Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 3794175, 13 pages
https://doi.org/10.1155/2018/3794175

Research Article

Processing Optimization of Typed Resources with Synchronized Storage and Computation Adaptation in Fog Computing

Zhengyang Song,[1] Yucong Duan,[1] Shixiang Wan,[2] Xiaobing Sun,[3] Quan Zou,[2] Honghao Gao,[4,5] and Donghai Zhu[1]

[1] *State Key Laboratory of Marine Resource Utilization in the South China Sea, College of Information Science and Technology, Hainan University, Haikou, China*
[2] *School of Computer Science and Engineering, Tianjin University, Tianjin, China*
[3] *School of Information Engineering, Yangzhou University, Yangzhou, China*
[4] *Computing Center, Shanghai University, Shanghai, China*
[5] *Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, Shanghai, China*

Correspondence should be addressed to Yucong Duan; duanyucong@hotmail.com

Received 27 January 2018; Accepted 16 April 2018; Published 30 May 2018

Academic Editor: Xuyun Zhang

Wide application of the Internet of Things (IoT) system has been increasingly demanding more hardware facilities for processing various resources including data, information, and knowledge. With the rapid growth of generated resource quantity, it is difficult to adapt to this situation by using traditional cloud computing models. Fog computing enables storage and computing services to perform at the edge of the network to extend cloud computing. However, there are some problems such as restricted computation, limited storage, and expensive network bandwidth in Fog computing applications. It is a challenge to balance the distribution of network resources. We propose a processing optimization mechanism of typed resources with synchronized storage and computation adaptation in Fog computing. In this mechanism, we process typed resources in a wireless-network-based three-tier architecture consisting of Data Graph, Information Graph, and Knowledge Graph. The proposed mechanism aims to minimize processing cost over network, computation, and storage while maximizing the performance of processing in a business value driven manner. Simulation results show that the proposed approach improves the ratio of performance over user investment. Meanwhile, conversions between resource types deliver support for dynamically allocating network resources.

1. Introduction

With the extensive development of IoT applications, it is necessary to promote storage, computing, and communication capability of IoT devices, such as efficient storage, semantic integration, and parallel processing. Due to powerful computing capability compared with edge devices, cloud computing puts all the computing tasks on the cloud to process data efficiently and provides highly vitalized storage and computing services on top of massively parallel distributed systems [1–3]. However, with the growing quantity of data generated at the edge, the speed is becoming the bottleneck for the cloud-based computing paradigm [4]. Also, future wireless access networks will face the challenge of transporting the ever-increasing volume of data centric traffic with ever tighter timing and quality of service (QoS) requirements [5]. In order to reduce the communication bandwidth needed between edge devices and the central data centre, nascent technologies and applications for mobile computing and IoT are driving computing towards dispersion [6]. Conventional methods, which are based on bandwidth allocation scheme, improve effectively bandwidth utilization efficiency [7, 8]. Some methods [9–11] achieve real-time data analysis, but they have not considered user investment [12, 13], processing cost [14], and the efficient utilization of IoT resources with reduced risks [15, 16] of security and robustness [17] issues.

Fog computing stresses the proximity to terminal users and client objectives, dense geographical distribution and local resource pools, latency reduction, and backbone bandwidth savings to achieve better quality of service (QoS) [18].

Meanwhile, the computing paradigm pushes data analytic and knowledge generation power away from centralized nodes to the logical extremes of a network near the source of the data. Therefore, a large number of heterogeneous, complexity, and hierarchy resources can be processed effectively in fog computing systems. However, storage and computation power in fog computing systems are not balanced. We argue that it is a challenge to provide better load balancing of resources in the Internet of Things. From the perspective of resource owners, obtaining cost-effective resources processing services remains the primary concern for adoption of fog computing services. In [19], the authors not only studied the tradeoff between network performance improvement and the communication overhead of transmitting network information but also proposed to optimize resource allocation in wireless networks.

Fog devices have increasingly generated a large amount of resources including data, information, and knowledge. In [20], Chaim illustrated the concepts of defining data, information, and knowledge. We use Knowledge Graph to solve the problem of extracting relationships from sources of knowledge [21, 22]. Knowledge Graph has become a powerful tool for representing knowledge in the form of labelled digraphs and gives textual information semantics. Knowledge base contains a set of concepts, examples, and relationships [23]. In [24], Duan et al. clarified the architecture of Knowledge Graph in terms of data, information, knowledge, and wisdom. In [25], the authors proposed to designate the form of Knowledge Graph as four basic forms: Data Graph, Information Graph, Knowledge Graph, and Wisdom Graph. In [26], the authors proposed to answer five Ws questions through constructing the architecture composed of Data Graph, Information Graph, Knowledge Graph, and Wisdom Graph. The development of software service system can be divided into data sharing, information transmission, and knowledge creation stages in terms of data, information, and knowledge [27]. In fog computing systems, the computing resources for providing device services are highly limited such as storage capacity, computing power, and network bandwidth. The demand for huge search space and computation power has become a big challenge. We propose a mechanism to improve the ratio of performance over investment when providing resources processing services. The proposed approach minimizes processing cost over network, computation, and storage while maximizing the performance of processing in a business value driven manner. Multitypes users' investment and corresponding benefit rate decide final resource combination cases, which makes the system have a relatively stable state. Furthermore, we design several simulations to verify the feasibility and evaluate the performance of the proposed approach. Currently we did not consider negative situations of which improper arrangement of resources might lead to penalties [28] instead of rewards.

The rest of this paper is organized as follows. In Section 2, we define typed resources and explain the proposed system model. Section 3 provides a running example. Section 4 presents practice of storage and computing collaborative adaptation towards typed IoT resources. Section 5 reports the simulation and summarizes the simulation results. Section 6 reports related work. Finally, we draw conclusions and outline aspects of future work in Section 7.

2. Definitions and System Model

In this section, we propose the definition of typed resources. Furthermore, we design the processing architecture of typed resources in the Internet of Things. In order to satisfy the transmission demands of nodes in a given wireless network, we construct the system model through restricting network resources.

2.1. Definitions of Typed Resources. Wide application of the Internet of Things has been acquiring a huge amount of typed resources. We classify resources into three types: data, information, and knowledge [27]. Data is not specified for stakeholder or machine. Data is the collection of discrete elements and concepts. Information is not specified for stakeholder or machine. Information is conveyed through conceptual mapping and combination of data and is used for interaction. Knowledge is used to reason and predict unknown resources such as data, information, and knowledge. Table 1 provides an explanation of typed resources of data, information, and knowledge. Definitions of resource elements and graphs are as follows.

Definition 1 (resource element ($\mathrm{RE}_{\mathrm{DIK}}$)). We define resource element as

$$\mathrm{RE}_{\mathrm{DIK}} \coloneqq \langle \mathrm{D}_{\mathrm{DIK}}, \mathrm{I}_{\mathrm{DIK}}, \mathrm{K}_{\mathrm{DIK}} \rangle . \tag{1}$$

$\mathrm{D}_{\mathrm{DIK}}$, $\mathrm{I}_{\mathrm{DIK}}$, and $\mathrm{K}_{\mathrm{DIK}}$ represent data, information, and knowledge in DIK hierarchy, respectively, where DIK is an abbreviation of $\mathrm{D}_{\mathrm{DIK}}$, $\mathrm{I}_{\mathrm{DIK}}$, and $\mathrm{K}_{\mathrm{DIK}}$.

Definition 2 (graphs). In our previous work [25], we specified Knowledge Graph in a progressive manner as four basic forms: Data Graph, Information Graph, Knowledge Graph, and Wisdom Graph. In this work, we propose to specify the existing concept of Knowledge Graph in three layers. We define graphs as

$$\mathrm{Graph}_{\mathrm{DIK}} \coloneqq \langle (\mathrm{DG}_{\mathrm{DIK}}), (\mathrm{IG}_{\mathrm{DIK}}), (\mathrm{KG}_{\mathrm{DIK}}) \rangle . \tag{2}$$

$\mathrm{DG}_{\mathrm{DIK}}$, $\mathrm{IG}_{\mathrm{DIK}}$, and $\mathrm{KG}_{\mathrm{DIK}}$ represent Data Graph, Information Graph, and Knowledge Graph, respectively. $\mathrm{DG}_{\mathrm{DIK}}$ is used to model the temporal and spatial features of $\mathrm{D}_{\mathrm{DIK}}$. $\mathrm{IG}_{\mathrm{DIK}}$ is a combination of related $\mathrm{D}_{\mathrm{DIK}}$. $\mathrm{IG}_{\mathrm{DIK}}$ expresses the interaction and transformation of $\mathrm{I}_{\mathrm{DIK}}$ between entities in the form of a directed graph. $\mathrm{KG}_{\mathrm{DIK}}$ is a collection of statistical rules summarized from known resources.

In order to optimize resources processing, it is important to convert the types of resources in the Internet of Things. We give the explanations of the type conversion mechanism as follows.

(a) For conversion from $\mathrm{D}_{\mathrm{DIK}}$ to $\mathrm{I}_{\mathrm{DIK}}$, in the absence of context, $\mathrm{D}_{\mathrm{DIK}}$ is meaningless. We convert $\mathrm{D}_{\mathrm{DIK}}$ to $\mathrm{I}_{\mathrm{DIK}}$ through reorganizing $\mathrm{D}_{\mathrm{DIK}}$. The new collection of $\mathrm{D}_{\mathrm{DIK}}$ corresponds to a different class or concept, which is $\mathrm{I}_{\mathrm{DIK}}$.

Table 1: An explanation of three types of resources.

	D_{DIK}	I_{DIK}	K_{DIK}
Semantic load	Not specified for stakeholder/machine	Specified for stakeholder/machine	Abstracting unknown resources
Form	Discrete elements	Related elements	Probabilistic or categorization
Usage	Identification of existence after conceptualization	Communication	Reasoning and predicting
Graph	DG_{DIK}	IG_{DIK}	KG_{DIK}

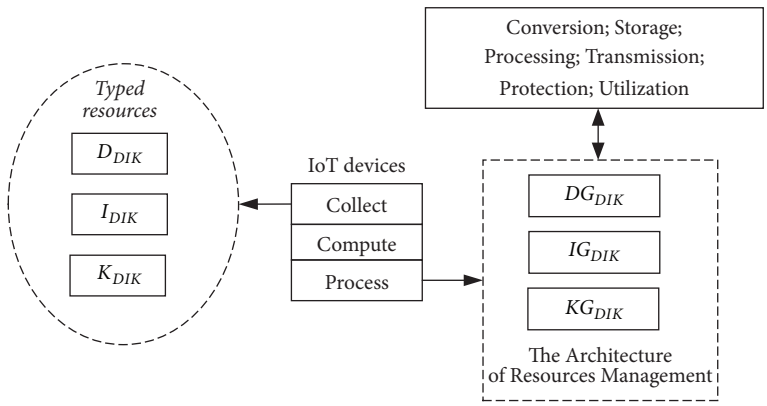

Figure 1: The processing architecture of typed resources in IoT.

(b) For conversion from D_{DIK} to K_{DIK}, D_{DIK} inherits semantic relationships from standard mode and is effectively integrated and reused by other applications. In the conversion process from D_{DIK} to K_{DIK}, we use knowledge validation technology to eliminate redundancy and inconsistency of D_{DIK} through linking D_{DIK} sources and semantic constraints. Then, we identify the most reliable D_{DIK} to form K_{DIK}.

(c) For conversion from I_{DIK} to K_{DIK}, I_{DIK} is used to express the interaction and collaboration between entities. Through the classifying and abstracting interactive records or behaviour records related to the dynamic behaviour of entities, we obtain K_{DIK} in the form of statistical rules. We infer K_{DIK} from known resources and collect necessary I_{DIK} in the process of inference through appropriate research techniques such as experiments and surveys.

(d) Regarding conversion from I_{DIK} to D_{DIK}, we know that conversion from I_{DIK} to D_{DIK} is the transition from concept set to resource instances. I_{DIK} expresses the dynamic interaction and collaboration between entities; we obtain D_{DIK} through observing an object at a certain time in a static state.

(e) For conversion from K_{DIK} to D_{DIK}, according to knowledge reasoning, we establish relevant examples for extracted collection of K_{DIK}. Relationships between K_{DIK} nodes are associated with instances in the form of attributes.

(f) For conversion from K_{DIK} to I_{DIK}, the schema-less feature of KG_{DIK} makes it possible to link and utilize a richer knowledge base to help users make decisions, from K_{DIK} retrieval to I_{DIK} creation.

2.2. Processing Architecture of Typed Resources in the Internet of Things. Generally, we obtain I_{DIK} through mining D_{DIK} and K_{DIK} through deducting I_{DIK} and intelligence from K_{DIK}. Therefore, D_{DIK}, I_{DIK}, K_{DIK}, and wisdom are the layers of a gradual relationship. The Knowledge Graph is divided into four levels: DG_{DIK}, IG_{DIK}, KG_{DIK}, and Wisdom Graph. Figure 1 shows the processing architecture of typed resources in IoT on the basis of DG_{DIK}, IG_{DIK}, and KG_{DIK}. In this architecture, graph types are allowed to convert between each other. We provide the definitions of DG_{DIK}, IG_{DIK}, and KG_{DIK} as follows.

Definition 3 (DG_{DIK}). We define DG_{DIK} as

$$DG_{DIK} := \text{collection}\{\text{array, list, stack, queue, tree, graph}\}. \tag{3}$$

DG_{DIK} is a collection of discrete elements expressed in the form of various data structures including arrays, lists, stacks, trees, and graphs. DG_{DIK} featuring a static state is more about expressing a structural relationship. DG_{DIK} is used to model the temporal and spatial features of DG_{DIK}.

Definition 4 (IG_{DIK}). We define IG_{DIK} as

$$IG_{DIK} := \text{combination}\{\text{related } D_{DIK}\}. \tag{4}$$

IG_{DIK} is a combination of related D_{DIK}. IG_{DIK} expresses the interaction and transformation of I_{DIK} between entities in the form of a directed graph. IG_{DIK} records the interaction between entities including both direct interaction and indirect interaction. Also, IG_{DIK} can be expressed using multiple tuples.

Definition 5 (KG_{DIK}). We define KG_{DIK} as

$$KG_{DIK} := \text{collection}\{\text{Statistic Rules}\}. \tag{5}$$

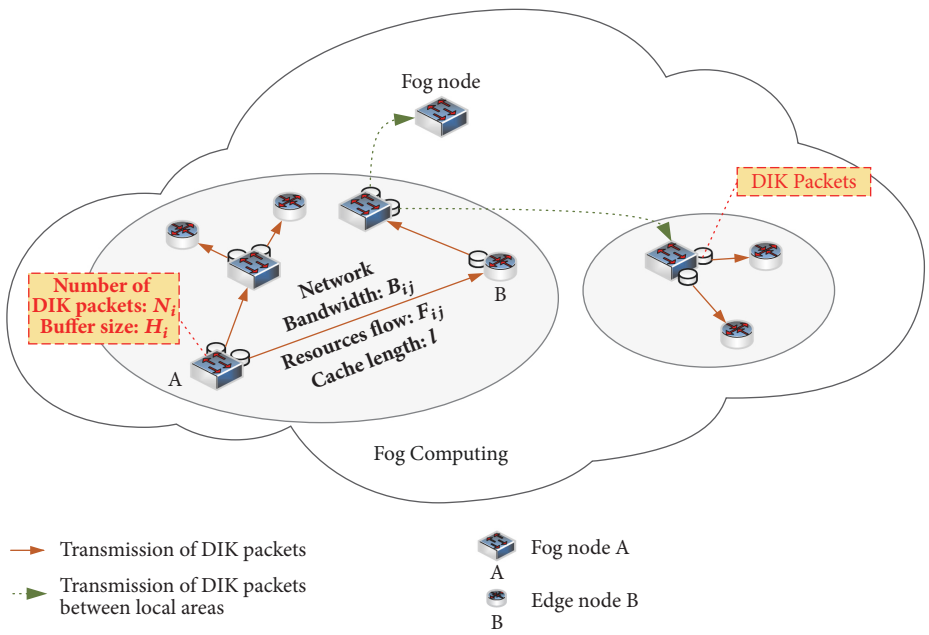

FIGURE 2: The transmission optimization model of typed resources in IoT (IoT_TOM).

KG_{DIK} is of free schema and expresses rich semantic relationships, which is conductive to have a completing mapping towards user requirements described through natural language. KG_{DIK} is a collection of statistical rules summarized from known resources. KG_{DIK} further improves and perfects the semantic relations between the entities on the basis of DG_{DIK} and IG_{DIK} and then forms a semantic network connected by a large number of interactive relationships.

2.3. System Model. We propose a transmission optimization model of typed resources in the Internet of Things, which is called IoT_TOM. Figure 2 shows a network composed of multiple wireless nodes. In this given network, A-B is a communication link between fog nodes A and B. We denote buffer size of each node as H_i. We denote the number of DIK packets to be forwarded as N_i. A DIK packet is a unit of resource made into a single package in forms of a resource combination in our proposed IoT_TOM. We denote the average length of DIK packets as l. We denote bandwidth between A and B as B_{AB}. We denote resource flow capacity on this link as F_{AB}. In practice, the distribution of network resources is not balanced in fog computing applications. The proposed mechanism of converting resource types achieves global optimization to deal with this problem. We use resources forwarding-waiting equilibrium and bandwidth utilization equilibrium to evaluate the performance of the proposed mechanism.

2.3.1. Bandwidth Utilization Equilibrium. Given a wireless network, we consider utilizing bandwidth efficiently. It is necessary to dynamically allocate bandwidth. Therefore, we consider transmitting DIK resource packets between IoT nodes, which aims to obtain enough storage and computing resources through consuming bandwidth. BE_{buse} denotes bandwidth utilization equilibrium. BE_{buse} is related to the network parameters such as bandwidth between IoT nodes and resource flow. Let IR_{ban} denote the bandwidth idle rate; then BE_{buse} is the variance of IR_{ban}. IR_{ban} and BE_{buse} can be illustrated as

$$IR_{ban} = \frac{B_{ij} - F_{ij} * l}{B_{ij}}, \quad \mathrm{i,j} = 1, 2, \ldots, \mathrm{n} \tag{6}$$

$$BE_{buse} = E\left(IR_{ban}{}^2\right) - \left(\left[E\left(IR_{ban}\right)\right]^2\right) \tag{7}$$

In (6), B_{ij} denotes bandwidth between IoT nodes. F_{ij} represents resource flow on the link.

2.3.2. Forwarding-Waiting Equilibrium. Resources forwarding-waiting time reflects the performance of the proposed IoT_TOM, which includes two parameters, forwarding-waiting rate (FR_{wait}) and forwarding-waiting equilibrium (BE_{wequ}). It is important to reduce the traffic load and congestion in the network in the absence of packet loss to balance resource flow. Let $N_i * l$ be the size of resources to be forwarded at node i; then BE_{wequ} is a variance of FR_{wait}. BE_{wequ} and FR_{wait} can be illustrated as

$$FR_{\mathbf{wait}} = \frac{N_i * l}{H_i}, \quad \mathrm{i} = 1, 2, \ldots, \mathrm{n} \tag{8}$$

$$BE_{\mathbf{wequ}} = E\left(FR_{\mathbf{wait}}{}^2\right) - \left(\left[E\left(FR_{wait}\right)\right]^2\right) \tag{9}$$

The cooperation between forwarding-waiting equilibrium and bandwidth utilization equilibrium enables users to utilize IoT resources more efficiently. We denote the objective function for defining utilization of typed resources in the Internet of Things as F, which can be illustrated as

$$\mathrm{F} = \alpha BE_{buse} + \beta BE_{\mathbf{wequ}}, \quad \alpha + \beta = 1 \tag{10}$$

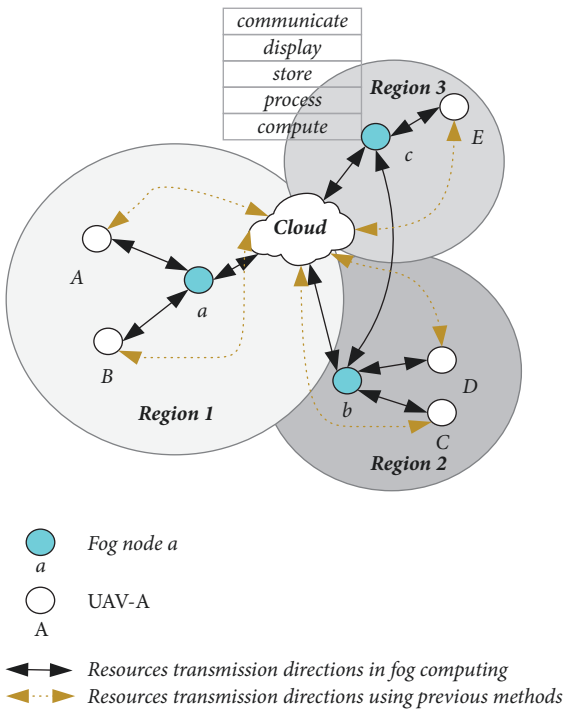

Figure 3: A UAVs courier service example in fog computing.

In (10), α and β, which can be obtained through data training, are parameters that contribute to BE_{buse} and BE_{wequ}, respectively. The smaller F is, the better resource flow capacity distribution performs.

3. Running Example to Provide UAVs Courier Service

Today's unmanned aerial vehicles (UAVs) courier service is a typical IoT application in fog computing. However, it is difficult to address bandwidth bottlenecks and management complexity of a UAV dispatch centre. For example, Figure 3 shows the geographical location distribution of UAVs providing courier services over the air. Dotted directional lines represent resources transmission directions using conventional methods, which may require expensive satellite navigation. The black directional lines display resource transmission directions of providing courier service using a fog computing system.

In an ideal situation, IoT resources without redundancy and inconsistency are transmitted in low user investments. The actual situation may not follow the ideal situation. For example, D_{DIK} resources of (*longitude* and *latitude*) featuring redundancy and inconsistency make the scale of D_{DIK} collection large. Therefore, if storage capacity at node UAV-A is not enough, we need to consume corresponding bandwidth to transmit these D_{DIK} resources to other nodes, which aims to obtain storage resources. We consider converting the resource type to I_{DIK} or K_{DIK} when transmitting these D_{DIK} resources, which aims to change the scale of initial D_{DIK} collection. Table 2 shows partial D_{DIK} resources generated by *UAV-A*. And Table 3 shows partial I_{DIK} resources generated by *UAV-B*.

Table 2: Partial D_{DIK} resources generated by *UAV-A*.

UAV	Unit D_{DIK} (8 bit)	Time range	Times	Space
UAV-A	(*longitude*, *latitude*)	2018.1.1, 9:00–12:00	5000	39.1 Kbit

For *UAV-A*, according to D_{DIK} resources generated from time t_1 (2018.1.1, 9:00) to t_2 (2018.1.1, 9:20), we sum up that two relationships between {*longitude, latitude*} and *time* are $longitude = k_1 * time + b_1$ and $latitude = k_2 * time + b_2$. In fact, we sum up 400 linear relationships between {*longitude, latitude*} and *time* roughly. Conversion from D_{DIK} to I_{DIK} is illustrated as

$$R\left(D_{DIK}\left[longitude, latitude\right]\right) \longrightarrow I_{DIK}\left[k_1 * time + b_1, k_2 * time + b_2\right] \tag{11}$$

We store these I_{DIK} resources converted from D_{DIK} resources in form of I_{DIK} $\{k_1 * time + b_1, k_2 * time + b_2)$, $(k_1, b_1, k_2, b_2, time$ $(t_1, t_2))$, which requires storage space of 32 bits. Then total storage space to store these I_{DIK} resources is $32 * 400 = 12.5$ Kbit.

Table 3 shows I_{DIK} resources generated by *UAV-B*. Through counting and summarizing these rules, we obtain 100 K_{DIK} resources roughly. Conversion from I_{DIK} to K_{DIK} is illustrated as

$$R\left(I_{DIK}\left[Path1, Path2\right]\right) \longrightarrow K_{DIK}\left[rain, down, UAV\text{-}B\right] \tag{12}$$

We store these K_{DIK} resources in form of K_{DIK} (*rain, down, UAV-B*), which requires storage space of 4 bits. Then total storage space to store these K_{DIK} resources is $4 * 250 = 0.97$ Kbit.

Hence, these conversions between resource types reduce the scale of collected resources. In our proposed resource type conversion mechanism, there are at most $27 * 27$ kinds of resource combinations in each node over the network. As shown in Figure 4, we take one resource combination case {(D, D, I), (DG, IG, IG)} as an example. Conversion from {D, D, I} into {I, I, K} aims to obtain storage resources through consuming computing capability. In general, we keep the system in a relatively stable state with storage and computing collaborative adaptation.

Because of complex courier tasks and geographic environment conditions of UAVs, resources distribution is unbalanced. Global optimization of the UAVs wireless network can solve this problem to a large extent. We design a mechanism that minimizes processing cost over network, computation, and storage while maximizing the performance of processing in a business value driven manner, which is illustrated in Section 4. Figure 5 displays resources allocation results in region 1 of the wireless network by using the proposed mechanism. Initial resource combinations on *UAV-A* and *UAV-B* are as follows:

Combination (UAV-A) = $\{(D_{DIK}, D_{DIK}, I_{DIK}), (DG_{DIK}, IG_{DIK}, IG_{DIK})\}$. And the resource scale on *UAV-A* is 12 Kbit, *bandwidth* $(A \rightarrow a)$ is 6 Kpbs, and *buffer size (A)* is 2 Kbit.

Table 3: Partial I_{DIK} resources generated by *UAV-B*.

UAV	Unit I_{DIK} (8 bit)	Time range	Times	Space
UAV-B	*Path 1 (sky, has, rain)* *Path 2 (rain, damage, UAV-B)*	2018.1.1, 9:00–12:00	800	6.3 Kbit

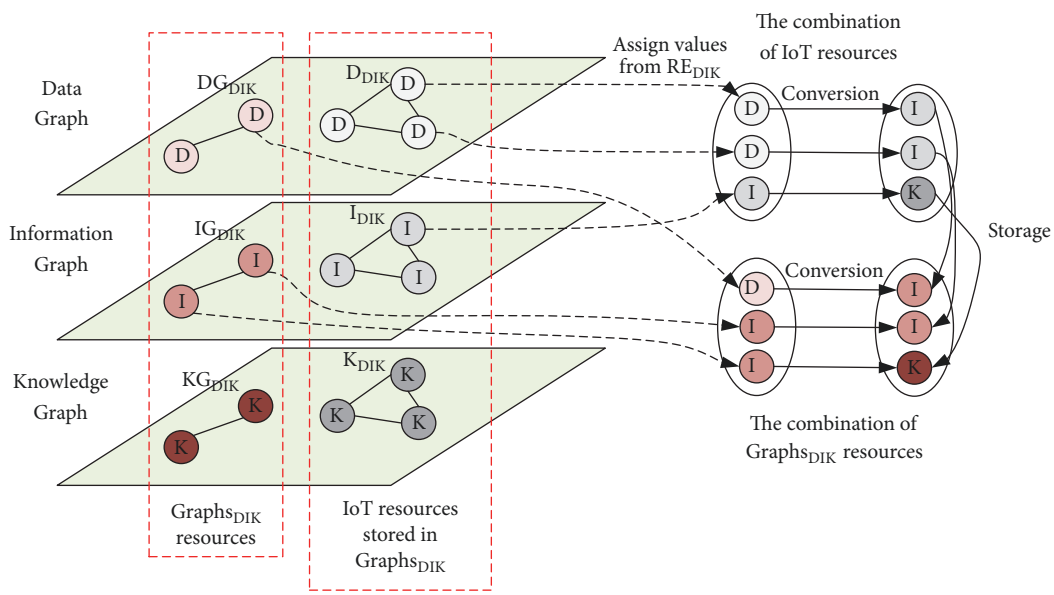

Figure 4: Storing resources according to the resource combination {(D, D, I), (DG, IG, IG)} which is one of the 27∗27 resource combinations.

Combination (UAV-B) = {(D_{DIK}, D_{DIK}, D_{DIK}), (KG_{DIK}, IG_{DIK}, IG_{DIK})}. And the resource scale on *UAV-A* is 26 Kbit, *bandwidth (B → a)* is 6 Kpbs, and *buffer size (B)* is 2 Kbit.

Total storage spaces at node *UAV-B* and node *UAV-B* are 5 Kbit and 10 Kbit, respectively. Thus, we convert resource types and transmit these resources to node a, which is illustrated as follows:

Combination (UAV-A)' = {(D_{DIK}, I_{DIK}, K_{DIK}), (DG_{DIK}, IG_{DIK}, KG_{DIK})}. As shown in Figure 5(a), the resource scale on *UAV-A* is 24 Kbit and *bandwidth (A → a)* = 4 Kpbs and *buffer size (A)* = 3 Kbit.

Combination (UAV-B)' = {(I_{DIK}, I_{DIK}, K_{DIK}), (DG_{DIK}, KG_{DIK}, IG_{DIK})}. As shown in Figure 5(b), the resource scale on *UAV-B* is 23 Kbit and *bandwidth (B → a)* = 3 Kpbs and *buffer size (B)* = 2 Kbit.

However, there are two problems in implementing this idea. First, storage and computation capabilities of IoT devices in fog computing systems are different. Thus, it is necessary to obtain storage capacity by transferring typed resources. Second, both the performance of processing resources and user investments should be considered to improve user investment benefit.

4. Storage and Computing Collaborative Adaptation towards IoT Resources

In fog computing systems, there may be insufficient storage and computation capability at some nodes. We consider transmitting IoT resources to other nodes, aiming to obtain storage and computation resources through consuming bandwidth. Given a wireless network, some nodes consume storage capacity to obtain enough computing resources. Some nodes consume computing capability to obtain enough storage resources. In general, conversion between resource types provides a support for balancing the distribution of network resources in the Internet of Things.

In this section, we describe optimizing processing mechanism of typed resources with collaborative storage and computation adaptation. We define IoT resources and resources on Grpah_{DIK} as follows.

Definition 6 (IoT resources). We define IoT resources as a tuple IR = < IRT, IRS >. IRT is the type set of IoT resources represented by a triad < irt_D, irt_I, irt_K >. IRS is the scale of different kinds of IoT resources represented by a triad < irs_D, irs_I, irs_K >. Each irs denotes the scale of resource in the form of irt.

Definition 7 (resources on Grpah_{DIK}). We define resources on Grpah_{DIK} as a tuple RoG = <RGT, RGS>. RGT is the type set of resources on Grpah_{DIK} represented by a triad < rgt_D, rgt_I, rgt_K >. RGS is the scale of different kinds of resources on Grpah_{DIK} represented by a triad < rgs_D, rgs_I, rgs_K >. Each rgs denotes the scale of resources in the form of rgt.

4.1. Computation of Resource Type Conversion Cost. Considering the disadvantages of conventional resources processing methods, we propose to convert both types of resources in

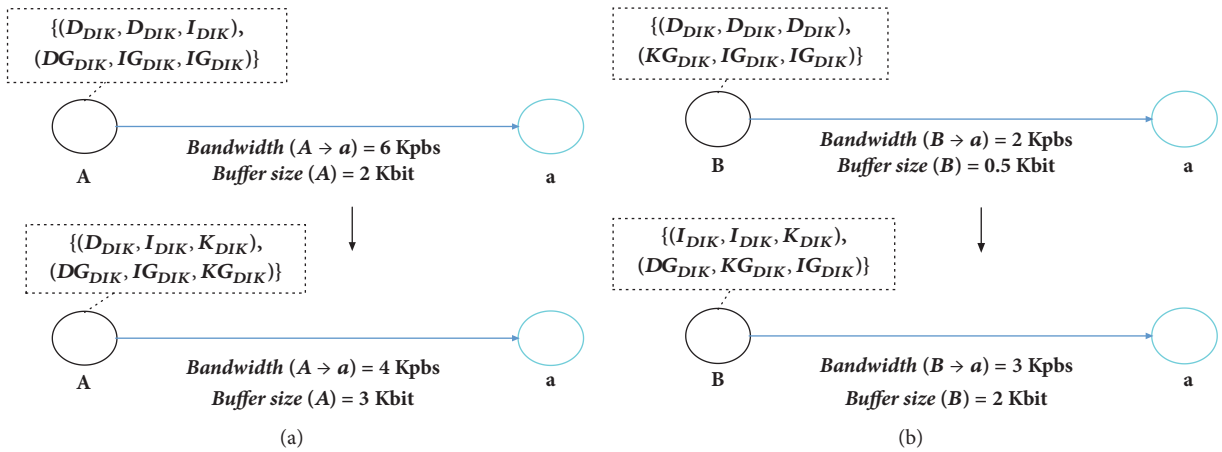

Figure 5: Resource allocation results over region 1 of the UAVs wireless network through using the proposed mechanism.

Table 4: Atomic conversion cost per unit resource in *IR*.

	D_{DIK}	I_{DIK}	K_{DIK}
D_{DIK}	$costCUnitIR_{D\to D}$	$costCUnitIR_{D\to I}$	$costCUnitIR_{D\to K}$
I_{DIK}	$costCUnitIR_{I\to D}$	$costCUnitIR_{I\to I}$	$costCUnitIR_{I\to K}$
K_{DIK}	$costCUnitIR_{K\to D}$	$costCUnitIR_{K\to I}$	$costCUnitIR_{K\to K}$

Table 5: Atomic conversion cost per unit resource in *ROG*.

	D_{DIK}	I_{DIK}	K_{DIK}
D_{DIK}	$costCUnitRG_{D\to D}$	$costCUnitRG_{D\to I}$	$costCUnitRG_{D\to K}$
I_{DIK}	$costCUnitRG_{I\to D}$	$costCUnitRG_{I\to I}$	$costCUnitRG_{I\to K}$
K_{DIK}	$costCUnitRG_{K\to D}$	$costCUnitRG_{K\to I}$	$costCUnitRG_{K\to K}$

IR and RoG. First, we would like to convert the types of resources in IR, which needs to invest corresponding cost. Specifically, we assign values from RE_{DIK} to each element of the type set IRT. Then we use these values to form the combination case IRT' = < irt_D', irt_I', irt_K' >, where irt_D', irt_I', and irt_K' belong to {D_{DIK}, I_{DIK}, K_{DIK}}. As shown in Table 4, we denote $costCUnitIR_{i\to j}$ as the atomic conversion cost per unit resource in IR. Conversion cost of resource types from IR to IRT' can be illustrated as

$$C_{IR} = \sum costCUnitIR_{i\to j} * irs_i, \quad i, j \in \{D, I, K\} \tag{13}$$

Second, we convert the types of resources in RoG. Specifically, we assign values from RE_{DIK} to each resource of the type set RGT of RG. Then we use these values to form the combination case RGT'= < rgt_D, rgt_I, rgt_K >, where rgt_D, rgt_I, and rgt_K belong to {D_{DIK}, I_{DIK}, K_{DIK}}. As shown in Table 5, we denote $costCUnitRG_{i\to j}$ as the atomic type conversion cost per unit resource in RoG. Conversion cost of resource types from RGT' to RoG can be illustrated as

$$C_{RG} = \sum costCUnitRG_{i\to j} * rgs_i, \quad i, j \in \{D, I, K\} \tag{14}$$

4.2. Computation of Cost of Processing IR in RoG. The proposed processing architecture of typed resources is adopted to process IoT resources. Cost of processing *IR* in *RoG* is related to the resource scale. As shown in Table 6, we denote $costPUnitIR_{i\to j}$ as the atomic cost of processing unit D_{DIK}, I_{DIK}, or K_{DIK} resource in corresponding graph resources. Then cost of processing *IR* in *RoG* can be illustrated as

Table 6: Atomic type conversion cost of RE_{DIK}.

	D_{DIK}	I_{DIK}	K_{DIK}
D_{DIK}	$costPUnitIR_{D\to D}$	$costPUnitIR_{D\to I}$	$costPUnitIR_{D\to K}$
I_{DIK}	$costPUnitIR_{I\to D}$	$costPUnitIR_{I\to I}$	$costPUnitIR_{I\to K}$
K_{DIK}	$costPUnitIR_{K\to D}$	$costPUnitIR_{K\to I}$	$costPUnitIR_{K\to K}$

Table 7: Atomic cost of transmitting unit D_{DIK}, I_{DIK}, or K_{DIK} resource.

	D_{DIK}	I_{DIK}	K_{DIK}
C_{transi}	C_{transD}	C_{transI}	C_{transK}

$$C_{pro} = \sum \left(rgs_j + costPUnitIR_{i\to j} * rgs_i'\right) * irs, \quad i, j \in \{D, I, K\} \tag{15}$$

We convert the resource types in hopes of reducing the scale of resources when transferring IR in fog computing systems, aiming to balance resources load in a given wireless network. We assume that bandwidth of some nodes is enough in a period of time. We transmit IR to obtain storage and computation resources of other nodes through consuming bandwidth, aiming to satisfy users' general requirements. As shown in Table 7, let C_{transD}, C_{transI}, and C_{transK} indicate the atomic cost of transmitting unit D_{DIK}, I_{DIK}, or K_{DIK}, respectively. Cost of transmitting IR can be illustrated as

$$C_{trans} = \sum C_{transi} * irs_i, \quad i \in \{D, I, K\} \tag{16}$$

4.3. Calculation of Users' Investment Benefit. We optimize the performance of processing resources through converting the type of resources and rational allocations of infrastructures in Internet of Things. According to previous subsections, we calculate the total cost of processing IR in fog computing

Table 8: Simulation setup.

No.	Input parameters	Setup
1	Number of sensor nodes	10
2	Number of fog nodes	4
3	Area of the wireless network field	500 m ∗ 400 m
4	Frequency band	2.4 GHz
5	Storage capability of each node (bit)	{A, B, C, D, E, F} → {128M, 8M, 256M, 8M, 16M, 4M}
6	Data width of CPU of each node	{A, B, C, D, E, F} → {8 bits, 16 bits, 16 bits, 8 bits, 32 bits, 8bits}
7	Energy model	Battery
8	Sensors	{A, B, C, D, E, F} → {Temperature, Humidity, GPS, Temperature, Photosensitive, GPS} sensors
9	Simulation time	10 minutes

applications by using our proposed mechanism. Total_Cost denotes the total cost of this resources processing service, which can be illustrated as

$$\text{Total\_Cost} = C_{IR} + C_{RG} + C_{pro} + C_{trans} \tag{17}$$

Let IV_{user} denote inquired user investment; then IV_{user} corresponding to each resource combination can be illustrated as

$$\text{IV}_{user} = \phi * \text{Total\_Cost} \tag{18}$$

where ϕ represents the atomic investment that can be obtained through data training. Different investment programs correspond to different benefit ratios (R_e). R_e is used to measure the ratio of performance over investment, which is illustrated as

$$R_e = \frac{C_{trans} + C_{pro}}{\text{IV}_{user}} \tag{19}$$

Then we compare R_e and IV_{user} of each program with R_{e0} and IV_{user0} to determine whether the condition "$R_e > R_{e0}$& IV_{user}<IV_{user0}" is satisfied. Let R_{e0} be equal to the current R_e when R_e is greater than R_{e0}. Algorithm 1 describes the specific process of investment driven resource processing approach.

Input: IR, RoG, IV_{user0}, R_{e0}
Output: The minimum R_{e0}.
  For: each IRT **Do**
    Assign value from RE_{DIK};
    Compute C_{IR};
    Compute C_{RG}; Compute C_{pro};
    Compute C_{trans}; Compute $Total\_Cost$;
    Compute IV_{user}; Compute R_e;
    If ($R_e < R_{e0}$ & $IV_{useri} < IV_{user0}$)
      $R_{e0} = R_e$;
  Return R_{e0}

Algorithm 1: Calculating R_e of each resource combination at all fog nodes.

5. Experiment

The goal of our experiments is to evaluate the proposed processing optimization approach towards typed resources. We first established a small wireless network as our simulation environment with discrete illustration [29]. We observed and analysed user investment benefit (R_e) of six nodes in this system. Then, we measured some necessary network indicators to illustrate effects of using the proposed mechanism to optimize resources processing. Finally, we compared our proposed IoT_TOM with a conventional mechanism.

5.1. Experiment Setup. As shown in Figure 6, we first established a small real wireless network with fourteen nodes. These nodes consisted of different kinds of Arduino MCUs. Six nodes of them were equipped with ESP8266 WI-FI modules and some sensors for collecting environment resources. As shown in Table 8, we set up some necessary input parameters that include senor nodes and energy model [30]. We deployed these nodes in four subareas in an area of 500 m ∗ 400 m.

5.1.1. Experiment Procedure. We aimed to illustrate the feasibility of our approach through providing users resources processing services. The proposed mechanism allowed each node to finish storage, computation, and transmission tasks. We assumed that a user demanded to observe environmental resources of this real wireless network anywhere. For convenience, we assigned values to some necessary parameters in the equations, such as $costCUnitIR_{\text{i}\to\text{j}}$, $costPUnitIR_{i\to j}$, and $costCUnitRG_{i\to j}$. We evaluated the performance of the proposed mechanism in terms of investment benefit and F value. The running time is ten minutes. In our experiment, we used some rules to process collected data into three resource collections of D_{DIK}, I_{DIK}, and K_{DIK}. For example, photosensitive sensors collected D_{DIK} resources of light intensity, which were {2, 4, 8, 16. . . 1024}. We processed these D_{DIK} resources into I_{DIK} resources in forms of 2^{x} (x= [1, 10]).

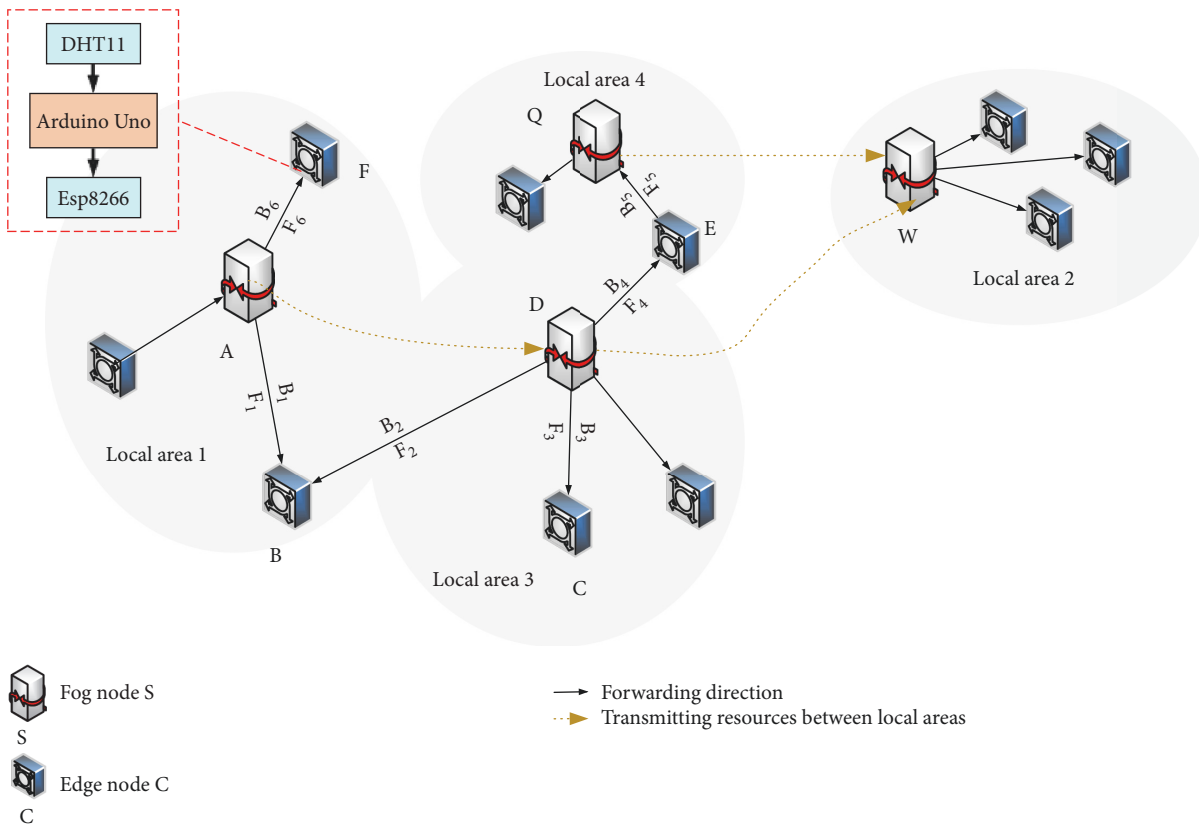

Figure 6: The established simulation environment.

Table 9 shows an example for classifying typed resources in a practical environment.

5.1.2. Statistics Analysis. For convenience, we only considered six resource combinations to illustrate the feasibility of the proposed mechanism. We estimated IV_{user} and R_e after calculating C_{IR}, C_{RG}, C_{trans}, and C_{pro}. As shown in Figure 7, points surrounded by the red box on the R_e curves indicate our recommended resource combinations. In these resource combinations, users are able to acquire resource services of maximum processing performance with minimum investment. In our experiments, the proposed mechanism dynamically recommended different resource combinations for different network environments to meet user demands better. Figure 7 shows that No. 6 of node A, No. 4 of node B, No. 5 of node C, No. 2 of node D, No. 4 of node E, and No. 2 of node F are final recommended resource combinations.

Furthermore, we measured some necessary parameters of the proposed IoT_TOM including F_{ij}, l, and N_i in order to observe the state of this wireless network. As shown in Table 10, FR_{wait} of link A-F is greater than that of other nodes. Storage capacity of node F is 4 Mbit, which is insufficient to store 7 Mbit IoT resources. Therefore node F consumed more bandwidths to obtain enough storage resources compared with other nodes. On the whole, some nodes consumed computing capability to obtain enough storage resources. Some nodes consumed storage capacity to obtain computing resources. Some nodes consumed bandwidth to obtain storage resources. In order to evaluate the performance of IoT_TOM, we estimated BE_{buse} and BE_{wait} which are 0.36 and 0.28, respectively. F value reflects resource flow capacity distribution. In this experiment, we calculated F value according to (10). The calculated F value is 0.34, which means that the proposed mechanism effectively allocated network resources through converting resource types. Table 10 shows that IR_{ban} and FR_{wait} of most nodes are smaller than 0.5, which is not the best result in terms of network parameters intuitively. In order to illustrate advantages of our proposed approach, in next subsection, we compared our proposed mechanism with a conventional mechanism under the same resources scale.

5.2. Comparison with Other Methods. We compared the proposed mechanism with a conventional bandwidth-allocation-based mechanism. We used the performance of traditional bandwidth-allocation-based mechanism as the baseline for comparison. We modelled this simulation on the basis of NSGA-II which is a modified version of NSGA (Nondominated Sorting Genetic Algorithm) released in 1994 used to optimize multiobjectives [31]. The objective function is illustrated as

$$f = F + \sum w_{D_i} + w_{I_i} + w_{K_i}, \quad i \in 1, 2, 3, 4, 5, 6 \tag{20}$$

Table 9: An example for classifying typed resources in a practical environment.

D_{DIK}	I_{DIK}	K_{DIK}
Light intensity {2, 4, 8, 16 . . . 1024}	$2^x(x = [1,10])$	Dim light (<2048)
Temperature {25, 25, 25,. . ., 25}	25 (n = 100 s)	Constant temperature (100 s)

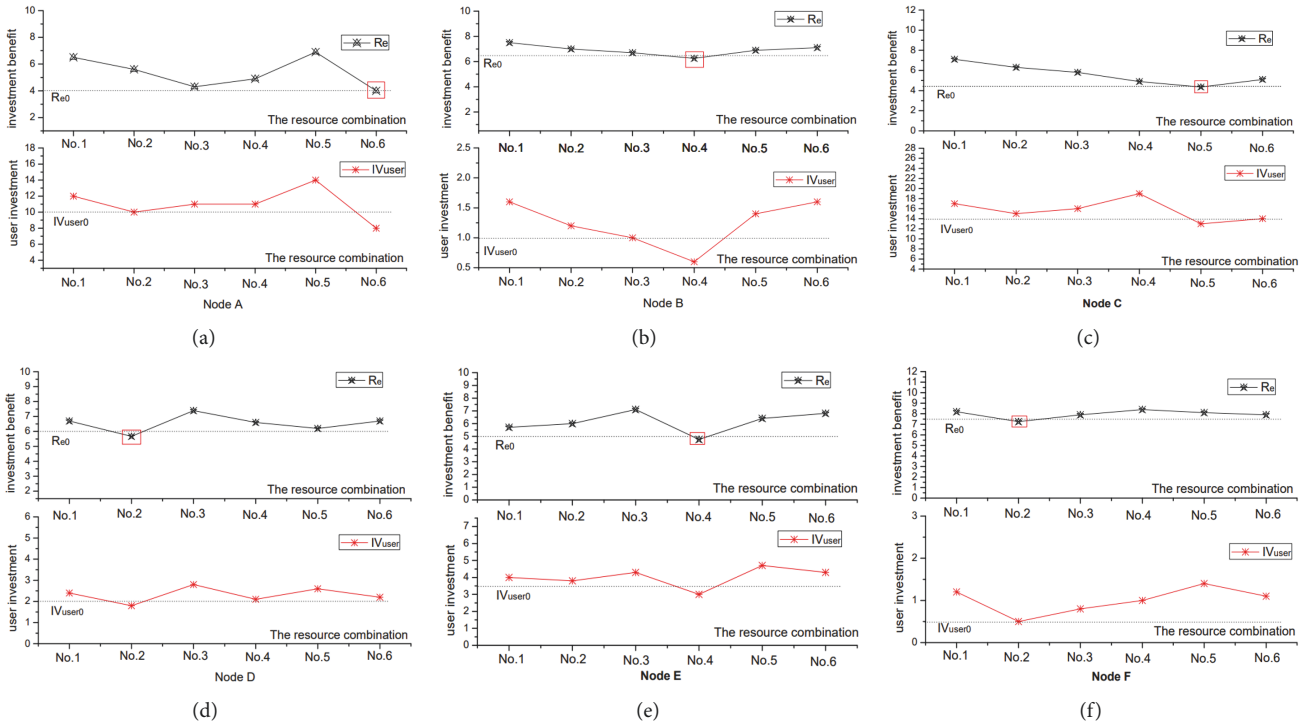

Figure 7: Estimating IV_{user} and R_e values of considered six nodes.

In (20), w_{D_i} is a parameter that contributes to the scale of irs_D. And six restrictive functions are as follows:

$$\sum_{i=1}^{6} w_{D_i} = 1 \tag{21}$$

$$\sum_{i=1}^{6} w_{I_i} = 1 \tag{22}$$

$$\sum_{i=1}^{6} w_{K_i} = 1 \tag{23}$$

$$10 \leq \sum_{i=1}^{6} w_{D_i} * irs_D + \sum_{i=1}^{6} w_{K_i} * irs_I + \sum_{i=1}^{6} w_{K_i} * irs_K \leq 30 \tag{24}$$

$$\alpha + \beta = 1 \tag{25}$$

$$irs_D + irs_I + irs_K = 30 \tag{26}$$

We calculated user investment benefits of these two mechanisms. Figure 8 shows the R_e values per node at corresponding investment to illustrate benefit difference between these two methods. In the simulation, we assigned values to parameters in the equations, such as α and β. But the actual value of each parameter should be obtained through data learning. According to previous experiment results, we find that investment benefit of our proposed method is smaller compared to conventional method under the same investment. The results show that the proposed method performs better when providing resources processing services.

Table 10: Some network parameters of six links on the wireless network.

Links	A-B	B-C	C-D	D-E	D-F	A-F
B_{ij}	2.5	0.8	3.2	0.9	0.7	1.2
H_i	1.4	3.5	1.1	1.0	0.4	2.3
F_{ij}	1.0	5.0	6.0	2.0	4.0	7.0
l	1.0	2.0	3.0	1.0	2.0	3.0
N	2.0	4.0	1.0	3.0	2.0	1.0
IR_{ban}	0.36	0.46	0.40	0.58	0.51	0.32
FR_{wait}	0.47	0.19	0.31	0.28	0.08	0.61

In order to reflect the performance of processing resources in this fog computing application compared with the conventional method, we analysed F value of a given wireless network. Figure 9 shows F value of two mechanisms

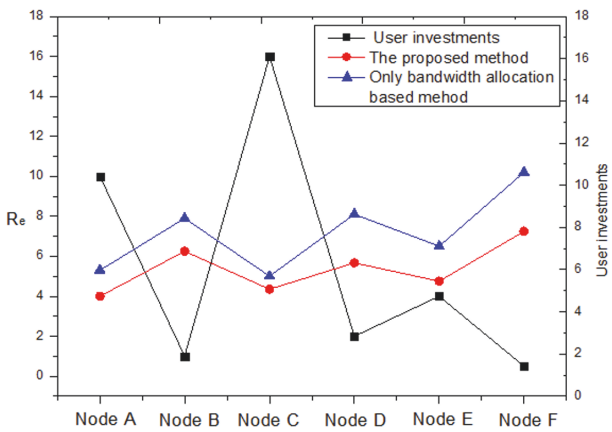

Figure 8: R_e values generated by different methods corresponding to user investment.

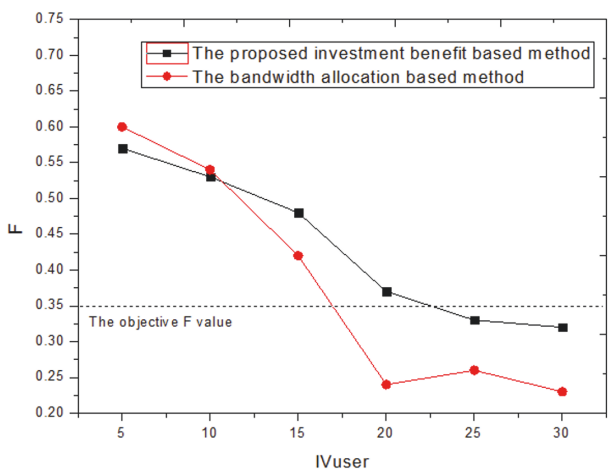

Figure 9: F value of two mechanisms under different investments.

under different investments. The simulation results indicate that the bandwidth-allocation-based method demands fourteen units cost to achieve objective F value. F value of the proposed method is smaller compared with the conventional method.

We used a social network to illustrate the impact of network size on performance of IoT_TOM. The network contains 1133 nodes and 10903 edges, which is from Datatang [32]. We used a sample set of 862 nodes and 1932 edges including five given areas. This simulation was conducted on a Lenovo ThinkPad E431 desktop with a 2.50 GHz Intel core i5 CPU and 8G RAM, running the *pycharm* Edit on Windows 10 operating system. As shown in Figure 10, (6, 12) represents that the number of nodes is 6 and the number of edges is 12. The results show that the bigger the scale of a given wireless network is, the better the distribution of network resources performs in a certain range.

6. Related Work

The dynamic reconstruction of computation and storage resources not only improves the utilization of resources but

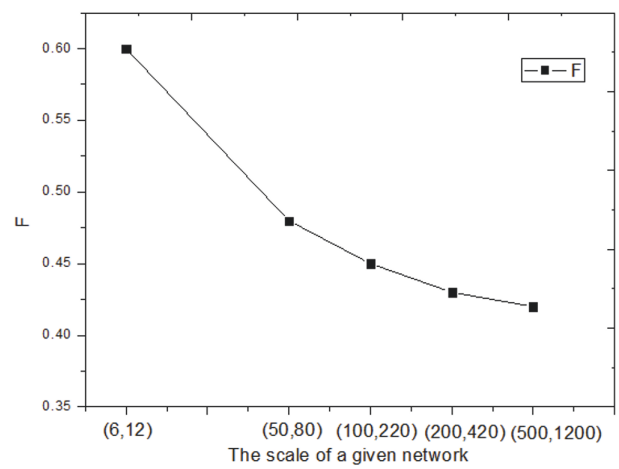

Figure 10: F values of different network scales.

also simplifies management. Some of the workloads that use common resource computing and storage technologies can handle the current cloud system to avoid saturated clouds [33]. However, the cloud computing paradigm is not suitable to solve the problem of unbalanced distributions of network resources in the Internet of Things. We argue that it is necessary to consume bandwidth to transmit resources between IoT nodes, aiming to obtain storage and computation resources of other nodes in order to satisfy user demands. In [34], the authors proposed a distributed multilevel storage (DMLS) model to solve the problems of restricted computation, limited storage, and unstable network. A wireless-based Software Defined Mobile Edge Computing (SDMEC) provided support for autoscaling network storage resource based on the network demand [35]. It is important to dynamically allocate bandwidth in fog computing application. In [36], the authors proposed a bandwidth allocation scheme based on collectable information. An integrated resource allocation method shared the limited bandwidth among multiple wireless body area networks (WBANs) [37]. In [38], the authors studied the tradeoff between the latency and reliability in task offloading to mobile edge computing (MEC). We argue that it is important to achieve global optimization over a wireless network in forms of converting the resource types. The bandwidths of a given wireless network are expensive in fog computing, which makes it a challenge to reduce latency of accessing resources. In [39], the authors proposed an efficient software defined data transmission scheme (ESD-DTS) to maximize data transmission throughput and consistently maintain a flow's latency. T. Taleb et al.'s proposed scheme enforces an autonomic creation of MEC services to allow anywhere-anytime data access with optimum Quality of Experience (QoE) and reduced latency [40]. A hierarchical model composing data, information, knowledge, and wisdom is usually represented in the form of a pyramid [41]. We divide resources in the Internet of Things into data, information, and knowledge. We consider converting resource types to improve user investment benefit in a business value driven manner. We proposed a three-tier resource processing

architecture based on Data Graph, Information Graph, and Knowledge Graph, aiming at optimizing storage, transfer, and processing typed resources.

7. Conclusions and Future Work

Physical services in the Internet of Things are heterogeneous, large-scale, and resource-constrained. Fog computing makes IoT devices more intelligent and saves communication bandwidth. But distributions of network resources in fog computing applications are not balanced. We proposed to convert resource types to deal with this problem. Based on different resource combinations that feature a better user investment benefit, conversions between resource types change the resource scale. In order to satisfy users' general requirements, we consider transmitting typed resources to obtain enough storage and computing resources of other nodes through consuming bandwidth. On the whole, optimized processing of typed resources with storage and computing collaborative adaptation effectively improves the ratio of performance over investment. Meanwhile, processing resources in our proposed three-tier architecture of Data Graph, Information Graph, and Knowledge Graph optimizes spatial and temporal efficiency. The proposed mechanism achieves global optimization over a wireless network to provide cost-effective resource processing services. According to simulation results, users invest minimum costs to obtain maximum performances of processing IoT resources. Generally, our proposed mechanism keeps most fog computing systems in a stable state, which is important for improving the user experience.

In the future, we will continue our study of resources management to improve performance of processing in a business value driven manner and conduct further experiments. We will also explore machine learning approach to optimize processing of resources including data, information, and knowledge.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is supported by Hainan Key Development Program (no. ZDYF2017128), NSFC (no. 61662021, no. 61363007, and no. 61502294), and the IIOT Innovation and Development Special Foundation of Shanghai (no. 2017-GYHLW-01037).

References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[2] M. H. U. Rehman, V. Chang, A. Batool, and T. Y. Wah, "Big data reduction framework for value creation in sustainable enterprises," *International Journal of Information Management*, vol. 36, no. 6, pp. 917–928, 2016.

[3] M. H. Ur Rehman, P. P. Jayaraman, S. Ur Rehman Malik, A. Ur Rehman Khan, and M. M. Gaber, "RedEdge: A novel architecture for big data processing in mobile edge computing environments," *Journal of Sensor and Actuator Networks*, vol. 6, no. 3, article no. 17, 2017.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[5] I. Tomkos, L. Kazovsky, and K.-I. Kitayama, "Next-generation optical access networks: Dynamic bandwidth allocation, resource use optimization, and QoS improvements," *IEEE Network*, vol. 26, no. 2, pp. 4–6, 2012.

[6] M. Satyanarayanan, "The emergence of edge computing," *The Computer Journal*, vol. 50, no. 1, pp. 30–39, 2017.

[7] Z. Ma, Q. Zhao, and J. Huang, "Optimizing bandwidth allocation for heterogeneous traffic in IoT," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 610–621, 2017.

[8] Y. Ito, H. Koga, and K. Iida, "A bandwidth reallocation scheme to improve fairness and link utilization in data center networks," in *Proceedings of the 13th IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops 2016*, pp. 1–4, March 2016.

[9] S. K. Sharma and X. Wang, "Live Data Analytics with Collaborative Edge and Cloud Processing in Wireless IoT Networks," *IEEE Access*, vol. 5, pp. 4621–4635, 2017.

[10] K. Staniec and M. Habrych, "Telecommunication platforms for transmitting sensor data over communication networks—state of the art and challenges," *Sensors*, vol. 16, no. 7, article no. 1113, 2016.

[11] X. Wu, S. Zhang, and A. Ozgur, "STAC: Simultaneous Transmitting and Air Computing in Wireless Data Center Networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 4024–4034, 2016.

[12] C. Yin and C. Wang, "The perturbed compound Poisson risk process with investment and debit interest," *Methodology and Computing in Applied Probability*, vol. 12, no. 3, pp. 391–413, 2010.

[13] C. Yin and Y. Wen, "An extension of Paulsen-Gjessing's risk model with stochastic return on investments," *Insurance: Mathematics and Economics*, vol. 52, no. 3, pp. 469–476, 2013.

[14] P. Li, M. Zhou, and C. Yin, "Optimal reinsurance with both proportional and fixed costs," *Statistics & Probability Letters*, vol. 106, pp. 134–141, 2015.

[15] D. Hua and L. Zaiming, "Optimal reinsurance with both proportional and fixed costs," *Applied Mathematics-A Journal of Chinese Universities*, vol. 27, no. 2, pp. 150–158, 2012.

[16] Y. Wang and C. Yin, "Approximation for the ruin probabilities in a discrete time risk model with dependent risks," *Statistics & Probability Letters*, vol. 80, no. 17-18, pp. 1335–1342, 2010.

[17] W. Sun and L. Peng, "Observer-based robust adaptive control for uncertain stochastic Hamiltonian systems with state and input delays," *Lithuanian Association of Nonlinear Analysts. Nonlinear Analysis: Modelling and Control*, vol. 19, no. 4, pp. 626–645, 2014.

[18] A. Brogi and S. Forti, "QoS-aware deployment of IoT applications through the fog," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1185–1192, 2017.

[19] J. Hong and V. O. Li, "Optimal resource allocation for transmitting network information and data in wireless networks," in *Proceedings of the IEEE International Conference on Communications, ICC '10*, pp. 1–5, 2010.

[20] C. Zins, "Conceptual approaches for defining data, information, and knowledge," *Journal of the Association for Information Science and Technology*, vol. 58, no. 4, pp. 479–493, 2007.

[21] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2181–2187, Austin , Texas, USA, 2015.

[22] J. Pujara, H. Miao, L. Getoor, and W. Cohen, "Knowledge Graph Identification," in *Proceedings of the 12th International Semantic Web Conference, The Semantic Web - ISWC '13*, vol. 8218, pp. 542–557, 2013.

[23] O. Deshpande, D. S. Lamba, M. Tourn et al., "Building, maintaining, and using knowledge bases: A report from the trenches," in *Proceedings of the 2013 ACM SIGMOD Conference on Management of Data, SIGMOD '13*, pp. 1209–1220, 2013.

[24] Y. Duan, L. Shao, G. Hu, Z. Zhou, Q. Zou, and Z. Lin, "Specifying architecture of knowledge graph with data graph, information graph, knowledge graph and wisdom graph," in *Proceedings of the 15th IEEE/ACIS International Conference on Software Engineering Research, Management and Applications, SERA 2017*, pp. 327–332, gbr, June 2017.

[25] L. Shao, Y. Duan, X. Sun, Q. Zou, R. Jing, and J. Lin, "Bidirectional value driven design between economical planning and technical im-plementation based on data graph, information graph and knowledge graph," in *Proceedings of the IEEE International Conference on Software Engineering Research, Management and Applications*, pp. 339–344, 2017.

[26] L. Shao, Y. Duan, X. Sun, H. Gao, D. Zhu, and W. Miao, "Answering Who/When, What, How, Why through Constructing Data Graph, Information Graph, Knowledge Graph and Wisdom Graph," in *Proceedings of the The 29th International Conference on Software Engineering and Knowledge Engineering*, pp. 1–6.

[27] L. Shao, Y. Duan, X. Sun, Q. Zou, R. Jing, and J. Lin, "Bidirectional value driven design between economical planning and technical implementation based on data graph, information graph and knowledge graph," in *Proceedings of the 15th IEEE/ACIS International Conference on Software Engineering Research, Management and Applications, SERA 2017*, pp. 339–344, gbr, June 2017.

[28] D. Kong, L. Liu, and Y. Wu, "Best approximation and fixed-point theorems for discontinuous increasing maps in Banach lattices," *Fixed Point Theory and Applications*, vol. 2014, no. 1, pp. 1–10, 2014.

[29] Y. Dong, L. Song, M. Wang, and Y. Xu, "Combined-penalized likelihood estimations with a diverging number of parameters," *Journal of Applied Statistics*, vol. 41, no. 6, pp. 1274–1285, 2014.

[30] W. Sun, K. Wang, C. Nie, and X. Xie, "Energy-based controller design of stochastic magnetic levitation system," *Mathematical Problems in Engineering*, vol. 2017, no. 3, pp. 1–6, 2017.

[31] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Parallel Problem Solving from Nature PPSN VI*, vol. 1917 of *Lecture Notes in Computer Science*, pp. 849–858, Springer, Berlin, Germany, 2000.

[32] Datatang, http://www.datatang.com/data/796.

[33] S. Alonso-Monsalve, F. Garcia-Carballeira, and A. Calderon, "Fog computing through public-resource computing and storage," in *Proceedings of the 2nd International Conference on Fog and Mobile Edge Computing, FMEC 2017*, pp. 81–87, May 2017.

[34] J. Xing, H. Dai, and Z. Yu, "A distributed multi-level model with dynamic replacement for the storage of smart edge computing," *Journal of Systems Architecture*, vol. 83, pp. 1–11, 2018.

[35] J. Al-Badarneh, Y. Jararweh, M. Al-Ayyoub, M. Al-Smadi, and R. Fontes, "Software Defined Storage for cooperative Mobile Edge Computing systems," in *Proceedings of the 4th International Conference on Software Defined Systems, SDS 2017*, pp. 174–179, esp, May 2017.

[36] Y. Ito, H. Koga, and K. Iida, "A bandwidth allocation scheme to meet flow requirements in mobile edge computing," in *Proceedings of the 2017 IEEE 6th International Conference on Cloud Networking (CloudNet)*, pp. 114–5118, September 2017.

[37] D.-R. Chen, "A QoS Bandwidth Allocation Method for Coexistence of Wireless Body Area Networks," in *Proceedings of the 25th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2017*, pp. 251–254, rus, March 2017.

[38] J. Liu and Q. Zhang, "Offloading Schemes in Mobile Edge Computing for Ultra-Reliable Low Latency Communications," *IEEE Access*, vol. 6, pp. 12825–12837, 2018.

[39] E. Kim and S. Kim, "An Efficient Software Defined Data Transmission Scheme based on Mobile Edge Computing for the Massive IoT Environment," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 2, pp. 974–987, 2018.

[40] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 38–43, 2017.

[41] J. Rowley, "The wisdom hierarchy: Representations of the DIKW hierarchy," *Journal of Information Science*, vol. 33, no. 2, pp. 163–180, 2007.