AFRL-RI-RS-TR-2024-023

MOBILITY-ADAPTIVE, CROSS-LAYER PROTOCOLS FOR AIRBORNE NETWORKS WITH SINGLE/MULTI-BEAM DIRECTIONAL ANTENNAS

SAN DIEGO STATE UNIVERSITY FOUNDATION

FEBRUARY 2024

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2024-023 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
ELIZABETH S. BENTLEY
Work Unit Manager

/ S /
NICK P. KOWALCHUK
RIT Chief Engineer
Computing & Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
FEBRUARY 2024	FINAL TECHNICAL REPORT	SEPTEMBER 2020	SEPTEMBER 2023

4. TITLE AND SUBTITLE

MOBILITY-ADAPTIVE, CROSS-LAYER PROTOCOLS FOR AIRBORNE NETWORKS WITH SINGLE/MULTI-BEAM DIRECTIONAL ANTENNAS

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
N/A	FA8750-20-1-1005	62788F
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
		R30S

6. AUTHOR(S)

Kumar, Sunil; Hu, Fei

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

San Diego State University Foundation
5250 Campanile Drive
San Diego CA 92182

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Air Force Research Laboratory/RITGB 525 Brooks Road Rome NY 13441-4505	AFRL/RI	AFRL-RI-RS-TR-2024-023

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report discusses the design of cross-layer MAC (medium access control) and routing layer protocols for data transmission in airborne networks equipped with single/multi-beam directional antennas; these protocols consider the QoS (quality of service). First, a novel artificial intelligence (AI)-based directional MAC layer protocol is proposed, followed by two cross-layer routing protocols, including a proactive mobility and congestion-aware routing protocol and an AI-based directional routing protocol. Then a novel route repair protocol is proposed for a network of multi-beam FDD (frequency division duplex) nodes. The performance of these protocols is evaluated through network simulations in ns-3, Python and Matlab. The protocols discussed in this report are fully decentralized and successfully adapt to the dynamic topology of airborne unmanned aerial vehicles (UAV) networks, while achieving superior performance.

15. SUBJECT TERMS

Airborne Networks, UAV, QoS, MAC Protocols, Routing Protocols, Directional Antennas

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
U	U	U	SAR	100

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
ELIZABETH S. BENTLEY	N/A

PREVIOUS EDITION IS OBSOLETE.

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

1 SUMMARY

This project began on September 9, 2020, and ended on September 14, 2023. The project aims to develop networking algorithms and protocols for airborne networks (AN) that are equipped with single-/multi-beam directional antennas. The use of single-/multi-beam directional antennas can significantly enhance the link throughput due to extended communication range, improved spatial reuse and better link quality. However, this also presents new challenges in terms of beam steering, neighbor discovery, deafness, and capture, and maintaining a link between mobile nodes.

In this project, we have designed novel ad hoc network protocols (in routing and medium access control (MAC) layers) by considering the network characteristics, distribution of traffic flows and RF interference. *First*, we have designed a directional MAC protocol by using a novel exclusive-region (ER) based directional interference model, followed by spatio-temporal ER-map prediction.

Second, we have designed a proactive mobility and congestion-aware optimized link state routing (MCA-OLSR) protocol for a decentralized airborne network; this protocol incorporates a multi-metric route selection scheme, along with preemptive route switching to provide longer lasting routes while avoiding the broken and congested ones. Intelligent queue management is also designed in this protocol.

Third, we have designed an intelligent proactive routing protocol based on the directional heat map from spatio-temporal deep learning. In this protocol, a directional heat map is predicted which captures the traffic density distribution in different network regions and thus helps in selecting an optimal path, while avoiding the congestion areas and interference from neighboring directional nodes.

Finally, we have designed a local repair scheme for bidirectional multipath AODV (BAOMDV) routes for a network of airborne nodes consisting of multibeam directional antennas. These nodes use the frequency division duplex (FDD) mode of communication. This scheme preserves the overlapping and link-disjoint (or node-disjoint) characteristics of these routes, while reducing the expensive route rediscovery.

Simulations are important for the study of airborne networks. We have used the the discrete event simulator ns-3, as well as, Matlab to evaluate the performance of the above MAC and routing protocols.

Our results have been published in the following four journal articles.

Journal Papers:

1. Z. Chu, F. hu, J. Zhao, L. He, E. S. Bentley, and S. Kumar, *Exclusive-Region-Map Based Medium Access Control in Mobile Networks with Directional Antennas Through Deep Interference Learning*, IEEE Trans. Cogn. Commun. Netw., vol. 9, no. 4, pp. 1012-1024, Aug. 2023.
2. Z. Chu, F. Hu, E. S. Bentley, and S. Kumar, *Intelligent Routing in Directional Ad Hoc Networks Through Predictive Directional Heat Map from Spatio-Temporal Deep Learning*, IEEE Trans. Mobile Comput. (accepted), March 2023.
3. S. Garg, A. Ihler, E. S. Bentley, and S. Kumar, *A Cross-Layer, Mobility and Congestion-Aware Routing Protocol for UAV Networks*, IEEE Aerosp. Electron. Syst., vol. 59, no. 4, pp. 3778-3796, Aug. 2023.
4. S. Devaraju, M. Parsinia, E. S. Bentley, and S. Kumar, *Multipath Local Route Repair Scheme for Bidirectional Traffic in an Airborne Network of Multi-Beam FDD Nodes*, IEEE Trans. Aerosp. Electron. Syst., vol. 54, no. 4, pp. 2983-2995, Aug. 2022.

2 INTRODUCTION

Airborne networks (ANs) must be capable of supporting diverse air force missions, platforms, and communication needs. These networks could consist of a constellation of dozens of aircrafts and hundreds of unmanned aerial vehicles (UAVs) transporting mission-critical information, such as command and control (C2), text, imagery, and real-time collaborative voice and video. The target network must be capable of forming a topology that is matched to the missions, platforms, and data transmission needs, with minimum pre-planning and operator involvement. The wireless nodes should be capable of establishing connections with other node(s), whether airborne, in space, or on the surface, as needed. The network connections may be point-to-point, broadcast, or multipoint/multicast, and could be used to relay, translate, or gateway the information, as needed. The connections could be established either based upon a prearranged network topology, or autonomously without prearrangements, and dynamically as needs arise [1-3].

The mission-aware information representation and QoS-aware cross-layer network protocols are *key enablers* in effectively deploying an AN infrastructure. Assets need to (*i*) share the time-sensitive information among themselves for situational awareness, and (*ii*) transfer it to the remotely located command and control center. The challenge is to organize a *reliable* AN in the presence of dynamic topologies, heterogeneous nodes, directional and intermittent links, and dynamic spectrum allocation.

In this project, we have developed and tested the following mission- and QoS-aware cross-layer MAC and routing protocols for information transmission over ANs, including the nodes equipped with directional and multi-beam antennas.

2.1 Exclusive Region-Map based MAC Protocol

The main task of the medium access control (MAC) protocols in mobile ad hoc networks (MANET) is to ensure that all nodes in a neighborhood can fairly share the RF channel without access conflicts. MAC in MANETs with omni-directional antennas (OAs) has been widely studied [4]. However, MAC protocols in MANETs with directional antennas (DAs) have not yet been deeply studied. One of the reasons for this is the difficulty in determining the exact interference range from neighboring directional links. Without knowing such an interference range, a node cannot accurately determine its communication parameters, such as the data rate, queue size, antenna orientation, etc.

We explain the challenges of a *directional* MAC (DMAC) design with the help of an example. As shown in Figure 1, MAC protocols focus only on one-hop neighborhood communications. Therefore, although G-A-B is a part of a complete routing path, here we focus on E's neighborhood, where E is a reference node, and exclude node G from the MAC design. However, since each node could switch to its upstream or downstream node, any active communication node can be in Tx (transmission) or Rx (reception) mode. A total of three active directional links (A-B, C-D, and E-F) are shown in Figure 1. In an OA-based network, only one of those three links can be active at any given time. Thanks to the use of DAs, all those links can possibly transmit data simultaneously, if their directional RF interference does not have impact on each other.

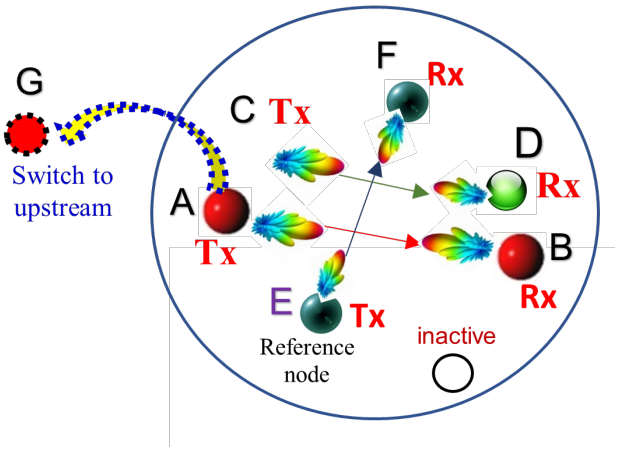


Figure 1. Example of directional MAC challenges.

We discuss the RF interference shape/range from a neighboring link. In Figure 1, both C-D and E-F can interference with link A-B. Note that the victim of interference is the receiver node (Rx), and not a transmitter (Tx). Here B is the Rx node, and its desired source is A (Tx). Other neighboring Tx sources (C or E) could interfere with it. However, C-D can cause a bigger interference since it is parallel to A-B, and C (a Tx node) is close to A. This means that B could receive signals from C more easily than from E. From B's viewpoint, what is the *exact* shape (called exclusive region (ER)) of the interference from each interferer (i.e., C or E)? Computing the ER shape can help us adjust the link A-B's properties, such as transmission starting time, packet sending rate, communication duration, queue size, etc.

The other issue is the *bursty packet loss* due to an interferer. For example, assume only A-B is active and there is no nearby Tx node in the beginning. To fully utilize the available bandwidth, A will use its allowable maximum rate to send out packets. Let us assume nodes C and D move closer to A and B, respectively. Since the two directional links (A-B and C-D) are close to each other, C (Tx) can cause a large interference to B (Rx), which makes the link A-B lose many packets within a short time. To avoid the bursty packet loss, a *proactive* MAC control should be used, i.e., a node's MAC protocol prepares for any possible ER changes from the nearby active links. Therefore, a node should be aware of its ER's evolutionary trend, in order to predict the next-time interference distribution around its location. If a Tx node (sender) predicts the approach of an interferer into the ER of its Rx node (receiver), it can prepare for the possible link outage by adjusting its sending rate, queue size, antenna orientation, and Tx/Rx schedule.

Existing directional MAC schemes [5-7] have considered multiple issues arising from the use of DAs, such as deafness [8], directional hidden terminal problem [9], directional neighbor discovery [10], etc. However, they did not consider the exact interference range from each nearby Tx node. In this project, we address the above challenges through the following three methods (see Figure 2 for the overall picture of our solution).

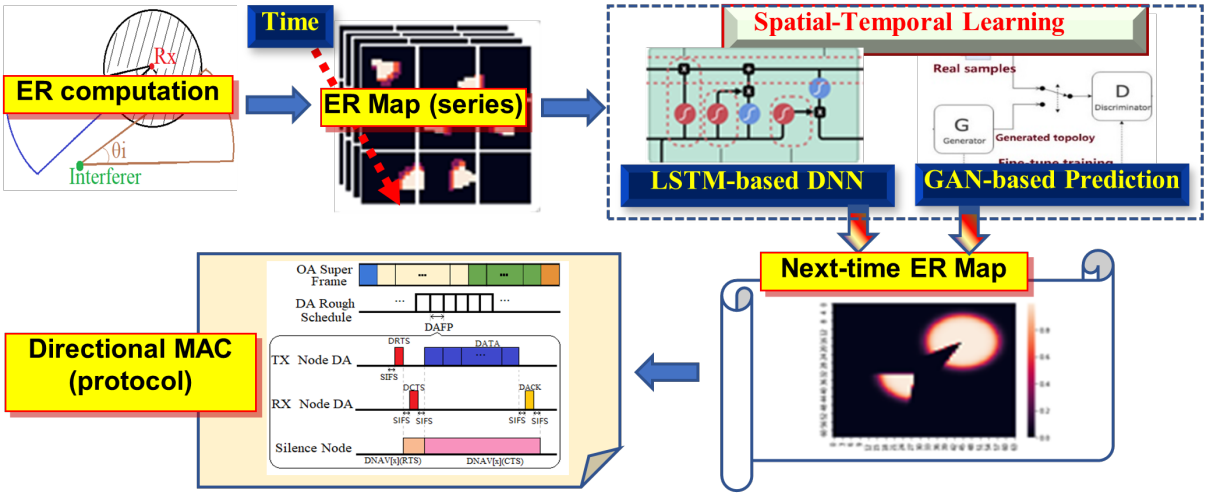


Figure 2. Overall Framework of our proposed DMAC based on learning/prediction of exclusive region.

(1) *ER+ model: Spatial expression of ER shapes in the format of a heat map (ER-map).* This helps to determine the exact interference ranges from nearby directional transmissions in all typical communication scenarios. Its shape could be a cone or circle, depending on the relative layout between two neighboring directional links. The ER map accurately reflects the impact of distance between any undesired Tx node and reference Rx node on the ER size. The ER map looks like a thermal image, in which colors of pixels represent the range and level of the interference. A reference node can use specially defined protocol messages to obtain the ER information from its 1-hop neighbors. Thus, a local ER map could be built and maintained by any node.

(2) *Learning model: Spatio-temporal ER map evolution prediction:* To achieve a proactive MAC control, we use an enhanced spatial-temporal residual networks (ST-ResNet+) [11] based deep neural networks (DNN) to describe both the spatial correlations (for ERs between neighboring links) and temporal correlations (capturing the time-varying ER dynamics). This ST-ResNet+ based DNN model will be used to predict the next-time ER map.

(3) *New: directional MAC protocol based on ER map prediction:* An ER-learning based MAC (**ERL-MAC**) is designed for channel reuse optimization. The ER map prediction results are used to determine the ERL-MAC operations, such as Tx/Rx schedule arrangement in the one-hop neighborhood, data transfer time duration, etc. The detailed MAC frame architecture and channel access operations are also defined in the protocol. The proposed scheme uses both random channel access and schedule-like Tx/Rx coordination among one-hop neighbors. Comprehensive simulations are conducted to verify the throughput efficiency of the ERL-MAC protocols as well as the accuracy of deep-learning-based ER-map prediction.

2.2 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 [12-18]. However, UAV networks experience varying network design configuration and communication constraints, which include 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 [12-14][18]. As a result, designing a robust communication mechanism for autonomous, decentralized (with no supervisory node) UAV networks is very challenging [12-25].

The widely used topology-based routing schemes require each node to maintain a routing table for its packet transmission and recompute the routes periodically [12-14][18]. 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 [19][23]. Therefore, some topology-based routing schemes (e.g., [12-15][18][21][25-26]) 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 [12-15][18]. For the same reason, machine learning driven approaches to predict future node trajectories (e.g., [16][23][27]) are difficult to use in autonomous UAV networks due to uncertain node mobility patterns [12][13][28]. 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., [15][19][21][25][29]) 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.

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*) the ability to 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*) consideration of practical node trajectories, and (*iv*) capability to work with decentralized network topologies.

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 Hop Count (*HC)*, Route Lifetime (*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.

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 optimized link state routing (OLSR) and multi-metric OLSR (MM-OLSR) protocols.

2.3 AI-based Directional Routing Protocol

We have designed an intelligent proactive directional routing protocol with the consideration of the global network traffic distribution in both time and spatial domains. Our goal is to proactively avoid congestion when multiple source-destination pairs are involved in active communications. The global traffic distribution could show certain evolutionary trends in both the time domain (such as traffic correlation between close time instants) and the spatial domain (such as a high correlation among the traffic flows close to each other). Machine learning models can be used to capture such temporal/spatial traffic trends.

In this scheme, we use machine learning models to perform accurate predictions of traffic distribution at different times and locations. The predicted traffic distribution is then used to design an intelligent routing protocol while avoiding network areas with heavy traffic. We use an example in Figure 3 to illustrate the above concept. To discover a new route between the source S1 and destination D1, the routing scheme generates a new path (marked in red). However, node A is already relaying the traffic for the S2-D2 pair (the green path). A hotspot will develop at node A if the total traffic is heavy, causing a congestion buildup leading to significant packet delays.

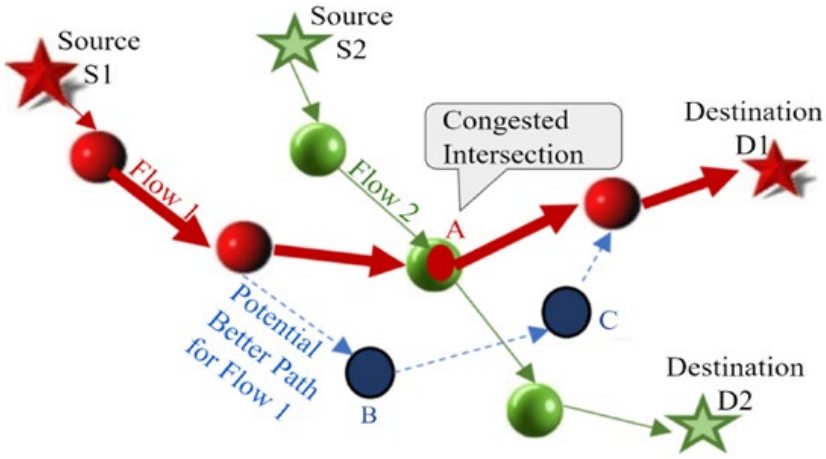


Figure 3. An example of network traffic distribution.

Our routing design is inspired by the intelligent transportation system (ITS) concept: A smart city can provide drivers with the real-time traffic distribution of the entire city. Thus, a driver can take a detour to avoid the heavily congested roads. The city traffic has strong spatial correlations among neighboring roads. The road traffic also shows temporal correlations; for example, the traffic jam on a road will not be cleared out soon. Therefore, the city traffic pattern can be described by a spatio-temporal model [30].

We propose an enhanced spatio-temporal deep learning model (st-DL+) to describe the data traffic distribution in the network. Here, it is important to capture the evolutionary trend of the traffic dynamics in each node/link. A new path should try to avoid the nodes that are already congested or are likely to get congested soon. Our routing algorithm uses the historical 'snapshots' of network traffic map to predict the traffic distribution in the next time-window. Such a prediction-based routing design is important since the path adjustment is often too late in conventional reactive (no prediction) routing schemes such as AODV and can lead to packet losses. Our scheme establishes a new route based on the understanding of the entire network's traffic map. In other words, we consider the potential impact of all active data flows on a new path, to avoid any busy network region that could cause long end-to-end delay and high packet loss rate. This is different from conventional cross-layer protocols [31-32], which consider the coupled design between the congestion control (in transport layer) and routing protocol (in network layer) for a single communication pair only, thus ignoring the impact from other communication pairs.

Besides different communication pairs, the impact of DA is considered in our routing protocol. With the emergence of 5G networks [33], millimeter waves [34], and Terahertz links [35], the DAs have been used to significantly reduce inter-node interference. If OAs are used, a new path must maintain a certain distance (typically over 2 hops away) from an existing path. On the other hand, the DAs allow a new path (for example, the path S2-D2 in Figure 3) to send data without much interference to a neighboring path (e.g., S1-D1 in Figure 4).

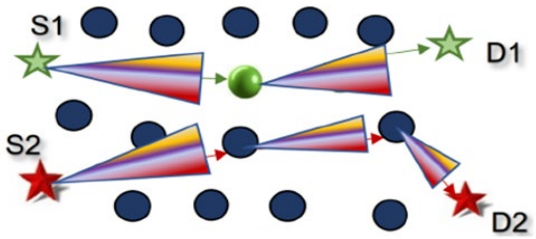


Figure 4. Illustration of routes using directional antennas.

In this project, we have built a directional heat map (DHM) that reflects both the traffic density distribution and directional signal propagation from DAs. A series of DHM snapshots can describe the traffic distribution dynamics across different regions (spatial) and time instants (temporal). As shown in Figure 5, our routing framework consists of the following three steps.

Step 1: (Sub-protocol 1) DHM establishment: This phase prepares for the establishment of DHM based on the available information such as each node's queuing status, link data rate, etc. The DHM contains information on the DA's attributes (such as beamwidth, orientation, etc.) and the congestion status (indicated by the queue size in each node) and can be calculated in a distributed manner. For example, in the cluster-based network, the cluster heads (CHs) could exchange the cluster member status information via CH-to-CH communication.

Step 2: (Sub-protocol 2) Prediction of the next-time traffic distribution: This phase uses the past information (such as node queue status, sender/receiver's signal levels, link utilization, etc.) to train a deep learning (DL) model. A DHM is predicted based on the DA information in each node. To use the learning model for the next-time DHM prediction, we propose a new st-DL+ model that takes past DHM snapshots as the input. Note that our st-DL+ model can capture the snapshot dynamics from three perspectives, i.e., trend (i.e., long-term evolution pattern), period (periodic traffic patterns), and closeness (i.e., short-term correlations between different snapshots).

Step 3: (Sub-protocol 3) Optimization algorithm for practical route establishment: Based on the predicted DHM snapshot, an optimization model is defined to find the best path that avoids the heavy-traffic regions, has low delay (in terms of the number of hops) and avoids the interference from the DAs of other paths' nodes. In this phase, the ground-truth DHM is used to update the st-DL+ model for higher prediction accuracy in the next learning round.

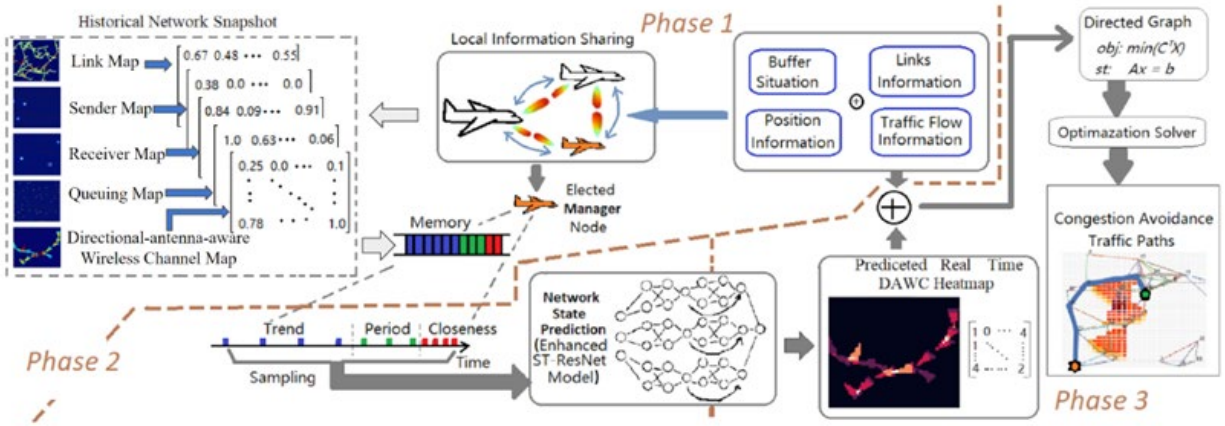


Figure 5. Directional heat map (DHM) based routing design (overview).

2.4 Local Route Repair Protocol for a Network of Multibeam FDD Nodes

The multi-hop transmission can use time division duplex (TDD) or frequency division duplex (FDD) modes, both having their pros and cons. FDD uses two distinct frequency channels for transmitting and receiving the packets at the same time, while TDD operates on only one frequency channel where the transmission and reception of the packet(s) is interleaved in time. Since FDD does not need a guard time interval to separate the transmission from the reception, it is more efficient in wide-area scenarios with long transmission links which introduces non-negligible signal propagation delays [36-37]. Some military systems, such as the widely used common data link (CDL), use FDD for the intelligence, surveillance, and reconnaissance (ISR) applications [38-39].

In an FDD system, the available frequency band (F) is typically divided into two distinct channels (F1 and F2), with a guard band to separate them [40]. The presence of a guard band can allow a node to cancel the self-interference by sufficiently attenuating the received side-band power from its own transmission by using the RF duplex filter. The node can therefore transmit the data on F1 and receive it on F2 at the same time. Since the transmissions on F1 and F2 channels in FDD systems are generally continuous, the receiver can feedback the most recent channel information to the transmitter. Thus, the delay is reduced for the channel information feedback, medium access control, and retransmission, which can enhance the system throughput [37].

We consider a decentralized, multi-hop airborne network consisting of the FDD nodes equipped with multi-beam antennas. This allows formation of multiple routes between a pair of source and destination nodes, where every forward and reverse route completely overlaps. Each of these routes can support bidirectional traffic. Earlier we proposed the Bidirectional Ad hoc On-Demand Multipath Distance Vector (BAOMDV) routing protocol for this purpose, where 'bidirectional communication' is supported by each link on the routes [37]. Two illustrations of multipath bidirectional routes (i.e., node-disjoint and link-disjoint routes) between the source node S and destination node D are shown in Figure 6. For example, node S can concurrently communicate with its two neighboring nodes (A and B) at the same time, if they fall into different beams of node S. Similarly, nodes C and D can also concurrently communicate with their respective neighbor nodes on different beams. As a result, two bidirectional flows can be supported between the source and destination pair in both topologies of Figure 6.

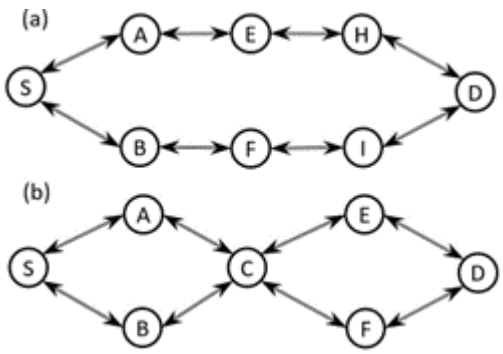


Figure 6. An illustration of (a) bidirectional node-disjoint multipath routes, and (b) bidirectional link-disjoint multipath routes. Here every link supports FDD-based bidirectional communication over a pair of frequency channels.

Node mobility can cause the routes to break frequently, forcing the source to rediscover new routes, which increases the routing overhead, packet losses and latency. Existing route maintenance and repair schemes do not address the issue of repairing the completely overlapped forward and reverse routes for transmitting bidirectional flows in a network of full-duplex nodes [41-50]. In fact, the existing schemes try to repair a route from both the upstream and downstream nodes that can generate routes with one or more non-overlapping portions. Note that, in a network of multibeam full-duplex nodes, the completely overlapped routes use fewer nodes and beams along the routes. Therefore, the repaired routes, with non-overlapping portions, cannot fully utilize the bidirectional communication capability of these nodes. This would not only waste the bandwidth but may also reduce the number of routes formed in the affected area, especially for

multi-beam nodes. From the MAC layer perspective too, the fully overlapped routes introduce less overhead and more efficiently support the bidirectional traffic.

We have proposed a local route repair scheme, called BAOMDV-LR, when these routes break due to node mobility. The proposed route repair scheme preserves the overlapping and link-disjoint (or node-disjoint) characteristics of the routes, while reducing the need for expensive route rediscovery. The proposed scheme significantly improves the overall network performance for bidirectional traffic (in terms of packet delivery ratio, end-to-end delay, and routing overhead) as compared to the BAOMDV and other existing routing schemes, especially at higher node speeds and data rates. The proposed BAOMDV-LR scheme has the following features:

- It locally repairs the routes while preserving their overlapping and node-disjoint (or link-disjoint) properties for supporting the bidirectional traffic.
- It maintains multiple routes between a source destination pair for a longer duration, and therefore reduces the need for frequent route discovery even at high node speeds.
- It achieves significant performance improvements, in terms of packet delivery ratio, end-to-end delay, and routing overhead.
- It is equally suited for in-band full-duplex nodes.
- It would work for any AODV or AOMDV based reactive routing protocol, if completely overlapped forward and reverse routes are formed between a pair of source and destination nodes.

3 METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 Exclusive Region-Map based Directional MAC: Algorithms and Design

3.1.1 Related Work

There are two major categories of DMAC protocols, i.e., schedule-based (i.e., TDMA - time division multiple access) and non-schedule based (i.e., CSMA – carrier sense multiple access). Both have drawbacks when used for directional ad-hoc networks. Our proposed design is a hybrid scheme.

Schedule-Based DMAC: Schedule-based schemes use time slots to arrange the Tx/Rx schedule of each node. Sedat in [51] introduced a TDMA MAC for directional networks, where a dynamic slot allocation table is built, and node location information is collected. Most collisions can be avoided here since each device knows its Tx time. However, the scheme assumes that the network has a center base-station that takes control of the schedule. Switching from a centralized to a distributed method has also been studied. A typical method is to elect a virtual header to take control of time slot allocations. In [52], a two-phase leading node election process was discussed. A fully distributed method without virtual center is discussed in [53] with multiple time slots to exchange synchronization and other control information. However, adjusting time slots in a fully distributed method may introduce considerable overhead [54]. Another distributed directional TDMA scheme was proposed in [55] by using a rank-based model.

Non-Schedule Based DMAC: A non-schedule MAC scheme was discussed in [56] to handle multi-beam transmissions in ad-hoc networks. The concept of a busy-tone was introduced in [57] that used two narrow-bandwidth, out-of-band busy tones to solve the hidden and exposed terminal problems. These two busy tones occupy two separate channels. During the handoff, a busy tone (BTt) is issued by a transmitter and a busy tone (BTr) is sent by a receiver. Any node sensing either BTt or BTr is not allowed to send an RTS. An improved busy-tone-based scheme was discussed in [58]. Two different busy-tone signal patterns, BT1 and BT2, were used to recognize deafness and interference issues. However, such a design still has an asymmetry-in-gain problem since different frequency bands can have different propagation properties.

Learning-based MAC Schemes: AI models have been used for MAC design. The reinforcement-learning-based schemes teach the best action for every single state. In [59] the deep reinforcement learning (DRL) agents were used to learn from past channel access collisions as well as successful transmissions. Each agent tries to figure out the strategy with persistent probability in packet transmissions. Another design in [60] allows the DRL agent to select the pre-built protocols from the pool and tune parameters to reach the optimal QoS performance. Through training, such a model can decide which protocol is the best one for throughput maximization. A similar scheme was presented in [61], where multiple nodes simultaneously use DRL agents to learn the optimal channel access by taking the actions of sending or holding packets in each time slot. A weakness of the above designs is the difficulty in defining states and actions. For example, in [59], there might be no optimal action of selecting the best candidate protocol to achieve the high throughput.

Our Design: Our proposed ERL-MAC scheme is a hybrid design with the advantages of both CSMA and TDMA. An ER map prediction model is used to help determine the Tx/Rx schedules for the next time window. OAs are used to exchange protocol messages through a low-frequency band (thus an OA has a longer signal propagation range than a DA and can more easily broadcast messages to other nodes). DAs are used for data transmission only. The ST-ResNet DNN model is used for a proactive MAC adjustment based on the ER prediction results.

3.1.2 The ER Model

In this section, we first explain the basic concept of an ER model in directional ad hoc networks [5]. A modified Sigmoid function is used to calculate the exact interference levels for two directional links close to each other. We then extend the common ER model to ER+ model by considering line-of-sight (LoS) and non-LoS radio paths. Finally, the concept of an ER map is proposed to describe the distribution of directional interference levels in the one-hop neighborhood. Such a map is used to analyze the evolution trends in each traffic hotspot with directional interference.

3.1.2.1 The ER Concept

The ER is a space model that indicates the keep-off areas in which the RF signals from a neighboring Tx node can interfere with a Rx [5]. Since a nearby transmitter inside the ER of the reference node becomes a serious interferer, all the neighboring Tx nodes must be kept outside the ER of the reference node.

For the directional networks, the area/shape of the ER is closely related to the antennas' orientation and beamwidth in both the reference node (Rx) and the interferer (Tx). To mathematically define ER, a parameter (a constant to reflect the quality of antenna signals) is defined as [5]

$$\gamma = \frac{kGP}{NB} \tag{1}$$

where G is the cross-correlation between any two concurrent link transmissions: one link is between the desired sender (Tx) and the target receiver (Rx); the other link is between the interferer node (Tx) and its Rx. P is transmission power of the interferer. N is one-sided spectral density of the white Gaussian noise. B is the channel bandwidth. k is a constant that depends on the signal wavelength λ as follows: $k \propto (\frac{\lambda}{4\pi})^{\alpha}$. The constant α is the path loss exponent depending on the RF propagation environment. The antenna gain is also a critical parameter.

We define the following variables throughout this section: (1) Reception gain within the beamwidth Θ , denoted as G_{RM}; (2) Rx gain outside Θ , denoted as G_{RS}; (3) Transmission main-lobe gain, denoted as G_{TM}; (4) Tx side-lobe gain, denoted as G_{TS}.

The worst-case scenario is when the interferer as well as the receiver are located within each other's beamwidth. In this case, the ER is a sector with the radius R and angle θ. The value of R can be calculated based on eq. (1) [5]:

$$R_1 = (\gamma G_{TM} G_{RM})^{1/\alpha} \tag{2}$$

Note that there could be a total of four ER scenarios, depending on: (1) whether the interferer is located within the beamwidth of the receiver, and (2) whether the receiver is located within the beamwidth of the interferer. Please refer to [5] for the mathematical models of those four scenarios. Note that all those four ER scenarios could exist simultaneously for a particular Rx node, since there could be multiple senders (interferers) in its neighborhood.

3.1.2.2 Extended ER Model (ER+): Handling Line-of-Sight (LoS) and NLoS

The LoS path exists between the sender and receiver when there is no obstacle between them. Besides LoS, the non-LoS (NLoS) paths may exist due to the radio reflection from nearby objects. The existence of LoS and NLoS paths can cause the "distortion" of the basic single-path ER shape. To model the effect of LoS and NLoS on the ER range, we propose a method called "virtual node + combination (VNC)." As shown in Figure 7(a), although the reference node wants to communicate with the desired sender, the interferer introduces both LoS and NLoS interference paths. To model such a complex ER mode, VNC uses the following three steps:

(1) ***Assumption of virtual nodes:*** For LoS and NLoS paths, VNC starts from the "reference node" and traces back each of those paths. This method assumes there is a virtual source at the other end of each path. It is only required to trace back the path along a straight line without considering the entire reflected path since only the final ER effect of each path needs to be considered. In Figure 7(a), two virtual nodes (i.e., a Virtual NLoS node and a Virtual LoS node) are established.

(2) ***ER calculations:*** We replace the interferer with virtual nodes and calculate the ER for each individual virtual node. Here, two ER ranges (each is a regular circle in this case) can be obtained for those two virtual nodes.

(3) ***ER combination:*** Due to the use of directional antennas, each ER is either a circle or a cone. If all the ERs are circular type, the ER with the largest radius is used as the final combined ER range (Figure 7(b)); If both circular and cone ERs are present, the final combined ER will be the total area of those ERs.

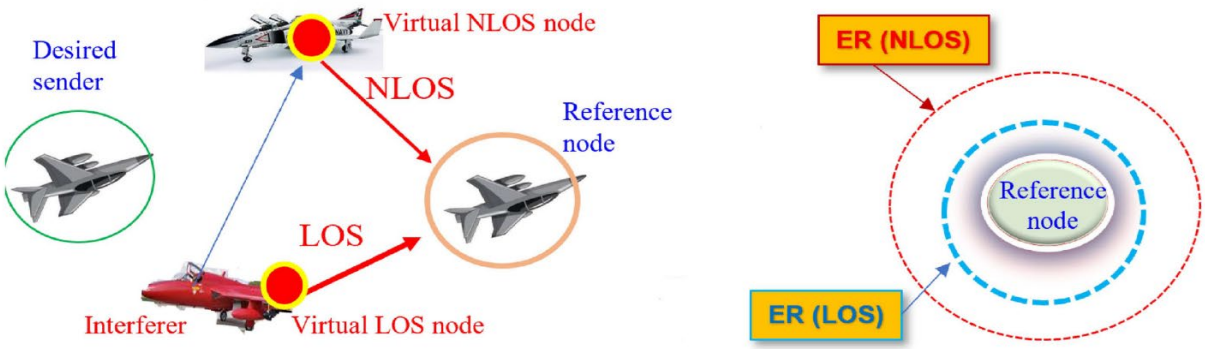


Figure 7. ER+ (virtual node + combination (VNC)-based ER model for LoS/NLoS).

3.1.2.3 Interference Level and ER Closeness Evaluation

In our DMAC scheme, the directional interference can be estimated by measuring the distance between an interferer (Tx) and the ER sector area of the reference node (Rx). To mathematically represent a potential interferer's proximity to an ER, we utilize the Sigmoid function which produces soft transactions from the minimum to maximum output when the input value is climbing up. We use an enhanced Sigmoid function such that it takes the interferer's location as input and its output approaches the maximum value as the interferer gets closer to the ER region. To start with, we use a general Sigmoid function as:

$$f(x) = \frac{1}{1+\exp(-x)} \tag{3}$$

The function has a value of 0.5 when x is 0; it approaches the minimum value of 0 (the maximum value of 1) when x is decreasing (increasing). To adjust the maximum value of the function, we multiply the function f with a constant c_1, that is,

$$f(x, c_1) = \frac{c_1}{1+\exp(-x)} \tag{4}$$

At this moment, $f(x, c_1)$ is centered with a value equal to half of the maximum c_1 at y-axis (i.e., a vertical line x = 0). The function increases from 0 to c_1 with a predefined increasing slope. Subtracting a constant c_2 from the term in the exponential function can shift the Sigmoid function to the right with a distance c_2. Dividing this term by a constant δ can adjust the function's slope. It generates the Sigmoid function as below:

$$f_1(x, c_1, c_2, \delta) = \frac{c_1}{1+\exp(\frac{-x+c_2}{\delta})} \tag{5}$$

To specify how far the Sigmoid function should be shifted and how steep the increasing slope is, we further require that: (1) the Sigmoid function value reaches its maximum value c_1 when $x = x_{min}$; (2) the Sigmoid function's output value is $\eta_1 c_1$ when the input x = $\eta_2 x_{min}$. That brings:

$$\frac{c_1}{1+exp(\frac{-\eta_2 x_{min}+c_2}{\delta})} = \eta_1 c_1 \tag{6}$$

Solving the expression for c2:

$$c_2 = \eta_2 \times x_{min} + \delta \times ln(\frac{1}{\eta_1} - 1) \tag{7}$$

The Sigmoid function we defined above is increasing from 0 to the maximum as the input increases. We can also derive the format for an adjusted Sigmoid function with the decreasing trend as below:

$$f(x, c_1, c_2, \delta) = \frac{c_1}{1+\exp(\frac{x-c_2}{\delta})} \tag{8}$$

To specify the value of c_2 in the above decreasing Sigmoid function, similar assumptions can be made: (1) the Sigmoid function maintains its max value c_1 until $x = x_{max}$; (2) the Sigmoid

function's output value is $\eta_1 c_1$ when the input $x = (2 - \eta_2)\, x_{max}$. Here η_2 is a number between 0 and 1, $(2 - \eta_2)$ results in a x value larger than x_{max} . This results in the following equation:

$$\frac{c_1}{1 + exp(\frac{(2-\eta_2)x_{max}+c_2}{\delta})} = \eta_1 c_1 \tag{9}$$

We then obtain the value of c_2 as:

$$c_2 = (2 - \eta_2) \times x_{max} - \delta \times ln(\frac{1}{\eta_1} - 1) \tag{10}$$

In conclusion, two modified Sigmoid functions, f_1 and f_2, are defined below:

$$f_1(x, x_{min}, \eta_1, \eta_2, c_1, \delta) = \frac{c_1}{1 + exp(\frac{-x+c_2}{\delta})} \tag{11}$$

$$f_2(x, x_{min}, \eta_1, \eta_2, c_1, \delta) = \frac{c_1}{1 + exp(\frac{x-c_2'}{\delta})} \tag{12}$$

where the parameters c_2 and c_2' can be expressed as:

$$c_2 = \eta_2 \times x_{min} + \delta \times ln(\frac{1}{\eta_1} - 1) \tag{13}$$

$$c_2' = (2 - \eta_2) \times x_{max} - \delta \times ln(\frac{1}{\eta_1} - 1) \tag{14}$$

The function f_1 is a Sigmoid function with the input x and has the maximum value bounded by c_1. f_1 increases with x until it reaches a threshold x_{min} (the threshold is set such that f_1 can reach the maximum value). Other parameters, such as δ, η_1, η_2, can determine the slope/shape of the curve. In general, both values of η_1 and η_2 should be within 0 to 1. Decreasing η_1 or increasing η_2 will both generate sharper, step-increasing curve for f_1. Here, δ is a number less than (but close to) 1. Larger δ will result in a slower function growth rate. Function f_2 is a decreasing Softmax function with all parameters acting similarly to f_1 case.

Based on the modified sigmoid function, we can evaluate how close a node N to a sector area through the function g:

$$g(\phi, \theta, v, \eta_{\theta 1}, \eta_{\theta 2}, c_{1\theta}, \delta_\theta, d, R, \eta_{r1}, \eta_{r2}, c_{1r}, \delta_r) = g_1 \times g_2 \times g_d$$
$$= f_1\left(\phi, v - \frac{\theta}{2}, \eta_{\theta 1}, \eta_{\theta 2}, c_{1\theta}, \delta_\theta\right) \times f_2\left(\phi, v + \frac{\theta}{2}, \eta_{\theta 1}, \eta_{\theta 2}, c_{1\theta}, \delta_\theta\right) \times f_1(d, R, \eta_{r1}, \eta_{r2}, c_{1r}, \delta_r) \tag{15}$$

Figure 8 shows a geometry for the understanding of function g in the polar coordinate system. Assume the ER has the shape of a sector with angle θ and radius R (polar coordinates). Here, θ is the beamwidth corresponding to the above case 1 or 2. The radius R has been calculated before. The interferer node N is in a position with angle ϕ and length d (in polar coordinates). The first term g_1 in g can be expanded as:

$$g_1 = f_1\left(\phi, v - \frac{\theta}{2}, \eta_{\theta 1}, \eta_{\theta 2}, c_{1\theta}, \delta_\theta\right) = \frac{c_{1\theta}}{1 + exp(\frac{-\phi + c_3}{\delta_\theta})} \quad (16)$$

where,

$$c_3 = \eta_{\theta 2} \times \left(v - \frac{\theta}{2}\right) + \delta_\theta \times ln(\frac{1}{\eta_{\theta 1} - 1}) \quad (17)$$

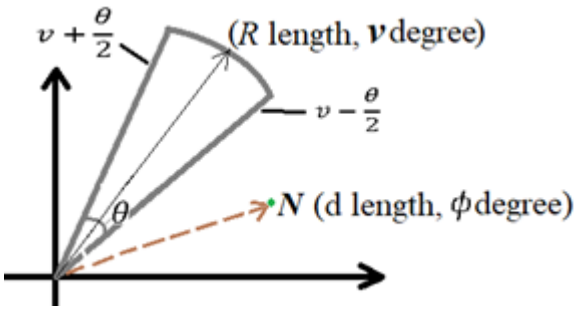


Figure 8. Measure Node-to-ER Closeness level.

g_1 approaches to the upper bound $c_{1\theta}$, as the value of angle ϕ of N becomes smaller than the upper angle bound of the sector, i.e., $(v - \frac{\theta}{2})$. The third term g_d can be expanded in a similar way:

$$g_d = f_1(d, R, \eta_{r1}, \eta_{r2}, c_{1r}, \delta_r) = \frac{c_{1r}}{1 + exp(\frac{-d + c_3''}{\delta_r})} \quad (18)$$

where

$$c_3'' = \eta_{r2} \times R + \delta_r \times ln(\frac{1}{\eta_{r1} - 1}) \quad (19)$$

g_d reaches c_{1r} as the N's length d decreases and approaches the ER sector radius R.

In short, the product of g_1 and g_2 is zero if N stays outside the two angles' boundaries; g_d is zero if N has a longer radius than the radius of the sector, i.e., $d > R$.

To simplify the model and parameters, we take $\delta_\theta = \delta_r = 0.80$, $\eta_{\theta 1} = \eta_{\theta 2} = 0.95$, $c_{1r} = c_{1\theta} = 1$.

Given the position of the interferer node N and the shape of ER, a high g value (close to 1) means that the interferer stays within the ER region and causes serious interference to the reference Rx. In our scheme, a value of g larger than the threshold is regarded as a warning sign for serious directional interference. Please note that four types of ER have different radii R and angles θ. This means that four combinations of R and θ values need to be applied to the above calculations.

3.1.2.4 Spatial Representation of ER

Based on the above ER model and closeness evaluation function g, we propose a spatial representation (called an ER map) for all ER regions around each node with active communications. Each Tx node in a link pair generates the corresponding ER spatial map around its surrounding area (i.e., 1-hop signal range). Instead of simply using binary values (1 and 0) to represent the affected and non-affected areas (in terms of directional interference), we use the ER closeness function g to provide a smooth transition from unaffected to affected areas. Hence, the nodes that are getting close to an ER region (but not entering it yet) can be warned.

In our scheme, each node is responsible for the generation and maintenance of the ER spatial maps for its involved active links that may belong to different routes. Corresponding map information will be saved in the local buffer. Such a map can evolve with time due to the change of node positions or Rx/Tx pairs. It is critical to capture the *evolutionary trend* of such a time-varying map to design a high-throughput MAC protocol for directional networks.

3.1.2.5 AI-based Prediction of ER Map

Directional networks need a proactive MAC control. For example, if a Rx node knows that an undesired Tx node (i.e., an interferer) will enter its ER in the next time window, the Rx node can either send out a warning message to its own (desired) Tx node, or it can pause its communication for a while until that Tx node passes by. Such a proactive MAC protocol can avoid link outage by managing its queue setup and Rx/Tx schedule.

To achieve the proactive MAC, it is critical to predict the evolution trend of an ER map. In this section, we describe an *enhanced spatio-temporal residual networks (ST-ResNet+)* based model, which can capture both spatial and temporal evolution trends of the ER map.

The basic ST-ResNet model was initially designed to predict the city vehicle flows [11]. The model can explore long-term, mid-term, and short-term trends from the past time series data. Figure 9 shows an example of data series with long-term linearly increasing trend, mid-term periodic changes, and short-term high-frequency oscillation.

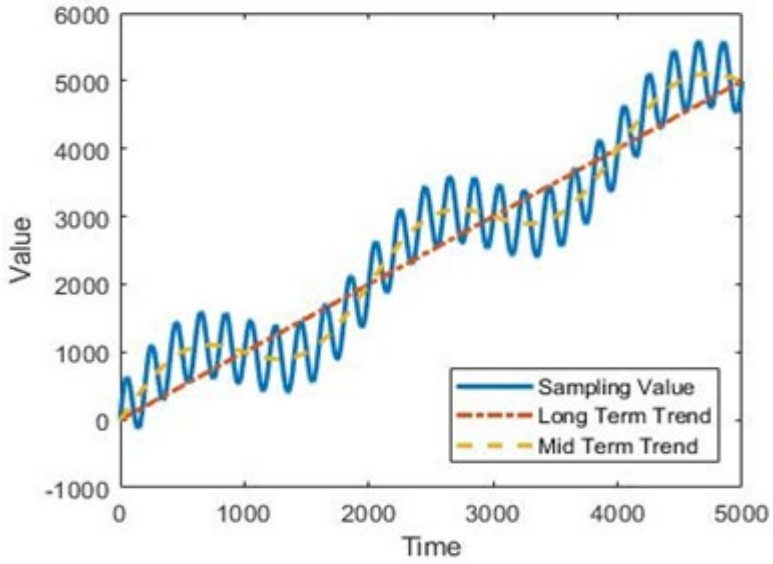


Figure 9. Data with Long-, mid- and short-term trends.

A prediction model that can capture those three trends can describe the evolution dynamics for the ER map of a mobile node. The long-term sampling considers ER evolutions within 60 seconds, which enables the model to capture the ER changes due to node mobility. The mid-term sampling considers ER conditions in 30 seconds, and the short-term sampling considers ER conditions in 5 seconds. The mid-term and short-term ER changes mainly come from antenna orientation variations.

For each ER case discussed, we use a separate ST-ResNet model with an identical structure to be trained. Our model's architecture is shown in Figure 10. Its inputs are ER maps at different time instants. The model outputs the predicted next-time ER spatial map. In our case, the ER map in the input and output layer of the DNN has a size of n × n pixels.

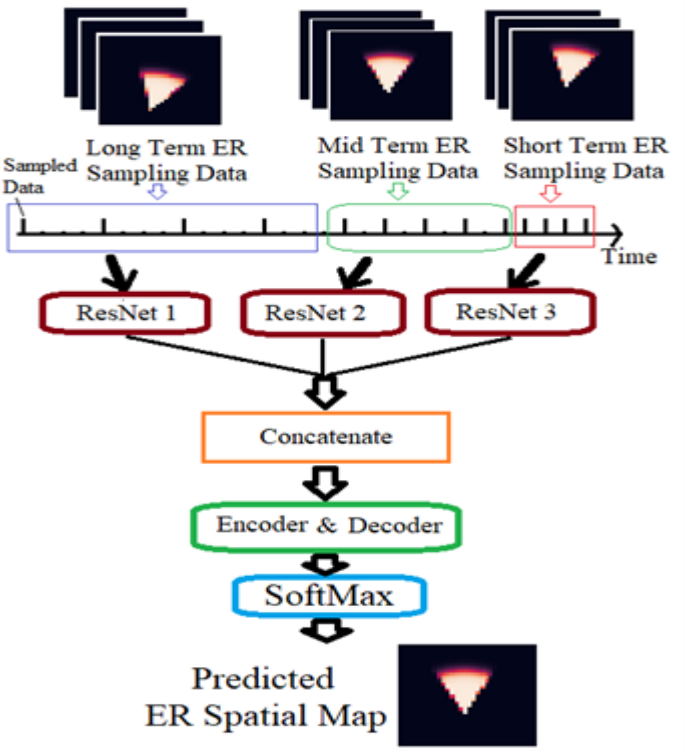


Figure 10. ST-ResNet model for ER map prediction.

Enhanced ST-ResNet Model: We further extend the basic model to ST-ResNet+, which has a parallel structure with three separate ResNets (each of them takes the inputs from long-, mid-, or short-term ER maps, separately). Those ResNets have the same structure, as shown in Figure 11. Each ResNet is stacked with ResNet units (highlighted in the rectangle box). The output Y of a ResNet unit can be represented as a function of input X as below. Here $F(X)$ is the equivalent function for all the layers' operations, including normalization, activation, and convolution. The ResNet unit adds the input value to the output layer to handle the vanishing gradient problem:

$$Y(X) = F(X) + X \tag{20}$$

After three ResNets perform parallel extractions of the characteristics from the input ER data, a concatenation process is used to stack all three characteristics matrices together in the time domain. To further explore the relationship between the long-, mid-, and short-term matrices, we append a concatenation layer to the encoder. After that a decoder is used to restore the encoded characteristics matrix into the output layer's prediction results.

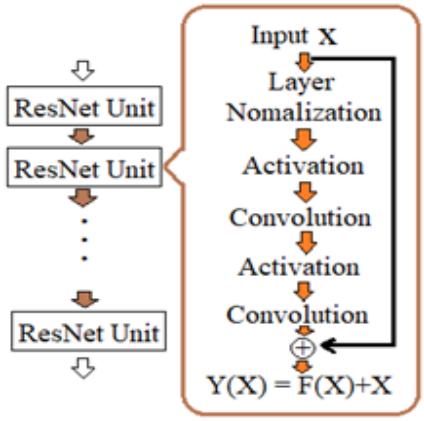


Figure 11. ResNet structure.

Figure 12 shows such an encoder / decoder layer design. The encoder compresses and extracts the input information. In each max pooling and convolution process, it doubles the data size in the last dimension and reduces the sizes of the first two dimensions by half. The output of the encoder is a characteristics matrix (with a size of $8 * 8 * 512$), which is also the input to the decoder.

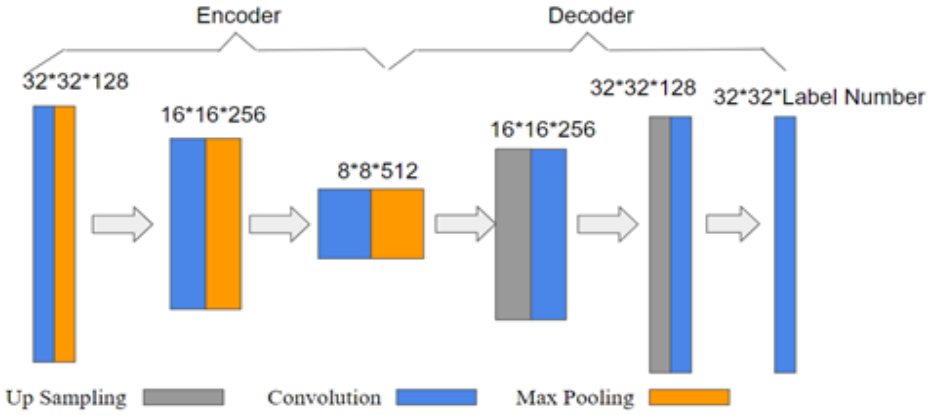


Figure 12. Encoder and decoder layer design.

The decoder adopts an inverse process of the encoder. It generates the output by doubling the first two dimensions and reducing the last dimension. Bilinear upsampling and convolution are applied to such an iteration process. The output of the decoder adopts a user-defined size, $m \times m \times k$, where m is the width of the targeted predicted ER spatial map and k is the label number. Instead of representing the exact pixel values of the ER spatial map as a floating-point number between 0 and 1, we use 5 labels to indicate different value levels. The conversion of values into different levels can significantly decrease the model complexity and the total number of repetitive layers required in the ST-ResNet model. Because the prediction results are labels instead of floating-point numbers, a SoftMax function is used at the end of the entire ST-ResNet model to select the label with the highest probability as the prediction result.

3.1.3 ERL-MAC Protocol Design

We assume that each node is equipped with both an OA and a DA. The OA is responsible for the protocol message exchanges (such as sharing ER information), and the DA is used for data transmission only. To avoid signal interference between the OA and DA, the OA operates in a lower frequency band (such as the license-free band, 868 MHz) to achieve a longer RF signal range. The separation of DA and OA enables high data throughput since the DA does not need to get involved in the control message exchanges.

3.1.3.1 OA-based ER Information Sharing

We first describe the OA-based protocol message exchange. It uses an OA superframe structure shown in Figure 13 for the sharing of ER information. The OA superframe starts with synchronization bits and ends with the guard field. It contains two different transmission windows: contention slots and beacon slots, where the contention slots are used by the nodes to join the information broadcasting schedule. A new node can use omni-directional antenna contention frames (OACFs) to join the protocol message transmission schedule. Beacon slots are used for a node to share its local information with its neighbors. Each beacon slot can be occupied by only one node. In our ERL-MAC protocol, each node can occupy multiple beacon slots corresponding to different links, since a node can participate in multiple routing flows simultaneously. During each beacon slot, the link information is shared through omni-directional antenna beacon frames (OABFs). The functionalities of OACFs and OABFs are discussed below.

Figure 13. OA-based message exchange: OA superframe structure.

Before a node broadcasts its local information through an OABF, it submits a beacon slot request message through an OACF in one of its contention slots. Each successful OACF is assigned a beacon slot. If two OACFs are announced in the same contention time slot, a random back-off will occur and nodes will try to access contention slots again during the next superframe. As shown in Figure 14, nodes A and C both request access to the OA channel by randomly selecting a contention slot to send out their respective OACF. If A and C claim different beacon slots, they can successfully add themselves into the beacon slot list in the next superframe. If they claim the same beacon slot (i.e., with the same value in Table 1, Field 5), the following three rules are followed: (1) The OACF with a higher QoS priority can access the beacon slot first; (2) With the same priority, the first claimed OACF has the priority to use the slot; (3) Any OACF that fails to access the beacon slot will be re-announced during the next OA superframe contention period with its priority incremented by 1.

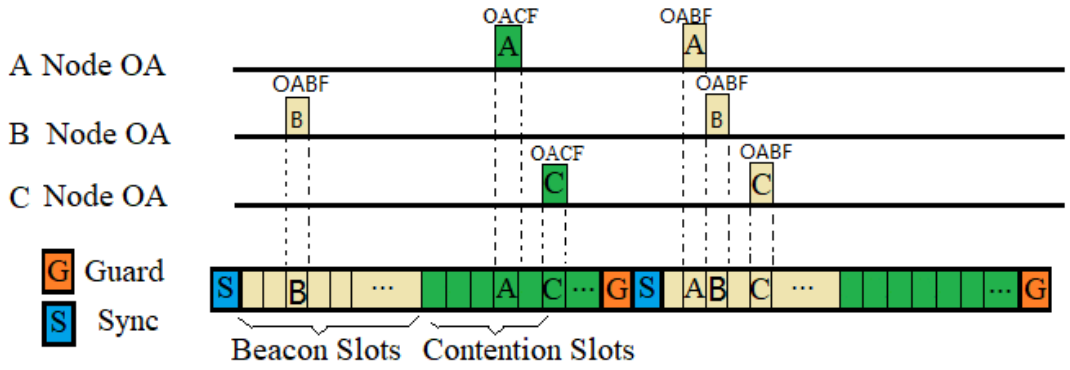


Figure 14. OACF contention scenario.

Table 1 shows the OACF format. Note that it contains two MAC addresses: OA-net MAC address (for the OA-based sub-net in which the protocol messages are exchanged via OAs) and DA-net Tx node MAC address (for the subnet in which DAs are used to send data packets).

Table 1. Format of omnidirectional antenna contention frame (OACF)

(1) Type	(2) OA-net MAC Address	(3) OA TX Power	(4) DA-net Tx MAC Address	(5) Preferred OA Time Slot Index	(6) Priority	(7) Checksum
1 byte	6 bytes	2 bytes	6 bytes	2 bytes	1 byte	4 bytes

A node uses OABF to broadcast its link information, as shown in Table 2. It contains the Tx node's antenna and position information. Through the analysis of the neighbor's Tx power level and its own received signal strength (RSS), each node can estimate the distance to its neighbor. Each OABF also includes the information for a directional link. Particularly, Field 5 has the DA link pair's MAC addresses. Field 6 gives the DA link's remaining transmission time (in the unit of ms). Field 7 contains the node's predicted ER information for the next superframe. In addition, the Rx and Tx nodes' positions in each ER map are also included. Here we use 4 labels (1 to 4) to represent the ER levels from the lowest to highest level. These 4 levels can be represented by 16 × 16 bits. Any Rx node could be interfered with by multiple ER source nodes and thus all 4 ER scenarios could coexist. This requires the representation of all 4 ER maps. It results in a total of 4096 bits to represent all 4 ER maps and 32 bits to represent Rx and Tx nodes' positions. It thus requires 512 bytes to be reserved in Field 7 of the OABF (Figure 15). Field 8 represents Tx node's queue state. Field 9 is used for queue-aware channel access, which will be discussed later.

Table 2. Omnidirectional antenna beacon frame (OABF)

(1) Type	(2) OA-net MAC Address	(3) OA-net TX Power level	(4) OA-net Rx MAC Address	(5) DA Link Pair's MAC	(6) DA Link Remaining Time	(7) Predicted ER	(8) Queue Length	(9) Random Access Dice Array	(10) Checksum
1	6	2	6	12	2	512	4	64	4

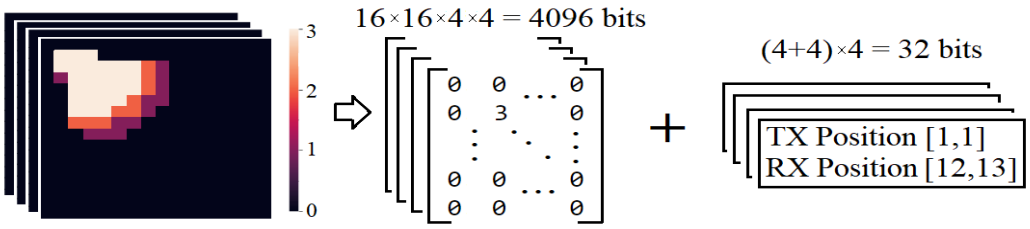


Figure 15. OABF Field 7 contents and ER information packaging.

To provide ER information for the one-hop neighborhood, each node extracts the ER information from the beacon slots and combines them to form the ER map with all neighboring links' ER states. Figure 16 shows the ER map fusion process from different nodes' ER reports. Each node can use such a fused map to predict the ER map for the next time window.

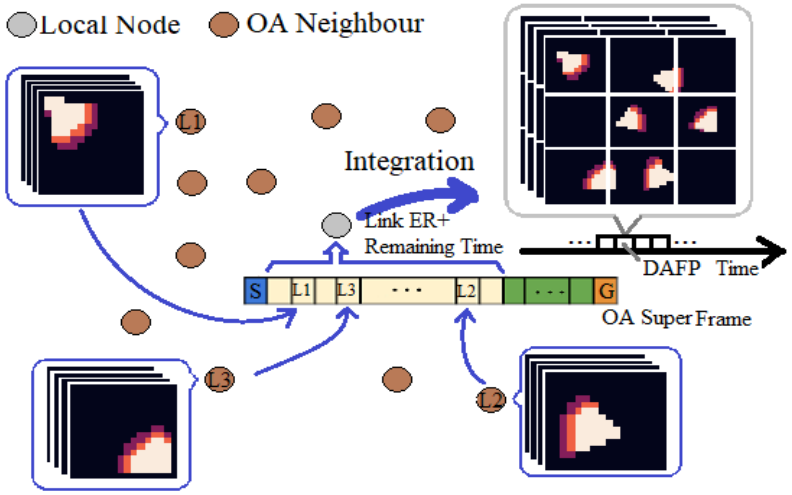


Figure 16. ER information fusion.

3.1.3.2 DA-based Data Transmission

Our ERL-MAC solution is a hybrid MAC scheme that uses the ER learning results for channel access control. It is hybrid since a node needs to follow certain schedule information (such as different contention slots in a superframe) and all neighbors need to use random channel contentions (by sending out OACF requests). Unlike TDMA-based schemes, our scheme does not need accurate time synchronization. The directional request-to-send (DRTS) and directional clear-to-send (DCTS) messages are used to negotiate the directional data transmission durations. The ER prediction result is used to find out the best time to send out DRTS and adjust the communication parameters. Hidden terminal and deafness problems can be easily solved since ER prediction and link status information can be shared among neighbors by using the OA to broadcast the out-of-band protocol messages. Our scheme can optimize the spatial and temporal reuse of the DA channels.

The DRTS and DCTS messages are used to negotiate the data transmission parameters. A virtual time slot, called DA frame period (DAFP), indicates the minimum time duration of data transmissions. ERL-MAC schedules the next several DAFPs at one time to indicate which link can use the RF channel for a specific DAFP.

Figure 17 shows the basic concept of DAFP, which indicates the minimum time required to finish a batch of directional data transmissions. The batch time includes the time for the following five operations: (1) Tx node sends DRTS; (2) All nodes within interference range set up the directional network allocation vector (DNAV); (3) Rx node sends back DCTS; (4) Data transmission time; (5) Rx sends DACK for feedback. The DAFP has a fixed time length and can be represented by eq. (21) below. Here, short interframe spacing (SIFS) is a guard time that ensures a wireless interface to process the received frame.

$$DAFP = (SIFS + DRTS) + (SIFS + DCTS) + Data + (SIFS + DACK) \quad (21)$$

Each DA superframe's period consists of multiple DAFPs. Each DAFP allows multiple non-interference links to transmit data concurrently. Given the information of link's remaining active time and ER spatial map, a node can calculate the total interference level from all neighboring nodes. The link with the highest QoS priority gets access to a DAFP. Table ***3*** and Table 4 show our proposed DRTS and DCTS formats, respectively. In DRTS, the serial number is increased by 1 whenever a new transmission batch starts. The expected start time and duration describe when and how long the data transmission occurs. The remaining fields describe the status of the DA of Tx node, including its operation frequency, direction, and Tx power level.

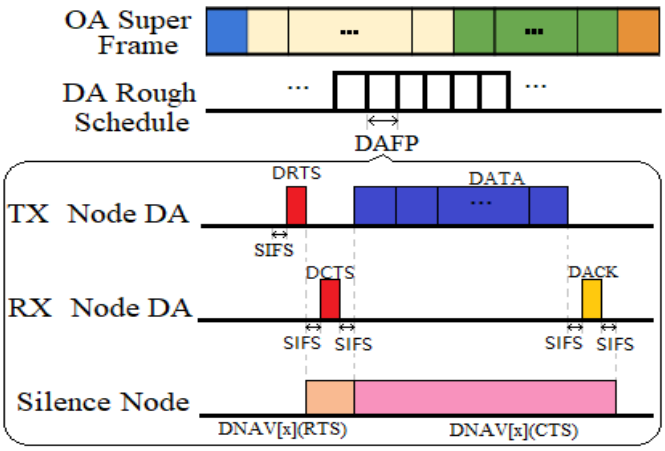


Figure 17. ERL-MAC design: Basic message exchange.

Table 3. DRTS message format

Frame Type	Serial Number	Directional Link TX MAC	Directional Link RX MAC	Date Length	Expected Transmission Start Time	Expected Data Transmission Duration	DA Channel ID	DA Tx Power Level	TX Pointed Direction	Check-sum
1	2	5	5	2	1	1	1	2	2	4

Table 4. DCTS message format

Frame Type	Serial Number	Directional Link TX MAC	Directional Link RX MAC	Permitted Time to Start	Permitted Transmission Time	Data Channel ID	RX Power Level	Tx's DA Direction	Checksum
1	2	6	6	1	1	1	2	2	4

Figure 18 shows the ERL-MAC protocol (main operations). Since the ER spatial map already has the link interference information, the DNAV message uses a simple structure with only two fields: (1) The 'keys' field has the affected link's Tx/Rx nodes' MAC addresses, and (2) The 'values' field has the link's remaining non-interference time. DNAV is updated in each batch.

Inputs:	Device link information; node queue state.
Outputs:	Link media access schedule for the next N period of DAFPs
1	For each node with active link, the next-period ER spatial map is predicted via LSTM-DNN
2	Claim OABF through OA contention slots for each involved active link.
3	Broadcast local buffer, links, and predicted ER information through OA beacon slots.
4	Read each neighbor's information on links, ERs, and buffer state through OA beacon slots.
5	According to queue state, calculate RADA for each link.
6	$n = 0$
7	while ($n < N$):
8	Estimate ER for each active link in the current DAFP
9	Run DA Sub Protocol 1 to decide which link gets the access to channel and also update RADA
10	Run DA Sub Protocol 2 to optimize DA data transmission duration length
11	Return Link media access schedule for the next N periods of DAFPs

Figure 18. ERL-MAC main protocol

3.1.3.3 Impact of Queue State

ERL-MAC also considers the node's queue state, which is stored in OABF. A node maintains separate queues for each of its links. Hence, the queue size Q_l for each link is considered. Our scheme gives more opportunities to the links with longer queues to access the channel. For each link, the TX node creates a random integer dice according to the queue status:

$$dice(Q_l) = randint\left[1, int\left(\frac{Q_l}{total\ buffer\ size} \times 100\right)\right] \quad (22)$$

The dice value for each link is renewed in each DAFP. The link with the highest dice value gets access to the channel. A field in OABF has a size of 64 bytes for dice values of the next several DAFPs. A random-access dice array (RADA) is also saved in this field. The i^{th} element in RADA can be represented by eq. (23) below, where $\underline{u}$ is the link rate,

$$RADA[i] = dice(Q_l - (i-1) \times u \times \text{DAFP}) \quad (23)$$

Each time a link successfully accesses the wireless channel, i is increased by 1. It, therefore, has a lower chance to access the media during the next DAFP. Figure 19 below shows the queue-aware channel access scheme.

Inputs:	RADA arrays indicating queue size for each outgoing link of the reference node; ER map for the considered RADA
Outputs:	Whether the link gets the access for this DAFP; produce an updated RADA array
1	Add all links into a pool P
2	Read RADA array for each link; Read ER for this DAFP
3	Create an array I_{RADA} to trace the index for each array's RADA
4	Create a pool Res that stores the info. of all links with the access to channel
5	while $(size(P) > 0)$:
6	For each link l in P, read $d_l = RADA[I_{RADA}[l]]$
7	The link $lmax$ with the highest d_l will have the access to the channel
8	Remove this link if it may cause contention from P based on ER analysis
9	Update I_{RADA} by decrementing $I_{RADA}[lmax]$
10	Return (Res and updated RADA arrays)

Figure 19. ERL-MAC sub-protocol 1: Queue-aware media access.

3.1.3.4 Protocol Optimization

As mentioned before, a single DAFP includes the periods of DRTS, DCTS, DATA, DACK, and SIFSs. Two consecutive DAFPs may correspond to different links that are granted access rights to the RF channel. When a link keeps the access to the channel in more than one DAFPs, we combine these DAFPs from the perspective of that link to eliminate excessive DRTS, DCTS, DACK, and SIFS.

Hence, DA data transmission is optimized, and the protocol overhead is reduced. As shown in Figure 20, we convert the problem into a unidirectional graph, where each DAFP slot is a vertex in a graph. The cost of the edge from one vertex to another is obtained by finding out how many links have switched between active and idle states when the node transitions from one DAFP to another (Figure 20, right). The goal is to find a minimum-cost path that walks through all vertices without a revisit. This converts our problem to the minimum cost Hamiltonian path problem [62] with the exception that there is no need to return to the start node. Such a problem can be solved by using the sorted edge algorithm [63].

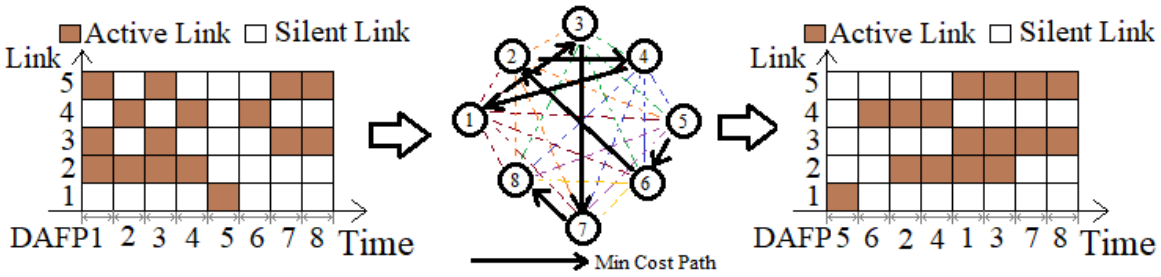


Figure 20. Sorted edges algorithm (SEA)-based DAFP sequence optimization.

3.1.3.5 Impact of High Mobility

For a network with high mobility, the node's sudden, big location changes may have a certain impact on the ER region. In this protocol, the ER distribution in each node can evolve with node mobility. Instead of directly predicting the aircraft's location changes, st-ResNet+ model is applied to predict the ER change in each receiver. The benefit of doing this is based on the following fact: While the exact aircraft positions may show much randomness, it does not have a significant impact on the node's interference level unless the ER's range exceeds a certain threshold. Therefore, the proposed ER prediction model can adapt to the high mobility of aircraft from the "interference impact" perspective.

3.2 Proactive Mobility and Congestion-Aware Protocol: Algorithms and Design

3.2.1 Related Work

Traditional proactive routing schemes compute the 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 [15][24]. To address this issue, in [24] a transmitter node excludes the fast-moving edge nodes from its 1-hop neighborhood by dynamically adjusting its effective communication area. In [64], 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 [19], 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 [21] 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 [29], 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 [22], 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 [21-22][29] select a route based only on the local network characteristics at the 1hop 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 [12][13] [22][65].

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 [15] 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 [66], 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 [26], 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 [67], 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 section, 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 the discussed challenges, and significantly improves the QoS, especially at high node speeds.

3.2.2 Overview of Standard OLSR

The OLSR protocol [68] 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 the 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.

3.2.3 Description of Proposed Cross-Layer MCA-OLSR Protocol

The modules of our proposed MCA-OLSR scheme are shown in Figure 21. 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.

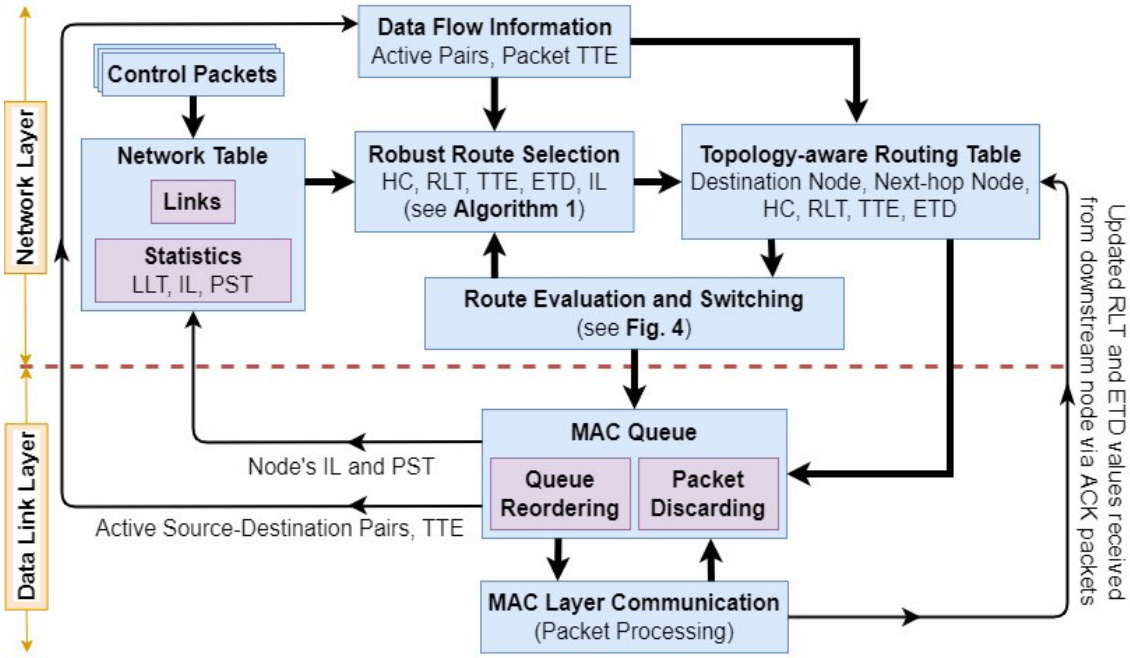


Figure 21. Modules in our proposed MCA-OLSR scheme.

Using the updated *LLT* values received via the control packets, a node removes the broken links from its *Network Table*. 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.

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. 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. 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.

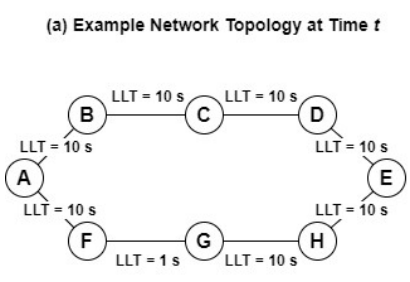
3.2.3.1 Network Modeling and Assumptions

The design and evaluation of routing protocols for ANs require mobility models that can produce realistic node movements by considering their aerodynamics [12-15][65]. FW-UAVs cannot make sharp turns due to their aerodynamics and high speeds. Therefore, we use a smooth-turn mobility model [84] in which each node independently selects a center and radius based on its past trajectory and rotates around the center in a clockwise or counterclockwise 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 [28][70]. 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.

3.2.3.2 New Enhancements

Routing a data packet in the OLSR protocol is shown in Figure 22. Each node in Figure 22(a) maintains a *Network Table*. Here, node A uses its *Network Table* (shown in Figure 22(b)) to compute a shortest hop route to each destination node, whenever it receives a new control packet. These routes are stored in its *Routing Table* shown in Figure 22(c). A node stores the data packets in its MAC queue in FIFO order, where each packet waits for its transmission.



(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 22. 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).

(A) Topology-aware Routing Table Construction

As discussed in Section 3.2.2, 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 [68]. 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 Figure 22(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 Figure 22(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 the 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 [16] [21-22][29][64][66-67][69-70]. However, significant interference from neighbor nodes in a dense network can cause inaccurate computation of RSSI values [26]. The *RLT* value can change because of the uncertain node movements in ANs [28], 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 change 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 3.2.3.2(D)).

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 [69],

$$RLT_R(i) = \min_{link\ l \in L_{Ri}} (LLT_l) \tag{24}$$

where L_Ri 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 [28][70].

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 [66],

$$ETD_R(i) = \sum_{j \in R^i} \left(\frac{1}{P_j} \sum_{\{p \in P_j\}} (PST)_p \right)_j \quad (25)$$

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* [29],

$$IL_R(i) = \sum_{j \in R^i} (IL_j^{\theta} + IL_j^{\phi}) \quad (26)$$

where R^i are the nodes on route *R* from node *i* to the destination. Here, IL^{θ}_j represents the recent *IL* value of node *j* received via the control packet, and IL^{ϕ}_j 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^{θ}_j value of node *j* in eq. (26). To identify the links in IL^{θ}_j, 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 3.2.3.2(A).

(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 [67][69-70].

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:

$$RLT_R(i) > TTE(i) + \delta \quad (27a)$$

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

$$\frac{TTE(i)}{ETD_R(i)} \geq \epsilon_1 \tag{27b}$$

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 *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 3.2.3.2(D) 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 (27a)).

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 (27a)) and (*ii*) the *TTE* of packets is ≥ 1.5×*ETD* (Constraint (27b)).

Step 2: Compute the cost of each route $R \in C^*(i)$ at node i as shown in eq. (28), and then select the lowest-cost route R^*:

$$Cost_R(i) = w_1\left(\frac{HC_R(i)}{HC_{min}}\right)^E + w_2\left(\frac{IL_R(i)}{IL_{min} + \alpha \times IL_R(i)}\right),$$

$$\text{where, E} = \begin{cases} n_i \times \log(n_i), if\ n_i \geq 2; \\ 1, \qquad\qquad otherwise. \end{cases} \tag{28}$$

Here, $(.)_{min} = \min_{\forall R \in C^*(i)} (.)_R$, and n_i is the shortest hop distance between the source node and node i. 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 pseudocode to select a long-lasting and less-congested route is given in Figure 23.

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

Figure 23. Pseudo Code for Route Selection at Node *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:

$$RLT_{R^*}(i) < TTE(i) + \delta \tag{29a}$$

$$\frac{TTE(i)}{ETD_{R^*}(i)} < \epsilon_2 \tag{29b}$$

$$HC_{min} > HC_R(i) \tag{29c}$$

The conditions (29a) and (29b) correspond to violations of the selection constraints (27a)– (27b). 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 (29c) 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 Figure 24.

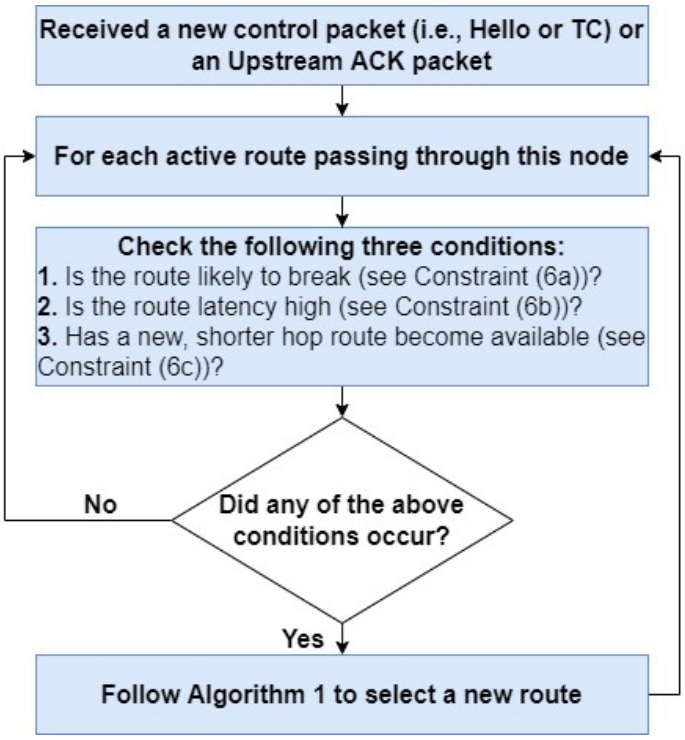


Figure 24. Flowchart of the route switching module in our proposed scheme.

(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) [85] and TSDP (two-step discarding policy) [86], 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.

3.2.3.3 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.

3.2.3.4 Route Computational Complexity

In the OLSR protocol, every node computes the 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 3.2.3.2), 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.

3.3 AI-based Directional Routing Protocol: Algorithms and Design

3.3.1 Related Work

The learning/prediction of network conditions can help to improve the routing performance. Recently deep learning models have been used for wireless routing enhancement. For example, the convolutional neural networks (CNNs) were used in [71] to select the best path. The CNN model accepts the input parameters (including the queue states, packet generation rate in each node, and link status), and outputs the network performance matrix that includes the QoS metrics. Other artificial intelligence (AI) models, such as deep reinforcement learning (DRL) models, have also been used in the routing layer [72-73]. The reward function and actions must be determined for these AI models. However, we are not aware of any routing design with deep-learning-based global flow prediction in both spatial and temporal domains.

Directional routing has also been investigated in some schemes. For example, a selection-region-based ad-hoc routing protocol with the use of DAs was discussed in [74]. In each hop, the antenna's parameters are determined for optimal data forwarding. In [75], an improved AODV protocol based on DAs was proposed to decrease the overhead due to query message flooding in the entire network. An extension of dynamic source routing (DSR) was discussed in [47]. DSR's message formats, route discovery process, and route maintenance process were all modified to adapt to DA's features. However, all those studies focus on how to utilize DAs to decrease the routing overhead and save the bandwidth. They have not considered the integration of network traffic distribution with antenna features for optimal routing design.

In this scheme, we integrate the features of directional links with st-DL+ model based DHM prediction for optimal routing. Unlike the above single-flow routing designs, we consider the impact of multi-flow traffic when establishing the path among light traffic areas.

3.3.2 St-DI^{+}-based DHM Prediction

Our spatio-temporal congestion-aware routing protocol is inspired by the city traffic modeling. However, there are significant differences between MANET data flows and city vehicle flows: While the vehicles can only run on certain roads, the MANET nodes can experience more serious RF signal fading. Moreover, to reflect the congestion distribution, the queue status in each node must be represented in the DHM. To incorporate the high-dimensional information (signal strength in all positions, queue status in all nodes, etc.) into the DL's inputs, we extend the ResNet-based DL model in [30] to a st-DL+ model with multiple components as discussed below.

3.3.2.1 DHM Model

The DHM model is a synthesis of five maps, i.e., link map, queuing map, traffic sender map, traffic receiver map, and directional-antenna-aware wireless channel map (DAWC). The link map indicates wireless links among all nodes as well as the traffic rate in each link. The queuing map shows the node's buffer status. The sender map reflects the sending rates in all active source nodes, and the receiver map shows the receiving data rates in all destination nodes.

We use an example of network topology in Figure 25 to explain the generation of the above five maps. The horizontal and vertical axes in this figure represent the longitude and latitude of the map (in the unit of meters). Figure 25 shows three active flows, i.e., from node 2 to 5, 40 to 36, and 6 to 8. The nodes can move around in a space of 5,000 × 5,000 meters. The average RF link distance is 500 m.

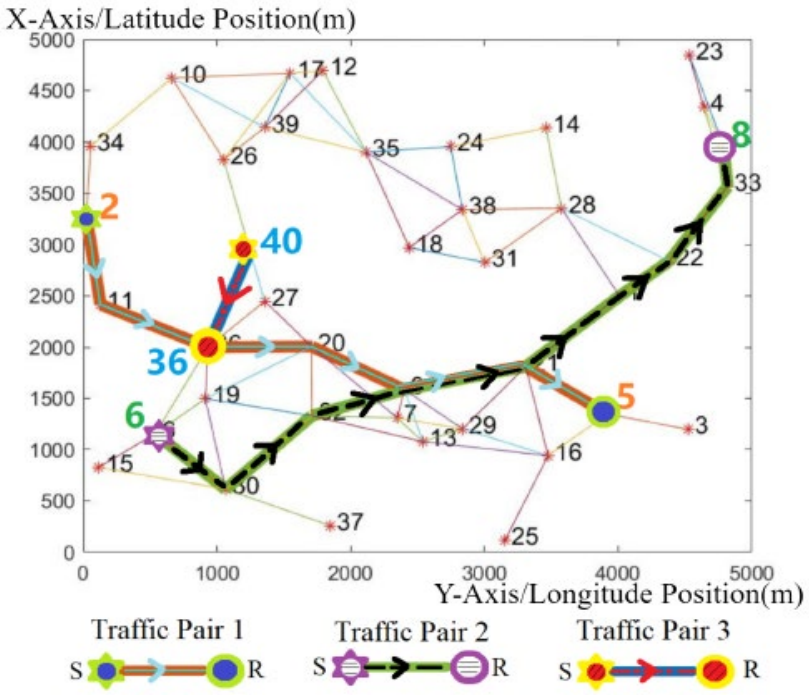


Figure 25. An example of MANET topology.

(i) Link map: A link exists between two nodes when they can reach each other with a satisfactory SNR level. The signal-to-interference-and-noise ratio (SINR) level is not used since link interference information has been represented as a spatial map (i.e., directional-antenna-aware wireless channel map) in our DHM model (see Figure 25). Link map highlights the maximum capacity that can be reached for each link. We use the red color (Figure 26) to represent the higher traffic rate in a link. The X and Y axes are normalized longitude and latitude coordinates with the range of 1 to 64. Since the wireless signals can radiate around, the traffic density at any point depends on the sum of all the RF signals from the nearby links. Therefore, we use the following formula to determine the density at a position (x, y) in the link map:

$$Link_{Map(x,y)} = \begin{cases} \min\left(C, \sum_k \varepsilon_k \times C\right), & if\ \exists d_{lk}(x,y) < \gamma \\ -C, & otherwise \end{cases} \quad (30)$$

where $d_{lk}(x,y)$ is the distance from point (x, y) to link k, and C is the link capacity (i.e., data transmission rate). A nearby link k adds ε_k to the value at the position (x, y) if its distance is less than a threshold γ. Here ε_k is the percentage of link capacity utilized for link k. Hence, $\varepsilon_k \times C$ is the actual flow rate for this link. We define $\gamma = (width\ of\ each\ pixel\ box)/\sqrt{2}$. Figure 26(a) illustrates this point. Nodes 1 and 2 in the figure have a link between them (i.e., link 0). To calculate the pixel value at location (x,y), the distance from the position (x, y) to link 0 is recorded as dl0(x, y). In this example dl0(x, y) > γ, the pixel value at (x, y) is −C (which indicates a weak RF signal).

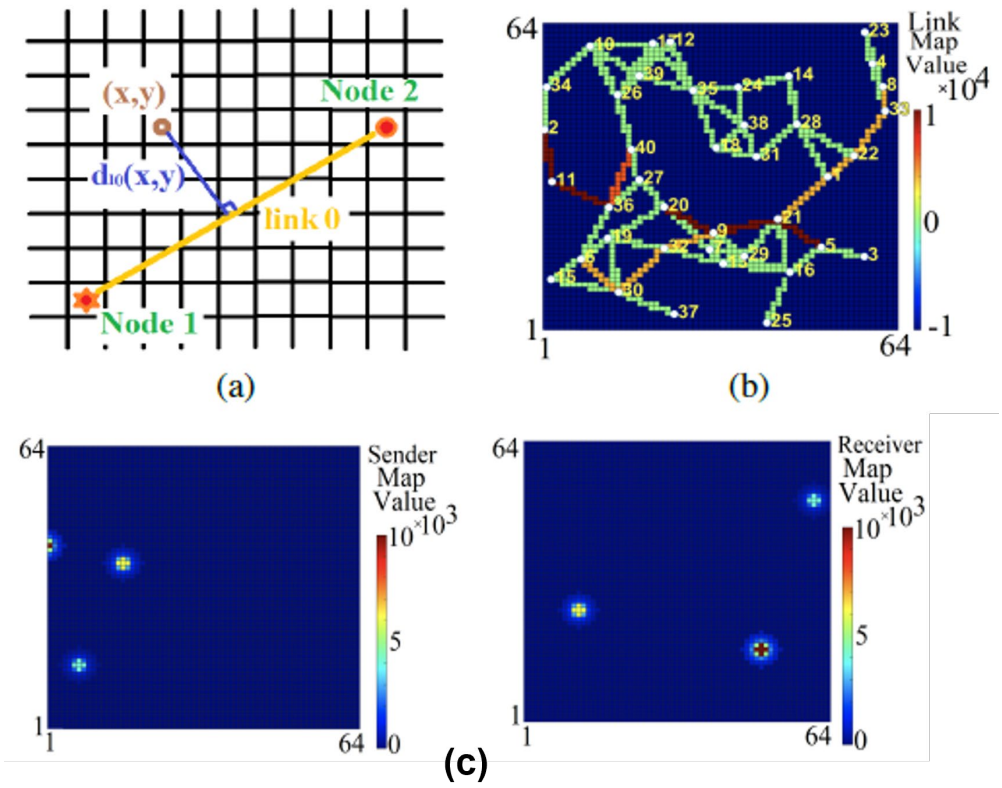


Figure 26. (a) Link map pixel values (example); (b) Generated link map; (c) sender/receiver map.

Figure 26(b) shows an example of the link map. Each link has a different traffic density level. Here, the red color represents a higher density. Note that due to the use of DAs, all signals are focused well in each link. We will discuss the effect of antennas later.

(ii) Sender/receiver signal map: The sender and receiver maps represent the traffic rates from the senders and receivers, respectively. At a position (x, y), the pixel value is determined as follows:

$$SenderMap(x,y) = \sum_m \frac{S_m}{(d_m(x,y))^2}, ReceiverMap(x,y) = \sum_n \frac{R_n}{(d_r(x,y))^2} \quad (31)$$

where m is the sender's ID and r is the receiver's ID. S_m is the sending rate for node m and $dm(x,y)$ denotes the distance from a map location (x,y) to the sender m. The eq. (31) shows the sum of signal strengths from all sender nodes. In eq. (31), n is the alias for the receiver node; R_n is the receiving rate for node n; $dr(x,y)$ is the distance from location (x, y) to receiver n. Both equations show that the signal radiation strength decreases proportionally with respect to the square of the distance. Figure 26(a) and Figure 26(b) show the examples of sender and receiver maps, respectively. The sender and receiver maps have certain pixel pattern similarity, which means that those regions may have multiple traffic hotspots at the same time due to the correlations between communication traffic and regional events. For example, when an event occurs at a

location, multiple nearby nodes may send messages back to their mission control center at the same time.

(iii) Queuing map: The queuing map reflects the packet waiting status in the queues of different nodes. The pixel value at position (x, y) is determined by the eq. (32), where OB stands for occupied buffer and B stands for buffer:

$$QueuingMap(x,y) = \begin{cases} -1, if\ no\ node\ at\ location\ (x,y) \\ \frac{\sum OB(x,y)}{\sum B(x,y)}, \qquad otherwise \end{cases} \tag{32}$$

In this map, a higher percentage of occupied queues in any region implies a longer packet delivery delay for the flows passing through that region. A full queue indicates a heavily congested network position. For the network topology shown in Figure 25, an example queuing map is shown in Figure 27, where the red color represents a full queue.

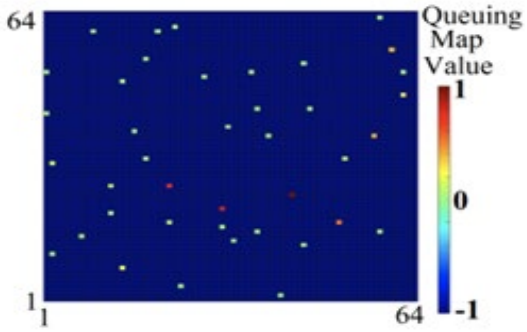


Figure 27. Queuing map.

(iv) Directional antenna-aware wireless channel map (DAWC): DAWC shows the DA's attributes (such as beamwidth, orientation, etc.) as well as the wireless channel occupancy status. It reflects the congested level of wireless channels at a specific location and time.

Figure 28 shows an example of DAWC with two path flows, where the red (or dark blue) color indicates a high (or low) usage of wireless channel.

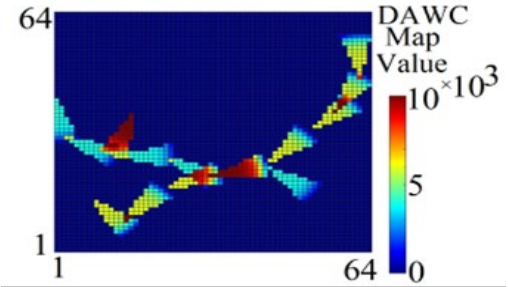


Figure 28. DAWC (example).

3.3.2.2 Use of Enhanced DL for DHM Prediction

To solve the problem of DHM prediction, we propose to use an extended DL scheme (st-DL+) with two features as follows: (1) Spatio-temporal traffic modeling: It models the spatial and temporal correlations in the entire network by using ResNet-based DNN (deep neural networks) with convolutional encoder/decoder; (2) Prediction of next-time DHM: st-DL+ can extract the evolutional patterns from a series of DHM snapshots, and then predicts the next-time snapshot. Such prediction results make the route selection more proactive since it can better adapt to the traffic distribution dynamics.

Figure 29 shows the architecture of the st-DL+ model. It takes multiple historical snapshots (at time instants $(t - T, t - T + 1, \ldots, t - 1, t)$ as inputs. In this project, we take long-, mid-, and short-term sampling with an interval of 60s, 30s, and 5s, respectively. Each term has three sampling data points (forming a network snapshot). The output is the predicted DHM snapshot at time $(t + 1)$. The model can thus capture the long-term, mid-term, and short-term patterns.

Figure 30 shows an example of signal patterns in terms of the trend, period, and closeness. The data shows a linear decrease, mid-term oscillations within a period of around 20 seconds, and short-term fluctuation in 1-second level. The st-DL+ model can therefore capture the features of high-dimensional data.

The st-DL+ model (Figure 29) extends the conventional ResNet-based DNN model [30] by adding the convolutional encoder and decoder network (CEDN) to extract the characteristic feature maps/matrix from the historical snapshots and generate directional traffic heatmap. In our implementation, each snapshot is a heatmap with the size [64, 64, 5], which means that each DHM snapshot contains five component maps (i.e., link map, sender map, receiver map, queuing map, and DAWC), and each component map has a size of [64, 64]. Three identical parallel ResNets are stacked together in the input layer, where each ResNet deals with 3 snapshots (reflecting the characteristics of trend, period, and closeness). With the sampling intervals discussed before, the first ResNet takes the network snapshot at the time of t− 48, t− 38, and t− 28. The second ResNet takes samples at the time of t− 18, t− 13, and t− 8. And the last ResNet takes samples at the time of t− 3, t− 2, and t− 1.

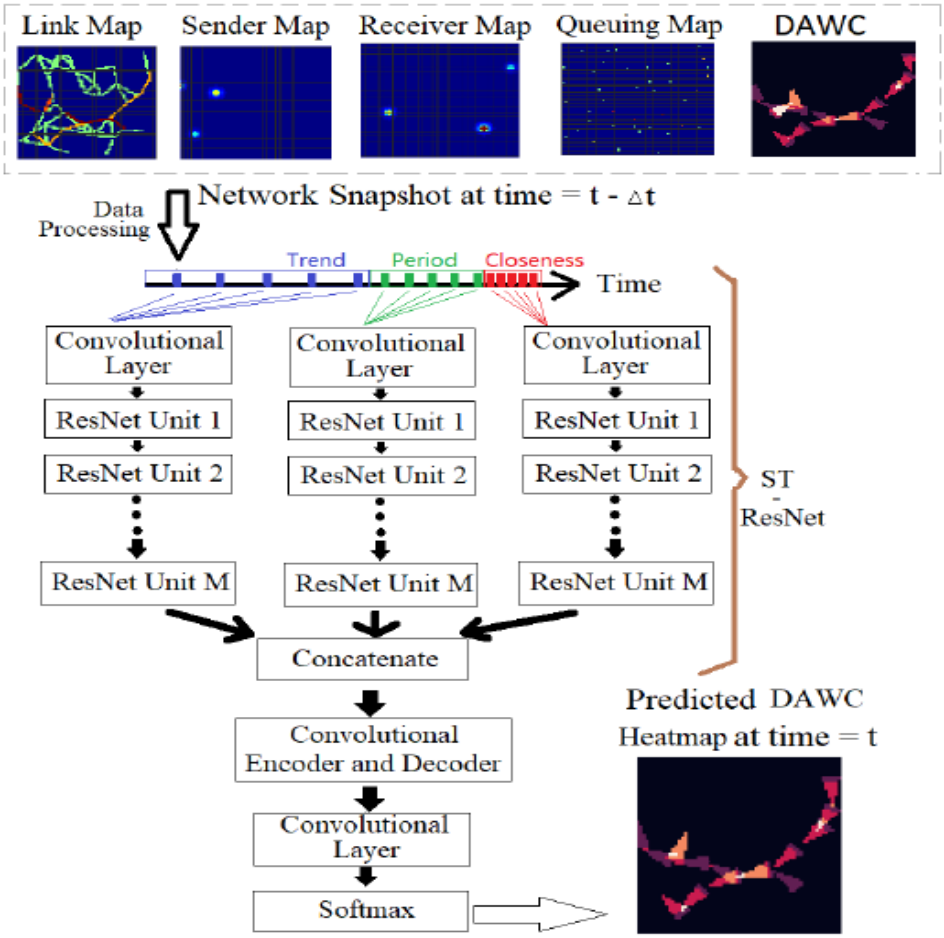


Figure 29. Enhanced DL model (st-DL+) for directional heatmap prediction.

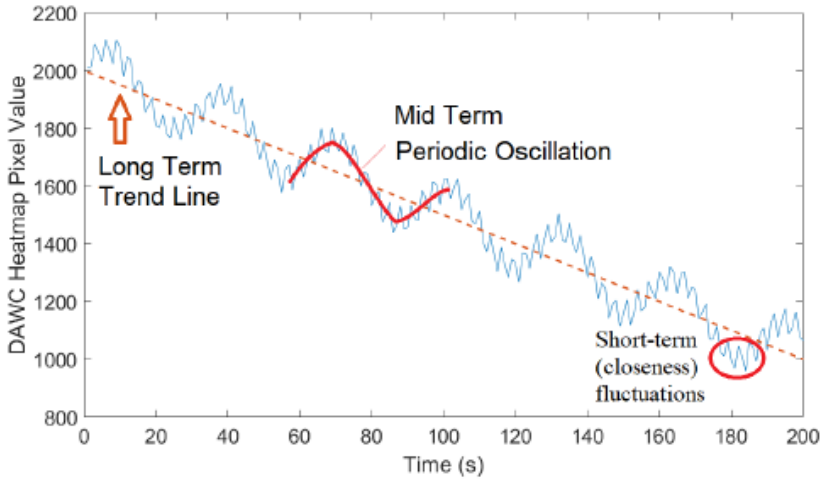


Figure 30. The concept of trend, period, and closeness.

3.3.2.3 DHM Information Collection Protocol

To implement the st-DL+ model, the system performs network information collection for DHM calculation. Each node needs to share and flood local information to its neighbors. Sub-protocol 1 (see Figure 31) is used for the sharing of node/link information throughout the network.

1.	**Input:** Local node's positions, buffer queuing levels, received signal strength (RSS), traffic sending rate in each source, and the source and destination IDs
2.	**Output:** Node/link information is shared among the network nodes; Snapshots (DHMs)
3.	Each node captures current local information. The information includes the position, queuing status, traffic rate, traffic source & destination, and link signal strength (including one-hop neighbors' IDs and the RSS in each receiver)
4.	Use RSS values to estimate the location if position information is not available from GPS.
5.	Renew and save table information in local memory.
6.	Encapsulate the above information into node and link information flooding packet (NLIP), and flood NLIP in the MANET.
7.	Each node maintains local DHM of MANET. Build Link Map, Receiver Map, Sender Map, Queuing Map, and DAWC based on information saved in local buffer. Save network DHM and delete outdated DHM.

Figure 31. Sub-Protocol 1. Node/Link Information Sharing and Flooding Protocol.

Node and link Information flooding packet (NLIP) is defined to contain node and link information. Table 5 shows the structure of NLIP, which can be packaged into a UDP packet. A check header field indicates that this is a routing message. The Count field is used to avoid duplicate message delivery. It also helps to update the local map information saved in the buffer. The remaining fields contain two tables which carry the node and link information for the neighborhood.

Table 5. NLIP format.

UDP Packet Header	Check Header	Local Nodes' IP	Count	Number of Nodes
Node Information Table				
Link Information Table				

The detailed carry-on information is shown in Table 6 and Table 7. Table 6 contains each node's IP, buffer status, location information, and clustering information. A device with low computation power can apply the clustering algorithms (such as the one in [76]) to find local cluster-head within 1-hop distance for assistance. The cluster-head then sends its computed DHM information to the nodes that ask for assistance through direct link communications.

Table 6. NLIP carry-on node information table.

Node 1's IP	Buffer State	Channel Occupancy State	GPS Info.	Cluster state
.....	.....	.....	.....	.....
Node n's IP	Buffer State	Channel Occupancy State	GPS Info.	Cluster State

Table 7. NLIP carry-on link information table.

Link 1 Nodes' IPs	Link 1 RSS Info.	Source IP	Destination IP	Flow rate
.....	.....	.....	.....	.....
Link n's Nodes IPs	Link n's RSS Info.	Source IP	Destination IP	Flow rate

The link information packet (shown in Table 7) contains each link's start and end nodes, received signal strength (RSS), and flow rate.

When the network snapshots get updated, the nodes with strong computation power can execute the st-DL+ algorithm in a distributive way. Other devices can ask local CHs for assistance (assuming a cluster-based network architecture is used). Sub-protocol 2 (see Figure 32) describes such a protocol for st-DL+ training and parameter distribution. This protocol defines how predicted network snapshots should be generated.

We define online training model update packets (MUP) as shown in Table 8. It is used to help nodes without enough computation power to get updated model parameters. The field 'Serial Number' increases with respect to model updating time. It makes sure that the node seeking assistance receives the most recent model parameters. The 'Sleep Time' field indicates the time for the CH to perform the next model update. It tells when it should ask MUP for the next time. The last field (2nd row of the table) contains the renewed model parameters. The purpose of online training is to update the model parameters and increase accuracy. In our routing scheme, any node can utilize its pretrained model when there is no trainable data at the starting phase.

1.	**Input:** Information gathered through sub-protocol 1; Snapshots (DHMs) of the network.
2.	**Output:** Renew AI model in each node.
3.	**if** local node has sufficient computation power**, then**
4.	Generate model input and output training data. Specifically, sampling on memory data, extract trend, period, and closeness network snapshots. Missing data can be estimated based on antenna model and links traffic information.
5.	Online training ST-DL+ model, renew model parameters.
6.	Pack renewed ST-DL+ model parameters, save parameters into local buffer. The buffer information can be further used in MUP.
7.	Replay node requires assistance with MUP if they ask.
8.	**else**
9.	Open socket, ask local cluster-head for MUP.
10.	Receive and read MUP. Update local model parameters by analyzing MUP carry-on information.
11.	**end if**

Figure 32. Sub-Protocol 2. DHM online AI model training and renewal protocol.

Table 8. Online training model update packets (MUP).

UDP Packets Header	Check Header	Header Nodes' IP	Serial Number	Sleep Time
Enhanced ST-ResNet Model Parameters				

3.3.3 Optimization of Directional Routing Protocol

After the DHM snapshot is predicted, the next step is to select the best path for a given source and destination pair. We propose an optimization algorithm to select a short (in terms of hop length) path, while avoiding the congested regions as well as the interference from the neighboring nodes' DAs.

First, let's see how the conventional routing scheme works. Figure 33(a) shows the path selected by using a shortest path scheme (e.g., AODV). As we can see, the shortest path is selected while ignoring the traffic distribution in the network. Here, the nodes located in heavily congested areas can cause large end-to-end delays.

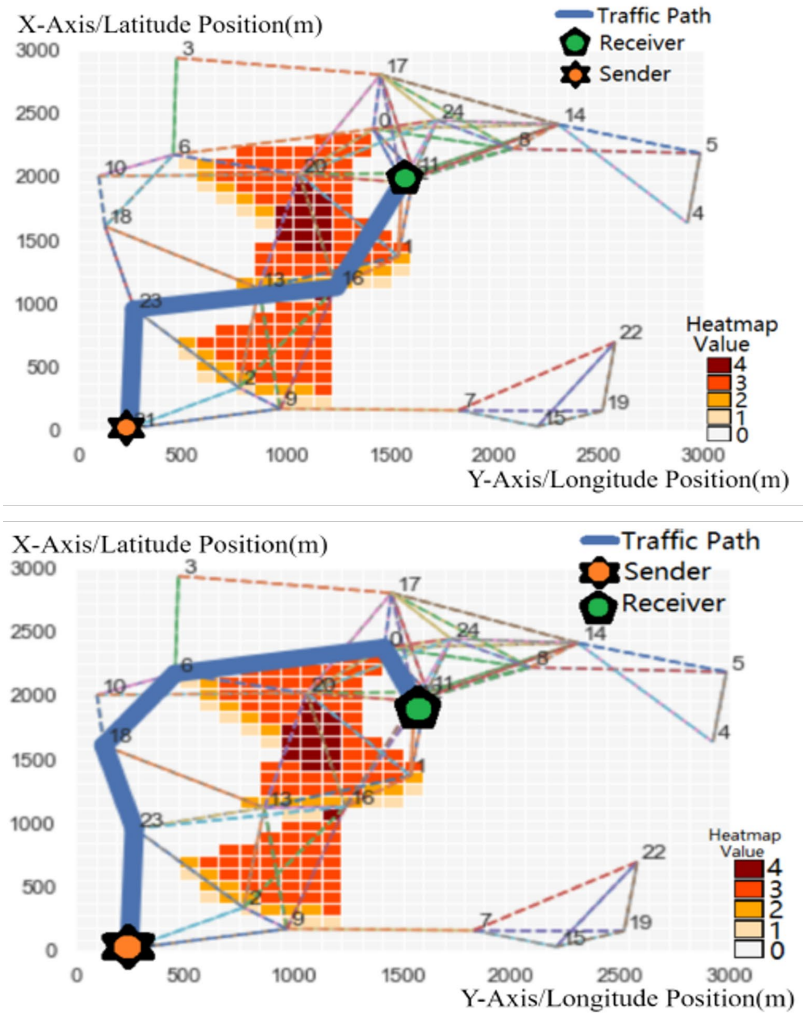


Figure 33. (a) Top: Conventional shortest-path routing, (b) Bottom: Generated path from our DHM scheme.

Figure 33(b) shows the single path selected by our DHM optimization scheme. A single path can satisfy QoS requirements in this case. However, more extreme cases with multiple paths and heavy congestion need to be considered. Our proposed scheme uses the global traffic map to generate paths, where multiple paths are allowed to share common links. Particularly, if a path happens to pass through the edge of a heavy-traffic area, the routing algorithm allows it to split into multiple branches and converge into one path after passing the congested areas.

Assume two end-to-end paths (path 1 from node 11 to 15, 2 from node 6 to 14) in Figure 34. Here path 2 passes along the edge of a heavy-traffic area. To maintain equivalent flow throughput in the entire end-to-end path (such that there is no bottleneck anywhere), our route optimization model splits flow on path 2 into two branches: one goes along the edge of the congested area (i.e., passing through nodes 7-2-10), and the other overlaps with path 1 (i.e., between nodes 7 and 5). Later, these two branches are aggregated into one path from node 10 to 14. By using split branches, we can guarantee that flow 2 has high throughput. This enables a routing scheme to avoid the region with a traffic bottleneck. Here the second branch of path 2 overlaps with path 1 in the link

7-5. This is acceptable since the second branch is not close to the congested area and thus has higher throughput to accommodate both paths' data flows.

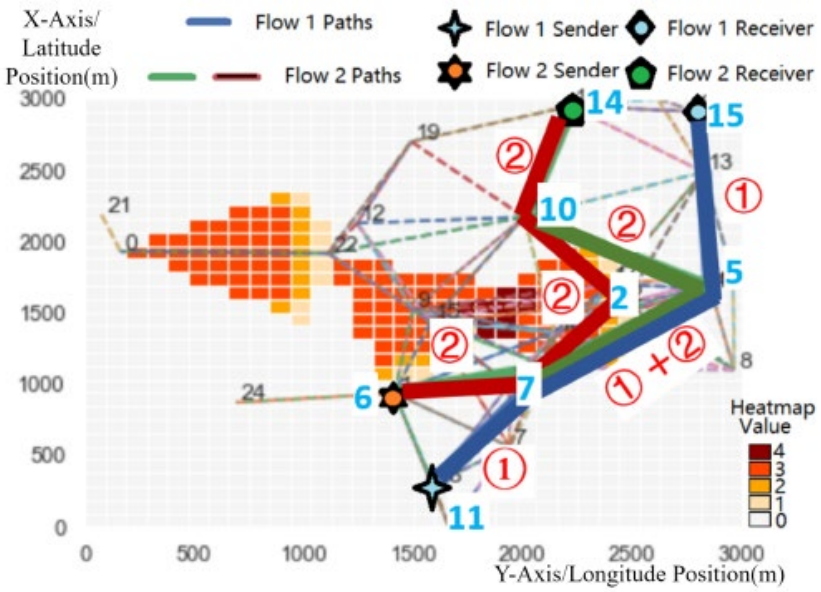


Figure 34. Our scheme: Path-split case.

3.3.3.1 Forming Directed Graph

We first represent a MANET with DAs as a directed graph as below:

$$G = \left(\mathcal{N}, \varepsilon = \{\varepsilon_{ij} | \varepsilon_{ij} \in \epsilon; i, j \in \mathcal{N}\}\right), \qquad U = [u_{ij}];$$

$$where\ u_{ij} = 1 + \max\left(D(x_i, y_i), D(x_j, y_j)\right)\ with\ i, j \in\ \mathcal{N} \tag{33}$$

Here, $\mathcal{N}$ is the set containing all nodes and ε represents the set of links. ε_{ij} is the link from a node i to j. Figure 35 shows the example of a directed graph. We define the link cost u_{ij} as the average heatmap pixel values in that link. The pixel value reflects the aggregated link traffic in that position. In the above equation, the cost has a constant 1 in it, which means that the cost increases each time the path crosses a new hop. This helps us to find a short path (i.e., with less hops). D is the matrix which stores the predicted heatmap values. $D(x_i, y_i)$ is the predicted DHM value at location (x_i, y_i).

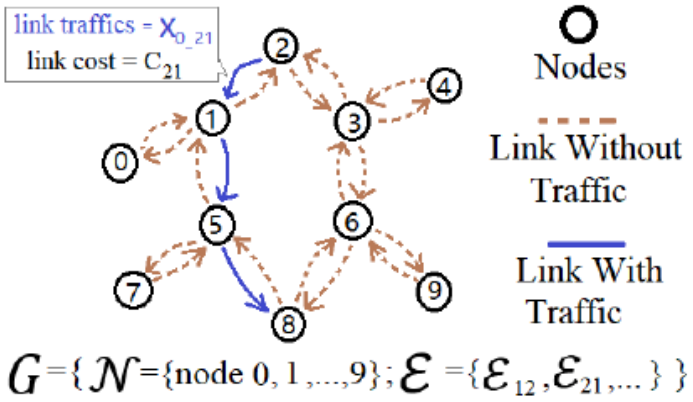


Figure 35. Example of directed graph.

3.3.3.2 Route Optimization Model

The objective function of the route seeking process can be expressed as follows:

$$min\left(\sum_i \sum_{i,j\in\mathcal{N}} u_{ij} \times x_{l\_ij}\right) \tag{34}$$

where l is path flow ID; i and j are the nodes connected by link ε_{ij} . x_{l_ij} represents the data rate on link ε_{ij} for the flow l. u_{ij} can be calculated based on DHM distribution. In this function, x_{l_ij} is the target matrix to be solved. We aim to minimize the sum of products for the link costs. Through the definition of u_{ij}, we guarantee that more hops and/or larger link cost will produce a larger value. The optimization result will be a path that avoids congestion areas and has a low hop number at the same time.

Now we consider the constraints for the above optimization model. We first define the traffic rate balancing condition (i.e., the total inflow rates should be equal to the total outflow rate):

$$A.X_l^T = B_l \tag{35}$$

In the above equation:

$$A = [a_{ij}] with\ a_{ij} = \begin{cases} 1, & if\ u_{ij} \neq \infty \\ 0, & otherwise \end{cases}; i,j \in \mathcal{N} \tag{36}$$

Furthermore,

$$B_l = [b_{l\_i}] with\ i \in \mathcal{N}; l\ is\ alias\ for\ flow\ name;$$

$$b_{l\_i} = \begin{cases} flow\ l\ data\ rate, & if\ i\ is\ sender \\ -1 \times flow\ l\ data\ rate, & if\ i\ is\ receiver \\ 0, & otherwise \end{cases} \tag{37}$$

In the above node matrix A, a_{ij} is 1 if the link from node i to j exists, otherwise it is 0. In the traffic summary matrix B_l, a positive b_{l_i} value indicates that flow l causes a net outflow at the sender; in contrast, a negative b_{l_i} value indicates a net inflow at the receiver.

For DAs, we use the following constraint to ensure that the sum of inflow and outflow data rate in a node is not greater than the channel capacity σ. For all flows (with different IDs), we use P to represent the set of σ in the format of a vector. We have already incorporated the impact of DAs in the directional traffic map. Here we guarantee that the selected relay node avoids queue overflow through the constraint in eq. (38). The first term $A \cdot X_l^T$ is the outflow amount from each node; the second term $A^T \cdot X_l$ is the inflow amount to each node. The sum of these two terms must be smaller than the total channel capacity P.

$$\sum_l (A.X_l^T + A^T.X_l) \leq P \tag{38}$$

where $P = [\dots, \sigma, \sigma, \sigma, \dots]^T$, l is the flow ID.

Solving the above optimization model: Algorithm presented in Figure 36 shows how the above defined optimization model can be solved based on the revised simplex method. Note that our algorithm also considers an extremity case- the desired traffic rate is too high even when we utilize all the possible paths in the network. In this case, there will be no solution. The entire algorithm finds a path with the minimum number of hops (among all possible candidate paths), while avoiding heavy-traffic regions. It also considers the impact of DAs and ensures that a new path is not within the directional signal range of any existing path node. Lines 1-3 formulate the path-seeking issue into an optimization problem. They are used for routing operations. In line 3, an optimization problem with objective function and restriction is given. The process after line 3 attempts to solve the optimization function. A revised simplex method is applied between lines 7 and 20 to solve the optimization function. It is an iteration process with each iteration storing the updated solution in X. The convergence result of the optimization function allows a path to be split into multiple branches if it is needed.

Input: Nodes number, link cost, traffic sender & receiver, expected traffic rate, DAWC heatmap
Output: Optimized path with congestion avoidance; Scheduled traffic rates in different paths if multiple paths need to be found.

1: Find link costs based on DAWC heatmap. (eq. 12))
2: Build initial optimization problem with: (i) objective function (eq 13); (ii) equality constraint (eq 14); (iii) inequality constraint (eq 17).
3: Re-write the optimization problem in the format of $argmin(\mathcal{U}\mathcal{X})$ with $\mathcal{A}\mathcal{X} = \mathcal{D}$ by doing: (1)Add slack variables for inequality constraints, convert inequality constraints to equality constraints. (2) Do row reductions for the constraints, and make sure the constrain matrix has a full rank.
4: **if** $argmin(\mathcal{U}\mathcal{X})$ with $\mathcal{A}\mathcal{X} = \mathcal{D}$ is infeasible **then**
5: Go back to Step 2 (without constraint (iii))
6: **end if**
7: Find initial base $\mathcal{B}$ and its complimentary set $\widehat{\mathcal{B}}$. Record feasible solution $\mathcal{X}_\mathcal{B}$, cost $\mathcal{U}_\mathcal{B}$, and $\mathcal{A}_\mathcal{B}$ corresponding to the base $\mathcal{B}$.
8: **while** True **do**
9: Find $\mathcal{Y}$ such that $\mathcal{A}_\mathcal{B}\mathcal{Y} = \mathcal{U}_\mathcal{B}$
10: Find term $\mathcal{U}_k > 0$ in $\bar{\mathcal{U}} = \mathcal{U}_{\hat{\mathcal{B}}} - \mathcal{A}_{\hat{\mathcal{B}}}^T\mathcal{Y}$, choose $\mathcal{X}_k$ as 'enter' variable, where k is the 'enter' base.
11: **if** $\bar{\mathcal{U}} \leq 0$ **then**
12: Solution found. Break.
13: **end if**
14: Find d such that $\mathcal{A}_\mathcal{B} d = \mathcal{A}_k$.
15: **if** $d \leq 0$ **then**
16: Unbounded. Go to line 5.
17: **end if**
18: Find minimum nonnegative term in $\{\frac{\mathcal{X}_\mathcal{B}}{d}\}$, record corresponding base v.
19: Renew bases $\mathcal{B}$, v should leave, k enter. Renew $\mathcal{U}_\mathcal{B}$, and $\mathcal{A}_\mathcal{B}$. Renew $\mathcal{X}_\mathcal{B}$ and $\mathcal{X}$:

$$\mathcal{X}_p = \begin{cases} 0 & \text{, if } p \notin \mathcal{B} \\ t & \text{, if } p = k \\ \mathcal{X}_p - t * d_p & \text{, otherwise} \end{cases}$$

20: **end while**
21: Convert $\mathcal{X}$ back to values of set $\{x_{l\_ij}\}$. For each traffic flow, find network traffic paths and scheduled data rate in each path for multi-path case.

Figure 36. Optimization-based path selection.

3.4 Local Route Repair Protocol for a Network of Multibeam FDD Nodes: Algorithm and Design

3.4.1 Literature Review

In this section, we review the multipath reactive routing protocols and the corresponding local route repair schemes. The ad hoc on-demand distance vector (AODV) [77] is a well-known reactive routing protocol that finds a single loop-free, shortest hop-count path between the source and destination nodes. To better utilize the available network resources, the AOMDV routing protocol [78] finds multiple 'node-disjoint' or 'link-disjoint' paths for a source and destination pair. Here, the node-disjoint routes do not have any nodes or links in common, whereas the link-disjoint routes do not have common links but may have some common nodes. The node-disjoint approach guarantees completely independent routes so that the failure of a route does not impact the other routes (Figure 6a). On the other hand, the link-disjoint routes have some common intermediate node(s) and the failure of such a node can break more than one route (Figure 6b). Therefore, the node-disjoint approach might discover fewer routes than the link-disjoint routing.

Recently, the BAOMDV routing protocol was proposed in [37], which is an extension of the AOMDV and OAOMDV protocols, and can discover multiple, completely overlapped forward and reverse routes between a source and destination pair in a network of directional FD or FDD nodes. Use of directional antennas in wireless networks has also led to the development of a few other routing protocols, such as [79][46][49-50]. These protocols make use of not only the node id information, but also the information about the beams the nodes lie in. A few routing schemes have also been reported in the literature for full-duplex communication [37][80-81]. However, these protocols (except BAOMDV [37]) do not establish completely overlapped, multiple bidirectional routes.

Several local route repair schemes (based on reactive routing protocols) exist to overcome the route breaks in ad hoc networks. However, only a few of these consider the use of directional antennas [46-47]. The directional routing protocol (DRP) [46], uses a dynamic source routing (DSR) based protocol for route discovery. For route maintenance, it uses location tracking and sends requests in the adjacent beams to find the node that has moved out of its current beam. The DRP scheme also introduced a two-hop directional local repair mechanism, where it locally sends directional RREQ and RREP packets to repair the broken link. In [47], the capability of directional antennas to transmit over a longer distance is used to perform local route repair.

With the emergence of full-duplex communication, bidirectional traffic applications are getting closer to practice. The AODV bidirectional fast adaptive backup with local repair (AODV-BFABL) scheme [48] maintains and repairs the routes by overhearing (using omni-directional antenna) the bidirectional data traffic between the source and destination. But the use of omni-directional antenna introduces the range asymmetry issue and may not be effective when using nodes with directional antennas. To the best of our knowledge, no route repair scheme exists for the overlapped routes for full-duplex or FDD nodes equipped with multi-beam antennas. Our proposed route repair scheme addresses these issues and is also suitable for the single as well as multipath reactive routing schemes, such as AODV [77], AOMDV [78] and BAOMDV [37].

3.4.2 Bidirectional Route Repair Challenges

In this section, we describe the need for bidirectional route repair, the type of route breaks, and route repair challenges for the links formed by multi-beam FDD nodes that allow the concurrent bidirectional packet transmission and reception among different nodes. The BAOMDV protocol [37] was designed to establish multiple, completely overlapped, bidirectional routes between a source and destination pair of fullduplex, multi-beam directional nodes so that the concurrent data transmission can be supported on multiple routes. Here, the interference among neighboring nodes that participate in different routes is eliminated by using multi-beam directional antennas. As a result, the BAOMDV protocol can support much higher data rates in both directions by using multiple active routes. A route break can therefore cause a significant rise in packet drops and end-to-end delay in BAOMDV. For example, if both S-A-E-H-D and S-B-F-I-D are active routes in Figure 6a, it is *highly beneficial if the protocol can locally repair any broken links on the route*, and resume the bidirectional data transmission, instead of relying on the source to discover another route.

3.4.2.1 Types of Route Breaks

In a network of mobile nodes, a route break between directional nodes occurs when one of the nodes moves out of the radio range of its neighboring nodes in the route. Here, the link breaks are classified in the following two categories.

Beam-Switch Link Break: In a network of FDD nodes with a multi-beam directional capability, an intermediate node in the route may move from the current beam to one of the adjacent beams of its previous (or next) hop node. This kind of node movement, from one beam to another, causes the links in either direction to break and is called, the 'beam-switch link break'. For example, node 3 moves to a new position (3′) in the adjacent beam of both nodes 2 and 4, in Figure 37. Hence, the links between nodes 2 to 3 and nodes 3 to 4 are broken. However, at its new position, node 3′ is still in the range of both its neighboring nodes 2 and 4, but accessible through their different beams. The beam-switch link breaks can be locally repaired without altering the already existing route.

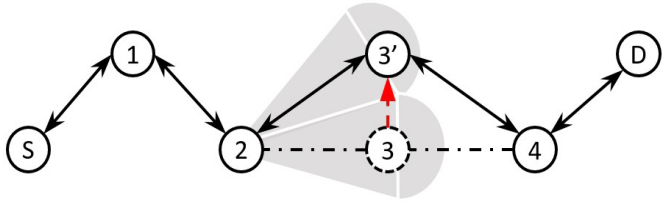


Figure 37. An illustration of a beam-switch link break due to movement of node 3 to a new position 3' (from the current beam to the adjacent beam of nodes 2 and 4).

Node-Switch Link Break*:* When a node in the route moves out of the transmission range of its neighbor nodes or becomes unreachable due to power failure (or insufficient transmit power), it causes the link to break. This type of break is called, the 'node-switch link break', where the failed node needs to be replaced and route be repaired to resume the data transmission. In Figure 38, when node 3 fails or moves out of the transmission range of its neighbor nodes, the links between

nodes 2 to 3 and nodes 3 to 4 are broken. In this situation, node 3 must be replaced with another node (node 5 or 6) for the route to be repaired. Thus, the node-switch link breaks can be locally repaired by replacing the unreachable node with another node in the neighborhood of the link break.

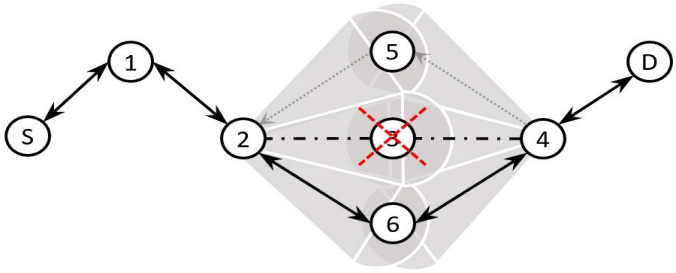


Figure 38. An illustration of a node-switch link break due to failure or movement of node 3.

Two major challenges for the repair of overlapped routes in BAOMDV protocol are:

- The route repair should provide a completely overlapped route. However, the independent local route repair for both the forward and backward routes may lead to the formation of non-overlapping routes between the source and destination nodes, which are not suitable for bidirectional traffic. For example, in the case of a route break in Figure 38, the forward route may recover through node 6 whereas the reverse route separately recovers through node 5 (as a replacement of node 3) which leads to a non-overlapping bidirectional route between S and D nodes. Note that the nonoverlapping routes are not as efficient.
- The locally repaired route should not be longer than the original route, since the AOMDV based routing protocols follow the condition of never forming a route longer than the one already advertised (the route acceptance rule), to maintain the loop freedom [78].

3.4.3 Control Packets for Bidirectional Local Route Repair

Before explaining the proposed route repair scheme (BAOMDV-LR), we describe the new control packets introduced for carrying out the local route repair. The proposed BAOMDV-LR scheme uses the node information over the next two hops to locally repair a broken route. The routing table structure of BAOMDV protocol has been modified to accommodate the *'Secondhop node'* address information (as shown in Table 9), which represents the node after the next-hop node along the route.

Table 9. Route list for supporting the bidirectional local route repair

Next_hop1	Next_hop_beam1	Second_hop1	Last_hop1	Hop_count1	Timeout1
Next_hop2	Next_hop_beam2	Second_hop2	Last_hop2	Hop_count2	Timeout2
.	.	.	.	.	.
.	.	.	.	.	.

The structure of the RREQ, RREP and RREP_ACK routing packets is similar to those used in BAOMDV protocol, with an additional information field called *'Previous Node Address'*, which represents the address of the node previous to the transmitting (or forwarding) node.

To perform bidirectional local route repair, we introduce three new control packets, called the *local repair route request packet* (*LR_RREQ*), *local repair route reply packet* (*LR_RREP*) and *local repair route update packet* (*LR_UPDATE*).

The *LR_RREQ* packet structure in Table 10 is enhanced from the RREQ packet sent during the route discovery by adding the Beam ID, Previous Node Address, Unreachable Node Address and Recovery Point Address fields. The time-to-live (TTL) of this packet is limited to one hop in its IP header. The *LR_RREQ* packet is broadcasted to the neighboring nodes on the forward-facing beams towards the destination node. Here, the *'Beam ID'* field contains the transmitting beam ID of the packet; the *'Unreachable Node Address'* contains the address of the next hop node along the route to which the link is no longer available; and the *'Recovery Point Address'* contains the address of the node after the unreachable node along the route (i.e., the second-hop node of the node broadcasting the *LR_RREQ*).

The *LR_RREP* packet in Table 11 is enhanced from the RREP routing packet by adding the Beam ID, Unreachable Node Address and Recovery Point Address fields. This is a unicast packet sent as a reply to the *LR_RREQ* on the beam on which the *LR_RREQ* packet was received. Here, the *'Beam ID'* field contains the transmitting beam ID.

Table 10. LR_RREQ packet format (28 bytes)

Packet ID	Type	Beam ID	Reserved
Source Node Address			
Destination Node Address			
Sequence ID			
Previous Node Address			
Unreachable Node Address			
Recovery Point Address			

Table 11. LR_RREP packet format (24 bytes)

Packet ID	Type	Beam ID	Reserved
Source Node Address			
Destination Node Address			
Sequence ID			
Unreachable Node Address			
Recovery Point Address			

The *LR_UPDATE* packet shown in Table 12 is similar to the *LR_RREP* packet, but contains a few extra fields needed to add or update the route in the neighboring nodes. The *'Unreachable Node Address'* contains the address of the unreachable node. The *'New Node Address'* is the new node through which the route is being repaired. The *'Previous Node Address'* information is used to update the second-hop node information in the routing table. The *'Recovery Point Address'* contains the address of the node after the unreachable node to which the route needs to be repaired.

Table 12. LR_UPDATE packet format (32 bytes)

Packet ID	Type	Beam ID	Hop Count	Reserved
Source Node Address				
Destination Node Address				
Previous Node Address				
Last Hop Node Address				
New Node Address				
Unreachable Node Address				
Recovery Point Address				

3.4.4 Description of BAOMDV-Local Route Repair Protocol

In this section, we explain the proposed BAOMDV-LR protocol with the help of an example.

3.4.4.1 Link Break Detection

When a node cannot successfully participate in its handshake mechanism and fails to transmit the RTS packet to its neighboring node (after the maximum allowed retransmissions), the link is considered broken. Then the node begins the local route repair process and moves all the packets to be forwarded along the broken link from its MAC queue to a new holding queue, called the repair-wait-queue for the duration of the local route repair process. During the local route repair, all the incoming packets are also buffered in the repair-wait-queue. In routes with bidirectional traffic, the link break can occur along both the forward and reverse route. Thus, the same process of link break detection may occur on the other end of the broken link. Therefore, both nodes on either side of the broken link may detect the break. However, if both ends were to initiate the route repair, both newly repaired routes may not overlap with each other. Therefore, only the upstream node (i.e., the node towards the source node) of the link break initiates the local repair of a broken link along the route, while the downstream node (i.e., towards the destination) passively waits for the route to be repaired.

3.4.4.2 Repair Phase

1. *LR_RREQ Generation:*

The first step in the process of local route repair is to determine whether the link break is due to a beam-switch or node-switch. First, it checks whether the unreachable node causing the link break is still within the range of any of its adjacent forward-facing beams. Then the availability of a substitute node, through which the route can be repaired, is checked. For this, the upstream node broadcasts an LR_RREQ packet (with the TTL set to 1-hop) on all its forward-facing beams and waits for the *LR_RREP* replies. In the example shown in Figure 37, the LR-RREQ packet is broadcasted by node 2, with the unreachable node address set as node 3 and the recovery point address set as node 4 in its forward beams.

2. *LR_RREQ Process:*

Upon receiving an *LR_RREQ* packet, a node first determines whether it is the unreachable node mentioned in the *LR_RREQ* packet. If the receiving node is the unreachable node and is still able to connect to the recovery point node address, the node will reply with an *LR_RREP* (beam-switch) packet. The node also updates the *'next_hop_beam'* of the reverse path in its routing table and moves the data packets from its repair wait-queue to its appropriate MAC queue.

Upon receiving an *LR_RREQ* packet for the beam switch scenario in Figure 37, node 3 modifies the forwarding beam with the new beam information for node 2 in its routing table. Then, it moves all the data packets waiting in the repair-wait-queue to its new appropriate MAC queue and sends an *LR_RREP* packet back to node 2.

On the other hand, if the node that has received the *LR_RREQ* packet is not unreachable, but is able to connect to the recovery point node mentioned in the *LR_RREQ* packet (node 4 in this case), it will reply with an *LR_RREP* packet stating that it is a suitable substitute (replacement) node through which the route can be locally repaired in a node-switch scenario. Note that every node knows about its 1-hop neighbors through a neighbor discovery scheme. Since node 3 is not reachable in Figure 38, node 2 will not get any *LR_RREP* reply from it, but nodes 5 and 6 will receive the *LR_RREQ* request from node 2. Since nodes 5 and 6 meet the criteria for substitution, they both will reply with a *LR_RREP* because both nodes are able to connect to the recovery point node 4.

Other factors may also determine whether a node is a suitable substitute for route repair. For example, when a node is already participating in another route formed between the source and destination node pair, the BAOMDV-LR scheme does not use it for repairing the route, as the repaired routes may not satisfy the link disjoint condition for the existing multipath routes and can also lead to congestion.

3. *LR_RREP Process:*

After initiating the local route repair (i.e., after transmitting the *LR_RREQ* packet), the node waits for a certain period (*LR_Wait_Time*) to receive all the *LR_RREP* packets being sent back from its neighboring nodes. During this wait period, if the node receives an *LR_RREP* packet from the unreachable node, it is an indication of a beam-switch link break, and a beam-switch repair is performed immediately. In the beam switch example shown in Figure 37, upon receiving the LR-RREP reply from the unreachable node (node 3), node 2 modifies its forwarding beam field with the new beam information (received from *LR_RREP* packet) for node 3 into its routing table. Thus,

its forward route is updated. Then, it moves all the data packets from the repair-wait-queue to the appropriate MAC queue beam for node 3. On receiving the *LR_RREQ* packet node 3 modifies the reverse route to node 2. Thus, the route is repaired in both forward and reverse directions.

If the node that has initiated the repair does not receive an *LR_RREP* packet reply from the unreachable node within the wait period (*LR_Wait_Time*), it indicates that a node-switch link break has occurred. Then, it selects one of the suitable substitute nodes that replied with an *LR_RREP* packet to repair the link to the recovery point node. For example, in Figure 38, node 2 will receive *LR_RREP* replies from both nodes 5 and 6. After waiting for the *LR_Wait_Time*, node 2 selects node 6's *LR_RREP* reply to locally repair the route to node 4. Here, it is assumed that the LR_RREP reply from node 6 is received before node 5.

3.4.4.3 Update Phase: LR_Update Process

The *LR_Update* packet is a part of the node-switch procedure and is designed to notify the other hops (that are participating in the route) on both forward and reverse route. After receiving the LR_RREP packet and selecting the appropriate node for route recovery/repair, the node that initiated the route repair (i.e., node 2 in our example) updates the forward path field in its routing table (based on the selected substitute node) and transmits the LR_Update packet to the node which is selected for route repair (see Figure 39).

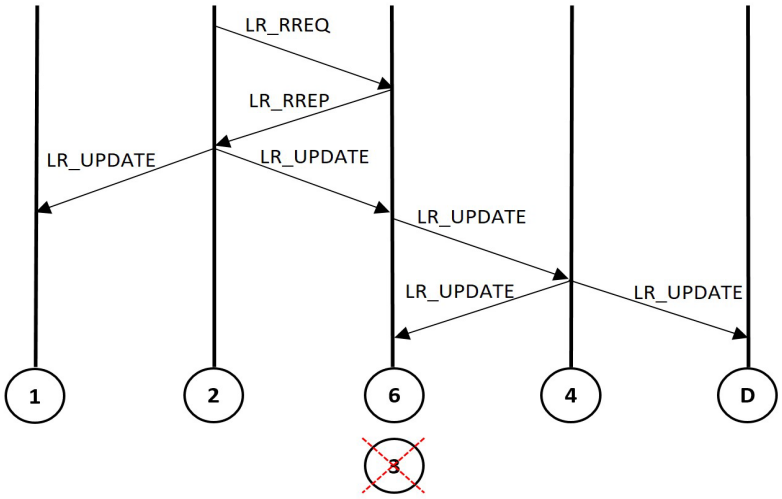


Figure 39. The node-switch local repair packet timing diagram.

Upon receiving the *LR_Update* with the *'New Node Address'* which is the same as itself, the new node (node 6 in Figure 39) realizes that it has been chosen for the nodes with repair and adds the route information in its routing table. Then it forwards the *LR_Update* packet to the recovery point node address (its next hop) after adding its own information to the packet. After receiving the update packet, the recovery point node modifies its existing route by updating the new node information in its routing table. Later, the new node (node 6 in our example) also receives an *LR_Update* from the downstream recovery point node updating it about the second-hop node along its forward path as shown in Figure 39. As the routing table of nodes contains the previous, next, and second-hop node information, the *LR_Update* packet is propagated up to two hops upstream

and downstream from the point of link break to update these nodes about the changes to the route due to node switch repair (see Figure 39).

In Figure 38, node 2 sends the *LR_Update* packet to nodes 6 and 1, with node 6 as the *New_Node*. Moreover, node 2 also modifies its next hop towards the destination as node 6 in its routing table. Upon receiving the *LR_Update* packet, node 1 updates its second-hop node towards the destination as node 6, whereas node 6 adds the route information into its routing table and forwards an updated *LR_Update* packet to node 4. Upon receiving this *LR_Update* packet, node 4 updates the next hop information in its routing table towards the source as node 6 and sends an updated *LR_Update* packet to nodes D and 6 and moves its own data packets from the repair wait-queue to the appropriate MAC queue beam. Finally, upon receiving the *LR_Update* packet, node 6 updates its second-hop node towards the destination as node D. Similarly, node D updates its second-hop node towards source as node 6 after it gets the *LR_Update* packet from node 4.

3.4.4.4 Local Repair Failure or Timeout

When the node that has initiated the route repair does not receive any reply within the *LR_Wait_Time* period, it assumes no suitable nodes exist for the local repair. Consequently, the route is marked as invalid and all the data packets in the repair-wait-queue are dropped, and an RERR message is sent to the source node. On the other end of the link break, the downstream route is also marked as invalid, the data packets are dropped from the repair-wait-queue and a RERR is sent to the destination node. A longer wait time of (2x*LR_Wait_Time*) is used at the downstream node, to allow the upstream node to detect and repair the link break, even when the downstream node detected the link break first. Moreover, in case of a node-switch repair, enough time is allowed for the *LR_Update* packet to be received by the downstream node.

4 RESULTS AND DISCUSSIONS

4.1 Performance Evaluation of ER Map-based Directional MAC Protocol

4.1.1 Simulation Setup

We evaluate the performance of our directional MAC scheme by comparing with a typical scheme, i.e., busy-tone based D-MAC [57], which doubles the contention window whenever there is a failure in channel access (due to no CTS response or busy-tone announcement indicating a busy channel). The test network is shown in Figure 40 with 17 nodes and four flows, where each flow is routed through three intermediate nodes. The maximum RF signal range is 1 km, and each node follows the Gaussian Markov mobility model. Four test cases are considered: (1) 1-flow test: Only one flow is active; (2) 2-flow test: Flow #4 and another selected flow are active; (3) 3-flow test: Flow #2, #4 and another selected flow are active; (4) 4-flow test: All four flows are active. We assume a surveillance application in which each node moves within its assigned mission region. Thus, each flow path consists of the same member nodes. Table 13 shows our parameter settings. To apply our ER spatial map and prediction model, we set up the ER map parameters as shown in Table 14. Link capacity is set to 10 Mbps and the beamwidth is 30°.

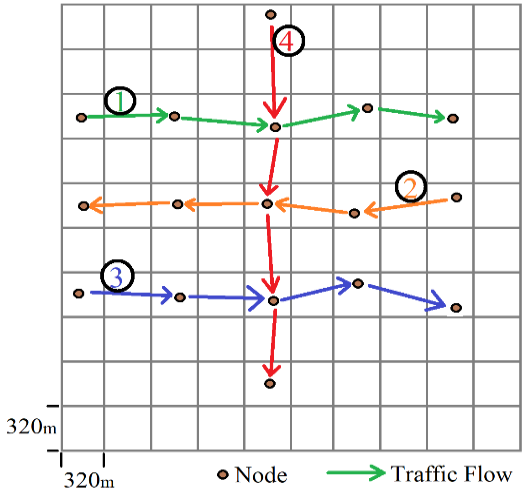


Figure 40. Test network scenarios.

Table 13. Simulation parameters.

Parameters	Values
Link Capacity	10 Mbps
Queue Size	5000 k bytes
Superframe Period	100 ms
DAFP Period	2 ms
Data Frame Size	1 k bytes
Simulation Duration	120 s
Hop Numbers Per Flow	Max 4
Data Rate of Each Flow	500 to 5000 kbps

Table 14. Antenna properties.

Parameters	Values
γ (ER shape coefficient)	100
Path Loss Exponent α	1.143
Beamwidth θ	30◦
Rx Gain Within Beamwidth G_{rm}	5 dBi
Rx Gain Outside Beamwidth G_{rs}	2 dBi
Tx Gain Within Beamwidth G_{tm}	6 dBi
Tx Gain Outside Beamwidth G_{ts}	3 dBi
Signal Range	1000 m
ER Case 1 Radius	1100 m
ER Case 2 Radius	553 m
ER Case 3 Radius	187 m
ER Case 4 Radius	94 m

4.1.2 Directional MAC Performance Analysis

Figure 41, Figure 43, Figure 45, and Figure 47 show the average throughput results for the four test cases. Figure 42, Figure 44, Figure 46, and Figure 48 show the average delay performances. Each test scenario was repeated multiple times with an average taken to generate the final results. The reference model busy-tone-based D-MAC, uses a basic contention window scheme and doubles its contention window whenever there is a failure to access the channel. It has some functions similar to our scheme, such as sharing the interference information among neighbors. Thus, its performance is better than general CSMA schemes. However, it does not have learning models to predict the interference changes.

In the single-flow case in Figure 41, our scheme performs similar to busy-tone-based D-MAC for lower data rates. However, for data rate > 4200 kbps, our scheme shows superior performance. As the number of flows increases, our scheme always performs better than the busy tone-based D-MAC, in terms of both throughput and delay. Our scheme achieves close-to-maximum allowable throughput. Note that the RF link can achieve a maximum throughput of half of the channel capacity due to the half-duplex communication. In contrast, the maximum throughput for busy-tone can only achieve 2200 kbps for 2-flow case and 1900 kbps for the 4-flow case.

By comparing Figure 41 and Figure 43, we can see that for the busy-tone-based scheme, the average throughput drops by over 1 Mbps for the 1-flow case after the flow rate reaches a certain value, and it drops by a total of 1.4 Mbps for 2 flows (thus each flow drops by 700 kbps on average). The dropping point for 2-flow case is at the flow rate of 2.2 Mbps, that is equivalent to a total rate of 4.4 Mbps for two flows, which matches the 1-flow case (see Figure 41). In contrast, our scheme has stable performance after saturation in all the cases and maintains its peak throughput with only slight variations.

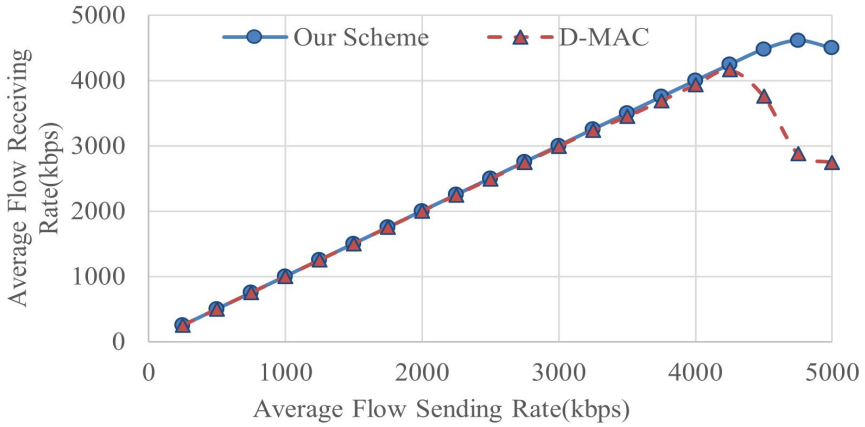


Figure 41. One-flow throughput test.

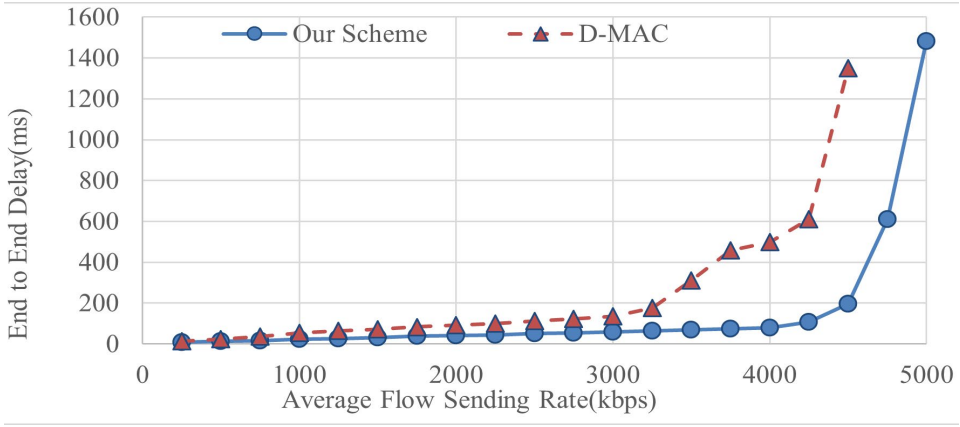


Figure 42. One-flow delay test.

The delay performance also shows similar trends. For the 1- flow case, the delay for busy-tone starts to get worse after the data rate exceeds 3200 kbps. Our scheme has a much better delay performance (3 times lower at high data rates) and maintains a good throughput even after 4000 kbps. For the 2, 3, and 4 flows, our scheme has a stable performance. However, the busy-tone scheme's delay performance becomes worse as the number of flows increases, especially after the data rate exceeds 2250 kbps.

The above results can be further explained as follows: while the busy-tone scheme tries to access the channel by using CSMA, our scheme uses superframes to coarsely coordinate the Tx/Rx schedule among one-hop nodes based on their traffic priorities. Therefore, our scheme has fewer collisions. More importantly, we have used ER map prediction to accurately determine the proper Tx time, which helps avoid serious directional interference. Thus, our scheme improves the QoS performance. Also, note that the busy-tone scheme ignores the queue state. At heavy traffic, such a random-access scheme may cause buffer overflow and packet losses.

As the number of flows increases, our scheme shows more advantages due to the fusion of all nodes' ER information into a local ER map and the prediction of ER map for the next time window. Our scheme can adjust each node's queue size and Rx/Tx schedule based on ER distribution. For example, a node without ER inference in the near future can speed up its Tx rate, while a node with an incoming directional interferer can slow down its rate to avoid a large packet loss rate; meanwhile, it can increase the queue size to accommodate more incoming packets from the upstream node.

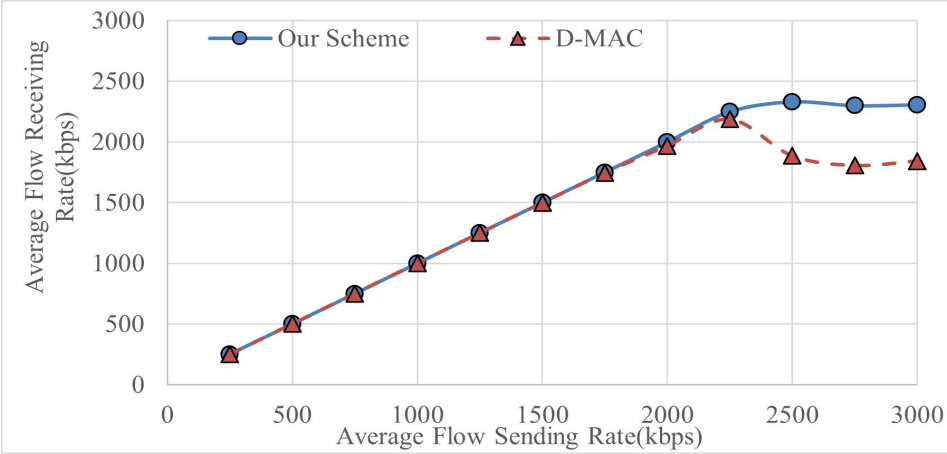


Figure 43. Two-flow throughput test.

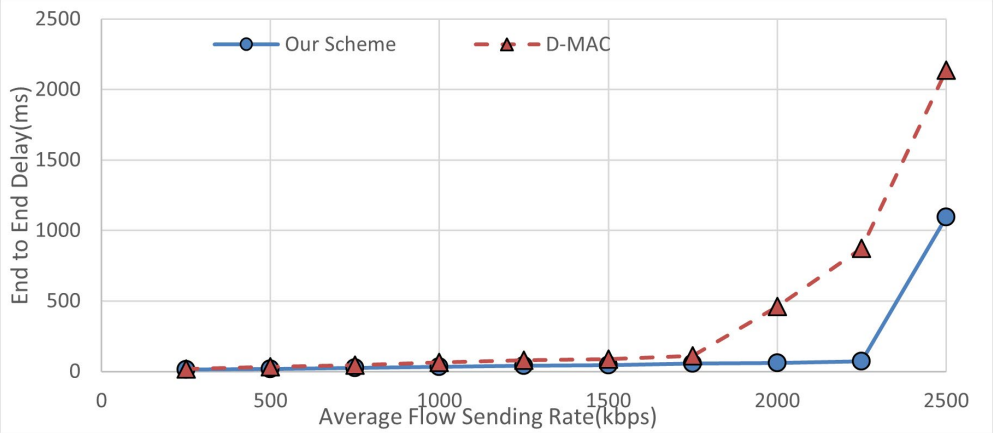


Figure 44. Two-flow delay test.

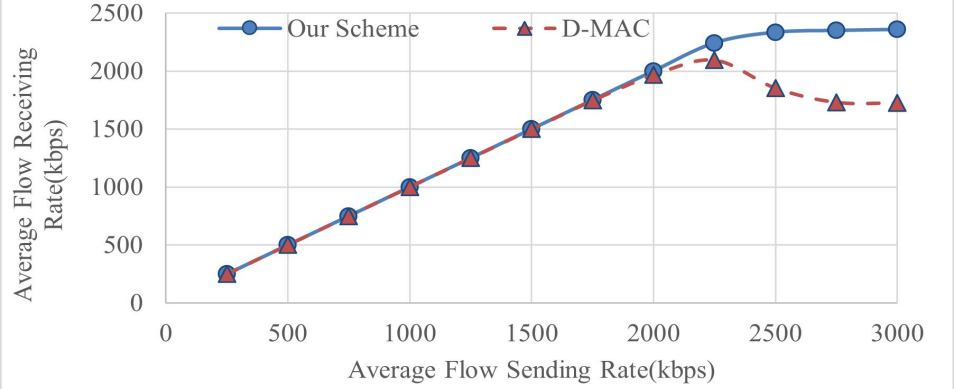


Figure 45. Three flow throughput test.

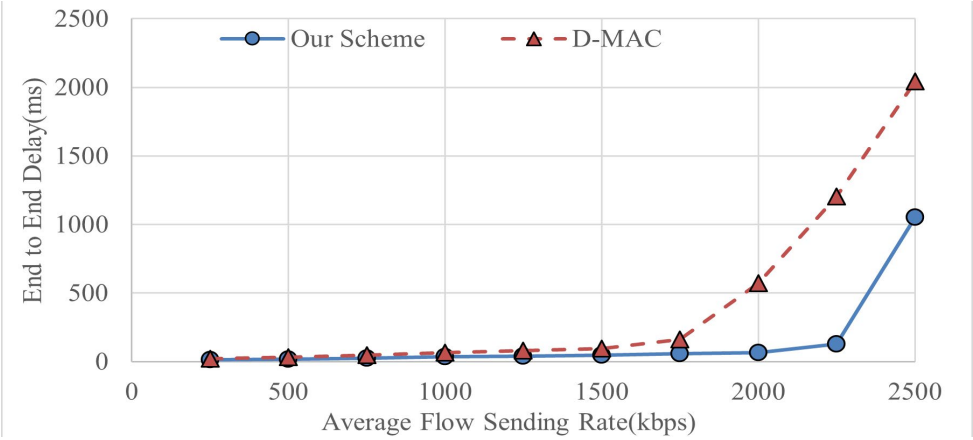


Figure 46. Three flow delay test.

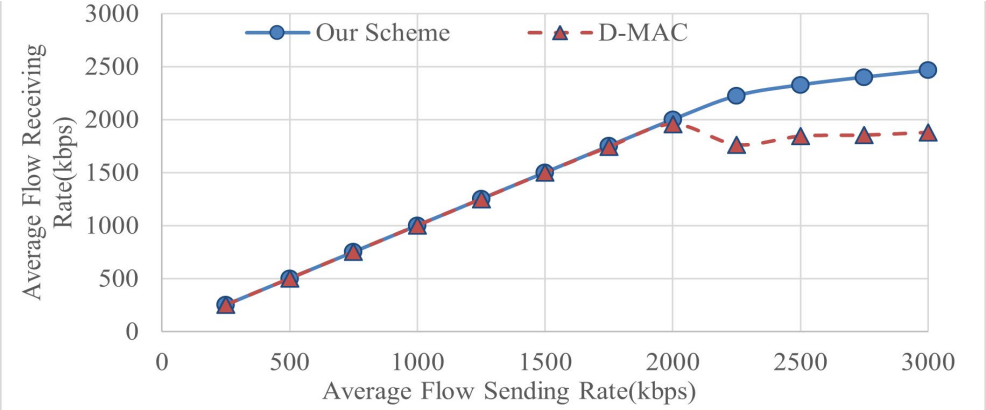


Figure 47. Four flow throughput test.

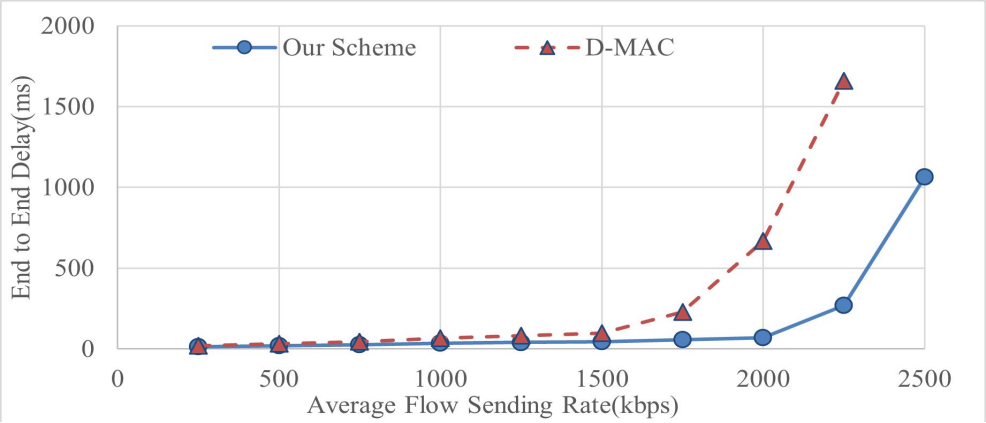


Figure 48. Four flow delay test.

4.1.3 Conclusions

We have proposed an intelligent MAC protocol in mobile ad hoc networks with directional antennas. Compared with traditional MAC schemes, our design has two main advantages: (1) the determination of the exact ER shape for typical link layouts, and (2) prediction of the next-time ER map for the optimized MAC parameter adjustment. This quantitative, predictive ER model helps to achieve a throughput-efficient MAC process. Our simulations have validated such a throughput efficiency.

4.2 Proactive Mobility and Congestion-Aware Routing Protocol: Performance Evaluation

4.2.1 Simulation Setup

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 [15][64][66-67][69]. 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 15. In our simulation, fixed-wing UAVs fly in a 8 × 8 km^2 area using the smooth-turn mobility model [84]. The 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, 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.

Table 15. 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	0.5 to 3.5 Mbps
Packet Size	1 kB
TTL	3 s
Simulation Duration	100 s, 200 s

4.2.2 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 × 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. The average number of routes computed is the total number of routes computed during the simulation, averaged over the simulation duration (100 s).

4.2.3 PDR Comparison

Here, we discuss the PDR comparison of all three schemes 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). 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).

4.2.3.1 Comparison of Instantaneous PDR Performance

Figure 49 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 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.

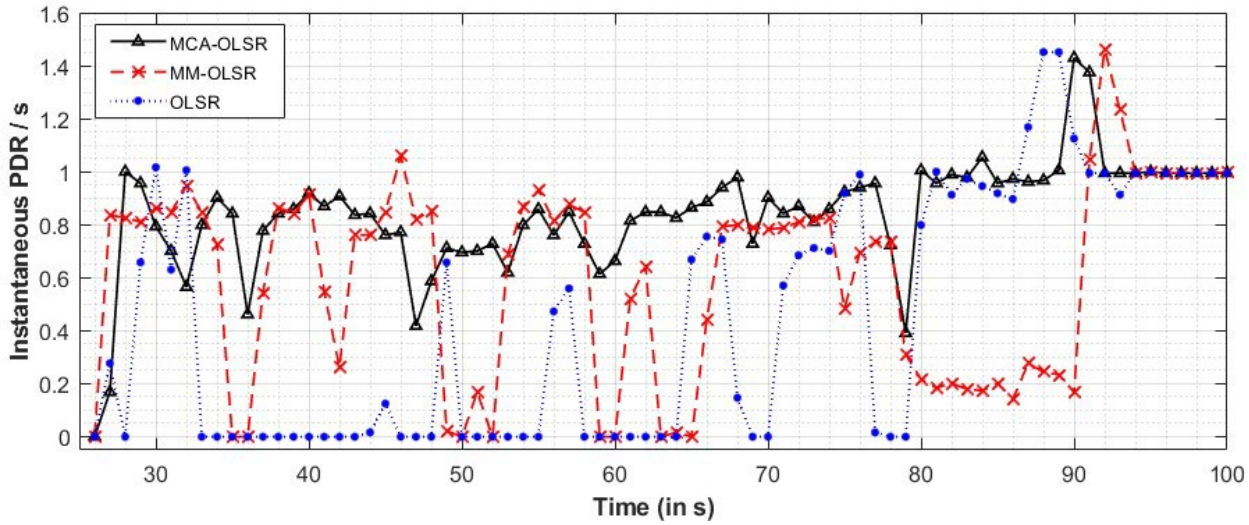


Figure 49. Instantaneous PDR comparison for our proposed MCA-OLSR, MM-OLSR and OLSR schemes, when 100 nodes fly at 50 m/s speed and one traffic flow generates data at 1.5 Mbps.

4.2.3.2 Impact of Number of Data Flows and Data Rate

Figure 50 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.

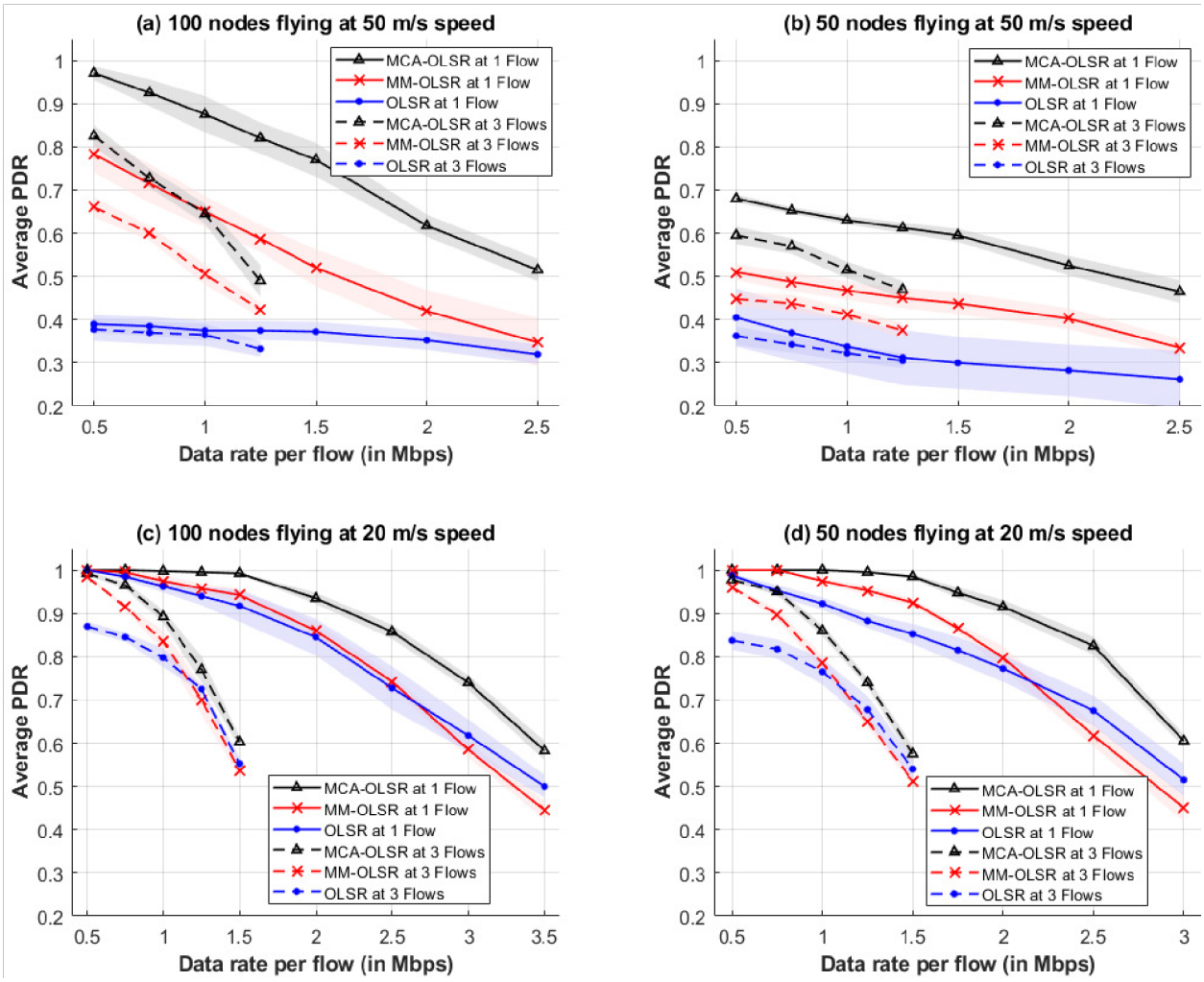


Figure 50. PDR comparison for our proposed MCA-OLSR, MM-OLSR and OLSR schemes for different number of data flows and data rates, for varying node density and speed. The subplots (a), (b), (c) and (d) have different ranges on the X axis.

Figure 50(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).

Figure 50(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.

Figure 50(c) shows 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.

Compared to Figure 50(a), the lower node speed results in fewer disconnected components and thus fewer packets lost to TTL expiry, giving higher PDR values. Similarly, Figure 50(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.

4.2.3.3 Impact of Node Speed

Figure 51 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 the 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*, 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.

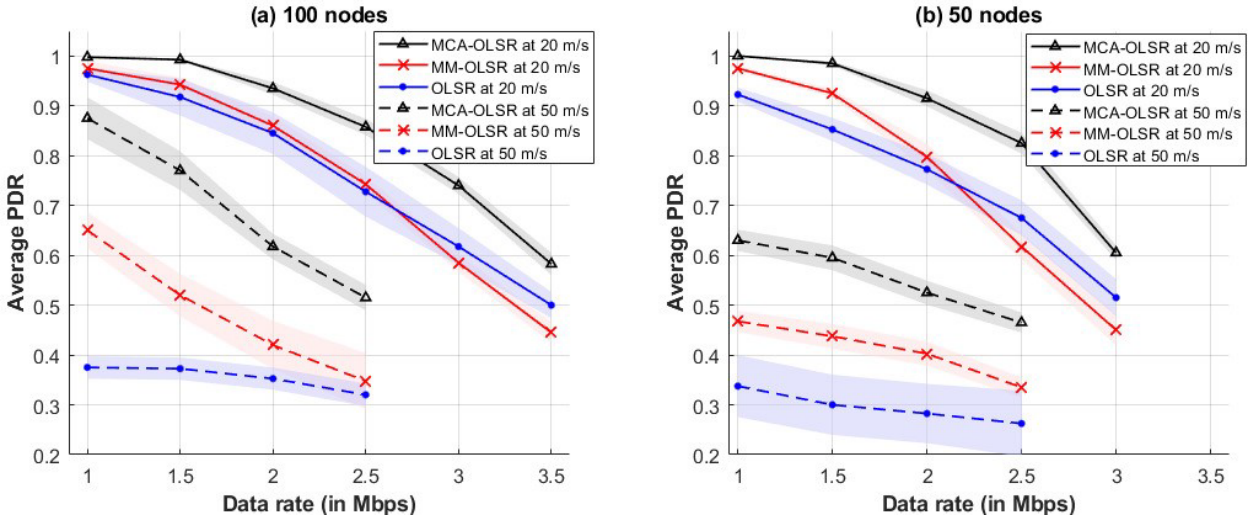


Figure 51. PDR comparison for our proposed MCA-OLSR, MM-OLSR and OLSR schemes at different node speeds, when the number of flows is one and node density is (a) 100 and (b) 50.

4.2.3.3 Impact of Node Density

Figure 52 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 Figure 52(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 (Figure 52(b)).

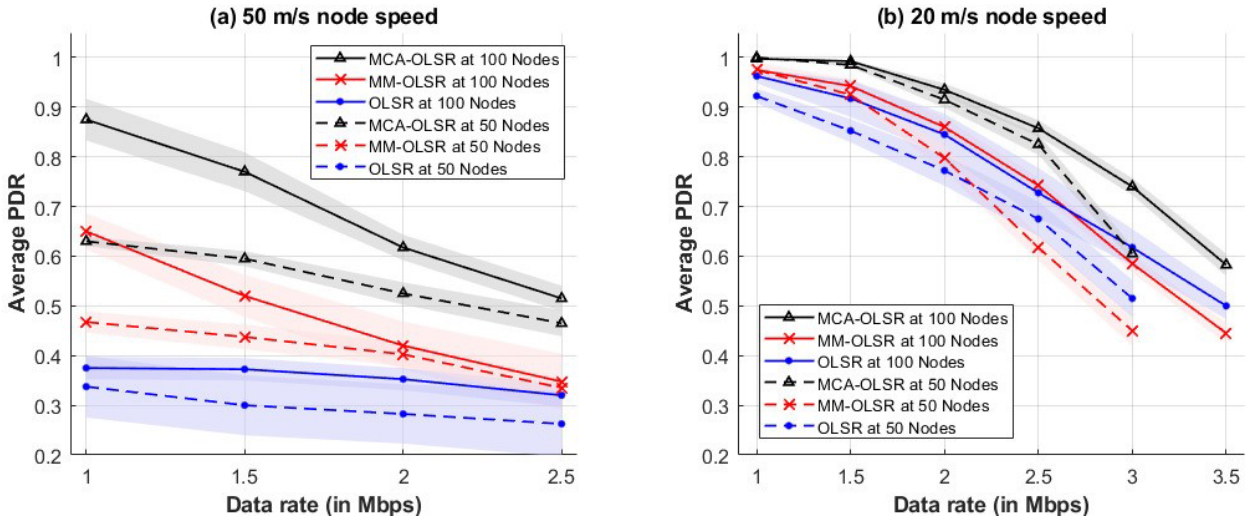


Figure 52. PDR comparison for our proposed MCA-OLSR, MM-OLSR and OLSR schemes at different node densities, when the number of flows is one and node speeds are (a) 50 m/s and (b) 20 m/s. Note that the subplots (a) and (b) have different ranges on the X axis.

4.2.4 End-to-End Delay Comparison

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 – 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 the shortest hop route, whereas MCA-OLSR can select a longer but more stable route, which increases the end-to-end delay.

4.2.5 Control (Signaling) Overhead Comparison

Due to the additional fields introduced in MM-OLSR and MCA-OLSR, their control overhead for both Hello and TC packets is higher than OLSR. We show the normalized control overhead in Table 16 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.

Table 16. 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

4.2.6 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 17. 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 17. 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

4.2.7 Conclusions

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.

4.3 Performance Evaluation of AI-based Directional Routing Protocol

4.3.1 Simulation Setup

We have simulated a directional MANET with the simulation parameters shown in Table 18. To test the impact of directional antennas on the route establishment, we have used different antenna beamwidths (30, 45, 60, 90 degrees). The simulation uses a canvas size of 4000 x 4000 meters with each node following a Gauss-Markov mobility model [82]. The average speed for each node is 60 m/s. Each simulation lasts 120 s to cover enough network topologies. The source and destination pairs are designated before the simulation starts.

Table 18. Simulation parameter settings.

Parameters	Values
Antenna beamwidth	30, 45, 60, 90 degrees
Effective RF signal range	1000 meters
Max link rate	20 Mbps
Network area (canvas size)	4k by 4k meters
Mobility model	Gauss-Markov
Average mobility speed	60 m/s
Average aircraft-aircraft distance	600 meters

4.3.2 Comparing Performance with Conventional Routing Scheme

We evaluate the QoS performance of each flow for different routing schemes with different antenna beamwidths as discussed below.

(1) Using OLSR: As shown in Figure 53(left), OLSR can identify a path with the minimum number of hops. Since it is unaware of the traffic density differences in different areas, the paths can pass through highly congested areas. Moreover, OSLR may not be able to handle multiple communication pairs well.

Here two paths have selected node 10 as their intermediate node. This can make node 10 become the traffic bottleneck since it needs to switch its directional antenna quickly between two paths (switching 4 times in a round).

(2) Using conventional QoS-based MANET routing protocol: Here we compare with non-prediction-based path quality scheme proposed in recent studies [25]. It is a classic reactive MANET protocol that calculates the path quality based on three metrics: path availability period (L), maximum residual path load capacity (PC), and path latency (LA). We use the following equation to calculate the path lost:

$$P_{QoS} = \frac{1}{3} \times \frac{L}{L_{max}} + \frac{1}{3} \times \frac{PC}{PC_{max}} + \frac{1}{3} \times \frac{LA}{LA_{max}} \tag{39}$$

where, L_{max} , $PC_{max,}$ $and\ LA_{max}$ are maximum path availability period, maximum residual path load capacity, and path latency, respectively.

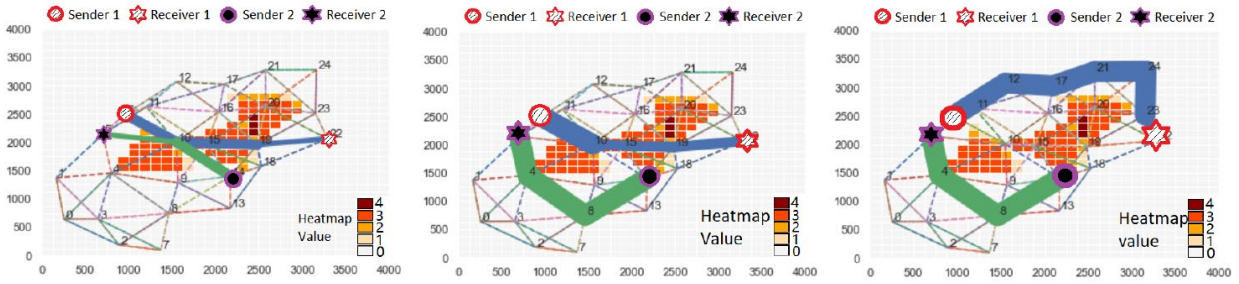


Figure 53. Comparison of three routing schemes. (left) OLSR routing protocol, (middle) using optimization model without DHM, (right) using optimization model with DHM. Here, X and Y axis represent the longitude and latitude in meters.

(3) Using optimization model only (without DHM): To investigate the benefit of DHM, we remove the DHM prediction scheme from our protocol and only keep the optimization-model-based routing algorithm. As we can see from Figure 53(middle), the optimization-based scheme is superior to OLSR because it observes each node's traffic load and avoids the intersection of multiple flows. This reduces the traffic bottleneck positions. However, the route still passes through congested areas.

(4) Using DHM-based optimization model (OPT with Heatmap): This represents our routing scheme. As shown in Figure 53(right), DHM completely avoids congested areas, while selecting a short path. It also avoids the intersection of multiple path flows.

To compare their packet delivery ratio (PDR) and end-to-end delay performance, we first set up the antenna's beamwidth to 30^0. Because our routing scheme can simultaneously establish multiple

flows with the consideration of mutual interference among flows, we investigate the performance with multiple active paths.

Figure 54 shows the PDR performance at the receiver. Our routing scheme (with optimization model and heatmap) achieves the highest PDR. For single flow, our scheme applies the optimized heatmap model to maintain satisfactory PDR as the data generation rate climbs up. The performance of the conventional QoS-based scheme decreases as the network becomes saturated with traffic. This is mainly because it only finds a single traffic path with the minimum path cost, while our scheme can find multiple paths that avoid heavy traffic areas when a single path cannot support the desired traffic load. This phenomenon can be clearly seen for the two-flows case. OLSR has 3× higher packet loss rate than our scheme, due to its ignorance of the interference between flows and the global traffic distribution. The conventional QoS-based scheme also has a significant performance drop when the flow rate exceeds 5 Mbps.

Figure 55 shows the end-to-end delay performance. Here the end-to-end delay mainly refers to queuing delay in each node. Although OLSR finds the shortest path, it does not have the lowest delay since end-to-end delay depends on multiple factors such as queue status, link rate, interference level, etc. For example, when a path has a node located in a heavy traffic area, that node will cause longer delay. Since our scheme considers both hop number and traffic distribution, it has the lowest delay. The conventional QoS-based routing scheme experiences performance degradation as the network traffic increases. The delay performance matches the above PDR trends, i.e., lower PDR indicates more congestion in the network, leading to higher routing delay.

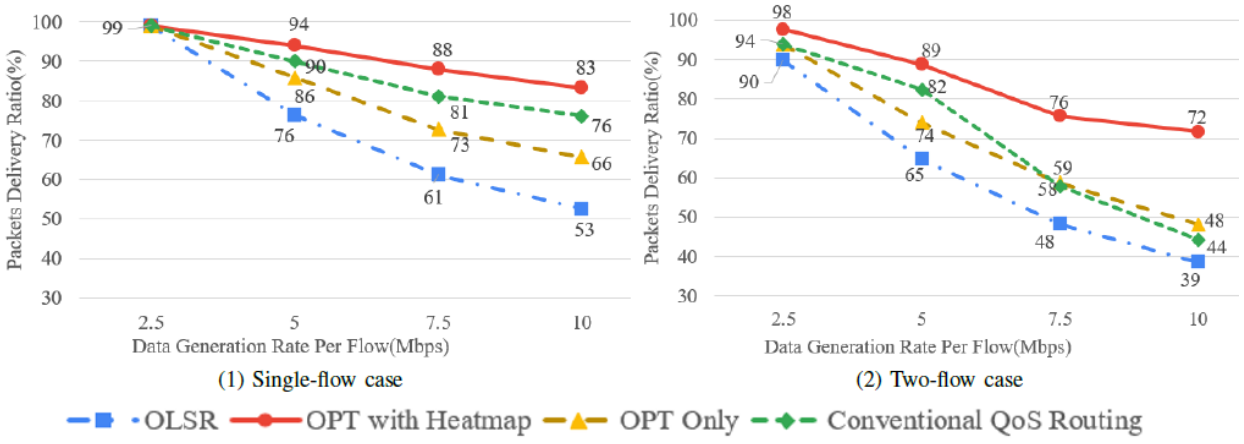


Figure 54. Packet delivery ratio (for 30^0 antenna beamwidth).

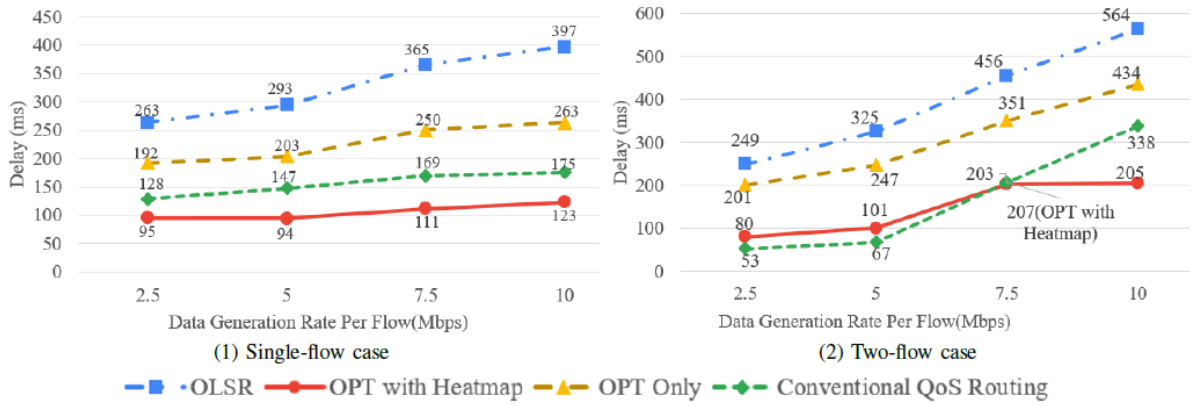


Figure 55. End-to-end delay (for 30^0 antenna beamwidth).

Figure 56 and Figure 57 show the QoS performance when the antenna's beamwidth is 60^0. Our scheme still has the highest PDR and the lowest end-to-end delay in both the single and two-flow cases. But the QoS performance gets worse as compared to the 30^0 beamwidth cases.

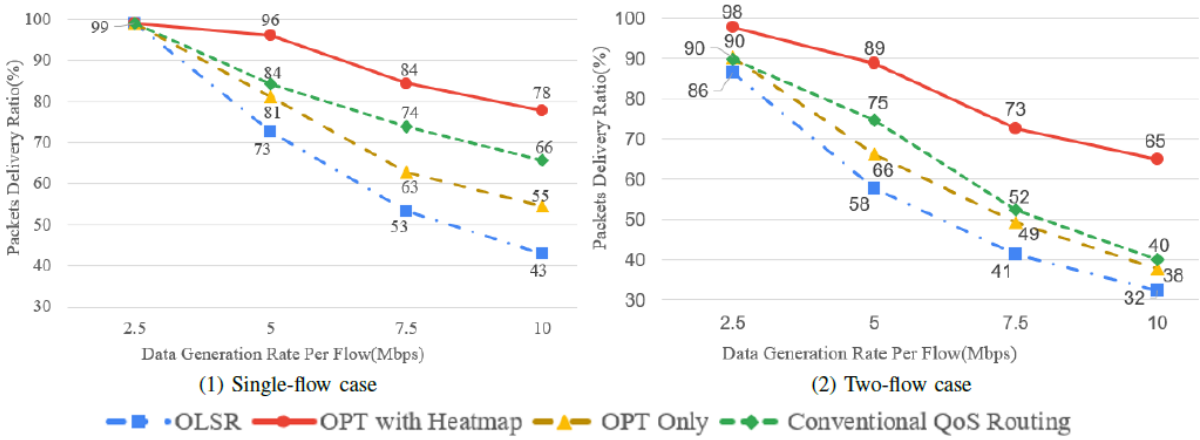


Figure 56. Packet delivery ratio (for 60^0 antenna beamwidth).

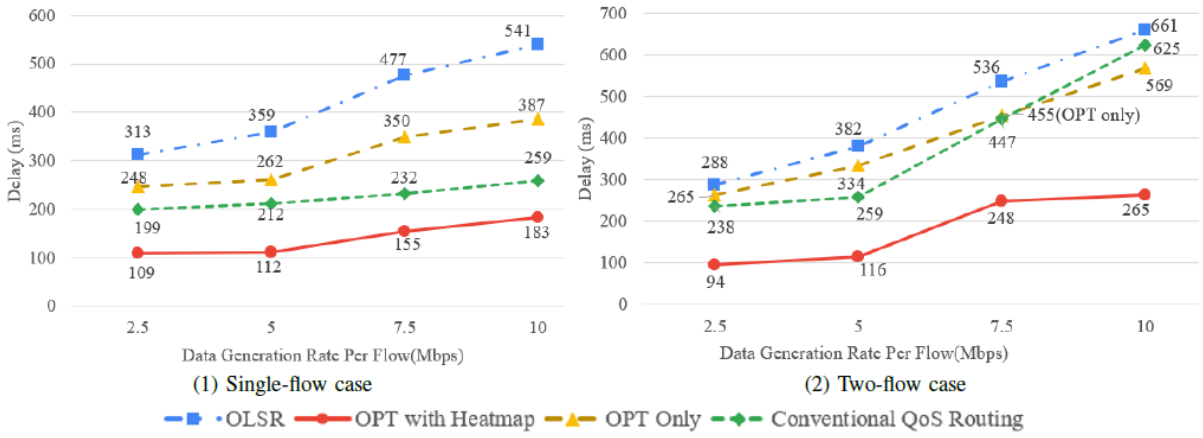


Figure 57. End-to-end delay (for 60^0 antenna beamwidth).

4.3.3 Conclusions

We proposed the concept of the directional traffic heatmap to trace the traffic distribution dynamics in both spatial and temporal domains. Particularly we used an evolutionary deep learning model to predict the next-time heat map. Such a predicted network snapshot is used to build a new path. Moreover, an optimization model was proposed to make the new path avoid both congested network regions and the signal interference from nearby DAs, while it has a smaller number of hops. We compared our routing scheme with OLSR, AODV and two directional routing protocols, and demonstrated a better QoS performance. Our next research will consider the 3D network scenarios that require the establishment of 3D heat map.

4.4 Performance Evaluation of BAOMDV-LR Protocol

In this section, we evaluate the performance of the proposed BAOMDV-LR scheme, in terms of average packet delivery ratio (PDR), average end-to-end delay and protocol overhead, for multipath bidirectional flows in a network of mobile multi-beam FDD nodes. Note that the results would also hold for full-duplex nodes.

- *Packet Delivery Ratio (PDR):* It is the ratio of the number of packets received at the destination to the number of packets generated by the source. For a bidirectional flow, we take the average of the PDR for the forward and reverse routes between the source and destination pairs.
- *Average End-to-End Delay:* It is the average of time taken by all the received packets to travel on the forward and reverse routes between the source and destination pairs.
- *Control Packet Overhead:* Average number of routing control packets generated in the network per second, including the route setup, route update, and maintenance packets (i.e., RREQ, RREP, RERR, LR_RREQ, LR_RREP and LR_UPDATE).
- *Number of Route Discoveries:* It is the total number of new route discoveries that are initiated by the source nodes for all the routes in the network.

We have used a discrete event driven simulation framework, consisting of 50 FDD nodes equipped with multi-beam directional antennas (consisting of 8 beams) with a beamwidth of 45° and a transmission range of 1.2 km, where the nodes are randomly positioned in a 6 km × 6 km area. Every node already knows about its 1-hop neighbors through a neighbor discovery scheme. The FDMAC protocol [83] is used that uses a full-duplex three-way handshake (using the Request-to-Send (RTS) and full-duplex Clear-to-Send (FCTS' and FCTS") packets) to achieve synchronization between two nodes. Thus, the two nodes can transfer data packets and ACKs to each other at the same time, which significantly increases their link level throughput.

Table 19 summarizes the network and protocol parameters used in our simulation. We set up five simultaneous bidirectional data flows with different route lengths (from two to five hops). Each source and destination pair can form multipath (up to three) routes to transmit its bidirectional flow.

Table 19. Simulation parameters.

Parameters	Values
Network size	6 km × 6 km
Number of nodes	50
Transmission range	1.2 km
Antenna Beamwidth	45°
Propagation Model	Friis
Time to live	500 ms
Channel transmission rate	2 Mbps
MAC protocol	IEEE 802.11 (FD-MAC [83])
MAC queue length	Drop tail, 100 packets
Size of data packets	1 Kbyte

4.4.1 Average PDR for Bidirectional Flows

The average PDR of five bidirectional flows is shown in Figure 58 for different data rates (from 1.5 to 3 Mbps) and node mobility of 20 m/s (72 km/h) and 60 m/s (216 km/h). The nomenclature used for representing different schemes in Figure 58 to Figure 61 is "Scheme Name - Speed". The numbers on each plot represent the difference in the average PDR between the forward and reverse routes for bidirectional flows. As expected, the average PDR generally decreases with increase in the data rate and/or node mobility. Here, the use of multipath routes and bidirectional communication on multi-beam FDD nodes helps in achieving higher data rates by all the protocols tested by us. But the BAOMDV scheme achieves a considerably higher PDR than the OAOMDV and AODV schemes due to the use of multipath overlapped routes.

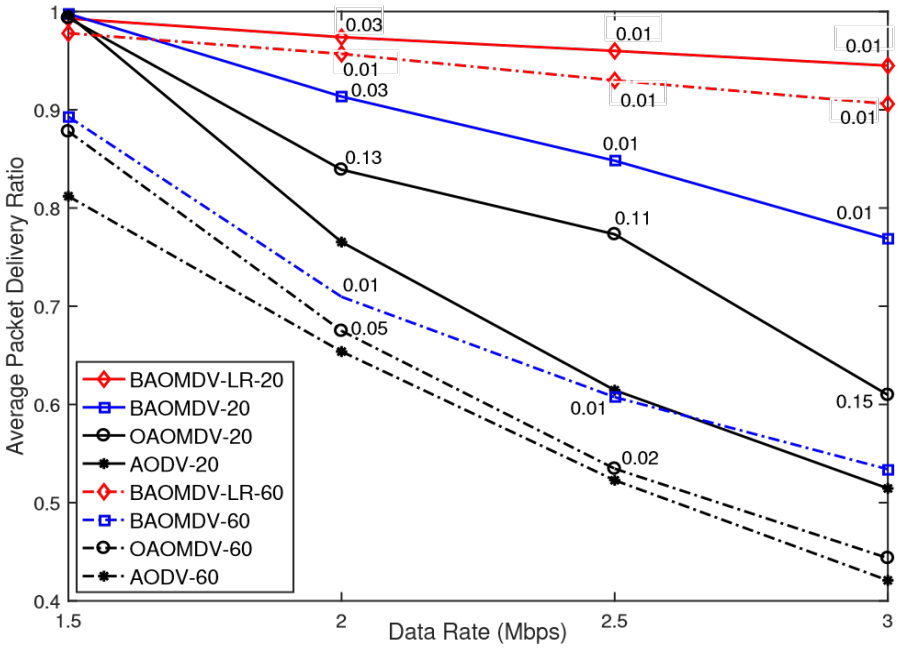


Figure 58. Average PDR for bidirectional traffic at medium and high node mobility.

In Figure 58, the proposed BAOMDV-LR protocol achieves a higher PDR than the BAOMDV routing protocol by more than 11% for a data rate of 2 Mbps and 25% for a higher data rate of 3 Mbps at 20 m/s node mobility. For high node mobility (60 m/s) where the multipath routes tend to break more frequently, the proposed BAOMDV-LR protocol achieves a significantly higher PDR than BAOMDV (more than 20% higher PDR for a data rate of 2 Mbps and 45% for a data rate of 3 Mbps). Unlike other routing schemes, the PDR achieved by our proposed BAOMDV-LR scheme does not decrease much when the data rate and/or node mobility increase. This demonstrates that our proposed local route repair scheme effectively repairs the routes and introduces low overhead and low packet losses even for high node mobility.

Our results also demonstrate that there is less than 3% difference in the average PDR for the bidirectional flows on all the forward and reverse paths in BAOMDV-LR and BAOMDV schemes, compared to 11% to 15% difference in OAOMDV scheme. This result demonstrates that the overlapped reverse paths cannot be formed in many cases in OAOMDV.

4.4.2 Average End-to-End Delay

Figure 59 shows that the BAOMDV-LR scheme achieves up to 20% lower average end-to-end delay for medium node mobility (20 m/s) and up to 35% lower delay for high node mobility (60 m/s), as compared to the BAOMDV scheme. Another interesting observation here is the impact of node mobility on each scheme. We observe up to 11% higher delay as the node speed increases from 20 to 60 m/s for the BAOMDV scheme (for different data rates). However, in the BAOMDV-LR scheme, we observe a persistent low delay and resiliency to node mobility due to its effective route recovery capability. This can play an important role in delay-sensitive and quality of service (QoS)-aware applications.

Further, the BAOMDV scheme has a lower end-to-end delay compared to the OAOMDV and AODV schemes for both medium and high node mobility. More specifically, BAOMDV has up to 25% lower delay for medium mobility and 10% lower delay for high mobility scenarios compared to the OAOMDV scheme.

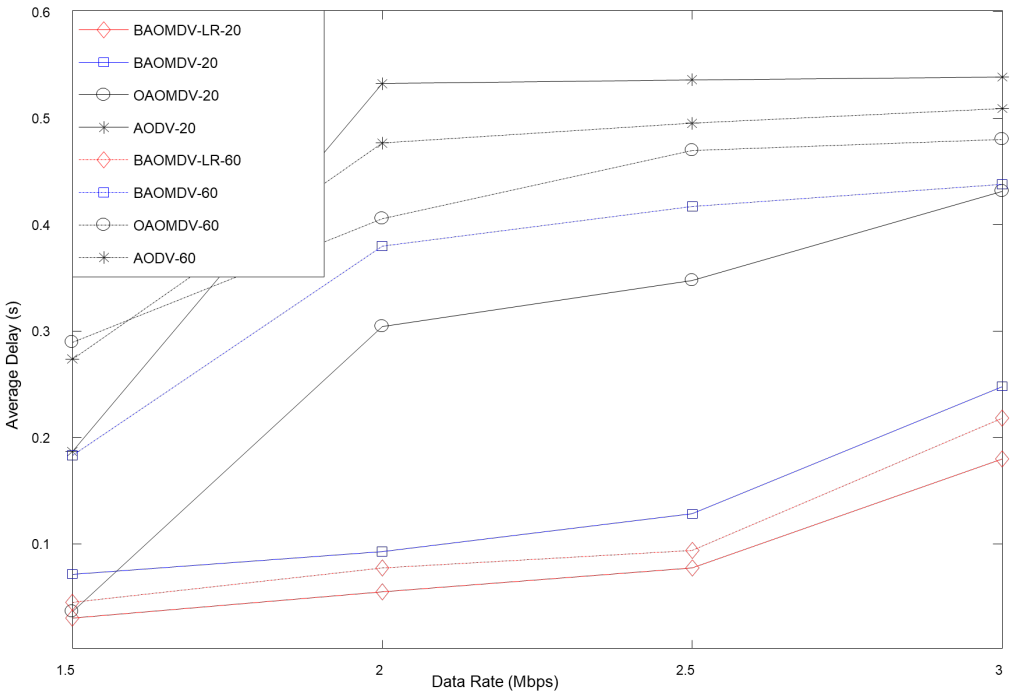


Figure 59. Average end-to-end delay for bidirectional traffic for medium and high node mobility.

4.4.3 Protocol Overhead

In Figure 60, the average number of control packets generated per second by the proposed BAOMDV-LR protocol is reduced to 53% at medium node mobility and to 28% at high node mobility, compared to the BAOMDV routing protocol. Also, in Figure 61, the number of route discoveries initiated by the source nodes is reduced to 50% at medium node mobility and 25% at high node mobility, compared to the BAOMDV routing protocol. Since the routes are maintained for longer duration due to route repair in BAOMDV-LR protocol, we see a significant reduction in route discovery frequency, which leads to a lower average overhead in the network.

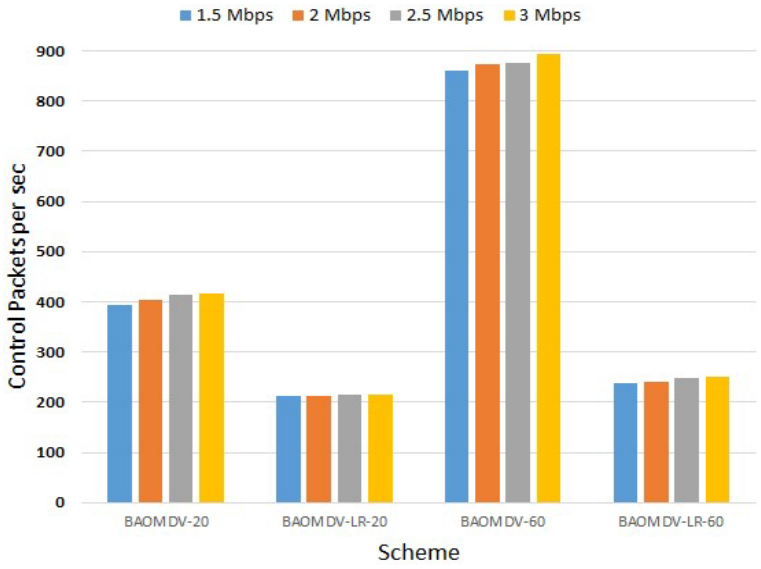


Figure 60. Average control packet overhead for medium and high node mobility per second.

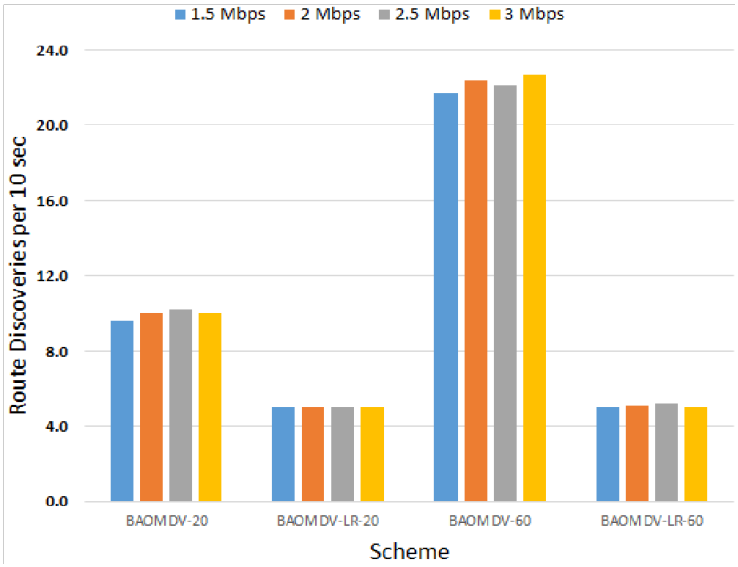


Figure 61. Average number of route discoveries initiated by all the source nodes for both low and high node mobility.

While the routing overhead and number of route discoveries significantly increase due to frequent route breaks at higher node speeds in BAOMDV protocol, the proposed BAOMDV-LR scheme effectively carries out the local route repair and hence does not experience a noticeable increase in the routing overhead and number of route discoveries. As a result, the proposed BAOMDV-LR scheme maintains multiple routes for a longer duration even in high mobility scenarios, thus improving the overall PDR and decreasing the end-to-end delay experienced by different flows in the network.

4.4.4 Conclusions

An airborne network of multi-beam FDD nodes was considered. An overview of BAOMDV routing protocol was provided to explain the establishment of multiple routes, where every forward and reverse route between a source and destination pair completely overlaps and supports the bidirectional traffic flows. Since the nodes use multi-beam directional communication, the packets can be simultaneously transmitted on multiple routes formed between a source and destination pair. Then, we investigated the effect of route breaks in the presence of multiple active routes among the source and destination pairs, and proposed a novel bidirectional, route repair scheme, called BAOMDV-LR. This scheme locally repairs the routes, while preserving their overlapping (in forward and reverse directions) and link-disjoint (or node-disjoint) characteristics, while avoiding the expensive route rediscovery. As a result, the proposed BAOMDV-LR scheme maintains multiple routes for a longer duration even in high mobility scenarios.

The simulation results demonstrated a significant performance improvement (in terms of packet delivery ratio, end-to-end delay, and routing overhead) as compared to the BAOMDV and OAOMDV protocols, especially at higher node speeds and data rates. This can play an important role in delay-sensitive and QoS-aware applications. The proposed scheme is equally suited for the in-band full-duplex nodes and would also work for other AODV and AOMDV based routing protocols, if they form the completely overlapped forward and reverse routes between a pair of source and destination nodes.

5 CONCLUSIONS

The use of directional communication presents new challenges for network protocol design. In this project, we designed novel MAC and routing protocols for ad hoc airborne networks by considering the network characteristics, distribution of traffic flows and RF interference. First, we discussed a directional MAC protocol by using a novel ER based directional interference model, followed by spatio-temporal ER-map prediction. Compared with traditional DMAC schemes, our protocol has two main advantages: (1) the determination of exact ER shape for typical link layouts, and (2) the prediction of the next-time ER map for optimized MAC parameter adjustment. This quantitative, predictive ER model helps to design a throughput-efficient MAC protocol. Our simulations have demonstrated its superior throughput efficiency.

Then we discussed a novel, cross-layer, mobility and congestion-aware proactive MCA-OLSR routing protocol for a decentralized airborne network. This protocol incorporated several new mechanisms, including multi-metric route selection, a longer lasting and low congestion route selection, preemptive route switching to avoid the broken and congested routes, and intelligent queue management. 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.

Next, we discussed a novel, AI-based proactive routing protocol based on the directional heat map from spatio-temporal deep learning. We introduced the concept of directional traffic heatmap to trace the traffic distribution dynamics in both spatial and temporal domains. Particularly we used an evolutionary deep learning model to predict the next-time heat map. Such a predicted network snapshot is used to build a new path. Moreover, an optimization model was built to make the new path avoid both congested network regions and the signal interference from nearby DAs. Our protocol demonstrated a better QoS performance compared to OLSR, AODV and two other directional routing protocols.

Finally, we discussed a novel local repair protocol (called BAOMDV-LR) for BAOMDV routes for a network of FDD nodes consisting of multibeam directional antennas. This scheme locally repairs the routes, while preserving their overlapping (in forward and reverse directions) and link-disjoint (or node-disjoint) characteristics and avoiding the expensive route rediscovery. As a result, the proposed BAOMDV-LR scheme maintains multiple routes for a longer duration even in high mobility scenarios. The simulation results demonstrated a significant performance improvement as compared to the BAOMDV and OAOMDV protocols, especially at higher node speeds and data rates. This can play an important role in delay-sensitive and QoS-aware applications. The proposed scheme is equally suited for the in-band full-duplex nodes.

Simulations are important for the study of airborne networks. We used the discrete event simulator, ns-3, as well as, Matlab, to evaluate the performance of the above MAC and routing protocols.

The protocols discussed in this report could be beneficial to other researchers studying the design of directional MAC and routing protocols for airborne networks.

6 REFERENCES

[1] "Airborne network architecture," USAF Airborne Network Special Interest Group, Version 1.1, Oct. 7, 2004.

[2] "Air force flight plan for aerial layer networking," USAF, July 16, 2010.

[3] "AFWARNET 2024: The USAF vision for warfighter networking," USAF.

[4] R. Jurdak, C. V. Lopes, and P. Baldi, "A survey, classification and comparative analysis of medium access control protocols for ad hoc networks," *IEEE Commun. Surv. Tut.*, vol. 6, no. 1, pp. 2-16, 2004. Doi: 10.1109/COMST.2004.5342231.

[5] L. X. Cai, L. Cai, X. Shen, and J. W. Mark, "Rex: A randomized exclusive region-based scheduling scheme for mmWave WPANs with directional antenna," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 113-121, Jan. 2010. Doi: 10.1109/TWC.2010.01.070503.

[6] H. Kim, S. Kim, C. Kim, and C. Park, "Directional antennas based MAC mechanism for spatial reuse," *2nd Int. Conf. Future Gener. Commun. Netw.,* pp. 334-337, 2008. Doi: 10.1109/FGCN.2008.175.

[7] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya, "On designing MAC protocols for wireless networks using directional antennas," *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 477-491, May 2006. Doi: 10.1109/TMC.2006.69.

[8] M. Takata, M. Bandai, and T. Watanabe, "A MAC protocol with directional antennas for deafness avoidance in ad hoc networks," *IEEE GLOBECOM 2007,* pp. 620-625. Doi: 10.1109/GLOCOM.2007.121.

[9] M. Sekido, M. Takata, M. Bandai, and T. Watanabe, "A directional hidden terminal problem in ad hoc network MAC protocols with smart antennas and its solutions," *IEEE GLOBECOM 2005,* pp. 5 pp.-2583. doi: 10.1109/GLOCOM.2005.1578227.

[10] G. Pei, M. A. Albuquerque, J. H. Kim, D. P. Nast, and P. R. Norris, "A neighbor discovery protocol for directional antenna networks," *IEEE MILCOM 2005*, vol. 1, pp. 487-492. doi: 10.1109/MILCOM.2005.1605730.

[11] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the 31st AAAI Conf. Artif. Intell. (AAAI'17)*. AAAI Press, pp. 1655–1661, 2017.

[12] 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. Tut.*, vol. 22, no. 2, pp. 1071–1120, 2020.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] J. Liu, Q. Wang, C. He, K. Jaffr`es-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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] B. Sliwa, C. Sch¨uler, 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.

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

[29] 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.

[30] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artif. Intell.*, vol. 259, 01 2017.

[31] A. Kurniawan, "Study on routing and congestion control with cross layer scheme in wireless ad hoc networks," in *Int. Conf. Elect. Eng. Comput. Sci. (ICEECS)*, 2014, pp. 79–84.

[32] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[33] C. Fischione, D. Koutsonikolas, S. Rangan, L. Simi´c, J. Widmer, X. Zhang, and A. Zhou, "Guest editorial millimeter-wave networking," *IEEE Sel. Areas Commun*., vol. 37, no. 12, pp. 2649–2652, 2019.

[34] X. Yu, T. Miyamoto, K. Obata, Y. Hosoda, J.-Y. Kim, M. Fujita, and T. Nagatsuma, "Direct terahertz communications with wireless and fiber links," in *44th Int. Conf. Infrared, Millimeter, and Terahertz Waves (IRMMW-THz)*, 2019, pp. 1–2.

[35] S. Rosati, K. Kru˙zelecki, L. Traynard, and B. R. Mobile, "Speed-aware routing for UAV ad-hoc networks," in *IEEE Globecom Workshops*, 2013, pp. 1367–1373.

[36] P.W.C. Chan, E.S. Lo, et al., "The Evolution Path of 4G Networks: FDD or TDD," *IEEE Commun. Mag.*, vol. 44, pp. 42-50, Dec. 2006.

[37] M. Parsinia and S. Devaraju and E. S. Bentley and S. Kumar, "Bidirectional AOMDV Routing Protocol for Full-Duplex AdHoc Networks," *Int. Conf. Comm. Workshops (ICC Workshops)*, pp. 1-6, May 2018.

[38] Common Data Link (CDL). [Online]. Available: https://www.globalsecurity.org/intell/systems/cdl.htm

[39] J. Boyd and J. Schlenzig, "Directional Networks for above 2GHz Software Defined Radios," *IEEE MILCOM*, Vol. 5, pp. 31763180, Oct. 2005.

[40] M. Parsinia, Q. Peng, S. Kumar, "Distributed Mode Selection for FDD Communication in Multihop Wireless Networks," *IEEE Trans. Aerosp. Electr. Syst.*, vol. 55(6), pp. 2921-2937, Dec. 2019.

[41] Z. Chen, L. Guan, X. Wang and X. Fan, "Ad hoc on-demand multipath distance vector routing with backup route update mechanism," *Int. Conf. High Performance Comp. Commun. and Int. Conf. Embedded Soft. Sys.,* pp. 908-913, June 2012.

[42] U. S. Kushwaha and P. K. Gupta, "AOMDV routing algorithm for wireless mesh networks with local repair (AOMDVLR)," *Int. Conf. Commun. Signal Processing,* pp. 818-822, April 2014.

[43] S. Kumar, S. Khimsara, et al., "Robust on-demand multipath routing with dynamic path upgrade for delay-sensitive data over ad hoc networks," *Comp. Net. Commun. Jour.*, vol. 2013, Article ID 791097, 13 pages, 2013.

[44] S.-C. Huang, H.-Y. Chang, "A bidirectional route repair method for wireless mobile ad hoc networks," *Computer Jour.,* vol. 58, pp. 338-353, Feb. 2015.

[45] S. Liu, Y. Yang, W. Wang, "Research of AODV routing protocol for ad hoc networks," *AASRI Procedia,* vol. 5, pp. 21-31, 2013.

[46] H. Gossain, T. Joshi, C. De Morais Cordeiro and D. P. Agrawal, "DRP: An efficient directional routing protocol for mobile ad hoc networks," *IEEE Trans. Parallel Distributed Sys.,* vol. 17, pp. 1438-1541, Dec. 2006.

[47] A. K. Saha and D. B. Johnson, “Routing improvement using directional antennas in mobile ad hoc networks,” *IEEE Global Telecomm. Conf. (GLOBECOM),* vol. 5, pp. 2902-2908, Dec. 2004.

[48] P. Zhou and W. Li, “A bidirectional backup routing protocol for mobile ad hoc networks,” *Int. Conf. Business Computing Global Informatization,* pp. 603-606, Oct. 2012.

[49] Z. Chen, R. D. Yates, and D. Raychaudhuri, “Dynamic node-disjoint multipath routing for millimeter wave networks using directional antennas,“ *Conf. Info. Sci. Sys. (CISS),* pp. 430-435, March 2016.

[50] L. Cikovskis and I. Slaidins, “Smart antennas for multi-path routing in ad-hoc wireless networks,” *Advances Wireless Optical Commun. (RTUWO),* pp. 268-271, Nov. 2017.

[51] Atmaca, Sedat & Celal, CEKEN & Erturk, Ismail. (2007). Improving Wireless TDMA/FDD MAC Performance with Multi-beam Directional Antennas. Lecture Notes in Engineering and Computer Science. 2167.

[52] Z. Wen, U. Heo and J. Choi, "A novel synchronization algorithm for IEEE 802.11 TDMA ad hoc network," *IEEE Int. Conf. Mobile Adhoc Sensor Syst.*, pp. 1-6, 2007. doi: 10.1109/MOBHOC.2007.4428712.

[53] Y. -x. Tian and K. -j. Wu, "A novel dynamic TDMA protocol for ad hoc networks using directional antennas," *Int. Conf. Elect. Inf. Control Eng.*, 2011, pp. 65-69, doi: 10.1109/ICEICE.2011.5777305.

[54] X. Zhong, S. Mei, Y. Wang and J. Wang, "Synchronization in TDMA ad hoc network," *IEEE 60th Veh. Technol. Conf.(VTC)*-Fall 2004, pp. 5011-5014, vol. 7. doi: 10.1109/VETECF.2004.1405052.

[55] 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, 26343-26361, Feb. 2021.

[56] S. Garg, N. Venkatraman, E. S. Bentley, and S. Kumar,, “An asynchronous multi-beam MAC protocol for multi-hop wireless networks,” in *IEEE Int. Conf. Comput. Commun. Netw. (ICCCN)* (pp. 1-9). July 2022.

[57] Z. J. Haas and Jing Deng, "Dual busy tone multiple access (DBTMA)-a multiple access control scheme for ad hoc networks," *IEEE Trans. Commun.*, vol. 50, no. 6, pp. 975-985, June 2002, doi: 10.1109/TCOMM.2002.1010617.

[58] A. A. Abdullah, L. Cai and F. Gebali, "DSDMAC: Dual sensing directional MAC protocol for ad hoc networks with directional antennas," *IEEE Trans. Veh. Technol.*, vol. 61, no. 3, pp. 1266-1275, March 2012, doi: 10.1109/TVT.2012.2187082.

[59] H. Bayat-Yeganeh, V. Shah-Mansouri, and H. Kebriaei, “A multi-state Q-learning based CSMA MAC protocol for wireless networks,” *Wireless Netw.*, vol. 24, pp. 1251–1264, 2018. https://doi.org/10.1007/s11276-016-1402-0.

[60] H. B. Pasandi and T. Nadeem, "Challenges and limitations in automating the design of MAC protocols using machine-learning," *Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, 2019, pp. 107-112. doi: 10.1109/ICAIIC.2019.8669008.

[61] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1277-1290, June 2019. doi: 10.1109/JSAC.2019.2904329.

[62] Introduction to Hamiltonian path problem: see WiKi, https://en.wikipedia.org/wiki/Hamiltonian_path_problem.

[63] B. Perunicic and N. Borovina, "New edge sorting criterion for maximum clique search algorithm in unit disk graphs," *XXIII Int. Symp. Inf., Commun. Automat. Technol.*, 2011, pp. 1-6, doi: 10.1109/ICAT.2011.6102096.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

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

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

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

[71] J. Du, X. Huang, F. Wu, and S. Leng, "Deep learning empowered QoS-aware adaptive routing algorithm in wireless networks," in *IEEE 20th Int. Conf. Commun. Technol. (ICCT)*, 2020, pp. 97–101.

[72] N. Abuzainab, et al., "QoS and jamming-aware wireless networking using deep reinforcement learning," in *IEEE MILCOM*, 2019, pp. 610–615.

[73] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surv. Tut.*, vol. 20, no. 4, pp. 2595–2621, 2018.

[74] D. Li, C. Yin, and C. Chen, "A selection region based routing protocol for random mobile ad hoc networks with directional antennas," *IEEE GLOBECOM*, 2010, pp. 1–5.

[75] A. Nasipuri, J. Mandava, H. Manchala, and R. Hiromoto, "On-demand routing using directional antennas in mobile ad hoc networks," in Proceedings 9th Int. Conf. Comput. Commun. Netw. (Cat.No.00EX440), 2000, pp. 535–541.

[76] M. Rezaee and M. H. Yaghmaee, “A new clustering protocol for mobile ad-hoc networks,” in *Int. Symp. Telecommun.*, 2008, pp. 376–381.

[77] C. Perkins, E. Belding-Royer and S. Das, “Ad hoc on-demand distance vector (AODV) routing,” *IETF, RFC 3561*, July 2003.

[78] M. K. Marina and S. R. Das, “Ad hoc on-demand multipath distance vector routing,” *Wireless Commun. Mob. Comput.,* vol. 6, pp. 969-988, Oct. 2006.

[79] K. Bao, F. Hu and S. Kumar, "AI-augmented, ripple-diamond chain shaped, rateless routing in wireless mesh networks with multibeam directional antennas," *IEEE Access*, vol. 6, pp. 24311-24324, 2018.

[80] M. O. Al-Kadri, A. Aijaz and A. Nallanathan, “X-FDR: A cross-layer routing protocol for multihop full-duplex wireless networks,” *IEEE Wireless Commun.,* vol. 26, pp. 70-77, April 2019.

[81] K. Kato and M. Bandai, “Routing protocol for directional full-duplex wireless,“ *Int. Sym. Personal Indoor Mob. Radio Commun. (PIMRC),* pp. 3239-3243, Sept. 2013.

[82] D. A. Korneev, A. V. Leonov, and G. A. Litvinov, “Estimation of mini-UAVs network parameters for search and rescue operation scenario with Gauss-Markov mobility model,” in *Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO),* 2018, pp. 1–7.

[83] W. Cheng, X. Zhang and H. Zhang, “RTS/FCTS mechanism based full-duplex MAC protocol for wireless networks,” *Proc. IEEE GLOBECOM*, pp. 5017-22, Dec. 2013.

[84] 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. Tut.*, vol. 16, no. 3, pp. 1221-1238, 2013.

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

[86] 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.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

ACK	Acknowledgement
AI	Artificial Intellingence
AN	Airborne Network
AODV	Ad-hoc On-demand Distance Vector
BAOMDV	Bidirectional Multipath AODV
BER	Bit Error Rate
BFS	Breadth First Search
BO	Buffer Occupancy
BPSK	Binary Phase Shift Keying
BT	Busy Tone
C2	Command and Control
CBR	Constant Bit Rate
CDL	Common Data Link
CEDN	Convolutional Encoder and Decoder Network
CH	Cluster Head
CHT	Channel Holding Time
CI	Channel Interference
CNN	Convolutional Neural Network
CSMA	Carrier Sense Multiple Access
DA	Directional Antenna
DACK	Directional ACK
DAFP	Directional Antenna Frame Period
DAWC	Directional-Antenna-aware Wireless Channel
DCF	Distributed Coordination Function
DCTS	Directional Clear to Send
DHM	Directional Heat Map
DIFS	Distributed Interframe Space
DL	Deep Learning
DMAC	Directional Medium Access Control
DNAV	Directional Network Allocation Vector
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
DRP	Directional Routing Protocol
DRTS	Directional Request to Send
DSR	Dynamic Source Routing
ER	Exclusive Region
ERL-MAC	ER-Learning based MAC
ETD	Estimated Time to Destination
FDD	Frequency Division Duplex

FEC	Forward Error Control
FIFO	First-In First-Out
GPS	Global Positioning System
HC	Hop Count
HOL	Head of Line
IL	Interference Links
ISR	Intelligence, Surveillance and Reconnaissance
ITS	Intelligent Transportation System
LLT	Link Life Time
LoS	Line-of-Sight
LR	Local Repair
MAC	Medium Access Control
MANET	Mobile Adhoc Network
MCA-OLSR	Mobility and Congestion-Aware OLSR
MDP	Markov Decision Process
MIMO	Multiple Input Multiple Output
ML	Maximum Likelihood
MM-OLSR	Multi-Metric OLSR
MOS	Mean Opinion Score
MPR	Multi Point Relay
MUP	Model Update Packet
NLIP	Node and Link Information Flooding Packet
NLoS	Non-LOS
OA	Omni-Directional Antenna
OABF	Omni-Directional Antenna Beacon Frame
OACF	Omni-Directional Antenna Contention Frame
OFDM	Orthogonal Frequency Division Multiple Access
OLSR	Optimized Link State Routing
PCF	Point Coordination Function
PDR	Packer Dropping Rate
PER	Packet Error Rate
PSNR	Peak SNR
PST	Packet Service Time
QoS	Quality of Service
RADA	Random Access Dice Array
ResNet	Residual Network
RF	Radio Frequency
RL	Reinforcement Learning
RLT	Route Lifetime
RREP	Route Reply

RREQ	Route Request
RSSI	Received Signal Strength Indicator
Rx	Reception
SEA	Sorted Edges Algorithm
SIFS	Short Interframe Space
SINR	Signal to Interference and Noise Ratio
SNR	Signal-to-Noise Ratio
ST	Spatio-Temporal
TC	Topology Control
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TSDP	Two-Step Discarding Policy
TTE	Time to Expiry
TTL	Time to Live
Tx	Transmission
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Packet
VNC	Virtual Node + Combination
WRED	Weighted Random Early Detection