Wireless Pers Commun (2017) 95:4601–4621
DOI 10.1007/s11277-017-4109-3

# RTEA: Real-Time and Energy Aware Routing for Industrial Wireless Sensor Networks

Omid Tavallaie[1] · Hamid Reza Naji[1] · Masoud Sabaei[2] · Narges Arastouie[2]

Published online: 17 April 2017

**Abstract** This paper proposes a real-time and energy aware routing protocol for industrial wireless sensor networks, called RTEA. Two-hop neighborhood information routing is adopted from THVR routing protocol, however our routing protocol has low control packet overhead and differentiates packets based on their deadlines. The packet deadline is updated in each hop without using globally synchronized clocks. In addition to velocity, the distance to the sink is considered for reducing the number of relaying hops. Also, the delay between two one-hop neighbors is estimated by a new mechanism which considers the time that packets spend in each node beside the link delay. Simulation results show that RTEA can improve the performance in terms of delivery ratio and energy consumption.

**Keywords** Energy aware routing · Industrial wireless sensor networks · Real-time communication · Timeliness · Two-hop information

## 1 Introduction

Industrial Wireless Sensor Networks (IWSN) are used in a large number of industrial applications such as industrial condition monitoring systems and process control systems. IWSNs consist of wireless sensors that sense data from the industrial environment, process

✉ Hamid Reza Naji
hamidnaji@ieee.org

Omid Tavallaie
omid.tavallaie@ieee.org

Masoud Sabaei
sabaei@aut.ac.ir

Narges Arastouie
arastouie@aut.ac.ir

[1] Department of Electrical and Computer Engineering, Graduate University of Advanced Technology, Kerman, Iran

[2] Department of Computer and IT Engineering, Amirkabir University of Technology, Tehran, Iran

and send it to the sink by collaborating with each other. The sink is connected to a control unit which analyzes data, adjusts the behavior of a system or equipment, and notifies users of any problems [1–5]. In condition monitoring systems, sensors are installed on industrial equipment and monitor its condition based on the parameters such as vibration, temperature, and pressure. These parameters are used in control units for estimating the performance of the equipment. Thus, any defect in the equipment is detected in a short time that makes the repairing cost reduced [1]. In process control systems, IWSNs are used to improve the quality of manufacturing. In these applications such as cold chain management [6], IWSNs evaluate all the processes and monitor the quality of a product in its entire life from raw materials till the final product that is delivered to the consumer. IWSNs are used in structural health monitoring applications to monitor functionality and safety of bridges, buildings, and other critical structures [7–10]. In these applications, wireless sensor nodes measure the response of a structure before, during and after a natural or man-made disaster, that can be used in damage detection algorithms. Also, the repairing cost can be minimized by observing the current state of structures.

In IWSNs, Quality of Service (QoS) parameters such as timeliness are extremely important and have significant impacts on the performance of a system. Wireless sensors must send data to the sink in a reliable and timely manner since lost or delayed data may lead to wrong decisions in the control unit [2]. Despite this importance, providing QoS guarantee in IWSNs is challenging due to nature of these networks. Sensor nodes usually have constrained resources in terms of energy, memory, CPU cycles, and bandwidth. Topology of the network is dynamically changed due to node failure and mobility [11–18]. Moreover, enhancing one QoS parameter may degrade another one. For example, reliability can be improved by using multi-path routing, however, it can increase delay due to duplicate transmission.

In real-time applications of industrial wireless sensor networks such as condition monitoring systems, data must be delivered to the sink within a time deadline and the performance is measured based on the number of packets delivered to the sink before deadline. This measure is called Deadline Delivery Success Ratio (DDSR) [19]. Also, energy consumption is another important performance measure since sensor nodes usually use battery for energy supply [3]. To increase DDSR, packets must be sent to the sink with desired speed. One class of real time routing protocols employs geographic routing and provides desired speed by choosing a neighbor as the forwarding node which can satisfy desired delivery speed [3, 20–23]. In these protocols, each node maintains information of its neighbors in a neighbor table and routing decisions are locally made based on the geographical location of the neighbors. Most of these routing protocols use one-hop neighborhood information for making routing decisions and some of them are based on multi-hop neighborhood information [22]. As shown in [3], timeliness can be improved by using multi-hop neighborhood information, which increases control packet overhead and decreases network lifetime. In other words, the choice of multi-hop information is a tradeoff between timeliness and control packet overhead.

Real-time applications are usually classified as hard or soft real-time. Hard real-time applications cannot tolerate packet loss and they consider out-of-date data as lost packets which can lead to system failure. In soft real-time applications such as multimedia applications, a significant amount of packet loss can be tolerated. These applications use best effort services which try to deliver packets to the sink before their deadline and may have not strict timeliness requirement. In this paper, we propose a soft real-time routing method for IWSN, called RTEA. Our routing method is based on two-hop neighborhood information with low control packet overhead which spans over MAC and network layers.

RTEA considers the energy consumption and differentiates packets based on their deadlines. It can reduce number of hops that packets relayed by considering the distance to the sink as a new metric to determine the next forwarding node. Also, it uses a new method to estimate delay between two one-hop neighbors based on the time that packets spend in each node before transmitting to the next hop. Simulation results show that our proposed protocol can improve the performance in terms of delivery ratio and energy consumption. We use a dynamic weighting factor for incorporating energy metric in the routing metric and differentiate packets based on their deadlines. Also, we propose a new method for updating packet deadline in each hop without using globally synchronized clocks.

The rest of this paper is organized as follows: Sect. 2 presents some related works. Section 3 describes the proposed routing protocol. Section 4 evaluates the performance of RTEA by simulation. Finally, the paper is concluded in Sect. 5.

## 2 Related Work

Several routing protocols have been proposed for WSNs that provide QoS support in timeliness domain [2, 3, 19–27]. Most of these protocols are cross layer solutions and some of them also consider other QoS parameters such as reliability [2, 21, 27].

SPEED protocol [20] uses geographic forwarding to provide soft real-time guarantee in WSNs. In SPEED, each node maintains position and delay of one-hop neighbors in a neighbor table. It estimates the relay speed for the neighbors that are closer to the sink. Forwarding candidate set is defined as the set of neighbors that have relay speed larger than the desired speed. The neighbor in this set with the largest speed is selected in the highest probability. Also, SPEED considers the tradeoff between load balancing and optimal delivery delay. In order to avoid congestion, SPEED drops some packets to reduce congestion and uses backpressure rerouting to inform other nodes of congestion.

Felemban et al. [21] proposed MMSPEED, a cross layer routing protocol which provides service differentiation in the reliability and timeliness domains. MMSPEED extends SPEED by providing multiple layers of speed guarantee and dynamic compensation of local decisions. Prioritization between speed layers is achieved by mapping speed layers to MAC priority queues. For supporting reliability, MMSPEED uses multi-path forwarding while it does not consider energy consumption.

In [25], Yuan et al. proposed an energy-efficient real-time routing protocol. They limit forwarding candidate set by proposing a novel concept of Effective Transmission. In addition to distance to the sink, Effective Transmission considers the distance to source. It defines forwarding candidate set by the neighbors which are closer to the sink and farther from the source. Also, in this protocol, end-to-end delay requirement is separated into sum of point-to-point path's delay.

THVR [3] is a two-hop neighborhood information-based routing protocol which enhances real-time delivery in WSNs. In THVR, a packet deadline is mapped to a velocity and routing decisions are made based on two-hop neighborhood information. In addition to the velocity, THVR considers node's remaining energy level to select a neighbor as the next forwarding node. For getting two-hop neighborhood information, two rounds of Hello messages are sent that leads to heavy message overhead. To improve reliability, automatic repeat-request (ARQ) is employed and Window Mean with Exponentially Weighted Moving Average (WMEWMA) [28] is adopted to estimate delay. Also, THVR conducts

Initiative Drop Control to enhance energy utilization efficiency when no node can provide desired speed.

EARQ [2] provides QoS guarantee in reliability and timeliness domains while it considers energy awareness. In EARQ, each node calculates the probability of selecting a path by estimating energy cost, delay and reliability of the path to the sink. Only the paths are selected that can deliver a packet to the sink before its deadline. The probability of selecting a path is inversely related to the energy cost.

Quang et al. [19] proposed THVRG for IWSN to enhance real-time performance with energy efficiency. THVRG Applies THVR [3] to a gradient-based wireless sensor networks and makes routing decisions based on the number of hops to the sink. As THVR, for selecting the next forwarding node, a joint metric is calculated that considers delay and energy. In addition, THVRG uses an acknowledgement control scheme in order to reduce energy consumption and computational complexity.

In our proposed protocol, deadline of packets is updated in each node without using global synchronized clock by considering link delay and the time that packets spend in each node. Different priorities are considered for energy and delay metrics Based on the deadline of packets and geographical location of nodes. In comparison with other routing protocols, RTEA has low control overhead while it uses two hop neighborhood information for routing operations.

# 3 Proposed Routing Protocol

## 3.1 Neighbor Table and Control Packets

Similar to other geographic routing protocols, every node in RTEA is aware of geographical location of itself and the sink by using global positioning system (GPS) or other localization techniques [29, 30]. RTEA uses three types of control packets, LOCATION-PCKT, ENERGY-PCKT and MAX-SPEED-PCKT. First type contains geographical location of a node and is used for exchanging location information between one-hop neighbors. Each node periodically broadcasts a LOCATION-PCKT to inform its one-hop neighbors of its location. The rate of broadcasting this packet depends on the node's mobility. This rate can be very low when network is static or nodes move slowly. In addition to LOCATION-PCKTs, each node periodically broadcasts ENERGY-PCKTs which contain the remaining energy of a node. So, nodes can consider energy balancing in their routing decisions. A MAX-SPEED-PCKT contains the maximum speed that a node can send a packet to its one-hop neighbors. The functionality of MAX-SPEED-PCKT will be discussed in part 3.5 of this section. Nodes update their neighbor tables when they receive control packets.

Unlike other two-hop information based routing protocols, in RTEA each node does not need to maintain information of two-hop neighbors and only maintains the maximum speed that a neighbor can send a packet to the next hop. So, the size of neighbor table and computing complexity of our protocol is similar to one-hop information based routing protocols. Each entry inside the neighbor table consists of seven fields: (*NeighborID*, *X*, *Y*, *EstDelay*, *MaxSpeed*, *ExpireTime, RemainingEnergy*). *X* and *Y* indicate geographical location of neighbors. If an entry is not updated before *ExpireTime*, it will be removed. *ExpireTime* and *RemainingEnergy* are updated every time that a node receives an

ENERGY-PCKT. *EstDelay* is the estimated delay to a neighbor which will be discussed in part 3.4.

### 3.2 Routing Algorithm

First, we introduce some symbols and definitions in Table 1. RTEA routes a packet in node *i* based on the current *NS(i)* with the following steps:

Step 1 If the packet deadline is zero, node *i* drops the packet. Otherwise step 2 is conducted.

Step 2 The neighbors with *MaxSpeed* larger than zero that are closer to the sink, are added to the Candidate Set (*CS*). Formally,

$$CS(i) = \{j | j \in NS(i), Dis(j,d) < Dis(i,d), MaxSpeed_j > 0\} \tag{1}$$

Step 3 If *CS(i)* has no member, node *i* drops the packet and broadcasts a MAX-SPEED-PCKT with zero value to inform upstream nodes which are closer to the source. Otherwise step 4 is conducted.

Step 4 For every node inside the *CS(i)* such as *j*, node *i* calculates *TwoHopSpeed* which is the progress speed toward the sink that can be provided in next two hops, i.e.,

$$TwoHopSpeed_j^i = \frac{OneHopeSpeed_j^i + MaxSpeed_j}{2} \tag{2}$$

$OneHopSpeed_j^i$ indicates the forwarding speed that node *i* can send a packet to node *j* and it is calculated by:

$$OneHopSpeed_j^i = \frac{Dis(i,d) - Dis(j,d)}{OneHopDelay_j^i} \tag{3}$$

where $OneHopDelay_j^i$ is estimated delay between nodes i and j and it will be discussed in part 3.4.

**Table 1** Symbols and Definitions

| *s* | Source |
|---|---|
| *d* | Sink |
| *Deadline* | The packet deadline which is updated in every hop |
| $MaxSpeed_i$ | The maximum speed that node *i* can send packets to the next forwarding node |
| *SetSpeed* | Desired delivery speed |
| $TwoHopSpeed_j^i$ | The progress speed toward the sink that can be provided by two-hop forwarding nodes *i* and *j* |
| *Dis(i,j)* | Distance between node *i* and node *j* |
| *NS(i)* | The neighbor set of node *i* which involves the nodes that are inside the range of node *i* |
| $E_j^0$ | Initial energy of node *j* |
| $E_j$ | Remaining energy of node *j* |
| $OneHopDelay_j^i$ | The estimated delay between node *i* and node *j* |

Step 5 *SetSpeed* is calculated by (4) that indicates the desired delivery speed for the packet.

$$SetSpeed = \frac{Dis(i,d)}{Deadline} \tag{4}$$

Step 6 The nodes in *CS(i)* that have *TwoHopSpeed* larger than *SetSpeed* are added in the Final Candidate Set (*FCS*). Formally,

$$FCS(i) = \{j | j \in CS(i), TwoHopSpeed_j^i > SetSpeed\} \tag{5}$$

Step 7 If *FCS(i)* has no member, Drop Control unit is called. Otherwise step 8 is conducted. Drop Control unit will be discussed in part 3.6.

Step 8 Forwarding metric is calculated for all the members of FCS. The neighbor with the maximum value is selected as the best candidate.

### 3.3 Forwarding Metric

Selecting a neighbor as the next forwarding candidate is the most important part of each routing protocol. Some routing protocols create the forwarding candidate set based on the relay speed and remaining energy of the neighbors, but none of them consider the distance to the sink for selecting the best candidate. RTEA selects the next forwarding node according to the relay speed, distance to the sink and remaining energy of the neighbors. As shown in (1) and (5), all the final candidates are closer to the sink and have *TwoHopSpeed* larger than *SetSpeed*. The neighbors with shorter distance to the sink can deliver packets to the sink with less hop counts, and they can reduce end-to-end delay of the packets. For better understanding, we use an example in Fig. 1. As shown in this figure, Nodes *B* and *C* are one-hop neighbors of node *A*. They also have the same *TwoHopSpeed* larger than *SetSpeed*. By considering only speed, these nodes have the same chance to be chosen as the forwarding node by node *A*. Also, node *B* can choose node *C* as its forwarding node since node *C* is inside the range of node *B*. By choosing nodes *B* and *C*, packet can send over two paths that are shown with white and black arrows. The path with white arrows has more hop counts. In other words, by selecting node *B* as the forwarding node, we increase the number of times that the packet must be forwarded and consequently end-to-end delay is increased. So, node *C* is better candidate than node *B* as it has less distance to the sink.

RTEA selects the next forwarding node by the following joint metric that considers end-to-end delay and energy balancing:

$$DE_j^i = W \times DelayMetric_j^i + (1 - W) \times EnergyMetric_j^i \tag{6}$$

where $W \in [0,1]$ is the weighting factor that indicates the importance of each metric in the joint forwarding metric. *DelayMetric* indicates the impact of selecting a neighbor as the next forwarding node on end-to-end delay of packets. This delay can be decreased by selecting nodes in each hop that can forward a packet to next hops with higher forwarding speed. Also, as mentioned before, the number of forwarding hops can be decreased by selecting neighbors as the next forwarding nodes that have shorter distance to sink.

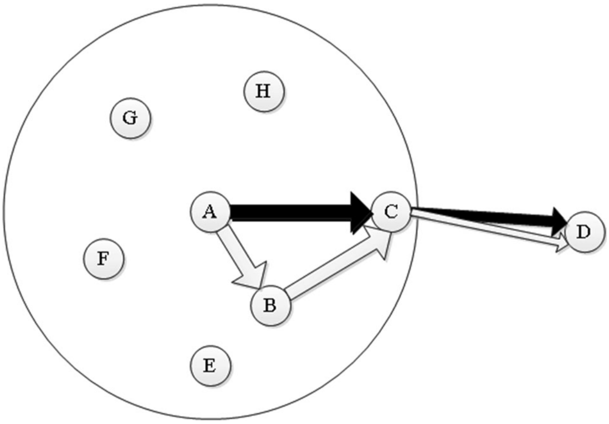


**Fig. 1** Effective fowarding

This leads to shorter end-to-end delay since packets spend some time in each node in the path for waiting in the queue, performing routing operations and successful forwarding to the next hop. So, *DelayMetric* can be defined as a joint metric that considers both *TwoHopSpeed* and distance to sink of each neighbor. This metric is defined as:

$$DelayMetric_j^i = \lambda \times \frac{TwoHopSpeed_j^i}{\sum_{k \in FCS(i)} TwoHopSpeed_k^i} + (1 - \lambda) \times \frac{Dis(i,d) - Dis(j,d)}{\sum_{k \in FCS(i)} (Dis(i,d) - Dis(k,d))} \tag{7}$$

where $\lambda \in [0,1]$ is the weighting factor for incorporating the distance to the sink in *DelayMetric*. It can be determined experimentally as a fixed coefficient. *EnergyMetric* is used for distributing traffic to the nodes that have higher energy level. So, larger *EnergyMetric* leads to better energy balancing. This metric is calculated by (7).

$$EnergyMetric_j^i = \frac{E_j / E_j^0}{\sum_{k \in FCS(i)} (E_k / E_k^0)} \tag{8}$$

By considering a fixed value for $W$ in (6), all the nodes treat the packets at the same manner. However, the importance of each metric must be changed based on the properties of each packet such as deadline and the distance to the sink. Energy balancing is more important for the nodes that are closer to the sink [31]. These nodes have higher load and consume more energy than the others. So, they died quickly which causes the network to be disconnected. In addition, by reducing deadlines of packets in each hop, their chance for on-time delivery is decreased and they must be sent with a higher speed.

$W$ must be defined as a joint metric to consider both remaining time to packet's deadline and average distance of neighbors to the sink. We assign a priority between zero and one for each one of these criteria which are shown by $\alpha$ and $\beta$.

$\alpha$ indicates the importance of end-to-end delay by considering the elapsed time of packets. It is calculated by (9).

$$\alpha = \frac{InitialDeadline - Deadline}{InitialDeadline} \tag{9}$$

*InitialDeadline* is the deadline of the packet which stamped on it in the source. Also, $\beta$ indicates the importance of energy balancing and is defined by (10) where $n$ is the number of final candidates in *FCS(i)*.

$$\beta = \frac{\sum_{j \in FCS(i)} Dis(i,j)}{n \times Dis(i,d)} \tag{10}$$

$W$ must be determined based on the comparison of $\alpha$ and $\beta$ and it must satisfy three requirements as follows:

1. The value of $W$ belong to [0,1].
2. $W = 0.5$ if $\alpha = \beta$.
3. $w > 0.5$ if $\alpha > \beta$ and vice versa.

Based on the aforementioned requirements, $W$ can be defined as:

$$W = \frac{\alpha}{\alpha + \beta} \tag{11}$$

by considering $W$ as a dynamic coefficient, we can achieve better energy balancing and we can prioritize packets based on their deadlines.

## 3.4 Delay Estimation and Updating Packet Deadline

*TwoHopSpeed* in (2) is calculated based on the estimated delay between neighbors. So, accuracy of delay estimation has a significant impact on the performance of RTEA. In addition to transmission delay, RTEA considers the time that a packet spends in each node. By increasing the load of each node, this time is increased and congested nodes have lower chance to be selected as the forwarding node. RTEA uses a QoS aware MAC layer which estimates delay of a packet from node $i$ to its one-hop neighbor $j$ by:

$$OneHopDelay_j^i = LinkDelay_j^i + NodeDelay_j \tag{12}$$

*LinkDelay* is the time required for transmitting a packet between two neighbors which includes transmission time and propagation delay. Transmission time is calculated by dividing the packet size on the data rate. Before sending a packet, node $i$ calculates this time and stamps it on the packet. $NodeDelay_j$ is the time that a data packet spends in node $j$ for waiting in the queue and routing operations. In other words, $NodeDelay_j$ is the interval between the times that the MAC layer receives a packet from the channel and the network layer. During this time, packet moves among MAC and network layers and RTEA determines the next forwarding node. Each node in the network has a *NodeDelay* and updates it every time that sends a packet. MAC layer uses a table for updating *NodeDelay* which is called delay table and it has has three columns, *PacketID*, *ReceivedTime*, and *SentTime*. When the MAC layer receives a packet, first adds a new entry in the delay table and inserts *PacketID* and *ReceivedTime*. Then it sends the packet to the network layer. After a forwarding node is determined, packet is sent to the MAC layer which searches *PacketID* in the delay table and calculates $NodeDelay_{new}$ by subtracting *ReceivedTime* from the current time. For updating *NodeDelay,* we employ Exponential Weighted Moving Average (EWMA) [32] as follows:

$$NodeDelay = \phi \times NodeDelay_{new} + (1 - \phi) \times NodeDelay \tag{13}$$

Li et al. [3] shows that $\phi = 0.5$ is the appropriate value. Every Node in IWSN, when receives a packet adds *LinkDelay* to *NodeDelay* and sends it with ACK to the receiver.

RTEA updates packet deadline by estimating the times that a packet spends in the node and the channel in each hop. When MAC layer wants to send a packet for the first time, it uses the delay table and updates the packet deadline by:

$$\begin{aligned} NewDeadline = Deadline - (CurrentTime - ReceivedTime) \\ - \ (TransmissionTime + PropagationDelay) \end{aligned} \tag{14}$$

Packets must be retransmitted due to collision. Automatic Repeat-Request (ARQ) is adopted for retransmitting packets up to 7 times. MAC layer updates packet deadlines for every retransmission by:

$$NewDeadline = Deadline - (CurrentTime - SentTime) \tag{15}$$

where *SentTime* shows the last time when the packet was sent.

### 3.5 Updating Maximum Speed

In RTEA, each node keeps *Max_Speed* and *Max_Speed_NodeID* as two properties in the network layer. These parameters are updated only when a node receives an ACK. *Max_Speed_NodeID* is the ID of a one-hop neighbor closer to the sink that the node can send a packet to it with *Max_Speed*. By receiving an ACK, RTEA updates the *EstDelay* and calculates *OneHopSpeed* related to the sender of ACK by (3). *Max_Speed* and *Max_Speed_NodeID* are updated in two cases:

1. Calculated *OneHopSpeed* is larger than $(1 + \varepsilon) \times Max\_Speed$. In this case, *Max_Speed* is updated with calculated *OneHopSpeed*.
2. Calculated *OneHopSpeed* is smaller than $(1 - \varepsilon) \times Max\_Speed$ and belongs to the neighbor that has *Max_Speed_NodeID*. In this case, *OneHopSpeed* is calculated for all the neighbors and *Max_Speed* is updated if the maximum *OneHopSpeed* is not between $(1 - \varepsilon) \times Max\_Speed$ and $(1 + \varepsilon) \times Max\_Speed$.

$\varepsilon$ is a threshold for updating *Max_Speed*. It has a small value lower than one which can be determined experimentally. As mentioned in part 3.1, after updating *Max_Speed*, a MAX-SPEED-PCKT is created and issued to inform upstream nodes.

### 3.6 Drop Control Unit

When no node in the candidate set can provide *TwoHopSpeed* larger than *SetSpeed*, drop control is conducted which decides whether or not to drop the packet. First, the neighbor with maximum *TwoHopSpeed* is chosen as the best candidate and forwarding probability $\psi$ is estimated by:

$$\psi = \frac{TwoHopSpeed}{SetSpeed} \tag{16}$$

If a randomly generated number is lower than $\psi$, the packet will be forwarded to the best candidate. Otherwise, packet is dropped to reduce the congestion.

### 3.7 Congestion and Void Avoidance

RTEA avoids the voids by broadcasting MAX-SPEED-PCKT with zero value. As shown in (1), RTEA selects only the neighbors as the next forwarding nodes that are closer to the sink and have *MaxSpeed* larger than zero. Nodes broadcast a MAX-SPEED-PCKT with zero value to inform upstream nodes of a void when they do not have any neighbors closer to the sink. So, they are not considered in routing operation.

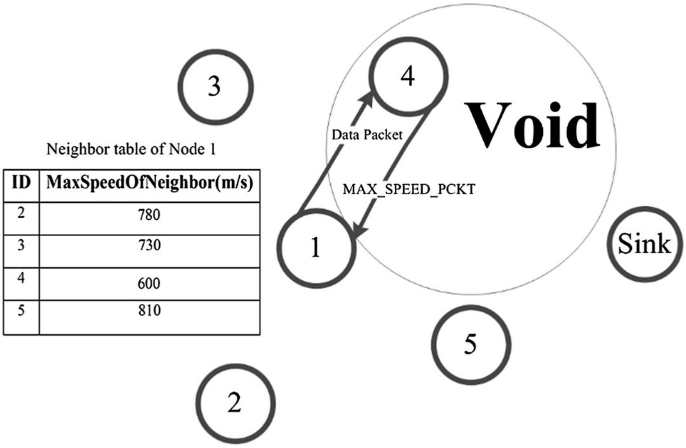
We use an example in Figs. 2 and 3 for better understanding. Node 1 has 4 on-hop neighbors. Among the neighbors, only nodes 4 and 5 are closer to the sink and they can be selected as the next forwarding node if they have $MaxSpeed > 0$. Node 4 is placed in the void because it does not have any one-hop neighbors closer to the sink. In Fig. 2, node1 is not aware of the void and it selects node 4 as the next forwarding node since it has $MaxSpeed > 0$ as shown in the node1's neighbor table. Node 4 receives the packet, drop it, and broadcast a MAX_SPEED_PCKT with zero value to inform its neighbors of the void.

In Fig. 3, node 1 receives MAX_SPEED_PCKT, update *MaxSpeed* of node 4 to zero in its neighbor table, and selects another neighbor as the next forwarding node. In this case, node 4 is not considered in the routing operations as it has $MaxSpeed = 0$ and its neighbors (nodes 1 and 3) can avoid the void. RTEA deals with the congestion as the same way. Nodes broadcast MAX-SPEED-PCKT when update their *Max_Speed.* So, every node in network is aware of situation of its neighbors and a congested node has less chance to be selected as the next forwarding node.

## 4 Performance Evaluation

We simulate RTEA on NS-2.34 simulator [33] and we compare its performance with THVR [3] and THVRG [19]. Simulation parameters are described in Table 2. We use IEEE 802.11 with 200 kb/s data rate as [2, 13, 20, 21]. The following criteria are evaluated in our simulation:

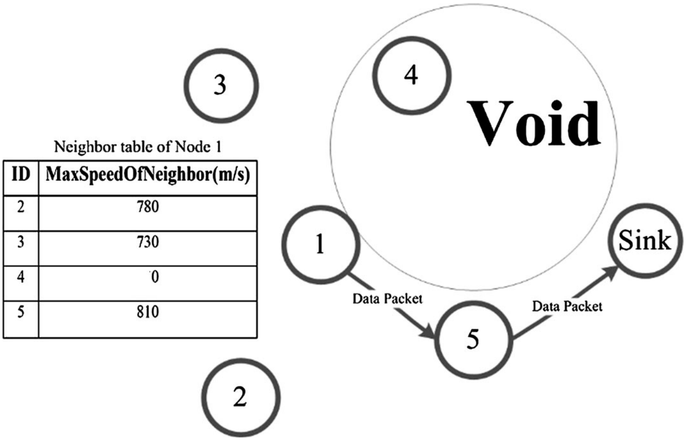
- Deadline Delivery Success Ratio (DDSR).
- End-to-end delay.



Neighbor table of Node 1

| ID | MaxSpeedOfNeighbor(m/s) |
|---|---|
| 2 | 780 |
| 3 | 730 |
| 4 | 600 |
| 5 | 810 |

**Fig. 2** Void avoidance scheme-case 1



Neighbor table of Node 1

| ID | MaxSpeedOfNeighbor(m/s) |
|---|---|
| 2 | 780 |
| 3 | 730 |
| 4 | 0 |
| 5 | 810 |

**Fig. 3** Void avoidance scheme-case 2

- Number of control packets.
- Number of hops.
- Energy consumption.
- Node Failure.

For evaluating performance with different network load, we increase the number of sources from 4 to 20. In addition, we consider two deadline requirements of 0.25 and 0.5 s as short and long deadlines. The value of $\lambda$ in (7) is set at 0.5. For updating *Max_Speed*, $\varepsilon$ is set at 0.05. In THVR, we set desired speed of 360 and 720 m/s for long and short deadlines. Sources are selected randomly in the left side of the terrain while the sink is located at (195, 190 m). Each source generates one CBR flow with the rate of 1 packet/s. All the experiments are repeated for 30 times with different sources and different random topologies. The average value is considered as the result of the simulation with 95% confidence interval.

We use an energy model to compare energy consumption of routing protocols. In this model, energy consumption of a node is expressed as the sum of consumed energy for packet transmitting ($E_{tx}$) and packet reception ($E_{rx}$). Thus, the total energy consumed is given by:

**Table 2** Simulation Parameters

| Routing | RTEA, THVRG, THVR |
|---|---|
| MAC layer | 802.11 |
| Propagation model | TWO-RAY GROUND |
| Data rate | 200 kb/s |
| Payload size | 32 byte |
| Terrain | (200, 200 m) |
| Number of nodes | 100 |
| Node placement | Uniform |
| Radio range | 40 m |
| Simulation time | 100 s |

$$\begin{aligned} E &= E_{tx} + E_{rx} \\ &= V \times (f \times I_{tx} \times T_{tx} + f \times I_{rx} \times T_{rx}) \end{aligned} \tag{17}$$

where $V$ is the voltage supply and $f$ is the packet size. $I_{tx}$ and $I_{rx}$ shows the current required during transmission, and reception. Also, $T_{tx}$ and $T_{rx}$ indicates the corresponding activity durations. The considered values of these parameters for 1 byte are shown in Table 3.

### 4.1 Deadline Delivery Success Ratio

Figure 4 shows the ratio of the packets that are delivered to the sink before their deadline. Letter D in the diagram's name, indicates the deadline in the experiment. As shown in this figure, RTEA outperforms THVR and THVRG because it updates desired speed of packets in each hop and selects the next forwarding node based on the distance to the sink besides the speed that can be provided in next two hops. When deadline is large, THVRG has better performance than THVR since it has lower control overhead and uses a dynamic weighting factor for incorporating energy level in the forwarding metric. By increasing the number of sources, the difference between routing protocols is increased. THVR sends control packets more than other routing protocols in the face of congestion that makes more congestion. In THVRG, forwarding candidate set has less members than THVR and RTEA, since it selects only the neighbors that have less hop counts to the sink. So, congestion occurs in less time due to high traffic load on the forwarding candidate neighbors. RTEA reduces congestion by dropping packets with zero deadlines and by calculating progress speed based on the time that packets spend in each node.

When deadline is short, THVR has better performance than THVRG as packets spend less time in the queue. The results show that all the protocols do not have good performance in face of severe congestion when deadline is short. Future works are needed to reduce the congestion and improve DDSR in hard real-time applications. Figure 5 Shows DDSR for 18 source nodes when deadline of packets is increased. As shown in this figure, RTEA outperform THVR and THVRG for all packets' deadline. Although, difference between protocols is reduced by increasing deadline. This shows that RTEA is a suitable routing candidate for applications that have strict timeliness requirements.

### 4.2 End-to-End Delay

The results for end-to-end delay are shown in Fig. 6. The end-to-end delay of a packet is defined as the time it takes to forward across the network from source to the sink. For the long deadline, RTEA has the shortest end-to-end delay since it prioritizes packets based on their deadline. While THVR and THVRG use the same desired speed for all the packets that have the same source. THVRG has the shortest end-to-end delay when deadline is long due to optimize the number of relaying hops. Also, it drops many packets in order to reduce congestion and it has low DDSR as shown in Fig. 4. THVR has the largest end-to-end

**Table 3** Energy Model Parameters

| | Transmission | Reception |
|---|---|---|
| Current (mA) | 154 | 85 |
| Activity duration (ms) | 0.04 | 0.04 |
| Voltage (V) | 3.3 | |

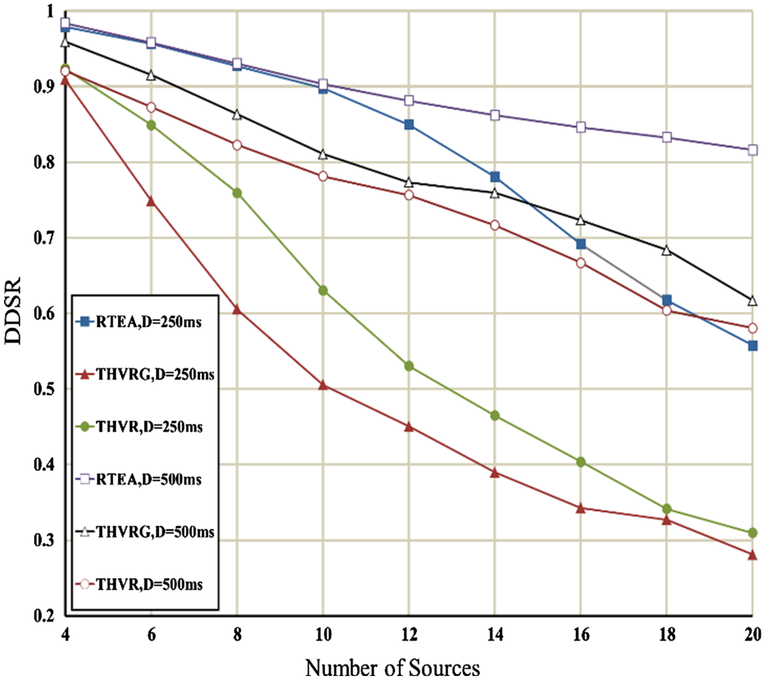


**Fig. 4** Deadline delivery success ratio versus number of sources

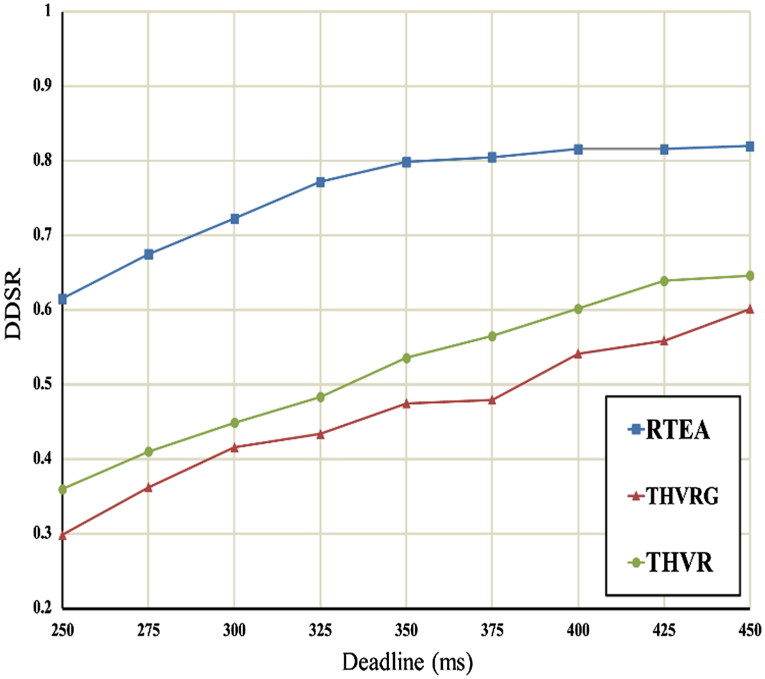


**Fig. 5** Deadline delivery success ratio versus packets' deadline

delay because of using too many control packets that increases collisions. Moreover, computation complexity in THVR is greater than RTEA and THVRG, which makes the packets spend more time for routing operations.

### 4.3 Number of Control Packets

The energy consumption of nodes is directly related to the number of control packets that they broadcast in the network. Also, collisions in the network are increased by sending more control packets. Figure 7 shows the number of control packets that nodes send them in the network. In all protocols, nodes use periodical beacons in order to inform their neighbors of their remaining energy. Moreover, they send different on-demand beacons. THVR has the highest control packet overhead because it uses two rounds of hello messages for exchanging geographical information between two-hop neighbors. Also, In THVR, every time that a node receives an ACK, It updates its neighbor table and sends a packet to inform upstream nodes of the new delay. RTEA has control packet overhead less than THVRG because it sends MAX-SPEED-PCKT only when Max_Speed of a node is changed or to inform other nodes of a void. While in THVRG, a node sends a feedback packet containing the forwarding link delay to its neighbors when its optimal forwarding link delay is changed. The optimal forwarder is determined by calculating a joint metric for all the neighbors. THVRG calculates this metric based on the packet's residual time to meet the deadline and the number of hops from the source of the packet to the sink. So, the optimal forwarder can be different for each data packet. Also, in the setup phase, each node forwards packets that contain the smallest number of hops to the sink.

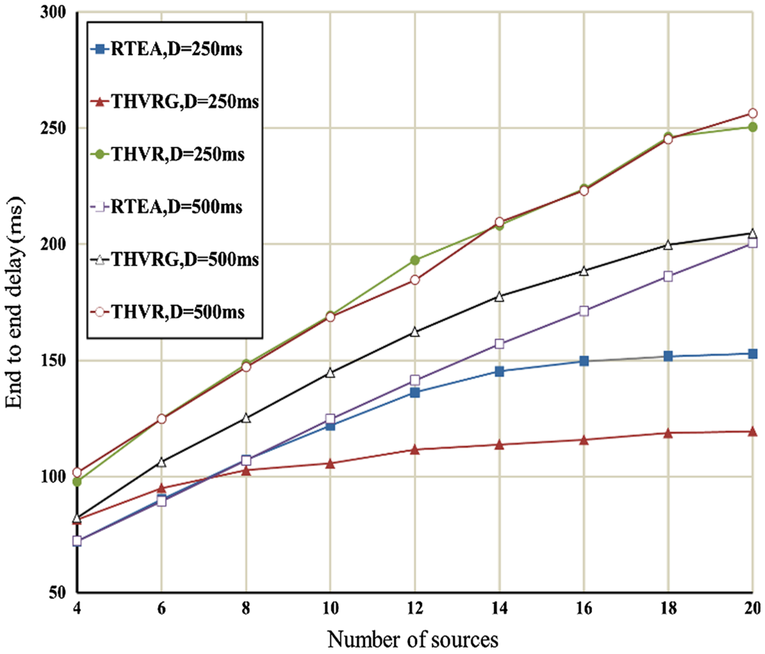


**Fig. 6** End-to-end delay versus number of sources

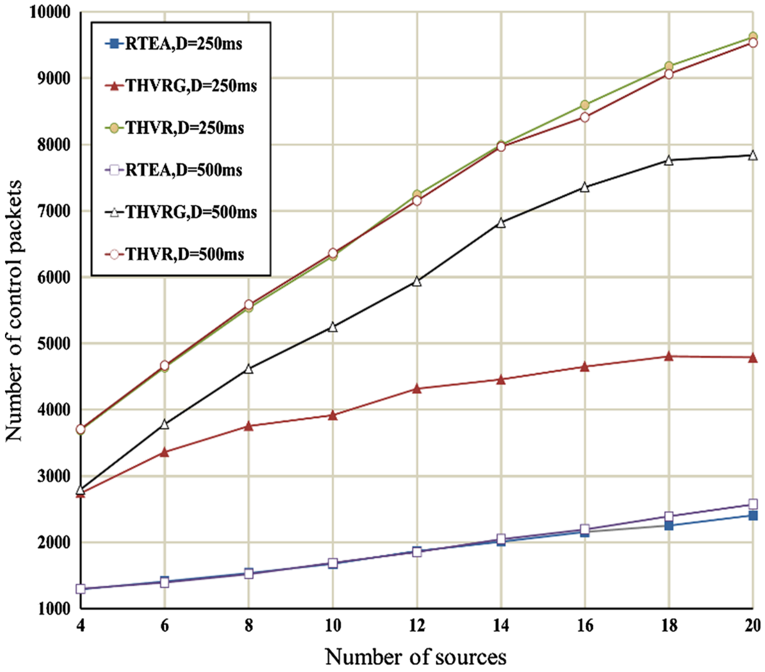


**Fig. 7** Number of control packets versus number of sources

## 4.4 Number of Hops

Figure 8 shows the average forwarding hop counts that data packets are forwarded between the source and the sink. In addition to control overhead, forwarding hop counts is directly related to the energy consumption. THVR outperforms RTEA due to use two hop neighborhood information that contains the link delay and the geographical location of two-hop neighbors. However RTEA and THVR deliver data packets to the sink with only two and one hop more than THVRG in our simulation scenario with 100 nodes.

## 4.5 Energy Consumption

The average used energy of nodes is plotted in Fig. 9. The increase in energy consumption is resulted by the increased control packets, data packets, forwarding hop counts and retransmissions for data packets due to collision and channel busy probability. When deadline is large, RTEA has the least energy consumption because it has the low control packet overhead as shown in Fig. 7. THVR used energy more than other routing protocols since it sends too many control packets that leads to network congestion. In the experiments with short deadline, THVRG shows better results than the other routing protocols. The reason is that THVRG delivers data packets to the sink with the least forwarding hop counts. Also, it has low DDSR as shown in Fig. 4 and drops too many data packets for missing the deadline. For comparing energy efficiency of protocols, the energy used per successfully delivered packet is shown in Fig. 10. RTEA outperforms THVR and THVRG since it has the highest DDSR as indicated in Fig. 4. RTEA used energy more than THVRG only when deadline is short and THVRG has low DDSR.

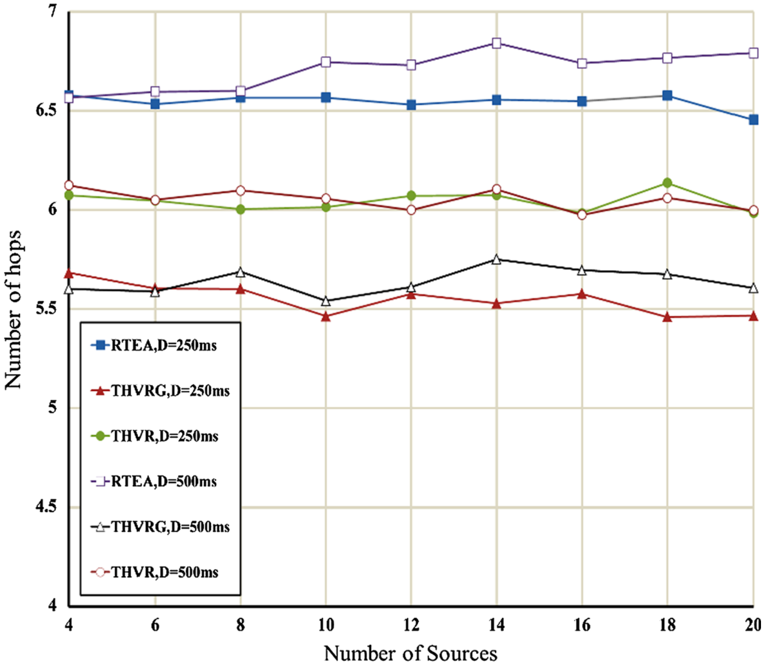


**Fig. 8** Number of forwarding hop counts versus number of sources

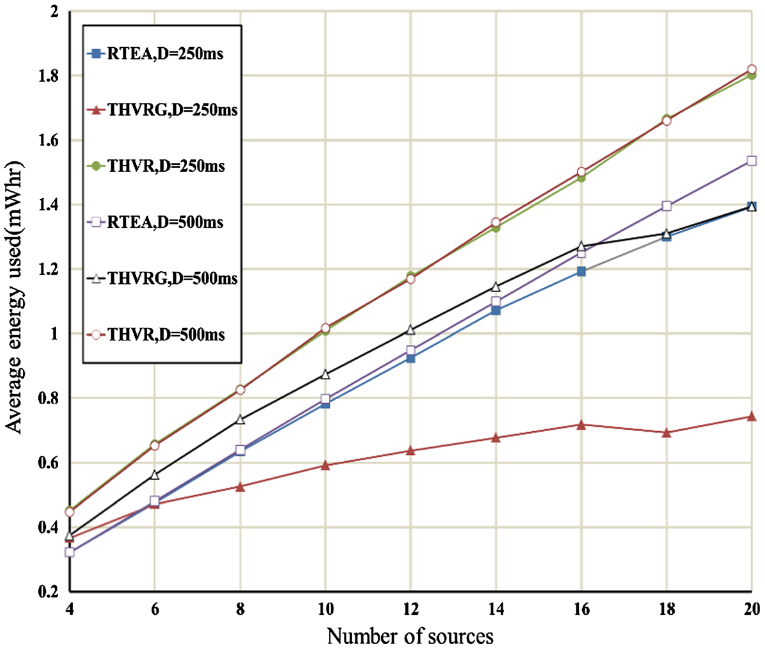


**Fig. 9** Average energy used versus number of sources

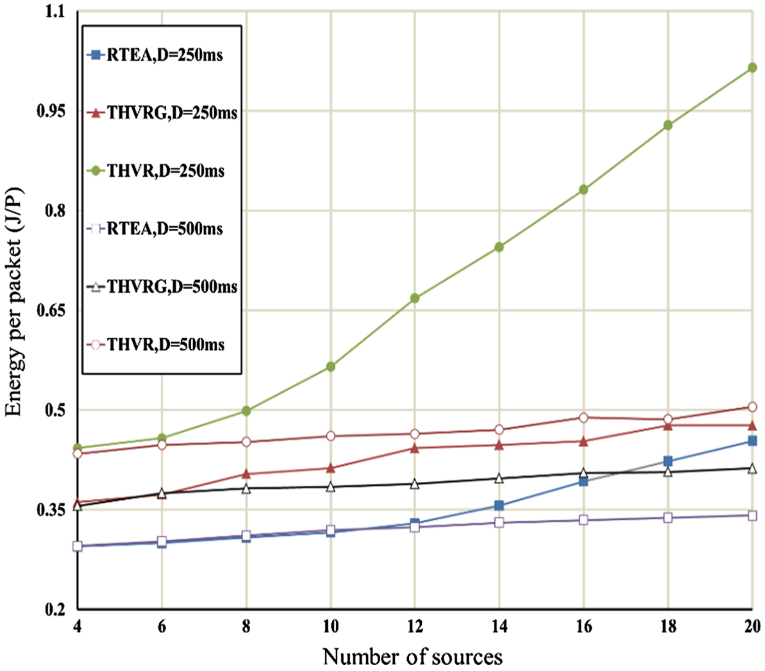


**Fig. 10** Energy consumed per successfully transmitted packet versus number of sources

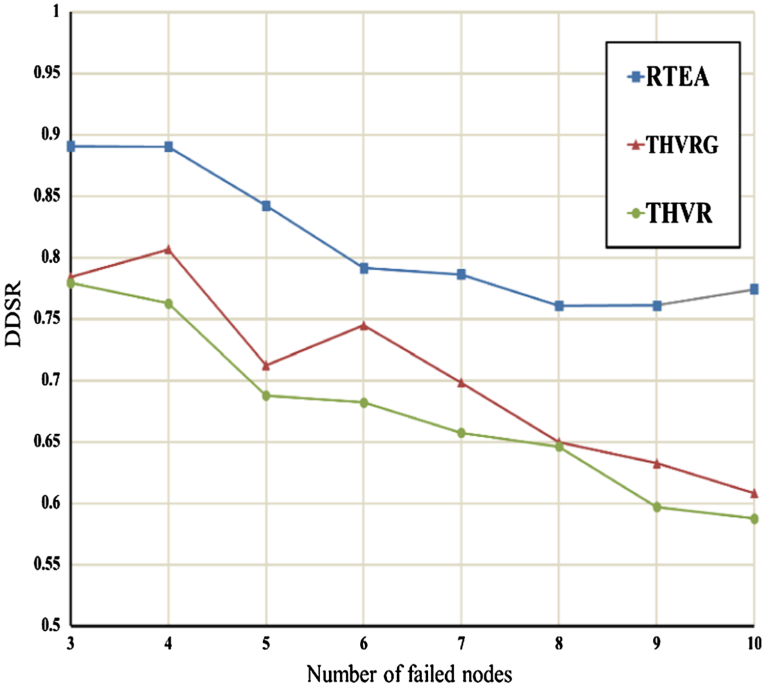


**Fig. 11** DDSR versus number of fail nodes

### 4.6 Node Failure

Figure 11 shows the performance of protocols in the face of node failure. 0.5 is considered as the packets' deadline and 3–10 nodes are failed for 40 s in experiments. As shown in this figure, RTEA has better performance since it avoids the voids by broadcasting MAX-SPEED-PCKTs with zero value. Moreover, in RTEA, each node broadcasts MAX-SPEED-PCKTs only when its *Max_Speed* is changed which makes number of control packets reduced. THVR has the lowest performance as it sends too many control packets in the face of congestion.

## 5 Conclusion

In this paper, a real-time routing protocol is proposed for IWSN, called RTEA. It maps packet deadline to the velocity and uses two-hop neighborhood information with low control overhead. It makes routing decisions based on the velocity, the distance to the sink and the remaining energy of neighbors. Deadlines of packets are updated in each hop without using globally synchronized clocks. When no node can provide the desired velocity, RTEA employs drop control unit and drops the packet with the probability which is proper to maximum speed that one-hop neighbors can send data packets to the sink. Simulation results show that RTEA improves performance in terms of delivery ratio and energy consumption.

## References

1. Gungor, V. C., & Hancke, G. P. (2009). Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transaction on Industrial Electronics, 56*(10), 4258–4265.
2. Heo, J., Hong, J., & Cho, Y. (2009). EARQ: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks. *IEEE Transaction on Industrial Informatics, 5*(1), 3–11.
3. Li, Y., Chen, C. S., Song, Y. Q., Wang, Z., & Sun, Y. (2009). Enhancing real-time delivery in wireless sensor networks with two-hop information. *IEEE Transaction on Industrial Informatics, 5*(2), 113–122.
4. Krishna, M. B., & Doja, M. N. (2013). Analysis of tree-based multicast routing in wireless sensor networks with varying network metrics. *International Journal of Communication Systems, 26,* 1327–1340.
5. Kułakowski, P., Calle, E., & Marzo, J. L. (2013). Performance study of wireless sensor and actuator networks in forest fire scenarios. *International Journal of Communication Systems, 26,* 515–529.
6. Bijwaard, D. J. A., Kleunen, W. A. P., Havinga, P. J. M., Kleiboer, L., & Bijl, M. J. J. (2011). Industry: using dynamic WSNs in smart logistics for fruits and pharmacy. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*.
7. Rice, J. A., Mechitov, K., Sim, S., Nagayama, T., Jang, S., Kim, R., et al. (2010). Flexible smart sensor framework for autonomous structural health monitoring. *Smart Structures and Systems, 6,* 423.
8. Yu, Y., Ou, J., & Li, H. (2010). Design, calibration and application of wireless sensors for structural global and local monitoring of civil infrastructures. *Smart Structures and Systems, 6,* 641.
9. Pakzad, S. N. (2010). Development and deployment of large scale wireless sensor network on a long-span bridge. *Smart Structures and Systems, 6,* 525.
10. Yu, Y., Ou, J., Zhang, J., Zhang, C., & Li, L. (2009). Development of wireless MEMS inclination sensor system for swing monitoring of large-scale hook structures. *IEEE Transaction on Industrial Informatics, 56*(4), 1072.
11. Misra, S., Woungang, I., & Mirsa, S. C. (2009). Quality of service in wireless sensor networks. In S. Misra, I. Zhang & S. C. Misra (Eds.), *Guide to wireless sensor networks* (pp. 305–321). Springer.

12. Mahapatra, A., Anand, K., & Agrawal, D. P. (2006). QoS and energy aware routing for real-time traffic in wireless sensor networks. *Computer Communications, 29*(4), 437–445.
13. Ben-Othman, J., & Yahya, B. (2010). Energy efficient and QoS based routing protocol for wireless sensor networks. *Journal of Parallel and Distributed Computing, 70*(8), 849–857.
14. Liu, M., Xu, S., & Sun, S. (2012). An agent-assisted QoS-based routing algorithm for wireless sensor networks. *Journal of Network and Computer Applications, 35*(1), 29–36.
15. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks, 38*(4), 393–422.
16. Liu, C.-X., Liu, Y., Zhang, Z.-J., & Cheng, Z.-Y. (2013). High energy-efficient and privacy-preserving secure data aggregation for wireless sensor networks. *International Journal of Communication Systems, 26,* 380–394.
17. Huang, H., Hu, G., & Yu, F. (2013). Energy-aware geographic routing in wireless sensor networks with anchor nodes. *International Journal of Communication Systems, 26,* 100–113.
18. Liu, G., Xu, B., & Chen, H. (2014). An indicator kriging method for distributed estimation in wireless sensor networks. *International Journal of Communication Systems, 27,* 68–80.
19. Quang, P. T. A., & Kim, D. (2012). Enhancing real-time delivery of gradient routing for industrial wireless sensor networks. *IEEE Transaction on Industrial Informatics, 8*(1), 61–68.
20. He, T., Stankovic, J., Lu, C., & Abdelzaher, T. (2005). A spatiotemporal communication protocol for wireless sensor networks. *IEEE Transaction on Parallel And Distributed Systems, 16*(10), 995–1006.
21. Felemban, E., Lee, C. G., & Ekici, E. (2006). MMSPEED: Multipath multispeed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks. *IEEE Transaction on Mobile Computing, 5*(6), 738–754.
22. Jung, J., Park, S., Lee, E., Oh, S., & Kim, S.H. (2010). OMLRP: Multi-hop information based real-time routing protocol in wireless sensor networks. In *Wireless Communications and Networking Conference (WCNC).*
23. Koulali, M.A., Kobbane, A., Elkoutbi, M., & Azizi, M. (2010). A QoS-geographic and energy aware routing protocol for wireless sensor networks. In *5th International Symposium on I/V Communications and Mobile Network.*
24. Li, Y., Chen, C.S., Song, Y.Q., & Wang, Z. (2007). Real-time QoS support in wireless sensor networks: A survey. In *7th IFAC international conference on fieldbuses & networks in industrial & embedded systems*.
25. Yuan, L., Cheng, W., & Du, X. (2007). An energy-efficient real-time routing protocol for sensor networks. *Computer Communications, 30*(10), 2274–2283.
26. Ryu, J., Lee, C., Kwon, T., & Han, J. (2009). Combined scheduling and routing for deterministic guarantee of end-to-end deadlines in cell structured sensor networks. *IEEE Sensors Journal, 9*(10), 1291–1301.
27. Zhixin, L., Lili, D., Liang, X., Xinping, G., & Changchun, H. (2011). Reliability considered routing protocol in wireless sensor networks. In *Control Conference (CCC)*, (pp. 5011–5016).
28. Woo, A., Tong, T., & Culler, D. (2003). Taming the underlying challenges of reliable multihop routing in sensor networks. In *ACM, Proceedings of the 1st international conference on Embedded networked sensor systems*, Los Angeles, California.
29. Stojmenovic, I. (2005). Introduction to wireless sensor networking. In I. Stojmenović (Ed.), *Handbook of sensor networks: Algorithms and architectures* (pp. 1–41). Wiley.
30. IEEE Publications. (2006). Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE P802.15.4aD/4. http://standards.ieee.org.
31. Liu, X., & Mohapatra, P. (2007). On the deployment of wireless data back-haul networks. *IEEE Transaction on Wireless Communications, 6*(4), 1426–1435.
32. Kurose, J. F., & Ross, K. W. (2012). Transport layer. In M. Hirsch (Ed.), *Computer networking a top-down approach featuring the internet* (pp. 185–303). Addison Wesley Longman Inc.
33. Issariyakul, T., & Hossain, E. (2009). *Introduction to network simulator NS2.* Berlin: Springer.

**Omid Tavallaie** received his B.Sc. degree in computer engineering from University of Science and Culture, Tehran, Iran, and the M.Sc. degree in Information Technology from Graduate University of Advanced Technology, Kerman, Iran in 2010 and 2013 respectively. During his M.Sc., he was working on real-time communication for wireless networks including adhoc and sensor networks. Omid is currently PhD student of Information Technology at the University of Sydney, Australia and his field of study is programming languages and networking. His research interests are wireless networks, real-time applications, and IP multimedia subsystems.

**Hamid Reza Naji** is Associate Professor of Computer Engineering at, Graduate University of Advanced Technology, Kerman, Iran. His research interests include embedded, reconfigurable, and multi-agent systems, networks, and security. Dr. Naji has a PhD in computer engineering from university of Alabama in Huntsville, USA. He is a professional member of the IEEE.

**Masoud Sabaei** received his B.Sc. degree in Computer Engineering from Isfahan University of Technology, Isfahan, Iran in 1992, and his M.Sc. and Ph.D. degrees from Amirkabir University of Technology (AUT), Tehran, Iran in 1995, and 2001, respectively. He joined the Computer Engineering and Information Technology (CEIT) Department of Amirkabir University of Technology in 2004. Currently, as an assistant professor, he is the director of the Mobile Ad hoc and Wireless Sensor Lab at AUT. His research interests include high speed networks, network management, mobile ad hoc and sensor networks, Opportunistic Network.

**Narges Arastouie** received her B.Sc. and M.Sc. degree in Computer Engineering from Amirkabir University of Technology (AUT), Tehran, Iran in 2007, and 2009, respectively. She has been studying as a Ph.D. candidate in the High Speed Wireless Sensor Network Lab at AUT. Her research interests include mobile ad hoc and sensor networks, opportunistic network, delay tolerant network and VANET.