Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 725421, 19 pages
doi:10.1155/2012/725421

Research Article

LiReTa: A Lightweight Reliable Transmission Scheme for Wireless Sensor Networks Using Cross-Layer Information

Ga-Won Lee and Eui-Nam Huh

Department of Computer Engineering, KyungHee University, Global Campus, 1-Seochon-dong, Giheung-gu, Gyeonggi-do, Yongin-si 446-701, Republic of Korea

Correspondence should be addressed to Eui-Nam Huh, johnhuh@khu.ac.kr

Received 6 January 2012; Revised 6 May 2012; Accepted 23 May 2012

Academic Editor: Regina B. Araujo

Information delivered through sensor networks is used in industries to increase quality of life (QoL). Lossless data in wireless sensor networks (WSNs) is a communications challenge that stands in the way of accurate data delivery. Although end-to-end data retransmission has evolved as a reliable mode of data transportation for the Internet, it is not applicable to WSNs due to the lack of reliable wireless links and resource constraints in sensor nodes. In this paper, we propose an efficient and reliable overhearing-based data transfer protocol for WSNs by introducing selective "direct acknowledgement (ACK_{dir})" or "implicit acknowledgement (ACK_{imp})" in cross-layer design. This protocol assesses the path (or link) quality and delegates the ACK message if it has good communication on paths. In addition, the protocol uses implicit ACK in order to be energy efficient, reducing traffic. Simulation results show that energy efficiency is improved by 30% compared with other approaches.

1. Introduction

In recent years, wireless sensor networks (WSNs) have expanded from simple environmental surveillance and information delivery systems to various mission critical applications such as Ubiquitous (u-) healthcare services, u-agriculture systems, and u-defense. These applications require the reliable delivery of high-priority events to sinks, reliable control and management of the sensor network structure, and the capacity for remote programming/re-tasking of sensor nodes in a controlled, reliable, robust, and scalable manner [1]. Importantly, all of these applications necessitate that all data are to be transmitted without loss within their respective WSNs.

However, unlike traditional networks (e.g., IP networks), reliable data transmission remains a challenge in WSN environments. WSNs are highly distributed self-organized systems that rely on significant numbers of scattered low-cost tiny devices/sensor nodes featuring major limitations with respect to processing, memory, communications, and power capabilities. Since sensor nodes are highly resource constrained, the design of reliable data transmission protocols is very challenging.

Many transport protocols have been proposed and implemented in the literature to improve the reliability of WSNs. These protocols are mainly designed (1) to confirm data transfer and lost data recovery by requesting an acknowledgement (ACK) message (called "direct ACK" denoted as ACK_{dir}) or notifying the sender of failure with a negative acknowledgement (NACK) message [1–4], (2) to increase data transfer success rates by delivering via multiple paths [5–7], and (3) to avoid data collision by using event-based approaches or collision detection approaches [8, 9]. To achieve lossless reliable data transfer, these data transmission protocols commonly select feedback message or recovery factors such as ACK, NACK, hop-by-hop recovery, end-to-end recovery, or the number of packet duplications. However, these static quality-based parameters have limitations with error-prone and unstable WSNs, since they do not apply to dynamic conditions in sensor nodes. Moreover, previous approaches focus on minimum data loss without considering energy consumption. Indeed, the lifespan of sensors and the unreliable nature of WSNs results in two tradeoff problems: energy consumption and lossless data transfer.

Nevertheless, few studies have been devoted to the design of reliable transport protocols using *path reliability* in a multipath routing environment. This approach is primarily used to choose reliable paths based on directed diffusion in cases of delay-sensitive data delivery over error-prone WSNs [10]. Channel error rate is mainly used to measure path reliability; however, this mechanism cannot ensure end-to-end reliability in a WSN environment, which is important for mission critical applications.

In this paper, we propose a Lightweight Reliable data Transmission (LiReTa) method using cross-layer information, which calculates the reliability of *every path (or link) of the node* using the power level received signal strength indicator (RSSI) value and the channel error rate of the node. Moreover, our proposed method overhears communications of neighbor nodes because the source node can overhear the forwarding signal of the receiver as it is sent to the neighbor of the receiver. Thus the sender recognizes successful delivery to its neighbor without receiving the ACK message. This is called "implicit ACK" denoted as ACK_{imp}.

The protocol proposed in this paper contributes to a quick error-recovery of pump-slowly fetch-quickly (PSFQ) [3] while it uses generic ACK/NACK for reliable transmission. In addition, energy efficiency is achieved using a selective ACK mechanism that requests ACK selectively if current path reliability is less than a threshold value. Simulations were conducted to show the effectiveness of LiReTa compared with several existing algorithms in terms of energy consumption and traffic reduction.

The rest of this paper is organized as follows: Section 2 introduces some fundamental factors that determine path-reliability and discusses several well-known reliable transfer schemes and their associated problems. Section 3 describes the proposed algorithm. Section 4 shows transmission use case scenarios. Simulation results of the proposed LiReTa algorithm are described in Section 5, and the conclusions of our work are detailed in Section 6.

2. Related Work

In this section, we first summarize some issues and problems of sensor networks regarding reliable transmission. We then briefly review some basic approaches for reliable data transfer in wireless sensor networks to show how our research builds on previous work.

2.1. Reliable Transport Protocols for WSNs. An exhaustive list and analysis of transport protocols for WSNs can be found in [1, 4, 11]. WSN transport protocols can be classified into five categories: ACK/NACK based schemes, multipath transfer schemes (short multipackets), collision avoidance schemes, and reliability schemes as shown in Table 1 along with some representative methods/examples.

The ACK/NACK mechanism provides *hop-by-hop* or *end-to-end* data dissemination by using ACK and NACK messages in cases of missing sequential packets. This mechanism, when applied to high channel error rate scenarios, causes an ACK implosion problem where the buffer is overflowing due to too frequent retransmission and unnecessary traffic packets and, consequently, is not applicable to WSNs [12].

Table 1: Reliable data transfer methods for WSN.

Type	Method
ACK-based scheme	RM21
NACK-based scheme	PSFQ/GARUDA/RMST
Multipackets	HHR/ReInForM/ReTrust
Collision avoidance	ESRT/CODA
Reliability scheme	Directed diffusion considering reliability

Whenever a node receives a message, it sends ACK_{dir} messages to notify the sender of transmission success. The ACK implosion problem occurs when multiple ACK messages are simultaneously received. Indeed, ACK implosion causes unnecessary traffic and data loss, which decreases the availability and performance of links due to repetitive message transfers.

Reliable multicast with ACK_{imp} and indirect recovery (RM21) [2] is an ACK-based protocol that uses ACK_{imp} messaging and indirect recovery to reinforce the disadvantages of NACK. When the error rate is low, energy consumption is more efficient and RM21, by utilizing ACK_{imp}, consumes less power than methods that send ACK_{dir} messages. However, ACK_{imp} message failure or methods of sensor deployment for RM21 has not been discussed clearly. Furthermore, error recovery rates increase rapidly when error rates increase, thereby reducing energy efficiency.

PSFQ [3] is a protocol that ensures reliability in WSNs. The key idea of the design of PSFQ is to distribute data from a source node by pacing data at a relatively slow speed (pump-slowly), but allowing nodes that experience data loss to fetch quickly (i.e., to recover any missing segments from their local immediate neighbors aggressively) [1]. PSFQ eliminates unnecessary traffic for NACK messages through retransmission requests at a middle node and minimizes the cost of loss recovery by using data localized among immediate neighbors to achieve loose delay bounds. However, the middle node in standby status is unable to transfer lost packets located in the buffer until the next node notices that a packet is missing or retransmission is complete. Thus, the entire data transmission time is much longer and increases the possibility of buffer overflow in the middle nodes.

A scalable approach for reliable downstream data delivery in wireless sensor networks (GARUDA) [1] solves the first sequence packet transfer problems found with NACK protocols. Specifically, it guarantees the reliability of the first packet by using a wait for the first packet pulse (WFP), where the core node acts as a recovery server when the data transmission fails using downstream data. However, energy consumption with this protocol is very high and, consequently, it is inappropriate for WSNs.

Table 2: Reliable data transfer method comparison.

Protocol	Reliable guarantee	Direction	Energy efficiency	Recovery	Data transfer speed
RM21	End-to-end/node-to-node	Up/down	Average	ACK based	Slow
PSFQ	End-to-end/node-to-node	Downstream	Low	NACK based	Slow
GARUDA	End-to-end	Downstream	Low	NACK based	Initially slow, after average
RMST	End-to-end	Upstream	Average	NACK based	Average
Heuristic	End-to-end	Upstream	Average	ACK/NACK	Average
ReInForM	End-to-end	Upstream	Based on reliability requirement	—	Slow
HHR	End-to-end	Upstream	Low	—	Slow
ReTrust	End-to-end	Upstream	Based on reliability requirement	—	Slow
ESRT	Event reporting	Upstream	Average	—	Average
CODA	Event reporting	Upstream	Average	—	Average
DD with reliability	End-to-end/node-to-node	Up/down	Low	—	Average

Reliable multi-segment transport (RMST) [4] is investigated through simulations of the tradeoff in implementation reliability between MAC, transport, and application layers. The conclusion is that hop-by-hop recovery plays an important role in achieving reliability and end-to-end recovery is inadequate. However, packet recovery using source nodes swamps the load to source node in non-caching mode.

In addition to ACK/NACK based protocols, hop-by-hop reliability (HHR) schemes [5] and reliable information forwarding using multiple paths (ReInForM) [6] provide reliable transmission by combining multi-packet and multi-path methods. HHR uses unicast transmission to transfer many copies of a single packet. This scheme considers packet loss rate, packet transfer possibility, and number of hops to create copies of the packet. However, if the channel quality is poor, all of the copies may be wasted.

On the other hand, ReInForM compensates for the disadvantages of HHR by transferring copies through random multiple paths to maintain a specified reliability and to prevent energy inefficiency in good quality paths due to traffic that is dispersed to many nodes. However, the channel error rate increases when the number of hops between source and sink is large, which causes the copies of packets and the number of paths to grow exponentially. Hence, ReTrust [7] improves upon the ReInForM framework by focusing on efficiently reducing such loads using intermediate source/sink (IS) in sensor networks; however, unnecessary traffic delays may remain and load problems may still occur in IS nodes.

In addition to the previous methods used in WSNs as described above, there are other methods such as event-to-sink reliable transport (ESRT) [8], congestion detection and avoidance (CODA) [9], and a method proposed in [10] that applies reliability in the routing path to guarantee path reliability. ESRT employs an event-to-sink reliability model to provide reliable event detection that embeds a congestion control component [3] and manages different events with different levels of reliability. CODA is an energy congestion control scheme that avoids collisions in WSNs and comprises three mechanisms: congestion detection, open-loop hop-by-hop back pressure, and closed-loop; however, such detection of loading states in channels consumes a significant amount of energy.

In [10], a reliable data transfer mechanism using directed diffusion in WSNs is proposed. This mechanism involves selecting a path with higher reach-ability and transferring data along the path chosen. The path choice is based on end-to-end reliability as calculated by the dissemination procedure of the Interest packets, while each node of a sensor network maintains only the information in its neighborhood. [10] only considers channel error rate for routing path reliability. Table 2 presents a comparative analysis of the existing reliable transport protocols for WSNs.

3. LiReTa: Lightweight Reliable Data Transmission Method

3.1. Preliminaries

3.1.1. Overhearing. Due to the broadcast nature of wireless channels, many nodes in the vicinity of a sender node overhear its packet transmissions even if they are not the intended recipients of these transmissions [13]. This redundant reception results in unnecessary expenditure of battery energy of the recipients. Turning off neighboring radios during a certain point-to-point wireless transmission can mitigate this cost [13, 14].

3.1.2. Received Signal Strength. In wireless communication, received signal strength indicator (RSSI) has a crucial role in detecting received power because, it provides the information necessary to adjust the receiver's gain.

RSSI is the relative received signal strength in a wireless environment employing IEEE 802.11 expressed in arbitrary units. RSSI measurements range from 0 to 255 and are expressible as a one-byte unsigned integer. For typical wireless communication applications, RSSI circuits should have a wide range over 60 dB with fast settling time using the received start-up signal. The maximum value, RSSI_Max is vendor-dependent (i.e., Cisco Systems cards return a RSSI value from 0 to 100 where RSSI_Max is 100, and Atheros Wi-Fi chipsets return an RSSI value from 0 to 127 (0×7f), with 128 (0×80) indicating an invalid value).

Since RSSI is not absolutely accurate and stable, the performance of dissemination reliability estimation largely depends on estimation methods.

3.2. Overview of the Proposed Approach. Reliable data transfer using *each path reliability* in WSNs has not previously been discussed in the literature. The primary motive of LiReTa is to provide reliable end-to-end data delivery in a WSN environment with minimum energy expenditure. It uses an *each path-reliability approach* to select between ACK_{dir} and ACK_{imp}. Further, LiReTa measures the path reliability using RSSI and channel error rate (CER), and also considers energy efficiency by minimizing the number of retransmissions and the number of ACK messages. In previous research, CER has only been applied to path reliability. However, because the operating condition of each node is influenced by the current network environment, the CER varies with time. Moreover, WSNs are composed of low energy power sensor nodes that are capable of sensing particular physical phenomena in their vicinities and communicating among themselves using wireless transceivers. Such low power wireless data communication features make WSN data dissemination unreliable. Therefore, we define reliability as the combination of CER with RSSI that represents the signal strength of neighbors for applying varying network conditions. RSSI is proportional to the input power level [15], and thus reliability using RSSI reflects the currently remaining sensor node power level, as shown in Figure 1.

We assume that sensor nodes are deployed in a general grid form in LiReTa. After sensor nodes are deployed, each sensor node shares RSSI values with neighboring nodes during network configuration. In this process, we calculate the path reliability using RSSI values and CER. When data is successfully transmitted, the calculated reliability is set as the threshold of reliability (used by base reliability later).

For data transfer, the node compares the base reliability and the current reliability. If the current reliability is lower than the base reliability, the ACK_{dir} is requested. Otherwise transfer success is confirmed using ACK_{imp}, which is obtained because nodes used in wireless communications can overhear transmissions to other nodes (discussed in more detail in Section 3.7). Hence, when ACK_{dir} is requested but the transmission success possibilities are higher in the next path, by applying ACK_{dir} delegation to the next node, unnecessary traffic and overhead caused by ACK/NACK can be eliminated, which results in increased energy efficiency. This method detects errors between nodes to provide quick error recovery. Importantly, focusing on single path (single channel) reliability is a significant difference from previous approaches using combined channel reliability. The overall LiReTa procedure consists of 6 steps and is outlined briefly as shown in Figure 2.

Step 1. Sensor nodes are deployed in grid form (described in Section 3.4).

Step 2. Consider the number of retransmissions. If the network requires a limited number of retransmissions, consider the number of retransmissions in the ACK selection procedure (described in Section 3.5).

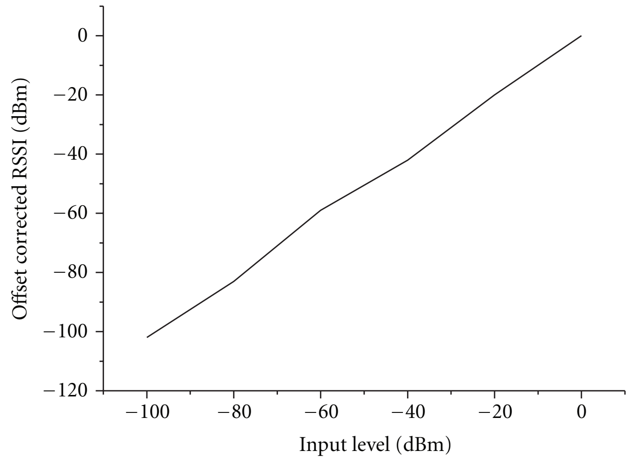


Figure 1: Offset corrected CC2520 RSSI versus input power level [15].

Step 3. Calculate reliability using RSSI and CER (described in Section 3.6).

Step 4. Select algorithm for reliable transmission considering overhead. That is, select ACK_{dir} or ACK_{imp} method according to path reliability considering the number of retransmissions (described in Sections 3.7.1 and 3.7.2).

Step 5. Delegate to improve energy efficiency. If ACK_{dir} is used, then the node performs the ACK_{dir} delegation process (described in Section 3.8).

Step 6. Update base reliability. If data transfer is successful, maintain the base reliability. If data transfer fails, increase the base reliability (described in Section 3.9).

In the proposed scheme, each node receives selective ACK feedback messages to guarantee reliability. The goal of LiReTa is to reduce the number of ACK messages using a selective ACK method that avoids the ACK implosion problem. In addition, well deployed sensor nodes such as grid topology can be used to reduce duplicated messages at each node, since a node can only send to the fixed neighbor node by using a well-coordinated node ID system. The ACK_{imp} can be reduced when data communication channels are scheduled efficiently by a MAC layer.

Moreover, the ACK_{dir} message is only used in the worst case when the path quality of a node is worse than the base reliability, or the number of network retransmissions is less than the required number of retransmissions. Use of the selective ACK method between ACK_{imp} and ACK_{dir} can thus reduce the ACK implosion problem significantly. In the next sections we explain each of these steps in more detail.

3.3. Cross-layer Design. Networks are organized as a series of layers, each one built upon the one below it. The main purpose of layered protocol architecture is to reduce

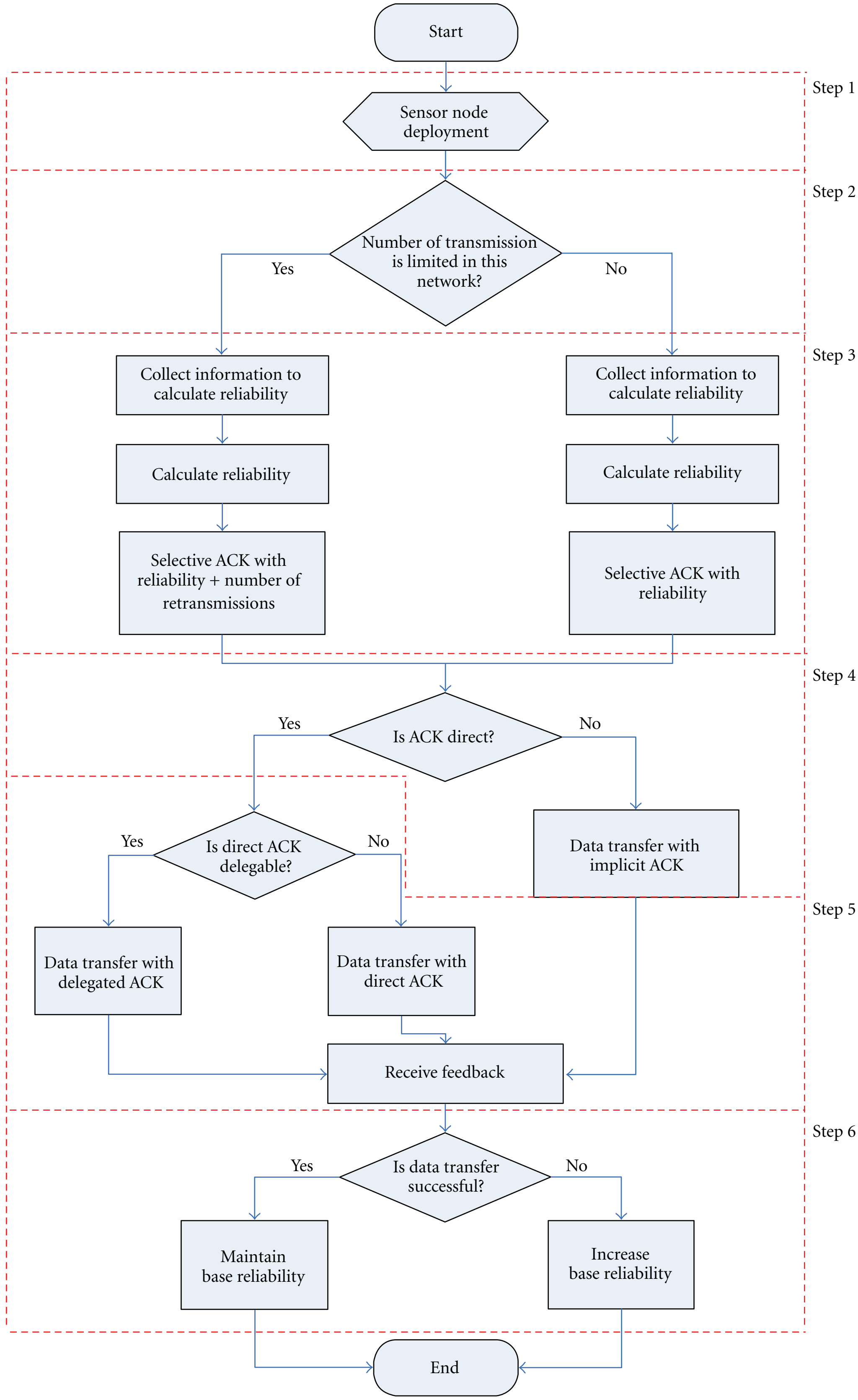


Figure 2: LiReTa procedure.

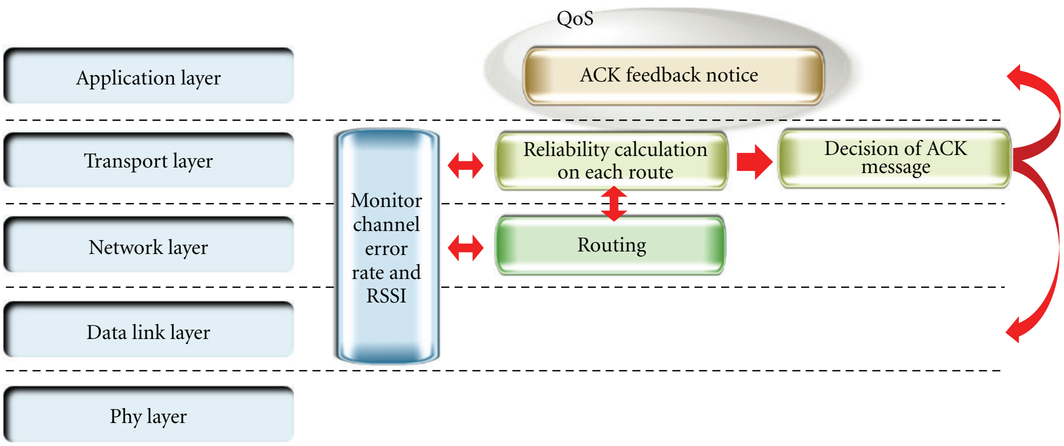


Figure 3: Cross-layer design in LiReTa.

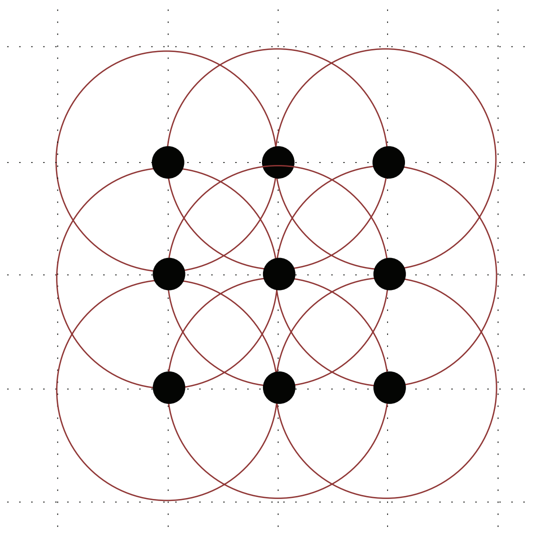


Figure 4: Sensor node deployment.

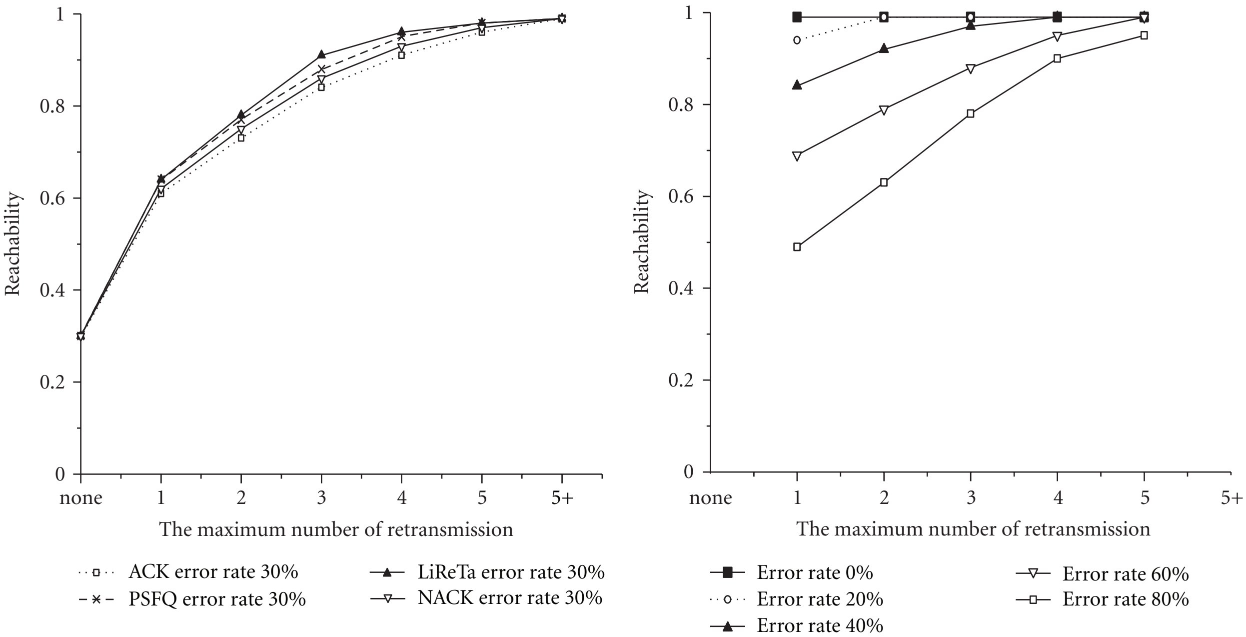
the complexity of system design. If the network is split into smaller modules with different functionalities, design is more manageable and implementation is easier. However, cross-layer design exploits the interactions between layers and promotes adaptability in all layers based on information exchange between layers. Moreover, cross-layer design produces tight interdependence between layers, especially in WSNs.

We designed LiReTa using cross-layer architecture with a holistic view of WSNs to maintain the layered approach, while accounting for interactions between various protocols at different layers. Figure 3 shows how cross-layer design is applied in LiReTa.

3.4. Basic Node Deployment Structure of LiReTa. The sensor node has mobility and works as both a host and a relay router. Therefore, broadcasting and multicasting are necessary for checking node position, signal strength, and network conditions [16, 17]. A *limited node transmission method* was studied to manage the excessive duplicated message problem known as a broadcast storm. These sets of message delivery nodes are connected dominating sets(CDS) within a given network and the solution to find the least-cost CDS was determined to be nonpolynomial complete (NP-complete) [18]. Thus, while various heuristics are used to find CDS, this paper will restrict CDS by deploying sensor nodes.

We assume that sensor nodes are deployed in a grid form as shown in Figure 4, which is a form that is commonly used in topology research; sensor nodes provide multipath routing to test neighbors' reliability and contains most of WSN constraints rather than other topologies. Most of the related work can also be easily implemented with this topology.

3.5. Reliability and the Number of Retransmissions. Our approach, LiReTa, also considers periodicity in WSNs since most applications in environmental, military, and medical environments sense and transmit data periodically. If there is no limit to the total number of retransmissions in WSNs,



(a) Transmission success rates with ACK, PSFQ, LiReTa, NACK

(b) Transmission success rates with 0%, 20%, 40%, 60%, 80% error rate cases

FIGURE 5: Transmission success rates with each scheme and error rate.

urgent data delivery is impossible due to surplus retransmission packets. Unlimited retransmissions also consume a great deal of energy. In general, the communication system limits the number of total retransmissions to prevent infinite transmission. Limited retransmission processes are key elements to support quality of service (QoS) for high performance networks. In WSNs with few sensor nodes, the number of retransmissions is critical for maintaining the reliability of whole networks. In this paper, we analyze the impact of retransmission for reliability as it applies to WSNs.

3.5.1. Consideration of a Limited Number of Retransmissions. We conducted several experiments to verify failure with a limited number of retransmissions. Simulations were performed on a single path topology with 10 nodes. Network conditions including packet loss rates and CER were randomly allocated within the range of error rates in the overall network. In our experiments, reachability was defined as the ratio of the number of packets that arrived at the sink node to the number of packets sent by a source node [19].

The reachability shown in Figure 5 converges to 1 when the number of retransmissions was limited to 5 by an NS-2 simulator, compared with ACK, PSFQ, NACK, and LiReTa schemes. In Figure 5(b), the results show varied reachability in terms of the number of retransmissions with different error rates. Error rates in the model were set at 0%, 20%, 40%, 60%, and 80%.

3.5.2. Relationship between RSSI and CER. As shown in Figure 6(b), when the number of retransmissions is 2 and the CER for two nodes is 80%, the success rate of packet delivery is very small. In order to find general metrics affecting retransmissions, RSSI and CER are investigated and analyzed as shown in Figure 6.

Based on Figure 6, the correlation analysis for RSSI and CER is calculated as follows:

$$\text{Correlation } P_{xy} = \frac{(1/n)\sum_{i=1}^{n}(x_i - \mu_x)\left(y_i - \mu_y\right)}{\sigma_x \times \sigma_y} = -0.9605, \tag{1}$$

where

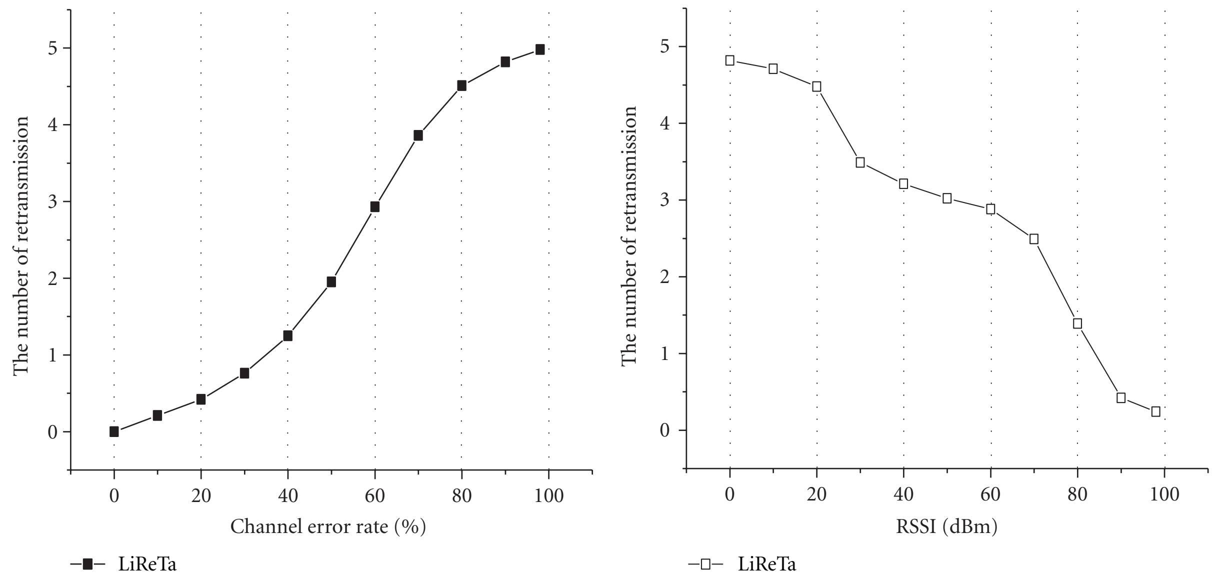
(i) x: average number of retransmissions of a sample for RSSI,

(ii) y: average number of retransmissions of a sample for CER,

(iii) n: sample size.

As shown in (1), the correlation between RSSI and CER with retransmissions is very high. We found that these two metrics are highly related to the number of retransmissions, and can be applied to design cross-layer-based reliable transmission.

3.5.3. Appropriate Number of Retransmissions for CER and RSSI. For mission critical systems requiring high communication environment reliability with fast delivery, setting a small number of retransmissions is appropriate. We simulate 100 packets data transmission between A-B nodes with 65% CER when the number of retransmission was limited 0 to 2 as shown in Table 3 by an NS-2 simulator. As tested, if the number of transmissions is set to 2, 20% failure occurs but an average of retransmissions observed by 1.84. To avoid



(a) CER with number of retransmission

(b) RSSI with number of retransmissions

Figure 6: RSSI and CER with Number of Retransmissions.

Table 3: Total retransmissions under the same conditions.

Number of retransmissions (when data delivery succeeds)	Failure
0	42
1	38
2	20

such failures in critical environments, configuring (setting) the number of retransmissions plays an important role in reliable transmissions.

To set the desired number of retransmissions RT_i on node i for successful delivery, we simply add the average and standard deviation of the number of retransmissions for successful delivery from our experiments to further increase the rate of successful delivery. Therefore, our approach is one in which each node decides to use $\mathrm{ACK}_{\mathrm{dir}}$ or $\mathrm{ACK}_{\mathrm{imp}}$ based on the path reliability and the required number of retransmissions, which previous approaches do not consider. RT_i is calculated as follows:

$$\mathrm{RT}_i = m + \sqrt{\frac{n\sum m^2 - \left(\sum m\right)^2}{n(n-1)} - 1}, \quad (2)$$

where

(i) m: average number of retransmissions on node i,

(ii) n: sample size.

To utilize this information, we need to investigate more facts (correlations) related to retransmissions and CER, and retransmissions and RSSI. Many simulations are performed using NS-2 to determine the minimum transmission time reported (RT_i) for successful delivery. The results are listed in Tables 4 and 5, respectively. The RT_i with CER and RSSI considered together is analyzed as shown in Figure 7.

3.6. Reliability Calculation. After sensor nodes are deployed and share information, each node calculates the path reliability for the selective ACK procedure. As mentioned above, our approach utilizes CER and RSSI for the reliability calculation as shown in Section 3.5. The monitored CER accumulates errors which occur on all paths. Generally, CER is average value of node A in WSN. In other words, CER is average value of whole path in node A. For example, if a node A and other neighbor nodes B, C, D are deployed as shown in Figure 8, the CER of node A denoted by E_A can be calculated according to (3). Applying CER in each path separately is one of our contributions in this paper:

$$E_A = \frac{\sum_{k=1}^{n} E_{ik}}{n}, \quad (3)$$

where

(i) A: current node to calculate reliability,

(ii) n: number of neighbor nodes,

(iii) E_A: CER of node$A$,

(iv) E_{AK}: $A \rightarrow k$ path CER.

In this paper, the overall reliability (E_A) of node A is separated into paths *A-B, A-C*, and *A-D*. If we consider the average of the CER or the summation of the CER, then some unfair and unreliable events may occur. For example, if the CER is 0.2, 0.6, and 0.1 for paths *A-B, A–C*, and *A–D*, respectively, node A's averaged CER value is 0.3. The average CER for node A seems acceptable even though one path (*A-D*) has very poor reliability. Instead of using the average,

Table 4: Number of retransmissions required for CER.

Channel error rate (CER)	Average	Standard deviation	Average + standard deviation	Total number of transmissions	Number of retransmissions
0–10	0.07	0.256	0.326	1	0
10–20	0.23	0.529	0.759	1	0
20–30	0.48	0.741	1.161	2	1
30–40	0.70	1.078	1.778	2	1
40–50	1.03	1.159	2.189	3	2
50–60	1.18	1.274	2.454	3	2
60–70	1.78	1.345	3.125	4	3
70–80	2.26	1.368	3.628	4	3
80–90	2.78	1.292	4.072	5	4
90–100	3.63	1.397	5.027	6	5

Table 5: Number of retransmissions required for RSSI.

RSSI	Average	Standard deviation	Average + standard deviation	Total number of transmissions	Number of retransmissions
(−100)–(−90)	4.05	1.009	5.059	6	5
(−90)–(−80)	3.07	1.297	4.367	5	4
(−80)–(−70)	2.67	1.477	4.147	5	4
(−70)–(−60)	2.28	1.505	3.785	4	3
(−60)–(−50)	1.85	1.566	3.416	4	3
(−50)–(−40)	1.65	1.572	3.222	4	3
(−40)–(−30)	1.01	1.275	2.285	3	2
(−30)–(−20)	0.46	0.797	1.257	2	1
(−20)–(−10)	0.25	0.575	0.825	1	0
(−10)–(−0)	0.08	0.273	0.353	1	0

Channel error rate (%)

RSSI (dBm)	100–90	90–80	80–70	70–60	60–50	50–40	40–30	30–20	20–10	10–0
−100 – −90	5	5	5	5	5	5	5	5	5	5
−90 – −80	5	4	4	4	4	4	4	4	4	4
−80 – −70	5	4	4	4	4	4	4	4	4	4
−70 – −60	5	4	3	3	3	3	3	3	3	3
−60 – −50	5	4	3	3	3	3	3	3	3	3
−50 – −40	5	4	3	3	3	3	3	3	3	3
−40 – −30	5	4	3	3	2	2	2	2	2	2
−30 – −20	5	4	3	3	2	2	1	1	1	1
−20 – −10	5	4	3	3	2	2	1	1	0	0
−10 – 0	5	4	3	3	2	2	1	1	0	0

Figure 7: Retransmissions resulting in successful delivery with CER and RSSI considered together.

we need to use the CER of each path to differentiate the *reliability of each path*.

We previously discussed the very strong relationship between CER and RSSI in Section 3.5. The RSSI value of a node allows neighbors to recognize that the sender node is healthy simply by observing the received field in the L2 (data link layer) frame, which enables a node to determine the reliability of the neighbor nodes. RSSIs are basically negative forms; therefore, normalization steps are required as follows:

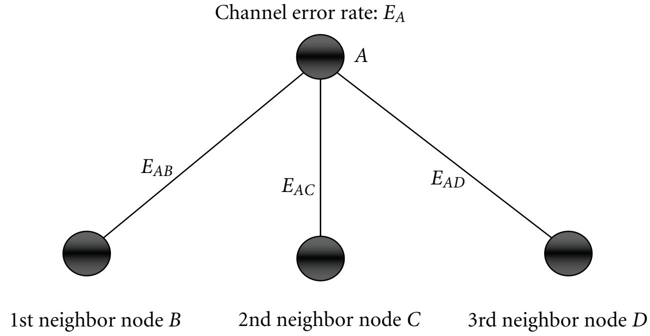


Figure 8: CER of node A.

$$\mathrm{NRSSI}_{AK} = 1 - \left| \frac{\mathrm{RSSI}_{AK}}{\mathrm{MaxRSSI}} \right|, \quad (4)$$

where

(i) RSSI_{AK}: $A \rightarrow K$ RSSI,

(ii) NRSSI_{AK}: $A \rightarrow K$ normalized RSSI.

From the average of CER of node A, using (3) and (4), we can calculate CER of specific path E_{AK}. In this paper, we present the reliability of each path. Hence, from the $1 - E_A$ equation, we can get the reliability of average of node A. For the separate path reliability, R_{AK}, the path reliability from node A to K, is calculated as follows:

$$R_{AK} = (1 - E_A) \times \frac{QRSSI_{AK}}{\sum_{k=1}^{n} QRSSI_{Ak}}, \quad (5)$$

where $R_{AK}: A \rightarrow K$ is the path reliability.

3.7. Selective ACK Method

3.7.1. Base Reliability. When a node successfully exchanges an RSSI, it computes the reliability using the received RSSI to set a base reliability, that is, a threshold value for the trigger request of the ACK. If the transfer to exchange RSSI fails, the ACK request is sent until the node reaches the base reliability. The reliability calculation is calculated using (2) and (3), and used as the initial base reliability.

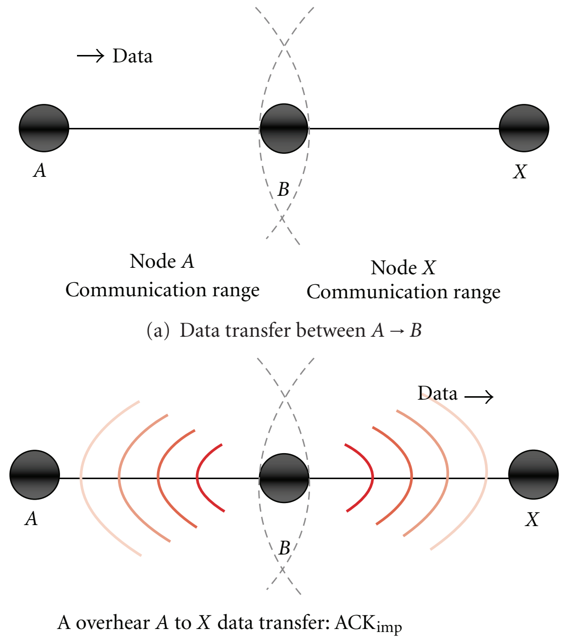
3.7.2. Implicit ACK. A sensor node uses radio channels for communication. Due to the broadcast nature of the wireless channel, many nodes in the vicinity of a sender node overhear its packet transmissions even if those are not the intended recipients of these transmissions [13]. This redundant reception results in so-called overhearing problems in IEEE 802.15.4 protocols. Turning off neighboring radios during point-to-point wireless transmission can mitigate this cost [13, 14]. Such overhearing problems are used positively as an implicit acknowledgement mechanism in the proposed LiReTa scheme.

Figure 9 shows the ACK_{imp} mechanism. Using the selective ACK method, we determined whether current path reliability was better than the base reliability to confirm data transmission achievement using ACK_{imp}. If path quality was lower than base reliability, ACK_{imp} transfer failure becomes high. In this low path quality case, ACK_{dir} was used to guarantee reliability and fast recovery. Otherwise, as in Figure 9(b), by listening to node B's forwarding signal, node A receives an implicit ACK message.

As shown in Algorithm 1, through comparing the current path reliability, R_{AB}, and the base reliability, BR_{AB} (initially received RSSI), and checking RT_A (required retransmissions for successful delivery) and the configured retransmission limit, NT_A, we decide on either ACK_{dir} or ACK_{imp}. After the proper ACK method is selected, data forwarding is started for the next hop.

The case, $R_{AB} > BR_{AB}$ and $RT_i \le NT_i$, means that current reliability is better than the base reliability and current path quality (RSSI value) of A requires less retransmission than the configured retransmission limit. That is, the current path is very reliable for delivering data successfully.

The case, $R_{AB} > BR_{AB}$ and $RT_i > NT_i$, means that current reliability is better than the base reliability and that the current path quality (RSSI value) of A requires more retransmissions than the configured retransmission limit. That is, base reliability is very poor, and therefore not enough



(a) Data transfer between $A \rightarrow B$

(b) Data transfer between $B \rightarrow X$

FIGURE 9: Implicit ACK using overhearing.

data exist to determine if the current reliability is good, or NT_i is configured to be too small to have a very strict QoS requirement.

3.8. Delegation. In this section, we classify four scenarios to compare current path reliability and base reliability for delegation of ACK requests between A-B and B-X.

Scenario 1: $R_{AB} > BR_{AB}$ && $R_{BX} > BR_{BX}$. In this scenario, the quality of all current paths is satisfied. Nodes A and B do not request ACK_{dir} when node B sends data to node X. Likewise, node A overhears this data transmission and accepts this as an ACK_{imp}.

Scenario 2: $R_{AB} > BR_{AB}$ && $R_{BX} \le BR_{BX}$. In the second scenario, node A-B path quality is satisfied and does not require ACK_{dir} messages while node B requests an ACK_{dir} message. When the transmission begins, node A receives ACK_{imp} and node B receives ACK_{dir} from node X.

Scenario 3: $R_{AB} \le BR_{AB}$ && $R_{BX} > BR_{BX}$. In this scenario, current path A-B reliability is worse than base reliability A-B. Node A confirms transmission success through an ACK_{dir} message. Node B compares the next path B-X with base reliability and the path B-X quality is confirmed. There is no request for an ACK_{dir} message to node X. Even if node A requests an ACK_{dir} message, B-X path quality is good enough to make an ACK message unnecessary. Therefore, node B delegates ACK_{dir} to node X and node A overhears this information as an ACK_{imp}.

```
*compare path reliability and required network retransmission limitation*/
If (R_AB > BR_AB)
{
  if (RT_i ≤ NT_i)
    Forward Data (B, X, Data); // ACK_imp
  else // RT_i > NT_i
    Forward Data (B, X, Data, ACK); // ACK_dir
}
else // R_AB ≤ BR_AB
{
  if (RT_i ≤ NT_i)
    Forward Data (B, X, Data, ACK); // ACK_dir
  else // (RT_i ≤ NT_i)
    Forward Data (B, X, Data, ACK); // ACK_dir
}
```

Algorithm 1: Selective ACK algorithm.

```
/*Change base reliability*/
// use ACK_imp, good path quality
If (R_AB > BR_AB)
    if (B, Recv Data (A))
    {
          BR_AB = BR_AB; // success: maintain base reliability
          Clear Buffer (A); // success: clear A buffer
    }
    else
          BR_AB = R_AB; // fail: increase base reliability
// use ACK_dir, bad path quality
Else // R_AB ≤ BR_AB
    if (B, Recv Data (A))
    {
        BR_AB = R_AB; // success : decrease base reliability
        Clear Buffer (A); // success: clear A buffer
    }
    else
    BR_AB = BR_AB; // fail: maintain base reliability
```

Algorithm 2: Base reliability change algorithm.

Scenario 4: $R_{AB} \leq BR_{AB}$ && $R_{BX} \leq BR_{BX}$. In the last scenario, the quality of path *A*-*B*-*X* is poor. Node *B* sends an ACK_{dir} message when *B* receives data from *A*. Node *X* sends an ACK_{dir} message when *X* receives data from *B*. In this case, path quality is disqualified for ACK_{imp}, and thus an ACK_{dir} message is highly recommended. When node *A* transfers data to *B* after deciding between ACK_{dir} and ACK_{imp} algorithms, node *B* initiates a delegation process as described in the third scenario. If A requests ACK_{dir} because the path reliability is poor, *B* must decide whether to use ACK immediately or to delegate ACK. If the current path reliability is better than the base reliability, data dissemination will succeed even though there are no ACK requests. Node *B* then compares the base reliability to the next path reliability to decide whether to delegate. In this case, one node request ACK_{dir} from a former node and, if the next path reliability is higher than the base reliability, the node delegates ACK_{dir} to the next node.

3.9. Update Base Reliability. After ACK_{dir} or ACK_{imp} selection, the type of ACK is marked and data is sent. Based on the success or failure of data transfer, the base reliability is changed. This base reliability change reflects current network and node conditions. The base reliability updates the algorithm for node *A*-*B* as shown in Algorithm 2.

If data transfer succeeds with ACK_{imp} (meaning good path quality), base reliability is maintained. Otherwise, we need to replace the base reliability with the current path reliability. Likewise, if data transfer succeeds with ACK_{dir} (meaning poor network quality), we update the base reliability with the current path reliability. If data transfer fails, we maintain the base reliability.

4. Use Case

In LiReTa, the data transfer procedure chooses the type of ACK to guarantee reliability: ACK_{dir} or ACK_{imp}. In addition,

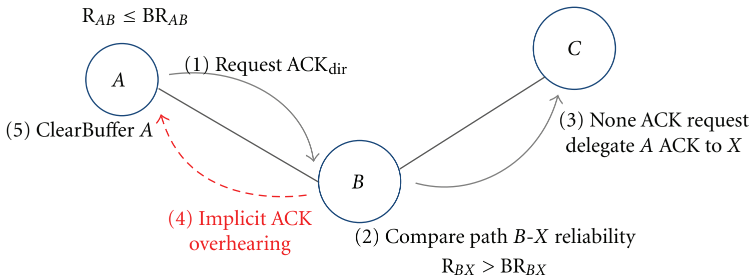


Figure 10: Delegation ACK.

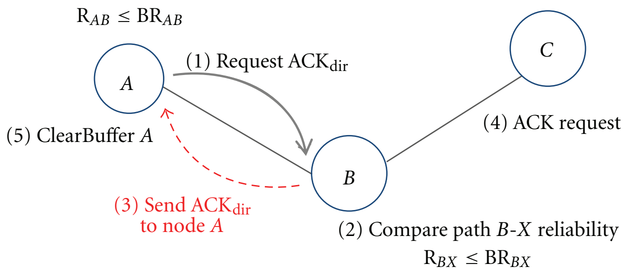


Figure 11: Receiving an ACK_{dir}.

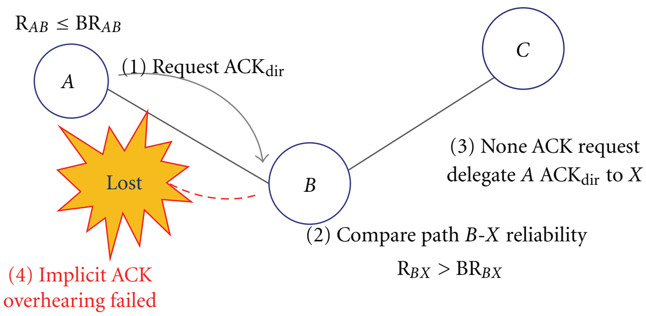


Figure 12: ACK_{dir} failure.

based on data transfer success or failure, this process changes the base of reliability and determines the ACK delegation.

We can categorize all data transfer cases as follows:

(i) $A \longrightarrow B$ data transfer with ACK_{dir}, B-X Quality is good enough to delegate direct Acknowledgment of node B (denoted ACK_{dir_B}) to XB-X Quality is not good, B returns ACK_{dir_B} to A No response.

(ii) $A \longrightarrow B$ data transfer with ACK_{imp}, B receives data successfully, and sends data to X (A overhears this transmission) No response.

In this section, we will explain these use cases in detail.

4.1. A Data Transfer $A \longrightarrow B$ with an ACK_{dir} Request: $R_{AB} \leq BR_{AB}$

4.1.1. ACK Delegation Occurs between Nodes B and X. Node B analyzes the chance of delegating after successful data transfer along the path A-B. When the path B-X quality (R_{BX}) is higher than the path B-X base reliability (BR_{BX}), ACK_{dir_A} is delegated to node X even though the path A-B quality (R_{AB}) is lower than the path A-B base reliability (BR_{AB}) because the path A-B data transfer has already succeeded. The path A-B base reliability, BR_{AB}, must be decreased to the reliability R_{AB} of the path A-B. This clears the buffer of A as shown in Figure 10.

4.1.2. Node A Receives ACK_{dir_B}. After A-B data transfer success, node B sends ACK_{dir_A} to node A when the path B-X quality is lower than the B-X base reliability. In the $R_{BX} \leq BR_{BX}$ case, node B requests ACK_{dir_B} when node B sends data to node X. At last, the buffer of A is cleared due to A-B data transfer success as shown in Figure 11.

4.1.3. Waiting a Specified Time without Hearing from the Next Node. This state may occur in two cases: data transfer failed from node B or ACK_{dir_A} was delegated but node A failed to overhear ACK_{imp}. The ACK_{dir_A} must be returned to node A or node A must overhear ACK_{imp} since ACK_{dir_A} was requested in the first place. Since nothing was heard from node B, node A sends data to B with the ACK_{dir_A} as shown in Figure 12. The path A-B current reliability decreases due to the incremental increase in the CER.

4.2. Data Transfer $A \longrightarrow B$ with No ACK_{dir} Request: $R_{AB} > BR_{AB}$

4.2.1. Node A Overhears Implicit Acknowledgment of Node B (ACK_{imp_B}). If node A overhears data transfer between node B and node X, data transfer from node A to node B succeeded. The buffer of A is then cleared due to A-B data transfer success as shown in Figure 13.

4.2.2. Waiting a Certain Time without Hearing from the Next Node. The data transfer from node A to node B failed as shown in Figure 14(a). In addition, Figure 14(b) shows that data transfer has succeeded but node A failed to overhear ACK_{imp}. In this case, node A sends data to node B with ACK_{dir} and increases A-B base reliability BR_{AB} to the current A-B reliability R_{AB}. The current A-B reliability decreases due to an incremental increase in the CER.

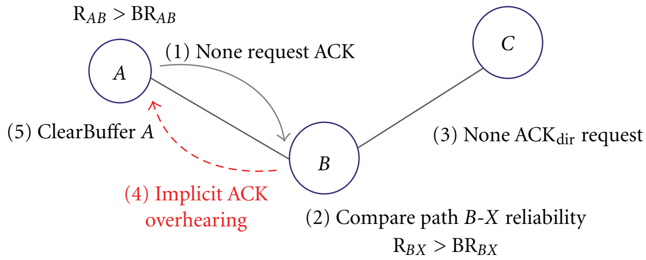
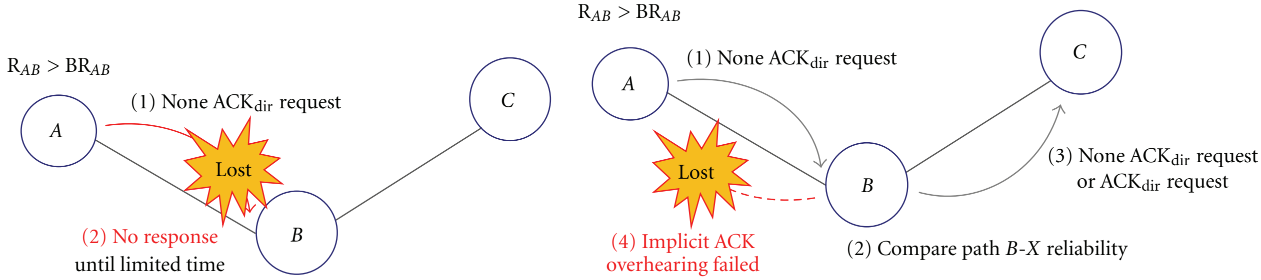


Figure 13: ACK_{imp} overheard.



(a) Node A to B data transfer failure

(b) Node A ACK_{imp} failure to overhear

Figure 14: No ACK feedback.

5. Performance Evaluation and Analysis

5.1. Simulation Environment. In this section, we compare the performance results of LiReTa with generic results of ACK, PSFQ and ReInForM. The performance metrics include the number of transmissions, fault tolerance, node energy consumption, and network lifetime.

For the simulations, the sensor nodes transfer packets to the sink node on a particular routing path. The experiments for ACK and PSFQ methods assumed a fixed error rate. However, the proposed method considers variable reliability and thus only the initial error rate was set. The simulations considered the total number of transmissions including retransmission based on the number of nodes, time, and simulated error-recovery by increasing the size of packets. In addition, the proposed scheme compared energy consumption rates with the generic ACK method.

Simulations were executed over a uniform topology consisting of 0~100 nodes deployed in a square grid form of 100 m × 100 m in the NS-2 network simulator. This method was selected due to the advantages of grid form as discussed in Section 3.4 with 1-hop neighbor overhearing communication radius of each sensor node. Network conditions were randomly allocated within the range of error rate parameter configurations. For example, if the range of the error rate was 50%, random rate values from 0% up to 50% were assigned to the packet loss rate and the CER, respectively. Each sensor node maintained a history of packet loss rates and a channel error count.

We applied the energy model, a network interface model, and an error model included in the NS-2 package. The initial energy level was set to 1 J and then compared to the generic ACK and pump-slowly fetch-quickly (PSFQ) protocols that are suitable for reliable transmission. To simplify the analysis, the traffic type was set as CBR, which generates packets periodically. The IEEE 802.15.4 package for NS-2 was used in the simulation. The maximum bandwidth was 250 kbps and the frames were transferred at a rate of 1 frame per second. The parameters used for energy consumption were the same as the sensor modes implementation using the CC2420 chipset [20] specification shown in Table 6. The parameter values were chosen considering the chip rate and bit rate of each radiofrequency band [11, 14].

5.2. Results and Analysis. For reliability simulation, we compared LiReTa with ACK method and PSFQ scheme. These two methods are the most relevant because of using ACK/NACK message and focused on reliable transmission. We choose these methods due to error recovery and the most well-known scheme in WSN. At first, we compare ACK method in view of confirmation for reliability. PSFQ helps to confirm LiReTa has good performance related to transmission time and total number of transmission.

For energy consumption simulation, we compared LiReTa with ACK method and ReInForM scheme. In related work, most of reliable transmission scheme focused on energy efficiency by reason of WSN nature.

5.2.1. Reliability without Retransmission Limitation. In this section, we present the results of transmission reliability simulations. We counted the number of transmissions until one packet successfully transmitted with no retransmission limitations in the network. The total number of transmissions, including retransmission, was used as a measurement of transmission reliability. The maximum number

Table 6: Parameters for power consumption analysis.

Description		868 MHz	915 MHz	2.4 GHz
Power consumption	P_{DOZE}		426 μA	
	P_{Receive}		19.7 mA	
	P_{Send}		17.4 mA	

of transmissions was set to six times per packet (i.e., 5 retransmissions and 1 initial transmission).

The first simulation shows the total number of transmissions including retransmission for 3 protocols: generic ACK, PSFQ, and LiReTa based on a gradual increment in the number of nodes. Therefore, Figure 15 indicates the total number of transmissions. Including retransmission, when the initial CER was 30% and the number of nodes increased from 0 to 100. The hop-by-hop recovery method used by the PSFQ scheme showed the highest total number of transmissions when the number of nodes exceeded 50 hops while ACK and LiReTa transfer methods were relatively low. However, these three methods were similar within a 10% tolerance until 30 hops. Importantly, our proposed scheme maintained a maximum number of transmissions that was 5–10% lower than the ACK method via a reduction in unnecessary traffic as well as by applying path reliability data and utilizing confirmation messages when path quality was high.

In a simulation with no limitations on the number of retransmissions, the success rate of the proposed method was similar to that of ACK with respect to guaranteeing data transfer reliability.

In the second simulation, we used time to indicate the total number of transmissions. We assumed that the 10 nodes settled for 60 minutes. Thus, the adaptive nature of our algorithm was highlighted compared with other methods. Figure 16 shows similarly reliable performance for our algorithm, although the ACK mechanism exhibited more reliable performance than the PSFQ mechanism. Initially, PSFQ conducted quick error recovery; however, after 40 minutes, the LiReTa method resulted in a reduction in retransmissions of 25% compared to PSFQ and 7% compared to the generic ACK scheme. The buffer overflow problem occurred within the PSFQ scheme due to buffering for data storage in middle nodes until error recovery was completed. Thus, ACK and LiReTa yielded a decreased number of transmissions because these schemes do not accumulate data in buffers.

Moreover, using the same initial error rate, LiReTa showed slightly better performance than the generic ACK method.

Figure 17 shows the total number of transmissions with respect to increasing packet size. In this simulation, the ACK method did not consider packet size while PSFQ provided fast recovery. Initially, these three methods show similar result for the packet size. Therefore, the previous PSFQ simulations exhibited better performance than the ACK method and were similar to the proposed scheme. However, even though LiReTa showed an incremental increase in error rate as the packet length increased, it still had the best performance. Over than 100 bytes packets, PSFQ is increased rapidly due to its recovery. 7% lower than ACK method and 24% lower than PSFQ when packet size is 150 bytes. LiReTa shows much better performance when packet size is bigger, for example, 11% more than ACK method and 38% more than PSFQ in 300 bytes packet.

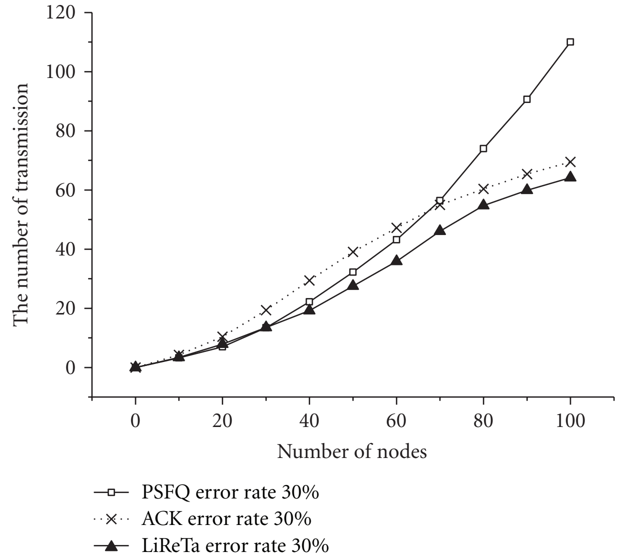


Figure 15: Total number of transmissions for an increasing number of nodes from 0 to 100.

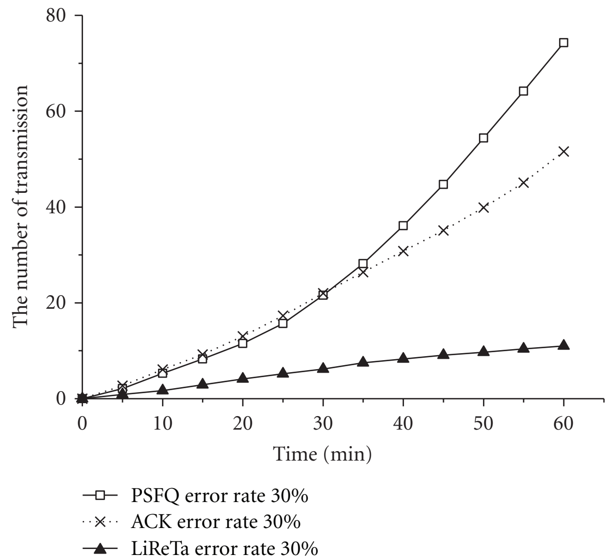


Figure 16: Cumulative transmissions for times between 0 and 60 min.

Figure 18 shows the total number of transmissions with 5 case path qualities in Table 7. It is similar results in three methods when path quality is bad (case 1). In case 5, when path quality is good, LiReTa shows the best performance than 51.2% of ACK method and 37.5% of PSFQ in case 5. The main concept of LiReTa is not send ACK message when data transfer with high path quality. This simulation means that our proposed scheme still maintains the number of

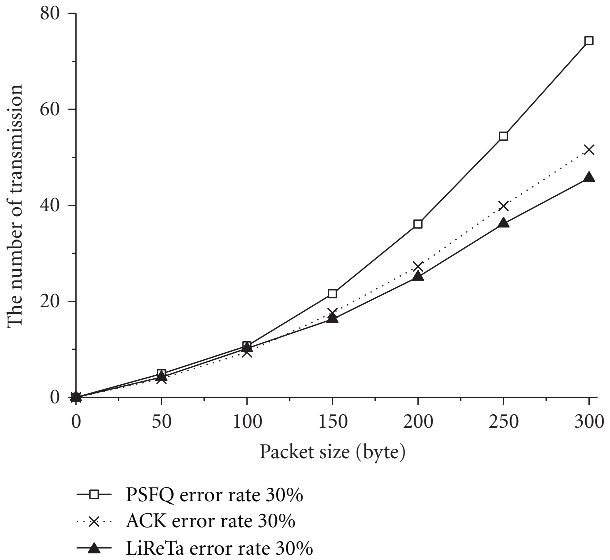


Figure 17: Total number of transmissions for packet sizes ranging from 0 to 300 bytes.

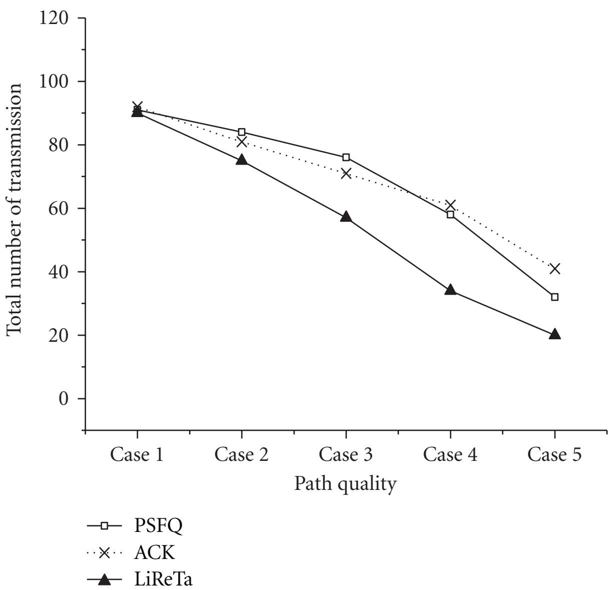


Figure 18: Total number of transmissions for path quality in 5 cases.

Table 7: Simulation conditions for path quality simulation.

Case	Path quality
1	RSSI −75 CER 75
2	RSSI −55 CER 55
3	RSSI −45 CER 45
4	RSSI −25 CER 25
5	RSSI −5 CER 5

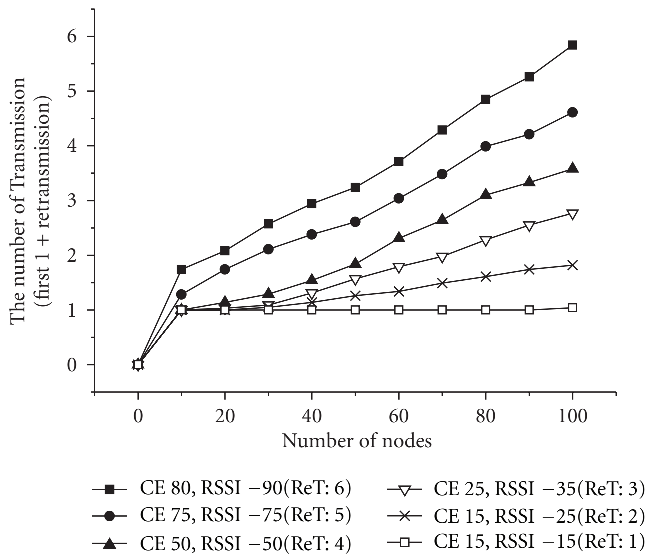


Figure 19: Average transmission time for each number of nodes for the six conditions in Table 8.

Table 8: Simulation conditions and number of minimum required transmissions for six simulations.

No.	Simulation conditions	Number of minimum transmissions required RT_i
1	CER 15, RSSI −15	1
2	CER 15, RSSI −25	2
3	CER 25, RSSI −35	3
4	CER 50, RSSI −50	4
5	CER 75, RSSI −75	5
6	CER 80, RSSI −90	6

transmissions lower than ACK method and PSFQ reducing unnecessary traffic by applying path reliability and none confirmation message if path quality is high.

5.2.2. Fault Tolerance with Limited Number of Retransmissions. The goal of the simulation shown in Figure 19 was to examine the reliability of the required number of retransmissions for a period of time according to the node proposed using RSSI and CER in Section 3.5. Based on the power consumption needed for a given network time as defined in Table 6, we proceeded to validate our results from Section 3.5. Specifically, to identify data transfer success, the number of sent packets was compared to the number of received packets in a certain period of time.

The conditions of the simulation are defined in Table 8. The number of nodes ranged from 0 to 100 and data were transferred in a network with a number of nodes that increased at a rate of 10 nodes per trial, that is, 0, 10, 20 · · · 100. The maximum number of transmissions in each case was 100, as shown in Figure 19.

The simulated numbers of transmissions did not exceed the number of minimum transmission times required RT_i. Therefore, the number of transmissions required and the matching table calculation proposed in Section 3.5 (Figure 7) was shown to be valid within the proposed scheme.

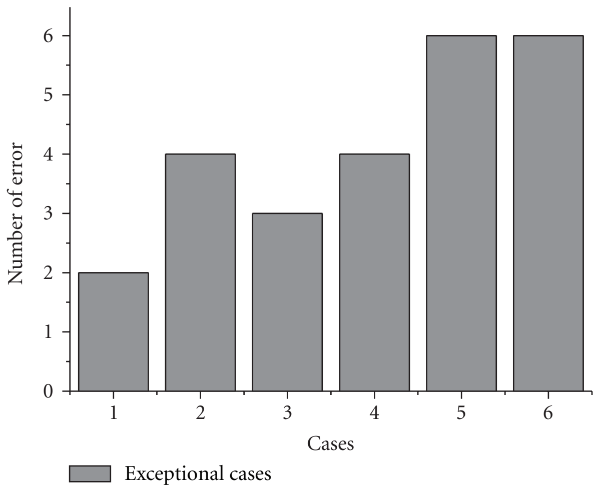


Figure 20: RT_i < Transmission time for actual simulations of 6 conditions (Table 8).

Fault tolerances are shown in Figure 20. These unusual cases exceeded the number of minimum transmissions required compared to RT_i values; however, the proportion was extremely low. Therefore, the average error rate was 4.1%, with a minimum error rate of 2% and a maximum error rate not exceeding 6%. This shows that reliability calculation with average plus standard deviation works well.

5.2.3. Energy Consumption Comparisons. To measure the energy consumption of each algorithm, we used the average consumed energy (J) of each node. The basic energy model in NS-2 applies different energy consumption levels except for other energy consuming stages such as routing. The initial energy was set to 1 J, idle electricity power to 5 μW, consumed power when receiving messages to 1.8 mW, and consumed power when sending messages to 27 mW.

Total Energy Consumption Compared with no Limitations for the Number of Transmissions. Figure 21 shows the energy efficiencies for error rates of 0%, 30, 60, and 98% when applied for ACK, ReInForM and LiReTa algorithms. The simulation results showed that the reliability of the proposed LiReTa scheme is as good as the other methods in terms of both reliability and energy efficiency. The ACK method energy consumption was set at 100% for the relative comparison between ACK, ReInForM and LiReTa. Our results showed that LiReTa had a 27% performance improvement while ReInForM had only a 20% performance improvement when the error rate was 0%. Moreover, when the error rate increased to 30%, the performance of LiReTa was improved by 29% while ReInForM had a 15% reduction in performance.

Therefore, when the CER was high, the number of retransmissions increased rapidly when using the generic ACK method. This was also the case for ReInForM due to the incremental increase in multi-paths needed to maintain reliability. On the other hand, LiReTa reduced the number of ACK messages by employing selective ACK and ACK_{dir}

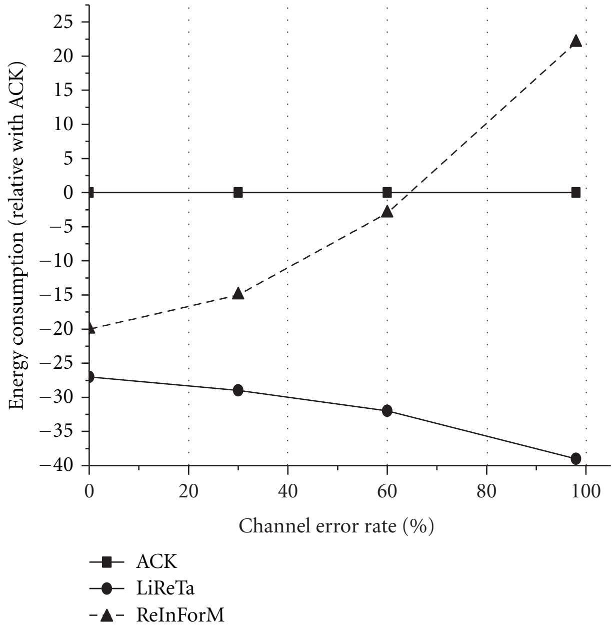


Figure 21: Energy consumption.

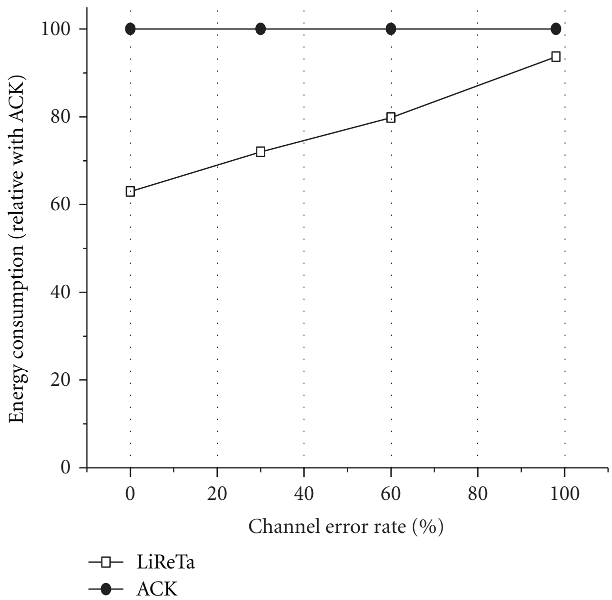


Figure 22: Energy consumption with a transmission limit of 4.

delegations. Hence, when the error rate was 60%, LiReTa had a 35% energy savings, which was even better than the ACK method, which had an average energy saving of 25%.

Energy Consumption Comparisons with Number of Retransmission Limitations. Moreover, we experimented with the initial error rate by setting it to 0%, 30%, 60%, and 98% in order to evaluate energy consumption when the transmission limit was initially set to 1 and the retransmission limit to 3 for a total of 4 transmission attempts. In this analysis, RSSI was set to −30% and the generic ACK method

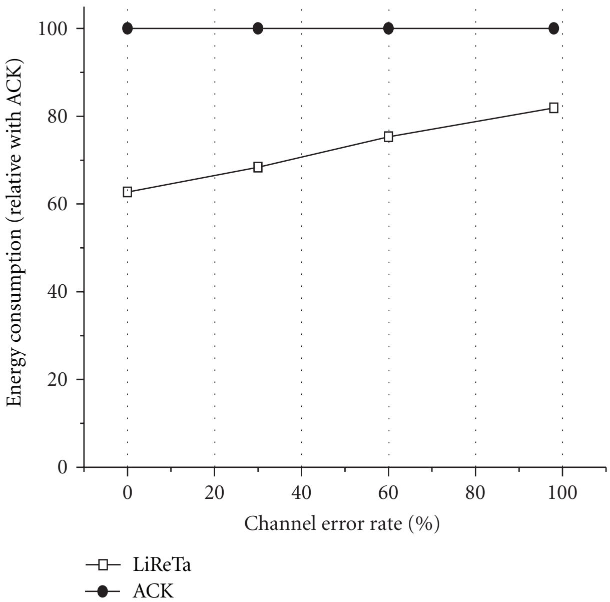


Figure 23: Energy consumption with a transmission limit of 6.

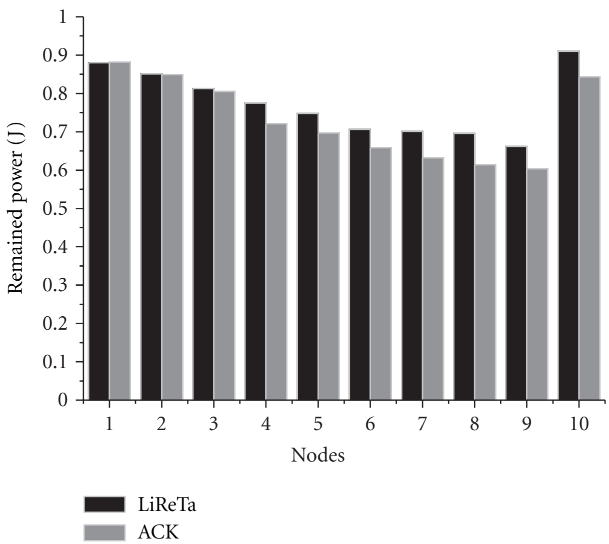


Figure 24: Remaining amount of energy in 10 nodes.

energy consumption was set to 100%. Figure 22 shows the proposed scheme employing overhearing and delegation with limited retransmissions. In this example, the energy consumption was reduced by up to 37% when the initial error rate is low. However, while the error rate increased, the ratio of ACK_{dir} increased substantially, up to 100%. These results showed similar energy consumption rates among the different methods analyzed when the error rate was near 100%, which was due to ACK_{dir} being used as many times as the generic ACK method. However, the LiReTa method showed more efficiency for larger retransmission limits as well as a lower initial error rate compared to the generic ACK method.

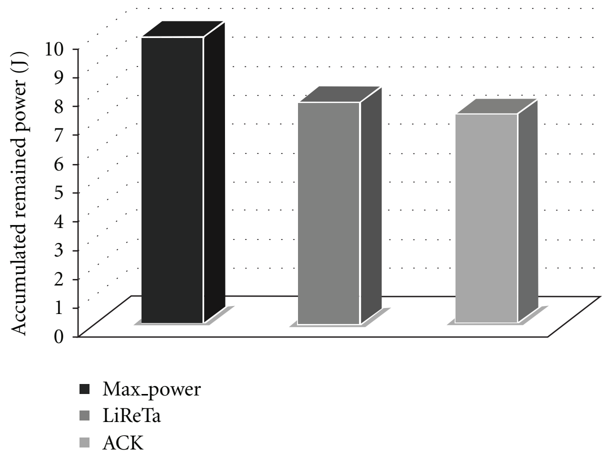


Figure 25: Total network lifetime.

Figure 23 shows the results of another experiment with the same conditions described in the previous experiment with the exception of a different total number of transmissions, 1 initial transmission and 5 retransmissions for a total of 6 transmissions. With 5 default retransmissions set in the MAC layer, the results of this experiment showed similar throughput to the case without retransmission limits. Further, the utilization of ACK_{dir} gradually increased because most of the ACK_{dir} were used while the number of transmission limits was replaceable with ACK_{imp}. This simulation showed that our scheme offers an average energy consumption rate that is 28% lower than the generic ACK method.

Network Lifetime Comparison. For network lifetime comparisons, experiments were performed over a linear topology comprised of 10 nodes and an initial CER set at 20%. We defined network lifetime as the total remaining power level in each node. Thus, we simulated the amount of energy remaining in nodes after 100 minutes, which we defined as network lifetime, as shown in Figure 24. The total amount of energy remaining is shown in Figure 25. The proposed scheme exhibited a 26% longer network lifetime compared with the generic ACK method.

Figure 26 shows two approaches for our proposal, namely, the LiReTa ACK_{imp}-based selective ACK method (IS-LiReTa) and the implicit ACK and delegation-based selective ACK method (IDS-LiReTa), where RSSI is fixed to −30 with variable error rates. The results show that IS-LiReTa and IDS-LiReTa had 22% and 30% higher energy efficiency compared with ACK, respectively. Finally, IDS-LiReTa, which utilizes delegation, exhibited a 35% performance improvement when the error rate was over 50%.

6. Conclusions

Since the power consumption of sensor nodes is highly influenced by data transmission, data loss must be minimized. Here, we present the LiReTa mechanism on WSNs for efficient data transmission. It considers network lifetime and

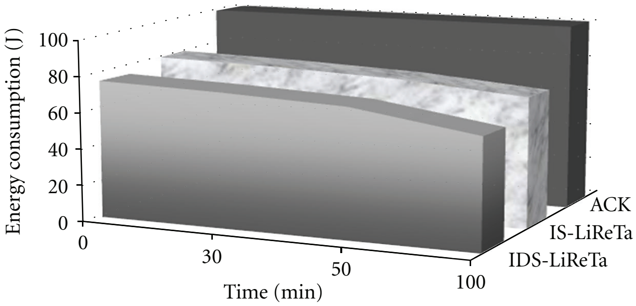


Figure 26: Energy consumption of generic ACK, IS-LiReTa, IDS-LiReTa protocols.

energy efficiency to support reliability for QoS requirements. To achieve reliable transfer in WSNs, several scenarios were discussed, implemented, and tested.

The contributions of this paper are threefold. First, we developed reliability calculations for each path. Second, we used the overhearing problem as an ACK_{imp} for supplementary purposes when the path quality was good in order to avoid ACK_{dir}. Last, we developed a selective mechanism to delegate ACK mechanisms that reduces the ACK-implosion problem.

In addition, we extended a reliability calculation method that uses only channel error rate in the previous research to a process that employs RSSI values and calculates reliability for each individual node path. Energy and traffic waste were reduced. The buffer overflow caused by error recovery using middle node buffers, as shown in NACK and PSFQ [3], was also reduced by using the generic ACK method as a base.

The performance evaluations of this method employed the NS-2 tool to analyze reliable data transfer. Hence, the proposed scheme is efficient for reliable data transfer in WSNs and offers a new method to reduce overhead, thereby improving energy consumption. Indeed, the proposed scheme exhibits increased energy efficiency when the initial error rate is high.

In future work, we will consider presented LiReTa scheme in different network densities, random form of sensor nodes and large-scale scenarios with big data. Proposed method can be affected by any routing mechanisms; therefore, find a suitable routing algorithm can be one of key way to maximize the performance. In addition, we will gather more prerequisite elements to achieve accurate reliability and our research will focus on the impact of reliability in WSN data transfer.

Acknowledgments

This work was supported by a Grant from the Kyung Hee University in 2011 (KHU-20111209) and the MKE (The Ministry of Knowledge Economy), Republic Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2012-(H0301-12-1004).

References

[1] S.-J. Park, R. Vedantham, R. Sivakumar, and I. Akyildiz, "GARUDA: achieving effective reliability for downstream communication in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 2, pp. 214–230, 2008.

[2] H. Yang, T. S. Park, C. Y. Park, and C. I. Jung, "Selective unacknowledged transmission in IEEE 802.15.4 considering energy efficiency," *Journal of Information Scientists and Engineers*, vol. 16, no. 6, pp. 631–751, 2010.

[3] C. Y. Wan, T. A. Campbell, and L. Krishnamurthy, "Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 862–872, 2005.

[4] J. Rousselot, A. El-Hoiydi, and J. D. Decotignie, "Performance evaluation of the IEEE 802.15.4A UWB physical layer for body area networks," in *Proceedings of the 12th IEEE International Symposium on Computers and Communications (ISCC '07)*, pp. 969–974, 2007.

[5] B. Deb, S. Bhatnagar, and B. Nath, "Information assurance in sensor networks," in *Proceedings of the 2nd ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 160–168, 2003.

[6] B. Deb, S. Bhatnagar, and B. Nath, "ReInForM: reliable information forwarding using multiple paths in sensor networks," in *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks*, pp. 406–415, 2003.

[7] B. H. Lee, H. W. Yoon, J. H. Park, M. Y. Chung, and T. J. Lee, "Reliable transmission using intermediate relay node-based transmission for reliability, sensor network," *Journal of Korea Information*, vol. 30, no. 9, pp. 850–857, 2005.

[8] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, "ESRT: event-to-sink reliable transport in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 1003–1016, 2005.

[9] C. Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: congestion detection and avoidance in sensor networks," in *Proceedings of the ACM SenSys Conference*, pp. 266–279, 2003.

[10] J. W. Choi and K. H. Lee, "A routing mechanism to prolong the lifetime of error-prone wireless sensor networks," *Journal of the Institute of Electronics Engineers of Korea*, vol. 46, no. 8, pp. 81–85, 2009.

[11] K. S. Yim, J. Kim, and K. Koh, "An energy-efficient routing and reporting scheme to exploit data similarities in wireless sensor networks," *Lecture Notes in Computer Science*, vol. 3207, pp. 515–527, 2004.

[12] S. J. Park and R. Sivakumar, "Congestion-aware topology controls for wireless multi-hop networks," *Ad Hoc Networks*, vol. 8, no. 3, pp. 295–312, 2010.

[13] K. I. Kim, "Sensor Network Technology-Analysis of Technology regarding energy efficiency issue," Report of KISTI Information Analysis team, 2008.

[14] Y. D. Yoo, J. H. Choi, and N. Kim, "Power consumption analysis of sensor node according to beacon signal interval in IEEE 802.15.4 wireless star sensor network," *Journal of Korea Information and Communications Society*, vol. 31, no. 9, pp. 811–820, 2006.

[15] P. Rand, "What's required for RF4CE, Resource document. The global electronics engineering community," 2008, http://www.eetasia.com/ART .

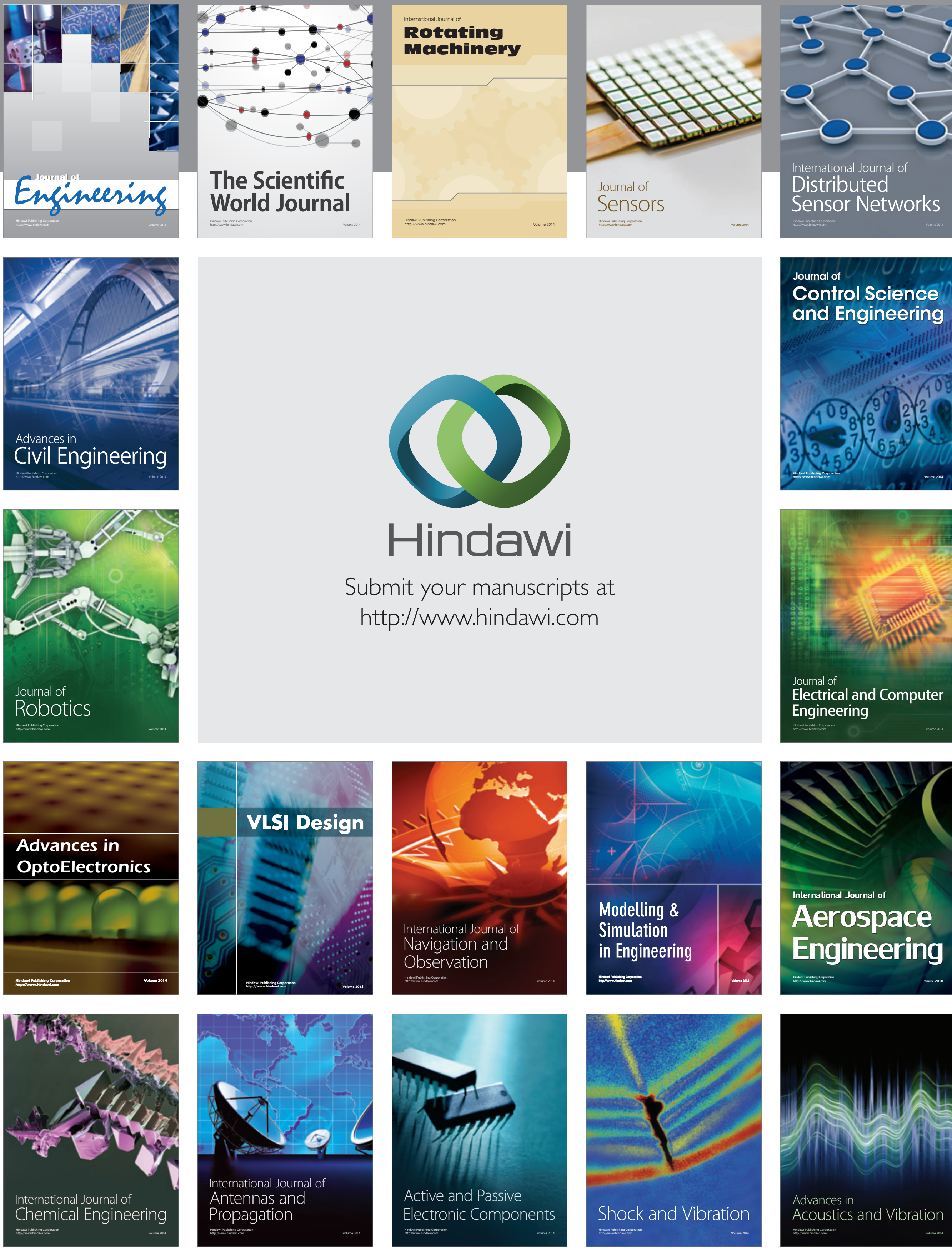
[16] Y. Tscha, "A reliable broadcasting scheme for directional antenna-based Ad Hoc networks," *Journal of Korean Institute of information Technology*, vol. 6, no. 6, pp. 66–74, 2008.

[17] Technologies and Savo G. Glisic, *Advanced Wireless Communications and Internet: Future Evolving*, John Wiley & Sons, 2011.

[18] H. Qayyum, S. Sohail, G. A. Shah, and A. Khanum, "A fuzzy decision maker for wireless sensor network in ubiquitous network environment," in *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops*, pp. 359–364, 2009.

[19] J. W. Choi, B. Y. Choi, S. J. Song, and K. H. Lee, "NQAR: network quality aware routing in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, Article ID 409724, pp. 224–233, 2010.

[20] Texas Instruments, "Single-Chip 2.4 GHz IEEE 802.15.4 Compliant and ZigBee-Ready RF Transceiver, CC2420 Datasheet," 2009, http://focus.ti.com/docs/prod/folders/print/cc2420.html.


Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
International Journal of Distributed Sensor Networks
Advances in Civil Engineering
Hindawi
Submit your manuscripts at
http://www.hindawi.com
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration