Advance Access publication on March 11, 2011
doi:10.1093/comjnl/bxr027

Fault Tolerance in Data Gathering Wireless Sensor Networks

Guangyan Huang[1]*, Yanchun Zhang[1], Jing He[1] and Jinli Cao[2]

[1] *Centre for Applied Informatics, School of Engineering & Science, Victoria University, Melbourne, VIC, Australia*
[2] *Department of Computer Science & Computer Engineering, La Trobe University, Melbourne, VIC, Australia*
**Corresponding author: abysshuang@gmail.com*

In data gathering wireless sensor networks, data loss often happens due to external faults such as random link faults and hazard node faults, since sensor nodes have constrained resources and are often deployed in inhospitable environments. However, already known fault tolerance mechanisms often bring new internal faults (e.g. out-of-power faults and collisions on wireless bandwidth) to the original network and dissipate lots of extra energy and time to reduce data loss. Therefore, we propose a novel Dual Cluster Heads Cooperation (CoDuch) scheme to tolerate external faults while introducing less internal faults and dissipating less extra energy and time. In CoDuch scheme, dual cluster heads cooperate with each other to reduce extra costs by sending only one copy of sensed data to the Base Station; also, dual cluster heads check errors with each other during the collecting data process. Two algorithms are developed based on the CoDuch scheme: CoDuch-l for tolerating link faults and CoDuch-b for tolerating both link faults and node faults; theory and experimental study validate their effectiveness and efficiency.

Keywords: wireless sensor networks; fault tolerance; data gathering; quality of information

Received 27 September 2010; revised 9 February 2011
Handling editor: Yu-Chee Tseng

1. INTRODUCTION

Recent advances in wireless communications and electronics technology have enabled the development of low-cost, low-power, small-sized wireless sensor nodes [1]. A large number of sensor nodes generally compose an Ad-Hoc Wireless Sensor Network (WSN) that collects useful data from the physical environment to the Base Station (BS) [1]. With the capability of achieving detailed information from the objective fields, WSNs have wide applications. Examples include:

- (i) monitoring the behavior of animals and people such as wild birds on remote islands, patients in hospitals and enemies in battlefields;
- (ii) detecting emergencies in broad areas such as fires in forests and torrent disasters on mountains;
- (iii) tracking moving objects in specific regions, such as vehicles in road networks and footballs in fields.

These applications require that the data, especially the data associated with critical unusual events [2], gathered by WSN, should be of high quality (e.g. quality of information [3–5] or quality of monitoring [6]) without important information loss, but it is inevitable that WSNs suffer from data loss due to random link failures and hazard node failures. Random link failures often happen, since sensor nodes have constrained radio capabilities (e.g. limited transmission distance and limited bandwidth) and densely deployed sensor nodes incur interference. Also, low-power or out-of-power failures happen frequently due to the non-rechargeable power supply [1]. Furthermore, sensors are generally deployed in inhospitable environments [7, 8] and random hazards cause sensor nodes to die before running out of power [9]. Naturally, data loss in the WSN is often reduced by tolerating link failures and node failures. However, many already known fault tolerance (FT) mechanisms generally dissipate lots of extra energy and time to detect and recover the failures and even use additional hardware and software resources [10, 11].

In this paper, we propose a novel Dual Cluster Heads Cooperation (CoDuch) scheme to reduce data loss and to minimize extra costs (e.g. energy and time) spending on FT. One advantage of CoDuch is that the data loss can be theoretically reduced from p to p^2 ($0 \leq p \leq 1$), where p is the link error rate. Another advantage is that the CoDuch scheme spends only a little more extra transmission cost than the single-CH method to gather data, taking advantage of the inherent broadcast nature of data transmission. The third advantage is that our CoDuch scheme can detect and tolerate both link faults and node faults.

To save costs, CoDuch adopts self-diagnosis methods that do not need extra hardware and software for FT. The most related work is 1+1 DHR (Dual Homed Routing) [12, 13], which used dual cluster heads to aggregate data and then to transmit data to the BS independently, but transmitting sensed data twice is energy-consuming. We use DHR to denote $1 + 1$ DHR throughout this paper. Our CoDuch scheme is totally different from DHR in two aspects: (1) dual cluster heads cooperate with each other to reduce extra costs by sending only one copy of sensed data to the BS; (2) dual cluster heads can check errors with each other while collecting data to the BS. Thus, CoDuch can tolerate both link faults and node faults. Experimental study shows that our CoDuch scheme outperforms DHR to reduce the data loss while expending less energy and time.

The remainder of this paper is organized as follows. In Section 2, we introduce preliminary and related work. In Section 3, we present the CoDuch scheme and its two implementations. In Section 4, we develop the dependability theory and analyze the energy dissipation for CoDuch. In Section 5, our experimental study based on simulation results validates the effectiveness and efficiency of the proposed algorithms. Finally, Section 6 concludes this paper.

2. PRELIMINARY AND RELATED WORK

In this section, we present the preliminary and related work for FT in WSN.

2.1. Preliminary work

A popular method to tolerate wireless link failures is retransmission, but we argue that retransmission may not suit WSNs. A disadvantage of retransmission is energy-consuming. Retransmission requires the receivers to confirm the receipt of messages; thus it consumes extra energy even if no retransmission is needed. We only allow as few as possible retransmissions in WSN due to the constraints of battery power supply. Another disadvantage is delay. If the number of retransmissions is great, then the data may be out-of-date and lose usefulness; this also brings great delays to the network to collect new data.

An effective approach (e.g. LEACH in [1]) avoids out-of-power node failures by maintaining fairness in the whole network: that is, to avoid the possibility of some sensors being out of power long before others. In this way, out-of-power failures only happen in a short period at the end of the network lifetime. This method can only postpone most out-of-power failures but cannot reduce the data loss when node failures happen.

Random node faults are more difficult to detect than out-of-power node faults since the former are unpredictable. But random higher level node failures are often the main reason for the data loss, since they lead to a group of data losses. Here, higher level nodes such as cluster heads (CH) in a cluster-based WSN generally are responsible for gathering data from a group of low level sensing nodes and then relaying aggregated data toward the BS. A scheme in [9] is provided to tolerate random node failures, where sensor nodes are assigned two statuses: active and sentry to save power. Sentry nodes perform as redundant nodes and become active only if active nodes fail, which periodically wakes up as scheduled to detect whether the active node has failed or not. However, this scheme incurs excessive additional costs on exchanging information between sentry and active nodes caused by frequently scheduled detection. Another limitation is that sentry nodes that are used especially for FT do not make the best use of node resources to collect data.

2.2. Related work

The FT techniques can be mainly classified into two groups: FT without detection and FT with detection. In FT without detection, redundant paths are often used to transmit data to destination, such as 1+1 DHR in [12, 13] and multi-path routing in [14], or to predict out-of-power faults in advance, such as 1:1 DHR in [12, 13]. However, it is energy-consuming to adopt the redundant paths and it limits the use of tolerating unpredictable node faults other than out-of-power faults.

FT with detection detects the faults through extra equipment or by the WSN itself. In a centralized fashion [15–21], BS is used to diagnose sensors but it is not feasible to require every sensor to contact BS directly. Another approach, such as that described in [10, 11] that explores additional powerful devices for fault detection is also not suitable since it requires additional traffic to be transported through WSN. In localized algorithms, collaboration [15, 19, 22, 23] or self-diagnosis [24] of sensor nodes are used to detect faults. In [8], the neighbor CHs help recover the faulty CHs; however, this requires special CHs that have more initial energy resources than ordinary sensors. In [25], CH failures are discovered by all the non-CHs by detecting its CH's status independently, but all the sensors in a local cluster must exchange information with each other and thus produce interference. In [9], random node failures are tolerated by redundant nodes but the extra traffic costs are high and the nodes' resources are wasted.

CoDuch uses FT without detection for tolerating link faults between non-CHs and dual CHs, and uses FT with detection for tolerating out-of-power and random node faults on dual CHs. It is feasible for the CoDuch scheme to adopt self-diagnosis in a distributed manner, thereby not requiring extra equipment or incurring extra traffic to maintain FT mechanism.

3. CODUCH SCHEME

In this section, we first provide an overview of the CoDuch scheme for cluster-based data gathering and then develop two algorithms for implementing FT.

3.1. Overview of CoDuch scheme

3.1.1. Cluster-based data gathering and difficulties of FT
We develop our CoDuch schemes based on the medium access control (MAC) in a popular cluster-based data gathering protocol, such as LEACH [1, 26], to save power. The main idea of the cluster-based data gathering protocol is as follows. First, each sensor node selects itself as a cluster head with a certain probability, which equals the ratio of the expected number of cluster heads to the total number of sensor nodes. Then, a sensor uses a carrier sense multiple access (CSMA) to transmit an advertisement on the CH status or a join-request message on non-CH sensors. Each non-CH sensor receives the advertisements from all CHs. The distance between a non-CH and a CH is computed based on received radio strength, and a non-CH chooses the nearest CH as its CH. The non-CH sensor uses a time division multiple access (TDMA) slot to send data to the cluster head with the DS-SS code specified by the cluster head. The advantage of this approach is that it allows nodes to remain in the sleep state as long as possible by using TDMA and also minimizes collisions by using DS-SS on top of TDMA or using CSMA. We suppose that all the sensor nodes are homogenous and they compose an Ad-Hoc WSN. This assumption accords with most of the applications of WSN. After a distributed cluster forming and scheduling setup, CHs begin to gather data from non-CHs and then locally fuse data by compressing multiple original readings into one [1, 27]. Data fusion on cluster heads is a good solution to reduce the number of remote transmissions between WSN and the BS.

Taking advantage of the above power saving MAC, we carefully design FT in the CoDuch scheme to satisfy our goal. For example, if a CH detects that a non-CH loses a packet, the CH cannot require the non-CH to retransmit data, since non-CH sensors are always in a sleep state to save 2/3 power (according to our analysis of LEACH code [26] implemented in the network simulator 2 (ns2) [28]) and do not wait for 'ACK' from CHs. Thus, all the FT mechanism must be scheduled in advance. However, frequently scheduled detection may cost a lot, like that in [9], since even if no faults exist, the detections are continuously carried out. Note that out-of-power faults can be predicted and thus they may be detected and tolerated by reasonable scheduling like 1:1 DHR in [12], where the primary CH notifies the backup CH to replace it when it is in low-power status. But link faults and random node faults happen unpredictably and thus it is more difficult to detect them.

3.1.2. CoDuch on routing layer
We mainly focus on the routing layer to develop our CoDuch scheme on top of the foregoing MAC layer. The method in our proposed CoDuch scheme for tolerating link faults between non-CHs and CHs is given as follows:

Step 1: Primary CH chooses a vice CH among non-CHs that has the highest remaining energy resource. Then, the primary CH advertises to other members within the cluster and sends a copy of TDMA scheduling list to the vice CH;
Step 2: Dual CHs (primary and vice CHs) receive data from non-CHs;
Step 3: The vice CH fuses the data it receives with its own data and then transmits the fused data to the primary CH. An 'ACK' is required to be sent from the primary CH to the vice CH;
Step 4: The primary CH checks whether the vice CH misses some packets, if so CH searches the loss data from its received list and then fuses new data and its own data with the vice CH's data;
Step 5: The primary CH relays data to the BS.

We ensure the reliability of the link between the vice CH and the primary CH by an 'ACK' message where the message type is 'ACK' and the message content is 1 or 0. If the message content is 1, vice CH can send the next data packet; otherwise, vice CH must retransmit the current data packet. Here, the 'ACK' scheme is reasonable since dual CHs do not sleep. No detection is adopted in the above method and two copies of data from non-CHs reduce the possibility of data loss. It dissipates around one time more energy for receiving and fusing non-CH data for tolerating link faults in the above method, as well as more energy and time spent on dual-CHs-based cluster forming than single CH scheme (e.g. LEACH).

To support tolerating out-of-power and random CH node faults and at the same time tolerating link faults, we detect node failures by the following steps:

Step 6: If vice CH tries to send data to the primary CH and does not receive 'ACK' within a threshold of time, then it decides to send data directly to the BS without the primary CH's data;
Step 7: If the primary CH does not receive any data from the vice CH when the time is overdue, it sends data to the BS without the vice CH's data.

The extra cost of a threshold of waiting time is spent on tolerating node failures. Note that if link failure happens between the vice CH and the primary CH, they both send data to the BS because dual CHs determine to send data independently, but the probability of this link failure is very low. We do not distinguish out-of-power node faults from random node faults because the mechanism of tolerating them is the same.

3.2. FT design in CoDuch

In this section, we first give the time line of the CoDuch scheme and then focus on the detailed algorithms of CoDuch-l for tolerating link faults and CoDuch-b for tolerating both link and node faults.

3.2.1. Time line of CoDuch
We assume that sensors are homogenous and all algorithms running on non-CHs and dual CHs are the same. We plot the

time line of the CoDuch scheme in Fig. 1, which comprises the set-up phase and the repeat phase. The FT mechanism is designed in the repeat phase, including three steps:

(i) FT1: check data from VCH,
(ii) FT2: check node error of VCH and
(iii) FT3: check node error of CH.

In the time line of CoDuch, dual CHs are selected in the set-up phase and both vice CH and CH receive data from non-CHs in the repeat phase, as well as FT consideration in the sub-FT phase at the end of the repeat phase. In a round, the set-up phase runs once and the repeat phase runs several times according to the threshold time of a round. For example, if one round equals to 20 s, suppose the set-up phase takes 2 s and one repeat phase takes 3 s, then the repeat phase repeats six times. There are mainly three steps in the time line of CoDuch ((1) and (2) for the set up phase and (3) for the repeat phase):

(1) *Cluster forming.* Sensor nodes that do not take the role of CHs recently select themselves as CHs randomly and broadcast their CH status [1]. Then all the non-CH sensor nodes know the CHs and choose the nearest one to join. CHs receive the join request message sent by non-CHs and keep a queue of the list of joined non-CH IDs as a TDMA scheduling queue.
(2) *Vice CHs selection.* CH chooses the non-CH with the highest energy as the vice CH and puts the vice CH ID at the end of the TDMA scheduling queue. CH broadcasts TDMA scheduling list and at the same time advertises the vice CH's status. Then the basic information (e.g. CH ID, vice CH ID and TDMA Schedule List of Node IDs) is backed up from CH to vice CH. Also, non-CHs keep their CH ID and vice CH ID.
(3) *Repeat phase.* Both the vice CH and CH receive data simultaneously from non-CHs according to the same TDMA list that has allocated a time slot for each non-CH in advance. Suppose non-CHs can adjust the transmitting power according to the maximum value of the distance to CH and the distance to vice CH in order to ensure that both of them receive the data. Dual CHs may not be located at the centre of the cluster, since we choose nodes with more remaining energy as dual CHs to balance energy dissipation for achieving longer network lifetime. The distance between non-CH and any CH (or vice CH) can be achieved by measuring the signal strength when non-CH receives the status broadcast from dual CHs. After finishing the data collection, the vice CH fuses data into one packet and then sends the fused data (with a representative node ID list, which denotes the original data comprised in the fused data) to the CH. The CH is waiting for data in the idle phase and once it receives data from vice CH, it fuses with new data including received data and its own into one packet and sends this to the BS. The repeat phase runs several times until a round is over.

3.2.2. *Algorithms of FT in CoDuch*

We develop two algorithms: CoDuch-l for tolerating link faults and CoDuch-b for tolerating both node faults and link faults.

(1) *CoDuch-l for Link Faults Tolerance*

The main idea of CoDuch-l is as follows: non-CHs broadcast one packet data sensed from the physical environment, which is received by both the vice CH and the CH. The vice CH fuses the received data with its own into one packet, which is sent to the CH together with the representative ID list of the fused data. In the phase FT1: check data from VCH, the CH checks the representative ID list to determine whether there are some loss packets. If a loss packet is found, the CH looks for it by searching its own received data list. Finally, after fusing the VCH's data with both the loss data it has found and its own data into one packet, the CH relays the data to the BS. We can see that

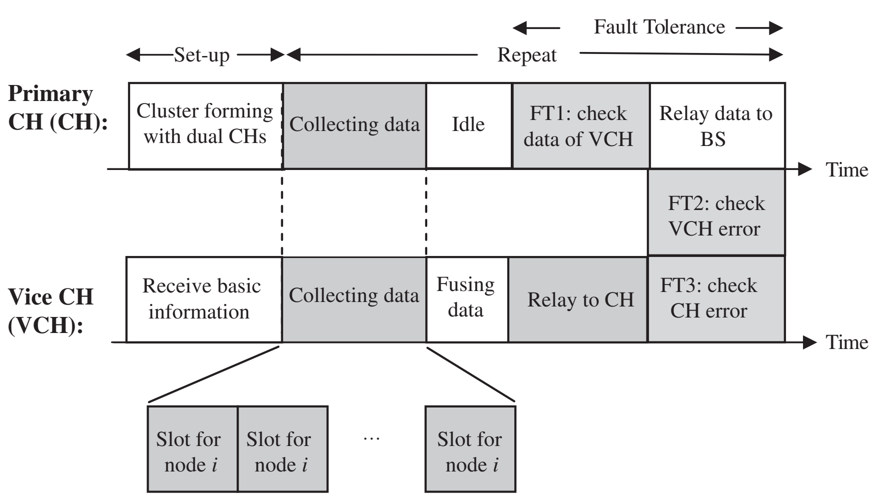


FIGURE 1. Time line of CoDuch scheme in a round.

the dependability can be improved from $1-\varepsilon$ to $1-\varepsilon^2$, where ε is the data loss rate, by tolerating the link faults of non-CHs.

(2) *CoDuch-b for Node Faults Tolerance*

The more complicated case is that both node faults and link faults happen in WSN; since if node faults happen on the vice CH and the CH then it may stop the function of tolerating of link faults by dual CHs. Figure 2 shows the flowchart of the FT mechanism for both link failures and node failures in CoDuch-b. Each sensor node runs CoDuch-b independently. If it is a non-CH node, it will do nothing. If it is a VCH, it will do FT3: by checking at time T1 whether node failure happens on the CH. If it is a CH, there are two cases: (1) if the CH receives data from VCH, then it goes on to check whether data are lost or not. If data are lost, it will do FT1: by finding lost data in its own data list and fusing the new data with VCH's data and then sends data to the BS; otherwise, it fuses VCH's data with its own and sends data to the BS; (2) if the CH fails to receive data from the VCH, then it will do FT2: by checking at time T2 whether node failure happens on the VCH. Note that FT1, FT2 and FT3 in Fig. 2 are in accordance with those in Fig. 1.

The key problem is to determine T_1 and T_2 in Fig. 2. Suppose that the repeat phase begins at time T_0 and that VCH is expected to send data to the CH at time $T_0+m\times s$, where $m = N/N_{\mathrm{CH}}-2$ (N is the total number of sensors in WSN and N_{CH} is the number of clusters) is the average number of non-CHs in the cluster and s is a slot of time allocated to each non-CH for transmitting data to dual CHs. Then, the CH is expected to receive data from VCH at time $T_0+(m+1)\times s$. Therefore, FT1 generally runs after time $T_0+(m+1)\times s$. After FT1, we also set the scheduling of FT2 and FT3 in advance to make both of them run at the near future time of $T_1 = T_2 = T_0 + f - \tau$, where f is the duration for the repeat phase that runs once and τ is a slot of time to ensure that all CHs and vice CHs send data to the BS one by one in a random access. Dual CHs retransmit data to the BS if link faults happen.

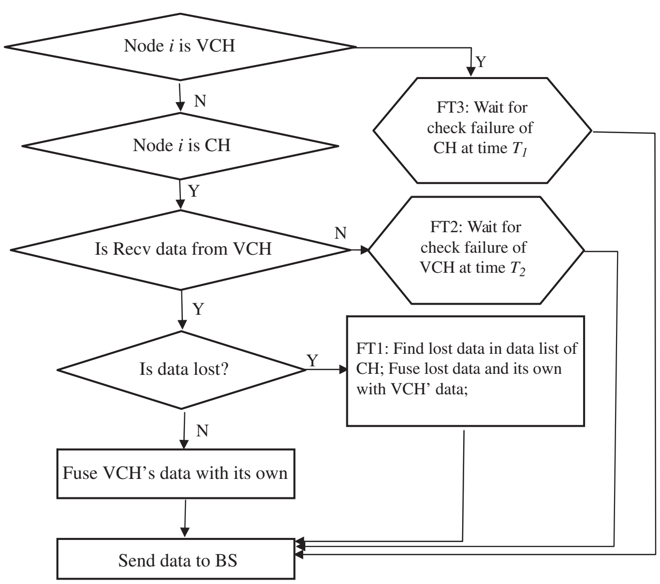


FIGURE 2. Flowchart of CoDuch-b.

Step FT1 is designed for tolerating link faults. Steps of FT2 and FT3 are designed to detect and tolerate node faults. If we remove FT2 and FT3 in Fig. 2, the flowchart is changed into another flowchart for the algorithm of CoDuch-l. Since steps of both FT2 and FT3 bring more delay and are designed especially for node faults, if only to tolerate link faults, CoDuch-l performs better in terms of energy consumption and time. The major advantages of this FT design include two aspects:

(i) there is no recovery time due to the distributed property. Once the fault is detected, the redundant data are available at hand to recover the faults;
(ii) it is a self-maintained scheme and new faults of vice CH can also be tolerated.

4. DEPENDABILITY AND ENERGY DISSIPATION ANALYSIS

In this section, we develop a dependability theory and analyze the energy dissipation for CoDuch.

4.1. Dependability analysis

There are two types of dependability: internal dependability and external dependability. The goal of this paper is to improve both external dependability and internal dependability in CoDuch and thus to achieve the lowest data loss rate.

4.1.1. Internal dependability

Internal faults may occur even if no outside faults are injected, and thus the dependability of the WSN is $<$ 100% (examples are shown in Fig. 3). There are three aspects that may cause internal faults, which are as follows:

Factor 1: Out-of-power faults (or low-power faults) may cause data loss in an uncertain Ad-Hoc WSN, though the fairness scheme in LEACH theoretically postpones the time of death of most nodes toward the end of network lifetime. Suppose a Cluster Head or vice Cluster Head

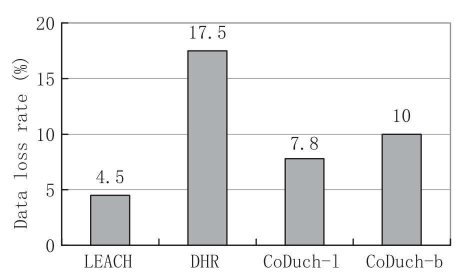


FIGURE 3. Internal dependability test.

dies when it has collected data from non-cluster head sensor nodes but does not send to the BS; thus the data are lost.

Factor 2: There is still some data loss due to collision and interference [14] that happens in a WSN with high sensor density now that sensor nodes generally collaborate with each other independently in a distributed manner, even if DS-SS and TDMA or CSMA are explored to minimize collision. For example, when CHs or vice CHs transmit data to the BS, they adopt CSMA to sense the channel before transmission. If the channel is currently busy, the node sets a back-off timer to expire after a random amount of time. This back-off policy for CSMA is effective when all nodes are transmitting packets with the same length. However, if link errors are injected into the WSN, CHs/ vice CHs may have different numbers of transmitting packets. Also, retransmitting happens on the link between CHs/vice CHs and the BS worsens the collision, because non-CHs are sleeping and are scheduled in advance; so they will send data as scheduled to CHs/vice CHs, but if CHs/vice CHs are busy, the data packets are lost.

Factor 3: Faults detection mechanisms more or less would impact on the original protocols and thus break the optimal balance of energy dissipation and transmission scheduling. For example, the cost of FT is often spent on dissipating extra energy, which may lead to unexpected data loss caused by out-of-power faults. Also, tolerating both link and node faults leads to more complex detection methods. Thus, FT mechanisms may worsen Factors 1–2.

For more detailed techniques about data packet loss due to power faults and collisions, refer to Appendix A in [23].

We can see from Fig. 3 that internal dependability of our proposed CoDuch-l (CoDuch-b) with the data loss rate of 7.8% (10%) are worse than that of LEACH with a data loss rate of 4.5%, since FT mechanisms in CoDuch incurs more internal faults due to Factor 1. However, LEACH cannot perform well when external faults are injected (we explain this in Section 4.1.2). Also, Fig. 3 shows that the internal dependability of our CoDuch scheme is better than DHR (with a data loss rate of 17.5%), because CoDuch results in less negative impact on the original network than DHR.

4.1.2. *External dependability*

Figure 4a–c illustrate the error models in LEACH, DHR and CoDuch, respectively; we mainly compare CoDuch with DHR as follows:

(i) There is a link from VCH to CH. This link means VCH and CH would cooperate with each other to ensure a more dependable data transmission between the WSN and the BS.

(ii) The link from VCH to the BS denoted by the dash arrow means VCH would transmit gathered data to the BS only if CH fails. Thus, in most cases, only one transmission between CH and the BS reduces the energy consumption.

To simplify the problem, two types of external faults: link failures and higher level node (e.g. CH/VCH) failures are abstracted in Fig. 4, where $\varepsilon(0 \le \varepsilon < 1)$ is the link error rate and $\mu(0 \le \mu < 1)$ is the node error rate. Suppose both link faults and node faults happen independently. ε^n denotes maximum of n times retransmission on the link. S_i denotes the non-CH sensor node, which does not retransmit data. We suppose that link failures happen uniformly and randomly with error probability, ε, given by the user and node error rate (the average probability that a node failure happens at any time) is given by

$$\mu = m/(\mathrm{MTTF} \times N_{\mathrm{CH}}), \tag{1}$$

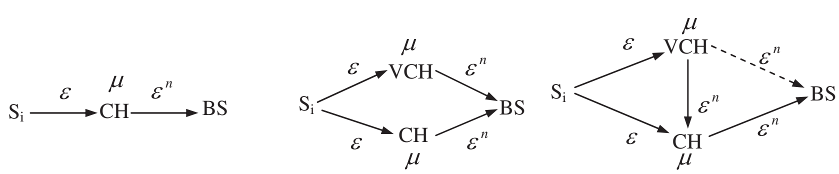
where m is the total number of CH/VCH nodes that fail due to random hazards within the network lifetime, N_{CH} is the number of CH/VCH nodes in each round and MTTF is the mean time to failure during which a node fails following Poisson distribution. Users can determine MTTF and m, and we will discuss the details of the method in Section 5. Suppose there is only one unit data on S_i, in Fig. 4a, the external dependability of LEACH is

$$E_{\mathrm{LEACH}} = (1-\varepsilon)(1-\mu)(1-\varepsilon^n). \tag{2}$$

In Fig. 4b, the external dependability of DHR is

$$E_{\mathrm{DHR}} = 1-(1-(1-\varepsilon)(1-\mu)(1-\varepsilon^n))^2 = 1-(1-E_{\mathrm{LEACH}})^2. \tag{3}$$

It is more complicated to compute the dependability of CoDuch, and thus we transfer Fig. 4c to an equivalent model denoted by



(a) LEACH error model. **(b) DHR error model.** **(c) CoDuch error model.**

FIGURE 4. Dependability models.

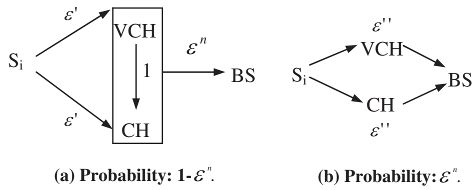


FIGURE 5. Equivalent CoDuch error model: (a)+(b). $\varepsilon' = 1-(1-\varepsilon)(1-\mu)$ and $\varepsilon'' = 1-(1-\varepsilon)(1-\mu)(1-\varepsilon^n)$.

two cases shown in Fig. 5. Figure 5a shows the first case that the link between VCH and CH never fails and Fig. 5b shows the second case that the link failure between VCH and CH happens. The probability of the first case is $1-\varepsilon^n$, where S_i sends data to VCH and CH and we can take VCH and CH as one receiver with the dependability of $1-\varepsilon'^2$ to collect data from S_i. Then, this virtual receiver sends data to the BS with the dependability of $1-\varepsilon^n$. Thus, the external dependability of CoDuch under this condition is

$$E_2 = (1-\varepsilon^n)^2(1-\varepsilon'^2). \quad (4)$$

The probability of the second case is ε^n and the external dependability of CoDuch is the same as that of DHR. That is

$$E_1 = \varepsilon^n E_{\mathrm{DHR}}. \quad (5)$$

Therefore, the external dependability of CoDuch is

$$E_{\mathrm{CoDuch}} = E_1 + E_2 \approx 1-\varepsilon'^2 = 1-(1-(1-\varepsilon)(1-\mu))^2. \quad (6)$$

4.1.3. Dependability theory

We can deduce the following theorems based on both internal dependability and external dependability. We model dependability of CoDuch, Q_{CoDuch}, by using

$$Q_{\mathrm{CoDuch}} = E_{\mathrm{CoDuch}} - I_{\mathrm{CoDuch}}, \quad (7)$$

where E_{CoDuch} is given in Equation (6) and I_{CoDuch} is the data loss rate due to internal faults of CoDuch.

THEOREM 4.1. *The dependability of CoDuch is greater than that of LEACH, if* $(1-\varepsilon)(1-\mu)(\varepsilon+\mu-\varepsilon\mu) > \alpha$, *where* $\alpha = I_{\mathrm{CoDuch}} - I_{\mathrm{LEACH}}$, I_{CoDuch} *and* I_{LEACH} *are the data loss rates due to internal faults of CoDuch and LEACH, respectively.*

Proof.

$$\begin{aligned} &Q_{\mathrm{CoDuch}} - Q_{\mathrm{LEACH}} \\ &\quad = (E_{\mathrm{CoDuch}} - E_{\mathrm{LEACH}}) + (I_{\mathrm{LEACH}} - I_{\mathrm{CoDuch}}) \\ &\quad = (1-\varepsilon)(1-\mu)(\varepsilon+\mu-\varepsilon\mu) - \alpha(\alpha > 0). \end{aligned} \quad (8)$$

□

For example, given $\alpha = I_{\mathrm{CoDuch}-l} - I_{\mathrm{LEACH}} = 0.078 - 0.045 = 0.033$ according to Fig. 3, let $\mu = 0$, if $\varepsilon(1-\varepsilon) > 0.033$, then $Q_{\mathrm{CoDuch}} > Q_{\mathrm{LEACH}}$.

THEOREM 4.2. *The dependability of CoDuch is greater than that of DHR, if* $\beta > \varepsilon^n(1-\varepsilon^n)^2(1-\varepsilon)^2(1-\mu)^2$, *where* $\beta = I_{\mathrm{DHR}} - I_{\mathrm{CoDuch}}$, I_{CoDuch} *and* I_{DHR} *are the data loss rates due to internal faults of CoDuch and DHR, respectively.*

Proof.

$$\begin{aligned} Q_{\mathrm{CoDuch}} - Q_{\mathrm{DHR}} &= (I_{\mathrm{DHR}} - I_{\mathrm{CoDuch}}) - (E_{\mathrm{DHR}} - E_{\mathrm{CoDuch}}) \\ &= \beta - \varepsilon^n(1-\varepsilon^n)^2(1-\varepsilon)^2(1-\mu)^2. \end{aligned} \quad (9)$$

□

Generally, $(E_{\mathrm{DHR}} - E_{\mathrm{CoDuch}}) \xrightarrow{n\to\infty} 0$. For example, according to Fig. 3, $\beta = I_{\mathrm{DHR}} - I_{\mathrm{CoDuch}-l} = 0.175 - 0.078 = 0.097 >> E_{\mathrm{DHR}} - E_{\mathrm{CoDuch}}$ and thus $Q_{\mathrm{CoDuch}} > Q_{\mathrm{DHR}}$.

4.2. Energy dissipation for FT

We use a radio energy model the same as that in [1], in which the energy dissipation $E_T(k,d)$ of transmitting k-bit data between two nodes separated by a distance of d meters is given as follows:

$$E_T(k,d) = \begin{cases} k(E_{\mathrm{elec}} + \varepsilon_{\mathrm{FS}} \times d^2)(d < d_0) \\ k(E_{\mathrm{elec}} + \varepsilon_{\mathrm{MP}} \times d^4)(d > d_0) \end{cases}, \quad (10)$$

where $d_0 = \sqrt{\varepsilon_{\mathrm{FS}}/\varepsilon_{\mathrm{MP}}}$, E_{elec} denotes electronic energy, $\varepsilon_{\mathrm{FS}}$ and $\varepsilon_{\mathrm{MP}}$ denote transmit amplifier parameters corresponding to the free-space and the two-ray models. The energy dissipation incurred in the receiver of the destination sensor node is

$$E_R(k) = k \times E_{\mathrm{elec}}. \quad (11)$$

Also, the energy dissipation of fusing k-bits data is

$$E_F(k) = k \times E_{\mathrm{df}}, \quad (12)$$

where E_{df} is the energy dissipation of fusing one bit data. The parameters used in this paper are given below: $E_{\mathrm{df}} = 5$ nJ/bit, $\varepsilon_{\mathrm{FS}} = 10$ pJ/bit/m^2, $E_{\mathrm{elec}} = 50$ nJ/bit, $\varepsilon_{\mathrm{MP}} = 0.0013$ pJ/bit/m^2 and $d_{\mathrm{toBS}} > d_0$.

To simplify the problem, we use d_{toCH} to denote transmission distance between S_i and CH (or VCH) since the intra-cluster distance is very small and the slight difference is neglected, as well as d_{toBS} to denote the distance between CH (or VCH) and the BS. Given that m is the number of sensor nodes in a cluster including VCH and CH, the energy consumption of CoDuch in most cases (with probability of $1-\varepsilon^n$) for each cluster in one round is

$$\begin{aligned} E_{\mathrm{CoDuch}} = (m-1)E_T(k, d_{\mathrm{toCH}}) + (2m-3)E_R(k) \\ + (2m-1)kE_{\mathrm{df}} + E_T(k, d_{\mathrm{toBS}}). \end{aligned} \quad (13)$$

The energy consumption of DHR for each cluster in one round is

$$\begin{aligned} E_{\mathrm{DHR}} = (m-2)E_T(k, d_{\mathrm{toCH}}) + (2m-4)E_R(k) \\ + (2m-2)kE_{\mathrm{df}} + 2E_T(k, d_{\mathrm{toBS}}). \end{aligned} \quad (14)$$

Considering the failure probability between the VCH and the CH, the exact energy consumption of CoDuch is

$$E'_{\text{CoDuch}} = (1 - \varepsilon^n)E_{\text{CoDuch}} + \varepsilon^n E_{DHR}. \tag{15}$$

Thus,

$$E_{\text{DHR}} - E'_{\text{CoDuch}} = (1 - \varepsilon^n) \times (E_T(k, d_{\text{toBS}}) - (E_T(k, d_{\text{toCH}}) + E_R(k) + kE_{\text{df}})). \tag{16}$$

According to the power model in WSN, $E_T(k, d_{\text{toBS}})$ is greater than $(E_T(k, d_{\text{toCH}}) + E_R(k) + kE_{\text{df}})$ since the distance between the WSN and the BS generally is far away. Therefore, CoDuch consumes less energy than DHR.

5. PERFORMANCE EVALUATION

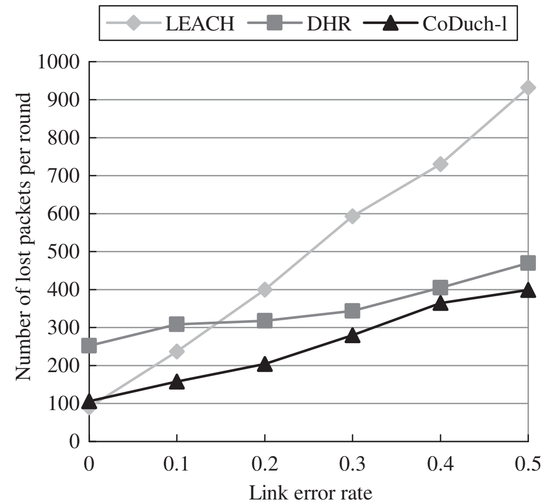
In this section, we demonstrate the effectiveness and efficiency of our CoDuch method including CoDuch-l for tolerating link faults and CoDuch-b for tolerating both types of faults by using two metrics: the number of lost packets per round and data loss rate. Then, we analyze the energy and time spent on improving dependability. All the experiments were run in the network simulator 2 (ns2) [28].

We now briefly describe our experimental setup as follows:

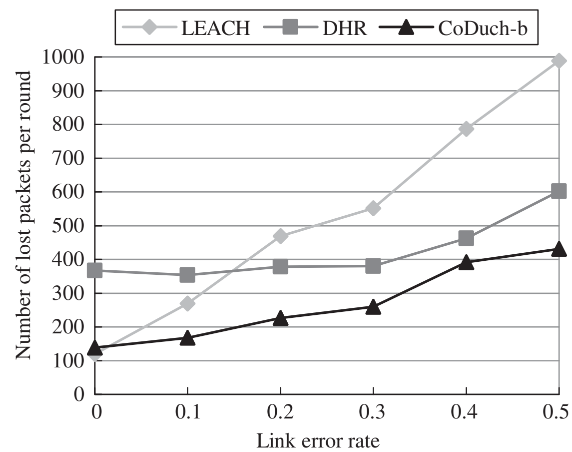
(1) We injected random uniform link errors and random Poisson distributed node errors into the WSN to study the performance of the three schemes. We injected link faults with the link error rates in the range [0, 0.5] to evaluate CoDuch-l and injected both node faults (totally 30%) and link faults with various link error rates in [0, 0.5] to evaluate CoDuch-b. The simulation of node failure was based on Poisson distribution with λ, where λ was the MTTF for node failures. Let $\lambda = 100$ and a total of 30% failed nodes during the whole network lifetime. In other words, fault injection time was set on each CH/VCH node based on Poisson distribution with MTTF $=$ 100 s. We use the algorithm given by Knuth in [29] with λ = MTTF to generate Poisson distributed random numbers, which are used as the node failure injection time. For example, if the Poisson random number is 98, then randomly selected m (m is less than the number of VCHs and CHs) node failures are injected at the 98th second. The actually total number of failed nodes was $n = 30 \pm 3$ in all the trials.

(2) The experiment parameters based on the power model in Section 4.2 are given as follows: the locations of $N = 100$ sensors are randomly generated by the tool in LEACH code in ns2 [28] in a square area S = 100 m × 100 m and we adopt the same parameters as the default setting in LEACH code in [28]: $N_{\text{CH}} = 5$, the initial energy resource of every sensor node, $E_0 = 2$ Joules and the duration of a round, $r = 10 \times E_0 = 20$ s. The BS is located at (100 m, 175 m). These parameters are the same in all the experiments.

(3) We plot our experimental results in Figs. 6–9 and each number on the curve is the average value of five trials. In a group of five trials, we injected link error/node error with the same rate and ran the WSN five times. In a trial, the WSN starts with the same initial energy resource on each sensor node and ends with no more than $N_{\text{CH}} - 1 = 4$ alive sensor nodes; that is, most sensor nodes that use up their energy resource are dead.



(a) Data loss in WSN with random link faults.

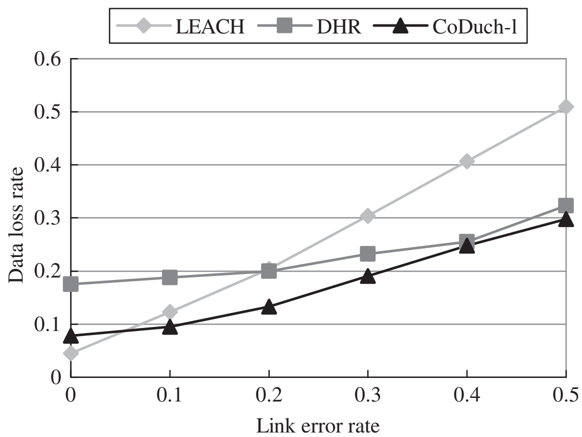


(b) Data loss in WSN with both random link faults and total 30% node faults.

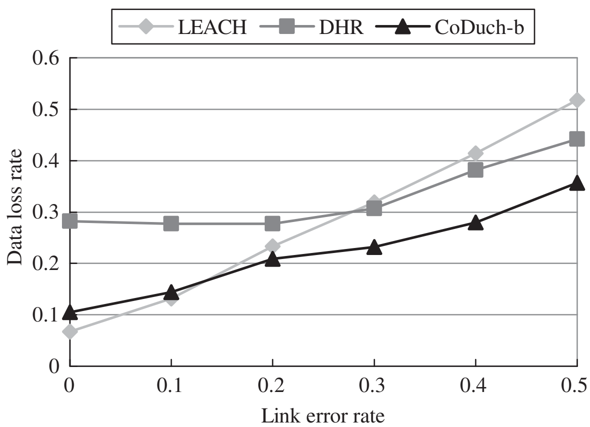
FIGURE 6. Data loss per round.

5.1. Effectiveness

In this subsection, we adopt two metrics: the number of lost packets per round and data loss rate to evaluate the effectiveness of our scheme. The number of lost packets per round, u, is defined as $u = (N_{\text{sensing}} - N_{\text{BS}})/L$, where N_{sensing} is the total number of sensed data packets, N_{BS} is the total number of data packets the BS has received and L is the total number of rounds (or the network lifetime). Data loss rate, θ, is defined

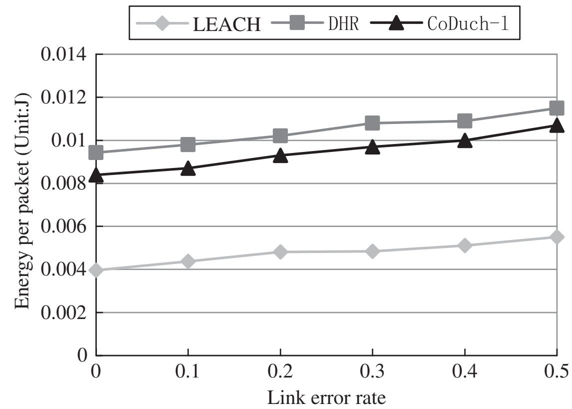


(a) Data loss rate in WSN with random link faults.

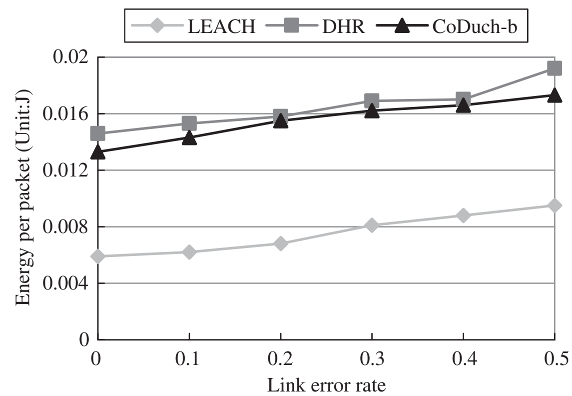


(b) Data loss rate in WSN with both random link faults and total 30% node faults.

FIGURE 7. Data loss rate.



(a) Energy dissipation analysis in WSN with random link faults.



(b) Energy dissipation analysis in WSN with both random link faults and total 30% node faults.

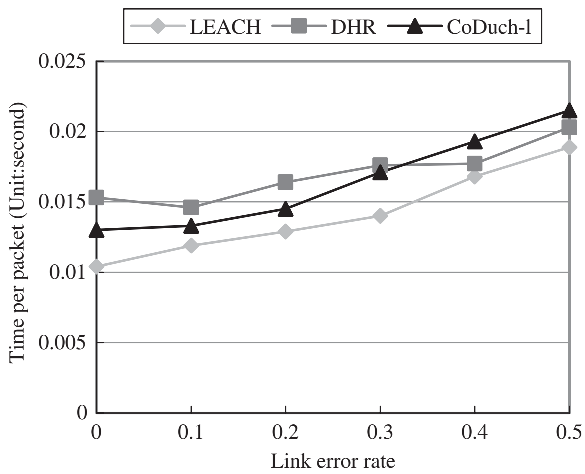
FIGURE 8. Energy efficiency.

as $\theta = 1 - N_{BS}/N_{sensing}$, where $N_{sensing}$ is the total number of sensed data packets and N_{BS} is the total number of data packets the BS has received. Note that we only count one copy of a sensor data at the BS in DHR and CoDuch, though two copies (one from CH and another from vice CH) of a sensor data may be received. The results are shown in Figs 6 and 7.

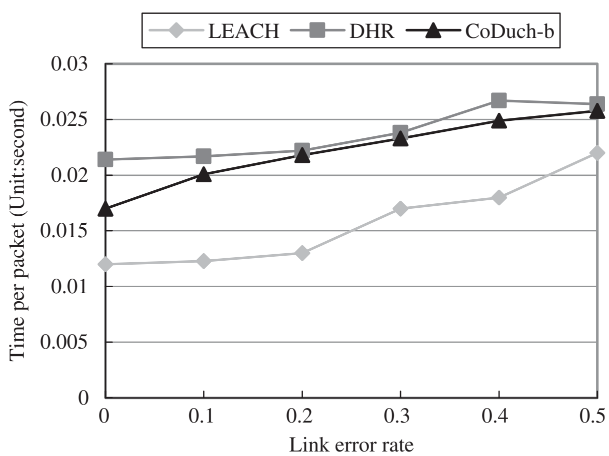
The overall trend in Figs 6 and 7 is that the CoDuch scheme (both CoDuch-l and CoDuch-b) performs better than LEACH and DHR in terms of the number of lost packets per round and data loss rate, since CoDuch always controls the data loss rate at the lowest level when we inject link faults or both node and link faults. Note that if no link/node faults are injected, CoDuch loses slightly more data packets than LEACH; we have explained this in Fig. 3 in Section 4.1.1.

Comparing Fig. 6a and b, more data packets are lost in LEACH, DHR and the CoDuch scheme when node faults are injected, but the trend remains the same: the more link faults are injected, the more data packets are lost in all of the three schemes. The CoDuch scheme is the best and DHR perfoms better than LEACH when link error rate is greater than or equal to 0.2 in Fig. 7a. A similar trend happens when we contrast Fig. 7b with Fig. 7(a). However, we notice that CoDuch-b tolerates node and link faults well in the same way as CoDuch-l tolerates link faults while DHR tolerates link faults better than it tolerates node faults. The evidence is that DHR performs far better than LEACH but slightly worse than CoDuch-l when the link error rate is greater than or equal to 0.2 in Fig. 7a while DHR performs slightly better than LEACH but far worse than CoDuch-b when the link error rate is greater than or equal to 0.3 in Fig. 7b.

Theoretically, CoDuch-l and DHR can reduce the data loss rate from ε to ε^2, where ε is the link error rate; for example, if we injected link error with 0.5 rate, the ideal data loss rate is 0.25. But the actual data loss rates at 0.5 link error rate are around 0.298 (in CoDuch-l) and 0.323 (in DHR), since internal faults are the other factor that incurs data loss. Also, Figs. 6 and 7 validate that CoDuch incurs less internal faults than DHR and thus tolerates faults better than DHR.



(a) Time analysis in WSN with random link faults.



(b) Time analysis in WSN with both random link faults and total 30% node faults.

FIGURE 9. Time efficiency.

5.2. Efficiency

In this section, we analyze the energy efficiency (in Fig. 8) and time efficiency (in Fig. 9) of our CoDuch Scheme.

We plot energy dissipation in WSN with only link faults in Fig. 8a and energy dissipation in WSN with both node and link faults in Fig. 8b. The energy consumption of CoDuch-l doubled that of LEACH as shown in Fig. 8a and the energy consumption of CoDuch-b tripled that of LEACH as shown in Fig. 8b due to the extra energy spent on the FT mechanism. Also, the energy dissipation curves of the three schemes have the common trend of increasing with the link error rates. However, we can see from both Fig. 8a and b that the CoDuch scheme (both CoDuch-l and CoDuch-b) consumes less energy than DHR when gathering data to the BS; these experimental results on the energy dissipation of CoDuch and DHR are consistent with our energy analysis in Section 4.2.

The time spent on gathering per packet to the BS in WSN with only link faults is plotted in Fig. 9a and in WSN with both node and link faults is plotted in Fig. 9b. Overall, CoDuch-l spends 16% more time on average than LEACH as shown in Fig. 9a while CoDuch-b spends 41% more time on average than LEACH as shown in Fig. 9b, since CoDuch-b spends more time than CoDuch-l to tolerate node faults. And LEACH remains nearly the same performance in both Fig. 9a and b.

Also, we observe from Fig. 9a that CoDuch-l performs better than DHR when the link error rates are no more than 0.3, since CoDuch-l receives more packets at the BS due to lower data loss rate. But CoDuch-l performs slightly worse than DHR when the link error rates are greater than 0.3 because the curve of DHR climbs more slowly with the link error rates than the curve of CoDuch-l. This can be explained by two aspects as follows:

(i) the DHR scheme does not change the time schedule of the WSN while CoDuch-l does. We first analyze the time spent in DHR: DHR removes redundant data from dual cluster heads at the BS and thus does not impact the time dissipation; also, a double number of cluster heads that send data to the BS using the CSMA scheme may only incur a little extra time by waiting to avoid collisions. Analyzing time spent in CoDuch-l, we find that CoDuch-l is more time consuming than the original WSN due to the two aspects: it removes redundant data gathered by dual cluster heads before sending data to the BS; and also it transfers data from vice cluster heads to cluster heads and then to the BS.
(ii) Figure 7a shows that the data loss rates of DHR and CoDuch-l are nearly the same when link error rate is no less than 0.3. The more data that is lost, the fewer number of packets are received by the BS; thus the time for each packet to be received by the BS is greater.

Thus, when the link error rate is greater than 0.3, CoDuch-l slightly loses its superiority because it is more time-consuming, compared with DHR.

However, CoDuch-b completely spends less time than DHR on gathering one packet to the BS as shown in Fig. 9b. We explain the reason for this in two ways.

(i) In DHR, dual cluster heads send data to the BS which is the furthest away and thus the energy consumption distribution in the whole WSN is uneven; this leads to some clusters having far more members than other clusters due to unevenly dead sensor nodes. However, since a TDMA scheme is adopted by dual cluster heads to gather intra-cluster data, the biggest cluster takes the longest time to gather data to the BS. When we injected node faults to dual cluster heads, the distribution of alive nodes becomes more uneven and thus produces uneven-sized clusters. Therefore, the total time in gathering data to the BS increases.
(ii) Figures 6b and 7b show that DHR loses more data than CoDuch-b and thus the BS receives less data in DHR.

Meanwhile, we can observe that many curves in Figs 6–9 do not have a linear development. The main reason is that the internal faults incur non-linear development of the data loss curves shown in Figs 6 and 7. According to the dependability theory, we provided in Equation (7) in Section 4.1.3, the final dependability is the result that the external dependability subtracts the data loss rate due to internal faults. Actually, it is not easy to estimate the internal faults due to the three factors we presented in Section 4.1.1. That is, (1) if no external faults are injected, Factor 1 in Section 4.1.1 is the main reason for incurring data loss; this can be used to explain that the data loss rate is greater than zero when link error rate is zero; and (2) if external faults are injected, both Factor 1 and Factor 2 need to be considered to explain why data loss is higher than the expected value under certain injected link and node faults. Moreover, time and energy dissipation in a distributed WSN with different link/node error rates also do not follow linear development, since the dynamically changed topology of the WSN brings unpredictable energy dissipation and time consumption to gather data.

In summary, CoDuch-l and CoDuch-b exceed both LEACH and DHR in tolerating link/node faults in terms of the number of lost packets per round and data loss rate and thus satisfy our goal of achieving lowest data loss. Moreover, CoDuch outperforms DHR in terms of energy per packet and time per packet in most cases.

6. CONCLUSIONS

We have provided a novel dual cluster heads cooperation (CoDuch) scheme for tolerating random link failures and node failures to improve the reliability in WSN. Two novel algorithms (CoDuch-l for tolerating link faults and CoDuch-b for tolerating both node faults and link faults) are developed based on the CoDuch scheme. We injected random uniform link errors and random Poisson distributed node errors into the WSN, and the experimental study shows that the CoDuch scheme performs better than its counterpart DHR in reducing data loss rate while spending less energy consumption and less time to tolerate link faults and node faults.

ACKNOWLEDGEMENTS

The authors are grateful to the referees for their useful and constructive comments on the early version of this paper.

FUNDING

This work was partially supported by Australian Research Council project (Grant No. LP0882957) and NSFC project (Grant No. 70602034).

REFERENCES

[1] Heinzelman, W.B., Chandrakasan, A.P. and Balakrishnan, H. (2002) An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. Wirel. Commun.*, **1**, 660–670.

[2] Gelenbe, E. and Ngai, E. (2010) Adaptive random re-routing for differentiated QoS in sensor networks. *Comput. J.*, **53**, 1052–1061.

[3] Gillies, D., Thornley, D. and Bisdikian, C. (2010) Probabilistic approaches to estimating the quality of information in military sensor networks. *Comput. J.*, **53**, 493–502.

[4] Gelenbe, E., and Hey, L. (2008) Quality of information: an empirical approach. *Proc. of the 5th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS'08)*, Atlanta, GA, USA, September 29–October 2, pp. 730–735. IEEE, New York, NY, USA.

[5] Calero, C., Caro, A. and Piattini, M. (2008) An applicable data quality model for web portal data consumers. *World Wide Web*, **11**, 465–484.

[6] Verma, D., Ko, B.J., Zerfos, P., Lee, K., He, T., Duggan, M., Stewart, K., Swami, A. and Sofra, N. (2010) Understanding the quality of monitoring for network management. *Comput. J.*, **53**, 541–550.

[7] Ruiz, L.B., Siqueira, I.G., Oliveira, L.B., Wong, H.C., Nogueira, J.M.S. and Loureiro, A.A.F. (2004) Fault management in event-driven wireless sensor networks. *Proc. of the 7th ACM Int'l Symp. on Modeling, Analysis and Simulation of Wireless and Mobile Systems 2004*, Venice, Italy, October 4–6, pp. 149–156. ACM New York, NY, USA.

[8] Gupta, G. and Younis, M. (2003) Fault-tolerant clustering of wireless sensor networks. *Proc. of IEEE WCNC'03*, New Orleans, LA, March 20–20, pp. 1579–1584. IEEE, Piscataway, NJ, USA.

[9] Yu, Y. and Zhang, Y. (2007) R-sentry: providing continuous sensor services against random node failures. *Proc. of DSN'07*, Edinburgh, UK, June 25–28, pp. 235–244. IEEE Computer Society, Los Alamitos, CA, USA.

[10] Ramanathan, N., Chang, K., Kapur, R., Girod, L., Kohler, E. and Estrin, D. (2004) Sympathy: a debugging system for sensor networks. *Proc. of IEEE International Conference on Local Computer Networks 2004*, Tampa, Florida, November 16-18, pp. 554–555. IEEE Computer Society, Los Alamitos, CA, USA.

[11] Technical Report 535 (2006) Snif: sensor network inspection framework. ETH Zurich, Zurich, Switzerland.

[12] UMASSD-CIS-TR-2006001 (2006) Evaluation of dual-homed fault-tolerant routing in wireless sensor networks. North Dartmouth, MA. USA.

[13] Jain, N., Vokkarane, V.M. and Wang, J.P. (2008) Performance analysis of dual-homed fault-tolerant routing in wireless sensor networks. *HST'08*, Waltham, MA, May 12–13, pp. 474–479. The Printing House, Inc.

[14] Huang, S. K., Ssu, K. F., and Wu, T. T. (2004) A fault-tolerant multipath routing protocol in wireless sensor networks. *International Computer Symposium 2004*, Taipei, Taiwan, December 15–17, pp. 966–971.

[15] Ding, M., Chen, D., Xing, K. and Cheng, X. (2005) Localized fault-tolerant event boundary detection in sensor networks. *Infocomm'05*, Miami, FL, March 13–17, pp. 902–913. IEEE, Piscataway, NJ, USA.

[16] Hill, J., Szewczyk, R., Woo, A., Hollar, S. and Heidemann, J. (2000) System architecture directions for networked sensors. *ACM SIGPLAN Notices*, **35**, 93–104.

[17] Madden, S., Franklin, M.J., Hellerstein, J.M. and Hong, W. (Dec. 2002) Tag: a tiny aggregation service for ad-hoc sensor networks. *OSDI'02*, Boston, MA, December 9–11, pp. 131–146. ACM New York, NY, USA.

[18] Tanachaiwiwat, S., Dave, P., Bhindwale, R. and Helmy, A. (2003) Secure locations: routing on trust and isolating compromised sensors in location-aware sensor networks. *SenSys'03*, Los Angeles, California, November 5–7, pp. 324–325. ACM New York, NY, USA.

[19] Marti, S., Giuli, T.J., Lai, K. and Baker, M. (2000) Mitigating routing misbehavior in mobile ad hoc networks. *ACM MOBICOM'00*, Boston, MA, August 6–11, pp. 255–265. ACM New York, NY, USA.

[20] Zou, Y. (2007) Redundancy analysis and a distributed self-organization protocol for fault-tolerant wireless sensor networks. *Int. J. Distrib. Sensor Netw.*, **3**, 243–272.

[21] CERCS, GIT-GERCS-06-09 (2006) Distributed fault-tolerance for event detection using heterogeneous wireless sensor networks. Georgia Institute of Technology.

[22] de Souza, L.M.S. (2007) Ft-cowisenets: a fault tolerance framework for wireless sensor networks. *SensorComm'07*, Valencia, October 14–20, pp. 289–294. IEEE Computer Society, Los Alamitos, CA, USA.

[23] Rost, S. and Balakrishnan, H. (2006) Memento: a health monitoring system for wireless sensor networks. *SECON'06*, Reston, VA, September 28–28, pp. 575–584. IEEE, Piscataway, NJ, USA.

[24] Harte, S. and Rahman, A. (June 2005) Fault tolerance in sensor networks using self-diagnosing sensor nodes. *IEEE Int'l workshop on Intelligent Environment 2005*, London, June 29, pp. 7–12. IEEE, London.

[25] Wang, P., Zheng, J. and Li, C. (2007) An agreement-based fault detection mechanism for under water sensor networks. *Proc. of the Global Telecommunications Conference 2007*, Washington DC, November 26-30, pp. 1195–1200. IEEE, Piscataway, NJ, USA.

[26] Heinzelman, W.B. (June 2000) Application specific protocol architectures for wireless networks. Ph.D Thesis, MIT.

[27] Akkaya, K., Demirbas, M. and Aygun, R. S. (2008) The impact of data aggregation on the performance of wireless sensor networks. *Wirel. Commun. Mob. Comput.*, **8**, 171–193.

[28] The network simulator (ns2). http://isi.edu/nsnam/ns, U.C. Berkeley.

[29] Knuth, D.E. (1997) Seminumerical Algorithms, *The Art of Computer Programming* (Vol. 2, 3rd edn). Addison-Wesley Longman.