# Fog computing power for telepresence suit as user terminal for Metaverse

**Vladislav Mogilatov, Valeria Kaijalainen, Artem Volkov, Ammar Muthanna, Andrey Koucheryavy**
The Bonch-Bruevich St Petersburg State University of Telecommunications

**ABSTRACT**

Fog computing represents an emerging paradigm in information and communication technologies that is becoming pivotal for meeting the growing performance requirements of modern computing systems, particularly in the context of the Internet of Things (IoT) and real-time services. One of the most promising 2030-network service classes enabled by fog is telepresence. This paper investigates both the theoretical and practical aspects of deploying fog-computing solutions in telepresence networks to minimize latency and enhance overall system performance. The study combines analytical modelling with discrete-event simulations carried out in Any Logic.

## 1 INTRODUCTION

The accelerating pace with which new technologies are conceived and deployed in the telecommunications domain has created an imperative for continuous adaptation. While fifth-generation (5G) networks are still maturing—and many layers, from standardisation through production, remain in flux—the rapid advancement of enabling technologies has already brought the notion of 2030 communication networks, commonly referred to as 6G, to the research forefront [1], [2], [3], [4]. This transition is not merely an incremental upgrade; it calls for the creation of genuinely new service categories that will define the coming decade of connectivity.

### A. Envisioned Capabilities of 6G Networks

Although detailed specifications and standards for 6G are yet to be ratified, the community broadly converges on several key development vectors:

- Ultra-high data rates: 100 Gbit/s – 1 Tbit/s
- Extreme low latency: ≈ 0.1 ms
- Massive device density: ≈ 100 devices per $m^3$
- Extended spectrum utilization: 1 – 10 THz
- Native integration of artificial intelligence for traffic prediction, resource allocation, and network self-optimization
- Support for novel service verticals such as holographic telepresence, tactile Internet, and immersive extended reality

One paradigm that can satisfy these unprecedented requirements is fog computing. Conceptually a further evolution of cloud computing, fog shifts processing and storage resources closer to the network edge, thereby curbing the latency incurred when data must traverse to centralized clouds and back.

### B. Fog Computing for IoT and Real-Time Services

Fog computing enables data processing on edge devices—computers, mobile handsets, sensors, or smart nodes—rather than exclusively in distant data centers. By handling computations at or near the "last mile," fog mitigates cloud-roundtrip delays and supports instantaneous interactions. It also allows a substantial share of operations to be executed locally, reducing backhaul congestion. Adoption is expected to accelerate owing to three converging factors:

1. Rising computational power of terminal equipment
2. Mature cloud-storage and processing ecosystems that complement, rather than replace, edge resources

Practical solutions to the scalability and latency bottlenecks inherent in largescale IoT deployments



https://doi.org/10.54389/GIFL4259

### C. Reference Architecture Considerations

In 2015, the Open Fog Consortium introduced a reference architecture specification that crystallizes the foundational traits of fog computing. While comprehensive, the Open Fog framework does not explicitly target data-storage locality; it is primarily designed to execute computations near clients to minimize latency, transport costs, and other network-bound constraints while meeting bandwidth requirements. Control elements—including configuration, access management, and network monitoring—are likewise deployed near the edge rather than orchestrated from a central gateway. The architecture further supports local analytics, with results securely synchronized to the cloud for deeper processing or long-term use [5], [6].

Open Fog's architecture emphasizes horizontal scalability and system interoperability, while maintaining high standards of security, elasticity, and flexibility (see Fig. 1.1).

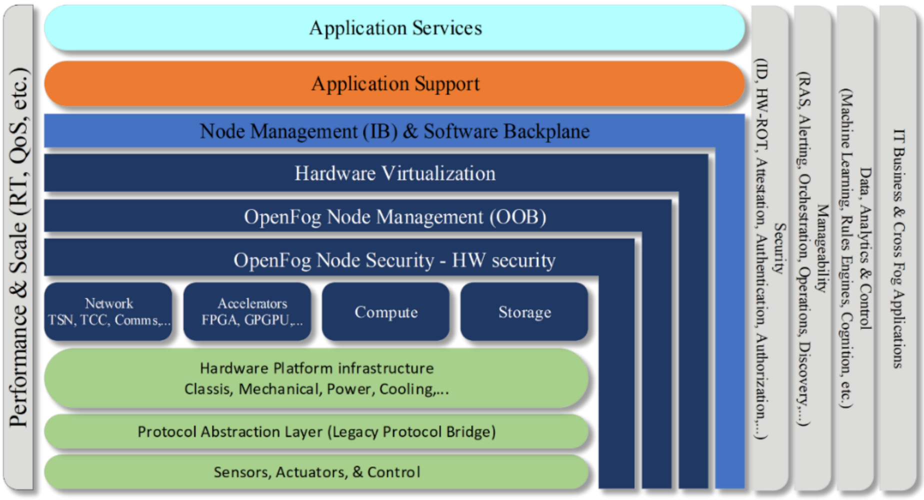


Figure 1.1 OpenFog Architecture

The reference architecture comprises the following key components [7]:

1. Physical Layer – encompasses all physical devices such as sensors, actuators, and edge-computing nodes.
2. Connectivity Layer – provides communication between physical-layer devices and the rest of the network, including access technologies, protocols, and interfaces.
3. Control Layer – manages resources, tasks, and security within the fog-computing environment; it offers mechanisms for performance monitoring, orchestration, and access control.
4. Processing Layer – dedicated to processing and analyzing data received from the physical layer, including real-time analytics, machine learning, and artificial-intelligence workloads.
5. Application Layer – hosts applications that consume data gathered and processed by the lower layers (e.g., monitoring, management, and optimization services).
6. Data-Management Layer – ensures efficient data handling across all layers, covering acquisition, storage, processing, and analysis.
7. Security Layer – a cross-cutting plane that shields data and devices at every tier from unauthorized access and cyber-attacks.

Fog computing embodies a distributed architecture that accelerates data processing, lowers latency, and locates computational resources close to end-users. By deploying compute nodes at the network edge, data can be processed nearer to its source. The Open Fog framework thus provides the foundation for scalable, flexible, and secure systems capable of real-time operation. In this sense, fog computing is not merely an alternative to cloud platforms but a necessary complement for the next wave of ultra-low-latency technologies. A broad body of literature addresses various facets of fog evolution and implementation.

## 2 RELATED WORKS

Table 1.1 Surveyed Contributions in Fog and Cloud Computing

| Author(s) | Year | Contribution |
|---|---|---|
| S. Delfin; N. P. Sivasanker; Nishant Raj; Ashish Anand | 2019 | Fog Computing: A New Era of Cloud Computing [8] – analyses the conceptual foundations of fog computing |
| Asma N. Elmoghrapi; Ahmed Bleblo; Younis A. Younis | 2022 | Fog Computing or Cloud Computing: A Survey [9] – compares advantages and drawbacks of the two paradigms |
| Pooyan Habibi; Mohammad Farhoudi; Sepehr Kazemian; Siavash Khorsandi; Alberto Leon-Garcia | 2020 | Fog Computing: A Comprehensive Architectural Survey [10] – offers an exhaustive overview from an architectural viewpoint |
| Muneer Bani Yassein; Ismail Hmeidi; Farah Shatnawi; Saif Rawasheh | 2020 | Fog Computing: Characteristics, Challenges, and Issues [11] – focuses on fog-computing security concerns |
| Manoj Muniswamaiah; Tilak Agerwala; Charles C. Tappert | 2021 | Modern Computing and the Internet of Things (IoT): A Survey [12] – explores IoT integration with cloud and fog computing |
| Ahmed Salem; Marc St-Hilaire; Muhammad Sajjad Khan | 2023 | Mathematical Models for SLA-Aware Task Allocation in Fog Computing [13] – proposes analytical models for load balancing and energy efficiency |
| Mani Sharifi; Abdolreza Abhari; Sharareh Taghipour | 2021 | Real-Time Application-Processor Scheduling in Fog Computing [14] – presents a MILP-based scheduling model |
| Engr. Urooj Yousuf Khan; Tariq Rahim Soomro | 2023 | A Review of iFogSim: A Simulation Tool for Future Fog Networks [15]– evaluates the iFogSim simulation software |

Delfin et al. [8] trace the origins of fog computing, highlight its principal advantages, catalogue representative use-case families, and discuss pathways for integrating fog with established cloud-centric solutions as the paradigm continues to mature.

Elmoghrapi et al. [9] expand the discussion by juxtaposing cloud and edge-local computing models. Their comprehensive survey assesses how each technology affects day-to-day digital interactions, clarifying that cloud and fog serve complementary—rather than interchangeable—roles in advanced network ecosystems. Habibi et al. [10] provide an exhaustive architectural panorama, comparing contemporary reference frameworks with application-specific designs. The review dissects algorithmic and technological facets, then maps the distinguishing features of cloud, edge, mobile-edge, and fog paradigms.

Security emerges as a recurrent theme. Bani Yassein et al. [11] analyze fog from a layered perspective, underscoring privacy and protection challenges that must be addressed before large-scale adoption is feasible. Because IoT workloads generate prodigious sensor data that demand near-instant analytics, Muniswamaiah et al. [12] investigate how tightly coupling fog resources with IoT platforms can unlock performance gains across heterogeneous devices.

Salem et al. [13] contribute formal task-allocation models for service-level-agreement (SLA) compliance in fog clusters. One formulation target load-balancing efficiency, whereas another emphasizes eco-sustainability by minimizing energy consumption. Sharifi et al. [14] advance this line of inquiry with a mixed-integer linear-programming (MILP) scheduler that delivers optimal dispatching decisions; they benchmark its efficacy against a FIFO baseline under two representative scenarios. Finally, Yousuf Khan and Soomro [15] evaluate iFogSim, a widely used discrete-event simulator that offers modular templates and class libraries for modelling next-generation fog topologies and application flows.

## 3 MODEL NETWORK

The integration of fog computing into the design of telepresence traffic-handling devices provides an effective approach to mitigating several critical challenges, including limited onboard computational capacity, rapid battery depletion, data acquisition constraints and scalability issues, as well as insufficient transmission speeds. By leveraging distributed processing at the network edge, fog-enabled telepresence systems can achieve higher efficiency, more compact form factors, and enhanced user experience due to reduced latency and optimized resource utilization.

Building on a rigorous analysis of traffic offloading mathematical models, we have developed a discrete-event simulation model that emulates data transmission from a telepresence suit to metaverse infrastructures. This simulation enables comprehensive performance evaluation, including delay metrics and load distribution across fog nodes, providing a framework for fine-tuning real-time applications.

The figures included in this study illustrate the prototype architecture and operational workflow of the proposed model, highlighting how fog-assisted offloading can effectively reduce end-to-end delays while improving throughput and energy efficiency in telepresence environments.

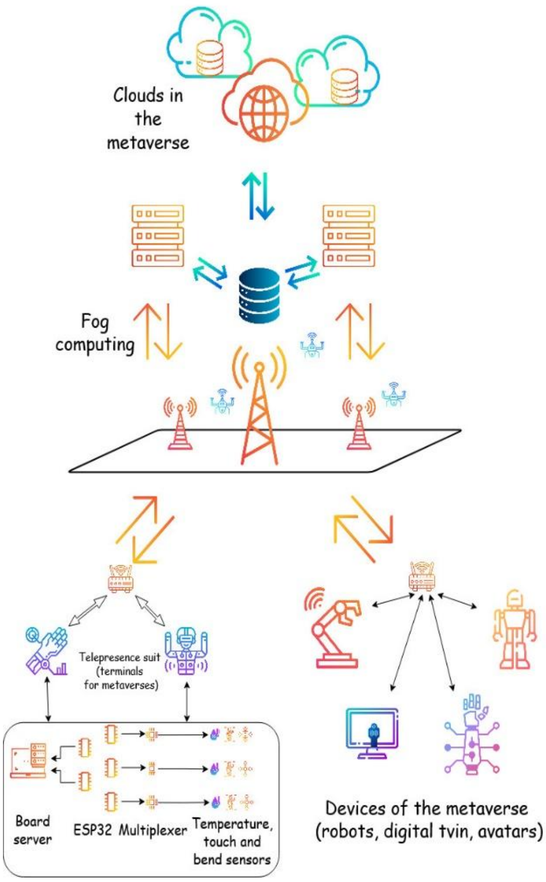


Figure 1.2 Model Architecture

### Mathematical Model

We consider a mobile-computation offloading scheme that comprises a single fog-computing node and a remote cloud-computing facility. The telepresence suit equipped with $M$ processing boards is treated as a multi-server queuing system capable of diverting traffic to both fog and cloud resources. *Figure 1.2* illustrates an exemplary network in which

- $M$ low-power devices represent the on-board processors,
- node $M + 1$ denotes the fog-computing element, and
- node $M + 2$ corresponds to the cloud-computing element.

During operation, the suit's boards consume substantial computing cycles and electrical power. According to the proposed model, a portion of the data-processing workload is offloaded to the fog node. Should the fog segment become saturated, overflow tasks are forwarded to the remote cloud to prevent congestion. Our design goal is to reduce the number of boards, minimize energy consumption, and utilize network resources efficiently while maintaining strict end-to-end latency constraints.

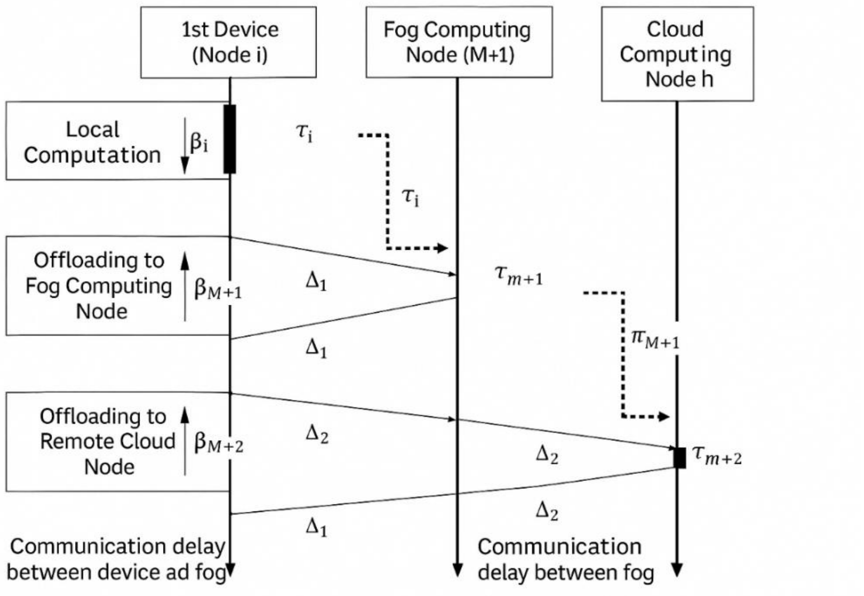


Figure 1.3 Network-Based Computation-Offloading Model

Let $\beta_i$ denote the random variable representing the response time of a task originating from board $i$.

- Upper branch — Local execution. The task is processed entirely on board $i$; hence the board sustains the greatest computational load, but the response time is minimal and equals the local processing duration:

$$\beta_i = t_i.$$

- Middle branch — Fog-level execution. The task is offloaded via a wireless link to the fog node. Although transmission incurs delay, this option conserves the board's processing cycles and energy. The response time is the sum of the round-trip transmission delay $2\Delta_1$ and the fog-processing time $t_{M+1}$:

$$\beta_i = 2\Delta_1 + t_{M+1}.$$

- Lower branch — Cloud-level execution. When the fog node is unable to handle the task, it forwards the job to the remote cloud, thereby relieving fog congestion at the expense of additional latency. The response time now includes the round-trip delay to the cloud $2\Delta_2$:

$$\beta_i = 2\Delta_1 + 2\Delta_2 + t_{M+2}.$$

The expressions in (1) capture the latency trade-offs among local, fog, and cloud processing paths, forming the basis for the performance analysis presented in [16].

This computation-offloading scheme can be formalized in terms of queueing networks. Let $M$ denote the number of processing boards—equivalently, the local-computation nodes—within the coverage area of the fog node. A board that acquires and forwards sensor data is indexed as node $i$, where $1 \le i \le M$. Each such node generates a Poisson stream of requests with rate $\lambda_i$ [requests $s^{-1}$]. Every task requires a specific computational workload, measured in operations or floating-point operations. Let $w_i$, $1 \le i \le M$, be the non-negative random variable representing the workload of a task originating on board $i$, with cumulative distribution function $W_i(x) = \Pr\{ w_i \leq x \}$. Denote by $\mu_i$ the constant service rate of board $i$ in instructions per second. A task is offloaded to the fog node whenever $w_i > w^*$, where $w^*$ is the workload threshold; otherwise, it is executed locally on the mobile device. The offloading probability for board $i$ is therefore $\pi_i = 1 - W_i(w^*)$. With the foregoing notation, the complete mathematical model of the system is depicted schematically in *Figure 1.3* [14].

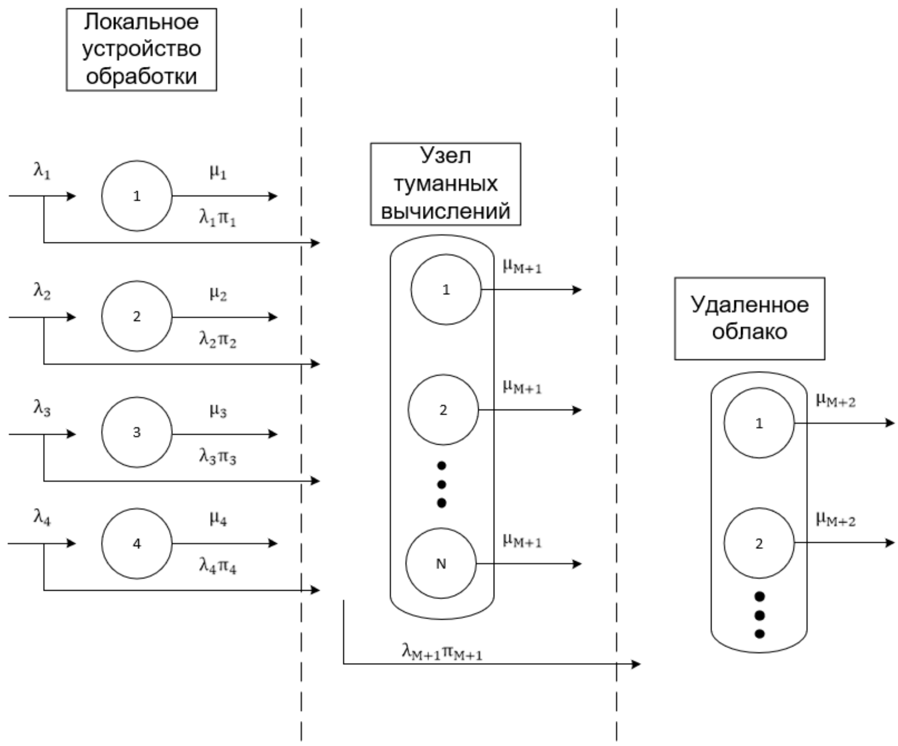


Figure 1.4 Queueing-System Model for Telepresence-Suit Traffic Offloading

Table 1.2 Probability Distribution of the Response-Time Random Variable $\beta_i$

| $\beta_i$ value | $t_i$ | $2\Delta_1 + t_{M+1}$ | $2\Delta_1 + 2\Delta_2 + t_{M+2}$ |
|---|---|---|---|
| Probability $P$ | $1 - \pi_i$ | $\pi_i(1 - \pi_{M+1})$ | $\pi_i\,\pi_{M+1}$ |

**Simulation Model**

Building on the mathematical framework outlined above, we developed a discrete-event simulation in AnyLogic and conducted a series of experiments to validate the analytical results. The simulator implements both fog-assisted and non-fog data-transmission scenarios, allowing direct comparison of end-to-end performance.

The accompanying figure depicts the experimental test-bed layout, while the corresponding table lists the parameter values used in the simulations. This model enables network planners to:

- Forecast system behavior when telepresence traffic is integrated with fog computing,
- Dimension fog clusters appropriately, and
- Estimate the user-side network resources required to sustain real-time service delivery.

These capabilities are instrumental for designing scalable, low-latency telepresence architectures in next-generation networks [17] [18].

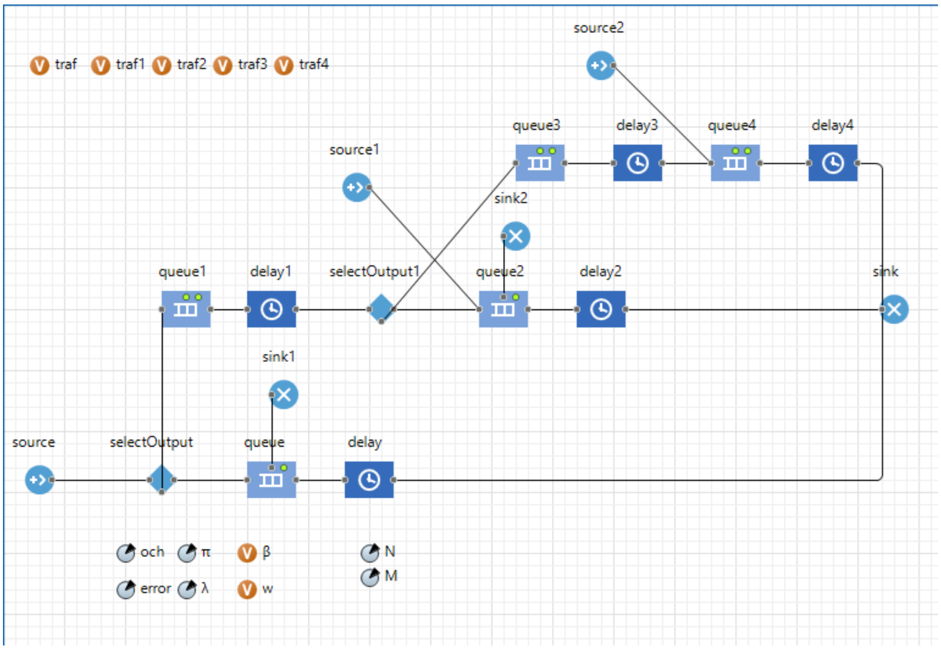


Figure 1.5 Discrete-Event Simulation Model of the Queueing System

Table 1.3 Model Parameters Used in the Experiments

| **Parameter (Name)** | **Symbol** |
|---|---|
| Flow arrival rate per board | $\lambda_i$ |
| Offloading probability | $\pi$ |
| Offloaded-workload threshold | $w^*$ |
| Local processing rate | $\mu_i$ |
| Fog-node processing rate | $\mu_{M+1}$ |
| Cloud processing rate | $\mu_{M+2}$ |
| One-way fog-transmission delay | $\Delta_1$ |
| Additional cloud-transmission delay | $\Delta_2$ |
| Mean response time | $\beta_i$ |
| Overload events (error) | *error* |
| Queue length | *och* |
| Number of processing boards | $M$ |
| Devices in the fog cluster | $N$ |

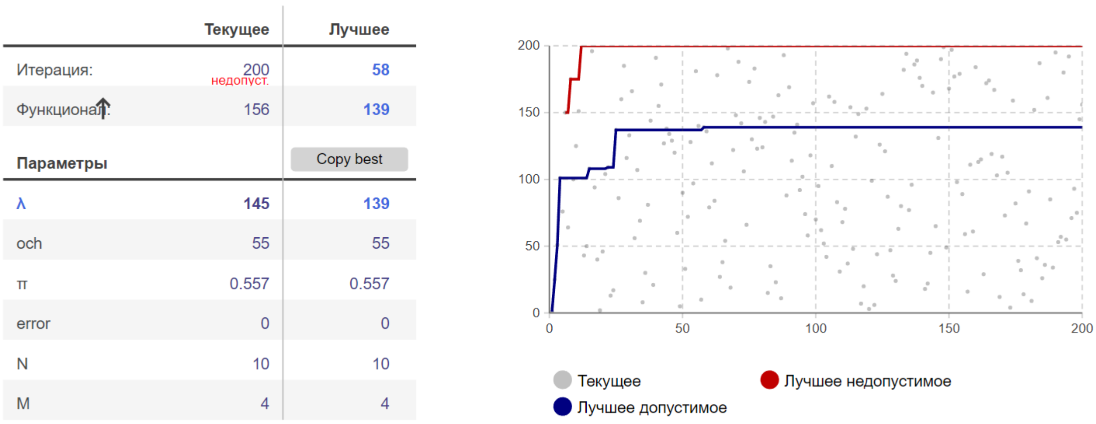


Figure 1.6 Optimisation-Model Outcome

In the first experiment, we determined the optimal values of the offloading probability and the queue length that minimize the mean system sojourn time. Because timely delivery of every packet is critical for telepresence traffic, the configuration must guarantee zero overload events.
The optimization yielded an offloading probability of $\pi = 0.557$, which is fully consistent with the analytical model, along with the corresponding minimal queue length. This configuration balances the volumes offloaded to fog, processed locally, and forwarded to the cloud, ensuring uniform workload distribution across the boards.

The second and third experiments employed parameter sets derived from the actual telepresence-suit hardware specification and refined through the simulation framework. We established,

- the maximum data volume that the boards can collect and process without fog support, and
- the performance gains—in terms of throughput and latency—achieved through fog-computing integration.

Table 1.4 Experimental Results

| **Metric** | **Baseline** | **Prototype** | **Fog Deployment** |
|---|---|---|---|
| Number of processing boards | 20 | 4 | 4 |
| Offloading probability | $\pi = 0.559$ | $\pi = 0$ | $\pi = 0.557$ |
| Mean response time $\beta$ [s] | 0.012 | 0.021 | 0.007 |
| Aggregate arrival rate $\lambda$ [tasks $s^{-1}$] | 80 | 48 | 139 |
| Normalized $\beta$ | 1.00 | 1.75 | 0.58 |
| Normalized $\lambda$ | 1.00 | 0.60 | 1.74 |

The introduction of fog computing into telepresence service architecture addresses critical challenges including limited on-board computational power, rapid battery depletion, data-acquisition bottlenecks, scalability constraints, and insufficient transmission throughput. As demonstrated, the fog-enabled configuration triples the data-collection and transfer capacity without adding extra hardware to the suit. Consequently, system efficiency is markedly improved while complexity and cost remain unchanged, enabling a lean yet high-performance telepresence solution suitable for next-generation 6G networks.

## 4 CONCLUSION

The application of analytical modelling and discrete-event simulation in AnyLogic has produced several quantitative outcomes: (i) a pronounced reduction in system response time, significantly exceeding the data-processing rates of current solutions; (ii) lower energy consumption owing to task execution at fog nodes; and (iii) a decrease in the amount of required on-board hardware. By integrating fog computing, the traffic intensity can be maximized without a three-fold increase in the number of processing boards, thereby demonstrating substantial economic efficiency.

The simulation model has experimentally validated the theoretical insights obtained from the mathematical analysis. This validation not only confirms the accuracy of the theoretical calculations but also identifies optimal operating parameters for the system under a variety of conditions, providing a reliable basis for further research on fog-cluster sizing.

The experiments clearly indicate the significant advantages of employing fog computing in telepresence devices, including a three-fold improvement in system performance and a corresponding reduction in response time. These findings confirm that the integration of fog technologies is a promising avenue for the advancement of telepresence systems, delivering higher efficiency and greater user convenience for end users.

## REFENRENCE:

[1] C.-X. Wang, J. Huang, H. Wang, X. Gao, X. You, and Y. Hao, “6G Wireless Channel Measurements and Models: Trends and Challenges,” *IEEE Veh. Technol. Mag.*, vol. 15, no. 4, pp. 22–32, Dec. 2020, doi: 10.1109/MVT.2020.3018436.

[2] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland, and F. Tufvesson, "6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities," *Proc. IEEE*, vol. 109, no. 7, pp. 1166–1199, July 2021, doi: 10.1109/JPROC.2021.3061701.
[3] A. Shi, F. Sun, and M. Li, "An Exploration of Holographic Scenarios in 6G Networks," in *2022 10th International Conference on Information Systems and Computing Technology (ISCTech)*, Dec. 2022, pp. 450–456. doi: 10.1109/ISCTech58360.2022.00076.
[4] A. Volkov, A. Muthanna, and A. Koucheryavy, "Fifth generation communication networks: on the way to networks 2030," *Telecom IT*, vol. 8, no. 2, June 2020, doi: 10.31854/2307-1303-2020-8-2-32-43.
[5] abid, "Introducing the Industrial Internet Consortium, Now Incorporating OpenFog!," Industry IoT Consortium. Accessed: July 21, 2025. [Online]. Available: https://www.iiconsortium.org/2019/02/introducing-the-industrial-internet-consortium-now-incorporating-openfog/
[6] "3. OpenFog Consortium Architecture Working Group, OpenFog architecture overview, OpenFog Consortium, - Google Search." Accessed: July 21, 2025. [Online]. Available: https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf
[7] Thang, D. V., Volkov, A., Muthanna, A., Koucheryavy, A., Ateya, A. A., & Jayakody, D. N. K. (2025). Future of Telepresence Services in the Evolving Fog Computing Environment: A Survey on Research and Use Cases. Sensors, 25(11), 3488.
[8] S. Delfin, N. P. Sivasanker., N. Raj, and A. Anand, "Fog Computing: A New Era of Cloud Computing," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Mar. 2019, pp. 1106–1111. doi: 10.1109/ICCMC.2019.8819633.
[9] A. N. Elmoghrapi, A. Bleblo, and Y. A. Younis, "Fog Computing or Cloud Computing: a Study," in *2022 International Conference on Engineering & MIS (ICEMIS)*, July 2022, pp. 1–6. doi: 10.1109/ICEMIS56295.2022.9914131.
[10] P. Habibi, M. Farhoudi, S. Kazemian, S. Khorsandi, and A. Leon-Garcia, "Fog Computing: A Comprehensive Architectural Survey," *IEEE Access*, vol. 8, pp. 69105–69133, 2020, doi: 10.1109/ACCESS.2020.2983253.
[11] M. B. Yassein, I. Hmeidi, F. Shatnawi, and S. Rawasheh, "Fog Computing: Characteristics, Challenges and Issues," in *2020 International Conference on Mathematics and Computers in Science and Engineering (MACISE)*, Jan. 2020, pp. 240–245. doi: 10.1109/MACISE49704.2020.00051.
[12] M. Muniswamaiah, T. Agerwala, and C. C. Tappert, "Fog Computing and the Internet of Things (IoT): A Review," in *2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, June 2021, pp. 10–12. doi: 10.1109/CSCloud-EdgeCom52276.2021.00012.
[13] A. Salem, M. St-Hilaire, and M. S. Khan, "Mathematical Models for Task Assignment with Service Level Agreement in Fog Computing," in *2022 International Conference on Smart Applications, Communications and Networking (SmartNets)*, Nov. 2022, pp. 1–8. doi: 10.1109/SmartNets55823.2022.9993983.
[14] M. Sharifi, A. Abhari, and S. Taghipour, "Modeling Real-Time Application Processor Scheduling for Fog Computing," in *2021 Annual Modeling and Simulation Conference (ANNSIM)*, July 2021, pp. 1–12. doi: 10.23919/ANNSIM52504.2021.9552113.
[15] Engr. U. Yousuf Khan and T. Rahim Soomro, "Overview of iFogSim: A Tool for Simulating Fog Networks of the Future," in *2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, Nov. 2022, pp. 1–4. doi: 10.1109/MACS56771.2022.10023395.
[16] Zhao, L. H., Zhang, J., Bah, M. J., Li, Z., Ying, J. J. C., Muthanna, A., & Elgendy, I. A. (2025). Energy-Efficient Task Offloading in Multi-server Mobile Edge Computing Networks: A Deep Reinforcement Learning Approach. Arabian Journal for Science and Engineering, 1-22.
[17] Vorobyova D. et al. IoT network model with multimodal node distribution and data-collecting mechanism using mobile clustering nodes //Electronics. – 2023. – T. 12. – №. 6. – C. 1410..
[18] van Thang, D., Volkov, A., Muthanna, A., Elgendy, I. A., Alkanhel, R., Jayakody, D. N. K., & Koucheryavy, A. (2025). A Framework Integrating Federated Learning and Fog Computing Based on Client Sampling and Dynamic Thresholding Techniques. IEEE Access, 13, 95019-95033.