

# A hybrid beaconless geographic routing for different packets in WSN

Chao Hong<sup>1</sup> · Zhongyang Xiong<sup>1</sup> · Yufang Zhang<sup>1</sup>

Published online: 18 July 2015

© Springer Science+Business Media New York 2015

**Abstract** The localized operation and stateless features of geographic routing make it become an attractive routing scheme for wireless sensor network (WSN). In this paper, we proposed a novel routing protocol, hybrid beaconless geographic routing (HBGR), which provides different mechanisms for different packets. Based on the requirement of application on latency, we divide the packets of WSN into delay sensitive packets and normal packets. HBGR uses two kinds of Request-To-Send/Clear-To-Send handshaking mechanisms for delay sensitive packets and normal packets, and assigns them different priority to obtain the channel. The simplified analysis is given, which proves that delay sensitive packets have lower latency and higher priority to obtain the channel than normal packets. Moreover, forwarding area division scheme is proposed to optimize the forwarder selection. Simulation results show that HBGR achieves higher packet delivery ratio, lower End-to-End latency and lower energy consumption than existing protocols under different packet generation rates in stationary and mobility scenario. Besides, compared with normal packets, delay sensitive packets have at least 10 % (9 %) improvement in terms of End-to-End latency. The improvement increases with the increasing of packet

generation rate, and achieves 58 % (73 %) when the packet generation rate is 24 packets per second in stationary (mobility) scenario.

**Keywords** WSN · Geographic routing · RTS/CTS handshaking · Forwarding area division · Channel contention

## 1 Introduction

Geographic routing protocols have received attention as a routing solution for wireless sensor network (WSN). Different from topology-based routing, geographic routing exploits the geographic information instead of topological connectivity information to move data packets to gradually approach and eventually reach the intended destination [1]. One of the most interesting features of geographic routing protocol is that it can operate without any routing tables, even without neighbor tables. Moreover, once the position of the destination is known, all operations are strictly local [2]. The localized operation and stateless features of geographic routing make it simple and scalable. Generally, sensing devices share common features, such as constrained energy resources, limited processing capability, vulnerable radio conditions, the real time nature of applications, and minimal direct human interaction [3]. Therefore, simplicity is essential for routing protocols in WSN. Moreover, as pointed in [4], experiences from real system implementations, not limited to WSNs, teach us that complicatedly integrated systems are prone to losing the attributes of scalability and efficiency. On the contrary, the principle with a simplified design is usually applied in practice. In addition, WSNs are envisioned to exist at tremendous scale, with possibly thousands or millions of

---

✉ Chao Hong  
hongchao8705@163.com

Zhongyang Xiong  
zyxiong@cqu.edu.cn

Yufang Zhang  
zhangyf@cqu.edu.cn

<sup>1</sup> College of Computer Science, Chongqing University, Chongqing 400044, China

nodes per network [5]. Thus, scalability is a significant factor for WSN.

Geographic routing is a routing scheme in which each sensor node is assumed to be aware of its geographic location using the Global Positioning System (GPS) or distributed localization services [6]. According to whether or not a node maintains the position information of its neighboring nodes, geographic routing protocols can be further classified as follows:

- *Beacon-based protocol* [6–12]: Each node periodically broadcasts a beacon message to its 1-hop neighbors to advertise its location information. By gathering all the beacon messages from its neighbors, a node could build a local knowledge about its neighbors, and it selects a forwarder using the local knowledge when it has a packet to send [7].
- *Beaconless protocol* [13–20]: Beaconless protocols operate reactively. Each node does not need to maintain any information of its neighbors. This kind of protocols usually uses Request-To-Send/Clear-To-Send (RTS/CTS) handshaking to select a forwarder only when a node needs to send. Therefore, they are purely on-demand routing protocols.

Although the beacon-based method may reduce the End-to-End latency when delivering a packet from the source to the Sink, it will increase the overhead for maintaining neighbor tables. Therefore, beaconless method is more robust than beacon-based method in dynamically changing environment. In this paper, we also use beaconless method to design our protocol.

A collection of small low-cost wireless nodes, which forms a WSN, will collaborate for some common applications such as intrusion detection, target tracking, environmental monitoring, smart building and so on. Some different kinds of sensors, which are used for sensing different data for different applications at the same time, could be integrated into one node. However, different applications may have different requirements on End-to-End latency. For instance, the packets of intrusion detection are generally more sensitive on latency than the packets of environmental monitoring. In this paper, we divide the packets of WSN into *delay sensitive packets* and *normal packets*.

With the goal of introducing a protocol that accelerates the delivery of delay sensitive packets with little impact on normal packets, we propose a novel geographic routing protocol, *hybrid beaconless geographic routing (HBGR)*. Firstly, HBGR uses two kinds of RTS/CTS handshaking mechanisms for delay sensitive packets and normal packets respectively. These make delay sensitive packets could deliver fast when the delay sensitive packets have obtained the channel. Moreover, with respect to channel contention,

HBGR also ensures delay sensitive packets are prior to normal packets. Meanwhile, forwarding area division scheme is introduced to optimize the forwarder selection that ensures the candidate forwarders, which are closer to the Sink, have higher opportunity to be selected as forwarder.

The rest of this paper is structured as follows: The related work on geographic routing is discussed in Sect. 2. The HBGR is presented in Sect. 3. In Sect. 4, we evaluate HBGR through NS-2 and present the comparisons with other protocols. Finally, we conclude the paper in Sect. 5.

## 2 Related work

In this section, we briefly review some geographic routing protocols proposed in the literature.

### 2.1 RTS/CTS handshaking

As mentioned above, geographic routing protocols can be further classified as either beacon-based protocol or beaconless protocol. Almost all of beaconless protocols use RTS/CTS handshaking mechanism to select the forwarder, including IGF [13], OGF [14], ALBA [15], Auction-based RSA [17], GeRaF [18, 19], EBGR [20], etc.

IGF is pure stateless protocol. It provides robust and effective communication in the highly dynamic network and performs excellent in dense topologies. In order to prevent interference among candidate forwarders, IGF shrinks the forwarding area to ensure that every node within the forwarding area is capable of hearing one another. This would make the performance of IGF degenerate in sparse topologies.

OGF is an on-demand and beaconless protocol. OGF uses RTS/CTS handshaking mechanism, called RTF/CTF in the paper, to select forwarder. OGF also uses a forwarding table to record the learned forwarder information. Although it could improve forwarder selection in stable network, it may have some negative impact in highly dynamic network since the forwarding table would out of date quickly. With the purpose of preventing interference among candidate forwarders, OGF adjusts the carrier sensing range to be at least twice the transmission range. However, the expansion of carrier sensing range would increase interference to the neighboring nodes and reduce the utilization of channel resource. And PSR is proposed to handle the local-minimum problem.

However, all protocols mentioned above use a single RTS/CTS handshaking mechanism for all packets. In this paper, HBGR uses two kinds of RTS/CTS handshaking mechanisms to serve for different kinds of packets in

order to improve the latency of delay sensitive packets while ensure the impact on normal packets as little as possible. Moreover, HBGR also ensures delay sensitive packets have higher priority than normal packets to obtain the channel.

## 2.2 Backoff time and forwarding area division

The RTS/CTS handshaking mechanism operates as following: the sender will send a RTS when a node has a packet to send; candidate forwarders, which receive this RTS and could provide positive progress, will backoff a certain time before sending the CTS; the forwarder is the first one to send out the CTS.

In order to select an optimal forwarder for the sender, IGF [13] and OGF [14] use the weighted average method to calculate the backoff time of CTS. However, it is very difficult to determine the appropriate weight of each metric.

GeRaF [18, 19] divides the forwarding area into several circular slices which are centered at the Sink. The sender issues RTS or CONTINUE (special RTS) message to search the subareas one by one for the forwarder. The nodes in the farthest nonempty subareas respond first with a CTS message. If more than one report back, GeRaF handles this collision through binary splitting. If only one node sends CTS, this node becomes forwarder. GeRaF ensures that packets are transmitted along the approximate shortest path. However since the nodes in the same subarea respond with a CTS message at the same time, the collision would increase with the increasing of node density.

ALBA [15] strives to channelize traffic toward non-congested network regions, rather than just maximizing the advancement towards the final destination. In addition to dividing the forwarding area into several subareas and using GPI to label them, ALBA takes congestion into consideration and introduces QPI to reflect the degree of congestion. The sender issues RTS messages to scan the QPI in increasing order. If more than one node answers the RTS message soliciting a given QPI, it starts to scan the GPI, just as GeRaF scans geographic subareas. ALBA could effectively balance the traffic load among the different nodes, thus reducing congestion. However, frequent collision during the forwarder selection would degrade performance.

Similar to GeRaF and ALBA, HBGR also divides forwarding area into several subareas. However, HBGR does not let the nodes in the same subarea send CTS at the same time, HBGR assigns different contention window (CW)

sizes to different subareas based on forwarding progress they can make, and nodes in different subareas randomly pick the backoff time in their contention window to send CTS after receiving a RTS.

## 2.3 Other related works

Geographic routing mainly relies on an extremely simple geographic greedy-forwarding strategy at each hop to move a data packet to a locally optimal forwarder with a positive progress towards the destination node. However, greedy forwarding may not always be possible. A sender may fail to locate a forwarder in its neighborhood which has a positive geographic progress towards the destination node. This undesirable phenomenon called communications void or local-minimum [1]. To overcome this problem, many techniques are proposed. GPSR [8] combines planarization and right-hand rule which called Perimeter Routing to conquer communications void. GDSTR [9] uses hull-tree to route around voids. PAGER-M [10] and GEAR [11] are cost-based void handling techniques. They ensure each node has at least one neighbor with a smaller cost by adjusting the cost of node. However, the above-mentioned techniques can only be used for beacon-based protocol. PSR of OGF which is flooding-based techniques and Rainbow mechanism of ALBA-R [16] which introduces a label to indicate the forwarding direction could be used for beaconless protocol. Although we don't propose a new technique to conquer local-minimum in this paper, HBGR can combine PSR or Rainbow mechanism to remedy this problem.

More related work could be found in [21–32]. The papers in [21–23] are devoted to Data Aggregation which process raw data at the sensor nodes or at intermediate nodes to reduce packet transmissions and save energy. Meng et al. [24] exploit spatial reusability of the wireless communication media to improve throughput. CodePipe [25, 26] is an opportunistic feeding and routing protocol which offer significant improvement in throughput, energy-efficiency and fairness for reliable multicast. Cheng et al. [27] propose an approximate optimized channel assignment strategy to minimize the overall network interference while preserving the original topology for multi-radio wireless network. Physarum Optimization [28, 29] is a biology-inspired algorithm which could be used for improving the quality of coverage in WSN. Inspired by open vehicle routing (OVR) problems, Yao et al. [30, 31] propose EDAL to generate routes that connect all source nodes with minimal total path cost. Multi-layer clustering routing algorithm is proposed in [32] to mitigate the hot spot problem.

### 3 Protocol description of HBGR

#### 3.1 RTS/CTS handshaking mechanism

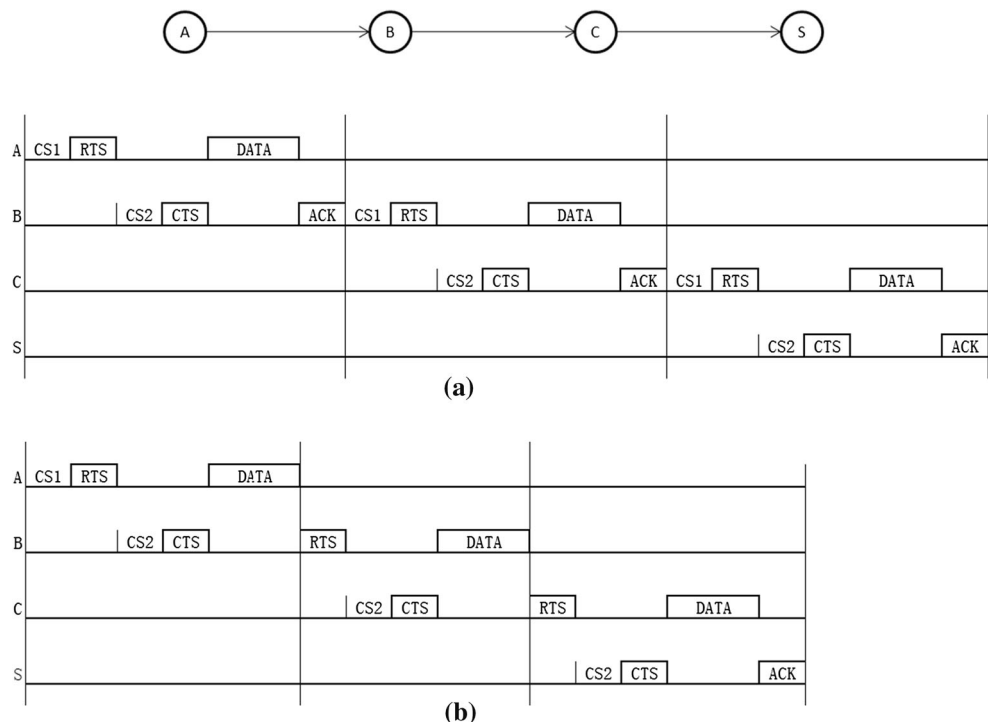
Most beaconless geographic routing protocols use traditional RTS/CTS handshaking to select forwarder. Once a node (sender) has a data packet to send, it begins carrier sense until channel is clear for a certain time, it broadcasts a RTS with its own position information and the location information of the Sink. Then, candidate forwarders, which received the RTS and were closer to Sink than the sender, contend to be a forwarder in a distributed manner. Using the progress as well as other information such as residue energy, each candidate forwarder calculates the backoff time to respond the sender. The first candidate forwarder to respond with a CTS becomes forwarder. All of candidate forwarders continue carrier sense before their backoff time expires. Meanwhile, whenever a transmission carrier is sensed, they conclude that another node with higher priority has sent out its CTS. Therefore, they cancel their CTS responses and quit the contention. After that, sender transmits DATA to the forwarder. When DATA transmission is finished, the forwarder sends ACK to sender. The above process is repeated until the packet reaches the Sink. As shown in Fig. 1(a).

RTS/CTS handshaking with implied acknowledgement can be used to accelerate packets' transmission. When DATA transmission is finished, the channel nearby

the forwarder is clear. Instead of sending ACK, the forwarder sends RTS to request the next forwarder and this RTS also as an acknowledgement (ACK) to sender by containing sender's ID. As shown in Fig. 1(b), when Node B has received DATA from Node A, it will send RTS as an acknowledgement to Node A and also as a request to search forwarder for itself. RTS/CTS handshaking with implied acknowledgement would save not only the time of ACK, but also the time of carrier sense for RTS since it avoids the channel from being preempted by other nodes.

HBGR uses traditional RTS/CTS handshaking to transmit normal packets, and uses RTS/CTS handshaking with implied acknowledgement for delay sensitive packets. Convergecast is a representative traffic pattern of WSNs, due to the need for collecting sensory data at a few Sinks [33]. Therefore the traffic was jammed near the Sink. In order to ensure delay sensitive packets have higher priority than normal packets in the interface queue of node. HBGR also sets two queues in each node, Q1 and Q2, for delay sensitive packets and normal packets, respectively. Q1 is prior to Q2. Whenever a node receives a DATA, it will be inserted into Q1 or Q2. After that the node checks Q1, the node will send ACK if Q1 is empty, otherwise a new RTS will be sent. The node could continue occupying the channel even though the node receives a normal packet if the Q1 is not empty.

**Fig. 1** The example of RTS/CTS handshaking mechanism: Delivering a packet from Node A, pass through Node B and Node C, eventually reach the Sink by using two different kinds of RTS/CTS handshaking mechanisms. **a** Traditional RTS/CTS handshaking, **b** RTS/CTS handshaking with implied acknowledgement



### 3.2 Latency analysis of RTS/CTS handshaking mechanism

We analyze the latency of traditional RTS/CTS handshaking and RTS/CTS handshaking with implied acknowledgement in the simple case that the traffic is very light, e.g., only one packet is moving through the network.

Suppose there are  $N$  hops from the source to the Sink. The carrier sense delay for RTS and CTS is random at each hop, and we denote their value at hop  $n$  by  $T_{CS\_RTS,n}$  and  $T_{CS\_CTS,n}$ , respectively. Their mean value are determined by the contention window (CW) size, and are denoted by  $T_{CW\_RTS}/2$  and  $T_{CW\_CTS}/2$ . And let  $T_{RTS}$ ,  $T_{CTS}$ ,  $T_{DATA}$  and  $T_{ACK}$  be the transmission delay of RTS, CTS, DATA and ACK, respectively.

We first look at the latency of traditional RTS/CTS handshaking which uses RTS/CTS/DATA/ACK session at the each hop. The latency at hop  $n$  is

$$D_n = T_{CS\_RTS,n} + T_{RTS} + T_{CS\_CTS,n} + T_{CTS} + T_{DATA} + T_{ACK} \tag{1}$$

The entire latency over  $N$  hops is

$$D(N) = \sum_{n=1}^N D_n = N(T_{DATA} + T_{RTS} + T_{CTS} + T_{ACK}) + \sum_{n=1}^N (T_{CS\_RTS,n} + T_{CS\_CTS,n}) \tag{2}$$

So the expected latency over  $N$  hops in traditional RTS/CTS handshaking is

$$E[D(N)] = N(T_{DATA} + T_{RTS} + T_{CTS} + T_{ACK}) + \sum_{n=1}^N E[T_{CS\_RTS,n} + T_{CS\_CTS,n}] = N(T_{DATA} + T_{RTS} + T_{CTS} + T_{ACK}) + \frac{N(T_{CW\_RTS} + T_{CW\_CTS})}{2} \tag{3}$$

Now we look at RTS/CTS handshaking with implied acknowledgement for delay sensitive packets. In this case, only the source node of packet needs carrier sense before sending RTS, and intermediate nodes send RTS by containing previous node's ID to request next forwarder and acknowledge previous forwarder (source), and only Sink (last node) needs to send ACK. The latency at hop  $n$  is

$$\begin{aligned} D'_1 &= T_{CS\_RTS,1} + T_{RTS} + T_{CS\_CTS,1} + T_{CTS} + T_{DATA} \\ D'_n &= T_{RTS} + T_{CS\_CTS,n} + T_{CTS} + T_{DATA} \quad (1 < n < N) \\ D'_N &= T_{RTS} + T_{CS\_CTS,N} + T_{CTS} + T_{DATA} + T_{ACK} \end{aligned} \tag{4}$$

The entire latency over  $N$  hops is

$$\begin{aligned} D'(N) &= \sum_{n=1}^N D'_n = D'_1 + \sum_{n=2}^{N-1} D'_n + D'_N \\ &= N(T_{DATA} + T_{RTS} + T_{CTS}) + T_{ACK} + T_{CS\_RTS,1} \\ &\quad + \sum_{n=1}^N T_{CS\_CTS,n} \end{aligned} \tag{5}$$

Hence, the expected latency over  $N$  hops in RTS/CTS handshaking with implied acknowledgement for delay sensitive packets is

$$E[D'(N)] = N(T_{DATA} + T_{RTS} + T_{CTS}) + T_{ACK} + \frac{T_{CW\_RTS} + NT_{CW\_CTS}}{2} \tag{6}$$

Compare the Eqs. (3) and (6), the expected latency improvement in this case is

$$E[D(N) - D'(N)] = (N - 1) \left( T_{ACK} + \frac{T_{CW\_RTS}}{2} \right) \tag{7}$$

The expected latency improvement of RTS/CTS handshaking with implied acknowledgement is increased linearly with the number of hops. It saves not only the time of ACK but also the delay of carrier sense for RTS at each hop except the first hop and the last hop.

### 3.3 Channel contention

HBGR is a contention window-based protocol. In contention protocols, the common channel which is assigned on demand is shared by all nodes in the network. The decision of which node should access the channel at a particular moment is handled by the contention mechanism [34]. Whenever a node has a packet to transmit, it starts carrier sense until the channel is clear, then it picks a random contention slot in  $[1, CW]$  to send RTS and continues carrier sense. A node wins in a certain slot if and only if it is the only one to choose this slot and all other nodes in its communication range choose the later slots. Collision occurs when more than one node select the same slot.

The RTS/CTS handshaking with implied acknowledgement only guarantees the fast delivery when the delay sensitive packets have obtained the channel. In order to ensure delay sensitive packets have higher priority to obtain the channel than normal packets, HBGR sets contention window size of RTS for delay sensitive packets lower than normal packets. For example, the CW of delay sensitive packets and normal packets set as  $2^i$  and  $2^j$  ( $i > j$ ), respectively. It is easy to calculate the probability of delay sensitive packets to obtain the channel is  $\frac{2^{i+1}-2^j-1}{2^j-1}$  times than normal packets, and analysis is given in next subsection.

HBGR also uses a mechanism of 802.11 which called it *memory* in [35]. When the nodes begin to compete, they set a countdown timer to record value (slot) that picked uniformly from the contention window. When the channel becomes busy, the node pauses the countdown timer. After that, once the channel becomes idle, instead of picking a new random value (slot) in the contention window, it resumes its countdown timer. Whenever the countdown timer reaches zero, the node sends a RTS to request the forwarder.

### 3.4 The probability of obtaining channel

The purpose of this subsection is to analyze the probability of obtaining channel when nodes set different contention window sizes. For simplicity, we just analyze channel contention between two nodes. Suppose node A randomly picks a slot in  $[1, CW_A]$  to send RTS after the channel is in idle state, and node B randomly picks a slot in  $[1, CW_B]$  where  $CW_A = 2^i$ ,  $CW_B = 2^j$ .

When  $i = j > 0$ , the probability of node A winning the channel in one round is

$$P_A^* = \sum_{k=1}^{CW_A-1} \frac{1}{CW_A} \left(1 - \frac{k}{CW_B}\right) = \sum_{k=1}^{2^i-1} \frac{1}{2^i} \left(1 - \frac{k}{2^j}\right) \quad (8)$$

$$= \frac{2^i - 1}{2^{i+1}} = \frac{1}{2} - \frac{1}{2^{i+1}}$$

And  $P_B^* = P_A^*$  since  $i = j$ .

There is a collision if node A and node B pick the same slot. So the probability of collision in one round is

$$P_{collide}^* = \sum_{k=1}^{CW_A} \frac{1}{CW_A} * \frac{1}{CW_B} = \frac{1}{2^i} \quad (9)$$

Node A and B will compete again in the next round if the conflict is occurred, therefore  $P_A = P_A^* + P_{collide}^* * P_A$ , where  $P_A$  is the probability of node A obtaining the channel eventually.

$$P_A = P_B = \frac{P_A^*}{1 - P_{collide}^*} = \frac{1}{2} \quad (10)$$

When  $i > j > 0$ . Since  $CW_A > CW_B$ , node A could win the channel only in  $[1, CW_B - 1]$  slot, the probability of node A winning the channel in one round is

$$P_A^* = \sum_{k=1}^{CW_B-1} \frac{1}{CW_A} \left(1 - \frac{k}{CW_B}\right) = \sum_{k=1}^{2^j-1} \frac{1}{2^i} \left(1 - \frac{k}{2^j}\right) = \frac{2^j - 1}{2^{i+1}} \quad (11)$$

Node B could win the channel in  $[1, CW_B]$  slot, the probability of node B winning the channel in one round is

$$P_B^* = \sum_{k=1}^{CW_B} \frac{1}{CW_B} \left(1 - \frac{k}{CW_A}\right) = \sum_{k=1}^{2^j} \frac{1}{2^j} \left(1 - \frac{k}{2^i}\right) = 1 - \frac{2^j + 1}{2^{i+1}} \quad (12)$$

Since the collision occurs if only if when two nodes select the same slot in  $[1, CW_B]$ , the probability of collision in one round is

$$P_{collide}^* = \sum_{k=1}^{CW_B} \frac{1}{CW_A} * \frac{1}{CW_B} = \sum_{k=1}^{2^j} \frac{1}{2^i} * \frac{1}{2^j} = \frac{1}{2^i} \quad (13)$$

Since  $P_A = P_A^* + P_{collide}^* * P_A$ , the probability of node A obtaining the channel eventually is

$$P_A = \frac{P_A^*}{1 - P_{collide}^*} = \frac{2^j - 1}{2^{i+1} - 2} \quad (14)$$

Similarly, the probability of node B obtaining the channel eventually is

$$P_B = \frac{P_B^*}{1 - P_{collide}^*} = \frac{2^{i+1} - 2^j - 1}{2^{i+1} - 2} \quad (15)$$

Based on the Eqs. (14) and (15), the ratio of the probability to obtain the channel between two nodes is

$$\frac{P_B}{P_A} = \frac{2^{i+1} - 2^j - 1}{2^j - 1} \quad (16)$$

Table 1 shows the ratio of the probability to obtain the channel between two nodes when setting different contention window sizes for them. Therefore, in order to ensure delay sensitive packets have higher priority to obtain the channel than normal packets. HBGR sets CW for delay sensitive packets lower than normal packets. In our simulation, the CW of RTS for normal packets is double for delay sensitive packets.

We just analyze  $i > j > 0$  in this paper since the analysis approach for  $j > i > 0$  is the same.

### 3.5 Forwarding area division scheme for forwarder selection

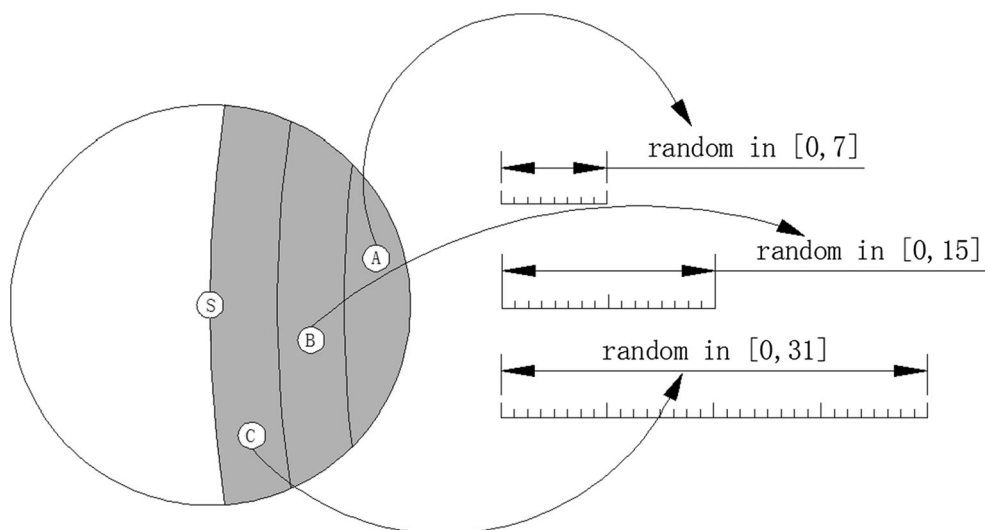
Whenever a node receives a RTS, it is considered as candidate forwarder if it offers positive progress toward the Sink. Instead of sending the CTS immediately after receiving RTS, it sends its CTS with backoff time. In order to ensure the candidate forwarders which are closer to Sink have higher opportunity to be selected as forwarder and reduce the probability of collision of CTS, the backoff time of CTS is determined as follow.

The forwarding area of the sender is defined as the portion of sender's coverage area which is closer to the Sink than the sender. HBGR divides the forwarding area into  $N_r$  regions  $A_1, \dots, A_{N_r}$  based on the circle arcs centered at the Sink, such that all node in  $A_i$  are closer to Sink than

**Table 1** The ratio of the probability to obtain the channel between two nodes when assigning different contention window sizes for them ( $CW_A = 2^i, CW_B = 2^j$ )

$\frac{P_B}{P_A}$	j = 1	j = 2	j = 3	j = 4	j = 5	j = 6	j = 7	j = 8	j = 9	j = 10
i = 1	1									
i = 2	5	1								
i = 3	13	3.667	1							
i = 4	29	9	3.286	1						
i = 5	61	19.667	7.857	3.133	1					
i = 6	125	41	17	7.4	3.065	1				
i = 7	253	83.667	35.286	15.933	7.194	3.032	1			
i = 8	509	169	71.857	33	15.452	7.095	3.016	1		
i = 9	1021	339.667	145	67.133	31.968	15.222	7.047	3.008	1	
i = 10	2045	681	291.286	135.4	65	31.476	15.110	7.024	3.004	1

**Fig. 2** The forwarding area of the sender is divided by the circle arcs centered at Sink, and each candidate forwarder uses different CW, based on which subareas it located, to calculate its backoff time



all node in  $A_j$  for  $j > i, i = 1, \dots, N_r - 1$ . Hence  $A_1$  offers the maximum progress.

HBGR assigns different contention window sizes of CTS to nodes in different subareas and defines as  $CW_{A_i} = 2^{i+c}, 0 < i \leq N_r, c$  is a positive integer. The nodes which located in  $A_i$  pick a random contention slot in  $[0, CW_{A_i} - 1]$ , therefore the backoff time of CTS is

$$T_{backoff,A_i} = random(0, CW_{A_i} - 1) * t_{slot} + t_{SIFS} \tag{17}$$

where  $t_{SIFS}$  is equal to the short Inter Frame Spacing time from 802.11. And  $t_{slot}$  is the value of slot duration.

Figure 2 illustrates the calculation of backoff time for CTS. Node S is the sender and request to send. The forwarding area is identified in gray, and is divided by circle arcs centered at the Sink. HBGR assigns different subareas with different contention window sizes. Node A, B and C are the candidate forwarders of Node S which located in different subareas. After receiving a RTS from Node S, Node A, B and C calculate the backoff time of CTS based on their contention window size.

### 4 Simulation result and analysis

For performance analysis, in addition to HBGR, we have implemented another three routing protocols: GF, OGF [14], and AODV [36]. GF is a basic beaconless geographic forwarding protocol which uses traditional RTS/CTS handshaking mechanism to select forwarder. Whenever a node has packet to send, it sends a RTS. Candidate forwarders backoff a random time to respond the RTS, the forwarder is the first one to send out the CTS. OGF is a purely on-demand and beaconless protocol, it integrates RTS/CTS handshaking with progress and residue energy of candidate forwarders to determine forwarder. AODV is chosen because it is a well-accepted routing protocol for wireless network.

#### 4.1 Performance metrics

We choose the following metrics to analyze the performance of HBGR and other protocols.

1. *Packet delivery ratio* defined as the percentage of generated packets which are successfully delivered to the Sink. This metric reflect the reliability of data delivery
2. *End-to-End latency* defined as the time from when a packet is generated to when it is delivered to the Sink.
3. *Energy consumption per packet* defined as the ratio of the total amount of energy consumption to the number of distinct data packets received by Sink.

## 4.2 Simulation parameter and scenario

In all simulation, three nodes are selected as sources and each source generates 3000 data packets with a payload of 250 bytes. Bandwidth is set as 250 kbps and transmission range is set as 15 m. Unless specially noted, the percentage of delay sensitive packets is 10 %, and the CW of RTS for normal packets is double for delay sensitive packets. The forwarding area is divided into three subareas and the CW of CTS for first subarea is 32. The simulation is terminated when the Sink receives all data packet generated by each source and every data point in our result is an average over 10 runs with different seeds.

Two kinds of scenarios are designed to evaluate the performance of the protocol.

*Stationary scenario:* There are 200 nodes in our simulation. In a  $100\text{ m} \times 100\text{ m}$  region, 100 nodes are uniformly deployed to guarantee the connectivity of topology, and another half nodes are randomly scattered to increase redundancy. Sink is located at (5 m,5 m) and three source nodes are located at (5 m,95 m), (95 m,5 m) and (95 m,95 m), respectively.

*Mobility scenario:* All 200 nodes are randomly scattered in a  $80\text{ m} \times 80\text{ m}$  region and follow the random waypoint model except Sink and source nodes. A node chooses a destination randomly in the simulated region, chooses a velocity randomly from a configuration range, and then moves to that destination at the chosen velocity. Upon arriving at the chosen waypoint, the node repeats the same process. And maximum speed is set as 1 m/s. Sink is located at a corner of region which location is (5 m,5 m). And three source nodes are also located at corner of region, their location are (5 m,75 m), (75 m,5 m) and (75 m,75 m), respectively.

The simulation parameters are summarized in Table 2.

## 4.3 Performance in stationary scenario

In this set of simulations, we evaluate the performance of protocols in stationary scenario. The packet generation rate per source is varied from 1 to 24 p/s (packets per second) to provide different levels of congestion.

Figure 3(a) shows the average End-to-End latency of a packet from source to the Sink under different packet generation rates. *HBGR-DS* denotes the delay sensitive packets that use HBGR protocol and *HBGR-N* denotes the normal packets. Since the GF is a basic beaconless geographic forwarding protocol, it could become the benchmark for performance comparing. The latency of GF increases slowly when the packet generation rate is lower than 14 p/s, but after that the latency increases quickly. OGF performs better than HBGR-N when the packet generation rate is lower than 4 p/s. However, the latency of OGF increases quickly and is even higher than GF when the rate of packet generation is beyond 10 p/s. The reason is that OGF sets the carrier sensing range to be at least twice the transmission range in order to prevent interference among candidate forwarders, it will reduce the utilization of channel resource. The delay performance of AODV is similar to HBGR-N when the packet generation rate is lower than 10 p/s, after that AODV performs worse than HBGR-N but is still superior to GF. The periodically beacon of AODV will reduce the utilization of channel resource, its impact can be ignored when the traffic is light, but when the network is congested, it will incur long delay. Another reason is that AODV needs rediscovery the routing path when the communication link has failed due to collision, and the probability of collision near Sink is high when network is congested. With the increasing of packet generation rate, no matter HBGR-DS or HBGR-N, the latency increases smoothly. HBGR-DS outperforms other protocols all the time in this scenario, and HBGR-DS has at least 10 % improvement than HBGR-N in terms of End-to-End latency. The improvement increases with the increasing of packet generation rate, and achieves 58 % when the packet generation rate is 24 p/s.

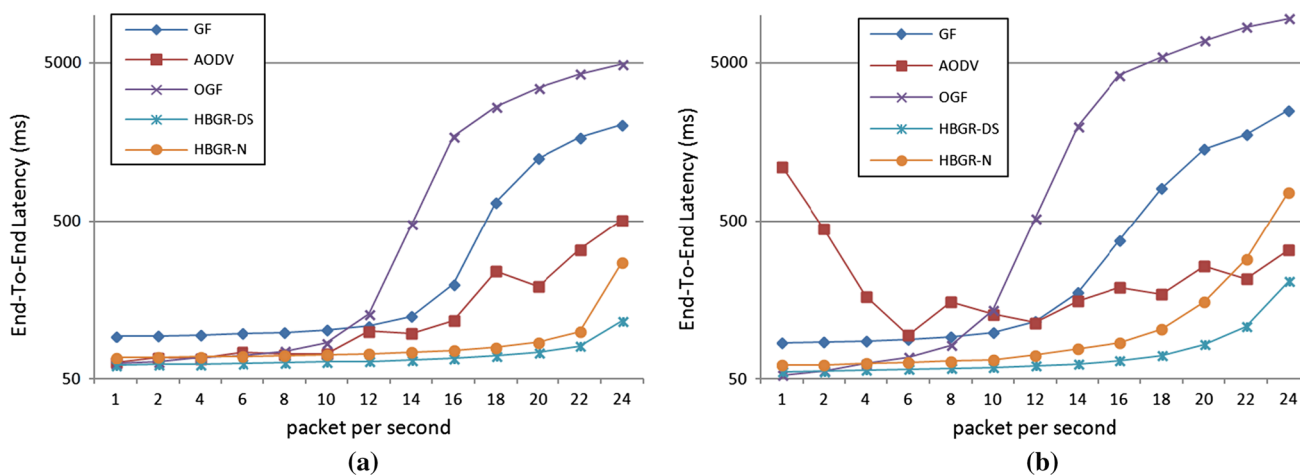
From Fig. 4(a), we can see that when the packet generation rate is less than or equal to 10 p/s, the packet delivery ratio of all protocols is almost close to 100 %. However, when the packet generation rate is beyond 10 p/s, all protocols start to decline. OGF declines fastest since the expansion of carrier sensing range would increase interference to the neighboring nodes which might degrade the performance, especially when the network is congested. The rate of decline of HBGR-DS and HBGR-N is lower than other protocols. And the effect of packet's composition on End-to-End latency and packet delivery ratio will be discussed in Sect. 4.5

Figure 5(a) shows the energy consumption per packet. It is almost impossible to separate the energy consumption on HBGR-DS and HBGR-N, since not only the overhead of control packet but also the energy consume by idle listening and overhearing are hard to separate. So we just show the energy consumption for all packets and denote as *HBGR*. It is worth to note that the energy consumption per



**Table 2** System parameter for simulation experiments

Parameter	Value
Data packet size	250 bytes CBR payload
Bandwidth	250 kbps
Transmission range	15 m
Receive/send/idle power	395 mW/660 mW/35 mW
Data packet generation rate	1–24 packets per second (p/s)
Total packet sent	3000 per source
The number of nodes	200
The number of sources	3
The percentage of delay sensitive packet	10 %
The percentage of normal packet	90 %
Terrain of stationary scenario	100 m × 100 m
Sink position in stationary scenario	(5 m,5 m)
Source position in stationary scenario	(5 m,95 m), (95 m,5 m), (95 m,95 m)
Terrain of mobility scenario	80 m × 80 m
Sink position in mobility scenario	(5 m,5 m)
Source position in mobility scenario	(5 m,75 m), (75 m,5 m), (75 m,75 m)
Maximum moving speed in mobility scenario	1 m/s



**Fig. 3** The average End-to-End latency of a packet from the source to the sink. **a** Stationary scenario, **b** mobility scenario

packet decrease with the increasing of packet generation rate, and the reason is that the simulation will run until the Sink receives all data packet generated by each source. Then the time of simulation would be longer under lower packet generation rate that causes more energy consumption on idle listening. We can see that HBGR achieves better energy efficiency than other protocols. The main reason is that HBGR achieves high packet delivery ratio, and RTS/CTS handshaking with implied acknowledgement further reduces the cost of control packet. OGF performs worst since the expansion of carrier sensing range incurs more energy consumption on overhearing. For instance, when a node sends a packet, more nodes will sense this packet.

### 4.4 Performance in mobility scenario

In this set of simulations, we evaluate the performance of protocols in mobility scenario in which the topology changes frequently due to node moving. The ranges of packet generation rate is varied from 1 to 24 p/s each source. And the maximum moving velocity is set as 1 m/s.

Figure 3(b) shows the average End-to-End latency of a packet from source to the Sink under different degrees of congestion. It is very weird that the latency of AODV decreases with the increasing of packet generation rate at first. The reason may be that it could deliver more packets from source to sink during the period from establishment of routing path to failure if accelerating the rate of packet

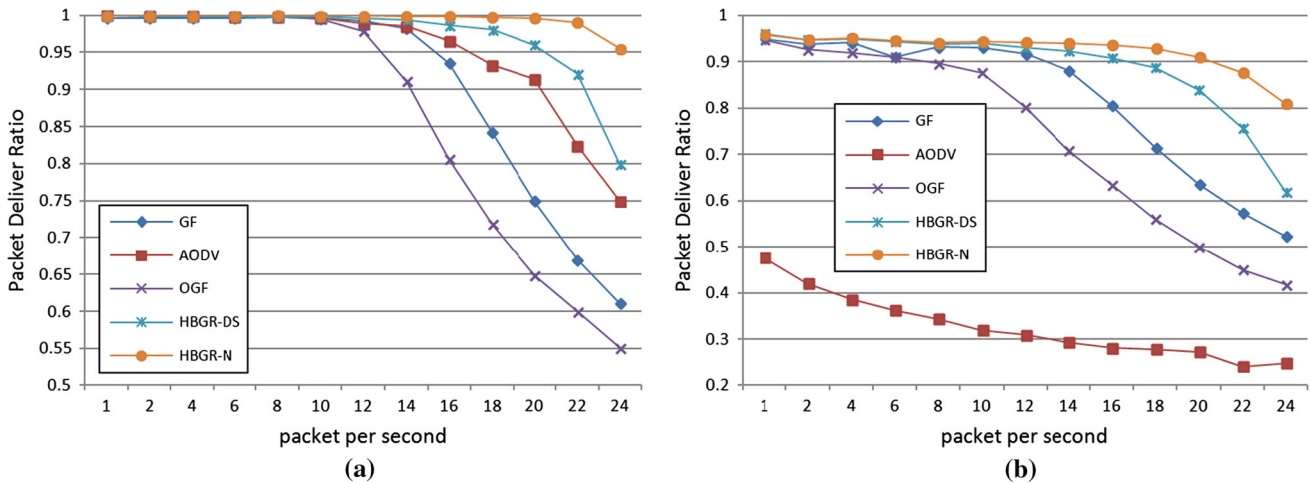


Fig. 4 Packet delivery ratio. **a** Stationary scenario, **b** mobility scenario

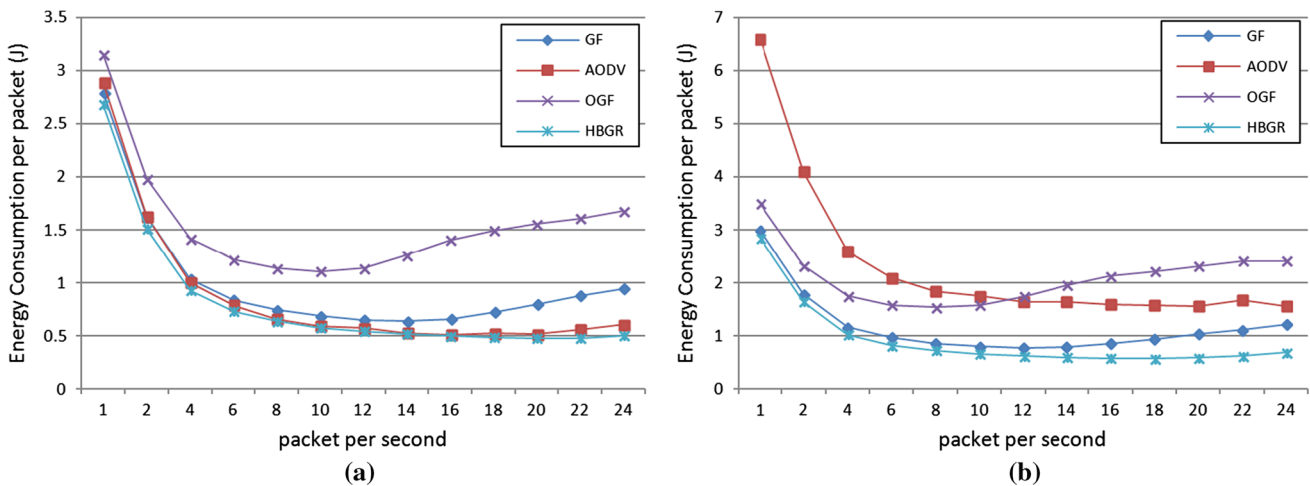


Fig. 5 Energy consumption for packet transmission. **a** Stationary scenario, **b** mobility scenario

generation. OGF performs best when the packet generation rate is 1 p/s, but after that it experiences a very rapid deterioration with the increasing of packet generation rate. In addition to expansion of carrier sensing range, recording the forwarder may have some negative impact since the forwarding table would stale quickly due to node moving. No matter HBGR-DS or HBGR-N performs excellent, and the latency of them increases smoothly. Compared with HBGR-N, the improvement of HBGR-DS increases with the increasing of packet generation rate, and achieves 73 % when the packet generation rate is 24 p/s.

Figure 4(b) shows the packet delivery ratio in mobility scenario. The packet delivery ratio of all protocols declines with the increasing of packet generation rate. HBGR-DS and HBGR-N perform better than other protocols in terms of packet delivery ratio. The rate of decline of HBGR-DS and HBGR-N are slowly than other

protocols. However, even when the traffic is very light, the packet delivery ratio of AODV is lower than 50 %. The reason is that the node moving invalidates pre-established routing path.

Figure 5(b) shows the energy consumption per packet in mobility topology. We can see that HBGR outperforms other protocols, this must thank to the high packet delivery ratio and RTS/CTS handshaking with implied acknowledgement. Compared with stationary scenario, energy efficiency of AODV degrades since packet delivery ratio is low and the cost of routing path rediscovery is high in mobility scenario.

#### 4.5 Effect of packet's composition

In order to study the effect of the composition of the packet, we evaluate the performance of HBGR in

stationary scenario, and the percentage of delay sensitive packet is varied from 0 to 100 %.

From Figs. 6 and 7, we can see that the End-to-End latency and packet delivery ratio of all packets, which denote as *Total*, decrease with the increasing of delay sensitive packet’s ratio.

When traffic is moderate (10 p/s), the latency of HBGR-N and HBGR-DS are not sensitive to the composition of the packet and keep almost constant. And the latency of all packets decreases with the increasing of delay sensitive packet’s percentage. However, the packet delivery ratio of all packets just reduces 0.3 % when the percentage of delay sensitive packet increases from 0 to 100 %. Undoubtedly, it is a good deal to exchange 10.9 % improvement in End-to-End latency with 0.3 % reduction in packet delivery ratio.

When traffic is heavy (20 p/s), the latency of HBGR-N and HBGR-DS decrease with the increasing of delay sensitive packet’s ratio. The reason is that the fast delivery of delay sensitive packets releases more network resources for normal packets. And the packet delivery ratio of all packets reduces 3.3 % when the delay sensitive packet’s percentage changes from 0 to 100 %.

We also simulate the case which the traffic is very light (1 p/s), and we not show it here since the result of End-to-End latency is similar to Fig. 6(a) and the effect on packet delivery ratio can almost be ignored.

From the above, it can be concluded that the effect of delay sensitive packet’s percentage on packet delivery ratio can almost be ignored when traffic is not heavy. For achieving low latency, all packets could be set as delay sensitive packets. In other words, only the RTS/CTS handshaking with implied acknowledgement is used for all packets. However, when the traffic is heavy, the tradeoff

between latency and packet delivery ratio must be considered.

### 4.6 Effect of forwarding area division

In order to study the effect of forwarding area division, we set all the packets as normal packets in the simulation. In other words, only the traditional RTS/CTS handshaking is used for all packets. And all the result in this subsection is simulated in the stationary scenario. We also introduce another metric which is the number of hops from the source to the Sink to evaluate the impact of forwarding area division.

AREA<sub>*n*</sub> denotes the area division scheme that divides the forwarding area into *n* subareas. The contention window size of first subarea (*A*<sub>1</sub>) is set as 32 for all area division schemes except AREA<sub>1</sub>. Since 32 is too short for whole forwarding area that will incur too much collision during forwarder selection, and the packet delivery ratio is lower than 70 % even though the traffic is very light. Therefore, we set size of contention window as 64 for AREA<sub>1</sub>.

Figures 8 and 9 show the effect of different area division schemes on the End-to-End latency and the number of hops, respectively. We don’t draw all the result in the Fig. 8. Since when the packet generation rate is higher than 20 p/s, the latency increases slight quickly that make it difficult to observe differences between different schemes. Performance in terms of the number of hops improves quite significantly with the increasing of the number of subareas. However when the number of subareas is greater than 4, the performance in terms of latency even degrades, especially when the traffic is heavy. One of the reasons is that the probability of first

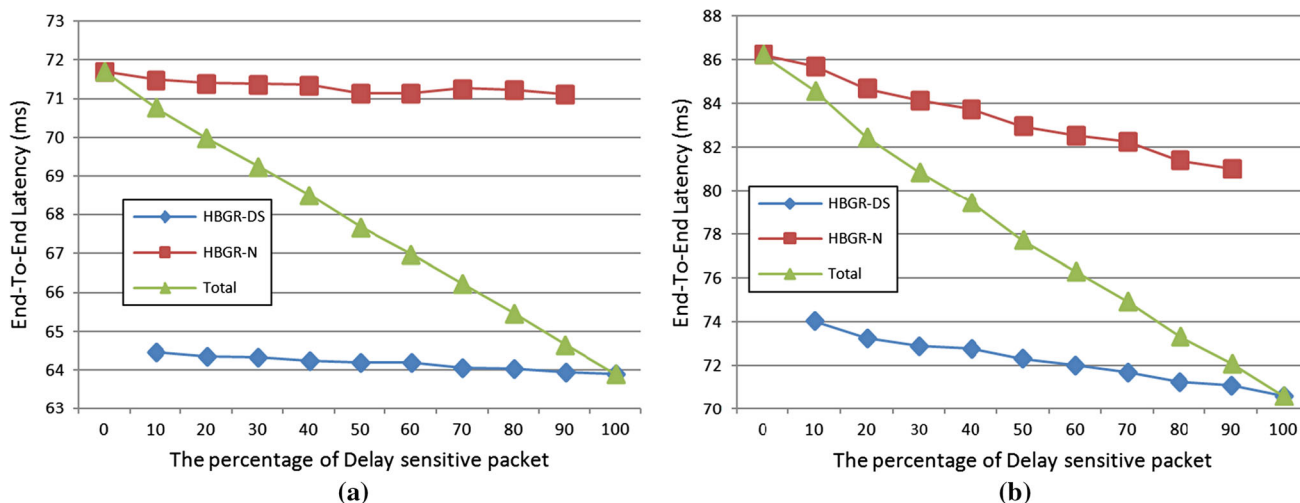


Fig. 6 The average End-to-End latency of a packet from the source to the sink. a 10 packets per second each source, b 20 packets per second each source

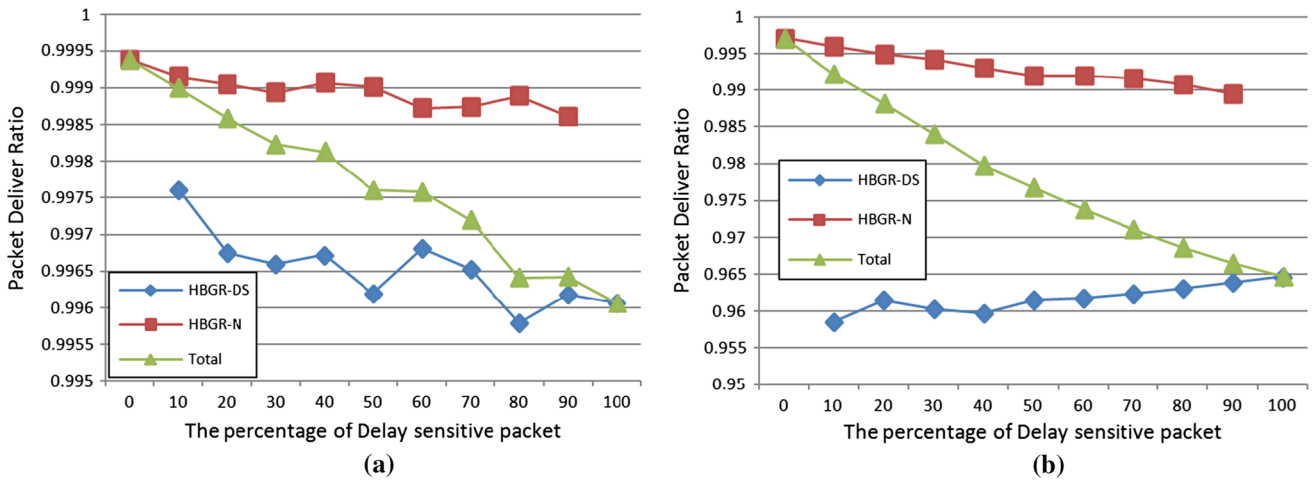


Fig. 7 Packet delivery ratio. a 10 packets per second each source, b 20 packets per second each source

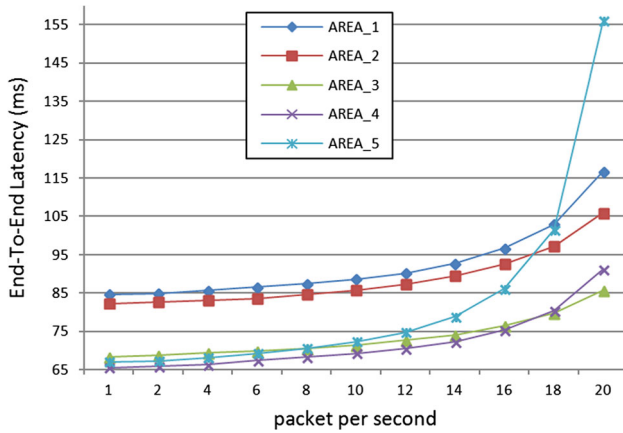


Fig. 8 The end-to-end latency

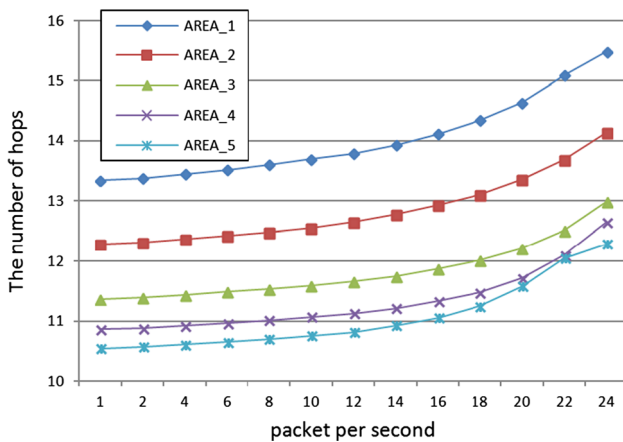


Fig. 9 The number of hops from the source to sink

several subareas is empty, which means that there is not node located in the first several subareas, increases with the increasing of the number of subareas. Another reason

is that the timeout of RTS must set large enough to guarantee the nodes which located in the last subarea could respond to RTS, and the timeout of RTS increases exponentially with the increasing of the number of subareas. Therefore, when a node does not receive the valid CTS due to the collision after sending RTS, it must wait longer to retransmit another RTS with the increasing of the number of subareas. And the probability of collision also increases with the increasing of the packet generation rate.

### 5 Conclusion

In this paper, we proposed a novel geographic routing protocol called HBGR for Wireless sensor network (WSN). HBGR uses different mechanisms to serve for different kinds of packets in order to improve the latency of delay sensitive packets with little impact on normal packets. And forwarding area division scheme is proposed to optimize the forwarder selection that guarantees the candidate forwarders, which are nearer to the Sink, have higher probability to become the forwarder. The simulation results exhibit the superior performance of HBGR, in terms of packet delivery ratio, End-to-End latency, and energy consumption per packet, under different packet generation rates in stationary and mobility scenario. We also study the effect of packet’s composition, the results show that we achieved our goal: improving the latency of delay sensitive packets with little impact on normal packets. Furthermore, we also evaluate the effect of forwarding area division scheme, the results show that forwarding area division could effective improve the forwarder selection.

## References

1. Chen, D., & Varshney, P. K. (2007). A survey of void handling techniques for geographic routing in wireless networks. *IEEE Communication on Surveys & Tutorials*, 9(1), 50–67.
2. Pantazis, N. A., Nikolidakis, S. A., & Vergados, D. D. (2013). Energy-efficient routing protocols in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, 15(2), 551–591.
3. Sheng, Z., Yang, S., Yu, Y., et al. (2013). A survey on the IETF protocol suite for the internet of things: Standards, challenges, and opportunities. *IEEE Wireless Communications*, 20(6), 91–98.
4. Li, M., Li, Z., & Vasilakos, A. V. (2013). A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues. *Proceedings of the IEEE*, 101(12), 2538–2557.
5. Vasilakos, A., et al. (2012). *Delay tolerant networks: Protocols and applications*. Boca Raton: CRC Press.
6. Chen, M., Leung, V. C. M., Mao, S., et al. (2009). Hybrid geographic routing for flexible energy-delay trade-offs. *IEEE Transactions on Vehicular Technology*, 58(9), 4976–4988.
7. Park, J., Kim, Y. N., & Byun, J. Y. (2013). A forwarder selection method for greedy mode operation of a geographic routing protocol in a WSN. In *Fifth international conference on ubiquitous and future networks (ICUFN), Da Nang* (pp. 270–275).
8. Karp, B., & Kung, H. T. (2000). GPSR: Greedy perimeter stateless routing for wireless networks. In *6th annual ACM/IEEE international conference on mobile computing and networking (MobiCom 2000), Boston*.
9. Leong, B., Liskov, B., & Morris, R. (2006). Geographic routing without planarization. In *3rd conference on networked systems design and implementation, San Jose* (pp. 25–39).
10. Zou, L., Lu, M., & Xiong, Z. (2007). PAGER-M: A novel location-based routing protocol for mobile sensor networks. In *2007 ACM SIGMOD international conference on management of data, Beijing* (pp. 1182–1185).
11. Yu, Y., Govindan, R., & Estrin, D. (2001). Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks. UCLA Computer Science Department Technical Report.
12. Zeng, Y., Xiang, K., Li, D., et al. (2013). Directional routing and scheduling for green vehicular delay tolerant networks. *Wireless Networks*, 19(2), 161–173.
13. Blum, B. M., He, T., Son, S., et al. (2003). IGF: A state-free robust communication protocol for wireless sensor networks. Technical Report CS-2003-11: Department of Computer Science, University of Virginia.
14. Chen, D., & Varshney, P. K. (2007). On-demand geographic forwarding for data delivery in wireless sensor networks. *Computer Communications*, 30(14–15), 2954–2967.
15. Casari, P., Nati, M., Petrioli, C., et al. (2006). ALBA: An adaptive load-balanced algorithm for geographic forwarding in wireless sensor networks. In *IEEE military communications conference (MILCOM), Washington, DC*.
16. Casari, P., Nati, M., Petrioli, C., et al. (2007). Efficient non-planar routing around dead ends in sparse topologies using random forwarding. In *IEEE international conference on communications, Glasgow* (pp. 3122–3129).
17. Lima, C., & Abreu, G. T. F. D. (2008). Game-theoretical relay selection strategy for geographic routing in multi-hop WSNs. In *5th workshop on positioning, navigation and communication 2008 (WPNC'08), Germany* (pp. 277–283).
18. Zorzi, M., & Rao, R. R. (2003). Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Energy and latency performance. *IEEE Transactions on Mobile Computing*, 2(4), 349–365.
19. Zorzi, M., & Rao, R. R. (2003). Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance. *IEEE Transactions on Mobile Computing*, 2(4), 337–348.
20. Zhang, H., & Shen, H. (2010). Energy-efficient beaconless geographic routing in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 21(6), 881–896.
21. Xiang, L., Luo, J., & Vasilakos, A. (2011). Compressed data aggregation for energy efficient wireless sensor networks. In *2011 8th annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks (SECON), Salt Lake City, UT* (pp. 46–54).
22. Wei, G., Ling, Y., Guo, B., et al. (2011). Prediction-based data aggregation in wireless sensor networks: Combining grey model and Kalman Filter. *Computer Communications*, 34(6), 793–802.
23. Liu, X., Zhu, Y., Kong, L., et al. (2014). CDC compressive data collection for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(8), 2188–2197.
24. Meng, T., Wu, F., Yang, Z., et al. (2015). Spatial reusability-aware routing in multi-hop wireless networks. *IEEE Transactions on Computers*. doi:10.1109/TC.2015.2417543.
25. Li, P., Guo, S., Yu, S., et al. (2012). CodePipe: An opportunistic feeding and routing protocol for reliable multicast with pipelined network coding. In *2012 proceedings IEEE INFOCOM, Orlando, FL* (pp. 100–108).
26. Li, P., Guo, S., Yu, S., et al. (2014). Reliable multicast with pipelined network coding using opportunistic feeding and routing. *IEEE Transactions on Parallel and Distributed Systems*, 25(12), 3264–3273.
27. Cheng, H., Xiong, N., Vasilakos, A. V., et al. (2012). Nodes organization for channel assignment with topology preservation in multi-radio wireless mesh networks. *Ad Hoc Networks*, 10(5), 760–773.
28. Song, Y., Liu, L., Ma, H., et al. (2014). Physarum optimization is a biology-inspired algorithm which could be used for improving the quality of coverage in WSN. *IEEE Transactions on Network and Service Management*, 11(3), 417–430.
29. Liu, L., Song, Y., Zhang, H., et al. (2015). Physarum optimization: A biology-inspired algorithm for the Steiner tree problem in networks. *IEEE Transactions on Computers*, 64(3), 819–832.
30. Yao, Y., Cao, Q., & Vasilakos, A. V. (2013). EDAL: an energy-efficient, delay-aware, and lifetime-balancing data collection protocol for wireless sensor networks. In *2013 IEEE 10th international conference on mobile ad-hoc and sensor systems (MASS), Hangzhou* (pp. 182–190).
31. Yao, Y., Cao, Q., & Vasilakos, A. V. (2015). EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks. *IEEE/ACM Transactions on Networking*, 23(3), 810–823.
32. Liu, Y., Xiong, N., Zhao, Y., et al. (2009). Multi-layer clustering routing algorithm for wireless vehicular sensor networks. *IET Communications*, 4(7), 810–816.
33. Han, K., Luo, J., Liu, Y., et al. (2013). Algorithm design for data communications in duty-cycled wireless sensor networks: A survey. *IEEE Communications Magazine*, 51(7), 107–113.
34. Chilamkurti, N., Zeadally, S., Vasilakos, A., et al. (2009). Cross-layer support for energy efficient routing in wireless sensor networks. *Journal of Sensors*, 2009, 1–9.
35. Jamieson, K., Balakrishnan, H., & Tay, Y. C. (2006). Sift: A MAC protocol for event-driven wireless sensor networks. In *3rd European workshop on wireless sensor networks (EWSN'06), Zurich, Switzerland* (pp. 260–275).
36. Perkins, C. E., & Royer, E. M. (1999). Ad-hoc on-demand distance vector routing. In *2nd IEEE workshop on mobile computing systems and applications*.



**Chao Hong** is currently a graduate student at College of Computer Science, Chongqing University, China. He received his B.E. degree in Computer Science and Technology from Nanjing University of Posts and Telecommunications, China in 2010. His research interests are in the fields of wireless sensor networks (WSNs) and software defined networks (SDNs).



**Yufang Zhang** received her M.S. degree in Computer Science from Chongqing University, China in 1988. She received the Ph.D. degree in Computer Software from Chongqing University in 2001. She is a professor in College of Computer Science in Chongqing University. Her research interests involve in data mining, cluster computing and network.



**Zhongyang Xiong** received his M.S. degree in 1988 and Ph.D. degree in 2004 from Chongqing University, China. He is a professor in College of Computer Science in Chongqing University. His research interests involve in computer network, big data processing.