Computer Networks Project Report

Determining the Optimal Values of Configuration Parameters in Wireless Networks for Effective Congestion Control using Machine Learning

Team Members:

Abdul Rasheed Mohamed Ali – 106120004

B.S. Siva Sundar – 106120026

J Sooraj Krishna - 106120048

Department: Computer Science and Engineering

Section: B

Faculty: Dr. B. Nithya

Course: Computer Networks

Course Code: CSPC53

Determining the Optimal Values of Configuration Parameters in Wireless Networks for Effective Congestion Control using Machine Learning

Abstract:

In this work, we will be determining the optimal value(s) for configuration parameters in a wireless network with the help of machine learning. In computer networks, the task of modelling and controlling network traffic to optimize network performance by decreasing packet congestion has been extensively studied, and many algorithms and solutions have been proposed. Congestion has a significant impact on Quality of Services (QoS) parameters such as packet delivery ratio (PDR), end-to-end delay and energy consumption in wireless networks. Previous research work on this area has focused on solutions such as - usage of MAC and network layer (cross-layer) schemes to categorize the existing congestion control techniques. increasing the size of storage buffers, usage of priority-based congestion avoidance schemes, comparisons between CoAP and CoCoA traffic routing algorithms in IoT wireless networks for congestion control, a QoS aware multipath routing protocol in wireless networks etc. Some issues that have been discovered from these proposed solutions include – heavy costs of infrastructure maintenance, restrictions in buffer storage sizes and many more. The solution we are proposing in this paper is to determine if congestion can be minimized by modifying the configuration parameters of the wireless network and find out the optimal values of these parameters. The evaluation will be done based on the values of different performance metrics attained such as throughput, end to end delay, number of packets dropped and packet delivery ratio. We will be creating a dataset of the corresponding values of configuration parameters and performance metrics and then by using 4 different machine learning algorithms – Linear Regression, Bayesian Regression, Decision Tree and Support Vector Machine (SVM), we will be finding out the optimal values of the configuration parameters. We will be simulating a wireless network on the network simulation software Network Simulator 2 (ns2). For the feasibility of creating a dataset, the configuration parameters that we will be modifying will be restricted to just 2 – Window Size and Packet Size.

Introduction:

Congestion refers to the state of a network when the network traffic is higher than what it can cope with. This results in a reduction of QoS (Quality of Service) parameters such as packet loss and latency and can cause other unwanted effects such as jitter and decrease in throughput.

Some reasons why network congestion can occur are -

i. When the bandwidth of a network is less, there is a high chance of congestion happening. An example of this is when a large amount of people are streaming the same sports match, buffer times get increased.
ii. Sometimes, some devices in a network can use more bandwidth than they are expected to be using, which can lead to congestion in the network lines.
iii. Sometimes the network configuration can be faulty which can introduce unwanted bugs which can indirectly lead to network congestion. These can be detected by network monitoring tools, which can identify the point of congestion and determine the point of misconfiguration.
iv. Outdated hardware can also cause bottlenecks to arise in the network which can cause congestion of critical traffic.
v. Concurrent TCP or any traffic flow facing port queues buffer tail-drop can degenerate into congestion activity; such congestion is termed TCP global synchronization.
vi. Burst of packets traffic/Flow over transmission time -- Intuitively defined by as “a group of consecutive packets with shorter inter-packet gaps than packets arriving before or after the burst of packets”; whereas to, it is a continuous transfer of data without interruption from one device to another. Overall, burstiness in packets flows is susceptible of causing congestion.

Presence of congestion in wide-scale networks may lead to congestive collapse. Congestive collapse is a condition where congestion prohibits the flow of critical information, which may lead to shutdown of important networks, like it happened in October 1986 when NSFNET’s phase-I network backbone dropped three orders of magnitude from its capacity of 32 kbit/s to 40 bit/s, which led to a massive loss of information. Detecting and mitigating congestion is hence, of utmost importance.

Congestion control mechanisms has been a hot research topic for over the past 4 decades due to its’ necessity. Many congestion control mechanisms have been invented, studied and even modified with to increase their effectiveness.

Previous research has yielded congestion avoidance/control protocols such as -

i. TCP/IP Congestion Avoidance Protocol – TCP networks enlist the use of a congestion window and a congestion policy. TCP Congestion Control Algorithm uses three phases – slow start, congestion avoidance and fast recovery to decide which operation to perform. Depending on the phase the network is currently in, the congestion policy dictates whether the congestion window should additively-increase or multiplicatively-decrease.

ii. Active Queue Management Techniques – These techniques involve the reordering or dropping of network packets based on the status of the network. congestion control algorithms are based on this technique like -
    a. Random Early Detection (RED)
    b. Robust Random Early Detection (RRED)
    c. Flow Based RED/RRED
    d. Explicit Congestion Notification

iii. TCP Window Shaping – A simple way of decreasing congestion is to reduce the amount of traffic that can flow into the network. This can be done by reducing the window advertisement which can restrict the size of data the remote servers can send.

iv. Priority-based Congestion Control Protocol (PCCP) is a congestion control approach for WSNs that uses a flexible and distributed rate adjustment in each sensor node. The strength of the protocol is that it considers that sensor nodes may have different priorities and would need different throughput. A sensor node with a higher priority index and those sensor nodes that inject more traffic get more bandwidth than other sensors in the network.

We have decided to focus specifically on congestion control in wireless area networks. Our research delves into the relationship between the configuration parameters of a wireless network and the subsequent performance metrics generated. We hope to determine whether a trend exists between the two and establish optimum values of these configuration parameters such that congestion is minimized, for which we look to use Machine Learning.

We will be implementing a wireless network in Network-Simulator 2 (ns2) software and adjusting configuration parameters to check for trends in performance metrics. Many configuration parameters exist for wireless networks such as – Congestion Window Size, Packet Size, Duplicate ACK Counter, Smoothed RTT, Variable RTT, Highest ACK etc but for the sake of feasibility, we will only be adjusting two parameters – Window Size and Packet Size as incorporation of more parameters will result in a dataset whose size, we will not be able to work with. We will then be creating a dataset of the corresponding values of configuration parameters and performance metrics and then by using 4 different machine learning algorithms – Linear Regression, Bayesian Regression, Decision Tree and Support Vector Machine (SVM), we will be finding out the optimal values of the configuration parameters.

Related Work:

a. Existing works:

i. An Optimized Service Differentiated Congestion Management protocol for delay constrained traffic in Healthcare WSN's [1]

This research paper has proposed a congestion control protocol for constrained delay traffic and optimized rate control for Healthcare Wireless Sensor Networks.There is distinction between high and low priority topics. The protocol has congestion avoidance and congestion control phases. For high priority, average end-to-end delay is minimised with node output scheduling weights. For low priority traffic, Active Queue Management (AQM) algorithm is used for congestion avoidance. It uses a distinct virtual queue to decide if received packets from child nodes should be dropped or not. If the packet is dropped, a three-state machine and a virtual queue is used to detect congestion. If congestion is present, the sending rate is modified using an optimization function. From the simulations run, the two-phase protocol is measured to be more effective than the FCFS protocol and the Fuzzy Congestion Control Protocol based on Active Queue Management (FCCPAQM) based on packets dropped, end to end delay and throughput as performance metrics.

ii. A Priority Based Congestion Avoidance Scheme for Healthcare Wireless Sensor Networks [2]

By dividing the data into two groups, critical data and non-critical data, this paper presents an algorithm for transmitting each group of data to medical centres.This scheme's objective is to avoid congestion and ensure secure data transmission. Congestion management protocol divides data into critical and non-critical data. Two fairness weighted scheduling queues, one for critical data and another for non-critical data is used. SVM model is used for transmitting critical data by finding shortest path. TOPSIS model based dynamic routing algorithm is used for transmission of non-critical data. To obtain an accurate classification of data, clustering can be used where each cluster has an SVM. The proposed algorithm has to be improved by using a simple linear function and creating a precise formula for queuing priority

iii. Congestion free opportunistic multipath routing load balancing scheme for Internet of Things (IoT) [3]

This paper reports that the demand for the Internet of Things (IoT) increases with the passage of the time, due to its induction in every aspect of life. Adil proposed an energy proficient routing scheme, which is known as Dynamic hop selection static routing protocol (DHSSRP). The proposed scheme manages the network traffic on priority-based information in a congested-free communication environment to balance the energy consumption of participating sensor devices and extend network lifetime.

They proposed a lightweight routing scheme to resolve the load balancing issue in IoT. The results observed for the proposed scheme during simulations in terms of load balancing, energy consumption, and communication metrics showed an average 8% improvement over the existing scheme. The limitation of the proposed model is its complex implementation in the initial phase or network deployment phase.

iv. **Congestion Free Routing Mechanism for IoT-Enabled Wireless Sensor Networks for Smart Healthcare Applications [4]**

This paper proposes a distributed congestion control algorithm for Internet of Things-enabled Wireless Sensor Network. The IoT is possible through the integration of several heterogeneous network infrastructures. The proposed scheme shows improved performance in an IoT-based healthcare system. Congestion control is a primary objective in several multihop WSNs. The proposed scheme considers two main parameters energy and delay. It considers two types of traffic: sensitive and non-sensitive. It is noted that the proposed scheme gives optimal results in terms of QoS in IoT-based healthcare applications. The limitation of the proposed model is that Routing Protocol for Low-Power and Lossy Networks has a very high impact on aggressive congestion control tactics such as CoCoA.

v. **Analysis of the interplay between RPL and the congestion control strategies for CoAP [5]**

The paper used WiSHFUL, a platform for extended network architecture experiments, to assess the efficacy of different CoAP load balancing solutions in a practical environment. In a real-world environment where path deviations and unreliable links are brought about by channel instability, the objective is to investigate alternative congestion control algorithms and how they interact with the routing algorithm. However, the RPL characteristics have a variable impact on various congestion control strategies, penalising the ones that are significantly more aggressive. The authors discovered that the straightforward CoAP pre-set traffic problems technique outperformed the more intricate CoCoA method in a real-world scenario with heavy traffic loads.

vi. **An IoT based Congestion Control Algorithm [6]**

This paper reports that Internet of Things (IoT) is the global network platform which offers interconnection between dissimilar devices having the ability to send data over the network. Different devices have their own requirements like communication speed, delay, and reliability. Application data is generally transported by the TCP, which provide congestion control and transmission rate adaptation policy. This paper proposed a new congestion control algorithm for IoT application protocols to quickly adapt the transmission rate according to network conditions. The proposed approach maintains fairness with TCP Cubic, which is widely implemented over the Internet.Simulation results show that the proposed approach

provides better results in terms of throughput and inter-protocol fairness with TCP Cubic. The proposed method is well suited for IoT applications protocol MQTT which work for lightweight smart home appliances, smart city monitoring systems, healthcare providers and sensors communicating to the server via satellite link, etc.

b. Summary of survey:

Algorithm & Year of publication	***Objective***	***Methodology***	***Input Metrics for making decision***	***Requirements /Assumptions***	***Achieved enhancements***	***Simulator/ Machine learning concept***	***Limitation, If any thing***
Weighted fair queuing scheduler and Active Queue Management, 2021	An Optimized Service Differentiated Congestion Management protocol for delay constrained traffic in Healthcare WSN's. This protocol's core objective is to evade, or if not likely, control congestion.	Service differentiation method is used to classify the packets based on their priorities. Node output scheduling weights for high priority traffic and AQM algorithm for low priority traffic.	AQM algorithm chooses to drop or accept packet based on probability of packet drop. Child priority (CHP) and Available Bandwidth (AB) is used as input parameters for the optimization function.	All the sensor nodes are stationary. QOS determines the priorities of the packets. Input traffic is classified as high priority or low priority. Constrained delay requirement of 0.25 seconds for high priority traffic.	For high priority class, delay is under the required constraint. For low priority class, traffic is decreased to reduce packet drop. Higher packet arrival rate, throughput and shorter queue length	Optimization function is executed in MATLAB and simulation is completed in OPNET. For simulation, MATLAB and OPNET run simultaneously.	High priority class traffics needs constrained delay bound, HP class queues scheduling weight periodically changes at the output WFQ scheduler at intermediate nodes to meet the bounded delay requirement.
Fairness weighted scheduling algorithm, Genetic algorithm, TOPSIS based dynamic routing, 2022	A Priority Based Congestion Avoidance Scheme for Healthcare Wireless Sensor Networks. This scheme's objective is to avoid congestion and ensure secure data transmission.	Congestion management protocol divides data into critical and non-critical data. Two fairness weighted scheduling queues, one for critical data and another for non-critical data is used.	Packet transfer time is determined by packet priority and elapsed time, Awareness Information, TOPSIS matrix	SVM model for transmitting critical data by finding shortest path. TOPSIS model based dynamic routing algorithm for transmission of non-critical data	Better performance in delay, energy consumption, packet loss and network performance.	The wireless sensor network was implemented using NS2 simulation software and the performance of SVM and TOPSIS was evaluated using MATLAB	To obtain a accurate classification of data, clustering can be used where each cluster has a SVM. Obtaining priority is time consuming and should be replaced by a linear function.

						version 2019b	
Dynamic hop selection static routing protocol (DHSSRP), 2021	Congestion free opportunistic multipath routing load balancing scheme for Internet of Things. This protocol's objective is to resolve the load balancing issue of IOT and prolong the networks lifetime.	Dynamic hop selection static routing protocol is implemented to prioritise the sensitive/critical information of sensors device with static routing and divert the neighbours' sensors communication with an alternate hop selection path.	Acknowledgement packet of sensor devices is used to prioritise their traffic in the network and the neighbours are assigned an alternate hop path.	The DHSSRP routing protocol requires the route request (RREQ) and route reply information (RREP) information of devices to prioritise traffic, routing table	Significant improvements in communication cost, computational cost, traffic congestion, throughput, PLR and network lifetime.	The dynamic hop selection static routing protocol was implemented in the OMNeT++ simulation tool	Complex implementation in the initial phase or network deployment phase. The model has to be implemented very accurately to achieve desired results.
Distributed traffic-aware congestion control schemes, 2020	Cross layer design that spans across different layers to provide congestion-free routing in the network layer and effective access control and transmission power control in MAC sub-layer	The proposed scheme used a priority-based data routing strategy to control congestion within the network. It classifies data packets into three different categories according to the priority of the data packets.	Proposed scheme considers two main parameters and delay. The two types of traffic considered are sensitive and non-sensitive. Priority is based on the priority of the patient and also the request time.	N number of static nodes arbitrarily located. Sensors attached to the patient's body should set high or low priority to an information. If the vital signs are out of normal range, it should be set as high priority.	Increase in percentage of successfully received packets, Reduction in average hop-by-hop delay, improved lifetime over traffic load	Extensive experiments on NS-2. The sensor nodes are arranged in random 2D topology. Collision free MAC protocol is used for the simulations.	Routing Protocol for Low-Power and Lossy Networks has a very high impact on aggressive congestion control tactics.
CocoA and CoCoA+ Algorithm, 2020	Analysis of the interplay between RPL and the congestion control strategies for CoAP where unstable links and route fluctuations are present	The CoCoA algorithm is composed of a policy to calculate the retransmission timeout (RTO), a back-off policy to set the RTO for retransmissions and ageing policy for the	CoCoA updates RTO valued based on measured Round Trip Times (RTTs) on messages correctly delivered with and without retransmissions. CoCoA	For each destination, the node has to maintain RTT_{strong},RTT_{weak},$RTTVAR_{strong,}$ $RTO_{weak,}$ RTO_{strong},$RTO_{overall}$. Variable backoff factor depends on the initial RTO value.	CoCoA ensured a better Carried load, Number of CoAP transmissions and dropped packets.	The simulations were run on WiSHFUL which is a platform for experimentation on network beds. Experiments were carried out on the Pisa	At high traffic loads, the CoAP default congestion control algorithm performs better than complex CoCoA algorithm. Additional network

	due to channel variability.	status information.	computes the new RTO value for retransmissions according to a variable backoff factor (VBF).			IoT Testbed (PINT) and Wilab testbed (WiLab)	topologies and traffic patterns have to be considered for conclusive results.+
Reactive and proactive TCP algorithms, 2020	An IoT based Congestion Control Algorithm. Objective of the algorithm is to adapt the transmission rate quickly whenever available bandwidth and delay changes.	Congestion control approach adjusts data transfer rate according to the requirement of IoT devices. Initial congestion window is decided according available bandwidth of the path.	Available bandwidth of the path is required to decide the initial congestion window size. The congestion detection factor β is dependent on RTT_{min}, RTT_{max} and RTT.	The modified TCP algorithm requires the Congestion Window, Slow-start threshold initialization, Modified Slow-start and Congestion avoidance, Congestion Control methods to adjust the transmission rate of the path.	The proposed TCP algorithm achieves a better throughput, average throughput and propagation delay.	All the simulation is conducted in NS-2 simulator. A dumbbell topology is used with New Reno, HTCP, CUBIC, Fast-Fit with CBR, FTP and VBR traffic generators.	The proposed method is only suitable for lightweight MQTT protocol based IoT devices.

c. Inference from the table

From the table we find some specific limitations with the various proposed algorithms such as Weighted fair queue scheduling algorithm, Genetic algorithm, TOPSIS based dynamic scheduling, Dynamic hop selection static routing protocol, Distributed traffic-aware congestion control, CoCoA and CoCoA+ Algorithms etc. Each of the algorithm has its own achieved enhancements while also trading off other factors. Healthcare applications are delicate by nature, and sometimes a provisioning delay will result in a patient's death. Therefore, prompt diagnosis and treatment are only achievable in a communication environment free from congestion.

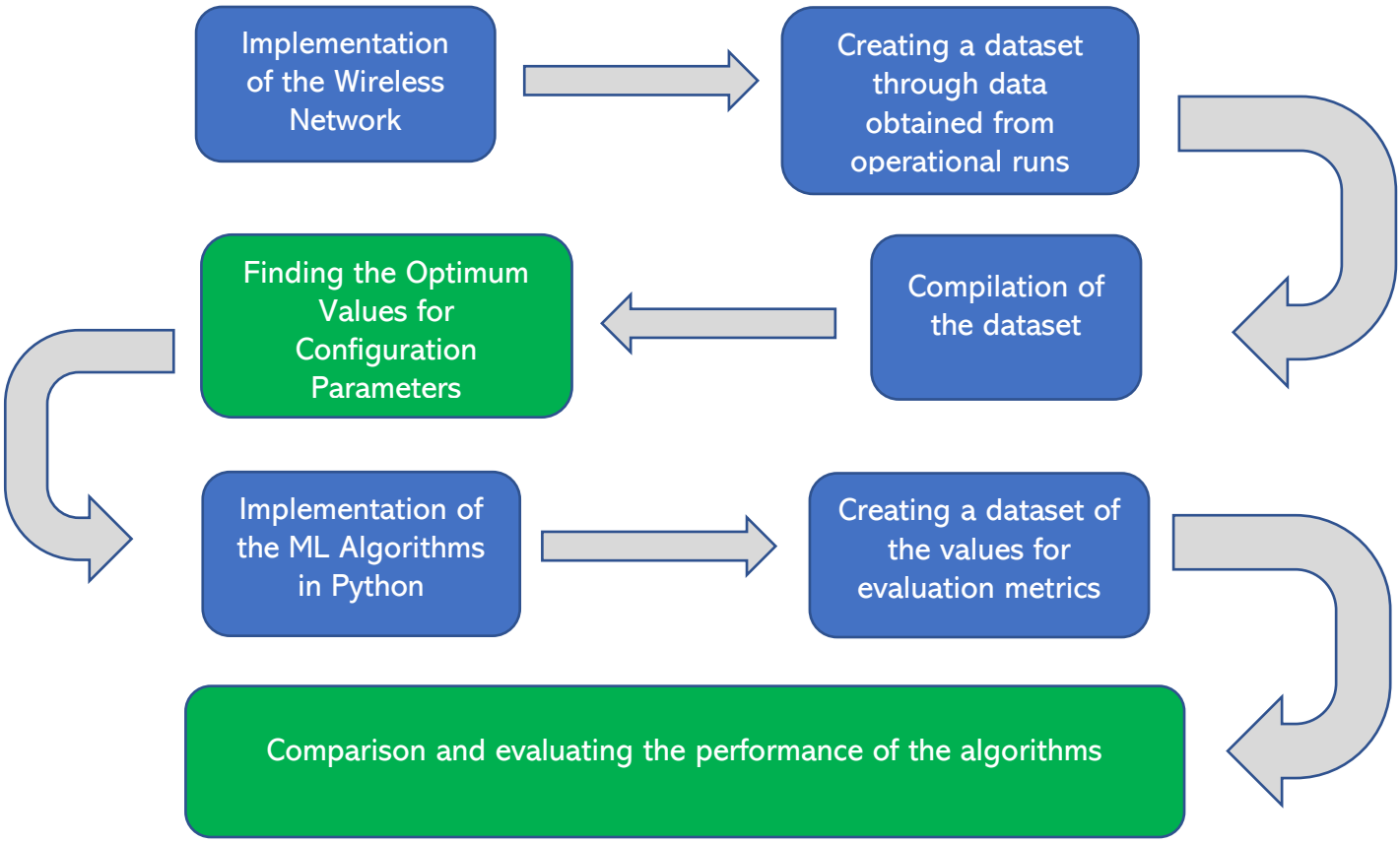
d. Motivation and contribution of our work

A congestion free routing mechanism is of the utmost importance for Healthcare Internet of Things sensor devices. Doctors, nurses, and patients may have communication breakdowns due to congestion, which could lead to a delayed or incorrect diagnosis. Therefore, a method for this congestion-free routing mechanism must exist. Additionally, the current congestion control system is unfit for use in the healthcare industry. The healthcare industry needs a mechanism that is more exact and efficient. The modification we propose aim to achieve a network for which congestion was found to be minimal through the evaluation of Packet Delivery Ratio, Number of Dropped Packets, Average End-to-End Delay and Average Throughput.

Proposed Methodology

All the codes used can be found in –
https://github.com/AbdulRasheed02/Computer-Networks-Project

The detailed flowchart of the step-by-step procedure for our research paper can be found below –



Our first step was the implementation of the wireless network. Our platform of choice for this step was Network-Simulator 2. We implemented a 7-node star topology wireless network using the programming language TCL.

We used a flat grid topography and general operations director for recording node movement and position. We used a Wireless Channel, Two-Ray radio propagation model and a 2.4Ghz WAVELAN DSSS network interface type. Each node had a priority queue associated with it of length 500 packets. The network routing was handled by the Ad hoc On-Demand Distance Vector Routing (AODV) protocol. The flatgrid was set to 200 metres x 100 metres. There were 6 source nodes and 1 sink nodes arranged in a star topology to replicate a wireless body area network. TCP Connection with FTP traffic was setup between every source and sink. The simulation was run for 50 seconds.

Once the wireless network was implemented in TCL, we ran the Wireless Network for 50 iterations with different values of Window Size and Packet Size in each iteration. The domain for Window Size was - (20, 30, 40, 50, 60) while the domain for Packet Size was (1000, 1010, 1020, 1030, 1040, 1050, 1060, 1070, 1080, 1090). The configuration parameters we decided to focus on for the best results were – Packet Delivery Ratio (PDR), Number of Dropped Packets, Average Throughput and Average End-to-End Delay. On doing so, we end up with fifty trace files from which we compile the dataset by calculating the values of the four-performance metrics with the help of awk files. The compiled raw dataset was saved in .xlsx format,

Our next step was to find the optimum values of Window Size and Packet Size for the least congestion in the wireless network. For this task, we decided to divide the 4 metrics and attach ranks for the corresponding combinations of window size and packet size. On doing so, we will end up with 4 values of ranks for each window size + packet size combination, 1 for each metric. The cumulative sum of these 4 ranks will give us a metric that will help us in finding out the combination of window size and packet size for which the congestion is minimum.

Our next step was to convert the .xlsx file to .csv format for further processing in Python.

Our next step was to select 4 Machine Learning algorithms which we would then use to compare and check which algorithm performed the best. We decided to use – Linear Regression, Bayesian Regression, Decision Tree and Support Vector Machines for this purpose. The performance metrics used to compare the 4 algorithms were – Mean Absolute Error, Mean Square Error, Root Mean Square Error, Maximum Error.

We implemented the 4 algorithms in Python and then ran the programs, which used the data from the .csv file created earlier.

On doing so, we obtained the results and performed a comparison analysis of them, which is further discussed in the Simulation and Analysis section.

Window Size	Packet Size	Packet Deliv	Dropped Pac	Avg Througp	Avg End to End Delay
20	1000	0.963728	251	1209.56	22.9817
20	1010	0.974506	168	1162.56	28.0714
20	1020	0.971706	218	1126.99	30.3915
20	1030	0.958193	272	1053.91	32.3363
20	1040	0.95273	17	131.501	59.496
20	1050	0.972488	170	46.355	968.618
20	1060	0.961158	245	59.3652	946.528
20	1070	0.963087	251	39.6188	959.218
20	1080	0.981772	93	929.51	72.7505
20	1090	0.9809	84	63.1461	1047.13
30	1000	0.978753	93	1210.89	36.7616
30	1010	0.968878	195	1091.83	30.3159
30	1020	0.972408	203	1111.73	36.1649
30	1030	0.949435	285	797.963	59.6258
30	1040	0.911929	19	80.0064	86.0233
30	1050	0.967373	237	1038.9	65.1736
30	1060	0.969405	175	1000.03	75.7588
30	1070	0.971868	181	82.5897	955.488
30	1080	0.982546	89	939.187	115.276
30	1090	0.975528	94	76.6061	1036.86
40	1000	0.974135	132	1217.22	33.627
40	1010	0.972736	155	38.9769	1180.37
40	1020	0.955361	49	58.9407	218.106
40	1030	0.828265	61	51.8736	81.934
40	1040	0.941332	52	134.957	129.252
40	1050	0.839604	12	47.088	121.766
40	1060	0.925883	109	241.296	68.2051
40	1070	0.976557	125	966.499	92.3146
40	1080	0.969621	200	953.165	110.703
40	1090	0.981422	101	93.1572	1079.2
50	1000	0.970126	169	1213.06	32.4715
50	1010	0.966819	187	1178.35	35.7462
50	1020	0.818985	26	40.1952	30.1965
50	1030	0.964242	221	1079.07	65.332
50	1040	0.968495	214	1054.25	82.0853
50	1050	0.947918	396	1014.83	51.3609
50	1060	0.95329	334	900.416	63.5357
50	1070	0.965585	245	975.557	93.9863
50	1080	0.984182	104	933.274	166.392

Raw dataset

```
Window Size,Packet Size,Packet Delivery Ra

20,1000,0.963728,251,1209.56,22.9817
20,1010,0.974506,168,1162.56,28.0714
20,1020,0.971706,218,1126.99,30.3915
20,1030,0.958193,272,1053.91,32.3363
20,1040,0.95273,17,131.501,59.496
20,1050,0.972488,170,46.355,968.618
20,1060,0.961158,245,59.3652,946.528
20,1070,0.963087,251,39.6188,959.218
20,1080,0.981772,93,929.51,72.7505
20,1090,0.9809,84,63.1461,1047.13
30,1000,0.978753,93,1210.89,36.7616
30,1010,0.968878,195,1091.83,30.3159
30,1020,0.972408,203,1111.73,36.1649
30,1030,0.949435,285,797.963,59.6258
30,1040,0.911929,19,80.0064,86.0233
30,1050,0.967373,237,1038.9,65.1736
30,1060,0.969405,175,1000.03,75.7588
30,1070,0.971868,181,82.5897,955.488
30,1080,0.982546,89,939.187,115.276
30,1090,0.975528,94,76.6061,1036.86
40,1000,0.974135,132,1217.22,33.627
40,1010,0.972736,155,38.9769,1180.37
40,1020,0.955361,49,58.9407,218.106
40,1030,0.828265,61,51.8736,81.934
40,1040,0.941332,52,134.957,129.252
40,1050,0.839604,12,47.088,121.766
40,1060,0.925883,109,241.296,68.2051
40,1070,0.976557,125,966.499,92.3146
40,1080,0.969621,200,953.165,110.703
40,1090,0.981422,101,93.1572,1079.2
50,1000,0.970126,169,1213.06,32.4715
50,1010,0.966819,187,1178.35,35.7462
50,1020,0.818985,26,40.1952,30.1965
50,1030,0.964242,221,1079.07,65.332
```

Dataset in CSV format

Window Size	Packet Size	Packet Delivery Ratio	Rank	Dropped Packets	Rank	Avg Througput	Rank	Avg End to End Delay	Rank	Total Rank
20	1000	0.963728	28	251	40	1209.56	5	22.9817	1	74
20	1010	0.974506	10	168	22	1162.56	8	28.0714	2	42
20	1020	0.971706	17	218	34	1126.99	9	30.3915	7	67
20	1030	0.958193	35	272	42	1053.91	18	32.3363	8	103
20	1040	0.95273	39	17	2	131.501	36	59.496	16	93
20	1050	0.972488	13	170	24	46.355	46	968.618	46	129
20	1060	0.961158	31	245	38	59.3652	42	946.528	43	154
20	1070	0.963087	30	251	41	39.6188	49	959.218	45	165
20	1080	0.981772	3	93	11	929.51	31	72.7505	23	68
20	1090	0.9809	5	84	9	63.1461	41	1047.13	48	103
30	1000	0.978753	7	93	12	1210.89	4	36.7616	14	37
30	1010	0.968878	21	195	30	1091.83	12	30.3159	6	69
30	1020	0.972408	14	203	32	1111.73	10	36.1649	13	69
30	1030	0.949435	40	285	43	797.963	33	59.6258	17	133
30	1040	0.911929	46	19	3	80.0064	39	86.0233	29	117
30	1050	0.967373	24	237	36	1038.9	19	65.1736	19	98
30	1060	0.969405	20	175	25	1000.03	21	75.7588	25	91
30	1070	0.971868	16	181	26	82.5897	38	955.488	44	124
30	1080	0.982546	2	89	10	939.187	29	115.276	35	76
30	1090	0.975528	9	94	13	76.6061	40	1036.86	47	109
40	1000	0.974135	11	132	18	1217.22	1	33.627	10	40
40	1010	0.972736	12	155	21	38.9769	50	1180.37	50	133
40	1020	0.955361	37	49	6	58.9407	43	218.106	42	128
40	1030	0.828265	48	61	8	51.8736	44	81.934	27	127
40	1040	0.941332	44	52	7	134.957	35	129.252	37	123
40	1050	0.839604	47	12	1	47.088	45	121.766	36	129
40	1060	0.925883	45	109	16	241.296	34	68.2051	22	117
40	1070	0.976557	8	125	17	966.499	24	92.3146	31	80
40	1080	0.969621	19	200	31	953.165	27	110.703	34	111
40	1090	0.981422	4	101	14	93.1572	37	1079.2	49	104
50	1000	0.970126	18	169	23	1213.06	3	32.4715	9	53
50	1010	0.966819	25	187	28	1178.35	7	35.7462	12	72
50	1020	0.818985	49	26	4	40.1952	47	30.1965	4	104
50	1030	0.964242	27	221	35	1079.07	14	65.332	20	96
50	1040	0.968495	23	214	33	1054.25	17	82.0853	28	101
50	1050	0.947918	42	396	50	1014.83	20	51.3609	15	127
50	1060	0.95329	38	334	47	900.416	32	63.5357	18	135

Optimal Window Size + Packet Size

Window Size	Packet Size	Total Rank
30	1000	37
40	1000	40
20	1010	42
50	1000	53
60	1010	58
20	1020	67
20	1080	68
30	1010	69
30	1020	69
50	1010	72
20	1000	74
30	1080	76
60	1030	79
40	1070	80
60	1000	82
60	1040	83
50	1080	85
30	1060	91
20	1040	93
50	1030	96
30	1050	98
50	1040	101
20	1030	103
20	1090	103
40	1090	104
50	1020	104
50	1090	105
60	1020	108
30	1090	109
40	1080	111
60	1090	111
30	1040	117
40	1060	117
50	1070	119
40	1040	123
30	1070	124
40	1030	127

Assigning rank to calculate the optimum TCP window size and Packet size combination

Simulation and Analysis

a. Simulation Environment

The wireless star topology simulations were done in NS2 to generate the required trace files. The machine learning algorithms used on the dataset were run in PyCharm IDE.

b. Simulation metrics

The performance metrics used for creating the dataset by varying TCP window size and Packet size are:

i. Packet Delivery Ratio - the ratio of data packets that are actually received at the receiver end to those which were originally sent by sender
ii. Dropped Packets – The number of packets of data not reaching their destination after being transmitted across a network
iii. Average throughput - total payload over the entire session divided by the total time
iv. Average end-to-end delay - the time taken for a packet to be transmitted across a network from source to destination

The performance metrics used to compare the prediction accuracy of the four machine learning algorithms run to find the optimal TCP window size and Packet size are:

i. Mean absolute error - Absolute error refers to the magnitude of difference between the prediction of an observation and the true value of that observation. Mean absolute error takes the average of absolute errors for a group of predictions and observations.
ii. Mean square error - The Mean Squared Error measures how close a regression line is to a set of data points. Mean square error is calculated by taking the average of the errors squared from data as it relates to a function.
iii. Root mean square error – It is the square root of mean square error
iv. Maximum error - It is the absolute value of the most significant difference between a predicted variable and its real value

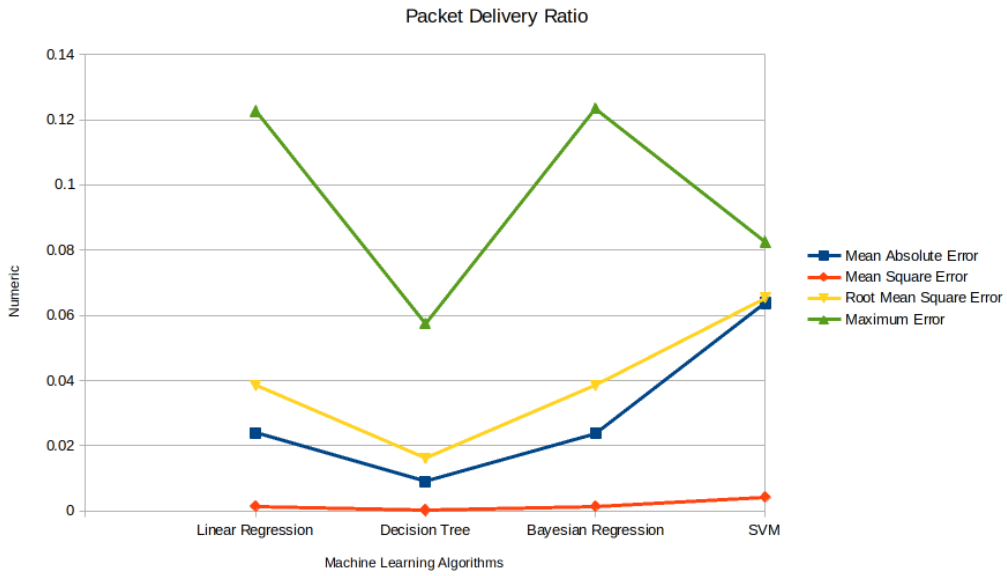
c. Performance Analysis

The performance metrics obtained for fifty combinations of TCP window size and packet size were tabulated to form the dataset for the machine learning algorithms. The dataset obtained was converted to CSV format to be compatible with the machine learning algorithms

i. Inferences from the results obtained by running the machine learning algorithms:

1. Packet Delivery Ratio:

Packet Delivery Ratio as dependent variable				
Machine Learning Algorithms	Mean Absolute Error	Mean Square Error	Root Mean Square Error	Maximum Error
Linear Regression	0.0239496043636363	0.00148283452899435	0.0385075905373778	0.12270512060606
Decision Tree	0.00894078	0.000259707697477499	0.0161154490312091	0.0575609999999998
Bayesian Regression	0.0238129655570338	0.00148364864763834	0.0385181599721275	0.123502328247383
SVM	0.06376286	0.00427122067809	0.0653545765657616	0.0825985



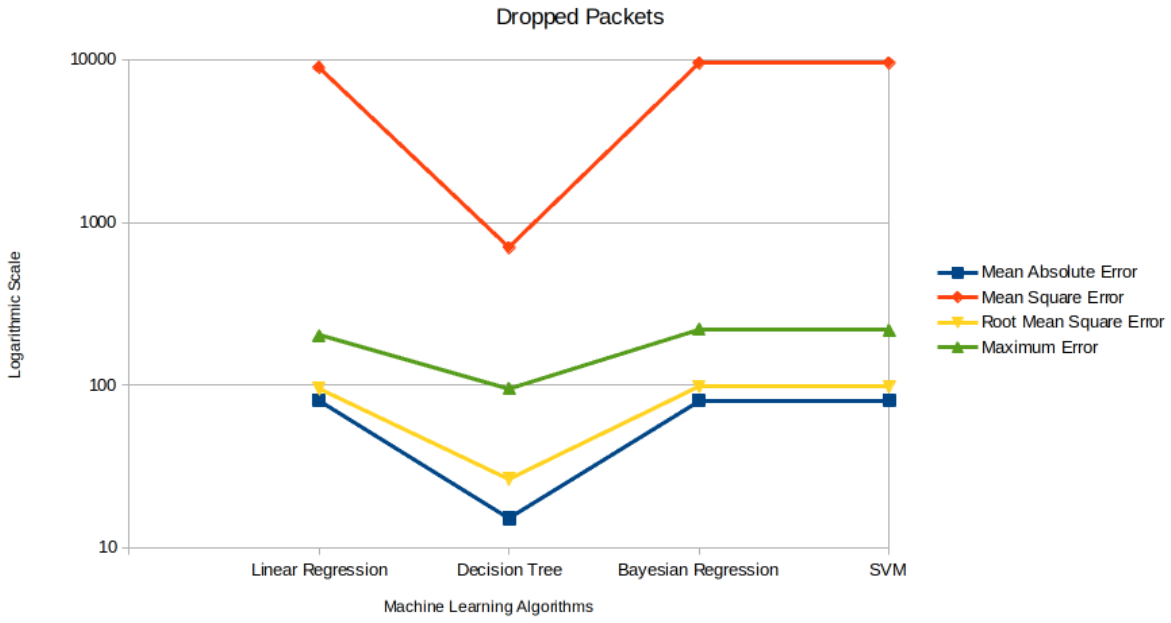
Preferred ML algorithm in terms of prediction accuracy for PDR considering

a. Mean Absolute Error: Decision Tree
b. Mean Square Error: Decision Tree
c. Root Mean Square Error: Decision Tree
d. Maximum Error: Decision Tree

Overall: Decision Tree

2. Dropped Packets:

Packets Dropped as dependent variable				
Machine Learning Algorithms	Mean Absolute Error	Mean Square Error	Root Mean Square Error	Maximum Error
Linear Regression	80.4692	8980.0956060606	94.7633663715077	202.665757575757
Decision Tree	15.14	697.736666666666	26.4147055002827	95
Bayesian Regression	79.9916607607852	9566.75820213976	97.8098062677754	220.288808232265
SVM	79.9982147316097	9573.81645522341	97.8458811357095	217.995826942798



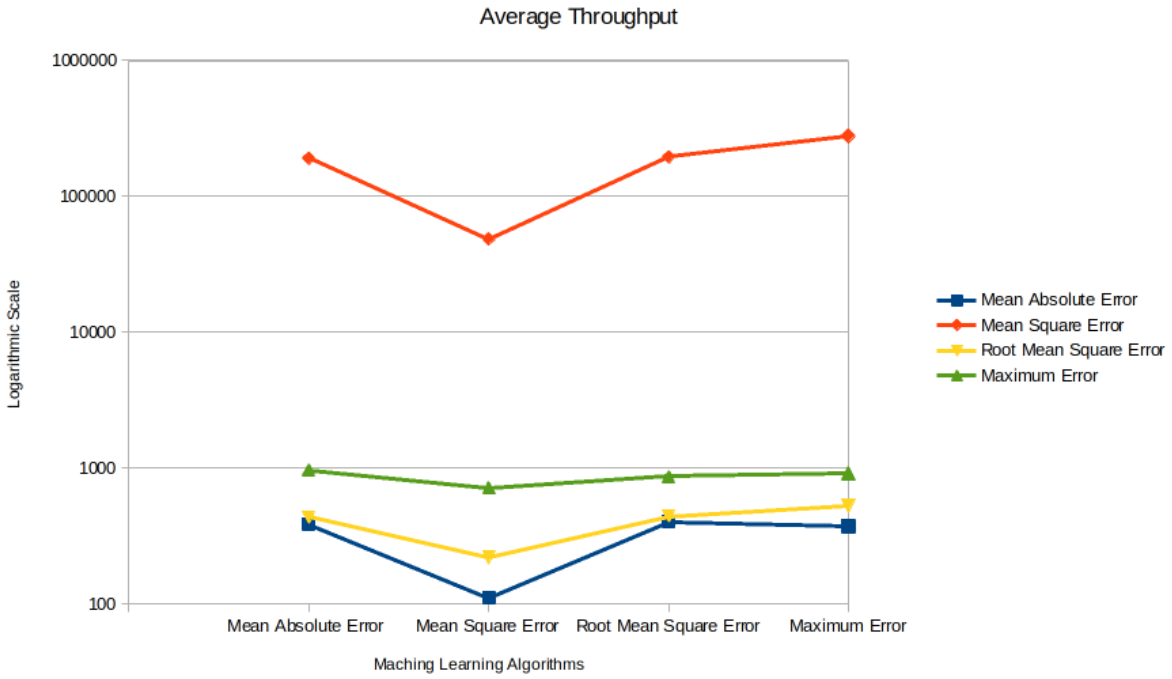
Preferred ML algorithm in terms of prediction accuracy for Dropped Packets considering

e. Mean Absolute Error: Decision Tree
f. Mean Square Error: Decision Tree
g. Root Mean Square Error: Decision Tree
h. Maximum Error: Decision Tree

Overall: Decision Tree

3. Average Throughput:

Average Throughput as dependent variable				
Machine Learning Algorithms	Mean Absolute Error	Mean Square Error	Root Mean Square Error	Maximum Error
Mean Absolute Error	386.594568387878	191770.099111107	437.915630128804	966.038199030303
Mean Square Error	110.855203	48685.1893301848	220.647205579823	716.549491666666
Root Mean Square Error	403.94761746751	195017.565344382	441.607931704563	874.899060953305
Maximum Error	373.909056812187	278117.795468896	527.368747148422	919.919483837704



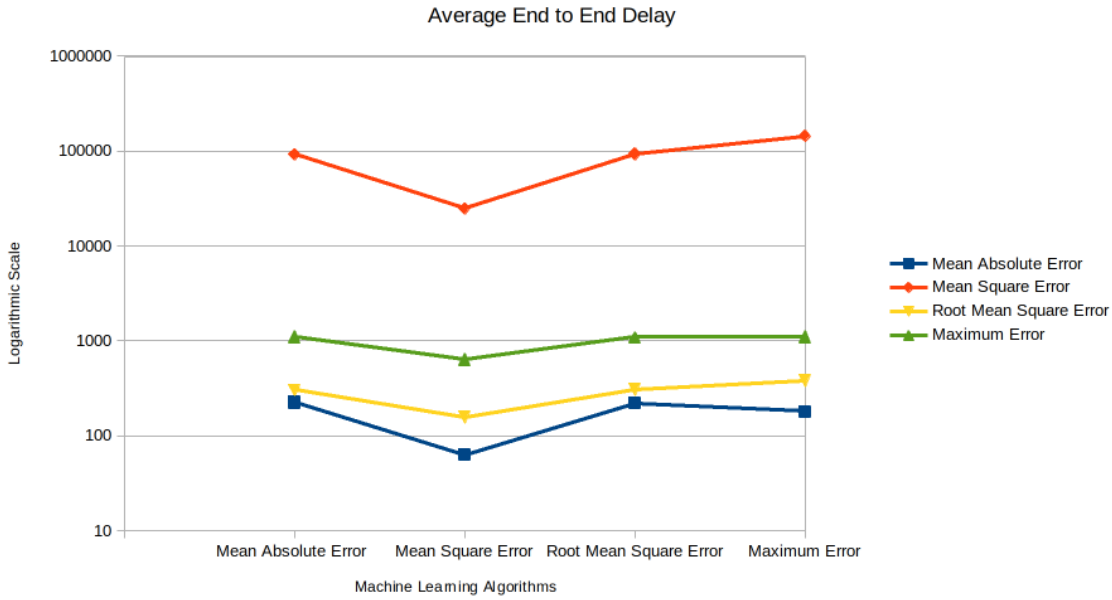
Preferred ML algorithm in terms of prediction accuracy for Average Throughput considering

a. Mean Absolute Error: Decision Tree
b. Mean Square Error: Decision Tree
c. Root Mean Square Error: Decision Tree
d. Maximum Error: Decision Tree

Overall: Decision Tree

4. Average End-to-End Delay:

Average End to End Delay as dependent variable				
Machine Learning Algorithms	Mean Absolute Error	Mean Square Error	Root Mean Square Error	Maximum Error
Mean Absolute Error	226.35572338909	92646.8299524196	304.379417754255	1108.31918545454
Mean Square Error	63.2718992	24772.652843663	157.393306222542	637.0426
Root Mean Square Error	222.589442023068	93370.0469484796	305.565127179918	1098.42484687859
Maximum Error	181.216564793341	145360.767006318	381.262071292593	1103.01473485819



Preferred ML algorithm in terms of prediction accuracy for Average Throughput considering

a. Mean Absolute Error: Decision Tree
b. Mean Square Error: Decision Tree
c. Root Mean Square Error: Decision Tree
d. Maximum Error: Decision Tree

Overall: Decision Tree

From the above data and graphs, we conclude ***Decision Tree to be the most fit Machine Learning Algorithm*** *for prediction when having Packet Delivery Ratio, Packets Dropped, Average Throughput, Average End-to-End delay as dependent variables and TCP Window Size and Packet Size as independent variables.*

ii. Finding the optimal TCP Window size and packet size:

Window Size	Packet Size	Packet Delivery Ratio	Rank	Dropped Packets	Rank	Avg Througput	Rank	Avg End to End Delay	Rank	Total Rank
20	1000	0.963728	28	251	40	1209.56	5	22.9817	1	74
20	1010	0.974506	10	168	22	1162.56	8	28.0714	2	42
20	1020	0.971706	17	218	34	1126.99	9	30.3915	7	67
20	1030	0.958193	35	272	42	1053.91	18	32.3363	8	103
20	1040	0.95273	39	17	2	131.501	36	59.496	16	93
20	1050	0.972488	13	170	24	46.355	46	968.618	46	129
20	1060	0.961158	31	245	38	59.3652	42	946.528	43	154
20	1070	0.963087	30	251	41	39.6188	49	959.218	45	165
20	1080	0.981772	3	93	11	929.51	31	72.7505	23	68
20	1090	0.9809	5	84	9	63.1461	41	1047.13	48	103
30	1000	0.978753	7	93	12	1210.89	4	36.7616	14	37
30	1010	0.968878	21	195	30	1091.83	12	30.3159	6	69
30	1020	0.972408	14	203	32	1111.73	10	36.1649	13	69
30	1030	0.949435	40	285	43	797.963	33	59.6258	17	133
30	1040	0.911929	46	19	3	80.0064	39	86.0233	29	117
30	1050	0.967373	24	237	36	1038.9	19	65.1736	19	98
30	1060	0.969405	20	175	25	1000.03	21	75.7588	25	91
30	1070	0.971868	16	181	26	82.5897	38	955.488	44	124
30	1080	0.982546	2	89	10	939.187	29	115.276	35	76
30	1090	0.975528	9	94	13	76.6061	40	1036.86	47	109
40	1000	0.974135	11	132	18	1217.22	1	33.627	10	40
40	1010	0.972736	12	155	21	38.9769	50	1180.37	50	133
40	1020	0.955361	37	49	6	58.9407	43	218.106	42	128
40	1030	0.828265	48	61	8	51.8736	44	81.934	27	127
40	1040	0.941332	44	52	7	134.957	35	129.252	37	123
40	1050	0.839604	47	12	1	47.088	45	121.766	36	129
40	1060	0.925883	45	109	16	241.296	34	68.2051	22	117
40	1070	0.976557	8	125	17	966.499	24	92.3146	31	80
40	1080	0.969621	19	200	31	953.165	27	110.703	34	111
40	1090	0.981422	4	101	14	93.1572	37	1079.2	49	104
50	1000	0.970126	18	169	23	1213.06	3	32.4715	9	53
50	1010	0.966819	25	187	28	1178.35	7	35.7462	12	72
50	1020	0.818985	49	26	4	40.1952	47	30.1965	4	104
50	1030	0.964242	27	221	35	1079.07	14	65.332	20	96
50	1040	0.968495	23	214	33	1054.25	17	82.0853	28	101
50	1050	0.947918	42	396	50	1014.83	20	51.3609	15	127
50	1060	0.95329	38	334	47	900.416	32	63.5357	18	135

Optimal Window Size + Packet Size

Window Size	Packet Size	Total Rank
30	1000	37
40	1000	40
20	1010	42
50	1000	53
60	1010	58
20	1020	67
20	1080	68
30	1010	69
30	1020	69
50	1010	72
20	1000	74
30	1080	76
60	1030	79
40	1070	80
60	1000	82
60	1040	83
50	1080	85
30	1060	91
20	1040	93
50	1030	96
30	1050	98
50	1040	101
20	1030	103
20	1090	103
40	1090	104
50	1020	104
50	1090	105
60	1020	108
30	1090	109
40	1080	111
60	1090	111
30	1040	117
40	1060	117
50	1070	119
40	1040	123
30	1070	124
40	1030	127

For finding the optimum window size and packet size, we gathered the data containing Packet Delivery Ratio, Dropped Packets, Average Throughput and Average End to End Delay. Packet Delivery Ratio and Average Throughput data was sorted in descending order (Higher is Better) while Dropped Packets and Average End to End Delay data was sorted in ascending order (Lower is Better). For all the four parameters, 'Rank' was assigned to the input combination of TCP Window Size and Packet Size, lower rank implying better performance. The ranks across the four parameters were summed to obtain an overall final rank. After sorting the final rank in ascending order (lower rank is better), we found the optimum combinations of Window Size and Packet Size.

Optimum Combinations:

i. Window Size - 30, Packet Size – 1000 bytes
ii. Window Size - 40, Packet Size - 1000 bytes
iii. Window Size - 20, Packet Size - 1010 bytes

From our analysis, we have found out that, among the possible combinations of TCP Window Size and Packet Size, **TCP Window Size of 30 and Packet Size of 1000 bytes leads to the optimum values of Packet Delivery Ratio, Dropped Packets, Average throughput**

Conclusion & future work

From the results obtained, we can conclude that –

i. The values of (Window Size, Packet Size) for which network congestion was found to be minimal through the evaluation of Packet Delivery Ratio, Number of Dropped Packets, Average End-to-End Delay and Average Throughput were **(30, 1000).**
ii. From the obtained data and graphs, we can conclude **Decision Tree** to be the most fit Machine Learning Algorithm for prediction when having Packet Delivery Ratio, Packets Dropped, Average Throughput, Average End-to-End delay as dependent variables and TCP Window Size and Packet Size as independent variables.

Future Work and Scope for Improvement -

i. For the sake of making a feasible dataset that we could work with, we chose to vary only 2 configuration parameters – Window Size and Packet Size and restricted the domain values. This work can be expanded to include many other configuration parameters available and test out to see if a trend in the variance of the evaluation metrics can be observed and to find if an even better combination of configuration metrics exists for which the network congestion is greatly reduced.
ii. Our work can be expanded to variants of TCP such as TCP Tahoe, TCP Reno, TCP New Reno, TCP Vegas, TCP SACK etc.

References

i. An Optimized Service Differentiated Congestion Management protocol for delay constrained traffic in Healthcare WSN's - https://www.researchgate.net/publication/349138043_An_Optimized_Service_Differentiated_Congestion_Management_protocol_for_delay_constrained_traffic_in_Healthcare_WSN's

ii. A Priority Based Congestion Avoidance Scheme for Healthcare Wireless Sensor Networks - https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4111427

iii. Congestion free opportunistic multipath routing load balancing scheme for Internet of Things (IoT) - https://www.sciencedirect.com/science/article/pii/S1389128620313049#:~:text=The%20proposed%20scheme%20manages%20the,devices%20and%20extend%20network%20lifetime.

iv. Congestion Free Routing Mechanism for IoT-Enabled Wireless Sensor Networks for Healthcare Applications - https://ieeexplore.ieee.org/document/9064704

v. Analysis of interplay between RPL and the congestion control strategies for CoAP - https://www.sciencedirect.com/science/article/pii/S1570870520306508#:~:text=In%20particular%2C%20it%20emerged%20that,instability%20of%20the%20routing%20protocol.

vi. An IoT based Congestion Control Algorithm - https://www.sciencedirect.com/science/article/pii/S2542660519302598