

Article

Load Balanced Data Transmission Strategy Based on Cloud–Edge–End Collaboration in the Internet of Things

Jirui Li ^{1,†}, Xiaoyong Li ^{2,*,†}, Jie Yuan ² and Guozhi Li ¹

¹ School of Information Technology, Henan University of Chinese Medicine, Zhengzhou 450046, China

² Key Laboratory of Trustworthy Distributed Computing and Service, Beijing University of Posts and Telecommunications, Beijing 100876, China

* Correspondence: lixiaoyong@bupt.edu.cn

† These authors contributed equally to this work.

Abstract: To improve the response speed and quality of Internet of Things (IoT) services and reduce system operating costs, this paper refines the edge layer according to the different data transmission capabilities of different edge devices, constructs a four-layer heterogeneous IoT framework under cloud–edge–end (CEE) collaboration, and gives the corresponding data hierarchical transmission strategy, so as to effectively process sensitive data such as real-time, near-real-time, and non-real-time data. Meanwhile, the link based high-performance adaptive load balancing scheme is developed to achieve the dynamic optimal allocation of system resources. Simulation results demonstrate that the data hierarchical transmission strategy based on a CEE collaboration framework can not only make full use of resources and improve the successful delivery rate of packets but can also greatly reduce the end-to-end transmission delay of data. Especially, compared with the cloud-mist framework without refining the edge layer, the data transmission rate based on CEE collaboration architecture is increased by about 27.3%, 12.7%, and 8%, respectively, in three network environments of light-weight, medium, and heavy load.

Keywords: internet of things; edge computing; cloud–edge–end collaboration; data layered transmission; load balancing; resource allocation



Citation: Li, J.; Li, X.; Yuan, J.; Li, G. Load Balanced Data Transmission Strategy Based on Cloud–Edge–End Collaboration in the Internet of Things. *Sustainability* **2022**, *14*, 9602. <https://doi.org/10.3390/su14159602>

Academic Editor: Zubair Baig

Received: 19 June 2022

Accepted: 1 August 2022

Published: 4 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The IoT mainly relies on perception layer technology (radio frequency identification, short-range wireless communication, sensors, etc.), network layer technology (wired or wireless access), and business and application layer technology (information discovery, intelligent processing, middleware, distributed computing, etc.) to connect all things in the world into information systems according to the agreement, so as to reduce the distance between information systems and the physical world [1,2]. Countries or regions such as the United States, the European Union, Japan, and South Korea have launched national IoT strategies, and China officially wrote the IoT into the “government work report” as early as 2009. With the vigorous promotion of key technologies such as perception technology, transmission technology, information processing technology, and common technology, the layout of the ecosystem of the IoT industry will continue to be promoted, and data circulation will become an important factor to accelerate the further development of the IoT market. Therefore, with the enthusiastic pursuit of a high quality of service (QoS) in the IoT and the further promotion and integration of IoT platform construction, the research on data transmission strategies as an organic part of the IoT is bound to become a hot spot [3,4].

However, the complex characteristics of the IoT environment, such as its dynamics, coordination, diversity, mobility, and openness, seriously affect the efficiency of data transmission and bring a series of challenges to the continuity of the IoT service supply. Among them, the primary factor affecting data transmission is the selection and construction of

IoT system architecture. A correct and effective IoT architecture can ensure that data can be transmitted to the right location at the right time. As for the current IoT service based on cloud computing [5–14], because the cloud server is far away from the terminals and the rate of data transmission over ultra-long distance is low, it will not only occupy a lot of network bandwidth but will also increase the transmission delay [15], so that the IoT system cannot respond to emergencies in real-time. As a result, the efficiency of service decision making is reduced, the waiting time of users is long, and the quality of the user experience is rapidly declined.

As the core technology of 5G, edge computing [16] can delegate data processing and other functions from the network center to edge nodes and supplement cloud technology by providing a large amount of storage, communication control, configuration, measurement, and management on edge devices [17–21]. Compared with cloud computing, it provides more advantages for traditional IoT applications in terms of service delay, energy consumption, network congestion, and user privacy [22]. For example, it can provide emergency responses, longer-lasting services, and higher QoS for sensitive requests such as emergency public opinion analysis or hypertension and diabetes diagnosis that need real-time feedback information. The literature [22–28] shows that applying edge computing can not only effectively avoid overloading the cloud center and reduce network congestion and service delay but can also provide short-range, decentralized, highly reliable, and fast computing services for IoT applications with real-time and non-real-time data processing requirements, forming a complementary mode with long-distance cloud computing that provides centralized big data processing functionality. The research on data transmission strategies in the IoT under CEE collaboration has more practical value.

Some of the literature [29–34] made use of the advantages of edge computing to carry out corresponding scheme design or performance improvement for data transmission in different IoT applications. Although these studies have achieved some achievements in delay, energy consumption, or load balancing, their practicability is not high because of the singleness of the problems they solve. For example, Mubarakali et al. [29] and Li et al. [30] only considered data transmission delay or delivery rate and ignored the problem of load balancing. Wang et al. [33] mainly focused on the efficiency of system resource allocation. Although Xia et al. [34] emphasized load balancing and energy consumption of the system, it ignored the importance of transmission delay. In addition, the devices involved in edge computing usually include mobile intelligent devices and basic network devices. The communication technologies and capabilities between different types of devices are different, and their effects on transmission are naturally different. However, almost all previous studies ignored this difference, resulting in over idealized experimental results. Moreover, although an IoT service scheme based on cloud or edge computing can alleviate or improve the problems such as data transmission delay and system power consumption to a certain extent, there are still some QoS problems in the transmission of sensitive data. For example, the delay or loss of sensitive data does not need to be processed at each layer of the IoT service architecture. Thus, this requires an IoT framework to have “on demand” layered data transmission and processing capabilities according to the needs of different types of data.

Therefore, to further improve the performance of IoT service, this paper intends to adopt an edge computing mode to improve the limitations of the “cloud-management-end” closed-loop integrated architecture in the original IoT service system, and builds an open CEE IoT service architecture that is coordinated by a central cloud, an edge cloud, and physical terminals via using 5G, edge computing, intelligent routers, and other equipment and technical means. Then, aiming at the IoT service data transmission belt between different types of devices such as terminal sensors, intelligent edge devices, and edge infrastructure network facilities in this architecture, how to design a systematic and reliable data transmission strategy to optimize resource allocation, balance system load, improve packet delivery rate, and reduce service delay is the purpose of this paper. The goal is

to achieve efficient, practical, intelligent, green, secure, and trusted IoT services [35]. The main work is as follows:

- Taking advantage of edge computing, this paper proposes a novel IoT architecture under CEE collaboration. This heterogeneous framework includes terminal sensing devices, edge devices, cloud center devices, and other related application entities. Its novelty is that the edge cloud is divided into the edge intelligent device layer and the edge infrastructure layer according to the different data transmission and processing capabilities of edge devices to meet different needs of different types of data and applications in the IoT ecosystem for task processing response time.
- Based on the cooperative CEE IoT architecture, a highly reliable layered data transmission strategy is designed. This strategy adopts the principle of dynamic resource allocation to set corresponding transmission paths for different types of IoT service data, such as real-time, near-real-time, and offline/batch processing modes; dynamically optimize system resources on demand; balance network load; reduce data transmission delay; and improve data delivery rate.
- Simulation is used for comparative analysis to verify the rationality and effectiveness of the layered data transmission strategy of the cooperative CEE IoT architecture proposed in this paper. The experimental results show that the division of the edge intelligent device layer and the edge infrastructure layer makes the data transmission service of IoT have the best performance in terms of the task load, data transmission delay, and data delivery rate. In particular, compared with the cloud–fog framework [36], which does not refine the edge layer, data transmission rate under CEE collaborative IoT architecture is increased by about 27.3%, 12.7%, and 8% on average under the three network environments of light-weight, medium, and heavy loads, respectively.

The rest of this paper is as follows: Section 2 introduces the related work. Section 3 describes in detail the IoT architecture under CEE cooperation proposed in this paper, and its layered data transmission strategy, resource dynamic allocation method, and system load balancing scheme. In Section 4, the experimental scenario, evaluation index, and parameter setting of IoT data transmission under CEE collaboration are constructed, and the experimental results are deeply analyzed. Section 5 briefly explains the existing problems and future work. Section 6 summarizes the whole work.

2. Related Work

With the rapid popularization of smart devices in recent years, there are more and more studies trying to further improve the performance of the IoT from various aspects such as the efficiency of data transmission, the load of the cloud center, and the system architecture, so that it can reduce the adverse effects of smart devices and give full play to their advantages.

Singh et al. [11] designed an intelligent power quality monitoring system integrating the IoT and the cloud. The system records the mechanical quantity of the driver and the power quality event data in the Firebase cloud, which limits the space demand for power quality event analysis and saves time. Mohiuddin et al. [12] analyzed the situation of time-varying workloads and data-intensive IoT applications when using cloud computing services and studied and investigated the challenges and strategies brought by cloud computing, so as to promote the transition of IoT applications to cloud security. Alkadi et al. [13] proposed a security-based distributed intrusion detection and privacy-based smart contract blockchain framework for the IoT. As a decision support system, this framework can help users and cloud providers migrate data in a timely and reliable manner. Arulanthu et al. [14] used the cloud–IoT model to design an online medical decision support system for chronic kidney disease prediction. For cloud computing mode based the IoT, Chinnasamy et al. [37] leveraged a hashing algorithm and a signature verification model to realize the secure access of data stored in cloud centers. Obviously, the combination of the IoT and cloud computing can effectively promote the development of innovative solutions in many different fields, including industry, military affairs, medical treatment, and so on.

However, the processing mode that only relies on the cloud center is limited by latency and power consumption. To overcome these limitations, Cisco introduced the concept of fog, namely, edge computing, in 2012. Edge/Fog computing can expand computing power and data analysis applications from the cloud center to the “edge” of the network, create a carrier level service environment with high performance, low latency, and high bandwidth; accelerate the distribution and download of various contents, services, and applications in the network; and let consumers enjoy a higher quality network experience [21].

Kaur et al. [23] explained the different characteristics of the cloud computing platform and the fog computing platform, introduced the detailed architecture of the two platforms, and made a comparative analysis based on the IoT application scenario. Zahmatkesh et al. [26] analyzed and studied the edge computing applications and potential enabling technologies for sustainable intelligent cities under the IoT environment. At the same time, they also comprehensively discussed the applications of different caching technologies, UAV (unmanned aerial vehicles) technologies, and various artificial intelligence and machine learning technologies in data caching in the IoT system based on edge computing. Mancini et al. [38] designed an open source Android library to act as a gateway between smart devices and edge/cloud computing devices to simplify the development of Android applications. Chinnasamy et al. [39] described and evaluated in detail the protection and confidentiality standards, intervention measures, and the latest technical methods used to avoid network risks in edge computing, classified and analyzed the common attack and defense technologies on edge devices, and points out its research direction on privacy and security. El-hasnony et al. [27] proposed a hybrid real-time remote monitoring framework based on cloud–fog integration to realize the continuous remote monitoring of patients. Sood et al. [28] proposed an IoT fog–cloud system for monitoring, evaluating, and controlling dengue virus. Dastjerdi et al. [40] used the edge computing mode to process massive data in the IoT and enhance its QoS.

Based on the definition of individual components of edge computing, Sarkar et al. [41] proposed a mathematical model of edge computing in the IoT environment, which is superior to traditional cloud computing in terms of energy consumption and service delay. Abbasi et al. [22] tried to use the concept of green energy to improve the energy model of network edge devices, so as to reduce the delay and power consumption of the multi-sensor framework in the secure IoT system. An et al. [42] designed a hyper-connected IoT ecosystem using contextual AI technology on the fog platform, aiming to provide smarter, faster, and more reliable IoT services. Loffi et al. [43] improved a multi factor mutual authentication method based on variable response time and challenge response function, which can be used in cloud and edge computing, so that the IoT services have better average processing time and lower energy consumption.

Mubarakali et al. [44] proposed a delay-sensitive data transmission algorithm based on edge computing to ensure low and predictable delay in delay sensitive IoT applications such as traffic monitoring and vehicle tracking. Li et al. [30] combined the long-distance data transmission function of fog infrastructure and the advantages of multi-opportunity routing of edge intelligent devices to design an efficient data-forwarding scheme for the mobile IoT, which not only reduces the data transmission delay and improves the successful delivery rate of packets but also ensures the safe and reliable transmission of data. Ghosh et al. [31] proposed a real-time mobile sensing framework under cloud–fog–edge cooperation, which realizes a hierarchical processing of information and uses a probability graph model to model the cumulative space–time trajectory of mobile agents, predict the next location of agents, and effectively reduce latency and power consumption. Leveraging the significant advantages of edge computing devices, Manogaran et al. [32] proposed an efficient resource allocation algorithm for the IoT with optimal node configuration to ensure the seamless transmission of non-delay-tolerant data from the required platform or infrastructure to the included users. Li et al. [45] introduced edge computing and a software-defined network (SDN) into the IoT, and put forward a secure edge framework based on SDN for a medical

IoT system, which can effectively improve the utilization of smart devices and the security of data access in this system.

The above research shows that applying edge computing can not only effectively avoid overloading the cloud center and reduce network congestion and service delay, but it can also provide a safe, reliable, and high-quality architecture support for the IoT system with different data transmission and processing requirements. Nevertheless, the application research of the integration of the IoT, cloud computing, and edge computing or fog computing is still in its initial stage, and a systematic reliable service model and method of the IoT have not yet been formed, especially in some important aspects such as data transmission, which need further in-depth research.

3. Load-Balanced, Data-Layered Transmission Strategy in IoT Architecture Based on Cloud–Edge–End (CEE) Collaboration

3.1. IoT Architecture under CEE Collaboration

As the terminal bodies of IoT services, various physical objects such as houses, bridges, local environments, plants, people, animals, etc., are also an indispensable part of IoT applications. Therefore, this paper refines the edge layer according to the type or communication mode of the devices and uses various communication technologies to build an open CEE collaborative IoT architecture based on physical terminals, intelligent edge devices, edge infrastructures, and cloud devices, as shown in Figure 1. The functions of the main components of the architecture are described below.

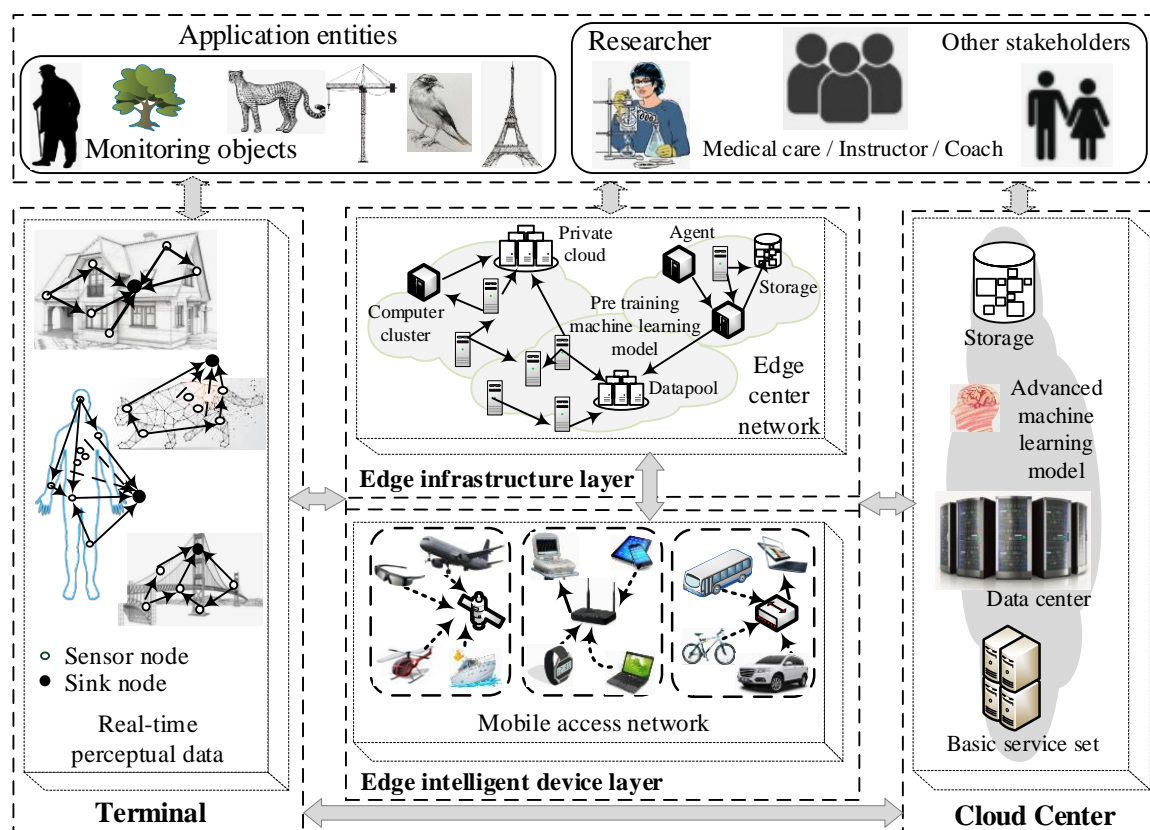


Figure 1. Open IoT service application architecture based on CEE collaboration.

(1) Physical terminal

It acts as data perception layer and is mainly responsible for identifying various physical objects and collecting real-time, near-real-time, and non-real-time context or index data from them. The data are mainly measured from the individual and their surrounding environment through small sensors, embedded systems, RFID tags and readers, medical diagnosis and healthcare equipment, medical and clinical imaging equipment, and any

equipment supporting data acquisition and transmission. These data are then transmitted to the corresponding sink node through a wireless ad hoc sensing network. As a short-range communication network centered on physical objects, wireless ad hoc sensor technology provides a good means to realize the real-time, objective, accurate, and low-cost IoT applications. It is the foundation of the research on IoT data transmission strategy.

(2) Edge cloud

- *Intelligent edge device layer*

The main significance of its existence is to facilitate the processing of data with strict requirements for reaction time. It includes many intelligent computing nodes such as various mobile intelligent devices or other small network sites similar to home routers, which can make the IoT service architecture form a decentralized architecture mode. With the help of sensors and execution controllers, intelligent edge devices can operate at the extreme edge of the architecture and can perform basic rule-based preprocessing operations (such as data filtering, aggregation, and fusion) on the data to be transmitted, which helps to optimize resource utilization. For example, Mahmud et al. [46] studied communication power consumption and computing power and pointed out that ensuring the objective on-demand system transmission rather than the on-demand transmission of users' subjective intention will be more helpful to optimize power consumption when communication consumes about five times the computing power. Obviously, the intelligent edge device layer is an important part of IoT application structure, which can provide more opportunities for the data transmission and computing of IoT.

- *Edge infrastructure layer*

This layer mainly includes various network infrastructures with small data center functions such as base stations, which can make computing resources and application services closer to the edge, so as to reduce the response delay. In terms of function, it supports local data storage, data filtering, data compression, data fusion, and intermediate data analysis; can conduct near/non-real-time comprehensive processing of perceived data; and feed the result information back to users as quickly as possible, so as to reduce one-time load in the cloud, improve the system performance and QoS, and save backbone bandwidth. Meanwhile, perception data can also be transferred to cloud center for more detailed analysis and comprehensive calculation.

(3) Cloud center

The cloud center can interact with physical terminals, the edge layer, and application entities. Among them, the index data gathered from edge layer can be transferred to cloud for long-term storage or for big data processing and advanced analysis. However, in fact, while using the cloud to calculate expensive operations or tasks, assigning appropriate computing tasks to the edge side will also greatly improve the efficiency and performance of the system.

In addition, the architecture can provide a user interface between some application entities at the top (such as researchers, instructors, medical care, coaches, applications and information systems) and the stakeholders/consumers of the service system. Through these interfaces, it can deliver various IoT services to their relevant users and also provide corresponding program developers and consumers with permissions or privileges to directly access relevant resources from the cloud center or the edge layer. Obviously, the application entities are the most interactive, but they have little analytical ability.

To sum up, the proposed architecture is an interoperable ecosystem that includes various devices, applications, and back-end systems. It can ensure that information/data flows are not disturbed, so as to make corresponding decisions accurately and timely [47]. Such an architecture can not only reduce the amount of data transmitted by IoT devices through rule-based data preprocessing but can also greatly reduce power consumption, delay, and computational complexity.

In this architecture, the main object of transmission and processing is terminal-aware data. Generally, IoT terminal sensors may generate three kinds of delay-sensitive data, namely, real-time data, near-real-time data, and offline/batch mode big data. Different demand types of data have different delay requirements for processing and transmission. Therefore, to ensure the delay-tolerant transmission of real-time data, near-real-time data, and big data, this paper proposes an adaptive hierarchical data transmission strategy based on the above IoT architecture under CEE collaboration.

3.2. Data Layered Transmission Scheme

Seen from Figure 1, based on data type, data flow size, and resource availability, computing tasks generated by rule-based preprocessing, pretraining machine learning, advanced machine learning, and big data analysis can be delegated to the most appropriate layer in the CEE collaborative IoT architecture for computing and processing, so as to realize the seamless communication between heterogeneous devices in the IoT. Therefore, this paper takes data as the key object and adopts the scheme of separating transmission path to classify real-time, near-real-time, and non-real-time data in order to obtain better QoS, lower delay, and minimum power consumption.

Figure 2 shows the data transmission and layered processing flow based on the architecture in Figure 1 and uses dotted lines to segment and clearly identify the important components of the architecture (including its hierarchical components and the data transmission and processing modules with self-expandable functions in the corresponding layers). Among them, the heterogeneous data source on terminals mainly includes the indicator data of the monitored objects, which provides basic data support for the operation of the entire IoT service.

Known from Figure 2, first, the terminal index data collected by sensing devices usually need to use a wireless ad hoc sensor network formed by surrounding sensors to transmit it to sink node. In this process, for different index data, compression processing and other technologies can be used to minimize the amount of data before transmission to reduce system energy consumption. Then, the flow direction is determined according to the attribute and function of indicator data. If a node in wireless ad hoc sensor network has no ability to process or save it, a high-performance data-forwarding module can be called directly to forward data to the corresponding sink node; otherwise, real-time and simple processing, such as for display, shall be carried out according to the preset rules. For example, in intelligent medical IoT applications, when a patient has blood pressure fluctuations and discomfort symptoms, it is necessary to first process the generated data and forward the decision to medical staff as soon as possible to prevent stroke. In this case, the data preprocessed from the mobile intelligent device layer is further processed in edge infrastructure layer and forwarded to the corresponding application entity to facilitate stakeholders to take necessary actions.

Sink nodes are generally sensor nodes with stronger processing capabilities such as smart phones and watches. These devices have certain data computing capabilities and can form a mobile access network environment of a certain scale. With the help of these devices, the opportunities for IoT data transmission increase and the system ability to process real-time/near-real-time data improves. Therefore, when the indicator data are transmitted to the edge-intelligent device layer, its flow direction can be determined according to its attributes and the computing capacity of devices. In case of real-time/near-real-time data, high-performance data-transmission mechanisms based on machine learning can be called to find target nodes, so as to timely process the data and feedback information. Otherwise, the data are directly transmitted to the edge center network equipment. For example, in the case of adverse drug reaction service in smart medicine application because adverse drug reaction has inherent universality, the drugs for treating specific diseases need to be diagnosed, and it is also necessary to understand or view the patient's past medical history. Therefore, the results diagnosed from patients or the data perceived by intelligent terminals are first forwarded to edge infrastructure network layer for drug identification.

Later, edge network infrastructure forwards the identified drugs to cloud center. After the cloud carefully analyzes the relevant electronic medical records and allergy files, it determines the compatibility of drugs, and finally sends the decision to relevant application entities for medical professionals to access.

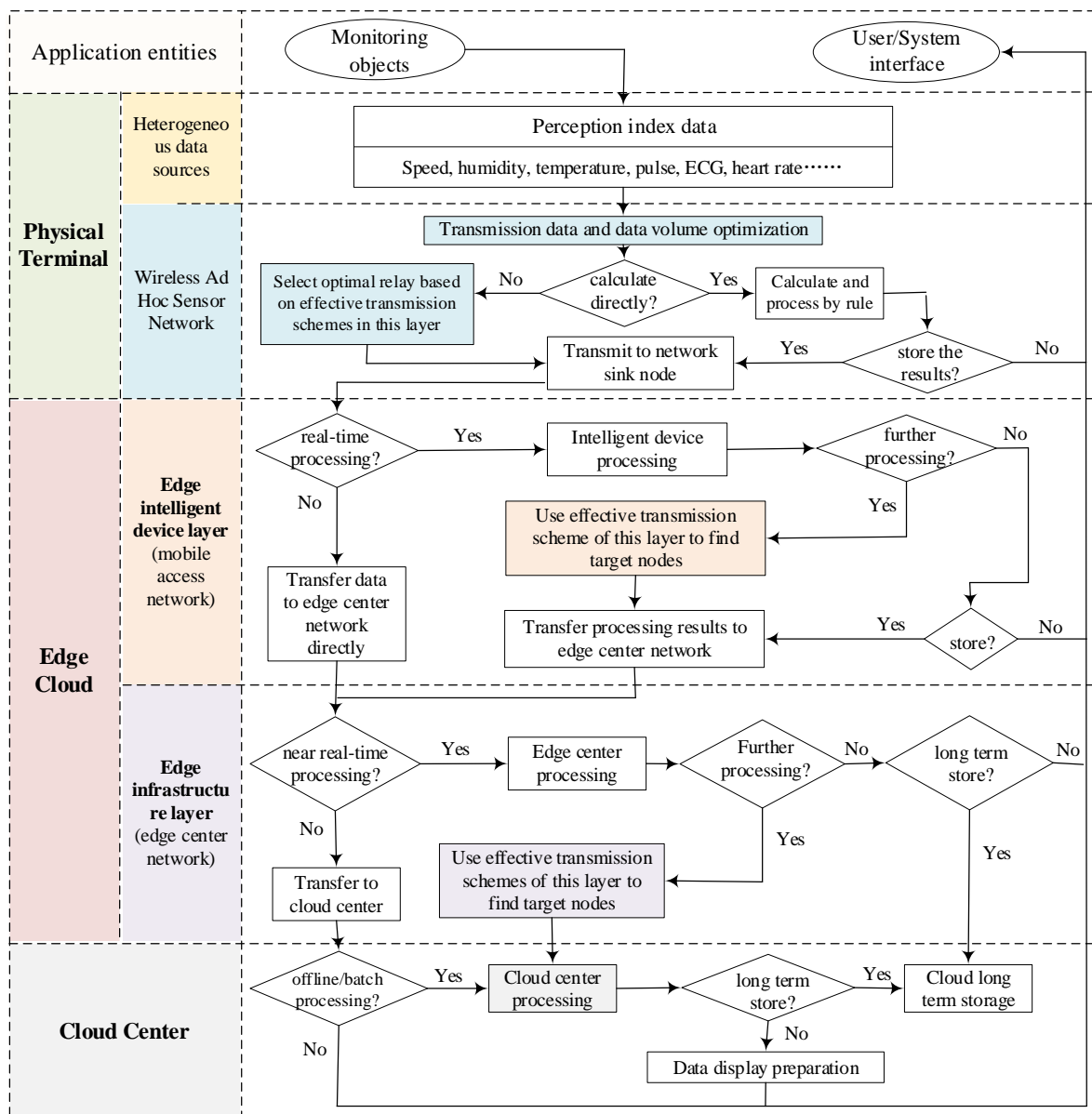


Figure 2. Hierarchical data transmission and processing flow based on CEE collaboration in IoT.

Usually, edge center network processes the indicator data of near-real-time or a small number of batch processing requirements. Therefore, if these types of data are not transferred, they will be directly transferred to the cloud center for analysis, calculation, storage, display, and other operations. Otherwise, optimal target nodes need to be found according to certain rules, and data will be transferred to it for processing, storage, or displaying to users. In addition, all non-real-time big data in the offline/batch mode need to be transferred to cloud center for processing operations such as computing, long-term storage, data display preparation, etc. For example, in the applications of intelligent medical IoT, each inspection of MRI will produce thousands of high-resolution images, which requires the cloud with more computing power and storage space to provide effective services. Thus, in this application, the image information will be directly forwarded to the cloud without any

processing or saving in the edge layer. Moreover, data from non-sensor sources such as electronic medical records, electronic health records, and electronic prescription platforms can also be seamlessly integrated in the cloud center.

To sum up, the layered data transmission strategy of IoT architecture under CEE collaboration has good scalability. That is, for the different network layers, the corresponding data processing schemes can be integrated at any time according to actual needs. For example, terminal wireless ad hoc sensor networks usually pay more attention to the energy consumption of sensor devices, mobile access networks pay more attention to the security of data transmission, and the edge network center and cloud center pay more attention to the energy efficiency of data transmission. Therefore, data transmission mechanisms with high energy efficiency, low energy consumption, or safety and reliability can be designed and developed at the corresponding layer.

3.3. Optimal Resource Allocation and Load-Balancing Policy

In the proposed IoT architecture under CEE collaboration, various types of data are first collected through sensors, and then, according to the data flow level or processing requirements, these data will be transmitted to the terminal convergence node, intelligent edge device layer, edge infrastructure layer, or cloud center for display, computing, or storage. In the whole process, the end-to-end delay, the cumulative delivery rate of packets, and system load balancing have become important indicators to measure the performance of the architecture and the quality of data transmission service. Therefore, to effectively process heterogeneous IoT data and obtain better user experience quality, it is necessary to set up a central control module to manage functions such as dynamic resource allocation and hierarchical data routing to meet the needs of various tasks. The corresponding application scenarios are shown in Figure 3. Seen from Figure 3, the central control module can manage the entire data transmission process and plays a vital role in the continuous and effective operations of the IoT. In practical applications, it can be considered to use SDN [48] to realize its functions because it can separate the control surface of network equipment from data surface and realize the flexible control of network traffic.

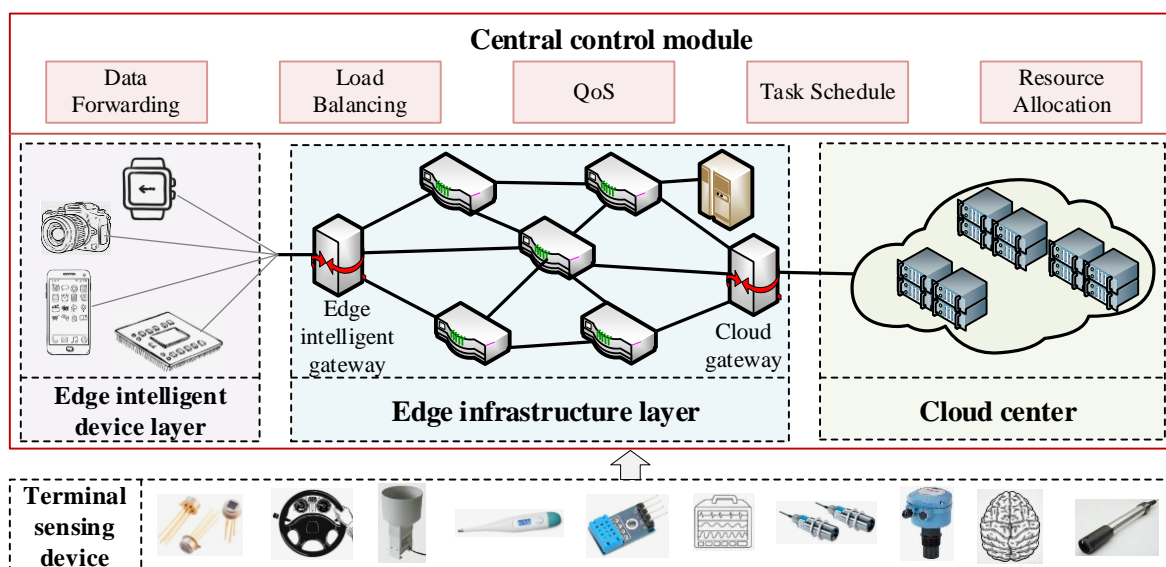


Figure 3. IoT data transmission scenario with central control module under CEE collaboration.

In addition, various heterogeneous data streams generated by IoT terminal devices are divided into three types of data: delay-sensitive, loss-sensitive, and hybrid data based on their different requirements for data transmission rate and queuing delay. Table 1 gives the basic characteristics of the three types of data such as transmission priority, service type, and typical application examples in the intelligent medical IoT. Meanwhile, to

effectively process heterogeneous IoT data and maintain a high QoS, this paper dynamically allocates the network resources based on hierarchical data transmission of IoT under CEE collaboration. The relevant parameters and performance indicators are as follows.

Table 1. Classification and characteristics of IoT Data.

Data Type	Data Rate	Queue Delay	Transmission Priority	Service Type	Typical Examples in Medical IoT
Delay-Sensitive	high	low	high	key data multimedia conference video data	real-time patient monitoring telephone conference video stream for elderly health monitoring or sports control
Loss-Sensitive	low	high	middle	judgement result image/video	electronic health records medical imaging
Hybrid	middle	middle	low	non-key professional parameters	measurement of outline physiological indexes of patients

3.3.1. Dynamic Resource Allocation Strategy

Reducing the data transmission delay (Td) and improving the cumulative delivery rate (Cdr) of packets can make IoT applications have a better QoS. Therefore, in the IoT architecture under CEE collaboration, all nodes in the edge mobile intelligent device layer can obtain optimal thresholds of data transmission delay and successful delivery rate by using a resource allocation optimization strategy.

In Figure 3, suppose that the total buffer length and output link capacity of an edge intelligent gateway node (such as an access site) are L and C , respectively, and the edge intelligent device layer has M mobile intelligent nodes connected to the gateway node. For ease of reading, the key parameters and their meanings of each intelligent node in the edge intelligent device layer are listed in Table 2.

Table 2. Mathematical symbols and their meanings.

Symbol	Meaning
P_{size}	the size of a packet on each intelligent node
Bd_i	the bandwidth demand of the i -th intelligent node
Bl_i	the buffer length requirement of the i -th intelligent node
Td_i	the total delay of all packets on the i -th intelligent node
Cdr_i	the cumulative delivery rate of all packets on the i -th intelligent node
P_{ic}	the ratio of the bandwidth demand of the i -th intelligent device node to the maximum capacity of edge intelligent gateway node
P_{il}	the ratio of the buffer length demand of the i -th intelligent device node to the total buffer length of edge intelligent gateway node
Q_{ic}	the demand ratio of the bandwidth of the i -th intelligent node
Q_{il}	the demand ratio of the buffer length of the i -th intelligent node
U_i	the total number of packets on the i -th intelligent node
R_i	the maximum resource assigned to the i -th intelligent device
t^{Td-i}	the transmission delay of packets on the i -th intelligent node
p^{Td-i}	the processing/calculation delay of packets on the i -th intelligent node
q^{Td-i}	the queuing delay of packets on the i -th intelligent node
Aql_i	the average queue length of packets on the i -th intelligent node

To meet the QoS demand of the i -th intelligent node, the users' requirements and corresponding resource allocation requirements are set to (Td_i, Cdr_i) and (Bd_i, Bl_i) , respectively. Since the resources allocated to the i -th device are proportional to that required by the user, R_i assigned to the i -th intelligent device is shown as follows:

$$R_i = \max \left\{ \frac{Bd'}{C}, \frac{Bl'}{L} \right\} = \max \left\{ \frac{Bd_i}{C} \frac{Bd'}{Bd_i}, \frac{Bl_i}{L} \frac{Bl'}{Bl_i} \right\} = \max \{ P_{ic} Q_{ic}, P_{il} Q_{il} \} \quad (1)$$

where $P_{ic} = Bd_i/C$ represents the ratio of the bandwidth demand of the i -th intelligent device node to the maximum capacity of the intelligent edge gateway node, and $P_{il} = Bl_i/L$ represents the ratio of the buffer length demand of the i -th intelligent device node to the total buffer length of the intelligent edge gateway node; $Q_{ic} = Bd'/Bd_i$ and $Q_{il} = Bl'/Bl_i$ are the demand ratios of the bandwidth and buffer length of the i -th intelligent node respectively.

Generally, the whole response time from the user sending the request to the system returning the result is determined by the end-to-end data delay, and the total end-to-end delay based on the M/D/1 queue model [49] includes the data transmission delay t^{Td-i} , the processing delay p^{Td-i} , the queuing delay q^{Td-i} , and the delay of the processing result return. However, since the delay of the result return is usually much smaller than the first three delays, it can be ignored. Then, the total delay Td_i can be calculated by Equation (2):

$$Td_i = t^{Td-i} + p^{Td-i} + q^{Td-i} = \sum_{cld} \sum_{e-ctr} \sum_{e-md} \left[\frac{U_i P_{size}}{C} + \left(\frac{\varepsilon}{2\kappa(\kappa - \varepsilon)} + \frac{1}{\kappa} \right) + a\varepsilon \right] \quad (2)$$

where cld represents the cloud center; $e-ctr$ and $e-md$ represent the edge infrastructure layer and the intelligent edge device layer, respectively; a is the constant duration required for a single processor to complete a task; and ε and κ refer to the arrival rate and service rate of packets, respectively.

When the average queue length of packets on the i -th device node is larger than the required buffer pool length, that is, $Aql_i > Bl_i/P_{size}$, packets may be lost. Therefore, based on the relationship between the loss rate and cumulative delivery rate of packets [49], the latter can be expressed by Equation (3) below:

$$Cdr_i = \frac{\frac{Bl_i}{P_{size}}}{Aql_i} \quad (3)$$

Finally, the problem of dynamic optimal allocation of system resources is transformed into the following equation :

$$\max\{(Q_{1c}, Q_{1l}), \dots, (Q_{ic}, Q_{il}), \dots, (Q_{Mc}, Q_{Ml})\} \quad s.t. \quad \begin{cases} \sum_{i=1}^M Bd_i \leq C & \& \sum_{i=1}^M Bl_i \leq L \\ \min\left(\sum_{i=1}^M Td_i\right) & \& \max\left(\sum_{i=1}^M Cdr_i\right) \end{cases} \quad (4)$$

3.3.2. System Load-Balancing Scheme

In each network access layer, the end-to-end transmission delay can be reduced through the link distribution and the link fusion technologies. In reality, multiple different network facilities or devices can transmit data at the same time, and thus, the link scheduler can choose multiple links in an access layer such as the intelligent edge device layer or the edge infrastructure layer, allocate data streams to reduce the end-to-end delay, and, finally, aggregate data streams at the other end of the corresponding network layer.

When data packets are transmitted from one end of a network layer to the other, the link scheduler needs to coordinate all devices in this layer for efficient collaborative transmission. Obviously, network throughput and cumulative packet delivery rate are the most suitable measures of the performance of this process. This is because the network throughput can reflect the utilization or saturation of equipment resources well and the packet delivery rate reflects how many packets the target node can successfully receive from the source node. The larger the network throughput, the more saturated the resource utilization; the higher the delivery rate, the more packets that successfully arrive at the target node, and the more conducive it is for the system to perform timely and rapid processing and responses to user requests. At the same time, the bandwidth and buffer length are also directly related to the performance of data transmission. Therefore, this paper considers the use of the throughput, packet delivery rate, and the requirements of the bandwidth and buffer length to build the utility of link calls. If the link scheduler chooses N

links according to users' requirements, the link-adaptive optimization problem is expressed as follows according to [49]:

$$\max \ell \left(Tp, \frac{1}{1-Cdr} \right) \quad s.t. \quad \left(\sum_{n=1}^N \xi_n Bd \leq C \quad \& \quad \sum_{n=1}^N \xi_n Bl \leq L \right) \quad (5)$$

where Tp is the system throughput, Bd and Bl are the bandwidth demand and buffer length demand of the corresponding gateway node, and ξ_n represents the percentage of bandwidth/buffer length allocated by load balancer, that is, the proportion of resources allocated in the n -th link. To reflect the objectivity of the experimental results, this paper uses the weighted sum of the ratio of the demand of the n -th link for each resource to the total demand of all links for each resource to calculate ξ_n in Equation (5), which is shown as follows:

$$\xi_n = \frac{Bd_n}{\sum_n Bd_n} \gamma + \frac{Bl_n}{\sum_n Bl_n} (1 - \gamma) \quad (6)$$

For delay-sensitive, loss-sensitive, and hybrid data in IoT applications, based on the characteristics of data types, γ is assigned the corresponding values: 1, 0, and 0.5, respectively.

In the proposed IoT architecture under CEE collaboration, its data-transmission layered processing strategy is actually to choose N links to coordinate the load and task allocation according to the data requirements. Obviously, optimizing the load balancing of the IoT system is to optimize the load balancing between the service links in the process of data transmission. The main pseudocode of the load-balancing scheme is shown in Algorithm 1.

In Algorithm 1, when the IoT business process starts, the data are divided into corresponding categories according to the actual needs. For each output link n , ξ_n is first calculated according to Equation (6). For the delay-sensitive and loss-sensitive service data, the load is allocated according to the required bandwidth and buffer length, respectively. For hybrid data, the load is allocated according to the larger demand, which is reflected by the operations when $r = 0.5$ (line 14–18) in Algorithm 1. That is to say, if the hybrid data have a large demand for bandwidth, the load will be allocated to the corresponding link according to the calculation standard of the bandwidth. On the contrary, the load will be allocated to the link according to the calculation standard of the buffer length. Obviously, the method can make the system more self-adaptive and objective when dealing with hybrid data, so as to ensure that the experimental results are more accurate. Simultaneously, when the number of available service links is N , the time complexity of the algorithm is $O(N)$, which is much lower than other existing frameworks. The main reason is that the framework proposed in this paper has the ability of hierarchical data processing and enables data to be calculated and processed on the corresponding network layer based on the corresponding category, greatly reducing the amount of calculation required by other network layers, thus reducing the computational complexity of the whole system.

Algorithm 1 Load-Balancing Scheme.

Input:	Given queue buffer length L ; gateway node output link capacity C ; bandwidth length requirement eBd ; buffer length requirement eBl ;
Goal:	Optimal load distribution.
1:	Initialize $\gamma, \chi, \lambda, \delta$;
2:	Link scheduler selects N links based on data requirements;
3:	SWICH (data transmission priority) {
4:	CASE 1: $\gamma = 1$; BREAK ;
5:	CASE 2: $\gamma = 0$; BREAK ;
6:	CASE 3: $\gamma = 0.5$; BREAK ;
7:	}
8:	FOR ($n = 1; n \leq N; n++$) {
9:	Calculate ξ_n according to Equation (6);
10:	$\chi_n = \xi_n \times eBd$; $\lambda_n = \xi_n \times eBl$;

```

11:    SWICH ( $\gamma$ ) {
12:        CASE 1: allocate load to link  $n$  based on  $\chi_n$ ;
13:        BREAK;
14:        CASE 0: allocate load to link  $n$  based on  $\lambda_n$ ;
15:        BREAK;
16:        CASE 0.5:  $\Omega = eBd / \sum_n Bd_n$ ;  $\bar{U} = eBl / \sum_n Bl_n$ ;  $\delta = \max(\Omega, \bar{U})$ ;
17:        IF  $\delta = \Omega$ 
18:            THEN allocate load to link  $n$  based on  $\chi_n$ ;
19:        ELSE allocate load to link  $n$  based on  $\lambda_n$ ;
20:        BREAK;
21:    }
22: }
```

In short, the proposed IoT architecture can choose the appropriate data-transmission path based on different data sources to reduce the delay and increase the cumulative delivery rate of packets. Secondly, by delegating the corresponding data flow to the appropriate layer with relatively less load, it can ensure the best use of system resources. Meanwhile, in the case of system load balancing, it takes into account minimizing the transmission delay of packets, ensures that the sensitive data are transmitted first, and realizes the most favorable resource allocation.

4. Experiments and Performance Analysis

4.1. Experimental Environment and Parameters Setting

To verify the feasibility, rationality, and robustness of the layered data transmission of the IoT architecture under CEE collaboration, this section will set up experimental environment and data proof based on the above conditions. Given the unreality of constructing real scenes, the following experiments will use MATLAB R2020b as the test platform for simulation verification, and the computer running environment configuration is the Intel Core i7-10700 CPU at 2.9 GHz 4.8 GHz, RAM 16G, and 64-bit Windows 10 OS.

Assume that there is one cloud server, 15 edge infrastructure nodes, and 150 mobile intelligent edge devices. The communication connection between the mobile intelligent edge devices and edge infrastructure nodes and between the edge infrastructure and cloud server adopt the IEEE 802.11 a/g standard. To improve spectrum efficiency, reduce the burden of the edge center network, the delay, and the power consumption, D2D (device-to-device) communication technology is used for data transmission or exchange between mobile edge intelligent devices. The network has a total of 20,000 data packets, and the size of each packet is randomly generated from 100 B to 50 KB. The packet type is the delay-sensitive or loss-sensitive data, and original data packets generated or collected by mobile edge intelligent devices can be processed by using the resources on edge infrastructures. When the edge infrastructure is difficult to process (for example, the processing delay or the resource occupancy rate is too high), these data packets can also be directly unloaded to the corresponding cloud server to improve system operation efficiency and QoS. In the whole process of data transmission and operation, the transmission delays of the cloud server, edge infrastructure, and edge mobile intelligent devices are the random numbers between 15 and 35 ms, 0.5 and 1.5 ms, and 1 and 2 ms, respectively. The availability of processes is randomly generated. The connection bandwidth between nodes and the length of task queues on nodes are also randomly distributed; however, the higher-level nodes are still selected optimally based on the actual needs of tasks and the processing delay. In addition, to make the experimental results more in line with the characteristic requirements of heterogeneity IoT under CEE collaboration, it is assumed that the processing rates of cloud server to edge infrastructure and the latter to mobile edge intelligent devices are set to 100:1 and 1000:1, respectively.

4.2. Comparison Metrics

Known from the proposed resource allocation and load-balancing scheme in Section 3.3, this paper uses the bandwidth and buffer length of the intelligent device nodes as the measurement indicators of resource utilization, and within the allowable range of the two, converts the problem of system resource optimal allocation under the proposed network architecture into the problem of making service data transmission delay small and cumulative delivery rate high. At the same time, this paper also chooses the link scheduler constructed by system throughput and cumulative delivery rate as the main processing method to maintain the load balancing of the system. Therefore, this paper takes the average delay and cumulative delivery rate of packets, as well as system throughput and task distribution during the service architecture operation as the four comparison indicators to measure the performance of the proposed optimal resource allocation scheme, the load balancing algorithm, and the CEE collaborative data layered transmission strategy based on the two. Their definitions in this paper are shown as follows.

- Average delay Ad

This indicator means the average of the total delay of all packets to be processed in the network and can be gained according to the total delay of all packets on each node (Td_i) defined in Equation (2) and the number of mobile intelligent nodes (M) in the intelligent edge device layer. Its calculation is shown as follows:

$$Ad = \frac{1}{M} \sum_{i=1}^M Td_i \quad (7)$$

- Cumulative delivery rate Cdr

The cumulative delivery rate represents the average delivery rate of all packets in the network. Since the average delivery rate of all data packets on the i -th intelligent node can be obtained by Equation (3), the cumulative delivery rate of data packets of the whole network can be calculated by Equation (8) below:

$$Cdr = \frac{1}{M} \sum_{i=1}^M Cdr_i \quad (8)$$

- System throughput St

System throughput refers to the total number of data bytes transmitted by all equipment nodes in the network in a unit time. Suppose that t_{send-i} means the time when the i -th intelligent node starts to send packets, t_{rec-i} is the time when the system ends receiving the packets on the i -th intelligent node, and $Byte_i$ represents the total number of bytes of packets successfully reaching the target nodes on the i -th intelligent node. Then, the calculation of system throughput is shown as follows.

$$St = \sum_{i=1}^M \frac{Byte_i}{t_{rec-i} - t_{send-i}} \quad (9)$$

- Task distribution Td

Task distribution usually refers to the ratio of the number of tasks running in the network to the total number of tasks put into the network at a certain time, and this paper regards a packet as a task. Suppose that there are dp_i packets of the i -th intelligent node running in the network within a certain period of time, and the network has a total of 20,000 packets. Thus, the task distribution of cloud center can be gained by the following equation:

$$Td = 1 - \frac{\sum_{i=1}^M dp_i}{20000} \quad (10)$$

4.3. Performance Analysis

According to the above experimental parameters and simulation environment settings, Figures 4–8 show the performance of the CEE collaborative data layered transmission scheme of the IoT based on the optimal resource allocation and load balancing algorithms in terms of the task distribution of network, the average delay and cumulative delivery rate of packets, and the system throughput under different conditions, respectively. In these experiments, the number of edge infrastructure nodes varies from 1 to 15 as needed to verify the adaptability, robustness, and effectiveness of the proposed strategy because its impact on the performance of the IoT system is more obvious than that of cloud and other intelligent devices.

Figure 4 shows the impact of different numbers of adjacent edge infrastructure nodes on system task allocation under the network architecture with only edge infrastructure layer and the architecture with edge infrastructure layer and cloud center. Simulation results show that, no matter which network architecture is adopted, when more adjacent edge infrastructure nodes are included in the data hierarchical transmission process, the load of each edge infrastructure node and cloud center gradually decreases with the improvement in the overall processing performance of edge infrastructure nodes. For example, when there is only one adjacent edge infrastructure node, system load containing only the edge infrastructure layer is about 55%, and the load of the cloud center containing both is about 73%. When the number of adjacent edge infrastructure nodes is nine, the corresponding load drops exponentially to 21% and 36%, respectively. After that, with the increase in the number of edge infrastructures, the load of the cloud center tends to be stable. For example, it only reduces by about 11% and 15%, respectively, when the number of adjacent edge infrastructure nodes is 15. This is because the total number of packets in the network is fixed. When the edge infrastructure reaches a certain number, it can help the system accommodate the packets put into it and process them quickly.

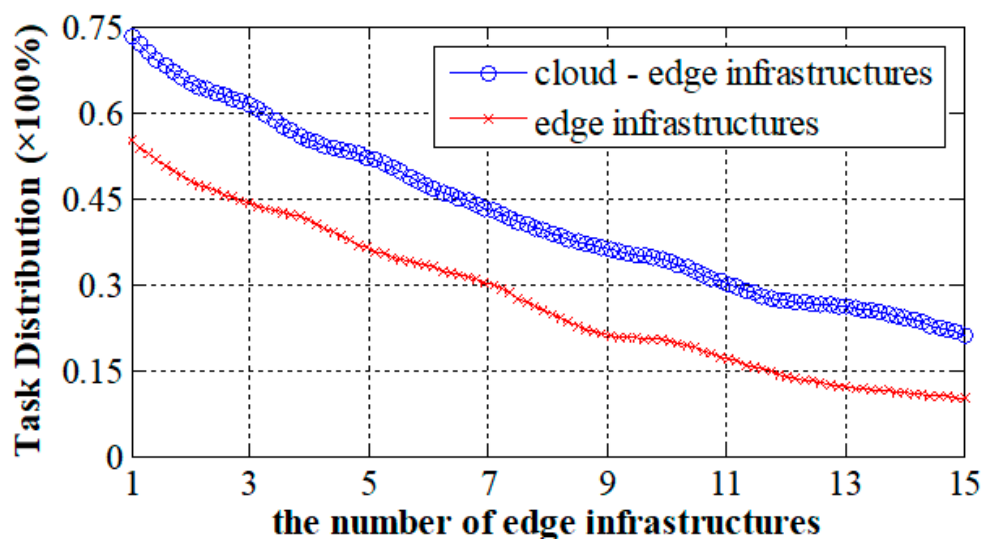


Figure 4. Influence of the number of adjacent edge infrastructure nodes on system task distribution under two different IoT architectures.

According to the experimental environment and simulation parameters set in Section 4.1, Figure 5 shows the average delay of the end-to-end data transmission and processing in the IoT architecture with different network hierarchies. Known from Figure 5, when mobile intelligent edge devices and edge infrastructures participate in the task computing together with the cloud center, the average delay of the end-to-end transmission is smaller, and especially, the average delay of packets in the architecture including more edge infrastructures is the smallest. For example, the average delay of packets in the architecture with 1 cloud, 15 edge infrastructures, and 150 mobile intelligent devices is about 71 ms, which

is 23.7%, 35.5%, and 51% lower than that of the architecture containing 1 cloud, 10 edge infrastructures, and 150 mobile intelligent devices; the architecture containing 1 cloud and 10 edge infrastructures; and the architecture containing only the cloud, respectively. In other words, the participation of more edge mobile intelligent devices and adjacent edge infrastructures can reduce the end-to-end delay because this process reduces the probability of network congestion and the queuing and transmission time of packets. Obviously, as the number of adjacent edge infrastructure nodes increases, the delay of task computing and data transmission will gradually decrease, and the response time of the system is also relatively shortened.

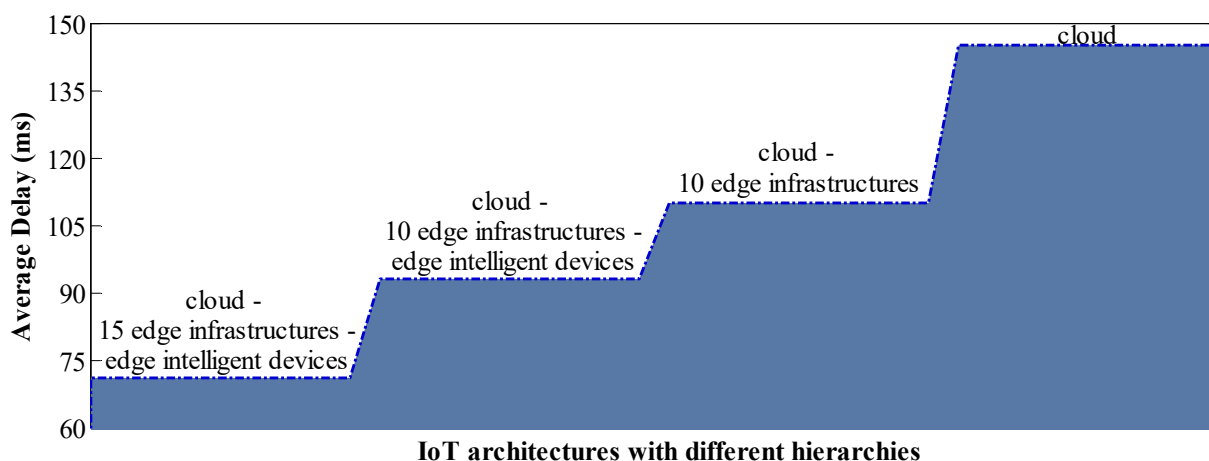


Figure 5. Comparison of the average delay of end-to-end data transmission and processing in four different IoT architectures.

For the IoT architecture with different network hierarchies or different numbers of edge infrastructure nodes, Figure 6 shows the impact of system buffer size on the cumulative delivery rate of packets. Apparently, with the gradual expansion of the buffer area, the cumulative delivery rate of the transmitted data packets in the IoT system with any architecture will increase, and accordingly, the data packet loss rate will decrease. Nevertheless, regardless of the buffer size, the cumulative delivery rate of packets in the IoT architecture with more edge infrastructures and mobile intelligent devices is the highest, that of packets in the architecture with only edge infrastructures is lower than the former, and that of packets in the architecture only including mobile intelligent devices is the lowest. For example, when the buffer size is 35 MB, the cumulative delivery rate of the two architectures with 15 and 10 edge infrastructures and mobile intelligent device is 73% and 66%, respectively, of which the largest ones are 12.3%, 15.9%, and 19.7% higher than that of packets in the architectures with only 10, 6, and 3 edge infrastructures, respectively, and about 2.5 times that of packets in the architecture only including mobile intelligent devices. Obviously, the greater the number of edge infrastructures is, the higher the cumulative delivery rate of packets in the system. The main cause is that compared with mobile intelligent devices, edge infrastructure has better computing and storage performance and has the advantages of being closer to users, being able to process medium and large-scale data, and being convenient for installation and maintenance compared with the cloud center. In addition, the increase in buffer size can usually increase the queuing delay of task data and the total delay. Therefore, to ensure that the system has a high packet delivery rate and low transmission delay, it is necessary to select the appropriate buffer size.

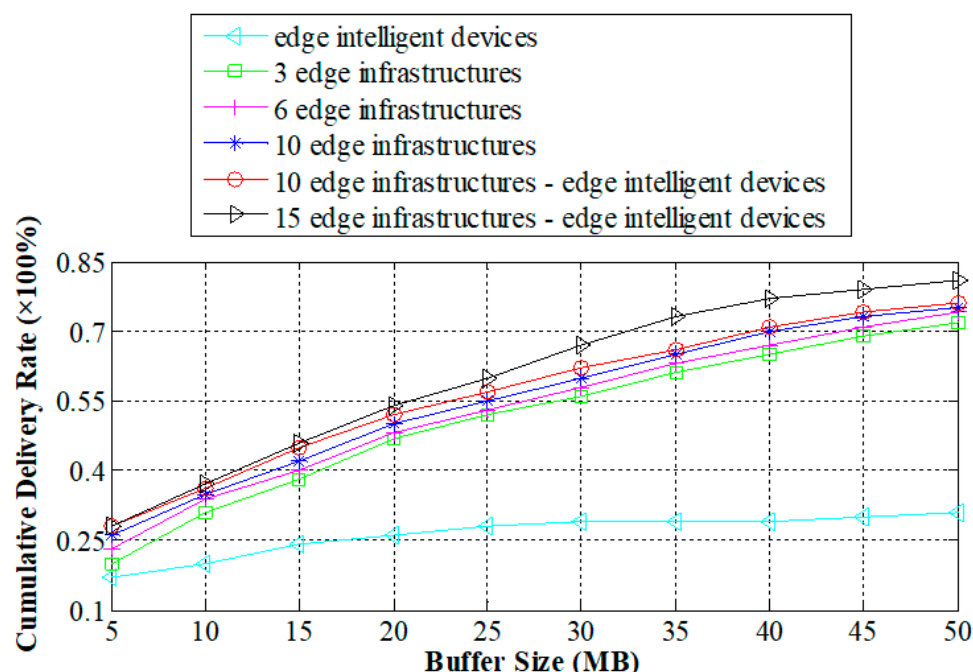


Figure 6. Impact of the buffer size on Cumulative delivery rate of packets under different IoT architectures.

Figure 7 reflects the change in the system throughput with the increase in the number of transmitted packets in the network under four different IoT architectures when applying the optimal resource allocation and load balancing scheme in Section 3.3. Seen from the figure, in the architecture that only includes edge mobile intelligent devices, due to the limited system resources and processing capacity of mobile devices, a large number of bottleneck congestion phenomena occur with the increase in the number of transmitted packets. These bottleneck congestions are not easy to solve or adjust. Therefore, the system throughput of the architecture rapidly drops after reaching a certain inflection point. When five edge infrastructure nodes are added to the above architecture, its ability to handle complex non-real-time computing is correspondingly enhanced. Thus, when the number of transmitted packets is the same, the IoT architecture, including edge infrastructures, has better performance on the system throughput. However, as the number of transmitted data packets increases, the number of links needs to increase continuously to ensure higher throughput, and meanwhile, different types of data are best transmitted to devices with corresponding capabilities for processing. However, due to the lack of the ability to process large and complex data such as comprehensive case analysis, and the limited number and capacity of links, the system throughput decreases with the increasing number of packets. Although the system throughput of the IoT architecture composed of cloud, edge infrastructures, and mobile intelligent edge devices increases first and then decreases with the increase in the number of transmitted packets, it still has the best performance on the system throughput compared with the first two architectures. This is because each transmission device can dynamically allocate data to the least congested link, and simultaneously, high-quality forwarding is also used to transmit packets to the most suitable target nodes for computing, so as to improve the overall network performance. In addition, the system throughput of the architecture with more edge infrastructures is bigger, and the main cause is same as that of the cumulative delivery rate of packets in Figure 6.

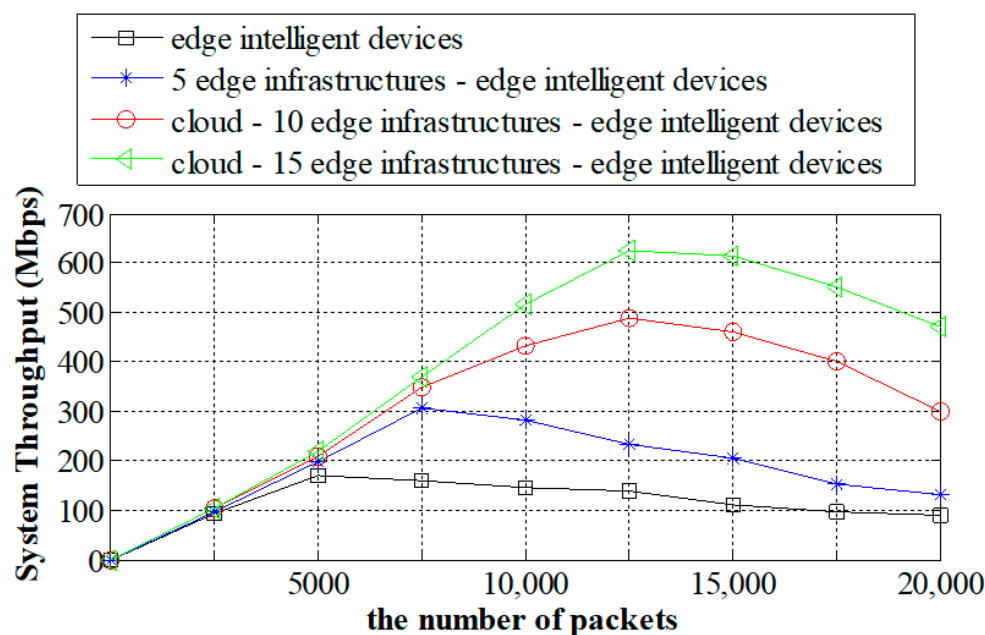


Figure 7. Impact of the number of transmitted packets on system throughput under the four IoT architectures.

To further verify the rationality and robustness of an IoT layered data transmission scheme under CEE collaboration, this paper compares it with the IoT framework [36] based on no subdivision of edge layer in terms of the average delay of packets for three different network environments: light load, medium load, and heavy load and the comparison results are shown in Figure 8. Obviously, in any system environment, when the same amount of data is transmitted, compared with the literature [36], the data layered transmission scheme proposed in this paper always takes the least time and has faster processing speed whether it contains 10 or 15 edge infrastructures. For example, when 80 KB data is transmitted, the average delay of the proposed architecture with 10 edge infrastructures is about 2.5 ms, 3.4 ms, and 5.7 ms, respectively, under the light, medium, and heavy load of the system, and the transmission speed is about 36%, 23.5%, and 1.7% higher than that of the framework in [36] under the same conditions. At the same time, comparing the proposed architectures with two different configurations, the average delay of the architecture with 15 edge infrastructures is obviously lower than that with 10 edge infrastructures, and the difference is about 9% on average between the two under the heavy load of the system, which means that the performance of the former must be better than that of the framework in [36]. To sum up, compared with the comparison framework, the framework of these two configurations proposed in this section can improve the service response efficiency by an average of 27.3%, 12.7%, and 8%, respectively, under the light, medium, and heavy load of the system. The above experiments fully illustrate the importance and necessity of subdividing the edge layer into the mobile intelligent edge device layer and the edge infrastructure network layer according to the type of data to be processed. Among them, the mobile intelligent device layer is used to process small-scale, real-time data, and the edge infrastructure layer can process near-real-time data that are not sensitive to delay or even some non-real-time big data.

Apparently, after subdividing the edge layer, the proposed IoT architecture under CEE collaboration has a better data-filtering function and can reasonably allocate resources according to the actual needs of users. It can not only reduce the time of data transmission and the amount of real-time data transmitted along the network, but it can also reduce the computational complexity of the application system based on this architecture, which significantly improves system operation efficiency, QoS, and user experience quality.

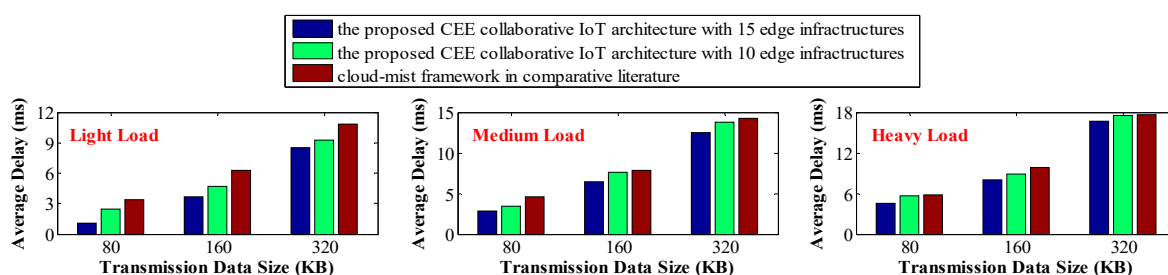


Figure 8. Comparison of Average Delay for two architectures to transmit the same amount of Data in three different Network environments.

5. Discussion

This paper refines edge cloud based on the different computing ability of edge intelligent devices and edge infrastructures, and proposes a CEE collaborative heterogeneous IoT architecture. Meanwhile, the corresponding data layered transmission strategy, the optimal resource allocation scheme and the load balancing algorithm are designed or developed for this IoT architecture. Although a large amount of experiments have proved the effectiveness and accuracy of this CEE collaborative architecture by measuring its task distribution, average delay, cumulative delivery rate and system throughput under different conditions, it still exists three major weaknesses as follows.

- (1) The proposed architecture does not take into account the differences in communication technology standards between different types of devices, especially among devices with different purposes, which makes the experimental results somewhat unrealistic.
- (2) This paper only uses a random way to select the optimal target nodes from multiple nodes, and does not provide corresponding optimal data forwarding models for each layer in the proposed CEE collaborative IoT architecture, which makes the experimental results lack some objectivity.
- (3) Although experiments verify that the IoT architecture with more edge infrastructures has the best performance in all aspects, this paper does not test and evaluate the optimal configuration proportion of the number of various devices, which may lead to a waste of resources, because the amount of data packets in the experiments is fixed and too many edge facilities will make some of them idle for a long time.

The above three shortcomings need to be improved or solved and will be our further research contents in the future.

6. Conclusions

According to the different types and processing requirements of data transmitted in IoT applications, this paper uses the advantages of edge computing to build a collaborative CEE and multi-layer heterogeneous IoT architecture, including the center cloud, edge cloud, and terminals, and designs the corresponding data-layered transmission strategy for it. In particular, considering the difference of the data-processing ability between mobile intelligent devices and devices that act as small data centers at the network edge, this paper first subdivides the edge cloud into the intelligent edge device layer and the edge infrastructure layer. Then advanced machine learning methods are used to identify heterogeneous data flows so as to route them hierarchically and reduce the overall computational complexity of IoT applications by improving data distribution. Based on this proposed architecture, the terminals can complete the transmission and convergence of index data after collection through wireless ad hoc sensor network formed by its surrounding sensors. Third, the data in the sink nodes will be transmitted to intelligent edge devices, edge infrastructure, or cloud center for judgment, calculation, analysis, and other operations according to its characteristics or the actual needs of users. Then, the corresponding result information will be fed back to the user–system interface for display, so that users can view or implement other decisions. Experimental results show that the layered data transmission scheme

of IoT architecture under CEE collaboration has good applicability. It can achieve the optimal allocation of system resources through data flow control, reduce transmission delay between devices, and improve packet delivery rate and system QoS.

Author Contributions: Conceptualization and methodology, J.L. and X.L.; writing, J.L.; supervision, X.L.; data curation and validation, J.L. and J.Y.; investigation and visualization, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Joint Fund of NSFC-General Technology Fundamental Research (grant number U1836215, X.L.), the National Natural Science Foundation of China (grant number 6200208, J.Y.), and the Key Scientific Research Projects in Colleges and Universities in Henan (grant number 21A520025, J.L.).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to appreciate the editors and the anonymous reviewers for their insightful suggestions to improve the quality of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
CEE	Cloud-Edge-End
QoS	Quality of Service
RFID	Radio Frequency Identification
D2D	Device-to-Device

References

1. Yaqoob, I.; Hashem, I.A.T.; Ahmed, A.; Kazmi, S.A.; Hong, C.S. Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges. *Future Gener. Comput. Syst.* **2019**, *92*, 265–275. [\[CrossRef\]](#)
2. Li, J.; Li, X.; Gao, Y.; Gao, Y.; Fang, B. Review on Data Forwarding Model in Internet of Things. *J. Softw.* **2018**, *29*, 196–224.
3. Kandhoul, N.; Dhurandher, S.K. An Efficient and Secure Data Forwarding Mechanism for Opportunistic IoT. *Wirel. Pers. Commun.* **2021**, *118*, 217–237. [\[CrossRef\]](#)
4. Chi, Z.; Li, Y.; Sun, H.; Huang, Z.; Zhu, T. Simultaneous Bi-Directional Communications and Data Forwarding Using a Single ZigBee Data Stream. *IEEE/ACM Trans. Netw.* **2021**, *29*, 821–833. [\[CrossRef\]](#)
5. Jiang, Y.; Ge, X.; Yang, Y.; Wang, C.; Li, J. 6G oriented blockchain based Internet of things data sharing and storage mechanism. *J. Commun.* **2020**, *41*, 48–58.
6. Haseeb, K.; Din, I.U.; Almogren, A.; Ahmed, I.; Guizani, M. Intelligent and Secure Edge-enabled Computing Model for Sustainable Cities using Green Internet of Things. *Sustain. Cities Soc.* **2021**, *68*, 102779. [\[CrossRef\]](#)
7. Canger, F.; Curran, K.; Santos, J.; Moffett, S.; Cadger, F.; Curran, K.; Santos, J.; Moffett, S. Location and mobility-aware routing for multimedia streaming in disaster telemedicine. *Ad Hoc Netw.* **2016**, *36*, 332–348.
8. Jamil, F.; Iqbal, M.A.; Amin, R.; Kin, D.H. Adaptive thermal-aware routing protocol for wireless body area network. *Electronics* **2019**, *8*, 47. [\[CrossRef\]](#)
9. Euch, J.; Zidi, S.; Laouamer, L. A hybrid approach to solve the vehicle routing problem with time windows and synchronized visits in-home healthcare. *Arab. J. Sci. Eng.* **2020**, *45*, 10637–10652. [\[CrossRef\]](#)
10. Prasad, C.R.; Bojja, P. A non-linear mathematical model-based routing protocol for WBAN-based health-care systems. *Int. J. Pervasive Comput. Commun.* **2021**, *17*, 447–461. [\[CrossRef\]](#)
11. Singh, R.R.; Yash, S.M.; Shubham, S.C.; Indragandhi, V.; Vijayakumar, V.; Saravanan, P.; Subramaniaswamy, V. IoT embedded cloud-based intelligent power quality monitoring system for industrial drive application. *Future Gener. Comput. Syst.* **2020**, *112*, 884–898. [\[CrossRef\]](#)
12. Mohiuddin, I.; Almogren, A. Security Challenges and Strategies for the IoT in Cloud Computing. In Proceedings of the 11th IEEE International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 7–9 April 2020.
13. Alkadi, O.; Moustafa, N.; Turnbull, B.; Choo, K.K.R. A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks. *IEEE Internet Things J.* **2021**, *8*, 94631–99472. [\[CrossRef\]](#)
14. Arulanthu, P.; Perumal, E. An intelligent IoT with cloud centric medical decision support system for chronic kidney disease prediction. *Int. J. Imaging Syst. Technol.* **2020**, *30*, 815–827. [\[CrossRef\]](#)

15. Zhang, J.; Gao, F.; Ye, Z. Remote consultation based on mixed reality technology. *Glob. Health J.* **2020**, *4*, 31–32. [\[CrossRef\]](#)
16. Tong, J.; Wu, H.; Lin, Y.; He, Y.; Liu, J. Fog-computing-based short-circuit diagnosis scheme. *IEEE Trans. Smart Grid* **2020**, *11*, 3359–3371. [\[CrossRef\]](#)
17. Velu, C.M.; Kunar, T.R.; Manivannan, S.S.; Saravanan, M.S.; Babu, N.K.; Hameed, S. IoT enabled Healthcare for senior citizens using Fog Computing. *Eur. J. Mol. Clin. Med.* **2020**, *7*, 1820–1828.
18. Zhou, Y.; Zhang, D. Near-end cloud computing: Opportunities and challenges in the post-cloud computing era. *Chin. J. Comput.* **2019**, *42*, 677–700.
19. Patra, B.; Mohapatra, K. *Cloud, Edge and Fog Computing in Healthcare*; Springer: Singapore, 2021; pp. 553–564.
20. Verba, N.; Chao, K.M.; James, A.; Goldsmith, D.; Fei, X.; Stan, S. Platform as a service gateway for the Fog of Things. *Adv. Eng. Inform.* **2017**, *33*, 243–257. [\[CrossRef\]](#)
21. Ma, L.; Liu, M.; Li, C.; Lu, Z.; Ma, H. A Cloud-Edge Collaborative Computing Task Scheduling Algorithm for 6G Edge Networks. *J. Beijing Univ. Posts Telecommun.* **2020**, *43*, 66–73.
22. Abbasi, M.; Mohammadi-Pasand, E.; Khosravi, M.R. Intelligent workload allocation in IoT-Fog-cloud architecture towards mobile edge computing. *Comput. Commun.* **2021**, *169*, 71–80. [\[CrossRef\]](#)
23. Kaur, A.; Singh, P.; Nayyar, A. *Fog Computing: Building a Road to IoT with Fog Analytics*; Springer: Singapore, 2020; pp. 59–78.
24. Rekha, G.; Tyagi, A.K.; Anuradha, N. *Integration of Fog Computing and Internet of Things: An Useful Overview*; Springer: Cham, Switzerland, 2020; pp. 91–102.
25. Byers, C.C. Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled IoT networks. *IEEE Commun. Mag.* **2017**, *55*, 14–20. [\[CrossRef\]](#)
26. Zahmatkesh, H.; Al-Turjman, F. Fog computing for sustainable smart cities in the IoT era: Caching techniques and enabling technologies—An overview. *Sustain. Cities Soc.* **2020**, *59*, 102139. [\[CrossRef\]](#)
27. El-Hasnony, I.M.; Mostafa, R.R.; Elhoseny, M.; Barakat, S. Leveraging mist and fog for big data analytics in IoT environment. *Trans. Emerg. Telecommun. Technol.* **2020**, *32*, e4057. [\[CrossRef\]](#)
28. Sood, S.K.; Kaur, A.; Sood, V. Energy efficient IoT-Fog based architectural paradigm for prevention of Dengue fever infection. *J. Parallel Distrib. Comput.* **2021**, *150*, 46–59. [\[CrossRef\]](#)
29. Lei, W.; Zhang, D.; Ye, Y.; Lu, C. Joint beam training and data transmission control for mmwave delay-sensitive communications: A parallel reinforcement learning approach. *IEEE J. Sel. Top. Signal Process.* **2022**, *16*, 447–459. [\[CrossRef\]](#)
30. Li, J.; Li, X.; Yuan, J.; Zhang, R.; Fang, B. Fog computing-assisted trustworthy forwarding scheme in mobile Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 2778–2796. [\[CrossRef\]](#)
31. Ghosh, S.; Mukherjee, A.; Ghosh, S.K.; Buyya, R. Mobi-iost: Mobility-aware cloud-fog-edge-IoT collaborative framework for time-critical applications. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 2271–2285. [\[CrossRef\]](#)
32. Manogaran, G.; Rawal, B.S. An Efficient Resource Allocation Scheme with Optimal Node Placement in IoT-Fog-Cloud Architecture. *IEEE Sens. J.* **2021**, *21*, 25106–25113. [\[CrossRef\]](#)
33. Wang, Y.; Ren, Z.; Zhang, H.; Hou, X.; Xiao, Y. “combat cloud-fog” network architecture for internet of battlefield things and load balancing technology. In Proceedings of the IEEE International Conference on Smart Internet of Things (SmartIoT), Xi’an, China, 17–19 August 2018.
34. Xia, B.; Kong, F.; Zhou, J.; Tang, X.; Gong, H. A delay-tolerant data transmission scheme for internet of vehicles based on software defined cloud-fog networks. *IEEE Access* **2020**, *8*, 65911–65922. [\[CrossRef\]](#)
35. Amiri, I.S.; Prakash, J.; Balasaraswathi, M.; Sivasankaran, V.; Sundararajan, T.V.P.; Hindia, M.; Tilwari, V.; Dimyati, K.; Henry, O. DABPR: A large-scale internet of things-based data aggregation back pressure routing for disaster management. *Wirel. Netw.* **2020**, *26*, 2353–2374. [\[CrossRef\]](#)
36. Awaisi, K.S.; Hussain, S.; Ahmed, M.; Khan, A.A. Leveraging IoT and Fog Computing in Healthcare Systems. *IEEE Internet Things Mag.* **2020**, *3*, 52–56. [\[CrossRef\]](#)
37. Chinnasamy, P.; Deepalakshmi, P.; Dutta, A.K.; You, J.; Joshi, G.P. Ciphertext-Policy Attribute-Based Encryption for Cloud Storage: Toward Data Privacy and Authentication in AI-Enabled IoT System. *Mathematics* **2021**, *10*, 68. [\[CrossRef\]](#)
38. Mancini, R.; Tuli, S.; Cucinotta, T.; Buyya, R. *iGateLink: A Gateway Library for Linking IoT, Edge, Fog, and Cloud Computing Environments*; Springer: Singapore, 2021; pp. 11–19.
39. Chinnasamy, P.; Rojaramani, D.; Praveena, V.; SV, A.J.; Bensujin, B. Data Security and Privacy Requirements in Edge Computing: A Systemic Review. In *Cases on Edge Computing and Analytics*; IGI Global: Hershey, PA, USA, 2021; pp. 171–187.
40. Dastjerdi, A.V.; Buyya, R. Fog computing: Helping the Internet of Things realize its potential. *Computer* **2016**, *49*, 112–116. [\[CrossRef\]](#)
41. Sarkar, S.; Misra, S. Theoretical modelling of fog computing: A green computing paradigm to support IoT applications. *IET Netw.* **2016**, *5*, 23–29. [\[CrossRef\]](#)
42. An, J.G.; Li, W.; Le-Gall, F.; Kovac, E.; Kim, J.; Taleb, T.; Song, J. EiF: Toward an elastic IoT fog framework for AI services. *IEEE Commun. Mag.* **2019**, *57*, 28–33. [\[CrossRef\]](#)
43. Loffi, L.; Westphall, C.M.; Grüttner, L.D.; Westphall, C.B. Mutual authentication with multi-factor in IoT-Fog-Cloud environment. *J. Netw. Comput. Appl.* **2021**, *176*, 102932. [\[CrossRef\]](#)

44. Mubarakali, A.; Durai, A.D.; Alshehri, M.; AlFarraj, O.; Ramakrishnan, J.; Mavaluru, D. Fog-based delay-sensitive data transmission algorithm for data forwarding and storage in cloud environment for multimedia applications. In *Big Data*; Mary Ann Liebert, Inc.: New Rochelle, NY, USA, 2020.
45. Li, J.; Cai, J.; Khan, F.; Rehman, A.U.; Balasubramaniam, V.; Sun, J.; Venu, P. A secured framework for sdn-based edge computing in IOT-enabled healthcare system. *IEEE Access* **2020**, *8*, 135479–135490. [[CrossRef](#)]
46. Mahmud, M.; Kaiser, M.S.; Hussain, A.; Vassanelli, S. Applications of deep learning and reinforcement learning to biological data. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2063–2079. [[CrossRef](#)]
47. Singh, K.D.; Sood, S.K. 5G ready optical fog-assisted cyber-physical system for IoT applications. *IET Cyber-Phys. Syst. Theory Appl.* **2020**, *5*, 137–144. [[CrossRef](#)]
48. Balasubramanian, V.; Aloqaily, M.; Resslein, M. An SDN architecture for time sensitive industrial IoT. *Comput. Netw.* **2021**, *186*, 107739. [[CrossRef](#)]
49. Thangaramya, K.; Kulothungan, K.; Logambigai, R.; Selvi, M.; Ganapathy, S.; Kannan, A. Energy aware cluster and neuro-fuzzy based routing algorithm for wireless sensor networks in IoT. *Comput. Netw.* **2019**, *151*, 211–223. [[CrossRef](#)]