Computer Networks 151 (2019) 114–123

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Aggregated multi-attribute query processing in edge computing for industrial IoT applications

Xiaocui Li[a], Zhangbing Zhou[a,b,*], Junqi Guo[c], Shangguang Wang[d], Junsheng Zhang[e]

[a] School of Information Engineering, China University of Geosciences (Beijing), Beijing 100083, China
[b] Computer Science Department, TELECOM SudParis, Evry 91001, France
[c] College of Information Science and Technology, Beijing Normal University, Beijing 100875, China
[d] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[e] Institute of Scientific and Technical Information of China, Beijing 100038, China

ARTICLE INFO

Article history:
Received 24 July 2018
Revised 6 December 2018
Accepted 25 January 2019
Available online 29 January 2019

Keywords:
Multi-attribute aggregation query
Energy-aware IR-tree
Edge node routing graph
Edge computing

ABSTRACT

The popularity of smart things constructs sensing networks for the Internet of Things (IoT), and promotes intelligent decision-makings to support industrial IoT applications, where multi-attribute query processing is an essential ingredient. Considering the huge number of smart things and large-scale of the network, traditional query processing mechanisms may not be applicable, since they mostly depend on a centralized index tree structure. To remedy this issue, this article proposes a multi-attribute aggregation query mechanism in the context of edge computing, where an energy-aware IR-tree is constructed to process query processing in single edge networks, while an edge node routing graph is established to facilitate query processing for marginal smart things contained in contiguous edge networks. This decentralized and localized strategy has shown its efficiency and applicability of query processing in IoT sensing networks. Experimental evaluation results demonstrate that this technique performs better than the rivals in reducing the traffic and energy consumption of the network.

1. Introduction

With the popularity of smart things being ubiquitously deployed, adopting smart things to facilitate industrial applications becomes a reality nowadays. Intuitively, smart things in the Internet of Things (IoT) include sensors, actuators, and smart embedded devices [1], and they can provide sensory data to promote the validity and applicability of a proper decision-making. Due to the fact that smart things are mostly scarce in their computational, communication, and energy resources, aggregating sensory data of certain IoT smart things, and functional combination and collaboration [3], requires to reduce the amount/size of data packets to be transmitted in the network, and thus, to decrease the energy consumption. With the swift growth of the number of smart things being deployed in tremendous fields, traditional centralized sensory data gathering mechanisms through constructing routing trees may not be an appropriate strategy, when sensory data of smart things located within a certain sub-region are interested. Instead, sensory data should be gathered, and processed whenever possible, in a localized fashion, while only the result should be aggregated and routed to the centre for further exploration. We argue that this strategy is proper, especially when sensory data, like multimedia data, are large in volume. Due to this concern, edge computing [2,4] has been proposed in recent years as the complement of cloud computing [32], where industrial IoT applications should be processed in a distributed and localized fashion as much as possible [5]. It is worth noting that sensory data query processing is an essential ingredient of typical industrial IoT applications [6]. Considering the functional diversity of smart things and the complexity of potential events to be studied, this article aims to explore the query processing, where various kinds of smart things contained in a certain sub-region in an IoT sensing network [7] are necessary to cooperate and collaborate for environment monitoring and potential event detection. Taking the assumption that the kind of smart things corresponds to a certain sensing attribute into consideration, an aggregated multi-attribute query processing mechanism is essential to support industrial IoT applications, where edge computing is applied to promote sensory data processing and aggregation at the network edge.

* Corresponding author at: School of Information Engineering, China University of Geosciences (Beijing), Beijing 100083, China.

E-mail addresses: lixiaocui@cugb.edu.cn (X. Li), zbzhou@cugb.edu.cn (Z. Zhou), guojunqi@bnu.edu.cn (J. Guo), wangjunping@bupt.edu.cn (S. Wang), zhangjs@istic.ac.cn (J. Zhang).

https://doi.org/10.1016/j.comnet.2019.01.022
1389-1286/

Traditional techniques have been developed to study the multi-attribute query processing. Generally, an index tree, like an R-tree, is built to manage smart things distributed in a network. Queries are processed leveraging this index tree, where the result can be (i) a single object, which can satisfy certain spatial and multi-attribute constraints [8–12], or (ii) a set of contiguous objects, which can collectively satisfy certain constraints [13–15]. Since objects may be unevenly distributed in the network, authors adopt proper mechanisms for handling objects contained in dense and sparse sub-regions. Objects in dense sub-regions should be prone to be recommended, since they can have more counterparts to be replaced when found improper [16]. Note that objects in certain directions may be more appropriate in certain settings, and thus, a direction-aware spatial keyword query method is proposed to satisfy direction-aware requirements [17]. Generally, these techniques construct a single index tree to support the query of spatial objects, where a single or multiple attribute(s) is/are to be examined. This centralized query processing strategy may not be appropriate when an IoT sensing network is large in scale, and things are huge in quantity. Besides, the network greenness requires to reduce the traffic and energy consumption of the network. Consequently, sensory data should be processed in a localized and distributed fashion when possible. In recent years, techniques have been developed to enable the search of IoT things, where a single thing is mostly interested [18,19]. Other techniques explore the network communication topology [20], an effective collection [21], management [22], and aggregation [23] of sensory data, a load-balancing routing [24], and the prolonging of network lifetime [25,26]. To the best of our knowledge, a distributed and localized mechanism has not been explored extensively to support the multi-attribute query processing in IoT sensing networks.

To address this challenge, this article proposes a *M*ulti-attribute *A*ggregation Query (*MAQ*) processing technique in edge computing. In this context, the network is divided into sub-regions, where these sub-regions, corresponding to the regions of edge networks, are regulated by respective edge nodes. Generally, an edge network can have one edge node. Queries are processed firstly at the network edge by edge nodes, and the results are aggregated and routed to the centre afterwards. It is worth emphasising that smart things regulated by contiguous edge nodes may satisfy the requirement in a collective fashion, which requires the examination of sensory data provided by marginal smart things contained in contiguous edge networks. Major contributions of this article are summarized as follows:

- *Query processing in single edge networks*. An *E*nergy *IR*-tree (i.e., *EIR*-tree) is constructed to facilitate the query processing of smart things contained in a single edge network. Besides the inverted files specified upon the R-tree for indexing attributes of smart things, an energy factor is adopted to estimate the amount of energy consumption with respect to the number and density of smart things in certain sub-regions.
- *Query processing for marginal smart things in contiguous edge networks*. Considering the amount of sensory data generated by smart things in the marginal sub-region of contiguous edge networks, a packet transmission graph is constructed upon edge nodes, in order to decrease the network traffic. Sensory data packets are transmitted between edge nodes, only when these sensory data are examined highly possible to benefit the query answering. The results with respect to independent and marginal edge networks are assembled and aggregated for processing this query.

Extensive experiments are conducted to evaluate the efficiency and applicability of our technique. The results demonstrate that this technique performs better than the rivals in reducing the network traffic and energy consumption of smart things.

The rest of this article is organized as follows. Section 2 introduces relevant concepts and the energy model, which are used in our query. Section 3 introduces the query processing which is applied to single edge networks. Section 4 presents sensory data routing mechanism in edge nodes and the query mechanism in marginal edge networks. Section 5 shows the implementation and evaluates the approach developed in this article. Section 6 reviews and discusses related techniques. Finally, Section 7 concludes this work.

2. Preliminaries: concepts and energy model

This section presents relevant concepts and the energy consumption model.

2.1. Concept definition

In edge computing, a network region can be represented by disjoint edge networks, where an edge node is responsible for managing smart things in the respective edge network. Edge nodes can be (i) a *super* smart thing, which can have more computational, communication, and energy resources than *ordinary* smart things, or (ii) an ordinary smart thing. In this setting, smart things should take the role of edge nodes in a rotation manner for instance, to ensure the overall energy consumption of smart things as balanced somehow at the network level as possible. A marginal edge network of sensory data routing for contiguous edge nodes is defined as follows:

Def. 1. Edge Node Data Routing Network. An edge node data routing network is defined as a tuple $g = (Dgn, Rlt, Cst)$, where:

- *Dgn* is the set of edge nodes contained in marginal edge networks.
- *Rlt* is the set of sensory data routing relationships between contiguous edge nodes.
- *Cst* is the set of sensory data routing cost for contiguous edge nodes, corresponding to the weights specified on the edges in *Rlt*.

In marginal edge networks, by means of edge computing, *g.Dgn* is responsible for data interaction transmission, which is only the result of localization processing. An edge node data routing network is represented in terms of a weighted directed graph, where the vertexes are edge nodes and the weights on the directed edges represent sensory data routing cost for contiguous edge nodes. The edge node routing graph is stored in the form of an adjacency matrix, which specifies the sensory data forwarding strategy between edge nodes.

Considering the diversity of smart things and the complexity of applications to be supported, various kinds of attributes are sensed by smart things. Without loss of generality and for simplicity, in this article we assume that a smart thing is relevant to a single kind of attribute. A query can be defined as follows:

Def. 2. Multi-Attribute Aggregation Query. A multi-attribute aggregation query is defined as a tuple $q = (Rgn, Kd, Cst)$, where:

- $Rgn = (x, y, wdt, hgt)$ is a regular region of q, such that x and y are the top-left x- or y-coordinate, and wdt and hgt are the width and height of query region.
- $Kd = \{k_1, k_2, \ldots, k_m\}$ is a set of attributes that are interested by q.
- *Cst* is a set of constraints defined upon *Kd* to specify the conditions that should be satisfied by neighboring smart things in a collective fashion.

Generally, *q.Rgn* is a rectangle and smart things are deployed in a two-dimensional network space. *q.Rgn* may be contained by an edge network, or by multiple contiguous edge networks. A sample

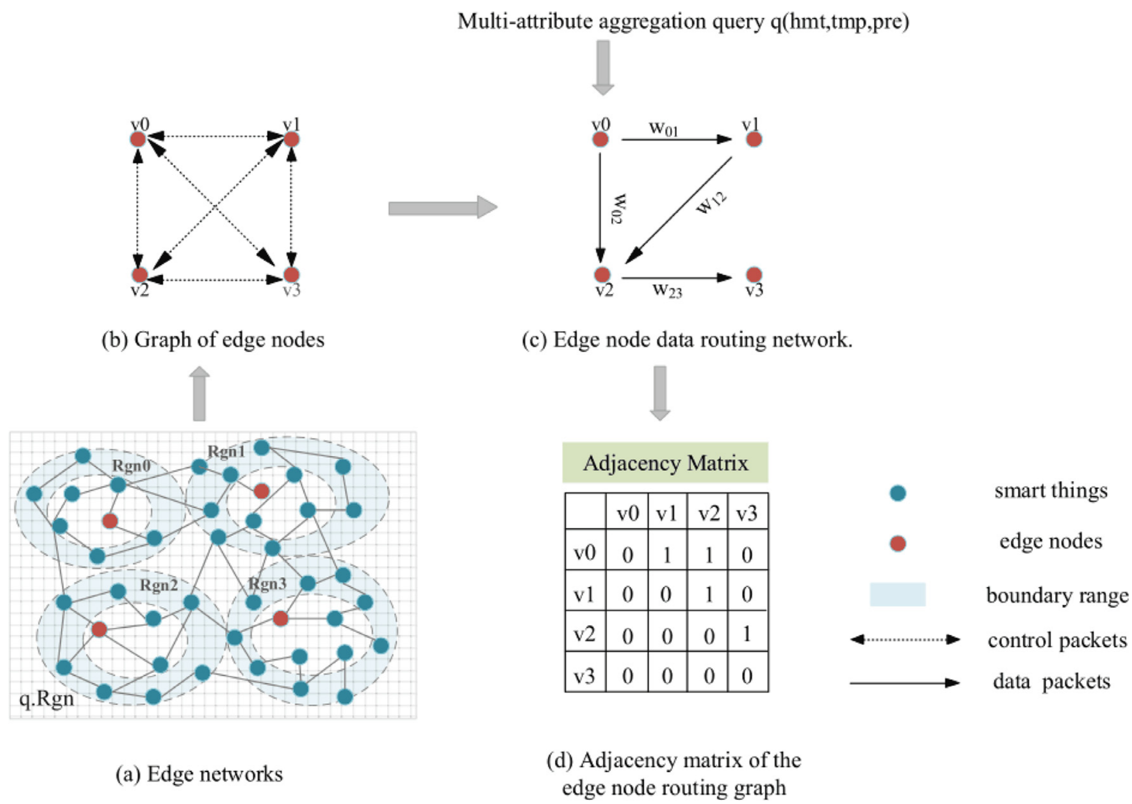


Fig. 1. A sample multi-attribute aggregation query network.

multi-attribute aggregation query network is presented as follows to illustrate the relationship between a multi-attribute aggregation query and the edge node data routing network:

A multi-attribute aggregation query q is specified in terms of three attributes *hmt, tmp* and *prs*, representing humidity, temperature and pressure, respectively. In Fig. 1(a), four edge networks (e.g., Rgn_0, Rgn_1, Rgn_2, Rgn_3) is displayed and $q.Rgn$ are determined. Besides, the boundary range of data communication between edge networks is identified. In Fig. 1(b), edge networks are represented in terms of a graph, where vertexes are edge nodes in the corresponding edge networks (e.g., v_0, v_1, v_2, v_3). Note that edge nodes are responsible for the propagation and localization of the query. Prior to data transmission, neighboring edge nodes send control packets to determine whether sensory data exchanges in-between are necessary or not. This strategy should decrease sensory data packets forwarding between neighboring edge nodes and thus, it can reduce the energy consumption of the query upon marginal edge networks. Subsequently, the edge node data routing network is built and represented as an adjacency matrix, as shown in Fig. 1(c), and (d), respectively, where the value is either 0 or 1. Note that 0 represents no data packets to be sent between edge nodes, 1 represents data packet to be sent between edge nodes. A query is typically injected into the network from an edge node, and this query should be processed by a single edge node, or through the collaboration of multiple edge nodes to achieve the multi-attribute aggregation in single edge network and marginal edge network.

2.2. Energy model

This article applies the first-order radio model [27], which has been widely adopted in wireless sensor networks (WSNs), to calculate the energy consumption between smart things, since sensor nodes in WSNs are indeed a typical kind of smart things, and WSNs can be regarded as a special type of IoT sensing networks. Parameters of this energy model are presented in Table 1.

Specifically, the energy consumption to transmit a k bit data packet with a distance d are denoted as $E_{Tx}(k, d)$, and the energy consumption to receive a k bit data packet are denoted as $E_{Rx}(k)$, which can be calculated as follows:

$$E_{Tx}(k, d) = E_{elec} \times k + \epsilon_{amp} \times k \times d^n \quad (1)$$

$$E_{Rx}(k) = E_{elec} \times k \quad (2)$$

Note that E_{elec} is the constant of energy consumption for transmission and receiver electronics, and ϵ_{amp} is the constant of transmission amplifier. In the course of transmitting a packet of k bits from one thing to another, the energy consumption $E_{ij}(k)$ is calculated as follows:

$$E_{ij}(k) = E_{Tx}(k, d) + E_{Rx}(k) \quad (3)$$

where the parameter d represents the distance between one smart thing nd_i and another nd_j. $E_{ij}(k)$ is assumed the same as $E_{ji}(k)$ for smart things and edge nodes. The parameter n of the attenuation index for packet transmission depends on the surrounding environment. Generally, when smart things are barrier-free for forwarding data packets, n is set to 2. Otherwise, n is set to a value between 3 to 5.

3. Single edge network query processing

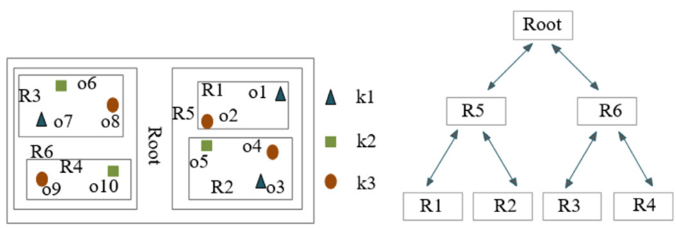
Leveraging an IR-tree [10], this section constructs an Energy *IR*-tree (*EIR*-tree) to support the multi-attribute query processing in a single edge network.

3.1. EIR-Tree Construction

Before presenting the construction of our *EIR*-tree, we briefly introduce the IR-tree as the background. Generally, a node in an

Table 1
Parameters in the energy model.

Name	Description
E_{elec}	Energy consumption constant of the transmit and receiver electronics.
ϵ_{amp}	Energy consumption constant of the transmit amplifier.
k	The number of bits in one packet.
d	The distance of transmission.
n	The attenuation index of transmission.
$E_{Tx}(k,d)$	The energy consumption to transmit a k bit packet with a distance d.
$E_{Rx}(k)$	The energy consumption to receive a k bit packet.
$E_{ij}(k)$	Energy consumption for transmitting a k bit packet from a smart thing SmT_i to a neighboring smart thing SmT_j.



(a) Smart things and bounding rectangles (b) *EIR*-tree

Fig. 2. Query processing of the attribute k_2 upon the *EIR*-tree.

Table 2
Sample inverted file for the *EIR*-tree as shown in Fig. 2.

IF_Node	k_1	k_2	k_3
R1	$(1,o_1)$	null	$(1,o_2)$
R2	$(1,o_3)$	$(1,o_5)$	$(1,o_4)$
R3	$(1,o_7)$	$(1,o_6)$	$(1,o_8)$
R4	null	$(1,o_{10})$	$(1,o_9)$
R5	(2,*R1, R2*)	(1,*R2*)	(2,*R1, R2*)
R6	(1,*R3*)	(2,*R3, R4*)	(2,*R3, R4*)
Root	(3,*R5, R6*)	(3,*R5, R6*)	(4,*R5, R6*)

IR-tree can be represented as a tuple (*id, mbr, O*), where (i) *id* is an identifier of this node, (ii) *mbr* is the *M*inimum *B*oundary *R*egion (*MBR*) covered by this node, and (iii) *O* refers to the set of objects contained in *mbr*. A node has a pointer to an inverted file, and attributes sensed by objects in *O* are recorded in this inverted file. Leveraging the IR-tree structure, an *EIR*-tree is constructed as presented by Algorithm 1, where the energy consumed for sensory data packets transmission between smart things and edge nodes is considered.

Algorithm 1 EIRTreeConstruction.

Require:
- MBR_{set} : the set of leaf nodes in an IR-tree

Ensure:
- tr : the root node of constructed *EIR*-tree

```
1: leaf nodes ← nodes in MBR_set
2: num ← the number of nodes in MBR_set
3: while num > 1 do
4:    for nd_i ∈ MBR_set do
5:       E(k) ← calculated by Eq. (3)
6:    end for
7:    tn ← nd_1 and nd_2 with the biggest E(k) in the MBR_set
8:    tn.mbr ← covered by nd_1 and nd_2
9:    tn.O ← contained by nd_1.O and nd_2.O
10:   MBR_set ← MBR_set - {nd_1, nd_2}
11:   MBR_set ← MBR_set ∪ {tn}
12:   num ← the number of nodes in MBR_set
13: end while
14: tr ← MBR_set
```

As presented by Algorithm 1, based on the IR-tree structure, we obtain the *mbr* collection that covers smart things. These smart things in this collection serve as the leaf nodes of our *EIR*-tree (line 1). For instance, in Fig. 2(a), for a single edge network, ten smart things (e.g., o_1, o_2, ..., o_{10}) are displayed. Meanwhile, according to the spatial division of [10], leaf nodes (e.g., R_1, R_2, R_3 and R_4) are identified. In addition, we deploy three attributes denoted as k_1, k_2 and k_3, which are represented in terms of triangle, square and circular, respectively. An inverted file is appended to represent the attributes sensed by tree nodes (leaf nodes and non-leaf nodes) (denoted k), the frequency of k, and the list of tree nodes or smart things which have the attribute k, where each tree node containing smart things as an item in the inverted file are is described by Table 2 (e.g., R_1, R_2, R_3 and R_4).

In this article, high energy consumption means that the intensity of data packets exchange is relatively strong. When constructing an index tree, energy consumption is considered as an essential factor, and a fusion strategy of energy consumption is adopted. Specifically, given a set of tree nodes, we calculate the energy consumption of each tree node in the collection MBR_{set} (lines 4–6). Here, the $E(k)$ represents the energy consumption of collecting sensory data in each tree node, which is calculated by Eq. (3) (line 5).

For instance, the weight of the tree node R_1, is computed as follows:

$$W_{R1}(k) = 2 \times E_{elec} \times k + \epsilon_{amp} \times k \times d^n_{o_1,o_2} \tag{4}$$

Note that a certain tree node in MBR_{set} has a relatively high energy consumption, which means that the intensity of sensory data exchange is large. Such tree nodes are selected as a merged new tree node according to their energy consumption. At each merging step, two tree nodes with the biggest weight are selected to be merged (lines 7–11). The *EIR*-tree is constructed through merging tree nodes from bottom to top, until the root node has been established (line 14). An example of constructed *EIR*-tree is shown in Fig. 2(b).

3.2. Query processing in single edge networks

In general, the single edge network query processing is performed by traversing *EIR*-tree, and the inverted file is used to check whether there is an attribute of interest in the edge network. By eliminating smart things that are not in the scope of interest for the query as early and prompt as possible, the query can avoid processing non-target things.

Leveraging the *EIR*-tree, Algorithm 2 presents the procedure of querying smart things with a set of attributes. In the similar fashion, the query q in each single edge network is executed. Moreover, the relevant definition of the involved parameters in the query is presented in Section 2.1. In general, the query starts at the root node of *EIR*-tree (line 2). When the inverted file of one tree node tn contains certain attribute, the query is propagated to the tree node tn's children (lines 3–7). This procedure iterates until (i) the inverted file of a non-leaf node does not contain any attribute, or (ii) the leaf node is reached. So far, we obtain a set that consists of collections, where each collection is associated with an attribute (lines 8–9). Consequently, via iteration, the result set that satisfies the query specification is constructed (lines 1–12).

Algorithm 2 IndexQuery.

Require:
- q : the tuple (Rgn, Kd, Cst)
- tn : the tree node to launch the query, and initially set to the root node of *EIR*-tree

Ensure:
- Rst_{set} : a set of collections, where each collection is associated with an attribute

1: $O_{set} \leftarrow \emptyset$
2: **if** $tn \neq NULL$ **then**
3:   **if** $\exists$ attribute $k_i \in Kd$ in tn.inverted file **then**
4:     **if** tn.hasChild() **then**
5:       IndexQuery(q, tn.leftChild)
6:       IndexQuery(q, tn.rightChild)
7:     **else**
8:       $O_{set} \leftarrow tn$.getFilterObject(Cst)
9:       $Rst_{set} \leftarrow Rst_{set} \cup O_{set}$
10:     **end if**
11:   **end if**
12: **end if**

For instance, smart things with the attribute of k_2 are to be retrieved. Based on the example of *EIR*-tree as shown in Fig. 2(b), the root node contains the attribute k_2 from Table 2, and the child nodes R_5 and R_6 contain k_2 as well. Therefore, the query is propagated to the non-leaf node R_5 and R_6. We also note that R_2, a child of R_5, contains k_2, while another child R_1 does not. At the same time, R_3 and R_4, the children of R_6, contain k_2. As the result, the query is propagated to the leaf nodes R_2, R_3, and R_4. Specifically, from Table 2, o_5, o_6 and o_{10} correspond to the smart things for R_2, R_3 and R_4, respectively, contain attribute k_2.

4. Marginal edge network query processing

To facilitate query processing leveraging smart things located in the marginal sub-regions of contiguous edge networks, this section constructs a packet transmission graph for specifying the sensory data forwarding strategy between edge nodes, and sensory data are gathered and routed along the paths in this graph for examining the fact that whether queries can be answered by these smart things in marginal edge networks or not.

4.1. Sensory data routing cost calculation for contiguous edge nodes

A parameter is used to denote the percentage of boundary distance λ, which represents a range about the ratio of the distance between a smart thing and corresponding edge node to the length of the current region, to specify the number of smart things which require to transmit sensory data transmission. Generally, given the coordinates of a smart thing P_0 (x_0, y_0) and an edge node P_1 (x_1, y_1), they have the following relationship:

$$JS = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \div rSide \tag{5}$$

where *rSide* refers to the size of the region in which the edge node is located. *JS* is used to judge the spatial scope of transmitted data. If the value *JS* is not more than the specified standard parameter λ, this means that the smart thing P_0 is within the scope of interactive data.

The presentation of Eq. (5) is to specify the number of smart things that need to transmit their sensory data. Defining boundary data transmission regulations, we can obtain the transmission data at the boundary which is delivered to the corresponding edge node. Edge nodes are responsible for sensory data transmission. Thereafter, we can use Eq. (3) to calculate the communication cost between edge nodes.

Algorithm 3 presents the cost calculation procedure for transmitting sensory data packets between edge nodes. Based on query results of single edge networks, we calculate the energy consumption of communication between edge nodes. For each single edge network, we obtain the result set of its region by Algorithm 2. When a result set in a certain single edge network exists, the boundary data of this region is performed (lines 4–15). Based on this result set, we acquire the negotiated transmission smart thing data from an edge node to its neighbors within the specified parameter of percentage of boundary distance λ and *JS* (lines 10–12). The amount of data transmission between edge nodes is identified by localization processing, which consists of collections of data smart thing identified by each attribute (line 14). The distance between two edge nodes gn_i and gn_j is defined as a 2-d Euclidean distance (line 18). Finally, the cost of sensory data transmission between edge nodes is calculated by Eq. (3) (line 19), and the result of sensory data routing cost for contiguous edge nodes is stored in the form of an adjacency matrix (line 20).

Algorithm 3 CostCalculation.

Require:
- λ : a parameter of percentage for boundary distance
- num : the number of edge nodes
- Rst_{sets} : sets consists of the result set in each edge node's region

Ensure:
- wgt_{mtx} : a weighted adjacency matrix, whose values represent the cost of sensory data communication energy between contiguous edge nodes

1: $gnData_{mtx} \leftarrow \emptyset$
2: **for** $i = 0$; $i < num$; $i++$ **do**
3:   **for** $j = 0$; $j < num$; $j++$ **do**
4:     **if** $i \neq j$ **and** gn_i and gn_j are contiguous **then**
5:       $gnRst_{set} \leftarrow \emptyset$
6:       **while** each $Rst_{set\,j} \subset Rst_{sets} \neq NULL$ **do**
7:         $Temp_{set} \leftarrow$ get one attribute set from $Rst_{set\,j}$
8:         $O_{set} \leftarrow \emptyset$
9:         **while** $Temp_{set} \neq NULL$ **do**
10:           **if** $JS \leq \lambda$ **then**
11:             $O_{set} \leftarrow O_{set} \cup \{o\}$
12:           **end if**
13:         **end while**
14:         $gnRst_{set} \leftarrow gnRst_{set} \cup O_{set}$
15:       **end while**
16:       $gnData_{mtx}[i][j] \leftarrow gnRst_{set}$
17:       $k \leftarrow$ Calculate the transmission data of $gnRst_{set}$
18:       $d \leftarrow$ Euclidean distance of gn_i and gn_j
19:       $E_{ij}(k) \leftarrow$ calculated by Eq. (3)
20:       $wgt_{mtx}[i][j] \leftarrow E_{ij}(k)$
21:     **end if**
22:   **end for**
23: **end for**

4.2. Edge node routing graph construction

Considering the amount of sensory data generated by smart things in the marginal sub-region, a packet transmission graph is constructed upon edge nodes, in order to decrease the network traffic. The edge node data routing can be modeled as an optimization problem, where the energy consumption is considered as the decision factor:

$$Z = \Sigma_{i=1}^{n} \Sigma_{j=1}^{n} w_{ij} \times c_{ij} \tag{6}$$

where:

$$c_{ij} = \begin{cases} 0 & \text{otherwise} \\ 1 & (w_{ji} \neq 0 \,\text{and}\, w_{ij} \leq w_{ji}) \end{cases} \tag{7}$$

where w_{ij} (non-zero value) represents the energy consumption of an edge node to another edge node, and c_{ij} is calculated depending on the comparison of the energy values between two edge nodes. By objective function, we can achieve a minimum of energy consumption for data communication within a reasonably acceptable range.

Based on this function, a two-step strategy for graph construction is presented as follows: (i) the filter step is to filter out sensory data packets that do not contribute to the query results. Some edges are filtered by heuristic greedy algorithm. According to the results of Algorithm 3, by traversing neighbor edge nodes in turn, we reserve the directed edge with the smallest energy value, so that the total transmitted energy is minimized in the edge node routing graph construction. For example, the energy consumption from an edge node gn_i to a contiguous edge node gn_j is w_1, and the energy consumption from gn_j to gn_i is w_2 ($w_1 \leq w_2$). We naturally reserve the edge from gn_i to gn_j, and remove the edge from gn_j to gn_i. After this step of filtering, we have preserved the one-way transmission edge between the edge nodes. Considering the situation that a loop exists in the process of sensory data transmission, we propose (ii) the refinement step is to avoid the repeated transmission of data packets. It is worth noting that the graph we built is used to integrate the results of the query between the regions, the ring is not allowed to exist. However, in the filter step, we consider that there may be one ring in the filtered graph. Hence, we adopt a strategy as a refinement step during the construction of the edge node routing graph, which detects whether there is a ring in current graph. If there is a ring, we change the flow of data between the newly added edges. Ultimately, a unidirectional acyclic routing graph is constructed accordingly to represent edge node routing graph.

4.3. Marginal edge network query mechanism

Sensory data packets are transmitted between edge nodes, when these data are examined highly possible to benefit query answering. A pruning method is adopted to accelerate the query data transmission progress.

As presented by Algorithm 4, we achieve the decrease of energy consumption. We adopt control package pruning strategy which is designed as reducing packet transmission. As the input for an edge node routing graph, we send a control packet to determine whether gn_i needs to send data to gn_j (line 5). If the neighbor edge node needs the data, current edge node sends data (line 7). Otherwise, the procedure will detect the next edge node (lines 2–13). Based on this pruning strategy, we can calculate the optimized energy consumption Z (line 8) by Eq. (6), which is greatly beneficial to improve the processing performance. Note that the enumeration procedure applies only to some situations where the number of possible solutions is not too large. Given the limited number of query attributes, we can take an enumeration strategy to get an enumerated set of query between regions (line 9). Meanwhile, the time complexity of the enumeration algorithm depends on the number of loop nesting, which is the number of query attribute keywords.

Algorithm 4 MarginalRegionQuery.

Require:
- $drgh_{mtx}$: an edge node routing graph

Ensure:
- $CrsRst_{set}$: a set of numerous groups, where each group on the whole satisfies the query

1: $Z \leftarrow 0$; $num \leftarrow drgh_{mtx}.row$
2: **for** $i = 0$; $i < num$; $i++$ **do**
3:     **for** $j = 0$; $j < num$; $j++$ **do**
4:         **if** $drgh_{mtx}[i][j] \neq 0$ **then**
5:             $flag \leftarrow$ check the data demand of neighbor node gn_j
6:             **if** $flag$ **then**
7:                 gn_i transmit data to gn_j
8:                 $Z \leftarrow$ calculated by Eq. (6)
9:                 $CrsRst_{set} \leftarrow$ get enumeration groups
10:             **end if**
11:         **end if**
12:     **end for**
13: **end for**

4.4. Query processing

A query, which combines the queries for single edge networks and marginal edge networks, is handled. The combinations of smart things, which can satisfy certain queries in a collective fashion, can be retrieved and evaluated. Generally, the more cohesive the smart things in a collection are, the more appropriate the collection of smart things is with respect to the specification of certain queries. The clustering technique involving the Euclidean distance is adopted for evaluating the cohesive of smart things in a collection. The objective function is presented as followed:

$$RC(g) = \sum_{i=1}^{K} dst(g_c, o_i)^2 \qquad (o_i \in g) \tag{8}$$

where K denotes the number of smart things in a collection, g_c denotes the geographical centre of these smart things in this collection, and dst denotes the Euclidean distance between the smart thing and the geographical centre of the collection.

The procedure of query processing is presented at Algorithm 5. Query processing in single edge networks is handled as presented by Algorithm 2 (lines 2–11). Besides, an enumeration combination method is adopted for the result combination of single edge networks into collections (line 5). Furthermore, Eq. (8) is adopted to calculate the score for each collection in all single edge network result sets (lines 6–10). In addition, the query of the marginal edge network is performed by Algorithm 4 (line 12), where the same collection scoring rules is adopted for the data processing of marginal edge network (lines 13–17). A queue is used to store global query result collections, where each collection is arranged in the descending order (lines 9,16).

5. Implementation and evaluation

The prototype has been implemented in a Java program. Experiments are conducted upon a desktop with an Intel i5-6500 CPU at 3.20GHz, 8-GB of memory and a 64-bit Windows 10 system. In the following we introduce experiment settings and discuss evaluation results.

5.1. Experiment settings

Table 3 presents the parameter settings of our experiments. Without loss of generality, a query is assumed to be relevant with 1 to 4 kinds of attributes, since queries are typically not very

Table 3
Parameters settings in the experiments.

Parameters name	Value
Network query region (m^2)	200 × 200
Number of smart things	200 to 1000
Skewness degree	10% to 50%
Kinds of queried attributes	1 to 4
Percentage of boundary distance	40% to 80%
Number of bits in one pocket (k)	1
Attenuation index of transmission (n)	2
Energy consumption constants of transmit and receiver electronics (E_{elec})	50 nJ/bit
Energy consumption constant for transmit amplifier (ϵ_{amp})	0.1 $nJ/(bit \times m^2)$

Algorithm 5 QueryProcessing.

Require:
- q : a tuple (Rgn, Kd, $Cstr$)
- tr_{set} : a set consists of the root nodes for each region
- $drgh_{mtx}$: an edge node routing graph

Ensure:
- $queue$: a max-priority queue, where it is ranked according to Eq. (8)

$IntrGRst_{set} \leftarrow \emptyset$; $ExtrGRst_{set} \leftarrow \emptyset$; $n \leftarrow tr_{set}.size$
for each $tr_i \subset tr_{set}$, where $i = 0, 1, \ldots, n$ **do**
  $IntrRst_{set} \leftarrow \emptyset$
  $IntrRst_{set} \leftarrow$ IndexQuery(q, tr_i)
  $IntrGRst_{set} \leftarrow$ get enumeration groups from $IntrRst_{set}$
  while $IntrGRst_{set} \neq NULL$ **do**
    $g \leftarrow$ extract certain group from $IntrGRst_{set}$
    $RC(g) \leftarrow$ calculated by Eq. (8)
    $queue$.Enqueue(g, $RC(g)$)
  end while
end for
$ExtrGRst_{set} \leftarrow$ MarginalRegionQuery($drgh_{mtx}$)
while $ExtrGRst_{set} \neq NULL$ **do**
  $g \leftarrow$ extract certain group from $ExtrGRst_{set}$
  $RC(g) \leftarrow$ calculated by Eq. (8)
  $queue$.Enqueue(g, $RC(g)$)
end while

complex for the majority of domain applications. Besides, when the kinds of attributes that queries interest are large in number, queries should hardly be clearly explained and easily understood. The number of smart things ranges from 200 to 1000 with an increment of 200, and a smart thing is randomly assigned with a sensing attribute. Due to the fact that smart things may be distributed unevenly in the network, a skewness degree (denoted sd) is adopted to quantify this character. Intuitively, sd is calculated in terms of $(dn - sn) \div N$, where (i) dn and sn refer to the number of smart things deployed in dense and sparse sub-regions, respectively, and (ii) N is the sum of dn and sn [28].

As far as we know, this is the first technique to explore the distributed and localized query processing in the context of edge computing, where an IoT sensing network is composed by edge networks. To evaluate the efficiency of our technique, we have compared our technique with the *LEACH* routing protocol [29], where a routing tree is constructed to aggregate and forward sensory data packets to the sink. Note that in our experiments, the smart thing located in the network centre is selected to serve as the sink. Without loss of generality, the sink node is assumed to have unlimited energy. Therefore, the energy consumed for receiving data packets is specified as follows:

$$E_{ij}(k) = \begin{cases} E_{elec} \times k + \epsilon_{amp} \times k \times d^n & \text{if } j \text{ is SN} \\ 2 \times E_{elec} + \epsilon_{amp} \times k \times d^n & \text{otherwise} \end{cases} \quad (9)$$

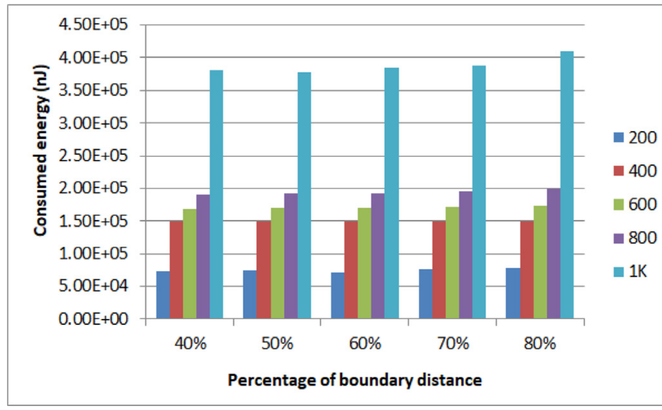


Fig. 3. Energy consumption for various percentages of boundary distance and numbers of smart things.

The results of experimental evaluation are presented and compared as follows, where various number of attributes, various skewness degrees, and different percentage of smart things deployed in the marginal region of edge networks are the factors to be considered in experiments. To reduce the randomness caused by the environmental configuration, experiments with a certain parameter setting is conducted ten times, and an average value is adopted as the final result as shown in the following figures.

5.2. Evaluation results

This section presents and discusses the experimental results about the performance of query processing.

5.2.1. Various percentages of boundary distance and numbers of smart things

Fig. 3 shows the comparison of energy consumption when the percentage of boundary distance ranges from 40% to 80% with an increment of 10%. The number of smart things varies from 200 to 1000, with the 40% skewness degree. The number of attributes is set to 4 in query specification. Generally, the percentage of boundary distance specifies the size of marginal regions in contiguous edge networks, which determines the number of smart things involved in marginal edge networks query processing. This figure shows that the energy consumption increases slightly, rather than significantly, when the percentage of boundary distance changes from a relatively small value to a quite large one, since the energy is mostly consumed by forwarding sensory data packets along the edge node routing graph for gathering and aggregating data in our experiments. However, in the case when there are few sensory data packets are to be transmitted, the energy consumption should be impacted largely by the percentage of boundary distance.

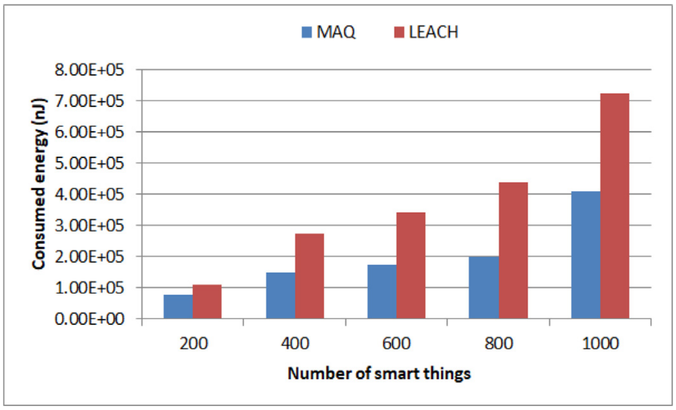


Fig. 4. Energy consumption for *MAQ* and *LEACH* when various numbers of smart things are deployed in the network.

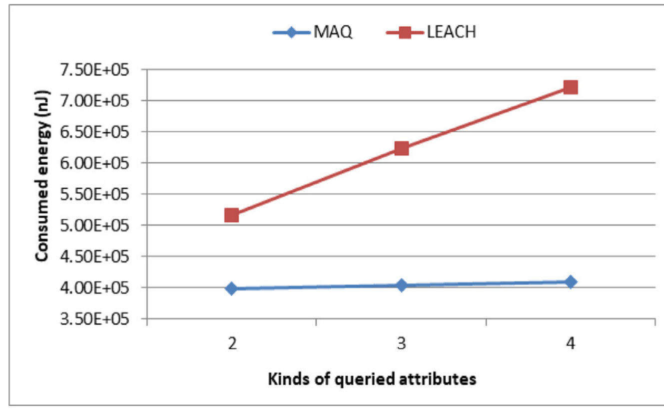


Fig. 5. Energy consumption for *MAQ* and *LEACH* when various kinds of attributes are specified in query specification.

5.2.2. Comparison for MAQ and LEACH considering various numbers of smart things

Fig. 4 shows the energy consumption for our *MAQ* and *LEACH*, when the numbers of smart things is set from 200 to 10,000 with an increment of 200. The percentage of boundary distance is set to 80%, and the other parameters are set to the same values as those in Fig. 3, which is convenient to eliminate the influence of other factors and interference on the experimental results. This figure shows that *LEACH* requires more energy consumption than *MAQ*. In fact, *LEACH* routes sensory data of smart things with attributes specified by query specifications to the centre for centralized processing. On the other hand, *MAQ* gathers sensory data of smart things in edge networks, processes these data in a localized fashion, and routes the result of certain edge networks to the centre. Note that sensory data of marginal smart things contained in contiguous edge networks are required to be route along the routing graph. However, the amount is much smaller than that of the packets to be transmitted in *LEACH*. This figure also shows that the increase of energy consumption for *LEACH* is much larger than that for *MAQ*. In fact, when smart things are relatively larger in number, the amount of sensory data that are processed locally by edge networks should be larger in percentage, and hence, more energy should be reduced by *MAQ* than *LEACH*. This result indicates that *MAQ* can perform better than *LEACH* in decreasing energy consumption when the network is relatively large in the number of smart things.

5.2.3. Comparison for MAQ and LEACH considering various kinds of queried attributes

Fig. 5 shows the energy consumption for *MAQ* and *LEACH*, when the number of attributes is set to 2, 3 or 4 in query specification. The number of smart things is set to 1000, and other parameters are set to the same values as those in Fig. 4. This figure shows that the energy consumption is largely increased in a linear manner with respect to the increasing of the attribute number. This result is reasonable since the number of attributes is proportional to the number of smart things to be explored. On the other hand, the increasing of energy consumption is much smaller in scale for our *MAQ* than *LEACH*, since the majority of the query processing task is conducted locally in edge networks, and we argue that this strategy should decrease the network traffic and energy consumption significantly.

5.2.4. Comparison for MAQ and LEACH considering various skewness degrees

Fig. 6 shows the energy consumption for *MAQ* and *LEACH*, when the skewness degree is set from 10% to 50% with an increment

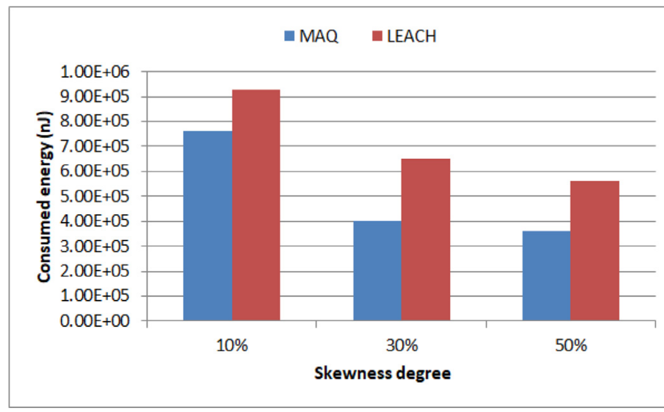


Fig. 6. Energy consumption for *MAQ* and *LEACH* when smart things are distributed in the network with various skewness degrees.

of 20%. Other parameters are set to the same values as those in Fig. 5. This figure shows that *LEACH* consumes much more energy than *MAQ*, due to the same reason as presented in Fig. 4. Besides, the energy consumption is relatively smaller when the skewness degree is larger (i.e., 50%). In fact, head nodes in *LEACH*, as well as edge nodes in *MAQ*, are mostly chosen from sensor nodes (or smart things) which are located within dense sub-regions. When the skewness degree is large, the majority of sensory data gathering and routing tasks should be conducted in dense sub-regions, and this suggests that the transmission distance of most packets should be shorter. On the other hand, when the skewness degree is small, which means that smart things are distributed in a relatively even manner in the network, sensory data packets should be longer in their average transmission distance. Generally, *MAQ* is more energy efficient when smart things are distributed in a skewed fashion.

6. Related works and comparison

Along with the huge and increasing number of smart things deployed in IoT sensing networks, multi-attribute query processing is considered as fundamental to support domain applications. Traditional techniques have been developed to support the query processing in single edge networks. In [15], authors explore the problem of retrieving a group of spatial web objects. The group's keywords require to cover the query's keywords, and the objects in the group should be geographically as close as possible. A cost function is defined to evaluate the merits of the results, which is composed of two kinds of semantic types. One takes into account the sum of the distance between each object in the group and the query location, which may fit with applications where the objects need to meet at the query location, such as incident handling or the finding of project partners. Another type is the maximal distance between any object in the group and the query location, which may be understood as the situation where tourists plan to visit several points of interest. This query for the object groups inspires the research presented in this article. Note that a centralized index tree is constructed to support the query of object groups. This strategy should be applied to single edge networks, but may not be applicable to large-scale IoT sensing networks composed of multiple edge networks.

In [14], authors present an R-tree-based indexing technique that stores compact histograms in node entries, while preserving reasonable node fanout. Leveraging the index and histogram, a pruning strategy is implemented to prune the search space and guide the search while considering the factors including group diameter, distance, and relevance to the query. Generally, this histogram for pruning the search space is a promising mechanism for supporting query processing. Hence, an improved pruning strategy is proposed in [16]. Since objects may be unevenly distributed in the network, authors adopt proper mechanisms for handling objects contained in dense and sparse sub-regions. Assuming there are two sets of groups that can satisfy the query, objects in one group is in a hotspot region, and objects in the other group is in a sparse region. When the distance cost is almost the same, objects in dense sub-regions should be prone to be recommended, since they can have more counterparts to be replaced when found improper. Therefore, dealing with spatial keyword queries, the region density is also a factor to be considered. Authors propose a method to calculate the lower bound of the density cost of a node, and to prune nodes with the lower bound of density cost than the past minimum cost.

To manage objects in a network, an index tree like an R-tree is usually constructed to support spatial and multi-attribute query processing. An R-tree index is proposed in [30] to handle spatial keyword queries. In computer aided design and geo-data applications, the mechanism about the search of massive information in spatial databases is fundamental. The processing of non-zero-sized data in a multidimensional space can hardly be solved with the traditional indexing method. Therefore, authors propose an R-tree to facilitate regular access methods in relational databases. Generally, this technique considers the spatial query processing, while the text relevancy is not the focus. To remedy this issue, an index tree integrating the inverted file for text retrieval and R-tree for spatial proximity query is developed [10], such that the spatial and text relevance is considered with respect to query specification. Besides, a range region query is proposed in [31], in order to retrieve objects with keywords in a certain range. A direction-aware spatial keyword query method [17] is proposed to inherently support object query within certain directions.

Note that searching strategy for smart things is popular nowadays. In [33], the concept of multi-region attribute aggregation query over sensors in skewness distribution is presented. Authors establish an energy-efficient spatial index tree to resolve the multi-region attribute aggregation query. Generally, this technique constructs an index tree to support query in all region, which is quite different from the aggregation query proposed in our technique. The processing of the multi-region attribute aggregation query inspires us to develop the marginal edge network query processing. With the popularity of big data applications [34,35], information is no longer stored in a single region. The distributed technology is increasingly used. In [36], interoperability is assumed as a challenge in implementing IoT applications.A distributed Internet-like architecture for things is proposed for the process of large-scale expansion of IoT. In general, this proposed distributed architecture helps intelligent decision-making and enables automated service creation. It is worth noting that some service matching and allocation strategies [38–40] are also beneficial for searching objects. In [38], considering the explosion of Internet of things, big data and fog computing in cloud computing environment, authors explore the scheduling strategy of cloud and fog resources. This exploration has an enlightening effect on the collaboration of multiple edge nodes in the edge computing environment. Other techniques explore the network communication topology [20], an effective collection [21], management [22], and aggregation [23] of sensory data, a load-balancing routing [24], and the prolonging of network lifetime [25,26], in the context of IoT. In [37], in order to solve the mobile environment, the data source can not be accessed due to the partition of the network. The author proposes *C*ontent *C*entric *N*etworks (*CCN*) use in-network caching. In general, based on the reliable strategies in networks of [37], this work provides reliable data transmission and routing mechanism for us to handle queries in the marginal edge network. However, sensory data fusion in marginal edge network and the query processing mechanism in single edge networks are not explored.

To summarize, current techniques construct a centralized index tree to support spatial and multi-attribute objects query processing. They are inspiring for us when developing our technique, however we argue that they should not be efficient when the network is large in scale. Due to this consideration, we propose a distributed and localized query processing mechanism to support multi-attribute query processing in edge computing.

7. Conclusions

With the swift growth of smart things being deployed in industrial environments, sensory data gathering and aggregation is fundamental to support IoT applications. Considering the large-scale of the network, the traditional centralized mechanism may not be efficient and applicable when considering the factors including network traffic and energy consumption, edge computing is adopted to promote the distributed and localized query processing. In this context, this article proposes a multi-attribute aggregation query mechanism in edge computing to support large-scale industrial IoT applications. Specifically, an energy-aware IR-tree is constructed to process query processing in certain edge networks, and an edge node routing graph is established for aggregating and forwarding sensory data packets between edge nodes, in order to facilitate query processing for marginal smart things in contiguous edge networks. Extensive experiments have been conducted to evaluate the efficiency and applicability of our technique. The results demonstrate that this technique performs better than the rivals in reducing the network traffic and energy consumption. This article retrieves the set of sensory data relevant to the query specification. This strategy requires to examine all IoT nodes in the query sub-region. In fact, when IoT nodes are densely deployed in the network, partial IoT nodes may reflect the fact with certain accuracy and may satisfy the requirement of domain application. Consequently, discovering partial IoT nodes in the query sub-region for satisfying certain requirements is our future research challenge.

Acknowledgments

This work was supported partially by the National Natural Science Foundation of China (Grant No. 61772479 and 61662021) and by the Fundamental Research Funds for the Central Universities (China University of Geosciences (Beijing), China).

References

[1] W. Feng, Y. Qin, S. Zhao, D. Feng, AAoT: lightweight attestation and authentication of low-resource things in IoT and CPS, Comput. Netw. 134 (2018) 167–182.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.
[3] S. Wang, A. Zhou, M. Yang, L. Sun, C.H. Hsu, F. Yang, Service composition in cyber-physical-social systems, IEEE Transactions on Emerging Topics in Computing PP (99), 2017. 1–1
[4] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, H. Zhang, Incentive mechanism for computation offloading using edge computing: a stackelberg game approach, Comput. Netw. 129 (2017) 399–409.
[5] K. Kaur, S. Garg, G.S. Aujla, N. Kumar, J.J.P.C. Rodrigues, M. Guizani, Edge computing in the industrial internet of things environment: software-defined-networks-based edge-cloud interplay, IEEE Commun. Mag. 56 (2) (2018) 44–51.
[6] D. Zhang, J. Wan, C.H. Hsu, A. Rayes, Industrial technologies and applications for the internet of things, Comput. Netw. 101 (2016) 1–4.
[7] S. Xiong, Q. Ni, X. Wang, Y. Su, A connectivity enhancement scheme based on link transformation in IoT sensing networks, IEEE Internet Things J. 4 (6) (2017) 2297–2308.
[8] Y. Zhou, X. Xie, C. Wang, Y. Gong, W.Y. Ma, Hybrid index structures for location-based web search, ACM Int. Conf. Inf. Knowl. Manage. (2005) 155–162.
[9] D. Harman, R. Baeza-Yates, E. Fox, W. Lee, Inverted files, in: Information retrieval, 1992, pp. 28–43.
[10] G. Cong, C.S. Jensen, D. Wu, Efficient retrieval of the top-k most relevant spatial web objects, in: Proceedings of the VLDB Endowment, 2009, pp. 337–348.
[11] Z. Li, K.C.K. Lee, B. Zheng, W. Lee, D.L. Lee, X. Wang, IR-tree: an efficient index for geographic document search, IEEE Trans. Knowl. Data Eng. 23 (4) (2011) 585–599.
[12] D. Wu, G. Cong, C.S. Jensen, A framework for efficient spatial web object retrieval, Int. J. Very Large Data Bases 21 (2012) 797–822.
[13] D. Zhang, Y.M. Chee, A. Mondal, A.K.H. Tung, M. Kitsuregawa, Keyword search in spatial databases: towards searching by document, in: IEEE International Conference on Data Engineering, 2009, pp. 688–699.
[14] A. Skovsgaard, C.S. Jensen, Finding top-k relevant groups of spatial web objects, Int. J. Very Large Data Bases 24 (2015) 537–555.
[15] X. Cao, G. Cong, C.S. Jensen, B.C. Ooi, Collective spatial keyword querying, in: ACM SIGMOD International Conference on Management of Data, 2011, pp. 373–384.
[16] L. Zhang, X. Sun, Z. Hai, Density-based spatial keyword querying, Future Gener. Comput. Syst. 32 (1) (2014) 211–221.
[17] G. Li, J. Feng, J. Xu, Desks: direction-aware spatial keyword search, in: IEEE International Conference on Data Engineering, 2012, pp. 474–485.
[18] N.K. Tran, Q.Z. Sheng, M.A. Babar, L. Yao, Searching the web of things: state of the art, challenges, and solutions, ACM Comput. Surv. (CSUR) 50 (4) (2017) 55.
[19] Y. Zhou, S. De, W. Wang, K. Moessner, Search techniques for the web of things: a taxonomy and survey, Sensors 16 (5) (2016) 600.
[20] G.A. Akpakwu, B.J. Silva, G.P. Hancke, A.M. Abu-Mahfouz, A survey on 5g networks for the internet of things: communication technologies and challenges, IEEE Access 5 (12) (2017) 3619–3647.
[21] C.T. Cheng, N. Ganganath, K.Y. Fok, Concurrent data collection trees for IoTapplications, IEEE Trans. Ind. Inf. 13 (2) (2017) 793–799.
[22] O. Diallo, J.J.P.C. Rodrigues, M. Sene, Real-time data management on wireless sensor networks: a survey, J. Netw. Comput. Appl. 35 (3) (2012) 1013–1021.
[23] F. Ren, J. Zhang, Y. Wu, T. He, C. Chen, C. Lin, Attribute-aware data aggregation using potential-based dynamic routing in wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 24 (5) (2013) 881–892.
[24] H. Wang, H. Xu, L. Huang, J. Wang, X. Yang, Load-balancing routing in software defined networks with multiple controllers, Comput. Netw. 141 (2018) 82–91.
[25] Z. Fadlullah, M. Fouda, N. Kato, A. Takeuchi, Toward intelligent machine-to-machine communications in smart grid, IEEE Commun. Mag. 49 (4) (2011) 60–65.
[26] B. Guo, J. Yu, B. Liao, D. Yang, L. Lu, A green framework for DBMS based on energy-aware query optimization and energy-efficient query processing, J. Netw. Comput. Appl. 84 (2017) 118–130.
[27] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, IEEE Comput. Soc. 18 (2000) 8020.
[28] Z. Zhou, D. Zhao, L. Shu, H.C. Chao, Efficient multi-attribute query processing in heterogeneous wireless sensor networks, J. Internet Technol. 15 (5) (2014) 699–712.
[29] F. Shang, Y. Lei, An energy-balanced clustering routing algorithm for wireless sensor network, J. Comput. Theor. Nanosci. (2010) 777–783.
[30] A. Guttman, R-trees: a dynamic index structure for spatial searching, ACM SIGMOD Int. Conf. Manage. Data 14 (2) (2016) 47–57.
[31] R. Hariharan, B. Hore, C. Li, S. Mehrotra, Processing spatial-keyword (SK) queries in geographic information retrieval (GIR) systems, in: International Conference on Scientific and Statistical Database Management, 2007. 16–16
[32] C. Zhu, H. Zhou, V.C.M. Leung, K. Wang, Y. Zhang, L.T. Yang, Toward big data in green city, IEEE Commun. Mag. 55 (11) (2017) 14–18.
[33] J. Tang, B. Zhang, Y. Zhou, L. Wang, An energy-aware spatial index tree for multi-region attribute query aggregation processing in wireless sensor networks, IEEE Access 5 (99) (2017) 2080–2095.
[34] R. Sowmya, K.R. Suneetha, Data mining with big data, in: International Conference on Intelligent Systems and Control, 2017, pp. 246–250.
[35] M. Gohar, S.H. Ahmed, M. Khan, N. Guizani, A. Ahmed, A.U. Rahman, A big data analytics architecture for the internet of small things, IEEE Commun. Mag. 56 (2) (2018) 128–133.
[36] C. Sarkar, U.N.S.N. Akshay, R.V. Prasad, A. Rahim, R. Neisse, G. Baldini, DIAT: a scalable distributed architecture for IoT, IEEE Internet Things J. 2 (3) (2017) 230–239.
[37] N. Sheneela, R.N.B. Rais, P.A. Shah, S. Yasmin, A. Qayyum, S. Rho, Y. Nam, A dynamic caching strategy for CCN-based MANETs, Comput. Netw. 142 (2018) 93–107.
[38] L. Ni, J. Zhang, C. Jiang, C. Yan, K. Yu, Resource allocation strategy in fog computing based on priced timed petri nets, IEEE Internet Things J. 4 (5) (2017) 1216–1228.
[39] X. Xue, S. Wang, L. Zhang, Z. Feng, Y. Guo, Social learning evolution (SLE): computational experiment-based modeling framework of social manufacturing, IEEE Transactions on Industrial Informatics PP (99), 2018. 1–1.
[40] X. Xue, S. Wang, L.j. Zhang, Z.y. Feng, Evaluating of dynamic service matching strategy for social manufacturing in cloud environment, Future Gener. Comput. Syst. 91 (2019) 311–326.

Xiaocui Li is a master student at School of Information Engineering, China University of Geosciences (Beijing). Her research interests include edge computing and IoT sensing network. Email:lixiaocui.cugb@gmail.com.

Zhangbing Zhou is a professor at China University of Geosciences (Beijing), China, and an adjunct professor at TELECOM SudParis, Evry, France. His research interests include wireless sensor networks, services computing and business process management. Email: zhangbing.zhou@gmail.com. (Corresponding Author)

Junqi Guo is an associate professor at Beijing Normal University, Beijing, China. His research interests include intelligent signal and information processing, Internet of Things and smart wear, and 5G communication physical layer algorithm optimization. Email: guojunqi@bnu.edu.cn.

Shangguang Wang is an associate professor at the State Key Laboratory of Networking and Switching Technology (BUPT) in State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include service computing, cloud computing, and mobile edge computing. Email: sgwang@bupt.edu.cn.

Junsheng Zhang is a professor at Institute of Scientific and Technical Information of China, Beijing, China. His research interests include information system technology, information management and intelligence analysis, and semantic link network. Email: zhangjs@istic.ac.cn.