IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 9, 2021, accepted May 18, 2021, date of publication May 21, 2021, date of current version June 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3082532

In-Network Caching for the Green Internet of Things

YUAN REN[1], XUEWEI ZHANG[1], TING WU[2], AND YIXUAN TAN[2]
[1]Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[2]School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Corresponding author: Xuewei Zhang (zhangxw@xupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61801382 and Grant 62071377, and in part by the Natural Science Foundation of Shaanxi Province under Grant 2020JQ-849 and Grant 2021JQ-705.

ABSTRACT By caching multiple copies of the sensed data and suppressing repeated content deliveries from the sensor nodes, the in-network caching exhibits strong potentials to reduce energy consumption and avoid service loss in the Internet of Things (IoT). In this paper, we review the fundamentals of the in-network caching in IoT, especially the deployment principles of the cache-enabled brokers. Since in-network caching is not the optimal choice in some cases, we propose the dynamic in-network caching scheme to decide whether the sensed data should be cached in brokers or not, where the location and delay requirement of the clients, traffic load of the brokers as well as the energy level of the sensor nodes are encapsulated in the proposed scheme. Next, we give a practical case study to demonstrate the effectiveness of the proposed in-network caching scheme in the IoT scenario. It shows that the proposed caching scheme is superior to the traditional sensor-to-client scheme in terms of energy saving. Finally, we conclude this paper, and summarize the future research directions to achieve the in-network caching for implementing the green IoT.

INDEX TERMS In-network caching, green Internet of Things (IoT), sensing service, broker, energy consumption.

I. INTRODUCTION

Nowadays, the number of Internet of Things (IoT) devices is escalating rapidly [1], [2], and IoT owns the potential to bring 11 percent economical gains in the year of 2025 [3]. With tremendous heterogeneous IoT devices and various applications, the real world is truly transformed into a global village, providing us unprecedented convenience and profoundly refining the manners that we live, work and study [4], [5].

Though IoT is not the main component for greenhouse emissions and global warming, it still adversely affect the environmental protection [6], [7], since there are tremendous and ubiquitous connectivities of the IoT devices [8], [9]. In this context, in-network caching has been regarded as a candidate technique to reduce the energy consumption caused by IoT devices [10]–[12], The operability and feasibility of in-network caching have been studied by the Information-Centric Networking Research Group (ICNRG) [13], [14]. In-network caching enables the broker, an intermediate node deployed with finite storage capacity, such as the gateway or router, to pre-store some popular data resources monitored by sensor nodes. When cached data resources are requested by the clients, they can be found in the local cache of brokers [15]. As a consequence, it can be concluded that the in-network caching is capable of suppressing superfluous content deliveries from the resource-hosting sensor nodes, and there will be less energy consumption resulted from sensor nodes [16], [17]. In addition, even if these sensor nodes are turned off into the sleeping mode for the purpose of energy saving, the clients can still access to the copies of the sensed data in brokers, thus the service loss can be largely avoided [18].

In content delivery networks (CDNs), such as cloud radio access networks (C-RAN) [19], [20], fog radio access network (F-RAN) [21], [22], heterogeneous networks (HetNets) [23], [24] and small cell networks (SCNs) [25], [26], the conventional wireless edge caching schemes have been extensively studied, aiming for alleviating the serious traffic burden of capacity-limited backhaul links and thus mitigating service delay. The pre-cached contents in the CDN scenarios, such as the video files, are typically insensitive

The associate editor coordinating the review of this manuscript and approving it for publication was Emanuele Lattanzi.

to time changes [27], [28]. For this reason, the popularity of these contents remains unchanged for a long time period, and can often be regarded as a known information when designing the caching schemes. Moreover, the content update can experience a long time period, e.g., a day or a week, depending on the specific types of video requests. As opposed to the former caching schemes appearing in CDN scenarios, in-network caching for the IoT poses distinct characteristics. The main characteristics of in-network caching are listed as follows.

- The observed data is time-variant for in-network caching. In the IoT scenario, the data resources obtained by sensor nodes, such as those for health monitoring, fire detection and traffic congestion can change faster than the cached contents in CDNs. The former data may *"expire"*, and new data demands will be generated in this case. In this light, the process of data update has to be realized frequently and periodically, so as to guarantee the freshness of the cached data.
- The sensed data is often small-sized in the IoT, owing to the limited battery capacity of the sensor nodes. Though the data resources are typically in the small sizes, the number of the sensed data bits will be relatively large, since there are tremendous sensor nodes around our daily lives.
- The primary aim of the in-network caching in IoT is to decrease the total energy consumption generated by battery-powered and unrechargeable sensor nodes, so as to guarantee the sustainability of IoT services. However, the conventional caching in CDNs emphasizes more on relieving the traffic burden of backhaul links and reducing time latency of end users.

From the above, we can conclude that realizing the green IoT is the supreme goal of in-network caching. To achieve this goal, we have to deal with the challenges brought by in-network caching. Owing to the unique characteristic of in-network caching mentioned above, the existing caching schemes cannot be directly extended into the in-network caching for the IoT scenario. Therefore, it is of great necessity to design the efficient in-network caching schemes coping with these tough issues.

In light of the above, in this paper, we investigate the energy-efficient in-network caching to implement the green IoT. Different from the traditional sensor-to-client transmission strategy, clients are allowed to fetch the requested data resource from brokers with the aid of in-network caching, which can relieve the traffic burden of sensor nodes and reduce energy consumption of these nodes. The key contributions of this work are summarized as follows.

- We present the fundamentals of in-network caching in the IoT architecture, including the request popularity, key impacting factors and application scenario. The deployment principles of cache-enabled brokers are also detailed specified, especially the details of the timer.
- We prompt the dynamic energy-efficient in-network caching for the green IoT to minimize the total energy consumption of the sensor nodes. The proposed scheme jointly considers the location and delay requirement of clients, the traffic load of brokers as well as the energy level of sensor nodes.
- Furthermore, a practical case study in the IoT scenario is given to present the effectiveness of the proposed in-network caching scheme. Numerical results clearly show that the in-network caching owns great potential to reduce energy consumption of the sensor nodes, and hence prolong the life span of the entire IoT.

The rest of the paper is organized as follows. Section II provides the fundamentals of in-network caching for the green IoT. Section III proposes the energy-efficient in-network caching scheme for the green IoT. In Section IV, a case study is performed to demonstrate the superiority of the proposed energy-efficient in-network caching scheme in the IoT scenario. Finally, Section V summarizes the paper and envisions future directions to implement the in-network caching for the green IoT.

II. THE FUNDAMENTALS OF IN-NETWORK CACHING FOR THE GREEN IoT

IoT is not a fresh concept, but gains more popularity and momentum recently. It can be widely applied in the deployment of smart city [29], which provides us the platform to improve working efficiency and the access to more comforts in the daily lives. IoT consists of three layers, namely, the perception layer, the network layer, and the application layer. An detailed example of this kind of layered structure of IoT is shown in Fig. 1. In specific, the perception layer contains many of heterogeneous IoT devices, such as smart sensor, laptop, wearable device, actuator and unmanned aerial vehicle (UAV), to execute the function of sensing, transferring and communicating; the network layer is responsible for providing the platform for the sensor nodes to communicate with each other; and the application layer is capable of supplying multiple data resources and smart services to clients upon their requests. In particular, the sensing service is the most common and widely used application in IoT, through which the clients can obtain the monitoring conditions of the surrounding environments and take actions accordingly.

A. TWO WIDELY USED PROTOCOLS IN THE APPLICATION LAYER

Zooming in on the application layer, there are two widely adopted protocols [30], namely, the constrained application protocol (CoAP) and CoAP publish/subscribe (CoAP Pub/Sub), as shown in Fig. 2. CoAP exhibits a rather simple transmission process, where the sensor node directly transmits the sensed data to the client requesting this data. On the contrary, in CoAP Pub/Sub, a broker is introduced to locally cache the sensed data from the resource-hosting sensor. Under this transmission protocol, the sensor first trans-

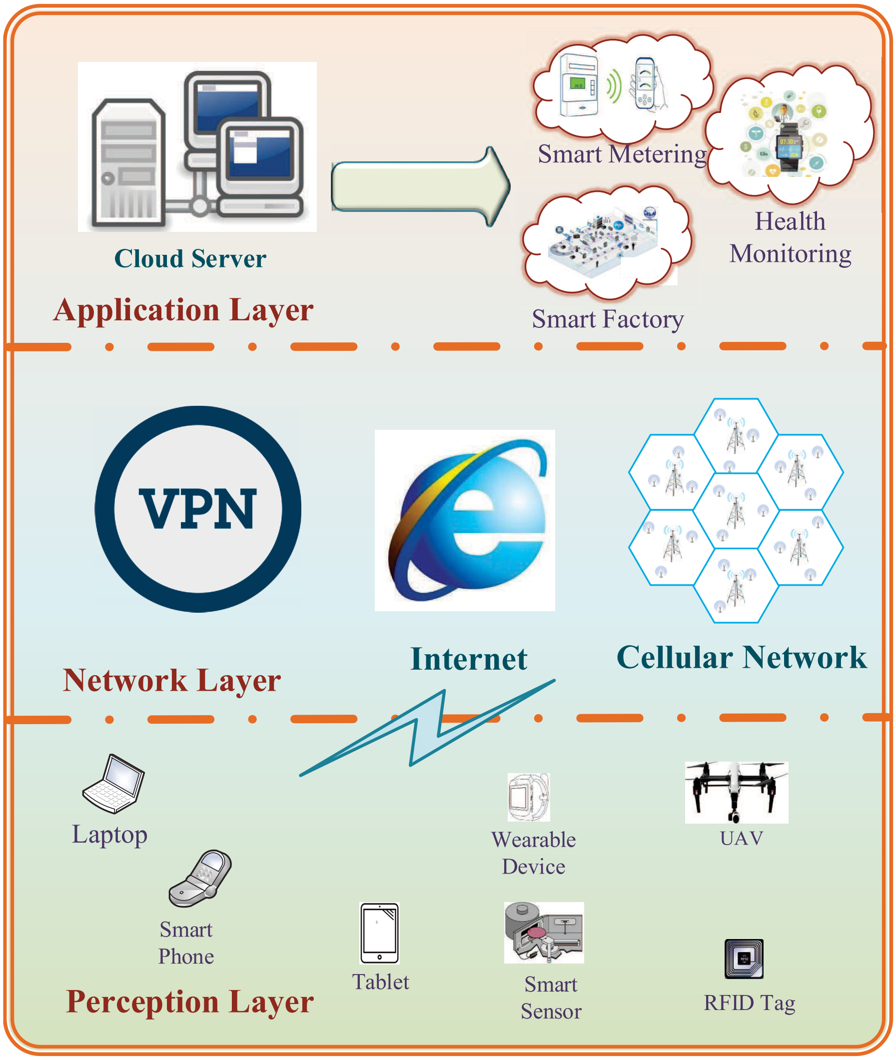

FIGURE 1. An example of the layered IoT structure.

mits the sensed data to the broker, then the client can access to the requested data via this broker. The primary motivation of CoAP Pub/Sub is to allow the client to obtain the sensed data even if the resource-hosting sensor is in the sleeping mode, avoiding frequently activating the sensor nodes and reducing energy consumption for data transmissions.

Due to multiple copies of the sensed data in brokers, CoAP Pub/Sub is superior to CoAP in terms of reducing energy consumption and avoiding service loss. However, the multi-hop transmission resulting from CoAP Pub/Sub leads to severer time delay as well as less freshness of the requested data than CoAP. As a consequence, the tradeoff between the total energy consumption of the sensors and multi-hop transmission should be fully considered in the practical IoT implementations.

Remark 1: Note that the network proxies are usually deployed in the computer network, and responsible for forwarding the requested information between the Internet and clients. If network proxies are used in the computer network, the data caching service can be provided by the architectures of content distribution network (CDN) and information centric network (ICN). Different from the host-to-host information transmission mode, in CDN, a new virtual network needs to be built above the current Internet for content distribution, leading to high operation expenditure and low communication efficiency. Moreover, if the ICN architecture is adopted, other extra data operations are supposed, such as data naming and positioning. Though these two architectures can provide data caching services for clients, the network architecture and protocols need to be carefully designed, and more deployment expenditure is needed. In our proposed caching scheme, the data caching service can be realized by only deploying brokers between sensor nodes and clients. Compared with CDN and ICN, there is little change of the network architecture, improving the communication efficiency and reducing the operating costs. The proposed caching scheme can also effectively reduce the total energy consumption, as shown by the simulation results. Particularly, it should be pointed

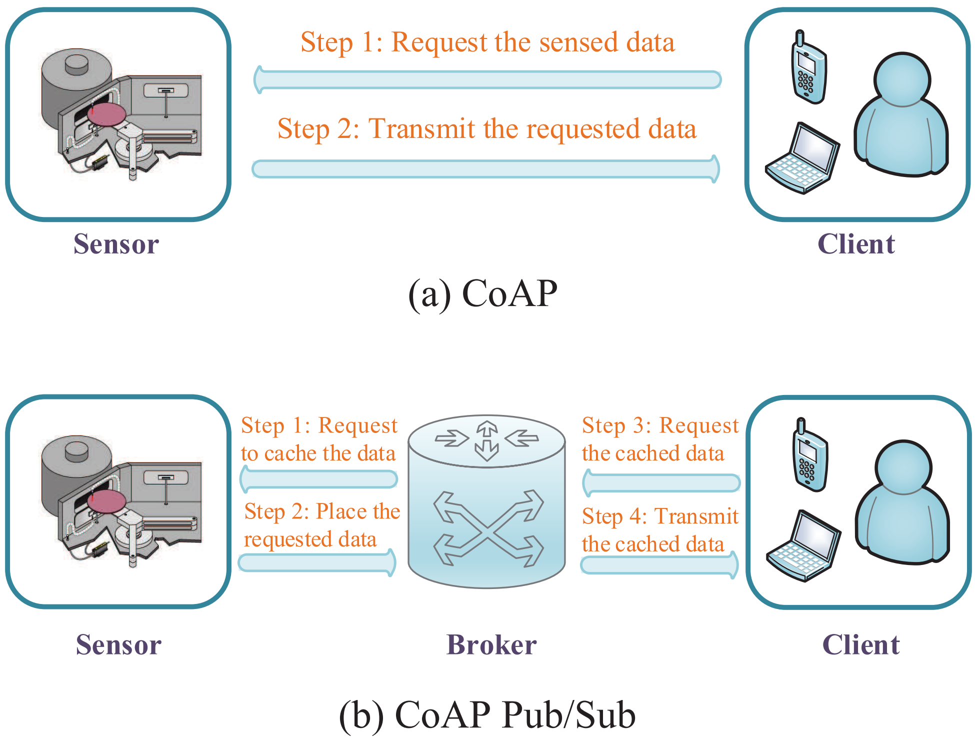

FIGURE 2. Two widely used protocols in the application layer, i.e., (a) CoAP and (b) CoAP Pub/Sub.

out that the proposed in-network caching scheme can also be applied in the cellular-IoT (C-IoT) scenario, if massive machine-type communications, one of the typical application cases in the fifth Generation (5G) communications, are needed.

B. THE IMPLEMENTATIONS OF THE BROKERS

The broker is a key component in the realization of CoAP Pub/Sub. More details are illustrated as follows. First of all, the **broker** is equipped with the storage capacity, and is endowed with the ability to cache the popular data resources hosted by nearby sensor nodes. When the cached data resources are requested by clients, they can be locally found in the brokers, and then transmitted to clients without retrieving them from the sensors. In practical implementations, the gateway or router can act as the broker. Unlike the battery-powered and energy-limited sensor nodes, the brokers have sufficient and steady energy supply, and hence can provide superior quality of service (QoS) to clients. Different from the traditional caching schemes studied in CDNs [19], [25], [31], the cached data in brokers is transient for the IoT scenario, and may lose its value quickly. Therefore, the freshness of the data resource should be paid a lot of attention. To this end, a timer is demanded when implementing the broker [32]. The timer is used to judge whether the current data is fresh to the client. To be more specific, if the timer does not reach the pre-defined time threshold, it means that the cached data is actually what the client requires, and the broker will not update its cached data resource. Otherwise, the broker is supposed to obtain the sensed data from sensors, update its own cached data, and then serve the clients.

Though caching the data resources in brokers can reduce the total energy consumption of the sensors and avoid service loss, in-network caching is not the optimal choice in the following cases:

- The sensed data resource is not popular, such as the data for personal health monitoring. In this case, there is little client requesting the data. When caching this kind of data, it will be a resource waste for the brokers, since caching still causes energy consumption. Moreover, owning to the multi-hop data transmission, it results in extra time delay to the clients requesting this data.
- The brokers are in traffic congestion. When there are considerable cached data in the local storage of the brokers, many of clients can fetch the requested data via these brokers, leading to serious traffic congestion. In this case, the clients have to suffer from unacceptable time latency, and this may have significant negative effects on some time-critical IoT applications, such as those for fire and toxic gas detections.
- The fresh time of the sensed data is extremely short. If the freshness of the data can only remain for a extremely short time period, this data should be better to be retrieved from the nearby sensors. This is due to the reason that the multi-hop transmission via brokers will lead to increased time delay and less freshness of the data.

- The delay requirements of the clients are critical, and these clients are farther from the brokers that cache the requested data. In this case, the requested data is supposed to be retrieved from the nearby sensor nodes.

Inspired by what listed above, the in-network caching in IoT deserves great attention, which can decide whether the sensed data should be cached in brokers or not. Afterwards, the combined retrieve scheme is correspondingly determined. The requested data resource can be fetched from either the sensors or the brokers, which is capable of reducing total energy consumption of the sensors, guaranteeing the delay requirement of clients and adapting to the dynamic network states.

III. THE IN-NETWORK CACHING SCHEME FOR THE GREEN IoT

Due to the unique characteristics of the in-network caching in IoT, caching the sensed data in the brokers are not always the optimal choice, as discussed above. Depending on different user requirements and network states, the clients can flexibly access to their requested data from sensors or brokers. In this section, we reasonably assume that the data resources of interest are popular for the clients, and then we propose the energy-efficient in-network caching scheme in IoT to minimize the total energy consumption of the sensor nodes.

A. THE PRELIMINARIES OF THE IN-NETWORK CACHING

The content popularity of the traditional caching schemes in CDNs is in-variant for a long time period. Generally, the probability of the content requests follows the Zipf's law. By selecting the proper skewness parameter, the request concentration can be accurately reflected. However, this principle cannot be tailored for IoT. First of all, we will give the definition of the data popularity in IoT [30]. For simplicity but without loss of generality, we focus on caching the popular data resources in brokers.

Definition 1: In a time slot, if the number of times that the clients request the data is larger than the number of times that the brokers update the data, this data can be regarded as a popular data.

It is also assumed that the storage of brokers are large enough, so that all the small-sized data resources generated by the sensing services considered in our scenario can be placed. Owing to the unique characteristics of IoT, when designing the optimal in-network caching scheme, the following issues should be taken into consideration:

- Consistency: The consistency of in-network caching requires that the cached data is consistent with the requested data. In this context, the fresh time of the data is introduced to guarantee the consistency issue.
- Relative locations of the clients, sensors and brokers: For some delay-critical and real-time sensing applications, the clients are more likely to obtain the requested data from the closer data container.
- Delay requirements: Multiple IoT clients may have diverse delay requirements, ranging from several milliseconds to several seconds, and the clients with loose delay requirements are supposed to retrieve the requested data from the brokers, avoiding activating data transmissions from the sensors.
- Traffic congestion in the brokers: Due to a myriad of IoT clients connecting to the network for various sensing services, the broker will be blocked if there are too many data retrieve requests, leading to serious service delay and degraded QoS. Therefore, the load balance between brokers and sensors are crucial and necessary.
- Battery conditions of the sensors: The sensors are always battery-powered, and it is impossible to recharge them before replacement. In this context, when the sensor nodes are in the low energy-level, they are more willing to transmit the sensed data to the nearby brokers to save energy, so as to avoid frequent repeated end-to-end data transmissions.

When implementing the in-network caching in IoT, the above mentioned issues need to be jointly considered. In the meanwhile, the unique features of the specific IoT applications should also be captured.

Aiming at the aforementioned issues, the following impacting factors should be encapsulated in the implementations of the in-network caching scheme in IoT, listed as follows:

1) The fresh time for each data;
2) The average distance between the clients and the resource-hosting sensors;
3) The delay requirements of the clients requesting the specific sensing services;
4) The number of times that the clients request the data, which is used to reflect the traffic load of the resource-hosting sensors;
5) The energy level of the sensors.

Remarkably, these factors are coupled with each other, and their tradeoffs should be carefully evaluated. For example, the energy level of the sensors may fail to satisfy the critical delay requirements of clients, which prompts us to make a preferable balance between them.

B. THE PROPOSED IN-NETWORK CACHING SCHEME

Focusing on the impacting factors shown above, we propose the energy-efficient dynamic in-network caching scheme to achieve the green IoT. More details are illustrated as follows:

- **Step 1: Divide the feasible value range of each impacting factor into multiple levels, and decide the value level for each factor.** As mentioned earlier, there are five impacting factors for each data and the corresponding resource-hosting sensor node, i.e., fresh time, average distance between clients and the resource-hosting node, delay requirements, the number of data requests of each sensor node and battery energy. Suppose that there are a total of N data resources in the considered IoT scenario. For the n-th data, the values of the five impacting factors are denoted as f_n^i, $i = 1, 2, \ldots, 5$, respectively. The value of each impacting factor is divided into several levels, according to the number of divided levels and the current values of f_n^i, $i = 1, 2, \ldots, 5$, $n = 1, 2, \ldots, N$. For example, the battery energy of the sensor nodes is divided into 5 levels, and higher level indicates there remains more energy in the sensors.

- **Step 2: Determine the weight for each impacting factor.** For the specific IoT applications and the sensed data resources, they may give priority to different impacting factors. In order to assign different importance for each considered factor, the weights for them can be correspondingly determined, which largely depends on the network states and user requirements. For the n-th data, the weights belonging to the five types of impacting factors are denoted as w_n^i, $i = 1, 2, \ldots, 5$, respectively, and the sum of these weights satisfies $\sum_{i=1}^{5} w_n^i = 1$.
- **Step 3: Formulate the cost function to decide whether the data should be cached or not.** From the above, a cost function is formulated to decide whether the data should be cached in the brokers or not. Each term in the cost function is the product of the value level and its corresponding weight, and the sum of these terms forms the cost function. Following this principle, the cost function is established as

$$C_n = \sum_{i=1}^{5} w_n^i f_n^i, \quad n = 1, 2, \ldots, N. \tag{1}$$

If the value of C_n is lower than the pre-defined caching threshold τ_c, this data should be cached in the nearby brokers; otherwise, the requested data should be directly retrieved from the resource-hosting sensors.

In Step 1, the number of value levels for different impacting factors can be dynamically adapted, depending to the value range of these factors. If the value is varied in a larger range, it can be divided into more levels. In the proposed caching scheme, the energy saving of the senors should not be at the cost of increasing the time delay of clients. Therefore, in Step 2, in order to guarantee the time delay of the clients in the acceptable range, the weight for the client's delay requirement can be a variable value. For each specific IoT sensing application, there is a timer threshold to guarantee the fresh time of the sensed data. If the time delay requirement of the clients is comparable to the given timer threshold, the weight for the time delay can be set as a moderate value. On the contrary, if the delay requirement is very critical, the weight for time delay can be set as a considerably large value, so that the value of the cost function is larger than the pre-defined caching threshold τ_c and the requested data should be retrieved from sensors, leading to less time latency. The details of the proposed caching schemes are summarized in Algorithm 1.

Algorithm 1 The Proposed Energy-Efficient Dynamic In-Network Caching Scheme

1) **Initialization:** the feasible value ranges of impacting factors, including
   - fresh time;
   - average distance between clients and the resource-hosting sensor node;
   - delay requirements;
   - the number of data requests of each sensor node;
   - battery energy;
2) **Input:**
   - the current value of each impacting factor;
   - the number of divided value level;
   - the caching threshold τ_c;
3) Determine the weight for each impacting factor;
4) **For each data resource**
   - Formulate the cost function for the data according to Step 3;
   - If the value of cost function is less than τ_c, cache the data;
   - Otherwise, retrieve the data from the resource-hosting sensor node.
5) **End**
6) **Output:** the caching decision for each data.

Remark 2: The proposed in-network caching scheme can be applied into the industrial application case, where the frequency of the access to the sensor nodes changes in time and the request popularity of sensed data resources is time-varying. To tackle this problem, the fresh time for each data resource is adopted under the proposed in-network caching scheme, and it can reflect whether the cached data is popular or not for the observed time period. If the requested data resource is popular and requested by many clients, the fresh time of the data is set to a large value, and vice versa. Moreover, a timer is equipped for each broker to count the fresh time and judge whether the current data is fresh or popular to the client. In addition, if some emergency conditions happen or the alarms are triggered, there can be multiple data requests bursting in a short time period. Confronting with these situations, the time delay requirements for obtaining the data and the congestion states of sensor nodes are two important factors that need to be jointly considered. In this paper, these two factors are included when designing the caching scheme, where the weights for these two factors can be set as a larger value in the cost function to cope with the emergency or alarm conditions.

Commonly, the in-network caching scheme is executed by the central controller, since it has the ability to collect the data requests from all clients periodically and it is equipped with strong computational capacity. Under the proposed in-network caching scheme, the detailed combined retrieving procedures are illustrated as follows, as shown in Fig. 3. In order to guarantee consistency between the cached data and

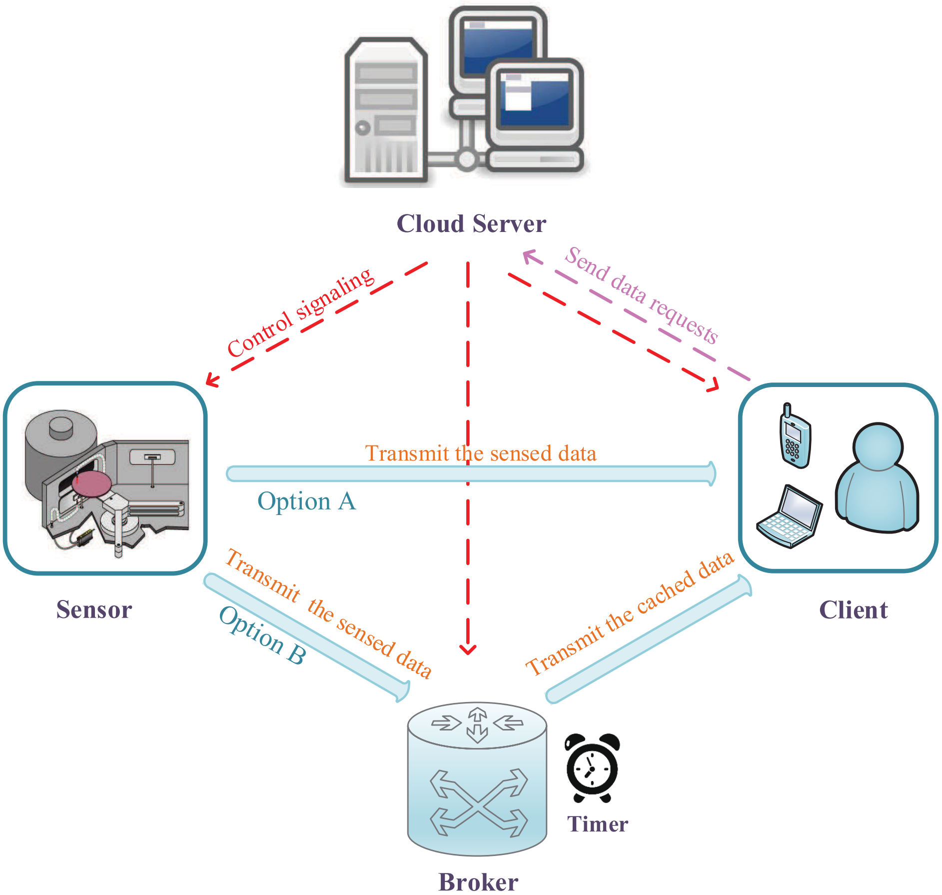

FIGURE 3. The combined retrieving procedures under the proposed in-network caching scheme.

the requested data, there is a timer for the broker to guarantee the freshness of the sensed data. At the beginning of each time slot, the timer starts. The central controller performs the energy-efficient in-network caching scheme. Under the caching decisions made by the controller, if the broker needs to cache the sensed data, it will generate a uniform resource identifier (URI), and the sensor can send the sensed data to the broker via this URI. Then, the clients, whose requested data resources are cached in the brokers, are informed to retrieve the requested data via the given URI. Otherwise, the clients will access to the sensors directly to obtain the requested data as in the traditional manner. Note that the caching and retrieving procedures are supposed to be completed within the stipulated timer threshold so as to guarantee the freshness of the collected data.

IV. CASE STUDY

In this section, we show the simulation results of the case study designed for the proposed energy-efficient in-network caching scheme in the IoT scenario. To present the effectiveness of the proposed caching scheme in terms of energy consumption, the sensor-to-client transmission scheme is adopted as the benchmark. Under the benchmark strategy, without the assistance of brokers, the resource-hosting sensor nodes directly transmit the requested data resources to clients.

In Figs. 4 and 5, we show the total energy consumption of the sensor nodes under different delay requirements and distances between clients and sensor nodes, respectively. From these figures, conclusion has been reached that the proposed in-network caching scheme is superior to the benchmark in terms of energy saving, since multiple copies of the

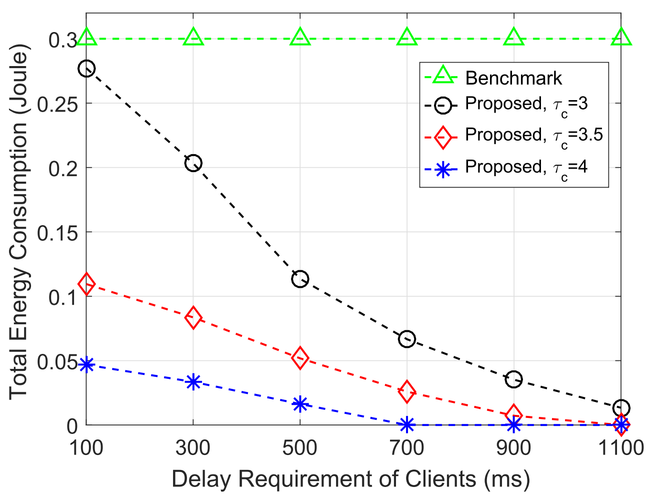

FIGURE 4. The total energy consumption of the sensor nodes under different time delay requirements of clients.

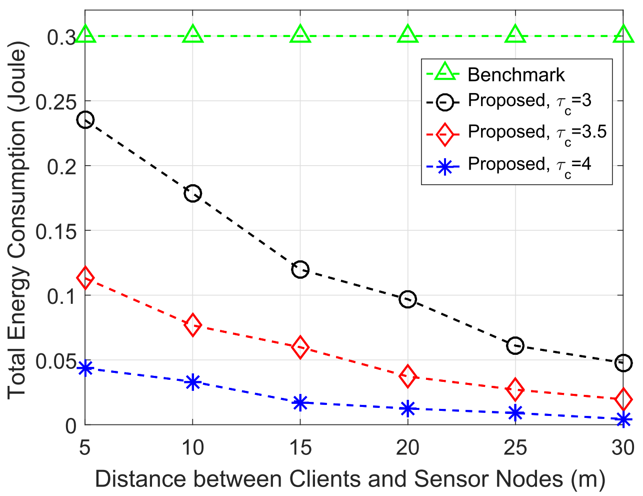

FIGURE 5. The total energy consumption of the sensor nodes under different distances between clients and sensor nodes.

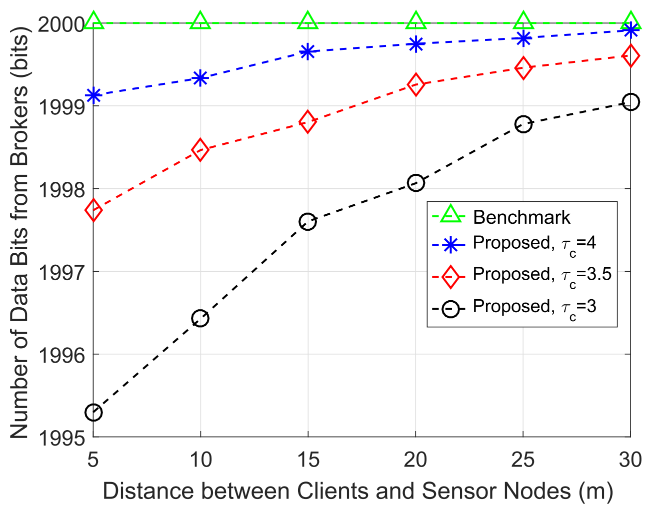

FIGURE 7. The number of the data bits retrieved from brokers under different distances between clients and sensor nodes.

sensed data are cached in the brokers and the frequent end-to-end transmissions of the resource-hosting sensor nodes are avoided. As the time delay requirement becomes looser, more sensed data will be offloaded at brokers, and the data transmissions from the sensor nodes are reduced, leading to less energy consumption. When the time delay requirement is relaxed enough, the total energy consumption of the sensors tends to be steady. We can also find that if the distances between the data-requesting client and the resource-hosting node is long, the data is more likely to be retrieved from the broker, and thus the total energy consumption of the sensor nodes will be reduced. Additionally, with larger caching threshold τ_c, the total energy consumption of the sensors reduces. The reason for this trend is that, if τ_c grows, the sensed data are more likely to be cached in the brokers, and more clients have the access to these data resources via brokers, leading to less energy consumption.

In Figs. 6 and 7, we show the number of the data bits that can be retrieved from brokers under varying delay

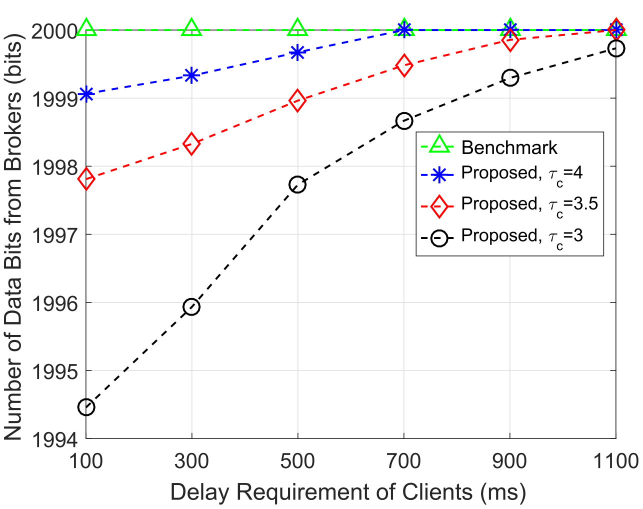

FIGURE 6. The number of the data bits retrieved from brokers under different time delay requirements of clients.

requirements and distances between clients and sensor nodes. Firstly, it can be seen that the proposed in-network caching scheme results in more offloaded data resources than that of the benchmark scheme. Moreover, with larger time delay requirements of clients and reduced distances, more sensed data can be found in the local cache of the brokers. We can also observe that, as τ_c grows, more data traffics are offloaded to brokers. In this case, the sensor nodes will bring less energy consumption for data transmission, which validates the conclusions drawn from Figs. 4 and 5 as well.

V. CONCLUSION AND FUTURE DIRECTION

By means of caching multiple copies of the sensed data and suppressing unnecessary end-to-end communications between the sensor nodes and clients, the in-network caching in IoT exhibits strong potentials to reduce energy consumption of these nodes, and hence prolongs the life span of the IoT. In this paper, we reviewed the fundamentals of the in-network caching for the green IoT, and proposed the in-network caching scheme to minimize the total energy consumption of the power-constrained sensor nodes. The numerical results shown in the case study demonstrated that the proposed caching scheme is superior to the traditional sensor-to-client scheme, and more data can be founded in the local cache of the brokers with looser delay requirements. In order to implement the in-network caching for the green IoT, the future research directions are summarized as follows:

- The limited storage of the broker should be more effectively utilized. Though the data sensed by the sensors are small-sized, the massive number of the data resources will demands a huge storage capacity. As a consequence, how to effectively utilize the limited storage capacity of the brokers is an important issue, and more advanced caching schemes are desired, such as the coded caching schemes proposed in CDNs [33].
- The security issue of the sensed data should be guaranteed. With caching the sensed data in the distributed

brokers, it poses great threat in flexible authorization and identity verification [34]. Thus, the access control is more challenging to achieve, and the unlicensed clients are more likely to obtain the confidential and private data. To this end, when implementing in-network caching in IoT, it is crucial to perform some access control mechanisms to prevent the illegal access.

- The edge computing can be adopted to achieve the in-network caching in IoT. As illustrated previously, the central controller is responsible for realizing the in-network caching scheme, and the computation load will be considerable large due to a variety of sensing services, leading to unacceptable demand for computation resource. In this context, some nodes for edge computing can be deployed nearby the clients, so that more computational tasks can be offloaded at these nodes and the severe traffic load of the central controller can be alleviated [35].
- The brokers can be much smarter to provide the improved IoT services. At present, the broker is only equipped with the cache capacity, and then forwards the cached data to clients. In the future, more complicated and intelligent data analysis functions can be realized by the smart brokers, such as data cleaning, data clustering and data predictions [36].

REFERENCES

[1] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of Things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 10–16, Jun. 2017.

[2] M. Tang, L. Gao, and J. Huang, "Communication, computation, and caching resource sharing for the Internet of Things," *IEEE Commun. Mag.*, vol. 58, no. 4, pp. 75–80, Apr. 2020.

[3] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[4] R. Yu, G. Xue, V. T. Kilari, and X. Zhang, "The fog of things paradigm: Road toward on-demand Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 48–54, Sep. 2018.

[5] J. Li, M. Dai, and Z. Su, "Energy-aware task offloading in the Internet of Things," *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 112–117, Oct. 2020.

[6] H. Tran-Dang, N. Krommenacker, P. Charpentier, and D.-S. Kim, "Toward the Internet of Things for physical Internet: Perspectives and challenges," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4711–4736, Jun. 2020.

[7] J. A. Ansere, G. Han, L. Liu, Y. Peng, and M. Kamal, "Optimal resource allocation in energy-efficient Internet-of-Things networks with imperfect CSI," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5401–5411, Jun. 2020.

[8] F. K. Shaikh, S. Zeadally, and E. Exposito, "Enabling technologies for green Internet of Things," *IEEE Syst. J.*, vol. 11, no. 2, pp. 983–994, Jun. 2015.

[9] O. Cetinkaya, D. Balsamo, and G. V. Merrett, "Internet of MIMO things: UAV-assisted wireless-powered networks for future smart cities," *IEEE Internet Things Mag.*, vol. 3, no. 1, pp. 8–13, Mar. 2020.

[10] S. Vural, P. Navaratnam, N. Wang, C. Wang, L. Dong, and R. Tafazolli, "In-network caching of Internet-of-Things data," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 3185–3190.

[11] Z. Piao, M. Peng, Y. Liu, and M. Daneshmand, "Recent advances of edge cache in radio access networks for Internet of Things: Techniques, performances, and challenges," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1010–1028, Feb. 2019.

[12] B. Chen, L. Liu, M. Sun, and H. Ma, "IoTCache: Toward data-driven network caching for Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10064–10076, Dec. 2019.

[13] X. Sun and N. Ansari, "Traffic load balancing among brokers at the IoT application layer," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 489–502, Mar. 2018.

[14] M. Zhang, B. Hao, R. Wang, and Y. Wang, "A pre-caching strategy based on the content relevance of smart Device's request in information-centric IoT," *IEEE Access*, vol. 8, pp. 75761–75771, 2020.

[15] J. Yao and N. Ansari, "Joint content placement and storage allocation in C-RANs for IoT sensing service," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1060–1067, Feb. 2019.

[16] J. Xu, K. Ota, and M. Dong, "Energy efficient hybrid edge caching scheme for tactile Internet in 5G," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 2, pp. 483–493, Jun. 2019.

[17] H. Wei, H. Luo, Y. Sun, and M. S. Obaidat, "Cache-aware computation offloading in IoT systems," *IEEE Syst. J.*, vol. 14, no. 1, pp. 61–72, Mar. 2020.

[18] F. Ganz, R. Li, P. Barnaghi, and H. Harai, "A resource mobility scheme for service-continuity in the Internet of Things," in *Proc. IEEE Int. Conf. Green Comput. Commun.*, Nov. 2012, pp. 261–264.

[19] M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud RAN," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6118–6131, Sep. 2016.

[20] T. X. Tran, D. V. Le, G. Yue, and D. Pompili, "Cooperative hierarchical caching and request scheduling in a cloud radio access network," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2729–2743, Dec. 2018.

[21] S. M. Azimi, O. Simeone, A. Sengupta, and R. Tandon, "Online edge caching and wireless delivery in fog-aided networks with dynamic content popularity," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1189–1202, Jun. 2018.

[22] Y. Jiang, M. Ma, M. Bennis, F.-C. Zheng, and X. You, "User preference learning-based edge caching for fog radio access network," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1268–1283, Feb. 2019.

[23] X. Zhang, T. Lv, W. Ni, J. M. Cioffi, N. C. Beaulieu, and Y. J. Guo, "Energy-efficient caching for scalable videos in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1802–1815, Aug. 2018.

[24] S. Zhang, W. Sun, and J. Liu, "Spatially cooperative caching and optimization for heterogeneous network," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11260–11270, Nov. 2019.

[25] X. Xu and M. Tao, "Modeling, analysis, and optimization of coded caching in small-cell networks," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3415–3428, Aug. 2017.

[26] J. Li, S. Chu, F. Shu, J. Wu, and D. N. K. Jayakody, "Contract-based small-cell caching for data disseminations in ultra-dense cellular networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 5, pp. 1042–1053, May 2019.

[27] L. Qiu and G. Cao, "Popularity-aware caching increases the capacity of wireless networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 173–187, Jan. 2020.

[28] R. Sun, Y. Wang, N. Cheng, L. Lyu, S. Zhang, H. Zhou, and X. Shen, "QoE-driven transmission-aware cache placement and cooperative beamforming design in cloud-RANs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 636–650, Jan. 2020.

[29] W. Ejaz, M. Naeem, A. Shahid, A. Anpalagan, and M. Jo, "Efficient energy management for the Internet of Things in smart cities," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 84–91, Jan. 2017.

[30] X. Sun and N. Ansari, "Dynamic resource caching in the IoT application layer for smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 606–613, Apr. 2018.

[31] F. Gabry, V. Bioglio, and I. Land, "On energy-efficient edge caching in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3288–3298, Dec. 2016.

[32] D. Niyato, D. I. Kim, P. Wang, and L. Song, "A novel caching mechanism for Internet of Things (IoT) sensing service with energy harvesting," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[33] D. Ko, B. Hong, J. H. Lee, and W. Choi, "Optimal file storing with cache memory in amorphous femto helper aided networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[34] R. Li, H. Asaeda, J. Li, and X. Fu, "A verifiable and flexible data sharing mechanism for information-centric IoT," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[35] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of Things with edge computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan. 2018.

[36] L. Dong and G. Wang, "Enhanced in-network capabilities of information-centric networks for emerging IoT applications," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Dec. 2016, pp. 573–577.

YUAN REN received the B.E. degree in information engineering and the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications, Beijing, China, in 2010 and 2017, respectively. He is currently a Lecturer with the School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an, China. His research interests include green communications, wireless caching, and cooperative communications.

XUEWEI ZHANG received the Ph.D. degree in information and communication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2020. She is currently an Associate Professor with the School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an, China. Her research interests include wireless edge caching, resource allocation, and heterogeneous networks.

TING WU received the B.E. degree in electronic and information engineering from Shangluo University, Shangluo, China, in 2018. She is currently pursuing the M.S. degree in electronics and communication engineering with the Xi'an University of Posts and Telecommunications, Xi'an, China. Her research interests include resource allocation and cooperative communications.

YIXUAN TAN received the B.E. degree in electronic and information engineering from the Shaanxi University of Technology, Hanzhong, China, in 2018. She is currently pursuing the M.S. degree in electronics and communication engineering with the Xi'an University of Posts and Telecommunications, Xi'an, China. Her research interests include non-orthogonal multiple access and cooperative communications.