EABS: An Event-Aware Backpressure Scheduling Scheme for Emergency Internet of Things

Tie Qiu, *Senior Member, IEEE*, Ruixuan Qiao, *Student Member, IEEE*,
and Dapeng Oliver Wu, *Fellow, IEEE*

Abstract—The backpressure scheduling scheme has been applied in Internet of Things, which can control the network congestion effectively and increase the network throughput. However, in large-scale Emergency Internet of Things (EIoT), emergency packets may exist because of the urgent events or situations. The traditional backpressure scheduling scheme will explore all the possible routes between the source and destination nodes that cause a superfluous long path for packets. Therefore, the end-to-end delay increases and the real-time performance of emergency packets cannot be guaranteed. To address this shortcoming, this paper proposes EABS, an event-aware backpressure scheduling scheme for EIoT. A backpressure queue model with emergency packets is first devised based on the analysis of the arrival process of different packets. Meanwhile, EABS combines the shortest path with backpressure scheme in the process of next-hop node selecting. The emergency packets are forwarded in the shortest path and avoid the network congestion according to the queue backlog difference. The extensive experiment results verify that EABS can reduce the average end-to-end delay and increase the average forwarding percentage. For the emergency packets, the real-time performance is guaranteed. Moreover, we compare EABS with two existing backpressure scheduling schemes, showing that EABS outperforms both of them.

Index Terms—Emergency internet of things, backpressure scheduling, event-awareness, network throughput

◆

1 INTRODUCTION

INTERNET of Things has been widely applied in many fields, such as industrial control [1], [2], cyber-physical systems [3], public safety device [4], environmental monitoring, military investigation [5], etc. As the scale of networks increases, a number of packets cause network congestion [6]. This problem becomes even more challenging in emergency situations where the packets need to be delivered to destination nodes in an appropriate time. Therefore, Emergency Internet of Things (EIoT) with high data load become a focus [7], [8]. There are different packets based on the type of events in EIoT. The sensor networks are an important component of EIoT. Most of the packets are regular packets. However, when the urgent events occur, some emergency packets are generated, which need to be delivered to destination node in a certain deadline.

There are many scheduling schemes for the sensor networks in EIoT, such as Collection Tree [9], ZigBee [10], etc. However, these schemes cannot guarantee the throughput of networks. Backpressure-based scheduling scheme is a distributed and adaptive scheme, which can effectively control the network congestion and guarantee the throughput of networks [11]. Furthermore, it can achieve dynamic adjustment in multi-hop networks by selecting the packet with the largest backlog difference among neighbor nodes. Backpressure scheduling scheme is also a robust method for time-varied network environments [12]. The deployment of the backpressure scheduling scheme is not affected by the packet arrival rate and the channel state. In recent years, lots of efforts have been proposed to improve the performance of the backpressure scheduling schemes in different networking environments [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23]. For the high-load data transmission, how to ensure the real-time performance of emergency packets brings challenges. Furthermore, backpressure scheduling scheme makes routing decisions only according to the queue backlog difference among neighbor nodes [17]. When the network load is low, it attempts to explore all the possible paths before the packets are delivered because there are not enough queue backlog difference gradients [20]. In addition, the backpressure scheduling scheme does not take the real-time performance of emergency packets into consideration. Poor real-time performance affects the deployment of backpressure scheduling scheme in the sensor networks of EIoT.

In this paper, we focus on the emergency internet of things with high data rate and propose an Event-Aware Backpressure Scheduling (EABS) scheme for EIoT. Our main contributions can be summarized as follows.

- We first propose a backpressure-based queue model with emergency packets according to the arrival process of different packets. The emergency packets will

---

- *T. Qiu and R. Qiao are with the School of Software, Dalian University of Technology, Dalian, Liaoning 116620, China. E-mail: qiutie@ieee.org, qiaoruixuan@mail.dlut.edu.cn.*
- *D. O. Wu is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611. E-mail: wu@ece.ufl.edu.*

Manuscript received 1 Oct. 2016; revised 9 Mar. 2017; accepted 4 May 2017. Date of publication 9 May 2017; date of current version 1 Dec. 2017. (Corresponding author: Tie Qiu.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2017.2702670

be prior forwarded, which reduces the waiting delay in queue for emergency event.

- Furthermore, the shortest path is combined with backpressure scheme for emergency packets, which can greatly reduce the forwarding times of emergency packets and avoid the network congestion. It greatly reduces the end-to-end delay and guarantees the real-time performance of emergency packets.
- We evaluate the performance of our proposed scheme in terms of end-to-end delay, forwarding percentage, and goodput. The experimental results demonstrate that our proposed scheme outperforms the existing schemes with respect to different performance metrics.

The rest of the paper is organized as follows. Section 2 discusses the related work and problem statement. In Section 3, our proposed event-aware backpressure scheduling scheme (EABS) for EIoT is presented. Section 4 elaborates the algorithm design. The simulation results of EABS, traditional backpressure scheduling (BS) scheme [11], and LIFO-based backpressure scheduling (LIFO-BS) scheme [24] are compared and analyzed in Section 5. Finally, we conclude the paper and present the future work in Section 6.

2 RELATED WORK AND PROBLEM STATEMENT

2.1 Related Work

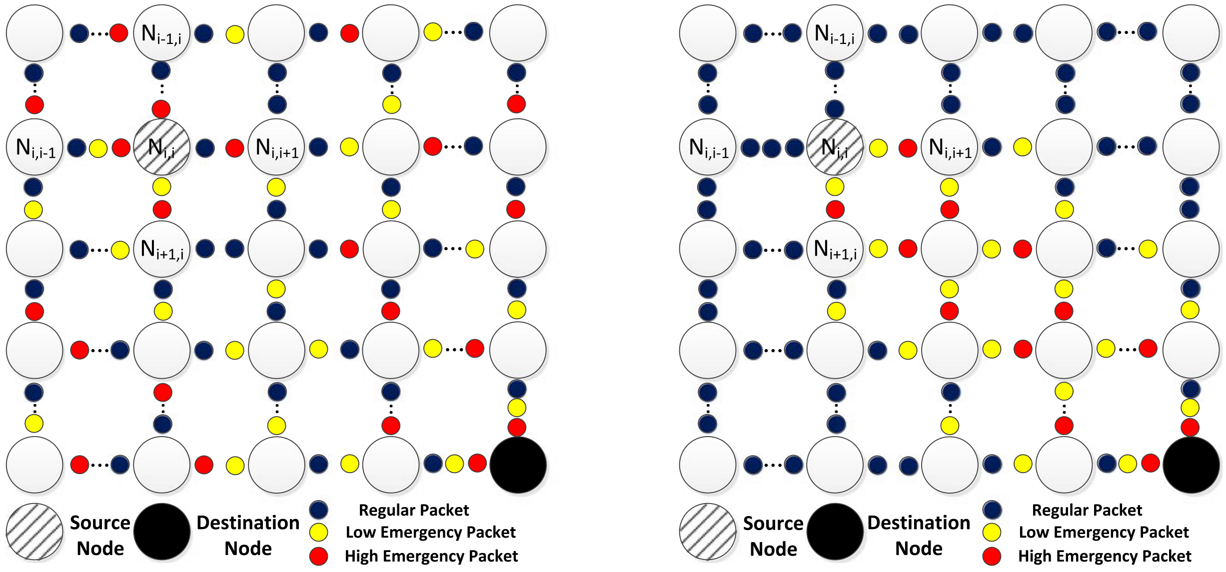
In this section, we briefly review recently proposed backpressure scheduling schemes. A traditional backpressure scheduling scheme is first proposed by Tassiulas and Ephremides [11] that is later extended to mobile networks including mobile ad-hoc networks [25], cellular networks [26], wireless sensor networks [14], energy harvest sensor networks [27], wireless multimedia sensor networks [28], and multi-hop wireless networks [29], [30].

In the past few years, backpressure scheduling schemes have attracted much attention in the field of sensor networks. Jiao et al. [31] propose a virtual queue-based method to solve the delay problem of backpressure scheduling scheme in wireless sensor networks. Hu et al. [32] propose the backpressure scheduling scheme based on the greedy algorithm and apply it to the smart grid sensor networks. The backpressure scheduling scheme is applied to wireless multimedia sensor networks, which solves the problem of congestion and dynamic priority [28]. In [33], a trust-based backpressure scheduling scheme for wireless sensor networks is proposed, which embeds trust factor into the link weights, so that the trusted links are scheduled to increase the throughput. Sridharan et al. [13] investigate the rate control protocol for wireless sensor networks based on the backpressure scheduling scheme. A distributed gradient aided backpressure framework for WSNs is proposed, and the method is proved to optimize the energy efficiency and throughput [34].

It should be noted that the backpressure scheduling scheme is different from the path-based scheme. Generally, the path-based scheme has no loops, and the packets are transmitted through a shorter path. This kind of routing is attractive, but the network throughput is reduced. The improvement for backpressure scheduling scheme must be in the premise of not losing the throughput. Recently, numerous improved schemes have been proposed to reduce the delay of the backpressure scheduling.

The majority of existing backpressure scheduling schemes focus on avoiding unnecessary long path and loop problems. This kind of problems is caused by calculating the weight only according to queue backlog difference. A routing loop penalty factor is introduced in the process of making routing decisions [9]. Similarly, Dvir et al. [18] use the same idea but the factors are different. The packets are guaranteed to be transmitted in a restricted area by the explicit hop-count constraint [35]. Furthermore, Ying et al. [36], [37] use the length of the shortest path as a hop constraint to route and schedule packets, shortening the path of backpressure scheduling scheme. Lu et al. [38] employ a calculation method with the weighted queue in wireless networks to effectively reduce the forwarding delay of packets. A simple data collecting scheme is proposed to increase the network throughput while the average energy consumption converges to the energy consumption that required maintaining the stability of the network [17]. For the last packets problem, the link weights are assigned according to the packet delay [39]. Alresaini et al. [40] propose a redundancy backpressure scheduling scheme in which the data packets are copied when the queue occupancy rate is low. It can increase the queue backlog difference in advance, reducing the end-to-end delay at low network load. However, this method is not suitable for sensor networks, because the redundancy packets need to be transmitted which wastes network energy. A virtual queue-based method is proposed [31]. It establishes the backlog of queues in advance, so that the network can generate sufficient backlog difference gradients when the network load is low. However, the schemes mentioned above cannot give the priority to the emergency packets when urgent events occur.

There are also some improvements in the aspect of queue modeling. The per-neighbor queue is used to replace the traditional backpressure queue model, which significantly reduces the end-to-end delay [20], [41]. Ying et al. [42] propose a backpressure scheduling scheme based on clustering in which each node maintains two types of queues, the inter-cluster queues and intra-cluster queues. It can greatly reduce the number of queues by this method. The traditional backpressure scheduling scheme based on First-In-First-Out (FIFO) queue is changed to Last-In-First-Out (LIFO) queue which reduces the end-to-end delay [14]. It makes the backpressure scheduling apply to wireless sensor networks. Furthermore, Moeller et al. [9] present an application of backpressure scheduling scheme in wireless sensor networks, which can achieve greater delivery ratio and throughput than the Collection Tree Protocol (CTP). Huang et al. [24] analyze the tradeoff between LIFO and FIFO queue models in detail, and get the conclusion that the average energy consumption of backpressure scheduling scheme will gradually converge. However, these related work doesn't propose an efficient queueing model to deal with emergency packets for EIoT. The inefficient queueing model for emergency packets affects the deployment of backpressure scheduling scheme in EIoT. There are also works aiming at detect the emergency situations. Aazam et al. work on fog computing and cloud computing for tackling emergency communication [43]. In our work, we focus



(a) (b)

Fig. 1. A backpressure scheduling scheme with emergency packets.

on the queueing and processing the emergency packets under backpressure scheduling scheme.

2.2 Problem Statement

Different types of packets can exist in a network. Some emergency packets are generated because of the happening of emergency events in EIoT, and they need to be delivered as fast as possible. As shown in Fig. 1, there is one flow in the network which consists of emergency packets and regular packets.

The LIFO backpressure scheduling scheme aims to change the traditional FIFO queue to LIFO. However, it makes routing decisions based on the queue backlog difference. There is not enough queue backlog difference under the low network load, resulting in the long path even the loop path problem. We consider the sensor network as shown in Fig. 1a. There are some packets being forwarded from the source node $N_{i,i}$ to node $N_{i-1,i}$ and $N_{i,i-1}$, including emergency packets and regular packets. The packets are forwarded far from the destination node. The forwarding number of packets increases so does the end-to-end delay. In particular, the scheduling scheme for the emergency packets is same as regular packets under LIFO backpressure scheduling scheme, resulting in the poor real-time performance of emergency packets.

We can select the shortest path scheduling scheme to forward the emergency packets when the network is not congested. We expect the emergency packets are delivered to the destination node as fast as possible. The expected forwarding process is shown in Fig. 1b. The emergency packets are forwarded from the source node $N_{i,i}$ to nodes $N_{i+1,i}$ and $N_{i,i+1}$, which is the shortest path to the destination node. However, the long path even the path with loops will still be possibly selected under the traditional backpressure scheduling scheme. Therefore, the lower the network load, the more loops are selected, causing the increase of end-to-end delay. Therefore, the real-time performance of emergency packets cannot be well guaranteed. Thus, we focus on the emergency situations under the backpressure scheduling scheme. In this paper, we explore an event-aware backpressure scheduling scheme for EIoT. The queueing model with emergency packets is proposed to reduce the waiting delay. Furthermore, we combine the shortest path with backpressure scheduling scheme to guarantee the real-time performance of emergency packets.

3 OUR PROPOSED EVENT-AWARE BACKPRESSURE SCHEDULING SCHEME

3.1 Scheduling Scheme for Regular Packets

In order to describe the scheme, symbols and notations are shown in Table 1.

If $D_{m,n} \leq R$, $(m,n)\epsilon E(G)$ for each pair of nodes m and n. The packet can be generated by any sensor node, and

TABLE 1
Symbols Definition

Symbols	Description
m, n	The sensor nodes No.
f	The queue No.
V	The set of nodes in the network
E	The set of links in the network
$G = (V, E)$	The sensor network model
R	The transmission range
$D_{m,n}$	The geographic distance of node m and node n
F	The set of queues
λ	The arrival rate (accord with Poisson process)
t	The time slot
$Q_n^f(t)$	The backlog of queue f in node n at time slot t
$Reg_n^f(t)$	The arrival regular packets of queue f to the network through node n at time slot t
$Emg_n^f(t)$	The arrival emergency packets of queue f to the network through node n at time slot t
$\mu_{m,n}^f(t)$	The transmission rate of queue f over link (m, n) at time slot t
$\mu = \{\mu_{m,n}\}$	The set of link rates in the network
$H_n^f(t)$	The hop-count between node n and the destination node of queue f
V_m	The set of the neighbor nodes of node m

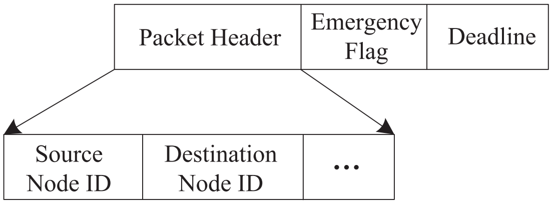


Fig. 2. The data packet format.

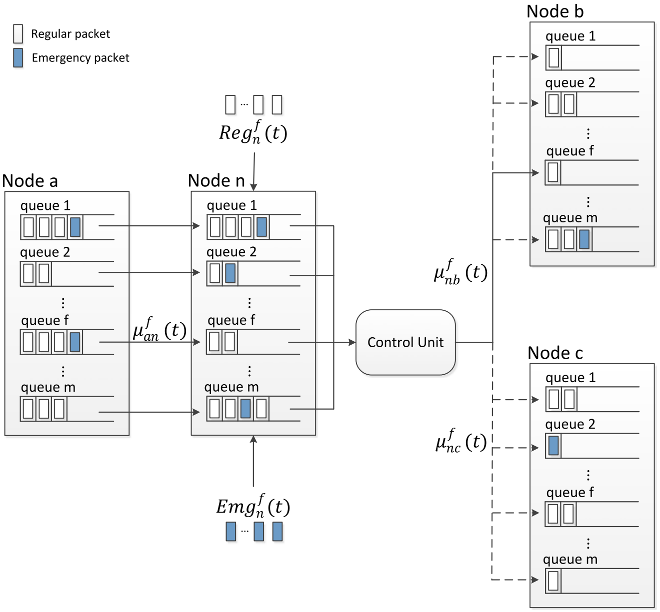


Fig. 3. The dynamic queue model with emergency packets.

finally, reach the destination node through the multi-hop path. The packets have the emergency flag when generate. The data packet format is defined in Fig. 2.

Each data packet includes the packet header, emergency flag and packet deadline. The packet header length will be different based on the different communication protocols. But it always includes the source node ID and the destination node ID. The emergency flag and the packet deadline describe the packet emergency information. If the emergency flag is true, then the packet is defined as the emergency packet. The nodes can detect whether the arriving packets are emergency packets or regular packets according to the emergency flag. The dynamic queue model maintains by each node for scheduling is shown in Fig. 3.

The backlog of queue f at time slot $t+1$ can be expressed by

$$\begin{aligned}Q_n^f(t+1) =& max\left[Q_n^f(t) - \sum_b \mu_{nb}^f(t), 0\right] + \sum_a \mu_{an}^f(t) \\ &+ Reg_n^f(t) + Emg_n^f(t).\end{aligned} \tag{1}$$

The queue backlog Q_n^f at time slot $t+1$ includes the external arriving packets and the difference between $Q_n^f(t)$ and $\sum_b \mu_{nb}^f(t)$, where $\mu_{nb}^f(t)$ is the transmission rate of queue f over link (n, b) at time slot t. We use the $\sum_b \mu_{nb}^f(t)$ to stand for the packets being forwarded from node n at time slot t. The external arriving packets consist of the arriving packets $\sum_a \mu_{an}^f(t)$, the arriving regular packets $Reg_n^f(t)$, and emergency packets $Emg_n^f(t)$, where $\mu_{an}^f(t)$ is the transmission rate of queue f over link (a, n) at time slot t. We use the $\sum_a \mu_{an}^f(t)$ to stand for the packets arriving at node n at time slot t. Packets are forwarded according to the LIFO scheduling policy. We attempt to improve the backpressure scheduling scheme in the steps of making routing decisions, which does not affect the link transmitting process. Thus, our scheme can be directly extended to transmission rates time-varying networks. To simplify our expression, we assume that the transmission rates are fixed. Node m calculates the backlog of each queue with each neighbor node n every time slot according to Eq. (2). Link (m, n) is used to transmit packets with the largest queue backlog difference

$$d_{m,n}^f = Q_m^f(t) - Q_n^f(t). \tag{2}$$

Making routing decisions only based on the queue backlog can cause the packets to be forwarded far from the destination node, which has a adverse effect on the delay performance of regular packets. We make the improvements in the step of selecting data packets and calculating weights for backpressure scheduling scheme on the basis of the LIFO queue. We define the selected queue $f_{m,n}^{sel}$ in Eq. (3) and forward the packet in queue $f_{m,n}^{sel}$ at time slot t

$$f_{m,n}^{sel} \triangleq argmax\{Q_m^f(t) - ((H_n^f - H_m^f)/H_m^f)Q_n^f(t)\}. \tag{3}$$

We use $(H_n^f - H_m^f)/H_m^f$ as a weight while calculating the queue backlog difference of node m and node n. If $H_n^f < H_m^f$, then the weight $(H_n^f - H_m^f)/H_m^f$ will be negative. The weighted queue backlog difference is larger than $Q_m^f(t)$. If $H_n^f > H_m^f$, then the weight $(H_n^f - H_m^f)/H_m^f$ will be positive. The weighted queue backlog difference is less than $Q_m^f(t)$. If $H_n^f = H_m^f$, then the weight $(H_n^f - H_m^f)/H_m^f$ will be zero. The weighted queue backlog difference is equal to $Q_m^f(t)$. After we use the weight, if the node n is closer to the destination node, then the weighted queue backlog difference will be larger than the one under LIFO backpressure scheduling scheme. The closer node n to the destination node, the larger the weighted queue backlog difference is.

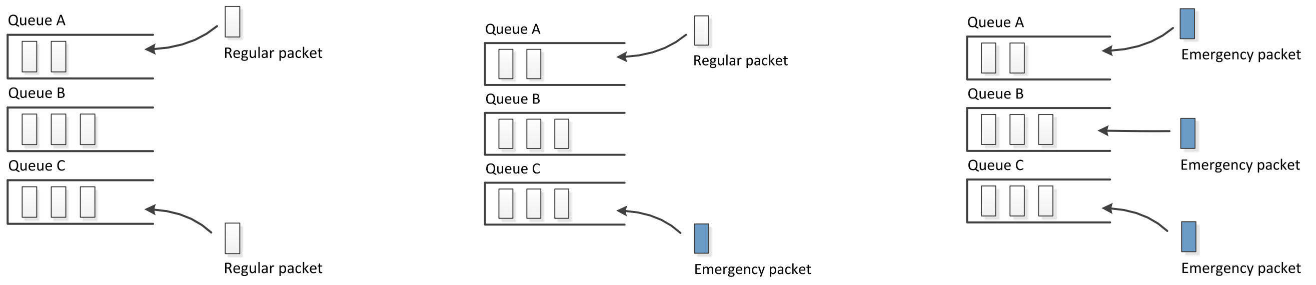
After using the distance weight, the probability of node closer to the destination node increases. Then, the packets have more probability to be forwarded to the closer node to destination node even if there are not enough queue backlog difference gradients. Distance weight acts as another pressure to the destination node. After selecting the forwarding queues, the distance weight w can be expressed by

$$w = (H_n^{f_{m,n}^{sel}} - H_m^{f_{m,n}^{sel}})/H_m^{f_{m,n}^{sel}}. \tag{4}$$

The weighted queue backlog difference $W_{m,n}(t)$ can be expressed by Eq. (5), where the weight w has been calculated in Eq. (4)

$$W_{m,n}(t) \triangleq \max\left\{\left(Q_m^{f_{m,n}^{sel}}(t) - w * Q_n^{f_{m,n}^{sel}}(t)\right), 0\right\}. \tag{5}$$

We select the neighbor node with the maximum distance weighted queue backlog difference $W_{m,n}(t)$ to forward the packet in queue $f_{m,n}^{sel}$ in each time slot t. When calculating the link weights, the distance factor is introduced, which is the number of hop-count from the neighbor nodes to the destination node. We can calculate the hop-count according to the Dijkstra algorithm after the whole sensor network



(a) no emergency packets arriving (b) one emergency packet arriving (c) more than one emergency packet arriving

Fig. 4. Different packets arriving cases.

has been deployed. Each node will obtain their minimum hop-count to every other node by the global broadcast once. We assume that all nodes in the networks are not mobile, and the connections are stable; then the factor remains unchanged. Further, each node can broadcast the neighbor node list each a period time slot. If the distance (hop-counts) changes, the neighbor node list will update. Thus, the result can be easily extended to the mobile sensor networks.

3.2 Scheduling Scheme for Emergency Packets

There are certain emergency packets in the network which need to be delivered to the destination node in a limited time. For example, when a fire occurs in a monitoring sensor network of EIoT, the packets carrying the temperature data need to be delivered as fast as possible. Generally, the number of emergency packets is small. The distance weighted method we mentioned in the previous section can reduce the end-to-end delay of the packets. However, it cannot guarantee the real-time performance of emergency packets.

We consider the different packets can arrive cases under the backpressure scheduling scheme as shown in Fig. 4.

Case 1. there is no emergency packets arriving at node n, as shown in Fig. 4a.

In this case, the regular packets should be selected and forwarded according to Eqs. (3), (4), and (5).

Case 2. there are emergency packets arriving at node n.

(i) There is only one emergency packet arriving at node n, as shown in Fig. 4b.

In this case, we should give the priority to the emergency packet and forward it to the node which is the closest to the destination node.

(ii) There is more than one emergency packet arriving at node n, as shown in Fig. 4c.

In this case, we should give the priority to the emergency packet with a minimum deadline, and forward it to the node which is the closest to the destination node.

However, the emergency packets are likely to be selected after the regular packets under Case 2, as shown in Fig. 5.

Emergency packets arrive at the Queue A and Queue C of node n at time slot $t = i$, as shown in Fig. 5a. We assume that the emergency packet in Queue C is forwarded at time slot $t = i + 1$. At the same time, there is one regular packet arriving at Queue A of node n, as shown in Fig. 5b. Then, the emergency packet in Queue A will always be forwarded after the regular packet, according to the LIFO queueing model shown in Fig. 5c. It cannot guarantee the real-time performance of the emergency packets. To deal with this problem, we propose the queue model with emergency packets. We design an emergency queue in each node in the network, as shown in Fig. 6.

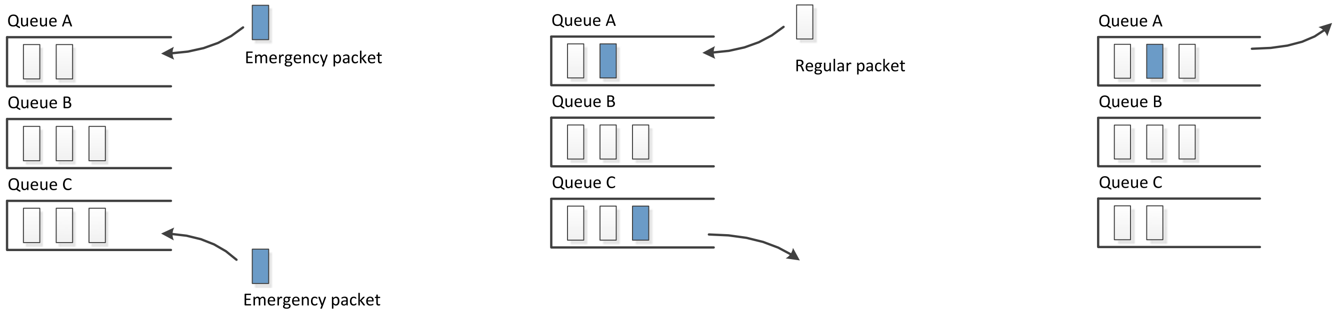
The emergency packets will arrive at the emergency queue, and the packets with the minimum deadline will be prior selected. The regular packets will arrive at the queues according to their destination nodes. Packets are forwarded according to the LIFO scheduling policy.

We make the improvements in the step of selecting data packets and calculating weights for backpressure scheduling scheme on the basis of the LIFO queue. In EABS, we define the selected queue $f^{sel}_{m,n}$ given by

$$f^{sel}_{m,n} \triangleq \begin{cases} argmax\{Q^f_m(t) - wQ^f_n(t)\} & \left(Q^{emg}_n(t) = 0\right) \\ emg & \left(Q^{emg}_n(t) \neq 0\right) \end{cases} \tag{6}$$

$$w = (H^f_n - H^f_m)/H^f_m \tag{7}$$

When the emergency queue is empty, the queue with the maximum backlog difference will be selected as the



(a) $t = i$ (b) $t = i + 1$ (c) $t = i + 2$

Fig. 5. More than one emergency packet arriving at node n.

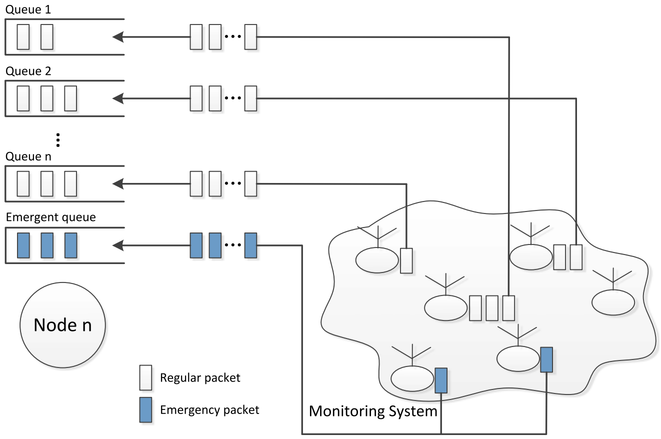


Fig. 6. The queue model with emergency packets.

forwarding queue. If there are emergency packets in the emergency queue, then the emergency packet with the minimum deadline will be prior selected.

Then we select the neighbor node to forward the packet in each time slot t after selecting the forwarding queues. It will be better to select the node which is the closest to the destination node of the emergency packets. However, when the number of emergency packets increases, the congestion can occur for emergency packets. We consider this problem by combining the backpressure scheduling scheme with the shortest path scheme. Thus, we define the next-hop nodes as follows:

1) The regular packets are selected ($f^{sel}_{m,n} \neq emg$).

We select the next-hop node n with the maximum $W_{m,n}(t)$ according to Eqs. (4) and (5).

2) The emergency packets are selected ($f^{sel}_{m,n} = emg$).

We find node n which is the closest node to the destination node of the selected emergency packet. Then, we first calculate the emergency queue backlog difference according to

$$d^{emg}_{m,n}(t) = Q^{emg}_{m}(t) - Q^{emg}_{n}(t). \tag{8}$$

If $d^{emg}_{m,n}(t) \geq 0$, then the emergency packet will be forwarded to node n.

If $d^{emg}_{m,n}(t) < 0$, then we calculate the forwarding weight according to Eqs. (9) and (10).

$$W_{m,n}(t) \triangleq \max\{(Q^{emg}_{m}(t) - wQ^{emg}_{n}(t)), 0\}, \tag{9}$$

where, w is the distance weight expressed by

$$w = (H^{emg}_{n} - H^{emg}_{m})/H^{emg}_{m}. \tag{10}$$

The node with the maximum forwarding weight $W_{m,n}(t)$ will be selected as the next-hop node.

All nodes in the whole networks broadcast their neighbor nodes list. Each node calculates its shortest path with all other nodes at the first time slot ($t = 0$). Next, at the beginning of each time slot, node m observes the queue backlog $Q^{f}_{n}(t)$ and the hop-count $H^{f}_{n}(t)$ of every neighbor node n. The EABS scheme consists of the routing decision and forwarding decision, which are described in details as follows.

Routing Decision:

(i) Calculate the queue backlog difference of each neighbor node n according to Eq. (6). The packet in queue $f^{sel}_{m,n}$ is selected waiting to be forwarded.

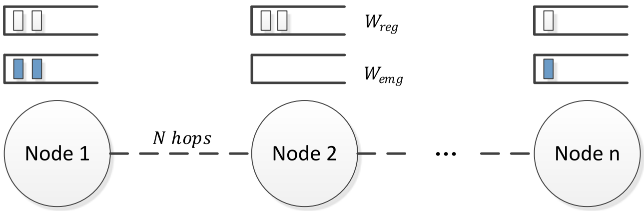


Fig. 7. No congestion of emergency packets on the shortest path.

(ii) If the regular packets are selected, the node with maximum $W_{m,n}(t)$ will be selected as the next-hop node according to Eqs. (4) and (5). If the emergency packets are selected and $d^{emg}_{m,n}(t) \geq 0$, the emergency packets will be forwarded to the node on the shortest path. If $d^{emg}_{m,n}(t) < 0$, the emergency packets will be forwarded to the next-hop node with the maximum $W_{m,n}(t)$ according to Eqs. (9) and (10).

Forwarding Decision. If $W_{m,n}(t) > 0$, forward as many packets as possible in the capacity of the link (m, n) under LIFO queue. Otherwise, the packets are not forwarded until the factor $W_{m,n}(t)$ is recalculated. Finally, the queue backlog $Q^{f}_{n}(t+1)$ is updated.

3.3 Real-Time Performance Analysis

The network delay D can be express by

$$D = \frac{1}{x}\sum_{i=1}^{x} T_i + W_{emg} + W_{reg}. \tag{11}$$

W_{emg} is the time of waiting emergency packets, and W_{reg} is the time of waiting regular packets in the path. T_i is the transmission delay of packet i. We assume that the number of regular packets equals to the number of emergency packets, denoted as x. There is only one flow in the network where the source node is Node 1 and the destination node is Node n. We set n for the hop-count of the shortest path. N is the hop-count of the sub-optimal path, and N' is the hop-count of another path. We assume $N \geq N'$, because the sub-optimal path is selected according to the distance weight.

1) When $d^{emg}_{m,n}(t) \geq 0$, the emergency packets on the shortest path are not congested. This case is shown in Fig. 7.

In this case, the emergency packets will be forwarded on the shortest path to the destination node. The end-to-end delay of emergency packets can be represented by

$$D_{emg} = \frac{1}{x}[n + (n+1) + (n+2) + \cdots + (n+x-1)] * t + W_{emg}. \tag{12}$$

If the regular packets take the shortest path, we may get the minimum delay. The end-to-end delay of regular packets can be represented by

$$\begin{aligned} D_{reg} = &\frac{1}{x}[n + (n+1) + (n+2) + \cdots + (n+x-1)] * t \\ &+ x * t + W_{emg} + W_{reg}. \end{aligned} \tag{13}$$

The end-to-end delay of regular packets is larger than that of emergency packets. The difference of D_{reg} and D_{emg} is $x * t + W_{reg}$ (the delay waiting for the emergency packets and the delay of waiting for other regular packets).

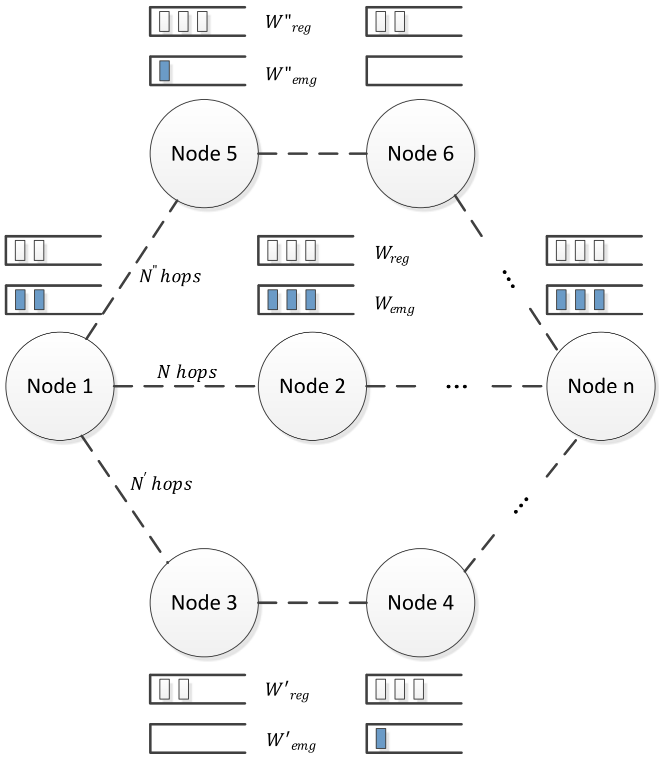


Fig. 8. Congestion of emergency packets on the shortest path.

2) When $d^{emg}_{m,n}(t) < 0$, the emergency packets on the shortest path are congested. This case is shown in Fig. 8.

In this case, the distance weighted queue backlog difference will be calculated, and the emergency packets will be forwarded on the sub-optimal path to the destination node (i.e., the path through Node 3). The hop-count of the sub-optimal path is N, and the waiting time of emergency packets is $W^{'}_{emg}$. We put these values into Eq. (11). The end-to-end delay of emergency packets is given by

$$D_{emg} = \frac{1}{x}[N + (N+1) + (N+2) + \cdots + (N+x-1)] * t + W^{'}_{emg}. \tag{14}$$

For the regular packets, there are three cases to be analyzed:

(i) The shortest path (i.e., the path through Node 2):

The emergency packets are congested on the shortest path. If the regular packets are forwarded on the shortest path, the end-to-end delay is obviously larger than that of emergency packets.

(ii) Another path (i.e., the path through Node 5):

The end-to-end delay of regular packets can be represented by

$$\begin{aligned} D_{reg} =& \frac{1}{x}[N' + (N'+1) + (N'+2) + \cdots + (N'+x-1)] * t \\ &+ x * t + W^{''}_{emg} + W^{''}_{reg} \end{aligned} \tag{15}$$

$$\begin{aligned} D_{reg} - D_{emg} &= (N' - N) * x * t + x * t + (W^{''}_{emg} - W^{'}_{emg}) + W^{''}_{reg} \\ &= (N' - N + 1) * x * t + (W^{''}_{emg} - W^{'}_{emg}) + W^{''}_{reg}. \end{aligned} \tag{16}$$

Because the emergency packets choose the sub-optimal path which is calculated according to the distance weight, so we can get $N \leq N'$, $W^{'}_{emg} \leq W^{''}_{emg}$. Thus, $D_{reg} - D_{emg} \geq 0$.

(iii) The same sub-optimal path as the emergency packets (i.e., the path through Node 3):

The end-to-end delay of regular packets can be calculated by

$$\begin{aligned} D_{reg} =& \frac{1}{x}[N + (N+1) + (N+2) + \cdots + (N+x-1)] * t \\ &+ x * t + W^{'}_{emg} + W^{'}_{reg}. \end{aligned} \tag{17}$$

The end-to-end delay of regular packets is larger than that of emergency packets according to the analysis of no congestion of emergency packets on the shortest path.

4 ALGORITHM DESIGN

In this section, we implement EABS according to the model proposed in Section 3. The variables used in the algorithms are defined in Table 1. We assume that all nodes in the networks are not mobile, and the connections are stable, then the factor remains unchanged. We also assume that the transmission rates are fixed.

Algorithm 1 describes the process of selecting packets in EABS. There are two loops in Algorithm 1. The upper bond of neighbor nodes is $n-1$, and there are n queues in each node. Thus, the complexity of Algorithm 1 is $\mathcal{O}(n^2)$, same as the packets selecting process of LIFO backpressure scheduling scheme. The algorithm works as follows.

Algorithm 1. Queues Selecting

```
Input: $G$, $F$, $Q^f_n(t)$
Output: $f^{sel}_{m,n}$
 1: procedure QUEUES-SEL($G$, $F$, $Q^f_n(t)$)
 2:   if $Q^{emg}_m(t) \neq 0$ then
 3:     $f^{sel}_{m,n} \leftarrow emg$
 4:   else
 5:     for all $n \in V_m$ do
 6:       for all $f \in F$ do
 7:         Calculate the difference $d^f_{m,n}$ using
 8:         $d^f_{m,n}(t) = Q^f_m(t) - ((H^f_n - H^f_m)/H^f_m)Q^f_n(t)$
 9:         if $d^f_{m,n}(t)$ is the maximum weighted queue
            difference then
10:           $f^{sel}_{m,n} \leftarrow f$
11:         end if
12:       end for
13:     end for
14:   end if
15: end procedure
```

At each time slot, node m observes the emergency queue. If the emergency queue is not empty, the packets in the emergency queue will be selected (Lines 2-3). If the emergency queue is empty, for each neighbor node n and each queue of node m, the distance weighted queue backlog difference will be calculated (Lines 5-8). Node m chooses the queue with the maximum weighted difference as the forwarding queue (Lines 9-11).

Algorithm 2 describes the process of the next-hop node selecting of EABS. There is only one loop in Algorithm 2. Thus, the complexity of the Algorithm 2 is $\mathcal{O}(n)$, same as the next-hop node selecting process of LIFO backpressure scheduling scheme. The algorithm works as follows.

Algorithm 2. Next-Hop Nodes Selecting

```
Input: $G$, $F$, $\mu$, $Q_n^f(t)$, $f_{m,n}^{sel}$
Output: next-hop nodes ($N_m^{sel}$)
 1: procedure $\mathrm{NODES-SEL}(G, F, Q_n^f(t), f_{m,n}^{sel})$
 2:     if $f_{m,n}^{sel} = emg$ then
 3:         for all $n \in V_m$ do
 4:             Find node n with the shortest path of the top
                packet in queue emg
 5:         end for
 6:         Calculate the difference $d_{m,n}^f$ using
 7:         $d_{m,n}^{emg}(t) = Q_m^{emg}(t) - Q_n^{emg}(t)$
 8:         if $d_{m,n}^{emg}(t) \geq 0$ then
 9:             $N_m^{sel} \leftarrow n$
10:         else
11:             for all $n \in V_m$ do
12:                 Calculate the difference $d_{m,n}^{emg}$ using
13:                 $d_{m,n}^{emg}(t) = Q_m^{emg}(t) - ((H_n^{emg} - H_m^{emg})/H_m^{emg})Q_n^{emg}(t)$
14:                 if $d_{m,n}^{emg}(t)$ is the maximum weighted queue
        difference then
15:                     $N_m^{sel} \leftarrow n$
16:                 end if
17:             end for
18:         end if
19:     else
20:         for all $n \in V_m$ do
21:             Calculate the difference $d_{m,n}^f$ using
22:             $d_{m,n}^f(t) = Q_m^f(t) - ((H_n^f - H_m^f)/H_m^f)Q_n^f(t)$
23:             if $d_{m,n}^f(t)$ is the maximum weighted queue
        difference then
24:                 $N_m^{sel} \leftarrow n$
25:             end if
26:         end for
27:     end if
28: end procedure
```

At each time slot, node m observes whether the selected queue $f_{m,n}^{sel}$ is the emergency queue. If $f_{m,n}^{sel}$ is the emergency queue, node m finds the node n which is the closest to the destination node of emergency packet (Lines 3-5). Then, node m calculates the emergency queue backlog difference $d_{m,n}^{emg}$ with node n (Lines 6-7). If $d_{m,n}^{emg} \geq 0$, node n will be selected as the next-hop node (Lines 8-9). If $d_{m,n}^{emg} < 0$, node m calculates the distance weighted emergency queue backlog difference for each neighbor node n (Lines 10-18). If $f_{m,n}^{sel}$ is the regular queue, node m chooses the node n with the maximum distance weighted queue backlog difference as the next-hop node (Lines 19-27).

Algorithm 3 describes the EABS scheme. It operates in each time slot. It calls Algorithm 1 and Algorithm 2 in turn. Thus, the complexity of Algorithm 3 is $\mathcal{O}(n^3)$, same as the LIFO backpressure scheduling scheme. The algorithm works as follows.

All nodes in the whole networks broadcast their neighbor nodes lists. Each node calculates its shortest path with all other nodes at the first time slot (Lines 3-6). At each time slot, node m first obtains the forwarding queue according to Algorithm 1. Then, it selects the next-hop node according to Algorithm 2. At last, node m forwards the packets in queue $f_{m,n}^{sel}$ to the next-hop node N_m^{sel}.

Algorithm 3. Event-Aware Backpressure Scheduling Scheme

```
Input: $G$, $f_{m,n}^{sel}$, $N_m^{sel}$
Output: scheduling scheme
 1: procedure $\mathrm{EABS}(G, f_{m,n}^{sel}, N_m^{sel})$
 2:     At each time slot $t$
 3:     if time slot $t = 0$ then
 4:         Broadcast the neighbor node list
 5:         Calculate the shortest path of every other node
 6:     end if
 7:     for all $m \in V$ do
 8:         Call Algorithm 1 to get the $f_{m,n}^{sel}$
 9:         Call Algorithm 2 to get the $N_m^{sel}$
10:         Node $m$ forwards the packets in queue $f_{m,n}^{sel}$ to the
      next-hop node $N_m^{sel}$
11:     end for
12: end procedure
```

5 SIMULATION AND ANALYSIS

In this Section, we use simulation experiments to verify the proposed scheme and evaluate the performance. The simulation is implemented on NS-2 simulation platform.

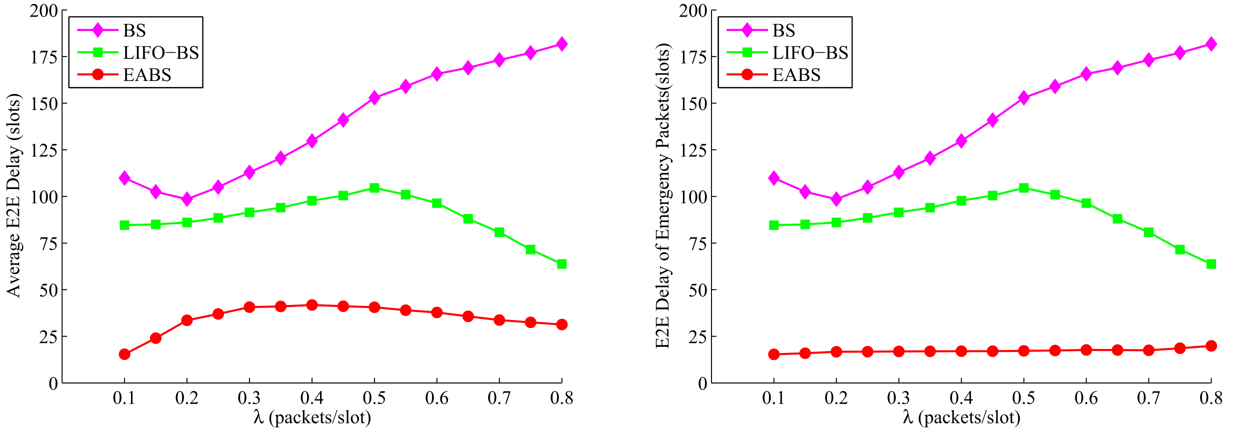
5.1 Simulation Setup

We deploy a random network topology with 100 nodes in a 2200 m $*$ 2200 m area. The transmission range of each sensor nodes is 250 m. The transmission capability is one packet in each direction for each time slot, and all the links are bidirectional. We randomly create 20 data flows in the network, and 5 percent of packets are emergency packets. The packet arrival rate of all network flows follows Poisson process, and all data flows have the same packet arrival rate, donated by λ (packets/slot). The arrival rate λ varies from 0.1 packets/slot to 0.8 packets/slot. We choose these parameters because the results are able to verify our analyzation of the traditional backpressure scheduling scheme. The end-to-end delay will first increase then decrease when the arrival rate varies from 0.1 packets/slot to 0.8 packets/slot.

We set the initial energy of each node to 100 J and the transmit power is set to 0.690 W. Furthermore, the received power is set to 0.395 W. The energy is mainly consumed in the process of data forwarding and receiving. We use the time of first node running out of energy as the evaluation metric for network lifetime. We set the data packet arrival rate λ (packets/slot) as the horizontal axis in the simulation. We change the value of λ to observe the performance of the proposed scheme under different network loads. The simulation experiments are implemented in 500 time slots, each of which is implemented for 10 iterations to avoid the random errors. We use AVG for the average metric of all packets, EMG for the emergency packets, and REG for the regular packets.

5.2 End-to-End Delay

Fig. 9 shows the average end-to-end delay, where "BS" stands for the traditional backpressure scheduling scheme, "LIFO-BS" stands for the LIFO backpressure scheduling



(a) Average end-to-end delay

(b) End-to-end delay of emergency packets

Fig. 9. End-to-end delay.

scheme, and "EABS" stands for our proposed event-aware backpressure scheduling scheme. It is the average time from the packet generation to delivery in the network. The emergency packets need to be delivered as fast as possible. Thus, the end-to-end delay is an important metric to evaluate the real-time performance of the scheduling scheme. It can be seen that with the increase of the packet arrival rate, the end-to-end delay of BS scheme first decreases then increases. The reason is that the BS cannot generate enough pressure at the low network load because there is not enough queue backlog difference gradient. BS scheme will explore all possible paths, so the end-to-end delay increases. When the network load increases, the end-to-end delay of BS scheme reduces because the queue backlog difference gradient is formed. Subsequently, the queue backlog starts to increase, which leads to the increase of end-to-end delay. LIFO-BS scheme improves the BS scheme by changing the queue from FIFO to LIFO. With the formation of the queue backlog gradient, the delay reduces according to the LIFO-BS scheme when the network load increases.

The average end-to-end delay of all packets is shown in Fig. 9a. The EABS scheme increases the weight of node which is closer to the destination node on the basis of LIFO-BS. The distance factors are weighted so that the data packets can still be forwarded to the destination node when the queue backlog difference gradient is not formed. The average end-to-end delay reduces with the reducing of hop-count. The average end-to-end delay of emergency packets is shown in Fig. 9b. Under the BS and LIFO-BS scheduling schemes, the emergency packets are not considered, the end-to-end delay of emergency packets is same as that of the regular packets. However, the emergency packets are given a high priority to be forwarded to the node which is closer to the destination node under the EABS scheme, so the end-to-end delay is minimized, which ensures the real-time performance of the emergency packets.

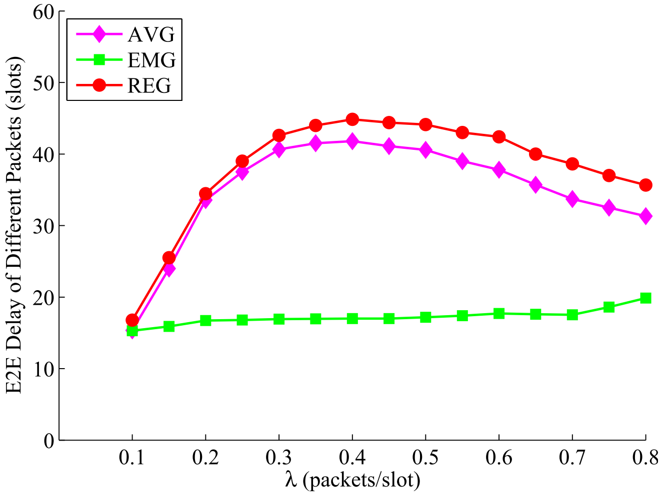


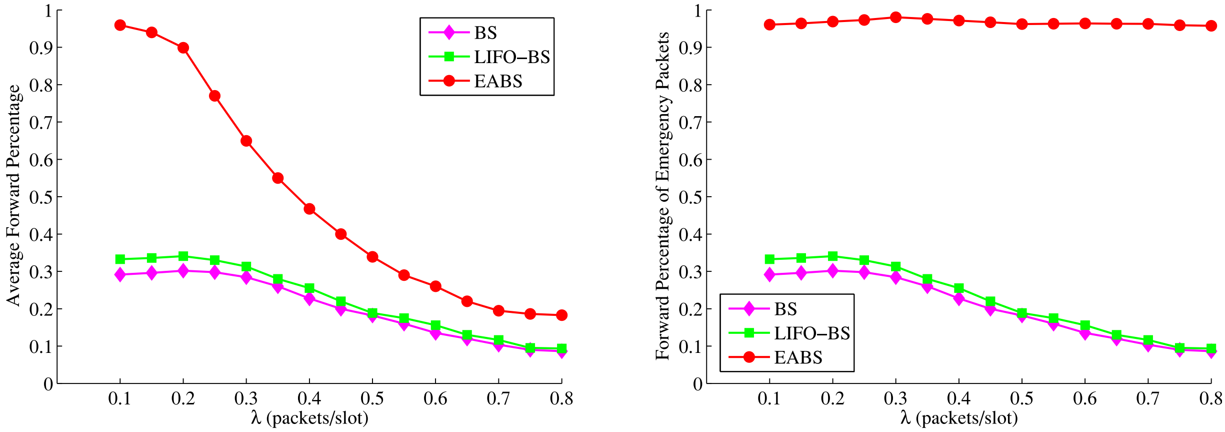
Fig. 10. End-to-end delay of different packets.

We can see from Fig. 10 that the average end-to-end delay is far lower than that of the regular packets. The end-to-end delay of regular packets is larger than the average level. However, in EIoT, the real-time performance of emergency packets is more important. Further, the end-to-end delay of regular packets is lower than that of LIFO-BS scheme and BS scheme. Meanwhile, the proportion of delivered emergency packets increases compared with the regular packets. Thus, the gap of end-to-end delay between REG and AVG increases.

5.3 Forwarding Percentage

We illustrate the forwarding percentage of BS, LIFO-BS, and EABS in Fig. 11. Because of the urgent situations, the emergency packets need to be delivered successfully. Thus, the forwarding percentage is also an important metric to evaluate the real-time performance of the scheduling scheme. It is the proportion of delivered packets and generated packets during 500-time slots. Due to the limited delivery capacity of the network, with the increase of packet arrival rate, the total number of packets increases, so the proportion of the delivered packets gradually reduces. For the regular packets, EABS scheme considers the distance factor on the basis of LIFO-BS scheduling scheme, and the packets can reach the destination node faster. Therefore, the number of delivered packets increases, so does the forwarding percentage as shown in Fig. 11a. For the emergency packets, they are given the priority to be delivered. Almost all the emergency packets are delivered during the simulation time. Thus, the forwarding percentage is higher (close to 1) as shown in Fig. 11b.

We show the average forwarding percentage of different packets in Fig. 12. The forwarding percentage of emergency



(a) Average forwarding percentage

(b) Forwarding percentage of emergency packets

Fig. 11. Forwarding percentage.

packets is higher than that of all packets under our proposed EABS scheme. The scheme should guarantee the success of the emergency packets delivering. It is important for the EIoT. Although, the forwarding percentage of regular packets is relatively lower than the average forwarding percentage, it is still higher than that of the traditional BS and LIFO-BS schemes.

5.4 Average Energy Consumption Per Packet

Fig. 13 shows the average energy consumption for each packet of BS, LIFO-BS, and our proposed EABS scheme. The average energy consumption per packet of the BS and LIFO-BS scheduling schemes is relatively high because of the lengthy path of packets. The more forwarding times, the more energy consumption of each packet is. Under our proposed EABS scheme, the forwarding times of regular packets reduce before they have been delivered because of considering the distance factor. Thus, the average energy consumption of each packet reduces as shown in Fig. 13a. For the emergency packets, the forwarding times are less, so does the energy consumption of each packet as shown in Fig. 13b.

Fig. 14 illustrates the energy consumption of each different packets under the EABS scheme. The forwarding times of emergency packets are less than that of regular packets. Thus, the average energy consumption per packet of emergency packets is less. However, the regular packets consume more energy before being successfully delivered.

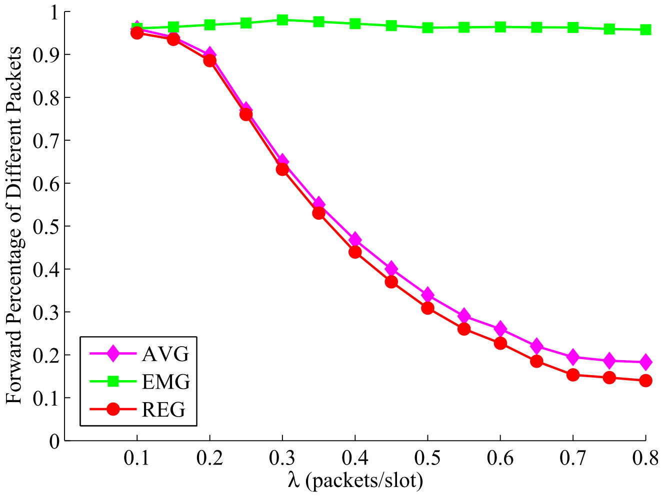


Fig. 12. Forwarding percentage of different packets.

5.5 Network Lifetime

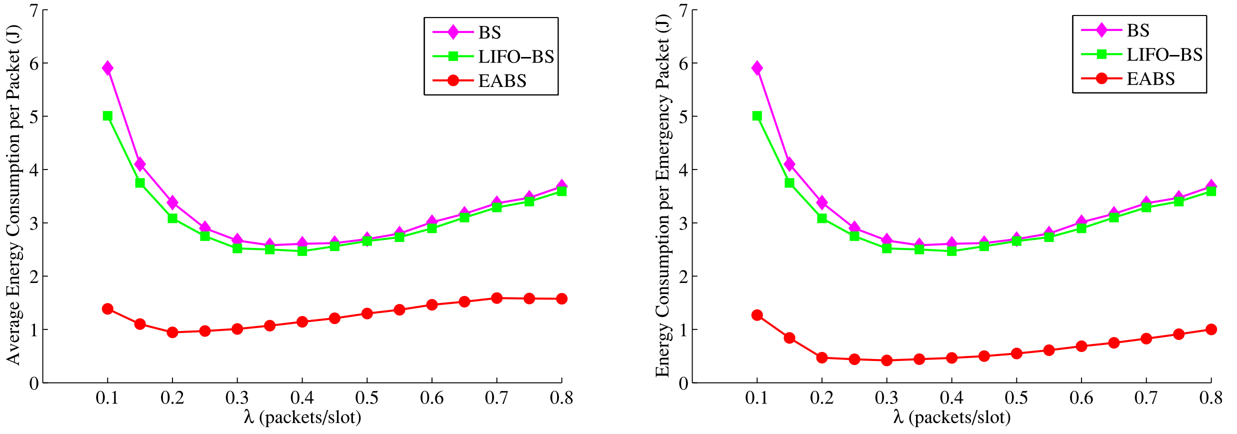
The lifetime of the network under BS, LIFO-BS, and our proposed EABS scheme are shown in Fig. 15. We assume that when a node with 100 J initial energy is run out, the node dies. We use the time that the first node is running out of energy as the lifetime of the network in this paper. Under the EABS scheme, each node consumes less energy than the BS and LIFO-BS scheduling schemes because of the less forwarding times. As a result, the network lifetime is longer, especially under the low network load. When every node forwards packet at each time slot with the network load increasing, the lifetime of EABS scheme tends to BS and LIFO-BS scheduling schemes, and the lifetime of network is finally converged.

5.6 Goodput

Fig. 16 shows the goodput of BS, LIFO-BS, and EABS. It can be seen that the goodput of LIFO-BS scheduling scheme is higher than the traditional BS scheme because the LIFO-BS scheme changes the queueing model and reduces the end-to-end delay. Under our proposed EABS scheme, the number of delivered regular packets increases because of the distance factor. While, the emergency packets have priority to be delivered. Almost all the emergency packets are delivered during the simulation time. As a result, the goodput of network increases.

6 CONCLUSION

In this paper, we propose a novel event-aware backpressure scheduling scheme to enhance the real-time performance of emergency packets for EIoT. In particular, we first designed a backpressure-based queue model according to the arrival process of different packets that reduces the waiting time of emergency packets in queues. Furthermore, we combine the shortest path method with backpressure scheme to select the next-hop node, which reduces the transmission delay of emergency packets. Finally, the performance of



(a) Average energy consumption per packet

(b) Energy consumption per emergency packet

Fig. 13. Energy consumption per packet.

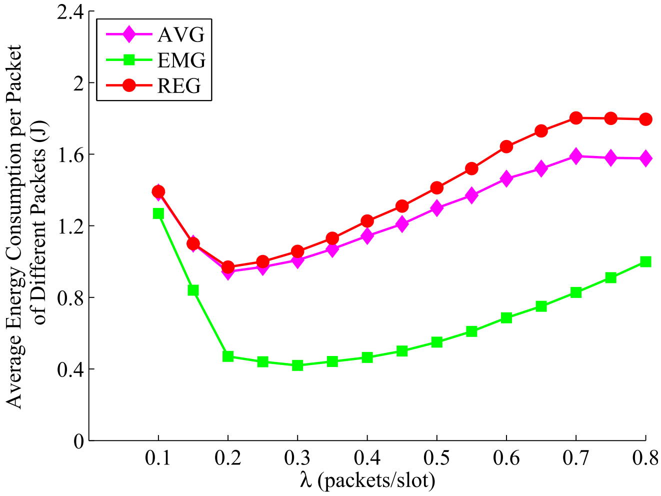


Fig. 14. Energy consumption per packet of different packets.

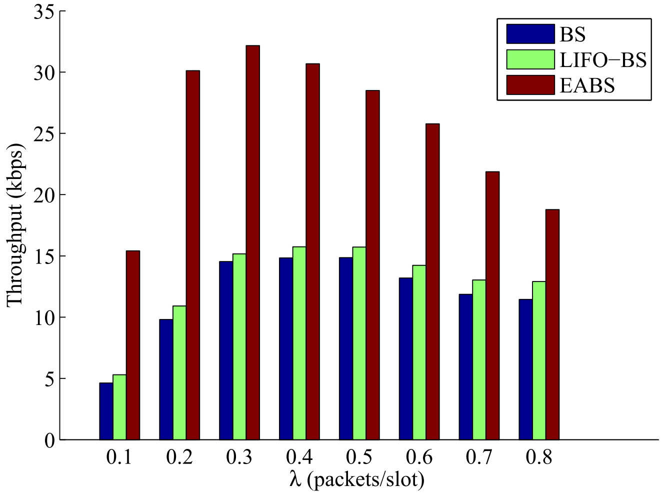


Fig. 16. Goodput.

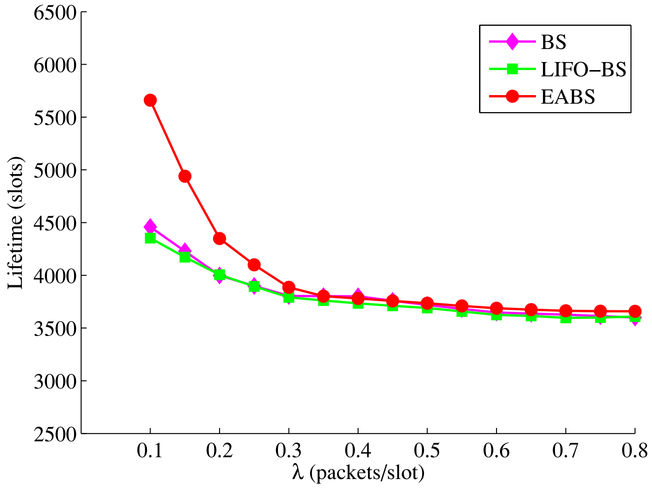


Fig. 15. Network lifetime.

EABS is evaluated. The simulation results show that EABS significantly reduces the end-to-end delay and increases the forwarding percentage compared with traditional backpressure scheme and LIFO-based scheme. For the emergency packets, the real-time performance is guaranteed, and the goodput of the whole network increases.

In the EABS scheme, we only consider the existence of emergency packets and regular packets. However, packets with different priorities can exist in the network. As an extension, our plan is to work on the arrival process of packets with multi-level priorities.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of P.R. China (Grant No. 61672131) and the Fundamental Research Funds for the Central Universities (Grant No. DUT16QY27).

REFERENCES

[1] S. Jeschke, C. Brecher, H. Song, and D. Rawat, *Industrial Internet of Things: Cybermanufacturing Systems*. Berlin, Germany: Springer, 2016.

[2] R. Squire and H. Song, “Cyber-physical systems opportunities in the chemical industry: A security and emergency management example,” *Process. Saf. Program.*, vol. 33, no. 4, pp. 329–332, 2014.

[3] H. Song, D. B. Rawat, S. Jeschke, and C. Brecher, *Cyber-Physical Systems: Foundations, Principles and Applications*. Burlington, MA, USA: Morgan Kaufmann, 2016.

[4] I. Butun, M. Erol-Kantarci, B. Kantarci, and H. Song, “Cloud-centric multi-level authentication as a service for secure public safety device networks,” *IEEE Commun. Mag*, vol. 54, no. 4, pp. 47–53, Apr. 2016.

[5] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.

[6] A. Bhorkar, M. Naghshvar, and T. Javidi, "Opportunistic routing with congestion diversity in wireless ad hoc networks," *IEEE ACM Trans. Netw.*, vol. 24, no. 2, pp. 1167–1180, Apr. 2016.

[7] A. Biondi, G. C. Buttazzo, and M. Bertogna, "Schedulability analysis of hierarchical real-time systems under shared resources," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1593–1605, May 2016.

[8] J. Teo, Y. Ha, and C. Tham, "Interference-minimized multipath routing with congestion control in wireless sensor network for high-rate streaming," *IEEE Trans. Mobile Comput.*, vol. 7, no. 9, pp. 1124–1137, Sep. 2008.

[9] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without routes: The backpressure collection protocol," in *Proc. ACM/IEEE Int. Conf. Inf. Process. Sens. Netw.*, 2010, pp. 279–290.

[10] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards," *Comput. Commun.*, vol. 30, no. 7, pp. 1655–1695, 2007.

[11] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *Proc IEEE Conf. Decision Control*, 1992, vol. 37, no. 12, pp. 1936–1948.

[12] 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.

[13] A. Sridharan, S. Moeller, and B. Krishnamachari, "Investigating backpressure based rate control protocols for wireless sensor networks," Ming Hsieh Dept. Electr. Eng., Univ. Southern California, Los Angeles, CA, USA, Tech. Rep. CENG-2008, vol. 7, 2008.

[14] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Backpressure routing made practical," in *Proc IEEE Conf. Comput. Commun. Workshops*, 2010, pp. 1–2.

[15] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," *Proc IEEE/ACM Trans. Netw.*, vol. 3, pp. 1333–1344, Dec. 2007.

[16] M. J. Neely, E. Modiano, and C. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE ACM Trans. Netw.*, vol. 16, no. 2, pp. 396–409, Apr. 2008.

[17] M. J. Neely and R. Urgaonkar, "Optimal backpressure routing for wireless networks with multi-receiver diversity," *Ad Hoc Netw.*, vol. 7, no. 5, pp. 862–881, 2009.

[18] A. Dvir and A. V. Vasilakos, "Backpressure-based routing protocol for DTNs," *SIGCOMM - Proc. SIGCOMM Conf.*, vol. 41, no. 4, pp. 405–406, 2011.

[19] H. Seferoglu and E. Modiano, "Diff-Max: Separation of routing and scheduling in backpressure-based wireless networks," in *Proc. IEEE INFOCOM*, 2013, pp. 1555–1563.

[20] L. X. Bui, R. Srikant, and A. Stolyar, "A novel architecture for reduction of delay and queueing structure complexity in the backpressure algorithm," *IEEE ACM Trans. Netw.*, vol. 19, no. 6, pp. 1597–1609, Dec. 2011.

[21] A. Warrier, S. Janakiraman, S. Ha, and I. Rhee, "DiffQ: Practical differential backlog congestion control for wireless networks," in *Proc. IEEE INFOCOM*, 2009, pp. 262–270.

[22] B. Radunović, C. Gkantsidis, S. Ha, D. Gunawardena, and P. Key, "Horizon: Balancing TCP over multiple paths in wireless mesh network," in *Proc Annu. Int. Conf. Mobile Comput. Netw.*, 2008, pp. 247–258.

[23] R. Laufer, T. Salonidis, H. Lundgren, and P. Le Guyadec, "A cross-layer backpressure architecture for wireless multihop networks," *IEEE ACM Trans. Netw.*, vol. 22, no. 2, pp. 363–376, Apr. 2014.

[24] L. Huang, S. Moeller, M. J. Neely, and B. Krishnamachari, "LIFO-backpressure achieves near-optimal utility-delay tradeoff," *IEEE ACM Trans. Netw.*, vol. 21, no. 3, pp. 831–844, Jun. 2013.

[25] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

[26] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "CDMA data QoS scheduling on the forward link with variable channel conditions," *Bell Labs., Tech. Memo*, vol. 4, pp. 1–45, Apr. 2000.

[27] Z. Liu, X. Yang, P. Zhao, and W. Yu, "Energy-balanced backpressure routing for stochastic energy harvesting WSNs," in *Proc. Lect. Notes Comput. Sci.*, 2015, pp. 767–777.

[28] A. Majidi and H. Mirvaziri, "BDCC: Backpressure routing and dynamic prioritization for congestion control in WMSNs," *Int. J. Comput. Netw. Inf. Secur.*, vol. 6, no. 5, 2014, Art. no. 29.

[29] A. L. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Syst.*, vol. 50, no. 4, pp. 401–457, 2005.

[30] A. Eryilmaz and R. Srikant,"Joint congestion control, routing, and MAC for stability and fairness in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1514–1524, Aug. 2006.

[31] Z. Jiao, B. Zhang, W. Gong, and H. Mouftah, "A virtual queue-based back-pressure scheduling algorithm for wireless sensor networks," *Eurasip J. Wireless Commun. Netw.*, vol. 2015, no. 1, pp. 1–9, 2015.

[32] B. Hu and H. Gharavi, "Greedy backpressure routing for Smart Grid sensor networks," in *Proc. IEEE Consum. Commun. Netw. Conf.*, 2014, pp. 32–37.

[33] R. Venkataraman, S. Moeller, B. Krishnamachari, and T. R. Rao, "Trust–based backpressure routing in wireless sensor networks," *Int. J. Sensor Netw.*, vol. 17, no. 1, pp. 27–39, 2015.

[34] Z. Jiao, Z. Yao, B. Zhang, and C. Li, "A distributed gradient-assisted anycast-based backpressure framework for wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, 2014, pp. 2809–2814.

[35] L. A. Maglaras and D. Katsaros, "Layered backpressure scheduling for delay reduction in ad hoc networks," in *Proc. IEEE Int. Symp. World Wireless Mobile Multimedia Netw. - Digital Process.*, 2011, pp. 1–9.

[36] J. Ryu, L. Ying, and S. Shakkottai, "Back-pressure routing for intermittently connected networks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–5.

[37] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, "On combining shortest-path and back-pressure routing over multihop wireless networks," *IEEE ACM Trans. Netw.*, vol. 19, no. 3, pp. 841–854, Jun. 2011.

[38] J. Lu, Z. Huang, N. Liu, and Q. Guan, "Distance-weighted backlog differentials for back-pressure routing in multi-hop wireless networks," in *Proc. IEEE/CIC Int. Conf. Commun.*, 2014, pp. 791–795.

[39] B. Ji, C. Joo, and N. B. Shroff, "Delay-based back-pressure scheduling in multihop wireless networks," *IEEE ACM Trans. Netw.*, vol. 21, no. 5, pp. 1539–1552, Oct. 2013.

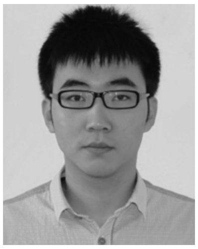
[40] M. Alresaini, M. Sathiamoorthy, B. Krishnamachari, and M. J. Neely, "Backpressure with adaptive redundancy (BWAR)," in *Proc. IEEE INFOCOM*, 2012, pp. 2300–2308.

[41] E. Athanasopoulou, L. X. Bui, T. Ji, R. Srikant, and A. Stolyar, "Back-pressure-based packet-by-packet adaptive routing in communication networks," *IEEE ACM Trans. Netw.*, vol. 21, no. 1, pp. 244–257, Feb. 2013.

[42] L. Ying, R. Srikant, D. Towsley, and S. Liu, "Cluster-based back-pressure routing algorithm," *IEEE ACM Trans. Netw.*, vol. 19, no. 6, pp. 1773–1786, Dec. 2011.

[43] M. Aazam and E. N. Huh, "E-HAMC: Leveraging Fog computing for emergency alert service," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops PerCom Workshops*, 2015, pp. 518–523.

Tie Qiu received the PhD and MSc degrees from the Dalian University of Technology (DUT), in 2005 and 2012, respectively. He is currently an associate professor in the School of Software, Dalian University of Technology, China. He was a visiting professor in electrical and computer engineering with Iowa State University in the USA (Jan. 2014-Jan. 2015). He serves as an associate editor of *Computers and Electrical Engineering* (Elsevier journal), an editorial board member of the *Journal of Advanced Computer Science and Technology (JACST)*, a guest editor of *Computers and Electrical Engineering* (Elsevier journal) and *Ad Hoc Networks* (Elsevier journal), a program chair of iThings2016, a TPC member of Industrial IoT2015 and ICSN16, a workshop chair of CISIS13 and ICCMSE15, a special session chair of WCC 2012 and CSA13, a TPC member of AIA13, EEC14, EEC15, and EEC16. He has authored/co-authored six books, more than 50 scientific papers in international journals and conference proceedings. He has contributed to the development of two copyrighted software systems and invented eight patents. He is a senior member of the China Computer Federation (CCF) and a senior member of the IEEE and ACM.



Ruixuan Qiao received the BE degree from the Dalian University of Technology, China, in 2014. He is working toward the master's degree in the School of Software, Dalian University of Technology (DUT), China. He participated in the "Open Source Hardware and Embedded Computing Competition" and won the first prize. He is an outstanding graduate student of DUT and has been awarded several scholarships in academic excellence and technology innovation. His research interests cover embedded system and internet of things. He is a student member of the IEEE.

Dapeng Oliver Wu received the PhD degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, Pennsylvania, in 2003. Since 2003, he has been on the faculty of the Electrical and Computer Engineering Department, University of Florida, Gainesville, Florida, where he is currently a professor. His research interests are in the areas of networking, communications, video coding, image processing, computer vision, signal processing, and machine learning. He received the University of Florida Research Foundation Professorship Award in 2009, AFOSR Young Investigator Program (YIP) Award in 2009, ONR Young Investigator Program (YIP) Award in 2008, NSF CAREER award in 2007, the IEEE Circuits and Systems for Video Technology (CSVT) Transactions Best Paper Award for Year 2001, the Best Paper Award in Globecom 2011, and the Best Paper Award in QShine 2006. Currently, he serves on the editorial board of the *IEEE Transactions on Circuits and Systems for Video Technology*, the *IEEE Transactions on Signal and Information Processing over Networks*, and the *IEEE Signal Processing Magazine*. He is the founder of the *IEEE Transactions on Network Science and Engineering*. He was the founding editor-in-chief of the *Journal of Advances in Multimedia* between 2006 and 2008, and an associate editor for the *IEEE Transactions on Wireless Communications* and the *IEEE Transactions on Vehicular Technology* between 2004 and 2007. He has served as general chair for the IEEE GlobalSIP 2015, technical program committee (TPC) chair for IEEE INFOCOM 2012, and TPC chair for the IEEE International Conference on Communications (ICC 2008), Signal Processing for Communications Symposium. He served as chair for the Award Committee, Technical Committee on Multimedia Communications, IEEE Communications Society. He is a fellow of the IEEE.