



Paper Type: Original Article

Optimization of Network Latency in Cloud-Based IoT Systems

Srinithi Mitra* 

School of Computer Science Engineering, KIIT University, Bhubaneswar, India; 22053291@kiit.ac.in.

Citation:

Received: 09 February 2024

Revised: 28 April 2024

Accepted: 17 July 2024

Mitra, S. (2024). Optimization of network latency in cloud-based IoT systems. *Metaversalize*, 1(3), 121-128.


Abstract


The growing adoption of cloud-based Internet of Things (IoT) systems has introduced significant challenges in managing network latency, adversely impacting real-time applications like healthcare, smart cities, and industrial automation. As IoT devices are often geographically dispersed, transmitting data over long distances to centralized cloud servers leads to delays, reducing system responsiveness and reliability. To address this issue, this study explores several latency optimization methods, including the implementation of edge and fog computing, advanced routing protocols, and data compression techniques. Edge and fog computing architectures bring data processing closer to the IoT devices, reducing the distance data must travel and minimizing latency. In addition, advanced routing protocols and packet scheduling are utilized to optimize data transmission. In contrast, data compression techniques are applied to reduce transmitted data volume, further improving transmission speeds. The results demonstrate a substantial decrease in network latency, with edge and fog computing reducing latency by up to 40% compared to traditional cloud-only systems. These improvements are further enhanced by applying optimized routing and compression methods, which streamline data flow and increase transmission efficiency. The implications for the field are significant, as these solutions improve the performance and scalability of existing cloud-based IoT systems and enable the development of future latency-sensitive applications such as autonomous vehicles and real-time analytics, making cloud-based IoT deployments more viable for critical real-time tasks.

Keywords: Network latency, Cloud-based IoT, Edge computing, Network optimization.

1 | Introduction

Network latency, the time it takes for data to travel between Internet of Things (IoT) devices and cloud-based servers, is a critical factor in the performance and responsiveness of IoT applications. Minimizing latency ensures real-time data processing, timely decision-making, and system efficiency. Several strategies can be employed to optimize network latency in cloud-based IoT systems [1]. Edge computing is one of the most effective approaches to reduce latency [2]. By moving data processing closer to the IoT devices, edge

 Corresponding Author: 22053291@kiit.ac.in

 <https://doi.org/10.22105/metaverse.v1i3.67>



Licensee System Analytics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

computing eliminates the need for data to travel long distances to the cloud. This reduces network congestion and minimizes transmission delays.

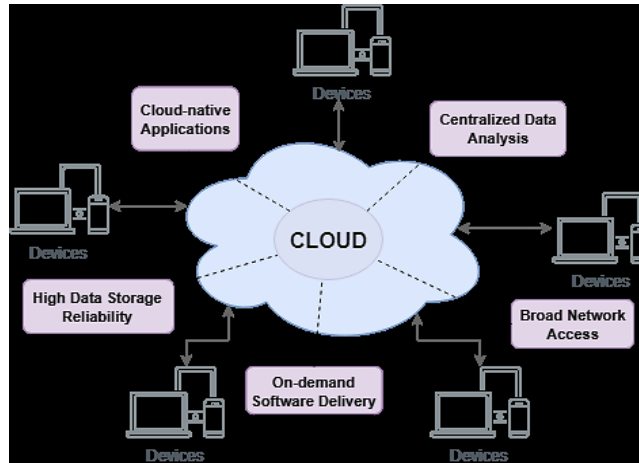


Fig. 1. Cloud computing architecture, key advantages, and functionalities.

Additionally, edge computing can enable local decision-making, allowing faster responses to real-time events without relying on cloud-based processing [3]. Optimizing network infrastructure is another crucial aspect of minimizing latency. Upgrading routers, switches, and other network components to support high-performance data transmission can significantly improve network throughput and reduce latency. Implementing Quality of Service (QoS) mechanisms can also prioritize IoT traffic, ensuring that critical data packets are delivered with minimal delay. Choosing the right communication protocols can also impact network latency. Protocols for low-latency communication, such as MQTT or COAP, are well-suited for IoT applications [4]. These protocols use efficient message formats and minimize overhead, resulting in faster data transmission. Cloud providers offer various network options, including regional and global data centers. Selecting the appropriate data center location can significantly reduce latency for IoT devices in specific geographic regions.

Additionally, leveraging Content Delivery Networks (CDN) can cache data closer to IoT devices, reducing the need for data fetched from distant servers. Finally, optimizing application design and data management practices can also reduce latency. Organizations can reduce network traffic and improve latency by minimizing the amount of data transmitted between IoT devices and the cloud and using efficient data compression techniques.

The motivation behind our work stems from the pressing need to reduce network latency in cloud-based IoT systems and help with its early detection and prevention through various networking models. The emergent IoT is thought to be the next generation of the internet, in which billions of things are interconnected. Examples are sensors, actuators, mobile phones, and cars communicating with each other to perform service objectives. Cloud computing is a new paradigm that enables users to elastically utilize a shared pool of cloud resources (e.g., processors, storage, applications, services) in an on-demand fashion.

2 | Literature Review

The emergent IoT is thought to be the next generation of the internet, in which billions of things are interconnected. Examples of these things are sensors, actuators, mobile phones, and cars communicating with each other to perform service objectives. Cloud computing is a new paradigm that enables users to elastically utilize a shared pool of cloud resources (e.g., processors, storage, applications, services) in an on-demand fashion. Recently, driven by the potential of complementing the ubiquitous data-gathering abilities of IoT devices with the powerful data storage and data processing capabilities of the cloud, the integration of the cloud and the IoT is attracting rising attention from academia and industry [5].

The data (alarm, security, climate, and entertainment) gathered by sensors are transmitted first to the gateway. The associate editor coordinating the review of this manuscript and approving it for publication was Adnan M. Abu-Mahfouz. Then, the received sensory data is transmitted to the cloud. Eventually, the cloud stores, analyzes, processes, and transmits the sensed data to the users on demand. During the entire data transmission process, if the data transmission from the sensor nodes to the cloud is unsuccessful, data are re-transmitted until successfully delivered. For this cloud-based IoT prototype, the IoT acts as the cloud's data source, while users are the cloud's data requesters. The users can access the needed sensory data from the cloud whenever and wherever there is a network connection. In these potential applications of cloud-based IoT integration, such as smart buildings of smart cities [6], a number of them require the IoT will reliably offer sensory data to the cloud based on users' requests. In general, sensors' limited battery power will be depleted by performing data sensing, processing, and transmission after a specific period, as they are usually supplied with non-rechargeable batteries, and their replacement may also be impractical [7]. Several approaches have been evolved VOLUME 7, 2019 2169-3536 2019 IEEE. Translations and content mining are permitted for academic research only. Personal use is also allowed, but republication/redistribution requires IEEE permission. However, approaches to reduce power consumption contrarily impact the network's reliability. An example of this approach is applied when part of the network works whilst other parts sleep. This approach is excellent for power consumption but not reliability because part of the network may be inaccessible due to an IoT node sleeping. Another example of this approach is multiple paths between a specific IoT node and the gateway (Fig. 2).

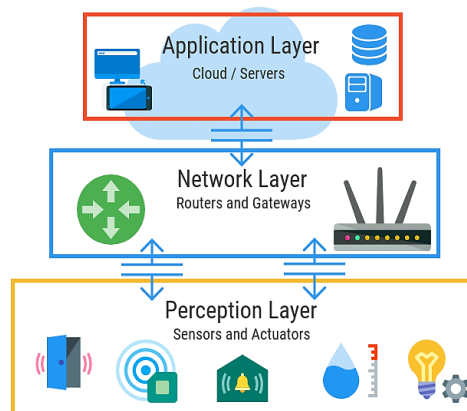


Fig. 2. Overview of cloud-based IoT.

In contrast to the previous example, this method is excellent for reliability but not for power consumption because it will use more than one route, which means more IoT nodes to transmit the same packet. Therefore, assessing the reliability of IoT by considering traffic power consumption is important. In this work, we have considered that IoT networks can fail in two points: links due to traffic congestion or interference and sensor nodes due to diminishing their energy. This paper proposes four models for achieving two goals; the energy efficiency of cloud-based IoT considering the reliability level. For instance, we used the following approaches to reduce total traffic power consumption. First, each model aims to minimize the total transmitted power by selecting the IoT device with the lowest energy per bit and idle power. Second, using the data compression technique reduces the amount of data transmitted. Thirdly, interference cancellation will reduce the re-transmission of the data lost due to interference. To achieve the reliability objective, we proposed two approaches; the first is to select reliable links with a 99% reliability level. The second is to choose a standby link as an alternative to the link failing. Finally, each proposed scheme has two optimization objectives: energy efficiency and reliability. The main contributions of this paper are summarized as follows:

- I. Virtualize cloud-based IoT network using the MILP model.
- II. Minimize the total traffic power of the cloud-based IoT network through the MILP optimization model.
Minimize interference.

- III. Achieving reliability in the cloud-based IoT network.
- IV. Distributing traffic through the gateways to avoid traffic congestion to the cloud in the IoT network. Handle more traffic demands by using the data compression technique.
- V. Jointly investigate the issues regarding energy efficiency and reliability from the viewpoint of cloud-based IoT integration.

This paper proposed four models related to the evaluation of cloud-based IoT networks. It considers the mote energy level as the main factor of failures of WSN nodes. It uses the routing algorithm to define the paths between different WSN regions and the sink node, and it automatically generates reliability models considering the elements above. This paper further proposes four schemes consisting of a Standby Routes Selection Scheme (SBRS), a Desired Reliability Level Scheme (DRLS), a Reliability-Based Sub-Channel (RBS) scheme, and a Reliability-Based Data Compression (RBDS) scheme aimed at improving the reliability of IoT networks and reducing total transmitted power. Specifically, an SBRS selectively chooses standby routes to overcome node failure problems and reduce transmission power. In addition, a DRLS is used when a specific reliability level is needed to guarantee the link's reliability while minimizing transmission power. Furthermore, due to its high reliability, an RBS uses sub-channels to mitigate interference and reduce overhead on links that are utilized by several IoT devices. Finally, an RBDS uses a sequential lossless entropy compression (S-LEC) data compression algorithm to overcome the capacity limits of the links and reduce transmission power. In the rest of this paper, Section II introduces the related work, Section III describes the background to this research, and Section IV presents the cloud-based IoT integration system model. Section V introduces the network optimization model of cloud-based IoT. Section VI presents our model objectives by introducing SBRS, DRLS, RBS, and RBDS, and Section VII evaluates the model results. In Section VIII, we state our conclusions regarding the research.

3 | Edge Cloud System with IoT

Edge computing and cloud computing have become essential in enhancing the performance of IoT systems. By integrating the computational capabilities of the cloud and the low-latency benefits of edge devices, edge-cloud systems offer a robust solution for IoT deployments that require real-time data processing and low energy consumption [8].

3.1 | Edge Computing and Cloud Integration in IoT

Edge computing refers to processing data at the network's periphery (closer to the data source) rather than in a centralized cloud. The main advantage is reduced latency, as data does not need to travel to a distant server for processing. On the other hand, cloud computing offers unlimited processing power and storage but introduces higher latency and potential network congestion.

Latency (L): One of the key factors determining the performance of an IoT system, defined as the time taken for a data packet to travel from the source (IoT device) to its destination (cloud/edge server).

$$L = t_{\text{edge}} + t_{\text{cloud}} + t_{\text{network}} = t_{\{\text{edge}\}} + t_{\{\text{cloud}\}},$$

where

T_{edge} is the processing time at the edge node, t_{cloud} is the processing time at the cloud server, and the network is the network transmission time. Bandwidth (BBB) is the available data transfer rate between IoT devices and edge/cloud servers, which influences the amount of data that can be processed within a specific time.

3.2 | IoT Device Constraints and Resource Management

IoT devices often have limited computational power, memory, and energy resources. Efficient resource management algorithms must be designed to offload tasks from IoT devices to edge or cloud servers.

Energy consumption (E): The energy consumed by an IoT device can be modeled as

$$E = P \cdot t,$$

where

P is the power consumption rate (in watts), and t is the time duration of the task (in seconds). Edge computing helps reduce the energy consumption of IoT devices by performing data-intensive computations closer to the data source, reducing the time IoT devices are active for communication.

Task offloading (Ot): A critical decision variable in edge-cloud systems determines whether a task should be executed on the IoT device, offloaded to the edge, or sent to the cloud.

$$O_t = \arg i \in \{\text{IoT}, \text{edge}, \text{cloud}\} \min (L_i + E_i).$$

L_i and E_i are the latency and energy consumption for the task at node i .

3.3 | Data Transmission and Network Models

Data transmission in IoT systems typically involves multiple communication channels, such as Wi-Fi, 5G, and Low Power Wide Area Networks (LPWAN). The communication model impacts data flow from IoT devices to edge/cloud servers.

Transmission delay (Dt): The time data moves across the network. This can be modeled as

$$D_t = S/B,$$

where

S is the size of the data being transmitted (in bits), and B is the available bandwidth (in bits per second).

Data queuing delay (Dq): The delay introduced by queuing in edge/cloud servers affects the total system latency.

$$D_q = \lambda / (\mu - \lambda),$$

where

λ is the arrival rate of data packets, μ is the service rate at the edge/cloud server.

3.4 | Optimization Techniques in Edge-Cloud System

Various optimization techniques have been proposed to enhance the performance of edge-cloud systems, such as minimizing latency, energy consumption, and task failure rates.

Latency optimization: To minimize total latency, processing tasks should be placed at nodes that minimize the sum of edge, cloud, and network latencies.

$$\min (\sum_{i=1}^N L_i) \min \left(\sum_{i=1}^N L_i \right) \min (i = 1 \dots N L_i).$$

Subject to constraints such as available bandwidth, computational power at the edge, and task priorities.

Energy-aware offloading: An energy-aware optimization model can aim to minimize the energy consumption across IoT devices, edge, and cloud servers:

$$\min (\sum_{i=1}^N E_i) \min \left(\sum_{i=1}^N E_i \right) \min (i = 1 \dots N E_i),$$

where E_i is the energy consumed by each node i (IoT, edge, or cloud).

Load balancing: Ensuring the computational load is evenly distributed among edge and cloud nodes to prevent overload and bottlenecks. A load balancing function can be expressed as

$$LB = \sigma/\mu,$$

where σ is the standard deviation of the workload across nodes, μ and is the mean workload.

3.5 | Security and Privacy Consideration

Security is paramount in IoT systems due to the sensitive nature of data [9]. Combining edge computing and cloud computing poses secure data transmission and storage challenges.

Encryption overhead (He): The processing time and computational resources required to encrypt data before transmission.

$$He = f(K, S),$$

where

K is the key size used for encryption, S is the size of the data.

This overhead must be balanced against the system's real-time performance requirements.

3.6 | Case Studies and Practical Implementation

Smart cities: Edge-cloud systems manage real-time traffic data in smart city applications. Variables such as Traffic Density (TD) and Average Speed (AS) are monitored using edge nodes to adjust traffic lights and reduce congestion. An optimization model might look like

$$O_{\text{traffic}} = \min(TD, \text{Ledge}).$$

Healthcare: IoT devices monitor patients in real time, and edge computing ensures that critical data, such as heart rate or glucose levels, is processed with low latency.

4 | Challenges Faced by Cloud-IoT Systems

4.1 | Security

Data from IoT was placed in the cloud for processing and retrieval. This involves encryption of data sent to or saved in cloud-based repositories and data security during cloud access and use. The degree to which there is a lack of cloud computing information is such that data owners do not understand their own data's physical position. Today, data is related to everything around us, so data security in the cloud IoT paradigm is the main topic.

5.2 | Storage and Computational Performance

Plans that include cloud-based IoT devices require high-performance goal requirements. Such specifications can be difficult to meet in all settings because cloud-based IoT devices are in motion for many applications.

5.3 | Reliability

The IoT devices depend on the cloud to work providers for time-critical apps, and the effect would directly reflect the program's output. For example, in cars, surgical instruments, or the security field.

5.4 | Big Data Storage

Nearly 50 billion IoT devices will be offered by about 2025, and that revenue will be a big obstacle for cloud service providers to rapidly have fast and safe access to data.

5.5 | Maintenance

Depending on what is learned in the above segment, extremely efficient techniques and plans are needed to track and manage protection and efficiency in the cloud environment to fulfill the requirements of as many as 50 billion IoT devices.

6 | Results and Discussion

After reviewing works improving IoT systems' performance, we found that most solutions propose improving one part of the system at a time: the network, the processing, or data quality. Per contra, reducing the sent data enhances the network's and the server's responsiveness and saves the IoT device's energy and resources. However, propositions in data transmission reduction are limited since most rely on classifiers, compression, or prediction methods. Those methods take a long time to train and add heavy processing on the resource-limited object, and on top of that, they are not one hundred percent error-free. Also, data transmission solutions focus on one IoT system, generally WSN containing temperature, humidity, and luminosity sensors, since they generate numeric data that can be predicted easily. Moreover, for all these solutions, no generalization is proposed. Therefore, our work is characterized by proposing a generalized (that can be projected on different IoT systems) and formalized (as we define mathematically all the functioning of the model) solution to reduce data offloaded from a connected device to the cloud. This improves the system's performance in several aspects.

7 | Conclusion

The optimization of network latency in cloud-based IoT systems is crucial to ensuring real-time data processing, improved user experience, and enhanced system performance. Techniques like edge computing, efficient data routing, protocol optimization, and load balancing can significantly reduce latency. Edge computing minimizes the distance between data generation and processing by offloading tasks to local servers, reducing round-trip times to the cloud. Adaptive data compression and caching further lower transmission delays, while advanced protocols like QUIC optimize data flow. Additionally, leveraging AI and ML for predictive analytics can preemptively address network bottlenecks, resulting in a more responsive and scalable IoT ecosystem. Ultimately, these strategies collectively enhance cloud-based IoT networks' overall efficiency, reliability, and speed.

Funding

This research received no external funding.

Data Availability

The data used and analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflict of interest regarding the publication of the paper.

References

- [1] Shukla, S., Hassan, M. F., Tran, D. C., Akbar, R., Paputungan, I. V., & Khan, M. K. (2023). Improving latency in internet-of-things and cloud computing for real-time data transmission: A systematic literature review (SLR). *Cluster computing*, 26(5), 2657–2680. <https://doi.org/10.1007/s10586-021-03279-3>
- [2] Abouaomar, A., Cherkaoui, S., Mlika, Z., & Kobbane, A. (2021). Resource provisioning in edge computing for latency-sensitive applications. *IEEE internet of things journal*, 8(14), 11088–11099. <https://doi.org/10.1109/JIOT.2021.3052082>

- [3] Bablu, T. A., & Rashid, M. T. (2025). Edge computing and its impact on real-time data processing for IoT-driven applications. *Journal of advanced computing systems*, 5(1), 26–43. <https://doi.org/10.69987/>
- [4] Lakshminarayana, S., Praseed, A., & Thilagam, P. S. (2024). Securing the IoT application layer from an MQTT protocol perspective: Challenges and research prospects. *IEEE communications surveys and tutorials*, 26(4), 2510–2546. <https://doi.org/10.1109/COMST.2024.3372630>
- [5] Lenka, R. K., Kolhar, M., Mohapatra, H., Al-Turjman, F., & Altrjman, C. (2022). Cluster-based routing protocol with static hub (CRPSH) for WSN-assisted IoT networks. *Sustainability*, 14(12), 7304. <https://doi.org/10.3390/su14127304>
- [6] Selvaraj, R., Kuthadi, V. M., & Baskar, S. (2023). Smart building energy management and monitoring system based on artificial intelligence in smart city. *Sustainable energy technologies and assessments*, 56, 103090. <https://doi.org/10.1016/j.seta.2023.103090>
- [7] Bathre, M., & Das, P. K. (2023). Smart dual battery management system for expanding lifespan of wireless sensor node. *International journal of communication systems*, 36(3), e5389. <https://doi.org/10.1002/dac.5389>
- [8] Pourqasem, J. (2024). Transforming user experience in the metaverse through edge technology. *Metaversalize*, 1(1), 21–31. <https://doi.org/10.22105/metaverse.v1i1.19>
- [9] Swain, B., Raj, P., Singh, K., Singh, Y., Singh, S., & Mohapatra, H. (2025). Ethical implications and mitigation strategies for public safety and security in smart cities for securing tomorrow. In *Convergence of cybersecurity and cloud computing* (pp. 419–436). IGI Global Scientific Publishing. <https://doi.org/10.4018/979-8-3693-6859-6.ch019>