ARTICLE IN PRESS

Journal of Network and Computer Applications ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

ERGID: An efficient routing protocol for emergency response Internet of Things

Tie Qiu [a], Yuan Lv [a], Feng Xia [a,*], Ning Chen [a], Jiafu Wan [b], Amr Tolba [c,d]

[a] School of Software, Dalian University of Technology, Dalian 116620, China
[b] School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510641, China
[c] Riyadh Community College, King Saud University, Riyadh 11437, Saudi Arabia
[d] Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Egypt

ARTICLE INFO

Article history:
Received 15 December 2015
Received in revised form
22 March 2016
Accepted 22 June 2016

Keywords:
Internet of things
Delay Iterative
Emergency response
Global information decision

ABSTRACT

In recent years, Internet of Things (IoT) has been applied to many different fields such as smart home, environmental monitoring and industrial control system, etc. Under the pressure of the continuous expansion of network scale, how to ensure the real-time emergency response ability during data transmission has become a challenging problem for researchers. In this paper, we propose a routing protocol for Emergency Response IoT based on Global Information Decision (ERGID) to improve the performances of reliable data transmission and efficient emergency response in IoT. Specifically, we design and realize a mechanism called Delay Iterative Method (DIM), which is based on delay estimation, to solve the problem of ignoring valid paths. Moreover, a forwarding strategy called Residual Energy Probability Choice (REPC) is proposed to balance the load of network by focusing on the residual energy of node. Simulation results and analysis show that ERGID outperforms EA-SPEED and SPEED in terms of end to end (E2E) delay, packet loss and energy consumption. Additionally, we also carry out some practical experiments with STM32W108 sensor nodes, and observe that ERGID can improve the real-time response ability of network.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

In Internet of Things (IoT) (Hasan and Curry, 2015; Tsai et al., 2014), we deploy large-scale sensor networks to connect each of objects with unique identity up in the physics world (Tang et al., 2014; Hsieh et al., 2013; Wan et al., 2014). The basic idea of IoT is to enable a variety of objects around us, such as smart phones, electronic tags (RFID), sensors, tablet computers and other equipment, to communicate with each other (Castellani et al., 2010). IoT-related technology (Qiu et al., 2016, 2012) makes people more closely in touch with the physical world, and provides Context-Aware intelligence for users based on real-time collection of each sensor nodes. Users can not only get personalized information from surrounding environment, but also can sense and control objects in surrounding environment. Recently, many applications based on IoT technologies (Tunca et al., 2013; Zhang et al., 2012), such as forest fire monitoring and warning systems (Yin et al., 2013), medical monitoring systems (Domingo, 2012), smart home systems (Ashraf and Habaebi, 2015), etc. are proposed, which makes the size of these systems continuously increase. Therefore, how to improve the real-time performance and reliability of data collection or transmission process has become a critical problem.

As the key technology of IoT, the topology with large-scale sensor nodes is facing three major challenges (Oteafy and Hassanein, 2012; Almeida et al., 2015): first, due to the high reliability cannot be realized by the lower layer, it needs to be realized by the routing protocol in the network layer. Retransmission is an effective method to improve the reliability, but it also can extend the transmission delay and affect the performance of real-time. The second challenge is how the real-time performance can be provided by the routing protocol. Third, the routing protocol should have high robustness, the workload of each node should be fully considered. Due to nodes memory space and energy are limited, we should minimize the number of task of forwarding nodes. Therefore, when designing routing protocols of IoT, these issues above should be considered.

Real-time performance is one of the most critical requirements in different applications (Linderman et al., 2014; Qiu et al., 2015;

* Corresponding author.
E-mail addresses: qiutie@ieee.org (T. Qiu), farmarklv@gmail.com (Y. Lv), f.xia@ieee.org (F. Xia), xnchenning@gmail.com (N. Chen), jfwan2013@163.com (J. Wan), atolba@ksu.edu.sa (A. Tolba).

http://dx.doi.org/10.1016/j.jnca.2016.06.009
1084-8045/© 2016 Elsevier Ltd. All rights reserved.

Please cite this article as: Qiu, T., et al., ERGID: An efficient routing protocol for emergency response Internet of Things. Journal of Network and Computer Applications (2016), http://dx.doi.org/10.1016/j.jnca.2016.06.009

ARTICLE IN PRESS

Lu et al., 2015). In order to improve the real-time response performance, researchers have done a lot of works. Many methods are applied to ensure real-time routing maintenance and reduce the transmission delay (Frye et al., 2006; Ashraf et al., 2008), where transmission delay needs to be estimated to select the paths with smaller transfer cost and lower queuing delay, meanwhile, the data transmission path should change with the state of the environment to avoid network congestion and redundant packet loss. Sensor nodes in the network equip a small battery, whose energy is fixed (Suryadevara et al., 2014). Thus, the load of network should be balanced to reduce redundant network loss and prolong the lifetime of the network.

In this paper, we propose ERGID protocol, which aims to achieve efficient emergency response for IoT called ErIoT. Our main contributions are as follows:

- We design a mechanism called Delay Iterative Method (DIM), which is based on delay estimation to solve the problem of ignoring valid paths. DIM mechanism puts the node into candidate set according to the value of delay. At the same time, the routing information table is periodically updated by neighbor communication to ensure real-time performance for ErIoT.
- In order to balance the load of network, we present a novel strategy called Residual Energy Probability Choice (REPC) for data forwarding. REPC mechanism introduces residual energy status of node into next hop selection. The node with more residual energy has greater probability to become forwarding node. Experimental results show that our algorithm outperforms EA-SPEED and SPEED in terms of E2E delay, packet loss and energy consumption.

The rest of this paper is organized as follows. Section 2 describes some proposed real-time routing protocols in IoT. In Section 3, the problem of global information routing decisions and local information routing decisions are discussed. Section 4 describes the mechanism we proposed for reducing E2E delay and balancing networks load. In Section 5, we use simulation and practical experiments to compare ERGID protocol's performance with SPEED protocol and EA-SPEED protocol. Finally, we conclude this paper.

2. Related works

In the past few years, many studies on real-time routing protocols have been proposed (Stankovic, 2014; Hasan and Curry, 2014). And most of them focus on two major problems about data transmission. First, the protocol needs to ensure reliability of real-time packets and reduce number of blank areas caused by loss and delay. Second, the protocol needs to balance energy consumption of network and avoid premature death of some nodes.

To solve the first issue, Oh et al. (2013) proposed an opportunistic real-time data dissemination protocol, which exploits both broadcasting nature and temporal opportunity concept. Generally speaking, the nodes with shorter time tolerated queue have higher forwarding priority, i.e., they are more likely to be selected as the forwarding node. He et al. (2003) proposed SPEED protocol. Obtaining deadline of data packets from source node to the destination node requires estimating transmission rates of neighbor nodes. If the rate is higher than the global rate, paths can be added to the candidate set. The protocol uses the geographic forwarding mechanism and the node can obtain the local information for routing design from neighbors, which assurances high-speed network packets forwarding and real-time performance. When some areas do not meet the requirements of the protocol, SPEED uses Stateless Nondeterministic Geographic Forwarding (SNGF) mechanism to inform forwarding nodes precursor. As the result, the node doesnt send packets to these areas any more. As a typical real-time routing protocol, SPEED protocol has better scalability for large-scale sensor networks, but it still depends on the local information decision. Meanwhile, nodes energy balance cannot be considered. Lee and Ekici (2006) proposed MM-SPEED protocol which is the extension of SPEED protocol by supporting multi-rate and multi-path transmission. MM-SPEED aims to improve the reliability of data transmission, which offers two QoS options: real-time performance and reliability. The packets can make the best choice according to the situation of network. Multi-path forwarding strategy increases the probability of avoiding a blank area. Zhao et al. (2007) proposed FT-SPEED protocol, which provides more efficient solution to solve the problem of the blank area. All the nodes around blank areas are marked as edge node. The ordinary node and the edge node use different forwarding policies. Chen and Gao (2014) proposed routing protocol RRAD. To achieve Quality of Service (QoS) guarantee, RRAD uses the real-time delivery strategy based on probability, which can ensure the performance of reducing E2E delay. Forwarding strategies above are based on geography location information, but they are unable to balance the residual energy of the network. If we want to ensure timely delivery of the nodes, the nodes energy consumption needs to make trade-off.

Han et al. (2007) proposed ARP protocol which uses energy packets for real-time performance and limitations. The protocol dynamically changes the transmission rate of data packets and adjusts the priority of transmission strategies. Heo et al. (2006) proposed EAR-RT real-time routing protocol which is based on the improvements of EAR and EAR-DPS protocol. EAR-DPS can be applied in multiple paths routing process. Each neighbor node is assigned a probability that is used for forwarding packets. This probability is inversely proportional to the sum of the residual energy. Aissani et al. (2013) proposed EA-SPEED protocol. The protocol takes the energy balance into consideration and provides a path for the node selection while the residual energy is calculated. It uses the weights and the local velocity formulas for routing selection, aiming to balance the network load and energy. Ababneh et al. (2012) proposed EEMM, which can maximize network throughput and improve energy utilization. Zheng et al. (2013) proposed a kind of location-based clone detection protocol which applies ring structure to ensure the success rate of cloning to detect attacks. Using ring structure greatly improves the performance of balancing energy consumption of network and prolongs the lifetime of sensor networks.

Real-time routing protocols listed above are either based on location information or node energy to avoid the rapid death of network. However, with the development of technology, we find that the local information routing decision has its own blindness that it is unable to provide the most accurate decision for routing. For this reason, some researchers pay their attention to solutions based on global information decisions.

3. Problem statement

3.1. Description of notations

The notations used in this paper are listed in Table 1.

3.2. Local information decision and global information decision

SPEED routing protocol is the representative of local information decisions route which has many advantages. First, the protocol has a smaller maintenance cost. SPEED just needs to exchange data information between neighbor nodes. Second, the

Please cite this article as: Qiu, T., et al., ERGID: An efficient routing protocol for emergency response Internet of Things. Journal of Network and Computer Applications (2016), http://dx.doi.org/10.1016/j.jnca.2016.06.009

Table 1
Notation explanation.

Notation	Description
Dx	The relative distance between neighbors
$Speed$	The value of transmission speed between neighbors
V_{SoN-NN}	The value of transmission speed between source node to normal node
V_{BD}	The value of Speed between Node B and Node D
V_{min}	The value of min-Speed define by SPEED protocol
T_{ABDS}	The value of delay via path $ABDS$
T_{ACES}	The value of delay via path $ACES$
$P(i, j)$	The shortest path from vertex i to j
Vi	Vertex i
$P'(k, s)$	The shortest distance from k to s
$dist[i]$	The shortest distance from the source node $V0$ to node i
TS	Timestamp
Tm	The current time of Node m
T_{Ax}	The delay between Node A and Node x
RA	The neighbor nodes set of Node A
T_{A-Sink}	The delay from Node A to sink node
$T_{A-Sink}(B)$	The delay from Node A to sink node via Node B
RT	The value of residual time that packet can exist
CN	The set of Node whose delay from itself to sink is smaller than RT
E_m	The residual energy of Node m
P_m	The forwarding probability of Node m

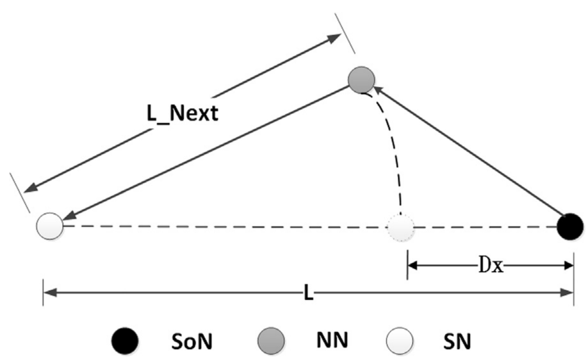


Fig. 1. SPEED Algorithm Model.

local information routing decision has good performance for large-scale sensor networks. The calculation method about the value of Speed between neighbors is shown in Fig. 1.

Here are the definitions of Node types:

- Sink Node, represented by *SN*.
- Next Node, represented by *NN*.
- Source Node, represented by *SoN*.

Due to *SN* is fixed, its location information is known. Each node needs to use GPS to obtain its location information, and then calculate the distance from itself to the *SN*. For example, we can use the following method to calculate the distance from *SoN* to *NN*, $Dx = L - L_{Next}$. According to the value of estimation delay D between the *SoN* and *NN*, we can get $V_{SoN-NN} = Dx/D$.

Since the delay estimation is between the neighbor nodes, SPEEDs routing decision algorithm is based on the local information decision. Local information decision route exists the blindness in large-scale networks (Laitrakun and Coyle, 2013). When the size of the network is large enough, the blindness can cause many serious problems, such as ignoring valid paths. Therefore, we hope to find a protocol which can use the global information decision strategy to solve the problem above.

The global information routing decisions have the following advantages: first, the global information node routing decisions can provide a more reliable forwarding path. With knowing the global information, the nodes in the network can choose the most efficient forwarding path. second, the global information routing decision can reduce network energy consumption, and balance the load of network. The decision can avoid forwarding data packet into the high-load area, which may cause rapid death of some nodes. finally, the global information routing decision can meet the requirements of some applications in IoT, such as real-time traffic monitoring, forest fire monitoring and warning system (AlHabashneh et al., 2011), etc.

Researchers have tried to use distributed base stations to store global information and provide global information to ordinary nodes around. The method is partly based on local information routing decisions, but it also needs extra maintenance cost and data transmission.

3.3. Problem of ignoring valid paths

In ErIoT, sensor nodes are randomly deployed (McDonald et al., 2013). Due to energy depletion or damage of sensor nodes, some areas without valid sensor nodes appear frequently. Using local information routing decisions, packets cannot be forwarded to this areas. For example, SPEED protocol uses SNGF mechanism to solve the problem.

As shown in Fig. 2, when the Node C perceives that there is no neighbor node which can be set as the next hop, it returns a pressure information packet to its predecessor. And then according to the cost comparison, the value of Speed between Node B and Node C is set as 0. When the Node A's packet is forwarded to Node B, Node B chooses Node D as its next hop. The strategy avoids choosing Node C, who may put the packet into the blank area and cause data loss.

Meanwhile, since the local information routing decision lacks global information, which may result in the problem of ignoring the valid paths. We can consider the following scene:

As shown in Fig. 3, the forwarding node put the neighbor node whose estimation delay is faster than Vmin into candidate set according to the SPEED protocol. Since the V_{BD} is 8 and it is less than V_{min}, Node A transmits a pressure information packet to Node B. As the result, the cost between Node A and Node B is set as infinity, In the process of transmission, Node A chooses Node C as its next hop.

Although the path $ABDS$ is ignored according to the SPEED protocol, it may still be a valid path when T_{ABDS} is lower than T_{ACES}. In this case, the SPEED may lead that network resources cannot be fully utilized and increase transmission delay through Node C. And then the path via Node C may not meet the requirements of protocol any more. At this moment, Node C returns a pressure information packet to Node A, which indicates the network condition is not suitable for data transmission. We assume that if Node B and Node C are the only successors of Node A, then Node A cannot transfer its data packets. As the result, Node A sends its own node status report to its precursor, which leads to all the paths via Node

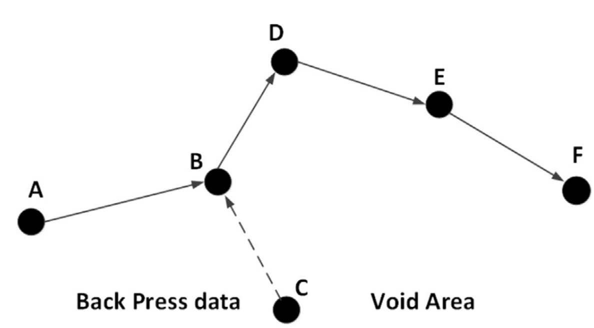


Fig. 2. SNGF mechanism.

Please cite this article as: Qiu, T., et al., ERGID: An efficient routing protocol for emergency response Internet of Things. Journal of Network and Computer Applications (2016), http://dx.doi.org/10.1016/j.jnca.2016.06.009

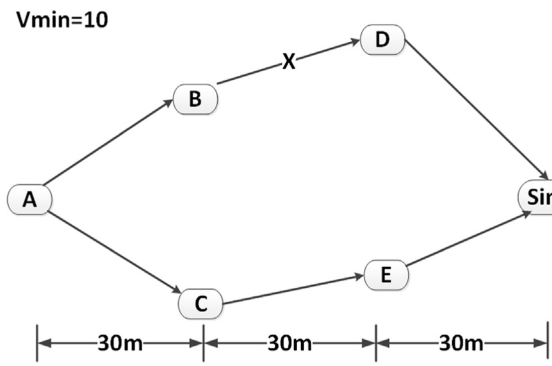


Fig. 3. Problem of ignoring valid paths.

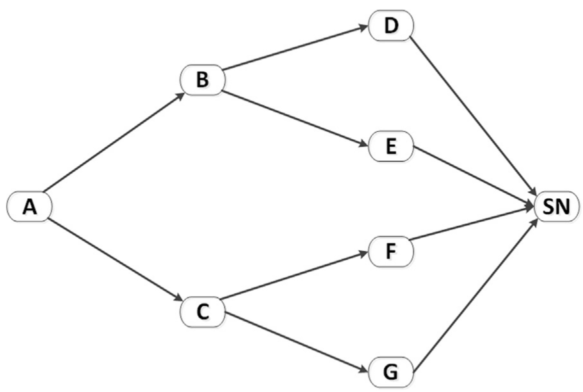


Fig. 4. ERGID Routing Protocol.

A cannot be used.

All of problems above may increase packet loss and energy consumption of node. Therefore, we want to find a routing mechanism which can meet the following three requirements :

- Use global information routing decisions, meanwhile, keep a low maintenance cost.
- Avoid the problem of ignoring valid paths, and make the full use of network resources.
- Balance the load of network and prolong the lifetime of network.

4. ERGID strategy

To solve the problem of ignoring valid paths caused by local information routing decision, we propose a protocol based on the global information, aiming to choose a path which meets the deadline requirements. In the protocol, we estimate the value of delay from *SoN* to *SN*, and focus on the energy-balance during the routing selection process.

For some special applications and scenarios, ensuring lower average E2E delay is of great importance. For example, temperature data of the forest fire monitoring and warning system is serial. We should ensure continuity of the temperature data and real-time performance in order to match it with the temperature curve. In this paper, we use DIM mechanism which is based on Dijkstra algorithm to obtain global delay estimation. Meanwhile, REPC mechanism is proposed to balance the node energy consumption.

4.1. DIM

We design DIM based on Dijkstra algorithm. The Dijkstra algorithm uses the greedy algorithm to build the model, which is the best known of the shortest path method.

(1) The optimal structure of the shortest path
The properties can be described as: if $P(i,j)=\{Vi....\ Vk..Vs...Vj\}$ is the shortest path from vertex i to j, k and s is an intermediate vertex on this path, and then $P(k,s)$ must be the shortest path from k to s.
We suppose that $P(i,j)=\{Vi....\ Vk..Vs...Vj\}$ is the shortest path from vertex i to j, then $P(i,j)=P(i,k)+P(k,s)+P(s,j)$. $P'(k,s)$ is the shortest distance from k to s. Then there must be a shortest path $P'(k,s)$ from k to s, then $P'(i,j)=P(i,k)+P'(k,s)+P(s,j)<P(i,j)$. And $P(i,j)$ is the shortest path from i to j in contradiction.

(2) Dijkstra algorithm
If there is one of the shortest path $(Vi.\ Vk,\ Vj)$ from i to k and Vi is a vertex in front of the Vk, then $(Vi.....Vk)$ must be the shortest path from i to k. Dijkstra proposed method that through accumulating the local shortest path to get the global shortest path. Considering the source node $V0$, we select the vertex Vi which is of the shortest length among the adjacent vertexes. Through calculating, we can obtain the shortest distance from $V0$ to Vj.

$$dist[j]=\min\{dist[j],\ dist[i]matrix[i][j]\} \tag{1}$$

According to above, we assume the existence of $G=\ <V,E>$. The source Node is $V0$, and $U=\{sourcevertex,V0\}$. $dist[i]$ is the shortest distance from $V0$ to i. $path[i]$ is to record the vertex in front of i from $V0$ to i.

- Select the $dist[i]$ minimum vertex i from $V-U$, and i will be added to the U.
- Update dist value which is directly adjacent to vertex i.
- Until $U=V$, the algorithm stops.

Based on Dijkstra algorithm, DIM mechanism assigns the calculation tasks to each node. The mechanism uses two steps to make it effective. The first step is to estimate the transmission delay by communicating with neighbors. The second step is to calculate the total delay from each node to the *SN*. We put the node into candidate set according to the value of delay. At the same time, we update routing information by neighbor communication periodically to ensure real-time performance in IoT.

The delay estimation method (He et al., 2003) is applied to measure the delay between neighbor nodes. With a current timestamp *TS* sent from Node *m* to Node *n*, Node *n* can immediately extract *TS*. And then Node *n* replies an ACK packet with the *TS*. After the Node *m* received the ACK packet, it extracts *TS* from the packet. And then, according to Eq. (2), we can calculate T_{mn}:

$$T_{mn}=(T_m-TS)/2 \tag{2}$$

Each node and its neighbor nodes sequentially estimate transmission delay. They store the delay information in the routing table. The value of *Delay*(*sink*) is 0. According to real-time network status, the information is updated regularly by exchanging the packets.

Each node is required to calculate its delay to *SN*. As shown in Fig. 4, for Node *A* need to plus the delay from itself to Node *B* and delay from Node *B* to *SN*. We mark delay via Node *A*'s path as T_{A-Sink}. We choose path which has the smallest delay value and then set first node of this path as the next hop. At the same time, we set the value as Node A's estimated delay to the *SN*. Iterative delay is calculated by Eq. (3):

Please cite this article as: Qiu, T., et al., ERGID: An efficient routing protocol for emergency response Internet of Things. Journal of Network and Computer Applications (2016), http://dx.doi.org/10.1016/j.jnca.2016.06.009

$$T_{A-Sink} = \min_{x \in RA} \{T_{Ax} + T_{x-Sink}\} \tag{3}$$

For example, when Node D and Node E transfer their delay T_{D-Sink} and T_{E-Sink} to Node B, we can get $T_{B-Sink}(D)$ and $T_{B-Sink}(E)$ according to Eq. (3).

Assuming that $T_{B-Sink}(D) > T_{B-Sink}(E)$, we take the delay $T_{B-Sink}(E)$ as estimation delay from Node B to SN. And Node E is set as the next hop of Node B. Similarly, Node A selects Node B as its next hop by comparing $T_{A-Sink}(B)$ with $T_{A-Sink}(C)$. We assume $T_{AC} < T_{AB}$ and $T_{A-Sink}(B) < T_{A-Sink}(C)$, then Node A still chooses Node B as its next hop according to DIM, which proves that local delay cannot affect the global routing selection. Thus, ERGID protocol can solve the problem of ignoring valid paths existed in SPEED protocol.

Algorithm 1 describes how to obtain iterative delay. The parameters are defined as follows. (Node i is the source node, and Node j is its neighbor node.).

Input parameters:

- $NoID_i$: ID number of Node i.
- $T_{i,j}$: the value of estimation delay between the Node i and Node j.
- $toSink_j$: the iterative delay of Node j.
- Ds_i: the set of iterative delay of Node i via neighbor nodes.

Output parameters:

- $toSink_i$: the iterative delay of Node i.

Algorithm 1. Iterative Delay.

```
1:  procedure getITERATIVEDELAY(NoID_i, T_i,j)
2:  broadcastRequestPacket(NoID_i)
3:  toSink_j ← receiveRequestPacket()
4:  toSink_i[j] ← toSink_j + T_i,j
5:  Ds_i ← toSink_i[j]
6:  for all toSink_i[k] ∈ Ds_i
7:      toSink_i ← getMinDelaytoSink(toSink_i[k])
8:  end for
9:  broadcast(toSink_i) return toSink_i
10: end procedure
```

Algorithm 1 works as follows. In line 2, Node i sends a broadcast request packet to its neighbor node for obtaining the iterative delay. The third and fourth line shows that we plus the extracted iterative delay and the estimation delay from Node i to its neighbor respectively. And then the result and iterative delay are stored into the Node i (line 5, 6). In line 7, the algorithm shows that Node i picks out the minimum delay from set of iterative delay as its own delay to SN. In line 8, Node i find out the first node of forwarding path, and then set it as the next hop. At the end of algorithm, Node i broadcasts its iterative delay to help neighbor nodes update their routing information.

4.2. REPC

The key idea of REPC mechanism is taking the residual energy status of node into consideration, and letting the node with more residual energy has greater probability to become the forwarding node. In order to achieve REPC mechanism, we need to judge if a node can be added into the candidate set. For the purpose of ensuring punctual delivery of packets, we define a global deadline for the network. For good performance, we set the deadline for 100 s.

When a node sends a packet, we define RT as the residual time of node. During transmission, when a node forwards the packet to the next hop, the value of the RT can be calculated by Eq. (4).We assume that the packet is sent from Node A to Node B.

$$RT = RT - T_{AB} \tag{4}$$

REPC compares the iterative delay of each node with the value of RT. If the delay is less than RT, the node can be added into the candidate set CN.

$$CN = \{A | T_{A-Sink} < RT\} \tag{5}$$

In addition, when the neighbor nodes communicate with each other, they exchange the information of their residual energy. The source node extracts the value of each nodes residual energy in the CN. The forwarding probability of each candidate node can be calculated by Eq. (6):

$$P_m = E_m \Big/ \sum_{k \in CN} E_k, \; m \in CN \tag{6}$$

Then we randomly select the node from CN as the next hop according to P_m. Compared with the SPEED protocol, ERGID can avoid repeatedly selecting the node with higher transmission rate as the forwarding node in Fig. 5. As the result, the problems of rapid energy consumption and blank area can be improved obviously, i.e., it is helpful to prolong the lifetime of the network.

Algorithm 2 describes how the REPC works. The parameters are defined as follows. (Node i is the source node, and Node j is its neighbor node)

Input Parameters:

- Ds_i: the set of iterative delay of node i via neighbor nodes.
- $E[k]$: Node k's residual energy.
- PC: the probability of Node i's forwarding node set.
- $T_{i,j}$: the value of estimation transmission delay between the Node i and Node j.
- CN: the candidate set of Nodes i.
- $p_i[k]$: the forwarding probability of a Node k.
- RT: the value of residual time stamp.

Output parameters:

- FID: forwarding node ID.

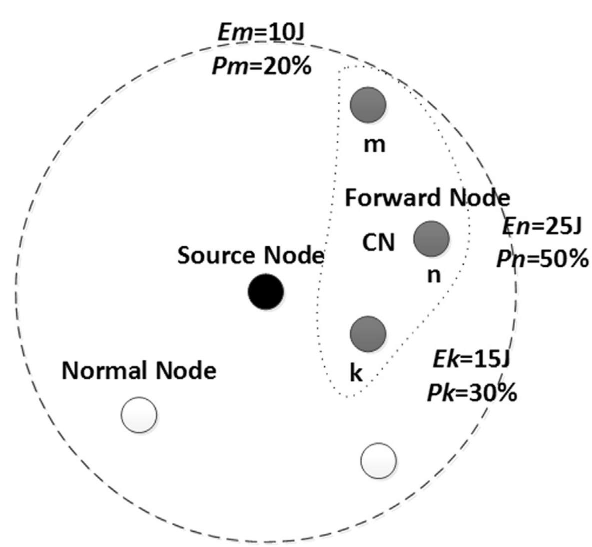


Fig. 5. Forwarding probability of Candidate set.

Please cite this article as: Qiu, T., et al., ERGID: An efficient routing protocol for emergency response Internet of Things. Journal of Network and Computer Applications (2016), http://dx.doi.org/10.1016/j.jnca.2016.06.009

ARTICLE IN PRESS

Algorithm 2. REPC.

1: **procedure** $PROBABILITYCHOICE(Ds_i, E[k])$
2: $RT \leftarrow receiveDataPacket()$
3: $T_{i,j} \leftarrow receiveDataPacket()$
4: $RT \leftarrow RT - T_{i,j}$
5: **for all** $toSink_i[k] \in Ds_i$ **do**
6: **if** $toSink_i[k] < RT$ **then**
7: $CN \leftarrow k$
8: **end if**
9: **end for**
10: **for all** k **do**
11: $E_{i-sum} = E_{i-sum} + E[k]$
12: **end for**
13: **for all** k **do**
14: $p_i[k] = E[k]/E_{i-sum}$
15: $PC \leftarrow p_i[k]$
16: **end for**
17: $FID \leftarrow ProbabilityChoice(PC)$
18: $ForwardingDataPacket(FID)$ **return**
19: **end procedure**

Algorithm 2 works as follows. In lines 2–4, when Node i received an information packet, it extracts RT and $T_{i,j}$. And then we update RT. In lines 5–9, we put node which meets the requirement into CN. In lines 10–12, we compute energy of nodes in the CN. In lines 13–15, we calculate the probability of each node in CN. In line 16, we select FID according to the probability of each node. In line 17, we forward the packet to the node which ID is equal to FID.

5. Performance evaluation

We use NS2 network simulator to carry out our simulation. In the simulation, we utilize SPEED protocol and EA-SPEED protocol as comparison algorithms from the following three aspects: average E2E delay, loss rate and average energy consumption of node. Besides, the simulation results evaluate the real-time performance and the ability of balancing network resources of each node of ERGID.

5.1. Simulation configurations

In our simulations, we pay more attention to real-time performance of data transmission. Take the forest fire monitoring and warning system for an example, we use multiple trigger nodes to simulate the situation of emergency fire in the forest. The average E2E delay in this case has directly effect on the temperature match of the fire information. In our simulation, simulation time is set as 100 s, and the packet latency period of ERGID protocol is 3.5 s, while SPEED and EA-SPEEDs minimum propagation speed are 700 m/s. And moreover, the initial energy of all the normal sensor nodes is set as 20 J. Table 2 shows some other main parameters in our simulation. We set 5 source nodes and the data packets are generated by CBR stream. Transmission rates of controller vary in 5 kb/s, 15 kb/s, 25 kb/s, 35 kb/s, 45 kb/s, 55 kb/s, 65 kb/s, 75 kb/s and 85 kb/s. And a total of 500 or 1000 nodes are randomly deployed in a sensor field of 1000*1000 m^2.

5.2. Average contract E2E delay

Figs. 6 and 7 illustrate the average E2E delay performance of

Table 2
Simulation parameters.

Parameters	Value
MAC layer	IEEE 802.11
Radio	RADIO-NONOISE
Antenna model	OmniAntenna
Queue model	Queue/DropTail/PriQueue
Queue size	50 packets
Channel of transmission	WirelessChannel
Wireless interface	WirelessPhy
Energy model	Ns-2 energy-model
Radio range	40 m
Transmission power	0.666 w

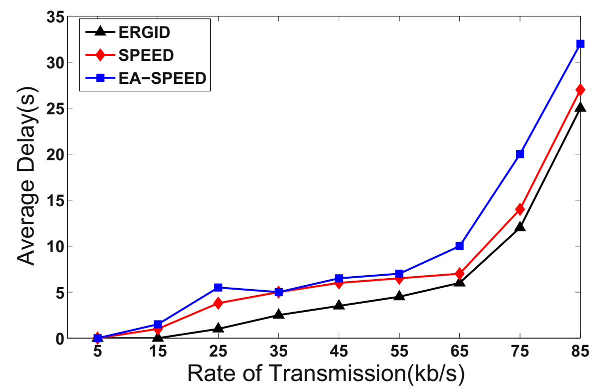


Fig. 6. Average delays of 500 nodes.

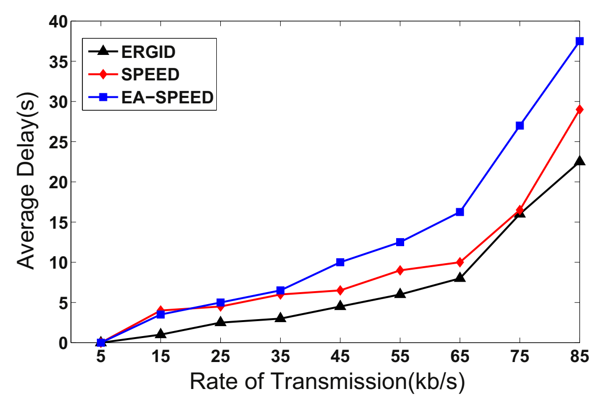


Fig. 7. Average delays of 1000 nodes.

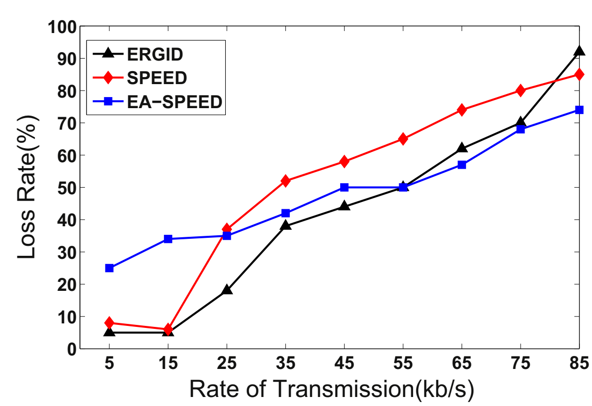


Fig. 8. Packet loss rate under different transmission rates.

Please cite this article as: Qiu, T., et al., ERGID: An efficient routing protocol for emergency response Internet of Things. Journal of Network and Computer Applications (2016), http://dx.doi.org/10.1016/j.jnca.2016.06.009

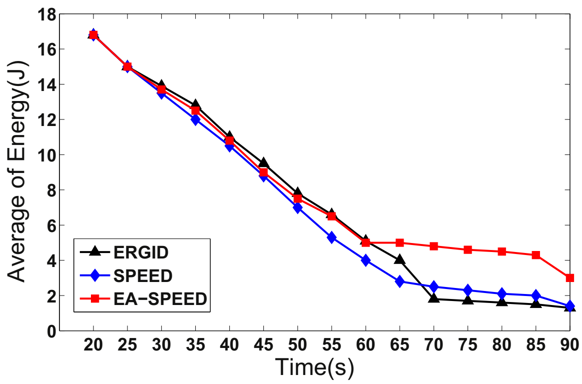


Fig. 9. Average energy of the 500 nodes.

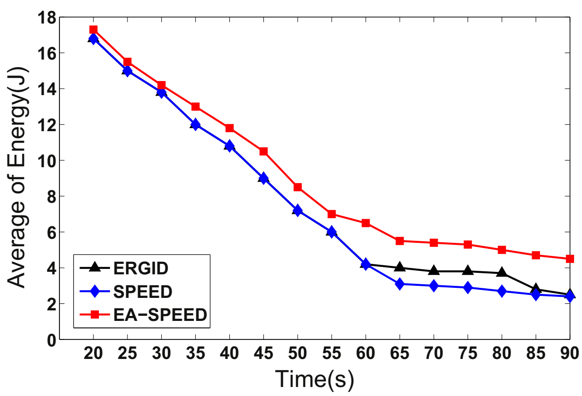


Fig. 10. Average energy of the 1000 nodes.

ERGID, SPEED and EA-SPEED. When the transmission rate is 5 kb/s, average E2E delay of three protocols are maintained at a very low level (2 s or less). When the transmission rate increases, the delay of SPEED and EA-SPEED increase rapidly. Meanwhile, ERGID outperforms both of them by 2 s. When the transmission rate is 75 kb/s or higher, due to the load has beyond networks endurance, the packet cannot meet the transmission requirements any more, but the performance of ERGID still outperforms SPEED and EA-SPEED.On the other hand, we also calculate the standard deviations of the delay times, which is another important aspect to evaluate the performance. The standard deviation of ERGID is less than 3.41 s, and more stable than SPEED (4.92 s) and EA-SPEED (5.30 s).

In addition, when we set the number of nodes as 1000, ERGID, SPEED and EA-SPEED protocols have the same performance as the node number is 500. One obvious difference is that when the transmission rate is lower than 65 kb/s, the gaps of ERGID, SPEED and EA-SPEED become much larger. Thus, when the number of node comes to 1000, ERGID still has better performance than the others.Meanwhile, the standard deviation of ERGID is less than 4.07 s, and has better performance than the others.

5.3. Packet loss

As shown in Fig. 8, we can observe that when transmission rate varies from 5 kb/s to 15 kb/s, loss rates of SPEED and ERGID are lower than 10%, while EA-SPEED is 25%. When the transmission rate increases to 25 kb/s, loss rates of ERGID and SPEED increase rapidly, indicating that the network begins to appear congestion. According to the requirements of the protocols, high-load nodes start discarding packets at the moment. Meanwhile, loss rate of ERGID is still about 20% lower than EA-SPEED and SPEED, proving that ERGID is of great capability of balancing networks load. When

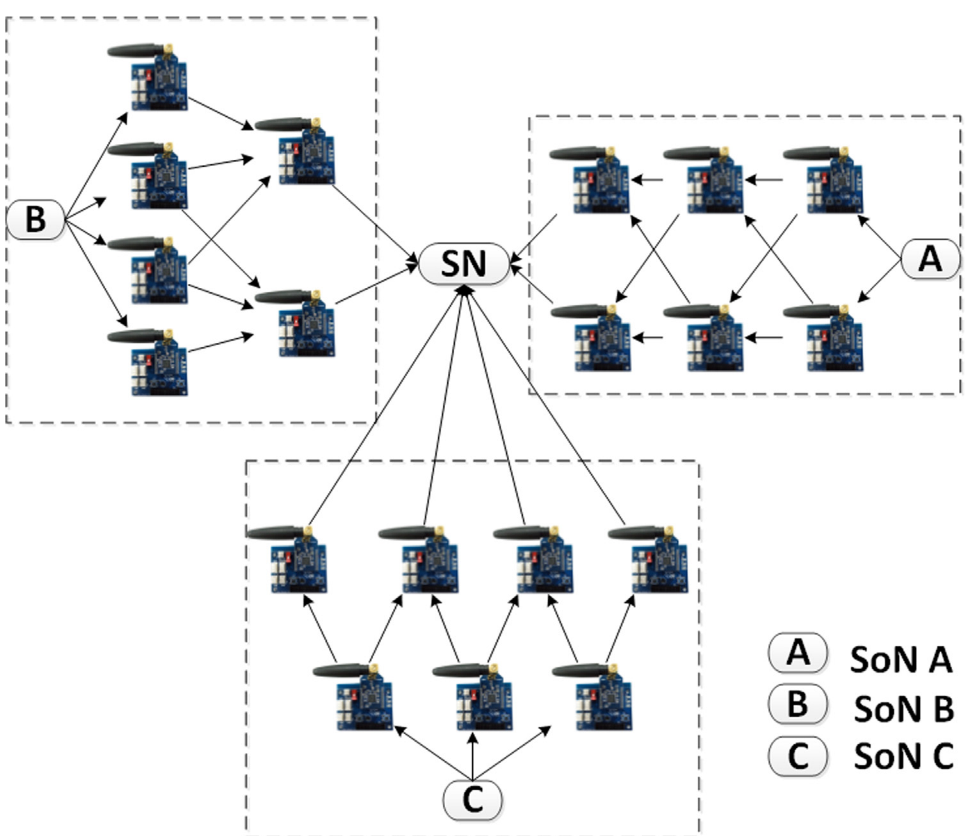
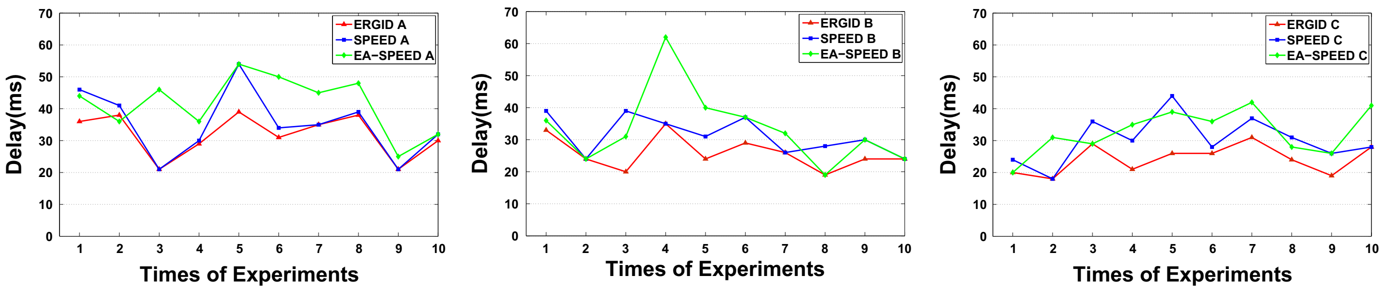


Fig. 11. Experimental topology.



(a) Delay of packet from *SoNA* (b) Delay of packet from *SoNB* (c) Delay of packet from *SoNC*

Fig. 12. Practical experimental transmission delay. (a) Delay of packet from *SoNA*, (b) Delay of packet from *SoNB*, (c) Delay of packet from *SoNC*.

transmission rate increases to 35 kb/s, loss rate of ERGID and EA -SPEED continues to get close. When the transmission rate larger than 45 kb/s, the loss rate of three protocols are all beyond 50%. When the transmission rate continues to increase, the loss rate of three protocols appear rapid growth, indicating that the load has reached the limit of network at this moment.

In general, we can find ERGID has a lower loss rate than SPEED and EA-SPEED. With the effect of DIM, ERGID can avoid packets falling into the area where the network load is high. Meanwhile, due to ERGID can find out every path that meets the requirements, it can avoid ignoring the valid paths and make the data packets more easily to be forwarded, which is helpful for reducing network load.

5.4. Average energy consumption

Figs. 9 and 10 depict the performance of energy consumption of SPEED, EA-SPEED and ERGID. When the number of node is 500 and the delivery rate is 30 kb/s, we can observe that from 20 s to 55 s, the curves of energy consumption linearly decrease, and we can infer that there is no node died in this period. After 55 s, some nodes begin to run out their energy. At this moment, EA-SPEED behaves the best performance of balancing the energy consumption among the three protocols. Although the performance of ERGID protocol is worse than EA-SPEED, it still outperforms SPEED, proving that ERGID protocol has advantages in reducing unnecessary energy consumption and prolonging the lifetime of the whole network.

In terms of energy, ERGID uses REPC mechanism to reduce energy consumption and avoid frequently selecting low energy node as the forwarding node. Meanwhile, due to the general energy consumption often occurs in the area with high E2E delay, the protocol prevents the data packets from being forwarded to these areas based on the global routing information decision, which ensures the balance of energy in the whole network indirectly.

5.5. Experimental verification based on STM32W108 nodes

We use chip STM32W108 based on 32-bit ARM Cortex-M3 core provided by STMicroelectronics corporation to do the experiments of SPEED, EA-SPEED and ERGID. The experimental topology is shown in the Fig. 11.

Our experimental area is 40 m*40 m which composed of 23 nodes. Node *A*, *B*, and *C* are *SoNs*. And the *SN* is connected to a PC through the serial port. The other 16 nodes are forwarding nodes. The transmission signal strength of sensor nodes are the same in simulation, but in actual experiment, different nodes have different signal strengths even though all the nodes are set the same RSSI. We conduct 10 groups of experiments for SPEED, EA-SPEED and ERGID. Each experiment node is set a random forwarding delay in order to simulate performance under different loads of the networks. Through several experiments tracking the data sent from node *A*, *B* and *C*, we can get an average delay. ERGID$_A$,ERGID$_B$ and ERGID$_C$ show the delay of packet from the *SoNA*, *SoNB* and *SoNC*, respectively. Similarly, we can get SPEED$_A$, SPEED$_B$, SPEED$_C$, and EA-SPEED$_A$, EA-SPEED$_B$, EA-SPEED$_C$.

As we can see in Fig. 12, ERGID outperforms SPEED during experiments. At the same time, EA-SPEED has the highest average E2E delay. Thus, we can conduct that the ERGID has a better performance in reducing the E2E delay than SPEED and EA-SPEED.

6. Conclusion

Local information routing decisions often lead to the blindness of path selection. Since forwarding node cannot effectively select the network transmission path without global information, unnecessary routing errors and packet loss may occur frequently. In this paper, a new protocol called ERGID is proposed based on the global information routing decisions. In ERGID, we design DIM and REPC mechanism to ensure more reasonable routing selection for each node. Simulation and practical experimental results show that ERGID has lower E2E delay and packet loss rate than SPEED and EA-SPEED, meanwhile, its energy consumption maintains at average level.

In general, ERGID can improve the efficiency of data transmission in network applications, and we make a trade-off between E2E delay and energy consumption. We will focus on improving energy consumption in large-scale ErIoT in the future.

Acknowledgment

This work was supported in part by National Natural Science Foundation of China (Grant nos. 61572106 and 61202443), and the Fundamental Research Funds for the Central Universities (nos. 2015ZZ079 and DUT16QY27).

References

Hasan, S., Curry, E., 2015. Thingsonomy: tackling variety in internet of things events. IEEE Internet Comput. 19 (2), 10–18.
Tsai, C., Lai, C., Vasilakos, A., 2014. Future Internet of Things: open issues and challenges. Wirel Netw 20 (8), 2201–2217.
Tang, C., Song, L., Balasubramani, J., Wu, S., Biaz, S., Yang, Q., Wang, H., 2014. Comparative investigation on CSMA/CA-based opportunistic random access for Internet of Things. IEEE Internet Things Journal 1 (2), 171–179.
Hsieh, H., Chen, J., Chen, Y., Kuo, S., 2013. Mobile IMS integration of the Internet of Things in ecosystem. In: Proceedings of the IEEE International Conference on Green Computing and Communications, IEEE Internet of Things, IEEE Cyber and Physical and Social Computing, Beijing, China, pp. 1870–1875.
Wan, J., Zhang, D., Zhao, S., Yang, L., Lloret, J., 2014. Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. IEEE Commun. Mag. 52 (8), 106–113.
Castellani, A., Bui, N., Casari, P., Rossi, M., Shelby, Z., Zorzi, M., 2010. Architecture and protocols for the internet of things: a case study. In: Proceedings of the 2010

8th IEEE International Conference on Pervasive Computing and Communications Workshops(PERCOM Workshops), IEEE, Mannheim, Germany, pp. 678–683.
Qiu, T., Luo, D., Xia, F., Deonauth, N., Si, W., Tolba, A., 2016. A Greedy Model with Small World for Improving the Robustness of Heterogeneous Internet of Things. Comput. Netw. 101 (6), 127–143.
Qiu, T., Ding, Y., Xia, F., Ma, H., 2012. A search strategy of level-based flooding for the internet of things. Sensors 12 (8), 10163–10195.
Tunca, C., Isik, S., Donmez, M., Ersoy, C., 2013. Performance evaluation of heterogeneous wireless sensor networks for forest fire detection. In: Proceedings of the 21st IEEE Signal Processing and Communications Applications Conference (SIU), IEEE, Haspolat, Turkey, pp. 1–4.
Zhang, G., Li, C., Zhang, Y., Xing, C., Yang, J., 2012. Semanmedical: a kind of semantic medical monitoring system model based on the IoTs sensors. In: Proceedings of the IEEE 14th International Conference on e-Health Networking, Applications and Services (Healthcom), Beijing, China, pp. 238–243.
Yin, J., Pei, J., Li, J., Guo, Y., Xu, W., 2013. Smart Home System Based on IOTS Technologies. In: Proceedings of the 5th International Conference on Computational and Information Sciences (ICCIS), Shiyan, Hubei, China, pp. 1789–1791.
Domingo, M.C., 2012. An overview of the internet of underwater things. J. Netw Comput. Appl. 35 (6), 1879–1890.
Ashraf, Q., Habaebi, M., 2015. Autonomic schemes for threat mitigation in internet of things. J. Netw. Comput. Appl. 49 (3), 112–127.
Oteafy, S., Hassanein, H., 2012. Towards a global iot: resource re-utilization in WSNs. In: Proceedings of the International Conference on Computing, Networking and Communications (ICNC), Maui, HI, United states, pp. 617–622.
Almeida, V., Doneda, D., Monteiro, M., 2015. Governance challenges for the Internet of Things. IEEE Internet Comput. 19 (4), 56–59.
Linderman, L., Jo, H., Spencer, B., 2014. Low-latency data Acquisition hardware for real-time wireless sensor applications. IEEE Sens. J. 15 (3), 1800–1809.
Qiu, T., Lin, C., Guo, W., Zhang, Y., 2015. STETS: a novel energy-efficient time synchronization scheme based on embedded networking devices. Microprocess. Microsyst. 39 (8), 1285–1295.
Lu, S., Wang, Z., Wang, Z., Zhou, S., 2015. Throughput of underwater wireless ad hoc networks with random access: a physical layer perspective. IEEE Trans. Wirel. Commun. 14 (11), 6257–6268.
Frye, L., Cheng, L., Du, S., Bigrigg, M., 2006. Topology maintenance of wireless sensor networks in Node failure-prone environments. In: Proceedings of the IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, United states, pp. 886–891.
Ashraf, U., Abdellatif, S., Juanole, G., 2008. Efficient route maintenance in wireless Mesh networks. In: Proceedings of the 3rd International Symposium on Wireless Pervasive Computing (ISWPC), IEEE, Santorini, Greece, pp. 712–716.
Suryadevara, N., Mukhopadhyay, S., Kelly, S., Gill, S., 2014. WSN-based smart sensors and actuator for power management in intelligent buildings. IEEE/ASME Trans. Mechatron. 20 (2), 564–571.
Stankovic, J., 2014. Research directions for the internet of things. IEEE Internet Things J. 1 (1), 3–9.
Hasan, S., Curry, E., 2014. Approximate semantic matching of events for the Internet of Things. ACM Trans. Internet Technol. (TOIT) 14 (1), 1–23.
Oh, S., Yim, Y., Lee, J., Park, H., Kim, S.-H., 2013. An opportunistic routing for real-time data in Wireless Sensor Networks. In: Wireless Communications and Networking Conference (WCNC), Shanghai, China, pp. 1157–1162.
He, T., Stankovic, J., Lu, C., Abdelzaher, T., 2003. SPEED: a stateless protocol for real-time communication in sensor networks. In: Proceedings of the 23th IEEE International Conference on Distributed Computing Systems, Providence, RI, US, pp. 46–55.
Lee, E.F.C., Ekici, E., 2006. MMSPEED: multipath multi-SPEED protocol for qos guarantee of reliability and timeliness in wireless sensor networks. IEEE Trans. Mob. Comput. 5 (6), 738–754.
Zhao, L., Kan, B., Xu, Y., Li, X., 2007. FT-SPEED: a fault-tolerant, real-time routing protocol for wireless sensor networks. In: 2007 International Conference on Wireless Communications, Networking and Mobile Computing(WiCom), Shanghai, China, pp. 2531–2534.
Chen, Q., Gao, H., 2014. Towards Reliable and Real-Time Routing with Active Slot Augmentation in Low-Duty-Cycle WSNs. In: Proceedings of the 9th International Conference on Wireless Algorithms, Systems, and Applications, Springer Verlag, Harbin, China, pp. 672–681.
Han, P., Zhou, X., Li, Y., Chen, X., Gao, C., 2007. An adaptive real-time routing scheme for wireless sensor networks. In: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW), Niagara Falls, ON, Canada, pp. 918–922.
Heo, J., Yi, S., Park, G., Cho, Y., Hong, J., 2006. EAR-RT: energy aware routing with real-time guarantee for wireless sensor networks. In: Proceedings of the 6th International Conference on Computational Science-ICCS 2006, Springer, Reading, United kingdom, pp. 946–953.
Aissani, M., Bouznad, S., Fareb, A., Laidoui, M., 2013. EA-SPEED: energy-aware real-time routing protocol for wireless sensor networks. Int. J. Inf. Commun. Technol. 5 (1), 22–44.
Ababneh, N., Ortiz, A., Timmons, N., Morrison, J., 2012. Energy-Efficient Multicast tree construction protocol for real-time data streaming in WSNs. In: Proceedings of the 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, Barcelona, Spain, pp. 26–33.
Zheng, Z., Liu, A., Cai, L., Chen, Z., Shen, X., 2013. ERCD: an energy-efficient clone detection protocol in WSNs. In: Proceedings of the 32nd IEEE Conference on Computer Communications, INFOCOM, Turin, Italy, pp. 2436–2444.
Laitrakun, S., Coyle, E., 2013. Optimizing the collection of local decisions for time-constrained distributed detection in WSNs. In: Proceedings of the 32nd IEEE Conference on Computer Communications, INFOCOM, Turin, Italy, pp. 1923–1931.
AlHabashneh, A., Ahmed, M., Husain, T., 2011. Reliability analysis of wireless sensor networks for forest fire detection. In: Proceedings of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC), Istanbul, Turkey, pp. 1630–1635.
McDonald, D., Sanchez, S., Madria, S., Ercal, F., 2013. A survey of methods for finding outliers in wireless sensor networks. J. Netw. Syst. Manag. 23 (1), 163–182.

Please cite this article as: Qiu, T., et al., ERGID: An efficient routing protocol for emergency response Internet of Things. Journal of Network and Computer Applications (2016), http://dx.doi.org/10.1016/j.jnca.2016.06.009