QUEUE ASSISTED CONGESTION AVOIDANCE SCHEME USING SOFTWARE DEFINED NETWORKS

By

SOBIA BIBI

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ISLAMABAD

March, 2023

Queue assisted Congestion Avoidance Scheme using Software Defined Networks

By

SOBIA BIBI

MIT, University of Sargodha, Sargodha, 2018

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

In **Computer Science**

To

FACULTY OF ENGINEERING & COMPUTER SCIENCE

NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

NATIONAL UNIVERSITY OF MODERN LANGUAGES FACULTY OF ENGINEERIING & COMPUTER SCIENCE

THESIS AND DEFENSE APPROVAL FORM

The undersigned certify that they have read the following thesis, examined the defense, are satisfied with overall exam performance, and recommend the thesis to the Faculty of Engineering and Computer Sciences for acceptance.

Thesis Title: Queue Assisted Congestion Avoidance Scheme Using Software Defined Networks

Submitted By: Sobia Bibi **Registration #:** 37 MS/CS/F19

Master of Science in Computer Science (MSCS)
Title of the Degree

Computer Science
Name of Discipline

Dr. Ata Ullah Name of Research Supervisor	Signature of Research Supervisor
Dr. Muhammad Nauman Malik Name of Dean (FE&CS)	Signature of Dean (FE&CS)
Brig. Syed Nadir Ali Name of Pro-Rector Resources/Director General	Signature of Pro-Rector Resources/DG

11th March, 2023

AUTHOR'S DECLARATION

I Sobia Bibi

Daughter of Ghulam Mutase

Registration # NUML/F19/35858

Discipline Computer Science

Candidate of **Master of Science in Computer Science (MSCS)** at the National University of Modern Languages do hereby declare that the thesis **Queue Assisted Congestion Avoidance Scheme Using Software Defined Networks** My original work, presented in partial satisfaction of the MSCS degree requirements, has not been previously submitted or published. I further vow solemnly that it will not be submitted in the future for any other degree from that or any another university or institution. I am also aware that if proof of plagiarism is discovered in my thesis or dissertation at any time, including after the degree has been awarded, the research may be rejected and the degree revoked.

Signature of Candidate

Sobia Bibi

Name of Candidate

11th March, 2023

Date

ABSTRACT

Title: Queue Assisted Congestion Avoidance Scheme Using Software Defined Networks

The internet of belongings is a topic that has gained popularity in recent decades and contains a significant amount of data. Utilizing and storing large amounts of data properly demands control. In this process, a substantial number of statistics is transmitted over the network. There is a chance that data packets will be dropped during transport and that buffer delays will occur. Continuously long delays at the network area can reduce its utility and cause congestion. To avoid this congestion, a hybrid community-managed design is introduced to fully utilize the community. Software Defined Congestion Control Plane is completely based on software-described network in which network statistics are collected by controller and by modifying transport layer parameters, behavior of end to end host are designated. It can also be used to alleviate security problems that could be referred as Feedback based Congestion Avoidance that can maintain a short router queue length with high network utilization. It automatically adjusts the congestion window according to feedback of remote control. The importance of this research study focuses on congestion in thenetwork links which is recognized by the flow table of the OpenFlow enabled switches. Software Defined Network tries to simplify the development and implementation of new congestion control algorithms by centralizing the control of the network in one entity called the controller. This research will lead to the exploitation of software based networks for the further improvement and management of various network components. We suggested a hybrid community-managed design to fully utilize the community in order to avoid this congestion. SDCCP is entirely based on software-described Networks that modifies delivery layer parameters. In this manner, we suggested a set of rules called FCA (comments-based fully Congestion Avoidance) that operates on SDCCP and completely changes the crowding window founded on feedback from the distant supervisor. FCA can maintain a short router queue length due to the threshold checks that are divided in small chunks. After every threshold chunk, average queue length is determined and decision is taken according to the residual buffer capacity. Due to time to time queue length threshold checks, proposed IAGRED schemes perform better results than previous schemes. The main contribution of the IAGRED scheme is towards packet loss and buffer delays. Proposed scheme decreases the packet loss and delays due to early detection which is clear from the results chapter with the help of different graphs. For Example from the Delay graph; at the start of packet arrival delay is tolerable in three schemes but as the time increased, proposed scheme behaves better than previous schemes.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	THESIS AND DEFENSE APPROVAL FORM	ii
	AUTHORS DECLARATION	iii
	ABSTRACT	iv
	TABLE OF CONTENTS	vi
	LIST OF TABLES	ix
	LIST OF FIGURES	x
	LIST OF ABBREVIATIONS	xi
	ACKNOWLEDGEMENT	xiii
	DEDICATION	Xiv
1	**INTRODUCTION**	**1**
	1.1 Overview	1
	1.2 Need for the Study	3
	1.3 Problem Statement	5
	1.4 Aim of Study	5
	1.5 Study’s Objective	5
	1.6 Research Question	5
	1.7 Scope of the Study	6
	1.8 Organization of the Thesis	6
2	**LITERATURE REVIEW**	**8**
	2.1 Overview	8
	2.2 Software Defined Networks	9
	2.2.1 Development of Software Defined Networks	11

	2.3	Routing Protocols	12
		2.3.1 Reactive (One Demand) Routing Protocol	13
		2.3.2 Multipath Routing Protocols	14
	2.4	Categories	15
		2.4.1 Sender-Based	15
		2.4.2 Receiver-Based	17
		2.4.3 Switch-Assisted	17
	2.5	Comparison of Energy Efficient Ad-hoc Network Routing Protocols	18
	2.6	Random Detection based Queue Management Scheme	19
	2.7	Resarch Gap and Directions	23
	2.8	Summary	26
3	**METHODOLOGY**		**27**
	3.1	Overview	27
	3.2	Operational Framework	27
	3.3	Research Design and Development	29
		3.3.1 Performance Parameters	30
		3.3.2 Throughput	30
		3.3.3 Link Utilization	30
		3.3.4 Mean Queue Length	30
		3.3.5 Packet Loss Probability	31
		3.3.6 End-to-End or Latency	31
	3.4	Simulation Framework	31
		3.4.1 Performance Metrics	32
	3.5	Summary	33
4	**Proposed Solution**		**34**
	4.1	Overview	34
	4.2	Adaptive GRED Scheme	34
	4.3	Improved AGRED Scheme:	36

		4.3.1 Queue Management	37
	4.4	Average Queue Length and Packet Drop Management	38
	4.5	Minimum Threshold and Maximum Threshold Parameterization	39
	4.6	Summary	45
5	**PERFORMANCE EVALUATION**		**46**
	5.1	Overview	46
	5.2	Experimentel analysis	47
	5.3	Results and Discussion	48
		5.3.1 Comparision of Link Utilization	50
	5.4	Summary	52
6	**CONCLUSION AND FUTURE WORK**		**53**
	6.1	Overview	53
	6.2	Conclusion	53
	6.3	Future Work	54
REFERENCES			55

LIST OF TABLES

TABLE NO.	TITLE	PAGE
2.1	Advantages and Disadvantages of Different Schemes	13
2.2	Summary of AQM algorithm	25
3.2	Fixed Parameters Used for Simulation	34
4.1	Parameters for Proposed IAGRED algorithm	51
4.2	Parameters for IAGRED algorithm	55

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
2.1	Basic SDN Architecture	11
2.2	Sender based information services	19
2.3	Topology for TCP and DCTCP experiments	20
2.4	A Schematic Diagram of a Sender-Receiver connection	22
2.5	Flow chart for Random Early Detection Algorithm	23
3.1	Software Defined Networks (SDN) Framework	30
3.2	A visualization of GloMoSim under execution with 50 nodes	35
4.1	Single router buffer for proposed IAGRED algorithm	49
4.2	Algorithm for the proposed IAGRED Algorithm	53
4.3	Estimating the typical writing time in a queue and packet loss	54
5.1	Probability of Packet Loss	61
5.2	Probability of Mean Queue Length	61
5.3	Throughput Vs Packet size	62
5.4	RTT in Open Flow	63
5.5	Comparison of Throughput	64

LIST OF ABBREVIATIONS

AGRED	-	Adaptive Gentle Random Early Detection
API	-	Application Programming Interface
AQL	-	Active Queue Length
AQM	-	Active Queue Management
ASIC	-	Application Specific Integrated Circuit
BER	-	Bit Error Rate
CIOQ	-	Combined Input and Output Queued
DCCS	-	Dynamic congestion Control Scheme
DRED	-	Dynamic Early Random Drop
EIRPG	-	Enhanced Interior Gateway Routing Protocol
EWMA	-	Exponential Weighted Moving Average
FCFS	-	First Come First Serve
FRED	-	Flow Random Early Drop
FRR	-	Fast Retransmit and Recovery
GRED	-	Gentle Random Early Detection
IAGRED	-	Improved Adaptive Gentle Random Early Detection
IETF	-	Internet Engineering Task Force
MAC	-	Medium Access Control
MQL	-	Mean Queue Length
MSS	-	Maximum Segment Size
NAT	-	Network Address Translators
NGN	-	Next Generation Network
ODP	-	Origin-Destination Pairs
ONF	-	Open Network Foundation
PDR	-	Packet Delivery Ratio
Qu’s	-	Quality of Service
QW	-	Queue Weight
RCP	-	Routing Control Platform
RD	-	Random Detection

RED	-	Random Early Detection
SDN	-	Software Defined Networks
SFB	-	Stochastic Fair Blue
SNMP	-	Simple Network Management Protocol
SRED	-	Stabilized Random Early Detection
TCP	-	Transmission Control Protocol
TM	-	Traffic Matrix
UDP	-	User datagram protocol
VLAN	-	Virtual Local Area Network
WTRED	-	Weighted Random Early Detection

ACKNOWLEDGMENT

First and foremost, I would want to express my gratitude to Almighty Allah for enabling this research. Without the honest assistance of many people, this study would not have been possible, and for that, I would like to convey my profound appreciation. Even so, there were significant individuals who contributed to where I am today. In particular, Associate Prof. Dr. Ata Ullah, who oversaw my data analysis, went above and above in supporting me through my research journey.

I will also be grateful to the administration of the Department of Computer Sciences, who supported me throughout my research career and enabled me to solve the issues I ran into. Thank you to everybody I forgot to mention but whose important work I will not ignore.

DEDICATION

This thesis is a tribute to my parents and instructors throughout my educational history who have not only unconditionally loved me but also encouraged me to work very hard for the belongings I aim to accomplish.

CHAPTER 1

INTRODUCTION

1.1 Overview:

The demand from computer network users has amplified as a result of the development and advancement of science and technology [1]. It increases the severity of problems related to network congestion [2].Applications like file sharing and downloading, internet browsing, and multimedia have all exacerbated network congestion issues. Networks, which must make a significant leap to keep up with the rapidly evolving technology, are seen to be growing slowly [3]. The researcher got to work on it and developed the idea of Software-Defined Networking (SDN) to satisfy the requirements. In the same way that servers and storage are being virtualized, SDN is a solution that allows networks to be easily configured and maintained [5].

In a standard IP network, devices like the data plane and control plane are combined into a single, streamlined unit. As a result, [6] setup becomes more difficult, complex and heterogeneity rise, and optimization falls. For this reason, data flow from the computer to the applications needs programmable structures and codes. In [7], one of the most sought-after developments in recent times is the widespread adoption of software well-defined infrastructure. In SDN, the control and forwarding functions of the network are decoupled for greater network flexibility, adaptability, and manageability [8]. Open Flow solutions enable customizable network traffic management, smarter systems, and provide a better foundation for Internet innovation.

To compare each AQM algorithm and predict the desired characteristics and performance flaws of each method the variables used are Bandwidth allocation, link usage, throughput, and mean queue length.

To locate open-source controllers that controls a sizable portion of the network. Algorithms with a performance focus on networks have been the subject of numerous research studies. Although the dispersed design of the control layer offers advantages, network virtualization and fast recovery are notoriously challenging. By concentrating network control in a single entity known as the controller, SDN seeks to make the creation and use of novel congestion control algorithms simpler. Only the controller's requested actions must be performed by the switch. As a protocol for statement among the switches and controller, Open Flow is introduced.

The demand on the network has increased due to the exponential development in file transfers and multimedia applications [14].The demand from users for computer networks has increased as a result of the development and advancement of science and technology. It increased how serious network congestion issues were. Applications like file sharing and downloading, internet browsing and multimedia have all exacerbated network congestion issues. Networks, which must make a significant leap to keep up with the rapidly evolving technology, are seen to be growing slowly. The researcher got to work on it and developed the idea of SDN to satisfy the requirements. This increases the likelihood of congestion, adds extra Round-Trip Time (RTT), and necessitates packet retransmission.[15][16] Software-Defined In the same way that servers and storage are being virtualized, networking is a solution to make networks easier to configure and maintain. In the conventional IP network architecture, network components like the data plane and control plane are integrated into a single device. This makes configuration challenging, increases complexity and heterogeneity, and hinders optimization. In order to enable data transmission from the network to the application, programmable models and codes are required. The most popular technological developments nowadays are software-defined, from infrastructure to apps. By separating the network control and forwarding tasks, SDN makes it possible for the network to be directly programmable, dynamic, and manageable [17]. Therefore, effective cramming control is a crucial component to guaranteeing the constancy and sturdiness of the system [18].

There are two strategies for resolving the network congestion issue [19]. The end-to-end strategy is the first, [20] while the network side approach is the second. In end to end congestion control strategy, network does not provide clear feedback to senders about when to reduce their speed. Instead, congestion is ascertained by packet loss. While in network side approach, routers provide clear information about how fast the end system should send. The difficulty with active queue management has shifted due to the decline in network performance brought on by the rising load (AQM) [21]. Active queue management aims to avoid full queues by alerting the transmitter to congestion in advance rather than waiting until it occurs. In a nutshell, congestion control is done so that senders could get their share of network resource but network resources must be used well.

1.2 Need for the Study:

It's crucial to identify and eliminate network congestion if we want to keep data communication running smoothly. Previous studies found that the typical IP network would need to adapt the size of the crowding window and queue length to account for variations in data traffic when the basic network topology and data of flow are unidentified [22]. These findings demonstrated that the standard IP network cannot ensure link operation and excellence of facility because it lacks direct control over the forwarding queue. This drives the research study's motivation to address the issue of network congestion. The worldwide nature of the existing algorithms for congestion monitoring means that they are not aware of every link in the system. The network cramming issue can be more effectively condensed using the software-defined method [23].

Troubleshooting is the responsibility of network managers, but finding the main reason why a network is congested is more difficult than it sounds [24]. Large amounts of data are shared by websites like Netflix and YouTube that are not linked to business. The transition to a digital learning process for the next generation of education systems and learning management systems will occur soon and call for smooth data flow [25].

There is a big profit margin for video traffic, big data, and mobile usage that also posing tough problem for network operators [26]. Due to an increase in mobile users, an enormous rise in the number of servers and computer-generated machineries in data center networks, and an increase in server-to-server communication traffic, mobile and telecom providers are experiencing spectrum congestion. The management of congestion will be crucial to enhancing network efficiency [27].

Congestion is alleviated in networks when the sender sends out segments of data packets to the recipient up to the specified window size. It holds up if the network is big enough to include sluggish intermediary links between both the sender and the recipient. To pinpoint and fix congestion issues in a network, three stages are required. The initial step is to verify congestion [28] which may be done by observing the situation. The huge amount of data and the fast rate of data transfer will cause a large number of dropped packets as well as the standing in line of packet header at the router buffer. An increase in end-to-end latency or congestion, as well as a decrease in throughput, are the results of a low percentage of packets delivered i.e., PDR, which is produced by a large number of dropped packets that compel the nodes to rebroadcast the data [29].

The aforementioned issue cannot be solved by the typical network architecture. As a result, a software-defined network will be a compelling option to tackle congestion-related problems because it offers global visibility and control over network flows. It offers consolidated control-plane astuteness, a distinct data level, and open, user-controlled management.

The primary responsibilities of a network administrator include access control, server load balancing, traffic monitoring, and routing. The network administrator must setup each device separately because it has its own control plane [30].It results in increased time, expense, troubleshooting, and lack of network visibility. Network administrators need an effective, adaptable, agile, and scalable network to meet these problems. The SDN strategy divides the control system from the statistics level, reduces networking equipment to basic forwarding devices, and offers software programmability. Through a software module, the software-defined controller in SDN takes control of the network. By constructing a new way that allows flooding on the ports, the controller may keep a network connected using the open source SDN component [31]

1.3 Problem Statement:

During the excessive messaging, congestion may occur when massive data is transmitted over the network whereas a few links are suffering from low bandwidth. In this situation, the congestion avoidance mechanisms suggest to set the transmission rate for video sharing equal to the lowest bandwidth between any two nodes in the entire path between sender and receiver. The main problem is that the queues at the intermediate nodes in the communication path are not fully utilized. It is mandatory to manage these queues especially in case of low bandwidth. It results in low throughput and excessive packet loss [8].

1.4 Aim of study:

The goal of this research learning is to avoid congestion in the SDN based network traffic by implementing the active queue management algorithm in the controller.

1.5 Study's Objectives:

The following are the study's main goals.

- To identify the effective queue management to manage transmission during lowbandwidth and enhancing the throughput.
- To reduce the packet loss during congestion scenarios.

1.6 Research Questions

The following are the main research questions.

- How to reduce congestion in low bandwidth scenario in SDN using residual BufferCapacity?
- How to reduce the packet loss during congestion with low bandwidth scenario?

1.7 Scope of the Study:

SDN has a problematic future as it has issues to overcome. The first issue is the console/remote capability with today's security issues many will not want to expose their network to a potential hacker takeover. It is an Open Source Technology. Another issue is the current state of Network Management policies and practices with single device or single path focus. When a Network Manager looks at SDN he only sees how it can help or hurt his network but SDN is much bigger and a lot of education still remains to be done to make SDN or similar technologies palatable.

SDN is a Human Centric Technology where today's technology is Device Centric which is and always will be a challenge to get managers to adopt especially when one person can completely change the network, storage, WAN etc. fabric.

Many talk about SDN's ability to help with the "Cloud" but before we get SDN involved we need to get the "Cloud" under control. There are also serious financial, training and primary deployment issues. SDN opens lots of questions for the future of advance networking and computing technologies. We need more responsive technology but not at the cost of security and control. In the long run SDN may be deployed in provider networks but individual corporations may find it just too much to deploy. SDN is in need of a lot more development and proof of being a secure and deployable technology [33].

1.8 Organization of the Thesis

This thesis is prepared through six chapters. Chapter 1 briefly describes the overall thesis; in this chapter introduction is given about all research work. Chapter 2 briefly reviews the previous works that pertain to the congestion in the networks and factors that favor the problem of congestion and methodology to avoid the congestion by considering various parameters. Chapter 3 presents the concepts of active queue management algorithms relating to the congestion issues. Chapter 4 proposes an enhanced AQM algorithm by improving the best existing by changing the parameters such as minimum threshold, the maximum threshold, queue weight, and maximum value of packet dropping probability. Chapter 5 describes the proposed research contribution by varying the values of queue management parameters in order to avoid packet drops. Chapter 6 presents the conclusion describing the successful implementation of proposed Improved Adaptive Gentle Random Early Detection algorithm in SDN and the future enhancement of the research work.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This chapter briefly reviews the previous works that pertain to the congestion in the networks and factors that favor the problem of congestion and methodology to avoid the congestion by considering various parameters. This chapter also review software defined networks, its protocols and characteristics under the perspective of various researchers. Random Early Detection Mechanism and Its Variants in these topic Active Queue Management (AQM) algorithms are an Implementing schemes, So that packets are transmitted with higher priority than others. Random Early Detection (RED) is the first active queue management algorithm proposed for deployment in TCP/IP networks. RED has some parameter tuning issues that need to be carefully addressed for it to give good performance under different network scenarios. Various algorithms come from RED such as Stabilized RED (SRED), Dynamic RED (DRED), Adaptive RED (ARED) and Flow RED (FRED) these algorithms control congestion by discarding packets with a load dependent probability whenever a queue in the network appear to be congested, this paper will introduced some features about RED and its variants.

RED doesn't able to stabilized the queue size, while DRED and SRED both stabilize the queue size very well and also they have more predictable packet delay inside the network [36]. SRED has higher drop probability and higher packet loss rate than DRED. RED, ARED and DRED monitors the queue length while SRED can monitor the queue length and packet header, RED has a hard time maintaining the queue size between the two thresholds. SRED effectively controls the queue size at the expense of a larger buffer size. DRED is continuously adjusting its drop probability to reflect any change in the traffic load, Compared to DRED.

The problem of congestion occurs at the buffers due to the available resources of a network not accommodating all arriving packets. Therefore, the performance of the network deteriorates such as obtaining the high mql, the low throughput (T), the high average queueing delay, the high PLoss and the high DP (DP). Active queue management (AQM) algorithms [8] can be applied to control the congestion situation.

2.2 Software Defined Networks:

Manor Jamal et al [37] observed at SDN, in which a network's control and data planes are treated as distinct entities to provide more adaptable and efficient network management and operation. Components of the network, such as routers and switches forward data packets in the management plane based on instructions from the operators. Open Flow interface is used to manage the SDN architecture in general. It is the first install the new designed to enable the data and control planes. Figure 2.1 depicts the data forwarding, control plane, and management plane as established by the survey.

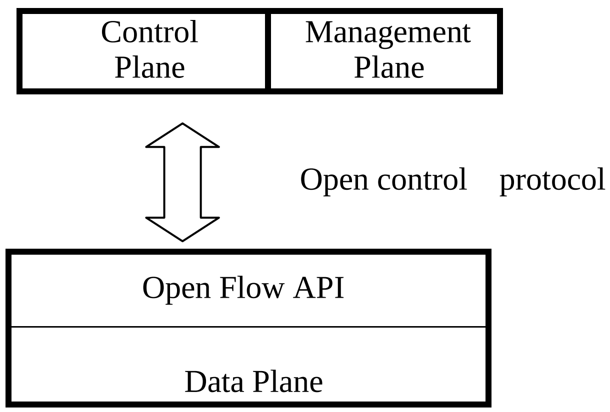


Figure2.1: Basic SDN architecture

Networking can be defined as the act of establishing a connection among the equipment's in order to exchange information. The traffic should be monitored for seamless data transfer. The traditional IP network is hard to reconfigure to the increasing load and changes. This is due to the tremendous growth in Internet users and large network traffic generated. SDN emerges as a new approach in computer networking as a solution for large complex network and seamless throughput over network. Software defined means that the functionality of the network component is defined by the software, that is, it establishes its function of the equipment controlled by the software. This study opens new approaches for further studies and progresses in the field of congestion control. The main goal of this review is to identify previous methods, concepts, and challenges in congestion control and development in SDN.

- **Data Plane**

Control plane mentions to the operations and procedures involved in the transmission of sequences or frames from one connection to another. It also is known as a transport aircraft. Data plane traffic via the router. The control plane is utilized by routers and switches to direct incoming packets to their destinations.

- **Control Plane**

The phrase "control plane" is used to describe the set of operations that, according to the routing protocol, choose which path to take and create a routing table. The router uses the control plane information it processes to modify the forwarding table. Additional features include system and administration setup, network architecture interchange, and so forth.

- **Management Plane**

Every tier of the network stack relies on the management plane for its administrative, measurement, and customization needs. Management planes are used to keep track of the health of various devices, and a common management protocol is the SNMP protocol.

2.2.1 Development of Software Defined Networks:

The Software Defined Network has arisen as a technical tendency that has involved in academics, hardware manufacturers, service providers, and consumer and software developers in a manner that has never been seen before. Historically, computer networks are administered with the skill of dealing with difficulties by bringing additional protocols to protocol suites and by resolving network operating issues. SDN generated tremendous enthusiasm in the field of telecommunication since it introduced modularity to networking for the first time.

The SDN differs fundamentally from conventional data networks in that the control plane and forwarding plane are kept completely separate. Perhaps most significantly, additions to both records and the control plane can develop in parallel. SDN provides a new level of adaptability, allowing for the programmability and control of large-scale networks. In the past, traditional networks have been unable to do anything of significance here. Very encouraging progress has been made recently in this line of inquiry, but many difficult problems still lie ahead. In Table 2.1 related schemes are discussed with respect to their basic idea, mechanism, advantages, and disadvantages:

Table 2.1: Advantages and Disadvantages of Different Schemes

Scheme	**Basic Idea**	**Mechanism**	**Advantages**	**Disadvantages**
DAPIF	To cope Broadcast Storm problem for content centric vehicular network	Neighbor density, Time based distance between sender and the receiver.	There is no need for vehicles to be aware of the location of the neighbors or a content provider	Difficult in parameter setting Insensitive to busty traffic

RBM	Network resources Consumption, Broadcast overhead	Effective communication distance	Saves network resources and controls the broadcast storm problem.	Loss rat of packet is high Fluctuation of congestion window
DISFIRE	Authentication & authorization	Grid network	Hierarchal cluster network with multiple SDN controllers implement a dynamic firewall to ensure authorization	Evaluation of framework is lacking. The protocols used is pollex which is not practically tasted
SDP	Infrastructure management and reconfiguration	Centralized controller with dumb data plane	The state full approach, reducing information exchange, Mobility, reconfiguration and localization	In-depth architectural details are missing and lacking security is reliability.

2.3 Routing Protocols:

Here, we describe in depth the different types of communication algorithms, and then we assess the effectiveness of different routing methodologies in ad hoc networks by trying to compare their respective procedures across a wide range of criteria, including path performance indicators, time complexity, computation complexity, and parameters of the phase transition. The navigation meter is an integral part of routing protocols in ad hoc networks since it is used to determine the most efficient path for packet forwarding [39]. The address space and building layout might be more or less flat depending on the route structure. The temporal complexity (TC) of a protocol is the amount of time required to carry out the stages of executing the protocol. The number of messages transmitted and received during a single protocol operation is known as its problems (CC).

2.3.1 Reactive (On Demand) routing protocol:

In order to use reactive routing approaches, a node must wait until it has to transmit a packet to a certain destination before beginning a route discovery [40].The route tables they use are not regularly updated to reflect the current state of the network. Consequently, there is reduced communication overhead, but more time is needed for the on-demand routing building process.

Dynamic source direction-finding is the on-demand routing technology used by Johnson et AL Principal (DSR). DSR is the name of the most used source routing protocol .Because each node must send control packets whenever necessary, battery-operated control is captured on the nodes. By delivering the routing message only when necessary, the network bandwidth overhead is reduced [41].

On-demand and ad hoc detachment vector (AODV) Perkins et al. [42] recommended employing AODV to produce loop-free pathways despite the need for predetermined failure paths. Time to Live (TTL) limits a node's capacity to forward unnecessary packets, hence decreasing control overhead. Since the performance of this protocol relies on connectivity and end-to-end latency, the route cache strategy is not employed. Temporarily well-ordered routing procedure (TORA) Park and Corson created TORA, a routing algorithm that is adaptable and scalable. TORA utilizes the "link reversal" technique. The planned use of the protocol is a mobile, highly self-motivated wireless communication situation. A directed graph (DAG) with a height as its root is formed at a point.

Ton developed an associatively-based distributed routing algorithm that is straightforward and efficient (ABR). ABR uses route stability as a factor for selecting the best route. The ABR procedure uses a method called associatively ticks to establish and preserve a "degree of associatively". There are no loopholes, deadlocks, or duplicate packets in the protocol [43].

What is variously referred to as a "route cache," "route table," "route metric," "shortest path," "routing structure," "hierarchical routing repository," and "flat routing repository" is actually all the same thing. Intricacy of communication (CC); The initials for the term "time complexity" are "TC;" where l is the diameter of the impacted network segment, y is the entire amount of nodes going to make up the directed path, p is the length of the reply's clear route, and x is the amount of clusters; z is the diameter of the directed path.

2.3.2 Multipath routing protocols:

The main goals of the multipath routing protocols are to increase the quality of service (Qu's) in an ad hoc setting, ensure load balancing, and enable reliable communication. Multiple pathways' identification and upkeep are difficulties that are addressed by multipath routing technologies [44]. Protocol for caching and multipath routing (CHAMP): The shortest multipath routing and data caching are both utilized by the Valera et al. The primary design objective is to reduce packet drops brought on by frequent route failures.

Mavropodi et al. [45] created the secure multipath routing (secure) protocol, which secures on-demand multipath routing. This protocol incorporates numerous security enhancement features to prevent security attacks from cooperating hostile nodes. Authority issues a certificate for the secret keys (CA). Techniques for geographic multipath routing with energy and mobility awareness (EM-GMR) Liang and Ren [46] have devised a multipath routing system that takes energy efficiency and user mobility into account. The EM-GMR specifies that a mobile node's next hop shall take into account the node's remaining battery life, the node's mobility, and the node's distance from the destination. Certificates for the private keys are issued by the proper authorities (CA). Tools for energy- and location-aware multipath routing (EM-GMR) Liang and Ren developed a multipath routing system that takes energy and mobility into account and is founded on a fuzzy logic method [47]. According to EM-GMR, a mobile node's next hop must take into consideration the node's remaining battery life, the node's movement, and the distance between the node's current location and the destinations. A custom fuzzy logic system is built into the next hop selection mechanism. The authors developed 27 criteria for the inductive inference set to use as criteria for choosing the next hop node. Braids for multi-hop routing (BMR) In order to improve network performance, Ganesa et al. [48] designed a braiding multipath routing protocol. The BMR protocol's route discovery phase involves every node discovering its own unique set of optimal pathways between a given source and a given destination [49]. Method for Authentic Multi-Path Routing Protocol (TMRP) Wang et al. proposed TMRP, which is applicable to network with greedy, non-cooperative nodes, and which determines the forwarding cost of a packet based on the resources available at each node. Based on traditional AODV, Marin and Das [50] proposed ad hoc on-demand spread spectrum shortest path routing (AOMDV).

This protocol's primary goal is to provide several loop-free, link-disjoint pathways. This protocol makes use of the "advertise hop count" suggested metric. The maximum permitted hop amount for each path logged at a node is referred to as the advertised hop count for that node.

2.4 Categories:

The research literature has grown rich with numerous promising solutions to the in-cast congestion issues in datacenter networks during the past few years. [51] These works can generally be divided into three categories:

2.4.1 Sender-based:

There is a disparity between TCP timeout clocks in the guests and the actual round-trip timings (RTTs) encountered in data center network DCNs, dependent on the sender. So as to allow TCP timeout detection in microsecond increments, it was proposed to modify the transmitter transmissions control Protocol TCP stack to include high-resolution timers. Data center TCP (DCTCP) suggested changing the Congestion control window modification function to respond appropriately to the urban agglomerations [52]. Unbiased Early Detection Active Queue Management settings for RED-AQM are adjusted to impose a tiny Explicit Congestion Notification.

While both approaches have the potential to lessen mouse traffic delays, implementing DCTCP successfully requires substantial adjustments to the TCP receiver and sender algorithms, in addition to fine-tuning the RED settings at the switches. When, it was discovered that the TCP timeout timers on the hosts were not in sync with the actual round-trip times (RTTs) encountered by DCNs [53] it was recommended that high-resolution clocks be used in the sender's TCP stack to allow TCP timeout detection on a microsecond time scale. DCTCP proposed modifying the TCP congestion windows optimization framework such that it would have a more linear relationship with congestion severity. Adjusting the RED-AQM settings to mandate a low nearly indistinguishable threshold reduces waiting times. Both approaches have the potential to lessen the delays in mouse communications, but they need tweaks to TCP receiver and sender algorithms and adjustments to RED settings at the switching for DCTCP [54] as shown in Figure 2.2.

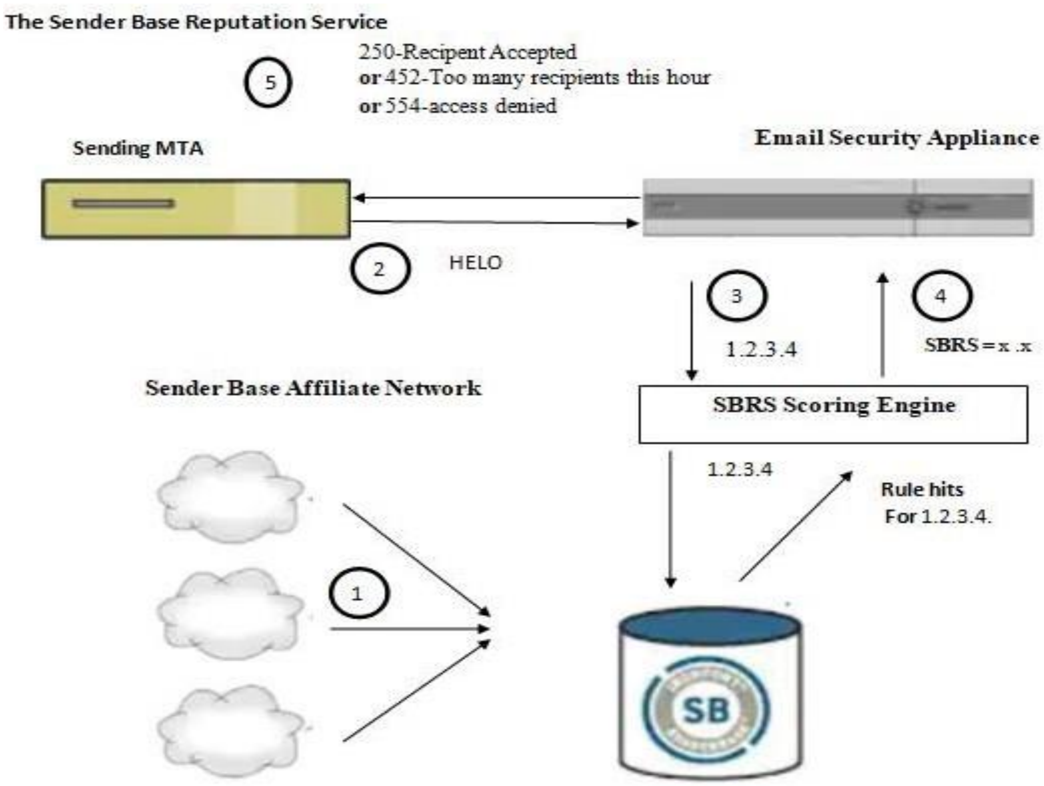


Figure 2.2: Sender base information services [55]

2.4.2 Receiver-based:

“Receiver-based ICTCP” was put forth as a TCP receiver alteration to handle in cast traffic. Prior to packets being discarded, In Cast Congestion Control for TCP (ICTCP) modifies the TCP receiver window. Tests on a real tested show that ICTCP can nearly fully eliminate timeouts and give a high performance for TCP in cast traffic. The problems of buffer building caused by elephant and mice sharing a buffer are not addressed by ICTCP, unhappily. Additionally, [56] it requires modifications to the TCP receiver algorithm and is only effective if in cast congestion occurs at the destination node.

Note that our tests were conducted with 100Mbps links, whereas the original paper used 10Gbps links. With these validation experiments complete; we were ready to move on to reproduce our main result [57].

2.4.3 Switch-assisted:

With slight adjustments to Drop Tail AQM, the authors suggested switch-assisted AQM strategies to control TCP transmission rate. To determine a fair share for each flow, RWNDQ keeps track of the number of established flows [58].The TCP receiver window is changed to explicitly inform TCP sources of the estimated share. Information Management Quality to forecast potential in cast congestion in the upcoming few RTTs, In-Quality Monitoring monitors the switch for TCP connection establishment and disconnection shown in Figure 2.3.

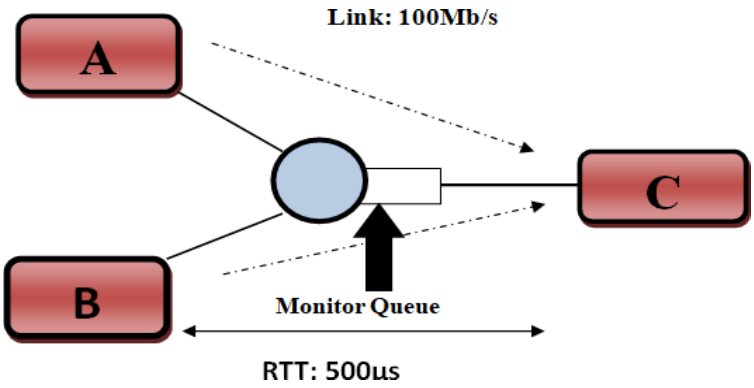


Figure 2.3: Topology for TCP and DCTCP experiments [58]

In order to allow space for incoming cast traffic, the receiver window of ACKs acknowledgment is reset to 1 MSS when congestion is approaching. Both strategies are demonstrated to reduce elephant traffic timeouts and provide high throughput, although they both call for switch software update [59].

2.5 Comparison of Energy Efficient Ad-hoc Network Routing Protocols:

We looked at different routing protocols used in MANETs and learned about them here . Based on the routing strategy and their updating mechanism, these protocols were selected as the classification of a number of routing schemes [60]. In the mobile ad hoc network, we talked about some key features of these routing techniques, including table-driven, reactive, and hybrid protocols. Finally, an effort has been made that focuses on proportional learning of a select group of routing techniques, including AODV, DSDV, DSR, TORA, WRP, CGSR, and ZRP. Each of the selected methods has unique benefits and drawbacks. A comparative table that details their proficiencies in various scenarios has also been set up.

2.6 Random Detection based Queue Management Schemes:

In extremely large networks with high traffic, congestion control is key to managing and network resources. Algorithms for active queue management are one method for tackling congestion issues. Most AQM algorithms concentrate mostly on single queued links. The attention is focused on combined input and output queued (CIOQ) switches because the throughput of enter queued switches is constrained and output queued switches need a significant speedup factor. Particqularly employed as a cramming display rather than only the output column distance is the weighted sum of the input and output queue lengths.

Abdeljaber et al. [8] proposed an AQM method with aim to enhance the performance of GRED method regarding to mean queue length (*mql*) and average queueing delay (D). AGRED improved the tuning of D_{max} and max*threshold* position. AGRED is similar to GRED except in case that the value of *aql* is between the positions of max*threshold* and *double*max*threshold*. In this case, AGRED calculated D_{init} in away different than that of GRED. The result of D_{init} varies from D_{max} value to 0.5 as the value of *aql* varies from max*threshold* to *double*max*threshold*.

The improved RED (IM-RED) algorithm is considered as one of the RED’s variants [7], two DP functions were used: nonlinear and linear. When the traffic load is either moderate or light, the nonlinear DP function is applied, and this nonlinear function is a quadratic function. When the traffic load is high, then the linear DP function is used. IM-RED employs three thresholds. THmin, THmax, and Target. Where THmin, and THmax are lower and upper queue threshold positions, respectively [7].

According to Hollo [61], the Random Detection (RD) Queue Management technique randomly selects and drops packets from a pool of incoming packets whenever congestion is identified (2002). Every time a packet comes, a random number is created, increasing the pool's N-packet total. Once saturation has been recognized, each incoming packet has a 1/N likelihood of being selected for dropping. The advantages of RD are not as well suited for congestion avoidance as they are for congestion recovery in smaller networks. Figure 2.4 represents the Schematic Diagram of a Sender-Receiver connection.

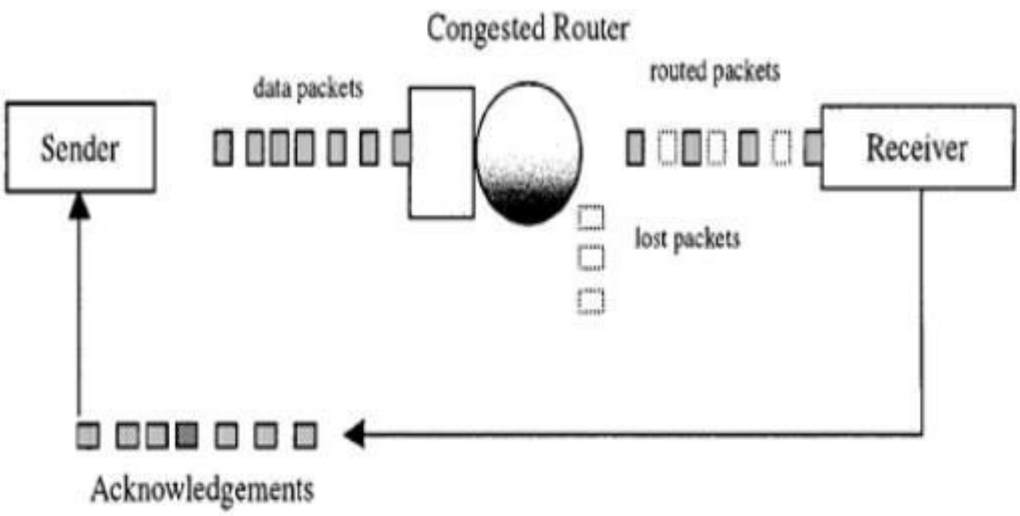


Figure 2.4: A Schematic Diagram of a Sender-Receiver connection [62]

The basic steps and flow of congestion detection in the random early detection algorithm is as shown in Figure 2.5.

Step 1: The parameters are initialized

Step 2: Then, in each time slot, a packet may be generated according to the probability of packet arrival and is sent to the queue.

Step 3: The packets that are generated and sent, maybe lost if the queue is full, or it might be dropped or queued according to a decision made based on the AQM and generate the dropping probability value.

Step 4: In the same time slot, a packet maybe departed according to the probability of packets departure. These processes are repeated in each time slot.

Step 5: Finally, the results are collected and reported

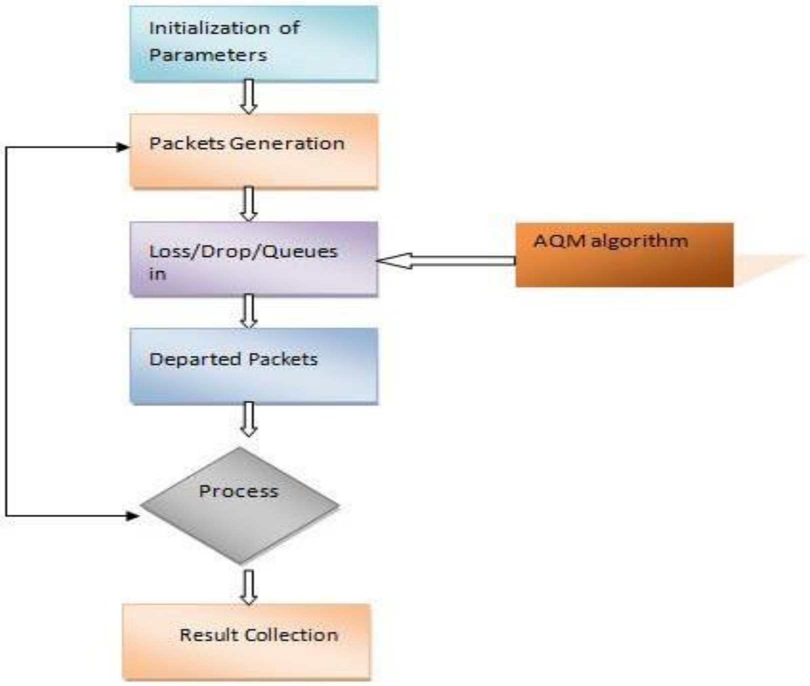


Figure 2.5: Flow chart for Random Early Detection Algorithm [63]

Random Detection (RD) suffers from a lack of fairness since sources with higher traffic volumes would discard more packets than sources with lower traffic volumes. Regardless of the congestion avoidance condition, where packages begin to drop, packet delivery continues, causing connections whose transmission rate has slowed to lose packets. Random Early Detection (RED), one of the many AQM techniques, has probably undergone the most research. The traits of RED include

(i) The sensitivity of RED performance to its system parameters.

(ii) For RED to function effectively, it is necessary to adjust the following four settings: highest threshold (math), minimum criterion (month), highest packet losing probabilities (map), and weighting factor (WE).

(iii) RED's performance is packet size dependent, so keep that in mind.

The success rate of the RED mechanism will be determined by optimal parameterization. RED stands for a group of queue management systems that do not keep track of the status of each flow. In other words, they combine all of the data from the flows into a single queue and concentrate on overall performance.

In packet switched networks, random early detection (RED) for congestion avoidance calculates the average queue size to identify first congestion. This set of rules alerts connections to congestion by either discarding packets coming in through the gateway or by inserting a few bits into the packet header. The entryway beads or marks every incoming packet with a positive chance when the average queue size reaches a predetermined threshold; the precise opportunity is a property of the average queue length. Gentle RED, Flow Random Early Drop (FRED), Dynamic Early Random Drop (DRED), and a few discrete queue analytical models are a few of the alternate AQM algorithms.

A variant of RED is called Flow Random Early Drop (FRED), which was developed by Dr. T. Bhakra Reddy [64]. The choice to drop a packet is based on bandwidth use and the number of current flows. This algorithm depends on the buffer size and its major goal is to maintain track of the packets in the buffer. The advantage of FRED includes:

- **(i)** Per flow queuing and Round Robin scheduling,
- **(ii)** Accepts data packets from minimum flow per buffer usage,
- **(iii)** When buffer space is full, two packets buffer is provided.

Another active queue management technique, Blue (W. Feng) [65] bases queue management on network utilization and packet loss. It is determined whether to mark or drop packets. The packets will be lost and the probability factor will be increased by p1 if the queue is full. The probability factor is decreased by p2 if the queue is empty. P1 should have a higher value than P2, and vice versa. To make use of the link, this is done. Another parameter employed by the FRED is termed freeze time, and it is used to update the probability prior to the effect being applied. The advantages and disadvantages various active queue management algorithms are summarized as in Table 2.2

Table 2.2: Summary of AQM algorithm [66]

Sr. No	AQM Algorithm	Advantage	Disadvantages
1	RED	• Early cendmongestion Detection • Avoid Busy Traffic	• Difficult in parameter setting • In sensitive to bust Traffic
2	SFQ	• Simple implementation of algorithm • Has low end-end delay	• Loss rate of packet is high • Congestion window
3	REM	• Computation of load Is low • High link utilization	• Throughput is slow
4	FRED	• Round Robin Scheduling	• Maintain per flow state
5	SRED	• Stabilized queue Occupancy	• Maintains additional list
6	BLUE	• High Throughput	• Use packet loss history

2.7 Research Gap and Directions:

With this survey as a preliminary point, various research lines happen that call for therein additional study. In the sequel, a few of these threads are highlighted. To implement consolidated control and cross-layer optimization in MANETs, there are two key bottlenecks that require further research in order to be addressed effectively:

- When the CP announcement is in-band, the overhead of control signaling and the resulting latency can cancel out the advantages of centralized optimization. Depending on the complexity of the networks and the phrasing of the optimization process, it may take too long for centralized optimization techniques to generate a real-time feasible solution. In order to save costs, a single-hop, out-of-band network manager is a viable option, and this is made possible by the ongoing development of state-of-the-art SDR designs and algorithms that enable the creation of high-quality dual spectrum transmitter for the first time. There have also been suggestions for more developed form of centralized control, which can reduce control overhead by generating future-proof solutions via prediction methods. These algorithms strike a solid compromise between centralized and decentralized control, aggressively compute and distribute answers under a variety of future conditions, and have showed enormous potential for minimizing the negative effects of in-band control overhead.

- For the second bottleneck, several iterations of the cross-layer optimization method have been tackled by developing sub-optimal algorithms that sacrifice precision for computational speed. Cross-layer optimization is still a challenging issue due to the complexity of the full-fledged total network optimization method, which includes consideration of routes, scheduling, MAC, and PHY layer features. It's hard to make informed decisions when we can't reliably determine the latency between both the two nodes. That way, the SDN protocol will only be able to track lag time between the active controller and the switches. To add insult to injury, even if such discoveries were possible by modifying the procedures, the data on time slot latencies would still be unavailable.

There are two patterns of congestion control algorithms, described by the way they examine congestion. Random Detection (RD) control, like tail-drop, RED and DRED, estimates only the number of packets in the buffer to calculate the strictness of congestion [67]. Arrival rate-based (RB) control schemes, such as REM or GREEN, estimate the packet arrival rate, as well as probably the backlog, to estimate congestion and determine P(t). The parts of AQMs into BB and RB reflects the historical development of AQMs, from being improvements to the packet dropping behavior of the tail-drop queue, to the current control systems approach for regulating the aggregate arrival rate X(t).

In this modern world, swift communication technologies are needed. The number of internet users is increasing rapidly. Latest communication technologies are indented to develop for transmission of huge amounts of information. A congestion control method is essential to keep any network energetic and decent for the users.

The objective of congestion control is to achieve efficiency, i.e., maximum utilization, minimum queue size and minimum packet drops while giving fair access to all the sources. The two main approaches to congestion control are end system congestion control and network centric congestion control [68]. In the end system congestion control, senders detect the congestion and react to it accordingly. TCP is an important example of end system approach. While a packet is dropped, the sender assumes that congestion has occurred and reduces the sending rate. When a packet is successfully transmitted, senders increase their rate.

The other approach is network centric congestion control. The idea behind this is that the routers have more information about the state of the network; and they can be useful in detecting congestion and should take part in the decision of congestion control. Routers measure the congestion level through comparing input traffic and by way of searching on the queue length; thus, they can send feedback as soon as they are aware that the queue length is developing. Consequently, the average queue length does not need to be as big as in the previous approach. Routers could also be used to give priorities to some sources as compared to others. An important example of network-centric congestion control is Active Queue Management.

In an SDN implementation, the controller typically oversees a number of Open Flow switches, each of which connects a set of hosts. When a packet comes, the switch searches the flow table for its entry. If the lookups are complete, the switch implements the operations in the matching table entry to the packet, typically forwarding it to the appropriate interface. If not, the switch sends a packet-in message to the higher SDN controller, suggesting that the packet should really be part of a brand-new flow. The controller generates the required flow rules and communicates them to the switch in packet-out or flow-mod packet format. Consequently, a flow is transmitted to the SDN controller.

The simulation compares the AGRED with the two active queue management algorithms RED and GRED in terms of packet loss and mean queue length. When traffic is produced by TCP sources with varying packet sizes, the simulation's results are obtained. The results, which are presented as a graph, demonstrate that the AGRED strategy is more effective than the first two approaches [69].

Different forms video and audio compression, less sensitive data to delays, and compressed video and audio), with varying levels of service quality, are handled simultaneously in modern telecommunication networks. As a result, techniques for managing traffic are crucial to its optimization and reducing network losses. To build a network approach (topology and its elements' characteristics), interactively replicate its operation mechanism, optimize its characteristics, analyses and manage the traffic, and try to assess the achievement bounds, researchers in the field of complex systems must use more sophisticated multi-method software and hardware tools.

2.8 Summary:

Different continuous object border tracking and detection algorithms are described in this chapter. The comparative study of various schemes in terms of energy savings, number of nodes employed, Hop-count, SN and new route, Local affiliate inquiry, Temporally Order Optimization Approach, and Zone routing protocols is offered. The fundamental, mechanism, and benefits and downsides of various methods are discussed. The performance and dependability of ad hoc mobile networks can be significantly enhanced by the implementation of efficient routing algorithms. Numerous routing protocols have been developed for such networks. Ad hoc on demand Distance Vector (AODV), Dynamic Source Routing Protocol (DSR), and Destination-Sequenced Distance-Vector Routing protocol (DSDV), Optimum Link State Routing are among the most used (OLSR). Despite the ubiquity of these protocols, little research effort has been devoted to analyzing their performance with changing bit rate (VBR).

CHAPTER 3

METHODOLOGY

3.1 Overview:

This chapter explores the main methodology followed to complete to this research work. We initially present the operational framework where main modules are explored. Next, we present the number of different AQM algorithms are in literature. Some of them were developed based on intuitive reasoning and rest by a modeling and analytical approach. Here we will try to give some structure to the different AQMs available, and describe the structural restriction of some approaches.

3.2 Operational Framework:

Although SDN providers provide a variety of competing topologies, at its most fundamental level, software-defined connectivity centralizes network control via off-device computing resources. The SDN construction contains of three levels, applications, management, and infrastructure as indicated in Figure 3.1. Controllers exist in numerous forms in all SDN setups, in addition to (southbound APIs) and (northbound APIs):

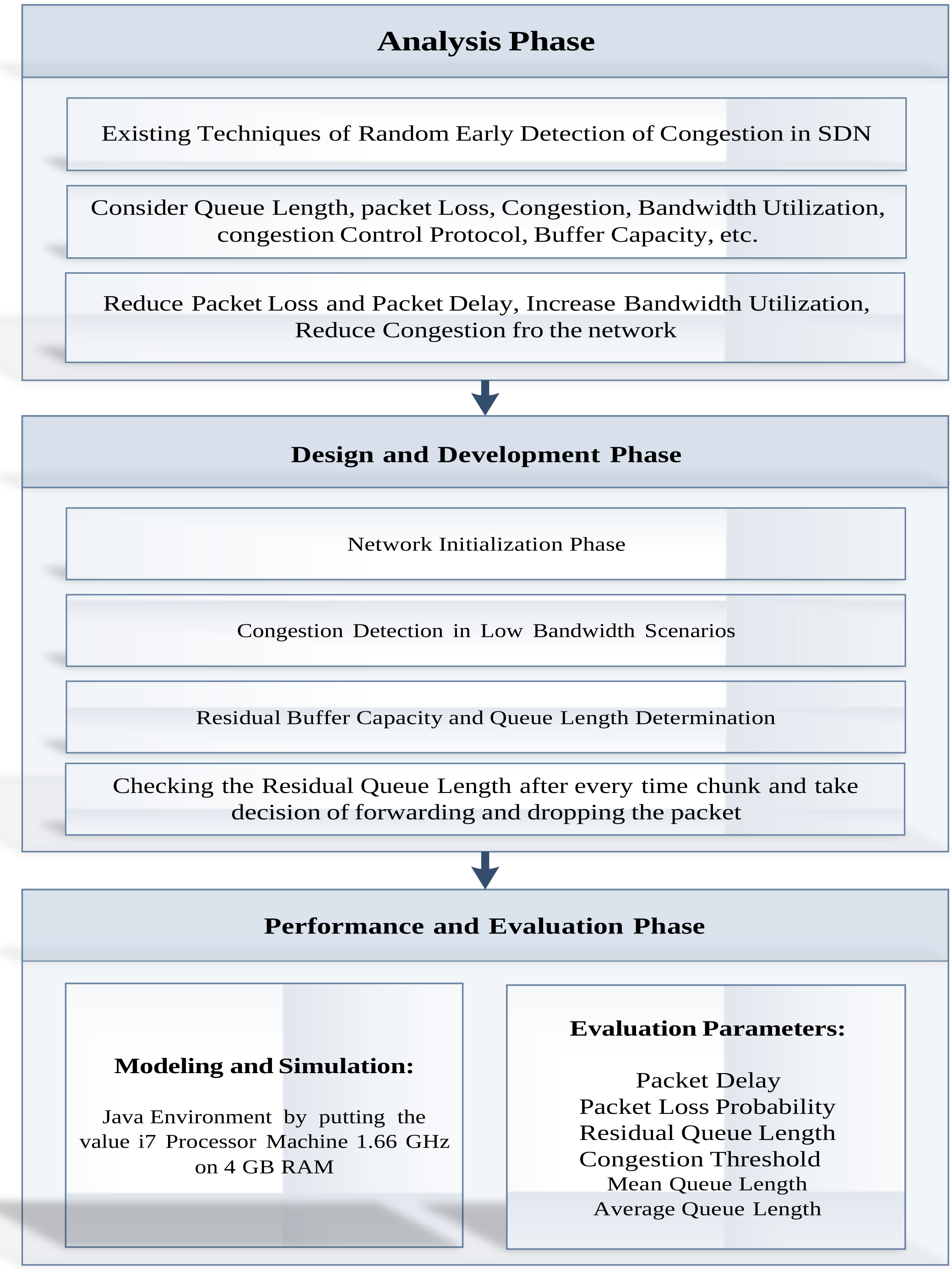


Figure 3.1: Software Defined Network (SDN) Framework

The first phase focuses on controllers, the "brains" of the network. Controllers, which give a centralized view of the whole network, enable network administrators to direct the underlying devices (such as switches and routers on how they should handle network traffic on the forwarding plane. In the second phase: congestion is detected in SDN to identify the problem that how can transmit data to the switches and routers "below." The first southbound API was Open Flow, which is still one of the most widely used protocols and is regarded as the first standard in SDN. Despite the fact that some people confuse Open Flow and SDN as being the same thing, Open Flow is only a small part of the larger SDN ecosystem. In the third phase: performance and evaluation is done by using Software Defined Networks to interface with applications and businesses. These aid network managers in automating service deployment and traffic shaping.

3.3 Research Design and Development:

Congestion control is a technique and mechanisms which can be classified as congestion detection and congestion avoidance Congestion recovery is often referred to as closed-loop congestion control, whereas congestion avoidance is also known as open-loop congestion control. Congestion control's goal is to maximize usage, reduce queue size, and reduce packet drops while ensuring that all sources have equal access. End system congestion management and network-centric congestion control are the two basic methods for managing congestion. Senders ultimately regulate system congestion by identifying the congestion and responding appropriately. TCP is a crucial illustration of the end-system approach. When a packet is dropped, the sender assumes there is congestion and slows down transmission. Senders speed up when a packet is successfully transferred.

Network-centric congestion control is the alternative strategy. The rationale behind this is that since routers have more knowledge about the condition of the network, they can help identify congestion and ought to be included in decisions regarding how to mitigate it. Routers compare input traffic and search the queue length to determine the level of congestion; as a result, as soon as people notice that the queue duration is lengthening, they can provide feedback. The average queue length does not have to be as long as it was in the previous way as a result. Additionally, routers can be utilized to priorities some sources over others. Active queue management is one of the most significant instances of network-centric congestion control.

3.3.1 Performance Parameters:

The major network parameters used for the performance are throughput, link utilization, mean queue length, packet loss is defined as follows:

3.3.2 Throughput:

It is specifying the ratio of the total packets arrived to the destination or it is the rate at which the packets are sent by a network source per unit time. It is calculated by the formula given in Equation 3.1.

$$Throughput = \frac{Total\ number\ of\ packets}{Time\ Duration} \tag{3.1}$$

3.3.3 Link Utilization:

It is defined as the link capacity being used for transferring packet. The current capacity of the network that is in use is termed as Link utilization. It is calculated by the formula given in Equation 3.2.

$$Link\ Utilization = \frac{Flow\ on\ the\ link}{Capacity\ of\ the\ link} \tag{3.2}$$

3.3.4 Mean Queue Length:

In a communication network, a queuing system is one where packets start waiting for services, wait for those services, and then, if they have waited for them, leave the system once they have been provided. As a result, the mean queue length is a crucial factor in assessing the effectiveness of active queue management in reducing congestion. The amount of time an input waits in the buffer might vary significantly depending on inter-arrival durations between inputs and the processor's processing speed. As a consequence, the size of the contribution queue can vary at random.

3.3.5 Packet Loss Probability:

It is determined by taking the total amount of serious decline packets by the total amount of input jam packets during a predetermined time period. Because the packet size is constant, the likelihood of packet loss may be computed using Equation 3.3.

$$Packet\ Loss\ Probability = \frac{Total\ amount\ of\ dropped\ packets}{Total\ Amount\ of\ Input\ Packets} \quad (3.3)$$

3.3.6 End-to-End or Latency:

It is the amount of time between the arrival of input packets in the communication queue and the actual information transmission packet. The demand on the communication bandwidth affects the latency.

3.4 Simulation Framework:

The appropriate parameterization of the AGRED over RED algorithm's variables is indul fundamental for increasing throughput, latency, connection usage, packet loss rate, and system fairness. For AGRED to be effective, adjusting the pace of congestion notification is one way to reduce network congestion and prevent the connections from being unused. Three criteria and two decision-dropping scenarios are included in AGRED. The router buffer quickly increased in size during periods of intense congestion and made the decision to discard packets once the all reached the double MA level. When the buffer is full, every packet will be dropped. However, the IAGRED algorithm

3.4.1 Performance Metrics:

In our simulations, the two key metrics are throughput and queue size. Each flow's throughput is used to show how fairly different flows are treated, and the overall throughput can be used to gauge how well resources are being used by comparing it to the bottleneck bandwidth as in Table 3.2. Queue size is a clear sign of how well a router is using its resources. The fairness of router resource allocation is demonstrated by the average queue size for each flow, which also demonstrates the distinctive features of various algorithms. We use an Exponentially Weighted Moving Average (EWMA) to calculate the average queue size, and the ageing weight is set to 0.002.

Table 3.2: Fixed Parameters used for simulation

Sr. No	Parameter Name	Value
1	Environment size	1000mX1500m
2	DoD value	3
3	Mobility Model	Random Waypoint
4	Traffic type	Constant Bit Rate
5	Packet size	512 Bytes
6	Packet Rate	10 Packets
7	MAC-Protocol	802.11
8	Radio Transmission Power	15Db
9	Simulation Time	100sec

The dimensions of the field selected for node mobility are 1000 m by 1500 m. Our analysis relied on continuous bit rate (CBR) traffic. Each data packet's size is preserved at 512 bytes, and the packet production rate is capped at 10 per second. The Random Waypoint Model is employed to determine how nodes move.

3.5 Summary:

In this chapter, we learn about issue at hand and examine several potential solutions. Operational frame work has considered that leads to proposed work. It explores the sampling design for the propose scheme. Its many components and operational factors are examined. From the results, it is clear by state that RED when compared to GRED and AGRED, can lead to an excessively high packet loss ratio for tiny packets. This unfairness results from RED's weighting of the dropping probability by packet size While AGRED produces evenly distributed decline and good results in terms of loss and equity discrimination. It is advised to employ AGRED, which has improved the PLR, as the network traffic is a mixture of different packet sizes.

CHAPTER 4

PROPOSED SOLUTION

4.1 Overview:

This chapter explores the proposed solution to resolve the main problem identified from the base schemes. We initially present the proposed algorithm to improve upon the in-efficiencies of the currently existing protocols, which are discussed in chapter 1 and chapter 2 respectively, for controlling the congestion in High speed Networks. First, we propose Congestion Detection Module. Once this module detects that congestion has set in (through the subsequent packet drops), the proposed "AQM" algorithm which is executing on the router / switch level initiates the measures for controlling the congestion.

4.2 Adaptive GRED Scheme

The random early detection algorithm is becoming a de-factor standard for congestion avoidance in the internet and other packet switched networks. Because of the incremental deployment of RED, numerous algorithms are based totally on RED and still being proposed to improve its performance. The Adaptive Gentle Random Early Detection (AGRED) algorithm drops the arriving packet with little more fairness than Random Early Detection and Gentle Random Early Detection (GRED). It shows marginal improvement by varying the unit value from Dmaxto. The Adaptive Gentle Random early detection algorithm has some limitations;

- ✓ Congestion control depends on the value of average queue length.

- ✓ All values are based on the queue weight, and the queue weight is fixed.

In this modern world, swift communication technologies are needed. The number of internet users increasing rapidly. Latest communication technologies are indented to develop for transmission of huge amounts of information. To maintain a network active and tolerable for its users, a congestion management strategy is required.

Network-centric congestion control is the alternative strategy. The rationale behind this is that since routers have more knowledge about the condition of the network, they can help identify congestion and ought to be included in decisions regarding how to mitigate it. Routers compare input traffic and search the queue length to determine the level of congestion; as a result, they can give feedback as soon as they see that the queue length is growing. As a result, there is no need for the average queue length to be as long as in the previous method. Routers may also be used to give some sources precedence over others. Active Queue Management is a key illustration of network-centric congestion control.

Congestion management centered on the network is another option. The reasoning for this is that routers should be included in the selection of congestion control since they know more about the condition of the network and can help with detection of congestion. By analyzing incoming traffic and conducting searches on the queue length, routers are able to gauge the amount of congestion and provide information as soon as they detect a growing backlog. As a result, it is not necessary to have such a long line on average as was the case before. It is also possible to priorities certain sources over others using a router. An important example of congestion control is Active Queue Management.

As a result of AQM's congestion control and packet delivery policy, the node's processing priorities for incoming packets are decided. Congestion is estimated using a number of metrics based on the functioning of the network. The downside of a queue-based approach is that it requires accumulation. The packets' arrival rate is used to infer the link's use, which is used to dictate congestion and limit the amount of effort expended.

4.3 Improved AGRED Scheme:

In the internet and other packet switched networks, the random early detection technique is quickly becoming the de facto norm for congestion avoidance. Numerous completely RED-based algorithms were presented and are still being developed to enhance the performance of RED as a result of its progressive implementation. When compared to Random Early Detection and Gentle Random Early Detection, the Adaptive Gentle Random Early Detection (AGRED) algorithm drops the arriving packet with slightly more fairness (GRED). It exhibits a slight improvement when the Unit value is changed from Dam to 1. There are some restrictions on the Adaptive Gentle Random early detection algorithm.

- Congestion control depends on the value of average queue length.
- The all value is based on the queue weight which was fixed.

The above-mentioned limitation can be rectified by the proposed algorithm, Improved Adaptive Gentle Random Early Detection (IAGRED) Algorithm by:

- Keep the queue weight as a dynamic value.
- Dynamically regulate the window size founded on the total value.
- The measurements of congestion of the proposed IAGRED algorithm are all and dynamic window size. The parameter settings are similar to the AGRED and RED. The parameters used for the congestion measure are discussed as follows:

4.3.1 Queue Management:

A queuing system in networks can be described as packets arriving for service, wait in for service if it is not immediate, and if having waited for service, leaving the system after being served. In most cases, four basic characteristics of queuing processes provide an adequate description of a queuing system in networks:

- Arrival pattern of packets
- Service pattern of schedulers

- Queue discipline
- System capacity

To model network behavior, analytical models require information about the properties of the queues.

4.3.1 Delay Management in Queue:

A packet's postponement is the amount of time it takes to go from one location to the another (such as a source location or network ingress) (e.g., destination premise or network degrees). The delay is important due because**:**

- ✓ If end-to-end delays between nodes are excessively high in comparison to a predetermined threshold value, some applications (such as audio and video applications) may not function well.
- ✓ Erratic latency fluctuation makes it challenging to accommodate a wide range of real-time applications.
- ✓ Maintaining high bandwidths for transport layer protocols becomes more challenging as delay values increase.
- ✓ When this metric has a low value, it indicates that the delay was mainly due to propagating and transmitting delay, as would be the case if the channel being travelled were not heavily utilized.
- ✓ Congestion on the path is indicated by values of this characteristic exceeding their minimal threshold. This attribute can be specified in a number of ways, including average latency, variability of delay, and pause bound.

Each packet produced by a source travels via a series of intermediary nodes before arriving at the destination. Thus, the total delay from beginning to end is the result of the delays at each node along the route. Each of these delays has two parts: a fixed part that accounts for the transmission delays at nodes and the propagation delays on links to subsequent nodes, and a variable part that accounts for the processing and queuing delays at each node. Packets released at the intermediary nodes as a result of a cushion excess are referred to as end-to-end loss.

4.4 Average Queue Length and Packet Drop Management:

To begin, estimate how long queues typically are by complete the formula [22]:

$$all \times (1 - q_i \times 2)n \qquad (4.1)$$

$$aql = aql \times (1 - qw \times 2) + qw \times 2 \times q_{instanteneous} \qquad (4.2)$$

Where, q_i is queue weight and $q_{instanteneous}$is instantaneous queue length. In the above two Equations 4.1 and 4.2, the value of qi is not fixed to the specific value equal to 0.002 as in the case of AGRED. The Equation 4.1 is used when the router buffer is empty and there is no dropping. But, in Equation 4.2, the value of qi plays an important role, in determining all that grows quickly to reach the max brink and double max beginning to decide on the dropping of the packet.

Congestion avoidance strategies screens network traffic loads to be able to expect and keep away from congestion at not unusual network bottlenecks. Congestion avoidance is accomplished through packet dropping. Among the most commonly used congestion avoid mechanisms, Random Early Detection (RED) algorithm is optimum for high-speed transit networks. The packet drop probability is based on the minimum threshold, maximum threshold and mark probability. When the average queue length is above the minimum threshold, RED starts dropping packets. The rate of packet drop increases linearly as the average queue size increases until the average queue size reaches the maximum threshold. The mark probability denominator is the fraction of packets dropped when the average queue length is at the maximum threshold. The formula for calculating packet dropping probability is given by probability to drop a packet related to minimum and maximum queue threshold value as in Equation 4.3 where P**b** is probability to drop a packet, min^{th} is minimum queue length threshold, max^{th} is maximum queue length, $threshold_{avg}$ is average queue size and maxp=upper bound on dropping probability [25].

$$P_b = (avg - min^{th})/(max^{th} - min^{th}) \qquad (4.3)$$

Probability to drop a packet related to packet size if the queue size is measured in bytes instead of packets as in Equation4.4 where packet size is arriving packet size in bytes and maximum packet size is maximum packet size allowed in bytes [25].

$$P_b = P_b \times \frac{Packet\ size}{Maximum\ Packet\ size} \qquad (4.4)$$

This algorithm has two separate parts. One is for computing the average queue size, which determines the degree of burstiness that will be allowed in the router queue. It takes into account the period when the queue is empty by estimating the number of small packets that could have been transmitted by the router during the idle period. After the idle period, the router computes the average queue size as if packets had arrived at an empty queue during that period. The alternative is used to calculate the packet-marking probability and determines how regularly the router marks packets, given the current stage of congestion. The goal is for the router to mark packets at fairly flippantly spaced periods, so that it will avoid biases and keep away from global synchronization and to mark packets sufficiently regularly to govern the average queue length. The following rules must be followed while setting parameters.

Rule 1: It is important to determine the typical length of a queue. Floyd's suggested value for the queue weight should be used.

Rule 2: Make sure the minimum threshold is high enough to prevent the typical queue length from being too small. If the average queue size is too small, the queue will get crowded before the output connection has had a chance to fully exploit its throughput potential.

Rule 3: The minimum standard and highest threshold should be large enough so that the likelihood of marking or having to drop packet data is low.

4.7 Minimum Threshold and Maximum Threshold Parameterization:

Consider the scenario where a source sends a packet and there occurs packet loss due to congestion. The packet is then resent by that source, but is once more lost. The process can continue with the network being completely loaded but no usable data being sent if the sender keeps resending the packet and the other network flows don't change then Congestion collapse will be developed. If sources recognize the beginning of congestion and utilize it as a clue to lower their transmitting rate, the congestion can be avoided and eased.

The number of packets that can be transmitted before being dropped should be large enough to account for the discrepancy between the month and math thresholds. Congestion would if there is a very small discrepancy between the month and the math then it would not be discovered in time. If we multiply the largest possible base queue delay by the available bandwidth, we get the number of months that have to wait. Throughput will suffer if monthly is set too small and latency will suffer if monthly is too large. To avoid global synchronization, calculations should be set at least twice a month. The difference between the month and the calculation will be significantly higher if the data transmission between networks is poor.

The proposed algorithm's formula for calculating packet dropping was revised and put to the test using a simulator to address the packet loss issue in AGRED. Through the use of a dynamic queue weight setting, this approach seeks to improve parameter settings. During the congestion, the IAGRED begins to drop the packets. IAGRED's congestion metric consists of both static and dynamic queue weight. The single router buffer for the active queue management algorithms AGRED and IAGRED is shown in Figure 4.1.

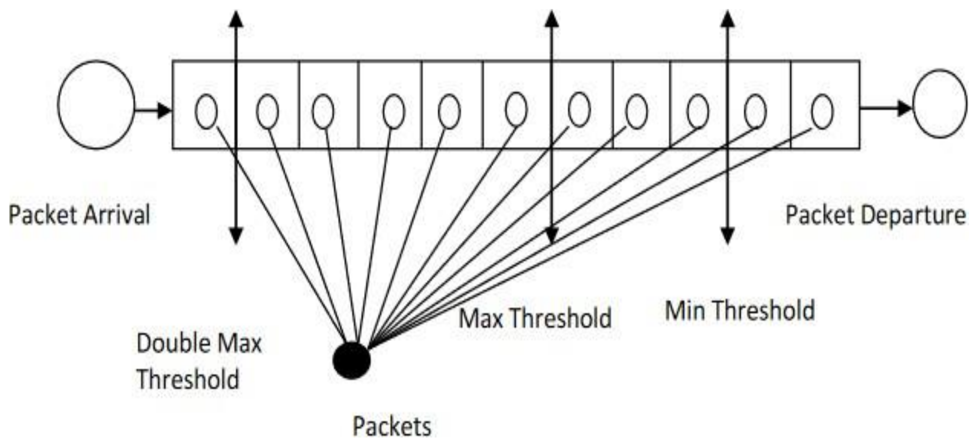


Figure 4.1 single router buffers for proposed IAGRED algorithm

The proposed algorithm is further improved by employing a dynamic value of qi to prevent the router buffer from overflowing by early-stage congestion control. The purpose of IAGRED algorithm is to increase the router buffer response to the dropping. In previous AQM method, qi value is set to 0.002. This is because qi, the queue weight, neither can be too high, because the average queue size can grow quickly; nor too low, then the average queue size will respond too slowly. Thus, the value of 0.002 was chosen and also was validated in AGRED method.

In AGRED, there are three thresholds and two decision dropping scenarios. When the heavy congestion occurs, the router buffer built its size quickly and decided to drop the packets when the all reach the double max threshold. Every packet will be dropped when the buffer becomes full. But in IAGRED algorithm, the value of qi is not fixed to the specific value equal to 0.002. As a result, the proposed IAGRED method keeps the buffer small as long as possible to prevent the router buffer overflow quickly. All calculations should use parameters that are not overly sensitive to changes in bandwidth. The following guidelines are offered to provide the best possible performance of the network.

According to the active queue management algorithm, the IAGRED algorithm deals with the congestion control according to the following scenarios:

- When all of the packets in the router buffer are below the minimum threshold, no packets are dropped.
- Random packets are deleted when all packets are within the minimum and maximum threshold.
- Increased probabilities of dropping random packets are in effect when all packets fall between the highest and double maximum thresholds.
- The likelihood of packets being dropped at random is increased when everything is between the maximum threshold and the double extreme beginning.

Table 4.1 and algorithm 4.1, respectively parameters for the proposed IAGRED method.

Table 4.1: A description of parameters used in IAGRED algorithm

Notation	**Description**
current _ time	The current time. This time is implemented by using a constant time unit called slot that is used in Discrete-time Queues technique (Woodward, 1993).
idle _ time	The beginning idle time at the router buffer.
N	The number of packets transmitted to the router buffer through an idle interval time
C	The counter that represents the number of packets arrived at the router buffer and have not dropped since the last packet was dropped.
D_{final}	The final packet dropping probability.
D_{init}	The initial packet dropping probability.
q_instantaneous	The instantaneous queue length.
D_{max}	The maximum value of D_{init}.
Qw	The queue weight.
q(time)	The linear function for the time.
K	The finite system capacity (set to 20 packets).

Algorithm 4.1: Improved AGRED Algorithm

Initialize C = −1 and *aql* = 0.0
For each arriving packet at router’s buffer **do**
  Calculate *aql* for the arriving packets.
  If (the queue at the router buffer = = empty) // Examine the queue status.
    Compute n, where n = q($current\ \_\ time - idle\ \_\ time$);
    $aql = aql \times (1 - qw)^n$
  Else
    $aql = aql \times (1 - qw \times 2) + qw \times 2 \times q_{instanteneous}$
  End if
End for
If (*aql* > 0% AND *aql* < 40%) // Determine the congestion status at the router buffer.
  Set D_{final} to 0.0// No packets have dropped

```
    Set C = −1
Else if (aql > 40% AND aql < 70%)
    Set C = C +1
    Calculate D_final value for the arriving packets as:
```

$$Dinit = \frac{aql \times (1 - \mathrm{Dmax})}{\mathrm{K}}, Dfinal = \frac{Dinit}{(1 - C \times \mathrm{Dinit})}$$

```
    Drop the arriving packet in terms of its D_final value
    Set C = 0
Else if (aql > 70% AND aql < 90%)
    Set C = C +1
    Calculate D_final for the arrival packets as:
```

$$Dinit = \frac{aql}{\mathrm{K}},\ Dfinal = \frac{Dinit}{(1 - C \times \mathrm{Dinit})}$$

```
    Mark/drop arriving packet in terms of its D_final value
    Set C = 0
Else // if (aql ≥ 80%)
    Mark/drop every arriving packet
    Set C = 0
End if
If the GRED router buffer becomes empty
    Set idle _ time = current _ time
End if
```

Thus, after calculating the average queue length and comparing it to two minimum and maximum thresholds, IAGRED decides whether to delete a packet. If the mean waiting line is less than the threshold, less packets dropped and there is no congestion. If the mean queue length is more than the limit, severe congestion will ensue, and all incoming traffic will be discarded with Dip = 1. To boot, the recommended method requires figuring out the Dip whether the value is typical in order to decide whether or not a packet should be included or left out. When compared to AGRED, the IAGRED method performs well in terms of packet loss and throughput by employing a double maximum threshold. Figure 4.3 shows the flowchart for the aforementioned algorithm.

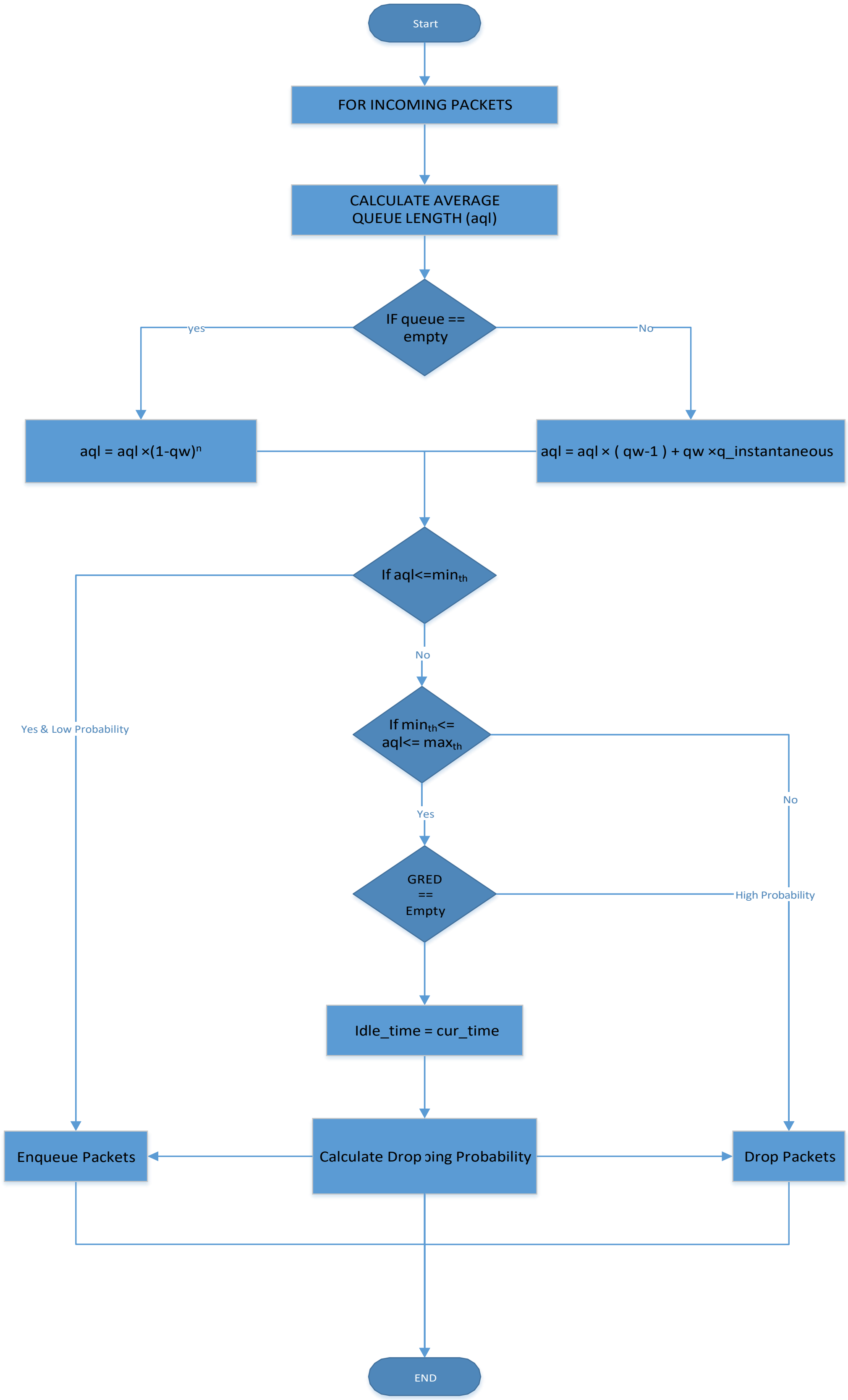


Figure 4.3: Estimating the typical waiting time in a queue and the likelihood of packet loss

4.8 Summary:

According to the findings, the Improved Adaptive Gentle Random Early Detection in the algorithm can identify and prevent cramming both before and during data flow between network nodes. As the first step towards realizing a fully autonomous network, this work leverages SDN and machine learning to create a congestion detection classifier. Due to the fast computation time and high resistance to noisy data, the decision tree algorithm was chosen for this task. Feature selection is the most important step in building the classifier. One-way delay and time between arrivals are chosen as the congestion detection feature. A total of 63 attributes using mean, min, max and standard deviation of one-way delay and time between packets were used. This includes the ratios between the mean, minimum, and maximum of a feature in all three windows, including the ratio between the maximum in one window and the minimum in another. The simulation results demonstrate that the suggested solutions outperform the other strategies covered in this division.

CHAPTER 5

PERFORMANCE EVALUATION

5.1 Overview:

The underlying network and flow information in conventional networks is unclear. The congestion window needs to be changed in order to address the traditional network's level of congestion. But the congestion issue can be efficiently resolved with the use of a software defined network. Software-defined congestion control is therefore suggested.

In order to allow for the gradual introduction of new technology designed to solve brand-new problems into the network at a reasonable cost, research on congestion control in computer networks has focused on a solution that enables the use of networks with more programming resources and less need for replacing hardware elements.

The introduction of programmable networks has made it possible to handle congestion effectively and has improved traffic flow in switching devices. As seen in Figure 5.1, the SDN's centralized management allows for total network visibility.

Decoupling a network's control and data planes is part of SDN. It is founded on the fact that a switch's sole function is to forward packets in accordance with rigid regulations. However, software is used to control the instructions that come with the switch to forward packets. SDN's goal of keeping network device layout simple is one of its driving forces. Another advantage is the ability to execute communication networks that would otherwise be impossible without the installation of supplementary software on each of the discussed in section. Network operating systems provide a bridge between switches and the applications that operate them.

5.2 Experimental Analysis:

Analyzing the transmission process necessitates prospective study on the variable of the queuing weight, which is in turn determined by variable packet size. The primary objective of this research is to examine the effects of dynamic packet size on performance, loss ratio of packets and queuing latency in a network designed to minimize congestion. Simulations are performed using Java Environment by putting the value i7 Processor Machine 1.66 GHz on 4 GB RAM.

The queuing discipline, in all the simulations, is First Come First Serve (FCFS). The probabilities of packet arrival and departures at the node determine the band width utilized. The packets are sent to other outer buffer and depart from the router buffer individually, i.e., packet by packet. Three groups of TCP resources or destinations share an equal network channel made up of a bottleneck connection moving at 30 ambits per second between two routers in the simulation scenario as discussed in Table 5.1.

Table 5.1: Parameters for IAGRED algorithm

Parameters	**Value**
Queue size	16
Simulation end time (ms)	1,500,000
Mean inter arrival time (ms)	0.18,0.33,0.48,0.63,0.78,0.93
Mean service time (ms)	0.6
Min threshold	4
Max threshold	10
Queue weight	Dynamic

The suggested Improved AGRED approach is evaluated against GRED and AGRED to determine its efficacy. We do five independent random-number runs, with unique inputs for each. As we compared the proposed scheme i.e. IAGRED with GRED & IGRED; and these schemes used queue size 16 so therefore we use queue size 16 in our parameters. And we can achieve the better results for our research by adjusting the queue size 16.

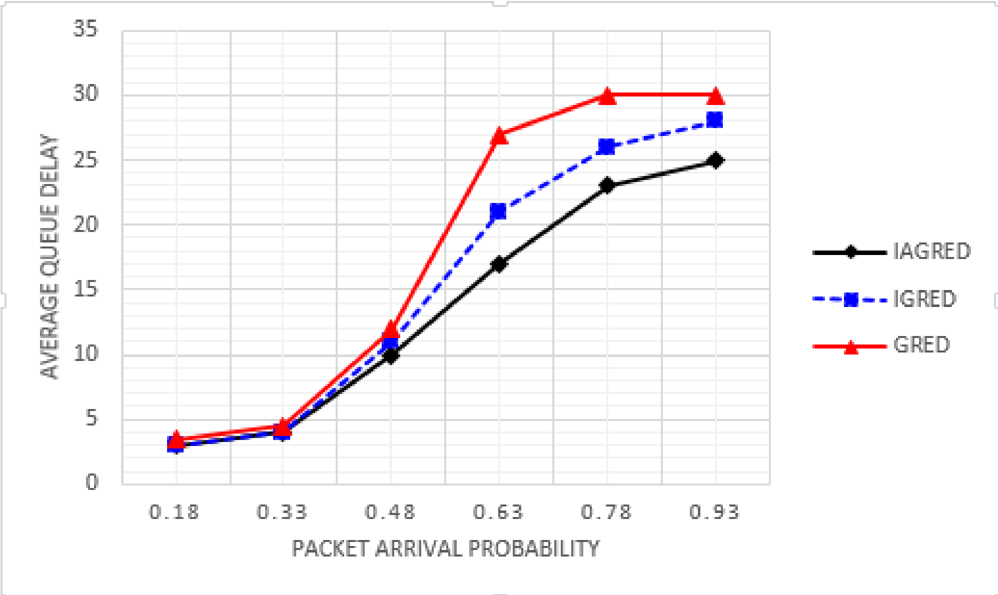
All three variants of GREEN begin with the same initial settings. Compression and non-congestion were both initiated by the arrival probability of packets reaching the buffer. Congestion was detected using a buffer size of 30 packets. There were 1.5 million minutes of airtime taken. With this number, we can incorporate a reliable performance gauge and have time to properly heat up. As soon as the system stabilizes, the warm-up phase is over. RED includes the practice the minimum threshold, maximum threshold, and maximum dynamic range (Dam) to 4, 10, and 0.1, respectively. At last, we've decided to follow GRED's lead and set the double-max barrier at 20.

As a result, the technique is further enhanced by using a variable value of qi to regulate the overcrowding in the routers buffers before it overflows. “Improved Adaptive Gentle Random Early Detection” is a technique designed to speed up the buffer in reaction to a drop. When there is significant congestion, the findings of average queue, average queuing time, and chance of packet loss outcomes that matter.

5.3 Results and Discussion:

The network path used in the simulation version is shared by three groups of TCP of sources and destinations across a bottleneck link of 30 ambits/s between the connected routers. The maximum Transmission Unit (MTU) for each group is, respectively, 1650, 1100, and 500 bytes. Each group consists of 20 TCP sources and destinations aiding selective acknowledgements. The 200ms timeout granularity was chosen. For the bottleneck linkages of 15ms and 80ms, two units of simulations with modest and large propagation delay values have been accomplished. The potential Loss Ratio for each MTU price is reported by simulation results; in other words, the quantity of dropped packets for each MTU is measured.

The Figure 5.1 & 5.2 show the better performance of the Improved AGRED algorithm compared to other active queue management algorithms such as IGRED and GRED. It could be noted in Figures that GRED, IGRED and the proposed IAGRED offer similar mql and D results when there is no congestion occurred at their router buffers. When congestion is occurred, the performance measure results of the proposed IAGRED with reference to mql and D are better than those of GRED & IGRED. But with heavy congestion such as α = 93.0, the mql and D results of IAGRED become slightly better than those results of IGRED & GRED.



‘ **Figure 5.1** Probability of Average Queue Delay

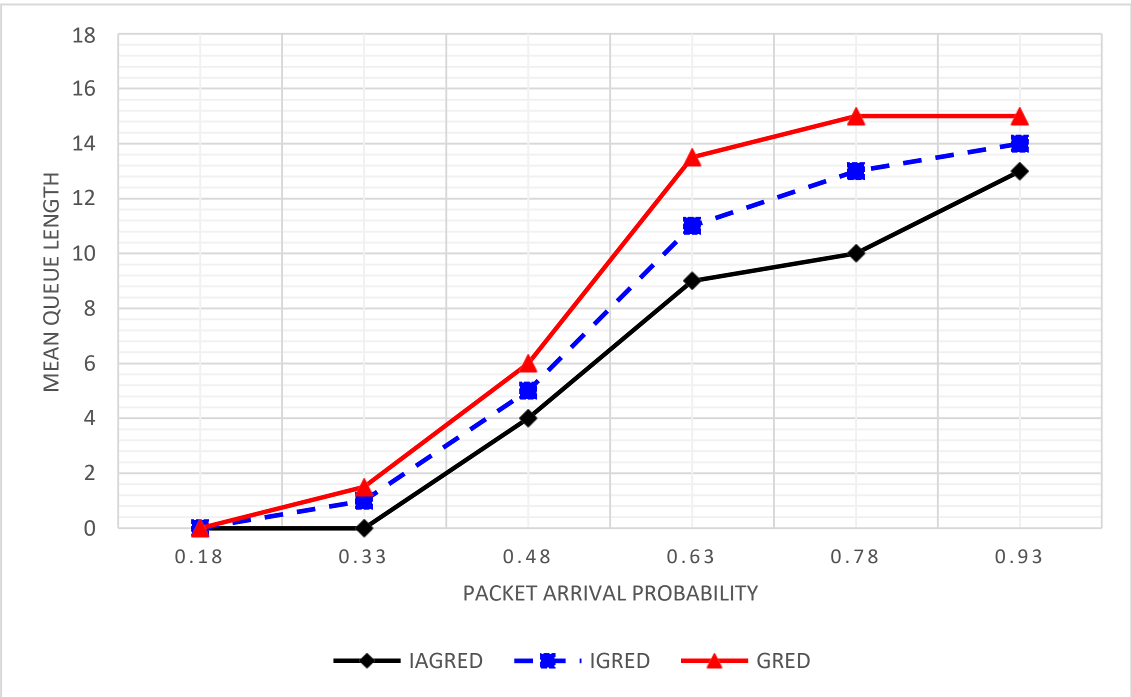


Figure 5.2: Probability of Mean Queue Length

5.3.1 Comparison of Link Utilization:

The hosts and switches that will be used to define the experiment must be specified. The optimum path to connect all hosts must be determined by the Open Flow controller. A substantial number of Open Flow messages are sent in order to test the performance of the proposed method and compare network performance before and after its implementation. With and without congestion detection, the graph of average RTT and throughput was displayed for various numbers of switches.

Figure 5.3 illustrates that GRED, AGRED and the proposed IAGRED do not lose arrival packets due to overflow when α value is less than 0.48. In other words, the router buffers of GRED, IGRED, and IAGRED will not overflow when α value is less than or equal to 0.48. When congestion is taken place, the number of times that the GRED's & IGRED's router became overflow is more than that of IAGRED.

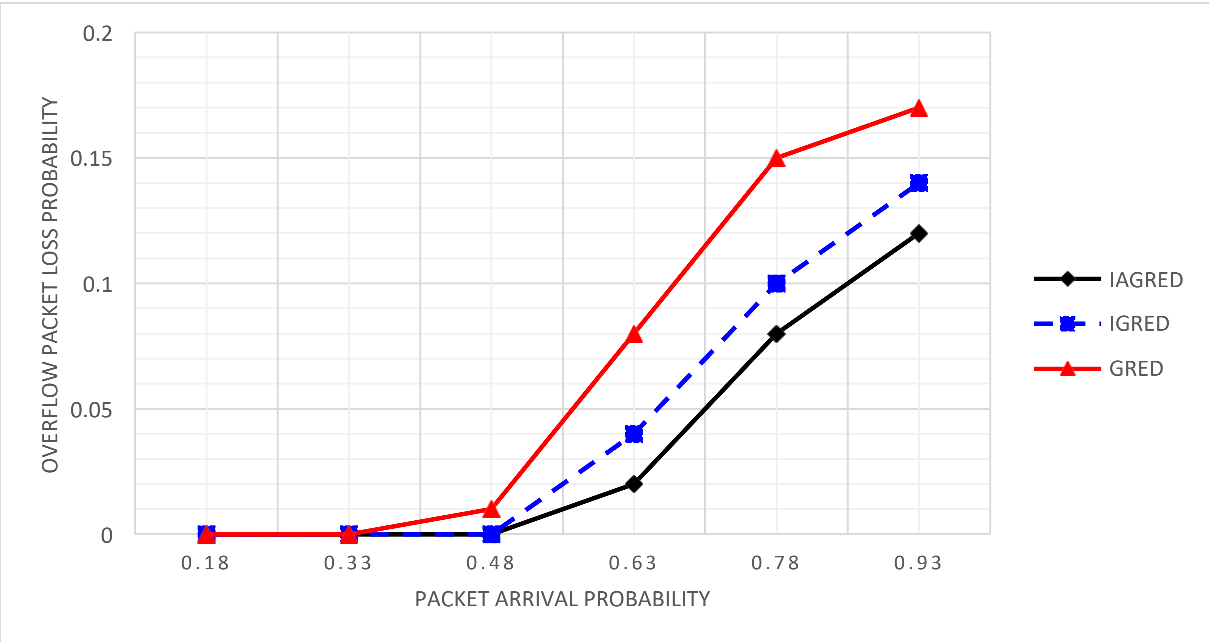


Figure 5.3 Overflow Packet Loss Probability

It is obvious in Fig. 5.4 that GRED & IGRED router buffer before it becomes full, it drops a number of packets more than that of IAGRED when α value is greater than or equal to 0.48. The reason of this is when congestion is occurred, GRED & IGRED's router buffer will overflow more than the router buffer of IAGRED, this means the number of packets will be dropped by GRED, IGRED before the router buffer is full is more than that of IAGRED, and this explains the better Dp performance of IAGRED.

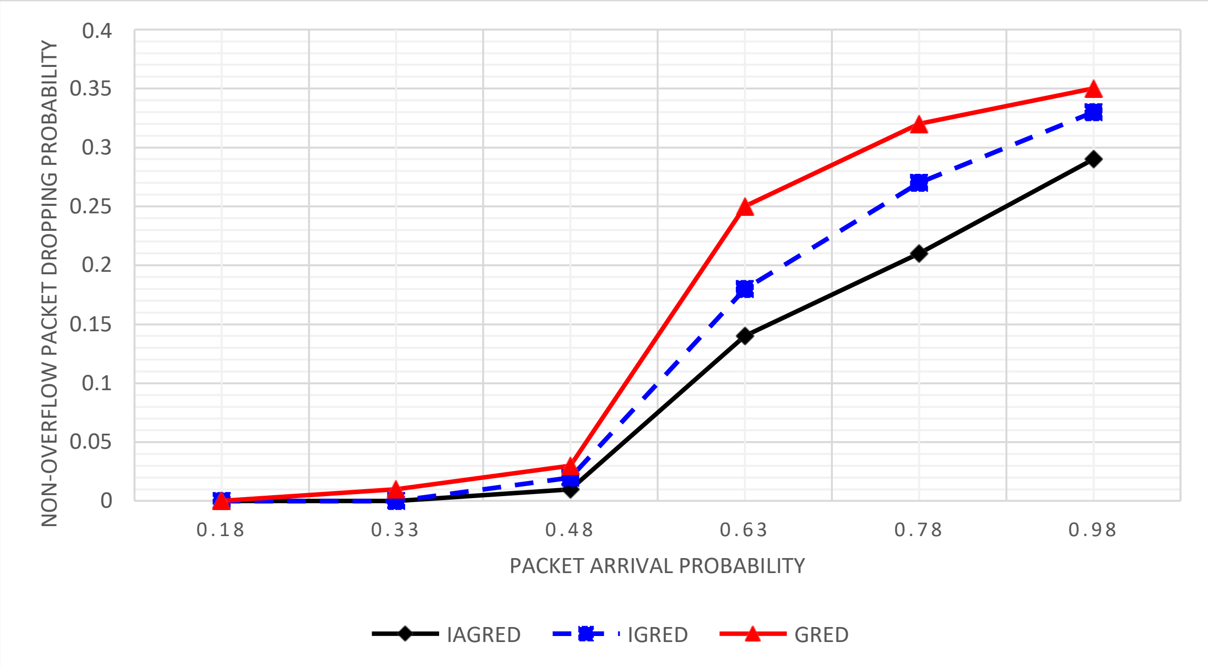


Figure 5.4: Non-overflow packet Dropping Probability

Figure 5.5 shows that the Throughput results of GRED, IGRED and the proposed IAGRED is almost similar but a slightly higher whether there is congestion or not.

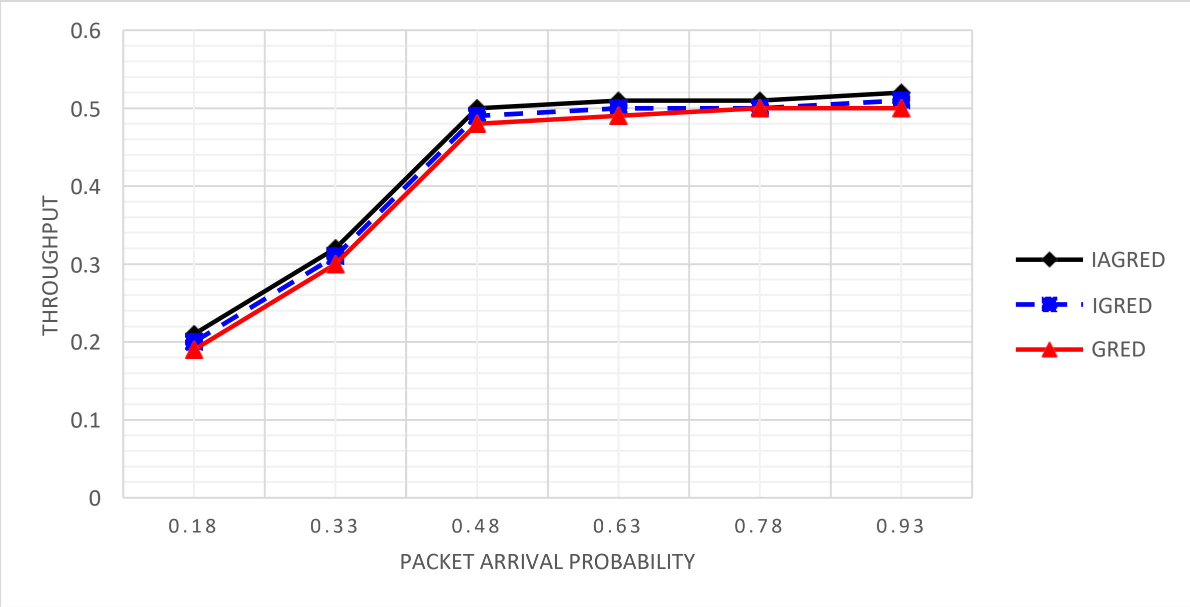


Figure 5.5: Comparison of Throughput

5.4 Summary:

Data traffic volume is continuously growing, as a result of which the information exchanges between user's growths. This consequence in growth is demand of resources and quality of services. This leads to application network congestion, resulting in higher round trip time (RTT) and packet loss. The amount of data that is successfully transmitted over the network is referred to as throughput. As the amount of data increases, the throughput should also increase. From the simulation results, it is clear that the planned algorithm in SDN outperforms regarding throughput and RTT. From this, it is concluded that the congestion in the network should be detected and avoided to increase the performance of the network.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1. Overview:

The present network topology has been drastically altered by the fast developing field of software-defined networks. Network Congestion Control has always been considered as an important issue after the emergence of packet switched network technology. This work has made an effort to deal with the issue of congestion control in high speed networks. We consider the network congestion as a distributed problem thus requires solution which is also of distributed nature. The source based congestion control and router based congestion control using AQM solve global problem of network congestion.

6.2. Conclusion:

In this work, the source based and router based congestion control approach is presented where the residual queue size is utilized to enhance the bandwidth utilization. The performance analysis of proposed AQM approach has been shown in presence of different counterparts. The proposed solution attempts to timely mange the congestion by deciding ad per the percentage of the queue filled. Simulations of the proposed schemes are performed using Java Environment by putting the value i7 Processor Machine 1.66 GHz on 4 GB RAM. As IAGRED works with the average queue length and queue weight, therefore, it shows better results than existing schemes in terms of packet loss and mean queue length. Proposed scheme uses threshold congestion chunks after short interval of time to detect the congestions on network. Due to which congestion is detected on early stage and decision is taken for further packets forwarding or dropping. Therefore due to Early Detection of congestion Delay time is decreased in IAGRED as compared to GRED and IGRED; which is clear from the graphs given in results chapter.

6.3. Future Work:

Further studies can be performed on understanding and utilizing the parameters of Random Early Detection algorithm to leverage the performance of the active queue management in the real-time network for emergency packet in Non-delay tolerant networks. This research study can be extended by comparing and evaluating the other mixes of traffic pattern in the Internet such as long- and short-lived flows.

REFERENCES

1. Hassan, S.O., Rufai, A.U., Agbaje, M.O., Enem, T.A., Ogundele, L.A., and Usman, S.A. (2022). Improved random early detection congestion control algorithm for internet routers. Indonesian Journal of Electrical Engineering and Computer Science, Vol. 28, No. 1, October 2022, pp. 384~395.
2. Naeem, F., Srivastava, G. and Tariq, M., 2020. A software defined network based fuzzy normalized neural adaptive multipath congestion control for the internet of things. IEEE transactions on network science and engineering, 7(4), pp.2155-2164.
3. Goto, Y., Ng, B., Seah, W.K. and Takahashi, Y., 2019. Queueing analysis of software defined network with realistic openflow–based switch model. Computer Networks, 164, p.106892.
4. Chakravarthy, V.D. and Amutha, B., 2020. Software-defined network assisted packet scheduling method for load balancing in mobile user concentrated cloud. Computer Communications, 150, pp.144-149.
5. Hertiana, S.N., Kurniawan, A. and Pasaribu, U.S., 2018. Effective Router Assisted Congestion Control for SDN. International Journal of Electrical & Computer Engineering (2088-8708), 8(6).
6. Ahmed, O., Ren, F., Hawbani, A. and Al-Sharabi, Y., 2020. Energy optimized congestion control-based temperature aware routing algorithm for software defined wireless body area networks. IEEE Access, 8, pp.41085-41099.
7. Bagewadi, A. and Babu, R.M., 2014. Towards an Ethernet Learning Switch and Bandwidth Optimization using POX Controller. International Journal of Advanced Research in Computer and Communication Engineering, 3(7), pp.7531-7535.
8. Abdel-jaber, H., Shehab, A., Barakat, M., & RASHAD, M. (2019). IGRED: An improved gentle random early detection method for management of congested networks. Journal of Interconnection Networks, 19(02), 1950004.
9. Letswamotse, B.B., 2018. Software defined networking based resource management and quality of service support in wireless sensor network applications (Doctoral dissertation, University of Pretoria).

10. Li, P., Guo, S., Pan, C., Yang, L., Liu, G. and Zeng, Y., 2019. Fast congestion-free consistent flow forwarding rules update in software defined networking. Future Generation Computer Systems, 97, pp.743-754.
11. Abdel-Jaber, H., Alkhateeb, J. H., & El-Amir, M. (2022, December). Evaluation of the Performance for IM-RED and IGRED Algorithms using Discrete-time Queues. In 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN) (pp. 23-28). IEEE.
12. Bagewadi, A. and Babu, R.M., 2014. Towards an Ethernet Learning Switch and Bandwidth Optimization using POX Controller. International Journal of Advanced Research in Computer and Communication Engineering, 3(7), pp.7531-7535.
13. Naeem, F., Srivastava, G. and Tariq, M., 2020. A software defined network based fuzzy normalized neural adaptive multipath congestion control for the internet of things. IEEE transactions on network science and engineering, 7(4), pp.2155-2164.
14. Goto, Y., Ng, B., Seah, W.K. and Takahashi, Y., 2019. Queueing analysis of software defined network with realistic openflow–based switch model. Computer Networks, 164, p.106892.
15. Chakravarthy, V.D. and Amutha, B., 2020. Software-defined network assisted packet scheduling method for load balancing in mobile user concentrated cloud. Computer Communications, 150, pp.144-149.
16. Hertiana, S.N., Kurniawan, A. and Pasaribu, U.S., 2018. Effective Router Assisted Congestion Control for SDN. International Journal of Electrical & Computer Engineering (2088-8708), 8(6).
17. Ahmed, O., Ren, F., Hawbani, A. and Al-Sharabi, Y., 2020. Energy optimized congestion control-based temperature aware routing algorithm for software defined wireless body area networks. IEEE Access, 8, pp.41085-41099.
18. Kaur, K., Garg, S., Aujla, G.S., Kumar, N., Rodrigues, J.J. and Guizani, M., 2018. Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay. IEEE communications magazine, 56(2), pp.44-51.
19. Abdelmoniem, A.M., Bensaou, B. and Abu, A.J., 2017, May. SICC: SDN-based incast congestion control for data centers. In 2017 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

20. Letswamotse, B.B., 2018. Software defined networking based resource management and quality of service support in wireless sensor network applications (Doctoral dissertation, University of Pretoria).
21. Li, P., Guo, S., Pan, C., Yang, L., Liu, G. and Zeng, Y., 2019. Fast congestion-free consistent flow forwarding rules update in software defined networking. Future Generation Computer Systems, 97, pp.743-754.
22. S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” IEEE/ACM Transactions on Networking, vol. 1, no. 4, pp. 397–413, 1993, doi: 10.1109/90.251892.Ghaffari, A., 2015. Congestion control mechanisms in wireless sensor networks: A survey. Journal of network and computer applications, 52, pp.101-115.
23. Feldmann, A., Chandrasekaran, B., Fathalli, S. and Weyulu, E.N., 2019, December. P4-enabled network-assisted congestion feedback: A case for nacks. In Proceedings of the 2019 Workshop on Buffer Sizing (pp. 1-7).
24. Su, K., Ramakrishnan, K.K. and Raychaudhuri, D., 2016, June. Scalable, network-assisted congestion control for the MobilityFirst future internet architecture. In 2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN) (pp. 1-2). IEEE.
25. AL-AWFI, K. S., & WOODWARD, M. E. DESIGN OF ACTIVE QUEUE MANAGEMENT BASED ON THE CORRELATIONS IN INTERNET TRAFFIC.
26. Pu, J. and Hamdi, M., 2008. Enhancements on router-assisted congestion control for wireless networks. IEEE transactions on wireless communications, 7(6), pp.2253-2260.
27. Manikandan, K. and Saleem Durai, M.A., 2014. Active queue management based congestion control protocol for wireless networks. International Journal of Enterprise Network Management, 6(1), pp.30-41.
28. Ndikumana, A., Ullah, S., Kamal, R., Thar, K., Kang, H.S., Moon, S.I. and Hong, C.S., 2015, August. Network-assisted congestion control for information centric networking. In 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS) (pp. 464-467). IEEE.
29. Abdelmoniem, A.M. and Bensaou, B., 2015, December. Efficient switch-assisted congestion control for data centers: an implementation and evaluation. In Proc. of Proceedings IEEE International Performance Computing and Communications Conference (IPCCC) (pp. 1-8).

30. Zhu, X. and Pan, R., 2013, December. NADA: A unified congestion control scheme for low-latency interactive video. In 2013 20th International Packet Video Workshop (pp. 1-8). IEEE.
31. Lall, S., Alfa, A.S. and Maharaj, B.T., 2016, July. The role of queueing theory in the design and analysis of wireless sensor networks: An insight. In 2016 IEEE 14th International Conference on Industrial Informatics (INDIN) (pp. 1191-1194). IEEE.
32. Aghdam, S.M., Khansari, M., Rabiee, H.R. and Salehi, M., 2014. WCCP: A congestion control protocol for wireless multimedia communication in sensor networks. Ad Hoc Networks, 13, pp.516-534.
33. Ghaffari, A., 2015. Congestion control mechanisms in wireless sensor networks: A survey. Journal of network and computer applications, 52, pp.101-115.
34. Feldmann, A., Chandrasekaran, B., Fathalli, S. and Weyulu, E.N., 2019, December. P4-enabled network-assisted congestion feedback: A case for nacks. In Proceedings of the 2019 Workshop on Buffer Sizing (pp. 1-7).
35. Su, K., Ramakrishnan, K.K. and Raychaudhuri, D., 2016, June. Scalable, network-assisted congestion control for the MobilityFirst future internet architecture. In 2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN) (pp. 1-2). IEEE.
36. Pu, J. and Hamdi, M., 2008. Enhancements on router-assisted congestion control for wireless networks. IEEE transactions on wireless communications, 7(6), pp.2253-2260.
37. Manikandan, K. and Saleem Durai, M.A., 2014. Active queue management based congestion control protocol for wireless networks. International Journal of Enterprise Network Management, 6(1), pp.30-41.
38. Ndikumana, A., Ullah, S., Kamal, R., Thar, K., Kang, H.S., Moon, S.I. and Hong, C.S., 2015, August. Network-assisted congestion control for information centric networking. In 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS) (pp. 464-467). IEEE.
39. Abdelmoniem, A.M. and Bensaou, B., 2015, December. Efficient switch-assisted congestion control for data centers: an implementation and evaluation. In Proc. of Proceedings IEEE International Performance Computing and Communications Conference (IPCCC) (pp. 1-8).

40. Zhu, X. and Pan, R., 2013, December. NADA: A unified congestion control scheme for low-latency interactive video. In 2013 20th International Packet Video Workshop (pp. 1-8). IEEE.
41. Lall, S., Alfa, A.S. and Maharaj, B.T., 2016, July. The role of queueing theory in the design and analysis of wireless sensor networks: An insight. In 2016 IEEE 14th International Conference on Industrial Informatics (INDIN) (pp. 1191-1194). IEEE.
42. Aghdam, S.M., Khansari, M., Rabiee, H.R. and Salehi, M., 2014. WCCP: A congestion control protocol for wireless multimedia communication in sensor networks. Ad Hoc Networks, 13, pp.516-534.
43. Bagewadi, A. and Babu, R.M., 2014. Towards an Ethernet Learning Switch and Bandwidth Optimization using POX Controller. International Journal of Advanced Research in Computer and Communication Engineering, 3(7), pp.7531-7535.
44. Naeem, F., Srivastava, G. and Tariq, M., 2020. A software defined network based fuzzy normalized neural adaptive multipath congestion control for the internet of things. IEEE transactions on network science and engineering, 7(4), pp.2155-2164.
45. Goto, Y., Ng, B., Seah, W.K. and Takahashi, Y., 2019. Queueing analysis of software defined network with realistic openflow–based switch model. Computer Networks, 164, p.106892.
46. Chakravarthy, V.D. and Amutha, B., 2020. Software-defined network assisted packet scheduling method for load balancing in mobile user concentrated cloud. Computer Communications, 150, pp.144-149.
47. Hertiana, S.N., Kurniawan, A. and Pasaribu, U.S., 2018. Effective Router Assisted Congestion Control for SDN. International Journal of Electrical & Computer Engineering (2088-8708), 8(6).
48. Ahmed, O., Ren, F., Hawbani, A. and Al-Sharabi, Y., 2020. Energy optimized congestion control-based temperature aware routing algorithm for software defined wireless body area networks. IEEE Access, 8, pp.41085-41099.
49. Kaur, K., Garg, S., Aujla, G.S., Kumar, N., Rodrigues, J.J. and Guizani, M., 2018. Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay. IEEE communications magazine, 56(2), pp.44-51.
50. Abdelmoniem, A.M., Bensaou, B. and Abu, A.J., 2017, May. SICC: SDN-based incast congestion control for data centers. In 2017 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

51. Letswamotse, B.B., 2018. Software defined networking based resource management and quality of service support in wireless sensor network applications (Doctoral dissertation, University of Pretoria).
52. Li, P., Guo, S., Pan, C., Yang, L., Liu, G. and Zeng, Y., 2019. Fast congestion-free consistent flow forwarding rules update in software defined networking. Future Generation Computer Systems, 97, pp.743-754.
53. Firoiu, V. and Borden, M., 2000, March. A study of active queue management for congestion control. In Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064) (Vol. 3, pp. 1435-1444). IEEE.
54. Floyd, S., 2001. A report on recent developments in TCP congestion control. IEEE Communications Magazine, 39(4), pp.84-90.
55. Shenker, S., Zhang, L. and Clark, D.D., 1990. Some observations on the dynamics of a congestion control algorithm. ACM SIGCOMM Computer Communication Review, 20(5), pp.30-39.
56. Ryu, S., Rump, C. and Qiao, C., 2003. Advances in internet congestion control. IEEE Communications Surveys & Tutorials, 5(1), pp.28-39.
57. Hespanha, J.P., Bohacek, S., Obraczka, K. and Lee, J., 2001, March. Hybrid modeling of TCP congestion control. In International Workshop on Hybrid Systems: Computation and Control (pp. 291-304). Springer, Berlin, Heidelberg.
58. Mo, J. and Walrand, J., 2000. Fair end-to-end window-based congestion control. IEEE/ACM Transactions on networking, 8(5), pp.556-567.
59. Morris, R., 2000, March. Scalable TCP congestion control. In Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064) (Vol. 3, pp. 1176-1183). IEEE.
60. Zhang, L., Shenker, S. and Clark, D.D., 1991, August. Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic. In Proceedings of the conference on Communications architecture & protocols (pp. 133-147).
61. Al-Saadi, R., Armitage, G., But, J. and Branch, P., 2019. A survey of delay-based and hybrid TCP congestion control algorithms. IEEE Communications Surveys & Tutorials, 21(4), pp.3609-3638.

62. Mathis, M. and Mahdavi, J., 1996. Forward acknowledgement: Refining TCP congestion control. ACM SIGCOMM Computer Communication Review, 26(4), pp.281-291.
63. Firoiu, V. and Borden, M., 2000, March. A study of active queue management for congestion control. In Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064) (Vol. 3, pp. 1435-1444). IEEE.
64. Floyd, S., 2001. A report on recent developments in TCP congestion control. IEEE Communications Magazine, 39(4), pp.84-90.
65. Shenker, S., Zhang, L. and Clark, D.D., 1990. Some observations on the dynamics of a congestion control algorithm. ACM SIGCOMM Computer Communication Review, 20(5), pp.30-39.
66. Ryu, S., Rump, C. and Qiao, C., 2003. Advances in internet congestion control. IEEE Communications Surveys & Tutorials, 5(1), pp.28-39.
67. Hespanha, J.P., Bohacek, S., Obraczka, K. and Lee, J., 2001, March. Hybrid modeling of TCP congestion control. In International Workshop on Hybrid Systems: Computation and Control (pp. 291-304). Springer, Berlin, Heidelberg.
68. Mo, J. and Walrand, J., 2000. Fair end-to-end window-based congestion control. IEEE/ACM Transactions on networking, 8(5), pp.556-567.
69. Morris, R., 2000, March. Scalable TCP congestion control. In Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064) (Vol. 3, pp. 1176-1183). IEEE.
70. Zhang, L., Shenker, S. and Clark, D.D., 1991, August. Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic. In Proceedings of the conference on Communications architecture & protocols (pp. 133-147).