CONGESTION CONTROL ALGORITHMS FOR END-TO-END COMMUNICATION OVER MMWAVE CELLULAR NETWORKS

By

ALRAMLI OMAR IMHEMED ABDULMAJID

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia,
in Fulfilment of the Requirements for the Degree of Doctor of
Philosophy

May 2025

DEDICATIONS

I dedicate this thesis to the soul of my father.

&

To All whom I love.

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of Doctor of Philosophy

CONGESTION CONTROL ALGORITHMS FOR END-TO-END COMMUNICATION OVER MMWAVE CELLULAR NETWORKS

By

ALRAMLI OMAR IMHEMED ABDULMAJID

May 2025

Chairman : Associate Professor Zurina binti Mohd Hanapi , PhD
Faculty : Computer Science and Information Technology

The widespread deployment of cellular networks has driven a surge in demand for high-performance services, particularly for applications such as virtual and augmented reality that require high data rates and low latency. Millimeter wave (mmWave) technology has emerged as a promising solution due to its high frequency, large bandwidth, and fast transmission speeds. However, the Transmission Control Protocol (TCP), which dominates Internet traffic, faces serious challenges in adapting to mmWave cellular networks. These networks exhibit characteristics—such as frequent transitions from Line of Sight (LoS) to Non-Line of Sight (NLoS), high signal attenuation, small buffer sizes, increased Round-Trip Time (RTT), and elevated Packet Error Rates (PER)—that degrade TCP performance and lead to inefficient bandwidth utilization and non-congestion-related packet losses. To mitigate these limitations and improve the TCP performance over mmWave cellular networks, three novel Congestion Control Algorithms (CCAs) are proposed:

RTTV-based CCA (Round Trip Time Variations-based): Designed to address NLoS-induced non-congestion losses by adjusting the congestion window (cwnd) based on RTT variations. Simulation results demonstrate that RTTV-CCA significantly improves throughput and reduces latency compared to baseline protocols, outperforming the benchmark TCP protocols (NewReno, HighSpeed, CUBIC, BBR) by (1253.96, 207.72, 1253.96, 5.95)% respectively under high PER conditions when buffer size matches the Bandwidth Delay Product (BDP).

MSS-based CCA (Maximum Segment Size-based) targets scenarios with small buffers and high packet error rates (PER), correlating congestion window (cwnd) growth with the MSS to better adapt to fluctuating network conditions. MSS-CCA demonstrates superior bandwidth utilization and reduced latency compared to existing protocols when the buffer size matches the bandwidth-delay product (BDP). Specifically, it outperforms benchmark protocols (NewReno, HighSpeed, CUBIC, and BBR) by (1486.72, 261.53, 1486.72, and 24.26)%, respectively, under high PER conditions, while maintaining low latency.

MRVHS-based CCA: Integrates principles from MSS, RTTV, and HighSpeed-based CCAs to optimize performance in end-to-end scenarios involving remote servers. Notably, MRVHS significantly outperforms the benchmark protocols (NewReno, HighSpeed, CUBIC, BBR, FB-TCP) by (1242.05, 347.35, 347.35, 4.75, 5.48)%, respectively, in the worst case of PER scenarios. Moreover, MRVHS demonstrates substantial throughput gains in medium- and low-PER conditions, consistently surpassing the benchmark TCP protocols.

A comprehensive performance assessment of the proposed CCAs is carried out through simulation using the ns-3 network simulator. This environment includes an advanced mmWave module specifically designed to emulate the distinctive propagation behav-

iors and rapidly changing conditions typical of mmWave cellular networks, particularly in urban settings. The simulation framework incorporated a range of network scenarios, reflecting realistic transitions between LoS and NLoS states, varying buffer sizes, and different configurations of PER and RTT. Furthermore, key performance metrics (KPI) such as throughput and latency were systematically measured and analyzed, providing a comparative assessment of the proposed CCAs against benchmark Algorithms.

The proposed CCAs demonstrate clear performance gains over current benchmarks, offering practical solutions to improve TCP efficiency in mmWave cellular networks. These contributions are significant for enabling reliable and high-speed connectivity in next-generation wireless systems, supporting the growing demands of emerging applications.

Keywords: mmWave Cellular Networks, MRVHS, MSS, RTTV, Transmission Control Protocol (TCP)

SDG: GOAL 9: Industry, Innovation and Infrastructure

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai
memenuhi keperluan untuk ijazah Doktor Falsafah

ALGORTIMA KAWALAN KESESAKAN UNTUK KOMUNIKASI HUJUNG-KE-HUJUNG MELALUI RANGKAIAN SELULAR MMWAVE

Oleh

ALRAMLI OMAR IMHEMED ABDULMAJID

Mei 2025

Pengerusi : Profesor Madya Zurina binti Mohd Hanapi, PhD
Fakulti : Sains Komputer dan Teknolologi Maklumat

Penggunaan rangkaian selular yang meluas telah mendorong lonjakan permintaan terhadap perkhidmatan berprestasi tinggi, terutamanya untuk aplikasi seperti realiti maya dan reality tambahan yang memerlukan kadar data yang tinggi dan kependaman yang rendah. Teknologi gelombang milimeter (mmWave) telah muncul sebagai satu penyelesaian yang menjanjikan kerana frekuensinya yang tinggi, lebar jalur yang besar dan kelajuan penghantaran yang pantas. Walau bagaimanapun, Protokol Kawalan Penghantaran (TCP), yang mendominasi trafik Internet, menghadapi cabaran serius dalam menyesuaikan diri dengan rangkaian selular mmWave. Rangkaian ini mempamerkan ciri-ciri— seperti peralihan yang kerap daripada garis nampak (LoS) kepada bukan garis nampak (NLoS), pengecilan isyarat yang tinggi, saiz penimbal yang kecil, peningkatan Masa Pergi Balik (RTT) dan Kadar Ralat Paket (PER) yang tinggi—yang menjejaskan prestasi TCP dan membawa kepada penggunaan lebar jalur yang tidak cekap dan kehilangan paket yang bukan disebabkan oleh kesesakan. Untuk mengurangkan had ini dan meningkatkan prestasi TCP melalui rangkaian selular mmWave, tiga Algoritma Kawalan Kesesakan (CCA) novel telah dicadangkan:

CCA berasaskan RTTV (Berasaskan Variasi Masa Pergi Balik): Direka bentuk untuk menangani kehilangan bukan akibat kesesakan yang disebabkan oleh keadaan NLoS dengan melaraskan tetingkap kesesakan (cwnd) berdasarkan variasi dalam Masa Pergi Balik (RTT). Hasil simulasi menunjukkan bahawa RTTV-CCA meningkatkan kadar pemindahan data yang ketara dan mengurangkan kependaman berbanding dengan protokol asas, mengatasi prestasi penanda aras protokol-protokol TCP (NewReno, HighSpeed, CUBIC, BBR) masing-masing sebanyak (1253.96, 207.72, 1253.96, 5.95)% di bawah keadaan Kadar Ralat Paket (PER) yang tinggi apabila saiz penimbal sepadan dengan Produk Kelewatan Lebar Jalur (BDP).

CCA berasaskan MSS (Berasaskan Saiz Segmen Maksimum) menyasarkan senario dengan penimbal kecil dan kadar ralat paket (PER) tinggi, mengaitkan pertumbuhan tetingkap kesesakan (cwnd) dengan MSS untuk menyesuaikan diri dengan keadaan rangkaian yang berubah-ubah dengan lebih baik. MSS-CCA menunjukkan kecekapan penggunaan lebar jalur dan pengurangan kependaman berbanding dengan protokol sedia ada apabila saiz penimbal sepadan dengan produk kelewatan lebar jalur (BDP). Khususnya, ia mengatasi prestasi protokol-protokol penanda aras (NewReno, HighSpeed, CUBIC dan BBR) masing-masing sebanyak (1486.72, 261.53, 1486.72, dan 24.26)%, di bawah keadaan PER yang tinggi, sambil mengekalkan kependaman yang rendah.

CCA berasaskan MRVHS: Mengintegrasikan prinsip daripada MSS, RTTV dan CCAs berasaskan Kelajuan Tinggi untuk mengoptimumkan prestasi dalam senario hujung ke hujung yang melibatkan pelayan jauh. Secara khusus, MRVHS mengatasi prestasi protokol-protokol penanda aras (NewReno, HighSpeed, CUBIC, BBR, dan FB-TCP) dengan ketara, masing-masing sebanyak (1242.05, 347.35, 347.35, 4.75, dan 5.48)%

dalam senario PER terburuk. Tambahan pula, MRVHS menunjukkan peningkatan daya pengeluaran yang besar dalam keadaan PER sederhana dan rendah, secara konsisten mengatasi protokol-protokol TCP penanda aras.

Penilaian prestasi komprehensif bagi CCA yang dicadangkan dijalankan melalui simulasi menggunakan simulator rangkaian ns-3. Persekitaran ini termasuk modul mmWave lanjutan yang direka khusus untuk meniru gelagat penyebaran yang tersendiri dan keadaan yang berubah pantas seperti rangkaian selular mmWave, terutamanya dalam tetapan bandar. Rangka kerja simulasi menggabungkan pelbagai senario rangkaian, mencerminkan peralihan realistik antara keadaan LoS dan NLoS, saiz penimbal yang berbeza-beza dan konfigurasi PER dan RTT yang berbeza. Tambahan pula, metrik prestasi utama (KPI) seperti pemprosesan dan kependaman diukur dan dianalisis secara sistematik, memberikan penilaian perbandingan CCA yang dicadangkan terhadap Algoritma penanda aras.

CCA yang dicadangkan menunjukkan peningkatan prestasi yang jelas berbanding penanda aras semasa, menawarkan penyelesaian praktikal untuk meningkatkan kecekapan TCP dalam rangkaian selular mmWave. Sumbangan ini penting untuk membolehkan sambungan yang boleh dipercayai dan berkelajuan tinggi dalam generasi sistem tanpa wayar akan datang, menyokong permintaan yang semakin meningkat bagi aplikasi-aplikasi yang sedang muncul.

Kata kunci: Rangkaian Selular mmWave, MRVHS, MSS, RTTV, Protokol Kawalan Penghantaran (TCP)

SDG: MATLAMAT 9: Industri, Inovasi dan Infrastruktur

ACKNOWLEDGEMENTS

I am extremely thankful to Allah, my Lord, for granting me the strength, wisdom, and perseverance to complete this thesis successfully.

I am indebted to my supervisor, Associate Professor Dr. Zurina Mohd Hanapi, for her unwavering support, guidance, and encouragement throughout the entire research and writing process. Her patience, motivation, and expertise have been invaluable, and I am forever grateful for her mentorship. I would also like to express my gratitude to the members of my supervisory committee, Prof. Mohamed Othman, Dr. Normalia Samian, and Dr. Idawaty Ahmad, for their valuable insights and feedback.

I am grateful to the Faculty of Computer Science and Information Technology, the staff of the Postgraduate Office, the School of Graduate Studies, the Library, and Universiti Putra Malaysia for providing me with a conducive research environment.

I am deeply grateful to my family, especially my mother, for her unwavering love, and my wife for her constant support and inspiration. My heartfelt thanks to my daughters and sons for their love throughout this Ph.D. journey. I also sincerely appreciate my sisters and brothers for their continuous encouragement.

I am also grateful to Drs. Alrashah, Ala'anzy, and Oluwatosin, as well as my lab mates Albadwi, Bouke, Khalaf, and Alsaaidah, for their support and encouragement throughout this journey. would also like to apologise to anyone whom I may have inadvertently omitted in this acknowledgment.

I certify that a Thesis Examination Committee has met on 9 May 2025 to conduct the final examination of Alramli Omar Imhemed Abdulmajid on his thesis entitled "Congestion Control Algorithms for End-to-End Communication Over mmWave Cellular Networks" in accordance with the Universities and University Colleges Act 1971 and the Constitution of the Universiti Putra Malaysia [P.U.(A) 106] 15 March 1998. The Committee recommends that the student be awarded the Doctor of Philosophy.

Members of the Thesis Examination Committee were as follows:

Rohaya binti Latip, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Zuriati binti Ahmad Zukarnain, PhD
Professor Ts.
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Internal Examiner)

Azizol bin Hj Abdullah, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Internal Examiner)

Jemal Abawajy, PhD
Professor
Faculty of Science, Engineering and Built Environment
Deakin University
Australia
(External Examiner)

RUSMAWATI BINTI SAID, PhD
Professor and Deputy Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 24 July 2025

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Doctor of Philosophy. The members of the Supervisory Committee were as follows:

Zurina binti Mohd Hanapi, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Mohamed bin Othman, PhD
Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

Normalia binti Samian, PhD
Assistant Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

Idawaty binti Ahmad, PhD
Assistant Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

ZALILAH MOHD SHARIFF, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 21 August 2025

Declaration by the Graduate Student

I hereby confirm that:

- this thesis is my original work;
- quotations, illustrations and citations have been duly referenced;
- this thesis has not been submitted previously or concurrently for any other degree at any institutions;
- intellectual property from the thesis and the copyright of the thesis are fully-owned by Universiti Putra Malaysia, as stipulated in the Universiti Putra Malaysia (Research) Rules 2012;
- written permission must be obtained from the supervisor and the office of the Deputy Vice-Chancellor (Research and innovation) before the thesis is published in any written, printed or electronic form (including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials) as stated in the Universiti Putra Malaysia (Research) Rules 2012;
- there is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld in accordance with the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2015-2016) and the Universiti Putra Malaysia (Research) Rules 2012. The thesis has undergone plagiarism detection software

Signature: Alramli Date: 3 Oct 2025

Name and Matric No.: Alramli Omar Imhemed Abdulmajid, GS57884

Declaration by Members of the Supervisory Committee

This is to confirm that:

- the research and the writing of this thesis were done under our supervision;
- supervisory responsibilities as stated in the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2015-2016) are adhered to.

Signature:
Name of Chairman of Supervisory Committee: **Assoc. Prof. Dr. Zurina binti Mohd. Hanapi**

Signature:
Name of Member of Supervisory Committee: **Prof. Mohamed bin Othman**

Signature:
Name of Member of Supervisory Committee: **Dr. Normalia binti Samian**

Signature:
Name of Member of Supervisory Committee: **Dr. Idawaty binti Ahmad**

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

TCP	Transmission Control Protocl
UDP	User Datagram Protocol
CCA	Congestion Control Algorithm
BBR	Bottleneck Bandwidth and Round-trip
RTTV-CCA	Round Trip Time Variations CCA
MSS-CCA	Maximum Segment Size CCA
MRVHS-CCA	Maximum Segment Size and Round Trip Time Variations CCA
Cwnd	Congestion Window
SS	Slow-Start
CA	Congestion Avoidance
SSTH	Slow Start Threshold
AIMD	Additive-Increase/Multiplicative-Decrease
RTT	Round Trip Time
RTO	Retransmission Time-Out
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
ACK	Acknowledgment
3DACK	Three Duplicate Acknologmnets
PER	Packet Error Rate
BDP	Bandwidth-Delay Product

mmWave	Millimeter Wave
5G	Fifth Generation
gNB	Base Station
UE	User Equipment
ITR	Instantaneous Transmission Rate
LoS	Line of Sight
NLoS	Non Line of Sight
SNR	Signal to Noise Ratio
SINR	Signal-to-Interference-plus-Noise Ratio
NS-3	Network Simulator 3
3GPP	3rd Generation Partnership Project
CDF	Cumulative Distribution Function

CHAPTER 1

INTRODUCTION

1.1 Millimeter Wave Cellular Networks

The landscape of mobile broadband communication is rapidly transforming, with the goal of achieving near-zero latency and higher data rates reaching tens of gigabits per second over wireless channels for mobile applications. Recent studies by the Third Generation Partnership Project (3GPP)[1] show that the number of mobile users surged from 5.1 billion in 2018 to 5.8 billion by 2025. Additionally, projections indicate a significant increase in IoT subscriptions, from 10 billion to 25 billion by 2025. Also, Ericsson report as Figure 1.1 shows the number of subscriptions of 5G is forecast to reach 5.6 billion by the end of 2029 (Ericsson report) [2]. These statistics highlight the critical importance of system capacity and Quality of Service (QoS) as key drivers of technological progress, from the current Fifth Generation (5G) to advancements in 5G and Beyond 5G (B5G), ultimately leading to the future Sixth Generation (6G) of mobile systems. As the demand for higher data rates in some applications, as shown in Figure 1.2, innovative features, and advanced services continues to escalate, the imperative to enhance bandwidth in 5G generation mobile networks becomes unavoidable (Mendonça et al., 2022; Sylla et al., 2022; Poorzare and Waldhorst, 2023). The advent of 5G networks brings forth three primary use cases: Enhanced Mobile Broadband (eMBB), which is designed for high data rates; Massive Machine Type Communication (mMTC), which facilitates connectivity for a large number of devices per square kilometer; and Ultra-Reliable Low-Latency Communication (URLLC), which aims for a 1 ms latency ideal for critical communications such as Vehicle-to-Everything (V2X) (Kanaya et al., 2020; Poorzare and Augé, 2020; Navarro-Ortiz et al., 2020).

---

[1] http://www.3gpp.org/

[2] https://www.ericsson.com/en/

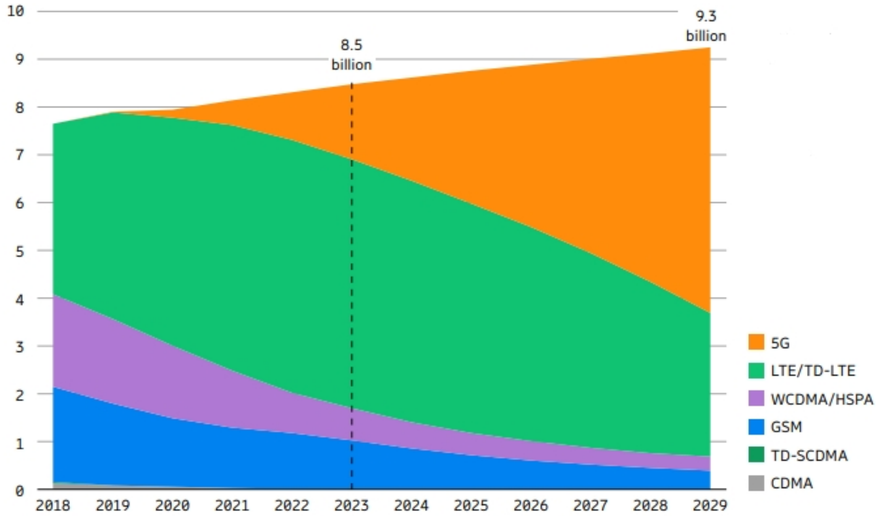


Figure 1.1: Mobile Subscriptions by Technology

(Source: Ericsson, 2024)

Figure 1.2: 5G Applications

(Source: 3GPP, 2024)

Additionally, disaster response and remote healthcare applications are poised to benefit significantly from the low-latency, high-bandwidth wireless access networks offered by 5G (Na et al., 2019; Sudhamani et al., 2023). The overarching goal is to establish flexible networks connecting everything, everywhere, and anytime. The advancements promised by mmWave in cellular networks over previous generations are significant, offering higher peak data rates and low latency (Poorzare and Calveras, 2023; Salahdine et al., 2023; Islam et al., 2025). A primary objective in 5G is to achieve seamless connections for mobile User Equipment (UE), aiming for uninterrupted mobility between UE and a gNodeB (gNB) base station (Poorzare and Augé, 2020). The deployment of 5G has seen rapid acceleration since the launch of the first 5G device in 2018, as Figure 1.1 shows. The transition from older generations to 5G is gaining momentum and significant growth.

To meet the bandwidth requirements of 5G, leveraging radio frequencies beyond 3 GHz, including mmWave, becomes indispensable. While mmWave offers high capacity and high data rates, it presents challenges such as limited coverage and poor penetration through materials like trees, buildings, and the human body. Additionally, despite the advantages of mmWave, challenges like blockages can impact network performance, particularly in terms of end-to-end reliable communication (Niu et al., 2015; Zhang et al., 2017; Poorzare and Augé, 2020; Kim and Cho, 2022; Xie et al., 2023). Transmission Control Protocol (TCP) (Postel, 1981b), a widely used transport layer protocol, plays a crucial role in end-to-end communication. However, it faces challenges in mmWave cellular networks, such as Non Line of Sight (NLoS) conditions, packet errors, small buffers, and server location. Addressing these challenges is crucial for maintaining stable connections and achieving optimal end-to-end communication within cellular networks.

For instance, the primary challenge in optimizing TCP performance within mmWave cellular networks lies in tackling blockages between User Equipment (UE) and gNB (base stations). These blockages result in the transition from Line-of-Sight (LoS) to Non-Line-of-Sight (NLOS) states, as illustrated in Figure 1.3, which can weaken the strength of mmWave signals by disrupting communication and influencing the TCP congestion control mechanism due to TCP's response to non-congestion cases (Zhang et al., 2019). TCP does not function effectively when frequent interruptions occur in the network, as it cannot differentiate whether a packet loss is a result of congestion or other deficiencies in the cellular network, such as blockages, random packet losses (Polese et al., 2017b; Mateo et al., 2019; Kanagarathinam et al., 2020). Furthermore, whether the server is located at the edge or remotely from the gNB station plays a crucial role in the performance degradation of TCP in these networks. Therefore, tackling these challenges to enhance TCP performance of end-to-end communication and maintain stable connections is imperative. Failing to do leads to a decline in the performance of end-to-end communication over mmWave cellular networks, preventing them from meeting the requirements of mobile applications. This challenge is particularly pronounced in mmWave due to the characteristics of mmWave cellular networks.

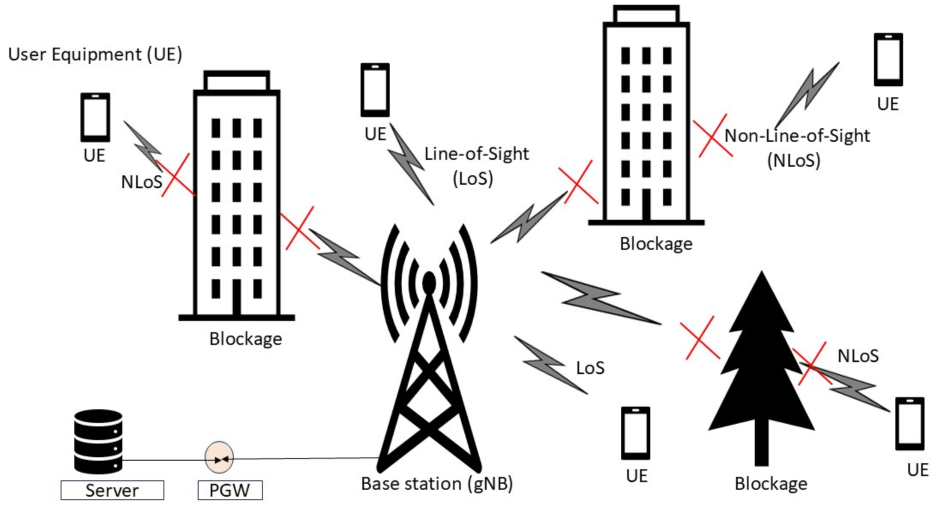


Figure 1.3: Cellular Network Architecture

1.2 Motivation

While TCPs have been widely deployed, CCAs face significant challenges to adjust cwnd in order to utilize the available bandwidth efficiently of mmWave cellular networks. The performance evaluation of state-of-the-art algorithms indicates that existing TCPs are notably inadequate, particularly in mmWave cellular networks. This deficiency provides an opportunity for researchers to enhance these CCAs continually. Furthermore, the rapid advancements in network technologies necessitate further improvements to TCP to meet the demands of cutting-edge applications in cellular networks, including virtual and augmented reality.

1.3 Problem Statement

The Transmission Control Protocol (TCP) is the dominant protocol for end-to-end communications on the Internet, valued for its reliability, in-order data delivery, and robust error-checking mechanisms. By establishing a connection-oriented communication model that employs acknowledgments, retransmissions, and flow control, TCP ensures reliable data transfer across diverse networks. This reliability makes TCP integral to a wide range of applications, including web browsing, file transfers, and email. As a result, TCP forms a foundational component of Internet architecture. TCP was originally designed for stable and predictable wired network environments. In contrast, fifth-generation (5G) millimeter wave (mmWave) cellular networks introduce highly dynamic and error-prone wireless conditions that significantly challenge TCP's effectiveness. When deployed in mmWave environments, TCP suffers from substantial performance degradation, including poor bandwidth utilization and increased latency. These issues limit the protocol's ability to meet the stringent performance requirements of emerging mobile applications such as augmented reality (AR), virtual reality (VR), and ultra-high-definition video streaming.

A major challenge arises from the frequent transitions between Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) conditions caused by environmental blockages (e.g., buildings, trees) and the mobility of User Equipment (UE) relative to the base station (gNodeB). During NLoS periods, the TCP sender often fails to receive timely acknowledgments, leading to retransmission timeouts and reductions in the congestion window (cwnd). This results in reduced transmission rates and underutilized available bandwidth, even though these losses are not congestion-related.

Another critical limitation is TCP's poor adaptability to the small buffer sizes typically present in the intermediary links between the UE and remote servers. Small buffers constrain the volume of in-flight data, limiting throughput and hindering TCP's ability to grow the cwnd efficiently. This slow growth exacerbates performance issues in high-speed mmWave environments where rapid adaptation is essential. At the same time, high Packet Error Rate (PER) factors compound TCP limitations in mmWave cellular networks due to signal interference, multipath fading, Doppler shifts, and environmental conditions. Elevated PER causes frequent packet losses and retransmissions, which TCP misinterprets as congestion. This misjudgment disrupts congestion control dynamics and further reduces throughput and bandwidth efficiency.

In addition, TCP faces difficulties in adapting to the delay variability introduced by different server deployment scenarios. Whether a server is located at the network edge or remotely, TCP struggles to maintain consistent performance due to delay-induced feedback inaccuracy. In remote server scenarios, the increased Round-Trip Time (RTT) leads to slower cwnd growth, reduced responsiveness, and overall throughput degradation.

These limitations highlight the need for adaptive and intelligent Congestion Control Algorithms (CCAs) that can effectively operate in mmWave cellular environments. Such algorithms must be capable of distinguishing between congestion and non-congestion-induced losses, respond dynamically to changing link conditions, accommodate small buffer sizes, adapt to high PER, and manage varying RTTs due to server location. Therefore, the central problem addressed in this thesis is the inadequacy of existing TCP congestion control mechanisms in coping with the distinctive challenges of mmWave cellular networks. This calls for the design and development of novel CCAs that can enhance TCP's performance, ensuring efficient and reliable data transmission in next-generation wireless systems.

1.4 Research Aims and Objectives

This research aims to enhance end-to-end communication performance over mmWave cellular networks by designing congestion control algorithms that address TCP issues caused by NLoS conditions, small buffers, high error rates, and server location, whether the server is positioned at the edge or remote. To achieve these aims, the following specific objectives have been outlined:

- To propose a CCA, named RTTV-based CCA, to improve TCP performance and increase bandwidth utilization due to non-congestion states that are caused by NLoS conditions over mmWave cellular networks. This objective directly targets the issue of non-congestion states due to NLoS conditions caused by UE mobility. The new algorithm aims to maintain a stable connection and optimal throughput under these conditions.
- To propose a CCA, named MSS-based CCA, to enhance TCP performance by maximizing throughput and lowering latency despite small buffers and high error rates over mmWave cellular networks. This objective addresses the challenges posed by small buffer sizes and high PERs in mmWave networks. The algorithm aims to adjust the congestion window efficiently and handle frequent retransmissions, thereby improving throughput and bandwidth utilization.
- To propose a CCA named MRVHS-based to enhance bandwidth utilization over mmWave cellular networks when the remote server is considered. This algorithm aims to improve TCP performance in scenarios where the server is located remotely, specifically addressing the issue arising from heightened RTT in the presence of high PER. MRVHS is based on Maximum Segment Size, Round-Trip Time Variation, and HighSpeed-TCP. MRVHS-based CCA is proposed to tackle the increasing RTT issue between UE and the server, which causes TCP performance degradation over cellular-cloud networks.

By systematically addressing these objectives, this research aims to develop adaptive CCAs that significantly enhance TCP performance in mmWave cellular networks, ensuring robust and reliable end-to-end communications over these mmWave networks.

1.5 Research Hypotheses

Hypothesis 1: The RTTV-based Congestion Control Algorithm (CCA) is expected to enhance TCP throughput and optimize bandwidth use in mmWave cellular networks, particularly under non-congestion losses caused by transitions to Non-Line-of-Sight (NLoS) conditions, when compared to conventional TCP variants.

Hypothesis 2: The MSS-based CCA is anticipated to deliver superior TCP performance by increasing throughput and minimizing latency, especially in environments characterized by limited buffer sizes and elevated packet error rates (PER), relative to standard TCP protocols.

Hypothesis 3: The MRVHS-based CCA is projected to offer improved TCP efficiency and bandwidth utilization in scenarios involving remote servers, where high round-trip times (RTT) and PER are present. Its performance is expected to surpass that of traditional and mmWave-optimized TCP schemes, including FB-TCP.

1.6 Research Significance

The relevance of this research stems from the demand for an effective TCP protocol capable of adapting to the unique characteristics of mmWave cellular networks to utilize the available high bandwidth and improve end-to-end communications. The main challenge is addressing the impact of NLoS conditions on TCP performance. Additionally, it aims to reduce susceptibility to small buffers and packet error losses to enhance overall throughput and decrease latency. Furthermore, it seeks to mitigate the impact of RTT issues on TCP performance when a remote server is considered. This research is significant in fulfilling the need for an efficient TCP solution tailored to overcome the challenges of mmWave cellular networks.

1.7 Research Scope

This thesis examines TCP and enhances end-to-end communication performance over mmWave cellular networks. The primary focus lies on refining the performance of the CCA of TCP exclusively at the sender side within the transport layer over end-to-end communications. These enhancements aim to address the substantial demand for applications such as augmented reality, virtual reality, and cloud gaming in cellular networks that necessitate high data rates and low latency.

The experimentation uses the well-established network simulator 3 (ns-3), a free and open-source simulator. It's important to note that all experiments are simulated exclusively over mmWave cellular networks. Consequently, applying the proposed algorithms to other network technologies falls beyond the scope of this thesis and the scope in detail as shown in Figure 1.4.

1.8 Thesis Organisation

The remainder of this thesis is structured as follows:

Chapter 2 - provides a comprehensive literature review and discusses state-of-the-art protocols.

Chapter 3 - outlines the research methodology utilized in this thesis, covering the research framework, experimental setup, network topologies, proposed methods, and

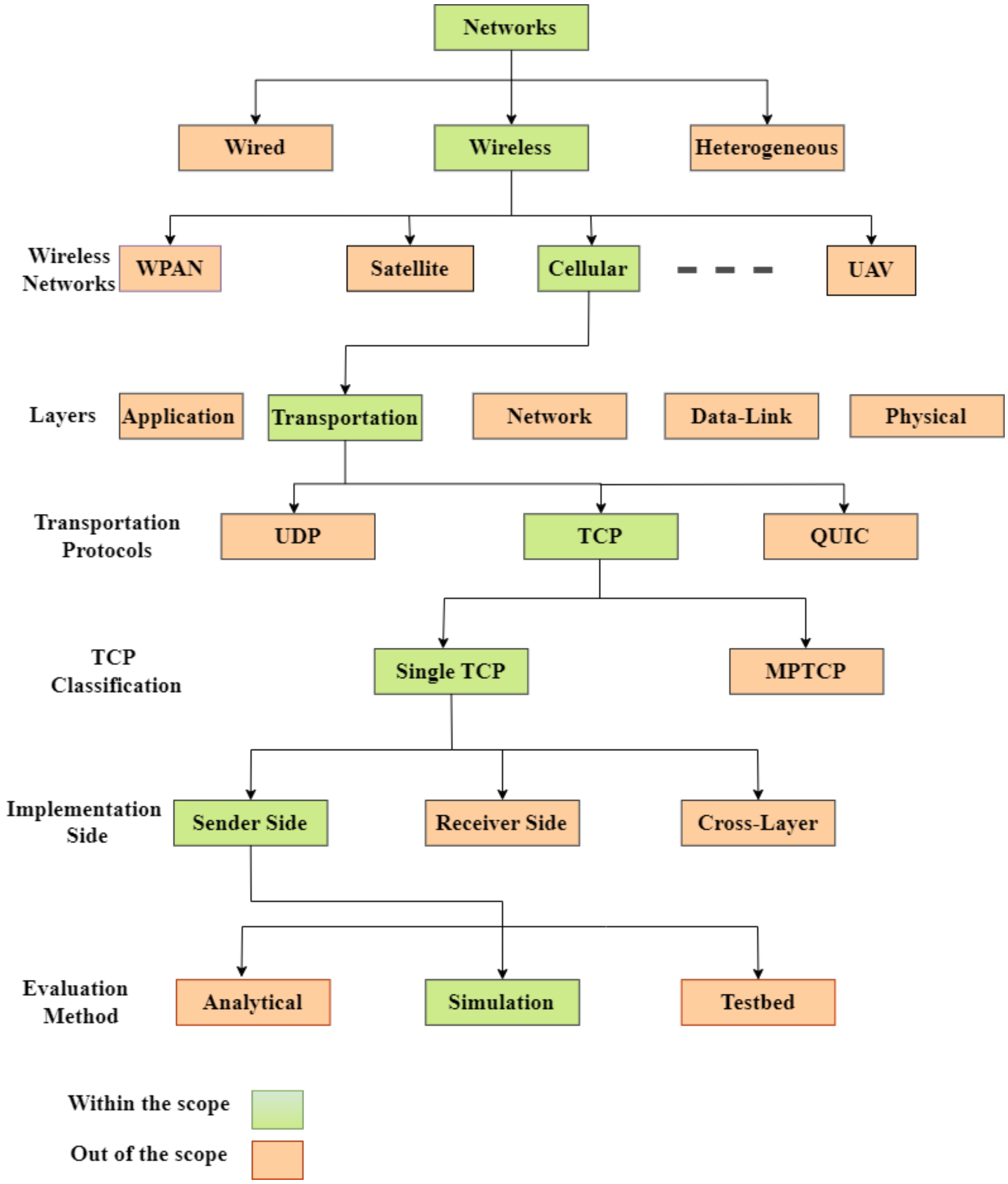


Figure 1.4: Thesis Research Scope

performance metrics. Furthermore, presents extensive simulation experiments to evaluate state-of-the-art benchmark variants in mmWave cellular networks to investigate and highlight the TCP protocols' weaknesses in mmWave cellular networks.

Chapter 4 - elucidates the proposed RTTV-based CCA, designed to address non-congestion states which are caused by Non-Line-of-Sight (NLos) challenges that degrade TCP performance in mmWave cellular networks.

Chapter 5 - introduces the proposed MSS-based CCA, designed to mitigate TCP performance issues arising from small buffers and high error rates in mmWave cellular networks.

Chapter 6 - presents MRVHS-CCA, which aims to enhance the performance of CCA by adapting the cwnd to varying delays in mmWave cellular networks when the server is located remotely from the gNB base station.

Finally, **Chapter 7 -** serves as the conclusion, summarizing the thesis, highlighting its contributions, and outlining avenues for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The introduction of TCP by Cerf and Kahn in 1974, initially designed for wired networks, heralded a new era in network communication. Since its inception, TCP has been subject to continuous refinement and adaptation across different operating systems, undergoing rigorous real-world testing. However, rapid advancements in network technologies, particularly the rise of wireless networks, have presented TCP with numerous challenges. These include congestion, redundant retransmissions, out-of-order delivery, bandwidth underutilization, inequitable resource allocation, error loss, and non-congestion-related loss (Afanasyev et al., 2010; Zhang et al., 2019; Alrshah et al., 2019; Poorzare and Augé, 2020; Kanagarathinam et al., 2020; Xie et al., 2023). In light of these challenges, researchers have extensively investigated TCP performance across various networks, considering different factors, and creating numerous TCP variants. Each variant is tailored to address particular issues; some prioritize achieving higher throughput and others minimize latency.

The performance degradation of TCP in mmWave cellular networks is particularly pronounced, given that its design did not account for the unique characteristics of these networks, such as channel loss, misalignment, rain, mobility, and blockage. In data communications, TCP efficiency hinges on its ability to utilize available bandwidth and minimize latency. Compared to the previous generations, the substantial bandwidths offered by 5G mmWave networks require special consideration (Mateo et al., 2019). Neglecting the distinctive features of mmWave bands can result in suboptimal TCP performance, especially with the anticipated high-speed rates in 5G, Beyond 5G, and future generation (6G).

Thus, this chapter comprehensively looks at the state-of-the-art TCPs and the latest TCP solutions for mmWave cellular networks. It then categorizes these solutions and details their key features, advantages, and design challenges. The taxonomy covered in this section encompasses fundamental methods and TCP protocols tailored for cellular networks. The research contributions are methodically summarized, enabling us to identify current challenges and point out promising avenues for new research.

2.2 Background on Transmission Control Protocol (TCP)

In recent years, Transmission Control Protocol (TCP), initially introduced by Cerf and Dalal (Cerf et al., 1974) and later developed by Cerf and Kahn (Cerf and Kahn, 1974), along with Postel's contributions (Postel, 1981a), has been widely adopted by numerous Internet applications, including file transfer, email, the World-Wide-Web, and remote administration. It is one of the original components of the Internet protocol suite, commonly called TCP/IP, complementing the Internet Protocol (IP). The crucial aspect of TCP lies in its Congestion Control Algorithm (CCA), which manages the data transmission rate of end-to-end communication over networks. The CCA gradually adjust the congestion window (cwnd) to increase and decrease the transmission rate according to the available network capacity while mitigating congestion risks. To provide a clearer insight, the subsequent subsections offer a concise explanation of the primary CCA components of TCP, and Figure 2.1 illustrates the general behavior of standard CCA of TCP protocol.

2.3 Slow-Start Mechanism

The fundamental principle underlying slow-start (SS) phase of CCA is to explore and assess the accessible bandwidth methodically. This estimated bandwidth serves as the basis for controlling Instantaneous Transmission Rate (ITR), ensuring that it remains equivalent to the quantity of data segments (cwnd) dispatched from the source to the destination within each Round Trip Time (RTT) as shown in Equation (2.1).

$$ITR = \frac{Cwnd}{RTT} \tag{2.1}$$

The congestion window (cwnd) is initially configured with a small value known as the initial congestion window (icwnd). Initially, the icwnd was established at either ”one” or ”two” segments, as specified in RFC2581 by Allman et al (Allman, 1999). Subsequently, in RFC3390, also by Allman et al (Allman et al., 2002). The IW was adjusted to a value between ”two” and ”four” segments. Later, Dukkipati et al. (Chu et al., 2013) from Google recommended a further increase in the icwnd to ”ten” segments. Once TCP initiates with cwnd set to icwnd, it undergoes exponential growth by ”one” for each non-duplicated Acknowledgement (ACK) arrival as shown in Figure 2.1. In other words, the new cwnd is incremented by doubling the previews cwnd every Round Trip Time (RTT). This phase concludes upon encountering either a packet loss event or reaching the slow-start threshold (ssthresh), depending on which event occurs first. In the case of a packet loss, the CCA reduces its cwnd using the multiplicative decrease factor (MD); otherwise, if the cwnd reaches ssthersh, it concludes this stage without reduction. Subsequently, CCA transitions to another phase known as congestion avoidance (CA), employing a linear increase mechanism. However, if the Retransmission Time Out (RTO) counter expires, the cwnd is reset to icwnd, initiating a new slow-start phase.

2.4 Congestion Avoidance (CA) Mechansim

The core concept behind this mechanism is to prevent network congestion by gradually adjusting the value of the cwnd by applying the Additive-Increase/Multiplicative-Decrease (AIMD) principle. This approach exhibits a more cautious strategy than the slow-start mechanism; it increases the cwnd by $1/cwnd$ such as in NewReno TCP. During the congestion avoidance phase, if packet loss is detected, CCA reduces its cwnd by the multiplicative decrease factor and initiates a new epoch using the same mechanism, as illustrated in Figure 2.1. Additionally, the CCA lowers its cwnd to the Initial Window to initiate a new slow-start phase in the event of an RTO expiration.

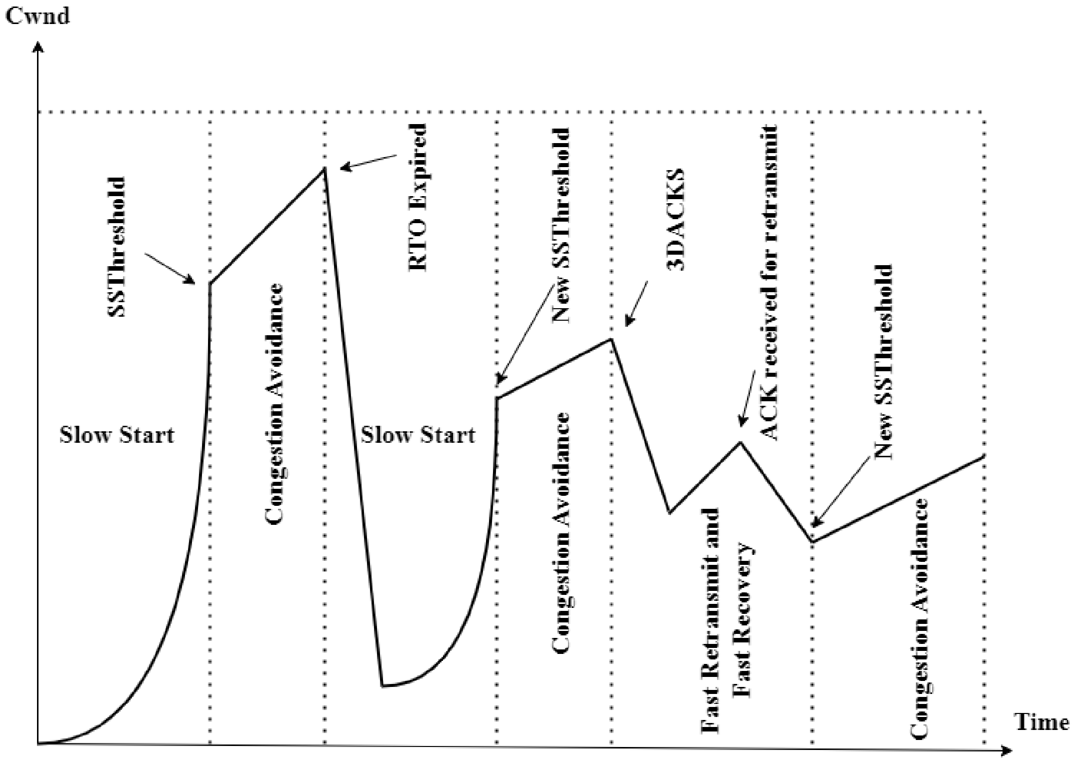


Figure 2.1: Transmission Contol Protocol Behevior

2.5 Congestion Control Algorithm (CCA) Approaches

TCP ensures the reliable delivery of data packets over end-to-end connection without relying on explicit feedback from the underlying network. Recognized as an end-to-end communication, TCP relies solely on communication between the source and destination. The CCA of TCP at source side uses two primary explicit feedback mechanisms to regulate its cwnd:

- Packet loss signal: This indicates network congestion, typically caused by buffer overflow, and is detected through either reception of three duplicated ACKs (3DACKS) or RTO expiration.

- Round Trip Time (RTT): This metric encompasses the time taken for a packet to travel from source to destination and the time taken from destination to source for acknowledgment. It also considers both propagation and queuing delays.

These explicit feedback mechanisms form the basis for three main congestion control approaches on the sender side of TCP, as illustrated in Figure 2.2. Afanasyev et al. (2010) categorized these approaches into three distinct types, briefly explained in the following paragraphs.

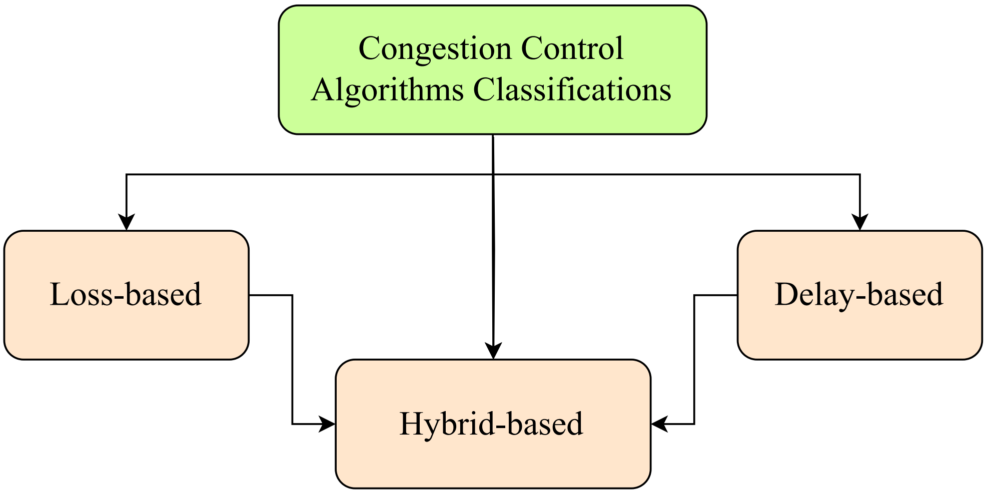


Figure 2.2: Congestion Control Algorithm Classification

2.5.1 Loss-Based CCAs

This strategy relies solely on packet loss signals to adjust the ITR. The CCA incrementally increases its cwnd unless a packet loss is detected. Upon loss detection, the cwnd is reduced through a multiplicative decrease factor. However, a notable drawback is the potential for unnecessary cwnd reductions when identified losses are not congestion-related. This scenario is prevalent in wireless, mobile, satellite, and long-distance networks, where packet losses may stem from wireless channel error rates, face NLoS conditions, or signal attenuation as in mmWave cellular networks.

2.5.2 Delay-Based CCAs

In this classification, the cwnd adjustment is tied to RTT, with TCP fine-tuning its ITR exclusively in response to variations in RTT. As long as the RTT remains low, TCP progressively increases ITR but proportionally decreases it when the RTT rises. This method excels in environments where the underlying network experiences significant delay fluctuations, prevalent in wireless, mobile, satellite, and long-distance networks. However, the stability of this approach falters in scenarios marked by frequent route changes and NLoS conditions, such as in mmWave cellular networks.

2.5.3 Hybrid CCAs

This approach integrates loss and delay-based classifications to achieve heightened scalability, robustness, and efficiency, as highlighted by Katto et al. (2008b, a). Many hybrid CCAs operate as multi-mode switching mechanisms, featuring reactive loss-based and proactive delay-based modes. In these CCAs, the delay-based mode is engaged to leverage residual bandwidth capacity without detected packet losses, while the loss-based mode is activated when losses are detected. Additionally, these algorithms rely on the observed RTT to transition from the loss-based mode back to the delay-based mode.

2.6 An Overview of TCP over MmWave Cellular Networks

The advent of cellular networks has highlighted several inadequacies in existing TCP variants, preventing them from effectively leveraging the available bandwidth in mmWave networks. Firstly, mobility between User Equipment (UE) and base stations (gNB) often results in Non-Line-of-Sight (NLoS) conditions. This leads to a decrease in the sending rate, regardless of whether the loss is due to actual congestion or non-congestion states (Poorzare and Augé, 2020). The congestion control algorithm's response to the absence of acknowledgements and degrading throughput and leaving available bandwidth underutilized. Secondly, mmWave cellular networks, as wireless networks, are prone to loss error rates. TCP's sensitivity to error loss deals with actual loss and reduces its congestion window ($cwnd$) after each error loss (Polese et al., 2017a; Xie et al., 2023). For instance, during the congestion avoidance phase, TCP cautiously probes network bandwidth, typically increasing its $cwnd$ by only one packet per Round-Trip Time (RTT) if no packet loss is detected. In case of error loss, the increase of $cwnd$ will not be tailored to the underlying network. Thirdly, a small buffer size can lead to packet drops in the network. This negatively impacts TCP performance by necessitating retransmissions of lost packets, which underutilizes the network bandwidth and increases latency. Last but not least, server location impacts TCP performance due to increasing RTT, resulting in varied TCP performance depending on whether the server is at the edge or remote (Poorzare and Augé, 2020, 2021b). Despite relying on different assumptions and techniques, TCP variants share a common objective: (a) ensuring efficient utilization of network bandwidth, (b) responding promptly to changes in network conditions, and (c) equitably sharing bandwidth with other flows in the network.

The reviewed literature comprehensively explores the performance of various TCP congestion control algorithms (CCA) in the context of 5G mmWave cellular networks, revealing key limitations and various methodological approaches. Most studies rely on simulation environments such as ns-3 or testbeds to evaluate TCP variants, including NewReno, CUBIC, BBR and several proposed adaptive protocols such as FB-TCP, RBBR and mmCPTP. Although these works contribute valuable information about the behavior of TCP under intermittent connectivity, NLoS conditions, and buffer size constraints, many of the proposed solutions lack adaptability to real-time changes in network conditions. A notable gap is the heavy reliance on predefined thresholds and historical data, which limits generalizability in dynamic mmWave networks.

Furthermore, this section provides a comprehensive understanding of the primary challenges faced by TCP in cellular networks by examining various variants of TCP that address various issues, as outlined in Table 2.1. To address the challenge of cellular networks on TCP performance, several variants of TCP have been developed, including D-TCP (Kanagarathinam et al., 2018), TCP-Drinc (Xiao et al., 2019), NexGen-TCP (Kanagarathinam et al., 2020), FB-TCP (Poorzare and Augé, 2021a), TCP-FLASH (Guo and Lee, 2021), mmS-TCP (Kim and Cho, 2022), RBBR (Haile et al., 2022), and Yinker-BBR (Xie et al., 2023). Despite the efforts of the researcher, none of these TCP variants fully leverages the available bandwidth in mmWave networks. The performance of these TCP variants remains suboptimal in cellular networks, primarily due to mmWave limitations, including obstacle penetration issues that result in NLoS conditions between the gNB and UE. Additionally, packet error loss, and whether the server is located at the edge or in the cloud (Mezzavilla et al., 2018; Zhang et al., 2019; Jeddou et al., 2023).

2.7 An Overview of Benchmark TCP Protocols

The benchmarking protocols are summarized in Table 2.1, and they are used in this thesis to evaluate the proposed protocols. NewReno was selected because it was the default TCP in many applications for many years and served as the base for designing subsequent protocols. CUBIC is chosen for its role as the default algorithm in Linux operating systems. HighSpeed is included due to its aggressive mechanisms to adjust the sending rate, making it suitable for networks with high BDP. BBR is selected as it is the new default protocol on Google servers and represents the latest era protocol. FB-TCP is evaluated alongside state-of-the-art protocols as it is specifically designed to enhance TCP performance over 5G mmWave cellular networks. The subsections below provide a more detailed discussion of these benchmarking protocols.

2.7.1 NewReno Transmission Control Protocol

NewReno (Floyd et al., 2004) is a modification of the TCP Reno protocol initially developed by Floyd and Henderson in 1999. Subsequent modifications by Floyd et al. in 2004 and Henderson et al. in 2012 (Henderson et al., 2012) aimed to address an issue in Reno known as FastRecovery. This problem became apparent due to multiple packet losses, which severely impacted Reno's performance in heavily congested networks.

In NewReno, the exit from the FastRecovery state is permitted only when all data from the initial $cwnd$ are acknowledged. The protocol achieves this by distinguishing between partial ACKs and new ACKs, sensing the former as an indication of additional losses in the initial $cwnd$. Specifically, the reception of a new data ACK signals the successful delivery of all data packets sent before the loss detection. At the same time, a partial ACK indicates losses within the initial $cwnd$.

NewReno is merely a sender-side modification, which increases and decreases cwnd based on Additive increase/Multiplicative decrease as Figure 2.3 shows. The $cwnd$ is increased by $1/cwnd$ in the congestion avoidance phase, as shown in Equation (2.2).

$$cwnd = cwnd + \frac{1}{cwnd} \tag{2.2}$$

2.7.2 HighSpeed Transmission Control Protocol

In 2003, Floyd introduced a novel TCP variant, named HighSpeed-TCP (Floyd, 2003), explicitly crafted to handle large $cwnd$ sizes. This variant aims to address the shortcomings of standard TCP, which has shown improved performance in high-speed networks. HighSpeed-TCP functions as a loss-based CCA without altering the behavior of standard TCP, thus mitigating the risk of congestion collapse. For instance, HighSpeed-TCP operates as a modification on the sender side, dynamically adjusting cwnd by

$\alpha/cwnd$ for increases and $\beta/cwnd$ for decreases.

The values of cwnd span from 1 to 70, with the parameter $\alpha/cwnd$ and $\beta/cwnd$ ranging from 0.5 to 0.1. This variation is contingent on the specific range of $cwnd$, varying from below or equal to 38 packets to greater than 84k packets. This adaptive approach caters to diverse network conditions, ensuring optimal performance (Afanasyev et al., 2010; Lar and Liao, 2013). HighSpeed updates its $cwnd$ after receiving each ACK in an RTT by $\alpha(cwnd)$, as shown in Equation (2.3), if congestion is not detected, and otherwise decreases the $cwnd$ by $\beta(cwnd)$ if the congestion is detected, as shown in Equation (2.4). Accordingly, Figure 2.4 shows the behavior of HighSpeed cwnd.

$$cwnd = cwnd + \alpha/cwnd \tag{2.3}$$

where $cwnd$ is the congestion window, α is a parameter in $\alpha/cwnd$ value to increase the congestion window in HighSpeed CCA.

$$cwnd = cwnd - \beta/cwnd \tag{2.4}$$

where $cwnd$ is the congestion window, β is a parameter in $\beta/cwnd$ value to decrease the congestion window in HighSpeed CCA.

2.7.3 CUBIC Transmission Control Protocol

CUBIC (Ha et al., 2008), introduced by Ha et al. in 2008, currently stands as the default TCP CCA in many Linux operating systems. This innovative approach deviates from the linear cwnd increase in traditional TCP variants, opting for a cubic function to enhance scalability, particularly over BDP networks. CUBIC's foundation lies in modifying its earlier work, BIC-TCP. Ha et al. adapted the cubic function for cwnd increase from the H-TCP approach, as expressed in Equation (2.5). Figure 2.5 illustrates the cwnd growth of CUBIC algorithm.

$$cwnd = cwnd + C \times \left(\Delta - \sqrt[3]{\frac{\beta \times cwnd}{C}} \right) \tag{2.5}$$

Where C is a preset constant, β is the multiplicative decrease factor, and Δ represents the elapsed time since the last loss.

2.7.4 BBR Transmission Control Protocol

Bottleneck Bandwidth and Round-Trip Time (BBR) (Cardwell et al., 2016), introduced by the Google research group, operates as a model-based CCA. Widely embraced in prominent applications such as Netflix and YouTube, BBR aims to optimize link utilization by estimating bottleneck bandwidth and RTT to maintain an uncongested bottleneck queue, as shown in Figure 2.7. BBR follows Kleinrock's optimal operating point (Kleinrock, 2018) as shown in Figure 2.6 to adhere to these principles. BBR calculates the receiver's delivery rate at the sender for every arriving ACK and uses this in the Bottleneck Bandwidth (BtlBW) estimation.

The algorithm employs a new delivery-rate estimation mechanism proposed alongside the CCA to control the increase/decrease of cwnd as shown in Figure 2.8. BBR removes noisy delivery measurements that could lead to an underestimation of the BtlBW by passing them through a max filter over 10 RTTs. The final BtlBW estimate is computed as the minimum of the max-filtered delivery rates and the sending rate. BBR algorithm probes for available bandwidth in a bandwidth probing cycle, referred to as ProbeBW, by increasing the pacing rate by 25% every 8th RTT. To probe for the minimum RTT, the algorithm significantly reduces the amount of in-flight data every RTprop, which is set to 10 seconds, provided a lower minimum RTT estimate has not been detected within this interval. The CCA enters the ProbeRTT state when it reduces in-flight data to detect changes in the minimum RTT.

Another important component of BBR is the mechanism to counter the effect of ACK aggregation. This mechanism estimates the level of aggregation by using the difference between the amount of ACKed data and the expected amount of data to be ACKed. Similar to the BtlBW computation, a max filter is applied over multiple RTTs to add an extra amount of data to the computed cwnd. This mechanism is closely tied to the BtlBW estimation and could result in longer packet delays as it implies injecting possibly excessive amounts of data into the network (Haile et al., 2022).

While successfully implemented in Google and YouTube infrastructure, BBR faces challenges in parameter estimation, requiring multiple cycles and resulting in significant delays. Additionally, fairness concerns arise when multiple flows utilizing different TCP protocols compete within the network (Atxutegi et al., 2018; Ji et al., 2023; Khorov et al., 2023; Gomez et al., 2024).

2.7.5 Fuzzy logic Transmission Control Protocol

In 2021, Reza Poorzare and Anna Calveras Augé proposed FB-TCP (Poorzare and Augé, 2021a). They aimed to address the challenges encountered by TCP in 5G mmWave networks by leveraging Fuzzy logic and introducing novel features and parameters. The functionality of FB-TCP is rooted in dividing the network into multiple sub-states and making decisions based on the current state. The primary objective of this clustering is to establish a spectrum of network segments representing various conditions, ranging from non-desirable

Table 2.1: An overview of TCP protocols over mmwave cellular networks

No.	Ref.	Article Title	Features	Compared Schemes	Evaluation Method	Limitations
[1]	(Floyd et al., 2004)	NewReno	It offers a Fast Recovery mechanism to overcome the issue of multiple losses in the Reno Algorithm.	Reno	Simulation	NewReno cannot distinguish the packet loss cause
[2]	(Floyd, 2003)	HighSpeed	Additive increase steps and multiplicative decrease factors based on the congestion window size, and Limited Slow-Start.	NewReno	Simulation	HighSpeed relies on predefined values to adjust the cwnd size, and this assumption does not fit with the nature of wireless networks.
[3]	(Ha et al., 2008)	CUBIC	The congestion window control as a cubic function of the time elapsed since the last congestion event.	BIC	Simulation	CUBIC depends on packet loss only, which results in an underutilization of bandwidth over high-BDP networks.
[4]	(Cardwell et al., 2016)	BBR	Adjusting transmission rate around Kleinrock's optimal operating point by optimizing the bandwidth and RTT.	Westwood	Simulation	High packet retransmission and fairness issues arise when it competes with loss-based TCP, also overestimating the bottleneck bandwidth, resulting in RTT fluctuations
[5]	(Poorzare and Augé, 2021a)	FB-TCP	Fuzzy rules are applied during the congestion avoidance phase of the protocol to adjust the sending rate and mitigate the effects of congestion.	Fuzzy rules	Simulation	FB-TCP relies on predefined settings, which may overestimate or underestimate the available bandwidth of dynamic cellular networks.

Table 2.1: Continued

No.	Ref.	Article Title	Features	Compared Schemes	Evaluation Method	Limitations
[6]	(Pieska and Kassler, 2017)	TCP Performance over 5G mmWave Links - Tradeoff between Capacity and Latency	Analysis of various TCP variants, performance over intermittent mmWave links, and impact of bufferbloat	Evaluation using ns-3 with real Linux TCP stacks and study of different TCP variants under various conditions	Simulation	Issues with fairness, slow recovery after NLOS periods, and bufferbloat impact on latency and throughput
[7]	(Polese et al., 2017b)	TCP in 5G mmWave Networks: Link Level Retransmissions and MP-TCP	Evaluation of various TCP variants over mmWave links, and the impact of lower-layer retransmissions	Uses ns-3 with real Linux TCP/IP stack, and analysis of link-layer retransmissions and multipath TCP performance	Simulation	Impact of link-level retransmissions on throughput and latency, and challenges with multipath TCP congestion control algorithms
[8]	(Polese et al., 2017b)	Advanced 5G-TCP: Transport Protocol for 5G Mobile Networks	Presented 5G-TCP based on HSTCP protocol	Implement the protocol in NS2 and compare its performance with HS TCP and Reno protocols	Simulation	The protocol relies on predefined value settings and may not respond to the network conditions
[9]	(Atxutegi et al., 2018)	On the Use of TCP BBR in Cellular Networks	Investigates the behavior of TCP BBR in real-world mobile networks as well as in emulated environments	Compares TCP BBR with TCP NewReno and TCP CUBIC in live mobile networks and through emulations, uses MONROE testbed and Aeroflex 7100 LTE emulator	Testbed	Fairness issues in specific conditions; TCP BBR flows do not always share bandwidth fairly

Table 2.1: Continued

No.	Ref.	Article Title	Features	Compared Schemes	Evaluation Method	Limitations
[10]	(Mateo et al., 2019)	Analysis of TCP Performance in 5G mmWave Mobile Networks	Examines effects of packet loss-based, delay-based, and hybrid congestion control protocols	Analyze the behavior of TCP in mmWave networks and study its impact on system-level performance	Simulation	Protocols like CUBIC experience difficulties during extended NLoS periods
[11]	(Xiao et al., 2019)	TCP-Drinc: Smart Congestion Control Based on Deep Reinforcement Learning	Protocol based on deep reinforcement learning	Evaluated with NewReno, CUBIC, Hybla, Vegas, and Illinois	Simulation	Relies on historical data, which may not reflect future conditions due to fast-changing networks
[12]	(Kanagarathinam et al., 2020)	NexGen D-TCP: Next Generation Dynamic TCP Congestion Control Algorithm	Estimates bandwidth to dynamically adjust cwnd	Compared with BBR, CUBIC, Reno, Westwood, Tahoe, and CLTCP	Simulation and live experiment	Overestimating bandwidth leads to incorrect cwnd calculation
[13]	(Kanaya et al., 2020)	A Study on Performance of CUBIC and BBR in 5G Environment	Compares CUBIC and BBR in real 5G environments	Measures throughput and RTT for both protocols in indoor and outdoor conditions	Real experiment	Buffer overflows require long-duration measurement for accurate analysis
[14]	(Scharnitzky et al., 2021)	Comparison of TCP SIAD and TCP BBR Congestion Control in Simulated 5G Networks	Proposes SIAD-TCP	Comparison between SIAD and BBR in 5G mmWave	Simulation	Performance drops with small RLC buffer and fast LoS-NLoS transitions
[15]	(Siddiqui and Chau, 2021)	Analysis of Transport Layer Congestion Control Algorithms over 5G Millimeter Wave Networks	Evaluates various TCP versions (e.g., NewReno, YeAH, Hybla, Westwood, Vegas)	Performance comparison across protocols in 5G mmWave	Simulation	High penetration loss and inefficiency with high RTT and packet loss

Table 2.1: Continued

No.	Ref.	Article Title	Features	Compared Schemes	Evaluation Method	Limitations
[16]	(Poorzare and Augé, 2021b)	How Sufficient is TCP When Deployed in 5G mmWave Networks Over the Urban Deployment?	Performance analysis of TCP variants in urban scenarios	Evaluation in diverse 5G mmWave scenarios	Simulation	TCP variant performance varies due to urban network dynamics
[17]	(Poorzare and Augé, 2021a)	FB-TCP: A 5G mmWave Friendly TCP for Urban Deployments	FB-TCP designed for urban mmWave challenges	Compared with legacy TCP variants	Simulation	Performance degradation when small buffer is applied
[18]	(Haile et al., 2022)	RBBR: A Receiver-Driven BBR in QUIC for Low-Latency in Cellular Networks	Uses Kalman filter for bandwidth estimation in receiver-driven model	Bandwidth estimation using Kalman filter for improved performance	Testbed	Complex to implement and increases overhead
[19]	(Xie et al., 2023)	Yinker: A Flexible BBR for High-Throughput, Low-Latency Over Wi-Fi and 5G	Adaptive pacing replaces fixed pacing_gain in BBR	Evaluated over Wi-Fi and 5G; adjusts pacing based on congestion/loss	Simulation and testbed	Lack of proof for extending queuing delay thresholds based on RTT variation
[20]	(Netalkar et al., 2023)	mmCPTP: A Cross-Layer Pull-based Transport Protocol for 5G mmWave Networks	Optimized for mmWave via cross-layer pull-based design	Cross-layer optimization and pull-based data transmission	Simulation	Performance highly dependent on cross-layer data; complex implementation

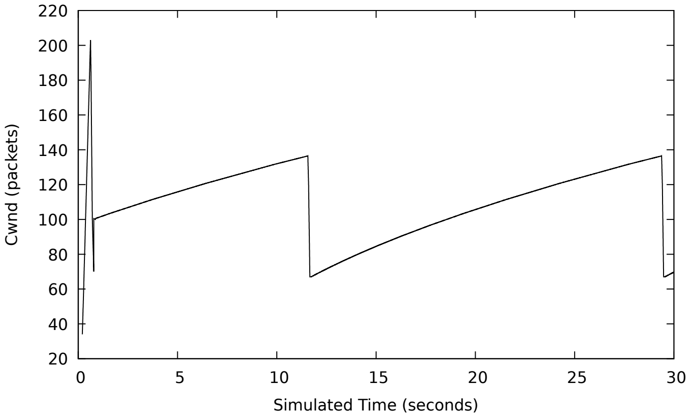


Figure 2.3: Cwnd of NewReno CCA

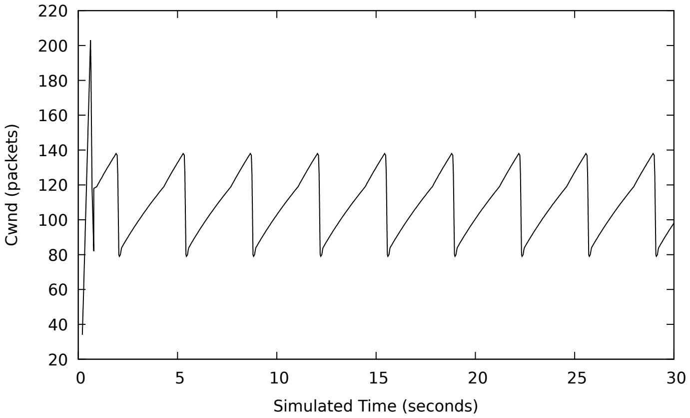


Figure 2.4: Cwnd of HighSpeed CCA

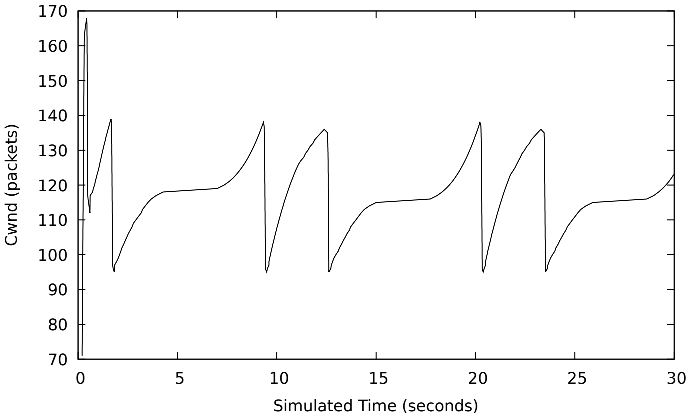


Figure 2.5: Cwnd of CUBIC CCA

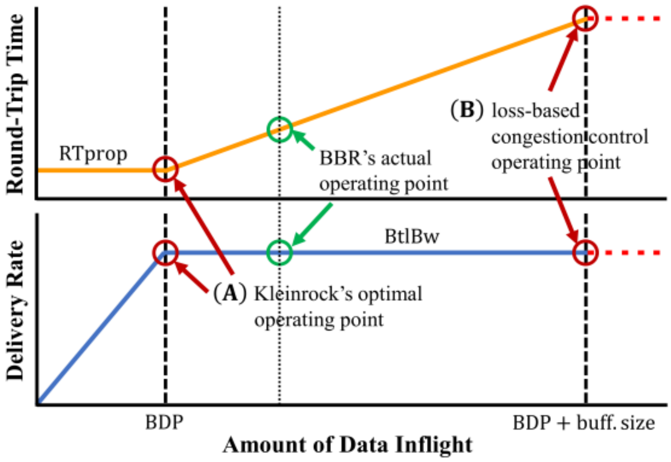


Figure 2.6: Delivery Rate and RTT Vs. Inflight
Source: (Kim and Cho, 2019)

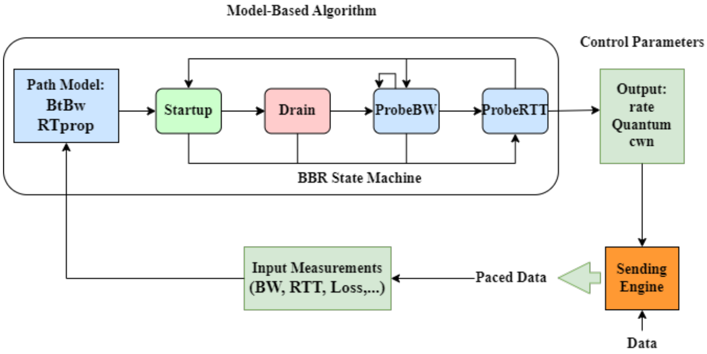


Figure 2.7: BBR Operational States
Source: (Cao et al., 2019)

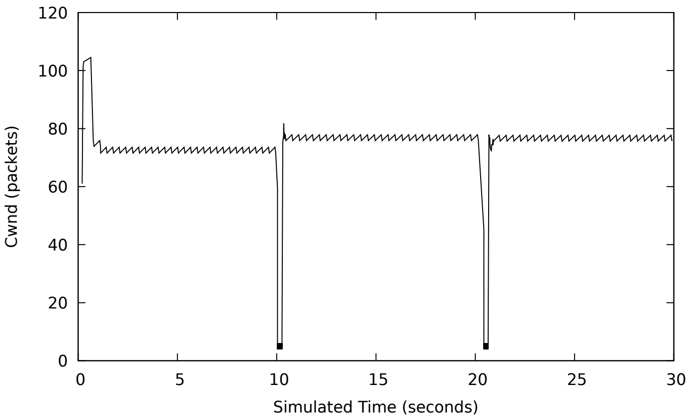


Figure 2.8: Cwnd of BBR CCA

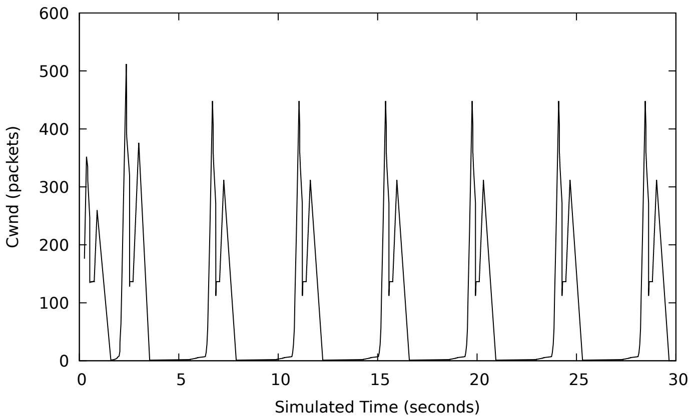


Figure 2.9: Cwnd of FB-TCP CCA

to desirable ones (Poorzare and Augé, 2021a). Figure 2.9 shows the cwnd behavior of FB-TCP CCA.

2.8 Literature Review Analysis

The literature reviewed in this chapter provides a broad understanding of how TCP behaves under various network conditions, with a particular focus on its performance in mmWave cellular networks. Although TCP has long been recognized for its stability and reliability in wired networks, its effectiveness becomes limited in mmWave environments, which are characterized by frequent and unpredictable fluctuations in link quality. Notably, frequent transitions between Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) conditions often result in non-congestion-related losses, which standard TCP mechanisms are not designed to handle efficiently in such scenarios.

Moreover, the literature consistently highlights that limited buffer capacities and high Packet Error Rates (PER) further degrade TCP performance. While several advanced TCP variants have been proposed to improve responsiveness, many rely on fixed parameters or assumptions that are poorly suited to the highly dynamic nature of mmWave cellular networks. As a result, these protocols often misinterpret losses caused by buffer limitations or channel errors as congestion, leading to reduced throughput and increased latency.

Furthermore, although hybrid approaches that combine delay-based and loss-based techniques have shown some promise, their ability to adapt swiftly to real-time variations remains inadequate, especially in cellular-to-cloud communication scenarios. Existing TCP congestion control methods are not well-equipped to handle the complex and rapidly changing Round-Trip Time (RTT) variations present in mmWave cellular networks connected to remote servers.

In summary, the literature reveals a significant research gap: the absence of TCP congestion control algorithms capable of intelligently adapting to the diverse challenges of mmWave cellular networks. These include distinguishing non-congestion-related losses caused by LoS–to–NLoS transitions, coping with limited buffer capacities, managing high PER, and responding effectively to RTT variations.

2.9 Summary

Numerous notable enhancements have been proposed within the literature to address TCP's inherent challenges in mmWave cellular networks. Some researchers have used Loss-based techniques, while others have used Delay-based techniques. Other researchers have combined both techniques, like loss and delay-based approaches, to enhance performance across varied scenarios. Additionally, some have employed RTT and/or bandwidth estimation to regulate TCP's cwnd. Furthermore, modifications to specific parts of the TCP algorithm, like slow-start and congestion avoidance phases, have been explored. In contrast, others focus on fine-tuning existing TCP parameters for superior performance. Despite these efforts, TCP grapples with several issues in mmWave cellular networks, including non-congestion states due to NLoS challenges, small buffers, packet error loss, and server location in achieving full bandwidth utilization of mmWave cellular networks.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

Although TCP remains the dominant transmission control protocol on the Internet, current studies indicate a decrease in bandwidth utilization, a high data rate, and low latency, particularly on mmWave cellular networks. Numerous researchers have addressed this issue in recent years, proposing various solutions to enhance TCP performance. The main focus of studies has been on making TCP work better by understanding the features of the networks it runs on. Even with these efforts, TCP still faces challenges, which are compounded by the fast changes in network technology, like using mmWave in cellular communications.

In this thesis, a simulation scenario is performed to evaluate and examine TCP variants, including the proposed TCPs, over mmWave cellular networks in urban areas. The scenario simulates trees and buildings to replicate both Line of Sight (LoS) and Non-Line of Sight (NLoS) conditions, as well as static and dynamic positions, including driving scenarios. Packet error loss and buffer size are considered in the scenario. Also, the location of a server is considered, including whether the server is an edge or remote server. The evaluation of TCP protocols involves measuring key performance metrics such as congestion window fluctuations, throughput, latency, and cumulative distribution function for the throughput and the latency.

3.2 Research Framework

The research framework of this study is presented in Figure 3.1. It is a chronological illustration of the various stages involved in the research, beginning with a review of existing TCP protocols, followed by problem formulation, the design of the proposed algorithms, and performance evaluation metrics. This framework provides a comprehensive perspective on the research approach, explaining the methodology adopted in this thesis.

The framework of this thesis is summarized in the following:

- Implement the benchmark protocols in the network simulator (ns-3) and evaluate them in a simulation scenario representing mmWave cellular networks. Furthermore, different buffers and different Packet Error Rates (PERs) are included in the simulation scenario.
- Formulate the research problem based on the insights from the literature and the benchmark protocols evaluation.
- Propose the RTTV protocol to mitigate non-congestion-related issues and improve TCP performance in mmWave cellular networks. Implement the protocol in ns-3, evaluate it in the simulation scenario, and compare its performance, measured in terms of throughput and latency, with that of the benchmark protocols.
- Propose the MSS protocol to address small buffer sizes and packet error issues, to improve TCP performance in mmWave cellular networks. Implement the protocol in ns-3, evaluate it in the simulation scenario, and compare its performance, throughput, and latency against the benchmark protocols.
- Propose the MRVHS protocol to mitigate the variation in RTT in mmWave cellular-to-cloud networks, thus improving TCP performance in such environments. Implement the protocol in ns-3, evaluate it in the simulation scenario, and compare its results with the results of the benchmark protocols in terms of throughput and latency.

3.2.1 Benchmark TCP Protocols

At this stage, a review of recent TCP protocols is carried out. The primary purpose is to establish a benchmark that can be used to effectively evaluate the proposed TCP protocols against specific QoS performance metrics. Due to their widespread popularity, this thesis extensively explores and utilizes NewReno, HighSpeed-TCP, CUBIC, and BBR for benchmarking in all experiments.

In this paper, FB-TCP is chosen as one of the benchmark protocols to evaluate the proposed protocols because it is a recent protocol and is specifically designed to address TCP issues in 5G mmWave cellular networks. CUBIC is the default TCP for the Linux kernel.

However, the default protocol in Google applications is recognized for video applications, especially streaming services such as YouTube and Netflix. Furthermore, it is a benchmark protocol that has been utilized in numerous studies such as (Zhang et al., 2016; Na et al., 2019; Zhang et al., 2019; Poorzare and Augé, 2021b,a; Khorov et al., 2023; Lim et al., 2023).

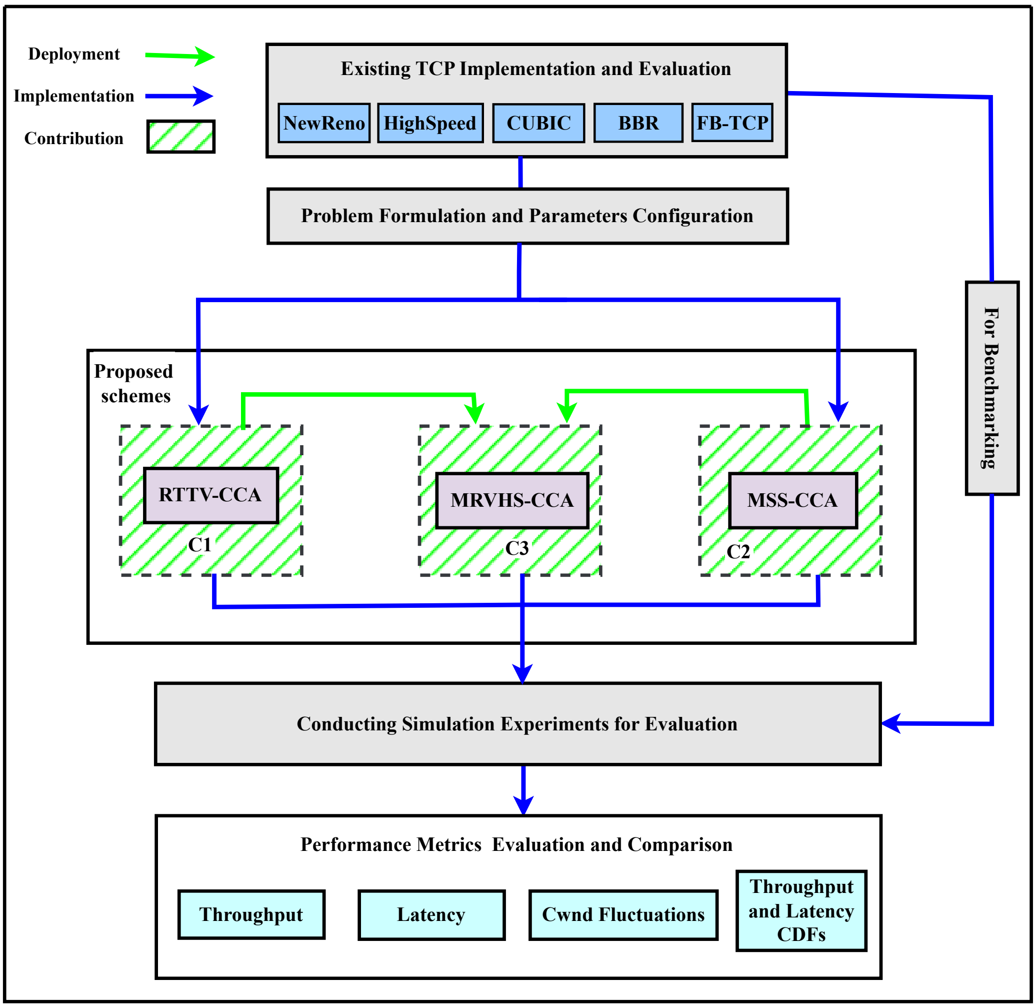


Figure 3.1: Research Framework

3.2.2 Software and Hardware Tools

A comprehensive assessment is necessary when designing a CCA or refining a specific element within a network. This evaluation can be effectively carried out through experimental deployments, analytical modeling, or simulation-based approaches such as the ns-3 network simulator. Ns-3 is a discrete-event network simulator specifically designed for research and education. As freely available software licensed under the GNU GPLv2 license, it is accessible for research, development, and practical applications. Version 3.35 of ns-3 is utilized in this research to simulate and evaluate the performance of the proposed algorithms.

In brief, the ”src” directory in ns-3 simulator contains many modules, such as antenna, network, buildings, and propagation. The Internet module is one of these modules. It includes many files for algorithms, such as TCP algorithm files, which contain both source code and header files. The ns-3 module supports a fully bidirectional TCP with connection setup and close logic. Several congestion control algorithms are supported, with CUBIC as the default. Additionally, the module supports NewReno, Westwood, Hybla, HighSpeed, Vegas, Scalable, Veno, Binary Increase Congestion Control (BIC), Yet Another HighSpeed TCP (YeAH), Illinois, H-TCP, Low Extra Delay Background Transport (LEDBAT), TCP Low Priority (TCP-LP), Data Center TCP (DCTCP), and Bottleneck Bandwidth and Round-trip propagation time (BBR).

The installation of ns-3 is carried out on the Ubuntu 20.04.4 Linux operating system. Ubuntu and ns-3 are implemented on a DELL OPTIPLEX 7010 machine, equipped with an Intel Core i5-3470 CPU @ 3.20GHz processor and 8GB of RAM. The experiments were carried out in the Wireless, Mobile and Quantum Computing Laboratory, A2.18, 2nd floor, Block A, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. Furthermore, Gnuplot software version 5.4 is installed to create graphs to compare the results of the proposed protocols with the results of the benchmarking protocol in terms of throughput, latency, and congestion window fluctuations.

3.2.3 Implementation of The Proposed CCAs in Ns-3

Simulation experiments were carried out in this study using the ns-3 and experiments procedures as described in Figure 3.3. During this phase, to evaluate the performance of existing and newly proposed protocols, we integrated the proposed algorithms (RTTV, MSS, MRVHS) into the Internet module within the ns-3 network simulator, as shown in Figure 3.2. This integration is a crucial step in assessing how these algorithms function within a simulated network environment of mmWave cellular networks, allowing for a comprehensive analysis of their effectiveness and impact on TCP performance. Furthermore, the notations used are listed in Table 3.1.

The primary objective was to assess the performance of the proposed algorithms by juxtaposing their outcomes against those of established state-of-the-art protocols employed as benchmarks. These experiments were conducted explicitly within a simulated urban cellular network environment as the testing ground for evaluating the RTTV-based, MSS-based, and MRVHS-based CCAs.

3.3 Simulation Setup and Deployment Topology

The examined algorithms have undergone evaluation within a deployment topology, as in Figure 3.4. An UE establishes a connection with a gNB operating at 28 GHz with a bandwidth of 1 GHz, positioned at a height of 15 meters. This gNB is linked to a server operating at a sending rate of 1 Gb/s. Initially, the UE pauses and starts walking while maintaining a LoS connection with a gNB. Subsequently, the user encounters trees, which mimic small obstacles and create NLoS conditions. The user pauses to simulate a static NLoS scenario with a tree for four seconds. Afterwards, the user resumes movement, regaining LoS with the base station until arriving at a building, mimicking a large obstacle. The user then pauses behind the building for four seconds to simulate NLoS conditions. Finally, the user moves towards a car, drives away from the base station, and stops after reaching a certain distance.

In the scenario, the performance of the studied TCP is assessed using various buffer sizes. Specifically, 0.25MB, 2.5MB, and 20MB are chosen to simulate small, BDP, and large buffer sizes, respectively. Additionally, four PERs of 10^{-7}, 10^{-8}, and 10^{-9} and zero PER are applied to examine their impact alongside each buffer size on TCP variants performance.

The simulation parameters are provided in Table 3.2. For instance, the simulation parameters used for this research are provided in the benchmark paper (Poorzare and Augé, 2021b).

To ensure the accuracy of the results, the simulation experiments are repeated 3 times, and the average values of Key Performance Indicators (KPI), such as the congestion window fluctuations, throughput, delay, and CDF of throughput and latency are presented for each set of parameter setups. For a better understanding, Section 3.5 defines the performance metrics used in this thesis.

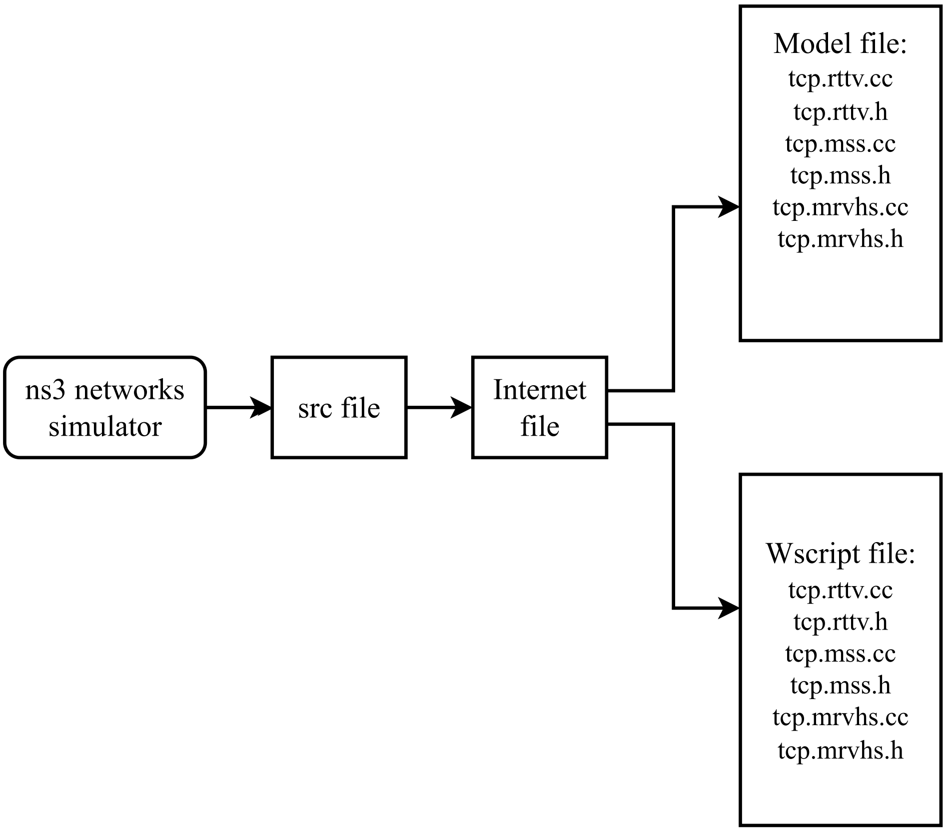


Figure 3.2: Implementation of the Proposed CCAs in NS-3

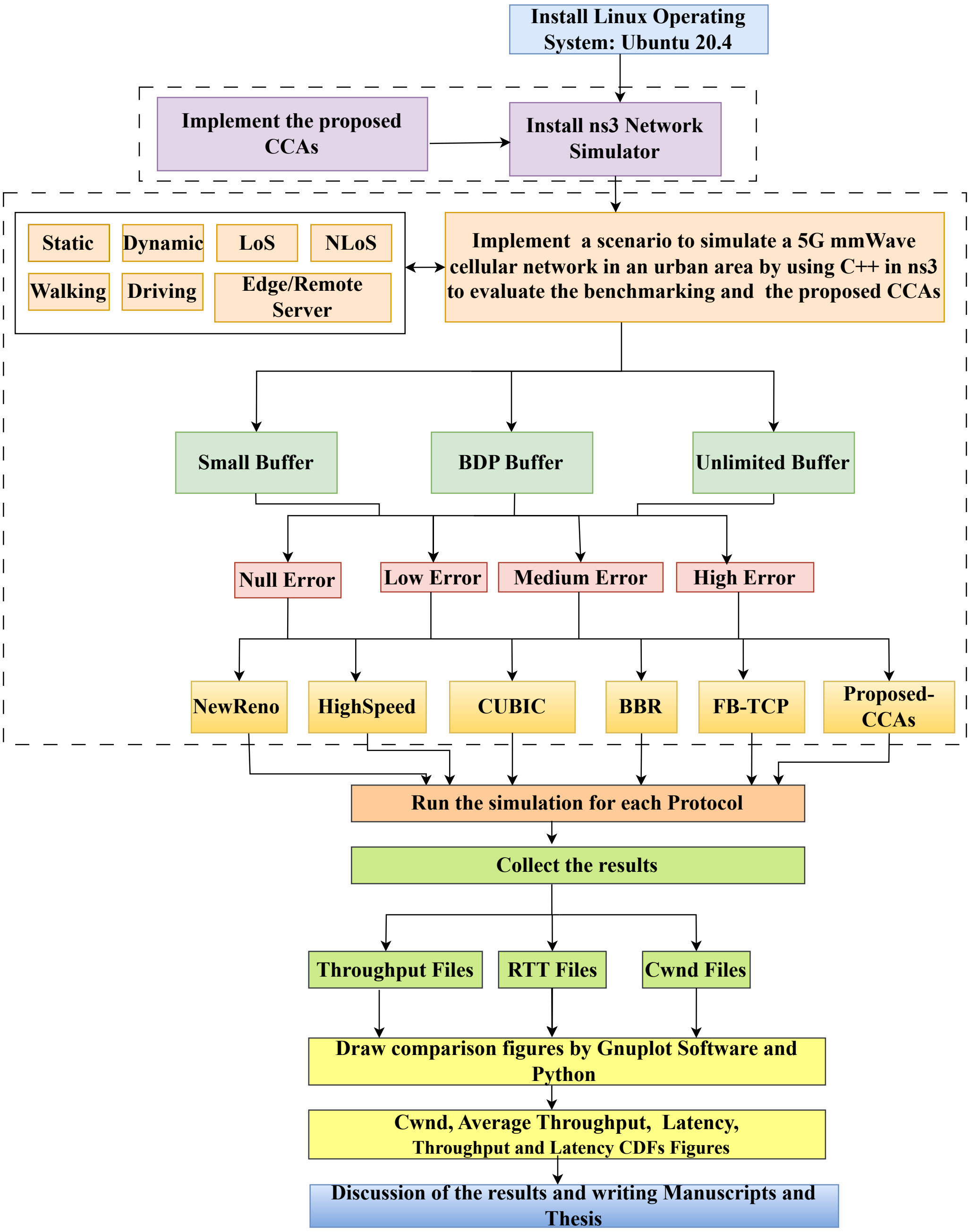


Figure 3.3: The Experimental Procedures

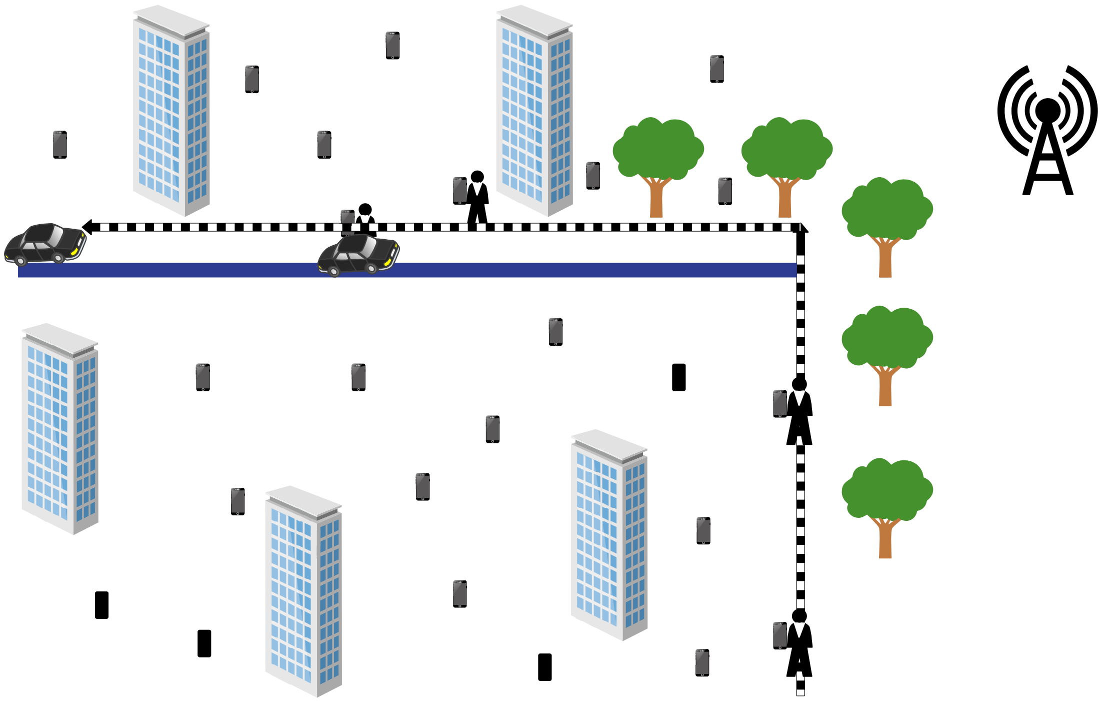


Figure 3.4: Simulation Experiments Topology

3.3.1 Key Performance Indicators

During this phase, the evaluation of key Performance Indicators (KPI) involves measuring three key performance metrics: cwnd fluctuations, throughput, and latency. These metrics are systematically measured for the proposed algorithms and established state-of-the-art protocols. The purpose is to facilitate a comprehensive comparison and benchmarking process. Section 3.5 defines the performance metrics utilized in this thesis to provide a more transparent comprehension.

3.4 Notations and Definitions

To simplify matters, Table 3.1 lists a set of notations for mathematical equations and algorithms that will be used in the subsequent chapters. Additionally, this section provides definitions that are frequently referenced throughout this thesis. CCA refers to the Congestion Control Algorithm, and TCP refers to the Transmission Control Protocol.

Table 3.1: Notations used

Notation/acronym	Description
TCP	Transmission Control Protocol
CCA	Congestion Control Algorithm
$Cwnd$	Congestion window
RTO	Retransmission-Time-Out
RTT	Round-Trip-Time
RTT_{max}	Maximum Round-Trip-Time
RTT_{cur}	Current Round-Trip-Time
RTT_{base}	Base Round-Trip-Time
$3DupACK$	Three duplicated Acknowledgments
β	Multiplicative decrease after congestion loss
$ssthresh$	Slow-start threshold
$RTTV$-CCA	Round-Trip-Time Variations CCA
MSS-CCA	Maximum segment size CCA
$MRVHS$-CCA	Max Segment Size and RTT Variations High-Speed CCA
KPI	Key Performance Indicator
UE	User equipment
gNB	Base station
LoS	Line of sight
$NLoS$	Non line of sight
PER	Packet error rate

Table 3.2: Simulation parameters

Attribute	Value
Carrier frequency	28GHz
Loss Model	Building Obstacle Propagation Loss Model
Bandwidth	1GHz
TCP protocols	NewReno, HighSpeed-TCP, CUBIC, BBR, FB-TCP, RTTV, MSS, MRVHS
Buffer size	0.25, 2.5, 20 MBytes
RTO	1 second
PER	0, 10^{-9}, 10^{-8}, and 10^{-7}
Simulation time	60 seconds
Operating System	Linux/Ubuntu 20.04
Simulation Program	ns3.35

3.5 Performance Metrics

The primary objective of this study is to enhance TCP's performance over mmWave cellular networks by addressing the challenges of NLoS conditions, small buffers, error loss, and server location. The assessment of TCP performance involves the measurement of KPI metrics, including cwnd fluctuations, throughput, latency, and CDF of throughput and latency.

3.5.1 Congestion Window (Cwnd) Fluctuations

The cwnd is a pivotal element in CCA, and its manipulation directly influences the system's throughput and latency. The measurement of cwnd is quantified by the number of packets within the sending window.

3.5.2 Throughput

Throughput signifies the rate at which data is successfully delivered over a network link from the source to the destination. Typically measured in bits per second (bps) or multiples like Mbps or Gbps. The average throughput is determined by dividing the amount of data received by the receiver by the time taken in seconds (Afanasyev et al., 2010), as shown in Equation (3.1).

$$Througput = \frac{Transmitted\ Data}{Time} \tag{3.1}$$

Where $Transmitted\ Data$ refers to the data sent from one host to another, and $Time$ represents the duration required to transfer the data between the hosts.

Conversely, Instantaneous Throughput is assessed by dividing the current $cwnd$ by the RTT (Afanasyev et al., 2010), as Equation (3.2) shows.

$$Instantaneous\ Throughput = \frac{cwnd}{RTT} \tag{3.2}$$

Where $cwnd$ is the congestion window, representing the number of packets sent from one host to another and acknowledged by the sender within a time interval equal to the RTT (Round-Trip Time).

3.5.3 Latency

Latency signifies the time spent transmitting data from the source to the destination (Afanasyev et al., 2010), as Equation (3.3) shows. It is typically measured in seconds (s) or multiples, such as milliseconds (ms) or microseconds. The latency quantifies the total duration of the data transmission process. Specifically, it is determined as the average value during the transmission period.

$$Latency = Receiving\ time - Sending\ time \tag{3.3}$$

Where the $Sending\ time$ is the time at which data transmission begins from the source host to the destination host, and the $Receiving\ time$ denotes the time at which the data is received by the destination host.

3.5.4 Cumulative Distribution Function (CDF)

Furthermore, the cumulative distribution function (CDF) is used to illustrate throughput and latency results over the simulation time of experiments.

For instance, from a mathematical point of view, the CDF represents the probability that a random variable X is less than or equal to a specific value x. It is commonly denoted as $F(x)$. The CDF of a random variable X is expressed mathematically as $F(x) = P[X \leq x]$.

Therefore, in this research, we incorporated CDF analysis to further explain the throughput and latency results over simulation time, considering varying buffer sizes and PER values. This CDF analysis supports the explanation of the results by providing a detailed view over the simulation time.

3.5.5 Problem Formulation

At this stage, a literature review and a comprehensive evaluation of state-of-the-art protocols are conducted to highlight the weaknesses of TCP variants in mmWave cellular networks, as shown in Figure 3.1. The issues that hinder TCP from achieving superior performance in mmWave cellular networks are identified. Many challenges are found in the environment of cellular networks, including NLoS issues, small buffer sizes, error loss, and server location. Ultimately, the research problems and scope are determined.

3.6 Performance Evaluation of Benchmark TCP Protocols in MmWave Cellular Networks

To investigate the effects of the challenges of mmWave cellular networks on TCP performance, an empirical evaluation of state-of-the-art benchmark TCP variants is performed in mmWave cellular networks. The primary focus is on the impact of NLoS conditions, varying buffer sizes, Packet Error Rates (PERs), and server location on the performance of these benchmark TCP protocols. The scenarios include various challenges, such as blockages to mimic LoS and NLoS conditions, transitions between LoS and NLoS, user mobility (walking and driving), proximity to the base station, moving away from the base station, and changes in movement speed.

The main contributions of this section are:

- Conducting empirical analyses of different TCP variants in mmWave cellular networks considering scenarios that include UE mobility, static conditions, LoS, NLoS, short NLoS, and long NLoS, and variations in the distance between the base station (gNB) and user equipment (UE).

- Analyze the influence of critical factors, including NLoS, buffer size, PER, and server placement, on the performance of TCP variants.
- Identifying the weaknesses of TCP protocols in mmWave cellular networks.

The analysis in this section identifies critical challenges in TCP performance over mmWave cellular networks, offering valuable insights into the feasibility of deploying TCP variants in such environments and highlighting the need for protocol development that addresses the unique issues posed by mmWave cellular networks. The analysis thoroughly explores the complexities of TCP variant performance in mmWave cellular networks. In particular, it examines the impact of non-congestion states (i.e., NLoS conditions), buffer size, error rates, and the effects of server placement on TCP protocol performance. Moreover, various TCP benchmark protocols, including NewReno, HighSpeed-TCP, CUBIC, BBR, and FB-TCP, are evaluated in experiments to assess their behavior when switching between LoS and NLoS, as well as under different buffer sizes and varying error rates in mmWave cellular networks. More details on the deployment topology and simulation parameters are provided in Table 3.2.

The discussion begins by exploring the intricacies of BDP-scaled buffer sizing and PERs, providing a perspective to understand the effect of buffer and PER on TCP performance. This examination is crucial for gaining insights into TCP behavior in practical scenarios, where finite buffer sizes are commonly adjusted based on the BDP in the presence of PER. Subsequently, this chapter traverses the domain of cloud remote server influence, contrasting it against edge server scenarios, thereby investigating the effects of increased Round-Trip-Time (RTT) on TCP performance and stability over mmWave cellular networks. The analysis includes graphs that show how each TCP variant performs in terms of throughput and latency when buffer sizes and PERs are changed in the presence of obstacles that mimic NLoS conditions.

Thus, by breaking down these performance factors, we can understand the balance needed to achieve high throughput while keeping latency low over mmWave cellular networks. This balance is important for TCP protocols to work smoothly in mmWave cellular networks. The goal of this chapter is not only to investigate the performance of benchmark TCP protocols but also to provide insights into how different buffers, PERs, and server placement can affect TCP performance over mmWave cellular networks.

3.6.1 Experimental Setup

Extensive simulation experiments are conducted using the well-known ns-3 network simulator to evaluate the benchmark protocols over mmWave cellular networks. The experimental setup is detailed in Section 3.3, and the specific simulation parameters are described in Table 3.2.

Before analyzing the throughput and latency results, we first examine the impact of switching between line-of-sight (LoS) and non-line-of-sight (NLoS) conditions on the signal-to-interference-plus-noise ratio (SINR), as this factor significantly influences TCP performance in mmWave cellular networks. Figure 3.5 shows how the signal strength of a mobile UE changes during the simulation as it shifts between LoS and NLoS conditions. From Figure 3.5, it is clear that the buildings and trees serve as obstacles. Therefore, NLoS conditions arise when a user is positioned behind an obstacle. Thus, Figure 3.5 illustrates that communication obstacles can weaken the connection between UE and a gNB. The reduced SINRs in NLoS conditions are a major factor contributing to unstable communication in mmWave cellular networks, which can adversely affect TCP performance. Consequently, an increase in obstacles can significantly disrupt TCP functionality.

3.6.2 Validation Results and Discussion

Different applications have different needs; some prioritize higher throughput, whereas others focus on minimal latency. In the upcoming subsections, the study will explore

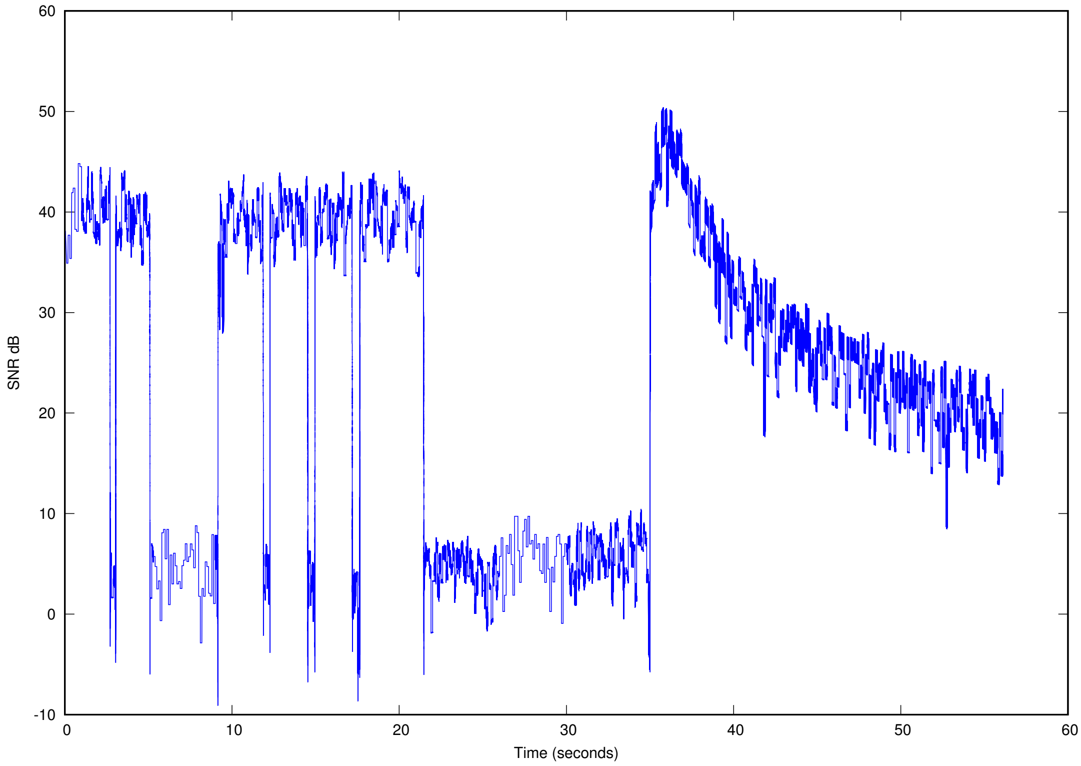


Figure 3.5: Signal-to-Interference-Plus-Noise Ratio

how congestion control mechanisms of TCP variants behave when the buffer size and PER change in the presence of cellular channel fluctuations and examine the reactions of TCP variant protocols to these alterations.

The following subsections carefully examine the findings, aiming to explain in detail how different benchmark TCP variants perform in the constantly changing environment of mmWave cellular networks. Specifically, the analysis looks at how varying buffer sizes, error rates, and server placements affect the throughput and latency of these TCP variants. Three buffer sizes were used: 20MB, 2.5MB, and 0.25MB, representing large, matching BDP, and small buffers, respectively. By using a large buffer, we can better understand how mmWave cellular networks function and avoid buffer overflows during Non-Line-of-Sight (NLoS) states in the simulation. In contrast, a small buffer tends to overflow quickly, especially during NLoS states, which makes it harder to observe the network's behavior in detail.

3.6.2.1 Large Buffer Impact

In this scenario, the buffer size is set to 20 MB to emulate a large buffer size. Figure. 3.6 presents the average throughput of the studied TCP variants, including NewReno, HighSpeed-TCP, CUBIC, BBR, and FB-TCP, which were extensively simulated with varying PER values for each scenario. In addition, it illustrates the throughput achieved by the evaluated protocols based on the applied buffer size and PERs. In error-free channels, the results demonstrate that loss-based TCPs perform similarly, achieving the same maximum throughput due to their mechanism based on packet loss. Moreover, clearly illustrates that the optimal conditions for achieving high throughput with all TCP variants involve a large buffer and zero error rate. However, the performance of loss-based TCPs degrades with high PERs due to frequent packet losses. This leads the protocols to perceive a reduction in bandwidth at the bottleneck. HighSpeed-TCP demonstrates superior performance compared to NewReno, CUBIC, BBR and FB-TCP. On the other hand, BBR and FB-TCP maintain higher throughput across high PER scenarios because they are not categorized as loss-based TCP. BBR aims to operate near Kleinrock's optimal point (Cardwell et al., 2016), modifying the transmission rate accordingly. The marked increase in throughput with BBR is attributable to the bandwidth phase, during which BBR strives to compensate by transmitting at the bottleneck bandwidth rate. As packet losses escalate, BBR's performance is more reliable under high PER conditions. Nevertheless, BBR may not excel in throughput and latency when the UE is mobile and distant from the gNB station. Furthermore, the latency increase stems from BBR's computation of bottleneck bandwidth rather than opting to send additional packets. However, under LoS conditions, the protocol's performance is comparable to FB-TCP's. These findings emphasize the minimal influence of large buffer size on most TCP variants, particularly in scenarios where the PER is negligible.

For more instance, as depicted in Figures 3.7a, 3.7b, 3.7c, and 3.7d, the results illustrate CDFs of the throughput for TCP variants. The data shows that when buffer sizes exceed the BDP and the PER is high, the throughput values for loss-based, such as NewReno, and CUBIC are similar when the high PER is applied. Within this range, the mean throughput values for these protocols vary from 0 to 200 Mb/s. However, as the PER decreases, the throughput significantly increases. For instance, with a null PER, the mean throughput for these protocols reaches approximately 1 Gb/s. In contrast, HighSpeed-TCP generally maintains higher throughput values in most cases of PER except high PER scenario. On the other hand, BBR attempts to send data at the highest possible rate without creating a queue.

Despite having similar throughput, all protocols yield the most equivalent results in terms of latency as shown in Figures 3.8, 3.9a, 3.9b, 3.9c, and 3.9d.

3.6.3 BDP Buffer Impact

This subsection explores buffer size's impact on TCP variants' performance aligned with the BDP. It investigates explicitly scenarios with reduced buffer sizes adjusted to match the BDP. The buffer size is set to 2.5 MB. This configuration comprehensively evaluates the buffer's impact when synchronized with the BDP. Figures 3.10 and 3.11 reveals a significant boost in throughput for HighSpeed-TCP and FB-TCP compared to other TCP variants in most PER scenarios. In high PER conditions, FB-TCP and BBR excel, showcasing relative resilience to packet loss. Among the three loss-based TCP variants, HighSpeed-TCP stands out by utilizing a 100% BDP buffer in most PER conditions, achieving peak throughput. However, as the PER intensifies, the effectiveness of loss-based TCPs diminishes due to their sensitivity to error loss, which causes them to mistakenly assume there is congestion in the network. Moreover, Figure 3.10 illustrates a decline in the performance of loss-based TCP under NLoS conditions, such as when objects like trees or buildings obstruct UE. This degradation is linked to the frequent occurrence of RTOs, prompting a reinitialization from the slow-start phase of the TCP protocol and resulting in diminished protocol efficiency.

The frequent occurrence of RTOs is due to packets that remain unacknowledged within a defined timeframe, leading to substantial fluctuations in the congestion window. The repeated initiation into a slow start exacerbates the decline in performance and impedes the optimal utilization of network bandwidth. Furthermore, reduced buffer sizes harm loss-based TCPs, primarily due to buffer overflows and subsequent packet drops, especially in NLoS circumstances.

Despite these limitations, HighSpeed-TCP maintains a slightly higher throughput than NewReno and CUBIC. On the other hand, Figures 3.12 and 3.13 demonstrate a nearly constant latency across most TCP variants in different PER conditions. This consistency persists despite the trade-off in throughput from reducing the buffer size from large to 100% BDP. The variations in these results are linked to the functioning of the congestion control algorithm employed.

3.6.3.1 Small Buffer Impact

This section explores the influence of reduced buffer size on the performance of TCP variants in mmWave cellular networks. The emphasis lies in understanding how different protocols address the challenges of smaller buffer sizes, especially in situations with varying PERs. Reducing the buffer size to 10% of the BDP unveils, in Figures 3.14 and 3.15, the consistent superior throughput performance of BBR across most various Packet PER scenarios. This demonstrates the efficacy of BBR compared to other TCP variants in addressing challenges posed by smaller buffers, which commonly lead to packet drops and buffer overflows, particularly in NLoS conditions. Conversely, loss-based TCPs exhibit decreased performance in such scenarios due to their susceptibility to packet loss caused by smaller buffers. The buffer size significantly influences the performance of TCP variants in mmWave cellular networks. On the other hand, as shown in Figures 3.16 and 3.17, the latency remains relatively consistent across small buffer sizes and PERs.

3.6.3.2 Impact of Remote Server Placement

In the previous subsections 3.6.2.1, 3.6.3, and 3.6.3.1, TCP variants performance is s analyzed and discussed when the edge server is considered. In this subsection, we discuss the effects of using a remote server over mmWave cellular-cloud network on the performance of benchmark TCP variants. Thus, the remote server is simulated by increasing the delay between the gNB and the server from 10 ms to 40 ms, resulting in a minimum RTT of 80 ms. Consequently, the buffer size is adjusted to 10 MB to match the BDP in the remote server scenario. Figure 3.18 exhibits uniform throughput among all TCP variants when PER is null, with HighSpeed-TCP and CUBIC demonstrating superior performance in most PER scenarios, except in the case of high PER. Figures 3.18 and 3.19 collectively indicate that remote server deployment augments latency and diminishes throughput in mmWave cellular networks. In high PER scenarios, BBR attains higher throughput than other TCP variants due to its mechanism.

It is crucial to note that all TCP variants experience a decline in performance in remote server scenarios involving high PERs, NLoS conditions, and the distance between the UE and the gNB station. This underscores the importance of improving TCP mechanisms to maximize throughput in mmWave cellular networks. On the other hand, the latency remains relatively consistent across different PERs in the remote server scenario as shown in Figures 3.20 and 3.21.

To sum up, a comprehensive study has been conducted to examine how TCP performs over mmWave cellular networks under various conditions to identify TCP issues in mmWave cellular networks. The study investigates the impact of having LoS or NLoS, whether the mobile device is stationary or moving, the distance between UE and the gNB station, and the drive away from the base station. Furthermore, the study highlights the importance of buffer size and error rate on TCP performance as illustrated in Table 3.3 and Table 3.4. The study also demonstrates how the server's location influences TCP performance in terms of throughput and latency.

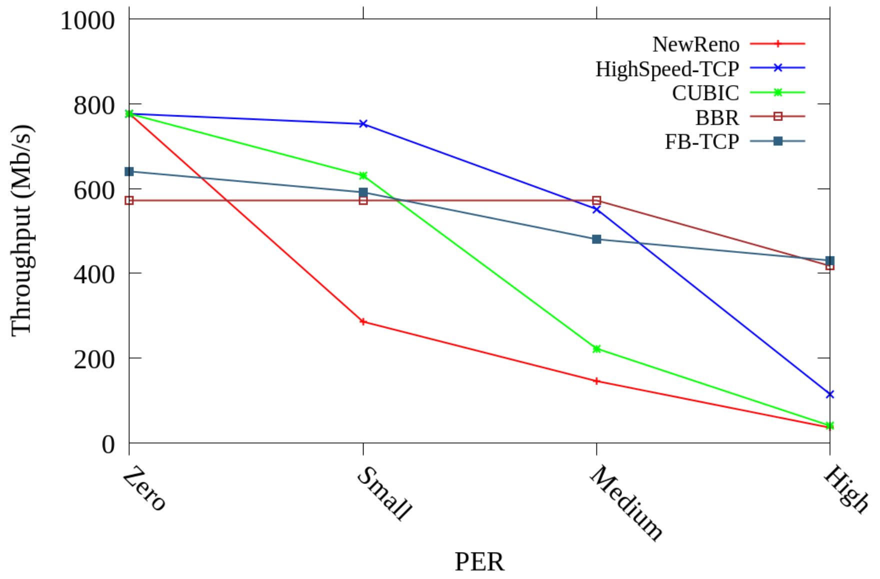
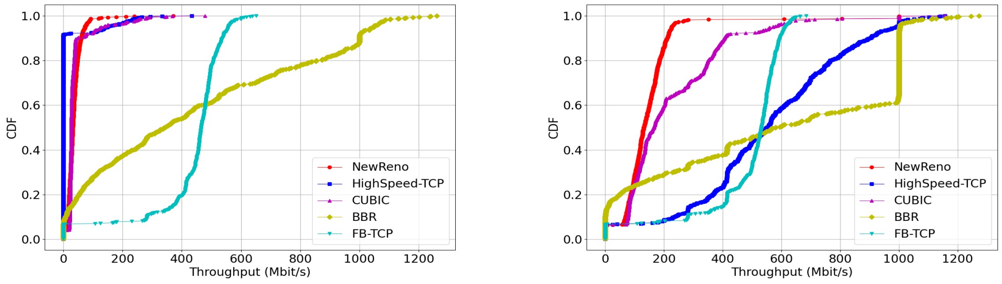
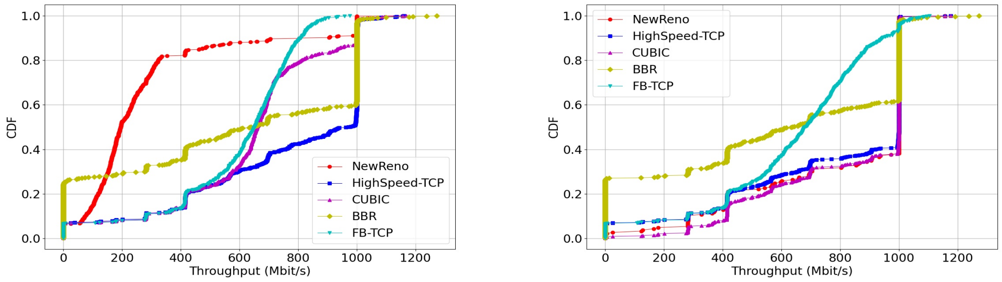


Figure 3.6: Average Throughput of TCP Variants Vs. Different PERs When The Buffer is Large (20MB)



(a) High PER

(b) Medium PER



(c) Small PER

(d) Null PER

Figure 3.7: CDF of Throughput of TCP Variants Vs. Different PERs When The Buffer is Large (20MB)

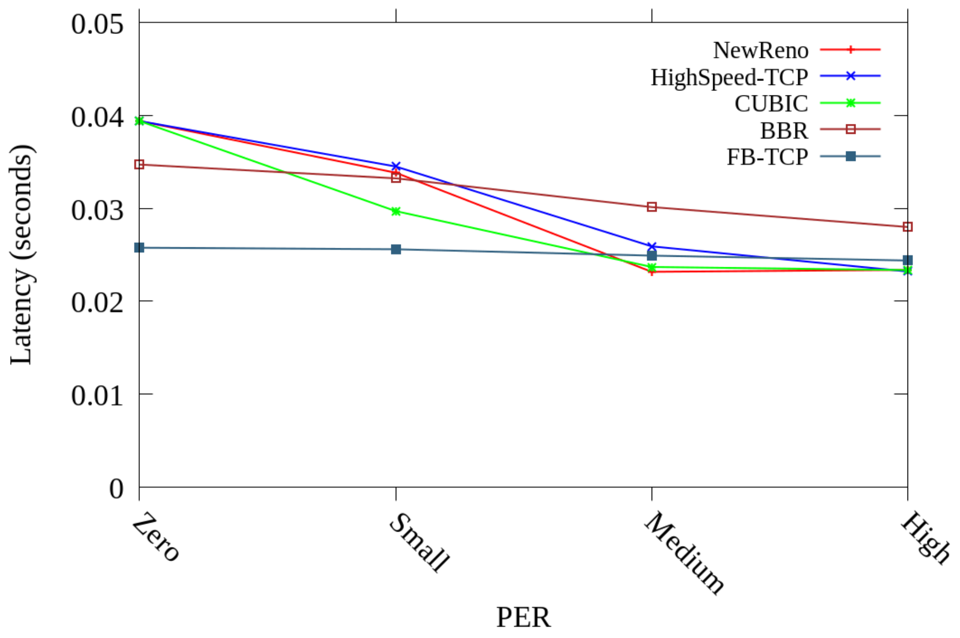
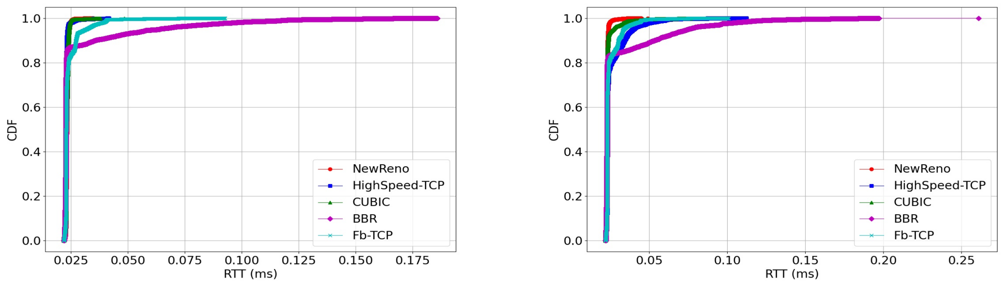
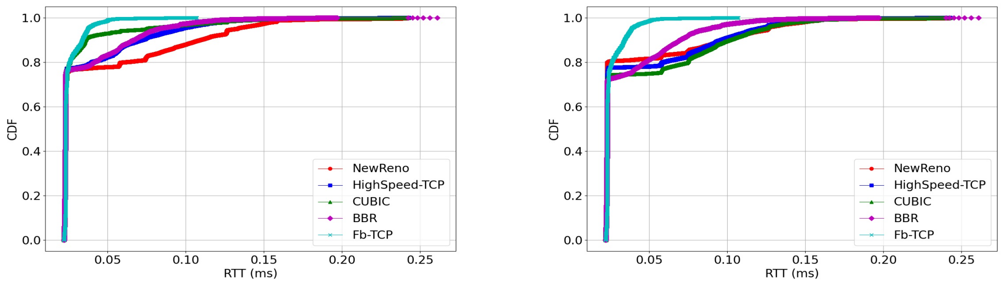


Figure 3.8: Average Latency of TCP Variants Vs. Different PERs When The Buffer is Large (20MB)



(a) High PER

(b) Medium PER



(c) Small PER

(d) Null PER

Figure 3.9: CDF of Latency of TCP Variants Vs. Different PERs When The Buffer is Large (20MB)

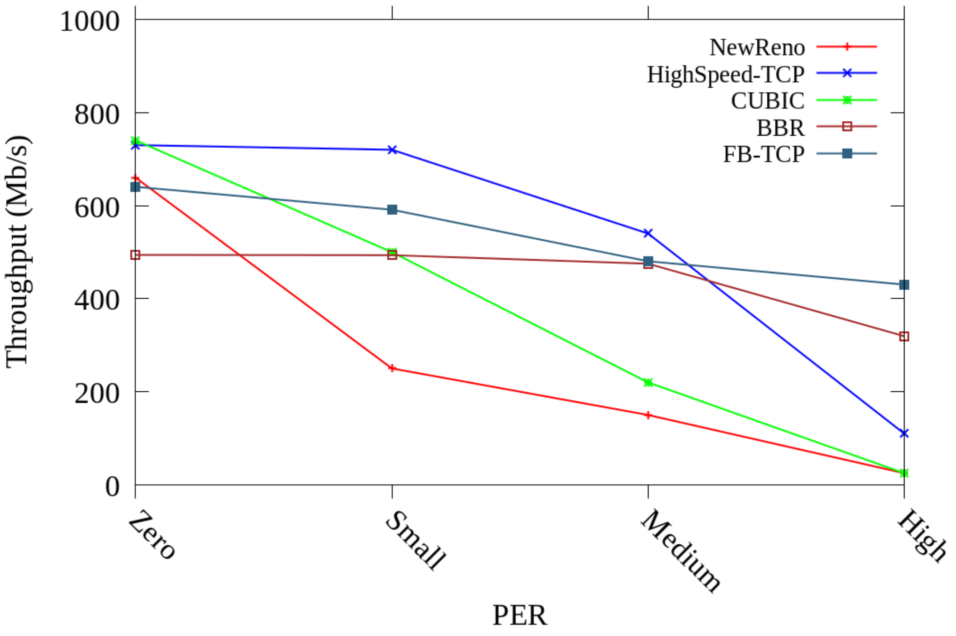
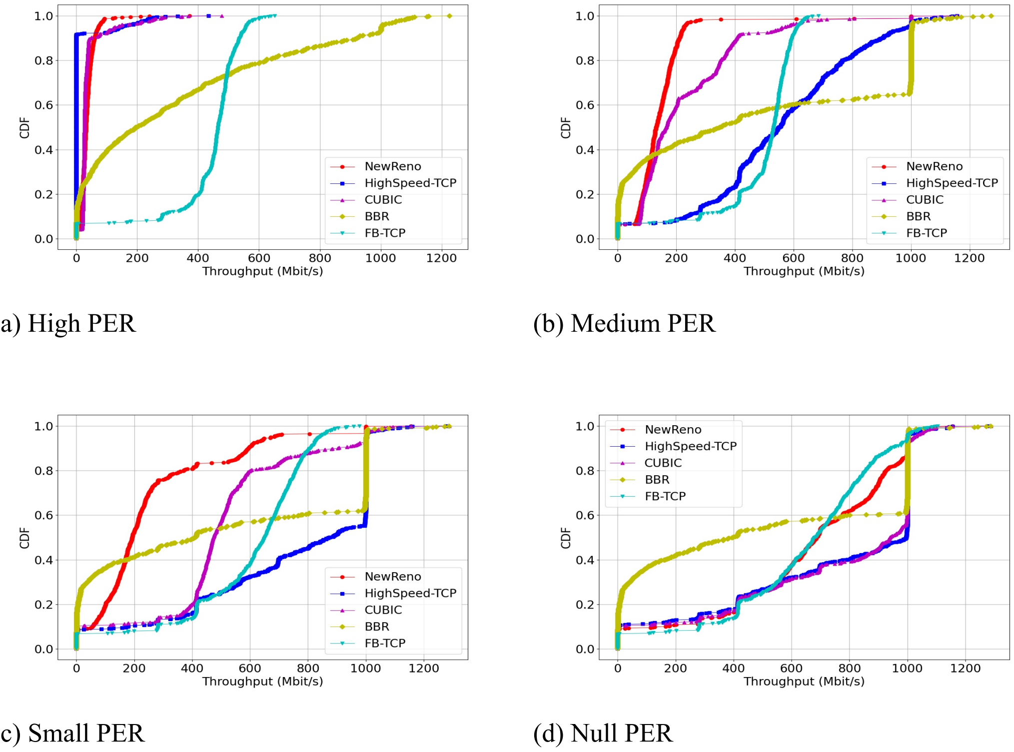


Figure 3.10: Average Throughput of TCP Variants Vs. Different PERs When The Buffer Matches BDP (2.5MB)



(a) High PER

(b) Medium PER

(c) Small PER

(d) Null PER

Figure 3.11: CDF of Throughput of TCP Variants Vs. Different PERs When The Buffer Matches BDP (2.5MB)

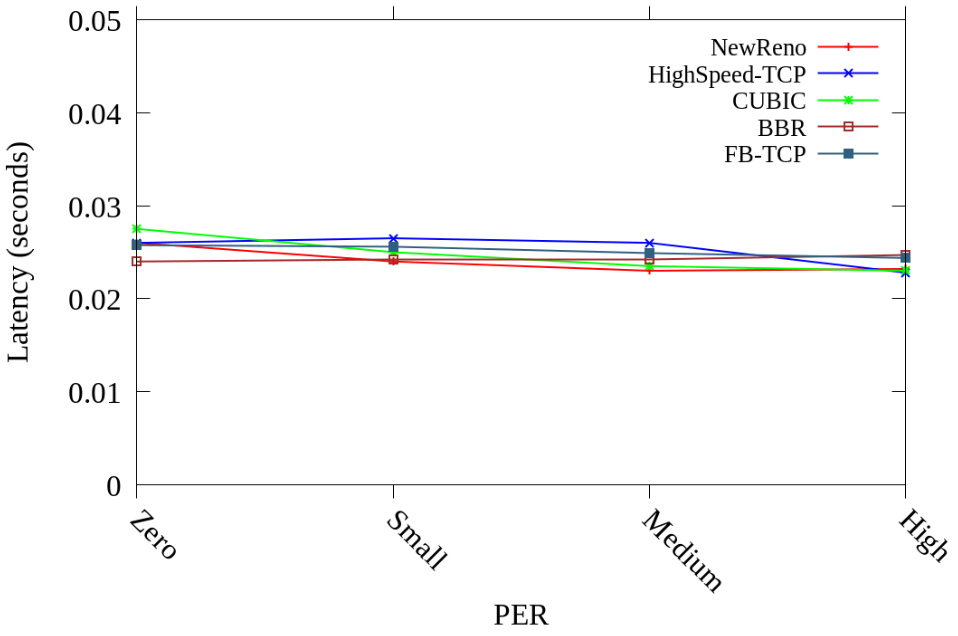
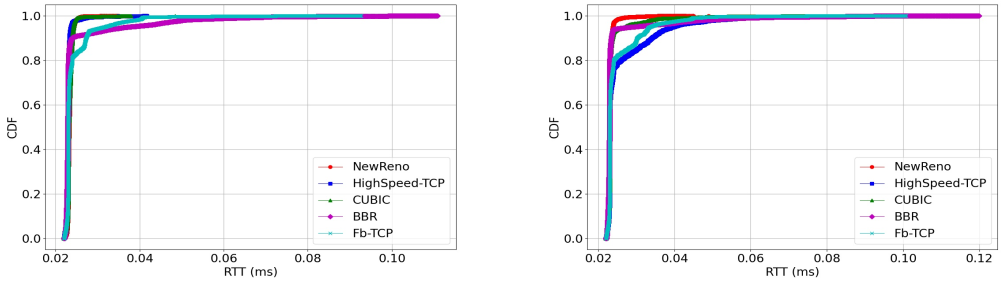
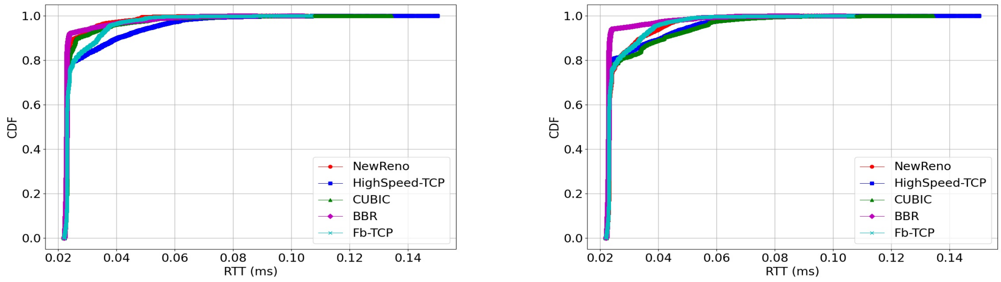


Figure 3.12: Average Latency of TCP Variants Vs. Different PERs When The Buffer Matches BDP (2.5MB)



(a) High PER

(b) Medium PER



(c) Small PER

(d) Null PER

Figure 3.13: CDF of Latency of TCP Variants Vs. Different PERs When The Buffer Matches BDP (2.5MB)

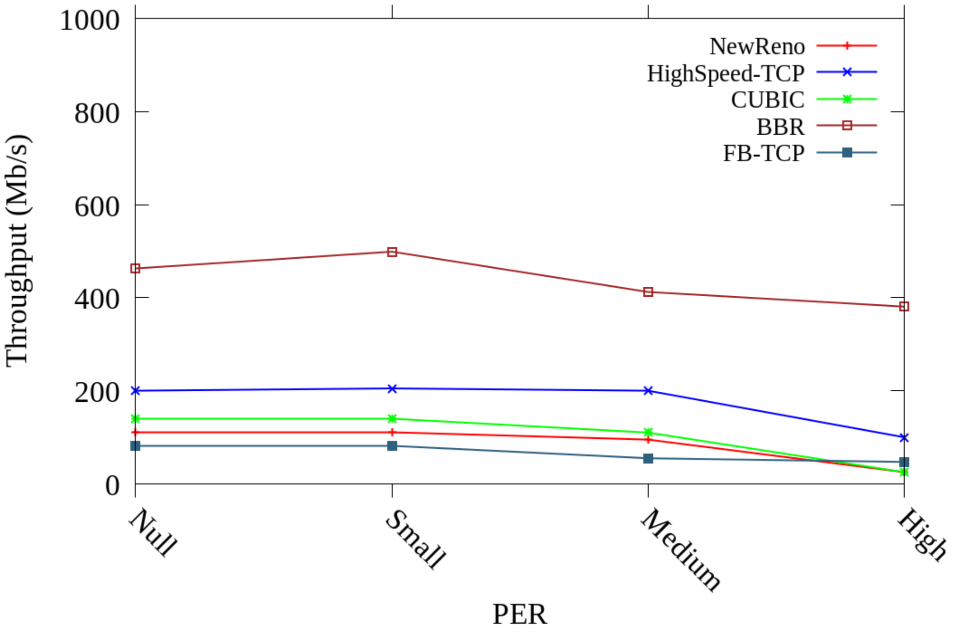
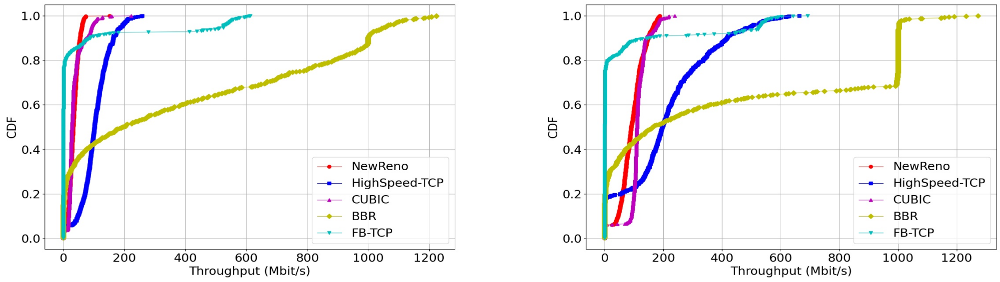
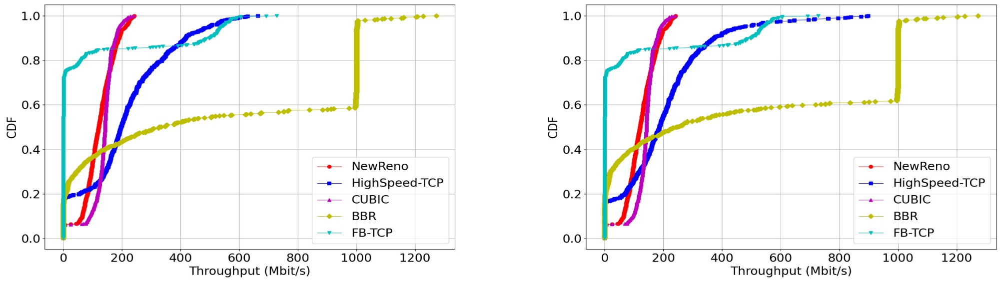


Figure 3.14: Average Throughput of TCP Variants Vs. Different PERs When The Buffer is Small (0.25MB)



(a) High PER

(b) Medium PER



(c) Small PER

(d) Null PER

Figure 3.15: CDF of Throughput of TCP Variants Vs. Different PERs When The Buffer is Small (0.25MB)

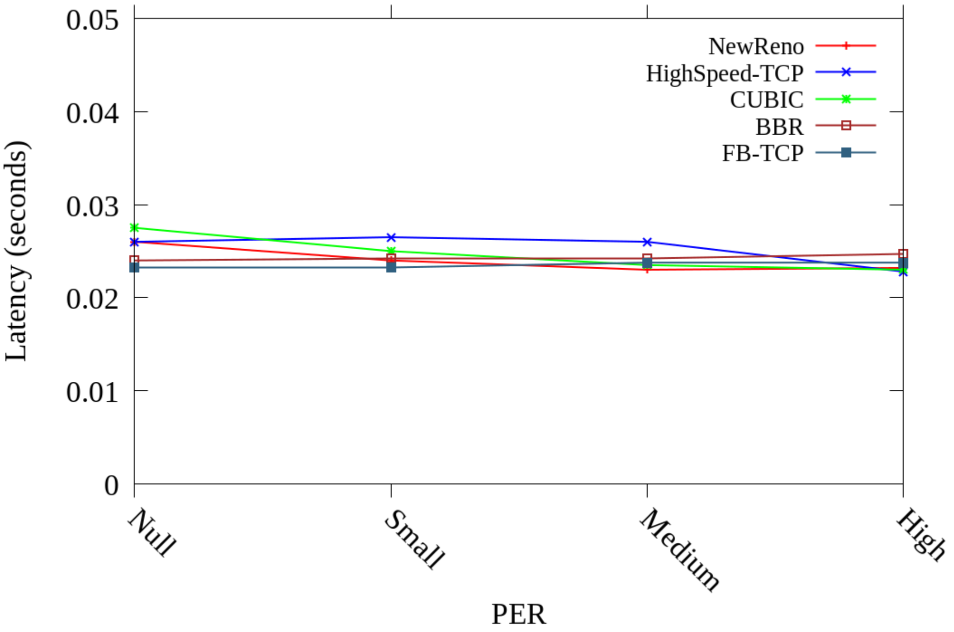
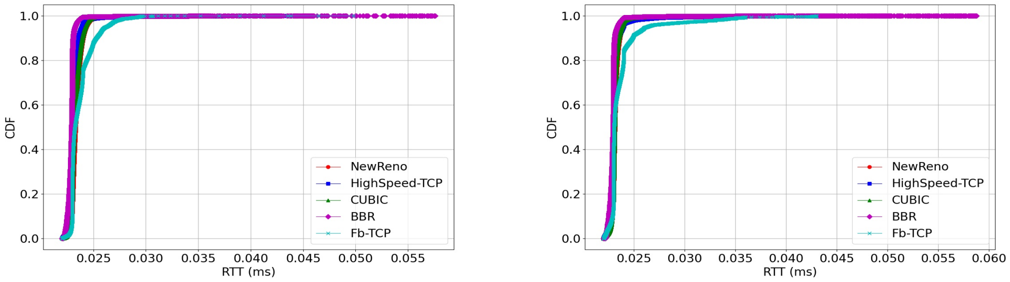
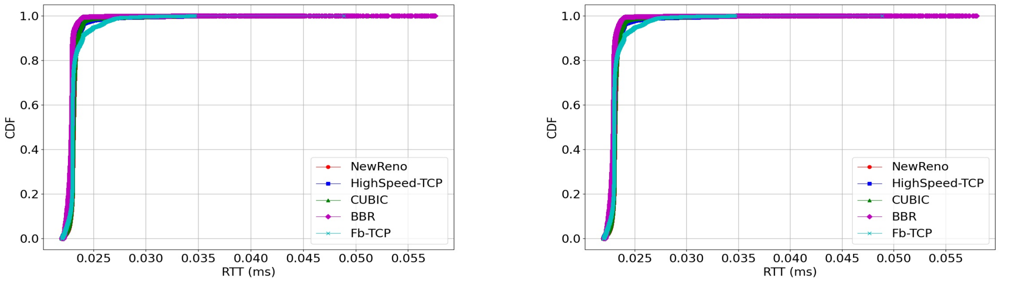


Figure 3.16: Average Latency of TCP Variants Vs. Different PERs When The Buffer is Small (0.25MB)



(a) High PER

(b) Medium PER



(c) Small PER

(d) Null PER

Figure 3.17: CDF of Latency of TCP Variants Vs. Different PERs When The Buffer is Small (0.25MB)

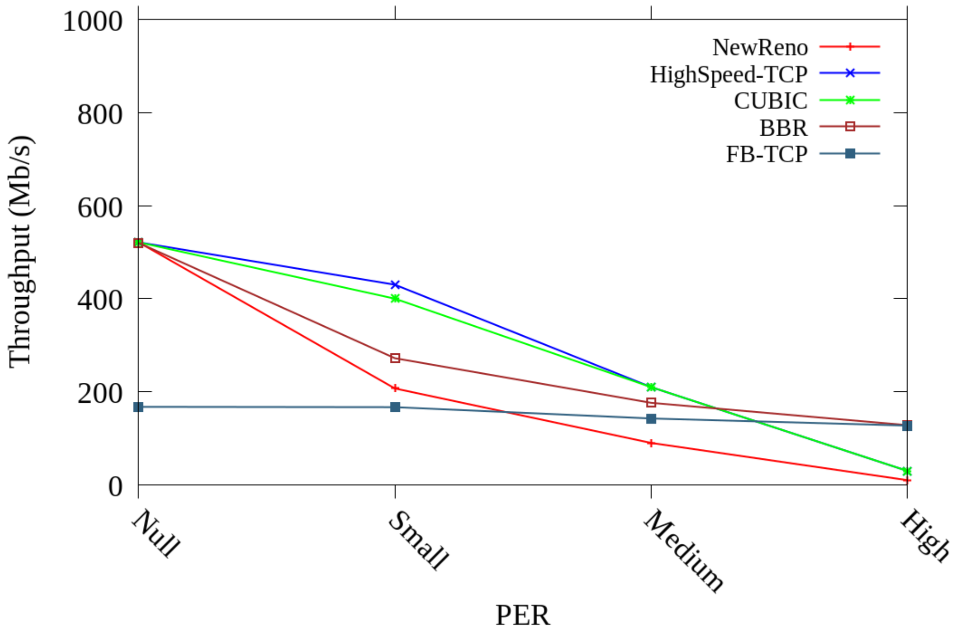
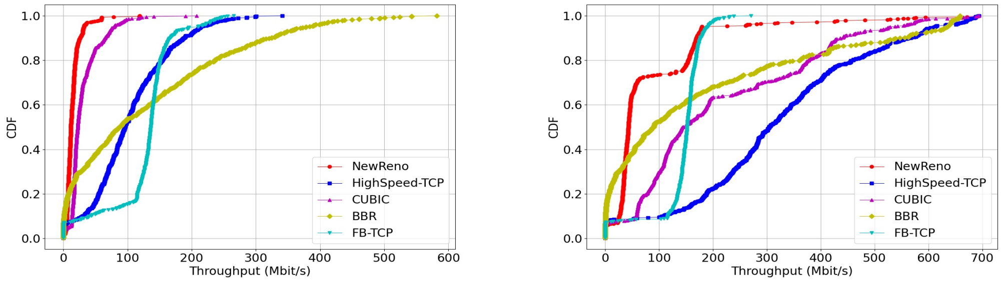
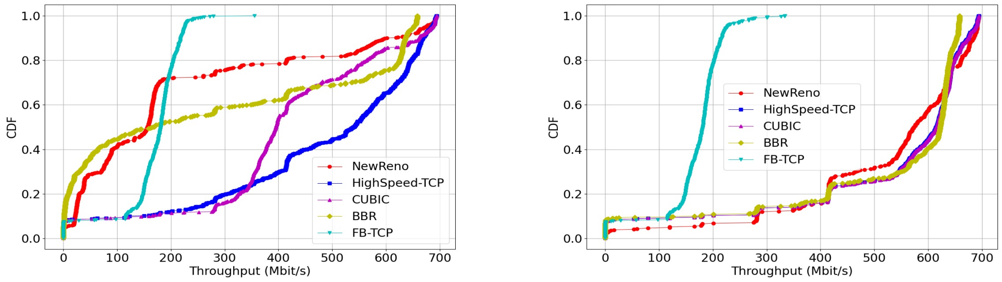


Figure 3.18: Average Throughput of TCP Variants Vs. Different PERs, Remote Server Scenario



(a) High PER

(b) Medium PER



(c) Small PER

(d) Null PER

Figure 3.19: CDF of Throughput of TCP Variants Vs. Different PERs, Remote Server Scenario

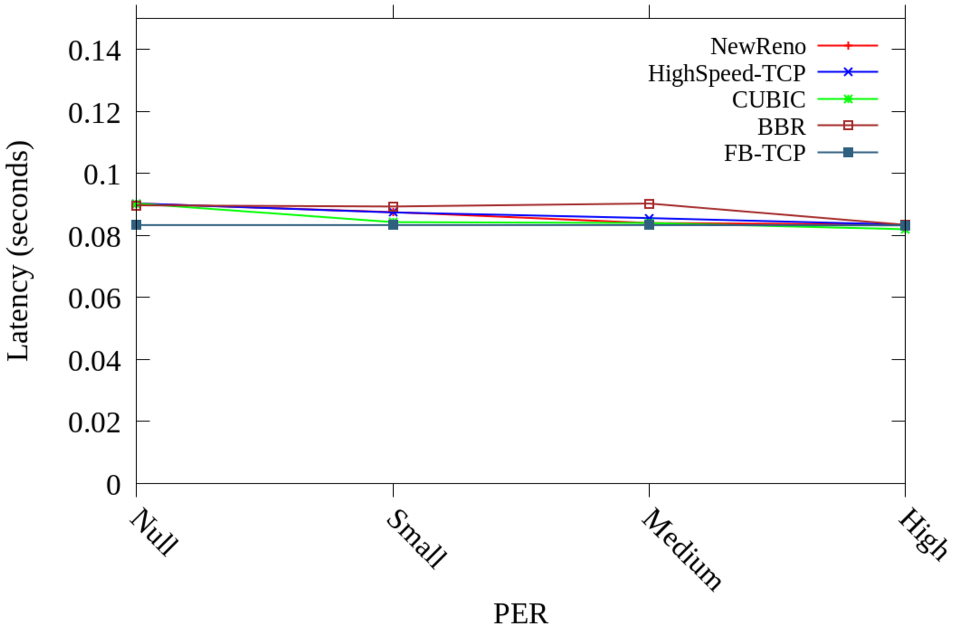
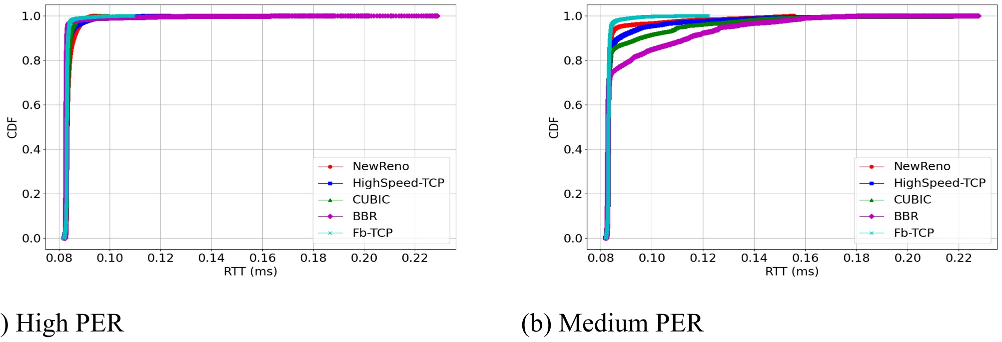
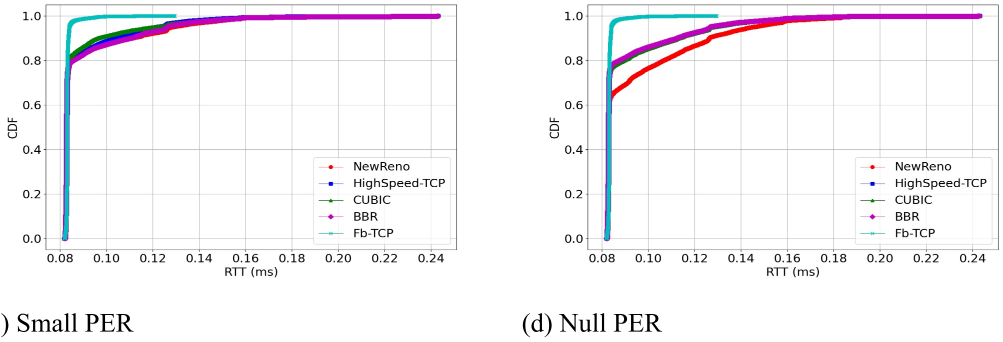


Figure 3.20: Average Latency of TCP Variants, Remote Server Scenario



(a) High PER

(b) Medium PER



(c) Small PER

(d) Null PER

Figure 3.21: CDF of Latency of TCP Variants Vs. Different PERs, Remote Server Scenario

Table 3.3: Average throughput (Mb/s) of TCP variants when different buffer and PER conditions are considered

PER	TCP Protocols				
	NewReno	**HighSpeed**	**CUBIC**	**BBR**	**FB-TCP**
			Small Buffer		
Null	110	200	140	463	82
Small	110	205	140	499	82
Medium	95	200	110	412	55
High	25	100	25	381	47
			BDP Buffer		
Zero	660	730	740	494	640
Small	250	720	500	494	591
Medium	150	540	220	475	480
High	25	110	25	319	430
			Large Buffer		
Zero	776	776	776	572	640
Small	286	752	631	572	591
Medium	146	551	222	572	480
High	36	115	41	418	430
			Remote Server		
Null	521	521	521	520	168
Small	207	430	400	272	167
Medium	90	210	210	176	143
High	10	30	30	128	127

Table 3.4: **Average latency (ms) of TCP variants when different buffer and PER conditions are considered**

PER	TCP Protocols				
	NewReno	**HighSpeed**	**CUBIC**	**BBR**	**FB-TCP**
			Small Buffer		
Null	26	26	27.5	24	23.26
Small	24	26.50	25	24.20	23.26
Medium	23	26	23.50	24.20	23.77
High	23.2	22.8	23	24.70	23.77
			BDP Buffer		
Zero	26	26	27.5	24	25.75
Small	24	26.5	25	24.2	25.59
Medium	23	26	23.5	24.2	24.90
High	23.2	22.8	23	24.7	24.38
			Large Buffer		
Zero	39.38	39.38	39.38	34.70	25.75
Small	33.83	34.50	29.70	33.22	25.59
Medium	23.16	25.90	23.68	30.13	24.90
High	23.37	23.19	23.35	27.98	24.38
			Remote Server		
Null	90.29	90.29	90.29	89.71	83.34
Small	87.5	87.42	84.3	89.33	83.35
Medium	84	85.59	84.0	90.29	83.34
High	83.5	83.47	82.0	83.42	83.25

3.7 Summary

In this chapter, the thesis's research methodology and the proposed schemes are expounded. The chapter begins by elucidating the notations and definitions employed throughout the thesis. Subsequently, it provides a comprehensive portrayal of the research framework, delineating the key stages of the study, encompassing problem formulation, assessment of state-of-the-art algorithms, and introduction of the proposed contributions. These contributions are then implemented and scrutinized for performance, specifically focusing on the experimental environments and performance metrics. The chapter culminates by summarizing the various stages of the research, elucidating their seamless integration into a cohesive study. The section on experimental environments furnishes a meticulous overview of the software and hardware tools utilized, ensuring the reliability and accuracy of the results.

Furthermore, this chapter focuses on investigating the weaknesses and challenges of benchmark TCP variants (NewReno, HighSpeed, CUBIC, BBR, FB-TCP) in mmWave cellular networks through extensive simulations. The study explored factors such as LoS, NLoS, buffer size, PER, and server location, showing how these elements affect TCP performance in terms of throughput, latency, and CDF for both. Interestingly, the study found that non-congestion issues affect TCP performance due to NLoS conditions. Additionally, the small buffer size caused degradation in TCP performance in the presence of high PER conditions.

Furthermore, the study shows an inverse relationship between packet error rate, throughput, and latency, where higher error rates led to throughput degradation and lower latency. Moreover, server location was found to play a significant role: when the server moved from the edge of the network to a remote location, throughput significantly decreased and latency increased for the benchmark TCP variants. For instance, the evaluation and results presented in Section 3.6 demonstrate that existing TCP implementations do not effectively utilize the available bandwidth of mmWave cellular networks to achieve high data rates and low latency. This limitation hinders the ability to meet the performance requirements of mobile applications, such as augmented reality, virtual reality, and cloud gaming. Therefore, designing new solutions that address these challenges is both valuable and feasible, intending to enhance overall TCP performance in terms of bandwidth utilization, high throughput, and low latency in mmWave cellular networks.

To sum up, the methodology and evaluation outlined in this chapter lay the groundwork for the subsequent chapters detailing the contributions.

CHAPTER 4

RTTV-BASED CCA: CONGESTION CONTROL ALGORITHM TO MITIGATE THE IMPACT OF NON-CONGESTION ISSUE IN MMWAVE CELLULAR NETWORKS

4.1 Introduction

There is a strong need for a new CCA that can reduce sensitivity to non-congestion issues and improve TCP performance in mmWave cellular networks. This chapter introduces a new CCA called RTTV-based CCA, which uses RTT variations to handle non-congestion problems effectively.

4.2 RTTV-Based CCA: The Proposed Algorithm

The main goal in designing a new protocol to meet the application requirements over cellular networks should be achieving the highest possible throughput while maintaining acceptable latency. Notably, addressing NLoS conditions is crucial, as they can significantly impact TCP performance in 5G mmWave cellular networks. In cellular networks, when a sender initiates communication with a server, the RTT will experience variations due to channel characteristics and can be characterized by three values, RTT_{base}, RTT_{cur}, and RTT_{max} as illustrated in Figure 4.1. Where the RTT_{base} is the shortest RTT, and it can take place when the network Bandwidth is not fully utilized, the buffer is considered empty, and there is no congestion in the network. The second RTT is RTT_{cur} and it is the current RTT, and it is practical RTT and its value lies between RTT_{base} and RTT_{max}, indicating and reflecting the current network situation. The third RTT is RTT_{max}, corresponds to the maximum RTT, and it composes that the network is under heavy load and the buffer size is almost full, and there is a delay to acknowledging the sender. RTT commonly and frequently varies in cellular networks. RTT variations due to switching UE from LoS to NLoS conditions and viers versa. Therefore, RTT variations significantly impact CCA to utilize the available bandwidth efficiently and can be used to adjust the cwnd size of the CCA. On the other hand, within a network context, the link Utilization Ratio (UR) often pertains to the extent to which resources like bandwidth or buffer space are utilized. A high UR signifies efficient resource utilization, whereas a low UR indicates potential for optimization or underutilized resources, as shown in Figure 4.1. Bandwidth utilization denotes the percentage of total available network bandwidth used at any given time. For instance, if a network with a bandwidth of 100 Mbps currently transmits data at 50 Mbps, the bandwidth utilization would be 50%.

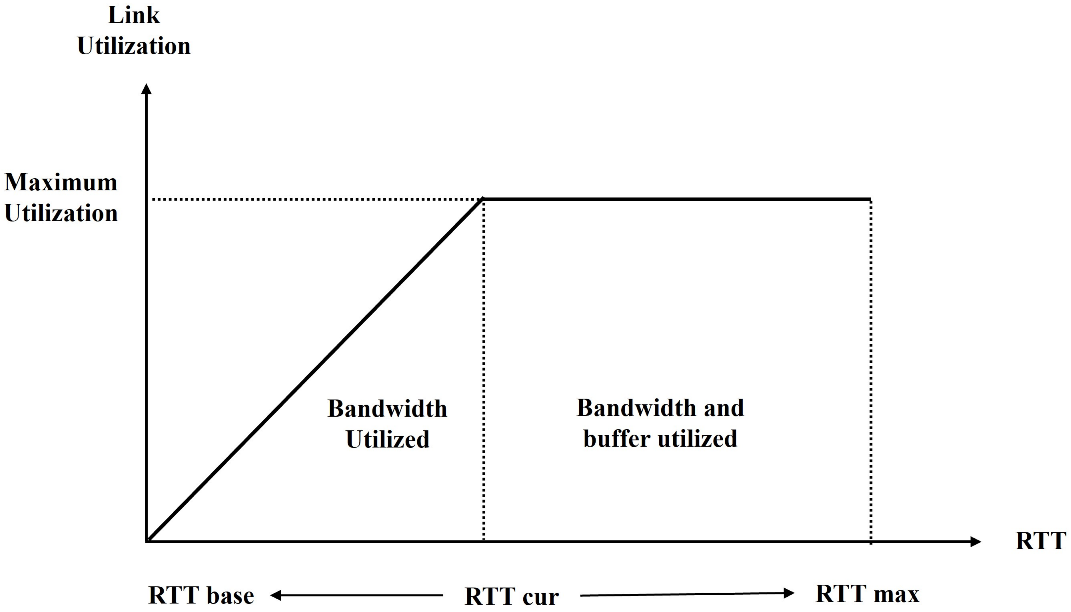


Figure 4.1: RTT Impact on Link Utilization

Conversely, buffer utilization refers to the percentage of buffer space currently occupied. Buffers temporarily store data packets when they arrive faster than they can be processed or transmitted. For example, if a buffer with a capacity of 100 packets currently holds 80 packets, the buffer utilization would be 80%. Although bandwidth and buffer utilization are interrelated, they represent different facets of network performance. Bandwidth utilization quantifies data transmission over the network, whereas buffer utilization measures the temporary storage of data packets. This divergence arises because even when bandwidth is fully utilized and data packets are transmitted at the maximum rate, the buffer might not be filled if the incoming data rate matches the outgoing data rate. However, buffer utilization hitting 100% can indicate a potential bottleneck in the network, indicating no available space to store incoming packets temporarily. This situation can lead to packet loss or bufferbloat phenomena.

As shown in Figure 4.1, the utilization ratio varies with the RTT. In this study, the UR is defined as a percentage of the utilized buffer and BDP. Within the framework of the RTTV-CCA protocol, the RTT variables in terms of RTT_{base}, RTT_{cur}, and RTT_{max} are incorporated as a supplementary parameter to formulate the utilization ratio as Equation (4.1) shows.

$$Utilization\ Ratio(UR\%) = \frac{RTT_{cur}}{RTT_{max}} \times 100\% \tag{4.1}$$

Where RTT_{cur} is the current RTT obtained from the last ACK, and RTT_{max} is the maximum RTT observed during the connection. The utilization ratio reaches 100% only when RTT_{cur} approaches RTT_{max}, and this occurs solely when the bandwidth and buffer at the bottleneck link are fully utilized. Thus, the utilization reaches the maximum when RTT_{cur} approaches the value equal to RTT_{max}, as shown in Equation (4.1). In this, the unutilized ratio will be zero. On the other hand, the weight factor enlarges to the maximum value when RTT_{cur} is equal to RTT_{base}, which implies the network is under light traffic. RTT_{cur} moves toward RTT_{max}, and the weight factor shrinks to the minimum value, indicating a heavily loaded network. The main purpose of the weight factor is to estimate the maximum possible cwnd for the underlying network, which is calculated according to Equation (4.2). A square root has been selected to generate a convex-up curve to minimize the underutilization area (Alrshah et al., 2019). Additionally, the chosen (F) should not be overly complex to ensure ease of implementation within an operating system such as the Linux kernel. The square-root is found to meet these requirements. Ultimately, the resulting value is used for the incremental value during the CA stage to increment the cwnd, as demonstrated in Equation (4.5).

$$weight\ factor = \frac{1}{UR} \tag{4.2}$$

Where UR is the bandwidth utilization ratio over the link.

and

$$F = \sqrt{(weight\ factor) \times cwnd} \tag{4.3}$$

Where $weightfactor$ indicates whether the network is under light or heavy traffic, and $cwnd$ is the congestion window.

Thus

$$F = \sqrt{\frac{(RTT_{max} + RTT_{base}) * cwnd}{(RTT_{cur})}} \tag{4.4}$$

According to Equation (6.3), the increment of cwnd in the CA stage of the CCA will be as Equation (4.5) shows.

$$cwnd = cwnd + \frac{F}{cwnd} \tag{4.5}$$

Where cwnd is the current congestion window, and the fraction $F/cwnd$ is the increment value of the current congestion window for each arrival of non-Duplicate ACK to the sender host in the congestion avoidance phase. The cwnd will be increased by an increment value equal to the (F) divided by cwnd in the CA phase, as mentioned in Equation (4.5). The value of the (F) parameter is calculated by Equation (4.4) based on the values of RTT max, RTT base, RTT current, and the magnitude of cwnd itself. The weight factor part in Equation (4.3) reflects the utilization ratio of the network according to the RTT values, as stated in Equation (4.2). According to Equation (4.5), the cwnd grows aggressively as illustrated in Figure 4.2 to utilize the network bandwidth fully and gently as the data rate approaches the network bandwidth limit.

For further elaboration, Figure 4.3 illustrates the control flow diagram of the proposed RTTV-based CCA, and Algorithm 4.1 outlines the algorithm's source code. Subsequent subsections comprehensively explain the unique mechanism employed by RTTV-based CCA.

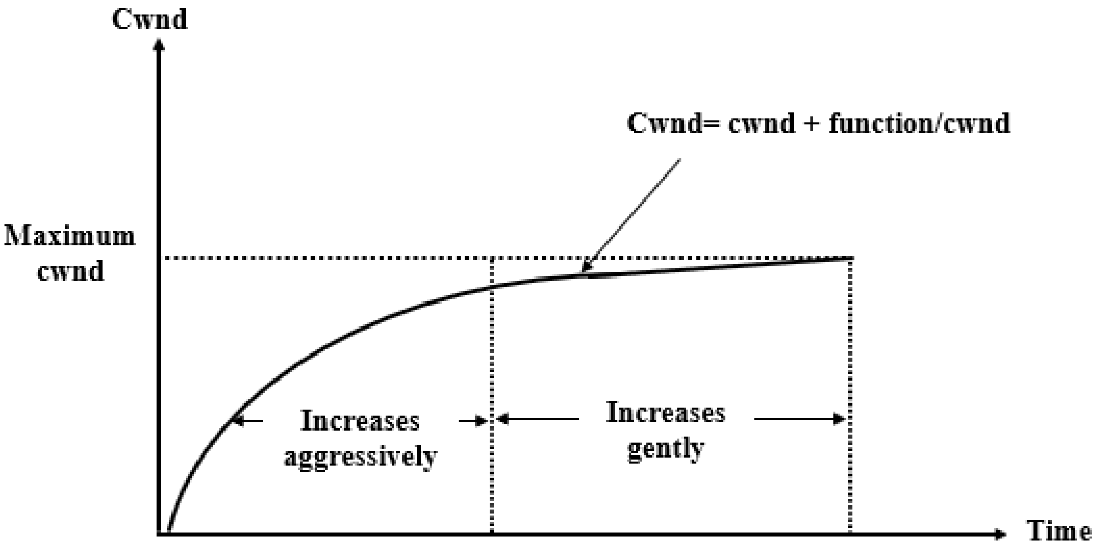


Figure 4.2: Cwnd Increase of RTTV-Based CCA

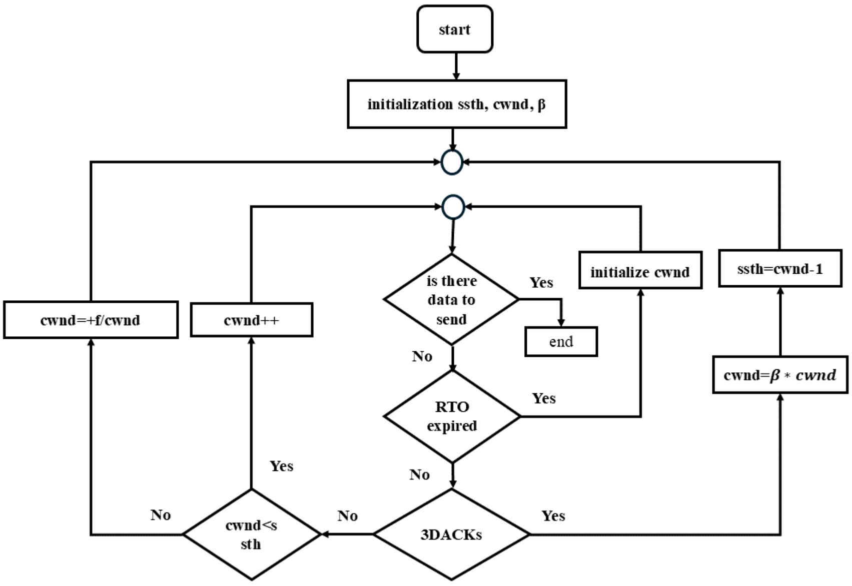


Figure 4.3: RTTV-Based CCA Flowchart

4.3 RTTV-Based CCA Mechanism

The mechanism proposed in this chapter aims to enhance TCP bandwidth utilization and maintain the latency over 5G mmWave cellular networks. RTTV-CCA utilizes the variation of the different $RTTs$ parameters along the cwnd magnitude to determine the increment of the cwnd during the CA stage of the algorithm. This approach differs from the methods employed by TCP variants such as NewReno, HighSpeed, CUBIC, BBR, and FB-TCP. Fortunately, RTTV-CCA can enhance bandwidth utilization while maintaining latency like other TCP variants.

RTTV-CCA employs an exponential increase of cwnd during its initial slow start phase. This increase continues until reaching the slow start threshold or encountering the first loss, signified by the reception of 3DACKs or the RTO expired. At this point, RTTV-CCA reduces the cwnd by applying the multiplicative decrease factor β. Subsequently, RTTV-CCA enters the CA stage as Algorithm 4.1 shows, increasing the cwnd using the contribution (F) as depicted in Equations (4.4) and (4.5) respectively, resulting in short epochs with convex-up curves of increase. As a result, implicitly, the short epochs decrease the unutilized bandwidth part of the network. If a packet loss occurs during this CA stage, RTTV-CCA decreases its cwnd by the multiplicative decrease factor β to initiate another epoch within the CA stage.

4.4 RTTV-Based CCA: Performance Evaluation

This chapter aims to demonstrate RTTV-based CCA, which enhances the bandwidth utilization of mmWave cellular networks. The ns-3 network simulator is used to evaluate the protocol's performance along TCP variants. RTTV-based CCA is implemented into ns-3 as a simulation module. The main goal is to improve TCP performance in mmWave cellular networks due to non-congestion issues, which are caused by UE moves from LoS to NLoS conditions, due to mobility and obstacles between UE and gNB station.

Algorithm 4.1: RTTV-Based CCA

```
Initialize: $cwnd, ssthresh, rttbase, rttcur, rttmax$;
if there is data to send then
    if RTO not expired then
        if no 3DACK then
            if $cwnd < ssthresh$ then
                $cwnd = cwnd + 1$;
            else
                $rtt_{cur} = Acktime - Sendtime$;
                where $Acktime$ is the time the acknowledgment is received at the
                  source host, and $Sendtime$ is the time the packet was sent from the
                  source host.
                if $rttbase > rttcur$ then
                    $rttbase = rttcur$;
                end
                if $rttmax < rttcur$ then
                    $rttmax = rttcur$;
                end
                $F = \sqrt{\left(\frac{(rttmax+rttbase)}{rttcur} \times cwnd\right)}$;
                $cwnd = cwnd + \frac{F}{cwnd}$;
            end
        else
            Multiplicative decrease;
        end
    else
        Slow start stage;
    end
end
```

4.4.1 Experimental Setup

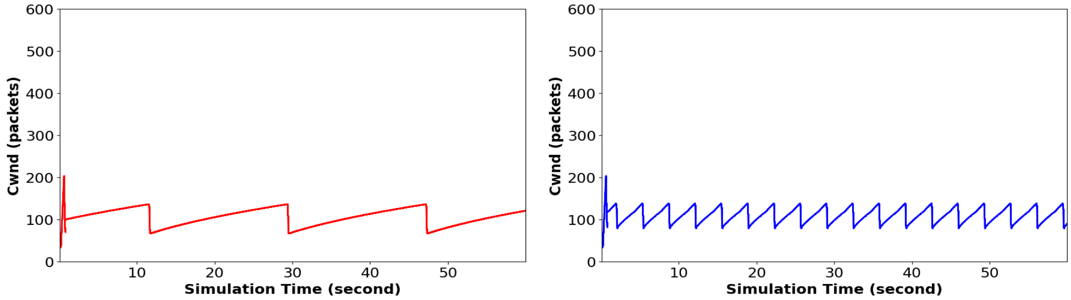
Extensive simulation experiments are conducted using the well-known ns-3 network simulator to evaluate the proposed CCA by comparing its performance to the benchmark protocols. The experimental setup is detailed in Chapter 3, and the specific simulation parameters are described in Table 3.2 of Chapter 3.

4.4.2 Results and Discussion

This section analyzes the performance of RTTV-CCA compared to other CCAs. It also presents the protocol's performance results concerning cwnd fluctuations, throughput, and latency. The objective is to demonstrate the influence of mmWave channel insufficiency on the overall performance of the algorithms in 5G mmWave cellular networks. More specifically, the impact of switching from LoS to NLOS conditions and vers versa on TCP performance.

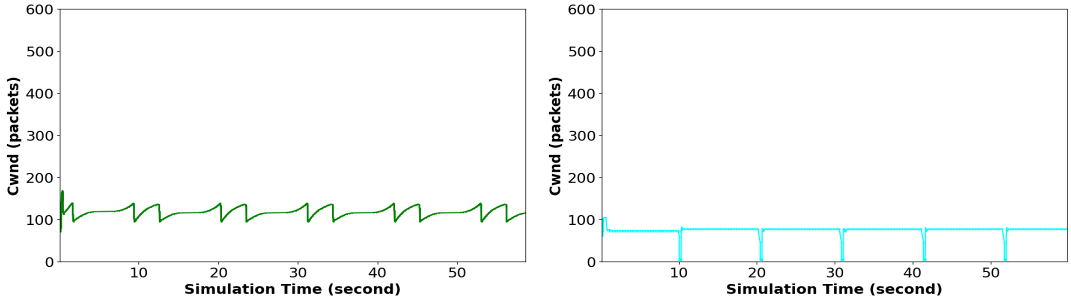
4.4.2.1 Congestion Window Fluctuations

Cwnd evolution is fundamental to all CCAs as it directly impacts various performance metrics, including throughput, delay, and bandwidth utilization. Premanly experiment in Figure 4.4, the evolution of cwnd is depicted for the studied CCAs. RTTV-CCA exhibits the fastest increase in cwnd as Figure 4.4f shows. HighSpeed and CUBIC exhibit slower oscillations in cwnd than RTTV-CCA, as Figures 4.4b and 4.4c show, respectively. Conversely, NewReno undergoes extended epochs while oscillating between 80 and 140 MSS, as shown in Figure 4.4a. On the other hand, in Figure 4.4d, BBR consistently oscillates around 80 MSS and periodically drains the queue every 10 seconds in accordance with its congestion control mechanism before resuming oscillations around the same value. The cwnd of FB-TCP varies from 0 to 450 segments with high variation as Figure 4.4e shows. Figure 4.4g shows and underscores a direct relationship between epoch duration, cwnd size, and throughput, where shorter epochs lead to larger cwnd and subsequently higher throughput, and vice versa.



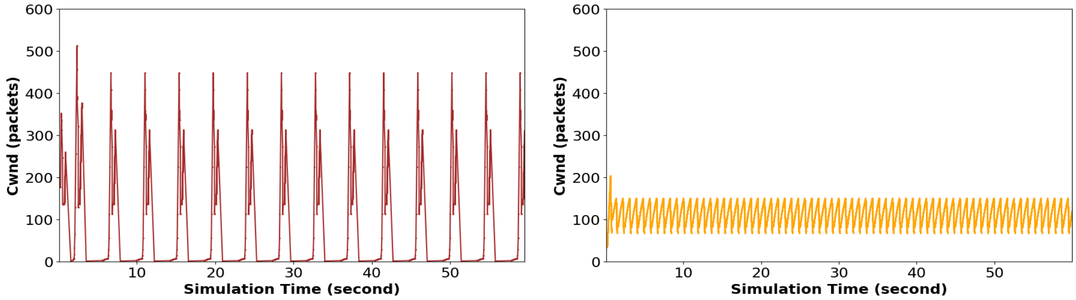
(a) NewReno cwnd

(b) HighSpeed-TCP cwnd



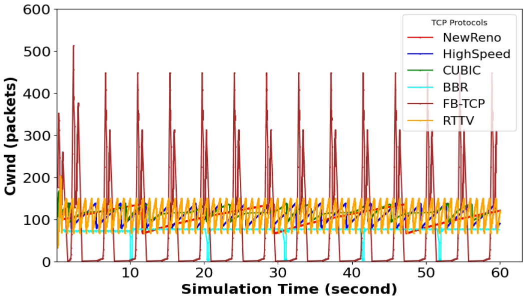
(c) CUBIC cwnd

(d) BBR cwnd



(e) FB-TCP cwnd

(f) RTTV-CCA cwnd



(g) Combined cwnds

Figure 4.4: Cwnd Comparison of Congestion Control Algorithms

Notably, RTTV-CCA reaches a near-maximum number of packets and then starts fluctuating, forming convex upward curves in very short epochs, as illustrated in Figure 4.4f. With its (F), RTTV-CCA possesses a notable edge over TCP variants that are anticipated to be common in 5G mmWave cellular networks. Consequently, it can be deduced that RTTV-CCA has the potential to enhance the effective utilization of the available bandwidth of 5G mmWave cellular networks.

4.4.2.2 Average Throughput

Figure 4.5 highlights the vulnerability of the evaluated protocols to the small buffer size and under different PERs compared with BDP and unlimited buffer sizes scenarios for NewReno, CUBIC and FB-TCP. The protocol's throughput performance deteriorates significantly as buffer size and error rates escalate, practically for loss-based TCPs and Fuzzy-based TCP. This suggests that the protocols may struggle to recover efficiently from high packet error rates and small buffer sizes in 5G mmWave cellular networks, resulting in slower data transmission and low bandwidth utilization. The RTTV-CCA protocol consistently exhibits higher throughput values than loss-based TCP protocols. This trend remains consistent across various buffer sizes and packet error rates.

The suggested RTTV-CCA exhibits improved average throughput compared to other CCAs, mainly because of its rapid cwnd increase, which stems from its distinctive CA mechanism. Notably, Figure 4.6 and Table 4.1 highlight the optimal performance when the buffer size matches the BDP. The RTTV-CCA protocol initially exhibits the highest throughput compared to loss-based TCP in all PER and buffer size scenarios. This indicates the protocol performs well under varying conditions, delivering a high data transfer rate. However, in the legacy loss-based protocols (NewReno, HighSpeed, CUBIC), the throughput experiences a significant degradation. The throughput decreases dramatically when high packet error rates are applied due to the loss-based TCPs mechanism to deal with congested networks and packet loss. In contrast, BBR demonstrates greater resilience to packet errors, maintaining the same throughput with slight differences across buffer sizes and error rates. BBR employs a model mechanism to calculate the delay and network bandwidth via probing the RTT and BW, mitigating the negative impact of buffer size and packet errors on the protocol performance. Figure 4.6 clearly illustrates that RTTV-CCA maintains superior throughput performance compared to other TCP variants when the applied buffer size equals BDP in all PER scenarios.

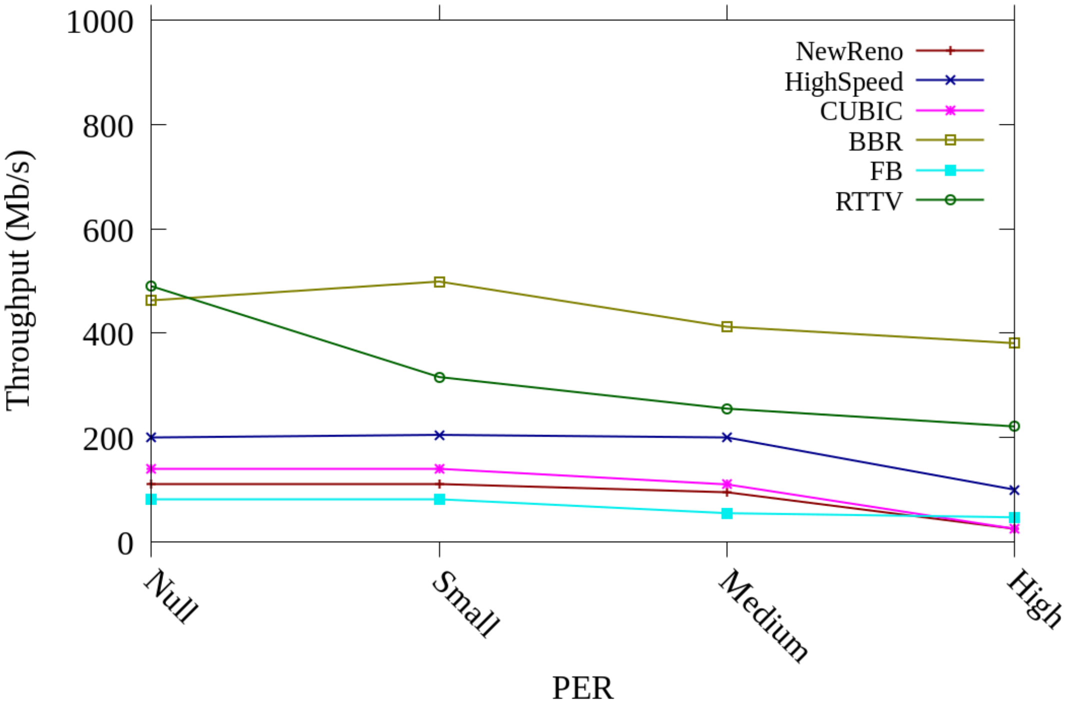


Figure 4.5: Average Throughput of TCP Variants Vs. Different PERs When The Buffer is Small (0.25MB)

Generally, when the buffer size corresponds with the BDP, the RTTV-CCA protocol consistently surpasses CCAs regarding throughput across different scenarios of PER. This advancement in performance leads to a significant augmentation in bandwidth utilization, showcasing enhancements compared to the loss-based TCP, BBR, and FB-TCP algorithms. Additionally, even when the buffer size is greater than the BDP buffer size, as Fig. 4.7 shows, RTTV-CCA continues to perform better than other protocols concerning the throughput. Even though the particular throughput values may vary from those recorded with smaller buffer sizes, RTTV-CCA always gives great throughput performance regardless of the buffer size. These findings show that the RTTV-CCA protocol performs exceptionally well in the throughput. This is the case regardless of the buffer size or PER. It routinely displays superior throughput compared to other protocols when applied to various settings. This highlights the efficacy and efficiency of the RTTV-CCA protocol in maintaining great data transmission speeds, particularly in 5G mmWave cellular networks.

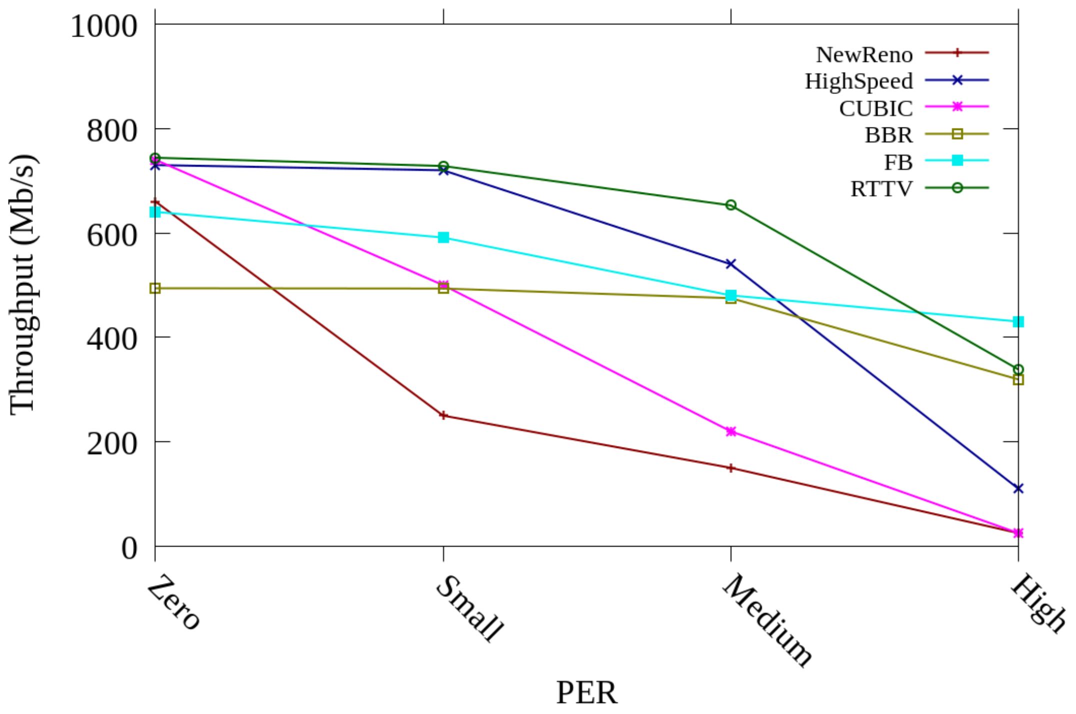


Figure 4.6: Average Throughput of TCP Variants Vs. Different PERs When The Buffer Matches BDP (2.5MB)

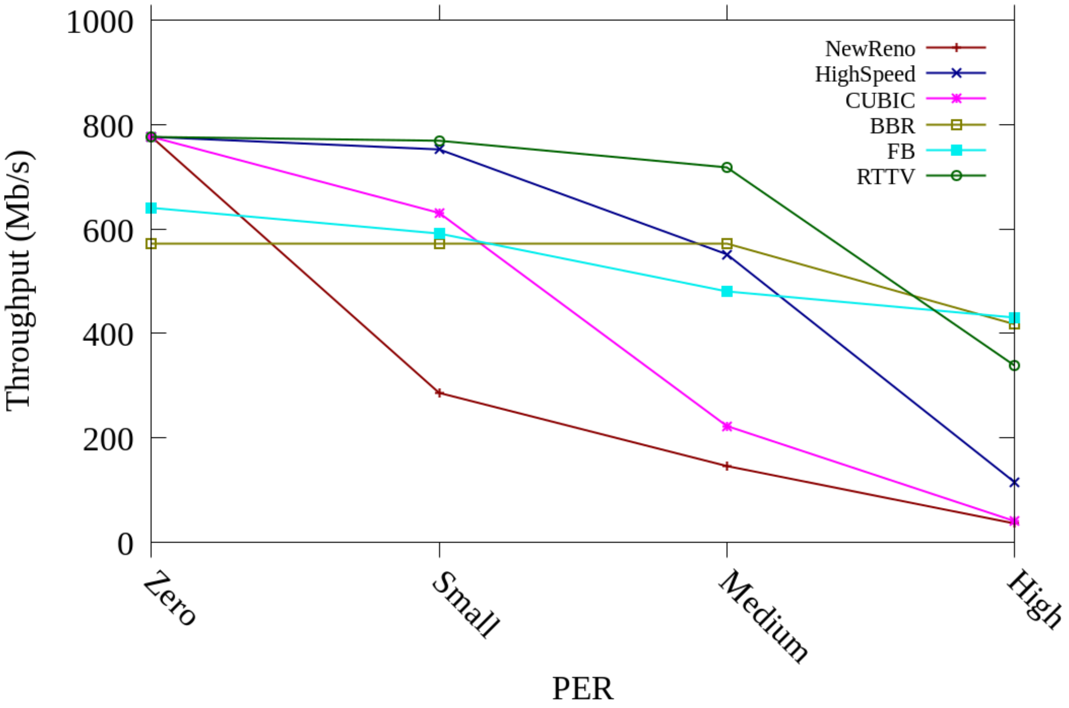


Figure 4.7: Average Throughput of TCP Variants Vs. Different PERs When The Buffer is Large (20MB)

Table 4.1: Percentage improvement of RTTV throughput compared to benchmark protocols

TCP	**PER**			
	Zero	Small	Medium	High
NewReno	12.73	191.26	335.16	1253.96
HighSpeed	1.92	1.13	20.79	207.72
CUBIC	0.54	45.63	196.64	1253.96
BBR	50.61	47.64	37.30	5.95
FB-TCP	16.20	23.17	35.88	-21.33

4.4.2.3 Average Latency

The RTTV-CCA protocol consistently demonstrates low delay values in Figure 4.8 when the buffer size is small, irrespective of the error rate. This indicates the protocol maintains low latency despite varying error rates under small buffer sizes, as illustrated in Figure 4.8 shows. Such performance emphasizes the protocol's effectiveness. Similarly, when the buffer size equals the BDP as in Figure. 4.9, RTTV-CCA maintains consistently low delay values with a slight increase compared to the loss-based TCP and BBR in small and BDP buffer cases. Given an appropriate buffer size adjustment, this showcases the protocol's ability to manage delays regardless of error rates efficiently.

In cases where the buffer size exceeds the BDP, RTTV-CCA slight increase in latency values as Figure 4.10 compares the TCP variants and simultaneously utilizes the networks' bandwidth and attains higher throughput than legacy protocols. However, while RTTV-CCA and FB-TCP achieve lower latency in an unlimited buffer scenario when high PER is applied, as Figure 4.10, RTTV-CCA has the advantage of attaining higher throughput.

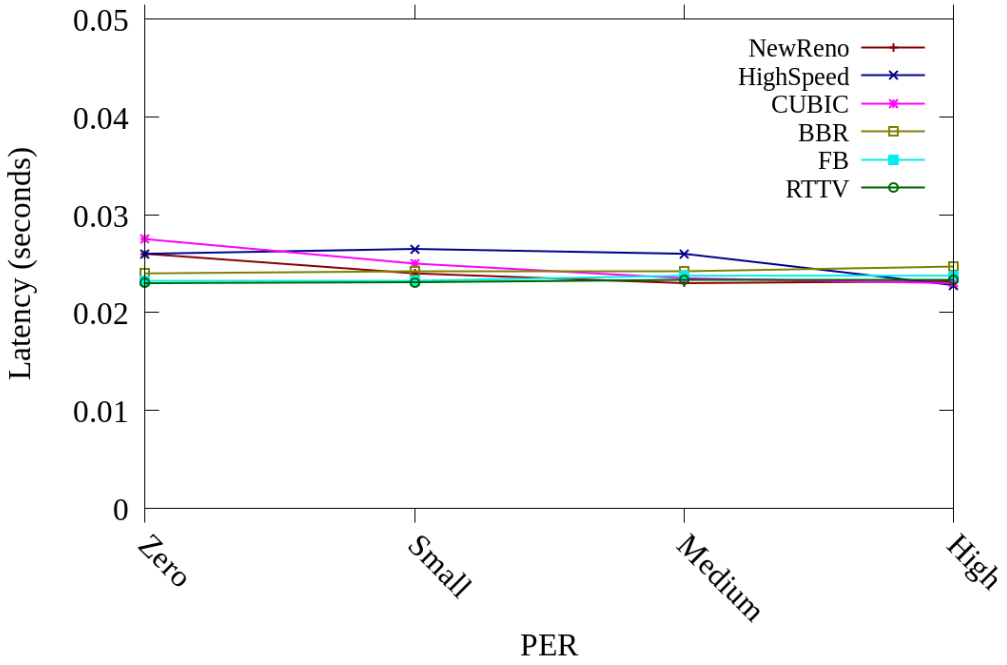


Figure 4.8: Average Latency of TCP Variants Vs. Different PERs When The Buffer is Small (0.25MB)

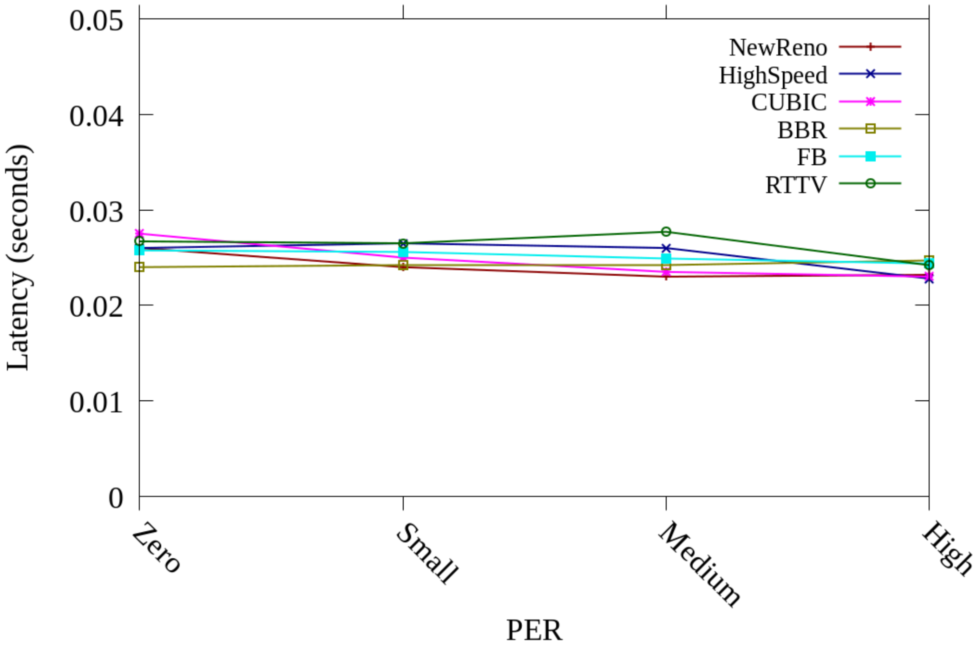


Figure 4.9: Average Latency of TCP Variants Vs. Different PERs When The Buffer Matches BDP (2.5MB)

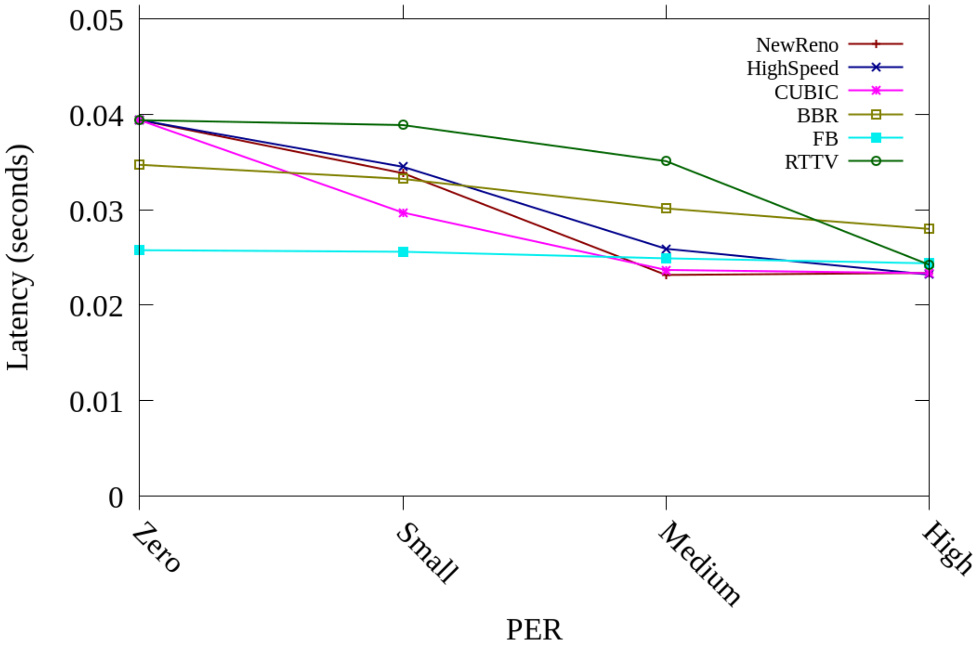


Figure 4.10: Average Latency of TCP Variants Vs. Different PERs When The Buffer is Large (20MB)

Overall, the findings suggest that the RTTV-CCA protocol performs admirably in the of delay, particularly with small buffer sizes or when the buffer size aligns with the BDP. It delivers lower delays across error rates, particularly when PER is high. Generally, the latency remains approximately the same for CCAs, with slight differences in the case of unlimited buffer size, with considering the RTTV-CCA to attain higher throughput than benchmark TCP protocols, Specifically when the buffer matches or exceeds the BDP.

4.5 Summary

An enhanced CCA to increment the cwnd in the congestion avoidance phase. RTTV-CCA has been proposed in this work to mitigate TCP sensitivity to non-congestion states due to switching UE from LoS to NLoS conditions. The proposed technique tackles the limitations of existing TCPs in mmWave cellular networks with its novel mechanism. RTTV-based CCA has been assessed and compared to benchmark TCP protocols such as NewReno, HighSpeed, CUBIC, BBR, and FB-TCP using extensive simulation. RTTV-CCA shows enhancement in performance over mmWve cellular networks in the presence of an edge server. The results demonstrate the utility of RTTV-CCA to enhance the utilization of the available bandwidth of 5G mmWave cellular networks, where legacy TCPs struggle with non-congestion issues due to NLoS challenges. For instance, RTTV-CCA significantly improves throughput and reduces latency compared to baseline protocols, outperforming the benchmark TCP protocols (NewReno, HighSpeed, CUBIC, BBR) by (1253.96, 207.72, 1253.96, 5.95)% respectively under high PER conditions when buffer size matches the Bandwidth Delay Product (BDP).

In general, RTTV-CCA exhibits remarkable performance compared to the benchmark TCP protocols examined, signifying a substantial leap forward in optimizing the bandwidth utilization of 5G mmWave cellular networks and mitigating NLoS conditions. Furthermore, RTTV-CCA's superior performance in terms of throughput compared to the benchmark TCP protocols, and also maintains latency stability in most PER cases with only a slight difference.

CHAPTER 5

MSS-BASED CCA: CONGESTION CONTROL ALGORITHM TO MITIGATE SMALL BUFFER AND PER IMPACTS IN MMWAVE CELLULAR NETWORKS

5.1 Introduction

In addressing the challenge of bandwidth underutilization in cellular networks due to small buffers and high PER, this chapter presents a novel CCA named MSS-based. The primary innovation of MSS-based CCA lies in its implementation of an increment factor (Inc) mechanism to augment the cwnd during the congestion avoidance phase of the TCP protocol. This Inc is contingent on the MSS and RTT.

5.2 MSS-Based CCA: The Proposed Algorithm

MSS-based CCA mechanism designed to mitigate the small buffer and high PER issues to improve overall TCP performance and bandwidth utilization over mmWave cellular networks. For instance, legacy CCAs deal with error loss states as congestion, therefore the CCA decreases the cwnd to mitigate the network congestion. However, this behaviour of CCAs is not efficient in mmWave cellular networks which are prone to channel implications such as error loss/PER. On the other hand, a small buffer leads to packet loss and forces CCA to decrease its cwnd which leads to throughput degradation. In mmWave cellular networks, the legacy CCA decreases its cwnd unnecessarily which causes degradation in the throughput and does not utilize available bandwidth efficiently. Subsequent subsections provide a more detailed explanation of the proposed algorithm.

5.3 MSS-Based CCA Mechanism

In legacy TCP, in the congestion avoidance stage, the cwnd is incremented by an increment value and depends on an Inc's value according to Equation (5.1). This Inc varies from one TCP to another TCP, for instance, in NewReno, the Inc is a predefined constant equal to 1. In CUBIC CCA, the Inc of cwnd is specified by a cubic function. For instance, in NewReno, the congestion window is increased by $1/cwnd$. On the other hand, in CUBIC, the increment value is determined as $C * (\Delta - \sqrt[3]{(\beta * cwnd)/C}$, where C is a preset constant, β is the multiplicative decrease factor, and Δ represents the elapsed time since the last loss.

MSS-based CCA increments its cwnd by a fraction during the congestion avoidance stage, similar to existing CCAs, but by a different Increment mechanism. However, it distinguishes itself by increasing cwnd through an Increment (Inc), forming a convex curve, unlike standard CCA. The primary contribution of MSS-based CCA is its unique cwnd growth function, relying on the Increment mechanism symbolized by the MSS and RTT, as illustrated in the following Equation. (5.6).

$$cwnd = cwnd + \frac{Inc}{cwnd} \tag{5.1}$$

Where $cwnd$ is the current cwnd and $\frac{Inc}{cwnd}$ is the increment value of the current congestion window in the congestion avoidance phase. As is known, MSS is equal to one segment size, as Equation (5.2) shows.

$$MSS = \frac{segments\ size}{number\ of\ segments} \tag{5.2}$$

Where $segments\ size$ is the size of sgements divided by $number\ of\ segments$.

Also, the variations in RTT can be utilized to measure the network's status (Afanasyev et al., 2010); (Wang and Crowcroft, 1992); (Brakmo and Peterson, 1995), (Jin et al., 2005); (Pan et al., 2023). Therefore, the Time Ratio is set as a function in three different RTTs, as Equation (5.3) shows. This Time Ratio has different values according to the network conditions.

$$Time\ Ratio = \frac{(RTT_{max} + RTT_{base})}{RTT_{cur}} \tag{5.3}$$

where RTT_{cur} is the current Round-Trip-Time, RTT_{base} is the minumum Round-Trip-Time, and RTT_{max} is the maximum Round-Trip-Time.

The magnitude of the maximum segment size is chosen alongside RTT as a parameter for incrementing cwnd during the congestion avoidance stage. MSS-based CCA calculates a dynamic parameter, the novel Inc, based on network states, as shown in Equation (5.4). This calculation involves multiplying Equation (5.2) with Equation (5.3), relying on MSS and Time Ratio values. The resulting value dictates the increment of cwnd accordingly, as Equation (5.5) shows.

$$F = \sqrt{MSS * Time\ Ratio} \tag{5.4}$$

Where MSS is the maximum segment size and the $Time\ Ratio$ is the ratio function between base, current, and maximum $RTTs$.

The novel increment (F) should dynamically create an adaptive convex-up curve as Figure 5.1 shows, decreasing the epoch time and maximizing the area under cwnd curve for optimal bandwidth utilization over mmWave cellular networks. The choice of a square root function to represent the novel Inc is based on fulfilling the adaptive increase requirement and its simplicity and suitability for integration into the core space of the Linux kernel.

Finally, Equation (5.1) becomes in the full formula as illustrated in Equation (5.5) and Equation (5.6) respectively to increase the cwnd in the congestion avoidance phase of

MSS CCA.

$$cwnd = cwnd + \frac{\sqrt{MSS * Time\ Ratio}}{cwnd} \tag{5.5}$$

$$cwnd = cwnd + \frac{F}{cwnd} \tag{5.6}$$

Where $cwnd$ is the current congestion window and $F/cwnd$ is the increment value of the congestion window in the congestion avoidance phase of the algorthim.

Like standard TCP CCAs, MSS-based CCA initiates the slow start stage, demonstrating an exponential increase by doubling cwnd in each RTT until reaching the slow start threshold (ssth). The cwnd grows gradually by triggering the congestion avoidance stage, preventing network congestion as the cwnd is increased incrementally rather than exponentially in the slow start stage. MSS-based CCA increases its cwnd by an Inc factor during this stage, as illustrated in the previous subsection, as Equation (5.6) shows, forming a convex curve. In packet loss detection, when the sender receives three Duplicate Acknowledgments (3 DACKs), MSS-based CCA reduces its cwnd by β as Algorithm 5.1 shows. However, if the timeout (RTO) is detected at any stage, MSS-based CCA resets its cwnd to the initial value and triggers a slow start stage again.

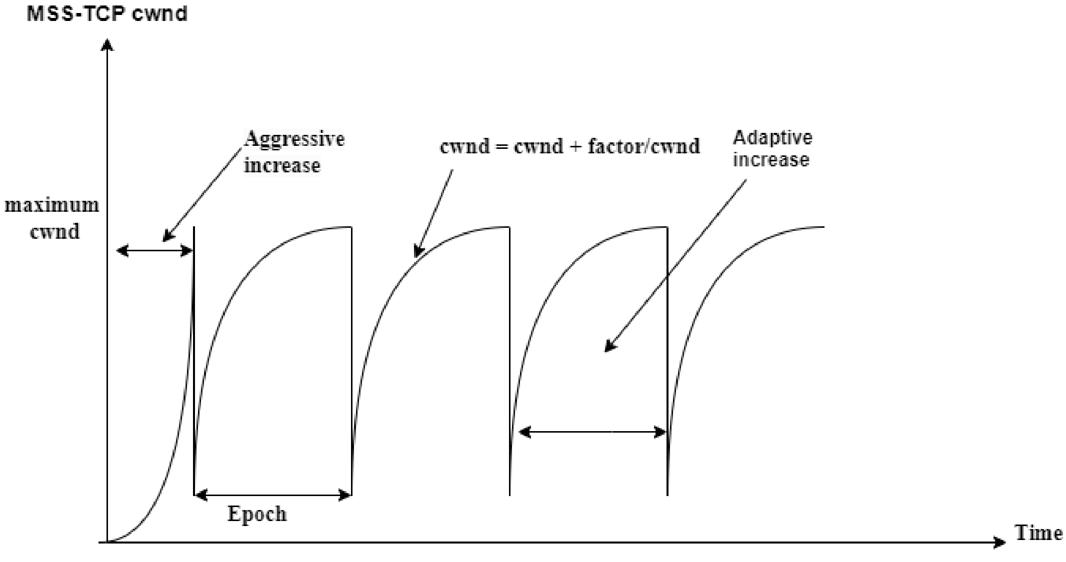


Figure 5.1: Cwnd Increase of MSS-Based CCA

Algorithm 5.1: MSS-Based CCA

```
Initialize: $cwnd$, $ssthresh$, $rttbase$, $rttcur$, $rttmax$;
if there is data to send then
    if RTO not expired then
        if no 3DACK then
            if $cwnd < ssthresh$ then
                $cwnd = cwnd + 1$;
            else
                $rttcur = Acktime - Sendtime$;
                where $Acktime$ is the time the acknowledgment is received at the source host, and $Sendtime$ is the time the packet was sent from the source host.
                if $rttbase > rttcur$ then
                    $rttbase = rttcur$;
                end
                if $rttmax < rttcur$ then
                    $rttmax = rttcur$;
                end
                $TimeRatio = \frac{(rttmax+rttbase)}{rttcur}$;
                $MSS = \frac{segmentsize}{numberofsegments}$;
                $F = \sqrt{MSS \times TimeRatio}$;
                $cwnd = cwnd + \frac{F}{cwnd}$;
            end
        else
            Multiplicative decrease;
        end
    else
        Slow start stage;
    end
end
```

5.4 MSS-Based CCA: Performance Evaluation

This chapter aims to demonstrate the MSS-based CCA, which enhances the bandwidth utilization of mmWave cellular networks. The ns-3 network simulator utilizes the mmWave module to evaluate the protocol's performance and compare its results with various TCP variants. The MSS-based CCA is implemented into ns-3 as a simulation module. The main goal of this work is to improve TCP performance in mmWave cellular networks to mitigate the small buffers and high PER issues.

5.4.1 Experiment Setup

Extensive simulation experiments are conducted using the well-known ns-3 network simulator to evaluate the proposed CCA by comparing its performance to the benchmark CCAs. The experimental setup is detailed in Chapter 3, and the specific simulation parameters are described in Table 3.2 of Chapter 3.

5.4.2 Results and Discussion

This section presents an analytical examination of the behavior exhibited by MSS-based CCA compared to benchmark CCAs. Additionally, the subsection showcases performance outcomes concerning cwnd fluctuations of MSS-based CCA, throughput, and latency to illustrate the impact of buffer size and PER on the overall performance.

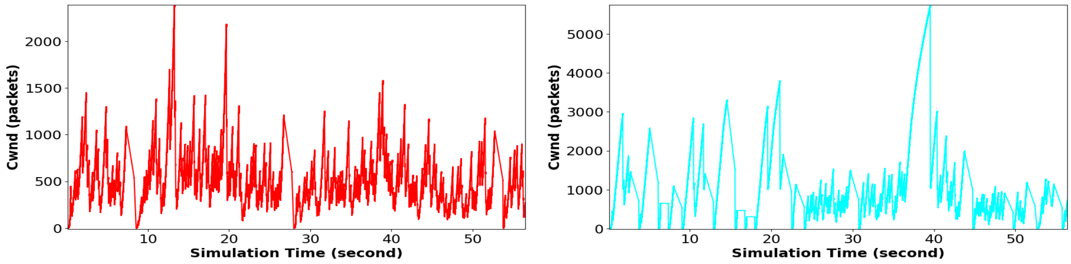
5.4.2.1 Congestion Window Fluctuation

Figures 5.2, 5.3, and 5.4 show the cwnd fluctuations of MSS-based CCA against different buffers and different PERs. Figure 5.2 illustrates that with a small buffer size, the cwnd behavior is significantly affected by varying PERs. Specifically, high PERs result in severe instability and low cwnd values due to frequent packet losses. Meanwhile, medium PERs cause less severe but still noticeable fluctuations and decreased cwnd values. Conversely, low PERs allow for higher and more stable cwnd levels, although the limited buffer still restricts the adjusting of cwnd. Furthermore, in the absence of packet errors, the cwnd grows more significantly and consistently, yet the constrained buffer size continues to limit the maximum achievable cwnd. Overall, small buffer sizes combined with higher PERs lead to substantial instability and restricted cwnd growth, whereas lower PERs and no errors improve performance but still face buffer size limitations.

Figure 5.3 demonstrates that when the buffer size matches the BDP, the cwnd size varies under different PERs. High PERs cause significant fluctuations and low cwnd values due to frequent congestion control interventions to adjust the cwnd. Medium PERs result in less severe but still noticeable instability and lower cwnd values. Small PERs lead to higher and more stable cwnd values, indicating better network performance. Without errors, the cwnd increases significantly and remains stable, fully utilizing the network capacity. Overall, matching the buffer size to the BDP improves network performance, especially in low PER scenarios.

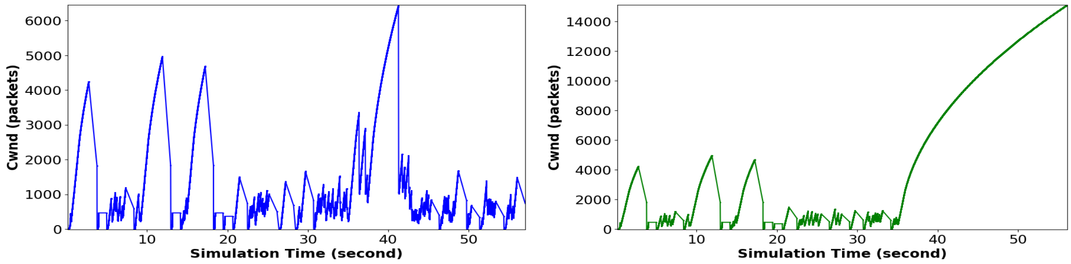
Figure 5.4 demonstrates that with an large buffer size, the cwnd behavior varies significantly with different PERs. High PERs, specifically, lead to frequent fluctuations and low cwnd values due to the mechanism of MSS CCA adjusting the cwnd according to the network conditions. In contrast, medium PERs cause less severe but still noticeable instability and lower cwnd values. Conversely, small PERs result in higher and more stable cwnd values, indicating better network performance. In the absence of packet errors, the cwnd increases significantly and consistently, fully utilizing the network's capacity.

In contrast to other TCPs, MSS-based CCA adjusts the cwnd according to network conditions. This adjustment is achieved by modifying the increment of cwnd using the MSS factor alongside calculations of RTT variations in the congestion avoidance stage of the algorithm. The cwnd adjustment varies based on the buffer size applied, showcasing MSS-based CCA's ability to react appropriately. Moreover, the protocol efficiently identifies the upper bound of the network during transitions from NLoS to LoS conditions when a small buffer and high PERs are applied, promptly utilizing available bandwidth by adjusting the cwnd and preventing network overflow. Cwnd adjustment for MSS-based CCA is depicted when high PER loss occurs in the network, illustrating MSS-based CCA's relative immunity to packet drops to some extent due to its congestion avoidance stage mechanism. This mechanism enables the protocol to operate around the maximum sending rate compared to other CCA variants, precisely adjusting its cwnd size under different buffers and PERs conditions.



(a) Cwnd vs high PER

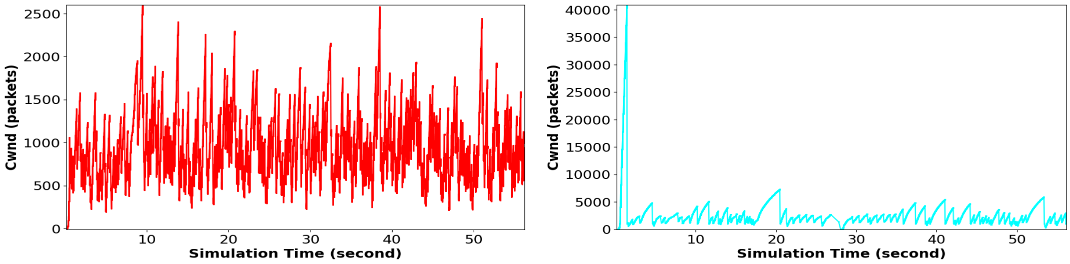
(b) Cwnd vs medium PER



(c) Cwnd vs small PER

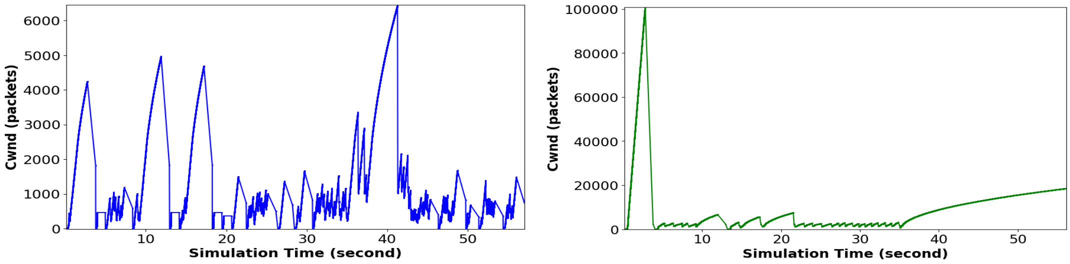
(d) Cwnd vs null PER

Figure 5.2: MSS CCA Cwnd Vs. Different PERs When The Buffer is Small (0.25MB)



(a) Cwnd vs high PER

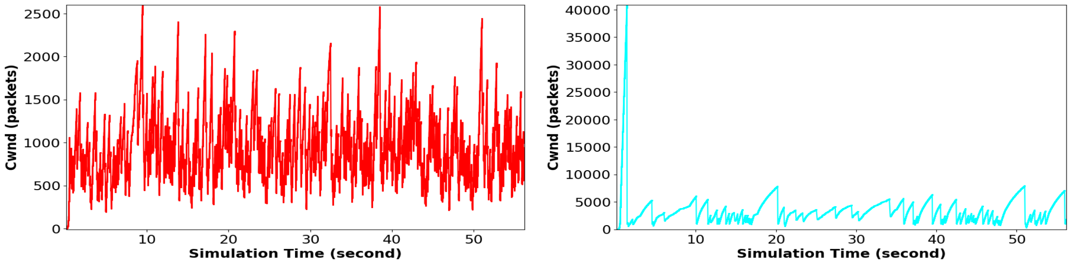
(b) Cwnd vs medium PER



(c) Cwnd vs small PER

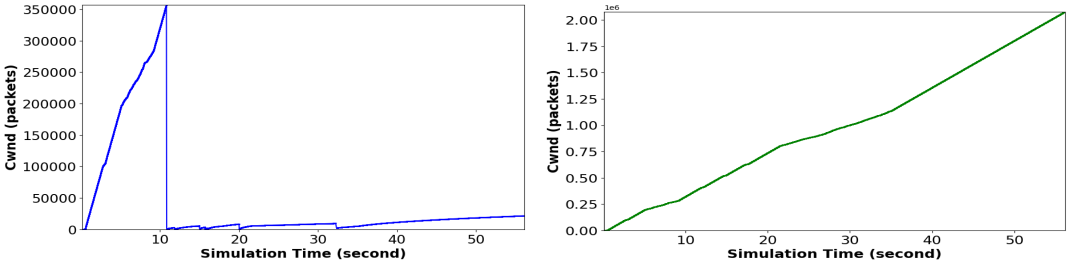
(d) Cwnd vs null PER

Figure 5.3: MSS CCA Cwnd Vs. Different PERs When The Buffer Matches BDP (2.5MB)



(a) Cwnd vs high PER

(b) Cwnd vs medium PER



(c) Cwnd vs small PER

(d) Cwnd vs null PER

Figure 5.4: MSS CCA Cwnd Vs. Different PERs When The Buffer is Large (20MB)

5.4.2.2 Average Throughput

Figures 5.5, 5.6, and 5.7 present the throughput results of the analyzed TCP algorithms across different buffer sizes and PERs within the examined network. MSS-based CCA consistently outperforms other CCAs under most conditions in terms of average throughput, as illustrated in the figures. This success is primarily attributed to its rapid cwnd growth facilitated by the novel Inc factor mechanism. Moreover, MSS-based CCA demonstrates lower sensitivity to varying buffer sizes and PERs compared to loss-based TCP and FB-TCP variants. These alternatives suffer significant performance deterioration with increasing PER due to their mechanisms for buffer filling, particularly when a small buffer and a high PER are applied in the scenario. This often leads to network congestion and subsequent packet drops, severely impacting their throughput. In contrast, MSS-based CCA effectively manages its cwnd growth, even in challenging network conditions, preventing the network from becoming congested and ensuring stable throughput. Its ability to swiftly adapt to changing network conditions and efficiently utilize available bandwidth makes it more robust and less prone to performance degradation than other TCP algorithms.

Overall, MSS-based CCA exhibits superior performance across various conditions, consistently outperforming loss-based, BBR, and FB-TCP algorithms as illustrated in Tables 5.1, 5.2, and 5.3 respectively according the applied buffer. The results demonstrate that MSS-CCA outperforms the compared CCAs in terms of average throughput and latency. MSS-CCA achieves a significant improvement in throughput, surpassing loss-based TCP algorithms by 45.43% compared to BBR and 20.32% compared to FB-TCP in certain PER scenarios when the buffer size matches the BDP. Its capability to adapt and maintain efficient throughput in different buffer sizes and PER scenarios underscores its robustness and effectiveness in diverse network environments. This adaptability and efficiency highlight MSS-based CCA as a reliable choice for maintaining high throughput and network stability under different buffers and PER scenarios.

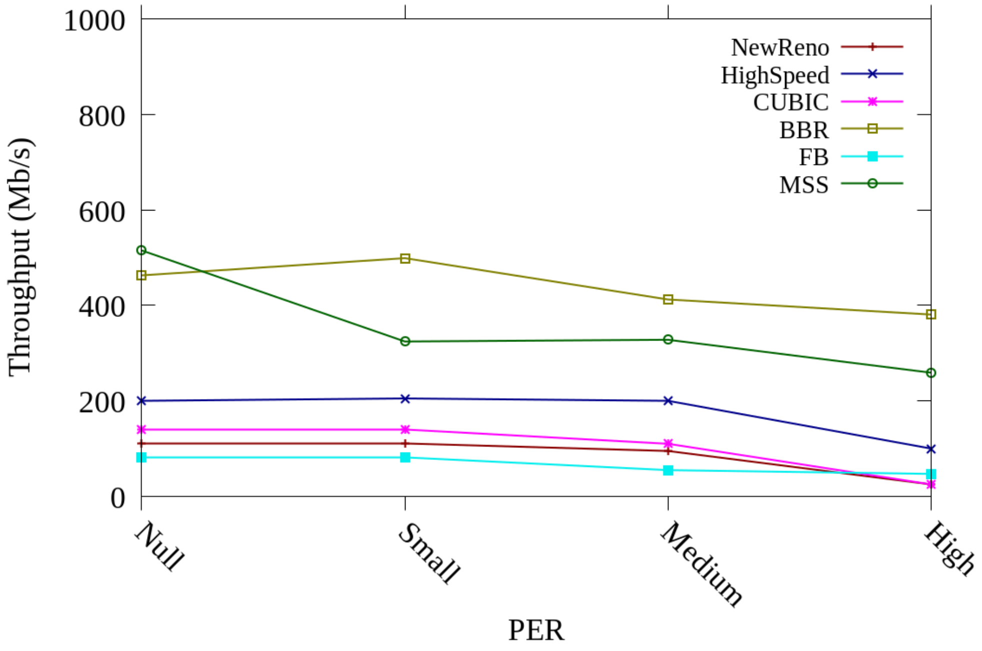


Figure 5.5: Average Throughput of TCP Variants Vs. Different PERs When The Buffer is Small (0.25MB)

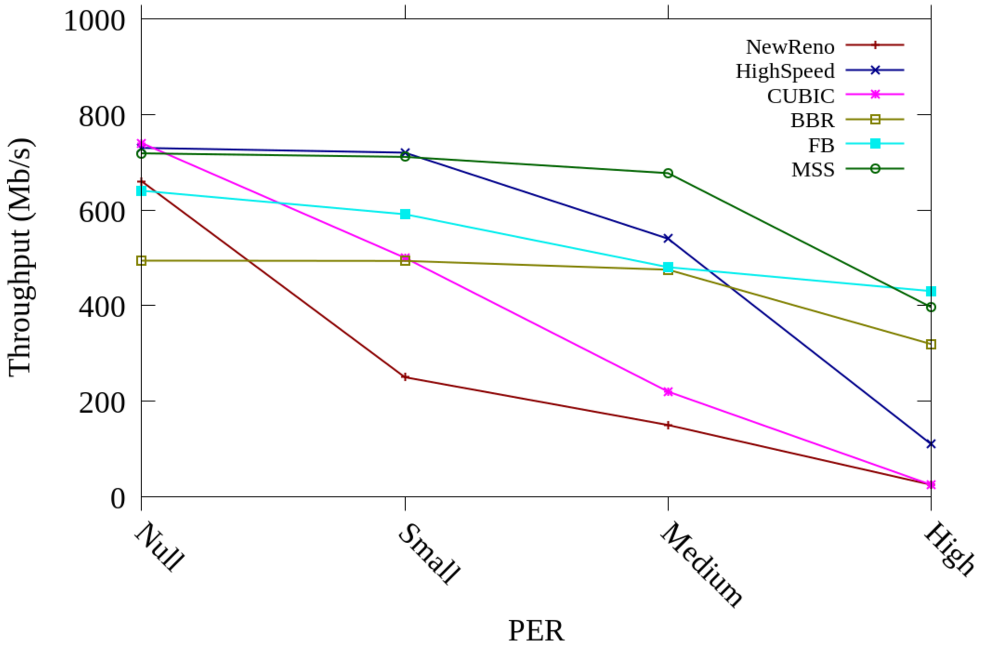


Figure 5.6: Average Throughput of TCP Variants Vs. Different PERs When The Buffer Matches BDP (2.5MB)

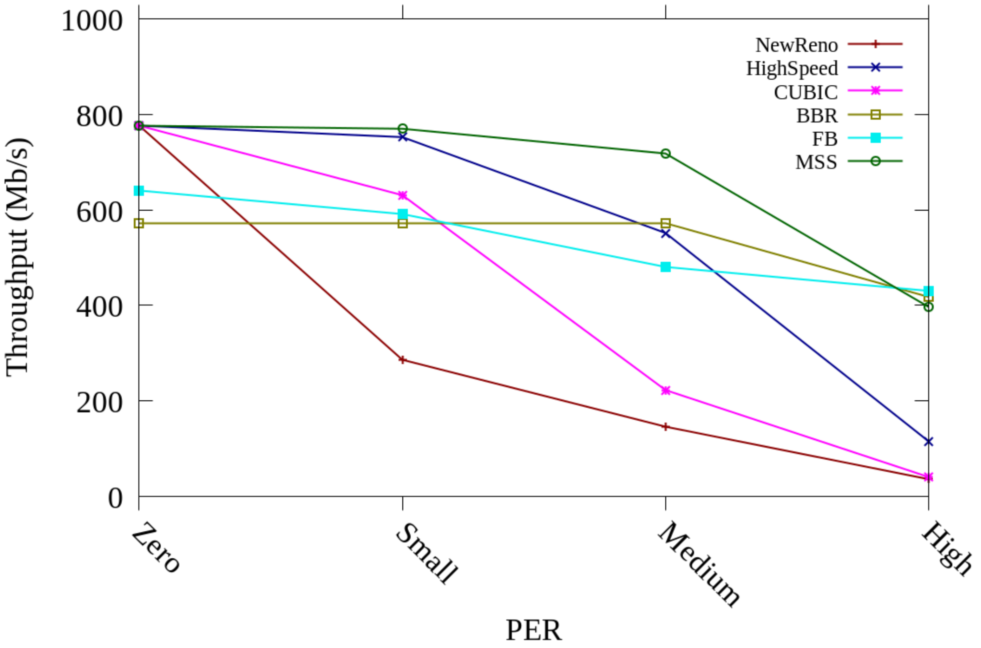


Figure 5.7: **Average Throughput of TCP Variants Vs. Different PERs When The Buffer is Large (20MB)**

Table 5.1: **Percentage improvement of MSS-CCA throughput compared to benchmark protocols when the buffer is small (0.25MB)**

TCP	**PER**			
	Null	**Small**	**Medium**	**High**
NewReno	368.07	195.70	244.11	935.52
HighSpeed	157.44	58.29	63.95	158.88
Cubic	267.77	131.65	198.09	935.52
BBR	11.23	-34.97	-20.47	-32.02
FB-TCP	526.21	294.32	498.54	450.51

Table 5.2: **Percentage improvement of MSS-CCA throughput compared to benchmark protocols when the buffer matches BDP (2.5MB)**

TCP	**PER**			
	Null	**Small**	**Medium**	**High**
NewReno	8.87	184.44	451.46	1486.72
HighSpeed	-1.56	-1.23	25.37	261.53
Cubic	-2.89	42.22	208.27	1486.72
BBR	45.43	44.37	42.55	24.26
FB-TCP	12.23	20.32	41.01	-7.83

Table 5.3: Percentage improvement of MSS-CCA throughput compared to benchmark protocols when the buffer is large (20MB)

TCP	**PER**			
	Zero	**Small**	**Medium**	**High**
NewReno	0	169.20	392.69	1005.10
HighSpeed	0	2.33	30.16	244.93
Cubic	0	22.13	222.18	777.88
BBR	35.61	34.52	25.45	-5.02
FB-TCP	21.21	30.23	49.47	-7.80

5.4.2.3 Average latency

Figures 5.8, 5.9, and 5.10 show latency performance of MSS-based CCA demonstrates its efficacy in managing average latency with minimal variations across different buffer sizes and PERs. This stability is largely attributed to its Inc factor mechanism, which allows MSS-based CCA to adaptively adjust its transmission rate based on network conditions, minimizing latency spikes that other TCP variants might experience. When compared to BBR, MSS-based CCA typically results in lower latency, particularly in scenarios with high PER. This indicates that MSS-based CCA handles packet loss and retransmissions more effectively, maintaining better performance under challenging network conditions. Additionally, MSS-based CCA consistently achieves lower latency across various buffer sizes compared to BBR, showcasing its robustness in diverse deployment scenarios. While MSS-based CCA may exhibit slightly higher latency than some TCP variants in specific PER cases, this marginal difference is acceptable given its significant improvements in throughput and bandwidth utilization. The algorithm's design ensures efficient data transfer and effective bandwidth usage, balancing high throughput with competitive latency performance. Thus, MSS-based CCA is well-suited for environments where both low latency and high throughput are crucial, making it a promising choice for mmWave cellular network deployments when the server is located at the edge and closer to the base station.

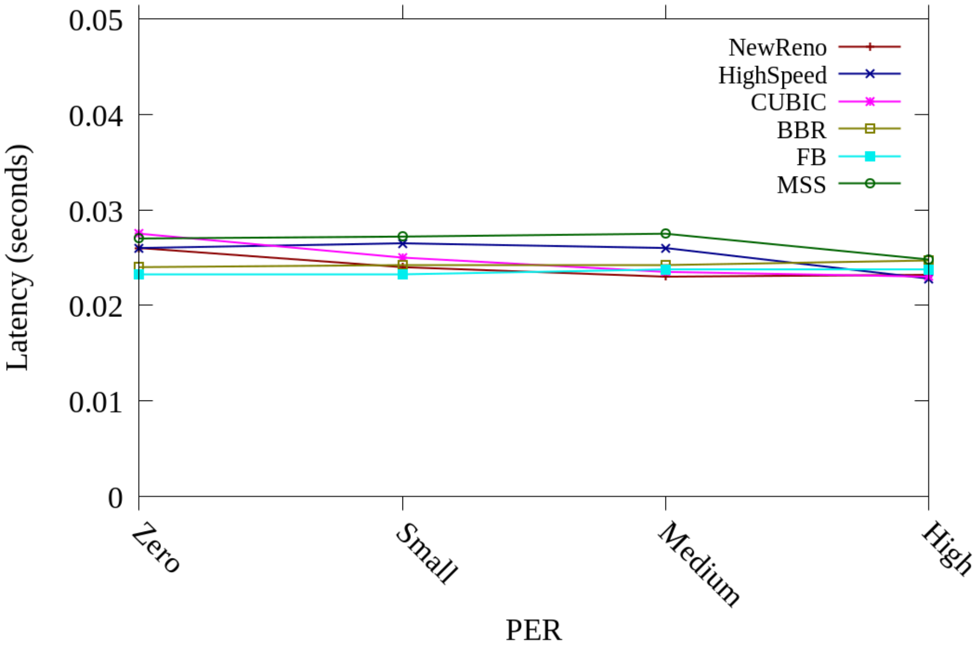


Figure 5.8: Average Latency of TCP Variants Vs. Different PERs When The Buffer is Small (0.25MB)

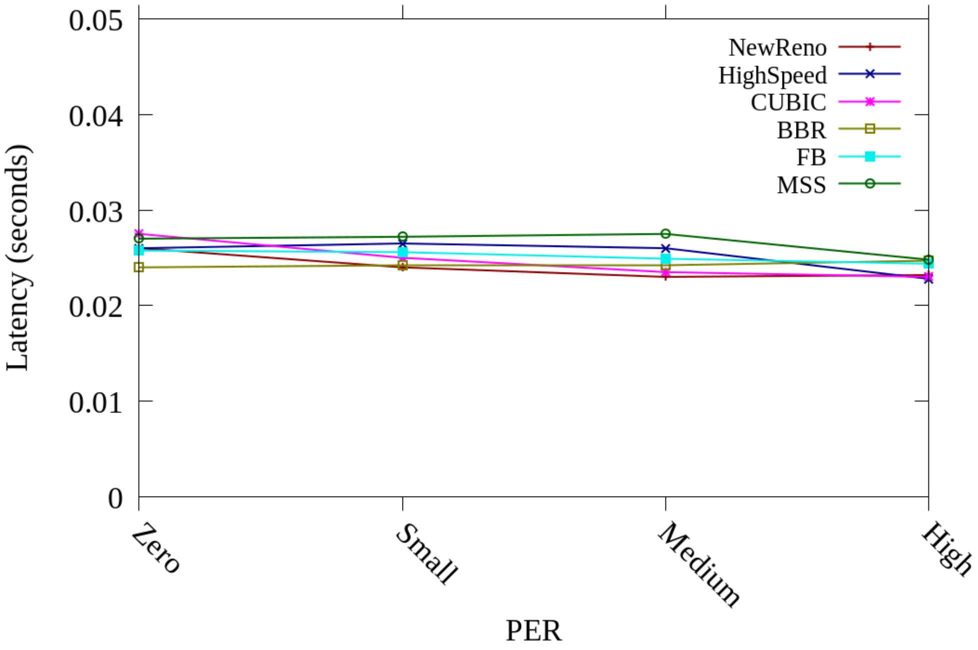


Figure 5.9: Average Latency of TCP Variants Vs. Different PERs When The Buffer Matches BDP (2.5MB)

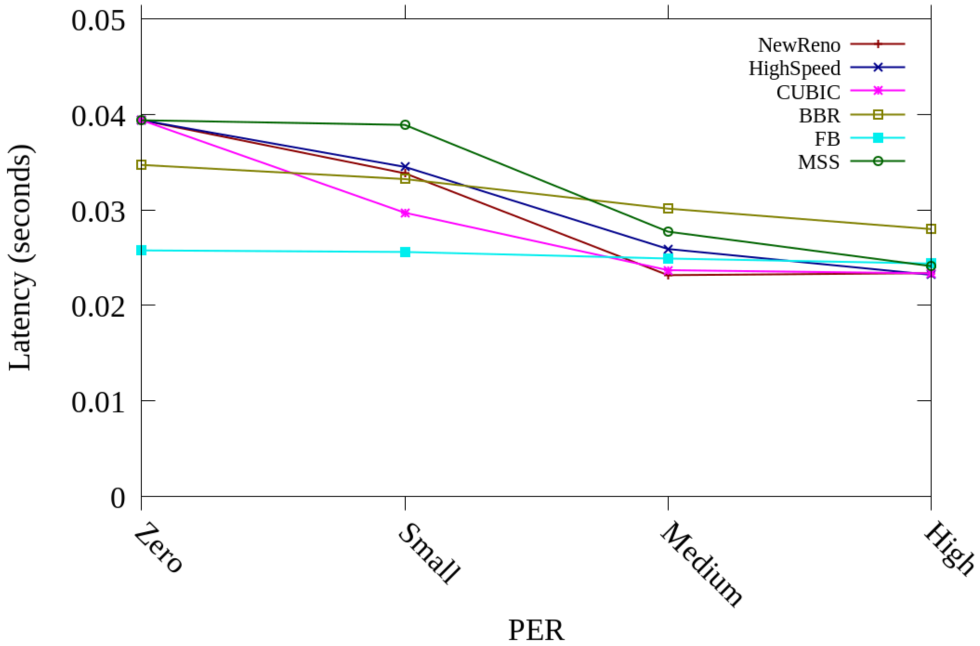


Figure 5.10: Average Latency of TCP Variants Vs. Different PERs When The Buffer is Large (20MB)

Table 5.4: Percentage improvement of MSS-CCA latency compared to benchmark protocols when the buffer is small (0.25MB)

TCP	**PER**			
	Zero	**Small**	**Medium**	**High**
NewReno	3.85	13.33	19.57	6.90
HighSpeed	3.85	2.64	5.77	8.77
CUBIC	-1.82	8.80	17.02	7.83
BBR	12.50	12.40	13.64	0.40
FB-TCP	16.08	16.94	15.71	4.33

Table 5.5: Percentage improvement of MSS-CCA latency compared to benchmark protocols when the buffer matches BDP (2.5MB)

TCP	**PER**			
	Zero	**Small**	**Medium**	**High**
NewReno	3.85	13.33	19.57	6.90
HighSpeed	3.85	2.64	5.77	8.77
CUBIC	-1.82	8.80	17.02	7.83
BBR	12.50	12.40	13.64	0.40
FB-TCP	4.83	6.30	10.44	1.74

Table 5.6: Percentage improvement of MSS-CCA latency compared to benchmark protocols when the buffer is large (20MB)

TCP	PER			
	Zero	Small	Medium	High
NewReno	0.00	14.34	44.17	6.38
HighSpeed	0.00	12.12	28.91	7.19
CUBIC	0.00	30.20	40.95	6.48
BBR	13.44	16.42	10.88	11.16
FB-TCP	52.86	51.22	34.09	1.99

5.5 Summary

In this chapter, a novel CCA named MSS-based CCA is introduced and evaluated over an mmWave cellular network, with an edge server considered in the setup. MSS-based CCA is proposed to address the small buffer issue in the presence of PER. The primary innovation of this proposed algorithm lies in incorporating the Inc factor mechanism, specifically tailored to MSS. The necessity for MSS-CCA arises from the limitations of existing TCPs in achieving optimal bandwidth utilization over mmWave cellular networks, particularly under small buffer regimes and varying PERs. Subsequently, the MSS-CCA module was integrated into the ns-3 network simulator to evaluate its performance, comparing it against baseline TCPs over mmWave cellular networks.

Intensive simulation experiments assessed MSS-CCA performance against loss-based (NewReno, HighSpeed, CUBIC), model-based (BBR), and fuzzy-based (FB-TCP) TCPs. The results reveal that MSS-CCA achieves higher bandwidth utilization than existing TCPs while maintaining low latency and demonstrating lower sensitivity to buffer size and PER variation. Moreover, it demonstrates higher efficiency and stability compared to benchmark protocols (NewReno, HighSpeed, CUBIC, BBR) by (1486.72, 261.53, 1486.72, 24.26)%, respectively, under high PER conditions, while maintaining low latency.

Importantly, MSS-CCA, functioning as a sender-side TCP module, introduces no changes to the receiver side or the network routers, utilizing and featuring a novel congestion avoidance algorithm driven by the MSS factor mechanism.

CHAPTER 6

MRVHS-BASED CCA: CONGESTION CONTROL ALGORITHM TO MITIGATE SERVER LOCATION IMPACT IN MMWAVE CELLULAR-TO-CLOUD NETWORKS

6.1 Introduction

Naturally, cellular networks are connected to the servers, whether the server is located at the edge of a base station (gNB) or in a remote location through the Internet cloud, as in Figure 6.1 shows. This capability is crucial for seamless user experiences, as in everyday use, cellular network users may need to access applications hosted on servers located in distant regions. Ensuring that the network can adapt to these varying server locations automatically is essential for maintaining optimal performance, minimizing latency, and providing reliable connectivity. This adaptability is essential in modern applications that demand high data rate and low latency such as augmented reality, virtual reality, and cloud gaming. Therefore, an efficient and adaptive approach to managing these connections is critical for the robustness and effectiveness of cellular networks to meet the demands of the era. However, legacy TCPs encounter issues due to variations in RTT over cellular-to-cloud networks. In such cases, the performance of these TCPs can be significantly degraded. This identified research gap has motivated us to develop a new congestion avoidance mechanism, based on MSS and RTT variations, and integrated with HighSpeed-TCP. The proposed approach is detailed in Section 6.2. To evaluate its effectiveness, the proposed protocol is compared with five benchmark TCPs: NewReno, HighSpeed, CUBIC, BBR, and FB-TCP, under a range of conditions with different PER. Furthermore, simulation experiments focus on KPIs such as cwnd fluctuations, throughput, and latency, providing a detailed analysis of the performance of the protocol.

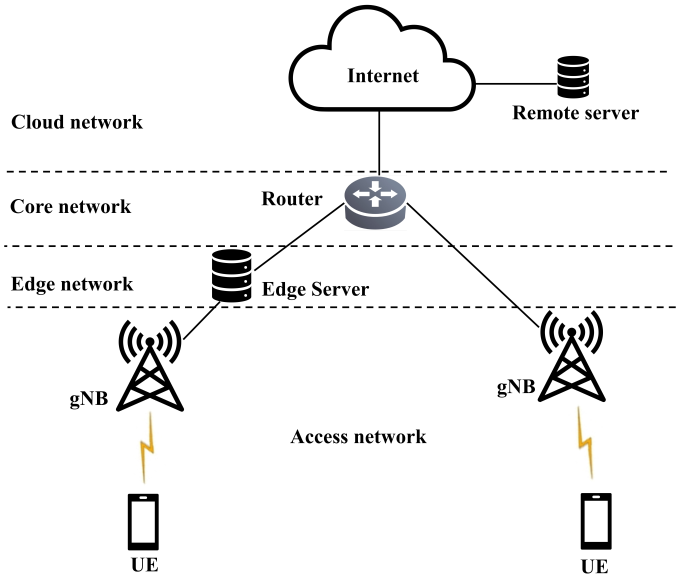


Figure 6.1: Edge and Remote Server Networks

6.2 MRVHS-Based CCA: The Proposed Algorithm

Before discussing the proposed algorithm, it is essential first to discuss the underlying mechanism of the foundation of our proposed TCP. This discussion will provide the necessary foundation to fully comprehend the workings of the proposed TCP, which is based on the principles of HighSpeed-TCP while incorporating MSS and RTT variations. In 2003, Floyd introduced HighSpeed-TCP (Floyd, 2003), explicitly crafted to handle large $cwnd$ sizes. This variant aimed to overcome the shortcomings of standard TCP, which demonstrated subpar performance in high-speed wired networks. HighSpeed-TCP functions as a loss-based TCP without altering the behavior of standard TCP, thus mitigating the risk of congestion collapse. HighSpeed-TCP operates as a modification on the sender side, adjusting cwnd by $\alpha/cwnd$ for increases and $\beta/cwnd$ for decreases by predefined values. HighSpeed updates its $cwnd$ after receiving each ACK in an RTT by $\alpha/cwnd$, as shown in Equation (6.1), if congestion is not detected, and otherwise decreases the $cwnd$ by $\beta/cwnd$ if the congestion is detected as shown in Equation (6.2). The values of cwnd span from 1 to 73, with the parameter $\alpha/cwnd$. This variation is contingent on the specific range of $cwnd$, varying from below or equal to 38 packets to greater than 84k packets. This adaptive approach caters to diverse wired network conditions, ensuring considerable performance (Afanasyev et al., 2010; Lar and Liao, 2013). This functionality is disrupted within the cellular networks due to non-congestion states. HighSpeed-TCP is a loss-based TCP protocol, as its performance deteriorates.

The underlying reason is that loss-based TCPs consider non-congestion states as packet loss. It is interpreted as a sign of network congestion, which can activate the decrement mechanism of the HighSpeed TCP to decrease the cwnd. However, the non-congestion states in mmWave cellular networks may arise from factors like signal blockage, such as buildings and trees, or trigger the RTO due to long delays in RTT over cellular-to-cloud networks. In HighSpeed CCA, the variation of RTT is not taken into account when updating the cwnd during the congestion avoidance stage. This approach leads to suboptimal performance in utilizing the available bandwidth under cellular-to-cloud networks. HighSpeed-TCP does not dynamically adapt to changes in RTT, which is particularly problematic given the inherent RTT variability in cellular-to-cloud networks involving a remote server.

$$cwnd = cwnd + \frac{\alpha}{cwnd} \tag{6.1}$$

where $cwnd$ is the congestion window, α is a parameter in $\alpha/cwnd$ value to increase the congestion window in HighSpeed CCA.

$$cwnd = cwnd - \frac{\beta}{cwnd} \tag{6.2}$$

where $cwnd$ is the congestion window, β is a parameter in $\beta/cwnd$ value to decrease the congestion window in HighSpeed CCA.

To enable TCP to operate adaptively over cellular-to-cloud networks involving a remote server, the proposed MRVHS-CCA integrates both RTTV and MSS mechanisms in Equations (6.3) and (6.4) into HighSpeed-TCP, as illustrated in Algorithm 6.1. Consequently, instead of updating the cwnd by preset values, the MRVHS-based CCA updates its cwnd during the congestion avoidance stage, as outlined in Equations (6.5) and (6.6), respectively. On the other hand, MRVHS decreases the cwnd whenever three duplicate acknowledgments (3DACKs) are received, following Equation (6.2). For instance, Figure 6.2 shows a comparison of the cwnd between MRVHS and HighSpeed CCAs. It is evident that MRVHS-CCA effectively utilizes the available bandwidth through its novel mechanism, optimizing the cwnd to enhance TCP performance and increase system throughput. Furthermore, the MRVHS-based CCA source code exhibits lower complexity compared to HighSpeed CCA while simultaneously maintaining higher performance.

$$TimeRatio = \frac{RTT_{max} + RTT_{base}}{RTT_{cur}} \tag{6.3}$$

where RTT_{cur} is the current Round-Trip-Time, RTT_{base} is the minumum Round-Trip-Time, and RTT_{max} is the maximum Round-Trip-Time.

$$MSS = \frac{segments\ size}{number\ of\ segments} \tag{6.4}$$

Where $segments\ size$ is the size of sgements divided by $number\ of\ segments$.

$$F = \sqrt{TimeRatio \times MSS} \tag{6.5}$$

Where MSS is the maximum segment size and the $Time\ Ratio$ is the ratio function between base, current, and maximum $RTTs$.

$$cwnd = cwnd + \frac{F}{cwnd} \tag{6.6}$$

where $cwnd$ is the congestion window, F is a parameter in $F/cwnd$ value to increase the congestion window in HighSpeed CCA.

Algorithm 6.1: MRVHS-based CCA

```
Initialize: $cwnd$, $ssthresh$, $\beta$, $rttbase$, $rttcur$, $rttmax$;
if there is data to send then
    if RTO not expired then
        if no 3DACK then
            if $cwnd < ssthresh$ then
                $cwnd = cwnd + 1$;
            else
                $rttcur = Acktime - Sendtime$;
                if $rttbase > rttcur$ then
                    $rttbase = rttcur$;
                end
                if $rttmax < rttcur$ then
                    $rttmax = rttcur$;
                end
                $TimeRatio = \frac{(rttmax+rttbase)}{rttcur}$;
                $MSS = \frac{segmentsize}{numberofsegments}$;
                $F = \sqrt{MSS \times TimeRatio}$;
                $cwnd = cwnd + \frac{F}{cwnd}$;
            end
        else
            $cwnd = cwnd - \frac{\beta}{cwnd}$;
        end
    else
        Slow start stage;
    end
end
```

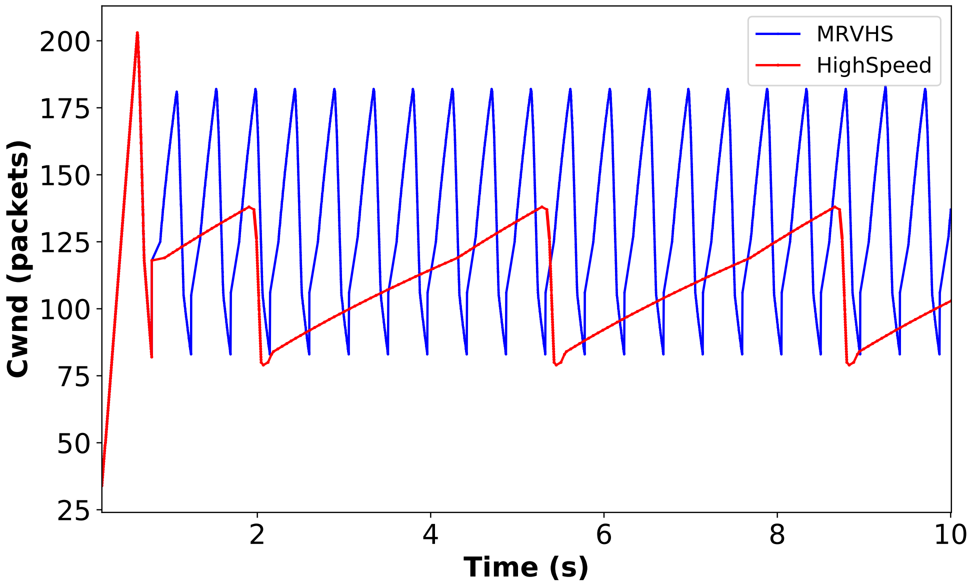


Figure 6.2: MRVHS and HighSpeed Congestion Windows Comparison

6.3 MRVHS-Based CCA Scheme

Figure 6.3 depicts the MRVHS-based CCA within a TCP framework, consisting of sending and receiving sides, each operating in the Transportation Layer and dynamically adjusting the cwnd to optimize data transmission based on real-time feedback. Specifically, the sending side utilizes several metrics such as RTTV, MSS, DACks counter, RTO counter, and SSTH to monitor the network state and control the network congestion. Consequently, each parameter plays a crucial role in determining the congestion level and adjusting cwnd accordingly, ensuring efficient and stable data transmission. Furthermore, the MRVHS-based CCA uses two primary mechanisms: cwnd Increase Mechanism and cwnd Decrease Mechanism. The increase mechanism expands cwnd when conditions are favorable, while the decrease mechanism shrinks it when congestion is detected, thereby maintaining network stability and optimal throughput. On the other hand, the receiving side generates ACKs for incoming packets, which are then sent back to the sender. Thus, these ACKs provide real-time feedback, helping the sender adapt cwnd to current network conditions accordingly.

Moreover, the data and ACK flow between sender and receiver form a continuous feedback loop, enabling the sender to dynamically adjust cwnd based on real-time network conditions, ensuring prompt responses to congestion or high delays. Compared to legacy TCPs protocols, MRVHS-CCA improves by incorporating real-time metrics and adaptive control using MSS and RTTV variations within HighSpeed-TCP, making it more suitable for variable and high-speed networks, such as mmWave cellular networks. In summary, MRVHS-CCA is a robust congestion control solution, effectively balancing high throughput and maintaining low latency.

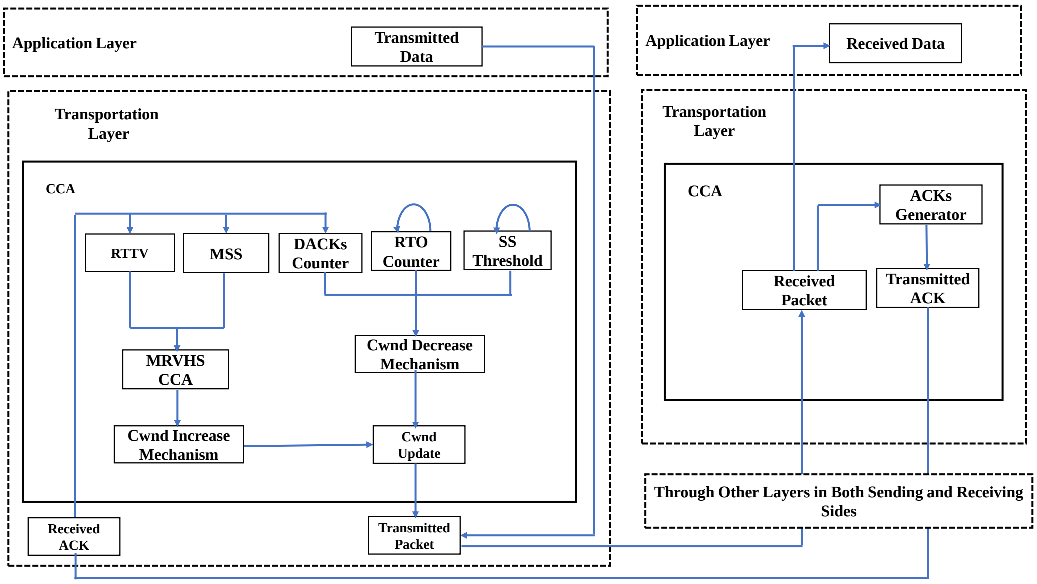


Figure 6.3: MRVHS-Based CCA Scheme

6.4 MRVHS-Based CCA: Performance Evaluation

For performance evaluation, we integrated the proposed MRVHS-CCA module into the ns-3 network simulator to conduct extensive simulation experiments. This integration allowed us to thoroughly assess the effectiveness and efficiency of MRVHS-CCA over cellular-to-cloud networks involving a remote server. The following subsections provide detailed information about the simulation setup, including the parameters used, the network topologies tested, and the specific conditions under which the experiments were performed.

6.4.1 Experiment Setup

Extensive simulation experiments are conducted using the well-known ns-3 network simulator to evaluate the proposed TCP by comparing its performance to benchmark TCPs. As shown in Figure 6.4, the network includes a UE communicating with a data center server through the cloud. The UE connects via cellular networks through a gNB station. To simulate a cellular-to-cloud involving a remote server in the network, the one-way propagation delay from the gNB to the server was adjusted to 40 ms, resulting in a minimum RTT of 80 ms.

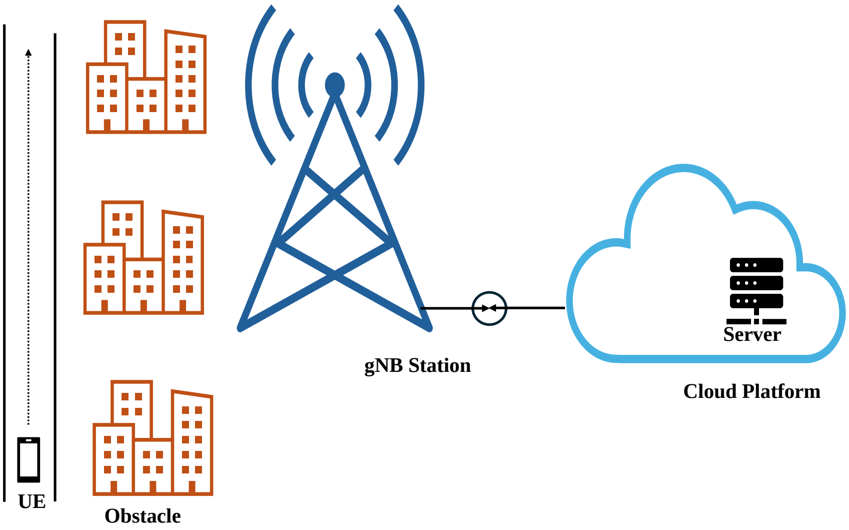


Figure 6.4: Network Topology

The scenario involves the UE moving in and out of LoS cases with the gNB station, transitioning to NLoS, stopping briefly, then moving again through LoS and NLoS, before stopping for a duration, and finally moving away from the gNB station and stopping. Some buildings are simulated as obstacles in the scenario to mimic NLoS conditions. This scenario is implemented to investigate the effects of transitioning between LoS and NLoS, the impact of stopping and resuming movement, and the influence of distance between the UE and the gNB station on TCP performance over cellular-to-cloud networks. For more detail on simulation parameters used, refer to Table 3.2 in Chapter 3.

6.4.2 Results and Discussion

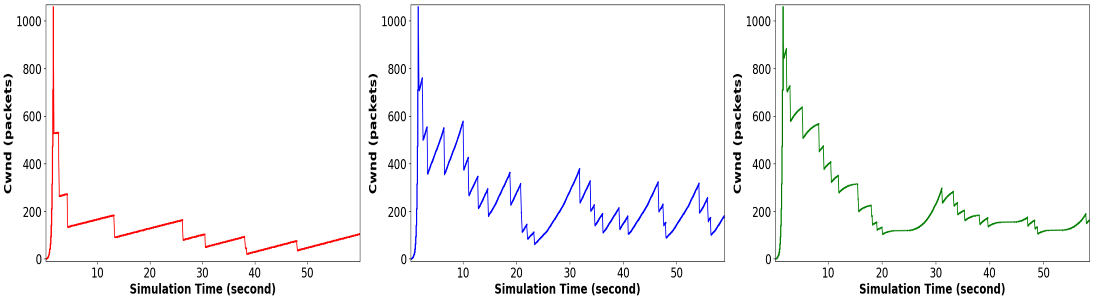
This section presents the simulation experiment results and a discussion of the behavior of MRVHS-CCA compared to benchmark TCPs. Additionally, it showcases performance outcomes related to cwnd fluctuations, throughput, and latency to illustrate the impact of PER on the overall performance of the proposed and benchmark TCPs over cellular-to-cloud networks in the presence of a remote server scenario.

6.4.2.1 Congestion Window Fluctuation

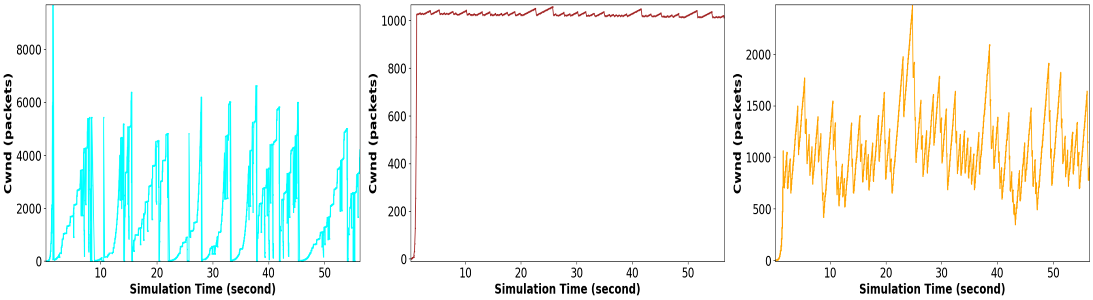
The congestion window size determines the rate at which data can be transmitted without requiring the corresponding ACK, as long as its sequence number falls within the sliding window. This is because the estimated in-flight data must be smaller than both the sender congestion window size and the receiver window size. As a result, the window size is set to the minimum of these two values. Therefore, a larger congestion window size and fewer loss events are indicators of better congestion control, which in turn leads to higher throughput and more efficient utilization of available bandwidth.

To compare the proposed algorithm with state-of-the-art protocols, we evaluated the congestion window size, as shown in Figures. 6.5, 6.6, 6.7, and 6.8. The figures show the cwnd behavior of CCAs over the simulation time under remote server-cloud scenarios with different conditions, such as movement, stopping, LoS, NLoS, walking, driving, and varying distance between UE and gNB station. MRVHS-CCA demonstrates the highest cwnd magnitude, varying from 500 to 2000 packets in high PER scenarios, as shown in Figure 6.5f, whereas HighSpeed-TCP's cwnd ranges from 100 to 400 packets, as depicted in Figure 6.5b. In contrast, Figure 6.5a shows that the cwnd of NewReno is limited to between 100 and 200 packets, whereas Figure 6.5c indicates that the cwnd of CUBIC hovers around 300 packets. The smaller cwnd sizes in loss-based CCAs are attributed to their sensitivity to packet loss, particularly in the presence of high PER in the network. However, the cwnd of BBR fluctuates between zero and 5000 packets, as shown in Figure 6.5d whereas FB-TCP cwnd swings around 1000 packets as Figure 6.5e.

The cwnd results show that MRVHS CCA adjusts the cwnd to values distinct from the lower cwnd typical of loss-based CCAs and the higher cwnd seen in BBR. MRVHS-CCA outperforms other TCP variants in terms of bandwidth utilization in the worst case of a high PER scenario over cellular-to-cloud networks in the presence of remote servers. This behavior results from the combination of RTTV and MSS mechanisms, which provide MRVHS-CCA with the flexibility to handle data transmission in remote server scenarios over cellular-to-cloud networks. This adjusting mechanism results in higher throughput and better utilization of the available bandwidth in the applied network.

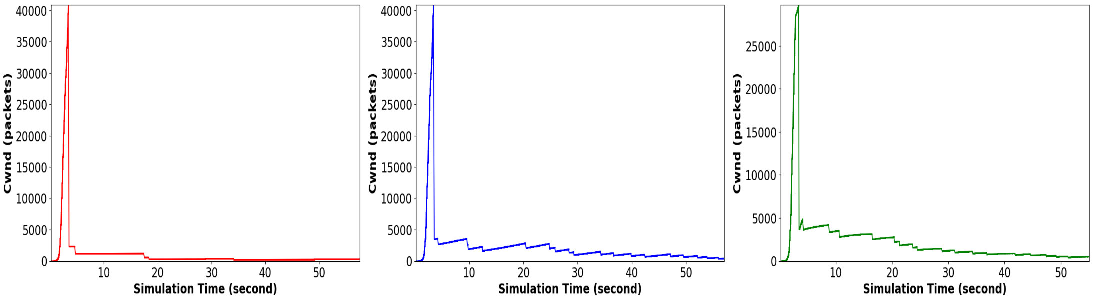


(a) NewReno (b) HighSpeed (c) CUBIC

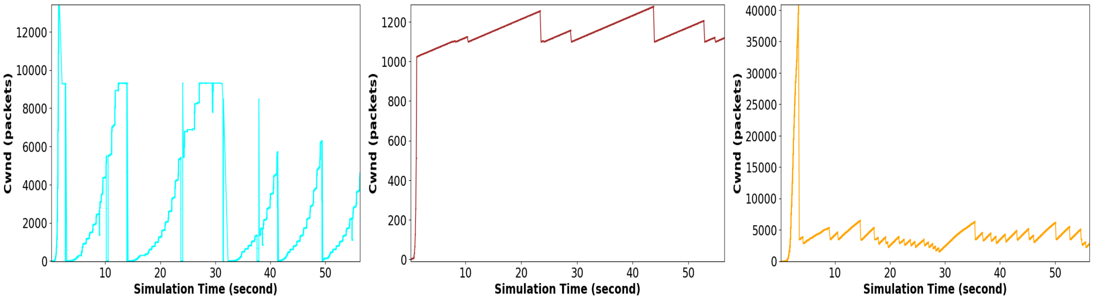


(d) BBR (e) FB-TCP (f) MRVHS

Figure 6.5: Cwnd Analysis of TCP Variants Vs. High PER

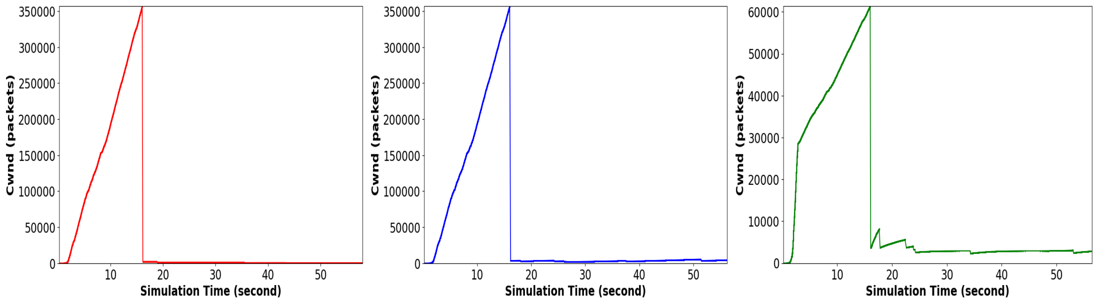


(a) NewReno (b) HighSpeed (c) CUBIC

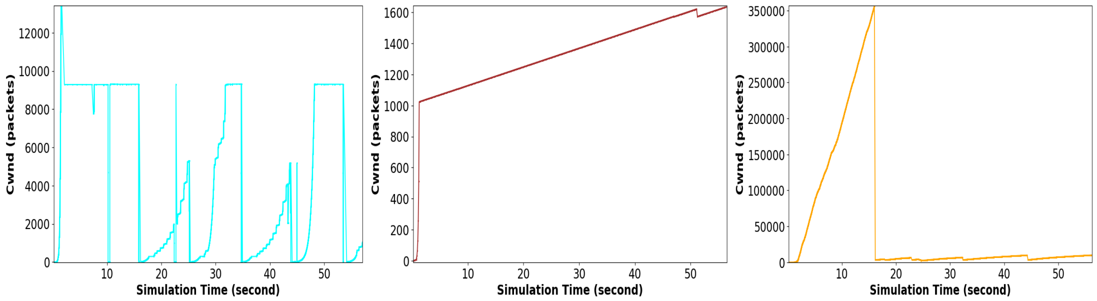


(d) BBR (e) FB-TCP (f) MRVHS

Figure 6.6: Cwnd Analysis of TCP Variants Vs. Medium PER

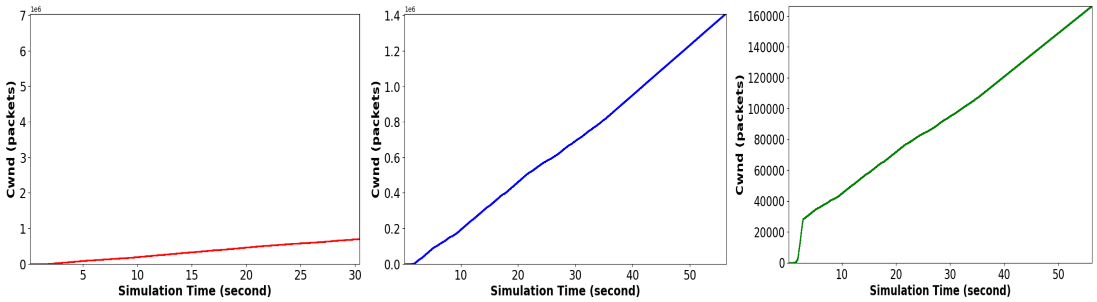


(a) NewReno (b) HighSpeed (c) CUBIC

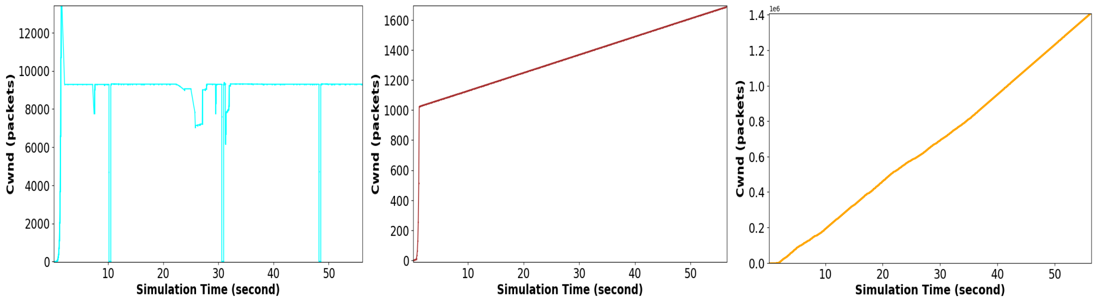


(d) BBR (e) FB-TCP (f) MRVHS

Figure 6.7: Cwnd Analysis of TCP Variants Vs. Small PER



(a) NewReno (b) HighSpeed (c) CUBIC



(d) BBR (e) FB-TCP (f) MRVHS

Figure 6.8: Cwnd Analysis of TCP Variants Vs. Null PER

6.4.2.2 Average Throughput

The throughput results for different PER scenarios demonstrate that MRVHS-CCA consistently achieves high average throughput across all PER scenarios when a remote server is employed in the cellular-to-cloud network. As shown in Fig. 6.9 and Table 6.1, MRVHS-CCA outperforms all five competing CCAs over PER scenarios. It achieves the highest average throughput, followed by HighSpeed-TCP, whereas the performance of other CCAs is significantly lower, particularly FB-TCP in the null and small PER scenarios. Additionally, Fig. 6.10 illustrates the instantaneous throughput of TCP variants over the simulation time of the experiment. As PER increases, the advantage of MRVHS-CCA becomes increasingly apparent due to the functionality of the MRVHS-based mechanism. Noticeably, the throughput increases in direct proportion to changes in the congestion window size. Thus, integrating the attributes of RTTV and MSS techniques into the HighSpeed-TCP is highly beneficial and promising for implementation in real operating systems. MRVHS-CCA effectively meets the high data rates and low latency of mobile applications over cellular-to-cloud networks.

6.4.2.3 Average Latency

In addition to achieving higher throughput compared to TCP variants across all scenarios with varying PERs, MRVHS-CCA maintains comparable latency to most other protocols as Fig. 6.11 shows, with only slight differences compared to loss-based and fuzzy-based protocols, as shown in Fig. 6.12. MRVHS even achieves lower latency in some cases, particularly in medium and small PERs scenarios compared to BBR. This demonstrates the protocol's effectiveness and suitability for applications requiring high speed and minimal delay within remote servers over cellular-to-cloud networks.

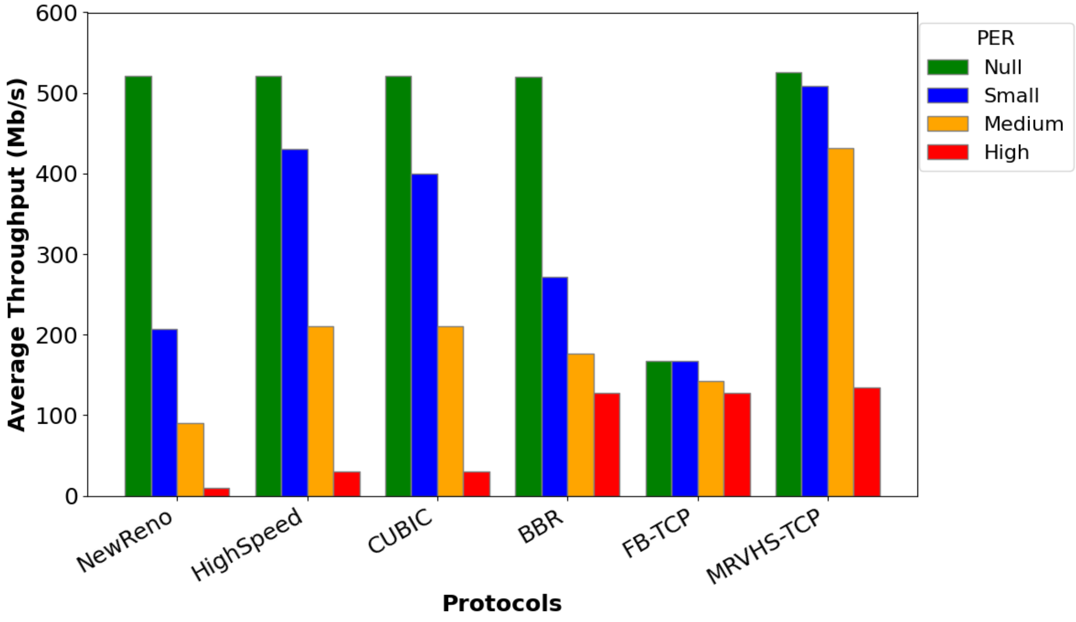
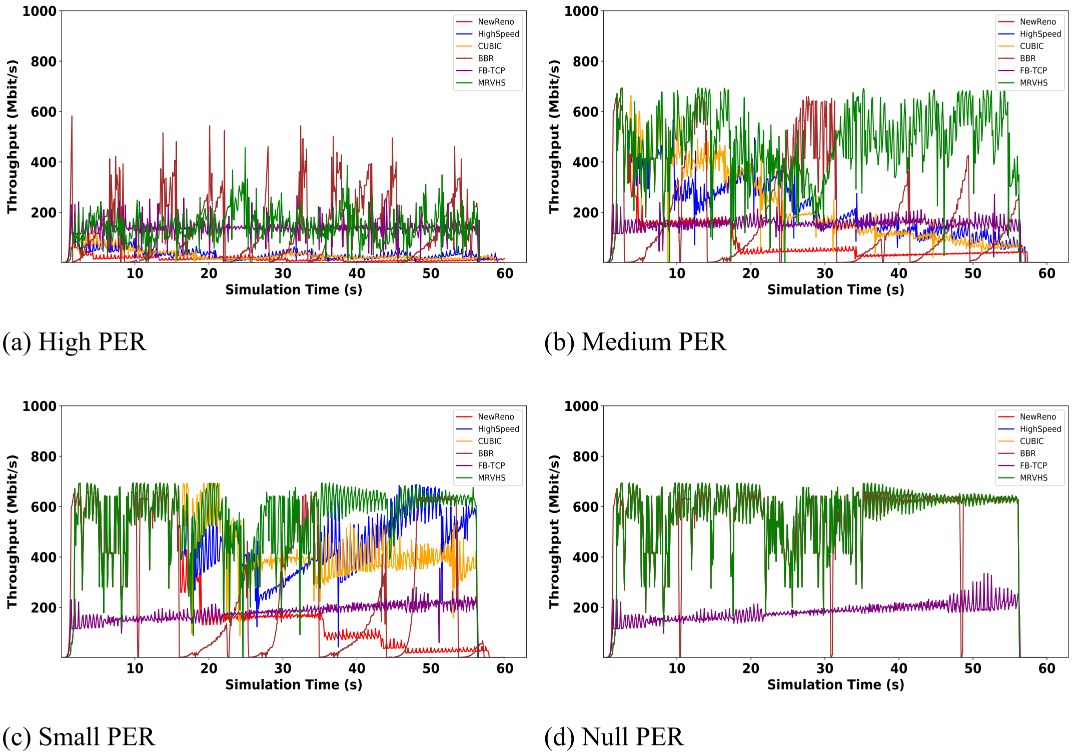


Figure 6.9: Average Throughput of TCP Variants Vs. Different PERs



(a) High PER

(b) Medium PER

(c) Small PER

(d) Null PER

Figure 6.10: Instantaneous Throughput Comparison of TCP Variants Vs. Different PERs

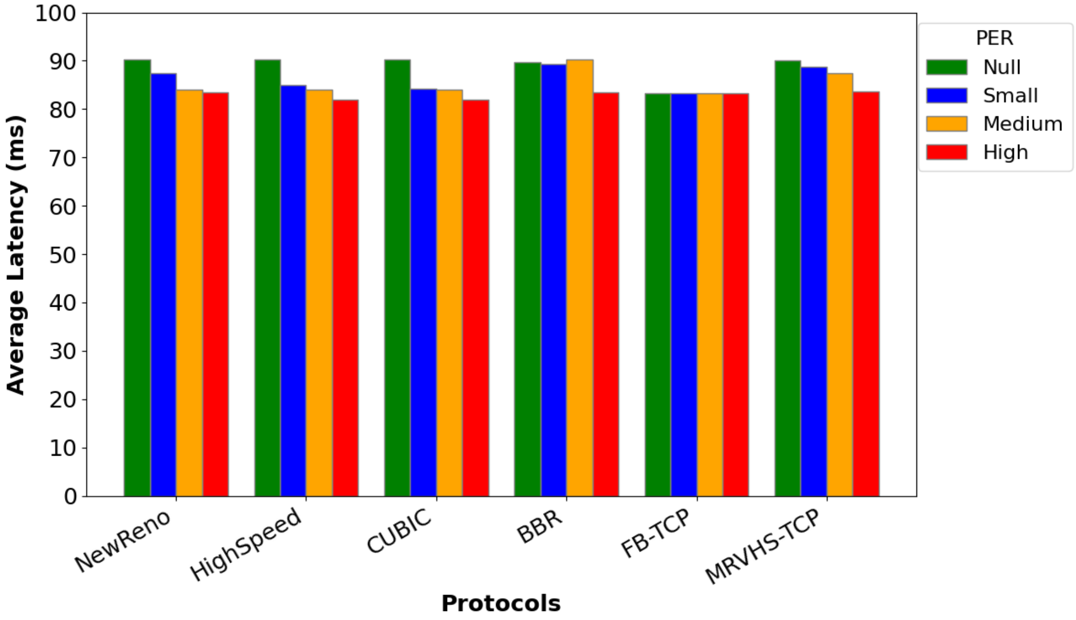
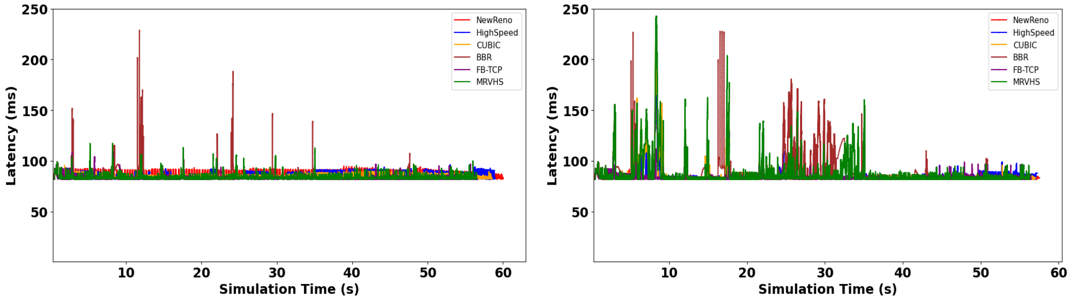
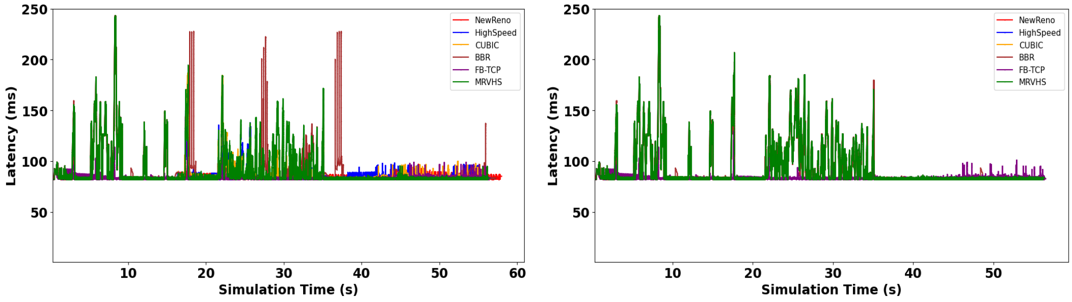


Figure 6.11: Instantaneous Latency Comparison of TCP Variants Vs. Different PERs



(a) High PER

(b) Medium PER



(c) Small PER

(d) Null PER

Figure 6.12: Instantaneous Latency Comparison of TCP Variants Vs. Different PERs

Table 6.1: Percentage improvement of MRVHS-CCA throughput compared to the benchmark protocols

PER	TCP Protocols				
	NewReno	HighSpeed	CUBIC	BBR	FB-TCP
Null	0.89	0.89	0.89	1.14	213.70
Small	146.01	18.42	27.31	87.19	205.06
Medium	379.02	105.29	105.29	144.64	202.53
High	1242.05	347.35	347.35	4.75	5.48

Table 6.2: Percentage improvement of MRVHS-CCA latency compared to the benchmark protocols

PER	TCP Protocols				
	NewReno	HighSpeed	CUBIC	BBR	FB-TCP
Null	-0.27	-0.27	-0.27	0.38	8.06
Small	1.36	4.34	5.21	-0.72	6.41
Medium	4.14	4.14	4.14	-3.11	4.98
High	0.20	2.04	2.04	0.30	0.50

6.5 Summary

In this chapter, a new adaptive TCP called MRVHS-CCA is introduced. This sender-side, loss-delay-based algorithm is designed to function adaptively in remote server scenarios within cellular-to-cloud networks. We conduct ns-3 simulation-based experiments to assess the performance of MRVHS-CCA in comparison with benchmark TCPs, including NewReno, HighSpeed-TCP, CUBIC, BBR, and FB-TCP. The results demonstrate that MRVHS-CCA achieves superior bandwidth utilization while minimizing latency compared to these existing algorithms. Overall, MRVHS-CCA outperforms the benchmark TCP protocols in terms of both average throughput and latency over remote server scenarios of mmWave cellular-to-cloud networks by utilizing the MRVHS CAA mechanism, which mitigates the TCP sensitivity to variations in

RTT and PER over mmWave cellular-to-cloud networks. Furthermore, in particular, MRVHS significantly outperforms benchmark protocols (NewReno, HighSpeed, CUBIC, BBR, FB-TCP) by (1242.05, 347.35, 347.35, 4.75, 5.48)%, respectively, in the worst case of PER scenarios. This performance suggests that MRVHS-CCA could be a promising candidate for implementation in real operating systems to support various networks.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

At first, it may seem obvious that increasing data transmission speeds would improve throughput. However, sending data too quickly without control can cause excessive packet loss, leading to long delays, network congestion, bufferbloat, or even network collapse. Therefore, TCP protocols are used to regulate data transmission according to the current network conditions. In mmWave cellular networks, several challenges hinder TCP performance from fully utilizing the available bandwidth. These challenges include NLoS conditions, signal attenuation, small buffers, high PER, and the server location, whether at the edge or remote. These issues result in non-congestion-related packet loss, buffer overflows, retransmissions, and long delays, which can trigger 3DACKs or RTO events in severe cases. Consequently, these factors prevent TCP from achieving high data rates and low latency effectively over mmWave cellular networks.

In this study, we conduct a comprehensive performance assessment to scrutinize TCP protocol weaknesses stemming from cellular network challenges, such as blockages inducing NLoS states in mmWave cellular networks. We evaluate benchmark protocols, including NewReno, HighSpeed, CUBIC, BBR, and FB-TCP protocols. Additionally, we investigate the impact of varying buffer sizes and packet error rates on TCP performance in mmWave cellular networks. After conducting extensive simulation experiments supported by theoretical analysis, this thesis has resulted in the following conclusions:

A Round-Trip-Time-Varations-based (RTTV-CCA) is proposed to enhance the high data rate over mmWave cellular networks in the presence of an edge server scenario. RTTV-CCA exhibits suitable sensitivity to NLoS conditions by mitigating non-congestion states. Simulation results demonstrate that the RTTV-CCA outperforms benchmark TCP protocols in terms of average throughput and latency. Furthermore, Simulation results demonstrate that RTTV-CCA significantly improves throughput and reduces latency compared to baseline protocols, outperforming the benchmark TCP protocols (NewReno, HighSpeed, CUBIC, BBR) by (1253.96, 207.72, 1253.96, 5.95)% respectively under high PER conditions when buffer size matches the Bandwidth Delay Product (BDP).

Maximum Segment Size-based TCP (MSS-CCA) is proposed to enhance bandwidth utilization of mmWave cellular networks in the presence of an edge server scenario, particularly under conditions of small buffer sizes and high PER. MSS-CCA introduces a novel function that correlates the increase in cwnd with the MSS magnitude. Moreover, it balances the gained increase using this function according to RTT variation, ensuring high throughput and low latency. Simulation results indicate that the MSS-CCA enhances TCP's adaptability to diverse network scenarios, and significantly enhances bandwidth utilization over mmWave cellular networks. Furthermore, it demonstrates higher efficiency and stability compared to benchmark protocols (NewReno, HighSpeed, CUBIC, BBR) by (1486.72, 261.53, 1486.72, 24.26)%, respectively, under high PER conditions, while maintaining low latency.

TCP should be able to handle networks regardless of whether the server is located at the edge of the gNB station or in remote positions. The variation in delay between the UE and the server hinders TCP from achieving a high data rate and maintaining latency at suitable levels. Therefore, an adaptive CCA based on MSS and RTT variations and integrated with HighSpeed-TCP (MRVHS-CCA) is proposed to enhance bandwidth utilization over mmWave cellular networks, particularly in remote server scenarios. Extensive simulation experiments validate that MRVHS-CCA outperforms the benchmark TCP protocols. The results show that MRVHS-CCA performs better than the benchmark protocols in most PER scenarios. In particular, MRVHS significantly outperforms benchmark protocols (NewReno, HighSpeed, CUBIC, BBR, FB-TCP) by (1242.05, 347.35, 347.35, 4.75, 5.48)%, respectively, in the worst case of PER scenarios. Thus, MVRHS-CCA is suitable for end-to-end communication for many applications in mmWave cellular-to-cloud networks.

7.2 Future Work

Building upon the contributions and findings presented in this thesis, several potential directions for future research are proposed to further enhance the robustness, applicability, and impact of the proposed solutions. These directions aim to address the aspects of transport protocol design across diverse network environments:

- **Fairness Evaluation Against State-of-the-Art Protocols:** A key area for future investigation involves conducting a comprehensive fairness analysis between our proposed congestion control approaches and existing state-of-the-art protocols. While the current study primarily focuses on improving latency and throughput, ensuring fairness among competing flows is crucial for widespread deployment in shared networks. Future work will involve implementing controlled experiments to measure fairness using established metrics such as Jain's Fairness Index, under both homogeneous and heterogeneous traffic scenarios. This analysis will help determine whether our approaches equitably share network resources and maintain performance stability under varying conditions.
- **Exploring the Adaptation of the Proposed CCAs for Satellite and UAV Networks:** To test the adaptability and robustness of the proposed mechanisms, future work plans to extend the evaluation beyond mmWave cellular networks. In particular, satellite networks and UAV (Unmanned Aerial Vehicle)-based com-

munication systems represent compelling environments for further study. These networks are characterized by unique challenges such as high propagation delays, intermittent connectivity, and mobility-induced disruptions. Investigating the performance of our protocols in such settings will enable us to identify necessary modifications and ensure that the proposed algorithms are generalizable and efficient across a broader spectrum of network topologies.

- **Deployment in High-Speed Networking Environments:** Another promising research direction is to assess the effectiveness of the RTTV, MSS, and MRVHS algorithms in high-speed networking environments, such as data center networks. These environments have stringent performance requirements, including extremely low latency, high bandwidth demands, and frequent short-lived flows. Evaluating our algorithms in such contexts will not only demonstrate their scalability but also highlight their adaptability to different performance constraints. Furthermore, this exploration may reveal optimization opportunities for applications involving real-time processing and latency-sensitive workloads in large-scale cloud infrastructure.
- **Experimental Validation Through Testbeds and Real-World Trials:** A critical step toward the practical adoption and standardization of our proposed solutions is their implementation and validation in real-world environments. Future work will focus on deploying the protocols on physical or emulated testbeds to evaluate their behavior under real traffic conditions, hardware constraints, and environmental variability. This will serve to validate simulation results, uncover practical implementation challenges, and provide empirical evidence of the protocols' effectiveness and reliability. Such experimental studies are essential for bridging the gap between research and deployment.

REFERENCES

3GPP (2024). 5g applications. Technical report, 3GPP. Available online: `https://www.3gpp.org/ftp/Inbox/Marcoms/3GPP_Poster%20v2.pdf` (accessed 14 August. 2024).

Afanasyev, A., Tilley, N., Reiher, P., and Kleinrock, L. (2010). Host-to-host congestion control for tcp. *IEEE Communications surveys & tutorials*, 12(3):304–342. `http://dx.doi.org/10.1109/SURV.2010.042710.00114`.

Allman, M. (1999). Tcp congestion control (rfc 2581). *http://www. ietf. org/rfc/rfc2581. txt*. Available online: `https://cir.nii.ac.jp/crid/1571417126306444928` (accessed 10 October. 2022).

Allman, M., Floyd, S., and Partridge, C. (2002). Rfc3390: Increasing tcp's initial window. Available online: `https://www.rfc-editor.org/rfc/rfc3782` (accessed 10 October. 2022).

Alrshah, M. A., Al-Maqri, M. A., and Othman, M. (2019). Elastic-tcp: Flexible congestion control algorithm to adapt for high-bdp networks. *IEEE Systems Journal*, 13(2):1336–1346. http://dx.doi.org/10.1109/JSYST.2019.2896195.

Atxutegi, E., Liberal, F., Haile, H. K., Grinnemo, K.-J., Brunstrom, A., and Arvidsson, A. (2018). On the use of tcp bbr in cellular networks. *IEEE Communications Magazine*, 56(3):172–179. `http://dx.doi.org/10.1109/MCOM.2018.1700725`.

Brakmo, L. and Peterson, L. (1995). Tcp vegas: end to end congestion avoidance on a global internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–1480. `http:10.1109/49.464716`.

Cao, Y., Jain, A., Sharma, K., Balasubramanian, A., and Gandhi, A. (2019). When to use and when not to use bbr: An empirical analysis and evaluation study. In *Proceedings of the Internet Measurement Conference*, IMC '19, p. 130–136, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/3355369.3355579.

Cardwell, N., Cheng, Y., Gunn, C. S., Yeganeh, S. H., and Jacobson, V. (2016). Bbr: Congestion-based congestion control: Measuring bottleneck bandwidth and round-trip propagation time. *Queue*, 14(5):20–53. `https://doi.org/10.1145/3012426.3022184`.

Cerf, V., Dalal, Y., and Sunshine, C. (1974). Specification of internet transmission control program. Technical report. Available online: `https://www.rfc-editor.org/rfc/rfc675?trk=public_post_comment-text` (accessed 8 August. 2022).

Cerf, V. and Kahn, R. (1974). A protocol for packet network intercommunication. *IEEE Transactions on Communications*, 22(5):637–648. `doi={10.1109/TCOM.1974.1092259}`.

Chu, J., Dukkipati, N., Cheng, Y., and Mathis, M. (2013). Rfc 6928: Increasing tcp's initial window. Available online: `https://datatracker.ietf.org/doc/html/rfc6928` (accessed 8 August. 2022).

Ericsson (2024). 5g mobile subscriptions. Technical report, Ericsson. Available online: `https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-subscriptions-outlook` (accessed 14 August. 2024).

Floyd, S. (2003). Highspeed tcp for large congestion windows. Available online: `https://www.rfc-editor.org/rfc/rfc3649.html` (accessed 15 September. 2022).

Floyd, S., Henderson, T., and Gurtov, A. (2004). The newreno modification to tcp's fast recovery algorithm. Tech. rep. Available online: `https://www.rfc-editor.org/rfc/rfc3782` (accessed 23 April. 2023).

Gomez, J., Kfoury, E. F., Crichigno, J., and Srivastava, G. (2024). Evaluating tcp bbrv3 performance in wired broadband networks. *Computer Communications*, 222:198–208. `https://doi.org/10.1016/j.comcom.2024.04.037`.

Guo, L. and Lee, J. Y. (2021). Tcp-flash-a fast reacting tcp for modern networks. *IEEE Access*, 9:68861–68879. `http://dx.doi.org/10.1109/ACCESS.2021.3077612`.

Ha, S., Rhee, I., and Xu, L. (2008). Cubic: a new tcp-friendly high-speed tcp variant. *ACM SIGOPS operating systems review*, 42(5):64–74. `https://doi.org/10.1145/1400097.1400105`.

Haile, H., Grinnemo, K.-J., Hurtig, P., and Brunstrom, A. (2022). Rbbr: A receiver-driven bbr in quic for low-latency in cellular networks. *IEEE Access*, 10:18707–18719. https://doi.org/10.1109/ACCESS.2022.3148998.

Henderson, T., Floyd, S., Gurtov, A., and Nishida, Y. (2012). The newreno modification to tcp's fast recovery algorithm. Available online: `http://dx.doi.org/10.17487/RFC3782` (accessed 17 January. 2023).

Islam, S., Zainab Abdulsalam, A., Anil Kumar, B., Kamrul Hasan, M., Kolandaisamy, R., and Safie, N. (2025). Mobile networks toward 5g/6g: Network architecture, opportunities and challenges in smart city. *IEEE Open Journal of the Communications Society*, 6:3082–3093. `https://doi.10.1109/OJCOMS.2024.3419791`.

Jeddou, S., Diez, L., Abdellah, N., Baina, A., and Calvo, R. A. (2023). On the performance of transport protocols over mmwave links: Empirical comparison of tcp and quic. *IEEE Open Journal of the Communications Society*, 4:2596–2608. `https://doi.10.1109/OJCOMS.2024.3419791`.

Ji, X., Han, B., Xu, C., Song, C., and Su, J. (2023). Adaptive qos-aware multipath congestion control for live streaming. *Computer Networks*, 220:109470. https://doi.org/10.1016/j.comnet.2022.109470.

Jin, C., Wei, D., Low, S. H., Bunn, J., Choe, H. D., Doylle, J. C., Newman, H., Ravot, S., Singh, S., Paganini, F., et al. (2005). Fast tcp: From theory to experiments. *IEEE network*, 19(1):4–11. https://doi.org/10.1109/MNET.2005.1383434.

Kanagarathinam, M. R., Singh, S., Sandeep, I., Kim, H., Maheshwari, M. K., Hwang, J., Roy, A., and Saxena, N. (2020). Nexgen d-tcp: Next generation dynamic tcp congestion control algorithm. *IEEE Access*, 8:164482–164496. http://dx.doi.org/10.1109/ACCESS.2020.3022284.

Kanagarathinam, M. R., Singh, S., Sandeep, I., Roy, A., and Saxena, N. (2018). D-tcp: Dynamic tcp congestion control algorithm for next generation mobile networks. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6. IEEE. http://dx.doi.org/10.1109/CCNC.2018.8319185.

Kanaya, T., Tabata, N., and Yamaguchi, S. (2020). A study on performance of cubic tcp and tcp bbr in 5g environment. In *2020 IEEE 3rd 5G World Forum (5GWF)*, pp. 508–513. IEEE. http://dx.doi.org/10.1109/TVT.2018.2825462.

Khorov, E., Krasilov, A., Susloparov, M., and Kong, L. (2023). Boosting tcp & quic performance in mmwave, terahertz, and lightwave wireless networks: A survey. *IEEE Communications Surveys & Tutorials*. https://doi.org/10.1109/COMST.2023.3301820.

Kim, G.-H. and Cho, Y.-Z. (2019). Delay-aware bbr congestion control algorithm for rtt fairness improvement. *IEEE Access*, 8:4099–4109. https://doi.org/10.1109/ACCESS.2019.2962213.

Kim, G.-H. and Cho, Y.-Z. (2022). mms-tcp: Scalable tcp for improving throughput and fairness in 5g mmwave networks. *Sensors*, 22(10):3609. https://doi.org/10.3390/s22103609.

Kleinrock, L. (2018). Internet congestion control using the power metric: Keep the pipe just full, but no fuller. *Ad Hoc Networks*, 80:142–157. https://doi.org/10.1016/j.adhoc.2018.05.015.

Lar, S.-u. and Liao, X. (2013). An initiative for a classified bibliography on tcp/ip congestion control. *Journal of Network and Computer Applications*, 36(1):126–133. https://doi.org/10.1016/j.jnca.2012.04.003.

Lim, H., Lee, J., Lee, J., Sathyanarayana, S. D., Kim, J., Nguyen, A., Kim, K. T., Im, Y., Chiang, M., Grunwald, D., et al. (2023). An empirical study of 5g: Effect of edge on transport protocol and application performance. *IEEE Transactions on Mobile Computing*, 23(4):3172–3186. https://doi.org/10.1109/TMC.2023.3274708.

Mateo, P. J., Fiandrino, C., and Widmer, J. (2019). Analysis of tcp performance in 5g mm-wave mobile networks. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–7. IEEE. http://dx.doi.10.1109/ICC.2019.8761718.

Mendonça, S., Damásio, B., de Freitas, L. C., Oliveira, L., Cichy, M., and Nicita, A. (2022). The rise of 5g technologies and systems: A quantitative analysis of knowledge production. *Telecommunications Policy*, 46(4):102327. https://doi.org/10.1016/j.telpol.2022.102327.

Mezzavilla, M., Zhang, M., Polese, M., Ford, R., Dutta, S., Rangan, S., and Zorzi, M. (2018). End-to-end simulation of 5g mmwave networks. *IEEE Communications Surveys & Tutorials*, 20(3):2237–2263. https://doi.org/10.1109/COMST.2018.2828880.

Na, W., Bae, B., Cho, S., and Kim, N. (2019). Dl-tcp: Deep learning-based transmission control protocol for disaster 5g mmwave networks. *IEEE Access*, 7:145134–145144. http://dx.doi.org/ 10.1109/ACCESS.2019.2945582.

Navarro-Ortiz, J., Romero-Diaz, P., Sendra, S., Ameigeiras, P., Ramos-Munoz, J. J., and Lopez-Soler, J. M. (2020). A survey on 5g usage scenarios and traffic models. *IEEE Communications Surveys & Tutorials*, 22(2):905–929. https://10.1109/COMST.2020.2971781.

Netalkar, P., Chen, J., and Raychaudhuri, D. (2023). mmcptp: A cross-layer pull based transport protocol for 5g mmwave networks. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–9. IEEE. https://doi.org/10.1109/NOMS56928.2023.10154364.

Niu, Y., Li, Y., Jin, D., Su, L., and Vasilakos, A. V. (2015). A survey of millimeter wave communications (mmwave) for 5g: opportunities and challenges. *Wireless networks*, 21:2657–2676. https://doi.org/10.1007/s11276-015-0942-z.

Pan, W., Xu, Y., and Liu, S. (2023). Making tcp bbr pacing adaptive with domain knowledge assisted reinforcement learning. *IEEE Transactions on Network Science and Engineering*, 10(4):2250–2264. https://doi.org/10.1109/TNSE.2023.3244864.

Pieska, M. and Kassler, A. (2017). Tcp performance over 5g mmwave links — tradeoff between capacity and latency. In *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 385–394. IEEE. https://doi.org/10.1109/WiMOB.2017.8115776.

Polese, M., Giordani, M., Mezzavilla, M., Rangan, S., and Zorzi, M. (2017a). Improved handover through dual connectivity in 5g mmwave mobile networks. *IEEE Journal on Selected Areas in Communications*, 35(9):2069–2084. http://dx.doi.org/10.1109/JSAC.2017.2720338.

Polese, M., Jana, R., and Zorzi, M. (2017b). Tcp in 5g mmwave networks: Link level retransmissions and mp-tcp. In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 343–348. IEEE. http://dx.doi.org/10.1109/INFCOMW.2017.8116400.

Poorzare, R. and Augé, A. C. (2020). Challenges on the way of implementing tcp over 5g networks. *IEEE access*, 8:176393–176415. https://doi.org/10.1109/ACCESS.2020.3026540.

Poorzare, R. and Augé, A. C. (2021a). Fb-tcp: A 5g mmwave friendly tcp for urban deployments. *IEEE access*, 9:82812–82832.

Poorzare, R. and Augé, A. C. (2021b). How sufficient is tcp when deployed in 5g mmwave networks over the urban deployment? *IEEE access*, 9:36342–36355. http://dx.doi.org/10.1109/ACCESS.2021.3063623.

Poorzare, R. and Calveras, A. (2023). Deep learning tcp for mitigating nlos impairments in 5g mmwave. *INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL*, 18(4).

Poorzare, R. and Waldhorst, O. P. (2023). Toward the implementation of mptcp over mmwave 5g and beyond: Analysis, challenges, and solutions. *IEEE Access*, 11:19534–19566. https://doi.org/10.1109/ACCESS.2023.3248953.

Postel, J. (1981a). Internet protocol. Technical report. Available online: `https://www.rfc-editor.org/info/rfc791` (accessed 14 August. 2021).

Postel, J. (1981b). Transmission control protocol. Std, document RFC 793. Available online: `www.rfc-editor.org/info/rfc793` (accessed 14 August. 2021).

Salahdine, F., Han, T., and Zhang, N. (2023). 5g, 6g, and beyond: Recent advances and future challenges. *Annals of Telecommunications*, 78(9):525–549. https://doi.org/10.1007/s12243-022-00938-3.

Scharnitzky, D., Krämer, Z., and Molnár, S. (2021). Comparison of tcp siad and tcp bbr congestion control in simulated 5g networks. *Acta Universitatis Sapientiae, Electrical and Mechanical Engineering*, 13(1):114–124.

Siddiqui, F. and Chau, Q. (2021). Analysis of transport layer congestion control algorithms over 5g millimeter wave networks. In *2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 1–6. https://doi.org//10.1109/ICUFN49451.2021.9528538.

Sudhamani, C., Roslee, M., Tiang, J. J., and Rehman, A. U. (2023). A survey on 5g coverage improvement techniques: Issues and future challenges. *Sensors*, 23(4):2356. https://doi.org/10.3390/s23042356.

Sylla, T., Mendiboure, L., Maaloul, S., Aniss, H., Chalouf, M. A., and Delbruel, S. (2022). Multi-connectivity for 5g networks and beyond: A survey. *Sensors*, 22(19):7591. https://doi.org/10.3390/s22197591.

Wang, Z. and Crowcroft, J. (1992). Eliminating periodic packet losses in the 4.3-tahoe bsd tcp congestion control algorithm. *ACM SIGCOMM Computer Communication Review*, 22(2):9–16. https://doi.org/10.1145/141800.141801.

Xiao, K., Mao, S., and Tugnait, J. K. (2019). Tcp-drinc: Smart congestion control based on deep reinforcement learning. *IEEE Access*, 7:11892–11904. http://dx.doi.org/10.1109/ACCESS.2019.2892046.

Xie, Y., Jiang, X., Gong, G., Jiang, Z., Jin, G., and Chen, H. (2023). Yinker: A flexible bbr to achieve the high-throughput and low-latency data transmission over wi-fi and 5g networks. *Computer Networks*, 222:109530. https://doi.org/10.1016/j.comnet.2022.109530.

Zhang, M., Mezzavilla, M., Ford, R., Rangan, S., Panwar, S., Mellios, E., Kong, D., Nix, A., and Zorzi, M. (2016). Transport layer performance in 5g mmwave cellular. In *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 730–735. IEEE. https://doi.org/10.1109/INFCOMW.2016.7562173.

Zhang, M., Mezzavilla, M., Zhu, J., Rangan, S., and Panwar, S. (2017). Tcp dynamics over mmwave links. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–6. https://doi.org/10.1109/SPAWC.2017.8227746.

Zhang, M., Polese, M., Mezzavilla, M., Zhu, J., Rangan, S., Panwar, S., and Zorzi, M. (2019). Will tcp work in mmwave 5g cellular networks? *IEEE Communications Magazine*, 57(1):65–71. http://dx.doi.org/10.1109/MCOM.2018.1701370.

BIODATA OF STUDENT

Alramli Omar Imhemed Abdulmajid received a bachelor's degree in Eletric. and Electronic Engineering from Benghazi (Garyounis) University, Libya, and a master's degree in Computer Engineering. Currently, he is a Ph.D candidate at faculty of Computer Science and Information Technology, UPM, Malaysia. His research interests are in the field of TCP protocols, congestion control algorithms, and mmWave cellular networks. Currently serving as a lecturer at Misurata University, Libya.

LIST OF PUBLICATIONS

International Refereed Journals

Omar Alramli, Zurina Mohd Hanapi, Mohamed Othman, Idawaty Ahmad, Normalia Samian. (2024). RTTV-TCP: Adaptive Congestion Control Algorithm Based on RTT Variations for mmWave Networks. *Ad Hoc Networks, 164C, 103611*. **(Published 2024, JIF = 4.4, Q1, ISI, JCR)**

Omar Alramli, Zurina Mohd Hanapi, Mohamed Othman, Normalia Samian, Idawaty Ahmad. (2025). MSS-TCP: A Congestion Control Algorithm for Boosting TCP Performance in mmWave Cellular Networks. *ICT Express*. **(Published 2025, JIF = 4.1, Q1, ISI, JCR)**

Omar Alramli, Zurina Mohd Hanapi, Mohamed Othman, Idawaty Ahmad, Normalia Samian. (2025). An Adaptive congestion control algorithm for improving Transmission Control Protocol performance over Cellular-to-Cloud Networks. *PeerJ Computer Science*. **(Accepted 2025, JIF = 3.5, Q1, ISI, JCR)**

Omar Alramli, Zurina Mohd Hanapi, Mohamed Othman, Idawaty Ahmad, Normalia Samian, Mohamed Alrashah. (2024). An Empirical Performance Analysis of TCP Variants in 5G mmWave Cellular Networks: Buffer Size, Error Rate, and Server Placement Considerations. *Wireless Personal Communications*. **(In the second round of revision, submitted on 03 Jun 2024, JIF = 1.9, Q3, ISI, JCR)**

UNIVERSITI PUTRA MALAYSIA

STATUS CONFIRMATION FOR THESIS / PROJECT REPORT AND COPYRIGHT

ACADEMIC SESSION : SECOND SEMESTER 2024/2025

TITLE OF THESIS / PROJECT REPORT : CONGESTION CONTROL ALGORITHMS FOR END-TO-END COMMUNICATION OVER MMWAVE CELLULAR NETWORKS

NAME OF STUDENT : ALRAMLI OMAR IMHEMED ABDULMAJID

I acknowledge that the copyright and other intellectual property in the thesis/project report belonged to Universiti Putra Malaysia and I agree to allow this thesis/project report to be placed at the library under the following terms:

1. This thesis/project report is the property of Universiti Putra Malaysia.
2. The library of Universiti Putra Malaysia has the right to make copies for educational purposes only.
3. The library of Universiti Putra Malaysia is allowed to make copies of this thesis for academic exchange.

I declare that this thesis is classified as :

*Please tick (√)

☐	**CONFIDENTIAL**	(Contain confidential information under Official Secret Act 1972).
☐	**RESTRICTED**	(Contains restricted information as specified by the organization/institution where research was done).
☑	**OPEN ACCESS**	I agree that my thesis/project report to be published as hard copy or online open access.

This thesis is submitted for :

☐	**PATENT**	Embargo from ________ (date) until ________ (date)

Approved by:

Alramli

(Signature of Student)
New IC No/ Passport No.:**AA186858**
Date : **3 Oct 2025**

(Signature of Chairman of Supervisory Committee)
Name: **Zurina Mohd Hanapi**
Date : 7 Oct 2025

[Note : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization/institution with period and reasons for confidentially or restricted.]