Discovery Storm: Scalability Analysis of DDS and Zenoh in Large-Scale Wireless Robotic Networks

Sanghoon Lee, Yeonwoo Choi, Jiyeong Chae, and Kyung-Joon Park
Department of Electrical Engineering and Computer Science, DGIST, Daegu, Republic of Korea
Email: {leesh2913, cyw040306, cowldud3, kjp}@dgist.ac.kr

IEEE INFOCOM 2026 - IEEE Conference on Computer Communications | 979-8-3315-4961-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/INFOCOM59046.2026.11571268

***Abstract*—While robotic swarm research envisions scaling to hundreds of agents, practical deployments frequently suffer from communication failures even with a small number of interconnected robots. A primary barrier hindering this scalability is the discovery protocol within the robot middleware. Although Robot Operating System 2 (ROS 2) has adopted Data Distribution Service (DDS) as its standard middleware, its default discovery mechanism, which relies on periodic multicasting, was originally designed for wired, static networks. In large-scale wireless environments, this mismatch triggers a "Discovery Storm," where the exponential growth of discovery traffic saturates limited wireless bandwidth. In this paper, we establish a deterministic mathematical framework to identify the theoretical scalability bounds of DDS and Zenoh, serving as a guideline for wireless swarm robotic deployment. By analytically modeling the system-wide required discovery traffic and airtime, we demonstrate that current P2P discovery mechanisms inevitably lead to network meltdowns in swarm scenarios operating over standard IEEE 802.11 Wi-Fi environments.**

***Index Terms*—Swarm Robotics, ROS 2, Discovery Protocol, Wireless Networks, DDS, Zenoh**

I. Introduction

The paradigm of robotics research is rapidly shifting from operating isolated single agents to deploying large-scale swarm systems comprising tens to hundreds of robots. These systems hold immense potential for revolutionizing industries, ranging from smart logistics and smart cities to defense operations. The fundamental prerequisite for realizing such swarm intelligence is seamless data sharing and collaboration among robots [1]. Since agents must exchange state estimates, sensor data, and task allocations in real-time, robust and reliable network connectivity is essential. However, communication failures are frequently observed in practical deployments, even with only a small number of interconnected robots. This instability poses a critical bottleneck in scaling up robotic systems.

To support distributed modular robotic systems, ROS 2, the de facto standard for robot software, has adopted the Data Distribution Service (DDS) as its standard middleware layer [2]. At its core, DDS relies on data centric publish-subscribe communication facilitated by a brokerless discovery protocol. These mechanisms operate on a fully distributed Peer-to-Peer (P2P) basis, enabling agents to automatically discover one another without a central server. While this brokerless design offers significant advantages in terms of "zero-configuration" convenience and flexibility for ad-hoc deployments, it imposes a fatal scalability cost. The requirement for every participant to continuously broadcast and process discovery messages creates an inherent trade-off, demanding excessive network resources as the system scales.

The root cause of this scalability issue is the mismatch between the design philosophy of DDS and the operational reality of robotic swarms. DDS and its underlying Real-Time Publish-Subscribe (RTPS) protocol were designed for wired Ethernet networks in aerospace and defense sectors [3]. While UDP/IP multicast is efficient in such bandwidth-rich environments, it becomes a vulnerability in IEEE 802.11 (Wi-Fi) networks. In Wi-Fi, multicast frames are transmitted at the lowest basic rate without MAC-layer ACKs, resulting in high airtime consumption and poor reliability. Moreover, the massive volume of discovery packets from every host saturates the wireless channel. Consequently, as the number of hosts increases, the network faces a "Discovery Storm," where connectivity collapses before any application data is exchanged.

To address these challenges, DDS vendors have introduced centralized discovery approaches. However, these approaches introduce a single point of failure (SPOF) and impose significant configuration complexity in dynamic ad-hoc swarm environments, making them incomplete solutions. Recently, Zenoh has emerged as a promising lightweight middleware alternative [4]; yet, quantitative analysis regarding its scalability limits in pure P2P mode remains insufficient. Prior works have predominantly focused on modeling [5], [6] and optimizing throughput [7] during application data exchange. Consequently, there is a distinct lack of analysis that targets the discovery traffic itself, particularly in terms of MAC/PHY-layer overheads.

To bridge this gap, this paper presents an analytical study quantifying the scalability limits of robot middleware discovery protocols in wireless robotic swarms. This work makes the following contributions: First, we formulate the discovery traffic load of both DDS and Zenoh as a function of three key variables: the number of hosts (H), processes (P), and endpoints (E). Second, we derive a system-wide airtime model that incorporates the specific characteristics of the IEEE 802.11 PHY and MAC layers. Third, through numerical analysis, we demonstrate the theoretical upper bounds of current P2P mechanisms and discuss potential mitigation strategies for enabling scalable, decentralized discovery.

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00442085, Development of V2X Infra Security Core Technologies for Autonomous Vehicle Services).

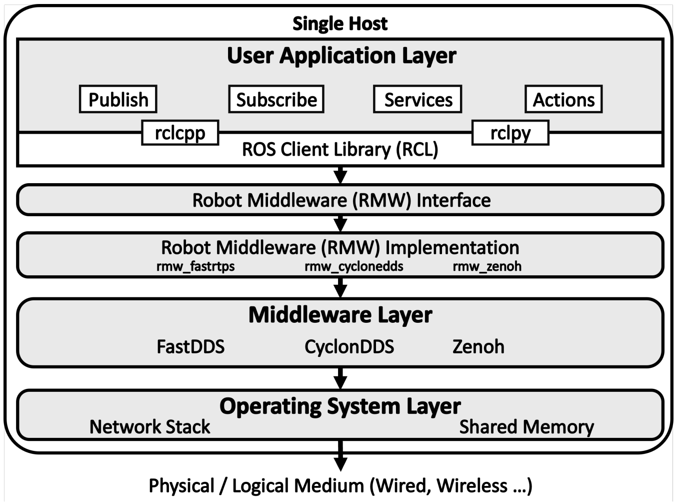

Fig. 1. Layered Architecture of ROS 2 System

II. Background

A. ROS 2 Architecture and Robot Middleware

ROS 2 is the de facto standard middleware framework for robotics. It provides a comprehensive set of libraries and tools that simplify the integration of complex systems. To accommodate diverse hardware configurations and meet the requirements of distributed robotic systems, ROS 2 adopts a robust hierarchical architecture. Fig. 1 illustrates the layered structure of a ROS 2 system within a single host.

Located at the top level, the user application layer implements the robotic logic by invoking high-level communication functions such as Publish, Subscribe, Services, and Actions via the ROS Client Library, including rclcpp and rclpy. These invocations then interact with the Robot Middleware (RMW) interface, which routes them to a specific RMW implementation. This implementation layer acts as an adapter that maps abstract commands to middleware-specific APIs and handles data serialization. Beneath this layer resides the middleware, such as Fast DDS, Cyclone DDS, or Zenoh, which functions as the actual network engine. The middleware is responsible for executing discovery protocols, enforcing QoS policies, and transporting byte streams using protocols like UDP, TCP, or SHM. Finally, the operating system manages the actual transmission over the physical or logical medium, ranging from wired and wireless interfaces to shared memory segments.

The primary advantage of this design is the strict abstraction of network complexity. Users can optimize network performance solely through environment variables and QoS configurations without modifying the source code, as the application layer remains agnostic to the underlying protocols. Consequently, the wireless traffic characteristics and scalability limits of the system are directly determined by the architectural behavior of the middleware. Therefore, this paper specifically analyzes the mechanisms within the middleware layer, focusing on the discovery protocol, which constitutes the critical bottleneck in wireless robotic swarms.

TABLE I
Hierarchical Entities in Discovery Models

Level	Symbol	DDS Entity (RTPS)	Zenoh Entity	Key Identifier
Host	H	Network Node (Host)	Network Node (Host)	IP Address / MAC
Process	P	DomainParticipant	Session (Runtime)	GUID / Zenoh ID (ZID)
Endpoint	E	DataWriter / DataReader	Publisher / Subscriber	Entity ID / Key Expression

B. Core Components of Discovery

The discovery procedure operates in a hierarchical manner. To conduct a scalability analysis, we first define the core components participating in the network. We categorize these entities into a three-tier hierarchy modeled as follows:

- **Host** (H)**:** A distinct computing unit equipped with a unique Network Interface Card (NIC) and IP address. Within a shared wireless channel, all hosts contend for the shared wireless medium, specifically the airtime. While a host is conceptually analogous to a single robot, modern robotic systems increasingly employ multi-computer architectures. Therefore, a single physical robot may consist of a cluster of multiple hosts, forming an intra-network.
- **Process** (P)**:** The fundamental execution unit within the ROS 2 framework, specifically corresponding to a ROS 2 Context. This entity serves as the primary communication anchor in the middleware layer. Functionally, modules for perception, planning, and control are often segregated into distinct processes to ensure fault isolation and maximize resource reusability. In the middleware domain, this entity maps to a DDS DomainParticipant or a Zenoh Session. Although processes within the same host share the IP and NIC, each process is distinguished by a middleware instance ID and explicitly occupies distinct OS-level resources, such as unicast ports and sockets.
- **Endpoint** (E)**:** A high-level communication object invoked by the user application layer. In ROS 2, examples include Publishers, Subscribers, Service Servers, and Clients. It is typically encapsulated within a parent Process. These entities map to DataWriters and DataReaders in DDS, or Resources in Zenoh. They act as the primary subjects of data production and consumption, utilizing the transport resources of their parent process. Each endpoint is assigned a unique identifier within the middleware, forms logical connections with peer endpoints based on matching Topic Names and Types, and operates under independent Quality of Service policies.

Table I summarizes the mapping between logical ROS 2 entities and their corresponding middleware abstractions. Discovery is defined as the procedure by which these components recognize one another. The connectivity between Hosts is managed by lower-layer protocols within the OS network stack, such as ARP or NDP. The middleware is responsible for the discovery of Processes (P) and Endpoints (E). It operates sequentially: the middleware first establishes connectivity between distinct processes, and subsequently exchanges information regarding the endpoints to logically match Publishers with Subscribers.

III. Scalability Analysis of Discovery Traffic

A. System Model and Assumptions

In this subsection, we define the key variables and assumptions for modeling discovery scalability. Let N_H, N_P, and N_E denote the number of hosts, processes per host, and endpoints per process, respectively. We assume a one-to-one robot-to-host mapping; thus, N_H represents the total number of robots. The system is considered homogeneous, where every host runs N_P processes. Each process maintains N_E endpoints, which are divided equally into $N_E/2$ publishers and $N_E/2$ subscribers for analytical symmetry.

The timing of discovery initiation is a critical factor influencing the instantaneous traffic load. We assume a synchronized startup scenario, where all entities in the system initiate the discovery process almost simultaneously. This assumption reflects common swarm robot operation patterns, such as a simultaneous power-on or a coordinated start command. Under these definitions and assumptions, we analyze the discovery traffic with respect to the variables N_H, N_P, and N_E.

To mitigate the scalability limitations of P2P discovery, middleware vendors offer centralized or hierarchical alternatives such as the Fast DDS Discovery Server or Zenoh Client modes. However, these architectures introduce potential single points of failure (SPOF) and typically require prior knowledge of server locators, such as static IPs or DNS names. Such dependency increases configuration complexity and reliance on fixed infrastructure, rendering them less suitable for dynamic ad-hoc environments where network topology changes frequently. Consequently, this study focuses its scalability analysis on the infrastructure-less, fully distributed discovery mechanisms: the standard Simple Participant Discovery Protocol (SPDP) and Simple Endpoint Discovery Protocol (SEDP) in DDS, and the peer mode in Zenoh.

The default configuration for rmw_zenoh is to run at least one Zenoh router per host as a dedicated separate process. This architecture enables key aggregation, which significantly reduces network overhead. However, this approach is vulnerable to a SPOF at the router node. To ensure a fair comparison with DDS, this study evaluates a ‘pure peer-to-peer’ mode to model the worst-case scenario where all nodes function as peers. Consequently, the scouting method is transitioned from the default configuration to multicast Scouting.

B. Analysis of DDS Discovery Traffic

The discovery process in DDS is governed by the RTPS protocol, which operates in two distinct phases: SPDP and SEDP. To rigorously quantify the impact of these protocols on the wireless medium, we model the discovery overhead as the traffic load L, defined as the number of logical messages injected into the network per discovery interval. The actual airtime consumption $\mathcal{T}$, representing the physical medium occupancy, is then expressed as a function of the uplink L^{up} and downlink L^{down} loads which are defined with respect to the wireless access point (AP).

The first phase, SPDP, is responsible for advertising the presence of DomainParticipants. Each process P periodically multicasts a single discovery message. From the perspective of a single host, the generated traffic depends solely on the number of active local processes (N_P). Thus, the host-level SPDP load $L_{\text{SPDP}}^{\text{host}}$ is:

$$L_{\text{SPDP}}^{\text{host}} = N_P \tag{1}$$

In an IEEE 802.11 infrastructure mode, the Access Point (AP) relays multicast traffic. A message transmitted by a host as an Uplink is buffered and re-transmitted by the AP as a Downlink to all associated hosts. Crucially, in standard Wi-Fi multicast, the AP transmits the frame only once at a basic data rate, allowing all associated clients to receive it simultaneously. Consequently, the SPDP load on the downlink equals the uplink:

$$L_{\text{SPDP}}^{\text{up}} = L_{\text{SPDP}}^{\text{down}} = \sum_{i=1}^{N_H} L_{\text{SPDP}}^{\text{host}_i} = N_H N_P \tag{2}$$

Note that while the message counts are identical, the physical airtime cost $\mathcal{T}(L^{\text{down}})$ differs from $\mathcal{T}(L^{\text{up}})$ due to the lower transmission rates forced upon multicast downlink traffic.

Upon SPDP discovery, the middleware initiates the SEDP phase to exchange endpoint information. Unlike SPDP, SEDP operates via reliable unicast. Each process transmits distinct SEDP messages for its N_E endpoints to all $(N_H - 1)N_P$ remote processes. Therefore, the SEDP load generated by a single host scales with the square of N_H:

$$L_{\text{SEDP}}^{\text{host}} = N_P \times [(N_H - 1)N_P] \times N_E = (N_H - 1)N_P^2 N_E \tag{3}$$

In addition to the SEDP messages, the reliable QoS policy (Heartbeat-ACK/NACK mechanism) introduces significant control overhead. Specifically, the SEDP exchange mandates each process to simultaneously advertise its own endpoint descriptors and acknowledge the receipt of remote descriptors across two distinct built-in topics: Publications and Subscriptions. Even under ideal lossless conditions, this bidirectional synchronization necessitates at least one Heartbeat-ACK pair per topic to confirm delivery. Therefore, a process pairing generates a total of four control messages (two Heartbeats as a sender and two ACKs as a receiver). The baseline control message load $L_{\text{control}}^{\text{host}}$ for a single host is thus derived as:

$$L_{\text{control}}^{\text{host}} = 4 \times N_P \times [(N_H - 1)N_P] = 4(N_H - 1)N_P^2 \tag{4}$$

The total SEDP load imposed on the wireless infrastructure is the sum of SEDP and control messages. Since the AP acts as a unicast relay, the uplink and downlink loads are symmetric:

$$\begin{aligned} L_{\text{SEDP}}^{\text{up}} = L_{\text{SEDP}}^{\text{down}} &= \sum_{i=1}^{N_H} \left(L_{\text{SEDP}}^{\text{host}i} + L_{\text{control}}^{\text{host}_i} \right) \\ &= N_H \left[(N_H - 1)N_P^2 N_E + 4(N_H - 1)N_P^2 \right] \\ &= N_H (N_H - 1) N_P^2 (N_E + 4) \end{aligned} \tag{5}$$

C. Analysis of Zenoh Discovery Traffic

In this subsection, we model Zenoh operating in a pure P2P mode, where sessions are established directly between peers without a centralized router. Similar to the DDS analysis, we decompose the traffic load L into uplink and downlink components to facilitate the airtime evaluation in Section IV. Zenoh's discovery process comprises two primary stages: Scouting and Declaration exchange.

In the Scouting phase, processes exchange multicast Scout messages and unicast Hello replies. From a single host's perspective, the traffic load is twofold. First, each process broadcasts periodic Scout messages ($L^{\text{host}}_{\text{Scout}} = N_P$). Second, scouting follows a request-reply pattern: upon receiving a remote Scout message, each process responds with a unicast Hello. Under the default configuration where no response suppression mechanisms are applied, this behavior leads to a fully connected logical topology during the discovery phase. Consequently, the host-level Scouting load $L^{\text{host}}_{\text{Scouting}}$ is:

$$L^{\text{host}}_{\text{Scouting}} = \underbrace{N_P}_{\text{Multicast}} + \underbrace{(N_H - 1)N_P^2}_{\text{Unicast}} \tag{6}$$

When mapped to the wireless infrastructure, we must distinguish between the multicast (Scout) and unicast (Hello) components, as they impact the medium differently. The aggregate uplink and downlink loads are derived as:

$$\begin{aligned} L^{\text{up}}_{\text{Scouting}} = L^{\text{down}}_{\text{Scouting}} &= \sum_{i=1}^{N_H} L^{\text{host}_i}_{\text{Scouting}} \\ &= \underbrace{N_H N_P}_{\text{Multicast}} + \underbrace{N_H(N_H - 1)N_P^2}_{\text{Unicast}} \end{aligned} \tag{7}$$

Following Scouting phase, Zenoh enters the Declaration exchange phase. A local peer initiates a unicast session with every discovered remote peer. The declaration load $L^{\text{host}}_{\text{Declare}}$ follows the same logical topology as DDS SEDP:

$$L^{\text{host}}_{\text{Declare}} = (N_H - 1)N_P^2 N_E \tag{8}$$

However, Zenoh achieves significant efficiency in control traffic. By consolidating reliability checks into the session layer via periodic KeepAlive frames, it requires only two control messages to maintain a link. The baseline control message load is thus:

$$L^{\text{host}}_{\text{control}} = 2(N_H - 1)N_P^2 \tag{9}$$

Combining these, the aggregate uplink and downlink loads for the Declaration exchange phase are:

$$\begin{aligned} L^{\text{up}}_{\text{Declare}} = L^{\text{down}}_{\text{Declare}} &= \sum_{i=1}^{N_H} \left(L^{\text{host}_i}_{\text{Declare}} + L^{\text{host}_i}_{\text{control}} \right) \\ &= N_H(N_H - 1)N_P^2(N_E + 2) \end{aligned} \tag{10}$$

IV. Numerical Analysis of Wireless Discovery

A. System Parameters and Environmental Assumptions

1) Packet Specifications: To translate the derived traffic volume into IEEE 802.11n airtime, we model each logical message as the payload of a single Wi-Fi data frame (MPDU), incorporating the requisite protocol overheads. For DDS, the nominal RTPS payload sizes are adopted from empirical packet captures in standard ROS 2 environments. While SPDP and SEDP sizes vary depending on locator lists and QoS parameters, Heartbeat and ACKNACK messages remain fixed or near-fixed in stable network conditions. For Zenoh, we distinguish between the scouting phase and the session phase: Scout and Hello messages are exchanged over UDP, whereas Declare and KeepAlive messages are transmitted over an established TCP session.

TABLE II
Discovery and Control Message Size and Wi-Fi Frame Mapping

Symbol	Range (B)	Payload (B)	Frame (B)
S_{SPDP}	$300-400$	350	414
S_{SEDP}	$300-500$	400	464
S_{Hb}	52	52	116
S_{ACK}	$48-52$	50	114
S_{Scout}	$8-32$	16	80
S_{Hello}	$40-200$	80	144
S_{Declare}	$60-250$	100	176
S_{REL}	$4-24$	8	84

Accordingly, we apply distinct per-frame overhead constants, H_{UDP} and H_{TCP}, to account for the IEEE 802.11 MAC header (including FCS), LLC/SNAP encapsulation, and the corresponding IPv4 and transport headers. We define these constants as $H_{\text{UDP}} = 64$ bytes and $H_{\text{TCP}} = 76$ bytes, respectively. All sizes reported in Table II represent the total Wi-Fi frame length ($S_{\text{frame}} = \text{Payload} + H$). The resulting frame size S_{frame} used in the airtime analysis represents a generic notation encompassing the message-specific frame sizes (e.g., S_{SPDP}, S_{SEDP}, S_{ACK}) listed in Table II. This model excludes optional fields such as security headers and QoS control fields, thus providing a minimum baseline for airtime computation under ideal conditions.

2) PHY Data Rates: To accurately estimate the airtime consumption, we define the data transmission speeds for multicast and unicast traffic under the IEEE 802.11n standard. For multicast traffic, the network prioritizes reachability over speed. Since these messages must be successfully received by every robot, AP is forced to use the most robust but slowest transmission speed. Consequently, we assume all multicast frames are transmitted at the basic rate of $R_{\text{multi}} = 6$ Mbps.

In contrast, unicast traffic takes advantage of the high-speed capabilities of IEEE 802.11n. The network can optimize the connection using modern techniques such as higher modulation schemes and multiple antennas (MIMO). Assuming a conservative configuration with 2 spatial streams (2×2 MIMO), we model the unicast rate at $R_{\text{uni}} \approx 144.4$ Mbps (MCS 7) [8].

B. Airtime Consumption and Channel Utilization

To quantitatively assess the feasibility of the derived traffic loads under wireless resource constraints, we adopt a deterministic airtime consumption model. This linear abstraction is widely established in IEEE 802.11 performance analysis to capture the dominant impact of protocol overheads on channel saturation. We define the airtime function $\mathcal{T}(L, R)$ as the total time required to transmit a load of L messages at a physical

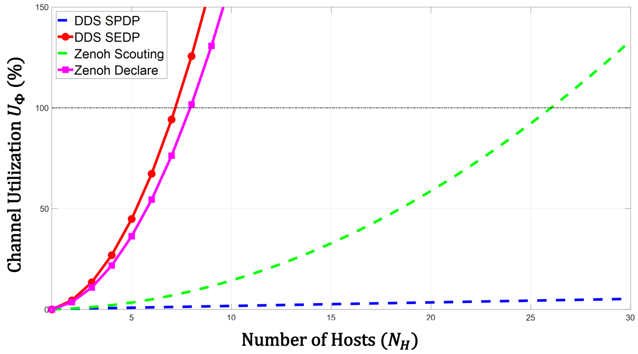

Fig. 2. Scalability Analysis for the Number of Hosts ($N_P = 2, N_E = 10$)

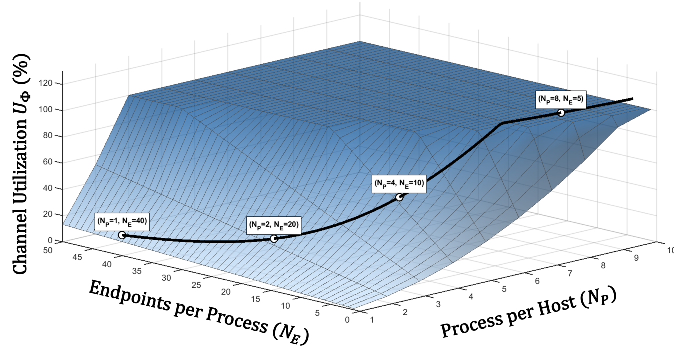

Fig. 3. Scalability Analysis for the Processes and Endpoints ($N_H = 3$)

data rate R. This function accounts for both the transmission delay and the protocol overheads:

$$\mathcal{T}(L, R) = L \times \left(\frac{S_{\text{frame}}}{R} + O_{\text{phy}} \right) \tag{11}$$

Here, S_{frame} represents the total frame size including lower-layer headers (as detailed in Table II), and R denotes the effective physical data rate corresponding to the transmission mode. Crucially, the term O_{phy} represents the fixed temporal overhead required to transmit any frame, regardless of its frame size. It accounts for all mandatory non-data intervals, such as physical headers, inter-frame waiting intervals (DIFS/SIFS), and acknowledgments. Based on the OFDM timing parameters defined in [9], we approximate these overhead constants as $O_{\text{multi}} \approx 120\mu s$ for multicast and $O_{\text{uni}} \approx 180\mu s$ for unicast.

Consequently, we define the total channel utilization U_Φ for a discovery phase Φ. U_Φ is formulated by summing the airtime consumed by uplink and downlink transmissions. Crucially, we apply distinct parameters for multicast and unicast traffic:

$$U_\Phi = \sum_{\phi \in \Phi} \left[\mathcal{T}(L_\phi^{\text{up}}, R_\phi^{\text{up}}) + \mathcal{T}(L_\phi^{\text{down}}, R_\phi^{\text{down}}) \right] \tag{12}$$

It is crucial to note that this model represents a theoretical lower bound. While we calculate utilization over a one-second average, our synchronized startup scenario implies that traffic is generated instantaneously rather than uniformly. In reality, this burstiness exacerbates collisions and exponential backoffs significantly beyond what our model predicts.

C. Stability Analysis and Discussion

In this subsection, we analyze the impact of the number of hosts, processes, and endpoints on scalability. Fig. 2 depicts the total channel utilization as a function of the number of hosts N_H. Given that typical autonomous mobile robot nodes exhibit an endpoint density of 5-12, we modeled a representative robotic setup with two processes ($N_P = 2$) and 10 endpoints ($N_E = 10$) to capture the discovery overhead relative to host scaling. For both DDS and Zenoh, the discovery process at the endpoint level leads to a quadratic increase ($O(N_H^2)$) in channel utilization as the host count grows. While Zenoh's Declaration exchange is slightly more efficient than DDS SEDP due to its reduced control message overhead, the performance gap remains marginal in the presence of high endpoint density. In the P2P mode, the wireless medium reaches a saturation point (100% utilization) at approximately 7 hosts for DDS and 8 hosts for Zenoh. Although the loads from the SPDP and Scouting phases are not dominant, Zenoh Scouting generates higher traffic than DDS SPDP. This is attributed to Zenoh's link-oriented discovery mechanism, which utilizes unicast Hello responses to prioritize session stability and context establishment in lossy wireless environments or WAN-scale deployments.

Discovery channel utilization can vary significantly depending on the architectural process design, even when the total volume of data exchange remains constant. Fig. 3 presents a scalability analysis regarding the number of processes N_P and endpoints N_E. As illustrated by the 3D surface in Fig. 3, we analyze the impact of process fragmentation by tracing a trajectory with a constant total endpoint count ($M = N_P \times N_E = 40$) across varying process-per-host (N_P) values. The black line clearly demonstrates that while a consolidated single-process configuration ($N_P = 1, N_E = 40$) maintains low channel utilization, subdividing the same functional density into multiple independent processes (e.g., $N_P = 8, N_E = 5$) triggers a quadratic surge in discovery traffic that quickly reaches 100% saturation. This phenomenon stems from the fact that discovery overhead, particularly in DDS SEDP, scales quadratically with the process count (N_P^2) due to the reliable unicast synchronization required between every pair of participants. These results suggest that network scalability in P2P-based swarm robotics can be improved by consolidating multiple nodes into a single process, for example using ROS 2 component nodes, thereby reducing the quadratic growth of control traffic.

This characteristic is closely related to the concept of composition in ROS 2, which reduces system overhead by executing the same number of nodes within a single process rather than distributing them across multiple independent processes. Similar to the Nodelet mechanism in ROS 1, ROS 2 provides explicit support for running multiple nodes within a single process while preserving their logical modularity and interfaces. This execution model minimizes inter-process communication overhead which would otherwise be incurred when nodes are isolated into separate processes.

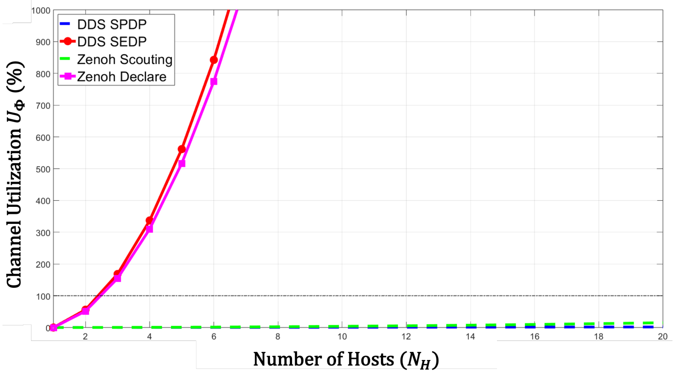

Fig. 4. Scalability Analysis of Nav 2 Scenario ($N_P = 1, N_E = 678$)

To evaluate scalability within a practical system context, we conducted a multi-robot scenario where TurtleBots perform autonomous navigation using the Nav2 stack without human intervention. As the most widely recognized stack for autonomous mobile robots, Nav2 consists of numerous modular processes responsible for localization, planning, control, and recovery behaviors. However, as previously noted, these components are not necessarily executed as separate processes; they can be co-located and executed within a single process through node composition.

Fig. 4 illustrates how overall channel utilization evolves as the number of hosts increases under this scenario. During Nav2 operations driven by sensor data, a single Nav2 process was observed to involve 678 endpoints, representing the total number of DataWriters and DataReaders participating in message exchanges. Our analytical model demonstrates that DDS SEDP or Zenoh Declaration exchanges alone can exceed 100% channel utilization with as few as three TurtleBots sharing the same wireless medium. This indicates that the discovery phase itself can saturate the communication channel. In practical deployments, additional processes—such as logging or teleoperation-typically run alongside navigation, further increasing network load and potentially pushing the wireless medium into sustained saturation.

The standard pure P2P discovery mode is highly vulnerable in multi-robot environments due to limited scalability and robustness. To address these limitations, Fast DDS offers the Discovery Server as an optional alternative to mitigate the excessive control traffic of distributed P2P discovery. In contrast, Zenoh avoids pure P2P by default, employing a hybrid architecture where each host runs a local Zenoh router (zenohd). This allows nodes to communicate primarily with their local router, while inter-host communication is managed through router-to-router interactions. This design significantly improves scalability by aggregating scouting and declaration traffic; however, both the Discovery Server and Zenoh router approaches introduce a potential SPOF, where a failure in these central entities can disrupt multiple communication sessions and degrade overall system connectivity.

V. Conclusion

In this paper, we presented a scalability analysis of the discovery protocols when DDS and Zenoh are operated in a pure peer-to-peer manner for wireless robotic swarms. By formulating a deterministic airtime consumption model, we quantified the impact of protocol overheads and identified the precise saturation points where the quadratic growth of discovery traffic exceeds wireless channel capacity. Our results confirm that current brokerless mechanisms are inherently constrained to small-scale clusters, providing a mathematical explanation for the “Discovery Storm” and connectivity failures frequently encountered in real-world deployments.

For future work, we plan to explore practical mitigation strategies to extend these scalability limits. Specifically, we will investigate the temporal dynamics of reliability mechanisms; while bandwidth saturation causes packet loss, a detailed analysis of retransmission behaviors may reveal conditions under which discovery eventually converges, albeit with increased latency. Additionally, we intend to examine traffic shaping techniques, such as randomized startup delays, to mitigate instantaneous bursts.

Furthermore, future work will focus on developing a resilient discovery strategy to mitigate the SPOF risks inherent in the hybrid architecture, which is the default configuration for rmw_zenoh. To validate these approaches, experimental evaluations on real-world robotic platforms under realistic wireless conditions will be conducted. These experiments are expected to complement our analytical findings by revealing how practical factors, such as timing jitter and packet loss, impact the effectiveness of our SPOF-prevention mechanisms. Ultimately, our goal is to design and implement a robust, scalable, and fault-tolerant discovery architecture for next-generation ROS 2 middleware.

References

[1] E. Latif, W. Song, and R. Parasuraman, “Communication-efficient reinforcement learning in swarm robotic networks for maze exploration,” in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, IEEE, 2023.

[2] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot operating system 2: Design, architecture, and uses in the wild,” *Science robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[3] Object Management Group (OMG), *The Real-time Publish-Subscribe Protocol (RTPS) DDS Interoperability Wire Protocol Specification, Version 2.2*, Sept. 2014. OMG Document Number: formal/2014-09-01.

[4] A. Corsaro, L. Cominardi, O. Hecart, G. Baldoni, J. E. P. Avital, J. Loudet, C. Guimares, M. Ilyin, and D. Bannov, “Zenoh: Unifying communication, storage and computation from the cloud to the microcontroller,” in *2023 26th Euromicro Conference on Digital System Design (DSD)*, pp. 422–428, IEEE, 2023.

[5] H.-S. Park, S. Lee, D. Um, H. Ryu, and K.-J. Park, “An analytical latency model of the data distribution service in ROS 2,” in *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications*, pp. 1–10, 2025.

[6] S. Lee, H.-S. Park, J. Chae, and K.-J. Park, “Probabilistic latency analysis of the data distribution service in ROS 2,” 2025.

[7] S. Lee, T. Kim, J. Chae, and K.-J. Park, “Optimizing ROS 2 communication for wireless robotic systems,” 2025.

[8] E. Lopez-Aguilera, E. Garcia-Villegas, and J. Casademont, “Evaluation of IEEE 802.11 coexistence in wlan deployments,” *Wireless Networks*, vol. 25, no. 1, pp. 87–104, 2019.

[9] M. Gast, *802.11 wireless networks: the definitive guide*. ” O’Reilly Media, Inc.”, 2005.