Research Article

A Reliable Data Collection Protocol Based on Erasure-Resilient Code in Asymmetric Wireless Sensor Networks

Jian-Jun Lei,¹ Taehyun Park,² and Gu-In Kwon²

¹ School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
² School of Computer and Information Engineering, Inha University, Incheon 402-751, Republic of Korea

Correspondence should be addressed to Gu-In Kwon; gikwon@inha.ac.kr

Received 23 December 2012; Revised 14 March 2013; Accepted 29 March 2013

Academic Editor: Adel Soudani

Copyright © 2013 Jian-Jun Lei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents RECPE, a reliable collection protocol for aggregating data packets from all the sensor nodes to the sink in a large-scale WSN (wireless sensor network). Unlike some well-known reliable data collection protocols such as CTP (Collection Tree Protocol) that uses ETX (expected transmission count) as the routing metric, RECPE exploits ETF (expected transmission count over forward links) to construct a one-way collection tree, which avoids missing some good routes and reduces the effect of asymmetric link in the network. Crucially, RECPE guarantees the reliability by erasure-resilient codes in the application layer without retransmission required by other reliable protocols. Therefore, some lower layers such as data link layer only need to conduct best-effort data delivery. Meanwhile, to improve efficiency, RECPE also exploits Trickle algorithm to reduce routing beacons and pipeline data delivery to prevent self-interference. We evaluate the performance of RECPE via TOSSIM simulations, and our results show that, in comparison with CTP (the de facto data collection protocol for TinyOS), RECPE can obtain significant performance in terms of delivery cost, latency, and packet loss rate for reliable data collection especially in asymmetric link networks.

1. Introduction

Many-to-one data collection is an important issue for WSN applications and protocols. Collection tree provides efficient approach for the higher layer routing and transport protocol. Most metrics employed in constructing collection tree from sensors to sink node in recent years tend to rely on the minimum hop-count [1, 2] or ETX [3, 4].

In high-power stable wireless networks, data loss is infrequent and hop-count can adequately capture the cost of packet delivery to destination. However, with lossy link, as found more often in WSN, each hop may require one or more retransmissions to compensate for the lossy channel. The real cost of packet delivery can be much larger than the hop distance. Therefore, the minimum hop-count ignores the possibility that a longer path might offer more efficient route and higher throughput. Additionally, the minimum hop-count also chooses arbitrarily among the different paths of the same minimum length, regardless of the often large difference in link quality among these routes. One alternative metric to fix this problem is to use ETX, which is calculated as \(1/(Q_f \times Q_r)\), where \(Q_f\) and \(Q_r\) are forward and backward link quality, respectively [3]. ETX-based routing protocols tend to avoid asymmetric links. However, many experiments [5–7] have shown that wireless links are complex and asymmetric. In general, the reverse link has less effect on data transmission, especially in some applications, a majority of data only need to be delivered from sensors to sink node. Also in the other real-time applications, retransmission data is almost useless. Therefore, path selecting based on bidirectional link quality often misses some good routes and discounts the potential resources.

In [5], Sang et al. proposed a reformative metric called ETF aiming at asymmetric network. They investigated the link asymmetric in WSN and demonstrated the improvement by exploiting ETF as the routing metric over ETX. ETF-based protocols choose the routes for data packet delivery solely according to its forward link quality toward the destination. Like other reliable protocols, ETF protocol also exploits retransmission mechanism to compensate for the packet loss.
However, in such protocols, the acknowledgement packets (ACKs or NACKs) that are transmitted in the reverse links usually are more important than data packets delivered in forwarding link. Even the synchronous ACKs exploited in [5] are demonstrated to obtain higher reliability; their loss will still be not tolerant for the reliable data delivery.

In this paper, we presented a protocol that uses erasure-resilient codes in application layer to guarantee the reliability of data delivery. Therefore, the reliable transmission is transferred into the best-effort data delivery according to the feature of erasure-resilient codes. Consequently, it can simplify the function for reliability requirement in data link layer and network layer. Our protocol can find high fidelity route by broadcasting asynchronous discovering beacons and form a single-direction collection tree, which avoids missing some good routes and reduces the effect of asymmetric link in the network. Meanwhile, in forwarder node, large buffer required by potential retransmission request also can be avoided. Additionally, in our protocol, some sophisticated strategies like Trickle algorithm [8] and pipeline mechanism are also exploited to reduce the control information and improve the efficiency of data delivery. This protocol is referred to as RECPE in this paper, which can benefit some applications significantly like real-time video delivery where retransmitting packets are useless. RECPE strives to provide the reliability for some applications in which bulk data needed to be delivered to the sink. The sender and sink have powerful processors, and the relaying nodes are some traditional resource-constrained nodes. Image collection and monitoring in the wild zone is a good example of application. We evaluate the performance of RECPE via TOSSIM, in comparison with CTP [9] protocol, RECPE significantly reduces the delivery cost, latency, and packet loss rate for reliable data collection especially in asymmetric link networks.

The remainder of the paper is organized as follows. In Section 2, we briefly review some data delivery approach in asymmetric network and application of erasure-resilient codes in WSN. In Section 3, we present the protocol details. Then, we provide some simulation results that show the protocol performance in Section 4. Finally, we make a conclusion in Section 5.

2. Related Work

In this section, we review briefly two related works: the data delivery in asymmetric network and the application of erasure-resilient codes in WSN.

Some earlier works indicated that link asymmetry is a real issue in low-power WSN. Zhou et al. [10] reported that about 30% links were asymmetric in their deployed system. Zamalloa and Krishnamachari [11] provided a comprehensive analysis of the root causes of link unreliability and asymmetry. Kannan et al. [12] found that most intermediate links are bursty and they shift between poor and good delivery. Based on these facts, some efficient mechanisms were proposed to tackle the asymmetry of link. ETF [5] has been proposed as a routing metric and shown to perform well in a variety of asymmetric wireless networks. Diversely, DEAL [13] proposed an approach to discover and manage the asymmetric link, which can reduce the expected packet count when they are exploited in the data collection application. Not just focusing on the routing metric, EERDC [14] proposed an energy-efficient reliable data collection protocol which uses the implicit ACK to cope with the asymmetry of network. Thus, our newly proposed RECPE protocol focuses on the reliable data delivery that is achieved by essentially combining the ETF metric and the packet-level erasure-resilient codes to avoid the retransmission. Unlike the DEAL and EERDC that need the acknowledgment packet and only provide the statistical reliability, RECPE is a package solution that does not require the reverse link and can guarantee the data delivery with absolute reliability. Additionally, in these proposed reliable schemes, the sender node and relaying nodes along the path must keep all transmitting packets in the buffer until they are acknowledged. Therefore, large buffer is required by each relay node, especially those that are close to the sink due to the need to relay packets for plenty of senders. Thus, these schemes would make buffer overflow at the relay node and sender on the large-scale WSN. Moreover, the use of a retransmission also introduces the duplicates and interference, even such as ETF scheme in which the dynamic retransmission threshold strategy is exploited. In contrast with these proposed protocols, RECPE can improve the throughput and reduce the delivery cost greatly.

The packet-level erasure-resilient codes first are presented as a mechanism for reliable and efficient multicast for file distribution system [15]. Recently, researchers started to use them to provide reliability for data delivery in WSN. Rateless Deluge [16] enhances the Deluge [17] protocol through a hybrid ARQ technique based on erasure-resilient codes. AdapCode [18] performs the encoding at each node during data dissemination; also the coding aggressiveness is adaptively changed according to the link qualities and the number of neighbors. SYNAPSE++ [19] disseminates reliably the data by implementing the hybrid ARQ similar to that in [16] but adds full support for pipelining through a joint design of MAC and Fountain codes. In [20], Cataldi et al. also exploited the erasure-resilient codes to disseminate the data in wireless vehicular network. As mentioned, all these applications focus on the reliable data dissemination in WSN.

In this paper, we present the reliable data collection approach which exploits the erasure-resilient codes to provide the reliability and avoids the requirement in the feedback channel. Furthermore, RECPE improves the efficiency of data delivery by constructing one-way collection tree using ETF metric. Since it is designed as a whole solution, the details of algorithm and frame format also are included in this paper. Furthermore, by modularization, our protocol also can be embedded into any other collection protocols to substantially improve their performance. To characterize the impact of our protocol, we use CTP protocol as the benchmark in our simulation.

3. Protocol Design

In this section, we present the motivations and demonstrate an example to show the improvement of exploiting ETF as
the metric in asymmetric network and then focus on the protocol design issues including exploiting efficient routing beacon mechanism to form a one-way collection tree for reliable data delivery. Our work puts heavy weight on optimization for function realization in transport and network layers.

3.1. Motivation. This work is mainly motivated from some experiments. During data delivery, substantial wireless links are asymmetric in manipulating a real-world system. Additionally, in data collection scenario, only forward links are needed to deliver the data packet to the sink. In our proposed protocol, due to the reliability guaranteed by the erasure-resilient codes, the reverse links required by the traditional reliable protocols are trivial. Therefore, we can construct the optimal one-way data collection tree for data delivery by only exploiting forwarding link quality. Particularly, ETX and minimum hop-count metrics cannot choose a better route when the deployment scenario experiences a great deal of asymmetry. An example in Figure 1 can demonstrate this situation. While node A delivers a packet to node E, Table 1 lists possible routes and their determination under different metrics. Obviously, it only takes 4.2 transmissions on average to deliver a data packet through route A → C → D → E chosen by metric ETF, while more than 3 times packets need to be delivered when other two metrics are used for routing.

3.2. Design Overview. RECPE is expected to aggregate data packets from all the nodes in the network to one sink efficiently. To this end, RECPE builds and maintains a collection tree that spans the entire network, in which each node has a one-way route to the sink. Meanwhile, the node attempts to deliver the packet to the best forwarder, which means that each node only needs to maintain a local routing table. RECPE finds a path with the minimum ETF obtained by adding all the ETF values of the links in the path. Some periodic beacons are exchanged between neighbors to estimate the ETF to neighbors and the whole ETF to sink. Consecutive sequence numbers are added to these beacons, so that neighbors can determine which beacons are lost during transmission. The fraction of lost beacons forms an estimation of the link quality from the neighbors. To reduce the control information, routing beacons are transmitted according to a variable timer, controlled by Trickle algorithm. Trickle will decrease the beacon frequency exponentially when the topology is stable. Once the inconsistency of topology is detected, the beacon frequency will be reset to the minimum value. This approach ensures the agile response to topology change while the control traffic overhead is minimized. Once the nodes have link estimation, they can build the routing tree. The routing beacons contain a node’s ETF to the sink. Upon receiving the beacon, a node can choose a parent that yields the smallest routing cost, which is the sum of the parent’s ETF and the ETF to its parent. Then, RECPE forwards the packet along the one-way collection tree. The reliability is guaranteed using erasure-resilient in the application layer, which can avoid the retransmission requirement of the data link layer. Therefore, the low layers of protocol only need to provide the best-effort one-way data delivery. Meanwhile, to prevent self-interference, RECPE uses a method like pipeline delivery to forward packets: after transmitting a packet the forwarder waits at least 2 packet transmitting time before delivering the next one.

3.3. One-Way Link Estimation and Construction of Collection Tree. Generally, there are two approaches to estimate link quality. One is hardware-assisted link quality estimation which uses signal strength provided by the radio as the estimation of link quality. This approach usually cannot give the accurate value for link selection and only can provide a qualitative indication about the link: low quality and high quality. Another approach broadcasts beacons periodically at the link layer, and the neighborhood nodes can estimate the link quality according to the sequence number of the beacons. Obviously, a high beacon rate can lead to a more agile response to the network dynamic, simultaneously introduce high control overhead. However, low beacon rate results in slow scalability for dynamic network, forming the loss of a lot of packets when the network changes frequently.

RECPE exploits beacons controlled by Trickle algorithm to get link estimation. Trickle obtains good tradeoff between agility and efficiency. Trickle broadcasts routing beacons using a variable timer which varies between 100 ms and 1 hour. When the timer expires, RECPE doubles it until the maximum value. Whenever the RECPE detects an event that indicates the topology changes, it resets the timer to the minimum value. If a node has a timer value \( \tau \), it will choose a random time within the interval of \([\tau/2, \tau]\) to broadcast beacon, which can prevent the collision among nodes whose timers synchronize. This strategy uses a few beacons to maintain collection tree without sacrificing agileness.

In RECPE, three events can reset the timer to the minimum value:

(i) a node receives a beacon packet with a “P” bit set.
(ii) A node is asked to forward a data packet whose routing cost is lower than its own.
(iii) The routing cost of a node decreases or increases significantly.

When a new node joins the network, it will broadcast a beacon with a "P" bit set, which enables the new node to join the network rapidly. The last two events are detected by the data frame and routing frame, respectively, which indicates that the topology changes significantly because of the node moving or dying. The network will evaluate the link qualities and create new routing table, which can avoid the loss of plenty of packets when route inconsistency emerges in the network.

Figure 2 shows the format of routing beacon, which contains several inbound link estimations and some control information. The beacon advertises the routing cost of the current node to the sink and the inbound link quality of its several closest neighbors. Then, the neighbor node that receives the beacon can know the routing cost if it chooses the broadcasting node as its parent. By comparison in the routing table, it can also choose the node with minimum ETF as its routing forwarder. At the same time, the neighbor node can calculate the inbound link estimation from the broadcasting node by the routing beacon. We calculate the reception probability by using a windowed moving average. The ETF estimation is \((\text{Seq}_{f_{n+k}} - \text{Seq}_i)/k\). When broadcasting the beacon, the node will choose several newest (updated newly) neighbors' ETF to insert into the routing frame. The number of link information entries can be different in each routing beacon, which is indicated in the field "Num_entry."

There are also some control bits in the routing frame. "Beacon_flag" differentiates the routing beacon to the data frame. The pull bit "P" is set when a new node joins networks, so that all the neighbors can broadcast routing beacon as soon as possible and the new node can get the inbound link estimation quickly. The bit "C" indicates the relay node is encountering congestion; thus, it will not be chosen as the forwarder by its neighbor. The beacon with "F" set only originated from the sink but can be forwarded by any other nodes, which means that a file with the sequence number "File_ID" is decoded successfully in the sink. Thus, the source node will stop to inject the encoding packets into network, and the forwarders also will give up delivering the data packets with the same file sequence number.

3.4. Data Delivery. In RECPE, the node transmits all received packets and encoding packets generated locally to the parent selected by the link estimation algorithm. Figure 3 shows the format of the data frame, whose size is eight bytes excluding data payload. One bit "Data_flag" can distinguish the data frame from the beacon frame. The "Origin_ID" in a data packet indicates the source node that generates the data packet. The field "File_ID" acts as an application dispatch identifier, which will dispatch the receiving packets to the corresponding decoder, and all packets produced by an application have their unique "File_ID" in the data frame. The "Packet_seq" can identify the unique packet generated by the same node. Particularly, the data frame also provides routing control field "Routing_cost" for dynamic route detecting, which can find link inconsistency quickly and reduce the control beacon by avoiding frequent routing frame exchanging.

The packets can be heard a few hops up and down the path. Therefore, to avoid self-interference, the node typically should wait at least two packets time when forwarding the data packets continuously. For example, considering the linear network shown in Figure 4, nodes A and D can simultaneously transmit the packets to their downstream neighbors without interference. The enforced delay depends on the packet rate of the radio and is empirically established by the following linear data delivery experiment.

A packet forwarding experiment shows the effect of a varying transmission wait time on a single node flow in the linear MICAz testbed. In the experiment, node A transmits some packets to node F without the end-to-end reliable control strategy; so the accumulative PLR (packet loss rate) can be defined as the ratio of the number of receiving packets in node F to the number of all generating packets in node A. Figure 5 shows the effect of interval time on packet loss.
The transmitting timers range from 2 ms to 18 ms. When the value is below 12 ms, PLR always decreases due to the self-interference between neighbor nodes. However, the PLR hardly changes when the transmitting interval time is more than 12 ms. If the interval time is too long, the network can be idle and the channel will be wasted. According to this experiment, the optimal waiting time is shown about 12 ms. It is reasonable that this optimal forwarding interval time is chosen based on the linear network since a lot of data packets often are generated in a burst by one node in WSN. This empirical interval time will be exploited in RECPE without considering the packet size.

3.5. Reliability Design. In RECPE, the reliability is guaranteed by encoding all packets in the application layer. Erasure-resilient codes can provide resilience to packet-level losses.

In the source node, the encoder produces a sequence of encoding packets from the set of input packets. For the erasure-resilient codes we use, each encoding packet is simply bitwise XOR of a specific subset of the input packets. In the sink, the decoder attempts to recover the original content from the encoding packets. Some simulation demonstrated that well-designed degree distribution can recover all original packets and only requires a few percent (less than 5%) of encoding packets beyond the number of original packets.

Provably good degree distributions for sparse parity check codes were first developed and analyzed in [21]. However, these codes are rate fixed, which means that only a predetermined number of encoding symbols are generated. Thus, in our application, it can lead to inefficiencies for decoding in the sink, as the source node will eventually be forced to transmit more encoding packets. Newer codes, called rateless codes, can avoid this limitation and allow unbounded numbers of encoding symbols to be generated on demand. Two examples of rateless codes, along with further discussion of the merits of ratelessness, may be found in [22, 23]. Both of these codes also have strong probabilistic decoding guarantees, along with low decoding overhead and average degrees. In our experiments, we use LT codes [22], and the degree distribution and importance sampling approach are described in [24]. The packets with the same file sequence number “File_ID” are forwarded to the same decoder of the application layer in the sink. The sink will broadcast a special beacon frame with “F” bit set when it succeeds to decode all original packets. This special noticing
4. Experimental Evaluation

4.1. Experimental Methodology. To compare RECPE with CTP in large network topology, we use the TOSSIM [25] simulation tool to evaluate the relative performance in a variety of network densities. Except specified especially in the following simulation, the codes for CTP and RECPE were implemented based on the default parameters of TinyOS2.x. We generated a network topology in terms of power level gain by applying a theoretical propagation model. This model is configured by parameters in three aspects: channel, radio, and topology, which are described carefully in [26]. Table 2 presents the radio parameters which can generate three kinds of networks for our simulations: symmetric, low asymmetric, and high asymmetric link networks. To improve the quality of radio simulation, CPM [27] algorithm is also used to simulate the RF noise and interference. We utilize the noise trace from the Meyer library, which generates a statistical model in CPM.

Figure 6 plots the network topology, in which 225 nodes are placed in grid with different space (the distance between neighbors). Since plenty of packets often are generated in a burst by one node in some application such as event monitoring, in our simulations only node 112 located at (8, 8) generates the data packets and transmits them to sink node (0, 0) after encoding by erasure-resilient codes.

![Figure 8: Packet loss rate of CTP and RECPE under different packet delivery traffic.](image)

Table 2: Radio parameters for simulation.

| Radio parameters | $S_{11}$ | $S_{12}$ | $S_{21}$ | $S_{22}$ | $N_F$ | $W_F-G_N$ |
|------------------|---------|---------|---------|---------|------|---------|
| High asymmetry   | 3.7     | -3.3    | -3.3    | 6.0     | -106.0 | 4       |
| Low asymmetry    | 0.9     | -0.7    | -0.7    | 1.2     | -105.0 | 4       |
| Symmetric link   | 0       | ×       | ×       | 0       | -105.0 | 4       |

beacon will be scattered rapidly in the entire network with the highest precedence.

Table 3: Packet loss number and network stable speed under symmetric, low and high asymmetric networks.

|                  | Symmetric | Low asymmetric | High asymmetric |
|------------------|-----------|----------------|-----------------|
| Packet loss num. | CTP 28    | 61             | 59              |
|                  | RECPE 20  | 58             | 41              |
| Network stable seq. | CTP 42 | 103            | 185             |
|                  | RECPE 27  | 94             | 176             |

4.2. Simulation Result. Firstly, we conducted a simulation to compare the average packet delivery time between CTP and RECPE when delivering 1000 packets from the sender to the sink. The packet traffic generated in the sender is 10 packets per second. The traveling time of each packet on the route is calculated from the sender to the sink, which is referred to as packet delivery time. Figure 7 shows the simulation result when the space is 2 meters.

Figure 7 showed that RECPE can obtain better latency when delivering data packet from the sender to the sink under symmetric, low and high asymmetric networks due to better route chosen and no retransmission. That indicates that RECPE also would spend less time to deliver a file under such networks.

At the same simulation, we counted the packet loss number and network stable speed. After delivering 1000 packet, RECPE only lost 20, 58, and 41 packets in symmetric, low and high asymmetric networks. However, CTP is 28, 61, and 59 packets, respectively, which are shown in Table 3. Simultaneously, we also observed the routing stable speed of protocol (when plenty of packets are generated in a burst in the resource node, CTP and RECPE usually spend time to form a stable route; before that, lots of packet are lost due to the variable route). CTP can find stable route under delivering 42, 103, and 185 packets, respectively, under three kinds of networks, while RECPE is faster, respectively, after 27, 94, and 176 packets delivery.

We also measured the effect of packet traffic on packet loss rate in the low asymmetric network when the space is 2 meters. In this simulation, 1000 packets are delivered from the sender to the sink. Figure 8 showed that RECPE always got lower packet loss rate than CTP, particularly, when the network had high traffic rate. The improved performance is contributed by less radio interference during packet transmission due to the pipeline data delivery and no retransmission requirement in RECPE.

We further investigated the efficiency of data delivery in terms of the delivery cost which accounts for all data and control overhead. Delivery cost is defined as the ratio of the total bytes transmitted or forwarded by all the nodes (including the source node) to the size of file delivered from the sender to the sink. Note that the overhead also includes the frame header and control information in each packet. Lower delivery cost means consuming less energy when delivering the same content in the network. In all simulation
scenarios, the file of 30k bytes packaged to 1000 packets is transmitted from the sender to the sink. In RECPE, these packets are encoded into enough packets by erasure-resilient codes in the sender and delivered into the network until the sink can recover all the original packets. Figure 9 plots the delivery cost of CTP and RECPE protocols in three kinds of networks.

As shown in Figure 9, in symmetric link network, RECPE obtained less delivery cost than CTP and kept the tendency under the other space simulations. In low and high asymmetric networks, the similar simulation results are obtained too. Another observation is that the delivery cost in RECPE almost increases linearly, while CTP fluctuates drastically. Totally, the performance of RECPE is hardly affected by link asymmetry of networks, which implies that RECPE will perform stably in almost all complex deployment scenarios.

5. Conclusion

In this paper, we present RECPE, a reliable bulk data collection protocol for large-scale multihop WSN. RECPE exploits beacons controlled by Trickle to obtain link estimation and constructs a one-way collection tree. The data packets are delivered through a pipeline approach. We also present the formats of beacon frame and data frame. RECPE differs from the state-of-the-art CTP protocol in its use of erasure-resilient codes to guarantee the delivery reliability, which avoids the requirement of retransmission in data link layer. RECPE can obtain significant performance in terms of energy efficiency and scalability in various density networks.

We evaluated the delivery cost of CTP and RECPE via TOSSIM simulation. Our results show that

(i) RECPE can get lower average packet delivery time than CTP under symmetric, low and high asymmetric networks. Also, RECPE obtain better packet loss rate than CTP under various packet delivery traffic.

(ii) The delivery costs of CTP and RECPE are comparable in symmetric and low asymmetric networks. However, RECPE performs much better in high asymmetric network.

(iii) RECPE also obtains significant stabilization in various asymmetric and different density networks.

Acknowledgment

This work was supported by Inha University, Republic of Korea.

References

[1] C. E. Perkins and P. Bhagwat, “Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers,” ACM SIGCOMM Computer Communication Review, vol. 24, pp. 234–244, 1994.
