

Dynamic Distributed TDMA Resource Allocation Algorithm for Robotic MANETs

Hyeongheon Cha^{*}, Taekyung Lee^{*}, Dong-Hwan Yoon^{**},
In-Jae Song^{**}, Jung-Kyu Sun^{**}, Sung-Ju Lee[°]

ABSTRACT

Robotic Mobile Ad-hoc Networks (MANETs) face significant challenges in efficiently allocating network resources due to frequent topology changes and varying communication demands. Traditional static TDMA schemes are inadequate for such highly dynamic environments, leading to inefficient resource usage and communication delays. To address these challenges, we propose D2TRA-RM, a dynamic and distributed TDMA resource allocation algorithm specifically designed for robotic MANETs. Through extensive simulations using NS-3, we demonstrate that D2TRA-RM improves network throughput by 198.7% and reduces packet delay by 59.5% compared to traditional static TDMA. Additionally, D2TRA-RM's meticulous design for Robotic MANETs outperforms a naïve Dynamic-Distributed TDMA scheme, offering a 125.7% increase in throughput and 68.7% reduction in delay. Although it generates a small number of MAC control messages for negotiation, D2TRA-RM provides substantial gains in network performance, making it a robust solution for real-time, resource-constrained applications in robotic MANET environments. Our results demonstrate that D2TRA-RM effectively handles dynamic traffic and resource reallocation, ensuring efficient communication in highly variable network conditions.

Key Words : Mobile ad hoc networks, resource allocation, robotic swarm

1. Introduction

Robotic Mobile Ad-hoc Networks (MANETs) have gained significant attention in recent years due to their potential in various mission-critical applications such as search and rescue, environmental monitoring, and military operations^[1] (Figure 1). These networks consist of autonomous robotic nodes that communicate with each other without relying on fixed infrastructure. The high mobility of robots leads to frequent changes in network topology, including dy-

namic merging and splitting of network partitions^[2,3]. This dynamic nature poses substantial challenges for efficient resource allocation and reliable communication.

However, the inherent characteristics of MANETs, particularly in robotic networks, pose numerous challenges for communication protocols. First, the high mobility of robots leads to frequent changes in network topology, which complicates the task of maintaining stable communication links. Additionally, the limited bandwidth and energy constraints of robotic

※ This work was supported by KRIT grant funded by the Korea Government(DAPA, KRIT-CT-22-006-03)

• 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, 중신회원

* Korea Advanced Institute of Science & Technology (KAIST), School of Electrical Engineering, taekyung@kaist.ac.kr

** LIG Nex1, donghwan.yoon@lignex1.com, 정회원; injae.song@lignex1.com, 정회원; jungkyu.sun@lignex1.com

논문번호 : 202410-264-C-RU, Received October 31, 2024; Revised December 31, 2024; Accepted January 7, 2025

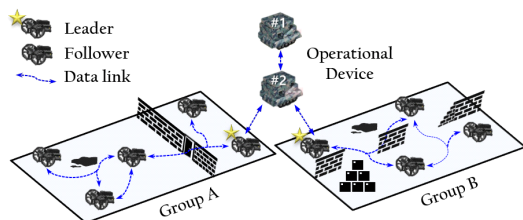


Fig. 1. Robotic Swarm MANET Scenario.

systems make it imperative to optimize the use of communication resources. The unpredictable nature of tasks in robotic networks further adds to the complexity, as communication requirements can vary significantly over time, requiring flexible and dynamic resource allocation mechanisms.

Time Division Multiple Access (TDMA) is a popular medium access control (MAC) protocol in MANETs due to its deterministic nature, where communication is scheduled in specific time slots, reducing the chances of packet collisions. However, traditional TDMA schemes suffer from inefficiency in highly dynamic environments like robotic MANETs^[4,5]. Static slot allocation in conventional TDMA can lead to underutilization of bandwidth, especially when nodes with low communication needs are assigned the same number of slots as more active nodes. Moreover, when network topology changes due to node mobility, the communication schedule must be frequently recalculated, leading to significant delays and overhead.

To address these issues, this paper proposes a **D2TRA-RM**: Dynamic and Distributed TDMA Resource Allocation for Robotic MANETs. Our approach introduces a flexible slot allocation mechanism that dynamically adjusts based on the communication demands and network topology changes. Each node in the network is capable of independently requesting additional slots or releasing unused ones, thus ensuring that bandwidth is efficiently allocated to nodes with higher communication demands. This dynamic allocation allows the network to adapt in real time to changes in both traffic and topology, improving overall network performance in terms of throughput, latency, and bandwidth utilization.

The proposed algorithm incorporates several key

innovations that distinguish it from traditional TDMA schemes. First, **dynamic slot allocation** enables nodes to request or release slots based on real-time communication needs, effectively reducing bandwidth waste and ensuring that high-priority tasks receive sufficient resources. Second, the algorithm operates in a **decentralized manner**, where each node makes slot allocation decisions independently, relying on local information without the need for global synchronization, thus reducing overhead. In addition, **priority handling for critical data** ensures that time-sensitive information is prioritized by preempting less critical communications, minimizing delays for high-priority messages. Finally, the algorithm is designed to **efficiently reallocate slots in dynamic topologies**, allowing nodes to adjust their slot assignments in response to frequent topology changes, thereby preventing communication interruptions caused by node mobility.

In this paper, we provide a comprehensive evaluation of the proposed D2TRA-RM algorithm through extensive simulations using the NS-3 network simulator. We compare the performance of our dynamic TDMA scheme with traditional static TDMA and a naïve Dynamic-Distributed TDMA scheme in various robotic network scenarios. Our results demonstrate that the proposed algorithm achieves a **198.7% improvement in network throughput** and a **59.5% reduction in packet delay** compared to traditional static TDMA. Compared to the naïve Dynamic-Distributed TDMA, D2TRA-RM provides a **125.7% increase in throughput** and **68.7% lower delay** while keeping the MAC overhead manageable. These results highlight the algorithm's ability to efficiently allocate resources in highly dynamic environments, significantly boosting overall network performance while maintaining minimal control overhead.

The remainder of this paper is organized as follows. Section II reviews related work and discusses their limitations in the context of robotic MANETs. Section III details the proposed dynamic distributed TDMA resource allocation algorithm. Section IV presents the simulation setup and performance evaluation results. Finally, Section V concludes the paper and suggests directions for future research.

II. Background

The development of resource allocation algorithms for MANETs has been an active area of research for several decades^[2,3,6,7]. These networks, which operate without fixed infrastructure, present significant challenges due to their dynamic nature, limited bandwidth, and the need for efficient communication protocols^[8]. In the context of robotic MANETs, these challenges are further exacerbated by the high mobility of the nodes, leading to frequent topology changes, network partitioning, and merging.

Early works on resource allocation for MANETs primarily focused on centralized algorithms, where a central controller would manage the allocation of communication resources among the nodes. Centralized TDMA algorithms^[9] were proposed to handle bandwidth constraints by dividing the available resources among nodes in a coordinated manner. However, these centralized approaches are unsuitable for robotic MANETs due to the inherent scalability issues and the single point of failure problem. In highly dynamic environments, such as those involving robotic nodes, centralized control can lead to delays in resource allocation and inefficient use of the available bandwidth, especially when the network is partitioned.

To address these limitations, decentralized and distributed algorithms have been developed, allowing nodes to coordinate resource allocation with their neighbors autonomously. Distributed TDMA-based protocols (e.g., DRAND^[10] and ADCA^[11]) enable nodes to negotiate time slots locally, reducing the need for a central controller. However, these protocols generally assume static or slowly changing network topologies, making them less effective in the highly mobile environments of robotic MANETs. In scenarios where the network topology changes frequently, these protocols may experience high control overhead and increased communication delays as nodes must constantly re-negotiate resources.

In response to the dynamic nature of robotic MANETs, several dynamic resource allocation algorithms have been proposed. For example, the dynamic slot assignment (DSA)^[12] algorithm adjusts time slots dynamically based on traffic conditions. Similarly,

mobility-tolerant TDMA protocol (M-TDMA)^[5] takes node mobility into account when scheduling time slots, reducing collisions and improving overall network performance.

Despite these advancements, current resource allocation algorithms for MANETs still face significant challenges when applied to robotic networks. The high mobility of robots, frequent changes in network topology, and the need for efficient partitioned operation have not been fully addressed. Moreover, existing solutions often fail to consider the dynamic and time-varying nature of communication demands in robotic networks, leading to suboptimal resource utilization and increased communication delays. The limitations of the existing resource allocation algorithm necessitate a decentralized approach tailored to robotic MANETs.

III. Design

The dynamic and distributed nature of robotic MANETs introduces unique challenges that traditional TDMA protocols often fail to accommodate, including rapid topology changes, varying traffic demands, and the need for distributed control in robotic networks. To overcome these limitations, we propose D2TRA-RM, a decentralized, demand-based resource allocation algorithm specifically designed for robotic MANETs. D2TRA-RM enables each node to independently coordinate resource allocation with its neighbors, ensuring efficient operation in both merged and partitioned network scenarios. By dynamically adjusting resources based on real-time communication demands, optimizing the TDMA frame structure, and minimizing overhead, D2TRA-RM effectively addresses mobility, network partitioning, and low-latency requirements while maximizing bandwidth efficiency.

3.1 Core System Architecture

This section describes the dedicated frame and slot structure required for the operation of D2TRA-RM, as well as the fundamental structure of its distributed dynamic resource allocation messages and slot state machine.

Slot Structure and TDMA Frame Design The slot structure, which serves as the basic unit of TDMA resource allocation in the proposed system, is designed to efficiently handle both control and data communication while ensuring accurate synchronization and gain control, drawing inspiration from prior studies^[13,14]. Each slot consists of three key fields: an *AGC field*, a *PRMB field*, and 8 *OFDM symbols*. The *AGC field* configures the receiver gain for the upcoming data in the slot, ensuring optimal signal reception. The *PRMB field* facilitates time and frequency synchronization, which is critical for maintaining coherent communication across the network. Additionally, there is a *mute time* between consecutive slots to enhance synchronization further and prevent interference.

The proposed TDMA frame structure divides the frame into **fixed slots** and **dynamic slots**, providing a flexible communication framework. While previous studies^[15-17] typically use a single, shared control period that focus on solely for control purposes, our approach employs per-node **fixed slots** that are evenly distributed across the frame. This ensures fair access to control resources and allows nodes to prioritize control messages, such as **HELLO** messages for neighbor discovery and network synchronization, in any slot they own. Additionally, fixed slots can also flexibly handle data transmissions when resources are available, supporting critical mission-specific data transmissions without prior negotiation. In parallel, **dynamic slots** are allocated to nodes based on real-time traffic demands via negotiation similar to prior dynamic TDMA works^[15-17]. This ensures efficient resource utilization and enables the system to adapt ef-

fectively to varying network conditions.

Furthermore, D2TRA-RM allows for a diverse ratio between fixed and dynamic slots. In scenarios with low traffic and a need for stability, more fixed slots can be allocated to ensure consistent control signaling. During periods of high traffic, dynamic slots are prioritized to maximize data throughput. Figure 2 illustrates the detailed structure of the proposed TDMA frame, including the slot composition and the mute time intervals between slots.

Periodic HELLO Messages For Neighbor Resource Information Sharing

D2TRA-RM employs periodic **HELLO** messages to ensure nodes maintain an up-to-date view of neighboring slot allocations. Each node broadcasts a **HELLO** message at regular intervals (typically every second) to all neighboring nodes. These messages contain a bitmap of the slots currently held by the node and the aggregated information about slot ownership from 2-hop neighbors. This allows each node to maintain a comprehensive view of resource usage across the local network (Figure 3, 4).

The periodic sharing of this information ensures that neighboring nodes are aware of available and occupied slots, preventing slot conflicts. Moreover, if discrepancies arise between a node's view of slot us-

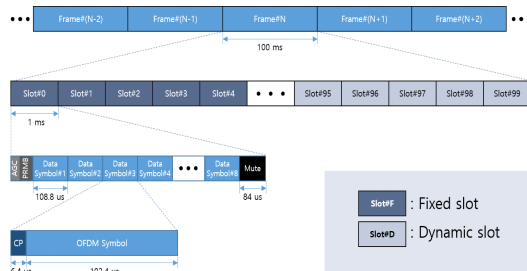


Fig. 2. TDMA frame structure with fixed and dynamic slots.

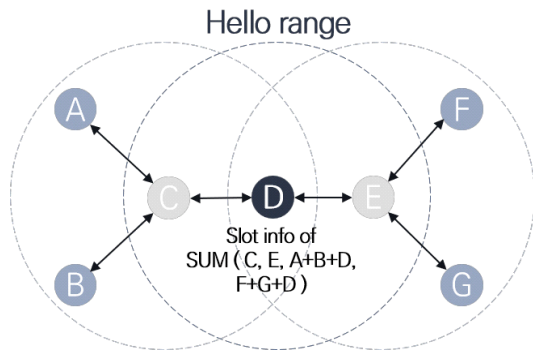


Fig. 3. Diagram of Hello message exchange between nodes.

1 byte																4*4 = 16 bytes															
Slot num	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Slot state	1	1	1	0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	
	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	
	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	1	0	0	0	0	0	0	
	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	
	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	
	97	98	99	100																											
	1	1	0	0																											

Fig. 4. Slot bitmap example (100 slots).

age and the **HELLO** messages it receives, the node can correct its slot allocation map, minimizing the chance of resource contention. While prior studies often rely on more detailed bitmaps to exchange 2-hop resource usage information^[15], our approach opts for a simpler binary representation to indicate slot availability within 1-hop. This design naturally accommodates 2-hop considerations based on the bitmaps of neighboring nodes, potentially reducing overhead. This streamlined **HELLO** message exchange helps maintain synchronized resource allocation across nodes within the communication range.

Slot-Specific State Machine Design To manage the dynamic allocation of slots, each node in D2TRA-RM operates a state machine for each individual slot. The state machine defines the current status of each slot and governs the transitions based on control messages received or sent. The primary states for each slot are as follows (Figure 5):

- **FREE**: The slot is unallocated and is available for use.
- **REQ_SENT**: The node has sent a request for the slot and is awaiting approval from neighbors.
- **GRANTED**: The slot has been granted by all relevant neighbors and is awaiting use.
- **ALLOCATED**: The slot is actively used by the node or has been allocated for future use.
- **RELEASED**: The slot has been released and is transitioning back to the **FREE** State.

Each slot's state machine allows D2TRA-RM to efficiently manage slot transitions, ensuring that each slot is either in use or readily available. By employing state machines for each slot, the system minimizes

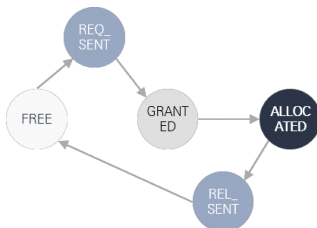


Fig. 5. Slot-specific state machine visualization.

underutilization and improves slot negotiation efficiency.

3.2 Resource Management Mechanisms

Dedicated Control Messages For Resource Allocation D2TRA-RM uses several dedicated control messages to facilitate the resource allocation process. These messages ensure efficient communication between nodes when negotiating, granting, or releasing slots. The main types of control messages used for resource allocation are shown in Table 1. These messages are designed to be lightweight, reducing the overhead of resource negotiation and enabling efficient communication even in highly dynamic networks. Each message is tailored to minimize unnecessary network traffic while ensuring reliable slot allocation.

Distributed Slot Negotiation Process The core of the algorithm is the distributed slot negotiation process, which allows nodes to autonomously request or

Table 1. Control messages in D2TRA-RM for resource allocation.

Message Type	Description
HELLO	Periodically broadcasts slot ownership and allocation status to neighboring nodes, enabling synchronization and conflict prevention.
REQUEST	Sent by a node to request additional slots when its current allocation is insufficient to meet traffic demands.
GRANT	Sent by neighboring nodes in response to a REQUEST message, granting the requesting node permission to use the specified slots.
REJECT	Sent when requested slots are unavailable, prompting the requesting node to retry or choose different slots.
RELEASE	Notifies neighbors that a slot is no longer needed and can be reallocated.
RELEASE_ACK	Acknowledges receipt of a RELEASE message, confirming that the slot has been freed.
COLLISION	Informs neighboring nodes of a slot conflict, ensuring that slot ownership discrepancies are resolved quickly.

release communication slots based on their current traffic needs. This decentralized approach avoids the need for a central coordinator and reduces the overhead typically associated with centralized resource allocation schemes.

When a node experiences increased traffic and needs more slots, it initiates a slot request procedure. The node broadcasts a **REQUEST** message to its neighboring nodes, specifying the number of slots it needs and providing its current traffic load and slot usage information. Neighboring nodes evaluate the request by checking their slot usage and the usage of their 2-hop neighbors. Neighbors respond with a **GRANT** message if the requested slots are available. Otherwise, they return a **REJECT** message, prompting the requesting node to retry with different slots or delay its request.

Algorithm 1 outlines this slot request procedure in detail.

Algorithm 1 Slot Request Procedure

```

1: Input: Current slot allocation, queue length
2: if queue length exceeds threshold then
3:   Broadcast REQUEST message to neighboring nodes
4:   Wait for responses from all neighbors
5:   if all neighbors return GRANT messages then
6:     Allocate requested slots
7:   else
8:     Retry with a new slot set or after a random
       delay
  
```

Similarly, when a node no longer needs its allocated slots, it releases them by sending its neighbors a **RELEASE** message. This ensures that the communication slots are efficiently utilized across the network and that idle slots are available to nodes with higher traffic demands. The distributed nature of this process allows each node to make slot allocation decisions independently without requiring global synchronization, which is crucial in mobile and decentralized environments like robotic MANETs.

Handling Topology Changes One of the key challenges in robotic MANETs is the frequent changes in network topology due to node mobility. The algo-

rithm addresses this issue by dynamically adapting to topology changes without requiring global coordination. Each node periodically broadcasts **HELLO** messages that contain a bitmap of the node's current slot usage. Neighboring nodes use this information to update their local views of the network, ensuring that slot allocations remain conflict-free. When a node detects that one or more neighbors are no longer reachable (indicated by the absence of **HELLO** messages over a certain period), it initiates a reallocation process to free up any slots previously shared with those neighbors. This process ensures that the network remains flexible and responsive to changes in connectivity, preventing communication breakdowns in highly mobile environments.

3.3 Dynamic Slot Allocation Algorithm Details

This section details the core mechanisms and design principles of the D2TRA-RM algorithm for resource allocation in highly dynamic robotic MANETs. By addressing key challenges such as traffic fluctuations, topology changes, and latency requirements, the algorithm ensures efficient and reliable communication.

3.3.1 Dynamic Slot Request and Release Strategy

In robotic MANETs, some nodes-particularly those serving as relay points or tasked with more complex operations-often experience higher traffic loads. These high-traffic scenarios occur when specific nodes handle a disproportionately high number of communications due to their role in the network topology or application requirements. This leads to two key challenges: (1) slot contention, where multiple nodes compete for the same communication slots, and (2) resource shortages, where high-traffic nodes fail to acquire sufficient slots to meet their communication demands. As a result, the network's overall efficiency is reduced, and packet loss becomes more frequent, particularly in dense or high-load environments. D2TRA-RM addresses these challenges through below complementary mechanisms.

Adaptive Slot Request Interval Each node *dynam-*

ically adjusts its slot request interval based on traffic-related metrics, such as transmission queue length, dropped packets, and the number of neighboring nodes. Specifically, nodes with longer transmission queues and more dropped packets shorten their request intervals, making them more likely to obtain additional slots and avoid congestion. This dynamic adjustment is designed to reduce resource shortages for heavily burdened nodes, directly addressing the disparity in resource distribution caused by traffic imbalances.

The interval is calculated as follows:

$$\text{Slot request interval} = \frac{10 \text{ sec} \times \# \text{ Neighbor nodes}}{\text{Tx queue length} + \text{Dropped packets}} \quad (1)$$

Equation 1 prioritizes nodes experiencing high communication demands, ensuring they receive additional slots more quickly. By integrating both traffic-related metrics (queue length and packet drops) and network density (number of neighbors), the model dynamically balances resource allocation among nodes. Nodes with longer queues need slots more urgently, so the interval between slot requests is shortened to ensure faster resource allocation. The dropped packets parameter reflects the number of packets lost due to insufficient slot allocation; nodes experiencing packet loss also decrease their request interval to prioritize recovering from the slot shortage. Finally, the number of neighboring nodes is included to avoid overwhelming the network with excessive slot requests in denser environments. Nodes with more neighbors slightly increase their request interval to balance the overall load on the network.

For example, in cases where a node has a long transmission queue and has dropped several packets, the slot request interval is reduced significantly, allowing the node to request slots more frequently. This ensures that nodes under high traffic load or suffering from packet drops are given priority when competing for available slots. On the other hand, nodes with shorter transmission queues and fewer dropped packets maintain a longer interval between slot requests, reflecting their lower priority for additional resources.

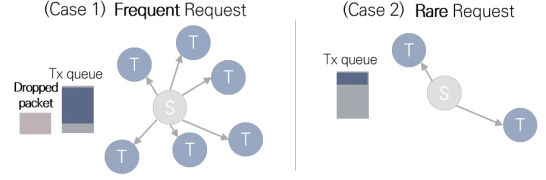


Fig. 6. Slot request interval adjustment based on traffic load and packet drops.

The following diagram illustrates two scenarios (Figure 6). In Case 1, a node with a long transmission queue and multiple dropped packets requests slots more frequently, resulting in higher priority for resource allocation. In Case 2, a node with fewer communication demands has a longer interval between requests, reducing its priority for additional slots. This adaptive slot request mechanism ensures that resource allocation is dynamically adjusted to match node-level traffic demands, reducing the impact of slot contention and resource shortages.

Prime Number-Based Slot Request To further reduce contention, D2TRA-RM employs a *prime number-based slot allocation* strategy. Each node is assigned a unique prime number and attempts to allocate slots based on multiples of that number. For example, a node with the prime number 13 will attempt to allocate slots 13, 26, 39, and so on. This staggered allocation method effectively spreads slot requests across the network, reducing the likelihood of collisions and ensuring efficient resource distribution.

Probability-Based Slot Release and Retention Traffic patterns fluctuate rapidly in highly dynamic robotic networks based on nodes' tasks and movements. Immediate slot release after traffic drops can lead to unnecessary reallocation cycles when traffic increases shortly after. In contrast, holding unused slots for too long reduces the available resources for other nodes.

D2TRA-RM addresses this problem with a *probability-based slot release* mechanism. Instead of releasing a slot once its utilization drops to zero, the system retains it with a probability of 50%. This approach balances the need to free up resources with the potential for short-term traffic fluctuations. By holding onto

slots that may be needed again shortly, D2TRA-RM reduces the overhead of frequent slot reallocations, optimizing overall network performance while minimizing unnecessary control traffic. Additionally, D2TRA-RM incorporates a *slot retention strategy* that ensures low-utilization slots are not immediately released. This strategy prevents excessive control overhead from slot release and reallocation processes, ensuring the system maintains a stable communication flow even during fluctuating traffic.

3.3.2 Relay Node Resource Management

In robotic MANETs, multi-hop (e.g., 5) communication is often required to reach nodes across large distances. Relay nodes along these paths can experience resource shortages, especially in high-traffic environments. Moreover, time-sensitive traffic, such as control commands or safety-critical data, requires guaranteed low-latency transmission, which is difficult to achieve under typical slot allocation processes.

To address resource shortages at relay nodes, D2TRA-RM implements a *slot-stealing mechanism*. When a relay node detects insufficient slots to forward packets, it sends a targeted resource allocation request to its predecessor node. This request is a dedicated control message called **REQUEST**, which contains the address of the relay node and the specific slots it wishes to steal from its predecessor. The predecessor node responds with a **GRANT** message for the requested slots, allowing the relay node to immediately use them for forwarding traffic.

This mechanism differs from the standard slot allocation process in D2TRA-RM, where slot requests are broadcast to all neighboring nodes, and the requesting node must receive a **GRANT** from all neighbors before using the slot. In the slot-stealing process, the relay node only communicates with its predecessor to expedite the resource transfer. This targeted approach significantly reduces the delay in securing additional slots, ensuring that relay nodes can quickly adapt to traffic surges.

For example, consider a scenario where Node 2, acting as a relay, receives packets from Node 1 (Figure 7). If Node 2's transmission queue contains eight slots worth of data from Node 1 but lacks the

necessary slots to transmit, it sends a **REQUEST** to Node 1, specifying four slots it wishes to steal (half of the required resources). Upon receiving the **GRANT** from Node 1, Node 2 immediately uses the stolen slots to forward the packets, ensuring smooth multi-hop communication without delay. By efficiently reallocating resources between relay nodes and their predecessors, the slot-stealing mechanism prevents resource bottlenecks and maintains the flow of multi-hop transmissions.

3.3.3 Slot Preemption for Critical Traffic

In time-sensitive applications on robotics MANET, such as control commands or safety-critical data transmission, D2TRA-RM implements a slot preemption algorithm to guarantee low-latency communication for specific high-priority traffic. This algorithm guarantees that critical packets, such as control commands or emergency data, can traverse a 5-hop path with a total delay of less than a certain bar (e.g., 50ms). When critical traffic such as control commands or emergency data is detected, the slot preemption algorithm is activated.

The slot preemption algorithm identifies preemptible slots along the 5-hop path from the source node S to the destination node T . The preemption interval (S_{int}) is set to 4 slots, ensuring that each hop introduces no more than a 4ms delay (D_{hop}). This design ensures that the critical packet is transmitted across the entire 5-hop path with a maximum end-to-end latency (L_{max}) of 50ms.

Once preemptible slots are identified at each node, any ongoing non-critical traffic using those slots is preempted, and the critical packet is allocated the pre-

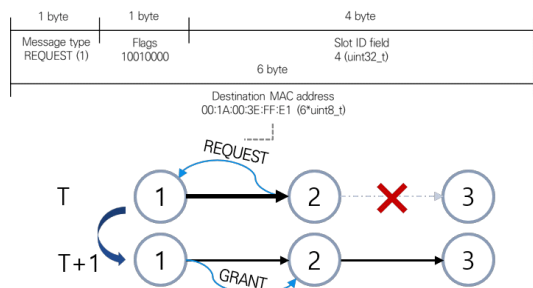


Fig. 7. Diagram of **REQUEST** message and slot-stealing mechanism for relay nodes.

Algorithm 2 Slot Preemption Algorithm for Critical Traffic

- 1: **Input:** Critical packet P , source node S , destination node T , 5-hop path (S, A, B, C, D, T)
 - 2: **Parameters:** $L_{\max} = 50\text{ms}$ (maximum end-to-end latency), $S_{\text{int}} = 4$ slots (preemption interval), $D_{\text{hop}} = 4\text{ms}$ (hop delay)
 - 3: **Initialize:** Preemptible slots at each node, spaced by S_{int} slots to ensure delay $\leq D_{\text{hop}}$ per hop.
 - 4: **for** each node $i \in S, A, B, C, D, T$ **do**
 - 5: Identify preemptible slots at node i .
 - 6: Preempt non-critical traffic in those slots.
 - 7: Allocate preempted slots to packet P for immediate transmission.
 - 8: Transmit P along the 5-hop path, ensuring hop delay $\leq D_{\text{hop}}$.
 - 9: Ensure total latency $\leq L_{\max}$.
-

empted slots for transmission. This guarantees that the critical traffic bypasses any lower-priority traffic and meets the strict latency requirements for time-sensitive applications.

This slot preemption mechanism is essential for ensuring that real-time communications, such as control commands or safety-critical data, are transmitted without delay, even under high network load. By enforcing the strict delay constraint, D2TRA-RM provides a reliable solution for mission-critical communication in robotic networks.

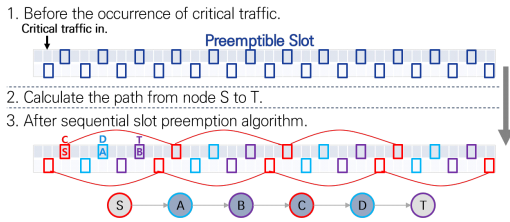


Fig. 8. Example of slot preemption for critical 5-hop traffic with less than pre-defined delay.

3.3.4 Control Overhead Reduction

In large-scale robotic MANETs, where many nodes compete for limited resources, the overhead from control messages used for slot allocation and release can become significant. Due to packet loss or network congestion, frequent communication failures further exacerbate this problem by causing repeated allocation attempts.

To reduce control overhead, D2TRA-RM implements a *probability-based forced slot allocation-release mechanism*. When a node fails to receive confirmation from all its neighbors after sending a slot request, it proceeds with the slot allocation with a 20% probability. This reduces the overhead from repeated allocation attempts, especially in high-traffic scenarios where control messages might be lost. If a slot conflict arises, the system uses collision resolution mechanisms to reassign slots, reducing the need for continuous control message exchanges and improving throughput.

3.3.5 Integrated Design Outcomes

The D2TRA-RM algorithm seamlessly integrates multiple adaptive strategies to address the challenges of resource allocation in robotic MANETs. By combining dynamic slot request intervals, prime number-based allocation, probability-based slot retention, and preemption mechanisms, the system ensures efficient resource utilization, low-latency communication for critical data, and reduced control overhead. These mechanisms collectively adapt to dynamic traffic demands, minimize slot contention, and support seamless multihop communication, enabling real-time, reliable performance in highly dynamic and large-scale networks.

IV. Performance Evaluation

In this section, we evaluate the performance of the proposed **D2TRA-RM** algorithm through simulations. The evaluation focuses on measuring the overall efficiency of the system under different network conditions, particularly in terms of solving the problems outlined in the previous section. The results demonstrate the effectiveness of D2TRA-RM in managing resources, reducing communication latency, and adapting to network topology changes in robotic MANETs.

4.1 Simulation Setup

The simulated network consists of 31 mobile nodes structured to emulate practical robotic MANET configurations such as those used in swarm robotics or

emergency response operations. The network includes one central operational device, five clusters of five nodes, each led by a leader, and five intermediary nodes positioned to facilitate communication between the clusters and the operational device. This layout mimics realworld scenarios involving collaborative robotic systems or coordinated response teams. The topology and node arrangement are illustrated in Figure 9.

The simulation runs for a total of 100 seconds and is divided into three key phases:

1. Deployment phase (0-10s): Nodes are initially deployed and move to predefined positions within the simulation area.
2. Communication initialization phase (10-20s): Task Coordination (TC), Task Management (TM), and data transmissions begin across the network.
3. Operational phase (20-100s): Nodes engage in random movement within assigned regions, with a maximum speed of 1 m/s. During this phase, every 20 seconds, a randomly selected node initiates a video upload to simulate high-bandwidth traffic.

The movement speed of 1 m/s was selected as it aligns with the typical mobility patterns observed in swarm robotics and related MANET simulations, en-

suring that the evaluation accurately reflects practical movement behaviors^[18,19].

The simulation scenario incorporates four distinct types of traffic, representative of the diverse communication needs found in typical robotic MANET applications:

- Task Coordination (TC): 30 kbps data stream from the operational device to all other nodes, facilitating the coordination of tasks.
- Task Management (TM): 30 kbps traffic from the leader nodes to the operational device, handling task-related communication.
- Video Traffic: High-bandwidth 1.5 Mbps video data sent from follower nodes to the operational device, simulating the upload of video feeds.
- Operational Data: 100 kbps of operational data transmitted from follower nodes to their respective leader nodes.

This variety in traffic types is intended to test the performance of D2TRA-RM under different communication loads, reflecting the typical demands of robotic networks, particularly in real-time task management and data-intensive applications.

The simulation takes place in a 100m x 100m environment, which includes physical obstacles modeled using a free space path loss model^[20]. These obstacles introduce an additional 10% attenuation, simulating walls or barriers commonly encountered in realworld operational settings. Each node has a communication range of 30 meters, sufficient to maintain contact with nearby nodes while accommodating potential gaps due to mobility.

The key parameters used in the simulation are summarized in Table 2. The selected parameters represent typical operational conditions for robotic MANETs in applications such as search-and-rescue missions, tactical deployments, and coordinated robotic operations. In addition, the routing protocol follows the basic RFC^[21] specification. D2TRA-RM is evaluated under these conditions using a dynamic TDMA protocol that allows deterministic slot assignment, ensuring collision-free communication in highly dynamic environments. This deterministic behavior is crucial for maintaining reliable and predictable communica-

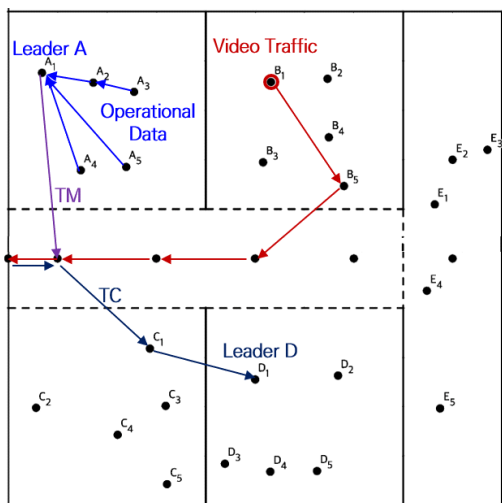


Fig. 9. Network topology and traffic patterns in the simulation scenario.

Table 2. Simulation parameters.

Parameter	Description
Network Simulator	NS-3
MAC Access Scheme	<i>D2TRA-RM</i>
Operation Channel	TX 1 channel, RX 1 channel
Transmission Mode	12 Mbps
Transmission Frame Structure	100 slots per frame, 1 ms per slot
Fixed:Dynamic Slot Ratio	Variable (e.g., 70:30, 60:40)
Routing Protocol	OLSR
Routing Metric	ETX
Simulation Area	100m $\times$ 100m square
Node Mobility	0~20s: Spread to random location 20~s ~ 100~s: Random direction, up to 1 m/s
Transmission Protocol	UDP
Obstacle Attenuation	10%

tion in mission-critical operations, where both throughput and low-latency delivery are essential for effective coordination.

4.2 Performance Metrics

To evaluate the performance of the D2TRA-RM system, we consider three key metrics: *Throughput (kbps)*, *MAC Overhead (kbps)*, and *Delay (msec)*. These metrics provide insights into how efficiently the system handles resource allocation, particularly under varying traffic demands and network conditions.

Throughput (kbps) Throughput is measured in kilo-bits per second (kbps) and reflects the total amount of data successfully transmitted across the network over time. Higher throughput indicates better network efficiency, which is critical in supporting high-bandwidth applications like video streaming or data-intensive sensor communications in robotic networks.

MAC Overhead (kbps) MAC overhead refers to the control message overhead introduced by the Medium Access Control (MAC) layer. It is measured in kilo-bits per second (kbps). Higher MAC overhead

suggests that more bandwidth is being used to manage slot allocation, synchronization, and other network operations, potentially reducing the bandwidth available for actual data transmission. Lower MAC overhead indicates more efficient resource management.

Delay (msec) Delay is the time required for a packet to travel from its source to its destination, measured in milliseconds (msec). Lower delay is essential for realtime applications in robotic systems, such as control commands or time-sensitive data transmissions. Minimizing delay ensures that critical data reaches its destination quickly, which is important for maintaining coordination between nodes in a robotic MANET.

These performance metrics provide a comprehensive understanding of the D2TRA-RM system's effectiveness in optimizing resource allocation, reducing communication delays, and managing network overhead, particularly in dynamic and decentralized environments.

4.3 Results and Analysis

The experimental results validate the effectiveness of the D2TRA-RM system across various network configurations and conditions. This section analyzes the results for each problem-solution pair, using throughput, MAC overhead, and delay as key performance indicators.

Table 3. Performance comparison: fixed vs dynamic slot allocation (various ratios).

Fixed:Dynamic	100:0	30:70	40:60	60:40
Throughput (kbps)	1008	1083	1177	1334
MAC Overhead (kbps)	0	188	172	<u>156</u>
Delay (msec)	672	973	941	<u>869</u>

4.3.1 Analysis on Proposed Dynamic TDMA Frame Structure

The first set of experiments compared network performance when using a fully fixed slot allocation (100:0) against a mixed fixed-dynamic slot allocation (60:40). Bold values indicate the best performance, while underlined values show the second-best. As shown in the results, the dynamic allocation provided a significant increase in throughput, from 1008 kbps

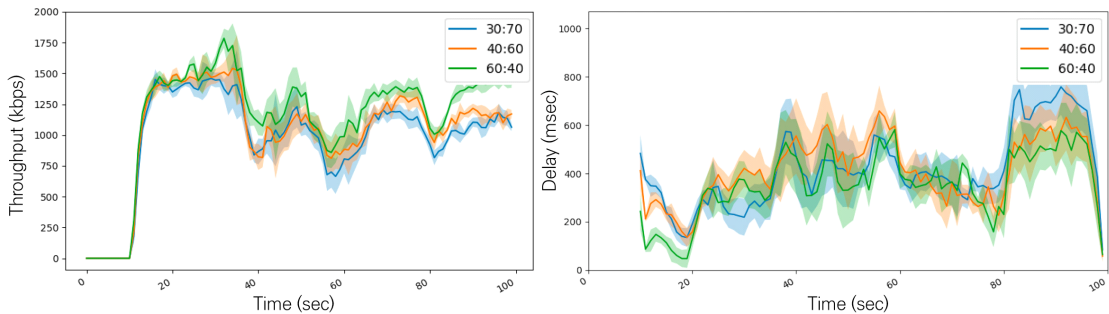


Fig. 10. Performance (throughput and delay) visualization for Various fixed-to-dynamic slot ratios. The translucent areas on the graph represent the variance when averaging over 10 random seeds.

to 1334 kbps, while introducing a moderate increase in MAC overhead (from 0 to 156 kbps) and delay (from 672 ms to 869 ms). These results illustrate the trade-offs between throughput and control overhead when dynamic slot allocation is used to better adapt to varying traffic demands. It is important to note that the reason for the lower delay observed in the fully fixed slot allocation (100:0) is that dropped packets are not included in the delay measurement. This means that, while the delay is lower, the lower throughput negatively impacts overall performance, as fewer packets are successfully transmitted. The lower throughput in the fully fixed slot allocation indicates that the system is less capable of handling varying traffic demands, leading to packet loss and lower overall performance.

The increased throughput observed in the dynamic scenario is primarily due to the system’s ability to allocate resources dynamically based on traffic requirements, preventing packet loss for critical applications such as video streaming. However, the increase in MAC overhead indicates the cost of managing this dynamic allocation process.

Proportional Slot Allocation for Different Traffic Loads Further experiments explored the effects of varying the fixed-to-dynamic slot ratio (e.g., 30:70, 40:60, and 60:40). As depicted in Figure 10 and Table 3, the 60:40 ratio achieved the best balance, with the highest throughput of 1334 kbps, while maintaining acceptable MAC overhead (156 kbps) and delay (869 ms). The 30:70 and 40:60 configurations also

performed well but with slightly lower throughput and higher overhead. These results show that adjusting the slot ratio based on network traffic can significantly improve performance by allocating more dynamic slots to handle peak loads without overly increasing overhead.

4.3.2 Ablation Studies on Algorithm Details
Adaptive Slot Request Intervals To address the problem of resource imbalance, the D2TRA-RM system adapts the slot request interval based on node traffic demands (Equation 1). Detailed evaluation results are in Table 4. When the interval is dynamically adjusted according to the node’s transmission queue length and dropped packets, throughput improved from 2429 kbps to 3001 kbps, while MAC overhead dropped from 665 kbps to 374 kbps. The delay remained relatively stable, demonstrating that adjusting the request interval leads to more efficient resource usage without significantly affecting latency. The results highlight that this mechanism resolves resource imbalances by ensuring nodes with higher traffic demands request slots more frequently, preventing delays and minimizing packet loss.

Probability-based Slot Allocation The system was

Table 4. Performance comparison: effect of adaptive slot request intervals on network performance

Metric	StaticInterval	AdaptiveInterval
Throughput (kbps)	2429	3001
MAC Overhead (kbps)	665	374
Delay (msec)	278	288

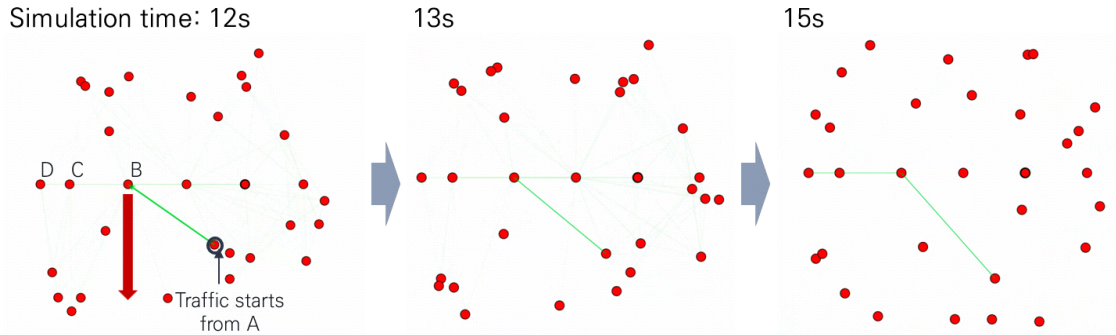


Fig. 11. Simulation of resource reallocation for relay nodes (time 12s to 15s). The green line represents packet throughput, and the red arrows indicate packet drops.

evaluated for its ability to allocate slots probabilistically when acknowledgments (ACKs) were not received from neighboring nodes. Specifically, we tested different slot allocation probabilities (1/3, 1/4, 1/5, and 1/6). The results demonstrated that the 1/5 probability yielded the best balance of throughput and delay, significantly improving network performance compared to the strict slot allocation scenario (0 probability).

The results showed that throughput improved from 1772 kbps to 2941 kbps with a 1/5 probability while keeping the delay to 273 ms. This mechanism is particularly advantageous in environments with high packet loss, as shown in the packet loss scenarios evaluated.

Probability-based Slot Release Another experiment tested the effectiveness of probabilistically retaining slots after their utilization became 0. Instead of releasing slots immediately, we introduced a mechanism that kept them with a certain probability (1/3, 1/2, or 1/3). This mechanism aimed to reduce unnecessary reallocation delays and slot overhead. The simulation result is in Table 6.

Table 5. Performance comparison: probability-based slot allocation.

Metric	0 (Strict)	1/3	1/4	1/5	1/6
Throughput (kbps)	1772	2765	2909	2941	2858
MAC Overhead (kbps)	415	196	210	218	214
Delay (msec)	364	198	253	273	284

The 50% probability setting yielded the best performance with throughput peaking at 3011 kbps, while both MAC overhead (274 kbps) and delay (272 ms) were minimized. This shows that retaining slots for a brief period, even after usage, optimizes resource utilization without causing significant overhead.

Relay Node Resource Shortage Handling The relay node resource shortage issue was examined by simulating a scenario where nodes needed to relay traffic but had insufficient slots. The system dynamically re-allocated slots by preempting resources from upstream nodes. The visualized simulation demonstrated the efficiency of this approach in ensuring seamless communication, even under high-load conditions. As in Figure 11, the resource preemption and redistribution algorithm successfully alleviated the resource bottleneck issue for multi-hop relay nodes. Details are the following: At 12 seconds, Node A generated video traffic, exhausting its resources and preventing Node B from forwarding packets to Node C. By 13 seconds, Node B preempted half of Node A's slots, resuming the relay. This process continued with Node C taking resources from Node B. By 15 seconds, the resource

Table 6. Performance comparison: slot release mechanism.

Metric	0 (Imm.)	1/3	1/2	2/3
Throughput (kbps)	2803	2890	3011	2947
MAC Overhead (kbps)	376	395	274	481
Delay (msec)	365	288	272	289

allocation stabilized, ensuring proper relay operation along the chain.

Slot Preemption for Critical Traffic In military and mission-critical scenarios where low-latency transmission is essential, the slot preemption algorithm is vital in ensuring that critical traffic is transmitted across a multi-hop path with an end-to-end latency of less than a certain value. We evaluated our algorithm on the scenario that consists of 5-hop with a 50ms latency bar. The evaluation shows that the maximum delay was reduced from 872 ms (without preemption) to 169 ms, with an average delay reduction from 365 ms to 64 ms (Table 7). This mechanism ensures that high-priority packets are given precedence over non-critical traffic, which is especially important in time-sensitive applications. As illustrated in Figure 12, the delay observed at around 20 seconds was caused by nodes sequentially receiving slot allocations. The red circle highlights this occurrence, showing how the preemption algorithm helped reduce the latency. After that circle, the lower delay is consistent for a while. This improvement confirms the preemption algorithm's role in ensuring low-latency communication, especially for critical data such as

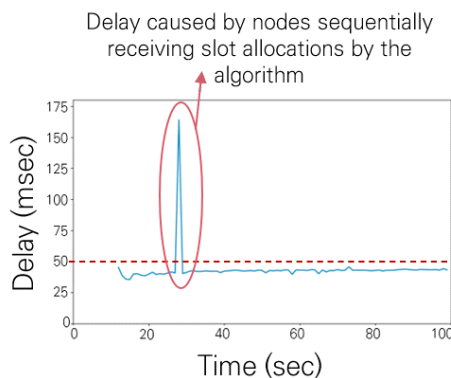


Fig. 12. The preemption algorithm ensures low-latency transmission for critical traffic.

Table 7. Performance impact of slot preemption for critical traffic

Metric (msec)	W/o Preemption	With Preemption
Maximum Delay	872	169
Average Delay	365	64

control commands.

4.3.3 Overall Performance of D2TRA-RM

The final experiment compared the performance of D2TRA-RM against two baseline schemes: a traditional TDMA (with all nodes equally allocated slots) and a naïve dynamic-distributed TDMA (with a 60:40 fixed-to-dynamic slot ratio). The following table summarizes the results (Table 8).

D2TRA-RM achieves significant performance gains over both TDMA and naïve Dynamic-Distributed TDMA schemes. In terms of throughput, D2TRA-RM delivers 3011 kbps, which represents a 198.7% improvement over TDMA (1008 kbps) and a 125.7% improvement over naïve Dynamic-Distributed TDMA (1334 kbps). These results highlight the system's ability to allocate resources dynamically, ensuring that high-demand nodes receive the necessary bandwidth.

Regarding delay, D2TRA-RM reduces latency by 59.5% compared to TDMA and by 68.7% compared to the naïve Dynamic-Distributed TDMA, ensuring that time-sensitive traffic is delivered more efficiently. While MAC overhead increases from 0 in the TDMA scheme to 274 kbps in D2TRA-RM, this is a small trade-off considering the substantial gains in throughput and latency.

These results demonstrate that D2TRA-RM handles dynamic traffic and resource reallocation more efficiently and far outperforms traditional and naïve dynamic TDMA approaches, making it a highly suitable solution for complex and dynamic robotic MANET environments.

Table 8. Overall performance comparison: D2TRA-RM vs baseline schemes (a traditional TDMA with all nodes equally allocated slots and a naïve dynamic-distributed TDMA with a 60:40 fixed-to-dynamic slot ratio).

Metric	Naïve		
	TDMA	Dynamic-Distributed TDMA	D2TRA-RM
Throughput (kbps)	1008	1334	3011
MAC Overhead (kbps)	0	156	274
Delay (msec)	672	869	272

V. Discussion and Conclusion

The proposed **D2TRA-RM** algorithm effectively addresses key challenges in robotic MANETs, such as dynamic topology changes, varying traffic demands, and decentralized control. Despite certain limitations, D2TRA-RM provides a robust framework for efficient resource allocation in these networks. In this section, we discuss the algorithm's overall effectiveness, its limitations, potential future improvements, and possible applications.

Effectiveness of D2TRA-RM D2TRA-RM consistently demonstrates high slot utilization and adaptability, ensuring efficient bandwidth usage even in dynamic environments. The probabilistic delay mechanism successfully reduces slot conflicts in high-mobility scenarios, allowing nodes to manage communication resources without centralized control. The priority handling system ensures that critical traffic, like control commands, experiences minimal delay, which is essential for real-time applications in robotic operations.

Moreover, the algorithm's low control overhead allows more bandwidth for data transmission, making it suitable for bandwidth-constrained environments. This efficiency is vital in scenarios such as emergency response or tactical operations, where reliable communication is critical.

Limitations D2TRA-RM demonstrates robust performance; however, certain challenges remain. In extremely dense networks, decentralized control may occasionally result in slot conflicts when multiple nodes simultaneously request the same slots, despite the random backoff mechanism. Furthermore, the algorithm's reliance on periodic **HELLO** messages for topology updates may introduce brief inefficiencies during rapid topology changes or high mobility scenarios. Lastly, while the fixed TDMA frame structure effectively balances stability and adaptability, it may limit the system's responsiveness under highly variable or bursty traffic conditions.

Future Work To further enhance D2TRA-RM, future

research could explore predictive models or machine learning techniques to anticipate and proactively resolve slot conflicts in dense network environments. For instance, D2TRA-RM could adapt work on machine learning and conflict prediction suggests that appropriate machine learning methodologies can offer substantial improvements in accuracy and performance^[22]. Refining the **HELLO** message mechanism by adopting event-driven or context-aware updates could improve the system's adaptability to rapid topology changes. Existing dynamic hello messaging scheme for neighbor discovery in on-demand MANET routing protocols indicates that making hello intervals proportional to event intervals can reduce unnecessary hello messages, thereby enhancing energy efficiency^[23]. Additionally, incorporating a hybrid TDMA protocol that combines fixed slots for routine tasks with flexible slots for dynamic traffic could enhance scalability and responsiveness under variable traffic loads. The development of such hybrid protocols has been discussed in the context of ad hoc networks, aiming to achieve high channel utilization under varying contention levels^[24]. These improvements would strengthen D2TRA-RM's applicability in mission-critical scenarios such as urban search and rescue or military operations requiring real-time, robust communication.

Potential Applications D2TRA-RM has immediate applications in areas such as disaster response, where autonomous robots need reliable communication to coordinate tasks in real time. Its ability to prioritize critical traffic makes it suitable for environments where timely data transmission is essential, such as in locating survivors or assessing hazards.

In military operations, D2TRA-RM's decentralized approach and low-latency performance are advantageous for tactical deployments, especially in hostile or infrastructure-limited environments. The algorithm can also be applied in industrial automation, where coordination between autonomous systems requires both efficiency and reliability, ensuring that critical commands are executed without delay.

Conclusion In conclusion, D2TRA-RM provides a flexible and efficient solution for resource allocation

in robotic MANETs. It handles the complexities of dynamic environments through adaptive slot allocation, effective prioritization of traffic, and low control overhead. While there are areas for improvement, the overall performance of D2TRA-RM makes it well-suited for real-world applications. Future enhancements, such as improved conflict resolution and more sophisticated mobility models, will further strengthen its role in advanced robotic communication systems.

References

- [1] B. Roh, M.-H. Han, M. Hoh, K. Kim, and B.-H. Roh, "Tactical manet architecture for unmanned autonomous maneuver network," in *MILCOM 2016*, pp. 829-834, 2016.
(<https://doi.org/10.1109/MILCOM.2016.7795432>)
- [2] G. A. Lewis, S. Simanta, M. Novakouski, et al., "Architecture patterns for mobile systems in resource-constrained environments," in *MILCOM 2013, IEEE*, 2013.
(<https://doi.org/10.1109/MILCOM.2013.121>)
- [3] 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>)
- [4] K. Amouris, "Position-based broadcast tdma scheduling for mobile ad-hoc networks (manets) with advantaged nodes," in *MILCOM 2005*, vol. 1, pp. 252-257, 2005.
(<https://doi.org/10.1109/MILCOM.2005.1605694>)
- [5] A. Jhumka and S. Kulkarni, "On the design of mobility-tolerant tdma-based media access control (mac) protocol for mobile sensor networks," in *Distributed Comput. and Internet Technol.*, T. Janowski and H. Mohanty, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 42-53, 2007, ISBN: 978-3-540-77115-9.
(https://doi.org/10.1007/978-3-540-77115-9_4)
- [6] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Commun.*, vol. 6, no. 2, pp. 46-55, 2001.
(<https://doi.org/10.1109/98.760423>)
- [7] S. Corson and J. Macker, *RFC2501: Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations*, 1999.
(<https://doi.org/10.17487/RFC2501>)
- [8] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. and Mobile Computing*, vol. 2, no. 5, pp. 483-502, 2002.
(<https://doi.org/10.1002/wcm.72>)
- [9] E. Hossain and V. Bhargava, "A centralized tdma-based scheme for fair bandwidth allocation in wireless ip networks," *IEEE J. Sel. Areas in Commun.*, vol. 19, no. 11, pp. 2201-2214, 2001.
(<https://doi.org/10.1109/49.963806>)
- [10] I. Rhee, A. Warriar, J. Min, and L. Xu, "Drand: Distributed randomized tdma scheduling for wireless ad-hoc networks," in *Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. and Comput.*, ser. Mobi-Hoc '06, pp. 190-201, Florence, Italy, 2006, ISBN: 1595933689.
(<https://doi.org/10.1145/1132905.1132927>)
- [11] C.-M. Wu and Y.-Y. Wang, "Adaptive distributed channel assignment in wireless mesh networks," *Wireless Personal Commun.*, vol. 47, pp. 363-382, 2008.
(<https://doi.org/10.1007/s11277-008-9484-3>)
- [12] F. Shad, T. Todd, V. Kezys, and J. Litva, "Dynamic slot allocation (dsa) in indoor sdma/tdma using a smart antenna basestation," *IEEE/ACM Trans. Netw.*, vol. 9, no. 1, pp. 69-81, 2001.
(<https://doi.org/10.1109/90.909025>)
- [13] H. Lee, "Design and implementation of real time agc for satellite tdma communication systems," *J. KICS*, pp. 298-304, 2013.
[Online] Available: <https://api.semanticscholar.org/CorpusID:110642505>.
- [14] D.-K. Ko and W.-S. Yoon, "A robust tdma

- frame structure and initial synchronization in satellite communication,” *J. KIICE*, vol. 16, no. 8, pp. 1631-1641, 2012.
(<https://doi.org/10.6109/jkiice.2012.16.8.1631>)
- [15] J. Lee, “Dynamic slot allocation scheme in tactical multi-hop networks for future soldier systems,” *J. KIMST*, vol. 24, no. 1, pp. 115-122, 2021.
(<https://doi.org/10.9766/KIMST.2021.24.1.115>)
- [16] J.-K. Lee, “Performance analysis of dynamic tdma and fixed tdma in tactical data link,” *J. KIMST*, vol. 21, no. 4, pp. 489-496, 2018.
(<https://doi.org/10.9766/j.2018.21.4.489>)
- [17] T. Yin, Y. Wang, M. Zhao, and J. Xiao, “A modified dynamic tdma slot allocation algorithm in ad hoc network,” in *2016 First IEEE ICCCI*, pp. 124-128, 2016.
(<https://doi.org/10.1109/CCCI.2016.7778891>)
- [18] 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. Automat., Control, and Intell Syst. (CYBER)*, pp. 342-347, 2012.
(<https://doi.org/10.1109/CYBER.2012.6392577>)
- [19] 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>)
- [20] 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 Antennas and Propag. Soc. Int. Symp. 1996 Digest*, vol. 2, pp. 1036-1039, 1996.
(<https://doi.org/10.1109/APS.1996.549773>)
- [21] T. H. Clausen and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*, RFC 3626, Oct. 2003.
(<https://doi.org/10.17487/RFC3626>)
- [22] C. Perry, “Machine learning and conflict prediction: A use case,” *Stability: Int. J. Security and Development*, vol. 2, no. 3, p. 56,

2013.

(<https://doi.org/10.5334/sta.cr>)

- [23] A. Nadda, “An dynamic hello messaging scheme for reducing energy consumption in on-demand manet routing protocols,” *Int. J. Eng. Trends and Technol.*, vol. 19, no. 5, pp. 247-252, 2015.
(<https://doi.org/10.14445/22315381/IJETT-V19P247>)
- [24] H. You and J. J. Garcia-Luna-Aceves, “La-mac: A load adaptive mac protocol for manets,” in *2009 IEEE Global Telecommun. Conf.*, pp. 1-6, 2009.
(<https://doi.org/10.1109/GLOCOM.2009.5425933>)

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.

Taeckyoung 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



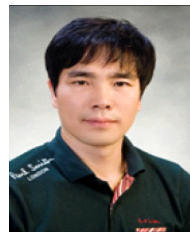
He received his B.S. degree in Electrical and Electronics Engineering from the University of Seoul in February 2003. He completed his M.S. in Electrical, Electronics, and Computer Engineering at the University of Seoul in February 2005. Since May 2007, he has been working at LIG Nex1. His main research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

In-Jae Song



He received his B.S. degree in Electrical, Electronics, and Computer Engineering from Inha University in February 2005. He completed his M.S. in Electrical and Electronics at Korea Advanced Institute of Science and Technology(KAIST) in August 2024. Since April 2013, he has been working at LIG Nex1. His main research interests include Wireless Communication Systems, Wireless Networking, and Routing Protocol.

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 Hewlett-Packard 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.