A Local-Optimization Emergency Scheduling Scheme With Self-Recovery for a Smart Grid

Tie Qiu, *Senior Member, IEEE*, Kaiyu Zheng, *Student Member, IEEE*, Houbing Song, *Senior Member, IEEE*, Min Han, *Senior Member, IEEE*, and Burak Kantarci, *Senior Member, IEEE*

***Abstract*—With the widespread applications of Internet of Things (IoT), the emergency response performance for large-scale network packets is facing serious challenge, especially for renewable distributed energy resources monitoring in a smart grid. Therefore, how to improve the real-time performance of the emergency data packets has been a critical issue. Traditional packet scheduling schemes and topology optimization strategies are not suitable for a large-scale IoT-based smart grid. To address this problem, this paper proposes a new packet scheduling scheme named LOES, which first combines the priority-based packet scheduling scheme with local optimization. We exchange local geographic information to reduce the hop counts and distance between distributed source nodes and sink nodes. Each destination node determines the packet scheduling sequence according to the received emergency information. Finally, we compare LOES with first come first serve, multilevel scheme, and dynamic multilevel priority packet scheduling scheme using packet loss rate, packet waiting time, and average packet end-to-end delay as metrics. The simulation results show that LOES outperforms these previous scheduling schemes.**

***Index Terms*—Emergency information, local optimization, packet scheduling, smart grid.**

I. Introduction

THE Internet of Things (IoT) [1] has a rapidly gained ground in recent years. The main idea of IoT is to interconnect the objects in cyber-physical systems [2] to get the information through the enabling technologies including sensor networks [3], wireless communication, embedded systems, information security [4], and topology optimization [5]. Due to the wide applications of IoT, the network scale becomes huge, especially for renewable distributed energy resources monitoring in a smart grid. As a result, the packet types become diverse. Therefore, we should design an efficient strategy to ensure the timeliness of the emergency packets. The previous research works that are focused on the managing the sleep–wake times of nodes [6], [7] and the improvement of routing algorithms [8], [9] cannot efficiently address the issue. Therefore, an efficient packet scheduling scheme is quite important, because it schedules the data packets based on their priorities and reduces the end-to-end delay [10], [11]. Most of the IoT systems mainly use first come first serve (FCFS), which determines the sequence of packet processing and packet forwarding based on the order of the packets arrive at the node. Furthermore, researchers usually combine FCFS with queuing theory to solve the scheduling problem of multipriority packets [12]. However, with the increase of the networking scale, multiple data packets are sent to the same node at the same time in many cases. In the situation, if we still use FCFS, the packet collision will occur, which leads to the result that the data packets need to be resent and cannot arrive at the sink node within their deadlines. Some other studies about packets scheduling [13], [14] avoid the packet collision efficiently. However, these schemes do not maintain the topology dynamically. When some nodes are broken, the networks cannot continue working effectively. Therefore, we need an efficient packet scheduling scheme that not only can schedule packets based on their emergency information but also withstand failure of nodes in the large-scale sensor networks.

In this paper, we first combine the packet scheduling scheme with topology maintenance and local optimization and propose a local-optimization emergency scheduling scheme with self-recovery for the large-scale IoT-based smart grid, which is called LOES. Our major contributions are as follows.

1) We propose the two-hop-based local optimization strategy for the network topology with multiple sink nodes. The optimal node is chosen as the father node periodically in the local area. The local optimization strategy reduces the overhead comparing with the global optimization.
2) A novel emergency-aware mechanism is proposed. The emergency data packets are forwarded and processed first

Manuscript received January 24, 2017; revised April 18, 2017; accepted June 9, 2017. Date of publication June 16, 2017; date of current version December 1, 2017. This work was supported by the National Natural Science Foundation of China under Grant 61374154 and Grant 61672131, and the Fundamental Research Funds for the Central Universities (DUT17ZD216 and DUT16QY27). Paper no. TII-17-0137. *(Corresponding author: Houbing Song.)*

T. Qiu and K. Zheng are with the School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: qiutie@ieee.org; kaiyuzheng@mail.dlut.edu.cn).

H. Song is with the Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114 USA (e-mail: h.song@ieee.org).

M. Han is with the School of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116620, China (e-mail: minhan@dlut.edu.cn).

B. Kantarci is with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: bkantarc@uottawa.ca).

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

Digital Object Identifier 10.1109/TII.2017.2715844

when the multiple nodes send data packets to the same destination node at the same time, which ensures the timeliness of emergency data packets.

3) We compare LOES with the previous packets scheduling schemes in terms of packet loss rate, end-to-end delay, and waiting time. The simulation results show that LOES outperforms FCFS, multilevel scheme, and dynamic multilevel priority packet scheduling scheme (DMP).

The remainder of this paper is organized as follows. In Section II, we briefly introduce related work and discuss the existing problem. Our proposed packet scheduling scheme is described in Section III. Section IV is the implementation of the scheduling scheme. Section V presents the simulation results that compare LOES with FCFS, multilevel scheme [15], and DMP [16]. Finally, we conclude this paper and discuss our future work in Section VI.

II. Related Work and Problem Statement

A. Related Work

A smart grid should be able to provide new abilities such as self-healing, high reliability, energy management, and real-time pricing [17], [18]. Therefore, the packet scheduling scheme is important in the research of the smart grid. FCFS, earliest deadline first (EDF) [19], and rate monotonic scheme (RMS) [20] are the previous typical packet scheduling schemes. In FCFS, the packets that arrive late at the intermediate nodes will require some extra overhead to be delivered to the sink node. However, it is not flexible to use FCFS in a large-scale network. In EDF, the priorities of packets are dynamically adjusted according to their deadlines. Therefore, the overhead of EDF is more than that of FCFS. The RMS is mainly used in real-time operating systems for periodic packet scheduling. Thus, the RMS is limited in applications for the large-scale IoT-based smart grid.

With the increase of the variety and scale of the networks, preemptive scheduling schemes and cooperative scheduling schemes have become focus in recent years. In the preemptive scheduling scheme, the packets with higher priorities can preempt packets with lower priorities [21]. EF-RM [22] is a preemptive scheduling scheme, which is used in TinyOS. This scheme is more efficient than the RMS. Yaghmaee and Adjeroh [23] propose a priority-based packet scheduling scheme using a cooperative scheduling method among the different queues.

In recent three years, the real-time requirement of packet scheduling schemes has become more and more important with the expansion of network scale and applications. The researchers have got some achievements. Chennakesavula *et al.* [24] propose an effective real-time packet scheduling policy, which schedules the incoming packets based on the remaining time and the remaining distance to the destination node. Yin *et al.* [25] propose a data-processing model using queuing theory to allocate the priority. Nidal *et al.* [16] propose the DMP scheme. A time-division multiple-access (TDMA) method is employed, which can effectively process data packets based on their priorities.

Furthermore, besides what have mentioned above, the optimization for the network topology can also reduce the end-to-end delay of data packets. Takashima and Ikezaki [26] propose the expanded spanning-tree protocol, which ensures that each data packet can establish an optimal path to the destination node in the tree-based network. Zhang *et al.* [27] design an novel energy-efficient routing protocol based on a least spanning tree. Lachowski *et al.* [28] propose a distributed algorithm for constructing spanning trees, which is based on the Bellman–Ford distributed asynchronous version.

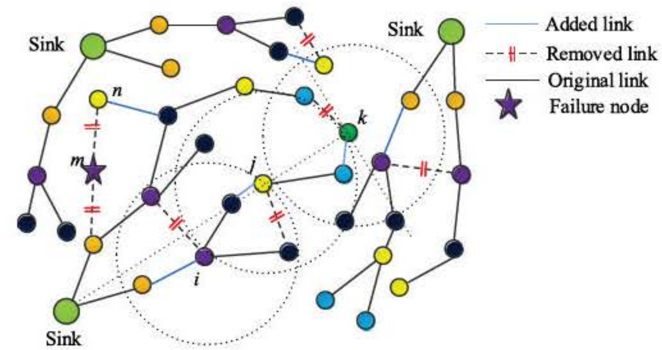

Fig. 1. Node failure and local adjustment in multisink sensor networks of the IoT-based smart grid.

B. Problem Statement

In the real large-scale sensor networks of the IoT-based smart grid, some nodes occasionally fail due to breakdown and energy depletion. However, these nodes play an important role in forwarding data packets for a self-organizing tree-based network. If we still employ the previous schemes, the failure of nodes will bring a great impact on the real-time performance of the smart grid. In the traditional packet scheduling schemes, they only emphasize the dynamic adjustment of packets priorities, so that the packets with higher priorities can be processed and forwarded first to ensure that they can arrive at the sink node as soon as possible. However, there is not an effective solution to the problem about the impact of failure of nodes on the link. Therefore, how to dynamically adjust the network topology is an urgent requirement in the packet scheduling scheme for the IoT-based smart grid.

The situation of node failure and local adjustment in multisink sensor networks of the IoT-based smart grid is shown in Fig. 1. The network is built in the spanning tree protocol. The node with the same color is at the same level. According to the local-optimization and local-adjustment strategy, the node i can be connected to a father node with the smaller hop counts from the sink node, nodes j and k can be connected to the closer nodes. Therefore, we need to make the local optimization. In this way, the data packets from node i, node j, and node k can arrive at the sink node more quickly, which can further ensure the timeliness of the emergency data packets and avoid the additional waste of energy. Moreover, if the node m fails, its subtree loses connection with the network, which leads to the packets from these nodes dropped. Therefore, how to ensure the recovery of networks is also a crucial issue for the sensor networks with failure of nodes.

In order to solve these problems above, we propose a local-optimization emergency scheduling scheme with self-recovery for multisink sensor networks of the IoT-based smart grid.

III. LOES Scheme

In the sensor networks of the IoT-based smart grid, the nodes are organized based on multisink tree-based networks. Each node records its node level according to the hop counts from the sink node when the tree-based network is constructed. The intermediate nodes can generate and forward the data packets. Every node in the network has an unique ID number and maintains a neighbor list. In the neighbor list, there are ID numbers of father and child nodes, MAC-address information, and geographic information. When the tasks at the node have been completed, the node will switch into the sleep mode. The multichannel MAC protocol is used when the different nodes on the same branch send the data packets at the same time. In addition, we adopt the basic location technology instead of the GPS system in order to decrease the energy consumption [29]. Through the chosen beacon node, every node can know their geographic coordinates before the packet scheduling scheme works.

The LOES scheme consists of three main phases: topology optimization, packets scheduling, and topology maintenance with self-recovery. Among them, there are eight types of messages in topology optimization and topology maintenance. The message format is defined as follows.

DestAddr	SrcID	Type	FathID	Level	Flag	FathPos

1) DestAddr: the MAC address of the destination node.
2) SrcID: the ID number of the sending node.
3) Type: the type of messages including ChkFN, ChkAck, RecNet, RecAck, OptHop, HopAck, OptDis, and DisAck.
4) FathID: the father node's ID number for the sending node. The value of the root node is -1.
5) Level: the node level at which the sending node is.
6) Flag: the network connection flag. If the node is in the network, the value is 1. Otherwise, the value is 0. We initialize the parameter flag to 0. After we build the network, we will update the flag of all nodes that are in the network to 1.
7) FathPos: the geographic information of the father node. The default value is -1.

A. Topology Optimization

1) Optimization Based on Hop Counts: After the tree network is constructed based on the spanning tree protocol, we first make the topology optimization. Each node in the network broadcasts a message OptHop in turn. Meanwhile, the sending node sets a timer, whose value is set based on the node communication range to ensure that the destination nodes at the edge of the communication range can reply to the source node within the timer. The nodes that have received the message OptHop will put their node level information into the messages HopAck and send them to the source node. When the timer expires, the source node sorts all the received messages HopAck in a heap. Then, we compare minimum $level$ in the heap with the $level$ of the current node. If the minimum $level$ is less than $level - 1$ of the current node, the father node is replaced by the node sending the reply message. Meanwhile, the $level$ of the current node needs to be updated. Otherwise, we continue the traversing process.

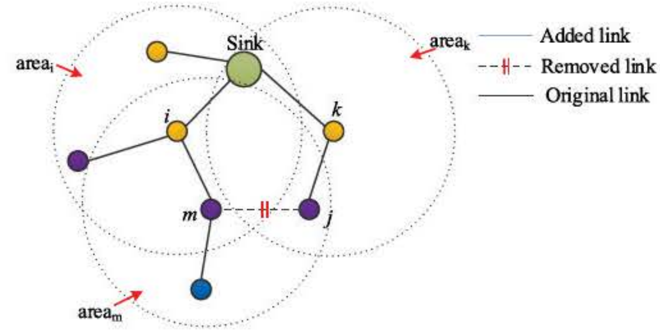

Fig. 2. Topology optimization based on hop counts.

TABLE I
DEFINITIONS OF VARIABLES

Symbols	Description
PA	MAC-address packet
H_{id}	The ID of the packet with the highest emergency
PN_i	The number of packets with pr_i
P_i	The set of pr_i packets
PK_a	The matrix of emergency information packets
T	The set including all nodes of tree topology
Pos	The location information
AR	The temporary array for storing data
NL	The neighbor list
N_r	The root node

The process of topology optimization based on hop counts from the sink node is shown in Fig. 2. Node m is the father node of node j. The node level of node m is 3. When node j sends the broadcast message, the minimum hop count in the reply message is 1. Therefore, node j disconnects the connection with node m and connects to node k. Algorithm 1 realizes the topology optimization based on hop counts. The definition of variables is shown in Table I. There is no loop in Algorithm 1. But Algorithm 1 calls the MinHeap Algorithm and the complexity of MinHeap Algorithm is $\mathcal{O}(nlogn)$. Therefore, the complexity of Algorithm 1 is $\mathcal{O}(nlogn)$.

After that, the hop counts from the sink node is the smallest for each node in the network. Then, each node reselects a more suitable father node based on the distance in the next work in this phase.

2) Optimization Based on Distance: Starting from the sink node, the nodes in the network broadcast the message OptDis according to the DFS method. The destination node sends a message DisAck to the source node after receiving the broadcast message. Then, the source node analyzes the $FathID$ and $level$ in the messages DisAck. If there is only one reply message in which the $level$ is less than the source node's, the source node

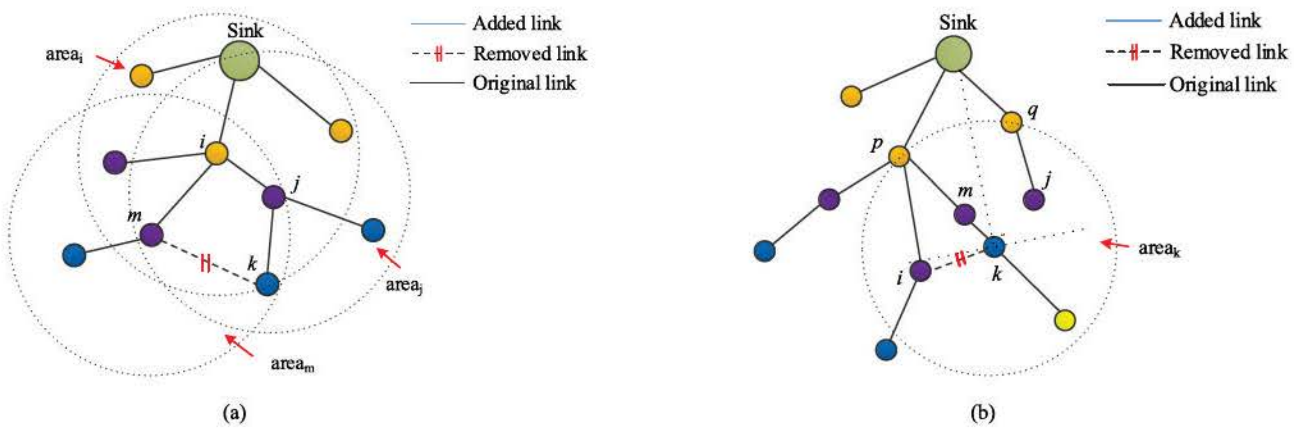

Fig. 3. Topology optimization based on distance. (a) Father nodes are same. (b) Father nodes are different.

Algorithm 1: Topology optimization based on hop counts.

Input: T
Output: T

□Upon the current node is traversed
Broadcast message OptHop;
Start a timer and wait for receiving the messages HopAck;

□Upon receiving a message HopAck
Put message HopAck into array AR;

□Upon timer expires
Call MinHeap algorithm with AR as the parameter based on the $HopAck.level$;
if minimum($HopAck.level$)< level-1 **then**
    $FathID \leftarrow HopAck.ID$
    $level \leftarrow HopAck.level - 1$
end if

does not do any processing. Otherwise, the source node compares the ID numbers in different reply messages with each other. At this moment, there are two situations as shown in Fig. 3. We assume that $d_{i,j}$ denotes the distance from node i to node j. $d_{i,s}$ is the distance from node i to sink node.

1) The father nodes are same as shown in Fig. 3(a). The node k has two possible father nodes: node m and node j, that is, there are two possible paths from node k to node i. Thus, we need two-hop geographic information of node k to obtain the distance information of two paths. Then, we compare them, $d_{i,m} + d_{m,k} > d_{i,j} + d_{j,k}$ is given. Therefore, we select node j as the father node of node k. In the following, we give Proposition 1.

Proposition 1: In the local area with two hops when the grandfather nodes are same, the shortest transmission path $d_{\min}$ corresponds with the following equation:

$$d_{\min} = d_{i,s} + d_{i,j} + d_{j,k}. \tag{1}$$

Proof: The distance from node k to the sink node is $d_{k,s} = d_{i,s} + d_{i,k}$, where $d_{i,k}$ may have two possible values: $d_{i,m} + d_{m,k}$ or $d_{i,j} + d_{j,k}$. Therefore, we can use (2) to represent the shortest distance from nodes k to the sink node.

$$d_{\min} = d_{i,s} + \min\{d_{i,m} + d_{m,k}, d_{i,j} + d_{j,k}\}. \tag{2}$$

Because $d_{i,m} + d_{m,k} > d_{i,j} + d_{j,k}$, we obtain

$$\min\{d_{i,m} + d_{m,k}, d_{i,j} + d_{j,k}\} = d_{i,j} + d_{j,k}. \tag{3}$$

Combining (2) and (3), the shortest distance $d_{\min}$ from node k to the sink node is calculated as (1).

Therefore, we select the node j as the father node of the node k.

2) The father nodes are different as shown in Fig. 3(b). Within the communication range of node k, there are three nodes (i, j, and m) at the upper node level of node k. Among them, the father node of node i and node m is node p, the father node of node j is node q, which are different nodes. Obviously, the strategy of the first situation is not applicable. At this moment, we need to analyze the node i, node j, and node m based on their geographic information. Then, we get that node i is further from the sink node compared with node k. However, node j and node m are close to the sink node. Therefore, we need to get a local optimal selection between node j and node m.

Proposition 2: In the local area when the grandfather nodes are different and the possible father nodes are located at different sides of the source node, we select the node at the closer side to the sink node as the local optimization choice.

Proof: When the selected nodes are located on different sides (the closer position to the sink node and the further position to the sink node) of the source node, we give the mathematical model as shown in Fig. 4(a). We assume that $d_{j,k} = d_{i,k}$. Because node j is closer to the side of the sink node, the angle range of θ_2 is $0^\circ < \theta_2 \leq 90^\circ$. When $\theta_2 = 90^\circ$, the relative distance between node j and the sink node is the farthest. At this moment, we assume that the actual transmission distance between them is the farthest. According to the Pythagorean theorem, we obtain

$$d_{s,j}{}^2 = d_{s,k}{}^2 + d_{j,k}{}^2. \tag{4}$$

Because $\theta_1 > 90^\circ$, we obtain

$$d_{s,i}{}^2 > d_{s,k}{}^2 + d_{i,k}{}^2. \tag{5}$$

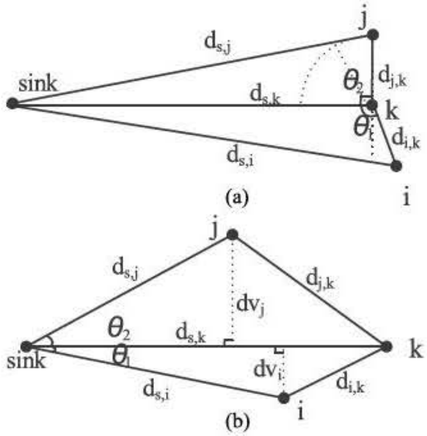

Fig. 4. Mathematical model based on distance. (a) Nodes are located on different sides. (b) Nodes are located on same sides.

Because $d_{j,k} = d_{i,k}$, $d_{s,i} > d_{s,j}$ can be deduced. Therefore, we select node j as the father node of node k. ∎

Proposition 3: In the local area when the grandfather nodes are different and the possible father nodes are located at the same side of the source node, we select the node that is closer to the connection line passing the source node and the sink node as the local optimization choice. The shortest transmission path $d_{\min}$ corresponds to

$$d_{\min} = d + d_{i,k}. \tag{6}$$

Proof: When the selected nodes are located at the same side of the source node, the mathematical model is shown in Fig. 4(b).

We assume that $d_{s,j} = d_{s,i} = d$ and $dv_j > dv_i$. Thus, the shortest distance $d_{\min}$ from node k to the sink node is shown as

$$d_{\min} = d + \min\{d_{j,k}, d_{i,k}\}. \tag{7}$$

According to the Cosine theorem, we obtain

$$\begin{cases} d_{j,k} = \sqrt{d^2 + d_{s,k}^2 - 2 * d * d_{s,k} * \cos\theta_2} \\ d_{i,k} = \sqrt{d^2 + d_{s,k}^2 - 2 * d * d_{s,k} * \cos\theta_1}. \end{cases} \tag{8}$$

Because $\cos\theta_1 = \sqrt{1 - \sin\theta_1^2}$ and $\cos\theta_2 = \sqrt{1 - \sin\theta_2^2}$, (9) is given combining (8).

$$d_{i,k}^2 - d_{j,k}^2 = 2 * d * d_{s,k} * \left(\sqrt{1 - \sin\theta_2^2} - \sqrt{1 - \sin\theta_2^2}\right). \tag{9}$$

According to the Sine theorem, $\sin\theta_1 = dv_i/d$ and $\sin\theta_2 = dv_j/d$ can be obtained. Because $dv_j > dv_i, 0 < \sin\theta_1 < 1, 0 < \sin\theta_2 < 1$, we know that $\sin\theta_2 > \sin\theta_1$. Thus, $d_{i,k}^2 - d_{j,k}^2 < 0$, that is, $d_{i,k} < d_{j,k}$.

Combining (7), the shortest distance $d_{\min}$ from node k to the sink node is calculated as (6). ∎

Algorithm 2 focuses on the topology optimization based on distance. There is no loop in Algorithm 2. Meanwhile, the other algorithms are not called. Therefore, the complexity of Algorithm 2 is $\mathcal{O}(1)$.

Algorithm 2: Topology optimization based on distance.

Input: T
Output: T

Broadcast message OptDis;
Start a timer and wait for receiving messages DisAck;

□Upon receiving the message DisAck
if $DisAck.level == level - 1$ **then**
    Put message DisAck in array AR;
end if

□Upon timer expires
if size(AR) ≥ 2 **then**
    if the $FathID$ of the elements in AR are same **then**
        Get the minimum distance between the current node and its grandfather node;
        Update $FathID$;
    else
        Delete the elements in AR below the line;
        Sort the elements in AR based on the distance between them and the current node;
        $FathID \leftarrow AR[0].SrcID$
    end if
end if

B. Packets Scheduling

In this phase, the data packets are processed, scheduled, and forwarded based on their emergency information. The data packets are divided into three types according to their priorities and deadlines.

1) Emergency data packets (pr_1). This type of packets needs the quickest response. Thus, these packets' end-to-end delays must be reduced as much as possible.
2) Normal data packets (pr_2). The most of data packets in network are of this type. The emergency data packets can preempt this type of data packets.
3) Nonemergency data packets (pr_3). They have the lowest requirement on delay. Compared with other data packets. Their deadlines are longer.

Fig. 5 shows the packets scheduling scheme. In this packets scheduling scheme, there are three units: access control unit (ACU), emergency-aware unit (EAU), and packet forward unit (PFU). The incoming packets not only have data packets, but also have emergency information packets and MAC-address packets. The packet analysis (PA) can distinguish the incoming packets. Through the analysis of PA, the data packets are sent to conditional access control to check whether the deadlines expire, the emergency information packets are sent to EAU, and the MAC-address packets are sent to PFU for further processing. Then, the packets within their deadlines are placed into priority queue. Corresponding to three different priorities, each node has three priority queues. In the same priority queue, the packets are sorted based on their deadlines. Among the three priorities, we select the packet with the highest emergency to extract its

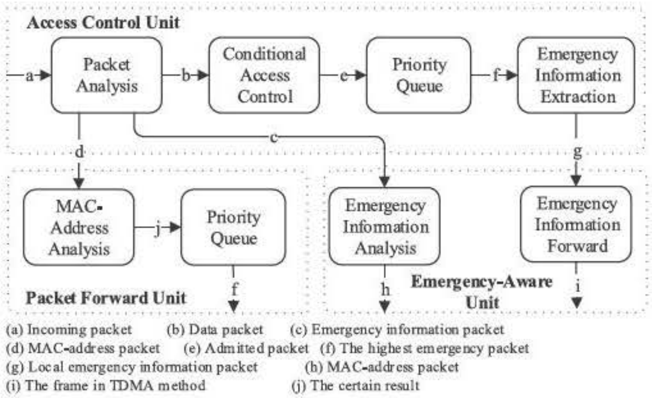

Fig. 5. Packet scheduling scheme.

emergency information. The generated emergency packet will be sent to EAU.

In EAU, emergency information forward (EIF) sends the emergency information packets that get from local's ACU and sibling nodes' ACUs to the destination node in the TDMA method. When there is only an emergency information packet at EIF, we send the data packet directly instead of sending the emergency information packet. EIA mainly analyzes the emergency information packets received from the child nodes. It will get the emergency information packet with the highest priority and the shortest deadline among these packets. Then, the MAC address of the node where the most emergency packet is sent can be known. We put the MAC address into the MAC-address packet and broadcast it.

The function of PFU is packet forwarding. When the MAC-address analysis module receives the MAC-address packet, it will compare the current node's MAC address with the MAC address in the received packet. If they are same, the current node will monopolize the channel to send the data packet. Otherwise, the node has been waiting until the next round of sending the emergency information packets. Before sending packets to the destination node, the node needs to check the channel state. If the channel state is idle, the node takes the channel to send packets. Otherwise, it needs to wait until the channel state has changed to idle. The function of Algorithm 3 is to process the packet scheduling scheme. There are two loops in Algorithm 3. The first loop is to scan the array PK_a and the size of PK_a is N. The complexity of this loop is $\mathcal{O}(n)$. The second loop is executed thrice. So its complexity is $\mathcal{O}(1)$. Therefore, the complexity of Algorithm 3 is mainly influenced by the first loop and it is $\mathcal{O}(n)$.

For different data packets, the forwarding time and processing time at the same node are equal. Therefore, the end-to-end delay is mainly influenced by the waiting time at the nodes for different kinds of packets. In order to analyze the timeliness of packets with different priorities, we formulate the waiting time of the LOES scheme. The pr_1 packets are sorted according to the deadlines. We assume that $N_{i,j}$ represents the number of sending pr_j packets from the node at node level L_i.

Algorithm 3: Process of packets scheduling.

Input: PK_a
Output: PA

```
□Upon receiving emergency information packets
for n ← 1 to N do
    if PK_a[n][1] == i then
        PN_i ← PN_i + 1
        Put H_id into P_i
    end if
end for
for i ← 1 to 3 do
    if PN_i > 1 then
        Sort the elements in P_i based on their deadlines;
    end if
end for
Get the packet with the highest priority and the shortest deadline to update PA;
Broadcast the PA;
The node whose MAC address is equal to PA sends the whole packet;
```

te_m denotes the time from sending m emergency information packets to receiving the broadcast packets. t_p denotes the forwarding delay. $k_{i,j}$ denotes the number of packets whose priorities are same as j, but the deadlines are shorter than current packet's. n is the original hop counts from the sink node. After we make the topology optimization, the value of n is likely to be reduced. Therefore, the total waiting time of pr_1 packet is given as

$$t_1' \le \sum_{i=1}^{n} \left[k_{i,1} * t_p + \sum_{m=N_{i,1}-k_{i,1}}^{N_{i,1}} te_m \right]. \tag{10}$$

The pr_2 packets need to wait for the transmission of the pr_1 packets, and they also need to wait for the pr_2 packets with the shorter deadlines. Thus, the total waiting time for pr_2 packet is given as

$$t_2' \le \sum_{i=1}^{n} \left[(k_{i,2} + N_{i,1}) * t_p + \sum_{m=N_{i,1}+N_{i,2}-k_{i,2}}^{N_{i,1}+N_{i,2}} te_m \right]. \tag{11}$$

Similarly, the waiting time of the pr_3 packet is given as

$$t_3' \le \sum_{i=1}^{n} (k_{i,3} + N_{i,1} + N_{i,2}) * t_p + \sum_{i=1}^{n} \sum_{m=N_{i,1}+N_{i,2}+N_{i,3}-k_{i,3}}^{N_{i,1}+N_{i,2}+N_{i,3}} te_m. \tag{12}$$

From the above, we can see that LOES can deal with the emergency data packet first. The local optimization of the topology can significantly reduce the hop counts from the sink node, which makes the emergency packets arrive at the sink node more quickly.

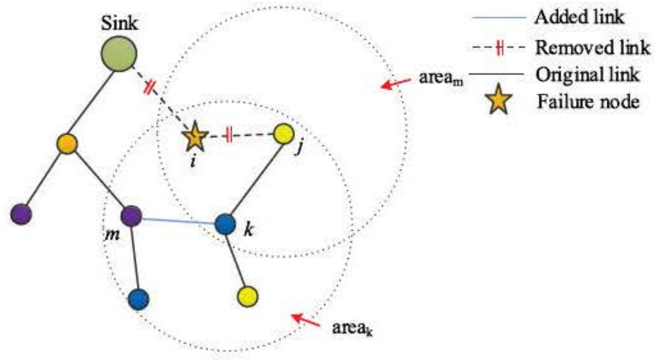

Fig. 6. Process of topology maintenance.

C. Topology Maintenance With Self-Recovery

The work of this phase mainly focuses on seeking the failure of nodes in network and bringing the network system back to normal. At the beginning of this phase, each node in the network broadcasts a message ChkFN to seek the failure of nodes. A timer is started when the message ChkFN is broadcasted. Once the destination node has received the message ChkFN, it analyzes whether the source node in the message ChkFN is the neighbor node. If the source node is the neighbor node, the destination node will reply a message ChkAck to the source node. Otherwise, the destination node will throw away the received message ChkFN. After the expiration of the timer, the source node will carry out statistics on all received messages ChkAck. If the all neighbor nodes send the message ChkAck to the source node, the neighbor nodes of the source node are working properly. Otherwise, the node that did not send the message ChkAck to the source node is the failure node. When the failure node is the source node's father node, we set the network connection $flag$ of every node in the subtree with the source node as root node to 0. Then, we traverse each node in the subtree using the DFS method. The node that is traversed will broadcast the message RecNet to all nodes in the communication range. The destination node that receives the message will reply to a message RecAck. Once the network connection $flag$ is 1 in the message RecAck, the source node sets the ID number of the father node to the $SrcID$ in the message RecAck. Whereafter, the network connection $flag$ and $level$ of the nodes in the subtrees are updated based on the received message RecAck. At this moment, the subtrees of the failure node reconnect to the network successfully. Algorithm 4 shows the process of topology maintenance. There is a loop in Algorithm 4 to scan array NL and the array length is constant. Thus, the complexity of Algorithm 4 is $\mathcal{O}(n)$.

The process of topology maintenance is shown in Fig. 6. When the first broadcast is completed, the sink node and node j find that node i has not replied the message ChkAck to them. Then, the $flag$ of the nodes in the subtree with node j as the root node are set to 0. Next, the DFS method is used to traverse every node in the subtree. When traversing to node j, node j cannot receive any message RecAck within the timer. Differently, when traversing to node k, node k will receive the message RecAck from node m. We know that node m is in the network by analyzing the received message. Therefore, node k sets its $FathID$ to the $SrcID$ number in the message RecAck and updates the $flag$ and $level$ of all nodes in the subtrees with node k as the root node. After this phase, the subtrees of the failure node have reconnected back to the network.

Algorithm 4: Seeking the failure of nodes.

Input: T, N_r
Output: Pos, T

Broadcast message ChkFN;
Start a timer and receive messages ChkAck from neighbor nodes during the timer;
Put the ID numbers in receiving messages ChkAck into array AR;

□Upon timer expires
for $i \leftarrow 0$ to size(NL)-1 **do**
    if $NL[i].ID$ is not in arry AR **then**
        $Pos \leftarrow NL[i].Pos$
    end if
end for
if Pos was updated **then**
    if $Pos == FathNode.Pos$ **then**
        Set $flag$ of each node in the subtrees to 0;
    else
        Send Pos to the sink node;
    end if
end if
Traverse the nodes in the subtree using the DFS method;
Broadcast message RecNet;
Start a timer and wait for receiving message RecAck.

□Upon receiving message RecAck
if $RecAck.flag == 1$ **then**
    $FathID \leftarrow RecAck.ID$
    $level \leftarrow RecAck.level + 1$
    Update the $level$ of every node in the subtree;
    Return T;
els if $RecAck.flag == 0$ **then**
    Wait for receiving message RecAck again;
end if

□Upon timer expires
Traverse the next node based on the DFS method;

IV. Simulation Results

A. Simulation Setup

In this section, we evaluate our proposed LOES with the waiting time, the end-to-end delay, and the packet loss rate as metrics. The simulations are run in NS2. In addition, we compare LOES with the other packet scheduling schemes including FCFS, multilevel scheme, and DMP. A total of 500 nodes are randomly deployed in a sensor field of 300 m $\times$ 300 m. Every data packet in the network needs to be processed at the sink

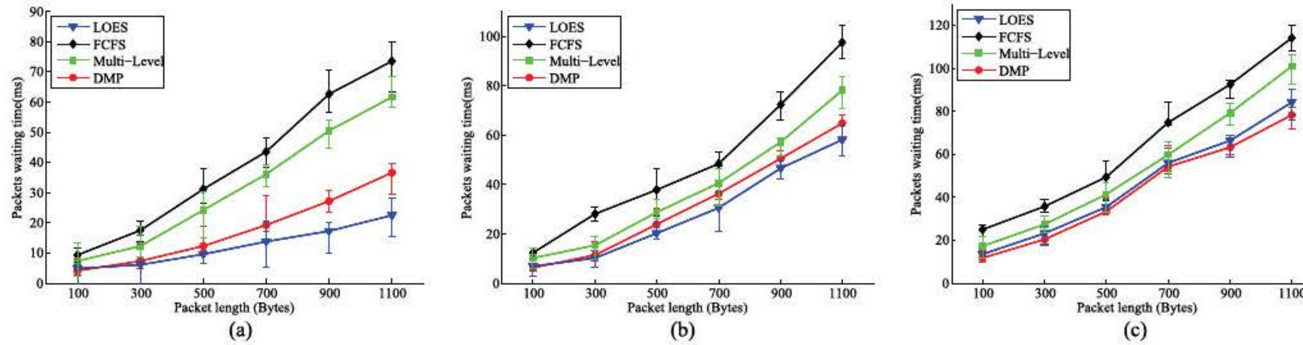

Fig. 7. Waiting time of packets with different priorities. (a) pr_1 packets. (b) pr_2 packets. (c) pr_3 packets.

TABLE II
SIMULATION SETTINGS

Parameter	Description
Network Size	300 m × 300 m
Communication Radius	20 m
Channel	Wireless channel
Traffic Patterns	CBR
Number of Nodes	Maximum 500
Transmission Speed	250 kb/s
Packet Size	100 B/300 B/B/500 B/700 B/900 B/1100 B
Simulation Time	100 s

node. The intermediate nodes randomly generate the data packets. Then, the priorities and the deadlines are randomly setting. The priorities of the data packets remain unchanged, but the deadlines are reduced with the executing time of LOES. The simulation settings are shown inTable II.

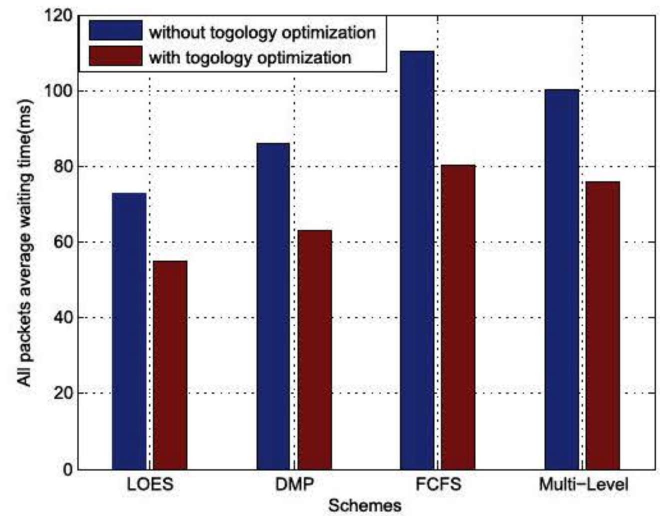

Fig. 8. Average waiting time in different situations.

B. Waiting Time

In the simulation experiments, we control the packet generation rate to simulate the normal network load. In order to improve the accuracy of the simulation, three different experimental situations are set up. The ratio of pr_1 packets, pr_2 packets, and pr_3 packets is set as $\{3{:}5{:}2,\ 1{:}1{:}1,\ 5{:}3{:}2\}$ corresponding to these three situations. The average value of the experimental results obtained from the three situations is the final result.

Fig. 7 illustrates the final simulation result. Each value in the chart represents an average of waiting time under three network conditions proposed above. The error bar represents the worst waiting time and the best waiting time. From Fig. 7(a)–(c), we can obtain that the waiting time of the data packet is reduced with the increase of the priority for any scheme in our simulation. It shows that these schemes can effectively complete the priority-based packet scheduling. Comparing them with each other in the worst waiting time, it can be seen that the performance of LOES is the best among these four schemes for pr_1 packets and pr_2 packets. For pr_3 packets, LOES and DMP have obvious advantages over FCFS and the multilevel scheme. Additionally, the waiting time of LOES is slightly longer than that of the DMP. The reason is that LOES allocates more network resources to process the packets with the higher emergency, which can ensure that they can be scheduled first. However, the emergency packets are treated equally with the nonemergency packets in the DMP due to using the TDMA method all the time. The average waiting time of data packets based on different schemes is illustrated in Fig. 8. The result shows that the packet scheduling scheme with the topology optimization is much better than the packet scheduling scheme without the topology optimization in the average waiting time, which proves that the topology optimization is efficient in decreasing the waiting time for the packet scheduling scheme. Among them, LOES with optimization is the best and FCFS without the topology optimization is the worst.

C. Packets Loss Rate

We make comparison with FCFS, DMP, and multilevel scheme in terms of the packet loss rate. Fig. 9 shows the packets loss rate under different packet lengths. We can see that LOES has the lowest packet loss rate than the other three schemes. The packets loss rate of the DMP increases obviously with the increase of packet length, far more than that of LOES. Both FCFS and multilevel scheme suffer from low efficiency. The reason is that the congestion can occur in FCFS and multilevel scheme. Then, the data packets will be resent to the destination node, which further aggravates the congestion of the network.

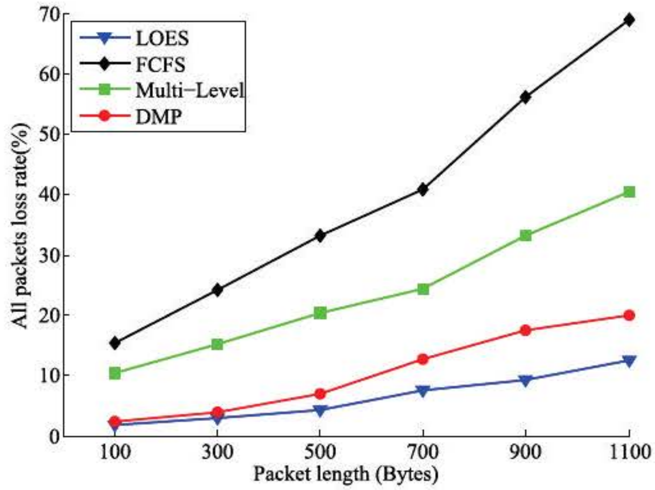

Fig. 9. Packet loss rate under different packet lengths.

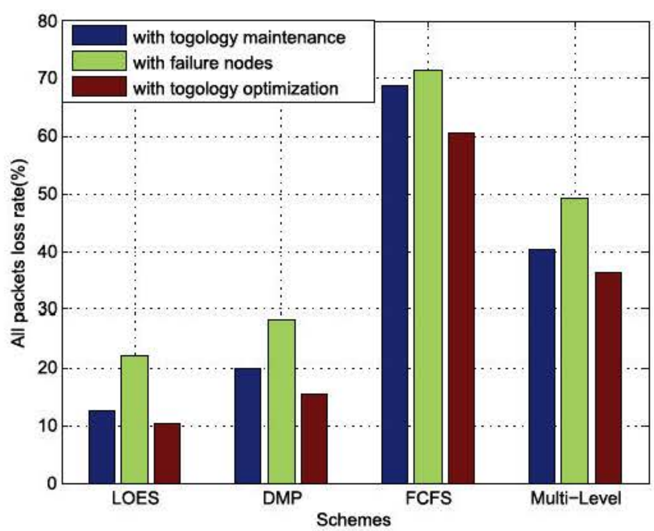

Fig. 10. Packet loss rate in different situations.

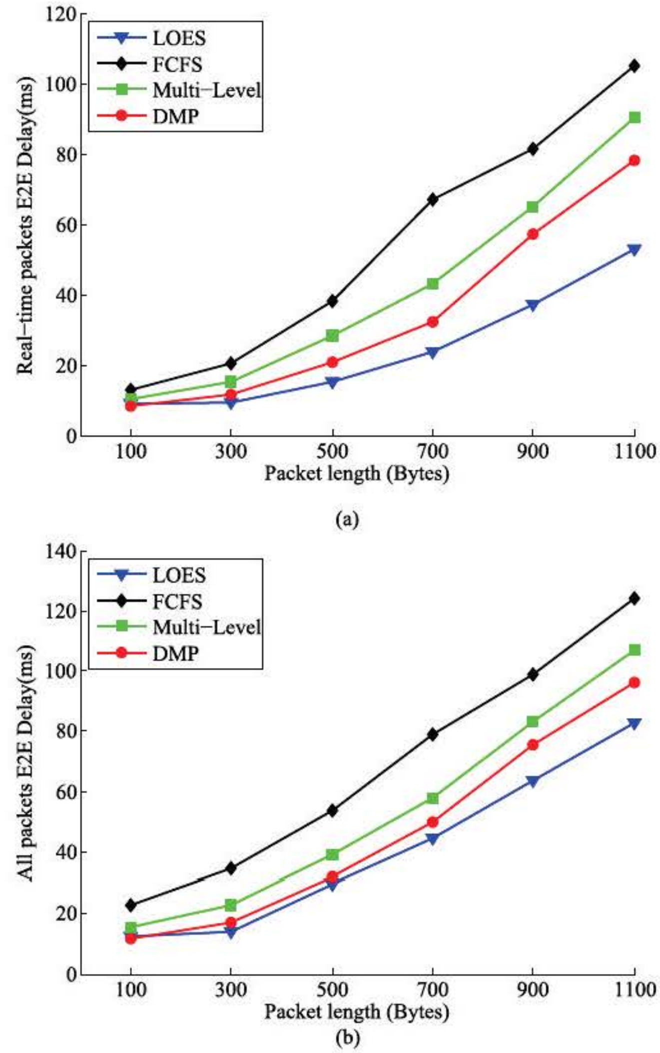

Fig. 11. End-to-end delay under different packet lengths. (a) End-to-end delay of emergency packets. (b) End-to-end delay of all packets.

However, LOES and DMP use a TDMA method to control the time slots. Thus, there is no packet collision in LOES and DMP, theoretically. Therefore, the packet loss rate of LOES and DMP are much lower than FCFS and multilevel scheme. In the DMP, the packet is discarded at the sink node when its deadline expires. When the packet length is short, sending a data packet is quite fast. Thus, the packets can arrive at the sink node within the deadline. When the packet length becomes longer, the time slot has to be extended. The accumulation of several time slots will lengthen the waiting time. Packets fail to get forwarded to the sink node within the deadline and dropped, which lead to increasing the packet loss rate. In LOES, the packets with the shortest deadlines will be forwarded first to ensure the efficiency.

Fig. 10 illustrates the packet loss rate in different situations. From Fig. 10, we can know that when there are failure of nodes in the network, the packet loss rate will increase compared with the network in the normal state for these four scheduling schemes. The reason is that the sink node cannot receive the data packets that are from the nodes in the subtrees of the failure node. After doing topology maintenance, the subtrees of the failure node reconnect to the network, and the data packets that come from the nodes in the subtrees can send to the sink node. Thus, the packet loss rate has reduced. The scheduling scheme with the topology optimization is the best.

D. End-to-End Delay

We compare LOES with FCFS, multilevel scheme, and DMP in the end-to-end delay. Fig. 11 illustrates the average end-to-end delay of emergency packets and all packets under different packet lengths. Comparing Fig. 11(a) with Fig. 11(b), it is clear that the results in Fig. 11(a) have the similar trend to Fig. 11(b). However, the end-to-end delay of all packets is larger than that of emergency packets under the same packet length. When the packet length is 100 bytes, the real-time performance of LOES and DMP are approximately similar. With the increase of the packet length, the advantage of LOES on real-time performance

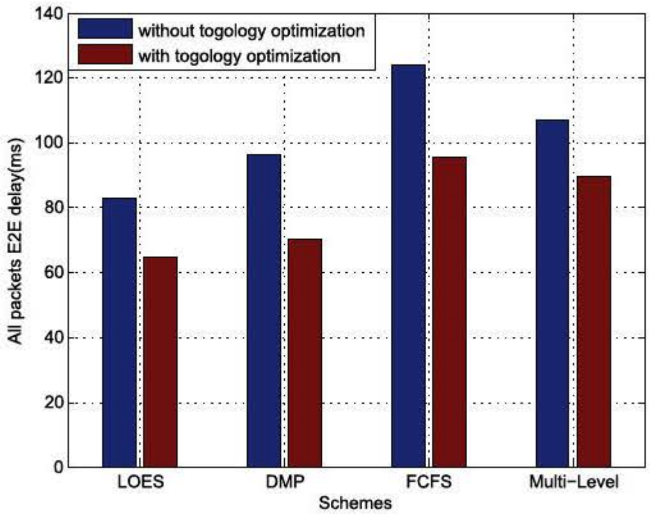

Fig. 12. Average end-to-end delay in different situations.

will be more obvious, especially in the average end-to-end delay of emergency packets.

Fig. 12 compares the average end-to-end delay in different situations based on four scheduling schemes. The results show that the end-to-end delay using the scheduling scheme with the topology optimization is lower than that using the scheduling scheme without the topology optimization. After the topology optimization, each node in the network has the minimum hop counts and the shortest distance from the sink node. This is the reason why the scheduling scheme with the topology optimization has the lower end-to-end delay.

V. Conclusion

In order to improve the real-time performance for emergency data packets, reduce the overhead of the packets transmission, and explore the recovery capability for large-scale sensor networks of the smart grid, this paper proposes LOES, a local-optimization emergency scheduling scheme with self-recovery for multisink sensor networks of the smart grid. We first combine the packet scheduling scheme with topology maintenance and local optimization. According to the packet scheduling scheme, the destination node can get emergency information of every data packet from the source node. Based on the emergency information of each data packet, we can allocate the network resources rationally to ensure the timeliness of emergency packets. The topology maintenance is to seek the failure of nodes in the whole network and reconnect the subtrees of the failure of nodes, which ensures that the network has recovery capability. The local-optimization strategy makes every data packet forward through the shortest path in local area networks. Finally, we carry out simulations to evaluate LOES. The simulation results show that LOES is better than FCFS, DMP, and multilevel schemes in terms of the waiting time, the average end-to-end delay, and the packet loss rate.

LOES explores the priority-based packet scheduling and improve the speed of emergency response. However, it only meets the packet scheduling in the tree-based networks. Our future work will focus on how to deal with the emergency packets in the star networks and the mesh networks to ensure the timeliness of emergency packets.

References

[1] S. Jeschke, C. Brecher, H. Song, and D. Rawat, *Industrial Internet of Things: Cybermanufacturing Systems*, 1st ed. Cham, Switzerland: Springer, 2017.

[2] H. Song, D. Rawat, S. Jeschke, and C. Brecher, *Cyber-Physical Systems: Foundations, Principles and Applications*, 1st ed. Boston, MA, USA: Academic, 2016.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[4] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, 2013.

[5] T. Qiu, D. Luo, F. Xia, N. Deonauth, W. Si, and A. Tolba, "A greedy model with small world for improving the robustness of heterogeneous Internet of Things," *Comput. Netw.*, vol. 101, no. 6, pp. 127–143, 2016.

[6] G. Anastasi, M. Conti, and M. Di Francesco , "Extending the lifetime of wireless sensor networks through adaptive sleep," *IEEE Trans. Ind. Informat.*, vol. 5, no. 3, pp. 351–365, Aug. 2009.

[7] Y. Xiao *et al.*,"Coverage and detection of a randomized scheduling algorithm in wireless sensor networks," *IEEE Trans. Comput.*, vol. 59, no. 4, pp. 507–521, Apr. 2010.

[8] A. A. Ahmed and N. Fisal, "A real-time routing protocol with load distribution in wireless sensor networks," *Comput. Commun.*, vol. 31, no. 14, pp. 3190–3203, 2008.

[9] W. M. Aioffi, C. A. Valle, G. R. Mateus, and A. S. Cunha, "Balancing message delivery latency and network lifetime through an integrated model for clustering and routing in wireless sensor networks," *Comput. Netw.*, vol. 55, no. 13, pp. 2803–2820, 2011.

[10] K.-H. Phung, B. Lemmens, M. Goossens, A. Nowe, L. Tran, and K. Steenhaut, "Schedule-based multi-channel communication in wireless sensor networks: A complete design and performance evaluation," *Ad Hoc Netw.*, vol. 26, pp. 88–102, 2015.

[11] Y. Xue, B. Ramamurthy, and M. C. Vuran, "SDRCS: A service-differentiated real-time communication scheme for event sensing in wireless sensor networks," *Comput. Netw.*, vol. 55, no. 15, pp. 3287–3302, 2011.

[12] O. Chipara, C. Lu, and G.-C. Roman, "Real-time query scheduling for wireless sensor networks," in *Proc. Real Time Syst. Symp.*, Dec. 3–6, 2007, pp. 389–399.

[13] R. Gomathi and N. Mahendran, "An efficient data packet scheduling schemes in wireless sensor networks," in *Proc. Int. Conf. Electron. Commun. Syst.*, Feb. 26–27, 2015, pp. 542–547.

[14] X. Xu, X. Li, and M. Song, "Distributed scheduling for real-time data collection in Wireless Sensor Networks," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 9–13, 2013, pp. 426–431.

[15] E.-M. Lee, A. Kashif, D.-H. Lee, I.-T. Kim, and M.-S. Park, "Location based multi-queue scheduler in wireless sensor network," in *Proc. Int. Conf. Adv. Commun. Technol.*, Feb. 7–10, 2010, pp. 551–555.

[16] N. Nidal, L. Karim, and T. Taleb, "Dynamic multilevel priority packet scheduling scheme for wireless sensor network," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1448–1459, Apr. 2013.

[17] R. E. Brown, "Impact of smart grid on distribution system design," in *Proc. IEEE Power Energy Soc. Gen. Meeting: Convers. Del. Elect. Energy Century*, Jul. 20–24, 2008, pp. 1–4.

[18] A. J. Conejo, J. M. Morales, and L. Baringo, "Real-time demand response model,"*IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 236–242, Dec. 2010.

[19] G. C. Buttazzo, "Rate monotonic vs. EDF: Judgment day," *Real Time Syst.*, vol. 29, no. 1, pp. 5–26, 2005.

[20] E. Bini, G. C. Buttazzo, and G. M. Buttazzo, "Rate monotonic analysis: The hyperbolic bound," *IEEE Trans. Comput.*, vol. 52, no. 7, pp. 933–942, Jul. 2003.

[21] G. C. Buttazzo, M. Bertogna, and G. Yao, "Limited preemptive scheduling for real-time systems. A survey," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 3–15, Feb. 2013.

[22] M. Yu, S. J. Xiahou, and X. Y. Li, "A survey of studying on task scheduling mechanism for TinyOS," in *Proc. Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Oct. 12–14, 2008, pp. 1–4.

[23] M. H. Yaghmaee and D. A. Adjeroh, "Priority-based rate control for service differentiation and congestion control in wireless multimedia sensor networks," *Comput. Netw.*, vol. 53, no. 11, pp. 1798–1811, 2009.

[24] P. Chennakesavula, J. Ebenezer, S. Murty, and T. Jayakumar, "Real-time packet scheduling for real-time wireless sensor networks," in *Proc. IEEE Int. Adv. Comput. Conf.*, Feb. 22–23, 2013, pp. 273–276.

[25] H. Yin, H. Qi, J. Xu, X. Huang, and A. He, "An efficient multitask scheduling model for wireless sensor networks," *J. Appl. Math.*, vol. 2014, 2014, Art. no. 969523.

[26] I. Takashima and M. Ikezaki, "An expanded spanning-tree protocol for home-oriented network management," *IEEE Trans Consum. Electron.*, vol. 37, no. 3, pp. 379–387, Aug. 1991.

[27] M. Zhang, Y. Lu, and C. Gong, "Energy-efficient routing protocol based on clustering and least spanning tree in wireless sensor networks," in *Proc. Int. Conf. Comput. Sci. Softw. Eng.*, Dec. 12–14, 2008, pp. 361–364.

[28] R. Lachowski, M. E. Pellenz, M. C. Penna, E. Jamhour, and R. D. Souza, "An efficient distributed algorithm for constructing spanning trees in wireless sensor networks," *Sensors*, vol. 15, no. 1, pp. 1518–1536, 2015.

[29] D. Estrin, L. Girod, G. Pottie, and M. Srivastava,"Instrumenting the world with wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 7–11, 2001, pp. 2033–2036.

Tie Qiu (M'12–SM'16) received the B.Sc. degree in mechatronics from Inner Mongolia University of Technology, Hohhot, China, and the M.Sc. and Ph.D. degrees in computer science from Dalian University of Technology (DUT), Dalian, China, in 2003, 2005, and 2012, respectively.

He is currently an Associate Professor with the School of Software, Dalian University of Technology. He was a visiting Professor with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA, from January 2014 to January 2015. He has authored/co-authored 8 books, over 60 scientific papers in international journals and conference proceedings. He has contributed to the development of 4 copyrighted software systems and invented 15 patents.

Dr. Qiu serves as an Associate Editor of IEEE ACCESS, *Computers and Electrical Engineering* (Elsevier journal), and *Human-Centric Computing and Information Sciences* (Springer Journal), an Editorial Board Member of *Ad Hoc Networks* (Elsevier journal) and the *International Journal on AdHoc Networking Systems*, a Guest Editor of *Future Generation Computer Systems* (Elsevier journal). He serves as the General Chair, PC Chair, Workshop Chair, Publicity Chair, Publication Chair or TPC Member of a number of conferences. He is a senior member of the China Computer Federation and the ACM.

Kaiyu Zheng (S'16) received the B.E. degree in software engineering from Dalian University of Technology (DUT), Dalian, China, in 2011, where he is working toward the master's degree with the School of Software.

He is an excellent graduate student of DUT and has been awarded several scholarships in academic excellence and technology innovation. His research interests include embedded systems and Internet of Things.

Mr. Zheng participated in "Open Source Hardware and Embedded Computing Contest 2012" and received the First Prize.

Houbing Song (M'12–SM'14) received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, USA, in August 2012.

In August 2017, he joined the Department of Electrical, Computer, Software, and Systems Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA, where he is currently an Assistant Professor and the Director of the Security and Optimization for Networked Globe Laboratory. He served as an Assistant Professor with the Department of Electrical and Computer Engineering, West Virginia University, Montgomery, WV, USA, and the Founding Director of West Virginia Center of Excellence for Cyber-Physical Systems sponsored by West Virginia Higher Education Policy Commission, from 2012 to 2017. In 2007, he was an Engineering Research Associate with the Texas A&M Transportation Institute. He is the Editor of four books, including *Smart Cities: Foundations, Principles and Applications* (Hoboken, NJ, USA: Wiley, 2017), *Security and Privacy in Cyber-Physical Systems: Foundations, Principles and Applications* (Chichester, U.K.: Wiley-IEEE Press, 2017), *Cyber-Physical Systems: Foundations, Principles and Applications* (Boston, MA, USA: Academic, 2016), and *Industrial Internet of Things: Cybermanufacturing Systems* (Cham, Switzerland: Springer, 2016). He has authored more than 100 articles. His research interests include cyber-physical systems, Internet of Things, cloud computing, big data analytics, connected vehicle, wireless communications and networking, and optical communications and networking.

Dr. Song is a member of the ACM. He received the Golden Bear Scholar Award, the highest faculty research award at West Virginia University Institute of Technology, in 2016.

Min Han (M'95–A'03–SM'06) received the B.S. and M.S. degrees in electrical engineering from Dalian University of Technology, Dalian, China, in 1982 and 1993, respectively, and the M.S. and Ph.D. degrees in electrical engineering from Kyushu University, Fukuoka, Japan, in 1996 and 1999, respectively.

Since 2003, she has been a Professor with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology. She has authored four books and more than 300 articles. Her current research interests include neural networks, chaos, and their applications to control and identification.

Burak Kantarci (S'05–M'09–SM'12) received the M.Sc. and Ph.D. degrees in computer engineering from Istanbul Technical University, Istanbul, Turkey, in 2005 and 2009, respectively.

From 2014 to 2016, he was an Assistant Professor with the Department of Electrical and Computer Engineering, Clarkson University, where he currently holds a courtesy appointment as an Assistant Professor. He was a Visiting Scholar with the University of Ottawa, where he completed the major content of his thesis. He is currently an Assistant Professor with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada. He has coauthored more than 100 papers in established journals and conferences and contributed to 11 book chapters.

Dr. Kantarci is a member of the ACM. He received the Siemens Excellence Award in 2005 for his studies in optical burst switching. He also serves as the Secretary of the IEEE ComSoc Communication Systems Integration and Modeling Technical Committee. He is an Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS and an Area Editor of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING.