Received June 3, 2018, accepted July 1, 2018, date of publication July 5, 2018, date of current version July 30, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2853134

Orchestrating Data as a Services-Based Computing and Communication Model for Information-Centric Internet of Things

ZHENGYAN DING[1], KAORU OTA[2], YUXIN LIU[1], NING ZHANG[3], (Student Member, IEEE), MING ZHAO[4], HOUBING SONG[5], (Senior Member, IEEE), ANFENG LIU[1,6], AND ZHIPING CAI[7], (Member, IEEE)

[1]School of Information Science and Engineering, Central South University, Changsha 410083, China
[2]Department of Information and Electronic Engineering, Muroran Institute of Technology, Muroran 050-8585, Japan
[3]Department of Computing Sciences, Texas A&M University, Corpus Christi, TX 78412, USA
[4]School of Software, Central South University, Changsha 410012, China
[5]Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA
[6]The State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China
[7]Department of Network Engineering, School of Computer, National University of Defense Technology, Changsha 410073, China

Corresponding author: Yuxin Liu (yuxinliu@csu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772554, Grant 61572528, and Grant 6157256, in part by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China, under Grant ICT1800391, in part by the National Basic Research Program of China (973 Program) under Grant 2014CB046345, in part by JSPS KAKENHI under Grant JP16K00117 and Grant JP15K15976, and in part by the KDDI Foundation.

ABSTRACT With the explosive growth of the Internet of Things, the gap between the rapidly growing demands of data rates and the existing bandwidth-limited network infrastructures has become increasingly prominent, leading to network congestion, high latency, and energy consumption and deterioration of the user's quality of service. To narrow this gap, an orchestrating data as services-based computing and communication (ODAS-CC) model is proposed to reduce the data rate, latency, and energy consumption for information-centric Internet of Things. The main innovations of the ODAS-CC model that differ from previous strategies are as follows. First, the services-based network architecture proposed in this paper can run on the current network effectively. In the proposed architecture, the data are orchestrated to services when they route to the data center; therefore, the data are transmitted after the service conversion, thereby forming a service-based network, which can greatly reduce the amount of data transmitted in the network, latency, and energy consumption. Second, a service conversion is performed in the process of routing such that the user's service request can be satisfied locally, thereby improving the user's quality of experience. Third, based on the ODAS-CC model, the performance evaluation model for energy consumption, latency, and load is provided in detail, and the performance of the ODAS-CC model is evaluated comprehensively. The theoretical analyses and experimental results show that compared with the previous approach, the ODAS-CC model can reduce the network traffic, the average latency of data upload by 26.3%, the service latency by up to 30.1%, and the data center load, thereby reducing energy consumption from 20.0% to 30.0%.

INDEX TERMS Orchestrating data as service (ODAS), quality of service, energy consumption, load, latency.

I. INTRODUCTION

The internet of things (IoT) [1]–[4], which leverages the ubiquity of smart sensor-equipped devices [5]–[7] to collect data at low cost and supports ubiquitous information exchange and content sharing [8]–[10] is a key enabler for various applications, such as smart cities [11], smart grids [12], [13], smart health [14], [15], and intelligent transportation systems [16], [17] [18]–[21]. With the explosive growth of the IoT, it is estimated that almost 50 billion devices will be interconnected by 2020 [8], [22]. A report from Cisco shows that the data throughput generated by the Internet of Things has already accounted for 69% of the total Internet throughput in 2014, which is 30 times that of the data throughput in 2000 [1], [22]. Moreover, the increment is still accelerating. With the explosive growth of the number of IoT devices and the amount of data they produce [23], [24], the gap between the rapidly growing demands of the data rate and the existing bandwidth-limited network infrastructures has become ever prominent [1], [21], [22]. Obviously, this tremendous increase in devices and data traffic has brought unprecedented challenges to the current data-transmission-based network [1], [21], [22], [25], [26]. Its potentially huge development prospects, have increasingly attracted studies and investments, making it faster and more challenging. However, the current network is based on the store and forwarding routing rule [27], so the scale of data volume will not change. Compared to the increase in the data rate demands, the growth of the transmission capacity of the physical line of network is much slower, which leads to the gap between the exponential growth of network data traffic and the limited transmission capacity of the network [1], [21], [22]. Without large-scale technological advances, it can be foreseen that the volume of the current network will soon be exhausted, the network congestion and the latency will sharply increase, and the Quality of Services (QoS) will deteriorate dramatically, which makes the Quality of Experience (QoE) poor. As a result, the network will face serious challenges in the future [28]–[30].

As a novel communication model, information-center networking (ICN) is different from the traditional IP address-centric model [1], [21], [22]. In ICN, there have been some studies that have developed content-based data transmission strategies and methods, which can explore the inherent nature of the large number of devices, such as caching, communications, computing capabilities. Therefore, with these strategies and methods, an efficient and smart information-centric IoT (IC-IoT) paradigm with great significance can be built. Additionally, content-based data routing strategies can improve the network content delivery [8]–[10]. For example, caching hot data is used (especially in multimedia streaming) to speed up data access and to reduce latency and data duplication [9], [10], [21], [26], [31]. However, a major drawback of the current method is that no matter what type of data transmission scheme is adopted, the raw packets transmitted by the network cannot be refined. Although these methods can help improve the network performance to a certain extent, they do not reduce the pressure on the network caused by the rapid growth of data.

In the context of IoT, we have observed that, if the data collected by smart devices is aggregated or converted to services whose information is reserved, then the amount of data that needs to be transmitted can be greatly reduced [8], [21], [22]. For example, the VTrack project [1], [22] provides timely traffic information, and the NoiseTube project provides urban noise distribution services [1], [22]. In such applications, when the task of collecting data is released, a large number of people with mobile phones can sense the data through the sensing device on their mobile phones, and then report data to VTrack or NoiseTube, which can be considered as the data center. However, there is a correlation between the data and similar collection positions. Therefore, the data with correlation can be aggregated by an aggregation micro-program, which we called AMP, when they meet on the route to the data center, which will reduce the amount of data that needs to be transmitted. In addition, this type of method has a wide range of application scenarios. It can be applied not only to the scientific data collection but also to social network data collection [16], [17], [32]. For example, when large-scale events (such as large-scale assemblies, concerts, large-scale competition events) occur, various smartphones in the area will collect images, videos, and sounds, and then upload them to the news center. There can be 100 or even 1000 of these devices used for on-site data collection under the same incident scene; therefore, there is considerable redundancy between the uploaded data [33]–[35]. In such a scene, every packet will be transmitted to the news center under the current traditional IP address-centric network, which places great pressure on the network. In fact, if the relevant data is aggregated and then uploaded, the network load can be reduced by 10 times or even 100 times.

Another method to effectively reduce the amount of data transmitted in the network is the service conversion approach [8], [21], [22]. This approach benefits from the current rise of the Software Defined Network (SDN) [36], [37]. In the SDN, the micro-programs for converting services, which we called service micro-programs (SMP), can be released to various network devices by the data center. Therefore, in addition to the data center, each network device can also convert raw data into a service, thereby implementing a network model centered on service computing to replace the current network model centered on the data transmission [8], [21], [22], [36], [37]. With this method, the amount of data that the network needs to transmit is greatly reduced, thereby bridging the gap between the rapidly growing demands of data rates and existing bandwidth-limited network infrastructures. Especially in inclement weather or hazy conditions, with this method, the data packets collected in a certain area may be 1000, but after the calculations provided by the meteorological department, the result may only have the capacity for 5 data packets, thereby greatly reducing the data transmission capacity.

Compared to data aggregation methods, service conversion technology can reduce data transmissions to a greater extent, but it often has higher requirements for storage space in devices because different services correspond to different SMPs. In the ODAS-CC model, when the data sensed by the network edge pass the network devices on the path to the data center, the data is aggregated (if it cannot be converted to services) or converted to a service and then forwarded, thereby reducing the network traffic and reducing the amount of data received by the data center. Additionally, if the service which is converted by the current device or has been cached already can satisfy the request of the user, it is immediately returned to the user. Otherwise, the request is forwarded to the upper-layer devices. The upper layer device has a more comprehensive service which can meet more user requests, thereby reducing the request flow, service flow, service latency and the load of the data center [21].

Because of the advantages of service-based networks, some researchers have made some initial investigations [21], [22]. In this type of service-based network, the amount of data which is routed to the data center from the data collectors can be reduced. In addition, the services can be converted during the routing, which makes the request service hit rates higher, thereby reducing the network load and optimizing the QoS of the user. However, the data center needs to distribute micro-programs (AMP and SMP) to the devices in the network. At the same time, these micro-programs need to be updated. As a result, it also brings new challenges to the network [36], [37]. To the best of our knowledge, there are no studies that provide comprehensive orchestrating data as services-based on the computing and communication model, and there are no detailed studies on ODAS-CC model performance, such as network traffic, latency and energy consumption. However, the establishment of a reasonable ODSA-CC model and its performance optimization and analysis conclusions are of great significance to the information-centric Internet of Things (IC-IoT). Therefore, an ODAS-CC model is proposed in this paper. Under this model, for the data center, the service micro-program (SMP) publishing scheme is given. For the network devices, the packet schedule scheme is given. In addition, the performance of the ODAS-CC model is evaluated comprehensively. The main innovations of this paper are as follows:

(1) The service micro-program (SMP) publishing and updating scheme given in this paper has not been studied in the previous research. Due to the limited capacity of the storage space, the network devices cannot store all types of SMP, so an effective SMP publishing and updating scheme is proposed to minimize the network traffic for a services-based network. In this paper, the DC dynamically publish and update appropriate SMP according to the type of data. The devices receive the data and convert the data as much as possible into a service before transmitting, so the network traffic can be greatly reduced.

(2) A packet forwarding schedule scheme is proposed, and different forwarding modes are adopted for different packets, including the data-upload packet, service-request packet, service-return packet, and micro-program packet. With this forwarding schedule scheme, the latency is lower, the load is more balanced, and the micro-program packet disseminates faster.

(3) The ODAS-CC model is given, and its performance analysis is performed comprehensively. The energy consumption calculation method for the ODAS-CC model is given in detail. Finally, we compare our proposed ODAS-CC model with the existing data-transmission-based model, which we called DTBN, via extensive simulations. The experimental results show that, compared to the DTBN, the ODAS-CC model can reduce the network traffic, reduce the average latency of data upload by 26.3%, reduce the service latency by up to 30.1%, and reduce the data center load. Therefore, the energy consumption is reduced by 20.0% to 30.0%.

The rest of this paper is organized as follows. In Section 2, the related works are reviewed. The network model and problem statements are described in Section 3. Section 4 elaborates on the design of ODAS-CC model. The performance analysis and comparisons are provided in Section 5. Finally, we present the study's conclusions in Section 6.

II. BACKGROUND AND RELATED WORK

A. THE REASONS FOR THE BIG GAP BETWEEN THE DEMANDS OF THE DATA RATE AND THE EXISTING BANDWIDTH-LIMITED NETWORK

The important reasons why the gap between the demands of data rate and the existing bandwidth-limited network infrastructures is increasing lies in the following important aspects:

(1) The number of smart devices, such as smartphones and wireless sensor networks, are increasing exponentially [38]–[42]. It is estimated that almost 50 billion devices will be connected to IoT by 2020 [1], [21], [22]. The increase in the number of devices naturally leads to the need for the rapid growth of the data rate.

(2) The capabilities of current smart devices, such as calculation, storage and communication are far beyond the personal computer 20 years ago [43], [44]. Additionally, with the development of wireless networks, the devices equipped with small sensors can collect data anytime, anywhere [45], [46]. For example, smartphones have high-precision camera devices that can be used in real-time to obtain multimedia with unimaginable capacity, such as video, images, and sound. Therefore, data acquisition has become more convenient and timely, resulting in the amount of data rising exponentially.

(3) Big data and related applications have been rapidly developed [12], [22], [45], which has led to a sharp increase in the demand for data rates and has brought greater pressure on current networks. With the development of smart devices, various new applications are spawned. In these applications, most of them rely on the acquisition of vast amounts of data. In many data acquisition methods, participation perception is one of the most promising methods [5]. In such a data acquisition method, the data requester publishes some requirements for the data, such as time, spatial scope, and gives a reward for collecting the data. Then, the huge number of sensor devices (or its owners) voluntarily participate in the data collection [5], [38], [41]. Due to the large number of widely distributed sensing devices, it is necessary to acquire data at a very low price for a long time and in a large space. Thus, the traditional method of data acquisition should be completely changed. For example, there is a study about a certain migratory bird. For this research, a long period of time is needed to study its migration route, time, population size, habits and laws. If relying on the observation of researchers solely, a lot of manpower and material resources are needed, but the observed data is still very limited. However, participatory

perception methods are fully applicable. Researchers only need to publish tasks for the birds that need to be observed. Then, a large number of people with smartphones will report their observation if they see these birds. The report can be received in various forms: text, sound, picture, video, etc. In this way, the information collected on the birds is more persistent in time, more detailed in the content of observations, and can be considered as having no boundaries in the scope of observations [5]. Moreover, it is very economical. Each piece of data obtained in this way is reduced by approximately 100 times the cost of data obtained by conventional methods.

Above all, the current network is undergoing profound changes, and the gap between the demands of the data rate and existing bandwidth-limited network infrastructures is expanding. Therefore, it is especially urgent to find a suitable solution. Researchers have realized this problem and have proposed some solutions, which are summarized as follows.

B. IMPROVE THE NETWORK ARCHITECTURE TO NARROW THE GAP

The current network is developed from an IP address-centric network. Because of the low transmission speed, the development of data-oriented applications in the network will be limited. However, the network bandwidth has been greatly developed in the past 10 years, which has led to the development of some applications which require vast data. To make full use of computing and storage resources in the network, some network architectures have been proposed, such as grid computing, the cloud computing, etc [8], [15], [21], [24], [49]. In this type of network architecture, the data calculation is placed at the core of the network, which can be considered as a virtual data center. After the data is processed by the data center, it provides various services to the outside world. In the case of small scale of peripheral devices (such as smart devices), grid computing as well the cloud computing are natural choices.

However, with the rapid development of IoT, this type of network architecture faces severe challenges. First, the number of smart devices connected to the Internet is extremely large, so the capabilities of computing and storage are no longer located at the core of the network but are distributed to a huge number of smart devices. If the center calculation method is still used, a large portion of the computing, storage, and communication capabilities of edge networks is wasted [3], [9]. Second, a large amount of data is generated at the edge of the network, and these data need to be transmitted to the network center through a long routing path, so it will deteriorate the network performance. On the other hand, the service generated by the data center is provided to many users (mostly via their smart devices) who are located at the edge of the network, which puts tremendous pressure on the network and causes the QoS of the user to decline [21], [22].

Researchers are aware of the above problems and have successively proposed some models, such as fog computing and edge computing, to solve the problems. The main idea of these models is decentralization. In other words, it takes advantage of the rich smart devices at the edge network to complete the calculation, thereby reducing the load on the data center. On the other hand, there are many studies that have proposed offload strategies and methods [3], [9]. However, these methods for reducing the load are not easy tasks. The evolution of the network architecture based on the progress of special equipment in the network is a gradual process. Although optimizing the network architecture can improve the network performance from the structure, there is not much research about how to reduce the amount of data that needs to be transmitted. Therefore, the solutions are still under exploration, such as decentralization and the scheduling of distributed tasks.

C. REDUCE THE DEMAND OF BANDWIDTH BASED ON CACHE TECHNOLOGY

In the current network, a large number of data collectors are located at the edge of the network, and the data center for processing data is often located at the center of the network. Therefore, when a user requires data, the request needs to be routed through a long path to the destination, and the requested data needs to be returned to the user over the same long path. In this way, the latency of the user is high. At the same time, the process of data uploading to the data center, as well as the user's request and result return all put a heavy load on the network.

Therefore, cache technology is proposed, which plays an active role in reducing network bandwidth [9], [10], [21], [26], [31]. This technique takes advantage of the temporal and spatial correlation of data access. If a user accesses the data for a certain period, the probability of accessing these data by another user is also high during the following period. Specifically, after certain data of the data center is accessed by the user, these data will be distributed by the data center to some key nodes in the network for caching. In this way, when another user accesses this data, it can be satisfied (caching hit) at the middle of the key nodes, so the amount of data distributed by the data center and the network bandwidth requirements are reduced [21], [31].

This type of technology can improve the network performance effectively with no requirement on the network architecture. However, there are some limitations:

(1) Caching technology has certain requirements for the storage capacity of the hardware. The greater the storage capacity, the better the caching technology performance. Therefore, this technology can play a good role in the network device with a certain amount of storage capacity. However, with the limited storage capacity, the performance is limited.

(2) Caching technology only relieves the network load to a certain extent. It is an indirect way to reduce the data rate. However, this technology does not reduce the size of the data; it only speeds up the user's access through storing data in network devices. The technology's efficiency often depends on the attributes of the accessed data. Different data attributes have completely different effects. For example,

caching technology works better in multimedia data. When a user is watching a movie or TV program, the data center can predict the data required by the user in advance so that it can be cached in advance, and it can cache based on the fiery degree of multimedia data, thereby improving network performance and getting better QoE. However, for a large amount of sporadic data, its performance is not good because of its limitations of the device's storage capacity.

(3) Caching technology currently only considers the downlink data from data center to the user. However, the caching problem of uploading data from the network periphery to the data center has seldom been studied. In the current context of IoT, the uplink data has already become the most important part of the network [21].

In summary, although caching is a method to effectively reduce the data rate, it cannot be regarded as a major method, but as an auxiliary technology.

D. REDUCE THE BANDWIDTH DEMAND BASED ON DATA AGGREGATION TECHNOLOGY

Data aggregation is a method that is widely used in IoT to effectively reduce the amount of data transmission [8]. It is based on the correlation between data sensed by various devices. When multiple correlated data encounters occur in routing, data aggregation can be performed to merge multiple packets into a much smaller packet, which can greatly reduce the amount of data that needs to be transmitted [8].

In the wireless sensor network, this method has become one of the standard data operations. Since the density of nodes deployed in the wireless sensor network is very large, there is a correlation between the data collected by nodes with similar times and locations. Data aggregation can be performed to reduce the amount of data. In some applications, the rate of data aggregation can reach a very high level. For example, in the application of a smart farm, the data about the average, maximum, and minimum temperature (humidity) is often required. In this way, with data aggregation technology, an infinite number of packets can be combined into one packet, which greatly reduces the amount of data [8].

However, the research on data aggregation applied to the Internet is limited. On the Internet, the chances of encountering related packets are low, and it is not easy to control. Xu *et al.* [8] proposed a data aggregation scheme that can be applied to the Internet, which is named the Intelligent Aggregation based on a Content Routing (IACR) scheme for cloud computing. In the IACR scheme, the route path is labeled according to the attribute number of the routing data. The data is routed through the same routing path as the data attribute. As a result, the probability of data aggregation can be increased because the correlation between data with the same attributes is larger, thereby improving the network performance. In addition, security also has good performance [47], [48].

E. REDUCE THE BANDWIDTH DEMAND BASED ON SERVICE CONVERSION TECHNOLOGY

The service conversion technology used to reduce the amount of data is based on the following ideas [21], [22]: The raw data is processed by SMP into a service, which can provide data to the user. In general, the size of the service is far less than the size of the original packet. For example, in meteorological services, the collected data is processed by the meteorological program to obtain the meteorological results. In such applications, the amount of data that needs to be collected may be tens of thousands if accurate results are to be obtained. However, the result may be less than a packet volume. As a result, the amount of data is reduced by thousands of times. Therefore, compared to data aggregation technology, this method is more effective. Additionally, it is not affected by the correlation between data. However, in this method, different services correspond to different SMPs, the devices need more spaces to store SMPs, which are provided by the application providers. Limited by the storage capacity of network devices, the network traffic will be increased by frequent SMPs updating. We proposed a service-based network architecture in [22]. However, we have not considered the various performances in this network architecture carefully and have not given various indicators, such as the energy consumption. In general, this approach can fundamentally change the pressure of current data explosion based on an IP address-centric network. In particular, with the rise of Software Define Network (SDN) technology, the service conversion in network devices becomes possible [49]. Apparently, for this emerging technology, there are still many places that need to be studied. Therefore, this article is a further in-depth study of service conversion.

III. THE SYSTEM MODEL AND PROBLEM STATEMENT

A. THE NETWORK MODEL

The network model in this paper is shown in FIGURE 1 which is the same as [22]. Its main body is composed of the following parts:

(1) Data Collection Layer. This layer is mainly composed of various types of sensors, such as weather sensors, smart phones, etc. These sensors transmit the collected data layer by layer to the data center, and the data center provides corresponding services based on these collected data.

(2) Fog Computing Layer. The functions of the device in this layer could be divided into 3 parts: a) Data can be forwarded and cached; b) SMP published from the data center can be received and run; c) Data can be aggregated or converted to service by running SMP or AMP, respectively.

(3) Core Network Layer. The functions of the router in this layer are the same as that of the devices in the fog computing layer. Therefore, in the subsequent design, the devices in both the fog network layer and the core network layer are collectively referred to as Intelligent Nodes.

(4) Data Center Layer. The data center can store and process the data and can also publish SMP. It is the center of application. In our paper, only one data center is considered.

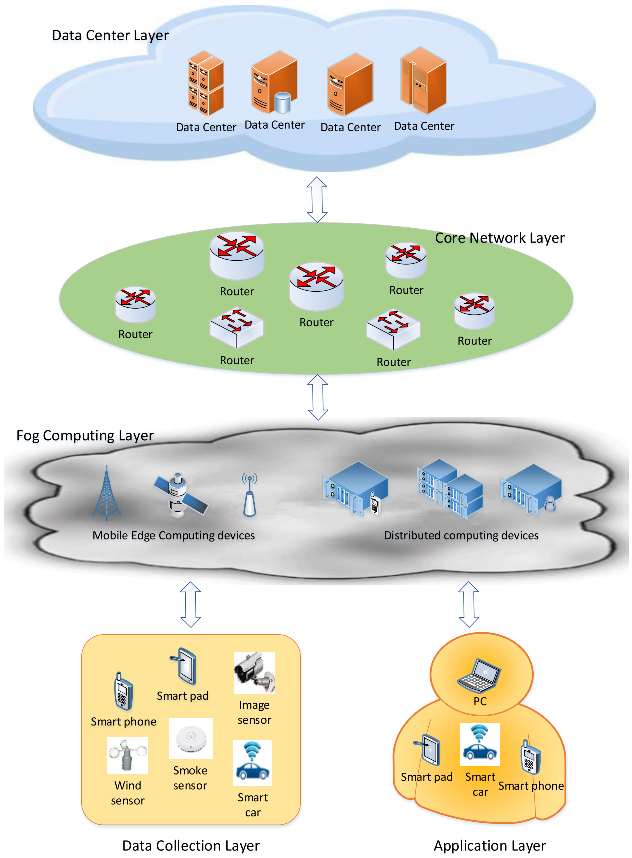


FIGURE 1. The structure of ODAS-CC.

(5) Application Layer. In this layer, the devices used by users are able to send service requests and get the required services. Additionally, it may also act as a data collector. There is no obvious boundary between this layer and the data collection layer.

B. SERVICE CONVERSION MODEL

The service conversion model is similar to [8]. Assume that the data already cached by the Intelligent Node is A_{old}, the newly arrived data is A_{new}, the function of $r_c(x, y)$ is that the x and y are converted to service, then the service A_c is as follows:

$$A_c = r_c(A_{old}, A_{new}) \tag{1}$$

For the data where the type is k, assuming that the amount of these data is Q_k, the amount of these data after service conversion is Q_k^s, the amount of these data after data aggregation is Q_k^d. Therefore, the degree of service conversion (r_k^s) and the degree of data aggregation (r_k^d) are shown as follows:

$$\begin{cases} r_k^s = \dfrac{Q_k^s}{Q_k} \\ r_k^d = \dfrac{Q_k^d}{Q_k} \end{cases} \tag{2}$$

According to formula (2), the degree of service conversion mainly depends on the SMP. As a result, different SMPs have different degrees of service conversion. However, for a certain type of data, the degree of service conversion is almost fixed.

C. ENERGY CONSUMPTION MODEL

According to the description of the network model, energy consumption is mainly divided into four parts: energy consumption of various types of sensors, energy consumption of Intelligent Nodes, energy consumption of data centers, and energy consumption of application equipment. To simplify the model of the energy consumption, the energy consumption of various sensors and application devices is not considered. Assume that the energy consumption of the Intelligent Node i is E_i^{node}, the energy consumption of the data center is E_c, the number of Intelligent Nodes is n, the mathematical expression of the total energy consumption E is as follows:

$$E = \sum\nolimits_{i=1}^{n} E_i^{node} + E_c \tag{3}$$

For the Intelligent Node, the energy consumption is mainly reflected in three aspects: calculation, cache, and forwarding. Therefore, if the energy consumption of the Intelligent Node i for calculation, cache, and forwarding are E_i^{cal}, E_i^{cac}, E_i^{fwd}, respectively, then the energy consumption of the Intelligent Node i is as follows:

$$E_i^{node} = E_i^{cal} + E_i^{cac} + E_i^{fwd} \tag{4}$$

According to the description of the energy formula of the fog computing device in [1], E_i^{cal}, E_i^{cac} and E_i^{fwd} are positively related to the amount of data, which means the larger the amount of data, the greater the energy consumption. Assume that the data amounts that needs to be calculated, cached, and forwarded are Q_i^{cal}, Q_i^{cac}, Q_i^{fwd}, respectively, and suppose that the energy consumption of per unit of data after calculation, cache, forwarding are θ_c, θ_s, θ_f, respectively. Next, the mathematical expression is shown as follows:

$$\begin{cases} E_i^{cal} = \theta_c \times Q_i^{cal} \\ E_i^{cac} = \theta_s \times Q_i^{cac} \\ E_i^{fwd} = \theta_f \times Q_i^{fwd} \end{cases} \tag{5}$$

However, for the data center, according to the description of the data center energy formula in [1], energy consumption is mainly reflected in four aspects: calculation, storage, forwarding, and data migration. Because of the complexity of the data center, it is difficult to consume the energy from these four aspects. Therefore, the energy consumption of the data center is estimated by the maximum power and load of the data center. Assuming that the maximum power of the data center is P_{max}, the load percentage is μ, the running time under this load percentage is t and η is the energy efficiency ratio, then the mathematical expression of E_c is as follows:

$$E_c = \eta \times \mu \times P_{max} \times t \tag{6}$$

Assume that the maximum number of data packets or service requests that the data center can handle per second is p_{max}, and the number of data packets or the number of service requests that need to be processed per second is p_t, then the mathematical expression of μ is as follows:

$$\mu = \frac{p_t}{p_{max}} \tag{7}$$

Formula (8) is obtained by substituting formula (7) into formula (6).

$$E_c = \eta \times \frac{p_t}{p_{max}} \times P_{max} \times t \tag{8}$$

D. PROBLEM STATEMENT

Based on the above analysis, the ODAS-CC model is implemented mainly for the following purposes:

(1) To minimize the number of packets transmitted in the network:

Assume that the number of packets forwarded by the Intelligent Node i is pk_i. If the number of packets forwarded by each Intelligent Node is the smallest, the number of packets transmitted in entire network is the smallest. However, it is difficult to minimize the number of packets forwarded by each Intelligent Node. Therefore, it is more realistic to use the sum of all packets transmitted in the entire network as the target. The equation (9) is shown as follows:

$$pk_{min} = min(\sum_{i=1}^{n} pk_i) \tag{9}$$

(2) To minimize the average latency of service:

The latency of service mainly consists of three parts: the time consumed by the Intelligent Node to process, the time consumed during the link transmission, and the time consumed by the data center to respond to this request. However, the time consumed by the Intelligent Node to process the service is small compared to the time consumed during the link transmission. Therefore, it can be considered that the latency of service mainly depends on the time consumed during the link transmission. If the service cannot be satisfied by all the Intelligent Nodes on the path to the data center, then the data center will respond to service requests. In this case, the time of the data center response service should be added.

Assuming that the function of $dis(x, y)$ is to calculate the distance from x to y, then the distance from the starting point of the requesting service i to the ending point of meeting this service, can be expressed as $dis(node_i^s, node_i^e)$. The transmission rate of the link is c, then the latency of service i can be expressed as follows:

$$t_i^{sev} = \begin{cases} \dfrac{dis(node_i^s, node_i^e)}{c} + \dfrac{dis\left(node_i^e, node_i^s\right)}{c}, & if\ node_i^e \neq DC \\ \dfrac{dis(node_i^s, node_i^e)}{c} + \dfrac{dis(node_i^e, node_i^s)}{c} + t_{dc} \end{cases} \tag{10}$$

Since it is difficult to minimize the latency of each service request, the average latency of all services is taken as our goal. Assuming that n_{req} is the number of service requests, then the equation (11) is shown as follows:

$$t_{min}^s = min(\frac{1}{n_{req}} \times \sum_{i=1}^{n_{req}} t_i^{sev}) \tag{11}$$

(3) To minimize the average latency of the data upload:

Regarding the data produced by the sensors of the data collection layer, it needs to upload to the data center node by node. The latency of the data upload mainly consists of the time consumed by the Intelligent Node to process and the time consumed during the link transmission. However, the time consumed by the Intelligent Node to process is small compared to the time consumed during the link transmission. Therefore, the time for uploading data to the data center depends on the time consumed by the link transmission. Suppose that $node_i^{up}$ is the starting point of the data i to upload, n_{up} is the number of data-upload packets, and then the minimized average latency of data upload is shown as follows (12):

$$t_{min}^{up} = min(\frac{1}{n_{up}} \times \sum\nolimits_{i=1}^{n_{up}} \frac{dis(node_i^{up}, DC)}{c}) \tag{12}$$

(4) To minimize the energy consumption:

According to the energy consumption model, to minimize the energy consumption, it is necessary to minimize the energy consumed by each Intelligent Node and data center (only one data center is considered in our paper). However, it is difficult to minimize the energy consumption of each Intelligent Node, so the sum of the energy consumption of all Intelligent Nodes is taken as the goal, which can be expressed as follows (13):

$$\begin{aligned} E_{min} = min(&\sum_{i=1}^{n} \theta_c \times Q_i^{cal} + \theta_s \times Q_i^{cac} \\ &+ \theta_f \times Q_i^{fwd}) + min(\mu) \times \eta \times P_{max} \times t \end{aligned} \tag{13}$$

Overall, the goal we want to achieve is shown as follows:

$$\begin{cases} pk_{min} = min(\sum_{i=1}^{n} pk_i) \\ t_{min}^s = \min\left(\frac{1}{n_{req}} \times \sum\nolimits_{i=1}^{n_{req}} t_i^{sev}\right) \\ t_{min}^{up} = min(\frac{1}{n_{up}} \times \sum\nolimits_{i=1}^{n_{up}} \dfrac{dis(node_i^{up}, DC)}{c}) \\ E_{min} = min\left(\sum_{i=1}^{n} \theta_c \times Q_i^{cal} + \theta_s \times Q_i^{cac} + \theta_f \times Q_i^{fwd}\right) \\ \qquad + min(\mu) \times \eta \times P_{max} \times t \end{cases} \tag{14}$$

TABLE 1. Parameter description in Chapter III.

Parameter	State
A_{old}	the data already cached
A_{new}	the newly arrived data
A_c	the data after service conversion
$r_c(x, y)$	the method to converted x and y to service
Q_k	the amount of raw data where the type is k
Q_k^s, Q_k^d	the amount of data where the type is k after service conversion, data aggregation, respectively
r_k^s, r_k^d	for the data where the type is k for the degree of service conversion, data aggregation, respectively
E_i^{node}	the energy consumption of the Intelligent Node i
E_c	the energy consumption of the data center
n	the number of the Intelligent Nodes
E_i^{cal}, E_i^{cac}, E_i^{fwd}	the energy consumption of the Intelligent Node i for calculation, cache, and forwarding, respectively
Q_i^{cal}, Q_i^{cac}, Q_i^{fwd}	the amount of data that needs to be calculated, cached, and forwarded, respectively
θ_c, θ_s, θ_f	the energy consumption per unit of data after calculation, cache, and forwarding, respectively
P_{max}	the maximum power of the data center
μ	the load percentage of the data center
η	the energy efficiency ratio of the data center
p_{max}	the maximum number of data packets or service requests that the data center can handle per second
p_t	the number of data packets or the number of service requests that need to be processed per second
pk_i	the number of packets forwarded by the Intelligent Node i
$node_i^s$	the starting point of the requesting service i
$node_i^e$	the ending point of the meeting service i
$node_i^{up}$	the starting point of the data i to upload
$dis(x, y)$	the method to calculate the distance from x to y
c	the transmission rate of the link
n_{req}	the number of service requests
n_{up}	the number of data-upload packets
t_i^{sev}	the latency of service i

To state the parameters of this chapter clearly, the main notions introduced in this chapter can be found in Table 1.

IV. THE DESIGN OF THE ODAS-CC MODEL

A. PACKET SCHEDULE SCHEME

The function of the Intelligent Node (IN) is explained in section III. In addition, in this part, the packet schedule scheme is designed specifically.

There are four types of packets in the network, the data-upload packet generated from various types of sensing devices, the service-request packet sent by users to request the service which users want, the service-return packet to respond to the service that users request and the program packet including AMP and SMP, which can be run by the Intelligent Node. The number of data types in the data-upload packet is the same as the number of corresponding SMPs. In the following, how these packets are dealt with is expounded.

(1) When a data-upload packet is received, first whether the type of SMP corresponding to the type of data in this packet exists are detected, and then, if it exists and the data in the Intelligent Node is sufficient to convert to service, the service is generated; otherwise, the AMP is used to aggregate the data. Then, the service or aggregated data is cached and forwarded to the upper Intelligent Node. The function of judging whether such data is sufficient to generate services is provided by the SMP, and no specific consideration is made here.

(2) When a service-request packet is received, first whether there is a service that can satisfy this request in its own cache space is checked, and then, if the service exists, the service is returned; otherwise, the service request is forwarded to the upper Intelligent Node. Additionally, the data center (DC) can satisfy all service requests. In other words, if the service request is forwarded to the DC, this service request will be satisfied absolutely.

(3) When a service-return packet is received, the service brought by this packet is cached and then forwarded. This allows the Intelligent Node to have a wider and broader range of services, and the next time the request for this service can be satisfied directly at this Intelligent Node without forwarding the service request up.

(4) When a program packet is received, first whether there is the same program in the cache space is checked, and then if the same program exists, this program packet is discarded; otherwise, this program is cached and then the flooding method is used to forward this program packet, that is, all ports are broadcasted except the interface that received this program packet. Under this approach, the broadcast storms can be avoided, and the program can be received by all the Intelligent Nodes in the network as soon as possible.

In addition to the program packet, for other types of packets, the forwarding table, which is called FTable, can be used to forward. Every Intelligent Node has its own forwarding table (for the details of the forwarding table design, refer to Part B of chapter IV). The entry of the forwarding table is searched according to the destination address in the packet and the packet is forwarded according to the entry. Specifically, if the packet is a service-request packet or a service-return packet, and there are multiple entries in the forwarding table of the Intelligent Node that have the same destination address but different ports, then one port is randomly selected for forwarding. However, if the packet is a data-upload packet, they are forwarded from all these ports. Its additional details are presented by Algorithm 1.

Algorithm 1 Forwarding Scheme Using FTable

Initialize: Every Intelligent Node records its own FTable (more detail in chapter IV Part B) and the set of SMP types as $Type_{IN}$ [ms] where ms is the maximum number of SMPs cached in the Intelligent Node. Define the variable s to record the set of ports. The variable $space_p$ represents the space of SMP in the Intelligent Node.

When receiving packet p:
  Case packet p **of**
    Case 1: packet p is the upload packet
      /* The function of $find_p$(p.desAddr) is finding the set of port from FTable where desAddr = p.desAddr*/
      s = $find_p$(p.desAddr);
      Forward packet from all the port $\in$ s;
    Case 2: packet p is the service request or return packet
      s =$find_p$(p.desAddr);
      Forward packet from random port $\in$ s;
    Case 3: packet p is the service program packet
      If this program exists **then**
        Discard packet p;
      Else
        If $space_p$ or $Type_{IN}$ [ms] is full **then**
          /*p_{rp}: the type number of SMP needs to be replaced, which only exist in the service program packet. */
          //delete the type number of p_{rp} program
          $space_p$.delete(p_{rp});
          $Type_{IN}$ [ms] .delete(p_{rp});
        End If
        //store the service program brought by packet p
        $space_p$.add(p);
        /* put the type number of the service program in $Type_{IN}$ [ms]*/
        $Type_{IN}$ [ms] .add(p);
        Forward packets from all the ports;
      End If
  End Case

Combining FIGURE 2 with the forwarding scheme, three points are determined as follows:

(1) The data-upload packet from Area B is forwarded to the DC through two paths, which are B->D->F->G->DC and B->C->E->G->DC. Before the data-upload packet reaches the DC, it will definitely go through the Intelligent Node G. However, to prevent the Intelligent Node G from forwarding the same data-upload packet twice, resulting in a large amount of data redundancy in the network, each Intelligent

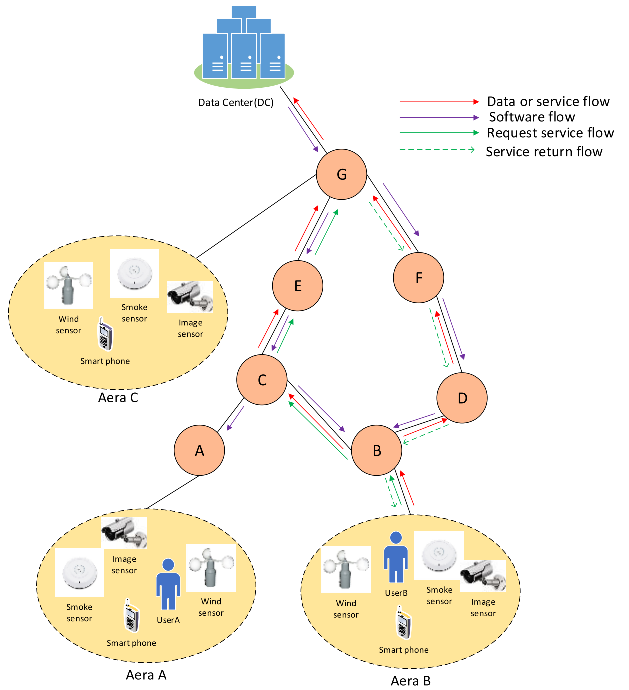


FIGURE 2. The running mechanism of ODAS-CC.

Node needs to perform data cleaning, that is, the redundant packets are directly discarded. In addition, in these paths to the DC, the shortest always exists, so the latency of upload is the lowest.

(2) When User A in Area A wants to request the service generated by sensors in Area B, this service can be satisfied by Intelligent Node C with a high probability according to the description of the forwarding scheme, thereby reducing the number of service requests to the DC. The purpose of this approach is that it allows all upper-level Intelligent Nodes to have a wider and broader range of services, to reduce the number of service requests to the data center, thereby reducing the latency of service.

(3) When User B in Area B wants to request the service generated by sensors in Area C, the service-request packet is forwarded through two paths randomly, one is B->C->E->G->DC and the other is B->D->F->G->DC. Similarly, if the request can be satisfied at the Intelligent Node G and there are two or more paths that can reach User B, a path is randomly selected for the service return. Under this scheme of making full use of multi-paths, the load of each Intelligent Node is balanced.

However, the storage space of the Intelligent Node is relatively small compared to the DC, so all data cannot be stored as a DC. Therefore, the least recently used (LRU) algorithm is used to manage the storage space. Additional details of the packet schedule scheme in the Intelligent Node are presented by Algorithm 2.

B. THE FORWARDING TABLE

To be able to forward data and requests to the destination address in accordance with the forwarding scheme

Algorithm 2 Packet Schedule Scheme

Initialize: Intelligent Node records Type_{IN} [ms]. Define the variable result to record the data produced by SMP or AMP. Define the variable $\text{service}_{\text{p}}$ to record the reply of service request. In addition, the variable space represents the cache space of the Intelligent Node.

```
When receiving packet p:
Use Algorithm3;
If packet p is a redundant packet then
    Discard packet p;
Else
    Case packet p of
     Case 1: packet p is a data-upload packet
        //p.typenum: the type number of data in packet p.
        If p.typenum ∈ Type_IN [ms] then
            /* start the service program where type number is p.typenum. In sp_p.typenum(p), if this type of data is not enough, start the aggregation program ap(p).*/
            result = sp_p.typenum(p);
        Else
            //start the aggregation program ap(p)
            result = ap(p);
        End If
        If space is full then
            Use LRU;
        End If
        //cache the result in the space
        space.add(result);
            Forward the data-upload packet using Algorithm1;
    Case 2: packet p is a service-request packet
        service_p= space.reply(p);
        /*service_p is not null, representing that the Intelligent Node can satisfy this request */
        If service_p is not NULL then
            Forward the service-return packet using Algorithm1;
        Else
            Forward the service-request packet using Algorithm1;
        End If
    Case 3: packet p is a service-return packet
        If space is full then
            Use LRU;
        End If
        //cache the service brought by packet p
        space.add(p);
        Forward the service return packet using Algorithm1;
    Case 4: packet p is a service program packet
        Forward the service program packet using Algorithm1;
    End Case
End If
```

requirements, the forwarding table (FTable) comes up. In the design of FTable, abnormal situations, such as a link failure, are not considered. The fields of the entries in the forwarding table (FTable) and their meanings are shown in Table 2.

TABLE 2. Description of the field name.

Field Name	State
DesAddr	The destination address for forwarding
Port	The interface for forwarding

The forwarding table (FTable) updates when the packet is received by the Intelligent Node. When receiving a packet, whether there is an entry where the source address of the packet received is the value of *DesAddr* and the interface receiving the packet is the value of *Port* in FTable is detected first. If not, the entry where the value of *DesAddr* equals the source address of the packet received and the value of *Port* equals the interface receiving the packet is added in FTable. The reason is that if a packet sent from a certain Intelligent Node called X1 enters a certain Intelligent Node called X2 from interface P, then the packet must be transmitted from P in the opposite direction to X1. Due to the limited storage space of the FTable, the least recently used (LRU) algorithm is used when the storage space is full. Additional details are shown in Algorithm 3.

Algorithm 3 Forwarding Table (FTable) Updating Approach

Initialize: The Intelligent Node records its own FTable, which consists of a series of entries including two fields, i.e., desAddr and port. Define the variable port_{in} to record the port which packet p enters.

```
When receiving packetp:
For each it ∈ FTable Do
    //p.srcAddr: the source address of packet p.
    If it.desAddr ≠ p.srcAddr or it.port ≠port_in then
        If FTable is full then
            Use LRU to delete an item from FTable;
        END If
        //Add new entry in the FTable.
        it_new.desAddr = p.srcAddr;
        it_new.port =port_in;
        FTable.add(it_new);
    End If
End For
```

C. SERVICE MICRO-PROGRAM PUBLISHING SCHEME

The data center (DC) in this paper has two major functions, one is to store and process data, and the other is to publish the program packet. However, in our paper, the latter is mainly designed.

Since the space in the Intelligent Node is limited, the total SMP cannot be stored. Therefore, to dynamically adapt to changes in the types of data brought by data-upload packets, thereby reducing the network traffic as much as possible, the DC must dynamically publish appropriate SMP for the type of data received. Therefore, a program publishing scheme is developed. That is, after receiving a certain number of packets in the DC, whether it is necessary to publish an SMP is detected according to the ratio of the number of different types of data received by the DC.

Assume that $Type_{all}$ includes all types of data, $Type_l$ includes the set of SMP types cached in each Intelligent Node and $Type_m$ includes the set of types where SMP is not cached in each Intelligent Node. For $Type_l$, it is same in every Intelligent Node, as is $Type_m$. The mathematical expression is described as follows:

$$\begin{cases} Type_l = \left\{x \mid x \in Type_{all}\right\}, & |Type_l| = l \\ Type_m = \left\{y \mid y \in Type_{all}\right\}, & |Type_m| = m \\ Type_l \cap Type_m = \emptyset \end{cases} \tag{15}$$

First, the maximum number of packets received by the DC where the type belongs to $Type_l$ and the minimum number of packets received by the DC where the type belongs to $Type_m$ are calculated.

Assume that the variable T_x records the number of packets where the type is x received by the DC, and the function of $N(x)$ is to calculate the number of data-upload packets received by the DC where the type is x. Therefore, the variable T_i represents the maximum number of packets received by the DC where the type is i while the variable T_j represents the minimum number of packets received by the DC where the type is j. The mathematical expression is shown in (16).

$$\begin{cases} T_i = min\{N(x), x \in Type_l\} \\ T_j = max\{N(y), y \in Type_m\} \end{cases} \tag{16}$$

Second, the threshold is calculated to determine whether to publish the program or not. The threshold f expression is shown in (17).

$$f = \alpha \times T_i - T_j \tag{17}$$

If the value of f is less than 0, then release the SMP where the type is j, otherwise the status is maintained. Additional details are shown in Algorithm 4.

To state the parameters of this chapter clearly, the main notions introduced in this paper can be found in Table 3.

TABLE 3. Parameter descriptions in Chapter IV.

Parameter	State
f	The threshold to determine whether to publish a program
α	The coefficient
T_x	The number of packets where the type received by the DC is x
$N(x)$	The method to calculate the number of data-upload packets received by the DC

Algorithm 4 Service Micro-Program Publishing Scheme

Initialize: The Data Center records the set of published SMP types as $\text{Type}_l\,[l]$, while the set of unpublished SMP types as $\text{Type}_m[\text{m}]$. Define the variable num[l+m] to record the quantity of every type number of upload packets. Define the variable min to set the minimum amount of data in $\text{Type}_l\,[l]$, and min_t to record the type of the minimum amount of data. Similarly, define the variable max to set the maximum amount of data in $\text{Type}_m[\text{m}]$, and max_t to record the type of the maximum amount of data.

```
When k upload packets are received:
//compute the num[].
For each packet p ∈ packet[k] Do
    If packet p is an uploaded packet then
        num[p.typenum] = num[p.typenum] +1;
    End If
End For
//find min and min_t.
For each t_l ∈ Type_l[l] Do
    If min > num[t_l] then
        min = num[t_l];
        min_t = t_l;
    End If
End For
//find max and max_t
For each t_m ∈ Type_m[m] Do
    IF max < num[t_m] then
        max = num[t_m];
        max_t = t_m;
    End If
End For
//compute the threshold
f = α × min − max;
If f < 0 then
    Publish the type of t_m SMP packet;
    Replace min_t to max_t in Type_l[l];
    Replace max_t to min_t in Type_m[m];
End If
```

V. THE EXPERIMENTAL RESULTS AND ANALYSIS

A. INITIALIZE THE NETWORK

To analyze the performance of ODAS-CC model, we assume that the network topology is as shown in FIGURE 3, where each node is an Intelligent Node. Among them, nodes numbered 1 to 4 are edge nodes, also called fog network nodes.

Before the packet is uploaded, the network needs to be initialized. That is, the data center (DC) publishes AMP packet and preset SMP packets. At the same time, the FTable of each Intelligent Node is also initialized.

Assume that the standard packet size is 64KB, a program packet size equals 2 standard packets, and the raw packet size equals a standard packet. Additionally, the packet contains the field of the packet size. In the experiment, there are 50 types of data types in total, corresponding to 50 types of SMPs. Data-upload packets or service-request packets are randomly generated from the areas managed by Intelligent Nodes numbered 1 to 4. For instance, the area managed by Intelligent Node numbered 1 is called Ar1.

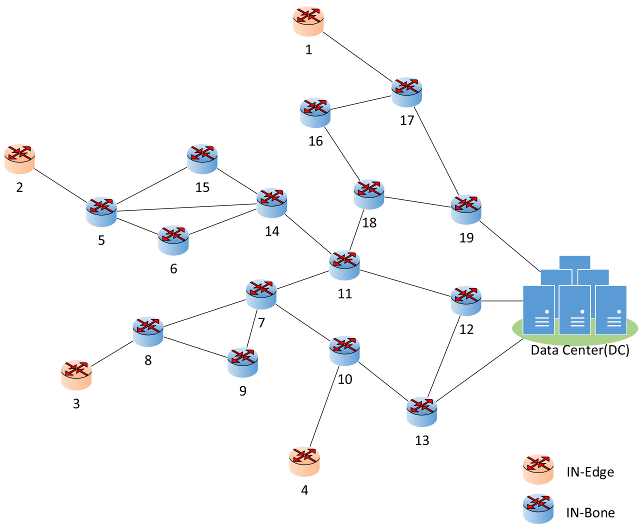


FIGURE 3. The topological structure of the network we assume.

However, the contrasted experiment object is the traditional data forwarding network, which is currently the TCP/IP [27] network centered on data transmission. We call it the data-transmission-based network (DTBN) in the following experiments. According to FIGURE 3, we can see that there are multiple paths from the edge node to the DC, but the traditional TCP/IP network, which selects only one of the paths for transmission, is a single-path transmission network, so we take the average value of these paths in the DTBN.

Next, we evaluate the performance of the ODAS-CC model from the three following aspects: network traffic, delay, and energy consumption.

B. NETWORK TRAFFIC ANALYSIS

We analyze the network traffic from two aspects, one is the data-upload packets generated from various types of sensors, and the other is the service-request packets sent by users.

Next, the network traffic is analyzed from the aspect of data-upload packets generated from various types of sensors.

To make the experimental results more effective and to be able to easily observe the impact of different degrees of service conversion on network traffic, we use two sets of degree of service conversion. The degree of service conversion is set from 0.5 to 0.7 in the set1 group while it is set from 0.3 to 0.5 in the set2 group, and the degree of data aggregation is set from 0.7 to 0.9 in both group. Additionally, whether a new SMP needs to be published under $\alpha = 2$ is detected when the DC receives 2000 non-redundant data-upload packets.

However, the different degrees of service conversion have no effect on the network traffic in DTBN since network nodes (routers) in the DTBN are not able to run the program. In other words, the network traffic under the DTBN is the same regardless of set1 or set2. Further, when p_s and sp_{max} are changed, the network traffic under DTBN remains unchanged.

In the first experiment, the probability of owning such an SMP and generating such a service (p_s) is set 0.3 in each Intelligent Node. In addition, the ratio of the maximum number of SMPs stored by each Intelligent Node to the number of total SMPs (sp_{max}) is set to 0.6. Then, 100000 data-upload packets are randomly generated.

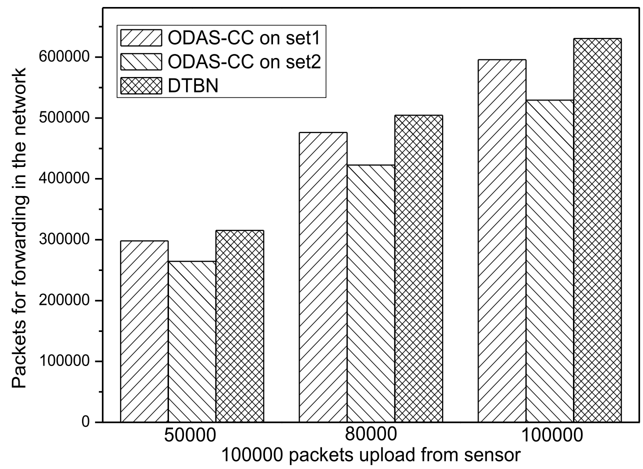


FIGURE 4. The number of packets for forwarding in the network with 100000 packets of ODAS-CC and DTBN.

In FIGURE 4, the vertical axis represents the number of standard packets, that is, the number of bytes transmitted in the network divided by the number of standard packet bytes. Same operations are performed in later experiments.

From FIGURE 4, we can see that for the amount of data transmitted in the network, ODAS-CC on set1 is less than that of DTBN, which is reduced by 5.5%, while in the ODAS-CC on set2 where the degree of service conversion is relatively large, the amount of data transmitted in the network is reduced even more, which is a reduction of 16.0% compared to the DTBN. From the above data, it can be known that the amount of data transmitted are reduced in the network when the degree of service conversion is increased. Although each Intelligent Node can process the data by data aggregation or service conversion, under our forwarding scheme, that data-upload packets are forwarded to all the paths that lead to the DC, and the amount of data forwarded by the network is not particularly reduced.

However, we can see that the amount of data received by the DC under ODAS-CC on set1 is smaller than the DTBN from FIGURE 5, which is reduced by 21.8%, while in the ODAS-CC on set2, where the degree of service conversion is relatively large, the amount of data received by the DC is reduced even more, which is a reduction of 35.3% compared to the DTBN. From the above data, the pressure on the DC is significantly reduced under the ODAS-CC. This is because the raw data under the ODAS-CC was processed by data aggregation or service conversion before it reached the DC and removed the redundant packets in each Intelligent Node.

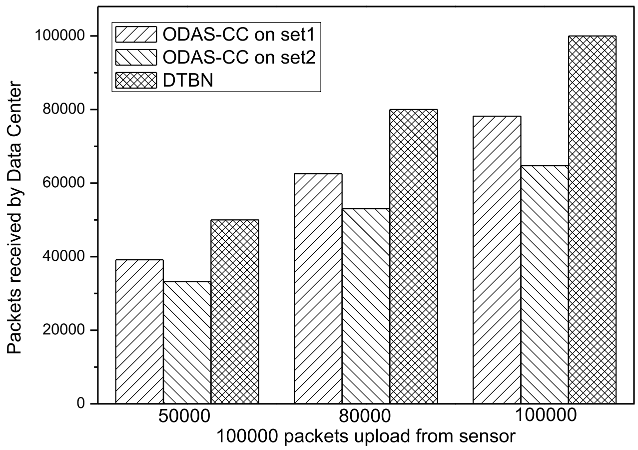


FIGURE 5. The number of packets received by the DC in the network with 100000 packets of ODAS-CC and DTBN.

In the following, we change the network parameters to observe the performance, mainly to analyze the impact of sp_{max} and p_s on network traffic.

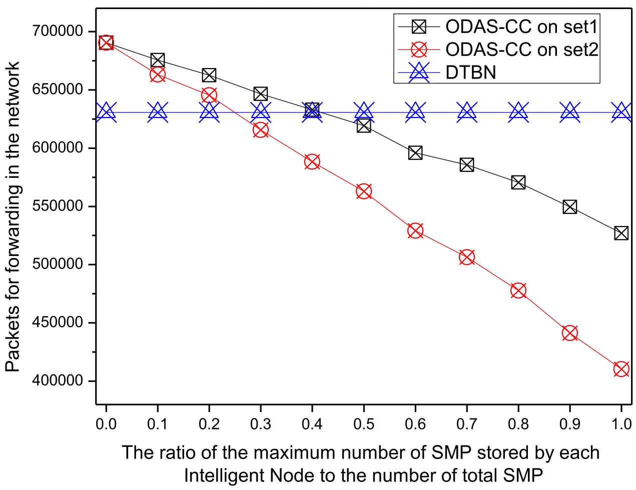


FIGURE 6. The number of packets for forwarding in the network under sp_{max} with 100000 packets of ODAS-CC and DTBN.

According to FIGURE 6, when sp_{max} is 40.0% or even lower, the amount of data transmitted in the network under ODAS-CC has become slightly larger than the DTBN because of the forwarding scheme. Specifically, there are few SMPs in each Intelligent Node, so the probability that the data-upload packets can convert to services becomes lower. Most of the data can only be forwarded after data aggregation while the degree of data aggregation is not particularly noticeable. However, with the increase of sp_{max}, the amount of data transmitted in the network under ODAS-CC is reduced compared to that of the DTBN. When sp_{max} is 60.0% to 80.0%, the amount of data transmitted under ODAS-CC on set1 is reduced by 5.0% to 10.2% compared to that of the DTBN, while on set2 is reduced by 15.1% to 25.2%.

However, as shown in FIGURE 7, from the perspective of the DC, the amount of data received by the DC under

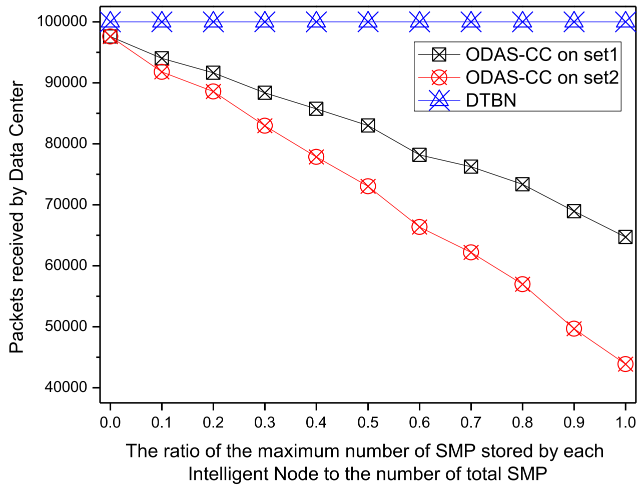


FIGURE 7. The number of packets received by the DC under different sp_{max} with 100000 packets of ODAS-CC and DTBN.

ODAS-CC is smaller than that of the DTBN no matter how sp_{max} changes. This is because the raw data will be processed to different degrees when passing through each Intelligent Node. Even if there is no SMP, the data will be aggregated. However, with the increase of sp_{max}, the amount of data received by the DC under ODAS-CC is decreased. When sp_{max} is 80.0%, the amount of data received by the DC under ODAS-CC on set2, where the degree of service conversion is relatively large, is reduced by 50.5% compared to that of the DTBN.

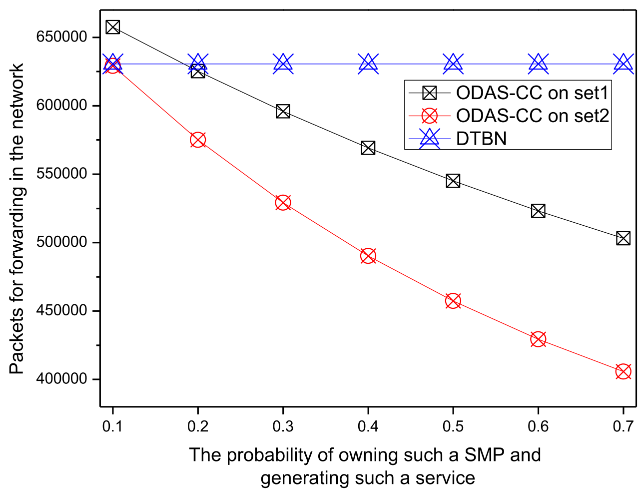


FIGURE 8. The number of packets for forwarding in the network under different p_s with 100000 packets of ODAS-CC and DTBN.

As shown in FIGURE 8 and FIGURE 9, the greater the p_s, the smaller the amount of data under ODAS-CC regardless of what is transmitted in the network or received by the DC. However, in FIGURE 8, if p_s is lower than 10.0%, the amount of data transmitted in the network under ODAS-CC on set1 is slightly larger than that of DTBN, which is increased by 3.5%. This is because ODAS-CC forwards the

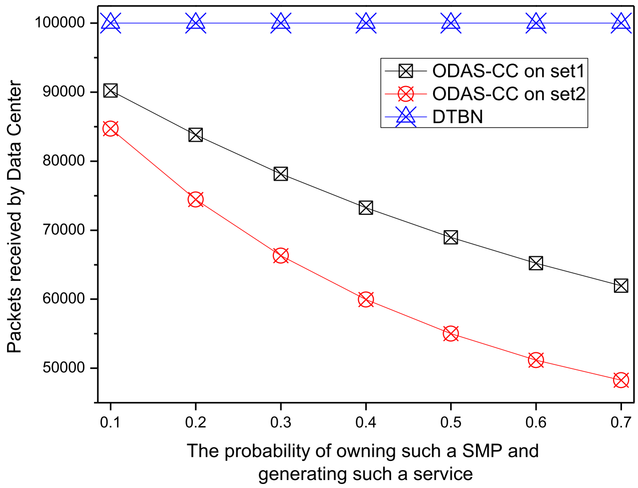


FIGURE 9. The number of packets received by the DC under different p_s with 100000 packets of ODAS-CC and DTBN.

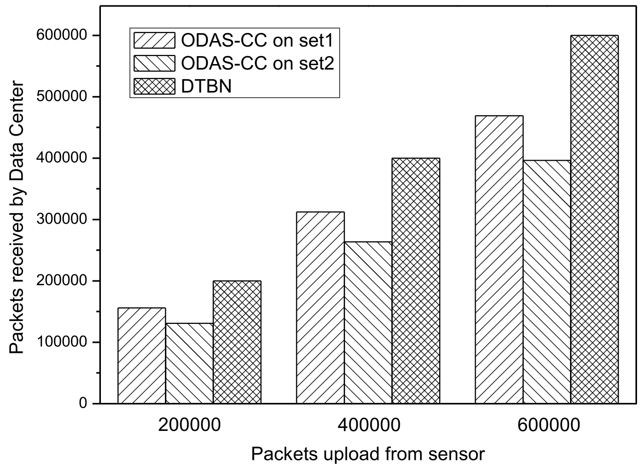


FIGURE 11. The number of packets received by the DC with 200000 to 600000 packets of ODAS-CC and DTBN.

data-upload packets to all the paths that can lead to the DC. Although sp_{max} is 0.6, p_s is low, and most of the data can only be forwarded after data aggregation. However, in the case of p_s from 30.0% to 50.0%, the amount of data transmitted in the network under ODAS-CC on set1 is reduced by 6.1% to 13.8% compared with that of the DTBN, and the amount of data received by the DC is reduced by 21.2% to 30.9%. For ODAS-CC on set2, the amount of data transmitted in the network and the amount of data received by the DC is smaller, reduced by 16.0% to 27.2% and 33.4% to 45.1%, respectively.

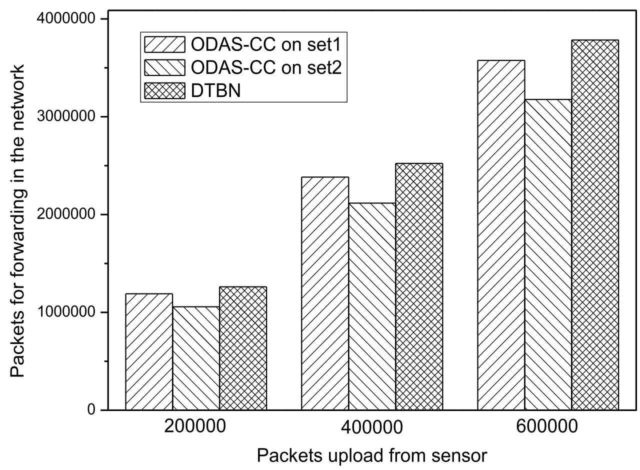


FIGURE 10. The number of packets for forwarding in the network with 200000 to 600000 packets of ODAS-CC and DTBN.

FIGURE 10 and FIGURE 11 show the trend of the amount of data transmitted in the network and received by DC. respectively, with the increase of the number of data-upload packets sent by sensors. For each 200000 data-upload packets increase, the increase in the amount of data transmitted under ODAS-CC on set1 is 94.1% of the DTBN while on set2, it is 84.0%. This demonstrates that with the increase in the number of data packets sent, the increase of the amount of data transmitted under ODAS-CC is smaller than that of the DTBN, and the larger degree of service conversion, the smaller the increase. There are similar rules for the amount of data received by the DC. For each 200000 data-upload packets increase, the amount of data received by the DC under ODAS-CC on set1 is 78.2% of the DTBN, while on set2, it is 65.8%. When the number of data-upload packet increased, the data are processed by SMP or AMP under ODAS-CC, but DTBN does not, so the increment is less than that of the DTBN.

Then, we analyze network traffic from the perspective of the service-request packets sent by users.

In this experiment, since the service-request packet and the service-return packet cannot be aggregated or converted to services under ODAS-CC, the impact of different degrees of service conversion on the network traffic is not considered. Assume that each Intelligent Node can cache various types of services, and each type of service cached has the same proportion of the total amount of such services (p_c). However, since the DTBN cannot cache data, the amount of data transmitted in the network remains unchanged when p_c is changed.

Then, 100000 service-request packets are generated from users.

As is shown in FIGURE 12, when p_c equals 3.0% to 11.0%, the amount of data transmitted in the network under ODAS-CC is 1192546.8 to 1009910.8, which is a 5.2% to 20.3% reduction in data transmitted compared to that of DTBN. Moreover, when p_c increases, the amount of data transmitted in the network decreases. The reason for this result is that, in the ODAS-CC, when a service-request packet is forwarded to an Intelligent Node on a higher layer, it is more likely to satisfy this service request. The greater the p_c value, the lower the probability that service-request packets are forwarded to upper Intelligent Nodes, thereby reducing the number of forwarding hops of the service request packet.

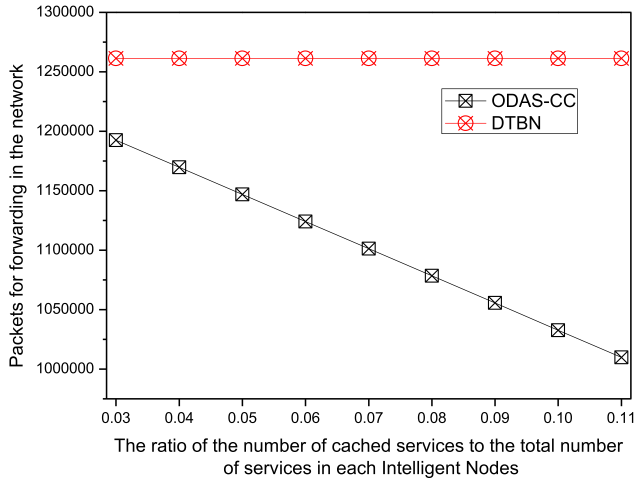


FIGURE 12. The number of packets for forwarding in the network under the different p_c with 100000 packets of ODAS-CC and DTBN.

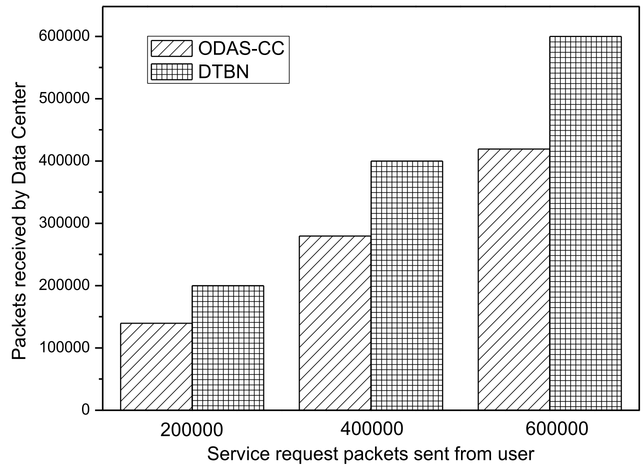


FIGURE 14. The number of packets received by the DC under a p_c of 7% with 200000 to 600000 packets of ODAS-CC and DTBN.

Therefore, the amount of data transmitted in the network is reduced.

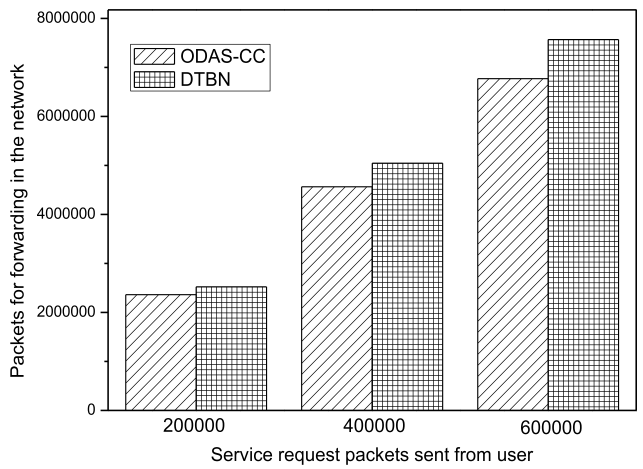


FIGURE 13. The number of packets for forwarding in the network under a p_c of 7% with 200000 to 600000 packets of ODAS-CC and DTBN.

FIGURE 13 and FIGURE 14 show, under a p_c equal to 7.0%, the trend of the amount of data transmitted in the network and received by the DC, respectively, with the increase of the number of service-request packets sent by users. In FIGURE13, regardless if 200000 or 600000 service-request packets are issued, the amount of data transmitted in the network under ODAS-CC is reduced by 12.9% compared to DTBN. In addition, for each 200000 service-request packets increase, the increase in the amount of data transmitted under ODAS-CC is 87.2% of the DTBN. In the ODAS-CC, service-request packet will not to be forwarded if it can be satisfied, so the number of hops for service-request packet transmission is reduced; therefore, the amount of data transmitted is less.

There are similar rules for FIGURE 14. No matter how many service-request packets are sent by the user, the service-request packet received by the DC under ODAS-CC is reduced by 29.8% compared to that of DTBN, and the increase in the amount of data received by the DC under ODAS-CC is 69.0% of that of the DTBN for each 200000 service-request packets increase. The Intelligent Node will return the requested service if it can satisfy the service request, so service-request packets need not to be forwarded to the DC every time, unlike the DTBN, thereby reducing the number of service-request packets received by the data center.

C. LATENCY ANALYSIS

Similar to Part B in this section, we also analyze the latency of ODAS-CC from two perspectives. One is the latency of data upload and the other is service latency.

In the experiment, the transmission time between two Intelligent Nodes is set between 15.0 ms and 50.0 ms. We analyze the latency of data upload first.

As is shown in FIGURE 15, the average latency of data uploaded to DC under ODAS-CC has not changed greatly in the case of 10000, 50000, and 100000 data-upload packets, all of which are 114.3 ms, while the average DTBN needs 155.1 ms. Therefore, the average latency of data uploaded to the DC in ODAS-CC is reduced by 26.3% compared to that of the DTBN.

However, as shown in FIGURE 16, no matter which area sends the data-upload packet, the latency of data upload under ODAS-CC is less than that of the DTBN. Compared with DTBN, the latency of data upload from Ar1 is reduced by 31.9%, and those from Ar2, Ar3 and Ar4 are 27.8%, 24.7%, and 20.5%, respectively.

For Intelligent Node numbered 1, there are 6 paths to the DC with latency times of 89.0 ms, 116.0 ms, 115.0 ms, 128.0 ms, 162.0 ms, and 175.0 ms, respectively. However, in ODAS-CC, data-upload packets are forwarded to all the paths that lead to the DC, and the shortest path always exists in these paths, so the first data-upload packet sent from Ar1 is received by the DC in 89.0 ms, and the subsequent packets

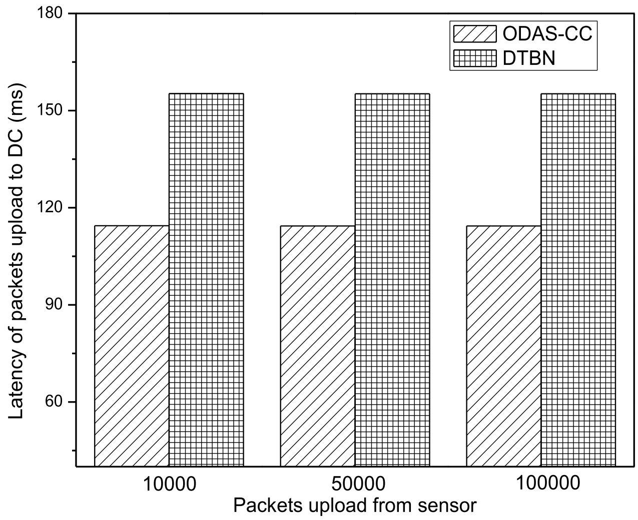


FIGURE 15. The average latency of upload packets forwarded to the DC with 100000 packets of ODAS-CC and DTBN.

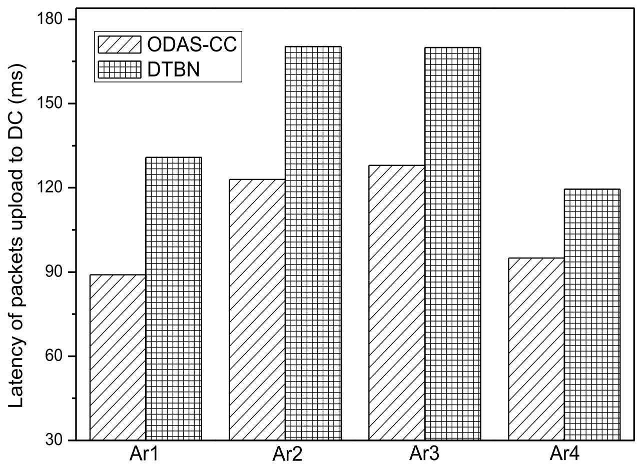


FIGURE 16. The average latency of upload packets from different areas with 100000 packets of ODAS-CC and DTBN.

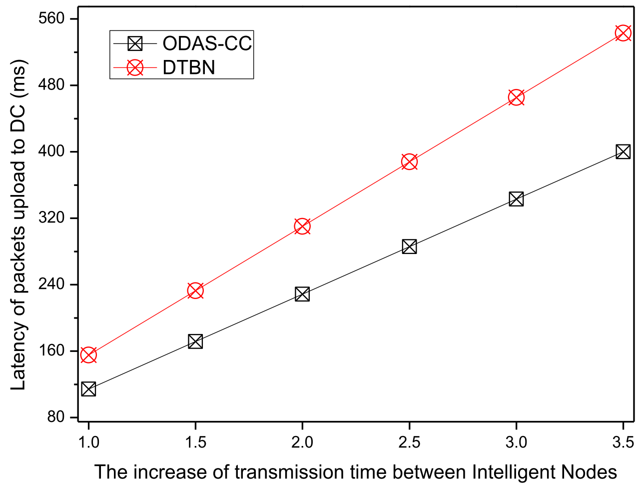


FIGURE 17. The average latency of upload packets forwarded to the DC under the different transmission times with 100000 packets of ODAS-CC and DTBN.

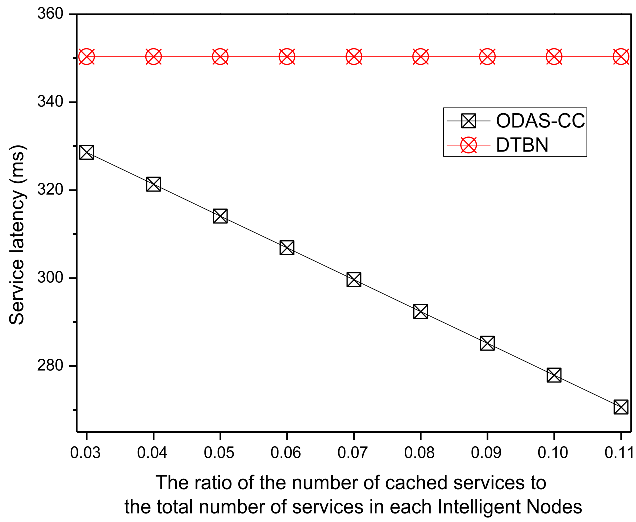


FIGURE 18. The average service latency under the different p_c with 100000 services request packets of ODAS-CC and DTBN.

are redundant data packets that will be cleaned up by the DC. Therefore, randomly sending data-upload packets from Ar1 to Ar4, the data-upload packet is received by the DC under ODAS-CC as quickly as possible compared to the DTBN.

It can be seen from FIGURE 17 that, regardless of the increase in the transmission time between the Intelligent Nodes, the average latency of data upload under ODAS-CC is less than that of the DTBN. While increasing the same transmission time between the Intelligent Nodes, the increment of the data upload latency under ODAS-CC is less than that of the DTBN, which is 73.7% of the DTBN.

Then, we analyze the average service latency. In this experiment, we assume that the DC needs 20 ms to respond to service requests, the transmission time between the Intelligent Nodes is between 15.0 ms and 50.0 ms and 100000 service-request packets are randomly sent from Ar1 to Ar4.

As is shown in FIGURE 18 and FIGURE 19, when the p_c equals from 3.0% to 11.0%, the average service latency and the amount of service request received by the DC under ODAS-CC are smaller than the DTBN. Among them, when p_c equals 7.0%, in ODAS-CC, the service latency is reduced by 14.5% compared with that of the DTBN, and the amount of service requests received by the DC is reduced by 30.1% compared to that of the DTBN. However, as p_c increases, in ODAS-CC, the average service latency and the amount of service requests received by the DC are reduced. For each 1% p_c increase, the average service latency can be reduced by 2.1%, and the amount of service requests received by the DC can be reduced by 4.3%. In ODAS-CC, the greater p_c, the lower the probability that service-request packets are forwarded to the upper Intelligent Node, thereby the service latency and the load of the DC is reduced. However, the nodes

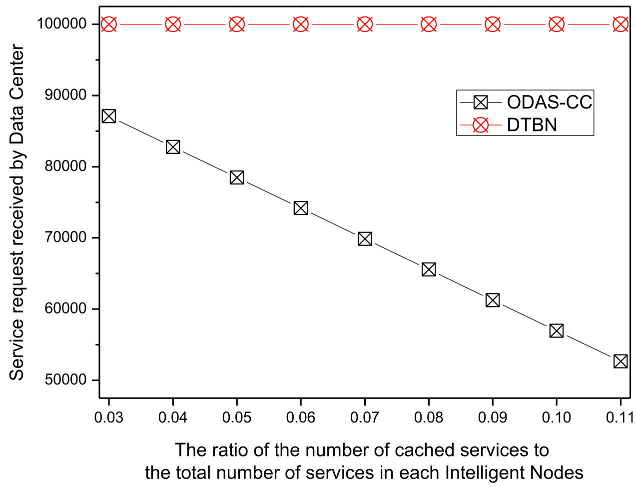


FIGURE 19. **The number of service request received by the DC under the different p_c with 100000 service-request packets of ODAS-CC and DTBN.**

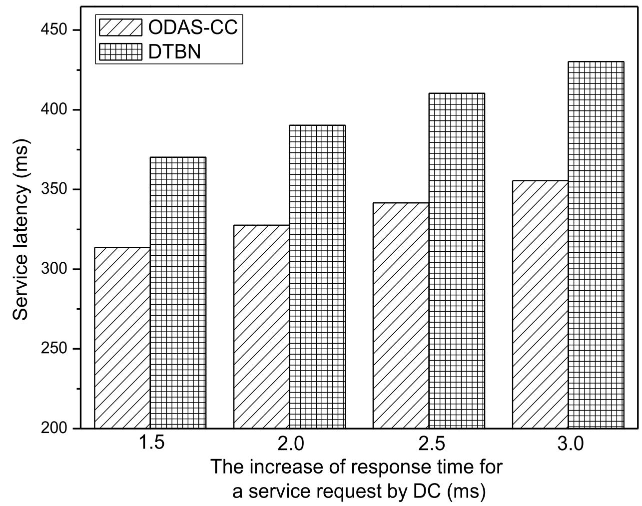


FIGURE 21. **The average service latency under the different response times by the DC with 100000 packets of ODAS-CC and DTBN.**

in the DTBN cannot cache data, so they cannot satisfy the service request, so the p_c is changed, regardless of if the average service latency and the amount of service requests received by the DC remains unchanged.

In the following, we increase the transmission time between the Intelligent Nodes and the response time by the DC for services to observe the changes in service latency.

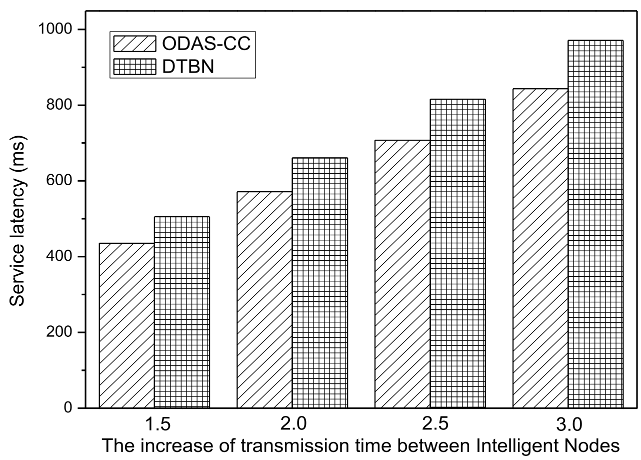


FIGURE 20. **The average service latency under the different transmission times with 100000 service request packets of ODAS-CC and DTBN.**

As is shown in FIGURE 20, when p_c equals 7.0%, as the transmission time between network nodes increases, the average service latency under ODAS-CC is always less than that of the DTBN. While increasing the same transmission time, the average service latency under ODAS-CC is less than that of the DTBN, which is 87.5% of the DTBN. If the distance between the user to the DC is long, the average service latency under ODAS-CC will be more obvious than DTBN, which makes the QoE of the user better.

However, as shown in FIGURE 21, when p_c equals 7.0%, as the response time by the DC for service requests increases, under ODAS-CC, the average service latency still has advantages over DTBN. While increasing the same response time by the DC for service, the average service latency under ODAS-CC is 69.9% of that of the DTBN. However, if the DC is busy, the response time by the DC for service will be increased. In addition, when the response time by the DC for service increases to be greater than the transmission time from the user to the DC, the advantages of ODAS-CC will be more obvious than those of the DTBN. The reason is that the Intelligent Node can return the service instead of always forwarding the request to the DC. Therefore, if the DC is extremely busy, it will not have a large impact on the ODAS-CC.

D. ENERGY CONSUMPTION ANALYSIS

The energy consumption of ODAS-CC consists of the energy consumption in the network, that is the energy consumption of all the Intelligent Nodes in the network, and the energy consumed by the DC. Similarly, we also analyze the energy consumption from two perspectives. The energy consumption from the perspective of data uploading is analyzed first.

In the experiment, $p_s = 0.3$, $sp_{max} = 0.6$, $\theta_c = \theta_s = 0.1$(J/MB) and $\theta_f = 1$(J/MB). The data-upload packet is sent at the maximum speed of the packet that the data center can process. However, the energy consumption in the network and the energy consumed by the DC under DTBN is the same regardless of set1 or set2 because the network nodes (routers) in DTBN are not able to run the program.

As shown in FIGURE 22, the energy consumption in the network under ODAS-CC on set1 is reduced by 2.5% compared to that of the DTBN, while the energy consumption under ODAS-CC on set2 is reduced by 13.3%. Additionally, with the increase in the amount of data-upload packet, the proportion of the reduction did not change significantly. The energy consumption in the network under ODAS-CC is less than that under DTBN, and the greater the degree of

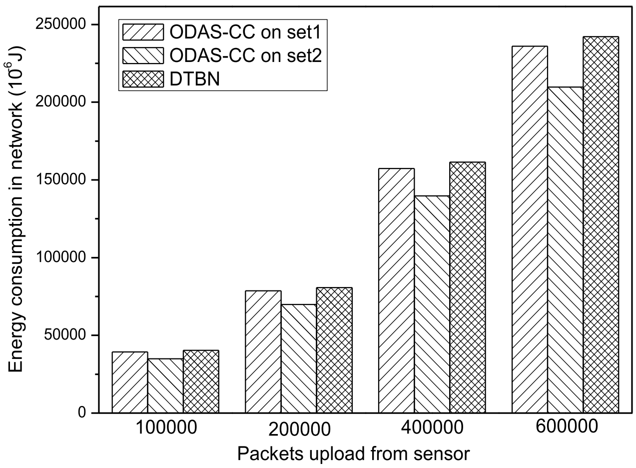


FIGURE 22. The energy consumption in the network because of upload packets with 100000 to 600000 packets of ODAS-CC and DTBN.

service conversion, the less energy consumed in the network under ODAS-CC. Compared to DTBN, energy consumption in the network under ODAS-CC comes from calculating and caching data in addition to forwarding data. In the experiment, both the energy consumption per unit of data by calculating and the energy consumption per unit of data by caching are small compared to the energy consumption per unit of data by forwarding, so the energy consumption in the network under the ODAS-CC is smaller than that under the DTBN. However, if the energy consumption per unit of data by calculating and the energy consumption per unit of data by caching is relatively large, the energy consumption in the network under ODAS-CC may exceed that of the DTBN.

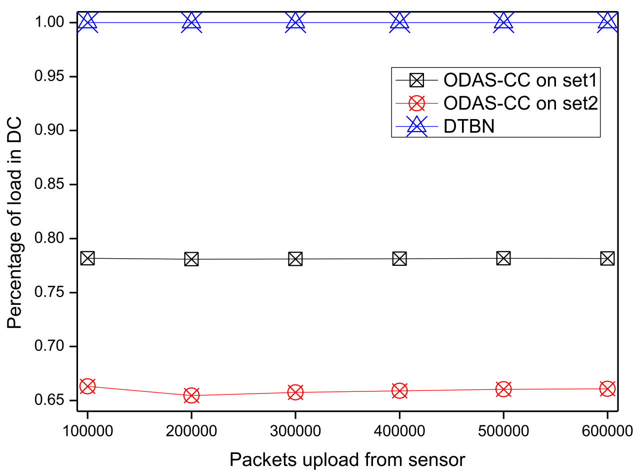


FIGURE 23. The percentage of load in the DC because of upload packets with 100000 to 600000 packets of ODAS-CC and DTBN.

As is shown in FIGURE 23, the DC under the DTBN is always in a state of full load. However, the load of the DC under the ODAS-CC is always lower than that of the DTBN. The load under ODAS-CC on set1 is 78.1%, and the load under ODAS-CC on set2 is 66.0%. However, according to formula (6), it can be inferred that the energy consumed by the DC under ODAS-CC is significantly less than that of the DTBN, and the greater degree of service conversion, the more energy is reduced. This is because the data-upload packets under the ODAS-CC have been processed (data aggregation or service conversion) and redundant packets are removed when they arrive at each upper Intelligent Node that is on the path that leads to the DC. Therefore, although data-upload packets are forwarded on all the paths that lead to the DC, the number of packets handled by the DC under ODAS-CC is relatively small, thereby reducing the load on the DC.

In the following, the energy consumption from the perspective of service requests is analyzed.

In the experiment, $p_c = 0.07$, $sp_{max} = 0.6$, $\theta_c = \theta_s = 0.1(\mathrm{J/MB})$ and $\theta_f = 1(\mathrm{J/MB})$. The service-request packet is sent at the maximum speed of the packet that the data center can process. However, the service-request packet and service-return packet cannot be processed under ODAS-CC, so the impact of different degrees of service conversion on both the energy consumption in the network and the energy consumed by the DC are not considered.

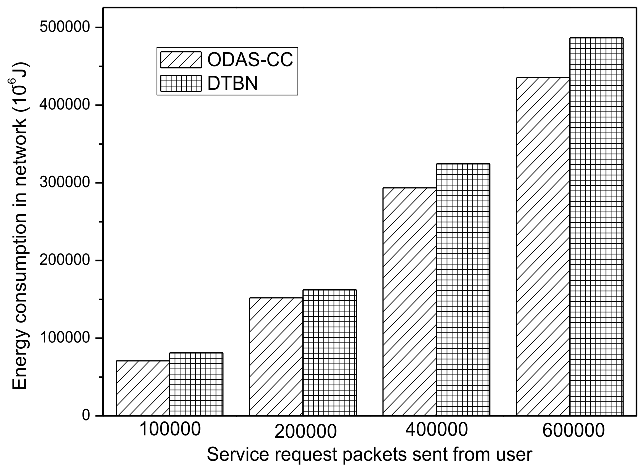


FIGURE 24. The energy consumption in the network because of service requests with 100000 to 600000 packets of ODAS-CC and DTBN.

As is shown in FIGURE 24, the energy consumption in the network under ODAS-CC is reduced by 12.7% compared to that under DTBN. In addition, with the increase in the amount of the service-request packet, the proportion of reduction does not change significantly. Under ODAS-CC, although the number of hops for a service request can be reduced, the service needs to be cached when the service returns. Therefore, in the case where the energy consumption per unit of data by caching is relatively small, the energy consumption in the network under ODAS-CC is smaller than the DTBN. However, if the energy consumption per unit of data by caching is relatively large, the energy consumption in the network under ODAS-CC may be greater than the DTBN.

As is shown in FIGURE 25, the DC under DTBN is always under a full load, while the load of the DC under ODAS-CC

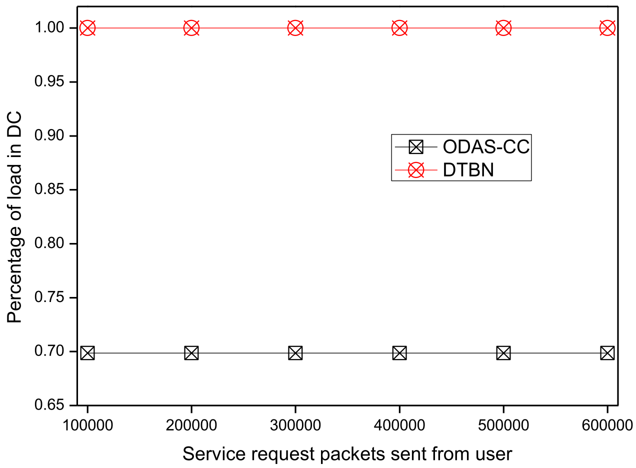


FIGURE 25. **The percentage of load in the DC because of service requests with 100000 to 600000 packets of ODAS-CC and DTBN.**

is always lower than that of the DTBN. The load of the DC under ODAS-CC is 70.3%, which means it can save 30.2% of the energy compared to that of the DTBN according formula (6). The reason is that under ODAS-CC, the Intelligent Node may be able to satisfy service requests instead of requesting the DC every time as in the DTBN, which reduces the load on the DC, thereby reducing the energy consumed by the DC.

VI. CONCLUSIONS

In our paper, a services-based network architecture is proposed, which is named ODAS-CC. Under this model, the service micro-program (SMP) publishing and updating scheme is performed in the data center, which can provide the SMP to network devices to convert data to services. The packet schedule scheme and services conversion are done in the network devices. In addition, the evaluation model for energy consumption, the evaluation performance indicators for latency and load are given with details provided. Additionally, the performance of the ODAS-CC model is evaluated comprehensively. The theoretical analyses and experimental results show that, compared with the previous approach, the ODAS-CC model can reduce the network traffic, reduce the average latency of data upload by 26.3%, the service latency by up to 30.1%, and the data center load, thereby reducing energy consumption by 20.0% to 30.0%.

In the context of the explosive growth of the Internet of Things (IoT), it is a good green low-latency solution. However, the main constraints of ODAS-CC performance are the storage space of Intelligent Nodes, the degree of service conversion and data aggregation. In the future, we will further explore the security of the ODAS-CC model and optimize the ODAS-CC model.

REFERENCES

[1] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of things," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, Jan./Mar. 2018.

[2] X. Liu, S. Zhao, A. Liu, N. Xiong, and A. V. Vasilakos, "Knowledge-aware proactive nodes selection approach for energy management in Internet of things," *Future Gener. Comput. Syst.*, to be published, doi: 10.1016/j.future.2017.07.022.

[3] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "REPLISOM : Disciplined tiny memory replication for massive IoT devices in LTE edge cloud," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 327–338, Jun. 2016.

[4] W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, and G. Wang, "Data processing and text mining technologies on electronic medical records: A review," *J. Healthcare Eng.*, vol. 2018, Apr. 2018, Art. no. 4302425, doi: 10.1155/2018/4302425.

[5] K. Ota, M. Dong, J. Gui, and A. Liu, "QUOIN: Incentive mechanisms for crowd sensing networks," *IEEE Netw.*, vol. 32, no. 2, pp. 114–119, Mar./Apr. 2018.

[6] J. Tang, A. Liu, J. Zhang, N. X. Xiong, Z. Zeng, and T. Wang, "A trust-based secure routing scheme using the traceback approach for energy-harvesting wireless sensor networks," *Sensors*, vol. 18, no. 3, p. 751, 2018, doi: 10.3390/s18030751.

[7] A. Liu, M. Huang, M. Zhao, and T. Wang, "A smart high-speed backbone path construction approach for energy and delay optimization in WSNs," *IEEE Access*, vol. 6, no. 1, pp. 13836–13854, 2018.

[8] J. Xu, X. Liu, M. Ma, A. Liu, T. Wang, and C. Huang, "Intelligent aggregation based on content routing scheme for cloud computing," *Symmetry*, vol. 9, no. 10, p. 221, 2017, doi: 10.3390/sym9100221.

[9] Q. Xu, Z. Su, Q. Zheng, M. Luo, and B. Dong, "Secure content delivery with edge nodes to save caching resources for mobile users in green cities," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2550–2559, Jun. 2018.

[10] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack, "WAVE: Popularity-based and collaborative in-network caching for content-oriented networks," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM Workshops)*, Mar. 2012, pp. 316–321.

[11] X. Liu *et al.*, "Construction of large-scale low-cost delivery infrastructure using vehicular networks," *IEEE Access*, vol. 6, no. 1, pp. 21482–21497, 2018.

[12] N. Zhang, P. Yang, J. Ren, D. Chen, Y. Li, and X. Shen, "Synergy of big data and 5G wireless networks: Opportunities, approaches, and challenges," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 12–18, Feb. 2018.

[13] H. Dai, G. Chen, C. Wang, S. Wang, X. Wu, and F. Wu, "Quality of energy provisioning for wireless power transfer," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 2, pp. 527–537, Feb. 2014.

[14] M. Z. A. Bhuiyan, G. Wang, J. Wu, J. Cao, X. Liu, and T. Wang, "Dependable structural health monitoring using wireless sensor networks," *IEEE Trans. Depend. Sec. Comput.*, vol. 14, no. 4, pp. 363–376, Jul./Aug. 2017.

[15] J. Li, Z. Liu, X. Chen, F. Xhafa, X. Tan, and D. S. Wong, "L-EncDB: A lightweight framework for privacy-preserving data queries in cloud computing," *Knowl.-Based Syst.*, vol. 79, pp. 18–26, May 2015.

[16] Q. Xu, Z. Su, and K. Yang, "Optimal control theory-based epidemic information spreading scheme for mobile social users with energy constraint," *IEEE Access*, vol. 5, pp. 14107–14118, 2017.

[17] J. Xu, A. Liu, N. Xiong, T. Wang, and Z. Zuo, "Integrated collaborative filtering recommendation in social cyber-physical systems," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 12, 2017.

[18] K. Xie, J. Cao, X. Wang, and J. Wen, "Optimal resource allocation for reliable and energy efficient cooperative communications," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 4994–5007, Oct. 2013.

[19] X. Liu, N. Xiong, N. Zhang, A. Liu, H. Shen, and C. Huang, "A trust with abstract information verified routing scheme for cyber-physical network," *IEEE Access*, vol. 6, pp. 3882–3898, 2018.

[20] Y. Liu *et al.*, "QTSAC: An energy-efficient MAC protocol for delay minimization in wireless sensor networks," *IEEE Access*, vol. 6, no. 1, pp. 8273–8291, 2018.

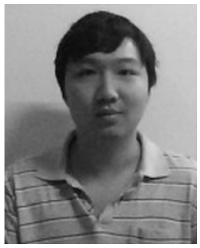
[21] M. Huang *et al.*, "A services routing based caching scheme for cloud assisted CRNs," *IEEE Access*, vol. 6, no. 1, pp. 15787–15805, 2018.

[22] X. Liu, Y. Liu, H. Song, and A. Liu, "Big data orchestration as a service network," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 94–101, Sep. 2017.

[23] A. Liu and Q. Liu, "On the hybrid using of unicast-broadcast in wireless sensor networks," *Comput. Elect. Eng.*, to be published, doi: 10.1016/j.compeleceng.2017.03.004.

[24] Y. Liu, A. Liu, S. Guo, Z. Li, Y. J. Choi, and H. Sekiya, "Context-aware collect data with energy efficient in cyber–physical cloud systems," *Future Gener. Comput. Syst.*, to be published, doi: 10.1016/j.future.2017.05.029.

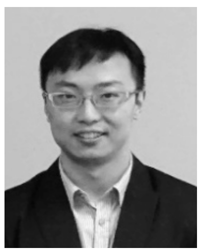
[25] A. Liu and S. Zhao, "High performance target tracking scheme with low prediction precision requirement in WSNs," *Int. J. Ad Hoc Ubiquitous Comput.*, to be published. [Online]. Available: http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijahuc
[26] T. Han and N. Ansari, "Network utility aware traffic load balancing in Backhaul-constrained cache-enabled small cell networks with hybrid power supplies," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2819–2832, Oct. 2017.
[27] T. V. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," *IEEE/ACM Trans. Netw.*, vol. 5, no. 3, pp. 336–350, Jun. 1997.
[28] X. Xu, N. Zhang, H. Song, A. Liu, M. Zhao, and Z. Zeng, "Adaptive beaconing based MAC protocol for sensor based wearable system," *IEEE Access*, vol. 6, pp. 29700–29714, 2018.
[29] A. Liu, W. Chen, and X. Liu, "Delay optimal opportunistic pipeline routing scheme for cognitive radio sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 4, 2018, doi: /10.1177/1550147717749745.
[30] X. Liu, M. Dong, Y. Liu, A. Liu, and N. N. Xiong. "Construction low complexity and low delay CDS for big data code dissemination," *Complexity*, vol. 2018, Jun. 2018, Art. no. 5429546, doi: 10.1155/2018/5429546.
[31] M. F. Majeed, M. N. Dailey, R. Khan, and A. Tunpan, "Pre-caching: A proactive scheme for caching video traffic in named data mesh networks," *J. Netw. Comput. Appl.*, vol. 87, pp. 116–130, Jun. 2017.
[32] W. Jiang, G. Wang, M. Z. A. Bhuiyan, and J. Wu, "Understanding graph-based trust evaluation in online social networks: Methodologies and challenges," *ACM Comput. Surv.*, vol. 49, no. 1, 2016, Art. no. 10.
[33] X. Li, A. Liu, M. Xie, N. N. Xiong, Z. Zeng, and Z. Cai, "Adaptive aggregation routing to reduce delay for multi-layer wireless sensor networks," *Sensors*, vol. 18, no. 4, p. 1216, 2018, doi: 10.3390/s18041216.
[34] F. Xiao, W. Liu, Z. Li, L. Chen, and R. Wang, "Noise-tolerant wireless sensor networks localization via multinorms regularized matrix completion," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2409–2419, Mar. 2018.
[35] M. Z. A. Bhuiyan, J. Wu, G. Wang, T. Wang, and M. M. Hassan, "E-sampling: Event-sensitive autonomous adaptive sensing and low-cost monitoring in networked sensing systems," *ACM Trans. Auton. Adapt. Syst.*, vol. 12, no. 1, 2017, Art. no. 1.
[36] X. Liu, G. Li, S. Zhang, and A. Liu, "Big program code dissemination scheme for emergency software-define wireless sensor networks," *Peer-to-Peer Netw. Appl.*, vol. 11, no. 5, pp. 1038–1059, 2018.
[37] J. Wang, A. Liu, T. Yan, and Z. Zeng, "A resource allocation model based on double-sided combinational auctions for transparent computing," *Peer-to-Peer Netw. Appl.*, vol. 11, no. 4, pp. 679–696, 2018.
[38] X. Liu, "A novel transmission range adjustment strategy for energy hole avoiding in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 67, pp. 43–52, May 2016.
[39] H. Y. Xin and X. X. Liu, "Energy-balanced transmission with accurate distances for strip-based wireless sensor networks," *IEEE Access*, vol. 5, pp. 16193–16204, 2017.
[40] P. Le. Nguyen, Y. Ji, Z. Liu, H. Vu, and K.-V. Nguyen, "Distributed hole-bypassing protocol in WSNs with constant stretch and load balancing," *Comput. Netw.*, vol. 129, pp. 232–250, Dec. 2017.
[41] Z. Liu, T. Tsuda, H. Watanabe, S. Ryuo, and N. Iwasawa, "Data driven cyber-physical system for landslide detection," in *Mobile Networks and Applications*. New York, NY, USA: Springer, 2018, pp. 1–12, doi: 10.1007/s11036-018-1031-1.
[42] X. Chen, J. Li, J. Weng, J. Ma, and W. Lou, "Verifiable computation over large database with incremental updates," *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 3184–3195, Oct. 2016.
[43] H. Li, K. Ota, and M. Dong, "Energy cooperation in battery-free wireless communications with radio frequency energy harvesting," *ACM Trans. Embedded Comput. Syst.*, vol. 17, no. 2, 2018, Art. no. 44.
[44] K. Ota, M. S. Dao, V. Mezaris, and F. G. B. De. Natale, "Deep learning for mobile multimedia: A survey," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 13, no. 3s, 2017, Art. no. 34.
[45] J. Xu, K. Ota, and M. Dong, "Real-time awareness scheduling for multimedia big data oriented in-memory computing," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2018.2802913.
[46] Y. Liu, M. Dong, K. Ota, and A. Liu, "ActiveTrust: Secure and trustable routing in wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 2013–2027, Sep. 2016.
[47] J. Li, X. Chen, F. Xhafa, and L. Barolli, "Secure deduplication storage systems supporting keyword search," *J. Comput. Syst. Sci.*, vol. 81, no. 8, pp. 1532–1541, 2015.
[48] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Inf. Sci.*, vol. 379, pp. 42–61, Feb. 2017.
[49] Q. Deng, X. Li, Z. Li, A. Liu, and Y.-J. Choi, "Electricity cost minimization for delay-tolerant basestation powered by heterogeneous energy source," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 12, pp. 5712–5728, 2017.



ZHENGYAN DING is currently pursuing the master's degree with the School of Information Science and Engineering, Central South University, China. His research interests include big data, fog computing networks, and wireless sensor networks.

KAORU OTA received the M.Sc. degree from Oklahoma State University, Stillwater, OK, USA, in 2008, and the Ph.D. degree from The University of Aizu, Japan, in 2012. She is currently an Assistant Professor with the Muroran Institute of Technology, Japan. Her research interests include wireless networks and ubiquitous computing.

YUXIN LIU is currently pursuing the degree with the School of Information Science and Engineering, Central South University, China. Her research interests are services-based networks, crowd sensing networks, and wireless sensor networks.



NING ZHANG (S'12) received the B.Sc. degree from Beijing Jiaotong University, Beijing, China, in 2007, the M.Sc. degree from the Beijing University of Posts and Telecommunications, Beijing, in 2010, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2015. He was a Post-Doctoral Research Fellow with the University of Waterloo and the University of Toronto. He is currently an Assistant Professor with the Department of Computing Sciences, Texas A&M University, Corpus Christi, TX, USA. His current research interests include sensor networks, next generation wireless networks, software-defined networking, green communication, and physical layer security.

MING ZHAO received the M.Sc. and Ph.D. degrees in computer science from Central South University, China, in 2003 and 2007, respectively. He is currently a Professor with the School of Software, Central South University. His major research interest is wireless networks. He is a member of the China Computer Federation.

HOUBING SONG (M'12–SM'14) received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, USA, in 2012. In 2017, he joined the Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA, where he is currently an Assistant Professor and the Director of the Security and Optimization for Networked Globe Laboratory. He served on the faculty of West Virginia University from 2012 to 2017. In 2007, he was an Engineering Research Associate with the Texas A&M Transportation Institute. He is an Editor of four books, including *Smart Cities: Foundations, Principles and Applications*, (Hoboken, NJ: Wiley, 2017), Security and Privacy in *Cyber-Physical Systems: Foundations, Principles and Applications*, (Chichester, U.K.: Wiley-IEEE Press, 2017), *Cyber-Physical Systems: Foundations, Principles and Applications*, (Boston, MA: Academic Press, 2016), and *Industrial Internet of Things: Cybermanufacturing Systems*, (Cham, Switzerland: Springer, 2016). He has authored over 100 articles. His research interests include cyber-physical systems, cyber security and privacy, Internet of Things, edge computing, big data analytics, unmanned aircraft systems, connected vehicles, smart and connected health, and wireless communications and networking. He serves as an Associate Technical Editor for the *IEEE Communications Magazine*.

He is a senior member of ACM. He was the very first recipient of the Golden Bear Scholar Award and the highest campus-wide recognition for research excellence at the West Virginia University Institute of Technology.

ANFENG LIU received the M.Sc. and Ph.D. degrees in computer science from Central South University, China, in 2002 and 2005, respectively. He is currently a Professor with the School of Information Science and Engineering, Central South University. His major research interest is wireless sensor networks. He is a member (E200012141M) of the China Computer Federation.

ZHIPING CAI (M'08) received the B.Eng., M.Sc., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), China, in 1996, 2002, and 2005, respectively. He is currently a Full Professor with the College of Computers, NUDT. His current research interests include network security and big data. He is a Senior Member of CCF. His Ph.D. dissertation received the Outstanding Dissertation Award from the Chinese PLA.