Congestion Control for Wireless Multimedia Sensor Networks

Uzma Majeed

Registration No. 107-FET-PHDEE-S16

Supervised By: Prof. Dr. Aqdas Naveed Malik

Co-Supervisor: Prof. Dr. Ijaz Mansoor Qureshi

Co-Supervisor: Dr. Nasim Abbas

A dissertation submitted to IIUI in partial

fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Ph.D.)

Department of Electrical and Computer Engineering
Faculty of Engineering and Technology
INTERNATIONAL ISLAMIC UNIVERSITY
ISLAMABAD

2025

DEDICATED TO

I, dedicate this Thesis to Almighty ALLAH for guidance, strength and power of mind, skills and healthy life. Then to my Prophet (Piece and blessings be upon him), Ehl-e-Bait, his companions and his followers, whole Muslim-ummah, my mother, my father, my teachers, my beloved brothers, family members and friends.

ABSTRACT

Wireless multimedia sensor networks (WMSNs) generate a huge amount of multimedia data. Congestion is one of the most challenging open issues in WMSNs. Congestion causes low throughput, high packet loss, and low energy efficiency. Congestion happens when the data carried by the network surpasses the available capacity. This dissertation, "**Congestion control for Wireless Multimedia Sensor Networks**" presents the implementation of a WMSNs system that is capable of controlling congestion in an efficient manner. To develop this system, firstly, a comprehensive review of different congestion control algorithms is provided.

Secondly, an energy-efficient distributed congestion control protocol (DCCP) is proposed to mitigate congestion and improve end-to-end delay. Compared to the other protocols, the DCCP protocol can alleviate congestion by intelligently selecting the best path. First, congestion is detected using two congestion indicators. Second, the traffic congestion map is used to calculate the ideal path. Therefore, the traffic is balanced on different routes, which reduces the end-to-end delay. Finally, a rate controller is designed to prevent congestion in the network by sending a congestion notification message to a source node. After receiving the congestion notification message, the source node immediately adjusts its transmission rate. Experimental results based on Raspberry Pi sensor nodes show that the proposed DCCP protocol significantly improves network performance and is superior to modern congestion control protocols.

Thirdly, a buffer occupancy-based congestion control protocol for wireless multimedia sensor networks (BOCC) is proposed. First, congestion is detected using two congestion indicators. Secondly, a rate controller is designed to protect high-priority I-frame packets during congestion. The proposed algorithm adjusts the rate by discarding low-priority P-frame packets from the source nodes. Therefore, high-priority packets are protected at the expense of low-priority packets. Experimental results show that the proposed BOCC protocol is superior to modern congestion control protocols.

Fourthly, the algorithm proposed in this dissertation is verified and evaluated on the Raspberry Pi board.

Keywords: wireless sensor networks; congestion control; wireless video transmission; wireless multimedia sensor networks

LIST OF PUBLICATIONS OUT OF THE WORK

LIST OF PUBLICATIONS

1. **Majeed, U**.; Malik, A.N.; Abbas, N.; Alfakeeh, A.S.; Javed, M.A.; Abbass, W. Buffer Occupancy-Based Congestion Control Protocol for Wireless Multimedia Sensor Networks. *Electronics* **2024**, *13*, 4454. *https://doi.org/10.3390/electronics13224454 (published)*

2. **Majeed, U**.; Malik, A.N.; Abbas, N.; Abbass, W. An Energy-Efficient Distributed Congestion Control Protocol for Wireless Multimedia Sensor Networks. Electronics **2022**, 11, 3265. https://doi.org/10.3390/electronics11203265
(IF 2.4) (Published)

3. Abbas, N., Yu, F., & **Majeed, U**. (**2018**, May). Reliability and end-to-end delay evaluation of outdoor and indoor environments for wireless multimedia sensor networks. In *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)* (pp. 764-768). IEEE.

ACKNOWLEDGEMENTS

First and foremost, I am deeply grateful to Almighty Allah for His countless blessings and guidance throughout this journey.
I am truly grateful to my supervisor Prof. Dr. Aqdas Naveed Malik for his kind supervision. I am also thankful to my co-supervisor Dr. Nasim Abbas for his continuous support.

To my beloved late father Abdul Majeed, whose love, prayers, and values continue to guide and inspire me. His memory has been my source of strength throughout this endeavor.

I extend my heartfelt gratitude to my mother for her endless prayers, unwavering support, and encouragement. You are the foundation of everything I have achieved, and I am forever grateful for your endless sacrifices and boundless encouragement.

To my brothers, who stood by me at every step, providing relentless support, guidance, and encouragement, I am forever indebted.

To my in-laws, thank you for your constant belief in me and your uplifting words during challenging times.

To my husband, who lovingly called me “half doctor” throughout this journey, your faith in me, your Love, and your endless support have been my greatest strengths. To my beloved daughters, Syeda Gul-e-Zahra and Syeda Ayat Zahra, balancing sleepless nights, your cries, and this degree was challenging, but your smiles made it all worthwhile. You are my inspiration, and this achievement is as much yours as it is mine. This milestone would not have been possible without the love, prayers, and support of my family.

Engr. Uzma Majeed

Table of Contents

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

WSN	Wireless sensor networks
WMSN	Wireless multimedia sensor networks
IoT	Internet of things
QoS	Quality of service
BoCC	Buffer occupancy-based congestion control protocol
AIMD	Additive increase multiplicative decrease
ACK	Acknowledgement
NACK	Negative Acknowledgement

Chapter 1

Introduction

A network that includes a significant number of, low- power and self-oriented devices (sensor nodes) is called wireless sensor networks (WSNs) [1]. Nowadays, WSNs are gaining more attention in the networking research community due to their low-cost, low- power, and extended wireless communication capabilities. The tiny sensor nodes that work together to create a network are called motes. Mote is generally a multipurpose and energy efficient device. There is a wide need for these motes to be used in various fields. Motes work together as a group to gather data from the surroundings to achieve certain goals. These small motes usually communicate with each other through a transceiver. The sensor nodes in a network can grow to hundreds and even thousands. The optimum performance can be achieved by interconnecting these devices in different topologies [2]. WSNs extend to wireless multimedia sensor networks (WMSNs). WMSNs consist of wireless multimedia sensor nodes. These WMSN nodes contain image, video or audio. Each node can sense, process and forward collected information to the sink node. The sensor nodes self-organize, create a network, analyze the data, and forward the data to the receiver through multi-hop communication [3].

This chapter presents the introduction of WMSNs, motivation and research problems, research objectives, main research contributions, research methodology, research scope, and organization of the thesis.

1.1 Wireless Multimedia Sensor Networks

1.1.1 Background and Significance

WMSNs are one of the promising paradigms for the Internet of Things (IoT) [1]. WMSNs consist of wirelessly connected devices that can capture video data from the environment. Sensor nodes are incorporating inexpensive video cameras and microphones [2]. These camera nodes capture this visual information, process it and transmit it to the base station through multi-hop or single-hop communication. WMSNs are used in various applications. The scope of WMSN applications includes monitoring of indoor and outdoor environments

[3], IoT-based smart agriculture [4] and object tracking [5]. The transport layer [6,7], the network layer [8], the physical layer [9] and the media access control layer [10,11] have been extensively researched in WMSNs.

Due to many practical and theoretical challenges, WMSNs have attracted the attention of many researchers. Many challenges need to be addressed, such as harsh environmental conditions [12], node failures [13], mobility of nodes [14], mobility of detected events [15], dynamic network topology [16], heterogeneity of nodes [17], large scale deployments [18] and unattended operations [19]. In addition, there is tremendous progress in internet traffic applications that require bandwidth. Industrial forecasts predict that large- scale video traffic will dominate global internet traffic in the future [20].

1.1.2 Characteristics of WMSNs

A WMSN consists of a network of sensor nodes. These sensor nodes can carry still images, audio, or video streams. Generally, WMSN architecture consists of source nodes, gateway nodes, and sink nodes [21-23]. In WMSN, various sensor nodes self-configure and establish a network. These sensor nodes observe the collected data and send the data to the receiver through multi-hop communication [24-25].

WMSN also has other features and challenges. This is because of the nature of real-time multimedia data, such as reliability, bandwidth, latency, and high bandwidth [26-27]. For instance, video surveillance applications need packets to reach the sink node without any loss. In WMSNs, different types of packets require different reliability constraints and priorities. I-frame packets have higher priority and demand higher reliability, and P-frame packets have lower priority therefore, these packets demand less reliability constraints. The packet loss in WMSNs is mainly due to congestion. Priority packets in WMSNs require both reliable and timely communication. Therefore, robust techniques should be applied to accomplish the need for WMSNs applications [28].

Video data always consumes a large bandwidth. Not only sensor nodes generate their new video data, but also, they have to relay the traffic of other sensor nodes. If a huge amount of data is sent over a single route, it will exhaust the route and result in network collapse. To solve this issue, multiple paths and multiple channels can be created in a spatially overlapped manner. Therefore, in WMSNs, large bandwidth constraints should be taken into account when planning congestion control algorithms [29-30].

1.1.3 Applications of WMSNs

WMSNs are utilized across variety of applications. The most common WMSNs application is surveillance [31, 32]. The existing surveillance techniques can be upgraded and enhanced using more sophisticated audio and video sensors. Video sensor nodes can record events like traffic violations and robberies. Sensor nodes will send the recorded event to the monitoring station or sink through multi-hop communication. Moreover, surveillance applications can also be used to visualize missing persons.

Another application in WMSNs is the traffic monitoring application. This application is used to observe traffic on roads. Sensor nodes can be deployed to suggest traffic routing to eliminate congestion [33, 34]. Furthermore, sensor nodes can also visualize the flow of vehicles on roads and obtain necessary information, such as total number of cars and average speed. In addition, sensor nodes can visualize vehicle traffic on the road and get important information such as a total number of cars and average speed. Sensor nodes can also sense breaches in traffic and send videos to relevant organizations to recognize the culpable. Moreover, buffered images and videos can be used for subsequent analysis during an accident. Furthermore, sensor nodes can build a smart and innovative parking system. This intelligent parking system monitors parking spaces and provides parking advice to drivers to enhance mobility in metropolitan areas.

Another application in WMSNs is the personal and health care [35] application. An example of an individual and health care application is a telemedicine sensor network. Ubiquitous health-care examination can be provided to patients by integrating third-generation networks into telemedicine sensor networks. Medical sensors will be attached to patients to visualize different features, such as blood pressure and body temperature. Moreover, remote medical hospitals can monitor their patients via audio and video sensors.

Another application that is most commonly used is WMSNs, which are industrial and environmental applications [36]. Habitat monitoring is one example of industrial and environmental applications in WMSNs. Oceanographers use an array of video sensor nodes to determine the development of sandbars. Video sensor nodes can transmit multimedia content in time-critical industrial process control.

1.2 Network Congestion

Network congestion decreases the efficiency and reliability of WMSNs. Moreover, eliminating congestion is of prime importance to smooth video traffic in WMSN. Therefore, due to high data rate requirements, congestion should be controlled efficiently in the WMSN [37-40].

1.2.1 Causes of Congestion

In WMSNs, there is a rapid increase in data rate at the sink node. Multiple nodes simultaneously send data to the sink node. In this case, the sink node cannot process further data from the incoming stream, resulting in a large amount of packet loss. WMSN need superior quality of service (QoS) guarantees, and congestion can reduce essential resources that can cause network collapse. Therefore, it is very important to obtain reliable data delivery in conjunction with end-to-end delay in WMSNs [41-43].

Figure 1.1 shows the rapid increase in data rate at the sink node. Multiple nodes simultaneously send data to the sink node. In this case, the sink node cannot process further data from the incoming stream, resulting in a large amount of packet loss. WMSN need superior QoS guarantees [99], and congestion can reduce important resources that can cause network collapse.

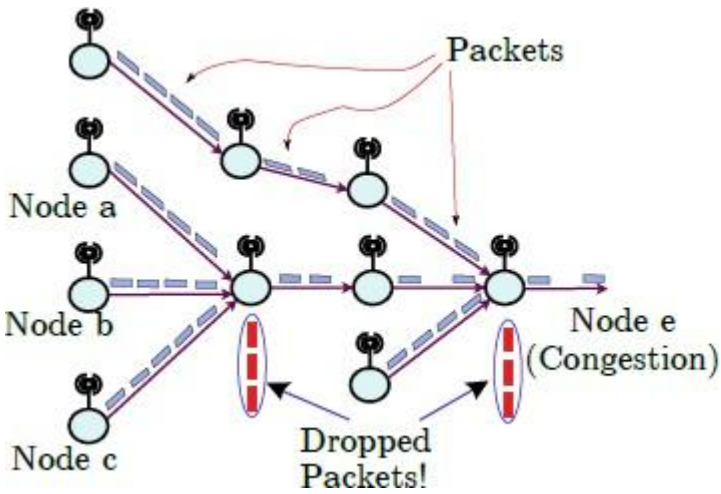

Figure.1.1 Data flow mechanism in WMSNs

1.2.2 Congestion Control Methods

Congestion in WMSNs is avoided using congestion detection, notification, and congestion control [44-45]. Now each of the following will be elaborated.

1.2.2.1 Congestion Detection: Congestion can be detected using four approaches

a) Buffer occupancy: The buffer occupancy process analyzes the congestion situation using buffer occupancy.

b) Wireless Channel Load: Congestion detection using wireless channel load only considers packet load in the channel. The congestion action in this scenario is taken when the time frame of a packet goes beyond the specified limit.

c) Buffer Occupancy and Wireless Channel Load: Congestion detection takes place using either buffer occupancy or wireless channel load. Clusters are formed and the traffic intensity is used to detect congestion.

d) Packet Transmission Time: Congestion control in this approach uses packet inter-arrival time, packet service time, or a combination of both.

Buffer occupancy is a very efficient procedure to detect congestion, but it relies too much on buffer length. The wireless channel load procedure efficiently solves the problems of medium collisions. However, this method does not work if the buffer is full and packets start dropping. Packet transmission time is another efficient method to detect congestion but it relies significantly on the applications.

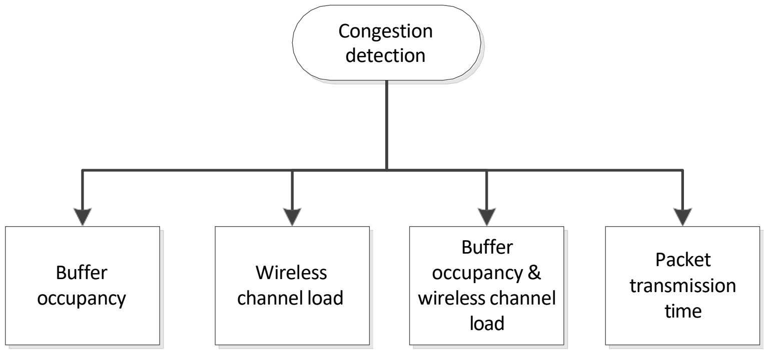

Figure 1.2 Congestion detection mechanisms

1.2.2.2 Congestion Notification: Congestion notification can either be implicit or explicit in nature.

Explicit Congestion Notification: In the explicit congestion notification process, the congested node informs about the congestion situation to other nodes using additional control packets. This technique is used in several congestion control mechanisms. However, these additional control packets need heavy traffic load unsuitable for WMSNs.

Implicit Congestion Notification: This mechanism is used in most congestion control algorithms. This mechanism provides a lower traffic load than the explicit congestion notification mechanism. This mechanism provides no additional overhead. Therefore, it is more suitable for WMSNs.

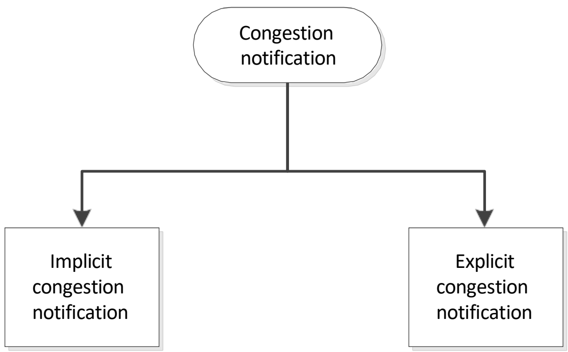

Figure 1.3 Congestion notification methods

1.2.2.3 Congestion Control:

When packets are lost due to buffer overflow, the transmission rate should be progressively decreased to prevent congestion. In response to how congested the neighbors are, the source nodes adjust their data transmission rates while maintaining their rate [44-46]. Every sensor node creates a path leading from it to the parent node. In order to improve the video quality of supplied data, this feedback mechanism communicates data rate information from the parent node to the child nodes [46].

1.3 Research Problem and Motivation

WMSNs are widely accepted in a variety of surveillance environments. However, there are considerable obstacles and barriers to efficiently deploying WMSNs in various real-time environments [47]. There is still a need to address important QoS matrices like reliability, packet loss, and latency in WMSNs. Congestion control has been studied in WMSNs in recent years [48-50]. Numerous congestion control algorithms have been investigated during the past decade. However, due to error-prone wireless communication and different characteristics (i.e, topologies, large-scale networks) of WMSNs, the majority of the congestion control algorithms are not applicable to WMSNs. Generally, conventional congestion control algorithms do not use priority of packets. The packet priority is very important parameter in WMSNs. Therefore, WMSN urgently needs robust congestion control techniques.

Although some work on congestion control has been done in WMSNs, it does not focus on real-time solutions.

In other words, congestion can lead to packet loss, service degradation and reduced quality of service. For instance, I-frame packets in WMSNs carry very important video data. I-frame packet loss results in severe degradation of video quality. Therefore, these very important I-frame packets should be retransmitted immediately to ensure the reliability of the network. Therefore, congestion, along with the retransmission of important packets, is very important in WMSNs. Moreover, congestion also diminishes wireless channel performance, leads to discontinuation of channel connection, and causes energy consumption. The effect of congestion becomes severe with a higher number of hops traversed. The more the packets travel, the higher the packets are lost. Congestion also results in high transmission delays and increased retransmissions. When bandwidth is fully occupied, forwarding data from sensor nodes cannot be accepted by a sink node, and this sensed data will be expired or diminished. This results in the loss of important data.

1.4 Research Contributions

Firstly, the design challenges of WMSNs are given, and then a thorough discussion of the congestion control algorithms already proposed in the literature is given.

Secondly, a distributed congestion control protocol (DCCP) is proposed to alleviate congestion and improve end-to-end delay. The DCCP protocol proposed in this dissertation can alleviate congestion by intelligently selecting the optimum path. First, Congestion is identified through the use of two congestion indicators. Second, a traffic congestion map is used to calculate the best path. As a result, traffic is distributed across different routes, minimizing end-to-end delay. Finally, a rate controller is designed to avoid network congestion by transmitting a congestion notification message to the source node. After receiving a congestion notification message, the source node immediately adjusts its transmission rate. Experimental results using Raspberry Pi sensor nodes demonstrate that the proposed DCCP protocol significantly enhances network performance and outperforms current advanced congestion control protocols.

Thirdly, a buffer occupancy-based congestion control protocol (BOCC) is proposed. Compared to the other protocols, the BOCC protocol proposed in this chapter can alleviate congestion by intelligently saving the data that contains more information. First, congestion is detected using two congestion indicators. Secondly, a rate controller is developed to safeguard high-priority I-frame packets during periods of congestion. BOCC prevents congestion in the network by sending a congestion notification message to a source node. After receiving a congestion notification message, the source node immediately adjusts its transmission rate. In the proposed algorithm, the rate adjustment is made by discarding low-priority P-frame packets from the source nodes. Therefore, high-priority packets are protected at the expense of low- priority packets. Experimental results utilizing Raspberry Pi sensor nodes indicate that the proposed BOCC protocol substantially enhances network performance and outperforms contemporary congestion control protocols.

Fourthly, the algorithm proposed in the chapter is verified and evaluated on the Raspberry Pi board.

1.5 Organization of Thesis

In this section, the contents and contributions of each chapter are discussed in detail. The primary focus of all the chapters is to study congestion and develop an algorithm to eliminate congestion. Chapter 1 explains the introduction, motivation and research problems, research objectives, main contributions, research methodology, and research scope of the thesis. Chapter 2 provides a detailed discussion of related work in the domain of congestion control for WMSNs.

Chapter 3 proposes and energy-efficient DCCP to mitigate congestion and improve end-to-end delay. First, congestion is detected using two congestion indicators. Secondly, traffic is distributed across various routes, resulting in a reduction of end-to-end delay. Finally, a rate controller is designed to avert network congestion by transmitting a congestion notification message to the source node. After receiving the congestion notification message, the source node immediately adjusts its transmission rate. Experimental results using Raspberry Pi sensor nodes demonstrate that the proposed DCCP protocol greatly enhances network performance and outperforms current congestion control protocols.

Chapter 4 proposes a BOCC. Compared to the other protocols, the BOCC protocol proposed in this chapter can alleviate congestion by intelligently saving the data that contains more information. First, congestion is detected using two congestion indicators. Secondly, a rate controller is designed to protect high-priority I-frame packets during congestion. BOCC prevents congestion in the network by sending a congestion notification message to a source node. After receiving congestion notification message, the source node immediately adjusts its transmission rate. In the proposed algorithm, the rate adjustment is done by discarding low priority P-frame packets from the source nodes. Therefore, high priority packets are protected at the expense of low priority packets.

Chapter 5 summarizes the key findings of the research, highlighting its contributions and impact. It also outlines potential future research directions to enhance and extend the proposed work.

Chapter 2

Background and Literature Review

In WMSNs, congestion happens when data carried by the network surpasses the available capacity. Congestion results in exhausted node energy, degradation of network performance, high packet loss, and an increase in network latency. Several factors cause congestion in WMSNs such as buffer overflow, the dynamic nature of a channel, and many-to-one communication. Therefore, designing congestion control protocols in WMSNs to control congestion efficiently is very important. In this chapter, a comprehensive review is presented. It is observed that the performance of each algorithm varies according to its deployment. It is also observed that a single metric cannot accurately detect and control congestion in WMSN.

2.1 Congestion Control Algorithms

2.1.1 Traffic-based Congestion Control Mechanism

This approach is further distributed into end-to-end and hop-by-hop-based traffic control approaches.

The packet information is stored in the source node in the end-to-end congestion control mechanism. If the packet is lost, the corresponding packet will be retransmitted immediately. In the hop-by-hop congestion control mechanism, sensor nodes establish a route from themselves to the parent node.

The option of using an end-to-end or hop-by-hop approach depends on the parameters like reliability and end-to-end delay.

A priority rate-based routing protocol (PRRP) uses a hop-by-hop-based mechanism to control congestion [51]. PRPP uses a three-way strategy. Congestion detection, congestion notification, and congestion control. PRRP uses buffer occupancy to detect congestion. Two different thresholds are used to compute buffer occupancy at each node in PRRP. If buffer occupancy falls below the minimum limit, the data rate is maintained and no congestion is detected. If buffer occupancy exceeds the upper limit, congestion is detected immediately and children nodes slow down their traffic rate to minimize congestion. PRRP cannot choose an optimal path from the source to the sink.

Node weight is introduced in weighted fairness guaranteed congestion control (WFCC) to reveal the significance of sensor nodes [52]. In WFCC, authors assign weights to high-priority packets. They assume that traffic at different sensor nodes differs considerably because of differences in the priorities of packets. The congestion control mechanism in WFCC is implemented by varying the incoming traffic rate regularly. WFCC generates feedback at each interval, which results in additional overhead.

Congestion control based on reliable transmission (CCRT) ensures reliable data transmission [53]. It uses an end-to-end congestion control technique. Each sensor node in CCRT has different priority buffers for different types of traffic. Higher priority is assigned to important data, and lower priority is assigned to less important data. All queues use first-in- first-out (FIFO) criteria. CCRT uses both buffer length and its variation together for congestion detection. If the queue variation rate exceeds a certain limit, there is a likelihood of congestion occurrence. The congestion will be mitigated if the queue variation rate is lower than the specified limit. On the other hand, if the data rate continuously increases, the queue will overflow, resulting in congestion. CCRT reduces end-to-end delay, improves packet loss, and increases throughput. However, CCRT does not consider link-layer congestion.

Priority-based application-specific congestion control clustering protocol (PASCCC) detects congestion by integrating heterogeneity and mobility of sensor nodes [54]. In

PASCCC, each node triggers its radio whenever the reading of the captured incident goes beyond a certain limit. The sensor nodes sense, collect, and forward the data to the sink. When congestion is detected, PASCCC ensures the delivery of high-priority packets to the destination by eliminating less important packets. Therefore, PASCCC achieves reliable data transmission for high-priority packets. Moreover, the sensor nodes, placed at a distant location from the cluster head get high priority over a neighboring node. Furthermore, using a novel queuing scheduling approach, PASCCC minimizes the overhead caused by remote location sensor nodes. The disadvantage of PASCCC is that it causes additional delays in the network. This delay is because of changes in the position of sensor nodes. Moreover, PASCCC drops humidity packets. This results in drastic changes in weather forecasting and environmental monitoring.

The buffer occupancy-based transport layer protocol uses the buffer occupancy-based mechanism for congestion detection [55]. Two different thresholds are used to compute buffer occupancy at each node. If the buffer occupancy surpasses the upper limit, congestion is detected immediately, and child nodes slow down the traffic rate to reduce congestion. However, it cannot choose the optimal path. The Adaptive Weight Firefly (AWF) algorithm reduces network congestion by clustering two algorithms [56]. The rate control mechanism is only used in the case of negative acknowledgment (NACK). Moreover, an Ant Colony Routing-based mechanism is used to enhance the throughput in the network. However, an acknowledgment in this algorithm increases the delay in the network. The Traffic- Aware Congestion Control Protocol (TACC) solves the rate adaptation problem in the transport layer of WMSNs [57]. TACC detects congestion using burst loss information and the presence of burst loss at the sink node. The sending rate is conservatively reduced if congestion occurs for the first time. TACC adjusts the sending rate aggressively if the network is continuously congested. However, TACC needs to be enhanced further to facilitate the conveyance of prioritized events to different flows.

2.1.2 Resource-based Congestion Control Protocols

In this section, the resource-based congestion control approach is explained. If the traffic-based rate control method cannot manage the needs of the wireless multimedia sensor network, a resource-based congestion control method is used. This method is particularly used in high-reliability applications that require lower time delays. It reduces congestion by balancing the traffic load by utilizing idle network resources or uncongested paths. Therefore, data packets have a higher probability of error-free communication. However, the resource control approach needs additional overhead in the form of end-to-end topology information, loop avoidance, and packet travel time for sensor nodes. This section is divided into two sub-sections. One is the dynamic alternative routes section, and the other is the allocation of available bandwidth section.

Opt-ACM mitigates congestion using different routes for data routing with different QoS [58]. The Mixed-Integer Linear Programming mechanism is used to verify the proposed algorithm. However, the proposed algorithm does not focus on energy efficiency in the network. A routing technique based on Deep Reinforcement Learning (DRL) minimizes the end-to-end delay using the unequal clustering scheme [59]. The proposed algorithm prevents the network by splitting the entire network into unequal clusters. However, the complex methodology is adapted to use unequal clustering. The Time Delay- based Multipath Routing (TMR) algorithm substantially enhances the network capacity by choosing an alternative path [60]. TMR maintains a unique ID and mitigates congestion using a reactive technique. TMR assigns higher priority to alert messages to send traffic to different routes that are not congested. The SLEB protocol combines load balancing and security verification mechanisms based on clustered WSNs [61]. This protocol is effective in balancing the energy of the network. Moreover, it also enhances the security overhead. Therefore, SLEB can prolong the network lifetime. SLEB has a much lower overhead.

However, in SLEB, multiple sensor nodes can send the same data to the receiver node. DHSSRP prioritizes significant data to mitigate congestion in the network. In DHSSRP, the authors assign weights to high-priority packets. They assume that traffic in different sensor nodes varies considerably due to differences in packet priorities. DHSSRP is a lightweight protocol that increases the lifespan of the deployed sensor nodes by diverting the data traffic to alternative routes.

2.1.3 Hybrid Congestion Control Protocols

Typically, a traffic-based congestion control approach is used in these protocols. However, if the traffic-based congestion control approach is not feasible then resource-based congestion control approach is used.

HRTC controls congestion using a hybrid congestion control approach [62]. HRTC uses a combination of resource-based congestion control protocol and traffic-based congestion control protocol. When congestion occurs in HRTC, congested sensor nodes inform the source nodes to decrease the data rate using multi-hop communication. When the congested downstream node receives the back-pressure information, HRTC tries to use the resource control technique by altering the path. If an alternative path is not available, it uses a traffic-based congestion control mechanism by reducing the data rate of the source node.

HOCA is composed of four phases [63]. In the first phase, the sink broadcasts data to all source nodes. This phase determines the node type, timing, and priority of data. In the second phase, the sink receives data from each node. In the third phase, the sink node determines the path using hop-by-hop communication. In the fourth phase, the data rate is adjusted to reduce congestion. HOCA achieves better energy efficiency and ensures fair allocation of resources. HTCCFL reduces congestion in three steps [64]. In the first step, HTCCFL creates a hierarchical tree. In the second step, is detects congestion by applying a fuzzy-logic mechanism. HTCCFL maintains priority queues during this step. Finally, HTCCFL controls congestion by selecting an alternative route if rate alteration is not possible.

Traffic-based Congestion Management (TCM) controls congestion using a hybrid congestion control approach [65]. When congestion occurs, the congested sensor nodes report to the source multi-hop communication to reduce the data rate. When a congested downstream node receives the back-pressure information, the TCM tries to use the resource control technique by changing the path. If an alternative route path is unavailable, it uses traffic-based congestion control protocols by reducing the data rate of the source node. ECA-HA controls congestion using a hybrid congestion control approach [66]. ECA-HA reduces congestion by using three phases. In the first phase, the ant colony optimization method selects an optimal route. In the second step, the backward ant ensures that paths are constructed successfully, and the forward ant creates congestion-free paths via multi-hop communication. Finally, congestion is controlled by defining an alternative route if rate modification is not feasible. However, the ECA-HA is only suitable for small search space

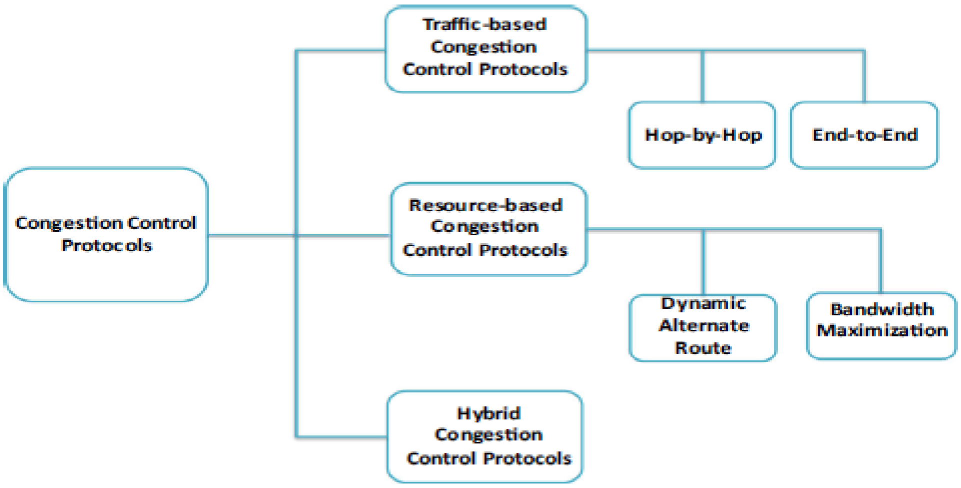

Figure 2.1 Congestion control protocols

2.2 Congestion Control Schemes

Congestion control schemes are divided into three types: (a) congestion control algorithms, (b) congestion avoidance algorithms, and (c) reliable data transmission

2.2.1 Congestion Control Algorithms

CODA: The congestion detection and avoidance algorithm (CODA) is the earliest algorithm in research to discuss congestion control in WSNs [68]. This algorithm is one of the most frequently cited algorithms and serves as the foundation for this research topic. CODA uses three strategies to combat congestion. In CODA, congestion detection has the utmost significance. The current buffer occupancy, previous channel loading circumstances, and congestion detection are all used. CODA uses a sampling approach that only enables channel monitoring when necessary because constant channel listening consumes much energy. When congestion is found, the nodes use a backpressure technique to alert their upstream neighbor nodes directly. Backpressure signals are transmitted in the direction of the source, and the nodes that receive them must decide how to minimize network traffic depending on their local congestion policies.

Furthermore, CODA employs the AIMD method to alleviate congestion by broadcasting numerous sources to a single sink. The source lowers its transmission rate when ACK packets are missing. However, the AIMD-like traffic pattern is not suitable for WMSNs. Moreover, the closed-loop mechanism in CODA leads to high end-to-end delay particularly during congestion.

CCF: The congestion control framework (CCF) detects congestion using packet service time and mitigates congestion using traffic control [69]. Every node makes a correct rate adjustment based on its child node count and available service rate. CCF assumes a tree routing topology with all data source nodes acting as the leaves. CCF frequently lead to congestion in the nodes closer to the base station.

Fusion:

Fusion typically monitors each node's queue size and performs channel sampling at predetermined intervals to identify congestion [70]. Fusion uses a congestion bit during congestion to provide implicit congestion notification to all nodes in a neighborhood. Fusion particularly incorporates three congestion control strategies. Each sensor node sets a congestion bit in the header of each transmitted packet. Therefore, explicit control messages that consume a significant amount of bandwidth are no longer required. Congestion mitigation and detection are the two core parts of hop-by-hop flow control. A congestion bit is set if the queue size of the sensor node exceeds a predetermined threshold.

COMUT: Karenos et al. presented the cluster-based congestion control method For supporting various kinds of traffic in WSNs (COMUT) [71]. Each cluster node in COMUT uses traffic intensity estimation to find congestion. The cluster head receives this estimation and uses it to measure the Congestion level. An AIMD-like traffic management is used to reduce congestion. The COMUT is divided into three components. They were estimating traffic intensity, rate control, and cluster formation. Sensor nodes are grouped together during the cluster-building process. COUMT uses ZRP (zone routing protocol) to create clusters. If high-priority packets are present in the congested path, the transmission rate of the low-priority packets are minimized. The results illustrate that COMUT is very effective in mitigating congestion in the network.

SenTCP. SenTCP is an open-loop hop-by-hop congestion control technique [72]. To detect congestion, it calculates the congestion level in each intermediate sensor node and uses the average packet service time and average packet inter-arrival time simultaneously. The use of packet service time and packet arrival time not only accurately determines the level of congestion, but also helps determine the cause of packet loss. Secondly SenTCP employs hop-by-hop congestion control. Each sensor node in SenTCP transmits a feedback signal backward and hop-by-hop. The feedback signal, carrying the buffer occupancy ratio and local congestion degree, is used by sensor nodes to modify their transmitting rate at the transport layer. Similarly, Hop-by-hop feedback control can reduce packet loss and quickly clear congestion. SenTCP is tested on a simulated simple linear topology with 20 nodes. According to research and a comparison with the TCP, SenTCP can reduce the number of dropped packets caused by buffer overflow.

BGR: Popa et al. design biased geographical routing (BGR) [73] protocol to deal with congestion. BGR is a reactive traffic-splitting protocol. BGR uses implicit congestion notification. Each node can determine the congestion state of neighboring nodes. BGR uses end to end packet scatter algorithm and an in-network packet scatter (IPS) algorithm to reduce congestion. IPS reduces transient congestion by distributing traffic before it becomes congested. EPS uses the AIMD method to reduce permanent congestion. NS-2- based 400-node random topology is used in BGR. BGR performs best when the distance between source and destination is large enough to allow multiple non-interfering paths. Multiple paths cannot be used when short-range flows interfere with long-range flows. BGR throughput is reduced by up to 14% in case of path interference. However, BGR requires node position data supplied by a coordinate system other than GPS. This cost is also not insignificant. Moreover, the BGR AIMD strategy causes a saw-tooth rate behavior. This AIMD strategy is not suitable in WMSNs

TARA: Kang et al. [65] propose the topology-aware resource adaptation (TARA) [74]. TARA focuses on accommodating additional network resources in case of congestion. To detect congestion, TARA uses both buffer occupancy and channel load. A sensor node turns into a hotspot node during congestion.

A detour path that begins at the distributor and ends at the merger is established. A merger combines these two flows while a distributor divides the input traffic between the original and detour path. The distributor and the merger are two crucial sensor nodes that the hot-spot node has to find rapidly. When a hotspot is formed due to congestion, traffic is diverted from the hotspot and moves to the merger, where traffic is merged. The effectiveness of TARA is assessed using an 81-node random topology that is simulated on ns-2. The results show that TARA can effectively handle the incoming traffic load. However, TARA is not scalable as it requires the information of the entire network topology.

PCCP: The PCCP algorithm counters the contentions made by congestion control protocols (such as CCF), which claim that all sensor nodes should have equal fairness [75]. PCCP provides various priority indexes like sensor nodes with a higher priority index will have a higher bandwidth

Any sensor node(s) in any specific location can have their priority index overridden dynamically by the application layer using PCCP. Numerous WSN applications may need this feature. PCCP is tested on two topologies: a linear topology with 10 to 40 nodes and a tree topology with seven nodes. PCCP assumes that all surrounding nodes can have equal access possibilities. According to the simulation results, PCCP (1) attains high link usage and better fairness; (2) reaches a small buffer size, which enables it to prevent packet loss, increase energy efficiency, and give shorter delays. When a node does not create enough traffic, CCF has lower throughput than PCCP. This is because of CCF's inability to utilize a work-conservation scheduling algorithm and properly allocate the system's remaining capacity. Moreover, it was demonstrated that PCCP outperformed CCF in terms of throughput despite packet losses.

HTAP: Sergiou et al. propose a distributed and scalable algorithm named as hierarchical tree alternative path (HTAP) [76]. HTAP reduces congestion in event-based networks and promises reliable and robust data transmission. This algorithm uses the implicit method of notifying about the **congestion**. The network congestion is mitigated by controlling the resource consumption. Therefore, an alternate path is created when the sensor node is near to congestion. An alternative path is created by using the network's unused nodes. The paths created through unused nodes are created dynamically. Therefore, they are not included in the initial calculation of the shortest path distance from the source node to the sink node.

The simulation results show that unused nodes result in a balanced resource consumption. Moreover, a network of "holes" is avoided which leads to extended network life. However, the authors do not consider fairness in their work.

The HTAP is comprised of four main parts

• Flooding with level discovery functionality: It helps the nodes to find their neighbor nodes and updates the neighbor table accordingly. Furthermore, the sensor nodes in the network are placed in phases from the source to the sink node.

• Alternative path creation algorithm:

Each congested node sends a backpressure message to the sending node at the receiving end to prevent congestion. In response to a backpressure message, the sender stops data transmission to that particular congested node. The sender node then looks through its neighbor table to identify a different congested receiving node. The dynamic variation of a congested receiver leads to the new paths from the source node to the sink. The advantage of this approach is that each receiver continuously shares the network traffic with the sending node.

• Handling of Powerless (Dead Nodes): While implementing the HTAP algorithm, a special care is taken for the nodes whenever their battery is drained. Therefore, when a node is about to lose power, it is instantly removed from the network.

CONSISE: Vedantham et al. have proposed a sink to sensor control for the congestion problem [77]. CONSISE handles the congestion by creating paths from sink nodes to source nodes rather than from source nodes to sink nodes. Two factors contribute in sink node to sensor node are: broadcast storms and reverse path contention. In CONSISE, sensor nodes can smoothly control and regulate their packet sending rate based on the determined packet congestion level. The feedback is that the CONSISE is used to adjust the sending rate of downstream nodes. Downstream nodes select their preferred upstream node to update it about the higher packet-sending data rate.

UHCC Wang et al. have presented an upstream hop-by-hop congestion control protocol (UHCC) [78]. UHCC is a cross layer-based protocol designed to reduce packet loss. It ensures reduced control overhead along with priority-oriented fairness. It comprises of two main components: A rate adjustment component and a congestion detection component. A congestion index is used to represent the congestion level of all the nodes. The buffer size and the traffic rate at the MAC layer are taken as input to a congestion index. Each of the upstream node uses the congestion index to adjust its traffic rate in order to reduce the congestion. The results show that UHCC improves the throughput and priority-based fairness.

FACC Yin et al. propose a “Fairness-Aware Congestion Control Scheme” (FACC) to improve the fairness [79]. FACC separates the intermediate nodes into near sink nodes and near source nodes. The near-source nodes monitor each flow and maintain good bandwidth allocation. When the intermediate node experiences congestion then traffic from source nodes is forcefully decreased to match the current available bandwidth. So overall network switches the throughput in accordance with the amount of traffic to avoid congestion.

Transmission control near sink nodes follows a probability-based dropping mechanism that takes action depending on the frequency of incoming flow and the queue size. Stateless queue management favors short flows to achieve fairness. Long flows have a higher probability of dropping from the queue. Hop-by-hop backpressure drops packets from the flow by communicating to near source nodes in order to alleviate congestion. NS-2 Simulator has been used in the implementation of FACC. The results show that FACC improves the packet delivery ratio, energy consumption, and throughput.

CADA: The level of congestion in a node is determined by aggregating channel utilization and buffer occupancy [80]. The node is congested if the buffer occupancy rises above a particular threshold. It suggests that there is channel congestion. Depending on the situation, CADA utilizes both resource control and rate control to reduce congestion. Resource control is employed when there is congestion at an intersection hotspot, whereas traffic control is used when congestion occurs in a convergence hotspot. Results show that CADA improves energy efficiency, throughput, and end-to-end delay.

DAlPaS: In case of congestion, DAIPas select a different route while taking a number of fundamental performance factors into account [81]. It is a distributed and dynamic algorithm. The two stages of DAlPaS operation are the hard and soft stages. The DAlPaS algorithm uses a resource control method to reduce congestion.

2.2.2 Congestion Avoidance Algorithms:

Siphon: Siphon is a source-to-sink congestion control protocol that introduces some virtual sinks (VS) [82]. The primary objectives of the siphon algorithm are to solve VS discovery, congestion detection, operational scope control, congestion avoidance, and traffic redirection. The Physical sink regularly transmits a control packet which includes a signature byte, which is part of the VS discovery process. Each regular sensor node maintains a list of neighbor nodes that can be used to connect to its parent VS. Furthermore, each VS keeps a list of its neighboring VS and has two radio interfaces: a conventional low-power radio to connect with the regular sensors, and a long-range radio to communicate with other physical sink or VSs. Moreover, Siphon uses CODA to detect congestion in addition to a PostFacto congestion detection scheme that allows physical sinks to conclude about potential network overload. Siphon uses several virtual sinks to divert traffic and reduce congestion.

Light Weight Buffer Management: The fundamental principle of this approach is that a sensor node 'y' will only send a packet to a sensor node 'x' if sensor node 'x' has sufficient buffer capacity to hold the packet [83]. Therefore, nearby nodes are able to listen or receive their neighbor's buffer state. The neighboring nodes decide whether to transmit a new packet based on the buffer state. The linked neighbor nodes reduce their data rate when upstream nodes are congested. This process is iterative and ultimately achieves the highest throughput without congestion. This algorithm is compared with the CODA backpressure mechanism, global rate regulation, and no congestion control. However, the authors do not consider energy consumption

CoSMoS: The added difficulties brought on by a movable sink are addressed in this work [84]. To ensure dependable data delivery, enhancing the rate of path reconfigurations is first necassary. Effective load estimation methods must also be put into place since path reconfiguration may cause abrupt changes in load along the paths. Finally, temporary periods of poor path quality must be actively avoided. A plan called CoSMoS is based on combining routing and congestion control. The Cosmos scheme includes two components: the regional load collection method to determine each node's maximum sustainable load in a region and along a path, and a low-cost, low-complexity routing method considering the path's reliability variations throughout sink mobility. In Mica-2 motes, the CoSMoS algorithm has been implemented. According to experimental findings, it balances congestion and reliability to increase delivery ratios while maintaining throughput. The authors offer no energy results.

Buffer and Rate Control-based Congestion Avoidance: This protocol regulates the rate of upstream nodes using the first two techniques [85]. This protocol offers two benefits. The first is that upstream nodes proactively lower their rate, which reduces congestion caused by media access contention. The second is that upstream nodes minimize the congestion caused by buffer overflow. Explicit ACK is avoided in the third approach (Snoop-based MAC level ACK). Results show that this protocol enhances packet delivery ratio, decreases collision drop rate, and increases network energy efficiency.

CAEE CAEE introduces the idea of mobile sinks. The network is separated into groups known as mini-sinks [86]. The data collector node is the cluster head. A data collector node's primary duty is to accept and store the data acquired from the sensor field at the mini-sink. Each mini-sink in the sensor field is visited by the mobile sink regularly to retrieve data. The CAEE protocol can extend the network's lifetime because the packets only need to travel a short distance. The mini-sinks represent numerous collection points. Therefore, congestion hot-spots can be reduced concurrently.

TADR This method is suggested for distributing surplus packets through different channels of unoccupied and lightly loaded nodes to avoid congested locations [87]. The TADR algorithm creates a mixed potential field utilizing depth and normalized queue length. The TOSSIM simulation tool is used for simulation. Moreover, TADR has decreased overhead which makes it suitable for massive sensor networks.

ANAR: A congestion-free probability is computed using the NAV vectors [88]. This probability is used to find the most practical route for delivering packets during the route discovery process. It can switch between selected routes depending on how congested the network is at any given time. The results show that ANAR outperforms another algorithm in terms of power consumption and end-to-end delay

Priority-based Medium Access Protocol: In accordance with the source count value, this MAC protocol grants proportional access [89]. Each node uses the specified equation to compute its contention window. A MATLAB-based simulation claims that the ideal contention window size can effectively reduce MAC layer collision and assist in the seamless transmission of all packets.

TALONet: Three strategies are used by TALONet: buffer management to prevent buffer level congestion, variable transmission power level to relieve data link layer congestion, and multi-path detouring strategy to increase resources for congested traffic flows [90]. Three phases make up TALONet's operation. These include network development, data distribution, and framework upgrading phases. In the network construction phase, each node generates a virtual grid framework ‘G’ and determines the coordinates of the virtual grid points. The nodes that are near the cross points of the grid are referred as TALON nodes. The TALON nodes are in charge of gathering and relaying the sensing data during the data dissemination phase.

Flock-CC The key concept is to steer flocks of birds (packets) away from congested areas and toward the sink (global attractor) (obstacles) [91]. Forces of attraction and repulsion influence a flock of packets' motion. Flock-CC determines congestion using both buffer occupancy and wireless channel load. Each packet calculates the forces of attraction and repulsion to and from its immediate neighbors. The packet then flows through the network in an oriented manner by avoiding congestion areas. Flock-CC is easy to implement at the node level (each node follows some rules).

LVCC: The Lotka-Volterra-based congestion control (LVCC) protocol is proposed by Antoniou et al. [92]. LVCC uses The Lotka-Volterra population model to control the rate at which each traffic flow occurs to avoid congestion in WSNs. LVCC identifies congestion based on buffer occupancy. LVCC controls the traffic flow rate at each node to offer hop-by-hop rate adaptability. Each sensor node is intended to control the traffic flow so that all receiving sensor nodes leading to the sink have sufficient buffer space to hold all received packets. Each network flow is affected by changes in the buffer occupancy of nodes. Each node controls its data rate using local information. LVCC protocol is decentralized and easy to deploy at a single node. LVCC achieves scalability, fairness across flows, adaptation to shifting traffic loads, and scalability.

2.2.3 Reliable Data Transport Algorithms:

ESRT: It is an end-to-end protocol used to control the frequency of sensor reports [93]. The feedback that is received from the source nodes is used by the sink node to send back notifications to change the reporting rate with two objectives:

i) To get enough packets from sink nodes

ii) Only receive the necessary number of packets to conserve energy and reduce congestion.

First, it calculates the factual reliability based on packets that have been successfully received over a period of time. Second, ESRT determines the frequency of the sensor node report. Thirdly, ESRT transmits information to all sensor nodes over a high-power channel. According to ESRT, there are five distinct operation regions.

When reliability exceeds expectations, ESRT carefully reduces the reporting frequency. Due to its self-configuring characteristics, ESRT is robust in WSNs with unpredictable and dynamic topologies. Energy conservation is a further advantage of ESRT because it can adjust sensor reporting frequency. All nodes in ESRT are handled identically. When there is congestion in one network area, all nodes lower their data rates. This results in a negative impact on the throughput of the network. Additionally, ESRT cannot manage events demanding different levels of reliability.

Directed Diffusion Intanagonwiwat et al. present directed diffusion algorithm [94]. The nodes of a directed diffusion network are all application-aware. This makes it possible for diffusion to save energy by choosing analytically efficient paths (with little latency) and processing data within the network (e.g., data aggregation). The four basic parts are interest, data messages, gradients, and reinforcements. An interesting message is a request made to the network by a sink node that details what the application is looking for. It includes a description of a sensing task supported by a sensor network. When data regarding a particular event, such as a physical phenomenon, is acquired or processed, it is named (addressed) using attribute-value pairs, and a sensing job is distributed over the sensor network as an interest for named data. The network's gradients created by this dispersion are intended to "draw" events (i.e., data matching the interest). Sink node strengthens the empirically “excellent pathways” (e.g., small delay) and their data flow is increased to increase performance and reliability. The faulty pathways (such as those with a significant latency) are adversely reinforced and eliminated.

GARUDA: Garuda is proposed by Park et al. in [95]. GARUDA offers reliable point-to-multipoint delivery when sending data from the sink node to sensors. The Following are the components of GARUDA:

• A two-stage NACK-based recovery technique that effectively lowers retransmission process overheads and uses forwarding to use of the substantial spatial re-use potential in a WSN.

• A straightforward candidate-based solution that efficiently support the many reliability concepts needed in a WSN.

STCP: Iyer et al. present the STCP [96]. Most the features of STCP are implemented in the sink, making it a general-purpose, scalable, and stable transport layer protocol. Multiple applications are supported by STCP, which also offers extra features like regulated variable dependability and congestion detection. The Sensor nodes in STCP notify the sink using a "Session Initiation Packet" before transmitting packets. The sink node receives information about the type of data, the number of flows that have been started from a source, the transmission rate, and the required reliability through this packet. The sink sends an ACK packet to the source node as soon as it gets this packet, allowing the source node to begin sending packets. Sequence number, Flow id, clock field, and congestion notification bit are components of an STCP packet header. The sink node keeps a timer on and sends a NACK if it doesn't get a packet within the expected time. The percentage of successfully delivered packets is used to measure reliability. The sink node receives the notification about buffer overflow of sensor nodes by looking at their congestion notification bit. Similarly, when congestion occurs the sink sends a congestion notification by setting the congestion bit on ACK packet to the source nodes. To alleviate congestion, source node can modify the transmission rate or the routing path.

RCRT Paek at al. present RCRT [97-98]. RCRT emphases the secure transmission of sensor data from the source node to the sink node and prevents congestion failure. The transport layer is primary focus of RCRT. RCRT aims to provide 100% reliable data delivery based on a NACK system. Therefore, when packet loss occurs, the sink notifies the source of the missing packets by issuing a NACK with a number of missing packets. Congestion detection in RCRT depends on round trip time, rate allocation and rate adaptation. RCRT employs a rate adaptation approach to mitigate congestion.

Extended DCCP: The datagram congestion control protocol is extended by Liu et al. with a new protocol [99]. DCCP offers a uniform method of integrating congestion control and putting these discussions into multimedia applications. The following extra features are included with Extended DCCP:

- The receiver's ability to store received packets in a buffer
- Sender's retransmission of faulty or missing packets
- Duplicate packets are found and removed at the receivers.
- Distribution of received packets to receivers' application programs in the correct order, in this instance, there are four states for the sender: normal, congested, failure, and the Error State (e.g., transmission error). In general, extended DCCP can offer a reliable processing with a strong throughput.

2.3 Conclusion

WMSNs face several open issues. Network congestion is one of the most challenging open issues for WMSNs. Congestion occurs when the received data rate of a sensor node goes beyond its transmission data rate. Congestion can also occur due to interference and contention in the channel. Congestion results in severe packet loss, which is not required in WMSN applications. In WMSNs, it is critical to detect, notify, and control congestion in an efficient manner. Congestion can be detected either by buffer occupancy, wireless channel load, combined buffer occupancy and wireless channel load, or packet transmission time. To prevent congestion, it is implicitly or explicitly notified after congestion detection. After congestion notification, congestion is controlled by open-loop congestion or closed-loop congestion. This chapter presents a comprehensive analysis of congestion control algorithms in WMSNs. These congestion control algorithms are divided into three groups according to their advantages and disadvantages. Traffic-based congestion control algorithms, resource-based and hybrid congestion control algorithms.

Chapter 3

Energy-Efficient Distributed Congestion Control Protocol for Wireless Multimedia Sensor Networks

This chapter presents an energy- efficient DCCP to mitigate congestion and improve end-to-end delay. Compared to the other protocols, the DCCP protocol proposed in this chapter can alleviate congestion by intelligently selecting the best path. First, congestion is detected using two congestion indicators. Second, each node aggregates the received data and builds a traffic congestion map. The traffic congestion map is used to calculate the best path. Therefore, the traffic is balanced on different routes, which reduces the end-to-end delay. Finally, a rate controller is designed to prevent congestion in the network by sending a congestion notification message to a source node. After receiving a congestion notification message, the source node immediately adjusts its transmission rate. Experimental results based on Raspberry Pi sensor nodes show that the proposed DCCP protocol significantly improves network performance and is superior to modern congestion control protocols.

3.1 Introduction

WMSNs are one of the promising paradigms for the IoT [5]. WMSNs consist of wirelessly connected devices that can capture video data from the environment. Sensor nodes are equipped with inexpensive video cameras and microphones [7]. These camera nodes capture this visual information, process it and transmit it to the base station through multi-hop or single-hop communication. WMSNs are used in various applications. The scope of WMSN applications includes monitoring of indoor and outdoor environments [46], IoT-based smart health [31,33] and object tracking [12]. The transport layer [32,34], the network layer [23], and the media access control layer [89] have been extensively researched in WMSNs.

Due to many practical and theoretical challenges, WMSNs have attracted the attention of many researchers. Many challenges need to be addressed, such as harsh environmental conditions [11], node failures [40], mobility of nodes [14], mobility of detected events [49], dynamic network topology [39], heterogeneity of nodes [64], large scale deployments [18] and unattended operations [19]. In addition, there is tremendous progress in internet traffic applications that require bandwidth. Industrial forecasts predict that large- scale video traffic will dominate global internet traffic in the future [20].

Network congestion in WMSN networks is among the most challenging open issues [32]. Video data in WMSN networks always consumes much bandwidth. The sensor node generates new video data and transmits the traffic of other sensor nodes. Several nodes simultaneously send data to the sink node. Therefore, the sink node cannot process more data from the incoming stream. This causes network congestion. *C*ongestion decreases efficiency and reliability in WMSN networks. In WMSN networks, congestion occurs when the data transmitted over the network exceeds what is allowed or stipulated as the available capacity. Many factors cause congestion in WMSNs, such as buffer overflow, the dynamic nature of the channel, and man-to-one end-to-end connectivity. Congestion leads to node energy exhaustion, network performance degradation, high packet losses, and transmission delays. Therefore, it is of prime importance to design congestion control protocols to efficiently detect, notify and control congestion [34].

Congestion notification can be either implicit or explicit in nature as shown in Figure 3.1. In the explicit congestion notification process, the congested node reports the congestion status to other nodes using additional control packets. This technique is used in several congestion control mechanisms. However, these additional control packets need to carry a heavy traffic load unsuitable for WMSN networks. In Implicit Congestion Notification, this mechanism is used in most of the congestion control algorithms. This mechanism provides less traffic load compared to the explicit congestion notification of the mechanism. This mechanism does not introduce any additional overhead. Therefore, it is more suitable for WMSN networks [37].

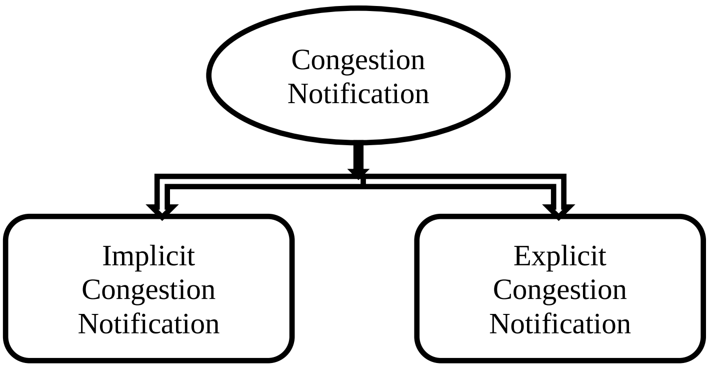

Figure 3.1 Congestion notification methods.

This chapter focuses on two main challenges. The first challenge is to control congestion in WMSNs. Another challenge is to improve the end-to-end delay of the wireless video transmission. In order to resolve these two challenges, an energy-efficient DCCP for WMSNs is developed. DCCP is a lightweight protocol that depends on UDP. As UDP is not reliable, DPCC uses its own congestion control algorithm to mitigate congestion efficiently. One of the limitations of the existing research is that most congestion control algorithms do not apply to WMSNs. The traditional Transmission Control Protocol (TCP) uses the Additive Increase and Multiplicative Decrease (AIMD) algorithmic technique. TCP continuously generates sawtooth behavior during congestion. The protocols that continuously represent sawtooth behavior are not suitable for video communication [96]. Because of these challenges, traditional protocols cannot be used for WMSNs. Therefore, designing a new congestion control system for WMSNs is challenging. Although traditional traffic control techniques that throttle the incoming traffic can effectively reduce congestion, these schemes do not increase end-to-end productivity. In addition, restricting important data packets during this period can nullify the purpose of WMSNs. Therefore, a robust congestion control algorithm for WMSN is urgently required [72].

Moreover, this chapter also focuses on applying a robust congestion control algorithm in WMSN networks to achieve reliable transmission source nodes to a base station through multi-hop communication. Although some work on congestion control has been performed in WMSNs, it is not focused on real-time solutions. These issues are addressed with the following contributions:

- The proposed DCCP algorithm improves the performance of various videos transmitted over the network by intelligently selecting the best path from source to destination
- A mechanism built on a dual buffer is proposed for congestion detection. This mechanism can effectively detect congestion.
- Each node aggregates the received data and builds a traffic congestion map. The traffic congestion map is used to calculate the best path. Therefore, the traffic is balanced on different routes, which reduces the end-to-end delay.
- DCCP prevents congestion in the network by sending a congestion notification message to the source node. After receiving a congestion notification message, the source node immediately adjusts its transmission rate.
- This research chapter demonstrates actual experiments using the Raspberry Pi Sensor Node. Each Raspberry Pi source node contains a unique IP address and establishes a route to the

parent node.

- Therefore, this research is more feasible than other algorithms that use computer simulations.

3.2 Related Work

The congestion control algorithms can be divided into three approaches: (1) traffic-based, (2) resource-based, and (3) hybrid-based congestion control approaches, as shown in Figure 3.2.

Traffic-based Congestion Control Protocols: These protocols are more widely distributed in end-to-end and hop-by-hop-based traffic control approaches.

The buffer occupancy-based transport layer protocol uses the buffer occupancy-based mechanism for congestion detection [97]. Two different thresholds are used to compute buffer occupancy at each node. If the buffer occupancy surpasses the upper limit, congestion is detected immediately, and child nodes slow down the traffic rate to reduce congestion. However, it cannot choose the optimal path. The AWF algorithm reduces network congestion by clustering two different algorithms [56]. The rate control mechanism is only used in the case of NACK. Moreover, an Ant Colony Routing-based mechanism is used to enhance the throughput in the network. However, acknowledgment in this algorithm increases the delay in the network. The TACC solves the rate adaptation problem in the transport layer of WMSNs [57]. TACC detects congestion using burst loss information and the presence of burst loss at the sink node. The sending rate is conservatively reduced if congestion occurs for the first time. TACC adjusts the sending rate aggressively if the network is continuously congested. However, TACC needs to be enhanced further to facilitate the conveyance of prioritized events to different flows.

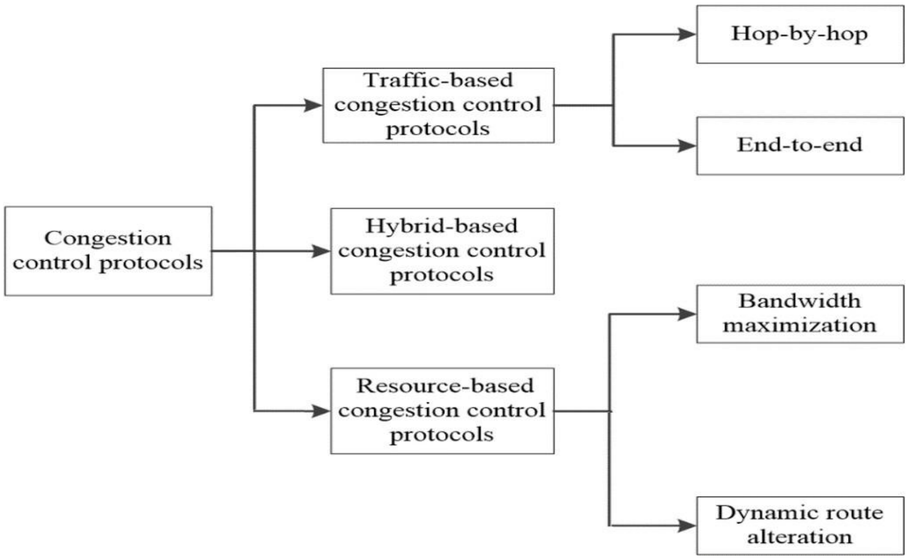

Figure 3.2 Congestion control protocols.

Resource-based Congestion Control Protocols: This method is particularly used in high-reliability applications that require lower time delays. It reduces congestion by balancing the traffic load either by utilizing idle network resources or using uncongested paths. Therefore, data packets have a higher probability of error-free communication. However, the resource control approach needs additional overhead in the form of end-to-end topology information, loop avoidance, and packet travel time for sensor nodes. Opt-ACM mitigates congestion using different routes for data routing with different QoS [58]. The Mixed-Integer Linear Programming mechanism is used to verify the proposed algorithm. However, the proposed algorithm does not focus on energy efficiency in the network. A routing technique based on DRL minimizes the end-to-end delay using the unequal clustering scheme [59]. The proposed algorithm prevents the network by splitting the entire network into unequal clusters. However, the complex methodology is adapted to use unequal clustering. The TMR algorithm substantially enhances the network capacity by choosing an alternative path [60]. TMR maintains a unique ID and mitigates congestion by using a reactive technique. TMR assigns higher priority to alert messages to send traffic to different routes that are not congested. The SLEB protocol combines load balancing and security verification mechanisms based on clustered WSNs [61]. This protocol is effective in balancing the energy of the network. Moreover, it also enhances the security overhead. Therefore, SLEB can prolong the network lifetime. SLEB has a much lower overhead. However, in SLEB, multiple sensor nodes can send the same data to the receiver node. DHSSRP prioritizes significant data to mitigate congestion in the network. In DHSSRP, the authors assign weights to high-priority packets. They assume that traffic in different sensor nodes varies considerably due to differences in packet priorities. DHSSRP is a lightweight protocol that increases the lifespan of the deployed sensor nodes by diverting the data traffic to alternative routes [101].

Hybrid Congestion Control Protocols: These protocols use a combination of both resource-based and traffic-dependent congestion control protocols. HOCA uses active queue management to control congestion. Sensitive traffic needs a high data rate, whereas non-sensitive traffic needs a low data rate. Their protocol comprises four phases. In the first phase, the sink performs data transmission to all source nodes. This stage determines the node type, timing, and data priority. In the second phase, the sink receives data from each node. In the third phase, the sink node determines the path using hop-by-hop communication. In the fourth phase, the data rate is adjusted to reduce congestion. HOCA achieves better energy efficiency and ensures suitable allocation of resources [63].

Traffic-based Congestion Management (TCM) controls congestion using a hybrid congestion control approach [65]. When congestion occurs, the congested sensor nodes report to the source to reduce the data rate using multi-hop communication. When a congested downstream node receives the back-pressure information, the TCM tries to use the resource control technique by changing the path. If an alternative route path is not available, it uses traffic-based congestion control protocols by reducing the data rate of the source node.

ECA-HA controls congestion using a hybrid congestion control approach [66]. ECA-HA reduces congestion using three phases. In the first phase, the ant colony optimization method is used to select an optimal route. In the second step, the backward ant makes sure that paths are constructed successfully, and the forward ant creates congestion-free paths via multi-hop communication. Finally, congestion is controlled by defining an alternative route if rate modification is not feasible. However, the ECA-HA is only suitable for small search space

The techniques mentioned above have various flaws, loopholes, and deficiencies; therefore, those strategies cannot be regarded as prodigious. To overcome that vacuum and weaknesses, the proposed technique can fill up the gap to mitigate the congestion and enhance the performance of the network.

3.3 Distributed Congestion Control Protocol for Wireless Multimedia Sensor Networks

3.3.1. Description of WMSN System

Figure 3.3 presents the WMSNs and the architecture of the node, stating that each node has a sensing unit, a processing unit and a transmission unit. The video camera represents a sensing unit, Raspberry Pi is a mini-computer with a powerful processing unit and Wi-Fi is a communication unit. The topology used in a network is a tree topology with a single sink node and twelve (12) source nodes. These sensor nodes are used to generate and relay traffic. The sink node receives the recorded video data through multi-hop communication.

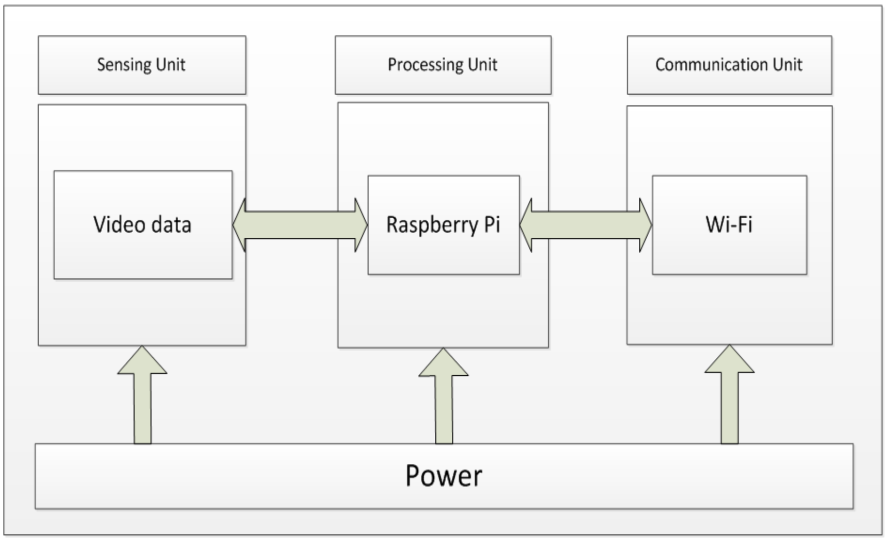

Figure 3.3 Sensor node architecture.

3.3.2. System Model

In the proposed algorithm, each node contains two types of data packets:

Generated packets: The node senses the environment and generates the data packets. These packets are called generated packets.

Relay Packets**:** In multi-hop communication, the sensor node receives packets from neighboring sensor nodes. These packets received from neighboring nodes are called relay packets. Table 1 shows a description of the parameters.

$$\lambda_i = \lambda_{ii} + \lambda_{i,relay} = \lambda_{ii} + \sum_{j \epsilon N_i^{in}} \lambda_{ji} \tag{3.1}$$

$$B_i(t+1) = B_i(t) + \lambda_i - \sigma_i \tag{3.2}$$

Equation (3.3) shows the threshold value of i

$$\alpha_i(t+1) = \begin{cases} \dfrac{\sigma_i(t)}{\lambda_i(t)}[B_{max} - B_i(t)] \\ \lambda_i(t) > 0,\ \sigma_i(t) > 0, and\ u_i(t) > v_i(t) \end{cases} \tag{3.3}$$

Table 3.1 Description of parameters.

Parameter	Description
λ_i	input rate of a node
σ_i	output rate of a node
λ_{ii}	generated packets
λ_{ji}	relay packets
$B_i(t)$	buffer occupancy at time (t).
$B_i(t+1)$	buffer occupancy at time (t + 1).
$\alpha_i(t+1)$	preferred queue level at $(t+1)$
B_{max}	Maximum buffer size
P_i	probability of node i
p_{ij}	probability of receiving a packet from another node
N_i	neighborhood nodes of node i
Q_i	probability that a node is adjacent to node i

The following mechanism is used to detect congestion:

1. Congestion occurs if $B_i(t)$ reaches the maximum value of B_{max} and $\lambda_i(t) > \sigma_i(t)$.

2. The node is not congested if $B_i(t) < \alpha_i(t)$.

3. The proposed algorithm detects congestion when $\alpha_i(t) < B_i(t) < B_{max}$.

Let $G(t)$ be the probability of arrival of one packet. Then,

$$G(t+1) = g(t+1) \tag{3.4}$$

$$G(t+2) = g(t+2) + g(t+1|t+1) \tag{3.5}$$

Similarly,

$$G(t+k) = g(t+k) + g(t+1|t+(k-1)) \ldots + g\,(t+(k-1)\,|t+1) \tag{3.6}$$

In general,

$$G(t+k) = \sum_{n=1}^{k-1} g\big(t+n\big|t+(k-n)\big) + g(t+k) \tag{3.7}$$

where $k = 2,3 \ldots\ldots ,(B_{max})$

Let p_{ij} is the probability of receiving a packet from another node and is given by

$$p_{ij} = \frac{1}{\mu} p_i \tag{3.8}$$

The probability of neighborhood $'\mu'$ is defined by:

$$G_i^{\mu} = g(|N_i| = \mu) = \binom{n-1}{n-1} - (1-Q_i)^{n-\mu} Q_i^{\mu-1} \tag{3.9}$$

The expected packet receiving probability of node i which is based on [36] is denoted by equation (3.6).

$$E[P(i)] = \sum_{\mu=1}^{n} p_{ij} P_i^{\mu} = \frac{p_i}{nQ_i}(1-(1-Q_i)^n) \tag{3.10}$$

The level of congestion in a node is computed using $\alpha_i(t)$ and equation (3.3).

3.3.3. Congestion Detection

In DPCC, the buffer occupancy change rate and buffer occupancy are used to handle congestion at each node in a proficiently. Let B be the total buffer occupancy and δ be the buffer occupancy change rate. The total occupancy of buffer B is categorized into two different levels of congestion. Let B_1 and B_2 be the two congestion levels, then the buffer occupancy change rate is given by Equation (3.11)

$$\delta = \frac{B(t+\Delta t) - B(t)}{\Delta t} \tag{3.11}$$

where buffer occupancy in the last round is given by $B(t+\Delta t)$ and in the current round is given by $B(t)$.

Greater δ represents the higher buffer occupancy.

Furthermore, the sensor nodes are further classified into three different states.

Normal state: Buffer occupancy varies between [0, B_1] and the buffer occupancy change rate is less than ρ. P is the pre-defined threshold level of δ. In the normal state, all packets are transmitted successfully.

Slow state: Buffer occupancy varies between $[B_1, B_2]$. The buffer occupancy grows sharply until the value of δ goes beyond the level of ρ. The slow state shows that sensor nodes will shortly face congestion.

Urgent State: In an urgent state, buffer occupancy varies between $[B_2, B_{max}]$ irrespective of δ. A source node faces congestion during an urgent state. Therefore, sensor nodes face heavy packet loss. Source nodes should decrease their transmission rates straightaway.

3.3.4. Traffic Congestion Level Dissemination

Nodes send their congestion estimation to other nodes because it is important to obtain data from other nodes that are out of communication to calculate the best path. Therefore, proposed DCCP has two methods for data sharing. The first method tends to disseminate congestion information to nodes that are in the one-hop range. The nodes include a dataset in their periodical beacon named local congestion estimation. The local congestion estimation has the congestion estimations of the last η paths covered by the node. The congestion record of only the current and previous paths is considered.

The second method disseminates the congestion notification to two or more hops. Each node has a fixed path section of maximum distance equal to the communication range of the node. Moreover, the node monitors its nearest hop location through the received beacons. First, the node finds its parent from a list of nodes. If a node has to choose its parent node among other nodes, then the nearest neighbor in its range can be chosen as its parent. Whenever a new entry is added into the database of the node, the information is aggregated into the database All congestion level estimations are used for given paths to compute the aggregated data. Moreover, higher weights are assigned to the most recent estimation

3.3.5. Distributed Rate Control

Let γ represent the congestion level. The congestion reduction module is used when the

γ is modified. The nodes use the modified γ to calculate the optimum paths independently. Every node has an initial point and a final point.

The DCCP mainly avoids the formation of bottleneck routes for two reasons. First, each node has a different γ, so different weights are allocated to different nodes. Second, the nodes always find their best routes. When the congestion level in a path increases, other nodes will not choose that path. Therefore, the traffic on different routes is balanced.

Each node establishes a route to its parent node and adjusts its transmission rate according to the degree of congestion of neighbor nodes. DCCP examines the buffer occupancy at each node separately. Later on, DCCP transmits the buffer occupancy information to its children nodes through a feedback mechanism. Furthermore, if congestion still occurs, source nodes receive feedback information, evaluate it, and handle adjustments accordingly.

Let β show the dropping ratio, S_n represents the reduced transmission rate, S_N shows the normal transmission rate of the respective nodes during congestion. Then, the transmission rate at node n is given by $S_n = \beta S_N$. The reduced transmission rate, the normal transmission rate and β can be calculated from experiments. Algorithm 3.1 shows the pseudocode of the proposed algorithm, with the following parameters.

Definition of Parameters:

- Total Buffer Size: B_{max} Current Buffer Occupancy: B_C
- B_1: first threshold of buffer: (1/3) * B_{max}
- B_2: Second threshold of buffer: (2/3) * B_{max}
- Buffer Occupancy in the current round: $B(t + \Delta t)$
- Buffer Occupancy in the last round: $B(t)$
- Buffer Occupancy Change rate: $\delta = \frac{B(t+\Delta t)-B(t)}{\Delta t}$
- Topology: Tree Topology

Algorithm 3.1: DPCC Algorithm for Tree Topology

Pseudocode:

Input:

- Buffer Occupancy *B*,
- Buffer occupancy change rate δ,
- Tree topology.

Output:

- Congestion Level detection
- Transmission Rate adjustment.

Initialization:

- Set total buffer size \B_{max}\
- 1st threshold B1 = (1/3) * B_{max}
- 2nd threshold B2 = (2/3) * B_{max}

Steps:

1: Compute current buffer occupancy and occupancy change rate
2: if $B_C > B_2$ and $\delta > \rho$
  (a) Mark node as congested
  (b) Adjust local data transmission rate accordingly
3: else if BC $< B_2$ and $\delta > \rho$
  (a) A sensor node is near to congestion
4: else if BC $< B_1$ and $\delta < \rho$
  (a) A node is not congested
5: end

3.4. Experimental Section

This section shows the performance of DCCP. Thirteen Raspberry Pi sensor nodes are used in experiments. The sink node receives recorded video data from each source node through multi-hop communication. Figure 3.4 presents the practical implementation at the sink node. Figure 3.5 illustrates the network topology used in experiments. Table 3.2 shows a description of the parameters.

Table 3.2 Simulation parameters.

Parameter	Value
Number of nodes	13
Simulation time	1800 sec
Routing protocol	AODV
Antenna	Omni-directional
Data packet size	1048 bytes
MAC Protocol	IEEE 802.11
Topology	Tree based topology
Frequency	2.4 GHz

Five parameters are used to find the efficiency of DCCP. These five parameters are buffer occupancy, packet delivery ratio, average end-to-end delay, energy consumption and throughput. The following algorithms are compared in this section:

(1) DCCP: This algorithm is executed in this chapter.

(2) IVSP: This algorithm is given in [43].

(3) PPI: This algorithm is given in [100].

(4) NoDCCP: No congestion control mechanism is used in NoDCCP.

Figure 3.4 Practical implementation at the sink node.

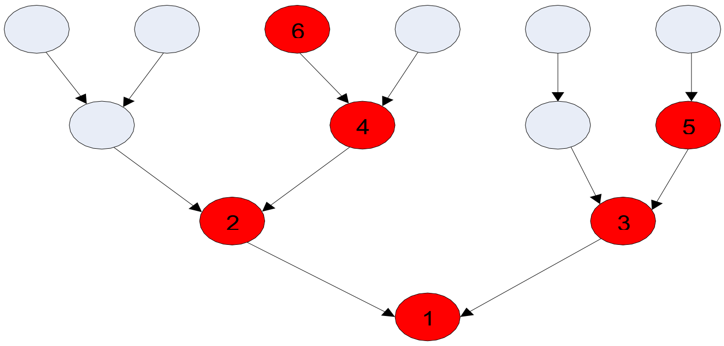

Figure 3.5 Routing topology used in DCCP.

3.4.1. Buffer Occupancy Analysis

Figures 3.6–3.9 show the buffer occupancy for NoDCCP. The buffer occupancy plays a pivotal role in the performance of the network. Figure 6 shows that only node 1 is congested and the rest are not at 0.1 Mbps. Figure 3.6 shows that the buffer occupancy at node 1 in NoDCCP immediately begins to overflow because the incoming rate exceeds the outgoing capacity of the buffer. For other nodes, the incoming rate is less than the outgoing capacity of the buffer. Therefore, different nodes are not congested in Figure 3.6. Similarly, Figure 3.7 shows that node 1 and node 2 get congested but the remaining nodes do not experience any congestion at 0.2 Mbps. Figure 3.8 shows that node 1 to node 3 gets congested by increasing the data rate to 0.5 Mbps. Figure 3.9 shows that congestion occurs at each node except node 6. Node 6 does not experience any congestion because data is generated only at node 6 and does not relay any traffic. So, the chances of congestion are minimal at node 6.

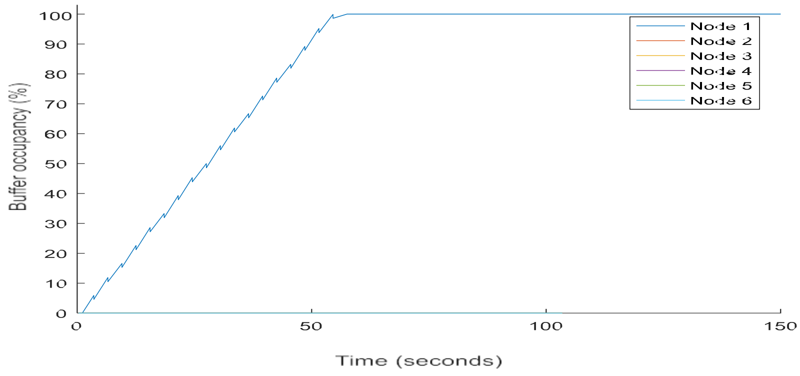

Figure 3.6 Buffer occupancy with NoDCCP at 0.1 Mbps.

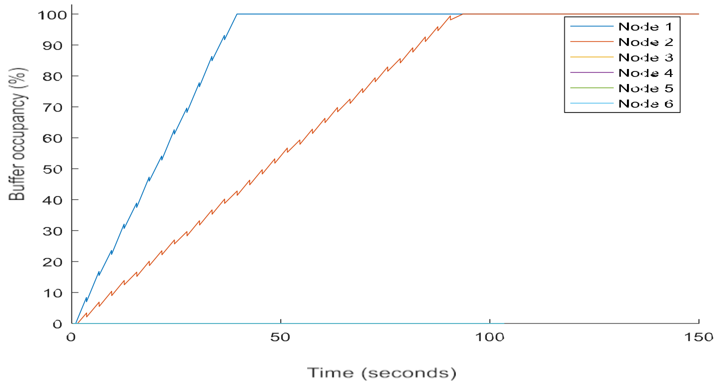

Figure 3.7 Buffer occupancy with NoDCCP at 0.2 Mbps.

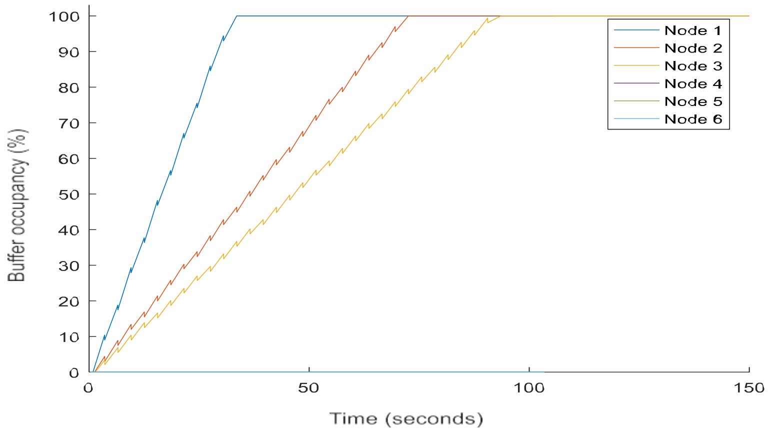

Figure 3.8 Buffer occupancy with NoDCCP at 0.3 Mbps.

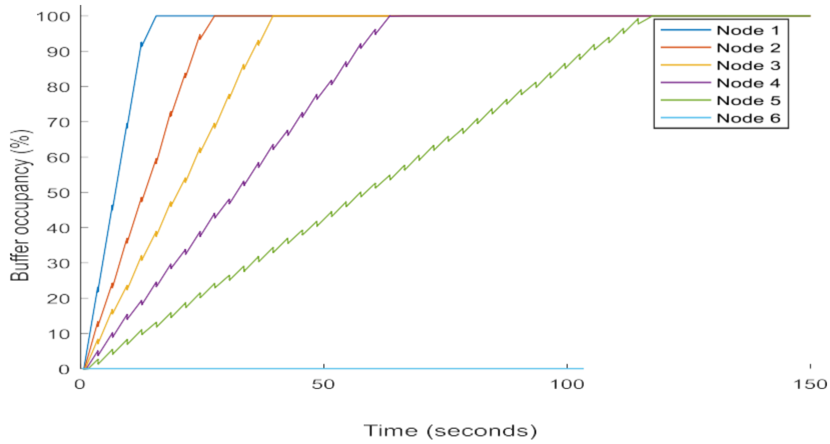

Figure 3.9 Buffer occupancy with NoDCCP at 0.5 Mbps.

Figures 3.10 and 3.11 show the improvement made with DCCP. DCCP adjusts the buffer occupancy according to the network condition. The buffer occupancy over-shoots first and then falls down. DCCP adjusts its transmission rate according to the degree of congestion of neighbor nodes. Therefore, DCCP achieves a superior congestion-free rate. Figures 6–9 show that the buffer occupancy in NoDCCP immediately begins to overflow because of the lack of congestion control mechanism.

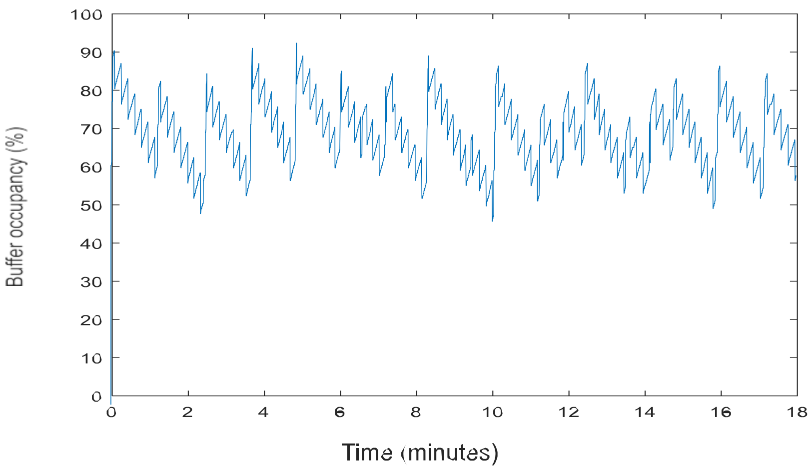

Figure 3.10 Buffer occupancy with DCCP at 0.2 Mbps.

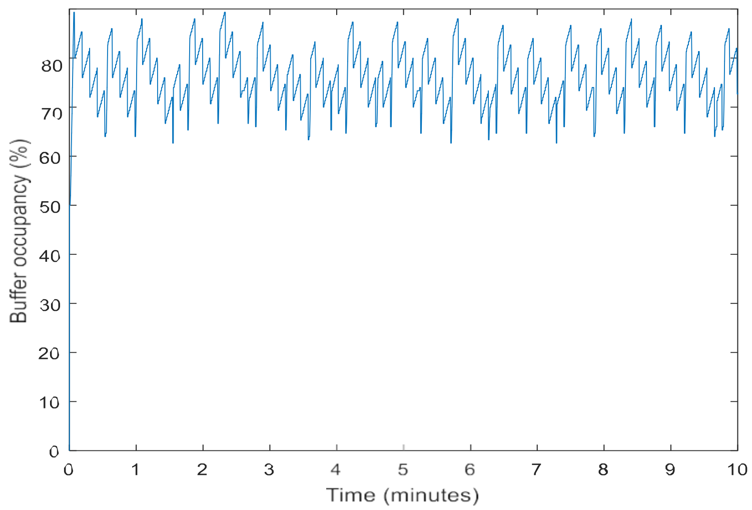

Figure 3.11 Buffer occupancy with DCCP at 0.5 Mbps.

Figures 3.10 and 3.11 show the variation of β. A lower beta results in a lower data injection rate, indicating that congestion is unlikely. A higher beta results in a higher data injection rate, which indicates that congestion will occur soon. Figure 3.10 shows that for a small value of β, buffer occupancy first comes to the normal state and then returns to the urgent state. Figure 3.11 shows that for high values of β, buffer occupancy does not go to the normal state and always varies between the urgent and the slow state. Therefore, the beta should be chosen intelligently.

3.4.2. Packet Delivery Ratio

Figure 3.12 represents the relationship between the packet delivery ratio and data rate. The packet delivery ratio decreases with the increase in data rate. The proposed DCCP algorithm has the best packet delivery ratio. Figure 3.12 shows that DCCP receives an 83% packet delivery ratio. Whereas, PPI, IVSP and NoDCCP achieve 79%, 77%, and 52% packet delivery ratio, respectively, at a data rate of 30 packets per second. IVSP only protects high-priority packets and eliminates low-priority packets in a congestion situation. Therefore, IVSP has a lower packet delivery ratio than DCCP. IVSP and PPI protocols show better performance than NoDCCP. NoDCCP has no congestion control mechanism therefore it shows insignificant performance than the other algorithms.

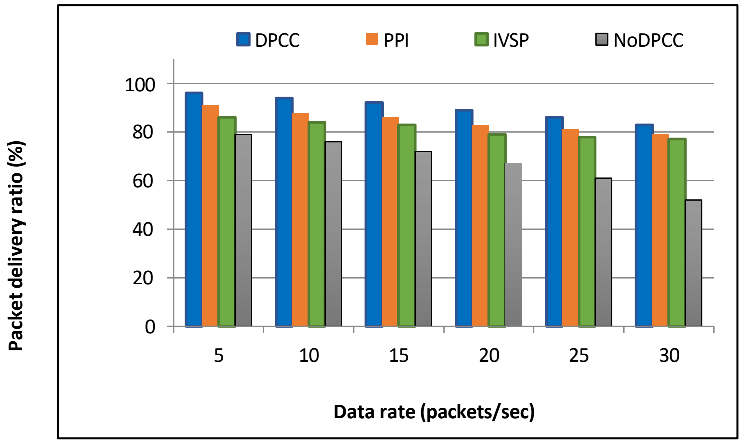

Figure 3.12 Packet delivery ratio vs data rate.

3.4.3. Average End-to-End Delay

Figure 3.13 represents the relationship between the average end-to-end delay and data rate. It is observed that a smaller data rate causes negligible delay. Figure 3.13 shows that the average end-to-end delay increases with increased data rate. Higher average end-to-end delay causes network saturation. Compared to other protocols, DCCP has the smallest average end-to-end delay. Figure 3.13 also illustrates that NoDCCP has the highest average end-to-end delay. DCCP shows superior performance than all the other algorithms. DCCP obtains 200 ms end-to-end delay. Meanwhile, PPI, IVSP and NODCCP have end-to-end delays of 220 ms, 250 ms, and 330 ms, respectively, at 30 packets per second.

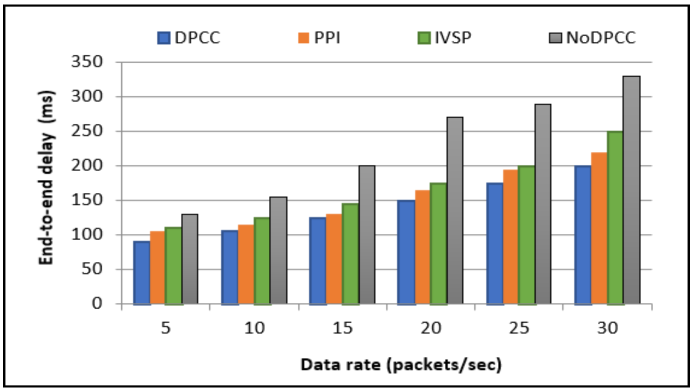

Figure 3.13 Average end-to-end delay vs data rate.

3.4.4. Energy Consumption

Energy consumption is one of the most challenging open issues in WMSNs. The purpose of DPCC is to balance the traffic by reducing the energy consumed by packets traveling from the source node to the sink node. Figure 3.14 shows the energy consumption at the sink node. The energy consumption of the proposed DPCC is lower than that of the other compared algorithms. The incoming rate is not reduced in the case of proposed DPCC. The reception and transmission numbers are higher in NoDPCC, IVSP, and PPI. Therefore, DPCC consumes less energy than other protocols. It is observed that the DPCC algorithm saves 9.09%, 14.28%, and 33.33% of energy when compared with PPI, IVSP, and NoDPCC at a data rate of 15 Kbps. NoDPCC consumes maximum energy because the buffer was not able to utilize all the packets.

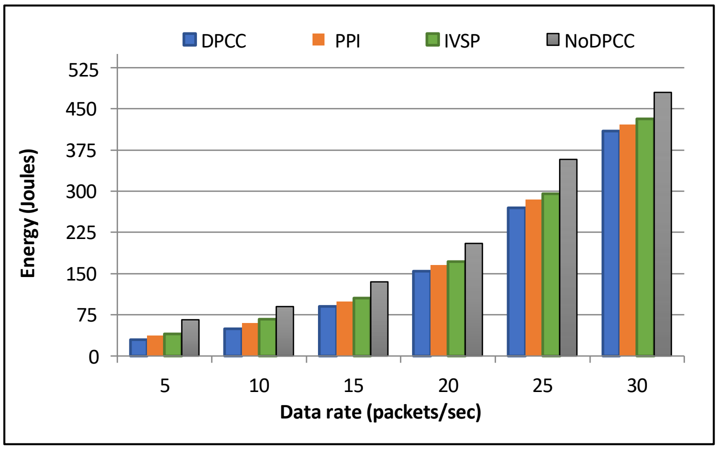

Figure 3.14 Energy consumption vs data rate.

Figure 3.15 shows the energy consumption at the node level. Four intermediate nodes from Figure 3.5 are chosen for this purpose. Node 4 receives the data from just one node therefore it has minimum energy consumption. Node 3 receives the data from four other nodes. Therefore, node 3 has more energy consumption than node 4. Node 2 receives the data from six different nodes. Therefore, node 2 consumes more energy than node 3 and

node 4. Similarly, node 1 receives data from all other nodes. Therefore, energy consumption at node 1 is maximum.

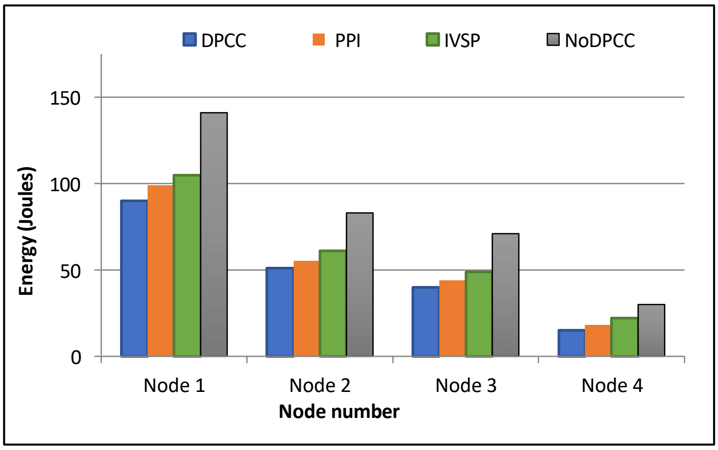

Figure 3.15 Energy consumption at each node.

3.4.5. Throughput

Figure 3.16 represents the relationship between the throughput and data rate. Throughput is also of prime importance in WMSNs. Sensor nodes that are close to the sink receive a heavy traffic load. Initially, throughput increases with data rate but stabilizes at a certain point due to congestion or latency. Therefore, the throughput value has to be improved for better WMSN performance. NoDCCP has no congestion control mechanism. DCCP has more throughput than other protocols because it buffers the data in a congestion situation and covers the whole network segment. Therefore, DCCP makes more efficient use of the available bandwidth. On the other hand, NoDCCP throws packets simultaneously without knowing the buffer condition which results in congestion in the network. DCCP shows better results than PPI, IVSP and NoDPCC. DCCP achieves 310 Kbps throughput. While PPI, IVSP and NoDCCP has throughputs of 270 Kbps, 230 Kbps, and 180 Kbps, respectively, at 30 packets per second.

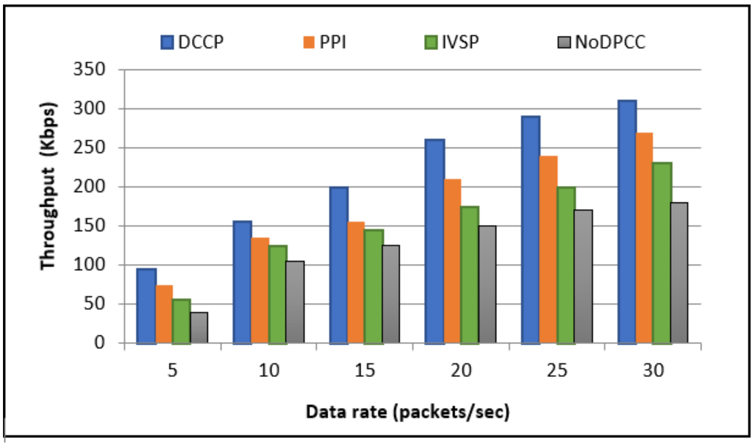

Figure 3.16 Throughput vs data rate.

3.5. Conclusion

This chapter presents an energy-efficient DCCP for WMSN. The proposed DCCP protocol improves the performance of numerous videos sent over a network. In DCCP, buffer occupancy change rate and buffer occupancy are used to deal with congestion at each node. The traffic congestion map is used to calculate the best path. Therefore, the traffic is balanced on different routes, which reduces the end-to-end delay. Furthermore, each node establishes a route to its parent node, retains its transmission rate and adjusts its transmission rate according to the degree of congestion of neighbor nodes. The experiment results show that the proposed DCCP performs better in terms of buffer occupancy, packet delivery ratio, end-to- end delay and throughput when compared with other well-known algorithms. Regarding future work, the experiment results implemented in this chapter will be used as a base to achieve further improvements in the proposed DCCP using more complex topologies.

Chapter 4

Buffer Occupancy Based Congestion Control Protocol for Wireless Multimedia Sensor Networks

WMSNs have stringent constraints and need to deliver data packets to the sink node within a predefined limited time. However, due to congestion, buffer overflow occurs and leads to the degradation of the QoS parameters of event information. Congestion in WMSNs results in exhausted node energy, degraded network performance, increased transmission delays, and high packet loss. Congestion occurs when the volume of data trying to pass through a network exceeds its capacity. First, the BOCC protocol uses two congestion indicators to detect congestion. One is the buffer occupancy and other is the buffer occupancy change rate. Second, a rate controller is proposed to protect high- priority I-frame packets during congestion. BOCC sends a congestion notification to the source node to reduce congestion in the network. After receiving the congestion notification message, the source node adjusts its data transmission rate. In the proposed algorithm, the rate adjustment is made by discarding Low-priority P-frame packets from the source nodes. Third, to further improve the performance of the BOCC protocol, the problem is formulated as a constrained optimization problem and solved using convex optimization and sequential quadratic programming (SQP) methods. Experimental results based on Raspberry Pi sensor nodes show that the BOCC protocol achieves up to 16% reduction in packet loss and up to 23% reduction in average end-to-end delay compared to state-of-the-art congestion control algorithms.

4.1. INTRODUCTION

The advancement of technology has led to the development of low-power tiny devices which integrate sensing, processing, and communication abilities. Networks of such devices with wireless communication capabilities are called WSNs [2]; these have wide spread applications. They contain many sensor nodes, connected wirelessly, to sense and process scalar data and communicate with the sink node. The number of nodes present in WSNs maybe hundreds or thousands, is scattered throughout the entire network. Every sensor node is basically comprising components for sensing, processing, transmission, reception, and power supply. he development of sensor-based micro-electromechanical systems (MEMSs) along with inexpensive complementary metal–oxide–semiconductor (CMOS) cameras integrated with microphones have led to the new era known as WMSNs [3]. WMSNs collect or capture streams of data, process them, store them for future requirements, and aggregate multimedia data collected from different nodes for onward transmission. They transmit video, audio, text, and images over wireless communication media, and in order to achieve this, the nodes can gather audio and visual data to create scalar and multimedia data. WMSNs expand WSNs' use to applications like tracking [12], environmental monitoring [46], automation [34], multimedia surveillance [13], traffic control systems [43], healthcare facilities [50], habitat monitoring [46], and industrial control systems [49].

Existing congestion control protocols for WMSNs primarily focus on either traffic-based or resource-based mechanisms, with some hybrid approaches attempting to balance both. However, these protocols often face limitations in high-density networks where buffer overflow and packet loss compromise QoS due to insufficient prioritization of critical data types and ineffective congestion detection. Traditional traffic-based methods are reactive, adjusting rates post congestion, while resource-based methods can over-utilize network resources, impacting longevity. Hybrid methods partially address these concerns but lack robust prioritization mechanisms for multimedia data, which is essential for applications requiring real-time video and image transmission. The requirement of QoS in multimedia communications introduces a great challenge because of high bandwidth requirements, unreliable channels, and restricted energy levels [12]. Usually, the total load at any node is calculated by computing the total packets generated and relayed over the node by the neighbor nodes. Because of the high density of nodes, the sensing range of nodes overlaps, hence there is redundancy in event detection by the nodes. This leads to generation of higher traffic and in turn increases the rate of input at sensor nodes. The output rate decreases due to simultaneous transmission attempts by the nodes. Hence, the input rate becomes more than the output rate and data packets accumulate in the buffer causing congestion at a node, which leads to buffer overflow and packet drops. Congestion at one node or part of the network spreads throughout the network and hampers the overall performance. This leads to packet loss and also degradation of network throughput [13]. Event tracking is also affected because the detected event characteristics cannot be reliably delivered towards the sink node. Congestion also adversely increases energy consumption at nodes and partitioning of the network [14]. Congestion is the accumulation of too many packets, which decreases the efficiency of the network due to insufficient network capacity. In other words, congestion is the process of building up of packets in the queue or buffer of the nodes, and this happens when the number of data packets at the receiving side is less than the number of data packets at the transmitting side. A typical congestion scenario is depicted in Figure 4.1. It forces the victimized nodes to ignore the incoming data packets due to lack of buffer space at that node, and drops the data packets. Thus, it requires data packet retransmission, causing more energy consumption at the nodes [15]. Retransmission of data packets leads to more energy consumption, because the transmission process consumes the highest amount of energy out of all the activities performed by a node. Furthermore, the injection of high-rate multimedia data packets in the network causes resources like energy and buffer space to be consumed quickly. Moreover, different traffic types flowing in the network with various characteristics along with QoS requirements need to support the transmission of bursty data with a high data rate. So, designing congestion control protocols for WMSNs with optimum efficiency is a challenging task. This signifies the importance of addressing the challenge of reducing the congestion for better energy efficiency and QoS for WMSNs [16].

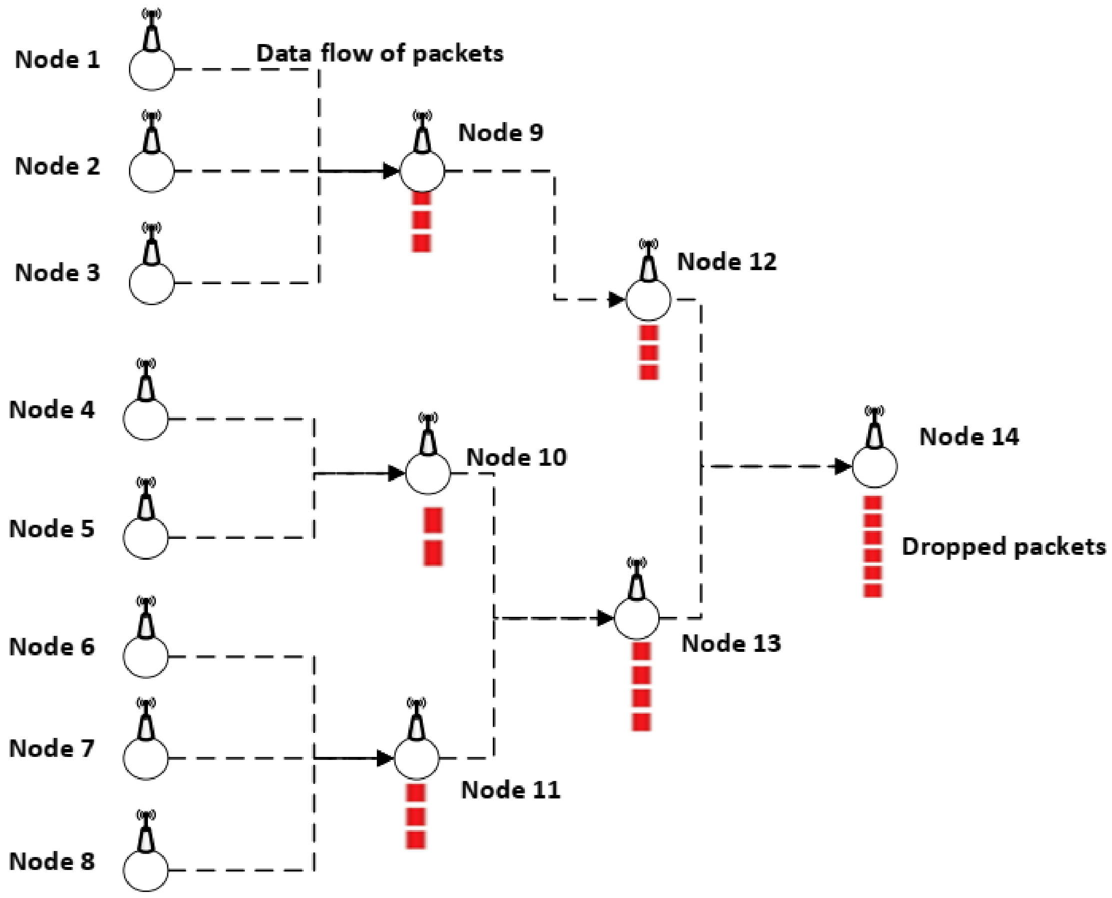

Figure 4.1 Packet flow in wireless multimedia sensor networks

The first solution is to adopt a centralized routing algorithm that calculates the routing decision for a given sensor node and distributes it to the sensor nodes. When the topology changes, route needs to be re-computed and distributed among the sensor nodes. Hence, because of this sort of redundancy, centralized routing algorithms are inappropriate for WSNs and WMSNs in particular [17]. In the case of distributed schemes, routing algorithms distribute the traffic load among the neighbor nodes evenly to make sure that every node in the network has enough buffer space to enqueue packets. In WMSNs, the event occurs randomly, ; hence, the traffic load and the available buffer at sensor nodes in the neighborhood of nodes are dynamic. This dynamism leads to unevenness of load during routing. So, distributing the load evenly without the support of the information of other layers (cross-layer) may result in non-uniform distribution of information on the level of the buffer in the neighborhood [18].

The BOCC protocol addresses these gaps by introducing a proactive congestion control method based on buffer occupancy metrics, which efficiently manages data transmission rates before severe congestion occurs. Unlike existing protocols, BOCC prioritizes high-priority I-frame packets over P-frame packets, ensuring that essential multimedia data reaches its. This strategy optimizes packet delivery by discarding lower-priority packets selectively, minimizing packet loss without exhausting network resources. Additionally, convex optimization and sequential quadratic programming (SQP) enable BOCC to dynamically adapt to network changes, enhancing packet delivery rates and reducing delays compared to traditional approaches. Above all, the key objectives achieved in this chapter are listed below.

- The proposed BOCC algorithm retains the high-priority I-frames, discarding further low-priority frames in scenarios where congestion occurs, hence improving the performance of different media sent over the network.
- A dual-buffer-based congestion control mechanism is modeled to predict the congestion in WMSNs. Congestion can be effectively detected by this approach.
- Moreover, to further optimize the performance of the BOCC protocol, the problem is mathematically modeled as a constrained optimization problem. Convex optimization

and SQP methods are further mathematically modeled to solve the optimization problem. Convex optimisation is observed to be better than the SQP method as it converges in less time.

- This chapter represents practical experiments conducted with a Raspberry Pi sensor node.

4.2. BUFFER OCCUPANCY BASED CONGESTION CONTROL ALGORITHM

In this section, the underlying principles and structure of the proposed BOCC algorithm is discussed. This algorithm is specifically designed to handle congestion in WMSNs by monitoring buffer occupancy levels and adjusting transmission rates based on congestion indicators. The following subsections detail video packet formation and the mathematical formulation of the BOCC algorithm.

4.2.1 The formation of video packets:

H.264 is the compression standard widely used to compress video because of its efficiency in reducing the bit rate, needed for video transmission. This standard ensures the quality of the video is maintained during transmission. The H.264 standard achieves this by exploiting frames of different types. Figure 4.2 represents the queue scheduler block diagram.

- **I-frames (intra-coded frames):** These are the main entities in video streaming and are considered reference points for other frames. I-frames are encoded separately, and contain all the necessary information to show the image. I-frames are commonly larger in size due to the addition of essential information for better quality. I-frames are given high priority to obtain better video quality. If an image with I-frames is lost, then the quality of video will be compromised as all other frames including P-frames rely on the I-frames.
- **P-frames (predictive-coded frames):** P-frames are encoded based on the difference between the current frame and the preceding I-frame or another P-frame. This predictive

coding allows P-frames to be smaller in size than I-frames, making them more efficient in bandwidth utilization. The loss of P-frames leads to error propagation as they depend on previous frames, affecting subsequent frames until the next I-frame is received. The P-frames are given lower priority than I-frames but are very important to maintain the overall video quality.

- **B-frames (bi-predictive frames):** B-frames are compressed frames in comparison with P-frames. They also use the subsequent frames as references. Due to their higher compression rates, additional complexity and delay are added in the encoding and decoding processes. The coding delay for B-frames is high because the encoder has to wait for the following frames to generate them. B-frames are ignored in the scenario where low latency is required due to their increased complexity and high delay factors. In this case, B-frames are excluded to reduce the coding delay and to simplify the transmission process.

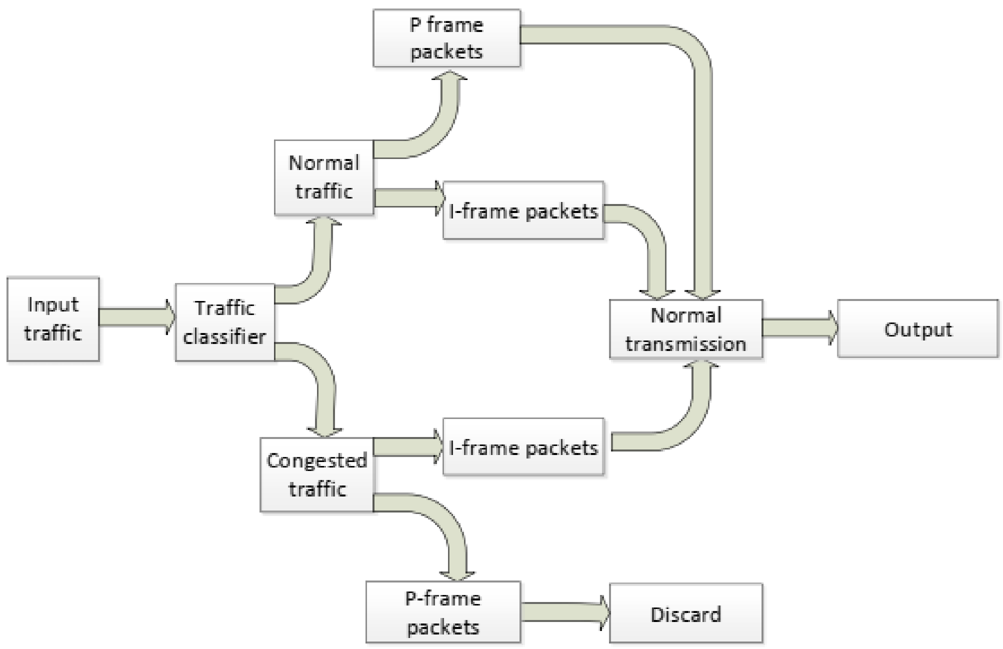

Figure 4.2 Queue Scheduler

Packetization is an important challenge that needs to be addressed in order to determine the efficiency and reliability of video streaming. Packetization is the process in which the video stream is divided into smaller data packets that are transmitted over networks. Factors including packet loss, latency, and network congestion depend on the length of these packets. Moreover, the encapsulation process is only completed when all the relevant frames are received and the packets to be encapsulated wait for all the frames for the video streams containing different frames. More delay is induced if the packet length is not optimized. The packet length should be optimized by defining threshold limits to cope with this challenge.

Smaller packets result in a more packets being transmitted, which can lead to increased overhead headers and reassembly at the receiving end. Moreover, selecting the appropriate packet length involves a trade-off between reducing the likelihood of packet loss and minimizing packetization overhead. If packets are too large, the risk of losing significant amounts of data in the event of packet loss increases, as more video data are encapsulated within each packet. Conversely, if packets are too small, the network overhead may become excessive, reducing the overall efficiency of the transmission.

Moreover, B-frame packets increase the coding delay. Furthermore, the encoder must wait for the next frame. Therefore, B-frame packets are not considered. Packet length is a significant factor in video transmission. For data packets with various frames, each packet waits until all video frames are ready. This leads to an additional delay. Thus, the packet length should be selected appropriately. Suppose that a represents the packet length. If the length of the packet surpasses length a, then the packet is split into several packets.

4.2.2 Problem formulation for buffer occupancy-based congestion control algorithm:

To define a mathematical model, the buffer occupancy, congestion levels, packet priority and transmission rate, feedback mechanism, packet drop probability, and the overall optimization problem are defined in Equation (4.1).

1. **Buffer occupancy:** Let $B(t)$ for a given time t represent the buffer occupancy, $B_{\text{prev}}(t)$ represent the buffer occupancy at the previous time step t - Δt, and $\delta(t)$ represent the rate of change in buffer occupancy; then, the rate of change in buffer occupancy is given by

$$\boldsymbol{\delta(t)} = \frac{\boldsymbol{B(t) - B_{\mathrm{prev}}(t)}}{\boldsymbol{\Delta t}} \tag{4.1}$$

An averaging technique for buffer occupancy over a short moving window is incorporated to handle the frequent changes in buffer states that lead to instability and increased overhead in the congestion control mechanism. This helps to mitigate the impact of frequent adjustments in the data transmission rate potentially overloading the network control mechanism. An averaging technique smooths out transient fluctuations and prevents rapid oscillations between congestion states. Specifically, an exponential moving average (EMA) is applied to buffer occupancy values over recent time slots. This is achieved by calculating the average buffer occupancy $\boldsymbol{B_{avg}(t)}$ given in equation (4.2).

$$\boldsymbol{B_{avg}(t-1) = \epsilon B(t) + (1-\epsilon) B_{avg}(t-1)} \tag{4.2}$$

where ϵ is a smoothing factor with a value ranging from 0.1 to 0.3, which is selected based on the network dynamics. $B_{avg}(t)$ is used instead of instantaneous B(t) for determining the congestion state. This effectively reduces short-term fluctuations, ensuring that sustained changes rather than momentary spikes trigger congestion control responses.. The use of averaging stabilizes the BOCC protocol, which helps in reducing the frequency of congestion notifications and minimizes unnecessary rate adjustments. This approach enhances system robustness by providing a smoother congestion response, aligning with goal of maintaining consistent QoS while efficiently managing network resources.

2. **Congestion levels:** Two congestion thresholds are defined: B_1 as the threshold for entering the slow state, and B_2 as the threshold for entering the urgent state. The system's state based on buffer occupancy can be categorized as follows:

- Normal state: $0 \le B(t) \le B_1$ and $\delta(t) \le \rho$.
- Slow state: $B_1 < B(t) \le B_2$ and $\delta(t) > \rho$.

– Urgent state: $B_2 < B(t) \leq B_{\max}$, where $B_{\max}$ is the buffer capacity.

Here, ρ is a predefined threshold rate of change that indicates when the buffer occupancy is increasing rapidly. The factor ρ is a critical parameter in the BOCC protocol as it determines when the buffer occupancy growth rate signals a transition between congestion states. Setting an effective ρ value is essential to balance responsiveness with stability. To determine ρ, both empirical analysis and network requirements are considered. This process typically involves the following steps:

– Empirical calibration: During the initial protocol testing phase, ρ is calibrated by analyzing network simulations under various traffic conditions. These simulations help observe typical buffer occupancy growth rates during normal and congested states. By analyzing buffer behavior across different traffic loads, an optimal threshold range for ρ can be identified.

– QoS and latency requirements: The value of ρ is also influenced by the specific QoS and latency requirements of the WMSN application. For applications sensitive to delays or packet losses, a lower ρ may be chosen to trigger congestion control more aggressively. Conversely, applications that can tolerate minor delays may use a higher ρ to prevent unnecessary rate adjustments in response to transient changes in buffer occupancy.

Adaptation through feedback: Additionally, an adaptive approach is proposed, where ρ can be dynamically adjusted based on network feedback. If the network experiences frequent but unnecessary congestion triggers, ρ can be increased incrementally. Similarly, if congestion frequently escalates without timely intervention, ρ can be decreased to improve responsiveness.

3. Packet priority and transmission rate: Let P_i denote the priority level of a packet, where $i \in \{1, 2\}$, P_1 represents the I-frame packets (highest priority), and P_2 represents the P-frame packets (lowest priority).

The transmission rate for node n at time t is represented as $S_n(t)$.

The transmission rate is adjusted based on the buffer occupancy:

- In the normal state: $S_n(t) = S_{\text{normal}}$ where S_{normal} is the normal data rate.
- In the slow state: $S_n(t) = \beta \cdot S_{\text{normal}}$ where $0 < \beta < 1$ is a factor determined experimentally.
- In the urgent state: $S_n(t) = \alpha \cdot S_{\text{normal}}$ where $0 < \alpha < \beta$ reduces the data rate further to prioritize high-priority packets.

4. Feedback mechanism: Each node receives feedback from its parent node about buffer occupancy. Let $F(t)$ denote the feedback received, which is a function of the parent node's buffer occupancy:

$$F(t) = f\left(B_{\text{parent}}(t)\right) \tag{4.3}$$

where $f(\mathrm{t})$ in equation (4.3) is a function that determines how the feedback influences the child node's behavior. Based on the feedback $F(t)$ computed in equation 4.4, the source node adjusts its transmission rate:

$$S_n(t) = S_n(t) \cdot \left(1 - F(t)\right). \tag{4.4}$$

This ensures that if the parent node is congested, the child node reduces its sending rate. In the BOCC protocol, each node adjusts its transmission rate, $S_n(t)$, based on the feedback $F(t)$, which reflects the buffer occupancy status of its parent node. The function $f\left(B_{\text{parent}}(t)\right)$ converts the buffer occupancy of the parent node, $B_{\text{parent}}(t)$, into a feedback value that modulates the child node's data transmission rate to prevent congestion. The purpose of $f\left(B_{\text{parent}}(t)\right)$ is to translate the parent node's buffer occupancy into a feedback multiplier, $F(t)$, that dynamically adjusts the data transmission rate of the child node. This adjustment controls the data flow to the parent node and avoids buffer overflow, effectively reducing congestion as the parent's buffer occupancy increases.

The feedback $F(t) = f\left(B_{\text{parent}}(t)\right)$ is not a data rate itself but a multiplier applied to the transmission rate $S_n(t)$ of the child node. Thus, $F(t)$ is a dimensionless value between 0 and 1, where 0 indicates no congestion (allowing the child node to transmit at full rate) and 1 indicates severe congestion (reducing the transmission rate to zero). The effective transmission rate of the child node then, is represented as equation (4.5)

$$S_n^{\text{adjusted}}(t) = S_n(t) \cdot \left(1 - F(t)\right) \tag{4.5}$$

where a higher $F(t)$ (indicating a higher buffer occupancy at the parent) results in a reduced transmission rate for the child node, helping to alleviate congestion at the parent node. The function $f\left(B_{\text{parent}}(t)\right)$ is a piecewise function as in equation (4.6), that maps buffer occupancy levels to feedback values:

$$f\left(B_{\text{parent}}(t)\right) = \begin{cases} 0, & \text{if } B_{\text{parent}}(t) \leq B_{\text{low}}, \\ \dfrac{B_{\text{parent}}(t) - B_{\text{low}}}{B_{\text{high}} - B_{\text{low}}}, & \text{if } B_{\text{low}} < B_{\text{parent}}(t) \leq B_{\text{high}}, \\ 1, & \text{if } B_{\text{parent}}(t) > B_{\text{high}}. \end{cases} \tag{4.6}$$

The thresholds B_{low} and B_{high} define the buffer occupancy levels where congestion control should begin (B_{low}) and where it should be maximized (B_{high}). Between these thresholds, $f\left(B_{\text{parent}}(t)\right)$ increases linearly from 0 to 1, allowing for a gradual response as buffer occupancy rises.

The BOCC protocol assumes that each node communicates with a single designated parent node, which results in a hierarchical tree-like structure. This model is relevant to hierarchical or cluster-based WMSNs, commonly found in applications such as surveillance or environmental monitoring. In real-world WMSNs, especially in multi-hop networks, data flows typically follow a tree structure to minimize data redundancy and optimize energy usage. Each node reports data to a central sink or gateway via intermediate parent nodes, making the single-parent assumption both practical and efficient. This hierarchical structure is suitable for scenarios where data aggregation and congestion control are crucial, as it allows the BOCC protocol to dynamically adjust transmission rates based on congestion levels and reduce overall network traffic.

5. Packet drop probability: The probability of packet drop $\mathbb{P}_{\text{drop}}$ is a function of buffer occupancy and congestion level. Let $\mathbb{P}_{\text{drop}}(t)$ represent the packet drop probability at time t. This can be modeled as

$$\mathbb{P}_{\text{drop}}(t) = \gamma \cdot \left(\frac{B(t)-B_1}{B_2-B_1}\right), \tag{4.7}$$

where γ is a scaling factor that controls how aggressively packets are dropped as the buffer occupancy increases.

6. Overall optimization problem: The goal of the BOCC algorithm is to minimize packet loss, as defined in Equation (4.7), while maintaining video quality and transmission rate, this can be formulated as an optimization problem:

$$\text{Minimize} \quad \mathbb{P}_{\text{drop}}(t) \cdot P_i + \lambda \cdot \delta(t), \tag{4.8}$$

subject to

$$0 \leq S_n(t) \leq S_{\text{normal}}, \tag{4.9}$$

$$0 \leq B(t) \leq B_{\text{max}}, \tag{4.10}$$

where λ is a weighting factor balancing the packet drop probability and the buffer occupancy rate of change.

The optimization problem presented in Equations (4.*8*) – (4.*10*) is intended to control congestion at each node by optimizing the data transmission rate, specifically for high-priority packets, to maintain video quality. The primary controllable parameter in this objective function is the transmission rate $S_n(t)$ for each node n at a given time t. This transmission rate is adjusted based on real-time feedback from congestion levels at the node itself and from its parent node. By controlling $S_n(t)$, the BOCC protocol can prioritize high-quality video frames (I-frames) while managing congestion to avoid buffer overflow and packet loss. Equation (4.5) is designed to minimize a combination of the packet drop probability $P_{drop}(t)$ and the rate of change of buffer occupancy $\delta(t)$. The $P_{drop}(t)$ represents the likelihood of packets being discarded due to buffer overflow. Minimizing $P_{drop}(t)$ is essential for maintaining high-quality multimedia transmission, as packet drops, particularly for I-frames, can degrade video quality significantly. The weight P_i in the term $P_{drop}(t)$ reflects the priority of each packet, emphasizing high-priority frames in the optimization.

The term $\lambda\delta(t)$ in the objective function penalizes rapid increases in buffer occupancy, which are indicative of potential congestion. By including this term, the optimization process encourages a steady flow of data rather than abrupt surges, thus stabilizing the buffer and preventing sudden congestion events. Together, these two components aim to balance the QoS for multimedia transmission with network stability. The weights P_i and λ allow the network to prioritize packet delivery and video quality while controlling buffer fluctuations. The formulation is initially presented for a single node n and a single time slot t as a baseline, focusing on controlling one high-priority frame type, typically an I-frame, which has the most significant impact on video quality. In a practical implementation, this single-transmission model serves as a foundation for continuous adjustment of $S_n(t)$ over multiple frames and time slots, adapting dynamically as network conditions change.

For a broader application, this single-instance formulation can be expanded to handle multiple nodes and frame types by applying the optimization iteratively for each time slot and adjusting $S_n(t)$ for each node based on its local congestion feedback. Equation (4.6) ensures that the transmission rate stays within a feasible range, bounded by a normal transmission rate S_{normal}, preventing the rate from either dropping to zero or excessively increasing and Equation (4.7) maintains the buffer occupancy within capacity limits, avoiding overflow. By keeping $B(t) \leq B_{max}$, the optimization process ensures that packets are not dropped due to excessive buffering.

4.3 Proposed Solutions:

This section outlines the proposed solutions for optimizing the BOCC algorithm using advanced mathematical techniques. To ensure the efficiency and effectiveness of the protocol, convex optimization and SQP methods are employed. These approaches are tailored to minimize packet loss and maintain video quality under varying network conditions. The subsections below describe each optimization approach and their specific applications within the BOCC algorithm framework.

4.3.1 Convex Optimization Framework for BOCC Algorithm:

The BOCC algorithm aims to reduce packet loss while maintaining the quality of video. The convex optimization framework is used to address this problem. The key steps involved as mentioned in Algorithm 1 in the convex optimization framework are given below:

- **Formulation of the optimization problem:** To obtain the optimal solution, the first step is to model and define the optimization problem with objectives to minimize the packet loss while maintaining the video quality defined in Equation (4.5), subject to the constraints defined in Equations (4.6) and (4.7). Where $S_n(t)$ represents the transmission rate at node n at time t, $B(t)$ depicts the buffer occupancy at time t, $\mathbb{P}_{\text{drop}}(t)$ is the probability of packet drop, P_i is the priority of the packet, λ is a weighting factor, and $\delta(t)$ is the rate of change of buffer occupancy.

- **Check for convexity:** The next step is to verify that the objective function and constraints are convex. The packet drop probability $\mathbb{P}_{\text{drop}}(t)$ is typically convex in $S_n(t)$, and the rate of change of buffer occupancy $\delta(t)$ is linear and convex. Hence, the sum of convex functions is convex, thus $f\big(S_n(t), B(t)\big)$ is convex. The constraints $0 \leq S_n(t) \leq S_{\text{normal}}$ and $0 \leq B(t) \leq B_{\max}$ are linear and convex.

Lagrangian function and dual problem: The next step is to formulate the Lagrangian and solve the dual problem. The Lagrangian function is given by equation (4.11).

$$\mathcal{L}(S_n(t), B(t), \mu, \nu) = \mathbb{P}_{\text{drop}}(t). P_i + \lambda \cdot \delta(t) + \mu(S_n(t) - S_{\text{normal}}) + \nu(B(t) - B_{\max}) \tag{4.11}$$

where μ and ν are the Lagrange multipliers. The dual problem is modeled to maximize the Lagrange dual function as in equation:

$$g(\mu, \nu) = \inf_{S_n(t), B(t)} \mathcal{L}(S_n(t), B(t), \mu, \nu), \tag{4.12}$$

subject to $\mu \geq 0$ and $\nu \geq 0$.

- **Karush–Kuhn–Tucker (KKT) conditions:** In the following step, KKT conditions is solved for optimality for different states for the functions defined in Equations (4.11) and (4.12).

Stationarity

$$\nabla_{S_n(t)} \mathcal{L}(S_n(t), B(t), \mu, \nu) = 0.$$

$$\nabla_{B(t)} \mathcal{L}(S_n(t), B(t), \mu, \nu) = 0.$$

Primal feasibility

$$0 \leq S_n(t) \leq S_{\text{normal}},$$

$$0 \leq B(t) \leq B_{\text{max}}.$$

Dual feasibility

$$\mu \geq 0, \quad \nu \geq 0.$$

Complementary slackness

$$\mu(S_n(t) - S_{\text{normal}}) = 0.$$

$$\nu(B(t) - B_{\text{max}}) = 0.$$

- **Solving the optimization problem:** After that, the optimal values of $S_n(t)$ and $B(t)$ is solved using the KKT conditions, and presented in algorithm 4.1.

Algorithm 4.1: Optimization using stationary equations.

Input:
Initial values for $S_n(t)$, $B(t)$, μ, and ν.
Output:
Optimized values for $S_n(t)$, $B(t)$, μ, and ν.
Initialization:

Set initial values for $S_n(t)$, $B(t)$, μ, and ν.
Iteration:
(i) Update $S_n(t)$ and $B(t)$ using the stationarity equations.

(ii) Check primal and dual feasibility.

(iii) Adjust μ and ν using complementary slackness.

(iv) Continue iteration until convergence to the optimal values.

Convergence:
Stop when the values converge to the optimal solution.
End.

- **Validation and fine-tuning:** The last step is to validate the results after obtaining the optimal solution by testing under different network conditions and tuning the weighting factor λ to balance the packet drop probability and buffer occupancy.

4.3.2 Sequential Quadratic Programming (SQP) Solution for BOCC Algorithm Optimization:

The key objectives of the BOCC algorithm are to minimize packet loss while maintaining video quality constraints. This problem is formulated as a constrained optimization problem.

The algorithm for SQP method is depicted in Algorithm *2* approximates the original nonlinear problem by solving a sequence of quadratic programming (QP) subproblems. This includes a series of steps.

1. Initialization: Begin with an initial guess x_0 for the decision variables $x = [S_n(t), B(t)]$. Initialize the Lagrange multipliers λ_0 for the equality constraints (if any) and μ_0 for the inequality constraints.

2. Formulate the quadratic programming (QP) subproblem: At each iteration k, formulate the QP subproblem equation (4.13) by approximating the Lagrangian function $\mathcal{L}(x, \lambda, \mu)$ around the current point x_k :

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_i \lambda_i \, c_i(x) + \sum_j \mu_j \, g_j(x), \tag{4.13}$$

where $f(x)$ is the objective function, $c_i(x)$ is the equality constraints, and $g_j(x)$ is the inequality constraints.

The QP subproblem defined in Equation (4.13) needs to be solved at the k-th iteration, which is given by equation (4.14)

$$\text{Minimize} \quad \frac{1}{2} \Delta x^\top H_k \Delta x + \nabla f(x_k)^\top \Delta x, \tag{4.14}$$

subject to

$$\nabla c_i(x_k)^\top \Delta x + c_i(x_k) = 0, \quad i = 1, \dots, m. \tag{4.15}$$

$$\nabla g_j(x_k)^\top \Delta x + g_j(x_k) \leq 0, \quad j = 1, \dots, p. \tag{4.16}$$

where $H_k = \nabla^2 \mathcal{L}(x_k, \lambda_k, \mu_k)$ is the Hessian matrix of the Lagrangian, $\nabla f(x_k)$ is the gradient of the objective function at x_k, and Δx is the search direction.

3. Solve the QP subproblem: Following the steps, then solve the QP subproblem using a QP solver. The solution provides the search direction Δx_k for the next iteration.

4. Update variables: The next step is to update the decision variables and Lagrange multipliers given by

$$x_{k+1} = x_k + \alpha_k \Delta x_k,$$

where α_k is the step size determined by a line search method and updates the Lagrange multipliers λ_k and μ_k accordingly.

5. Convergence check: The next step is to check if the solution has converged or not by examining the norm of the gradient and the magnitude of the constraint violations. If the solution has not converged, return to step 2 with iteration $k = k + 1$ in Equations (4.14)–(4.16).

6. Termination: Once the optimal convergence is achieved, then the algorithm terminates with the optimal solution x^*.

4.3.3 Quadratic Programming (QP) Subproblem:

In the SQP method, the original nonlinear optimization problem is approximated by a sequence of QP subproblems, as defined in Equation (4.14). In each iteration, the QP subproblem is computed by linearizing the constraints and approximating the objective function using a second-order Taylor expansion of the Lagrangian function, which is defined in equation (4.17).

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i \, c_i(x) + \sum_{j=1}^{p} \mu_j \, g_j(x), \tag{4.17}$$

where $f(x)$ is the objective function, $c_i(x) = 0$ is the equality constraints for $i = 1, \ldots, m$. $g_j(x) \leq 0$ is the inequality constraints for $j = 1, \ldots, p$. λ_i and μ_j are the Lagrange multipliers associated with the equality and inequality constraints, respectively. The upper limits m and p represent the number of equality and inequality constraints, respectively. m is the number of equality constraints $c_i(x) = 0$ and p is the number of inequality constraints $g_j(x) \leq 0$.

(a) Formulation of the QP Subproblem:

The QP subproblem at the k-th iteration is obtained by approximating the Lagrangian function defined in Equation (4.17) around the current iteration x_k, given by equation (4.18).

$$\text{Minimize} \quad \frac{1}{2}\Delta x^{\top} H_k \Delta x + \nabla f(x_k)^{\top}\Delta x, \tag{4.18}$$

subject to

$$\nabla c_i(x_k)^{\top}\Delta x + c_i(x_k) = 0, \quad i = 1, \dots, m, \tag{4.19}$$

$$\nabla g_j(x_k)^{\top}\Delta x + g_j(x_k) \leq 0, \quad j = 1, \dots, p, \tag{4.20}$$

where $H_k = \nabla^2 \mathcal{L}(x_k, \lambda_k, \mu_k)$ is the search direction from the current iteration x_k. $H_k = \nabla^2 L(x_k, \lambda_k, \mu_k)$ is the Hessian matrix of the Lagrangian function with respect to x at x_k. $\nabla f(x_k)$ is the gradient of the objective function at x_k, and $\nabla c_i(x_k)$ and $\nabla g_j(x_k)$ are the gradients of the equality and inequality constraints at x_k, respectively.

(b) Interpretation of the QP Subproblem:

The QP subproblem is a quadratic optimization problem in Δx, with the following:

- A quadratic objective function:

$$\frac{1}{2}\Delta x^{\top} H_k \Delta x + \nabla f(x_k)^{\top}\Delta x. \tag{4.21}$$

- Linear equality constraints:

$$\nabla c_i(x_k)^{\top}\Delta x + c_i(x_k) = 0, \quad i = 1, \dots, m. \tag{4.22}$$

- Linear inequality constraints:

$$\nabla g_j(x_k)^{\top}\Delta x + g_j(x_k) \leq 0, \quad j = 1, \dots, p. \tag{4.23}$$

(c) Solution of the QP Subproblem:

The QP subproblem can be solved using standard QP solvers, which are designed to handle problems of this form efficiently.

- Solve the quadratic objective function by finding the search direction Δx_k that minimizes the function while satisfying the constraints defined in Equations (4.18)–(4.20). This can be achieved using methods such as the active-set method, interior-point method, or others, depending on the specific QP solver used.

- Update the decision variables: Once the optimal search direction Δx_k is found, then update the decision variables $x_{k+1} = x_k + \alpha_k \Delta x_k$, where α_k is a step size determined by a line search or trust-region approach.

- Update the Lagrange multipliers: Update the Lagrange multipliers λ_k and μ_k based on the solution of the QP subproblem, ensuring that the Karush–Kuhn–Tucker (KKT) conditions are satisfied.

- Check for convergence: Finally, check if the solution has converged. Convergence is achieved when the norm of the gradient of the Lagrangian function is sufficiently small, and the constraints defined in Equations (4.21) – (4.23) are satisfied with minimal violation.

The QP subproblem is a key step in the SQP method, providing a way to iteratively approximate and solve the original nonlinear constrained optimization problem. By solving the QP subproblem at each iteration, the SQP method effectively navigates the feasible region to find the optimal solution to the BOCC algorithm's optimization problem.

4.4 Solving Quadratic Programming Problem Using the Active-Set Method (ASM):

The ASM technique is the iterative method that exploits both inequality and equality constraints to solve QP problems. This method maintains and updates a list of active constraints (i.e., those that are binding at the solution). The QP problem can be represented in a generalized form given by

Minimize $$\frac{1}{2}x^{\top}Hx + f^{\top}x, \tag{4.24}$$

subject to

$$A_{\mathrm{eq}}x = b_{\mathrm{eq}}, \tag{4.25}$$

$$A_{\mathrm{ineq}}x \leq b_{\mathrm{ineq}}, \tag{4.26}$$

where H is the Hessian matrix, f is the gradient vector of the objective function, A_{eq} and A_{ineq} are matrices representing the coefficients of the equality and inequality constraints, respectively, and b_{eq} and b_{ineq} are vectors representing the right-hand side of the equality and inequality constraints.

4.4.1 Active-Set Method:

The active-set method solves the QP problem by iteratively adjusting the set of active constraints, solving a sequence of equality-constrained QP subproblems.

1. Initialization:

- Start with an initial feasible point x_0 that satisfies all the constraints.
- Initialize the working set W_0 to include all the equality constraints and any inequality constraints that are active (i.e., those for which $A_{ineq}x_0 = b_{ineq}$).

2. Iterative steps. At each iteration k, the following steps are performed:

- Solve the equality-constrained QP subproblem: Solve the QP subproblem defined in Equations (4.24) – (4.26) that considers only the constraints in the current working set W_k. The solution to this subproblem gives a search direction Δx_k.

- Check for optimality: Compute the Lagrange multipliers λ_k associated with the constraints in the working set W_k. The current solution is optimal if all the Lagrange multipliers corresponding to inequality constraints are non-negative. If not, remove the constraint with the most negative multiplier from the working set and go to the next step.

- Determine step length: Compute the maximum step length α_k that maintains feasibility with respect to all inequality constraints:

$$\alpha_k = \min\left(1, \min_{j \notin W_k} \frac{b_j - A_j^\top x_k}{A_j^\top \Delta x_k}\right), \tag{4.27}$$

where W_k is the set of active constraints at iteration k, A_j and b_j represent the coefficients and the right-hand side of the inequality constraints, and Δx_k is the search direction and updates the solution.

$$x_{k+1} = x_k + \alpha_k \Delta x_k. \tag{4.28}$$

- Update the working set: If $\alpha_k < 1$, the step is blocked by some inequality constraint not currently in the working set. Add the most restrictive constraint to the working set W_{k+1}.

- Repeat: Repeat steps 1 through 4 updating the iterations in Equations (4.27) and (4.28) until convergence is achieved.

4.4.2 Convergence: subproblems by adjusting the set of active constraints. This approach ensures that each iteration considers only the constraints actively influencing the current solution. Convergence occurs when the optimal solution meets the Karush–Kuhn–Tucker (KKT) conditions, ensuring that both the objective function and constraints are satisfied.

The proposed model is purposefully designed to address congestion in WMSNs by optimizing two core parameters: the packet drop probability and the buffer occupancy change rate. These two components are critical in WMSNs where QoS requirements are high, especially in real-time video and multimedia streaming applications. The parameter $P_{drop}(t)$ captures the likelihood of packet loss due to buffer overflow, which directly affects video quality. By minimizing $P_{drop}(t)$, the model prioritizes high-quality multimedia transmission, particularly for I-frame packets, which are crucial in maintaining overall video clarity and integrity. The parameter $\delta(t)$ minimizes abrupt increases in buffer occupancy, which are early indicators of potential congestion. The buffer occupancy change rate directly impacts network stability; hence, by controlling $\delta(t)$, the model prevents rapid, destabilizing congestion triggers. This is essential for reducing jitter and maintaining smooth video playback across the network.

Together, these two parameters provide a balanced approach that directly targets the critical QoS aspects of packet delivery and network stability in a manner suitable for WMSNs. The objective function in Equation (4.5) is a weighted combination of $P_{drop}(t)$. The weights P_i and λ are carefully selected to reflect the priority of packet types (i.e., high-priority I-frames) and the network's tolerance for changes in buffer occupancy. Using a simple, interpretable objective, computational efficiency is ensured, which is essential for real-time network adjustments in WMSNs. This design choice keeps the optimization lightweight, allowing it to be implemented in resource-constrained sensor nodes without overwhelming computational demands.

The optimization model and its results were rigorously validated through empirical testing, as described in the following Section *5*. The simplicity of the optimization model is designed to achieve efficient congestion control without unnecessary computational complexity. The experimental results validate that proposed model captures the necessary dynamics to support QoS in WMSNs.

4.5 Experimental Results:

The experimental performance of BOCC using 13 Raspberry Pi-4 Model B devices (Raspberry Pi Foundation, Cambridge, UK) is presented in this section as shown in Figure 4.3. These nodes are placed in fixed positions in cabin enclosures. The camera is used to record videos. Every Raspberry Pi node stores a pre-recorded video. Each source node sends captured video data to the sink node using multi-hop communication. The efficiency of BOCC is determined by four parameters. These four parameters are the average end-to-end delay, peak signal-to-noise ratio, packet delivery ratio, and buffer occupancy. This section compares the following algorithms:

BOCC: This algorithm presented in this chapter

ECODA: This algorithm is presented in. [41]

PPI: This algorithm is given in. [100]

CDTMRLB: This algorithm is provided in. [42]

NoBOCC: No congestion control mechanism is employed in NoBOCC.

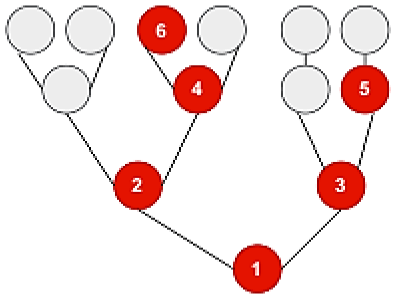

Figure 4.3 Routing topology used in BOCC.

4.5.1 Packet Delivery Ratio of I-Frame Packets:

Figure *4.4* depicts how the percentage of I-frames that successfully reach their destination changes as the data transmission rate varies. Figure *4.4* shows that BOCC achieves an 83% packet delivery ratio, while PPI, ECODA, CDTMRLB, ECODA, and NoBOCC attain 79%, 77%, 73%, and 52% packet delivery ratios at a data rate of 30 packets per second, respectively. BOCC discards some P-frame packets when congestion occurs to save the I-packets. Thus, BOCC has a better I-frame packet delivery ratio in comparison with the other protocols. The ECODA, PPI, and CTRLB protocols perform better than NoBOCC. NoBOCC offers no congestion control technique; so, it shows minimal improvement compared to the other algorithms.

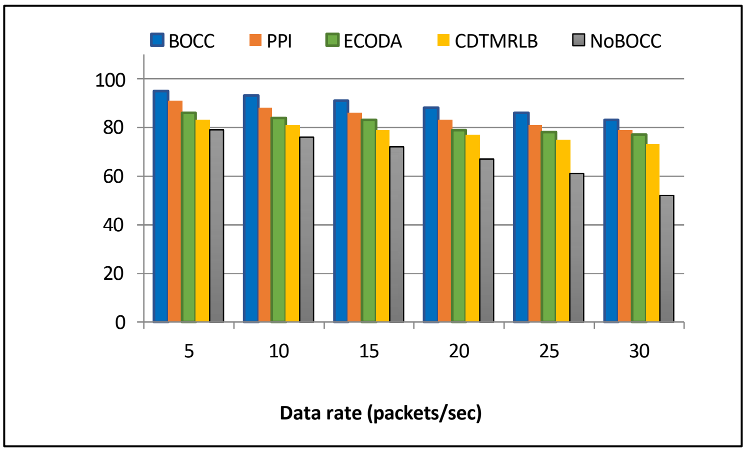

Figure 4.4 Packet loss of I-frame packets vs. data rate.

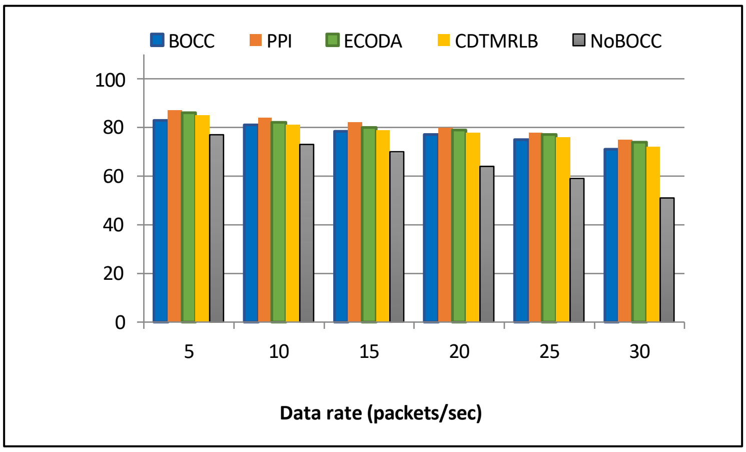

Figure 4.5 Packet loss of P-frame packets vs. data rate.

4.5.2 Packet Delivery Ratio of P-Frame Packets:

Figure *4.5* illustrates the connection between the P-frame packet delivery ratio and the data rate. NoBOCC offers no congestion control mechanism. Therefore, NoBOCC contains the highest number of lost P-frame packets. BOCC discards P-frame packets in congested conditions to protect I-frame packets. Thus, the CDTMRLB, PPI, and ECODA protocols have a small number of P-frame packet losses compared to BOCC. This is because ECODA, CDTMRLB, and PPI use the same effective technique for both I- and P-frame packets. Figure *8* shows that PPI has a 75% packet delivery ratio, whereas ECODA, CDTMRLB, BOCC, and NoBOCC achieve 74%, 72%, 71%, and 51% packet delivery ratios, respectively, at a data rate of 30 packets per second.

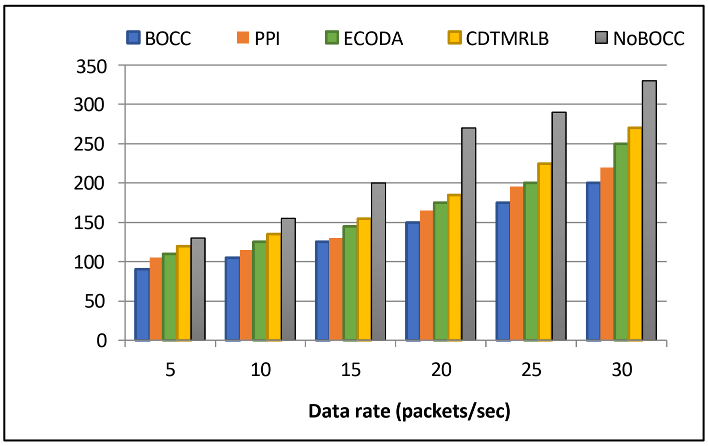

Figure 4.6 Average end-to-end delay vs. data rate.

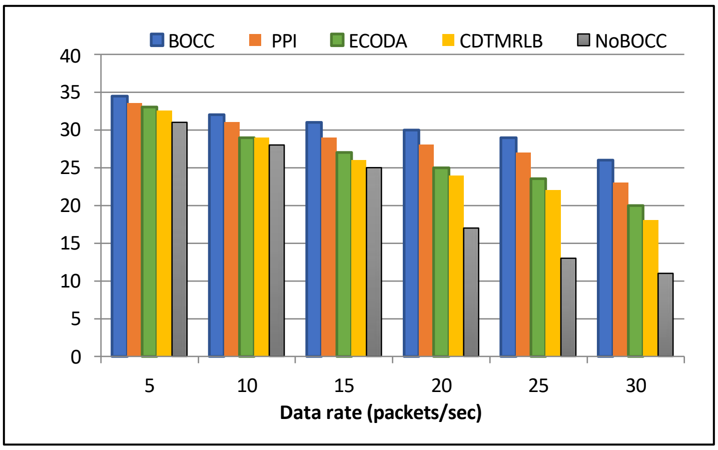

Figure 4.7 Average video quality vs. data rate.

4.5.3 Average End-to-End Delay vs. Data Rate:

It is observed that sending data at a slower rate (smaller data rate) does not cause any noticeable delays. Figure *4.6* shows that an increased data rate results in increased end-to-end delay. Increased end-to-end delays can saturate the network. Compared to other protocols, BOCC has the smallest average end-to-end delay. NoBOCC has the highest average end-to-end delay, as illustrated in Figure *4.6*. BOCC outperforms all the other algorithms in terms of performance. BOCC has an end-to-end delay of 200 ms. PPI, ECODA, CDTMRLB, and NoBOCC has end-to-end delays of 220 ms, 250 ms, 270 ms, and 330 ms, respectively, at 30 packets per second.

4.5.4 PSNR:

Figure *4.7* represents the relationship between the peak signal-to-noise ratio (PSNR) and data rate. The PSNR shows the received video quality. A pre-recorded H.264 video stream is stored in each Raspberry Pi to evaluate the effect of the PSNR. The PSNR decreases with increasing data rate. Obviously, a lower number of I-frames results in superior video quality. NoBOCC has the lowest PSNR. The ECODA, PPI, and CDTMRLB protocols lose fewer I-frame packets than NoBOCC. Therefore, these algorithms perform better than NoBOCC. BOCC shows superior video quality compared to the ECODA, PPI, CDTMRLB, and

NoBOCC protocols. This is because BOCC has the fewest lost I-frame packets. BOCC obtains 26 dB PSNR while PPI, ECODA, CDTMRLB, and NoBOCC have PSNRs of 23 dB, 20 dB, 18 dB, and 11 dB, respectively, at a data rate of 30 packets per second. This illustrates the significance of saving I-frame packets in WMSNs.

Figure *4.8* shows how different metrics interact and change over time for the data received using WMSN nodes. The figure depicts two metrics, i.e., the packet loss graph shows how the probability of packet loss varies over time. The video quality graph shows how the quality of the video (i.e., peak signal-to-noise ratio; PSNR) changes over time. A sinusoidal pattern with exponential decay shows periodic packet loss fluctuations, due to varying network conditions such as traffic loads. The presence of random noise indicates that packet loss is not entirely predictable and can be influenced by unpredictable network factors.

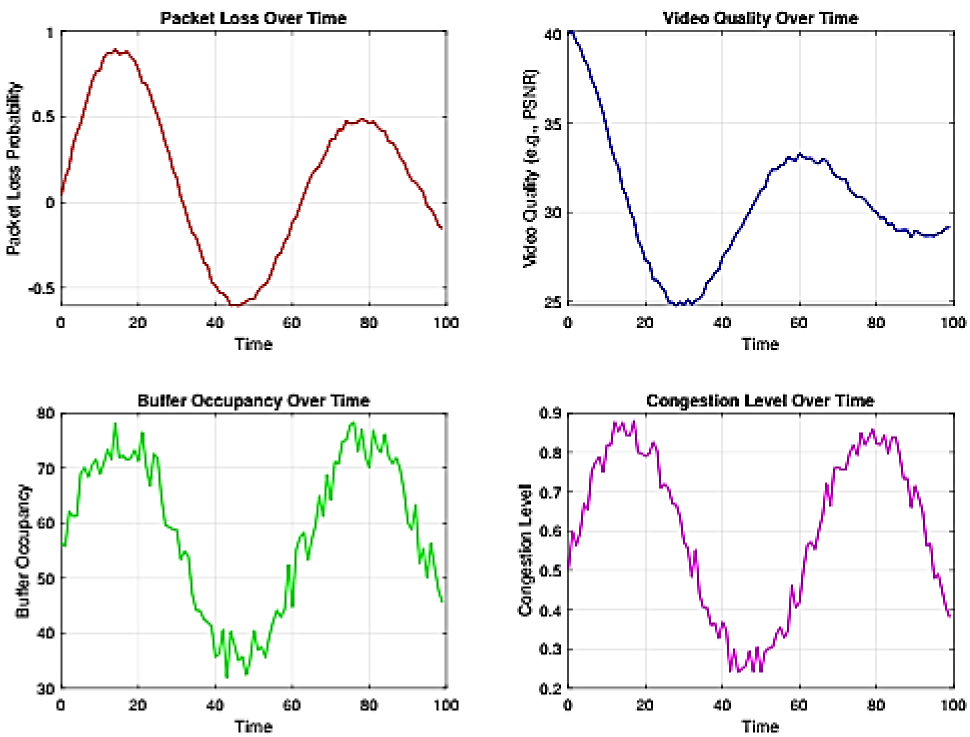

Figure 4.8 Performance metrics over time.

Periodic increases in packet loss indicate specific times when the network experiences higher load or interference. The exponential decay suggests that, over time, the packet loss probability decreases, due to improved network conditions and the adaptation mechanisms. In the second graph in Figure 4.*8*, the cosine pattern with exponential decay indicates that video quality oscillates but tends to decrease over time. Random noise adds variability, showing that video quality is not consistently stable. Fluctuations in video quality could be related to varying network conditions or buffer occupancy levels. The decay in quality over time indicates that network congestion or packet loss is affecting the video stream. A sinusoidal trend with random noise indicates periodic changes in buffer occupancy. The buffer occupancy fluctuates but stays within a bounded range (0 to 100), indicating controlled buffer usage. Periodic increases and decreases in buffer occupancy are related to varying data arrival rates. High buffer occupancy is correlated with higher packet loss or congestion, impacting video quality. A sinusoidal pattern with clipping between 0 and 1 indicates the periodic congestion levels. Random noise introduces variability, indicating that congestion levels can fluctuate unpredictably. Periods of high congestion levels correlate with increased packet loss and decreased video quality.

The proposed BOCC algorithm is effective in congestion management and buffering techniques to maintain optimal network performance. The proposed technique carefully balances the trade-offs between packet loss, video quality, buffer occupancy, and congestion. It also monitors and adjusts the parameters like the weighting factor λ, which can help optimize the performance and maintain a balance between packet drop probability and buffer occupancy, as shown in Figures 4.9 and 4.10 for convex optimization and SQP optimization, respectively. The convex optimization and SQP techniques are used to assess and optimize the performance of the network by iterating over different λ values. They collect and visualize the results to find the optimal balance between packet loss and video quality. Figures 4.9 and 4.10 help to visualize how changing λ affects both packet loss and video quality.

Figure 4.9 shows the packet loss rate vs. λ, and the second graph in this figure shows the video quality vs. λ, solved using the convex optimization technique. These graphs show how packet loss varies with different values of λ; as λ changes, packet loss also varies due to the trade-off between network congestion and data prioritization. In the context of optimization, λ value is considered where packet loss is minimized without compromising video quality. The second graph illustrates how video quality changes with different λ values. Higher λ values prioritize reducing packet loss, potentially at the cost of video quality, while lower values focus more on maintaining video quality even if it means higher packet loss. A balance is desirable where video quality remains high while packet loss is minimized. At λ=0.4040, the objective function reaches its minimum value, indicating the best trade-off between minimizing packet loss and maintaining video quality given the specific weights α and β. If the weights α and β are changed, then the optimal λ might shift. Therefore, this result is specific to the current weight settings and may need adjustment for different network conditions or priorities.

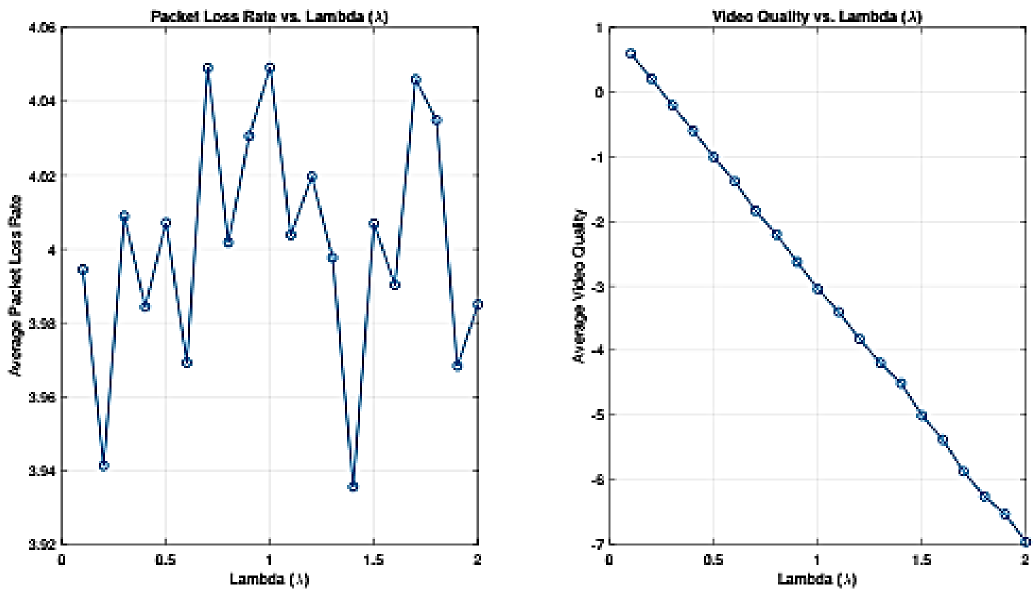

Figure 4.9 Optimization parameters (convex optimization).

Figure 4.10 shows the packet loss rate vs. λ and video quality vs. λ, solved using the SQP optimization technique. In the SQP optimization technique, at $\lambda = 0.6102$ the objective function reached its minimum value, indicating the best scenario, where video quality is maintained while minimizing the packet loss. These results show that the BOCC algorithm is capable of finding the most suitable value for different network parameters and conditions to effectively achieve better video quality with minimum packet loss.

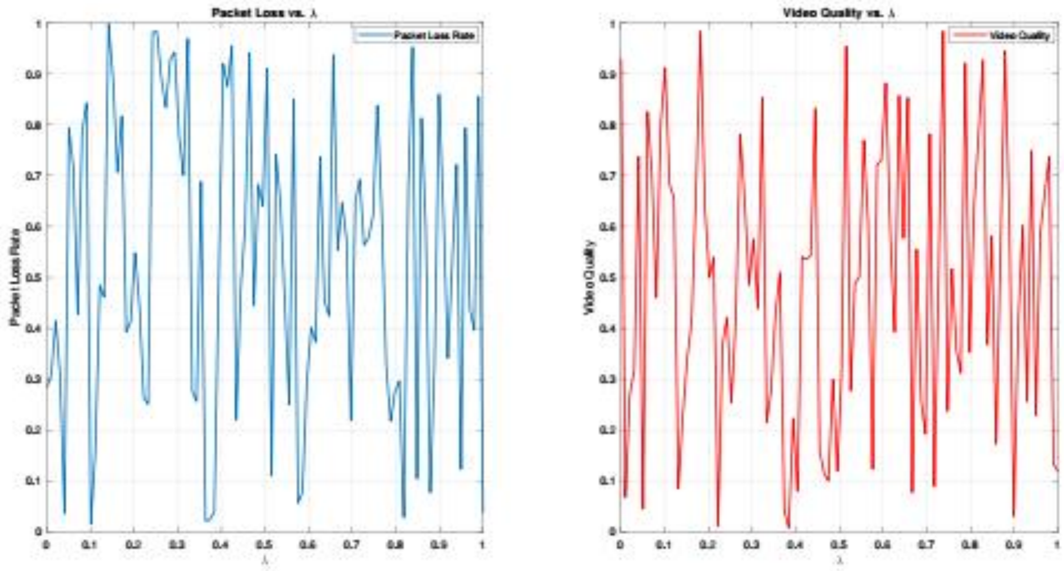

Figure 4.10 Optimization parameters (SQP optimization).

4.6 Conclusions:

The BOCC protocol effectively mitigates congestion in WMSNs by leveraging buffer occupancy metrics and change rates to adjust data transmission dynamically. With a feedback-based rate controller that prioritizes high-priority packets, BOCC ensures efficient video stream transmission even under congested conditions. The experimental results in a multi-hop, tree-topology WMSN environment with Raspberry Pi nodes demonstrate up to 16% reduction in packet loss and up to 23% reduction in average end-to-end delay compared to state-of-the-art congestion control algorithms. Additionally, optimization through convex programming and SQP enhances BOCC's performance, with convex optimization providing faster convergence. Future work will explore BOCC's adaptability to more diverse topologies and its integration with machine learning techniques to enhance congestion prediction and rate control. These advancements will further solidify BOCC as a scalable, efficient solution for dynamic WMSN environments.

Chapter 5

Conclusion and Future Work

Conclusion:

WMSNs are able to capture video data from the environment. This video data can be application specific, event-based or real-time in nature. In WMSNs, congestion happens when data carried by the network surpasses the available capacity. Congestion results in exhausted node energy, degradation of network performance, high packet loss, and an increase in network latency. Therefore, designing congestion control protocols in WMSNs to efficiently detect, notify, and control congestion is very significant. Moreover, the protocols should also ensure. Moreover, the protocols should also ensure reliable delivery of data in resource-constrained WMSNs.

First, a DCCP is proposed to mitigate congestion and improve end-to-end delay. The DCCP protocol proposed in this dissertation can alleviate congestion by intelligently selecting the best path. First, congestion is detected using two congestion indicators. Second, a traffic congestion map is used to calculate the optimum path. Therefore, the traffic is balanced on different routes, which reduces the end- to-end delay. Finally, a rate controller is designed to prevent congestion in the network by sending a congestion notification message to a source node. After receiving the congestion notification message, the source node immediately adjusts its transmission rate. Experimental results illustrate that DCCP significantly improves network performance and is superior to existing modern congestion control protocols.

Secondly, a BOCC is proposed. Compared to the other protocols, the BOCC protocol proposed in this chapter can alleviate congestion by intelligently saving the data that contains more information. First, congestion is detected using two congestion indicators. Secondly, a rate controller is designed to protect high priority I-frame packets during congestion. BOCC sends a congestion notification to the source node to reduce congestion in the network. The proposed algorithm adjusts the rate by discarding low-priority P-frame packets from the source nodes. Moreover, to further improve the performance of the BOCC protocol, the problem is formulated as a constrained optimization problem and solved using convex optimization and SQP methods.

Future Work:

In future security-aware congestion control strategies need to be developed to prevent potential attacks, ensuring the reliable transmission of data in resource-constrained environments.

References

[1] U. Majeed, A. N. Malik, N. Abbas, W. Abbass, “An Energy-Efficient Distributed Congestion Control Protocol for Wireless Multimedia Sensor Networks,” *Electronics* **2022**, vol. *11*, 3265, 2022.

[2] U. Majeed, A. N. Malik, N. Abbas, A.S. Alfakeeh, M.A. Javed, W. Abbass, “Buffer Occupancy-Based Congestion Control Protocol for Wireless Multimedia Sensor Networks.”, *Electronics* **2024**, vol. *13*, 4454, 2024.

[3] M. Usman, M. A. Jan, X. He and J. Chen, "A Mobile Multimedia Data Collection Scheme for Secured Wireless Multimedia Sensor Networks," in *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 274-284, 1 Jan.-March 2020, doi: 10.1109/TNSE.2018.2863680.

[4] B. -N. Trinh, L. Murphy and G. -M. Muntean, "A Reinforcement Learning-Based Duty Cycle Adjustment Technique in Wireless Multimedia Sensor Networks," in *IEEE Access*, vol. 8, pp. 58774-58787, 2020, doi: 10.1109/ACCESS.2020.2982590.

[5] A. Yazici, M. Koyuncu, S. A. Sert and T. Yilmaz, "A Fusion-Based Framework for Wireless Multimedia Sensor Networks in Surveillance Applications," in *IEEE Access*, vol. 7, pp. 88418-88434, 2019, doi: 10.1109/ACCESS.2019.2926206.

[6] M. Koyuncu, A. Yazici, M. Civelek, A. Cosar and M. Sert, "Visual and Auditory Data Fusion for Energy-Efficient and Improved Object Recognition in Wireless Multimedia Sensor Networks," in *IEEE Sensors Journal*, vol. 19, no. 5, pp. 1839-1849, 1 March1, 2019, doi: 10.1109/JSEN.2018.2885281.

[7] M. Civelek and A. Yazici, "Automated Moving Object Classification in Wireless Multimedia Sensor Networks," in *IEEE Sensors Journal*, vol. 17, no. 4, pp. 1116-1131, 15 Feb.15, 2017, doi: 10.1109/JSEN.2016.2638853.

[8] Y. Zhou, W. Xiang and G. Wang, "Frame Loss Concealment for Multiview Video Transmission Over Wireless Multimedia Sensor Networks," in *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1892-1901, March 2015, doi: 10.1109/JSEN.2014.2366511.

[9] T. Mekonnen, P. Porambage, E. Harjula and M. Ylianttila, "Energy Consumption Analysis of High Quality Multi-Tier Wireless Multimedia Sensor Network," in *IEEE Access*, vol. 5, pp. 15848-15858, 2017, doi: 10.1109/ACCESS.2017.2737078.

[10] G. Mali and S. Misra, "TRAST: Trust-Based Distributed Topology Management for Wireless Multimedia Sensor Networks," in *IEEE Transactions on Computers*, vol. 65, no. 6, pp. 1978-1991, 1 June 2016, doi: 10.1109/TC.2015.2456026.

[11] X. Zhang and Q. Zhu, "Information-Centric Virtualization for Software-Defined Statistical QoS Provisioning Over 5G Multimedia Big Data Wireless Networks," in *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1721-1738, Aug. 2019, doi:

10.1109/JSAC.2019.2927088.

[12] M. Z. Hasan, H. Al-Rizzo and F. Al-Turjman, "A Survey on Multipath Routing Protocols for QoS Assurances in Real-Time Wireless Multimedia Sensor Networks," in IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1424-1456, thirdquarter 2017, doi: 10.1109/COMST.2017.2661201.

[13] C. Küçükkeçeci and A. Yazici, "Multilevel Object Tracking in Wireless Multimedia Sensor Networks for Surveillance Applications Using Graph-Based Big Data," in *IEEE Access*, vol. 7, pp. 67818-67832, 2019, doi: 10.1109/ACCESS.2019.2918765.

[14] F. Al-Turjman, A. Radwan, S. Mumtaz and J. Rodriguez, "Mobile traffic modelling for wireless multimedia sensor networks in IoT", *Comput. Commun.*, vol. 112, pp. 109-115, Nov. 2018.

[15] A. B. Noel, A. Abdaoui, T. Elfouly, M. H. Ahmed, A. Badawy and M. S. Shehata, "Structural health monitoring using wireless sensor networks: A comprehensive survey", *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1403-1423, 3rd Quart. 2017.

[16] M. E. E. D. Abd El Kader, A. A. A. Youssif and A. Z. Ghalwash, "Energy Aware and Adaptive Cross-Layer Scheme for Video Transmission Over Wireless Sensor Networks," in *IEEE Sensors Journal*, vol. 16, no. 21, pp. 7792-7802, Nov.1, 2016, doi: 10.1109/JSEN.2016.2601258.

[17] H. Li, Q. Qi, J. Liu, P. Zhao and Y. Yang, "Mobile Wireless Multimedia Sensor Networks Image Compression Task Collaboration Based on Dynamic Alliance," in *IEEE Access*, vol. 8, pp. 86024-86037, 2020, doi: 10.1109/ACCESS.2020.2992795.

[18] L. Guo, Z. Ning, Q. Song, L. Zhang and A. Jamalipour, "A QoS-Oriented High- Efficiency Resource Allocation Scheme in Wireless Multimedia Sensor Networks," in *IEEE Sensors Journal*, vol. 17, no. 5, pp. 1538-1548, 1 March1, 2017, doi: 10.1109/JSEN.2016.2645709.

[19] T. -L. Lin, H. -W. Tseng, Y. Wen, F. -W. Lai, C. -H. Lin and C. -J. Wang, "Reconstruction Algorithm for Lost Frame of Multiview Videos in Wireless Multimedia Sensor Network Based on Deep Learning Multilayer Perceptron Regression," in *IEEE Sensors Journal*, vol. 18, no. 23, pp. 9792-9801, 1 Dec.1, 2018, doi: 10.1109/JSEN.2018.2865916.

[20] T. Mekonnen *et al*., "Energy Consumption Analysis of Edge Orchestrated Virtualized Wireless Multimedia Sensor Networks," in *IEEE Access*, vol. 6, pp. 5090-5100, 2018, doi: 10.1109/ACCESS.2017.2783447.

[21] C. M. Park, R. A. Rehman and B. -S. Kim, "Packet Flooding Mitigation in CCN-Based Wireless Multimedia Sensor Networks for Smart Cities," in *IEEE Access*, vol. 5, pp. 11054-11062, 2017, doi: 10.1109/ACCESS.2017.2715407.

[22] L. Nachabe, M. Girod-Genet and B. El Hassan, "Unified Data Model for Wireless Sensor Network," in *IEEE Sensors Journal*, vol. 15, no. 7, pp. 3657-3667, July 2015, doi: 10.1109/JSEN.2015.2393951.

[23] A. Bavarva, P. V. Jani and K. Ghetiya, "Performance Improvement of Wireless Multimedia Sensor Networks using MIMO and Compressive Sensing," in *Journal of Communications and Information Networks*, vol. 3, no. 1, pp. 84-90, March 2018, doi: 10.1007/s41650-018-0011-8.

[24] F. Liu, H. Lu, T. Wang and Y. Liu, "An Energy-Balanced Joint Routing and Charging Framework in Wireless Rechargeable Sensor Networks for Mobile Multimedia," in *IEEE Access*, vol. 7, pp. 177637-177650, 2019, doi: 10.1109/ACCESS.2019.2958268.

[25] X. Qiu, Z. Du and X. Sun, "Artificial Intelligence-Based Security Authentication: Applications in Wireless Multimedia Networks," in *IEEE Access*, vol. 7, pp. 172004- 172011, 2019, doi: 10.1109/ACCESS.2019.2956480.

[26] S. K. Satyanarayana and L. N. Devi, "A Reliable Link Selection and Data Aggregation Strategy for Wireless Multimedia Networks," in *IEEE Sensors Letters*, vol. 6, no. 10, pp. 1-4, Oct. 2022, Art no. 7501104, doi: 10.1109/LSENS.2022.3178756.

[27] Y. Tian and R. Liao, "Multinode Collaborative Image Compression Algorithm for Wireless Multimedia Sensor Networks Based on LBT," in *IEEE Sensors Journal*, vol. 20, no. 20, pp. 12065-12073, 15 Oct.15, 2020, doi: 10.1109/JSEN.2020.2990475.

[28] L. Cheng, J. Niu, M. Di Francesco, S. K. Das, C. Luo and Y. Gu, "Seamless Streaming Data Delivery in Cluster-Based Wireless Sensor Networks With Mobile Elements," in *IEEE Systems Journal*, vol. 10, no. 2, pp. 805-816, June 2016, doi: 10.1109/JSYST.2015.2459071.

[29] M. Usman, N. Yang, M. A. Jan, X. He, M. Xu and K. -M. Lam, "A Joint Framework for QoS and QoE for Video Transmission over Wireless Multimedia Sensor Networks," in *IEEE Transactions on Mobile Computing*, vol. 17, no. 4, pp. 746-759, 1 April 2018, doi: 10.1109/TMC.2017.2739744.

[30] F. Al-Turjman and A. Radwan, "Data Delivery in Wireless Multimedia Sensor Networks: Challenging and Defying in the IoT Era," in *IEEE Wireless Communications*, vol. 24, no. 5, pp. 126-131, October 2017, doi: 10.1109/WCM.2017.1700054.

[31] Y. A. Ur Rehman, M. Tariq and T. Sato, "A Novel Energy Efficient Object Detection and Image Transmission Approach for Wireless Multimedia Sensor Networks," in *IEEE Sensors Journal*, vol. 16, no. 15, pp. 5942-5949, Aug.1, 2016, doi: 10.1109/JSEN.2016.2574989.

[32] P. Chanak and I. Banerjee, "Congestion Free Routing Mechanism for IoT- Enabled Wireless Sensor Networks for Smart Healthcare Applications," in *IEEE Transactions on Consumer Electronics*, vol. 66, no. 3, pp. 223-232, Aug. 2020, doi: 10.1109/TCE.2020.2987433.

[33] M. I. Alipio and N. M. C. Tiglao, "RT-CaCC: A Reliable Transport with Cache-Aware Congestion Control Protocol in Wireless Sensor Networks," in *IEEE Transactions on Wireless Communications*, vol. 17, no. 7, pp. 4607-4619, July 2018, doi: 10.1109/TWC.2018.2827986.

[34] P. Chanak and I. Banerjee, "Congestion Free Routing Mechanism for IoT-Enabled Wireless

Sensor Networks for Smart Healthcare Applications," in *IEEE Transactions on Consumer Electronics*, vol. 66, no. 3, pp. 223-232, Aug. 2020, doi: 10.1109/TCE.2020.2987433.

[35] D. Lee and K. Chung, "Adaptive duty-cycle based congestion control for home automation networks," in IEEE Transactions on Consumer Electronics, vol. 56, no. 1,pp. 42-47, February 2010, doi: 10.1109/TCE.2010.5439124.

[36] M. C. Vuran and I. F. Akyildiz, "XLP: A Cross-Layer Protocol for Efficient Communication in Wireless Sensor Networks," in *IEEE Transactions on Mobile Computing*, vol. 9, no. 11, pp. 1578-1591, Nov. 2010, doi: 10.1109/TMC.2010.125.

[37] V. C. Gungor, Ö. B. Akan and I. F. Akyildiz, "A Real-Time and Reliable Transport (RT)2 Protocol for Wireless Sensor and Actor Networks," in *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 359-370, April 2008, doi: 10.1109/TNET.2007.900413.

[38] Z. Hu, X. Wang and Y. Bie, "Game Theory Based Congestion Control for Routing in Wireless Sensor Networks," in *IEEE Access*, vol. 9, pp. 103862-103874, 2021, doi: 10.1109/ACCESS.2021.3097942.

[39] J. Tan *et al*., "An Efficient Information Maximization Based Adaptive Congestion Control Scheme in Wireless Sensor Network," in *IEEE Access*, vol. 7, pp. 64878-64896, 2019, doi: 10.1109/ACCESS.2019.2915385.

[40] M. A. Kafi, D. Djenouri, J. Ben-Othman and N. Badache, "Congestion Control Protocols in Wireless Sensor Networks: A Survey," in *IEEE Communications Surveys & Tutorials*,vol. 16, no. 3, pp. 1369-1390, Third Quarter 2014, doi: 10.1109/SURV.2014.021714.00123.

[41] C. Sergiou, P. Antoniou and V. Vassiliou, "A Comprehensive Survey of Congestion Control Protocols in Wireless Sensor Networks," in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1839-1859, Fourthquarter 2014, doi: 10.1109/COMST.2014.2320071.

[42] L. Q. Tao and F. Q. Yu, "ECODA: enhanced congestion detection and avoidance for multiple class of traffic in sensor networks," in *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1387-1394, Aug. 2010, doi: 10.1109/TCE.2010.5606274.

[43] A. M. Ahmed and R. Paulus, "Congestion detection technique for multipath routing and load balancing in WSN," Wireless Networks, vol. 23, no. 3, pp. 881-888, 2017.

[44] N. Abbas, F. Yu, and Y. Fan, "Intelligent Video Surveillance Platform for Wireless Multimedia Sensor Networks," *Applied Sciences*, vol. 8, 2018. [SCI]

[45] N. Abbas and F. Yu*, "A comprehensive Analysis of End-to-End Delay for Wireless Multimedia Sensor Networks" Journal of Electrical Engineering & Technology, 2018; 13(6):

[46] N. Abbas and F. Yu*, "A Traffic Congestion Control for Wireless Multimedia Sensor Networks", in Proc. IEEE SENSORS'18, Oct. 27-30 2018, New Delhi, India.

[47] N. Abbas, F. Yu, and U. Majeed, "Reliability and End-to-End Delay Evaluation of Outdoor and Indoor Environments for Wireless Multimedia Sensor Networks," in *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2018, pp. 764-768.

[48] N. Abbas and F. Yu, "Performance analysis of end-to-end delay and reliability for linear wireless multimedia sensor networks," in *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2017, pp. 554-558.

[49] N. Abbas and F. Yu*, " Design and Implementation of a Video Surveillance System for Linear Wireless Multimedia Sensor Networks" In IEEE international Conference on Image, Vision and Computing (ICIVC), 2018, Chongqing, China.

[50] J. Horneber and A. Hergenröder, "A Survey on Testbeds and Experimentation Environments for Wireless Sensor Networks," *IEEE Communications Surveys & Tutorials*, vol. 16, pp. 1820-1838, 2014.

[51] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, and S.-S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," Multimedia Syst., vol. 21, pp. 49-60, 2015.

[52] L. Tshiningayamwe, G.-A. Lusilao-Zodi, and M. E. Dlodlo, "A Priority Rate-Based Routing Protocol for Wireless Multimedia Sensor Networks," in *Advances in Nature and Biologically Inspired Computing*, Cham, 2016, pp. 347-358.

[53] L. Guohua, L. Jianzhong, and Y. Bo, "Lower bound of weighted fairness guaranteed congestion control protocol for WSNs," in 2012 Proceedings IEEE INFOCOM, 2012, pp. 3046-3050.

[54] S. Hua, "Congestion control based on reliable transmission in wireless sensor networks," Journal of Networks, vol. 9, no. 3, pp. 762– 768, 2014.

[55] M. A. Jan, P. Nanda, X. He, and R. P. Liu, "PASCCC: Priority-based application-specific congestion control clustering protocol," *Computer Networks*, vol. 74, pp. 92-102, 2014/12/09/ 2014.

[56] H. Ahmed, M.J. Arshad, "Buffer Occupancy-Based Transport to Reduce Flow Completion Time of Short Flows in Data Center Networks," *Symmetry* **2019**, *11*, 646. *https://doi.org/10.3390/sym11050646*.

[57] V. Srivastava, S. Tripathi, K. Singh, L.H. Son "Energy efficient optimized rate based congestion control routing in wireless sensor network,". *J. Ambient Intell. Humaniz. Comput.* 2019, *11*, 1325–1338. *https://doi.org/10.1007/s12652-019-01449-1*.

[58] S. Javaid, H. Fahim, Z. Hamid, F.B. Hussain, "Traffic-aware congestion control (TACC) for wireless multimedia sensor networks," Multimed. Tools Appl. 2018, 77, 4433–44522. [58]R. Kumar, V. Tiwari, "Opt-ACM: An Optimized load balancing based Admission Control

Mechanism for Software Defined Hybrid Wireless based IoT (SDHW-IoT) network, Comput.Netw.2021, 188, 107888. *https://doi.org/10.1016/j.comnet.2021.107888*.

[59] X.W. Liu, *J.* Cai, Q. Chen, "Wang, Y. DRL-R: Deep reinforcement learning approach for intelligent rout-ing in software-defined data-center networks," *J. Netw. Comput. Appl.* **2021**, *177*, 102865.

[60] Y. Ghaemi, H. El-Ocla, N.R. Yadav, M.R. Madana, D.K. Raju, V. Dhanabal, V. Sheshadri, "Intelligent Transport System Using Time Delay-Based Multipath Routing Protocol for Vehicular Ad Hoc Networks," Sensors 2021, 21, 7706.

[61] J. Zhou, Z. Lin, X. Jiang, "Secure load-balanced scheme for cluster-based WSNs," In Proceedings of the 2019 6th International Conference on Systems and Informatics (ICSAI), Shanghai, China, 2–4 November 2019; pp. 804-809.

[62] C. Sergiou and V. Vassiliou, "HRTC: A Hybrid Algorithm for Efficient Congestion Control in Wireless Sensor Networks," in 2014 6th International Conference on New Technologies, Mobility and Security (NTMS), 2014, pp. 1-5.

[63] A. A. Rezaee, M. H. Yaghmaee, A. M. Rahmani, and A. H. Mohajerzadeh, "HOCA: Healthcare Aware Optimized Congestion Avoidance and control protocol for wireless sensor networks," Journal of Network and Computer Applications, vol. 37, pp. 216-228, 2014/01/01/ 2014.

[64] J. SAYYADA, N.K. Choudhari, "Hierarchical tree-based congestion control using fuzzy logic for heterogeneous traffic in wsn," International Journal of Current Engineering and Technology, vol. 4, no.3, pp. 4136–4143, 2014.

[65] Tshilongamulenzhe, T.M.; Mathonsi, T.E.; Mphahlele, M.I.; Plessis, D.D. Traffic-based Congestion Management algorithm for Wireless Sensor networks. In Proceedings of the 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 16–18 December 2020; pp. 202–207.

[66] Yadav, S.L.; Ujjwal, R.; Kumar, S.; Kaiwartya, O.; Kumar, M.; Kashyap, P.K. Traffic and energy aware optimization for congestion control in next generation wireless sensor networks. J. Sens. 2021, 2021, 5575802.

[67] Y. Zhou, W. Xiang, and G. Wang, "Frame Loss Concealment for Multiview Video Transmission Over Wireless Multimedia Sensor Networks," *IEEE Sensors Journal*, vol. 15, pp. 1892-1901, 2015.

[68] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," in *SenSys '03: Proceedings of the 1st international Conference on Embedded Networked Sensor Systems*, 2003, pp. 266–279. [Online].

[69] C. T. Ee and R. Bajcsy, "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks," in *SenSys '04: Proceedings of the 2nd international Conference on Embedded*

Networked Sensor Systems, 2004, pp. 148–161.

[70] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating Congestion in Wireless Sensor Networks," in *SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, 2004, pp. 134–147.

[71] K. Karenos, V. Kalogeraki, and S. V. Krishnamurthy, "Cluster-Based Congestion Control for Supporting Multiple Classes of Traffic in Sensor Networks," in *EmNets '05: Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors*, 2005, pp. 107–114.

[72] C. Wang, K. Sohraby and B. Li, "SenTCP: A Hop-by-hop Congestion Control Protocol for Wireless Sensor Networks," In Proceedings of IEEE INFOCOM 2005 (Poster Paper), Miami, Florida, USA, Mar. 2005.

[73] L. Popa, C. Raiciu, I. Stoica, and D. S. Rosenblum, "Reducing Congestion Effects in Wireless Networks by Multipath Routing," in *ICNP*. IEEE Computer Society, 2006, pp. 96–105. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/ICNP.2006.320202

[74] J. Kang, Y. Zhang and B. Nath, "TARA: Topology-Aware Resource Adaptation to Alleviate Congestion in Sensor Networks," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, pp. 919-931, July 2007, doi: 10.1109/TPDS.2007.1030.

[75] D. Patil and S. N. Dhage, "Priority-based Congestion Control Protocol (PCCP) for controlling upstream congestion in Wireless Sensor Network," *2012 International Conference on Communication, Information & Computing Technology (ICCICT)*, Mumbai, India, 2012, pp. 1-6, doi: 10.1109/ICCICT.2012.6398232.

[76] C. Sergiou, V. Vassiliou, and A. Paphitis, "Hierarchical Tree Alternative Path (HTAP) algorithm for congestion control in wireless sensor networks," Ad Hoc Netw., vol. 11, pp. 257-272, 2013.

[77] R. Vedantham, R. Sivakumar, and S. J. Park, "Sink-to-sensors congestion control," in *Proceedings of ACM (MobiHoc '04)*, adhoc networks, Year 2007 vol 5 number 4 page 462-485.

[78] C. Wang, B. Li, K. Sohraby, M. Daneshmand, and Y. Hu, "Upstream Congestion Control in Wireless Sensor Networks through Cross-Layer Optimization," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 786 –795, may 2007.

[79] X. Yin, X. Zhou, R. Huang, Y. Fang and S. Li, "A Fairness-Aware Congestion Control Scheme in Wireless Sensor Networks," in *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 5225-5234, Nov. 2009, doi: 10.1109/TVT.2009.2027022.

[80] W.-W. Fang, J.-M. Chen, L. Shu, T.-S. Chu, and D.-P. Qian, "Congestion avoidance, detection and alleviation in wireless sensor networks," *Journal of Zhejiang University: Science C*, vol. 11, no. 1, pp. 63–73, 2010.

[81] C. Sergiou and V. Vassiliou, "DAlPaS: A performance aware congestion control algorithm in Wireless Sensor Networks," in 2011 18th International Conference on Telecommunications,

2011, pp. 167-173.

[82] C.-Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, “Siphon: overload traffic management using multiradio virtual sinks in sensor networks,” in *Proceedings of the 3rd ACM International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 116–129, San Diego, Calif, USA, November 2005.

[83] S. Chen and N. Yang, “Congestion Avoidance Based on Lightweight Buffer Management in Sensor Networks,” *IEEE Transactions on Parallel and Distributed Systems,* vol. 17, no. 9, pp. 934–946, 2006.

[84] K. Karenos and V. Kalogeraki, “Facilitating Congestion Avoidance in Sensor Networks with a Mobile Sink,” in IEEE Real Time Systems Symposium (RTSS), 2007, pp. 321– 332.

[85] M. M. Alam and C. S. Hong, “Buffer and Rate Control Based Congestion Avoidance in Wireless Sensor Networks,” in *KIPS,* May 2007, pp. 1291–1293.

[86] M. I. Khan, W. N. Gansterer, and G. Haring, “Congestion Avoidance and Energy Efficient Routing Protocol for Wireless Sensor Networks with a Mobile Sink,” *JNW*, vol. 2, no. 6, pp. 42–49, 2007.

[87] F. Ren, T. He, S. K. Das and C. Lin, "Traffic-Aware Dynamic Routing to Alleviate Congestion in Wireless Sensor Networks," in IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 9, pp. 1585-1599, Sept. 2011, doi: 10.1109/TPDS.2011.24.

[88] Y.-P. Hsu and K.-T. Feng, "Cross-layer routing for congestion control in wireless sensor networks," in Radio and Wireless Symposium, 2008 IEEE, IEEE, Ed. Orlando, FL, 2008, pp. 783-786.

[89] G. Rajsekar, M. Mathew, N. Dineshraj, S. Barath, M. Sudha, and M. Valarmathi, “Collision Avoidance Scheme in Energy Constrained Wireless Sensor Networks using MAC Protocol,” in International Conference on Computing, Communication and Networking (ICCCN), 2008, pp. 1 –4.

[90] J.-M. Huang, C.-Y. Li, and K.-H. Chen, “TALONet: A Power-Efficient Grid-Based Congestion Avoidance Scheme Using Multi-Detouring Technique in Wireless Sensor Networks,” in Wireless Telecommunications Symposium, 2009, pp. 1 –6.

[91] P. Antoniou, A. Pitsillides, A. Engelbrecht and T. Blackwell, "Mimicking the bird flocking behavior for controlling congestion in sensor networks," 2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010), Rome, Italy, 2010, pp. 1-7, doi: 10.1109/ISABEL.2010.5702785.

[92] P. Antoniou and A. Pitsillides, “A Bio-Inspired Approach for Streaming Applications in Wireless Sensor Networks Based on the Lotka-Volterra Competition Model,” Comput. Commun., vol. 33, pp. 2039–2047, November 2010. [Online]. Available:

http://dx.doi.org/10.1016/j.comcom.2010.07.020

[93] O. B. Akan and I. F. Akyildiz, "Event-to-sink reliable transport in wireless sensor networks," in IEEE/ACM Transactions on Networking, vol. 13, no. 5, pp. 1003-1016, Oct. 2005, doi: 10.1109/TNET.2005.857076.

[94] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva, "Directed diffusion for wireless sensor networking," in IEEE/ACM Transactions on Networking, vol. 11, no. 1, pp. 2-16, Feb. 2003, doi: 10.1109/TNET.2002.808417.

[95] S.-J. Park, R. Vedantham, R. Sivakumar, and I. F. Akyildiz, "A Scalable Approach for Reliable Downstream Data Delivery in Wireless Sensor Networks," in MobiHoc '04: Proceedings of the 5th ACM International Symposium on Mobile Ad-Hoc Networking and Computing, 2004, pp. 78–89.

[96] Y. G. Iyer, S. Gandham and S. Venkatesan, "STCP: a generic transport layer protocol for wireless sensor networks," Proceedings. 14th International Conference on Computer Communications and Networks, 2005. ICCCN 2005., San Diego, CA, USA, 2005, pp. 449-454, doi: 10.1109/ICCCN.2005.1523908.

[97] J. Paek and R. Govindan, "RCRT: Rate-controlled reliable transport protocol for wireless sensor networks," ACM Trans. Sen. Netw., vol. 7, pp. 1-45, 2010.

[98] J. Paek and R. Govindan, "RCRT: rate-controlled reliable transport for wireless sensor networks," presented at the Proceedings of the 5th international conference on Embedded networked sensor systems, Sydney, Australia, 2007.

[99] Kumar, S.S., & Karthikeyan, S. (2024). Improving Quality of Service (QoS) in Wireless Multimedia Sensor Networks. *Measurement Science Review*, 24(4), 113-117.

[100] Y.-M. Liu and X.-H. Jiang, "An Extended DCCP Congestion Control in Wireless Sensor Networks," in International Workshop on Intelligent Systems and Applications, 2009, pp. 1 –4.

[101] M.P. Shelke, A. Malhotra, P. Mahalle "packet priority intimation-based data transmission for congestion free traffic management in wireless sensor networks," Computers & Electrical Engineering, 64, 248-261, 2017.

[102] Adil, M. Congestion free opportunistic multipath routing load balancing scheme for Internet of Things (IoT). Computer Networks 2021, 184, 107707.