UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Design of Cross-Layer MAC and Routing Protocols for Autonomous UAV Networks

Permalink

https://escholarship.org/uc/item/13r496fs

Author

Garg, Shivam

Publication Date

2022

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

SAN DIEGO STATE UNIVERSITY
&
UNIVERSITY OF CALIFORNIA, IRVINE

Design of Cross-Layer MAC and Routing Protocols for Autonomous UAV Networks

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computational Science

by

Shivam Garg

Dissertation Committee:
Professor Sunil Kumar (SDSU), Co-Chair
Professor Alexander T. Ihler (UCI), Co-Chair
Professor A. Lee Swindlehurst (UCI)
Professor Duy H. N. Nguyen (SDSU)
Professor Athina Markopoulou (UCI)

2022

Portion of Chapter 2 © 2021 EWSN
Chapter 3 © 2021 IEEE
Portion of Chapter 4 © 2021 EWSN
All other materials © 2022 Shivam Garg

DEDICATION

To my family and friends.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ACK	Acknowledgement
AN	Airborne Network
AODV	Ad-Hoc On-Demand Distance Vector
BER	Bit Error Rate
BFS	Breadth First Search
BO	Buffer Occupancy
CDMA	Code Division Multiple Access
CNFM	Confirm
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear-to-Send
CUR	Channel Utilization Ratio
DSTO	Distributed Scheduling using Topological Ordering
EB-ET-DRAND	Distributed TDMA Scheduling Algorithm based on the Exponential Backoff Rule and Energy-Topology factor
ETD	Estimated Time-to-Destination
E-T-DRAND	Distributed TDMA Slot Scheduling Algorithm based on the Energy-Topology Factor
FCFS	First-Come-First-Serve
FDMA	Frequency Division Multiple Access
FIFO	First-In-First-Out
FW	Fixed-Wing
GPS	Global Positioning System
HC	Hop Count
HOL	Head of Line
IEEE	Institute of Electrical and Electronics Engineers
IL	Number of Interfering Links
IP	Internet Protocol
LAN	Local Area Network
LEPR	Link Stability Estimation-based Preemptive Routing
LLT	Link Lifetime
MAC	Medium Access Control
MANET	Mobile Ad-Hoc Network

MCA-AODV	Mobility and Congestion-Aware Ad-Hoc On-Demand Distance Vector
MCA-OLSR	Mobility and Congestion-Aware Optimized Link State Routing
MM-OLSR	Multi-Metric Optimized Link State Routing
MPR	Multi-Point Relay
OLSR	Optimized Link State Routing
PDR	Packet Delivery Ratio
PR	Piggyback Reservation
PST	Packet Service Time
QoS	Quality of Service
REP	Reply
REQ	Requisition
RERR	Route Error
RLT	Route Lifetime
RREP	Route Reply
RREQ	Route Request
RSSI	Received Signal Strength Indicator
RSWT	Route Switch
RTS	Request-to-Send
RW	Rotatory-Wing
SBA	Single-Beam Directional Antenna
SDN	Software Defined Networking
SIFS	Short Inter-Frame Space
SNR	Signal-to-Noise Ratio
SRCM	Semi-Random Circular Movement
ST	Smooth-Turn
SWAP	Size, Weight and Power
TC	Topology Control
TDMA	Time Division Multiple Access
TSDP	Two-Step Discarding Policy
TTE	Time-to-Expiry
TTL	Time-to-Live
UAV	Unammed Aerial Vehicle

WRED Weighted Random Early Detection

ACKNOWLEDGMENTS

First, I would like to gratefully acknowledge and express my sincere thanks to my advisors, Professor Sunil Kumar and Professor Alexander T. Ihler, for their continuous guidance and support throughout my doctoral study. My special thanks go to Professor Sunil Kumar for giving me a wonderful opportunity to pursue my doctoral study at San Diego State University (SDSU) and the University of California, Irvine (UCI). His patience, timely comments, invaluable support, and guidance have certainly helped me complete this dissertation. My co-supervisor, Professor Alexander T. Ihler, has always been ready to give me advice and suggestions that I found important and helpful during my doctoral study. I have learned so much from their remarkable technical knowledge and research enthusiasm. I would also like to express my gratitude to Professor Duy H. N. Nguyen at SDSU, Professor A. Lee Swindlehurst and Professor Athina Markopoulou at UCI, for serving in my Ph.D. committee, giving me insightful comments and raising many interesting questions during my presentations.

I am grateful to Professor Fei Hu of University of Alabama, Tuscaloosa, for his advice and help throughout the process. I would like to acknowledge the contributions of my fellow labmate Venu Sri Sushma Kuchipudi in the design of directional TDMA MAC protocols, discussed in Chapter 3. I would also like to extend my warmest thanks to my current and former labmates, Shreyas Devaraju, Dr. Moein Parsinia, Dr. Btissam el Khamlichi, Gitanjali Ghadge, Vishakha Vijaykumar, Sourabh Dhoubhadel, Sheshmani L. Yadav, Mohit Agarwal, Dr. Abdelkabir Sahnoun and Ameena Badarudeen, for their help and kindness, and for the wonderful time I had during my doctoral study. I want to thank the Computational Science Research Center and the U.S. Air Force Research Laboratory for providing financial support during my PhD research.

I cannot imagine my doctoral journey without my friends, Krishna Dogney, Nandini Venkatraman, Puneet Bhushan, Dr. Van Ly Nguyen, Dr. Ghanshyam Mishra and Anjali Chaure, who have always been with me and showed their support whenever I needed.

On a more personal level, I would like to express my deepest love and gratitude to my parents, Mr. Braj Mohan Garg and Mrs. Nisha Garg, and other family members. They have always encouraged me to pursue my passion, and gave me their unwavering support. I hope this accomplishment makes you all proud of me.

VITA

Shivam Garg

EDUCATION

Doctor of Philosophy in Computational Science — **2022**
University of California — *Irvine, California*
San Diego State University — *San Diego, California*

Master of Science in Electrical Engineering — **2017**
San Diego State University — *San Diego, California*

Bachelor of Technology in Electronics and Communications Engineering — **2015**
Jaypee Institute of Information Technology — *Noida, India*

RESEARCH EXPERIENCE

Graduate Research Assistant — **2015–2022**
San Diego State University — *San Diego, California*

Intern — **Summer 2017**
University of Alabama — *Tuscaloosa, Alabama*

Intern — **Summer 2014**
San Diego State University — *San Diego, California*

PUBLICATIONS and AWARDS

1. **S. Garg**, V. S. S. Kuchipudi, E. Bentley, and S. Kumar, "A real-time, distributed, directional TDMA MAC protocol for QoS-aware communication in multi-hop wireless networks," *IEEE Access*, vol. 9, pp. 26343-26361, Feb. 2021.

2. **S. Garg**, A. Ihler, E. S. Bentley, and S. Kumar, "A cross-layer, mobility and congestion-aware routing protocol for UAV networks," *IEEE Trans. Aerosp. Electron. Syst.* (under 2nd round of review).

3. **S. Garg**, A. Ihler, and S. Kumar, "A cross-layer, hybrid reactive routing protocol for autonomous UAV networks," (under preparation for journal submission).

4. **S. Garg**, N. Venkatraman, E. S. Bentley, and S. Kumar, "An asynchronous multi-beam MAC protocol for multi-hop wireless networks," *IEEE Int. Conf. Comput. Commun. Netw.*, pp. 1-9, 2022. [**Received the best paper award**]

5. **S. Garg**, "An adaptive and low-complexity routing protocol for distributed airborne networks," in *Proc. Int. Conf. Embedded Wireless Syst. Netw.*, pp. 187-191, Feb. 2021.

6. **S. Garg**, A. Ihler, and S. Kumar,"Accurate link lifetime computation in autonomous airborne UAV networks," *arXiv:2202.00056* [cs.NI], Jan. 2022.

7. Qualcomm award for the best poster presentation entitled, "A QoS-aware routing protocol for airborne wireless networks," *ACSESS 2021*, SDSU.

8. Qualcomm award for the best poster presentation entitled, "A real-time, distributed, directional TDMA MAC protocol for multi-hop wireless networks," *ACSESS 2019*, SDSU.

ABSTRACT OF THE DISSERTATION

Design of Cross-Layer MAC and Routing Protocols
for Autonomous UAV Networks

By

Shivam Garg

Doctor of Philosophy in Computational Science

San Diego State University and University of California, Irvine, 2022

Professor Sunil Kumar (SDSU), Co-Chair
Professor Alexander T. Ihler (UCI), Co-Chair

Autonomous networks of unmanned aerial vehicles (UAVs) have many civilian and military applications. These networks experience a wide variety of network configurations and communication constraints (including node density, speed, and trajectory), resulting in a highly dynamic and unpredictable network topology. In addition, these networks support diverse and time-varying applications that can include different traffic types and priorities, data generation rates, session lengths, and reliability and latency tolerance.

In this dissertation, we develop distributed, cross-layer medium access control (MAC) and routing protocols to provide robust and reliable communication in autonomous and decentralized UAV networks, in which the network topology and traffic conditions change frequently and the future node trajectories are not known.

First, we present a mathematical framework to compute the link lifetime for a realistic node mobility model, followed by the design of a novel, distributed time division multiple access (TDMA) scheme for directional communication in multihop networks. This scheme includes a low-complexity, rank-based scheduling mechanism, which effectively adapts to the changes in the network and quality of service (QoS) demands in real time with significantly reduced

overhead and delay, and improves both channel utilization and fairness in channel access allocation.

In the subsequent chapters, we focus on routing protocols, which discover and select high-quality routes, and switch to alternate routes in response to changes in the available communication resources, observed traffic patterns, and performance demands to make the best use of the network resources.

Traditional topology-based routing schemes are slow to adapt to changes in topology and traffic, and typically select a route without considering the effect of intra-flow interference on the selected route. To address these issues, we present an adaptive, cross-layer, mobility and congestion-aware proactive routing protocol for decentralized UAV networks. Our protocol includes a novel, multi-step and multi-metric, inter- and intra-flow interference-aware route selection mechanism, which selects a stable, longer-lasting and less congested route. It uses a preemptive route switching mechanism to prevent potential packet drops due to congestion and topology changes, and a periodic queue management mechanism to prioritize transmitting packets with a lower survivability score, and discard packets that are likely to expire before reaching their destination.

Proactive routing protocols can incur large control and computational overhead, and may be vulnerable to the security threats. In contrast, reactive routing protocols incur much lower control and computation overhead, but the resulting, on-demand route discovery introduces large routing overhead and delay in settings with frequent topology changes and link breaks, such as UAV networks. We address these issues via a novel, hybrid mobility- and congestion-aware reactive routing protocol, which discovers routes on demand and preemptively switches to another high-quality route within the region around the selected route. This significantly reduces the number of route discoveries and overhead from route control and computation. Despite having limited network topology information, our proposed routing scheme provides superior flow throughput performance.

We show via network simulation results that our proposed MAC and routing protocols significantly outperform existing schemes across a variety of different network and traffic settings.

Chapter 1

Introduction

1.1 Decentralized, Autonomous UAV Networks

An airborne network (AN) is an essential component for the next generation air transportation system, which can self-configure to provide seamless, low-cost and secure connectivity to both aerial and surface nodes [1–4]. Such ANs consist of multiple manned (aircrafts) and unmanned aerial vehicles (UAVs), which use flight-to-flight communication for low latency information sharing, and provide scalability [1–3, 5–7]. Applications of ANs in civilian and military sectors include environmental sensing, disaster management, traffic and urban monitoring, patrolling, surveillance, search and rescue, and relaying networks [1–4, 8]. These applications have varying network design configuration and communication constraints, which include node density, speed, predefined or self-generated trajectories or group mobility, radiation pattern, communication range, computation capacity, degree of cooperation, control and coordination [1–4].

Due to the reduced cost of UAVs, fast deployment, device autonomy and increased flight time capabilities, the autonomous UAV networks can provide network reliability and fault

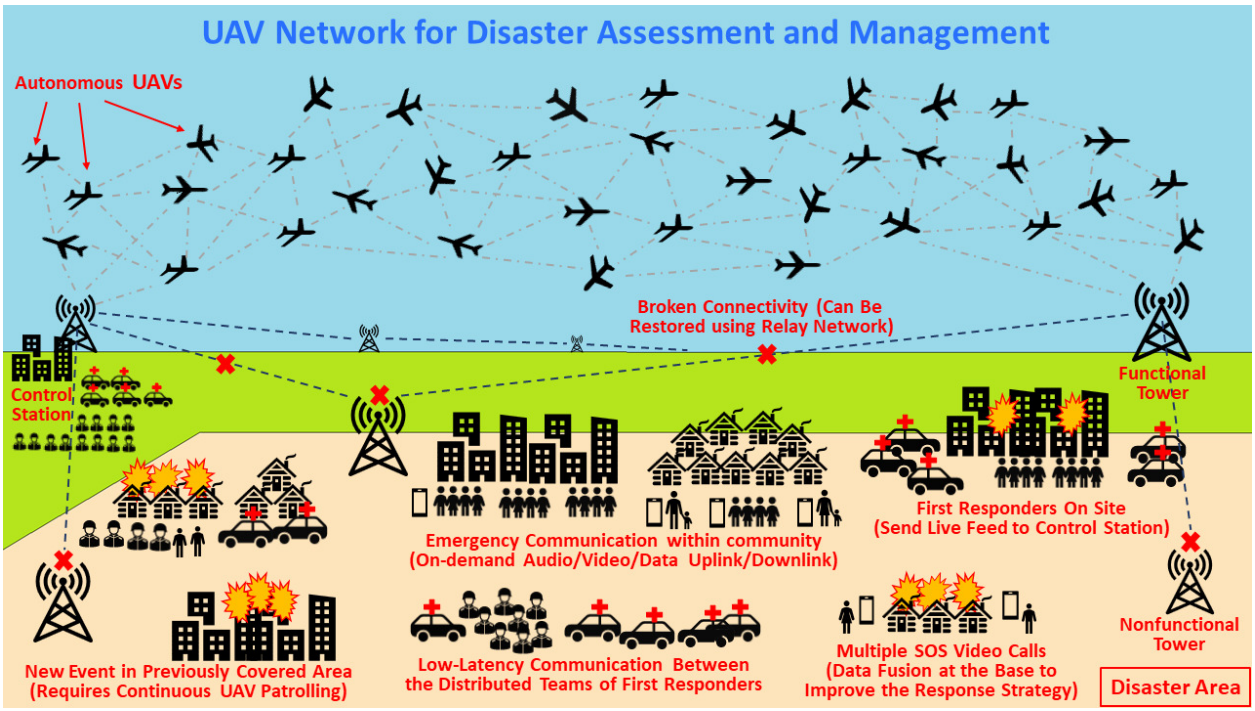


Figure 1.1: Illustration of an autonomous UAV ad-hoc network for disaster assessment and management in an inaccessible area.

tolerance, reduce task completion time through collaboration, and adapt to dynamic task requirements [1–4, 7, 9–11]. For example, Fig. 1.1 shows a disaster area (e.g., earthquake damaged area) where latency-sensitive and high-throughput audio, video, or other data events (such as damaged infrastructure, fire breakouts, SOS requests, and live updates) can occur randomly. Since the existing communication infrastructure in the disaster area may be damaged, a decentralized, multi-hop, ad-hoc network of UAVs is used to provide reliable and low-cost communication for quick damage assessment (i.e., information gathering) and response (e.g., task and resource allocation to the first responders) [7,8]. Since it is difficult to predict these events beforehand, the UAVs must patrol the affected area continuously, which often requires for the UAVs to fly on unspecified trajectories and switch to a new trajectory from time to time. This results in a highly dynamic network topology with intermittent links and traffic congestion. In addition, the source and destination nodes may be inside or outside the disaster area, and the number of traffic flows can vary over time. Therefore, the relay UAV nodes should find efficient, stable and less-congested routes to other nodes in the

area to provide reliable and low-latency communication.

In addition to the distinct network characteristics discussed above, ANs also accommodate application-specific task requirements in an infrastructure-less environment, which makes it significantly different from traditional wireless networks [12]. The application-specific requirements include traffic type (e.g., audio, video, statistics and sensor values of a node) and priority, data generation rate, session length, reliability and latency tolerance [13]. Note that these can also vary with time, which results in varying quality-of-service (QoS) demands. Designing a robust communication mechanism for autonomous, decentralized (no supervisory node) UAV networks is, therefore, a challenging task because the network topology also changes frequently [1–4, 9, 11, 13–20].

1.2 Adaptive and Scalable Protocols

A holistic system design can facilitate a reliable communication in autonomous, infrastructure-less ANs if it could identify the region(s) of interest in the network, understand the traffic needs, and provide a tailored solution for each region by making the best use of available resources, including the network architecture and interplay between its layers (e.g., physical, MAC (medium access control), routing, transport, application) [13]. It requires a comprehensive knowledge of the following factors and their interactions with each other.

(i) Network architecture: It includes network statistics, such as network size and node density, which impacts the network connectivity (e.g., dense vs. sparse, always connected vs. disconnected). In addition, the following network information can also be helpful: Availability of regional server(s) to obtain an updated information of a region; centralized vs. distributed architecture; GPS location of node; are nodes SDN (software defined networking) capable?; is it a leader-follower formation (it determines which node(s) can control

other nodes and how much)?; is it a homogeneous network (i.e., nodes are equipped with the same antenna, follow the same communication protocols and have similar storage and computational capabilities)?

With this knowledge, a source node can identify (and actively track) the region around its destination node. This affects the routing (signalling) overhead, which, in turn, impacts the channel utilization for data packets, computational overhead and route reliability. For example, if the links within a region are unstable and the source node cannot obtain an accurate location information of the destination node, it can use a proactive routing protocol, which immediately provides new route(s) to the destination node, as needed. Whereas, position-based routing protocols are suitable when the region information is available because they use greedy forwarding and incur very low routing overhead.

(ii) Network Traffic: The number of flows in the network can vary with time. Moreover, the source nodes can have data for the common sink or gateway node(s) or other nodes in the network. In the former case, the flows converge at a common destination node, resulting in node starvation around it, and thereby high congestion. Existing solutions to address this problem include the use of a contention-free channel access mechanism, which assigns a unique resource (e.g., time-slot/ frequency/ code) to each node in the neighborhood of destination node to alleviate congestion. In the latter case where flows are distributed in the network, the routes selected by the source nodes may intersect, resulting in a high congestion at hotspot nodes (i.e., node that serves more than one flow). This requires an intelligent routing protocol design, which can find the interference-aware routes by using node statistics (e.g., packet service delay, number of interfering links in the 1-hop neighborhood) and preemptively switch to an alternate route to prevent congestion buildup.

In addition, each flow can have its own unique requirements, such as the flow priority, latency, data generation rate, and session duration. Since these requirements can also change with time, we need a queue management mechanism at the node-level, which can determine the

packet priority on-the-fly and transmit the high-priority packets. A node can rearrange the packets in its queue based on the following metrics: Estimated remaining distance (both in hops and time) to the destination node, packet's remaining time-to-expiry (TTE), packet and/or flow priority, flow fairness, and the availability of the downstream nodes and their current link quality. In addition, a node can discard packets which are likely to expire before reaching the destination node, due to the limitations imposed by the current network topology and traffic. This prevents resource wastage and can improve the flow throughput.

(iii) Selection of routing protocol: Selecting an efficient routing protocol design is essential to support an application over a multi-hop network. The performance of a routing protocol depends on the node distribution and their connectivity [21, 22]. The node connectivity, in turn, depends on the node trajectory, and thereby, on the underlying mobility model [21, 22]. Therefore, designing and evaluating routing protocols for ANs require the use of mobility models, which can produce realistic node movements [2, 21–23].

The high node speeds in ANs result in the frequent changes in network topology, wherein the existing links break and new links form. As a result, the routes between the source-destination pairs and traffic conditions on them change frequently, which make routing a very challenging task. To ensure an uninterrupted and reliable communication with the destination node, a source node should frequently monitor the route quality in terms of route lifetime (RLT) (i.e., duration after which route would break), path latency, congestion along the route, route capacity (i.e., how many packets can be sent over the route) and stability. The selected route should adapt to these varying network and traffic conditions by taking preemptive measures, such as switching to an alternate route when the quality of current route degrades, balance the traffic load over non-interfering routes to accommodate high traffic flows, collaborate with other source nodes to find mutually beneficial routes, or assign more resources (e.g., relay nodes or channels) along the route to improve the connectivity of the region and decrease the congestion along the route.

However, designing a routing protocol with this type of adaptability requires collecting the node and link statistics, such as the load and queuing delay at a node, number of flows, interference, connectivity with 1-hop neighbors and how long would they remain connected, their link capacity, bit error rates (BER), signal to noise ratios (SNR), remaining battery life and willingness to participate as relays, and the route length in hops. Since these metrics can change with time, source node should collect their updated values periodically, which incurs high signalling and computational overheads in a decentralized network. Therefore, a scalable hybrid routing protocol (e.g., combination of proactive and reactive routing) design is desired, which can decrease the control overhead without degrading the adaptability.

Another routing design challenge is that the values of many of the above metrics change after the source node starts data transmission on the selected route, due to the induced intra-flow interference. As a result, selecting a route based on such metrics can quickly degrade the route quality. Therefore, a source node should use only the relatively stable metrics in the route selection to reduce frequent route switching or rediscovery.

(iv) Channel access mechanism: MAC protocol at the data link layer is responsible for forwarding the packets hop-by-hop towards the destination node, and performing error correction. Its design directly affects the QoS of the traffic flow. Therefore, a MAC protocol should efficiently resolve the channel contention and packet collision issues, and reduce the node starvation problem by fairly allocating the channel access to all nodes.

Recently, the interest in the use of directional antennas has increased notably due to their ability to significantly improve the network throughput by enabling the spatial reuse, reduce interference, and provide longer transmission range for a given transmit power [6, 24–27]. Therefore, the directional communication is used in many different systems, including the 5G systems, radar systems, wireless LANs (local area networks), MANETs (mobile ad-hoc networks) and ANs [6, 25–30]. However new problems arise with directional beams, such as the hidden terminal, node deafness, MAC layer capture, etc. [29].

Each node in a multi-hop mobile network must track its 1-hop neighbor nodes and their statistics (such as BER, load, queuing delay, packet forwarding rate, channel access collision rate) to successfully forward its packets towards the destination node. This requires frequent neighbor discovery, which is time consuming (especially when using directional antennas).

The performance of a multi-hop network is improved when the routing protocol can optimally use the relay nodes along the route. This requires a careful design of the MAC protocol, based on the network architecture and node capabilities. For example, if the network consists of power constrained nodes which are communicating over a long distance using a directional antenna, a contention-free MAC scheme is suitable in which both the transmitter and receiver nodes know when to communicate, instead of sensing the channel continuously. However, these schemes incur a considerable overhead and delay in computing a non-conflicting schedule for all nodes. On the other hand, contention-based schemes are more suitable for the decentralized and dynamic network topology, where the routes between source-destination pairs can change frequently. However, these schemes suffer from the hidden and exposed terminal problems, which increase the packet collisions and reduces flow throughput.

(v) Physical layer specifications: The carrier frequency, transmit power, channel error rate, choice of modulation scheme, coding, and signal propagation characteristics impact the packet reception at the receiver node [31]. At the same time, the antenna type (e.g., omnidirectional or beamforming, fixed or flexible, number of spatial streams) and beamwidth affect the transmission range, coverage and interference. The Doppler effect also becomes crucial when node speed is high. Therefore, a careful consideration of the propagation environment and underlying physical layer are important in ANs.

An application-specific communication mechanism can consider different combinations of the above parameters for a robust cross-layer protocol design. We have summarized these parameters in Table 1.1. Such a system design should consist of new (i) routing protocol to

select high quality (long-lasting and less-congested) routes with low routing overhead and delay, and preemptively switch to an alternate route when the quality of primary route(s) degrades or better quality routes becomes available, (*ii*) MAC protocol capable of adapting to the changes in network topology in real-time, which can assist upper layers in decreasing the need for a fresh route discovery, and (*iii*) QoS-aware packet scheduling, which can fairly prioritize critical data for transmission, even when the traffic changes with time. In addition, these cross-layer protocols should consider the characteristics of the environment, underlying physical layer, practical node trajectories and QoS requirements; and have a lower computational overhead [12, 32].

Table 1.1: Parameters for cross-layer protocol design.

Category	**Parameters**
Network architecture	Network topology (ad-hoc, infrastructure-based), network area, node density, network connectivity, centralized or distributed, SDN-assisted, GPS availability
Network traffic	QoS parameters (delay, jitter, throughput), priority, traffic type (audio, video, sensor data), reliability requirement, data generation rate, session duration
Routing layer	Multi-path/single-path, multi-cast/uni-cast, hop count, path throughput, latency, stability and lifetime, link quality, route discovery overhead and delay, No. of interfering links
MAC layer	Contention-based vs. contention-free, robustness to packet collisions, fair channel access, link BER, neighbor discovery and tracking, queue management
Physical layer	SNR, modulation scheme, coding methods, link capacity, signal propagation characteristics, channel bandwidth, Doppler effect, radiation pattern
Node property	Speed, mobility model, transmission power, reception sensitivity, queue length, computational power, battery life, antenna type, willingness to act as relay

1.3 Dissertation Contributions and Organization

This dissertation deals with the challenges of providing a robust and reliable communication in autonomous and decentralized UAV networks, where the network topology and traffic conditions change frequently, and the future node trajectories are not known. For this, we have designed distributed, cross-layer routing and MAC protocols, which can discover and select high quality routes and switch to alternate routes in response to the changes in the available communication resources, offered traffic patterns and performance demands to make the best use of the network resources. We show via simulation results that the proposed solutions significantly outperform the existing methods for a variety of different network and traffic settings. The remainder of the dissertation is organized as follows:

Chapter 2 provides a background for the MAC and routing protocols, including their classifications. Then, we discuss the importance of using realistic mobility models for ANs and node trajectory generation, followed by the mathematical framework to compute the LLT.

We discuss a novel, distributed, directional, TDMA (time-division multiple access) MAC scheme in **Chapter 3**, developed in collaboration with Venu Sri Sushma Kuchipudi [33], which uses a low-complexity, rank-based scheduling mechanism. While the use of directional antenna increases the spatial reuse and network capacity, it incurs the well-known hidden terminal and node deafness problems [34–43]. The contention-free TDMA MAC schemes [44] can resolve these problems since they can provide a conflict-free transmission schedule. However, the existing TDMA schemes are not effective in providing real-time conflict-free schedules for users in a decentralized, directional, multihop network with dynamic topology. The proposed directional TDMA scheme adapts to the changes in the network and QoS demands in real-time with significantly lower overhead and delay, and improves the channel utilization and fairness in channel access allocation. The simulation results and comparison

with other recent distributed TDMA-based schemes show that our scheme provides higher throughput with very low control overhead for different network and traffic settings.

In **Chapter 4**, we address the problems of discovering and selecting high quality routes in a dynamic UAV network. Since the traditional topology-based routing schemes are slow in adapting to topology and traffic changes, many proactive routing schemes [7, 9, 14, 17, 18, 45, 46] use multiple metrics to improve the performance. However, they select a route without considering the effect of intra-flow interference, which is induced after the source node starts transmission on the selected route. To resolve the above issues, we present an adaptive, cross-layer, mobility and congestion-aware proactive routing protocol for UAV networks. It includes a novel, multi-step and multi-metric, inter- and intra-flow interference-aware route selection mechanism to select a stable, longer-lasting and less congested route. We also use preemptive route switching and periodic queue management mechanisms. The former prevents potential packet drops due to the congestion and topology changes. The latter prioritizes the transmission of packets which have a lower survivability score and discards the packets which are likely to expire before reaching the destination node. The proposed routing scheme provides a significantly higher data throughput for delay-sensitive traffic flows at different data rates, node densities and speeds, as compared to the standard OLSR (optimized link state routing) and multi-metric OLSR routing protocols.

The proactive routing protocols incur large control and computational overhead and are vulnerable to the security threats as breaching one node can reveal the entire network topology information, including the number of nodes, their relative locations and IP addresses. In contrast, the reactive routing protocols (such as the adhoc on-demand distance vector (AODV) protocol) search a route on-demand, and incur a much lower control and computation overheads. However, the on-demand route discovery incurs overhead and delay, which can increase significantly in a UAV network due to frequent topology changes and link breaks. We propose a hybrid reactive routing protocol in **Chapter 5**, which discovers

the routes on-demand and monitors the region around the selected routes to preemptively switch to another high-quality route within the region. As a result, it significantly reduces the number of route discoveries, and the resulting overhead. At the same time, its control and computational overheads are significantly lower than the proactive routing schemes. Despite having a limited network topology information, the proposed routing scheme provides a superior flow throughput performance at different network and traffic settings.

Chapter 6 presents concluding remarks and potential directions for future work.

Chapter 2

Background

As discussed in Chapter 1, the MAC and routing layer protocols and node mobility model play a crucial role in adaptive and scalable communication. Therefore, in this chapter, we first provide a background for the MAC and routing protocols in Sections 2.1 and 2.2, respectively. Section 2.2 also includes a discussion of three routing protocols we have used to compare the performance of our proposed routing protocols. Then, we discuss the mobility models and link lifetime (LLT) computation in Section 2.3.

2.1 MAC Protocols

MANETs require the nodes to coordinate and collaborate over a wireless channel in order to efficiently complete the assigned task. This requires peer-to-peer communication which is the responsibility of the MAC layer protocols. The MAC layer resides inside the data link layer (i.e., Layer 2 in the OSI model). It resolves the conflict between multiple nodes to access the channel and also performs error correction for anomalies occurring in the physical

Portion of Section 2.3.3 has been presented at the 2021 International Conference on Embedded Systems and Networks (EWSN) [47].

layer. Since data is transmitted in a hop-by-hop fashion along a route, the MAC protocol design affects the QoS of the flow. Therefore, the MAC layer protocol should address the issues caused by the mobility of nodes and an unreliable time varying channel.

2.1.1 Classification of MAC Protocols:

The MAC schemes can be broadly categorized in contention-free and contention-based schemes. In contention-free MAC schemes (e.g., TDMA, FDMA (frequency division multiple access), and CDMA (code division multiple access)), each node is assigned a non-conflicting resource (i.e., time slot in TDMA, frequency in FDMA and code in CDMA) to transmit its data [44]. Since the overhead and delay incurred to compute a non-conflicting schedule for all the nodes is high, these schemes are more suitable for centralized and/or static network topologies, which do not require frequent computation of non-conflicting schedule [48].

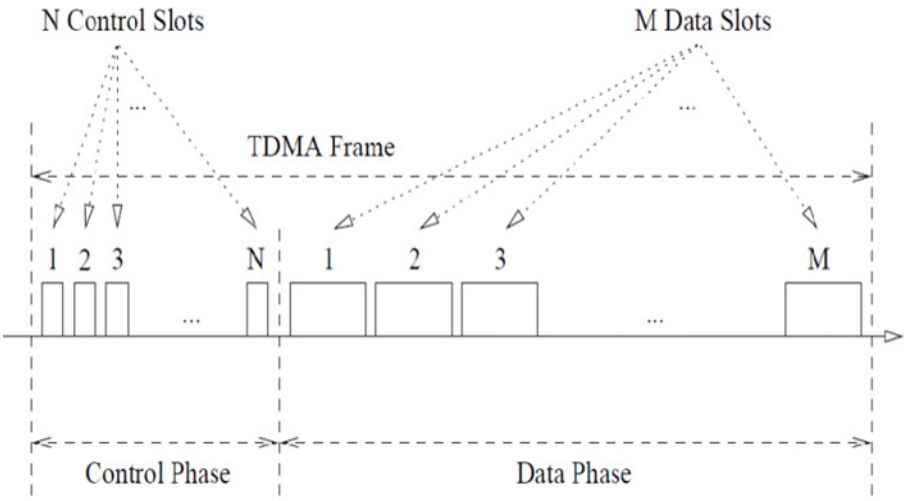


Figure 2.1: Illustration of the TDMA scheme.

In the TDMA protocol, each node in the network is time synchronized. Most TDMA schemes divide time into multiple frames during which the channel access is assigned to different nodes. Each frame is sub-divided into control and data traffic phases, where the phase duration is determined by the number of slots and the slot size (see Fig. 2.1). During the

control phase, transmitter nodes negotiate the data traffic slots with their receiver nodes. Here, both transmitter and receiver nodes agree upon those data traffic slots which do not overlap with the data traffic slots selected by their 1-hop neighbor nodes. As a result, nodes do not interfere with each other during the data traffic phase, which alleviates packet collision. However, control packets can collide during the slot negotiation (i.e., control phase) especially in a distributed and dynamic network, where nodes may not always know the potential transmitters in their 1-hop neighborhoods. This creates uncertainty regarding the optimal frame length, which could provide enough control and data traffic slots for all the transmitter nodes to meet their QoS demands. Most TDMA schemes repeat the data traffic phase multiple times, before starting a new control phase.

On the other hand, contention-based MAC schemes use the 'listen before talk' concept, where a transmitter node first senses the channel and starts its transmission only if the channel is idle. Otherwise, it defers its transmission to prevent the packet collision at its neighboring nodes. These schemes are suitable for decentralized and dynamic network topologies because each node decides its own transmission schedule [48]. The carrier sense multiple access with collision avoidance (CSMA/CA) protocol [44] is one of the most popular contention-based MAC scheme, which waits for a random duration before accessing the channel to reduce the packet collision probability. However, the packets can still collide at a receiver node, when it receives more than one packets simultaneously. Therefore, CSMA/CA uses the request-to-send (RTS) and clear-to-send (CTS) packets, which prevent the 1-hop neighbor nodes of the transmitter and receiver nodes from accessing the channel at the same time. Due to its high effectiveness, majority of wireless devices compatible to IEEE 802.11 based standards use the CSMA/CA MAC scheme.

2.2 Routing Protocols

Routing protocols allow the nodes to communicate with one another via multi-hop relaying in a network. In a bandwidth constrained network with a dynamic topology, a routing protocol should quickly adapt to the link failures and node additions/deletions while minimizing the control overhead. However, the routing protocol design can be challenging as the nodes may not know the complete information of the constantly evolving network topology and traffic [49].

Several routing protocols have been proposed in the literature for different types of wireless networks, such as sensor network, mesh network, MANET, AN, etc. While these protocols may work well for the considered network type, their performance can degrade significantly in other types of networks. For example, the routing protocols designed for ground-based MANET are not suitable for ANs because the airborne nodes fly at a significantly higher speed and their mobility patterns are different, resulting in a highly dynamic network topology with more frequent link disruptions [2–4, 9–11, 14, 15, 17, 45, 50]. Therefore, a number of AN-specific routing protocols have been proposed in the literature, which can be broadly categorized into *position-based* and *topology-based* schemes. The position-based routing schemes (e.g., [1–4, 8, 11, 16, 18, 19, 51]) assume that the recent destination node location can be obtained by querying a central or regional location server. Typically, these location servers are mesh nodes (or sink nodes) forming a wireless mesh network (or sensor network) [52]. They remain connected to each other, and can query and store the node locations within a fixed geographical region without incurring a large overhead and delay, by using some form of location services (e.g., [52]). However, the position-based routing schemes are not suitable for a decentralized network with no central or regional location servers [2–4].

On the other hand, topology-based routing schemes require each node to maintain a routing table for packet transmission, and recompute the routes periodically [1–4]. These schemes can

be categorized into *reactive* and *proactive* routing protocols. The reactive routing schemes (e.g., AODV [53]) discover routes on-demand for a given source-destination pair. While these schemes have a low control overhead, they incur a large route discovery delay because a route to the destination node is searched on-demand [1–4, 9, 15, 17, 18, 20, 45]. The frequent link breaks and degraded route quality in a dynamic AN significantly increase the route rediscovery frequency [1]. Whereas in the proactive routing protocols (e.g., OLSR [54]), each node maintains the updated information for all nodes in the network, a route to a destination node can be immediately found without any route discovery delay. However, periodically exchanging the control messages among the nodes in the network incurs a large signaling overhead [1–4, 7, 18, 19, 46].

Note that position-based, proactive and reactive routing protocols use different strategies to find the route, and therefore, each have distinct advantages over the others. Some routing protocols [55–57] combine the features of different routing categories to further improve the network performance. These schemes are usually referred to as hybrid routing schemes in the literature.

The classical reactive and proactive routing schemes are slow to adapt to the topology changes in a highly dynamic AN, and the ensuing broken routes degrade the flow throughput [1, 15, 19]. Therefore, most existing AN-specific topology-based routing schemes (e.g., [2–4, 9, 14, 17]) focus on selecting a longer-lasting route, and predict link failure (or link stability) to discard broken or unstable routes. However, these schemes assume that UAVs fly using a simple mobility pattern in order to simplify the calculation of RLT [50]. Note that the machine learning driven approaches (e.g., [11, 19, 58]) cannot accurately predict the future node trajectories in autonomous ANs due to the uncertain trajectory changes [2, 3, 50, 51].

Another key issue is that existing routing schemes cannot distinguish whether a packet is lost due to a link break or congestion. A node stores the network topology information at its routing layer, and can quantify the congestion by examining the packets stored in (and

forwarded from) its MAC queue. A few AN-specific routing schemes (e.g., [7, 9, 14, 15, 17, 45, 51, 59]) use this cross-layer information at the time of route selection. However, they do not consider the adverse impact of inter- and intra-flow interference and/or topology changes on the route quality *after* the data transmission starts on a selected route.

In summary, a routing protocol for autonomous UAV networks should have the following characteristics to support the latency-constrained flows: (*i*) Low route discovery overhead and delay, (*ii*) anticipate potential packet drops, identify their cause (link break and/or congestion), and take preventive measures such as route switching and queue management, (*iii*) consider practical node trajectories, and (*iv*) work with decentralized network topologies.

We discuss below the AODV, OLSR and link stability estimation-based preemptive routing (LEPR) protocols. These routing schemes are used to compare the performance of our proposed proactive and hybrid reactive routing schemes in Chapters 4 and 5.

2.2.1 Standard AODV Protocol

The AODV protocol uses four types of control packets: RREQ (route request), RREP (route reply), RERR (route error), and Hello packet. The RREQ messages are broadcast in the entire network, RREP and RERR messages are transmitted using unicast communication, and Hello packets are broadcast in the 1-hop neighborhood [53].

The source node searches a route on-demand by flooding the network with RREQ packets (see Fig. 2.2). The RREQ packet includes the IP addresses of the source and destination nodes, the sequence number of the RREQ originator node (i.e., source node), the last known sequence number of the destination node, and the *Hop Count* field which represents the distance of the current node from the RREQ originator node. When a node receives an RERQ packet, it creates a reverse route to the RREQ originator node in its routing table,

and rebroadcasts the RREQ packet, if it has not done so earlier and does not know a route to the destination node. When the destination node receives the first RREQ packet, it transmits an RREP packet towards the source node. The RREP packet contains the IP addresses of the source (i.e., RREQ originator) and destination (i.e., RREP originator) nodes, the updated sequence number of the destination node and the *Hop Count* field.

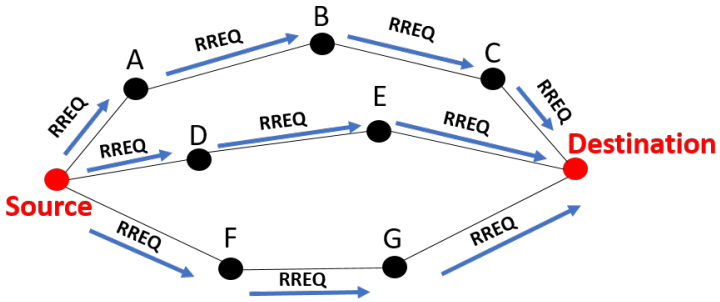


Figure 2.2: RREQ flooding in the standard AODV scheme.

Upon receiving an RREP packet, a node creates a new route entry for the RREP originator node in its routing table, or updates the already existing route entry if the previously known route for the RREP originator node was outdated or had a longer hop length. As a result, the source node selects the shortest hop count (HC) route to the destination node in the standard AODV routing protocol.

Each node broadcasts its Hello packet after the Hello interval (the default value is 1 s). The Hello packet sturcture is similar to the RREP packet, except the IP address of the RREQ originator node. When a node does not receive any control packet from its 1-hop neighbor node for the ALLOWED_HELLO_LOSS $\times$ Hello_Interval duration (the default value is 3 s), it assumes a link break with that 1-hop neighbor node. When an intermediate node of the route detects a link break, it generates an RERR packet for the source node. Upon receiving an RERR packet, the upstream intermediate nodes and source node remove the route entry for the corresponding destination node from their respective routing tables. The source node then starts a new route discovery.

2.2.2 Standard OLSR Protocol

The OLSR protocol [54] is a table-driven proactive routing protocol, where control messages are flooded periodically to maintain the network topology information at each node. As compared to the pure link state routing protocol, its control overhead is much lower because it uses a controlled flooding of the routing messages through the multi-point relay (MPR) nodes. Unlike the reactive routing schemes, it does not require additional control packets to notify link failure, and a shortest hop route is always available to every node in the network.

OLSR uses two types of control packets: Hello and TC (topology control). Each node includes information about its 1-hop neighbors in its Hello packet, which is broadcast periodically after a Hello interval (the default value is 2 s). Hello packets are used to construct the 1- and 2-hop *Neighbor Sets* ($N^1(X)$ and $N^2(X)$, respectively) at node X. Each node then finds the smallest subset (called MPR set) of its $N^1(X)$ nodes required to cover all its $N^2(X)$ nodes, and includes this information in its Hello packet. Each MPR node maintains an MPR selector set to store information of its 1-hop neighbors that have included it in their MPR set.

Each MPR node includes information of its MPR selector nodes in its TC packet, which is broadcast periodically after the TC interval (default value is 5 s). A node uses TC packets to construct its *Topology Set*, which stores information of the links between an MPR node and its MPR selectors. The MPR nodes forward the TC packets coming from their respective MPR selector node(s).

Together, $N^1(X)$, $N^2(X)$ and *Topology Set* represent the node's current knowledge of the network. We call this the *Network Table*. In OLSR, each entry in the *Neighbor* and *Topology Sets* has a default validity duration of 3× the Hello interval and 3× the TC interval, respectively. Upon receiving a new control message, the node resets this validity duration in its respective *Neighbor* and *Topology Sets*. Each node then uses Dijkstra's algorithm on the

graph built using its *Network Table* to find a shortest hop route to the destination node.

2.2.3 LEPR Protocol

Standard AODV routing protocol (discussed in Section 2.2.1) incurs considerable route discovery overhead and delay due to the frequent link breaks in the UAV networks, which degrades flow throughput. The multipath on-demand routing protocols, such as [60, 61], try to reduce the frequent route discoveries by caching multiple routes. When the primary route breaks, these schemes select an alternate route from the cached routes without evaluating their availability, which can degrade the flow throughput. Few other schemes, such as [62,63], prevent the selection of broken routes by periodically monitoring the cached routes, which introduces considerable overhead.

To address the above issues, the LEPR protocol [57] computes multiple stable link-disjoint routes during the route discovery and proactively switches to an alternative route before the primary route breaks. Its two main components are reactive route discovery and semi-proactive route maintenance. Fig. 2.3 shows the different modules of LEPR scheme, which are explained below.

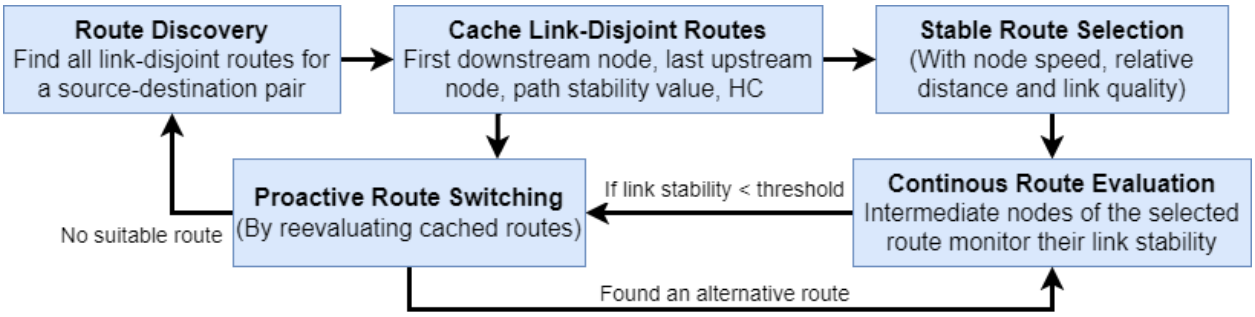


Figure 2.3: Modules used in LEPR scheme.

For route discovery, a source node floods the network with RREQ packets. Upon receiving an RREQ packet, the destination node responds with an RREP packet. To compute the link disjoint routes, the RREQ and RREP packets carry the information of the first hop and last

hop node id on the route, respectively (see Fig. 2.4). Each node computes its link stability metric with each of its 1-hop neighbor node by using the locally available information (their link quality, relative distance and mobility) and updates the *link stability metric field* in the RREQ and RREP packets. The path stability of a route is the minimum of all the link stability metrics along the route. Both the source and destination nodes thus know the link-disjoint routes and their stability.

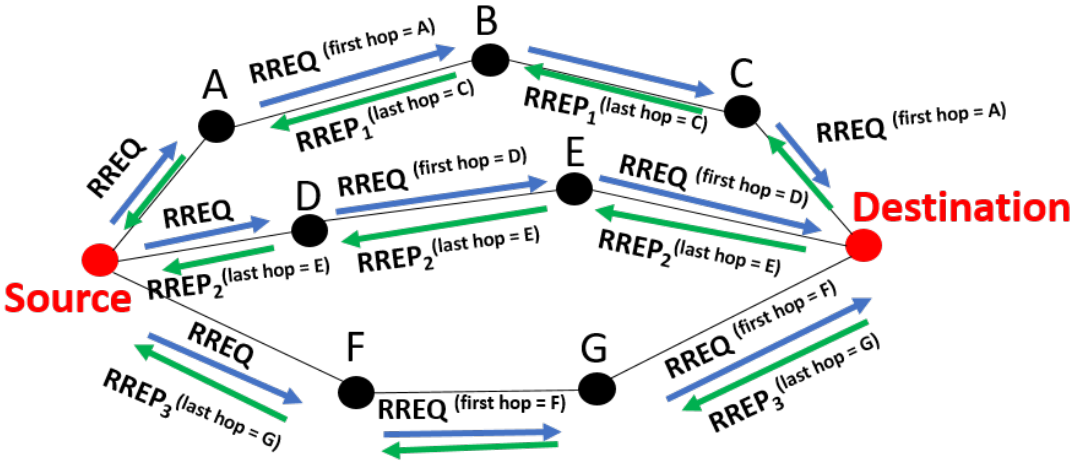


Figure 2.4: RREQ flooding and route reply mechanism in the LEPR scheme for the network topology shown in Fig. 2.2.

The source node then selects a route with the highest path stability value and caches the remaining routes. Intermediate nodes of the selected routes continuously monitor the link stability value with their upstream and downstream nodes on the route. When the link stability value degrades below a threshold k (see Fig. 2.5(a)), the node notifies the destination node by sending a RSWT (route switch) packet. Upon receiving an RSWT packet, the destination node transmits a new RREP packet towards the source node on each cached route (see Fig. 2.5(b)), which carries the updated path stability metric value of the route. The source node switches to any of the cached routes if it's path stability value is greater than a threshold k', otherwise it triggers a new route discovery. As a result, LEPR scheme decreases the total number of route discoveries, which reduces the flow interruptions, control overhead and delay. The value of thresholds k and k' can vary from 0 to 1.

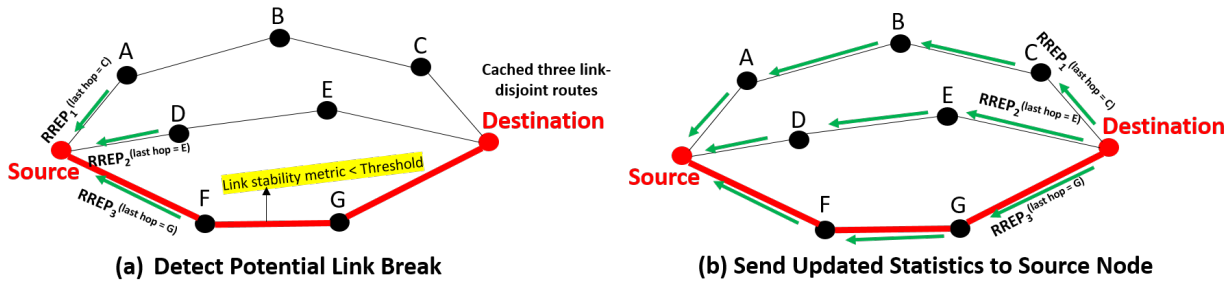


Figure 2.5: Proactive route switching mechanism in LEPR scheme. Here, the primary route is shown in red. The other two routes are cached.

Control and Computational Overhead: In addition to the fields used in the RREQ, RREP, RERR and Hello packets in the standard AODV protocol, LEPR scheme also adds the following information:

- Each node includes its GPS coordinates (uses 6 bytes for (x,y,z) coordinates) in its Hello packet.
- IP address of the first hop and last hop nodes of the route in the RREQ and RREP packets, respectively (4 bytes).
- The RREQ and RREP packets also include the link stability metric value (1 byte).
- RSWT packet includes the IP addresses of the source and destination nodes (8 bytes), sequence number of the destination node (4 bytes), and the link stability metric value (1 byte).

A node in the LEPR scheme periodically recomputes the link stability metric value for each of its 1-hop neighbor, which has a worst time complexity of $O(V)$, where V is the number of nodes in the network.

2.3 Mobility Models

The performance of a routing protocol depends on the node distribution and their connectivity [21, 22]. The node connectivity, in turn, depends on the node trajectory, and thereby, on the underlying mobility model [21,22]. Therefore, the design of routing protocols for ANs should use the mobility models, which can produce realistic node movements [2, 21–23]. The existing mobility models for ANs are summarized below.

Several mobility models, including random walk, random waypoint, and Manhattan grid, have been proposed for mobile networks in the literature [10, 23, 64]. However, they assume node movement in 2D space, and therefore, are not suitable for ANs [10, 21, 23, 64–70]. Further, a fast-moving airborne node follows a smooth trajectory due to its aerodynamic constraints [10,23,67], and cannot abruptly change its direction as its movement depends on its previous locations, speed and heading direction [22,67,71]. Therefore, time-based mobility models are preferred for ANs where the node movement is controlled under mathematical equations [2, 23, 64, 71]. A few time-based mobility model in the literature are the paparazzi model [66], smooth-turn (ST) model [67], semi-random circular movement (SRCM) [68], and Gauss-Markov model [69, 70].

2.3.1 Trajectory Generation

Our proposed routing schemes use the ST mobility model in which each node independently selects a center and radius based on its past trajectory and rotates around the center in the clockwise (or counter-clockwise) direction for a randomly selected duration, called Wait Time [23, 67]. Other time-based mobility models are ill-suited because they generate limited mobility patterns. For example, node movement in the paparazzi model is limited to eight possible trajectories [10, 23]; nodes rotate around a static common center in SRCM model

[10, 23]; and Gauss-Markov model cannot reproduce typical UAV turns [10, 23].

In addition, the above-mentioned mobility models use either a reflection boundary model (in which a UAV makes a sharp 180^0 turn at the boundary) or wrap-around boundary model (which causes sudden node appearances and disappearances) [64]. To address this issue, we use a buffer boundary model [64], which forces a UAV to select its new direction such that it does not go out of the boundary, while complying with its aerodynamics. Possible directions for a UAV to fly are *clockwise*, *counter-clockwise* and *straight*. Note that a UAV does not change its direction from clockwise to counter-clockwise and vice-versa without flying in the straight direction for at least a small duration, in order to maintain its stability [22].

Moreover, the turn radius R of an airborne node depends on its velocity v as, $R = \frac{v}{\omega}$, where ω is the angular velocity. The Wait Time duration for which a node maintains its current trajectory is uniformly selected from the range [$minWaitTime$, $maxWaitTime$], which corresponds to the minimum and maximum duration for which the node stays on its current trajectory.

2.3.2 Use of Link Lifetime in Routing Protocols

Many of the existing routing schemes (e.g., [9,72–76]) compute the link stability using the link and node statistics, such as signal-to-noise (SNR) ratio, received signal strength indicator (RSSI), variance in the node distance, past (LLT) values and the number of acknowledgement packets received during an interval. On the other hand, some other schemes (e.g., [77–79]) propose a mathematical formulation to compute the LLT value by using the characteristics of underlying mobility model. Both approaches are discussed below.

The AN topology evolves with time, which makes the identification of unbroken routes a challenging task [72–74]. A transmitter node retransmits a unicast packet seven times in

the CSMA/CA MAC protocol before recognizing a link break, which reduces the channel utilization and increases the queuing delay for the remaining packets at the node [73]. In addition, frequent topology changes can increase congestion in the network [73], which results in the packet drop due to buffer overflow [72].

To prevent the packet transmission over broken routes, each node in [73,74] includes the GPS locations of itself and its 1-hop neighbor nodes in its control messages in order to create a cartography of the network at each node. Based on these locations, source node selects a route such that links do not break before the reception of new control messages. However, the control packet lengths in [73,74] increases significantly in a dense network, which results in a higher control overhead and packet collision probability. In addition, periodic reconstruction of the cartography increases the computational overhead at each node, which reduces its residual battery life.

The shortest-hop routing schemes generally select the edge nodes. This typically results in receiving packets that have a lower signal strength, which increases the packet loss ratio. To address this issue, the routing scheme in [9] differentiates links based on their RSSI values using Chebyshev inequality and selects stable links with a lower variance in RSSI values. However, an accurate computation of RSSI values is difficult due to high interference from neighbor nodes in a dense network [75].

A mobility-aware route selection scheme is proposed in [72], where the link stability is determined by the variance in the distance values (computed using GPS) of a node pair. Node pairs with a smaller variance in their distance values are expected to remain in the range of each other for a longer duration. However, [72] fails to select links where the UAVs come closer because of the high variance in their monotonically decreasing distance values.

The scheme discussed in [76] uses the distribution of the past LLT and the current link age (i.e., the time duration since the node pair was connected until the current timestamp) to

estimate the residual LLT (i.e., the duration from the current timestamp after which the link between the node pair would expire). However, [76] does not consider the effect of a trajectory change, which can result in an inaccurate LLT computation [80].

Mathematical formulation is proposed in [77–79] to compute LLT for a node pair using their speed, directions of movement and current coordinates. However, these formulations are limited to ground vehicles and cannot be used for the airborne node, where the direction of movement can continuously change.

2.3.3 Accurate LLT Computation

We consider a UAV network, where each UAV includes its trajectory information in its Hello message, which is broadcast periodically to its 1-hop neighbors. Here, trajectory information includes the GPS location, movement state (i.e., clockwise, counter-clockwise or straight), current center, radius, and slope (if UAV is moving in a straight direction). Alternatively, a UAV can compute its trajectory, center, radius and movement state by using its three consecutive GPS locations. Its speed can be computed using the distance travelled (or angular displacement) during a Hello interval, when the UAV goes straight (or travels on a curved trajectory). We also assume that each UAV flies at a unique altitude, to prevent node collisions.

When a UAV pair knows its current location and trajectory details, it can compute the corresponding LLT using the following steps:
Step 1: Find the coordinates of the future location for both UAVs at time t.

Step 2: The link between a UAV pair (U_1,U_2) breaks when their distance d exceeds the node transmission range D. Hence, an equation with one unknown variable t is computed

using,

$$d^{(t)}_{(U_1,U_2)} \geq D \tag{2.1}$$

Step 3: Select the root which best approximates (2.1).

Note that two UAVs establish a link at the *Link Establishment Time*, when they exchange their Hello packets for the first time. The link between the UAV pair terminates, when they move out of each other's transmission range. This time is called *Link Termination Time*. Therefore, *LLT* is the diffrence between *Link Termination Time* and *Link Establishment Time*.

Based on the possible movement states for a UAV (which are clockwise, counter-clockwise and straight), the following three cases are possible for a UAV pair. The *LLT* computation for each case is different as discussed below.

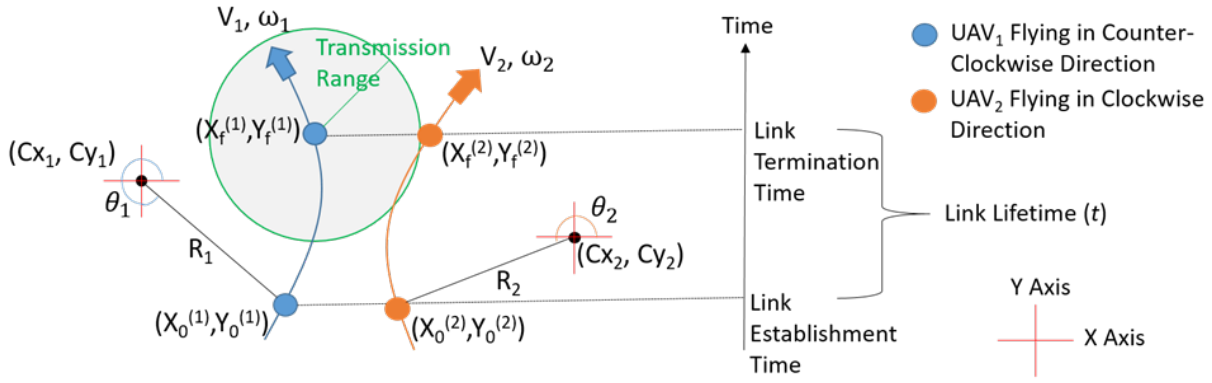


Figure 2.6: *LLT* computation for two curved UAV paths.

Case A. Both UAVs fly in a curve

For a UAV pair in Fig. 2.6 (represented below as U_1 and U_2, respectively), assume their current GPS locations are $(X_0^{(1)},Y_0^{(1)})$ and $(X_0^{(2)},Y_0^{(2)})$, the altitudes are $Z_0^{(1)}$ and $Z_0^{(2)}$, the current trajectory centers are (Cx_1,Cy_1) and (Cx_2,Cy_2), current radii are R_1 and R_2, velocities are V_1 and V_2, and the movement directions are γ_1 and γ_2, respectively. Here, γ is -1 for clockwise direction and +1 for counter-clockwise direction.

The angular velocity for the UAV pair is computed as, $\omega_1 = \frac{V_1 \times \gamma_1}{R_1}$ and $\omega_2 = \frac{V_2 \times \gamma_2}{R_2}$. The initial displacement at the *Link Establishment Time* for each UAV is computed as, $\theta_1 = \tan^{-1}\left(\frac{Y_0^{(1)} - Cy_1}{X_0^{(1)} - Cx_1}\right)$ and $\theta_2 = \tan^{-1}\left(\frac{Y_0^{(2)} - Cy_2}{X_0^{(2)} - Cx_2}\right)$. Hence, the future location coordinates at time t for both UAVs are,

$$\left(X_f^{(1)}, Y_f^{(1)}, Z_f^{(1)}\right) = \left(Cx_1 + R_1\cos(\theta_1 + \omega_1 t),\ Cy_1 + R_1\sin(\theta_1 + \omega_1 t),\ Z_0^{(1)}\right) \text{ for } U_1, \text{ and}$$

$$\left(X_f^{(2)}, Y_f^{(2)}, Z_f^{(2)}\right) = \left(Cx_2 + R_2\cos(\theta_2 + \omega_2 t),\ Cy_2 + R_2\sin(\theta_2 + \omega_2 t),\ Z_0^{(2)}\right) \text{ for } U_2.$$

The link distance $d_{(U_1,U_2)}^{(t)}$ between a UAV pair at time t is

$$d_{(U_1,U_2)}^{(t)} = \left(\left(X_f^{(1)} - X_f^{(2)}\right)^2 + \left(Y_f^{(1)} - Y_f^{(2)}\right)^2 + \left(Z_f^{(1)} - Z_f^{(2)}\right)^2\right)^{\frac{1}{2}} \tag{2.2}$$

Without the loss of generality, we assume the difference in the altitudes of the UAV pair is negligible as compared to their distance in the X and Y axis. Therefore, the term $\left(Z_f^{(1)} - Z_f^{(2)}\right)^2$ is dropped from (2.2) for simplicity. Upon plugging the values of future coordinates of both UAVs, RHS of (2.2) changes to,
$\Big(\big(Cx_1 + R_1\cos(\theta_1 + \omega_1 t) - Cx_2 - R_2\cos(\theta_2 + \omega_2 t)\big)^2 + \big(Cy_1 + R_1\sin(\theta_1 + \omega_1 t) - Cy_2 - R_2\sin(\theta_2 + \omega_2 t)\big)^2\Big)^{\frac{1}{2}}$.

It can be further simplified to,

$$\begin{aligned}\Big(\big(Cx_1 - Cx_2\big)^2 + \big(Cy_1 - Cy_2\big)^2 + R_1^2 + R_2^2 - 2R_1R_2\big(\cos(\theta_1 + \omega_1 t - \theta_2 - \omega_2 t)\big) \\ + 2R_1\big((Cx_1 - Cx_2)\cos(\theta_1 + \omega_1 t) + (Cy_1 - Cy_2)\sin(\theta_1 + \omega_1 t)\big) \\ - 2R_2\big((Cx_1 - Cx_2)\cos(\theta_2 + \omega_2 t) + (Cy_1 - Cy_2)\sin(\theta_2 + \omega_2 t)\big)\Big)^{\frac{1}{2}}\end{aligned} \tag{2.3}$$

The link between a UAV pair breaks when $d_{(U_1,U_2)}^{(t)} \geq D$. If $a = |Cx_1 - Cx_2|$, $\hat{m} = \frac{(Cx_1 - Cx_2)}{a}$, and $b = |Cy_1 - Cy_2|$, $\hat{n} = \frac{(Cy_1 - Cy_2)}{b}$, we can use $\sin(\alpha) = \frac{b}{\sqrt{a^2+b^2}}$, $\cos(\alpha) = \frac{a}{\sqrt{a^2+b^2}}$, and

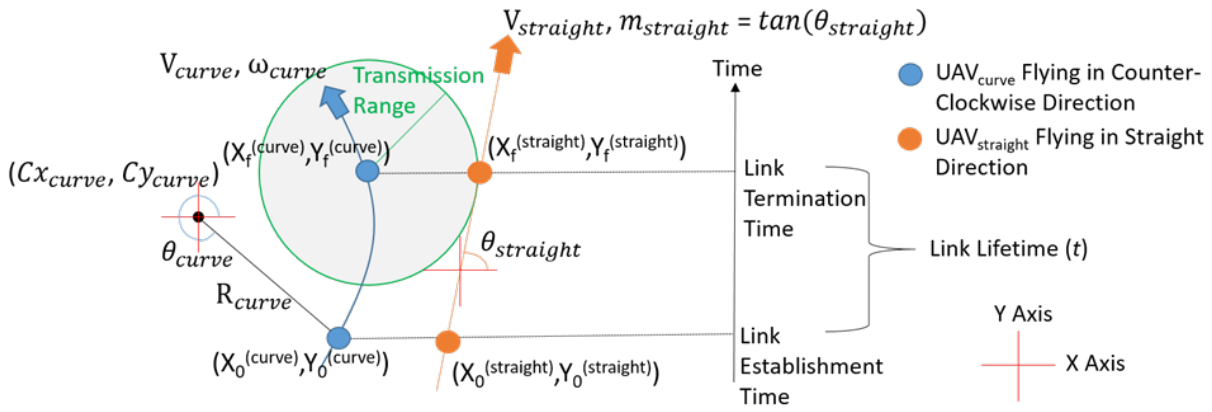


Figure 2.7: *LLT* computation when one UAV flies in a curve while other UAV moves in a straight direction at an angle $\in [0, 2\pi]$ with respect to X-axis.

$\alpha = \cos^{-1}\left(\frac{a}{\sqrt{a^2+b^2}}\right)$ to simplify (2.2) to,

$$\begin{aligned} d^{(t)}_{(U_1,U_2)} &= \big[a^2 + b^2 + R_1^2 + R_2^2 - 2R_1R_2\big(\cos(\theta_1 + \omega_1 t - \theta_2 - \omega_2 t)\big) \\ &+ 2\hat{m}R_1\sqrt{a^2+b^2}\ \cos(\theta_1 + \omega_1 t - (\hat{m}\cdot\hat{n})\alpha) \\ &- 2\hat{m}R_2\sqrt{a^2+b^2}\ \cos(\theta_2 + \omega_2 t - (\hat{m}\cdot\hat{n})\alpha)\big]^{\frac{1}{2}} \end{aligned} \tag{2.4}$$

Here, (2.4) is a polynomial equation with one unknown variable t.

Case B. One UAV flies in a curve and other UAV flies in straight direction at any angle $\in$ [0, 2π] with respect to X-axis

Assume the current GPS locations for the pair U_1 and U_2 (i.e., UAVs moving on straight and curve trajectories, respectively) in Fig. 2.7 are $(X_0^{(s)}, Y_0^{(s)})$ and $(X_0^{(c)}, Y_0^{(c)})$. For the UAV flying in a straight direction, let the speed be V_s and slope is m_s. For the UAV flying in a curve, current trajectory center is (Cx_c, Cy_c), radius is R_c, velocity is V_c, angular velocity is ω_c, initial displacement at the time of link establishment is θ_c, and movement direction is γ_c. Here, γ_c is -1 for clockwise direction and +1 for counter-clockwise direction.

Using the approach discussed for case A, the polynomial equation for this case is,

$$\begin{aligned} d^{(t)}_{(U_1,U_2)} &= \big[(V_s \times t)^2 + a^2 + b^2 + R_c^2 + 2\hat{m}R_c\sqrt{a^2+b^2}\ \big(\cos(\theta_c + \omega_c \times t - (\hat{m}\cdot\hat{n})\alpha)\big) \\ &- 2R_c \times V_s \times t\cos(\theta_c + \omega_c \times t - \tan^{-1}(m_s)) \\ &- 2V_s \times t\cos(\tan^{-1}(m_s) - (\hat{m}\cdot\hat{n})\alpha)\big]^{\frac{1}{2}} \end{aligned} \tag{2.5}$$

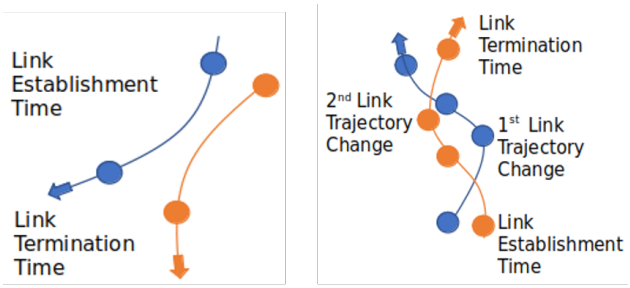
Here, $a = \left|\left(Cx_c - X_0^{(s)}\right)\right|$, $\hat{m} = \frac{\left(Cx_c - X_0^{(s)}\right)}{a}$, $b = \left|\left(Cy_c - Y_0^{(s)}\right)\right|$, $\hat{n} = \frac{\left(Cy_c - Y_0^{(s)}\right)}{b}$, and $\alpha = \cos^{-1}\left(\frac{a}{\sqrt{a^2+b^2}}\right)$.

Case C. When both UAVs fly in a straight direction at random angles $\in$ [0, 2π] with respect to X-axis

For the UAV pair (U_1, U_2), assume the current GPS locations are $(X_0^{(1)}, Y_0^{(1)})$ and $(X_0^{(2)}, Y_0^{(2)})$, slopes are m_1 and m_2, and velocities are V_1 and V_2, respectively. Using the approach discussed in case A, the polynomial equation obtained is,

$$\begin{aligned} d^{(t)}_{(U_1,U_2)} = \Bigg[& t^2\Big(V_1^2 + V_2^2 - 2V_1V_2\cos\big(\tan^{-1}(m_1) - \tan^{-1}(m_2)\big)\Big) \\ & + t\Big(2V_1\big[(X_0^{(1)} - X_0^{(2)})\cos(\tan^{-1}(m_1)) + (Y_0^{(1)} - Y_0^{(2)})\sin(\tan^{-1}(m_1))\big] \\ & - 2V_2\big[(X_0^{(1)} - X_0^{(2)})\cos(\tan^{-1}(m_2)) + (Y_0^{(1)} - Y_0^{(2)})\sin(\tan^{-1}(m_2))\big]\Big) \\ & + \big(X_0^{(1)} - X_0^{(2)}\big)^2 + \big(Y_0^{(1)} - Y_0^{(2)}\big)^2\Bigg]^{\frac{1}{2}} \end{aligned} \tag{2.6}$$

Note that we expand the trigonometric functions in cases A and B up to 12 steps using Taylor's expansion to get their close approximation. Then, we compute the roots of the equation using numerical method and select a root that approximates the polynomial equation the best. We observed a very low approximation error in *LLT* computation if the UAV trajectories do not change before the *Link Termination Time* (see Fig. 2.8(a) for example). Note that this approach can also be extended for elliptical UAV trajectory by using the



(a) (b)

Figure 2.8: Link breakage. The link between a UAV pair can break (a) without any trajectory change (see left image) or (b) after multiple Link Trajectory Changes (see right image).

minor and major axis of the elliptical trajectory.

(A) Recomputing *LLT* when UAV Trajectory Changes

During the lifetime of a link, any UAV of a given UAV pair can change its trajectory. For example, both UAVs in Fig. 2.8(b) change their respective trajectory once before the link termination (see 1^{st} and 2^{nd} Link Trajectory Change in Fig. 2.8(b)). Since the trajectory change of a UAV may not be known a priori in an autonomous UAV network, the LLT value should be reevaluated when either of the UAVs changes its trajectory before the link breaks. Therefore, a UAV pair calculates its LLT when the link is first established and then updates the value whenever the UAV trajectory changes. This allows an accurate computation of LLT value regardless of the past changes in the node trajectory.

Then, a node can select the longest-lasting route R^* using the LLT values of the links in the network as,

$$R^* = \underset{Route\ R}{\arg\max} \left(\min_{Link\ l \in L_R} \left(LLT_l \right) \right) \tag{2.7}$$

where L_R represents a set of links on route R.

Chapter 3

A Distributed, Directional TDMA MAC Protocol

The use of directional antennas in wireless communication reduces the co-channel interference and extends the coverage range, thereby increasing the spatial reuse and network capacity. To improve the performance of a multi-hop network, the routing protocol should optimally use the directional beam(s) of relay nodes along the route. This requires designing a MAC protocol, which can provide reliable communication between a node pair equipped with directional antenna. In this chapter, we describe a novel directional TDMA MAC scheme for distributed, multi-hop wireless networks.

3.1 Introduction

Nodes equipped with single-beam directional antennas (SBA) are widely-used in disaster response networks, airborne networks, military networks and sensor networks [34–36, 81].

Chapter 3 is based on the materials published in the journal paper [33], and represents work done in collaboration with Venu Sri Sushma Kuchipudi.

Since the coverage area of a directional node is limited by its beamwidth (θ), it must beamform in the direction of the transmitter to successfully receive the packets. Since a receiver node in CSMA-based MAC scheme remains unaware of when its transmitter (or neighbor) node will start the communication, it fails to beamform in the direction of transmitter and therefore misses the packet(s) which aggravates the deafness problem [34–43]. To address this issue, some CSMA-based schemes [41, 43, 82] use two antenna system, where omnidirectional antenna overhears transmissions of neighbor nodes and SBA is used for data packet transmission. However, the use of an omnidirectional antenna reduces the benefits of spatial reuse and leads to the well-known gain-asymmetry problem [34, 83, 84]. In some other schemes [38, 85, 86], a node steers its antenna at least ($360^0/\theta$) times to scan and notify its entire neighborhood, which introduces significant sweeping delay [32]. Moreover, none of the above-mentioned CSMA-based schemes completely resolve the issues of deafness, hidden terminal, and capture effect [37, 39, 40, 42, 83, 87]. Therefore, they are not suitable for directional communication.

The TDMA schemes are widely used for directional communication since they can provide a conflict-free transmission schedule and avoid deafness and capture effect [37, 39, 40, 42, 43, 82–84, 87, 88]. TDMA-based schemes also offer a better QoS support than random access-based schemes since they reserve a guaranteed period of time for each node to access the channel [88, 89]. However, TDMA-based schemes can introduce large overhead and delay and are unable to adapt to topology changes in a multi-hop network in real-time [90], as discussed below.

3.1.1 Review of Directional TDMA Schemes

Most directional TDMA-based MAC schemes (such as [83, 84, 87, 91]) divide a frame into three phases: neighbor discovery, reservation, and data traffic. During the neighbor discovery

phase, nodes search for their neighbors and agree upon a reservation slot in which they negotiate data traffic slots. These schemes perform neighbor discovery only at the start of each frame, and repeat the other two phases until the end of the frame. In a multi-hop network, a node may be required to serve flows on multiple routes. Combining the slot selection for the reservation period with neighbor discovery limits the node's ability to serve these flows on multiple routes in the same frame. Further, since the scheduling of traffic slots depends on the order of reservation slots, nodes with higher reservation-slot indices may not get enough traffic slots to transmit their packets, which can degrade their throughput and impact the fairness of the network [37, 91].

Schemes in [39,40] use a master-slave like approach in which nodes transmit control packets in the contention-period (random access phase) to compete for the conflict-free data traffic slots. The successful nodes become master nodes which then control the communication of the unsuccessful slave nodes in the conflict-free period. As mentioned before, the random access approaches suffer from the deafness and collision problems [88], which make these schemes unsuitable for multihop topologies, where the number of hidden nodes (hence, deafness problem) increase significantly. Moreover, the above-mentioned TDMA schemes (i.e., [39, 40, 83, 84, 87, 91]) do not provide QoS support.

To obtain the QoS-aware conflict-free schedules, many TDMA-based schemes (both centralized [92, 93] and distributed [37, 89, 90, 94, 95]) use the *graph coloring* techniques, which require each node to transmit its data traffic demand and each neighbor node to receive (or overhear) that packet. In centralized schemes, the information gathered at the central node (which can also act as a single point of failure) easily becomes obsolete when the topology changes, whereas a large *notification overhead*[1] and delay incur in distributed schemes since each node periodically retransmits its local schedule until a globally converged (or a feasible)

[1]Each directional node needs to transmit its data traffic slot schedule to its 1-hop neighbor nodes so that they can detect conflict in their schedules with this node [32]. This process is repeated until the conflict at each node is resolved [88], which introduces a large notification overhead and delay.

schedule is obtained [88]. Therefore, none of them provide a *real-time* solution (i.e., compute a conflict-free schedule instantly or with very low delay), when the network topology and data rate, routing table, flow priority, etc., change dynamically [88].

To address the *notification overhead* problem, schemes in [6,96] calculate the *rank matrix* at each node in the single-hop network, by using the node ids in the hash function. Based on this rank matrix, each node computes a conflict-free schedule for the reservation period in real-time. However, [6,96] are not suitable for a multi-hop topology because the neighboring nodes can generate contradicting rank matrices, when their 1-hop neighbors are different [88]. To resolve this issue and avoid conflict, each node in [97] informs its 1-hop neighbors about its selected reservation slots, which introduces significant overhead and leads to wastage of slots. Further, the length of reservation as well as other control periods of an n-node network in these rank-based MAC schemes is of the order of $O(n)$, which results in a large control overhead [88].

The above-mentioned directional TDMA schemes cannot adapt in real-time to the variations in link rates and topology changes as new flows (or nodes) are admitted or revoked. They also introduce a large overhead and delay, which increase with network size, making them unsuitable for multihop topology. Moreover, these schemes incur a large overhead for supporting the QoS requirements.

3.1.2 Contributions of our Proposed Scheme

To the best of our knowledge, no real-time, distributed, directional TDMA scheme exists in literature, which can provide a conflict-free schedule for a dynamic, multi-hop network. In this chapter, we propose a novel, distributed, pure directional TDMA MAC scheme for multi-hop networks, which adapts to the topology changes and/or flow requirements in real-time, and facilitates QoS-aware communication with no notification overhead. Here, pure direc-

tional means nodes do not use omnidirectional antenna. Like many other TDMA schemes (such as [39, 42, 94, 98–100]), our proposed TDMA MAC scheme assumes the knowledge of 2-hop neighborhood which can be obtained using a gossip-based neighbor discovery scheme, such as [101, 102]. Note that, neighbor discovery is an essential part of directional MAC schemes since the node cannot transmit its packets omnidirectionally [101, 102].

The **main contributions** of our scheme are as follows:

1. A Low-Complexity Rank-based Scheme for Multi-hop Topology: As discussed in Section 3.1.1, the rank-based scheduling schemes are not suitable for a multi-hop network as the contradicting ranks can be generated for nodes. In our scheme, each node divides its 1-hop neighborhood in *fully connected 1-hop neighborhoods*, where every node is in the 1-hop transmission range of all other nodes. It then independently generates a rank matrix for each of its *fully connected 1-hop neighborhood*. It refers to one of these rank matrices in each slot of the Hello period (which is partially analogous to the reservation period mentioned above), and chooses an action (transmits or listens to a neighbor).

2. Real-time and Fully-Distributed Scheduling Scheme: Since every node independently generates its rank matrix for each of its *fully connected 1-hop neighborhood*, our proposed rank-based scheme is fully-distributed. These rank matrices are non-contradicting (see Section 3.2 for details); therefore, the nodes are not required to notify their rank matrices to their neighbors to resolve the scheduling conflict. As a result, a scheduling solution is obtained in real-time as the *notification overhead* and the resulting delay are completely eliminated[2]. Each node in our scheme can easily detect changes in the network topology, by using its updated 2-hop neighborhood information (obtained via neighbor discovery mechanism) and recalculate its *fully connected 1-hop neighborhood(s)* in real-time, which makes

[2]In our scheme, nodes broadcast their 1-hop neighborhood information during neighbor discovery so that fully connected 1-hop neighborhoods can be constructed at each node. Note that every directional TDMA scheme available in literature employs the neighbor discovery for multi-hop and/or mobile topology. Since nodes, in our scheme, do not broadcast their rank matrices and schedule, the notification overhead is completely prevented.

our scheme suitable for a dynamic network topology.

3. Low Control Overhead: Unlike traditional rank-based schemes, (e.g., [6, 96]), where the length of reservation and other control periods of an n-node network is of the order of *O(n)* [88], our proposed scheme depends only on the number of nodes in a *fully connected 1-hop neighborhood*. Therefore, the rank matrix computed in our scheme is smaller and requires fewer reservation and other control slots, which significantly reduces the control overhead and delay.

4. Real-time Adaptation to Dynamic QoS Requirements: With the flexibility to choose from multiple rank matrices, a node can select the link (i.e., next-hop node) it wants to serve in the current frame based on the QoS-metric value of the packets stored in its buffer. This allows a node to easily *adapt to the dynamic QoS requirements in real-time*.

5. Improved Slot Utilization: We use a new (but optional) REQ (*requisition*) period to notify the intended receiver about the updated slot requirement. This period reduces slot wastage and increases channel utilization, and thereby, network throughput at hotspot[3] nodes.

6. A Fair Slot Allocation Mechanism: We use a *throughput scaling* mechanism, which increases fairness by accommodating all traffic requests regardless of the order of their arrival, and therefore, helps in congestion management. Note that this is a major drawback in many distributed schemes, such as [83, 84, 87].

7. Improved Spatial Reuse: We use an optional *piggyback reservation* mechanism, which further increases the spatial reuse of our scheme by enabling an intermediate transmitter (or receiver) node to accommodate multiple flows in a frame.

Chapter Organization: This chapter is organized in six sections. We discuss our dis-

[3]A hotspot node serves more than one flow.

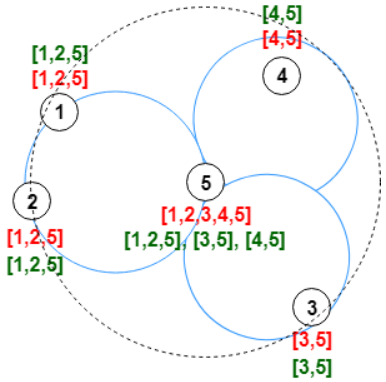


Figure 3.1: Illustration of a 2-hop network topology. For nodes marked 1 to 5, the dotted black circle represents the communication range of node 5, which includes all its 1-hop neighbors. The blue circles represent three fully connected 1-hop neighborhoods of node 5. The rank matrix obtained at each node with the traditional rank-based scheduling scheme, such as $[6, 96]$, is shown in red, whereas the rank matrices in green are obtained using our proposed scheme.

tributed rank-based scheduling scheme for multi-hop network topology in Section 3.2, followed by different control periods and mechanisms used in our scheme in Section 3.3. Then, we discuss the working principle of our proposed scheme in Section 3.4, simulation results and comparison analysis in Section 3.5, and conclusions in Section 3.6.

3.2 A Distributed Rank-based Scheduling Scheme for Multi-hop Topology

In rank-based schemes, every node independently calculates the hash value (i.e., rank) for each of its 1-hop neighbor nodes [88]. These schemes then allocate the reservation slots to the nodes in the descending order of their ranks. They provide a solution for 1-hop network topologies, where every node has a direct link to all other nodes, but fail in a multi-hop topology, where some nodes have different 1-hop neighbor nodes because a node can have a different rank at each of its 1-hop neighbor nodes [88]. For example, Fig. 3.1 shows a 5-node, 2-hop network where node 5 has all the four nodes (nodes 1, 2, 3, and 4) in its 1-hop neighborhood, whereas nodes 3 and 4 are not in the 1-hop neighborhood of nodes 1

and 2. Here, nodes 3 and 4 have only node 5 in their respective 1-hop neighborhood. The rank matrix obtained at each node by using any rank-based scheduling scheme available in literature (e.g., [6, 96]) is shown in red color in Fig. 3.1. Here, node id is used as the hash value for simplicity, where a smaller value represents a higher rank. Every node transmits its packet in a unique time slot based on its rank. For example, as per the rank matrix at node 2 ([1,2,5]), the nodes 1, 2, and 5 transmit in slots 1^{st}, 2^{nd}, and 3^{rd}, in that order. Here, node 2 receives a packet from node 5 in its 3^{rd} slot. However, according to the rank matrix at node 5 ([1,2,3,4,5]), it transmits to node 2 in its 5^{th} slot. As a result, nodes 2 and 5 do not steer their beams towards each other simultaneously and therefore cannot communicate with each other.

For such random and/or multi-hop topology, the existing rank-based schemes (e.g., [6, 96]) require local convergence to obtain conflict-free rank matrices at neighboring nodes. For example, after local convergence, the rank matrices at nodes 1 to 5 in Fig. 3.1 would be [1,2,3,4,5], which allow node 2 to receive a packet from node 5 in the 5^{th} slot. However, the local convergence in a multi-hop and mobile network topology would introduce an overhead and delay as each node must repeatedly notify its 1-hop neighbors about its conflicting slots until the conflict is resolved. Furthermore, the number of slots used during the reservation period are of the order of *O(n)* in a network of n nodes, which results in a larger control period [88].

Our distributed rank-based scheduling scheme for multi-hop topology is discussed below.

3.2.1 Construction of Fully Connected 1-hop Neighborhoods

To address this issue, each node in our scheme divides its 1-hop neighborhood into *fully connected 1-hop neighborhoods*. Using the available 2-hop neighbor information, each node identifies the nodes in each of its *fully connected 1-hop neighborhoods*, where every neighbor

node forms a direct link with all other nodes. For example, since node 5 in Fig. 3.1 has 1-hop neighbor information of nodes 1, 2, 3, and 4, it forms three *fully connected 1-hop neighborhoods* of rank matrices ([1,2,5], [3,5], and [4,5]) to cover its entire 1-hop neighborhood (shown in green color in Fig. 3.1). If node 5 wants to transmit to node 2, it knows from its 1^{st} rank matrix ([1,2,5]) that node 2 will steer its beam toward node 5 in the 3^{rd} slot, and hence, resolves the conflict between nodes 2 and 5. Moreover, using its rank matrices, node 5 chooses between nodes 1, 3, and 4 in the 1^{st} slot based on the available route information and/or flow or link priorities (see Section 3.3.1 for slot selection process). Similarly, it decides whether to listen to node 2 or transmit to node 3 or 4 in the 2^{nd} slot. The pseudocode to obtain fully connected 1-hop neighborhoods is given in Algorithm 1.

Algorithm 1: Pseudocode for construction of fully connected 1-hop neighborhoods

1 **Input:** Node x and its 1-hop neighborhood $N^1(x)$
2 Variable FC1HN stores the fully connected 1-hop neighborhoods of node x
3 ComputeNeighborhood ($N^1(x)$, x, FC1HN) *//It computes fully connected neighborhoods*
4 Remove subsets from FC1HN *//For example, if FC1HN contains [1,2,3,4] and [1,2,3], remove [1,2,3]*
5 **Output:** FC1HN contains length(FC1HN) unique fully connected 1-hop neighborhood(s)

3.2.2 Computing Node's Rank Matrix

For each of its fully connected 1-hop neighborhoods, a node constructs its rank matrix as follows:

Step 1: Calculate the hash value for node i of the j^{th} fully connected 1-hop neighborhood (FC1HN) of node x as,

$$\begin{aligned} &Rank(i) = MD5(node\ id(i)),\ where, \\ &i \in FC1HN_j(x)\ and\ \cup_j FC1HN_j = N^1(x). \end{aligned} \tag{3.1}$$

Here, $N^1(x)$ is the set of 1-hop neighborhoods of node x.

Step 2: Arrange the nodes of a fully connected 1-hop neighborhood in their decreasing order of ranks, Rank(i).

Note that the schemes in [6, 96] include the timestamp as a random seed in their hash function so that the transmitter node with a higher reservation-slot index (i.e., low rank) can get a fair chance to schedule its data traffic slots. Instead, each node in our scheme uses the *throughput scaling* mechanism (discussed in Section 3.3.2), which allows a receiver node to fairly distribute its data traffic slots among all of its transmitter nodes regardless of their ranks (i.e., reservation-slot indices).

The rank matrix in traditional rank-based schemes depends on the network size (n) (i.e., order of *O(n)* [88]), because each transmitter node needs unique slots for the reservation and other control periods. However, the size of rank matrix in our proposed scheme depends only on the number of nodes in a *fully connected 1-hop neighborhood*. Therefore, the rank matrix computed in our scheme is much smaller (see an example below) and requires fewer reservation and other control period slots, which significantly reduces the protocol overhead and delay for a multi-hop network.

Example: The rank matrix lengths in our scheme for varying network sizes and two different network topologies are shown in Fig. 3.2. Here, nodes are randomly placed and each experiment is repeated 10 times. Note that our scheme requires only 13 and 19 slots for a 250-node network in Fig. 3.2a and 3.2b, respectively, which is significantly less than $O(250)$ slots required in traditional rank-based schemes [88].

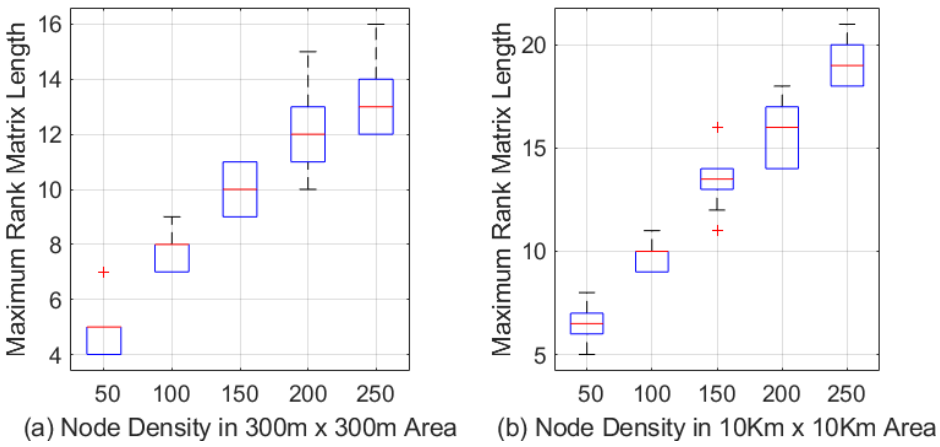


Figure 3.2: Maximum rank matrix length (k) in our proposed scheme at varying network sizes in (i) 300 m x 300 m and (ii) 10 km x 10 km network area. Note that each control period in our scheme has k slots.

3.3 Different Control Periods and Mechanisms in our Proposed TDMA Scheme

In our proposed TDMA scheme shown in Fig. 3.3, a frame is divided into three to five periods Hello, REQ (Requisition), reservation, piggyback reservation (PR), and data traffic. The first four periods use control packets to schedule conflict-free traffic slots, whereas data is transmitted during the data traffic period. Note that only two control periods (i.e., Hello and Reservation) are mandatory in our scheme, similar to the existing TDMA schemes,

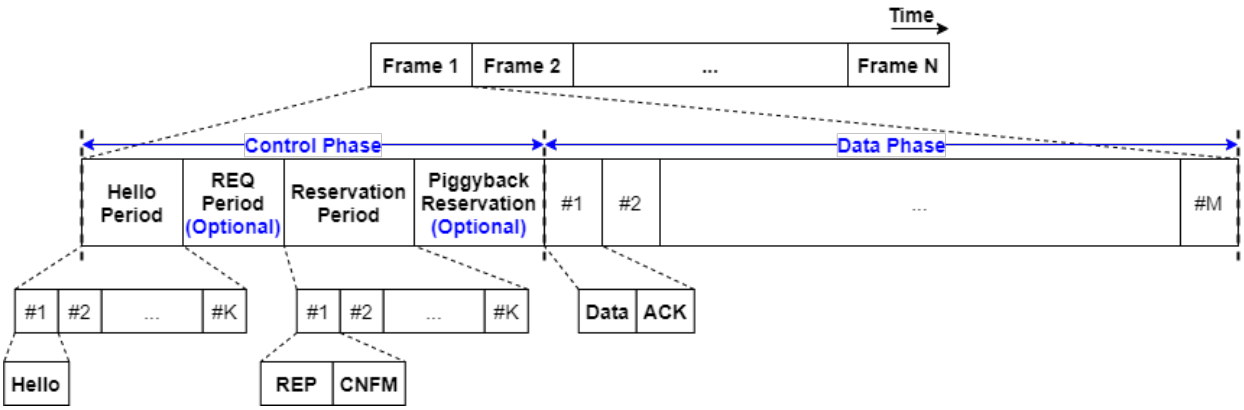


Figure 3.3: The frame structure used in our proposed TDMA scheme. Note: Hello, REQ, and both Reservation periods use K slots each, where K is the maximum number of nodes present in a *fully connected 1-hop neighborhood*. Here, the REQ and PR periods are optional as discussed in Section 3.3.

whereas the remaining two control periods (i.e., REQ and PR) are optional. Our scheme introduces a new REQ period to reduce the slot wastage and increase link throughput. Using the PR period increases spatial reuse at a node. Using a *throughput scaling* mechanism after the Hello and REQ periods increases fairness and introduces the adaptive slot scheduling capability, which helps in congestion management. These periods along with their packet structure are described below.

3.3.1 Hello Period

The Hello period reserves a conflict-free handshake slot in which the transmitter and receiver pair can negotiate and schedule data traffic slots. During this period, only the transmitter node sends Hello packet which includes the number of its data packets (called *desired throughput*) to be sent in the current frame. The Hello packet structure (shown in Fig. 3.4) includes the node id (i.e., MAC address) of the transmitter and receiver nodes, beam id of transmitter node, timestamp to store this packet's origination time, slot index in which this Hello packet is transmitted, and whether the transmitter node wants to request the use of the PR period in the next frame to negotiate data traffic slots. Note that the use of *Timestamp* and *Hello slot index* fields help in node synchronization.

During the Hello period, each node decides its action (whether to transmit or listen) in each Hello slot, based on the available route information and QoS-metric value, as discussed below. In a multi-hop topology, several complex situations can arise, which are described below with the help of an example topology shown in Fig. 3.1.

Tx node MAC address (48 bits)	Tx node beam Id (3 bits)	Rx node MAC address (48 bits)	Time stamp (32 bits)	Hello slot index (5 bits)	Desired throughput (7 bits)	Enable redundant slot? (1 bit)

Figure 3.4: 18 byte Hello packet structure. Note: the REQ packet is similar to the Hello packet except that it does not contain the *Enable Redundant Slot?* field.

(A) Computing the QoS Metric:

Similar to the cross-layer schemes presented in [93, 98–100], our scheme assumes that each node has the route information, such as previous and next hops of the route, remaining hop count and flow priority for the routes passing through it, which both proactive and reactive routing schemes can provide.

If a node has packets for multiple receiver nodes in its buffer, it selects the receiver node based on a QoS-metric which is explained below. In our proposed scheme, the node calculates the packet urgency U for a packet i in its queue as:

$$U_i = \frac{P_i \times H_i}{(TTE)_i} \tag{3.2}$$

Here, P represents the priority value of a flow that the packet i belongs to. H represents the number of remaining hops the packet needs to traverse on a given route in order to reach the destination. TTE is the time to expiry which represents the remaining packet lifetime based on its time-to-live (TTL). The QoS metric allows a node to prioritize the packets, which belong to a high priority flow, are relatively far from the destination, and have a small TTE, so that they can reach the destination before their expiry.

All the packets in the node queue are arranged in the decreasing order of their urgency value. Once a packet with the highest urgency value is selected, the node checks its receiver node id, and transmits the Hello packet to that receiver node during the corresponding Hello slot.

(B) Possible Situations in Each Hello Slot:

The following three situations are possible: **(a) When a node X forwards data packets to multiple receiver nodes in different fully connected 1-hop neighborhoods:** Consider the rank matrices of node 5 in Fig. 3.1 ([1,2,5], [3,5], [4,5]), where node 5 has data

packets for nodes 3 and 4 which are in two different *fully connected 1-hop neighborhoods*. Since node 5 has the same rank in both of these rank matrices, it can transmit a Hello packet to either node 3 or node 4 in the 2^{nd} Hello slot in a frame. Here, node 5 uses the QoS metric to select between nodes 3 and 4 and sends a Hello packet accordingly. On the other hand, when a node has different ranks in its *fully connected 1-hop neighborhoods*, it uses each of the Hello slot (corresponding to its rank) to transmit the Hello packet towards the corresponding receiver node. For example, if node 5 has data packets for nodes 1 and 3, it transmits Hello packets towards nodes 3 and 1 in the 2^{nd} and 3^{rd} Hello slots, respectively, based on its rank matrices.

(b) When a node X forwards data packets to multiple receiver nodes in the same fully connected 1-hop neighborhood: Consider the rank matrix [1,2,5] of node 5 in Fig. 3.1. Here, nodes 1 and 2 are in the same *fully connected 1-hop neighborhood* of node 5. Hence, if node 5 has data packets for both nodes 1 and 2, it can transmit a Hello packet to either of these nodes in the 3^{rd} Hello slot.

(c) When a node X does not have data packets to forward and chooses to listen: If node X is not a part of any route, it stays idle. However, if it belongs to multiple routes, it listens to that node for which it is the next hop. If there are more than one such nodes, node X listens to the node it did not listen to in previous frame(s).

3.3.2 Throughput Scaling

During the Hello period, a node can receive/transmit multiple Hello packets, each with varying *desired throughput* T requirement. The node cannot serve all the requests if the sum of all *desired throughput* requests exceeds the number of available data traffic slots D. Traditional TDMA-based schemes schedule the slots in the first-come first-served (FCFS) order which can not only result in increased queuing delay but also starve one or more nodes.

To address this, each node, in our scheme, scales down the individual data requirement as:

$$T_i^s = \left\lfloor \frac{T_i \times D}{\sum_{j=1}^{k} T_j} \right\rfloor \tag{3.3}$$

Here, T_i^s is the *scaled desired throughput* of i^{th} request, k is the total number of Hello packets transmitted and received by a node in the Hello period, and i corresponds to the i^{th} Hello packet, such that $i \in [1, k]$. The denominator is the sum of data traffic slots requested in the individual (i.e., j^{th}) Hello packet.

In Fig. 3.1, node 1 (rank matrix [1,2,5]) transmits a Hello packet to node 2 in the 1^{st} Hello slot with a request of T_1 data traffic slots and also receives a Hello packet from node 5 in the 3^{rd} Hello slot with a request of T_2 data traffic slots. At the end of the Hello period, node 1 has the total desired throughput of $T_1 + T_2$ slots. If node 1 has D_1 data slots available and if $T_1 + T_2$ is greater than D_1, then the accepted data slots (at node 1) for nodes 2 and 5 are $T_1 \times D_1/(T_1 + T_2)$ and $T_5 \times D_1/(T_1 + T_2)$, respectively. Hence, *throughput scaling* allows a node to serve multiple links (and, hence, routes) within a frame by fairly distributing the data traffic slots among all requests, and helps in congestion management by reducing queue overflow.

3.3.3 REQ Period

After *throughput scaling*, the *desired throughput* of a transmitter node can decrease. If the transmitter node fails to tell its updated requirement to its receiver, the receiver node would waste its data traffic slots. Hence, we use a separate REQ period after the Hello period which has the same number of slots as the Hello period. A node transmits an REQ packet towards its receiver only when it has an updated *desired throughput*. Note that the REQ and Hello slot indices are the same. In the example given in Section 3.3.2, node 1 transmits REQ packet to node 2 in its 1^{st} REQ slot with T_1^s *desired throughput*. The structure of REQ

packet is identical to the Hello packet except that it does not have the *Enabled Redundant Slot?* field (see Fig. 3.4).

3.3.4 Reservation Period

We use the reservation period to schedule conflict-free data traffic slots in the data traffic period. Here, each slot is divided into two sub-slots, namely REP (reply) and CNFM (confirm). The receiver nodes reply to the received Hello packet, in the first sub-slot, with an REP packet (shown in Fig. 3.5), which includes the transmitter and receiver node id (i.e., MAC address), beam id of the transmitter node, packet generation time, scaled throughput, and available data traffic slots at the receiver. Since the number of data traffic slots assumed in our scheme is 100, the *Traffic Slots* field in the REP packet is of 100 bits where each bit corresponds to a data traffic slot. We use the other two fields *Enable Redundant Slot?* and *Redundant Slot Request Accepted?* to enable the PR period as discussed in Section 3.3.5.

In the second sub-slot, the transmitter node replies with a CNFM packet which has a packet structure similar to REP, except that it does not contain the *Enable Redundant Slot?* field. The CNFM packet includes the accepted data traffic slots (i.e., corresponding bit in the *Traffic Slots* field is set to 1). Transmitting the REP and CNFM packets reduce interference at the receiver node by preventing the receiver node's 1-hop neighbors, which are exposed to the communication of this transmitter-receiver pair, from scheduling the same data traffic slots. If a transmitter node does not receive a response (i.e., REP packet) from its receiver during the reservation period, it can retransmit its Hello packet in the next frame.

Tx node MAC address (48 bits)	Tx node beam Id (3 bits)	Rx node MAC address (48 bits)	Time stamp (32 bits)	Scaled desired throughput (7 bits)	Traffic slots (100 bits)	Enable redundant slot? (1 bit)	Redundant slot request accepted? (1 bit)

Figure 3.5: 30 byte REP packet structure. The CNFM packet has the same structure, except the *Enable Redundant Slot?* field.

3.3.5 Piggyback Reservation Period

Unlike the reservation period where a receiver node responds to the transmitter node, the piggyback reservation (PR) period represents the receiver- or transmitter-initiated communication. As discussed in Section 3.3.1(B), when a node X has the same rank in its two or more rank matrices, it can select only one of them in a frame. If this node is unable to schedule all its data traffic slots with the selected node, it can schedule its unscheduled data traffic slots for communication with another node on a different link by using the piggyback reservation. Similar to the reservation period, each slot in the PR period is divided into REP and CNFM. During the PR period, a node X uses the REP packet to notify its unscheduled data traffic slots to the nodes on its other link(s).

However, since nodes use a directional antenna, the intended receiver of this REP packet must know when to steer in the direction of node X in order to receive the REP packet and schedule the data traffic slots. This is enabled by using the *Enabled Redundant Slot?* field in the Hello packet where the transmitter node requests the receiver node to use the PR period in the following frame so that it can transmit its Hello packet on a different link in the next frame, and thereby, accommodate both flows. Note that the receiver node can also request to use the PR period by setting the *Enabled Redundant Slot?* to 1 in its REP packet while responding to the transmitter in the reservation period of the current frame. In both cases, the other node must agree on using the PR period in the next frame by setting its *Redundant Slot Request Accepted?* field to 1 in the current frame. Thus, the receiver and transmitter nodes give their consent on using the PR period via their REP or CNFM packet, respectively.

For using the PR period, the transmitter and receiver nodes store the reservation slot number of the current frame. In the next frame, the receiver node transmits the REP packet with its unscheduled data traffic slots and 0 value for the *Scaled Desired Throughput* towards the

intended transmitter, in the corresponding slot of the PR period. The transmitter node then selects the common available data traffic slots, updates corresponding bits in the *Traffic Slots* field, and transmits the CNFM packet. By this way, the transmitter and/or receiver node serves more than one rank matrices in which it has the same rank.

The use of PR period is optional since it is required only in the following three cases: (*i*) At an intermediate node of a multi-hop route, (*ii*) When a node has data packets for its multiple 1-hop receiver nodes in the same or different rank matrices, and (*iii*) When a node receives data packets from its multiple 1-hop transmitter nodes of different rank matrices. Further, it is useful for light and moderate traffics, where a node can schedule its unused slots to non-conflicting flows.

3.3.6 Data Traffic Period

Each slot in this period is divided in two sub-slots - data and ACK (acknowledgment). Transmitter node transmits data packet in the first sub-slot as per the negotiated schedule, and receiver node responds with ACK packet in the second sub-slot. The frame finishes with the end of the data traffic period and all nodes recalculate their *fully connected 1-hop neighborhoods* for the next frame.

3.4 Our Proposed Distributed, Directional TDMA Scheme

Fig. 3.6 shows the working of a directional TDMA-based MAC scheme for a 3-hop route. Our proposed TDMA scheme divides a frame into three periods (Hello, reservation period and data traffic period) (see Fig. 3.3). Here, the first two periods represent the control

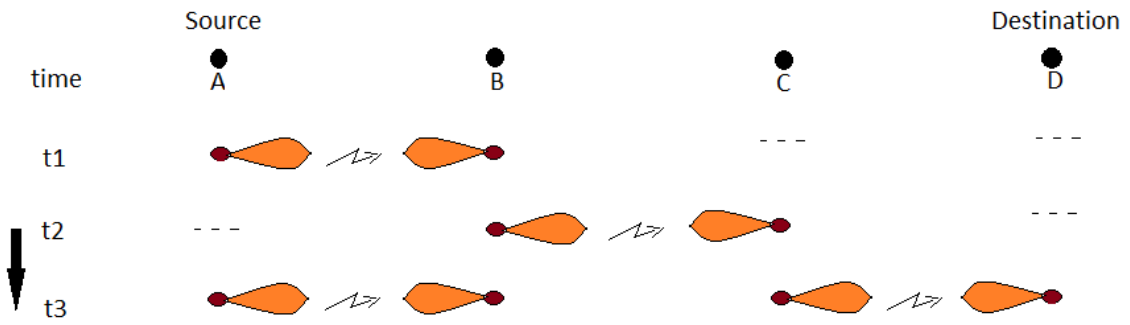


Figure 3.6: Illustration of a directional TDMA scheme for a 3-hop topology, where source node A transmits packets to destination node D over the 3-hop route A-B-C-D. In time slot t1, node A forwards its packet to intermediate node B. In slot t2, node B steers its beam towards node C and forwards the packet. In time slot t3, spatial reuse is achieved as links A-B and C-D communicate simultaneously.

period of the protocol. Our scheme also adds two new optional and conditional REQ and PR control periods as discussed in Section 3.3. The number of control time slots depends on the maximum number of nodes in a *fully connected 1-hop neighborhood*.

At the beginning of a frame, every node calculates the rank matrix for each of its *fully connected 1-hop neighborhood*. In each **Hello Period slot**, a node selects a node id from its rank matrices, towards which it steers its beam to either transmit or receive the Hello packet. This allows the node to reserve a conflict-free slot for the transmitter and receiver pair to negotiate the scheduling of data traffic slots. Note that this order of the selected node id (obtained during the Hello period) also remains the same in the REQ and both reservation periods. If a node decides to transmit the Hello packet in a slot, it updates the *desired throughput* field with the count of data packets it has in its queue for that receiver (see Section 3.3.1 for more details).

At the end of the Hello Period, if a transmitter can no longer utilize the total number of reserved traffic slots, it sends an **REQ packet** to its receiver node(s) with the updated *desired throughput*. Using an REQ packet, therefore, prevents the wastage of unused data traffic slots at a node. If the total requested data traffic slots exceed its available data traffic slots, the node uses *throughput scaling* to suitably scale down the requested *desired throughputs* as discussed in Section 3.3.2. The throughput scaling mechanism improves the

network *fairness* and helps in congestion management.

Upon receiving a Hello packet, the receiver node records the transmitter node id and the *desired throughput*, sends a REP packet during the **Reservation Period** with a list of its own available data traffic slots, and waits for the CNFM packet from transmitter during the same reservation slot. The transmitter node accepts the common slots and sends a CNFM packet to the receiver node. Since a collision occurs when a receiver node receives more than one signal at the same time, transmitting the REP and CNFM packets ensure that the nodes, exposed to the communication of this transmitter-receiver pair, are prevented from scheduling the same data traffic slots. Finally, data packets are transmitted in the **Data Traffic Period**.

We discuss the pseudocode of our proposed scheme in Algorithms 2 to 4 and an example to explain its working is given below. Although the introduction of the optional REQ and PR periods can slightly increase the frame length, the channel utilization, spatial reuse, and link fairness increase considerably. Moreover, it allows nodes to receive and forward packets in the same frame which reduces the end-to-end delay. We compare the control overhead and running time metrics of our scheme with other recent, fast, distributed TDMA schemes in Section 3.5.2.

Example: This example explains our proposed scheme, when an intermediate node receives and forwards data packets to/from nodes in the same or different rank matrices. Here, we use the network topology shown in Fig. 3.1 where intermediate node 5 receives packets from source node 2 and forwards them to destination node 3. As per the rank matrices of node 5 (i.e., [1,2,5], [3,5], [4,5]), it can either receive Hello packet from node 2 or transmit its own Hello packet to node 3 in the 2^{nd} Hello slot. As a result, it cannot communicate with both nodes in a given frame. Hence, it uses PR period as discussed below and shown via a timing diagram in Fig. 3.7. Here, we assume that a source node generates 40 packets per frame, the total data traffic slots are 100, and one data packet is transmitted per data traffic slot.

Algorithm 2: Pseudocode of our TDMA scheme for frame f at node i (Part 1)

Global variables: Maximum number of nodes in a *fully connected 1-hop neighborhood* (K), total data slots D in a frame, a Kx1 vector *curr_PR_status* to store node id(s) with which node i agreed (in frame f -1) to communicate in PR period of current frame (i.e., frame f). If k^{th} PR slot of node i is free, its *curr_PR_status*[k] = 0.
//Frame starts
Step 1: Construction of rank matrices
At the beginning of the frame, node obtains its 2-hop neighborhood information via neighbor discovery
Create rank-matrix using Algorithm 1 and Section 3.2.2
Step 2: Functionality in Hello period
Initialize Kx1 size vectors for *order, desired_throughput*, and *next_PR_status*
for ***each*** *slot k in Hello period (i.e., k ∈[1,K])* **do**
    Select a node id (n) of rank k from its rank matrices (as discussed in Section 3.3.1)
    order[k] = n *//use this order in subsequent subperiods*
    if *node i is transmitter for link i-n* **then**
        desired_throughput[k] = min(D, packets for node n stored in the buffer of node i)
        if *node i wants to use PR period in next frame & next_PR_status[k] is 0* **then**
            //see details in Section 3.3.5
            Set *Enable Redundant Slot?* field to 1
            next_PR_status[k] = n
        end
        Transmit Hello packet
    else
        Receive Hello packet and store requested desired throughput in *desired_throughput* [k]. If *Enabled Redundant Slot?* field is 1, set *next_PR_status*[k] = n.
    end
end
Step 3: Perform *throughput scaling* at **transmitter** nodes
if *total requested desired throughput at node i > D* **then**
    perform *throughput scaling* (see Section 3.3.2)
    is_REQ_required = True
end
Step 4: Functionality in REQ period
for ***each*** *slot k in REQ period (i.e., k ∈[1,K])* **do**
    if *is_REQ_required & node i is transmitter for link i-order[k]* **then**
        Transmit REQ packet towards receiver node
        order[k] with the updated *desired_throughput*[k]
    else
        if *node i is receiver for link i-order[k] & it receives REQ packet* **then**
            Update its *desired_throughput*[k]
        end
    end
end
Step 5: Perform *throughput scaling* at **receiver** nodes
Continued in Algorithm 3

Frame 1:

Step 1: Frame 1 starts. Node 5 decides to listen to node 2 in its 2^{nd} Hello slot (HP S2) and

Algorithm 3: Pseudocode of our TDMA scheme for frame f at node i (Part 2)

```
1  Step 6: Functionality in Reservation period
2  for each slot k in reservation period (i.e., k∈[1,K]) do
3      if node i received Hello packet in k^th Hello slot then
4          Set bits corresponding to its available data slots to 0 in Traffic Slots field, and update
             Scaled Desired Throughput
5          if next_PR_status[k] is not 0 then
6              Set Redundant Slot Request Accepted? = 1
7          else
8              if wants to use PR period in next frame then
9                  Set Enabled Redundant Slot? field to 1
10                 next_PR_status = order[k]
11             end
12         end
13         Transmit REP packet; receive CNFM packet
14         Update its reserved data traffic slots
15         if Redundant Slot Request Accepted? is 0 then
16             next_PR_status[k] = 0
17         end
18     else
19         if node i sent Hello packet in k^th Hello slot then
20             Wait for REP packet
21             if no REP received then
22                 Retransmit Hello packet in next frame
23             end
24             Select up to Scaled Desired Throughput common available data slots and update
                 Traffic Slots field
25             if Enabled Redundant Slot? field is 1 & its next_PR_status[k] is 0 then
26                 Set Redundant Slot Request Accepted? = 1
27             end
28             Transmit CNFM packet
29         end
30     end
31 end
32 Continued in Algorithm 4
```

receives a Hello packet from it, with the *desired throughput* value of 40 data packets.

Step 2: Node 5 does not receive any REQ packet from node 2 during 2^{nd} REQ slot (i.e., REQ S2).

Step 3: Node 5 transmits REP packet to node 2 in Reservation Period slot 2A (RP S2A) with the *Enable Redundant Slot?* field set to 1. This allows node 5 to ask node 2 whether it agrees for using PR Period as it needs to send Hello packet to node 3 in the next frame (i.e., Frame 2) and can have unused Data slots which can be reserved for node 2.

Step 4: Node 2 receives REP packet, selects 40 common available data traffic slots, and

Algorithm 4: Pseudocode of our TDMA scheme for frame f at node i (Part 3)

```
1  Step 7: Functionality in PR period
2  for each slot k in PR period (i.e., k∈[1,K]) do
3      if curr_PR_status[k] is not 0 then
4          if node i is receiver on link i-order[k] then
5              Set bits for available data slots to 0 in Traffic Slots field and Scaled Desired
                 Throughput field to 0
6              Transmit REP packet towards node order[k]
7              Wait for CNFM packet
8              Upon receiving CNFM packet, update the reserved data traffic slots
9          else
10             Wait for REP packet
11             Upon receiving REP packet, select common available data slots, and update Traffic
                 Slots and Scaled Desired Throughput fields
12             Transmit CNFM packet
13         end
14     end
15 end
16 Step 8: Communicate during Data Traffic period
17 Step 9: curr_PR_status=next_PR_status //for next frame
18 //Frame completes
```

transmits CNFM packet towards node 5 in Reservation Period slot 2B (RP S2B), with the *Redundant Slot Request Accepted?* field set to 1.

Step 5: Node 5 receives CNFM packet, stores the selected data traffic slots, and notes that node 2 has agreed to use PR period in the next frame (i.e., Frame 2). *Note:* Node 2 or 5 does not transmit or receive any packet in the PR Period of current frame (i.e., Frame 1).

Step 6: Node pair 2 and 5 exchange data packets in the 40 reserved data traffic slots.

Step 7: Frame 1 is completed. Frame 2 starts.

Frame 2:

Step 8: Node 5 transmits Hello packet to node 3 in Hello period slot 2 (i.e., HP S2) with *desired throughput* request of 40 and *Enabled Redundant Slot?* field set to 1. No REQ packet is sent in this case.

Step 9: Node 3 receives this Hello packet of node 5 in its HP S2 slot.

Step 10: Node 3 transmits REP packet in Reservation Period slot 2A (i.e., RP S2A) with its 100 available data traffic slots and sets the *Redundant Slot Request Accepted?* field to 1.

Step 11: Node 5 receives REP packet, selects 40 common available slots, and then transmits

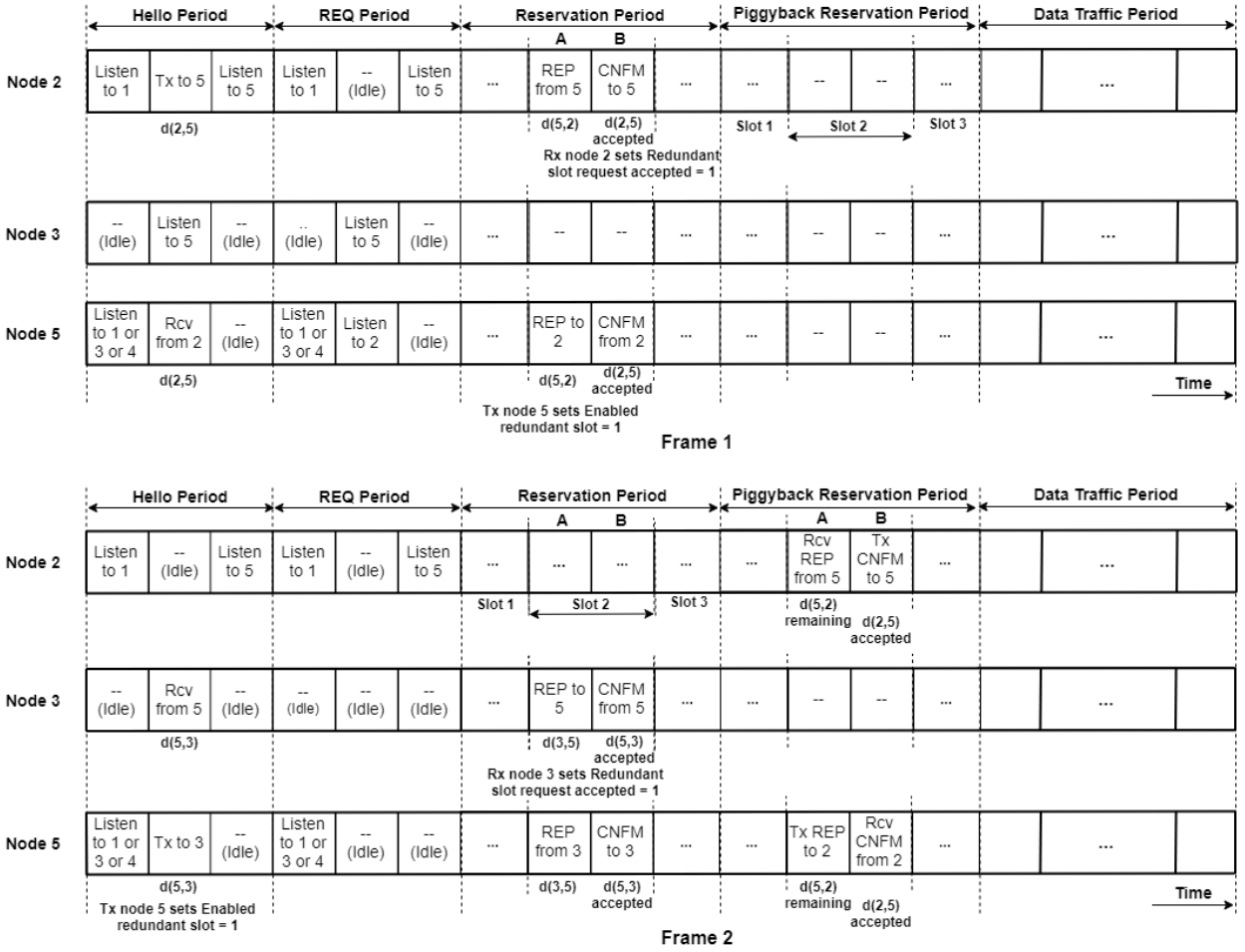


Figure 3.7: Timing diagram of our proposed MAC scheme where node 5 (in Fig.3.1) is an intermediate node which receives packets from source node 2 and forwards them to destination node 3. Here, we show the use of PR period which allows node 5 to talk to both nodes 2 and 3 in the same frame (from Frame 2 onwards).

the CNFM packet towards node 3 in Reservation Period slot 2B (i.e., RP S2B).

Step 12: Based on its exchange with node 2 in Frame 1, Node 2 steers its beam towards node 5 in the PR Period slot 2A (i.e., PRP S2A). Node 5 transmits REP packet with its 60 available data traffic slots towards node 2 in PRP S2A. The *Scaled desired throughput* field is set to 0 in this REP packet. (*Note:* node 5 has scheduled its 40 out of 100 slots with node 3 in Step 11. Hence, it sends its remaining 60 unreserved data traffic slots to node 2).

Step 13: Node 2 receives the REP packet and selects 40 common available data traffic slots, updates the *Traffic slots* and *Scaled desired throughput* fields in the CNFM packet, and then transmits it in PRP S2B to node 5.

Note: Since node 5 had 60 available slots, source node 2 was able to forward all 40 packets to node 5. However, if the data rate increases (e.g., 60 packets per frame are generated at source node 2), node 5 can offer only 40 unreserved slots to node 2, which would leave the remaining 20 packets in the buffer at node 2. This would lead to congestion at node 2 in the subsequent frames and increase the queuing delay. If during this PR period in Frame 2, node pair 2 and 5 agrees on using PR period in Frame 3, packets can expire due to higher queuing delay. Therefore, node pair 2 and 5 must talk in the Hello period of the next frame (i.e., Frame 3) to ensure fairness for link 2-5. Therefore, the REP and CNFM packets in the PR Period do not set the *Enable Redundant Slot?* and *Redundant Slot Request Accepted?* fields to 1.

Step 14: Node 5 receives CNFM packet of node 2 in PRP S2B.

Step 15: Node 5 communicates over links 2-5 and 5-3 in the data traffic period. Frame 2 ends and Frame 3 starts.

Frame 3:

Step 16: Node 2 transmits Hello packet to node 5 in Hello period slot 2 (i.e., HP S2), where the *desired throughput* is set to 40. Node 5 receives the Hello packet in its HP S2 slot.

Step 17: Node 5 transmits REP packet in Reservation period slot 2 mini slot A (i.e., RP S2A) with its 100 available data traffic slots and the *Enable Redundant Slot?* field set to 1.

Step 18: Node 2 receives the REP packet, selects 40 common available slots, sets *Redundant Slot Request Accepted?* field to 1, and transmits the CNFM packet to node 5 in RP S2B.

Step 19: In PRP S2A (i.e., PR period slot 2A), node 3 transmits REP packet towards node 5 with its available data traffic slots.

Step 20: Node 5 replies with CNFM packet where it selects 40 common data traffic slots in PRP S2B.

Step 21: Node 5 communicates over links 2-5 and 5-3 in the same frame. Frame 3 ends and Frame 4 starts.

Step 22: Go to Step 8 (*note:* Frame 4 is same as Frame 2).

3.5 Simulation Results and Performance Comparison

The performance of our proposed rank-based TDMA scheme is evaluated in Section 3.5.1 for real-time traffic flows over multi-hop routes by using different data rates, TTL values, and QoS metric. Then the control overhead of our scheme is compared with recent, fast, distributed TDMA schemes proposed in [98–100] in Section 3.5.2, followed by the analysis of their performance comparison for different static and mobile scenarios in Section 3.5.3.

3.5.1 Performance of the Proposed TDMA Protocol

We first describe the simulation setup, followed by a discussion on the maximum achievable flow throughput. Then, the performance of our scheme is evaluated for different experiments.

(A) Simulation Setup:

The simulations are run in MATLAB version R2017b for the network topology consisting of 14 nodes shown in Fig. 3.8. The rank matrix for each node is shown in Table 3.1. Each node is equipped with a directional antenna with a beam-width of 45^0 and 2 km transmission range. We assume an ideal beam with no side or back lobes. The network size is 10 km x 10 km, where nodes are randomly placed. The channel capacity is 10 Mbps. The number of data traffic slots in each frame is 100 and the data packet size is 1000 Bytes. The buffer size at each node is infinite. We assume that each node knows about its neighborhood and the route(s) passing through it.

We assume that a *fully connected 1-hop neighborhood* can have up to 10 nodes. Therefore, the number of slots in Hello, REQ, and both reservation periods is 10. These periods (i.e., L_{Hello} for Hello period and L_{RP} for reservation period) are calculated as shown in (3.4) and

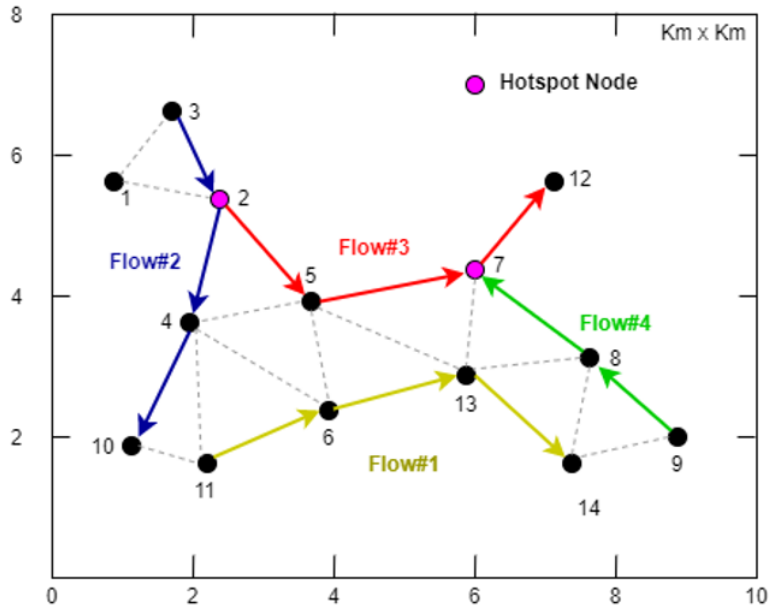


Figure 3.8: Directional network topology. Each node is equipped with a single-beam directional antenna. Tested routes are (*i*) 11 to 14 (Flow#1) (*ii*) 3 to 10 (Flow#2), (*iii*) 2 to 12 (Flow#3), and (*iv*) 9 to 7 (Flow#4). Nodes 2 and 7 are hotspot nodes, as they are serving more than one route.

Table 3.1: Nodes of Fig. 3.8 and their Rank Matrices in their 1-Hop Fully Connected Neighborhoods

Node	**Rank matrices for 1-hop fully connected neighborhoods at each node**
1	[3,2,1]
2	[3,2,1], [4,5,2]
3	[3,2,1]
4	[4,5,2], [4,5,6], [4,6,11], [4,11,10]
5	[4,5,2], [4,5,6], [5,7,6,13]
6	[4,5,6], [4,6,11], [5,7,6,13]
7	[5,7,6,13], [7,8,13], [7,12]
8	[7,8,13], [8,14,9], [8,13,14]
9	[8,14,9]
10	[4,11,10]
11	[4,6,11], [4,11,10]
12	[7,12]
13	[5,7,6,13], [7,8,13], [8,13,14]
14	[8,13,14], [8,14,9]

(3.5) below and their lengths are given in Table 3.2. The MAC control period overhead (which includes Hello, REQ, and both reservation periods) is just 3.34% in our scheme. For

Table 3.2: Duration of Different Periods of a Frame and its Sub-Frames

Control/Data Period	**Length (in ms)**
Hello period (of 10 slots)	0.403
REQ period (of 10 slots)	0.403
Reservation period (of 10 slots)	1.097
PR period (of 10 slots)	1.097
Data Traffic period (of 100 slots)	86.760
Frame length	89.760

a unit decrease in the number of fully connected 1-hop neighbor nodes, the control period decreases linearly by 0.3 ms. The data traffic period length (L_{Data}) is calculated in (3.6), where N_{Hello} and N_{Data} represent the number of Hello and data traffic slots in a frame, respectively. The S_{Hello}, S_{REP}, S_{CNFM}, S_{Data}, S_{ACK}, S_{PLCP}, and S_{MAC} represent the size of respective packets and headers. The slot length in the control and data traffic periods include maximum propagation delay (δ), which allows a packet to be received in the same slot by the nodes located at the coverage boundary of transmitter node. A node waits for a short inter-frame space (SIFS) duration t_{SIFS} of 10 μs before responding to the received packet. R_c represents the channel rate.

$$L_{Hello} = N_{Hello} \times \left[\frac{(S_{Hello} + S_{PLCP})}{R_c} + \delta \right] \tag{3.4}$$

$$L_{RP} = N_{Hello} \times \left[\frac{(S_{REP} + S_{CNFM} + 2 \times S_{PLCP})}{R_c} + 2 \times \delta + t_{SIFS} \right] \tag{3.5}$$

$$L_{Data} = N_{Data} \times \left[\frac{(S_{Data} + S_{MAC} + S_{PLCP} + S_{ACK})}{R_c} + 2 \times \delta + t_{SIFS} \right] \tag{3.6}$$

We study the performance of our proposed MAC scheme for the network topology in Fig. 3.8, where the source nodes 11, 3, 2, and 9 generate packets for the destination nodes 14, 10, 12, and 7, respectively. The routes corresponding to these four source-destination node pairs are 11-6-13-14 (a 3-hop route for Flow#1), 3-2-4-10 (a 3-hop route for Flow#2), 2-5-7-12 (a 3-hop route for Flow#3), and 9-8-7 (a 2-hop route for Flow#4). Here, we call nodes 2 and

7 as the *hotspot nodes* because they forward the data packets of more than one flows and therefore can experience congestion.

(B) Performance Metrics:

The following performance metrics are used in our simulation:

- Per node CUR (channel utilization ratio) is the fraction of time that a node either transmits or receives data packets over the total simulation time [83].
- PDR (packet delivery ratio) for a flow is the ratio of total packets received by the destination node over total packets generated at the source. Since PDR represents a normalized throughput, we can compute the flow throughput as PDR $\times$ Data Rate.
- The end-to-end delay plot is shown only for Section 3.5.1(F) which includes a scenario when packet TTL is not used. Since all other experiments have a TTL value, we have omitted their end-to-end delay plots.

(C) Maximum Achievable Flow Throughput:

The nodes use directional antenna and can share their data traffic slots among one or more active links, where an active link is a part of a route on which data is transmitted in each frame. For example, node 6 in Fig. 3.8 has two active links, 11-6 and 6-13. The throughput at a node decreases as the number of its active links increases. Therefore, the link throughput cannot exceed 50% of channel capacity in a multi-hop route because the node receives and forwards the data packets. In fact, after considering the control packet overhead, the maximum achievable link throughput for a flow on a multi-hop route would be less than 50% of the channel capacity. In our scheme, each frame (as shown in Table 3.2) consists of a control period and a data traffic period where S_{Data}, S_{MAC}, S_{ACK}, and S_{PLCP} are 1000, 34,

14, and 24 bytes, respectively. Thus, the size of a 1000-byte data packet increases to 1058 bytes after adding the MAC and physical layer protocol headers (the higher layer protocol headers are ignored here). The receiver acknowledges a successful reception of data packet by sending a 14-byte ACK packet to the transmitter in the same data traffic slot.

As shown in Table 3.2, the frame consisting of 100 data traffic slots has a length of 89.76 ms. Since each data traffic slot can forward a packet carrying 1000-byte data payload, a 10 Mbps channel can support the maximum flow data rate of 8.9 Mbps, when the source and destination are 1-hop away. For a latency-constrained streaming application over a $\geq$ 2-hop path, the maximum flow data rate (when using directional communication shown in Fig. 3.6) would decrease to half of 8.9 Mbps (4.45 Mbps). Note that a higher flow data rate over a multi-hop path would further decrease the maximum achievable data rate due to congestion and TTL-based packet expiry.

(D) Channel Utilization for Different Data Rates:

The maximum per node CUR is 0.97 for the simulation setup considered in Section 3.5.1. The hotspot node 2 is an intermediate node of Flow#2 and the source of Flow#3 (see Fig. 3.8). Since it uses all its data slots at 3 to 5 Mbps data rates to accommodate both flows, its CUR is maximum in Fig. 3.9. Note that it allocates the same number of data slots to its downstream nodes 4 (for Flow#2) and 5 (for Flow#3), when QoS metric is not used. Therefore, intermediate nodes 4 and 5 also have a constant CUR in Fig. 3.9. Since the number of data slots required by a node increases with traffic density, CUR values of the remaining nodes in Fig. 3.9 increase with the data rate.

Source nodes 3, 9 and 11 in Fig. 3.8 only forward their packets to the next-hop nodes, whereas intermediate nodes 2, 4 to 8 and 13 receive and forward the packets. Therefore, the CUR value is the lowest at the source nodes 3, 9 and 11 for all traffic densities in Fig. 3.9.

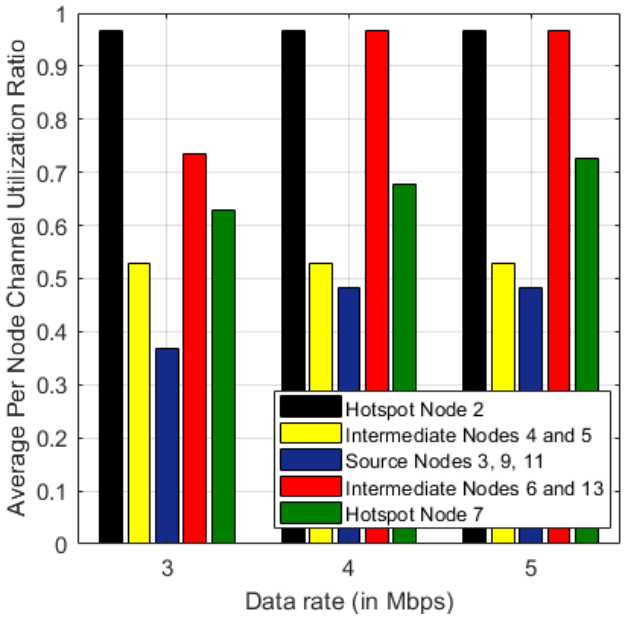


Figure 3.9: Average per node channel utilization ratio in the proposed TDMA scheme at different data rates, when the QoS metric is not used.

Note that the congestion at hotspot node 2 increases with data rate. As a result, hotspot node 7 receives and forwards fewer packets of Flow#3. Since it also receives packets of Flow#4 (see Fig. 3.8), its CUR is higher than that of intermediate nodes 4 and 5 but lower than the CUR of intermediate nodes 6 and 13, in Fig. 3.9.

(E) Performance for Different Data Rates, at TTL = 0.5 s:

We evaluate the performance of our proposed TDMA scheme for the four real-time flows (for network in Fig. 3.8) at a constant data rates of 2 to 5 Mbps per flow which represent a low, moderate, and heavy traffic load, at a data packet TTL value of 0.5 s. Since a node does not use QoS metric, it arranges and forwards the packets of each flow in the FIFO (first-in-first-out) order from the queue.

- Performance of Flow#1: Flow#1 uses an independent route with no hotspot node. The PDR of Flow#1 (see yellow bars in Fig. 3.10) is 100% for up to 4 Mbps data rates, which are less than the maximum achievable flow throughput of 4.45 Mbps (as

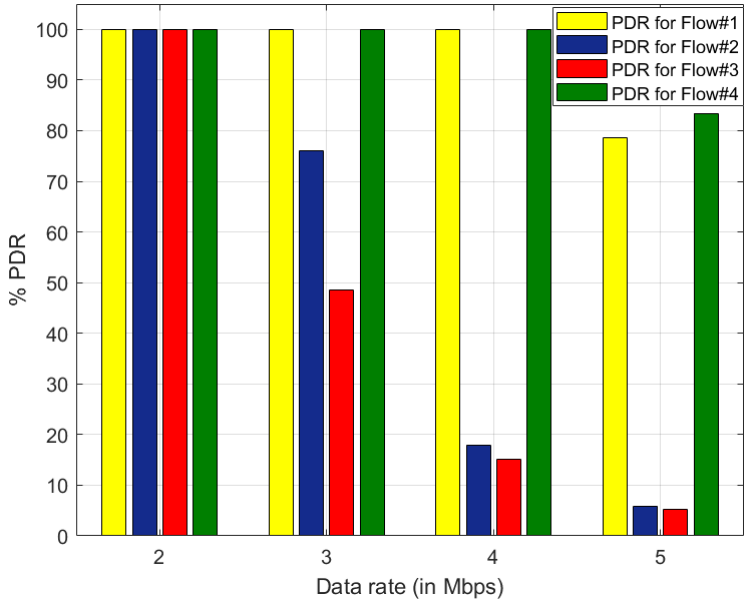


Figure 3.10: PDR performance of the proposed TDMA scheme at different data rates, when packet TTL is 0.5 s.

discussed in Section 3.5.1(C)). At 5 Mbps data rate, congestion builds at the source node 11, which leads to packet expiration due to an increased queuing delay, and the PDR of Flow#1 degrades to 79%.

- Performance of Flow#2 and Flow#3: These flows attain 100% PDR at the low data rate of 2 Mbps in Fig. 3.10 because their intermediate nodes do not experience congestion. Note that the flows serviced by a hotspot node experience a higher queuing delay, and therefore, have a higher packet drop due to TTL expiry. In addition, the congestion experienced at a hotspot node increases with the data rate. Therefore, PDR is the highest in Flow#1, then in Flow#2, followed by Flow#3 at 3 to 5 Mbps data rates in Fig. 3.10 because they have zero, one and two hotspot nodes, respectively (see Fig. 3.8).

  Since hotspot node 2 attains the maximum CUR at 3 Mbps data rate (see Fig. 3.9), an increase in the data rate further aggravates its congestion. Therefore, the PDR values of Flow#2 and Flow#3 considerably degrade when data rate increases from 3 Mbps to 5 Mbps in Fig. 3.10.

- Performance of Flow#4: Due to the high congestion-induced queuing delay experienced at hotspot node 2 at 3 to 5 Mbps data rates, fewer packets of Flow#3 reach intermediate node 7. As a result, intermediate node 8 of Flow#4 can reserve sufficient slots with the destination node 7 in each frame, and attains a 100% PDR for Flow#4.

  At 5 Mbps data rate, congestion builds up at the source node 9, which leads to packet drop due to TTL expiry, and thereby, reduces PDR of Flow#4 to 83%. Since Flow#1 is one hop longer than Flow#4, its end-to-end delay and, thereby, packets dropped due to TTL expiry are higher than that of Flow#4. Therefore, the PDR of Flow#1 is lower than PDR of Flow#4 at 5 Mbps data rate in Fig. 3.10.

(F) Performance for Different TTL Values at a 5 Mbps Data Rate:

Here, we evaluate the performance of our proposed TDMA scheme in the presence of traffic congestion for different flow latency, and show that a higher throughput is achieved when the flow can tolerate a higher latency.

We study the performance of our proposed TDMA scheme at 5 Mbps data rate (where all four flows experience congestion) for TTL values of 0.5 s, 1 s, 3 s, and when TTL is not used. Fig. 3.11(a) shows that PDRs of all the four flows increase with an increase in the TTL value because less packets expire due to congestion-induced queuing delay. As explained in the previous section, PDRs of Flow#2 and Flow#3 are lower than the PDRs of the other two flows due to congestion induced packet drops. Although the PDR of Flow#4 at node 7 is 100% for TTL $\geq$ 1 s and no TTL, the end-to-end delay is higher when TTL is not used (see green bars in Fig. 3.11(b)) because nodes 2 and 5 do not drop the packets of Flow#3. Hence, node 7 receives more Flow#3 packets for forwarding them to node 12. As a result, node 7 schedules fewer data traffic slots with node 8, which increases the queuing delay of packets of Flow#4. Note that the end-to-end delay increases with simulation time when the

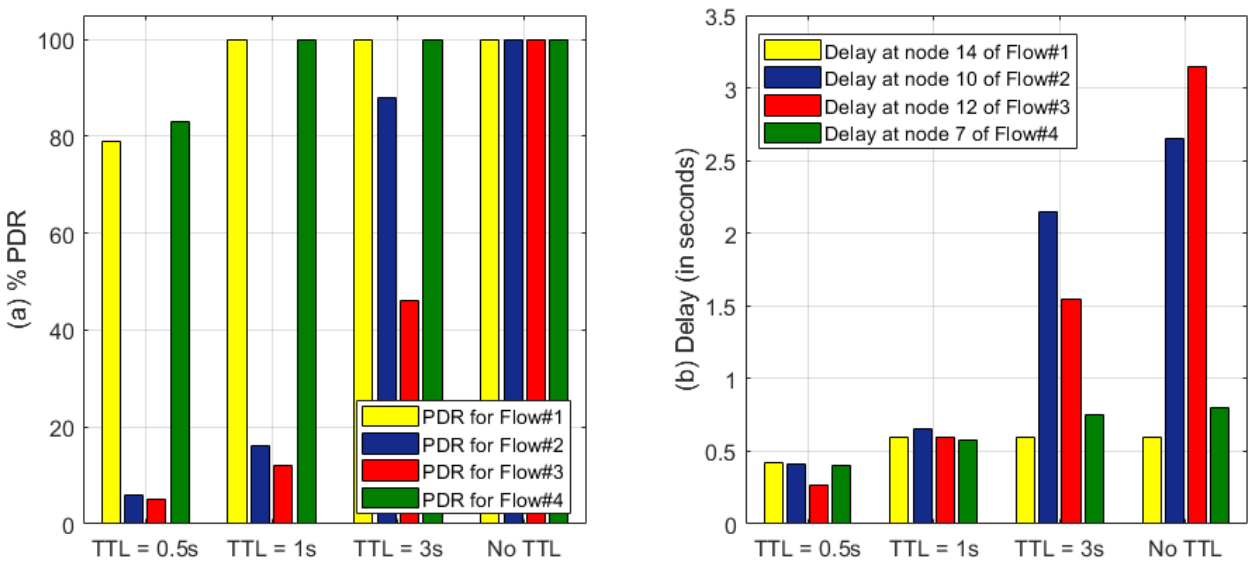


Figure 3.11: Impact of packet TTL value on (a) PDR and (b) end-to-end delay performances of the proposed TDMA scheme for heavy traffic (5 Mbps per flow).

data rate is high, causing the congestion. Since packets do not expire in the queues of source and/or intermediate nodes when TTL value is not used, all four flows achieve 100% PDR.

(G) Performance of QoS-aware TDMA Scheme:

We evaluate the performance of our proposed TDMA scheme with the QoS metric at different data rates. While Flow#3 has a lower priority (Priority = 1), the remaining three flows have a higher priority (Priority = 2). The value of TTL for a packet is 0.5 s.

Since Flow#1 uses an independent route (i.e., it does not have any hotspot node), its PDR and end-to-end delay are the same in the QoS-aware and without-QoS MAC schemes (see yellow bars in Fig. 3.12 and 3.10, respectively). As Flow#2 has a higher priority (2×) than Flow#3, the hotspot node 2 forwards more packets of Flow#2, which increases the queuing delay of packets of Flow#3. As a result, PDR of Flow#2 (see blue bars in Fig. 3.12) in QoS-aware MAC scheme increases at the cost of Flow#3 (see red bars in Fig. 3.12). Although a higher queuing delay experienced by Flow#3 packets at node 2 decreases their TTE value, along with their higher value of remaining hop count (which is 3), the QoS metric for Flow#3

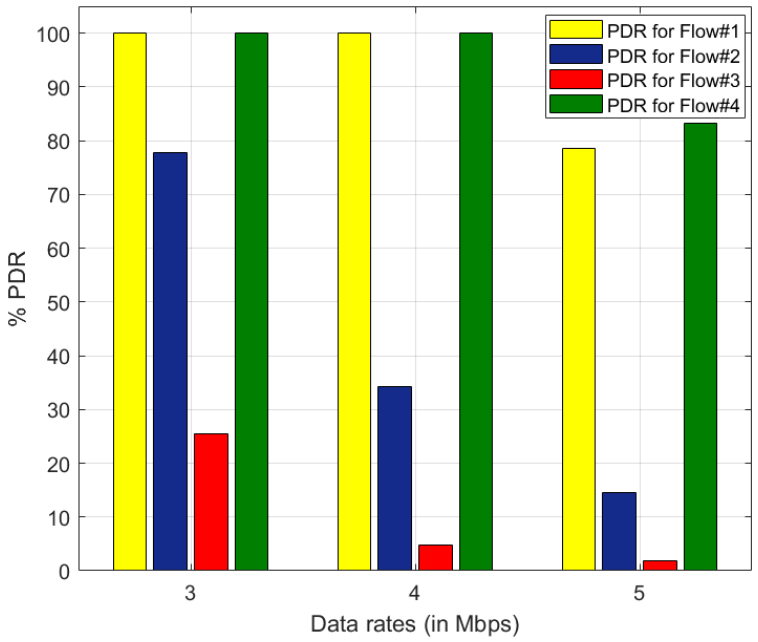


Figure 3.12: PDR at destination nodes for the QoS-aware proposed TDMA scheme at different data rates.

is still lower than that of Flow#2 (due to a higher priority of Flow#2). Here, the PDR of Flow#3 is low because its packets expire in the queues at the intermediate nodes 5 and 7. Since fewer packets of Flow#3 arrive at node 7, congestion does not build up at node 8. As a result, the PDR of Flow#4 remains the same at all data rates in the QoS-aware and without-QoS MAC schemes (see green bars in Fig. 3.12 and 3.10).

Observation 1 (QoS Metric): *In the QoS-aware MAC scheme, PDR of higher QoS flow(s) increases at the expense of lower QoS flows. However, the use of the QoS metric does not impact the PDR of independent flows.*

(H) Advantage of Piggyback Reservation Period in the Proposed TDMA Scheme:

In this section, we study the impact of using the piggyback reservation on the performance of our proposed MAC scheme for the network topology shown in Fig. 3.8. Here, the flow data rate is 5 Mbps and TTL is 0.5 s.

Node 6 in Fig. 3.8 is an intermediate node of Flow#1 and has rank matrices of [4,6,11] and [5,7,6,13] (see Table 3.2). To schedule data traffic slots with both previous and next hops of Flow#1 (i.e., nodes 11 and 13, respectively), it requires two unique conflict-free reservation slots. In the absence of *PR period*, it can reserve only one reservation slot in a frame by either transmitting its Hello packet to node 13 or receiving the Hello packet of node 11, in the 3^{rd} Hello slot. As a result, node 6 schedules data traffic slots either with node 11 or node 13, in a frame, and wastes its remaining unutilized data traffic slots. This reduces spatial reuse at node 6 and increases the queuing delay of packets of Flow#1, which leads to packet drop. Hence, PDR of Flow#1 degrades from 79% (of the proposed MAC scheme which uses PR period) to 38% (proposed MAC scheme *without PR period*) in Fig. 3.13.

Hotspot nodes 2 and 7 experience the same situation (as node 6) for links 2-4 and 2-5, and 5-7 and 7-12, respectively. However, unlike node 6, they experience high congestion. Hence, they schedule all of their data traffic slots in the Reservation period, which leaves zero data traffic slot for the PR period. For this reason, removing PR period does not degrade the PDRs of Flow#2, Flow#3, and Flow#4.

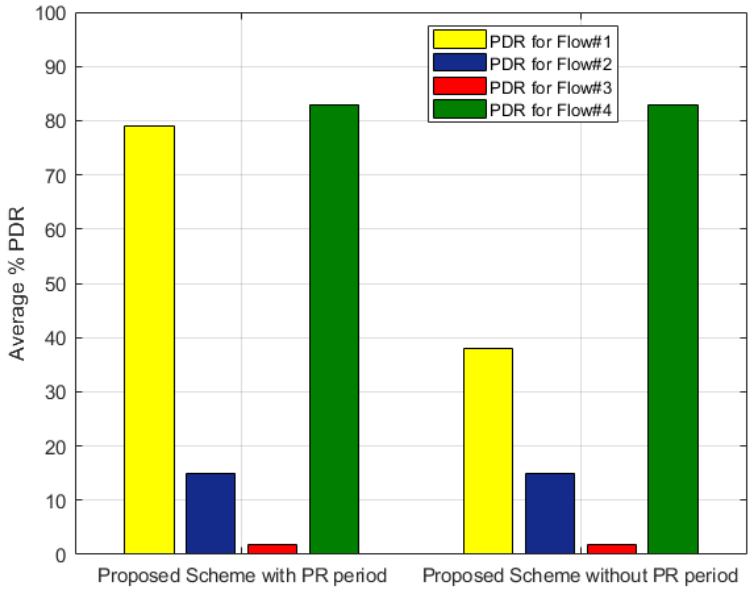


Figure 3.13: PDR performance of the proposed TDMA scheme with and without using Piggyback Reservation period, when data rate is 5 Mbps and TTL = 0.5 s.

Observation 2 (Piggyback Reservation): *The PR period allows an intermediate node to receive and forward packets in the same frame, which reduces the queuing delay and improves the PDR.*

3.5.2 Control Overhead Comparison with Other TDMA Schemes

The use of two additional (optional) control periods can slightly increase the control overhead and frame length of our scheme. In this section, we compare the control overhead of our scheme with the recently published distributed TDMA schemes[4] (i.e., DSTO (distributed scheduling using topological ordering) [98], EB-ET-DRAND (distributed TDMA scheduling algorithm based on the exponential backoff rule and energy-topology factor) [99], and E-T-DRAND (distributed TDMA slot scheduling algorithm based on the energy-topology factor) [100]). These schemes reduce the control overhead and running time required to obtain a conflict-free reservation slot for each node in a dynamic, multi-hop network. Note that these schemes do not consider the data traffic period. As a result, a frame corresponds to the Hello and Reservation control periods in our scheme. The following two metrics are used for the comparison:

1. Number of transmissions is the total control messages transmitted by all nodes to obtain a conflict-free schedule for the reservation period. A lower transmission count reduces the control overhead [98].

2. Average running time is the time taken for all nodes to acquire a conflict-free reservation slot. A lower running time is desired for dynamic topology [98].

[4]Schemes in [98–100] use an omnidirectional antenna and require each node to broadcast its schedule in its 1-hop neighborhood. Note that the directional variants of these schemes incur a huge sweeping delay and overhead, which we have ignored in this comparison, for simplicity.

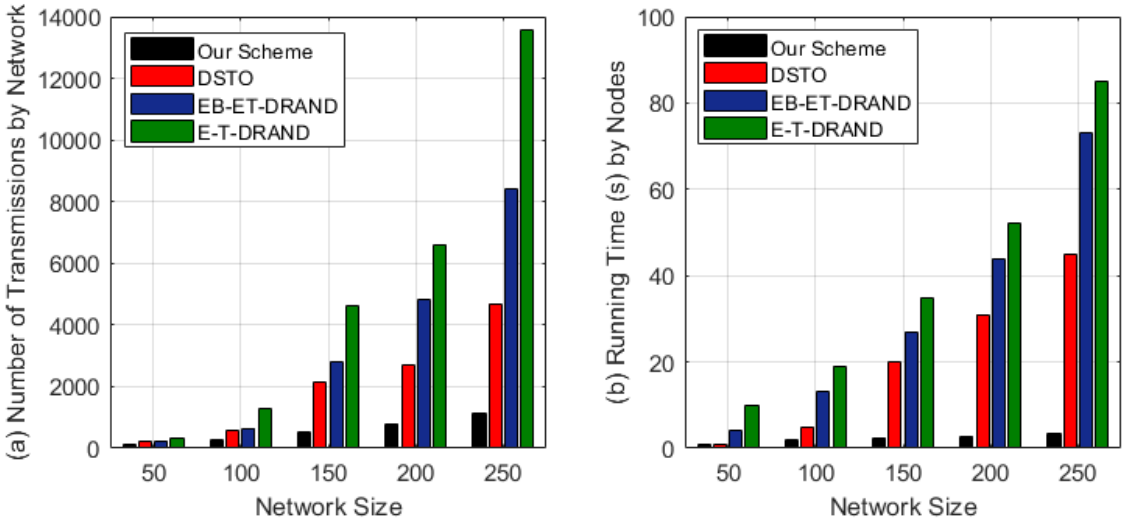


Figure 3.14: Comparison of (a) total message transmissions and (b) average running time to obtain a conflict-free reservation slot at each node for different network sizes.

We have used the simulation setup used by [98–100], where the network size is varied from 50 to 250 nodes in a 300 m x 300 m simulation area, with a signal transmission range of 40 m. Each node randomly selects its node pair (i.e., receiver node) from its 1-hop neighborhood and each receiver node knows its transmitter node(s). Each experiment is run 10 times.

As shown in Fig. 3.14(a), the **average number of transmissions** required in our proposed scheme is significantly lower than the distributed TDMA schemes in [98–100] for different network sizes. Note that each node repeatedly broadcasts its updated schedule to resolve a conflict with its 1-hop neighbors in [98–100]. As shown in Fig. 3.14(b), our scheme has a significantly lower **average running time** as compared to the schemes in [98–100], due to its lower notification overhead. Note that the number of slots required in the control period in our scheme at a network density is determined from Fig. 3.2(a).

3.5.3 Performance Comparison with Other TDMA Schemes

To the best of our knowledge, no other directional, distributed TDMA scheme exists which can provide a conflict-free schedule for a dynamic, multi-hop topology in real-time with no

notification overhead and delay. Therefore, we have compared our scheme with a recently published omnidirectional, distributed TDMA scheme [98], which provides a conflict-free reservation slot for each node in a dynamic, multi-hop network, while minimizing the running time and control overhead. We refer to the scheme in [98] as typical distributed TDMA scheme from here onward.

In this section, we compare the performance in terms of PDR and end-to-end delay, for varying traffic densities in static and mobile network topologies. Note that the control overhead and running time were compared in the previous section. For a fair comparison, we consider that the typical distributed TDMA scheme [98] uses an omnidirectional antenna to obtain reservation slot schedule (i.e., sweeping overhead and delay are not considered) and an SBA in data traffic period.

The simulation setup is discussed below, followed by the comparison analysis for random flows and mobile nodes.

(A) Simulation Setup:

The simulations are run in MATLAB version R2017b for the network topology consisting of 50 nodes. Each experiment is run for 100 s and repeated 10 times. Here, the channel capacity is 10 Mbps, TTL is 0.5 s, packet size is 1000 Byte, and queue size is infinite. We consider a frame with 100 data slots and a slot length of 8.7 μs which allows the reception of ACK packet in the same data slot. We use Fig. 3.2 and Table 3.2 to determine the length of rank matrix and each control period in our scheme, respectively.

Note that our scheme can recompute reservation slots frequently because it has a very low reservation slot allocation overhead (see Table 3.2). On the other hand, the reservation slot allocation overhead is large for typical distributed TDMA scheme (e.g., the average running time of DSTO scheme for a 50-node network is 1 s in Fig. 3.14(b)). Therefore, we consider

a 5 s frame for the typical distributed TDMA scheme, which includes a reservation slot allocation period of 1 s and a 4 s data traffic period[5]. The reservation slot allocation period is used only at the start of the simulation in typical distributed TDMA scheme for the static scenarios where links do not break with time.

(B) Performance Comparison for Random Flows:

In this experiment, we randomly select 10 source-destination pairs for a static network of 50 nodes randomly placed in a 300 m x 300 m area. Here, the flow hop-count varies from 1 to 5.

Fig. 3.15(a) and 3.15(b) show the average PDR and end-to-end delay, respectively, for random flows with different hop counts and traffic densities. At 2 Mbps data rate, nodes do not experience congestion. Therefore, PDR of each flow is 100%. The end-to-end delay increases with the flow hop-count, which results in an increase in the number of packets dropped due to TTL expiry. Therefore, the flow PDR for each scheme degrades with the flow hop-count at all traffic densities.

Since the typical distributed TDMA scheme schedules data traffic slots on the FCFS basis, its reservation slot schedule remains the same for the simulation duration, which degrades the average network PDR. In addition, an intermediate node in the typical distributed TDMA scheme may not be able to forward the received packets when it schedules the majority of its data slots with its upstream node (e.g., PDR for the flow with 5 hop-count in typical distributed TDMA scheme is 0 at 4 and 5 Mbps data rates in Fig. 3.15(a)). Whereas, each node in our proposed scheme fairly distributes its data traffic slots among all the *desired throughput* requests by using the *throughput scaling* mechanism. Therefore, the total average

[5]The control overhead of the typical distributed TDMA scheme (i.e., DSTO [98]) decreases when a longer data traffic period is used. We observed the best performance for the typical distributed TDMA scheme at 5 s frame length.

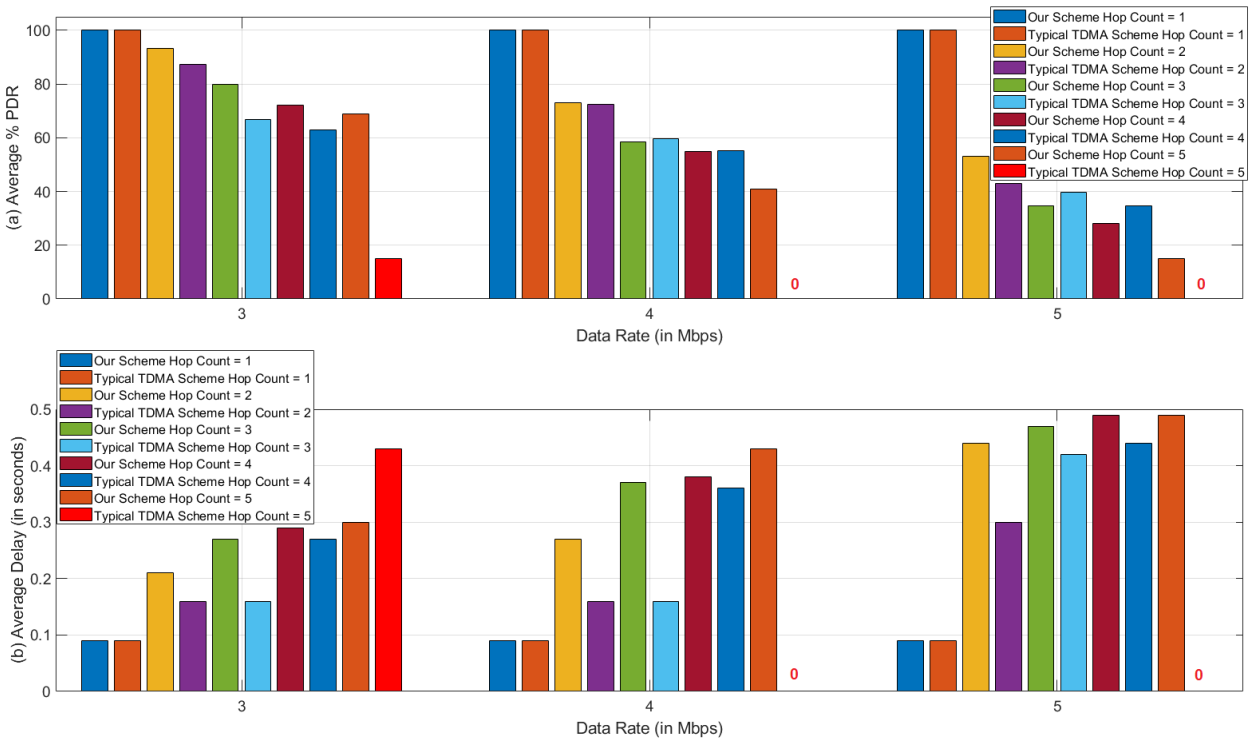


Figure 3.15: Comparison of the average (a) PDR and (b) end-to-end delay for a random static topology of 50 nodes at different traffic densities when the packet TTL is 0.5 s.

network PDR for all flows is higher in our scheme as compared to the typical distributed TDMA scheme.

A hotspot node in our scheme selects the flow(s) to serve based on the QoS-metric value of the packets stored in its queue, which increases the queuing delay of the packets of other flows, whereas a hotspot node always prefers the same flow in typical distributed TDMA scheme which leads to a lower delay for the packets of the selected flow. Therefore, the average end-to-end delay is generally higher in our scheme as compared to the typical distributed TDMA scheme at all traffic densities in Fig. 3.15(b).

(C) Performance Comparison for Static and Mobile Topologies:

In this section, we compare the performances of the schemes for (a) 420 m $\times$ 420 m static grid topology and (b) 1 km $\times$ 1 km mobile topology, where nodes move at (i) 2 m/s and (ii)

10 m/s, under the random-waypoint mobility model with zero pause time. Here, we select 25 flows randomly for each scenario, where each source node randomly selects a receiver node from its 1-hop neighborhood.

With a **static topology**, both schemes do not experience congestion at 2 Mbps data rate, and therefore, have 100% PDR. Since the typical distributed TDMA scheme reserves a slot on the FCFS basis and cannot accommodate multiple flows, its PDR is lower than our scheme in Fig. 3.16(a). However, the end-to-end delay of our scheme in Fig. 3.16(b) is slightly higher than the typical distributed TDMA scheme.

For a **mobile topology**, frequent link breaks lead to a high queuing delay, which increases the end-to-end delay and, thereby, the total packets dropped due to expiry of packet TTL. Therefore, the PDR is lower in Fig. 3.16(a) and the end-to-end delay is higher in Fig. 3.16(b) for mobile scenarios as compared to the static scenario. The PDR of both schemes further degrade as the node speed increases from 2 m/s to 10 m/s due to an increase in the link breaks. Since the typical distributed TDMA scheme takes a long time to adapt to the topology changes and recompute the reservation slot schedule, its PDR is lower than our

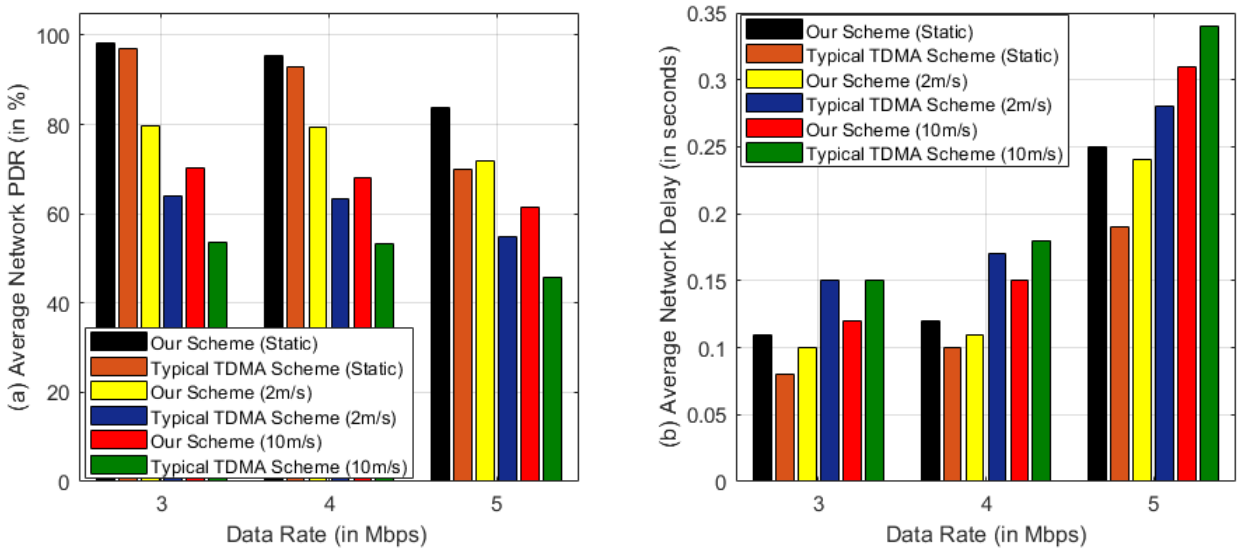


Figure 3.16: Comparison of average (a) PDR and (b) end-to-end delay of our scheme with the typical TDMA-based scheme (e.g., DSTO [98]) at different traffic densities and node speeds for a 50-node network topology, where the packet TTL is 0.5 s.

scheme at all traffic densities for mobile scenarios in Fig. 3.16(a).

Congestion in the network increases with the traffic density, which leads to a high delay and, therefore, lower PDR values for both schemes in static and mobile scenarios.

3.6 Conclusions

We presented a novel, real-time, distributed, and directional TDMA MAC scheme for multi-hop wireless networks. This scheme adapts to the topology changes and/or flow requirements in *real-time*, and facilitates QoS-aware communication with no *notification overhead*. In the proposed scheme, each node divides its 1-hop neighborhood into *fully connected 1-hop neighborhoods*, which allows the node to intelligently serve multiple routes without requiring a globally converged scheduling solution. This feature allows the use of a rank-based mechanism to obtain a real-time transmission schedule for a random multi-hop network.

We also added the following new features in the proposed scheme: (*i*) REQ period which reduces slot wastage, (*ii*) *throughput scaling* which ensures fairness, (*iii*) PR period which increases the spatial reuse and adapts to the dynamic requirements of multiple flows in real-time. The use of these features is optional, which allows a node to customize its frame based on the flow requirements and traffic conditions. Table 3.3 shows the usefulness of these features for different conditions, such as light, moderate, and high traffic loads, and **i**ndependent **r**outes (IR) with no hotspot node, as well as **r**outes **w**ith at least one **h**otspot node (RwH).

The control-period overhead and running time in our scheme are significantly low as compared to recent, distributed TDMA schemes, and linearly change with the number of nodes in a *fully connected 1-hop neighborhood*. Simulation results showed that our scheme achieved a high PDR and per node channel utilization ratio for real-time traffic, and has a superior

Table 3.3: Usefulness of Different Mechanisms for Different Flow Types and Traffic Conditions

Period/ Mechanism	**Light Traffic**		**Moderate Traffic**		**Heavy Traffic**	
	IR	RwH	IR	RwH	IR	RwH
Throughput Scaling	×	✓	×	✓	✓	✓
REQ Period	×	✓	×	✓	×	✓
Piggyback Reservation	✓	✓	✓	✓	×	×

performance over recent, distributed TDMA schemes at different traffic densities for static and mobile network topologies.

Chapter 4

A Proactive, Mobility and Congestion-Aware Routing Protocol

Due to the reduced cost of UAVs, their fast deployment, device autonomy and increased flight time capabilities, autonomous UAV networks can provide network reliability and fault tolerance, reduce mission completion time through collaboration, and adapt to dynamic application requirements [2–4,9–12]. However, the UAV networks experience varying network design configuration and communication constraints, which include the UAV density, speed and trajectory, and traffic rates.

The high node density and fast mobility result in a dynamic UAV network topology with frequent link disruptions, high co-channel interference, and significant control and computational overhead [2–4, 12]. As a result, designing a robust communication mechanism for autonomous, decentralized (with no supervisory node) UAV networks is very challeng-

The materials presented in Chapter 4 have been presented at the 2021 International Conference on Embedded Systems and Networks (EWSN) [47] and submitted to IEEE Trans. Aerospace & Electronic Systems (under 2nd round of review) [103].

ing [2–4, 9, 11, 12, 15–20, 104].

The widely used topology-based routing schemes require each node to maintain a routing table for its packet transmission, and recompute the routes periodically [2–4, 12]. However, these schemes are slow to adapt to topology changes in a UAV network. This degrades the flow throughput due to packet transmission over a broken route [15, 19]. Therefore, some topology-based routing schemes (e.g., [2–4, 9, 12, 17, 46, 104]) select a longer-lasting route and predict link-failure (or link-stability) to discard broken (or unstable) routes. Since the design and performance of a routing scheme depend on the underlying node mobility model, these routing schemes may not be suitable for practical autonomous UAV networks [2–4, 10, 12]. For the same reason, machine learning driven approaches to predict future node trajectories (e.g., [11, 19, 58]) are difficult to use in autonomous UAV networks due to uncertain node mobility patterns [2, 3, 50]. Another important issue is that existing routing schemes cannot distinguish whether a packet is lost due to link break or congestion. A few AN-specific routing schemes (e.g., [9, 15, 17, 45, 104]) address this issue by using the MAC layer information at the time of route selection. However, they do not consider the adverse impact of inter- and intra-flow interference and/or topology changes on the route quality *after* data transmission starts on the selected route.

As discussed in Section 2.2, a routing protocol for autonomous UAV networks should have the following characteristics to support the latency-constrained flows: (*i*) Low route discovery overhead, complexity and delay, (*ii*) anticipate the potential packet drops, identify the cause for the drop (link break and/or congestion), and take preventive measures such as route switching and queue management, (*iii*) consider practical node trajectories, and (*iv*) work with decentralized network topologies.

4.1 Introduction

In this chapter, we propose an adaptive, cross-layer, mobility and congestion-aware proactive routing protocol for decentralized, autonomous UAV networks to address the discussed issues. Its major contributions are:

1. A novel, multi-step and multi-metric route selection mechanism, which uses HC, RLT, estimated route latency and the inter- and intra-flow interference along the route. It selects a stable, longer-lasting and less congested route.

2. A preemptive route switching mechanism which prevents potential packet drops due to the congestion and topology changes. This improves the quality of service (QoS).

3. Routes are computed only for the *active source-destination pairs* rather than for each node in the network, significantly reducing route computation overhead.

4. A periodic queue management mechanism is used to prioritize transmission of packets which have a lower survivability score (i.e., a lower time-to-expiry or higher estimated time-to-destination), and discard the packets which are likely to expire before reaching the destination node.

5. The proposed routing scheme provides a significantly higher data throughput for delay-sensitive data flows at different data rates, node densities and speeds, as compared to standard OLSR and multi-metric OLSR (MM-OLSR) protocols.

We first discuss existing schemes for finding a stable, mobility-aware and congestion-free route in a dynamic AN in Section 4.2, followed by a description of our proposed cross-layer, proactive routing protocol (Section 4.3). We compare its performance empirically with the standard OLSR and MM-OLSR protocols in Section 4.4, and give conclusions in Section 4.5.

4.2 Related Work

Traditional proactive routing schemes compute a shortest hop route to the destination node. Unfortunately, shortest hop often selects nodes at the edge of the network, which are also often the highest mobility nodes and lead to frequent link breakage [9, 20]. To address this issue, in [20] a transmitter node excludes the fast-moving edge nodes from its 1-hop neighborhood by dynamically adjusting its effective communication area. In [59], social network-inspired criteria such as a node's connectivity degree are used for the next-hop selection to reduce the chance of packet loss at an edge node. A hybrid, Q-learning-based routing approach is proposed in [15], which proactively selects a shortest hop route when the network topology is stable, and sends duplicate data packets over multiple paths to reliably deliver them to the destination node. However, shortest-path routing schemes often do not find a long-lasting and low congestion route.

For this reason, multi-metric routing schemes are often preferred for dynamic networks. One mobility and delay aware OLSR scheme [17] uses a Kalman filter to predict the link lifetime of 1-hop neighbor nodes. Then, a transmitter node selects among the shortest hop routes passing through its longer-lasting neighbor nodes to identify a route with low path latency. In [45], the transmitter node makes sequential binary decisions to find an energy-efficient and least-congested route passing through its most stable 1-hop neighbor nodes. In the Q-learning-based routing scheme [18], a transmitter node uses the queuing delay and energy consumption of its 1-hop neighbor nodes in the reward function, and adapts to the dynamic topology changes by adjusting its learning rate and discount factor. Note that all these schemes [17, 18, 45] select a route based only on the local network characteristics at the 1-hop neighbor nodes and do not consider the link stability of the downstream nodes of the selected route. Therefore, they can experience high latency and low throughput due to packet rerouting at intermediate nodes, when downstream link(s) break due to mobility [1–3, 18].

More holistic routing schemes compare the network characteristics of all the routes between a source-destination pair before selecting a route. The link quality and traffic load aware OLSR scheme of [9] differentiates links based on their RSSI values using the Chebyshev inequality. It also considers buffer occupancy (BO) values of all intermediate nodes to select a route with low congestion. In [14], a normalized weighted sum of the estimated link quality, movement direction, node stability, residual energy, HC and latency for each route is considered in the route selection. In the source routing scheme of [46], the complete route is inserted into the data packets by the source node. It uses a link failure prediction mechanism by which an intermediate node notifies the source node before its link breaks; this allows the source node to initiate a new route discovery before the current route breaks. In [7], a centralized controller is used to select a short-distance, low-traffic, small-backlog and low-hop route for a flow using the node distance, packet arrival rate, BO and HC metrics.

However, none of these existing schemes consider the effect of intra-flow interference on the selected route, which increases route congestion. For example, routing metric values (such as BO, path load and latency) change significantly *after* the data transmission starts on the selected route, which degrades the route quality and flow throughput.

In this chapter we propose a cross-layer, **m**obility and **c**ongestion-**a**ware OLSR (MCA-OLSR) protocol, which is specifically designed for decentralized autonomous ANs. It addresses all of the discussed challenges, and significantly improves the QoS, especially at high node speeds (see Section 4.4.2 for details).

4.3 Description of Proposed Cross-Layer MCA-OLSR Protocol

We discuss our network model and assumptions in Section 4.3.1, followed in Section 4.3.2 by the novel features of our proposed routing scheme: Topology-aware routing table construction, routing metric selection, robust route selection, preemptive route switching, and a proactive queue management mechanism. Then, we discuss the control overhead and computational complexity of our proposed scheme in Sections 4.3.3 and 4.3.4, respectively.

The modules of our proposed MCA-OLSR scheme are shown in Fig. 4.1. Our scheme uses modified Hello and TC control packets, which include node and link statistics such as LLT, number of interfering links (IL) and packet service time (PST). Here, LLT is the time duration after which the link is predicted to break; IL of a node is the number of its links with 1-hop neighbors that are contending for the channel for their data transmission; and PST is the duration a data packet stays in a node's MAC queue. A node obtains its updated IL and PST values and the flow(s) it serves from the MAC layer.

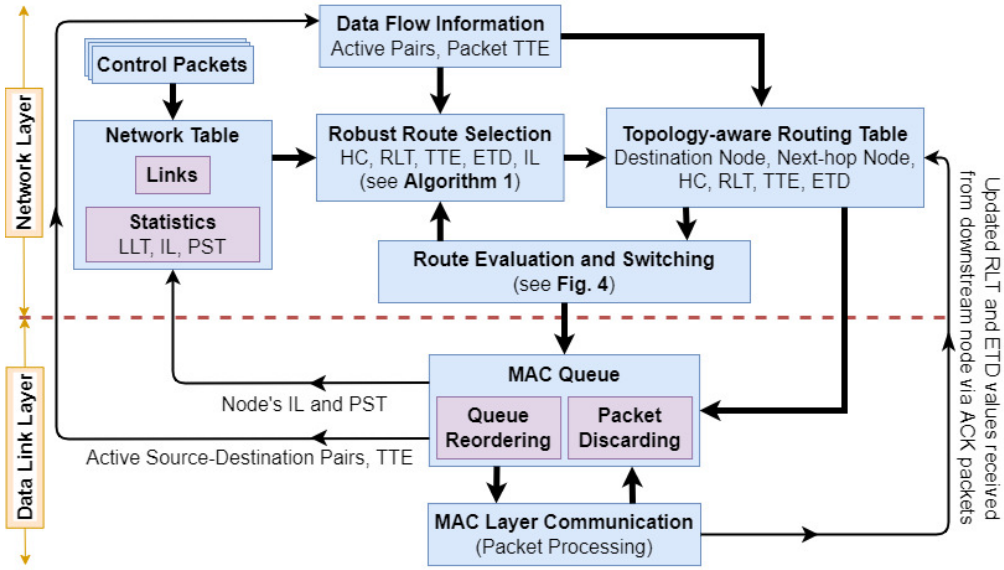


Figure 4.1: Modules in our proposed MCA-OLSR scheme.

Using the updated LLT values received via the control packets, a node removes the broken links from its *Network Table* (discussed in Section 4.3.2). In addition, it uses the cross-layer information from its *Network Table* and MAC layer to compute the HC, RLT, estimated time-to-destination (ETD) and IL values of a route, and then selects a long-lasting and low congestion route for each flow it serves (discussed in Sections 4.3.2 and 4.3.2).

A node stores the selected route in its *Routing Table* and tracks its quality, which varies due to the congestion buildup and/or topology changes in the network. A route switch is triggered if the route quality degrades below a threshold (discussed in Section 4.3.2(D)). Further, a node periodically rearranges its MAC queue using a two-step queue management policy: (i) queue reordering to prioritize transmission of packets with a low survivability score, and (ii) discarding packets that cannot be delivered to the destination node before their expiry (discussed in Section 4.3.2(E)). These features significantly improve the performance of our proposed MCA-OLSR scheme as compared to the OLSR and MM-OLSR protocols, as demonstrated via simulations in Section 4.4.

4.3.1 Network Modeling and Assumptions

Airborne UAV nodes can be categorized into fixed-wing (FW) and rotatory-wing (RW) UAVs [2,10,59]. For dynamic tasks, FW-UAVs are preferred as they can attain higher speeds with a longer flight time due to their better aerodynamic design, which gives them stability against harsh environmental factors such as air drag [2, 10, 59]. Therefore, we consider a network of low SWAP (size, weight and power) FW-UAVs [2, 10, 105].

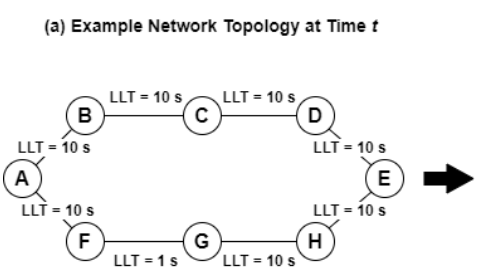
The design and evaluation of routing protocols for ANs require mobility models that can produce realistic node movements by considering their aerodynamics [1–4, 10]. FW-UAVs cannot make sharp turns due to their aerodynamics and high speeds. Therefore, we use a smooth-turn mobility model [105] in which each node independently selects a center and

radius based on its past trajectory, and rotates around the center in a clockwise or counter-clockwise direction for a randomly selected duration. A very large radius results in a straight trajectory.

We assume that each node broadcasts its trajectory information (i.e., GPS location, movement (i.e., clockwise, counter-clockwise or straight), center and radius) to its 1-hop neighbors. A node can accurately compute the LLT value for each of its 1-hop neighbor nodes using the mathematical formulation described in [47, 50]. A UAV pair uses the current trajectory information to compute its LLT value when the link is first established, and then updates the LLT value when either UAV in the pair changes its trajectory. In our scheme, the LLT value of a link is included in the control message, which is broadcast periodically in the network.

4.3.2 New Enhancements in Our Proposed Scheme

Routing a data packet in the OLSR protocol is shown in Fig. 4.2. Each node in Fig. 4.2(a) maintains a *Network Table*. Here, node A uses its *Network Table* (shown in Fig. 4.2(b)) to compute a shortest hop route to each destination node, whenever it receives a new control



(b) Network Table at Node A at Time t

Link	Obtained From	Validity Duration
A - B	$N^1(A)$	6 s
A - F	$N^1(A)$	6 s
B - C	$N^2(A)$	6 s
F - G	$N^2(A)$	6 s
G - H	Topology Set (A)	15 s
C - D	Topology Set (A)	15 s
D - E	Topology Set (A)	15 s
H - E	Topology Set (A)	15 s

(c) Routing Table at Node A at Time t

Destination Node	Next-hop Node	Distance (in Hops)
B	A	1
F	A	1
C	B	2
G	F	2
H	F	3
D	B	3
E	F	4

Figure 4.2: Network topology at time t is shown in (a), where the LLT value of each link is shown. Note that link F-G will break at time (t+1) s. The *Network Table* at node A at time t in the OLSR protocol is shown in (b). The routing table at node A is shown in (c).

packet. These routes are stored in its *Routing Table* shown in Fig. 4.2(c). A node stores the data packets in its MAC queue in FIFO order, where each packet waits for its transmission.

(A) Topology-aware Routing Table Construction:

As discussed in Section 2.2.2, each entry in the *Neighbor* and *Topology Sets* in the OLSR protocol has a default validity duration of 3× the Hello interval and 3× the TC interval, respectively [54]. In OLSR, a node uses the *Neighbor* and *Topology Sets* to compute the shortest hop route to the destination node. Since the links can frequently break due to high node mobility in ANs, and OLSR cannot immediately adapt to link breaks, OLSR may select a broken or short-lived route, which degrades flow throughput. To address this issue in our proposed routing scheme, the LLT value of a link is used as the validity duration in the *Network Table*, which prevents the inclusion of obsolete links in the route selection mechanism.

For example, node A in Fig. 4.2(a) has two shortest hop routes for node E (i.e., A-F-G-H-E, A-B-C-D-E). If route A-F-G-H-E is selected by OLSR, node A may send data packets to the destination node E via node F during its validity duration, (t, t+6) in Fig. 4.2(b), until the order of the entries in its *Network Table* changes. However, route A-F-G-H-E breaks at time (t+1) because the LLT of link F-G is only 1 s, resulting in packet transmission over a broken route. In our proposed MCA-OLSR scheme, the *Network Table* at node A at time (t+1) does not include the F-G entry. Node A uses route A-B-C-D-E, which has a minimum LLT of 9 s at time (t+1), to send its data to destination node E; this avoids transmission over broken links.

(B) Selection of Routing Metrics:

In the OLSR protocol, a node computes a shortest hop route for each node in the network using Dijkstra's algorithm. Besides HC, the RSSI, RLT, BO and ETD metrics have been used for route cost computation in the literature [7, 9, 11, 14, 17, 18, 45, 47, 51, 59]. However, significant interference from neighbor nodes in a dense network can cause inaccurate computation of RSSI values [46]. The RLT value can change because of the uncertain node movements in ANs [50], and the values of BO and ETD can increase significantly after the source node starts data transmission on the selected route. Therefore, we find that these routing metrics are not sufficiently reliable.

Our scheme computes a longer-lasting and low congestion route where HC and IL are used for the route cost metric, while RLT and ETD are used for route evaluation and switching after data transmission starts on the selected route. Note that the value of the IL metric of a route changes only when the local topology changes for the nodes participating on the route. This means that our proposed route quality changes less frequently than existing multi-metric schemes. If the topology changes or congestion buildup reduces the route quality, a preemptive route switching mechanism is used (see Section 7).

To define our metrics more precisely: the hop count, HC, is the route length in terms of number of links. The route lifetime, RLT, of a route R is the time duration after which the route is likely to break, and is computed at node i as [51],

$$RLT_R(i) = \min_{link\ l \in L_{R^i}} (LLT_l), \tag{4.1}$$

where L_{R^i} represents a set of links on route R that connect node i to the destination via its downstream nodes, and the lifetime of each link l, LLT_l, is computed by using the node location and trajectory [47, 50].

The estimated time to destination, ETD, at node i is the total estimated delay a packet will experience while traveling from node i to the destination on route R. It is computed as [14],

$$ETD_R(i) = \sum_{j \in R^i} \left(\frac{1}{P_j} \sum_{p \in P_j} (PST)_p \right)_j, \tag{4.2}$$

where R^i are the nodes on route R from node i to the destination node, and P_j is the set of data packets successfully transmitted by node j during the previous Hello interval. The PST of a packet p is the duration for which it stays in the MAC queue of node j (i.e., from the time it enters the queue until it is forwarded to the next hop node and an ACK packet is received).

Finally, the interfering link score at node i, $IL_R(i)$, is the sum of the number of interfering links at all nodes from node i to the destination on route R [45],

$$IL_R(i) = \sum_{j \in R^i} \left(IL_j^{\theta} + IL_j^{\phi} \right), \tag{4.3}$$

where R^i are the nodes on route R from node i to the destination. Here, IL_j^{θ} represents the recent IL value of node j received via the control packet, and IL_j^{ϕ} represents the new intra-flow interfering links that will be created on node j when the data transmission starts on route R. Note that each intermediate node in route R can create up to two new intra-flow interfering links – one with its upstream node and another with its downstream node. However, these intra-flow interfering links are considered only when they have not already been included in the IL_j^{θ} value of node j in (4.3). To identify the links in IL_j^{θ}, node j includes an $isLinkActive \in \{0, 1\}$ value for each of its 1-hop neighbor nodes; the value of $isLinkActive$ is 1 if the link is already being used for data transmission and is otherwise 0.

In its Hello packet, each node includes the PST and IL values for itself and its 1-hop neighbors, along with the LLT values of links with its 1-hop neighbor nodes. Similarly, each

MPR node includes these values for each of its MPR selector nodes in its TC packet. Then, the *Network Table* is constructed at each node as discussed in Section 4.3.2.

(C) Robust Route Selection:

In our scheme, the routes are computed only for the *active source-destination pair(s)*[1] instead of all the nodes in the network. This significantly reduces the route computation overhead and the size of the *Routing Table* as compared to the OLSR [7, 47, 51].

Using a breadth first search (BFS) algorithm, a candidate route set $C(i)$ is constructed for a flow at node i, which contains all the routes between node i and the destination node. A two-step process is then used for route selection:

Step 1: Construct a set of routes $C^*(i) \subseteq C(i)$ such that each route satisfies both of the following constraints:

$$RLT_R(i) \; > \; TTE(i) \; + \; \delta \tag{4.4a}$$

$$\frac{TTE(i)}{ETD_R(i)} \; \geq \; \epsilon_1 \tag{4.4b}$$

Here, $RLT_R(i)$ is the residual route lifetime of route R and $TTE(i)$ is the time-to-expiry value of the head of line (HOL) packet of the flow at node i at a given time; ϵ_1 is a constant; and δ is a control parameter, discussed next.

In a heavily loaded and dynamic network, the congestion can worsen and topology can change significantly, even within one TC interval. In such situations, using a route for the entire TC interval can degrade the flow throughput. On the other hand, reducing the TC interval can significantly increase the control packet overhead. To address this issue, each

[1] An active source-destination pair consists of a source node actively transmitting data packets to the destination node.

active destination node d includes its PST value in the ACK packet after duration δ. When an intermediate node i of a route receives the PST_j value from its downstream node j, it includes the $(PST_i + PST_j)$ value and the RLT value of the route (from node i to node d) in the ACK packet which is sent to its upstream node. As a result, the source and intermediate nodes periodically receive updated values of ETD and RLT for the current route, and so can quickly flag a route switch when needed (see Section 7 for details about route switching). Therefore, the RLT of a route must be at least δ longer than the TTE value, to prevent data transmission over a broken route (Constraint (4.4a)).

The empirically selected values of δ and ϵ_1 are $\frac{1}{2}\times$ the Hello interval, and 1.5, respectively. Hence, the route selection mechanism considers only those routes which (i) will not drop packets due to a link break (Constraint (4.4a)) and (ii) the TTE of packets is $\geq 1.5 \times ETD$ (Constraint (4.4b)).

Step 2: Compute the cost of each route $R \in C^*(i)$ at node i as shown in (4.5), and then select the lowest-cost route R^*:

$$\begin{aligned} Cost_R(i) &= w_1 \left(\frac{HC_R(i)}{HC_{min}} \right)^E + w_2 \left(\frac{IL_R(i)}{IL_{min} + \alpha \times IL_R(i)} \right), \\ \text{where} \qquad E &= \begin{cases} n_i \times \log(n_i), & \text{if } n_i \geq 2; \\ 1, & \text{otherwise.} \end{cases} \end{aligned} \tag{4.5}$$

Here, $(\cdot)_{min} = \min\limits_{\forall R \in C^*(i)} (\cdot)_R$, and n_i is the shortest hop distance between the source node and node i. The exponent E penalizes downstream nodes for selecting a longer hop route to the destination node. The scaling factor α is used to convert the IL cost values to the same range as HC. The weights of the normalized HC and IL metrics are w_1 and w_2, respectively, where $w_1 + w_2 = 1$. In our experiments, we take $w_1 = w_2 = 0.5$, and $\alpha = 0.3$. The pseudo code to select a long-lasting and less-congested route is given in Algorithm 5.

Algorithm 5: Pseudo Code for Route Selection at Node i

Input: Network topology known at current node i, active source-destination pair (s-d), MAC queue of node i
Output: Return the selected route R^*
$C(i)$ = Set of routes between node i and node d
Remove routes from $C(i)$ that can drop packets before their expiry (i.e., Constraint (4.4a))
Remove routes from $C(i)$ with route latency higher than packet's remaining lifetime (i.e., Constraint (4.4b))
Compute route cost for the remaining routes using (4.5)
$R^* = \underset{R \in C(i)}{\arg\min}(Cost_R(i))$

(D) Preemptive Route Switching:

The congestion along a route can build up after a source node starts data transmission. Obtaining an accurate prediction of congestion buildup at the time of route selection is difficult in a decentralized, mobile network topology, where each node independently selects a route for each of the flows passing through it. Therefore, a node in our proposed scheme continuously monitors the active routes, and proactively switches to a new route when any one of the following three conditions occur:

$$RLT_{R^*}(i) \; < \; TTE(i) \; + \; \delta \tag{4.6a}$$

$$\frac{TTE(i)}{ETD_{R^*}(i)} \; < \; \epsilon_2 \tag{4.6b}$$

$$HC_{min} \; > \; HC_R(i) \tag{4.6c}$$

The conditions (4.6a) and (4.6b) correspond to violations of the selection constraints (4.4a)–(4.4b). We take $\epsilon_2 = 1.1$, i.e., the selected route R^* is used as long as the packet $TTE \geq 1.1 \times ETD$. Since links break and new links form frequently in ANs, condition (4.6c) allows a node to initiate a new route discovery whenever a new, shorter length route R becomes available. Then, the next hop node address of the data packet of a flow stored at the MAC queue is updated. The flowchart for the route switching module is given in Fig. 4.3.

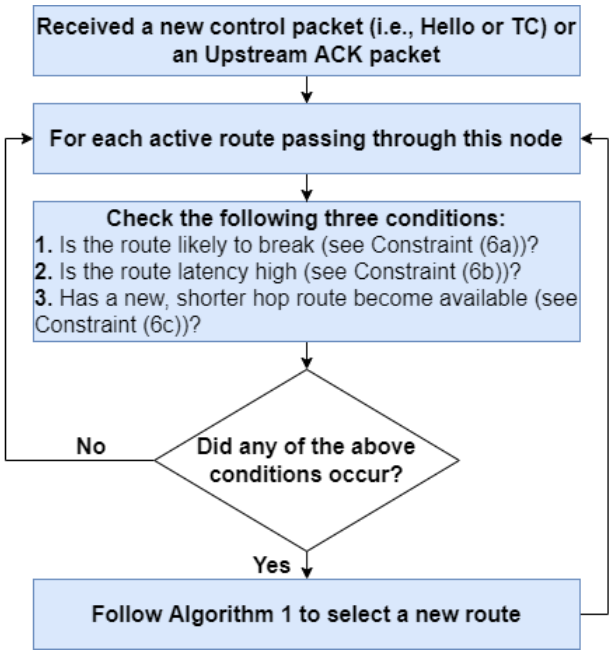


Figure 4.3: Flowchart of the route switching module in our proposed scheme.

(E) Proactive Queue Management:

The congestion at a node can increase due to the high data rate, inter- and intra-flow interference and frequent link breaks, which increases the number of packets in the queue and packet queuing delay. As a result, the packets are dropped due to the TTL expiry and buffer overflow. Some existing congestion prevention schemes, such as WRED (weighted random early detection) [106] and TSDP (two-step discarding policy) [107], discard packets before the queue is full, but this only partially addresses congestion buildup. These schemes do not consider the packet TTE and ETD, and therefore cannot identify which packets are likely to expire before reaching their destination. Further, these schemes do not reassess the status of the packets in the queue but simply transmit them in FIFO order.

To address these issues, our scheme uses the following two-step queue management policy:

Step 1: Instead of using the default FIFO order, the packets of the queue at a node are rearranged in the ascending order of their survivability score, computed as $\frac{TTE}{ETD}$. Thus,

packets with a low survivability score are prioritized for transmission, which increases their chances of reaching the destination node before their TTL expiry.

Step 2: Packets with a survivability score lower than a threshold are dropped. These packets likely cannot be delivered to the destination node before their expiry due to the limitations imposed by the current network topology and network traffic. However, it is possible that a new, shorter route with low ETD could become available later due to the frequent network topology changes. Therefore, our proposed scheme discards only from the first N packets starting from the HOL. In practice we select threshold 0.7 and $N = 50$.

Each node periodically reevaluates its queue using this two-step queue management policy.

4.3.3 Control (Signaling) Overhead

In addition to the information sent in the OLSR protocol, each node in our proposed scheme also sends the following information to its 1-hop neighbor nodes.

- Its own GPS location (uses 6 bytes for (x,y,z) coordinates) and trajectory information, which includes the center coordinates (6 bytes), and radius and node movement direction (2 bytes).
- Its own PST and IL values (1 byte).
- The LLT, PST, IL and $isLinkActive$ values for each 1-hop neighbor node advertised in its control packet (2 bytes).

Note that a node includes its GPS location and trajectory information only in its Hello packet. Further, a node sends its trajectory information only when it forms a new link or changes its current trajectory. The remaining fields are included in both Hello and TC packets.

4.3.4 Route Computation Complexity

In the OLSR protocol, every node computes a shortest hop route to all other nodes in the network using Dijkstra's algorithm, when it receives a new control packet. However, in our proposed scheme, only the nodes participating in data transmission compute a route to the *active* source-destination pair using the BFS algorithm, whenever a new flow passes through them or a route switch is flagged. This significantly reduces the number of route computations as compared to OLSR. For a network of V nodes and E links, the time complexity of finding the shortest hop route in Dijkstra and BFS are $O(E \log(V))$ and $O(E + V)$, respectively. The worst-case time complexity of OLSR and our proposed scheme are $O(T\,V\,E\,\log(V))$ and $O(T'\,V^2\,(E+V))$, where T and T' are the number of route computations in OLSR and our scheme, respectively.

In addition to the route selection and switching mechanisms, a node in our scheme also reassesses its MAC queue periodically (as discussed in Section 4.3.2), which has worst-case time complexity of $O(S\,\log(S))$ for queue reordering (Step 1) and $O(N)$ for packet discarding (Step 2), where S is the total number of packets the MAC queue can store.

4.4 Simulation Setup and Results

We have implemented the MCA-OLSR and MM-OLSR routing schemes in the discrete event simulator, NS-3 version 3.29. The MM-OLSR protocol compares the link and node statistics of all routes between a source-destination pair [7, 9, 14, 51, 59]. It uses three widely used metrics – normalized HC, normalized RLT and normalized load capacity – to compute a long-lasting and less congested route.

Our simulation and network parameters are summarized in Table 4.1. In our simulation, fixed-wing UAVs fly in a 8×8 km^2 area using the smooth-turn mobility model [105]. The

Table 4.1: Simulation Parameters

Parameters	**Values**
Simulation Area	$8 \times 8\ km^2$
Channel Rate	11 Mbps
Transmission Range	1 km
Node Density	50, 100, 200
Node Speed (in m/s)	20, 30, 40, 50
Number of Traffic Flows	1, 3, 10, 20, 40
Flow Rate	40 kbps to 3.5 Mbps
Packet Size	1 kB
TTL	3 s
Simulation Duration	100 s, 200 s

performance is evaluated for different network settings: moderate to high node densities (50, 100 and 200 UAV nodes), low to high network loads (i.e, 40 kbps to 8 Mbps), and 20 m/s to 50 m/s (minimum stalling speed to high) UAV node speeds. Here, a line-of-sight communication is assumed and the channel fading and noise are ignored.

The channel rate is 11 Mbps and each node has a 1 km transmission range. The packet size and TTL values are 1 kB and 3 s, respectively, and the MAC queue can store up to 1000 packets. The MAC layer protocol is CSMA/CA (carrier sense multiple access with collision avoidance) protocol. Each simulation is run for 100 s and each experiment is repeated 40 times, with the source-destination pair(s) of a flow selected randomly in each run. We consider two different types of traffic: video streams and sensor data. Each source node generates traffic at a constant bit rate.

4.4.1 Performance Metrics

We compare the performance of our proposed cross-layer MCA-OLSR scheme to standard OLSR and MM-OLSR, evaluating the three protocols in terms of four performance metrics:

1. The **PDR** of a flow is the ratio of total data packets received at the destination node to the total data packets generated at the source node. PDR represents a normalized throughput, and is equivalent to several other common metrics; for example, the *flow throughput* can be computed as PDR $\times$ data rate, while the *packet loss ratio* equals 1-PDR.

2. **End-to-end delay** is the average time taken by all the data packets to travel from the source node to destination node.

3. **Normalized control (signaling) overhead** is the ratio of the size of control packets transmitted during the simulation duration (averaged over 100 simulation seconds) to the total traffic load in the network.

4. **Average number of routes computed** is the total number of routes computed during the simulation, averaged over the simulation duration (100 s).

4.4.2 PDR Comparison

Here, we discuss the PDR comparison of all three schemes. For video traffic, Section 4.4.2 compares the PDRs for one and three flows at varying data rates (0.5 to 3.5 Mbps), two different node densities (50 and 100 nodes) and two speeds (20 m/s and 50 m/s). For sensor data, Section 4.4.2 compares PDRs as the number of data flows is varied from 1 to 40 at two different data rates, 5 packets/s (40 kbps) and 25 packets/s (200 kbps), for different node densities and speeds.

(A) Video Traffic

First, we compare the instantaneous PDR performance of the three schemes. Then, we discuss the impact of number of data flows and data rate on the average PDR performance, followed by the impact of node speeds and densities. When plotting average PDR, we also indicate the 95% confidence interval (shaded region).

(i) Comparison of Instantaneous PDR Performance:

Fig. 4.4 shows the instantaneous PDR values for each scheme for one simulation setting: 100 nodes flying at 50 m/s speed and a single data flow at 1.5 Mbps. The simulation is run for 100 s, and the initial 25 s are used for network stabilization. Since the packet TTL value is 3 s, a destination node can receive data packets generated up to 3 s prior to the current time, so that the instantaneous PDR value can sometimes exceed 1.

Unlike OLSR and MM-OLSR, our proposed MCA-OLSR scheme provides a consistently higher instantaneous PDR because it quickly adapts to the dynamic network characteristics by anticipating the congestion buildup and link breaks. In fact, our scheme ensures an uninterrupted communication throughout the simulation duration despite a highly dynamic network topology. Instantaneous PDR plots for the other simulation settings are omitted for space, but also showed superior performance over OLSR and MM-OLSR.

(ii) Impact of Number of Data Flows and Data Rates:

Fig 4.5 shows the PDR of each scheme at one and three flows across varying data rates (up to the point that the network becomes congested). Our proposed MCA-OLSR scheme selects stable and less congested routes, preemptively switches to a new route when the current route becomes unsuitable due to the topology change(s) and/or congestion buildup, and uses a proactive queue management policy, giving it a much higher average PDR than standard OLSR and MM-OLSR.

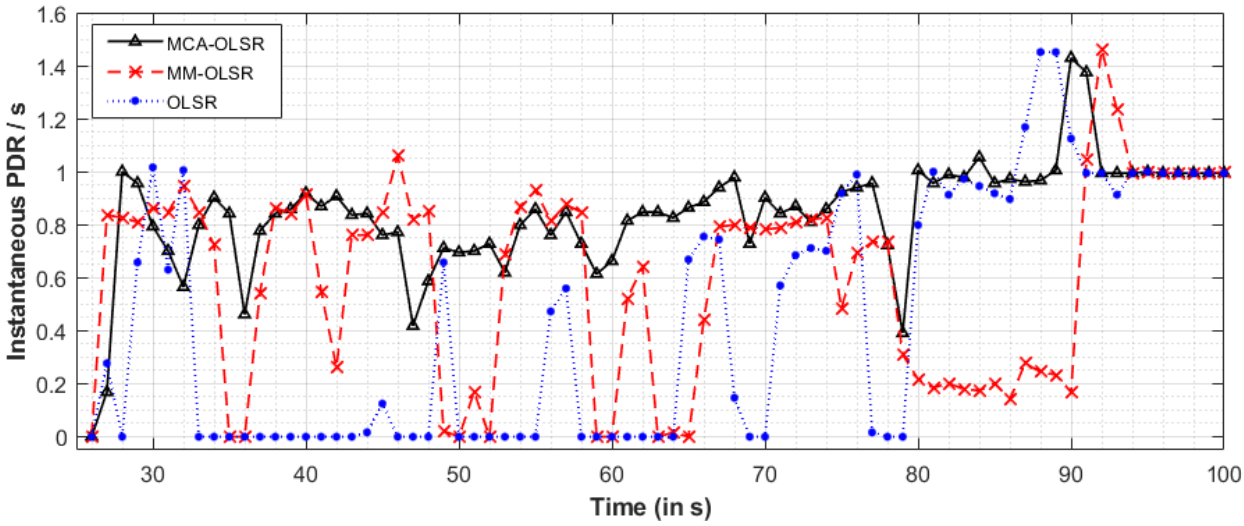


Figure 4.4: Instantaneous PDR comparison for our proposed MCA-OLSR, MM-OLSR and OLSR schemes, when **100** nodes fly at **50 m/s** speed and **one** traffic flow generates data at **1.5 Mbps**.

Fig. 4.5(a) shows PDR performance for 100 nodes at 50 m/s. Our MCA-OLSR achieves a significantly higher PDR (up to 149% and 59% at the low and high data rates, respectively) as compared to OLSR. Our scheme also achieves up to 26% and 50% higher PDR at the low and high data rates, respectively, as compared to MM-OLSR. At a high node speed, the number of connected components in the network increases due to frequent topology changes, which results in situations where no route is available temporarily. This increases the number of packets dropped due to TTL expiry. Further, the congestion increases with the data rate and/or number of flows. Therefore, the PDRs of all the schemes decrease as the data rate increases and the network becomes heavily congested at 2.5 Mbps (for one flow) and 1.25 Mbps data rates (for three flows).

Fig. 4.5(b) shows a similar situation for 50 nodes flying at 50 m/s. Here, MCA-OLSR achieves up to 100% and 40% higher PDR compared to OLSR and MM-OLSR, respectively. The decreased node density means fewer short, long-lasting routes are available, which reduces all methods' PDR compared to the setting with 100 nodes.

Fig. 4.5(c) show the PDR for 100 nodes flying at only 20 m/s. The PDR performance of MCA-OLSR is up to 19% and 34% higher as compared to OLSR and MM-OLSR, respectively.

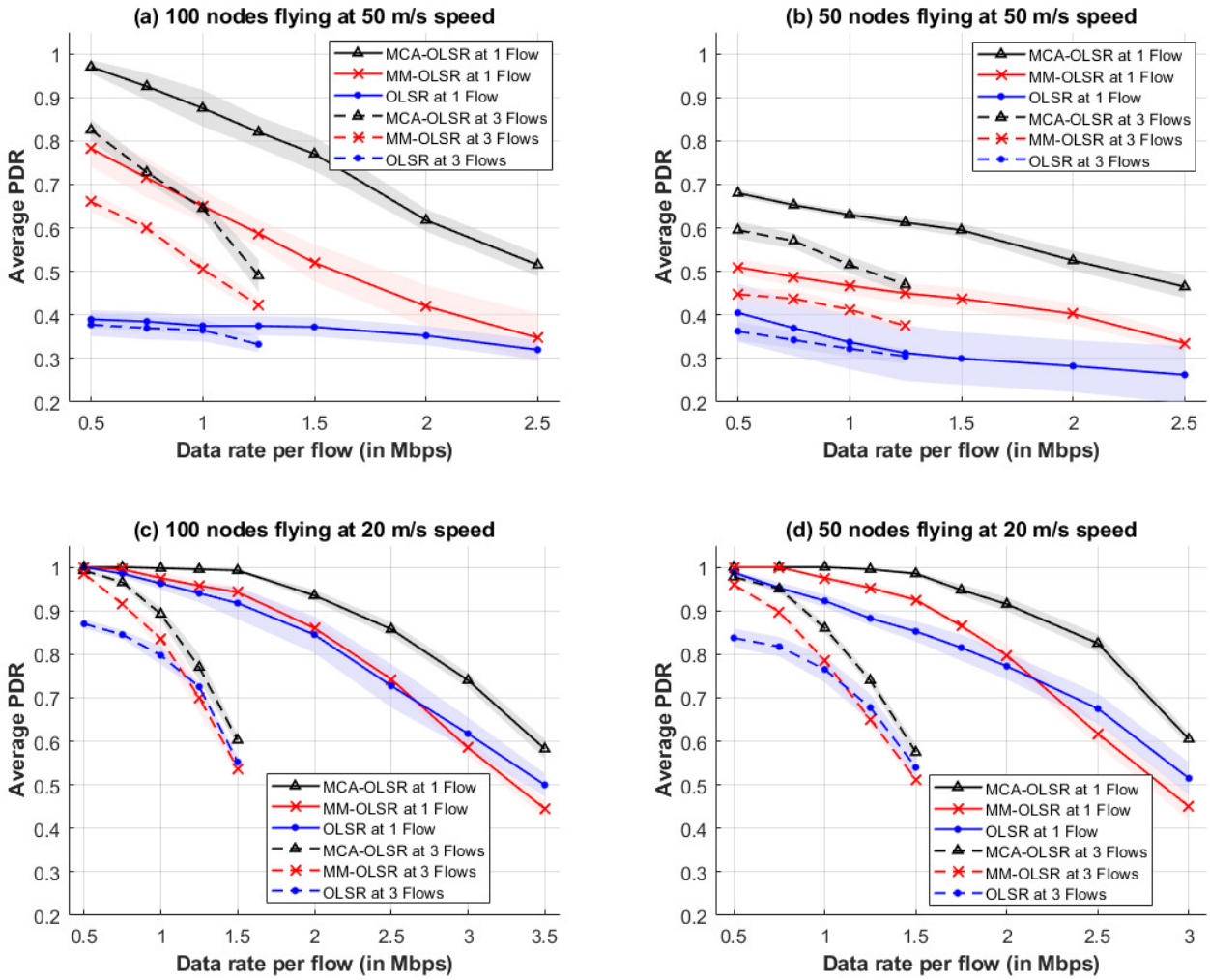


Figure 4.5: PDR comparison for our proposed MCA-OLSR, MM-OLSR and OLSR schemes for different number of data flows and data rates, for varying node density and speed. The subplots (a), (b), (c) and (d) have different ranges on the X axis.

Compared to Fig. 4.5(a), the lower node speed results in fewer disconnected components and thus fewer packes lost to TTL expiry, giving higher PDR values. Similarly, Fig. 4.5(d) shows 50 nodes at 20 m/s. Again, we see that MCA-OLSR's PDR is up to 21% and 34% higher than OLSR and MM-OLSR, respectively. At heavy congestion, the use of multiple routing metrics in MM-OLSR leads to a longer hop route than OLSR, increasing packet drops due to their TTL expiry. Our proposed scheme addresses this issue by preemptively switching from the congested route.

(iii) Impact of Node Speed

Fig. 4.6 examines the impact of different node speeds on PDR. At higher speed, there are more frequent topology changes and more (dis)connected components, resulting in situations where no route is available for a flow and decreasing PDR performance. Our MCA-OLSR constructs a topology-aware *Network Table* and uses the *RLT* metric in the route selection and switching. In contrast, OLSR always selects a shortest hop-count route; this often selects edge nodes, which result in frequent link breaks. In addition, OLSR and MM-OLSR are slow to update their *Network Tables* (see Section 4.3.2), and so may use broken routes for data transmission. Thus as node speed increases, MCA-OLSR's PDR decreases less than that of OLSR and MM-OLSR. In our experiments, MCA-OLSR achieves up to 149% and 50% higher PDR compared to OLSR and MM-OLSR, respectively, at the high speed of 50 m/s.

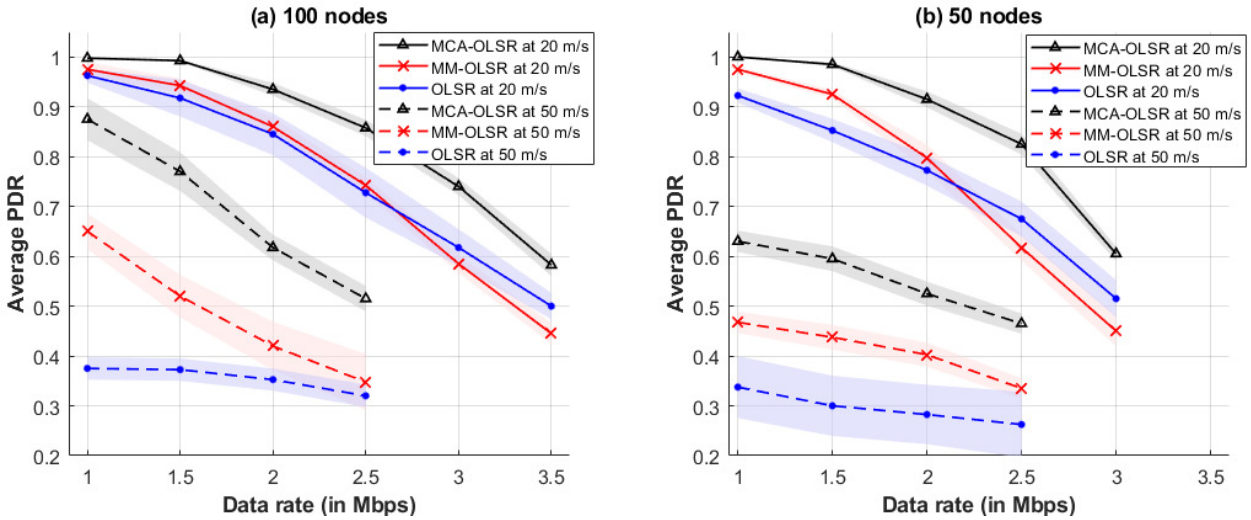


Figure 4.6: PDR comparison for our proposed MCA-OLSR, MM-OLSR and OLSR schemes at different node speeds, when the number of flows is **one** and node density is (a) **100** and (b) **50**.

(iv) Impact of Node Density

Fig. 4.7 illustrates the impact of node density on PDR performance. At 50 m/s, an increase in the node density from 50 to 100 nodes increases the number of available short, stable routes, which improves the PDR achieved by MCA-OLSR scheme by up to 39% (see Fig. 4.7(a)). In contrast, at 20 m/s the network topology already stays connected, and the gain in PDR due to higher node density is small (Fig. 4.7(b)).

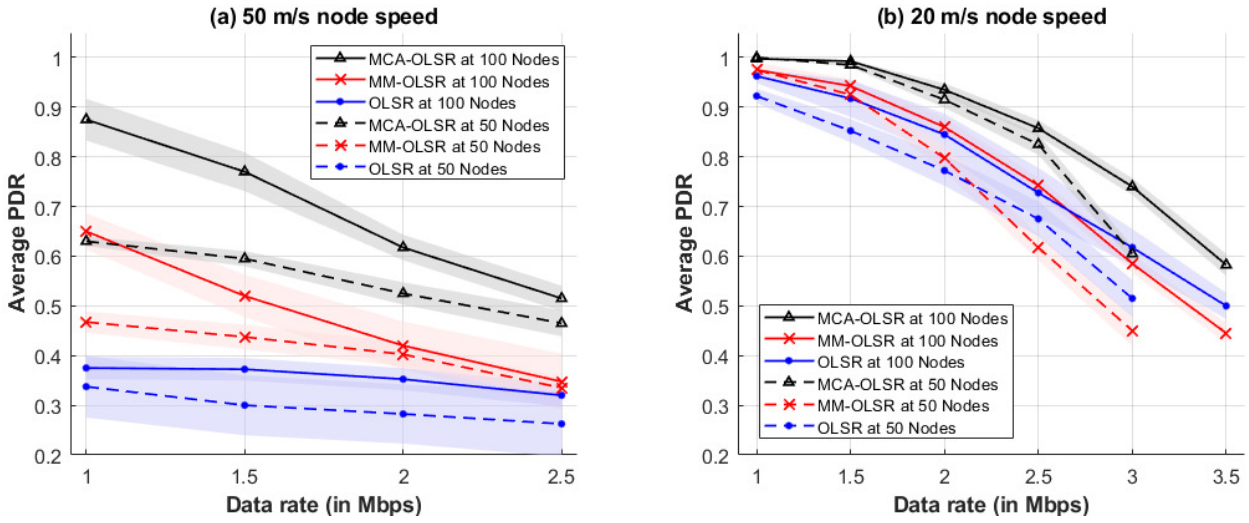


Figure 4.7: PDR comparison for our proposed MCA-OLSR, MM-OLSR and OLSR schemes at different node densities, when the number of flows is **one** and node speeds are (a) **50 m/s** and (b) **20 m/s**. Note that the subplots (a) and (b) have different ranges on the X axis.

(B) Sensor Data

The average PDR values for sensor data in all three schemes for varying node densities and speeds, and network loads are shown in Fig. 4.8. The MCA-OLSR provides higher PDR values as compared to the standard and MM-OLSR schemes except at very low network load and node speed (where the PDR values of all three schemes are almost the same). This is because MCA-OLSR uses robust route selection and preemptively switches to a better route when the current route's quality degrades. As the network load and/or node speed increase, the PDR values in all three schemes decrease because the changes in the network topology and/or congestion buildup increase the number of packets dropped. The performance of all three schemes improves at a higher node density for a given value of network load and node speed.

The impact of simulation duration on PDR performance for 50 nodes is shown in Table 4.2 for different node speeds and number of flows for the standard OLSR and MCA-OLSR schemes. At low node speed, the average PDR values in both schemes decrease when the simulation duration is increased from 100 s to 200 s due to the congestion buildup. At high

Node Speed	Protocol (#Nodes)	[Number of Flows, Traffic Load per Flow (in packets/s)]									
		[1, 5]	[1, 25]	[3, 5]	[3, 25]	[10, 5]	[10, 25]	[20, 5]	[20, 25]	[40, 5]	[40, 25]
20 m/s	OLSR (50)	0.9	0.9	0.9	0.9	0.89	0.82	0.89	0.76	---	---
	OLSR (100)	0.92	0.91	0.91	0.91	0.89	0.85	0.88	0.76	0.87	0.55
	OLSR (200)	0.92	0.91	0.92	0.91	0.9	0.85	0.88	0.77	0.88	0.57
	MM-OLSR (50)	0.99	0.99	0.98	0.98	0.97	0.95	0.97	0.79	---	---
	MM-OLSR (100)	1	1	1	0.99	0.99	0.96	0.98	0.84	0.98	0.54
	MM-OLSR (200)	1	1	1	1	1	0.98	1	0.88	0.99	0.67
	MCA-OLSR (50)	1	1	1	1	0.99	0.98	0.99	0.83	---	---
	MCA-OLSR (100)	1	1	1	1	1	1	1	0.87	0.99	0.67
	MCA-OLSR (200)	1	1	1	1	1	1	1	0.92	1	0.82
30 m/s	OLSR (50)	0.87	0.86	0.86	0.85	0.81	0.79	0.78	0.72	---	---
	OLSR (100)	0.89	0.88	0.88	0.88	0.84	0.8	0.8	0.75	0.78	0.51
	OLSR (200)	0.9	0.88	0.9	0.89	0.85	0.81	0.82	0.75	0.8	0.55
	MM-OLSR (50)	0.99	0.98	0.96	0.95	0.93	0.9	0.91	0.77	---	---
	MM-OLSR (100)	1	1	0.99	0.98	0.94	0.92	0.92	0.8	0.91	0.5
	MM-OLSR (200)	1	1	1	1	0.99	0.96	0.94	0.83	0.92	0.63
	MCA-OLSR (50)	1	0.99	1	0.99	0.98	0.97	0.97	0.83	---	---
	MCA-OLSR (100)	1	1	1	0.99	1	0.98	0.98	0.84	0.94	0.61
	MCA-OLSR (200)	1	1	1	1	1	0.99	0.99	0.89	0.96	0.7
40 m/s	OLSR (50)	0.71	0.69	0.71	0.68	0.65	0.63	0.54	0.51	---	---
	OLSR (100)	0.73	0.72	0.72	0.7	0.67	0.64	0.58	0.55	0.51	0.45
	OLSR (200)	0.76	0.76	0.74	0.72	0.7	0.67	0.65	0.6	0.58	0.51
	MM-OLSR (50)	0.84	0.82	0.82	0.81	0.72	0.7	0.69	0.65	---	---
	MM-OLSR (100)	0.87	0.87	0.85	0.83	0.76	0.75	0.72	0.7	0.63	0.47
	MM-OLSR (200)	0.91	0.91	0.9	0.87	0.79	0.78	0.76	0.72	0.67	0.58
	MCA-OLSR (50)	0.89	0.89	0.88	0.88	0.79	0.77	0.73	0.69	---	---
	MCA-OLSR (100)	0.92	0.91	0.9	0.89	0.85	0.82	0.81	0.73	0.8	0.59
	MCA-OLSR (200)	1	1	0.97	0.93	0.93	0.87	0.89	0.78	0.84	0.64
50 m/s	OLSR (50)	0.45	0.41	0.41	0.4	0.36	0.34	0.3	0.27	---	---
	OLSR (100)	0.48	0.43	0.43	0.42	0.43	0.41	0.4	0.4	0.39	0.3
	OLSR (200)	0.64	0.61	0.61	0.59	0.51	0.49	0.46	0.42	0.42	0.33
	MM-OLSR (50)	0.54	0.52	0.48	0.47	0.43	0.41	0.37	0.32	---	---
	MM-OLSR (100)	0.8	0.66	0.71	0.64	0.64	0.6	0.62	0.57	0.57	0.37
	MM-OLSR (200)	0.91	0.83	0.85	0.73	0.72	0.64	0.7	0.61	0.61	0.5
	MCA-OLSR (50)	0.69	0.69	0.62	0.62	0.57	0.56	0.54	0.46	---	---
	MCA-OLSR (100)	0.98	0.9	0.86	0.84	0.83	0.79	0.8	0.58	0.78	0.46
	MCA-OLSR (200)	1	1	0.93	0.92	0.91	0.84	0.86	0.74	0.81	0.54

Figure 4.8: Comparison of average PDR values for our proposed MCA-OLSR, MM-OLSR and OLSR schemes for different node density at varying node speeds, number of traffic flows and traffic load per flow for 100 s simulation duration.

node speed, the routes break frequently, which result in packet transmission over different routes and hence the network load is distributed. It partially reduces the congestion buildup at intermediate nodes, and therefore, marginally increases the PDR values in both schemes.

4.4.3 End-to-End Delay Comparison

For video traffic, the OLSR, MM-OLSR and MCA-OLSR schemes have end-to-end delay of up to 0.67 s, 1.32 s and 1.47 s, respectively – well below the packet TTL value of 3 s –

Table 4.2: Impact of Simulation Duration on PDR Performance

Node Speed	Protocol	Simulation Duration	Number of Flows		
			3	10	20
20 m/s	OLSR	100 s	0.9	0.82	0.76
		200 s	0.75	0.71	0.59
	MCA-OLSR	100 s	1	0.98	0.83
		200 s	0.95	0.93	0.77
50 m/s	OLSR	100 s	0.4	0.34	0.27
		200 s	0.44	0.38	0.32
	MCA-OLSR	100 s	0.62	0.56	0.46
		200 s	0.65	0.6	0.49

Node Speed	Protocol (#Nodes)	[Number of Flows, Traffic Load per Flow (in packets/s)]									
		[1, 5]	[1, 25]	[3, 5]	[3, 25]	[10, 5]	[10, 25]	[20, 5]	[20, 25]	[40, 5]	[40, 25]
20 m/s	OLSR (50)	5	5	5	9	8	50	10	150	---	---
	OLSR (100)	8	5	9	7	20	40	23	160	30	570
	OLSR (200)	8	15	10	20	30	40	30	180	38	670
	MM-OLSR (50)	4	4	5	5	6	20	7	190	---	---
	MM-OLSR (100)	4	4	5	5	6	30	7	200	10	760
	MM-OLSR (200)	5	4	5	6	6	40	8	230	12	880
	MCA-OLSR (50)	4	4	5	5	7	10	7	220	---	---
	MCA-OLSR (100)	6	6	6	6	7	40	8	290	10	810
	MCA-OLSR (200)	7	7	6	7	7	60	11	310	14	923
30 m/s	OLSR (50)	6	6	6	7	9	50	10	150	---	---
	OLSR (100)	9	6	9	10	12	53	15	153	40	590
	OLSR (200)	9	15	12	21	16	60	20	160	55	710
	MM-OLSR (50)	5	5	6	6	8	25	8	210	---	---
	MM-OLSR (100)	5	5	10	13	13	33	10	280	40	950
	MM-OLSR (200)	7	10	11	13	19	41	30	320	55	980
	MCA-OLSR (50)	4	3	6	10	8	10	10	268	---	---
	MCA-OLSR (100)	6	4	7	5	11	12	29	283	50	830
	MCA-OLSR (200)	7	7	7	8	16	35	48	300	63	1,020
40 m/s	OLSR (50)	6	6	6	8	9	50	10	151	---	---
	OLSR (100)	8	9	9	10	14	59	20	162	40	630
	OLSR (200)	9	8	14	17	8	49	21	180	50	750
	MM-OLSR (50)	7	6	7	9	9	29	41	225	---	---
	MM-OLSR (100)	7	6	12	14	17	35	190	270	80	870
	MM-OLSR (200)	10	11	12	15	25	50	210	333	151	925
	MCA-OLSR (50)	6	5	7	10	9	10	61	280	---	---
	MCA-OLSR (100)	7	6	8	10	14	18	120	337	127	941
	MCA-OLSR (200)	9	7	13	16	23	41	200	350	175	1,102
50 m/s	OLSR (50)	6	6	6	7	8	50	10	150	---	---
	OLSR (100)	5	9	9	10	20	70	20	170	40	580
	OLSR (200)	9	7	14	12	20	80	32	180	40	610
	MM-OLSR (50)	5	6	7	8	8	40	50	130	---	---
	MM-OLSR (100)	9	7	15	11	20	46	120	140	130	760
	MM-OLSR (200)	11	11	22	29	30	60	190	210	241	806
	MCA-OLSR (50)	7	7	7	7	8	10	60	250	---	---
	MCA-OLSR (100)	8	9	10	10	20	30	160	320	160	850
	MCA-OLSR (200)	7	7	21	21	32	55	210	340	240	925

Figure 4.9: Comparison of average end-to-end delay values (in ms) for our proposed MCA-OLSR, MM-OLSR and OLSR schemes for different node density at varying node speeds, number of traffic flows and traffic load per flow for 100 s simulation duration.

when the network is saturated due to a higher data rate and more flows. End-to-end delay is higher in our MCA-OLSR scheme than in OLSR and MM-OLSR protocols because:

- The MCA-OLSR scheme achieves a higher PDR, so that the total number of transmitted data packets is higher, increasing queuing delay.
- OLSR selects a shortest hop route, whereas MCA-OLSR can select a longer but more stable route, which increases the end-to-end delay.

For sensor data, the average end-to-end delay values are shown in Fig. 4.9 for different node densities, network loads and node speeds. At low network loads and node speeds, all three schemes achieve a high PDR performance (see the PDR values in Fig. 4.8). However, our proposed MCA-OLSR scheme has lower end-to-end delay values compared to the standard OLSR and MM-OLSR schemes because it preemptively switches to a better route when the current route's quality degrades. As the network load and/or node speed increases, the increase in end-to-end delay values in the MCA-OLSR scheme is higher compared to standard OLSR and MM-OLSR for the aforementioned two reasons.

Table 4.3: Comparison of Normalized Control Overhead

Scenario		**OLSR**	**MM-OLSR**	**MCA-OLSR**
50 Nodes	**20 m/s**	0.2	0.3	0.4
	50 m/s	0.2	0.2	0.3
100 Nodes	**20 m/s**	0.6	1.2	2.4
	50 m/s	0.5	1	1.6
200 Nodes	**20 m/s**	3.1	5.7	10.1
	50 m/s	2.2	4.5	7.6

4.4.4 Control (Signaling) Overhead Comparison

Due to the additional fields introduced in MM-OLSR and MCA-OLSR (see Section 4.3.3), their control overhead for both Hello and TC packets is higher than OLSR. We show the

normalized control overhead in Table 4.3 when three source nodes generate traffic at 1 Mbps. Note that the normalized control overhead increases significantly with the number of nodes in the network, and is inversely proportional to the total network traffic load.

At a higher node speed, the number of 1-hop and 2-hop neighbor nodes is reduced due to the increased number of (dis)connected components. As a result, the number of advertised neighbors in the control packet decreases, resulting in a lower control packet size. This effect causes the control overhead to decrease when the node speed increases from 20 m/s to 50 m/s.

4.4.5 Average Number of Route Computations

The average number of routes computed in the three schemes for different node densities and speeds are shown in Table 4.4. The OLSR and MM-OLSR protocols compute a route to all other nodes in the network whenever a control packet is received. However in MCA-OLSR, only the nodes participating in data transmission compute a route to the *active* source-destination pairs. Therefore, the number of routes computed is significantly lower in MCA-OLSR.

Table 4.4: Average Number of Routes Computed

Scenario		**OLSR / MM-OLSR**	**MCA-OLSR**
50 Nodes	**20 m/s**	63,004	75
	50 m/s	46,378	256
100 Nodes	**20 m/s**	2,77,510	50
	50 m/s	2,04,420	987
200 Nodes	**20 m/s**	11,92,147	29
	50 m/s	8,94,443	5,116

4.5 Conclusion

In this work we presented a cross-layer, mobility and congestion-aware MCA-OLSR protocol for autonomous and decentralized UAV networks. Our scheme introduced several new mechanisms: (i) A multi-metric routing mechanism, which incorporates HC, RLT, ETD and IL metrics; (ii) A long-lasting and low congestion route selection; (iii) Preemptive route switching which prevents packet transmission over broken and congested routes; (iv) Queue management to prioritize transmission of packets which have a lower TTE or a higher ETD; and (v) Resource management to identify and remove packets that are likely to expire before reaching their destination. The proposed MCA-OLSR routing protocol achieved significantly higher data throughput and lower route computation overhead for delay-sensitive data flows across a range of different data rates, node densities and node speeds, as compared to the standard OLSR and MM-OLSR protocols.

Chapter 5

A Hybrid Reactive Routing Protocol

The proactive routing schemes (e.g., OLSR, MM-OLSR, MCA-OLSR) discussed in Chapters 2 and 4 introduce a large control and computational overhead. This can increase congestion in the network and degrade the flow throughput. In addition, the proactive routing schemes are vulnerable to the security threats as breaching one node can reveal the entire network topology information, including the relative node locations and their IP addresses.

On the other hand, the reactive routing schemes (e.g., AODV and LEPR reviewed in Chapter 2) search a route on-demand, and incur a much lower control and computation overheads. However, the on-demand route discovery incurs the considerable overhead and delay due to the frequent link breaks amid continuous topology changes in a UAV network. Since the reactive routing schemes trigger a new route discovery only after the current route(s) breaks, these schemes remain unaware of the frequent changes in network topology. Therefore, unlike proactive routing schemes, a source node in reactive routing schemes does not always know the updated network topology, and therefore can select a sub-optimal route for its packet transmission.

In this chapter, we describe our proposed hybrid reactive routing scheme, called mobility and

congestion-aware ad-hoc on-demand distance vector (MCA-AODV) routing, and evaluate its performance for different network and traffic settings.

5.1 Related Work

Traditional reactive routing protocols suffer from the broadcast storm problem due to the flooding of RREQ packets in the network [108, 109]. This results in large control overhead and delay in dynamic network topologies, due to frequent route rediscoveries. To address this problem, nodes in [108] reduce their RREQ forwarding probability as their queue occupancy increases. This decreases the number of congested nodes contesting for the channel access to forward the control packets. Instead of triggering a new network-wide route discovery, the source node in [109] requests the intermediate nodes of the broken route to search for an alternate route in their k-hop neighborhood, which reduces the control overhead. In [46], a node identifies the RREQ forwarding node(s) for each of its different zones based on its neighbor node's directions and their local connectivity. However, these schemes [46, 108, 109] can result in poor route selection.

QoS-aware variants [110–112] of reactive routing protocols use the parameters like congestion along a route, its bandwidth, traffic load, delay, jitter, node mobility, link stability, signal strength and remaining battery life. For example, nodes in [110] rebroadcast the RREQ packet only when they can satisfy the flow QoS requirements (such as flow TTL, minimum bandwidth required for a route and maximum jitter allowed). When an intermediate node on the selected route can no longer meet the flow QoS requirements, it notifies the source node to find a new route. However, [110] does not track the node mobility and link stability, which results in frequent route discoveries in ANs. In [111], an intermediate node adjusts to the changes in the network topology and traffic conditions by reselecting its downstream node (towards the destination node) based on the neighbor distance and its latency, load

and reliability. However, changing routes at intermediate nodes can result in longer and sub-optimal routes.

The multipath reactive routing schemes [113–116] find multiple node- or link-disjoint paths at the time of route discovery and use them when the quality of current route degrades, which decreases the number of route discoveries [117]. However, the quality of remaining routes can also degrade by the time the primary route breaks in dynamic ANs, which causes frequent flow interruptions, and large route switchover overhead and delay [57,62]. Schemes in [62,63] prevent the selection of broken routes by periodically monitoring the quality of the remaining routes, which improves the flow reliability [117]. A fuzzy-logic based multipath routing scheme is proposed in [14], which selects a shorter and low-latency route by using the node mobility, residual energy, link quality and stability metrics. It also repairs the broken route using the local node information, and initiates a new route discovery if no suitable route is found.

However, the schemes in [14,62,63] incur large control overhead. To address this issue, a semi-proactive route switching mechanism is proposed in LEPR [57], wherein the intermediate nodes of the primary route notify the destination node whenever the link quality with their downstream nodes degrades. The destination node then sends a new RREP packet to the source node on each cached route, which carries the recent route stability value. As a result, the source node switches to an alternate route without incurring a large overhead. However, [57] cannot adapt to the changes in traffic and does not provide a route repair mechanism.

5.2 Our Proposed MCA-AODV Routing Protocol

The MCA-AODV scheme searches the route(s) on-demand, and then proactively switches to an alternative route before the quality of current route degrades below a threshold. It minimizes the number of route discoveries, and the resulting overhead, delay and flow interruptions, as compared to the existing on-demand routing schemes (standard AODV and LEPR schemes). At the same time, its control and computational overheads are significantly lower than the proactive routing schemes. Despite having a limited network topology information, the flow throughput and end-to-end delay performances of our MCA-AODV scheme are superior to the classical routing schemes (i.e., standard OLSR and AODV), and comparable (or slightly better) to the modified routing schemes (i.e., MCA-OLSR and LEPR), at different network and traffic settings.

Fig. 5.1 shows the modules of our proposed MCA-AODV routing scheme. A source node triggers a route discovery, when it does not have a high quality route to its destination node. For route discovery, each source node broadcasts an RREQ message. The destination node responds to the received RREQ messages by transmitting RREP messages towards the source node. The source node evaluates all the received RREPs, and selects a long-lasting and less-congested route. Here, each RREP carries the HC, ETD and IL statistics of the route. To ensure packet transmission over a high quality route, the source node must know the updated network topology information. Therefore, the intermediate nodes of the selected route send their 2-hop neighborhood information periodically to the source node. We use the term 'pipe' to refer to the 2-hop neighborhood around the selected route. Using the pipe information, the source node reevaluates the current route and switches to an alternative route within pipe, when needed. If no route within the pipe has the required route quality, the source node proactively triggers a network-wide route discovery. Like the MCA-OLSR scheme, our MCA-AODV scheme uses the proactive queue management mechanism, which was discussed in Section 4.3.2.

We discuss the modules of Fig. 5.1 in Sections 5.2.1 to 5.2.3, followed by a discussion of control and computational overhead in Section 5.2.4. The working of our proposed MCA-AODV scheme is given in Algorithm 6.

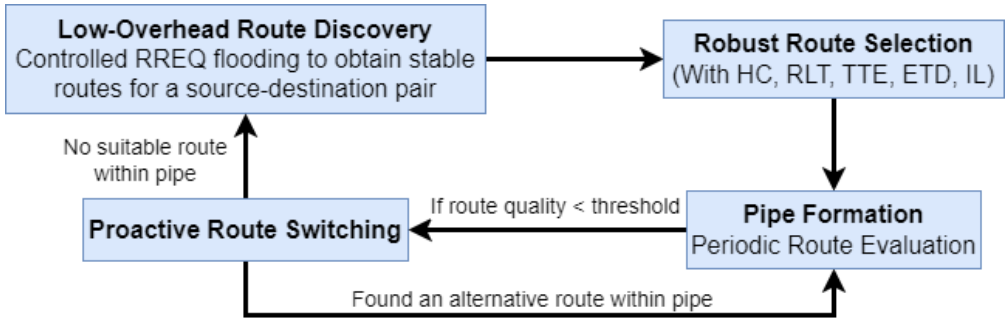


Figure 5.1: Modules used in our MCA-AODV scheme.

Algorithm 6: Overview of our MCA-AODV Scheme

Route discovery: Source node starts a network-wide route discovery with controlled RREQ flooding (see Section 5.2.1)
Destination node sends RREP packet(s), which carries the HC, ETD and IL values of the route (see Section 5.2.2)
Route selection: Select a stable and less-congested route R^* from all the received RREPs using (4.4) and (4.5)
Pipe formation: Intermediate nodes of the selected route R^* collect their 2-hop neighborhood information to create a virtual pipe (see Fig. 5.2)
Network topology update: Periodically send the pipe information to the source node
Route reevaluation and switching: Source node checks if any of the three conditions of (4.6) occurred
if Any of the three conditions of (4.6) occurred **then**
    if An alternative route R with quality $>$ exists within the pipe **then**
        Switch to route R (i.e., $R^* = R$)
        Go to the pipe formation step
    else
        Go to the route discovery step
    end
end

5.2.1 Low-Overhead Route Discovery

When a node receives a control packet (e.g., RREQ, RREP or RERR) in standard AODV protocol, it stores the packet originator node, destination node and packet's sequence number

to avoid reprocessing the same packet. Since the RREQ packets are broadcast, they traverse different routes to reach the destination node. Therefore, a destination node knows about different routes to the source node. However, it originates a RREP packet only when the RREQ packet is received via a shorter hop route than the previously known route or when the sequence number of the received RREQ packet is higher.

Like standard AODV, a source node in MCA-AODV broadcasts RREQ packet when it wants to find a route to a destination node. However, instead of blindly forwarding the RREQ packets, a receiver node i considers its link stability with the RREQ transmitter node j. The RREQ packet is broadcast, only if the following link stability condition is satisfied.

$$LLT_{i,j}(i) \; > \; TTL \; + \; \delta(i) \tag{5.1}$$

Here, TTL represents the time-to-live value of the flow, which is determined based on the application. In our scheme, the source node includes the TTL value in RREQ packets. δ is a control parameter, which is a function of node speed and how quickly the neighborhood of a node changes; it can be tuned locally at each node. Thus, the RREQ packets traverse towards destination node on longer-lasting routes. This reduces the number of control packet transmissions. If a source node cannot find a route that satisfies (5.1), this condition is bypassed.

5.2.2 Robust Route Selection

The route selection mechanism in MCA-AODV is inspired by the two-step route selection mechanism of MCA-OLSR (see Section 4.3.2). In order to obtain the *HC*, *ETD* and *IL* statistics of the complete route, only the destination node originates the RREP packet in our scheme. Note that the *Hop Count* field in RREP packets represents the hop length of

the route from the current node to the destination node. When a node along the reverse route (i.e., towards the source node) receives the RREP packet, it adds its own *PST* and *IL* values in the RREP packet's *ETD* and *IL* fields, respectively (see (4.2) and (4.3) for details on *ETD* and *IL*, respectively). When a source node receives multiple RREPs, it selects the best route using (4.4b) and (4.5).

5.2.3 Pipe Formation and Proactive Route Switching

Due to the frequent changes in the network topology, the quality of current route can degrade with time and new better-quality routes can become available. In order to improve the flow throughput and reduce flow interruptions, the source node should proactively switch to the highest quality route, if available. Since the future node trajectories are not known, predicting a high quality route based on the current network topology information is very difficult. Therefore, the nodes in proactive routing schemes broadcast their information periodically in the entire network. However, there is a tradeoff between selecting a better quality route vs. control overhead. Therefore, the source node in MCA-AODV collects a limited network topology information around the currently used route for enabling route switching. A new route discovery is initiated only when the current network information cannot provide a route with the required quality.

(A) Pipe Formation:

Each node in MCA-AODV scheme includes its GPS coordinates in its Hello packet in order to compute the *LLT* value for its links with 1-hop neighbor nodes. Unlike the proactive routing schemes, the nodes do not inform about their 1-hop neighbor nodes by default. When a node receives a data packet, it informs its 1-hop neighbor nodes about its participation in packet

forwarding process by including an '*isActive*' flag in its Hello packet[1]. Upon receiving a Hello packet with '*isActive*' flag set, a node includes its 1-hop neighborhood information (i.e., node IP address and their links) in its hello packet (see Fig. 5.2(a)). Since the Hello packet includes node's GPS coordinates, a node can know which of its 1-hop neighbors are in the transmission range of each other. As a result, the intermediate nodes of the selected route know their 2-hop neighborhood information. We refer to the complete 2-hop neighborhood information along a route as its pipe (see Fig. 5.2(b)). If a node does not receive a data packet to forward for the ACTIVE_ROUTE_TIMEOUT duration (its default value in the standard AODV protocol is 3 s [53]), it turns off the '*isActive*' flag.

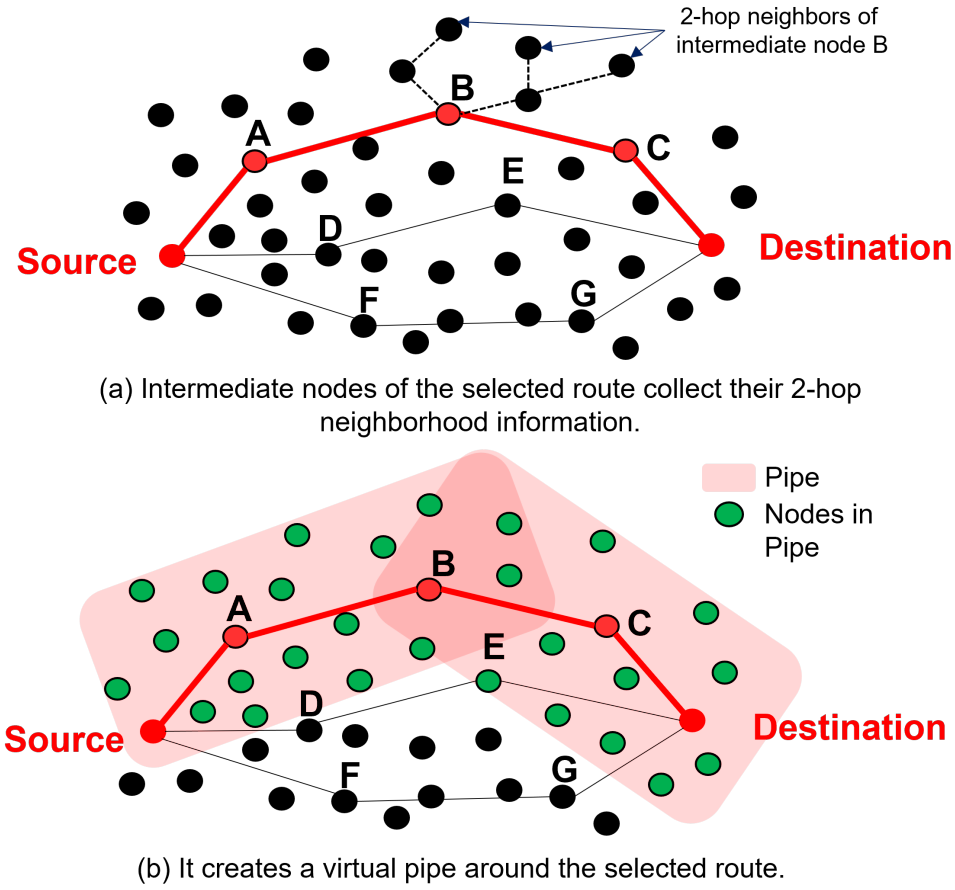


Figure 5.2: Pipe formation in our MCA-AODV scheme.

Impact of pipe width: The network topology known in our scheme is limited by the pipe

[1]Note that a node can also find its active neighbors by passively listening to the channel.

width (i.e., 2-hop in this case). Increasing the pipe width would increase the fraction of the known network topology, which can provide better quality route(s), at the cost of higher control and computational overhead.

(B) Sending Pipe Information to the Source Node:

After the destination node receives the first data packet, it schedules an interrupt to send a Notify_Source packet to the source node after each Hello interval. The Notify_Source packet passes through all the intermediate nodes of the route and appends their 2-hop neighborhood information along with their *PST*, *LLT* and *IL* values. If the destination node does not receive a data packet for ACTIVE_ROUTE_TIMEOUT duration, it suspends the Notify_Source interrupt.

To reduce the control overhead, an intermediate node constructs a graph using the received information of the 2-hop neighborhood from its downstream node and includes its own information to compute the cliques [118]. Then, it includes both the node statistics (*PST* and *IL* values) and link statistics (*LLT* value and *isLinkActive* flag[2]) only once, and forwards to its upstream node.

(C) Route Switching Mechanism:

Route switching mechanism of MCA-AODV scheme is inspired by MCA-OLSR scheme (see Section 4.3.2). If a new shorter hop-count route becomes available within the pipe, the source node selects the highest quality route from the routes available within the pipe using (4.5). When the quality of current route degrades below a threshold (as discussed in (4.6a) and (4.6b)), the source node searches for a new better-quality route within the pipe. If a suitable route is found, source node switches to the new route, otherwise it triggers a new route

[2]Required to compute IL^{ϕ} in (4.3).

discovery. When the source node switches to a new route, the existing pipe is abandoned and a new pipe is formed around the new route. The route switching mechanism, therefore, reduces the number of route discoveries and the resulting discovery overhead and delay. It also minimizes the interruption in flow throughput.

5.2.4 Control and Computational Overhead

(A) Control Overhead:

In addition to the information sent in the RREQ, RREP, RERR and Hello packets in the standard AODV protocol (see Section 2.2.1 for details), each node in the MCA-AODV scheme includes the following information in its Hello packet:

- Its GPS location (uses 6 bytes for (x,y,z) coordinates) and trajectory information, which includes the center coordinates (6 bytes), and radius and node movement direction in ST mobility model (2 bytes) [67]. Note that a node sends its trajectory information only when it forms a new link or changes its current trajectory.
- Its PST and IL values (1 to 2 bytes).
- If the node is a 1-hop neighbor of an active node (which is participating in data packet forwarding), it broadcasts the LLT, PST, IL and $isLinkActive$ values for each of its 1-hop neighbor nodes (variable size).

In addition, each destination node in MCA-AODV periodically sends a Notify_Source packet to the source node, which carries the 2-hop neighborhood information (i.e., node IP address and their PST, IL and LLT values) of all intermediate nodes on the route (variable size).

(B) Computational overhead:

In MCA-AODV scheme, a source node finds all routes to the destination node within the pipe using the BFS (breadth first search) algorithm, which has a worst time complexity of $O(T{V_p}^2(E_p+V_p))$, where T is the number of times routes are recomputed, and V_p and E_p are the number of nodes and links, respectively, within the pipe. In addition, the intermediate nodes compute the cliques of their 2-hop neighborhood using Bron-Kerbosch algorithm [118], which has a worst time complexity of $O(3^{V_p/3})$. Note that the clique computation is optional and done to reduce the control overhead.

5.3 Simulation Setup and Results

We have implemented the MCA-AODV and LEPR schemes in the discrete event simulator, NS-3 version 3.29. Our simulation and network parameters are summarized in Table 5.1. In our simulation, fixed-wing UAVs fly in a 8×8 km^2 area using the smooth-turn mobility model discussed in Section 2.3.1. The performance is evaluated for different network settings: node density (50 and 100 UAVs), low to high network loads (i.e., 0.5 to 3 Mbps at constant bit rate), and minimum to high stalling speeds (i.e., 20 m/s and 50 m/s, respectively). Here, a line-of-sight communication is assumed and channel fading and noise are ignored.

The channel rate is 11 Mbps and each node has a 1 km transmission range. The packet size and TTL values are 1 kB and 3 s, respectively, and the MAC queue can store up to 1000 packets. The MAC layer protocol is CSMA/CA protocol. Each simulation is run for 100 s and each experiment is repeated 40 times, with the source-destination pair(s) of a flow selected randomly in each run.

The metrics used for the performance evaluation are discussed in next section. We first compare the performance of our MCA-AODV scheme to the standard AODV and LEPR

Table 5.1: Simulation Parameters

Parameters	**Values**
Simulation Area	$8 \times\ 8\ km^2$
Channel Rate	11 Mbps
Transmission Range	1 km
Node Density	50, 100
Node Speed (in m/s)	20, 50
Number of Traffic Flows	1, 3
Flow Rate	40 kbps to 3 Mbps
Packet Size	1 kB
TTL	3 s
Simulation Duration	100 s

scheme in Section 5.3.2, followed by the comparison with proactive routing schemes (i.e., standard OLSR and MCA-OLSR) in Section 5.3.3. The standard AODV and OLSR protocols and LEPR scheme were discussed in Chapter 2 and the MCA-OLSR scheme was discussed in Chapter 4.

5.3.1 Performance Metrics

We use the following four performance metrics:

1. **PDR** of a flow is the ratio of total data packets received at the destination node to the total data packets generated at the source node. When plotting average PDR, we also indicate the 95% confidence interval (shaded region).

2. **Number of route discoveries** is the average number of RREQ packets generated per flow at the source node during the simulation. A lower value is desired as it signifies more stable route(s).

3. **Number of route control packets** is the total number of route setup, update and maintenance packets (i.e., RREQ, RREP, RERR and Hello) transmitted over the network during the simulation duration. This metric also includes the number of Notify_Source packets in MCA-AODV scheme and RSWT packets in LEPR scheme.

4. **Control (signaling) overhead** is the size of the total route control packets transmitted during the simulation.

5.3.2 Comparison with Reactive Routing Schemes

Here, we evaluate the performance of the proposed routing scheme in terms of PDR, number of route discoveries, and route control overhead.

(A) PDR Comparison:

(i) Impact of Traffic Rates: Fig. 5.3 shows the PDR of each scheme at one and three flows across varying data rates (until the network becomes heavily congested). Since our proposed MCA-AODV scheme decreases the RREQ flooding overhead, selects stable and less congested routes, and preemptively discovers a new route when the current route degrades due to the topology change(s) and/or congestion buildup, it provides a higher average PDR than the standard AODV and LEPR schemes.

Fig. 5.3(a) shows the PDR comparison for 50 nodes at 20 m/s speed. Our MCA-AODV achieves up to 30% and 12% higher PDR as compared to the AODV and LEPR schemes, respectively. Since the links break less frequently at lower node speed, the PDR values are high in all three schemes for both one and three flows at low data rate. However, the PDRs of all the schemes decrease as the traffic load increases and the network becomes heavily congested at 3 Mbps (for one flow) and 1.25 Mbps data rates (for three flows).

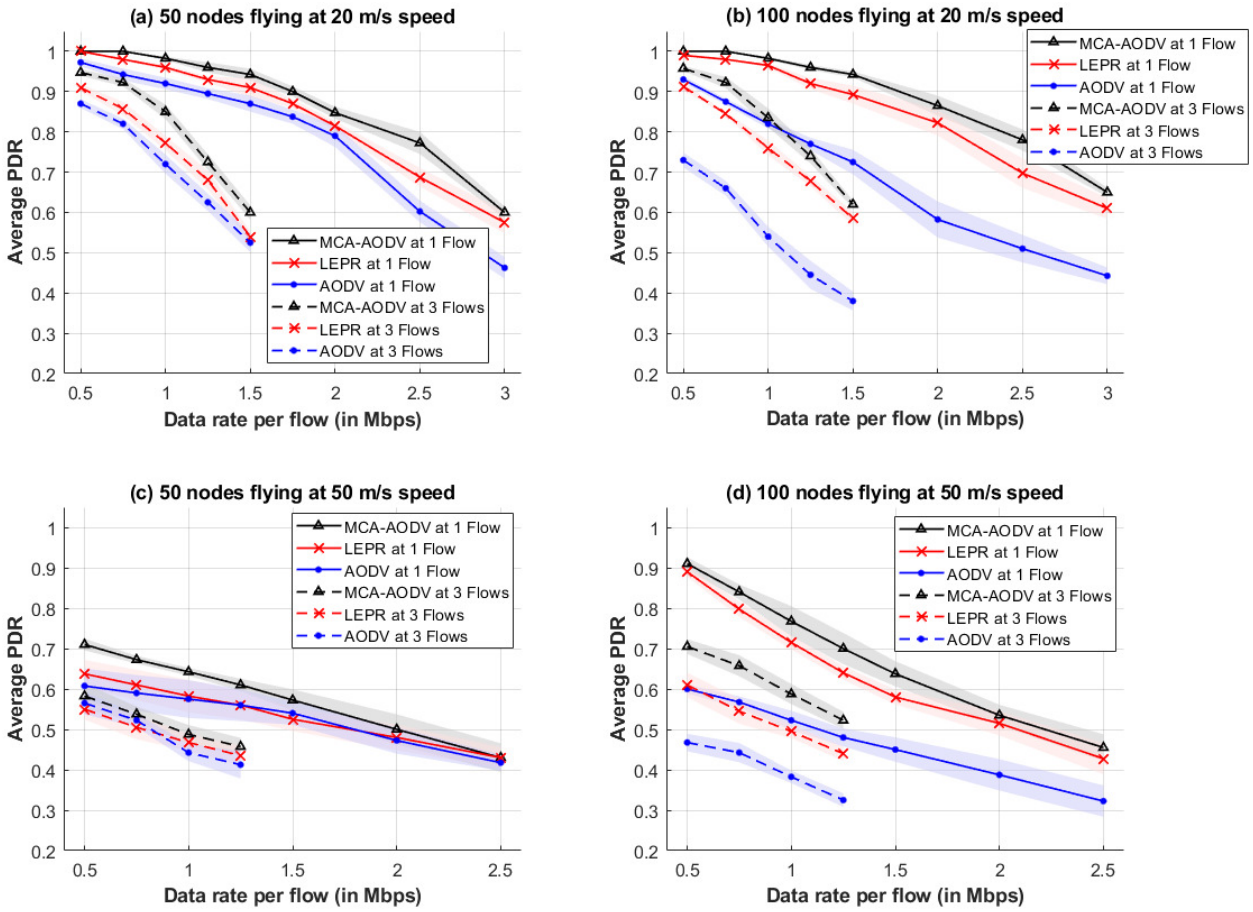


Figure 5.3: PDR comparison for MCA-AODV, LEPR and AODV schemes for different number of data flows and data rates, for varying node density and speed.

For 100 nodes (higher node density) flying at 20 m/s in Fig. 5.3(b), MCA-AODV achieves up to 64% and 11% higher PDR compared to AODV and LEPR schemes, respectively. Although more routes may become available for a source-destination pair in the higher density network, the reactive routing schemes continue to use the current route until a new route discovery or route switching is triggered. Further, the routes break less frequently at 20 m/s which results in low number of route discoveries. As a result, the increased density does not improve the PDR performance. However, the PDR performance of AODV scheme degrades at a density of 100 nodes due to the higher control overhead which increases the control packet collisions and thereby results in a higher number of route discoveries. Since a source node tracks the nodes within the pipe around the active route (in MCA-AODV) or caches different link-disjoint routes spanning over the network topology (in LEPR), it

switches to a less-congested route, whenever needed. Therefore, the PDR performances of MCA-AODV and LEPR is not impacted.

Fig. 5.3(c) shows the PDR for 50 nodes flying at 50 m/s speed. The PDR performance of MCA-AODV is up to 16% and 11% higher as compared to AODV and LEPR, respectively. Since the links break more frequently at 50 m/s, which require a higher number of route discoveries, the packets loss due to the TTL expiry increases. Thus, the PDR performance degrades for all the three schemes as compared to node speed of 20 m/s in Fig. 5.3(a).

Fig. 5.3(d) shows the PDR performance for 100 nodes at 50 m/s. The PDR performance of MCA-AODV is up to 58% and 20% higher than AODV and LEPR, respectively. Although the route discoveries increase at 50 m/s, the number of routes available between a source-destination pair increases with an increase in the node density. As a result, the source node in both MCA-AODV and LEPR successfully switches to an alternate route before the primary route breaks. Therefore, the PDR values in both these schemes improve at 100 nodes as compared to 50 nodes.

(ii) Impact of Node Speed: Fig. 5.4 examines the impact of different node speeds on PDR at node densities of 50 and 100 nodes. At higher speed, the network topology changes more frequently and gets fragmented in more connected components. This results in frequent route breaks and sometimes no route is available for a flow, which decrease the PDR performance. Our MCA-AODV scheme selects a longer-lasting and less-congested route and performs the route switching within the pipe when the current route degrades. It, therefore, spends less time in finding a new better quality route as compared to the AODV and LEPR schemes. Note that AODV does not monitor the route quality and, therefore, needs extra time to detect a route break. Whereas, LEPR scheme evaluates the stability of all cached routes before initiating a new route discovery, which incurs considerable delay when links break frequently. Thus, MCA-AODV achieves up to 48% and 10% higher PDR compared to AODV and LEPR, respectively, at the high speed of 50 m/s.

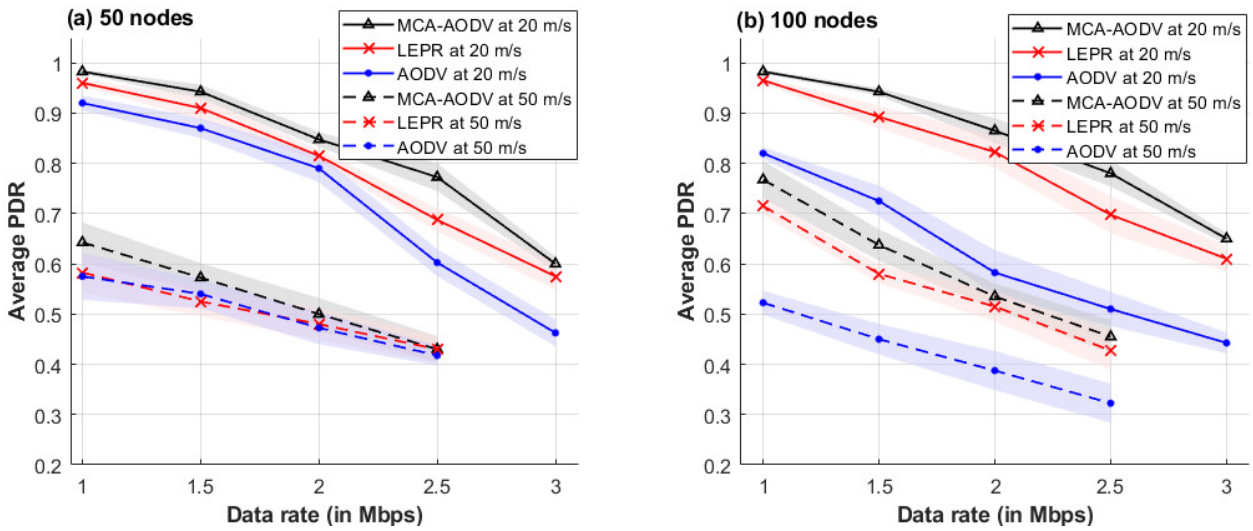


Figure 5.4: PDR comparison for MCA-AODV, LEPR and AODV schemes at different node speeds, for one flow at node density of (a) 50 and (b) 100.

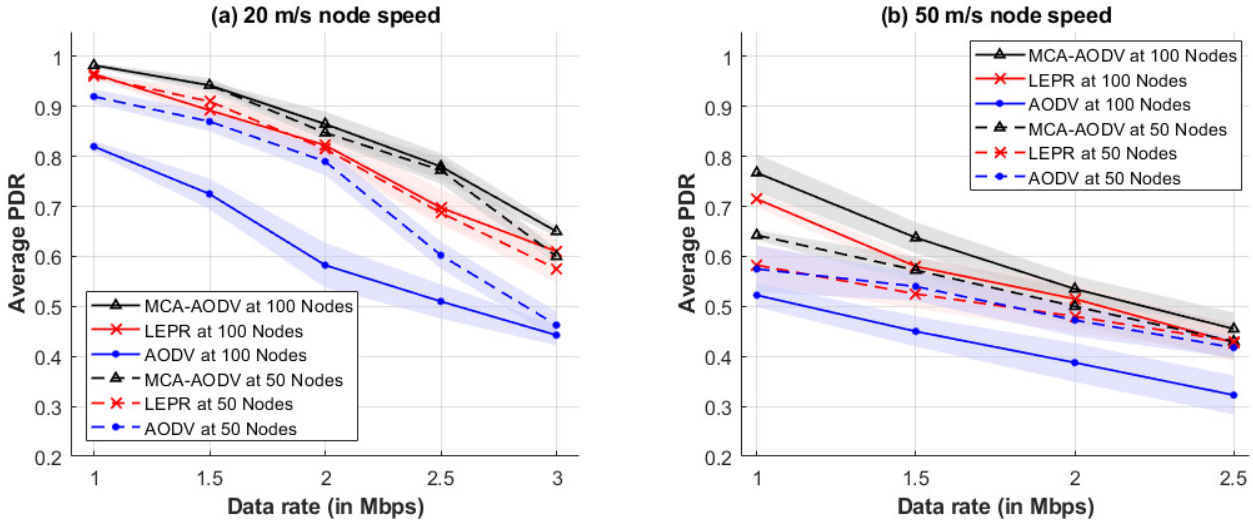


Figure 5.5: PDR comparison for MCA-AODV, LEPR and AODV schemes at different node densities, for one flow at node speeds of (a) 20 m/s and (b) 50 m/s.

(iii) Impact of Node Density: Fig. 5.5 illustrates the PDR performance when the node density increases from 50 to 100 nodes, at node speeds of 20 m/s and 50 m/s. The impact of node density at 20 m/s was explained in the discussion on Fig. 5.3(b). At 50 m/s, the PDR performance of MCA-AODV and LEPR schemes improves by up to 20% and 24%, respectively, when the node density increases from 50 to 100 nodes. As explained above, the routes break more frequently at 50 m/s than at 20 m/s. Since more routes are available at 100 node density, both these schemes are able to switch to an alternate route when the

current route breaks.

Traffic Flows	Performance Metric	Data Rate (in Mbps)	Node Speed (in m/s)	Protocol and Node Density					
				AODV		LEPR		MCA-AODV	
				50	100	50	100	50	100
1	No. of Route Discovery	1	20	5	19	9	10	3	3
			50	18	22	21	21	19	13
		2.5	20	17	26	12	13	6	6
			50	27	27	27	26	20	16
	No. of Route Control Packets (× 1,000)	1	20	7	95	6	11	5	12
			50	7	42	6	12	6	12
		2.5	20	18	113	7	12	6	13
			50	11	51	8	15	8	15
	Control Overhead (in MB)	1	20	0.2	4.2	0.2	0.7	0.2	0.9
			50	0.1	1.5	0.1	0.5	0.1	0.5
		2.5	20	0.3	4.8	0.2	0.7	0.2	0.8
			50	0.2	1.7	0.1	0.5	0.1	0.5
3	No. of Route Discovery	0.5	20	13	29	12	15	3	3
			50	24	24	24	20	20	14
		1.25	20	35	35	29	27	6	6
			50	29	30	27	22	20	15
	No. of Route Control Packets (× 1,000)	0.5	20	20	156	7	14	5	12
			50	11	73	7	14	7	14
		1.25	20	39	144	10	18	7	19
			50	18	83	9	17	8	17
	Control Overhead (in MB)	0.5	20	0.5	6.4	0.2	0.9	0.3	1.2
			50	0.2	2.5	0.2	0.7	0.2	0.6
		1.25	20	0.8	5.9	0.3	1.2	0.2	1
			50	0.3	2.7	0.2	0.7	0.1	0.6

Figure 5.6: Comparison for MCA-AODV, LEPR and AODV schemes in terms of the number of route discoveries, number of route control packets and control overhead at different number of traffic flows, node speeds, densities and data rates.

(B) Comparison of Number of Route Discoveries:

Fig. 5.6 shows the number of route discoveries in the three routing schemes for one and three traffic flows at two data rates, at two different node speeds and densities each. Here, MCA-AODV shows a superior performance over the AODV and LEPR schemes because it searches for an alternate route within the pipe before initiating a new route discovery. This is partially similar to the route repair mechanism [62, 111, 113, 116] in which a node searches for an alternate route using its local information before notifying the source node about the route break. Since AODV and LEPR do not use any route repair mechanism, they have a higher number of route discovery as compared to MCA-AODV.

(i) Impact of Traffic Load: When the traffic load increases and the network gets congested, the number of packet collisions and channel access time increase. When a node does not hear from its 1-hop neighbor node for a certain time duration, it assumes a link break which can trigger a route discovery in AODV and LEPR schemes. An intermediate node in MCA-AODV computes the LLT value for each of its 1-hop neighbor node and therefore know when the link is likely to break. Moreover, it constructs a 2-hop neighborhood graph using the information from all of its 1-hop neighbor nodes. Recall that each 1-hop neighbor node of an active intermediate node includes its 1-hop neighborhood information in its Hello packets. Since the information is coming from different neighboring nodes, the possibility of mistakenly assuming a link break with a node is low.

(ii) Impact of Node Speed: The links break more frequently when the node speed increases, resulting in a higher number of route discoveries in all three schemes. However, the increase is lower in MCA-AODV as compared to other two schemes when the node speed increases from 20 m/s to 50 m/s because it can switch to an alternate route within the pipe without initiating a new route discovery.

(iii) Impact of Node Density: The number of routes available between a source-destination pair increases with the node density. However, we observe different trends for the three schemes when the node density increases from 50 to 100 nodes, as discussed below. At 20 m/s node speed, the number of route discoveries in MCA-AODV and LEPR schemes remain either unchanged or vary marginally for both node densities, whereas it increases for higher node densities in AODV. At 50 m/s node speed, the number of route discoveries decreases in MCA-AODV, whereas it remains either unchanged or decreases marginally for LEPR scheme. For AODV scheme, the number of route discoveries either remains unchanged or increases marginally at higher node densities.

(C) Comparison of Route Control Packets:

As compared to AODV, the number of route control packets is low in LEPR and MCA-AODV schemes at different node densities, speeds and traffic loads. At high traffic loads, it increases marginally in LEPR and MCA-AODV schemes as compared to AODV. It increases significantly in AODV scheme as compared to only a marginal increase in LEPR and MCA-AODV schemes, when node density increases. When the node speed increases from 20 m/s to 50 m/s, the number of route control packets decreases significantly in AODV due to an increase in the number of connected components in the network. The number of route control packets changes only marginally in LEPR and MCA-AODV schemes.

(D) Control Overhead Comparison:

As compared to AODV, the control overhead is low in LEPR and MCA-AODV schemes at different node densities, speeds and traffic loads. At higher traffic load, it changes marginally in LEPR and MCA-AODV schemes, whereas it increases in AODV scheme due to an increase in the number of route rediscoveries. It increases with node density in all three schemes. However, the increase is much higher in AODV scheme. At high node speed, it decreases for all the three schemes due to an increase in the number of connected components in the network.

5.3.3 Comparison with Proactive Routing Schemes

Here, we compare the performances of the standard OLSR and AODV schemes, and MCA-AODV and MCA-OLSR schemes in terms of their PDR and control overhead.

(A) PDR Comparison:

Fig. 5.7 shows the comparison of our proposed MCA-AODV and MCA-OLSR schemes and classical OLSR and AODV schemes for different node densities, speeds, data rates and number of flows. Unlike the proactive routing protocols, the reactive routing protocols remain unaware of the new, better quality routes formed due to the changes in network topology and/or traffic conditions until a fresh route discovery or switching is triggered. Since MCA-AODV scheme knows only a fraction of the network topology (which depends on the pipe

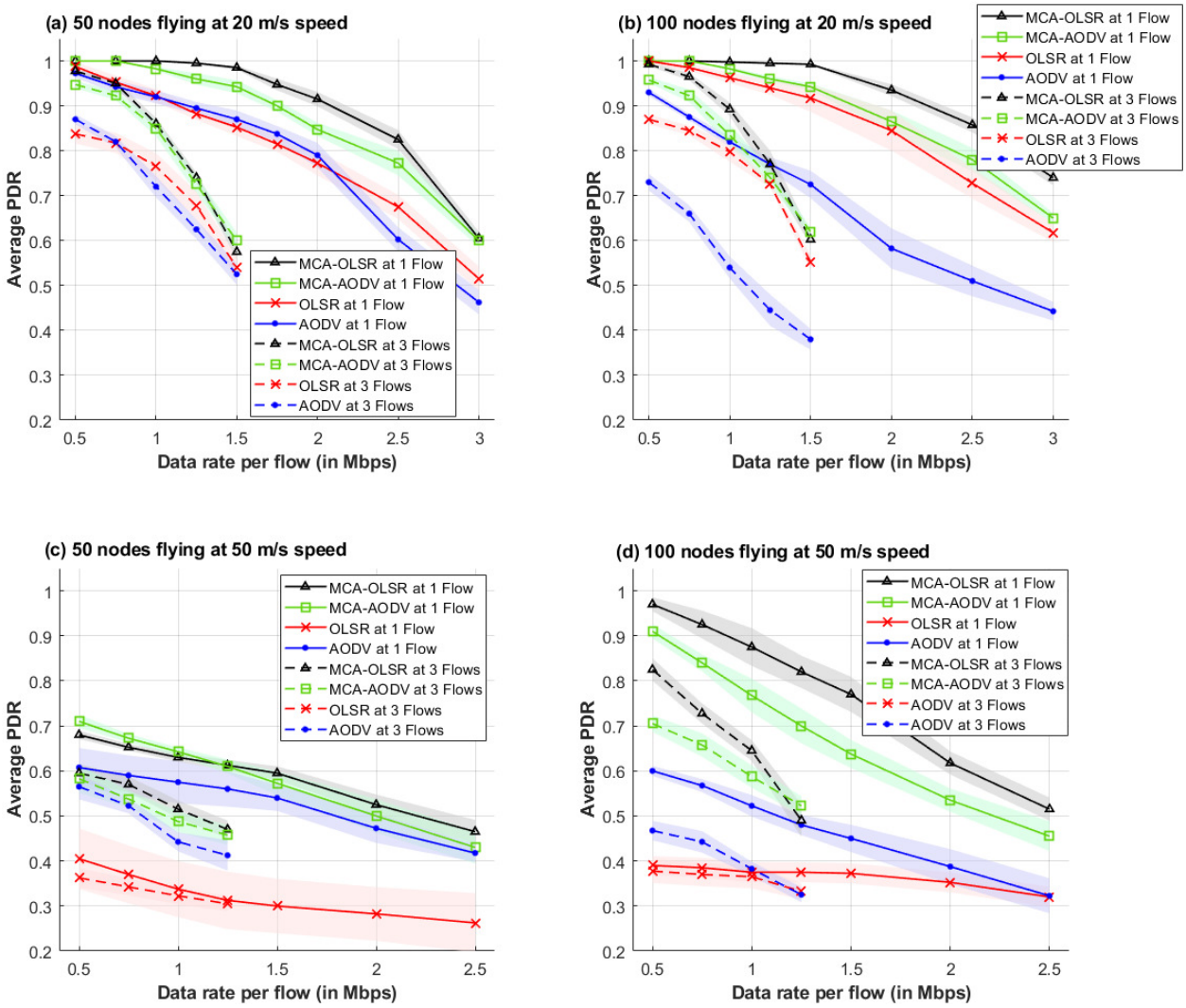


Figure 5.7: PDR Comparison for MCA-OLSR, MCA-AODV, and standard OLSR and AODV schemes for different number of data flows and data rates, for varying node density and speed.

width), its PDR performance is lower than MCA-OLSR. However, its PDR performance is superior to the standard OLSR because it uses longer-lasting and less-congested routes, and preemptively switches to an alternative route (or triggers a new route discovery), when the quality of current route degrades.

The PDR performance of the standard AODV protocol is lower or nearly the same as standard OLSR for 50 nodes at 20 m/s (see Fig. 5.7(a)). Fig. 5.7(b) shows the PDR performance for 100 nodes at 20 m/s. Due to the high node density and low node speed, more stable routes become available between a source-destination pair. Therefore, the PDR performance of standard OLSR improves for both one and three flows, as compared to the node density of 50 nodes. However, the PDR performance in standard AODV decreases when the node density increases from 50 to 100 nodes due to the increased control overhead. Therefore, the PDR performance of standard OLSR is significantly superior to standard AODV in Fig. 5.7(b).

Since the standard OLSR always selects a shortest hop route, it uses the edge nodes. This causes frequent link breaks at high node speed and therefore leads to packet transmission over short-lived routes. In contrast, the standard AODV does not necessarily selects the shortest hop route. As a result, it experiences a lower number of route breaks which decreases the number of packets transmitted on short lived routes. Therefore, the total packets dropped due to TTL expiry is lower in standard AODV, which results in a higher PDR as compared to standard OLSR in Fig. 5.7(c) and 5.7(d).

(B) Control Overhead Comparison:

As compared to the standard OLSR and MCA-OLSR schemes, the control overheads in both AODV and MCA-AODV schemes are significantly lower for both node densities and speeds in Table 5.2. Since a node periodically sends its control messages (i.e., Hello and TC) in the

proactive routing schemes, their control overheads are independent of the number of traffic flows. In contrast, the control overhead increases in both AODV and MCA-AODV schemes as the traffic load and node density increase. Since the number of connected components increases at 50 m/s, the control overhead for all the schemes is lower as compared to 20 m/s.

Table 5.2: Comparison of Control (Signaling) Overhead (in MB)

Scenario		**OLSR**	**MCA-OLSR**	**AODV**		**MCA-AODV**	
				1 Flow	**3 Flows**	**1 Flow**	**3 Flows**
50 Nodes	**20 m/s**	11.5	18.7	0.3	0.8	0.2	0.9
	50 m/s	1.1	6.7	0.2	0.3	0.1	0.5
100 Nodes	**20 m/s**	64.7	181.3	4.8	6.4	0.5	1.2
	50 m/s	47.4	103.2	1.7	5.9	0.2	0.6

5.4 Conclusion

In this chapter, we presented a mobility and congestion-aware, hybrid reactive routing protocol for autonomous and decentralized UAV networks. The proposed MCA-AODV scheme searched route(s) on-demand, monitored the dynamic region around the selected route, and then proactively switched to an alternative route before the quality of current route degrades below a threshold. It minimized the number of route discoveries, the resulting overhead, and flow interruptions, as compared to the standard AODV and LEPR routing schemes. At the same time, its control and computational overheads were significantly lower than the proactive routing schemes (standard OLSR and MCA-OLSR schemes). Despite having a limited network topology information, the flow throughput performance of our MCA-AODV scheme was superior to the classical routing schemes (i.e., standard OLSR and AODV), better than LEPR scheme, and comparable to the MCA-OLSR scheme at different network and traffic settings.

The experiments reported in the literature [119–121] to compare the performance of proactive

and reactive routing schemes are not comprehensive because they evaluate their performance only for certain network or traffic settings. Since both proactive and reactive routing protocols use different approaches for route selection and switching, their performance depends on the network and traffic conditions [122]. For example, the standard AODV scheme provides a better PDR performance as compared to the standard OLSR scheme at high node speeds, whereas OLSR provides a superior performance at high node density and/or traffic loads, as shown by our experiments in Section 5.3.3.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Due to the reduced cost of UAVs, fast deployment, device autonomy and increased flight time capabilities, the autonomous UAV networks achieve reliability and fault tolerance, reduce task completion time through collaboration, and adapt to dynamic task requirements. The autonomous, decentralized UAV networks typically experience varying network design configuration and communication constraints, including the node density, speed, and trajectory, which result in a highly dynamic and unpredictable network topology. In addition to the distinct network characteristics, these networks should also support diverse applications which include different traffic types and priority, data generation rate, session length, reliability, and latency tolerance. These requirements can vary with time, which result in varying QoS demands. Therefore, designing a robust communication mechanism for autonomous, decentralized UAV networks is a challenging task.

To facilitate a reliable communication in these networks, we need novel cross-layer protocols which could identify the region(s) of interest in the network, understand the traffic needs,

and provide a tailored solution for each region by making the best use of available resources. In this dissertation, we have designed the distributed, cross-layer routing and MAC protocols for providing a robust and reliable communication in autonomous and decentralized UAV networks, where the network topology and traffic conditions change frequently, and the future node trajectories are not known. These protocols can discover and select high quality routes and switch to alternate routes in response to the changes in the available communication resources, offered traffic patterns and performance demands to make the best use of the network resources.

We discussed a mathematical framework to compute the LLT for a realistic node mobility model (ST model) in **Chapter 2**. Then, we presented a novel, distributed TDMA scheme in **Chapter 3**, by designing a low-complexity, rank-based scheduling mechanism for directional communication in multihop networks. The proposed scheme adapted to the changes in the network and QoS demands in real-time with significantly lower overhead and delay, and improved the channel utilization and fairness in channel access allocation. The proposed scheme provided a higher throughput with very low control overhead as compared to other state-of-the-art schemes for different network and traffic settings.

The traditional topology-based routing schemes are slow in adapting to topology and traffic changes, and select a route without considering the effect of intra-flow interference on the selected route. To resolve the these issues, we designed an adaptive, cross-layer, mobility and congestion-aware proactive routing protocol for decentralized UAV networks in **Chapter 4**. It includes a novel, multi-step and multi-metric, inter- and intra-flow interference-aware route selection mechanism, which selects a stable, longer-lasting and less congested route. We also used preemptive route switching and periodic queue management mechanisms. The former prevents potential packet drops due to the congestion and topology changes. The latter prioritizes the transmission of packets which have a lower survivability score and discards the packets which are likely to expire before reaching the destination node. The proposed

routing scheme achieved a significantly higher data throughput for delay-sensitive traffic flows at different traffic and network settings, as compared to the standard and multi-metric OLSR routing protocols.

The proactive routing protocols incur large control and computational overhead and are vulnerable to the security threats. In contrast, the reactive routing protocols (such as AODV) incur a much lower control and computation overheads, but the on-demand route discovery introduces the routing overhead and delay, which increase significantly due to frequent topology changes and link breaks in UAV networks. We presented a hybrid mobility and congestion aware reactive routing protocol in **Chapter 5**, which discovers the routes on-demand and preemptively switches to another high-quality route within the region around the selected route. This significantly reduced the number of route discoveries and route control and computational overheads. Despite having a limited network topology information, the proposed routing scheme provided a superior flow throughput performance at different network and traffic settings.

6.2 Future Work

In addition to the contributions described, this thesis also raises a number of open directions for future work, which are discussed below.

1. **Graph inference to optimize route selection in a multi-flow, high traffic network topology:** In decentralized network topologies, nodes do not know about other traffic flows and/or their time-varying requirements. Even in a proactive routing scheme, a node does not know all links in the network topology because the MPR nodes notify only about their MPR selector nodes. As a result, the source nodes in the network cannot collaborate to find (and change as needed) mutually beneficial,

non-interfering routes. To address this issue, the graph inference techniques, such as [123], can be used to infer the missing links and their statistics (such as LLT, and $isLinkActive$, etc.). It would decrease the variance in the network topology known at each node, and thereby allow for a joint optimization for route selection. It can also help in finding multiple, interference-free node-disjoint routes for a source-destination pair, which would allow a source node to accommodate high-traffic flows by distributing its load over multiple routes.

2. **Pipe-aware UAV mobility model:** In this thesis, we used a generalized mobility model wherein each node selects its own trajectory, which can result in frequent network fragmentation, especially at high node speeds. However, many applications require UAVs to maintain the network-connectivity by adjusting their trajectories while performing the assigned task (e.g., surveillance). To ensure a robust and seamless communication in such applications, new UAV mobility models can be designed, where UAVs adjust their trajectories to form a pipe around the ongoing communication(s). This would increase the number of high-quality routes between a source-destination pair. In such network settings, our proposed hybrid reactive routing protocol can work in tandem with the mobility model.

3. **TDMA MAC protocol for multi-beam communication:** The single-beam directional TDMA MAC protocol proposed in Chapter 2 can be extended to support multi-beam communication, which would further improve the spatial reuse and network capacity. This would require an integrated design of MAC and routing layer protocols in order to optimally use the multiple beams in a multi-hop network topology.

Bibliography

[1] M. M. Alam and S. Moh, "Survey on Q-learning-based position-aware routing protocols in flying ad hoc networks," *Electron.*, vol. 11, no. 7, Art. no. 1099, 2022.

[2] D. S. Lakew, U. Sa'ad, N.-N. Dao, W. Na, and S. Cho, "Routing in flying ad hoc networks: A comprehensive survey," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 2, pp. 1071–1120, 2020.

[3] M. Y. Arafat and S. Moh, "Routing protocols for unmanned aerial vehicle networks: A survey," *IEEE Access*, vol. 7, pp. 99 694–99 720, 2019.

[4] A. Rovira-Sugranes, A. Razi, F. Afghah, and J. Chakareski, "A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook," *Ad Hoc Netw.*, vol. 130, Art. no. 102790, 2022.

[5] S. Devaraju, M. Parsinia, E. S. Bentley, and S. Kumar, "A multipath local route repair scheme for bidirectional traffic in an airborne network of multibeam FDD nodes," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 4, pp. 2983–2995, 2022.

[6] X. Li, F. Hu, J. Qi, and S. Kumar, "Systematic medium access control in hierarchical airborne networks with multibeam and single-beam antennas," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 2, pp. 706–717, 2018.

[7] M. Zhang, C. Dong, P. Yang, T. Tao, Q. Wu, and T. Q. Quek, "Adaptive routing design for flying ad hoc networks," *IEEE Commun. Letters*, 2022.

[8] B. Alzahrani, O. S. Oubbati, A. Barnawi, M. Atiquzzaman, and D. Alghazzawi, "UAV assistance paradigm: State-of-the-art in applications and challenges," *J. Netw. Comput. Applications*, vol. 166, Art. no. 102706, 2020.

[9] C. Pu, "Link-quality and traffic-load aware routing for UAV ad hoc networks," in *Int. Conf. Collaboration Internet Comput.* IEEE, 2018, pp. 71–79.

[10] A. Guillen-Perez and M.-D. Cano, "Flying ad hoc networks: A new domain for network communications," *Sensors*, vol. 18, no. 10, pp. 3571–3594, 2018.

[11] Q. Sang, H. Wu, L. Xing, H. Ma, and P. Xie, "An energy-efficient opportunistic routing protocol based on trajectory prediction for FANETs," *IEEE Access*, vol. 8, pp. 192 009–192 020, 2020.

[12] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 4, pp. 2624–2661, 2016.

[13] V. K. Jain, A. P. Mazumdar, P. Faruki, and M. C. Govil, "Congestion control in internet of things: Classification, challenges, and future directions," *Sustainable Comput.: Informatics Syst.*, vol. 35, Art. No. 100678, 2022.

[14] S.-W. Lee, S. Ali, M. S. Yousefpoor, E. Yousefpoor, P. Lalbakhsh, D. Javaheri, A. M. Rahmani, and M. Hosseinzadeh, "An energy-aware and predictive fuzzy logic-based routing scheme in flying ad hoc networks (FANETs)," *IEEE Access*, vol. 9, pp. 129 977–130 005, 2021.

[15] M. Johnston, C. Danilov, and K. Larson, "A reinforcement learning approach to adaptive redundancy for routing in tactical networks," in *IEEE Military Commun. Conf.* IEEE, 2018, pp. 267–272.

[16] A. A. Deshpande, F. Chiariotti, and A. Zanella, "SMURF: Reliable multipath routing in flying ad-hoc networks," in *Medit. Commun. Comput. Netw. Conf.* IEEE, 2020, pp. 1–8.

[17] M. Song, J. Liu, and S. Yang, "A mobility prediction and delay prediction routing protocol for uav networks," in *Int. Conf. Wireless Commun. Signal Processing.* IEEE, 2018, pp. 1–6.

[18] J. Liu, Q. Wang, C. He, K. Jaffrès-Runser, Y. Xu, Z. Li, and Y. Xu, "QMR: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks," *Comput. Commun.*, vol. 150, pp. 304–316, 2020.

[19] D. Liu, J. Cui, J. Zhang, C. Yang, and L. Hanzo, "Deep reinforcement learning aided packet-routing for aeronautical ad-hoc networks formed by passenger planes," *IEEE Trans. Veh. Tech.*, vol. 70, no. 5, pp. 5166–5171, 2021.

[20] Q. Lin, H. Song, X. Gui, X. Wang, and S. Su, "A shortest path routing algorithm for unmanned aerial systems based on grid position," *J. Netw. Comput. Appl.*, vol. 103, pp. 215–224, 2018.

[21] J.-A. Maxa, M. S. B. Mahmoud, and N. Larrieu, "Survey on UAANET routing protocols and network security challenges." *Adhoc Sensor Wireless Netw.*, vol. 37, 2017.

[22] E. Kuiper and S. Nadjm-Tehrani, "Mobility models for UAV group reconnaissance applications," in *Int. Conf. Wireless Mobile Commun.* IEEE, 2006, pp. 33–33.

[23] A. Bujari, C. T. Calafate, J.-C. Cano, P. Manzoni, C. E. Palazzi, and D. Ronzani, "Flying ad-hoc network application scenarios and mobility models," *Int. J. Distr. Sensor Netw.*, vol. 13, no. 10, pp. 1–17, 2017.

[24] G. Wang and Y. Qin, "MAC protocols for wireless mesh networks with multi-beam antennas: A survey," in *Future Inf. Commun. Conf.* Springer, 2019, pp. 117–142.

[25] I. Stevanovic, A. Skrivervik, and J. R. Mosig, "Smart antenna systems for mobile communications," Tech. Rep., 2003.

[26] H.-T. Chou, "An effective design procedure of multibeam phased array antennas for the applications of multisatellite/coverage communications," *IEEE Trans. Antennas Propag.*, vol. 64, no. 10, pp. 4218–4227, 2016.

[27] M. Parsinia, Q. Peng, and S. Kumar, "Distributed mode selection for FDD communication in multihop wireless networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 6, pp. 2921–2937, 2019.

[28] D. Johnson, "5G beam-steering antennas: More accurate less power hungry," *IEEE Spectrum*, 2018.

[29] S. Say, N. Aomi, T. Ando, and S. Shimamoto, "Circularly multi-directional antenna arrays with spatial reuse based MAC for aerial sensor networks," in *IEEE Int. Conf. Commun. Workshop*. IEEE, 2015, pp. 2225–2230.

[30] J. Wang, Y. Fang, and D. Wu, "Enhancing the performance of medium access control for WLANs with multi-beam access point," *IEEE Trans. Wireless Commun.*, vol. 6, no. 2, pp. 556–565, 2007.

[31] T.-S. Kim, G. Jakllari, S. V. Krishnamurthy, and M. Faloutsos, "An integrated routing and rate adaptation framework for multi-rate multi-hop wireless networks," *Wireless Netw.*, vol. 19, no. 5, pp. 985–1003, 2013.

[32] H. Gossain, T. Joshi, C. D. M. Cordeiro, and D. P. Agrawal, "DRP: An efficient directional routing protocol for mobile ad hoc networks," *IEEE Trans. Parallel Distr. Syst.*, vol. 17, no. 12, pp. 1438–1541, 2006.

[33] S. Garg, V. S. S. Kuchipudi, E. S. Bentley, and S. Kumar, "A real-time, distributed, directional TDMA MAC protocol for QoS-aware communication in multi-hop wireless networks," *IEEE Access*, vol. 9, pp. 26 343–26 361, 2021.

[34] O. Bazan and M. Jaseemuddin, "A survey on MAC protocols for wireless adhoc networks with beamforming antennas," *IEEE Commun. Surv. Tutor.*, vol. 14, no. 2, pp. 216–239, 2011.

[35] R. Sharma, G. Kadambi, Y. A. Vershinin, and K. Mukundan, "A survey of MAC layer protocols to avoid deafness in wireless networks using directional antenna," in *Mobile Comput. Wireless Netw.: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2016, pp. 1758–1797.

[36] D. T. C. Wong, Q. Chen, and F. Chin, "Directional medium access control (MAC) protocols in wireless ad hoc and sensor networks: A survey," *J. Sensor Actuator Netw.*, vol. 4, no. 2, pp. 67–153, 2015.

[37] J. Wang, Y. Zhang, and L. Jiang, "A novel time-slot allocation scheme for ad hoc networks with single-beam directional antennas," in *IEEE Int. Conf. Commun. Software Netw.* IEEE, 2015, pp. 227–231.

[38] P. Li, H. Zhai, and Y. Fang, "SDMAC: Selectively directional MAC protocol for wireless mobile ad hoc networks," *Wireless Netw.*, vol. 15, no. 6, pp. 805–820, 2009.

[39] A. El Masri, L. Khoukhi, and D. Gaiti, "A TDMA-based MAC protocol for wireless mesh networks using directional antennas," in *Int. Conf. Commun. Theory, Reliability, Quality of Service.* Citeseer, 2011, pp. 95–98.

[40] D. Hao and D. P. Liu, "A distributed TDMA-based MAC protocol for ad hoc networks with directional antennas," in *Int. Conf. Wireless Commun. Signal Processing.* IEEE, 2012, pp. 1–6.

[41] L. Shaohua and D.-H. Cho, "Directional antenna based time division scheduling in wireless ad hoc networks," in *IEEE Military Commun. Conf.* IEEE, 2008, pp. 1–7.

[42] Y.-x. Tian and K.-j. Wu, "A novel dynamic TDMA protocol for ad hoc networks using directional antennas," in *Int. Conf. Electric Information Control Engineering.* IEEE, 2011, pp. 65–69.

[43] J.-J. Chang, W. Liao, and J.-R. Lai, "On reservation-based MAC protocol for IEEE 802.11 wireless ad hoc networks with directional antenna," *IEEE Trans. Veh. Tech.*, vol. 60, no. 6, pp. 2669–2679, 2011.

[44] C. K. Toh, *Ad hoc mobile wireless networks: protocols and systems.* Pearson Education, 2001.

[45] A. A. Ateya, A. Muthanna, I. Gudkova, Y. Gaidamaka, and A. D. Algarni, "Latency and energy-efficient multi-hop routing protocol for unmanned aerial vehicle networks," *Int. J. Distr. Sensor Netw.*, vol. 15, no. 8, 2019. doi/10.1177/1550147719866392.

[46] B. H. Khudayer, M. Anbar, S. M. Hanshi, and T.-C. Wan, "Efficient route discovery and link failure detection mechanisms for source routing protocol in mobile ad-hoc networks," *IEEE Access*, vol. 8, pp. 24 019–24 032, 2020.

[47] S. Garg, "An adaptive and low-complexity routing protocol for distributed airborne networks," in *EWSN*, 2021, pp. 187–191.

[48] S. Kumar, V. S. Raghavan, and J. Deng, "Medium access control protocols for ad hoc wireless networks: A survey," *Adhoc Netw.*, vol. 4, no. 3, pp. 326–358, 2006.

[49] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proc. IEEE Int. Multi Topic Conf.* IEEE, 2001, pp. 62–68.

[50] S. Garg, A. Ihler, and S. Kumar, "Accurate link lifetime computation in autonomous airborne UAV networks," *arXiv:2202.00056 [cs.NI]*, 2022.

[51] M. Gharib, F. Afghah, and E. S. Bentley, "LB-OPAR: Load balanced optimized predictive and adaptive routing for cooperative UAV networks," *Ad Hoc Networks*, vol. 132, Art. no. 102878, 2022.

[52] F. Cadger, K. Curran, J. Santos, and S. Moffett, "A survey of geographical routing in wireless ad-hoc networks," *IEEE Commun. Surv. Tutor.*, vol. 15, no. 2, pp. 621–653, 2012.

[53] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," *RFC 3561*, 2003.

[54] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," *RFC 3626*, 2003.

[55] Z. J. Haas and M. R. Pearlman, "The performance of query control schemes for the zone routing protocol," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 4, pp. 167–177, 1998.

[56] N. El Houda Bahloul, S. Boudjit, M. Abdennebi, and D. E. Boubiche, "A flocking-based on demand routing protocol for unmanned aerial vehicles," *J. Comput. Science Tech.*, vol. 33, no. 2, pp. 263–276, 2018.

[57] X. Li and J. Yan, "LEPR: Link stability estimation-based preemptive routing protocol for flying ad hoc networks," in *IEEE Symp.Comput. Commun.* IEEE, 2017, pp. 1079–1084.

[58] B. Sliwa, C. Schüler, M. Patchou, and C. Wietfeld, "Parrot: Predictive ad-hoc routing fueled by reinforcement learning and trajectory knowledge," in *IEEE Veh. Tech. Conf.* IEEE, 2021, pp. 1–7.

[59] A. M. Rahmani, S. Ali, E. Yousefpoor, M. S. Yousefpoor, D. Javaheri, P. Lalbakhsh, O. H. Ahmed, M. Hosseinzadeh, and S.-W. Lee, "OLSR+: A new routing method based on fuzzy logic in flying ad-hoc networks (FANETs)," *Veh. Commun.*, p. 100489, 2022.

[60] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *Wireless Commun. Mobile Comput.*, vol. 6, no. 7, pp. 969–988, 2006.

[61] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A framework for reliable routing in mobile ad hoc networks," in *IEEE INFOCOM*, vol. 1. IEEE, 2003, pp. 270–280.

[62] S. Kumar, S. Khimsara, K. Kambhatla, K. Girivanesh, J. D. Matyjas, and M. Medley, "Robust on-demand multipath routing with dynamic path upgrade for delay-sensitive data over ad hoc networks," *J. Comput. Netw. Commun.*, vol. 2013, 2013.

[63] P. Li, L. Guo, and F. Wang, "A multipath routing protocol with load balancing and energy constraining based on aomdv in ad hoc network," *Mobile Netw. Appl.*, pp. 1–10, 2019.

[64] M. N. Anjum and H. Wang, "Mobility modeling and stochastic property analysis of airborne network," *IEEE Trans. Netw. Science Engineering*, vol. 7, no. 3, pp. 1282–1294, 2019.

[65] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 2, pp. 1123–1152, 2015.

[66] O. Bouachir, A. Abrassart, F. Garcia, and N. Larrieu, "A mobility model for UAV ad hoc network," in *Int. Conf. Unmanned Aircraft Syst.* IEEE, 2014, pp. 383–388.

[67] Y. Wan, K. Namuduri, Y. Zhou, and S. Fu, "A smooth-turn mobility model for airborne networks," *IEEE Trans. Veh. Tech.*, vol. 62, no. 7, pp. 3359–3370, 2013.

[68] W. Wang, X. Guan, B. Wang, and Y. Wang, "A novel mobility model based on semi-random circular movement in mobile ad hoc networks," *Inform. Sciences*, vol. 180, no. 3, pp. 399–413, 2010.

[69] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for PCS networks," in *IEEE INFOCOM*, vol. 3. IEEE, 1999, pp. 1377–1384.

[70] J.-D. M. M. Biomo, T. Kunz, and M. St-Hilaire, "An enhanced Gauss-Markov mobility model for simulations of unmanned aerial ad hoc networks," in *Wireless Mobile Netw. Conf.* IEEE, 2014, pp. 1–8.

[71] 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*, vol. 7, pp. 81 057–81 105, 2019.

[72] Y. Zheng, Y. Wang, Z. Li, L. Dong, Y. Jiang, and H. Zhang, "A mobility and load aware olsr routing protocol for UAV mobile ad-hoc networks," in *Int. Conf. Infor. Commun. Technol.* IET, 2014, pp. 1–7.

[73] M. A. Abid and A. Belghith, "Asynchronous locally self adjusted routing protocol for mobile multi hop ad hoc networks," in *ACS/IEEE Int. Conf. Comput. Systems and Applications-AICCSA 2010*. IEEE, 2010, pp. 1–8.

[74] M. Belhassen, A. Belghith, and M. A. Abid, "Performance evaluation of a cartography enhanced OLSR for mobile multi-hop ad hoc networks," in *Wireless Advanced*. IEEE, 2011, pp. 149–155.

[75] M. Rondinone, J. Ansari, J. Riihijärvi, and P. Mähönen, "Designing a reliable and stable link quality metric for wireless sensor networks," in *Proc. Workshop Real-world Wireless Sensor Netw.*, 2008, pp. 6–10.

[76] Z. Guo, S. Malakooti, S. Sheikh, C. Al-Najjar, and B. Malakooti, "Multi-objective OLSR for proactive routing in MANET with delay, energy, and link lifetime predictions," *Applied Mathematical Modelling*, vol. 35, no. 3, pp. 1413–1426, 2011.

[77] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *Int. J. Netw. Management*, vol. 11, no. 1, pp. 3–30, 2001.

[78] Z. Li and Z. J. Haas, "On residual path lifetime in mobile networks," *IEEE Commun. Letters*, vol. 20, no. 3, pp. 582–585, 2016.

[79] Y. Huang, R. Xie, B. Gao, and J. Wang, "Dynamic routing in flying ad-hoc networks using link duration based MPR selection," in *IEEE Veh. Tech. Conf.* IEEE, 2020, pp. 1–5.

[80] M. Gerharz, C. de Waal, M. Frank, and P. Martini, "Link stability in mobile wireless ad hoc networks," in *IEEE Conf. Local Comput. Netw.* IEEE, 2002, pp. 30–39.

[81] G. Kuperman, R. Margolies, N. M. Jones, B. Proulx, and A. Narula-Tam, "Uncoordinated MAC for adaptive multi-beam directional networks: Analysis and evaluation," in *Int. Conf. Comput. Commun. Netw.* IEEE, 2016, pp. 1–10.

[82] J. Wang, H. Zhai, P. Li, Y. Fang, and D. Wu, "Directional medium access control for ad hoc networks," *Wireless Netw.*, vol. 15, no. 8, pp. 1059–1073, 2009.

[83] G. Jakllari, W. Luo, and S. V. Krishnamurthy, "An integrated neighbor discovery and MAC protocol for ad hoc networks using directional antennas," *IEEE Trans. Wireless Commun.*, vol. 6, no. 3, pp. 1114–1024, 2007.

[84] Z. Zhang, "Pure directional transmission and reception algorithms in wireless ad hoc networks with directional antennas," in *IEEE Int. Conf. Commun.*, vol. 5. IEEE, 2005, pp. 3386–3390.

[85] E. Shihab, L. Cai, and J. Pan, "A distributed directional-to-directional MAC protocol for asynchronous ad hoc networks," in *IEEE GLOBECOM*. IEEE, 2008, pp. 1–5.

[86] M. Takata, M. Bandai, and T. Watanabe, "A directional MAC protocol with deafness avoidance in ad hoc networks," *IEICE Trans. Commun.*, vol. 90, no. 4, pp. 866–875, 2007.

[87] Y. Tu, Y. Zhang, and H. Zhang, "A novel MAC protocol for wireless ad hoc networks with directional antennas," in *IEEE Int. Conf. Commun. Tech.* IEEE, 2013, pp. 494–499.

[88] A. Bhatia and R. Hansdah, "Rd-tdma: A randomized distributed TDMA scheduling for correlated contention in WSNs," in *Int. Conf. Advanced Information Netw. Appl. Workshops*. IEEE, 2014, pp. 378–384.

[89] P. Djukic and S. Valaee, "Delay aware link scheduling for multi-hop TDMA wireless networks," *IEEE/ACM Trans.Netw.*, vol. 17, no. 3, pp. 870–883, 2008.

[90] A. Bhatia and R. Hansdah, "A classification framework for tdma scheduling techniques in WSNs," *arXiv:2002.00458 [cs.NI]*, 2020.

[91] Z. Bai, B. Li, Z. Yan, M. Yang, X. Jiang, and H. Zhang, "A classified slot re-allocation algorithm for synchronous directional ad hoc networks," in *Int. Conf. Heterogeneous Netw. Quality, Reliability, Security Robustness.* Springer, 2017, pp. 194–204.

[92] R. Gunasekaran, S. Siddharth, P. Krishnaraj, M. Kalaiarasan, and V. R. Uthariaraj, "Efficient algorithms to solve broadcast scheduling problem in wimax mesh networks," *Comput. Commun.*, vol. 33, no. 11, pp. 1325–1333, 2010.

[93] S. Kumar and V. Sharma, "Joint routing, scheduling and power control providing QoS for wireless multihop networks," in *National Conf. Commun.* IEEE, 2015, pp. 1–6.

[94] D. J. Vergados, N. Amelina, Y. Jiang, K. Kralevska, and O. Granichin, "Toward optimal distributed node scheduling in a multihop wireless network through local voting," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 400–414, 2017.

[95] C.-T. Chiang, H.-C. Chen, W.-H. Liao, and K.-P. Shih, "A decentralized minislot scheduling protocol (DMSP) in TDMA-based wireless mesh networks," *J. Netw. Comput. Appl.*, vol. 37, pp. 206–215, 2014.

[96] F. Hu, X. Li, E. S. Bentley, L. Hu, K. Bao, and S. Kumar, "Intelligent multibeam transmissions for mission-oriented airborne networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 2, pp. 619–630, 2018.

[97] T. Wang, X. Wang, X. Tian, and X. Gan, "HD-MAC: A throughput enhancing TDMA-based protocol in high dynamic environments," in *Int. Conf. Mobile Ad-Hoc and Sensor Netw.* IEEE, 2016, pp. 105–109.

[98] T.-T. Nguyen, T. Kim, and T. Kim, "A distributed TDMA scheduling algorithm using topological ordering for wireless sensor networks," *IEEE Access*, vol. 8, pp. 145 316–145 331, 2020.

[99] Y. Li, X. Zhang, T. Qiu, J. Zeng, and P. Hu, "A distributed TDMA scheduling algorithm based on exponential backoff rule and energy-topology factor in internet of things," *IEEE Access*, vol. 5, pp. 20 866–20 879, 2017.

[100] Y. Li, X. Zhang, J. Zeng, Y. Wan, and F. Ma, "A distributed TDMA scheduling algorithm based on energy-topology factor in internet of things," *IEEE Access*, vol. 5, pp. 10 757–10 768, 2017.

[101] S. Vasudevan, J. Kurose, and D. Towsley, "On neighbor discovery in wireless networks with directional antennas," in *Proc. IEEE 24th Annual Joint Conf.*, vol. 4, 2005, pp. 2502–2512. doi:10.1109/INFCOM.2005.1 498 535.

[102] F. N. Nur, S. Sharmin, M. Habib, M. Razzaque, M. Islam, A. Almogren, M. M. Hassan, A. Alamri *et al.*, "Collaborative neighbor discovery in directional wireless sensor networks: algorithm and analysis," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, no. 1, pp. 1–15, 2017.

[103] S. Garg, A. Ihler, E. Bentley, and S. Kumar, "A cross-layer, mobility and congestion-aware routing protocol for UAV networks," *IEEE Trans. Aerosp. Electron. Syst. (under 2nd round of review)*, 2022.

[104] Q. Luo and J. Wang, "Multiple QoS parameters-based routing for civil aeronautical ad hoc networks," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 804–814, 2017.

[105] J. Xie, Y. Wan, J. H. Kim, S. Fu, and K. Namuduri, "A survey and analysis of mobility models for airborne networks," *IEEE Commun. Surv. Tutor.*, vol. 16, no. 3, pp. 1221–1238, 2013.

[106] K. Wallace, "Cisco IP telephony flash cards: Weighted random early detection (WRED)," 2004.

[107] N. Toorchi, F. Hu, E. S. Bentley, and S. Kumar, "Skeleton-based swarm routing (SSR): intelligent smooth routing for dynamic UAV networks," *IEEE Access*, vol. 9, pp. 1286–1303, 2020.

[108] A. K. Dogra *et al.*, "Q-aodv: A flood control ad-hoc on demand distance vector routing protocol," in *Int. Conf. Secure Cyber Comput. Commun.* IEEE, 2018, pp. 294–299.

[109] N. Ahmad, S. Sethi, and M. Ahmed, "Cache-aware query-broadcast to improve QoS of routing protocols in MANETs," *Wireless Pers. Commun.*, vol. 113, no. 1, pp. 481–498, 2020.

[110] C. E. Perkins and E. M. Belding-Royer, "Quality of service for ad hoc on-demand distance vector routing," 2003.

[111] D. Jinil Persis and T. Paul Robert, "Review of ad-hoc on-demand distance vector protocol and its swarm intelligent variants for mobile ad-hoc network," *IET Netw.*, vol. 6, no. 5, pp. 87–93, 2017.

[112] P. Srinivasan and P. Kamalakkannan, "RSEA-AODV: route stability and energy aware routing for mobile ad hoc networks," *Int. J. Comput. Commun.*, vol. 8, no. 6, pp. 891–900, 2013.

[113] M.-Y. Su and C.-W. Yang, "A resilient routing approach for mobile ad hoc networks," in *Int. Conf. High Perform. Comput. Simul.* IEEE, 2015, pp. 617–622.

[114] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proc. Int. Conf. Netw. Protocols.* IEEE, 2001, pp. 14–23.

[115] A. O. Fapojuwo, O. Salazar, and A. B. Sesay, "Performance of a QoS-based multiple-route ad hoc on-demand distance vector protocol for mobile ad hoc networks," *Canadian J. Elect. Comput. Eng.*, vol. 29, no. 1/2, pp. 149–155, 2004.

[116] S.-J. Lee and M. Gerla, "AODV-BR: Backup routing in ad hoc networks," in *IEEE Wireless Commun. Netw. Conf.*, vol. 3. IEEE, 2000, pp. 1311–1316.

[117] T. K. Saini and S. C. Sharma, "Recent advancements, review analysis, and extensions of the AODV with the illustration of the applied concept," *Ad Hoc Netw.*, vol. 103, Article No. 102148, 2020.

[118] H. Johnston, "Cliques of a graph-variations on the Bron-Kerbosch algorithm," *Int. J. Comput. Inf. Sciences*, vol. 5, no. 3, pp. 209–238, 1976.

[119] A. V. Leonov and G. A. Litvinov, "Considering AODV and OLSR routing protocols to traffic monitoring scenario in FANET formed by mini-UAVs," in *Int. Sci.-Tech. Conf. Actual Problems Electron. Instrum. Eng.* IEEE, 2018, pp. 229–237.

[120] A. AlKhatieb, E. Felemban, and A. Naseer, "Performance evaluation of ad-hoc routing protocols in (FANETs)," in *IEEE Wireless Commun. Netw. Conf. Workshops.* IEEE, 2020, pp. 1–6.

[121] A. Nayyar, "Flying adhoc network (FANETs): simulation based performance comparison of routing protocols: AODV, DSDV, DSR, OLSR, AOMDV and HWMP," in *Int. Conf. Advances Big Data, Comput. Data Commun. Syst.* IEEE, 2018, pp. 1–9.

[122] T. Clausen, P. Jacquet, and L. Viennot, "Comparative study of routing protocols for mobile ad hoc networks," in *Med-hoc-Net*, 2002.

[123] J.-P. Vert and Y. Yamanishi, "Supervised graph inference," *Advances Neural Inf. Process. Syst.*, vol. 17, 2004.