Improving multicast in 802.11s mesh networks



Maria João Marques Pais
Master in Network and Information Systems Engineering
Department of Computer Science
Faculty of Sciences of University of Porto
2025

Improving multicast in 802.11s mesh networks

Maria João Marques Pais

Dissertation carried out as part of the Master in Network and Information Systems Engineering

Department of Computer Science

2025

Supervisor

Prof. Rui Prior, Professor Auxiliar, Faculty of Sciences of the University of Porto

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Rui Prior, for his guidance, availability, and support throughout the development of this dissertation.

I am deeply grateful to my family for their constant encouragement and unwavering support, without which the completion of this work would not have been possible.

Finally, I wish to thank my friends, colleagues, and professors for their contributions to both my personal and professional growth over the course of my academic journey.

Resumo

Este trabalho aborda a questão da eficiência inadequada do suporte *multicast* em redes mesh IEEE 802.11s. O *multicast* é um modo de comunicação relativamente negligenciado neste tipo de redes, implementado por padrão, de acordo com a norma, utilizando um mecanismo de inundação controlada, tal como o *broadcast*. Isto tem os efeitos indesejáveis de congestionar redes densamente povoadas com transmissões redundantes e de enviar pacotes *multicast* para zonas da rede sem recetores interessados. Para mitigar essas deficiências, propomos um novo mecanismo baseado no princípio de *flood-and-prune*, que melhora o desempenho *multicast* reduzindo seletivamente o encaminhamento desnecessário de pacotes, melhorando assim a eficiência e a escalabilidade geral da rede. Implementámos o mecanismo no simulador ns-3 e realizámos uma série de experiências para avaliar o seu desempenho e compará-lo com o mecanismo de inundação controlada de referência. Os resultados mostram que o mecanismo de *prune* reduz significativamente a ocupação do canal e as retransmissões redundantes, levando a uma redução da latência média e a uma melhoria na taxa de entrega de pacotes, particularmente em cenários congestionados e de alta densidade. Além disso, mitiga o impacto negativo do *multicast* nas transmissões *unicast* simultâneas, melhorando a fiabilidade geral da comunicação.

Keywords: network, mesh, multicast, ns3, 802.11s, flood-and-prune

Abstract

This work addresses the issue of inadequate efficiency of multicast support in IEEE 802.11s mesh networks. Multicast is a relatively overlooked mode of communication in this type of networks, implemented by default, as per the standard, using a controlled flooding mechanism, just like broadcast. This has the undesired effects of congesting densely packed networks with redundant transmissions and of sending multicast packets to zones of the network with no interested receivers. To mitigate these shortcomings, we propose a new mechanism based on the flood-and-prune principle, which enhances multicast performance by selectively reducing unnecessary packet forwarding, thereby improving the overall network efficiency and scalability. We implemented the mechanism in the ns-3 simulator and performed a series of experiments to evaluate its performance and compare it to the baseline controlled flooding. The results show that the pruning mechanism significantly reduces channel occupancy and redundant retransmissions, leading to reduced average latency and enhanced packet delivery ratio, particularly in congested and high-density scenarios. Furthermore, it mitigates the negative impact of multicast on concurrent unicast transmissions, enhancing overall communication reliability.

Keywords: network, mesh, multicast, ns3, 802.11s, flood-and-prune

Table of Contents

List of Tables vi
List of Figures vii
List of Abbreviations ix

1. Introduction 1
   1.1. Aims and Objectives 2
   1.2. Organization 3
2. Background 4
   2.1. Mesh networks 4
      2.1.1. Architecture 5
      2.1.2. Characteristics 7
   2.2. 802.11s 8
   2.3. Multicast Routing Protocols for Ad Hoc Networks/ WMN 10
   2.4. Multicast Service Models 14
   2.5. HWMP 15
      2.5.1. On-demand Mode 17
      2.5.2. Proactive Mode 17
      2.5.3. Multicast HWMP 17
   2.6. Multicast in cabled networks — PIM Dense Mode 18
3. State of the Art 20
   3.1. Flooding-Based Approaches 20
   3.2. Tree-Based/Mesh-Based Protocols 21
   3.3. Network Coding and Multi-Layer Multicast 22
   3.4. HWMP-Based Enhancements 24
   3.5. Opportunistic and Resource-Aware Approaches 25
   3.6. Comparative Summary Table 27
4. Design and Implementation 29
   4.1. Design 29
   4.2. Implementation 31
      4.2.1. Integration with the Peer Management Protocol 32
      4.2.2. TTL Handling and Node Position Estimation 32
      4.2.3. Duplicate Packet Issue and Improved Detection 32
      4.2.4. EWMA-Based Node Position Estimation 34
      4.2.5. Implementation Prune Messages 34
      4.2.6. Soft-State Prune Table 34

4.2.7. StartPrune Routine and TTL Processing 35
4.2.8. Prune Decision Logic 36
4.2.9. Integration with Route Request Logic 38
5. Experiments 39
5.1. Validation experiments 41
5.2. Evaluation experiments 46
5.2.1. Topologies with 10 nodes 49
5.2.2. Topologies with 30 nodes 53
5.2.3. Multicast and Unicast interference 64
5.3. Conclusion 66
6. Conclusion 68
6.1. Difficulties 69
6.2. Future Work 70

Bibliography 73

List of Tables

2.1. Summary Table of Routing Protocols 13

3.1. Comparison of articles and multicast approaches in WMN 28

4.1. Metadata used in the PRUNE message. 34

List of Figures

2.1. Infrastructure/Backbone WMN.[2] 6
2.2. Client WMN.[2] 6
2.3. Hybrid WMN.[2] 7
2.4. 802.11 header [5] 10
2.5. a) On-demand mode. b) Proactive mode. [12] 16

4.1. α = weight parameter, θ_t = Value of the series in the current period i, W_i = EWMA for i (with $W_0 = firstTTL$). 30
4.2. a) Prune only to nodes above. b) Prune to nodes above and the same level. 37

5.1. Validation Experiment 1 43
5.2. Validation Experiment 2 44
5.3. Validation Experiment 3 45
5.4. Validation Experiment 4 46
5.5. Distribution of 10 nodes in 100 m$\times$100 m. 48
5.6. Distribution of 30 nodes in 100 m$\times$100 m. 49
5.7. Distribution of 30 nodes in 300 m$\times$300 m. 49
5.8. Global Channel Occupancy and Global PDR for 10N, 2R and high traffic 50
5.9. Global E2E Delay for 10N, 2R and high traffic 50
5.10 Global Channel Occupancy and Global PDR for 10N, 2R and low traffic 51
5.11 Global E2E Delay for 10N, 2R and low traffic 51
5.12 Global Channel Occupancy and Global PDR for 10N, 2R and high traffic 52
5.13 Global E2E Delay for 10N, 2R and high traffic 52
5.14 Global Channel Occupancy and Global PDR for 10N, 5R and low traffic 53
5.15 Global E2E Delay for 10N, 5R and low traffic 53
5.16 Global Channel Occupancy and Global PDR for 30N, 5R and high traffic 54
5.17 Global E2E Delay for 30N, 5R and high traffic 54
5.18 PDR for each receiving node with flood-and-prune protocol 55
5.19 PDR for each receiving node with controlled flooding protocol 55
5.20 Global Channel Occupancy and Global PDR for 30N, 5R, high traffic and area 300 m$\times$300 m 56
5.21 Global E2E Delay for 30N, 5R, high traffic and area 300 m$\times$300 m 56
5.22 Topology of 30 nodes in an area of 300 m$\times$300 m 57
5.23 PDR for each receiving node with flood-and-prune protocol 57
5.24 PDR for each receiving node with controlled flooding protocol 58
5.25 Global Channel Occupancy and Global PDR for 30N, 5R and low traffic 59
5.26 Global E2E Delay for 30N, 5R and low traffic 59
5.27 Global Channel Occupancy and Global PDR for 30N, 15R and high traffic 60
5.28 Global E2E Delay for 30N, 15R and high traffic 60
5.29 PDR for each receiving node with flood-and-prune protocol 61
5.30 PDR for each receiving node with controlled flooding protocol 61
5.31 Global Channel Occupancy and Global PDR for 30N, 15R, high traffic and area 300 m$\times$300 m 62
5.32 Global E2E Delay for 30N, 15R, high traffic and area 300 m$\times$300 m 62
5.33 PDR for each receiving node with flood-and-prune protocol 63

5.34.PDR for each receiving node with controlled flooding protocol 63
5.35Global Channel Occupancy and Global PDR for 30N, 15R and low traffic 64
5.36Global E2E Delay for 30N, 15R and low traffic 64
5.37Multicast and Unicast interference 65

List of Abbreviations

AP Access Point. 5, 9

ASM Any-Source Multicast. 14, 15, 30

EWMA Exponentially Weighted Moving Average. iv, vii, 29– 31, 34, 35, 38, 42, 45, 46, 71

HWMP Hybrid Wireless Mesh Protocol. iv, 2– 4, 9, 15– 18, 20, 24, 25, 31, 32, 35, 39, 68, 70

MAC Medium Access Control. 5

MP Mesh Point. 15, 17

PDR Packet Delivery Ratio. vii, viii, 39, 41, 49– 69

PHY Physical layer. 5

PIM Protocol-Independent Multicast. 18

QoS Quality of Service. 22, 25, 26, 28

SSM Source-Specific Multicast. 14, 15, 30

STAs Stations. 5, 9, 16, 24

TTL Time To Live. 29– 32, 34, 35, 37, 44, 46, 71

WLAN Wireless Local Area Network. 5, 8, 9

WMN Wireless Mesh Network. iv, vi, vii, 1, 4– 8, 10, 20, 21, 23– 28

1. Introduction

Wireless communication technologies have become the foundation of modern connection, enabling a wide range of applications that demand dependable and scalable data transmission. Wireless Mesh Network (WMN) have emerged as a possible approach for creating adaptable, decentralized, and self-healing communication infrastructures. Mesh networks, which interconnect nodes in a multi-hop configuration, eliminate the need for centralized infrastructure, making them ideal for applications such as community networks, industrial automation, and emergency communications systems.

In this context, multicast communication fulfills a crucial function in the effective distribution of data to numerous recipients. Multicast is a transmission paradigm that facilitates the delivery of a singular data stream to multiple recipients concurrently. This approach is distinct from unicast, wherein the sender is obligated to transmit a distinct copy of the data to each receiver, and from broadcast, which delivers data to all nodes in the network irrespective of their interest. The transmission of data once for multiple receivers is at the core of multicast's functionality. This approach has been shown to reduce bandwidth consumption, lower transmission overhead, and improve scalability.

The significance of multicast is further accentuated in mesh networks. Mesh networks confront particular challenges when it comes to efficient group communication due to their multi-hop architecture. To create multicast utilizing unicast transmissions, the source must replicate and send the same packet to each destination node separately. This strategy would substantially increase network load and energy consumption. In contrast, broadcast-based techniques frequently result in data being transmitted to nodes that are not part of the multicast group. Multicast strikes a balance between the two approaches, allowing for the efficient and targeted distribution of information across the mesh.

Multicast communication has been demonstrated to have numerous practical applications in mesh networks. In the context of smart city scenarios, multicast communication can be utilised for the dissemination of sensor readings, software updates, or control commands to groups of devices. In the context of public safety networks, multicast facilitates the concurrent dissemination of critical alerts and coordination messages to first responders. Multimedia applications such as live video streaming, online education, and teleconferencing also benefit from multicast, as they often involve delivering the same data stream

to multiple participants. Furthermore, collaborative and distributed computing applications depend on multicast to ensure the consistency of the state among the participating nodes.

Although multicast has advantages in IEEE 802.11s mesh networks, the current technology is limited. The standard's default routing mechanism, Hybrid Wireless Mesh Protocol (HWMP), handles multicast and broadcast traffic via network-wide controlled flooding. In this approach, the sender sends a multicast packet, which is then re-broadcast once by each mesh node until it reaches all network nodes. While flooding ensures delivery in small and moderate-sized networks, it has substantial scaling difficulties.

The fundamental downside of flooding-based multicast is inefficiency. It is clear that as each node retransmits every multicast packet, the number of transmissions grows with network size. This phenomenon is frequently known as the 'broadcast storm dilemma'. This behaviour has been proven to increase the likelihood of packet collisions, misuse important wireless bandwidth, and cause excessive energy expenditure. Furthermore, the unregulated nature of flooding has the ability to generate duplicate packets and increase end-to-end latency. In view of these limitations, enhancing multicast efficiency in 802.11s mesh networks remains a relevant research challenge.

In this thesis, we address these challenges by proposing and implementing a pruning mechanism for multicast traffic in IEEE 802.11s mesh networks. The suggested approach aims to reduce unnecessary retransmissions by allowing nodes to voluntarily stop sending multicast packets once it is judged that additional forwarding would not improve coverage. The suggested mechanism's effectiveness is assessed by simulations with the ns-3 network simulator, which provides insights into its performance under various topologies.

1.1. Aims and Objectives

This dissertation aims to find gaps in the HWMP protocol and improve multicast in 802.11s mesh networks, namely by implementing a broadcast-and-prune mechanism on the HWMP protocol.

In order to achieve the aims of this project, the work was divided into objectives to be accomplished:

- Research existing work and document it in the State of Art chapter (3);
- Design the mechanism to be implemented;
- Implement the chosen mechanism in ns-3;

- Conduct experiments to determine whether the mechanism used improves multicast and by how much.

1.2. Organization

This document is organised into five chapters, including this Introduction. Chapter 2 discusses related background knowledge on Mesh Networks, 802.11s, multicast routing protocols, and, specifically, the multicast mechanism of HWMP. Chapter 3 is focused on providing an overview of approaches employed by related works and research in enhancing multicast functionality. Chapter 4 details the design and implementation of the solution. Chapter 5 presents and discusses the results obtained in the experiments conducted with the new protocol. Chapter 6 concludes this dissertation, summarizing the main achievements, the difficulties encountered, and identifying possible paths for further development of this work.

2. Background

In this chapter, we provide essential background information required to understand the context and technologies explored in this research. Initially, section 2.1 focuses on mesh networks, detailing their architecture, operational principles, advantages, and typical application scenarios. Section 2.2 examines the relationship between IEEE 802.11 standards and multicast communication, highlighting the specific challenges and considerations when deploying multicast in wireless environments. In Section 2.3, various multicast routing protocols are reviewed, providing insights into their operational characteristics, strengths, and limitations. Finally, Section 2.4 presents an in-depth analysis of HWMP, which serves as the primary protocol under investigation in this study, discussing its fundamental operations, modes, and relevance to wireless mesh network performance.

2.1. Mesh networks

WMN have emerged as a promising solution for providing robust, flexible, and scalable wireless connectivity. Composed of static or mobile nodes that cooperatively forward data on behalf of others, WMNs are self-organizing and self-healing, which makes them highly resilient to node failures and dynamic network conditions. These characteristics render mesh networks particularly attractive for a wide range of applications, including community networking, military communications, emergency response, and smart cities. In general, a WMN can be described as a multi-hop communication network consisting of wireless nodes and links, and can also be regarded as a form of ad hoc network. The typical architecture of a WMN is composed of three key elements: gateways, mesh routers, and mesh clients. The gateway serves as the Internet access point and forms the backbone of the mesh network. Mesh routers operate as wireless access points that forward packets in a multi-hop fashion, ensuring communication even in hard-to-reach areas. Mesh clients, in turn, include mobile devices and end-user terminals that connect to the network via wired or wireless links.

In this context, multicasting emerges as an essential communication paradigm, where a single source transmits the same data simultaneously to a set of receivers. The development of wireless broadband technologies aimed at offering services comparable to those

of wired networks, providing support for both voice communications with mobility and high-bandwidth data transfer for multiple simultaneous users. Depending on coverage range, wireless networks are traditionally classified into:

- Wireless Local Area Network (WLAN): coverage of up to a few hundred meters, typically used in residential and office environments;
- WMAN (Wireless Metropolitan Area Network): coverage on a metropolitan scale, spanning entire cities;
- WWAN (Wireless Wide Area Network): large-scale networks with cell radio of up to 50 km, covering areas beyond a single city.

Among these categories, WLANs are the main focus of this study. At present, IEEE 802.11 is the most widely deployed WLAN technology, often referred to interchangeably with Wi-Fi. A typical WLAN consists of a Access Point (AP), user devices, antennas and a number of Stations (STAs) connected to it.

There are basically two modes in which communication can take place. In the centralized mode of communication, communication to/from a STAs is always carried over the APs. There is also a decentralized mode in which communication between two STAs can take place directly without the requirement of an AP, in an ad hoc fashion.

The standardization process highlighted significant challenges in the design of the Physical layer (PHY) and the Medium Access Control (MAC) layer. At the PHY, these challenges include ensuring mobility support, as WMN must be capable of accommodating user movement; enabling multiple transceivers, which allow simultaneous transmissions through multiple transmitters; and maintaining robust transceiver performance, requiring fast switching between transmit and receive modes along with accurate synchronization. At the MAC layer, one of the key difficulties lies in MAC protocol design, particularly for channel assignment, which improves network capacity by efficiently allocating channels for data transmission. Additionally, the use of smart antennas introduces complexity, as they must dynamically adjust their direction, communicate effectively with multiple neighbours, and track the position of mobile users. [1]

2.1.1 Architecture

According to [1], WMN architectures are grouped into three categories: infrastructure/backbone, client, and hybrid.

Infrastructure/backbone WMN (Fig. 2.1) consists of interconnected mesh routers that provide access to clients. In this architecture, wireless clients are linked to one of the mesh routers, and they do not forward traffic. The architecture can also be improved using wired nodes. Furthermore, mesh routers can serve as gateways to other networks, such as the Internet and sensor networks. Although the dominant 802.11 technology may be the apparent choice, any wireless technology can be utilized.

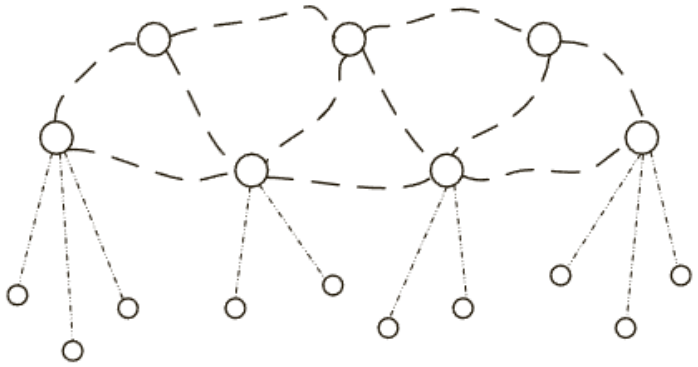


Figure 2.1: Infrastructure/Backbone WMN.[2]

In client WMN (Fig. 2.2), there is no mesh router involved, and the mesh clients create a peer-to-peer network, communicating directly. Client nodes not only serve end applications, but they also have the ability to route each other's traffic. However, because there is no viable connectivity to other networks, all generated traffic remains local.

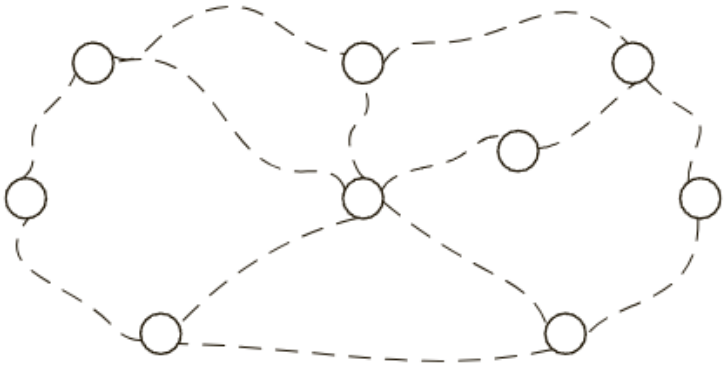


Figure 2.2: Client WMN.[2]

Hybrid WMN architecture is the combination of the previous two. Mesh routers create a backbone network, and mesh clients can connect directly to a mesh router or via another mesh client that routes traffic to the backbone network.

The second classification, in [3], also groups WMN in three architectures: flat, hierarchical, and hybrid. The flat architecture resembles the client WMN. In hierarchical WMNs,

which are similar to infrastructure WMN, a backbone is formed by mesh routers that only use 802.11 links. Hybrid architecture refers to hierarchical WMN that incorporates additional technologies, such as Worldwide Interoperability for Microwave Access, satellite, or cellular networks.

The definition of client/flat network architecture is essentially the same as the definition of MANETs. The hybrid architecture in Fig. 2.3 expands the WMN coverage, but it may introduce difficulties if both static mesh routers and MANET clients should route under the same protocol, as their characteristics of mobility, power consumption, and so on, can vary substantially.

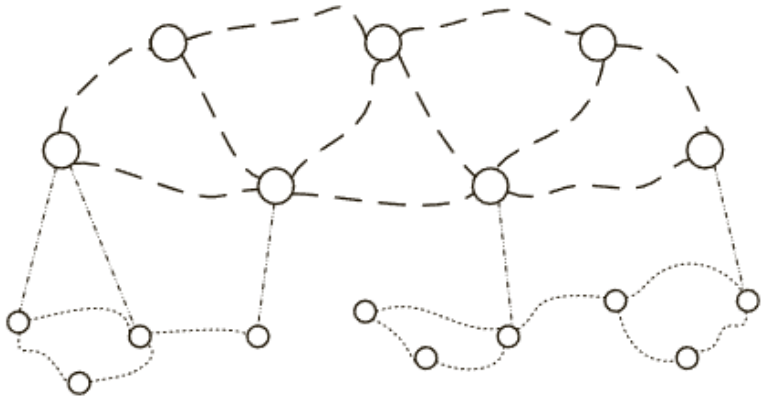


Figure 2.3: Hybrid WMN.[2]

2.1.2 Characteristics

According to the survey [1], the primary characteristics of WMN are as follows:

- **Dynamic self-configuration and self-organization:** One of the key features of WMN is their ability to self-configure and self-organize. This characteristic enables easy and rapid deployment, as nodes automatically establish connectivity without requiring extensive manual configuration;
- **Adaptation:** WMNs are capable of adapting to changes in their environment. In the event of a link failure, the network can automatically reroute data through alternative paths, thus maintaining service continuity. This dynamic routing capability is crucial for maintaining network reliability and ensuring consistent quality of service;
- **Fault tolerance and robustness:** Due to their inherent redundancy, mesh networks exhibit a high degree of fault tolerance and robustness. Multiple alternative paths

exist between any two nodes, which mitigates the impact of individual node or link failures;

- **Multi-hop wireless network:** The objective of developing WMNs is to extend the coverage range of existing wireless networks without compromising channel capacity. Another objective is to provide non-line-of-sight connectivity between users, thereby eliminating the need for direct line-of-sight links. The mesh-style multi-hopping method is thus imperative, as it reduces link distances, mitigates interference between nodes, improves frequency reuse efficiency, and enhances overall throughput;
- **Support for ad hoc networking and capability of self-forming, self-healing, and self-organization:** WMNs enhance network performance due to their flexible architecture, ease of deployment, fault tolerance, and mesh connectivity;
- **Mobility dependence:** While mesh routers typically have limited mobility, mesh clients can be either stationary or mobile. This distinction allows WMNs to support a wide range of use cases, from fixed wireless access to mobile user connectivity;
- **Multiple types of network access:** WMNs support multiple forms of network access. Additionally, WMNs can be seamlessly integrated with other wireless networks to provide end-user services.
- **Dependence of power consumption constraints on node type:** Mesh routers generally do not face strict power limitations, whereas mesh clients may require energy-efficient communication protocols;
- **Compatibility and interoperability with existing wireless networks:** WMNs are required to comply with existing IEEE 802.11 standards, ensuring that they support both mesh-capable and conventional Wi-Fi clients. Furthermore, they must be interoperable with other wireless and cellular networks to guarantee seamless communication across heterogeneous systems.

2.2. 802.11s

IEEE 802.11s is an extension of the IEEE 802.11 standard that defines mechanisms for wireless mesh networking. It specifies how wireless devices can interconnect to form a WLAN mesh, applicable both to static topologies and to ad-hoc deployments. The development of 802.11s began in 2003 as a Study Group within IEEE 802.11, later evolving

into a Task Group in 2004. A call for proposals in 2005 resulted in the submission of 15 candidate solutions. Through a series of evaluations, eliminations, and mergers, the proposals were ultimately consolidated into two, SEE-Mesh and Wi-Mesh, which merged into a joint proposal in 2006.

Devices operating under IEEE 802.11s are referred to as mesh stations. These devices establish mesh links with one another, forming the basis for mesh paths, which are maintained by routing protocols. The standard defines the HWMP as the default routing protocol. HWMP uses a common set of protocol elements, generation and processing rules inspired by Ad Hoc On-Demand Distance Vector (AODV) protocol, adapted for MAC address-based path selection and link metric awareness. Nevertheless, 802.11s remains flexible, allowing vendors to implement alternative routing protocols.

Mesh STAs, being independent devices, employ mesh services to communicate with other nodes in the WLAN. They may also be co-located with conventional IEEE 802.11 AP, extending WLAN access through the mesh. In addition, mesh STAs can be co-located with an IEEE 802.11 portal, acting as a gateway to external, non-802.11 networks. In such cases, the standard provides proxy mechanisms to enable address resolution and communication with non-mesh devices, ensuring transparency of external addressing within the mesh.

Beyond routing, IEEE 802.11s introduces support for deterministic medium access, congestion control, and power-saving mechanisms. Unlike traditional client–server models, mesh networks defined by 802.11s do not impose fixed roles, there are no dedicated initiators, responders, clients, or servers. Consequently, mesh security protocols must operate in a fully peer-to-peer manner, enabling either peer to initiate secure associations. To achieve this, the standard specifies the Simultaneous Authentication of Equals (SAE), a password-based key establishment protocol. SAE is derived from the Diffie–Hellman key exchange, which itself lacks inherent authentication. To address this limitation, SAE incorporates a pre-shared mesh password and the MAC addresses of both peers into the exchange. Authentication occurs only upon mutual discovery of peers. Upon successful SAE completion, each peer verifies that the other possesses the correct mesh password. This process not only authenticates the peers but also produces a cryptographically strong key. The derived key is then used within the Authenticated Mesh Peering Exchange to establish secure peerings and derive session keys that protect mesh traffic, including routing messages.

The 802.11 frame format (Fig. 2.4) provides four fields necessary for addressing over multiple intermediate devices. The source address indicates the station that generated the frame (initial hop), and the destination address indicates the intended receiver (final hop). Both addresses remain unchanged in a concatenated set of multiple wireless hops. The transmitting and receiving station addresses, which denote the stations that actually forwarded the frame, change with every hop. The mesh-specific fields are transported inside the Body field of the frame. [4]

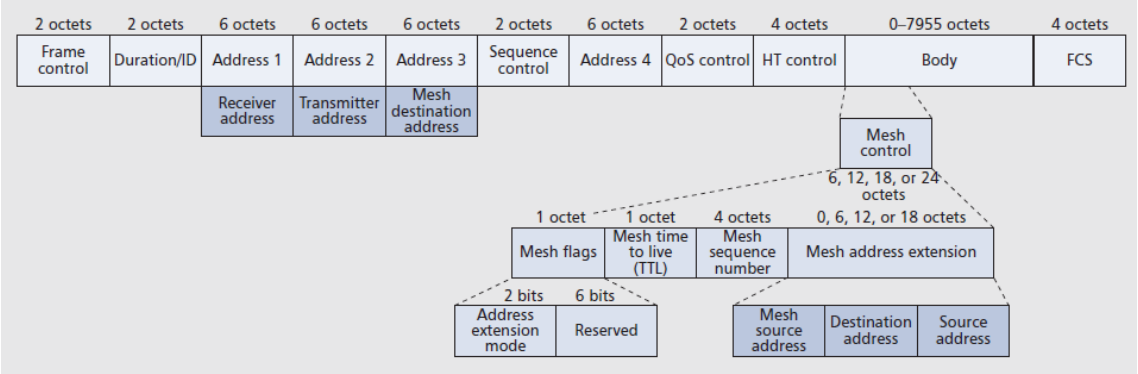


Figure 2.4: 802.11 header [5]

2.3. Multicast Routing Protocols for Ad Hoc Networks/ WMN

Multicast routing is a fundamental component of efficient communication in mobile networks, enabling mobile nodes to dynamically establish wireless connections with each other. Due to the frequent changes in topology, limited bandwidth and energy constraints of these networks, specialised multicast routing protocols are required.

The Adaptive Demand-Driven Multicast Routing (ADMR) protocol is a reactive, on-demand multicast routing protocol for Mobile Ad hoc Network (MANET) specifically designed to minimise proactive components and reduce control overhead, thereby improving protocol efficiency. This is achieved by using the acronym packet, which is expanded only on the first use. ADMR has been demonstrated to create and maintain per-source multicast forwarding trees exclusively in circumstances where data transmission is occurring. In such cases, forwarding packets are directed along minimum-delay paths, and rerouting is initiated around links that are temporarily non-operational. The elimination of periodic control messages, the avoidance of neighbour sensing, and the utilisation of adaptive state expiration based on application traffic patterns, supplemented by passive acknowledgments for reliability, are key features of the system. In the event of a sender becoming inactive, exponentially spaced keep-alives are transmitted prior to the silent expiry of the state without the execution of an explicit teardown. Disconnections are detected via missing data or

keep-alives, and local repair is triggered using RECONNECT messages or global rejoin procedures from receivers. In the context of high mobility, ADMR employs a strategy of temporarily flooding data packets to ensure delivery, subsequently reverting to tree-based forwarding once the topology stabilises. However, this approach compromises efficiency under constant mobility conditions. New sources employ a bootstrapping technique to establish multicast states via an initial network flood, followed by a buffer and subsequent transition to tree flooding upon receiver participation. The protocol also prunes unused branches automatically based on the absence of downstream acknowledgments, thereby reducing unnecessary forwarding. [6]

The On-Demand Multicast Routing Protocol (ODMRP) is a mesh-based, on-demand multicast routing protocol designed for mobile ad hoc networks (MANETs). ODMRP creates and maintains a separate mesh for each multicast group, this provides redundant paths and makes the network more resilient to changes in mobility and topology. The protocol can handle both unicast and multicast traffic and can also be used with any unicast protocol. When a source has data, it periodically sends JOIN QUERY messages, to which receivers respond with JOIN REPLY messages that propagate along reverse paths, thereby activating the forwarding group. These nodes only forward multicast data if packets are not duplicates, and state is maintained through a soft-state approach that expires unless refreshed. Reliability is ensured via passive acknowledgements, whereby nodes overhear next-hop transmissions. Retransmissions occur if acknowledgements are missing. While ODMRP achieves high delivery ratios and resilience in dynamic topologies, periodic flooding and redundant forwarding incur significant overhead, impacting scalability and energy efficiency, particularly in scenarios with many sources. However, it remains competitive with protocols such as ADMR in terms of packet delivery. [7]

The Destination-Sequenced Distance-Vector (DSDV) protocol is a proactive routing protocol designed for ad hoc networks, which lack fixed infrastructure and rely on dynamically changing topologies. The algorithm is based on the Bellman-Ford distance-vector algorithm, but introduces sequence numbers to prevent routing loops and the ”count-to-infinity” problem. Each node is responsible for maintaining a routing table that contains a comprehensive list of all reachable destinations, along with the respective hop counts, next hops, and destination-generated sequence numbers. Routing updates are periodically disseminated or initiated in response to substantial alterations in topology. In the event of an update, nodes demonstrate a preference for routes with newer sequence numbers. In the event of a tie, the route with the lowest metric is selected. In order to optimise

overhead, DSDV employs full dumps and incremental updates. The management of link breaks is achieved through the utilisation of an infinite metric and a new sequence number, which is expeditiously disseminated to other nodes. Despite the fact that DSDV ensures the absence of loops in routes, it is afflicted by elevated levels of control overhead, the possibility of fluctuations in routes, and restricted scalability. This is due to the necessity for each node to retain the complete network topology. Furthermore, the assumption of bidirectional links has the capacity to isolate nodes in real deployments. These limitations were the impetus for the development of more efficient, on-demand protocols. [8]

The Ad hoc On-Demand Distance Vector (AODV) protocol is a reactive routing algorithm designed for ad hoc networks, where nodes cooperate to establish communication without a centralised infrastructure. In contrast to proactive protocols such as DSDV, which rely on periodic updates to maintain complete routing tables, AODV only discovers routes when they are required. This approach reduces overhead and enhances scalability in dynamic topologies. AODV utilises destination sequence numbers, which have been inherited from DSDV, to ensure the absence of loops in route configuration. In the event of a source node requiring a path to a destination, it initiates a route discovery process by broadcasting a Route Request (RREQ) to its neighbours. Each intermediate node in the network functions as a recorder of the reverse route back to the source. Upon reaching either the destination or a node with a fresh route to it, the RREQ generates and transmits a Route Reply (RREP) along the reverse path, thereby establishing a forward route that the source can then utilise for data transmission. Each node maintains a routing table containing the next hop, hop count, destination sequence number, and expiration time for each active route. In instances where multiple routes lead to the same destination, the one with the most recent sequence number is favoured. If sequence numbers are equal, the route with the smallest hop count is selected. AODV incorporates mechanisms for route maintenance, whereby link breaks are detected using either HELLO messages or link-layer feedback. In the event of a break, affected nodes propagate a Route Error (RERR) message upstream, thereby invalidating the route. In the event that communication remains necessary, the source has the capacity to reinitiate the process of route discovery. [9]

The Multicast Ad-hoc On-Demand Distance Vector (MAODV) protocol is a reactive multicast routing protocol for mobile ad hoc networks (MANETs) that builds on the AODV protocol to support multicast communication through a shared-tree approach. MAODV dynamically builds and maintains a single multicast tree per group. It uses destination

sequence numbers, which are periodically updated and disseminated by a designated multicast group leader, to ensure loop-free and up-to-date routes. The first node to request a route to a multicast group becomes the group leader, responsible for generating sequence numbers. New members join the tree by selecting the optimal upstream node based on the freshest sequence number and minimum hop distance. This activates the chosen path while letting alternate routes expire. Leaf nodes can remove themselves from the tree when they are no longer interested, and broken links trigger downstream nodes to reinitiate the join process. If no response is received, a new leader may be elected. MAODV relies on periodic neighbour sensing and group HELLO messages to maintain multicast state, even in the absence of active senders. This introduces control overhead. Although MAODV offers efficient bandwidth usage, loop prevention and low state storage per group, it struggles with high mobility or heavy traffic due to increased control load and session initiation latency. Its single-path nature also lacks the redundancy of mesh-based protocols such as ODMRP. Its dependency on explicit prune messages also makes it vulnerable to stale forwarding state if such messages are lost. [10]

Protocol	Routing Type	Forwarding Strategy	Strengths	Weaknesses
ADMR	Reactive	Per-source multicast tree	Low overhead, adaptive state, tree pruning, no neighbour sensing	Efficiency drops under constant high mobility due to flooding
ODMRP	Reactive	Mesh per multicast group	High delivery ratio, robust under mobility, mesh resilience	Higher energy and bandwidth cost, redundant forwarding
DSDV	Proactive	Flat table-driven routing	Loop-free routes always available	High overhead, route fluctuations, poor for large networks
MAODV	Reactive	Shared multicast tree	Bandwidth-efficient, loop-free, low state storage	Control overhead under high mobility, no redundancy, prune loss can cause stale state

Table 2.1: Summary Table of Routing Protocols

2.4. Multicast Service Models

In addition to multicast routing protocols, it is essential to understand the multicast service models that define the relationship between senders and receivers. In the context of broadcasting, the sender disseminates a copy of each plot to all potential nodes within the network. This approach is regarded as the most elementary technique for ensuring that content is disseminated to as numerous destinations as possible. Multicast, on the other hand, is a more intricate process, as the sender is only responsible for transmitting frames to interested parties. Consequently, it is necessary for the sender to maintain a list of nodes to which these frames should be directed. The two most widely used paradigms are Any-Source Multicast (ASM) and Source-Specific Multicast (SSM), which can be used in wired and wireless networks. [11]

ASM is a traditional multicasting technique that allows consumers to receive data from many sources inside the same multicast group. The primary feature of ASM is its capacity to accommodate many sources, resulting in a flexible environment for data delivery. ASM settings do not require receivers to specify the source of the information they want to receive, which simplifies setup and operations to some extent.

SSM is a more advanced technique designed for cases in which data streams originate from certain sources. This paradigm improves the direct correlation between the sender and receiver, making SSM especially useful for applications like video conferences and live broadcasts when the source is known.

In SSM, only explicitly designated sources can distribute data to receivers who have explicitly joined the group and are aware of the data's provenance. This improves SSM security while reducing the complexity and network overhead involved with maintaining several possible sources.

When deciding whether to use ASM or SSM for multicast communication, it is critical to consider not only the operational feasibility of each approach but also the distinct advantages and potential drawbacks of each, as these factors can have a significant impact on network design and multimedia distribution strategies.

The ASM model's primary benefit is its ability to manage various sources, which provides a high degree of flexibility in scenarios with dynamic participants. Its compatibility with legacy multicast infrastructure makes integration possible without the need for significant

redesign. However, this flexibility adds management complexity and security issues because ASM must maintain data pathways from potentially numerous and unpredictable senders, raising the chance of receiving unwanted or malicious traffic and potentially causing network floods or inefficiencies.

SSM's source-specific architecture prioritizes efficiency by only establishing forwarding state for explicitly joined <S,G> pairs. This eliminates control overhead, lowers latency, and increases transmission efficiency. The security protocol has been strengthened to ensure that data transfer to the group is limited to approved sources. This measure is intended to reduce the possibility of multicast exploitation and improve overall security. Nonetheless, SSM's one-source-per-group requirement can limit its effectiveness in cases needing several concurrent sources or frequent source changes, and its implementation may necessitate updates to existing multicast infrastructure, increasing cost or complexity.

In conclusion, ASM and SSM are two complementary paradigms: ASM provides maximum flexibility and is appropriate for multi-source collaborative contexts, whereas SSM prioritizes efficiency, lower latency, and tighter source control, making it excellent for content distribution with known senders. The choice between these approaches is determined by a number of criteria, including network needs, security concerns, scalability goals, and the nature of the multicast application.

2.5. HWMP

The HWMP has combined the flavour of reactive and proactive routing strategy by employing both on-demand path selection mode and proactive tree building mode. On-demand mode allows two Mesh Point (MP) to communicate using peer-to-peer paths. This mode is mainly used by nodes that experience a changing environment and when there is no root MP configured. On the other hand, proactive tree building mode can be an efficient choice for nodes in a fixed network topology. In HWMP, the proactive mode is typically utilised by root nodes, nodes connected to the wired network that function as gateways to the Internet. Given the assumption that a considerable number of mesh nodes will communicate via these gateways, the proactive construction of a tree facilitates the network's utilisation of pre-established paths, characterised by minimal delay. The mandatory routing metric used in HWMP is the airtime cost metric that measures the link quality. In HWMP, both on-demand and proactive mode can be used simultaneously. [12]

All HWMP modes of operation share common processing rules and primitives. HWMP elements include the PREQ (path request), PREP (path reply), PERR (path error), and RANN (root announcement) elements. The metric cost of the links influences the paths HWMP creates. The PREQ, PREP, and RANN elements have a Metric field to transfer metric information between mesh STAs.

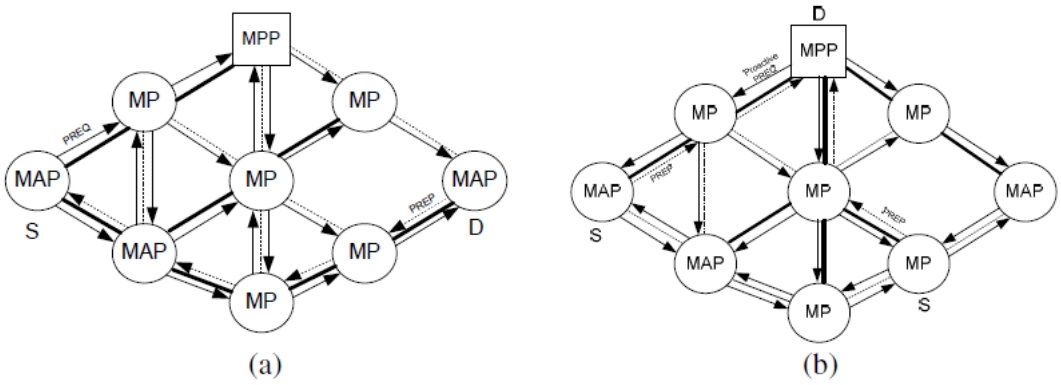


Figure 2.5: a) On-demand mode. b) Proactive mode. [12]

Path selection in HWMP employs a sequence number approach that allows mesh STAs to distinguish between current and stale path information at all times, ensuring loop-free connectivity. Each mesh STAs has its own HWMP sequence number (HWMP SN), which is communicated to other mesh STAs in the HWMP elements.

HWMP uses sequence numbers to prevent the creation of path loops and to distinguish stale and fresh path information. Each mesh STAs keeps its own HWMP SN that it increments, uses in HWMP elements, and processes according to the HWMP rules.

HWMP depends on each mesh STAs in the network to own and maintain its HWMP SN to guarantee the loop-freedom of all paths toward that mesh STAs. A mesh STAs increments its own HWMP SN in the following two circumstances:

- If it is an originator mesh STAs, it shall increment its own HWMP SN immediately before it starts a path discovery;
- If it is a target mesh STAs, it shall update its own HWMP SN to maximum (current HWMP SN, target HWMP SN in the PREQ element) + 1 immediately before it generates a PREP element in response to a PREQ element.

2.5.1 On-demand Mode

In an On-demand mode, a source MP broadcast path request (PREQ) message requesting a route to the destination. The PREQ is processed and forwarded by all intermediate MPs and sets up the reverse path from the destination to the source of route discovery. The destination MP or any intermediate MP with a path to the destination may unicast a path reply (PREP) to the source MP that creates the forward path to the destination. As shown in Fig. 2.5 a), source MAP S broadcasts PREQ message and intermediate nodes re-broadcast PREQ after updating their path to the source as shown in solid lines. The destination MAP D, unicast a PREP message to the source MAP S using the reverse path shown in solid lines.

2.5.2 Proactive Mode

In Proactive Tree Building mode, the MP that is configured as a root MP (MPP) can initiate the route discovery process in one of two alternative ways.

In the first approach, it announces its presence by periodically broadcasting a root announcement RANN message that propagates metric information across the network. Upon reception of a RANN message, an MP that has to create or refresh a path to the root MP sends a unicast PREQ to the root MP. The root MP then unicast a PREP in response to each PREQ. The unicast PREQ creates the reverse path from the root MP to the originating MP, while the PREP creates the forward path from the MP to the root MP.

In the second approach, the root MP proactively disseminates a proactive PREQ message to all the MPs in the networks with the intent to establish a path as shown in Fig. 2.5 b). An MP after receiving a proactive PREQ, creates or updates its path to the root MP by unicasting a proactive PREP, if and only if the PREQ contains a greater sequence number, or the sequence number is the same as the current path and the PREQ offers a better metric than the current path to the root MP. Thus, a routing tree is created with the MPP being the root of the tree.

2.5.3 Multicast HWMP

HWMP does not natively support multicast, instead the multicast packets are handled in the same way of a broadcast packet.

Although broadcast in HWMP is necessary for network management and route discovery, it depends on simple flooding, which, if left unchecked, can result in redundancy and network congestion. HWMP uses controlled flooding as its method of operation. After a brief random delay, every node always passes a received broadcast packet to all of its neighbours for the first time. These neighbours then rebroadcast the packet, spreading the message across the mesh. Any duplicate packets are thrown out right away. The fact that every node takes part in the rebroadcasting operation allows this approach to achieve a high reachability. Controlled flooding, on the other hand, might result in a significant and intense contention, raise the possibility of collisions, and produce a large number of redundant packets. Another problem is that this type of flooding does not provide any kind of feedback, which results in a lack of adaptation in the transmission rate and an inferior user experience compared to unicast.

2.6. Multicast in cabled networks — PIM Dense Mode

Although Protocol-Independent Multicast (PIM) DM is used in cable networks, it contains the core ideas of the mechanism implemented.

PIM protocol is a multicast transmission protocol that can transmit multicast data on existing IP networks and is independent of routing protocols. It can work in four modes: dense mode (PIM-DM), sparse mode (PIM-SM), Bidir- PIM, and source-specific mode (PIM-SSM). PIM-DM sends multicast packets to all routers and network ports as if every connected device wishes to receive them. Routers with no multicast members connected to them will later send prune messages to avoid receiving unnecessary traffic.

PIM-DM is a multicast routing protocol that uses the underlying unicast routing information base to flood multicast datagrams to all multicast routers. Prune messages are used to prevent future messages from propagating to routers without group membership information.

PIM-DM assumes that when a source starts sending, all downstream systems want to receive multicast datagrams. Initially, multicast datagrams are flooded to all areas of the network. PIM-DM uses RPF to prevent looping of multicast datagrams while flooding. If some areas of the network do not have group members, PIM-DM will prune off the forwarding branch by instantiating the prune state. The Prune state has a finite lifetime. When that lifetime expires, data will again be forwarded down the previously pruned branch. Prune state is associated with an (S,G) pair. When a new member for a group G appears

in a pruned area, a router can ”graft” toward the source S for the group, thereby turning the pruned branch back into a forwarding branch. [13]

3. State of the Art

Multicast routing plays a crucial role in efficiently delivering data from a single source to multiple destinations, particularly in wireless mesh networks and other bandwidth-constrained environments. While IEEE 802.11s provides a standardized framework for mesh networking, it lacks robust support for multicast operations, prompting a significant body of research to address this limitation. This chapter presents a review of the main approaches proposed in the literature to improve multicast routing in WMN, grouping them into five categories: Flooding-Based, Tree/Mesh-Based, Network Coding and Layered Multicast, HWMP Enhancements, and Opportunistic/Resource-Aware strategies. Each approach is examined through representative works to highlight its mechanisms, strengths, and limitations, laying the groundwork for the advancements proposed in this study.

As a foundation for further analysis, a comparative study of representative multicast routing approaches is presented in Table 3.1. The following table synthesises the main contributions from prior studies, highlighting each approach's core mechanisms, strengths, and limitations within the context of WMNs.

3.1. Flooding-Based Approaches

Flooding represents the most fundamental mechanism for multicast delivery, whereby each node rebroadcasts every received message to its neighbours. Its stateless and topology-independent nature renders it particularly well-suited for highly dynamic networks, as it ensures that packets are disseminated extensively without necessitating route discovery or maintenance. However, this simplicity comes at the cost of excessive redundancy, increased channel contention, and energy inefficiency, particularly in dense networks where many nodes perform simultaneous retransmissions.

The work of Katia Obraczka [14] provides a critical evaluation of the reliability of flooding-based multicast in highly dynamic and mobile ad-hoc networks, thus motivating the search for alternative solutions. The authors posit that conventional state-based multicast mechanisms encounter challenges in ensuring delivery guarantees in rapidly evolving topologies. Consequently, flooding emerges as a promising alternative due to its minimal state requirements and its ability to withstand topology changes. The central hypothesis of this work challenges the conventional assumption that flooding inherently guarantees high

reliability. Instead, the hypothesis demonstrates that flooding performance degrades significantly as node mobility increases.

In their study, the flooding application was run simultaneously on “n” nodes, randomly arranged in the simulated field. Each node attempted to flood “m” messages to all other nodes. Upon reception of a packet, a node awaited a uniformly distributed time interval (between 0 and flooding-interval) prior to retransmitting it. The packet loss factor was calculated as the difference between the expected number and the actual number of messages received, collectively, divided by the total number of packets sent.

For a field measuring 1000 x 1000 m² with 50 nodes, the optimal range was determined to be between 200–250 m. Below this range, disconnectivity was observed to prevail, while above it, an increase in collisions was recorded. Consequently, a distance of 250 metres was designated as the primary parameter for transmission range. The authors observed a clear relationship between node mobility and packet loss. As node speed increased from 10 m/s to 100 m/s, the packet loss ratio rose from approximately 44% to 62%, highlighting the inability of flooding to sustain high delivery ratios in rapidly changing neighbourhoods. At elevated velocities, nodes expeditiously depart the transmission neighbourhood, thereby impeding the propagation of the ’flooding wave’. Furthermore, both duplicates and collisions decreased with increased mobility, as a result of the lower number of neighbours per node in fast scenarios.

It was concluded that, although flooding is simple and topology-independent, it does not guarantee reliability in high-mobility scenarios. To address these shortcomings, the authors propose three research directions: (i) the implementation of more robust versions of flooding with minimal state at the nodes, (ii) the rigorous quantification of overhead, and (iii) the use of scoped flooding to reduce collisions.

3.2. Tree-Based/Mesh-Based Protocols

Tree-based and mesh-based multicast routing protocols provide structured approaches to data dissemination in WMN. Tree-based protocols generally construct a spanning tree that connects the multicast source to all receivers, thereby ensuring path efficiency by missing the duplication of transmissions. However, tree-based structures are highly sensitive to topology changes, such as node mobility or link failures, which can disrupt the

tree, resulting in costly route reconstructions. Conversely, mesh-based protocols establish a network of redundant paths between the source and receivers. This added redundancy has been demonstrated to improve resilience and fault tolerance, but can lead to higher control overhead and occasional duplicate packet deliveries.

The article by Kaur and Kaur [15] offers a comprehensive survey of multicast routing techniques, categorizing them into tree-based, mesh-based, and hybrid solutions. Their work outlines the trade-offs in protocol design, emphasizing the vulnerability of tree-based systems in mobile scenarios and the improved robustness of mesh-based methods.

For the Tree-based protocols, the article talks about the Genetic Algorithm and Greedy Zone Routing. In the Genetic Algorithm, the chromosome is represented by the multicast tree structure. The process functions in accordance with the tree-based multicast forwarding protocol. The application of crossover and mutation operations directly to the tree is a key feature, with the purpose of simplifying the processes of encoding and decoding. This approach is both energy-efficient and energy-conscious, and it solves multicast forwarding problems with Quality of Service (QoS). The Greedy Zone Routing is a two-level tree-based routing approach that uses zone geographical coordinates. The establishment of routing within a designated zone is contingent upon the utilisation of local routing trees, employing the Routing Protocol for Low-power and Lossy Networks (RPL). The model exhibits enhanced scalability and performance, with efficient simulation outcomes when compared to tree-based structures, attributable to the interconnectedness of multiple trees across disparate zones.

For the Mesh-Based protocols, the article talks about Ant Colony Optimization-based multicast routing algorithm. In this algorithm, the formation of a mesh multicast structure is achieved through the secretion of pheromone values by 'ants' at each link node. The approach is inspired by the foraging behaviour of ants and is based on the principle of swarm intelligence. The utilisation of simulations has yielded results that demonstrate superiority in terms of throughput, packet delivery rate, and multicast group size when compared to ODMRP and MAODV.

3.3. Network Coding and Multi-Layer Multicast

Network coding and multi-layer multicast techniques have been developed for the purpose of improving bandwidth utilisation and enhancing the overall quality of service by transmitting data in encoded or layered forms. These approaches have been shown to be particularly effective in scenarios where receivers have heterogeneous capabilities or

experience varying network conditions. Rather than transmitting a single, uniform data stream that is constrained by the receiver and may be the weakest link, these methods enable different receivers to obtain different portions of the content according to their available bandwidth. This maximises efficiency and user satisfaction.

The article by Rezaei and Farzinvash [16] investigates the challenge of multicast communication in WMN, especially in scenarios with heterogeneous receivers in terms of bandwidth. The primary objective is to overcome the limitation that throughput is restricted by the slowest link when data is transmitted to all receivers at the same rate. In order to achieve the objective outlined above, the proposed algorithm employs a multi-layer scheme. This involves the encoding of multicast data in multiple layers. The number of layers received by each receiver is dependent on the available bandwidth at that particular receiver. Furthermore, two techniques are employed to enhance the number of layers obtained by the receivers: Network Coding and Wireless Broadcast Advantage. The incorporation of Wireless Broadcast Advantage introduces complexity to the design, as it necessitates the prior determination of the 'children' of each node.

The algorithm is divided into two main phases:

1. Determining the Children of the Nodes (Phase 1),
2. Layer Processing (Phase 2).

In Phase 1, the WMN is divided into groups in order to minimise the number of binary variables and ensure the manageability of the problem. Each group is processed independently, thus representing a reduced version of the original problem. The objective is to achieve equilibrium in the network load, thereby minimising the maximum load (interference). The organisation and processing of the groups is polynomially complex, as the number of links in each group is limited.

In Phase 2, the layers are considered successively, ensuring fairness between receivers. The subsequent application of grouping is directed towards the receivers, resulting in the limitation of the size of each group. The group model is designed to optimise the number of receivers that receive the layer, while ensuring that the load does not affect the decision-making process. Following each iteration, the network variables (physical flow and transmission time) are updated. The optimal model seeks to maximise the weighted sum of the layers received, assigning greater weight to the initial layers to reinforce fairness: receiving an initial layer is considered more valuable than receiving later layers.

The simulations demonstrated that the proposed algorithm significantly outperforms previous approaches, such as MLNC, in terms of both average throughput per receiver and the number of layers received. A substantial enhancement in throughput was observed, with increases reaching 250% in 30-node WMN and 150% in 50-node networks. These results indicate notable performance gains.

The findings further substantiated the enhanced reception of layers, thereby ensuring an optimal service quality and maintaining equitable treatment among receivers, even in scenarios where the total number of receivers is augmented.

3.4. HWMP-Based Enhancements

The HWMP is the default path selection mechanism in IEEE 802.11s. However, it was not designed with multicast in mind. Despite the fact that it provides both proactive and reactive routing capabilities, HWMP was not designed with multicast as a primary consideration. Consequently, multicast delivery in HWMP relies on controlled flooding, a method which introduces redundancy and scalability issues in large or dense topologies. Consequently, there have been calls for modifications to HWMP with the aim of enhancing its capacity for multicast support and mobility management.

Majumder [17] presents the Forward Pointer-Based Routing (FPBR) scheme as an extension of HWMP to manage the mobility of external STAs in WMN.

The key innovation of FPBR is the substitution of HWMP's proxy table and route update mechanism with a system of routing pointers. In the event of an external station changing access point collocated with Mesh Gate (APMG), rather than generating multiple route maintenance packets (PREQ, PREP, Proxy Update and their acknowledgements), the new APMG establishes dynamic pointers to redirect the station's traffic. This is achieved by exploiting the neighbour relationships (parent, child, or sibling) maintained in enriched routing and neighbour tables.

This enhancement results in significant improvements to hand-off efficiency:

- Downstream traffic is directed immediately by the Gateway via the pointer chain, thereby minimising the number of management messages.
- Upstream traffic follows existing routes to the Gateway, thus avoiding redundant discovery processes,

- The introduction of additional messages (ACK, NACK, PCA) enables the confirmation or invalidation of old routes without maintaining proxy tables.

With regard to implementation, FPBR introduces the following:

- Additional fields in messages (RANN, PREQ, PREP and beacons) to carry level information and hierarchical relationships;
- Additional control message types (PCA, ACK, NACK) to support selective route updating;
- Expanded routing and neighbourhood tables, capable of storing information on the level of the node in the tree, the root node and relationships with neighbours.

Through the use of simulation, FPBR has been demonstrated to achieve a reduction in signalling overhead, a decrease in end-to-end delay, and an enhancement in throughput. Consequently, this enables seamless mobility support and addresses some of the key limitations of the original HWMP design.

3.5. Opportunistic and Resource-Aware Approaches

Opportunistic and resource-aware multicast routing methods have been developed to optimise packet delivery by dynamically adapting to network conditions such as link quality, mobility patterns, bandwidth availability, and traffic load. These approaches are intended to progress beyond the limitations of conventional single-metric routing by taking into account multiple QoS constraints in a simultaneous manner. This progression is believed to enhance reliability, stability, and efficiency in WMN.

The study by Seetha et al. [18] proposes a multi-constraint multicast routing framework in WMN, motivated by the growing need for strict QoS in real-time applications such as Voice over Internet Protocol and videoconferencing.

The approach under discussion is based on the selection of multicast routes that give due consideration to multiple constraints, namely, bandwidth, delay, jitter, loss rate, network load, and distance, as opposed to reliance on a solitary metric. The article analyses three classes of existing solutions:

- **Fuzzy logic-based approaches**, which manage uncertainty and derive near-optimal routes with low computational overhead;

- **Optimisation-based techniques**, which focus on satisfying global QoS parameters while reducing temporal complexity;
- **Multi-metric solutions**, which combine several constraints to improve reliability and accelerate route discovery.

The results of the study indicate that fuzzy logic protocols are capable of achieving stable routes while maximising throughput, although in certain circumstances they may lead to the selection of random paths. Optimisation approaches, on the other hand, have been shown to guarantee service availability exceeding 90% while simultaneously reducing temporal complexity. Finally, multi-metric algorithms demonstrate the ability to produce more reliable routes, maintaining polynomial execution time and ensuring efficient performance.

It is concluded that the proposed framework, by considering multiple QoS constraints simultaneously, provides more reliable multicast routes, with fast discovery and efficient packet delivery, meeting the growing demands of multimedia applications in WMN.

Similarly, the work by Sanni et al. [19] addresses the challenge of multicast source mobility in WMN, which has the potential to cause disruptions in the multicast tree and compromise sensitive applications such as healthcare, video surveillance, or emergency operations. The proposal consists of formulating multicast routing as a constrained Steiner tree (CST) problem and solving it with a Differential Evolution (DE) algorithm, called DE-CRMT (Differential Evolution-based Core Rooted Multicast Tree). The architecture utilises a shared tree structure, where a core router functions as a fixed root. The operation of this architecture is divided into two stages: the optimisation of the unicast path from the mobile source to the core, and the construction of the multicast tree for the receivers. The implementation of DE has been demonstrated to facilitate the diversification of solutions and accelerate the process of convergence, thereby reducing the financial burden associated with tree maintenance.

It has been demonstrated that the algorithms in question do indeed converge after a small number of iterations and that they demonstrate an $O(n^3)$ level of complexity. The findings indicate that DE-CRMT exhibits superior performance in terms of stability under source mobility when compared with DE-SRMT (source-rooted tree). This enhancement leads to a substantial reduction in the cost and delay associated with tree reconstruction. In

scenarios involving multiple movements, larger multicast groups, or larger networks, DE-CRMT demonstrated a cost reduction of up to 74% and a delay reduction of 70% compared to DE-SRMT.

It is concluded that DE-CRMT is more stable and efficient for supporting mobile multicast sources in WMN, offering greater robustness in critical applications.

3.6. Comparative Summary Table

To provide a clearer overview of existing multicast approaches in wireless mesh networks, Table 3.1 presents a comparative summary of representative protocols. The comparison highlights the main strengths and weaknesses of each approach, focusing on aspects such as reliability, scalability, computational cost, and adaptability to mobility. This synthesis aims to support the identification of design trade-offs relevant to multicast routing in IEEE 802.11s mesh networks.

Article / Approach	Strengths	Weaknesses
Flooding for reliable multicast in MANETs [14]	• Simplicity; • Topology independent; • Robust in static or low-mobility scenarios.	• High packet loss under high mobility; • Significant overhead; • Collisions and redundancy of packets.
Tree-based and mesh-based protocols [15]	• Tree-based: energy efficiency and optimized paths; • Mesh-based: redundancy and resilience against failures.	• Tree-based: fragile under topology changes; • Mesh-based: higher forwarding overhead.
Network coding and multi-layer multicast [16]	• Better bandwidth utilization; • Fairness among receivers; • Substantial throughput gains (up to 250%).	• Higher algorithmic complexity; • Requires prior determination of node children; • Processing overhead.
FPBR – Forward Pointer-Based Routing for HWMP [17]	• More efficient handoff; • Reduced signalling overhead; • Increased throughput and lower delay; • Seamless mobility without proxy tables.	• Added complexity in tables and messages; • Maintaining hierarchical relationships may be costly in highly dynamic scenarios.
Multi-constraint framework: fuzzy logic, optimization, multi-metric [19]	• Considers multiple QoS constraints; • Fuzzy protocols achieve stable routes with low overhead; • Optimization ensures $>$90% availability; • Multi-metric approaches balance reliability with polynomial time.	• Fuzzy protocols may select random paths in tie cases; • Optimization approaches depend on proper parameter tuning; • Higher implementation complexity.
DE-CRMT – Differential Evolution for mobile multicast sources [19]	• Higher multicast tree stability under source mobility; • Fast convergence (O(n^3)); • Up to 74% cost reduction and 70% delay reduction vs. DE-SRMT; • Suitable for critical applications.	• Higher computational cost in large networks; • Dependency on evolutionary algorithm parameters for performance.

Table 3.1: Comparison of articles and multicast approaches in WMN

While each protocol exhibits distinctive advantages, it is evident that trade-offs exist between efficiency, complexity, and adaptability. Flooding-based methods are characterised by their simplicity, but they are also prone to redundancy. In contrast, optimization-based and evolutionary approaches achieve higher performance, but at the expense of increased computational demand.

4. Design and Implementation

This chapter details the design rationale and implementation process of the proposed flood-and-prune mechanism for multicast routing in IEEE 802.11s mesh networks. Section 4.1 delineates the conceptual design and underlying logic of the mechanism, while Section 4.2 provides a detailed exposition of the implementation steps within the NS-3 simulator. Collectively, these sections demonstrate the process of transforming theoretical principles into a functional simulation model.

4.1. Design

The proposed mechanism aims to enhance multicast efficiency in IEEE 802.11s mesh networks by introducing a broadcast-and-prune approach. Initially, multicast packets are disseminated to all nodes, similar to the behaviour of PIM Dense Mode (PIM-DM), and redundant paths are progressively pruned to limit unnecessary transmissions. This design seeks to improve scalability and reliability while maintaining robustness in dynamic topologies.

The initial phase of the study involved an attempt to comprehend the relative positioning of the nodes in relation to each other. In order to accomplish this objective, the Time To Live (TTL) that was already present in the 802.11s header was utilised, as illustrated in Fig. 2.4. By comparing the TTL of received frames against those transmitted by the node itself, each device can infer whether a neighbour is upstream, downstream or at the same level. This inferred relationship provides the foundation for deciding which nodes should transmit or suppress multicast packets.

However, in dynamic mesh networks, raw TTL values are subject to significant fluctuation due to routing changes, congestion, and packet loss. To obtain a stable and adaptive estimation of node position, an Exponentially Weighted Moving Average (EWMA) filter was selected. This method smooths short-term fluctuations while remaining responsive to gradual changes in topology, providing a balanced representation of the network state. When compared with simple averages or instantaneous measurements, EWMA offers superior adaptability under variable conditions [20].

To ensure that the estimation reflects a sufficiently stable pattern of received TTL values, pruning decisions are not allowed immediately at startup. Instead, each node waits until at least 10 multicast packets have been received before enabling the pruning mechanism.

This prevents premature or erroneous pruning in the early phases of packet dissemination, where the TTL samples may still be inconsistent or insufficient to determine reliable upstream–downstream relationships.

The EWMA is a statistical process monitoring tool that calculates the mean of the data, weighting it according to its time of occurrence. The selection of the EWMA control technique is based on the premise that the determination of the process's state of control is contingent on the EWMA statistic, which is defined as an exponentially weighted average of all previous data, including the most recent measurements. The selection of the weighting factor α enables the EWMA control procedure to exhibit sensitivity to both minor and incremental fluctuations in the process. The initialisation of the EWMA is done with the value of the first sample of the TTL with which the current node will send a transmission.

$$W_i = \begin{cases} 0, & i = 0 \\ \alpha \cdot W_{i-1} + (1 - \alpha) \cdot \theta_i, & i > 0 \end{cases} \tag{4.1}$$

Figure 4.1: α = weight parameter, θ_t = Value of the series in the current period i, W_i = EWMA for i (with $W_0 = firstTTL$).

After establishing how to estimate stable upstream-downstream relationships, it was necessary to select the appropriate multicast transmission model. Following a comparative analysis of ASM and SSM, as discussed in Chapter 2.4, the SSM model was chosen. This decision was motivated by its suitability for single-source scenarios, by its simpler configuration and simplicity that will translate into less protocol overhead, which aligns with the objectives of this study.

Before implementing the pruning mechanism, it was necessary to determine the most appropriate type of pruning to be employed: soft state pruning or hard state pruning. In the soft state prune, the prune created expires automatically after a certain time, and the same prune is only created again if it is renewed with a prune message. In the context of hard state pruning, the prune remains valid until it is explicitly removed, that is, until it receives a message indicating that the prune should be forgotten. In view of this characteristic, the implementation of soft state pruning was selected, since this approach does not necessitate explicit control messages to remove or modify the state. Furthermore, this method is more straightforward to implement within the timeframe available.

In contrast to protocols such as ADMR, where the protocol automatically prunes unused

branches based on the absence of downstream acknowledgments, in this case, the explicit pruning is necessary for the pruning to occur.

The flood-and-prune mechanism operates as follows:

1. Multicast packets are initially flooded through the network;
2. Each node observes the TTL values of received packets and, using the EWMA algorithm, estimates its relative position with respect to its neighbours;
3. Nodes maintain information about active peers, obtained through the Peer Management Protocol;
4. If a node detects that all downstream peers have issued prune messages and no local receiver is present, it initiates its own prune toward upstream or same-level nodes;
5. Prune messages are transmitted via unicast to ensure reliability, preventing the loss of control information that could occur in broadcast mode;
6. When a node receives a prune message stores it in a prune table;
7. When a node is trying to retransmitted the packet if the the prune condition is met, all the bellow peers sent a prune message, the retransmission of multicast packets from that node is stopped until the prune lifetime expires.

4.2. Implementation

The practical implementation of the proposed mechanism was performed using the NS-3.43 simulator. The work involved modifications to the HWMP protocol to incorporate multicast support and to implement the flood-and-prune functionality described in Section 4.1.

The initial phase of the study involved the configuration of a rudimentary mesh network, with no modifications made to the system, to ascertain the fundamental behaviour of HWMP multicast. It was at this point that the initial issue was encountered: namely, the transmission of mesh multicast frames was not occurring as expected. Following extensive research on this issue, a former student presented a solution in a Git issue*. Currently, multicast frames are not being treated as group-addressed frames (equivalent

*https://gitlab.com/nsnam/ns-3-dev/-/issues/485

to broadcast) but are instead being classified as unicast, as the majority of code primarily checks for broadcast, particularly during forwarding. This results in the prevention of multicast frames from being transmitted, as the resulting Path Request will be devoid of a corresponding Path Response. Following the implementation of the change that was presented as a solution to the problem, the programme achieved the expected results.

4.2.1 Integration with the Peer Management Protocol

Following the resolution of the previous issue, the pre-existing functions from the Peer Management Protocol class were utilised to obtain the active peer links for each node. It was necessary to establish an additional connection between the HWMP protocol and Peer Management Protocol classes, since these two classes were not contained within the same folder.

A map of the `std::map<Mac48Address, uint32_t> m_lastActivePeerAddrs` data type was created for the purpose of storing the list of active peer links for each node. This map is used to store the last peer addresses and their status, which are categorised as follows: 0 = downstream, 1 = upstream, 2 = at the same level.

4.2.2 TTL Handling and Node Position Estimation

Following the aggregation of the peer links for each node, it was imperative to capture the TTL of each packet received by the node in order to initiate the determination of the position of these peer links. As previously stated, the TTL is a field that is present in the packet header.

4.2.3 Duplicate Packet Issue and Improved Detection

While observing the behaviour of the received packets and TTLs, another error was found where only one packet sent by the source was recorded as a cycle of retransmitted packets. The source sent a packet to the other nodes, which retransmitted it back to the source. Instead of being discarded, these packets were received and retransmitted again. These retransmitted packets also had a different source address to the original packet, but the sequence number remained the same, creating a cycle. According to section 10.38.6, ’Detection of duplicate MSDUs/MMPDUs’, of 802.11-2020 [21], each station must cache the tuples ’<Mesh SA, Mesh Sequence Number>’. These fields would not change with each retransmission of group-addressed frames. The Transmitter Address and the Mesh

TTL change, with the TTL being decremented with each hop. To address this issue, a class developed by a previous student, `LruGroupSeqNo` *, was adopted to improve duplicate detection. This LruGroupSeqNo class was used in the DropDataFrame function that eliminates duplicate packets.

In the original `DropDataFrame` implementation, the protocol only keeps track of the last sequence number seen for each source. This means that any packet with a sequence number equal to or less than the last recorded one is automatically discarded. While this approach ensures monotonicity in packet reception, it presents several practical problems in real mesh networks. Since packets can follow different paths to their destination, it is common for them to arrive out of order, especially in topologies involving multiple hops, delay variation, MAC-level backoffs or route changes. The old version discards packets with a sequence number lower than the last one received, even if they are the first legitimate copy received by that node via an alternative path. This prevents the node from correctly forwarding the packet and breaks the flooding mechanism, which relies on each unique packet being resent exactly once. The modified version uses the LruGroupSeqNo class to solve these problems by replacing the unique value stored by the source with an LRU (least recently used) cache of sequence numbers. Rather than simply checking whether the packet is newer than the last one seen, the new method stores a window of recent sequence numbers and accepts any packet that has not yet been recorded in the cache. This behaviour is closer to the ideal for mesh networks, as it enables the reception of out-of-order packets and ensures that no legitimate packet is lost simply because it arrived later.

After implementing this class, the problem was almost solved. As a newer version of the ns-3 simulator was available, the versions were updated in the hope that this would resolve the problem. When the versions were changed, it was observed that the packet cycle was resolved even without the LruGroupSeqNo class. However, as the LruGroupSeqNo class is more suitable for real mesh network scenarios, solving the problem of artificial packet loss, ensuring the delivery of unique packets even when there are multiple paths and delay variation, increasing the reliability of the protocol and improving the user experience for loss- and delay-sensitive applications, it was also implemented in the presented solution.

*`https://github.com/eSoares/ns3-updated/commit/7693d9a62cc564566cc32c6386603bfaf23c4be4`

4.2.4 EWMA-Based Node Position Estimation

The EWMA algorithm was implemented as a function named `EWMANode`, with TTL as its input parameter. This function was later integrated into the `StartPrune` routine to dynamically update node positioning. The formula used for the exponential moving average is expressed in Equation 4.1.

4.2.5 Implementation Prune Messages

As 802.11s did not implement Prune messages, it was necessary to create them from scratch. The creation of these prune messages was based on the utilisation of Perr messages as a model.

The `IePrune` class is responsible for implementing a custom Information Element (IE) for 802.11s mesh networks. The purpose of this element is to instruct nodes to cease forwarding packets to specific destinations. The class maintains an internal list of pairs (destination, reasonCode) representing the nodes to be pruned and the reason for pruning. The software is equipped with a range of functions to facilitate the management of entries. These include the ability to add entries individually using the `AddPruneUnit()` method, to set all entries simultaneously using the `SetEntries()` method, and to retrieve the complete vector using the `GetPruneUnits()` function. Furthermore, it stores metadata that is essential for PRUNE to function. The metadata associated with this class is displayed in table 4.1.

Unit	Metadata	Usage
Mac48Address	IP receiver	Intended receiver of the PRUNE message
uint32_t	Index Interface	Interface through which the message is sent
uint8_t	TTL	Associated time-to-live value
Mac48Address	Multicast Group Address	Group address to which the PRUNE applies
Mac48Address	MAC address Originator	MAC address of the originator of the flow or packet

Table 4.1: Metadata used in the PRUNE message.

The `IePrune` class encapsulates all the information necessary to construct, send and interpret PRUNE messages in the context of mesh forwarding. This ensures that nodes receive precise instructions on which destinations should be blocked and in what context.

4.2.6 Soft-State Prune Table

Following the establishment of the message base, the subsequent stage of the process involved the implementation of all the functions that assist prune. In light of the necessity

to establish a global repository for the storage of prunes, the structure `m_pruneTable` was devised. Each entry in this table associates a tuple of MAC addresses and a timestamp, the tuple consists of the source (peer that received the prune), destination (peer who sent the prune) and multicast group affected by the prune.

In order to implement soft state prune,each prune entry is associated with a timeout value, which denotes the timestamp of the prune's creation. The implementation of soft state prune is complemented by the `AddPruneEntry(Mac48Address src, Mac48Address dst, Mac48Address multicastGroup)` function, which is triggered when a node receives a prune message and needs to register it. Within this function, there is a Scheduler that is triggered after `m_pruneTimeout` and calls the `ExpirePruneEntry(Mac48Address src, Mac48Address dst, Mac48Address multicastGroup)` function, which deletes the entry from the m_pruneTable. This is the method by which the soft state prune mechanism is implemented.

4.2.7 StartPrune Routine and TTL Processing

In order to activate the prune mechanism, it was necessary to implement the `StartPrune(Ptr<const Packet> packet, Mac48Address transmitter, Mac48Address source, uint32_t interfaceIndex, Mac48Address group, bool use)` function. This function is called whenever a packet is received, regardless of whether it is discarded. When observing packets in Wireshark, packets that were identified as duplicates never reached the HWMP as expected. However, without information about the TTL sent by this node to the current node, it was impossible to determine this node's position relative to the current node. Whenever this node sent a packet, it was treated as a duplicate and the current node never received this information. One solution is to call this function before packets are dropped. However, since EWMA is calculated in this function, the extra TTLs would interfere with the calculations. Therefore, the use parameter distinguishes between using it for the EWMA calculation and not using it. This function is called with the value 'true' if the TTL obtained for EWMA calculations of the actual node is to be used, and with the value 'false' if the node position is to be calculated only. Within this function, a check is made on whether the packet is multicast; if not, it terminates immediately. After confirming that the received TTL is to be used in EWMA, the function is called with the TTL that the node will retransmit, i.e., if it receives a TTL of 30, what will be sent to the EWMA calculation function is 29. The TTL calculated with EWMA facilitates the ranking of the position of the neighbouring

nodes of the current node. In order to have sufficient information about the neighbours, the delivery of prune messages only initiates after the node has received 10 packets.

4.2.8 Prune Decision Logic

The process of pruning can be delineated as such: in the absence of an active prune, the node must first ascertain whether all peers below it have transmitted a prune message. The reason for this is that if any of them have not sent one, it signifies the presence of a node below the current node that is awaiting the receipt of the packet transmission. Subsequent to confirming receipt of a prune message from the designated set of nodes, the current node will then confirm its own desire to receive the packet. In the event of such a scenario, no observable outcome is evident. In the event that a device does not wish to receive the multicast transmission, that is, in the event that the device is not a member of the multicast group, the `SendPrune` function will be invoked. This will result in the transmission of the prune message to the node that transmitted this message. Prune messages are transmitted to nodes that are either above or at the same level as the node in question. This approach was adopted in preference to the alternative of only sending to nodes above, with the result that this pruning of unnecessary branches is more effective.

The figure 4.2 illustrates the comparison between two pruning strategies: pruning limited to superior nodes and pruning extended to both superior nodes and those at the same hierarchical level.

In the first case (a), where the prune message is transmitted exclusively to nodes above, issues can arise when nodes at the same level still intend to receive multicast transmissions. It has been established that, since these same-level nodes do not receive the prune message, they may continue to retransmit packets upon detecting that one of their peers has not done so. Such behaviour can result in unnecessary retransmissions and redundant packet deliveries to nodes that are no longer interested in receiving the data, thereby increasing channel occupancy and overall interference.

Conversely, the second case (b) extends the pruning mechanism to include nodes at the same level. It is imperative to ensure that all peers within the same tier are informed of the prune state, thus effectively suppressing redundant retransmissions. This results in a more efficient multicast dissemination process, reducing packet duplication and improving overall network performance in terms of interference mitigation and resource utilisation.

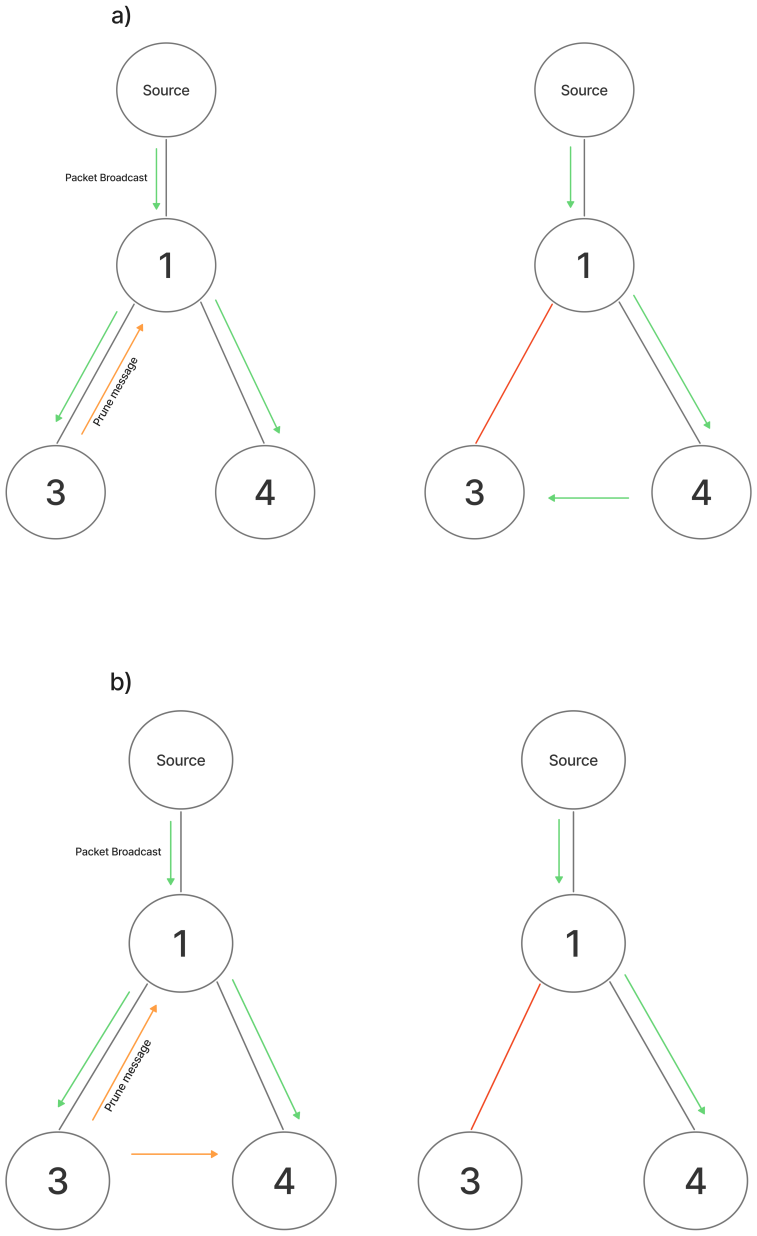


Figure 4.2: a) Prune only to nodes above. b) Prune to nodes above and the same level.

The prune message contains supplementary information. In addition to the recipient (addr1) and the sender (addr2), the header's addr3 and addr4 fields were used to transmit information of the target multicast group and the originator of the packet, respectively. As the prune message only needs to be propagated to the intended node, the TTL is set to 1 by default so that, once the message reaches its destination, it is not propagated further, even though being unicast it wouldn't happen anyway, it is still specified. Upon transmission of the message, the receiver will receive the message with the `ReceivePrune` function

trigger, where a new prune is added to the m_pruneTable using the `AddPruneEntry` function.

When implementing the prune message, it could be sent in two ways: either by unicast or by broadcast. While in broadcast it is only necessary to send one message that would reach all neighbours above the node, in the event of a reception failure the prune would be lost, as it would not be retransmitted. In the case of unicast, more messages need to be sent, but in the event of packet loss, a retransmission is made, which makes it more reliable. Moreover, since unicast transmissions support acknowledgment mechanisms, they can also employ higher Modulation and Coding Schemes (MCS), allowing for higher data rates. With this in mind, prune messages were implemented in unicast.

4.2.9 Integration with Route Request Logic

The final step in this process is to prune the paths present in the prune table. In order to accomplish this objective, a series of conditions were incorporated into the `ResquestRoute` function prior to the initiation of the broadcast retransmission with the purpose of ascertaining whether this retransmission would be undertaken. The initial step entailed ascertaining the nodes that were below the current node. This was followed by verifying whether all the nodes below had transmitted a prune message to the current node. As packet retransmission is conducted in broadcast mode, it is either disseminated to all nodes or not disseminated to any nodes. Therefore, it is crucial to ascertain whether all nodes have transmitted a prune message. In the event that all nodes below have transmitted a prune message, retransmission is pruned from the current node onwards. As the position of each node in relation to the current node is still being calculated in the first packets received, the path is only pruned from the second packet onwards to allow sufficient time for the EWMA to calculate the position of the nodes.

The code for this implementation can be found in my personal GitHub repository (`https://github.com/mariajoao722/ns-3.43.git`).

5. Experiments

In order to evaluate the efficacy of the proposed modifications, a series of comparative experiments were conducted using the implementation described in Chapter 4 within the ns-3.43 simulator. The objective of these experiments was to ascertain whether the proposed approach yielded quantifiable enhancements in comparison with the standard HWMP protocol. In order to perform a comprehensive evaluation, appropriate performance metrics were selected, with a focus on the Packet Delivery Ratio (PDR), end-to-end delay, and channel occupancy.

Packet Delivery Ratio (PDR): The objective of this study is to ascertain whether the percentage of non-duplicate packets received by the intended recipients matches the total number of packets that were intended to be sent to them. It is important to note that the definition of transmitted packets does not include retransmissions. The metric was computed in two distinct ways.

Firstly, the global PDR was calculated using the equation 5.1, in which the numerator is defined as the summation of the number of packets received, P_r, for each individual receiver, r, across all receivers, R. The denominator is defined as the total quantity of packets that should have been received, which is calculated by multiplying the number of receivers by the number of sent packets, S, excluding retransmissions.

$$PDR = \frac{\sum_{r \in R} \|P_r\|}{\|R\| \times \|S\|} \tag{5.1}$$

Secondly, the per-node PDR was calculated using the equation 5.2, where the metric was evaluated individually for each receiver node, using its own number of received packets, P_r, as the numerator and the total number of transmitted packets, S, as the denominator. This per-node analysis facilitates the evaluation of disparities in packet reception across the network.

$$PDR_{node} = \frac{\|P_r\|}{\|S\|} \tag{5.2}$$

Average End-to-End Delay: This metric is defined as the average time required for a packet to travel from the source to the destination. The mean value is determined by the

calculation of the sum of the latencies of all successfully delivered, non-duplicate packets, and the subsequent division of this by the total number of packets received. The delay of each packet p is defined as the difference between its transmission time St_p and its reception time Rt_p^r at receiver r, as seen in the equation 5.3.

The global average end-to-end delay was initially computed by considering all packets received by all receivers. Furthermore, the metric was calculated on a per-node basis, where solely the packets received by a specific receiver, designated as r, were considered in both the numerator and denominator. This per-node computation facilitates the identification of potential latency disparities between nodes in the same network topology.

$$AvgEED = \frac{\sum_{r \in R} \sum_{p \in P_{recv_r}} \left(Rt_p^r - St_p \right)}{\sum_{r \in R} \| P_r \|} \tag{5.3}$$

Channel Occupancy: This metric quantifies the proportion of time that a wireless channel is occupied by transmissions, receptions, or other activities that render the channel unavailable for new transmissions. For each Wi-Fi radio (WifiPhy), the ns-3 simulator emits a state trace event whenever the PHY layer leaves a state, providing the state name, its start time, and the duration spent in that state. The relevant PHY states are: **IDLE**: channel sensed free, **TX**: transmitting, **RX**: receiving/decoding, **CCA_BUSY**: channel sensed busy (energy detected) but not decoding. The total busy time, T_{busy}, is computed as the summation of the durations spent in the *TX*, *RX*, and *CCA_BUSY* states, while the total simulation time, T_{total}, is obtained by summing the durations of all states, including *IDLE*. The Channel Occupancy is then expressed as the percentage of time the channel is busy relative to the total simulation time, as shown in equation 5.4.

$$\begin{aligned} T_{busy} &= TX + RX + CCA_BUSY \\ T_{total} &= T_{busy} + IDLE \\ ChannelOccupancy &= 100 \times \frac{T_{busy}}{T_{total}} \end{aligned} \tag{5.4}$$

In the simulations, the YansWifiChannelHelper::Default() configuration was used, which establishes a wireless channel comprising two key components: a propagation delay model and a propagation loss model.

The propagation delay model is ns3::ConstantSpeedPropagationDelayModel, which assumes electromagnetic signals travel at the speed of light in a vacuum. This means that

the delay between transmission and reception is directly proportional to the physical distance between nodes, with no random variations included. Using a deterministic delay model ensures that any observed differences in packet delivery and timing are caused by network behaviour rather than variability in the channel delay itself.

The propagation loss model employed is the ns3::LogDistancePropagationLossModel, which utilises the log-distance path loss equation, $L = L_0 + 10nlog_{10}(\frac{d}{d_0})$, to calculate the received power. This model deterministically represents the decrease in signal strength with increasing distance, thereby directly impacting the probability of successful packet reception. In NS-3, the default parameters for this model include a path loss exponent (n) of 3.0, a reference distance (d_0) of 1 metre, and a reference loss (L_0) of approximately 46.68 dB, corresponding to the free-space path loss at 5 GHz. When combined with the default physical layer configuration the model yields a theoretical communication range of approximately 150 metres under ideal free-space conditions. In practice, however, due to factors such as modulation-dependent sensitivity, interference, and background noise this range changes.

It is imperative to comprehend these models in order to accurately interpret the experimental results. The propagation loss model is a key factor in defining the range and reliability of wireless communication, with a consequential influence on metrics such as PDR and throughput. The propagation delay model exerts influence on timing-related behaviours, including contention, backoffs, and route discovery latency.

The experiments conducted were categorised into two distinct classifications: validation experiments and evaluation experiments. In the validation experiments, the correctness of the implementation of the proposed flood-and-prune protocol is assessed in relation to the expected functionality. In contrast, the evaluation experiments are designed to identify any potential advantages that the implemented mechanism may offer in comparison to the controlled flooding protocol.

5.1. Validation experiments

As previously stated, validation experiments are conducted to verify whether the protocol functions as anticipated. To this end, four distinct scenarios were considered. In all scenarios, a total simulation time of 80 seconds was used, with packet transmission commencing at 10 seconds and continuing at a rate of 90 packets per second until the end of the simulation, the size of the packets is 1024 bytes, the prune lifetime is 30 seconds, the

sending node is identified by a blue circle, the receivers are identified by a green circle, and the rest of the nodes by a red circle. The nodes were carefully positioned to ensure that the transmission range of each node was sufficient to reach its immediate neighbour but not the following one, thereby enabling a controlled and predictable propagation of messages.

In the first scenario, a line of four nodes is presented, with the sending node designated as node 0 and the receiving node as node 1, as illustrated in figure 5.1. In this scenario, given that node 3 is not a member of the multicast group and has no nodes below it, it is expected to transmit a prune message to node 2. Consequently, upon receiving this message and not being part of the multicast group, node 2 sends a prune message to node 1, as all nodes below it have sent a prune message, and it itself does not want to receive the packet. Finally, this part of the network no longer receives traffic within the prune lifetime.

As indicated by the experimental logs, the observed behaviour is consistent with the anticipated sequence of events. Specifically, node 3 transmits a prune message to node 2, subsequent to the reception of the prune message, node 2 transmits a prune message to node 1. It can also be observed that both nodes cease to receive packets during the time period stipulated for soft state prune with the help of channel occupancy. In the version with prune, node 3 exhibits a channel occupancy rate of 0.12%, whereas in the standard controlled flooding it is 13%. This observation is consistent with the findings for node 2, where the version employing pruning exhibits a busy rate of 2.64%, while the version without pruning demonstrates a busy rate of 24.27%. This finding indicates that these nodes did not receive a high volume of transmissions, nor did they retransmit a significant amount.

The second scenario, illustrated in figure 5.2, aims to demonstrate the behaviour when there are two nodes at the same level. It was hypothesized that upon confirming the absence of nodes below node 4, a prune message would be transmitted to both node 3 and node 2, given their superior position above it. Additionally, a prune message should be transmitted between nodes 2 and 3; however, the specific node responsible for sending this message is contingent upon the EWMA calculation, which can be subject to variation due to the retransmission delay. Finally, nodes 2 and 3 confirm that all nodes below them have sent a prune message, and then send a prune message to node 1. The medium occupancy feature facilitates the verification of this pruning process, ensuring that the

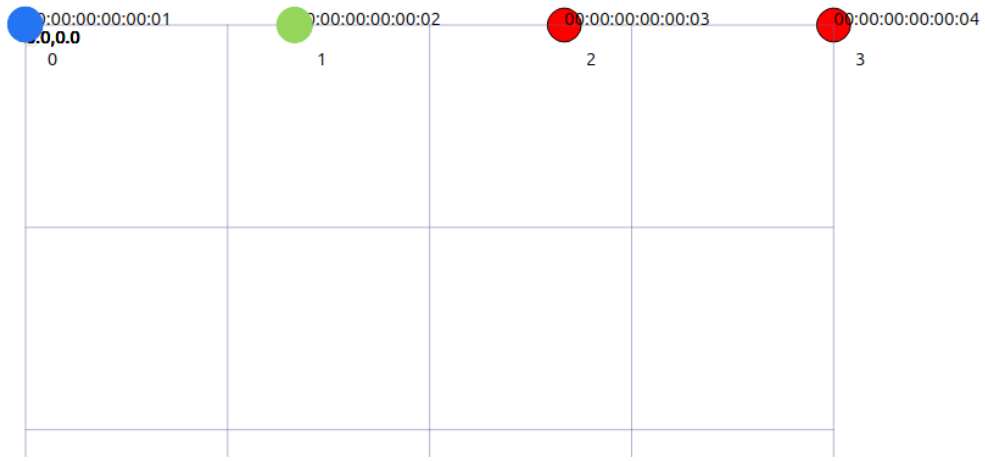


Figure 5.1: Validation Experiment 1

nodes cease to receive retransmissions as intended.

The behaviour observed in the simulation of this experiment is in accordance with the expected sequence of events, wherein node 4 initiates by transmitting a prune to node 2, subsequently followed by node 2 transmitting a prune to node 3. This sequence is initiated due to node 2 being the first to receive a prune from the preceding nodes. Following this pruning procedure, a subsequent pruning action is initiated from node 4 to node 3. Upon realising that all nodes below it have sent a prune, Node 2 sends a prune to Node 1, and the same process is repeated for Node 3. As is apparent from the channel occupancy, the nodes that received a prune exhibited a reduced number of retransmissions, thereby ensuring the efficacy of the prune. This is due to the fact that, in the presence of the prune, nodes 4, 3, and 2 are busy 0.18%, 2.68%, and 2.68% of the time, respectively. In contrast, in the absence of the prune, node 4 is busy 18.14%, node 3 is busy 27.13%, and node 2 is busy 27.13%.



Figure 5.2: Validation Experiment 2

The third scenario, illustrated in the figure 5.3, is a 3x2 node grid. It is expected that prune messages will originate from node 5, as it is the most distant from the source. Node 5 receives packets from nodes 2 and 4 that carry the same TTL value, indicating that they originate from transmissions at the same hop distance from the source. Consequently, it initiates a prune to these two nodes. Subsequent to this message, it is anticipated that node 4 will transmit a prune to both node 3 and node 1. In accordance with the same logic, it can be deduced that node 2 should transmit a prune message to node 1. Finally, upon receiving a prune message from all nodes below it, node 1 transmits a prune message to node 0. It is anticipated that the pruning will be executed on all the pathways that have been included in the prune table, with the exception of the pruning from node 0 to node 1, as node 0 does not execute pruning.

The behaviour observed in the simulation is consistent with the expected behaviour. The sequence initiates with node 5 transmitting a prune to node 2, followed by node 2 transmitting a prune to node 1. Upon returning to node 5, a prune is transmitted to node 4, and subsequently, node 4 disseminates a prune to node 1. Upon receipt of a prune message

from the nodes beneath it, node 1 then transmits a prune message to node 0. Finally, node 4 transmits a prune to node 3. It is evident that the pruning process has been executed, except for the path node 0 -> node 1, as node 0 does not execute the pruning function. With the help of channel occupancy, in conjunction with the presence of prune, it can be observed that nodes 5, 4, and 2 are occupied by 0.26%, 2.84%, and 0.29%, respectively. In the absence of the prune, it is observed that nodes 5, 4, and 2 are occupied at rates of 29.99%, 32.73%, and 28.62%, respectively. Consequently, it can be deduced that the prune version results in reduced packet retransmissions for these nodes.

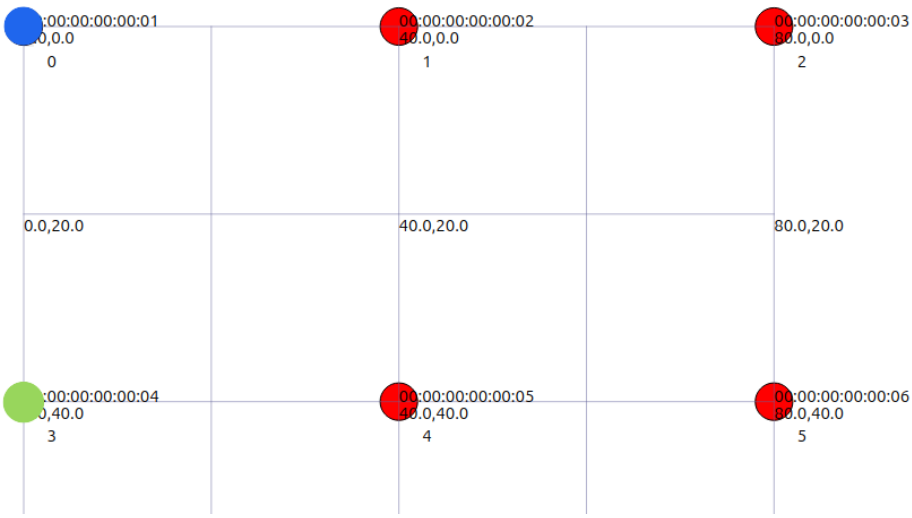


Figure 5.3: Validation Experiment 3

The final scenario consists of a 3x3 grid of nodes with two receivers, as illustrated in the figure 5.4. It is expected that the pruning will begin at node 8, which will then transmit a prune message to nodes 7 and 5. Subsequently, node 7 will transmit a prune to nodes 6 and 4, and node 5 will transmit a prune to nodes 2 and 4. After receiving prune messages from all nodes below it, node 4 will send a prune to nodes 3 and 1. Finally, node 2 transmits a prune to node 1.

In this case, the observed behaviour does not match the expected behaviour. Rather than pruning starting at node 8, it starts at node 4. This unexpected initial behaviour can be further justified by the fact that the transmission of prune messages only begins after a node has received 10 packets, allowing sufficient time for the EWMA and relative node positioning to stabilise. In this scenario, node 4 likely reached this 10-packet threshold earlier than node 8, thus triggering pruning first. The fact that node 4 sends a prune message for all of its peers can be explained by the EWMA calculation at node 4, which was affected by the transmissions it received. Specifically, while nodes 1 and 3 consistently

transmitted packets with a TTL of 31, node 4 also received packets from nodes 5 and 7, which carried lower TTL values. As a result, the inclusion of these packets reduced the EWMA at node 4 to around 29. This caused node 4 to interpret that retransmissions to node 4 are always above or at the same level as it. Consequently, pruning starts from node 4 to node 1. Then, as node 4 thinks that node 7 is at the same level, it also sends a prune to node 7. The next prunes are from node 4 to nodes 5 and 3. Next, node 8 sends a prune to node 7, as expected. Node 7 then sends a prune to node 6; node 8 sends a prune to node 5; node 5 sends a prune to node 2; and finally, node 2 sends a prune to node 1. The behaviour in the first prune is not as expected, but after the prunes have been forgotten, the protocol behaves as expected: node 7 is now considered a node below node 4, as is node 5. To confirm that pruning is effective, channel occupancy was observed. Overall, this changed from 36% without pruning to 15% with pruning, nodes with active pruning had much lower channel occupancy.



Figure 5.4: Validation Experiment 4

5.2. Evaluation experiments

As previously stated, evaluation experiments are conducted to identify any potential advantages that the implemented mechanism may offer in comparison to the controlled flooding protocol. In order to address this issue, random scenarios with different characteristics and a 4x3 grid were used to test the interference of multicast and unicast.

Furthermore, these scenarios were also designed to evaluate the advantages and potential drawbacks of the flood-and-prune mechanism when compared to controlled flooding. In order to more accurately reflect the reality of mesh networks, we decided to use a larger number of nodes and introduce randomness in this study.

For the experiments, two types of topologies were considered: topologies with 10 nodes (5.2.1) and topologies with 30 nodes (5.2.2). This choice is intended to illustrate the behaviour of the protocols in networks of different sizes. For the 10 nodes topologies, we have 2 nodes and 5 nodes as receivers to demonstrate how it works when only a small part of the network wants to receive, and when half of the network wants to receive packets. The same happens for the 30 nodes topologies, where we have 5 nodes and 15 nodes as receivers to demonstrate how it works when only a small part of the network wants to receive, and when half of the network wants to receive packets. To ensure statistical significance and reduce the influence of random variations in node placement and channel conditions, each scenario was repeated twenty times with different seeds, and the results were averaged across all runs.

The random topologies and characteristics to be studied are as follows:

1. 10 nodes with 2 receivers and with heavy network traffic (90 packets per second);
2. 10 nodes with 2 receivers and with lower network traffic (30 packets per second);
3. 10 nodes with 5 receivers and with heavy network traffic (90 packets per second);
4. 10 nodes with 5 receivers and with lower network traffic (30 packets per second);
5. 30 nodes with 5 receivers and with heavy network traffic (90 packets per second);
6. 30 nodes with 5 receivers and with lower network traffic (30 packets per second);
7. 30 nodes with 15 receivers and with heavy network traffic (90 packets per second);
8. 30 nodes with 15 receivers and with lower network traffic (30 packets per second);
9. 10 nodes with 2 receivers and with lower network traffic (30 packets per second) and unicast (UDP) at the same time;
10. 10 nodes with 2 receivers and with heavy network traffic (90 packets per second) and unicast (UDP) at the same time.

On the other hand, to study the interference of multicast communications with unicast communications, we used a 4x3 grid of nodes with less network traffic.

For the experiments with 10 nodes and 30 nodes, the area is the same (100 m $\times$100 m). With more nodes in the same area, the density is higher, and the network would be more congested, but there are more alternative paths for the packets to get to the destination. In the figures below, fig. 5.5 and fig. 5.6, we can see an example of the distribution of the 10 nodes and the 30 nodes, respectively. As another case study, in the experiences 5 and 7 with 30 nodes, we used a bigger area of 300 m $\times$300 m to observe how the propagation model and transmission range affect connectivity and packet delivery in sparser topologies. Fig. 5.7 illustrates the distribution of the 30 nodes in this configuration.

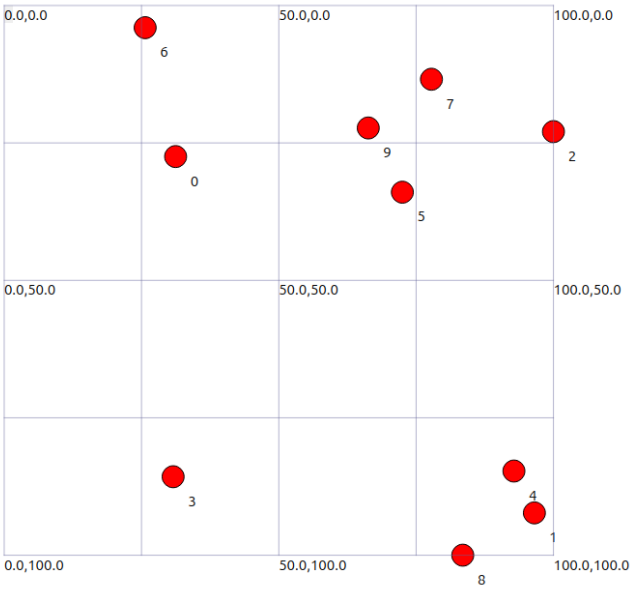


Figure 5.5: Distribution of 10 nodes in 100 m$\times$100 m.

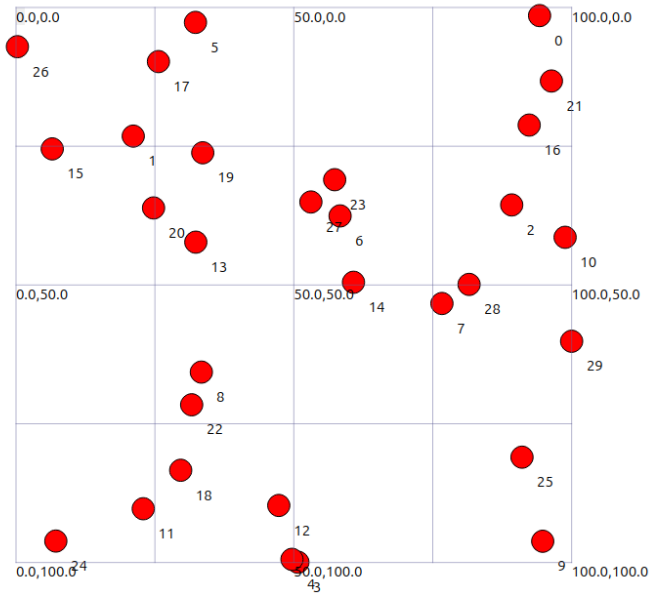


Figure 5.6: Distribution of 30 nodes in 100 m×100 m.

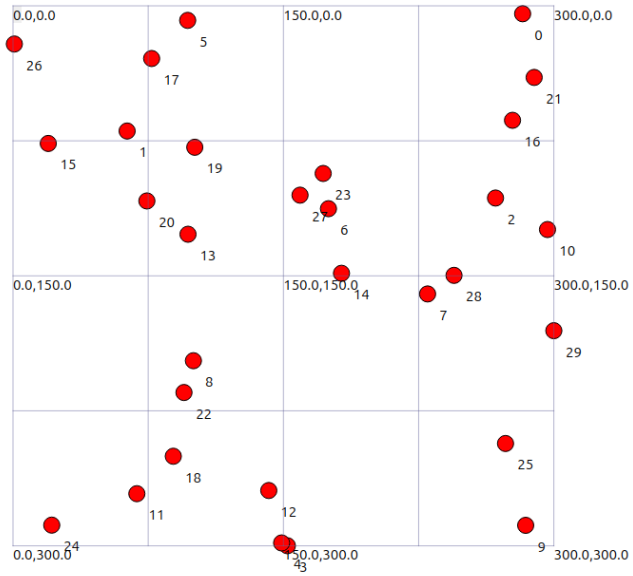


Figure 5.7: Distribution of 30 nodes in 300 m×300 m.

5.2.1 Topologies with 10 nodes

In the initial scenario, consisting of 10 nodes, 2 receivers, and a high packet traffic (90 packets per second), it is observed, fig. 5.8 and fig. 5.9, that the PDR for the flood-and-prune protocol is 94.83% and for the controlled flooding protocol is 86.82%, the delay is significantly reduced, from an average of 8.43 milliseconds to an average of 3.02 milliseconds in the flood-and-prune protocol. Finally, the average channel occupancy with prune is also lower, going from an average of 51.19% to 21.61%. In this kind of scenario,

the flood-and-prune protocol has proved to be more efficient than the controlled flooding protocol.

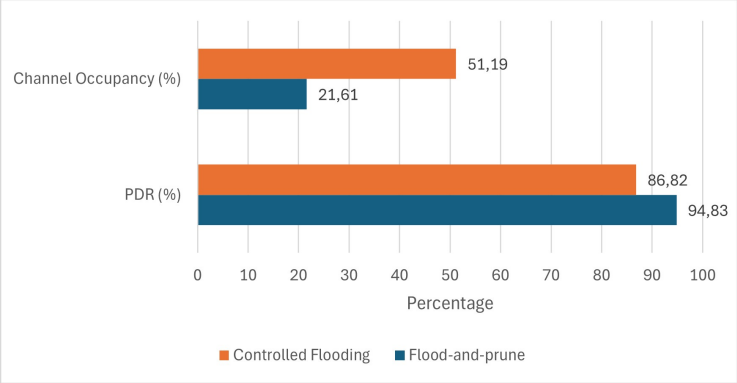


Figure 5.8: Global Channel Occupancy and Global PDR for 10N, 2R and high traffic

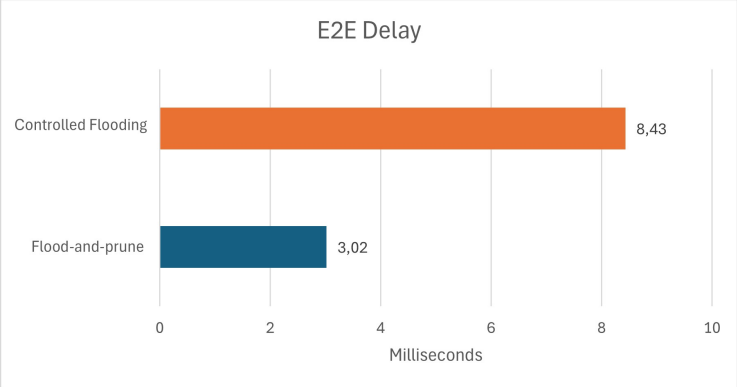


Figure 5.9: Global E2E Delay for 10N, 2R and high traffic

For the second scenario, consisting of 10 nodes, 2 receivers, and a lower packet traffic (30 packets per second), it is observed, fig. 5.10 and fig. 5.11, that the PDR for the flood-and-prune protocol is 94,88% and for the controlled flooding protocol is 87.06%, the delay goes from an average of 4.71 milliseconds to 3.06 milliseconds with the flood-and-prune protocol. The channel occupancy goes from 49.33% to 20.94% with the new implementation. As the packet traffic is lower, congestion will be lower too, which explains the results obtained.

While the controlled flooding approach may theoretically achieve close to 100% delivery, this is not realised in practice due to several factors. Firstly, it is important to note that multicast and broadcast transmissions at the MAC layer do not employ acknowledgements or retransmissions. This means that any packet loss caused by interference or collisions is permanent. Secondly, even with a traffic rate of 30 packets per second, short-term bursts of transmissions can momentarily saturate the channel, producing local congestion despite the relatively low average load.

In this kind of scenario, the flood-and-prune protocol has proved to be more efficient than the controlled flooding protocol.

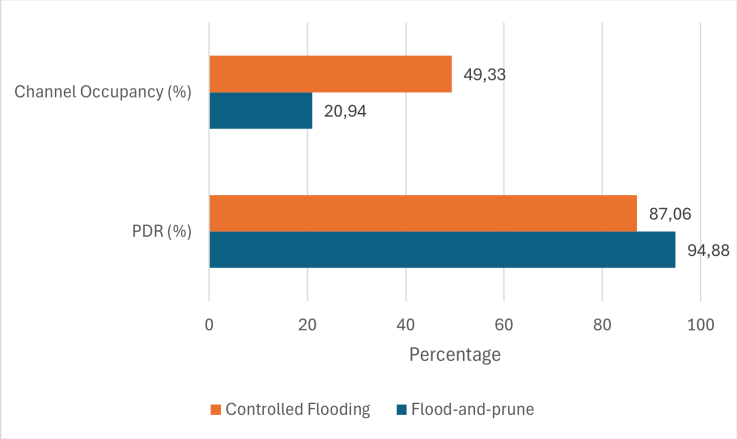


Figure 5.10: Global Channel Occupancy and Global PDR for 10N, 2R and low traffic

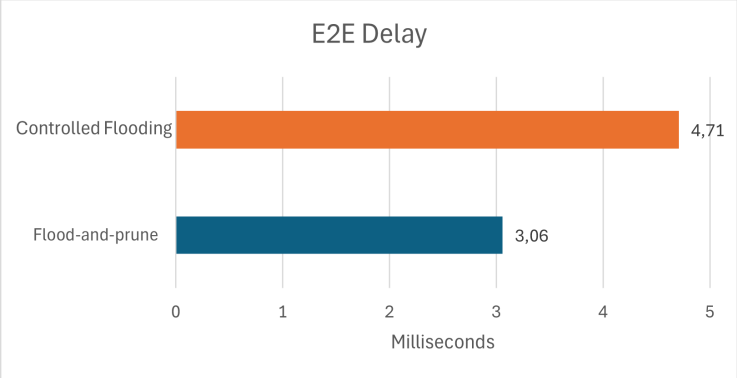


Figure 5.11: Global E2E Delay for 10N, 2R and low traffic

On the third scenario, consisting of 10 nodes, 5 receivers, and a high packet traffic (90 packets per second), it is observed, fig. 5.12, that the PDR for the controlled flooding protocol is 85.06% and for the flood-and-prune protocol is 92.97%. The delay,fig. 5.13, goes from an average of 8.05 milliseconds to 3.04 milliseconds with the protocol with the prune. And the channel occupancy with prune is lower than without prune, going from an average of 51.19% to 29.25%.

Although both protocols exhibit high delivery ratios under this higher traffic rate, the controlled flooding scheme experiences a more significant performance degradation as the channel becomes increasingly saturated. With 90 packets per second, the simultaneous transmissions generated by multiple neighbouring nodes lead to high contention and overlapping transmissions. The flood-and-prune protocol mitigates this effect by suppressing redundant retransmissions, thus reducing the number of concurrent transmissions and

the overall channel load This reduction in contention is reflected in the channel occupancy values. The lower occupancy translates directly into fewer collisions and shorter medium access delays, explaining the substantial improvement in end-to-end delay. The gain in PDR further confirms that, under high traffic, pruning mechanisms not only alleviate congestion but also enhance delivery reliability.

In this kind of scenario, the flood-and-prune protocol has proved to be more efficient than the controlled flooding protocol.

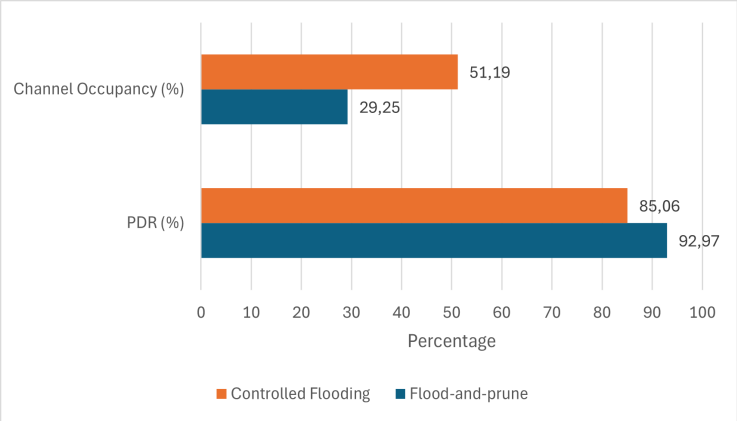


Figure 5.12: Global Channel Occupancy and Global PDR for 10N, 2R and high traffic

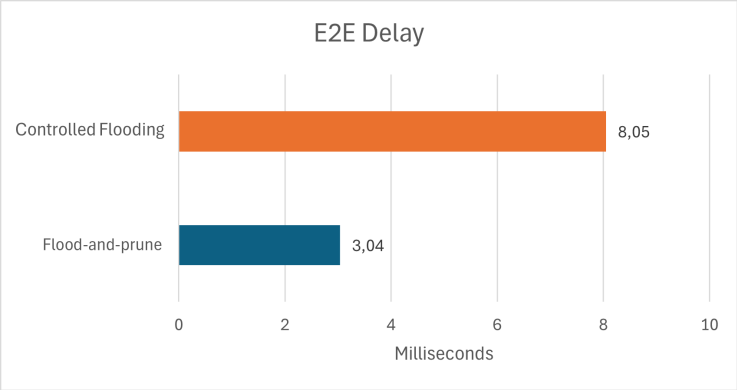


Figure 5.13: Global E2E Delay for 10N, 2R and high traffic

On the fourth scenario, consisting of 10 nodes, 5 receivers, and a lower packet traffic (30 packets per second), it is observed, fig. 5.14 and fig. 5.15, that the PDR remains practically identical, with 95.6% for the controlled flooding protocol and 95.22% for the flood-and-prune protocol, and the delay also remains almost the same with the controlled flooding protocol with an average of 3.17 milliseconds and the flood-and-prune protocol with an average of 2.99 milliseconds. The channel occupancy with prune is lower than without prune, going from an average of 18.99% to 10.55%. As the packet traffic is lower, congestion will be lower too, which explains the results obtained.

In this scenario, where the results are practically the same, the channel occupancy is of significance, with prune offering only this advantage. Even without pruning, the medium usage stays far from saturation, which allows multicast transmissions to perform efficiently under controlled flooding. The prune mechanism, therefore, offers only a marginal improvement by further reducing the number of redundant broadcasts. However, this reduction still slightly mitigates interference with concurrent unicast transmissions. It is evident that the presence of minimal packet traffic results in low congestion levels. Consequently, this leads to elevated PDR in both versions, with minimal discrepancy observed in delay.

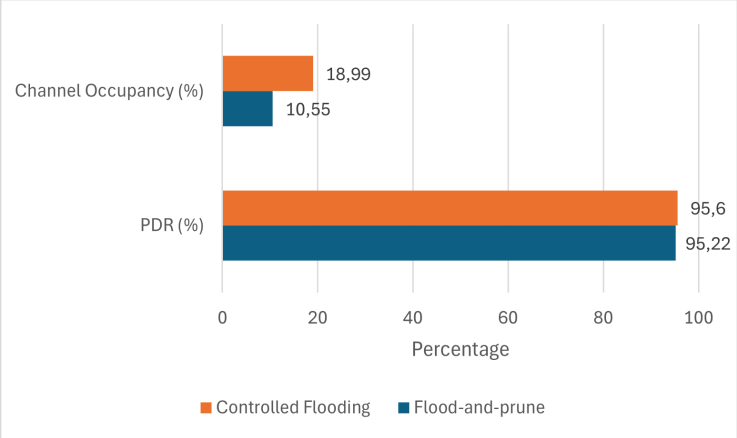


Figure 5.14: Global Channel Occupancy and Global PDR for 10N, 5R and low traffic

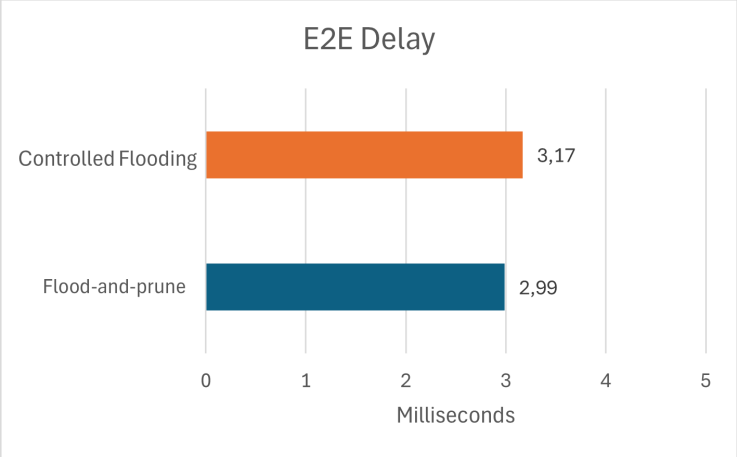


Figure 5.15: Global E2E Delay for 10N, 5R and low traffic

5.2.2 Topologies with 30 nodes

In the fifth scenario, consisting of 30 nodes, 5 receivers, and a high packet traffic (90 packets per second) and area of 100 m$\times$100 m, the flood-and-prune protocol was found to result in a higher PDR, with an average of approximately 53.95%, in comparison to the controlled flooding protocol, which exhibited an average rate of 37.19%, fig. 5.16. On

the other hand, the delay, fig. 5.17, in the flood-and-prune protocol is greater than in the controlled flooding protocol, but not by much, with the controlled flooding protocol having an average of 472.46 milliseconds and the flood-and-prune protocol having an average of 475.05 milliseconds. The channel occupancy, fig. 5.16, is lower in the flood-and-prune protocol, going from 62.21% to 54.42%.

In scenarios characterised by congestion, the flood-and-prune protocol demonstrated superiority across all metrics with the exception of delay. The elevated delay metrics observed in both versions could be a consequence of prolonged waiting periods for retransmissions, potentially due to congestion. In the case of the flood-and-prune protocol, the slightly higher delay can also be explained by the reduced number of forwarding nodes in highly congested networks with 30 nodes and high traffic, which may increase the average hop distance between some communicating pairs and cause temporary bottlenecks at intermediate relays, contributing to an increase in the delay under heavy load.

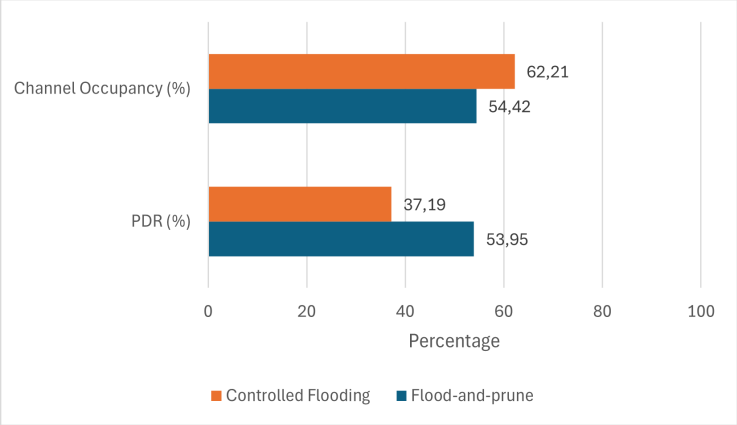


Figure 5.16: Global Channel Occupancy and Global PDR for 30N, 5R and high traffic

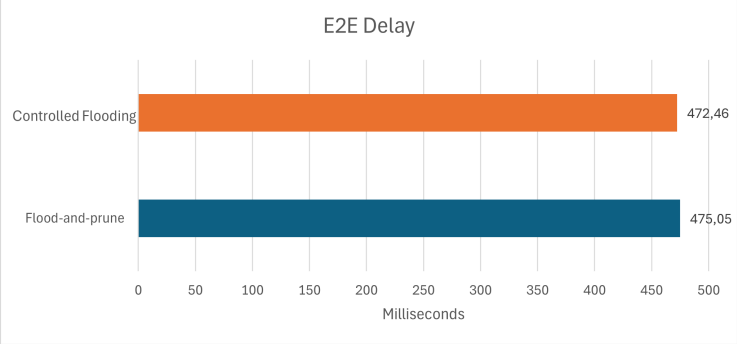


Figure 5.17: Global E2E Delay for 30N, 5R and high traffic

One hypothesis for the PDR being below 50% could be that some receiving nodes did not receive the desired packets. However, an analysis of figures 5.18 and 5.19, which present

the average PDR per receiving node over 20 independent simulation runs, reveals that this is not the case, as all five nodes receive an average number of packets.

In this scenario, characterised by elevated levels of packet traffic, congestion levels tend to be commensurately high. This phenomenon results in a PDR that falls below 50% in both versions of the protocol. Despite this, the flood-and-prune protocol has been shown to achieve a higher PDR, thus demonstrating an advantage in this implementation. This is further compounded by the superior channel occupancy characteristics of the flood-and-prune protocol.

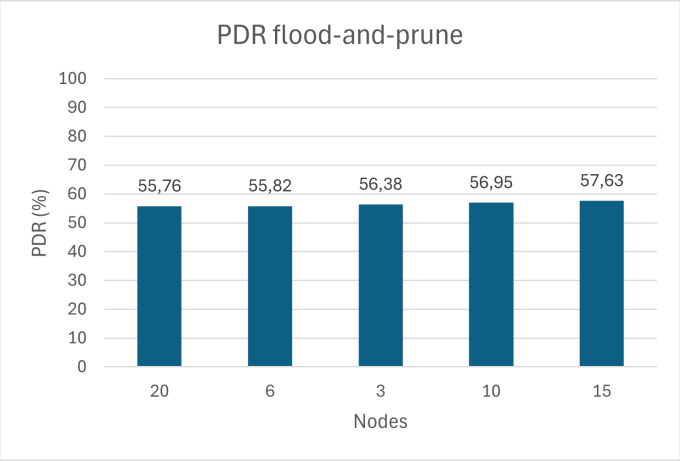


Figure 5.18: PDR for each receiving node with flood-and-prune protocol

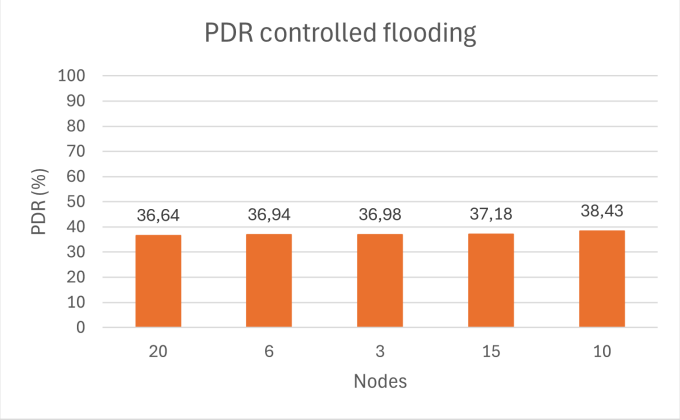


Figure 5.19: PDR for each receiving node with controlled flooding protocol

As the area utilised in this scenario at the beginning was equivalent to that employed in the scenarios with 10 nodes, the network became exceedingly dense, resulting in the outcomes depicted above. For the same scenario above, the area where the nodes are located was increased from 100 m$\times$100 m to 300 m$\times$300 m.

In contradistinction to the preceding scenario, in this case all metric values decreased as a result of the transition from a denser network to a sparser one, so there is less congestion.

The results demonstrate a decline in both delay and channel occupancy in the protocol with the prune, from an average of 4.48 milliseconds to 3.66 milliseconds in delay and from 11.24% to 5.27% in channel occupancy.

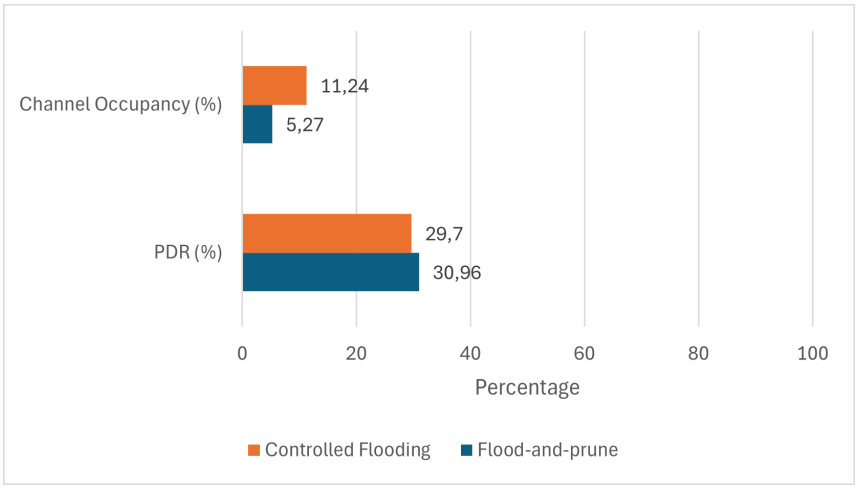


Figure 5.20: Global Channel Occupancy and Global PDR for 30N, 5R, high traffic and area 300 m×300 m

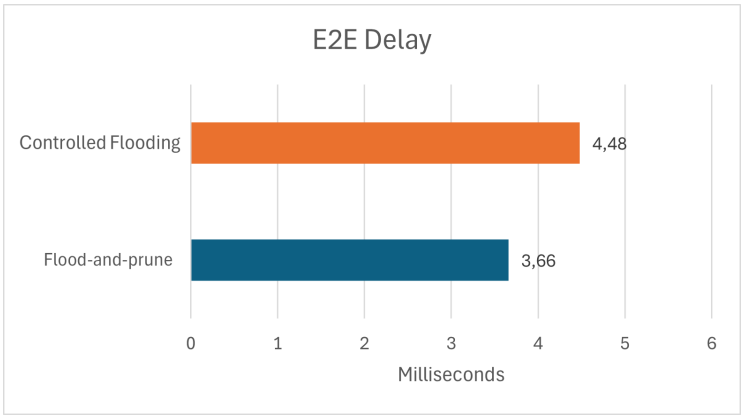


Figure 5.21: Global E2E Delay for 30N, 5R, high traffic and area 300 m×300 m

In this case, the PDR also remains below 50% in both versions, but with the flood-and-prune protocol at 30.95% and the controlled flooding protocol at 29.7%. This phenomenon can be attributed to the presence of isolated receiver nodes, which are characterised by their larger area. Consequently, these nodes are unable to receive the desired packets. As illustrated in the figure 5.22 representing one of the topologies employed in this simulation, node 10, designated a receiving node, is isolated. This is due to the fact that the range of each node is 50m, and in this particular scenario, node 10 is out of range of its neighbours.

This phenomenon can be attributed to the presence of isolated receivers in several of the simulated topologies, rather than a single disconnected node. In some simulation

instances, one or more receivers fall outside the transmission range of their neighbours, resulting in a PDR of 0% for those nodes. When averaged across all 20 runs, these cases significantly lower the global PDR, even though in well-connected topologies the delivery ratio remains substantially higher. As shown in figure 5.22, node 10 illustrates one such case where isolation occurs, but similar situations appear sporadically across different random topologies. Consequently, the overall PDR falls below 50%, as observed in figures 5.23 and 5.24, which represent the average PDR of each node for all different simulations. However, even in these circumstances, the flood-and-prune protocol has been shown to be superior.

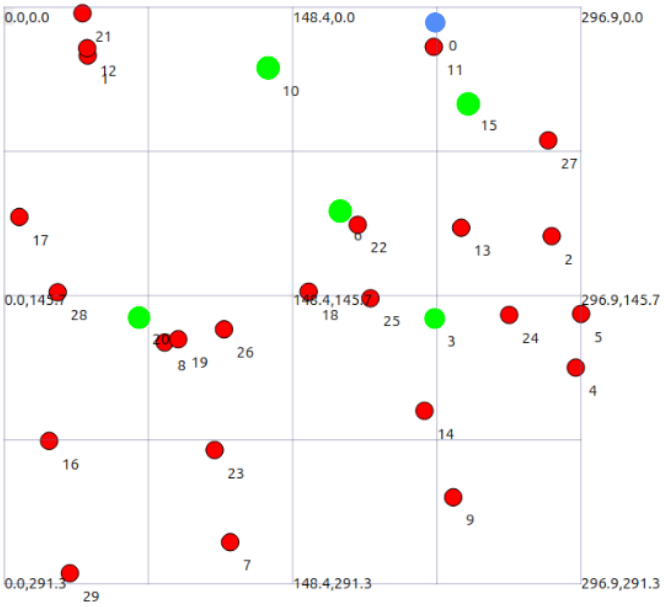


Figure 5.22: Topology of 30 nodes in an area of 300 m×300 m

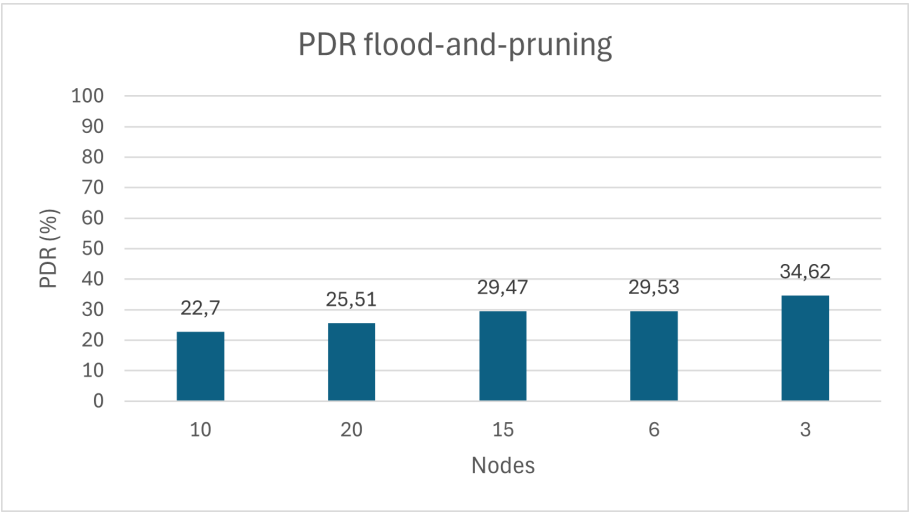


Figure 5.23: PDR for each receiving node with flood-and-prune protocol

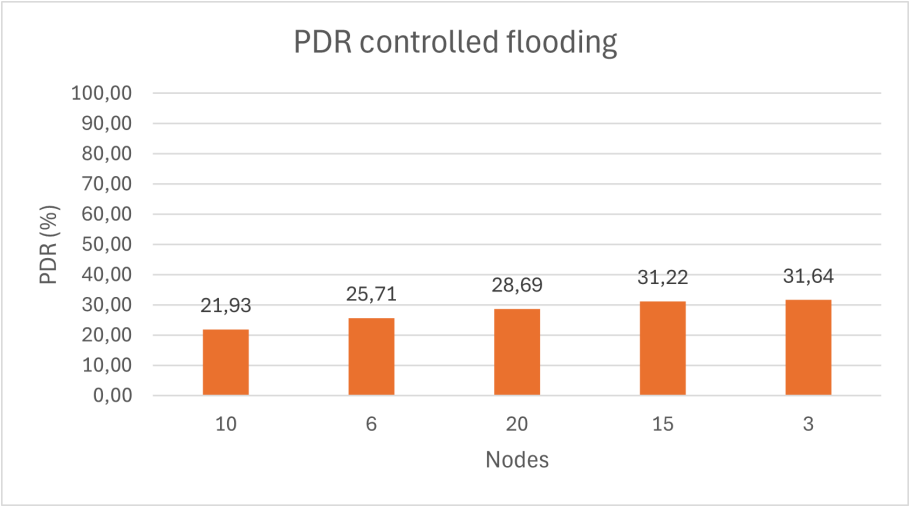


Figure 5.24: PDR for each receiving node with controlled flooding protocol

In the sixth scenario, consisting of 30 nodes, 5 receivers, and a lower packet traffic (30 packets per second), the delivery rate was found consistently high in both simulations, with minimal difference observed between the two, the controlled flooding protocol had an PDR of 94.56%, and the flood-and-prune protocol had 96.28%, fig. 5.25. In contrast, the delay and channel occupancy, fig. 5.25 and fig. 5.26, was lower in the flood-and-prune protocol, the delay went from 66.75 milliseconds to 19.87 milliseconds and the channel occupancy went from 58.14% to 33.62%. As the packet traffic is lower, congestion will be lower too, which explains the results obtained.

In this scenario, where congestion levels are low, the flood-and-prune protocol has been shown to be more advantageous than the controlled flooding protocol. This is because, despite having a practically equivalent PDR, the other metrics demonstrate a clear advantage. This may seem faulty, as pruning is expected to provide greater benefits under higher congestion conditions. However, even at lower packet rates, controlled flooding can still generate redundant transmissions that momentarily increase contention and extend medium access delays. These transient bursts of activity, although insufficient to cause noticeable packet loss, can delay frame transmissions and increase end-to-end latency.

In contrast, the flood-and-prune protocol avoids these short-term congestive effects by suppressing unnecessary broadcasts, which leads to lower and more stable channel occupancy. The resulting decrease in contention allows packets to be transmitted more promptly, explaining the significantly lower delay observed despite similar delivery ratios.

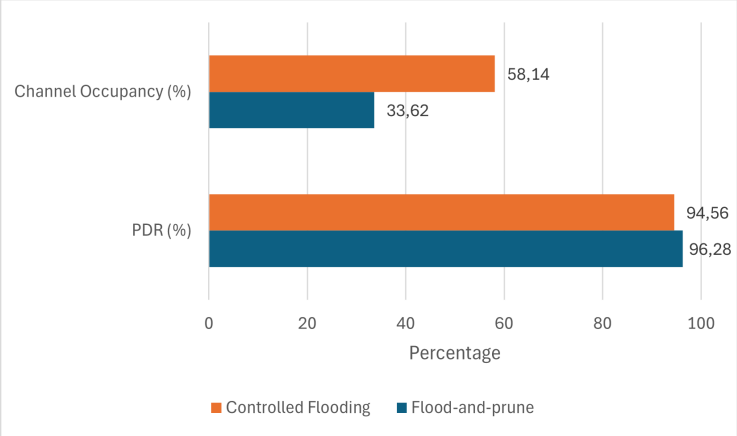


Figure 5.25: Global Channel Occupancy and Global PDR for 30N, 5R and low traffic

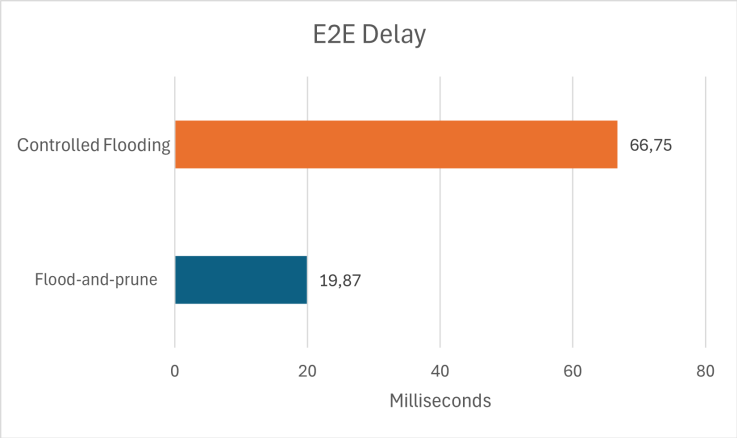


Figure 5.26: Global E2E Delay for 30N, 5R and low traffic

For the seventh scenario, consisting of 30 nodes, 15 receivers, and a high packet traffic (90 packets per second) and area of 100 m$\times$100 m, for both simulations, the PDR was below 50%, fig.5.27, still the flood-and-prune protocol was higher, having 45.74%, and the controlled flooding protocol had 36.62%. The delay, fig.5.28, in the flood-and-prune protocol proved to be greater, where it went from 472.8 milliseconds in the controlled flooding protocol to 637.44 milliseconds. The channel occupancy, fig.5.27, was practically the same, going from 62.21% in the controlled flooding protocol to 61.77%.

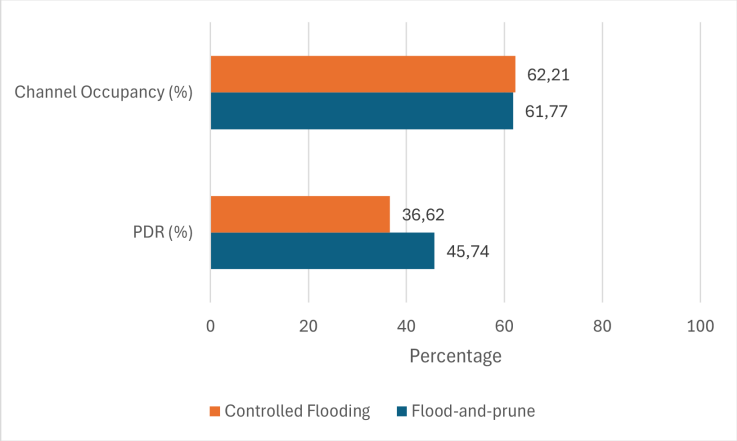


Figure 5.27: Global Channel Occupancy and Global PDR for 30N, 15R and high traffic

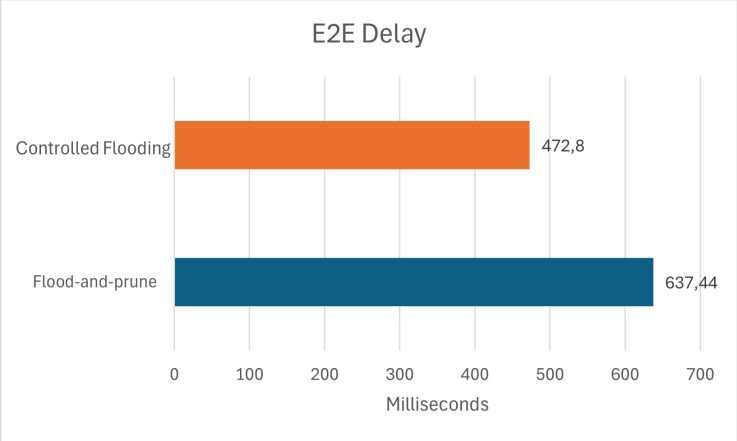


Figure 5.28: Global E2E Delay for 30N, 15R and high traffic

One hypothesis for the PDR being below 50% could be that some receiving nodes did not receive the desired packets. However, an analysis of figures 5.29 and 5.30, which present the average PDR per receiving node over 20 independent simulation runs, as the PDR of each receiving node reveals that this is not the case, as all fifteen nodes receive an average number of packets.

This uniformity of PDR may appear unexpected, as packet delivery in mesh networks typically varies depending on node placement and local interference conditions. However, in this specific topology, the nodes were evenly distributed within a relatively compact area, and the transmission range of each node was sufficient to ensure consistent connectivity across the network. As a result, all receivers experienced similar link quality and contention levels, leading to comparable PDR values.

In this scenario, characterised by elevated packet traffic levels, congestion levels tend to be proportionally high. The phenomenon under discussion results in a PDR that falls below 50% in both versions of the protocol. Despite this, the flood-and-prune protocol

has been shown to achieve a higher PDR, thus demonstrating an advantage in this implementation. This assertion is further substantiated by the superior delay characteristics exhibited by the protocol, wherein the delay is reduced, thereby conferring a distinct advantage.

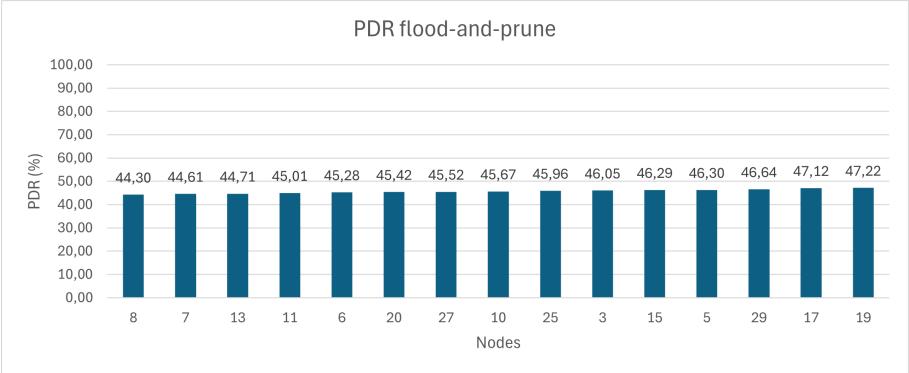


Figure 5.29: PDR for each receiving node with flood-and-prune protocol

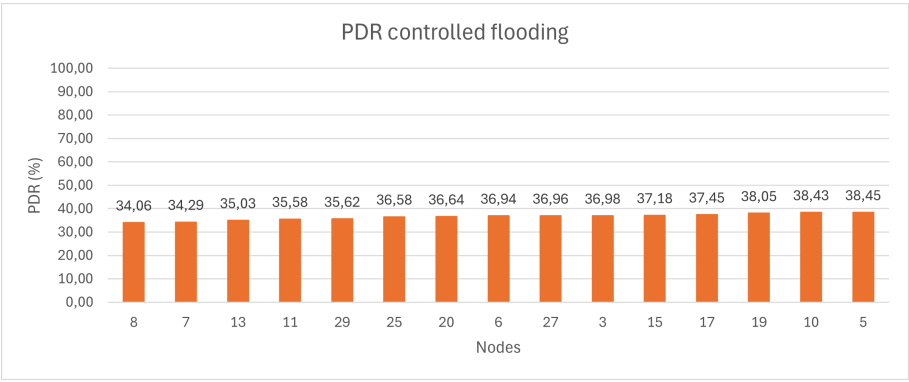


Figure 5.30: PDR for each receiving node with controlled flooding protocol

As the area utilised in this scenario at the beginning was equivalent to that employed in the scenarios with 10 nodes, the network became exceedingly dense, resulting in the outcomes depicted above. For the same scenario above, the area where the nodes are located was increased from 100 m$\times$100 m to 300 m$\times$300 m.

In contrast to the preceding scenario, in this case all metric values decreased due to the transition from a denser network to a less dense one, so there is less congestion. The results demonstrate a decline in both delay and channel occupancy in the protocol with the prune, from an average of 4.73 milliseconds to 4.07 milliseconds in delay and from 11.24% to 7.52% in channel occupancy.

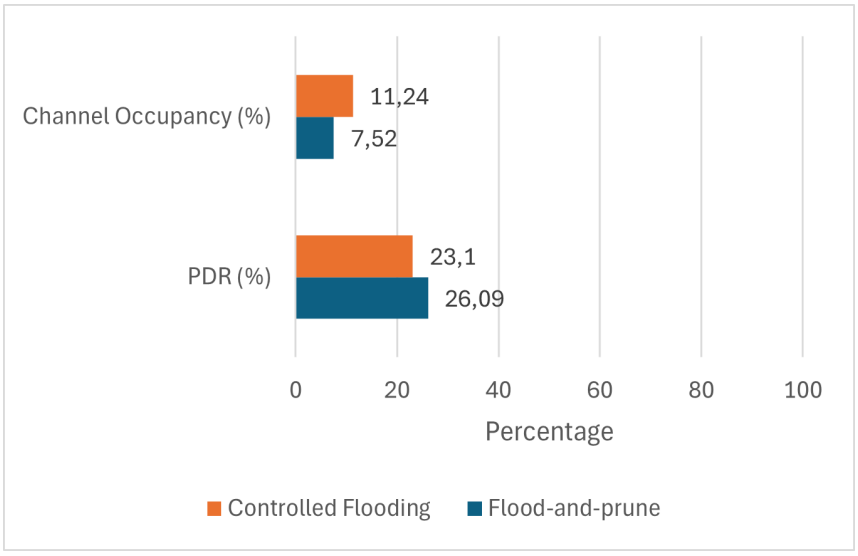


Figure 5.31: Global Channel Occupancy and Global PDR for 30N, 15R, high traffic and area 300 m×300 m

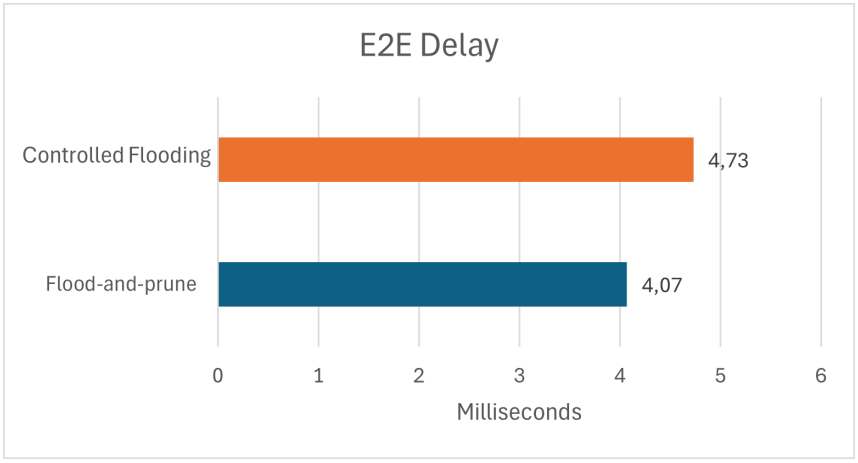


Figure 5.32: Global E2E Delay for 30N, 15R, high traffic and area 300 m×300 m

In this case, the PDR also remains below 50% in both versions, but with the flood-and-prune protocol at 26.4% and the controlled flooding protocol at 23.1%. This behaviour can be explained by the presence of isolated or poorly connected receivers in some of the simulated topologies. Although figures 5.33 and figure 5.34, which represent the average PDR of each node for all simulations, it is important to note that several topologies contained one or more receivers located beyond the 50 m transmission range of their neighbours. In those specific simulations, such nodes achieved a PDR close to 0%, drastically reducing the overall mean.

Because the figures represent averages over multiple random topologies, the isolation of certain nodes does not appear as complete disconnections in the plots, but its cumulative effect explains the globally reduced PDR. In the remaining, well-connected topologies,

the protocols perform consistently better, confirming that the low overall PDR results primarily from topological variability rather than protocol inefficiency. However, even in these circumstances, the flood-and-prune protocol has been shown to be superior.

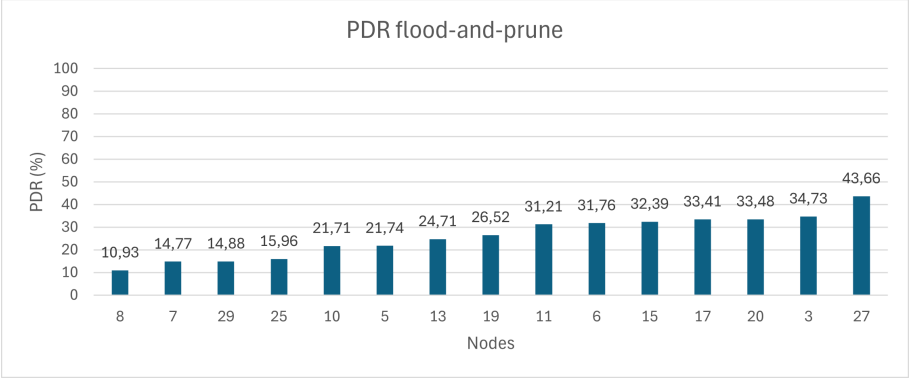


Figure 5.33: PDR for each receiving node with flood-and-prune protocol

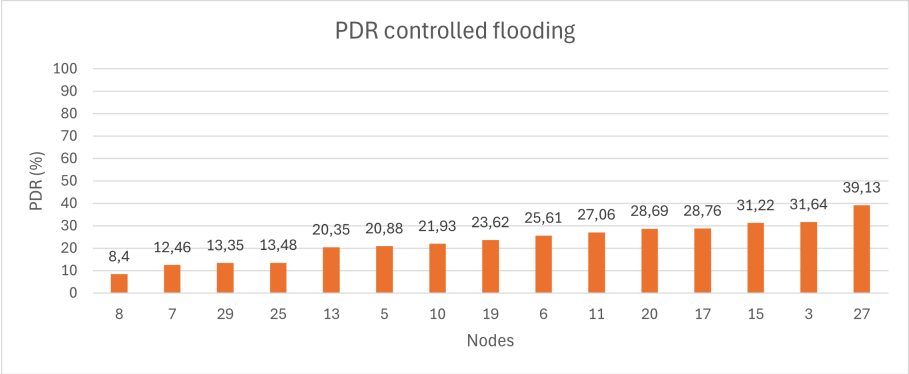


Figure 5.34: PDR for each receiving node with controlled flooding protocol

For the eight scenario, consisting of 30 nodes, 15 receivers, and a lower packet traffic (30 packets per second), it is observed that even though the delivery rate, fig.5.35, for the controlled flooding protocol is high, the flood-and-prune protocol can be higher, going from 94.57% to 97.27%. The delay, fig.5.36, went from 67.02 milliseconds on the controlled flooding protocol to 27.47 milliseconds on the flood-and-prune protocol. The channel occupancy was lower with the protocol with the prune, where it went from 58.41% to 48.91%.

In this scenario, where congestion levels are moderate rather than low, the flood-and-prune protocol has been shown to be more advantageous than the controlled flooding protocol. This is because, despite having a practically equivalent PDR, the other metrics demonstrate a clear advantage.

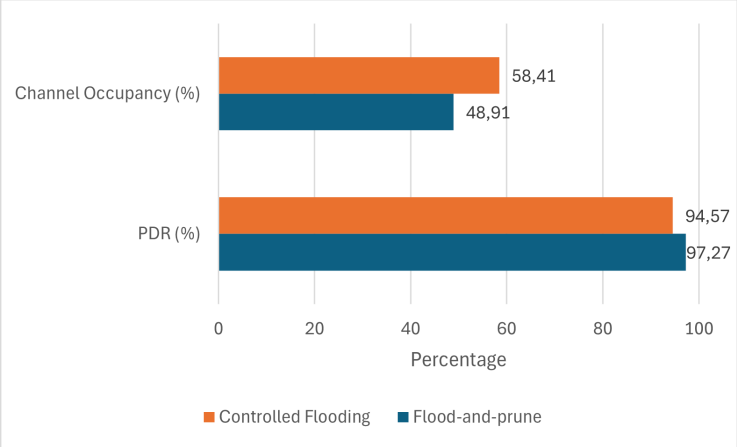


Figure 5.35: Global Channel Occupancy and Global PDR for 30N, 15R and low traffic

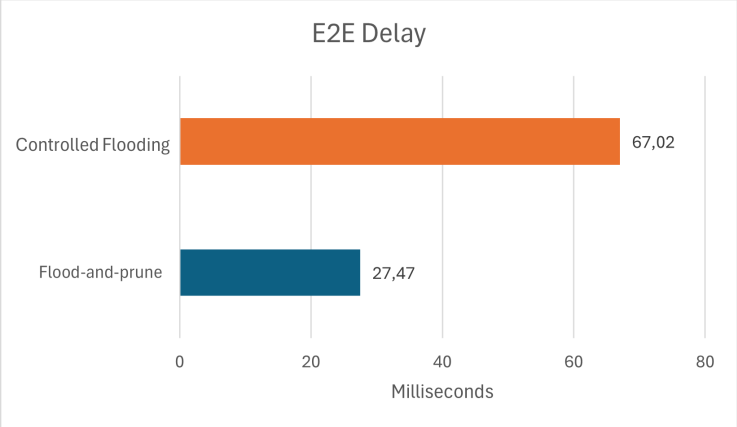


Figure 5.36: Global E2E Delay for 30N, 15R and low traffic

5.2.3 Multicast and Unicast interference

In the context of mesh networks, multicast communication is not usually employed in isolation; frequently, multicast and unicast communications function concurrently on the same network. In order to understand whether the presented protocol offers advantages in terms of multicast and unicast interference, experiments were conducted with both communications occurring concurrently.

For this scenario, the objective was to observe the interaction between multicast and unicast transmission. The 4x3 grid was used for this purpose: figure 5.37 features a blue circle to denote the source, two green circles to indicate the receivers, and yellow circles to mark the nodes participating in unicast traffic.

From this simulation, for multicast packets, we obtain a PDR of 90% for the flood-and-prune protocol and 86% for the controlled flooding protocol. The delay was practically

the same, where the flood-and-prune protocol had a delay of 3 milliseconds and the controlled flooding had a delay of 3.50 milliseconds. For the channel occupancy, there was a difference where it went from 17% with the controlled flooding protocol to 3.6% with the flood-and-prune protocol. To evaluate the interaction between multicast and unicast, the delivery rate of the unicast packets was measured. In this scenario, it was observed that the unicast delivery rate when using the controlled flooding protocol was higher, 59% of packets lost, than when using the flood-and-prune protocol for multicast, 34% of packets lost. The results obtained in the grid scenario demonstrate a clear improvement in the interaction between multicast and unicast transmissions with the introduction of the flood-and-prune protocol. This development has the consequence of facilitating a more stable coexistence of both traffic types, since the multicast transmission no longer has such a severe effect on the performance of the unicast.

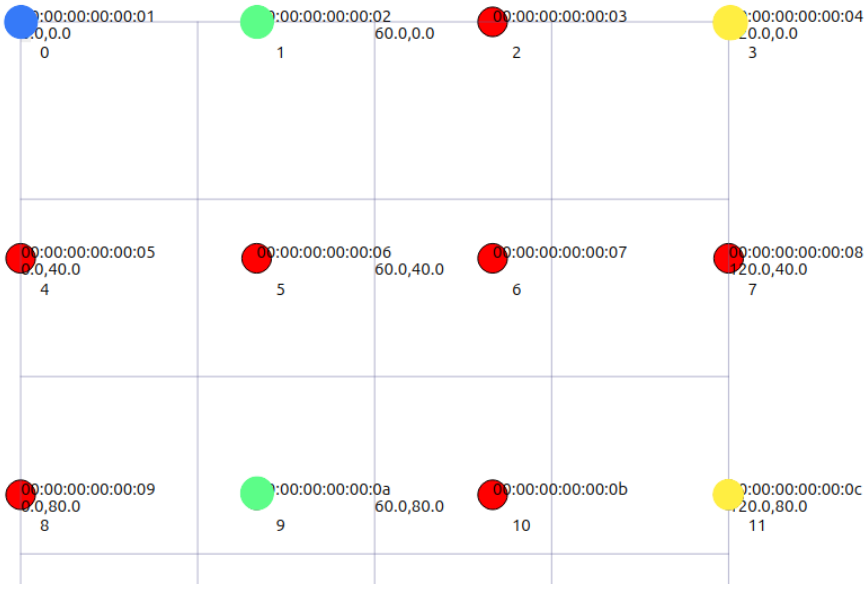


Figure 5.37: Multicast and Unicast interference

In the night scenario, which involves a random topology consisting of 10 nodes with 2 receivers, lower packet traffic (30 packets per second), and unicast communication over UDP between two nodes that do not belong to the multicast group, it can be observed that even with unicast communications at the same time, the PDR in both versions is high, the flood-and-prune protocol had PDR of 82% and the controlled flooding protocol had a PDR of 80%. The delay is lower in the flood-and-prune protocol, going from 12.46 milliseconds in the controlled flooding protocol, to 3.83 milliseconds in the flood-and-prune protocol. The channel occupancy went from 34.34% in the controlled flooding protocol to 21% in the flood-and-prune protocol. In order to measure the interference of multicast in

unicast communications, the delivery rate of unicast communications was calculated. In the controlled flooding protocol, 19% of packets were lost, while in the flood-and-prune protocol, only 6% were lost.

For the tenth scenario, which involves a random topology consisting of 10 nodes with 2 receivers, higher packet traffic (90 packets per second), and unicast communication over UDP between two nodes that do not belong to the multicast group, the PDR in flood-and-prune protocol was 72% while in the controlled flooding protocol was 66%. The delay went from 73.52 milliseconds in the controlled flooding protocol to 7.85 milliseconds with the flood-and-prune protocol. The channel occupancy went from 54% in the controlled flooding protocol to 32% in the flood-and-prune protocol. To measure the interference of multicast in unicast communications, the delivery rate of unicast communications was calculated. The analysis revealed that in the controlled flooding protocol, 58% of unicast packets were lost, whereas in the flood-and-prune protocol, only 31% of unicast packets were lost.

In both scenarios, the flood-and-prune protocol demonstrate a clear improvement in the interaction between multicast and unicast transmissions.

5.3. Conclusion

The experimental results clearly demonstrate the performance improvements achieved by the proposed flood-and-prune mechanism when compared to the standard controlled flooding protocol. Overall, pruning redundant transmissions led to higher delivery ratios, lower latency, and significantly reduced channel occupancy, particularly in dense and high-traffic scenarios.

In the 10 node topologies, the flood-and-prune protocol consistently outperformed the controlled flooding version. Under heavy traffic (90 packets per second), the PDR increased from 86.8% to 94.8%, the average end-to-end delay decreased from 8.4 ms to 3.0 ms, and channel occupancy dropped from 51% to 22%. Even under lower traffic (30 packets per second), where congestion was minimal, the pruning version maintained similar delivery ratios but continued to achieve lower delay and medium usage, confirming its efficiency even in less demanding conditions.

In the larger 30 node networks, the benefits of pruning became more evident. Under high traffic, the PDR improved from 37.2% to 53.9%, while channel occupancy decreased from 62.2% to 54.4%. Although the average delay was slightly higher (475 ms vs 472 ms), this

increase was marginal compared to the substantial gain in delivery performance. When the simulation area was expanded from 100 m$\times$100 m to 300 m$\times$300 m, both protocols experienced lower congestion, the pruning version again performed better, with PDR increasing from 29.7% to 30.9% and channel occupancy nearly halved (from 11.2% to 5.3%).

Even in the most demanding case with 30 nodes and 15 receivers, the flood-and-prune protocol achieved higher delivery (45.7% vs 36.6%) despite the overall PDR remaining below 50% due to extreme congestion. The same trend was observed under lower traffic, where PDR rose from 94.6% to 97.3%, and delay improved from 67 ms to 27 ms.

When evaluating the interaction between multicast and unicast transmissions in the grid topology, unicast loss decreased substantially from 59% with controlled flooding to 34% with the pruning approach. Similar improvements were observed in random topologies, where unicast loss decreased from 19% to 6% in lower traffic and from 58% to 31% in heavy traffic.

In summary, the proposed flood-and-prune mechanism achieved consistent performance gains across all scenarios. These results validate the pruning strategy as a scalable and efficient alternative to traditional controlled flooding in mesh-based multicast networks.

6. Conclusion

The findings obtained throughout this work allow the conclusion to be made that the introduction of the pruning mechanism in the HWMP protocol for IEEE 802.11s mesh networks constitutes a significant improvement in overall network performance, particularly in high traffic load scenarios. A comparative analysis between the original version of the protocol and the modified version indicates that pruning substantially reduces channel occupancy, thereby allowing for more efficient use of the shared medium. This decline in occupancy is directly associated with a decrease in the number of redundant retransmissions, which, in turn, translates into a decrease in the average latency observed in packet delivery. This behaviour is especially relevant in scenarios characterised by high congestion and high node density, in which excessive propagation of multicast frames tends to saturate the network.

With regard to PDR, the results consistently demonstrate that the proposed solution exhibits higher delivery rates in comparison to the original protocol, particularly in scenarios where multicast traffic is substantial or where a limited number of network nodes function as receivers. In scenarios where the system is under a low data load, there is only a small difference in PDR between the pruning version and the original version. However, the pruning version is still competitive due to its lower channel occupancy, which is a crucial aspect for system scalability in larger topologies. In topologies comprising 30 nodes, in scenarios of extremely high traffic, the modified protocol attains more favourable outcomes, thereby demonstrating that the pruning mechanism contributes to a more rational utilisation of network resources. Following the expansion of the network space to 300 m$\times$300 m, a general reduction in congestion was observed, with additional gains in latency and channel occupancy. However, the overall PDR was still conditioned by the spatial distribution of the nodes.

A particular aspect of relevance is the interference between multicast and unicast transmissions. It has been demonstrated through experimental means that the process of pruning has the capacity to significantly reduce the impact of multicast traffic on the performance of unicast transmissions. This reduction in impact has been shown to result in a considerable decrease in the unicast packet loss rate, thereby improving the reliability of concurrent communications. This outcome is of significant practical importance, given that multicast communications rarely occur in isolation in real-world environments.

Despite the manifest advantages, it was found that in dense networks and under very heavy traffic, latency may increase slightly when compared to the original version of the protocol. This behaviour is likely related to secondary effects introduced by the pruning mechanism. Specifically, the reduced number of forwarding nodes can create temporary bottlenecks at intermediate relays, where packets accumulate before transmission. In addition, since the pruning process relies on unicast control exchanges, these transmissions can introduce a small coordination delay under high load. However, this impact is offset by improvements in PDR and channel occupancy, resulting in a favourable overall cost-benefit ratio.

The experimental results obtained from this study provide a high level of evidence that validates the proposed implementation. This suggests that the pruning mechanism functions as expected and contributes significantly to enhancing the performance of IEEE 802.11s mesh networks. Moreover, the mechanism is conceptually simple and computationally lightweight, requiring no substantial additional processing or network resources. These characteristics make it highly suitable for deployment in large-scale or resource-constrained environments. Overall, the findings suggest that the use of intelligent multicast retransmission suppression techniques is a promising avenue for the evolution of forwarding protocols in multi-hop wireless networks, allowing for improved scalability, reduced resource consumption, and increased overall communication efficiency.

The objectives defined at the beginning of this work were fully achieved. A comprehensive bibliographic survey and state-of-the-art analysis were conducted, which allowed for the identification of gaps in the implementation of multicast in the HWMP protocol in the ns-3 simulator. Following a thorough analysis, a broadcast-and-prune mechanism was designed and specified, taking into account the characteristics of the protocol. This mechanism was implemented in ns-3, followed by extensive experimentation that included topologies of different sizes, scenarios with varying traffic intensity, and cases with interference between multicast and unicast. A detailed analysis of the results enabled the assessment and quantification of the impact of the proposed solution. This demonstrated the improvement in performance obtained and answered the central question that motivated the development of this dissertation.

6.1. Difficulties

Throughout the development of this work, a number of challenges were encountered that had a considerable impact on both the research and implementation process. A

primary challenge encountered pertained to the scarcity of comprehensive documentation and resources elucidating the operational mechanisms of multicast functionality within the HWMP implementation in the ns-3 simulator. The ns-3 simulator contains numerous source files, each comprising hundreds of lines of code, which may contain pertinent information for this project. A substantial investment of time and effort was necessary to analyse these files, with the objective of identifying which portions were truly relevant to the study's objectives.

A further substantial impediment materialised as a consequence of an inherent problem within the NS-3 simulator itself. In the initial phases of experimentation, it became apparent that the transmission of mesh multicast frames was not occurring in accordance with the anticipated parameters. Following a thorough investigation, it was determined that multicast frames were incorrectly classified as unicast, rather than group-addressed frames (equivalent to broadcast). The majority of the forwarding code is designed to check for broadcast addresses; therefore, this misclassification resulted in multicast frames not being transmitted, as the corresponding Path Request messages did not receive a Path Response. The implementation of the solution, as outlined in a Git issue reported by a former student, successfully resolved the issue and enabled the simulations to produce the expected results.

Finally, the absence of 802.11s support in NetAnim posed an additional difficulty. In the absence of graphical visualisation, comprehension of the packet flow relied exclusively on the analysis of log outputs, a process that was substantially more time-consuming. The provision of a visual interface would have facilitated a more rapid and clearer comprehension of the network behaviour during the simulation.

6.2. Future Work

There are several opportunities for future work that could extend and enhance the results presented in this dissertation. Initially, the present implementation presupposes a solitary source node, and in certain scenarios where the source address was required, it was hardcoded. It is recommended that future work extend this logic in order to support multiple source nodes. This would allow for a more general and flexible evaluation of multicast behaviour in mesh networks.

Another significant direction for future research would be to develop a more user-friendly approach for configuring which nodes belong to a multicast group within HWMP. This

would serve to simplify the process of defining multicast memberships and make experimentation more accessible.

From a metric perspective, the current use of the EWMA could be further refined by incorporating the EWMA values of neighbouring nodes. This would provide a more accurate estimation of the relative position of nodes in the network. As it stands, the TTL values may fluctuate depending on the chosen path, leading to oscillations in the perceived relative positioning of nodes.

It is evident that the present implementation of the prune mechanism is such that prune messages are forwarded to nodes that are either upstream or at the same level in the topology. It would be beneficial to investigate alternative approaches, such as the exclusive transmission of prune messages to upstream nodes, and to analyse which strategy yields superior performance in terms of network efficiency and packet delivery.

A potential area for future research would be the implementation and evaluation of a graft mechanism, similar to the one defined in PIM-DM, 2.6, which facilitates the reactivation of previously pruned branches when new receivers join the multicast group. This feature was not addressed in the present work, but it has the potential to enhance the adaptability of the pruning process, thereby reducing the delay in re-establishing connectivity and improving overall multicast efficiency in dynamic network scenarios.

Further research could be directed towards extending the present study to analyse the interaction between multicast and unicast traffic when utilising the TCP transport protocol. All experiments presented in this work were conducted with UDP, which avoids congestion control and thus isolates the effects of the routing protocol and medium access. However, the introduction of TCP flows would provide a more realistic and complex scenario, as TCP's congestion avoidance and retransmission mechanisms are highly sensitive to packet loss and delay variations. The evaluation of the proposed flood-and-prune mechanism is of particular interest in this context, as it has the potential to offer novel insights into fairness, throughput stability, and overall network efficiency under mixed traffic conditions.

In the process of implementing a forwarding protocol that supports PRUNE messages, the issue of interoperability naturally arises in networks where not all nodes support this new mechanism. In a hybrid network, with some nodes capable of interpreting PRUNEs and others operating in the traditional manner, it is expected that compatible nodes will

start sending PRUNE messages whenever they no longer wish to receive a particular multicast stream. However, since the old nodes do not recognise this type of message, they continue to forward packets as normal. This results in a scenario where the node that supports PRUNE is compelled to transmit successive pruning messages whenever it receives undesirable traffic. However, this has no practical effect, since its neighbours do not modify their behaviour. This results in an inefficient utilisation of resources, as the node expends time and bandwidth signalling information that will not be observed, and the flooding remains unaltered. In order to address this issue, a potential solution would be to implement a memory mechanism within each node, with the purpose of storing status information for each neighbour. This information would include the list of known peers, the number of PRUNEs sent, and even a maximum rate of messages that can be sent to a given neighbour. Consequently, following a predetermined number of unsuccessful attempts, the node is able to infer that the neighbouring node does not support pruning. This results in the cessation of the transmission of new messages, thereby preventing the unnecessary consumption of signalling traffic. This mechanism would also enable the node to differentiate between neighbours that respect pruning and those that continue to forward traffic regardless, adopting different strategies for each case. Another potential enhancement to interoperability would be the implementation of an explicit capability indicator at each node, for example, a simple Boolean that indicates whether or not it supports PRUNE. In the event of a node receiving signalling packets, it would be possible for it to check this flag in order to decide whether to process or ignore the pruning code. In this manner, nodes that are incompatible would simply bypass this logic, thus ensuring the retention of classic behaviour. In contrast, updated nodes would take full advantage of this novel mechanism.

It would be beneficial to conduct an experimental study to assess the extent to which this type of hybrid approach can reduce flooding on the network. It is hypothesised that simulations with different percentages of PRUNE compatible nodes could demonstrate the point at which a significant improvement in metrics such as delivery rate, bandwidth consumption, or the number of control messages sent becomes apparent. The results of such study would assist in determining the minimum coverage required to substantiate the effective implementation of the new protocol, in addition to providing guidance for a phased network migration strategy. Consequently, even in a scenario where the upgrade is not complete, it would be possible to obtain practical benefits and pave the way for a progressive transition to a fully pruning-compatible network, gradually reducing flooding

and improving multicast forwarding efficiency.

The propagation model currently employed is deterministic, which means that for a given scenario and distance between transmitters and receivers, the outcome is always consistent. However, in reality, the behaviour of the communication channel is much more unpredictable. It has been demonstrated that fading phenomena and signal variations can result in the reception of a packet being lost for a period of time, followed by its subsequent reception in a subsequent moment, despite the apparent connection conditions remaining relatively unchanged. These fluctuations have a direct impact on the results obtained in the simulations, since packet delivery or loss is not only a consequence of topology or distance, but also of random factors that affect channel quality. It would therefore be a worthwhile experiment to introduce these effects into the model, in order to understand how they interfere with network performance metrics. The incorporation of fading and other sources of variability would facilitate the observation of whether the results remain consistent or whether unexpected behaviours arise, such as fluctuations in delivery rates or changes in route stability.

References

[1] K. K.C, “Wireless mesh network: A survey,” in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016, pp. 1966–1970. [Cited on pages 5 and 7.]

[2] C. Zachos, J. Loo, and S. Khan, *Wireless Mesh Network: Architecture and Protocols*, 04 2016, pp. 425–447. [Cited on pages vii, 6, and 7.]

[3] S. Srivathsan, N. Balakrishnan, and S. S. Iyengar, *Scalability in Wireless Mesh Networks*. London: Springer London, 2009, pp. 325–347. [Online]. Available: https://doi.org/10.1007/978-1-84800-909-7_13 [Cited on page 6.]

[4] S. Banerji, “On IEEE 802.11: Wireless LAN technology,” *CoRR*, vol. abs/1307.2661, 2013. [Online]. Available: http://arxiv.org/abs/1307.2661 [Cited on page 10.]

[5] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, “Ieee 802.11s: The wlan mesh standard,” *IEEE Wireless Communications*, vol. 17, no. 1, pp. 104–111, 2010. [Cited on pages vii and 10.]

[6] J. G. Jetcheva and D. B. Johnson, “Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks,” in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, ser. MobiHoc ’01. New York, NY, USA: Association for Computing Machinery, 2001, p. 33–44. [Online]. Available: https://doi.org/10.1145/501422.501423 [Cited on page 11.]

[7] M. G. Sung-Ju Lee, William Su, “On-demand multicast routing protocol in multihop wireless mobile networks,” *Kluwer Academic Publishers*, 2002. [Online]. Available: https://doi.org/10.1023/A:1020756600187 [Cited on page 11.]

[8] G. He, “Destination-sequenced distance vector (dsdv) protocol,” 2002. [Online]. Available: https://api.semanticscholar.org/CorpusID:12383459 [Cited on page 12.]

[9] C. Perkins and E. Royer, “Ad-hoc on-demand distance vector routing,” in *Proceedings WMCSA’99. Second IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90–100. [Cited on page 12.]

[10] L. Junhai, Y. Danxia, X. Liu, and F. Mingyu, “A survey of multicast routing protocols for mobile ad-hoc networks,” *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 78–91, 2009. [Cited on page 13.]

[11] netseccloud.com, “Asm multicast vs. ssm: Which one should you use?” https://netseccloud.com/asm-multicast-vs-ssm-which-one-should-you-usel, 2024. [Cited on page 14.]

[12] M. S. Islam, M. A. Hamid, and C. S. Hong, *SHWMP: A Secure Hybrid Wireless Mesh Protocol for IEEE 802.11s Wireless Mesh Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 95–114. [Online]. Available: https://doi.org/10.1007/978-3-642-10649-1_6 [Cited on pages vii, 15, and 16.]

[13] J. Nicholas, A. Adams, and W. Siadak, “Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised),” RFC 3973, Jan. 2005. [Online]. Available: https://www.rfc-editor.org/info/rfc3973 [Cited on page 19.]

[14] G. T. Katia Obraczka, Kumar Viswanath, “Flooding for reliable multicast in multi-hop ad hoc networks,” 2001. [Online]. Available: https://doi.org/10.1023/A:1012323519059 [Cited on pages 20 and 28.]

[15] A. Kaur and G. Kaur, “Highlighting the efficient multicast routing algorithms in mobile ad-hoc network,” in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 1, 2016, pp. 1–6. [Cited on pages 22 and 28.]

[16] A. Rezaei and L. Farzinvash, “Enhancing multi-layer multicast routing in wireless mesh networks by exploiting wireless broadcast advantage,” in *2020 28th Iranian Conference on Electrical Engineering (ICEE)*, 2020, pp. 1–7. [Cited on pages 23 and 28.]

[17] R. S. Majumder, A., “Implementation of forward pointer-based routing scheme for wireless mesh network,” 2015. [Online]. Available: https://doi.org/10.1007/s13369-015-1965-6 [Cited on pages 24 and 28.]

[18] S. Seetha, S. A. John Francis, E. G. Mary Kanaga, E. Daniel, and S. Durga, “A framework for multi-constraint multicast routing in wireless mesh networks,” in *2019 5th International Conference on Advanced Computing Communication Systems (ICACCS)*, 2019, pp. 445–451. [Cited on page 25.]

[19] M. L. Sanni, A.-H. A. Hashim, W. H. Hassan, F. Anwar, and G. S. M. Ahmed, “Optimized routing algorithm for mobile multicast source in wireless mesh networks,” in *2015 11th International Conference on Innovations in Information Technology (IIT)*, 2015, pp. 104–109. [Cited on pages 26 and 28.]

[20] T. Chavarría, “Exponentially weighted average,” Mar 2022. [Online]. Available: https://medium.com/@tobias-chc/exponentially-weighted-average-5eed00181a09 [Cited on page 29.]

[21] “Ieee standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications,” 2020. [Online]. Available: https://standards.ieee.org/ieee/802.11/7028/ [Cited on page 32.]