Opportunistic Routing for Load-balancing and Reliable Data Dissemination in Wireless Sensor Networks

Min Chen, Wen Ji, Xiaofei Wang, Wei Cai and Lingxia Liao

November 12, 2009

Contents

Chapter 1

Opportunistic Routing for load-balancing and reliable data dissemination in wireless sensor networks

Min Chen[1,2], Wen Ji[3], Xiaofei Wang[4], Wei Cai[4] and Lingxia Liao[1]
[1]University of British Columbia
[2]Hebei Polytechnic University
[3]Chinese Academy of Sciences
[4]Seoul National University

Advances in microelectronics and communications enable inexpensive sensors to be deployed on a large scale and in harsh environments, where sensors need to operate unattended in an autonomous manner. As sensor nodes communicate over error-prone wireless channels with battery power, reliable and energy-efficient data delivery is crucial. These characteristics of wireless sensor networks (WSNs) make the design of routing protocols challenging [1].

Many studies are done in WSNs such as energy-efficiency, load balancing, and reliability. However, these design goals are generally orthogonal to each other. For example, most of the load balancing schemes are not robust to high link failure rate. In this chapter, the existing load balancing schemes are classified into two categories: *local load-balancing* [2, 3, 4] and *global load-balancing* [5]. These are also referred as hop-by-hop balancing and end-to-end balancing, respectively. To evaluate the performance of load balancing, we define the "lifetime" of a WSN as the time until the first node in the WSN drains its battery power and dies.

Local load-balancing is based on a "node-centric" approach, where a hello message is broadcast by each sensor node periodically during the network operation, in order to notify its neighbors of its energy changes. The interval of broadcasting such a hello message provides a tradeoff between control overhead and timeliness of local energy information. Local load-balancing may cause a data packet to enter an "energy bottleneck" region, where the energy levels of the sensor nodes are relatively low while the sensors outside the region may still have higher remaining energy levels. Thus, some load balancing schemes aim to find globally load balanced paths to achieve a higher network lifetime.

In [2, 5], to reduce packet losses due to frequent link failures, a forwarding node also uses alternative (backup) nodes by setting up multiple backup next hop nodes in advance. If the primary next hop node fails, the medium access control (MAC) layer is not able to deliver a packet to this unreachable primary node. After several retransmission attempts, the MAC layer simply drops the packet and notifies the network layer of the transmission failure. The routing protocol then selects a backup next hop and hands the same packet (stored in the cache) down to the MAC layer. If the backup next hop also dies, these retransmissions are repeated. When the node failure rate is high, trying multiple backup nodes along with data caching severely increases the delay, reduces the effective available bandwidth, and wastes energy for unnecessary transmissions. Nevertheless, the above operation is widely used in traditional routing protocols for ad hoc and sensor networks [2, 5], which operate in the following two-step manner: 1) select the next hop node first based on a neighbor information table (denoted by NIT), as shown in Fig. 1.2; 2) forward packet to the selected node until a predetermined number of transmissions fail. We call this a “transmitter-oriented” approach.

In this chapter, a novel opportunistic routing protocol is proposed, which facilitates load-balancing and reliable data dissemination in wireless sensor networks. Compared to traditional transmitter-oriented approach, “receiver-oriented” is exploited to achieve both load balancing and reliability for large-scale WSNs. Thus, the proposed opportunistic routing scheme is called load-balancing reliable routing protocol (RLRR) [7]. In the receiver-oriented approach, a next-hop solicitation message is broadcast by a forwarding node, and all the neighbors receive the message. First, the hop count of a neighbor candidate to the sink should be less than that of the forwarding node. Then each neighbor’s eligibility as a next hop is decided by its remaining energy level, which is in turn reflected into the *temporal gradient*. The next hop candidate with the least temporal gradient (or highest remaining energy) will reply with a next-hop response message with the shortest backoff time. Without central coordination, the candidate with the least *temporal gradient* is selected to deliver data packets toward the sink and suppress the other candidates.

The receiver-oriented approach employed by RLRR has the following advantages: (1) The protocol is stateless. No hello message beaconing is needed to update neighbors’ energy information periodically; (2) The energy information is used locally in each next hop candidate in load-balanced routing decision, and hence is always accurate and up-to-date.

If there is no next hop candidates whose hop count to the sink is smaller, RLRR utilizes peer neighbors whose hop count is equal to that of the forwarding node can be exploited for reliability and load balancing while guaranteeing loop-free routing. Our receiver-oriented idea is close to GeRaF [9] and ExOR [10] where efficient methods of using multi-receiver diversity for packet forwarding are explored. However, unlike GeRaF and ExOR, RLRR does not rely on geographical information provided by expensive GPS devices.

We carry out extensive simulations to show that RLRR mostly achieves higher reliability than EDDD [5], DD [8] and GEAR [2]. More importantly, RLRR also exhibits longer network lifetime. The overall performance gain of RLRR, taking into account of reliability, lifetime, and data delivery latency, increases as the link failure rate increases.

The rest of this chapter is organized as follows. We describe RLRR design issues and its algorithm in Sections 1.2 and 1.3, respectively. Simulation model and experimental results are presented in Section 1.4. Finally, Section 1.5 concludes the chapter.

1.1 Related Work

In addition to the background presented in the previous section, our work is also related to cooperative communications, and the reliable data transfer scheme in WSNs. We will give a brief review of the work in these two aspects.

A large number of cooperative communication protocols have been proposed recently. Cooperation diversity gains, transmitting, receiving and processing overheads, are investigated by [11]. Cooperative issues across the different layers of the communication protocol stack, self-interested behaviors and possible misbehaviors are explored in [12]. [13] proposed a cooperative relay framework which accommodates the physical, medium access control (MAC) and network layers for wireless ad-hoc networks. In the network layer, diversity gains can be achieved by selecting two cooperative relays based on the average link signal-to-noise ratio (SNR) and the two-hop neighborhood information. A cooperative communication scheme combining relay selection with power control is proposed in [14], where the potential relays compute individually the required transmission power to participate in the cooperative communications. A variety of cooperative diversity protocols are proposed by [15], namely, amplify-and-forward, decode-and-forward, selection relaying, and incremental relaying. The performance of the protocols in terms of outage events and associated outage probabilities are evaluated respectively. Coded cooperation [16] integrated cooperation with channel coding and works by sending different parts of each user's code word via two independent fading paths. [17, 18] implemented a cooperation strategy for mobile users in a conventional code division multiple access (CDMA) systems, in which users are active and use different spreading code to avoid interferences. In [19], distributed cooperative protocols, including random selection, received SNR selection and fixed priority selection, are proposed for cooperative partner selection. The outage probability of the protocols are analyzed respectively. CoopMAC, a cooperative MAC protocol for IEEE 802.11 wireless networks, is presented by [20]. CoopMAC can achieve performance improvements by exploiting both the broadcast nature of the wireless channel and cooperative diversity.

There are increasing research efforts on studying the issue of reliable data transfer in WSNs [21, 22, 26, 23, 24, 25]. In these studies, hop-by-hop recovery [21, 22], end-to-end recovery [26], and multi-path forwarding [23, 24, 25] are the major approaches to achieve the desired reliability. PSFQ [21] works by distributing data from source nodes in a relatively slow pace and allowing nodes experiencing data losses to recover any missing segments from immediate neighbors aggressively. PSFQ employs hop-by-hop recovery instead of end-to-end recovery. In [22], the authors proposed RMST, a transport protocol that provides guaranteed delivery for application requirements. RMST is a selective NACK-based protocol that can be configured for in-network caching and repair. In [23], multiple disjoint paths are set up first, then multiple data copies are delivered using these paths. In [24], a protocol called ReInForM is proposed to deliver packets at a desired level of reliability by sending multiple copies of each packet along multiple paths from sources to sink. The number of data copies (or, the number of paths used) is dynamically determined depending on the probability of channel error. Instead of using disjoint paths, GRAB [25] uses a path interleaving technique to achieve high reliability. It assigns the amount of credit α to each packet at the source. α determines the "width" of the forwarding mesh and should be large enough to ensure robustness but not to cause excessive energy consumption. It is worth noting that although GRAB [25] also exploits data broadcasting to attain high reliability, it may not be energy-efficient because it may involve many next-hop nodes in order to achieve good reliability and an unnecessarily large number of packets may be broadcast. Considering the asymmetric many-to-one communication pattern from sources to sink in some sensor applications, data packets collected for a single event exhibit high redundancy. Thus, some reliable techniques [21, 22] proposed for WSN would either be unnecessary or spend too much resources on guaranteeing 100% reliable delivery of data packets. Exploiting the fact that the redundancy in sensed data collected by closely deployed sensor nodes can mitigate channel errors and node failures, ESRT [26] intends to minimize the total energy consumption while guaranteeing the end-to-sink reliability. In ESRT, the sink adaptively achieves the expected event reliability by controlling the reporting frequency of the source nodes. However, in the case that many sources are involved in reporting data simultaneously to ensure some reliability (e.g., in a highly unreliable environment), the large amount of communications are likely to cause congestion.

1.2 RLRR Design Issues

1.2.1 Accurate and Up-to-date Energy Information

In local load-balancing protocols, beaconing is required periodically for setting up energy information tables. During the interval between two beacons, the energy information stored in the table does not reflect the actual energy information, since sensor nodes likely consume energy continuously over time. Thus, the interval of broadcasting such a hello message provides a tradeoff between control overhead and timeliness of local energy information. In contrast, with the receiver-oriented approach in RLRR, a neighbor node uses its own energy information, which is always accurate and up-to-date, to evaluate its eligibility to be selected as a next hop node.

1.2.2 Load Balancing

We assume that every node starts with the same energy level corresponding to full battery capacity. In RLRR, the current energy levels (remaining battery capacities) of the sensor nodes are discretized into integer-valued quantized-energy-levels (QELs). Given the example shown in Fig. 1.1, assuming the full energy level (E_{max}) of a battery is equal to 10000, and the "unit energy" (the unit of the quantization, E_{unit}) is equal to 2000. Then, the maximum value of QEL is ($QEL_{max} = \lceil \frac{E_{max}}{E_{unit}} \rceil = 5$. In this chapter, we do not differentiate the energy levels of sensor nodes with the same QEL. For example, both energy levels 6500 and 6750 have the same QEL of 4. With effective load balancing in a WSN, the sensor nodes close to one another (e.g., within one hop distance) will have similar QELs after an extended period of network operation, because neighbors with higher QELs will be selected to forward data until their QELs are decreased to levels no higher than those of other neighbors. The larger the range of QELs, i.e., the smaller the unit energy used in the quantization of the energy levels, the better the load balancing performance should be. In this case, a longer expected lifetime is likely to be achieved, but at the expense of a higher control overhead to carry out more frequent route oscillations. Thus, QEL_{max} should be optimized to achieve the best tradeoff between load balancing and control overhead.

1.2.3 Reliability

With the receiver-oriented approach, the property of broadcasting is exploited to attain high reliability. In RLRR, the source node and any intermediate sensor node broadcast a route selection message. Neighbors that receive the route selection message successfully have the responsibility of choosing the next hop among themselves. In the case that no such available neighbor is found, the node will mark itself a deadend node and inform the upstream node to discover a new route that bypasses the dead end. Especially, RLRR exploits peer neighbors whose hop count is equal to that of the upstream node to increase reliability. However, RLRR faces the challenge of maintaining loop freedom when peer neighbors are exploited.

1.2.4 Loop Freedom

In order to guarantee loop freedom, many routing schemes based on neighbor information only adopt the set of “minimum hop count” nodes as backup next hop nodes to counteract frequent route failures. A “minimum hop count” node has a hop count to the sink that is 1 less than the hop count of the current node. Therefore these schemes exclude the neighbors whose hop counts are the same as that of the current node (i.e., peer neighbors) as potential next hop nodes. In RLRR, the number of backup nodes is exploited to a maximum extent possible by also involving peer neighbors to route data packets in order to achieve better load balancing and reliability. With the receiver-oriented approach, loop freedom is guaranteed with no additional control overhead, as will be explained in detail in Section 1.3.4.

1.2.5 Low-Cost Sensor Design

Traditional sensor routing protocols usually require a sensor node to maintain the information of multiple neighbors (e.g., backup routes and energy levels). In very large scale and dense WSNs, the amount of such information may pose an additional challenge for the sensor nodes with low storage capacity. However, with RLRR, sensor nodes do not need to store any additional routing and energy-related information except for the identifier of its next hop node and its upstream node for each flow. Though stateless geographical routing schemes also do not need to set up route tables, they need to obtain geographical information using GPS devices. By comparison, RLRR does not need any geographical information to achieve stateless routing.

1.3 The RLRR Protocol

1.3.1 The RLRR Mechanism

In RLRR, each node has a “flow-entry” which indicates the identifier of its next hop node for forwarding data to the sink. Initially, a sink floods interest packets to the network. Each sensor sets up its hop count gradient to the sink. Sensor(s) matching the interest will become the source node(s) [8]. Unlike minimum hop count-based routing schemes, the flow-entry is not set up during interest flooding in RLRR, since load balancing cannot be attained simply by considering hop count. Instead, the flow-entries of all the sensor nodes are still empty after interest flooding.

We denote a forwarding node (the source or an intermediate node) by “h”. The arrival of a sensory data packet (from the application layer of the source node or from the upstream node) triggers h to check its flow-entry. Since the flow-entry does not exist initially, h stores the data, starts a “route selection” process immediately to set up the flow-entry, and then transmits the stored data to the selected next hop node. As illustrated in Fig. 1.3(a), suppose node i with QEL of 5 is selected as the next hop node of node h. After the flow-entry was set up, data packets will be unicast directly to the next hop node recorded in the flow-entry.

As time goes on, node i will consume its energy faster than its neighbors. To achieve load balancing, node i should keep track of its own QEL in order to prevent excessive energy consumption for packet forwarding. When its QEL is decreased by 1, node i asks its upstream node h to re-select a new next hop node. In the example shown in Fig. 1.3(b), when QEL of node i changes from 5 to 4, it unicasts a next-hop-reselection message (RESEL) to its upstream node h. Upon receiving RESEL, h deletes its current flow entry and initiates route reselection. Assuming node j is selected due to its higher energy level, node i will be replaced by node j as the new next hop node of h.

In addition to balancing the energy consumption, route reselection is also triggered to recover a link failure. In the example shown in Fig. 1.3(c), node h fails to deliver a data packet to node i according to the existing flow-entry, and receives feedback information from its MAC layer that indicates a transmission failure. Then node h deletes its current flow entry and initiates route reselection. Assuming the wireless link to node j is in a good condition and other factors (such as remaining energy and hop count) are favourable, hence this node is selected as the next hop and recorded in the flow-entry. The original next hop node i is now replaced by node j.

In addition, route selection/reselection (denoted by Sel/Resel, respectively) itself may fail. For example, if all the eligible neighbors (whose hop count to the sink is less than or equal to that of h) of node h have either depleted their energies or failed, node h becomes a dead end node. In this case, node h transmits a RESEL message to its upstream node (e.g., node g in Fig. 1.3(d)), which triggers a new route Reselection by node g, and so forth. The flowchart of the basic RLRR Protocol is shown in Fig. 1.4.

Fig. 1.5 shows the mechanism of receiver-oriented route Sel/Resel in RLRR. In order to deliver a data packet to a next hop node in dynamic network environments, node A broadcasts a probe message at first. The neighbor nodes (i.e., nodes B, C and D), which receive this message and are closer to the sink than node A, will start their backoff timers, as shown in Fig. 1.5(c). They are also called "live candidates" (LCs), such that the links between the transmitter and the candidates are in good status. Since multiple LCs usually starts their backoff timers (denoted by TG-Timers) simultaneously, the one (i.e., node C, as shown in Fig. 1.5(d)) with the least TG (*Time Gradient*) will expire first and becomes a "reserved next hop" (RNH), which means it is highly likely to be selected as the next hop node later. Note that the TG of individual LC indicates its eligibility level to be selected as the next hop.

Ideally, all the LCs except the RNH should cancel their TG-Timers and delete the packet from their forwarding buffers when the RNH's TG-Timer expires. To achieve this, RLRR operates as follows:

1. RNH broadcasts a "reply" message (REP) to node h;
2. If node h receives the REP from RNH, it will broadcast a "selection" message (SEL) with the identifier of the RNH, and start the selection-retransmission-timer (SEL-ReTx-Timer). To guarantee that only one LC be selected as the next hop node, node h only accepts the first REP sent by the RNH while ignoring the later ones. Note that the LCs overhearing the REP will back out (i.e., cancel their TG-Timers the drop the data from their caches) instantly;
3. If the RNH receives the SEL, it becomes the next hop node and relays the data by broadcasting. When other LCs receive the SEL, they will cancel their TG-Timers and drop the data sent by their forwarding caches;
4. If node h receives the broadcast data from its next hop node (the above RNH), it will cancel its SEL-ReTx-Timer. Otherwise, it will re-broadcast the SEL when the SEL-ReTx-Timer expires, and will start the timer again until the retry limit reaches.
5. If the RNH receives re-transmitted SEL, it will "unicast" an "selection-reply" message (SEL_REP) to node h;
6. If node h receives SEL_REP, it will cancel its SEL-ReTx-Timer.

Note that in step (1) two (or more) LCs with similar TGs broadcast their REPs simultaneously. If collision happens, both of the LCs will not be selected, and other LCs broadcast REPs later when their TG-Timers expire will be selected.

Furthermore, in the above step (2) it is possible that the SEL may collide with a new REP from other LC, which would cause the following two disadvantages: (a) RNH may fail to receive the SEL; (b) other LCs (non-RNH nodes) do not delete the data from their caches in early time. The case (b) only increases data caching time and control overhead, while the case (a) will cause the failure of the current data delivery if left without any measure. To ensure that the RNH receives the SEL at least once, node h should send the SEL again when SEL-ReTx-Timer expires.

1.3.2 Time Gradient Calculation

In Fig. 1.6, the LCs of node h are divided into two groups: (1) less-hop-count group (*L-Group*), consisting of LCs which are 1-hop closer to the sink than node h; and (2) equal-hop-count group (*E-Group*), consisting of the LCs having the same hop count as node h. Obviously, the LCs in *L-Group* should have higher priority than those in *E-Group*. In Fig. 1.6, the *L-Group* includes nodes $LC1$, $LC2$, $LC3$, $LC4$, $LC5$; and the *E-Group* includes nodes $LC6$, $LC7$, $LC8$, $LC9$, $LC10$.

Recall that node h broadcasts a PROB to initiate route Sel/Resel. The PROB contains the QEL and hop count of node h. In Fig. 1.6, QEL_{max} is equal to 10, and the QEL of each LC is indicated by the number in the respective circle. Upon receiving the PROB, an LC first decides which group it belongs. Then, it calculates the gap between its own QEL and the upstream node h's QEL, which is denoted by ΔE. Since time gradient (i.e., TG) determines the delay of sending a REP back to h, its value has a large impact on the data latency. In order to make TG as small as possible while achieving sufficient differentiation among all the LCs, we should avoid using large TG values to differentiate the LCs. Thus, we adopt ΔE instead of QEL to differentiate the LCs in the same group, since ΔE can be much smaller than QEL in a load balanced WSN.

Let TG_i denote the TG of node i. TG_i is calculated by Eqn.(1.1), where x is a parameter reflects both ΔEs and the type of group which an LC belongs to.

$$\begin{aligned}
\Delta E_i &= \begin{cases} QEL_h - QEL_i, & if\ QEL_h = QEL_{max} \\ QEL_h - QEL_i + 1, & if\ QEL_h > QEL_i - 1 \\ 0, & if\ QEL_h \leq QEL_i - 1 \end{cases} \\
x_i &= \begin{cases} \Delta E_i, & \text{i} \in \textit{L-Group} \\ \Delta E_i + \alpha & \text{i} \in \textit{E-Group} \end{cases} \\
TG_i &= f(x_i) = x_i \times \Delta TG + rand(\Delta TG)
\end{aligned} \tag{1.1}$$

In Eqn.(1.1), α is a positive constant used to differentiate between LCs in different groups by favoring the *L-Group* over the *E-Group*, ΔTG is a constant, and $rand(\Delta TG)$ is a random value between 0 and ΔTG

used to differentiate the LCs that have the same x. In other words, it is used to differentiate between multiple LCs that have the same QEL and belong to the same group (either *L-Group* or *E-Group*).

ΔTG should be set as small as possible to decrease the Sel/Resel delay, but if it is set too low, collisions of REP messages will occur frequently, because many LCs will likely try to send REPs within the small time period of ΔTG. Thus, ΔTG should be set according to the node density. Let N be the total number of sensor nodes in a WSN that has an area A. The node density of the WSN is equal to: $\delta = \frac{N}{A}$. Let r be the transmission range of a sensor node. Roughly, LCs are located within approximately one third of the whole transmission range in Fig. 1.6. Then, the number of LCs can be approximated by:

$$L = \frac{1}{3} \cdot \pi \cdot r^2 \cdot \delta \tag{1.2}$$

Among the LCs in the same group, on the average, half of them will have the same QEL in a *load balanced* WSN. Let *S-Group* denote the set of the LCs with the same QEL in the same group (i.e. *L-Group* or *E-Group*). Our goal is to make a contention time long enough to differentiate the LCs in the same *S-Group*. Let T_{REP} be the average time to successfully deliver a REP message. In order to minimize collisions with other *LC*s in the same *S-Group*, at least T_{REP} should be reserved for each LC. Thus, ΔTG is approximately equal to:

$$\Delta TG = \frac{L}{2} \cdot T_{REP} \tag{1.3}$$

Here $\frac{L}{2}$ is the average number of *LC*s in an *S-Group*. Let α be 2. In the example shown in Fig. 1.6, we can get four *S-Groups*: $LC2$, $LC3$, and $LC5$ with the $x = 0$; $LC1$ and $LC4$ with $x = 1$; $LC6$, $LC7$, and $LC9$ with $x = 2$; $LC8$ and $LC10$ with $x = 3$. The *TG*s of the *LC*s in each *S-Group* are randomly distributed over a range of ΔTG. In the example in Fig. 1.6, the increasing order of the *TG*s of all the *LC*s is: $TG3$, $TG2$, $TG5$, $TG1$, $TG4$, $TG6$, $TG9$, $TG7$, $TG10$, $TG8$. It corresponds to the decreasing order of LC's eligibility level as h's next hop node: $LC3$, $LC2$, $LC5$, $LC1$, $LC4$, $LC6$, $LC9$, $LC7$, $LC10$, $LC8$. As time goes on, the *TG-Timer* of $LC3$ will expire first, which causes $LC3$ to be selected as the next hop node.

1.3.3 Solving the Dead End Problem

The so-called dead end problem [28] arises when a packet is forwarded to a local optimum, i.e., a node with no neighbor of closer hop distance to the destination. The problem can be solved as follows: (1) If node h does not receive any REP until its *NoREP-Timer* expires, it will mark itself an unavailable node and unicast a RESEL to its upstream node u. An unavailable node will not participate in route Sel/Resel until the sink floods a new control message. The frequency of the sink flooding a control message should be traded off between control overhead and the timeliness of mitigating the deadend problem; (2) On receiving the RESEL, node u initiates route reselection and finds a new next hop to replace node h.

1.3.4 Loop Freedom in RLRR

Exploiting multiple backup nodes or multipath for data delivery can increase reliability. In general, the nodes in equal-hop-count group (*E-Group*) are not used as backup nodes to guarantee loop freedom in conventional routing schemes in ad hoc and sensor networks. By comparison, in RLRR, the *LC*s in *E-Group* are exploited to achieve better performance in terms of both load balancing and reliability. While using the *LC*s in *E-Group*, it is critical to ensure that an *LC* in the *E-Group* not be selected as a next hop again by another *LC* in the same *E-Group*. The receiver-oriented approach of RLRR makes this goal easily achieved with no additional control overhead, as illustrated in Fig. 1.7.

In Fig. 1.7(a), assume that node a is the only minimum-hop-count neighbor of node h, and it fails. Then, node h initiates route Resel by broadcasting a PROB message and starting a *Drop-PROB-Timer*. The route Resel results in node h selecting its peer neighbor node b (in *E-Group*) as the next hop node. Assuming that the flow-entry of node b does not exist, in Fig. 1.7(b), node b initiates route selection. Note that both h and c are peer neighbors of node b. When node b broadcasts a PROB and node h replies first, node h is selected as b's next hop node and a loop is formed.

To prevent node h from being selected as a next hop node by its peer neighbors, it ignores the PROBs and never participates in the selection until its *Drop-PROB-Timer* expires. When the *Drop-PROB-Timer* of node h expires, it can participate in the next hop selection process again. In general, the time value of *Drop-PROB-Timer* ($T_{Drop-PROB-Timer}$) should be long enough, such as:

$$T_{Drop-PROB-Timer} = N_{E-Group} \cdot \ t_{sel}. \tag{1.4}$$

In Eqn.(1.4), $N_{E-Group}$ denotes the maximum number of *LC*s in *E-Group*. t_{sel} denotes the time for one-hop route selection. Considering the worst case where all the *LC*s in the *E-Group* have no minimum-hop-count neighbors, each of them will initiate route selection once and finds a peer neighbor in the same *E-Group* as its next hop node. Then the accumulated time for all of the route selection attemps is equal to Eqn.(1.4), which guarantees that a node (e.g., node h) initiating route Sel/Resel will never be selected as a next hop node of any other *LC*s in the same *E-Group*. Thus, loop freedom is guaranteed.

1.4 Performance Metrics

In order to demonstrate the performance of RLRR, we compare it with several representative existing routing protocols for WSNs by extensive simulation studies.

We choose a global load-balancing scheme (i.e., EDDD [5]), a local load-balancing scheme (i.e. GEAR [2]), and a non-load-balancing scheme (i.e., DD [8]) to compare with RLRR. We implement the HGR protocol and perform simulations using OPNET Modeler [32, 33]. The sensor nodes are battery-operated except for the sink, which is assumed to have an infinite energy supply. The network with 800 nodes is uniformly deployed over a 500m × 500m field. As in [30], we let one sink stay at a corner of the field and one source node be located at the diagonal corner. Each source node generates sensed data packets at a constant bit rate with a 5 second interval between packets (1K Bytes each). As in [6], we use IEEE 802.11 DCF as the underlying MAC, and the radio transmission range (R) is set to 45m. The data rate of the wireless channel is 2 Mbps. All messages are 128 bytes in length. We assume both the sink and sensor nodes are stationary. In DD, EDDD, and RLRR, the sink will initiate interest flooding to carry out a new task. Interest packets are propagated hop-by-hop throughout the network. Among the target sensor nodes, while several nodes may match the interest, only one of these nodes will become a source node for each instance of interest flooding. We assume that a mechanism exists to elect one source node among several nodes that matches the interest, e.g., based on the remaining energy. In addition to the initial interest flooding, the sink also floods the interest packet periodically to update stale information in terms of hop count and energy. Since RLRR does not rely on the periodical flooding for local repair, the sink only floods interest once until network lifetime is reached. We employ the energy model used in [30, 31] and link failure model used in [29]. For each set of results, we simulate the WSN sixty times with the specific set of parameters and different random seeds.

In this section, five performance metrics are defined:

- *Number of Successful Data Deliveries during Lifetime* - It is the number of data packets delivered to the sink before network lifetime is reached. It is denoted by n_{data}, which is also used as an indication of the lifetime in this chapter.

- *Packet delivery ratio* - It is the ratio of the number of data packets delivered to the sink to the number of packets generated by the source nodes.
- *Average End-to-end Packet Delay* - It includes all possible delays during data dissemination, caused by queuing, retransmissions due to collisions at the MAC layer, and transmission time.
- *Energy Consumption per Successful Data Delivery* - It is denoted by e. It is the ratio of network energy consumption to the number of data packets delivered to the sink during the network lifetime. The network energy consumption includes all the energy consumption due to transmitting and receiving during the simulation. As in [29, 30, 6, 31], we do not account for energy consumption during the idle state, since this element is approximately the same for all the schemes considered.
- *Number of Control Messages per Successful Data Delivery* - It is denoted by n_{ctrl} and is the ratio of the number of control messages transmitted to the number of data packets delivered to the sink during the network lifetime.

We use n_{data} as an approximate indication of the network lifetime. If the packet delivery ratio is 100%, then n_{data} is exactly proportional to the network lifetime, due to the CBR traffic model used in the simulations. We believe that n_{data} is the most important metric for WSNs.

1.4.1 Effects of ΔTG

In these experiments, we change ΔTG from 2ms to 20ms by the step size of 2ms. In Fig. 1.8(a), n_{data} increases as ΔTG is increased, since the larger is ΔTG, the less collisions will happen, and the more data packets will be delivered successfully to the sink.

In Fig. 1.8(b), when ΔTG is small, end-to-end data delay of RLRR is high. The smaller is ΔTG, the more likely will the REPs transmitted by LCs with similar TGs collide, which causes LCs with lower TGs not to win the opportunity to become a next-hop node. With increasing ΔTG, the delay decreases, and reaches its minimum value when ΔTG is equal to 8 milliseconds. It is unnecessary to increase ΔTG more if the value is large enough to differentiate the *LC*s in the same *S-Group*, since a large ΔTG also increases the time for route Sel/Resel. Thus, when ΔTG goes beyond 8ms, the delay begins to increase again.

In Fig. 1.8(c) and Fig. 1.8(d), both e and n_{ctrl} decreases with ΔTG increasing. The larger is ΔTG, the less likely collision happens. Thus, the control overhead decreases.

1.4.2 Comparison of RLRR, EDDD, DD and GEAR with Variable Link Failure Rates

In this section, we change the link failure rate from 0 to 0.5 by the step size of 0.05. Fig. 1.9(a) shows that the packet delivery ratios of EDDD and DD are more sensitive to link failures than those of GEAR and RLRR, and EDDD has the lowest reliability because the load balanced path is not robust to link failures, since the failure of any link along the path will cause data delivery failure. RLRR yields higher reliability than GEAR because it exploits *E-Group* for alternating routing. In most cases, the numbers of nodes in the *L-Group* and *E-Group* are larger than the number of backup nodes in GEAR. Thus, RLRR keeps achieving more than 90% packet delivery ratio until the link failure rate is larger than 0.35.

In Fig. 1.9(b), when the link failure rate is 0, n_{data} of EDDD is larger than that of RLRR and GEAR, which illustrates the advantage of global load-balancing (EDDD) over local load-balancing (RLRR, GEAR) in reliable environments. Note that n_{data} is closely related to the network lifetime. Since DD has no load balancing mechanism, its lifetime is the lowest. With increasing link failure rate, RLRR exhibits consistently higher reliability and n_{data} than the other schemes, which shows that the proposed receiver-oriented scheme can achieve load-balancing with a lower control overhead and handle link failure better than conventional transmitter-oriented schemes.

According to the simulation results, we observe the following: 1) lifetime is greatly prolonged if a load balancing mechanism is adopted (e.g., EDDD, RLRR, and GEAR vs. DD); 2) in a reliable environment, a global load-balancing scheme exhibits a longer lifetime than local load-balancing schemes (e.g. EDDD vs. RLRR and GEAR); 3) RLRR exhibits more consistent and relatively higher reliability and longer lifetime than EDDD, DD, and GEAR in unreliable environments.

1.5 Conclusion

Routing protocols in wireless sensor networks (WSNs) typically employ a transmitter-oriented approach in which the next hop node is selected based on neighbor or network information. This approach incurs a large overhead when the accurate neighbor information is needed for efficient and reliable routing. Additionally, in unreliable communication environments, traditional routing protocols may fail to deliver data in a timely manner since global route discovery may be needed to handle link failures.

In this chapter, a novel opportunistic routing protocol (denoted by RLRR) is proposed for delivering data in a load-balancing and reliable fashion. In the proposed scheme, an intermediate node solicits next hop candidates, each of which is to respond with its own backoff time dubbed a temporal gradient (TG). In RLRR, the energy and hop count information of each "live candidate (LC)" is converted to a TG that is used to evaluate the eligibility of the node as a next-hop node. The set of LCs of a forwarding node h includes all neighbors of transmitter, whose hop counts to the sink are less than or equal to that of transmitter. To perform route selection, the transmitter broadcasts a probe message (PROB) that is received by its LCs. Each LC sets its "*TG-Timer*" to the calculated TG value and sends a "reply" message (REP) back to the transmitter when its "*TG-Timer*" expires. The LC that originated the first reply message received by the transmitter is selected as the next hop node. The best LC will have the least TG; therefore, its *TG-Timer* will expire first among all the LCs and it will be selected as the next hop node. In this way, the next hop is selected without any central coordination on a packet-by-packet basis. Thus, each node needs not maintain any neighbor information. The remaining energy level used to determine the TG is always accurate and up-to-date. The upstream node of a broken link broadcast a route request message received by all the live neighbors with a good link. By taking this "local" approach, route repair is fast and reliability is enhanced even in highly unreliable environments. Furthermore, neighbor nodes whose hop count is less than the soliciting node participate in the next-hop selection process with loop-free operation guarantee.

We have presented simulation results to show that the related parameters of the protocol need to be selected carefully to achieve load balancing with energy-efficiency while minimizing the control overhead. Simulations also show that the proposed protocol achieves relatively longer network lifetime and higher reliability than other existing schemes.

References

[1] J. Al-Karaki, and E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Personal Communications*, Vol.11 , No. 6 , pp.6-28, Dec. 2004

[2] Yu Y, Govindan R, Estrin D. Geographical and Energy Aware Routing. *UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023*, May 2001.

[3] H. Dai and R. Han, "A node-centric load balancing algorithm for WSNs," *In Proceedings of IEEE GLOBECOM*, vol. 1, pp.548–552, December 2003

[4] J. Gao, and L. Zhang, "Load balancing shortest path routing in wireless networks," *Proc. IEEE INFOCOM*, vol. 2, pp. 1098–1107, Mar. 2004.

[5] M. Chen, T. Kwon, and Y. Choi, "Energy-efficient differentiated directed diffusion for real-time traffic in wireless sensor networks," *Elsevier Computer Communications*, vol.29, no.2, pp.231–245, Jan. 2006.

[6] M. Chen, V. Leung, S. Mao, and Y. Yuan, "Directional geographical routing for real-time video communications in wireless sensor networks," *Elsevier Computer Communications*, vol.30, no.17, pp.3368–3383, Nov. 2007.

[7] M. Chen, V. Leung, S. Mao, and T. Kwon, "Receiver-oriented load-balancing and reliable routing in wireless sensor networks," *Wireless Communications and Mobile Computing Journal*, vol.9, no.3, pp.405–416, Mar. 2009.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. ACM MobiCom 2000*, pp. 56–67, Boston, MA, August 2000.

[9] M. Zorzi and R. Rao, "Geographic Random Forwarding (GeRaf) for ad hoc and sensor

networks: multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, No. 4, pp. 337–348, 2003.

[10] S. Biswas and R. Morris, "ExOR: Opportunistic Multi-Hop Routing for Wireless Networks," *In Proceedings of ACM SIGCOMM*, pp.133–144, 2005.

[11] Y. W. A.K. Sadek and K. R. Liu, "When does cooperation have better performance in sensor networks?" in *Proceedings of the 3rd IEEE Sensor and Ad Hoc Communications and Networks (SECON'06)*, pp. 188–197, 2006.

[12] E. G. M. Conti and G. Maselli, "Cooperation issues in mobile ad hoc networks," in *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops(ICDCSW'04)*, pp. 803–808, 2004.

[13] J. S. Y. Lin and V. W. Wong, "Cooperative protocols design for wireless ad-hoc networks with multi-hop routing," *Mobile Networks and Applications*, vol. 4, no. 2, pp. 143–153, 2009.

[14] J. C. Z. Zhou, S. Zhou and S. Cui, "Energy-efficient cooperative communication based on power control and selective single-relay in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 8, pp. 3066–3078, 2008.

[15] D. N. T. J. N. Laneman and G. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, 2004.

[16] T. Hunter and A. Nosratinia, "Cooperation diversity through coding," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT'02)*, p. 220–225, 2002.

[17] E. Erkip, A. Sendonaris and B. Aazhang, "User cooperation diversity-part I: System description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, 2003.

[18] E. Erkip, A. Sendonaris and B. Aazhang, "User cooperation diversity-part II: Implementation aspects and performance analysis," *IEEE Transactions on Communications*, vol. 50, no. 11, pp. 1939–948, 2003.

[19] T. Hunter and A. Nosratinia, "Distributed protocols for user cooperation in multi-user wireless networks," in *Proceedings of the 47th IEEE annual Global Telecommunications Conference (GLOBECOM'04)*, pp. 3788–3792, 2004.

[20] Z. L. E. E. P. Liu, Z. Tao and S. Panwar, "Cooperative wireless communications: a

cross-layer approach," *IEEE Wireless Communications*, vol. 13, no. 4, pp. 84–92, 2006.

[21] C. Wan, A. Campbell, and L. Krishnamurthy, "Pump-Slowly, Fetch-Quickly (PSFQ): A Reliable Transport Protocol for Sensor Networks," *IEEE Journal of Selected Areas in Communications*, vol.23, pp.862-872, April. 2005.

[22] F. Stann and J. Heidemann, "RMST: reliable data transport in sensor networks," *in Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications*, pp.102-112, May 2003.

[23] D. Ganesan, R. Govindan, S. Shenker and D. Estrin, "Highly Resilient, Energy Efficient Multipath Routing in Wireless Sensor Networks," In *Mobile Computing and Communications Review (MC2R)*, vol 1., no. 2, pp. 10-24, 2002.

[24] B. Deb, S. Bhatnagar and B. Nath, "ReInForM: reliable information forwarding using multiple paths in sensor networks," *IEEE LCN*, pp.406-415, Oct.2003.

[25] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks," *ACM Wireless Networks*, Vol.11, No.3, pp.285-298, 2005.

[26] Y. Sankarasubramaniam, O. Akan and I. Akyildiz, "ESRT: Event-to- Sink Reliable Transport in Wireless Sensor Networks," *Proc. of ACM MobiHoc'03*, pp.177-188, June 2003.

[27] I. Stojmenovic, "Position-Based Routing in Ad Hoc Networks," *IEEE Comm. Magazine*, Vol.40, No.7, pp.128-134, July 2002.

[28] L. Zou, M. Lu, and Z. Xiong, "A distributed algorithm for the dead end problem of location-based routing in sensor networks," *IEEE Trans. Vehicular Technology*, vol. 54, pp. 1509-1522, July 2005.

[29] M. Chen, T. Kwon, S. Mao, Y. Yuan, and V. Leung, "Reliable and energy-efficient routing protocol in dense wireless sensor networks," *International Journal on Sensor Networks*, vol. 4, no. 12, pp. 104–117, Aug. 2008.

[30] M. Chen, T. Kwon, Y. Yuan, Y. Choi, and V.C.M. Leung, "Mobile agent-based directed diffusion in wireless sensor networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 219-241, 2007.

[31] M. Chen, T. Kwon, S. Mao and V. Leung, "STEER: Spatial-Temporal Relation-Based Energy-Efficient Reliable Routing Protocol in Wireless Sensor Networks," *International*

Journal on Sensor Networks, vol.5, No.2, pp. 129–141, 2008.

[32] http://www.opnet.com

[33] M. Chen, “OPNET Network Simulation,” *Press of Tsinghua University*: Beijing, 2004.

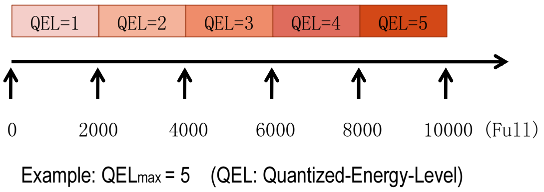


Figure 1.1: Illustration of Node Energy Model Used in RLRR.

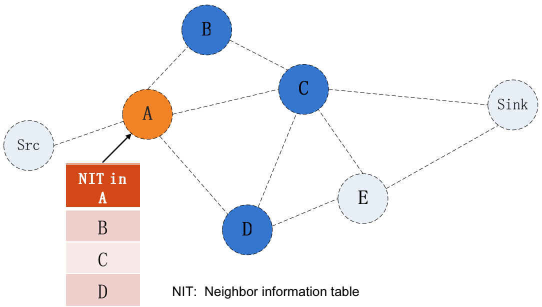


Figure 1.2: Illustration of Neighbor Information Table in a Transmitter-oriented Approach.

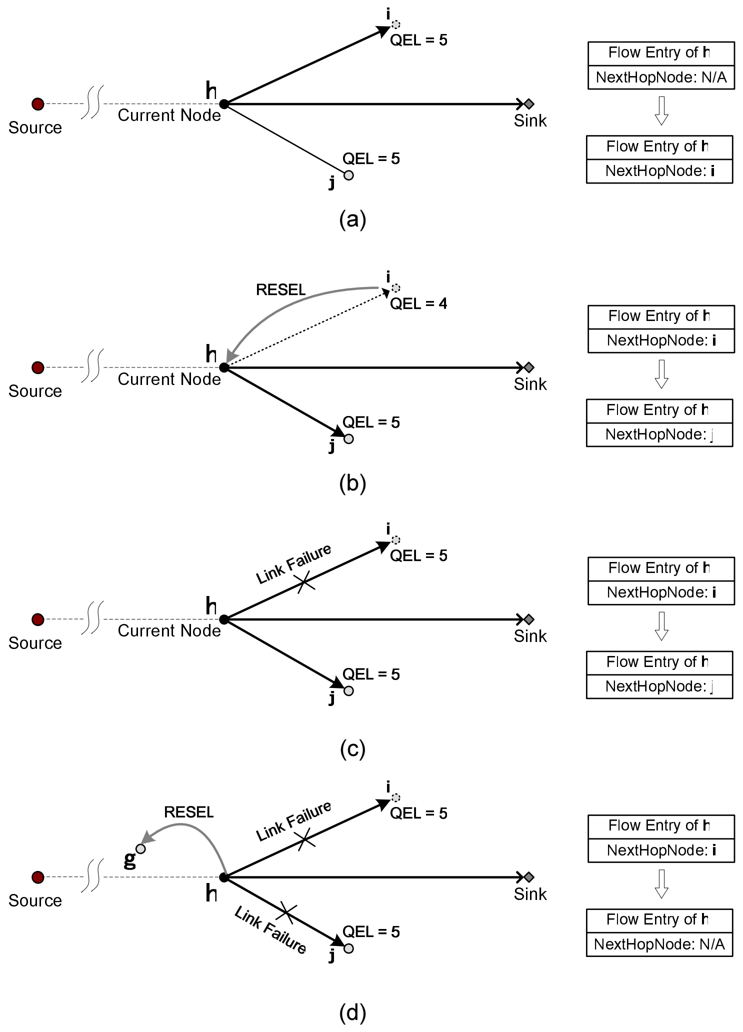


Figure 1.3: Setup/Update Flow Entry in RLRR.

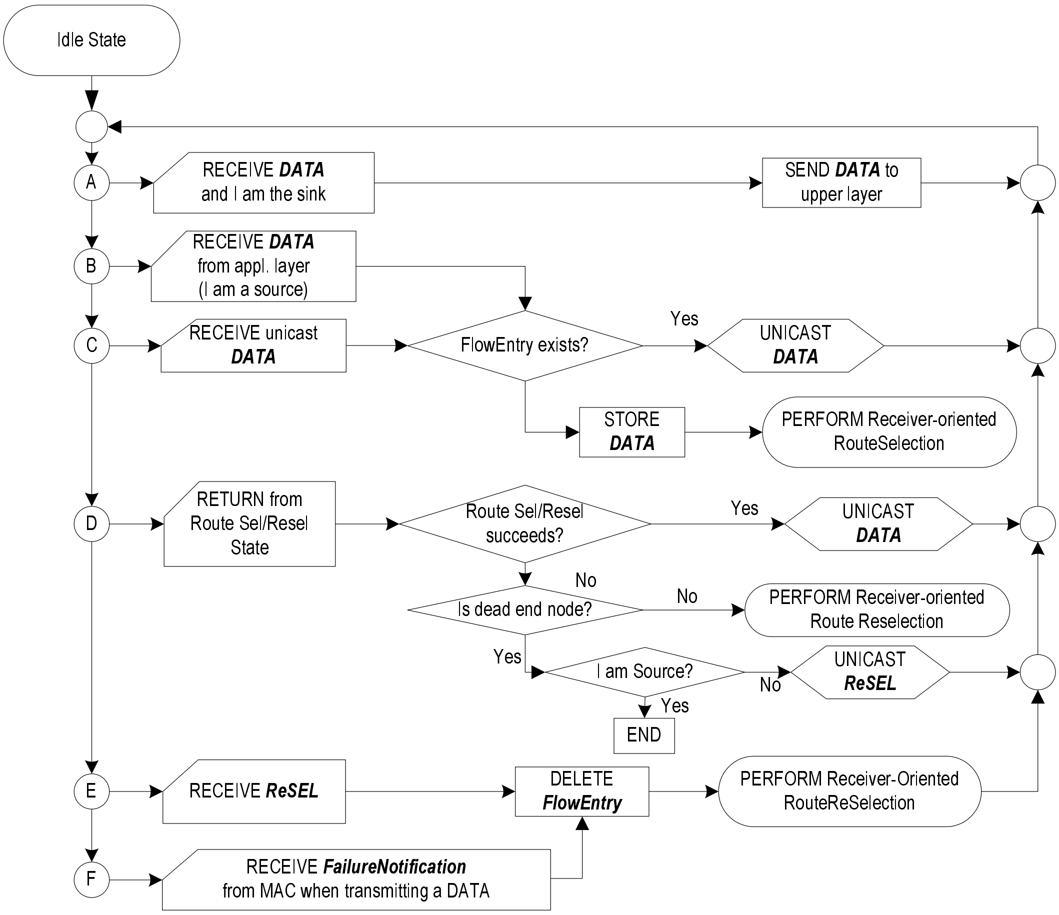
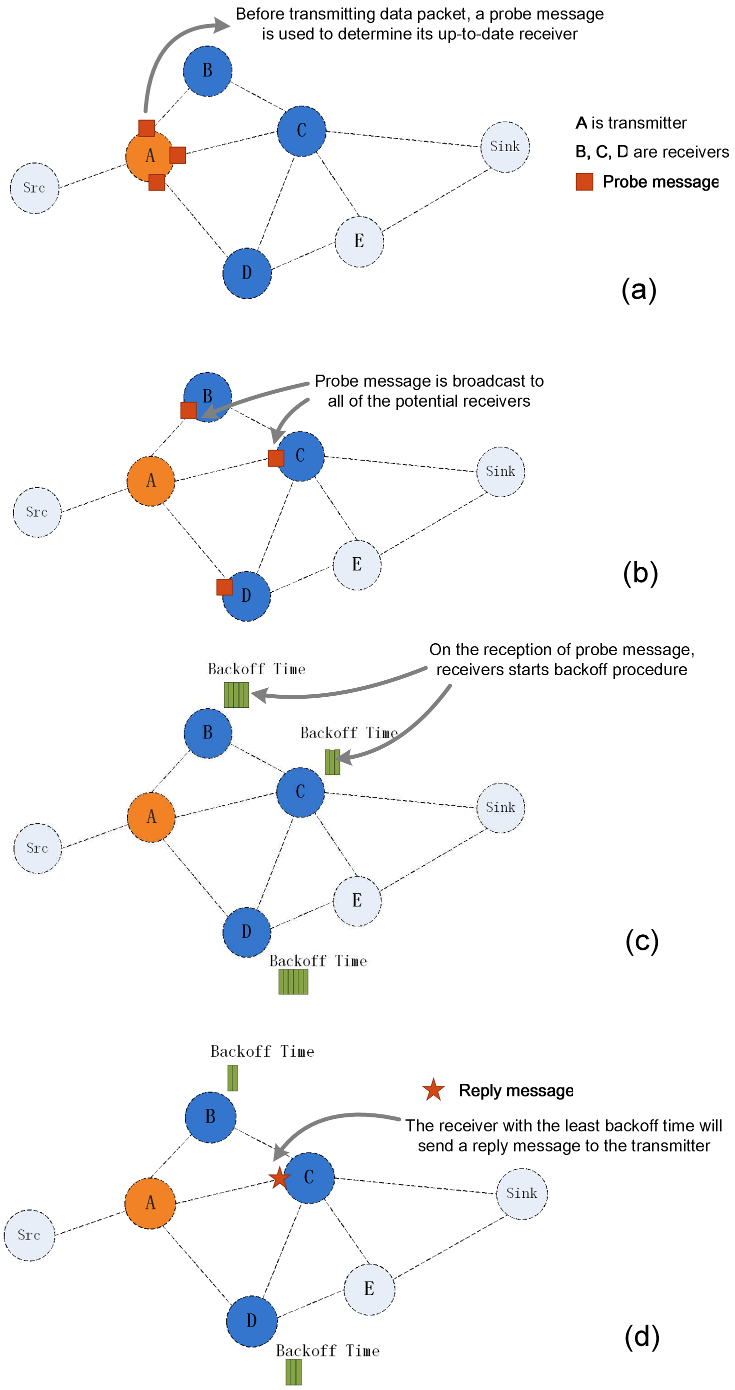


Figure 1.4: Flowchart of the Basic RLRR Protocol.



Before transmitting data packet, a probe message is used to determine its up-to-date receiver

A is transmitter

B, C, D are receivers

🟥 Probe message

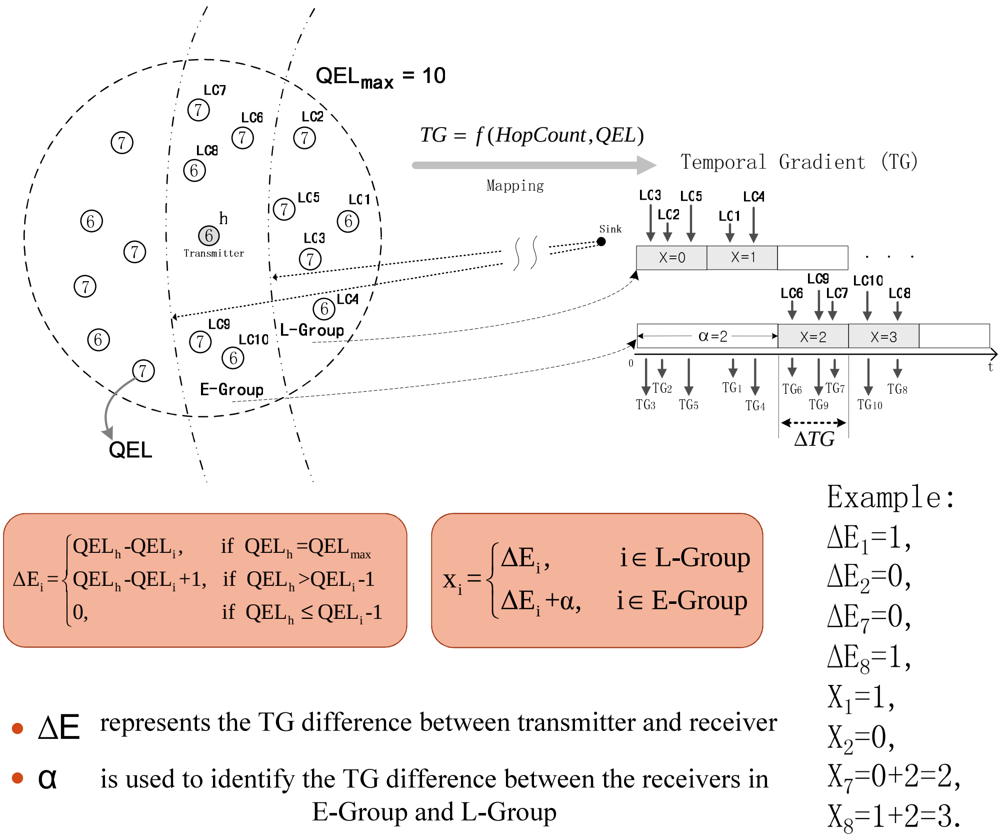
Probe message is broadcast to all of the potential receivers

On the reception of probe message, receivers starts backoff procedure

★ Reply message

The receiver with the least backoff time will send a reply message to the transmitter

Figure 1.5: Illustration of Receiver-oriented Mechanism in RLRR.



$$\Delta E_i = \begin{cases} QEL_h - QEL_i, & \text{if } QEL_h = QEL_{max} \\ QEL_h - QEL_i + 1, & \text{if } QEL_h > QEL_i - 1 \\ 0, & \text{if } QEL_h \leq QEL_i - 1 \end{cases}$$

$$x_i = \begin{cases} \Delta E_i, & i \in \text{L-Group} \\ \Delta E_i + \alpha, & i \in \text{E-Group} \end{cases}$$

- ΔE represents the TG difference between transmitter and receiver
- α is used to identify the TG difference between the receivers in E-Group and L-Group

Example:
$\Delta E_1=1$,
$\Delta E_2=0$,
$\Delta E_7=0$,
$\Delta E_8=1$,
$X_1=1$,
$X_2=0$,
$X_7=0+2=2$,
$X_8=1+2=3$.

Figure 1.6: Converting Energy and Hop Count information into Temporal Gradient.

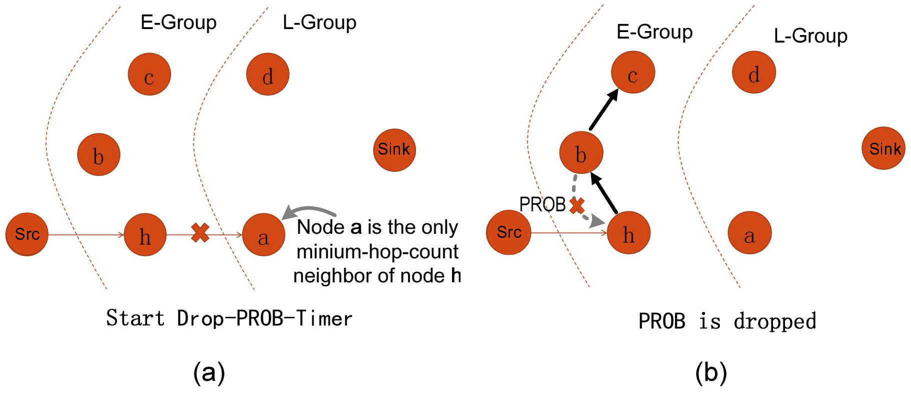
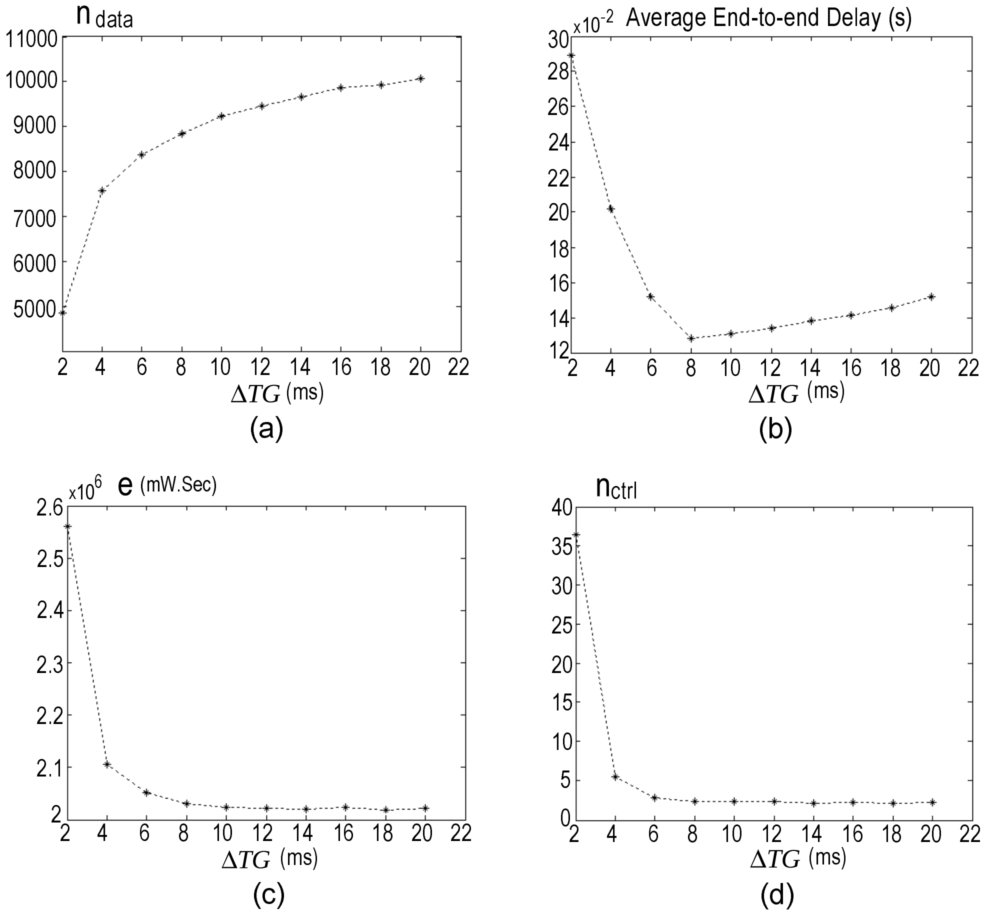


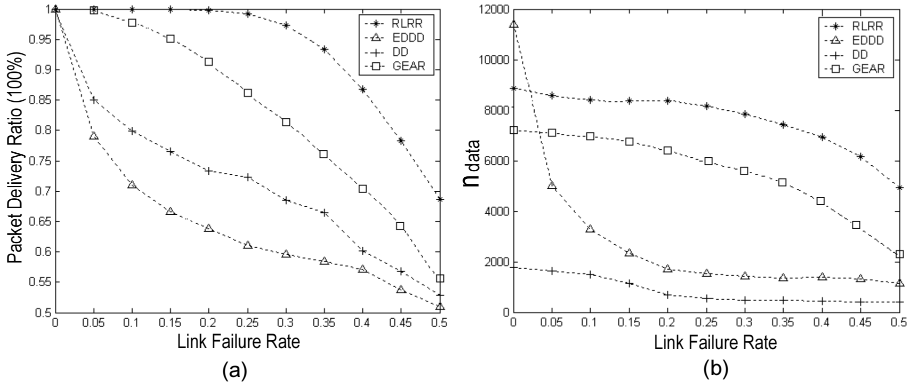
Figure 1.7: Illustration of Guaranteeing Loop Freedom.



(a) (b)

(c) (d)

Figure 1.8: The impact of ΔTG on: (a) n_{data}; (b) end-to-end delay; (c) e and (d) n_{ctrl}.



(a) (b)

Figure 1.9: The impact of link failure rate on: (a) reliability and (b) lifetime (n_{data}).

Index

Chen, M.

Ji, W.

Wang, X. F.

Cai, W.

Liao, L. X.