

Enhancing Window-Based Congestion Control Algorithm for the Internet

**Matthew Ojo AYEMOWA
LCU/PG/001006**

**Being a M.Sc. Thesis Submitted to the Department of Computer Science, Faculty of Basic
Medical and Applied Sciences, Lead City University, Ibadan, Oyo State, Nigeria**

**In Partial Fulfilment of the Requirement for the Award of Master Degree (MSc) in
Computer and Information Science**

2022

Certification

This is to certify that Matthew Ojo AYEMOWA with matriculation number LCU/PG/001006 carried out this research work titled “Enhanced Window-Based Congestion Control Algorithm for the Internet” in the Department of Computer Science, Faculty of Engineering and Technology, Lead City University Ibadan, Oyo State for the award of Masters of Science in Computer Science and that this has been previously submitted.

Dr. A.A Waheed
(Supervisor)

Date

Dr. Sakpere
(Head of the Department)

Date

Dedication

This Project is dedicated to my Creator for His mercy upon my life, without Him in my life am nothing and to my wonderful Children.

DO NOT COPY. LEAD CITY UNIVERSITY, NIGERIA

Acknowledgement

I will like to express my deepest gratitude to the Management of Lead City University for given me the opportunity to study in their conducive environment.

I want to specially express my appreciation to my able Supervisor, Dr. A.A Waheed for his guidance and motivation in making this project a success. I want to thank you for being very patient and helpful to me throughout the development process of my M.Sc. Programm. I equally want to express my appreciation to all the Lecturers in the Department of Computer Science for their useful contributions during my Course Work, Proposal defense, Pre-field and Post-field, thank you all for unique encouragement and for sharing your resources and acquired knowledge with me.

I am indebted to Dr.(Mrs.) F.I Oyeyinka and my Boss, Dr. I.K Oyeyinka, Rector, Gateway (ICT) Polytechnic Saapade, Ogun State for their effort, financial support and useful advice to run this program successfully, thank you for been there for me Sir/Ma. I want to specially appreciate my Children and my Lovely Wife for her patience, advice, cooperation, care in everything, thank you Dear. To my Parents who are always there for me, thanks for your caring and prayers day and night, you will surely eat the fruit of your labour in Jesus name and to my Late Grandma who always tell me to be hardworking and be focused, thank you even though you are no more.

“Even though the above-mentioned Institution and persons have assisted in the process of this research work, I alone stand responsible for the errors, if any, found in the work”

Abstract

The Transmission Control Protocol (TCP) or Internet Protocol (IP) has gone through a number of changes. There have been numerous ideas to alter the mechanisms of TCP congestion control in order to improve its performance. Many research had been done on congestion avoidance and control; however, previous studies on window-based congestion control were mostly based on the starting network, adaptive congestion, and end-to-end mechanisms; however, network overloading will be avoided, and packet losses should be minimized by using the PERC Algorithm. The goal of this study is to develop a Proactive Explicit Rate Control (PERC) algorithm for max-min fair rates that avoids network overcrowding and packet losses through sharing. For max-min fair rates, we devised the PERC algorithm, which can prioritize short flows as needed.

Motivated by the need for fast congestion control, this thesis focuses on a different class of congestion control algorithms, called proactive explicit rate-control (PERC) algorithms, which decouple the rate calculation from congestion signals in the network. The switches and Network Interface Cards (NICs) exchange control messages to run a distributed algorithm to pick explicit rates for each flow. PERC algorithms proactively schedule flows to be sent at certain explicit rates. They take as input the set of flows and the network link speeds and topology, but not a congestion signal. As a result, they converge faster and their convergence time depends only on fundamental dependency chains, essentially couplings between links that carry common flows, that are a property of the traffic matrix and the network topology. We argue that congestion control should converge in a time limited only by fundamental dependency

Our main contribution is to use PERC Algorithm to control Network Overloading and to equally control packet loss In simulation and on a P4 programmed FPGA hardware test bed, s-PERC converges an order of magnitude faster than reactive schemes such as TCP, DCTCP, and RCP.

Keywords: Congestion Control, Congestion Window, Network Overloading, Network Protocols, Packet Losses.

Word Count: 300

Table of Contents

Title	Page
Certification	ii
Dedication	iii
Acknowledgement	iv
Abstract	v
Table of Contents	vi
List of Figures	xi
List of Tables	xii
Chapter One Introduction	
1.1 Background to the Study	1
1.2 Statement of the Problem	6
1.3 Aim and Objectives of the Study	6
1.4 Scope of the Study	7
1.5 Significance of the Study	7
1.6 Organization of Report	7
1.7 Operational Definition of Terms	8
Endnotes	11
Chapter Two Literature Review	
2.1 Conceptual Review	
2.1.1 Internet Congestion	13

2.1.2 Internet Protocol(IP) Networking	14
2.2 Conceptual Framework	
2.2.1 Algorithms that are Proactive	20
2.2.2 The Models	21
2.2.3 Fairness (Maximum-Minimum)	21
2.2.4 A Per-Flow PERC Algorithm with a Max-Min	22
2.2.5 Algorithm for Calculating a Local Maximum and Minimum Rate	22
2.2.6 The Algorithm's Characteristics	23
2.2.7 The Fair Algorithm's Convergence	23
2.3 Review of the Related Works	24
2.3.1 Results of the Simulation	24
2.3.2 Scheduling in the Network	25
2.3.3 The Link-Layers	26
2.3.4 Traffic Shaping and Classification	27
2.3.5 Scheduling of Queues	28
2.3.6 Protocols for Transport	29
2.3.7 Application	34
2.3.8 Routing	35
2.3.9 Approaches with Multiple Layers	36
2.4 Theoretical Framework	
2.4.1 The Congestion of Window	37
2.4.2 Congestion control	38
2.4.3 Networks of Different Types	42

2.4.4	Mechanisms Based on Windows	43
2.4.5	Begin slowly (Exponential Increase)	44
2.4.6	Avoiding Congestion (Additive Increase)	45
2.4.7	Recovery from Congestion (Multiplicative Decrease)	45
2.4.8	What is the Cause of Congestion?	46
2.4.9	Costs of Congestion	47
2.4.10	What is the Definition of Congestion Control?	47
2.4.11	Policies to Reduce Congestion	47
2.4.11	Algorithms for Congestion Control	49
2.4.12	Transport Layer	50
2.4.13	The Transport Layer is a Service Provider	50
2.4.14	Error Prevention	52
2.4.15	Sequence Management	52
2.4.16	Loss Management	52
2.4.17	Duplication Management	53
2.4.18	Flow Management	53
2.4.19	Multiplexing	53
2.4.20	Problems of Congestion Control	50
2.4.21	Problems with Congestion and Their Solutions	54
2.4.22	The Different Types of Congestion Issues	55
2.4.23	Why Is It So Difficult to Solve the Congestion Problem?	59
2.4.24	The Policies that disturb the Congestion Control	60
2.4.25	What is the Difference Between Web and Window Applications?	63

2.4.26 Editions of Windows	65
2.4.27 Why Microsoft Windows is Called Windows?	67
2.5.1 Policies Adopted by Open Loop Congestion Control	71
2.5.2 Causes of Network Congestion	73
2.5.3 Packet Loss	78
2.5.4 Causes of Packet Loss	79
2.5.5 Effects of Packet Loss	82
2.5.6 How to Reduce Packet Loss	83
2.5.7 How to Fix Packet Loss with the Right Tool	84
2.5.8 How to Reduce Packet Loss	85
2.5.9 How to Fix Packet Loss with the Right Tool	85
2.5.10 A Classification of Multimedia Applications	88
Endnotes	92
Chapter Three: Methodology	
3.1 n-PERC: a Naive PERC Algorithm without Per-Flow State	97
3.1.1 The PERC Algorithm is a Method for Calculating the Probability of a Certain Event Occurring	102
3.2 The s-PERC Algorithm is Stateless PERC with a Fixed Convergence Speed	104
3.2.1 PERC Algorithm's Variables	105
3.2.2 The PERC Algorithm (s-PERC)	106
3.3 In Action with s-PERC	107
3.4 Simulations with PERCs Algorithms	110
3.5 s-PERC Convergence Proof	111
3.5.1 Algorithms for Centralized Water Filling	111

Endnotes	114
Chapter Four: Results and Discussion of Findings	
4.1 Simulation of Algorithms for Packet Controls and Network Overloading	116
4.2 Hardware s-PERC Prototype	116
4.3 End Host in MoonGen PERC	119
4.4 Time to Converge	120
4.5 Time to Complete a Flow	121
4.6 Hardware Evaluation of NetFPGA	123
Endnotes	130
Chapter Five: Conclusion	
5.1 Summary of Findings	131
5.2 Recommendations	132
5.3 Contribution to the Knowledge	133
5.4 Suggestion for Further Studies	134
Bibliography	136
Biodata	139
University Compliance Certification	143

List of Figures

Figure	Title	Page
2.1	The OSI Networking Stacks.	14
2.2	Transport Layers	51
2.3	Computer Window	64
2.4	Web Application	69
2.5	Open Loop Congestion Control	71
2.6	Picture of Network Overloading	73
2.7	Multicasting	75
2.8	Packet Tracer	79
2.9	Network Congestion	80
2.12	Picture of Packet Loss	87
2.13	Software Bug	88
3.1	Congestion Architecture	97
4.1	Convergence of RTT	117
4.2	Time Series of Flow Transmission Rates	118
4.3	Flow of Links	121
4.4	Example Convergence Behavior	124
4.5	An Example for Which n-PERC	125
4.6	Examples of Indirect Precedent Links	126
4.7	CDF of Convergence Times	127
4.8	PERC vs. TFRC	127

List of Tables

Table	Title	Page
2.1	Differences Between Window Applications and Web Applications	70
3.1	n-PERC Flow fG's First Control Packet	98
3.2	Algorithm for Network Overloading	100
3.3	Algorithm for Control Packet Processing	105
3.4	Initial Control Packet for Flow fG in s-PERC	105
3.5	s-PERC Arrangement	108
3.6	PERC Algorithm Configuration	113
4.1	Algorithm for the Processing of Control Packet	124
4.2	On a NetFPGA Tested, Illustrates the Results of Incast Experiments	124