VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B.Tech. Winter Semester 2024-25

School Of Computer Science and Engineering

Network Connectivity: Ensuring reliable network connectivity

Apurva Mishra: 22BCE2791

Date: 29 March, 2025

Contents

1 Introduction **2**

2 Challenges **2**

2.1 Reliability & Availability 2

2.2 Performance 2

2.3 Management and Operation 3

2.4 Resource Constraints 3

2.5 Mobility 3

2.6 Security 3

3 Solution Proposed **3**

3.1 Factors Considered 3

3.2 Performance Evaluation 4

3.3 Proposals 5

4 Benefit Evaluation **7**

4.1 Hybrid Connectivity Method 7

4.2 AI-Powered Predictive Connectivity Maintenance 8

4.3 Cost Evaluation 8

5 Future Works **8**

6 Conclusion **8**

Bibliography **9**

1 Introduction

Modern distributed computing is increasingly dependent on Fog and Edge Computing Paradigm. Fog was a term initially coined by CISCO [1] which was used in enterprise context for placing compute capabilities near the data source, thus extending the cloud. This layer is distributed in nature due to varied nature of data sources. Similarly edge computing refers to processing data at or near the data source utilizing edge devices like sensors, IoT devices, smartphones, etc. However, these approaches seem to be converging, which is evident from the inter-changeable uses of these use of these terms in related literature. [2]

However, due to the distributed nature, reliable connectivity is one of the most important metrics for networks connectivity. Seamless integration between Cloud-Fog-Edge layers depends on the networks protocols and infrastructure in place. [4] Poor connectivity can have various adverse effects, not only affecting the latency and reliability but also compromising the safety and security of the networks, high bandwidth costs, governmental complicance, etc. One of the fundamental strengths of Fog and Edge computing is low latency, real-time processing and local autonomy. All three of these are directly linked to the stability and reliability of the network.

Therefore, this report aims to provide a comprehensive analysis of the challenges concerning network connectivity challenges pertaining to fog and edge computing, and comparing solutions through several relevant metrics.

Applications	**Objectives**	**Devices Involved**
Factories	Detect abnormal events on Assembly line	Humidity, Light and Gas Sensor, Single board computer (SBC)
Home	Fast Motion Detection, Reduce Energy consumption	Infrared Sensors, Raspberry Pi
Vehicles	Fog Computing in Transport Systems	Body Sensor Network
Agriculture	Edge Computing in Agriculture	Infrared Sensors, Raspberry Pi
Transportation	Computing in Transport Systems	Sensor Network
Transportation	Edge Computing in Transportation	Camera-based
Transportation	Edge Computing in Transportation	Raspberry Pi and Camera-based
Transportation	Edge Computing in Transportation	Sensors and Camera-based
Healthcare	Edge Computing in Healthcare	Raspberry Pi and Camera-based
Healthcare	Edge Computing in Healthcare	Wearable Sensors
e-commerce Systems	Computing in e-commerce	Distributed Network

Table 1: Application of Fog and Edge Computing [3]

2 Challenges

As touched on earlier, the differentiating factor for fog and edge computing from the common architecture is its distributed nature. Unlike the monolithic architectures, there are more layers of distributed compute and thus greater variance in quality of service pertaining to factors like connectivity, reliability, security, etc which we shall discuss in more detail in the following sections.

2.1 Reliability & Availability

1. **Intermittent Connectivity**: Fog and edge nodes work in areas which have fluctuating network availability. This is even more prevalent in remote areas or industrial areas due to interference from other devices. Intermittent connectivity can lead to data loss, interference etc. [4]
2. **Unreliable Fog/Edge Nodes**: This intermittent nature of fog and edge nodes can cause rapid re-scheduling of unfinished request. This can prevent fulfilment of requests even on a fully functioning node.[5]

2.2 Performance

1. **Latency**: Various factors can affect the latency for fog and edge nodes. These can be: network congestion, suboptimal routing, incorrect load balancing, etc. This becomes very problematic for real-time software where low-predictable latency is a key requirement.

2. **Bandwidth Limitation**: Aggregation of data from IoT devices can lead to bandwidth bottleneck. This becomes more problematic as we move to data types with more density and bigger sizes like images, video streaming, etc. [6]

2.3 Management and Operation

1. **Network Mangagement**: The heterogenous nature of distributed nodes with varying hardware, software stack poses significant management challenges pertaining to configuring, monitoring and troubleshooting of these devices.
2. **Interoperatiblity**: Similarly the heterogeneous nature of devices pose limitation on effective interopability. This is because often each device optimises for specific task and lack of common standards.

2.4 Resource Constraints

1. **Limited Power**: Many edge devices operate on battery power in remote locations. This makes efficient use of energy essential. Thus energy-efficient protocols, management and compute is an essential part of IoT device development. [7]

2.5 Mobility

Edge devices can often be deployed or used in mobile environments such as smartphones. These constantly hop between networks, making authentication, continuity of service and identification difficult.

2.6 Security

Due to distributed design, it is difficult to ensure sufficient security for each endpoint in the network [8]. Some key challenges in the are include:

1. **Expanded Attack Surface**: Proliferation of edge devices increase the potential entry points for cyber attack. Each device is a potential target for cyber attack.
2. **Insecure Physical Protection**: The cloud server's security model does not apply to edge devices due to distributed nature with high volume and diversity.
4. **Energy Attack**: Here the attacker tries to render the device useless by draining the power source of the device like the battery useless. This can be done by utilizing hardware resources of a device like sensors. [9]
5. **Firmware Modification**: This is a common attack vector. Here the attacker changes the firmware of the device. This is helped by the lack of security checks before the firmware loads and aided by the de-centralised nature of deployment of edge devices. [10]

3 Solution Proposed

Based on analysis and literature review several challenges pertaining to network connectivity in fog and edge computing have been identified. Now we review solutions proposed for these challenges. Among these cost-effectiveness and practical deployment will be major factors of consideration.

3.1 Factors Considered

For comparing different solutions, we make use of the following factors used in paper "Quality of Service-Based Resource Management in Fog Computing: A Systematic Review" [11].

1. **Delay**: refers to the time it takes for data to travel from the source to the destination across a network. Reducing delay is crucial for applications in fog and edge computing, such as autonomous devices. By processing data closer to the source, these paradigms minimize the distance data travels, thus reducing latency and improving responsiveness.
2. **Energy Consumption**: measures the amount of power used by computing resources during data processing and transmission. Energy consumption is an important QoS factor that measures the overall amount of energy required by the local IoT device to execute the incoming user request.
3. **Cost**: encompasses the expenses associated with deploying and maintaining computing infrastructure and network resources. This includes the monetary cost a user pays to the service provider for services such as computation, communication, network, and storage capacity for a given time instance. This cost can vary from time to time depending upon the demand and supply model.
4. **Deadline**: In computing, a deadline is the latest acceptable time by which a task must be completed.
5. **Resource Utilization**: refers to the efficient use of computing and network resources, including CPU, memory, and bandwidth.

Fog nodes have several types of resources, including storage, processing, network bandwidth, and CPU power. Although these resources are limited and need to be utilized efficiently in order to serve as many user requests as possible.

6. **Availability**: measures the degree to which a system is operational and accessible when required.
7. **Scalability**: is the capability of a system to handle a growing amount of work or its potential to accommodate growth.
8. **Security and Privacy**: involves protecting data and resources from unauthorized access, while privacy ensures that personal information is handled appropriately.
9. **Mobility**: refers to the ability of devices to move within a network while maintaining seamless connectivity.
10. **Throughput**: is the rate at which data is successfully transmitted over a network.

3.2 Performance Evaluation

Overview of Papers Reviewed

Paper	Latency Reduction	Throughput / Bandwidth Utilization	Scalability & Reliability	Energy Efficiency	QoS / Adaptability
1. Adaptation21 in Edge Computing (2024) [12]	High – proactive vs. reactive adaptation reduces latency by dynamic routing	Moderate – adapts data paths in real time	High – quickly reconfigures to handle node changes	Evaluated via consumption metrics	Provides adaptability index and reconfiguration speed
2. Sustainable Edge Computing (2024) [13]	Improved response times through local processing	Optimizes bandwidth via load balancing	Scales through sustainable resource allocation	Emphasis on low energy overhead	Focus on maintaining QoS in dynamic scenarios
3. Connecting the Dots (2023) [14]	Significant latency reduction via cloud–fog–edge continuum	Enhanced throughput by offloading non-critical data	High reliability with distributed processing	Not a primary focus, but gains from localized processing	Comprehensive QoS metrics integrated
4. Fog and Edge Security (2021) [15]	Indirectly improves connectivity by mitigating attacks that cause delays	Reduces packet loss by securing network channels	Improves reliability by reducing downtime from attacks	Evaluates impact on energy due to security overhead	Enhances overall network stability (QoS)
5. Decentralized and Trusted Platform (2021) [16]	Lower latency via localized consensus and distributed control	Maintains high throughput through blockchain-enabled load sharing	Excellent scalability and fault tolerance	Energy consumption measured in consensus protocols	Trust metrics and scalability scores are provided
6. Edge and Cloud for IoT Review (2024) [17]	Reviews latency impacts across heterogeneous networks	Examines efficient bandwidth use across diverse protocols	Discusses interoperability and scalability challenges	Highlights trade-offs between local and cloud processing	Offers a comprehensive view of QoS and connectivity

Paper	Latency Reduction	Throughput / Bandwidth Utilization	Scalability & Reliability	Energy Efficiency	QoS / Adaptability
7. Network Connectivity Optimization (2022)	Quantitative reductions in average latency via adaptive routing	Detailed analysis of throughput and packet error rate	Focuses on fast adaptability and network reconfiguration	Evaluates energy impact of dynamic adjustments	Provides robust QoS metrics and adaptability benchmarks
8. QoS Metrics for Connectivity (2023) [11]	Provides end-to-end delay benchmarks	Assesses jitter and throughput under varying loads	Analyzes scalability under heavy traffic	Examines energy efficiency as part of QoS	Offers standardized QoS metrics for connectivity

,

3.3 Proposals

3.3.1 Hybrid Connectivity Method

One of the most promising method is hybrid connectivity. Here fog/edge nodes utilize several communication channels based on real time conditions and requirements. This directly helps with the problem of reliability and availability.

In a normal OSI based network stack, the data is broken into packets and then trasfered over the network. It is not necessary what route the packets take. At both sender and receiver we have mechanism to integrate error checking and packet sequence management. We can leverage a similar system here.

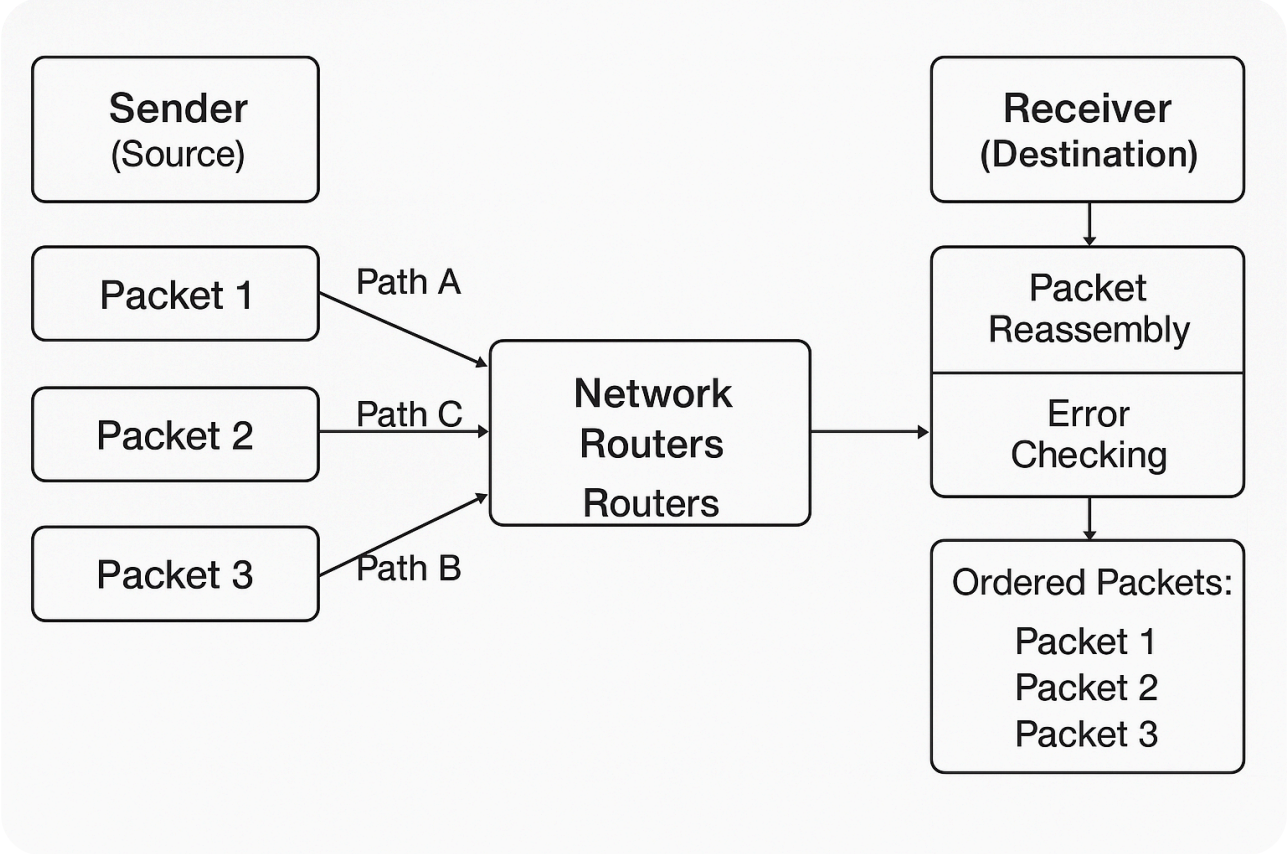

Figure 1: Mechanism of packet travel over the network through different path. These are then recombined at the receiver side using metadata.

Instead of the IP protocol, here we can use language neutral serialization mechanism [18], [19]. Using a neutral mechanism is important to allow for inter-operability between different high level protocols over different communication channels.

For our purposes we will use Protocol Buffers, which is a popular language-neutral, platform-neutral extensible mechanisms for serializing structured data.. Here we separate data in buffers with format:

```
syntax = "proto3";

message DataPacket {
  string message_id = 1;          // Unique ID for the original full message
  int32 sequence_number = 2;      // Position of this packet in the original message
  int32 total_packets = 3;        // Total number of packets the full message was split into
  bytes payload = 4;              // Chunk of serialized data
  string checksum = 5;            // Hash/Checksum of the payload
},
```

Listing 1: Here metadata in buffers like `sequence_number` is used for packet re-assembly.

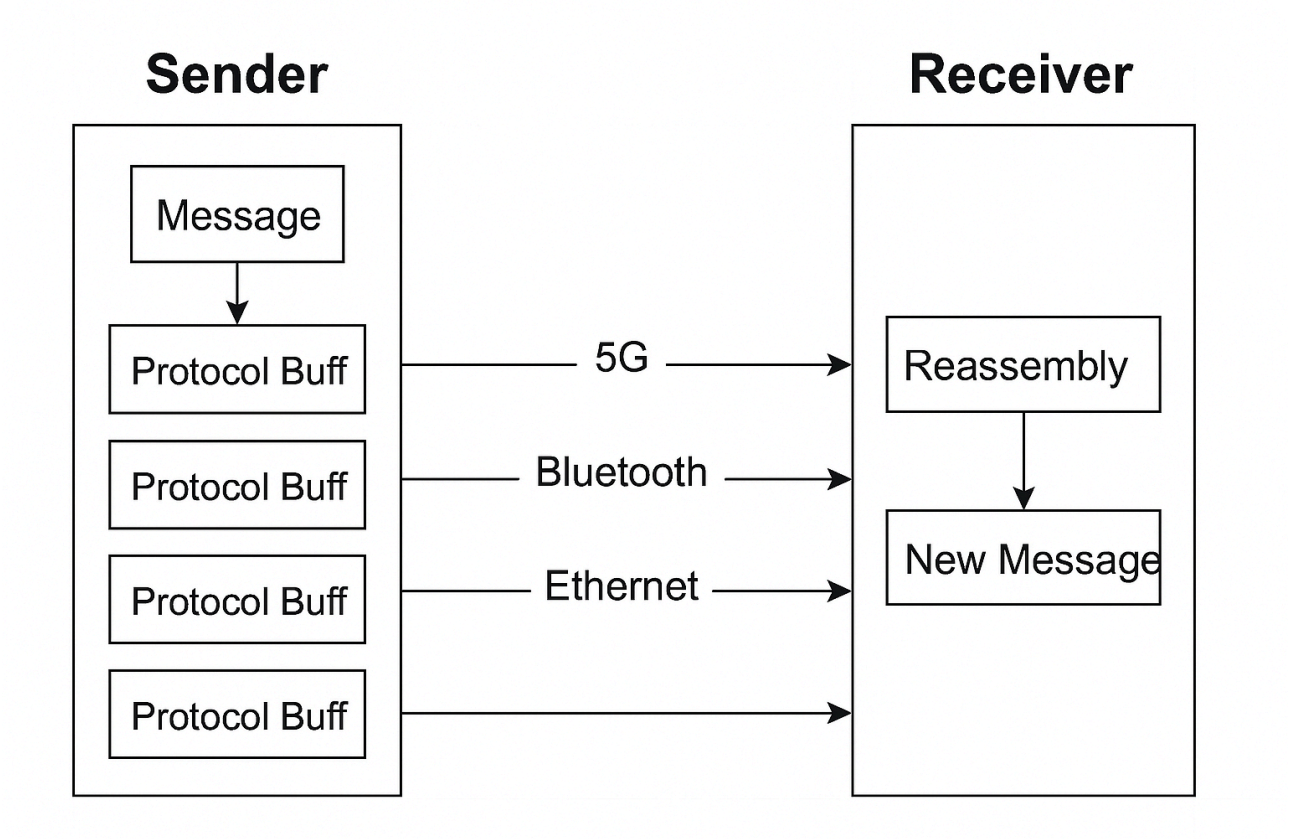

Figure 2: Mechanism of Hybrid Connectivity Method

3.3.2 AI-Powered Predictive Connectivity Maintenance

The above mechanism allows to integrate modern machine learning methods for network connectivity prediction and intelligent routing.

Current Large Language Models which have been instrumental in significant advances in generative modelling are fundamentally heurestic based prediction engines. They are based on Transformer Architecture [20] and work in auto-regressive fashion.

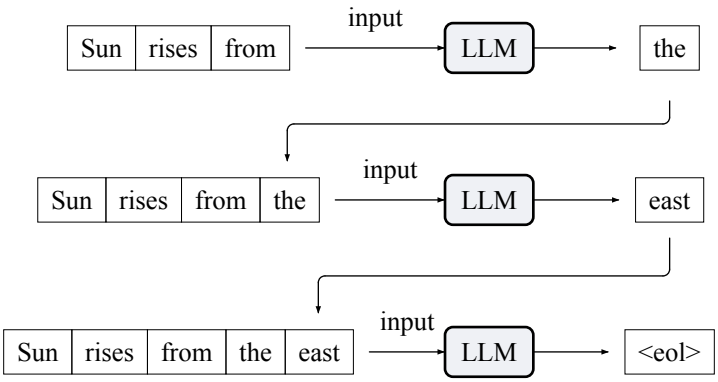

Figure 3: Example for an autogressive LLM policy. The LLM is trained to predict the next most probable word. Then this is appended to the original string and the process continues until the end of line special token is received.

For an input string $\{s_1..s_n\}$ and output string $\{s_{n+1}..s_{n+k+1}\}$, a model works on the policy of:

$$\begin{gathered} p(s_{n+1}..s_{n+k+1}|s_1..s_n) \\ p(s_{n+1}|\ s_1..s_n)p(s_{n+2}|\ s_1..s_n+1)..p(s_{n+k}|\ s_1..s_{n+k-1}) \end{gathered} \tag{1}$$

Thus each successive string is sampled based all the previous set of strings in sequence. This policy can be generalized to simply predict the next event and repeated.

Therefore we can train such a model to take past events in a network connection and then predict the next probably state of network connection i.e. its future networks drops, reliability, latency, etc. This data can then be used by the hybrid router to intelligently route the buffers over appropriate communication channel. [21]

Quadratic complexity. Transformer based attention architecture requires quadratic time complexity over the context length. Specifically QK^T multiplication requires $O(n^2)$ computation and memory. This is the vanilla attention and here the output token can attend to all the input tokens. This can be visualized using the attention mask.[22]

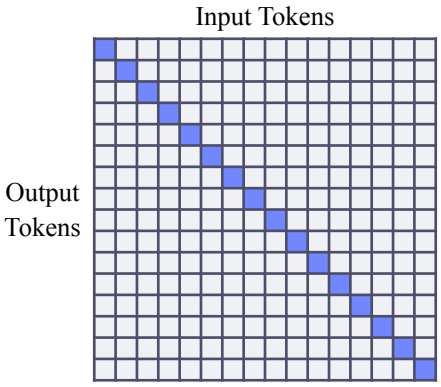

Figure 4: Attention Mask of Vanilla Transformer

Timeseries data of a network can be very long and therefore may be expensive or not possible to utilize a transformer based machine learning model on edge devices which are often resource constrained. Therefore we can optionally use variant of the vanilla attention with linear attention or RNN. Our recommendation is to use **RWKV** [23]. These have linear computation cost and easier to deploy on fog/edge devices.

3.3.3 Integrated Approach

Both- hybrid connectivity method and ai powered predictive connectivity can be used together and designed to work hand in hand. The architecture for hybrid connectivity remains same, however a middleware gets added sender side. This middleware will house the ai model. It will keep track of past network events on various communication channel and dynamically routes the buffers as appropriate on different routing channels.

4 Benefit Evaluation

4.1 Hybrid Connectivity Method

Advantages:

1. **Redundancy**: If one channel fails, others can still deliver them message.

2. **Increased Availability**: Depending on network strength, congestion, data-size, energy-conservation, different paths can be taken. This ensure availability in changing environments.
3. **Platform Interoperability**: Protocol buffers are language neutral thus they would work over heterogenous fog/edge network with varied types of devices..
4. **Scalability**: They architecture is flexible to incorporate new communication mechanism and scale with more nodes.
5. **Energy Consumption**: We can utilize low latency channels for sensitive data and low power channels for rest of the data. Thus allowing for best of both worlds.

4.2 AI-Powered Predictive Connectivity Maintenance

Advantages:

1. **Proactive Routing**: Instead of the hybrid routing being reactionary now the system can be pro-active. This would help with latency spikes, packet losses, etc.
2. **Increase Reliability**: They system can pro-actively change from channels which are degrading before congestion becomes a problem.
3. **Optimized resource utilization**: Forecasting additional metrics for certain predicted use can help better balance traffic and optimize for required metrics. Thus improving on efficiency efficiecyof the system.

Factor	Evaluation
Technical Feasibility	High – Uses lightweight, well-supported components
Operational Complexity	Medium – Multi-interface management and model tuning
Economic Viability	Positive ROI within 1–2 years
Scalability	Modular design supports incremental rollout
Risk	Requires good initial dataset for predictive modeling

4.3 Cost Evaluation

4.3.1 Infrastructure Setup

Current fog and edge nodes would need to be upgraded with capabilities for running machine learning models. . Efficient variants like RWKV are capable of running on very small hardware [24]. However, this if difficult to estimate as this is a per case basis case.

4.3.2 Software Development and Model Training

Integrating AI models into edge devices is a significant challenge, however with maturing ecosystem, developers can use ready to use pre-trained models from online websites like Huggingface.

Additionally these pre-trained models can be cheaply fine-tuned for specific uses cases. [25]

4.3.3 Maintenance

Maintenance and downtime is significant of portaion of lost revenue for networking companies. Similar deployments as in our proposal have shown to provide significant cost savings [26]:

- 5 to 10% cost savings on operations, and maintenance, repairs, and operations (MRO)
- 10 to 20% with increased equipment uptime
- 20 to 50% on reduced maintenance planning time

5 Future Works

In this work we have proposed solutions which improve reliability and availability. We have not provided solution for security challenges of the current system. In addition additional works and analysis needs to be done for deployment AI prediction models on edge/fog devices for network forecasting.

6 Conclusion

Network connectivity is very important for effective communication for fog and edge devices. As discussed in the report, their heteregenous and distributed nature introduces significant challenges to reliability, availability, performance, etc. Solving these challenges is essential for latency sensitive fog and edge paradigm.

In this light we analyze relevant metrics for fog and edge computing. Then building upon then, we proposed a message delivery architecture for fog and edge devices. We utilize hybrid message connectivity method which utilizes several communication channels based on use case and efficiency. Additionally we propose use of a machine learning model for predict future state of network channels. This data can then be used by hybrid model for pro-active routing before network degradation.

Our evaluation shows that this approach would allow for improved network redundancy, availability, scalability and network interoperability. Future work should focus on incorporating robust security measures for this architecture. Ultimately, integrating intelligent and resilient connectivity solutions is paramount for building robust and efficient Fog and Edge ecosystems.

Bibliography

[1] F. Saeik *et al.*, "Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions," *Computer Networks*, vol. 195, p. 108177, 2021, doi: https://doi.org/10.1016/j.comnet.2021.108177.

[2] A. M. Alwakeel, "An Overview of Fog Computing and Edge Computing Security and Privacy Issues," *Sensors (Basel)*, vol. 21, no. 24, p. 8226, Dec. 2021, doi: 10.3390/s21248226.

[3] T. Vo, P. Dave, G. Bajpai, and R. Kashef, "Edge, Fog, and Cloud Computing : An Overview on Challenges and Applications." [Online]. Available: https://arxiv.org/abs/2211.01863

[4] D. Bermbach, S. Lucia, V. Handziski, and A. Wolisz, "Towards Grassroots Peering at the Edge," *CoRR*, 2022, [Online]. Available: https://arxiv.org/abs/2201.03462

[5] S. N. Srirama, "A decade of research in fog computing: relevance, challenges, and future directions," *Software: Practice and Experience*, vol. 54, no. 1, pp. 3–23, 2024.

[6] S. F. Ahmed, S. Shuravi, S. Afrin, S. J. Rafa, M. Hoque, and A. H. Gandomi, "The Power of Internet of Things (IoT): Connecting the Dots with cloud, edge, and fog computing," *arXiv preprint arXiv:2309.03420*, 2023.

[7] M. Fahimullah, S. Ahvar, and M. Trocan, "A review of resource management in fog computing: Machine learning perspective," *arXiv preprint arXiv:2209.03066*, 2022.

[8] X. Jin, C. Katsis, F. Sang, J. Sun, A. Kundu, and R. Kompella, "Edge security: Challenges and issues," *arXiv preprint arXiv:2206.07164*, 2022.

[9] A. Kundu, Z. Lin, and J. Hammond, "Energy attacks on mobile devices," in *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 2020, pp. 107–117.

[10] B.-C. Choi, S.-H. Lee, J.-C. Na, and J.-H. Lee, "Secure firmware validation and update for consumer devices in home networking," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 1, pp. 39–44, 2016.

[11] V. Jain and B. Kumar, "Quality of Service-Based Resource Management in Fog Computing: A Systematic Review," *International Journal of Cloud Applications and Computing*, vol. 12, pp. 1–27, 2022, doi: 10.4018/IJCAC.309934.

[12] F. Golpayegani *et al.*, "Adaptation in Edge Computing: A Review on Design Principles and Research Challenges," *ACM Trans. Auton. Adapt. Syst.*, vol. 19, no. 3, Sep. 2024, doi: 10.1145/3664200.

[13] P. Arroba, R. Buyya, R. Cárdenas, J. L. Risco-Martín, and J. M. Moya, "Sustainable edge computing: Challenges and future directions," *Software: Practice and Experience*, vol. 54, no. 11, pp. 2272–2296, 2024, doi: https://doi.org/10.1002/spe.3340.

[14] S. F. Ahmed, S. Shuravi, S. Afrin, S. J. Rafa, M. Hoque, and A. H. Gandomi, "The Power of Internet of Things (IoT): Connecting the Dots with Cloud, Edge, and Fog Computing." [Online]. Available: https://arxiv.org/abs/2309.03420

[15] A. M. Alwakeel, "An Overview of Fog Computing and Edge Computing Security and Privacy Issues," *Sensors*, vol. 21, no. 24, 2021, doi: 10.3390/s21248226.

[16] L. Cui, S. Yang, Z. Chen, Y. Pan, Z. Ming, and M. Xu, "A Decentralized and Trusted Edge Computing Platform for Internet of Things," *IEEE Internet of Things Journal*, p. 1, 2019, doi: 10.1109/JIOT.2019.2951619.

[17] F. C. Andriulo, M. Fiore, M. Mongiello, E. Traversa, and V. Zizzo, "Edge Computing and Cloud Computing for Internet of Things: A Review," *Informatics*, vol. 11, no. 4, 2024, doi: 10.3390/informatics11040071.

[18] S. Coy, A. Czumaj, C. Scheideler, P. Schneider, and J. Werthmann, “Routing Schemes for Hybrid Communication Networks.” [Online]. Available: https://arxiv.org/abs/2210.05333

[19] C. N. K. Reddy and M. Anusha, “Hybrid Intelligent Routing with Optimized Learning (HIROL) for Adaptive Routing Topology management in FANETs.” [Online]. Available: https://arxiv.org/abs/2406.15105

[20] A. Vaswani *et al.*, “Attention Is All You Need,” *CoRR*, 2017, [Online]. Available: http://arxiv.org/abs/1706.03762

[21] I. Bhattacharjee, “AI-Driven Routing: Transforming Network Efficiency and Resilience<span class="EOP SCXW51976932 BCX0" data-ccp-props=""134233117":false,"134233118":false,"134245418":true,"13 [216],"469777927":[0],"469777928":[2]">,” *Preprints*, Feb. 2025, doi: 10.20944/preprints202502.2005.v1.

[22] Q. Fournier, G. M. Caron, and D. Aloise, “A practical survey on faster and lighter transformers,” *ACM Computing Surveys*, vol. 55, no. 14s, pp. 1–40, 2023.

[23] B. Peng *et al.*, “RWKV-7" Goose" with Expressive Dynamic State Evolution,” *arXiv preprint arXiv:2503.14456*, 2025.

[24] W. Choe, Y. Ji, and F. Lin, “RWKV-edge: Deeply Compressed RWKV for Resource-Constrained Devices,” *arXiv preprint arXiv:2412.10856*, 2024.

[25] XYZ Labs, “Berkeley Researchers Replicate DeepSeek R1's Core Tech for Just $30: A Small Model RL Revolution.” [Online]. Available: https://xyzlabs.substack.com/p/berkeley-researchers-replicate-deepseek

[26] CoreBTS, “Predictive Maintenance with AI: A Comprehensive Guide.” [Online]. Available: https://corebts.com/blog/predictive-maintenance-with-ai/