WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2009; **9**:405–416
Published online 4 September 2007 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.543

Receiver-oriented load-balancing and reliable routing in wireless sensor networks

Min Chen[1], Victor C. M. Leung[1], Shiwen Mao[2] and Taekyoung Kwon[3]*,†
[1]*Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada*
[2]*Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, U.S.A.*
[3]*School of Computer Science and Engineering, Seoul National University, Seoul, Korea*

Summary

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. In this paper, a novel receiver-oriented load-balancing and reliable routing (RLRR) protocol is proposed. In RLRR, an intermediate node solicits next hop candidates, each of which is to respond with its own backoff time dubbed a *temporal gradient* (TG). 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. Furthermore, neighbor nodes whose hop count is less than the soliciting node participate in the next-hop selection process with loop-free operation guarantee. Comprehensive simulations are carried out to show that RLRR achieves relatively longer network lifetime and higher reliability than other existing schemes. Copyright © 2007 John Wiley & Sons, Ltd.

KEY WORDS: load balancing; reliability; energy efficient; wireless sensor network

1. Introduction

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.

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 paper, the existing load-balancing schemes are classified into two categories: local load balancing [1–3] and global load balancing [4]. 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

*Correspondence to: Taekyoung Kwon, School of Computer Science and Engineering, Seoul National University, Seoul, Korea.
†E-mail: tkkwon@snu.ac.kr

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 (e.g., EDDD [4]) aim to find globally load-balanced paths to achieve a higher network lifetime.

In References [1,4], 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 [1,4], which operate in the following two-step manner: (1) select the next hop node first based on a neighbor information table; (2) forward packet to the selected node until a predetermined number of transmissions fail. We call this a 'transmitter-oriented' approach.

In this paper, a novel 'receiver-oriented' load-balancing reliable routing protocol (RLRR) is proposed. RLRR aims to achieve both load balancing and reliability for large-scale WSNs. 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* (TG). The next hop candidate with the least TG (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 TG 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 to enhance reliability while guaranteeing loop-free routing. Our receiver-oriented idea is close to GeRaF [5] and ExOR [6] 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 [4], DD [7] and GEAR [1]. 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 paper is organized as follows. We describe RLRR design issues and its algorithm in Sections 2 and 3, respectively. Simulation model and experimental results are presented in Sections 4 and 5, respectively. Finally, Section 6 concludes the paper.

2. RLRR Design Issues

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.

DOI: 10.1002/wcm

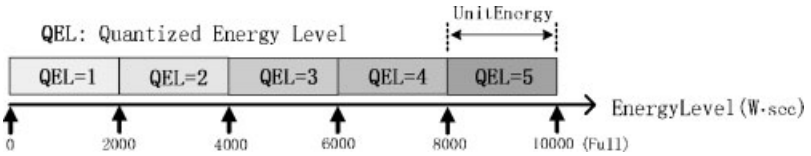

Fig. 1. Node energy model.

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). An example is given in Figure 1. Assuming the full energy level ($E_{\max}$) of a battery is equal to 10 000, and the 'unit energy' (the unit of the quantization, E_{unit}) is equal to 2000. Then, the maximum value of QEL is ($\text{QEL}_{\max} = \lceil \frac{E_{\max}}{E_{\text{unit}}} \rceil = 5$. In this paper, 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, that is, 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, $\text{QEL}_{\max}$ should be optimized to achieve the best tradeoff between load balancing and control overhead.

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.

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, peer neighbors can also be backup nodes 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 Subsection 3.4.

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 identifiers of its next hop node and upstream node of 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.

3. The RLRR Mechanism

3.1. The RLRR Protocol

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

DOI: 10.1002/wcm

network. Each sensor sets up its hop count gradient to the sink. Sensor(s) matching the interest will become the source node(s) [7]. 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 Figure 2(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 Figure 2(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*.

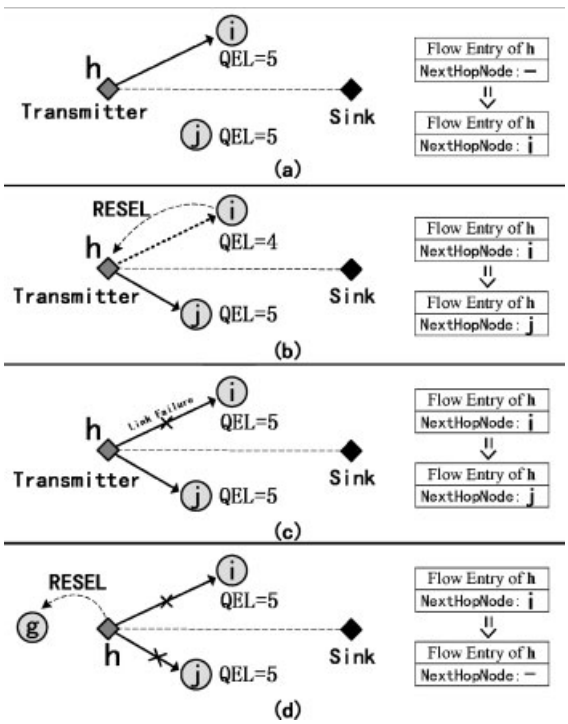

Fig. 2. Setup/update flow entry in RLRR.

In addition to balancing the energy consumption, route reselection is also triggered to recover a link failure. In the example shown in Figure 2(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 favorable, node *j* 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 Figure 2(d)), which triggers a new route Reselection by node *g*, and so forth. We employ the similar *route selection/reselection mechanism* proposed in Reference [8].

3.2. Time Gradient Calculation

In RLRR, the energy and hop count information of each 'live candidate' (LC) node 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 *h*, which hop counts to the sink are less than or equal to that of *h*. To perform route Sel/Resel [8], *h* 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 *h* when its '*TG-Timer*' expires. The LC that originated the first REP received by *h* 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 Figure 3, 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*.

DOI: 10.1002/wcm

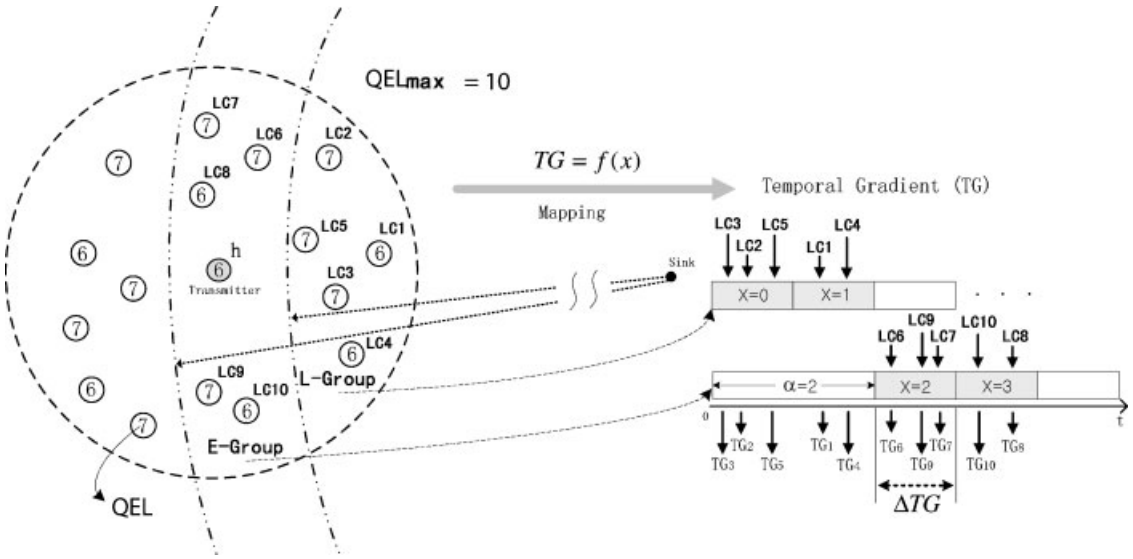

Fig. 3. Converting energy and hop count information into temporal gradient.

Obviously, the LCs in *L-Group* should have higher priority than those in *E-Group*. In Figure 3, 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 Figure 3, $\text{QEL}_{\max}$ 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 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 Equation (1), where x is a parameter reflects both ΔEs and the type of group which an LC belongs to.

$$\Delta E_i = \begin{cases} \text{QEL}_h - \text{QEL}_i, & \text{if } \text{QEL}_h = \text{QEL}_{\max} \\ \text{QEL}_h - \text{QEL}_i + 1, & \text{if } \text{QEL}_h > \text{QEL}_i - 1 \\ 0, & \text{if } \text{QEL}_h \le \text{QEL}_i - 1 \end{cases}$$

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

$$TG_i = f(x_i) = x_i \times \Delta\text{TG} + rand(\Delta TG) \quad (1)$$

In Equation (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 $rand(\Delta\text{TG})$ 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: $\delta = N/A$. Let r be the transmission range of a sensor node. Roughly, LCs are located within approximately one-third of the whole transmission range in Figure 3. Then, the number of LCs can be approximated by:

$$L = \frac{1}{3} \cdot \pi \cdot r^2 \cdot \delta \quad (2)$$

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

DOI: 10.1002/wcm

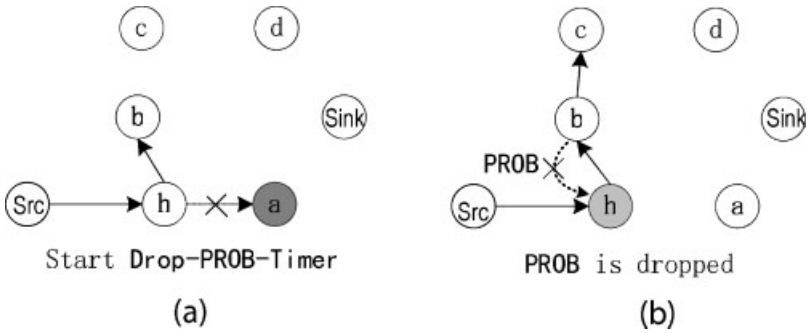

Fig. 4. Illustration of guaranteeing loop freedom. (a) Start Drop-PROB-Timer (b)PROB is dropped.

other LCs in the same *S-Group*, at least T_{REP} should be reserved for each LC. Thus, ΔTG is approximately equal to:

$$\Delta \mathrm{TG} = \frac{L}{2} \cdot T_{\mathrm{REP}} \tag{3}$$

Here, $L/2$ is the average number of LCs in an *S-Group*. Let α be 2. In the example shown in Figure 3, we can get four *S-Group*s: 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 TGs of the LCs in each *S-Group* are randomly distributed over a range of ΔTG. In the example in Figure 3, the increasing order of the TGs of all the LCs 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.

3.3. Solving the Dead End Problem

The so-called dead end problem [9] arises when a packet is forwarded to a local optimum, that is, 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*.

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 LCs in *E-Group* are exploited to achieve better performance in terms of both load balancing and reliability. While using the LCs 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 Figure 4.

In Figure 4(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 Figure 4(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* ($T_{Drop\text{-}PROB\text{-}Timer}$) should be long enough, such as:

$$T_{Drop\text{-}PROB\text{-}Timer} = N_{E\text{-}Group} \cdot t_{\mathrm{sel}}. \tag{4}$$

In Equation (4), $N_{E\text{-}Group}$ denotes the maximum number of LCs in *E-Group*. t_{sel} denotes the time for one-hop route selection. Considering the worst case

DOI: 10.1002/wcm

where all the LCs 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 Equation (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 LCs in the same *E-Group*. Thus, loop freedom is guaranteed.

4. Simulation Methodology

4.1. Simulation Settings

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 [4]), a local load-balancing scheme (i.e., GEAR [1]), and a non-load-balancing scheme (i.e., DD [7]) to compare with RLRR. We implement the RLRR protocol and perform simulations using OPNET Modeler [10,11]. 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 $500 \times 500\,\mathrm{m}^2$ field. As in Reference [12], 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 s interval between packets (1 K Bytes each). As in Reference [13], we use IEEE 802.11 DCF as the underlying MAC, and the radio transmission range (R) is set to 45 m. 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, for example, 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 References [8,12] and link failure model used in Reference [14]. For each set of results, we simulate the WSN 60 times with the specific set of parameters and different random seeds.

4.2. Performance Metrics

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 paper.
- *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 References [8,12–14], 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.

5. Performance Evaluation

5.1. Performance Evaluation of RLRR With Different Design Parameters

There are three main design parameters in RLRR, including $\mathrm{QEL}_{\max}$, α, and $\Delta\mathrm{TG}$. In the following, we examine the impact of these parameters on the RLRR performance.

DOI: 10.1002/wcm

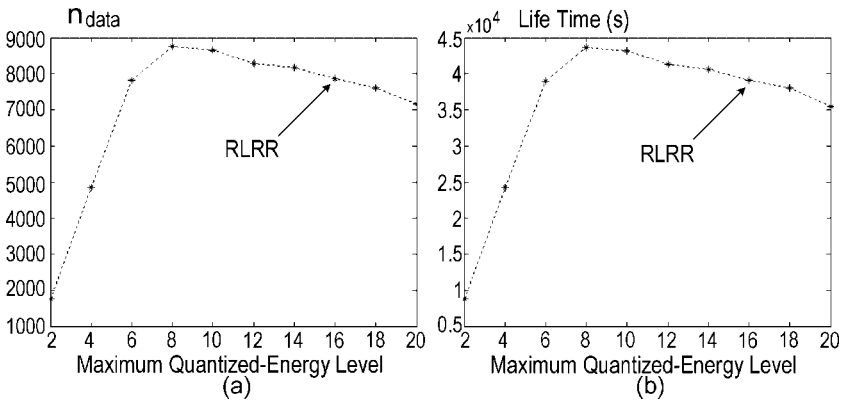

Fig. 5. The impact of $\text{QEL}_{\max}$ on: (a) n_{data} and (b) lifetime.

(a) *Effects of* $\text{QEL}_{\max}$: In these experiments, we change $\text{QEL}_{\max}$ from 2 to 20. Recall that $\text{QEL}_{\max}$ indicates the maximum quantized energy level when the battery capacity is full.

In Figure 5(a), n_{data} grows as $\text{QEL}_{\max}$ increases when $\text{QEL}_{\max}$ has a small value, and reaches its maximum when $\text{QEL}_{\max}$ is equal to 8. If $\text{QEL}_{\max}$ exceeds 8, the increased $\text{QEL}_{\max}$ causes n_{data} to decrease since the limited performance enhancement of load balancing cannot compensate for the effect of the increased control overhead. In Figure 5(b), the trend of lifetime follows closely that of n_{data} because the packet delivery ratio is always 100%.

As mentioned in Subsection 3.1, to achieve load balancing, a node on the path transmits a RESEL to its upstream node when its QEL is reduced by 1. The larger is $\text{QEL}_{\max}$, the higher is the frequency of next-hop-oscillation, and the more control overhead is incurred for route Resel. Thus, in Figure 6(a), e increases when $\text{QEL}_{\max}$ is increased.

However, in Figure 6(b), with increasing $\text{QEL}_{\max}$, n_{ctrl} decreases initially and then increases. Let N_f be the number of messages during the interest flooding phase. Note that N_f is a fixed value since the sink only floods control messages once in RLRR. Let N_s be the total number of control messages used for route Sel/Resel. Then, n_{ctrl} is equal to:

$$n_{\text{ctrl}} = \frac{N_f + N_s}{n_{\text{data}}} \tag{5}$$

If $\text{QEL}_{\max}$ is very small (say, 2), n_{data} is small, which makes n_{ctrl} large in Figure 6(b). When $\text{QEL}_{\max}$ is up to 8, the quickly increased n_{data} dominates Equation 5, and makes n_{ctrl} decrease. Furthermore, when $\text{QEL}_{\max}$ goes beyond 8, n_{data} begins to decrease slightly (see Figure 5(a)). However, N_s increases in proportion to $\text{QEL}_{\max}$, and dominates Equation 5. Thus, n_{ctrl} increases again.

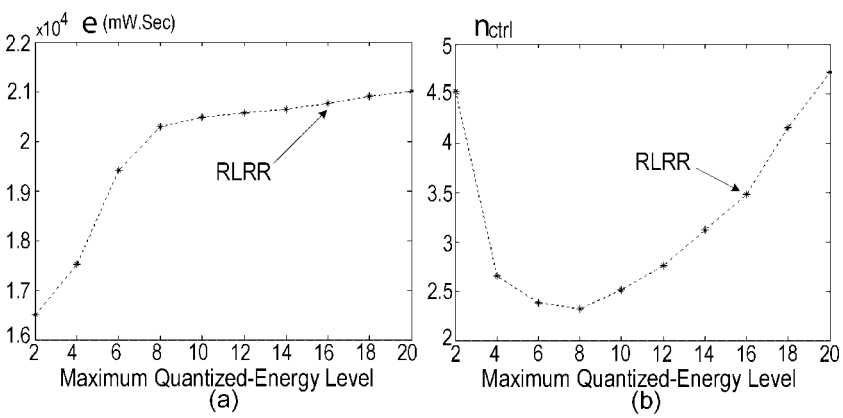

Fig. 6. The impact of $\text{QEL}_{\max}$ on: (a) e and (b) n_{ctrl}.

DOI: 10.1002/wcm

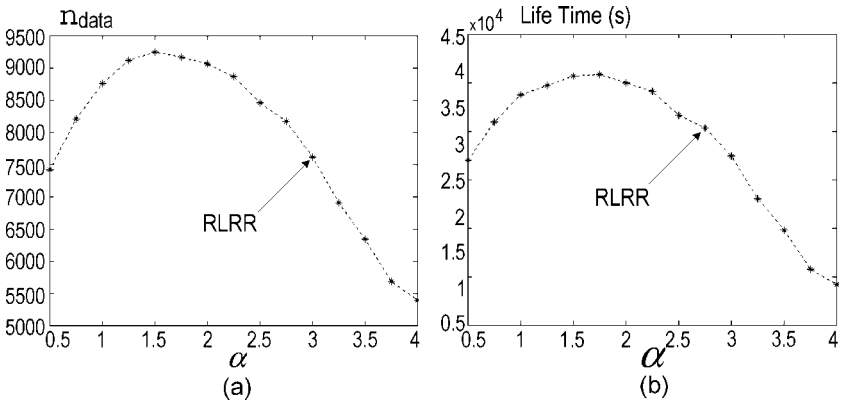

Fig. 7. The impact of α on: (a) n_{data} and (b) lifetime.

(b) *Effects of* α: In these experiments, we change α from 0.5 to 4 by the step size of 0.25. Recall that α is an offset used to differentiate the TGs of LCs between the *L-Group* and the *E-Group*. The lower is α, the smaller is the difference in TG values between the *L-Group* and the *E-Group*. As an extreme example, if $\alpha = 0$, the x values of the LCs in the *L-Group* and the *E-Group* will overlap (the case of $\alpha = 2$ is shown in Figure 3). Thus, twice the number of *TG-Timer*s may expire in the same ΔTG period, which is more likely to cause REPs to collide with each other. In Figure 7(a), n_{data} increases with α, and reaches its maximum value when $\alpha = 1.5$. Then n_{data} decreases when α is further increased, since the larger is α, the less load balancing is achieved among all the LCs in both the *L-Group* and the *E-Group*. As an extreme example, if α is close to ∞, the LCs in the *E-Group* will never be selected as the next hop node. When the LCs in the *L-Group* deplete their energy, the LCs in the *E-Group* may still have plenty of battery power. This phenomenon also demonstrates the positive effect of adopting *E-Group* for alternate routing. Figure 7(b) shows a similar trend as Figure 7(a).

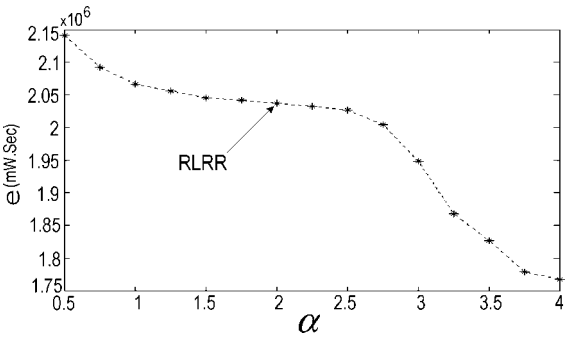

Fig. 8. The impact of α on e.

The larger is α, the less load balancing is achieved, the less is the frequency of route reselection, and hence the less control overhead is needed. Thus, e decreases with α as shown in Figure 8.

(c) *Effects of* ΔTG: In these experiments, we change ΔTG from 2 to 20 ms by the step size of 2 ms. In Figure 9(a), n_{data} increases as ΔTG is increased,

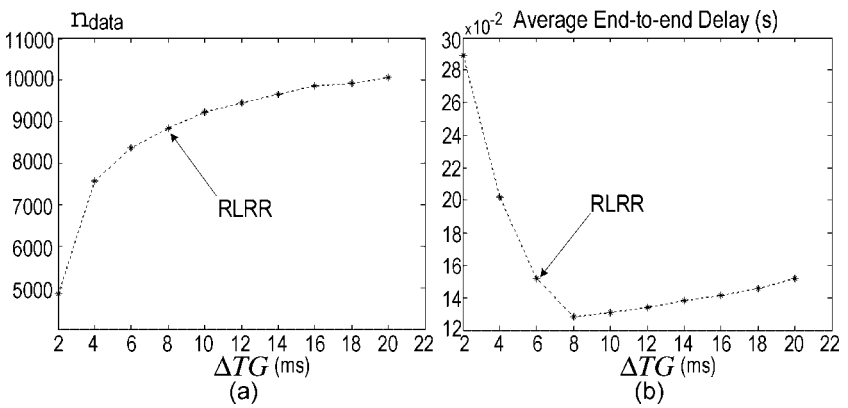

Fig. 9. The impact of ΔTG on: (a) n_{data} and (b) end-to-end delay.

DOI: 10.1002/wcm

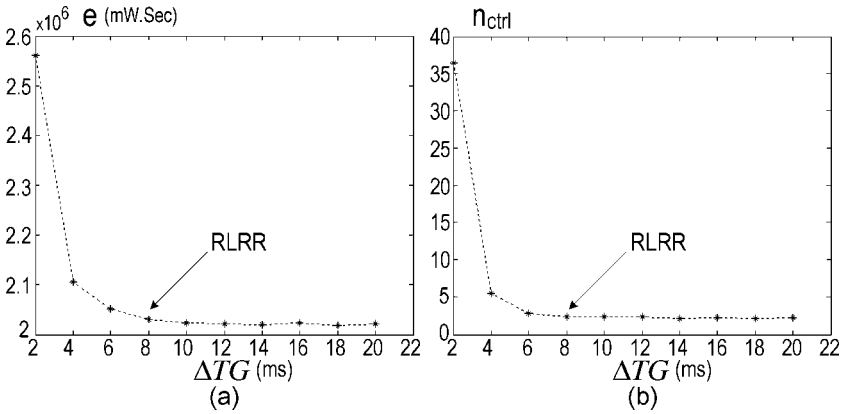

Fig. 10. The impact of ΔTG on: (a) e and (b) n_{ctrl}.

since the larger is ΔTG, the less collisions will happen, and the more data packets will be delivered successfully to the sink.

In Figure 9(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 ms. It is unnecessary to increase ΔTG more if the value is large enough to differentiate the LCs in the same *S-Group*, since a large ΔTG also increases the time for route Sel/Resel. Thus, when ΔTG goes beyond 8 ms, the delay begins to increase again.

In Figure 10(a) and 10(b), both e and n_{ctrl} decreases with ΔTG increasing. The larger is ΔTG, the less likely collision happens. Thus, the control overhead decreases.

5.2. 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. Figure 11 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 per cent packet delivery ratio until the link failure rate is larger than 0.35.

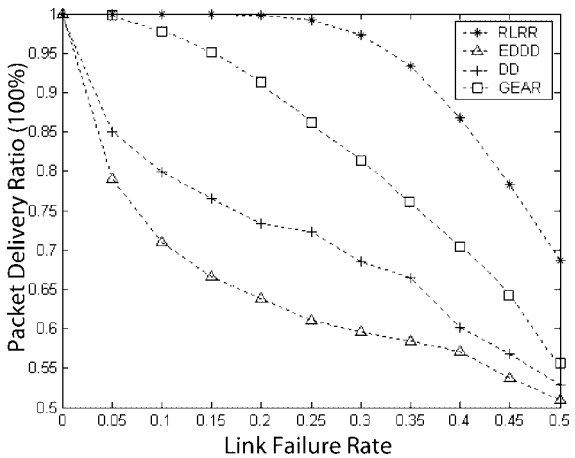

Fig. 11. Comparison of reliability.

In Figure 12, 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

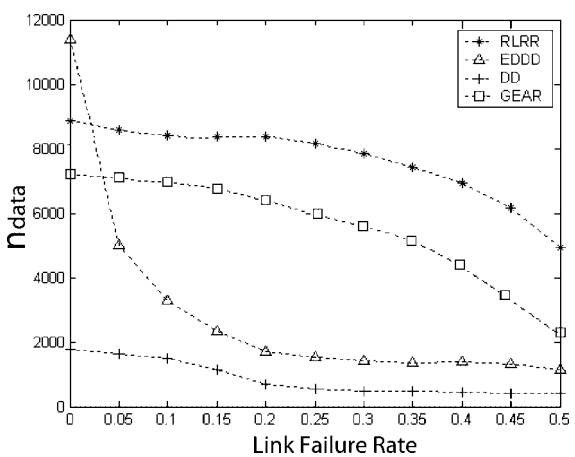

Fig. 12. Comparison of lifetime (n_{data}).

DOI: 10.1002/wcm

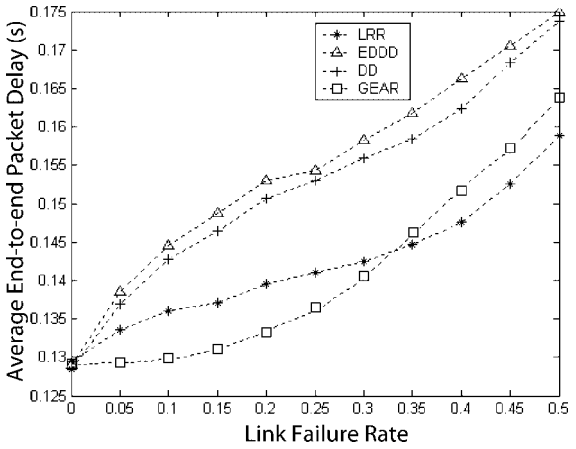

Fig. 13. Comparison of end-to-end delay.

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.

In Figure 13, the end-to-end delays of all the schemes increase as link failure rate is increased. RLRR exhibits a lower end-to-end delay than EDDD and DD in case of link failure. It is because RLRR recovers from link/node failure faster than EDDD or DD. When the link failure rate is smaller than 0.35, RLRR has higher delays than GEAR because next hop selection delay is introduced during route Sel/Resel. If the link failure rate goes beyond 0.35, the delay of RLRR becomes the lowest.

Since EDDD and DD rely on message flooding to set up/update gradients, if all the backup nodes are broken, sensor nodes will wait for the sink node to initiate a new interest flooding to refresh the stale routes. In Figure 14, the e performances of RLRR and GEAR are better than those of EDDD and DD, since GEAR and RLRR never use control message flooding to repair a route. However, GEAR needs to exchange hello messages to update energy information. Thus, e values of RLRR are less than those of GEAR. In addition, note that GEAR utilizes geographical information and requires that sensor nodes are equipped with GPS receivers. In contrast, RLRR does not need any geographical information. Furthermore, RLRR does not need to store the neighbor information in terms of energy and backup nodes as GEAR and EDDD do. Thus, RLRR can support low cost design for sensor nodes while achieving good performance as described above.

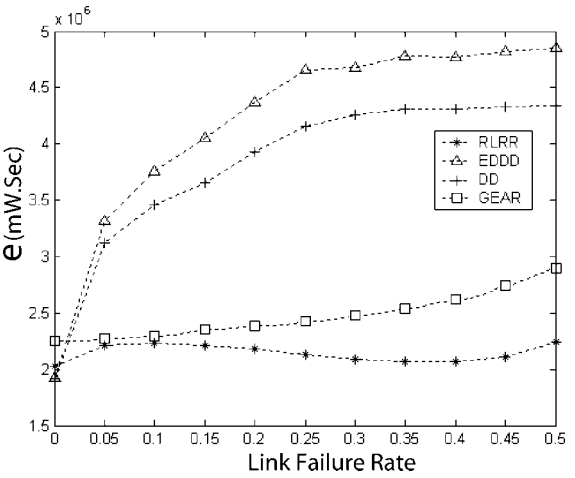

Fig. 14. Comparison of energy consumption.

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.

6. Conclusion

We have proposed the novel RLRR protocol that employs a receiver-oriented approach to achieve both load balancing and reliability simultaneously in large-scale WSNs. 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 RLRR, 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. We have presented extensive simulation results to show that load balancing is achieved at the expense of energy for transmitting control messages. Thus, the related parameters should be selected carefully to achieve load balancing with energy efficiency while minimizing the control overhead.

Acknowledgments

This work was supported in part by the Canadian Natural Sciences and Engineering Research Council

DOI: 10.1002/wcm

under grant STPGP 322208-05, the Basic Research Program of the Korea Science and Engineering Foundation under grant No. (R01-2004-000- 10372-0), and the OPNET University Program.

References

1. Yu Y, Govindan R, Estrin D. Geographical and Energy Aware Routing. *UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023*, May 2001.
2. Dai H, Han R. A node-centric load balancing algorithm for WSNs. In *Proceedings of IEEE GLOBECOM*, December 2003
3. Gao J, Zhang L. Load balancing shortest path routing in wireless networks. In *Proceedings of IEEE INFOCOM*, 2004
4. Chen M, Kwon T, Choi Y. Energy-efficient differentiated directed diffusion for real-time traffic in wireless sensor networks. *Computer Communications* 2006; **29**(2): 231–245.
5. Zorzi M, Rao R. Geographic random forwarding (GeRaf) for ad hoc and sensor networks: multihop performance. *IEEE Transactions on Mobile Computing* 2003; **2**(4): 337–348.
6. Biswas S, Morris R. ExOR: opportunistic multi-hop routing for wireless networks. *In Proceedings of ACM SIGCOMM*, 2005
7. Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of ACM MobiCom*, Boston, MA, August 2000.
8. Chen M, Kwon T, Mao S, Leung VCM. Spatial temporal relation-based energy-efficient reliable routing protocol in wireless sensor networks. *International Journal on Sensor Networks* 2007 (in press).
9. Zou L, Lu M, Xiong Z. A distributed algorithm for the dead end problem of location-based routing in sensor networks. *IEEE Transactions on Vehicular Technology* 2005; **54**: 1509–1522.
10. http://www.opnet.com
11. Chen M. *OPNET Network Simulation*. Press of Tsinghua University: Beijing, 2004.
12. Chen M, Kwon T, Yuan Y, Choi Y, Leung V.C.M. Mobile agent-based directed diffusion in wireless sensor networks. *EURASIP Journal on Advances in Signal Processing* 2007. DOI:10.1155/2007/36871
13. Chen M, Leung VCM, Mao S, Yuan Y. Directional geographical routing for real-time video communications in wireless sensor networks. *Elsevier Computer Communications* 2007. DOI:10.1016/j.comcom.2007.01.016
14. Chen M, Kwon T, Mao S, Yuan Y, Leung VCM. Reliable and energy-efficient routing protocol in dense wireless sensor networks. *International Journal on Sensor Networks* 2007 (in press).

Authors' Biographies

Min Chen was born in December 1980. He received the B.S., M.S., and Ph.D. degrees from the Department of Electronic Engineering, South China University of Technology, in 1999, 2001, and 2004, respectively. Since 2006, he is a postdoctoral fellow in the Communications Group, Department of Electrical and Computer Engineering, University of British Columbia. He was a post-doctoral researcher in the School of Computer Science and Engineering, Seoul National University for $1\frac{1}{2}$ years.

Victor C. M. Leung received the B.A.Sc. (Hons.) and Ph.D. degrees, both in Electrical Engineering, from the University of British Columbia (UBC) in 1977 and 1981, respectively. He is a professor in the Department of Electrical and Computer Engineering, UBC. Dr. Leung is a Fellow of IEEE and a voting member of ACM. He is a member of the editorial boards of the *IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, International Journal of Sensor Networks, and International Journal of Communication Networks and Distributed Systems*.

Shiwen Mao received the B.S. degree and the M.S. degree from Tsinghua University, Beijing, P.R. China in 1994 and 1997, respectively, both in Electrical Engineering, and the M.S. degree in System Engineering from Polytechnic University, Brooklyn, NY, in 2000. He received the Ph.D. degree in Electrical and Computer Engineering from Polytechnic University in 2004. Currently, he is an Assistant Professor in the Department of Electrical and Computer Engineering at Auburn University, Auburn, AL. Before joining Auburn University, he had been a Research Scientist in the Bradley Department of Electrical and Computer Engineering at Virginia Tech. for over 2 years.

Taekyoung Kwon has been an Assistant Professor in the School of Computer Science and Engineering, Seoul National University (SNU) since 2004. Before joining SNU, he was a Post-Doctoral Research Associate at UCLA and at City University New York (CUNY). He obtained the B.S., M.S., and Ph.D. degrees from the Department of Computer Engineering, SNU, in 1993, 1995, 2000, respectively.

DOI: 10.1002/wcm