DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, BUET

A Machine Learning Approach to End-to-End RTT Estimation and its Application to TCP

CSE 322 Project Report

A.H.M.Osama Haque
1805002

February 27, 2023

1 INTRODUCTION

The instantaneous smoothed RTT (RTT estimation) and the RTT variation are calculated in ns2 source code as follows:
Let $t(k)$ be the kth RTT sample collected upon ACK reception. Also, let $\bar{t}(k)$ and $\sigma_t(k)$ be the kth RTT estimation and kth variance. Then,

$$\bar{t}(k+1) = \alpha * \bar{t}(k) + (1-\alpha) * t(k+1) \tag{1.1}$$

$$\sigma_t(k+1) = \beta * \sigma_t(k) + (1-\beta) * |t(k+1) - \bar{t}(k)| \tag{1.2}$$

In my selected paper titled **"A Machine Learning Approach to EndtoEnd RTT Estimation and its Application to TCP"**[1], a novel RTT estimation technique is proposed that uses a machine-learning based approach called the Experts Framework. The Experts Framework uses "on-line" learning, where the learning process happens in "trials". At every trial, a number of "experts" contribute to an overall prediction, which is compared to the actual value (e.g., obtained by measurement). The algorithm uses the prediction error to refine the weights of each expert's contribution to the prediction; the updated weights are used in the next iteration of the algorithm. We contend that by employing our prediction technique, network applications and protocols that make use of the RTT do not have to measure it as frequently and can rely on our predictions.

2 PROPOSED ALGORITHM

Parameters: $\eta > 0$ and $0 \le \alpha \le 1$
Initialization: $w_{1,1} = ... = w_{1,N} = \frac{1}{N}$
Steps:

- Prediction : $\hat{y}_t = \frac{\sum_{i=1}^{n} w_{t,i} * x_i}{\sum_{i=1}^{n} w_{t,i}}$
- Computing the Loss : $\forall i \in \{1, 2, \dots, N\}$

$$L_{i,t}(x_i, y_t) = \begin{cases} (x_i - y_t)^2 & x_i \ge y_t \\ 2 * y_t & x_i < y_t \end{cases}$$

- Exponential Updates : $\forall i \in \{1, 2, \dots, N\}$

$$w'_{t,i} = w_{t,i} * e^{-\eta L_{i,t}(y_t, x_i)}$$

- Sharing Weights : $\forall i \in \{1, 2, \dots, N\}$

$$pool = \sum_{i=1}^{N} \alpha * w'_{t,i}$$

$$w_{t+1,i} = (1-\alpha) * w'_{t,i} + \frac{1}{N} * pool$$

3 Network Topologies

My ID is 1805002 (%8 = 2). I am assigned the following topologies:

- Wireless 802.11 (mobile)
- Wireless 802.15.4 (mobile)

3.1 Wireless 802.11 (mobile)

In this scenario, every node is equipped with a wifi network device. Each wifi device maintains IEEE 802.11b standard. Every node operates as AdhocWifi mode. In the network layer, Internet protocol version 4 is used. Ad-hoc On demand Distance Vector (AODV) routing protocol is installed in the network layer. TCP is used as transport later protocol. In the application layer, FTP is used for generation of packets. Client and server applications are bound to ports such that there is a specific number of flow generated in the simulation.

3.2 Wireless 802.15.4 (mobile)

Here every node has a device that supports low rate wireless personal area communication. Each device maintains IEEE 802.15.4 standard.In the Application layer I have used the same method as earlier. For mobility and position allocation the previous mobility model is used.

4 Variation of Parameters

In the simulation the following parameters are considered:

- **Number of nodes :** Varied as 20, 40, 60, 80, and 100
- **Number of flows :** Varied as 10, 20, 30, 40, and 50
- **Number of packets per second** : Varied as 100, 200, 300, 400, and 500
- **Speed of nodes** : Varied as 5, 10, 15, 20, and 25 m/s

Baseline Parameters: While varying one parameter, we keep other parameters fixed as below:

- Number of Nodes: 40
- Number of flows: 20
- Number of packets/sec: 200
- Speed : 10 m/s

5 Core Modification

In the ns2 source code, I made significant changes in the following files:

- tcp/tcp.h
- tcp/tcp.cc

5.1 New Variables

The following variables are newly declared in tcp.h:

- *int64_t* m_nSamples [Number of samples]
- *double* expertPrediction[100]
- *double* m_sum_prediction [used for predication calculation]
- *double* m_prediction [prediction for this round]
- *double* m_sum_weight [sum of weight m_rtts]
- *double* m_pool
- *double* m_range_min [min of expected rtt]
- *double* m_range_max [max of expected rtt]
- *double* m_weights[100]
- *double* m_loss[100]
- *double* m_expert_penalty
- *double* m_ETA [user defined learning rate]
- *double* m_share
- *int* num_Experts

5.2 New Methods

Methods defined in tcp.h:

- *virtual int64_t* CheckForReciprocalPowerOfTwo (double val) const;
- *virtual int64_t* GetNSamples (void) const;

5.3 Primary Change

The core change of the source code is in file tcp.cc.

- In the method ***void TcpAgent::rtt_update(double tao)*** I applied the methodology explained in the paper using the new defined variables and methods.
- Next I calculated a modified ***RTT Estimation*** defined as ***t_srtt_***.
- In the ***TcpAgent::TcpAgent()*** contructor, I initialized some of my new defined variables.
- Also in the ***TcpAgent::reset()*** method, I reset the variables to their initial values.
- Finally I defined a boolean varaible **USE FIXED SHARE**, which if set true, runs our model, otherwise it run the old methods.

6 SIMULATION RESULTS

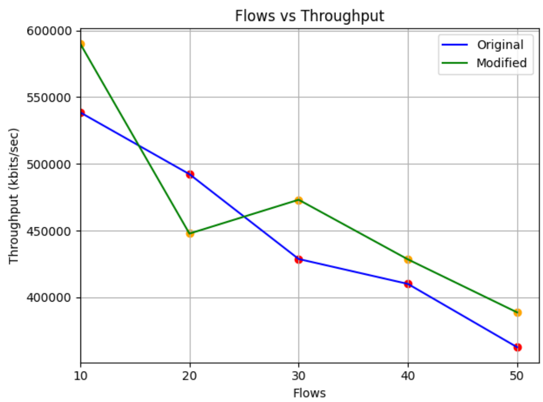
6.1 Topology 1: Wireless (mobile) 802.11

6.1.1 Description of the simulation

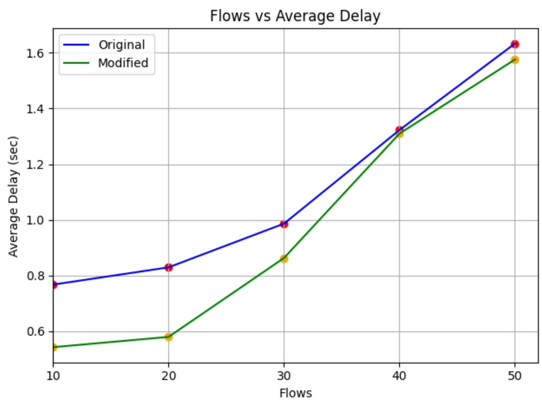
The simulation is based on the IEEE 802.11(mobile) standard. Routing is based on the AODV protocol.The simulation is based on the following parameters:

- Channel: Wireless channel
- Propagation: Two-ray ground
- Antenna: Omnidirectional
- Link: LL
- Queue: Drop-tail
- Routing: AODV
- Mobility: Random waypoint
- Position: Grid
- Area: 500 x 500 meters
- Flow: Random source to random destination
- Packet size: 20 bytes
- Number of nodes: Variable
- Number of flows: Variable
- Packet rate: Variable
- Speed: Variable
- Simulation time: 49 seconds

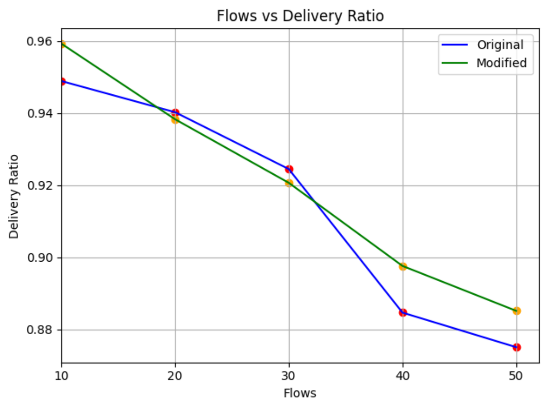
6.1.2 Generated Graphs



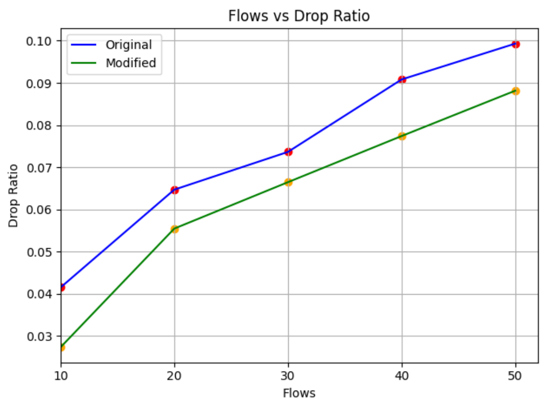
(a) Flows vs Throughput



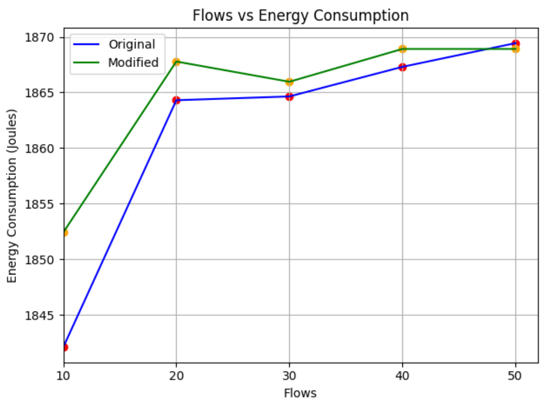
(b) Flows vs Average delay



(c) Flows vs Delivery Ratio

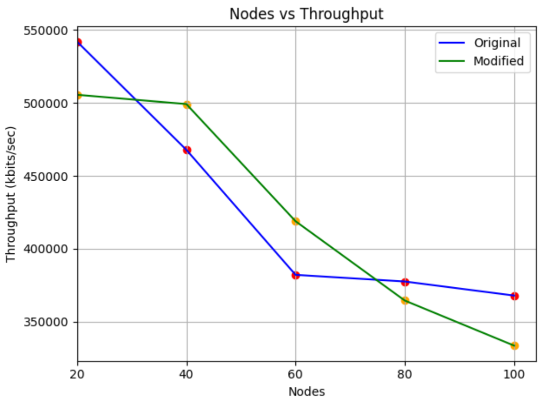


(d) Flows vs Drop Ratio

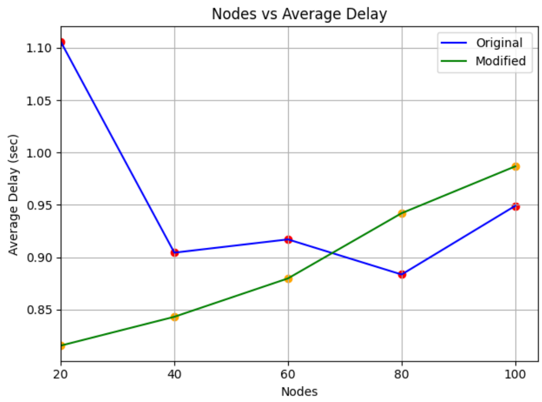


(e) Flows vs Energy Consumption

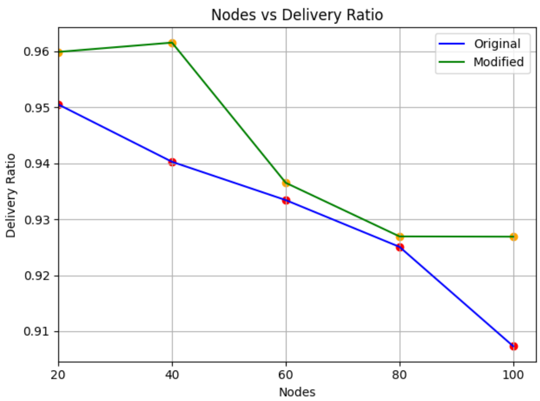
Figure 6.1: Varying Number of Flows



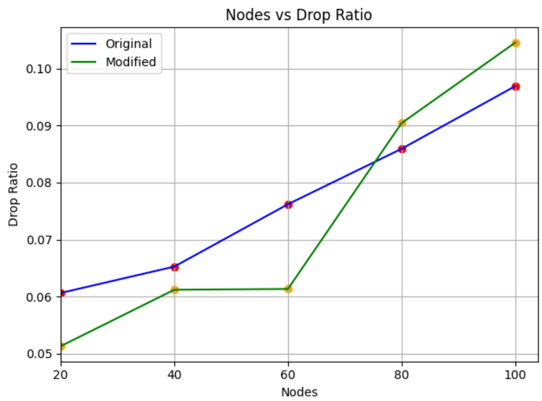
(a) Nodes vs Throughput



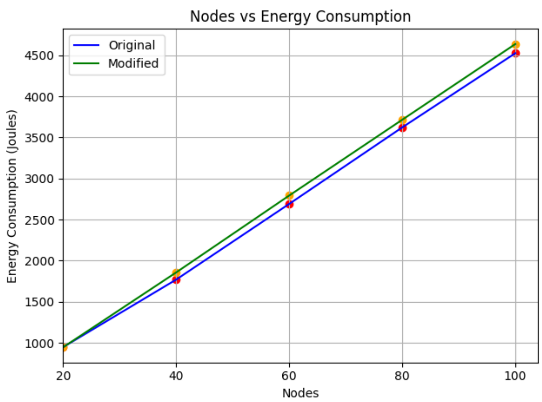
(b) Nodes vs Average delay



(c) Nodes vs Delivery Ratio

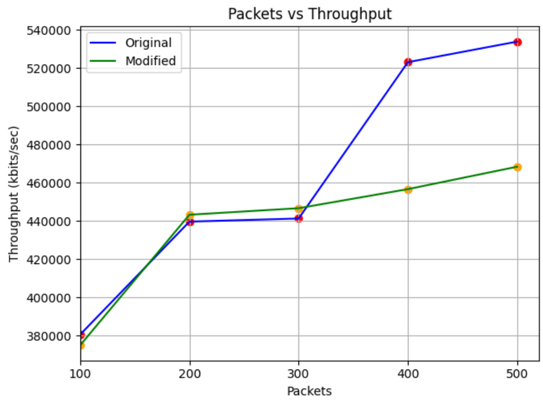


(d) Nodes vs Drop Ratio

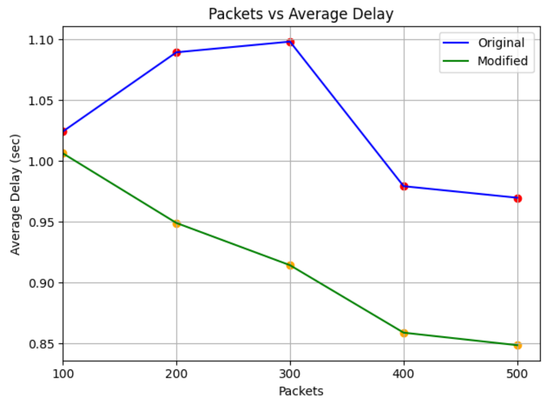


(e) Nodes vs Energy Consumption

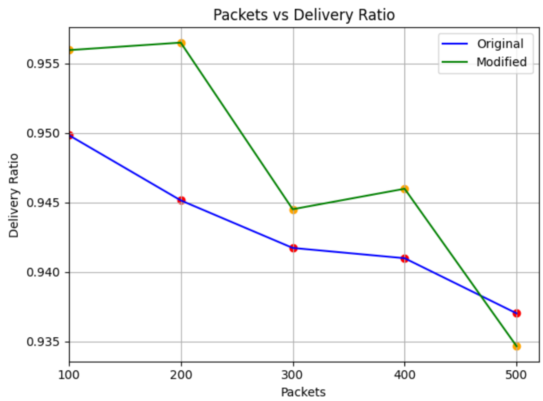
Figure 6.2: Varying Number of Nodes



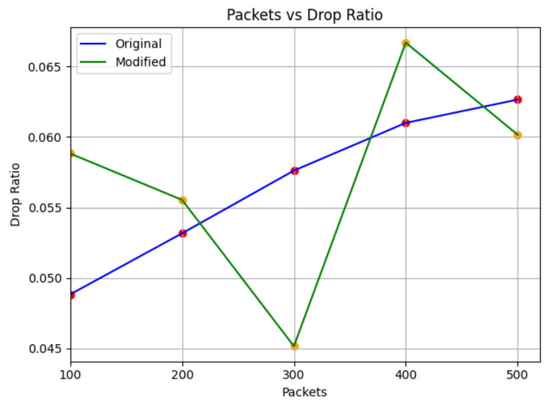
(a) Packets/sec vs Throughput



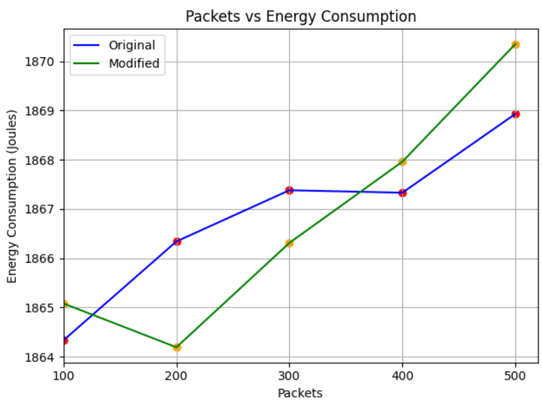
(b) Packets/sec vs Average delay



(c) Packets/sec vs Delivery Ratio

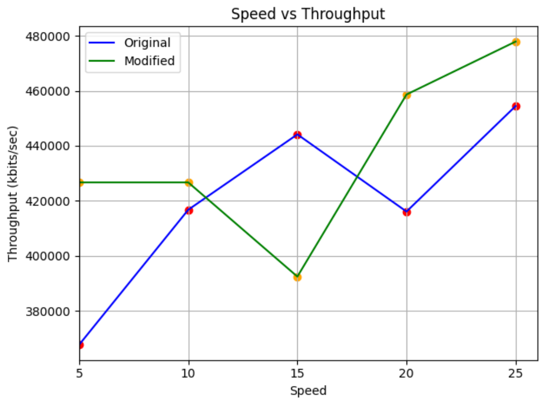


(d) Packets/sec vs Drop Ratio

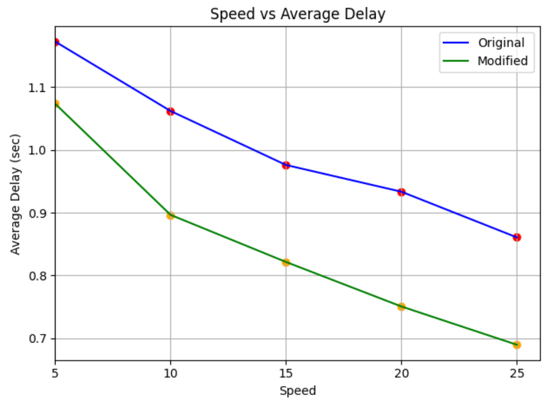


(e) Packets/sec vs Energy Consumption

Figure 6.3: Varying Number of Packets/sec

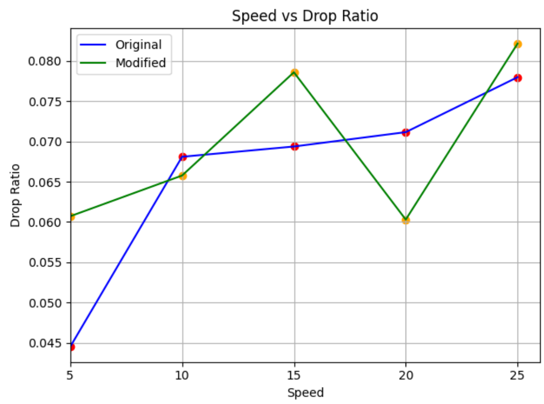


(a) Speed vs Throughput

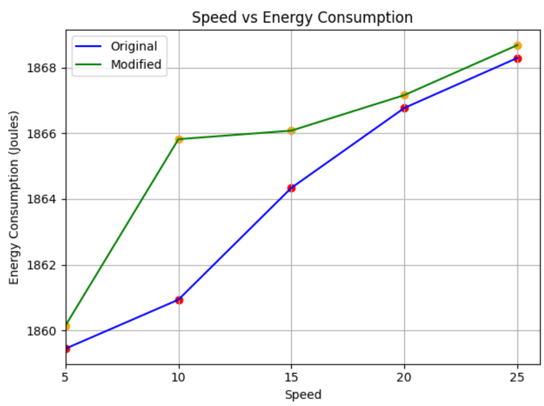


(b) Speed vs Average delay

(c) Speed vs Delivery Ratio



(d) Speed vs Drop Ratio



(e) Speed vs Energy Consumption

Figure 6.4: Varying Speed

6.2 Topology 2: Wireless (mobile) 802.15.4

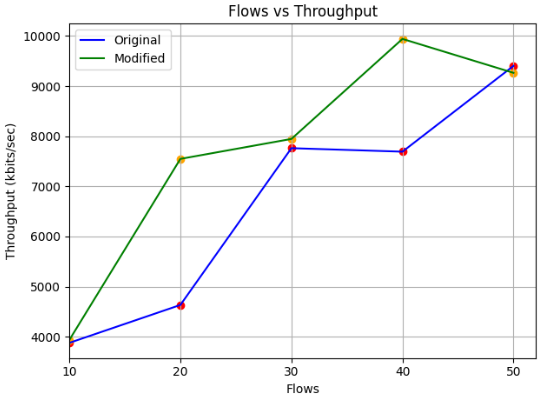
6.2.1 Description of the simulation

The simulation is based on the IEEE 802.15.4(mobile) standard. Routing is based on the AODV protocol.The simulation is based on the following parameters:

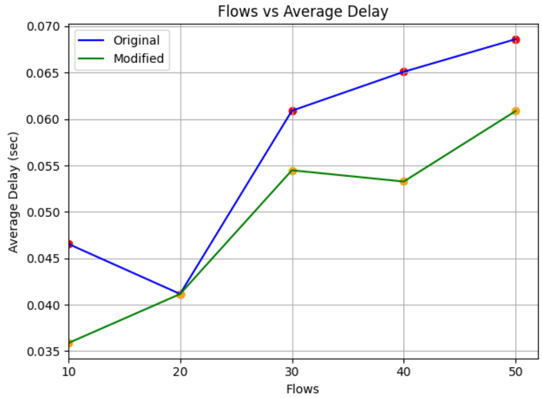
- Channel: Wireless channel
- Propagation: Two-ray ground

- Antenna: Omnidirectional
- Link: LL
- Queue: Drop-tail
- Routing: DSDV
- Mobility: Random waypoint
- Position: Grid
- Area: 500 x 500 meters
- Flow: Random source to random destination
- Packet size: 20 bytes
- Number of nodes: Variable
- Number of flows: Variable
- Packet rate: Variable
- Speed: Variable
- Simulation time: 49 seconds
- Energy Model : EnergyModel
- Initial Energy : 1000
- Idle Power : $712e^{-6}$
- RX Power : $35.28e^{-3}$
- TX Power : $31.32e^{-3}$
- Sleep Power : $144e^{-9}$

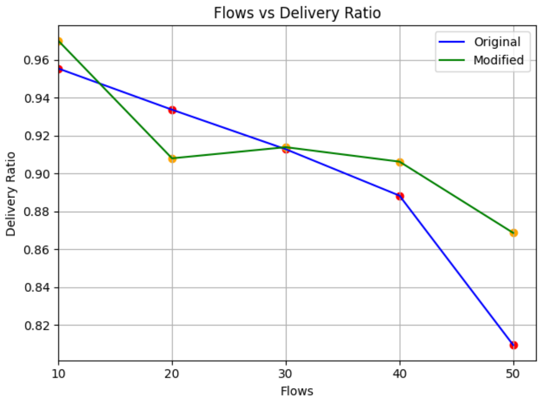
6.2.2 *Generated Graphs*



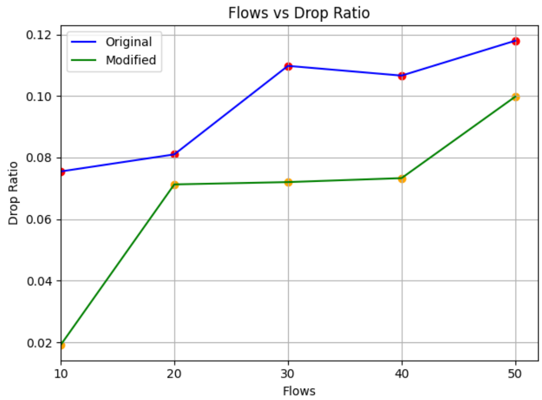
(a) Flows vs Throughput



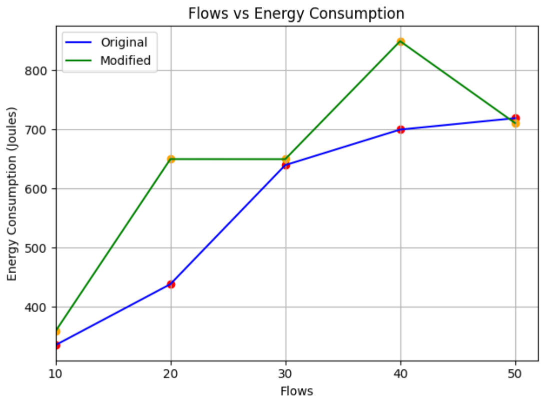
(b) Flows vs Average delay



(c) Flows vs Delivery Ratio

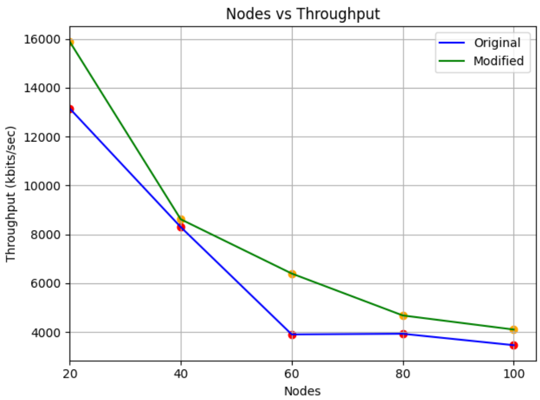


(d) Flows vs Drop Ratio

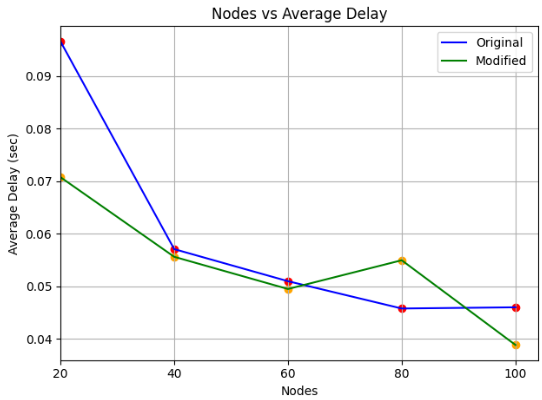


(e) Flows vs Energy Consumption

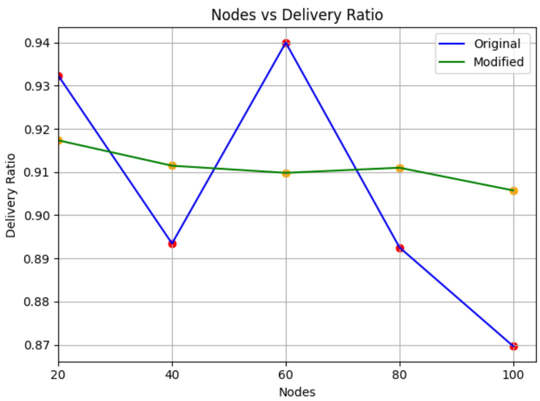
Figure 6.5: Varying Number of Flows



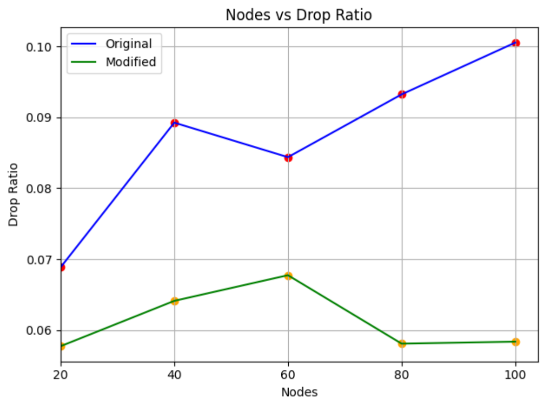
(a) Nodes vs Throughput



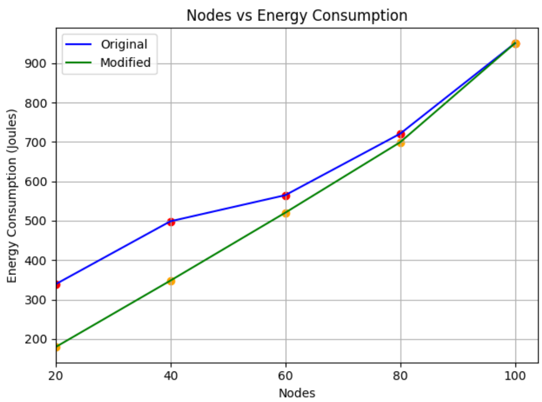
(b) Nodes vs Average delay



(c) Nodes vs Delivery Ratio

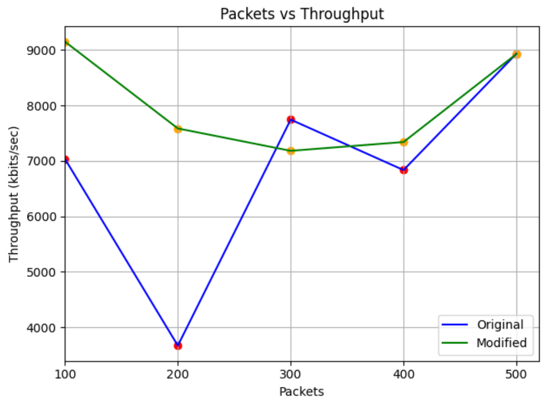


(d) Nodes vs Drop Ratio

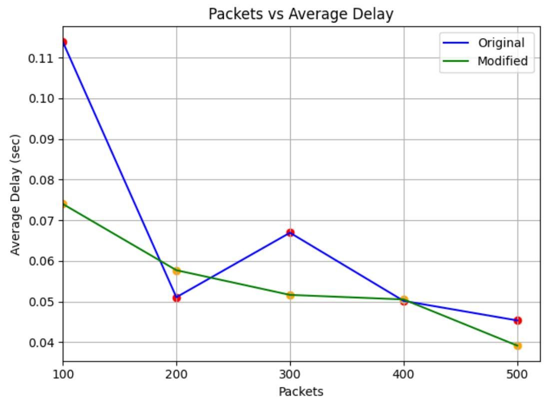


(e) Nodes vs Energy Consumption

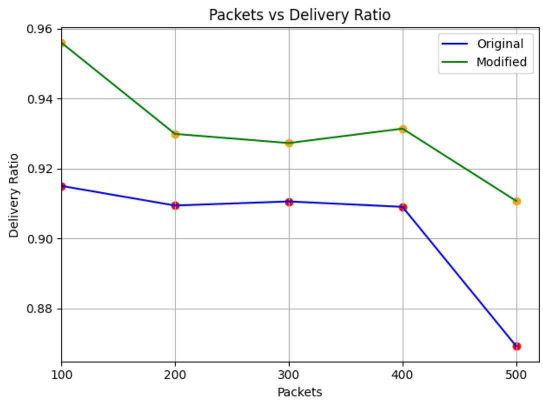
Figure 6.6: Varying Number of Nodes



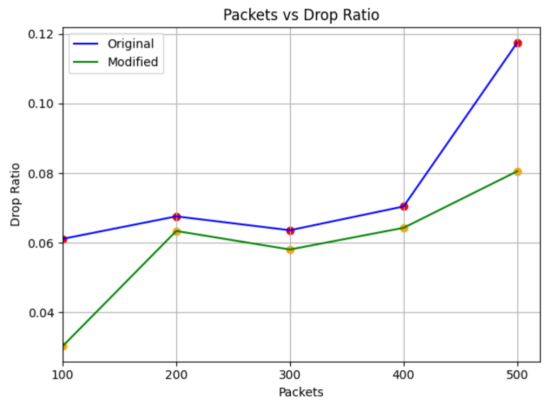
(a) Packets/sec vs Throughput



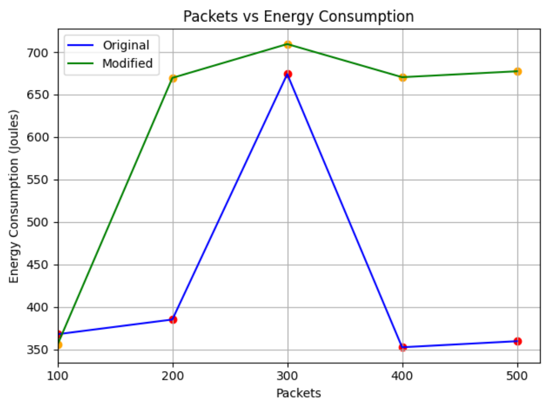
(b) Packets/sec vs Average delay



(c) Packets/sec vs Delivery Ratio

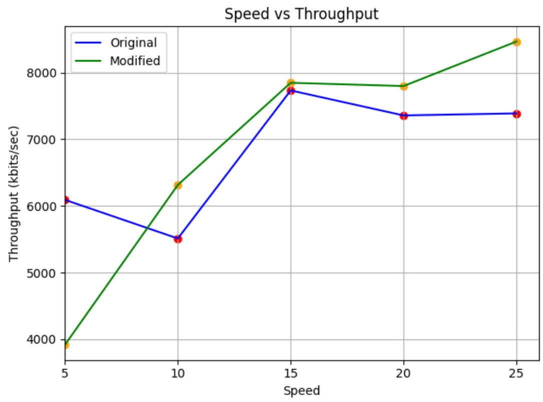


(d) Packets/sec vs Drop Ratio

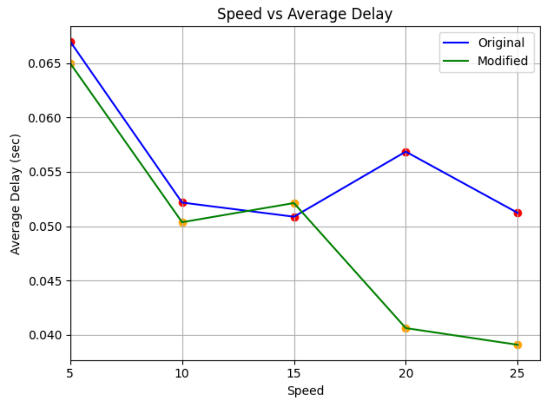


(e) Packets/sec vs Energy Consumption

Figure 6.7: Varying Number of Packets/sec

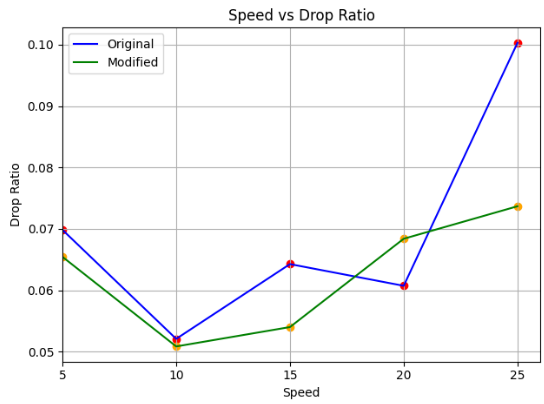


(a) Speed vs Throughput

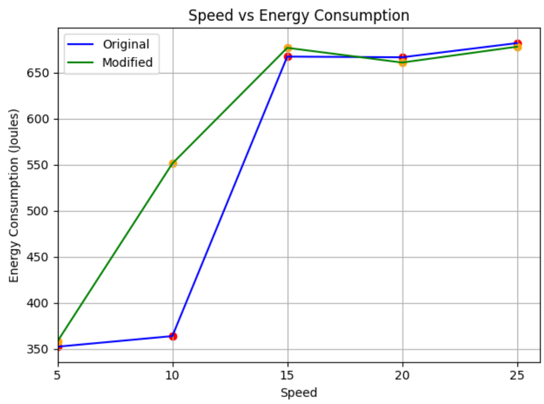


(b) Speed vs Average delay

(c) Speed vs Delivery Ratio



(d) Speed vs Drop Ratio



(e) Speed vs Energy Consumption

Figure 6.8: Varying Speed

7 Observation

As both 802.11 and 802.15.4 standards was simulated at mobile topologies where nodes move at random direction, the results are found to be quite similar to each other.

7.1 Varying Number of Flows

- As flows increase, the throughput of the network generally increases as more packets are transmitted. Our modified simulation results in higher throughput than the original
- The average delay generally increases as well with the number of flows. However, our proposed model is mainly focused on reducing the delay and our results match the argument.
- Delivery Ratio increases with the flow. The reason may be because of the congestion. Packets get dropped more frequently as a result. Drop ratio is similar and increases with number of flows
- Energy consumption increases with the number of flows, as more packets are transmitted and received, more energy is being consumed by the system.

7.2 Varying Number of Nodes

- As nodes increase, the throughput of the network generally decreases. This is because in the same network area if the number of nodes increases, so does the density of nodes in the network. This can potentially decrease the possibility of successful packet transmission.
- The average delay generally decreases as well with the number of flows. However, our proposed model still manages to hold a lower average delay than the original.Delivery Ratio decreases with the number of nodes for the same reason. And drop ratio increases.
- Energy consumption increases with the number of nodes. As more nodes are generated they consume more power.

7.3 Varying Number of Packets/sec Sent

- If more packets are sent per unit of time, more throughput is likely to be observed. And results show that as well. Our model does a great job having more throughput than the original.
- The average delay generally decreases with packets/sec. As the more packets are sent through the medium, successful transmission can be better guaranteed. Our model also follows this characteristics with less delay than the original.
- Delivery Ratio and drop ratio do not follow any general characteristics for this case. So the results found are random.
- Energy consumption increases with the number of packets sent per second. As more packets are transmitted and received, more energy is being consumed by the system.

7.4 Varying Speed

- In our simulation, we increase speed keeping other parameters constant. However, the nodes move at random destinations at each point of time. The average throughput generally increases with speed.
- The average delay generally decreases with speed as some transmissions are now much faster than before.
- However, delivery ratio and drop ratio do not follow any specific characteristics as nodes that goes far from the target do not ensure successful packet transmission. Also the nodes that get close to their targets ensure faster communication. So the graph is stochastic.
- Energy consumption increases with the speed, because kinetic energy also gets added to the system. So if speed increases, total energy consumption also increases.

8 References

References

[1] B. A. Nunes, K. Veenstra, W. Ballenthin, S. Lukin, and K. Obraczka, "A machine learning approach to end-to-end rtt estimation and its application to tcp," in *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2011, pp. 1–6.