Improve Efficiency and Reliability in Single-Hop WSNs with Transmit-Only Nodes

Jia Zhao, *Student Member*, *IEEE*, Chunming Qiao, *Fellow*, *IEEE*,
Raghuram S. Sudhaakar, *Member*, *IEEE*, and Seokhoon Yoon, *Member*, *IEEE*

Abstract—Wireless Sensor Networks (WSNs) will play a significant role at the "edge" of the future "Internet of Things." In particular, WSNs with transmit-only nodes are attracting more attention due to their advantages in supporting applications requiring dense and long-lasting deployment at a very low cost and energy consumption. However, the lack of receivers in transmit-only nodes renders most existing MAC protocols invalid. Based on our previous study on WSNs with pure transmit-only nodes, this work proposes a simple, yet cost effective and powerful single-hop hybrid WSN cluster architecture that contains not only transmit-only nodes but also standard nodes (with transceivers). Along with the hybrid architecture, this work also proposes a new MAC layer protocol framework called Robust Asynchronous Resource Estimation (RARE) that efficiently and reliably manages the densely deployed single-hop hybrid cluster in a self-organized fashion. Through analysis and extensive simulations, the proposed framework is shown to meet or exceed the needs of most applications in terms of the data delivery probability, QoS differentiation, system capacity, energy consumption, and reliability. To the best of our knowledge, this work is the first that brings reliable scheduling to WSNs containing both nonsynchronized transmit-only nodes and standard nodes.

Index Terms—Wireless sensor networks, medium access control, transmit-only node, Internet of things, data communication

1 Introduction

A key issue in Wireless Sensor Network (WSN) system design is to minimize the overall system cost (deployment, operation, maintenance, and abolishment) and power consumption. Most of the existing WSN systems rely on standard full-fledged transceivers equipped on each sensor node for communication with sinks and each other. However, in most cases, the receiver module of a transceiver is more costly and consumes more energy than the transmitter [1], [2]. Moreover, many WSN applications normally contain hundreds of sensor nodes whose ultimate goal is simply to report sensed data to the sink periodically and/or when a threshold is reached, without need for any external control. The receiver module is, thus, not necessary for these nodes.

Take the (futuristic) intravehicular sensor network application [3] for example, where a few hundreds of in-car sensor nodes are used to sample the data from different parts of the vehicle. Since these sensors are only responsible for reporting the data to the Electronic Control Unit (ECU), which is within one hop, they do not need to have any receiving capability. Similar applications include tracking in Wireless Body Area Networks (WBANs) [4] and Wireless Personal Area Networks (WPANs) [5], household activities inference [6], [7], and telemetry in larger scale WSN systems such as precision agriculture [8], industrial automation [9], where there are a large number of sensors, each of which needs to send a small sample (i.e., sensed data) to a sink (or a special relay/mesh node) within its transmission range once in a while. Since the overall system cost is an important factor that affects the adoption of the technology, WSN systems that adopt function-reduced and energy-efficient nodes, such as those with only transmitters have received increasing attention.

Besides the overall system cost, Quality of Service (QoS) differentiation is also an important feature demanded by today's WSN applications. More specifically, sensor nodes may require different level of QoS support since they perform tasks of different level of importance in the system. In the above intravehicular sensor network example, brake status is obviously much more important than the outside temperature and thus should have a higher priority in the system.

Moreover, whether the WSN systems could continuously provide reliable data-gathering service is another critical issue. In fact, system reliability is a top requirement in the industry applications. Generally, the factors that affect the wireless communications can be categorized into *three* types: 1) radio frequency interference that could occur almost anytime and anywhere; 2) network change that is either predictable as a result of planned adjustment or unpredictable due to the node failures; and 3) environmental effects (e.g., bad weather and blocking objects) that affect the wireless channel condition. It is desirable to have robust WSN systems that are able to cope with these negative effects.

To realize the potential cost and energy saving benefits of transmit-only nodes, we studied WSNs consisting of just

- *J. Zhao and C. Qiao are with the Department of Computer Science and Engineering, The State University of New York (SUNY), Davis Hall, Buffalo, NY 14260. E-mail: jiazhao@buffalo.edu, qiao@computer.org.*
- *R.S. Sudhaakar is with Cisco Systems, Inc., San Jose, CA 95134 and at SJC24/3/3, 510 McCarthy Blvd, Milpitas, CA 95036. E-mail: rsudhaak@cisco.com.*
- *S. Yoon is with the Department of Computer Engineering and Information Technology, School of Electrical Engineering, University of Ulsan, Room #206, 7th Building, Ulsan 680749, Korea. E-mail: seokhoonyoon@ulsan.ac.kr.*

Manuscript received 15 Oct. 2011; revised 10 Apr. 2012; accepted 5 May 2012; published online 16 May 2012.
Recommended for acceptance by K. Li.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-10-0769.
Digital Object Identifier no. 10.1109/TPDS.2012.148.

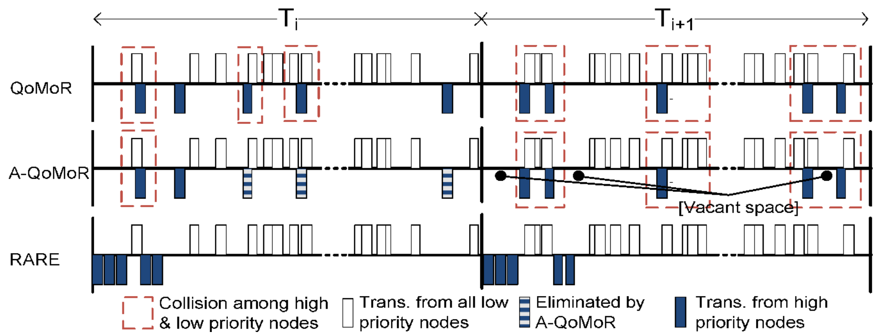


Fig. 1. Comparison of different protocols/framework.

transmit-only nodes (and a receiver-only sink). However, such a setting presents many challenges. First, since the transmit-only nodes cannot perform carrier-sensing as in the Carrier Sense Multiple Access (CSMA) systems, data transmissions by these nodes are completely uncoordinated. This characteristic rules out the use of most existing MAC protocols such as IEEE 802.11 [10], B-Mac [11], S-Mac [12], and TSMP [13]. Second, although the cost and power savings due to the use of transmit-only nodes can be significant in a WSN with larege number of nodes, when a greedy scheme is used by these nodes in an attempt to achieve a better data delivery performance, the transmissions will quickly saturate the shared channel, and collisions will deteriorate the overall performance. Third, our previous studies on WSNs with pure transmit-only nodes [14], [15] showed that one can achieve a satisfactory data delivery performance with up to a few hundred of such nodes within one-hop from the sink. In other words, in a denser deployment environment requiring more than a couple of hundreds of sensor nodes, having transmit-only nodes is not sufficient.

Taking above requirements into consideration, this work proposes to study a **hybrid** WSN cluster that contains not only the transmit-only nodes but also the standard nodes with tranceivers. The objective is to appropriately trade a small increase in cost/complexity for a significant performance improvement by separating data transmissions of the different priority categories of nodes. In order to tackle the new challenges in such a hybrid WSN, we propose a robust MAC protocol framework based on our preliminary works in [16] and [17]. In addition to integrating pieces from our previous effort together, the framework design has been improved to achieve better efficiency and resiliency. We also present a more detailed performance analysis that takes into account channel characteristics, temporal and spatial diversity that were neglected earlier. The rest of the paper is organized as follows. Section 2 describes motivation of the work and related works. Section 3 elaborates the proposed protocol framework and supporting techniques. Sections 4 and 5 present analytic and simulation results respectively and Section 6 draws the conclusion.

2 Background and Motivation

Our work on WSNs with transmit-only nodes was first motivated by intravehicular sensor network research, where we studied the possibility of replacing the hundreds of wired sensors in a regular vehicle with the wireless sensors that only have the UWB transmitters [14], [15]. It is also envisioned that in a modern assisted-living environment for aged people, many transmit-only sensor nodes will be deployed for low-priority monitoring tasks (e.g., windows, doors, temperature, motion, etc.) along with a certain number of standard sensor nodes for high-priority tasks (e.g., ECG, respiration rate, blood pressure, push-to-call device, etc.). Even though a facility may be covered by multiple dedicated data sinks, there are cases when a single sink will have to handle a large number of high-priority and low-priority nodes at the same time (e.g., when many people gather in a conference room), giving rise to the need to consider efficient and reliable management for this type of systems.

We formulated the basic problem as follows: there are a large number of Low-Priority (LP) transmit-only nodes, a relatively small number of High-Priority (HP) transmit-only nodes and one data sink in a single-hop, single channel WSN. After every T time units, a new data packet is sensed on each node, and the packet needs be delivered to the sink before the next interval of T units of time begins. We designed a QoS-aware, optimal retransmission-based MAC protocol, called QoMoR, where each node randomly transmits its packet an optimal number of times (i.e., the smallest number of transmissions to guarantee the statistical delivery probability requirement. Note that "transmission" and "retransmission" will be considered same in this work) within every T interval. An HP node may transmit more times than an LP node to achieve a higher required data delivery probability. We evaluated a range of settings through systematic analysis and extensive simulations, and provided formulas to guide the setup for different optimization problems.

The QoMoR protocol demonstrated a good data delivery performance in relative small scale systems (e.g., about 97 percent delivery probability in a 100-node WSN), but the performance is degraded when there are a larger number of nodes in the system. Moreover, in many cases, a data packet is delivered to the sink with fewer attempts than the precalculated optimal number of transmissions, which implies wastage in the channel capacity. Fig. 1 shows an example of QoMoR, where every node transmits exactly *five* times during each T. However, during interval T_i, for an HP node whose packet is successfully delivered to the sink after *two* transmissions, the remaining *three* transmissions are not only useless, but also can cause collisions to other transmissions as indicated by the red dot frames. Motivated by the above observation, the Asymmetric QoMoR (A-QoMoR) protocol was then proposed, which requires each HP node to use a rudimentary receiver to receive simple acknowledgement from the sink. The basic idea of A-QoMoR is that the HP node can stop transmitting after its previous transmission is acknowledged as illustrated in the middle row of Fig. 1 during interval T_i. Due to fewer collisions of transmissions by the HP nodes, a better data delivery performance could be achieved by all the HP and LP nodes, and more energy will be saved on the HP nodes. We have demonstrated the effectiveness and benefits of this A-QoMoR approach in [18].

Exploring the idea of A-QoMoR further, we note that since both HP and LP nodes transmit randomly, it is quite possible that all the attempts of the HP nodes will fail even though there exists a plenty of vacant time space among the transmissions by all the LP nodes as shown in the middle row of Fig. 1 during interval $T_i + 1$. If the HP nodes transmit among the vacant time space, they could avoid failures due to collisions with other LP and HP nodes, and hence the unnecessary multiple transmissions. At the same time, the LP nodes will suffer fewer collisions caused by the HP data transmission and have a better chance to deliver their own packets in time. This motivates us to find a way to estimate the vacant time space and design a collision-free scheduling scheme for the HP nodes as depicted at the bottom row of Fig. 1.

There have been a few related works that used transmit-only nodes for sensing applications [19], [20], [21] and [22]. In particular, the work in [19] analyzed a MAC scheme for Wireless BAN using transmit-only UWB devices. Each device runs on different UWB pulses/bit, similar to using different number of retransmission. However, the approach may suffer from the resulting correlation problem, and the work did not address how to choose proper number of samples per interval as we did in QoMoR analysis. A recent work in [20] used low-frequency transmit-only nodes for long range agricultural sensing applications. The work evaluated the path loss effect as well as the link quality. It also demonstrated the performance in a small scale setting but did not present a concrete protocol design. There are also a few early works that discussed how to solve the signal reception problem on the lower layer when the transmit-only nodes exist in the network [21], [22]. Besides, prior works have looked at the reliability data transmission, but those are either rely on multichannel resource and capability [13], [23], [24] or various TDMA schemes that require time synchronization scheme [25], [26], [27], which is not applicable to our case. Our approach, however, excels in its simplicity since it does not require extra hardware capability or suffer excessive communication overhead.

3 The RARE Framework Design

This section defines the problem and presents the design details of the proposed RARE framework for the hybrid WSN cluster architecture.

3.1 Problem Definition

We consider a single-hop wireless sensor network cluster consisting of a large number (n) of low-priority category nodes, a certain number (m) of high-priority category nodes and one data sink. The nodes in both categories have their minimal QoS requirement at the sink (P^{LP} and P^{HP}). Each node generates a new data packet every T units of time (which corresponds to the data generation interval), and attempts to transmit the packet to the sink before next data is generated. The packet transmission duration of each category, T_{LP} and T_{HP}, is assumed to be the same, and $(T_{LP} = T_{HP}) \ll T$. In this paper, we propose to use standard nodes with receivers as HP nodes and transmit-only nodes as LP nodes to form a hybrid WSN cluster.

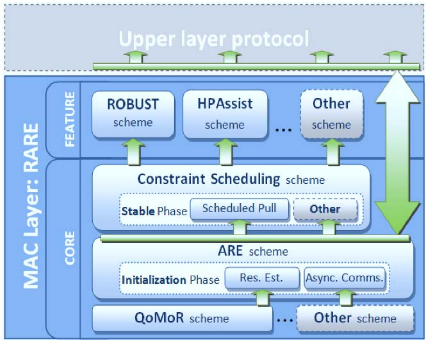


Fig. 2. RARE protocol framework.

The core problem to be addressed is to: 1) find the optimal number of transmission x for LP category nodes, such that if each LP node retransmits its data packet x times in every T, it can achieve a data delivery probability of $\mathrm{P}^{\mathrm{LP}}_{sink}$ at the sink, where $\mathrm{P}^{\mathrm{LP}}_{sink} \geq P^{LP}$; 2) manage the data transmission of HP category nodes to maintain its data delivery probability such that $\mathrm{P}^{\mathrm{HP}}_{sink} \geq P^{HP}$.

Moreover, we want to achieve the following improvements over the previous QoMoR work:

1. Improved data delivery performance or lower energy consumption for LP nodes.
2. Maximized data delivery performance and optimized energy consumption for HP nodes.
3. Reliable data delivery performance against various interference and environmental effects for HP node.
4. Dynamic network change support for node addition and node removal.

3.2 Overview of RARE

Robust Asynchronous Resource Estimation (RARE) is a MAC layer protocol framework designed to manage the operation of the low cost, low power and densely deployed hybrid WSN within a single-hop communication range. As shown in Fig. 2, the RARE framework consists of *CORE* stack and optional *FEATURE* stack. It manages the transmit-only nodes and standard nodes in different ways: transmit-only LP nodes access the channel randomly and will respond to no sink control as designed in underlying QoMoR, while channel access of the standard HP nodes is managed by the sink. From the perspective of operation phase, the RARE framework consists of *two* phases, the initialization phase and the stable phase. The involved data and control packet structures are shown in Fig. 3. The following sections will describe its components in more details.

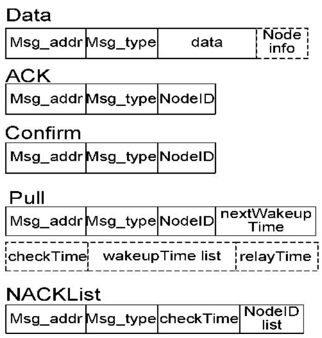


Fig. 3. Packet structure (dot frame is optional).

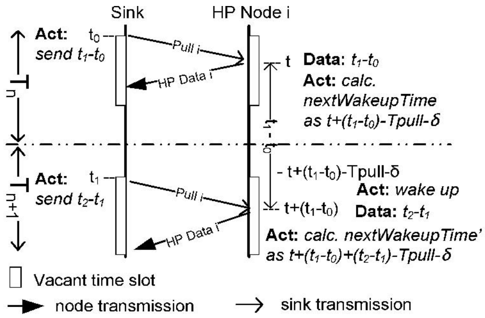


Fig. 4. Flow of ARE ("Data": received content from previous communication; "Act": action to perform).

3.3 Core Stack—The QoMoR Scheme

The QoMoR scheme is the fundamental component in the core stack and governs the random transmission. The optimal number of transmissions is precalculated based on the system requirement (e.g., data delivery probability, transmission rate, packet size, etc.) [14], [15] and programmed to the nodes before the deployment. In order to minimize the power consumption, the LP nodes only wake up before transmitting the sensed data and remain in the SLEEP mode in the rest of the time. Moreover, to support the operation of upper layer schemes, the LP nodes will attach the node information to their data packets for certain duration (see Sections 3.4.1 and 3.4.3). Note that the QoMoR scheme used in this work is replaceable by other schemes as long as they support similar transmission prediction discussed the next section.

3.4 Core Stack—The Asynchronous Resource Estimation (ARE) Scheme

The ARE scheme is another fundamental component in the core stack and serves as an abstraction layer of the underlying components. It defines the asynchronous communication and supports resource estimation that is performed during the initialization phase and used by the scheduling scheme.

3.4.1 Resource Estimation

We define the vacant time slots as the resource of the proposed RARE framework. The sink estimates the vacant time space and generates the vacant time slots for the entire operation via the ARE scheme.

In the ARE scheme, collision-free scheduling is achieved through LP data transmission estimation. This is because in QoMoR each LP node uses a *Pseudo random Number Generator* (PRNG) with a distinct seed to pick its random transmission time. If LP nodes can successfully share their seeds and other necessary node information (such as node ID, node type, valid generation range, etc.) to the sink, the sink can use the same PRNG strategy (e.g., how to use collected node information to generate valid random numbers for data transmission event time) to generate the same sequence of random time as those generated by each LP node. With this simple yet powerful property, the sink can build the complete transmission schedule of all the LP nodes as soon as it has the necessary node information. Hence, the sink is able to generate the vacant time slots out of the vacant time space in between the estimated transmissions of each interval for different purpose. Note that in practice, the sink only needs to generate the slots for up to *two* intervals at the beginning of each T and the slot size is currently fixed to the size that can fit *one* data request from the sink and *one* data reply from the node. Besides, at the time of the initial deployment, the sink only knows the PRNG strategy.

3.4.2 Asynchronous Communication

Another key idea of the ARE scheme is asynchronous communication. In order to conserve energy, all the HP nodes are designed to stay in the SLEEP mode whenever possible. Therefore, to schedule HP data transmission within the vacant time slots, the sink needs to inform the HP nodes when to wake up exactly (or a little earlier to ensure the reception) for the control packets reception or data transmission. However, time synchronization scheme will cause extra communication overhead (which will lead to extra energy consumption to the involved parties) that will drain the power of the nodes faster. The ARE scheme provides an asynchronous approach between the sink and the nodes (we assume there is no clock drifting on the nodes in this work). Specifically, as showed in Fig. 4, where "Data" label shows the received data content from previous communication and "Act" label indicates action to perform (same labels are used in other figures), instead of sending the exact time value t_1 (when the sink plans to send next *Pull* command as data request), the sink sends the time delay value $(t_1 - t_0)$ (the time difference between the current and next *Pull* command transmission). Upon receiving the delay value, the HP node calculates the time instance when the sink tries to deliver the next *Pull* command as $(t + t_1 - t_0 - T_{Pull})$. After transmitting the sensed data during the current interval, the HP node goes back to SLEEP mode and remains in that mode for $(t + t_1 - t_0 - T_{Pull})$. Note that although the processing delay is ignored in this work, there is signal propagation delay between the sink and the nodes. In practice, we use the largest delay value δ as the guard band to make sure the HP node wakes up for the *Pull* command earlier than the expected time. Therefore, at $(t + t_1 - t_0 - T_{Pull} - \delta)$, which is during the next interval, the HP node will wake up to receive the new *Pull* command that contains the necessary information to proceed. All time related communication in the proposed framework works this way and as a result, the ARE scheme is able to provide more vacant time slot resource without the time synchronization overhead.

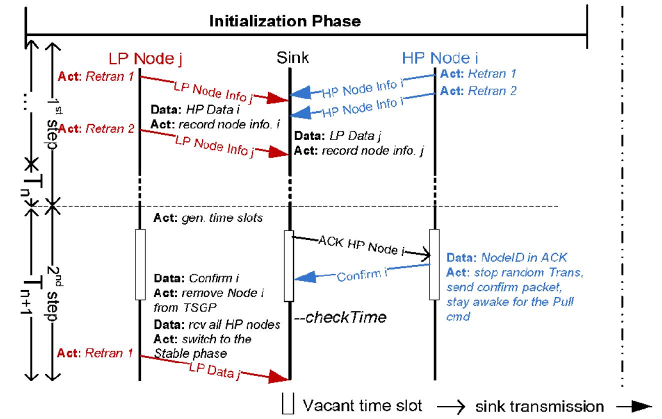


Fig. 5. Data flow of the initialization phase ("Data": received data content from previous communication; "Act": action to perform).

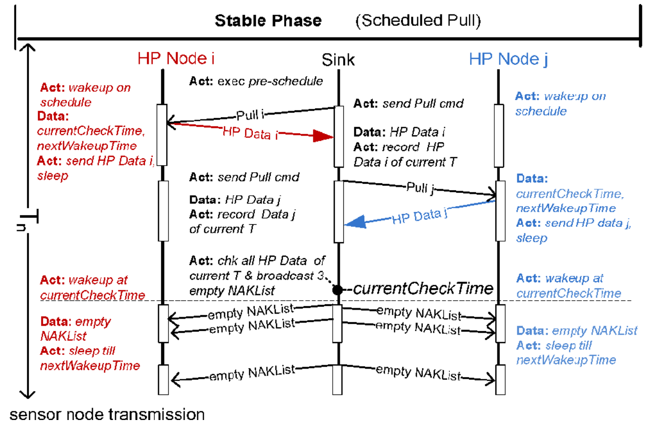


Fig. 6. Data flow of the stable phase ("Data": received data content from previous communication; "Act": action to perform)

3.4.3 The Initialization Phase

The initialization phase is a preparation phase. During this phase, the sink will collect node information from both LP and HP nodes to prepare for the scheduling and data collection in the stable phase. As shown in Fig. 5, at the beginning of this phase, all the nodes randomly transmit data packets that contain their node information to the sink optimal number of times. The sink records new node information received from all the nodes during the process. The initialization phase is further divided into *two* steps. The first step lasts until all existing nodes in range successfully deliver at least *one* data packet to the sink. The duration of this step is estimated based on the system requirement and programmed to the nodes before the deployment.

The following T interval after the first step is the second step, in which the sink not only begins to execute the *Time Slot Generation Procedure* (TSGP) that estimates the vacant time slots based on received node information (including both LP and HP nodes), but also dedicates the whole T interval to acknowledge the received HP nodes in order. This step is designed to last only *one* T interval.

Specifically in the second step as shown in Fig. 5, the sink picks a next available vacant time slot for *ACK* and *Confirm* transmission (the slot is large enough to fit both of the packets) for each HP node whose node information is received by the sink. The start time of these picked slots is the scheduled time when the sink sends an *ACK* packet containing the *NodeID* to each of these HP nodes. Besides, the sink picks another vacant time slot for acknowledgement check purpose. The start time of this slot is called "*checkTime*," at which the sink checks whether all the HP nodes have been acknowledged. On the HP node side, upon receiving the *ACK*, the HP node will stop random transmission, send a *Confirm* packet back to the sink immediately and stay AWAKE until it receives its first *Pull* command from the sink. If the sink receives the *Confirm* packet, it then excludes that HP node from the TSGP. The second step ends at the *checkTime* when the sink receives *Confirm* packet from all the HP nodes. Otherwise the sink will start above process on the unacknowledged HP nodes all over again until it receives all the confirmation or there are no more vacant time slots in the second step.

The initialization phase ends with the second step. In case some node failed delivering any of its data packets to the sink during the initialization phase or some HP node failed getting acknowledged during the second step of the initialization phase, it will be handled as a new node that tries to join the network in the stable phase.

3.5 Core Stack—The Constrained Scheduling Scheme

The constrained scheduling scheme is the major component of the core stack, through which the sink manages all communications of the HP nodes. The enhancement components in the upper feature stack also rely on this scheme to schedule different operations within the vacant time slots during each interval. From the perspective of the operation phase, this scheme operates in the stable phase, and in this paper the sink works through the Scheduled Pull approach.

3.5.1 The Stable Phase

The stable phase starts from the next interval immediately after the initialization phase. It is the main working phase, during which the sink schedules the transmission of all the HP nodes and other possible enhancement components during each interval using the vacant time slots and the HP node information obtained in the initialization phase. The HP nodes will follow the optimal transmission schedule generated by the sink while the LP nodes still perform optimal random transmission within each T. Note that different application scenarios may result in different scheduling schemes. Below we describe the Scheduled Pull approach used by this work.

3.5.2 The Scheduled Pull Approach

In this approach, the sink sends *Pull* commands to and receives data packets from each HP node during the vacant time slots. Different from conventional pull style approaches, this approach eliminates energy waste on channel sensing and idle listening through the *Preschedule* process. Basically, as demonstrated in Fig. 6, this approach contains *three* steps: scheduling; transmission and reception check; and additional rounds of transmission and reception check if necessary. To support the operations, the sink maintains *two* sets of fields for **every** HP node: *currentTransTime* (transmission time in the current interval) and *nextWakeupTime* (wakeup time in the next interval), as well as *currentCheckTime* (time when the sink checks HP data reception in the current interval), while each HP node maintains a set of local fields: *checkTime* (time when the sink may send the reception acknowledgement in the current interval) and *wakeupTime* (wakeup time of this HP node in the next interval). Note that these time values are also the start time of the picked vacant time slots maintained by the sink.

During every interval, there are *three* steps on the sink side: **Step 1**: The sink first calculates the optimal transmission schedule for all HP nodes for the next interval in advance using the *Preschedule* algorithm described in Algorithm 1. Specifically, at the beginning of each interval, the sink generates the vacant time slots of the next interval via TSGP. Then for **every** HP node, the sink replaces the *currentTransTime* with the *nextWakeupTime*, and picks a new *nextWakeupTime* from the vacant time slot array for next T interval, as well as a new *currentCheckTime* for current T interval.

Alg. 1 PRE-SCHEDULE IN THE STABLE PHASE

```
Sink runs following operations at the beginning of each T:
Set requestDuration to that of Pull cmd & HP data as:
  requestDuration = T_Pull + δ + T_HP + δ;
for each (node in HP node collection)
  get slotIndex in the nextSlotArray[];
  if such a slot exists in the nextSlotArray[] with slotIndex,
    Set node.nextWakeupTime = nextSlotArray[slotIndex].start;
    Update .start and .duration of nextSlotArray[slotIndex];
  end if
end for each
get slotIndex' in the nextSlotArray[];
Update .start and .duration of nextSlotArray[slotIndex'];
for each (node in HP node collection)
   Set node.checkTime = nextSlotArray[slotIndex].start;
end for each
```

Note that in the first interval of the stable phase, the sink does not have *currentTransTime* for any of the HP nodes and the HP nodes do not have *wakeupTime* either. Therefore, to start the stable phase, the sink will execute the *Pre-Schedule* algorithm twice during the first interval: first using *currentSlotArray []* to get transmission schedule for current interval and then using *nextSlotArray []* for the next interval. The HP nodes, at the same time, need to stay in AWAKE before receiving the first *Pull* command (as mentioned in the second step of the initialization phase), after which the sink and nodes will obtain the necessary information to proceed;

Step 2: The sink sends out the *Pull* commands (including the new *nextWakeupTime* and *currentCheckTime*) to all the HP nodes one by one at each of the corresponding *currentTransTime*, and receives the HP data right away. At *currentCheckTime*, if the data from all HP nodes is received, the sink broadcasts *three* (from experimental results to achieve robustness) consecutive empty *NAKList* packets indicating the end of the HP data reception for the current interval. Otherwise, the sink broadcasts a *NAKList* (including all the un-received HP *NodeIDs* and *currentCheckTime′*) to notify the HP nodes of the data reception results. This serves as the first round of data reception check and from the perspective of communications involved, this step mainly consists of {*Pull-HP Data*} pairs and finally the single or *three* *NAKList* packets as shown in Fig. 6.

Step 3: In case there is missing HP data, the sink starts following rounds of reception check to minimize the impact when certain important packets are lost. Specifically, during the second round of check, the sink schedules *two* vacant time slots for each missing HP node, *one* for the *Pull* command as well as HP data reply and another for the *NAKList*. In addition, the sink schedules a single *currentCheckTime″* for all missing HP nodes of that round. Then at *currentCheckTime′*, the sink sends, to each of the missing HP node, the *Pull* command including the corresponding *nextWakeupTime* and the same *currentCheckTime″*, receives the HP data, and sends *one NAKList* as scheduled. At the *currentCheckTime″*, the sink will check the data reception and responds accordingly as does in the Step 2 to end the check round. The rounds of reception check continue until all the missing HP data is received or all the remaining vacant time slots of current interval are used up. The difference of these reception check rounds from the first one is that they consist of {*Pull-HP Data-NAKList*} communication pairs instead.

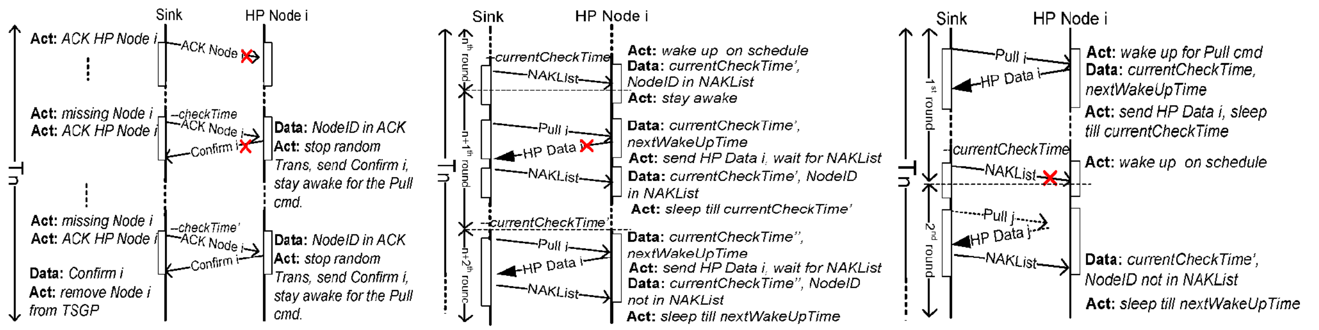
Correspondingly on the HP nodes side, if a *Pull* command is received, the HP node calculates the expected *nextWakeupTime* and *currentCheckTime*, updates its local *wakeupTime* and *chekTime* with the newly received values and sets up its timer accordingly. Then the node transmits its data and stays awake expecting the *NAKList*. If the first received *NAKList* is empty or the matching *NodeID* is not found, which means the data is received, the HP node will go to SLEEP until its *nextWakeupTime* in the next T. Otherwise, the HP node will set a wakeup timer using the just updated local *checkTime*, or the same *currentCheckTime* extracted from either a *NAKList* or a *Pull* command before going to SLEEP. At the *checkTime*, the HP node will wake up for the next round of reception check. The same process is followed in the rest intervals.

3.6 Feature Stack—The ROBUST Scheme

As mentioned in Section 1, external signal interference, environmental effects and scheduled network adjustment or unexpected node failure are the main causes that lead to the wireless communication failure. The ROBUST scheme is an enhancement component on the *FEATURE* stack to address these issues and to guarantee the nodes of high-priority category will still meet high-performance requirement. Basically, it utilizes the vacant time slot resource in each interval and redundancy in the packet structure to handle possible packet loss and support dynamic network change. Note that the basic supporting operations of the ROBUST scheme and necessary data redundancy in the packet structures have already been built into the *CORE* stack components, such as the second step in the initialization phase and the third step of scheduled pull approach in the stable phase. Below, we will focus on how the framework will handle the packet loss and network change.

3.6.1 Handling Packet Loss

Fig. 7 shows some possible scenarios of packet loss in the initialization phase and stable phase. Fig. 7a shows that in the initialization phase, if *ACK* or the corresponding *Confirm* packet is damaged, the sink will try to acknowledge the HP node with rounds of acknowledgment check



(a) *ACK/Confirm* is lost (b) HP Data in the later reception check rounds is lost (c) Non-empty *NAKList* in the first reception check round is lost

Fig. 7. Scenarios of packet loss handling in the initialization phase and stable phase ("Data": received data content from previous communication; "Act": action to perform).

until the end of the second step as described in Section 3.4.3. Note that once the initialization phase ends, the sink will exclude all the un-received or unacknowledged HP nodes from scheduling in the stable phase operation and handle them as node addition.

During the stable phase, there are *three* types of packets: *Pull* command, Data, *NAKList*. Depending on the time of the packet failure, there could be *seven* different scenarios, among which *two* of them are shown in Fig. 7 and the rest is detailed in the supplementary file, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2012.148.

The first is illustrated in Fig. 7b, wherein the HP data in the later rounds other than the first round is assumed to be lost. Subsequently, the sink will send the *NAKList* containing the missing *NodeID*. The HP node then extracts the *currentCheckTime′* from the *NAKList* and sleeps until that time to try to receive the *Pull* command in the next round of reception check. The second scenario is illustrated in Fig. 7c, wherein a nonempty *NAKList* is lost. Then, any following *NAKList* will acknowledge the involved HP nodes and then the operation continues. However, these HP nodes will have to stay AWAKE after the first *currentCheckTime* until they receive another *NAKList*. Other multiple packet failures could be categorized into one of the *seven* cases and handled by the framework properly. Moreover, necessary redundancy fields are already built into the packet structures to ensure reliable and energy efficient operation on the HP nodes.

3.6.2 Network Adjustment and Change

Network adjustment and change usually include node removal and addition. On one hand, due to the collision, signal fading, external interferences, hardware/battery failure or manual removal, certain nodes may not be able to communicate with the sink. These nodes will be marked as inactive on the sink when there is no successful data communication for certain duration (e.g., *three* consecutive T intervals). 1) If an LP node fails to send its data for certain duration, the sink will consider the node to be down, remove it from TSGP, but still keep updating the corresponding random seed state in case the communication is only temporarily disturbed by the interferences. 2) If it is an HP node, the sink will not schedule the transmission for it in the later intervals. Accordingly, on the node side, when an HP node does not receive the *ACK*, *Pull* command or *NAKList* packet for it for certain duration, it will assume the sink already excluded it from the system and will behave as a newly deployed node again in order to enter the network.

On the other hand, node addition usually has *two* scenarios: new nodes are manually added into the system to adjust the sensing coverage or density; certain inactive nodes start operation again. The ROBUST scheme is able to automatically incorporate the new nodes or inactive nodes into the system.

More specifically, 3) when the sink receives the data packet from an LP node containing the node information, it knows the node is an inactive node or new one based on previously collected information. The sink will then either replace the existing node information or add the new one to the TSGP. Hence from the next T, data packets from this LP node will be collected and will not affect the schedules any more. However, when no node information found, the sink will recognize whether the data packet is from an active or an inactive LP node (due to temporary interference). If the data packet is from the former, the sink simply records it, but if from the latter, the sink will mark the node active and include it back into TSGP. Since the seed state on the sink is updated properly (as in case (1) in Section 3.6.2), all future transmission events on the sink and the LP node are still synchronized. 4) When a new HP node is added to the system after the deployment, it operates as an LP node initially. After the sink receives the HP data packet, it first adds the node to the HP node list and the TSGP. Then the sink acknowledges the new HP node in the next available time slot. For smooth operation, the *Pull* command is used in the stable phase instead *of ACK-Confirm* pair as seen in the initialization phase. When the new HP node receives the first *Pull* command, it will stop the random transmission and begin operating as other HP nodes. Once the sink receives the data from the same HP node right after the *Pull* command, it will remove this node from TSGP, and the vacant time space previously occupied by this HP node will be reclaimed for other operation.

3.7 Feature Stack—The HPAssist Scheme

The HPAssist scheme is another enhancement that works on the *FEATURE* stack. The goal of this scheme is to improve the data delivery capability of the LP nodes with the assistance of selected HP nodes. From the simulation results, we observed that the total energy consumption of

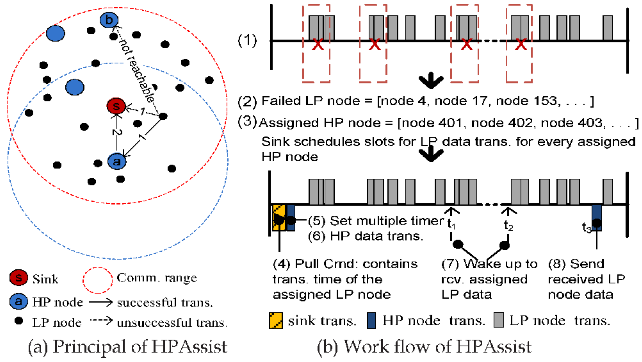


Fig. 8. Description of the HPAssist scheme.

the HP nodes over a certain period of operation time is far less than that of the LP nodes. This is essentially good because the constrained scheduling scheme lowers the HP energy consumption than the previous QoMoR does. However, in practical scenarios, power depletion of the LP nodes and the HP nodes may not occur around the same time. If the system is serviced whenever needed, extra maintenance will be required and this actually increases the overall maintenance cost of the system. Therefore, we designed this HPAssist scheme to improve system performance metrics with the tradeoff of the extra energy consumption on the selected HP nodes. In addition, this scheme tries to explore the spatial and temporal diversity of the wireless communication. The intent behind this scheme is shown in Fig. 8a. When transmission 1 fails at the sink, it may succeed at a different location. If the data is received by the HP node a in the range, it could be successfully relayed to the sink at a different time.

Specifically, as shown in Fig. 8b:

1. the sink predicts the possible failed LP data transmission by checking the overlap of all the transmissions during current T, and
2. identifies the possible failed LP nodes (that might not have at least *one* data packet delivered during the interval);
3. then the sink tries to assign each of these failed LP nodes to the existing active HP nodes. A vacant time slot at the end of the interval is also scheduled for each assigned HP node to send the received LP data to the sink (we call this relay assignment);
4. before sending the *Pull* command, the sink attaches the data relay time (e.g., t_3) and transmission time of the assigned LP node (e.g., t_1, t_2);
5. upon receiving the *Pull* command, the assigned HP nodes will set timer to wake up at the received time values accordingly;
6. HP data transmission;
7. if the LP data at one of the scheduled time (e.g., t_1) is received, the HP node will skip the rest wake up time (e.g., t_2); and
8. at the data relay time (e.g., t_3), the assigned HP node will wake up and send the data to the sink.

If no LP data is received, the HP node will just remain in SLEEP mode until the next interval.

TABLE 1
Symbol Used in Analysis

Symbol	Description
P_{rcv}/P_{tran}	Power consumption of reception/ transmission
$T_{Pull}/T_{HP}/T_{LP}/T_{NAKList}$	Transmission duration of the Pull command/HP data/LP data/ *NAKList* packet
n/m	Number of the LP/HP nodes
$P^{HP}_{sink}/P^{LP}_{sink}/P^{LP}_{HP}$	Successful delivery probability of the HP/LP node on the sink/HP node
Retran	Number of retransmissions
$Retran_0$	Number of retransmission to achieve the highest delivery probability
N	Number of total intervals in the stable phase
α	Probability of transmission errors due to reasons other than collision (e.g. fading, shadowing, pathloss, etc.)

Note that the system estimates the energy consumption of HP nodes and LP nodes in the initialization phase, and it will execute this optional scheme only when there is sufficient energy on the HP nodes. In addition, we do not assume the system has the knowledge of the node deployment in its communication range. Therefore, in order to balance the energy consumption among the HP nodes, the sink tracks the assignment times of each HP node and always randomly picks the HP nodes from those have the lowest time of assignment. Moreover, the actual assignment in the step (3) will depend on the specific application. For example, the sink might use a best effort approach and assign multiple HP nodes to each failed LP node to improve the delivery chance when the HP energy is sufficient. In order to understand the effectiveness of this scheme, we will evaluate the improvement in the overall data delivery probability in Section 5.2, and the ratio of the total LP data packets that are successfully relayed to the sink by the HP nodes to the total relay attempts, called the hit ratio hereafter, in Section 5.6.

3.8 Cross-Layer Protocol Design Support

Last but not least is the support for the cross-layer protocol design for upper layer protocols. Take multicluster communication for example, due to the random nature of the LP node transmission, any careless communication between the hybrid WSN clusters could be damaged and affect existing transmissions too. All upper layer protocols should utilize the vacant time slot information from the MAC layer for more efficient and reliable communications, as well as better operation strategies.

4 Theoretical Analysis of RARE

In this section, we analyze the performance of the proposed RARE protocol framework with respect to the data delivery probability, system capacity, initialization phase duration and energy consumption. We define common symbols for analysis in Table 1.

4.1 Data Delivery Probability

Data delivery probability is a major performance metric in our work due to the existence of the transmit-only nodes.

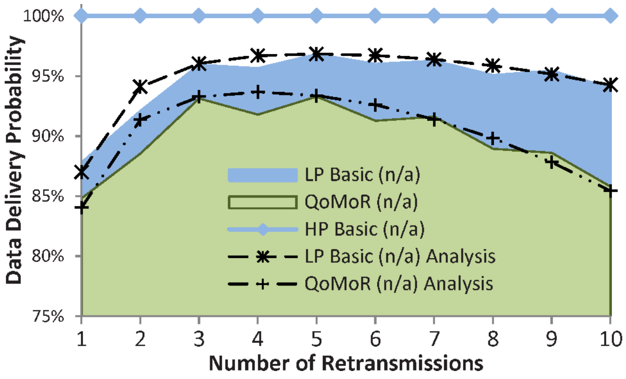


Fig. 9. Comparison of RARE and QoMoR in ideal environment.

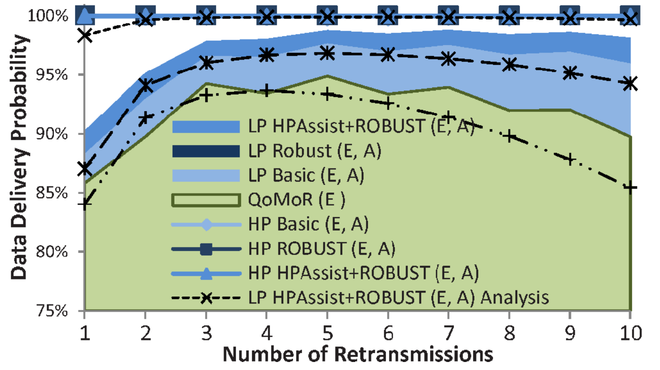


Fig. 10. Comparison of RARE and QoMoR with environmental effects (E: environmental effects, A: node addition).

We define the data delivery probability as the *average* data delivery probability achieved among all the nodes over the entire operation duration in the stable phase. Note that in a practical system, the data delivery probability is one of requirements to determine other parameters. In order to demonstrate the capability of the hybrid WSN architecture and the proposed framework, we focus on the maximum data delivery probability the system is able to achieve, and analyze the following *three* different settings under the RARE framework: 1) when using only the basic scheduling scheme; 2) when using the optional ROBUST scheme; and 3) when using both optional ROBUST and HPAssist scheme. And we will compare the results in Section 5.2 (Figs. 9 and 10).

4.1.1 Basic Scheduling Analysis

Ideally, the data delivery probability of the HP nodes will be 100 percent as a result of the constrained scheduling scheme. On the other hand, that of the LP nodes can be modeled the same as in [15]:

$$P_{sink}^{HP} = 100\%, \tag{1}$$

$$P_{sink}^{LP} = 1 - [1 - e^{\frac{-2 \cdot Retran \cdot (n-1) \cdot T_{LP}}{T}} \cdot (1-\alpha)]^{Retran.} \tag{2}$$

4.1.2 The ROBUST Scheme

In a harsh deployment environment or when a network undergoes some change, wireless communication could be suffering from interferences. On one hand, the ROBUST scheme can effectively minimize the negative environmental effects via strategic retransmissions and still ensure the 100 percent HP data delivery probability with the abundant vacant time slot resource from the ARE scheme. On the other hand, since the probability is calculated over the entire operation duration, any HP node will eventually be incorporated into the system and no longer affect the LP data delivery, only the newly added or removed LP nodes will eventually affect overall LP delivery performance. Suppose there are i LP nodes added to and j LP nodes removed from the system, then the new LP data delivery probability will change to

$$P_{sink}^{LP} = 1 - \left[1 - e^{\frac{-2 \cdot Retran \cdot (n+i-j-1) \cdot T_{LP}}{T}} \cdot (1-\alpha)\right]^{Retran.} \tag{3}$$

4.1.3 The HPAssist Scheme

Under the HPAssist scheme, the LP data reception on either the sink or the assigned HP node will contribute to the data delivery probability. Ideally, the LP data delivery probability on the assigned HP node could be modeled same as in (2), where n' is the number of the LP nodes in the communication range of the assigned HP node

$$P_{HP}^{LP} = 1 - \left[1 - e^{\frac{-2 \cdot Retran \cdot (n'-1) \cdot T_{LP}}{T}} \cdot (1-\alpha)\right]^{Retran} \tag{4}$$

$$(0 \leq n' \leq n).$$

Assume the received LP data could always be relayed from the HP node to the sink in the same interval. The probability that the same LP transmission fails at both the sink and the assigned HP node is $(1 - P_{sink}^{LP}) \cdot (1 - P_{HP}^{LP})$. Therefore, the statistical delivery probability of the specific LP node could be modeled as

$$P^{LP} = 1 - \left[\left(1 - P_{sink}^{LP}\right) \cdot \left(1 - P_{HP}^{LP}\right)\right]. \tag{5}$$

Note that since we do not assume that the system has knowledge of the HP node deployment, the exact value of n' is usually unknown to the system. In this work, the target sensing area is within one hop, therefore n' is considered the same as n.

4.2 System Capacity and Slots Utilization

Thanks to the ARE scheme, there are sufficient vacant time slots available towards different uses. In order to get the comparable results with those in previous works, the slot is defined as the time duration of both *Pull* Command and HP data transmission (depending on the application requirement, slots can be defined to suit different purpose). This section analyzes the availability of the resource in terms of the maximum number of HP nodes the system can support if all the resource is used towards data collection. Besides, the current slot utilization is analyzed to examine the potential of the proposed framework as well as the optional enhancement schemes.

First, from the perspective of system design, the parameters will be chosen to achieve the required performance. The $Retran_0$ to get the highest $\mathrm{P}_{sink}^{\mathrm{LP}}$ can be derived as a function of n, T_{LP} and T by letting the first derivative of $\mathrm{P}_{sink}^{\mathrm{LP}}$ to *zero*. Only the values from *one* to $Retran_0$ are reasonable values of retransmission. Second, in the worst case in a T interval, if none of the LP node transmission overlaps and the gap between any *two* consecutive transmissions is just less than the time duration that could fit *one* pair of *Pull* command and HP Data transmission. Therefore, the maximum time the LP data transmissions could take from each T is

$$T_{max,LP} = (T_{LP} + T_{HP} + T_{Pull}) \cdot n \cdot Retran_0. \tag{6}$$

Hence ideally the minimum number of HP nodes that can be supported is

$$Num_{min,HP} \geq \frac{T - T_{max,LP}}{T_{HP} + T_{Pull}}. \tag{7}$$

The capability of the framework for other potential usages could be estimated similarly.

We now derive the upper bound on the slot utilization to further analyze the potentials of the RARE framework. In an ideal environment, where there is no fading, shadowing or pathloss effect, the vacant time slots are used for delivering HP data, performing operations for the ROBUST scheme and the HPAssist schemes. During each interval, the number of vacant time slots needed for HP data delivery equals to the total number of HP nodes, and the ROBUST scheme only takes *3* slots for its operations. Besides, the HPAssist consumes the same number of vacant time slots as the number of failed LP nodes within each T interval. Based on the successful data delivery probability in (2), we can safely estimate the number of possible failed LP nodes to be $\mathrm{n} \cdot (1 - \mathrm{P}^{\mathrm{LP}}_{sink})$. Hence the upper bound on the slot utilization when running both the ROBUST and HPAssist schemes in an ideal environment can be evaluated as

$$P_{util.} = \frac{m + n \cdot (1 - P^{LP}_{sink})}{Num_{\min,HP}}. \tag{8}$$

4.3 Initialization Phase Duration

The initialization phase is an important step to RARE since the actual duration of the phase determines the usability of the framework. Let T_{init1} be the time duration that all the nodes in the network have delivered at least *one* packet to the sink. After this duration, each node will have transmitted its packet $Retran \cdot \mathrm{T}_{init1}/T$ times. Hence, the probability of at least *one* copy of the node information will be received by the sink after T_{init1} becomes

$$P^{LP}_{sink} = 1 - \left[1 - e^{\frac{-2 \cdot Retran \cdot (n-1) \cdot T_{LP}}{T}} \cdot (1 - \alpha)\right]^{\frac{\mathrm{T}_{init1} \cdot Retran}{T}}. \tag{9}$$

It is theoretically impossible to guarantee with 100 percent probability that all nodes have delivered their packet to the sink within a given time. So we approximate $\mathrm{T}_{\mathrm{init1}}$ to be the time taken for each node to achieve a delivery probability of at least 0.9999. As long as *Retran* is determined, the upper bound of the $\mathrm{T}_{\mathrm{init1}}$ can be calculated. Moreover, the sink will spend at most *one* T interval to acknowledge the received HP nodes in the second step, therefore the upper bound of the initialization phase,$\mathrm{T}_{\mathrm{init}}$, could be modeled as

$$T_{init} = T_{init1} + T. \tag{10}$$

Normally, the phase duration is so short that there will be no manual network adjustment or node failure that could affect the duration.

4.4 Energy Consumption

Energy consumption is a measurement of the protocol efficiency. Since the duration of the initialization phase and idle listening time before receiving the first *Pull* command are usually insignificant compared to that of the entire operation, this section only analyzes the energy consumption from the second T interval in the stable phase operation: 1) with only basic scheduling scheme; 2) when the optional ROBUST scheme is taking effect; and 3) when the optional HPAssist scheme is taking effect.

4.4.1 Basic Scheduling Analysis

During the basic operation in the stable phase, the LP nodes will randomly transmit data packets *Retran* times. Therefore, if there are N intervals in the stable phase, the energy consumption on the LP nodes will be

$$E_{LP} = P_{tran} \cdot T_{LP} \cdot Retran \cdot n \cdot N. \tag{11}$$

Recall that in the Scheduled Pull approach, the HP nodes initially stay in AWAKE during the first T before receiving the first *Pull* command from the sink. Compared to the entire operation duration, this energy consumption is also insignificant. Then during the rest of the time, the HP nodes only receive *one Pull* command and *one NAKList* from the sink as well as transmit *one* data packet to the sink. Thus, we can get the typical energy consumption as following:

$$E_{HP} = (P_{tran} \cdot T_{HP} + P_{rcv} \cdot T_{Pull} + P_{rcv} \cdot T_{NAKList}) \cdot m \cdot N. \tag{12}$$

The total energy consumption in the basic scenario can be simply modeled as

$$E = E_{LP} + E_{HP}. \tag{13}$$

Therefore, from (12), we know that RARE consumes less energy than QoMoR as long as *Retran* satisfies following inequality: $Retran \geq 1 + P_{rcv} \cdot (\mathrm{T_{Pull}} + \mathrm{T_{NAKList}})/P_{\mathrm{tran}} \cdot T_{HP}$.

4.4.2 The ROBUST Scheme

The ROBUST scheme requires additional energy to provide reliable data delivery for the HP nodes. Suppose there are x LP nodes and y HP nodes added to the system and they randomly transmit k times in each interval, each of their transmissions could cause other HP transmission failure and extra energy to recover the failure. Since the LP nodes always transmit an optimal number of times every interval, this section only discusses the extra energy consumption on the HP nodes.

From the perspective of failure recovery from individual packet loss, as discussed in Section 3.6 and the supplementary file, which is available online, in order to restore the operation from these scenarios, the ROBUST scheme will require extra energy consumption. The energy consumption of the *two* scenarios that are shown in Fig. 7 is

$$E_1 = P_{rcv} \cdot (T_1 + T_{Pull} + T_{NAKList}) + P_{tran} \cdot T_{HP}, \tag{14}$$

$$E_2 = P_{rcv} \cdot (T_{Pull} + T_{HP} + T_{NAKList} + T_2) \cdot (p - 1). \tag{15}$$

Note that p is the number of current unreceived HP nodes; T_1 and T_2 are time duration the HP node has to remain in AWAKE before receiving next *Pull* command for it from the sink. The actual value might vary from 0 up to $(\mathrm{T_{Pull}} + \mathrm{T_{HP}} + \mathrm{T_{LP}}) \cdot Retran$ in the worst case, where $0 \leq Retran \leq Retran_0$.

From the perspective of failure recovery from the network adjustment and change, if each packet has the similar size, in the worst case, each transmission of the newly added nodes may damage *two* adjacent transmissions. According to the protocol operation and the failure scenarios, there are *three* valid cases: 1) HP data packet from node i and *Pull* command for node j are damaged; 2) *NAKList* and *Pull* command for node i are damaged; and 3) HP data packet from node i and *NAKList* are damaged. Among above, case (3) will require more energy consumption on the HP nodes. Next, we will derive the formula based on case (3) to obtain the energy consumption upper bound.

When the HP data and *NAKList* are both damaged, according to Fig. 7, the sink will be in the first reception check round. As analyzed above, when these *two* packets are damaged, the extra energy consumption will be $(E_1 + E_2)$. Hence, the maximum energy caused by node addition in the stable phase could be modeled as

$$E_{extra} = (x + y) \cdot k \cdot (E_1 + E_2). \tag{16}$$

4.4.3 The HPAssist Scheme

The HPAssist scheme utilizes the extra energy on the HP nodes to help improve the LP data delivery performance. As the protocol design, the assigned HP node will wake up at most *Retran* times to try to receive LP data and *one* time to relay the data during each interval. Suppose q is the number of potential failed LP nodes in the specific interval. Therefore, the extra energy of these HP nodes in the worst case can be calculated as

$$E_{HPAssist} = q \cdot (P_{rcv} \cdot T_{LP} \cdot Retran + P_{tran} \cdot T_{LP}). \tag{17}$$

5 Performance Study of RARE

In this section, we evaluate the performance of the proposed RARE framework against QoMoR protocol in terms of data delivery probability, system capacity, and energy consumption. Moreover, we demonstrate the effectiveness of the RARE framework and the proposed enhancement schemes by evaluating the initialization phase duration, vacant time slot utilization, and the hit ratio in HPAssist. Furthermore, we compare the proposed RARE with the ZigBee standard in details in terms of data delivery probability and energy consumption udner the comparable simulation setups in the supplementary file, which is available online. All results are obtained through extensive simulations using the Qualnet simulator [28].

5.1 Simulation Setup

The simulation involves 500 nodes that are uniformly placed in a flat grid of size $55\ \mathrm{m} \times 55\ \mathrm{m}$. These nodes are divided into *two* priority categories, where 400 nodes are LP nodes, 100 nodes are HP nodes. *One* data sink is located in the center of the sensing area. Among the LP nodes, 390 nodes are initially deployed and another 10 nodes will be added to the network at the following time (Seconds): 0.5, 1, 1.5, 7, 9.5, 10, 11, 12, 15, and 20. Since both types of nodes will behave as LP nodes when added and these addition times cover all the stages of the framework, this configuration is valid to demonstrate the performance of the proposed RARE framework under a dense WSN setting. Every 300 ms (T interval), a new data packet is generated by the node and it is expected by the sink before the start of the next interval. Each data packet measures 72 bytes and will be transmitted *Retran* times every T interval. The size of the *ACK* and *Confirm* packet is 48 bytes. The *Pull* command is 48 bytes and at most 120 bytes when the HPAssist scheme is activated, while that of the *NAKList* packet could vary from case to case and can go up to 120 bytes. The transmission rate is set to 11 Mbps and the power of transmission and reception is 660 mW and 350 mW respectively as to be consistent with our previous works. The experiments simulate for 30 s and results are averaged from 10 different global seed settings (global seed is used to generate other random seeds used for simulation in Qualnet [28]). Additionally, *TWO-RAY* path loss model is in effect in all experiments but *CONSTANT* shadowing model and *RICEAN* fading model are applied when simulating the real world deployment environment. Note that based on (2), the maximum LP data delivery probability is achieved when retransmission is set to 4. Hence only part (case 1 to 4 out of 10 cases) of the following graphs is more interesting to us.

5.2 Data Delivery Probability

To demonstrate the performance improvement of the proposed RARE framework, Fig. 9 first compares the maximum data delivery capability in an ideal deployment without environment effects using QoMoR as the baseline performance metric. While in QoMoR, the HP nodes and LP nodes directly compete for the channel access, basic scheduling in RARE (as "Basic" in the figures) is able to deliver much less dependent results between the HP and LP nodes because their transmissions are separated by the constrained scheduling scheme. As we can see from the graph, the HP nodes easily achieve 100 percent delivery probability for all retransmission cases and the data delivery probability of the LP nodes also shows a 4.47 percent improvement in average over QoMoR. Note that the data delivery probability is one of the most important performance metrics of WSN applications. Any small improvement means better capability and QoS differentiation potential.

In a more realistic deployment environment, Fig. 10 compares the data delivery performance among QoMoR, basic scheduling in RARE (as "Basic" in the figures), RARE with ROBUST scheme (as "ROBUST" in the figures) as well as RARE with ROBUST and HPAssist (as "HPAssist + ROBUST" in the figures). Clearly, when there exist environmental effects, wireless communication will suffer from fading and shadowing. First, QoMoR enjoys an interesting performance gain compared to that in Fig. 9 at around 2.15 percent, which is usually unexpected from the traditional belief that aforementioned environmental effects are always harmful. This is because in the multiple-random-transmission scheme like QoMoR, the failure of certain transmissions caused by the interference may assure the success of the other critical ones that could have been damaged. Second, in the "Basic" setting, the HP nodes no longer maintain 100 percent probability (averagely at

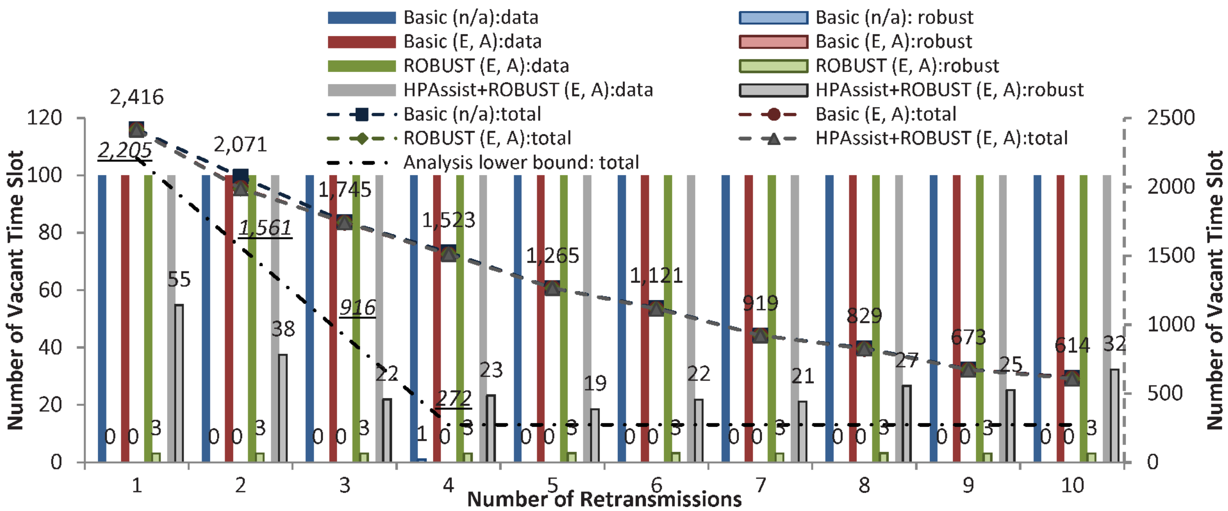


Fig. 11. Vacant time slot availability and usage breakdown (E: environmental effects, A: node addition).

99.94 percent in this case) due to the fading, shadowing, path loss effects as well as the unknown random transmission from the newly added nodes. All of above causes ruin the sink's HP data transmission schedule. Averagely, HP nodes show a 0.06 percent probability decrease, but LP nodes have 1.11 percent probability increase, which results a 3.70 percent improvement over QoMoR. Third, when enabling the ROBUST scheme, the system successfully mitigates the negative environmental effects and provides a reliable 100 percent delivery probability for the HP nodes again, while the LP nodes do not suffer any performance loss as of which could happen in QoMoR. This is achieved due to the fact that operation of the ROBUST scheme does not affect LP data transmission. Last but not the least, after enabling the HPAssist scheme, the system showed another performance boost of 1.63 percent on the LP data delivery probability that results a 5.34 percent improvement in average over QoMoR.

In terms of the data delivery probability, the proposed RARE framework delivers a significant performance improvement on both the high-priority category nodes and low-priority category nodes. Even when working in a harsh environment with unexpected network change, RARE is able to achieve the highest possible probability on the HP nodes and further boost the performance on the LP nodes, which is not achievable in previous QoMoR work. This improvement meets one of our design expectations.

5.3 System Capacity and Slots Usage

Fig. 11 depicts the average availability of the vacant time slots during every T interval and the usage break down of each aforementioned scheme (legend format is "setting (environment): category"). The figure captures the RARE framework running in *three* different settings ("Basic," "Robust," and "HPAssist+ROBUST") in both ideal and nonideal deployment environment. The "data" category indicates the average number of the slots dedicated for HP data transmission and the "robust" category indicates the slots used by the ROBUST scheme or both of the ROBUST and HPAssist scheme for performance improvement. The "total" category represents the total number of the vacant time slot.

First, the total number of vacant time slots indicates the maximum number of HP nodes that current system can support, and it decreases with the increase of the retransmission as expected. That is to say the simulation setup can easily support 400 LP nodes as well as up to 1,523 (when $Retran = 4$) HP nodes if necessary. The dotted line without marker represents the lower bound of the total vacant time slots during each T interval, which are more than 272 slots under current simulation settings. This demonstrates a promising system capacity when needed.

Second, Fig. 11 also shows the detailed vacant time slot usage by the proposed enhancement schemes. When the ROBUST scheme works to guarantee the HP data delivery, it costs slightly over *three* slots in average to recover the lost HP data packets, where *three* empty *NAKList* packets are the cause of the usage. When the HPAssist scheme is enabled to improve the LP data delivery probability at the same time, the sink needs to schedule extra slots for data relay from those assigned HP nodes. We noticed that the consumption of the slots in this case is inversely proportional to the LP data delivery probability. This is because when LP data delivery probability is low, more LP nodes will fail. Hence, more slots will be allocated by the sink to recover the LP data packets. In the simulation, the combination of the ROBUST and the HPAssist scheme averagely costs 19 slots at the lowest and 55 slots at the highest.

Third, HP data transmission will cost the same amount of the slots as the number of the active HP nodes. The slot utilization of the data and robust categories over the total category is shown in the Fig. 12, where the ratio is less than 10 percent for all reasonable retransmission cases. Moreover, the utilization upper bound plotted in Fig. 12 indicates that with the existing enhancement schemes such as the ROBUST and HPAssist scheme, the RARE framework uses only at most half of its resource. These results essentially demonstrate the great potential of the proposed hybrid WSN architecture and the RARE framework. Additional enhancements could be added to provide more useful features.

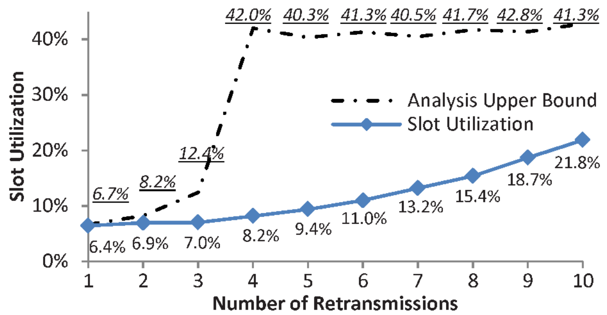


Fig. 12. Vacant time slot utilization.

5.4 Initialization Phase Duration

Fig. 13 shows the initialization phase duration and its analysis upper bound. We can see that the values relate to the LP data delivery probability. The higher the delivery probability translates to the shorter time duration of the preparation and it is statistically bounded by (10). The simulation results range from 0.7 s to 1.1 s for all reasonable cases in the ideal communication environment. When considering the environmental effects and network changes (node addition only), the initialization phase produces the similar results.

5.5 Energy Consumption

The design of the RARE framework ensures that it consumes less overall energy using basic scheduling scheme than previous QoMoR in common scenarios. Therefore, this section only evaluates the energy consumption of the enhancement schemes on the *FEATURE* stack. Fig. 14 shows the energy consumption on the HP nodes, where there are environmental effects and network change. First, compared to the basic scheduling of RARE, the ROBUST scheme consumes a little more when it comes to larger number of retransmission. This is because at larger number of retransmission, newly added LP nodes might cause more damage to the sink's HP data transmission schedule. Therefore, it will cause more operation to recover the loss. However, RARE has a good capability in handling the dynamic changes. Hence the extra operations are still kept to minimal. Second, operations of the HPAssist scheme consume the most energy among three. According to (17), most consumption comes from the attempted reception and it depends on the number of retransmission and the number of possible failed LP nodes. Even though it is relative high in the reasonable range (1 to 4 in this case), we are able to adjust the operations (as mentioned in

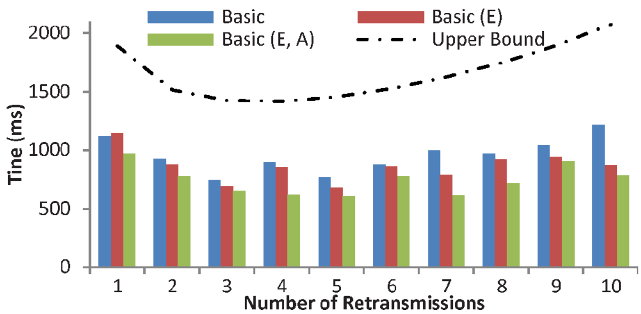


Fig. 13. Time duration of the initialization phase (E: environmental effects, A: node addition).

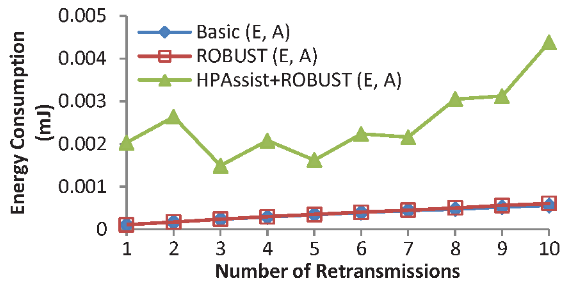


Fig. 14. Energy consumption on HP nodes (E: environmental effects, A: node addition).

Section 3.7) to suit the system requirement. For example, we can reduce or completely disable the operations of the HPAssist scheme when LP nodes only require lower data delivery probability.

5.6 Hit Ratio in HPAssist

In order to understand the effectiveness of the HPAssist scheme, we plot the hit ratio of the successful LP data relays to the total relay attempts. Since the number of attempts and successful relays vary from interval to interval, we average the hit ratio calculated from each T interval over the entire operation. The result shows a decent ratio ranging from 28.6 to 74.1 percent. The reason why it increases with the number of retransmission is that the assigned HP nodes simply wake up more times trying to receive the LP data and this potentially increases the chance of the data relay. Nevertheless, even though the increase in the data delivery probability of the LP nodes is a direct result of the HPAssist scheme, a higher hit ratio does not necessarily contribute to the large probability increase in Fig. 15. This is because in a nonideal environment, when some presumably failed LP nodes happened to successfully deliver the data packet to the sink, the HP data relay operations not only become unnecessary but also a waste of energy. Moreover, according to (4), the farther the picked assigned HP node is away from the sink (still in one hop), the lower the probability of collision with the nearby LP data communications will be. Hence the assigned HP node has a better chance to relay the received LP data to the sink eventually. Therefore, one can expect a higher hit ratio in a larger area deploying the single-hop hybrid WSN.

6 Conclusion

In this paper, we have studied the hybrid WSN cluster that contains both standard nodes and transmit-only nodes within a single-hop of the sink. This hybrid WSN cluster architecture finds good operation tradeoffs between the

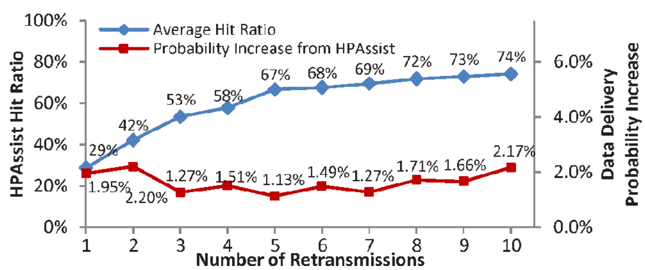


Fig. 15. HPAssist hit ratio versus probability increase.

traditional and transmit-only WSNs. It is more cost effective and energy efficient than traditional WSNs with only standard nodes. On the other hand, it is more flexible and better capable than the ones with just transmit-only nodes. To fully explore the potential of such hybrid WSN cluster architecture, we have proposed the RARE protocol framework to manage the hybrid system.

The system has been shown to have significantly improved data delivery capability in both HP and LP priority categories as well as the optimized energy consumption compared to the transmit-only WSNs. Furthermore, the hybrid WSN cluster is able to provide a reliable performance guarantee for the high-priority category nodes through the strategic retransmissions even in the harsh deployment environment (and existence of multiple LP node categories as demonstrated Section 2 in the supplementary file, which is available online) and always maintain a smooth operation when the system is facing either scheduled network adjustment or unexpected node failures. Moreover, the experiment results have also demonstrated the architectural benefits of exposing the underlying MAC layer resource information to other system components including upper layer protocols. There are *two* major limitations imposed by the transmit-only nodes, however: 1) the precalculated optimal number of retransmissions needs to be programmed to the devices before their deployment; and 2) significant clock drifting on the nodes may degrade the performance over time.

Our future work, besides addressing the above two issues, will further explore the potentials of this hybrid WSN cluster architecture in building a larger system with multicluster and multihop communications (based on sinks and/or HP standard nodes). In particular, we will evaluate the cost and energy saving potentials of our approach whereby routing is done in only a few hops even in a large system, and accordingly, a majority of nodes can be transmit-only nodes since they do not need to relay the data of other nodes, and compare our approach with other approaches, like GreenOrbs [29] whereby all nodes have full transceivers in order to serve as potential relays and support routing in multiple hops.

ACKNOWLEDGMENTS

The authors would like to thank Professor D. Koutsonikolas for his valuable editorial comments, Mr. Guan Tong for his help on simulations, and anonymous reviewers for their constructive comments.

REFERENCES

[1] Q. Shi, "Power Management in Networked Sensor Radios - A Network Energy Model," *Proc. IEEE Sensors Applications Symp. (SAS '07)*, pp 1-5, 2007.
[2] T. Elbatt, C. Saraydar, M. Ames, and T. Talty, "Potential for Intra-Vehicle Wireless Automotive Sensor Networks," *Proc. IEEE Sarnoff Symp.,* 2006.
[3] A.M. Orndorff, "Transceiver Design for Ultra-Wideband Communications," Master's thesis, Electrical Eng., Virginia Polytechnic Inst. of and State Univ., May 2004.
[4] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V.C.M. Leung, "Body Area Networks: A Survey," *Mobile Networks and Applications,* vol. 16, pp. 171-193, 2011.
[5] "Personal Area Networking Profile," http://www.ieee802.org/15/pub/TG4.html, 2012.
[6] S.S. Intille et al., "Using a Live-In Laboratory for Ubiquitous Computing Research," *Proc. Fourth Int'l Conf. Pervasive Computing,* pp. 349-365, 2006.
[7] J. Krumm, "Ubiquitous Advertising: The Killer Application for the 21st Century," *IEEE Pervasive Computing,* vol. 10, no. 1, pp. 66-73, Jan.-Mar. 2011.
[8] J.-S. Lin and C.-Z. Liu, "A Monitoring System Based on Wireless Sensor Network and an SoC Platform in Precision Agriculture," *Proc. IEEE 11th Int'l Conf. Comm. Technology (ICCT),* pp 101-104, Nov. 2008.
[9] T. Culter, "Deploying ZigBee in Existing Industrial Automation Networks," *CIRRONET, Inc.,* http://www.industrial-embedded.com/, 2012.
[10] "IEEE 802.11 wg, Part 11: Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specification," *IEEE Standard,* Aug. 1999.
[11] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *Proc. Second Int'l Conf. Embedded Networked Sensor Systems (SenSys '04),* pp. 95-107, 2004.
[12] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Proc. IEEE INFOCOM,* 2002.
[13] K.S.J. Pister and L. Doherty TSMP: Time Synchronized Mesh Protocol, *IASTED DSN,* Nov. 2008.
[14] S. Yoon, C. Qiao, R.S. Sudhaakar, J. Li, and T. Talty, "QoMoR: A QoSaware MAC Protocol Using Optimal Retransmission for Intra-Vehicular Wireless Sensor Networks," *Proc. INFOCOM,* 2007.
[15] R.S. Sudhaakar, S. Yoon, J. Zhao, and C. Qiao, "A Novel QoS-Aware MAC Scheme Using Optimal Retransmission for Wireless Networks," *IEEE Trans. Wireless Comm.,* vol. 8, no. 5, pp. 2230-2235, May 2009.
[16] J. Zhao, R.S. Sudhaakar, S. Yoon, and C. Qiao, "Constrained Scheduling in Hybrid Wireless Sensor Networks with Transmit-Only Nodes," *Proc. IEEE Int'l Conf. Comm. (ICC),* 2010.
[17] J. Zhao, R.S. Sudhaakar, and C. Qiao, "Providing Reliable Data Services in Hybrid WSNs with Transmit-Only Nodes," *Proc. IEEE Global Telecomm. Conf. (Globecom '10),* 2010.
[18] R.S. Sudhaakar, C. Qiao, S. Yoon, and J. Zhao, "A MAC Protocol for Real-Time Sensing Applications Using Asymmetric Transceivers," *Proc. IEEE Sixth Int'l Conf. Mobile Ad Hoc and Sensor Systems (MASS),* 2009.
[19] H.C. Keong and M.R. Yuce, "Analysis of a Multi-Access Scheme and Asynchronous Transmit-Only UWB for Wireless Body Area Networks," *Proc. IEEE Ann. Int'l Conf. Eng. in Medicine and Biology Soc. (EMBC),* 2009.
[20] C. Huebner, S. Hanelt, T. Wagenknecht, R. Cardell-Oliver, and A. Monsalve, "Long Range Wireless Sensor Networks Using Transmit-Only Nodes," *Proc. Eighth ACM Conf. Embedded Networked Sensor Systems (SenSys '10),* Nov. 2010.
[21] B. Radunovic, H.L. Truong, "Receiver Architectures for UWB-Based Transmit-Only Sensor Networks," *Proc. IEEE Int'l Conf. Ultra-Wideband (ICU '05),* pp 379-384, 2005.
[22] B. Blaszczyszyn and B. Radunovic, "Using Transmit-Only Sensors to Reduce Deployment Cost of Wireless Sensor Networks," *Proc. IEEE INFOCOM,* pp 1202-1210, 2008.
[23] Y. Kim, H. Shin, and H. Cha, "Y-MAC: An Energy-Efficient Multi-Channel MAC Protocol for Dense Wireless Sensor Networks," *Proc. Seventh Int'l Conf. Information Processing in Sensor Networks (IPSN),* 2008.
[24] Y. Yang and Y. Weidong, "Rainbow: Reliable Data Collecting MAC Protocol for Wireless Sensor Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC),* 2011.
[25] L.F.W. van Hoesel and P.J.M. Havinga, "A Lightweight Medium Access Protocol for Wireless Sensor Networks," *Proc. First Int'l Workshop Networked Sensing Systems (INSS),* 2004.
[26] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," *Proc. ACM First Int'l Conf. Embedded Networked Sensor Systems (SenSys),* 2003.
[27] N. Burri, P.V. Rickenbach, and R. Wattenhofer, "Dozer: Ultra-Low Power Data Gathering in Sensor Networks," *Proc. Sixth Int'l Symp. Information Processing in Sensor Networks (IPSN),* 2007.
[28] QUALNET Simulator, http://www.scalable-networks.com/, 2012.
[29] GreenOrbs, http://greenorbs.org/, 2012.

Jia Zhao received the BS and MS degrees in computer science and technology from Soochow University, Suzhou, China, in 2003 and 2006, respectively. He is working toward the PhD degree at State University of New York (SUNY) at Buffalo. His research interests include the areas of wireless communication and distributed systems, especially the architecture and protocol design for the hybrid wireless sensor networks and Internet of Things. He is a student member of the IEEE.

Chunming Qiao directs the Lab for Advanced Network Design, Analysis, and Research (LANDER), which conducts cutting-edge research with current foci on optical networking and survivability issues in cloud computing, human-factors and mobility issues in wireless networks, low-cost and low-power sensors and mobile (robotic) sensor networking for cyber physical systems. He has published more than 100 and 160 papers in leading technical journals and conference proceedings, respectively, with an h-index of about 50 (according to Google Scholar). He pioneered research on Optical Internet, and in particular, the optical burst switching (OBS). One of his paper on OBS alone has been cited for more than 2,000 times. In addition, his work on integrated cellular and ad hoc relaying systems (iCAR), started in 1999, is recognized as the harbinger for today's push toward the convergence between heterogeneous wireless technologies, and has been featured in BusinessWeek and Wireless Europe, as well as at the websites of New Scientists and CBC. His Research has been funded by nine US National Science Foundation (NSF) grants as a PI including two ITR awards, and by major IT and telecommunications companies including Alcatel Research, Fujitsu Labs, Cisco, Google, NEC labs, Nokia, Nortel Networks, Sprint Advanced Technology Lab, and Telcordia, as well as Industrial Technology Research Institute (in Taiwan). He has chaired and cochaired a dozen of international conferences and workshops. He was an editor of *IEEE Transactions on Networking* and *Transaction on Parallel and Distributed Sytems*, and a guest editor for several *IEEE Journal on Selected Areas in Communications (JSAC)* issues. He was the chair of the IEEE Technical Committee on High Speed Networks (HSN) and currently chairs the IEEE Subcommittee on Integrated Fiber and Wireless Technologies (FiWi) which he founded. He has contributed to optical and wireless network architectures and protocols. He is a fellow of the IEEE.

Raghuram S. Sudhaakar graduated with the PhD degree in computer science from the State University of New York at Buffalo in 2010 and is currently a researcher at Cisco Systems. His research is mainly focused on Vehicular and Sensor Networks and is collaborating on efforts to bring the many years of research in this field into products. He is a member of the IEEE.

Seokhoon Yoon received the MS and PhD degrees in computer science and engineering from the State University of New York (SUNY) at Buffalo, in 2005 and 2009, respectively. After receiving the PhD degree, he worked as a senior research engineer in defense industry where he designed several tactical mobile ad hoc network solutions. Currently, he is an assistant professor at the University of Ulsan, South Korea, where he leads the Advanced Mobile Networking Lab. His research interests include ad hoc networks, mobile sensor and actuator networks, networked robots, and underwater networking. He is a member of the IEEE and the IEEE Computer Society.