Wireless Sensor Network, 2010, 2, 528-537
doi:10.4236/wsn.2010.27065 Published Online July 2010 (http://www.SciRP.org/journal/wsn)

Contention-Based Beaconless Real-Time Routing Protocol for Wireless Sensor Networks*

Chao Huang, Guoli Wang
School of Information Science & Technology, Sun Yat-Sen University, Guangzhou, China
E-mail: *superchaohuang@gmail.com, isswgl@mail.sysu.edu.cn*
Received April 12, 2010; *revised May* 4, 2010; *accepted May* 17, 2010

Abstract

This paper presents a novel real-time routing protocol, called CBRR, with less energy consumption for wireless sensor networks (WSNs). End-to-End real-time requirements are fulfilled with speed or delay constraint at each hop through integrating the contention and neighbor table mechanisms. More precisely, CBRR maintains a neighbor table via the contention mechanism being dependent on wireless broadcast instead of beacons. Comprehensive simulations show that CBRR can not only achieve higher performance in static networks, but also work well for dynamic networks.

Keywords: Wireless Sensor Network, Real-Time Routing Protocol, Contention-Based Scheme, Beaconless

1. Introduction

WSNs are comprised of tiny, low power sensor nodes, which are densely deployed in a monitored area and collaborate to forward the sensed data to a base station through multiple hops. WSNs can be widely used in many applications, such as environmental monitoring, military surveillance, disaster recovery, and healthcare related applications [1]. However, many mission-critical applications require that WSNs can guarantee the satisfied quality of service (QoS), especially with real-time constraints. e.g., in forest fire detection, the abnormal temperature information should be delivered to a base station as soon as possible. Otherwise, the outdated data will be irrelevant and even have negative effects on the applications.

WSNs have particular features different from other wireless networks [2-4]. Firstly, each sensor node has severe resource constraints on bandwidth, memory, processing capability, and especially energy. Secondly, the number of sensor nodes is very large and the nodes are always densely deployed. Thirdly, it has been recognized that the network topologies of WSNs may change constantly due to sleep policy, node mobility, node failure, and so on. The features mentioned above will pose challenges to fulfill the real-time requirements.

*This work is supported by Nature Science Foundation of China under Grant 60775055 and Guangdong Nature Science Foundation of China under Grant 06023190.

Due to the high routing overhead and the poor scalability of global routing decisions, several real-time routing protocols [5-7] utilize the localized geographic information to heuristically find a satisfied path to a target. These protocols are mostly governed by the conventional geographic routing schemes, in which each node has to periodically broadcast beacons for obtaining the accurate information of its neighbors and to store the information in its neighbor table for routing decisions. Although the neighbors' information can be collected by broadcasting beacons, some drawbacks inevitably arise. Firstly, redundant beacons can lead to more energy consumption and higher communication cost. Secondly, only one neighbor or very small subset of neighbors can take part in routing decisions, but each node should store all of neighbors' information in a neighbor table. Therefore, the utilization of a neighbor table is much inefficient, and the unused neighbors' information still occupies some of the limited memory. Thirdly, when a network topology changes constantly, the collected neighbor information will be outdated quickly, which in turn leads to higher packet miss ratio, and nodes should broadcast beacons more frequently to update the neighbor tables. It inevitably incurs not only longer delay, but also consumes significant energy. Furthermore, frequently broadcasting beacons can result in more serious collisions thus higher packet miss ratio.

Contention-based beaconless scheme [8-12] has been proposed to handle with the issues abovementioned. The main idea is that when a node has a DATA packet to be

transmitted, it first broadcasts a contention request, and one neighbor with the shortest wait delay is selected as the most suitable next-hop forwarder. The contention-based scheme is completely stateless and reactive because each node can on-line select an available next-hop forwarder without any prior knowledge of its neighbors. This results in less energy consumption. Furthermore, it can be found that the current forwarder can obtain the information of the selected next-hop forwarder during a contention procedure. Although the information of the selected neighbor is not utilized in the scheme, it can benefit to future routing decisions. The character indicates that the approach for selecting the information of neighborhoods is energy efficient.

This paper presents a Contention-based Beaconless Real-time Routing protocol for WSNs, called CBRR. CBRR aims at fulfilling end-to-end real-time requirements with less energy consumption through integrating the contention and neighbor table mechanisms. The advantage of our approach is to remove the limitations of the beacon-based routing schemes, especially, in dynamic networks. Three remarkable aspects of CBRR are emphasized as follows. Firstly, the key to CBRR is that the information in a neighbor table is obtained through a contention mechanism without beacons, and CBRR employs the contention mechanism to select a new forwarder only when there is no an available neighbor, which meets the real-time requirements, in the neighbor table. Secondly, the neighbor table allows CBRR to select a next-hop forwarder by direct unicast or contention-based forwarding. As a consequence, the end-to-end real-time requirements can be fulfilled with speed or delay constraint at each hop. Thirdly, CBRR can collect the information from the two-hop neighbors during the wireless broadcasting, and the two-hop neighbor table can be helpful for further meeting the real-time requirements in the two-hop range.

The rest of the paper is organized as follows. Section 2 outlines the related works. Section 3 presents the proposed real-time routing protocol. Section 4 reports the experimental results.

2. Related Works

Recently, real-time routing protocol has aroused much research interests in WSNs. SAR [13] was the first QoS protocol for WSNs. It could find multiple paths between a source and a target with different priority in terms of energy efficiency and fault tolerance. EQR [14] tried to find a least-cost yet energy efficient path for real-time data, and to maximize the throughput for non-real-time data with a class-based queue model, simultaneously. However, both SAR and EQR were based on global routing decisions, which result in much higher routing overheads. Therefore, localized geographic routing decisions have been popular employed in a variety of real-time routing protocols. SPEED [5] was designed to provide soft end-to-end real-time guaranty with a desired delivery speed, $S_{setpoint}$, which could support the real-time communication between a source and a sink, through a combination of feedback control and non-deterministic geographic forwarding. Only the node, whose progress speed was higher than $S_{setpoint}$, could be selected as the next-hop forwarder. MMSPEED [6] was an extension to SPEED with additional service differentiations both in the timeliness and reliability domains. RPAR [7] fulfilled the end-to-end real-time requirements with low energy consumption by dynamically adapting transmission power and assigning one-hop velocity. Unfortunately, these localized routing protocols had to broadcast beacons to collect the neighbors' information and were inevitably suffered from the aforementioned drawbacks.

Contention-based beaconless scheme has been developed to address the issues aforementioned. IGF [8] only allowed the nodes within the 60-degree sector towards the sink to participate in the contention. The wait delay of each available node was determined by the combination of the progress distance to a sink, the remaining energy of the node and a random delay. GeRaF [9,10] used the busy tone to avoid collisions and replaced the wait delay function by time slots, which were assigned to different regions of forwarding areas. SGF [11] introduced a forwarding scheme through integrating the contention mechanism and gradient for unexpected node/link failures in highly dynamic networks. OGF [12] combined the contention mechanism with a neighbor table to forward packets in slowly dynamic networks. OGF was similar to our approach, but it did not consider the real-time constraints.

3. Main Results

Our design goal is to fulfill the end-to-end real-time requirements with less energy consumption for WSNs. Two versions of CBRR protocol are proposed, such as CBRR-OneHop and CBRR-TwoHop, which are based on one-hop neighbor table and two-hop neighbor table, respectively. CBRR-OneHop protocol is the foundation of CBRR, and CBRR-TwoHop protocol is an extension to CBRR-OneHop. Before describing CBRR protocol, we first introduce the relevant definitions and assumptions.

3.1. Definitions and Assumptions

Denote source node and sink node as $\boldsymbol{S}$ and $\boldsymbol{D}$. Let $d(i, j)$ and $Delay_i^j$ be the Euclidean distance and the delay between node i and node j, respectively. $Deadline(D)$ is the required deadline of a DATA packet for D and is

carried in the header of the DATA packet. R_c is the radio range.

Neighbor Set of Node *i*: NS_i is the set of nodes within the radio range of node *i*. Formally, $NS_i = \{j \mid d(i,j) \le R_c\}$.

Contention Candidate Set of Node *i*: CCS_i is the set of nodes which are in NS_i and closer to *D* than node *i*. Formally, $CCS_i = \{j \mid d(i,D) > d(j,D), j \in NS_i\}$.

Relay Speed: $Speed_i^j$ is the velocity between node *i* and *j*, which is the ratio of the distance to the delay between node *i* and *j*. Formally, $Speed_i^j = d(i,j) / Delay_i^j$.

Required Speed: $Speed_i^D$ is the velocity, which meets the real-time requirements. It is the ratio of the distance between node *i* and *D* to the remaining time of a DATA packet's deadline. Formally, $Speed_i^D = d(i,D) / (Deadline(D) - T_{now})$, where T_{now} is the current time.

Estimated Hop: Hop_i^D is the estimated hop number between node *i* and *D*, which is the ratio of the distance between node *i* and *D* to R_c. Formally, $Hop_i^D = d(i,D) / R_c$.

Required Delay: $Delay_i^D$ is the one-hop delay, which meets the real-time requirements. It is the ratio of the remaining time of the packet's deadline to the estimated hop. Formally, $Delay_i^D = (Deadline(D) - T_{now}) / Hop_i^D$.

Speed Constraint: Speed constraint is used for each node to find an available neighbor, whose relay speed is no less than the required speed, in its neighbor table. If having an available neighbor, the node can send the DATA packet to the neighbor by direct unicast. Otherwise, the node has to invoke a contention procedure to select a new next-hop forwarder.

Delay Constraint: Delay constraint is used for each node to determine whether its contention candidates can take part in the contention or not. If there is an available neighbor, whose one-hop delay is no less than the required delay, in the neighbor table of the contention candidate, this candidate can be admitted to participate in the contention. Otherwise, the candidate is forbidden to do that.

Assume that each node is aware of its position, and it has the same radio range and initial energy with a unique ID and an out-of-band busy tone.

3.2. CBRR-OneHop Protocol

CBRR-OneHop protocol is the foundation of CBRR, and it consists of five components, such as unicast forwarding policy, contention forwarding policy, contention function with wait delay, delay estimation and one-hop neighbor table, which is created or updated only during a contention procedure. When a source intends to send its first DATA packet, the node has to employ the contention forwarding policy to select a next-hop forwarder because the neighbor table of each node has not been created. After the first DATA packet is successfully forwarded to the sink, each node, which is on the previous forwarding path, has at least one neighbor in its one-hop neighbor table. Therefore, the first DATA packet can be regarded as a special packet for the path discovery. In the future forwarding, the one-hop neighbor table can play an important role to help with selecting an available next-hop forwarder, which meets the speed constraint or delay constraint, by the unicast forwarding policy or the contention forwarding policy.

- **Unicast Forwarding Policy**

As shown in **Figure 1**, the unicast forwarding policy can utilize the speed constraint to determine that a DATA packet is forwarded by direct unicast or invoking the contention forwarding policy. If the neighbor table has an available neighbor, which meets the speed constraint, the current forwarder selects the neighbor as the next-hop forwarder and unicasts the DATA packet to the neighbor directly. Otherwise, the current forwarder has to invoke the contention forwarding policy to select a new next-hop forwarder.

- **Contention Forwarding Policy**

If the unicast forwarding policy fails to unicast a DATA packet, the contention forwarding policy can employ ERTS-ECTS-DATA-ACK handshake to select a new next-hop forwarder. The delay constraint is used to determine whether a contention candidate can take part in the contention or not. ERTS is an extension to RTS with additional information including the positions of the current forwarder and the sink, and the required delay.

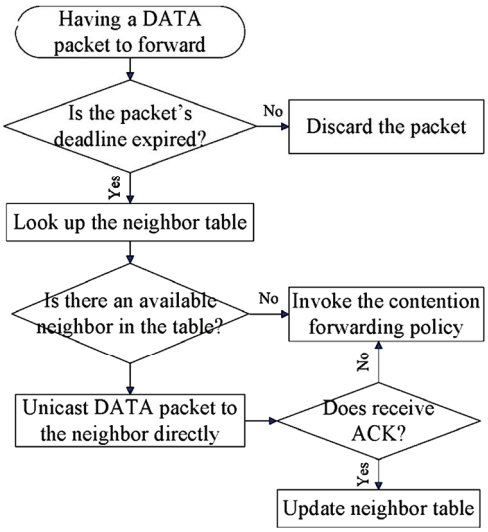


Figure 1. Unicast forwarding procedure.

ECTS is also an extension to CTS with additional position of CTS's sender. Busy tone is an out-of-band signal which is used to avoid multiple candidates taking part in the contention, simultaneously. The current forwarder and its contention candidates work as follows.

As shown in **Figure 2**, the current forwarder first broadcasts an ERTS packet to start a contention, and then waits for receiving an ECTS packet. If overhearing the ECTS packet, the current forwarder should send a busy tone immediately, and continues receiving the ECTS packet. After finishing receiving the ECTS, the current forwarder can unicast the DATA packet to the selected next-hop forwarder directly. When the current forwarder finishes receiving the ACK packet sent by the next-hop forwarder, it inserts the information of the new forwarder including the ID, position and one-hop delay, or updates the one-hop delay, in its neighbor table. If the current forwarder fails to select a next-hop forwarder, it just discards the DATA packet.

The contention candidate's contention procedure is shown in **Figure 3**. After receiving the ERTS packet, each contention candidate has to determine whether to participate in the contention or not according to its one-hop neighbor table. If its neighbor table has an available neighbor, which meets the speed constraint, the candidate can be assigned higher contention priority with short wait delay, T_{short}. If none of the neighbors in its one-hop neighbor table can meet the speed constraint, the candidate is forbidden to take part in the contention. Otherwise, if the neighbor table is empty, which means that the candidate has never been the forwarder, the candidate is also admitted to take part in the contention but with long wait delay, T_{long}. The candidate with the shortest wait delay can first respond an ECTS packet thus to win the qualification of the new forwarder. Other candidates, which receive the busy tone during their wait delay, must quit the contention immediately. When finishing receiving the DATA packet, the new forwarder should respond an ACK packet to the current forwarder.

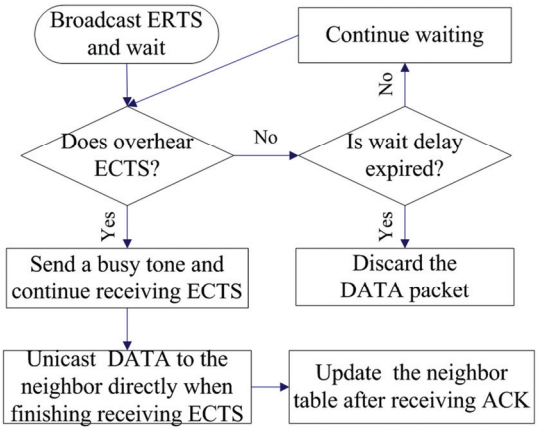


Figure 2. Current forwarder's working procedure.

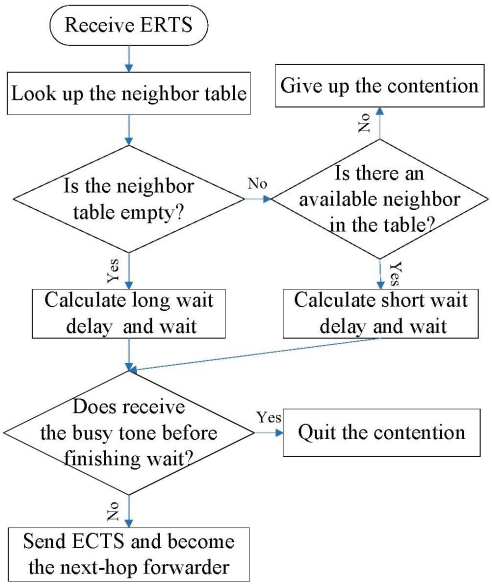


Figure 3. Contention candidate's contention procedure.

It needs to be pointed out that although the delay constraint never takes the wait delay into account, it does not affect the real-time performance because the selected neighbor is the most suitable forwarder, which has the shortest wait delay.

- **Contention Function and Wait Delay**

Contention function is the key factor to the contention-based beaconless scheme because it determines the wait delay of each contention candidate. Due to real-time constraint and energy efficiency, our contention function takes into account the combination of the progress distance toward a sink, the remaining energy of the node and the number of packets waiting in the output queue. For node i, its contention priority is computed as follows.

$$P_i = (q_i + 1)(\alpha d_i / R_c + \beta E_i / E_t + \gamma(1 - q_i / Q_t) + \eta r_i) \quad (1)$$

where P_i is the contention priority; d_i is the progress distance towards a sink; E_i is the remaining energy and E_t is the initial total energy; q_i is the number of packets waiting in the queue and Q_t is the total queue size; r_i is a random value between 0 and 1; α, β, γ and η are weights assigned to distance, energy, queue and random value, respectively, and meet $\alpha + \beta + \gamma + \eta = 1$.

Using the contention function, we can calculate the wait delay of T_{short} and T_{long} as follows.

$$T_{short} = SIFS + 0.5SIFS(1 - P_i) \quad (2)$$

$$T_{short} = SIFS + SIFS(1 - P_i) \quad (3)$$

where *SIFS* is Short Inter Frame Spacing and is defined as 10 μs in IEEE 802.11 standard [15]. From (2) and (3), it can be guaranteed that T_{short} is smaller than T_{long}.

- **Delay Estimation**

Delay estimation is responsible to calculate the one-

hop delay between two neighborhoods. The current one-hop delay is evaluated based on two timestamps. One is the time $T_{arriving}$ when the received DATA packet enters the tail of the output queue. Another is the time T_{ACK} when the node receives the ACK packet responded by the next-hop forwarder. The time interval between these two timestamps is employed to characterize the one-hope delay here.

The new one-hop delay can be calculated by the combination of the currently measured delay and the previous one-hop delay as follows.

$$Delay_{new}^{onehop} = \mu(T_{ACK} - T_{arriving}) + (1-\mu)Delay_{previous}^{onehop} \quad (4)$$

- **One-Hop Neighbor Table**

As shown as in **Table 1**, NeighborID and NeighborPosition are obtained through an ECTS packet; OneHopDelay can be computed from (4). Counter is used to record how many times a neighbor has been a next-hop forwarder.

One-hop Neighbor table is managed as follows.

1) If the newly selected forwarder is not in the neighbor table, the current forwarder has to add the information of the new forwarder in its neighbor table after receiving the ACK packet sent by the new forwarder. Otherwise, the current forwarder needs to update the OneHopDelay of the neighbor in its neighbor table.

2) If the Counter in the entry of a neighbor is up to MaxCount, the neighbor is forbidden to take part in the current forwarding, and its Counter should be cleared to zero. MaxCount is the maximal times that a node can be a next-hop forwarder. If the network topology is stable, a neighbor in the neighbor table can always be valid so as to be the next-hop forwarder continuously. In this case, the node not only consumes away its energy quickly, but also deprives other neighbors of being a next-hop forwarder. Therefore, MaxCount is used to balance the network load thus to prolong the lifetime of WSNs.

3.3. CBRR-TwoHop Protocol

CBRR-TwoHop protocol is an extension to CBRR-OneHop protocol. It can depend on wireless broadcast to collect the information of the two-hop neighborhoods for building two-hop neighbor table. With the help of two-hop neighbor table, CBRR-TwoHop is able to further meet the real-time requirements in the two-hop range.

- **Two-Hop Delay Estimation**

As shown in **Figure 4**, node A, B and C are regarded as the last-hop forwarder, the current forwarder and the next-hop forwarder, respectively. While node B unicasting a DATA packet to node C, node A can also overhear the DATA packet, which carries the position information of node C.

In order to obtain the two-hop delay, each node needs to create a Sent_DATA table for recording the DATA packets, which have been sent by the node successfully. As shown in **Table 2**, the record of a DATA packet includes the items of source's ID, sequence number and Received_Time, which is the timestamp that the DATA packet enters the tail of the output queue. With the help of the Sent_DATA table, the two-hop delay estimation between node A and C can be obtained as follows.

1) After receiving the ACK packet sent by node B, node A needs to add the relevant information of the DATA packet, which has been sent to node B previously, in its Sent_DATA table.

2) While node B unicasting a DATA packet to node C, node A can overhear the packet so as to look up its Sent_DATA table to see if A has sent this DATA packet previously. If true, the current two-hop delay between node A and C is approximated by the interval between the Received_Time in A's Sent_DATA table and the time T_{now} when node B finishes transmitting the DATA packet to node C. After estimating, node A needs to delete the entry of this DATA packet from its Sent_DATA table for reducing the storage overhead. However, if the DATA packet is not in the Sent_DATA table, node A has to ignore the packet and to do nothing.

We also compute the new two-hop delay by the combination of the currently measured delay and the previous two-hop delay as follows.

$$Delay_{new}^{twohop} = \mu(T_{now} - T_{\mathrm{Received\_Time}}) + (1-\mu)Delay_{previous}^{twohop} \quad (5)$$

- **Two-Hop Delay Estimation**

The two-hop neighbor table extends the one-hop neighbor table with additional information of the two-hop neighbor including ID, position and two-hop delay, as

Table 1. Structure of one-hop neighbor table.

NeighborID	NeighborPosition	OneHopDelay	Counter

Table 2. Structure of Sent_DATA table.

SourceID	SequenceNo.	Received_Time

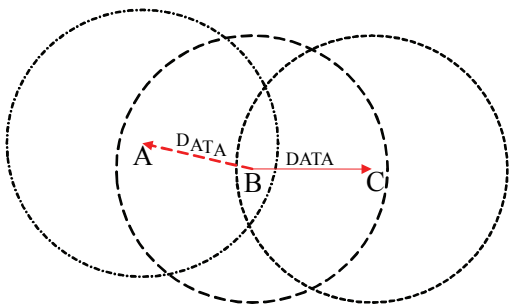


Figure 4. An example of wireless broadcast.

shown in **Table 3**. The ID and position of the two-hop neighbor are obtained through a DATA packet, and Two-Hop Delay can be computed from (5). Other items are the same as those in one-hop neighbor table.

It should be noted that if a node fails to select a next-hop forwarder, the node maybe encounters a void around it. In this case, the node will not unicast the received DATA packet, and its last-hop forwarder can never overhear the packet. As a result, the information of the two-hop neighbor should be null in the two-hop neighbor table of the last-hop forwarder. For avoiding more invalid contentions, if the two-hop table has more than two records, which have null information of the two-hop neighbor, the node maybe has a serious void problem in its two-hop range thus to be forbidden to take part in the latter contentions.

- **Forwarding policy**

The forwarding policy of CBRR-TwoHop is similar to that of CBRR-OneHop. The difference is that CBRR-TwoHop needs to employ the speed constraint between the two-hop neighborhoods. If there is a one-hop neighbor, which the two-hop relay speed between the current forwarder and the two-hop neighbor is no less than the required two-hop speed, in the two-hop neighbor table, the current forwarder can uincast the DATA packet to the one-hop neighbor directly. Otherwise, the current forwarder has to invoke the contention forwarding policy with the two-hop delay constraint to select a next-hop forwarder.

4. Experimental Studies

To validate the CBRR protocol proposed in the paper, CBRR-OneHop and CBRR-TwoHop are compared with SPEED. Experimental studies are conducted by means of J-Sim simulator [16], which is an open-source, component-based network simulation environment and is developed entirely in Java.

Table 3. Structure of two-hop neighbor table.

OneHop ID	OneHop Position	OneHop Delay	TwoHop ID	TwoHop Position	TwoHop Delay	Counter

Table 4. Parameters for experiments.

Parameter	Values	Parameter	Values
Network Size	200m×200m	MAC Layer	802.11
Node Number	200	Initial Energy	100 J
Radio Range	40 m	Bandwidth	2 Mbps
Packet Size	512 bytes	Send/Receive/Idle power(mW)	660/395/35

4.1. Simulation Setting

We randomly choose four source nodes in the left of network, and two sink nodes in the right of network. We test the above protocols in two network topologies:

- **Static network**, where the topology is invariable including packet generation rate and node density scenarios.
- **Dynamic network**, where the topology is changeable due to node mobility or sleep policy.

We choose constant bit rate (CBR) traffic and set CBR at 2 packets/s in all experiments except in the packet generation rate scenario. Unless specially noted, all parameters for experiments are set as shown in **Table 4**.

4.2. Performance in Static Networks

In static networks, the position of each node is not changeable. Therefore, we set the interval of beacon broadcast at 10 s in SPEED. Two scenarios are evaluated in the static networks including packet generation rate and packet size.

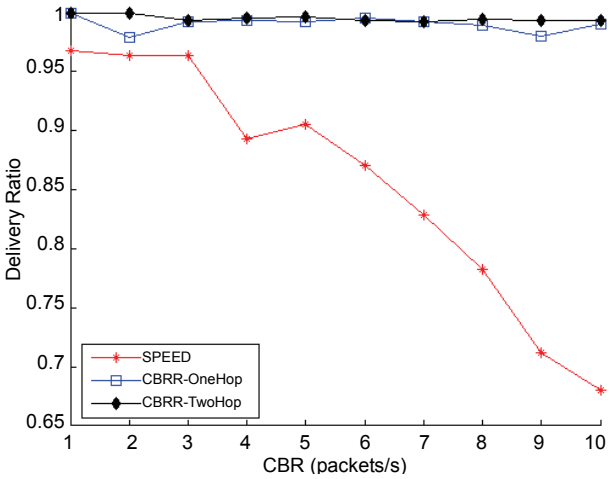
- **Impact of packet generation rate**

The comparative results between CBRR and SPEED are plotted in **Figure 5**. It can be seen that the both CBRR protocols achieve nearly 100% delivery ratio (**Figure 5(a)**) and stable end-to-end delay around 0.05s (**Figure 5(b)**). All these in CBRR contribute to the Routing/MAC cross-layer design, which can timely collect the state information of wireless channel thus to avoid more collisions during the forwarding procedures. In contrast, higher packet generate rate may bring forth more packet collisions thus lead to higher packet miss ratio (**Figure 5(a)**) and longer delay (**Figure 5(b)**) in SPEED. In **Figure 5(c)**, SPEED consumes the average energy about two times more than those of the two CBRR protocols due to its beacon broadcasting. It is worth noting that the average energy consumption of each protocol decreases slowly as the packet generation rate increasing. The reason is that when the packet generation rate is small, most nodes are always kept in idle state, whose energy consumption is the main part of the total consumed energy. However, more and more packets have been forwarded thus can lead to lower average energy consumption as the generate rate increasing. Furthermore, **Figure 5** illustrates that the overall performance of CBRR-TwoHop is little better than that of CBRR-OneHop because the two-hop neighbor table can be helpful for further meeting the real-time requirements in the two-hop range.

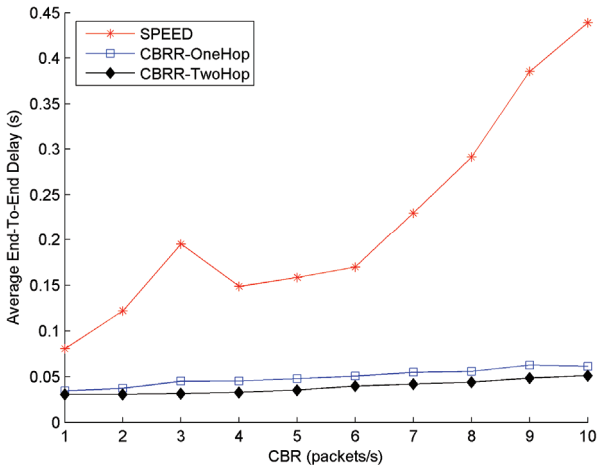
- **Impact of packet size**

Larger DATA packets may lead to higher probability of the collisions between a DATA packet and other packets, such as RTS/ERTS, CTS/ECTS, ACK or beacon. **Figure 6** illustrates the comparative results between

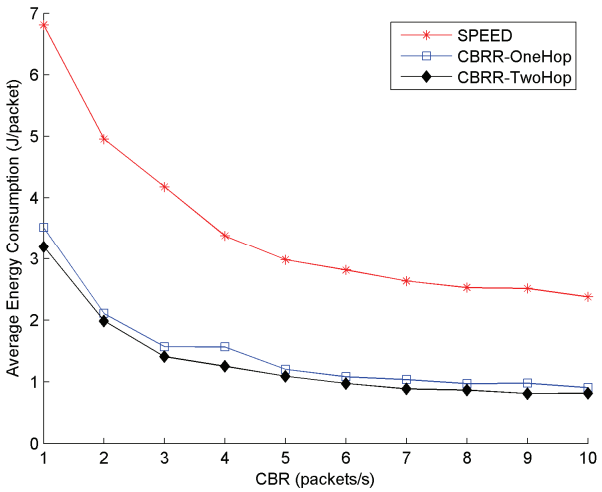
CBRR and SPEED in the packet size scenario. It can be observed in **Figure 6(a)** that the delivery ratio of SPEED



(a) Packet Delivery Ratio

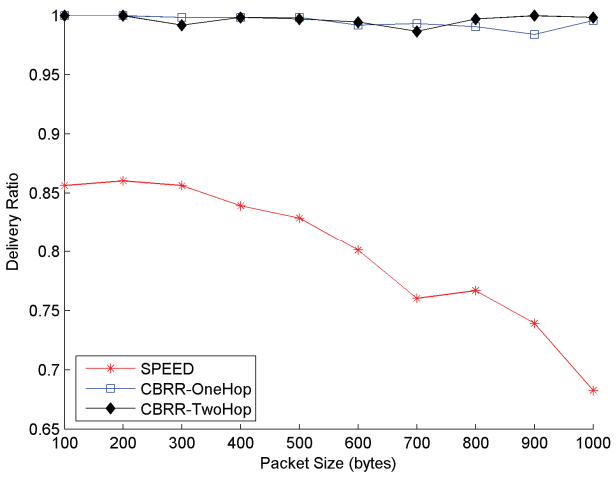


(b) Average End-to-End Delay

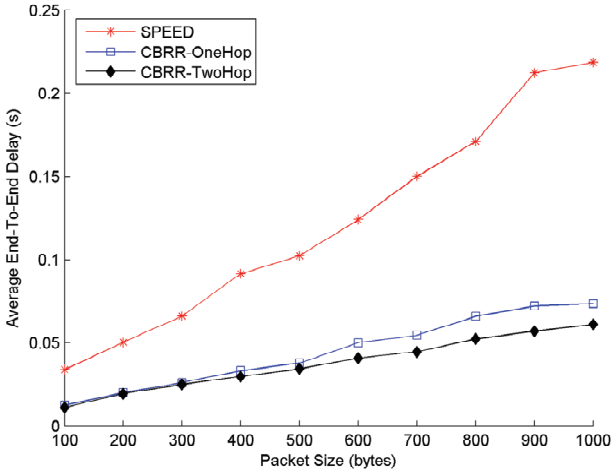


(c) Average Energy Consumption

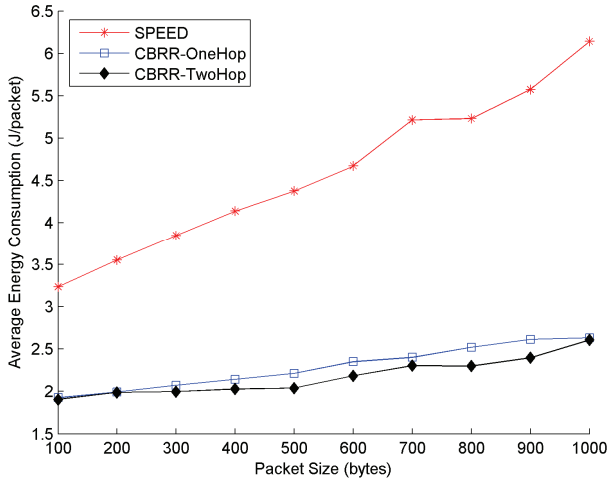
Figure 5. Impact of packet generation rate.



(a) Packet Delivery Ratio



(b) Average End-to-End Delay



(c) Average Energy Consumption

Figure 6. Impact of packet size.

drops quickly as the packet size increasing, but the packet size has very little impact on the two CBRR pro-

tocols for their nearly 100% delivery ratio. The results may indicate that the collisions between DATA packets and beacons are much more severe than others between DATA and RTS/ERTS, CTS/ECTS or ACK packets. As a result, with larger packet size, more serious packet collisions can lead to longer end-to-end delay (**Figure 6(b)**) and more energy consumption (**Figure 6(c)**) than those of the two CBRR protocols.

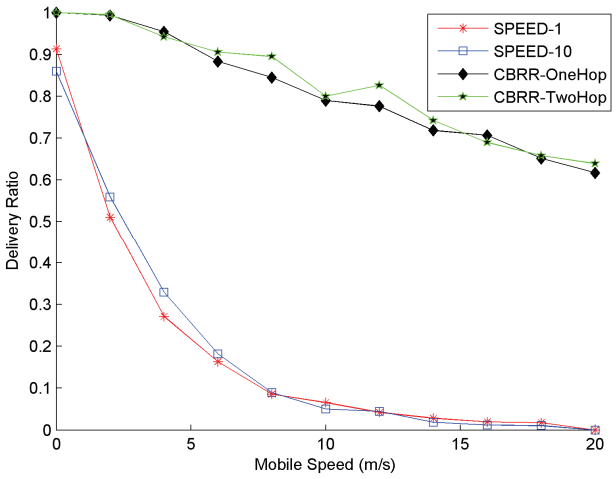
It can be viewed from above experimental results that the two CBRR protocols have much better performance than SPEED in the static networks. Furthermore, CBRR-TwoHop can outperform than other two protocols due to the help of the two-hop neighbor table, which can be helpful for further meeting the real-time requirements in the two-hop range. In addition, it can also be suggested that broadcasting beacons can aggravate the packet collisions thus to degrade the performance of the beacon-based routing protocols.

4.3. Performance in Dynamic Networks

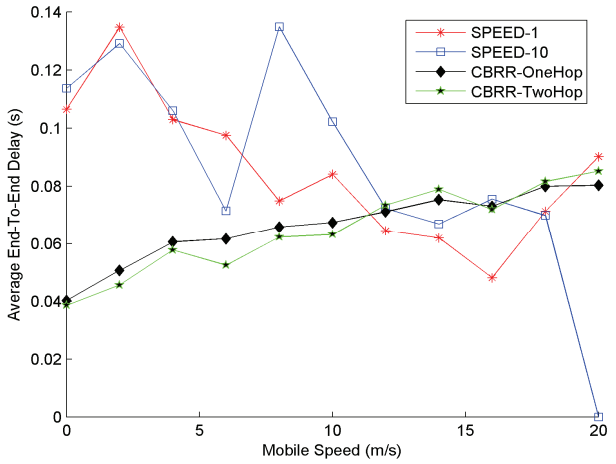
WSNs are highly dynamic networks and their topologies can change constantly due to node mobility, sleep policy, node failure, and so on. In the following experiments, we compare CBRR with SPEED in node mobility and sleep policy scenarios. In addition, we set the frequency of broadcasting beacons at 1 s and 10 s, respectively, for SPEED to timely collect fresh information of the neighborhoods.

- **Impact of node mobility**

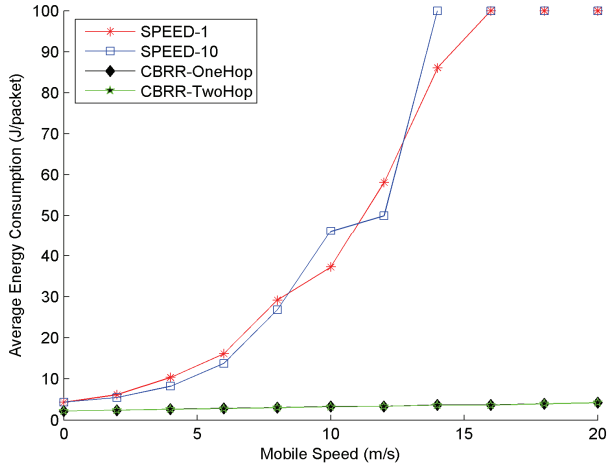
We adopt the Random Waypoint mobility model with zero pause time in the experiment. **Figure 7** plots the comparative results between the two CBRR protocols and SPEED in the node mobility scenario. **Figure 7(a)** illustrates that the delivery ratio of SPEED drops much more quickly, which shows that SPEED can hardly work when the mobile speed is higher than 10 m/s. This is because the information of the neighbor table is totally outdated thus to be useless for the forwarding. In contrast, although the two CBRR protocols drop more packets at higher mobile speed, they can still achieve about 60% delivery ratio at 20 m/s in contribution to the contention forwarding policy, which can on-line select a favorite next-hop forwarder. It can be observed in **Figure 7(b)** that the average end-to-end delay of SPEED becomes much instable for its extremely high packet miss ratio. However, at higher node mobility, the both CBRR protocols have little longer delay due to the failure of direct unicast. **Figure 7(c)** shows that the two CBRR protocols have very close performance, which consume far less energy than SPEED. It needs to point out that we set 100 as the maximum in **Figure 7(c)**, and the values of SPEED, which is plotted as 100, are actually more than 100. Furthermore, it can be observed in **Figure 7** that the performance of SPEED-1 is no better than that of SPEED-10, which suggests that higher frequency of beacon broadcasting can not improve the performance of SPEED in the node mobility scenario.



(a) Packet Delivery Ratio



(b) Average End-to-End Delay



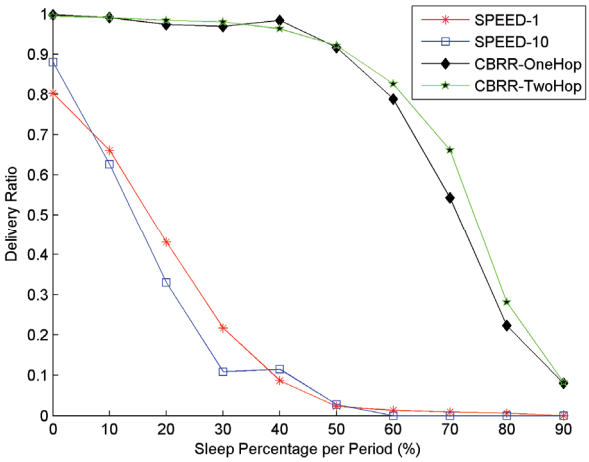
(c) Average Energy Consumption

Figure 7. Impact of node mobility.

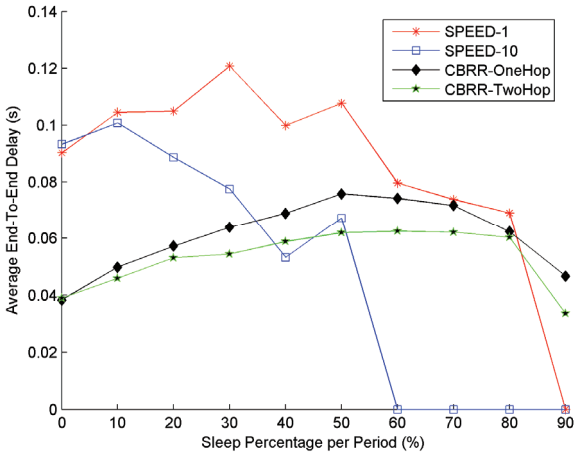
- **Impact of sleep policy**

In order to support energy conservation in WSNs, the most practical way is to use node sleep policy. Our sleep policy is designed as that the lifetime of each node is divided into several same periods, and each period includes an active sub-period and a sleep sub-period.

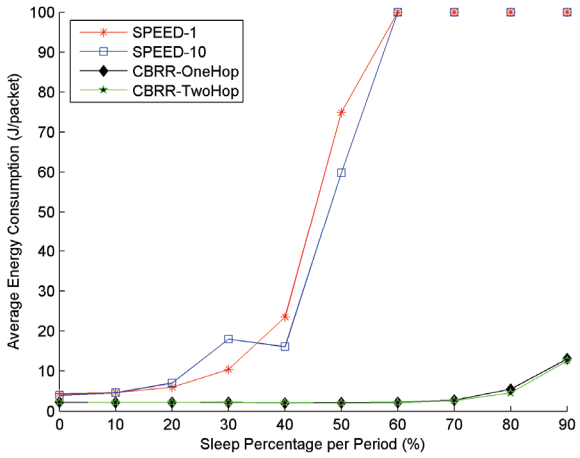
Figure 8 plots the comparative results between the two CBRR protocols and SPEED in the dynamic sleep policy scenario. Similar to the node mobility scenario, the sleep policy has far more negative impact on SPEED than those on the two CBRR protocols. The impact can be observed in **Figure 8(a)** that if the sleep percentage is larger than 50%, the delivery ratio of SPEED is approximate 0. The reason is that the next-hop forwarder, which is selected by SPEED in the neighbor table, would be usually in sleep state, and SPEED needs to retransmit more packets. However, the both CBRR protocols achieve more than 90% in delivery ratio at 50% sleep percentage. Although dropping more packets after 50%, the two CBRR protocols still outperforms SPEED very much. Other results are similar to those in the node mobility scenario as shown in **Figure 8(b)** and **Figure 8(c)**.



(a) Packet Delivery Ratio



(b) Average End-to-End Delay



(c) Average Energy Consumption

Figure 8. Impact of sleep policy.

It can be concluded from the above experimental results that CBRR is not only particularly suitable for the dynamic networks, but also has fair well performance than SPEED in the static scenarios with much less energy consumption. Furthermore, it can be observed that CBRR-TwoHop outperforms the other two protocols due to the help of the two-hop neighbor table. It also suggests that SPEED is totally not suitable for the dynamic networks, and increasing the frequency of beacon broadcasting can not improve but degrade the performance of the beacon-based routing protocols.

5. Conclusions

This paper presents a novel real-time routing protocol, CBRR, for WSNs. The point distinguishing our approach from the existing schemes is that CBRR collects the information of neighborhoods by the contention mechanism instead of beacons. This contribution can lead to more energy efficiency. In addition, it is notable that the end-to-end real-time requirements are well fulfilled with speed or delay constraint at each hop, which attributes to the combination of the contention and neighbor table mechanisms. The validity of CBRR is illustrated by means of experimental studies. It has been shown that CBRR can outperform SPEED in terms of delivery ratio, end-to-end delay and energy consumption, especially in dynamic networks.

Our future work is to conduct the theoretical analysis on the energy consumption of CBRR. It is also interested to investigate how to provide reliability in CBRR.

6. References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, Vol. 38, No. 4, March 2002, pp. 393-422.

[2] D. Z. Cheng and P. K. Varshney, "QoS Support in Wireless Sensor Networks: A Survey," *International Conference on Wireless Networks* (*ICWN*'04), LasVegas, Vol. 1, June 2004, pp. 227-233.

[3] Y. J. Li, C. S. Chen, Y. Q. Song and Z. Wang, "Real-time QoS Support In Wireless Sensor Networks: A Survey," *Proceedings of Seventh IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems*, Toulouse, November 2007, pp. 373-380.

[4] M. Younis, K. Akayya and A. Wadaa, "On Handling QoS Traffic in Wireless Sensor Networks," *Proceedings of the* 37*th Annual Hawaii International Conference on System Sciences*, Hawaii, January 2004, pp. 292-301.

[5] T. He, J. A. Stankovic, C. Y. Lu and T. F. Abdelzaher, "A Spatiotemporal Communication Protocol for Wireless Sensor Networks," *IEEE Transaction on Parallel Distributed Systems*, Vol. 16, No. 10, October 2005, pp. 995-1006,.

[6] E. Felemban, C. G. Lee and E. Ekici, "MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, Vol. 5, No. 6, June 2006, pp. 738-754.

[7] O. Z. Chipara, Z. M. He, G. L. Xing, Q. Chen and X. R. Wang, "Real-Time Power-Aware Routing in Sensor Networks," *Proceedings of* 14*th IEEE International Workshop on Quality of Service* (*IWQoS* 2006), New Haven, June 2006, pp. 83-92.

[8] T. He, B. M. Blum, Q. Cao, J. A. Stankovic, S. H. Son and T. F. Abdelzaher, "Robust and Timely Communication over Highly Dynamic Sensor Networks," *Real-Time Systems*, Vol. 37, No. 3, December 2007, pp. 261-289.

[9] M. Zorzi and R. R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Delay Performance," *IEEE Transactions on Mobile Computing*, Vol. 2, No. 4, October-December 2003, pp. 349-365.

[10] M. Zorzi and R. R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance," *IEEE Transactions on Mobile Computing*, Vol. 2, No. 4, October-December 2003, pp. 337-348.

[11] P. Huang, H. Y. Chen, G. L. Xing and Y. D. Tan, "SGF: A State-Free Gradient-Based Forwarding Protocol for Wireless Sensor Networks," *ACM Transaction on Sensor Networks*, Vol. 5, No. 2, March 2009, pp. 1-25.

[12] D. Z. Chen and P. K. Varshney, "On-Demand Geographic forwarding for Data Delivery in Wireless Sensor Networks," *Computer Communications*, Vol. 30, No. 14-15, October 2007, pp. 2954-2967.

[13] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network," *IEEE Personal Communications*, Vol. 7, No. 5, October 2000, pp.16-27.

[14] K. Akkaya and M. Younis, "Energy-Aware QoS Routing in Wireless Sensor Networks," *Cluster Computing*, Vol. 8, No. 2-3, July 2005, pp. 179-188.

[15] "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ISO/IEC 8802-11: 1999(E)," IEEE, August 1999.

[16] "DRCL J-Sim," 2005. http://j-sim.cs.uiuc.edu/index.html

