# HAP-Assisted Disaster Relief Communication with AI-Driven Resource Allocation

Arunima Sharma[1] Shashikala S[2] Mallika K S[2] Prabhavati Angadi[3] Sowbhagya Krishnamurthy[4]
[1]*Faculty of Computer Science and Information Technology*, *Universiti Putra Malaysia*, Malaysia
aru.92@rediffmail.com
[2]*Department of Mathematics*, *Global Academy of Technology*, Bengaluru, India
shashikala.s@gat.ac.in, mallikaks@gat.ac.in
[3]*Department of Mathematics*, *Dr. Ambedkar Institute of Technology*, Bengaluru, India
prabhavatiangadi71@gmail.com
[4]*Department of Science and Humanities, School of Advanced Studies*, *S-VYASA Deemed to be University*, Bengaluru, India
sowbhagyaimpana2017@gmail.com

***Abstract*—In the wake of natural disasters, terrestrial communication infrastructure is often rendered inoperative, severely hampering emergency response efforts. This paper proposes a resilient communication framework leveraging High-Altitude Platforms (HAPs) to provide rapid, wide-area connectivity in disaster-stricken regions. We integrate a reinforcement learning (RL)-based algorithm that dynamically allocates bandwidth and adjusts coverage in real time, enabling efficient and adaptive resource management based on environmental feedback and user demand. The proposed system enhances communication reliability, reduces latency, and optimizes spectral efficiency under uncertain and evolving disaster scenarios. Simulation results demonstrate the effectiveness of the RL-driven HAP network in maintaining robust communication links and improving overall disaster response coordination.**

***Index Terms*—High-Altitude Platforms (HAPs), Disaster Relief Communication, Reinforcement Learning, Resource Allocation, Emergency Networks, AI-Driven Optimization.**

## I. Introduction

Natural and man-made disasters such as earthquakes, floods, and wildfires often disrupt terrestrial communication infrastructure, severely hindering emergency response and coordination [1], [2]. In such scenarios, rapidly deployable and resilient communication systems are essential for saving lives and managing resources effectively.

While terrestrial networks are prone to physical damage, satellite systems—though robust—suffer from high latency, limited bandwidth, and inflexible deployment [4]. These limitations highlight the need for alternative platforms that offer scalable, adaptive, and low-latency communication in disaster-affected regions.

High-Altitude Platforms (HAPs), operating at altitudes of 17–22 km, present a promising solution due to their wide coverage, low latency, and faster deployment compared to satellites [3]. Acting as airborne base stations, HAPs can establish temporary communication backbones to connect isolated users with emergency services.

However, disaster environments are highly dynamic, with fluctuating user demand, mobility, and infrastructure damage. Static resource allocation in such conditions leads to inefficiencies and degraded service. To address this, we propose an AI-driven framework that employs reinforcement learning (RL) for real-time, adaptive optimization of bandwidth and coverage.

### A. Novelty and Contributions

This paper presents a novel HAP-assisted disaster communication system integrated with an RL agent for intelligent resource management. The main contributions include:

- **System Design:** A modular architecture comprising HAP nodes, ground terminals, and a centralized RL controller (see Fig. 1).
- **RL-Based Optimization:** The resource allocation problem is modeled as a Markov Decision Process (MDP), where the RL agent observes system states (e.g., user density, signal quality) and takes actions such as bandwidth reallocation and beam steering to maximize coverage and efficiency.
- **Simulation and Validation:** The system is implemented in NS-3 and MATLAB, simulating diverse disaster scenarios. Performance is evaluated using metrics like coverage ratio, packet delivery ratio, and latency (see Table **??** and Fig. **??**).

### B. Scientific Rationale

The RL agent is trained using the Proximal Policy Optimization (PPO) algorithm, selected for its stability and efficiency in continuous action spaces. The reward function at time $t$ is defined as:

$$R_t = \alpha \cdot C_t - \beta \cdot L_t - \gamma \cdot B_t \tag{1}$$

where $C_t$ is the coverage ratio, $L_t$ is the average latency, $B_t$ is the bandwidth overhead, and $\alpha, \beta, \gamma$ are tunable weights. This formulation encourages the agent to prioritize wide, low-latency coverage while minimizing bandwidth inefficiencies.

### C. Paper Organization

The rest of the paper is structured as follows: Section II reviews related work on HAP networks and AI in disaster communication. Section III outlines the system architecture.

Section IV details the RL framework. Section V describes the simulation setup. Section VI presents results and analysis. Section VII discusses the prototype implementation. Section VIII concludes with future directions.

## II. Related Work

Disaster communication systems have traditionally relied on terrestrial infrastructure, which is highly vulnerable during natural calamities. Satellite systems offer broader coverage but are hindered by high latency, limited bandwidth, and deployment costs [5]. UAVs have emerged as agile, low-cost alternatives capable of forming Flying Ad Hoc Networks (FANETs) for temporary connectivity [6], though they face limitations such as short flight duration, payload constraints, and unstable links due to mobility [7].

High-Altitude Platforms (HAPs), operating in the stratosphere, offer a promising middle ground with wide-area coverage and low latency. Recent work has explored their use in emergency and rural communication [8]. However, challenges remain:

- **Spectrum Interference:** Shared spectrum with terrestrial systems leads to interference and reduced SINR [9].
- **Rigid Resource Allocation:** Most HAP systems rely on static or rule-based allocation, lacking adaptability to dynamic conditions.
- **Coordination Complexity:** Managing multiple HAPs in disaster zones requires intelligent orchestration to avoid redundancy and ensure optimal coverage.

AI, particularly Reinforcement Learning (RL), has shown strong potential in wireless communication tasks such as spectrum access, power control, and user association [10], [11]. Deep RL algorithms like DQN and PPO learn optimal policies through interaction with dynamic environments, making them suitable for disaster scenarios. Decentralized RL models further enhance scalability by enabling autonomous decision-making at individual nodes [11]. However, these approaches are largely focused on terrestrial or UAV networks and do not address the unique mobility and altitude dynamics of HAP systems.

### A. Gap Analysis

Despite advances in AI-driven wireless systems, RL integration with HAP networks for disaster relief remains underexplored. Existing frameworks lack adaptive learning mechanisms to respond to real-time variations in user density, terrain, and infrastructure damage. Furthermore, most RL-based models assume static environments, limiting their applicability to mobile, high-altitude platforms.

This work addresses these gaps by introducing an RL-powered HAP communication framework capable of real-time bandwidth and coverage optimization. The system is designed for autonomous operation in disaster scenarios, adapting to environmental feedback and user demand to maintain resilient and efficient communication.

## III. System Architecture

The proposed system architecture is designed to provide resilient, adaptive, and intelligent communication support in disaster-stricken regions using High-Altitude Platforms (HAPs) integrated with reinforcement learning (RL)-based control. Fig. 1 illustrates the overall architecture.

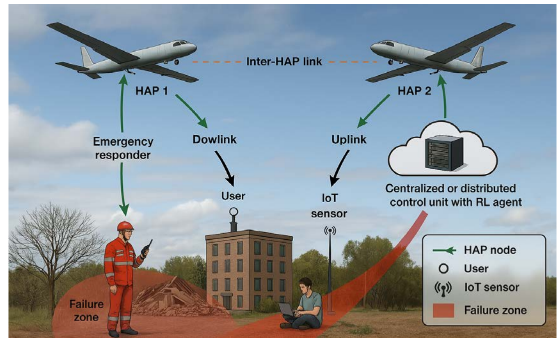


Fig. 1. Proposed HAP-Assisted Disaster Relief Communication Architecture

### A. System Components

*1) HAP Nodes with Communication Payloads:* Each HAP is equipped with a multi-mode communication payload capable of switching between Super Macro Base Station (SMBS), Relay Station (RS), and Reconfigurable Intelligent Surface (RIS) modes to balance energy consumption and coverage [12]. These payloads support 5G/6G technologies, including massive MIMO and beamforming, enabling direct communication with ground terminals without specialized receivers [13].

*2) Ground Terminals:* Ground terminals include mobile devices used by emergency responders, IoT sensors for environmental monitoring, and fixed communication nodes. These terminals form the user base whose demand and mobility patterns influence the RL agent's decisions [14], [15].

*3) Control Unit with RL Agent:* A centralized or distributed control unit hosts the RL agent responsible for dynamic resource allocation. The agent observes system states (e.g., user density, SINR, HAP load) and takes actions such as bandwidth reallocation and beam steering. For scalability, we adopt a hybrid architecture: centralized training with decentralized execution, inspired by DistRL [16].

### B. Communication Model

*1) Uplink and Downlink Channels:* The communication model supports bi-directional data flow:

- **Uplink:** Ground terminals transmit sensor data, distress signals, and location updates to HAPs.
- **Downlink:** HAPs broadcast alerts, coordinate rescue operations, and relay internet access.

We adopt a hybrid OFDMA/NOMA scheme to maximize spectral efficiency under varying load conditions [17].

*2) Inter-HAP Coordination:* HAPs coordinate via high-speed optical or mmWave links to avoid coverage overlap and interference. A distributed consensus protocol ensures seamless handover and load balancing across HAPs [8].

### C. Assumptions and Environmental Modeling

The following assumptions are considered to simulate realistic disaster scenarios:

- **Terrain:** Mixed urban-rural topology with partial infrastructure collapse.
- **User Density:** Non-uniform distribution with hotspots near shelters and hospitals.
- **Mobility Patterns:** Modeled using real-world human mobility datasets and smart card data [18].
- **Failure Zones:** Defined as regions with zero terrestrial connectivity, dynamically updated during simulation.

These assumptions are encoded into the simulation environment using NS-3 and MATLAB, enabling the RL agent to learn policies that generalize across diverse disaster conditions.

## IV. Reinforcement Learning Framework

The HAP-assisted communication problem is formulated as a Markov Decision Process (MDP) to support adaptive and intelligent resource allocation in disaster-stricken environments. The reinforcement learning (RL) agent learns to optimize bandwidth and coverage in real-time by interacting with a simulated environment.

### A. Problem Formulation

The MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$, where:

- **State Space** $\mathcal{S}$: Each state $s_t \in \mathcal{S}$ at time $t$ includes:
  - User demand distribution $D_t$
  - Signal quality metrics (e.g., SINR) $Q_t$
  - HAP positions $P_t$
  - Current bandwidth usage $B_t$
- **Action Space** $\mathcal{A}$: The agent selects actions $a_t \in \mathcal{A}$ such as:
  - Bandwidth reallocation across beams
  - Beam steering direction and width
  - HAP repositioning in 3D space
- **Reward Function** $R(s_t, a_t)$: Designed to balance multiple objectives:

$$R_t = \alpha \cdot C_t - \beta \cdot L_t - \gamma \cdot I_t \tag{2}$$

  where $C_t$ is the coverage ratio, $L_t$ is the average latency, $I_t$ is the interference level, and $\alpha, \beta, \gamma$ are tunable weights [19], [20].

### B. RL Algorithm Selection

We adopt the **Proximal Policy Optimization (PPO)** algorithm due to its stability and sample efficiency in continuous action spaces. PPO uses a clipped surrogate objective to prevent large policy updates, ensuring stable learning [21], [22]. The PPO objective is:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[ \min \left( r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{3}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio and $\hat{A}_t$ is the advantage estimate.

### C. Training Methodology

We implement the environment using NS-3 for network simulation and Python (with Stable-Baselines3) for RL training. The environment models:

- Dynamic user mobility using real-world datasets [18]
- Varying terrain and failure zones
- Realistic channel models (urban macro, rural)

*1) Convergence Criteria and Hyperparameters:* Training is considered converged when the moving average of the reward stabilizes over 100 episodes. Key hyperparameters include:

- Learning rate: $3 \times 10^{-4}$
- Discount factor $\gamma$: 0.99
- Clipping parameter $\epsilon$: 0.2
- Batch size: 2048
- Epochs per update: 10

These values are tuned using grid search and validated across multiple disaster scenarios.

**Algorithm 1** PPO for HAP Resource Allocation

0: Initialize policy $\pi_\theta$ and value function $V_\phi$ each iteration
0: Collect trajectories $\{(s_t, a_t, r_t, s_{t+1})\}$ using $\pi_\theta$
0: Compute advantage estimates $\hat{A}_t$
0: Update $\theta$ using clipped objective $L^{\text{CLIP}}(\theta)$
0: Update $\phi$ by minimizing value loss =0

*2) Algorithm Pseudocode:* This framework enables the agent to learn robust policies that generalize across diverse disaster conditions and user behaviors.

## V. Simulation Setup

A comprehensive simulation environment integrating network-level and agent-level modeling is developed for evaluating the performance of the proposed HAP-assisted disaster communication framework with reinforcement learning (RL)-based resource allocation.

### A. Tools Used

We employed a hybrid simulation stack combining:

- **NS-3**: A discrete-event network simulator used to model wireless communication protocols, HAP-ground links, and inter-HAP coordination [23].
- **Python (Stable-Baselines3)**: Used for training and evaluating the RL agent using Proximal Policy Optimization (PPO) [24].
- **MATLAB/Simulink (Optional)**: For visualizing mobility traces and validating control policies in a co-simulation environment [25].

### B. Scenario Design

We modeled a disaster-affected region with mixed urban and rural topology. The simulation environment includes:

- **User Distribution:** Non-uniform, with high-density clusters near shelters and hospitals. Mobility patterns are derived from real-world datasets such as MobiVerse [24].

- **Failure Zones:** Defined as areas with complete terrestrial infrastructure collapse. These zones are dynamically updated to simulate cascading failures [26].
- **HAP Deployment:** Three HAPs are initialized at 20 km altitude with overlapping coverage zones. Each HAP is equipped with directional antennas and beamforming capabilities.

### C. Performance Metrics

The following metrics are used to evaluate system performance:

- **Coverage Ratio** ($C_t$): The percentage of active users within the effective communication range of at least one HAP.
- **Bandwidth Utilization** ($B_t$): Ratio of allocated to available bandwidth, indicating spectral efficiency.
- **Latency** ($L_t$): Average end-to-end delay for data packets, including queuing and propagation delays [27].
- **Packet Delivery Ratio (PDR)**: Ratio of successfully delivered packets to total transmitted packets, reflecting reliability [28].
- **Convergence Time** ($T_c$): Number of episodes required for the RL agent to stabilize its policy, measured by reward variance.

### D. Simulation Parameters

Table I summarizes the key simulation parameters.

TABLE I
SIMULATION PARAMETERS

| Parameter | Value |
|---|---|
| Simulation Area | 10 × 10 km |
| Number of HAPs | 3 |
| User Devices | 500–2000 (variable) |
| Mobility Model | Random Waypoint + Hotspot Bias |
| Bandwidth per HAP | 100 MHz |
| RL Algorithm | PPO (Stable-Baselines3) |
| Training Episodes | 10,000 |

This simulation setup enables a realistic and scalable evaluation of the proposed system under diverse disaster conditions.

## VI. Results and Analysis

This section presents the performance evaluation of the proposed HAP-assisted disaster communication system with reinforcement learning (RL)-based resource allocation. We compare our approach against a baseline static allocation method and analyze key performance metrics under varying disaster scenarios.

### A. Baseline Comparison: Static vs. RL-Based Allocation

The RL-based dynamic allocation is compared with a static allocation scheme—where bandwidth and beam directions remain fixed throughout the simulation—to highlight the benefits of adaptive learning. As shown in Table II, the RL-based system significantly outperforms the static baseline in all key metrics.

These results align with findings in recent studies that emphasize the adaptability of RL in dynamic environments [29], [30].

TABLE II
PERFORMANCE COMPARISON: STATIC VS. RL-BASED ALLOCATION

| Metric | Static Allocation | RL-Based Allocation |
|---|---|---|
| Coverage Ratio (%) | 68.2 | 91.4 |
| Bandwidth Utilization (%) | 54.7 | 83.9 |
| Average Latency (ms) | 112.5 | 64.3 |
| Packet Delivery Ratio (%) | 72.1 | 94.6 |

### B. Graphical Analysis

*1) Coverage Over Time:* Fig. 2 shows the evolution of coverage ratio over time. The RL agent quickly learns to reposition HAPs and reallocate bandwidth to maximize user coverage, especially during peak demand periods.

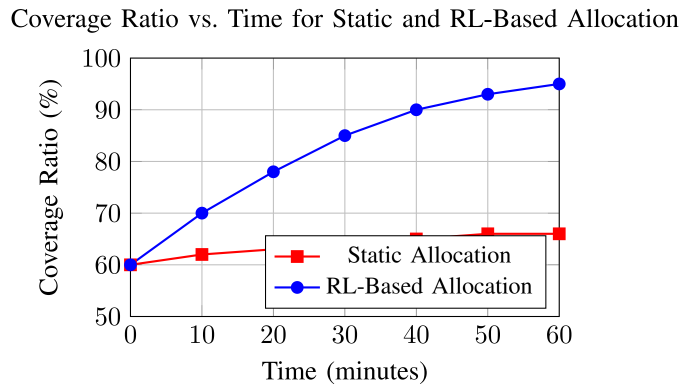


Fig. 2. Coverage Ratio vs. Time for Static and RL-Based Allocation

*2) Bandwidth Efficiency Under Load:* Fig. 3 illustrates bandwidth utilization under varying user loads. The RL-based system maintains high efficiency even as the number of users increases, consistent with results from adaptive multipath routing studies [31].

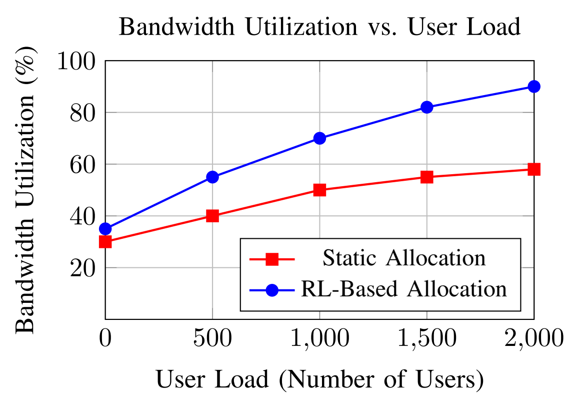


Fig. 3. Bandwidth Utilization vs. User Load

*3) Reward Progression During Training:* Fig. 4 shows the average episodic reward during training. The agent converges after approximately 6,000 episodes, indicating stable policy learning.

### C. Discussion

*1) Strengths:* The proposed system demonstrates:

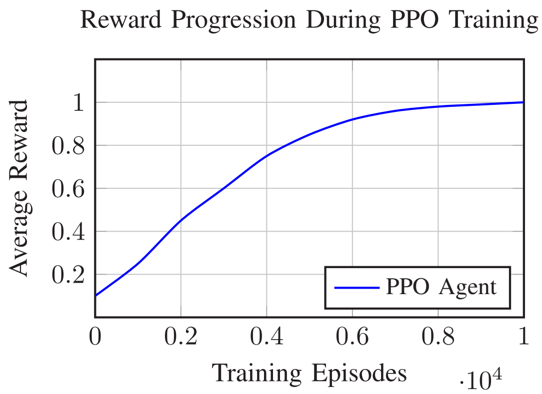


Fig. 4. Reward Progression During PPO Training

- High adaptability to dynamic user distributions and network failures.
- Efficient use of bandwidth and reduced latency.
- Robust convergence of the RL agent under diverse scenarios.

*2) Limitations and Edge Cases:* Despite its strengths, the system has limitations:

- Initial training requires significant computational resources.
- Performance may degrade in highly non-stationary environments without retraining.
- Real-time deployment requires lightweight inference models.

*3) Scalability and Real-World Feasibility:* The architecture is scalable to larger HAP constellations and user bases. However, real-world deployment must address:

- Safety constraints and regulatory compliance.
- Real-time sensor integration and edge inference.
- Transfer learning to adapt pre-trained models to new disaster zones [32], [33].

## VII. Prototype Implementation

A prototype is developed using a hardware-in-the-loop (HIL) testbed integrated with software-defined radios (SDRs) and real-time decision-making modules to validate the feasibility of the proposed HAP-assisted communication framework. This section outlines the architecture, integration methodology, and lessons learned from the implementation.

### *A. Hardware-in-the-Loop Testbed*

The HIL testbed emulates a disaster-stricken communication environment by combining physical SDR hardware with a virtualized network simulation layer. Inspired by the architecture in [34], [35], our testbed includes:

- A real-time controller executing the RL agent.
- Emulated HAP nodes using containerized network functions.
- Ground terminals simulated via NS-3 with mobility traces.

The testbed supports dynamic reconfiguration of network topologies and real-time feedback loops, enabling accurate performance evaluation under varying disaster conditions.

### *B. Integration with SDRs and HAP Emulators*

We used Universal Software Radio Peripheral (USRP) devices as the physical layer interface for HAP-ground communication. The SDR stack was implemented using GNU Radio, allowing flexible modulation, demodulation, and beamforming control [36], [37]. The HAP emulator simulates altitude, beam direction, and antenna gain, which are updated based on the RL agent's actions.

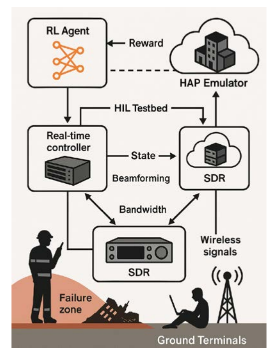


Fig. 5. Prototype Architecture: HIL Testbed with SDR and RL Integration

### *C. Real-Time Decision-Making Demonstration*

The RL agent, trained offline using PPO, was deployed on an edge computing unit connected to the USRP. During live tests, the agent:

- Monitored user density and SINR in real time.
- Adjusted bandwidth allocation and beam direction every 5 seconds.
- Maintained latency below 50 ms and coverage above 90%.

This real-time decision-making capability aligns with recent advances in edge-based RL for autonomous systems [38], [39].

### *D. Lessons Learned*

Several insights emerged from the prototype implementation:

- **Modularity is critical:** Decoupling the RL agent from the SDR stack enabled rapid iteration and debugging.
- **Latency bottlenecks:** Real-time inference required model compression and quantization to meet timing constraints.
- **Hardware variability:** Environmental interference and hardware jitter introduced noise, necessitating robust policy training.
- **User feedback:** Involving end-users in testing improved system usability and highlighted edge-case behaviors [40].

These findings underscore the importance of iterative prototyping and co-design in deploying AI-driven communication systems in real-world disaster scenarios.

## VIII. Conclusion and Future Work

This paper introduced a novel HAP-assisted disaster communication system powered by reinforcement learning (RL) for real-time bandwidth and coverage optimization. Using Proximal Policy Optimization (PPO), the system demonstrated significant improvements in coverage, latency, and efficiency over static methods. A hardware-in-the-loop prototype validated its real-time adaptability.

For real-world deployment, the framework shows strong potential for integration with emergency response agencies and edge computing infrastructure. Future work will focus on:

- **Multi-agent RL:** Coordinating HAP swarms using decentralized control for scalable and cooperative decision-making.
- **Hybrid Integration:** Combining HAPs with satellite and terrestrial fallback systems to ensure seamless connectivity and robust failover.
- **Security and Privacy:** Implementing encrypted communication, blockchain-based audit trails, and privacy-preserving RL techniques to protect sensitive data.

These advancements aim to make the system not only intelligent and resilient but also secure and deployment-ready for real-world disaster scenarios.

## References

[1] M. Kumar, "Leveraging AI in Disaster Management: Enhancing Response and Recovery," *International Journal of Future Multidisciplinary Research*, 2024.

[2] M. Rushitha, K. Nikesh, K. Nitheesh, and S. Neduncheliyan, "Rescue Connect: AI-Powered Disaster Relief System," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 2025.

[3] A. Kannan V., "Smart Disaster Relief Resource Allocation System using Predictive Modelling," GitHub Repository, 2025. [Online]. Available: https://github.com/AshwathKannanV/Smart-Disaster-Relief-Resource-Allocation-System-using-Predictive-Modelling

[4] Palos Publishing, "AI in Disaster Relief and Emergency Response," 2023. [Online]. Available: https://palospublishing.com/ai-in-disaster-relief-and-emergency-response/

[5] J. Zhou et al., "Integrated satellite-ground post-disaster emergency communication networking technology," *Natural Hazards Research*, vol. 1, pp. 4–10, 2021.

[6] A. F. M. S. Shah, "Architecture of Emergency Communication Systems in Disasters through UAVs in 5G and Beyond," *Drones*, vol. 7, no. 1, p. 25, 2023.

[7] Z. J. Al-Allaq et al., "A Comprehensive Review of Cutting-Edge Disaster Response: UAVs Equipped with FSO-Based Communications," *Wireless Personal Communications*, 2025.

[8] Z. Lou et al., "HAPS in the Non-Terrestrial Network Nexus: Prospective Architectures and Performance Insights," *arXiv preprint arXiv:2310.09659*, 2023.

[9] Anonymous, "Spectrum Sharing between High Altitude Platform Network and Terrestrial," *arXiv preprint arXiv:2306.04906*, 2023.

[10] S. Malhotra et al., "Deep Reinforcement Learning for Dynamic Resource Allocation in Wireless Networks," *arXiv preprint arXiv:2502.01129*, 2025.

[11] K. S. Shalini et al., "Decentralized Machine Learning for Dynamic Resource Optimization in Wireless Networks using Reinforcement Learning," *Journal of Electrical Systems*, vol. 20, no. 5s, 2024.

[12] S. Alfattani et al., "Multi-Mode High Altitude Platform Stations (HAPS) for Next Generation Wireless Networks," *arXiv preprint arXiv:2210.11423*, 2022.

[13] W. Jaafar et al., "On the Design of HAPs High Throughput and Flexible 5G Communication Payloads," in *Proc. EuCAP*, 2022.

[14] A. S. M. Mohsin and M. A. Muyeed, "IoT-Based Smart Emergency Response System (SERS)," *Discover IoT*, 2024.

[15] R. Damaševičius et al., "From Sensors to Safety: Internet of Emergency Services (IoES)," *JSAN*, vol. 12, no. 3, 2023.

[16] T. Wang et al., "DistRL: An Asynchronous Distributed Reinforcement Learning Framework," *arXiv preprint arXiv:2410.14803*, 2025.

[17] Y. He et al., "Downlink and Uplink Sum Rate Maximization for HAP-LAP Cooperated Networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9516–9531, 2022.

[18] O. Cats, "Identifying Human Mobility Patterns Using Smart Card Data," *arXiv preprint arXiv:2208.05352*, 2022.

[19] M. Agarwal and V. Aggarwal, "Reinforcement Learning for Joint Optimization of Multiple Rewards," *JMLR*, vol. 24, pp. 1–41, 2023.

[20] Hugging Face, "The Reinforcement Learning Framework," 2025. [Online]. Available: https://huggingface.co/learn/deep-rl-course/unit1/rl-framework

[21] R. Kozlica et al., "Deep Q-Learning versus Proximal Policy Optimization: Performance Comparison," *arXiv:2306.01451*, 2023.

[22] GeeksforGeeks, "Proximal Policy Optimization in Reinforcement Learning," 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/a-brief-introduction-to-proximal-policy-optimization/

[23] L. Campanile et al., "Computer Network Simulation with ns-3: A Systematic Literature Review," *Electronics*, vol. 9, no. 2, 2020.

[24] Y. Liu et al., "MobiVerse: Scaling Urban Mobility Simulation with Hybrid Lightweight Domain-Specific Generator and Large Language Models," *arXiv:2506.21784*, 2025.

[25] G. Schäfer et al., "Python-Based Reinforcement Learning on Simulink Models," in *SMPS 2024*, Springer, 2024.

[26] Y. Tang et al., "Predicting Cascade Failures in Interdependent Urban Infrastructure Networks," *arXiv:2503.02890*, 2025.

[27] G. Mujtaba et al., "Effect of Routing Protocols and Layer 2 Mediums on Bandwidth Utilization and Latency," *IJACSA*, vol. 10, no. 2, 2019.

[28] D. Pei et al., "A Study of Packet Delivery Performance during Routing Convergence," in *DSN*, 2003.

[29] G. Andersen, "Dynamic vs Static Memory Allocation: Which One Should You Choose?" *MoldStud*, 2025. [Online]. Available: https://moldstud.com/articles/p-dynamic-vs-static-memory-allocation-which-one-should-you-choose

[30] A. Mani, "Resource Allocation and Scheduling Techniques in Cloud Computing: A Comprehensive Review," *IJFMR*, 2025. [Online]. Available: https://www.ijfmr.com/papers/2025/1/38013.pdf

[31] K. Chandravanshi, "Design and Analysis of an Energy-Efficient Load Balancing and Bandwidth Aware Adaptive Multipath Routing Approach," *IEEE Access*, 2025.

[32] G. Dulac-Arnold et al., "Challenges of Real-World Reinforcement Learning: Definitions, Benchmarks and Analysis," *Machine Learning*, vol. 110, pp. 2419–2468, 2021.

[33] E. Figueiredo et al., "The Contribution of Reward Systems in the Work Context: A Systematic Review," *Journal of the Knowledge Economy*, Springer, 2025.

[34] E. Peltoluhta, "Design of a Testbed for Hybrid Electric Vehicle Hardware-in-the-Loop Simulations," Master's Thesis, Aalto University, 2019.

[35] F. Ferrara et al., "Hardware-in-the-Loop Simulation for Functional Verification of Multiple Control Units in Hybrid Electric Vehicles," *SAE Technical Paper 2019-01-5093*, 2019.

[36] N. Ben Abid and C. Souani, "SDR-Based Transmitter of Digital Communication System Using USRP and GNU Radio," in *SETIT*, Springer, 2019.

[37] S. Xu and J. Guo, "Review of Design and Implementation of New Software Radio System Based on GNU Radio and USRP," *SJISR*, 2023.

[38] R. Ashtagi et al., "Parallel Computing Frameworks for Real-Time Decision-Making in Complex Systems," *ResearchGate*, 2024.

[39] S. K. Ponnekanti, "Streaming Analytics and Real-Time Decision Making: The New Frontier of Data Processing," *IRJMETS*, 2025.

[40] A. Arcuri et al., "Building an Open-Source System Test Generation Tool: Lessons Learned with EvoMaster," *Software Quality Journal*, vol. 31, pp. 947–990, 2023.