Optimizing Routing Performance in Flying Ad-hoc Networks using an Adaptive Hello Interval Scheme

by

Telavane Jui Sunil
202111062

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

December, 2023

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree

ii) due acknowledgement has been made in the text to all the reference material used.

Telavane

Telavane Jui Sunil

Certificate

This is to certify that the thesis work entitled "Optimizing Routing Performance in Flying Ad-hoc Networks using an Adaptive Hello Interval Scheme" has been carried out by TELAVANE JUI SUNIL for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.

28-12-2023

Prof. Yash Vasavada
Thesis Supervisor

Acknowledgments

I would start by expressing my gratitude to my research supervisor and mentor, Prof. Yash Vasavada, for his guidance and constant support in every aspect of my research work. His continuous guidance and dedicated involvement always encouraged me whenever I faced obstacles. Also, his humbleness and focused vision steered me in the right direction. He taught me a proper methodology for conducting research. I am grateful to him for providing me with a workspace and making me a part of the Communication, Signal Processing and Machine Intelligence Lab (CSPMI Lab). I also thank him for providing me with a workstation to conduct my research.

I also want to express my gratitude to the lab members for their support throughout my thesis work. Their willingness to engage in brainstorming and technical discussion helped me to get clarity in my moments of confusion. Getting through my M. Tech curriculum and research required more than just academic support, and I thank my friends and the entire batch of M.Tech-2021 for helping me throughout the course of two years. I would also like to thank my friends Vijay Banerjee, Dhara Mehta, Arunanghsu Dutta, Prabhanshu Yadav, Avinash Subramaniam, Radha Agrawal, Kavan Hajare, Khushali Ratanghayara and many others for sticking with me through thick and thin and making the entire journey a memorable one. Above all, I would like to thank my parents and the whole family for constantly believing in me and pushing me throughout the process.

Contents

Abstract

Hello messaging is a widely used scheme to obtain local link connectivity information. Traditional routing protocols that are used for FANETs scenarios, including the well-known Ad hoc On-Demand Distance Vector Routing (AODV) and Optimized Link State Routing (OLSR), make use of fixed hello interval for periodic hello messages to realize the change in the topology to maintain the local connections up-to-date. However, it involves a tradeoff where a shorter value of hello interval ensures quick detection of link changes but also leads to an increase in the overhead and energy consumption. On the other hand, a longer hello interval reduces overhead and energy consumption but compromises the ability to discover new neighbors and detect link breaks promptly. One of the approaches to balance this tradeoff is to make the hello interval scheme adaptive such that the value of the hello interval is not fixed but is adjusted according to the network conditions.

This work proposes an adaptive hello interval scheme which sets the hello interval based on three network parameters, namely the transmission range of the UAV, the network density of the node, and the relative speed of UAV with respect to neighboring UAVs. Considering relative speed ensures that the UAV evaluates the movement dynamics of nearby UAVs when setting the hello interval. This refinement contributes to improved performance in terms of throughput and overhead efficiency while simultaneously reducing the network's overhead and energy requirements.

List of Tables

List of Figures

CHAPTER 1

Introduction

Flying Ad-hoc Networks (FANETs) have emerged as a promising technology in recent years, offering unique capabilities for communication in aerial environments. FANETs have numerous applications such as disaster management, surveillance, military reconnaissance, film-making, and pollution monitoring.

These networks are characterized by various aspects such as dynamic network topology and resource constraints such as energy consumption and limited bandwidth. Single UAV applications include search and rescue, patrolling, delivery of goods, military, and civil, whereas multi UAV applications include border monitoring, civil security, agriculture, remote sensing estimate, wind estimation, relay network, destruction and search operation, crisis, etc [4]. A co-operative system of Multiple UAVs through an Ad-hoc wireless network that connects UAVs and GCS (ground control station) is known as Flying Ad-hoc Network (FANET). While FANETs share similarities with other types of Ad-hoc networks like Mobile Ad-hoc Networks (MANETs) and Vehicular Ad-hoc Networks (VANETs), they also exhibit several distinct characteristics in terms of connectivity, quality of service, sensor type, node movement features, data delivery, etc [12].

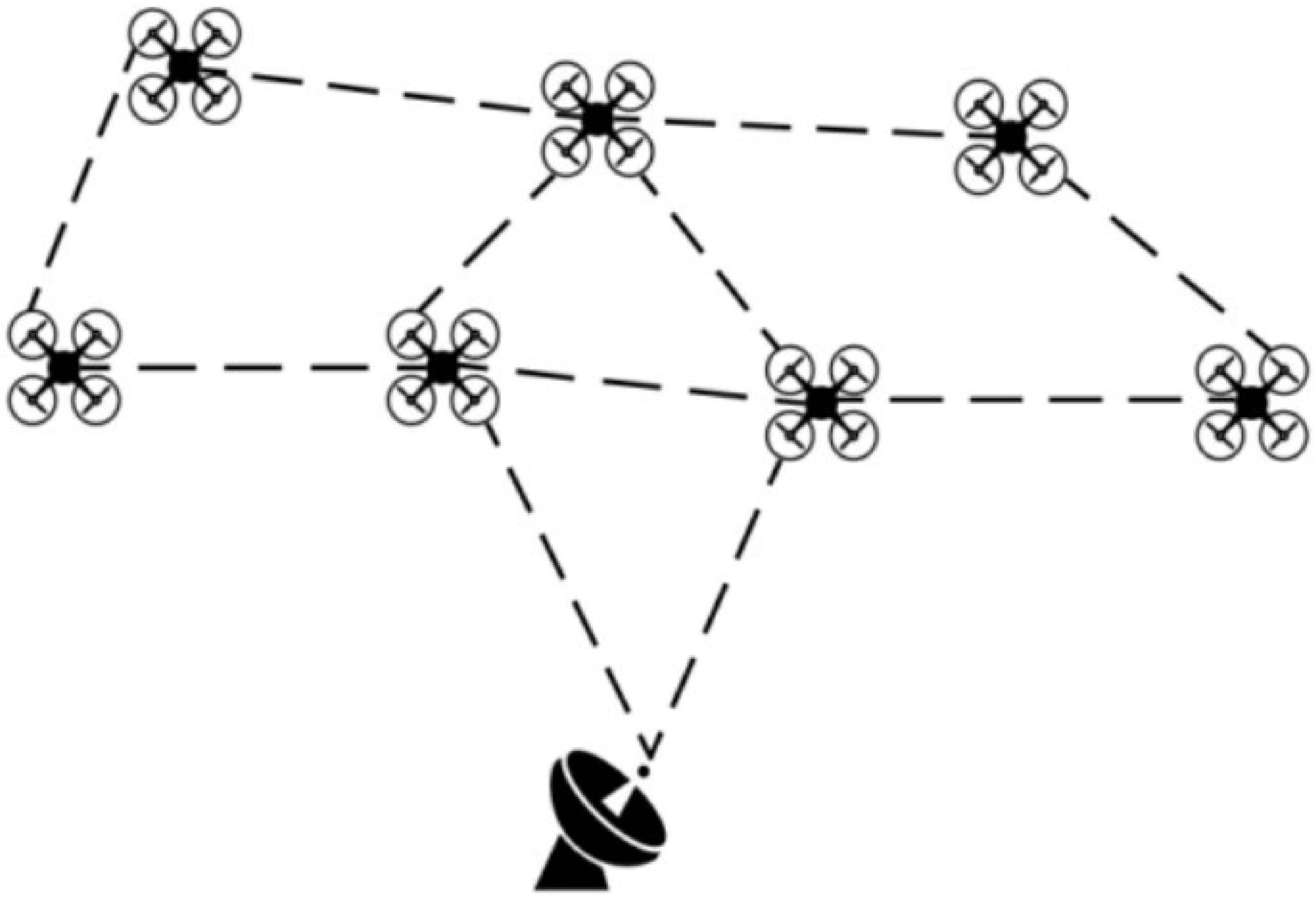


Figure 1.1: Flying Ad hoc Network (FANET)

Understanding these differences is crucial for developing efficient and optimized protocols for FANETs [22]. Here, we discuss some of the key aspects that set FANETs apart from other Ad-hoc network types:

1.1 Features of FANETs

1. **Mobility of Nodes:** One of the defining features of FANETs is the mobility of nodes, i.e., the UAVs. Unlike MANETs and VANETs, where nodes primarily move on the ground, FANETs operate in three-dimensional airspace. The mobility patterns of UAVs greatly influence network connectivity, routing, and communication reliability in FANETs.

2. **Altitude and Range Considerations:** In FANETs, UAVs can operate at different altitudes, ranging from a few meters above the ground to several kilometres high. This altitude variability introduces challenges related to transmission range, energy consumption, and link quality. Addressing these challenges requires specialized routing and communication protocols tailored to the unique characteristics of FANETs.

3. **Dynamic Network Topology:** Due to the mobility of UAVs, the network topology in FANETs experiences frequent and rapid changes. Nodes enter and leave the network, and new links are established or broken. This dynamic nature of the network topology demands efficient routing protocols capable of adapting to changing conditions in real-time.

4. **Limited Energy Resources:** UAVs in FANETs are typically equipped with limited energy resources, such as batteries. Efficient utilization of energy becomes crucial to extend the network's operational lifetime. Routing protocols need to consider energy awareness to optimize power consumption and enhance the overall network performance.

5. **Communication Interference:** FANETs often operate in shared frequency bands, leading to potential interference from other coexisting wireless systems. Effective interference management techniques and intelligent routing algorithms are essential to mitigate the impact of interference on FANET communication.

6. **High Node Density:**

   In certain scenarios, FANETs can exhibit high node densities, with multiple UAVs operating in close proximity. This high density introduces challenges related to contention, collision avoidance, and efficient resource allocation. Protocols should be designed to handle dense networks effectively.

7. **Scalability and Network Size:** FANETs can vary in terms of network size, ranging from small-scale deployments to large-scale systems involving hundreds or thousands of UAVs. Scalability becomes a critical factor for ensuring efficient network operations, managing control overhead, and accommodating a growing number of nodes.

8. **Security and Privacy:** Security and privacy are vital concerns in FANETs, given the sensitive nature of the data transmitted and the potential for malicious attacks. Robust security mechanisms, authentication protocols, and encryption techniques need to be integrated into the routing protocols to protect the network from unauthorized access and data breaches.

9. **Quality of Service (QoS) Requirements:** Certain FANET applications, such as video streaming or real-time data delivery, demand specific quality of service parameters, including low latency, high throughput, and reliable transmission. Routing protocols should be optimized to meet these QoS requirements while considering the unique characteristics of FANETs.

10. **Network Localization and Self Organization:** FANETs often operate without any centralized infrastructure, relying on distributed algorithms for network localization and self-organization. These algorithms enable autonomous UAVs to determine their positions, discover neighboring nodes, and establish efficient routes dynamically.

1.2 Need of Routing Protocol Optimization in FANETs

Routing protocols play a crucial role in determining the efficiency, reliability, and overall performance of FANETs. The above-discussed differences between FANET and conventional Ad-hoc networks make it necessary to alter the current routing protocols so that they can accommodate all of the unique characteristics of FANET and carry out data routing appropriately. By tailoring routing protocols specifically for FANETs, we can address these issues and achieve efficient and reliable communication in aerial networks. Optimized routing protocols can adapt to dynamic network conditions, incorporate efficient data dissemination strategies, and mitigate the impact of node mobility, ensuring seamless connectivity and improved overall network performance in FANET environments.

The routing protocols can be optimized by enhancing various aspects of their working, such as optimizing load balancing to distribute the network traffic evenly among UAVs by intelligently distributing communication loads [1], Performing link quality estimation by estimating metrics such as signal strength, packet loss, and channel conditions, enables the routing protocols to select better routes [15], [16], [11], or performing multi-metric routing by considering multiple network parameters, such as energy, delay, link quality, and residual bandwidth, during route selection [26]. By optimizing routing decisions based on these metrics, FANETs can achieve better load balancing, improved QoS, and enhanced network performance.

One such aspect that can be considered while optimizing the routing protocols to make them suitable for FANETs is the idea of making hello interval adaptive. By making the hello interval adaptive, the network can dynamically adjust the frequency of neighbor discovery, reducing control overhead and improving energy efficiency. The concept of routing and hello interval are discussed in detail in the further section.

CHAPTER 2

Background

Ad-hoc routing protocols can be broadly classified into three types: proactive, reactive, and hybrid. Each type offers different trade-offs in terms of routing overhead, latency, and scalability. Each of these types is explained further, along with examples of some well-known protocols of each type [13].

2.1 Types of Wireless Ad-Hoc Network Routing Protocols

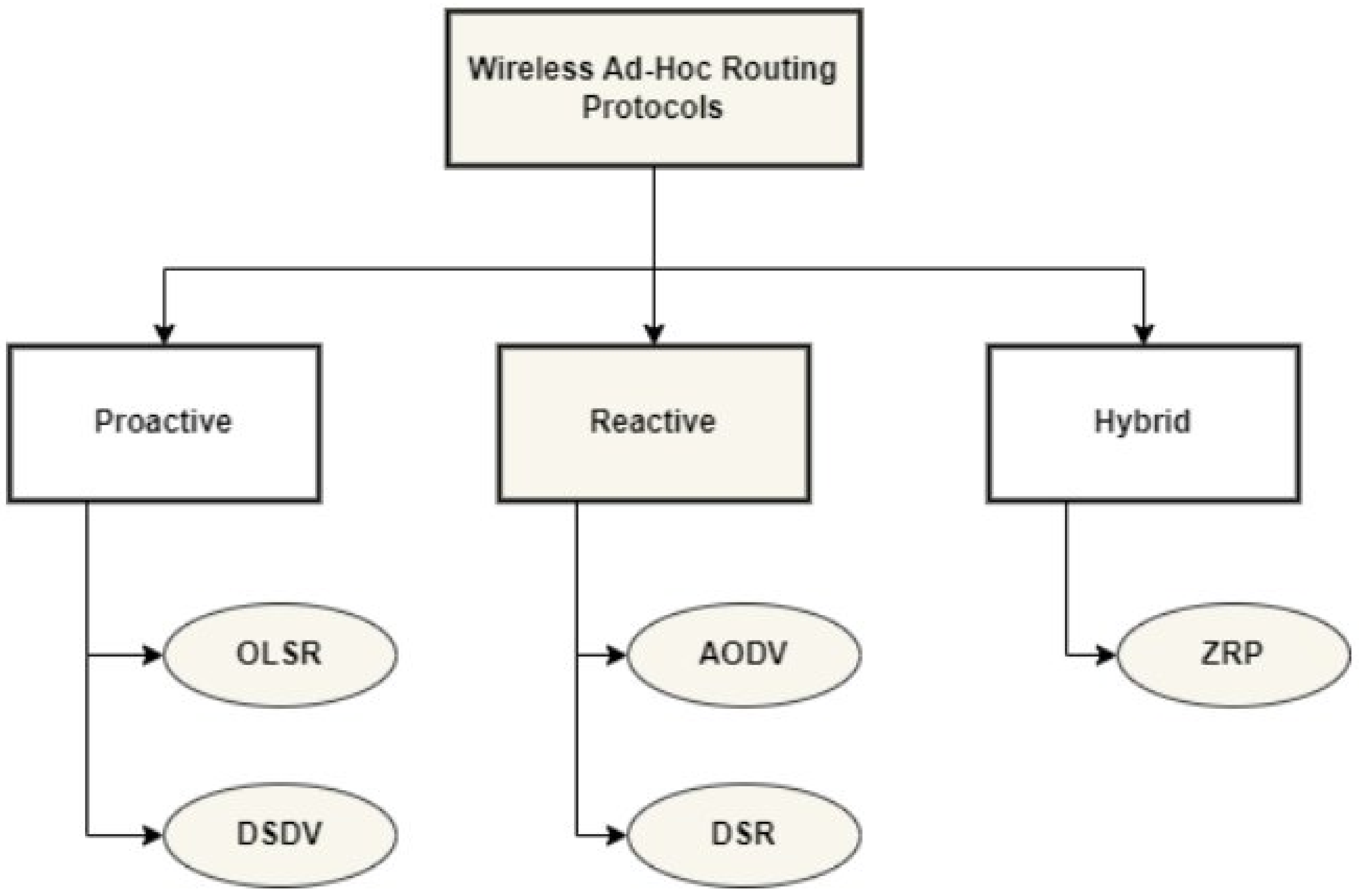


Figure 2.1: Classification of Wireless Ad-hoc Network Routing Protocols

2.1.1 Proactive Routing Protocols

They are also known as table-driven protocols because they use a table to record all routing information about each node in the network before sending the data packets. At each topology change, the tables must be updated. It is easy to choose a path from the source to the destination because this sort of routing provides the most recent information on the routes that are stored in the routing table, thereby significantly lowering the delivery wait [7].

However, there are several obvious weaknesses. Firstly, because there are so many messages sent back and forth between nodes, routing techniques are unable to effectively use bandwidth, which is a limited resource in FANET networks. Secondly, it reacts slowly whenever a failure or a change in topology happens. Therefore, proactive protocols may be appropriate if, and only if, some important updates are implemented.

In proactive routing, the commonly used protocols are OLSR (Optimized Link State Routing) and DSDV (Destination-Sequenced Distance Vector). These protocols employ proactive network maintenance, distributing updated routing information throughout the network. While there are several other protocols in this category, OLSR and DSDV have gained significant popularity for their efficient, proactive routing strategies.

2.1.2 Reactive Routing Protocols

They are known as on-demand protocols because, in this type of protocols, a route is established only when a node wants to join other nodes (i.e., to send a packet). This protocol can address the proactive routing protocol's overhead problem. The messages that are exchanged are of two different types, namely route requests messages and route reply messages [7]. Route request messages are sent by the sender node to all the neighbor nodes using the flooding technique in order to find the path to the destination. Each node in the neighbor further uses the same technique until the route request message reaches the destination node.

The route reply message is created and sent by the destination node to the source node but this is a normal unicast message and not a broadcast message as in the case of the route request message. In this situation, each node saves the currently used path, not all the paths. Hence, there is no need to update all tables in the network. Most of the time, this category experiences high delay and latency times as a result of the discovery process, which also results in a sizable overhead, especially when the network is severely fragmented.

The most widely used reactive routing protocols in Ad hoc networks are AODV (Ad hoc On-Demand Distance Vector) and DSR (Dynamic Source Routing). These protocols are renowned for their on-demand route discovery mechanisms and are among the prominent options in this category, although there are several other protocols available.

2.1.3 Hybrid Routing Protocols

As the name suggests, hybrid routing protocols are a combination of proactive and reactive protocols and help to address their shortcomings in them [7]. This protocol can be used to lower the overhead of control messages in proactive protocols and to decrease the latency of the initial route identification phase in reactive protocols. These types of protocols are especially adaptable for large networks. The network is divided into a number of zones, with intra-zone routing carried out using the proactive approach and inter-zone routing carried out using the reactive method. This sort of protocol is very flexible for big networks.

Information is difficult to acquire and preserve in FANET due to the mobility of nodes and connection behavior. As a result, it is challenging to change routing strategies. ZRP (Zone Routing Protocol) is a widely recognized hybrid routing protocol that combines features of both reactive and proactive protocols. By dividing the network into zones, ZRP employs reactive routing within each zone and proactive routing between zones. While there exist other hybrid routing protocols, ZRP stands out as a notable choice for achieving a balance between reactive and proactive approaches.

In all the mentioned types of ad hoc routing protocols, the concept of the hello interval plays a crucial role. It is utilized to ensure continuous link connectivity between nodes and to keep them informed about any changes in the network. The hello interval serves various purposes, such as maintaining neighbor relationships, detecting link failures, updating routing tables, and facilitating route discovery [19]. To delve deeper into the specifics of the hello interval and to shed more light on the workings of the hello messaging scheme, a detailed explanation of the hello interval is given below to understand how this integral component of routing protocols ensures effective communication and dynamic link management within the network.

2.2 Hello Messaging in Routing Protocols

Hello message and hello interval are two key concepts in routing protocols that are used to maintain connectivity and exchange information between network devices. The hello message is a type of network message that is sent periodically by a network device to inform other devices of its presence and status.

The hello interval refers to the time interval between successive hello messages sent by a network device. A shorter hello interval means that the device will send hello messages more frequently, while a longer hello interval means that the device will send hello messages less frequently. The choice of hello interval depends on various factors such as the size and complexity of the network, the type of protocol being used, and the level of redundancy and fault tolerance required.

By default, the hello interval is usually set to a relatively short time period, such as a few seconds. This ensures that neighboring nodes are kept up-to-date with each other's status and that any changes in the network topology are quickly detected. The hello message scheme works by having each network device send hello messages at regular intervals. When a device receives a hello message from another device X, it creates an entry for X in its neighbor table if it does not have one, else it updates the entry for X. If a device does not receive a hello message from another device within a certain time period, it may assume that the other device has failed or become disconnected and removes the entry for X from its neighbor table.

A node considers all nodes in its neighbor table as its active neighbors and, thus, a link between them. The hello message scheme helps to ensure that network devices remain connected and that problems can be detected and resolved in a timely manner. It is typically designed to be resilient to node failures and topology changes, allowing the network to adapt dynamically to changes in connectivity.

2.3 Working of Traditional Hello Messaging Scheme

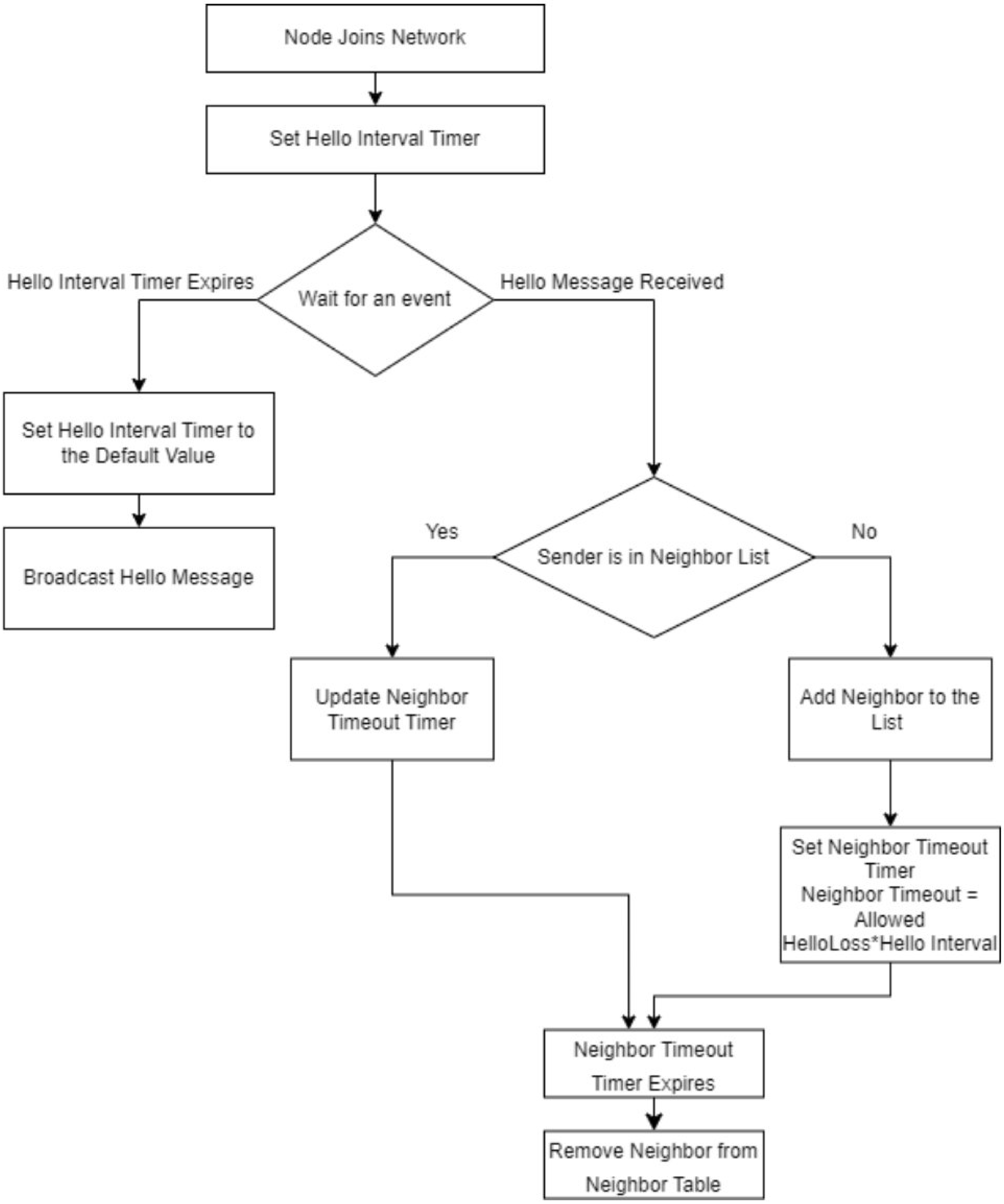


Figure 2.2: Flow Chart for Traditional Hello Messaging Scheme

This flow diagram represents the main steps involved in the hello messaging scheme. The process starts when a node joins the ad-hoc network and sets the hello interval timer. The node then waits for either of the two events i.e. receiving a hello message or expiration of the hello interval timer. If the hello interval timer expires, the node sets the hello interval timer according to the default value and then broadcasts a hello message to all its neighbors. The message contains essential information, including the sender's address, sequence number, and other fields.

The node also listens for incoming hello messages from its neighbors. A node receiving a hello message from another node checks if the sender is in its neighbor list. If the sender is already in the neighbor list, it just updates the neighbor timeout timer else, if the sender is not in the neighbor list, it updates its neighbor table with the sender's information, including address, sequence number, and other relevant parameters. The node then checks the neighbor timeout for each neighbor in its neighbor table. If a Hello message is not received from a particular neighbor within the specified timeout period, it assumes that the neighbor is no longer available and removes the corresponding entry from the neighbor's table. After updating the neighbor table, the node updates its routing table based on the received hello messages. This step ensures that the routing information remains up-to-date and reflects the current network topology.

This process repeats periodically, allowing the node to maintain neighbor information, monitor the network topology, and adapt to any changes that occur.

2.4 Problem Statement

The hello messaging scheme in routing protocols serves crucial functions such as neighbor discovery, topology monitoring, and route maintenance. However, it involves a tradeoff in terms of the hello interval. A shorter interval enables rapid detection of new neighbors or link breaks but at the expense of increased overhead and energy consumption. Conversely, a longer interval reduces overhead and energy consumption but compromises the ability to promptly discover new neighbors and detect link breaks.

To address this tradeoff, an adaptive hello interval becomes imperative. By dynamically adjusting the hello interval based on network conditions several advantages can be achieved. These include optimized resource utilization, reduced control traffic, energy conservation, and efficiency of the ad hoc routing protocol. Incorporating adaptivity into the hello interval enhances overall performance, striking a balance between quick network change detection and minimized overhead.

2.5 Thesis Contribution

In order to address the tradeoff between energy consumption and quick link detection in the fixed hello interval scheme, as discussed in section 2.4, we propose an adaptive hello interval scheme. According to this scheme, the value of the hello interval is decided based on an hello interval equation which is derived based on three important network parameters, namely, the Network density of the simulation area, the Transmission range of UAVs and the Relative speed of UAVs with respect to a reference UAV.

We leverage the state-of-the-art work done by [19] by adding a feature that sets hello interval by considers not only just speed of individual UAV but also by considering the relative speed of reference UAVs with respect to UAVs in their transmission range which helps to understand the condition in the neighborhood of a UAV. This feature helps to make a more accurate prediction of the hello interval value because the relative speed consideration helps to get a more precise idea of the overall network conditions surrounding a UAV. Due to the ability of the proposed approach to make hello interval value more adaptive and accurate based on the network conditions, this approach helps to provide better performance in terms of considered performance metrics such as throughput, overhead efficiency, packet loss ratio and energy consumption in the network.

2.6 Thesis Outline

The organization of the further content is as follows:

- Chapter 3 explains the various approaches in the literature that are used for making hello interval adaptive.
- Chapter 4 describes AODV and OLSR routing protocols and compares them in order to understand which of them is more suitable for FANET scenarios.
- Chapter 5 explains the proposed adaptive hello interval scheme and compared the values of hello-interval obtained using a referred and proposed approach.
- Chapter 6 is divided into two parts; the first section shows the results of comparing AODV and OLSR routing protocols, and the second section shows the results of comparing different hello messaging schemes.
- Chapter 7 discuss the conclusion and future work for our work.

CHAPTER 3

Literature Survey

This section provides a literature review of existing approaches used to make the hello interval adaptive. This review explores various strategies and techniques employed to dynamically adjust the hello interval based on network conditions, such as link change rate, frequency of message transmission, speed of node, network density etc. By examining these approaches, we aim to gain insights into effective implementation strategies that ensure efficient communication in ad hoc networks while mitigating the tradeoff associated with the hello interval.

3.1 Different Approaches Used for Making Hello Interval Adaptive

3.1.1 Based on Link Change Rate

Hernandez-Cons *et al.* [15] proposed an approach for making hello interval adaptive. This approach is based on calculating the link change rate by considering the number of links newly established and the number of links terminated in a specified time interval. This idea is based on the understanding that if a particular node does not face many link changes, then it implies that the node's neighborhood remains stable. Whereas if there are high link changes for a particular node, then it implies that the node's neighborhood is dynamically changing. Thus as the link change rate increases, the value of the hello interval is reduced accordingly and vice versa.

Even though this approach effectively minimizes the overhead, the work does not consider Flying Ad-hoc network scenarios that are characterized by rapid topological changes as unmanned ariel vehicles(UAVs) move away from each other, which leads to longer hello intervals and can cause loss of link detectability. The work only focuses on the MANET scenarios that have stable network topologies with only a few link changes.

Huang *et al.* [16] suggested a method based on the pace at which linkages between network nodes change. A link indicates a direct line of communication with a neighbor, and the rate of change of links quantifies the evolution of a node's network of links. To determine the rate of link change, the authors use an analytical model created by Samar and Wicker [28]. The model takes into account one node's speed while assuming that all other node speeds are random. However, a number of network factors, including network density, maximum and minimum node speeds, and node motion direction, must be understood in order to use the model in real-world scenarios.

While some of these factors may be measured using tools like GPS, others are more difficult to measure precisely. In Huang's technique, the refresh timer of a protocol is adjusted using a multiplicative increase additive reduction controller, offering a dynamic method to change the timer. The link change rate formula's dependence on network measurements to determine its parameters, however, raises questions regarding how well it can be applied in practical situations.

3.1.2 Based on the Number of Neighbor Nodes

Ernst *et al.* [9] proposed Adaptive HELLO (AH) protocol designed for dynamic scenarios with fast- and slow-moving nodes. It utilizes a fine-grained adaptation scheme and two operation modes: Network Search Mode (NSM) and Normal Operation Mode (NOM). In NSM, nodes with fewer neighbors have a fixed short HELLO interval for faster neighbor discovery. Once a node has enough neighbors, it switches to NOM to maintain minimal protocol overhead. AH dynamically adjusts HELLO interval based on link changes, using multiplicative and additive factors.

It also adjusts neighbor hold time-based on the current network state. To coordinate HELLO intervals, AH listens for incoming HELLO messages and adjusts its own interval accordingly. This dependence on incoming messages may introduce delays in the adjustment process, especially in scenarios with limited or intermittent connectivity. Additionally, the reliance on multiplicative and additive factors for interval adjustments may lead to suboptimal performance in highly dynamic environments with frequent link changes.

3.1.3 Based on Frequency of Message Transmission in Network

Han *et al.* [14] proposed an adaptive hello interval scheme which can reduce the unnecessary hello messages in the network, thereby also reducing the energy utilization in the traditional routing protocols while preserving the ability of link detection. It considers the cases of message transmission, reception, and forwarding as events and then checks the time interval between the consecutive events of a type. Based on this information, it decides the value of the hello interval. If no event is recorded in a particular time interval for a specific node, then it increases the hello interval for that node, as this node does not need to maintain link status.

Thus the hello interval increases or decreases based on the occurrence of events in a specific time interval. However, in Flying Ad-hoc networks, some UAVs may not actively engage in communication but still play a crucial role in maintaining communication links; in that case, the above approach will fail to consider such UAVs for setting the hello interval according to this proposed scheme.

3.1.4 Based on Speed of Node and Transmission Range

Park *et al.* [23] performed the analysis to study the effect of the speed of the node and the transmission range of the node on the hello interval. Based on this study, they proposed a scheme to determine the hello interval based on these two factors. However, the factors such as the allowed simulation area of active unmanned ariel vehicles, the number of nodes in the scenario, and the network density are also very important factors, according to the study by Hernandez-Cons *et al.* [15]

Thus due to the constantly changing environment and varying requirements of the FANET network, this approach is not perfectly suitable for the FANET scenarios. As a result, the challenge to come up with a reliable and effective method for figuring out an adaptive hello interval for conventional routing protocols in FANETs that operate effectively and efficiently while maintaining network throughput still persists.

3.1.5 By Restricting Hello Messaging for Certain nodes

Sharma *et al.* [29] proposed an adaptive hello interval approach in which hello messages are transmitted only by nodes that are part of an active route or have formed a routing table. This means that only nodes actively participating in the routing process, either as the source, intermediate, or destination nodes, send hello messages to their neighbors. It focuses on transmitting hello messages where they are most needed, contributing to the optimization of network resources and the scalability of the routing protocol. However, there is one very important drawback of this approach, nodes that are not actively participating in routing, such as newly joining nodes or nodes that have not yet formed a routing table, do not transmit hello messages. This can lead to a delay in discovering and establishing connections with these nodes.

If the network is highly dynamic, with frequent node mobility or link changes, the delay in neighbor discovery caused by the limited transmission of hello messages may impact the efficiency of routing protocols. It may take longer for nodes to identify and establish routes to new or changed neighbors, potentially leading to slower convergence and suboptimal routing decisions.

3.1.6 Based on Network Density, Speed and Transmission Range of nodes

Giruka and Singhal *et al.* [10] presented an alternative approach for MANETs, where nodes transmit hello messages at predefined intervals based on the distance travelled. In their method, nodes with higher speeds are assigned shorter hello intervals, while nodes with lower speeds have longer intervals. Although their approach showed promising results, they did not provide a clear methodology for determining the specific distance. This lack of definition poses a challenge in diverse FANET scenarios, as using the same specific distance can lead to significant performance degradation.

Mahmud *et al.* [19] The proposed EE-Hello scheme aims to minimize energy consumption in traditional FANET routing protocols by reducing unnecessary hello messages. It utilizes the continuous measurement of UAV speed and mission-related parameters such as the transmission range, airspace, number of UAVs, speed ranges, etc to dynamically adjust the hello interval. The scheme also incorporates an exponential weighted moving average to estimate the average speed of UAVs. The value of the hello interval is proportional to the UAV's transmission range and inversely proportional to network density. To ensure link detectability and energy efficiency, the hello interval is constrained within specified limits.

The scheme enables effective link detection while minimizing energy consumption in FANET scenarios, particularly suitable for surveillance, search, and rescue, and reconnaissance missions. One drawback of the above paper is that it solely relies on the speed of individual nodes to determine the hello interval, neglecting the relative speed of neighboring nodes. However, considering the relative speed is important as it allows for more accurate hello interval configurations. For instance, if a node is moving at a high speed, but the relative speed with its neighboring nodes is zero (indicating no relative movement), setting a small hello interval would be inappropriate. By considering relative speed, the hello interval can be appropriately adjusted, ensuring more precise and efficient communication in dynamic network scenarios.

Zhenge *et al.* [32] proposes an adaptive hello mechanism (AHM) in order to address the issue of temporary communication blindness (TCB). In order to save up on the channel resources spent by the nodes that are not active, the authors propose to split the set of nodes into working nodes and idle nodes. Author introduces a concept of departure probability P, which is the probability that node j moves out of the transmission range of node i after Δt. The simulation results show that the proposed approach provides good adaptability to dynamic network topology. AHM mitigates Temporary Communication Blindness (TCB) by addressing node mobility and network connectivity. However, it adds complexity in categorizing nodes, calculating varying Hello periods, and managing state changes.

3.2 AI-based approaches for Adaptive Hello Interval

Ayub *et al.* [3] presents AI-Hello, an intelligent hello dissemination model for FANET routing protocols using Reinforcement Learning. AI-Hello intelligently reduces unnecessary hello messages while maintaining network performance. It incorporates adaptive learning rate and reward factor parameters to adapt to dynamic network conditions. The model is implemented in AODV and OLSR routing protocols, demonstrating superior performance in terms of throughput, packet delivery ratio, end-to-end delay, routing overhead, and energy consumption. By employing appropriate task representation, initial Q-values, and a dense reward structure, AI-Hello effectively learns and improves network performance in dynamic FANET scenarios.

Qiu *et al.* [25] propose an adaptive link maintenance method based on deep reinforcement learning (DRL_MLsA), which can dynamically adjust the time interval of broadcasting hello packets. The reinforcement learning model contained five basic elements namely finite state set S, finite action set A, P is a state transition matrix, reward function R and a discount factor γ. Each time, based on the action chosen by the model, the value of the hello interval is increased or decreased which in turn leads to the next state of the agent and generates some reward. In this way, the model learns strategies for adapting to the changing environment based on the reward function. The superiority of the improved solution was analyzed through testing using performance indicators, such as network throughput, packet loss, and link maintenance overhead. Although the described research paper’s deep reinforcement learning (DRL) methodology provides an advanced technique for adaptive link maintenance in FANET, its reliance on Q-learning adds complexity to the implementation and raises the possibility of convergence issues.

CHAPTER 4

Comparision of AODV and OLSR Ad-hoc Routing Protocols in FANET Scenarios

The selection of AODV and OLSR as the routing protocols for this study is supported by multiple research papers that have thoroughly compared the performance of various routing protocols in different scenarios. Rahma *et al.* [27] analyzed the different MANET routing protocols and concluded that, AODV outperforms DSR in large-scale networks and at higher speeds. This is because DSR exhibits significant delays at higher speeds due to stale routing information. Another conclusion that is drawn from their work is that DSDV drops more packets than OLSR. This can be due to the differences In the route update mechanism where DSDV updates routes on demand which may lead to higher packet drops due to the unavailability of routes.

Another study by AlKhatieb *et al.* [2] carries out a performance evaluation of FANET protocols. The results show that among proactive protocols, OLSR, and reactive protocol, AODV experiences the utmost stable performance with all the mobility models compared to the other routing protocols.

Gupta *et al.* [13] performs an evaluation of reactive routing protocols from a UAVs perspective, demonstrating that AODV performs better than DSR, both in terms of throughput and packet delivery ratio, particularly as the number of nodes in the network increases. This advantage of AODV is attributed to its ability to handle local link repairs, which DSR lacks. Additionally, when comparing DSDV and OLSR, OLSR proves to have lower control overhead. DSDV continuously updates each entry of its table after every packet transmission, even when unnecessary, resulting in increased network load. OLSR, however, avoids such overhead, which enhances its overall performance.

Numerous other research papers [30], [20], [21] have also contributed valuable insights, all pointing towards the superiority of AODV and OLSR under various network parameters. These findings collectively reinforce the confident selection of AODV and OLSR as the preferred routing protocols for our investigation.

4.1 Optimized Link State Routing Protocol (OLSR)

The Optimized Link State Routing (OLSR) protocol is a proactive routing protocol. It aims to establish and maintain routes in a dynamic and decentralized network environment. It is also known as table-driven routing protocol due to the fact that the OLSR operates by periodically exchanging topology information between neighboring nodes to construct and update routing tables. This proactive approach ensures that routes are readily available when needed. The working of the OLSR protocol has the following phases [8].

4.1.1 Neighbor detection

Periodically, each node in the network broadcasts a hello message to all of its neighboring nodes to discover the neighbors. The hello message contains information such as the node's identity, its willingness to participate in routing, and its link status. Each node then creates a local link set containing information about the connectivity between the nodes and its neighbor. This information is helpful in building and maintaining accurate routing tables.

4.1.2 MPR Selection

OLSR uses a technique called Multipoint Relaying (MPR) to minimize control overhead. Nodes select a subset of their neighbors as MPR nodes based on their willingness to route packets and coverage of the network. MPR nodes are responsible for relaying topology information on behalf of their non-MPR neighbors, reducing redundancy in the network. Thus Instead of flooding the control messages to all neighbors, nodes leverage the MPR nodes to relay the information. This optimization significantly reduces the number of control messages and improves the scalability and efficiency of the routing protocol.

4.1.3 Topology Control Message Diffusion

Once the MPR set is established, each node in the MPR set generates a TC (Topology Control) message. After generating the TC message, each MPR node floods it throughout the network. Every node that receives a TC message performs duplicate message detection to avoid unnecessary processing and redundant flooding.

Nodes that receive TC messages then update their topology information based on the information contained in the message. They update their knowledge of MPR nodes, their addresses, and the link metrics associated with the MPR selectors. After processing the TC message, nodes may also update their routing tables based on the updated topology information. To maintain an up-to-date view of the network's topology, the TC message diffusion process is periodically repeated. This step ensures that the network routing information remains current and accurate.

4.1.4 Route Calculation

To calculate the routes, OLSR employs Dijkstra's algorithm, a well-known algorithm used for finding the shortest path in a graph. The route calculation starts with a particular source node. This could be the node itself if it wants to determine routes to other nodes or a node that wishes to forward packets on behalf of other nodes. During the algorithm's execution, each node keeps track of the shortest distance it has discovered to reach a particular node. As the algorithm explores the graph, if it finds a shorter path to a node, it updates the distance accordingly. Once the algorithm completes, each node constructs its routing table based on the shortest paths calculated. The routing table contains entries that specify the destination node and the next-hop node to forward packets toward that destination.

4.1.5 Route Maintenance

OLSR periodically refreshes the routing tables to adapt to network changes. Nodes exchange information about their links and update their routing tables accordingly. If a link or a node fails, the affected nodes reroute traffic to alternative paths based on the updated routing tables.

Thus OLSR is a protocol that operates proactively by exchanging topology information among the nodes to establish and maintain routes and is also suitable for FANET scenarios due to its proactive nature and ability to handle mobility, link quality, and network size.

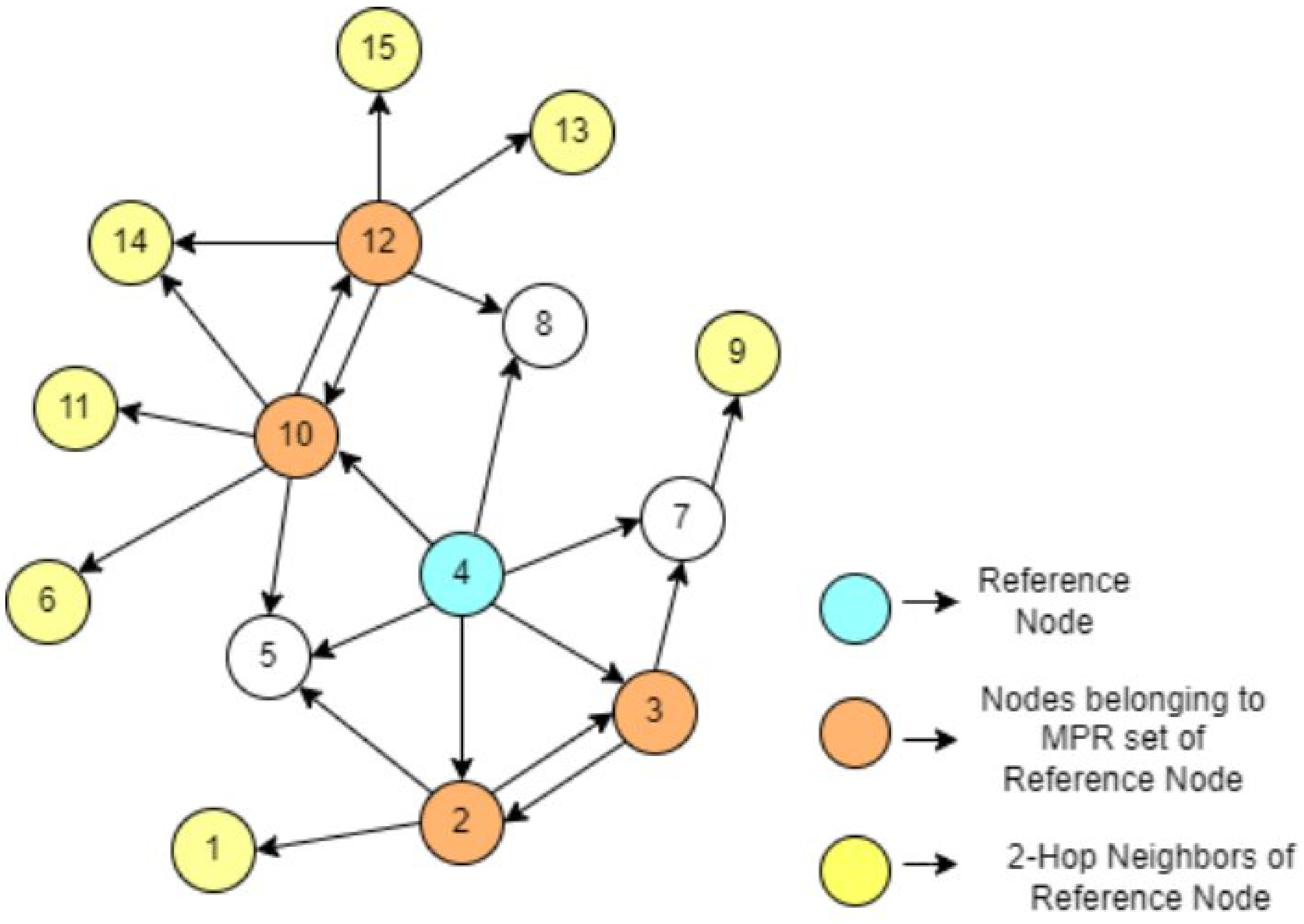


Figure 4.1: Multipoint Relay Selection in OLSR

4.2 Ad-hoc on-Demand Distance Vector (AODV)

Ad-hoc on-Demand Distance Vector (AODV) AODV is a reactive routing protocol that can withstand a variety of network behaviours, such as node mobility, link failures, and packet losses. In AODV source node requests a route only when it wants to send data to the desired destination node; hence the node does not need to maintain the path to each node in the network all the time, unlike in OLSR. There are four kinds of packets through which nodes communicate with other nodes. The nodes maintain the information of their neighbors through the HELLO packet. AODV protocol works in two phases: the route discovery phase and the route maintenance phase [24].

4.2.1 Route discovery phase

Route request (RREQ) and route reply (RREP) packets are used for communication in this phase. The source node requests a route to a destination by broadcasting the RREQ packet to all its neighbors. Each neighbor checks if the destination node in the packet matches its node identity. If it does not match, it, in turn, forwards the RREQ packet t its neighbors. It also stores the reverse route to the source, which will be useful while sending the response. This process continues until the destination node receives the RREQ packet. Then the destination node responds with an RREP packet. This packet unicasts along the reverse route of the intermediate node until it reaches the source node. In this way, a bidirectional route is established between the source and destination node. Due to this reactive property of AODV protocol to establish route dynamically, the source node gets updated routing information even if the topology of the network changes at any point of time [31].

4.2.2 Route maintenance phase

The route maintenance phase in AODV utilizes a Route Error (RERR) message. If a node or a link becomes damaged or fails, neighboring nodes are notified about the link failure through the RERR message. This mechanism helps to maintain accurate routing information by removing broken or invalid routes from the network. When a node detects a link failure or receives a RERR message, it updates its routing table accordingly and propagates the information to its neighboring nodes. This process continues until the network is informed about the broken route, allowing affected nodes to find alternative paths and avoid routing packets through the failed link. By incorporating the route maintenance phase, AODV ensures that the routing tables remain up-to-date and that packets are routed efficiently, even in the presence of changing network conditions or link failures.

Overall, the AODV protocol's reactive approach and two-phased operation allow for dynamic route establishment and efficient handling of network changes. The protocol's route discovery phase enables the source node to request routes on-demand, while the route maintenance phase ensures that broken routes are promptly removed from the network, maintaining the accuracy and efficiency of routing operations [17].

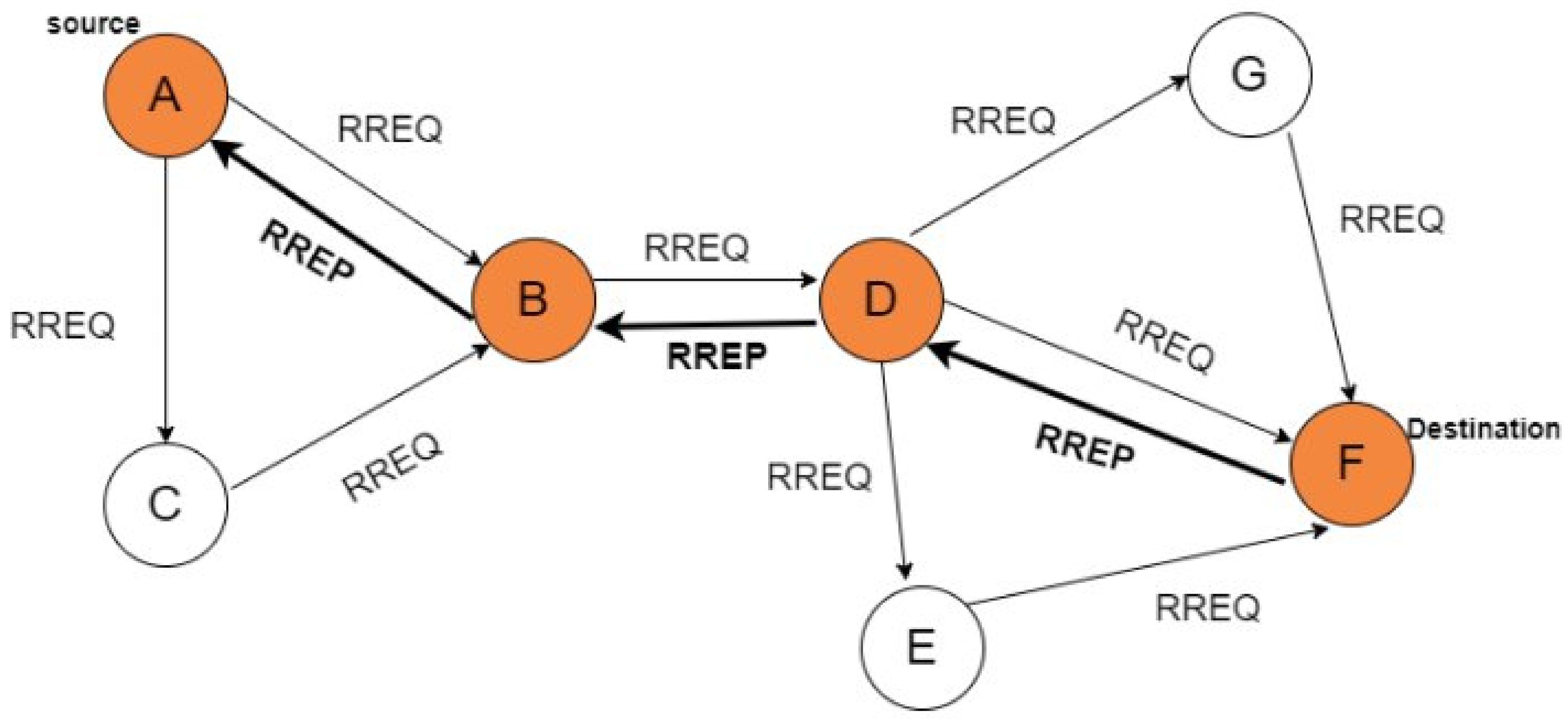


Figure 4.2: Route Request and Route Reply Packet Flow in AODV

4.3 Major Differences Between OLSR and AODV Adhoc Routing Protocols from FANET Perspective

In this section, we try to understand the factors in which AODV and OLSR routing protocols differ from each other from the FANETs perspective.

4.3.1 Route Discovery Mechanism

AODV adopts an on-demand route discovery mechanism, which means that routes are established only when needed. In contrast, OLSR relies on proactive route discovery, where routes are continuously maintained even if they are not immediately required. Thus in FANET scenarios with dynamic topologies and limited bandwidth, AODV's on-demand approach can reduce control overhead and conserves resources more compared to OLSR.

4.3.2 Resource Utilization

AODV typically exhibits higher throughput than OLSR in FANET scenarios. The on-demand nature of AODV allows for efficient utilization of available network resources. When a route is established, AODV ensures that packets are forwarded directly along the established path, optimizing the packet delivery process. In contrast, OLSR's proactive approach may lead to suboptimal routing decisions in highly dynamic FANETs, resulting in longer routes or increased packet loss. These inefficiencies can reduce the overall throughput of the network compared to AODV.

4.3.3 Network Latency/Delay

The proactive nature of OLSR allows it to continuously update routing information, enabling pre-established routes that minimize the time required for route discovery and setup. In contrast, AODV's reactive approach necessitates route discovery when needed, introducing an additional delay. Consequently, OLSR's proactive route maintenance reduces end-to-end delay by optimizing route selection and reducing the time taken for route establishment.

4.3.4 Link Stability Awareness

OLSR takes into account link quality metrics such as signal strength, packet loss, and delay when determining routes. It uses these metrics to select the most stable and reliable paths for data transmission. In a FANET, where link quality is highly variable due to factors like node mobility and interference, OLSR's ability to consider link stability enhances packet delivery rates and overall network performance. AODV, being reactive, does not have built-in mechanisms to consider link stability, which can result in suboptimal route choices and increased packet loss in dynamic FANET scenarios.

Apart from the factors considered above, many other factors such as scalability, Network congestion handling, Impact of mobility, Multi-Path Routing, etc can be studied for understanding the difference between AODV and OLSR in more detail.

We also performed ns-3 simulations to analyze the performance of both these protocols in a FANET scenario.

The details of the simulator used for the experiments along with the results and justification are explained further in Chapter 6.

CHAPTER 5

Proposed Adaptive Hello Interval Approach

As mentioned in the literature survey Mahmud *et al.* [19] proposed an EE-Hello scheme for adaptive hello interval. This approach uses 3 network parameters namely the speed of the UAV, the transmission range of the UAV, and network density as the major factors to decide the value of the hello interval of a particular node at a particular timestamp. This equation proves efficient for setting the value of hello-interval based on the changes in the network and also gives a better performance in terms of throughput, Packet delivery ratio, Overhead, and energy consumption in the network.

5.1 Referred Equation for Adaptive Hello Interval as Proposed by Mahmud *et al.* [19]

$$T_H(n) = \frac{T_x}{VN_D}\beta\frac{v_{max}}{v_{min}} \tag{5.1}$$

Where,
$T_H(n)$ - Value of Hello Interval
T_x - Transmission Range of UAV
V - Speed of UAV
N_D - Network Density
β - Tuning constant
v_{min} - Minimum speed that a UAV can attain
v_{max} - Maximum speed that a UAV can attain

One limitation of the mentioned paper is its exclusive reliance on the speed of individual nodes when determining the hello interval, without considering the relative speed of neighboring nodes. However, taking relative speed into account is crucial for the accurate configuration of the hello interval. For example, if a node is moving quickly but has zero relative speed with its neighboring nodes (meaning they are not moving relative to each other), it would be inappropriate to set a small hello interval. By factoring in the relative speed, the hello interval can be adjusted appropriately, leading to more precise and efficient communication in dynamic network scenarios.

In contrast to the approach proposed by Mahmud *et al.* [19], which solely relies on the speed of individual nodes for determining the hello interval, our proposed approach incorporates the relative speed of neighboring nodes. By considering the relative speed, our approach ensures that the hello interval configuration takes into account the conditions within the node's neighborhood. This enhancement prevents the situation where an inappropriate hello interval value is set based solely on the speed of the node of interest. Consequently, our approach offers a more accurate and effective means of determining the hello interval, leading to improved communication in dynamic network scenarios. The details of the proposed approach are discussed in the following sections.

5.2 Hello Interval Versus Transmission Range

Firstly, we try to understand the impact of the transmission range of UAVs on the value of the hello interval.

The previous study by Park *et al.* [23] shows that the hello interval is directly proportional to the transmission range of the UAV. If the transmission range is large, the links constructed between the node and its neighbors will remain maintained for a longer period of time thus, the node can broadcast a hello message slowly. On the contrary, if the transmission range is small, the link constructed between the node and its neighbor will remain maintained only for a shorter period of time, thus the node has to advertise a hello message with a shorter interval in order to remain updated with the neighbors.

Thus the relational expression of the hello interval and transmission range of UAVs can be defined as follows:

$$T_H(n) \propto T_x \tag{5.2}$$

5.3 Hello Interval Versus Network Density

Next, we try to understand the impact of the network density of UAV on the value of the hello interval.

According to Hernandez-Cons *et al.* [15], the hello interval is inversely proportional to the density of the network. Low network density implies fewer chances of a node experiencing interference or route instability that can disrupt communication. Thus hello interval must be set to a larger value in such cases. On the contrary, if the node density is high the possibility of nodes experiencing interference is high. In such cases, the value of the hello interval must be set to a smaller value so that the link changes due to interference are detected quickly.
Thus the relational expression of the hello interval and network density can be defined as follows:

$$T_H(n) \propto \frac{1}{N_D} \tag{5.3}$$

5.4 Hello Interval Versus Relative Speed of UAV

Third, we try to understand the impact of the node's relative speed on the value of the hello interval.

The previous study by Park *et al.* [23] shows that the hello interval is inversely proportional to the speed of the UAV. But instead of considering only the speed of the UAV, considering the relative speed between UAV and its neighbors is a more appropriate measure of the stability of the UAVs neighborhood. The high relative speed between a UAV and its neighbors indicates less stability in their links due to their quick changes in positions. In such cases to ensure the timely detection of topology changes, the hello interval needs to be reduced and vice versa.
Thus the relational expression of the hello interval and the relative speed of a UAV can be defined as follows:

$$T_H(n) \propto \frac{1}{S_{relative}} \tag{5.4}$$

5.5 Computation of Transmission Range (T_x)

In our simulations, we assume that all the UAVs have omnidirectional antennas and identical transmission ranges. Thus the value of the transmission range remains the same for all UAVs within the simulation.

5.6 Computation of Network Density (N_D)

To determine the density of the network, we calculate the number of UAVs required to connect the mission area in 3D space. We assume stationary UAVs equipped with omnidirectional antennas and calculate the number of UAVs that can be accommodated inside the airspace by dividing the volume of allowed airspace (V_M) *i.e.* a cuboid by the volume of the transmission area of a UAV (V_{T_X}) *i.e.* a sphere. By assuming that all UAVs have the same transmission range, we can calculate the maximum number of UAVs that can be accommodated in the airspace as

$$U_{accom} = \frac{V_M}{V_{T_X}} = \frac{3LWH}{4\pi T_x^3} \tag{5.5}$$

Where L, W, and H are the length, width, and height of the simulation area. However, as the transmission volume is not perfectly spherical in practice, the minimum number of UAVs required is two times the calculated number of accommodated UAVs.

$$U_{req} = 2U_{accom} = \frac{6LWH}{4\pi T_x^3} \tag{5.6}$$

In a dynamic network where UAVs are moving, assuming the mission involves U_M available UAVs, we define the network density (N_D) using the below equation.

$$N_D = \frac{U_M}{U_{req}} = \frac{4\pi T_x^3 U_M}{6LWH} \tag{5.7}$$

5.7 Computation of Relative Speed ($S_{relative}$)

Algorithm 1 shows the steps for computing the relative speed of a UAV with respect to its neighbors. When a UAV receives hello messages from its neighbors, it also receives the value of the neighbor's speed which is appended in the hello message itself. Thus, each UAV is aware of the number of UAVs in its transmis-

```
Algorithm 1: Computation of Relative Speed
input     : ID of reference UAV and List of UAVs in its transmission range
            represented by UAVofInterest and UAVsInRange respectively
1 $S_{relative}$, relative speed of UAVofInterest for $i \in$ UAVsInRange do
2     relativeSpeed = $|V_{UAVofInterest} - V_i|$
3     RelativeSpeedofUAVsInRange.pushback(relativeSpeed)
4 $RS^{max}$ = max (RelativeSpeedofUAVsInRange)
5 $S_{relative}$ = min ($s_{max}$,max($s_{min}$,$RS^{max}$))
6 Step 5 above ensures that the value of $S_{relative}$ is bounded between some
    maximum and minimum values defined by $s_{min}$ and $s_{max}$.
```

sion range and their speed values. Thus the UAV finds its relative speed with respect to each UAV in its transmission range and then chooses the maximum among these relative speeds as RS^{max}.

By selecting the maximum relative speed among these nodes as the final value, the scheme prioritizes the node that has the highest potential to significantly impact the stability of its neighborhood. Because the node, with the highest relative speed, is more likely to introduce instability and therefore serves as a crucial determinant for setting the hello interval, ensuring that it is appropriately adjusted to maintain a stable neighborhood.

Then we perform thresholding to make sure that the value of $S_{relative}$ is bounded between some maximum and minimum values defined by s_{min} and s_{max}.

The purpose of performing thresholding operation is as follows:

As RS^{max} is a relative speed, its value can vary from zero to almost $2s_{max}$. But the value of the hello interval is inversely proportional to $S_{relative}$ as discussed in section 5.4. Thus, setting $S_{relative}$ to zero or $2s_{max}$ will result in the hello interval getting set to either infinity or a very small value which is not an ideal situation. Thus to avoid the above situation, we perform thresholding. The appropriate values of s_{min} and s_{max} are found through experiments.

Thus we get the value of the relative speed of reference UAV i.e $S_{relative}$ which can then be used in the equation for the hello interval.

5.8 Proposed Equation for Adaptive Hello Interval

From Eqs. (5.1), (5.2), and (5.3), we can derive an equation of hello-interval as follows:

$$T_H(n) = \frac{T_x}{S_{relative} N_D} \beta \tag{5.8}$$

Where β is a tuning constant.

For a particular simulation scenario,

1. The value of T_x is fixed as discussed in section 5.5.
2. Value of N_D, which depends on the constants, namely, simulation area, number of UAVs, and the transmission range is also fixed for a particular scenario as discussed in section 5.6.
3. Thus, the value of hello interval varies only based on the value of $S_{relative}$.

Thus the above equation can be used to set the hello interval according to our proposed approach. Here the factor $S_{relative}$ helps to consider the relative speed between the UAVs, which in turn gives more precise insight into the actual condition in the neighborhood of a particular UAV (regarding the stability of the neighborhood) before setting the hello interval for it.

5.9 Comparision of Hello Interval Values Obtained Using the Referred and Proposed Equation for Hello Interval

The plots presented here aim to demonstrate the effectiveness of our proposed equation in setting the hello interval to a value that better suits the conditions in the UAV's neighborhood. In contrast to the referred equation, which only considers the UAV's individual speed and neglects the surrounding UAVs, our approach takes into account the neighborhood situations.

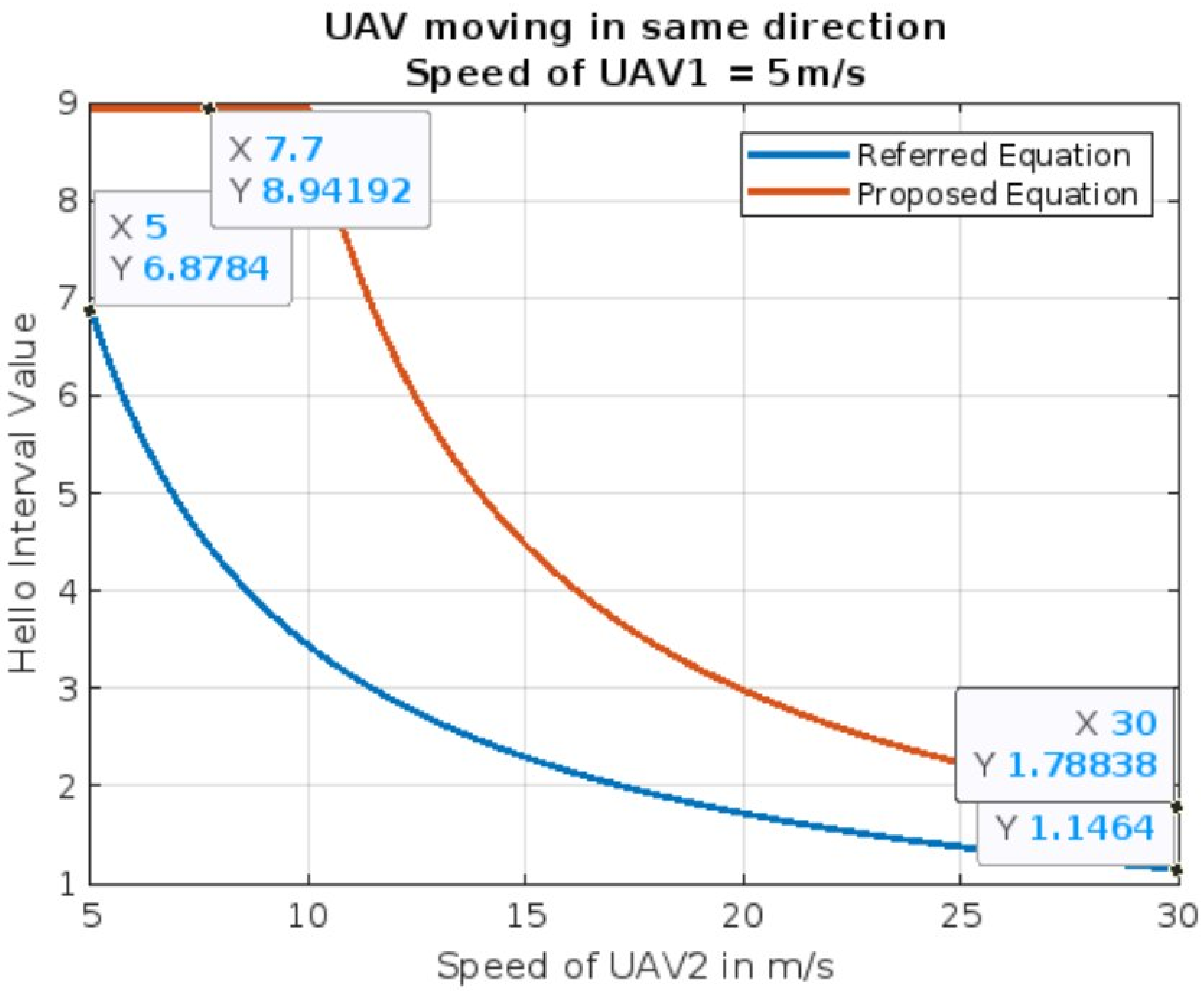
We illustrate two different scenarios: one where two UAVs move in the same direction with specific speeds, and another where two UAVs move in opposite directions. In the plots (Fig 5.1), we observe that when a UAV maintains its individual speed but has a zero relative speed to other UAVs (i.e., the neighborhood remains unchanged), the hello interval can be set to a higher value than that prescribed by the referred equation. This adjustment leads to reduced battery and bandwidth consumption in the network, improving efficiency.

In the plots (Fig 5.2), we demonstrate that when UAVs move in opposite directions, they may quickly move out of each other's transmission range, causing frequent changes in the neighborhood. In such cases, our proposed equation sets the hello interval to a smaller value than the one provided by the referred equation. This adaptation ensures that a UAV must be able to quickly detect the changes in its neighbourhood by transmitting hello messages at a smaller interval.

Overall, our proposed equation helps optimize the hello interval based on the UAV's relative speed to other UAVs in the network, leading to enhanced communication efficiency and resource utilization.

The values of different network parameters considered for the below plots are as follows:

T_x = Transmission Range of UAV = 150 m/s,
U_M = Number of UAV's in the Network = 20 UAVs.
L, W, H = Length, Width and Height of Simulation Area = 600 m, 600 m, 150 m,
N_D = Network Density = $4\pi T_x^3 U_M/6\text{LWH}$ = 2.616
v_{max} = 5 m/s,
v_{min} = 30 m/s
V = Values between v_{min} and v_{max},



(5.1.a) UAV1 moving with a speed of 5m/s

Figure 5.1: Comparision of Hello Interval Values Obtained using Referred and Proposed Equation for a case of 2 UAVs moving in the same directions for varying speeds of UAV2

Figure 5.1.a shows variation in the values of hello interval obtained using the referred and the proposed equation for hello interval. We consider a scenario involving two UAVs, UAV1 and UAV2, moving in the same direction. where UAV1 moves with a constant speed of 5m/s while UAV2's speed varies from 5 m/s to 30 m/s. The purpose of studying this scenario is to compare how two different equations set the hello interval when two UAVs are moving in the same direction at the same speed.

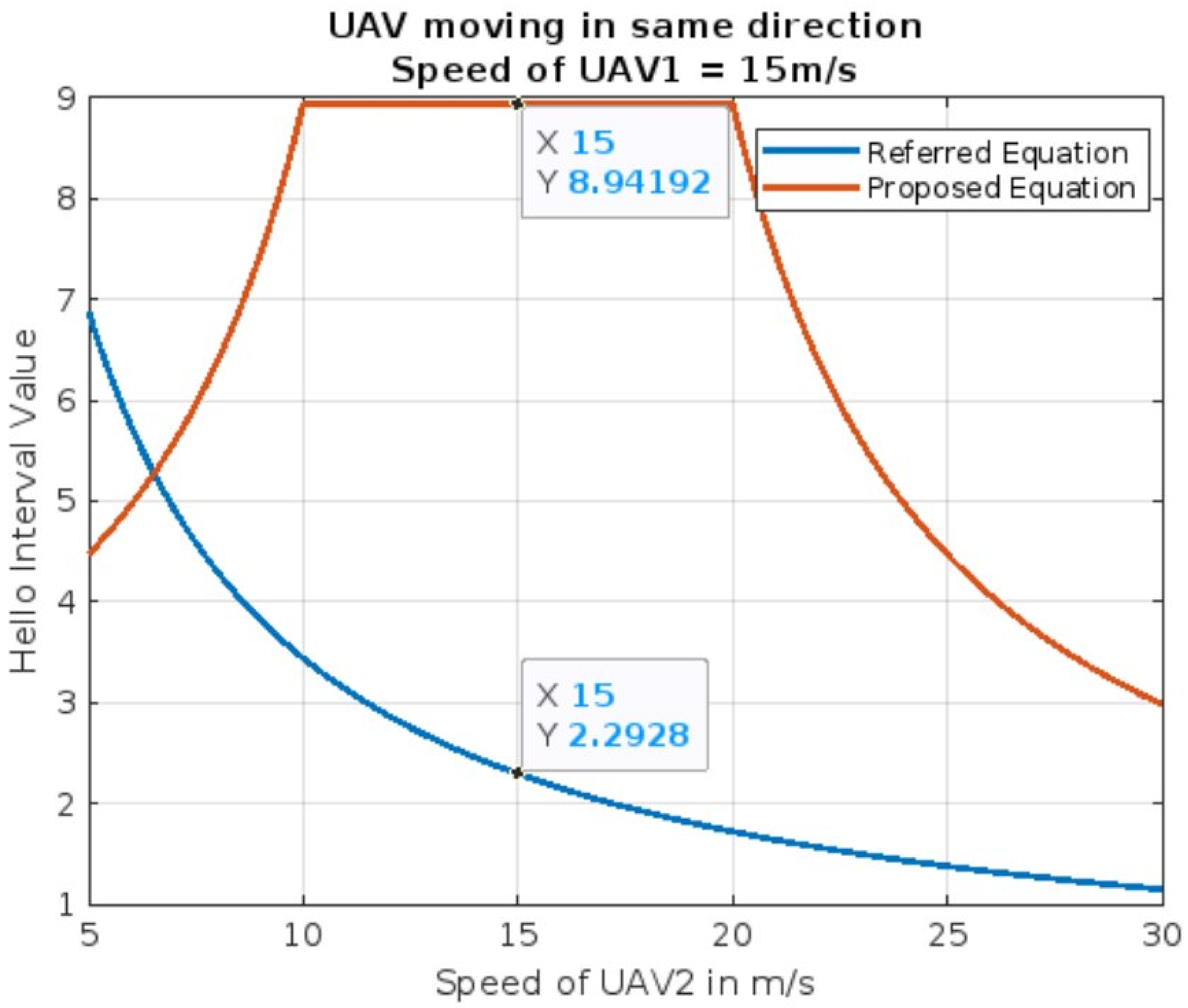
The blue line shows the values of hello interval obtained using the referred equation as mentioned in section 5.1 and the red line shows the values of hello interval obtained using the proposed equation as mentioned in section 5.8.

When UAV2's speed is 5m/s, we see a noticable difference in the hello interval values obtained using the two equations. The referred equation, which considers the absolute speed of UAV2, sets the hello interval to 6.8784 seconds. In contrast, the proposed equation, which takes into account the relative speed of UAV2 with respect to UAV1, sets the hello interval to 8.9419 seconds.

The reason behind this discrepancy lies in the relative motion between UAV1 and UAV2. In the case of the proposed equation, even though UAV2 is moving at 5m/s, its relative speed concerning UAV1 becomes zero, as both UAVs are moving in parallel.

Consequently, the proposed equation allows for a higher hello interval value compared to 6.8704 seconds because UAV2's neighborhood remains unchanged.

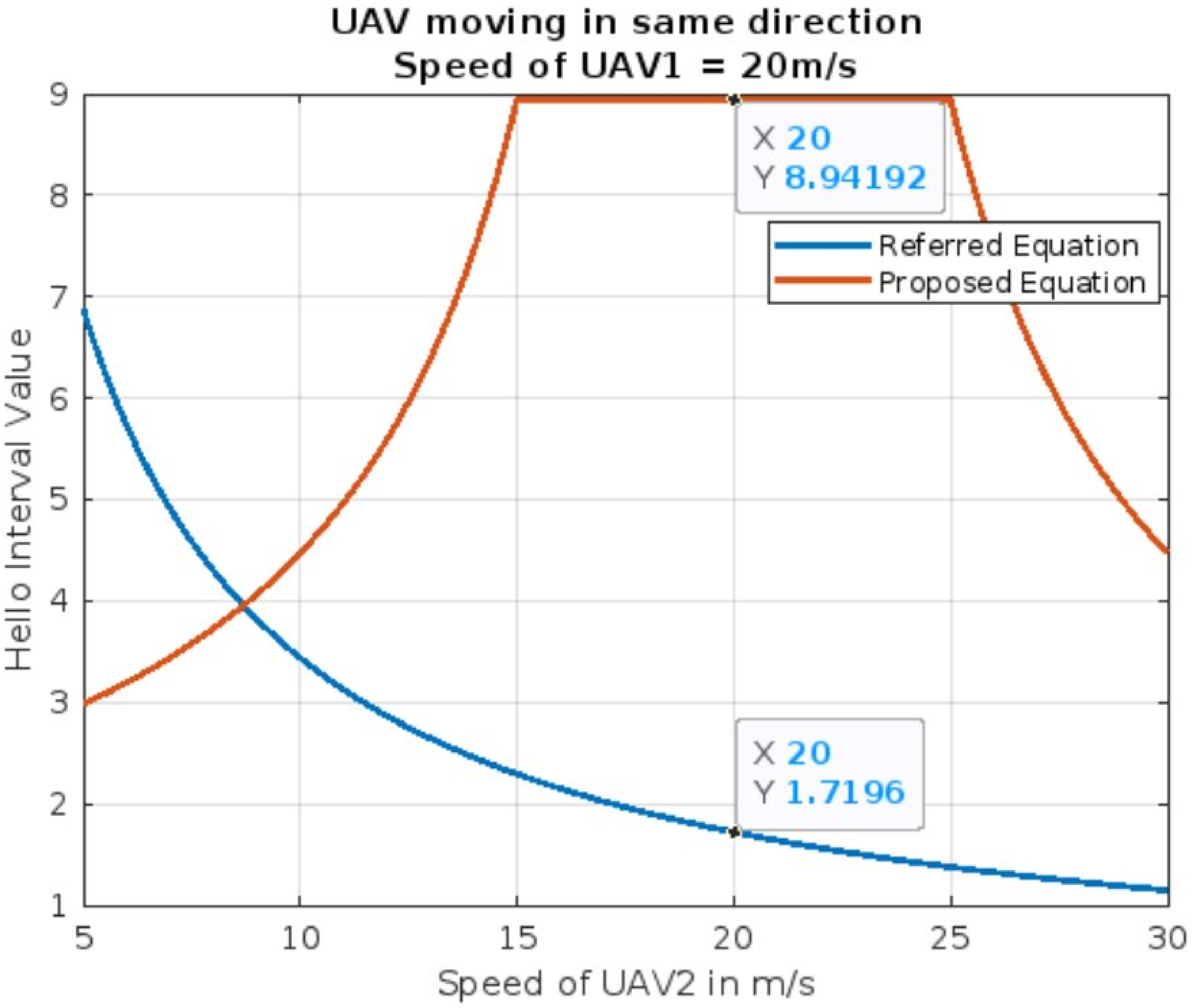
In summary, the proposed equation takes into consideration the relative speed of UAVs, leading to different hello interval values under specific motion scenarios, highlighting its potential advantages over the referred equation in certain operational conditions.



(5.1.b) UAV1 moving with a speed of 15m/s

Figure 5.1 (cont.)

Fig 5.1.b shows a plot similar to Fig 5.1.a and is based on similar considerations. The difference lies in the speed considered for UAV1. Fig 5.1.b considers that UAV1 moves with a constant speed of 15m/s. When UAV2's speed is also 15m/s, we observe that the value of hello interval using a referred equation is 2.2928 seconds whereas using the proposed equation it is 8.9419 seconds. The significant difference in the values can be attributed to how each equation considers the speed of UAV2. The reason being similar to the one explained in the case of Fig 5.1.a., i.e. the referred equation solely looks at the speed of UAV2 at a particular instance and sets the hello interval based on that value. On the other hand, the proposed equation takes into account the relative speed of UAV2 with respect to UAV1. In this case, since both UAV1 and UAV2 have the same speed of 15m/s, their relative speed becomes zero. As a result, the proposed equation sets a hello interval value which is comparatively greater than the one obtained using referred equation.



(5.1.c) UAV1 moving with a speed of 20m/s

Figure 5.1 (cont.)

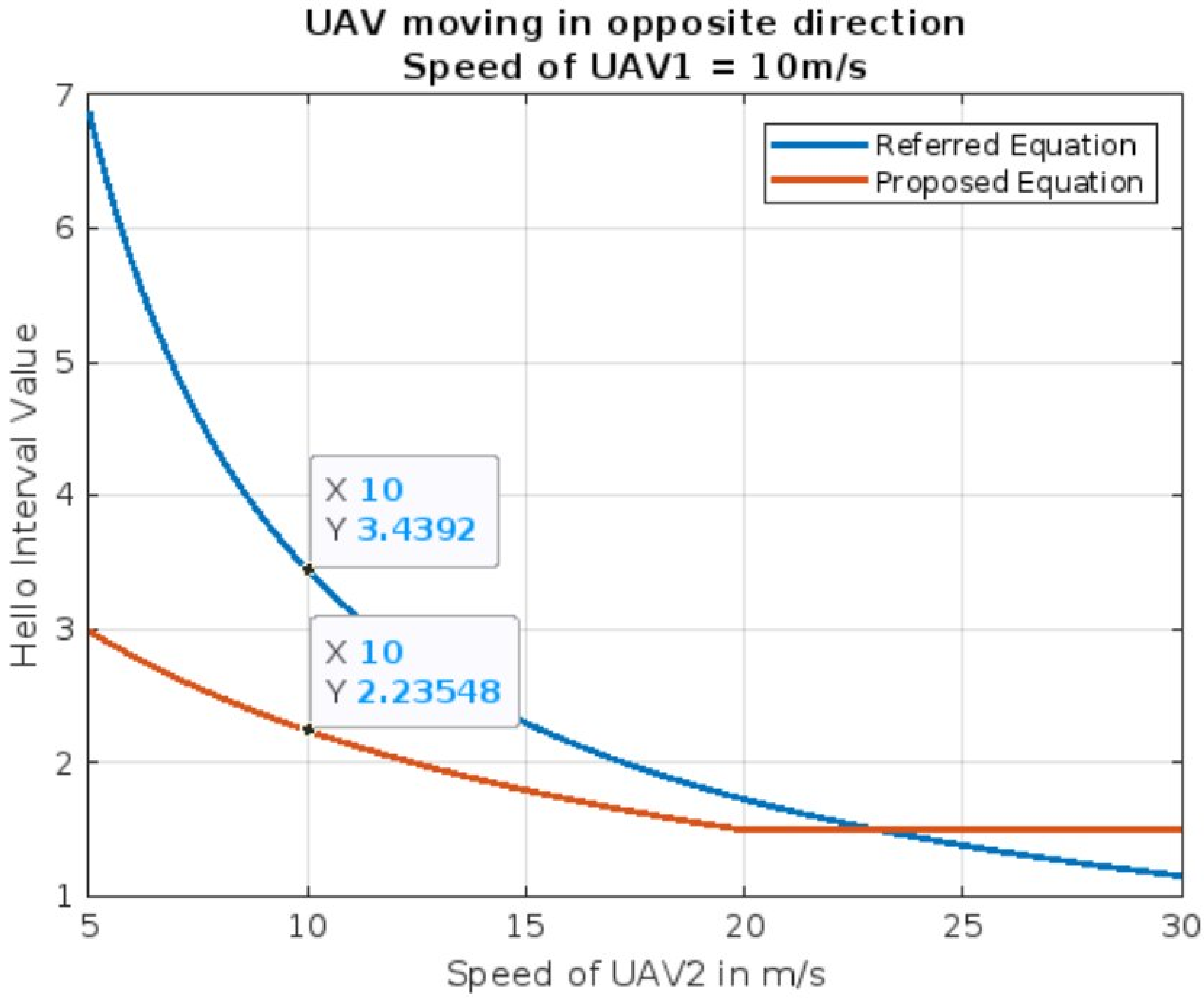
Fig 5.1.c shows a plot similar to Fig 5.1.a and 5.1.b and is based on similar considerations. The difference lies in the speed considered for UAV1. Fig 5.1.c considers that UAV1 moves with a constant speed of 20m/s. When UAV2's speed is also 20m/s, we observe that the value of hello interval using a referred equation is 1.7196 seconds whereas using the proposed equation it is 8.9419 seconds. The significant difference in the values can be attributed to how each equation considers the speed of UAV2. The reason being similar to the one explained in the case of Fig 5.1.a and 5.1.c.

Thus, notably, in each plot, where the speeds of UAV1 and UAV2 match (i.e., zero relative speed), Equation 2 sets the hello interval to a higher value, demonstrating its reliance on the relative speed rather than just the absolute speed of UAV2 which in turn helps in battery and bandwidth consumption in such cases.

In the same way as in Fig 5.1 and its subplots, Fig 5.2.a and Fig 5.2.b also observe variations in the hello interval values obtained using two different equations for a scenario involving two UAVs, UAV1 and UAV2 but in this case the UAVs move in the opposite direction.

since in this case, UAV2 moves in the direction opposite to UAV1, the relative speed of UAV2 is more than its actual speed. In such cases the possibility that UAV1 will move out of the transmission range of UAV2 is higher and hence hello interval must be set to a comparatively smaller value than the one obtained using the referred equation so that the dynamic changes in the network can be captured accurately.

The blue line represents the hello interval values obtained using the referred equation mentioned in section 5.1. On the other hand, the red line represents the hello interval values obtained using the proposed equation from section 5.7.

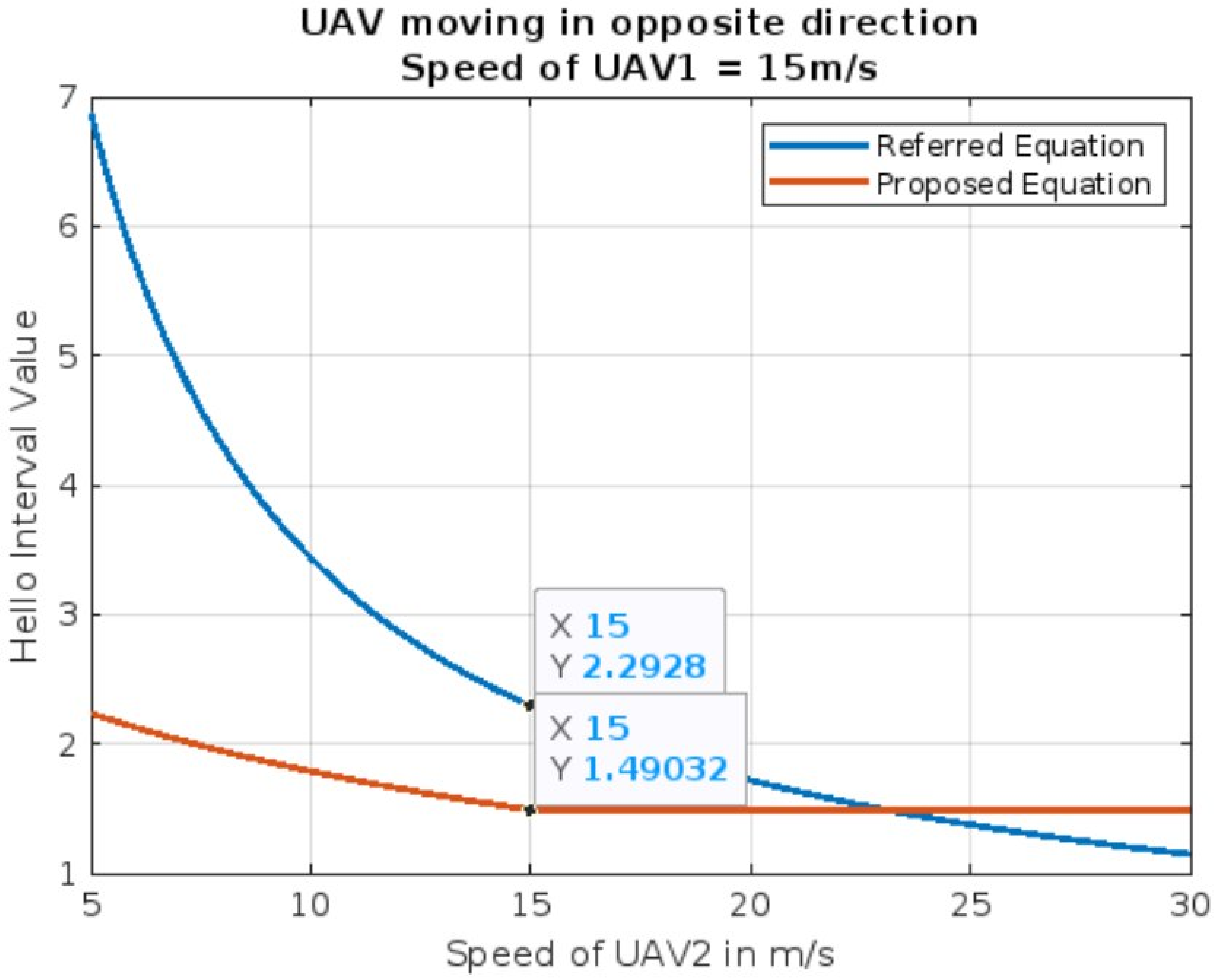


(5.2.a) UAV1 moving with a speed of 10m/s

Figure 5.2: Comparision of Hello Interval Values Obtained using Referred and Proposed Equation for a case of 2 UAVs moving in the opposite directions for varying speeds of UAV2

Fig 5.2.a shows a scenario where UAV1 maintains a constant speed of 10 m/s, while UAV2's speed varies from 5m/s to 30m/s. we can observe that when

UAV2's speed is 10m/s, the relative speed of UAV2 with respect to UAV1 becomes 20m/s, and the probability that both the UAVs will move out of each other's transmission range is higher. Since the referred equation considers only the actual speed of the UAV, it sets the hello interval value to 3.4392 seconds; on the other hand, the proposed equation sets the hello interval value to a smaller value of 2.23548 seconds which helps in comparatively faster transmission of hello messages in order to detect the dynamically changing neighbourhood of the UAV more precisely.



(5.2.b) UAV1 moving with a speed of 15m/s

Figure 5.2 (cont.)

A similar kind of observation can be made from Fig 5.2.b, where UAV1 maintains a constant speed of 15 m/s, and all other considerations are similar to those in Fig 5.2.a. We can observe that when UAV2's speed is 15m/s, the relative speed of UAV2 with respect to UAV1 becomes 30m/s. Since the referred equation considers only the actual speed of the UAV, it will set the hello interval proportional to a speed of 15m/s, i.e. around 2.2928 seconds. However, our proposed equation sets the hello interval based on the relative speed, i.e. 30m/s in this case which leads to a hello interval of 1.49032 seconds which helps in comparatively quicker detection of neighborhood changes for UAV2.

5.10 Working of Proposed Adaptive Hello Messaging Scheme

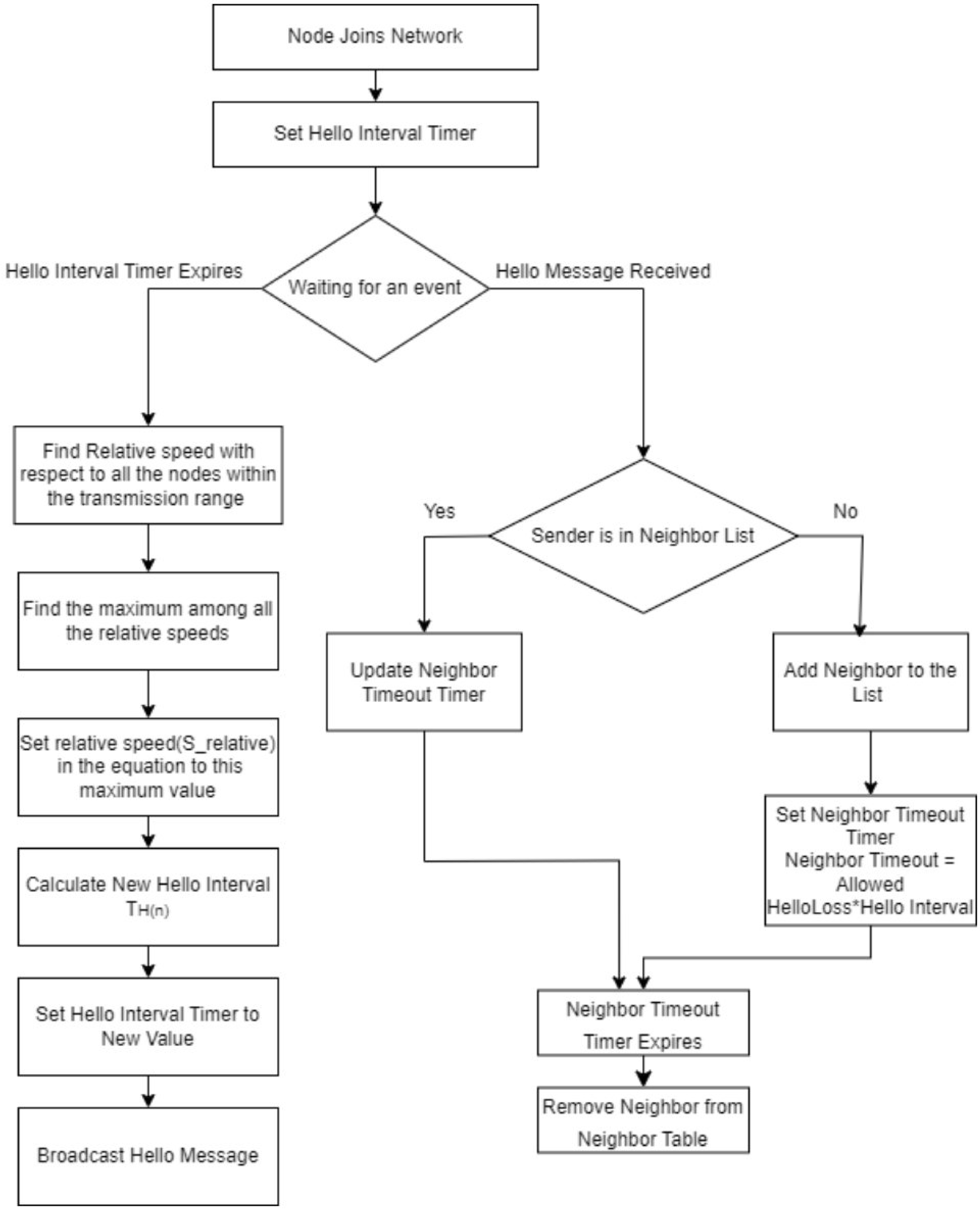


Figure 5.3: Flow Chart for Proposed Adaptive Hello Messaging Scheme

Figure 5.3 represents the flow diagram for the proposed adaptive hello messaging scheme. The overall working of the hello messaging scheme remains the same as explained in section 2.2 but in the proposed approach, while resetting the hello interval timer, instead of setting it to a fixed hello interval value, it calculates the value of the hello interval using the above-mentioned equation. This ensures that the hello interval is set by considering the relative speed of the UAV with respect to its neighbor.

Moreover, the equation also takes into account the value of the transmission range and network density and then sets the hello interval accordingly.

Thus deciding the value of the hello interval based on the discussed network parameters helps to balance the tradeoff between unnecessary energy consumption in the case of a stable network and a need for quick detection of link changes in the case of an unstable network.

CHAPTER 6

Simulation Setup and Results

6.1 About ns-3 Network Simulator

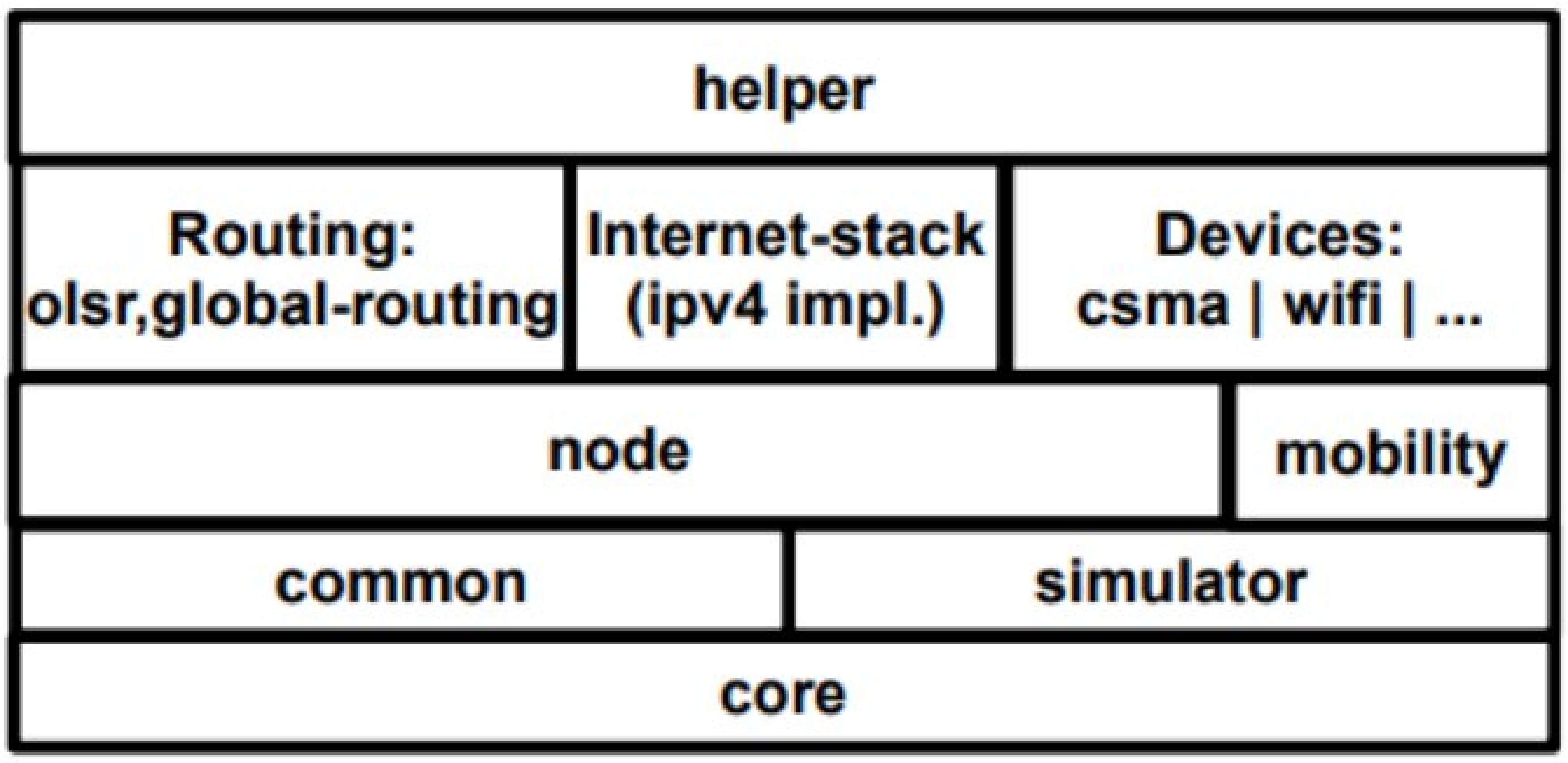


Figure 6.1: Overview of the main ns-3 modules

ns-3 is a discrete-event network simulator targeted primarily for research and educational use. ns-3 is free software, licensed under the GNU GPLv2 license, and is publicly available. The simulator is organized into different modules, each serving a specific purpose.
At the top level, ns-3 includes a "helper" module that simplifies the usage of lower-level classes. This module encapsulates complex functionalities with straightforward classes, resembling the facade design pattern.
The ns-3 simulation “core” module supports research on both IP and non-IP-based networks. However, the large majority of its users focus on wireless/IP simulations which involve models for Wi-Fi, WiMAX, or LTE for layers 1 and 2 and a variety of static or dynamic routing protocols such as OLSR and AODV for IP-based applications.
ns-3 comprises several other modules, including "common" for packet and header manipulation, "simulator" for time manipulation and event scheduling, "node" for

fundamental network simulator features such as node representation, layer-2 interface, and address types, and "mobility" for incorporating mobility models. It also offers routing models like OLSR and global routing, an "internet-stack" module implementing a UDP/TCP/IPv4 stack, and NetDevice implementations for WiFi, CSMA, and PointToPoint links.

One notable characteristic of ns-3 is its tracing architecture. Tracing allows researchers to observe significant events occurring within the simulation and analyze their conditions. In addition to generating text files like most simulators, ns-3 also supports storing events in PCAP files. However, the primary tracing system in ns-3 is callback-based tracing.

It defines trace sources associated with specific object classes, and programmers can register functions or methods to be called when these trace sources generate new events. This enables various uses such as writing event data to files, collecting statistics, and real-time parameter modification for experimentation.

Overall, ns-3 is a highly capable network simulator with a robust architecture and continuous development. Its comprehensive features, including advanced tracing capabilities and modular organization, make it a preferred choice for network researchers and developers [6].

6.2 Comparision of AODV and OLSR Routing Protocols (Simulation 1)

6.2.1 Simulation Setup Details for Simulation 1

The ns-3 simulator was used to compare the performance of AODV and OLSR routing protocols in FANET scenarios. The simulation area was set to 1000 x 1000 units and the Gauss-Markov mobility model was employed for node movements [5]. The transmission range was fixed at 150 m, and the simulation lasted for 200 seconds. Two sets of experiments were performed. In the first set, the speed of the nodes was fixed at 20 m/s, and the number of nodes varied from 10 to 50. The objective was to observe the impact of network size on the throughput and end-to-end delay of both routing protocols. In the second set of experiments, the number of nodes was fixed at 20, and the speed range was varied from 10 to 70 m/s. This experiment aimed to examine how different node speeds affected the throughput and end-to-end delay in FANETs, comparing the performance of AODV and OLSR. The results were analyzed to compare the performance of

Table 6.1: Network Simulation Parameters for Simulation 1

Simulation Parameters	***Value***
Network simulators	*ns-3*
Simulation area	*1000 x 1000 m square*
Number of UAVs	*10,20,30,40,50*
UAVs speed	*[10,20,30,40,50,60,70] m/s*
Transmission range of a UAV	*150 m*
Simulation time	*200 s*
Mobility model	*Random Way Point*
Routing protocols	*AODV,OLSR*
Packet size	*512 bytes*
Data type	*CBR*
Antenna type	*Omnidirectional*
PHY/MAC protocol	*802.11b*
PropagationLossModel	*FriisPropagationLossModel*
Channel Type	*Wireless*
Data rate	*16 Kbps*

AODV and OLSR. Factors such as network size, node speed, throughput, and transmission range were considered. This analysis aimed to determine the suitability of each routing protocol for FANET deployments. The findings of this study provide valuable insights into the performance characteristics of AODV and OLSR in FANETs and contribute to the understanding of the most suitable routing protocol for such scenarios. These results can aid in the design and optimization of FANETs by considering the network parameters, node mobility, and routing protocol's performance requirements.

6.2.2 Performance Metrics for Simulation 1

- **Received Rate**

  It refers to the rate at which data packets are received during each second of the simulation. This can help to compare the effectiveness of AODV and OLSR routing protocols in delivering data packets from source to destination within a specific time frame.

- **Throughput**

  Throughput refers to the amount of data successfully transmitted or received over a network within a given time period. It provides a quantitative measure to evaluate the efficiency, capacity, and network utilization of the protocol. Higher throughput implies a higher rate of successful data delivery, indicating better utilization of available network resources

$$\frac{\sum Rec\_Pkt_{size} * 8}{Transmission\_Interval * 1024} \tag{6.1}$$

- **End-to-End Delay**
  The end-to-end delay in a network refers to the total time taken for a data packet to travel from the source node to the destination node, including all intermediate processing and transmission delays. end-to-end delay helps to evaluate the protocol's ability to minimize packet delivery time, reduce latency, and ensure timely communication.

$$\frac{\sum Delay}{Rec\_Data\_Pkt_{size}} \tag{6.2}$$

6.2.3 Results of Simulation 1

1. **Received Rate**

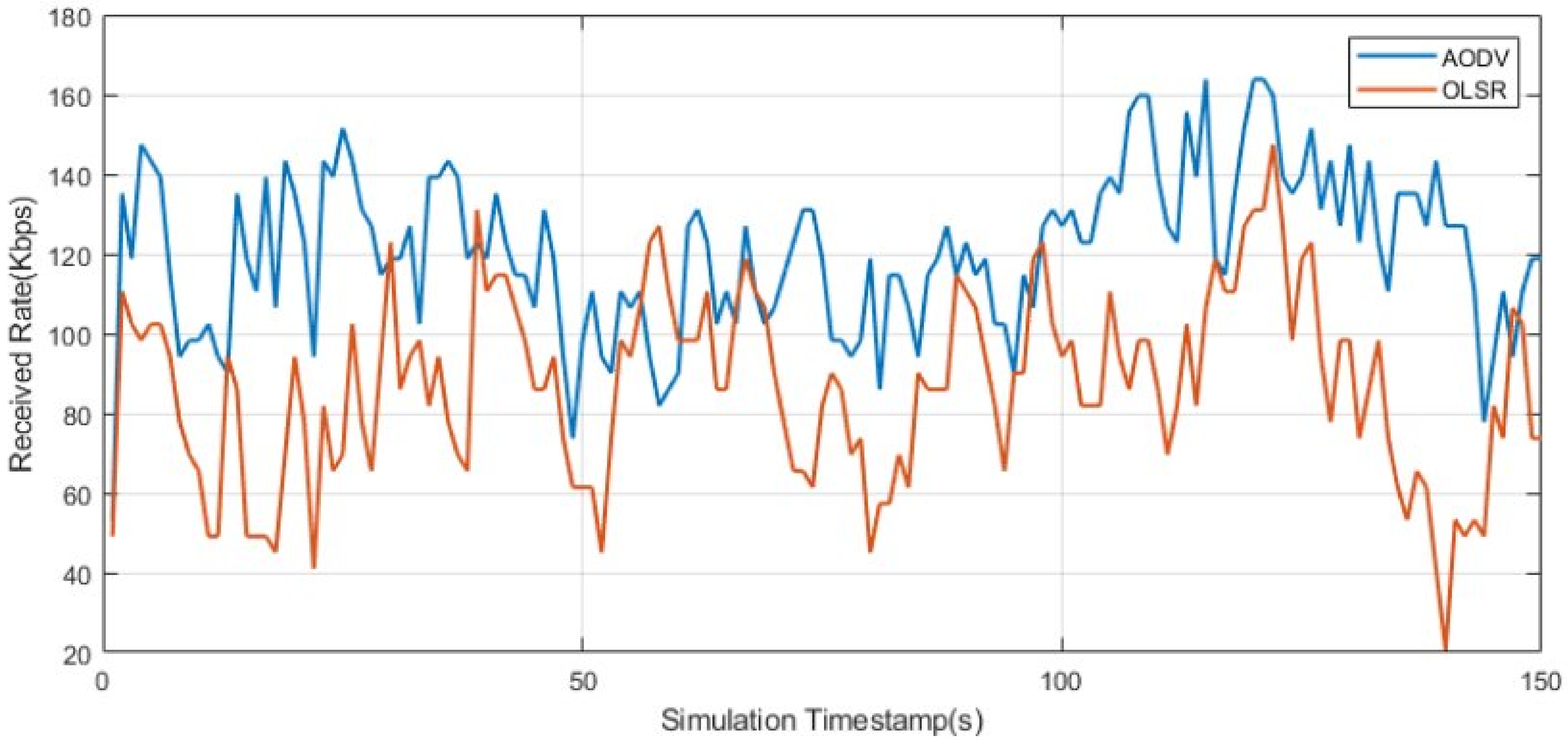


Figure 6.2: Comparision of Received Rate at different Simulation Timestamps

As shown in Fig 6.2, AODV shows about 37.88% higher received rate than OLSR, This is because OLSR updates its routing table periodically and thus does not always ensure fresh routes however, AODV established routes on demand and thus always ensures updated routes. Thus AODV ensures that the packets are always sent on a route that is more reliable for communication leading to an increase in the efficiency of the protocol to route packets which leads to an increase in the received rate.

2. **Throughput**

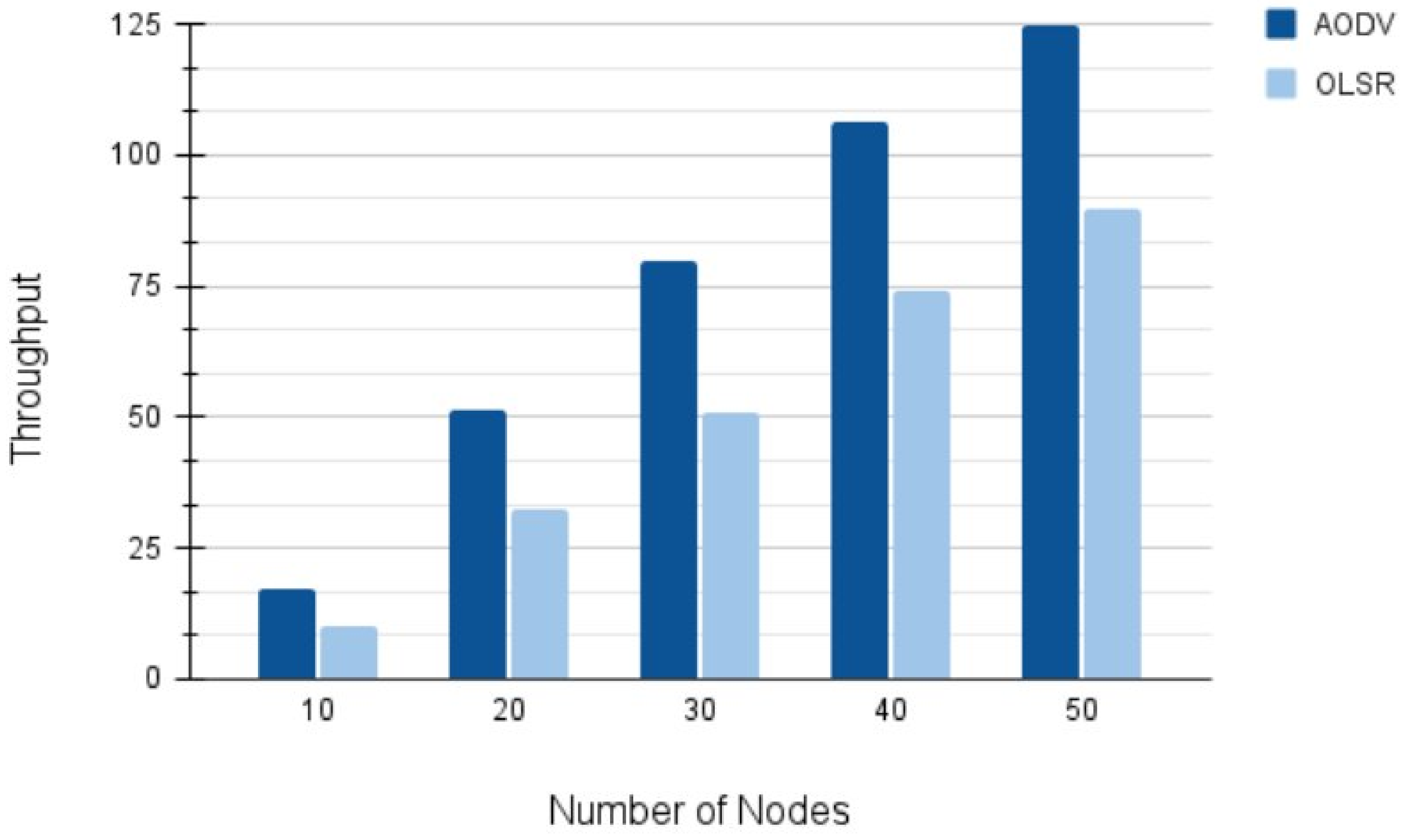


Figure 6.3: Comparision of Throughput at Varying Number of UAVs

Fig 6.3 shows that AODV shows 47.38% more throughput than OLSR for varying numbers of nodes in the network. In large networks with more number of nodes efficient utilization of resources becomes challenging but due to its reactive nature, AODV initiates route discovery only when needed, resulting in more efficient utilization of network resources whereas OLSR being proactive, incurs additional overhead in maintaining routing tables, which may limit its ability to scale as effectively as AODV.

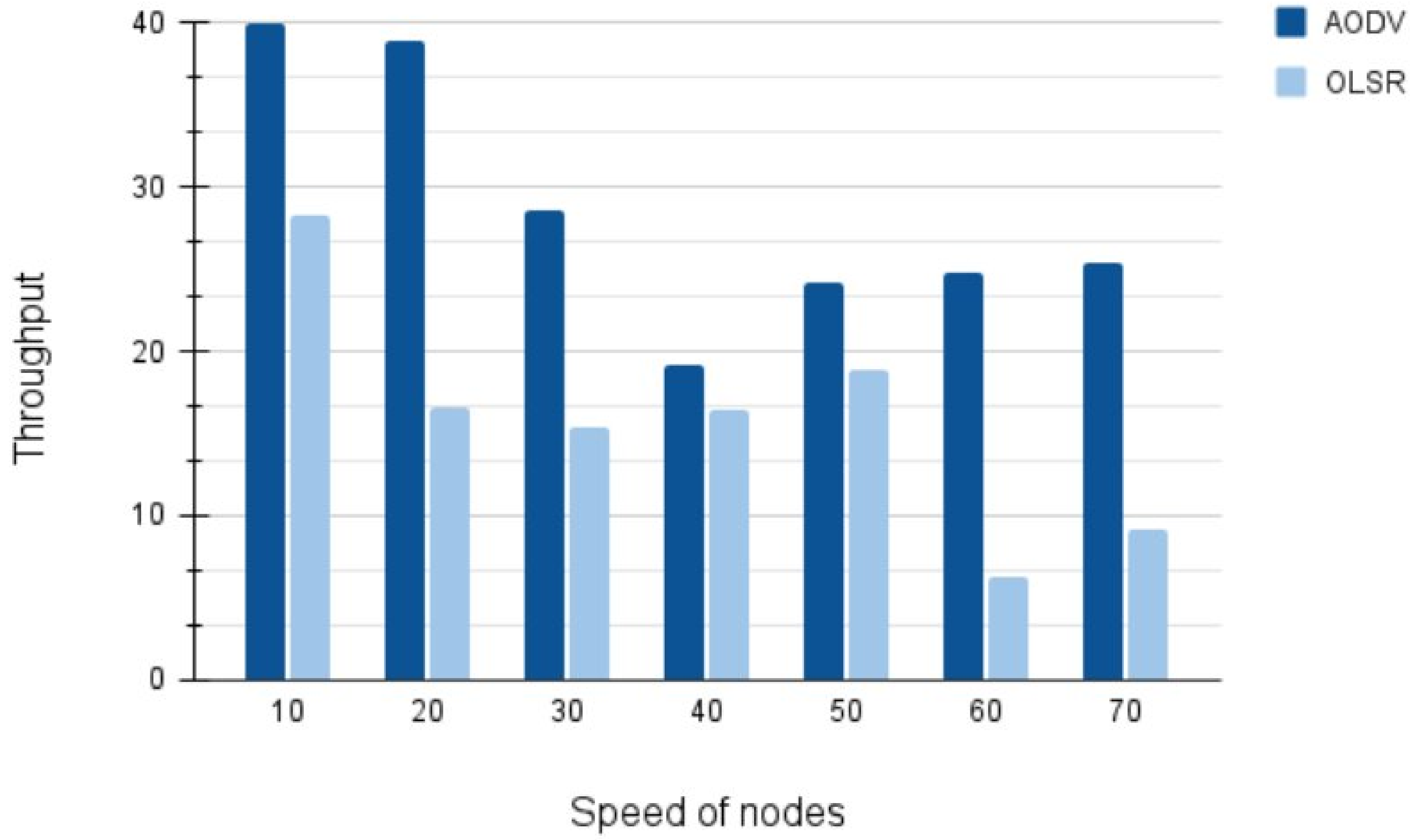


Figure 6.4: Comparision of Throughput at Varying Speed of UAVs

As shown in fig 6.4, as the speed of nodes increases, the throughput reduces but the throughput of AODV is more than OLSR at varying speeds. However, due to major differences in their values, it is not possible to draw quantitative conclusions. the reduction in the throughput for OLSR is more than that for AODV. As the speed of the node increases the networks become unstable. Due to the reactive nature of the AODV routing protocol and its ability to always fetch recent updates about the changes in the network topology, it is able to handle this instability in the network in a more effective way than OLSR.

3. **Average End-to-End Delay**

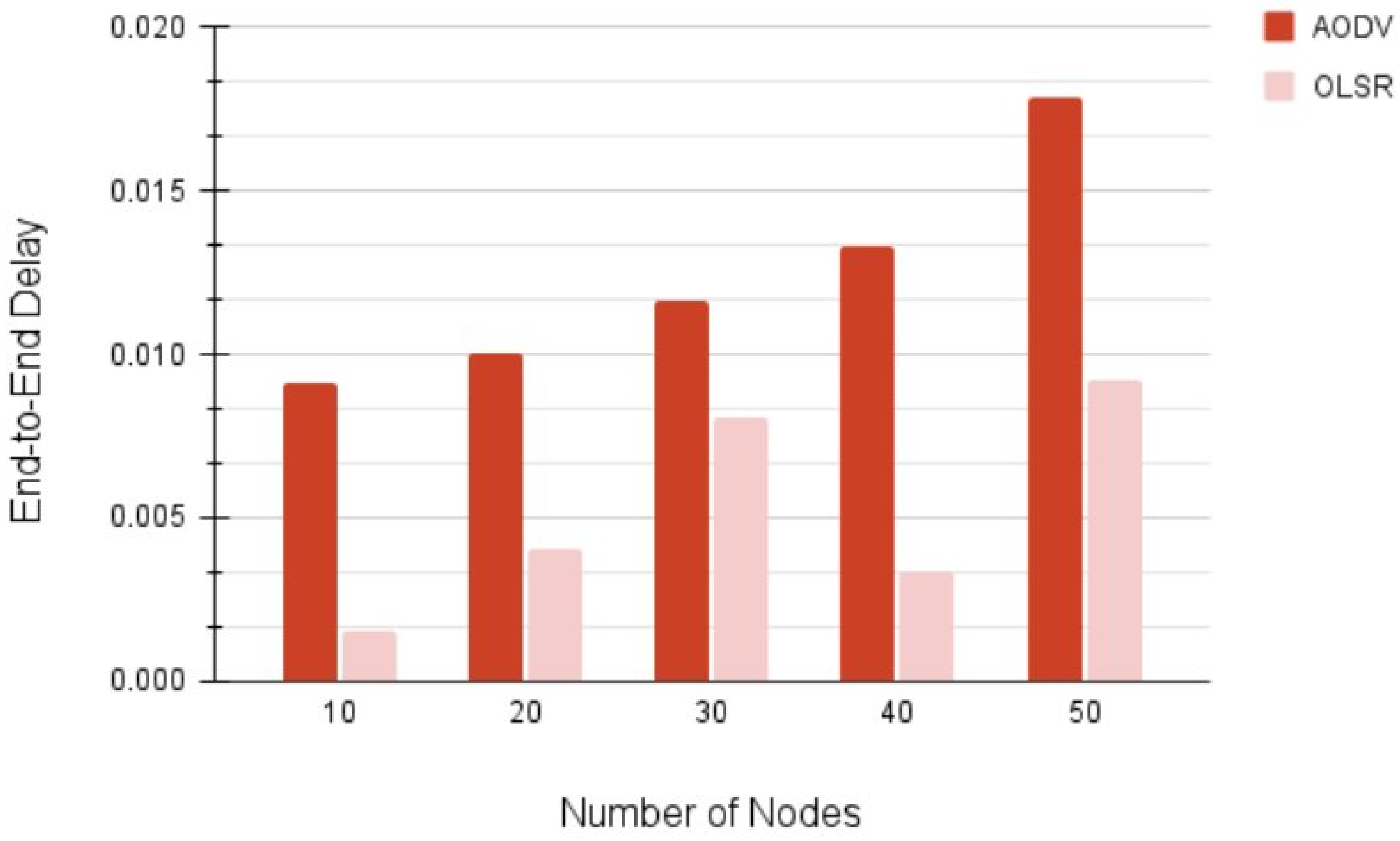


Figure 6.5: Comparision of End-to-End Delay at Varying Number of UAVs

Fig 6.5 shows that as the number of nodes increases, the increase in the end-to-end delay in the case of AODV is more than OLSR. On average AODV shows 3.16 times more end-to-end delay than OLSR. OLSR being a table-driven protocol, does not encounter long route setup time as updated routes are present in the routing table; on the other hand, AODV does not reuse routing information and has to introduce the route discovery process again and again when node data is to be transmitted. Hence, in the time-sensitive network, topology AODV shows poor delay bounds compared to proactive protocol OLSR.

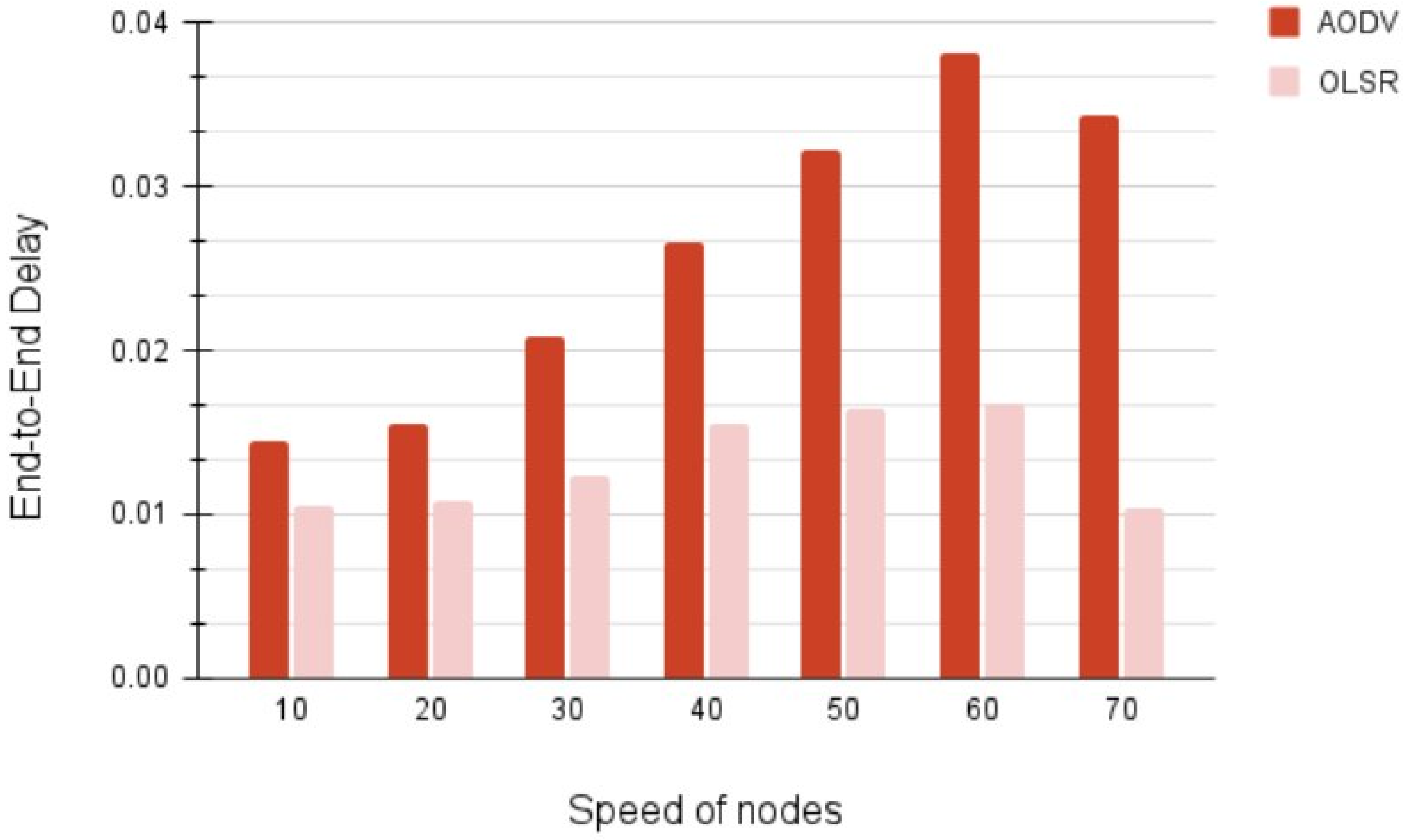


Figure 6.6: Comparision of End-to-End Delay at Varying Speed of UAVs

As shown in Fig 6.6, as the speed of the node increases, the increase in the end-to-end delay for AODV is more than OLSR. On average AODV shows 2 times more end-to-end delay than OLSR. Nodes moving with high speed introduce propagation and transmission delays in the network which gets added up to the already existing delay due to the AODVs feature of finding routes on demand. Thus as the number of nodes increases, the overall delay in AODV is more as compared to OLSR

6.2.4 Observations for Simulation 1

Based on the conducted simulations comparing AODV and OLSR protocols in FANET scenarios, the obtained results indicate that AODV exhibits superior performance in terms of throughput and received data rate when compared to OLSR. However, it is worth noting that AODV also introduces higher end-to-end delay compared to OLSR. But, In line with the research conducted by Leonov *et al.* [18], a comprehensive comparison of AODV and OLSR in terms of packet delivery ratio, end-to-end delay, average network throughput, and routing overhead revealed similar observations to our study.

Despite the increase in end-to-end delay for AODV compared to OLSR, According to the analysis by Leonov *et al.* [18] and Gupta *et al.* [13] we can conclude that AODV outperforms OLSR due to its superior performance in critical parameters such as received rate and throughput, which hold greater significance in FANET scenarios. Moreover, the inherent characteristics of AODV, such as its reactive

nature and on-demand routing, make it more suitable for highly dynamic and resource-constrained environments like FANETs, further justifying its preference over OLSR despite the tradeoff of increased end-to-end delay.
Thus based on these observations we conclude AODV reactive routing protocol to be more suitable for routing in FANET scenarios over the OLSR routing protocol.

6.3 Performance Analysis of Proposed Adaptive Hello Interval Approach (Simulation 2)

6.3.1 Simulation Setup Details for Simulation 2

For simulation, we used ns-3 simulator version 3.27. we used the AODV reactive routing protocol for analysis. To incorporate our proposed idea, we modified the AODV module in ns-3 to implement the proposed concept of adaptive hello interval. We compared three different approaches of setting hello interval values in the network, namely, AODV with a fixed hello scheme, EEAODV with the adaptive hello interval scheme, and the proposed modified EEAODV with an adaptive hello interval scheme. We considered a simulation area of 600m x 600m x 150m and used the Gauss Markov mobility model for UAV moments [5]. UAVs can move within this simulation area with a fixed transmission range of 150 m and UAVs speeds ranging from 5 to 30 m/s.
AODV being a reactive routing protocol, helps to always get updated and accurate information about the network topology and information about other network factors. By default, all nodes that join the network for data transmission send a hello message after every 1 sec to broadcast their presence in the network to other nodes in the network. That is the hello interval is fixed to 1 second. However, to balance the tradeoff explained before, we make the hello interval adaptive by setting its value dynamically based on the proposed approach. Thus the value of the hello interval is not fixed to 1 sec but is adjusted according to the proposed equation. The simulation time is set to 300 seconds.
We performed a total of 30 simulations, 10 each for a particular approach out of the three mentioned approaches. In each of these approaches, we vary the speed of nodes and check the output that shows the values of the network performance parameters mentioned. We record these values for each simulation for further analysis.

Table 6.2: Network Simulation Parameter for Simulation 2

Simulation Parameter	***Value***
Network simulators	*ns-3*
Simulation area	*600 x 600 x 150 m square*
Number of UAVs	*20*
UAVs speed	*[0-10,15,20,25,30,35,40,45,50] m/s*
Transmission range of a UAV	*150 m*
Simulation time	*300 s*
Mobility model	*Gauss-Markov 3D mobility model*
Routing protocols	*AODV*
Packet size	*512 bytes*
Data type	*CBR*
Antenna type	*Omnidirectional*
PHY/MAC protocol	*802.11b*
PropagationLossModel	*FriisPropagationLossModel*
Channel Type	*Wireless*
Data rate	*16 Kbps*

6.3.2 Performance Metrics for Simulation 2

- **Throughput**

  Throughput refers to the amount of data successfully transmitted or received over a network within a given time. It is calculated as a ratio of the number of bits transmitted and the time duration between the first packet transmission and the last packet reception.

  $$\frac{\sum Rec\_Pkt_{size} * 8}{Transmission\_Interval * 1024} \tag{6.3}$$

- **Overhead Size**

  To determine the Overhead efficiency, we first determine the overhead size by considering the size of all the control packets sent and received in the network.

  $$\sum Overhead_{size} = \sum ControlPacket_{size} - \sum DataPacket_{size} \tag{6.4}$$

- **Overhead Efficiency (OE)**
  In the context of the hello interval, there is a need to maintain a balance between the ability to detect links and the amount of overhead generated by the interval. A metric, Overhead Efficiency (OE), is used to understand this balance better. This allows us to analyze the trade-off. Essentially, a higher number of data packets received indicates that the links between nodes are being properly maintained and that link detection is effective.

  $$OE = \frac{\sum Rec\_Data\_Pkt_{size}}{\sum Overhead_{size}} \tag{6.5}$$

  A higher OE implies a better network because it can generate better throughput with a comparatively lower overhead cost

- **Energy Consumption**
  To evaluate energy consumption, we quantified the total energy used by all UAVs for transmitting and receiving control overheads per second. To estimate energy consumption, we utilized Han and Lee's energy consumption model [14], which assumes 200 µW of energy consumed for each byte of overhead transmission and 150 µW of energy consumed for each byte of overhead reception.

6.3.3 Results of Simulation 2

1. **Throughput**

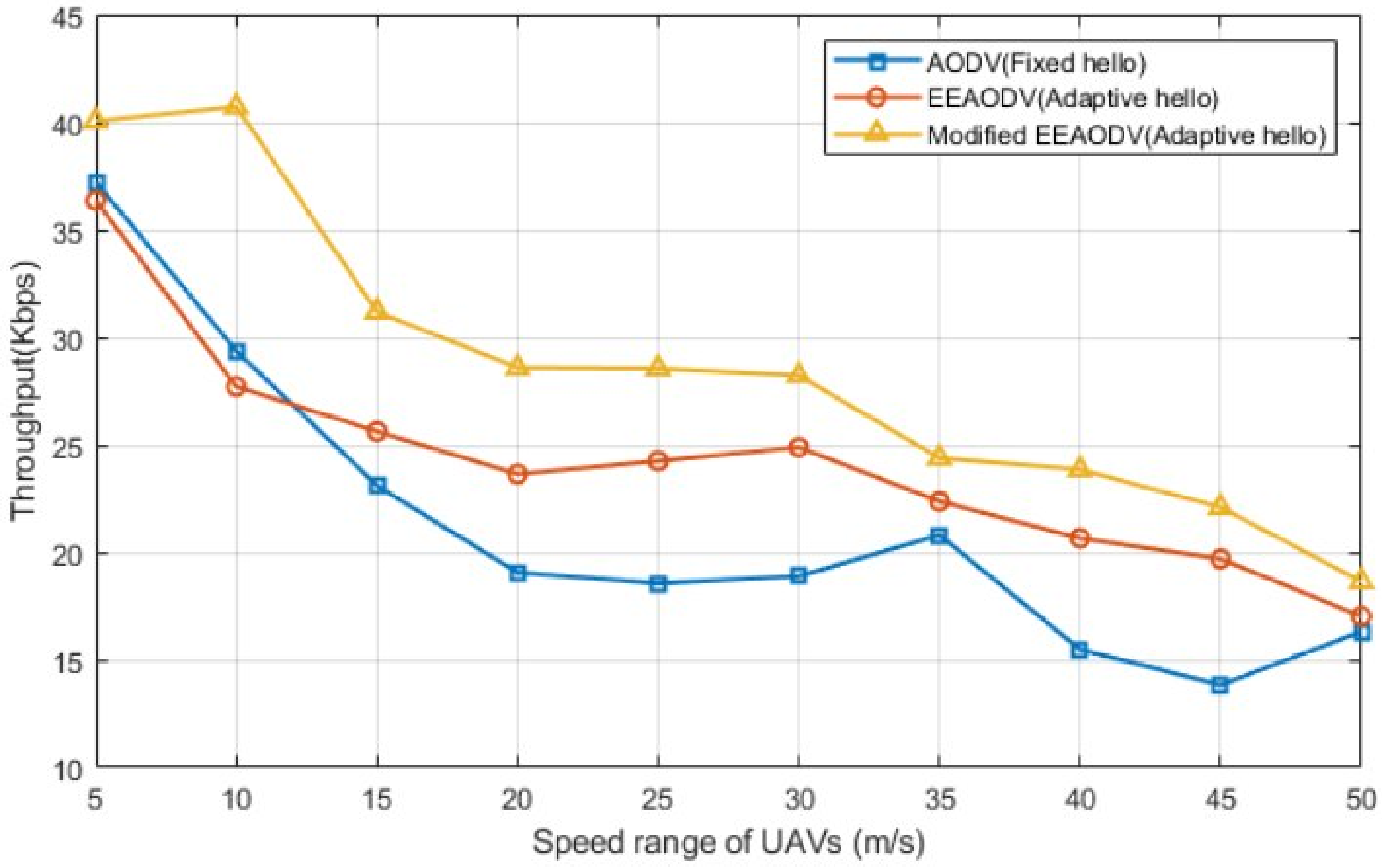


Figure 6.7: Comparision of Throughput at Varying Speed of UAVs

The throughput of a scenario depends on the efficiency of the routing protocol. As shown in Fig 6.7, the adaptive hello interval technique helps to improve the routing protocol's efficiency by improving the ability to detect link changes quickly compared to the fixed hello interval.

Moreover, the proposed scheme, which considers the relative speed of a node with respect to its neighbors instead of the speed, leads to better throughput. By incorporating the relative speed, the scheme gains a more accurate understanding of the node's neighborhood. This enables it to differentiate between cases where a node's speed is high but the relative speed is not, indicating a stable neighborhood, and cases where the node's speed is low but the relative speed is high, showing an unstable neighborhood. By dynamically adjusting the hello interval based on this information, the scheme effectively detects changes in the neighborhood and optimizes communication, resulting in improved throughput.

2. **Total Overhead**

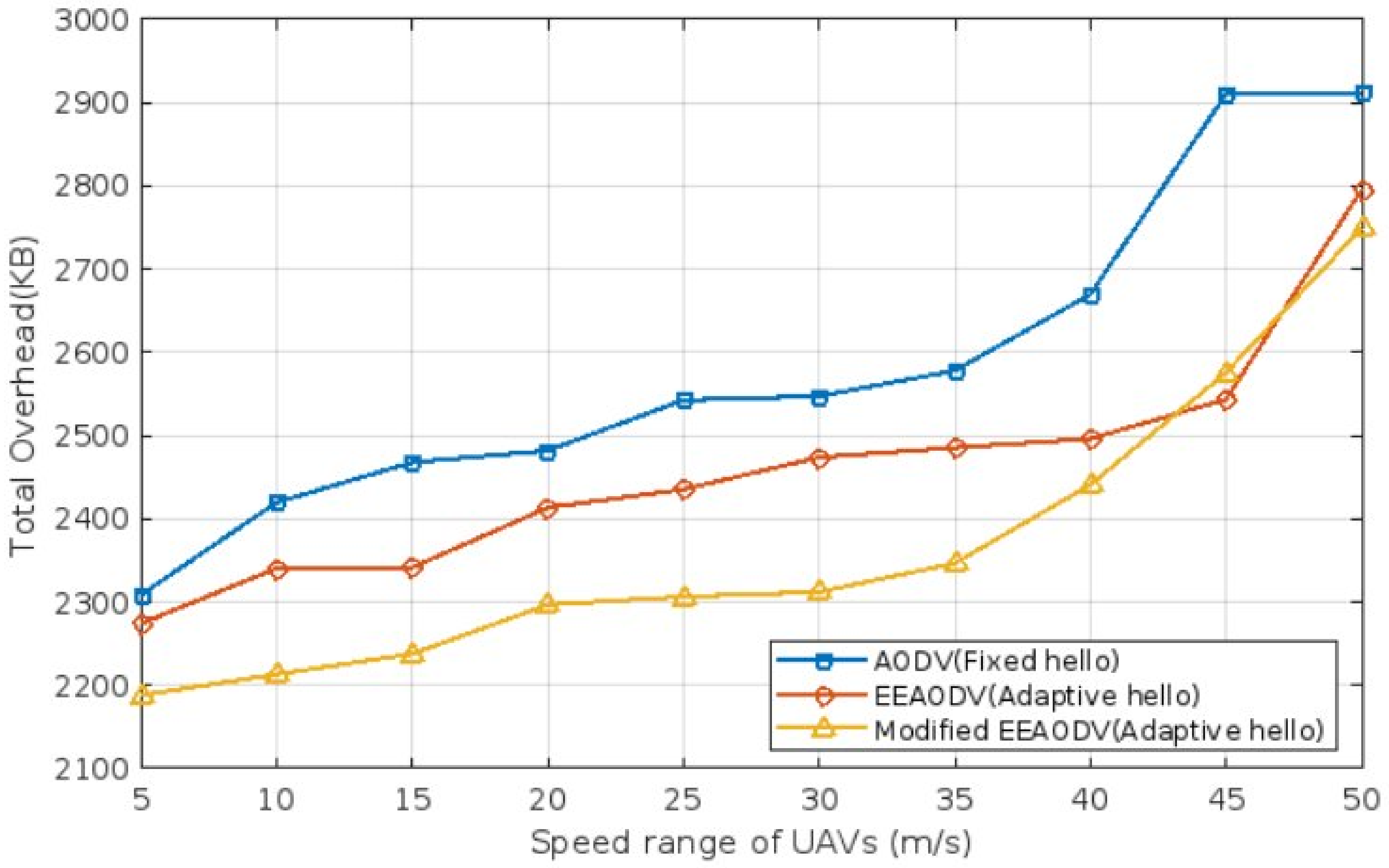


Figure 6.8: Comparision of Total Overhead at Varying Speed of UAVs

Hello messages being the control packets they add to the network overhead. With the fixed hello interval, nodes send hello messages even when there are no changes in the network topology. However, as shown in Fig 6.8, the adaptive hello interval reduces the frequency of hello messages by setting the hello interval value based on network conditions.

Moreover, the proposed scheme which sets the hello interval based on relative speed further reduces the overhead because it leads to the improved estimation of hello interval by also considering the situation in the neighborhood of the node as well.

3. **Overhead Efficiency**

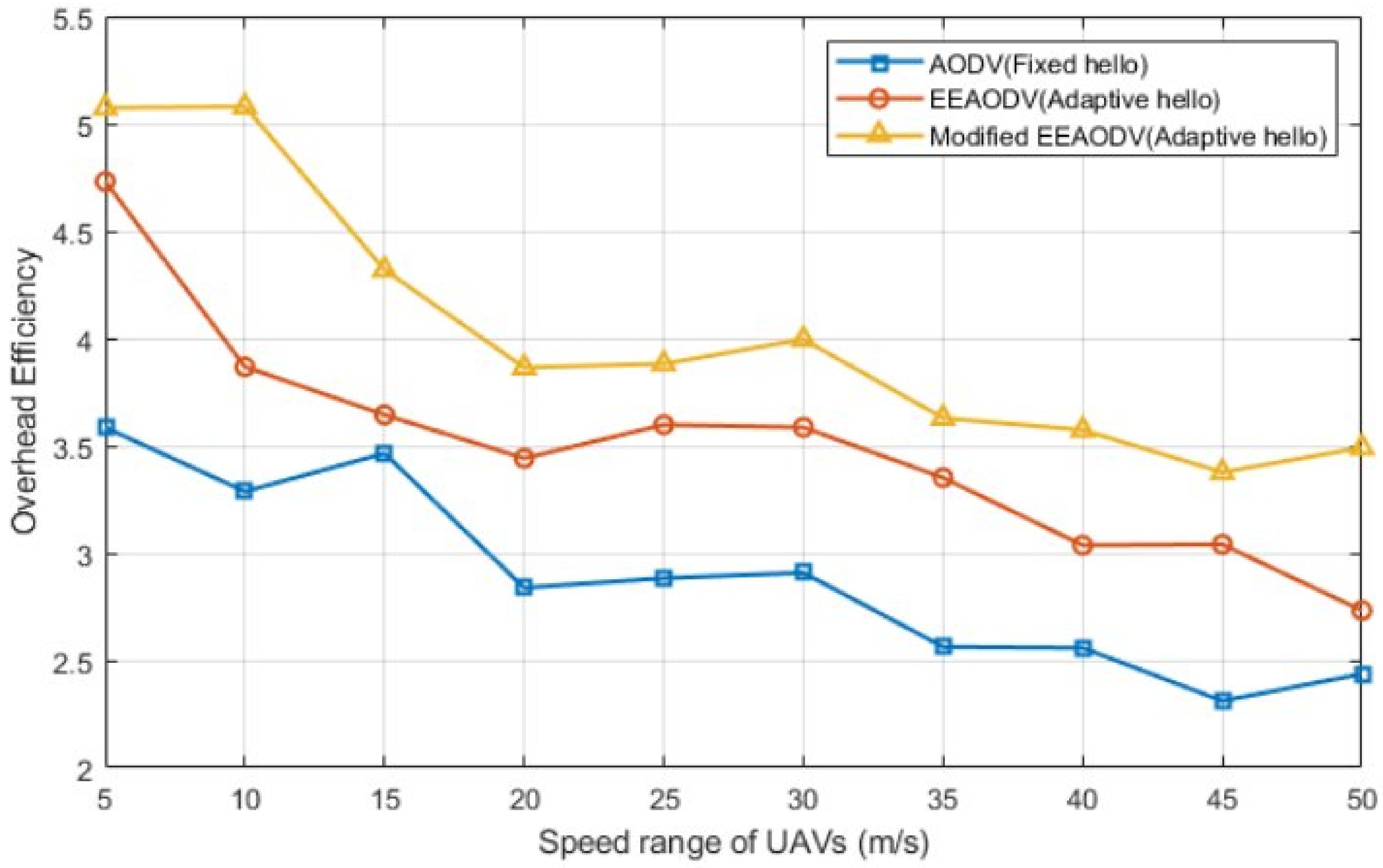


Figure 6.9: Comparision of Overhead Efficiency at Varying Speed of UAVs

Overhead Efficiency is defined as the ratio of the total data packets received in the network and overhead size. In fig. 6.8 we observed that the adaptive hello interval scheme helps to reduce the overhead in the network by suppressing unnecessary hello messages in the network. Now, since the overhead efficiency is inversely proportional to the overhead in the network, the reduction is overhead leads to an increased overhead efficiency as shown in fig 6.9.

4. **Energy Consumed**

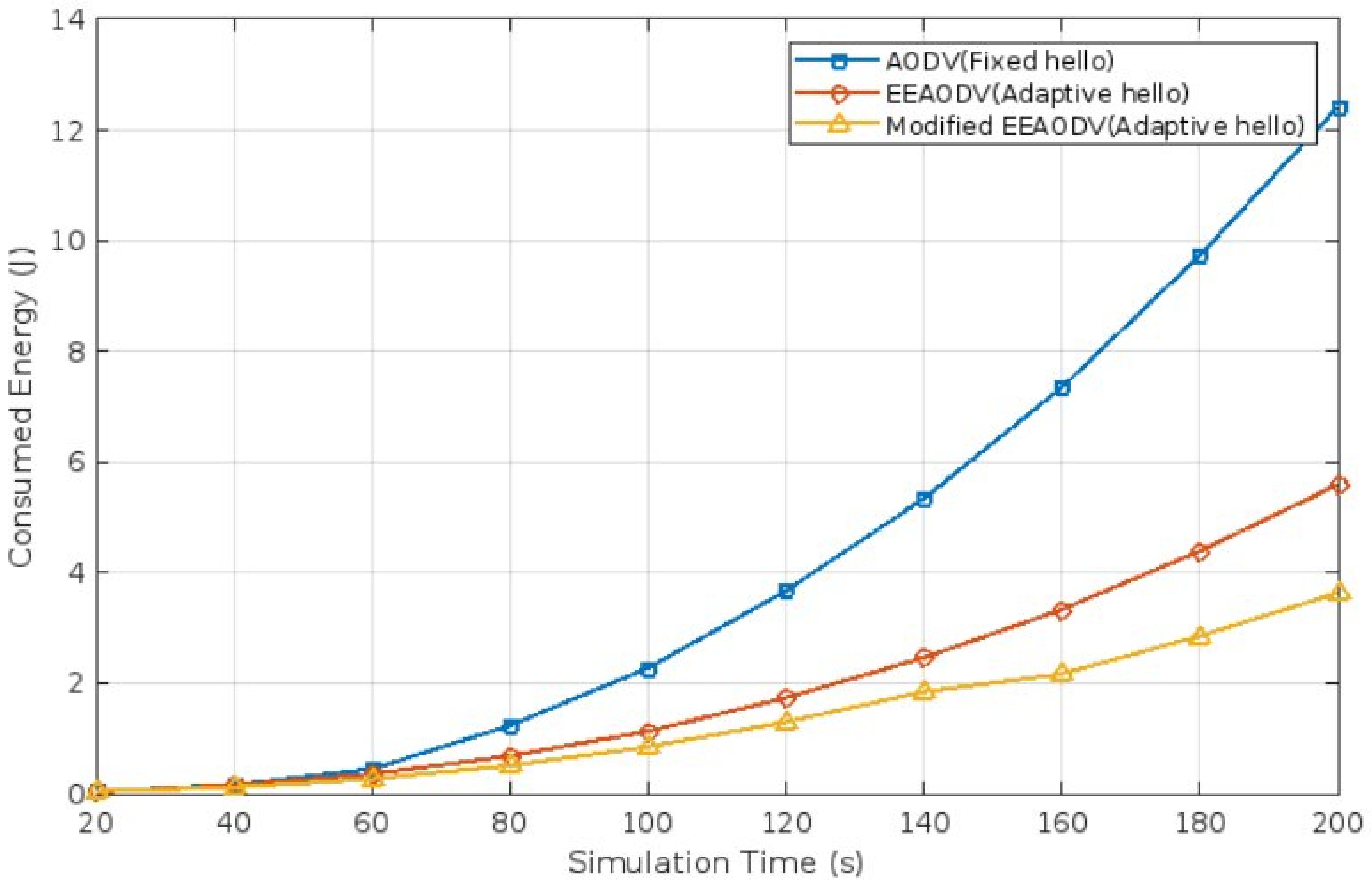


Figure 6.10: Comparision of Energy Consumed at Varying Simulation Time

The transmission and reception of hello messages consume energy. Thus the amount of energy consumed is directly proportional to the overhead expended. As observed in fig. 6.8 the adaptive hello interval scheme reduces the overhead in the nodes thus reducing the energy consumption of the nodes as well.

Further, the dynamic adaptation of the hello interval based on the relative speed information optimizes energy usage, resulting in lower energy consumption than the original equation as shown in Fig 6.10.

6.3.4 Observations for Simulation 2

The results show that in comparison to the fixed hello interval the other two adaptive hello interval schemes perform better in terms of various network parameters. This is due to the ability of the adaptive hello interval approach to balance the tradeoff between limited energy and quick link detectability in FANETs. We also, observe that the results obtained by the Modified EEHello based on the relative speed of the UAV gives better performance than the referred EEHello scheme this is due to the ability of the modified EEHello scheme to precisely consider the condition in the neighborhood of the UAV and setting the hello interval accordingly.

CHAPTER 7

Conclusion And Future Work

Conclusion

The proposed adaptive hello interval scheme successfully strikes a balance between saving energy consumption by increasing the hello interval at a cost of slow neighbor detectability and achieving quick neighbor detectability by reducing the hello interval but at a cost of high energy consumption. In order to measure the stability of the network at different timestamps, it considers factors such as network density, relative speed of UAV with respect to neighboring UAVs, and transmission range of UAV. An appropriate value for the hello interval at a particular timestamp is decided based on these factors.

Considering relative speed alongside absolute speed proves to be crucial in enhancing overall network performance. It prevents unnecessarily lowering the hello interval when a node has high absolute speed but its neighboring nodes have low relative speed, indicating a stable neighborhood. Likewise, it prevents setting the hello interval too high when a node has low absolute speed but its neighboring nodes have high relative speed, indicating an unstable neighborhood.

In simulation experiments, we considered a practical 3D scenario for FANETs and measured the performance of the proposed scheme with the default protocols AODV using different metrics. We observe that the proposed scheme could achieve improvements in terms of throughput, overhead, overhead efficiency, and energy consumption in the network. From these results, we conclude that the proposed scheme achieves an excellent trade-off and that existing protocols could become more balanced by the addition of the proposed scheme.

Future Work

The proposed scheme considers only three network parameters for deciding the value of the hello interval. We plan to study other network parameters that can affect the value of the hello interval and then incorporate those parameters into the equation for the hello interval. This will help in getting a broader sense of the network conditions thus helping in choosing the optimum value of the hello interval accordingly.

We also plan to check how reinforcement learning techniques such as Q-learning can be used in order to make a model that learns to adjust the hello interval based on the network conditions automatically.

References

[1] A. M. Abd Elmoniem, H. M. Ibrahim, M. H. Mohamed, and A.-R. Hedar. Ant colony and load balancing optimizations for aodv routing protocol. *Int J Sens Networks Data Commun*, 1:1–14, 2012.

[2] A. AlKhatieb, E. Felemban, and A. Naseer. Performance evaluation of ad-hoc routing protocols in (fanets). In *2020 IEEE wireless communications and networking conference workshops (WCNCW)*, pages 1–6. IEEE, 2020.

[3] M. S. Ayub, P. Adasme, D. C. Melgarejo, R. L. Rosa, and D. Z. Rodríguez. Intelligent hello dissemination model for fanet routing protocols. *IEEE Access*, 10:46513–46525, 2022.

[4] S. Banik, M. M. Mowla, and P. Sarkar. A new routing scheme for multi-cluster flying ad hoc networks. In *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, pages 323–328. IEEE, 2019.

[5] R. Bilal and B. M. Khan. Analysis of mobility models and routing schemes for flying ad-hoc networks (fanets). *International Journal of Applied Engineering Research*, 12(12):3263–3269, 2017.

[6] G. Carneiro, P. Fortuna, and M. Ricardo. Flowmonitor: a network monitoring framework for the network simulator 3 (ns-3). In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, pages 1–10, 2009.

[7] A. Chriki, H. Touati, H. Snoussi, and F. Kamoun. Fanet: Communication, mobility models and security issues. *Computer Networks*, 163:106877, 2019.

[8] T. Clausen and P. Jacquet. Rfc3626: Optimized link state routing protocol (olsr), 2003.

[9] R. Ernst and P. Martini. Adaptive hello for the neighborhood discovery protocol. In *37th Annual IEEE Conference on Local Computer Networks*, pages 470–478. IEEE, 2012.

[10] V. C. Giruka and M. Singhal. Hello protocols for ad-hoc networks: overhead and accuracy tradeoffs. In *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, pages 354–361. IEEE, 2005.

[11] C. Gomez, M. Catalan, X. Mantecon, J. Paradells, and A. Calveras. Evaluating performance of real ad-hoc networks using aodv with hello message mechanism for maintaining local connectivity. In *2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pages 1327–1331. IEEE, 2005.

[12] L. Gupta, R. Jain, and G. Vaszkun. Survey of important issues in uav communication networks. *IEEE communications surveys & tutorials*, 18(2):1123–1152, 2015.

[13] V. Gupta and D. Seth. Unmanned aerial vehicles (uavs): Evaluation of olsr, dsdv, aodv, and dsr dynamic routing protocols. In *2023 International Conference on Device Intelligence, Computing and Communication Technologies,(DICCT)*, pages 63–68. IEEE, 2023.

[14] S. Y. Han and D. Lee. An adaptive hello messaging scheme for neighbor discovery in on-demand manet routing protocols. *IEEE communications letters*, 17(5):1040–1043, 2013.

[15] N. Hernandez-Cons, S. Kasahara, and Y. Takahashi. Dynamic hello/timeout timer adjustment in routing protocols for reducing overhead in manets. *Computer Communications*, 33(15):1864–1878, 2010.

[16] Y. Huang, S. Bhatti, and S.-A. Sorensen. Adaptive manet routing for low overhead. In *2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–6. IEEE, 2007.

[17] V. Kanakaris, D. Ndzi, and K. Ovaliadis. Improving aodv performance using dynamic density driven route request forwarding. *arXiv preprint arXiv:1107.3630*, 2011.

[18] A. V. Leonov and G. A. Litvinov. About applying aodv and olsr routing protocols to relaying network scenario in fanet with mini-uavs. In *2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, pages 220–228. IEEE, 2018.

[19] I. Mahmud and Y.-Z. Cho. Adaptive hello interval in fanet routing protocols for green uavs. *IEEE Access*, 7:63004–63015, 2019.

[20] S. Makkar, Y. Singh, and R. Singh. Performance investigation of olsr and aodv routing protocols for 3 d fanet environment using ns3. *J. Commun. Eng. Syst*, 8(2):1–10, 2018.

[21] K. K. Naik, C. Balaswamy, D. Prabhakar, S. Kandula, and S. Begum. Performance of various ad hoc routing protocols using ns-3 simulator. *Annals of the Romanian Society for Cell Biology*, pages 133–138, 2021.

[22] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain. Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives. *IEEE Access*, 7:81057–81105, 2019.

[23] N.-U. Park, J.-C. Nam, and Y.-Z. Cho. Impact of node speed and transmission range on the hello interval of aodv routing protocol. , 43(2):241–251, 2018.

[24] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. Technical report, 2003.

[25] X. Qiu, Y. Yang, L. Xu, J. Yin, and Z. Liao. Maintaining links in the highly dynamic fanet using deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 72(3):2804–2818, 2022.

[26] V. K. Quy, N. T. Ban, and N. D. Han. A multi-metric routing protocol to improve the achievable performance of mobile ad hoc networks. In *Modern Approaches for Intelligent Information and Database Systems*, pages 445–453. Springer, 2018.

[27] M. U. Rahman and S. Abbas. Simulation-based analysis of manet routing protocols using group mobility model. In *2016 International Conference on Inventive Computation Technologies (ICICT)*, volume 1, pages 1–5. IEEE, 2016.

[28] P. Samar and S. B. Wicker. On the behavior of communication links of a node in a multi-hop mobile environment. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 145–156, 2004.

[29] D. K. Sharma, T. C. Behra, and C. Kumar. Notice of removal: An adaptive hello process for overhead reduction in manets. In *2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, pages 1–4. IEEE, 2015.

[30] K. Singh and A. K. Verma. Experimental analysis of aodv, dsdv and olsr routing protocol for flying adhoc networks (fanets). In *2015 IEEE international con-*

ference on electrical, computer and communication technologies (ICECCT), pages 1–4. IEEE, 2015.

[31] X. Tan, Z. Zuo, S. Su, X. Guo, X. Sun, and D. Jiang. Performance analysis of routing protocols for uav communication networks. *IEEE access*, 8:92212–92224, 2020.

[32] B. Zheng, K. Zhuo, H.-X. Wu, and H. Zhang. A novel adaptive hello mechanism based geographic routing protocol for fanets. In *International Conference on Ad Hoc Networks*, pages 33–45. Springer, 2021.