RESEARCH PAPER
Future Research in AI & IoT, 2026, 2(1)
Homepage: https://frai.journals.umz.ac.ir/
DOI: 10.22080/frai.2026.30847.1041

Fuzzy Reinforcement Learning in Opportunistic Routing for Wireless Sensor Networks

Toktam Kamali Yazdi [1], Nastaran Evaznia [2]*, Sobhan Esmaeili [3], Javad Rezazadeh[4]

[1] *Department of Computer Engineering, Ma.C., Islamic Azad University, Mashhad, Iran.*

[2] *Department of Computer Engineering, Ma.C., Islamic Azad University, Mashhad, Iran.*

[3] *Faculty of Engineering and Technology, University of Mazandaran, Babolsar, Mazandaran, Iran.*

[4] *Deputy Head of School IT, Crown Institute of Higher Education (CIHE), Sydney, Australia.*

Article Info
Received 19 December 2025
Accepted 30 January 2025
Available online 31 January 2025

Keywords:
Wireless sensor networks;
Candidate selection;
Opportunistic routing;
Fuzzy systems;
Reinforcement learning.

Abstract:
In recent years, routing in wireless sensor networks (WSNs) has emerged as a key research challenge due to the dynamic characteristics and constrained resources of these networks. Opportunistic Routing (OR) has emerged as an effective model that leverages the broadcast capabilities of wireless communication to improve network efficiency. The fundamental concept of OR is to select a suitable candidate subset at each node: upon receiving a packet, only the best candidate forwards it, while the others discard it, thereby enhancing reliability and minimizing redundancy. This paper aims to identify the optimal candidate set in opportunistic routing. This document proposes a new hybrid routing method, FRLOR (Fuzzy Reinforcement Learning-based Opportunistic Routing), that integrates Fuzzy Logic (FL) and Reinforcement Learning (RL) to enable smart, dynamic, and adaptive candidate selection in opportunistic routing. The fuzzy inference system assesses three fundamental input factors—geographical distance, neighbor node density, and link probability—to identify an initial candidate set. The RL element subsequently enhances this collection by continuously learning from network feedback and optimizing policies to select the most effective forwarding nodes. The effectiveness of the proposed FRLOR technique was assessed and compared with current algorithms, such as EEFLPOR, POR, and DPOR, using the Expected Number of Transmissions (ENT), Execution Time, End-to-End Delay (E2E Delay), Packet Delivery Ratio (PDR), and Energy Consumption. Simulation results indicate that integrating fuzzy reasoning with reinforcement learning substantially improves routing efficiency and network performance compared with conventional approaches.

***Corresponding Author:** nastaran.evaznia@iau.ir

Supplementary information: Supplementary information for this article is available at https://frai.journals.umz.ac.ir/

Please cite this paper as: Kamali Yazdi, T., Evaznia, N., Esmaeili, S., & Rezazadeh, J. (2026). Fuzzy Reinforcement Learning in Opportunistic Routing for Wireless Sensor Networks. Future Research on AI and IoT, 69-83. DOI: 10.22080/frai.2026.30847.1041

1. Introduction

Wireless networks have become central to everyday life in the modern era. One important feature of these networks is their broadcast nature. When a node transmits a packet, neighboring nodes can detect and receive that transmission. Routing is a key challenge in wireless networks. In conventional methods and schemes, a route is first determined and then used to forward packets [1, 2]. However, conventional routing protocols fail to account for the natural broadcasting capability of wireless environments. To enhance network performance and better utilize their broadcast characteristics, Opportunistic Routing (OR) was introduced [3, 4]. OR leverages the benefits of the wireless environment to improve the performance. Unlike traditional routing, OR sends a packet to a group of nodes instead of one specific recipient. The results in [1] show that OR outperforms other traditional algorithms. To design OR, three parameters are considered: how to select forwarders, how to compute a routing metric for prioritization, and how to coordinate among those forwarders. Among these, forwarder selection is a principal

ISSN 3092-7552

challenge. Each node in the network executes a decision algorithm and then, based on its output, chooses its candidates. The objective of this selection process is to minimize the total transmissions required for successful packet delivery. Metric calculation, which is necessary to select and prioritize the candidate set, requires a measure or standard [5, 6]. Generally, metrics are divided into two categories: local and end-to-end approaches [7]. A local metric considers only the local information of neighboring nodes to send packets. In fact, the choice of the next node is determined by factors such as link probabilities and node geographic information. In an end-to-end metric, all node information and status are considered to determine the optimal path. Although this method may select the optimal route, it significantly increases computational cost. The next parameter in OR is candidate coordination, a mechanism for detecting the best and highest-priority node. Subsequently, other nodes in the candidate set discard the packet. Several coordination methods have been proposed among candidates, including timer-based, acknowledgment-based, RTS/CTS, and network coding [8, 9]. One common candidate coordination method is timer-based. In this method, each node waits for a preset time; if the node with the highest ranking fails to transmit the packet, the subsequent node sends it. In most existing work on OR, nodes are fixed and do not move within the network. The aim of opportunistic routing is to reduce the expected number of transmissions from source to destination, and this goal can be achieved only by selecting a suitable candidate set [10, 11]. Given the uncertainty and dynamism of wireless networks, intelligent mechanisms are needed to support effective candidate selection. Fuzzy logic is very useful for modelling uncertainty and imprecision in complex systems and provides robust decision-making capabilities in environments with ambiguous or incomplete information [12, 13].

This paper presents a novel forwarder selection method utilizing Fuzzy Reinforcement Learning in Opportunistic Routing (FRLOR). The fuzzy system inputs comprise geographical distance, link probability, and the number of neighbors for each candidate node. Reinforcement learning optimizes the candidate selection strategy based on network feedback. The paper is organized as follows: in Section 2, we review prior work; in Section 3, we describe the proposed algorithm, FRLOR. The proposed algorithm is compared with other candidate selection algorithms in Section 4, and Section 5 concludes.

2. Related Work

A core problem in Opportunistic Routing is determining the set of forwarders [1, 14]. One of the popular candidate selection algorithms is Extremely Opportunistic Routing (EXOR) [4]. EXOR applies the ETX metric to select and prioritize candidates. However, ETX is one of the simplest metrics and is not highly accurate. In [15], Opportunistic Any-Path Forwarding introduced a new metric, Expected Any-Path Transmission (EAX), which was more accurate than ETX. In [16], a greedy approach that leverages neighborhood information is employed to enhance network performance. This paper introduces a method to reduce beacon size by transmitting a subset of k neighboring data points within an LDACS time interval. The results demonstrate improved performance.

In [17], a reusable RL-based routing algorithm for SDN has been presented. The authors propose RLSR-Routing, which modifies SARSA and uses Segment Routing to aggregate actions and apply dual rewards. It demonstrates improved load balancing and faster convergence than traditional methods. In [18], an RL-enhanced Epidemic Routing protocol for OppNets has been introduced. The method integrates Q-learning and PPO to intelligently select forwarding neighbors, significantly reducing overhead and latency while maintaining high delivery rates in resource-constrained environments. In [19], an Energy-Efficient Mixture Opportunistic Routing (EMOR) for lunar surface networks has been proposed. Using an Actor-Critic architecture and a hybrid table-timer mechanism, it optimizes delay and energy consumption, greatly extending network lifetime and improving delivery ratio.

In [20], an Opportunistic Routing using Q-learning with Context Information (ORQLCI) has been presented. By integrating node meeting probability and buffer state into the Q-learning update mechanism, it achieves higher delivery rates and lower overhead than existing protocols such as Epidemic and Prophet. The authors of [21] present a two-layer model for opportunistic networks that integrates cybersecurity and blockchain concepts. The first layer introduces a fuzzy logic-based trust protocol (FT-OLSR) to isolate malicious nodes, while the second layer proposes a novel routing mechanism. Simulation results demonstrate that this approach achieves a higher delivery probability, a lower overhead ratio, and lower latency than established routing algorithms. The authors in [22] have presented SROR, a secure and reliable opportunistic routing protocol for Vehicular Ad Hoc Networks (VANETs). The method employs a deep reinforcement learning framework to select forwarding nodes based on parameters such as relative speed. It aims to improve packet delivery ratio and reduce delay while defending against blackhole and gray attacks. Results show that SROR outperforms existing protocols, achieving a higher delivery probability and lower latency. In [14], researchers introduced OptiE2ERL, a model that uses reinforcement learning to enhance energy efficiency in the Internet of Vehicles (IoV).

In [23], the authors proposed a new metric, TLG, based on link quality, geographic location, and remaining energy. In this algorithm, nodes with the best link quality, the shortest distance, and suitable energy efficiency are added to the candidate set. In [24], Position-based Opportunistic Routing (POR) is proposed for MANETs. It leverages geographic information and the broadcast nature of wireless channels to enable multiple forwarders. The protocol reduces control overhead and demonstrates high robustness, maintaining a delivery ratio of over 90% even when 50% of nodes maliciously drop packets. In [5], a Distance Progress based Opportunistic Routing (DPOR) is introduced. This hop-by-hop algorithm uses the Expected Distance Progress (EDP) metric for candidate selection. It achieves performance close to that of the optimal method but requires less information and has significantly lower execution time.

In [25], an Energy-Efficient Fuzzy Logic Prediction-based Opportunistic Routing (EEFLPOR) protocol is introduced for WSNs. This protocol applies a fuzzy inference system to forecast future node states, including remaining battery life, channel reliability, and transaction count, which inform its forwarder selection process.

3. Methods and Materials

In this section, the proposed 3. The Fuzzy Reinforcement Learning Based on Opportunistic Routing (FRLOR) method combines Fuzzy Logic Systems and Reinforcement Learning (RL) to support Opportunistic Routing (OR), aiming to find the optimal path and minimize the number of transmissions. This method employs the fuzzy system described in this section to address uncertainties in wireless networks (e.g., link probability and geographical distance). Building upon fuzzy OR, it first defines a candidate set for each node. Subsequently, it incorporates reinforcement learning to learn optimal candidate selection policies over time. In fact, a fuzzy system alone functions as a fixed guide that selects suitable candidates based on default rules, but it cannot adapt to sudden environmental changes (e.g., link outages or network traffic).

The proposed FRLOR framework achieves a tight coupling between the Mamdani fuzzy inference system (FIS) and the reinforcement learning (RL) paradigm. This integration enables real-time handling of wireless uncertainty via fuzzy logic while allowing long-term policy optimization through experience-driven learning. The core mechanism is to embed the FIS output as a core state feature within the Markov Decision Process (MDP), ensuring that RL decisions are informed by linguistically interpretable metrics derived from uncertain network parameters. With the inclusion of reinforcement learning, the system operates as a smart driver that learns from past mistakes and selects shorter, safer routes from the candidate set. The following explains the proposed method step by step, focusing first on the fuzzy system and then on reinforcement learning. Figure 1 illustrates the structure of the proposed mechanism in a wireless sensor network (WSN).

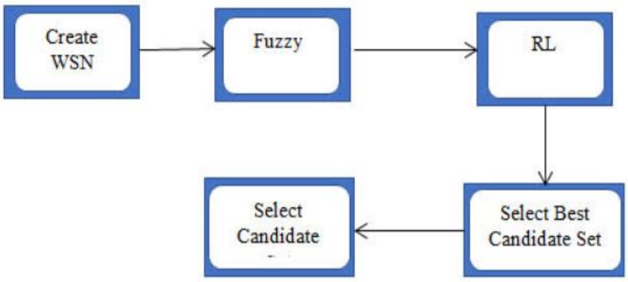

Figure 1. Proposed method

As shown in Figure 1, in the first stage, the network and environmental parameters are analyzed and provided as inputs to the fuzzy inference system. Three key parameters — geographical distance, the number of neighboring nodes, and link probability — are considered as fuzzy input variables. Based on these parameters, the fuzzy system evaluates the routing environment and generates a candidate set of potential forwarding nodes for opportunistic routing. The main objective is to determine the most efficient route for data transmission from the source to the destination.

3.1. Opportunistic routing (OR) system

In traditional routing, a predetermined number of nodes is used to forward packets, thereby effectively determining the best path. However, if one of the nodes leaves the path for any reason, the entire network becomes disconnected. Compared with traditional routing, OR uses a dynamic set of candidate nodes, allowing the source node to transmit data via multiple alternative paths rather than selecting a single fixed node for forwarding. Its primary objective is to minimize the total number of transmissions required for end-to-end packet delivery. In this approach, the transmitting node does not designate a specific next-hop address. Instead, it identifies a prioritized set of potential forwarders based on defined metrics and broadcasts the packet to this set. These candidates have priority. If one of the priority candidates deviates from the path for any reason, the next available candidate assumes control and forwards the packet. Designing an appropriate candidate selection algorithm reduces the time required to generate a candidate set and the number of transmissions between the source and destination nodes, thereby improving network performance (as illustrated in Figure 2).

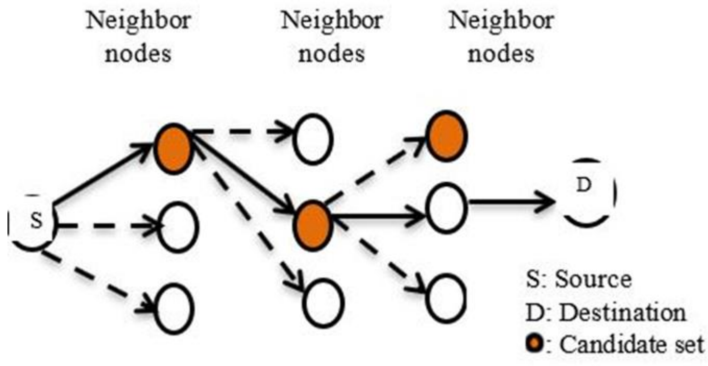

Figure 2. Opportunistic Routing

3.2. The structure of the fuzzy system

The architecture of the proposed approach utilizes a Fuzzy Logic Controller (FLC), as referenced in [26]. The general structure of this fuzzy system is delineated in the block diagram presented in Figure 3. Fuzzy systems are characterized by their inherent flexibility and adaptability, specifically the capacity to define and configure inference rules in a user-defined format. This attribute ensures that the FLC provides a robust and suitable metric for the methodical selection of candidates.

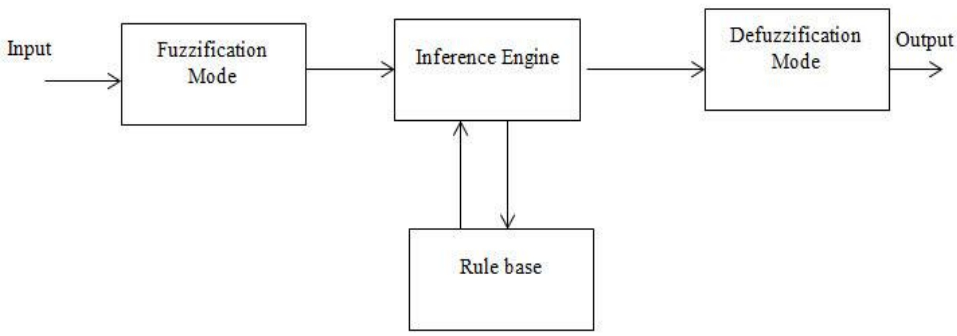

Figure 3. Diagram of Fuzzy System

A fuzzy logic system (FLS) comprises several distinct yet interconnected modules, as detailed in [26]:

- **Fuzzification Module:** This component is responsible for transforming crisp (non-fuzzy) input data into appropriate fuzzy sets. Essentially, it translates the inputs into the linguistic variables required by the subsequent inference engine. Common fuzzifier methods include the singleton, triangular, and Gaussian functions.
- **Inference Engine:** Acting as the computational core, the inference engine is a program designed to derive logical conclusions based on the established rule base. The most commonly used types of fuzzy inference systems are the Mamdani, Takagi–Sugeno, and Sugeno models.
- **Rule Base:** Considered the foundational element of any fuzzy system, the rule base consists of a set of 'IF–THEN' linguistic rules. The careful design and formulation of these fuzzy rules are critical determinants of the controller's effectiveness and operational success.
- **Defuzzification Module:** The final stage involves converting the fuzzy output set back into a single, crisp (usable) output value. Its primary function is to identify a single representative point that best encapsulates the fuzzy system's result. Widely used defuzzification techniques include the maximum membership, centroid, and weighted average methods.

In this paper, we use the Mamdani FIS due to its interpretability and centroid defuzzification, which yields a smooth output and is suitable for candidate ranking. Among defuzzification techniques, the centroid method (also known as the center-of-gravity method) is among the most widely used and accurate. In this method, the final crisp output represents the center of gravity of the aggregated output membership function [26]. The defuzzified output is calculated as follows:

$$f^* = \int f.\mu_c(f).df \Big/ \int \mu_c(f)df \quad (1)$$

Based on Eq. (1), f denotes the output universe of discourse, and $\mu_c(f)$ represents the aggregated membership function obtained after rule evaluation and fuzzy inference. The numerator $\int f.\mu_c(f)$ computes the weighted contribution of all possible output values, while the denominator $\int \mu_c(f)df$ normalizes this by the total area under the aggregated membership function. The resulting crisp value f^* is used as the final fuzzy output F_{out}.

3.3. Candidate selection in a fuzzy system

The inputs of the Mamdani fuzzy system include geographical distance, the local connection of each node, and the link delivery probability. In the candidate selection step, the geographical distance parameter is used as an important factor in selecting and creating candidate sets. Based on this parameter, the distance from all nodes to each other and the distance from all nodes to the destination node are calculated, and then the neighbors of each node are identified. The next parameter is the local connection of each node. Since the number of neighbors depends on geographical distance, a lower number of neighbors for a node does not necessarily indicate that its distance to the destination node is smaller than that of other neighboring nodes, but rather, a high number of neighbors indicates high connectivity, which is beneficial. The third parameter in this algorithm is the link delivery probability. Since low geographical distance alone is not always a suitable parameter for candidate selection, in addition to this parameter, we also use the link probability from each node to its neighboring nodes. The probability of a link is calculated based on the shadowing propagation model, as shown in Eq. (2). P(d) represents the received power at distance d [27]:

$$p(d) = 10\log\left(\text{RXThresh}.L(4\pi^2).d^\beta \Big/ P_tG_tG_r\lambda^2\right) + x_{d\beta} \quad (2)$$

Based on Eq. (2), the transmitted power is denoted by P_t, while G_t and G_r represent the gains of the transmitting and receiving antennas, respectively. The parameter L accounts for system losses, and λ corresponds to the signal wavelength (c/f, with c = 3 × 10^8 m/s). Additionally, β denotes the path loss exponent, and $x_{d\beta}$ is a Gaussian random variable with zero mean and standard deviation $\sigma_{d\beta}$. A packet is successfully received if the received power meets or exceeds the reception threshold, RXThresh. In our simulations, we set $\sigma_{d\beta}$=6dbs and β=2.7. The Network Simulator (NS-2) [28] was used to conduct the experiments, with the relevant simulation parameters summarized in Table 1. With a suitable design of fuzzy rules, the influence of all these parameters can be integrated together. The fuzzy system output is the Candidate Selection Metric, which is used to create a candidate set for each node. By comparing the fuzzy output on the neighbors, priority candidate set is created. The comparison process continues until the number of candidates equals the maximum candidate number (Max). Then, the priority candidate set is arranged for each node.

Table1. Default values for the shadowing propagation in NS-2.

Metric	Value
P_t	0.28183815 watt
RXThresh	3.652*10^{-10} watt
G_t, L, G_r	1.0
f	914*10^6 H

The membership functions for the inputs are defined as follows:

- Number of neighbors: Triangular functions over [0, 20].
  - Small: (0, 0, 5)
  - Average: (3, 7, 11)
  - Large: (9, 20, 20)

- Distance: Triangular over [0, 1200] meters.
  - Low: (0, 0, 450)
  - Average: (350, 750, 1000)
  - High: (900, 1200, 1200)
- Link Probability: Triangular over [0, 1].
  - Weak: (0, 0, 0.4)
  - Medium: (0.3, 0.5, 0.7)
  - Strong: (0.6, 1, 1)
- Output (Candidate Selection Metric): Triangular over [0, 1].
  - Very Bad: (0, 0, 0.2)
  - Bad: (0.15, 0.3, 0.45)
  - Average: (0.4, 0.5, 0.6)
  - Good: (0.55, 0.7, 0.85)
  - Very Good: (0.8, 1, 1)

These inputs are combined through fuzzy rules to compute the candidate selection metric. The system generates 27 distinct rules, covering all possible combinations of the three inputs. Each rule provides a unique mapping of input parameters to the output.

Figure 4 displays the membership function of the link probability parameter. In this scheme, the interval of link probability is considered within [0, 1].

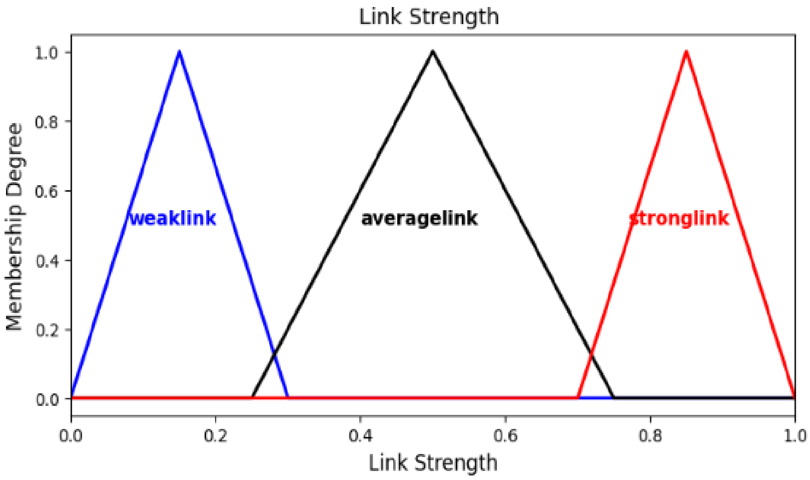

Figure 4. Membership function for link.

As depicted in Figure 5, the defuzzification step starts with the interpretation of fuzzy outputs resulting from applying inference rules to the input variables. These interpreted outputs are assigned one of five membership functions: “very bad”, “bad”, “average”, “good” and “very good”. The output of the fuzzy system refers to the candidate selection metric.

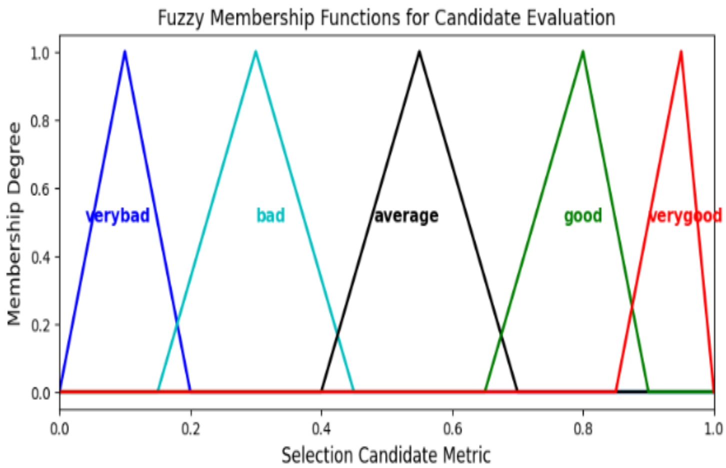

Figure 5. Membership Function for output

In this paper, the defined Fuzzy system has three inputs geographical distance, the number of neighbors for each node, and the link probability. The system output is the selection of candidate's metric according to the established rules that are shown in Table 2. Table 2 will be updated to include explicit weights (default uniform weight of 1.0 for all 27 rules, as no differential weighting was used).

Table 2. Rule Base of FRLOR.

Number of Neighbors	Distance	Link Probability	Output of Fuzzy system
Small	Low distance	Weak link	Average
Small	Low distance	Average link	Very good
Small	Low distance	Strong link	Very good
Small	Average distance	Weak link	Bad
Small	Average distance	Average link	Good

ISSN 3092-7552

Small	Average distance	Strong link	Good
Small	High distance	Weak link	Very bad
Small	High distance	Average link	Bad
Small	High distance	Strong link	Average
Average	Low distance	Weak link	Average
Average	Low distance	Average link	Good
Average	Low distance	Strong link	Very good
Average	Average distance	Weak link	Bad
Average	Average distance	Average link	Average
Average	Average distance	Strong link	Good
Average	High distance	Weak link	Very bad
Average	High distance	Average link	Bad
Average	High distance	Strong link	Average
Large	Low distance	Weak link	Bad
Large	High distance	Average link	Average
Large	High distance	Strong link	Good
Large	Average distance	Weak link	Very bad
Large	Average distance	Average link	Bad
Large	Average distance	Strong link	Average
Large	High distance	Weak link	Very bad
Large	High distance	Average link	Bad
Large	High distance	Strong link	Bad

3.4. Reinforcement learning

To overcome the limitations of the static nature of the fuzzy system—which, despite its accuracy in uncertainty management, cannot respond to dynamic network changes (such as link outages or traffic changes)—a reinforcement learning framework is integrated into the fuzzy system. This combination transforms the candidate selection policy from a rule-based approach to an adaptive and experience-optimized policy. The main objective is to minimize the number of expected transmissions.

The proposed FRLOR framework establishes an integrated, theory-based link between the Mamdani Fuzzy Inference System (FIS) and a reinforcement learning algorithm. This link is achieved by including the defuzzified output of the fuzzy system — denoted by F_{out} € [0, 1] — as a key component in the state vector of the Markov Decision Process (MDP). This approach combines linguistic rule-based guidance with experience-based policy optimization.

Specifically, the fuzzy system processes the numerical inputs (geographical distance, number of neighbors, and link delivery probability) through the steps of fuzzification, rule inference (using the 27 IF-THEN rules listed in Table 2 with equal weights), and defuzzification by the center of gravity method to produce the candidate selection criterion F_{out}. This criterion represents the suitability of each node as a forwarder in the form of a normalized and linguistically interpretable score (from “very bad” to “very good”), and prioritizes nodes that are closer to the destination, have stable links, and have balanced neighborhood density.

The steps of implementing the reinforcement learning algorithm are detailed below:

Step 0: Define Markov Decision Process (MDP)

MDP forms the RL decision framework and is defined as (S, A, T, R, γ):

S (states): The set of environmental states, including fuzzy and network information.

A (actions): A finite set of actions, representing the set of selections of a candidate.

T (s a, s') (transfer function): The probability of transitioning from state S to S' with action a, which is extracted from the shadowing model.

R (s, a) (reward): the reward function, representing the immediate feedback to the agent for selecting action a in state S. The immediate reward for selecting the candidate, which encourages the shortest path.

γ (discount factor): the discount factor, which controls the importance of future rewards (γ € [0,1]). We consider it (γ=0.9) in this paper.

This MDP formulation ensures the Markov property. Future states and rewards depend only on the current state and action, handling uncertainties in wireless Networks via fuzzy integration. Unlike traditional OR algorithms, this MDP enables RL to learn optimal policies for dynamic scenarios.

Step 1: State Definition

The initial step in the FRLOR algorithm, within the MDP framework, involves defining the state s € S. The state is formulated based on candidate selection (from fuzzy logic) and routing conditions to support minimizing the number of expected transmissions. Accordingly, the state (s_t) is defined in Eq. (3):

$$s_t = \{F_{out}, D_g, N_n, L_p, T_e, C_a\} \quad (3)$$

Where:

F_{out}: output of the fuzzy system, which is calculated from D_g, N_n, L_p .
D_g: The average geographical distance between each node and the candidate set.
N_n: Number of candidate neighbors.
L_p : Link probability between the current node and its neighbors.
T_e: Expected number of transmissions.
C_a: C_a is candidate capacity.
This state vector uses the fuzzy output (F_{out}) to handle uncertain information and is sufficient for shortest-path decision-making.
This mode uses the fuzzy output (F_{out}) to handle uncertain information, and is sufficient for shortest-path decision-making.

Step 2: Action Definition

In the second step, the agent can take discrete actions to modify the fuzzy system's parameters or routing strategy.

Defining actions, a € A within the MDP, involves the agent's decision to select a candidate that minimizes the transmission number. Specifically, a_t€ {1, 2, 3..., n} selects from up to n, informed by fuzzy metrics (Table 2). The action triggers transitions via T (s, a, s').

Step 3: Reward Definition

The reward function serves to evaluate the efficacy of the agent's decision in terms of network efficiency and Quality of Service (QoS). The multi-objective reward function used in FRLOR is defined in Eq. (4):

$$r_t = \alpha\left(1 - {}^{D_g}/_{D}\right) + \beta\left({}^{N_n}/_{N_{max}}\right) + \omega\left(1 - {}^{T_e}/_{T_{min}}\right) \quad (4)$$

Base on Eq. (4), D is the network diameter (1200 m), N_{max} is the maximum number of neighbors, and T_{min} represents the minimal number of transmissions. All parameters α, β, ω >0 is weighting parameters.

This reward encourages:

- ✓ selecting nodes closer to the destination.
- ✓ choosing neighborhoods with higher availability.
- ✓ minimizing the expected number of transmissions.

Step 4: Q-Learning Update Rule

The Q-table maintains a value for all valid (state, action) pairs, as shown in Eq. (5) [16].

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta[r_t + \omega \max Q(s_{t+1}, a') - Q(s_t, a_t)] \quad (5)$$

Based on Eq. (5), where η is the learning rate and ω =0.9 is the discount factor, this update allows FRLOR to gradually learn which candidates produce the most efficient long-term routing paths.

Step 5: Action Selection

To balance exploration and exploitation, the action for each state is selected according to the ε-greedy strategy. Based on Eq. (6), the agent chooses a random candidate with probability ε, while with probability 1−ε it selects the action that maximizes the current Q-value:

$$a_t = \begin{cases} \text{random candidate}, & \text{with probability } \varepsilon \\ \arg\max Q(s_t, a_t), & \text{with probability } 1 - \varepsilon \end{cases} \quad (6)$$

Table 3 summarizes the hyperparameters used in the Reinforcement Learning module of the proposed model. Since the RL component is based on a lightweight tabular Q-Learning method, only the parameters relevant to Q-value updates and the ε-greedy exploration strategy are included. The action selection process follows Eq. (6), while the Q-value update rule is given in Eq. (5).

Where:

Episode length: 100 steps (one full source-to-sink transmission).
Exploration: ε-greedy (ε=0.1, decay 0.99/episode).
Learning rate (η): 0.1 for Q-value updates.
Number of episodes: 1000 (convergence when value loss < 0.01 for 50 episodes).

Stopping criteria: Early stopping if no improvement in average reward over 100 episodes.

Discount factor (γ): 0.9.

Table 3: Hyperparameters for RL Training

Parameter	Value
Episodes	1000
Episode Length	100 steps
Exploration	ε-greedy (ε=0.1, decay=0.99)
α, β, ω	0.5, 0.3,0.2
Learning Rate	0.1
Discount (γ)	0.9
Convergence	Q-values stabilize for 50 consecutive episodes.

The selected hyperparameters—such as the ε-greedy exploration settings and the learning rates—were empirically tuned through iterative testing, although these parameters can also be systematically optimized using established hyperparameter search techniques such as random search or grid search.

The following section presents the training procedure of the proposed FRLOR protocol, where reinforcement learning is integrated with the fuzzy-based candidate selection mechanism.

Algorithm 1: FRLOR Training Procedure
Input: environment (NS-2 wrapper), fuzzy system, hyperparameters
Initialize Q-table Q (s, a) ← 0 for all discretized states and actions
for seed = 1 to 100 do
    set_random_seed(seed)
    for episode = 1 to MaxEpisodes do
        s ← env.reset()
        discretize_state(s)
        for t = 1 to MaxEpisodeLen do
            With probability ε, select a random action a
            otherwise select a = argmax Q(s, a)
            Execute action based on Eq. 6
            Q-value update rule based on Eq. 5
            If done, then break
        end for
        evaluate performance metrics (PDR, delay, energy)
        if early_stop_condition then break
    end for
end for
Output: optimized Q-table Q*

4. Results

To rigorously evaluate the FRLOR algorithm, a two-phase hybrid simulation framework was developed that integrates offline computation in Python (for both fuzzy candidate ranking and reinforcement learning policy optimization) with online packet-level routing in NS-2.34. This methodology ensures high-fidelity modeling of both intelligent decision-making and realistic wireless dynamics. In this section, we compare and examine the algorithms under study with the FRLOR algorithm. This algorithm is compared with routing algorithms and protocols, including POR [24], DPOR [5], and EEFLPOR [25], to demonstrate its strong performance.

1. Defined Metrics

To evaluate the performance of the algorithms, NS-2.34 [28] was used. In our simulation, we placed nodes in a 1200×1200 area, similar to the Python environment, obtained the candidate set for each node using Python, and then applied the candidates in the NS-2 environment. The network topology remained the same in both environments. A parameter analysis is first presented to demonstrate the effects of different parameters on the proposed protocol. As noted earlier, the OR protocol comprises two key phases: candidate selection/prioritization and coordination. In order to introduce the set of candidates for each node, the candidate set is included in the header of each packet, and we used the time-based coordination method to coordinate between candidates in all algorithms. In this method, a waiting time interval is considered for each candidate, which is calculated using Eq. (7) [29]:

$$T_i = (i - 1) + T_{Default} \qquad (7)$$

Based on Eq. (7), $T_{Default}$ is the default waiting time, which is considered to be 50 ms here, i refers to the candidate number, with values (1, 2, 3..., n). The highest-priority candidate is the first to submit the packet, and there is no waiting time for this candidate.

To evaluate the performance of the proposed algorithm, we study the algorithm in terms of the Expected Number of Transmissions (ENT) from source to destination. The formula in [15] calculates the ENT. We also consider Execution Time, Average End-to-End Delay (E2E Delay), Packet Delivery Ratio (PDR), and Energy Efficiency as metrics for evaluating protocol performance.

Average End-to-End Delay: This parameter indicates the packet transfer time between the source and destination and is actually the sum of all types of delay times from the start of packet transmission to the end of the operation [29]. This parameter is calculated using the formula given in [30].

Packet Delivery Ratio (PDR): This parameter calculates the network performance based on the ratio between the number of packets successfully received by the destination and the number of packets sent by the source. The PDR is calculated from the formula described in [30].

Energy Efficiency: The ratio of the number of packets successfully received by the sink or destination to the product of the energy consumed by the network to forward a packet toward the destination and the number of deployed nodes [31-34].

2. Evaluation metrics

A. Evaluate the expected number of transmissions

In order to evaluate this parameter, we distributed the nodes in a grid with dimensions of 1200 × 1200. Since the main objective of opportunistic routing algorithms is to reduce the ENT from source to destination, we considered four different scenarios. First, the algorithm was tested with different numbers of nodes (200 < N < 400) that were randomly placed in the environment, and then the algorithm was examined for different sizes of the candidate set (candidate set size = 2, 3, 4, 5).

In Figures 6 and 7, the number of nodes is fixed (N = 200 and N = 400, respectively), and different numbers of candidates are considered.

In Figure 6, it can be seen that the POR algorithm shows the worst performance than the other algorithms because this algorithm only considers the geographical distance, while the FRLOR algorithm optimizes the minimum number of transmissions better than the others. Of course, it should also be noted that increasing the number of candidates requires more coordination between them, which may cause higher overhead. If there is not perfect coordination between them, it may lead to duplicate transmissions by other candidates. Since all candidates must be listed in the header, increasing the number of candidates also increases the size of the packet header.

In Figure 7, the results were calculated with N =400. In this case, the algorithm's behavior with a large number of candidates is similar to that of the EEFLPOR and DPOR algorithms, with increasing overhead.

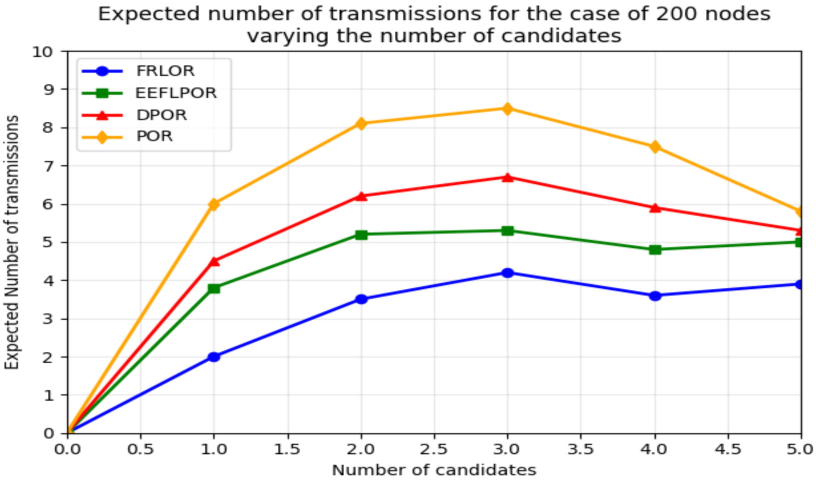

Figure 6. Evaluate the ENT with a variety of candidates. The case of 200 nodes, varying the number of candidates

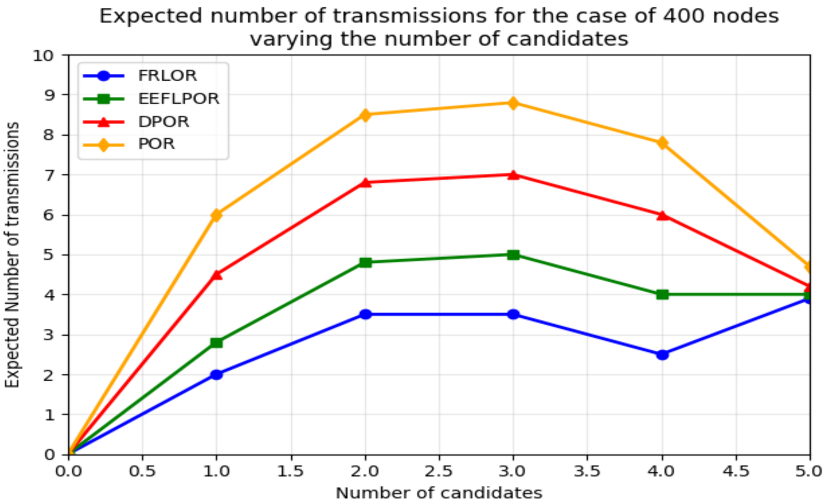

Figure 7. The ENT for the case of 400 nodes, varying the number of candidates

The performance of the proposed FRLOR protocol is evaluated in two network configurations, each comprising 200 or 400 nodes. Figures 6 and 7 depict the ENT as the number of forwarding candidates increases. In both scenarios, FRLOR consistently achieves the lowest ENT compared to EEFLPOR, DPOR, and the baseline POR scheme. In the 200-node network, FRLOR reduces the average ENT by approximately 20% relative to EEFLPOR, 35% relative to DPOR, and 48% compared to POR. In the 400-node scenario, FRLOR achieves reductions of 27%, 44%, and 56%, respectively. This superior performance is attributed to FRLOR's hybrid fuzzy–reinforcement learning mechanism, which dynamically selects optimal forwarding candidates and minimizes redundant transmissions. Overall, the results demonstrate that FRLOR maintains high efficiency and scalability, significantly reducing transmission overhead and improving routing performance in both small and large networks

.

B. Execution Time

This section evaluates the computational overhead required to construct the forwarder set. The DPOR algorithm incurs the greatest processing time, approximately 95 minutes. Expanding the forwarder list size directly raises the computational burden. Our proposed method achieves lower runtime than DPOR. While the POR algorithm is computationally faster than DPOR, it yields inferior performance in terms of the expected number of transmissions required for delivery. The execution time outcomes are depicted in Figure 8.

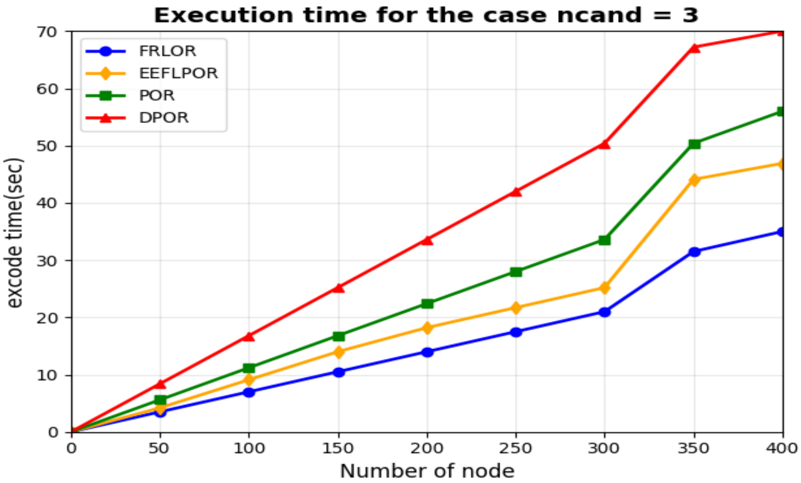

Figure 8. The execution time for the number of candidates=3

Figure 8 illustrates the execution time of the evaluated routing schemes for a fixed candidate size (n_cand = 3) as the network scales from 50 to 400 nodes. The proposed FRLOR consistently achieves the lowest execution time, indicating superior computational efficiency compared to EEFLPOR, POR, and DPOR. While EEFLPOR shows moderate overhead, POR and, in particular, DPOR exhibit rapid increases in processing time as the network becomes denser. On average, FRLOR reduces execution time by approximately 22% relative to EEFLPOR, 36% compared to POR, and 54% compared to DPOR. These results confirm the scalability and low complexity of FRLOR, making it suitable for real-time, large-scale opportunistic routing scenarios.

C. Packet Delivery Ratio (PDR)

In Figure 9, algorithm FRLOR achieves the highest packet delivery rate among the algorithms. In addition to considering distance, this algorithm uses link probabilities and each node's neighbor density to select and construct a candidate set. The appropriate and accurate design of fuzzy rules in the fuzzy system, together with the optimization of the candidate set by the reinforcement learning algorithm, ensures that all candidate selection paradigms work together effectively, thereby creating a high-quality candidate set. Consequently, the packet delivery rate and reliability of this algorithm are higher than those of other algorithms.

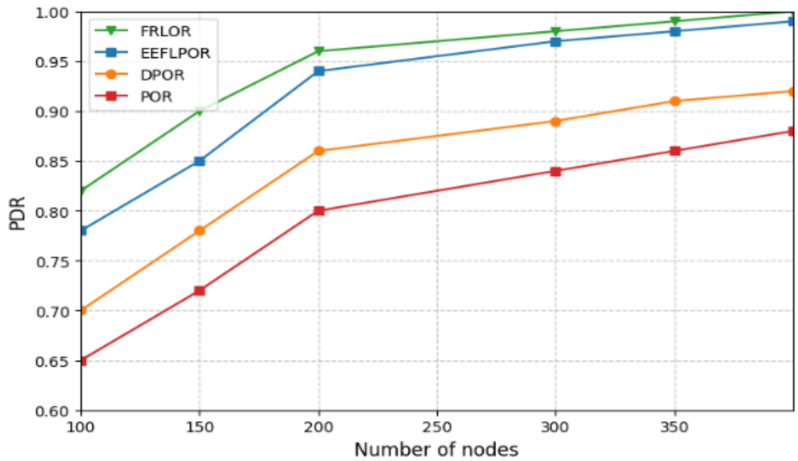

Figure 9. Comparisons of PDR with different nodes

Experimental results in Figure 9 demonstrate the clear superiority of FRLOR over EEFLPOR, DPOR, and POR across all network densities, assuming the number of candidates considered is 5. As shown in Figure 8, FRLOR consistently attains the highest PDR, achieving near-perfect delivery in dense networks. This improvement stems from its adaptive forwarder selection and efficient link-quality estimation, which enable robust performance under both sparse and congested conditions. In contrast, the baseline methods exhibit slower scalability and reduced reliability, confirming FRLOR's effectiveness as a more stable and efficient opportunistic routing solution.

D. End-to-End Delay

ISSN 3092-7552

This parameter reports the packet transfer time between the source and destination and computes the total delay from the start of packet transmission to the end of the operation. The goal is to minimize this parameter in the network. In evaluating this parameter, we considered varying numbers of nodes while keeping the number of candidates constant. When a source node sends a packet to the destination, the appropriate candidate set is identified, and the best path for the packet is determined. The graph of this parameter is shown in Figure 10. In the FRLOR algorithm, selecting an appropriate candidate set enables the destination to receive the transmitted packet more quickly. As shown, the E2E Delay in the FRLOR algorithm is lower than that of other algorithms and yields better performance.

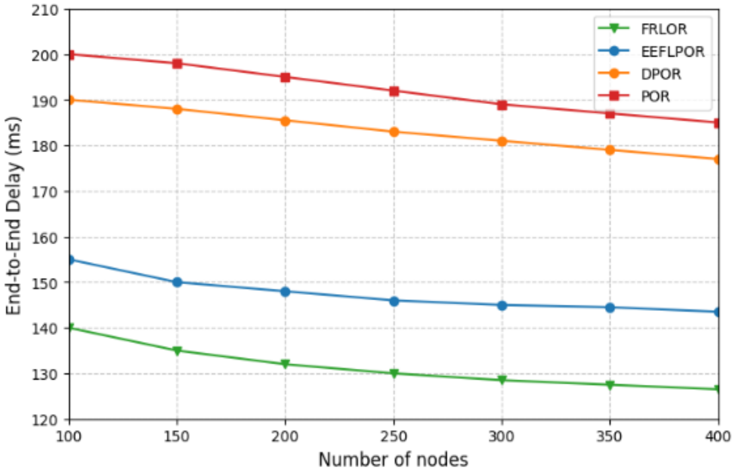

Figure 10. Comparisons of End-End Delay with different nodes

Figure 10 presents the E2E Delay obtained by FRLOR, EEFLPOR, DPOR, and POR under different network densities. The results indicate that FRLOR consistently achieves the lowest delay across all scenarios. As the number of nodes increases, FRLOR benefits from its adaptive forwarder selection and efficient path coordination, reducing the delay from approximately 140 ms at 100 nodes to nearly 127 ms at 400 nodes. EEFLPOR shows the second-best performance. DPOR experiences comparatively higher delay and demonstrates limited scalability, while POR yields the poorest performance with delays exceeding 185 ms even in dense networks. Overall, the delay evaluation confirms that FRLOR substantially improves transmission efficiency relative to existing opportunistic routing schemes.

E. Energy Consumption

The energy consumption performance is similar across different parameter values and is better than that of others, with the lowest energy consumption. The graph of this parameter is shown in Figure 11. This is due to the RL-based framework's ability to learn from the environment and adapt to topological change.

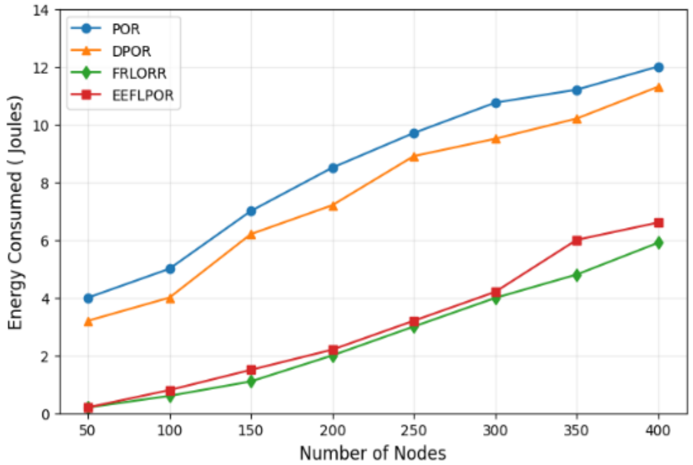

Figure 11. Comparisons of energy with different algorithms

In Figure 11, the proposed FRLORR algorithm demonstrates a clear performance advantage over all benchmark schemes. Owing to its intelligent fuzzy–RL decision-making mechanism, FRLORR achieves significant energy-efficiency improvements, reducing energy consumption by up to 47.4% compared to the traditional DPOR method and by 50.0% relative to the baseline POR scheme. When compared with advanced fuzzy-based approaches such as EEFLPOR, FRLOR maintains superior performance.

5. Conclusion

This paper presents three parameters, namely the number of neighboring nodes, link probability, and geographical distance, as inputs to the fuzzy system. The proposed algorithm uses these inputs to select the candidate set in Opportunistic Routing (OR). Subsequently, Reinforcement Learning (RL) in FRLOR improves performance by optimizing the candidate selection policy. To evaluate the proposed method, we considered the following criteria: Expected Number of Transmissions (ENT), Energy Consumption, End-to-End Delay (E2E Delay), Packet Delivery Ratio (PDR), and Execution Time. Overall, FRLOR not only reduces the number of transmissions and execution time but also provides a robust, data-driven solution for opportunistic routing in wireless networks through continuous learning. The future development of this research can be extended in multiple significant ways. While the present study demonstrates the efficacy of FRLOR in reducing transmissions and enhancing routing stability, practical validation on real medical IoT platforms is required to substantiate these findings with real physiological data traffic patterns. Additionally, forthcoming research should incorporate robust security measures to ensure the confidentiality and integrity of sensitive healthcare information shared via opportunistic channels. An additional suggestion is the automated adjustment of RL hyperparameters via structured search methods such as grid search or randomized search, which can improve efficacy across changing healthcare settings. By incorporating real-world implementation, security improvements, and automated hyperparameter tuning, the FRLOR framework can become a more reliable and scalable option for future medical IoT systems.

6. Acknowledgements

AI-Assisted Technology Declaration:

During the preparation of this manuscript, artificial intelligence tools were used as research assistants solely for translation and language polishing purposes. All scientific content, analyses, interpretations, and conclusions presented in this paper were entirely developed by the author(s).

7. Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

8. Competing Interests Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

9. References

[1] Chakchouk, N. (2015). A survey on opportunistic routing in wireless communication networks. IEEE Communications Surveys & Tutorials, *17*(4), 2214–2241. DOI:10.1109/COMST.2015.2411335

[2] Almuzaini, K. K., Joshi, S., Ojo, S., Aggarwal, M., Suman, P., Pareek, P. K., & Shukla, P. K. (2024). Optimization of the operational state's routing for mobile wireless sensor networks. Wireless Networks, *30*(6), 5247–5261. DOI:10.1007/s11276-023-03246-3

[3] Boukerche, A., & Darehshoorzadeh, A. (2015). Opportunistic routing in wireless networks: Models, algorithms, and classifications. ACM Computing Surveys, *47*(2), 1–36. DOI:10.1145/2635675

[4] Biswas, S., & Morris, R. (2005). ExOR: Opportunistic multi-hop routing for wireless networks. In Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '05) (pp. 133–144). ACM. DOI:10.1145/1080091.1080108

[5] Darehshoorzadeh, A., & Cerda-Alabern, L. (2012). Distance progress based opportunistic routing for wireless mesh networks. In 2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC) (pp. 179–184). IEEE. DOI:10.1109/IWCMC.2012.6314199

[6] Darehshoorzadeh, A., & Boukerche, A. (2014). Opportunistic routing protocols in wireless networks: A performance comparison. In 2014 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 2504–2509). IEEE. DOI:10.1109/WCNC.2014.6952782

[7] Darehshoorzadeh, A., Cerdà-Alabern, L., & Pla, V. (2013). Opportunistic routing in wireless mesh networks. In I. Woungang, S. K. Dhurandher, A. Anpalagan, & A. V. Vasilakos (Eds.), Routing in Opportunistic Networks (pp. 289–330). Springer. DOI:10.1007/978-1-4614-3514-3_11

[8] Das, S., Panda, K. G., & NaderiAlizadeh, N. (2025). Opportunistic routing in wireless communications via learnable state-augmented policies. arXiv preprint arXiv:2503.03736. DOI:10.48550/arXiv.2503.03736

[9] Lata, A. A., Kang, M., & Shin, S. (2025). FCM-OR: A local density-aware opportunistic routing protocol for energy-efficient wireless sensor networks. Electronics, *14*(9), 1841. DOI:10.3390/electronics14091841

[10] Hashemi, M., & Moghim, N. (2023). An efficient multicast multi-rate reinforcement learning based opportunistic routing algorithm. Multimedia Tools and Applications, *82*(17), 26613–26630. DOI:10.1007/s11042-023-14645-1

[11] Fuste-Vilella, D., Garcia-Vidal, J., & Morillo-Pozo, J. D. (2008). Cooperative forwarding in IEEE 802.11-based MANETs. In 2008 1st IFIP Wireless Days (pp. 1–5). IEEE. DOI:10.1109/WD.2008.4812845

[12] Ghasemi, J., Ghaderi, R., Mollaei, M. K., & Hojjatoleslami, S. A. (2013). A novel fuzzy Dempster–Shafer inference system for brain MRI segmentation. Information Sciences, *223*, 205–220. DOI:10.1016/j.ins.2012.08.026

[13] Ghasemi, J., Ghaderi, R., Mollaei, M. R. K., & Hojjatoleslami, A. (2011). Separation of brain tissues in MRI based on multi-dimensional FCM and spatial information. In 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) (Vol. 1, pp. 247–251). IEEE. DOI:10.1109/FSKD.2011.6019589

[14] Hussain, Q., Noor, A. S. M., Qureshi, M. M., Li, J., Rahman, A. U., Bakry, A., Mahmood, T., & Rehman, A. (2025). Reinforcement learning based route optimization model to enhance energy efficiency in internet of vehicles. Scientific Reports, *15*(1), 3113. DOI:10.1038/s41598-025-86608-5

[15] Zhong, Z., Wang, J., Nelakuditi, S., & Lu, G.-H. (2006). On selection of candidates for opportunistic anypath forwarding. ACM SIGMOBILE Mobile Computing and Communications Review, *10*(4), 1–2. DOI:10.1145/1215976.1215978

[16] Ahmed, M., Fuger, K., Kuladinithi, K., & Timm-Giel, A. (2024). Enhancing geographic greedy routing in sparse LDACS air-to-air networks through k-hop neighborhood exploitation. In 2024 IEEE 49th Conference on Local Computer Networks (LCN) (pp. 1–7). IEEE. DOI:10.1109/LCN60385.2024.10639650

[17] Wumian, W., Saha, S., Haque, A., & Sidebottom, G. (2024). Intelligent routing algorithm over SDN: Reusable reinforcement learning approach. arXiv preprint arXiv:2409.15226. DOI:10.48550/arXiv.2409.15226

[18] Do, Q. H., Abreu, T., Kusi, B., Diop, N. C., & Souihi, S. (2025). An efficient epidemic routing protocol with reinforcement learning algorithm in opportunistic networks. In *ICC 2025 - IEEE International Conference on Communications* (pp. 3497–3502). IEEE. DOI:10.1109/ICC52391.2025.11161186

[19] Wang, Y., Qu, Z., Zhao, Z., Cao, X., Liu, Y., & Quek, T. Q. S. (2025). EMOR: Energy-efficient mixture opportunistic routing based on reinforcement learning for lunar surface ad-hoc networks. IEEE Transactions on Communications, *73*(6), 4307–4320. DOI:10.1109/TCOMM.2024.3490499

[20] Liu, X., Cui, J., Seah, W. K. G., Xu, X., Xu, G., & Wu, C. (2025). Opportunistic routing using Q-learning with context information. In International Computing and Combinatorics Conference (COCOON 2025) (pp. 355–367). Springer. DOI:10.1007/978-981-96-1093-8_30

[21] Khalil, A., & Zeddini, B. (2024). A secure opportunistic network with efficient routing for enhanced efficiency and sustainability. Future Internet, *16*(2), 56. DOI:10.3390/fi16020056

[22] Xu, H., & Wang, Y. (2024). SROR: A secure and reliable opportunistic routing for VANETs. Vehicles, *6*(4), 1730–1751. DOI:10.3390/vehicles6040084

[23] Zhao, Z., Rosario, D., Braun, T., Cerqueira, E., Xu, H., & Huang, L. (2013). Topology and link quality-aware geographical opportunistic routing in wireless ad-hoc networks. In 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC) (pp. 1522–1527). IEEE. DOI:10.1109/IWCMC.2013.6583782

[24] Yang, S., Zhong, F., Yeo, C. K., Lee, B. S., & Boleng, J. (2009). Position based opportunistic routing for robust data delivery in MANETs. In *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference* (pp. 1–6). IEEE. DOI:10.1109/GLOCOM.2009.5425351

[25] Nagadivya, S., & Manoharan, R. (2023). Energy efficient fuzzy logic prediction-based opportunistic routing protocol (EEFLPOR) for wireless sensor networks. Peer-to-Peer Networking and Applications, *16*(5), 2089–2102. DOI:10.1007/s12083-023-01516-7

[26] Belman-Flores, J. M., Rodríguez-Valderrama, D. A., Ledesma, S., García-Pabón, J. J., Hernández, D., & Pardo-Cely, D. M. (2022). A review on applications of fuzzy logic control for refrigeration systems. Applied Sciences, *12*(3), 1302. DOI:10.3390/app12031302

[27] Cerda-Alabern, L., Darehshoorzadeh, A., & Pla, V. (2010). On the maximum performance in opportunistic routing. In 2010 IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM) (pp. 1–8). IEEE. DOI:10.1109/WOWMOM.2010.5534897

[28] Issariyakul, T., & Hossain, E. (2009). Introduction to network simulator 2 (NS2). In Introduction to Network Simulator NS2 (pp. 1–18). Springer. DOI:10.1007/978-0-387-71760-9_2

[29] Fateminasab, S. S., Evaznia, N., Memarian, S., Romero-Ternero, M. D. C., Miró-Amarante, G., & Tabbakh, S. R. K. (2025). An efficient blockchain-based resource allocation and secure data storage model using Fire Hawk Optimization and entropy in health tourism. Journal of Cloud Computing, *14*(1), 63. DOI:10.1186/s13677-025-00792-3

[30] Khan, M. F., Felemban, E. A., Qaisar, S., & Ali, S. (2013). Performance analysis on packet delivery ratio and end-to-end delay of different network topologies in wireless sensor networks (WSNs). In *2013 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)* (pp. 324–329). IEEE. DOI:10.1109/MSN.2013.74

[31] Shen, Z., Yin, H., Jing, L., Liang, Y., & Wang, J. (2022). A cooperative routing protocol based on Q-learning for underwater optical-acoustic hybrid wireless sensor networks. IEEE Sensors Journal, *22*(1), 1041–1050. DOI:10.1109/JSEN.2021.3128594

[32] Evaznia, N., & Ebrahimi, R. (2023). Providing a Solution for Optimal Management of Resources using the Multi-objective Crow Search Algorithm in Cloud Data Centers. In 2023 9th International Conference on Web Research (ICWR) (pp. 179–184). IEEE. DOI:10.1109/ICWR57742.2023.10139192

[33] Bahrepour, D., Evaznia, N., & Khodabakhshi, T. (2024). A New Resource Allocation Method Based on PSO in Cloud Computing. International Journal of Web Research, *7*(2), 13–21. DOI:10.22133/ijwr.2024.457539.1216

[34] Evaznia, N., Ebrahimi, R., & Bahrepour, D. (2025). An Energy-Aware Approach to Virtual Machine Consolidation Using Classification and the Dragonfly Algorithm in Cloud Data Centers. Journal of Information Systems and Telecommunication (JIST), *4*(48), 280. DOI:10.61186/jist.48021.12.48.280