LDTS: A Lightweight and Dependable Trust System for Clustered Wireless Sensor Networks

Xiaoyong Li, Feng Zhou, and Junping Du

***Abstract*—The resource efficiency and dependability of a trust system are the most fundamental requirements for any wireless sensor network (WSN). However, existing trust systems developed for WSNs are incapable of satisfying these requirements because of their high overhead and low dependability. In this work, we proposed a lightweight and dependable trust system (LDTS) for WSNs, which employ clustering algorithms. First, a lightweight trust decision-making scheme is proposed based on the nodes' identities (roles) in the clustered WSNs, which is suitable for such WSNs because it facilitates energy-saving. Due to canceling feedback between cluster members (CMs) or between cluster heads (CHs), this approach can significantly improve system efficiency while reducing the effect of malicious nodes. More importantly, considering that CHs take on large amounts of data forwarding and communication tasks, a dependability-enhanced trust evaluating approach is defined for cooperations between CHs. This approach can effectively reduce networking consumption while malicious, selfish, and faulty CHs. Moreover, a self-adaptive weighted method is defined for trust aggregation at CH level. This approach surpasses the limitations of traditional weighting methods for trust factors, in which weights are assigned subjectively. Theory as well as simulation results shows that LDTS demands less memory and communication overhead compared with the current typical trust systems for WSNs.**

***Index Terms*—Reputation, self-adaptivity, trust management, trust model, wireless sensor network.**

I. Introduction

FOR cluster wireless sensor networks (WSNs) such as LEACH [1], EEHC [2], EC [3], and HEED [4], clustering algorithms can effectively improve network scalability and throughput. Using clustering algorithms, nodes are grouped into clusters, and within each cluster, a node with strong computing power is elected as a cluster head (CH). CHs together form a higher-level backbone network. After several recursive iterations, a clustering algorithm constructs a multilevel WSN structure. This structure facilitates communication and enables the restriction of bandwidth-consuming network operations such as flooding only to the intended clusters [5].

Establishing trust in a clustered environment provides numerous advantages [6]–[8], such as enabling a CH to detect faulty or malicious nodes within a cluster [9]. In the case of multihop clustering [4], a trust system aids in the selection of trusted routing nodes through which a cluster member (CM) can send data to the CH. During intercluster communication, a trust system also aids in the selection of trusted routing gateway nodes or other trusted CHs through which the sender node will forward data to the base station (BS) [8].

A. Motivation

The resource efficiency and dependability of a trust system should undoubtedly be the most fundamental requirements for any WSN (including clustered WSNs). However, existing trust systems developed for clustered WSNs are incapable of satisfying these requirements because of their high overhead and low dependability. A universal trust system designed for clustered WSNs for the simultaneous achievement of resource efficiency and dependability remains lacking.

First, *limited work has focused on the resource efficiency of clustered WSNs.* A trust system should be lightweight to serve a large number of resource-constrained nodes in terms of accuracy, convergence speed, and additional overhead [7], [8], [10]. Based on an integrated comparison, a number of innovative works have been developed for clustered WSNs, such as GTMS [8], TCHEM [13], HTMP [9], ATRM [20]. However, most of these works failed to consider the problem of resource constraints of nodes or used complex algorithms to calculate nodes' trustworthiness. Implementing complex trust evaluation algorithms at each CM or CH is unrealistic. Although GTMS uses several novel mechanisms to improve the resource efficiency of clustered WSNs, this approach relies on a broadcast-based strategy to collect feedback among CMs, which requires a significant amount of resource and power.

Furthermore, *limited work has focused on the dependability of the trust system itself.* In existing trust mechanisms for WSNs, trust management systems collect remote feedback and then aggregates such feedback to yield the global reputation for the node that can be used to evaluate the global trust degree (GTD) of this node. However, an open or hostile WSN environment contains a large number of undependable (or malicious) nodes. Feedback from these undependable nodes may yield incorrect evaluation. The dependability of a trust system is undoubtedly an important requirement for any WSN environment. That is,

Manuscript received June 13, 2012; revised October 25, 2012; accepted January 06, 2013. Date of publication January 15, 2013; date of current version May 16, 2013. This work was supported by the National Nature Science Foundation of China (61003281), by the National Basic Research Program of China (973 Program) (2012CB821206), by Beijing Natural Science Foundation (4122051), by the Program for New Century Excellent Talents in University (NCET-12-0794), by Fok Ying Tung Education Foundation (132032), and by the Chinese Universities Scientific Fund (BUPT 2012RC0210). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yan Sun.

The authors are with Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: lxyxjtu@163.com; zfeng@bupt.edu.cn; junpingdu@126.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2013.2240299

a trust system should be highly dependable in terms of providing service in an open or hostile WSN environment. However, most previous studies lack feasible alternatives to solve the problem of malicious feedback, which significantly affects system dependability and feedback availability. Recent studies for clustered WSNs (e.g., TCHEM [13], HTMP [9]), the authors adopt simple weighted average approaches to aggregate feedback trust information without considering the problem of malicious feedback. This may result in misjudgment of the trust decision-making process.

B. Our Contributions

To the best of our knowledge, we are the first to conduct a systematic study of a trust management system for clustered WSNs from the perspective of both dependability and resource efficiency. The key features of LDTS go beyond existing approaches in terms of the following aspects:

1) *A lightweight trust evaluating scheme for cooperations between CMs or between CHs*. Within the cluster, the indirect trust of a CM is evaluated by its CH. Thus each CM does not need to maintain the feedback from other CMs, which will reduce the communication overhead and eliminate the possibility of a bad-mouthing attack by compromised CMs. The feedback of a CH is applied a similar manner to obtain the same benefits.
2) *A dependability-enhanced trust evaluating approach for cooperations between CHs*. Considering that CHs take on large amounts of data forwarding and communication tasks, a dependability-enhanced trust evaluating approach is defined for cooperations between CHs. This approach can effectively reduce networking consumption while preventing malicious, selfish, and faulty CHs.
3) *A self-adaptive weighting method for CH's trust aggregation*. This approach overcomes the limitations of traditional weighting methods for trust factors, in which weights are assigned subjectively.

These new designs and other specific features (e.g., independent of any specific routing scheme and platform and so forth) collectively make the LDTS a lightweight, self-adaptive, and dependable solution that can be used in any clustered WSN.

This paper will provide both theoretical foundations and experimental results to validate the designs of the LDTS. The remainder of this paper is organized as follows: Section II gives an overview of related work. The lightweight scheme for trust decision-making is described in Section III. Section IV discusses trust modeling and evaluation mechanism in LDTS. Sections V and VI respectively provide the theoretical and simulation-based analyses and evaluation of LDTS. Section VII concludes this paper.

II. Related Work

Research on trust management systems for WSNs received considerable attention from scholars. A number of studies have proposed such systems for WSNs [5], [10]–[12], [19], [21]. However, these systems suffer from various limitations such as the incapability to meet the resource constraint requirements of the WSNs, more specifically, for the large-scale WSNs. Recently, very few trust management systems have been proposed for clustered WSNs, such as GTMS [8], TCHEM [13], HTMP [9], and ATRM [20]. To our best knowledge, a universal trust system designed for clustered WSNs to achieve dependability and resource efficiency remains lacking.

Shaikh *et al.* [8] proposed GTMS, a group-based trust management scheme for clustered WSNs. GTMS evaluates the trust of a group of nodes in contrast to traditional trust schemes that always focus on the trust values of individual nodes. This approach gives WSNs the benefit of requiring less memory to store trust records at each node. GTMS aids in the significant reduction of the cost associated with the trust evaluation of distant nodes. However, GTMS relies on a broadcast-based strategy to collect feedback from the CMs of a cluster, which requires a significant amount of resources and power.

Bao *et al.* [9] proposed HTMP, a hierarchical dynamic trust management protocol for cluster-based WSNs that considers two aspects of trustworthiness: social trust and QoS (quality-of-service) trust. The authors developed a probability model utilizing stochastic Petri net techniques to analyze protocol performance and then validated subjective trust against the objective trust obtained based on ground truth node status. However, implementing such a complex trust evaluation scheme at each CM of the cluster is unrealistic.

Crosby *et al.* [13] proposed TCHEM, a trust-based cluster head election mechanism. Its framework is design in the context of a cluster-based network model with nodes that have unique local IDs. This approach can decrease the likelihood of malicious or compromised nodes from becoming CHs. The mechanism does not encourage sharing of trust information among sensor nodes. Thus, this approach reduces the effect of bad mouthing attacks. However, TCHEM does not cover trust in detail, because of which numerous key issues of trust management are not introduced.

Boukerche *et al.* [20] proposed ATRM, an agent-based trust and reputation management scheme. ATRM introduces a trust and reputation local management strategy with the aid of the mobile agents running on each node. The benefit of a local management scheme for trust and reputation is that centralized repositories are not required, and the nodes themselves capable of providing their own reputation information whenever requested. Therefore, reputation computation and propagation is performed without network-wide flooding and with no acquisition-latency. However, ATRM assumes that mobile agents are resilient against malicious nodes that try to steal or modify information that such agents carry. In numerous applications, this assumption may be unrealistic [8].

III. Lightweight Scheme for Trust Decision-Making

A. Network Topology Model and Assumptions

Our primary goal is to develop a trust-based framework for cluster-based WSNs as well as a mechanism that reduces the likelihood of compromised or malicious nodes being selected (or elected) as collaborative nodes. A node in the clustered WSN model can be identified as a CH, or a CM (See Fig. 1). Members of a cluster can communicate with their CH directly. A CH can forward the aggregated data to the central BS through other CHs. We assume that nodes are organized

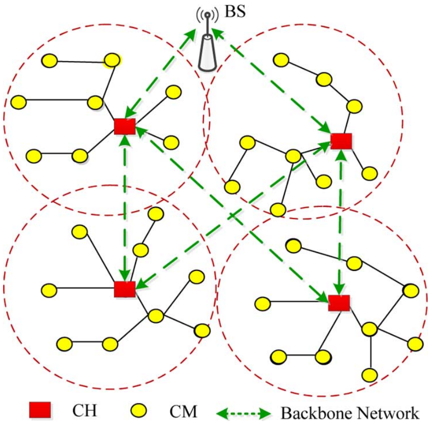


Fig. 1. Roles and identities of nodes in a clustered WSN model.

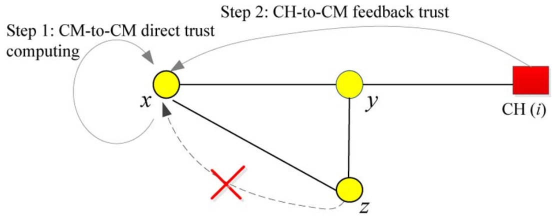

Step 2: CH-to-CM feedback trust
Step 1: CM-to-CM direct trust computing
x
y
CH (i)
z

Fig. 2. Trust decision-making at CM level.

into clusters with the help of a proposed clustering scheme such as [1] and [4]. We assume that all nodes have unique identities, which is similar to the assumptions of [8], [9], and [13]. In a number of sensor network models, nodes do not have unique identities similar to the Internet protocol in traditional networks. However, to uniquely identify nodes and to perform communication in such environments, a class-based addressing scheme [22] is used, in which a node is identified by a triplet $< \mathrm{location},\ \mathrm{node\ type},\ \mathrm{node\ subtype} >$ [8]. To protect trust values from traffic analysis or fabrication during transfer from one node to another, we also assume a secure communication channel, which can be established by using any key management scheme [25]–[27].

B. Lightweight Scheme for Trust Decision-Making

Our proposed LDTS facilitates trust decision-making based on a lightweight scheme. By closely considering the identities of nodes in clustered WSNs, this scheme reduces risk and improves system efficiency while solving the trust evaluation problem when direct evidence is insufficient (See Section IV-B). This scheme is described as follows:

1) Trust Decision-Making at CM Level: A CM calculates the trust value of its neighbors based on two information sources (Fig. 2): direct observations (or direct trust degree, DTD) and indirect feedback (or indirect trust degree, ITD). DTD is evaluated by the number of successful and unsuccessful interactions. In this work, interaction refers to the cooperation of two CMs. All CMs communicate via a shared bidirectional wireless channel and operate in the promiscuous mode, that is, if node x sends a message to CH i via node y, then node x can hear wether node y forwarded such message to CH i, the destination. If x does not overhear the retransmission of the packet within a threshold time from its neighboring node y or if the overheard packet is found to be illegally fabricated (by comparing the payload that is attached to the packet), then x will consider the interaction unsuccessful.

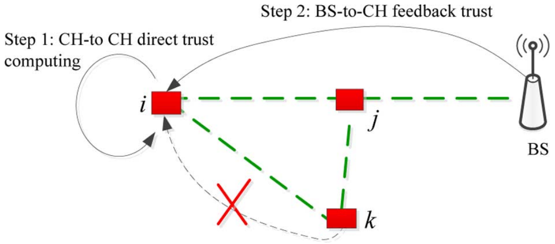

Step 2: BS-to-CH feedback trust
Step 1: CH-to CH direct trust computing
i
j
BS
k

Fig. 3. Trust decision-making at CH level.

Unlike most existing reputation or trust models, which rely on broadcast-based strategy to collect feedback from the whole cluster, consequently increasing the system communication overhead significantly, our LDTS does not utilize a broadcast-based strategy and instead sets the value of ITD is based on the feedback reported by the CH about a specific node. Thus, each CM does not need to share trust information with its neighbors. This indirect feedback mechanism has numerous advantages such as the effective mitigation of the effect of malicious feedback, thereby reducing the networking risk in an open or hostile WSN environment. Given that the feedback between CMs need not be considered, this mechanism can significantly reduce network communication overhead, thus improving system resource efficiency. As an example of trust decision-making at the CM level, if a node x wants to communicate with node y, x first checks whether it has any past interaction records with y during a specific time interval. If a past interaction record exists, then x makes a decision directly; otherwise, x will send a feedback request to its CH.

2) Trust Decision-Making at CH Level: In cluster WSNs, CHs form a virtual backbone for intercluster routing where CHs can forward the aggregated data to the central BS through other CHs. Thus, the selection of CHs is a very important step for dependable communication. In our LDTS, the GTD of a CH is evaluated by two information sources (Fig. 3): *CH-to-CH* direct trust and *BS-to-CH* feedback trust. During *CH-to-CH* communication, the CH maintains the records of past interactions of another CH in the same manner as CMs keep interaction records of their neighbors. Thus, the direct trust value can be computed according to the number of successful and unsuccessful interactions. The BS periodically asks all CHs for their trust ratings on their neighbors. After obtaining the ratings from CHs, the BS will aggregate them to form an effective value of ITD.

Similar to the trust decision-making process at the CM level, in our LDTS, the ITD of a CH only depends on the feedback reported by the BS. Thus, in the CH-to-CH communication case, when a CH i wants to interact with another CH j, it will send a feedback request to the BS, at the maximum. Therefore, including the response message form the BS, the total communication overhead is two packets. Thus, this mechanism can also greatly reduce network communication overhead and consequently improve the system's resource efficiency. Compared with trust decision-making at the CM level, trust decision-making at the CH level has to calculate for direct trust and feedback trust simultaneously. As an example of trust decision-making at the CH level, if a CH i wants to communicate

TABLE I
ALL TRUST RELATIONSHIPS IN LDTS

Trust Levels	Trust Relationships	Trust Decision
Intra-cluster (CM level)	CM-to-CM direct trust ($\mathcal{T}_{x,y}(\Delta t)$)	either $\mathcal{T}_{x,y}(\Delta t)$ or $\mathcal{R}_{ch,y}(\Delta t)$
	CH-to-CM feedback trust ($\mathcal{R}_{ch,y}(\Delta t)$)	
Inter-cluster (CH level)	CH-to-CH direct trust ($\mathcal{C}_{i,j}(\Delta t)$)	both $\mathcal{C}_{i,j}(\Delta t)$ and $\mathcal{F}_{i,j}(\Delta t)$
	BS-to-CM feedback trust ($\mathcal{F}_{bs,j}(\Delta t)$)	

with another CH j, i first calculates CH-to-CH direct trust for y based on the past interaction records with j during a specific time interval. Meanwhile, i sends a feedback request to the BS. After receiving the request, the BS will send a response message to i, in which j's feedback trust value (BS-to-CH feedback trust) is embedded. Then, i will aggregate these trust sources into a GTD, after which i will make a final decision based on j's GTD.

C. Summary of Trust Relationships in LDTS

As shown in Figs. 2 and 3, LDTS needs to maintain two levels of trust: intercluster trust and intracluster trust. Intracluster trust evaluation has two kinds of trust relationship: *CM-to-CM* direct trust and *CH-to-CM* feedback trust. Likewise, intercluster trust evaluation also has two kinds of trust relationship, *CH-to-CH* direct trust and *BS-to-CH* feedback trust. All trust relationships in LDTS are summarized in Table I. The calculation methods for these trust relationships are introduced in Section IV.

IV. LIGHTWEIGHT AND DEPENDABILITY-ENHANCED TRUST CALCULATION

A. Domain of Trust Values

The trust relationship is generally expressed as a specific quantitative value. This value can be a real number between 0 and 1 (e.g., [15]–[18]) or an integer between 0 and 100 (e.g., [8]). In this work, we transform this value into an unsigned integer in the interval between 0 and 10. Although presenting the trust values as a real number or an integer may be insignificant in traditional networks, this issue is of critical importance for WSNs because of limited memory as well as transmission and reception power [8]. This domain of trust values has the following benefits:

1) *Less memory overhead*. An unsigned integer between 0 and 10 only needs 4 bits (0.5 bytes) of memory space, thus saving save 50% memory space compared with trust values represented as an integer between 0 and 100 (1 bytes) and 87.5% compared with trust values represented as a real number (4 bytes).
2) *Less transmission overhead*. Given that a smaller number of bits require transmission during the exchange of trust values between nodes, we gain the benefit of less overhead of transmission and reception power.

B. Intracluster Trust Evaluation

1) CM-to-CM Direct Trust Calculation: The trust evaluation approach at CMs is defined by the following equation:

$$\mathcal{T}_{x,y}(\Delta t) = \left\lceil \left(\frac{10 \times s_{x,y}(\Delta t)}{s_{x,y}(\Delta t) + u_{x,y}(\Delta t)}\right)\left(\frac{1}{\sqrt{u_{x,y}(\Delta t)}}\right)\right\rceil \tag{1}$$

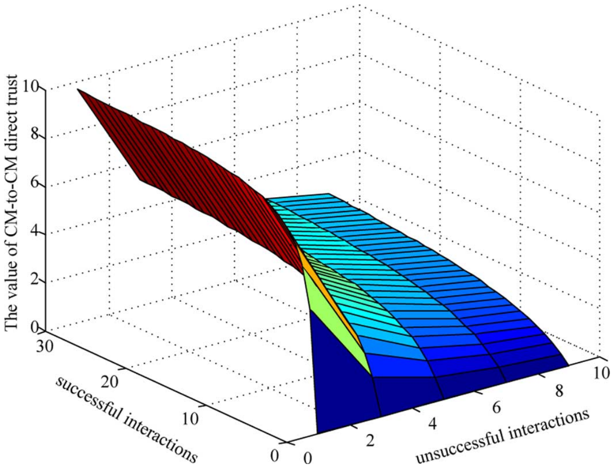


Fig. 4. Values of CM-to-CM direct trust.

where Δt is a window of time. The length Δt could be made shorter or longer based on network analysis scenarios. Thus, as time elapses, the window forgets old experiences but adds newer experiences. $\lceil\cdot\rceil$ is the nearest integer function, such that $\lceil 0.4567\rceil = 5$. $s_{x,y}(\Delta t)$ is the total number of successful interactions of node x with y during time Δt, $u_{x,y}(\Delta t) \neq 0$ is the total number of unsuccessful interactions of node x with y during time Δt. In special cases, if $s_{x,y}(\Delta t) \neq 0$ and $u_{x,y}(\Delta t) = 0$, we set $\mathcal{T}_{x,y}(\Delta t) = 10$. If $s_{x,y}(\Delta t) + u_{x,y}(\Delta t) = 0$, which denotes no interactions between node x and node y during time Δt. We set $\mathcal{T}_{x,y}(\Delta t) = \mathcal{R}_{ch,y}(\Delta t)$ (the formula of $\mathcal{R}_{ch,y}(\Delta t)$ see (3)). Integer $\mathcal{R}_{ch,y}(\Delta t) \in [1, 10]$ is the feedback trust toward node y reported by ch. Thus, a CM calculates the trust value of its neighbors based on two information sources: $\mathcal{T}_{x,y}(\Delta t)$ (CM-to-CM direct trust) and $\mathcal{R}_{ch,y}(\Delta t)$ (CH-to-CM indirect trust).

Given that the feedback between CMs need not be considered, this mechanism can greatly save on system resources. Moreover, from (1), we see that expression $1/\sqrt{u_{x,y}(\Delta t)}$ approaches 0 rapidly with an increase in the number of unsuccessful interactions, which indicates the strict punishment feature of LDTS for unsuccessful interactions. The strict punishment feature of LDTS can effectively prevent sudden attacks from malicious nodes with higher accumulated trustworthiness. Fig. 4 shows the values of *CM-to-CM* direct trust against successful and unsuccessful interactions. For example, the trust value is 9 with 1 unsuccessful and 10 successful interactions, and the trust value is 6 with 2 unsuccessful and 10 successful interactions.

2) CH-to-CM Feedback Trust Calculation: Supposing the existence of $(n-1)$ CMs in a cluster. The cluster head ch will periodically broadcast the request packet within the cluster. In response, all CMs in the cluster will forward their trust values toward other CMs to ch. Then, ch will maintain these trust values in a matrix $\mathcal{H}$, as shown below:

$$\mathcal{H} = \begin{pmatrix} \mathcal{T}_{1,1} & \mathcal{T}_{1,2} & \cdots & \mathcal{T}_{1,n-1} \\ \mathcal{T}_{2,1} & \mathcal{T}_{2,2} & \cdots & \mathcal{T}_{2,n-1} \\ & & \ddots & \\ \mathcal{T}_{n-1,1} & \mathcal{T}_{n-1,2} & \cdots & \mathcal{T}_{n-1,n-1} \end{pmatrix} \tag{2}$$

where $\mathcal{T}_{x,y}$ $(x \in [1, n-1], y \in [1, n-1])$ is the direct trust of node x on y. On the other hand, $x = y$, which means this value is a node's ratings towards itself. To reduce boasting, this value will be discarded by ch during feedback trust aggregation.

In [29], Whitby and Jøang proposed the beta feedback system, which is based on the theory of statistics and is characterized by flexibility and simplicity. Inspired by [29], we use the beta probability density functions to compute $\mathcal{R}_{ch,y}(\Delta t)$:

$$\mathcal{R}_{ch,y}(\Delta t) = \lceil 10 \times E(\varphi(p|r,v)) \rceil \tag{3}$$

where $\lceil \cdot \rceil$ is the nearest integer function, p denotes the posteriori probabilities of binary events (r, v), r is the amount of positive feedback $(\mathcal{T}_{x,y} \geq 5)$ towards node y counted from matrix $\mathcal{H}$, and v is the amount of negative feedback $(\mathcal{T}_{x,y} < 5)$ towards node y. $E(\varphi(p|r,v))$ is the probability expectation value of the beta distribution $\varphi(p|r,s)$:

$$E(\varphi(p|r,v)) = \frac{r+1}{r+v+2}. \tag{4}$$

Analyzing (3) and (4), our feedback aggregation mechanism is found to be a lightweight method with very simple mathematical formulas, which is suitable for resource-constrained nodes in a large-scale sensor network.

However, a possible attack scenario to the trust system must be considered. If a CH behaves badly in indirect trust feedback to its CMs, the CMs will have no idea that the feedback from the CH is actually misleading. Thus, the selection of a trustworthy node as the CH is one of the most significant requirements in cluster WSNs. This problem has been studied by several scholars. In TCHEM [13], Crosby *et al.* proposed a novel selection mechanism to reduce the likelihood of a malicious node to be selected as a CH. In [14], Ferdous *et al.* proposed an interesting scheme for the selection of a trustworthy CH that can provide secure communication via cooperative nodes. To make our LDTS independent of any specific clustering protocol, in this work, we assume that a trustworthy node has been selected as the CH of the cluster by using any selection protocol. That is, we assume that the CH is trustworthy within its cluster. For a possible CH selection solution, the readers can refer to the scheme in [13] or [14].

C. Dependability-Enhanced Intercluster Trust Evaluation

In accordance with the characteristics of clustered WSNs, both CMs and CHs are resource-constrained nodes, and BSs have more computing and storage capacity and no resource constraint problem. Thus, energy conservation remains a basic requirement for trust calculation at CHs. In LDTS, we propose a dependable and energy-saving scheme, which is suitable for large-scale and clustered WSNs.

1) CH-to-CH Direct Trust Calculation: During *CH-to-CH* communication, the CH maintains a record of past interactions with other CHs in the same manner as CMs keep records of other CMs. The direct trust between a CH i toward another CH j is defined as:

$$\mathcal{C}_{i,j}(\Delta t) = \left\lceil \left(\frac{10 \times S_{i,j}(\Delta t)}{S_{i,j}(\Delta t) + U_{i,j}(\Delta t)} \right) \left(\frac{1}{\sqrt{U_{i,j}(\Delta t)}} \right) \right\rceil \tag{5}$$

where $U_{i,j}(\Delta t) \neq 0$. $S_{i,j}(\Delta t)$ is the total number of successful interactions of CH i with CH j during time window Δt, and $U_{i,j}(\Delta t)$ is the total number of unsuccessful interactions of CH i with CH j. As a special case, if $S_{i,i}(\Delta t) \neq 0$ and $U_{i,j}(\Delta t) = 0$, we set $\mathcal{C}_{i,j}(\Delta t) = 10$.

2) BS-to-CH Feedback Trust Calculation: Supposing that m CHs exist in the network. The base station bs will periodically broadcast the request packet within the network. In response, all CHs in the network will forward their direct trusts for other CHs to bs. bs will maintain these trust values in a matrix $\mathcal{B}$, as shown below:

$$\mathcal{B} = \begin{pmatrix} \mathcal{C}_{1,1} & \mathcal{C}_{1,2} & \cdots & \mathcal{C}_{1,m} \\ \mathcal{C}_{2,1} & \mathcal{C}_{2,2} & \cdots & \mathcal{C}_{2,m} \\ & & \ddots & \\ \mathcal{C}_{m,1} & \mathcal{C}_{m,2} & \cdots & \mathcal{C}_{m,m} \end{pmatrix} \tag{6}$$

where $\mathcal{C}_{i,j}(i \in [1, m], j \in [1, m])$ is the direct trust of CH i toward CH j. Moreover, $i = j$, which means that this value is a CH's ratings for itself. To reduce boasting, this value will be discarded by the BS during feedback trust aggregation.

One of the difficulties of computing for *BS-to-CH* feedback trust is the question of malicious feedback. In [28], Liang and Shi found that the lightweight average aggregation algorithm performs better than complex algorithms, especially when a considerable number of bad raters exist in the system. Inspired by [28], [29], we use an enhanced beta probability density function to compute for *BS-to-CH* feedback trust:

$$\mathcal{F}_{bs,j}(\Delta t) = \left\lceil \frac{10 \times E(\varphi(p|g,l)) + \overline{\mathcal{C}_{k,j}}(\Delta t)}{2} \right\rceil \tag{7}$$

where p denotes the posteriori probabilities of binary events (g, l), g is the amount of positive feedback $(\mathcal{C}_{k,j} \geq 5)$ towards a CH j, and l is the amount of negative feedback $(\mathcal{C}_{k,j} < 5)$.

$$E(\varphi(p|g,l)) = \frac{g+1}{g+l+2} \tag{8}$$

which is the probability expectation value of the beta distribution function $\varphi(p|g,l)$ ([29]). $\overline{\mathcal{C}_{k,i}}$ is the average value of aggregated feedback from $(g + l)$ CHs in the network:

$$\overline{\mathcal{C}_{k,j}}(\Delta t) = \frac{\sum_{k=1}^{g+l} \mathcal{C}_{k,j}(\Delta t)}{g+l} \tag{9}$$

where $\mathcal{C}_{k,j}(\Delta t)$ is the feedback of CH k toward CH j.

Analyzing (7) to (9), our BS-to-CH feedback mechanism not only considers the amount of feedback $(g + l)$, but also considers the quality of each feedback $(\mathcal{C}_{k,i}(\Delta t))$. Therefore, our approach is more aligned with the habit of human cognition on feedback, which is an innovation of LDTS beyond approaches in [8], [9], [13], [20].

3) Self-Adaptive Global Trust Aggregation at CHs: We adopt the idea that the GTD of a CH comprises two parts (which is adopted by most studies on trust management [28]): the firsthand trust (*CH-to-CH* direct trust) and the secondhand trust

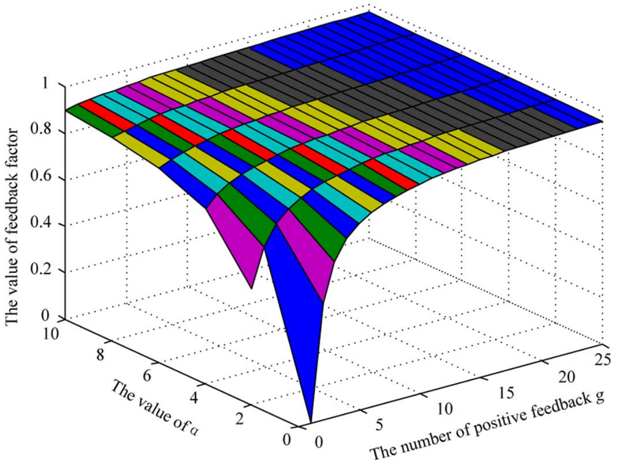


Fig. 5. Values of $\Phi(g)$ with different α and g.

(*BS-to-CH* feedback trust). Thus, the CH j's GTD is aggregated by the following equation:

$$O_{i,j}(\Delta t) = \lceil 10 \times (w_1 \times \mathcal{C}_{i,j}(\Delta t) + w_2 \times \mathcal{F}_{i,j}(\Delta t)) \rceil \quad (10)$$

where $\lceil \cdot \rceil$ is the nearest integer function. w_1 is the weight of $\mathcal{C}_{i,j}(\Delta t)$, and correspondingly, w_2 is the weight of $\mathcal{F}_{i,j}(\Delta t)$. The weights w_1 and w_2 meet $w_1 + w_2 = 1$. $\mathcal{C}_{i,j}(\Delta t)$ and $\mathcal{F}_{i,j}(\Delta t)$ can be computed according to (5) and (7), respectively. However, the level of accuracy of the values of w_1 and w_2 is a key question to be considered by this work.

How to avoid the effect of individual favoritism on the weight allocation of trust sources is a key task of trust management [23], [24], [28]. In this work, we define a self-adaptive approach to calculate the values of w_1 and w_2:

$$w_1 = \frac{\Phi(S)}{\Phi(S) + \Phi(g)}, \quad w_2 = \frac{\Phi(g)}{\Phi(g) + \Phi(S)} \quad (11)$$

where $\Phi(S) \in [0, 1]$ and S denote the total amount of successful interactions of CH i with j during Δt. $\Phi(g) \in [0, 1]$ is called the feedback factor. Constant g is provided by the BS, g is the amount of positive feedback ($\mathcal{C}_{k,j} \geq 5$) toward CH j. $\Phi(x)$ is a positive function that increases with the number of positive feedback g or the total amount of successful interactions S, which is defined as follows:

$$\Phi(x) = 1 - \frac{1}{\alpha + x} \quad (12)$$

where $\alpha \geq 0$ is a positive constant that can be tuned by the trust system accordingly.

The function $\Phi(x)$ has a desirable property in that with increasing α (α could be any positive integer), the function quickly approaches 1. Notably, instead of the above function $\Phi(x)$, we could have used any other function that has the property of quickly approaching 1 with increase in the argument. Our choice of the above function is aimed at brevity and ease of calculation. Fig. 5 shows the changes in the values of $\Phi(g)$ with different amounts of positive feedback g and adjustable constant α. The feedback factor $\Phi(g)$ is found to approach 1 rapidly with increasing α and positive feedback g. To increase the dependability of the trust system, we suggest that a smaller value of α, such as $\alpha = 1$, be set. Thus, the value of $\Phi(g)$ primarily depends on the amount of positive feedback g. For example, if $\alpha = 1$, $g = 4$, then $\Phi(g) = 1 - 1/(1 + 4) = 0.80$.

V. Theoretical Analysis and Evaluation

A. Dependability Analysis Against Malicious Attacks

In this section, we analyze the dependability of the LDTS protocol against attacks on a trust management system. In clustered WSNs, the main attacks from a malicious node primarily include two kinds of patterns:

1) *Garnished attack*. In such an attack, malicious nodes behave well and badly alternatively with the aim of remaining undetected while causing damage. For instance, garnished malicious nodes may suddenly conduct attacks as they accumulate higher trustworthiness.
2) *Bad mouthing attack*. As long as feedback is considered, malicious nodes can provide dishonest feedback to frame good parties and/or boost trust values of malicious nodes. This attack, referred to as the bad mouthing attack [6], is the most straightforward attack.

After providing evidence of the malicious nodes' objectives, we can prove that our trust management system at both the CM and CH levels is dependable against attacks from malicious nodes because this system can detect the malicious behavior and can prevent such nodes from fulfilling their objectives.

We broadly categorize two types of nodes (CMs or CHs): good ones and malicious ones. Our assumption is that good nodes interact successfully most of the time and submit true feedback. Conversely, malicious nodes try to launch garnished attacks or bad mouthing attacks. InSection VI, we define this concept more rigorously, capture the behavior of malicious nodes, and model how such nodes might try to gain an unfair advantage in our trust scheme. Then, we prove our trust system's dependability against such malicious attacks.

Definition 1: In the *LDTS* protocol, a CM y for a CM x is said to be trusted if its trust value $\mathcal{T}_{x,y}(\Delta t) \geq 5$. Accordingly, a CH j for a CH i is said to be trusted if its trust value $\mathcal{C}_{i,j}(\Delta t) \geq 5$.

Definition 2: In the *LDTS* protocol, a CM y is said to be malicious for a CM x if it has interacted with x at least once and $u_{x,y} > s_{x,y}$. A malicious CM y for a CM x is said to have deceived x if $\mathcal{T}_{x,y}(\Delta t) \geq 5$. Accordingly, a CH j is said to be malicious for a CH i if it has interacted with i at least once and $U_{i,j} > S_{i,j}$. A malicious CH j for a CH i is said to have deceived i if $\mathcal{C}_{i,j}(\Delta t) \geq 5$.

Definition 3: A trust management system is said to be dependable against deception by a malicious node (CM or CH) if no malicious node can deceive another node (CM or CH).

Theorem 1: In the *CM-to-CM* direct trust decision-making at CMs, the proposed LDTS is dependable against the deceptive behavior of malicious CMs.

Proof: Suppose, on the contrary, that a malicious CM y for a CM x that successfully deceived x. Then, according to the Definitions 1 and 2: $u_{x,y} > s_{x,y}$ and $\mathcal{T}_{x,y}(\Delta t) \geq 5$. Three cases can be considered.

(1) If $s_{x,y} \geq 1$, CM y has interacted with a CM x within the time stamp t. Let a denote the real number $u_{x,y}/s_{x,y}$.

Given that $u_{x,y} > s_{x,y}$, we can derive $a \geq 1$. Hence, given that $u_{x,y} + s_{x,y} \neq 0$, at the time of the last interaction, the trust calculation can be performed by using the past interaction evaluation, according to (1):

$$\begin{aligned}\mathcal{T}_{x,y}(\Delta t) &= \left(\frac{10 \times s_{x,y}}{s_{x,y} + u_{x,y}}\right)\left(\frac{1}{\sqrt{u_{x,y}}}\right)\\ &= \left(\frac{\frac{10 \times s_{x,y}}{s_{x,y}}}{\frac{s_{x,y}}{s_{x,y}} + \frac{u_{x,y}}{s_{x,y}}}\right)\left(\frac{1}{\sqrt{u_{x,y}}}\right)\\ &= \left(\frac{10}{1+a}\right)\left(\frac{1}{\sqrt{u_{x,y}}}\right), \quad a = \frac{u_{x,y}}{s_{x,y}}\\ &= \left(\frac{10}{1+a}\right)\left(\frac{\frac{\sqrt{u_{x,y}}}{s_{x,y}}}{a}\right) = \frac{10\sqrt{u_{x,y}}}{s_{x,y}(1+a)a}.\end{aligned}$$

Given that $s_{x,y} \geq 1$ and $u_{x,y} > s_{x,y}$, we obtain

$$u_{x,y} > 1 \Rightarrow \sqrt{u_{x,y}} > 1.$$

Given that $\mathcal{T}_{x,y}(\Delta t) \geq 5$, we obtain

$$\begin{aligned}&5 \leq \frac{10\sqrt{u_{x,y}}}{s_{x,y}(1+a)a} < \frac{10}{s_{x,y}(1+a)a}\\ &\Rightarrow 5 < \frac{10}{s_{x,y}(1+a)a} \Rightarrow \frac{1}{2} < \frac{1}{s_{x,y}(1+a)a}\end{aligned}$$

which implies that $s_{x,y}(1+a)a < 2$. Since $s_{x,y} \geq 1$, $a \geq 1$ and $(a+1) \geq 2$, which is obviously impossible and yields the contradiction $s_{x,y}(1+a)a < 2$. Thus, by using our trust evaluation approach, the condition $u_{x,y} > s_{x,y} \Rightarrow \mathcal{T}_{x,y}(\Delta t) \geq 5$ is impossible, which proves Theorem 1.

(2) If $s_{x,y} = 0$. We consider $u_{x,y} \geq 1$. Given that $u_{x,y} + s_{x,y} \neq 0$, at the time of the last interaction, the trust calculation can be performed by using the past interaction evaluation, according to (1):

$$\begin{aligned}\mathcal{T}_{x,y}(\Delta t) &= \left(\frac{10 \times s_{x,y}}{s_{x,y} + u_{x,y}}\right)\left(\frac{1}{\sqrt{u_{x,y}}}\right)\\ &= \left(\frac{10 \times 0}{0 + u_{x,y}}\right)\left(\frac{1}{\sqrt{u_{x,y}}}\right) = 0.\end{aligned}$$

Apparently, this condition contradicts the hypothesis $\mathcal{T}_{x,y}(\Delta t) \geq 5$, which proves Theorem 1.

(3) If $s_{x,y} = 0$ and $u_{x,y} = 0$, CM y has no interaction with CM x at all within the time t. In such case, x will rely on the feedback reported by the CH.

□

Theorem 1 shows that the LDTS is dependable against deception by a malicious CM. Meanwhile, Theorem 1 indirectly proves that the LDTS is dependable against the garnished attacks. From the theoretical basis of Theorem 1, the expression $1/\sqrt{u_{x,y}(\Delta t)}$ indicates a strict punishment feature for unsuccessful interactions. The strict punishment feature of LDTS can effectively prevent sudden attacks from malicious nodes with higher accumulated trustworthiness.

Theorem 2: In the *CH-to-CH* direct trust decision-making at CHs, the proposed LDTS is dependable against the deceptive behavior of malicious CHs.

Proof: Similar to Theorem 1. □

Definition 4: A set of really malicious CMs are said to be collaborating with one another if they provide false trust values of a particular CM to the CH.

Definition 5: A collaboration of really malicious CMs is successful against a CH ch toward a CM y, if the following conditions hold: (1) $v > r$ and $\mathcal{R}_{ch,y}(\Delta t) \geq 5$; or (2) $v < r$ and $\mathcal{R}_{ch,y}(\Delta t) < 5$.

Theorem 3: In the *CH-to-CM* feedback trust decision-making at CMs, the proposed LDTS is dependable against the deceptive, collaborative feedback of malicious CMs.

Proof:

(1) Conditions $v > r$ and $\mathcal{R}_{ch,y}(\Delta t) \geq 5$ cover the badmouthing scenario where nodes collaborate to lie about a bad node. Suppose, on the contrary, that a collaboration that successfully deceived ch exists. Then, according to Definitions 4 and 5:

$$v > r \text{ and } \mathcal{R}_{ch,y}(\Delta t) \geq 5.$$

Therefore at the time of the last interaction, the trust calculation can be performed by using the beta probability density functions, according to (3) and (4):

$$\begin{aligned}\mathcal{R}_{ch,y}(\Delta t) &= \frac{10(r+1)}{r+v+2}, \quad \mathcal{R}_{ch,y}(\Delta t) \geq 5\\ &\Rightarrow \frac{10(r+1)}{r+v+2} \geq 5 \Rightarrow \frac{r+1}{r+v+2} \geq \frac{1}{2}\\ &\Rightarrow 2(r+1) \geq r+v+2 \Rightarrow 2r \geq r+v \Rightarrow r \geq v.\end{aligned}$$

Apparently, this condition contradicts the hypothesis $v > r$, which proves Theorem 3.

(2) Conditions $v < r$ and $\mathcal{R}_{ch,y}(\Delta t) < 5$ cover the badmouthing scenario where a group of nodes collaborate to lie about a good node. The proof is similar to that of Case (1).

□

Definition 6: A set of really malicious CHs are said to be collaborating with one another if they provide false trust values of a particular CH to the BS.

Definition 7: A collaboration of really malicious CHs is successful against a BS bs toward a CH j, if the following conditions hold: (1) $l > g$; (2) $\mathcal{F}_{bs,j}(\Delta t) \geq 5$.

Theorem 4: In the BS-to-CH feedback trust decision-making at BSs, the proposed LDTS is dependable against the deceptive, collaborative feedback of malicious CHs.

Proof: The proof is straightforward. We need to prove that when $v > r$, $\mathcal{R}_{ch,y}(\Delta t) \leq 5$. According to (7), (8), and (9), the *BS-to-CH* feedback trust can be calculated by using the enhanced beta probability density functions:

$$\begin{aligned}\mathcal{F}_{ch,j}(\Delta t) &= \frac{10 \times E(\varphi(p|g,l)) + \overline{\mathcal{C}_{k,j}}(\Delta t)}{2}\\ &= \frac{10 \times \frac{g+l}{g+l+2} + \overline{\mathcal{C}_{k,j}}(\Delta t)}{2}\\ &= \frac{10 \times \frac{g+l}{g+l+2}}{2} + \frac{\overline{\mathcal{C}_{k,j}}(\Delta t)}{2}.\end{aligned}$$

We need to prove $\mathcal{F}_{ch,j}(\Delta t) \le 5$, that is

$$\frac{10 \times \frac{g+l}{g+l+2}}{2} + \frac{\overline{\mathcal{C}_{k,j}}(\Delta t)}{2} \le 5.$$

Similar to Theorem 3, the value of $10\times(g+l)/(g+l+2) \le 5$. Thus, $(10 \times (g+l)/(g+l+2)) \le 2.5$. In the condition where $l > g$, the negative feedback outweighs the positive feedback. According to (9), $\overline{\mathcal{C}_{k,j}}(\Delta t) \le 5$, and we obtain $(\overline{\mathcal{C}_{k,j}}(\Delta t)/2) \le 2.5$. Thus, we can derive:

$$\frac{10 \times \frac{g+l}{g+l+2}}{2} \le 2.5, \frac{\overline{\mathcal{C}_{k,j}}(\Delta t)}{2} \le 2.5$$
$$\Rightarrow \frac{10 \times \frac{g+l}{g+l+2}}{2} + \frac{\overline{\mathcal{C}_{k,j}}(\Delta t)}{2} \le 5.$$

This condition proves Theorem 4. □

B. Communication Overhead Analysis and Comparison

To evaluate the communication overhead under full-load conditions, we assume a worst-case scenario which is similar to [8], in which every CM wants to communicate with every other CM in the cluster, and every CH wants to communicate with the rest of the CHs in the network. At the same time, each CH needs to collect feedback reports from its CMs, and the BS has to collect feedback reports from its CHs. Let us assume that the network consists of m clusters and that the average size of clusters is n (including the CH of the cluster).

In intracluster trust evaluation, when node x wants to interact with node y, node x will send a maximum of one CH feedback request, for which node x will receive one response. If node x wants to interact with all the nodes in the cluster, the maximum communication overhead will be $2(n-2)$. If all nodes want to communicate with one another, the maximum communication overhead is $2(n-2)(n-1)$. When a CH wants to collect feedback from its n members, it will send n requests and receive n responses, thus resulting in a total communication overhead of $2n$. Thus, the maximum intracluster communication overhead is $C_{intra} = 2(n-2)(n-1) + 2n$.

In the intercluster communication case, when CH i wants to interact with CH j, it will send one BS feedback request to the BS, at the maximum. Thus, the communication overhead is two packets. If CH i wants to communicate with all the CHs, then the maximum communication overhead will be $2(m-1)$ packets. If all the CHs want to communicate with one another, the maximum communication overhead $2(m-1)(m-1) = 2(m-1)^2$. When the BS wants to collect feedback from its m CHs, it will send m requests and receive m responses, thus resulting in a total communication overhead of $2m$. Thus, the maximum intercluster communication overhead is $C_{inter} = 2(m-1)(m-1) + 2m = 2(m-1)^2 + 2m$.

Therefore, the maximum communication overhead $C_{\max}$ introduced by the LDTS to the entire network is:

$$\begin{aligned} C_{\max} &= m \times C_{intra} + C_{inter} \\ &= 2m[(n-2)(n-1) + n] + 2(m-1)^2 + 2m. \end{aligned} \tag{13}$$

Fig. 6 shows the communication overhead of various trust management systems under a large-scale clustered WSN environment, which has a total of 10,000 nodes. On the whole curve, we can see that our LDTS requires less communication overhead than two other notable trust systems, GTMS and ATRM. Based on the first part of the curve, LDTS and ATRM need less communication overhead than GTMS. However, as the number of clusters increases in the network, LDTS and GTMS introduce less communication overhead than ATRM. Thus, the important condition that we need to note here is that our LDTS is highly suitable for large-scale WSNs with either a small or a large size of clusters and having large size of clusters, thus outperforming GTMS and ATRM.

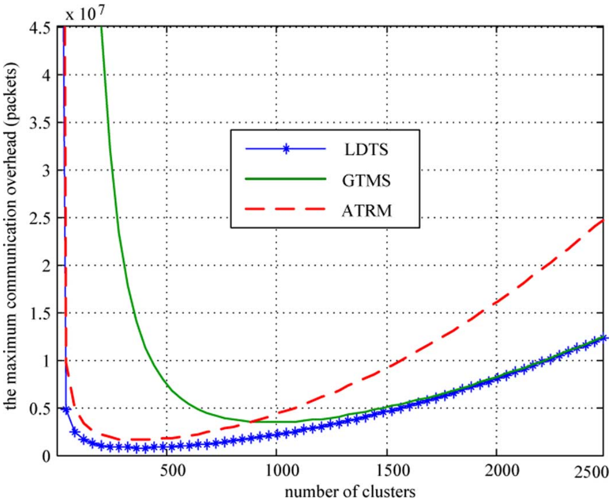


Fig. 6. Communication overheads with 10,000 nodes.

Node ID	the number of interactions		CM-to-CM direct trust	CH-to-CM feedback trust
	$s_{x,y}$	$u_{x,y}$		
2 bytes	2 bytes	2 bytes	0.5 bytes	0.5 bytes

(a)

Node ID	the number of interactions		CH-to-CH direct trust	BS-to-CH feedback trust
	$S_{i,j}$	$U_{i,j}$		
2 bytes	2 bytes	2 bytes	0.5 bytes	0.5 bytes

(b)

Fig. 7. Storage overhead of the trust database at each node. (a) Trust database at each CM (7 bytes). (b) Trust database at each CH (7 bytes).

C. Storage Overhead Analysis and Comparison

Each CM has to maintain a small trust database, as shown in Fig. 7(a). The size of each record is 7 bytes. Therefore, the storage requirement for LDTS at each CM is $7(n-1)$ bytes, where $(n-1)$ represents the number of CMs in a cluster. The size of the trust table mainly depends on the size of the cluster. Each CH maintains two tables, one of which is used to store the feedback matrix $\mathcal{H}$ (see (2)), thus resulting in a total storage overhead of $0.5(n-1)^2$. In the second table, each CH maintains a trust database as shown in Fig. 7(b). The size of each record also is 7 bytes. Therefore, storage requirement for m CHs is $7(m-1)$ bytes, where $(n-1)$ represents the number of CMs in a cluster. The total storage overhead at the CH for both tables is $C_{m-max} = 7(m-1) + 0.5m(n-1)^2$.

The formulas for the storage requirements of three trust management systems LDTS, GTMS, and ATRM, are given in Table II, in which n represents the average number of CMs

TABLE II
ANALYSIS AND COMPARISON OF STORAGE REQUIREMENTS FOR LDTS, GTMS, AND ATRM

models	CM nodes	CH nodes
LDTS	$7(n-1)$	$7(m-1)+0.5m(n-1)^2$
GTMS	$(n-1)(4+4\Delta t)$	$(m+n-2)(4+4\Delta t)$
ATRM	$30n+8(k-1)$	$30(m+n)+2(4k-19)$

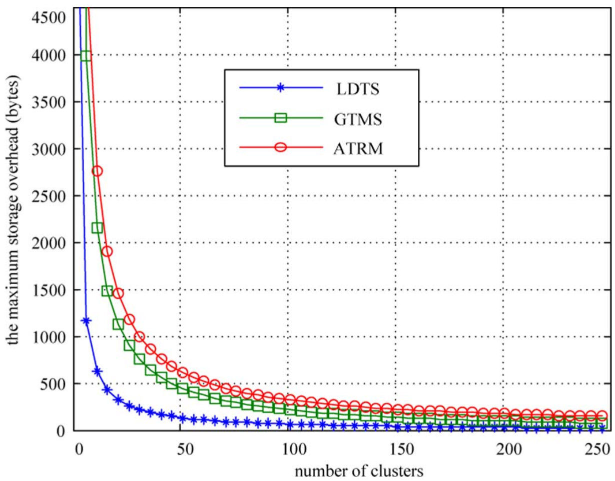


Fig. 8. Storage overhead at each CM with 1,000 nodes.

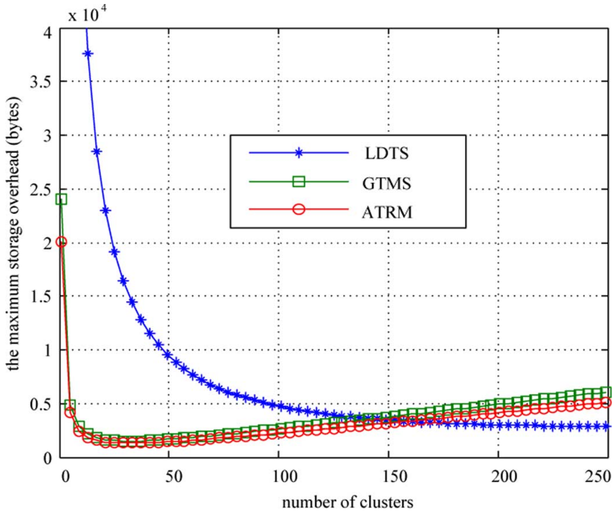


Fig. 9. Storage overhead at each CH with 1,000 nodes.

in each cluster, m represents the total number of CHs in the network, Δt is the time window defined by GTMS, and k represents the number of contexts described in ATRM (for details about the storage requirements of GTMS and ATRM, please see [8]).

Figs. 8 and 9 show the storage overhead of three trust management systems under a clustered WSN environment, which has a total of 1,000 nodes. On the whole curve of Fig. 8, we can see that our LDTS needs less storage overhead than the two other trust systems, GTMS and ATRM. This condition proves that LDTS at the CM level consumes less memory than the two other models. Fig. 9 shows that as the number of clusters increases in the network, the LDTS introduces less storage overhead at the CH level compared with the two other systems, which indicates that LDTS is more suitable for large-scale WSNs having a small size of clusters. The results in Fig. 9 can be easily explained by Table II. In LDTS, the total storage overhead $M_{\max}$ at the CH level is $7(m-1)+0.5m(n-1)^2$ bytes. Evidently, the value of M_{max} primarily depends on the number of nodes at each cluster. As the number of nodes at each cluster increases, the storage consumption requirement also increases at the CH. As the number of nodes at each cluster decreases, the storage consumption requirement also decreases linearly at the CH.

TABLE III
ROLE AND CLASSIFICATION OF A NODE IN THE SIMULATOR

Node classification	Node's role	Node's behavior
Cluster member (CM)	Collaborator	GCM, BCM
	rater	HCM, MCM
Cluster head (CH)	Collaborator	GCH, BCH
	Rater	HCH, MCH
Base station (BS)	Rater	Always honest

VI. SIMULATION-BASED ANALYSIS AND EVALUATION

By extending the Netlogo-based trust simulation engine [23], [28], we implemented a simulator to test the feasibility of the proposed LDTS. For the purpose of comparison, we also added GTMS [8] into the simulator, because both LDTS and GTMS are independent of any specific routing scheme and platform. We did not implement the ATRM system [20] because it requires a specific agent-based platform.

A. LDTS Simulator and Environment

In the simulator, three kinds of nodes exist based on their identities (Table III), i.e., as a CM, as a CH, and as a BS. A CM or a CH can be a collaborator or a rater toward other nodes. The behavior of a CM as a collaborator can be one of two types: good CM (GCM) and bad CM (BCM). GCMs will provide successful interaction for the requested messages, whereas BCMs will provide an unsuccessful interaction. The behavior of a CM as a rater can be one of two types: honest CM (HCM) and malicious CM (MCM). An HCM always gives the appropriate rating for any CM, whereas an MCM always gives a random rating between 0 and 10 for other CMs. Similar to a CM, a GCH always provide successful interaction, whereas a BCH provide an unsuccessful interaction. An HCH always gives an appropriate rating, whereas an MCH always gives random rating between 0 and 10.

Based on discussions in Section III and IV, we can see that LDTS works with two topologies: the intercluster (CH-to-CH) topology, where distributed trust management is used, and intracluster (CM-to-CM) topology, where the centralized trust management approach is employed. We also find that different calculation mechanisms are employed for intracluster and intercluster trust evaluations. According to these characteristics of LDTS, in this simulator, we separately evaluate the performance of LDTS based on intracluster and intercluster cases. This approach will not affect the results of performance evaluation and will greatly reduce the complexity of the simulator. Instead of using the physical running time, we use the notion of time-step, which is introduced in Netlogo, to calculate the simulation time. The simulation parameters and default values used in the experiments are listed in Table IV.

TABLE IV
PARAMETERS AND THEIR POSSIBLE VALUES

Symbol	**Description**	**Possible Values**
$N = m \times n$	the number of nodes	$160 - 1800$
n	the number of CMs in a cluster	$8 - 18$
m	the number of clusters	$20 - 100$
t	Time-Steps of simulation runs	1000
α	adjustable constant in Eq.(12)	1

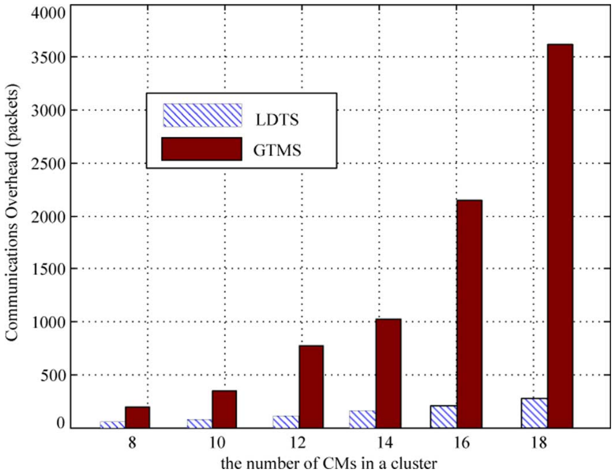


Fig. 10. CM-to-CM communication overhead in a cluster.

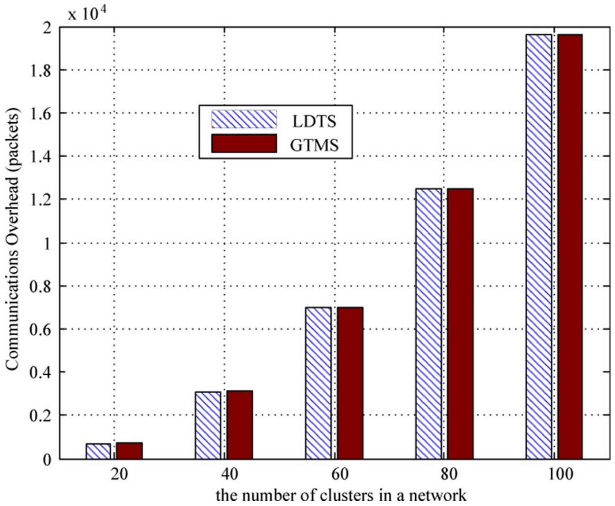


Fig. 11. CH-to-CH communication overhead in a network.

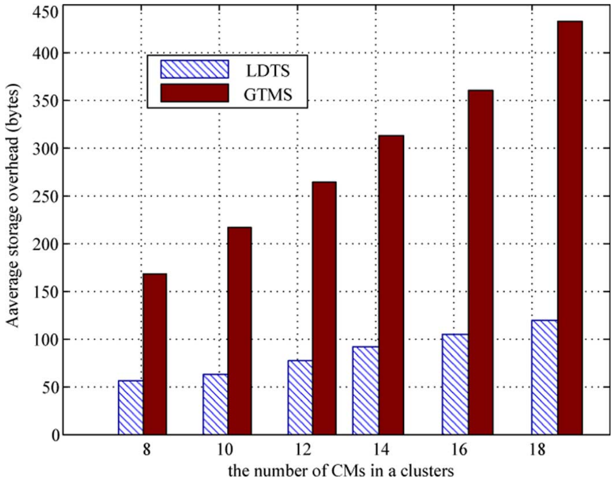


Fig. 12. Average storage overhead at each CM in a cluster.

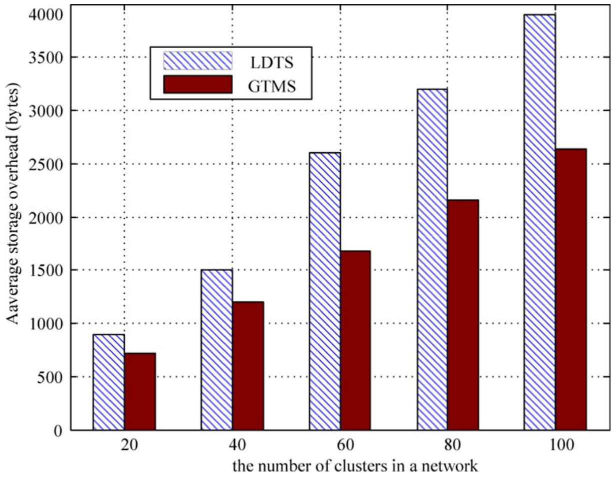


Fig. 13. Average storage overhead at each CH in a network.

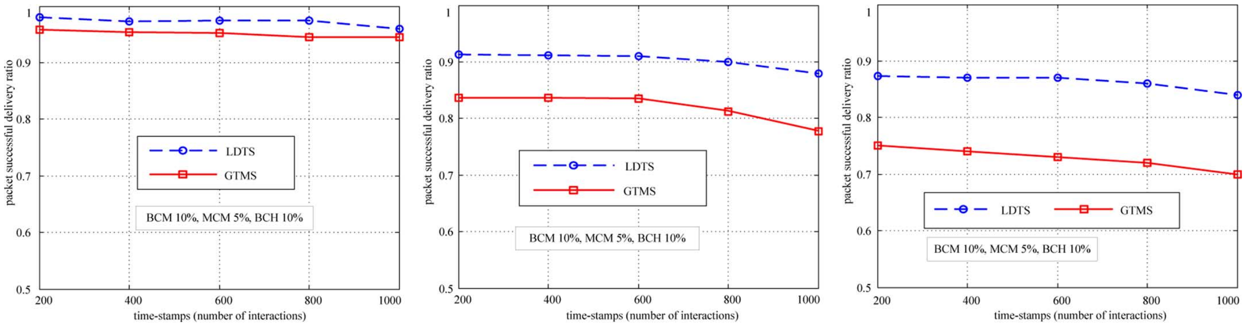
B. Overhead Evaluation and Comparison

We aim to study the effect of the trust management system in a WSN community, which closely resembles a real network environment. We suppose that most CMs and CHs are good, where only 20% CMs and CHs are malicious. The comparison results are shown in Fig. 10. With the increasing the number of CMs in a cluster, the CM-to-CM communication overhead of GTMS rapidly increased according to a exponential curve. However, the CM-to-CM communication overhead of LDTS slowly increased with the increasing number of CMs. This finding further confirms our conclusions from the theoretical analysis in Section VI, that is, given that feedback between CMs need not be considered, this trust calculation mechanism in LDTS can greatly reduce communication overhead.

Fig. 11 shows the comparison results of the CH-to-CH communication overhead between LDTS and GTMS. LDTS and GTMS have a relatively close network overhead as the network size increases, which indicates that both LDTS and GTMS are suitable for large-scale clustered WSNs. However, by comprehensively analyzing the results in Figs. 10 and 11, LDTS is more suitable for large-scale clustered WSNs with a large size of clusters, thus outperforming GTMS.

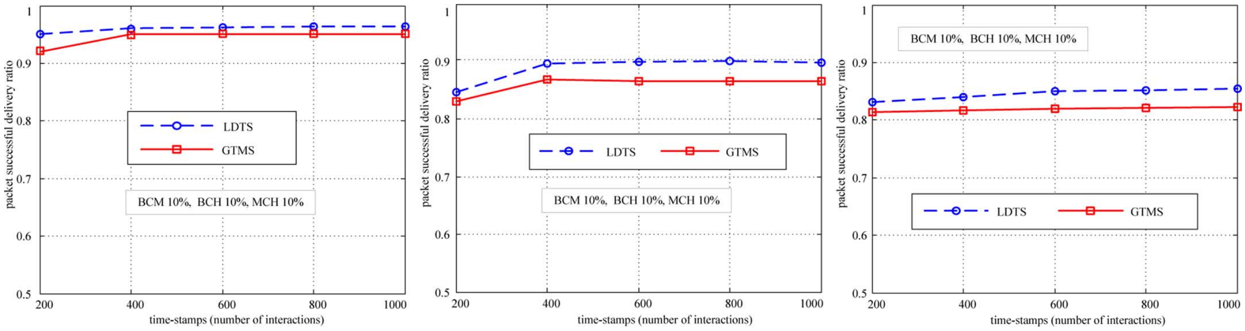
Fig. 12 shows the comparison results of average storage overhead at each CM in a cluster. With the increasing number of CMs in a cluster, the average storage overhead of GTMS gradually increased according to a linear curve. However, the average storage overhead of LDTS was less than a third of that of GTMS and slowly increased with the increasing number of CMs. This finding confirms our conclusions from the theoretical analysis in Section VI.

Fig. 13 shows the average storage overhead of the two trust systems at each CH in a WSN network having an equal size of clusters (10 nodes). We find that as the number of clusters increases in the network the GTMS introduces slightly less storage overhead compared with LDTS. The results in Fig. 13 can be easily explained by (2). Each CH has to maintain an additional table, which is used to store the feedback matrix $\mathcal{H}$ (see (2)). The total storage overhead is $0.5(n-1)^2$. Although the introduction of matrix $\mathcal{H}$ increases the storage overhead of



(a) (b) (c)

Fig. 14. PSDR comparison with different percentages of MCHs. (a) MCH 5%. (b) MCH 10%. (c) MCH 20%.



(a) (b) (c)

Fig. 15. PSDR comparison with different percentages of MCMs. (a) MCM 10%. (b) MCM 20%. (c) MCM 30%.

a CH node, this matrix can significantly enhance the dependability of CH-to-CM trust evaluation.

C. Dependability Evaluation and Comparison

We compute the *packet successful delivery ratio* (PSDR) [6] to reflect the dependability of trust management systems. A higher PSDR indicates higher dependability. We suppose that most CMs and CHs are good in the WSN community, where BCMs and BCHs each comprise only 10%. This WSN environment closely resembles a real situation, where most CMs are honest and most CHs are good.

Fig. 14 shows the PSDR comparison results under different percentages of malicious cluster heads (MCHs). In this group simulation, we suppose that in the WSN community, where 95% of CMs are honest. The remaining 5% of CMs are MCMs. We separately set the percentage of MCHs as 5%, 10%, and 20%, which respectively indicate that the WSN environment is honest, relatively honest, and dishonest community, with 50, 100, and 200 dishonest CHs separately. Fig. 14(a) shows an honest WSN environment, where the percentage of MCHs is only 5%. We can see that both LDTS and GTMS have a high PSDR, which reflects that these two models have a high dependability under an honest WSN environment. In Fig. 14(b) and (c), the simulation results when MCHs is 10% and 20% have larger differences compared with $\mathrm{MCHs} = 5\%$. With the increase in the percentage of malicious CHs, the performance of both LDTS and GTMS show a marked decline. Relatively, LDTS has a robust performance under a dishonest WSN environment. These results are consistent with a real situation, i.e., in a dishonest WSN community, malicious CHs may conduct a bad-mouthing attack, which can greatly affect the performance of the WSN system. To reduce the risk of trust evaluation, we adopt the idea that the GTD of a CH is adaptively merged by two parts (which is not aggregated by GTMS): *CH-to-CH* direct trust and *BS-to-CH* feedback trust. This can significantly improve the dependability of LDTS.

Fig. 15 shows the PSDR comparison results under different percentages of MCMs. We find that LDTS also has a more robust dependability than the GTMS scheme. Fig. 15(a) shows the experimental results under an honest environment. In the simulation, the total percentage of MCMs is 10%, and the total percentage of MCHs is likewise 10%, which indicate that the community is a relatively honest community (i.e., with fewer MCHs and MCMs). Both LDTS and GTMS have relatively stable performance within 1,000 time-steps, even if their PSDRs change from 0.92 to 0.96. Fig. 15(b) shows the experimental results under a relatively honest environment, where 20% of CMs are dishonest. The results show that LDTS has a higher PSDR than GTMS. Fig. 15(c) shows the experimental results under a highly dishonest environment, where 30% of CMs are dishonest. Under this case, LDTS still shows better dependability than GTMS.

VII. Conclusion

In this work, we proposed LDTS for clustered WSNs. Given the cancellation of feedback between nodes, LDTS can greatly improve system efficiency while reducing the effect of malicious nodes. By adopting a dependability-enhanced trust evaluating approach for cooperations between CHs, LDTS can effectively detect and prevent malicious, selfish, and faulty CHs. Theory as well as simulation results show that LDTS demands less memory and communication overhead as compared with other typical trust systems and is more suitable for clustered WSNs.

REFERENCES

[1] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[2] D. Kumar, T. C. Aseri, and R. B. Patel, "EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks," *Comput. Commun.*, vol. 32, no. 4, pp. 662–667, Apr. 2009.

[3] Y. Jin, S. Vural, K. Moessner, and R. Tafazolli, "An energy-efficient clustering solution for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 11, pp. 3973–3983, Nov. 2011.

[4] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for Ad-Hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Oct. 2004.

[5] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Trans. Sensor Netw.*, vol. 4, no. 3, pp. 1–37, May 2008.

[6] Y. Sun, Z. Han, and K. J. R. Liu, "Defense of trust management vulnerabilities in distributed networks," *IEEE Commun. Mag.*, vol. 46, no. 2, pp. 112–119, Feb. 2009.

[7] H. Yu, Z. Shen, C. Miao, C. Leung, and D. Niyato, "A survey of trust and reputation management systems in wireless communications," *Proc. IEEE*, vol. 98, no. 10, pp. 1752–1754, Oct. 2010.

[8] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, and S. Lee, "Group-based trust management scheme for clustered wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 11, pp. 1698–1712, Nov. 2009.

[9] F. Bao, I. Chen, M. Chang, and J. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE Trans. Netw. Service Manag.*, vol. 9, no. 2, pp. 169–183, Jun. 2012.

[10] G. Zhan, W. Shi, and J. Deng, "Design and implementation of TARF: A trust-aware routing framework for WSNs," *IEEE Trans. Depend. Secure Comput.*, vol. 9, no. 2, pp. 184–197, Apr. 2012.

[11] E. Aivaloglou and S. Gritzalis, "Hybrid trust and reputation management for sensor networks," *Wireless Netw.*, vol. 16, no. 5, pp. 1493–1510, Jul. 2010.

[12] A. Rezgui and M. Eltoweissy, "$\mu\mathrm{RACER}$: A reliable adaptive service-driven efficient routing protocol suite for sensor-actuator networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 5, pp. 607–622, May 2009.

[13] G. V. Crosby, N. Pissinou, and J. Gadze, "A framework for trust-based cluster head election in wireless sensor networks," in *Proc. Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems*, 2006, pp. 10–22.

[14] R. Ferdous, V. Muthukkumarasamy, and E. Sithirasenan, "Trust-based cluster head selection algorithm for mobile ad hoc networks," in *Proc. 2011 Int. Joint Conf. IEEE TrustCom-1111/IEEE ICESS-11/FCST-11*, pp. 589–596.

[15] Z. Liang and W. Shi, "TRECON: A trust-based economic framework for efficient internet routing," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 1, pp. 52–67, Jan. 2010.

[16] R. Zhou and K. Hwang, "Power-trust: A robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 5, pp. 460–473, May 2007.

[17] Y. Sun, W. Yu, Z. Han, and K. J. R. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 305–317, Feb. 2006.

[18] G. Theodorakopoulos and J. S. Baras, "On trust models and trust evaluation metrics for ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 318–328, Feb. 2006.

[19] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," in *Proc. ACM Workshop Security of ad hoc and Sensor Networks (SASN'04)*, Oct. 2004, pp. 66–67.

[20] A. Boukerche, X. Li, and K. EL-Khatib, "Trust-based security for wireless ad hoc and sensor networks," *Computer Commun.*, vol. 30, pp. 2413–2427, Sep. 2007.

[21] Z. Yao, D. Kim, and Y. Doh, "PLUS: Parameterized and localized trust management scheme for sensor networks security," in *Proc. Third IEEE Int. Conf. Mobile Ad-Hoc and Sensor Systems (MASS'06)*, Oct. 2006, pp. 437–446.

[22] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Hoboken, NJ, USA: Wiley, 2010.

[23] X. Li, F. Zhou, and X. Yang, "Scalable feedback aggregating (SFA) overlay for large-scale P2P trust management," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 10, pp. 1944–1957, Oct. 2012.

[24] X. Li, F. Zhou, and X. Yang, "A multi-dimensional trust evaluating model for large-scale P2P computing," *J. Parallel Distrib. Comput.*, vol. 71, no. 6, pp. 837–847, Jun. 2011.

[25] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: Security protocols for sensor networks," *Wireless. Netw.*, vol. 8, no. 5, pp. 521–534, May 2002.

[26] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in *Proc. 10th ACM Conf. Computer and Comm. Security (CCS'03)*, 2003, pp. 62–72.

[27] C. Karlof, N. Sastry, and D. Wagner, "TinySec: A link layer security architecture for wireless sensor networks," in *Proc. Second Int. Conf. Embedded Networked Sensor Systems (SenSys'04)*, Nov. 2004, pp. 162–175.

[28] Z. Liang and W. Shi, "Analysis of recommendations on trust inference in open environment," *Perform. Evaluation*, vol. 65, no. 2, pp. 99–128, Feb. 2008.

[29] A. Whitby, A. Jøang, and J. Indulska, "Filtering out unfair ratings in bayesian reputation systems," in *The Autonomous Agents and Multi Agent Systems 2004*, New York, Jul. 2004.

Xiaoyong Li received the Ph.D. degree in computer science from Xi'an Jiaotong University in 2009.

Now, he is an associate professor of computer science at Beijing University of Posts and Telecommunications (BUPT). In 2009, he was awarded outstanding doctoral graduates in Shaanxi Province, China. In 2012, he was awarded New Century Excellent Talents in University, China. His current research interests mainly include cloud computing, network security, and trusted systems.

Feng Zhou received the M.S. degree in computer science from Beijing University of Posts and Telecommunications in 1989.

He is a full professor and the Director of the Center of Computer Architecture (CCR) at Beijing University of Posts and Telecommunications University. His research interests include mobile internet, embedded computing, and communication protocols. He is the author and coauthor of a high number of papers published in journals and conference proceedings.

Junping Du received the Ph.D. degree in computer science from Beijing University of Science and Technology in 1998.

She is currently a full Professor and Doctoral Supervisor of Computer Science at Beijing University of Posts and Telecommunications (BUPT). She is also the Director of the Computer Applications Center at BUPT. Her research interests include artificial intelligence and intelligent information systems. She has published more than 120 papers in journals and conferences.