An Edge-Computing Paradigm for Internet of Things over Power Line Communication Networks

Yuwen Qian, Long Shi, Jun Li, Xiangwei Zhou, Feng Shu, and Jiangzhou Wang

ABSTRACT

Power line communication (PLC) technology has created a niche use in the Internet of Things (IoT) by offering flexible and reliable connection among power-driven IoT devices/sensors over existing wired networks. In IoT over PLC networks, massive real-time data generated by the ever-growing connected devices will eventually pose an overwhelming burden on the IoT cloud, which in turn severely degrades the network performance. To cope with these issues, edge computing (EC) has emerged as a complement to cloud computing, aiming at offloading a portion of computing in the cloud to the network edges closer to the IoT devices. However, confronting a practical scenario that some electrical devices cannot communicate with wireless and mobile networks directly, existing EC paradigms may not be directly applied to IoT over PLC networks. In this paper, we propose a novel EC-IoT over PLC paradigm to reduce the transmission latency while migrating a portion of computing from the cloud to the edges. First, we develop a distributed EC platform to serve terminal users (TUs) in different IoT systems with various IoT services. Second, we put forth a cache-enabled scheme to store the popular contents from the cloud and edge sensors to reduce redundant data transmissions between TUs and the cloud. Finally, our experimental results demonstrate that the proposed EC-IoT over PLC network can significantly reduce energy consumption and transmission latency.

INTRODUCTION

The Internet of Things (IoT) is a revolutionary technology that connects people, machines, and devices together to the Internet [1]. IoT systems are capable of providing terminal users (TUs) and devices with seamless information transmissions and real-time decisions. In recent years, research on IoT related topics has surged in both academia and industry. A vast variety of IoT applications have been deployed in consumer, business, and infrastructure spaces, such as smart homes, smart grids, and environmental monitoring.

Generally, it is difficult for the IoT devices to execute tasks that demand intensive computation (e.g., data analysis and image processing) due to its limited power supply and computing capability. To deal with the shortage of computational resources, the cloud is employed to offer remote computing services that process the data generated by distributed IoT devices. The enticing merits of cloud computing include minimal management effort, rapid elasticity, and ubiquity. Thanks to these benefits, cloud computing has been widely accepted in IoT systems [2].

Nevertheless, cloud computing in the IoT has its limitations. For instance, cloud services may not be able to directly obtain local contextual information, such as real-time user locations, local network conditions, and behavior information of mobile users [3]. As a result, cloud computing can hardly meet the requirements of delay-sensitive applications, such as mobility support and location awareness [4]. To cope with these problems, edge computing (EC) has emerged as a promising technology for processing, storing, and executing applications locally rather than sending all to the cloud [5]. A typical use of the EC for mobile cellular networks is referred to as the mobile EC, where mobile subscribers can access computing services in the proximity of a radio access network [6]. As such, mobile EC enables business oriented EC to provide delay-sensitive and context-aware applications.

As smart homes and smart grids prevail, more and more electrical devices in our daily life will become intelligent IoT terminals, resulting in a huge consumption of spectrum resources. Meanwhile, some remote or rural areas are not fully covered by mobile and wireless networks due to economic and technical reasons. To address these issues, power line communication (PLC) can enhance connections between smart electrical devices and the Internet via alternating current (AC) meters, sensors, etc., for saving limited spectrum resources [7], [8]. In addition, power line networks are the most-covered networks in the world. Therefore, the PLC network can be deployed in remote areas upon their existing power grids by installing PAUs, routers, and PLC modems. Furthermore, PLC is a competitive solution for information transfer in smart grid, owing to its advantages of easy use, low cost, and simple installation [9]. Undoubtedly, as smart electrical devices proliferate, there is a growing trend that PLC will play a crucial role in IoT systems.

Existing EC schemes in the IoT are generally accessed through cables, Wi-Fi, or base stations rather than PLC components such as AC meters.

Digital Object Identifier:
10.1109/MNET.001.1900282

Yuwen Qian, Jun Li, and Feng Shu are with Nanjing University of Science and Technology; Long Shi is with Singapore University of Technology and Design; Xiangwei Zhou is with Louisiana State University; Jiangzhou Wang is with the University of Kent. The corresponding authors are Jun Li and Long Shi.

0890-8044/20/$25.00 © 2020 IEEE

In this context, a new EC paradigm that caters to IoT services in PLC networks is urgently needed. Recently, EC-aided PLC networks have been designed in [10] to offer real-time services for large scale networks such as the smart grid and manufacturing industries. Nevertheless, the EC scheme proposed in [10] has been used for a specific smart factory, but the generalized architecture of EC-IoT over PLC was not presented. Thus, an efficient and generalized EC-IoT paradigm over PLC remains challenging.

In this paper, we propose a novel EC paradigm for the IoT over PLC networks, namely, EC-IoT over PLC, consisting of edge servers (ES) that host EC for IoT applications, the sensor networks that perceive the environment, and the access networks that enable the terminal devices to access edge networks. In the proposed EC-IoT, we further develop an EC software platform to host local computing of the context-sensitive IoT applications in a distributed way. In particular, threefold contributions are listed as follows:

- An EC-aided IoT paradigm over PLC is proposed, which can host local computing of context-sensitive IoT services for TUs in PLC networks. First, we design a PLC sensor network for the IoT, where mobile and wireless sensors can access the edge network through communicating with nearby electrical devices. Second, we develop a PLC access network operated by a multiple-input multiple-output (MIMO) PLC access unit to cooperatively provide network access for TUs.
- A distributed EC software platform is developed comprising a virtualization layer, resource management layer, and sensor layer. In the virtualization layer, a virtualization scheme is utilized to offer TUs with the IoT services. In the sensor layer, we propose a bi-directional caching system for EC-IoT over PLC, including a download-caching sub-system and upload-caching sub-system, to reduce redundant transmissions over the backhaul.
- An EC-IoT over PLC system is implemented to evaluate the real-time performance. Our experimental results show that the proposed system can not only alleviate network traffic in the backbone network, but also reduce energy consumption and transmission latency.

The remainder of this paper is organized as follows. The next section describes the architecture of EC-IoT over PLC. Then we introduce the core designs of EC-IoT over PLC. Next, we design a distributed software platform for the proposed system, and then we develop a testbed for EC-IoT over PLC. The final section concludes this paper.

Architecture of EC-IoT over PLC

Generally, a power grid system consists of generation, transmission, distribution, and consumption. The power is first generated in the high voltage (HV) grids and then distributed by the medium voltage (MV) and low voltage (LV) grids to regional areas. Note that the LV and MV grids can be exploited as the PLC networks, where PLC protocols are applied for the in-home (IH) users to connect with the PLC access networks [11].

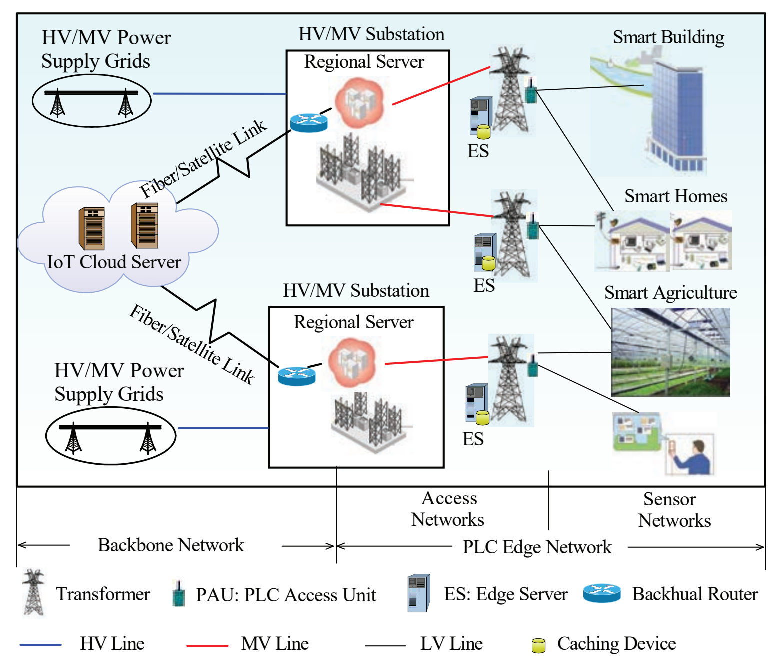

FIGURE 1. The proposed EC-IoT over PLC network, where EC is implemented in the access and sensor networks and the regional server can store and forward data for the ESs.

As shown in Fig. 1, the proposed EC-IoT over PLC consists of the backbone network, EC-aided PLC access networks, and PLC sensor networks[1]. The backbone network can cover a large area by employing the fiber/satellite links. In addition, cloud computing servers are installed in the backbone network, and the broadband signals are forwarded to these cloud servers through the backhaul routers.

The EC-aided PLC access networks, located in the MV to LV grids, provide a pathway for signal transmission from the backbone network to TUs. In the access network, we install the PLC access units (PAUs) at the transformers to provide network access to those smart grid systems. Notably, we deploy an ES in the access network to aggregate traffics generated by sensors of IoT applications, such as autonomous vehicles, implanted medical devices, and mobile devices. Each ES connects with the backbone network with the MV power lines, which can be treated as a broadband PLC channel.

The PLC sensor networks, resided in the LV grids, operate over narrow-band PLC channels. In general, there are two types of devices in PLC sensor networks. First, smart electrical devices connected in the power grid are operated by TUs to obtain IoT services. Second, wireless and mobile sensors are used to collect the related data from the environment. Furthermore, equipped with both PLC and wireless interfaces, these sensors can communicate with smart electrical devices. Each gateway in the sensor network connects with the nearest PAU to provide network service for the wireless sensors and electrical devices.

Overall, the proposed EC-IoT system is developed to enable ubiquitous, convenient, and on-demand network access locally to a pool of computing resources (e.g., networks, servers, stor-

---

[1] The proposed architecture builds upon our finalized project, which was funded by State Grid Corporation of China and Hong Kong Productivity Council.

age, applications, and services) that can be rapidly provisioned and released with minimal management effort. As later sections will elaborate, EC-IoT over PLC benefits from the integration of EC to enable IoT services locally.

CORE DESIGNS OF EC-IOT OVER PLC

To describe EC-IoT over PLC, we first clarify the cooperations among the ES, sensors, and smart devices, and then we specify the designs of the PLC access and sensor networks.

COMPUTING AT EDGE DEVICES

In the proposed architecture, edge devices fall into three categories: sensor cluster, smart electrical device, and edge server.

First, a sensor cluster is a set of sensors sharing their resources. Note that the sensors are clustered according to distinct IoT services and the sensors in the same cluster correspond to the same IoT service. Each cluster is registered in the ESs in the access networks. When a new sensor joins any sensor cluster to launch an application, this new member sends its private information, such as memory size, computational availability, and battery level, to the ESs. Based on the information, the ESs decide to reject or accept the sensor's registration.

Second, a smart electrical device is operated by the TUs to obtain the computing services from the ESs. To be specific, the smart devices can communicate with the mobile terminals of TUs, such that the TUs can remotely control these smart electrical devices.

Third, as the core component in the access network, the ESs respond to the TU requests and provide the corresponding IoT services by collaborating with sensor clusters and smart electrical devices. To this end, each ES enables the access network to create a PLC/wireless access point, and schedules the nearby sensor clusters to offer sensing services. To cope with the hardware fault, we adopt a scanning scheme that enables sensors and devices to contact with the associated ESs periodically. The administrator will be notified in case of any system error.

With these edge devices, EC-IoT over PLC provides IoT services by the following steps. In the first step, the ESs expose the context of the network access point (NAP) to the smart devices. Note that the context of NAP, including the IP address, socket, authority information, and operation system version, is used to provide access for TUs or smart devices. In the second step, a service ID is assigned to each IoT service. TUs specify the service ID for each request, such that the request can be processed by the corresponding IoT service. Upon receiving the IoT requests from each TU, the smart device associated with the requesting TU delivers the service ID as well as the sizes of input and output data to nearby ESs with the help of the context of NAP. In the third step, according to the service ID, the ESs first analyze the requests from TUs, such as scaling the sensing capability of the sensors, and then schedule the input task to the corresponding sensor cluster to execute the sensing. As such, with the efficient cooperation among the ESs in the access networks and sensors/smart electrical devices in the sensor networks, the proposed EC-IoT over PLC can provide TUs with their requested IoT services.

ACCESS NETWORKS

The access networks for EC-IoT over PLC rely on the characteristics of PLC networks, where the PAU with a PLC modem is installed as the network gateway. The role of the PAU is to inject signals into the LV power lines, regenerate and amplify signals. From [12], the PLC modem can extract signals from the MV power lines and make them compatible with IEEE 802.11a/b/g standards. Therefore, with the PLC modem, each PAU can provide network access for smart devices. We stress that the PAUs connect directly with the ESs over the high-speed cables, such that the ESs can reliably and quickly provide TUs with the context of NAP. Moreover, the PAUs serve as not only the network access points but also the transmission relays. To improve the network resiliency, we can employ multiple PAUs in sensor networks and access networks to cooperatively serve the TUs and sensors.

In the access networks, each PAU interacts with the smart devices within its covering area and collects the data from nearby sensor clusters. Then, each PAU aggregates and compresses the collected data in order to improve the efficiency of storage and transmission. In addition, each access network deploys a backhaul router as the gateway to directly connect the PAUs with the cloud servers.

At the TU side, the narrow-band PLC is adopted to connect the in-home sensor networks with IoT applications. A critical issue is that the narrow-band PLC suffers from the severe signal fading and impulsive noises over the PLC channel, resulting in unreliable communications from the PAUs to TUs. To improve transmission reliability, we employ the multiple-input multiple-output (MIMO) PLC technique that brings significant diversity gain. Generally, the MIMO PLC channel is identified as the combination of three-phase power lines, comprising phase ($\mathcal{P}$), neutral ($\mathcal{N}$), and protective earth ($\mathcal{PE}$) [13]. Signals are carried by the voltage difference between any pair of power lines. In this way, three transmit-receive port pairs, i.e., $\mathcal{P}$ to $\mathcal{N}$, $\mathcal{P}$ to $\mathcal{PE}$, and $\mathcal{N}$ to $\mathcal{PE}$, are generated. We remark that each port pair is equivalent to a transmit-receive antenna pair in wireless communication systems. The power lines that connect with switchers and variable-frequency devices are not capable of carrying IoT edge traffic.

SENSOR NETWORKS

We design a general sensor network that fuses wireless communications and PLC, allowing wireless and PLC sensors to communicate with each other. To connect the sensor networks, the PAUs in the access networks are equipped with both wireless and PLC interfaces, which are used to communicate with the smart electrical devices and wireless sensors. As an illustrating example, Fig. 2 shows a sensor network wherein smart devices act as either routers or terminal nodes.

In Fig. 2, $\mathcal{A}_i$ and $\mathcal{B}_i$ denote the charging sockets for electrical vehicles and smart power meter in smart house i, $i = 1, 2$ respectively. There are two scenarios for sensors in an electrical vehicle to access the edge network. In the first scenario, the electrical vehicle is charged in its owner's house. In this case, both $\mathcal{A}_i$ and $\mathcal{B}_i$ belong to the same sensor network. Then, $\mathcal{A}_i$ communicates with $\mathcal{B}_i$ to charge the electrical vehicle by checking the user ID of the vehicle. In the second scenario, the electrical vehicle does not belong to the sensor network of the owner's house. In this case, the electricity consumption data cannot be transmitted directly to the associated meter. For example, the vehicle of smart house 1 can be charged in the smart house 2 with $\mathcal{A}_2$. Then, $\mathcal{A}_2$ contacts its serving ES2 and sends the user ID. Then, ES2 forwards the electricity consumption data to ES1. After that, ES1 forwards the data to $\mathcal{B}_1$ for the electricity bill payment.

SOFTWARE PLATFORM OF EC-IoT OVER PLC

We design a software platform of EC-IoT over PLC to support an efficient and multi-tenancy computing host by the virtualization technology. With the software platform, the ES, sensors, and smart electrical devices cooperate to provide the EC services for TUs. First, we develop a virtualization scheme to provide IoT services for various applications, such as smart cars, smart agriculture, and smart home. Second, we develop application programming interfaces (APIs) for TUs to run the IoT applications. In addition, the APIs, open to third-party partners, can enable innovative applications and services toward subscribers and enterprises of the power grid. As shown in Fig. 3, we develop a three-layer software platform for EC-IoT over PLC.

IoT APPLICATION LAYER

The IoT application layer provides the user interfaces (UIs) and resource management. First, each TU sends its request through UIs, and its smart devices call common APIs to run the IoT applications. Second, the IoT application layer provides authority management. The private information of TUs can be protected through the authority management, although different TUs utilize the same devices, a set of common APIs, and ESs to run different IoT applications.

EDGE COMPUTING LAYER

This layer implements the virtualization to furnish the real-time EC service for IoT applications. The functions of this layer are two-fold. First, the computational resources are allocated to distinct IoT services according to different real-time requests from TUs. Second, these computational resources are managed in an efficient way. In view of the two functions, this layer is divided into two sub-layers as follows:

Virtualization Management Sub-Layer: This sub-layer focuses on the computational resource allocation in the first function. To this end, we use the virtualization to enable EC for IoT applications in a flexible, efficient, and scalable way. In particular, we implement containers in the operation systems (OS), e.g., X86 and ARMv7, as the virtual environment.

The container is a lightweight, portable, and high performance alternative for virtualization, since it can be rapidly deployed with near-native performance in CPU, memory, disk, and network. To support a pool of containers, we employ a container engine as a part of the OS kernel. Therefore, containers with the same engine run in the same OS kernel. The container engine can support the containers with software and hardware resources. This sub-layer runs containers by using Docker, which can run on Windows and Linux to perform the OS-level virtualization for developers. In practice, Docker has been widely used in the cloud systems [14]. Multiple Docker containers can be run simultaneously by a single OS kernel and are thus more lightweight than virtual machines. In this sense, compared with virtual

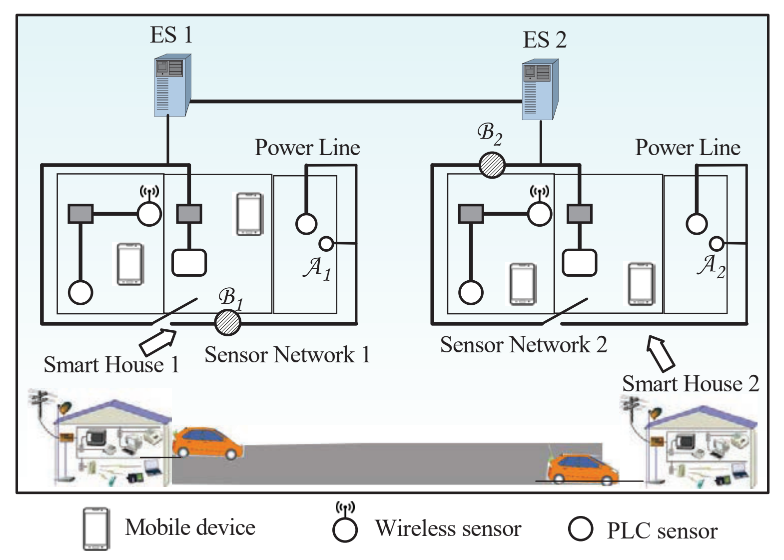

FIGURE 2. An example of a sensor network where wireless sensors are distributed in two neighboring rooms and PLC sensors are installed at power loads.

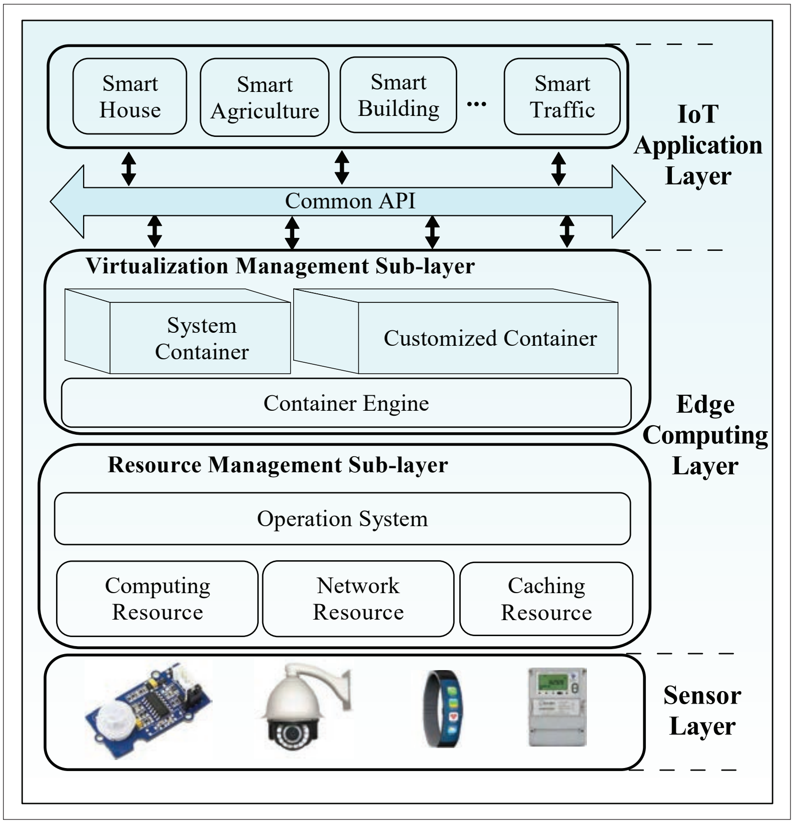

FIGURE 3. The software platform of EC-IoT over PLC.

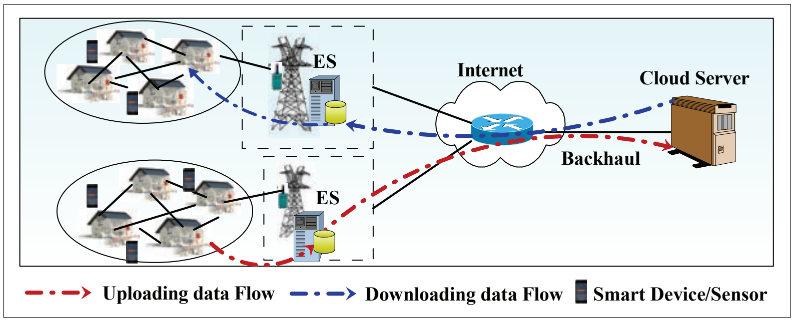

FIGURE 4. The EC-IoT over PLC, where the backhaul is used to transmit messages between the local network and the cloud.

machines, Docker is more suitable for the ESs that have limited resource capacity. In this way, this sub-layer can run hundreds of containers over a single OS to ensure the diversity of IoT services.

The containers are categorized into two classes: system containers and TU-customized container. Regarding the system containers, this sub-layer enables five primary services.

- Service Registry and Discovery: The service registry and discovery is developed to enable efficient management of applications and services at the edge. Initially, each new service is registered with the container to own a service ID. According to the service ID, the ES can run the services associated with the TU requests.
- User Request Monitor: The user request monitor listens to the network to capture the requests from TUs, and then extracts the network package submitted with the requests. In addition, the container can dispatch the requests to the nearby ESs according to the locations of requesting TUs.
- Traffic Offloading: The traffic offloading service first splits IoT applications into small tasks. Then, the delay-tolerant and computation-intensive small tasks are executed in the cloud, while the delay-sensitive tasks are processed in the local edge to ensure quick response.
- Location Awareness: The location awareness service is used to locate each connected device for location-based services and applications.
- Error Detection: The error detection service can capture the errors and unexpected exceptions in the execution of services. Then, the results are returned to the TUs.

Let us brief the function of the customized container. According to the TU requests, the third parties can develop the customized containers for their services in the edge. For example, a container can be customized to analyze the abnormal power equipment for a power equipment monitoring system.

Resource Management Sub-Layer: The resource management sub-layer is used to manage physical resources, network, and storage devices. First, the resource management can efficiently allocate resources of the ESs (e.g., memory, energy, and CPU). Second, the sub-layer provides network management for the ESs with the PLC and wireless communication protocols. With these protocols, the data generated by sensors and requests submitted from the TUs can be transmitted to the ESs and the cloud. Third, the sub-layer manages the storage devices embedded in the ESs to improve reliability and meet data integrity requirements.

Sensor Layer

In the sensor layer, a variety of sensors are deployed in the scattered areas according to the TU requests. These sensors collect real-time data and forward them to nearby ESs, and the ESs can schedule sensors according to their registered sensor IDs. To forward the popular contents to the terminal devices, we propose a cache-assisted sensor network by installing caches in the ESs as shown in Fig. 4, consisting of download-caching and upload-caching sub-systems.

In the download-caching sub-system, each ES first pre-downloads popular contents in its local cache from the cloud in off-peak hours, and then the TUs can download the contents from the ESs in peak hours. Therefore, the download-caching sub-system can alleviate redundant data transmissions from the remote cloud to the TUs by redirecting downloading requests to the ESs, thereby reducing the latency and energy consumption over backhaul. Note that the download-caching system can serve a large portion of the Internet contents, including software, documents, and on-demand streaming media.

In the upload-caching sub-system, each sensor uploads its generated data to the caches in the ESs in off-peak hours, and then the ESs deliver the requested data to the smart electrical devices and TUs in peak hours. The following two issues motivate the design of this upload-caching sub-system. First, sensors generate a volume of data through testing and monitoring the target objects. In contrast with limited memory in each sensor, each ES with larger cache memory can store the data of sensors. Second, the ESs can also package the data rarely requested by the TUs and upload to the cloud.

We illustrate the uploading cache by two application scenarios. First, consider the data reading from an air temperature sensor combined with a humidity meter. Due to the insufficient memory, the sensor can only store air temperature and humidity data for a few days. In this case, the sensor uploads the daily generated data that is beyond its memory capacity to the ESs with sufficient memory during off-peak hours. Then, the TUs can consult the outdated data stored in the ESs. Second, consider that the TUs request data generated by the sensors located in a different edge network. For example, a TU located in the edge network (called N1) requests the data collected form the sensors in a different edge network (called N2). In this example, the sensors in N2 first upload the required data by the TU to the ES in N2, and the ES further uploads the data to the cloud. Then, the cloud forwards the data to the ESs in E1 close to the TU. As a result, the TU can download the data from the nearby ESs.

Implementation of EC-IoT over PLC

Our experimental system of EC-IoT over PLC is deployed in the Gaochun district, Nanjing, China, which covers an area of around 80km^{2}. As shown in Fig. 5, we select 12 villages to deploy 35 ESs and ensure that a total of 4200 in-home networks can access the experimental system. In addition, we install five regional servers to coordinate communications among ESs and communications between ESs and the cloud. Smart meters for gas, water, temperature, and humidity are employed. We adopt MySQL in the cloud as the central database, and each ES also adopts MySQL to manage the data collected from the sensors. A data processing module is equipped in the smart devices to analyze the data collected from sensors. For the communications among sensors with the ESs, we employ Zigbee for wireless sensors and HomePlug AV2 compatible with IEEE 1901a for PLC. We deploy different IoT systems in different villages, such as automated meter reading systems, PLC based smart home systems, and smart agricultural systems. For each IoT system, we customize the containers to provide specified IoT services. For example, in the smart home systems, we customize several services using the temperature control container, light intensity control container, and smart guarding container. For comparison purpose, we adopt three distinct paradigms, i.e., IoT over PLC without EC (IoT over PLC), EC-IoT over PLC without cache (EC-IoT over PLC), and our proposed EC-IoT over cache-assisted PLC (EC-IoT over CPLC).

The experiments evaluate the three IoT paradigms with four performance metrics: the network traffic in the backbone network, the average packet delay, the energy consumption of sensors, and the transactions per second. First, let $\rho^{cloud} = \lambda M\bar{X}$ be the network traffic in the backbone network, where λ is the packet arrive rate, M is the number of PAUs, and X is the packet length. Second, the packet delay is defined as $D = T_1 + T_2 + T_3$, where T_1 is the average polling cycle time of the access networks, T_2 is the time used for offloading the packets from users to the ESs, and T_3 is the time used for offloading the packets from the ESs to the cloud. Accordingly, the following results are based on our experiments that run the systems with around 6,000 sensors over 100 days.

Fig. 6(a) shows the average network traffic in the backbone network with respect to the total transmission rate of IoT devices. First, we observe that EC-IoT over CPLC achieves the lowest network traffic among the three paradigms, since the EC hosts the TU requests locally and the proposed caching system reduces the data transmissions between the cloud and the devices. Compared with IoT over PLC, EC-IoT over CPLC reduces the network traffic in the backbone network more than 50 percent. This is due to the fact that the uploading cache receives nearly all the data from the sensors, which significantly reduces the network traffic. Second, the incremental of network traffic in EC-IoT over CPLC and EC-IoT over PLC is less significant than that in IoT over PLC as the data rate increases. This observation occurs because our software platform adopts a traffic offloading container to analyze the network traffic of the backbone network. When the network traffic of the backbone network exceeds a certain threshold, the container reduces the TU requests submitted to the cloud.

Fig. 6(b) compares the average packet delay of the three paradigms with respect to network traffic load of the PLC sensor network. Notably, the network traffic refers to the amount of data moving across a network at a given point of time. Our experiment adopts IEEE 802.3ah EPON as the access network protocol, where the network traffic load ranges from 0.05 to 0.9 [15]. We observe that EC-IoT over CPLC achieves the smallest packet delay among the three paradigms. The first reason is that the proposed upload-caching system alleviates the network traffic load in the access network by uploading packets from the sensors to ESs. As a result, the number of packet collisions in access networks reduces, which further decreases the packet delay. The second reason is that the software platform of EC-IoT over CPLC deploys the service registry and discovery container. With the aid of the container, the computational task can be dispatched to the proper ES directly in EC-IoT over CPLC, without polling all the ESs to host computation. This dramatically reduces the number of broadcasting packets in the edge networks.

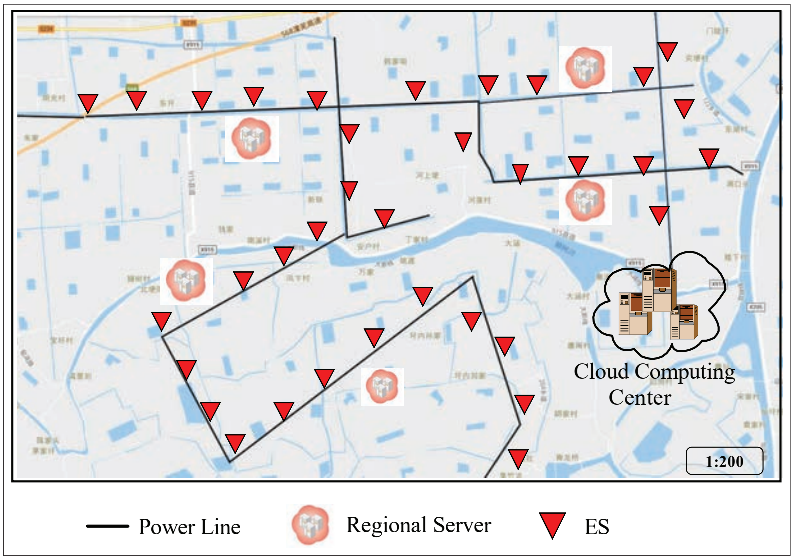

FIGURE 5. The deployment of EC-IoT over PLC in Gaochun, Nanjing, China.

Fig. 6(c) illustrates the average energy consumption of all sensors in the three systems against the network traffic load of the PLC sensor network. The transmit power of each sensor is obtained according to its transmission rate and channel gain between the associated PAU and itself. It is seen that EC-IoT over CPLC consumes much less energy than IoT over PLC and EC-IoT over PLC. For instance, when the network traffic load is 0.8, the energy consumption of IoT over PLC and EC-IoT over PLC is about four and two times that of EC-IoT over CPLC, respectively. The reason is that the proposed caching scheme can reduce the energy consumption by bringing the popular contents closer to the TUs and alleviating the redundant packet transmissions from the edge networks to the cloud.

Fig. 6(d) compares the number of transactions per second between EC-IoT over CPLC and IoT over PLC versus the running time. These transactions include request submissions from the TUs or data collection from the sensors. First, we observe that EC-IoT over CPLC completes more transactions than IoT over PLC. This is due to the fact that, compared with IoT over PLC, EC-IoT over CPLC conducts more transactions locally without uploading to the cloud. This observation further

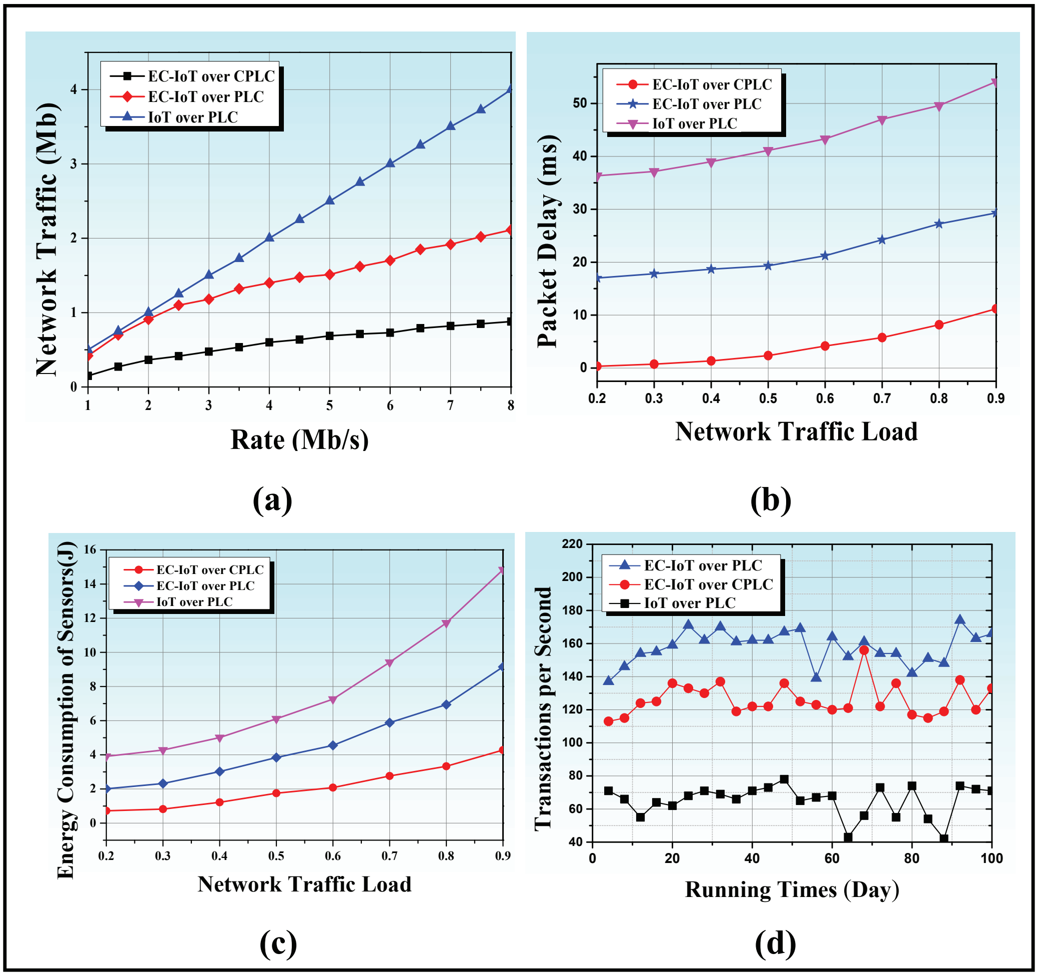

FIGURE 6. Experimental results of the IoT over PLC, EC-IoT over PLC, and EC-IoT over CPLC systems:(a) offloading network traffic versus rate in the backbone network; (b) average packet delay versus network traffic load; (c) average energy consumption of total sensors in the edge networks against network traffic load; (d) average processed transactions per second.

corroborates that the number of transactions in the IoT systems can be improved by the EC. Second, it is observed that EC-IoT over PLC achieves more transactions than EC-IoT over CPLC. This is due to the fact that, in comparison with EC-IoT over PLC, the caching system in EC-IoT over CPLC stores more data from sensors and thereby reduces the transactions on data collection.

CONCLUSIONS AND FUTURE WORK

We have proposed an edge computing paradigm over the PLC network to efficiently provide the TUs with various IoT services. To improve the communication reliability, we designed a MIMO PLC scheme for PAUs to cooperatively serve the TUs. By adopting the virtualization technology, we developed a software platform for EC-IoT over PLC to orchestrate the ESs, sensors, and smart devices to execute the TU requests. Furthermore, we designed a bi-directional caching scheme, with which terminal devices can download data from local caching devices and data generated by the sensors can be stored at the caching devices. In practice, we implemented the proposed EC-IoT over PLC in a district of Nanjing, China. Experimental results demonstrated that the proposed paradigm can significantly reduce the energy consumption of sensors and the transmission latency.

This work serves as a first foray into the design of EC-IoT over PLC. Many interesting directions follow this work and deserve further investigation. First, the environment in the IoT systems is time-varying in reality. Based on massive historical data of the environment, how to dynamically allocate the computing resources with the deep reinforcement learning methods remains open. Second, social attributes and social relations among the IoT devices can be explored to reduce the computation cost in the proposed caching system. To this end, a tentative solution is to select some key in-home networks to install the cache devices according to their social importance. Third, the security issues in the proposed diagram such as how to protect the privacy of TUs and design the secure protocol for the sensor networks deserve further investigation.

ACKNOWLEDGMENTS

This work is supported in part by the National Key R&D Program under Grant 2018YFB1004800, and in part by the National Natural Science Foundation of China under Grants 61872184, 61727802.

REFERENCES

[1] A. Fuqaha *et al.*, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, Jun. 2015, pp. 2347–76.

[2] 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, Jan. 2018, pp. 96–101.

[3] M. Gregori *et al.*, "Joint Transmission and Caching Policy Design for Energy Minimization in the Wireless Backhaul Link," *Proc. IEEE ISIT*, Hong Kong, Oct. 2015.

[4] W. Shi *et al.*, "Edge Computing: Vision and Challenges," *IEEE Internet Things J.*, vol. 3, no. 5, Oct. 2014, pp. 637–46.

[5] X. Sun and N. Ansari, "Edge-IoT: Mobile Edge Computing for the Internet of Things," *IEEE Commun. Mag.*, vol. 54, no. 12, Dec. 2016, pp. 22–29.

[6] H. Li, K. Ota, and M. Dong, "ECCN: Orchestration of Edge-Centric Computing and Content-Centric Networking in the 5G Radio Access Network," *IEEE Wireless Commun.*, vol. 25, no. 3, June 2018, pp. 88–93.

[7] F. Canete *et al.*, "A Channel Model Proposal for Indoor Power Line Communications," *IEEE Commun. Mag.*, vol. 49, no. 12, Dec. 2011, pp. 166–174.

[8] L. Li, K. Ota, and M. Dong, "When Weather Matters: IoT-based Electrical Load Forecasting for Smart Grid," *IEEE Commun. Mag.*, vol. 55, no. 10, Oct. 2017, pp. 46–51.

[9] A. Sarafi, I. Tsiropoulos, and P. Cottis, "Hybrid Wireless-Broadband over Power Lines: A Promising Broadband Solution in Rural Areas," *IEEE Commun. Mag.*, vol. 47, no. 11, Jun. 2009, pp. 140–147.

[10] E. Oyekanlu *et al.*, "Edge Computing for Industrial IoT and the Smart Grid: Channel Capacity for M2M Communication over the Power Line," *Proc. IEEE NIGERCON*, Owerri, Nigeria, Nov. 2017, pp. 2377–697.

[11] G3-PLC, "Open standard for Smart Grid Implementation [online]," PLC Open Standard Center, Technical Report, 2005.

[12] P. S. Henry, "Interference Characteristics of Broadband Power Line Communication Systems Using Aerial Medium Voltage Wires," *IEEE Commun. Mag.*, vol. 43, no. 4, Apr. 2005, pp. 92–98.

[13] A. Pittolo, A. M. Tonello, and F. Versolatto, "Performance of MIMO PLC in Measured Channels Affected by Correlated Noise," *Proc. IEEE ISPLC*, Glasgow, UK, May 2014, pp. 261–65.

[14] M. Alam *et al.*, "Mobile-Edge Computing Architecture: The Role of MEC in the Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 9, 2018, pp. 118–23.

[15] B. Rimal, D. Van, and M. Maier, "The Emerging Enernet: Convergence of the Smart Grid with the Internet of Things," *IEEE Ind. Appl. Mag.*, vol. 55, no. 2, Apr. 2017, pp. 192–200.

BIOGRAPHIES

YUWEN QIAN received the Ph.D. degree in automatic engineering from Nanjing University of Science and Technology, Nanjing, China, in 2011. From July 2002 to June 2011, he was a lecturer in the Automation School of Nanjing University of Science and Technology. Since May 2019, he has been an associate professor in the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, China.

LONG SHI received the Ph.D. degree in electrical engineering from the University of New South Wales, Sydney, Australia, in 2012. From 2013 to 2016, he was a postdoctoral fellow at the Institute of Network Coding, Chinese University of Hong Kong, Hong Kong SAR, China. From 2014 to 2017, he was a lecturer at Nanjing University of Aeronautics and Astronautics, Nanjing, China. Currently, he is a research fellow at the Singapore University of Technology and Design.

JUN LI received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, P.R. China in 2009. From January 2009 to June 2009, he worked at Alcatel Lucent Shanghai Bell as a research scientist. From June 2009 to April 2012, he was a postdoctoral fellow at the University of New South Wales, Australia. From April 2012 to June 2015, he was a research fellow at the University of Sydney, Australia. From June 2015 to now, he has been a professor at the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China.

XIANGWEI ZHOU received the B.S. degree in communication engineering from Nanjing University of Science and Technology, Nanjing, China, the M.S. degree in information and communication engineering from Zhejiang University, Hangzhou, China, and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2005, 2007, and 2011, respectively. Since August 2015, he has been an assistant professor with the Division of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, USA. Prior to that, he was an assistant professor with the Department of Electrical and Computer Engineering, Southern Illinois University, Carbondale, IL, USA, from 2013 to 2015, and a senior systems engineer with Marvell Semiconductor, Santa Clara, CA, USA, from 2011 to 2013.

FENG SHU received the Ph.D., M.S., and B.S. degrees from the Southeast University, Nanjing, in 2002, XiDian University, Xian, China, in 1997, and Fuyang Normal University, Fuyang, China, in 1994, respectively. From October 2003 to October 2005, he was a post-doctor researcher with the National Key Mobile Communication Lab at Southeast University. From September 2009 to September 2010, he was a visiting post-doctor at the University of Texas at Dallas. In October 2005, he joined the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China, where he is currently a professor and supervisor of Ph.D. and graduate students. He serves as an editor for a number of international journals, including *IEEE Systems Journal* and *IEEE Access*.

JIANGZHOU WANG is currently a professor and the former Head of the School of Engineering and Digital Arts at the University of Kent, U.K. He has published over 300 papers in international journals and conferences and four books in the areas of wireless mobile communications. He is a Fellow of the Royal Academy of Engineering, U.K., and a Fellow of the IEEE. He received the Best Paper Award at IEEE GLOBECOM2012. He was an IEEE Distinguished Lecturer from 2013 to 2014. He was the Technical Program Chair of the 2019 IEEE International Conference on Communications (ICC2019), Shanghai, the Executive Chair of the IEEE ICC2015, London, and the Technical Program Chair of the IEEE WCNC2013. He has served as an editor for a number of international journals, including *IEEE Transactions on Communications* from 1998 to 2013.