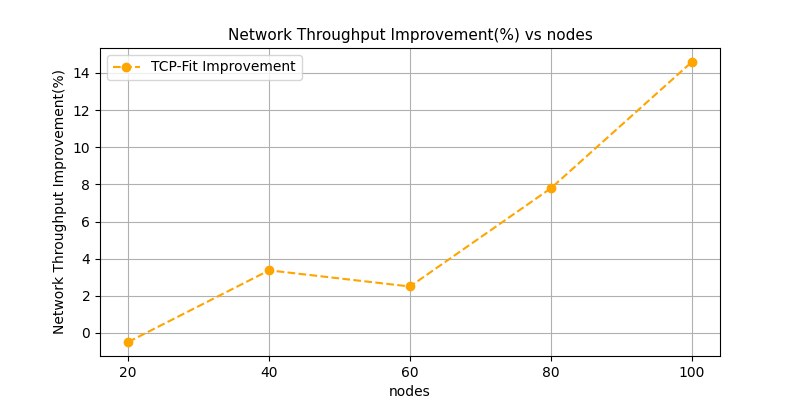
### 

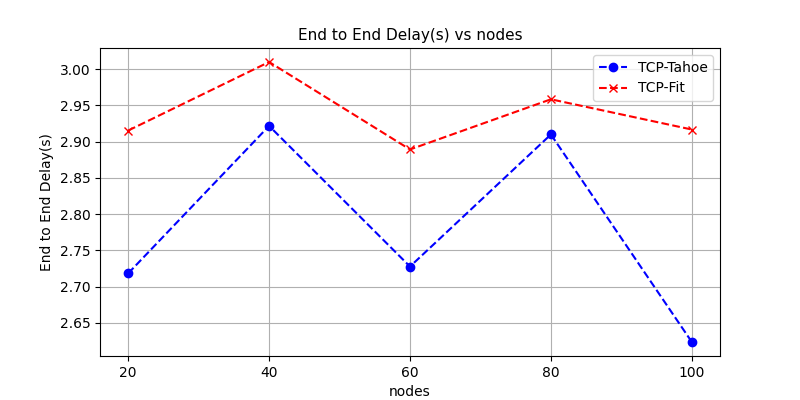
### 

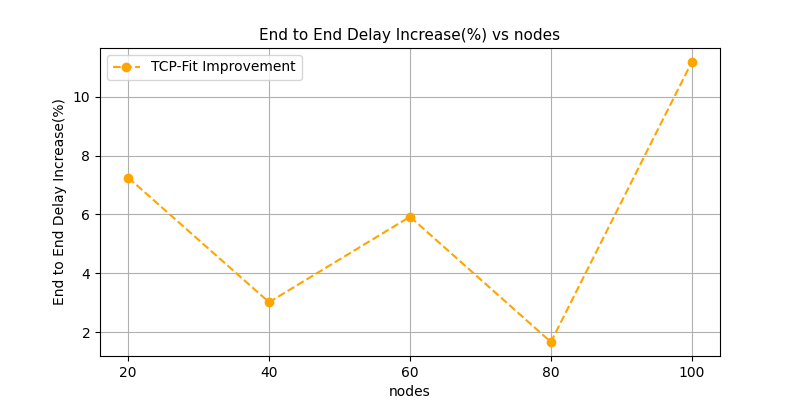
### 

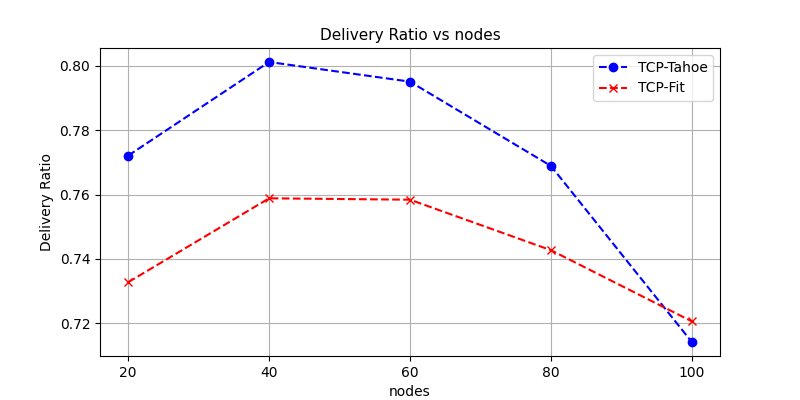
### Varying nodes

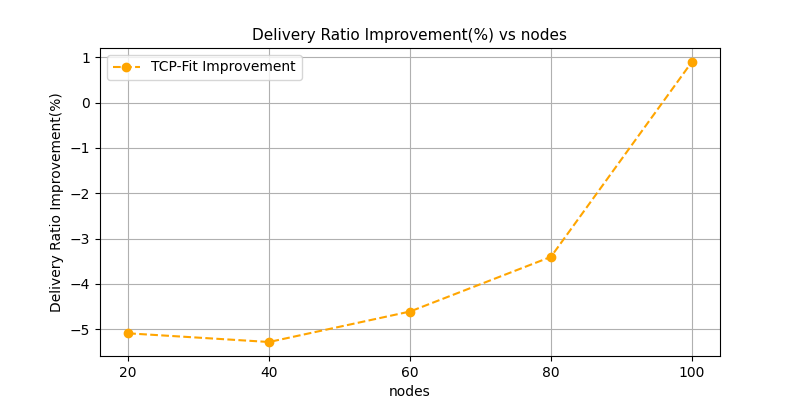


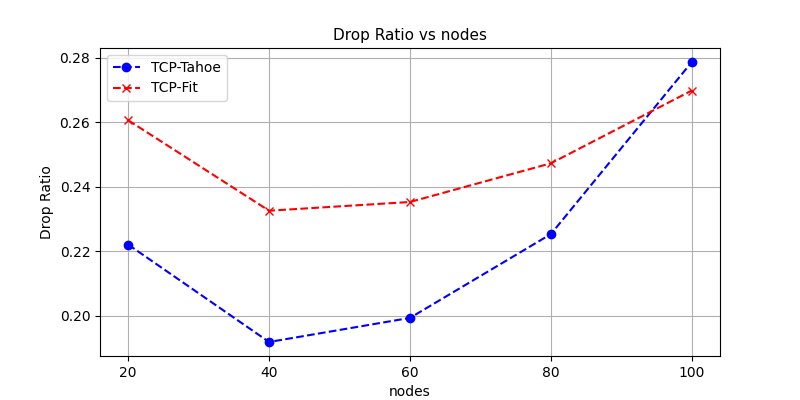


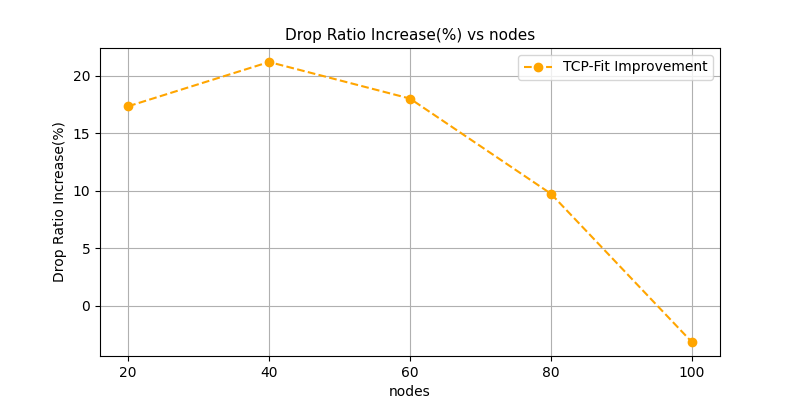


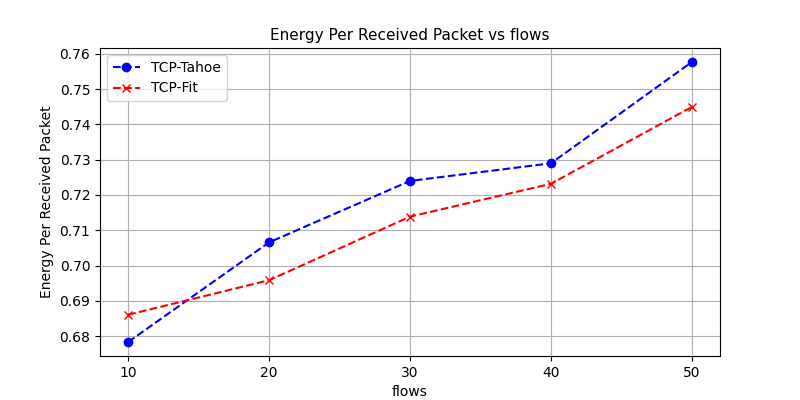


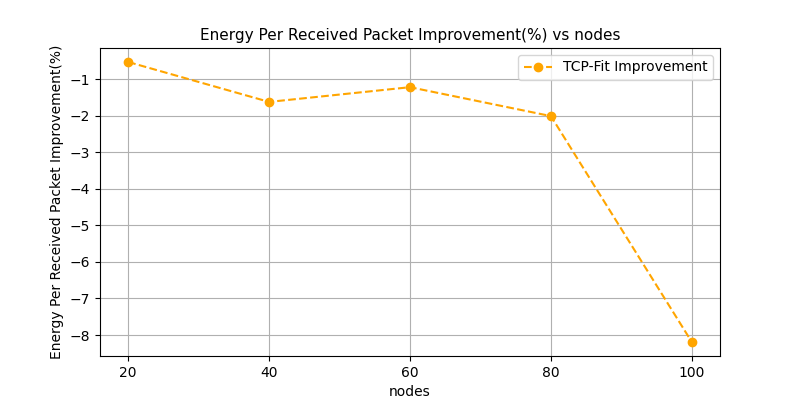








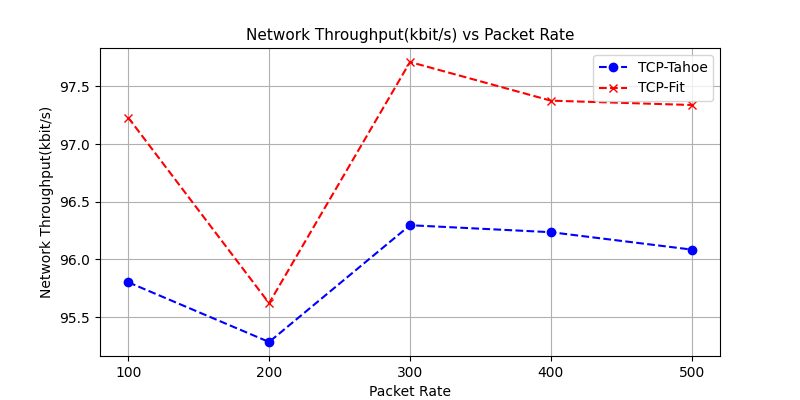


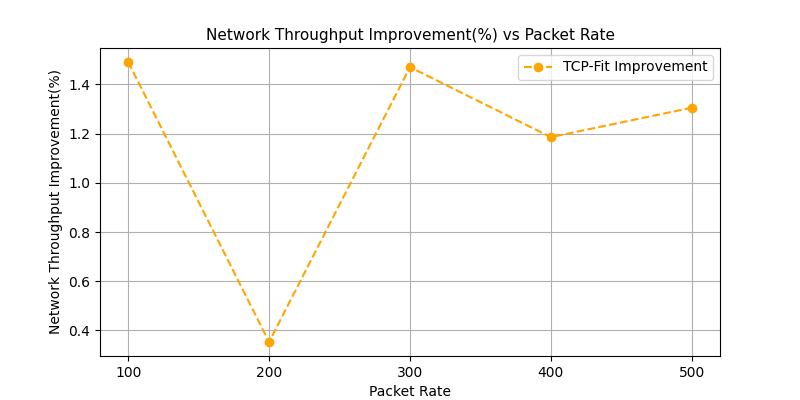


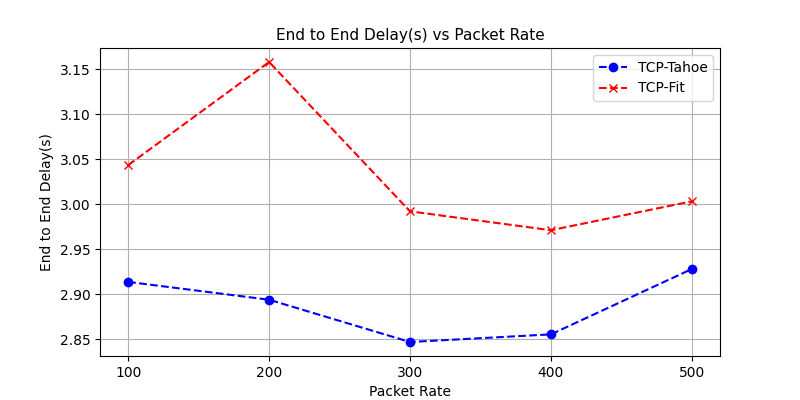
#### Observations

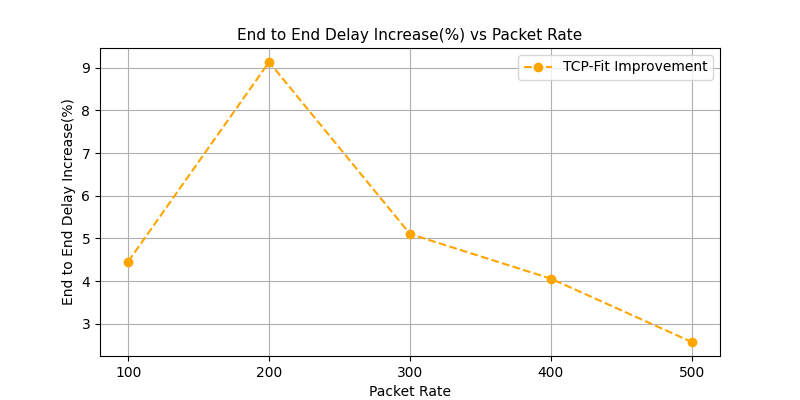
1. Throughput decreases rapidly with increasing node count. However the end to end delay does not show such a trend. This indicates that the increased interference due to higher node count is most likely the cause of decreased throughput
2. End to end delay is increased with TCP-Fit algorithm
3. Energy per received packet also increased
4. Throughput improvements are not as much as previous topologies

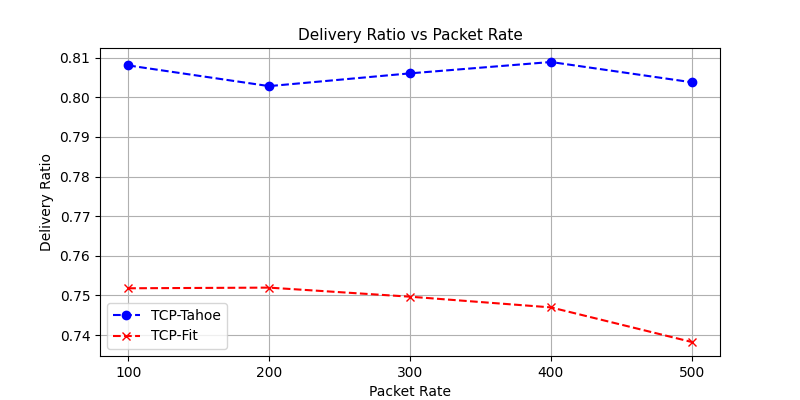
### Varying packet rate

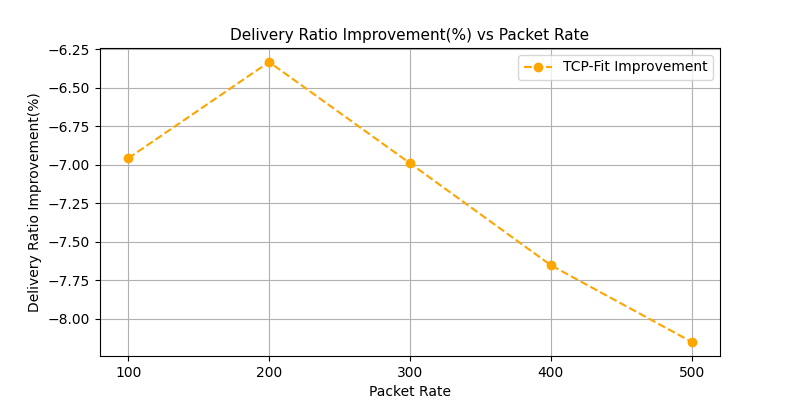


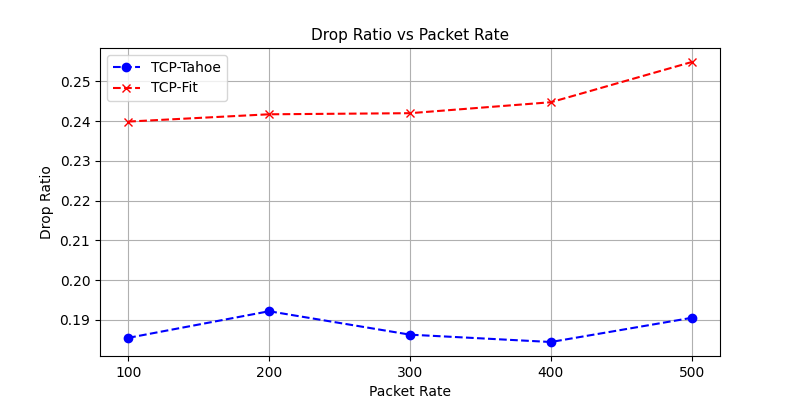


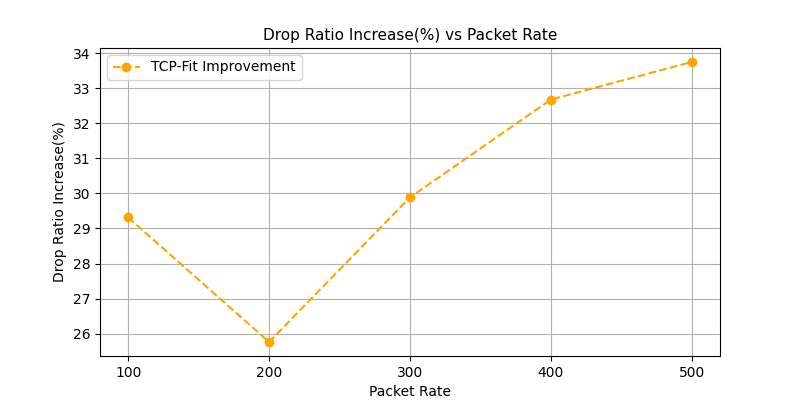


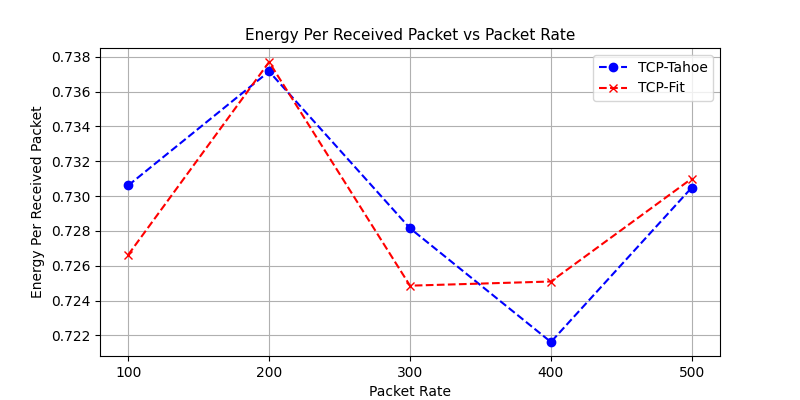


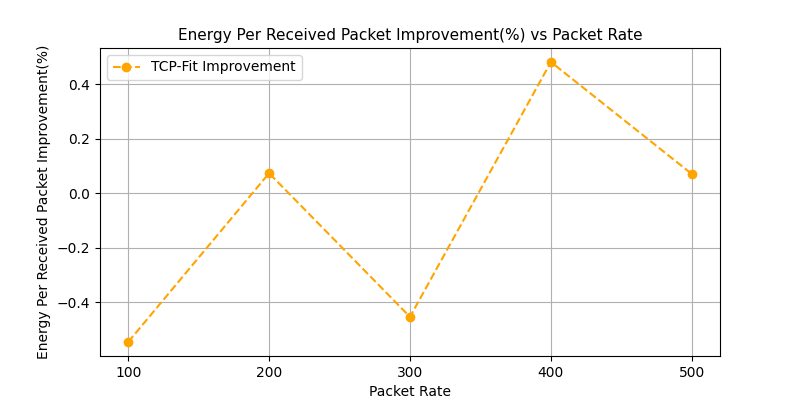








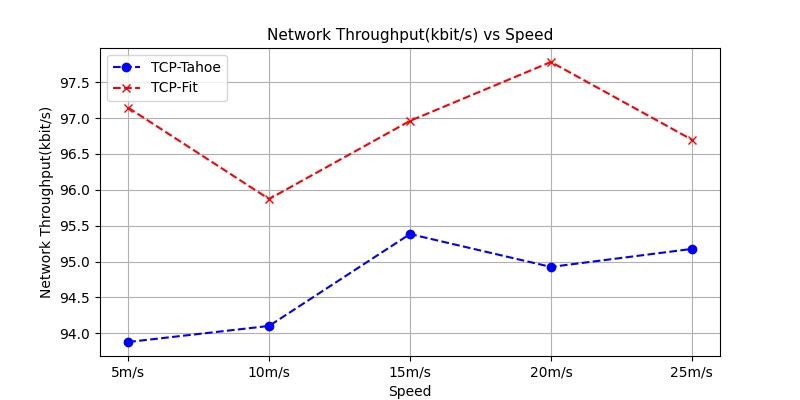


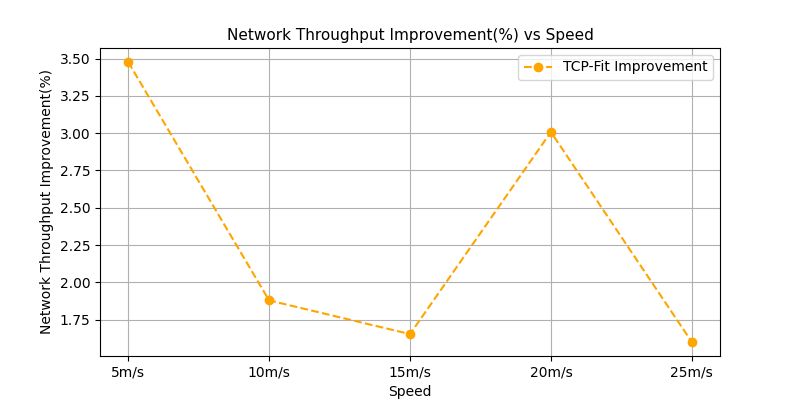


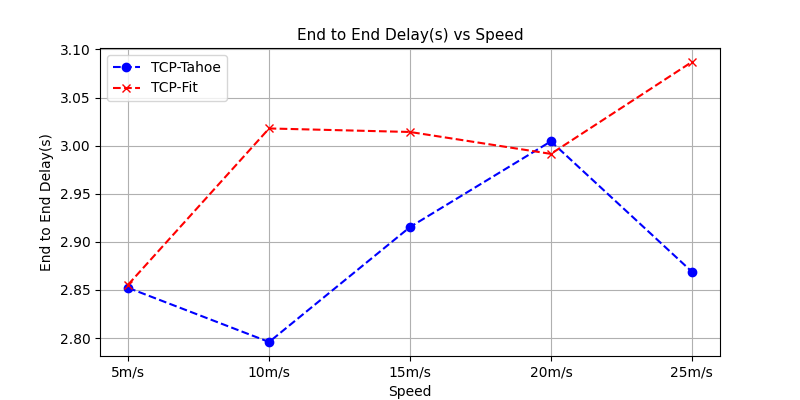
#### Observations

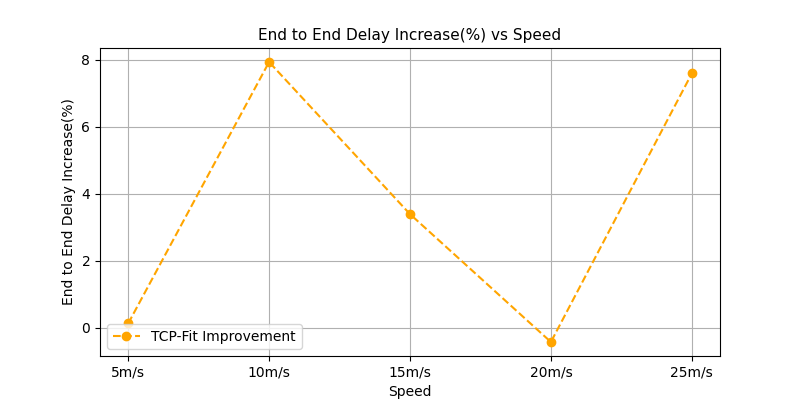
1. The metrics does not seem much affected by the packet rate indicating that bandwidth utilization for both algorithms were at their maximum capacity for all the packet rate
2. TCP-Fit performs very poorly. Its 1% increases in throughput comes at a cost of higher end to end delay, higher drop rate, lower delivery ratio

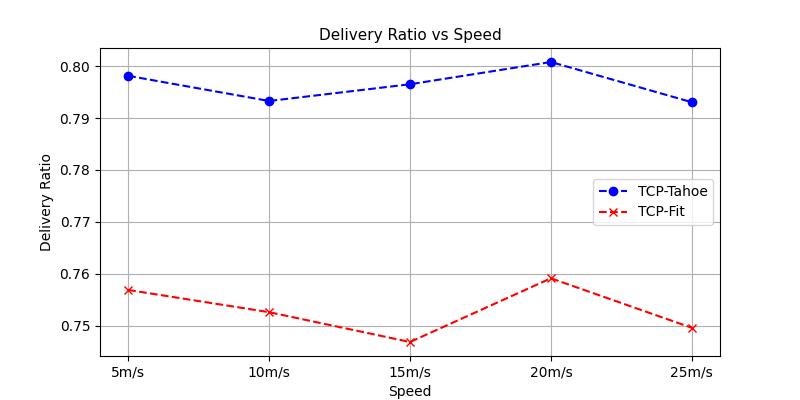
### Varying speed

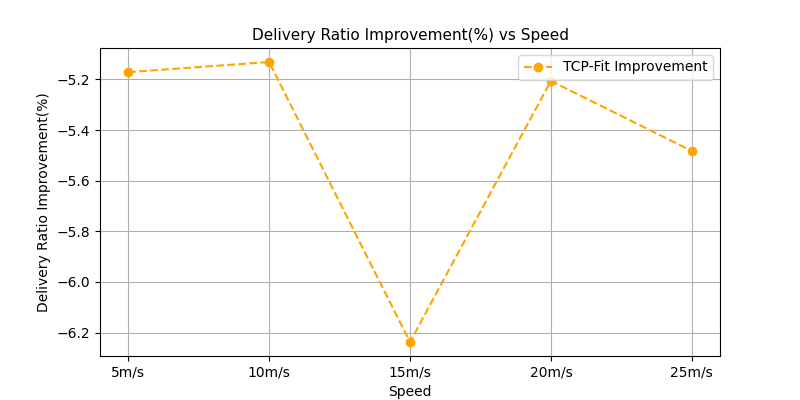


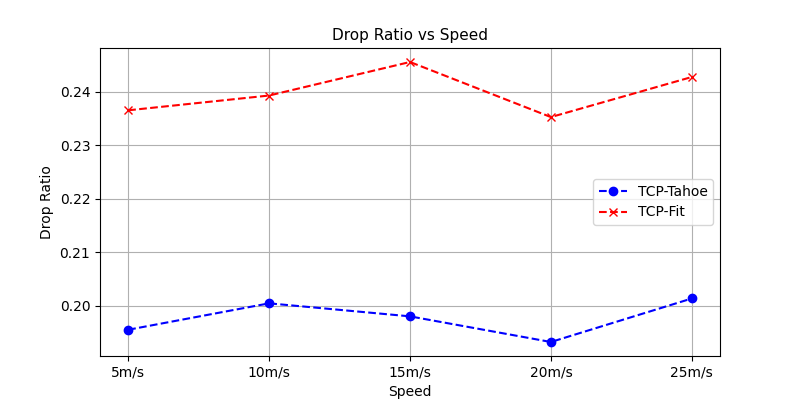


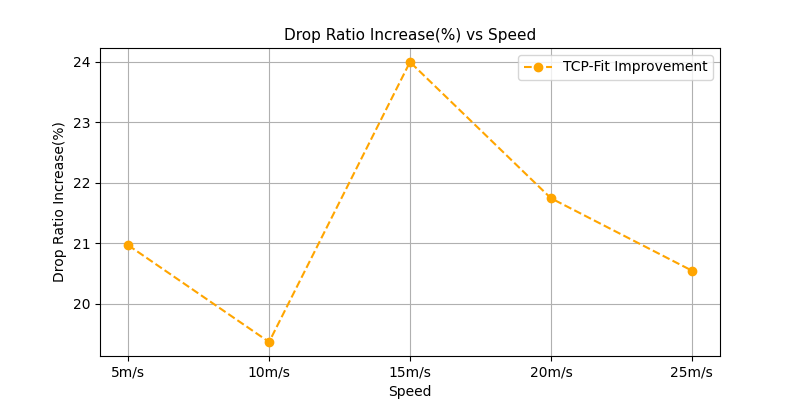


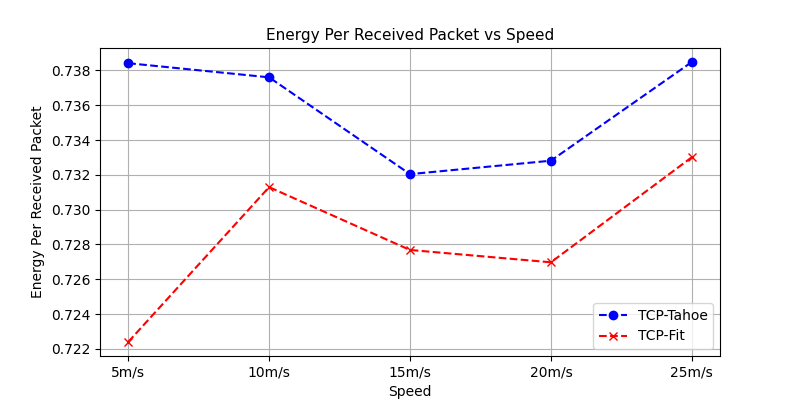


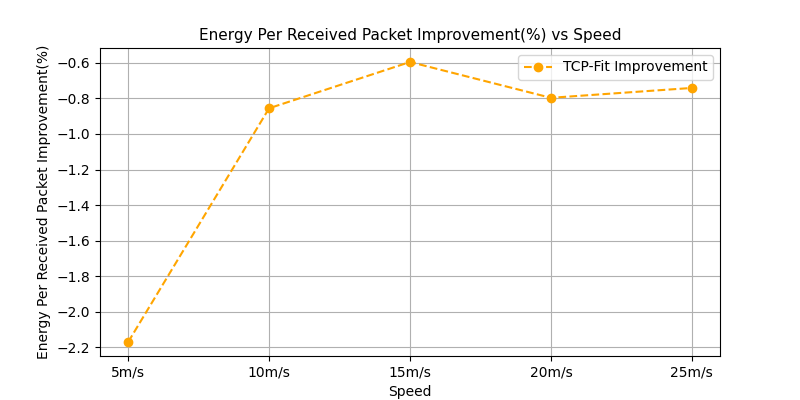










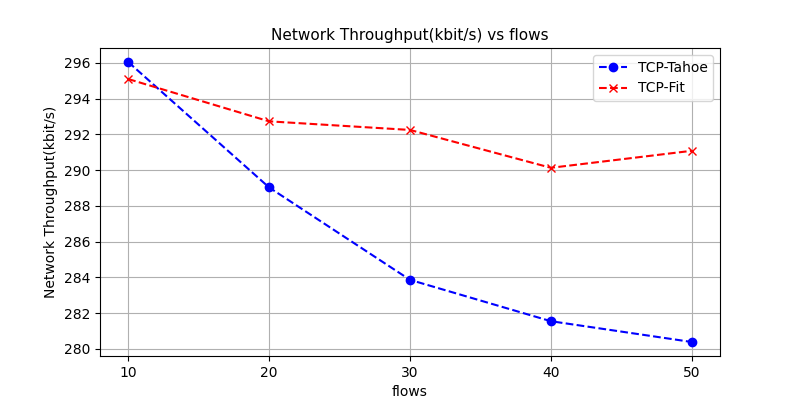


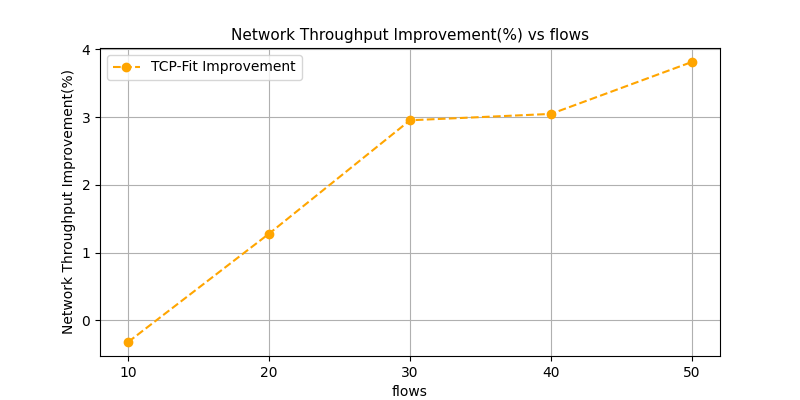
#### Observations

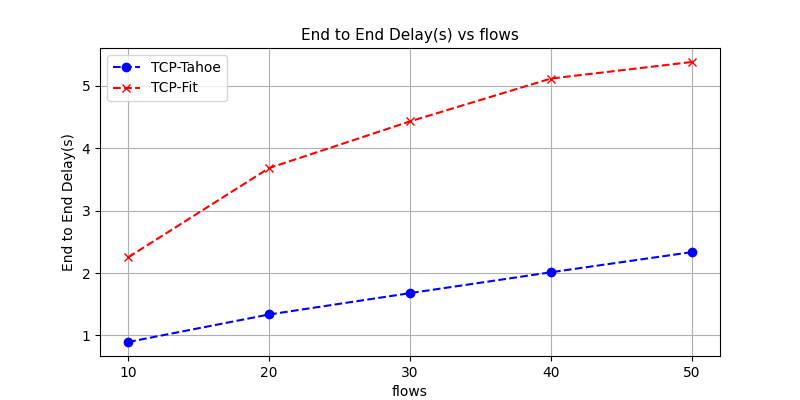
1. Again varying speed does seem to have very marginal difference on the metrics
2. TCP-Fit performs is poor overall

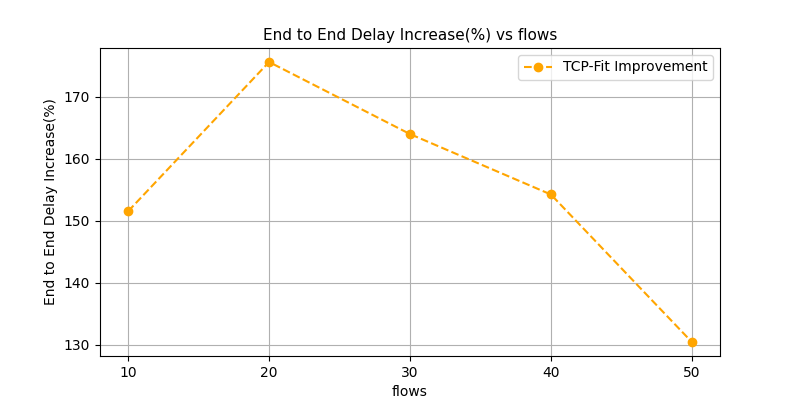
## Wireless Random

### Varying flows

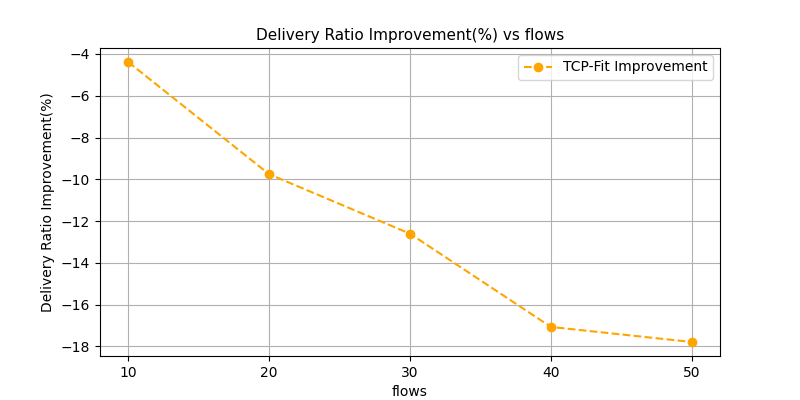


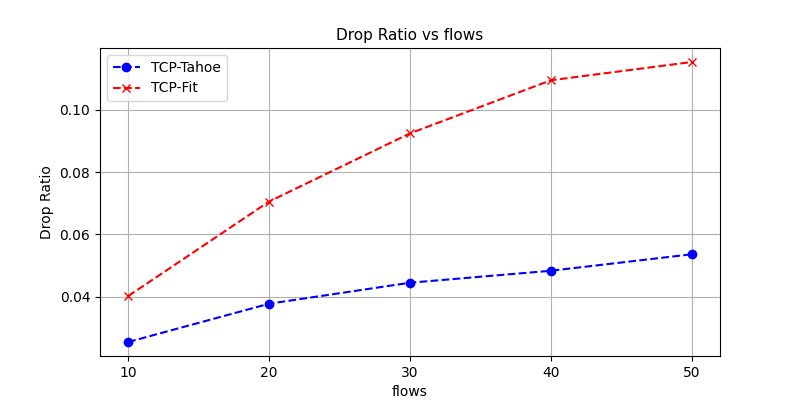


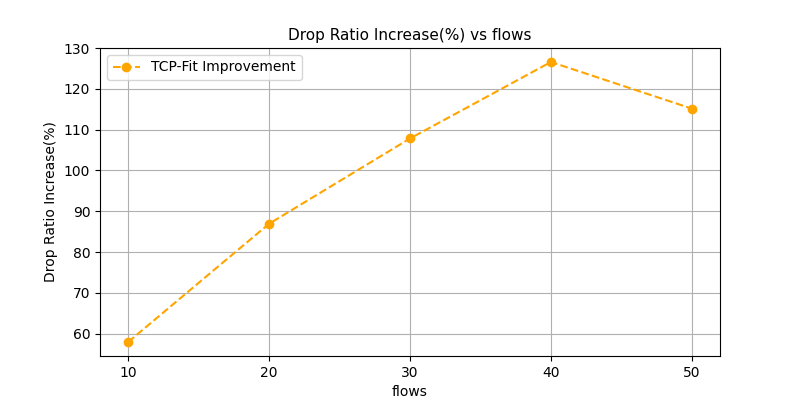








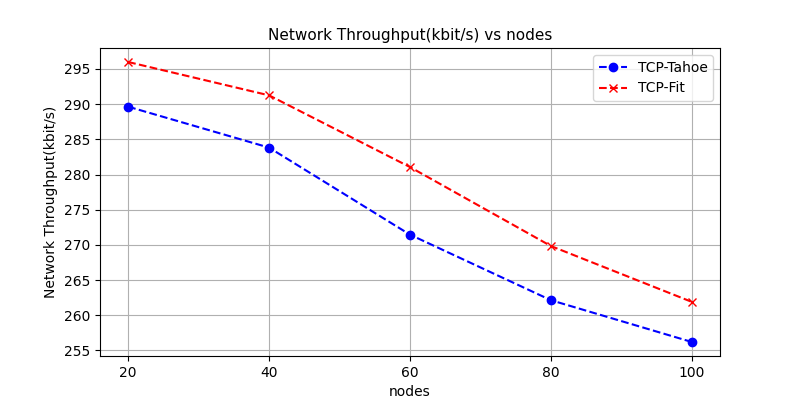


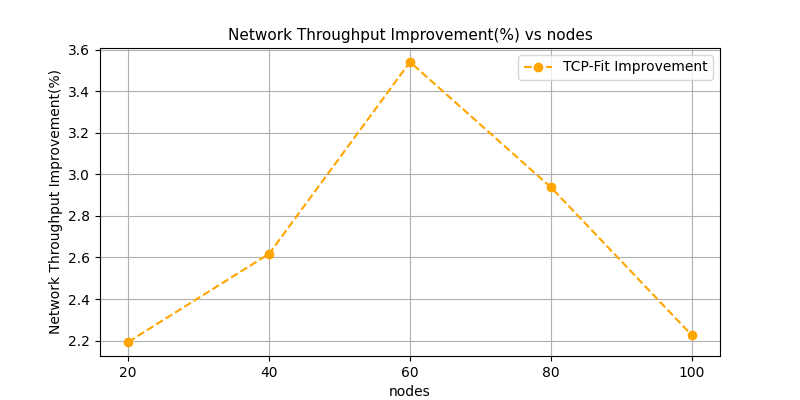


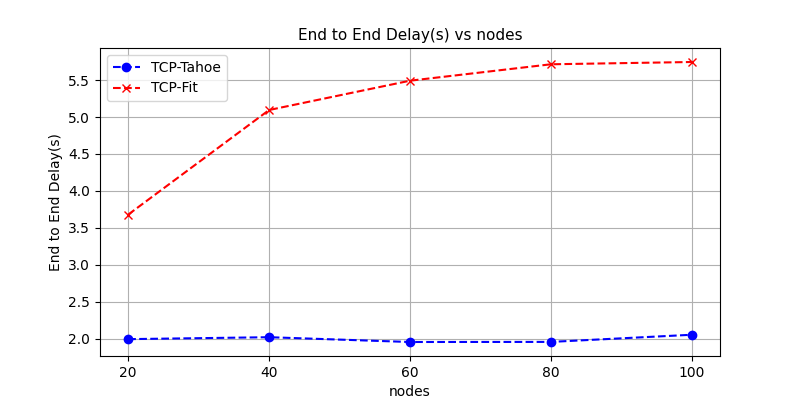
#### Observations

1. Throughput decreases with increasing flows indicating congested network. Slight improvement of throughput by TCP-Fit
2. Massive increase in end to end delay in TCP-Fit
3. Massive decrease of delivery ratio with TCP-Fit
4. Massive increase in drop rate as well with TCP-Fit

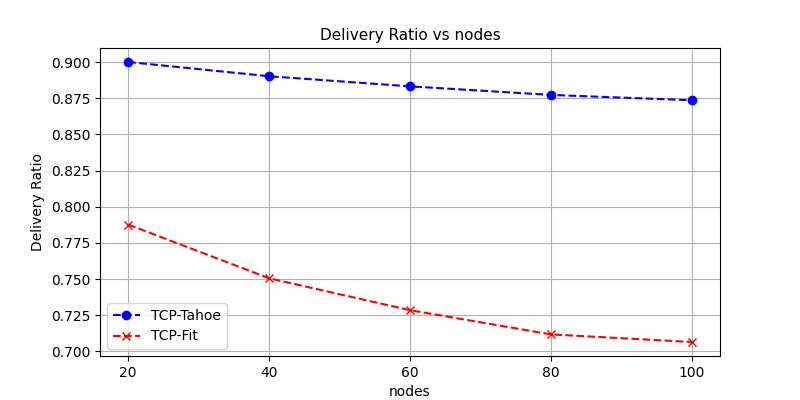
### Varying nodes

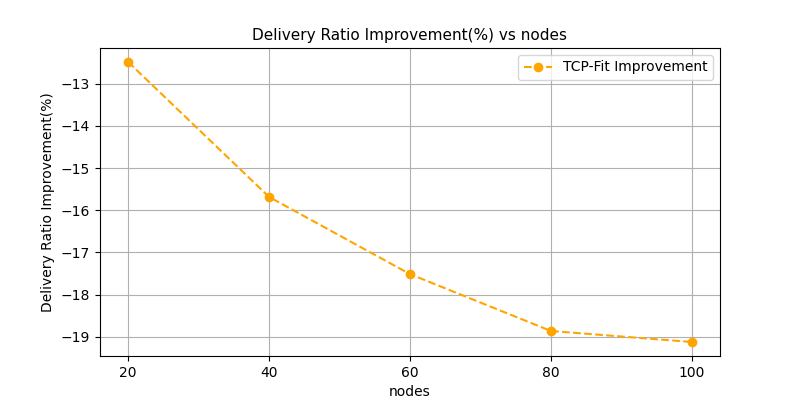
****

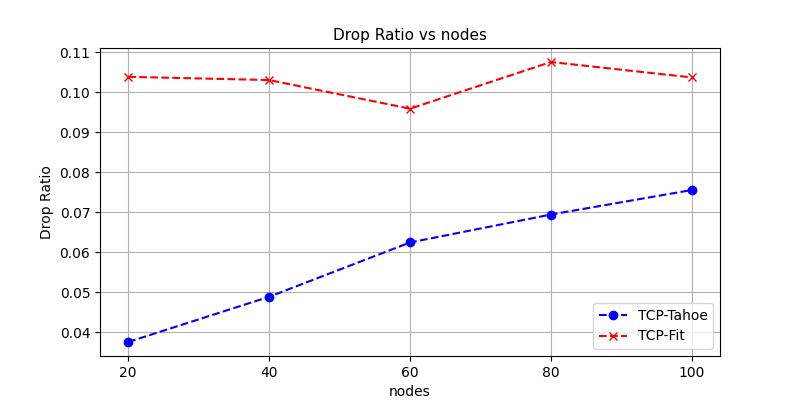
****

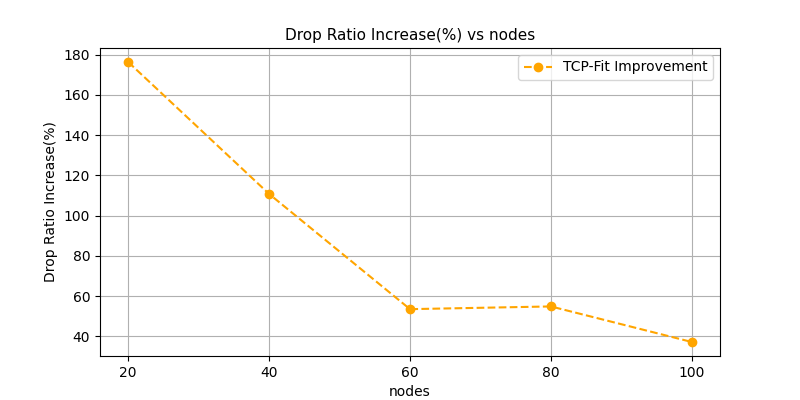
****

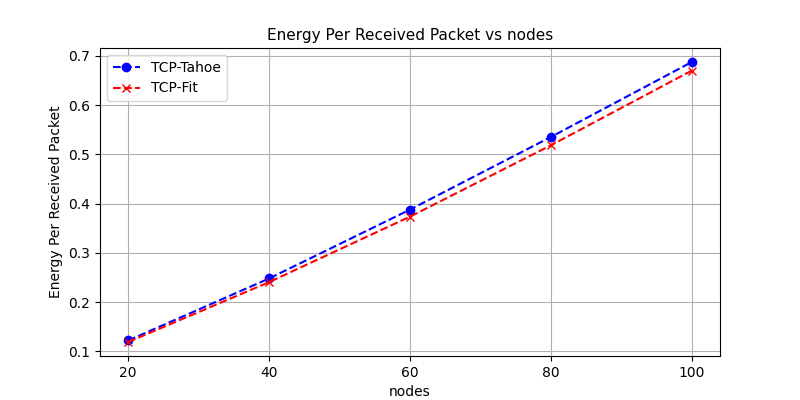
****

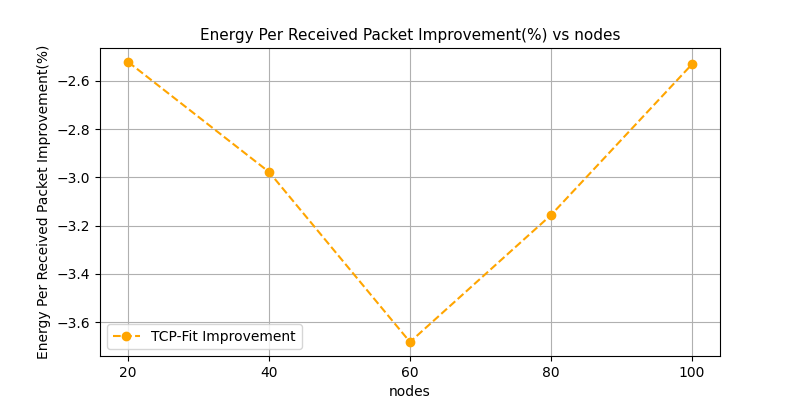
****

****

****

****

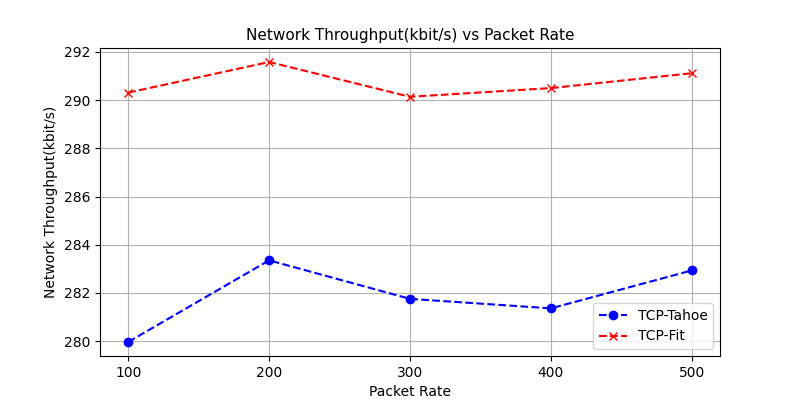
****

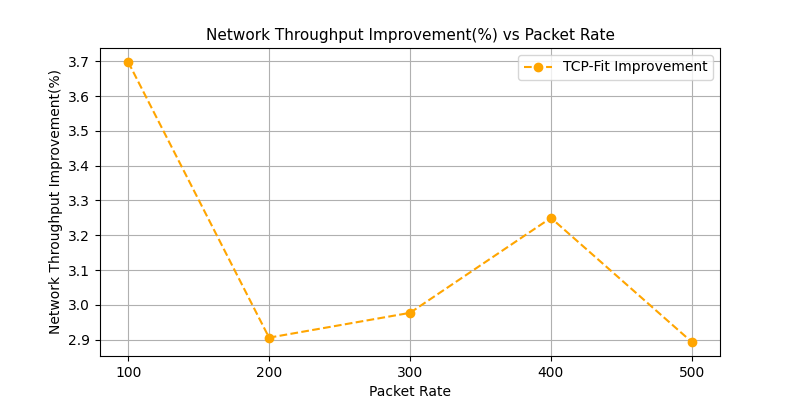
****

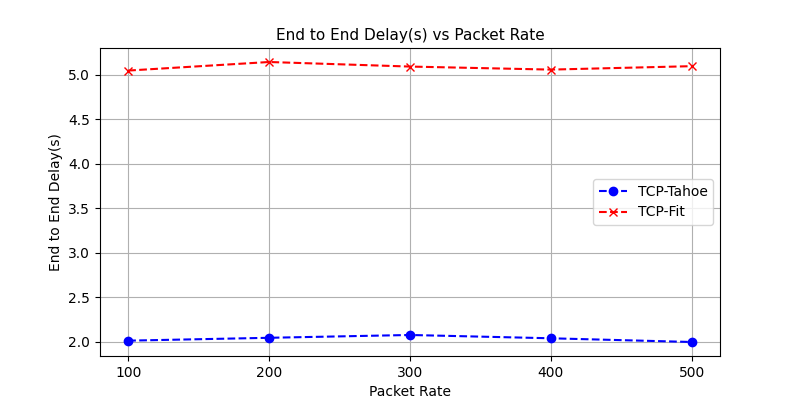
#### Observations

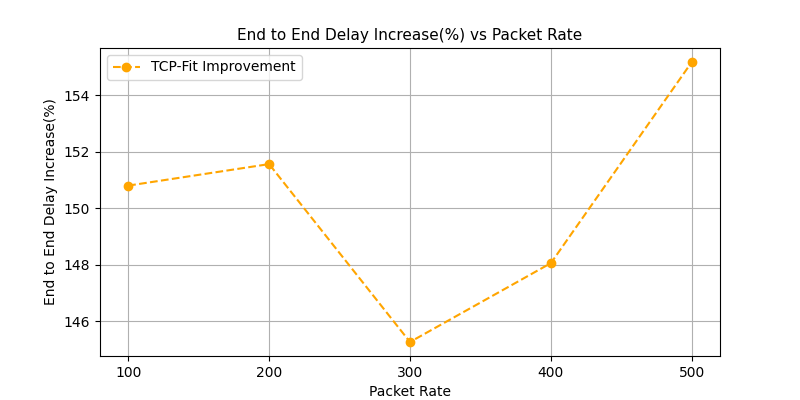
1. Throughput drops with increasing node numbers. The reasoning could be similar to WIreless-bottleneck topology
2. Massive increase in end to end delay for TCP-Fit algorithm
3. Delivery ratio worsens, and drop ratio increases for TCP-Fit algorithm

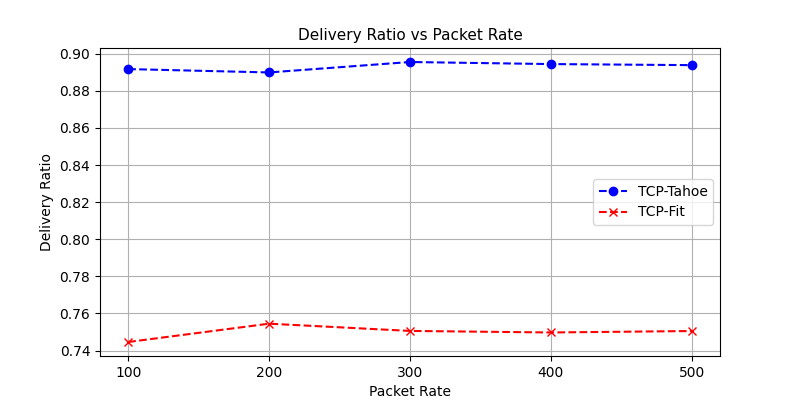
### Varying packet rate

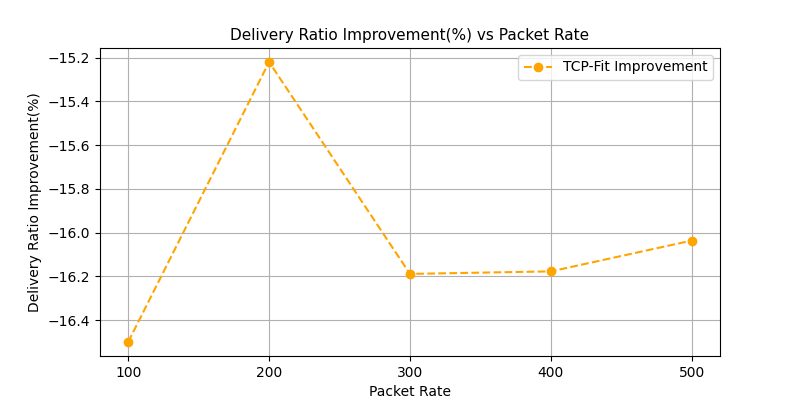
****

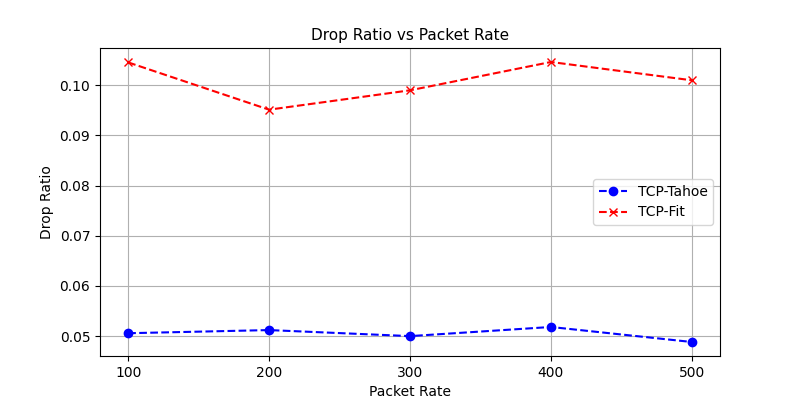
****

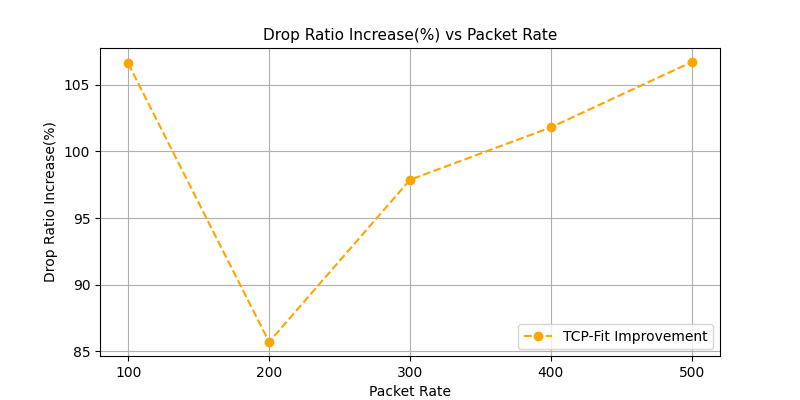
****

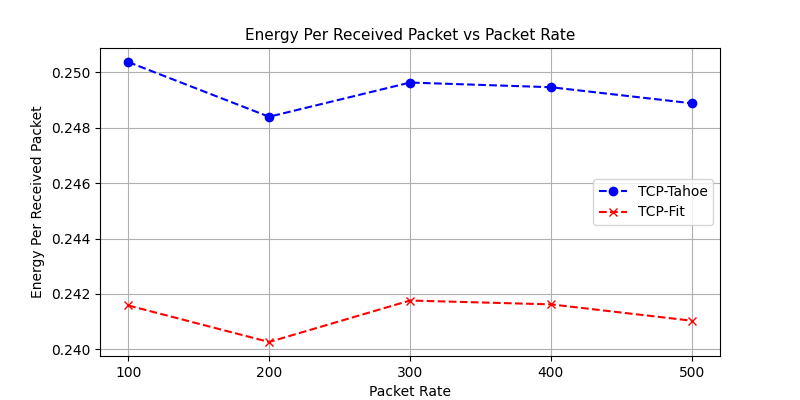
****

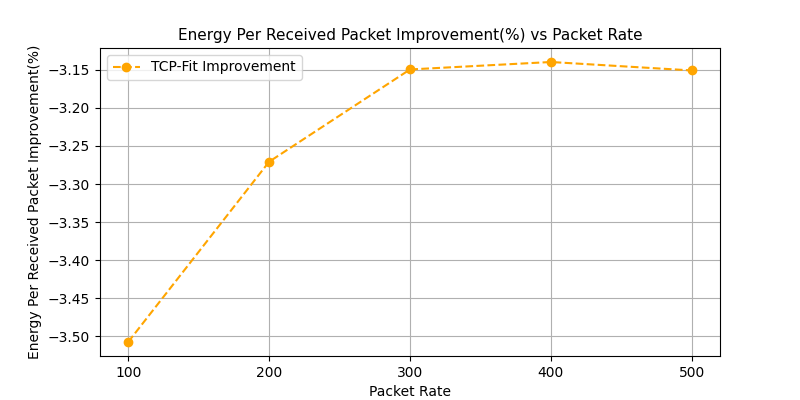
****

****

****

****

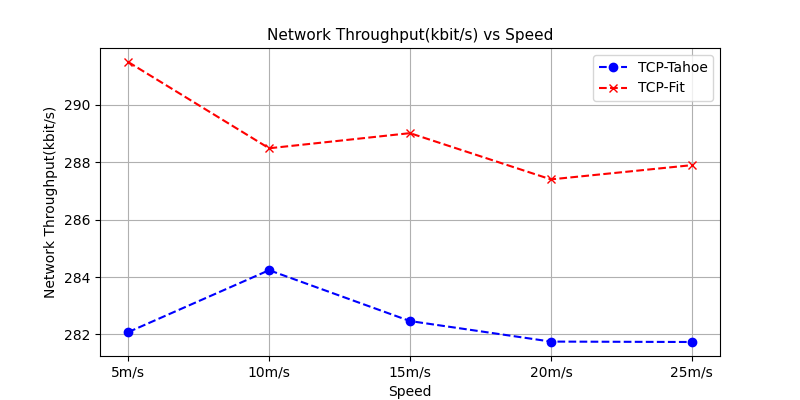
****

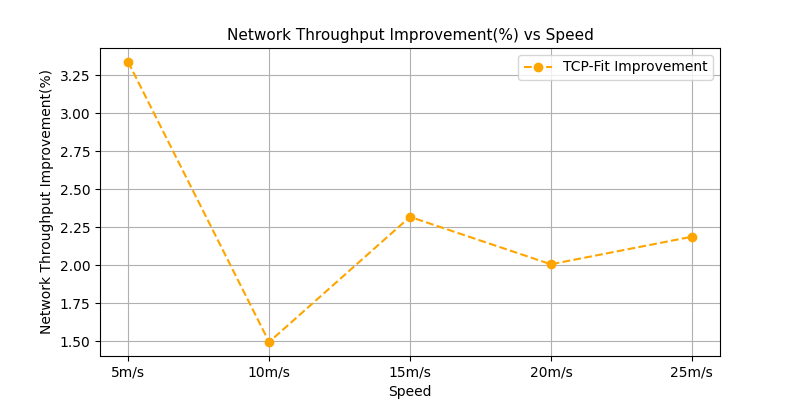
****

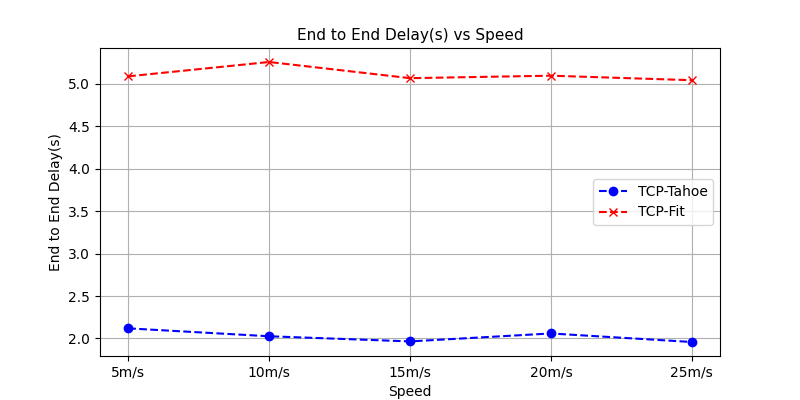
#### Observations

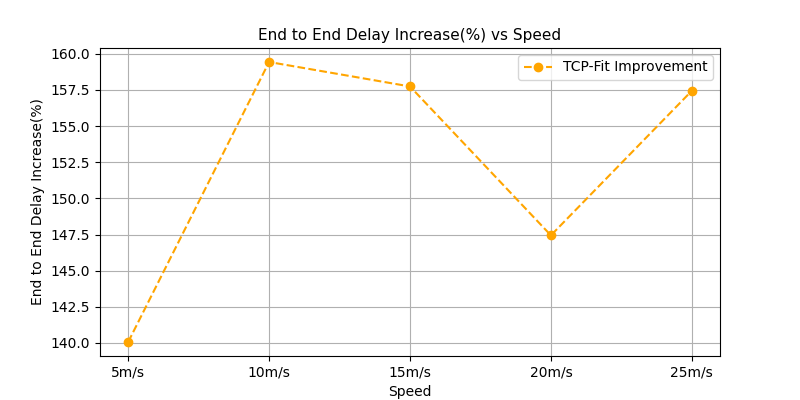
1. Again, metrics do not show any notable trend with change in packet rate of the chosen values
2. Again, quite worse performance of TCP-Fit in case of end to end delay, delivery ratio and drop ratio
3. Energy per packet is lower for TCP-Fit due to increased throughput

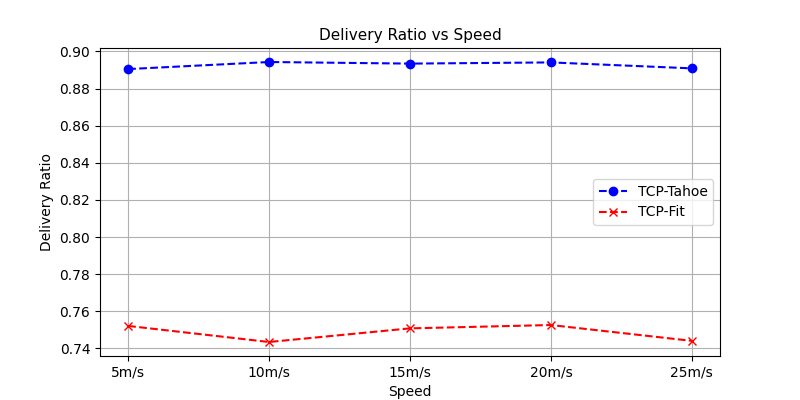
### Varying speed

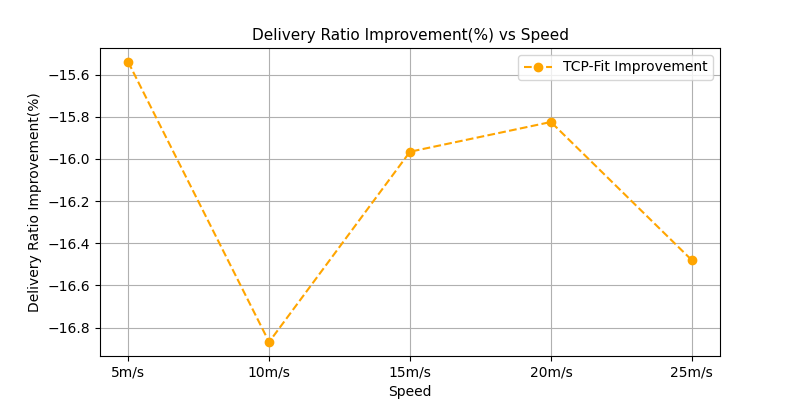
****

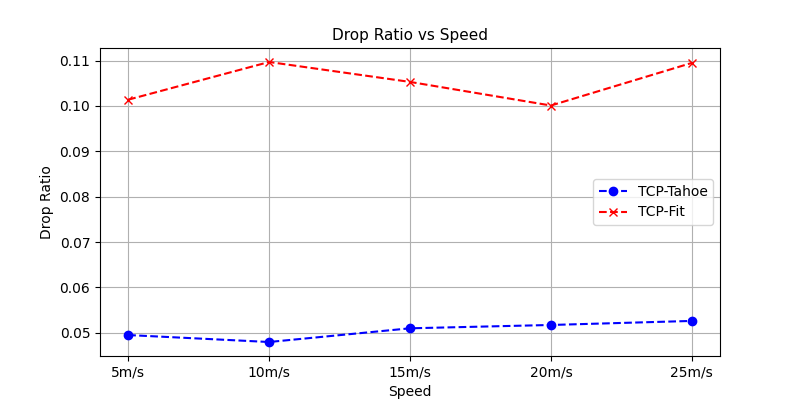
****

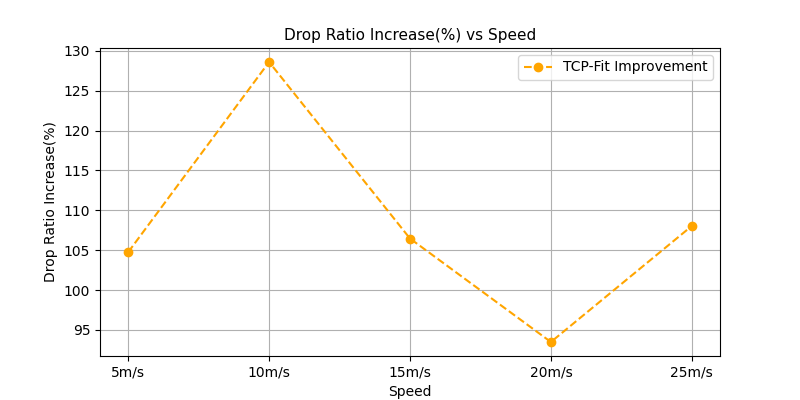
****

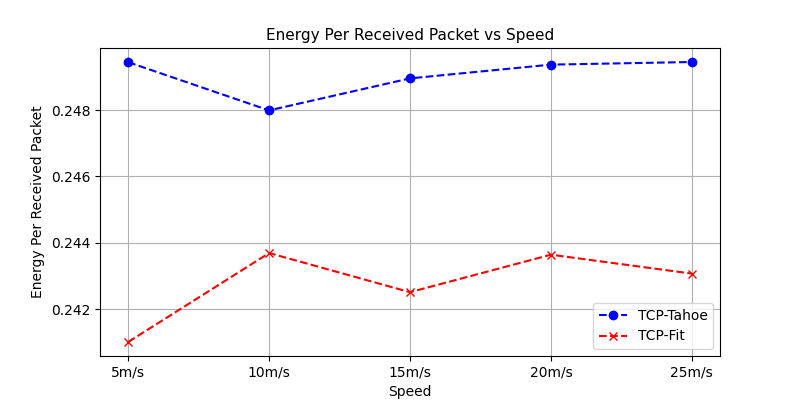
****

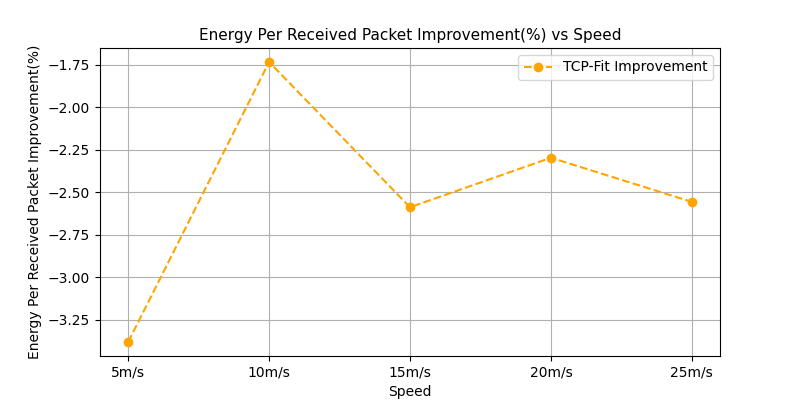
****

****

****

****

****

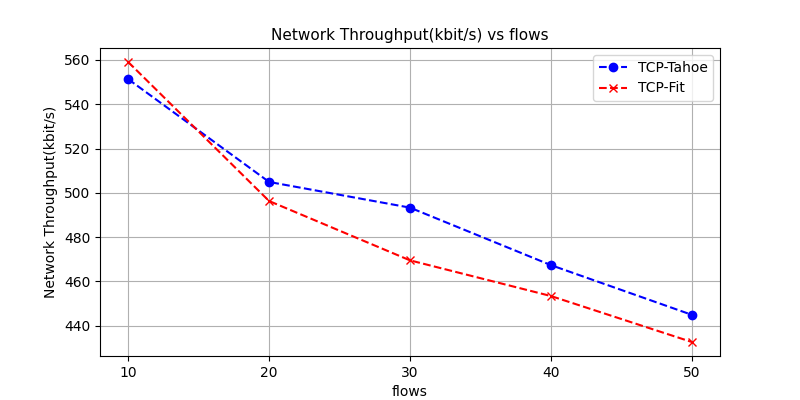
****

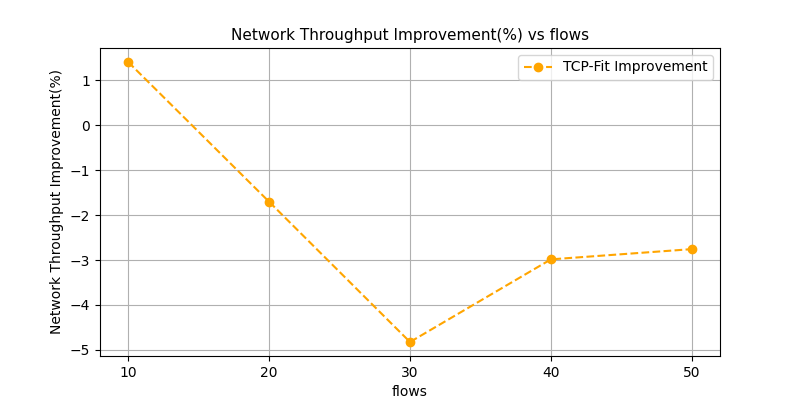
#### Observations

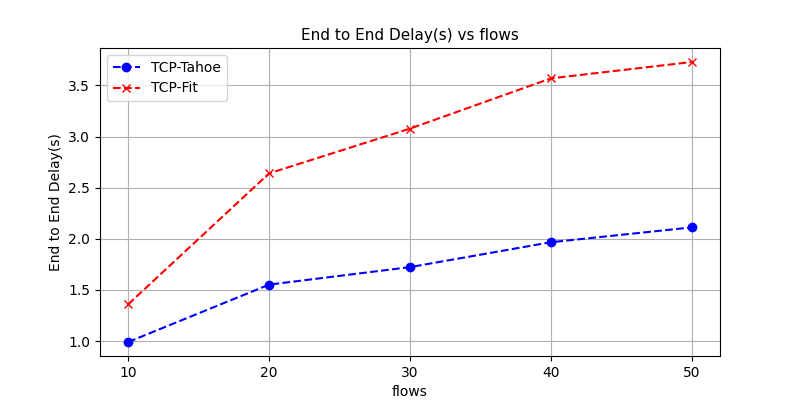
1. SImilar observations as previous topology

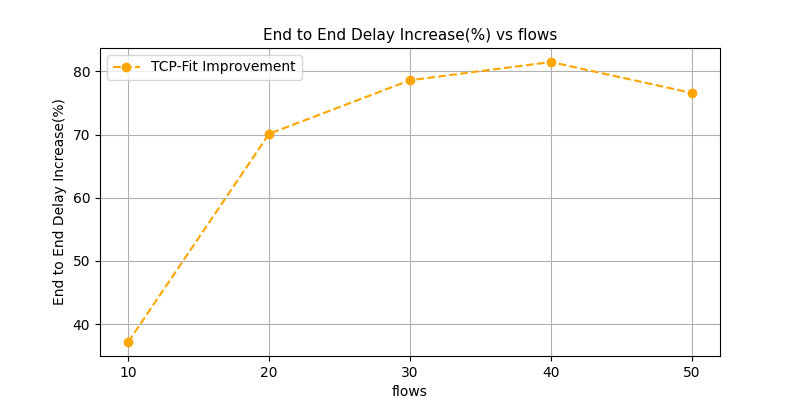
## Wireless-Grid

### Varying flows

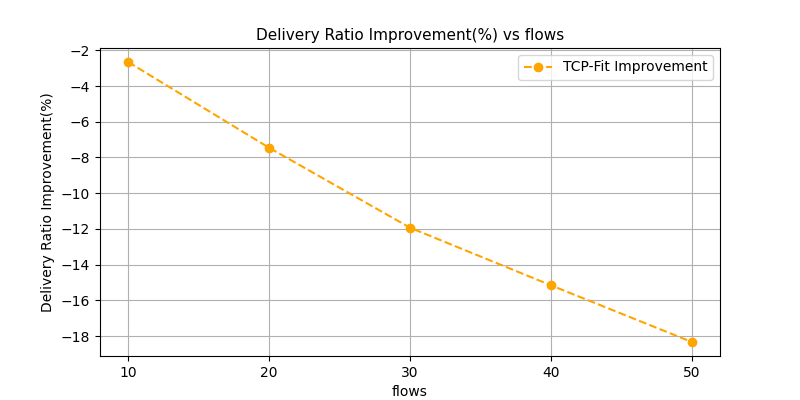
****

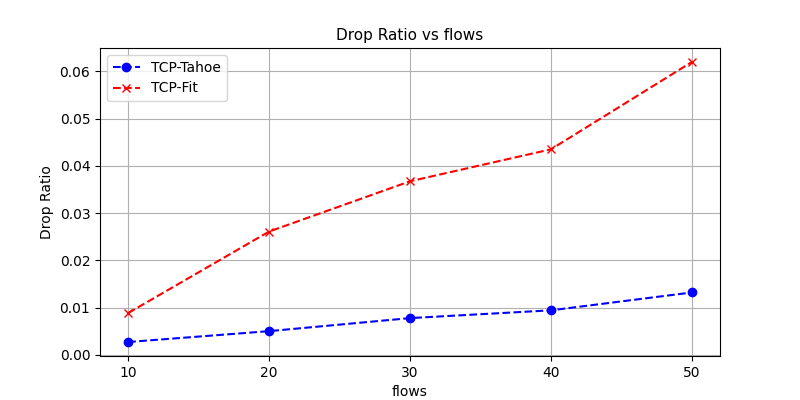
****

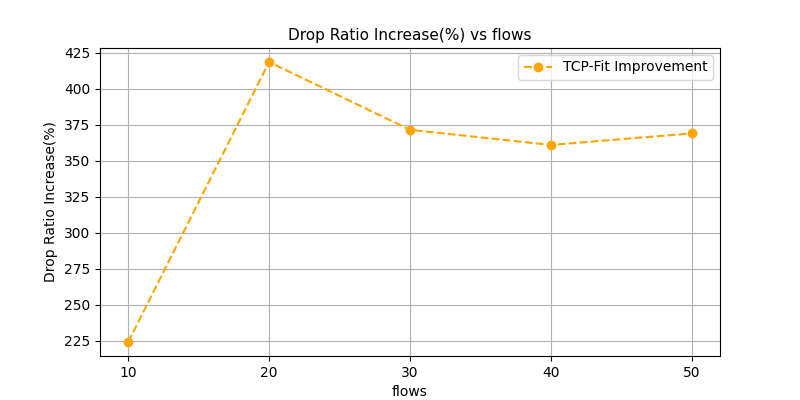
****

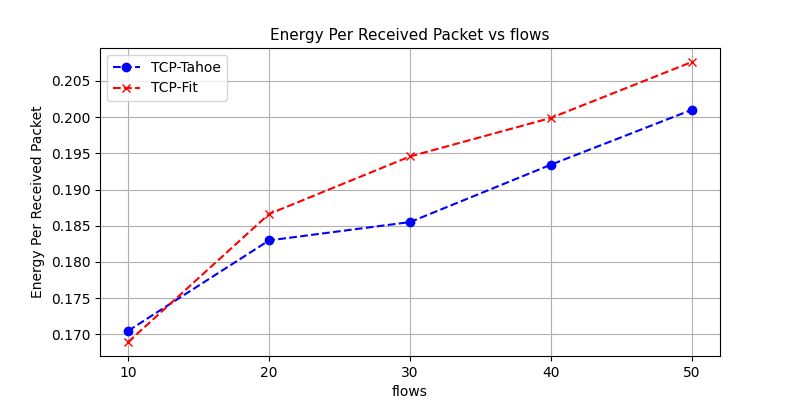
****

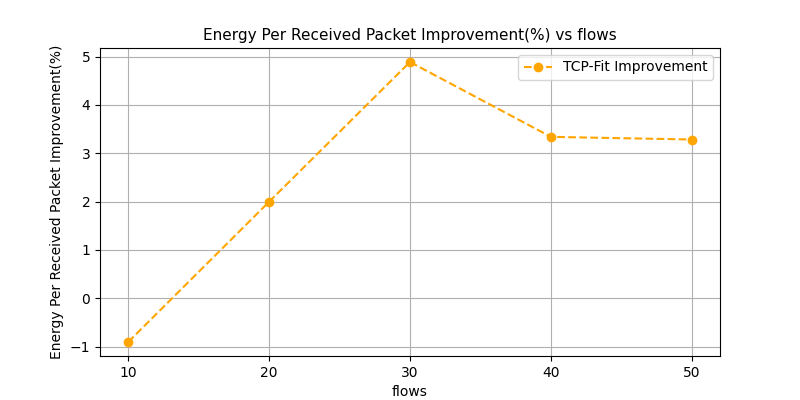
****

****

****

****

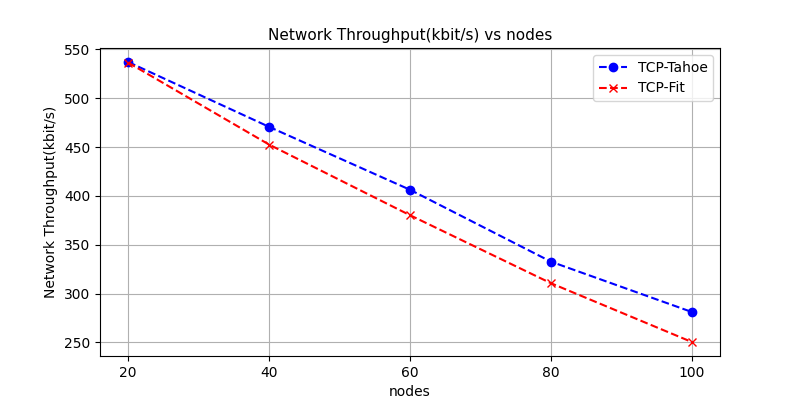
****

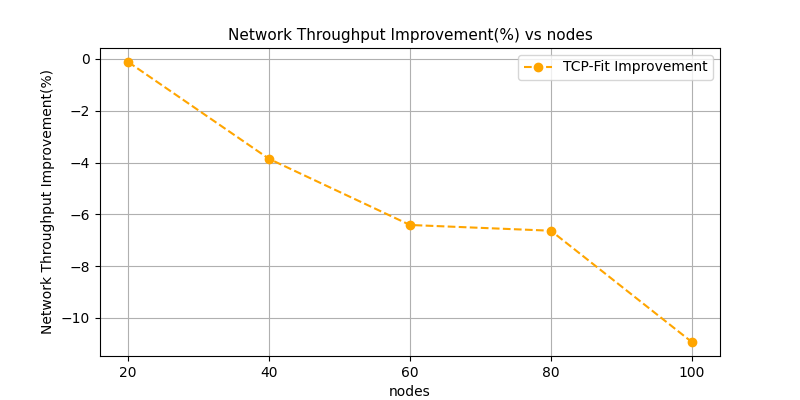
****

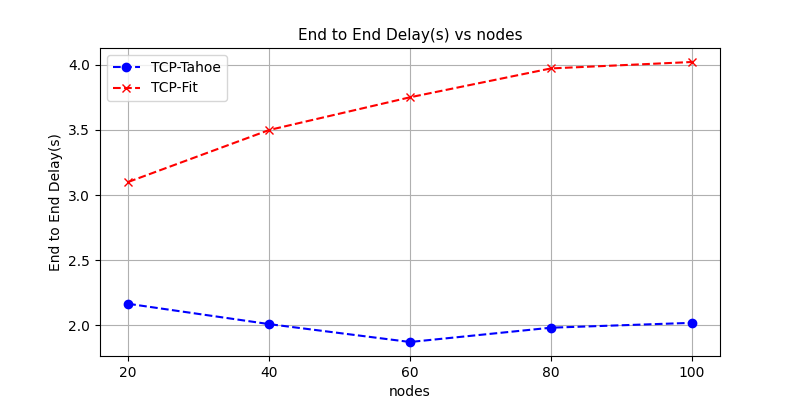
#### Observations

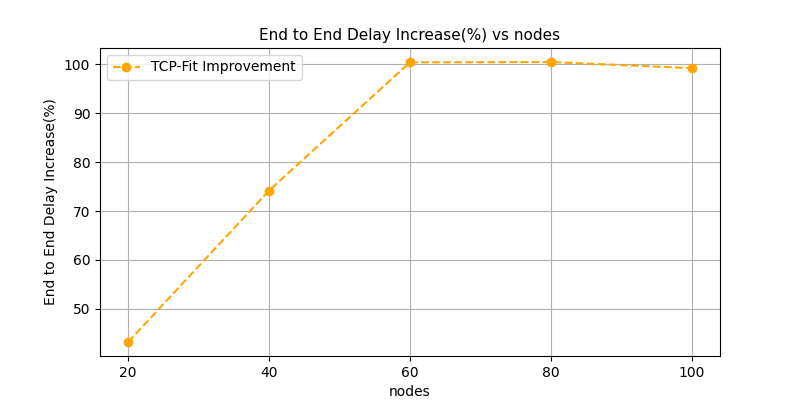
1. Worst performance of TCP-Fit seen so far. No improvement at all in any metric. Rather very poor performance in end to end delay, delivery ratio and drop ratio

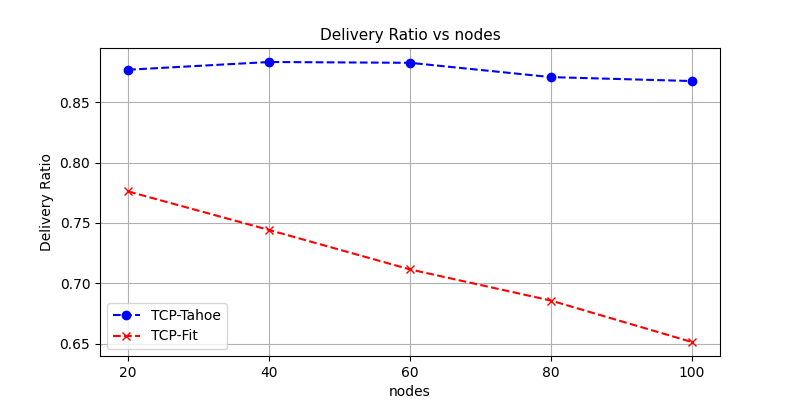
### Varying nodes

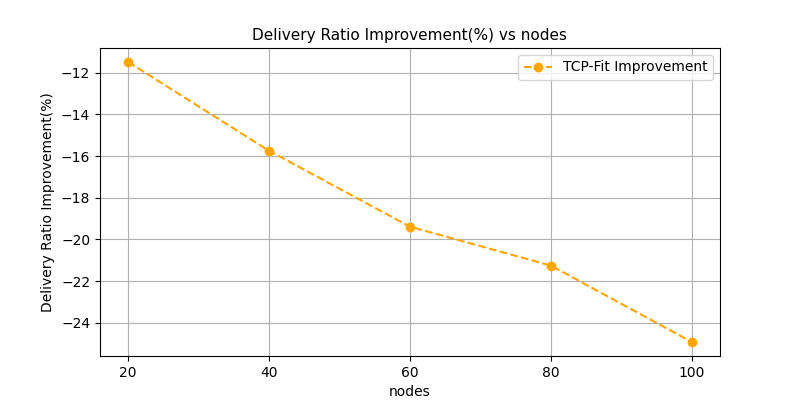
****

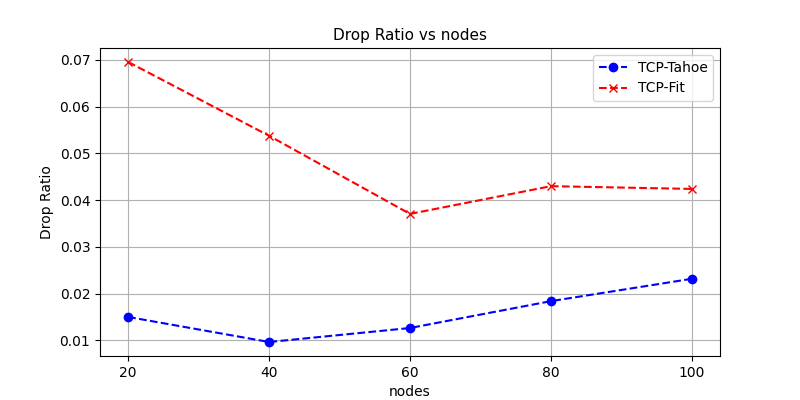
****

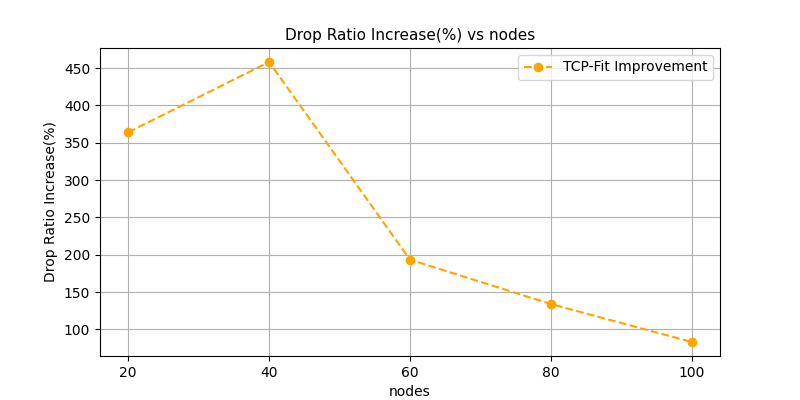
****

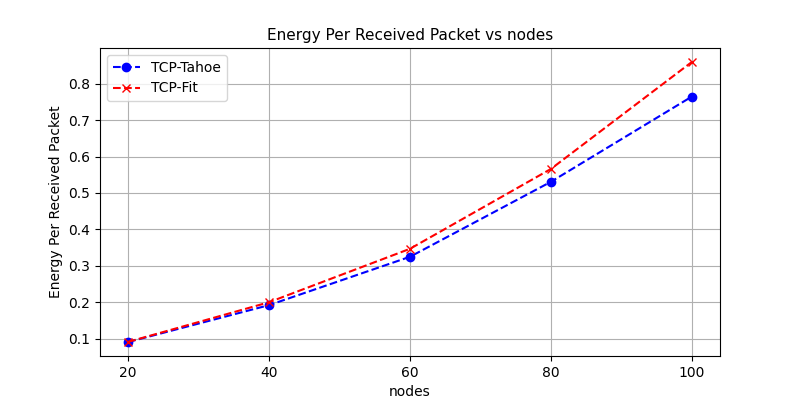
****

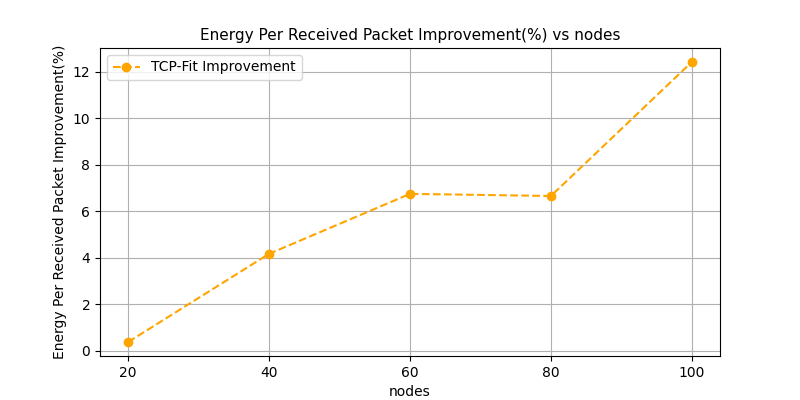
****

****

****

****

****

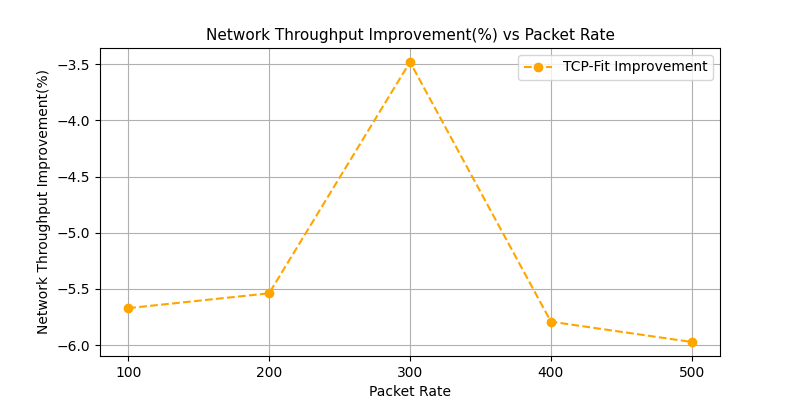
****

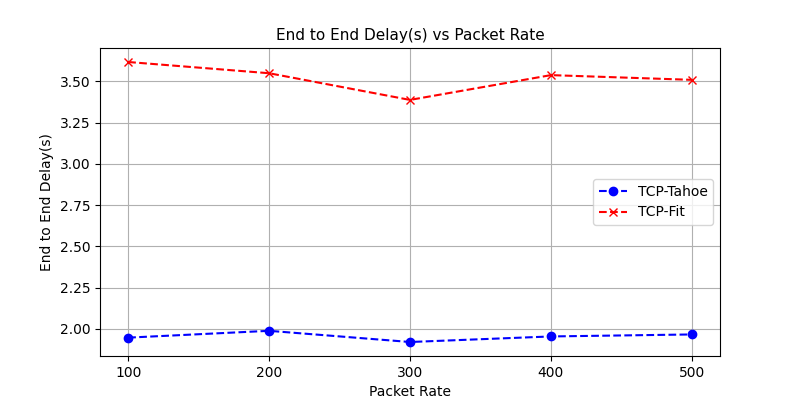
#### Observations

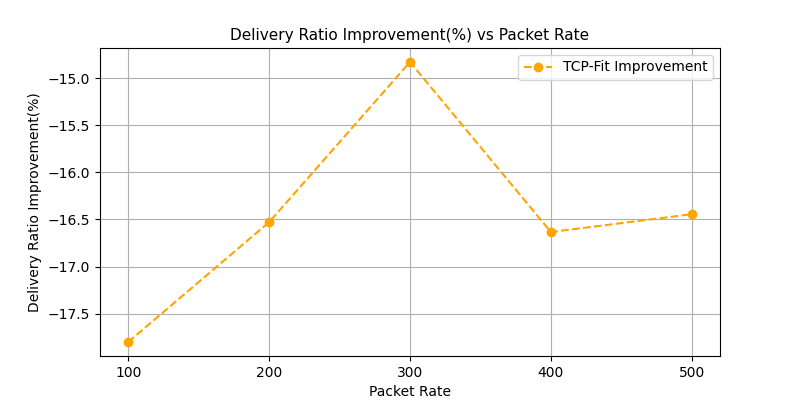
1. Similar observations as the previous topology
2. Very poor performance of TCP Fit algorithm

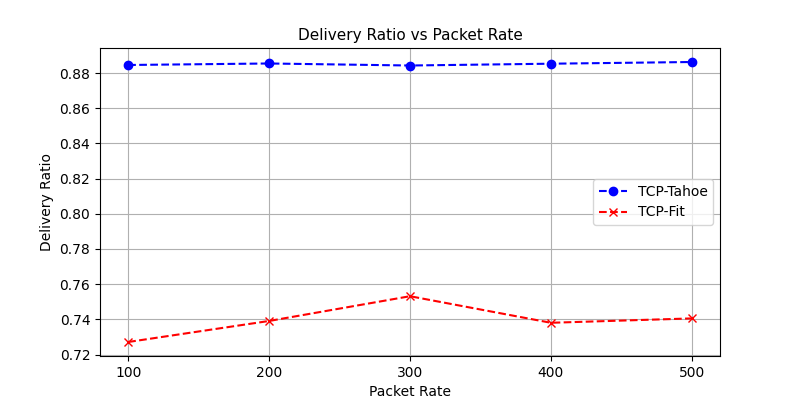
### Varying packet rate

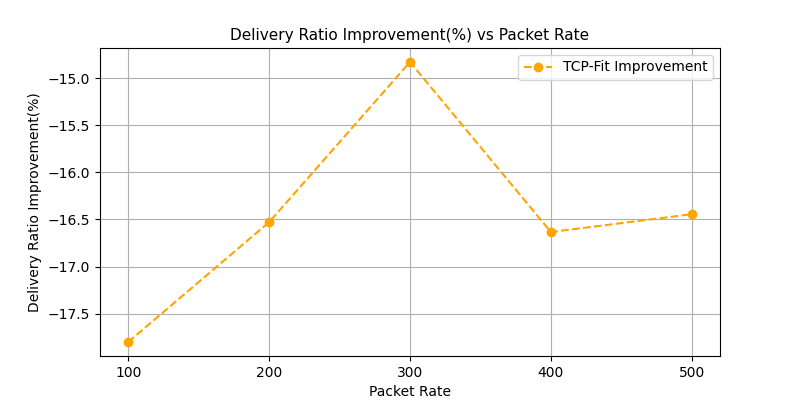
****

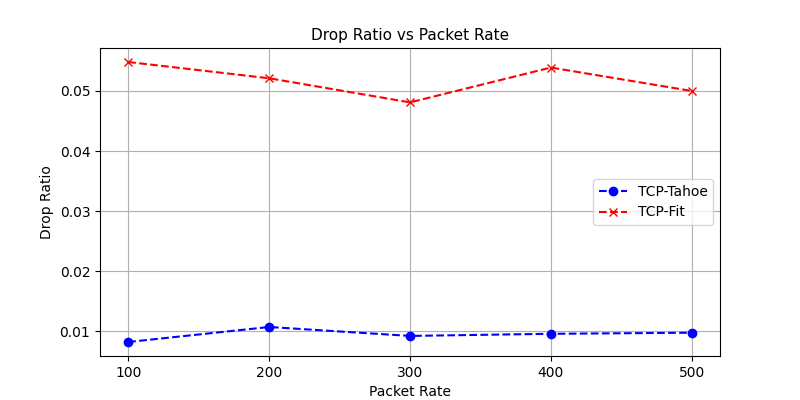
****

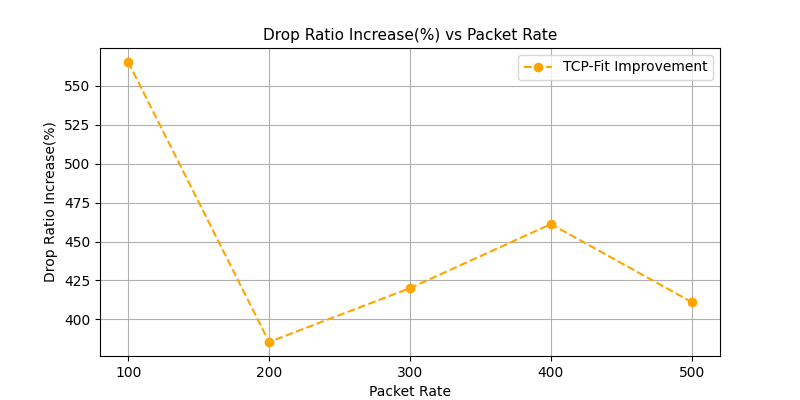
****

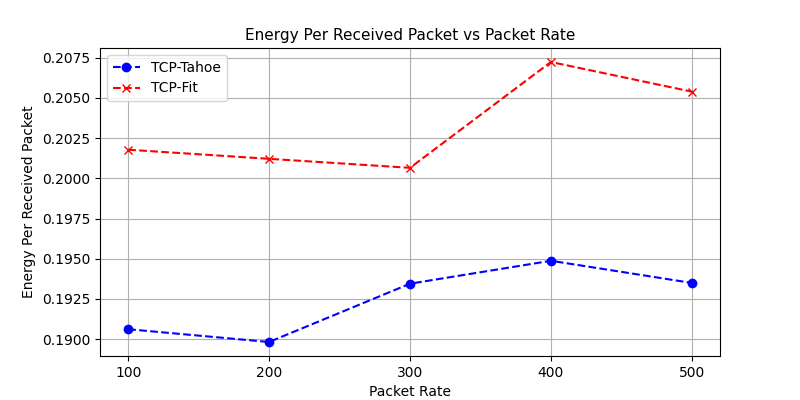
****

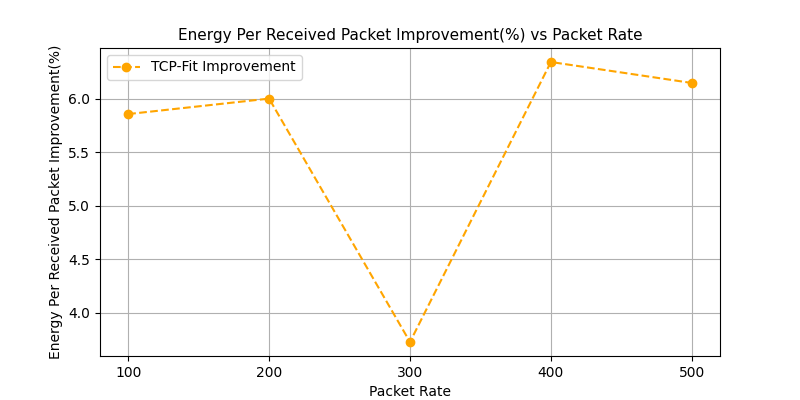
****

****

****

****

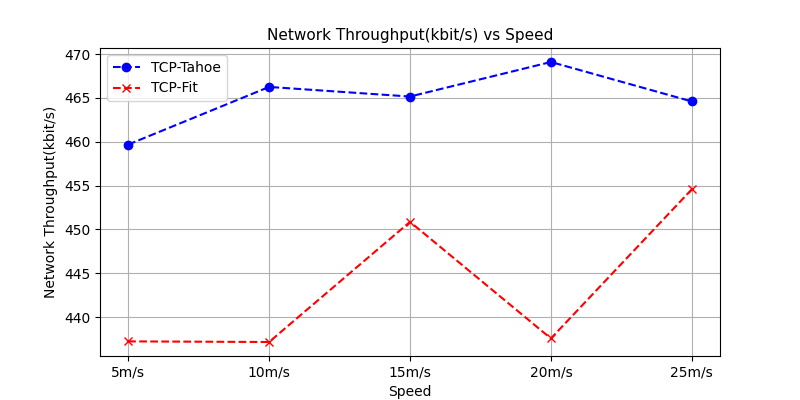
****

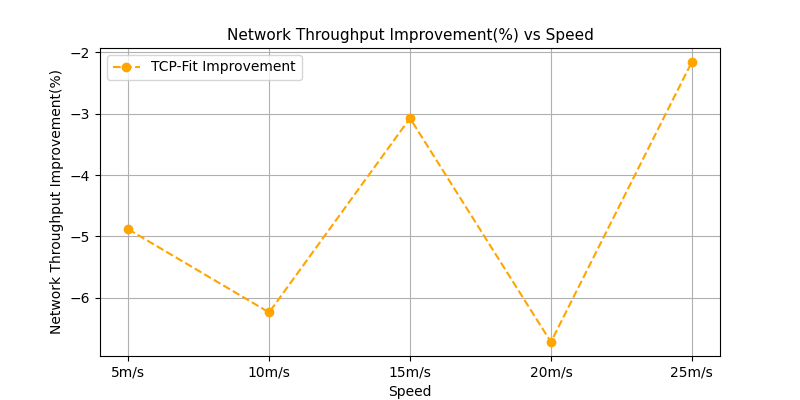
****

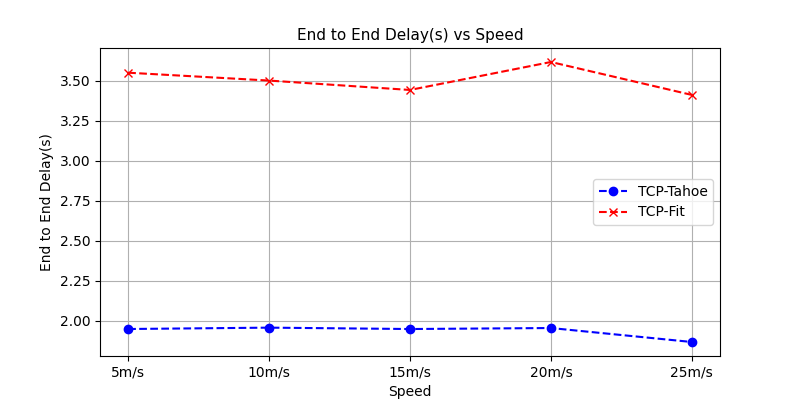
#### Observations

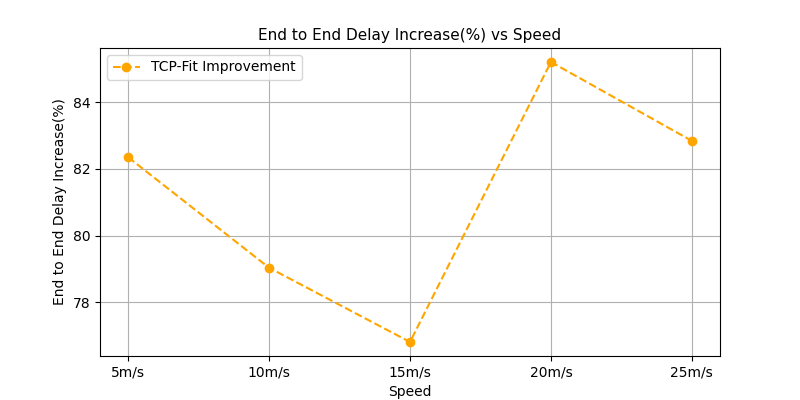
1. Similar observations as previous topology
2. TCP-Fit continues to show poor performance

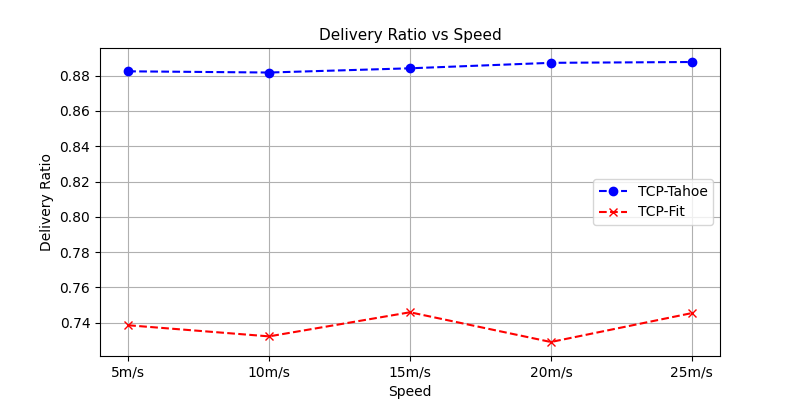
### Varying speed

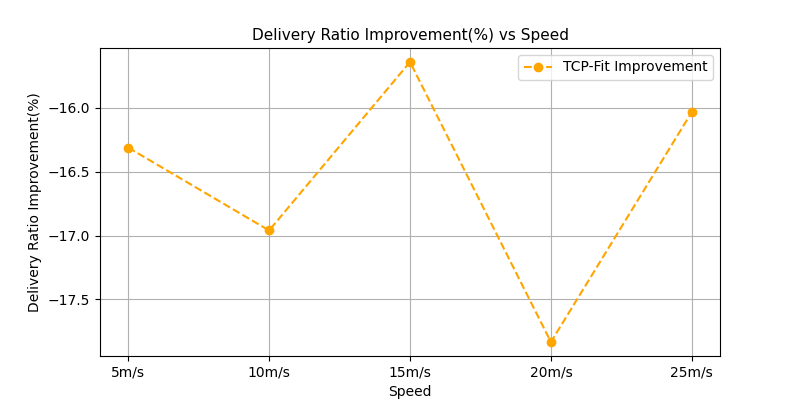
****

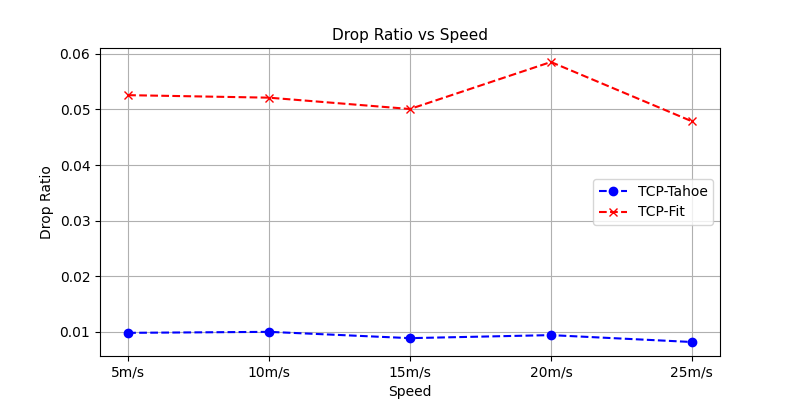
****

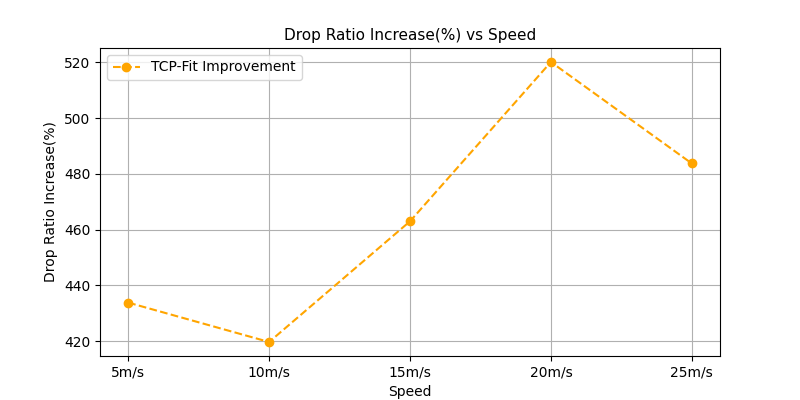
****

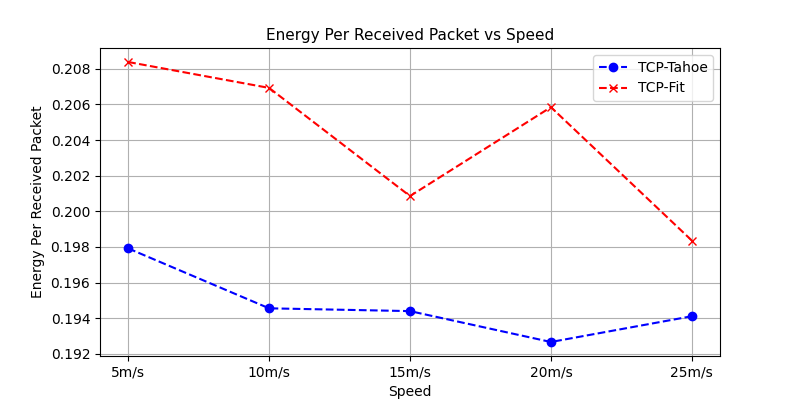
****

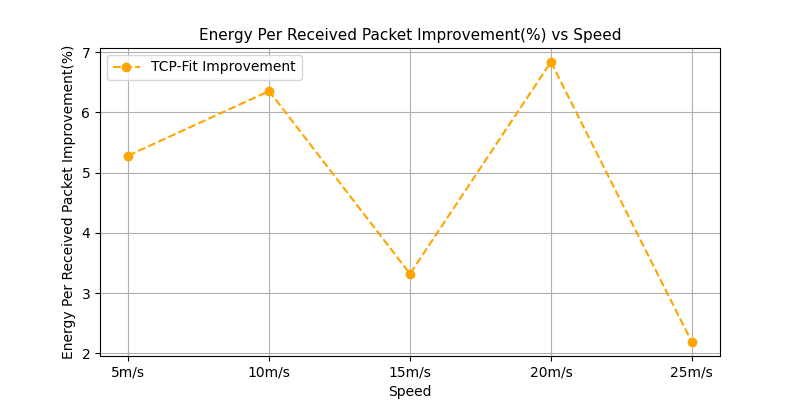
****

****

****

****

****

****

#### Observations

1. Similar observations as previous topology
2. Tcp Fit shows poor performance

# Summary Findings

The TCP-Fit algorithm seems to provide great improvement of throughput in networks where there are considerable amounts of packet drops that are not due to congestion. The wired and wired-wireless topology simulates these by adding random packet drop on the bottleneck link. We find a 30% to 50% increase in throughput with a very small increase in end to end delay.

However, the performance does not hold up in congested networks as seen in the Wireless networks. The throughput improvement is not as much (5% - 10%), sometimes even negative.

However the end to end delay can become very high (20%-190%), the delivery ratio can drop quite low and drop ratio can become very high compared to TCP-Tahoe.

The algorithm seems to fail to detect that a network is congested. It tries to send more packets in a congested network and so end to end delay, delivery ratio, drop ratio all worsens. The more the network is congested, the worse TCP-Fit seems to perform.

In summary

1. The algorithm works well in topology considered in the paper, scenarios with packet drops that are not caused by congestion but rather due to the medium,
2. The algorithm does not perform that well in congested networks, it can barely increase the throughput, but increases end to end delay, drop ratio and worsens delivery ratio
3. The algorithms’ performance seems topology dependent. It fails to perform well in other wireless topology such as random or grid topology. So the algorithm does not generalize well.