Cooperative Contention-Based Forwarding for Wireless Sensor Networks *

Long Cheng†‡, Jiannong Cao‡, Canfeng Chen†§, Hongyang Chen†, Jian Ma§, Joanna Izabela Siebert‡
†State Key Lab of Networking & Switching Tech., Beijing Univ. of Posts and Telecomms., China
‡Department of Computing, Hong Kong Polytechnic University, Hong Kong
§Nokia Research Center, Beijing, China
† Institute of Industrial Science, The University of Tokyo, Tokyo, Japan
Email: {csrcheng, csjcao, csjsiebert}@comp.polyu.edu.hk, {canfeng-david.chen, jian.j.ma}@nokia.com, hongyang@ncl.iis.u-tokyo.ac.jp

ABSTRACT
Cooperative forwarding has been considered as an effective strategy for improving the geographic routing performance in wireless sensor networks (WSNs). However, we observe that existing works do not fully utilize the forwarding opportunities provided by available neighboring nodes. In this paper, we redesign the cooperative forwarding process and present a novel cooperative contention-based forwarding (CCBF) protocol for WSNs. CCBF extends the scope of cooperation and attains the full potential of cooperative forwarding at the expense of sending one additional control message on demand. We conduct extensive simulations to study the performance of the proposed CCBF compared with existing protocols. Simulation results demonstrate that CCBF decreases the end-to-end hop counts, hence, further improves the end-to-end energy efficiency and latency. Remarkably, it provides up to 50% improvement for the packet loss ratio when the retransmission mechanism is not adopted.

Categories and Subject Descriptors
C.2.1 [Computer-Communication Network]: Network Architecture and Design—Wireless communication
C.2.1 [Computer-Communication Network]: Network Algorithms, Design

Keywords
Wireless sensor networks; Cooperative geographic routing; Contention-based forwarding

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IWCMC’10, June 28-July 2, 2010, Caen, France
Copyright 2010 ACM 978-1-4503-0062-9/10/06 ...$5.00.

1. INTRODUCTION
Geographic routing is quite commonly adopted for information delivery in wireless sensor networks (WSNs) where nodes need to know only the location information of their direct neighbors and where the destination is. The basic idea for geographic routing is greedily forwarding data packets to the neighbor geographically closest to the destination [1]. However, several recent studies [2] have stressed that the realistic link conditions in wireless networks can be highly unreliable due to many factors such as interference, attenuation, and fading. For each hop in greedy forwarding, although the neighbor closest to the destination has the maximum advance, it is also likely to be farthest from the current forwarding node and may have poor link quality. In this case, the maximum-distance greedy forwarding will perform poorly in realistic conditions as it tends to forward packets on lossy links, which may cause the drastic reduction of packet delivery rate and increase the energy consumption of sensor nodes if retransmission mechanism is adopted.

This observation brings to the fore the concept of the cooperative geographic routing in WSNs, which takes advantage of the broadcast nature of wireless communication. All nearby nodes which overhear transmissions can be used as potential forwarders. For each hop, it dynamically chooses the forwarder closer to the destination based on node availability, exploiting the occasionally successful transmissions over unreliable links. Obviously, this routing mechanism would be attractive in dense sensor networks. However, applying cooperative routing in WSNs requires all neighbors work in close cooperation, otherwise packet duplications and collisions will be introduced, which instead deteriorate the routing performance. Although cooperative forwarding has been considered as an effective strategy for improving the routing performance in WSNs [3], we observe that existing studies [4] [5] do not fully utilize the forwarding opportunities provided by available neighboring nodes. We argue that further gain of cooperative forwarding is possible.

In this paper, we present a cooperative contention-based forwarding (CCBF) protocol to improve the cooperative geographic routing performance in WSNs, which extends the scope of cooperation and also avoids packet duplications. CCBF differs from [4] and [5] in that the cooperation scope is not limited within a certain range of the next-hop receiver (we call the next-hop centered cooperative forwarding), thus it...
fully utilizes the forwarding opportunities provided by available neighboring nodes.

The remainder of this paper is organized as follows. Section 2 surveys related work. Section 3 elaborates the detailed algorithm design of CCBF. Section 4 provides the simulation results followed by the conclusions in Section 5.

2. RELATED WORK

The broadcast nature of wireless communications suggests that a sender signal transmitted towards the destination can be "overheard" at neighboring nodes. Cooperative communications have been proposed to take advantage of broadcast transmission to send packets through multiple neighbors' cooperation [6]. The majority of existing literature tackles the issue from the physical layer perspective, where the receiver is capable of combining and decoding the signal from several simultaneous transmissions [7]. In this paper, we consider cooperative forwarding in WSNs from a MAC-layer perspective, which means a receiver that can only decode one transmission at a time [8].

Two recent protocols, cluster-based forwarding (CBF) [4] and [5] (we refer to as CRL), have been proposed for cooperative forwarding in WSNs. In CBF, each node forms a cluster such that any node in the nexthop's cluster may take forwarding responsibility. As shown in Fig. 1(a), when A transmits a packet to B, C is an intermediate "helper" that works by shifting the forwarding task from a weak link, AB, to a strong link, CB; D is a distant "helper" that is exploited opportunistically if it receives A's packet. In CRL (see Fig. 1(b)), when the nexthop receiver B fails to receive a packet from A, as a relay node, C will forward the packet to B (still taking B as the nexthop). After the retransmission from a relay, there may be a node D that is further advanced towards the destination than the nexthop receiver B that has successfully received the packet. As a leapfrog (LPF) node, D will take on the role of a new sender and attempt to forward the packet. It can be seen that both [4] and [5] are of the nexthop centered cooperative forwarding.

![Cooperative forwarding examples in CBF and CRL](image)

**Figure 1:** Cooperative forwarding examples in CBF and CRL. Both are of the nexthop centered cooperative forwarding, the upper bound the cooperation scope is within the radio range of the nexthop receiver.

3. PROTOCOL DESCRIPTION

In this section, we present the design of the cooperative contention-based forwarding (CCBF) protocol.

3.1 Observations

Before providing the detailed design of CCBF, we first illustrate some intuitive observations that motivate the design of our protocol. Consider an example scenario as shown in Fig. 2, where sender A transmits a packet to the nexthop receiver B at time $t_0$. B fails to receive the data, while C and D successfully receive the data that B has lost. Thus, C and D contend with each other to be the "intermediate helper" of B. As a winner of the contention, C relays the data packet to B as an "intermediate helper" at time $t_1$. When C retransmits the data to B, E also overhears the data. Thus, E will take the forwarding responsibility as a "distant helper" of B by sending an ACK to C. Unfortunately, however, D knows nothing that C has taken the forwarding responsibility as an "intermediate helper". Therefore, D also relays the data packet to B as an

![Cooperative forwarding examples in CBF and CRL](image)

**Figure 2:** An example illustrating the potential room for improvement in CBF and CRL, both CBF and CRL limit the cooperation scope.

Assume that B and E fail to receive a packet from A, but C receives the packet. In CBF, C will not be considered as an "intermediate helper" if it is not located within the fully connected region of B. In CRL, as a relay node, C participates in the relay contention and will retransmit the lost data to B. During the retransmission from C to B, D also overhears the data. In fact, compared with B, D is more appropriate to be the nexthop forwarder of C. However, in CRL, D will not be considered as a "distant helper" (called LPF in CRL) of B. This is because D is located out of the radio range of B, thus D cannot take the forwarding responsibility as a "distant helper". Both CBF and CRL limit the cooperation scope within certain range of the nexthop receiver.

Since C has successfully received the data packet, we observe that it would be better if C takes forwarding responsibility directly as a new sender, instead of still taking B as the nexthop receiver.

Another example scenario is shown in Fig. 3, which illustrates the "hidden helper" problem. Sender A transmits a packet to the nexthop receiver B at time $t_0$, B fails to receive the data, while C and D successfully receive the data that B has lost. Thus, C and D contend with each other to be the "intermediate helper" of B. As a winner of the contention, C relays the data packet to B as an "intermediate helper" at time $t_1$. When C retransmits the data to B, E also overhears the data. Thus, E will take the forwarding responsibility as a "distant helper" of B by sending an ACK to C. Unfortunately, however, D knows nothing that C has taken the forwarding responsibility as an "intermediate helper". Therefore, D also relays the data packet to B as an
“intermediate helper” at time $t_2$, which will cause the packet duplications. We observe that the packet duplications can be avoided if the nexthop receiver broadcasts a notification message to its potential helpers.

3.2 Protocol Design

From the observations, we know that the cooperation scope is restricted within the cluster (always within the fully connected region) of the nexthop receiver in CBF. In CRL, in order to avoid the “hidden helper” problem, the radius of the cooperation scope must be less than the half radio range. The design objective of CCBF is to extend the scope of cooperation and also avoids packet duplications.

CCBF is composed of two contention-based phases: nexthop selection and cooperative forwarding. The nexthop selection phase is similar with other contention-based forwarding protocols [9]. CCBF differs from existing protocols in the cooperative forwarding phase.

3.2.1 Nexthop selection

In CCBF, there is no need to maintain one-hop neighbor tables, the nexthop selection is based on an RTS/CTS exchange on demand, which is more adaptive to topology changes. When a sender has a packet to send, it first listens to check whether the channel is clear. If it is, it broadcasts an RTS message, which contains the location of the destination as well as its own and the identifier of the data packet, then waits for response for a predefined maximum time of $T_{\text{max}}$. If the channel is busy, the sender backs off and reschedules another attempt at a later time. The sender’s neighbors who receive the RTS contend to serve as the nexthop receiver. To avoid radio interference, the contention process is achieved by calculating a metric for assigning different time slots to coordinate between neighbors.

Each neighbor receiving an RTS calculates a metric \( EADV \) (expected advance) defined as (1) [10] in a distributed fashion. We use the expected transmission count as the link cost, the packet reception ratio (PRR) of a link can be estimated by the link quality indicator (LQI) or received signal strength indicator (RSSI) reported by the physical layer of IEEE 802.15.4.

$$EADV(s, n, d) = \frac{\text{Dist}(s, d) - \text{Dist}(n, d)}{1 - \text{PPR}}$$

where $s$ and $d$ are the sender and the destination respectively, $n$ is the node that calculates this metric and $\text{Dist}(a, b)$ represents the Euclidean distance between $a$ and $b$.

After computing the metric, each neighbor that has positive $EADV$ starts a timer whose value is defined as (2). Let $r$ denote the radio range of a node. The larger $EADV$ is, the earlier time slot it will assign.

$$t(s, n, d) = (1 - \frac{EADV(s, n, d)}{r}) \cdot T_{\text{max}} + i \cdot \text{random}(0, \tau)$$

where $T_{\text{max}}$ is a constant representing the maximum delay time that a forwarding node will wait for the answer from its neighbors, $i$ is the times sender broadcasting the RTS, and $\tau$ is a parameter much smaller than $T_{\text{max}}$. It is possible that multiple neighbors have the same $EADV$ and thus send CTS simultaneously, the random function aims to mitigate the radio interference. If CTS collision occurs, the sender will rebroadcast an RTS message including a collision counter.

When the timer expires, the neighbor will broadcast a CTS message, which contains its own position. At the same time, other contenders will cancel their timers if overhearing this CTS. The neighbor first replying a CTS wins the competition to be the nexthop receiver. Once receiving a CTS, the sender will immediately send the data packet to the nexthop receiver. Notice that it is possible some neighbors cannot overhear the replied CTS because of their positions. However, they will cancel the timers once overhearing the data packet from the sender.

If the sender does not receive a CTS within $T_{\text{max}}$ time, it will rebroadcast the RTS message up to a maximum times. After that, if the sender has not received a CTS message, it means that the sender has no neighbors providing positive advance toward the destination. In this case, a recovery mechanism should be adopted to circumvent the so-called “holes” in geographic routing.

3.2.2 Cooperative forwarding

We introduce the dynamic “cooperative area” in CCBF to avoid packet duplications. The primary aim of selecting the nexthop receiver is to dynamically determine the “cooperative area” as shown in Fig. 4. Only neighbors located in the “cooperative area” are the potential helpers that will contend to be a forwarder. They are required to satisfy following three conditions: 1) have successfully received a data packet from the sender and stored the data packet; 2) have received the corresponding CTS from the nexthop receiver; 3) have positive advances toward the destination. It is noteworthy that the nexthop receiver is not necessarily the actual forwarder, since it depends on the availability of neighbors and successfully receiving data. Therefore, successful receiving of a data packet by the selected nexthop receiver is not required.

![Figure 4: An example illustrating the “cooperative area” in CCBF](image)

Similar to the nexthop selection phase, after the sender transmits a data packet to the nexthop receiver, all nodes in “cooperative area” that have successfully received the data will contend to serve as the actual forwarder. The data packet contains the location of the destination and the sender, the identifier of the data packet, and designates the nexthop receiver. The cooperative forwarding process is also achieved by timer-based contention, however, since competitors are those who have held the data packet, the metric computed at each node considers only the packet advance.

$$ADV(s, n, d) = \text{Dist}(s, d) - \text{Dist}(n, d)$$

The node which has maximum $ADV$ defined as (3) assigning the earliest time slot becomes a new sender. When the
timer expires, it will broadcast an ACK to the sender (for reducing overhead cost, the following RTS can also serve as an ACK). The ACK contains a flag bit to designate if it is the selected nexthop receiver. There are two cases in terms of the actual forwarder.

![Figure 5: Sender A transmits a packet to the nexthop receiver B and it serves as the actual forwarder. In this case, there is no need to send an additional control message.

<table>
<thead>
<tr>
<th>0</th>
<th>SIFS</th>
<th>DIFS</th>
<th>SIFS</th>
<th>DIFS</th>
<th>SIFS</th>
<th>DIFS</th>
<th>RTS</th>
<th>CTS</th>
<th>DATA</th>
<th>ACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>SIFS</td>
<td>DIFS</td>
<td>SIFS</td>
<td>CONF</td>
<td>SIFS</td>
<td>DIFS</td>
<td>RTS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>SIFS</td>
<td>CTS</td>
<td>SIFS</td>
<td>ACK</td>
<td>DIFS</td>
<td>RTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a) As shown in Fig. 5, the nexthop receiver serves as the actual forwarder. If the ACK is sent by the nexthop receiver to the sender, this ACK can suppress other potential helpers’ competitions. Then, it will broadcast an RTS to find its nexthop receiver.

![Figure 6: Sender A transmits a packet to the nexthop receiver B, a helper C serves as the actual forwarder. In this case, the sender will send an additional control message on demand.

<table>
<thead>
<tr>
<th>0</th>
<th>SIFS</th>
<th>DIFS</th>
<th>SIFS</th>
<th>DIFS</th>
<th>SIFS</th>
<th>DIFS</th>
<th>RTS</th>
<th>CTS</th>
<th>DATA</th>
<th>ACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>SIFS</td>
<td>DIFS</td>
<td>SIFS</td>
<td>CONF</td>
<td>SIFS</td>
<td>DIFS</td>
<td>RTS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>SIFS</td>
<td>CTS</td>
<td>SIFS</td>
<td>ACK</td>
<td>DIFS</td>
<td>RTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>SIFS</td>
<td>DIFS</td>
<td>SIFS</td>
<td>CONF</td>
<td>SIFS</td>
<td>DIFS</td>
<td>RTS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b) As shown in Fig. 6, one of the potential helpers serves as the actual forwarder. If the ACK is sent by a helper, in order to solve the potential “hidden helper” problem, the first time slot will be reserved for the sender to broadcast a confirmation (CONF) massage to notify other competitors to cancel their timers. Then it enters into a new nexthop selection phase. The neighbors receiving the following RTS contend with each other to serve as a new nexthop receiver.

4. SIMULATION

In order to evaluate the further gain by redesigning the cooperative routing process, in this section, we present simulation results to evaluate the performance of CCBF using ns-2 by comparing it with CBF [4] and CRL [5]. All the results have been averaged over 100 runs.

4.1 Simulation settings

In the implementation of our simulation, sensor nodes are randomly placed in a 200m × 200m field. The random topologies are all generated by the setdest tool in ns-2. A sink node is positioned at bottom left (0m, 0m), and the source node is located at the top right (200m, 200m). A data packet generated by the source node is forwarded toward the sink over multiple hops. The sensor transmission range r is taken 40m. Let γ denote the packet size ratio between a control message and a data packet. In our simulation, γ is 0.1. T_{max} is set 0.01s. The MAC layer protocol is a modified version of IEEE 802.11 MAC in ns-2, T_{SIFS} = 10us, T_{DIFS} = 50us. We set no retransmission, that is, if a sender fails to get an ACK, the data packet will be dropped.

We use the Nakagami distribution defined as (4) to describe the power x of a received signal:

\[ f(x, m, \Omega) = \frac{m^m x^{m-1}}{\Gamma(m)\Omega^m} \exp(- \frac{mx}{\Omega}) \]  

where Γ is the Gamma function, m denotes the Nakagami fading parameter and Ω is the average received power. We set m = 1 in our simulation. Assuming TwoRayGround signal propagation, Ω can be expressed in (5) as a function of d, the distance between the sender and receiver.

\[ \Omega(d) = \frac{P_t G_t G_r h^2 h_f^2}{d^5 L} \]  

where P_t is the transmission power, G_t and G_r are the antenna gains, h_t and h_r are the antenna heights, L is the loss factor, and n is the path-loss exponent. We set G_t = G_r = 1, h_t = h_r = 1.5m, L = 1, and n = 4 in our simulation.

We assume a packet is received successfully if the received signal power is greater than the receiving power threshold. Then by using (4) and (5), we can derive the PRR at a certain distance d.

4.2 Evaluation metrics

We select four evaluation metrics: 1) average advance per hop; 2) end-to-end delivery delay; 3) normalized sending cost; 4) packet loss ratio. These metrics measure the effectiveness of a cooperative forwarding protocol. The average advance per hop is the ratio of the sum of one-hop advances to the number of hops. The end-to-end delay metric measures how long it takes for a data packet to arrive at the sink from the source node. The cost of a transmission consists of the sending cost of the sender, and the receiving cost of one-hop neighbors. To some extent, the transmission cost is proportional to the sending cost. Hence, the sending cost metric measures the communication overhead and energy efficiency. We define the normalized energy cost for sending a data packet as one cost unit. Therefore, the energy cost for sending a control packet is 0.1 (γ = 0.1) cost unit. The packet loss ratio is defined as the ratio between the number of lost packet and the total number of packet sent by the sender.

4.3 Simulation results

We evaluate the performances of CCBF, CBF and CRL under different node densities. In this paper, the node density is defined as the number of nodes deployed in a 200m × 200m field. Fig. 7 illustrates the changes of the average advance per hop under different node densities. We observe that, for CCBF and CBF, the average advance per hop increases as the node density increases. However, for CRL, the change of node density does not affect the average advance per hop. This is partly because CRL neglects the occasionally successful transmissions over long but unreliable links, it always chooses the node with maximum value of the defined metric in the nexthop selection phase, instead of the node with maximum advance. Compared with CBF and CRL, CCBF significantly increases the advance per hop, which a direct benefit is the decrease of hop counts. This is be-
cause CCBF always chooses the neighbor which receives a data successfully and has the maximum advance as the actual forwarder, it fully utilizes the forwarding opportunities provided by available neighboring nodes.

Fig. 8 and Fig. 9 show the end-to-end delivery delay and the normalized sending cost as the node density changes. It is seen that CCBF outperforms CBF and CRL for both metrics. This is because CCBF increases the average advance per hop which means decreasing the hop counts. Thus, the end-to-end delivery delay and normalized sending cost are decreased. From section 3, we know CCBF sends more control messages (the confirmation message) compared with CBF and CRL when a helper serves as the actual forwarder. However, it is obvious that this tradeoff is worthwhile since the end-to-end energy efficiency and latency have been improved.

![Figure 7: Average advance per hop](image1)

![Figure 8: End-to-end delivery delay](image2)

![Figure 9: Normalized sending cost](image3)

![Figure 10: Average packet loss ratio](image4)

Fig. 10 illustrates the packet loss ratio of three protocols under different node densities. We also observe that, with the increase of the node density, the packet loss ratio of all three protocols decreases remarkably, take CCBF for example, from 0.68 to 0.01. It is shown that CCBF achieves its advantage over CBF and CRL. Averagely, it can provide up to 50% improvement for the packet loss ratio.

5. CONCLUSION

In this paper, we proposed CCBF, a novel cooperative contention-based forwarding protocol for geographic routing in WSNs. The feature of CCBF is that it extends the cooperation scope at the same time avoids packet duplications, therefore, fully utilizes the potential forwarding opportunities among nearby nodes. We compared CCBF with the existing cooperative forwarding protocols using ns-2. Simulation results demonstrated that our proposed CCBF could effectively improve the routing performance. Compared with existing protocols, it increased the average advance per hop, further improved the end-to-end energy efficiency and latency. Remarkably, it could provide up to 50% improvement for the packet loss ratio when the packet retransmission mechanism is not adopted.

6. REFERENCES