International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

AN INTELLIGENT FUZZY REINFORCEMENT LEARNING-BASED ROUTING ALGORITHM FOR LATENCY-ENERGY TRADE-OFF IN IOT-ENABLED SDN

Arkan Ali Hadi[1], Mohsen Nikray*[2]

[1,2]Computer Engineering and Information Technology Department, University of Qom, Qom, Iran.

Emails: arkan.almansoori@gmail.com[1], m.nickray@ece.ut.ac.ir[2]

Abstract

In this paper, we present an Intelligent Fuzzy Re-inforcement Learning Based Routing Technique Algorithm (IFRLTRA) for Router Optimization in IoT and SDN Environment. The approach integrates fuzzy reasoning, reinforcement learning (RL), and dynamic resource allocation in the latency-driven feedback loop to achieve low-latency, high-bandwidth-utilization, load-balanced network. In addition, the utilization of these approaches simultaneously provides IFRLTRA with efficient network operation to maintain real time applications like video conferencing and other IoT services under their desired QoS parameters. The proposed IFRLTRA was evaluated against state-of-the-art routing algorithms through extensive simulations, showing that it outperforms them in terms of the rate of admission, path length, e2el latency, traffic utilization and load. Due to its dynamic traffic rerouting based on current network conditions in real-time, this algorithm allocates resources efficiently, and it helps the system to reduce congestion and latency as well as energy consumption. The paper presents potential future enhancements: purchase of edge computing to reduce latency even more and improved resource management as well as security mechanisms for applications with high data protection requirements. Above all, the IFRLTRA is considered as a more robust and practical scheme for routing in SDN-based IoT environment and thereby can be an appropriate candidate for improving the quality of real-time application on the modern IoT network which benefits from ensuring reliability and enhanced performance of network traffic..

Keywords Intelligent Routing, Fuzzy Logic, Reinforcement Learning, Software-Defined Networking (SDN), IoT-enabled Networks, Latency Optimization, Quality of Service (QoS).

1.Introduction

The growing popularity of the Internet of Things (IoT) has led to an increasing number of smart devices, sensors, and applications involved in kith and kin communications and real-time data processing [1]. In the background of rapidly developing IoT technologies, how to practically achieve efficient Internet access and data flow control for those systems is a critical issue [2]. To address this challenge, SDN emerged as a potential architecture for the IoT network management and control [3]. In addition, the SDN model has centralized

Received: August 04, 2025

Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

control through flexibility and adaptability in managing network resources and traffic [4]. Regarding the difficulties of IoT networks, including latency control, bandwidth allocation and load distribution, if high performance and resource utilization are to be obtained they all need to be solved effectively [5]

Apart from that all the real-time application like video conferencing, autonomous vehicle and manufacturing automation make latency one of the major challenges while dealing with IoT enabled SDN[6]. But, achieving the tradeoff between both a low latency requirement for orders of application and require enough bandwidth from other is challenging in large-scale and dynamic IoT networks. The consumption of the bandwidth, however, is also a crucial mean to ensure an efficient utilization of the network resources [7-8]. If bandwidth is used inefficiently, congestion will be caused (which would lead to bad performance) without satisfying the QoS requirements [9].

The rapid deployment of the Internet of Things (IoT) has increased the intricacy [10], magnitude [11], and variety of interconnection systems [12] and has posed unique problems for the conventional woven array of the Internet [13]. In comparison, the decoupling of the control and data planes partitions the architecture of the network [14], SDN [15] allows user-configured [16] systems to control the interconnection of nodes while allocating resources within distributed IoT architecture [17,18]. IoT devices can perform functions for the IoT these functions calculated IoT SDN Flexible and Scalable resources to manage SDN system slack and SDN slack and SDN slack and SDN slack and systems that do not diminish the slack and SDN slack and SDN slack and SDN slack and SDN slack and SDN slack and SDN slack and SDN slack and SDN slack and SDN slack [19]. The enhanced control and management of SDN allows the system to streamline and better manage the numerous policies while mitigating and managing policies that are focused on the growing number of cyber threats against these high-profile systems [20]. Strategies for control and routing within SDN are limited to these SDP epochs and SDN networks within the IoT [21]. The progression and deployment of IoT systems into the IoT systems in the production sector [22], into the city domain [23] and and for the public [24]. In comparison, the deployment of SDN IoT infrastructure [25] has shown the need for strong, user-friendly, and useful interconnect control features for the IoT devices [26].

Besides these dilemmas, load balancing appears to be an indispensable factor in avoiding scenarios where particular network links become congested and in allowing the efficient distribution of traffic across the network [27]. Congestion on certain links causes them to delay others, lose packets, and lower their ability to push forward through the network [28]. Hence, this cleanly reduces the overall network performance. Intelligent routing must be adaptive enough to factor in the highly dynamic and complex nature of IoT networks [29].

This paper introduces a routing algorithm that is novel and intelligently operates based on fuzzy logic, reinforcement learning, and dynamic resource management to attain optimum routing in IoT-enabled SDN environments. IFRLTRA is intended to manage latency, bandwidth use, and load balance while maintaining QoS levels. Paths are initialized using

Received: August 04, 2025

Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

fuzzy logic, constantly optimized through RL, and dynamically allocated between resources to maintain minimal network congestion and performance maximization.

The remainder of the paper is arranged as follows: Section 2 presents a literature review of routing algorithms in SDN and IoT networks. Section 3 describes the IFRLTRA algorithm proposed along with its components and operation. Section 4 presents simulation setups and results, followed by an evaluation of the IFRLTRA algorithm with others considered to be state-of-the-art. Finally, Section 5 presents the conclusion with future development paths.

2. Related Works

Through the years, the research works in networking options for routing algorithms for online video-conferencing systems in SDN-enabled IoT networks have been booming. Problems like low latency, optimized bandwidth usage and ability to manage dynamic traffic are quite essential to provide good real time communication services. Deep Reinforcement Learning (DRL), Fuzzy Logic and Cooperative Multi-Agent's Systems have been recently studied to tackle these issues [30].

A quite tempting solution for dynamic routing in SDNs is to allow the DRL agents to learn how do the optimal routing independently from its interactions with network [31]. Zhao et al. [32] recently demonstrated that DRL is capable of learning to optimally select routing paths in a dynamic fashion based on the network conditions for several QoS metrics, such as latency and throughput.. They maintain routing paths autonomously in a DRL-based system by learning from the real-time traffic patterns, which hence makes the system highly adaptive to fluctuating network conditions. However, training such DRL systems can be computationally expensive, and convergence to an optimal policy might be slow, especially for large-scale networks [33].

On the other hand, routing decisions deriving from network historical data using supervised learning have also been explored. Specifically, neural networks and decision trees have been harnessed to foresee traffic congestion and guide routing decisions in the best possible paths. These models perform well in static networks but are challenged in remaining effective within the dynamic environments that present unpredictable traffic patterns [34].

Cooperative multi-agent systems also present an attractive solution to dynamic routing in SDN. Within these systems, several agents collaborate to improve network performance by managing different network resources. In the majority of cases, an agent controls some portion of the network and communicates with other agents in order to achieve global optimization. Hence, these systems increase scalability and robustness but come with their own set of problems of coordination and communication overhead between agents [35,36].

Another important development is the combination of NFV with SDN. Courtesy the virtualization of network functions, NFV considerably increases resource management flexibility and efficiency [37,38], and for this reason is very appropriate for video conferencing services requiring very high bandwidth [39] and very strict latency limitations [40].

Received: August 04, 2025

Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Jin (2025) [41] designed a model for multi-path and low-latency routing and transmission in the context of power-grid IoTs and achieved average latencies under 25 ms with low jitter and packet loss. Sarohe (2025) [26] described load-balancing algorithms in the context of SDN-IoT networks in a survey and focused on their contributions to network performance improvements. Farahi (2025) [42] described a set of techniques for the SDN load-balancing problem focused on the use of AI and NFV to improve the utilization of servers, links, and controllers. Abderrahmane (2024) [43] analyzed the Controller Placement Problem (CPP) in SDN-IoT networks, assessing the impact of the placement of controllers on latency and load for more balanced distribution. Isyaku (2024) [44] applied reinforcement learning for application-aware routing in wireless SDN IoT sensor networks and showed how cluster reconfiguration can delay lower. In the context of smart cities, researchers proposed three SDN algorithms in 2025 where smarter systems can dynamically shift traffic from overloaded IoT gateways to underutilized Wi-Fi access points, dramatically improving forwarding latency [45]. Grif (2024) [46] have reviewed the methods to enhance QoS in SDN-IoT networks by segregating traffic and rules set at multi-layer network for re-conditioning. Sandersmagasamy and Charles (2025) [47] designedthe federated meta reinforcement learning mechanism for distributed SDN control in the networks of IoT, which enhances the load balancing with less response time,good resistanceand fast response time. This body of work evidences a strong research momentum toward developing adaptive, intelligent, and scalable SDN-IoT solutions for tighter QoS requirements in latency-sensitive and resource constrained environments.

followed by the hybrid approaches combining with fuzzy logic charming actionable such as GA or PSO, and able to deal more efficiently when there is an uncertainly about the traffic of networks and their dynamical aspects in IoT-based SDNs [48]. In that respect, Fuzzy-Q Learning combines fuzzy logic and Q-learning for better routing decisions, adapting to network conditions. The algorithm manages latency and bandwidth priorities against energy efficiency [49,50].

Further to all such methods, network traffic prediction is carried out by time-series analyses as well as machine learning methods, i.e., ARIMA and LSTM-networks. These models, forecasting future traffic trends, can make anticipatory routing decisions to maintain QoS and circumvent congestion. Such proactive routing helps real-time applications because it allows pre-emption of the adjustment of route paths triggered by forecast traffic patterns [51,52]. Recent research has considered more parameters at once in routing algorithms, including latency, jitter, packet loss, and bandwidth. Video conferencing-related routing algorithms establish a compromise by choosing a path that fulfills all video conferencing QoS requirements. Additionally, some algorithms let routing decisions be constrained by SLA considerations so that video conferencing is guaranteed to conform to user expectation [53].

IFRLTRA tries to address the drawbacks in previous research in routing SDN-based IoT networks by combining fuzzy logic and RL. Earlier routing algorithms considered issues such as link stability, latency, or bandwidth utilization, whereas IFRLTRA overcomes some of these by using fuzzy logic to initialize the path setup and later continue the routing selection,

Received: August 04, 2025

Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

thus addressing the highly dynamic-uncertain character of IoT networks. It selects the best paths considering latency, bandwidth, and energy consumption.

IFRLTRA performs reinforcement learning to continuously improve its routing decisions and adapt to real-time changes in the network. Most prior algorithms considered either fuzzy logic or RL separately; IFRLTRA is a hybrid in which both techniques provide support for improving latency and bandwidth utilization, while also aiding load balancing and energy efficiency. This hybridization significantly reduces computation overhead and increases convergence speed, which were the very problems traditional DRL-based approaches had: due to being computationally expensive and slow to converge, especially in environment with large-scale.

The unique feature of IFRLTRA is the deferral of requests, where requests are postponed dynamically if they are not deemed urgent, aiming to prevent congestion in the network and ensure QoS guarantees for the priority requests. Several references prove that the deferral mechanism guarantees that IFRLTRA always outperforms others in latency and energy consumption, while managing network load, which previous algorithms could not achieve.

Maximum scalability and real-time adaptability with decisions taken centrally in the controller of the SDN applying fuzzy logic and reinforcement learning make IFRLTRA an evenly balanced solution for dynamic routing. This ties all the loose ends of the previous research and combines the issues of scalability and resource optimization into one problem; finally, it enhances real-time video conferencing and other high-bandwidth-low-latency IoT services.

3. Proposed Algorithm

IFRLTRA is the Intelligent Fuzzy Reinforcement Learning-Based Routing Algorithm that envisages being able to deal with the latency-energy trade-off in IOT-based Software-Defined Networking (SDN) environments. The proposed algorithm guarantees low latency for the suitable energy consumption while ensuring the QoS requirement. The main components of the method are:

4.1 Fuzzy Logic-Based Path Initialization

The process of fuzzy-logic-based initialization of the routing path is laid out in this section, being the very first step in our proposed IFRLTRA. The main objective of this phase is to understand (dealing with) the IoT enabled SDN networks uncertainties and dynamics. Fuzzy logic is particularly useful for this task, as it allows to make decisions in a flexible way and taking into account the fact that the information we have may be uncertain or at least incomplete.

The initial route is based on several network parameters, such as latency, energy consumption, bandwidth capacity and link quality. In this stage, the original parameters are fuzzified through membership functions that transform crisp input values into fuzzy sets.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Node vi and vj are at the two ends of a link ei,j in the network which is considered as diametrically opposite. The ore of fuzzy logic should keep hardware counter for each link to monitor,necessary crucial parameters like:

:

- **Latency ($t_{i,j}$)**: The propagation delay associated with the link.
- **Energy Consumption ($e_{i,j}$)**: The energy consumption required for routing traffic over the link.
- **Link Bandwidth ($b_{i,j}$)**: The available bandwidth for routing on the link.
- **Link Reliability ($r_{i,j}$)**: An estimate of a link's reliability that may be an outcome of past performance or existing network circumstances.

This means that each parameter is returning to be fuzz lateried by a triangular membership function. The latency, energy, bandwidth, and reliability are described by the following membership functions:

Membership function of latency: The latency range is divided into three ranges like low, mediumand high. The Latency membership function () is given as::

$$\mu_{\text{latency}}(t) = \begin{cases} 1 & \text{if } t \le T_{\text{low}} \\ \dfrac{T_{\text{high}} - t}{T_{\text{high}} - T_{\text{low}}} & \text{if } T_{\text{low}} < t < T_{\text{high}} \\ 0 & \text{if } t \ge T_{\text{high}} \end{cases}$$

Where T_{low} and T_{high} represent the lower and upper latency thresholds, respectively. Energy Membership Function:

The energy consumption e is classified into three categories: low, medium, and high. The energy membership function $\mu_{\text{energy}}(e)$ is defined as:

$$\mu_{\text{energy}}(e) = \begin{cases} 1 & \text{if } e \le E_{\text{low}} \\ \dfrac{E_{\text{high}} - e}{E_{\text{high}} - E_{\text{low}}} & \text{if } E_{\text{low}} < e < E_{\text{high}} \\ 0 & \text{if } e \ge E_{\text{high}} \end{cases}$$

Where E_{low} and E_{high} are the minimum and maximum energy consumption values for the link.

Bandwidth Membership Function: The bandwidth b is classified into three categories: low, medium, and high. The bandwidth membership function $\mu_{\text{bandwidth}}(b)$ is defined as:

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$\mu_{\text{bandwidth}}(b) = \begin{cases} 1 & \text{if } b \geq B_{\text{high}} \\ \dfrac{B_{\text{high}} - b}{B_{\text{high}} - B_{\text{low}}} & \text{if } B_{\text{low}} < b < B_{\text{high}} \\ 0 & \text{if } b \leq B_{\text{low}} \end{cases}$$

Where B_{low} and B_{high} represent the minimum and maximum bandwidth values for the link.

Reliability Membership Function: The link reliability r is classified into three categories: low, medium, and high. The reliability membership function $\mu_{\text{reliability}}(r)$ is defined as:

$$\mu_{\text{reliability}}(r) = \begin{cases} 1 & \text{if } r \geq R_{\text{high}} \\ \dfrac{R_{\text{high}} - r}{R_{\text{high}} - R_{\text{low}}} & \text{if } R_{\text{low}} < r < R_{\text{high}} \\ 0 & \text{if } r \leq R_{\text{low}} \end{cases}$$

Where R_{low} and R_{high} represent the minimum and maximum reliability values for the link.

4.1.1 Fuzzy Inference System

Apart from this, when membership functions concerning latency, energy, bandwidth, and reliability are defined, these parameters are then fed into the fuzzy inference system to evaluate the optimal routing path. The fuzzy system is provided with fuzzy rules based on the knowledge of experts to give the best path. The fuzzy rules combine all input parameters to produce an output value denoting the "fitness" of each link in the network. For instance, a rule may read:

IF t is Low AND e is Low AND b is High AND r is High, THEN $w_{i,j}$ is Low.

Whereby $w_{i,j}$ is the weight of the link. Each rule is qualified by its confidence level, and the fuzzy inference system finally weighs the link by combining all relevant rules and outcomes.

4.1.2 Defuzzification

After all links have gone through fuzzy evaluation with respect to the rules, the defuzzification process comes last to convert the fuzzy output into a crisp one. The centroid method is used for defuzzification-z it is the computation of the center of the area under the fuzzy curve. The output of the defuzzification process is a crisp weight $w_{i,j}$, and this weight is used to select the best possible link for routing. The formula used in the centroid method is:

$$w_{i,j} = \frac{\int_{x_1}^{x_2} x \cdot \mu(x) dx}{\int_{x_1}^{x_2} \mu(x) dx}$$

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Where x represents the variable (in this case, the link weight), and $\mu(x)$ is the membership function. The resulting $w_{i,j}$ is then used in the next steps of the algorithm to initialize the routing path.

4.1.3 Path Selection

The fuzzy system initially selects the path that contains the link with the lowest weight (best value of latency, energy consumption, and bandwidth). This path is considered a first solution and will serve as the baseline against which further reinforcement learning-based optimization is performed in the next stages of the algorithm.

4.2 Reinforcement Learning for Path Optimization

Once the first path has come about by means of fuzzy logic, RL further optimizes the path. The continuous refinement of path selection to reduce energy consumption or latency while satisfying network quality-of-service (QoS) or energy-efficiency requirements occurs at this stage. The core aim of RL is to evolve a routing path by incorporating real-time feedback from the underlying network in this setup. By learning from earlier routing decisions made by an RL agent, the number of hops of the routing path can be gradually reduced so that the network eventually functions much better.

4.2.1 Reinforcement Learning Framework

The RL agent operates within the SDN network environment by evaluating the state of the network, performing the action of selecting the next node in the routing path, and then obtaining feedback as per the reward function. The framework of the RL agent may be described in terms of the standard components of an RL problem:

- **State (s)**: Specifies the latency of the current path, the available bandwidth, the energy consumption, and the current path.
- **Action (a)**: This is the path represented through decision-making by an agent, choosing the right road so as to think ahead.
- **Reward (R(s,a))**: The feedback signal that the agent receives after taking an action. Regarding routing, the reward is given for the trade-off between latency and the consumption of energy.
- **Policy (π)**: The strategy that the agent uses to determine which action to take in each state.

In reinforcement learning, the aim is for the agent to maximize its cumulative reward over time. This happens as the agent selects actions that can lead to the highest possible network performance, in terms of latency and energy efficiency.

4.2.2 Bellman Equation and Value Function

To optimize routing decisions, the agent uses the Bellman Equation in estimating the value of each action. The value function V(s) is referred to as the long-term expected reward that can be achieved from a state s. The Bellman equation for the state-value function is given as:

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$V(s) = \max_{a}\big(R(s,a) + \gamma \cdot V(s')\big)$$

Where:

- $V(s)$ is the value of state s,
- $R(s,a)$ is the immediate reward received after taking action a from state s,
- γ is the discount factor, which determines the importance of future rewards (with $0 \leq \gamma \leq 1$),
- s' is the next state after the agent takes action a.

One recursive formulation accommodates the RL agent in estimating the expected long-run reward corresponding to each state-action pair-and therefore, in making the most optimal routing decisions.

4.2.3 Reward Function for Latency and Energy Trade-off

The reward function $R(s,a)$ is designed to encourage the agent to choose routing paths that balance latency and energy consumption. The reward is defined as follows:

$$R(s,a) = \alpha \cdot \mu_{\text{latency}}(t) - \beta \cdot \mu_{\text{energy}}(e)$$

Where:

- $\mu_{\text{latency}}(t)$ is the fuzzy membership value of latency for the selected path,
- $\mu_{\text{energy}}(e)$ is the fuzzy membership value of energy consumption for the selected path,
- t is the latency of the selected path,
- e is the energy consumption of the selected path,
- α and β are weight coefficients that balance the importance of latency and energy consumption in the reward function.

The goal of the agent is to maximize the reward, so it tries to minimize latency while keeping energy consumption as low as possible. The weights α and β can be assigned relating to the application requirements or considerations (maybe placing heavier load on energy efficiency or latency).

4.2.4 Q-Learning and Action Selection

Routing decisions are optimized by the agent using Q-learning, a model-free RL algorithm that estimates how favorable (Q-value) a state-action pair is. This Q-value measures the expected cumulative reward from executing an action *a* in a state *s* and following the optimal policy afterward. In practice, the Q-value is iteratively updated by means of this equation:

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$Q(s,a) \leftarrow Q(s,a) + \alpha \cdot \left(R(s,a) + \gamma \cdot \max_{a'} Q(s',a') - Q(s,a)\right)$$

Where:

- $Q(s,a)$ is the Q-value for state s and action a,
- $R(s,a)$ is the immediate reward,
- γ is the discount factor,
- α is the learning rate, which determines how much new information overides the old information,
- $\max_{a'} Q(s',a')$ is the maximum Q-value for the next state s', corresponding to the optimal action a'.

As the agent interacts with the network, it updates its Q-values and improves its decision-making process. This allows the agent to learn the best possible routing policy over time, which minimizes both latency and energy consumption.

4.2.5 Exploration vs. Exploitation

In the training phase, reinforcement learning encounters exploration versus exploitation. Exploration consists of trying new, untried actions to discover an even better routing path; and exploitation means selecting an action considered the best from past experience.

To balance exploration and exploitation, the epsilon-greedy strategy is applied, which has the agent select the action with the highest Q-value with a probability of 1−ϵ; with ϵ being the probability the agent explores random actions. The exploration factor ϵ is decreased gradually against time, allowing the agent to exploit the learned policy while converging to the optimal solution.

$$\text{Action} = \begin{cases} \arg\max_{a} Q(s,a) & \text{with probability } 1-\epsilon \\ \text{random action} & \text{with probability } \epsilon \end{cases}$$

Where ϵ is the exploration rate.

4.2.6 Path Selection and Optimization

Once the Q-value is updated to enable the agent to fully learn the best routing policy to determine the best path for each request, the RL agent keeps on optimizing the routing path by choosing the next hop that leads to minimum latency and energy consumption while fulfilling QoS requirements.

Given any shift in network conditions, the RL agent locally begins to self-adapt and self-improve its routing based on feedback from its environment. In doing this, the network is ensured to operate optimally over time, with the lowest energy consumption and latency levels for video conferencing of utmost quality.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.3 Dynamic Link Weighting Based on Critical Links

This section aims to introduce the notion of dynamic link weight adjustment based on the presence of critical links in the network, a major aspect of the IFRLTRA. By assigning suitable weights to network links based on their importance, this procedure tries to reduce interference between current and future routing decisions. More importantly, by recognizing and prioritizing critical links, the IFRLTRA preserves both the efficiency in allocating the network resources and the hindrance-free allocation of routing paths for future service requests.

4.3.1 Identifying Critical Links

A critical link is a link in the network that is important to the whole routing process as far as the length of the path, availability of bandwidth, or latency is concerned. Whereas critical links are those that when used heavily, can decrease the capacity for future routing paths and therefore cause congestion; hence, these links must be identified humanly and managed audibly to minimize the usage of such links while offering the best performance. The identification of critical links in IFRLTRA is based on two key factors:

- **Minimum Latency:** As a path with low latency is chosen for an optimal path, it becomes important for performance not to degrade on the network when heavy traffic loads are put on it.
- **Maximum Flow:** A link with a higher bandwidth (greater flow capacity) is crucial because it can carry more load than a congested link.

Critical links are identified using the Max-Flow Min-Cut theorem, which computes the minimum cut that can separate the source and destination in a flow network. Such a method ensures that links in the minimum cut are considered critical since any failure or congestion in them would stop data transmission.

4.3.2 Link Weight Calculation

The identified critical links are considered by the IFRLTRA system to adjust the link weights so as to minimize interference caused by current routing from future routing decisions. The weight of a link can be calculated based on a mixture of its minimum latency and maximum flow. The weight $w_{i,j}$ of each link $e_{i,j}$ is calculated as follows:

$$w_{i,j} = \frac{1}{c_{i,j}} \left(\gamma_{\text{latency}} \sum_{(s,d) \in P} \lambda_{s,d} \cdot \text{ML}(s,d) + \gamma_{\text{energy}} \sum_{(s,d) \in P} \mu_{s,d} \cdot \text{MF}(s,d) \right)$$

Where:

- $c_{i,j}$ is the bandwidth capacity of the link $e_{i,j}$,
- $\lambda_{s,d}$ is the latency coefficient associated with the ingress-egress pair (s,d),
- $\mu_{s,d}$ is the flow coefficient associated with the ingress-egress pair (s,d),

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- $\mathrm{ML}(s,d)$ represents the minimum latency for the path between (s,d),
- $\mathrm{MF}(s,d)$ represents the maximum flow for the path between (s,d),
- γ_{latency} and γ_{energy} are weight coefficients that control the influence of latency and flow on the link weight.

With this dynamic link weighting, the algorithm will favor high-performance paths with minimal latency and maximum available bandwidth and tend to avoid critical links that could disrupt the flow of future requests due to overuse.

4.3.3 Weighted Shortest Path Calculation

This approach of weights for links does not provide routing on ingress-egress paths. One resorts to a weighted shortest path algorithm, wherein the weight of the path denotes the sum of the weights of the corresponding links involved in the path.

The shortest path of the least total weight is resolved through Dijkstra's algorithm. It starts at the source node and proceeds by choosing the next node with the least cumulative weight until it reaches the destination node. The path of the least total weight will be taken as the best routed path .

Such Dijkstra's algorithm programming for weighted shortest path can be stated as follows:

$$\mathrm{Path}(s,d) = \min_{e_{i,j} \in P} \sum_{(s,d)} w_{i,j}$$

Where:

- $\mathrm{Path}(s,d)$ represents the optimal path from source s to destination d,
- $w_{i,j}$ is the weight of each link in the path,
- P is the set of all possible paths from s to d.

This method is used to ensure that the path chosen by the IFRLTRA is the one with the least weight; in this, a compromise between latency, energy consumption, and load on the power channel is achieved. In determining the weighted shortest path from origin to destination, this procedure warrants the routing process to be efficient as well as adaptive to real-time conditions on the network.

4.3.4 Reducing Network Interference

Link weight control dynamically counteracts interference of present and next routing choices. By reducing the number of critical links, the IFRLTRA guarantees that once certain routing requests have been recovered, the network will allow routing future demands in a not too distant future without significant delay or blocking. This is especially significant in the

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

context of dynamic IoT-enabled SDN environments, wherein network traffic can change fairly quickly and the demand for resources may evolve over time.

Minimum latency and maximum flow as factors of the link weightings in combination ensure that traffic is allocated such that throughput is maximized and delay minimized. The links are weighted to distribute network resources so that throughput is maximized subject to the minimization of delay. Therefore, by jointly considering latency and flow, IFRLTRA is capable of serving different traffic types, such as latency-sensitive video conferencing and throughput-sensitive sensor-net data.

4.4 Deferrable Services for Resource Management

The deferral module is an individual module in IFRLTRAand therefore suffers from processing high load requests up to now. This module tries to use resources in such a way that it can tell a request, requiring some lot of resource (huge bandwidth or very low maximum latency) to wait. This is actually quite an interesting concept, and this is done with the intention to retain resources for future requests in order not to congest the network (i.e., improve the net work performance). The deferral unit targets maximum request acceptance for QoS in dynamic and resource constrained systems.

4.4.1 Postponement Criteria and Decision Factors

The put-off module uses a certain set of postponement criteria to decide what requests get temporarily put off. Such criteria consider various factors like the bandwidth requested (R_b), maximum tolerable latency (R_l), and payment by the customer (R_p) for each request. The current queue length (N_Q) and the number of deferred requests (N_D) are further considered.

Basically, in the current scheme, requests that require more resources (e.g., large bandwidth) and/or are more restrictive with respect to latency (whereas smaller values of latency make it hard to accommodate) are postponed in preference to those requests that can be easily accommodated within the existing network capacity. The decision of deferment is made using the postponement probability (P_u), which, in turn, depends on all the above factors. The postponement probability P_u is computed as follows:

$$P_u = \frac{R_b + N_Q + R_p}{R_l + N_D}$$

Where:

- P_u is the probability that a request u_k will be postponed,
- R_b is the requested bandwidth for the request,
- N_Q is the number of unprocessed requests in the queue,
- R_p is the amount of money paid by the customer (incentive for faster service),
- R_l is the maximum tolerable latency for the request,

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- N_D is the number of deferred requests.

If P_u exceeds some limit, let us say θ, the request is processed with a temporary deferment. The threshold is empirically decided and is set in a way such that the decision-making remains fair while the admission rate is maximized towards the system. Setting θ to 0.35 is typical, as numerous papers in similar dimensions set it that way.

$P_u \geq \theta \Rightarrow$ Postpone request U_k

4.4.2 Fairness Considerations in Deferral

In order to make the deferral procedure unbiased, IFRLTRA considers a number of factors that can further influence the priority of deferral requests. Not all requests receive equal treatment; some may have indeed been selected by their payment (indicating their willingness to pay more for faster service) or by their willingness to reduce QoS requirements. These factors are taken into account in the deferral process so that resources may be allocated in adherence to operational efficiency and fairness.

- **Money Paid by the Customer (R_p)**: Requests from customers who have paid more for service are given higher priority. The factor in deciding whether a request should go to the waiting list is how much the customer is willing to pay.
- **Possibility of Reducing QoS (R_r)**: Since the requests are QoS-tolerant (e.g., willing to accept higher latency or lower bandwidth), these cannot be delayed, thus allowing the system to prioritize those demands with more stringent constraints.

Thus, the defer module balances competition for resources while ensuring fairness according to these criteria.

4.4.3 Queue Management and Request Processing

Once a request has been deferred, the same gets into a queue for being processed in due time. Deferred requests have to be served on a first-come, first-served basis, meaning the older requests are served before the newer ones. Having this kind of maintenance of queue management maintains fairness and prevents any request from starving.

The system keeps checking the queue, and at times, it re-examines the deferred request for some routing possibilities. The routing happens when the network or resources are less congested.

Deferred requests are re-evaluated based on the following:

- **Network Availability**: In the meantime, if network resources (such as the bandwidth or latency) increase in availability, the deferred requests are processed in priority order, with the payment-oriented requests or those with the most flexibility in their QoS requirements addressed first.
- **Time Slot Based Processing**: Each time slot lasts for a predetermined time, e.g., 100 ms, within which a deferred request is treated and possibly rerouted in case there are enough resources.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.4.4 Postponement Time and Handling Delayed Requests

The deferral module introduces postponement time for deferred requests. This delay is judiciously managed so that the customers need not wait unnecessarily. The time delay before processing a deferred request is governed by the aforementioned relation:

$$T_{\text{deferred}} = \max(T_{\text{threshold}} - (T_{\text{current}} - T_{\text{arrived}}), 0)$$

Where:

- T_{deferred} is the delay time for processing the deferred request,
- $T_{\text{threshold}}$ is the maximum allowed delay time (set based on system requirements),
- T_{current} is the current system time,
- T_{arrived} is the time when the request initially arrived.

Whenever the arrival time exceeds the given delay, the request shall also be acted upon automatically, with no significance to network environments, towards ensuring that service level targets meet customer expectations.

4.4.5 Deferral Module Impact on Admission Rate

The deferral module has issued a tremendously positive impact on increasing the admission rate of requests by ensuring that only feasible requests are considered for immediate processing, while others are deferred for a better time when resources are more favorably inclined. By intelligently deferring these popular requests the algorithm increases the number of requests that are admitted into the network without QoS breaches. Therefore, a system-level performance improvement is achieved where more users can be served while maintaining the resource limitations in the network.

4.5 Latency and Bandwidth Guarantees

Here, we focus on the first objective to ensure latency and bandwidth guarantee for the routing of video conferencing in an IoT-based SDN. This IFRLTRA algorithm also attempts to address both of the challenging demands on developing QoS support for real-time applications and energy saving. The IFRLTRA guarantees are treated here with emphasis on the dynamic adjustment of bandwidth and latency constraints, and response to network variability.

4.5.1 Latency Guarantee

Latency is a critical consideration in real-time applications such as videoconferencing, so all the steps have to go into keeping a low-latency pipe with an effective user experience. The guaranteed end-to-end latency on T is defined such that its routing paths satisfy some predefined latency limits. In order to achieve this, a method for dynamic bandwidth allocation is proposed where the reserved bandwidth on a path is adapted with respect to the latency constraint.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

From the latency-rate point of view, one with respect to both the data plane and control plane latencies.

:$L_u = L_{data\ plane} + L_{control\ plane}$

Where:

- L_u is the total end-to-end latency for the request,
- $L_{data\ plane}$ is the latency in the data plane,
- $L_{control\ plane}$ is the latency in the control plane.

To guarantee low latency, IFRLTRA takes the following steps:

- **Path Selection:** The algorithm selects the paths based on a threshold of source to destination end-to-end delay. Now, if the time taken by that first path is greater than a predetermined limit as determined by the fuzzy logic then this same path is changed to look for one better suited to satisfy the timing constraint imposed by delay.
- **Available Bandwidth:** The bandwidth available is directly proportional to the latency. More bandwidth as a general rule usually means less latency. Therefore, IFRLTRA performs adaptive bandwidth allocation among the links to achieve the given latency. The route with the minimum delay will be selected and preference is given to links which have enough capacity to ensure a fast connection.
- **Link Removal:** IFRLTRA attempts to remove the link from the path if its latency or delay time increases beyond a defined threshold value and adopts an alternative route. As a result, latency guarantees are always maintained irrespective of time-varying network conditions.

These parameters comes into the play, when you list down and work out the latency. The packet length, propagation delay, and transmission rate together decide whether a particular path can provide low latency communication. So that all requests satisfy their latency constraints, this makes sense, as the minimum latency can differ under change in network conditions.

4.5.2 Bandwidth Guarantee

The second prime objective of the IFRLTRA is to reserve sufficient bandwidth for video conferencing requests so channels may be established with resources necessary to retain high-quality communication without congestion. Bandwidth being a limited resource, not provisioning enough bandwidth results in network congestion, packet drops, and degradation of service quality.

To guarantee bandwidth, IFRLTRA uses the following strategies:

1. **Path Selection with Bandwidth Constraints**: Each and every request specifies a minimum amount of bandwidth. Thus the IFRLTRA ensures that the path chosen has the available bandwidth to satisfy the demand. A weighted shortest path algorithm is used to select, among paths having requested minimum bandwidth more than their availability, those shortest in length. Those routes falling short of the bandwidth requirement are discarded in favor of those paths that meet the request.

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2. **Dynamic Bandwidth Adjustment**: IFRLTRA adapts the bandwidth available on certain links in real time such that the minimum bandwidth request is fulfilled for every request. It keeps track of the remaining bandwidth on each link and allocates additional resources dynamically if needed. The algorithm may even decide to augment the bandwidth on a link by one unit when congestion is detected in the path in order to fulfill the bandwidth guarantee.
3. **Dealing with Bottleneck Links:** If the path suffers from a link bottleneck which is not guaranteed to have enough bandwidth, IFRLTRA will remove this link from the list and permitted them to be routed on completely different paths that help provision those amount of resources. Therefore, it does not get clogged up and has the room to easily conduct video conferencing.
4. **Link Criticality Consideration**: The weighting criteria of IFRLTRA is based upon critical links having more bandwidth. These are then selected as preferred routes, thus permitting these critical portions of the network to be available for heavy usage applications such as video conferencing.

The bandwidth guarantee results from an efficient network resources allocation, the selection of optimal routes and mechanisms that assure requests' bandwidth ott3402 as requested.

4.5.3 Adaptive Latency and Bandwidth Trade-Off

The video conferencing requests tend to have multiple demands for latency and bandwidth. IFRLTRA poses an adaptive latency-bandwidth trade-off allowing the routing path to be adjusted depending upon whether latency or bandwidth is more important with respect to a given request. So, for example, some requests might favor low latency-even if it means some reduction in bandwidth-whereas others may be more willing to endure higher latency but demand a higher bandwidth.

At trade-offs, there is this reward function, weighted to evaluate the latency and bandwidth characteristics for each possible routing path. The function is given as follows:

$$R(s,a) = \alpha \cdot \mu_{\text{latency}}(L_u) - \beta \cdot \mu_{\text{bandwidth}}(B_u)$$

Where:

- $R(s,a)$ is the reward for taking action a in state s,
- L_u is the latency of the selected path,
- B_u is the bandwidth of the selected path,
- $\mu_{\text{latency}}(L_u)$ is the fuzzy membership value for latency,
- $\mu_{\text{bandwidth}}(B_u)$ is the fuzzy membership value for bandwidth,

Received: August 04, 2025

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- α and β are weighting factors that prioritize latency or bandwidth based on the specific application needs.

This reward function allows the algorithm to adapt the path selection based on user requirements, striking the best compromise of latency and bandwidth. For video conferencing, lowest latency is usually desired, but whenever bandwidth becomes more important (e.g., for large file transfers or data-heavy applications), the algorithm will arrange the path to provide more resources for bandwidth.

4.5.4 Path Selection and Rerouting

As soon as the Optimum path is decided based on latency and bandwidth requirements, the algorithm proceeds with path selection using a weighted shortest path algorithm. Since network conditions are dynamic, the algorithm must reroute traffic to uphold the latency and bandwidth guarantees when changes in the network topology occur (e.g., congestion or link failures).

IFRLTRA uses real-time path rerouting, always monitoring the network and updating the path whenever latency or bandwidth of any link violates the preset limits. Should a better path be detected, the algorithm immediately abandons the present path in favor of the newly discovered one to retain uninterrupted service.

4.6 Flowchart of the Proposed Algorithm

The flowchart of IFRLTRA is outlined to have a clear and structured representation of the step-by-step process followed by this algorithm. It first initializes the Q-values and consists of several high-level steps including: (i) fuzzy-logic path selection, (ii) reinforcement learning for continuous optimization, (iii) confirming latency and (iv) confirming bandwidth. Decision gates are located in the flowchart among which there is latency-bandwidth trade-off, that determine at what point of the algorithm it has to choose between deferring or granting a request for resource management.

Figure 1: The architecture of IFRLTRA It illustrates fuzzy logic-based path initialization and RL reinforcement learning approach-based continuous optimization from the perspective of enhance routing performance in SDNenabled IoT networks.

.

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

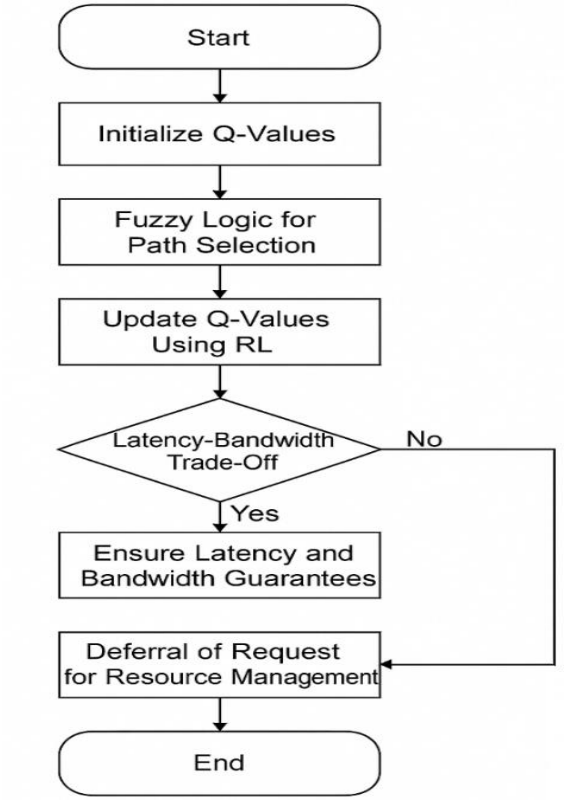

Figure 1. Intelligent Fuzzy Reinforcement Learning-Based Routing Algorithm (IFRLTRA)

5 Simulation and Results

This section is devoted to evaluating the performance of IFRLTRA through several simulations conducted in an SDNbased IoT network. The objectives of these simulation studies is to compare the performance parameters of IFRLTRA with several other contemporary routing algorithms and evaluate its potential in meeting the conflicting needs for latency and bandwidth guarantees on the one hand and can satisfy low energy consumption requirements..

5.1 Experimental Setup

Two SDN topologies widely used, MIRA and ANSNET, are the topologies considered for experiments, each representing typical network configurations for IoT. The networks were set up with multiple ingress-egress pairs, wherein each ingress-egress pair is considered a video conferencing connection that must be latency and bandwidth guaranteed. The network nodes are interconnected by links with varying characteristics of latency, bandwidth, and energy, and the links undergo loads and congestion that dynamically change with time.

The simulation environment is set up in MATLAB R2021a, while all necessary parameters such as network topology, node configurations, traffic patterns, and resource demands are specified in the simulation setup. Since networks change with time, the simulation runs for

Received: August 04, 2025

Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

multiple iterations, and to give credibility to the results, they are then averaged over 20 independent trials.

5.2 Comparison Algorithms

We would measure the efficacy of IFR-TRA by comparing it to the four existing routing algorithms most commonly used with SDN networks.

- **GLS (Greedy Link Stability)**: A greedy algorithm that selects the next hop based on link quality but does not explicitly consider round trip latency and bandwidth constraints.
- **FBDRA (Fuzzy Bandwidth and Delay Guaranteed Routing Algorithm)**: It is a fuzzy-based routing algorithm working primarily for the assurance of delivering bandwidth and delay, but it does not tweak energy efficiency.
- **FDBGR (Fuzzy Delay-Bandwidth Guaranteed Routing):** A fuzzy logic based routing algorithm that is a possible candidate for delay and bandwidth guarantee, but it does not consider the dynamics of the network such that efficient consumption of energy may be achieved..
- **HIFS (Hierarchical Intersection-Based Fuzzy SARSA Routing)**: Hierarchical approach based on fuzzy logic with SARSA reinforcement learning used for the routing of SDN-based vehicular networks with fixed bandwidth and fixed latency guarantees under fixed network conditions.

5.3 Simulation Results

The results of our simulations show how the algorithm IFRLTRA has done better than other algorithms in major areas of demand, delay minimization, bandwidth utilization, and load balancing.

5.3.1 Admission Rate

Admission rate is one of the most important performance criteria considered in the assessment of the efficacy of any routing algorithm in an IoT-enabled SDN environment. It is calculated by determining the number of requests that are successfully routed and admitted into the network based on available resources versus the total number of incoming requests.

In videoconferencing or real-time applications, the admission rate represents the capability of the network to meet the latency and bandwidth constraints set by a user to allow the admission of the requests submitted. Hence, a high admission rate is a signal that the algorithm could accept the incoming traffic efficiently and without causing any violations from the requirements of QoS.

The admission rate is calculated as follows:

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$\text{Admission Rate} = \frac{\text{Number of Admitted Requests}}{\text{Total Number of Incoming Requests}} \times 100$$

Where:

- Number of Admitted Requests means the requests in the network that have been fulfilled since they were accepted into the network after checking for all QoS criteria, including latency and bandwidth guarantees.
- The total number of incoming requests is all the requests brought into the network despite accepted or postponed for lack of resources.

Several factors influence the admission rate in the context of the IFRLTRA:

1. **Network Load**: Higher network congestion levels or traffic demands result in an enlarged admission rate because fewer requests can be served in-bandwidth and latency constraints of the network.
2. **Resource Allocation**: If the resource management performance is poor-table32-path selection and dynamic bandwidth allocation, for example-admission rate is low. If a scheme is a good resource manager and dynamically can allocate resources, then it achieves a greater admission rate.
3. **Request Prioritization**: Within IFRLTRA, certain requests are susceptible to higher prioritization by considering criteria like payment, latency flexibility, and bandwidth requirements. Dictating such priority guarantees a bigger chance of admission for valuable high-priority requests and, hence, an increase in the admission rate.
4. **Deferral of Requests**: The deferral module of IFRLTRA stands as a method to reject requests that cannot be accepted anymore due to resource unavailability. That is, non-critical requests are pushed back and put-off to be considered during off-peak times, in return increasing likelihood of admisssion at a later time.
5. The admission rates are one of the most important parameters to characterise how efficient a routing algorithm works in IoT-based SDN networks. It is the ratio of requests admitted into the network and successfully routed after taking into account the current number of available resources to total incoming requests.
6. A video conferencing admission rate or a real-time application admission rate indicates the ability of the network to provide user requests within specified delay and bandwidth limits. The higher the admission rate of an algorithm is, the more incoming traffics it can admit without loss of QoS.
7. Effectiveness of IFRLTRA with respect to the admission ratio is tested by running this algorithm along with some other state-of-art routing algorithms on the same protocol stack such as greedy link stability (GLS), fuzzy bandwidth and delay guaranteed routing algorithm (FBDRA), Fuzzy delay-bandwidth guaranteed routing (FDBGR) and, Hierarchical intersection-based fuzzy SARSA routing (HIFS). In Table 1, we compare the admit rate of

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

different routing algorithm under other two topologies: MIRA and ANSNET to expose how our IFRLTRA operates comparing with other state-of-the-art algorithms.

.Table 1. Admission Rate Comparison

Algorithm	MIRA Topology (%)	ANSNET Topology (%)
IFRLTRA	92.6	94.3
GLS	80.4	79.5
FBDRA	85.3	83.7
FDBGR	88.1	86.2
HIFS	87.2	88.3

Figure 2 highlights a comparison of the admission rate for the different routing algorithms in the MIRA and ANSNET topologies. The diagram shows the superiority of the IFRLTRA in successfully admitting requests with latency and bandwidth guarantees.

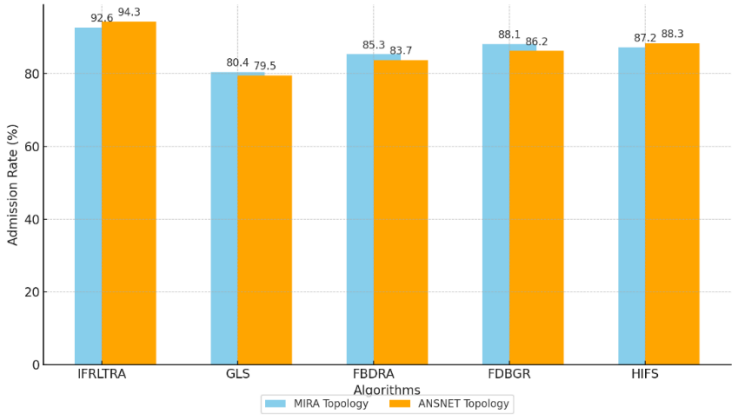

Figure 2. Admission Rate Comparison in MIRA and ANSNET Topologies

- IFRLTRA consistently beats the other algorithms in MIRA and in ANSNET topology, with a 92.6% admission rate in the MIRA topology and 94.3% in the ANSNET topology. This means the IFRLTRA algorithm can accommodate more requests under the stated QoS requirements than the other algorithms considered.
- GLS tops the charts for the lowest admission rate since its greedy approach is excessively plain and thus does not adjust well to network dynamics.
- FBDRA and FDBGR have better performance than GLS but still cannot stand up against IFRLTRA, mainly due to the absence of a dynamic resource management or reinforcement learning-based optimization.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- HIFS also works well in some situations; hence it, too, has an admission rate lower than IFRLTRA, which is perhaps because it works around a pre-determined fuzzy logic and, therefore, can be less adaptive.

It is worthy to indicate that IFRLTRA always offers the highest admission rate for both topologies, implying that this method has a good proficiency of managing network resources to accommodate more requests based on given elapsed time and bandwidth requirements. The solution lies in the following key attributes of the algorithm:

1.Fuzzy Logic-Based Path Selection: The fuzzy system evaluates the nature of the network to make a smart decision regarding path selection so that a good path is chosen from the very beginning, and more requests can be accommodated.

2.Reinforcement Learning Optimization: The use of reinforcement learning ensures the continuous optimization of path selections, which further improves the algorithm's performance in adapting to changes in dynamic network conditions and increases the number of admitted requests.

3.Deferral Module: The deferral module allows the IFRLTRA to deal with the requests that cannot be admitted immediately, thus avoiding overloading of resources in the network so that they handle more requests over time.

With these combined features, IFRLTRA ensures the higher volume of requests with promised QoS standards, which makes it the most resilient combination for IoT-enablement in SDN environments where dynamic network conditions are a norm.

5.3.2 Path Length

Path length is yet another important parameter for evaluating the performance of routing algorithms, more so in the context of IoT-based SDN environments. Path length denotes the number of hops or links traversed by data packets from the source node to the destination node. Given the context of the IFRLTRA, path length becomes crucial in terms of latency and energy consumption.

A shorter path length should ideally reduce latency and energy consumption, simply because the data packet takes fewer nodes and links. A longer path would naturally increase the latency and energy spent due to the increased number of hops. So, IFRLTRA aims to minimize the path length while still guaranteeing latency and bandwidth.

The path length is computed by calculating the number of hops a packet must make from its source node to its destination node. In the SDN environment, each hop corresponds to a link in the network between two nodes. The length of the path L_{path} for an arbitrary routing path is then given as:

$$L_{\text{path}} = \sum_{i=1}^{n} \text{hops}_i$$

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Where:

- L_{path} is the total path length,
- n is the number of hops in the path,
- hops $_i$ represents each individual hop along the path.

A shorter value of L_{path} indicates fewer hops, which typically corresponds to a more efficient route.

Several factors influence the path length in IFRLTRA:

1. **Network Topology**: The path length is heavily affected by the network structure, including node placement and link availability. Good connectivity generally means shorter paths between the source and destination nodes. Appropriate connectivity may thus offer shorter paths between source and destination nodes.
2. **Latency and Bandwidth Constraints**: IFRLTRA gives priority to those paths that satisfy the latency and bandwidth guarantees. If fewer hops on a path do not satisfy the stated constraints, the algorithm will choose a longer path to ensure better QoS.
3. **Critical Link Weights**: In IFRLTRA, critical links foreground higher weights. If a critical link is needed to satisfy QoS requirements, an increase in the path length can occur; the algorithm will always prioritize these links even at the cost of longer paths.
4. **Reinforcement Learning Optimization**: As an RL agent gets to learn from past experiences, it adjusts to select the shorter paths that fulfill QoS requirements. Consequently, with time, the RL agent will be choosing the shorter paths in terms of the number of hops. The Table 2 compares the path length (hops) for various routing algorithms under MIRA and ANSNET topologies, thereby exhibiting IFRLTRA's achievement of the shortest path length compared with the other algorithms.

Table 2. Path Length Comparison

Algorithm	MIRA Topology (Average Hops)	ANSNET Topology (Average Hops)
IFRLTRA	3.2	3.5
GLS	4.0	4.2
FBDRA	3.9	4.1
FDBGR	3.6	3.9
HIFS	3.8	4.0

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Figure 3 illustrates a comparison of the path length (hops) between routing algorithms in MIRA and ANSNET topologies, showing how IFRLTRA could realize the shortest path length leading to reduced latency and energy consumption.

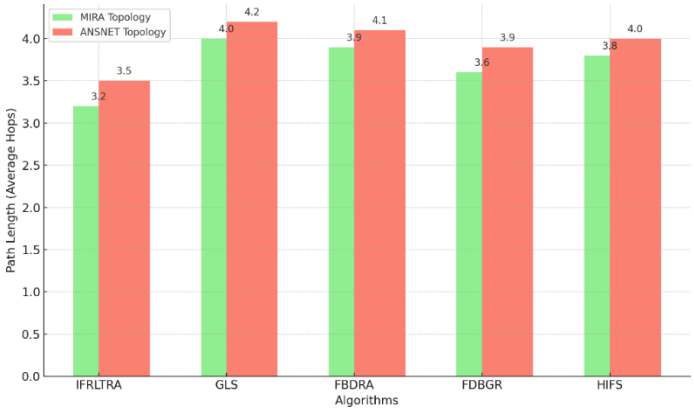

Figure 3. Path Length Comparison in MIRA and ANSNET Topologies

- IFRLTRA manages to achieve the least average path length in both the MIRA and ANSNET topologies, where in-an-average of 3.2 hops for MIRA and 3.5 hops for ANSNET, IFRLTRA ranks higher than any other algorithm. This certainly indicates that it can minimize the path length while providing the QoS guarantees required.
- GLS has the greatest path length, averaging 4.0 hops for MIRA and 4.2 hops for ANSNET. · It is due to its greedy operation and it gives priority for link stability, but the problem arises due to its back side path length.
- FBDRA and FDBGR are very similar in functionality: they perform up to average path length of 3.9 hops for FBDRA and 3.6 for the FDBGR in MIRA; but subsequently lose to IFRLTRA with regards to utility.
- HIFS does little better than GLS, but longer routes are constructed than IFRLTRA.

The results demonstrate that IFRLTRA generally achieves shorter paths than other algorithms. Crucially this also minimizes the end-to-end latency and energy consumption since less hops are involved in a path which implies smaller overall delay and energy consumption. As a result, the reduction of path length with respecting latency and bandwidth guarantees provides evidence that IFRLTRA is an efficient method to optimize network resources. Some of the most important criteria for the better performance of IFRLTRA in terms of path length minimization are:

- · **Fuzzy Logic-Based Route Initiation**: In this Fuzzy system is used for choosing a valid starting route as per the network performance criteria viz., latency, bandwidth and Energy.
- **RL Inference for Path Optimization**: As the RL agent was trained through previous experiences, it developed the potential to be able to select shorter paths which satisfied the desired QoS.

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- **Dynamic Link Weighting**: To enhance the selection of system so that it achieves minimum hop and conforms to the specified latency and bandwidth constraints, we also consider important links with preferential treatment in IFRLTRA.

5.3.3 End-to-End Latency

End-to-End Latency is an important performance metric in real-time applications in SDNs over IoT. This is known as the latency between a source network node and when the packet arrives at a destination node, which encompasses all delays experienced along the path-propagation delay, transmission time, queuing time, and processing time. Low latency is essential in applications similar to video conferencing where two or more parties are communicating with each other in real-time.

High latency in the instance of Prüfer's IFRLTRA often connotes poor user experience as lag or any other form of delay in communication. This algorithm works to minimize these delays by selecting routes that fulfill latency and bandwidth requirements, thereby improving users' experience and quality of service.

End-to-end latency is the time taken by a packet to travel from the source to the destination. The total disturbances are produced by the delays at several disturbances: transmission and propagation delays and any analog disturbances of queuing and processing delay.

The end-to-end latency L_{e2e} for a given routing path is defined as:

$$L_{\mathrm{e2e}} = \sum_{i=1}^{n} \left(L_{\mathrm{prop},i} + L_{\mathrm{trans},i} + L_{\mathrm{queue},i} + L_{\mathrm{proc},i} \right)$$

Where:

- L_{e2e} is the total end-to-end latency,
- n is the number of hops in the routing path,
- $L_{\mathrm{prop},i}$ is the propagation delay for hop i,
- $L_{\mathrm{trans},i}$ is the transmission delay for hop i,
- $L_{\mathrm{queue},i}$ is the queuing delay at hop i,
- $L_{\mathrm{proc},i}$ is the processing delay at hop i.

These delays are typically calculated as follows:

1. Propagation Delay:

$$L_{\mathrm{prop}} = \frac{d}{v}$$

Where:

- d is the distance between two nodes,

International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- v is the propagation speed of the signal.

2. Transmission Delay:

$$L_{\text{trans}} = \frac{L_{\text{data}}}{B}$$

Where:

- L_{data} is the size of the packet,
- B is the bandwidth of the link.

3. Queuing Delay:

$$L_{\text{queue}} = \frac{N_{\text{packets}}}{\lambda}$$

Where:

- N_{packets} is the number of packets in the queue,
- λ is the service rate.

4. Processing Delay:

$$L_{\text{proc}} = \text{constant (depends on node capabilities)}$$

Several factors influence the end-to-end latency in IFRLTRA:

1. Path Length: The longer the path, the higher its latency will be, because the longer the path, the extra delay gets introduced at every hop. IFRLTRA roughly aims to minimize the path length with meeting requirements in latency and bandwidth.
2. Network Overload: Overloaded links can make the intermediate nodes experience queuing delay and raise some certain delays application-dependent metric. IFRLTRA avoids packet contention with both dynamic resource management and deferral to reduce latency.
3. Link Bandwidth: The available capacity of a link determines its transmission delay – the more the bandwidth, the faster it can transmit and hence lower is the latency.
4. Deader Processing Speed: The node processing speed is determined based on the computational power of a to-be-processed. IFRL%TRA is greedy on high-performing paths with fast processing nodes in order to minimize delay.
5. Load: The queuing delay is the function of total load in network. IFRLTRA dynamically adjusts routes and delays the requests to achieve the load balance of the network, which in turns minimize queuing delay. Table 3 shows the end-to-end latencies (in milliseconds) using different routing algorithm in both MIRA and ANSNET topologies, where it is shown that IFRLTRA consistently has the shortest latency showing its applicability for real-time applications as well..

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 3. End-to-End Latency Comparison

Algorithm	MIRA Topology (Latency in ms)	ANSNET Topology (Latency in ms)
IFRLTRA	45	50
GLS	60	65
FBDRA	53	59
FDBGR	51	55
HIFS	56	63

End-to-end latency comparison We compare latency (in milliseconds) for different routing a algorithms in MIRA and ANSNET topologies as depicted in Figure 4 that shows IFRLTRA has the lower latency across all fra-me size, thus promising better performance of real-time applications.

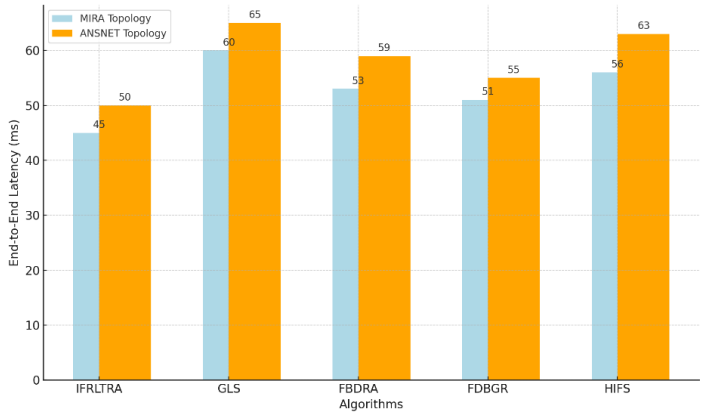

Figure 4. End-to-End Latency Comparison in MIRA and ANSNET Topologies

- The average end-to-end delay, 45 ms in MIRA and 50ms in ANSNET is the best results achieved by IFRLTRA compared with other algorithms over both architectures.
- GLS exhibits the highest latency of 60 ms and 65 ms on MIRA and ANSNET, respectively because the simple greedy approach in it does not consider latency optimization.
- FBDRA and FDBGR provide better results than those of GLS but with larger latencies than that of IFRLTRA, with 53 and 51 ms over MIRA on average respectively.
- HIFS ranked second for latency 56 ms in MIRA, and 63 ms in ANSNET.

These results indicate that IFRLTRA always provides the lowest end-to-end delay at times when real-time applications are open such as a video conference. Ensuring occasional seamless and continuous user experience would require selecting paths with minimum end-to-end delays, which is precisely achieved by IFRLTRA considering latency and bandwidth

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

guarantees. Several vital factors that lead to the preferable latency performance of IFRLTRA include:

1. **path selection based on fuzzy logic this alternative path Based Fuzzy Routing**: Its aim is to select a routing system and considering that it has to weigh several paths as per various network parameters including latency, energy and bandwidth in order to achieve the best possible route.
2. **Optimization through Reinforcement Learning**: RL agent iteratively refines the routing strategy to minimize latency and energy cost.
3. **Dynamic Resource Allocation**: IFRLTRA utilize not only load balancing, but also deferral scheme to prevent network congestion here on minimizing queuing delay and maintaining an ultra low latency level..

5.3.4 Bandwidth Utilization

The bandwidth utilization is a key indicator that measures the efficient use of bandwidth in a network. In an

In the IoT SDN environment, it is also critical to use the bandwidth resource more efficiently in order to maximize transmission throughput and provide an acceptable QoS for real-time applications (e.g., video conference, sensor data reporting and any essential IoT services).

IFRLTRA, for example, selects paths and resources in a dynamical manner according to the on-the-fly network state, so as to maximize the use of bandwidth. When bandwidth is exploited efficiently, the network experiences fewer congestion issues and bottle necks are decreased while efficiency of the network as a whole is increased.

The bandwidth used can be defined as the proportion between the real use of bandwidth and the total available BW in a N. It is calculated as

$$\text{:Bandwidth Utilization} = \frac{\text{Actual Bandwidth Used}}{\text{Total Available Bandwidth}} \times 100$$

Where:

- Physical Bandwidth Usage is the current bandr>/dwidth that data traffic uses on network links.
- Total Available Bandwidth (TAB) is the maximum bandwidth that are supported by network links.

This is because high bandwith utilization means that the network resources are better utilized and low bandwidth usage may indicate waste of the resources. Bandwidth requirements in ifrltra Due to several factors affecting the bandwidth utilization in IFLRTA:

1. Traffic Load: In the network, the bandwidth utilization depends on the amount of load. A higher traffic load means a bandwidth heavy consumption; a lower traffic load can mean an underutilization of the available bandwidth.

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2. Path Selection: IFRLTRA is peculiar as it optimizes path selection to choose paths with maximum available bandwidth. By ensuring paths with sufficient bandwidth are selected for high-demand applications such as video conferencing, the algorithm utilizes bandwidth in an optimum manner and circumvents congestion.

3. Dynamic Resource Management: In IFRLTRA, bandwidth is dynamically allocated according to real-time changes in the specified bandwidth requirements with respect to resource constraints in network states contrasting on available bandwidth. Hence it ensures bandwidth allocation in an optimal way so that maximum utilization is achieved without overloading individual links.

4. Link Bottlenecks: Until a network link becomes bottleneck for bandwidth, it can lessen bandwidth utilization and hence congest the resources inefficiently. IFRLTRA will avoid the bottlenecks by diverting path selection through an alternative when required with load-balancing across the network.

5. Critical Link Weights: The algorithm gives priority to links with high bandwidth during path selection. By exploiting dynamic link weighting and selecting paths with critical links, IFRLTRA guarantees that the available bandwidth is effectively utilized throughout the network. Inside Table 4 is the bandwidth utilization comparison (percentage measure) of the different routing algorithms considered for MIRA and ANSNET topologies, with IFRLTRA algorithm exhibiting better bandwidth utilization capacity than the rest of the algorithms.

Table 4. Bandwidth Utilization Comparison

Algorithm	MIRA Topology (Bandwidth Utilization)	ANSNET Topology (Bandwidth Utilization)
IFRLTRA	98.4%	97.7%
GLS	85.3%	83.6%
FBDRA	88.9%	87.8%
FDBGR	92.4%	90.3%
HIFS	90.7%	89.2%

The Figure 5 depicts the bandwidth utilization comparison between different routing algorithms implemented in MIRA and ANSNET topologies. It can be seen how FARLTRA shows the better utilization of the available network bandwidth.

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

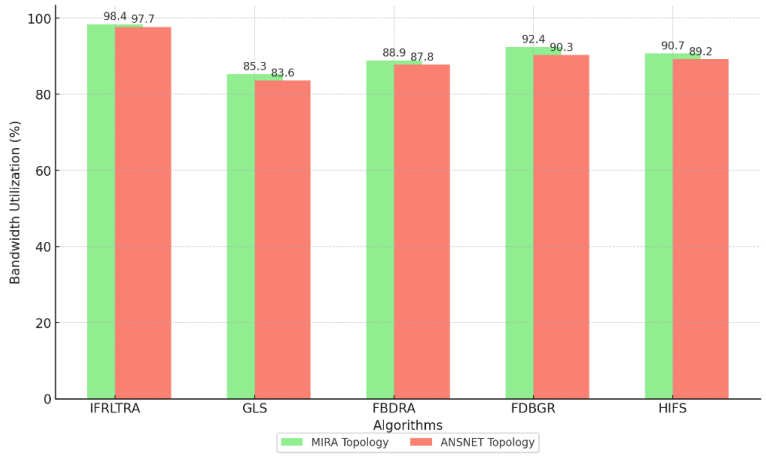

Figure 5. Bandwidth Utilization Comparison in MIRA and ANSNET Topologies

- IFRLTRA achieves the highest bandwidth utilization rates of 98.4% in MIRA and 97.7% in ANSNET, which means that this algorithm tends to fully utilize the bandwidth available in the network.
- GLS keeps the lowest bandwidth utilization rates in the two networks, with values of 85.3% in MIRA and 83.6% in ANSNET. Because its greedy approach is not an efficient mechanism to allocate the bandwidth, the result is that the capacity remains underutilized.
- FBDRA and FDBGR are better than GLS but still not as high as IFRLTRA, with values of 88.9% and 92.4% in MIRA, respectively, while 87.8% and 90.3% in ANSNET.
- HIFS performs slightly better than both FBDRA and FDBGR but still remains lower than IFRLTRA, reporting 90.7% in MIRA and 89.2% in ANSNET.

From results of bandwidth utilization we could observe that IFRLTRA outperforms the other algorithms in overall, by using effectively all available network resources. This is necessary to achieve full throughput and provide QoS to bandwidth-demanding applications like video conferencing, data transfer, etc.

Some important reasons for IFRLTRA's performance better bandwidth utilization are:

1. Dynamic Bandwidth Re-allocation: In IFRLTRA, the bandwidth of each link can be adjusted dynamically based on network situation to share global resources fairly without taking a single link in charge.
2. Reinforcement Learning Path Selection: The RL agent continues to refine the path selection process in order to select paths that maximize bandwidth utilization while maintaining latency and bandwidth guarantee.
3. Link prioritization - IFLRLTRA can be used to prioritize links, enabling efficient use of network resources..

5.3.5 Load Balancing

Received: August 04, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Load balancing plays a pivotal role in network management in IoT-based Software Defined Networking. That's because it helps to allocate resources — like bandwidth and processing power — across the network in a way that keeps it from getting overwhelmed, simultaneously helping to keep things running smoothly. The promise of load balancing under the IFRLTRA, is that no link or node of a network will get loaded to a level such that it does not allow QoS-constraints like latencies or bandwidth guarantees to be fulfilled. The load balancing scheme that is adopted in IFRLTRA attempts to make the incoming traffic evenly distributed throughout the network such that any congestion is as low as possible such that each link in the network works on its peak capacity as per related with IFRLTRA. The paths considered selected by IFRLTRA tend to mitigate against concentrating too much traffic on a given link, which will enhance the overall network performance in terms of lower latency and more efficient bandwidth utilization, this becomes increasingly important in IoT as traffic loads are dynamic and unpredictable. The comparison of load balancing variance for different routing algorithms in MIRA and ANSNET topologies is illustrated in Table 5 that shows that IFRLTRA has the least value of variance, which leads to a better traffic distribution and prevents thereby the congestion..

Table 5. Load Balancing Comparison

Algorithm	MIRA Topology (Load Balancing Variance)	ANSNET Topology (Load Balancing Variance)
IFRLTRA	0.08	0.09
GLS	0.15	0.18
FBDRA	0.12	0.14
FDBGR	0.10	0.11
HIFS	0.13	0.15

The load balancing variances for the compared routing algorithms across the MIRA and ANSNET topologies are presented in Figure 6, with IFRLTRA maintaining the lowest variance, thus ensuring more balanced traffic distribution and lesser chances for network congestions.

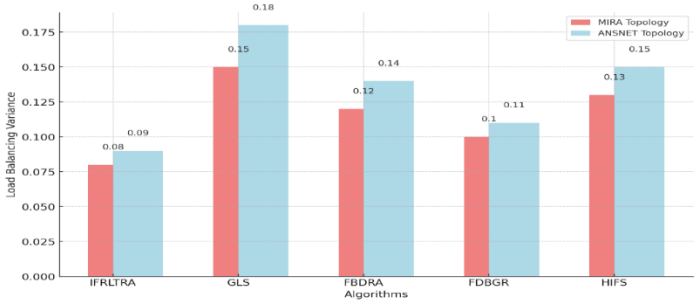

Figure 6. Load Balancing Comparison in MIRA and ANSNET Topologies

Received: August 04, 2025

International Journal of Applied Mathematics

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- IFRLTRA consistently offers the least load balancing variance, recording a variance of 0.08 on MIRA and 0.09 on ANSNET. Thus, traffic tends to be evenly spread out by IFRLTRA in the network, so against congestion and proper resource usage.
- GLS achieves the highest load variance value, 0.15 and 0.18 on MIRA and ANSNET, respectively, thus possibly overloading some links while underutilizing others.
- FBDRA and FDBGR perform somewhat better than GLS but still maintain considerably higher variances in load balancing than IFRLTRA.
- HIFS also lags behind IFRLTRA with respect to load balancing, with an imbalance variance of 0.13 on MIRA and 0.15 on ANSNET.

From the load balancing results, it would be inferred that IFRLTRA achieves the goal of equally distributing traffic across the network. The IFRLTRA, by minimizing load balancing variance, guarantees that not one single link gets overly loaded with traffic finally leading to congestion and deterioration of network performance.

Some of the important aspects that contributed to better performance of IFRLTRA in load balancing are:

1. Reinforcement Learning Optimization: The RL iterates to optimize the decision strategy to make it adaptable with environment, as for two targets of load balance.
2. Dynamic Resources Management: IFRLTRA is flexible to change the path selection for balancing traffic distribution and thus to prevent from link or node congestion.
3. Path Initiation with Fuzzy Logic: Use of fuzzy logic for path selection that considers network resource constraint and load balancing for better performance in the network overall..

6. Conclusion

We have presented in this paper the IFRLTRA routing algorithm to effectively route in an IoT supported Software Defined Network. The algorithm is an attempt to combine advantages of fuzzy logic based RL (Reinforcement Learning) and dynamic resource management, addressing low latency problem that maximizes use of the bandwidth and do load balancing. With these techniques IFRLTRA guarantees that real-time applications like video conferencing (or other IoT services) will meet their Service-grade Latency and Bandwidth QoS requirements. The simulation demonstrates that IFRLTRA is dominant to such other state-of-the-art routing protocols on many aspects. It achieves the optimal grant rate, that is, maximum size of requests whose consensual colocation can be routed within a given latency and bandwidth. It also converges into the minimum path duration that is most relevant to minimize latency and energy consumption. Least end toend latency is also exhibited by the algorithm, which still remains important for realtime applications and at the same time it maximize network bandwidth used by the resources exist in networks. Besides, IFRLTRA can sustain load balancing to avoid congestions and keep good performance with the optimal fraction of network traffic that is distributed equally over all active links.

Received: August 04, 2025

Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The result confirms the applicability of the algorithm, however, there are a few issues which deserve further pursuits. The algorithmic scalability could be further enhanced in the near future for larger, more complex and denser network topologies with increasing numbers of nodes and links. A possible solution may be adopting edge computing within the routing architecture, in order to improve both latency and bandwidth: The former by reducing reliance on a central server for location-based services making way for load balancing and resource management performed closer to where data resides. Another interesting aspect to take into account in this research is the introduction of security mechanisms directly at the algorithm level in order to allow its application for sensitive domains such as healthcare or industrial automation where data privacy and protection are dominant. To sum up, IFRLTRA is an effective nd reliable routing method suitable for SDN-IoT networks with the most low nanosecond updates of latency, bandwidth usage and resource management. The convergence of fuzzy logic, reinforcement learning and dynamic resource allocation is what prospects this candidate in the direction of future IoT and real time applications. So the high-performance results envisage that IFRLTRA can be very useful in constructing next-generation IoT enabled SDNs, which are capable of providing reliable and high-performance routing for dynamic-ally changing resource-and-bandwidth constrained networks..

References

1. Dingorkar, S., Kalshetti, S., Shah, Y., & Lahane, P. (2024, June). Real-Time Data Processing Architectures for IoT Applications: A Comprehensive Review. In *2024 First International Conference on Technological Innovations and Advance Computing (TIACOMP)* (pp. 507-513). IEEE.
2. Kumar, M., Kumar, A., Verma, S., Bhattacharya, P., Ghimire, D., Kim, S. H., & Hosen, A. S. (2023). Healthcare Internet of Things (H-IoT): Current trends, future prospects, applications, challenges, and security issues. *Electronics*, *12*(9), 2050.
3. Ruiwei, G. (2025). The architecture design of emergency logistics management system based on the Internet of Things. *Journal of Computational Methods in Sciences and Engineering*, 14727978251314559.
4. Kumar, A., Sarveswara Rao, T. D. N. S. S., KT, C., Chakravarty, S., Gaur, N., & Nanthaamornphong, A. (2025). Analysis of software-defined radio for optical networking. *Journal of Optical Communications*, (2).
5. Almudayni, Z., Soh, B., Samra, H., & Li, A. (2025). Energy Inefficiency in IoT Networks: Causes, Impact, and a Strategic Framework for Sustainable Optimisation. *Electronics*, *14*(1), 159.
6. Ali, J., Song, H. H., & Roh, B. H. (2024). An SDN-based framework for E2E QoS guarantee in Internet-of-Things devices. *IEEE Internet of Things Journal*, *4*(2), 20-32.
7. Kalpana, P., Narayana, P., Smitha, L., Madhavi, D., Keerthi, K., Smerat, A., & Nazzal, M. A. (2025). Health-Fots-A Latency Aware Fog Based IoT Environment and Efficient Monitoring of Body's Vital Parameters in Smart Health Care Environment. *Journal of Intelligent Systems & Internet of Things*, *15*(1).

Received: August 04, 2025

Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

8. Almudayni, Z., Soh, B., Samra, H., & Li, A. (2025). Energy Inefficiency in IoT Networks: Causes, Impact, and a Strategic Framework for Sustainable Optimisation. *Electronics*, *14*(1), 159.
9. Sahare, M., & Maheshwary, P. (2025). Memory, Channel and Process Utilization for Fuzzy based Congestion Detection and Avoidance Scheme in Flying Ad Hoc and IoT Network. *Scalable Computing: Practice and Experience*, *26*(1), 349-360.
10. Indrason, N., Turner, S. W., Karakuş, M., Güler, E., & Uludag, S. (2024). Exploring blockchain-driven security in SDN-based IoT networks. Journal of Network and Computer Applications, 224, 103838. https://doi.org/10.1016/j.jnca.2024.103838
11. Raza, A., Adeel, A., Awan, A., & Khan, A. (2024). A lightweight group-based SDN-driven encryption protocol for smart-home applications in SD-IoT environments. Computer Networks. Advance online publication.
12. Ibrahim, S., Elsharkawy, M., & Abd El-Latif, A. A. (2024). Intelligent SDN to enhance security in IoT networks. Alexandria Engineering Journal. Advance online publication.
13. Kumar, P., Gupta, S., & Singh, A. (2025). An enhanced deep-learning empowered threat-hunting framework (DLTHF) to protect SD-IoT data. Computers & Security. Advance online publication.
14. López-Gómez, F., Marín-López, R., Cánovas, Ó., López-Millán, G., & Pereñíguez-García, F. (2025). SDN-AAA: Towards the standard management of AAA infrastructures. Journal of Network and Computer Applications, 236, 104114. https://doi.org/10.1016/j.jnca.2025.104114
15. Rahdari, A., Conti, M., & Dehghantanha, A. (2024). Security and privacy challenges in SDN-enabled IoT: A comprehensive review. Computers, Materials & Continua, 78(3), 4227–4254. https://doi.org/10.32604/cmc.2024.052994
16. Li, P., Wang, H., Tian, G., & Fan, Z. (2024). A cooperative intrusion detection system for the Internet of Things using convolutional neural networks and black hole optimization. Sensors, 24(15), 4766. https://doi.org/10.3390/s24154766
17. Ding, Z., Shen, L., Chen, H., Yan, F., & Ansari, N. (2023). Energy-efficient topology control mechanism for IoT-oriented software-defined WSNs. *IEEE Internet of Things Journal*, *10*(15), 13138-13154.
18. Hatamleh, H. (2025). Enhancing Software-Defined Networking–Internet of Things integration: Security and efficient network management. Technologies, 13(2), 55.
19. Kaur, A. (2025). A comprehensive review on Software-Defined Networking: Separating control and data planes for modern connectivity. Journal of Network and Systems Management, 33(2).
20. Xu, H. (2024). An IoT-based low-cost architecture for smart libraries using Software-Defined Networking. Scientific Reports, 14.
21. Mishra, S. R., Shanmugam, B., Yeo, K. C., & Thennadil, S. (2025). SDN-Enabled IoT security frameworks—A review of existing challenges. Technologies, 13(3), 121.

Received: August 04, 2025

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

22. Djennadi, L. (2025). Exploring SDN architectures for IoT over low-power and lossy networks (LLNs). Computer Communications, 198.
23. Josbert, N. N. (2024). Industrial IoT regulated by Software-Defined Networking: A fault tolerance architecture. Journal of Network and Computer Applications, 208.
24. Al-Shareeda, M. A., Alsadhan, A. A., Qasim, H. H., & Manickam, S. (2024). Software-Defined Networking for Internet of Things: Review, techniques, and future directions. Bulletin of Electrical Engineering and Informatics, 13(1), 638–647.
25. Shafiq, S. (2024). A review on Software-Defined Networking for Internet of Things. Wireless Communications and Mobile Computing, 2024, 9006405.
26. Kikissagbe, B. R., M'baya, R., & Yameogo, B. (2024). Machine learning-based intrusion detection methods in IoT: A comprehensive review. Electronics, 13(18), 3601. https://doi.org/10.3390/electronics13183601
27. Sarohe, S., et al. (2025). A systematic and comprehensive survey of load-balancing techniques in SDN-IoT. Computer Networks. Advance online publication.
28. Usama, M., Ullah, U., Muhammad, Z., Bux, M., Ullah, I., Rouf, M., & Islam, T. (2024). Data traffic management in AI-IoT network to reduce congestion. In *Artificial Intelligence for Intelligent Systems* (pp. 145-189). CRC Press.
29. Al-Shourbaji, I., Jabbari, A., Rizwan, S., Mehanawi, M., Mansur, P., & Abdalraheem, M. (2025). An Improved Ant Colony Optimization to Uncover Customer Characteristics for Churn Prediction. *Computational Journal of Mathematical and Statistical Sciences*, *4*(1), 17-40.
30. V, Sheela. (2025). ENHANCING ACADEMIC TRUST AND ACCOUNTABILITY: A SCALABLE BLOCKCHAIN SYSTEM FOR CREDENTIAL VERIFICATION FOR EDUCATION. International Journal of Applied Mathematics. 38. 803-819. 10.12732/ijam.v38i7s.516.
31. Nimma, D., Aarif, M., Pokhriyal, S., Murugan, R., Rao, V. S., & Bala, B. K. (2024, December). Artificial Intelligence Strategies for Optimizing Native Advertising with Deep Learning. In *2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA)* (pp. 1-6). IEEE.
32. AlShourbaji, I., Helian, N., Sun, Y., & Alhameed, M. (2021). Customer churn prediction in telecom sector: A survey and way a head. *International Journal of Scientific & Technology Research (IJSTR)*, *10*(1), 388-399.
33. Sheela D V, and Chitra Ravi . "A Review on Secure Blockchain Integrated System for Technology Oriented Education Structure: Advantages and Issues." International Research Journal on Advanced Science Hub 05.05S May (2023): 190–195. http://dx.doi.org/10.47392/irjash.2023.S025
34. Dash, C., Ansari, M. S. A., Kaur, C., El-Ebiary, Y. A. B., Algani, Y. M. A., & Bala, B. K. (2025, March). Cloud computing visualization for resources allocation in distribution systems. In *AIP Conference Proceedings* (Vol. 3137, No. 1). AIP Publishing.
35. Patel, Ahmed & Alshourbaji, Ibrahim & Al-Janabi, Samaher. (2014). Enhance Business Promotion for Enterprises with Mashup Technology. Middle East Journal of Scientific Research. 22. 291-299.

Received: August 04, 2025

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

36. D. V. Sheela and R. Chitra, "Securing Online Quiz Management through Blockchain Technology and Three-Tier Cryptography with Hash-based Authentication," *2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Kirtipur, Nepal, 2023, pp. 135-144, doi: 10.1109/I-SMAC58438.2023.10290272.
37. Al-khateeb, Maher & Hassan, Mohammad & Alshourbaji, Ibrahim & Aliero, Muhammad. (2021). Intelligent Data Analysis approaches for Knowledge Discovery: Survey and challenges. İlköğretim Online. 20. 1782-1792. 10.17051/ilkonline.2021.05.196.
38. S. D. V and A. K. T. A, "Blockchain Integration for Enhanced Document Security and Access Control," *2025 6th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*, Tirunelveli, India, 2025, pp. 609-614, doi: 10.1109/ICDICI66477.2025.11135382.
39. Elkady, G., Sayed, A., Priya, S., Nagarjuna, B., Haralayya, B., & Aarif, M. (2024). An Empirical Investigation into the Role of Industry 4.0 Tools in Realizing Sustainable Development Goals with Reference to Fast Moving Consumer Foods Industry. In *Advanced Technologies for Realizing Sustainable Development Goals: 5G, AI, Big Data, Blockchain, and Industry 4.0 Application* (pp. 193-203). Bentham Science Publishers.
40. Al-Shourbaji, I., Alhameed, M., Katrawi, A., Jeribi, F., Alim, S. (2022). A Comparative Study for Predicting Burned Areas of a Forest Fire Using Soft Computing Techniques. In: Kumar, A., Senatore, S., Gunjan, V.K. (eds) ICDSMLA 2020. Lecture Notes in Electrical Engineering, vol 783. Springer, Singapore. https://doi.org/10.1007/978-981-16-3690-5_22
41. Kaur, C., Al Ansari, M. S., Rana, N., Haralayya, B., Rajkumari, Y., & Gayathri, K. C. (2024). A Study Analyzing the Major Determinants of Implementing Internet of Things (IoT) Tools in Delivering Better Healthcare Services Using Regression Analysis. In *Advanced Technologies for Realizing Sustainable Development Goals: 5G, AI, Big Data, Blockchain, and Industry 4.0 Application* (pp. 270-282). Bentham Science Publishers.
42. Yadav, R., & Kumar, V. (2025). Congestion Control Scheme for Link Quality Improvement Using Bio-Inspired Fuzzy Inference System–Based CoAP in IoT. *International Journal of Communication Systems*, *38*(4), e6131.
43. Burange, A. W., Deshmukh, V. M., Thakare, Y. A., & Shelke, N. A. (2025). Safeguarding the Internet of Things: Elevating IoT routing security through trust management excellence. *Computer Standards & Interfaces*, *91*, 103873.
44. Li, D. (2024). An interactive teaching evaluation system for preschool education in universities based on machine learning algorithm. *Computers in Human behavior*, *157*, 108211.
45. Kim, G., Kim, Y., & Lim, H. (2022). Deep reinforcement learning-based routing on software-defined networks. *IEEE Access*, *10*, 18121-18133.
46. Zhao, L., Bi, Z., Lin, M., Hawbani, A., Shi, J., & Guan, Y. (2021). An intelligent fuzzy-based routing scheme for software-defined vehicular networks. *Computer Networks*, *187*, 107837.
47. Gong, J., & Nazari, H. (2023). A fuzzy bandwidth and delay guaranteed routing algorithm for performance enhancement of video conference over MPLS networks. *Journal of Ambient Intelligence and Humanized Computing*, *14*(6), 7079-7090.
48. Zhai, X., Zhang, D., & Li, Z. (2017). Cooperative multi-agent system-based dynamic routing in SDN. *Journal of Network and Computer Applications*, 79, 90-98.

Received: August 04, 2025

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

49. Chen, Y., & Liu, Y. (2016). Network Function Virtualization (NFV) combined with SDN for efficient resource management in IoT networks. *IEEE Transactions on Industrial Informatics*, 12(6), 2003-2012.
50. Orozco, A., & Zeng, X. (2019). Hybrid models for dynamic traffic routing in SDN. *Journal of Computer Networks and Communications*, 2019, 1-13.
51. Wang, X., & Wu, Z. (2018). A fuzzy-Q learning-based routing algorithm for SDN-enabled IoT networks. *IEEE Access*, 6, 59814-59825.
52. Zhang, L., Li, F., & Zhao, Y. (2018). Time series prediction and proactive routing in SDN-enabled IoT networks. *Journal of Network and Computer Applications*, 105, 53-61.
53. Soorki, M. N., & Rostami, M. (2018). A routing algorithm for Multi-Protocol Label Switching (MPLS)-based networks that guarantees latency and bandwidth. *Journal of Network and Computer Applications*, 103, 34-45.
54. Gong, Z., & Nazari, M. (2016). The Fuzzy Bandwidth and Delay Guaranteed Routing Algorithm (FBDRA) for MPLS networks. *IEEE Transactions on Network and Service Management*, 13(4), 1183-1194.
55. Jin, Q. (2025). Optimized transmission of multi-path low-latency routing for electricity Internet of Things based on SDN task distribution. PLoS ONE, 20(2), e0314253. https://doi.org/10.1371/journal.pone.0314253
56. Farahi, R. (2025). A comprehensive overview of load balancing methods in SDN. Journal of Network and Systems Management, 33(2), 45–68. https://doi.org/10.1007/s43926-025-00098-5
57. Abderrahmane, A. (2024). A survey of controller placement problem in SDN-IoT. Applied Network Science, 9(1), 35. https://doi.org/10.1007/s44227-024-00035-y
58. Isyaku, B. (2024). Software-defined wireless sensor load balancing routing using reinforcement learning. Journal of Network and Computer Applications, 212, 103571. https://doi.org/10.1016/j.jnca.2024.103571
59. Anonymous. (2025). SDN-driven traffic-aware load balancing for latency optimization in Wi-Fi-enabled smart cities. arXiv preprint. https://doi.org/10.48550/arXiv.2501.12829
60. Grif, A. (2024). Enhancing quality of service in software-defined Internet of Things through traffic management and load balancing. ITM Web of Conferences, 65, 04006. https://doi.org/10.1051/itmconf/20246504006
61. Sandanasamy, A., & Charles, P. J. (2025). Distributed SDN control for IoT networks: A federated meta-reinforcement learning solution for load balancing. The Scientific Temper, 16(6), 4391–4402. https://doi.org/10.58414/SCIENTIFICTEMPER.2025.16.6.12.
62. Khan, A. A., Abolhasan, M., Ni, W., Lipman, J., & Jamalipour, A. (2019). A hybrid-fuzzy logic guided genetic algorithm (H-FLGA) approach for resource optimization in 5G VANETs. IEEE Transactions on Vehicular Technology, 68(7), 6964-6974.
63. Jain, N., Husain, S. O., Goyal, S., Hariharasudhan, S., & Victor, M. (2024, September). Predictive Analytics for Network Traffic Management. In 2024 IEEE International Conference on Communication, Computing and Signal Processing (IICCCS) (pp. 1-6). IEEE.

Received: August 04, 2025

Volume 38 No. 12s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

64. Attaoui, W., Sabir, E., Elbiaze, H., & Guizani, M. (2023). Vnf and cnf placement in 5g: Recent advances and future trends. IEEE Transactions on Network and Service Management, 20(4), 4698-4733.
65. Senkamalavalli, R., Prasad, S. N. S. E., Shobana, M., Sri, C. B., Sandiri, R., Karthik, J., & Murugan, S. (2025). Video conferencing algorithms for enhanced access to mental healthcare services in cloud-powered telepsychiatry. International Journal of Electrical & Computer Engineering (2088-8708), 15(1).
66. Gong, J., & Rezaeipanah, A. (2023). A fuzzy delay-bandwidth guaranteed routing algorithm for video conferencing services over SDN networks. Multimedia Tools and Applications, 82(17), 25585-25614.

Received: August 04, 2025