A Novel Routing Protocol for MANET-Based Smart Indoor Environments

Naazim Ali Khan †‡ Zhe Lou † Alba Jano ‡ Wolfgang Kellerer ‡
†Huawei Technologies Co., Ltd
‡ Chair of Communication Networks, Technical University of Munich
naazim.khan@tum.de, zhe.lou@huawei.com, alba.jano@tum.de, wolfgang.kellerer@tum.de

***Abstract*—Mobile Adhoc Networks (MANETs) provide a flexible and infrastructure-free communication framework, making them suitable for smart indoor environments. They enable direct, dynamic, and efficient communication between devices. Nevertheless, achieving optimal routing in such environments remains a critical challenge. Routing plays a critical role in MANETs, as it ensures efficient packet delivery between mobile and dynamically connected nodes. In this paper, we propose and evaluate a novel routing protocol for MANETs, specifically designed for Smart Home networks. Our approach is tailored to address the unique challenges of smart indoor environments, such as energy efficiency, latency, and adaptability to topology changes. Through extensive simulations, we demonstrate that our protocol significantly outperforms state-of-the-art routing protocols, decreasing power consumption by 50.46%, control overhead by 60.85%, while maintaining end-to-end delay, making it a promising solution for Smart Home networks.**

***Index Terms*—Mobile Adhoc Network, Smart Home, routing, power consumption, control overhead, delay.**

I. Introduction

Mobile Adhoc Networks (MANETs) play an important role in wireless communication, offering a decentralized and infrastructure-free solution for dynamic networks [1]. Their self-organizing nature ensures seamless communication establishment, even in the absence of pre-existing infrastructure. A MANET enables smart devices, like phones, laptops, sensors and appliances, to communicate with each other directly without requiring a fixed central wireless framework. Smart indoor environments, characterized by a growing number of connected mobile devices and high demand for flexibility, represent a key application where MANETs are particularly well-suited [2]. In such a scenario, devices typically comprise of nodes with limited processing and constrained energy resources. Therefore, the adaptability of MANETs is tempered by challenges, including dynamic topologies, constrained node resources, and maintaining efficient communication in such environments.

Efficient routing is essential in MANETs, as it ensures smooth communication between mobile nodes. Without robust routing mechanisms, the network performance can degrade significantly, leading to issues such as delay and energy inefficiency. MANET routing protocols facilitate packet forwarding by utilizing information about the links within the network. These protocols can be categorized based on their update mechanisms into proactive, reactive, and hybrid protocols [3]. Proactive protocols determine paths between node pairs using routing information stored in forwarding tables. These protocols have minimal latency because paths are precomputed regardless of whether data transmission is required. However, maintaining routes that may not be required results in high overhead, especially when updating routing information to keep pace with a changing topology. Reactive protocols establish routes only when necessary. These protocols favour resource efficiency and low power consumption at the cost of incurring high communication-setup latency, as communication is delayed by the time taken to discover routes. Hybrid routing protocols combine the benefits of both proactive and reactive protocols, offering the potential for better scalability compared to purely reactive or proactive protocols.

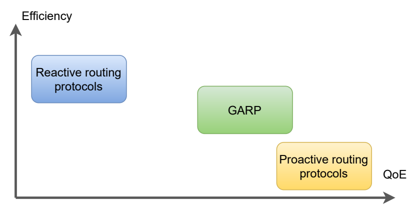


Fig. 1: GARP in comparison with existing protocols.

The energy constraints combined with application requirements call for a routing approach that can achieve a balance between power consumption and latency while maintaining a low control overhead. In this paper, we introduce a novel routing approach termed as *Green Adhoc Routing Protocol* inspired by our previous work [4]. GARP focuses on improving the efficiency of topology updates while also introducing a reactive component to discover routes which are unknown. We analyze GARP's performance in comparison to the foundational proactive and reactive approaches that are widely studied in MANETs. Hybrid protocols, designed for broader scalability and dynamic environments, are resource-intensive and less relevant to the constraints and requirements of smart indoor networks [5]. Fig. 1 illustrates the performance of GARP in relation to efficiency and Quality of Experience (QoE), compared to proactive and reactive protocols. Proactive protocols typically consume high power (low efficiency) along with a large control overhead while achieving low latency (high QoE). In contrast, reactive protocols consume significantly low power (high efficiency) along with low control overhead but experience high latency (low QoE).

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

To verify the performance of GARP, we perform extensive simulations in an indoor smart home scenario, including nomadic mobility and dynamism. Our results demonstrate that GARP is positioned between these two categories of routing protocols, reducing the power consumption by an average of 50.46% compared to proactive, reducing mean end-to-end delay by an average of 94% compared to reactive, while also reducing mean control packet length by an average of 80.2% and 41.5% compared to proactive and reactive respectively.

The paper is structured as follows. Section II presents a thorough state-of-the-art review of existing protocols. We elaborate on the main features of GARP protocol in Section III. The results of our proposed solution are elaborated in Section IV. Finally, the paper is concluded in Section V.

II. Related Work

The research on MANETs aims to improve end-to-end delay, control overhead and power consumption experienced by devices. These existing works are categorized here based on the improvements to - (i) Adhoc On-Demand Distance Vector (AODV) (ii) Optimized Link State Routing (OLSR).

The early research defined in [6] introduces AODV as a routing protocol that enables dynamic, self-starting, multi-hop routing between nodes to establish and maintain adhoc networks. When the source node lacks routing information about the destination node, it initiates the route discovery process. The paper [7] introduces the A-LSEA protocol, which enhances routing by considering both link stability and residual energy. This approach is able to enhance packet delivery ratio and end-to-end delay at high speed. However, for lower speed of nodes, the delay is observed to be higher than traditional AODV. Authors in [8] improve the traditional AODV by proposing the ALQ-AODV algorithm which selects the final path on the basis of link quality, node's residual energy and degree of an intermediate node. While this approach increases route efficiency and performance, its reliance on the use of *HELLO* messages and additional energy-level monitoring leads to increased power consumption, memory usage and processing in resource-constrained devices. The work in [9] introduces the QL-AODV protocol, incorporating Q-learning and evaluating path congestion and queue length of nodes during route discovery. This approach achieves lower delay and normalized routing overhead, but does not account for the increased computation due to additional metrics.

The work in [10] proposes OLSR as a proactive routing protocol which uses the link-state algorithm and paths to all destinations in the network are calculated and maintained before a data packet is sent from a source node. Nodes use *HELLO* messages for neighbor discovery. Topology control (TC) messages are used to discover and broadcast link-state information throughout the network periodically. It reduces the end-to-end latency and improves the QoE. Although a set of multi-point relays (MPR) [11] is selected by each node to forward control traffic efficiently, the frequent periodic refreshing leads to higher power consumption, which is not desirable for battery-enabled devices. The paper [12] improves the OLSR protocol by integrating energy efficiency and security measures and optimizing MPR selection using multiple parameters. The approach enhances energy efficiency, network lifetime and packet delivery. The integration of multiple metrics necessitates complex decision-making, potentially leading to increased processing delay and high control message overhead. Authors in [13] propose improvements to the MPR selection process by incorporating metrics such as node mobility and energy levels. The enhanced algorithm achieves better packet delivery ratio, reduced delay, and improved energy efficiency compared to the standard OLSR protocol. However, its reliance on additional metrics for MPR selection may require more frequent updates in highly dynamic networks.

The related works focus on efficiency and energy optimization while identifying challenges like increased computation, delays and control overhead. This paper addresses these limitations by proposing a balanced approach for routing in resource-constrained smart indoor environments.

III. Design and Features of GARP

GARP protocol operates as a table-driven, distance vector protocol exchanging topology information regularly with neighboring nodes [4]. The fundamental idea of GARP is to improve the efficiency of topology updates. It functions in a fully distributed manner, without any centralized control entity. Therefore, each node in the network maintains topology information in local tables, to route packets hop-by-hop. While this behavior classifies GARP as a proactive protocol, its design also incorporates a reactive component to address the discovery of routes that are not immediately available.

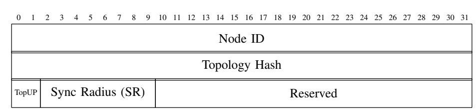


Fig. 2: GARP HELLO packet structure.

A. Messages structure

HELLO message: In GARP, nodes periodically send HELLO messages to their 1-hop neighbors to establish and maintain the neighbor relationship. The default interval of HELLO messages is 1 s. The format of the HELLO message is displayed in Fig. 2. The *Node ID* field contains an unique identifier for each node. Upon receiving a HELLO message, a node updates its Neighbour Information Table (NIT). The HELLO message is also used to verify whether the topology information is synchronized between neighbors. The *Topology Hash* field is a 32-bit hash value calculated from information available in the node's Topology Information Database (TIDB). When a node receives the same hash value from a neighbor, it indicates that the topology information of both nodes is synchronized. The 2-bit *TopUp* field is used as an indicator of the changes in the node's links since the last HELLO message transmission. This value determines which information will be shared with the node's neighbors in the Topology Synchronization Message (TSM). The default value of *TopUp* is 11, indicating that multiple changes have occurred to the node since the previous HELLO message transmission. The *Sync Radius* field is a 1-byte value, N, which indicates that all nodes within N hops share the same topology information at the time of transmission. The *Sync Radius* value provides an easy and efficient method to determine whether additional topology information is required.

TSM message: A node sends and receives TSM messages based on requirement - (i) when it receives a HELLO message with a different *Topology Hash*, TSM messages are exchanged between neighbors to share updated topology information. (ii) When the reactive component of GARP is triggered due to a route requirement. The format of the TSM message is shown in Fig. 3. The *Type* field is a 8-bit field which is used to identify the type of the TSM message (1 = Request; 2 = Reply; 3 = Flag; 4 = TopSync). The *Node ID* field is an unique router identifier. The *No. of Records* field shows the number of records attached to the TSM message. The *ParentID* field is a 4-octet random value created by the sender of the TSM message. This field will be returned in the subsequent reply and is used as an index to identify the corresponding request when a reply is received.

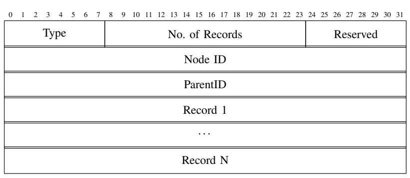


Fig. 3: GARP TSM packet structure.

B. Partial Topology Update

The partial topology update algorithm is a key differentiating factor of GARP. This mechanism enables GARP to significantly reduce control overhead and power consumption in comparison with proactive protocols like OLSR. The main idea of this algorithm is to share topology information among neighbors based on the content of the HELLO message. In OLSR, redundant topology information such as information of stationary neighbors and unchanged MPR nodes, is often exchanged between nodes causing a large control overhead which increases nodes' power consumption. Instead, in GARP, nodes, using *TopUp* field information, only share the updates that have occurred since the last transmission of HELLO message. Fig. 4 illustrates the call flow chart of GARP partial topology update in a scenario with three nodes.

Step 1: Let us assume two nodes, A and B, during startup. Nodes A and B exchange HELLO messages. By default, the 2-bit *TopUp* field has a value of 11 which indicates that nodes A and B will exchange their full topology information.

Step 2: Upon exchanging HELLO messages, nodes A and B create and update their NITs and TIDBs.

Step 3: Nodes A and B exchange TSM messages (Request and Reply). The content of the TSM message is determined by the value of the *TopUp* field mentioned in Step 1. Next, A and B update their TIDBs. Now, both TIDBs are identical, leading to the same topology hash value, denoted as THV(X).

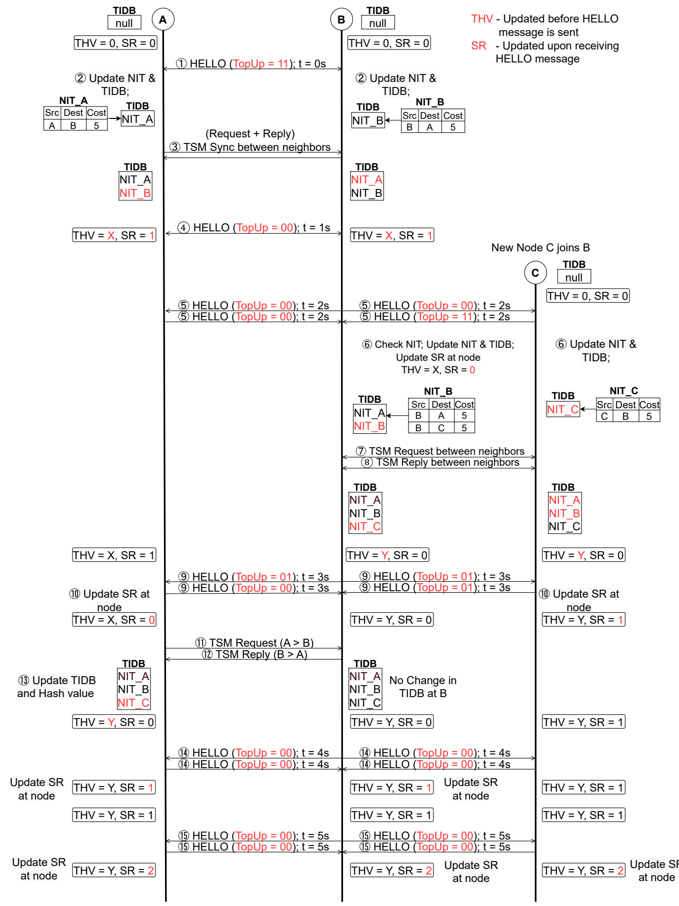


Fig. 4: GARP Partial Topology Update Call Flow

Step 4: At $t = 1\,\text{s}$, nodes A and B exchange HELLO messages once again. As no additional topology changes have occurred in either node, the *TopUp* field becomes 00. Node A and B update their sync radius (SR) to 1, upon verifying that the exchanged hash values are identical.

Step 5: Consider a third node C has joined B between $t = 1\,\text{s}$ and $t = 2\,\text{s}$. At $t = 2\,\text{s}$, nodes A, B and C transmit the next HELLO message. Node A has not experienced any changes and hence, the *TopUp* field has a value of 00. Node B, still unaware of node C's presence, also transmits a HELLO message with *TopUp* value 00 and same hash value THV(X). In contrast, node C, being a newly introduced node, uses the default value 11. Upon receiving the HELLO message from node B, node A compares the received hash value with its own and refrains from requesting additional information, as both hash values match. Node B follows the same procedure upon receiving the HELLO message from node A.

Step 6: The HELLO message transmitted by node C is received by node B, which compares the received hash value THV(0) with its own hash value THV(X). As the values do not match, node B immediately resets its *Sync Radius* to zero and updates its NIT and TIDB to reflect the network changes. Similarly, node C, upon receiving the HELLO message from node B, creates and updates its own NIT and TIDB.

Steps 7 and 8: Nodes B and C exchange TSM messages

(Request and Reply). Given that the *TopUp* value of node C in Step 5 was 11, node B sends its entire TIDB information to node C. Node C, also sends its entire TIDB information to node B. Through this exchange, node C becomes aware of node A's connection to node B. As a result, both nodes B and C now possess synchronized topology information and calculate their updated hash value, denoted as THV(Y).

Step 9: At $t = 3$ s, during the next HELLO message exchange, node A is notified of the topology change at node B by decoding the received HELLO message. The different topology hash value, along with a *TopUp* value of 01 indicates that a new node has joined node B. Simultaneously, nodes B and C decode the HELLO messages received from each other. Although the value of *TopUp* field is 01, nodes B and C have the same hash value THV(Y). This confirms that both nodes have the same topology information, eliminating the need for further TSM message exchange between them.

Step 10: Node A, upon receiving a HELLO message with different hash value from node B, resets its *Sync Radius* to zero. At node C, the *Sync Radius* is incremented by one.

Steps 11 and 12: Node A initiates a TSM Request to node B. In response, node B sends a TSM Reply containing only the updated topology records that are missing at node A. This highlights the advantage of using the partial topology update.

Step 13: Using this information from TSM Reply message, node A updates its TIDB and updates it hash value THV(Y).

Step 14: At $t = 4$ s, all nodes transmit HELLO messages. At this point, all nodes share the same hash value, THV(Y). Node A, upon receiving the HELLO message from node B, increments its SR to one. Similarly, node B increments its SR to one upon receiving HELLO messages from nodes A and C. However, node C's SR remains at one, as in Step 10. Node B transmits its HELLO message with the same hash value, but its SR is still equal to zero during transmission. When the hash value of received HELLO message is equal to the node's hash value, the SR is calculated based on the minimum value between the node's SR and SR values of all received HELLO messages whose hash value is equal.

Step 15: At $t = 5$ s, the nodes transmit HELLO messages. As no additional changes have occurred in the network topology, all three nodes have the same hash value. Consequently, upon receiving HELLO messages, nodes A, B and C increase their SR value to two. This process demonstrates that, through the partial topology update mechanism employed by GARP, the network of three nodes reaches synchronization and the same topology information within five seconds.

C. Partial Flooding and Forwarding

Each node maintains a routing table which is used to forward data packets to the other nodes in the network. This routing table is calculated based on information in the TIDB. Typically, routes in the routing table of a node N consist of paths to nodes that are at most one hop beyond its current SR. When node N needs to communicate with a destination node beyond its SR, it requires an efficient mechanism to discover routes by acquiring topology information from parts of the network outside of its current knowledge. GARP addresses this requirement by introducing a reactive route finding mechanism known as *Partial Flooding and Forwarding*. A conventional reactive protocol like AODV would initiate a broadcast to find the destination and obtain the route. Instead, our proposed mechanism leverages existing routing information based on the SR. GARP avoids broadcasting, full flooding and retransmission of messages among intermediate nodes that lie upto the SR.

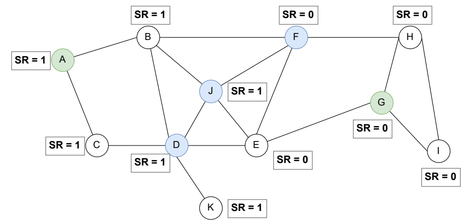


Fig. 5: Illustrative example of Partial Flooding and Forwarding mechanism, where node A discovers node G.

This mechanism is illustrated and explained through a toy example in Fig. 5. The SR of each node is displayed beside the node. Between $t = 2$ s and $t = 3$ s, nodes G, H and I join the network. Node A wants to communicate with node G at $t = 3.25$ s. At this time, node A has routes to nodes D, F and J (in blue). Therefore, node A initiates unicast TSM TopSync messages (*Type* = 11) to the farthest reachable nodes according to its TIDB, which are D, F and J, respectively. Node A places an unique value in the *ParentID* field, which is used by other nodes to identify the originator of this message. Upon reception of the message, a *TopSync* flag is set at nodes D, F and J. The responsibility of discovering topology further beyond the SR of node A is transferred to these nodes.

At nodes D, F and J, based on the SR value, a decision is taken on the type of TSM message that is to be sent. If SR < 2, these nodes initiate the transmission of TSM Flag message (*Type* = 10). This transmission is only done to the neighbors which do not have information of the source node (node A) of the received TSM Topsync message. Hence, TSM Flag messages are not transmitted to nodes B and C. A timer is initiated to wait for the corresponding TSM Reply.

Once the TSM Flag message has been transmitted, the nodes which receive it use the algorithm to either forward the same message, or, if *Topsync Flag* is set, drop the received message. If the message is not dropped, the process of forwarding is repeated based on the SR value of the node. Upon reaching the edge of the network where further forwarding or flooding is not possible, a TSM Reply message is generated back to the immediate parent of this node. This TSM Reply contains the TIDB records of this node which will be used by its parent node to update the TIDB and send a TSM Reply message back to the TSM originator nodes. The subsequent TSM Reply messages are used by nodes to update their TIDBs. Once node A receives its TSM Reply messages, it updates its TIDB and is able to update its routing table to initiate communication and send data packets to node G.

IV. EVALUATION

In this section, we demonstrate the performance of GARP lies between proactive and reactive algorithms in terms of packet delay and power consumption of the network. To show the improvements introduced in this work, we compare our proposed routing protocol GARP with two representative state-of-the-art works: AODV [6] and OLSR [10]. We analyze the results focusing on key performance metrics, including power consumption, end-to-end delay, and control packet overhead. GARP is implemented on a OMNET++ v6.0.2 environment [14], with integrated INET framework [15]. To ensure a fair comparison with the state-of-the-art, we utilize the existing implementations of AODV and OLSR in INET.

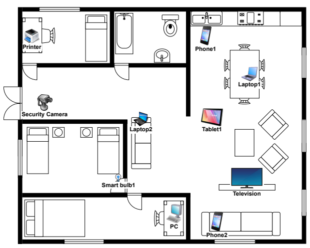


Fig. 6: Smart Home network scenario

We perform our evaluation in a smart home network as shown in Fig. 6, where we choose a realistic number of ten nodes connected to the network. During the evaluation, we differentiate between three scenarios: Static, Nomadic, and Dynamic, denoted in the results with S, N, and D labels. Table I summarizes the details of each scenario. In the static scenario, we assume all the connected nodes are static throughout the simulation. In the Nomadic scenario, nomadic mobility is enabled for specific nodes to mimic the realistic behavior in smart home MANETs, where the mobility of nodes is irregular and unpredictable. The Dynamic scenario builds on the Nomadic scenario by introducing dynamism by switching on and off some nodes during the simulation. The results are generated by averaging over 15 simulation repetitions, where each simulation has a duration of $5\,\mathrm{min}$.

A. *Node power consumption*

Power consumption of nodes has been calculated using the standard state-based energy consumption model of the INET framework [15], where transceiver energy usage is derived from state-specific power consumption and accumulated over time to compute overall power consumption. In Fig. 7, we present the reduction in power consumption of the nodes when utilizing GARP, compared to OLSR. The initial increase in the power consumption for GARP is attributed to the transient phase during which nodes are not yet synchronized. In this phase, a minimum of three control packets are exchanged between each pair of unsynchronized neighbors. This leads to a higher control packet overhead, specifically the transmission of *HELLO* and *TSM* packets in a 3:1 ratio compared with OLSR. Nevertheless, the final cumulative power consumption achieved by GARP is reduced by 54.7% (Static), 53.7% (Nomadic) and 43% (Dynamic), compared to OLSR. In the Dynamic scenario, we observe OLSR's power consumption reduces significantly due to the prolonged shutdown of nodes, resulting in fewer transmissions of HELLO and TC messages. This reduction in power consumption is diminished in GARP as the active nodes get synchronized, only reducing the number of exchanged HELLO messages. Compared to AODV, GARP increases the nodes' cumulative power consumption by an average of 95% across all scenarios. The existence of a proactive component in GARP makes it impractical to expect its power consumption to be closer to AODV.

TABLE I: Simulation scenarios.

Scenario	Description
Static (S)	Routing without mobility
Nomadic (N)	Nomadic movement of phones and laptops PC communication with Printer - UDP application Phone 2 with TV - UDP streaming application Security camera and smart bulb to Phone1 - Periodic data
Dynamic (D)	extend Nomadic At t = 55s, Smart bulb is turned off At t = 150s, Laptop1 is turned off At t = 185s, Smart bulb is turned on

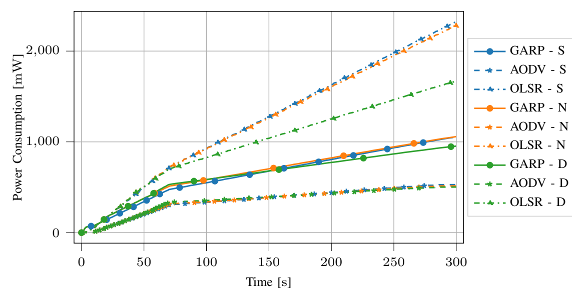
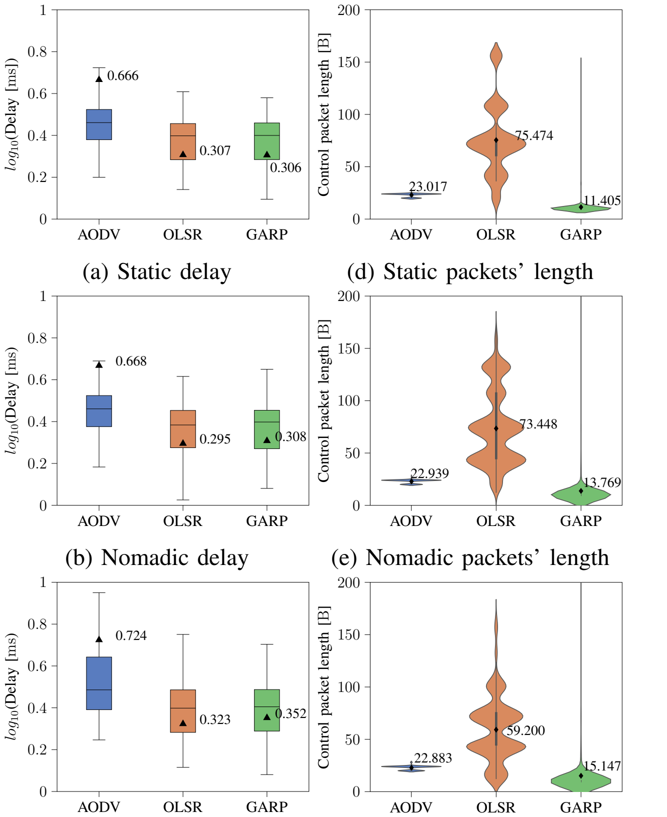


Fig. 7: Cumulative power consumption of nodes in a smart home network for AODV, OLSR and GARP.

B. *End-to-end delay*

Figures 8a, 8b and 8c display the end-to-end delay of transmitted packets using box plots in logarithmic scale to improve the readability and clarity of results. In all scenarios, GARP achieves similar performance to OLSR related to mean delay. The highest mean delay for GARP observed is $3.637\,\mathrm{ms}$ in the Dynamic scenario. This increase in delay is related to the route re-calculation and re-establishment resulting from mobility and switching (on/off) of nodes. The same applies also for AODV, which experiences higher packet transmission delay compared to GARP and OLSR. As a result, the maximum delay for GARP and AODV is observed to be $78\,\mathrm{ms}$ and $495\,\mathrm{ms}$, respectively. The extremely high mean delay of AODV, along with its high maximum delay, is due to data packets associated with periodic applications. These applications send periodic bursts of data packets at fixed intervals and in the smart home network, periodic applications are established between Phone-1 and nodes such as smart bulb ($interval = 10\,\mathrm{s}$) and security camera ($interval = 25\,\mathrm{s}$). The existence of a route timeout feature (default = $5\,\mathrm{s}$) in AODV results in having routes to be calculated for every interval.



(a) Static delay (d) Static packets' length

(b) Nomadic delay (e) Nomadic packets' length

(c) Dynamic delay (f) Dynamic packets' length

Fig. 8: Network nodes' end-to-end delay and control packet length for AODV, OLSR and GARP.

C. Control packet length

Fig. 8d, Fig. 8e and Fig. 8f present violin plots illustrating the distribution of control packet length. Due to the novel structure of GARP's messages, it exhibits the smallest control packet length compared to AODV and OLSR in all scenarios, especially for the Static scenario, where the mean control packet length is 11.405 B. This is 50.5% and 84.9% lower compared to AODV and OLSR, respectively. The constant size of HELLO packets (10 B), combined with nomadic mobility of nodes, significantly contribute to GARP's low mean packet lengths. AODV's control packet length is similar across all scenarios as it is a purely reactive protocol with control packet transmission occurring only due to route establishment and re-calculation. OLSR has a high mean control packet length as well as an elevated violin plot due to the variable length of its control packets and frequent transmission of redundant control information. OLSR's mean packet length decreases in Dynamic scenario due to the prolonged shutdown of specific nodes. However, we observe an increase in the case of GARP as the Dynamic scenario triggers GARP's reactive component due to mobility and shutdown of nodes. Nevertheless, GARP's mean control packet length is the least in Nomadic and Dynamic. For Nomadic, the mean of GARP is 40% and 81.3% lower compared to AODV and OLSR respectively. For Dynamic, the mean of GARP is 33.9% and 74.5% lower compared to AODV and OLSR respectively. On average, we observe a decrease of control packet length by 41.5% compared to AODV and 80.2% compared to OLSR.

V. Conclusion

In this paper, a novel routing protocol, GARP has been proposed for MANET indoor scenarios. The drawbacks exhibited by widely used MANETs' protocols, such as AODV and OLSR are addressed in GARP. Its unique differentiating features, such as partial topology update and partial flooding and forwarding, contribute to reducing the power consumption compared to OLSR, minimizing control overhead compared to both AODV and OLSR, and maintaining low end-to-end delay similar to OLSR. In a realistic smart home network, we are able to observe the full potential of GARP's benefits by simulating nomadic mobility and deploying realistic applications between nodes. Future research could focus on testing GARP's performance and scalability in larger networks. Another important aspect of future work is incorporating comprehensive security considerations to ensure that GARP is resilient in smart home environments. Further, there is also the possibility of exploring advanced mobility models that simulate complex and realistic patterns in highly dynamic environments.

References

[1] A. H. Alwan *et al.*, "Monitoring the impact of the nodes density on the manets' environment," in *2022 5th Int. Conf. on Eng. Technol. and its Appl. (IICETA)*, 2022.

[2] V. Dattana, A. Kumar, A. Kush, and S. I. Ali Kazmi, "Manet for stable data flow in smart home and smart city," in *2019 4th MEC Int. Conf. on Big Data and Smart City (ICBDSC)*, 2019.

[3] T. Leenas and S. Shriparen, "Comparison of proactive, reactive, and hybrid routing protocols in mobile ad hoc networks," in *2021 10th Int. Conf. on Inf. and Automat. for Sustain. (ICIAfS)*, 2021.

[4] D. Lou, L. Iannone, D. Trossen, and Z. Shi, "Sloppy Topology Updates for ad-hoc Routing Protocols (STURP)," Internet-Draft draft-lou-manet-sturp-02, Internet Engineering Task Force, 2024. Work in Progress.

[5] R. Saleh and I. Zebari, "Enhancing network performance: A comprehensive analysis of hybrid routing algorithms," *Asian J. of Res. in Comput. Sci.*, 2025.

[6] C. Perkins, E. Belding-Royer, and S. Das, "RFC3561: Ad hoc On-Demand Distance Vector (AODV) Routing," 2003.

[7] S. Hamad, S. Belhaj, and M. Muslam, "Average link stability with energy-aware routing protocol for manets," *Int. J. of Adv. Comput. Sci. and Appl.*, 2018.

[8] P. Pandey and R. Singh, "Efficient routing in mobile ad hoc networks based on awareness of link quality," in *Proc. of Int. Conf. on Comput. Commun. and Intell. Syst. (ICCCIS)*, 2021.

[9] E. Temesgen, A. O. Salau, E. Yitayal, and S. L. Braide, "Efficient manets routing algorithm to mitigate latency based on aodv," in *2023 Int. Conf. On Cyber Manage. And Eng. (CyMaEn)*, 2023.

[10] T. H. Clausen, C. Dearlove, P. Jacquet, and U. Herberg, "The Optimized Link State Routing Protocol Version 2." RFC 7181, 2014.

[11] S. Dong and H. Zhang, "An mpr set selection algorithm based on set operation," in *2021 IEEE 5th Adv. Inf. Tech., Electron. Autom. Control Conf. (IAEAC)*, 2021.

[12] H. Kanagasundaram and K. Ayyaswamy, "Eimo-esolsr: energy efficient and security-based model for olsr routing protocol in mobile ad-hoc network," *IET Commun.*, 2019.

[13] Y. Hamzaoui, C. Mohammed, and A. Mohamed, "Enhanced mpr selection algorithm based on metrics for the olsr routing protocol," 2023.

[14] A. Varga and R. Hornig, "OMNeT++ version 6.0.2." https://omnetpp.org, 2022.

[15] A. Varga and R. Hornig, "INET version 4.5.2." https://inet.omnetpp.org, 2023.