

# Edge-Computing-Based Channel Allocation for Deadline-Driven IoT Networks

Weifeng Gao , *Student Member, IEEE*, Zhiwei Zhao , *Member, IEEE*, Zhengxin Yu, Geyong Min , *Member, IEEE*, Minghang Yang, and Wenjie Huang

**Abstract**—Multichannel communication is an important means to improve the reliability of low-power Internet-of-Things (IoT) networks. Typically, data transmissions in IoT networks are often required to be delivered before a given deadline, making deadline-driven channel allocation an essential task. The existing works on time-division multiple access often fail to establish channel schedules to meet the deadline requirement, as they often assume that transmissions can be successful within one transmission slot. Besides, the allocation and link estimation incur considerable overhead for the IoT nodes. In this article, we propose an edge-based channel allocation (ECA) for unreliable IoT networks. In ECA, we explicitly consider the impact of allocation sequences and employ a recurrent-neural-network-based channel estimation scheme. We utilize link quality and retransmission opportunities to maximize the packet delivery ratio before deadline. The allocation algorithms are executed on edge servers such that: 1) the channel allocation can be updated more frequently to deal with the wireless dynamics; 2) the allocation results can be obtained in real time; and 3) channel estimation can be more accurate. Extensive evaluation results show that ECA can significantly improve the reliability of deadline-driven IoT networks.

**Index Terms**—Channel allocation, deadline-driven, edge computing, Internet-of-Things.

## I. INTRODUCTION

THE Internet of Things (IoT) has received much attention and has been increasingly deployed recently. In many IoT scenarios, such as smart home [1] and building monitoring [2], a low-power wireless network is often deployed to measure various kinds of sensor data, such as PM2.5, humidity, temperature, etc. [3], [4]. The sensor data are usually required to be delivered

Manuscript received January 3, 2020; accepted January 26, 2020. Date of publication February 13, 2020; date of current version June 22, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61972074 and Grant 61972075, in part by the China Postdoctoral Science Foundation under Grant 2018M640909, and in part by the National Postdoctoral Foundation for Innovative Talents under Grant BX201700046. Paper no. TII-20-0040. (Corresponding authors: Zhiwei Zhao; Zhengxin Yu.)

Weifeng Gao, Zhiwei Zhao, Minghang Yang, and Wenjie Huang are with the School of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu 610051, China (e-mail: weifeng@mobinets.org; zzw@uestc.edu.cn; minghang@mobinets.org; wenjie@mobinets.org).

Zhengxin Yu and Geyong Min are with the Department of Computer Science, University of Exeter, Exeter EX4 4QF, U.K. (e-mail: zy246@exeter.ac.uk; g.min@exeter.ac.uk).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TII.2020.2973754

to a sink node for further processing before a given deadline, in order to keep real-time monitoring of the targets [5]. For example, in smart building applications, the temperature in a building is supposed to be recorded, so the fire alarm can be triggered in time.

Due to the lossy nature of low-power wireless communications, it is nontrivial to guarantee that data are delivered to the sink node before a given deadline. Besides, various inter/intra network interference further leads to the difficulty for reliable packet transmissions. Multichannel communication is an effective way to improve communication reliability for low-power IoT nodes, where each node can operate on different channels or time according to the schedules. Interfering links are usually assigned different channels within the same slot to improve transmission efficiency.

To support deadline-driven data collection in IoT networks, the existing works consider the deadline constraint in the process of channel allocation [6]–[11]. Dao *et al.* [12] try to guarantee that the packets can be delivered before deadline with a specific probability and minimize energy consumption at the same time. Alinia *et al.* [13] focus on constructing an optimal tree with deadline constraint in wireless sensor networks with data aggregation in the time-division multiple-access (TDMA) manner, where time is divided into multiple time slots, and an IoT node is active at a few time slots and sleeps at the others. TDMA allows us to schedule the transmission of the nodes to avoid collisions, so the reliability and throughput can be improved. The data aggregation process is jointly considered in the scheduling, and the energy consumption can be further reduced compared to the works without data aggregation.

In the existing works, the deadline for a given path is met by assigning enough time slots before it [14]–[16]. For example, if the deadline is the fourth slot, a path with three hops will be assigned three slots before the fourth slot, such that the transmissions can arrive before the fourth slot.

*There are three limitations for the existing works.* First, the existing works often assume that the transmissions can always be successful within one slot, which is not true in real-world wireless networks. To ensure that the transmissions can be successful within one slot, one alternative is to increase the slot length to allow more retransmissions in one slot. However, the lengthened slot will cause extra delay for links with good qualities, as the slot lengths should be the same for all nodes for synchronization [17]. Second, we have to predict the link quality (e.g., packet delivery ratio—PDR) as a basic parameter before allocating channels. However, the link quality prediction

tasks can consume a large amount of energy and time for low-power wireless nodes, as they are based on energy-constraint hardware [18]–[20]. Third, when allocating slots/channels to the flow paths in a network, the impact of the allocation sequence for different paths is overlooked, which may reduce the utilization of wireless channels (as explained in Section II). Besides, in the existing works, many time slots are left unassigned after the channel allocation. The reason is that the fewer slots are used, the more energy can be saved. We observed that these unused slots could be potentially employed for retransmissions in the case of packet losses, so that the transmissions can meet the deadline with more chances.

To address the above limitations and improve the reliability before deadline, we propose an edge-based channel allocation (ECA) for unreliable IoT networks. In order to predict the PDR of the IoT nodes, we rely on the edge computing architecture so that the prediction overhead can be migrated from low-power nodes to powerful edge servers. Edge computing has been a promising solution to reduce the delay of IoT applications [21]–[25]. It provides services using edge servers that are close to users. With edge servers, we can achieve much lower response delay and save bandwidth. In this article, we adopt a recurrent neural network (RNN)-based method on the edge servers, which can accurately predict link quality, and the prediction will not consume energy on low-power IoT nodes.

For channel allocation, we first sort all paths to be assigned and prioritize the “bottleneck” paths with urgent deadlines and more collisions. After that, ECA starts the allocation process in a “two-round” manner. In the first round, the slot/channel pairs are assigned to links considering the link quality and the link dependence on the path. Specifically, slots/channels that maximize the packet delivery ratio before deadline (PDR-BD) are selected. In the second round, we further assign the unused slot/channel pairs for potential retransmissions. By assigning those slot/channel pairs to lossy links, the PDR-BD can be further improved.

We implement ECA with TinyOS/TelosB and conduct both simulation and testbed experiments. The evaluation results show that compared to the existing works, ECA greatly improves reliability in terms of the PDR-BD. The major contributions of this article are summarized as follows.

- 1) We propose a path prioritization scheme for channel allocation in IoT networks. Paths are prioritized based on link quality, deadline, and number of collisions, respectively, to improve channel utilization.
- 2) We propose an ECA for unreliable IoT networks, where the slot/channel pairs that maximize the PDR-BD are selected.
- 3) We propose a retransmission mechanism to further improve the PDR-BD, which efficiently utilizes spare slots to accommodate potential retransmissions from lossy links.

## II. MOTIVATION

In this section, we use two examples to illustrate the impact of wireless lossy nature on path assignment. Despite the limited

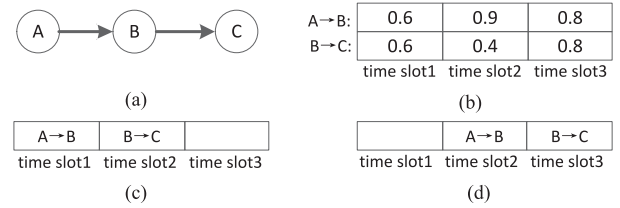


Fig. 1. (a)–(d) Example for unreliable wireless links.

number of channels and possible interference for IoT networks, we argue that the link quality, path assignment sequence, and retransmissions in TDMA channel allocation have a large impact on real-time performance.

### A. Impact of Unreliable Wireless Links

We first illustrate a simple example to show the impact of link quality in TDMA-based channel allocation [26], [27].

Fig. 1(a) shows two links, where node A has to send one packet to node C through node B. The deadline for the path and the length of a slotframe are three time slots. Assuming that there is only one available channel, the two links “A-B” and “B-C,” respectively, experience different link qualities in different time slots, which are illustrated as the decimals in the table in Fig. 1(b). In this article, we represent the link quality as the PDR of a wireless link.

Following the scheme in Wave [28] which intends to assign links within minimum time slots, as shown in Fig. 1(c), the links are assigned in the first two time slots. This assignment seems to guarantee the deadline of the transmission; however, it is obvious that the two link qualities are relatively low, which leads to poor PDR of 0.24 and a large delay if fail links are retransmitted in the following slotframes.

Furthermore, an intuitive solution to the above problem is to assign links with the highest link qualities, as shown in Fig. 1(d), which can increase the PDR-BD to 0.72. However, the transmission cannot be completely reliable due to the lossy nature of wireless links. Note that there is still a spare time slot that is not assigned to any link, so we exploit the spare time slots for retransmission to further improve the PDR. The detailed scheme is discussed in Section III-D.

### B. Impact of Path Assignment Order

The limited number of channels in IoT networks impedes efficient and reliable data transmission. There are only 16 channels available in IEEE 802.15.4, out of which most overlap with three commonly used WiFi channels, causing external interference to most of the IEEE 802.15.4 channels. To be precise, there are only four channels that do not overlap with WiFi and a few channels that partially overlap. Another reason for the limitation is inefficient channel assignment priority, which is illustrated in Figs. 2 and 3.

There are three paths in the example, where nodes A, C, and F have to send one packet to nodes B, E, and H, respectively. The directed dotted lines represent interfering links. The deadline for the paths and the number of available channels are both two.

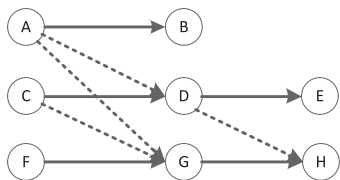


Fig. 2. Example for assignment priority.

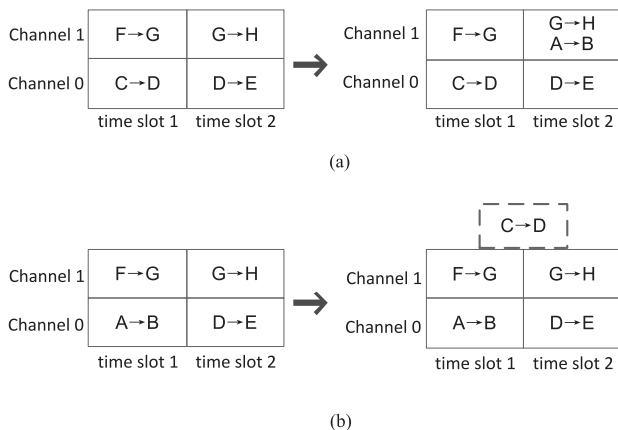


Fig. 3. Assignment with different priorities. (a) Successful allocation without conflict: Links G-H and A-B can be assigned in the same time slot and channel. (b) Failed allocation with one unavailable link: Link C-D cannot be assigned in any time slot or channel.

Fig. 3 shows the assignment process with different assignment orders. Fig. 3(a) shows a successful allocation. We first allocate links  $F-G$  and  $G-H$  on channel 1 and then  $C-D$  and  $D-E$  on channel 0, as shown in the left subfigure of Fig. 3(a). In the right subfigure of Fig. 3(a), the left link  $A-B$  can be assigned the same channel as  $G-H$  because they do not interfere with each other. In this case, there is no allocation conflict. However, Fig. 3(b) depicts a failed allocation. We first allocate  $A-B$  on channel 0, as shown in the left subfigure of Fig. 3(b). When we want to assign  $C-D$  and  $D-E$ , there is no available channel or time slot for  $C-D$  because all the assigned links interfere with  $C-D$ . In this case, the allocation fails. Fig. 3 implies the requirement of carefully considering the allocation sequence; otherwise, the resources cannot be efficiently used.

### C. Impact of Retransmissions

For the simplicity of protocol design, the existing works [28]–[31] often assume that a packet can be always delivered in one slot, which is not true for real-world networks. One alternative is to increase the slot length to accommodate retransmissions within one slot. However, as the slot length for every slot is the same in order to synchronize the channel switching, the lengthened slots lead to extra delay for links with few packet losses. Some works assign retransmissions in the next transmission round, inducing at least one full-cycled delay.

Based on the above analysis of existing works, we identify an optimization opportunity as follows. With the existing works, on the one hand, potential retransmissions cannot be scheduled in advance as they are unpredictable; on the other hand, there are

some slots left unused after the channel schedule. Our key idea is to exploit those unused spare slots for possible retransmissions in a probable manner. From the above illustrative examples, we can see the following.

- 1) The lossy nature of wireless links should be considered in the channel assignment. The potential packet losses will delay the packet delivery if retransmission happens and can bring potential security risks when packets are lost, which cannot be tolerant for most applications.
- 2) The assigning sequence should be carefully designed in channel assignment. Note that transmission on a certain link can conflict with links in its interference area if they utilize the same channel. The inefficient assigning priority decision can cause unnecessary conflicts, further leading to a lack of available channels or time slots.

To address the above problems, we consider three factors related to the path assignment order; a promising metric is designed exploiting these factors, in which the length of a path, the collision situation, and the number of generated packets are considered. Then, we propose deadline-driven channel allocation, which intends to choose time slots/channels with high link qualities for the links. Besides, in order to further improve the reliability of data collection, appropriate retransmission schemes are studied after the channel allocation, which is expected to greatly increase the PDR-BD for the source nodes. The detailed design is presented in Section III.

## III. MAIN DESIGN OF ECA

In this section, we present the main design of ECA, combining both channel diversity and retransmission opportunities.

### A. Overview

Due to the limited processing ability of the low-power wireless nodes, link predictions are hard to be implemented. We employ edge computing servers to perform such complex tasks. The results are then returned to the low-power network nodes. We propose an edge-based framework for channel allocation. All the computational tasks are executed on the edge servers, such that 1) little additional overhead is incurred at the low-power IoT nodes and 2) the allocation results can be obtained in real time. As a result, the allocation can be run frequently to deal with the time-varying channels. The data measurement on edge servers are described in Section III-B.

The ECA approach mainly consists of two parts. In the first stage, we propose the path-based assignment, in which we successively schedule the links based on a metric of their paths, as detailed in Section III-C2. The path metric is obtained according to the urgent deadlines and collisions, which are both related to the residual available slots/channels. Since each link experiences varying link quality on different slots and channels, a link involved in any path is assigned high link qualities, as described in Section III-C3. The second stage happens after the links are assigned. To further improve the PDR, we adopt a novel retransmission scheme to reuse the unassigned time slots and channels, as described in Section III-D. The retransmission scheme sorts all the links and sequentially picks the idle



TABLE I  
NOTATIONS USED IN THIS ARTICLE

Symbols	Notations
$u_i$	Available time slots before deadline of path $i$
$D_i$	Deadline of path $i$
$L_i$	Path length of path $i$
$c_i$	The number of conflicting links for path $i$
$p_{t,ch}^l$	Link quality of $l$ at time slot $t$ and channel $ch$
P, T, CH	The set of path, time slots and channels
As	Links that are assigned already
$b_{t,ch}^l$	Whether link $l$ is assigned to time slot $t$ and channel $ch$
$q_b$	Link quality before retransmission
$q_n$	Link quality after retransmission
$pr_l$	Quality improvement for link $l$
$pr_g$	Quality improvement for link set $g$

slots/channels for the links retransmission. During the channel allocation process, link quality is estimated using the in-packet corruptions [32], and the 4b link estimation [33] is used to measure the link quality for each pair of channels and slots. The notations used throughout this article are summarized in Table I.

## B. Data Measurement on Edge Servers

In order to run the ECA algorithms, the edge servers need to measure the necessary link information as input: the set of links, the number of available slots, and the link quality for each link with the corresponding channel and slot. As we are targeting at the IoT networks with fixed deployment, the links and slots can be directly collected from the IoT networks. The key problem is to obtain the time-varying link quality with given channels and slots. Fortunately, due to the use of edge servers, we can employ a link prediction method based on the RNN by utilizing the historical snapshots of the IoT network [34]. Typically, the link quality is measured first based on the well-known 4-bit method [33]. An RNN is trained online with the Kalman filter and real-time particle swarm optimization to predict the channel condition. This RNN-based method can accurately predict the link qualities and can even learn temporal statistics from previous link conditions. Based on Chris Potter's experiments, the RNN-based method can achieve the prediction mean square error of  $10^{-3}$ .

Note that some research studies utilize temporal link quality prediction [35]–[37] to capture the immediate link condition and improve throughput. Although the RNN-based method can also learn the temporal link statistics, we do not need to adapt to the immediate link qualities, and we only estimate the overall link quality within a period (time slot), so that this time slot can be allocated according to this overall link quality, in which the impact of temporal link quality is included.

## C. Path-Based Channel Assignment

1) *Extending the Link Schedule to Path Schedule*: To assign the transmissions of data collection, links are ordered by the total traffic traveling through them [28] and sequentially assigned from the earliest time slot, and the collection delay is minimized.

Since they do not consider the lossy nature of wireless links and we are going to take the link quality into account, it is desired to choose good-quality slots/channels for links. Note that each link has its preference of time slots and channels according to its quality, choosing the best quality slot/channel for a link may lead to two unacceptable consequences. First, the preference of time slots of different links in a path may conflict with their transmission order, e.g., link 1 prefers time slot 2, while its subsequent link 2 prefers time slot 1. Second, once a link chooses the time slot and channel, the remaining slots may be insufficient for the packet arriving at the sink node before the given deadline. Given the weakness above, we extend the link-based prioritizing and channel assignment to the path based one.

2) *Path Prioritization*: Given the available channels, ECA intends to assign high-quality time slots and channels to the links, to maximize the packet received ratio before deadline of the data collection. ECA assigns time slots and channels for links among a path in the network; the algorithm iteratively processes one path at a time until all paths are assigned.

After assigning a path, the remaining paths will have fewer available time slots and channels. Long paths consume more slots than short paths, so the assigning sequence fundamentally influences the performance of channel allocation. In this case, the number of residual time slots and channels for a path is important in channel allocation. On the one hand, the less residual slots and channels are available for a path, the earlier it should be allocated due to its urgent deadline. On the other hand, the residual time and channels are required to be jointly considered. For example, a path with plenty of time slots before its deadline cannot be assigned if its links conflict with all the assigned links. On this occasion, we propose a path prioritizing metric to order the paths. The urgent deadline and collisions are considered to avoid that too many paths cannot be assigned due to the deadline and interference constraints.

At first, since every IoT node should send its data to sink node before a given deadline, the urgency of a path can be defined as  $u_i = d_i - l_i$ , where  $d_i$  denotes the deadline of path  $i$  and  $l_i$  denotes the length of path  $i$ . So, paths with earlier deadline and containing more links represent that they are more urgent and have fewer available time slots for transmission. Thus, the urgent paths are supposed to be allocated early in case that time slots are occupied by those not-urgent paths.

In addition to the urgent deadline constraint, the conflict relationship between links also has a significant impact on the path assignment order. There are multiple available channels on a time slot, while a link may not be allowed to use a channel or time slot if it conflicts with the assigned links, so the available channels and time slots are limited. In this article, we take the conflict relationship into account for prioritizing metric. Note that links involved in a network experience different conflict relationships due to the different network topologies and locations of the nodes, so links that conflict with more other links have less available channels and time slots to be assigned and should be allocated earlier.

However, the impact of urgency and conflicts are intertwined because an unavailable time slot may be caused by either exceeding time slots or conflict channels. Besides, the number of total available time slots and channels for a path depends on the

allocation of those assigned paths, which is decided by the path prioritizing. So, it is nontrivial to design an optimal prioritizing metric, which precisely utilizes the available slots and channels.

In this case, we propose the prioritizing metric based on the urgent deadline and conflicts. The metric contains two components:  $u_i$ , denoting the urgency of a path, and the number of conflict links that represents the conflict relationship of a path. Note that the conflict situation of a link in different slots/channels is different, so that most paths can be allocated without conflict; the priority of a path is constrained by the worst link with the most conflict links, which is represented as  $c_i$ .

We use the conflicting link set to demonstrate the conflict relationship of a path; the number of conflict links with the worst case in path  $i$  is calculated as  $c_i = \max C$ , where  $C$  is the set that contains the number of conflict links of every link in path  $i$ . Note that as the paths are allocated, the unavailable slots and channels increase significantly, which is because links that are conflict with the assigned paths cannot transmit data on those slots and channels. The paths with larger  $c_i$  mean higher probability to conflict with other links, so these paths are supposed to be served early. In conclusion, we propose the path prioritizing metric as follows:

$$m_i = \alpha u_i + (1 - \alpha)c_i \quad (1)$$

where  $\alpha$  is the weight factor that determines the priority of data urgency and conflict conditions. For example,  $\alpha$  larger than 0.5 means that the urgency of a path is considered more important than its interference situation. Since the conflict relationship between paths can be significantly different with different network topologies and wireless environment, the parameter  $\alpha$  can be adjusted according to the practical scenario and requirements. Using the above metric, the longer paths with more collisions are first allocated.

**3) Link-Quality-Aware Channel Allocation:** With the path priority, we assign time slots and channels for links in different paths. Since the quality of a link varies among time and channels, the allocation is based on the average link quality within a time slot. In this case, the link quality of a link can be dramatically different among time slots and channels. Besides, the links in a path experience different link qualities, leading to the complexity of channel allocation for a path. The target of the intrapath allocation is to maximize the PDR-BD, which is the multiplication of all the link qualities. However, the preferred time slots or channels of links may break the dependence of the links in a path, which increases the complexity of the channel allocation. With the constraint of path assignment sequence and link conflicts, the problem of allocation is described as follows:

$$\begin{aligned} \max \quad & \prod_{b_{i,ch}^l=1} p_{t,ch}^l, (l \in P_i, t \in T, ch \in C) \\ \text{s.t.} \quad & \forall l \in P_i, t_l < t_{l+1} \\ & \forall l \in P_i, 0 < t_l < d \\ & \forall l \in P_i, \forall m \in A_s, b_{t,ch}^l \neq 1, \text{ if } \text{hear}(\text{recv}(m), l) \\ & \forall l \in P_i, \forall m \in A_s, b_t^l \neq 1, \text{ if } \text{adjacent}(m, l) \end{aligned} \quad (2)$$

---

**ALGORITHM 1:** Link Scheduling Within A Path.

---

**Input:** The set of links in the path,  $L$   
The number of links,  $l$  The number of available time slots,  $n$   
The link quality of link  $s$  in time slot  $t$  and channel  $ch$ ,  $q_{t,ch}^s$   
**Output:** Link scheduling,  $b_{t,ch}$   
**for each**  $s \in L$  **do**  
    | Judge( $s$ )  
**end**  
**Function** Judge( $s$ )  
**if**  $t = \text{arcm}ax(q_{t,ch}^s) \in (t_{min}, t_{min} + \frac{n}{l})$  **then**  
    |  $b_{t,ch}^s = 1$   
    |  $n = n - (t - t_{min})$   
    |  $l = l - 1$   
**end**  
**else**  
    | Remove( $q_{t,ch}^s$ )  
    | Judge( $s$ )  
**end**

---

where  $p_{t,ch}^l$  is the link quality of link  $l$  at time slot  $t$  and channel  $ch$ ,  $P_i$  denotes the path  $i$ ,  $t_l$  represents the time slot for link  $l$ ,  $d$  is the data deadline, and  $b_{t,ch}^l$  indicates whether a link  $l$  transmits data on time slot  $t$  and channel  $ch$ . The first two constraints mean that a link should transmit data before its following links and deadline; the other two reflect the effect of interference and adjacent links, in which two links cannot transmit concurrently in the same channel or time slot.

The above problem can be reduced to the task scheduling with the dependence constraint problem which is NP-hard [38]. On this occasion, a heuristic algorithm is proposed for the allocation of links, as shown in Algorithm 1. Specifically, the total available time slots are divided into  $L$  pieces so that all the links have the same period  $\frac{N}{L}$  for allocation. We then consider every link  $s$  within the path. If the time slot of the best link quality lies in its own period, we allocate this time slot and corresponding channel to the link. Otherwise, we try to find the second-best quality time slot and judge whether this time slot is in its period. By iterating the above processing (Function Judge( $s$ ) in Algorithm 1), we can finally allocate the time slots and channels for all the links within a path.

Due to the varying quality of links, the packets generated by source nodes cannot always reach the destination node; in ECA, we apply the retransmitting scheme to improve the reliability of transmission.

### D. Channel Allocation for the Retransmission Scheme

Due to the varying link quality, the packets cannot always reach the sink node; to further improve the PDR-BD, we utilize unused time slots and channels for retransmissions.

Optimally, it is expected that the PDR-BD can be maximized with all the retransmissions. However, it is extremely complex because the improvement on each time slot and channel for links can be different, and it is much more difficult when the PDR-BD means multiplying the link qualities of several links. In this article, we propose a retransmission scheme, which considers both the improvement of link quality and the available time slots and channels of a link for retransmission.

Our retransmission scheme works in two steps. First, for each time slot and channel, determine the links that are available

for transmission. If a link meets the sequence and conflicting requirements of the assigned links before, it means that this link can be assigned for retransmission here, and it is added to the retransmission link set of this slot/channel. In the second step, the retransmission link set above is divided into conflict-free sorted subsets; the best subset is then chosen for retransmission assignment. The sorting process is based on two factors, i.e., the quality profit and retransmission chances, which are the number of available retransmission slots/channels. For an available link in a retransmission set, the quality profit of retransmissions on a specific time slot and channel can be obtained as follows:

$$pr_i = 1 - (1 - q_b)(1 - q_n) - q_b \quad (3)$$

where  $q_b$  represents the succeeding probability without this retransmission and  $q_n$  represents the succeeding probability with retransmission on this slot/channel. The quality profit represents the improvement of link quality by retransmitting the packet at this time slot and channel. The links with higher profit are desired to be more important for retransmission because they can improve the reliability significantly.  $q_b$  reflects the probability of successful transmission with the help of all the previous retransmissions; once a link is assigned retransmission,  $q_b$  will turn to  $q_n$ , and the profit of next retransmission will decrease so that a retransmitted link has less chance to retransmission on other time slots. The quality profit represents the improvement of the overall link quality with the existence of retransmission.

Note that a time slot/channel pair can be assigned to multiple links retransmissions concurrently if they do not interfere with each other, so the quality profit is extended to a set of links. Specifically, the quality profit of a set of links at a time slot and channel can be expressed as the summation of all the links quality profit

$$pr_g = \sum_{i=1}^G pr_i \quad (4)$$

where  $G$  represents the set of links, which can be assigned retransmissions concurrently with the same channel. This metric aims at assigning a set of links on an idle time slot and channel so that the overall link quality can be improved.

Furthermore, when assigning time slots and channels for retransmission, the retransmission chances differ from each other due to collisions and link dependence among the paths. Links with fewer retransmission chances mean that they need this time slot for retransmission with more expectation and have higher priorities to be retransmitted. Above these two aspects, we propose a specific metric to prioritize the links waiting to be retransmitted as follows:

$$mr_g = \beta pr_g + (1 - \beta) \sum_{i=1}^g r_i \quad (5)$$

where  $r_i$  is the available time slots to retransmit link  $i$ . With this retransmission metric, link sets with better quality profit and less available slots/channels should be first served.

**1) Discussion on the Additional Overhead of Retransmission:** As mentioned before, pushing the retransmissions to the next duty cycle leads to unexpected latency, and the retransmissions can still fail and thus break the deadline

constraint. In this article, however, we try to retransmit the lost packets within the same duty cycle so that the deadline constraint can be guaranteed. Besides, the retransmissions are also carefully scheduled to reduce the failure probability of them. The extra latency and energy overhead occur only if packet loss happens, so the allocated slots for retransmissions may be idle, and the overhead is related to the PDR of the links. ECA allocates high-quality retransmission slots to links with low PDR so that the retransmission overhead can be reduced.

The assignment may be violated when there are too many packet losses. In this case, the available time slots and channels may not be enough for satisfying retransmission. As a result, many transmissions would fail, and the deadline constraint could not be guaranteed.

**2) Algorithm Complexity Analysis:** The time complexity of the proposed algorithm that allocates channel and time slots for a wireless network is  $\mathcal{O}(L * (C * T) \log(C * T))$ , where  $L$  is the number of links in the network, and  $C$  and  $T$  are the numbers of channels and time slots, so the overhead of this algorithm is dependent on the scale of the networks. However, both the link quality prediction and allocation algorithm are run on the edge servers, which do not have power constraint and are considered as powerful computation nodes compared with low-power sensors, so the overhead for prediction and channel allocation does not impact the lifetime of IoT nodes.

## E. Distributed ECA

The operation of ECA requires global information of the whole network, for example, computing the conflict situation has to consider all the other paths. However, IoT networks applications are usually connected with dynamic network deployment and wireless environment, and nodes cannot obtain the varying link qualities in advance. So, the centralized scheme may not be appropriate in practice; in order to adapt to the dynamics of application scenarios and to deal with the difficulties of collecting global information, we extend the ECA into a distributed one, as shown in Algorithm 2, which requires the information from only one-hop neighbors.

Specifically, if a node  $i$  has a packet to send, it first determines its preferred channel (i.e., the channel with highest link quality, and is found by Function *Sort(channel, quality)* in Algorithm 2) and compares its own priority (Function *Prioritize()*) with the priorities of the neighbors. Communicating with neighbors (Functions *Notice()* and *Receive()*), if its priority is the largest, it transmits the packet immediately. Otherwise, it will expect to transmit on another channel according to its priority (Function *Del()*) and compare new priorities with the new channel, which is shown as Function *Decide()* in Algorithm 2. If all the channels are unavailable, it suppresses the transmission to the next time slot. Besides, if transmission failure happens, the node will update its priority with consideration of the retransmission cost.

Since distributed ECA requires nodes estimating the link quality locally, nodes will consume more energy for estimation, and the accuracy will not as high as the RNN method. The advantage of distributed ECA (i.e., adaptive to dynamic wireless networks) can be improved with a more powerful link quality prediction method in the future. In Section IV, we first



**ALGORITHM 2:** Distributed Allocation.

```

while ready(i) do
  Sort(ch, p); //Finding the best channel
  pri = Prioritize(ch); //Calculating its priority
  Notice(Ready, ch[0], pri); //Notifying its neighbors
  Receive();
  Decide();
end
Function Decide()
//Comparing with priorities of neighbors
for each j ∈ Ni do
  if chj == chi then
    C.append(j);
  end
end
if pri > max(prj ∈ C) then
  Send();
  if Fail then
    Update(pri); //If failed, updating its priority
  end
end
else
  Del(c, c[0]); //Validating the second best channel
  Decide();
end

```

evaluate the performance of distributed ECA with simulation by assuming the knowledge of link qualities.

#### IV. PERFORMANCE EVALUATION AND ANALYSIS

In this article, the performance of ECA is evaluated by both simulations with TOSSIM [39] and real testbed experiments containing 20 TelosB motes, in terms of the following metrics.

- 1) *PDR-BD*: It represents the probability of packets successfully arriving at the sink node and can imply the effectiveness of data transmission. The PDR-BD can be obtained by gathering the link quality among the transmission path.
- 2) *Resource utilization*: The utilization is calculated as the proportion of slots/channels that are actually assigned in the channel allocation scheme. It reflects the influence of the retransmission assignment. The resource utilization metric does not show the energy consumption state, because the retransmission slots would not be active if the last transmission succeeds.
- 3) *The number of insufficient paths*: Since colliding links cannot be assigned to the same channel and slot, in some cases, there may not be enough slots to be assigned for the paths. With an appropriate assignment, the insufficient paths can be decreased.
- 4) *The number of retransmissions*: When packet losses happen, retransmissions are required. Reasonable retransmissions can significantly improve transmission reliability. If more retransmissions happen, the advantage of ECA to the existing works is larger.

We compare the performance of ECA with Wave [40]. Wave is the latest work on channel allocation for low-power wireless networks. It adopts a simple and efficient distributed channel allocation algorithm. Compared with Wave, ECA further considers path assignment priority and retransmissions.

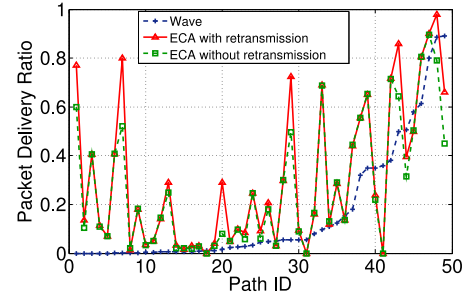


Fig. 4. Packet delivery ratio before deadline in simulation.

#### A. Simulation Evaluation

In this section, we first evaluate the performance of distributed ECA with TOSSIM. The network contains 50 nodes, which are randomly deployed in an area. One node acts as the sink node, and all other 49 nodes are source nodes; source nodes also perform to relay packets for other nodes. The simulation assumes that nodes are aware of the local link qualities, so they can make allocation decisions on their own.

The source nodes generate random numbers (for simplicity, we assume that this number is less than three) of packets every duty cycle. The size of a time slot is set to be 100 ms, and there are up to 200 time slots in a duty cycle, which is also used to reflect the data deadline. The packets generated from source nodes traverse to the sink node through a specific path, which is built based on the least hop count. The evaluation and results analysis is demonstrated in the following subsection. The parameter  $\alpha$  that determines the weight of urgency and path length is set to 0.5.

Fig. 4 shows the PDR-BD among ECA and Wave. The different PDR among paths is due to the different path lengths and packet loss rates. It can be seen that the PDR-BD of ECA without retransmission is larger than that of Wave. The reason is that ECA intends to select slots/channels with better link qualities, while the link qualities in Wave are often worse, leading to spare slots and channels. In this case, ECA with retransmission performs better than that without retransmission. It can be inferred that by using the spare slots, the reliability of data collection can be significantly improved. There are a few paths that Wave performs better than ECA; this is because ECA assigns a few links on worse slots/channels for the purpose of overall PDR-BD improvement reflected in path priority and retransmission assignment.

Fig. 5 compares the resource utilization of ECA and Wave. The utilization is calculated as the proportion of the assigned slots/channels over the total number of slots/channels. Since the assigned slots/channels include slots for retransmission, a larger utilization does not mean more energy consumption; the retransmission slot/channel would be active only if transmissions failed. The figure depicts that when there are two available channels, the utilization of ECA is lower than Wave. With limited channels, ECA chooses high-quality slots and channels. Some links cannot be assigned, leading to more conflicts than Wave. While as the available channels increase, ECA performs better than Wave. This is because extra retransmissions are allowed in

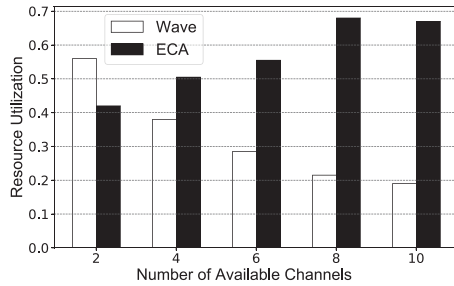


Fig. 5. Comparison of resource utilization.

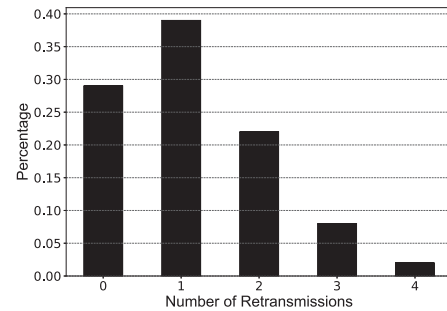


Fig. 8. Number of retransmissions.

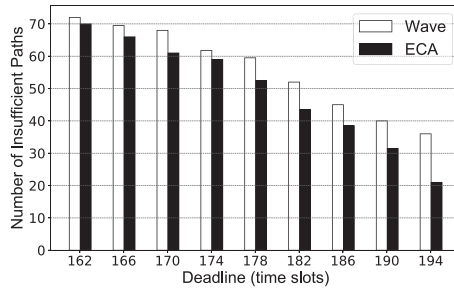


Fig. 6. Insufficient paths with fixed channels.



Fig. 9. Testbed for performance evaluation.

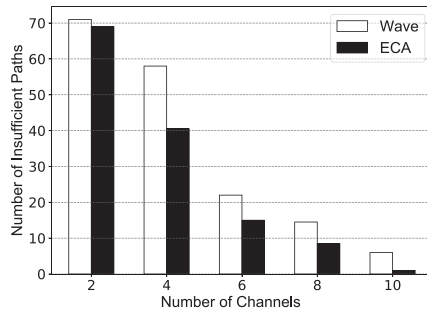


Fig. 7. Insufficient paths with fixed deadline.

ECA, while in Wave, a full-cycled delay will be incurred when packets fail.

Figs. 6 and 7 compare the number of insufficient paths between ECA and Wave [41]; Wave aims at minimizing completion delay, while not considering the insufficient cases before deadline. Since some source nodes generate more than one packet, their path will repeat and be assigned for multiple times. We account the number of paths, which cannot be sufficiently assigned before the deadline as the number of available channels increases. Fig. 6 uses a fixed number of channels; as the deadline postponed, the number of insufficient paths decreases, and ECA always has less insufficient paths than Wave. Fig. 7 fixes the deadline, while accounting the insufficient paths with the number of channels increases. The proposed ECA can always perform better than Wave whether increasing the number of channels or postponing the deadline; furthermore, as the deadline or number of channels increases, the advantage of ECA also increases; this is because ECA prioritizes the paths considering colliding paths and schedules these paths when the available slots are sufficient.

The retransmission statistics is demonstrated in Fig. 8. The cases without a retransmission scheme are not considered in this figure. Among all the bars, 29% links do not need the retransmission and can succeed for once, and about 39% links use one retransmission. There are about 22% links having two retransmissions, and retransmissions happen three times on 8% links. At last, 2% links are using four retransmissions. We can get that most links are lossy and retransmissions are necessarily required, and for these cases with retransmissions, the PDR-BD is greatly improved.

### B. Performance Evaluation With Testbed

We evaluate the performance of centralized ECA with a local testbed. We use 20 TelosB motes with node ID from 1 to 20. One mote is attached to a PC to perform as the edge server. It overhears the network statistics and performs 1) RNN-based channel estimation and 2) allocation algorithm. Fig. 9 depicts the nodes that are installed on the ceiling of our office. The algorithm is burnt on these motes concurrently through USB cables. Source nodes generate 100 packets and transmit them to node #20 through a shortest-path tree. Three channels (channels 11, 18, and 26) are used for data transmission.

This article use TelosB motes as the low-power wireless devices, which is an open-source domain that has been made to enable the state-of-the-art experimentation for the research community. On the other hand, the trend of using other low-power end devices such as computational RFID (CRFID) tags for low-power wireless communication is rising. These wireless tags can achieve a much lower energy consumption and have a shorter communication range. Typically, networks such as body sensor networks composed of the RFID tags also have the transmission



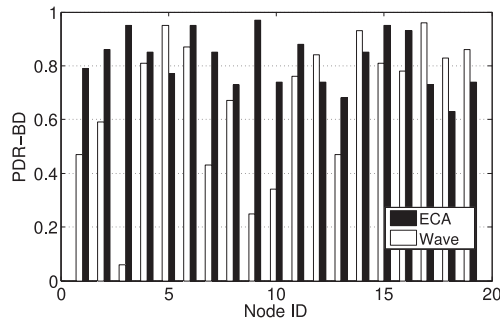


Fig. 10. Packet delivery ratio before deadline in testbed.

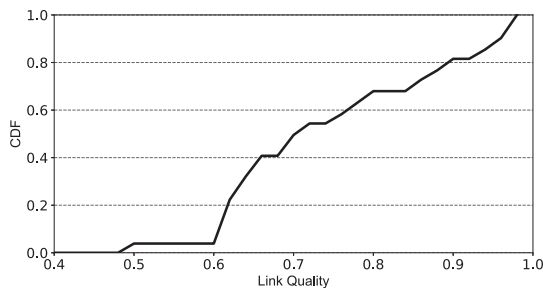


Fig. 11. Link qualities in the testbed.

deadline problem, especially for health monitoring applications, so the basic idea of ECA still works for CRFID networks. The difference would appear that there are fewer available channels for RFID networks, so the allocation was more strict. Besides, both time synchronization and link quality prediction may be more difficult because of the constraint capability of low-power RFID tags.

We illustrate different link qualities among these nodes in Fig. 11, where most links (80%) experience good link qualities larger than 0.6. Fig. 10 shows the PDR-BD of the distributed ECA and Wave. It is obvious that the PDR-BD of ECA is larger than that of Wave in most source nodes, which are all above 0.6. The average PDR-BD in ECA is improved around 22% compared to that in Wave. The reason is that the proposed ECA considers the varying link quality and tends to choose higher quality for transmission links. Furthermore, ECA takes advantage of retransmitting the failing links, which leads to a larger PDR-BD compared with Wave that ignores the chance of high link quality and retransmission. For some nodes, the PDR-BD of Wave could be larger than ECA, because the two algorithms adopt different nodes assigning orders. To guarantee the transmissions of nodes with higher priority, ECA might delay the transmissions of a few nodes that are not urgent. Such a sacrifice is acceptable by considering the increase in global performance.

## V. CONCLUSION

In this article, we proposed a novel channel assignment scheme for deadline-driven IoT networks, called ECA, which maximized the PDR with lossy links in IoT networks, while guaranteeing a given deadline. A path-based assignment scheme

was presented to adapt to the situation of deadline assignment, and a novel prioritizing metric was proposed, which jointly considered link quality, path length, and possible collisions. Moreover, we developed a retransmission scheme to make efficient use of the unassigned idle slots/channels, which further improved the reliability without incurring extra energy overhead. Extensive simulation and testbed experiments demonstrated that ECA significantly improves the protocol performance.

## REFERENCES

- [1] S. Feng, P. Setoodeh, and S. Haykin, "Smart home: Cognitive interactive people-centric Internet of Things," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 34–39, Feb. 2017.
- [2] O. Guerra-Santin, S. Silvester, and N. R. Herrera, "Building monitoring to determine occupancy patterns in renovation projects," in *Proc. Conf. Sustain. Built Environ.: Transition Zero*, 2016, pp. 65–74.
- [3] Y. Gao *et al.*, "Mosaic: A low-cost mobile sensing system for urban air quality monitoring," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.
- [4] A. Fabre *et al.*, "Deploying a 6LoWPAN, CoAP, low power, wireless sensor network: Poster abstract," in *Proc 14th ACM Conf. Embedded Netw. Sens. Syst.*, 2016, pp. 362–363.
- [5] E. Fadel *et al.*, "A survey on wireless sensor networks for smart grid," *Comput. Commun.*, vol. 71, pp. 22–33, 2015.
- [6] M. Bagaa, M. Younis, D. Djenouri, A. Derhab, and N. Badache, "Distributed low-latency data aggregation scheduling in wireless sensor networks," *ACM Trans. Sens. Netw.*, vol. 11, 2015, Art. no. 49.
- [7] M. Bagaa, M. Younis, and I. Balasingham, "Optimal strategies for data aggregation scheduling in wireless sensor networks," in *IEEE Global Commun. Conf.*, 2015, pp. 1–6.
- [8] M. Dong, K. Ota, A. Liu, and M. Guo, "Joint optimization of lifetime and transport delay under reliability constraint wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 225–236, Jan. 2016.
- [9] R. Souza, P. Minet, and E. Livolant, "DiSCA: A distributed scheduling for convergecast in multichannel wireless sensor networks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage.*, 2015, pp. 156–164.
- [10] D. De Guglielmo, G. Anastasi, and A. Seghetti, "From IEEE 802.15.4 to IEEE 802.15.4e: A step towards the Internet of Things," in *Advances onto the Internet of Things*. New York, NY, USA: Springer, 2014, pp. 135–152.
- [11] Y.-D. Lee and W.-Y. Chung, "Wireless sensor network based wearable smart shirt for ubiquitous health and activity monitoring," *Sens. Actuators B, Chem.*, vol. 140, no. 2, pp. 390–395, 2009.
- [12] T.-N. Dao, S. Yoon, and J. Kim, "A deadline-aware scheduling and forwarding scheme in wireless sensor networks," *Sensors*, vol. 16, no. 1, 2016, Art. no. 59.
- [13] B. Alinia, M. H. Hajiesmaili, and A. Khonsari, "On the construction of maximum-quality aggregation trees in deadline-constrained WSNs," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 226–234.
- [14] S. C. Ergen and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Netw.*, vol. 16, no. 4, pp. 985–997, 2010.
- [15] J. Ma, W. Lou, Y. Wu, X.-Y. Li, and G. Chen, "Energy efficient TDMA sleep scheduling in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2009, pp. 630–638.
- [16] D. Karaboga, S. Okdem, and C. Ozturk, "Cluster based wireless sensor network routing using artificial bee colony algorithm," *Wireless Netw.*, vol. 18, no. 7, pp. 847–860, 2012.
- [17] A. Mihnea and M. Cardei, "Multi-channel wireless sensor networks," in *Recent Development in Wireless Sensor and Ad-hoc Networks*. New York, NY, USA: Springer, 2015.
- [18] Z. Zhao, W. Dong, J. Bu, T. Gu, and G. Min, "Accurate and generic sender selection for bulk data dissemination in low-power wireless networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 948–959, Apr. 2017.
- [19] B. Silva, R. M. Fisher, A. Kumar, and G. P. Hancke, "Experimental link quality characterization of wireless sensor networks for underground monitoring," *IEEE Trans. Ind. Informat.*, vol. 11, no. 5, pp. 1099–1110, Oct. 2015.
- [20] T. Liu and A. E. Cerpa, "Data-driven link quality prediction using link features," *ACM Trans. Sens. Netw.*, vol. 10, 2014, Art. no. 37.
- [21] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
- [22] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2017.

- [23] Z. Wang, Z. Zhao, G. Min, X. Huang, Q. Ni, and R. Wang, "User mobility aware task assignment for mobile edge computing," *Future Gener. Comput. Syst.*, vol. 85, pp. 1–8, 2018.
- [24] Z. Zhao, G. Min, W. Gao, Y. Wu, H. Duan, and Q. Ni, "Deploying edge computing nodes for large-scale IoT: A diversity aware approach," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3606–3614, Oct. 2018.
- [25] C. Shu, Z. Zhao, Y. Han, G. Min, and H. Duan, "Multi-user offloading for edge computing networks: A dependency-aware and latency-optimal approach," *IEEE Internet Things J.*, to be published, doi: [10.1109/JIOT.2019.2943373](https://doi.org/10.1109/JIOT.2019.2943373).
- [26] P. Tuset-Peiro, F. Vazquez-Gallego, J. Alonso-Zarate, L. Alonso, and X. Vilajosana, "LPDQ: A self-scheduled TDMA MAC protocol for one-hop dynamic low-power wireless networks," *Pervasive Mobile Comput.*, vol. 20, pp. 84–99, 2015.
- [27] W. Gao, Z. Zhao, G. Min, Y. Cao, H. Duan, L. Liu, Y. Long, and G. Yin, "Link quality aware channel allocation for multichannel body sensor networks," *Pervasive Mobile Comput.*, vol. 41, pp. 354–364, 2017.
- [28] R. Soua, P. Minet, and E. Livolant, "Wave: A distributed scheduling algorithm for convergecast in IEEE 802.15.4e TSCH networks," *Trans. Emerg. Telecommun. Technol.*, vol. 27, no. 4, pp. 557–575, 2015.
- [29] N. Accettura, M. R. Palattella, G. Boggia, and L. A. Grieco, "Decentralized traffic aware scheduling for multi-hop low power lossy networks in the internet of things," in *Proc. World Wireless, Mobile Multimedia Netw.*, 2013, pp. 1–6.
- [30] C. Liu and J. Wu, "Routing in a cyclic mobispace," in *Proc. 9th ACM Int. Symp. Mobile Netw. Comput.*, 2008, pp. 351–360.
- [31] T. Qiu, Y. Lv, F. Xia, N. Chen, J. Wan, and A. Tolba, "ERGID: An efficient routing protocol for emergency response Internet of Things," *J. Netw. Comput. Appl.*, vol. 72, pp. 104–112, 2016.
- [32] U. Ashraf, S. Abdellatif, and G. Juano, "An interference and link-quality aware routing metric for wireless mesh networks," in *Proc. IEEE 68th Veh. Technol. Conf.*, 2008, pp. 1–5.
- [33] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Four-bit wireless link estimation," in *Proc. 6th Workshop Hot Topics Netw.*, 2007.
- [34] C. Potter, "RNN based MIMO channel prediction," in *Differential Evolution in Electromagnetics*. New York, NY, USA: Springer, 2010, pp. 177–206.
- [35] T. Liu and A. E. Cerpa, "Temporal adaptive link quality prediction with online learning," *ACM Trans. Sens. Netw.*, vol. 10, no. 3, 2014, Art. no. 46.
- [36] Z. Zhao, W. Dong, G. Chen, G. Min, T. Gu, and J. Bu, "Embracing corruption burstiness: Fast error recovery for zigbee under Wi-Fi interference," *IEEE Trans. Mobile Comput.*, vol. 16, no. 9, pp. 2518–2530, Sep. 2017.
- [37] K. Lei, M. Qin, B. Bai, G. Zhang, and M. Yang, "GCN-GAN: A non-linear temporal link prediction model for weighted dynamic networks," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 388–396.
- [38] H. Chetto, M. Silly, and T. Bouchentouf, "Dynamic scheduling of real-time tasks under precedence constraints," *Real-Time Syst.*, vol. 2, no. 3, pp. 181–194, 1990.
- [39] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in *Proc. 1st Int. Conf. Embedded Netw. Sens. Syst.*, 2003, pp. 126–137.
- [40] R. Soua, P. Minet, and E. Livolant, "A distributed joint channel and slot assignment for convergecast in wireless sensor networks," in *Proc. Int. Conf. New Technol. Mobility Secur.*, 2014, pp. 1–5.
- [41] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Real-time scheduling for WirelessHART networks," in *Proc. 31st IEEE Real-Time Syst. Symp.*, 2010, pp. 150–159.



**Weifeng Gao** (Student Member, IEEE) received the B.S. degree in computer science and technology in 2015 from the College of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, where he is currently working toward the Ph.D. degree in computer science and technology.

His research interests include mobile edge computing.



**Zhiwei Zhao** (Member, IEEE) received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2015.

He is currently an Assistant Professor with the College of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include wireless computing, edge computing, and heterogeneous Internet of Things.

Dr. Zhao is a Member of the Association for Computing Machinery and China Computer Federation.



**Zhengxin Yu** received the M.Sc. degree in information technology management for business in 2016 from the University of Exeter, Exeter, U.K., where she is currently working toward the Ph.D. degree in computer science with the College of Engineering, Maths and Physical Science.

Her research interests include deep learning, federated learning, and mobile edge computing.



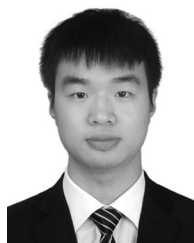
**Geyong Min** (Member, IEEE) received the B.Sc. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 1995, and the Ph.D. degree in computing science from the University of Glasgow, Glasgow, U.K., in 2003.

He is currently the Chair Professor and Director of the High Performance Computing and Networking Research Group, University of Exeter, Exeter, U.K. He joined the University of Bradford, Bradford, U.K., as a Lecturer in 2002 and became a Senior Lecturer in 2005, a Reader in 2007, and a Professor in Computer Science in 2012. His main research interests include computer networks, wireless communications, parallel and distributed computing, ubiquitous computing, multimedia systems, modeling and performance engineering.



**Minghang Yang** received the B.S. degree in computer science and technology in 2017 from the College of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, where he is currently working toward the M.S. degree in computer science and technology.

His research interests include mobile edge computing.



**Wenjie Huang** received the B.S. degree in automation from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2018. He is currently working toward the M.S. degree in computer science and technology with the University of Electronic Science and Technology of China, Chengdu, China.

His research interests include mobile edge computing.