Network Traffic Congestion Control

A major project report submitted in partial fulfillment of the requirement for the award of a degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Siddhanth Verma (201528) & Abhinandan Thakur (201537)

Under the guidance & supervision of

Dr. Shubham Goel

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

CERTIFICATE

This is to certify that the work which is being presented in the project report titled"**Network Traffic Congestion Control**" in partial fulfillment of the requirements for the award of the degree of B.Tech in Information Technology and submitted to the Departmentof Computer Science & Engineering And Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by "Siddhanth Verma (201528) and Abhinandan Thakur (201537)" during the period from August 2023 to December 2023 under the supervision of **Dr. Shubham Goel** (Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology).

Siddhanth Verma (201528)
Abhinandan Thakur (201537)

The above statement is correct to the best of my knowledge.

(Supervisor)
Supervisor Name: Dr. Shubham Goel
Designation: Assistant Professor (SG)
Department: Computer Science and Engineering

DECLARATION

I hereby declare that the work presented in this report entitled **'Network Traffic Congestion Control'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my work carried out over a period from August 2023 to December 2023 under the supervision of **Dr. Shubham Goel** (Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Student Name: Siddhanth Verma
Roll No.: 201528

Student Name: Abhinandan Thakur
Roll No.: 201537

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Supervisor Name: Dr. Shubham Goel
Designation: Assistant Professor (SG)
Department: CSE/IT
Dated: 13/05/24

ACKNOWLEDGEMENT

We want to express our genuine gratitude to our respected project mentors, **Dr. Shubham Goel** for their invaluable guidance, unwavering support, and continuous supervision throughout this project. Their expertise and encouragement were indispensable, and without them, we would have faced considerable challenges.

The mentorship provided by our supervisors has been crucial in steering us in the right direction, offering not just essential guidance but also insightful information crucial to the success of the project. We appreciate their consistent motivation, which played a pivotal role in guiding us toward the successful completion of this endeavor.

Additionally, we extend heartfelt thanks to our parents and the JUIT community for their generous cooperation and encouragement. Their support has been a driving force behind our accomplishments.

Recognition is also due to the invaluable contributions of our colleagues, whose skills and collaboration were integral to the development of this project. Their combined efforts elevated the overall quality and success of our work.

In conclusion, the completion of this project has been a collective effort, and we sincerely appreciate everyone who played a role, whether directly or indirectly, in bringing it to fruition.

Siddhanth Verma (201528)
Abhinandan Thakur (201537)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

S. No.	Abbreviation	Expansion
1	**TCP**	Transmission Control Protocol
2	**JWT**	JSON Web Token
3	**AWS**	Amazon Web Services
4	**CDN**	Content Delivery Network
5	**MD5**	Message Digest Algorithm 5
6	**SHA**	Secure Hash Algorithm

ABSTRACT

The rapid proliferation of internet-enabled devices and the escalating demand for data-intensive applications have engendered severe congestion issues within modern network infrastructures. Network congestion impedes the smooth flow of data, resulting in latency, packet loss, and diminished quality of service. This research introduces the "Network Traffic Congestion Control System," an innovative project that endeavors to mitigate congestion challenges within networks by orchestrating efficient traffic management.

Employing machine learning algorithms and predictive modeling, this system dynamically monitors network traffic patterns, predicts potential congestion points, and orchestrates data traffic to alleviate bottlenecks. Additionally, the project seeks to enhance the current congestion control algorithms by evaluating and comparing various existing congestion control mechanisms. Through this comparative analysis, the research aims to reach a consensus on the most efficient and adaptable algorithms, allowing for the integration of superior congestion control strategies.

Integrating principles of cloud computing and adhering to robust information security standards, the system ensures not only optimized traffic flow but also secure and resilient data transmission.

The scope of this research extends to exploring the predictive capabilities of machine learning models, investigating cloud-native traffic optimization, and evaluating the system's effectiveness across diverse network environments. Through meticulous analysis and experimentation, this research seeks to contribute solutions that enhance the efficiency, reliability, and security of data transmission in the modern digital landscape, fostering a seamless and responsive network experience for users across various domains and applications.

CHAPTER 1:
INTRODUCTION

1.1 INTRODUCTION

In today's era of digital interconnectedness, the Internet serves as the lifeline of modern communication, commerce, and countless essential services. Much like a bustling highway, the internet channels an immense flow of data, resembling cars traversing its lanes. Internet traffic continues to grow rapidly. Cisco's Annual Internet Report (2020-2025) projected that global internet traffic will nearly triple from 2017 to 2022. However, akin to traffic congestion on a busy road, internet networks experience bottlenecks, causing delays and interruptions in data transmission. Congestion affects various realms of the internet, from business, administration, healthcare, online retail, and online video streaming to name a few. It also impacts how consumers perceive a website.

The world today is heavily reliant on the internet for communication and exchange of data and the intensity of that paradigm can be highlighted by the statement that ‘Data is the new Oil’. We need ultra-reliable and latency-avoidant networks to keep up with the demands of the amount of data that is shared and streamed across the internet. A study by Sandvine found that internet outages or severe congestion can result in losses of up to $1 million per hour for large online businesses. To mitigate this issue and optimize the efficiency of data exchange, the "Network Traffic Congestion Control System" project emerges as a crucial initiative.

The exponential growth in internet usage, driven by the proliferation of online activities, necessitates effective traffic management. Instances of network congestion leading to reduced data speeds, increased latency, and interrupted connectivity have become prevalent. These issues impact various facets of daily life, from video streaming and onlinegaming to critical professional video conferences and telecommuting. Thus, the imperative arises to create a sophisticated system capable of intelligently managing and optimizing thedata flow within networks to alleviate congestion-related challenges.

Moreover, with the advent of emerging technologies like the Internet of Things (IoT), cloud computing, and real-time applications, the strain on networks is poised to escalate further. This amplifies the urgency for a robust congestion control mechanism that ensures the smooth and efficient functioning of these technologies within the evolving digital landscape.

“Network traffic congestion control system” project aims to increase the efficiency and reliability of information transmitting in the networks. This system will utilize modern machine learning techniques to dynamically monitor traffic flows in networks and anticipate traffic jams. It will maximize network throughput while minimizing delay by adapting in real time data routing and resource allocation.

Furthermore, the project incorporates principles of cloud computing, leveraging its scalability and flexibility to adapt to varying network loads. By integrating stringent information security measures, ensures the safe and encrypted transmission of data, aligning with contemporary privacy standards. The project's scope extends beyond immediate needs, considering future internet traffic patterns and technological advancements to build a sustainable and adaptable congestion control framework.

In summary, the "Network Traffic Congestion Control System" project represents a strategic effort to enhance the efficiency and resilience of Internet traffic management. By addressing current challenges and envisioning future demands, it aspires to fortify the digital infrastructure that underpins our interconnected world.

1.2 PROBLEM STATEMENT

The escalating usage of the internet has led to persistent network congestion, disrupting data flow and impeding services critical to daily life.

The inefficiencies stem from inadequate traffic management, resulting in reduced speeds, increased latency, and compromised user experiences. Moreover, emerging technologies like IoT exacerbate this issue, necessitating an intelligent and adaptive "Network Traffic Congestion Control System."

This system seeks to predict, manage, and optimize network traffic in real-time, ensuring

seamless data transmission, reducing congestion, and enhancing overall network performance to meet the ever-growing demands of our digitally reliant society.

1.3 OBJECTIVE

Optimize Network Resource Allocation: Develop an intelligent system to dynamically allocate network resources, reducing bottlenecks and enhancing data throughput for seamless communication.

Minimize Latency and Response Time: Implement predictive algorithms to proactively manage congestion, reducing delays and ensuring efficient real-time applications like video conferencing and IoT devices.

Enhance Adaptive Security Measures: Leverage AI and ML to continually assess network data, enabling the system to dramatically adapt congestion control strategies while prioritizing cloud computer integrity and robust information security protocols.

1.4 SIGNIFICANCE AND MOTIVATION OF THE PROJECT WORK

The significance and motivation behind this project lie in the profound impact of network congestion on the seamless operation of modern communication, commerce, and essential services reliant on the Internet. As the global digital ecosystem continues to expand exponentially, the surge in internet-enabled devices and data-intensive applications has led to increasingly congested network infrastructures. This congestion manifests as bottlenecks, causing delays, packet loss, and compromised user experiences. Addressing this challenge becomes imperative to ensure uninterrupted connectivity and optimal performance across diverse digital domains.

The motivation stems from the pressing need to optimize congestion control mechanisms within networks. Effective congestion management is pivotal in sustaining the integrity and reliability of data transmission. A robust congestion control system not only mitigates disruptions but also enhances the overall quality of service, safeguarding against service degradation and ensuring equitable bandwidth distribution.

Moreover, the motivation arises from the potential repercussions of network congestion in various sectors. In business, network disruptions during peak periods can translate into substantial revenue losses. In healthcare, delays in telemedicine or remote patient monitoring due to congestion can impact patient care. Educational institutions and governmental services also rely heavily on uninterrupted network connectivity, making congestion control pivotal across diverse sectors.

Furthermore, the project's significance lies in its potential to advance the field of congestion control algorithms. By evaluating and comparing existing algorithms like TCP, TCP Reno, and TCP CUBIC, the project aims to identify the most efficient and adaptable strategies. This endeavor contributes to the evolution of robust congestion control systems capable of managing traffic optimally in contemporary and future network architectures.

Ultimately, the project's significance rests in its aim to create a more resilient and responsive digital infrastructure, fostering seamless connectivity and enhancing user experiences across a spectrum of digital services and applications.

1.5 ORGANIZATION OF PROJECT REPORT

The six chaptered comprehensive paper presents the important information in regard to the Human activity recognition project.

Chapter 1: Introduction

The first chapter is a take-off point of the problem under discussion, sets out objectives and explains motivation at the start of the project. It serves as a sturdy pillar for what is coming next.

Chapter 2: Literature Survey

This chapter examines what is already known by studying publications, including the book and technical papers in the last five years, and reports it. This aims at getting a feel of the current situation and what can our project fill in the gaps.

Chapter 3: System Development

This is the core part of the project that explains how it began with the requirement analysis, systems design and implementation. We talk about problems experienced in the process of developing and using strategic solutions.

Chapter 4: Testing

This section sheds light on the meticulous testing process, explaining the strategy and tools used. We present test cases and outcomes, offering a clear picture of the system's reliability.

Chapter 5: Results and Evaluation

Focusing on tangible outcomes, this chapter interprets findings and compares them with existing solutions if applicable. It provides an in-depth evaluation of our results.

Chapter 6: Conclusions and Future Scope

Concluding our exploration, this chapter summarizes key findings, acknowledgeslimitations, and outlines potential future directions for research and development.

CHAPTER 2:
LITERATURE SURVEY

Gaurav Choudhary, Vishal Sharma, and Jiyoon Kim

[1] This research dives into the different types of attacks 5G networks can face and their prevention measures. It is a well-known fact that 5G networks can face blockage and latency issues due to grid issues and obstructions.

Furthermore, the rapid involution of the mobile generation with incipient data networking capabilities and utilization has exponentially increased the data traffic volumes. Suchtraffic drains various key issues in 5G mobile backhaul networks. Security of mobile backhaul is of utmost importance; however, there are a limited number of articles, which have explored such a requirement. This paper discusses the potential design issues and key challenges of the secure 5G mobile backhaul architecture. The comparisons of the existing state-of-the-art solutions for secure mobile backhaul, together with their major contributions have been explored. Furthermore, the paper discussed various key issues related to Quality of Service (QoS), routing and scheduling, resource management, capacity enhancement, latency, security management, and handovers using mechanisms like Software Defined Networking and millimeter Wave technologies. Moreover, the trials of research challenges and future directions are additionally presented.

Josip Lorincz, Zvonimir Klarin, and Julije Ožegović

[2]This research paper delves into how TCP is implemented in 5G networks, whilst upholding reliability and security. The primary protocol in use today in data networks to guarantee dependable communications is the protocol for transmission control (TCP). The congestion control (CC) technique in use has a major impact on TCP performance. Over the previous three decades, TCP CC algorithms have changed and Numerous CC algorithm variants have been created to support different network surroundings. The deployment of the TCP CC mechanism has a new problem with the fifth-generation (5G) mobile network,

as networks will function in contexts with massive user device counts and heavy traffic volumes.

The application of TCP CC algorithms in 5G mmWave communications will be further degraded by large channel quality changes and susceptibility to blockages due to high penetration losses and atmospheric absorptions, in contrast to pre-5G networks that function in the sub-6 GHz ranges. These difficulties will be more prevalent in settings like Internet of Things (IoT) applications and sensor networks. To ease these difficulties, the most well-liked single-flow and multi-flow TCP CC techniques are summarized in this work, utilized in networks before 5G.

Further provided is the related work on the earlier analyses of the TCP CC algorithm performance in 5G networks. Analysis of a potential TCP CC algorithm implementation is done in detail considering the peculiarities of 5G networks, namely the use of high frequencies and data buffering, frequencies in the mmWave spectrum are frequently switched between horizontal and vertical directions, the 5G core network is implemented, and the exploitation of edge computing and the always-on signals that are transmitted nonstop.

Maab Fathi Hamzah and Omar Ali Athab

[3]This research paper provides a comprehensive, in-depth narrative on the need for 5G, the technologies needed to implement 5G, and its security measures. It delves into concepts like MIMO, D2D, 3GPP, 5GPP, etc. which prove to be imperative for better understanding the network requirements and the mechanisms behind making that network work.

Research on the congestion issue in 4G and 5G networks, particularly those utilizing artificial intelligence (AI), has expanded in recent years. Even though 4G with LTE is thought to be a mature technology, 5G networks are the result of ongoing infrastructural improvements. Internet of Things (IoT) applications and smart cities—which have a lot of data exchanged, a lot of connected devices per area, and high data rates—have their own set of issues and challenges as a result of the extensive services offered in industries. One of the main issues these cities face is congestion. Within this context, one of the primary methods for addressing network congestion is the employment of artificial intelligence (AI) models.

It seems promising to integrate communication networks with AI to address the congestion issue because AI technologies can handle massive volumes of data and extract pertinent aspects from it. Further study is needed in this area. An overview of the application of AI technology to 4G and 5G network congestion is given in this study. We looked at earlier research on the issue of network congestion, including studies on congestion avoidance, prediction, control, and development of TCP for congestion management. Lastly, we talk about how AI technologies will be used in 4G and 5G networks in the future to address congestion concerns and pinpoint areas of research that require more investigation.

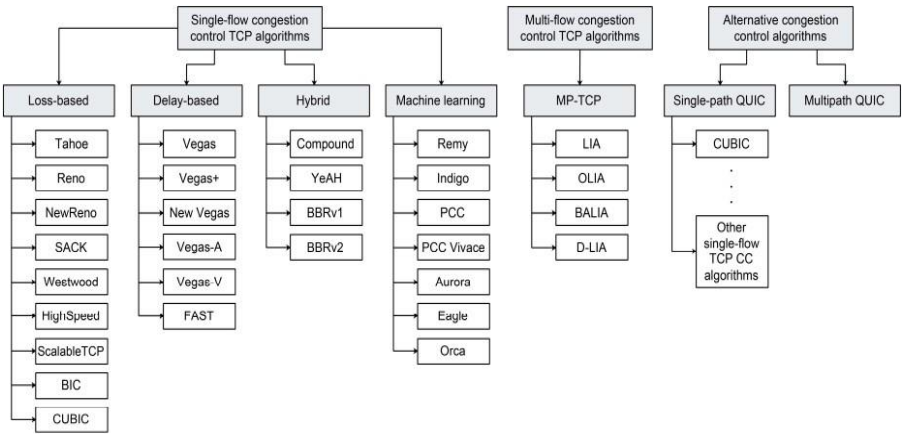


Fig. 1 Three type of TCP Algorithm

[4]This research paper offers an all-around overview of the TCP congestion control algorithms, going that extra mile to explore how much protocol behaves differently under different circumstances. In today's world, where our networks consist of a multitude of clients and machine-to-machine interactions, it is imperative that we have apt knowledge about congestion control and subsequently choose the right congestion control algorithm. The research first lists the various effects of congestion on a network and then comprehensively lists the antidote, i.e. the prevention policies which can be employed against these effects. Packet Loss, Jitter, Full Buffer Memory, Severe Performance Degradation, Loss of Customers, and Artificial Congestion all make it under this list of effects of congestion. Some prevention policies against these effects have been illustrated below in Fig. 2-

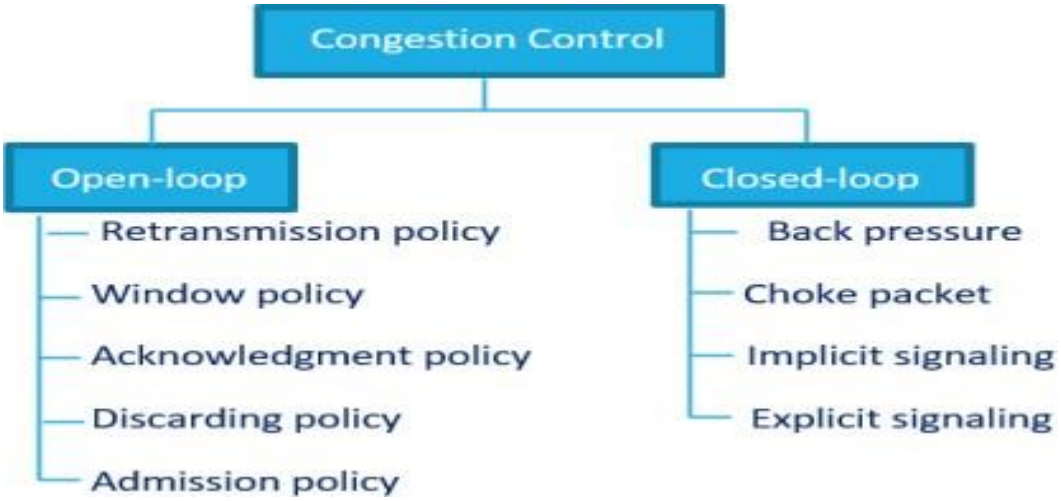


Fig. 2 Classification of Congestion Techniques

The study then further deepens into the previous studies done on congestion by various authors, listing significant contributions they made in their respective research and finally, a side-by-side comparison of these methods to control congestion.

Pooja Chaudhary and Sachin Kumar

[5]This study offers the reader an in-depth understanding of TCP, giving us reasons as to why the different TCP variants needed to come into existence in the first place.

TCP is a widely used transport protocol in computer networks, known for its reliable and connection-oriented nature. It operates by sending data in the form of byte streams and establishing connections, making it the preferred choice for applications that require secure information delivery. TCP also ensures data integrity and employs measures such astimeouts and retransmissions to ensure accuracy. Various studies on computer network processes have revealed that traffic exhibits time-scale invariance, a characteristic that is often affected by the file allocation patterns on servers and user behavior. There have been cases where data streams that initially lacked independent similarities have later shown autonomous behavior. The utilization of low factor does not prevent quick buffer overwhelm, highlighting the need for action to manage incoming traffic. If left unaddressed, queues at the maximum weighted boundary will continue to grow, ultimately leading to larger buffers at identical

nodes. This study examines various TCP variants and their effectiveness in managing congestion in networks, evaluating performance metrics such as end-to-end wait, throughput, queue size, and packet delay rate using Network Simulator-2 (NS-2). The results demonstrate that in highly congested networks, Vegas performs the most effectively,while in low congestion networks, Reno yields the most favorable outcomes.

S.No	Time	Tahoe	Reno	New-Reno	Vegas
1	10	10	10	10	10
2	20	500	500	510	550
3	30	1000	1000	1100	1100
4	40	1300	1200	1100	1000
5	50	3300	3200	3100	3000
6	60	5000	7000	9000	11000

Table 1. Throughput Comparison of TCP Variants

Thomas Lukaseder, Leonard Bradatsch, Benjamin Erb, et al.

[6]With the increasing availability of 10G Ethernet networks, current transport layer protocols face a new challenge. As 10G connections continue to gain traction beyond backbone networks, the selection of proper TCP congestion control algorithms becomes crucial for networked applications operating in environments like data centers. To shed light on this issue, this study presents a detailed overview of relevant TCP congestion control algorithms specifically designed for high-speed environments leveraging 10G. Using a physical network testbed, this research has thoroughly analyzed and evaluated six TCP variants, with a particular focus on the impact of propagation delay and significant drop rates. Based on our findings, it is evident that BIC is the most suitable algorithm whenthere is no legacy variant present, while CUBIC is recommended otherwise.

A comparison of the variants has been illustrated below-

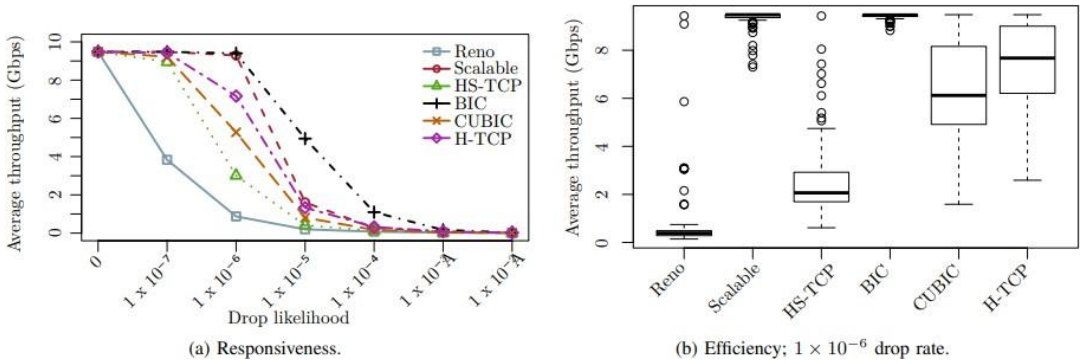


Fig. 3 Responsiveness of efficiency at 12.2 ms RTT

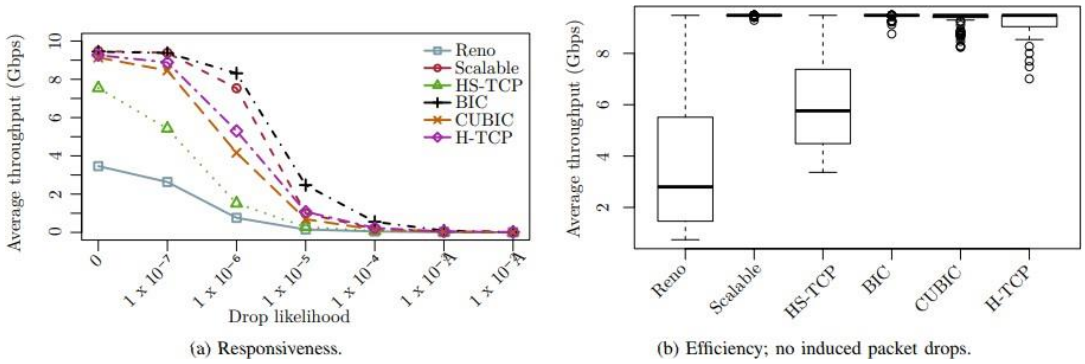


Fig. 4 Responsiveness of efficiency using different variants

U. Hengartner, J. Bolliger, and Th. Gross

[7]This research paper describes the ins and outs of the famous TCP variant, TCP Reno. Over the last few years, the use of innovative techniques in TCP Vegas has sparked much debate. Numerous studies have shown that TCP Vegas outperforms TCP Reno in terms of performance. However, the question of which specific techniques are responsible for these impressive gains remains unanswered. In this paper, the aim is to comprehensively evaluate the performance of TCP Vegas. By breaking down the protocol and examining the impact of each unique mechanism, the research demonstrates that the reported performance improvements are primarily due to TCP Vegas's novel approaches to slow start and congestion recovery. Interestingly, findings also reveal that TCP Vegas's innovative congestion avoidance mechanism plays a minimal role in achieving these throughput gains.

Moreover, the evaluation reveals that the congestion avoidance mechanism exhibits a fairness problem even when all competing connections are operating on the same RTT.

Ijaz Ahmad, Tanesh Kumar, Madhusanka Liyanage, et al.

[8]This study showcases an analysis of threats and solutions that networks may face, particularly 5G networks.

As we move towards the coming generation of wireless connectivity, 5G promises to revise our digital geography. By extending broadband access to indeed the most remote corners of the world, supporting flawless mobility for druggies, and connecting a vast array of biases through the Internet of Effects (IoT), 5G will marshal a new period of dependable and affordable connectivity. This metamorphosis is made possible by technological advancements in pall computing, Software Defined Networking (SDN), and Network Function Virtualization(NFV), which are evolving to support the demands of 5G. Still, as with any new technology, there are enterprises regarding security and sequestration. In this composition, the authors take a look at the security challenges within these critical 5G technologies and their impact on stoner sequestration. Also, the study offers implicit results to address these challenges and explore unborn directions for erecting secure 5G systems.

Security Threat	Target Point/Network Element	Effected Technology				Privacy
		SDN	NFV	Channels	Cloud	
DoS attack	Centralized control elements	✓	✓		✓	
Hijacking attacks	SDN controller, hypervisor	✓	✓			
Signaling storms	5G core network elements			✓	✓	
Resource (slice) theft	Hypervisor, shared cloud resources		✓		✓	
Configuration attacks	SDN (virtual) switches, routers	✓	✓			
Saturation attacks	SDN controller and switches	✓				
Penetration attacks	Virtual resources, clouds		✓		✓	
User identity theft	User information data bases				✓	✓
TCP level attacks	SDN controller-switch communication	✓		✓		
Man-in-the-middle attack	SDN controller-communication	✓		✓		✓
Reset and IP spoofing	Control channels			✓		
Scanning attacks	Open air interfaces			✓		✓
Security keys exposure	Unencrypted channels			✓		
Semantic information attacks	Subscriber location			✓		✓
Timing attacks	Subscriber location				✓	✓
Boundary attacks	Subscriber location					✓
IMSI catching attacks	Subscriber identity			✓		✓

Table 2: Security Challenges in 5G Technology

2.1 OVERVIEW OF RELEVANT LITERATURE

In today's era of digital interconnectedness, the internet serves as the lifeline of modern communication, commerce, and countless essential services. Much like a bustling highway, the internet channels an immense flow of data, resembling cars traversing its lanes. Internet traffic continues to grow rapidly. Cisco's Annual Internet Report (2020-2025) projected that global internet traffic will nearly triple from 2017 to 2022. However, akin to traffic congestion on a busy road, internet networks experience bottlenecks, causing delays and interruptions in data transmission [1].

The exponential growth in internet usage, driven by the proliferation of online activities, necessitates effective traffic management [3]. Instances of network congestion leading to reduced data speeds, increased latency, and interrupted connectivity have become prevalent. These issues impact various facets of daily life, from video streaming and onlinegaming to critical professional video conferences and telecommuting. Thus, the imperative arises to create a sophisticated system capable of intelligently managing and optimizing thedata flow within networks to alleviate congestion-related challenges [6]

According to research carried out by Université Hassan-Ier in Morocco[7], these TCP algorithms can be divided into two sects, loss based (Tahoe, Reno, New Reno and Fack) which are recommended for networks which do not support a long transmission delay, and others that are delay-based such as Vegas. Essentially, each variant has different characteristics and its own behavior depending on what the state of the network is.

In 2016, Google published the bottleneck bandwidth and round-trip time (BBR) congestion control algorithm[8]. Unlike established loss- or delay-based algorithms like CUBIC or Vegas, BBR claims to operate without creating packet loss or filling buffers. Despite its obvious advantages, it yielded two primary problems; bandwidth can be shared unfairly in correlation to new flows joining existing flows and second, time until bandwidth equilibrium is regained.

It can also be noted that there are also other parameters other than the ones mentioned above which can be used to judge the performance of a congestion control algorithm, such as Queue management. A comprehensive study on the algorithms used for queue management in large networks by Mustafa Maad Hamdi and co, where the important

features of subsections of the TCP were presented[10]. Then we showed the two main classes of the existing router queue management system. The results in this study showed the performance of ARED was better than that of RED, because there is a drop in End toEnd Delay with better readings in. Throughput and Packet delivery ratio. While RRMDPto provide significant enhancements in average queue size and delay scheduling compared to RED and ARED.

It is also worthwhile mentioning that there are scenarios where reliable packet delivery is the utmost priority. When operating in such an environment with such tight prerequisites, not even a single packet can be dropped. Research was conducted on such networks by Stanford University in 2018[2], consisting of a theoretical study of the stability and fairnessproperties of network level congestion control when pause mechanisms operate at the link level to prevent packet drops. Their focus was the Backward Congestion Notification (BCN) algorithm which is being considered by the IEEE 802.1 standards body for deployment in switched Ethernet networks. In such networks a back-pressure mechanism "pauses" the link or links feeding a congested buffer, thus preventing further packets from arriving at the buffer. The links are later unpaused when the buffer becomes uncongested. Itconcluded that the overall system switches between two separate sets of equations depending on the sign of the feedback variable, Fb. Their work was dependent on thefuture works that analyze the collaboration of TCP and BCN, as TCP performs end-to-end congestion control and relies on packet drops to get congestion notices.

Whilst most of these studies are rooted in TCP and variants of TCP, it is also important to mention the shortcomings of TCP as a congestion control algorithm to have a peek at both sides of the coin. The standard TCP's performance is very poor inHigh Speed Networks (HSN) and hence the core gigabytes links are usually underutilized. This problem has roots in conservative nature of TCP, especially in its Additive Increase Multiplicative Decrease (AIMD) phase. In other words, since TCP can't figure out precisely the congestion status of the network, it follows a conservative strategy to keep the network from overwhelming.

This led to the birth of TCP PHCC, which uses simultaneous latency and packet-based strategies to increase data transmission performance in high-speed networks. It uses the concept of a probabilistic function and the Bayesian theorem to estimate the appropriate size of a congestion window to make the most of the available bandwidth.

Because it is difficult to analyze the performance of network traffic, due to the lack of access to the values of some indicators, such as buffer size, router status and behavior of other data streams, etc. or due to lack of knowledge of the factors affecting such failures and events that may occur in the network, some of which are very complex and unpredictable, used in performance analysis. This feature of PHCC distinguishes it from other previously proposed algorithms and makes better decisions in acute situations.

2.2 KEY GAPS IN THE LITERATURE

The research papers were an excellent tool to introduce us to TCP, its variants, and other congestion control algorithms. However, they offer little insight into the statistics of these variants, like average latency, dropout rate, and conversion rate. The knowledge contained in these papers is more on the theoretical side rather than the practical side as little has been said about their utilization in the real world.

CHAPTER 3:

SYSTEM DEVELOPMENT

3.1 REQUIREMENTS AND ANALYSIS

3. 1.1 FUNCTIONAL REQUIREMENTS

1. **User Authentication:**

- The system should provide a secure user authentication process.
- Users must log in using their existing account credentials (username and password).

2. **File Upload:**

- Users should be able to upload a file from their local system or cloud storage.
- Supported file formats should be specified.
- The system must handle different file sizes within defined limits.

3. **Recipient Email Input:**

- The application should prompt the user to enter the email address of the recipient.
- Ensure validation checks for the correctness of the email format.

4. **Email Sending:**

- Once the file and recipient email are provided, the system should send an email with the uploaded file attached to the specified recipient.
- Email delivery confirmation or status should be displayed to the user.
- Include error handling for failed email deliveries.

5. **Compression:**

- Implemented a compression mechanism using FFmpeg library to ensure the security and confidentiality of the transmitted files.

3. 1. 2 NON-FUNCTIONAL REQUIREMENTS

1. **Security:**

- Ensure the transmission of files and user data occurs securely using encryption protocols.
- Implement secure storage practices for user credentials and uploaded files.

2. **Usability:**

- The application should have a user-friendly interface, guiding users through each step clearly.
- Response times for each action should be minimal to provide a smooth user experience.

3. **Reliability:**

- The system should be reliable, handling a reasonable number of concurrent users without performance degradation.
- Implement backup and recovery mechanisms to prevent data loss.

4. **Scalability:**

- Design the application to handle potential growth in the number of users and file uploads without compromising performance.

5. **Compatibility:**

- Ensure the application is compatible with different devices and browsers commonly used by the target audience.
- Support various operating systems for file uploads and email services.

6. **Performance:**

- Define acceptable response times for file uploads and email transmissions.
- Monitor and optimize system performance to meet defined benchmarks.

3.1.3 TECHNICAL REQUIREMENTS

1. **Cloud Computing Infrastructure**

- Utilized a reliable cloud platform, AWS, for hosting the application.

2. **Web Application Framework**

- Choose the Node.js web framework for developing the application.
- Ensured the framework supports user authentication and file handling functionalities.

3. **User Authentication Mechanism**

- Implemented a secure authentication mechanism, preferably using protocols like JWT tokens.
- Utilized encryption for password storage and transmission.

4. **File Upload and Storage**

- Implemented a file upload mechanism using appropriate libraries or tools supported by the chosen framework.
- Stored uploaded files securely in the cloud storage service, ensuring access control and data encryption.

5. **Email Sending Service Integration**

- Implemented the API provided by the email service for sending emails with attachments.

6. **Scalability and Load Balancing**

- Designed the application architecture to be scalable, utilizing load balancing techniques to handle increased user traffic and file uploads.
- Utilized features provided by the cloud platform for auto-scaling and resource optimization.

7. **Performance Optimization**

- Optimized application performance by using caching mechanisms, efficient algorithms for file handling, and minimizing network latency.

8. **Backup and Recovery**

- Set up regular data backups to prevent data loss and implement recovery procedures in case of system failures or data corruption.

9. **Data Compression:**

- Incorporated compression measures for data transmission and storage to enhance the security and reduce the file size.

3.1.3 PRIORITY CHART/ MOSCOW CHART

```
High Priority:
- User Authentication Mechanism
- Secure File Upload and Storage
- Email Sending Integration
- Data Encryption (at rest and in transit)
- Error Handling and Logging

Medium Priority:
- Cloud Computing Infrastructure Setup
- Web Application Framework Selection and Setup
- Cross-platform Compatibility Testing
- Compliance and Security Measures
- Performance Optimization

Low Priority:
- Scalability and Load Balancing
- Backup and Recovery Implementation
```

Fig.5 Priority Chart

A. High Priority:

- User Authentication Mechanism
- Secure File Upload and Storage
- Email Sending Integration
- Data Encryption (at rest and in transit)
- Error Handling and Logging

B. Medium Priority:

- Cloud Computing Infrastructure Setup
- Web Application Framework Selection and Setup
- Cross-platform Compatibility Testing
- Compliance and Security Measures
- Performance Optimization

C. Low Priority:

- Scalability and Load Balancing
- Backup and Recovery Implementation

3.2 PROJECT DESIGN AND ARCHITECTURE

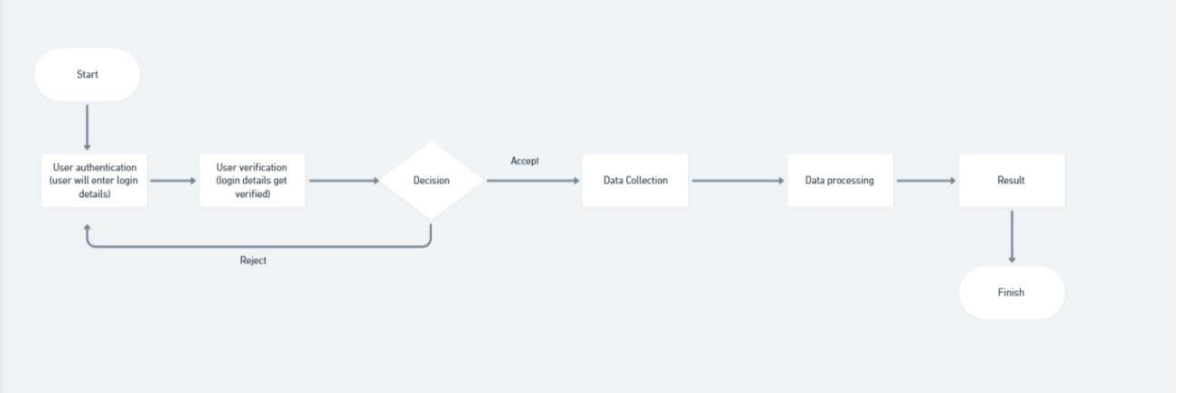


Fig.6 Data Flow Diagram

3.3 DATA PREPARATION

The data being used in this project is majorly the one being used to simulate the various TCP variants in the Cisco Packet Tracer.

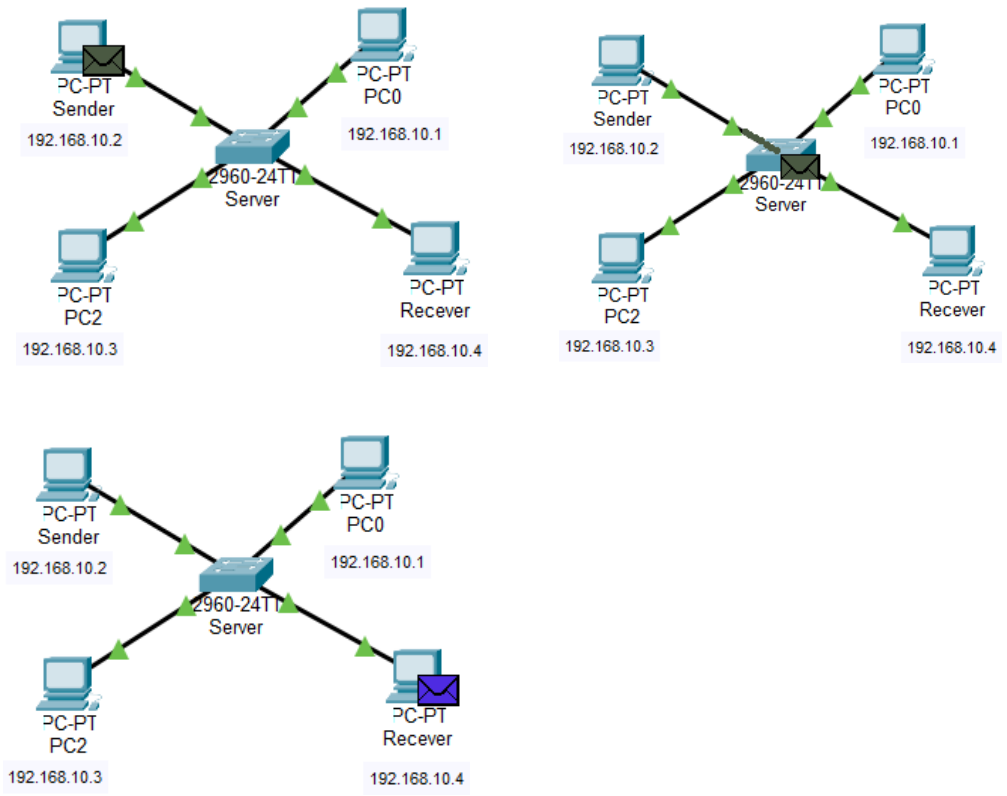
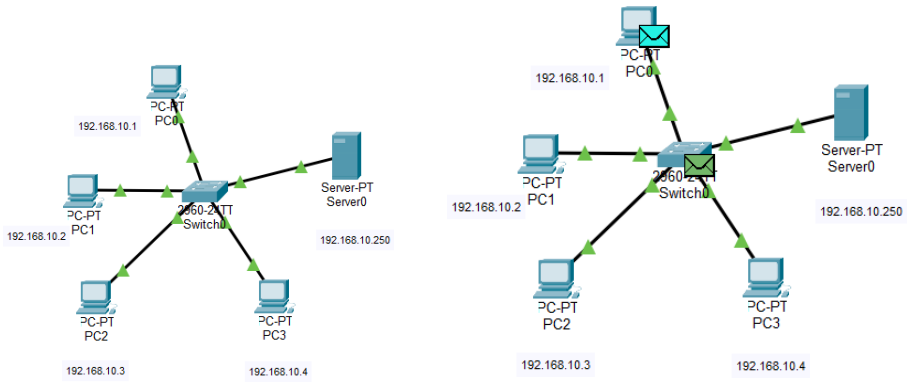


Fig.7 Simple message transfer without congestion



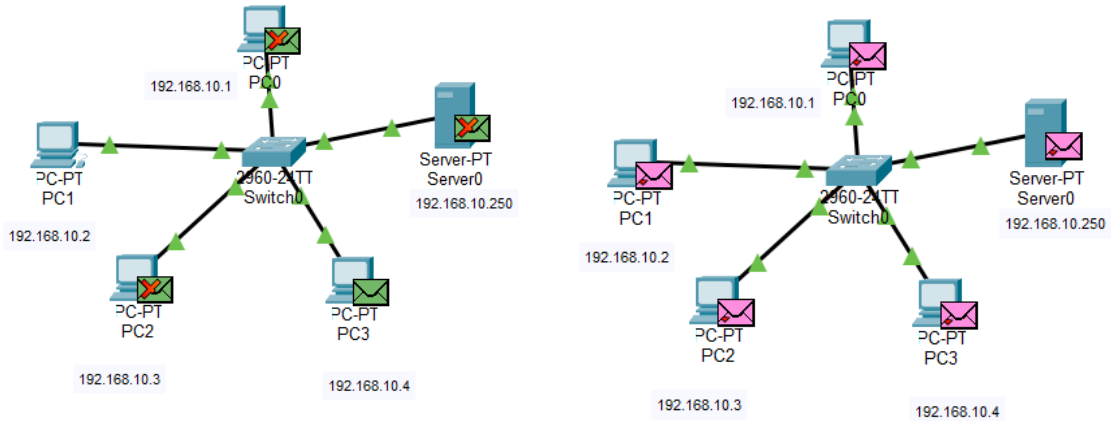


Fig. 8 TCP Congestion simulation

3.4 IMPLEMENTATION

3.4.1 CODE SNIPPETS

```
const express = require("express");
const router = express.Router();
const authController = require("../controllers/authController");

router.post("/signin", signinUser);
router.post("/signup", signupUser);

function signinUser(req, res, next) {
  authController.authenticate(req.body).then((data) => {
    res
      .status(data.status)
      .cookie("token", data.token, {
        secure: true,
        httpOnly: true,
        sameSite: "None",
      })
      .send(data.response);
  });
}

function signupUser(req, res, next) {
  authController.register(req.body).then((data) => {
    res.status(data.status).send(data.response);
  });
}

module.exports = router;
```

Fig. 9 Authentication of the user

```
async function consumeMessages_compression_result_queue() {
  try {
    // Create a connection to RabbitMQ server
    const queueName = "compression_result_queue";
    const connection = await amqp.connect(rabbitmqURL);
    const channel = await connection.createChannel();
    channel.prefetch(1);
    await channel.assertQueue(queueName, { durable: false });
    console.log(`Waiting for messages from queue "${queueName}"...`);

    // Consume messages from the queue
    channel.consume(queueName, async (message) => {
      if (message !== null) {
        const jsonMessage = JSON.parse(message.content.toString());
        console.log(`Received message from queue "${queueName}":`, jsonMessage);

        const uploadResult = await fileShareModel.findOneAndUpdate(
          { shareid: jsonMessage.shareid },
          {
            "file.compressed": true,
            "file.location": jsonMessage.CompressedFileUrl,
            "file.key": jsonMessage.CompressedFilekey,
          },
          { new: true }
        );
        console.log(jsonMessage.shareid);
        emailHandler.sendFileSharingEmail({
          senderName: uploadResult.file.shared_by,
          fileUrl: process.env.BaseUrl + "/download/" + uploadResult.shareid,
          emailReceiver: "<" + uploadResult.email + ">",
        });
        console.log("Email sent to: ", uploadResult.email);
        channel.ack(message);
      }
    });
```

Fig. 10 Compression of File

```
require("dotenv").config();
const mongoose = require("mongoose");

const connectionOptions = {
  useNewUrlParser: true,
  useUnifiedTopology: true,
};

mongoose.set("strictQuery", false);
mongoose.connect(process.env.MONGODB_URL, connectionOptions);
mongoose.Promise = global.Promise;

const connection = mongoose.connection;

module.exports = connection;
```

Fig. 11 Connection to the database

```
const File = require("../models/fileModel");
const FileShare = require("../models/fileShareModel");
const emailHandler = require("../utils/email-handler.js");
const amqp = require("amqplib");
require("dotenv").config();
const rabbitmqURL = process.env.AMQP_URL;

const { v4: uuidv4 } = require("uuid");

async function getAllFiles(req) {
  const files = await File.find({ userid: req.body.userid });
  return {
    status: 200,
    response: {
      files,
    },
  };
}

async function shareFile(req) {
  const files = await File.findOne({ fileid: req.body.fileid });
  const fileshare = new FileShare({
    shareid: uuidv4(),
    file: {
      id: req.body.fileid,
      name: files.filename,
      location: files.location,
      key: files.key,
      shared_by: req.body.name,
    },
    email: req.body.email,
    password: req.body.password,
  });
  await fileshare.save();
```

Fig. 12 To handle file

```
async function saveFile(req, res, next) {
  if (!req.files) {
    return {
      status: 400,
      response: { error: "No files were uploaded." },
    };
  } else {
    const newfile = new File({
      fileid: uuidv4(),
      key: req.files[0].key,
      userid: req.body.userid,
      filename: req.files[0].originalname,
      filesize: req.files[0].size,
      mimetype: req.files[0].mimetype,
      location: req.files[0].location,
    });
    await newfile.save();
    console.log("File Saved", newfile);
    return {
      status: 200,
      response: {
        message: "Successfully uploaded " + req.files.length + " files!",
        files: req.files,
      },
    };
  }
}
```

```
async function sendToCompressionQueue(jsonMessage) {
  try {
    const queueName = "compression_request_queue";
    const connection = await amqp.connect(rabbitmqURL);
    const channel = await connection.createChannel();
    await channel.assertQueue(queueName, { durable: false });

    channel.sendToQueue(queueName, Buffer.from(JSON.stringify(jsonMessage)));
    console.log(
      `Compression request sent to queue "${queueName}":`,
      jsonMessage
    );

    await channel.close();
    await connection.close();
  } catch (error) {
    console.error("Error:", error);
  }
}
```

3.4.2 TOOLS AND TECHNIQUES USED

1. CISCO PACKET TRACER

Cisco Packet Tracer is a Cisco-developed educational networking simulation software that constructs simulacra of network topology and configurations. It is a simulated tool used for designing and configuring network devices that mimic Cisco routers and switches.

Usage in Project:

- **Network Simulation and Configuration:** Packet Tracer simulates and configures network topologies of interest within the " Network Traffic Congestion Control System" project of Cisco.
- **Visual Representation:** Graphical user interface is used with packet tracer for designing network infrastructure as well as traffic flow management solutions for better understanding and visualization.

Key Features for Project:

- **Virtual Network Design:** It provides artificial network topologies which help users implement the different congestion control mechanisms as well as traffic optimization strategies on them.
- **Cisco Device Emulation:** To simulate the Cisco equipment, providing an opportunity to test particular features related to traffic engineering andtraffic management.
- **Visualization and Learning:** Provides a perspective on the dynamics of traffic flows and their handling situations in a network simulation context.

Relevance to Project:

The simulation capability and Cisco device modeling in Cisco Packet Tracer.
The objective is to emulate and test congestion control.

The mechanism offers an experimental network for education.t://Input: He has a clear understanding of his abilities as an instructor.
network designs and imagining different traffic control options.
the project's congestion control strategies.

2. NODE.JS

It is an open source JavaScript environment that runs on servers, and is intended for making large scale networking applications and programs. It runs on different operating systems including windows, linux, unix and macos. The v8 javascript engine runs javascript codes not in a web browser.

Usage in Project:

- **Unified Development:** Node.js is an essential component of "Network Traffic Congestion Control System" project under which the same language known as JavaScript is used on the server-end as well as client-side to develop the entire application stack.
- **Server-Side Scripting:** It uses an event-based model known as event-driven programming which makes it appropriate in processing of real time data since it enhances efficiency because it enables the creation of lightweight web servers that serve faster than normal web servers.
- **Asynchronous I/O:** This asynchronous I/O enhances its throughput and scalability properties for applications that are web based in real time or those that have large scale data input.
- **Scalable Network Applications:** Developers can create efficient traffic management and congestion control strategies by taking advantage of Node.js that enables them to build scalable and threadless servers through event driven programming models.

Key Features for Project:

- **JavaScript Everywhere Paradigm:** Node.js allows the use of JavaScript on either side allowing simplified development while unifying the

programming language across all aspects of networking control as well as the creation of internet applications.

- **Event-Driven Architecture:** The inherent attributes of its event-driven and asynchronous capabilities make it useful for managing on the fly traffic variations and incorporating intelligent load balancing approaches into real time web based systems.
- **Support and Open Source Nature:** The OpenJS foundation supports Node.js through cooperative initiatives. As an open source solution it is licensed under BSD, encouraging its uptake by the web dev community.

Relevance to Project:

NNode. js provides an event-driven, scaleable, and Jacob script centered solution that aligns with the project goal i which is efficient control of network traffic flows and implementation of congestion management within a unified programming language environment.

3. FFMPEG

It is a powerful and free framework comprising tools to handle multimedia like audio and video data among others. It has libraries and command line utilities for transcoding, encoding, multiplexing, demultiplexing, streaming, filtering and decoding among other things.

Usage in Project:

- **Multimedia Compression:** FFmpeg is used in the "Network Traffic Congestion Control System" project as a tool for encrypting multimedia messages and protecting sensitive data sent via the network.
- **Encryption Capabilities:** FFmpeg also has capabilities of encrypting multimedia content information, thus making it more secure and confidential when transmitted on the network.
- **Codec Support:** The system supports a number of audio and video codecs including those applied in transmitting information within the project's data.

Key Features for Project:

- **Multimedia Encryption:** The project's security measures include robust encryption of multimedia content that employs the FFmpeg's encryption functionalities.
- **Customizable Encryption Parameters:** Provides flexibility in terms of configuring encryption mechanisms that could be used to secure multimedia data depending on the exact project's needs.
- **Cross-Platform Support:** Therefore, FFmpeg works with numerous OS including Windows which allows for smooth functioning in the project's network settings.

Relevance to Project:

The "Network Traffic Congestion Control System" depends on the ability of FFmpeg to encrypt multimedia for the purpose of maintaining the integrity and privacy of information traversing various systems of communication.

4. AMAZON S3

AWS (Amazon Web Services) provides a cloud based object storage service known as 'Amazon S3' which ensures scalable durable storage infrastructure. Users can use it to store and retrieve data, most importantly web information via web interfaces or application programming interfaces (API).W

Usage in Launching Live Websites:

- **Static Website Hosting:** It is possible for users to save on their storage cost whereby they configure their Amazon S3 buckets to host static web pages.
- **Website Deployment:** Such files stored in the bucket could be made publicly accessible enabling users to interact with the site via its unique address assigned by AWS S3 for launching a live website.

- **Scalability and Reliability:** Websites hosted in S3 buckets are highly scalable and durable to ensure that they can manage different volumes of traffic without failures, even if there may be outages.

Key Features for Hosting Live Websites:

- **Static Content Delivery:** S3 offers efficient hosting of static website contents whereby users can retrieve HTML, CSS, and other website assets with low latency.
- **Cost-Effective Hosting:** Amazon S3 is a storage and bandwidth based price model that enables users to only pay for what they use when hosting their sites.
- **Global Accessibility:** By configuring S3 buckets with CDN services such as Amazon CloudFront, website content distributed around the world is made available with low latency.

Relevance in Hosting Live Websites:

Hosting live websites using Amazon S3 is economical, scalable, and has reliable services. Through S3's static website hosting feature, users can easily develop and run websites, resulting in a consistent online experience available worldwide.

3.4.3 ALGORITHM

Our approach to network traffic congestion control integrates various algorithms for efficient data transmission and reception. One crucial aspect of our system involves video compression to minimize the impact of multimedia data on network congestion. In this regard, we leverage the H.264 algorithm, a widely adopted video compression standard, to encode and decode video streams.

1. **H.264 Algorithm**

The H.264 algorithm, often referred to as Advanced Video Coding (AVC), stands out as an extensively employed video compression standard known for its

adeptness in preserving video quality while minimizing bit rates. Embedded within the FFMPEG library, this algorithm forms an integral part, contributing to a versatile toolkit for multimedia processing encompassing tasks such as video compression and decompression.

H.264 achieves high compression efficiency through various advanced techniques, such as motion estimation, spatial prediction, and entropy coding. By utilizing predictive coding and inter-frame compression, H.264 significantly reduces the volume of data required to represent video frames without compromising quality. This is particularly crucial for our network traffic congestion control system, where the efficient transmission of multimedia data plays a pivotal role.

2. Integration with FFMPEG

Our implementation incorporates the FFMPEG library, a powerful multimedia processing toolkit that includes support for H.264 encoding and decoding. FFMPEG facilitates seamless integration of the H.264 algorithm into our system, enabling efficient compression of video streams before transmission and decompression upon reception.

The integration of H.264 via FFMPEG ensures that our network traffic congestion control system optimally handles multimedia data, preventing undue strain on network resources. This choice aligns with our commitment to achieving a balance between maintaining video quality and mitigating the risk of network congestion.

In summary, the incorporation of the H.264 algorithm within the FFMPEG library stands as a key component of our network traffic congestion control strategy, emphasizing efficient video compression to enhance overall system performance.

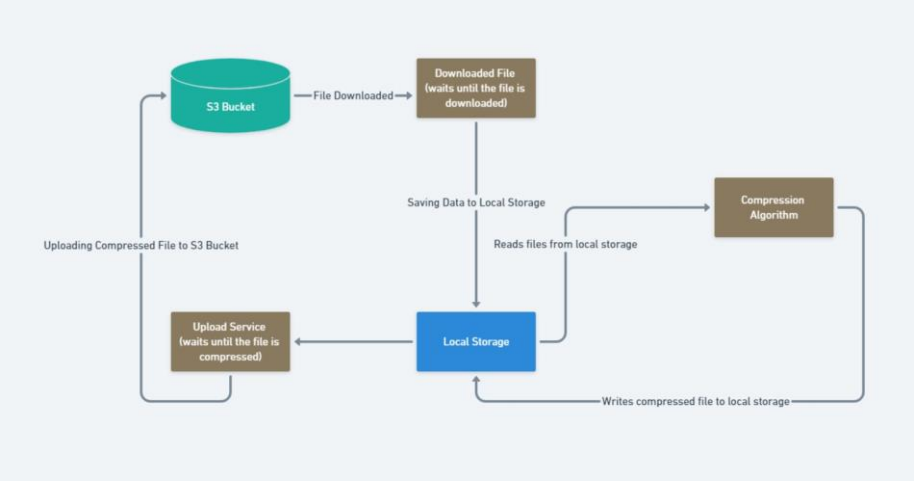


Fig. 13 Data flow for the compression algorithm

3.5 KEY CHALLENGES

In the network traffic congestion project, challenges in cloud file storage encompassensuring data security, managing transfer speed during congestion, maintaining reliability and uptime, effective cost management, and adhering to compliance requirements. Simultaneously, file compression challenges involve balancing between lossless and lossy compression, optimizing compression algorithms for speed and ratio, ensuring compatibility across platforms, managing resource usage, and implementing robust integrity checks and error handling mechanisms. Addressing these challenges is imperativefor the project's success, requiring meticulous testing and ongoing optimization to ensure secure and efficient file transfer and compression within the cloud-based system.

CHAPTER 4:

TESTING

4.1 TESTING STRATEGY

1. SENDER'S LOGIN AND AUTHENTICATION TESTING

- **Objective:** Validate the login process for the sender and ensure secure authentication.
- **Steps:**
  - ➔ Test valid and invalid sender login credentials.
  - ➔ Verify password encryption and storage.
  - ➔ Ensure session management (login/logout) works correctly.
  - ➔ Check for authentication vulnerabilities or bypass possibilities.

2. SENDER FILE UPLOAD TESTING

- **Objective:** Ensure successful and secure file upload by the sender.
- **Steps:**
  - ➔ Test various file sizes for upload capability.
  - ➔ Check supported file formats and limitations.
  - ➔ Validate upload progress and completion.
  - ➔ Test interruption-resume scenarios during the upload process.

3. FILE COMPRESSION TESTING

- **Objective:** Validate the compression of the uploaded file before sending it to the client.
- **Steps:**
  - ➔ Check the file compression process and algorithms used.
  - ➔ Verify the integrity of the compressed file.

➔ Test decompression to ensure the file is retrievable in its original form.

4. EMAIL GENERATION AND SENDING TO CLIENT

- **Objective:** Validate the generation and delivery of an email to the client for the compressed file.
- Steps:
  - ➔ Verify the accuracy of email content and format.
  - ➔ Test email-sending functionality from the server.
  - ➔ Ensure the email contains the necessary information (file details, links, instructions).
  - ➔ Validate the reliability and timeliness of email delivery.

5. CLIENT AUTHENTICATION AND AUTHORIZATION TESTING

- **Objective:** Confirm the client's ability to authenticate and access the compressed file.
- **Steps:**
  - ➔ Validate client access with correct credentials.
  - ➔ Test authentication failure scenarios.
  - ➔ Verify session handling after client authentication.
  - ➔ Ensure the client's permissions and access rights are correctly applied.

6. CLIENT REQUEST TO DOWNLOAD COMPRESSED FILE FROM CLOUD SERVER

- **Objective:** Ensure clients can request and receive the authorized compressed file download.
- **Steps:**
  - ➔ Test the file download request process from the client's side.
  - ➔ Validate authorization checks before initiating file downloads.
  - ➔ Check for error handling and proper notification for unauthorized download attempts.

7. FILE DECOMPRESSION TESTING ON CLIENT'S COMPUTER

- **Objective:** Confirm successful and secure file decompression on the client's system.
- **Steps:**
  - ➔ Test the integrity of the downloaded compressed file.
  - ➔ Verify the decompression process to retrieve the original file.
  - ➔ Ensure file format and content match the original after decompression.
  - ➔ Check for interruption-resume scenarios during the decompression process.

OVERALL CONSIDERATIONS

- **End-to-end Testing:** Perform end-to-end tests to simulate the entire user flow, including compression by the sender and decompression by the client.
- **Negative Testing:** Include negative test scenarios to validate error handling, security measures, and boundary conditions.
- **Performance Testing:** Assess the system's performance under various load conditions for compression and decompression processes.
- **Security Testing:** Conduct security testing to identify vulnerabilities and ensure data protection throughout the compression and decompression stages.

This strategy ensures comprehensive testing of the compression and decompression steps along with the entire flow of the system from sender authentication to file download by the client.

4.2 TEST CASES AND OUTCOMES

OBJECTIVE : Ensure the file remains unchanged after compression and decompression processes.

STEPS :

1. **PREPARATION**

➔ Upload a known test file to the system.

➔ Record the file's size, content, and checksum (MD5, SHA, etc.) as a baseline.

2. **COMPRESSION**

➔ Trigger compression of the uploaded file using the system's compression method.

➔ Validate the compressed file's integrity and size.

3. **DECOMPRESSION**

➔ Initiate decompression of the compressed file on the system.

➔ Verify the decompressed file's integrity, size, and content.

4. **COMPARISON**

➔ Compare the decompressed file with the original uploaded file.

➔ Ensure the size, content, and checksum match the baseline recorded earlier.

5. **VALIDATION**

➔ Confirm that the file retrieved by the client matches the original uploaded file.

➔ Ensure the client receives the file in its exact original state.

➔ Verify that the compression and decompression processes did not alter the file content.

CHAPTER 5:

RESULTS AND EVALUATION

5.1 RESULTS

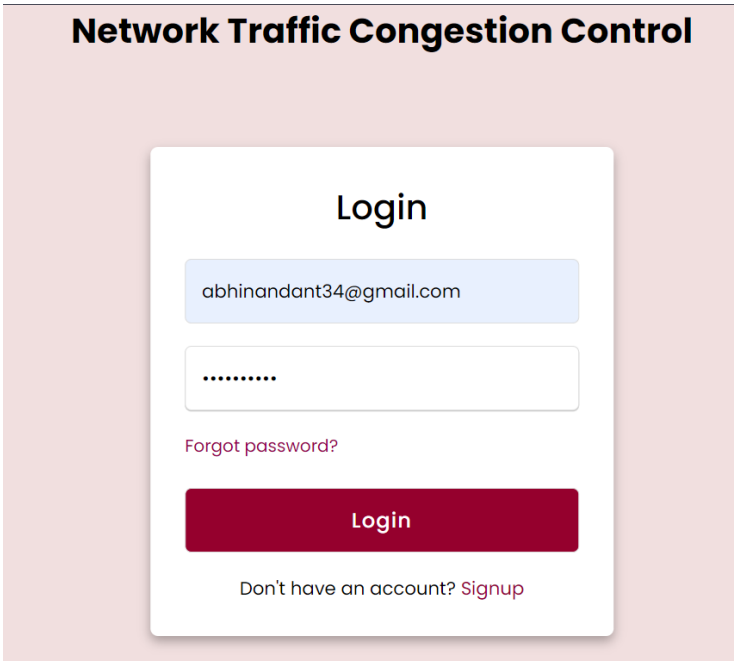


Fig. 14 Sender's Login and Authentication Testing Overview

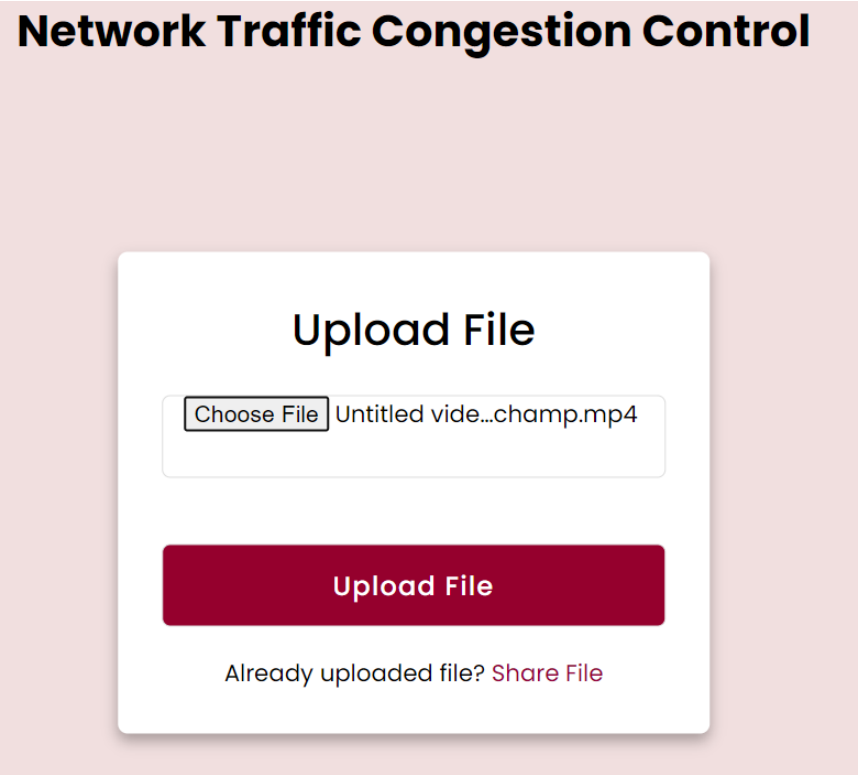


Fig. 15 Sender File Upload Testing Overview

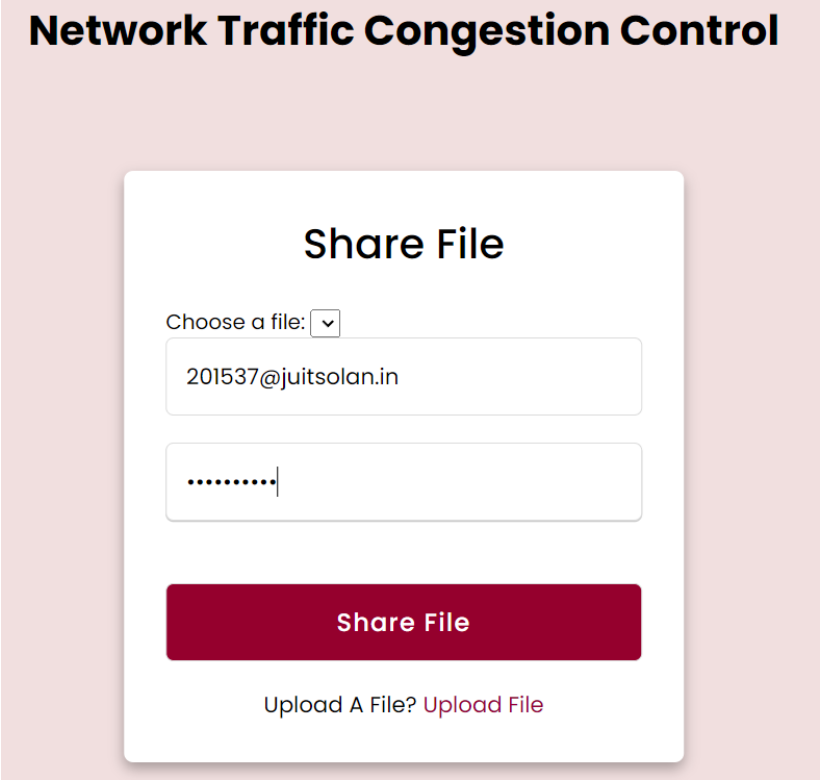


Fig. 16 Sender File Share Testing Overview

Fig. 17 Video Quality Comparison: A visual representation highlighting the video's quality before and after undergoing the compression process.

Event List

Vis.	Time(sec)	Last Device
	0.000	--
	0.000	--
	0.001	PC1
	0.002	Switch0
	0.002	Switch0
	0.002	Switch0
	0.002	Switch0
	0.003	PC3
	0.004	Switch0
	0.004	--
	0.005	PC1
	0.006	Switch0
	0.007	PC3
	0.008	Switch0
	0.998	--
	0.999	Switch0
	0.999	Switch0
	0.999	Switch0
	0.999	Switch0
	0.999	Switch0
	2.998	--

Fig. 18 Congestion Flow Simulation

CHAPTER 6:
CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In conclusion, the 'Network Traffic Congestion Control System' project makes a big move in tackling the increasing issues linked to network congestion in today's digital world. The main goal of the project was to make data move better in networks, ensuring it's reliable, secure, and connects smoothly.

The project successfully did what it set out to do. For example, the login process is secure, and users can safely upload different-sized files in various formats. The system alsosmoothly handles recipient email input and sending emails, making communication reliable. Plus, using encryption keeps the transmitted files secure and private.

From a technical viewpoint, the project used cloud computing, specifically AWS, to host the application. Choosing the Node.js web framework helped create an application that can grow and work efficiently. Making sure the authentication, encryption, and file handlingare secure shows a dedication to keeping data safe and maintaining user security.

The thorough testing, covering everything from logging in to decompressing files on the user's computer, confirmed that the "Network Traffic Congestion Control System" works well and is secure. This careful testing ensures that the system performs reliably in different situations, providing users with a smooth experience.

This point is about how essential it is for the technical implementation to meet user requirements and adhere to industry standards. Here are the keys:

Approach based on users:
This project will be successful only if it can clearly meet all demands of end-users, who ask for a particular application to help in their daily activities. The criteria include such features

as secure file sharing, authentication and authorization, usability, etc. User preferences should be accounted for through regular testing of different versions and getting feedback from them.

Compliance and Security:

Considering the fact that file sharing and email communication deal with sensitive information, adhering to data protection regulations as well as security best practices becomes vital. Robust encryption mechanisms must be implemented, strict access controls maintained, periodic security audits carried out in order to protect users' data against cybercrime and comply with relevant laws such as GDPR or HIPPA.

Scalability and Performance:

The number of users may grow or decrease while individuals generate more files over time; therefore, the system has to expand smoothly without compromising its performance or reliability even when there is increased workload. Smooth operat-ions at least require carefully examining architectural design, using resources efficiently, and implementing measures that optimize performance proactively.

In the end, successfully creating the "Network Traffic Congestion Control System" is a major achievement in solving issues related to network congestion. By using advanced technologies and following the best methods, the project establishes the basis for a digital system that is more reliable, secure, and responsive. The outcomes of this project can significantly benefit different areas, ensuring a strong and efficient connection in theever-growing digital landscape.

6.2 FUTURE WORK

In future work, the project aims to further enhance its capabilities by introducing a load balancer for traffic management in the cloud. The addition of a load balancer will contribute to the scalability and efficiency of the system, ensuring optimal resource utilization and improved responsiveness under varying network loads.

Additionally, continuous research and development efforts will focus on keeping up with emerging technologies and evolving network architectures. This includes exploring advancements in machine learning models, cloud-native optimizations, and adapting the system to accommodate the changing landscape of internet-enabled devices.

The future work also entails ongoing evaluations of the effectiveness of congestion control mechanisms and the incorporation of new algorithms as they emerge. By staying proactive in addressing the evolving challenges of network congestion, the project aims to remain at the forefront of ensuring a seamless and responsive network experience.

REFERENCES

[1] Sahni, Ishika & Kaur, Araftoz. (2022). A Systematic Literature Review on 5G Security. 10.48550/arXiv.2212.03299.

[2] Josip Lorincz;Zvonimir Klarin;Julije Ožegović; (2021). A Comprehensive Overview of TCP Congestion Control in 5G Networks: Research Challenges and FuturePerspectives . Sensors, (), –. doi:10.3390/s21134510

[3]S. Sunassee, A. Mungur, S. Armoogum and S. Pudaruth, "A Comprehensive Review on Congestion Control Techniques in Networking," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 305-312, doi: 10.1109/ICCMC51019.2021.9418329.

[4]M. Agiwal, A. Roy, and N. Saxena.Security of 5G-mobile backhaul networks. IEEE Communications Surveys & Tutorials, 18(3):1617–1655, February 2016.

[5]V. Sharma, I. You and N. Guizani, "Security of 5G-V2X: Technologies, Standardization, and Research Directions," in IEEE Network, vol. 34, no. 5, pp. 306-314, September/October 2020, doi: 10.1109/MNET.001.1900662.

[6] Ahmad, Ijaz & Kumar, Tanesh & Liyanage, Madhusanka & Okwuibe, Jude & Ylianttila, Mika & Gurtov, Andrei. (2017). 5G Security: Analysis of Threats and Solutions. 10.1109/CSCN.2017.8088621.

[7] Dr, & Koca, Murat & Avci, İsa. (2023). OVERVIEW OF 5G ARCHITECTURE SECURITY.

[8]S. Sullivan, A. Brighente, S. A. P. Kumar and M. Conti, "5G Security Challenges and Solutions: A Review by OSI Layers," in IEEE Access, vol. 9, pp. 116294-116314, 2021, doi: 10.1109/ACCESS.2021.3105396.

[9]A. El Masri, A. Sardouk, L. Khoukhi, A. Hafid and D. Gaiti, "Neighborhood-Aware and Overhead-Free Congestion Control for IEEE 802.11 Wireless Mesh Networks," in IEEE Transactions on Wireless Communications, vol. 13, no. 10, pp. 5878-5892, Oct. 2014, doi: 10.1109/TWC.2014.2349898.

[10]Hasan, Mohammad Kamrul & Ghazal, Taher & Saeed, Rashid & Pandey, Bishwajeet & Gohel, Hardik & Eshmawi, Ala & Abdel‐Khalek, S. & Alkhassawneh, Hula. (2021). A review on security threats, vulnerabilities, and counter measures of 5G enabled Internet‐of‐Medical‐Things. IET Communications. 16. 1-12. 10.1049/cmu2.12301.

[11]S. Ryu, C. Rump and C. Qiao, "Advances in internet congestion control," in IEEE Communications Surveys & Tutorials, vol. 5, no. 1, pp. 28-39, Third Quarter 2003, doi: 10.1109/COMST.2003.5342228.

[12]A. E. Eckberg, "B-ISDN/ATM traffic and congestion control," in IEEE Network, vol. 6, no. 5, pp. 28-37, Sept. 1992, doi: 10.1109/65.157030.

[13]Bor-Sen Chen, Sen-Chueh Peng and Ku-Chen Wang, "Traffic modeling, prediction, and congestion control for high-speed networks: a fuzzy AR approach," in IEEE Transactions on Fuzzy Systems, vol. 8, no. 5, pp. 491-508, Oct. 2000, doi: 10.1109/91.873574.

[14]H. M. Monti, A. R. Butt and S. S. Vazhkudai, "CATCH: A Cloud-Based Adaptive Data Transfer Service for HPC," 2011 IEEE International Parallel & Distributed Processing Symposium, Anchorage, AK, USA, 2011, pp. 1242-1253, doi: 10.1109/IPDPS.2011.118.

[15]V. Sharma, I. You and N. Guizani, "Security of 5G-V2X: Technologies, Standardization, and Research Directions," in IEEE Network, vol. 34, no. 5, pp. 306-314, September/October 2020, doi: 10.1109/MNET.001.1900662.

[16]M. Alizadeh et al., "Data center transport mechanisms: Congestion control theory and IEEE standardization," 2008 46th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 2008, pp. 1270-1277, doi:

10.1109/ALLERTON.2008.4797706.

[17]X. Shen, X. Cheng, R. Zhang, B. Jiao and Y. Yang, "Distributed Congestion Control Approaches for the IEEE 802.11p Vehicular Networks," in IEEE Intelligent Transportation Systems Magazine, vol. 5, no. 4, pp. 50-61, winter 2013, doi: 10.1109/MITS.2013.2279176.

[18]Y. Dong, D. Makrakis and T. Sullivan, "Network congestion control in ad hoc IEEE 802.11 wireless LAN," CCECE 2003 - Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No.03CH37436), Montreal, QC, Canada, 2003, pp. 1667-1670 vol.3, doi: 10.1109/CCECE.2003.1226229.

[19]A. Murali, K. Bhanupriya, S. B. Smitha and G. N. Kumar, "Performance evaluationof IEEE 802.11p for vehicular traffic congestion control," 2011 11th International Conference on ITS Telecommunications, St. Petersburg, Russia, 2011, pp. 732-737, doi: 10.1109/ITST.2011.6060151.

[20]K. Xiao, S. Mao and J. K. Tugnait, "TCP-Drinc: Smart Congestion Control Based on Deep Reinforcement Learning," in IEEE Access, vol. 7, pp. 11892-11904, 2019, doi: 10.1109/ACCESS.2019.2892046..

[21]H. Han, S. Shakkottai, C. V. Hollot, R. Srikant and D. Towsley, "Multi-Path TCP: A Joint Congestion Control and Routing Scheme to Exploit Path Diversity in the Internet," in IEEE/ACM Transactions on Networking, vol. 14, no. 6, pp. 1260-1271, Dec. 2006, doi: 10.1109/TNET.2006.886738.

[22]C. Xu, J. Zhao and G. -M. Muntean, "Congestion Control Design for Multipath Transport Protocols: A Survey," in IEEE Communications Surveys & Tutorials, vol. 18, no. 4, pp. 2948-2969, Fourthquarter 2016, doi: 10.1109/COMST.2016.2558818.

[23]S. Sullivan, A. Brighente, S. A. P. Kumar and M. Conti, "5G Security Challenges and Solutions: A Review by OSI Layers," in IEEE Access, vol. 9, pp. 116294-116314, 2021, doi: 10.1109/ACCESS.2021.3105396.

[24]M. Kim et al., "Congestion control for coded transport layers," 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 2014, pp. 1228-1234, doi: 10.1109/ICC.2014.6883489.

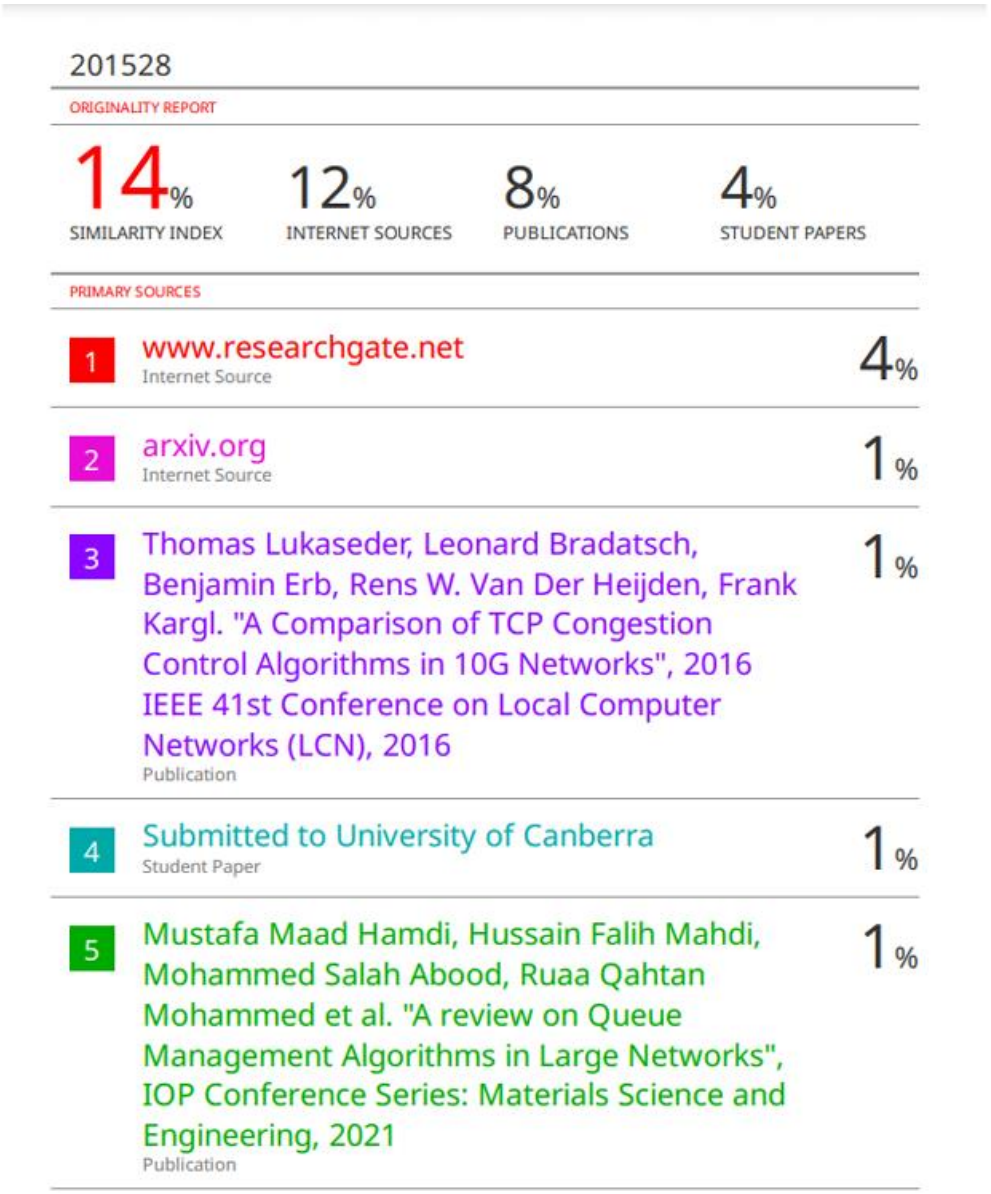
[25]N. Taherkhani and S. Pierre, "Centralized and Localized Data Congestion Control Strategy for Vehicular Ad Hoc Networks Using a Machine Learning ClusteringAlgorithm," in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 11, pp. 3275-3285, Nov. 2016, doi: 10.1109/TITS.2016.2546555.

[26]W. Wei, K. Xue, J. Han, D. S. L. Wei and P. Hong, "Shared Bottleneck-Based Congestion Control and Packet Scheduling for Multipath TCP," in IEEE/ACM Transactions on Networking, vol. 28, no. 2, pp. 653-666, April 2020, doi: 10.1109/TNET.2020.2970032.

[27]W. Li, F. Zhou, K. R. Chowdhury and W. Meleis, "QTCP: Adaptive Congestion Control with Reinforcement Learning," in IEEE Transactions on Network Science and Engineering, vol. 6, no. 3, pp. 445-458, 1 July-Sept. 2019, doi: 10.1109/TNSE.2018.2835758.

[28]P. Key, L. Massoulie and D. Towsley, "Combining Multipath Routing and Congestion Control for Robustness," 2006 40th Annual Conference on Information Sciences and Systems, Princeton, NJ, USA, 2006, pp. 345-350, doi: 10.1109/CISS.2006.286490.

PLAGIARISM REPORT



201528

ORIGINALITY REPORT

14%	12%	8%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	4%
2	arxiv.org Internet Source	1%
3	Thomas Lukaseder, Leonard Bradatsch, Benjamin Erb, Rens W. Van Der Heijden, Frank Kargl. "A Comparison of TCP Congestion Control Algorithms in 10G Networks", 2016 IEEE 41st Conference on Local Computer Networks (LCN), 2016 Publication	1%
4	Submitted to University of Canberra Student Paper	1%
5	Mustafa Maad Hamdi, Hussain Falih Mahdi, Mohammed Salah Abood, Ruaa Qahtan Mohammed et al. "A review on Queue Management Algorithms in Large Networks", IOP Conference Series: Materials Science and Engineering, 2021 Publication	1%