https://doi.org/10.7840/kics.2025.50.1.82

Adaptive and Efficient Routing in Robotic MANETs

Hyeongheon Cha◆, Taeckyung Lee*, Dong-Hwan Yoon**, Eun-Jae Lim**, Jung-Kyu Sun**, Sung-Ju Lee°

ABSTRACT

Maintaining efficient communication is crucial in dynamic and resource-constrained environments such as military operations involving swarm robots. Due to their high computation and communication, traditional protocols such as Optimized Link State Routing (OLSR) often struggle with frequent topology changes and limited resources. We propose Adaptive Hello OLSR (AH-OLSR) and Adaptive Hello Expected Transmission Count (AH-ETX), which dynamically adjust control message intervals based on neighboring nodes' mobility. Our approach minimizes overhead and improves network throughput without modifying the message structure of OLSR. Simulation results demonstrate that AH-OLSR and AH-ETX significantly enhance efficiency (throughput normalized by control overhead), achieving up to a 143% increase compared with OLSR with static intervals. These advancements make AH-OLSR and AH-ETX suitable for real-world robotic swarm applications, supporting robust communication in challenging conditions.

Key Words : Mobile ad hoc networks, routing, robotic swarm

Ⅰ. Introduction

Mobile Ad Hoc Networks (MANETs) are critical communication paradigms in dynamic and resourceconstrained environments. In MANETs, mobile nodes cooperate to achieve common objectives such as emergency response, environmental monitoring, and military operations[1,2]. Within this context, robotic swarm MANETs represent a specialized and dynamic application of MANETs, where groups of autonomous robots collaborate to perform complex tasks[3] (Figure 1). The collaborative nature of these networks, combined with their inherent challenges of frequent topology changes, limited bandwidth, high mobility, and varying link quality[1,2,4], requires innovative routing solutions. Traditional protocols struggle to maintain efficient communication in these conditions, often leading to increased latency and excessive control message overhead[4-6].

Optimized Link State Routing (OLSR) is a proactive protocol designed for MANETs[7]. However, OLSR's static control message intervals fail to adapt to dynamic environments typical of (robotic swarm) MANETs[8-10], resulting in outdated routing information, suboptimal decisions, increased latency, and potential packet loss[4,11]. While dynamic interval adjustments have been proposed[12,13], these solutions often introduce additional complexity and energy consumption, further straining limited network resources. Harrag et al.[12] use complex calculations for each node based on predefined specific scenarios. Badis et al.[13] introduced a real-time adaptation mechanism but

※ This work was supported by KRIT grant funded by the Korea Government(DAPA, No. 20-107-C00-008-03)
◆ First Author : Korea Advanced Institute of Science & Technology (KAIST), School of Electrical Engineering, hyeongheon@kaist.ac.kr, 학생회원
° Corresponding Author : Korea Advanced Institute of Science & Technology (KAIST), School of Electrical Engineering, profsj@kaist.ac.kr, 종신회원
* Korea Advanced Institute of Science & Technology (KAIST), School of Electrical Engineering, taeckyung@kaist.ac.kr
** LIG Nex1, donghwan.yoon@lignex1.com; eunjae.lim@lignex1.com, 정회원; jungkyu.sun@lignex1.com
논문번호 : 202408-176-C-RU, Received August 16, 2024; Revised September 18, 2024; Accepted September 23, 2024

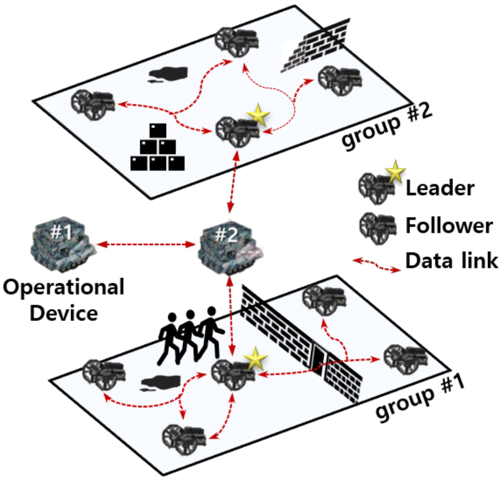

Fig. 1. Robotic Swarm MANET Scenario.

added communication and computation overhead from complex control messages, heuristic MPR selection, and complex path computations.

We propose the Adaptive Hello OLSR (AH-OLSR) protocol to address these limitations. AH-OLSR improves upon OLSR by enabling individual nodes to adjust their control message transmission intervals dynamically. These adjustments are based on current network conditions, particularly the changes in neighboring links. We achieve this adaptability with minimal modifications to the existing OLSR protocol without changing the structure of OLSR Hello messages. This approach avoids significant increases in computational overhead. By automatically adapting transmission frequency based on neighbor mobility, AH-OLSR reduces routing overhead and increases network throughput, improving overall network efficiency.

In addition to routing protocols, link metrics play a key role in MANETs. Existing link metrics, such as Expected Transmission Count (ETX)[14], face a significant challenge in AH-OLSR that uses variable Hello intervals. When a packet loss occurs, ETX might not accurately determine the time interval during which packets were lost. For instance, in OLSR with a fixed Hello interval, the ideal number of Hello packets expected during a given period is known, making ETX calculations straightforward. However, in AH-OLSR, where nodes can autonomously change their Hello packet intervals, the receiving node cannot accurately determine the ideal number of Hello packets that should have been received. This uncertainty in the expected Hello packet count hampers the accurate calculation of ETX. Consequently, not understanding the timing and expected frequency of packet receptions can lead to inaccurate link quality assessments, potentially causing suboptimal routing decisions in dynamic MANET scenarios.

To address this issue, we propose a new link metric named Adaptive Hello Expected Transmission Count (AH-ETX). Instead of counting expected transmissions, AH-ETX quantifies the expected duration of stable connectivity by calculating the stable periods between successfully received Hello messages. This approach aims to provide a more accurate link quality assessment in dynamic MANET conditions such as robotic swarms.

We evaluated AH-OLSR and AH-ETX in a simulated robotic swarm MANET scenario, where multiple groups work towards common goals. This scenario exemplifies coordinated search and rescue operations or tactical military deployments. Our results show that AH-OLSR and AH-ETX significantly outperform traditional and static OLSR. Compared with OLSR’s shortest control message interval, our approach reduces routing overhead by up to 59%. When compared with OLSR’s longest interval, it increases network throughput by up to 25%. These improvements combine to boost overall network efficiency by 143%. Such enhancements demonstrate that AH-OLSR and AH-ETX are suitable for robotic swarm networks requiring high efficiency and adaptability.

Ⅱ. Background

Optimized Link State Routing (OLSR)[7] is designed for Mobile Ad Hoc Networks (MANETs) to enhance efficiency in decentralized environments. As a proactive protocol, OLSR maintains up-to-date routes to all nodes, ensuring optimal paths are immediately available for data transmission.

A key feature of OLSR is multipoint relays (MPRs). OLSR employs a selective forwarding mechanism through MPRs, where each node selects a set of MPRs from its immediate neighbors to reach all nodes two hops away. This approach reduces redundant message transmissions and network overhead, improving efficiency in dynamic networks.

OLSR utilizes several types of control messages: Hello messages for neighbor discovery, TC (Topology Control) messages for topology dissemination, MID (Multiple Interface Declaration) messages for announcing multiple interfaces, and HNA (Host and Network Association) messages for network association information.

In robotic swarm MANETs, OLSR faces challenges maintaining efficient routing while adapting to topology changes[15], including balancing the update frequency and overhead. Frequent updates can maintain accurate routing information but increase network traffic and resource consumption. In resource-constrained environments such as robotic MANETs, the balance between routing performance and efficient use of bandwidth and processing power is crucial.

To address these challenges, existing works have proposed various improvements to OLSR. These include incorporating location data[16-18], enhancing MPR selection strategies[19,20], and adjusting control message frequencies[15]. Other efforts focus on developing better routing metrics and adapting to network dynamics[14,15].

Despite these improvements, optimizing OLSR for efficiency in dynamic environments remains challenging. Current solutions often struggle to fully address the need for adaptive routing that maintains accuracy while minimizing overhead. The trade-off between routing precision and network efficiency is particularly important in robotic swarms, where resources may be limited and network topology can change.

This underscores the ongoing need for more adaptive routing mechanisms that can efficiently handle the demands of modern MANETs while optimizing resource usage. Developing such solutions is crucial for enhancing OLSR's performance in diverse network conditions, making it more suitable for emerging applications in dynamic, resource-constrained environments like robotic swarm MANETs.

Ⅲ. Routing Design

The limitations of OLSR in highly dynamic environments, particularly in robotic swarm MANETs, necessitate a new approach to routing. While OLSR's use of MPRs reduces network overhead, its static control message intervals fail to adapt to rapid topology changes. This inflexibility can lead to outdated routing information, increased packet loss, and inefficient use of network resources.

In robotic swarm scenarios where nodes collaborate towards common objectives, these issues become even more severe. Timely and accurate routing information is crucial for maintaining coordination among nodes. However, simply increasing the frequency of control messages is not a viable solution, as it would lead to excessive overhead and overwhelm the network.

3.1 Adaptive Hello OLSR

Adaptive Hello OLSR (AH-OLSR) builds on the foundational structure of OLSR, retaining its core protocol and MPR selection process. However, AHOLSR introduces a significant enhancement by allowing each node to dynamically adjust its Hello message transmission interval based on the mobility of its neighboring nodes. This dynamic adjustment improves network performance by reducing unnecessary control message overhead and adapting to the network's changing topology. By autonomously adjusting the frequency of Hello messages, AH-OLSR maintains upto-date routing information in highly mobile scenarios while minimizing network congestion in more stable conditions. Additionally, AH-OLSR ensures that the OLSR-specific control messages (TC, MID, and HNA) are adjusted to twice the Hello message interval, following RFC 3626[21] guidelines to balance update frequency and overhead. This synchronization helps maintain consistency in the network's routing information dissemination, further enhancing the overall network performance.

The design of AH-OLSR centers on two key concepts: the network mobility metric p and dynamic Hello interval adaptation. The network mobility metric p measures local network dynamics by tracking changes in a node's immediate neighbors over a set time window (Figure 2). This window is set to three times the Hello interval to ensure that even with the longest Hello interval of AH-OLSR, changes in neighboring nodes are detected. We use the RFC 3626[21] recommended Hello interval of 2 seconds as the maximum interval and include shorter intervals (1 second, 0.5 seconds, 0.25 seconds) to effectively adjust to dynamic network fluidity. Thus, the window for capturing mobility and updating intervals is set to six seconds. During this period, each node counts the changes in its direct neighbors, producing a mobility metric p that reflects local network dynamics.

AH-OLSR uses this mobility metric to dynamically adjust Hello message intervals. The intervals are chosen based on the observed ranges of p to ensure timely updates without causing unnecessary overhead. The specific ranges (5, 10, 15) were determined through extensive simulations and empirical testing within robotic MANET scenarios, showing that these thresholds effectively capture different mobility levels in Figure4. However, the network's operational situation might require different intervals and criteria. This adaptability allows AH-OLSR to be tailored for various deployment scenarios, enhancing performance for specific network requirements. Table 1 summarizes the example of Hello intervals based on the mobility metric p.

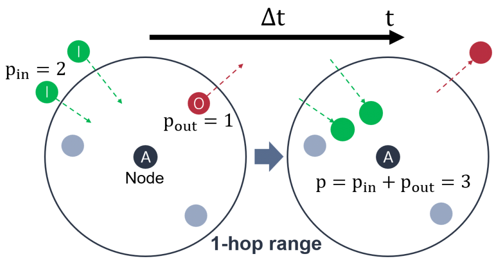

Fig. 2. Dynamic hello interval adjustment based on mobility metric.

Table 1. Example of adaptive Hello intervals based on mobility metric p.

Mobility Metric p	Hello Interval
$0 \leq p < 5$	2 seconds
$5 \leq p < 10$	1 second
$10 \leq p < 15$	0.5 seconds
$p \geq 15$	0.25 seconds

The parameters for AH-OLSR, including the time window (Δt), mobility metric ranges, and corresponding Hello interval adjustments, are designed to be flexible. This adaptability allows AH-OLSR to be tailored for various deployment scenarios, from relatively stable to highly dynamic environments. The values provided are a baseline configuration and can be adjusted to suit specific network requirements. When reducing network overhead is more important, longer intervals than two seconds can be added to the adaptive interval candidates, and the mobility range could be further subdivided as needed. Conversely, in situations where throughput performance is more critical and there are sufficient resources, shorter intervals than 0.25 seconds could be used to ensure rapid updates and minimal latency. This flexibility ensures that AH-OLSR could enhance performance and manage control overhead across diverse network conditions, improving its robustness and versatility.

AH-OLSR's dynamic adjustment of Hello intervals enables rapid adaptation to changing network conditions. In stable environments, nodes send fewer Hello messages, reducing overhead. Conversely, in dynamic conditions, nodes increase the frequency of updates to maintain accurate routing tables. This adaptive approach optimizes the balance between control message overhead and the need for timely routing information, thereby enhancing overall network performance.

Figure 2 demonstrates the concept of dynamic Hello interval adjustment. It shows node A as the node in focus. At time $t - \Delta t$, node A has a specific set of neighbors within its communication range. As time progresses, some neighbors leave while new ones join. Node A calculates these changes to determine its local mobility metric (p) and adjusts its Hello message interval accordingly.

Algorithm 1 outlines the dynamic Hello interval adjustment process in AH-OLSR. This mechanism enhances OLSR by enabling autonomous adaptation of control message intervals based on network conditions. AH-OLSR is particularly well-suited for scenarios where network efficiency is crucial, such as swarm robotics in dynamic and unpredictable environments. By adjusting Hello message intervals

Algorithm 1 Dynamic hello interval adjustment.

Input: Δt (time window), *incoming_nodes*, *outgoing_nodes*
Output: *hello_interval*
function calculate_mobility_metric(*incoming_nodes*, *outgoing_nodes*, Δt)
  $p_{in} \leftarrow \sum$ *incoming_nodes* in last Δt seconds
  $p_{out} \leftarrow \sum$ *outgoing_nodes* in last Δt seconds
  $p \leftarrow p_{in} + p_{out}$
  return p
function adjust_hello_interval(p)
  if $0 \leq p < 5$ **then**
    hello_interval $\leftarrow 2$
  else if $5 \leq p < 10$ **then**
    hello_interval $\leftarrow 1$
  else if $10 \leq p < 15$ **then**
    hello_interval $\leftarrow 0.5$
  else
    hello_interval $\leftarrow 0.25$
  return *hello_interval*
For every Δt **seconds:**
$p \leftarrow$ calculate_mobility_metric(*incoming_nodes*, *outgoing_nodes*, Δt)
hello_interval $\leftarrow$ adjust_hello_interval(p)

according to neighboring node mobility, AH-OLSR maintains reliable and efficient communication even in highly variable network topologies.

AH-OLSR retains OLSR's core routing functionalities while incorporating dynamic interval adjustment. This approach addresses the challenges of ad hoc networks, especially in rapidly changing environments such as military or exploratory missions. By reducing control message overhead without compromising routing information accuracy and timeliness, AH-OLSR offers a robust solution for efficient network management in MANETs.

3.2 Adaptive Hello Expected Transmission Count

AH-ETX offers a novel approach to link quality assessment in dynamic MANET environments. While ETX becomes less accurate when Hello intervals vary, AH-ETX adapts by focusing on the temporal aspects of link stability. By measuring the expected duration of stable connectivity, AH-ETX provides a more accurate representation of network dynamics in systems with variable Hello intervals.

The key feature of AH-ETX is estimating the 'effective uptime' of a link, regardless of Hello message frequency. This method accommodates the variable intervals of AH-OLSR and utilizes the information provided by these adaptive patterns. As a result, AH-ETX enables more informed routing decisions in dynamic environments.

AH-ETX calculation involves a temporal analysis of received Hello messages. Instead of simply counting packets, it considers the duration of stable periods between successfully received Hello messages. This method incorporates variable intervals and provides a more continuous link quality assessment than discrete counting methods.

For example, in an observation window with Hello messages received at varying intervals, AH-ETX is calculated as:

$$\text{AH-ETX} = \frac{\text{Total Observation Time}}{\text{Sum of Stable Connectivity Periods}}.$$

To calculate AH-ETX, we utilize the *Htime* field of Hello messages (Figure 3). The *Htime* field represents the time interval between Hello message transmissions, providing essential information about the frequency of these messages. The specific algorithm is as follows:

1. **Hello Message Counting:** Count the number of Hello messages received within a variable time window, denoted as T_w.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved             |     Htime     |  Willingness  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Link Code   |   Reserved    |       Link Message Size       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Link Code   |   Reserved    |       Link Message Size       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 3. RFC 3626[21] standard OLSR hello message field diagram.

2. **Weighting Hello Messages:** Use the *Htime* value, converted to seconds if necessary, to apply different weights to Hello messages, reflecting their transmission intervals. For example, if *Htime* values are (0.25/0.5/1/2) seconds, corresponding weights are (0.25/0.5/1/2), ensuring the total weight sums to T_w in each window. Using the *Htime* field value of the received Hello message as the weight, the expected transmission count value between the node links is approximately calculated. Using this value, the routing metric AH-ETX that automatically adapts to control messages with various Hello message intervals is obtained, creating a synergy effect with AH-OLSR.
3. **AH-ETX Calculation:** The weight is obtained from the received Hello packet's *Htime* value, which is converted to seconds. Define *Wi* as the weight for the *i*-th interval and *Pi* as the number of packets for the *i*-th interval. AH-ETX is then calculated as:

$$\text{AH-ETX} = \frac{T_w}{\sum_{i=0}^{n}(W_i \times P_i)}.$$

Here, *n* represents the total number of different *Htime* intervals of Hello messages received from a specific node during T_w time window, i.e., the types of *Htime* fields. This can also be expressed as *Htime_type*. Detailed calculation methods is in Algorithm 2.

This algorithm demonstrates how AH-ETX calculates link quality by focusing on the duration of stable connectivity rather than packet counts. Using the Htime field from each received Hello message, AHETX provides a meaningful link quality metric even in scenarios with variable Hello intervals, where ETX calculations would be problematic.

Although AH-ETX may initially be viewed as a performance metric, its primary function lies in its integration into the routing decision process within AHOLSR, where it actively informs route selection based on real-time link quality assessments. AH-ETX adapts to fluctuating Hello intervals based on node mobility, providing real-time link quality assessments that directly influence throughput, routing overhead, and network efficiency. Traditional ETX assumes fixed Hello intervals, which reduces its accuracy in dynamic environments. In contrast, AH-ETX works effectively with AH-OLSR by adjusting to varying Hello intervals, enabling more precise route selection and reducing overhead in highly dynamic networks.

Algorithm 2 AH-ETX calculation.

Input: *received_hellos*, T_w
Output: *AH_ETX*
weighted_sum ← 0
Htime_types ← unique set of Htime types from *received_hellos* of time window T_w
for all *Htime_type* **in** *Htime_types* **do**
    weight ← convert *Htime_type* to seconds
    count ← number of hello packets with *Htime_type* in *received_hellos*
    weighted_sum ← *weighted_sum* + (*weight* × *count*)
AH_ETX ← $\frac{T_w}{weighted\_sum}$
return *AH_ETX*

In summary, AH-ETX is integral to the AH-OLSR routing mechanism, optimizing network performance across key metrics by adapting to variable conditions, which is essential for dynamic and resourceconstrained environments like robotic MANETs.

Ⅳ. Performance Evaluation

To assess the effectiveness of AH-OLSR and AHETX in robotic swarm MANET environments, we conducted extensive simulations using the NS-3 network simulator. Our evaluation focuses on comparing the performance of our protocols against the original OLSR under various network conditions. We designed our simulation scenarios to reflect the dynamic and resource-constrained nature of robotic swarm MANETs.

4.1 Simulation Setup

The simulated network comprises 31 nodes: one operational device, five groups of five nodes, each with a leader node in each group, and five intermediate nodes, emulating scenarios such as swarm robotics or emergency response teams (Figure 4).

The 120-second simulation time progresses in three phases:

1. Initial deployment (0-10s): nodes disperse to designated positions.
2. Communication initialization (10-20s): TC, TM, and operational data traffic commence.
3. Operational phase (20-120s): nodes move randomly (max 1 m/s) within assigned areas. Every 20 seconds, a random node initiates a video upload.

The maximum movement speed of 1 m/s was selected based on typical speeds used in similar robotic swarm MANET designs and simulations[22,23].

We use four traffic types that represent diverse communication in typical robotic MANETs:

- Task Coordination (TC): 30 kbps (operational device to all nodes)
- Task Management (TM): 30 kbps (leaders to the operational device),

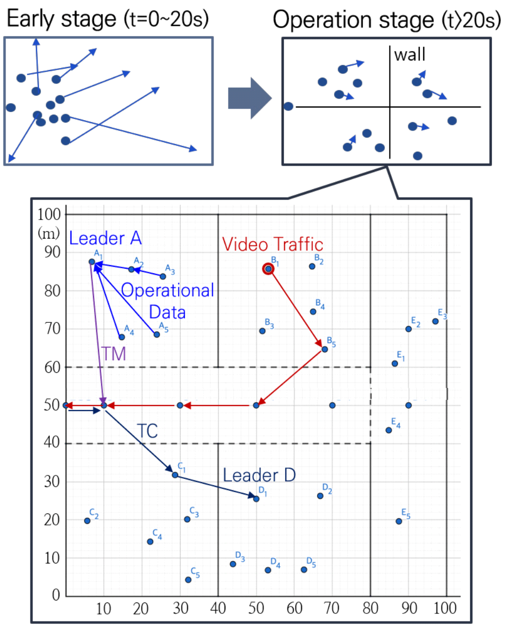

Fig. 4. Network operation scenario with topology and traffic patterns.

Table 2. Simulation parameters.

Parameter	Description
Network Simulator	NS-3
MAC Access Scheme	TDMA-based dynamic resource allocation
Operation Channel	TX 1 channel, RX 1 channel
Transmission Mode	12 Mbps
Transmission Frame Structure	100 slots per frame, 1 ms per slot
Routing Protocol	OLSR or AH-OLSR
Routing Metric	ETX or AH-ETX
Simulation Area	100m × 100m square
Node Mobility	0~20s: Spread to random location 20~s ~ 120~s: Random direction, up to 1 m/s
Transmission Protocol	UDP

- Video Traffic: 1.5 Mbps (followers to the operational device),
- Operational Data: 100 kbps (followers to leaders).

The simulation area is 100m × 100m with walls, using a free space path loss model[24] with 10% additional attenuation at walls and a network range of 30 meters. Figure 4 illustrates the spatial constraints and group structures.

Table 2 summarizes key simulation parameters, including MAC protocol, propagation characteristics, and mobility patterns. The chosen simulation parameters reflect typical operational conditions for robotic swarm MANETs in scenarios such as urban search and rescue or tactical military operations. We use dynamic TDMA protocol due to its deterministic nature and ability to avoid collisions in highly dynamic and congested environments, which is crucial for maintaining reliable communication in tactical and emergency scenarios[25-27]. This ensures predictable performance, which is often required in military and critical response operations.

4.2 Performance Metrics

We evaluate the performance using three key metrics: throughput, routing overhead, and efficiency.

Throughput measures the total amount of data successfully delivered over the network per unit time, expressed in kbps. Routing overhead quantifies the amount of control traffic generated by the routing protocol, also expressed in kbps, reflecting the cost of maintaining network routes. Efficiency is defined as the ratio of throughput to routing overhead, providing a comprehensive view of network performance by balancing data delivery against control traffic.

Efficiency is particularly crucial in our evaluation as it captures the trade-off between maximizing data delivery and minimizing control overhead. In robotic swarm MANETs, where resources are often constrained, this balance is essential for network operation. A higher efficiency value indicates that a protocol can maintain high throughput while keeping routing overhead low, which is desirable in dynamic, resourcelimited environments.

4.3 Results and Analysis

4.3.1 Performance Comparison of OLSR, AH-OLSR, and AH-OLSR with AH-ETX

Table 3 presents the performance comparison between OLSR with different fixed Hello intervals, AHOLSR, and AH-OLSR combined with AH-ETX.Bold values indicate the best performance, while underlined values show the second-best. Red values highlight the highest routing overhead and the lowest throughput among the compared configurations, underscoring the worst comparison performance. We should clarify that we did not include separate experimental results for OLSR with AH-ETX because, theoretically, there would be no performance difference between using AH-ETX and ETX in the context of static OLSR. In standard OLSR, the Hello interval is fixed throughout the network' s operation, which means that the dynamic adaptation of link quality metrics, as offered by AH-ETX, cannot be leveraged. Under static conditions, AH-ETX does not provide additional advantages over ETX since both metrics would operate under identical conditions, rendering their impact on routing decisions and network performance the same.

OLSR's performance with different Hello intervals demonstrates a clear trade-off between throughput and routing overhead. A short 0.25s interval allows quick adaptation to topology changes, yielding high throughput (3289 kbps) but significant overhead (568 kbps). Conversely, a 2s interval reduces overhead (199 kbps) at the cost of responsiveness, resulting in lower throughput (2569 kbps). Intermediate intervals of 0.5s and 1s offer a balance, with 0.5s achieving 3107 kbps throughput and 389 kbps overhead and 1s reaching 2922 kbps throughput with 263 kbps overhead.

AH-OLSR dynamically adjusts Hello intervals based on network conditions, striking a balance between throughput and overhead. It achieves 3111 kbps throughput with only 231 kbps overhead, resulting in an efficiency of 13.47. This demonstrates AH-OLSR's ability to maintain high efficiency by adapting to dynamic environments while reducing routing overhead.

By providing a more reliable link quality measure and adapting to varying Hello intervals, this approach achieves 3236 kbps throughput with 229 kbps overhead, yielding an efficiency of 14.13. This efficiency is *more than twice* that of OLSR with a 0.25s interval (5.79), highlighting the significant improvement offered by our proposed methods.

Table 3. Performance Comparison of OLSR, AH-OLSR, and AH-OLSR with AH-ETX.

Protocol	Link Metric	Hello Interval	Throughput(kbps)	Overhead(kbps)	Efficiency
OLSR	ETX	0.25s	**3289**	568	5.79
OLSR	ETX	0.5s	3107	389	7.98
OLSR	ETX	1s	2922	263	11.11
OLSR	ETX	2s	2569	**199**	12.90
AH-OLSR	ETX	Variable	3111	231	<u>13.47</u>
AH-OLSR	**AH-ETX**	Variable	<u>3236</u>	<u>229</u>	**14.13**

4.3.2 Control Message Overhead and Adaptability

The control message overhead comparison, depicted in Figure 5, provides valuable insights into the adaptive nature of AH-OLSR. Initially, when node mobility is high and the network topology is rapidly changing (10<t<30), AH-OLSR's overhead pattern resembles that of OLSR with a shorter Hello interval (e.g., 0.25s). This is because the adaptive protocol reduces the Hello interval to quickly respond to the frequent topology changes, maintaining robust connectivity and higher throughput.

As time progresses and nodes become relatively stable (t>30), the overhead pattern of AH-OLSR shifts, aligning more closely with OLSR configurations using longer Hello intervals (e.g., 1s or 2s). This adaptive behavior minimizes unnecessary control traffic when the network topology is stable, thereby reducing routing overhead and ensuring efficient communication. This dynamic auto-adjustment mechanism allows AHOLSR to maintain high efficiency by balancing the trade-off between rapid response to topology changes and minimizing control traffic. Furthermore, the dynamic intervals enable effective control message management, improving throughput under similar routing overhead conditions compared with static Hello interval settings.

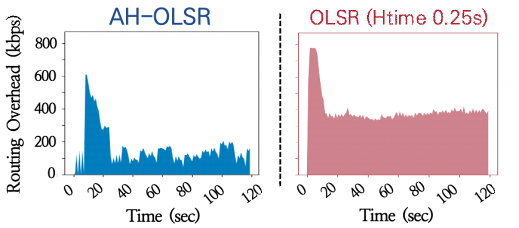

Fig. 5. Control message overhead comparison (AH-OLSR vs OLSR with static hello interval).

4.3.3 Network Efficiency in Dynamic Environments

Figure 6 illustrates the relationship between network efficiency and node mobility for OLSR, AH-OLSR, and AH-OLSR+AH-ETX. The experiment is conducted by changing only the maximum speed of node mobility to 0.5, 1, 2, 3, and 4 m/s from

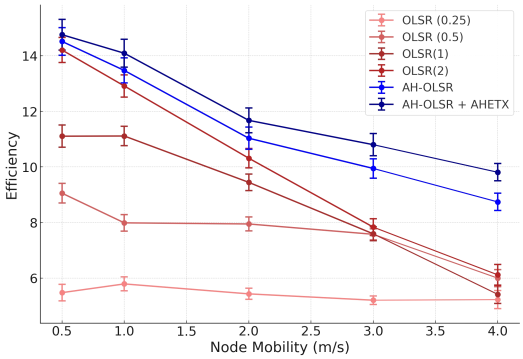

Fig. 6. Comparison of network efficiency with increasing node mobility for different routing protocols (OLSR, AHOLSR, and AH-OLSR+AH-ETX).

Table 2. As node mobility increases, the efficiency of all protocols decreases due to the challenges associated with maintaining stable routes in a rapidly changing topology. However, the rate of efficiency decline varies among the protocols.

OLSR shows a significant drop in efficiency as mobility increases, particularly with longer Hello intervals (e.g., 2s). This is likely due to the delayed route updates, which become less effective in highly dynamic environments. As the Hello interval increases, OLSR's initial efficiency is higher but suffers more under high mobility due to the slow adaptation to topology changes. The primary reason for this greater efficiency drop with longer Hello intervals is that while the routing overhead remains relatively constant, the throughput experiences a substantial decrease under high mobility conditions.

In contrast, AH-OLSR demonstrates better adaptability to increasing node mobility. Its ability to adjust Hello intervals dynamically based on network conditions allows it to maintain higher efficiency across different mobility levels. This adaptability ensures that AH-OLSR can quickly respond to topology changes when necessary while minimizing overhead during more stable periods. Although the overhead may slightly increase due to shorter Hello intervals under high mobility, the throughput does not decrease significantly, allowing AH-OLSR to maintain high efficiency.

The combination of AH-OLSR and AH-ETX further enhances performance. This approach leverages the adaptive Hello interval mechanism of AH-OLSR and the improved link quality estimation of AH-ETX, resulting in the highest efficiency across all mobility levels. Even as node mobility reaches 4.0 m/s, AHOLSR+AH-ETX maintains a significant efficiency advantage over OLSR, demonstrating the robustness and scalability of our proposed methods.

These results demonstrate that AH-OLSR and AHETX are well-suited for deployment in high-efficiency scenarios, such as swarm robotics operations in dynamic and resource-constrained environments. The adaptive nature of our approach allows it to maintain high performance across various network conditions, a crucial feature for robotic swarm MANETs.

Ⅴ. Discussion and Conclusion

We proposed Adaptive Hello OLSR (AH-OLSR) and Adaptive Hello Expected Transmission Count (AH-ETX), designed to enhance routing in robotic swarm Mobile Ad Hoc Networks (MANETs). These protocols address the unique challenges of dynamic, resource-constrained environments where node cooperation is crucial.

Our simulation results, tailored to robotic swarm MANET scenarios, demonstrate significant improvements. AH-OLSR and AH-ETX achieved up to a 143% increase in efficiency over OLSR. This improvement is particularly relevant for applications such as swarm robotics and emergency response teams, where efficient coordination among nodes is essential.

The dynamic adjustment of Hello intervals in AHOLSR allows efficient response to varying network conditions, while AH-ETX provides more accurate link quality measurements. These features are beneficial in the unpredictable and resource-constrained environments typical of robotic swarm MANETs.

While our simulations provide promising results for medium-sized networks, further investigation is needed to ensure the protocol’ s scalability in larger networks. The computational overhead from frequent Hello interval adjustments could become more significant in larger deployments. To address this, the protocol can be improved by applying more detailed and conservative criteria for mobility measurement and interval assignment. By refining the mobility metrics and using more precise thresholds for adjusting intervals, unnecessary adjustments can be minimized, thus reducing overhead without compromising adaptability. This approach would allow the protocol to better manage scalability while maintaining efficient performance. Hierarchical network structuring could also be explored in future work, where control message adjustments are confined to localized clusters, further reducing the overall overhead in larger-scale networks.

Additionally, while the simulation results demonstrate the effectiveness of AH-OLSR and AH-ETX, real-world testing is essential for further validation. Real-world robotic swarm networks may encounter challenges not fully captured in simulations, such as interference from physical obstacles, hardware limitations, and environmental factors that could affect communication performance. To address these challenges, we are planning field tests in real-world robotic swarm MANET scenarios. These tests will focus on evaluating the protocols’ performance under constraints like network interference, limited resources (processing power and battery life), and signal obstruction. These experiments will help identify practical issues and provide insights for refining AH-OLSR and AHETX to ensure they are robust and adaptable for realworld applications.

Despite these challenges, the significant efficiency improvements demonstrated by AH-OLSR and AHETX suggest their potential to enhance the performance of robotic swarm MANETs in various demanding applications. The adaptive nature of these protocols makes them well-suited for the diverse and changing conditions typical in robotic swarm MANETs, potentially inspiring new approaches to adaptive routing in other types of dynamic networks.

References

[1] G. A. Lewis, S. Simanta, M. Novakouski, et al., “Architecture patterns for mobile systems in resource-constrained environments,” in

MILCOM 2013, 2013.
(https://doi.org/10.1109/MILCOM.2013.121)

[2] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Netw.*, vol. 3, no. 3, pp. 325-349, 2005. (ISSN: 1570-8705)
(https://doi.org/10.1016/j.adhoc.2003.09.010)

[3] M. Li, K. Lu, H. Zhu, M. Chen, S. Mao, and B. Prabhakaran, "Robot swarm communication networks: Architectures, protocols, and applications," in *2008 Third Int. Conf. Commun. and Netw.* pp. 162-166, China, 2008.
(https://doi.org/10.1109/CHINACOM.2008.4684993)

[4] C. Perkins, E. Belding-Royer, and S. Das, "*Rfc3561: Ad hoc on-demand distance vector (aodv) routing*," RFC editor, 2003.
(https://doi.org/10.17487/RFC3561)

[5] F. Pereira, R. Correia, P. Pinho, S. I. Lopes, and N. B. Carvalho, "Challenges in resourceconstrained iot devices: Energy and communication as critical success factors for future iot deployment," *Sensors*, vol. 20, no. 22, p. 6420, 2020.
(https://doi.org/10.3390/s20226420)

[6] J. G. Kim, T. Gong, and S.-J. Lee, "The design, implementation, and evaluation of a user-level mobile ad hoc network routing with cots devices," *J. KIICE*, vol. 23, no. 7, pp. 845-851, 2019.
(https://doi.org/10.6109/jkiice.2019.23.7.845)

[7] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proc. IEEE Int. Multi Topic Conf., 2001. IEEE INMIC 2001. Technol. for the 21st Century.*, pp. 62-68, 2001.
(https://doi.org/10.1109/INMIC.2001.995315)

[8] J. Singh, G. Singh, D. Gupta, G. Muhammad, and A. Nauman, "Oci-olsr: An optimized control interval-optimized link state routing-based efficient routing mechanism for ad-hoc networks," *Processes*, vol. 11, no. 5, p. 1419, 2023.
(https://doi.org/10.3390/pr11051419)

[9] W. Wei, H. Wu, Y. He, and Q. Li, "A multi-objective optimized olsr routing protocol," *PLOS ONE*, vol. 19, no. 4, e0301842, 2024.
(https://doi.org/10.1371/journal.pone.0301842)

[10] M. Hiyama, S. Sakamoto, E. Kulla, M. Ikeda, S. Caballe, and L. Barolli, "Performance comparison of olsr protocol by experiments and simulations for different tc packet intervals," in *IEEE 2014 Eighth Int. Conf. Complex, Intell. and Software Intensive Syst.*, pp. 38-43, 2014.
(https://doi.org/10.1109/CISIS.2014.6)

[11] D. Johnson, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing/Kluwer Academic Publishers*, 1996.
(https://doi.org/10.1007/978-0-585-29603-6_5)

[12] N. Harrag, A. Refoufi, and A. Harrag, "Neighbor discovery using novel de-based adaptive hello messaging scheme improving olsr routing protocol performances," in *2017 6th ICSC*, pp. 308-312, 2017.
(https://doi.org/10.1109/ICoSC.2017.7958731)

[13] H. Badis and K. Al Agha, "Qolsr, qos routing for ad hoc wireless networks using olsr," *Eur. Trans. Telecommun.*, vol. 16, no. 5, pp. 427-442, 2005.
(https://doi.org/10.1002/ett.1067)

[14] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. 9th Annual Int. Conf. Mobile Computing and Netw.*, pp. 134-146, 2003.
(https://doi.org/10.1145/938985.939000)

[15] M. Benzaid, P. Minet, and K. Al Agha, "Integrating fast mobility in the olsr routing protocol," in *IEEE 4th Int. Wkshp. Mobile and Wireless Commun. Netw.*, pp. 217-221, 2002.
(https://doi.org/10.1109/MWCN.2002.1045725)

[16] Y. Mostafaei and S. Pashazadeh, "An improved olsr routing protocol for reducing packet loss ratio in ad-hoc networks," in *IEEE 2016 Eighth Int. Conf. IKT*, pp. 12-17, 2016.
(https://doi.org/10.1109/IKT.2016.7777778)

[17] S. Sharma, "P-olsr: Position-based optimized link state routing for mobile ad hoc networks,"

in *2009 IEEE 34th Conf. Local Computer Netw.*, pp. 237-240, 2009. (https://doi.org/10.1109/LCN.2009.5355100)

[18] S. Gangopadhyay and V. K. Jain, "A position-based modified olsr routing protocol for flying ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 9, pp. 12087-12098, 2023. (https://doi.org/10.1109/TVT.2023.3265704)

[19] H. Badis and K. Al Agha, "Qolsr, qos routing for ad hoc wireless networks using olsr," *Eur. Trans. Telecommun.*, vol. 16, no. 5, pp. 427-442, 2005. (https://doi.org/10.1002/ett.1067)

[20] F. De Rango, M. Fotino, and S. Marano, "Eeolsr: Energy efficient olsr routing protocol for mobile ad-hoc networks," in *MILCOM 2008*, pp. 1-7, 2008. (https://doi.org/10.1109/MILCOM.2008.4753611)

[21] T. H. Clausen and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*, RFC 3626, Oct. 2003. (https://doi.org/10.17487/RFC3626)

[22] E. M. H. Zahugi, A. M. Shabani, and T. Prasad, "Libot: Design of a low cost mobile robot for outdoor swarm robotics," in *2012 IEEE Int. Conf. Cyber Technol. in Automat., Control, and Intell. Syst. (CYBER)*, pp. 342-347, 2012. (https://doi.org/10.1109/CYBER.2012.6392577)

[23] M. A. Labrador, "Communication-assisted topology control of semi-autonomous robots," in *Proc. 2006 31st IEEE Conf. Local Computer Netw.*, pp. 563-564, 2006. (https://doi.org/10.1109/LCN.2006.322170)

[24] C.-F. Yang, C.-J. Ko, and B.-C. Wu, "A free space approach for extracting the equivalent dielectric constants of the walls in buildings," in *IEEE Ant. and Propag. Soc. Int. Symp. 1996 Digest*, vol. 2, pp. 1036-1039, 1996. (https://doi.org/10.1109/APS.1996.549773)

[25] N. B. Long and D.-S. Kim, "Efficient cooperative relaying selection scheme based on tdma for military tactical multi-hop wireless networks," in *MILCOM 2015*, pp. 1679-1684, 2015. (https://doi.org/10.1109/MILCOM.2015.7357686)

[26] S. Woo, H.-W. Park, H.-S. Lee, Y.-S. Yoo, and B.-G. Jung, "A power control-based mftdma resource allocation scheme for next generation military satellite communication systems," *J. KICS*, vol. 37, no. 11, pp. 1138-1147, 2012. (https://doi.org/10.7840/kics.2012.37c.11.1138)

[27] J. Stevens, M. Feeney, L. Tipton, J. Nilsson, M. Skold, and T. Ericson, "Scenario based analysis of dynamic tdma ad-hoc tactical battlefield networking," in *IEEE MILCOM 2003*, vol. 2, pp. 1332-1336, 2003. (https://doi.org/10.1109/MILCOM.2003.1290419)

Hyeongheon Cha

He received the BS (magna cum laude) degree in electrical engineering from Korea Advanced Institute of Science and Technology(KAIST). He is working toward a PhD in electrical engineering at KAIST. His research interests include mobile wireless networks, on-device AI, mobile computing, ubiquitous sensing, and applied machine learning.

Taeckyung Lee

He is a Ph.D. student at KAIST, working under the guidance of Prof. Sung-Ju Lee. His research focuses on mobile AI with machine learning adaptation and personalization, particularly on robust and reliable test-time adaptation without source or labeled data. He completed his B.S. in the School of Computing at KAIST, graduating Magna Cum Laude. He then earned his M.S. in the School of Electrical Engineering from KAIST, also under the supervision of Prof. Sung-Ju Lee.

Dong-Hwan Yoon

He received his B.S. degree in Electrical and Electronics Engineering from the University of Seoul in February 2003. He completed his M.S. in Electrical, Electronics, and Computer Engineering at the University of Seoul in February 2005. Since May 2007, he has been working at LIG Nex1. His main research interests include Wireless Communication Systems, Wireless Networks, and Deep Learning.

Eun-Jae Lim

He received his B.S. degreei n Information and Communication Engineering from Kyungnam University in February 2008. He completed his M.S. in Information and Communication Engineering at Kyungnam University in February 2010. He earned his Ph.D. in IT Convergence Engineering from Kyungnam University in February 2016. Since December 2015, he has been working at LIG Nex1. His main research interests include Wireless Communication Systems, Modem Design, and Communication Signal Processing.

Jung-Kyu Sun

He received his B.S. degree in Computer Engineering from Chonnam National University in February 2000. He completed his M.S. in Defense Convergence Engineering at Yonsei University in February 2019. Since June 2010, he has been working at LIG Nex1. His main research interests include Wireless Communication Systems and Communication Signal Processing.

Sung-Ju Lee

He earned his Ph.D. in Computer Science from the University of California, Los Angeles (UCLA) in 2000. He started his industry career at the HewlettPackard Company, serving as a Principal Research Scientist and Distinguished Mobility Architect. Subsequently, he was a Principal Member of Technical Staff at the CTO Office of Narus, Inc. In 2015, Dr. Lee transitioned to KAIST, where he holds the KAIST Endowed Chair Professorship. His research spans the area of mobile computing, wireless networking, mobile AI, network security, and human-computer interactions. Dr. Lee received the HP CEO Innovation Award in 2010 for his pivotal role in bringing innovative products to market. He has also been honored with the test-of-time paper award at ACM WiNTECH 2016, the best paper awards at IEEE ICDCS 2015 and ACM CSCW 2021, and the methods recognition award at ACM CSCW 2021. Additionally, he received the Technology Innovations Award from KAIST. Dr. Lee was the General Chair of ACM MobiCom 2014 and co-TPC Chair of IEEE INFOCOM 2016 and ACM Mobi-Com 2021. He is an IEEE Fellow and an ACM Distinguished Scientist.