Information Sciences 424 (2018) 118–136

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An adaptive virtual relaying set scheme for loss-and-delay sensitive WSNs

Anfeng Liu[a], Zhuangbin Chen[a], Neal N. Xiong[a,b,*]

[a] School of Information Science and Engineering, Central South University, ChangSha 410083 China
[b] Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA

ARTICLE INFO

Article history:
Received 22 January 2017
Revised 24 August 2017
Accepted 3 September 2017
Available online 13 September 2017

Keywords:
Wireless sensor networks
Relay selection
Virtual relaying set
Delay
Reliability
Network lifetime
Energy balance

ABSTRACT

In loss-and-delay sensitive wireless sensor networks (WSNs), especially when the duty cycles of nodes are extremely low, it is a challenge to ensure that data can be transmitted to sink nodes with high reliability and low delay. To address this problem, in this paper, we propose a data collection scheme named Adaptive Virtual Relaying Set (AVRS) where a set of relay nodes with more reliable connections to the sender node is selected to form its Virtual Relaying Set (VRS) to help transmit packets. In ring-based WSNs, each node in a VRS helps send packets in turn to the upper ring before the transmission is successful, or the packet is dropped if they all fail. Therefore, the larger the VRS (implying more retransmission chances), the higher the packet transmission reliability and the lower the delay will be. On the other hand, as the sender node has to stay active during the transmission stage, having a large VRS will cause huge energy consumption. Combined with the fact that the energy consumption of different parts of WSNs is unbalanced, nodes in the near-sink area (i.e., hotspots) have extremely high energy cost while nodes in the far-sink area (i.e., non-hotspots) still have ample energy remaining when the network dies. The main idea of the AVRS is the following rule. The size of the VRS of nodes is determined adaptively according to its energy usage pattern, making the size small in hotspots and relatively large in non-hotspots. The AVRS takes advantage of the residual energy of nodes in far-sink areas to achieve improved network data collection performance. Meanwhile, as nodes in near-sink areas have small VRS, the network can maintain a long lifespan without any reductions. Both theoretical analyses and simulative results demonstrate that the AVRS can improve data transmission reliability by more than 50% and reduce network delay by at least 33%.

1. Introduction

Wireless Sensor Networks (WSNs) are commonly used for environmental monitoring, surveillance operations, and home or industrial automation [1,3,8,16,19–21,24]. Together with current crowd-sensing networks and cloud computing [5,13,33,38,41], it is regarded as one of the most promising techniques for the future [12,14,22,25,37]. Since replacing or recharging sensor node batteries is extremely difficult, WSNs switch active and dormant states cyclically to save energy [27,42]. *When the sensor is in the active stage, it can perform all data operations, such as transmitting and receiving packets and*

* Corresponding author.
E-mail address: Xiong31@nsuok.edu (N.N. Xiong).

http://dx.doi.org/10.1016/j.ins.2017.09.036
0020-0255/

detecting and sensing environments [9,36,39]. In the dormant state, the sensor cannot perform any operations and just sleeps, but this state consumes 1000 times less energy than the active state [4,17,20]. From the view of saving energy, sensors should be set to the dormant stage as much as possible, but too long of a dormant period will lead to a decline in network performance, such as an increased probability of missed target detection, detection delay, and notification transmission latency. Sensor nodes in WSNs usually operate at a very low duty cycle style, implying that the sensor nodes sleep most of the time and stay active for only a short duration of each duty cycle [7,10]. Under such circumstances, when a sender node needs to deliver a message, it always has to wait until its destination node activates, thus resulting in sleeping latency. The sleeping latency in WSNs with large hop counts to sink nodes will lead to very long end-to-end delivery latencies. However, WSNs are usually applied to significant situations for detecting emergency incidents, such as fire and disasters, where a long end to end (e2e) delay may result in disastrous consequence. Thus, it is of great significance to be able to transmit sensed data to sink nodes rapidly. Owing to the inherent fallibility of wireless transmission, packets could be dropped during the collection process. Hence, another challenge is how one can guarantee that the sensed data can be reliably delivered to sink nodes. By selecting reliable relay nodes, sender nodes can have the best chance to transmit data. Many variants of this basic method have been proposed [2,29,30].

Currently, two general strategies are usually applied to enhance reliability. First is retransmission technology, which repeatedly transmits the failed packets, allowing nodes to provide more reliable communications. Second is increasing data transmitting power since increased power can improve the quality of network links, which in turn increases the probability of successful transmission. There are also two general methods for reducing transmission delay. First, by increasing the duty cycle of nodes, large duty cycles can shorten the interval of two adjacent active states, so nodes can transmit and receive packets earlier instead of waiting. Second, by increasing the size of the relaying nodes set, this can shorten the waiting time of sender nodes for its destination node (a node in the set) to become active, thus reducing the sleeping delay.

Lifespan is also a pivotal issue for WSNs. However, the aforementioned strategies all need to consume more energy, which may hasten the death of network. We observed that there is a special phenomenon called an "energy hole" in sensor networks [31]. To be more specific, nodes all send their sensed data to the sink node, which is at the center of the entire sensor network. However, this "many-to-one" data collection mode causes an imbalance. Nodes in the near-sink region help relay the packets generated by the outside nodes, so the amount of data they take is much greater than those in the far-sink area. Transmitting data is the main source of energy consumption, resulting in premature death of the nodes in hotspots and the network. Some related studies have shown that, due to the impact of the energy hole, there still remains up to 90% energy in the network when it dies [31]. Based on the above observation, we propose an innovative data collection scheme named Adaptive Virtual Relaying Set (AVRS) in this paper to achieve high network reliability and low delay without reducing the lifespan of networks by making use of the residual energy. The main contributions of our work are as follows:

(1) An Adaptive Virtual Relaying Set (AVRS) data collection scheme is proposed to correct for loss and delay in sensitive WSNs. Based on the unbalanced energy usage pattern, in an AVRS, a large number of nodes are selected to form a VRS in the far-sink region, while in the near-sink region fewer nodes are selected. This simple yet novel trick can greatly improve the overall data collection performance and, more importantly, not harm the network lifespan.
(2) The mathematical relations between the size of a VRS, energy consumption of nodes, data transmission reliability, and delay are provided. Based on these we design three algorithms (Section 4.3) that address the selection of the size of a VRS of nodes in different places in the network.
(3) Extensive simulations were conducted and the results were consistent with our theoretical analysis. In this paper, we measure the overall quality of the network using the weighting method, which uses the weighted sum of the nodal quality in proportion with the entire network. Both theoretical and simulative results demonstrate that the AVRS is efficient in enhancing both data collection reliability and delay under network lifespan constraints, which, on average, improves the network reliability by more than 50% and reduces transport delay by at least 33%.

The remainder of this paper is organized as follows. Section 2 reviews related works and compares them with our proposed scheme. Section 3 describes our network model and defines problem statements in the paper. Section 4 elaborates on the design of Adaptive Virtual Relaying Set (AVRS) for loss and delay in sensitive WSNs. In Section 5, the performances of AVRS are theoretically analyzed. Section 6 presents experimental results and comparisons. Finally, we conclude in Section 7.

2. Related works

In WSNs, lifespan, delay and reliability are the three most important properties that must be ensured. Much research has been done in this field [2,5,10,14,16,19,23,24,29,32,33,35,37,38,40]. This section briefly reviews some related works.

(1) *Large data transmitting power mechanism.* According to the basic principles of wireless communication, higher transmitting power leads to a larger signal-noise ratio, allowing easier packet reception. However, nodes will have larger energy consumption since their power increases, which shortens the lifespan of nodes. [40] obtained the values of data transmission reliabilities with different transmitting powers for different channel models (i.e., additive white Gaussian noise, Rayleigh fast fading and Rayleigh block-fading) in a linear network. Their conclusions are further verified using 2-dimensional Poisson networks using simulations. In [18], nodes in close-sink areas use higher transmitting power to improve reliability, whereas nodes in near-sink areas use relatively low power while guaranteeing

the lower bound of reliability requirements. It also prolongs network lifespan to the greatest extent and guarantees overall network reliability.

(2) *Retransmission-based mechanism.* Several approaches that achieve great data reliability have been proposed. The most profound one is the retransmission-based mechanism, such as the Automatic-Repeat-Request (ARQ) [32]. Once the transmitter sends a packet, it waits for an acknowledgment (ACK, a message sent by the receiver), indicating its data packet was correctly received. ARQ protocols include three basic schemes: Stop-and-wait (SW), Go-Back-N (GBN) and Selective Repeat (SR). With SW, the transmitter does not send any further packets until it receives an ACK signal or a timeout. While with GBN and SR, the transmitter sends packets continuously without the need to wait for individual ACKs from the receiver, with only the ACK for the last packet being required [32]. In [23], Liu et al. proposed a scheme to reduce the retransmission of packets at a very low energy cost by increasing the number of ACK transmissions [28]. Since ACKs contain only the serial number of a data packet, the length of an ACK message is much shorter compared to the data packet. Therefore, the transmission of data packets instead of ACKs is the main source of energy cost for a network. However, it may prolong the network delay since a node waits for more ACK messages. Network coding techniques are another reliability mechanism that is based on the redundant coding [11]. Although network coding can have a shorter delay than retransmission, coding and decoding packets require more computing resources and time.

(3) *Large duty cycle mechanism.* The duty cycle of a sensor node is defined as its active/working period [2,18]. [18] analyzed the relationship between network delay and duty cycle length, which can guide us on how to find the shortest duty cycles of nodes while guaranteeing network reliability to maximize the lifespan. [40] investigated the tradeoff between energy consumption and the latency of communications in a multi-hop wireless network using a realistic, unreliable link model. They proposed a method to find the optimal data transmission powers of nodes while meeting the lower bound of delay required by applications.

(4) *Relaying set selection strategy.* Naveen et al. [29,30] studied how to select the best forwarding nodes in an asynchronous wake-up node work model to reduce the end-to-end delay in WSNs. In this, each sensor node has several choices to send packets to its neighbors that wake up at different times in a duty cycle. If a sender node selects the first-wake-up node, though the packet is sent early, it may need more hops to reach the sink node since the first-wake-up node could be a far-sink node. However, if it gives up the first-wake-up node and chooses the next-wake-up one, it may be even further than the first node, so the e2e delay is worse. Therefore, Naveen et al. transformed it into a multi-objective optimization problem and found a comprehensive optimal solution. Similar research can also be seen in [25,26,29,30].

The work most closely related to ours is Cao et al. [2]. In [2], they designed a Robust Multi-Pipeline Scheduling (RMPS) algorithm that combines the staggered wakeup scheduling and the multi-parents forwarding scheme. However, they did not give analytical results for the relationship between network delay and reliability with different sizes of Virtual Forwarding Sets (VFS). More importantly, similar to all other previous studies, the RMPS algorithm cannot guarantee the lifespan while improving network reliability and delay. Hence, the AVRS proposed in this paper aims to fill this gap.

3. System model

3.1. Network model

(1) N homogeneous sensor nodes with sensing range rm are randomly deployed in a circular area with a density of ρ nodes per unit area. The circular area has a sink node (base station) situated at the center, whose radius is $\Re$m, as shown in Fig. 1.

(2) As the radio links between neighbor nodes are imperfect, failures of transmissions always exit the process of data collection. For a node v_i, the probability that successful data transmission originates from node v_i to its neighbor v_j is denoted as q_{ij}, which is determined by the link quality between them. In practice, the 4-bit estimator proposed in [6] addresses link dynamics by actively using packets and periodic beacons to measure link quality.

(3) To reduce the energy consumption of the system, sensors work in a low-duty-cycle mode with periodically alternating active and dormant states. Time is slotted and is fixed as γ seconds corresponding to a single packet transmission. Each sensor node selects one or several consecutive time slots as its wake-up time slot, whose length depends on the traffic data load. The transmitter serves new arrival packets on a FCFS (First Come First Serve) basis.

(4) In our network model, the distance between a certain node and the sink node is measured by hop count, with each hop at most rm [15,28,34]. Nodes with a hop count equaling k are named as ring k nodes, and thus the packet collection process can be regarded as being conducted in a ring-by-ring manner. For node v_i^k in ring k, its candidate relay nodes are defined as those within its sensing range and have shorter distances to the sink node than v_i^k. The nodes in the upper ring that could be a potential relayer to send a packet between ring k and ring $k-1$. All candidate relayers compose the Relaying Set (RS) of node v_i^k, as shown in the shadow region of v_i^k's sensing cycle in Fig. 1. In addition to the RS, each node also maintains a Virtual Relaying Set (VRS) that is a subset of the RS and contains the nodes with more reliable links to node v_i^k than those not within the VRS. A node in the VRS is also called the parent of node v_i^k. An example of this RS and VRS is depicted in Fig. 2. Node v_i and v_j in Ring 3 have selected some

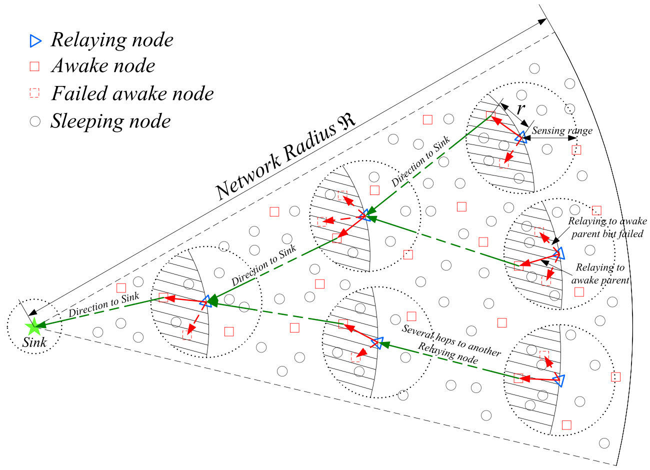

Fig. 1. Overall network model.

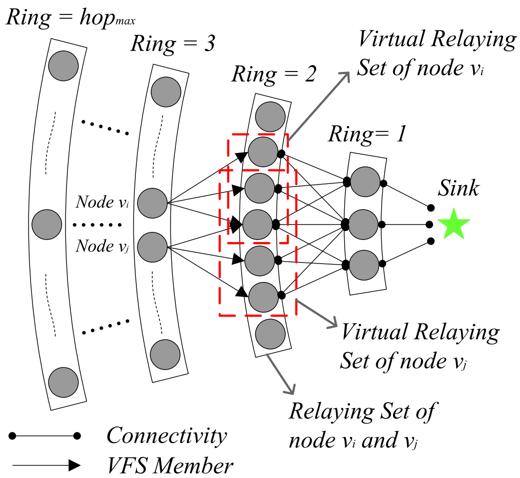

Fig. 2. An example for illustrating RS and VRS.

nodes as their parents in Ring 2 to deliver their sensed packets to the upper ring, i.e., Ring 2. Additionally, nodes in Ring 2 will choose their own VRS members to help them transmit their received packets until the packets have been transmitted to the sink node. The number of nodes in the VRS is adjustable in our scheme, which means the sender node can adjust its size of the VRS adaptively according to its data load to achieve a better network performance (lower transmission delay and higher reliability), while the network lifespan is guaranteed.

(5) The operation of the network system is broken into rounds, and each node has a probability of λ to generate a packet in each round. After generating a packet, sender v_i will select a node from its VRS that wakes up earliest as the relaying node to transmit its packet to the upper ring. If this fails, v_i will stay active, waiting for another node in the VRS to wake up and attempt to transmit its packet through this new-wake-up node. This continues until the packet is transmitted successfully or all nodes in the VRS fail to do so, with an unsuccessful packet dropped by v_i. As shown in Fig. 1, the solid red line in the sensing cycle represents successful packet transmission, while the dashed red line means transmission failure.

All nodes who receive the packet perform the above operation to transmit the packet to the sink node, which can be summarized as follows. After forming the VRS, a node will not stop transmitting a packet unless it is successful or it has been transmitted for the first R_{max} (R_{max} denotes the maximum times of retransmissions) times but all fail, then the packet will be dropped. The R_{max} of a certain node is assigned equally to the number of parents in its VRS. Intuitively, the more parents a node selects, the higher the one-hop reliability and the lower the one-hop delay, which can be proven in a later section.

Table 1
Network parameters.

Symbol	Description	Value
t_{duty}	Total duty cycle duration	100 ms
γ	Time slot duration	2 ms
ε_t	Transmission power consumption	0.0511 W
ε_r	Reception power consumption	0.0588 W
ε_s	Sleep power consumption	2.4×10^{-7} W
S_r	Preamble duration	0.26 ms
S_{ak}	Ack window duration	0.93 ms
$\hbar$	Duty cycle	*Calculation specific*
λ	Common network link quality	*Calculation specific*
δ	Common size of VRS	*Calculation specific*

Table 2
Notations.

Notation	Description
ω_{LPL}	The power required when performing the LPL operations
ω_r	The power used for receiving a packet
ω_t	The power used for transmitting an alert packet
ω_s	The power used when node is in sleep stage
ς	The expected number of one packet transmission times
ζ_t	The transmitting packet load of a node
ζ_r	The receiving packet load of a node
λ	Event production rate

3.2. Energy consumption model

For data collection, energy efficiency is a vital issue since replacing or recharging the battery of a sensor node is extremely difficult. To save energy, if the number of events is small, it would be better if nodes were set to sleep-wake mode to execute their tasks (the nodes can periodically switch on and off according to a normalized duty cycle). The duty cycle (denoted as $\hbar$) of a sensor node is defined as its active/working period ratio.

$$\eta = t_a/t_{duty} = t_a/(t_a + t_{off}), \tag{1}$$

where t_{duty} is the duty cycle time, t_a and t_{off} are the time durations when the sensor node is active and asleep, respectively. The duty cycle starts over between two consecutive periods. A node can detect events and communicate with its neighbor nodes when it is in its active stage. In contrast, it will close all communication units when it is in its dormant stage and cannot detect events or communicate with other nodes.

The power consumption of a node is mainly composed of (1) the power that is used to send or receive packets, (2) the power required to perform the Low-Power Listening (LPL) operations, and (3) the power consumed in its sleep state. The main parameters of the system model adopted in this paper are similar to those in [21,22]. The device-specific values are taken from the internal datasheet of a prototype sensor node by Thales. The values of these parameters are listed in Table 1. Table 2 summarizes the notations used in this paper.

3.3. Problem statement

Definition 1. End-to-end delay for event report (denoted D_{e2e}). D_{e2e} refers to the time between the generation of an event and the report packet to be transmitted to a sink node. Let d_i stand for the delay of the packet arrival at the i^{th} hop of multi-hop routing to the sink node. Then, the end-to-end delay minimization can be expressed as:

$$\min\,(D_{e2e}) = \min\left(\sum_{i \subseteq route} d_i\right). \tag{2}$$

Definition 2. Network lifetime (denoted ℓ). ℓ depends on the energy consumption speed of nodes. The energy consumed by node i consists of (a) communication energy (transmitting,e_t^i and receiving e_r^i, packet), (b) the energy needed in the Low-Power Listening (LPL) state, e_{LPL}^i, and (c) the energy consumed in the sleep state, e_s^i. Since the death time of the first node is defined as the network lifespan [23], maximizing the lifespan minimizes the energy consumption speed of the first node, which can be expressed by the following equation, where E_i represents the initial energy of node i:

$$\max\,(\ell) = \max \min_{0<i\leq n} \left(E_i/\left(e_t^i + e_r^i + e_{LPL}^i + e_s^i\right)\right). \tag{3}$$

Definition 3. Effective energy utilization (denoted ξ_e). ξ_e refers to the ratio of energy efficiently utilized and the total energy in the network, expressed as the following equation. Our target is to maximize the energy utilization of the whole network. e_i represents the energy consumption of node i:

$$\max(\xi_e) = \max\left(\left(\sum_{1\le i\le n} e_i\right)\Big/\left(\sum_{1\le i\le n} E_i\right)\right). \tag{4}$$

Definition 4. End-to-end data collection reliability (denoted ϕ_{e2e}). This should be higher than or at least equal to the minimum reliability, $\wp$ required by the application. ϕ_{e2e}^{j} refers to the end-to-end reliability of a data packet transmitted to the sink node from node j. Let β_i stand for the reliability of the packet at the i^{th} hop of multi-hop routing to the sink node. If there are n nodes in the network, then the e2e reliability can be expressed as:

$$\underset{j\in\{1,\ldots,n\}}{\phi_{e2e}^{j}} = \prod_{(i\in path\, j)} \beta_i \ge \wp \tag{5}$$

Obviously, the goal of the AVRS is to minimize the end-to-end delay D_{e2e}, maximize the network life ℓ, maximize effective energy utilization ξ_e, and maximize data collection reliability ϕ_{e2e}. This can be summarized as:

$$\begin{cases} Minimize\ D_{e2e}, Maxmize\ \ell, Maxmize\ \xi_e \\ \min\ (D_{e2e}) = \min\left(\sum\limits_{i\subseteq route} d_i\right) \\ \max(\ell) = \max \min\limits_{0<i\le n}\left(E_i/\left(e_t^i + e_r^i + e_{LPL}^i + e_s^i\right)\right) \\ \max(\xi_e) = \max\left(\left(\sum\limits_{1\le i\le n} e_i\right)\Big/\left(\sum\limits_{1\le i\le n} E_i\right)\right) \\ s.t.\ \underset{j\in\{1,\ldots,n\}}{\phi_{e2e}^{j}} = \prod\limits_{(i\in path\, j)} \beta_i \ge \wp \end{cases}. \tag{6}$$

4. Main design of AVRS

4.1. Research motivation

The research motivations of AVRS are primarily based on our comprehensive studies on WSNs below.

(1) In many previous works, nodes in near-sink regions consume much more energy when undertaking more packets, while nodes in the far-sink region have over 90% energy remaining when the network dies [23].
(2) If a node can have more relay nodes to help its packet transmission, then the expected reliability (i.e., ϕ_{e2e}) and transmission speed (i.e., D_{e2e}) can be improved. However, increasing the size of the VRS comes at the cost of increasing energy consumption, which may cause a decline in the network lifespan.

Moreover, the AVRS gives the mathematical relationship between the size of the VRS and energy consumption, network reliability and network delay, respectively, which has not been done by previous works. This can help determine the size of the VRS of different regions according to their energy consumption patterns. Therefore, we first evaluate the energy usage conditions and then introduce the AVRS algorithm.

4.2. Data load and energy consumption analysis

4.2.1. Data load of nodes

As depicted in Fig. 1, nodes in different areas of the network have different data loads. Nodes in close-sink areas should take the task of relaying packets generated by nodes in far-sink areas.

Theorem 1. In the AVRS, for a node whose distance from sink node is dm, its receiving and transmitting data load (denoted as ζ_r^d and ζ_t^d respectively) are:

$$\zeta_r^d = \left\{\sum_{j=1}^{z}\left[1-(1-\hat{\lambda})^{\delta}\right]^j + \frac{\sum\limits_{j=1}^{z}\left[1-(1-\hat{\lambda})^{\delta}\right]^j \cdot jr}{d}\right\}\cdot\lambda, \tag{7}$$

and

$$\zeta_t^d = \left\{ \sum_{j=1}^{z} \left[1-(1-\lambda)^{\delta}\right]^j + \frac{\sum_{j=1}^{z}\left[1-(1-\lambda)^{\delta}\right]^j \cdot jr}{d} + 1 \right\} \cdot \lambda \cdot \sum_{x=1}^{\delta} \Pr(x) \cdot x. \tag{8}$$

Proof. Generally, the probability $\Pr(n)$ that the packet transmission by node v_i fails at the first $n-1$ times while is successful at the n^{th} attempt is:

$$\Pr(n) = \prod_{j=1}^{n-1} \left(1-q_{ij}\right) \cdot q_{in}, \tag{9}$$

where q_{ik} is the link quality between node v_i and its selected kth relay node (its No.k parent in its VRS (see Section 5.2.1)).

Consider a sector ring region with width d_x, angle φ and dm apart from sink node, node v_d in this region is called the L_{v_d}-ring node, whose value can be easily calculated as $L_{v_d} = \lceil d/r \rceil$. Node v_d will undertake the packets generated by nodes in other z sector ring regions, which are $d+r, d+2r, \ldots, d+zr$(m) far from the sink node, respectively, where z is the exact integer that makes $d+zr$ less than $\Re$. $(L_{v_d}+1), (L_{v_d}+2), \ldots, (L_{v_d}+z)$ are the hop counts away from the sink node of sector ring regions $d+r, d+2r, \ldots, d+zr$, respectively. As sender nodes will try every parent in its VRS to relay its packet, the upper ring can successfully receive the packet if any one of these attempts is successful whose probability is $1-\prod_{k=1}^{\delta_{v_d}} (1-q_k^{v_d})$, where $q_k^{v_d}$ stands for the link quality from v_d to its No.k parent node and δ_{v_d} is the number of parents in v_d's VRS. The area of sector ring region $d+kr|k \in \{1,2,\ldots,z\}$ is $S_{d+kr} = \varphi(d+kr)d_x$, and the total number of nodes in this region is given by$N_{d+kr} = S_{d+kr}\rho = \varphi(d+kr)d_x\rho = \varphi(d+kr)\rho d_x$. For a certain node $v_{k,\,i}$ in this region, the probability that its packet can be received successfully by region d can be computed as $\prod_{y=1}^{k} (1-\prod_{x=1}^{\delta_{v_{k,i}}} (1-q_{y,x}^{v_{k,i}}))$, where $q_{y,x}^{v_{k,i}}$ stands for the link quality from the node relaying the packet to its No.x parent that will take the relay task of the packet at the y^{th} hop of multi-hop routing to the sink node and $\delta_{v_{k,i}}$ represents node $v_{k,\,i}$'s parents amount. Thus, we can obtain the amount of packets that region d receives from region $d+kr$ as $\sum_{i=1}^{N_{d+kr}} \prod_{y=1}^{k} (1-\prod_{x=1}^{\delta_{v_{k,i}}} (1-q_{y,x}^{v_{k,i}}))$, and we can further obtain the total number of packets that region d undertakes in one data round as $\sum_{k=1}^{z}\sum_{i=1}^{N_{d+kr}} \prod_{y=1}^{k} (1-\prod_{x=1}^{\delta_{v_{k,i}}} (1-q_{y,x}^{v_{k,i}}))$. Suppose that d_x is infinitesimal, then the data load of nodes in sector ring region d can be considered the same. As region d has $N_d = \varphi d\rho d_x$ nodes, the number of packets each node in this region undertakes can be calculated as:

$$\zeta_r^d = \frac{\sum_{k=1}^{z}\sum_{i=1}^{N_{d+kr}}\prod_{y=1}^{k}\left(1-\prod_{x=1}^{\delta_{v_{k,i}}}\left(1-q_{y,x}^{v_{k,i}}\right)\right)}{\varphi d\rho d_x}. \tag{10}$$

For the convenience of the calculation, link quality and the size of VRS will be set at two common values, which are λ and δ, respectively. Therefore, ζ_r^d can be rewritten as:

$$\zeta_r^{d*} = \frac{\sum_{j=1}^{z}\left[1-(1-\lambda)^{\delta}\right]^j \cdot N_{d+jr}}{\varphi d\rho d_x} = \sum_{j=1}^{z}\left[1-(1-\lambda)^{\delta}\right]^j + \frac{\sum_{j=1}^{z}\left[1-(1-\lambda)^{\delta}\right]^j \cdot jr}{d}. \tag{11}$$

The above calculation is based on the precondition that each node will generate one packet in a duty cycle. However, in our network model, the event production rate is λ, thus the actual value of ζ_r^d should be multiplied by the event generation probability that is given as Eq. (7). λ can be treated as a variable and, in practice, is determined by the number of nodes and detection targets of the network. Generally, the larger λ, the more energy consumed for transmitting more data packets during the data collection stage. In this way, λ will affect the network lifespan.

As for the amount of transmitting packets in one round (denoted ζ_t^d), an extra packet generated by itself should be considered. We can calculate its value as:

$$\zeta_t^{d*} = \zeta_r^d + \lambda = \left\{ \sum_{j=1}^{z}\left[1-(1-\lambda)^{\delta}\right]^j + \frac{\sum_{j=1}^{z}\left[1-(1-\lambda)^{\delta}\right]^j \cdot jr}{d} + 1 \right\} \cdot \lambda. \tag{12}$$

As previously discussed, if a node transmits a packet, it will not stop the transmission unless the packet is transmitted successfully or the first R_{max} attempts fail. The expectation times $\varsigma_\delta^{v_d}$ of transmitting a single packet are calculated in later Section 5.1.2 and the exact value of ζ_t^d should be multiplied by $\varsigma_\delta^{v_d}$, which is shown in Eq. (8). ■

4.2.2. Energy consumption

This section provides the energy cost calculations of nodes in different positions. The values are given by theoretical analysis. To clearly express the idea of this paper, instead of focusing on the specific calculations, two reasonable assumptions will be made. One assumption is that all of the nodes in the network adopt the same duty cycle, $\hbar$. The second assumption is that every node has the same link quality, λ, with any one of its parents. After calculating the energy consumptions of nodes, the size of the VRS can be adjusted according to its residual energy.

Theorem 2. The total energy consumption of the node at *d*m away from the sink node in the network can be calculated by:

$$E_{tot}^{d} = \varpi_{LPL}^{d} \cdot t_{duty} + \varpi_{r}^{d} \zeta_{r}^{d} + \varpi_{t}^{d} \zeta_{t}^{d}. \tag{13}$$

Proof. ϖ_t^d, ϖ_r^d, and ϖ_{LPL}^d are the required powers for transmitting an alert packet, receiving a packet, and the LPL operations, respectively.

The average power of ϖ_t^d can be computed as:

$$\varpi_t^d = \varepsilon_t S_p + \frac{(1-\hbar)t_{duty}}{2(S_r + S_{ak})}(\varepsilon_t S_r + \varepsilon_r S_{ak}). \tag{14}$$

The first term on the right side of the above equation, $\varepsilon_t S_p$, is the energy spent for transmitting data in a packet. To notify the receiving node of the packet's arrival, the second term corresponds to the energy consumption for the periodic preamble transmission [21].

The power for receiving a packet can be calculated as:

$$\varpi_r^d = \varepsilon_r S_p + (\varepsilon_r S_r + \varepsilon_t S_{ak}), \tag{15}$$

where $\varepsilon_r S_p$ is the energy used for receiving a packet and $(\varepsilon_r S_r + \varepsilon_t S_{ak})$ is for the reception of the preamble and the transmission of the ACK message [21].

According to [21], the power associated with LPL operations can be calculated as:

$$\varpi_{LPL}^d = \varepsilon_r \hbar + \varepsilon_s (1-\hbar) - \tau_t^d - \tau_r^d. \tag{16}$$

Subtracting τ_t^d and τ_r^d from ϖ_{LPL}^d is necessary since, in the active state, some energy is used for data transmission and reception, and this part has been calculated in Eqs. (14) and (15). Obviously, the nodes closer to the sink node load use more data, thus the time duration for sending and receiving data is increased while it is decreased for LPL operations. ϖ_{LPL}^d will be smaller with the increments of τ_t^d and τ_r^d.

According to [21], τ_t^d can be calculated as:

$$\tau_t^d = \left\{ \varepsilon_s \left[\frac{(1-\hbar)t_{duty}}{2} + S_r + S_{ak} \right] + \varepsilon_r S_r \right\} \cdot \frac{\zeta_t^d}{t_{duty}}. \tag{17}$$

According to [21], τ_r^d can be calculated as:

$$\tau_r^d = [(S_{ak} + S_p)\varepsilon_s + \varepsilon_r S_r] \cdot \frac{\zeta_r^d}{t_{duty}}. \tag{18}$$

Eq. (13) calculates the total energy consumption. ■

We know from Eq. (16) that the energy consumption for a node to perform LPL operations is equal to its total energy consumption in its active state, subtracting two parts of energy used for packet transmission and reception. Note that the energy consumption for a node to perform LPL operations does not always increase with network link quality. Poor link quality will decrease the probability of collecting data, reducing the total number of packet reception and retransmission in one round, which makes the time duration of performing LPL operations increase in the active state and consume more energy accordingly. Eq. (15) reflects that the energy consumption for packet reception merely relates to the number of packets. From Eq. (7), nodes receive more packets with the network link quality, which increases the energy consumed by a node to receive packets in one round.

4.3. AVRS algorithms

In this section, the AVRS algorithm addressing the size selection of every network node in the VRS is introduced. The AVRS algorithm mainly consists of three stages: *Preparing, Searching* and *Tuning*. An overall setting of the variables in algorithms is given in Table 3.

In Algorithm 1, we first set the size of each node of the VRS as identical to the RMPS [2], which is the lowest bound of reliability of the application. The rationale is that a small size of VRS leads to low energy consumption, and thus a long network lifespan can be archived. The determined original size of the VRS will be stored in an array named *sizeOfVRSArray*, whose index stores the original ID of each node in the network. Then, we calculate the energy consumption of every node using Eq. (13) and save the values and original ID of every node into another array named *energyConsArray,* whose element

Table 3
Algorithm variable setting.

Variable	Description
sizeOfVRSArray	Stores the size of the VRS of each node in the network. The ID of each node will be stored as the index of this array.
largestEnergyCons	Stores the largest energy consumption of node in the network.
energyConsClass	It is a class that has two attributes. One is *energyCons*, used to store the energy consumption. The other is *OriginalID*, used to store the original ID of each node (The function of *OriginalID* will be explained later).
energyConsArray	Stores the energy consumption and original ID of each node.

Algorithm 1 Preparing.

```
For each node v_i in network Do
   sizeOfVRSArray[i] := the lower bound of size of VRS;
   // The lower bound of each node's size of VRS is determined
   // by the same strategy as RMPS.
End for
For each node v_i in network Do
   energyConsArray[i].OriginalID := i;
   energyConsArray[i].energyCons := the energy consumption of node v_i;
 // Using Eq. (13) to calculate the energy consumption of node v_i with each node in network
 // has the corresponding size of VRS in sizeOfVRSArray (the same hereinafter).
End for
Sort the array energyConsArray in the descending order of energy consumption;
largestEnergyCons := energyConsArray[0].energyCons;
// Obviously, the first object of this array stores the largest
// energy consumption and the ID of node in network.
```

Algorithm 2 Searching.

```
For each member i in sizeOfVRSArray Do
   If i != energyConsArray[0].OriginalID then
      sizeOfArray[i] := the threshold size of VRS;
      // The threshold size of VRS means the biggest
      // number of parents a node can have in its VRS.
   End if
End for
While (If the energy consumption of any one of the nodes is larger than largestEnergyCons)
   For each node i in network Do
      energyConsArray[i].energyCons := the energy consumption of node v_i;
   End for
   Sort the array energyConsArray in the descending order of energy consumption;
   For each object i in energyConsArray Do
      If energyConsArray[i].energyCons > largestEnergyCons then
         IndexID := energyConsArray[i].OriginalID;
         If sizeOfVRS[IndexID] > 0 then
            sizeOfVRS[IndexID] := sizeOfVRS[IndexID] − 1;
         End if
   // Here shows the function of OriginalID, we can only find the original node through it instead of
   // the index of the array, because the sequence of the array has changed after sorting.
      End if
      Else
         Exit for loop;
      End else
   End for
End while
```

is a previously defined class named *energyConsClass*. Next, we find the node with the largest energy consumption and store its energy into a variable called *largestEnergyCons*. Finally, we sort *energyConsArray* in descending order of the attribute *energyCons*. With another attribute, *originalID*, unchanged, we can trace the energy consumption changing process for each node in the network.

Algorithm 2 adjusts the size of the VRS. After sorting *energyConsArray* in Algorithm 1, *energyConsArray*[0] stores the node with the largest energy cost in the network. We set the size of the VRS of every node, except for the one in *energyConsArray*[0], at the threshold size of VRS, which is the largest number of parents a node can have in its VRS (since VRS cannot be infinite, we set a threshold size of VRS for each node). In practice, the deployment of the network allows each node find sufficient neighbors to form its VRS easily. However, if a node fails to do so, it will just use all neighbors it can find. Then,

Algorithm 3 Preparing.

```
For each element i in energyConsArray Do
  If energyConsArray[i].energyCons < largestEnergyCons then
    IndexID := energyConsArray[i].OriginalID;
    While sizeOfVRS[IndexID] < threshold size of VRS Do
      sizeOfVRS[IndexID] := sizeOfVRS[IndexID] + 1;
      For each node v_j in network Do
        energyConsArray[j].energyCons := the energy consumption of v_j;
        If energyConsArray[j].energyCons > largestEnergyCons then
          sizeOfVRS[IndexID] := sizeOfVRS[IndexID] - 1;
          Exit while loop;
        End if
      End for
    End while
  End if
End for
```

we recalculate the energy consumption of nodes and sort the array *energyConsArray* again. Since the energy consumption of nodes is greatly increased, they can exceed *largestEnergyCons*, which does harm the network lifespan. This causes the size of VRS to need adjustment.

It is noteworthy that the energy consumption of each node is not independent of others. That is, if neighbors change their size of the VRS, a node v_i may have different energy consumption even its number of parents stays the same since the neighbors' VRS will increase the successful packet receiving probability of v_i. Therefore, the node in *energyConsArray*[0] also should reduce its VRS, which is necessary to maintain network lifespan.

Based on the above analysis, the adjustment strategy is as follows. Scan *energyConsArray* from the head to end. If *energyConsArray*[i].*energyCons* $(1 \le i < n)$ is larger than *largestEnergyCons*, then the VRS of the node whose original ID equals *energyConsArray[i].originalID* decreases by one if it is still larger than 0. Since the attribute *originalID* always remains the same, the changing process of every node can be saved into *energyConsArray*. After adjusting for the first round, compute the energy consumption again. If there is a node consuming more energy than *largestEnergyCons*, then a second round of adjustments should be conducted, and so forth, until the energy consumption of all nodes is less than or equal to *largestEnergyCons*.

After the adjustment of Algorithm 2, the size of the VRS of nodes can be generally determined. However, as discussed before, energy consumption of nodes is dependent on others. There will be some deviation, which means the general VRS setting may not be the optimal solution. To obtain a more precise VRS size, a tuning phase is necessary, as shown in Algorithm 3. For every element in array *energyConsArray*, if *energyConsArray*[i].*energyCons* $(1 \le i < n)$ is less than *largestEnergyCons*, then the VRS of the node whose original ID equals to *energyConsArray[i].originalID* increases by one if it is still less than the threshold size of the VRS. Next, calculate the energy consumption. If any one of the nodes consumes more energy than *largestEnergyCons*, then cancel the adjustment on the VRS. Otherwise, increase the VRS by another one and repeat the above process.

5. Performance analysis in theory

In this section, we provide the theoretical performance of our proposed AVRS algorithm under different network settings. Similar to Section 4.3, assume that setting the nodes' original size of VRS at 5 can meet the network reliability requirement and the threshold size of the VRS is set at 12.

5.1. Network reliability analysis

5.1.1. The reliability with link quality

The size of the VRS reflects the tradeoff consideration between energy cost and data reliability. With a small VRS, as a node will try fewer times to transmit its packet to the upper ring, the energy can be saved, but the reliability of this hop cannot be guaranteed. With a large VRS, increased retransmission will be attempted, so a high reliability can be achieved, but it consumes more energy, reducing the lifespans of nodes which is unwanted.

The VRS of a node can be decided based on its link quality to nodes in the RS and predetermined one-hop packet delivery reliability, which can be configured as a system parameter. Suppose the threshold of one-hop delivery reliability that an application requires is given as ϕ. Then, the size of the VRS and which node should be in the set can be determined by the following strategy.

First, node v_i sorts the links to all neighboring nodes in the RS. Assume the link quality set is $\{q_{i1}, q_{i2}, \ldots, q_{in}\}$, where $q_{i1} \ge q_{i2} \ge \cdots \ge q_{in}$. The probability that the packet is successfully transmitted at least once within the first j attempts can be calculated as $1-(1-q_{i1})(1-q_{i2})\cdots(1-q_{ij})$. To satisfy the threshold of one-hop reliability, the lower bound of the size of

VRS, which equals δ, should meet both of the following two formulas:

$$1 - \prod_{j=1}^{\delta-1} \left(1 - q_{ij}\right) < \phi, \tag{19}$$

and

$$1 - \prod_{j=1}^{\delta} \left(1 - q_{ij}\right) \geq \phi, \tag{20}$$

where q_{ij} denotes the link quality from node v_i to its parents v_j, $1 \leq j \leq \delta$ in VRS, respectively.

5.1.2. The reliability with size of VRS

In this section, we calculate the expected one packet retransmission times versus different size of VRS and the corresponding data reliability.

Theorem 3. In AVRS, for a node v_i with δ parents in VRS, the retransmission times of its packets can be calculated as:

$$\varsigma_{\delta}^{v_i} = \sum_{x=1}^{\delta} \Pr(x) \cdot x = \sum_{x=1}^{\delta} \left(\prod_{j=1}^{x-1} \left(1 - q_{ij}\right) \cdot q_{ix} \cdot x \right). \tag{21}$$

It can also be written as:

$$\varsigma_{\delta}^{v_i} = \sum_{x=1}^{R_{max}} \Pr(x) \cdot x = \sum_{x=1}^{R_{max}} \left(\prod_{j=1}^{x-1} \left(1 - q_{ij}\right) \cdot q_{ix} \cdot x \right), \tag{22}$$

■

According to Eqs. (19) and (20), we can calculate the one-hop reliability after deciding the size of the VRS. Suppose all nodes in the network have a common size of VRS, δ, as a packet generated by node v_i will be transmitted for at most δ times. The probability that it can be delivered successfully to the upper ring is:

$$p^{\delta} = 1 - \prod_{j=1}^{\delta} \left(1 - q_{ij}\right). \tag{23}$$

where q_{ij} stands the link quality from node i to its No.j parent node.

5.1.3. Reliability of AVRS

This section provides the reliability of our proposed AVRS algorithm in terms of one-hop reliability, end-to-end reliability and network weighted e2e reliability.

Theorem 4. In AVRS, the e2e data collection reliability of node whose distance to Sink is xm can be calculated as:

$$p_{e2e}^{x} = \prod_{i=1}^{z} p_i^{\delta_a^x}, \tag{24}$$

where $z = \lfloor (\Re - x)/r \rfloor$.

Proof. p_i^{δ} is the probability of successful packet transmission at the i^{th} hop of multi-hop routing to the sink node when the node has δ parents. This can be computed by Eq. (23). δ_a^x is the adjusted size of the VRS of the node, which is xm far away from the sink node.

The e2e reliability is defined as the probability that a packet can be successfully received by the sink node after several hops of transmission from the source node. Therefore, its value is the product of the packet's reliability of every single hop along the routing path to the sink node. ■

Theorem 5. In AVRS, we denote the network weighted reliability of end-to-end data collection as p_w^a, which can be calculated as follows:

$$p_w^a = \int_0^{\Re} \int_0^{2\pi} p_{e2e}^{x} \cdot x \cdot dx \cdot d\theta, \tag{25}$$

where p_{e2e}^{x} is the e2e data collection reliability of node whose distance to sink node is xm.

Proof. In the place where the distance from the network center is $x | x \in \{0, 1, ..., \Re\}$, we take a fraction of the fan-shaped ring with an angle $d\theta$ and a width of dx. The area of this region is $x \cdot dx \cdot d\theta$. The probability of packet collection of the

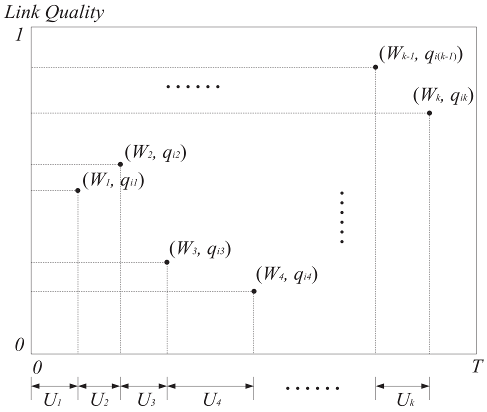

Fig. 3. Illustration for sorted nodes in VRS.

network can be expressed as $x \cdot dx \cdot d\theta \cdot p^{x}_{e2e}$. Integral to the entire region, the weighted reliability of e2e data transmission is as follows:

$$p^{a}_{w} = \int_{0}^{\Re} \int_{0}^{2\pi} p^{x}_{e2e} \cdot x \cdot dx \cdot d\theta .$$

■

5.2. Delay analysis in theory

5.2.1. Delay with the size of VRS

A small probability event may occur when the first R_{max} all fail. In this case, the packet would be dropped by node v_i. Note that the probability that the packet is transmitted successfully at the n^{th} attempt is under the condition that the packet is delivered successfully within the first R_{max} attempts. The conditional probability can be represented as:

$$\Pr(n)_{Cond} = \frac{\prod_{j=1}^{n-1}\left(1-q_{ij}\right)\cdot q_{in}}{1-\prod_{j=1}^{R_{max}}\left(1-q_{ij}\right)}. \tag{26}$$

Assume a parent node k where $1 \le k \le \delta_i$ in the VRS of node v_i becomes available to v_i at T_k. Node k wakes up at time instant T_k, which is called the *wakeup instant* of k. We have $T_m \ne T_n$ for any $m \ne n$, where $m, n \in \{1, 2, \ldots, \delta_i\}$, because simultaneous wakeup times can be avoided by utilizing a technique called *on-the-fly shifting* [2]. Let $W_1, W_2, \ldots, W_{\delta_i}$, represent the increasing order statistics of $T_1, T_2, \ldots, T_{\delta_i}$. The $\{W_x\}$ sequence is the $\{T_y\}$ sequence sorted in increasing order. After sorting the parent node according to their *wakeup instant*, the node whose *wakeup instant* ranks k among all nodes in VRS will be named the No.k parent node of v_i. We define $U_{k-1} = W_k - W_{k-1}$ for $k = 2, 3, \ldots, \delta_i + 1$, particularly $W_{\delta_i+1} = W_1 + T$. U_k is the wakeup time instant between the two consecutive nodes, as shown in Fig. 3.

Suppose node v_i wakes up at time instant Θ_k. For $1 \le k \le \delta_i$, k indicates $\Theta_k \in [W_k, W_{k+1})$, the delay to reach its 1^{st} parent node $k+1$, whose *wakeup instant* is W_{k+1}, will be $W_{k+1} - \Theta_k$. The delay to reach the 2^{nd} parent node is $W_{k+2} - \Theta_k$, i.e., $W_{k+1} - \Theta_k + (W_{k+2} - W_{k+1})$, as shown in Fig. 4. Generally, the delay for a packet to be delivered at the n^{th} attempt can be calculated as:

$$d_n = \begin{cases} W_{k+n} - \Theta_k,\ for 1 \le n \le (\delta_i - k + 1) \\ W_{n-k+1} - \Theta_k + T,\ for (\delta_i - k + 1) < n \le \delta_i \end{cases}. \tag{27}$$

Additionally, we can compute the probability that the packet is transmitted from v_i to its $|k+n|_{\delta_i}$ parent node successfully at n^{th} attempt:

$$\Pr(k, n)_{Cond} = \frac{\prod_{j=1}^{n-1}\left(1-q_{i|k+j|_{\delta_i}}\right)\cdot q_{i|k+n|_{\delta_i}}}{1-\prod_{j=1}^{R_{max}}\left(1-q_{ij}\right)}, \tag{28}$$

where $|k+n|_{\delta_i}$ is the modular parent node serial number and can be defined as:

$$|k+n|_{\delta_i} = \begin{cases} k+n,\ for (k+n) \le \delta_i \\ (k+n) - \delta_i,\ for (k+n) > \delta_i \end{cases}, \tag{29}$$

and $q_{i|k+j|_{\delta_i}}$ represents the link quality between v_i and its $|k+n|_{\delta_i}$ parent node in its VRS.

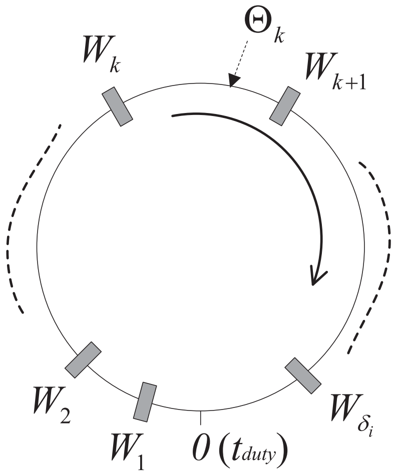

Fig. 4. Time cycle schedule.

Thus, the expectation delay for a packet from node v_i to reach upper ring can be written as:

$$D_{i,k} = \sum_{j=1}^{R_{max}} d_j \Pr(k, j)_{Cond}. \tag{30}$$

Additionally, for the convenience of calculation, we assume that $U_1 = U_2 = \cdots = U_{\delta_i} = \chi = \frac{t_{duty} - \gamma \cdot \delta_i}{\delta_i}$ and $q_{i1} = q_{i2} = \cdots = q_{i\delta_i} = \hat{\lambda}$, then $d_n = (W_{k+1} - \Theta_k) + (n-1)\chi,\ 1 \le n \le \delta_i$; therefore, the one-hop expectation delay can be rewritten as:

$$\begin{aligned} D_{i,k} &= (W_{k+1} - \Theta_k) \cdot \Pr(k, 1)_{Cond} + \sum_{j=2}^{R_{max}} (j-1)\chi \cdot \Pr(k, j)_{Cond} \\ &= (W_{k+1} - \Theta_k) \cdot \frac{\hat{\lambda}}{1 - (1-\hat{\lambda})^{R_{max}}} + \sum_{j=2}^{R_{max}} \left((j-1)\chi \cdot \frac{(1-\hat{\lambda})^{j-1}\hat{\lambda}}{1-(1-\hat{\lambda})^{R_{max}}} \right), \end{aligned} \tag{31}$$

We can further obtain the end-to-end delay for this packet to reach sink node as:

$$D_{e2e} = \sum_{i \subseteq route} D_i, \tag{32}$$

where D_i stands for the delay of the packets at the i^{th} hop of multi-hop routing to sink node.

5.2.2. Delay of AVRS

In this section, the e2e delay of AVRS is provided.

Theorem 6. In the AVRS, we denote the network weighted end-to-end data collection delay as D_w^a, which can be calculated as follows:

$$D_w^a = \int_0^{\Re} \int_0^{2\pi} \left(\sum_{i=1}^{z} D_i^x \right) \cdot x \cdot dx \cdot d\theta, \tag{33}$$

where D_i^x stands for the expectation delay of the packet at the i^{th} hop of multi-hop routing to sink node whose distance to the sink node is xm.

Proof. In the place where the distance from the sink node is $x|x \in \{0, 1, ..., \Re\}$, take a fraction of the fan-shaped ring Φ with an angle $d\theta$ and a width of dx. The area of this region is $x \cdot dx \cdot d\theta$. The e2e delay of packet collection of the network can be expressed as $x \cdot dx \cdot d\theta \cdot (\sum_{i=1}^{z} D_i^x)$. Integral to the entire region, the weighted delay of e2e data transmission is as follows:

$$D_w^a = \int_0^{\Re} \int_0^{2\pi} \left(\sum_{i=1}^{z} D_i^x \right) \cdot x \cdot dx \cdot d\theta.$$

■

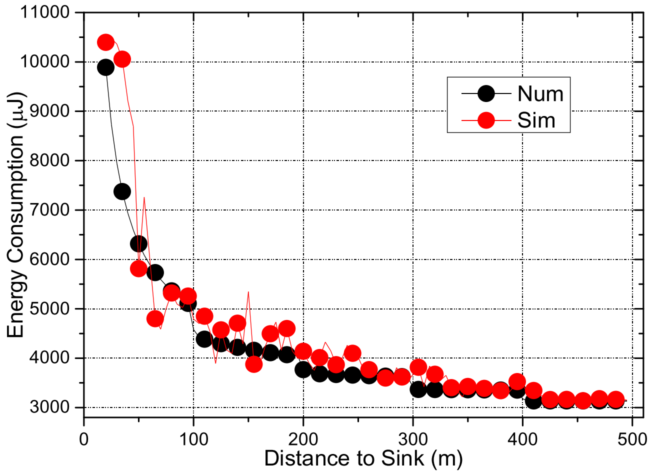

Fig. 5. Energy consumption of the numerical and simulative analysis with $\delta = 3$.

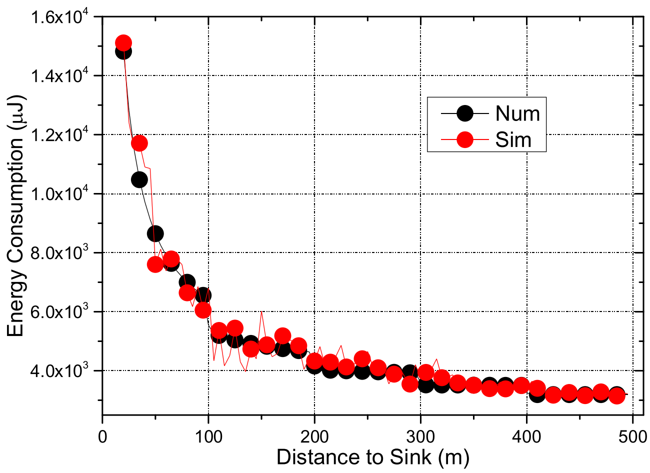

Fig. 6. Energy consumption of the numerical and simulative analysis with $\delta = 5$.

6. Experimental results and analyses

OMNET ++ is employed for experimental verification [31]. Without losing generality, the network parameters are radius $\Re = 500\,\text{m}$, packet transfer radius $r = 100\,\text{m}$, event generation probability $\lambda = 0.1$, and 1000 nodes. The VRS setting is the same as the theoretical analysis where the nodes' original size of the VRS is 5 and the threshold size of the VRS is 12. For other parameter settings, refer to Table 1.

6.1. Theoretical and experimental results of energy consumption

This section provides some numerical and experimental results of energy consumption for all stages with the AVRS.

Figs. 5 and 6 are two comparisons between theoretical and simulative results of the energy consumption of nodes. We can see the theoretical results fit the simulation results very well.

In influence of the common size of the VRS on the energy consumption of nodes is shown in Fig. 7. A larger VRS makes nodes consume more energy, but the difference is minimal.

Similarly, Fig. 8 provides the results of how network link quality affects the energy consumption of nodes. Similar to the theoretical analysis in Section 5, the energy consumption does not have a positive relationship with link quality, and we have the highest energy consumption when link quality is 0.5.

Finally, the energy consumption used for data transmission in different regions of the network is given in Fig. 9. With the AVRS algorithm, the energy consumption of all nodes increases apart from those in the close-sink area. However, energy consumption is not completely balanced. When the network is dead, there is still much remaining energy in the far-sink area.

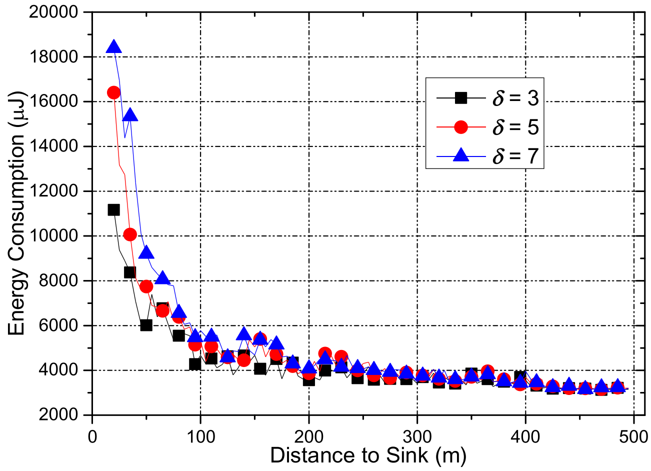

Fig. 7. Energy consumption of nodes with $\lambda = 0.5$.

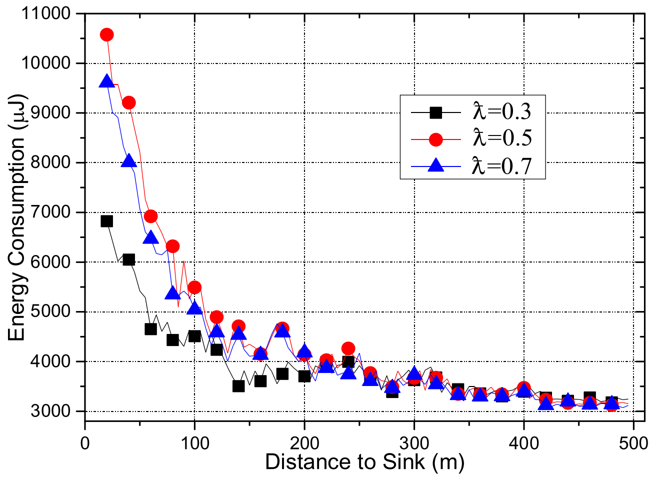

Fig. 8. Energy consumption of nodes with $\delta = 3$.

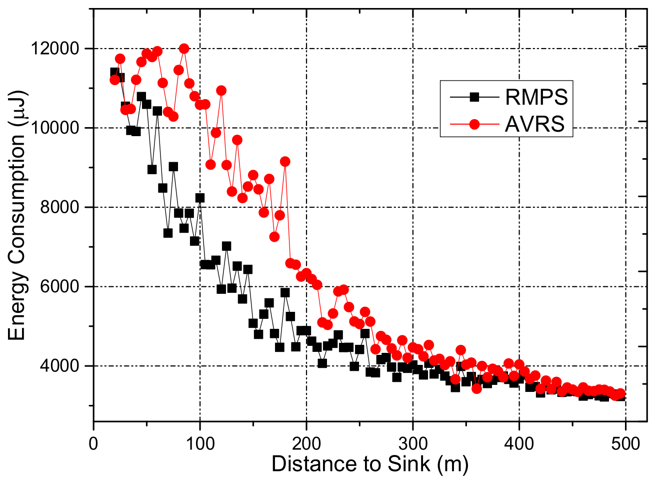

Fig. 9. Energy consumption of nodes.

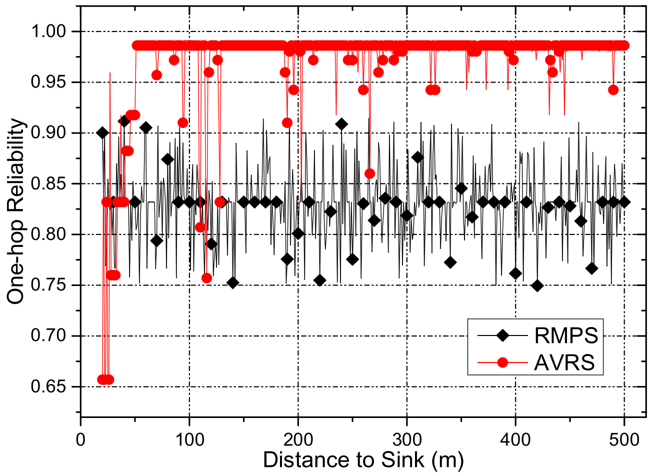

Fig. 10. Simulative comparison of the one-hop packet delivery reliability with $\lambda = 0.3$.

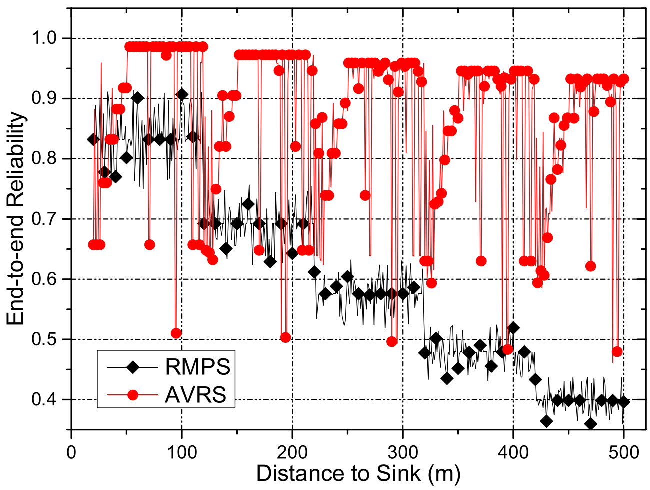

Fig. 11. Simulative comparison of the e2e packet delivery reliability with λ= 0.3.

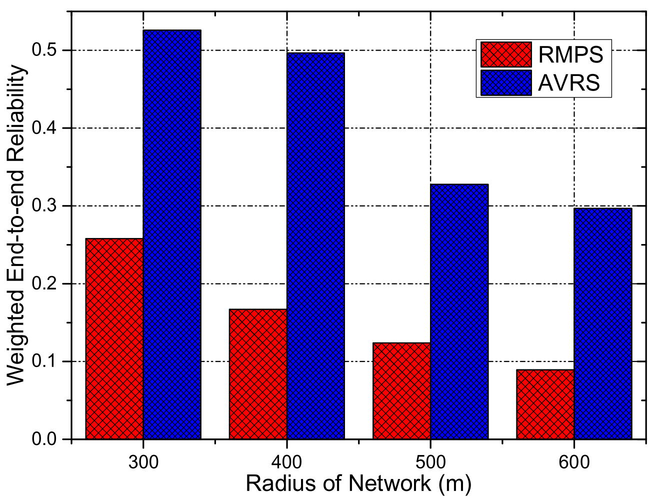

Fig. 12. Weighted e2e reliability versus different sizes of networks with $\lambda = 0.3$.

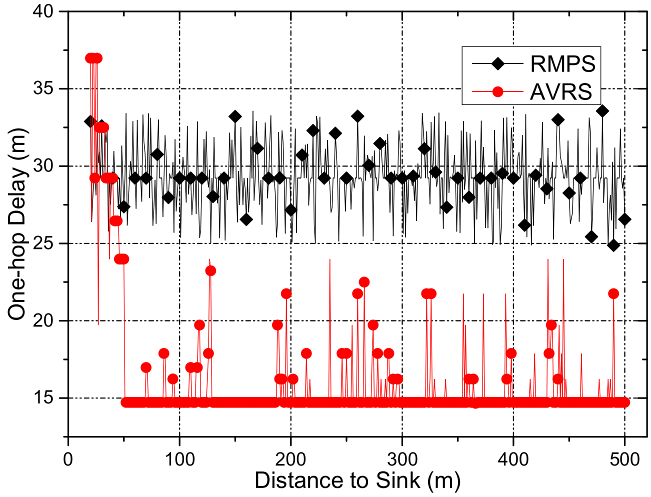

Fig. 13. Simulative comparison of the one-hop packet delivery delay with $\lambda = 0.5$.

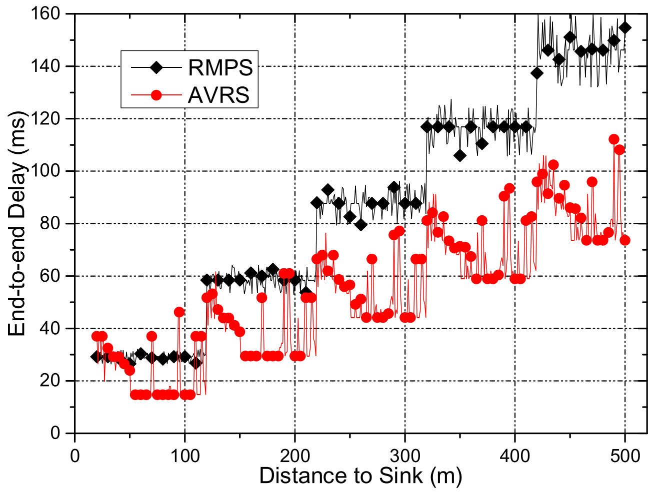

Fig. 14. Simulative comparison of the e2e packet delivery delay with $\lambda = 0.5$.

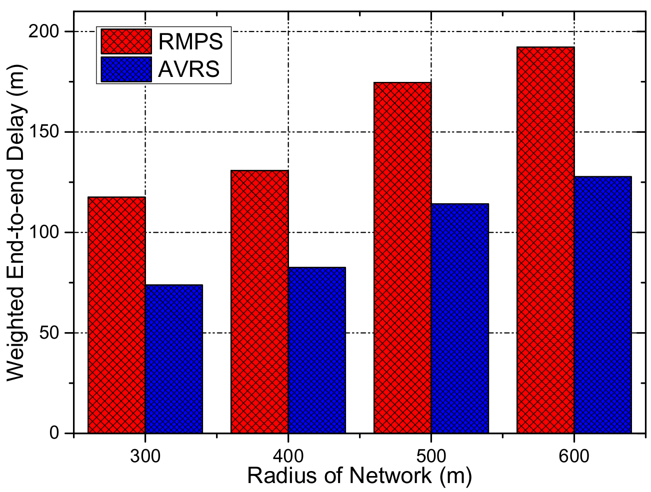

Fig. 15. Weighted e2e delay versus different sizes of networks with $\lambda = 0.3$.

6.2. The reliability of AVRS

Figs. 10 and 11 show the simulative results of one-hop and end-to-end reliability with different regions of the network, respectively. In the near-sink area, the reliability performance of AVRS is worse than that of the RMPS, but it is much better in the far-sink area. The results are consistent with the theoretical analyses.

The weighted e2e reliability is shown in Fig. 12, which reflects the universality of enhanced reliability with the AVRS. Compared with the RMPS, the AVRS can save the weighted e2e reliability by 50.9%−69.9%.

6.3. The delay of AVRS

Figs. 13 and 14 show the experimental results of one-hop and end-to-end reliability in different places of the network, respectively. Obviously, these two kinds of delay are reduced in most of the entire network. The comparative results of the network's weighted e2e delay between the AVRS and RMPS are shown in Fig. 15, which shows that the weighted e2e delay can be decreased on average by 33.5%−37.1%. This fully illustrates the effectiveness of the AVRS.

7. Conclusion

In WSNs, high data transmission reliability and low delay are the two most important aspects for many applications. However, both of them come at the cost of increased energy consumption, which will harm network lifespan. Thus, it can be a challenge to implement them while concurrently maintaining a long lifespan.

In this paper, we transformed this problem into a trade-off optimization problem and proposed a novel data collection scheme named Adaptive Virtual Relaying Set (AVRS) for loss-and-delay sensitive WSNs with long lifespan guaranteed. The essential reason why the AVRS has good performance is that it takes full advantage of a network's residual energy. In WSNs, owing to the large amount of unused energy, more nodes in the non-hotspots area can be selected to form a larger VRS, which will consume more energy but not reduce the network lifespan. More importantly, this trick can improve transmission reliability and reduce delay.

We further give the mathematical relations between packet transmission reliability and delay with different sizes of the VRS of nodes. We analyze and prove the correction and effectiveness of the AVRS in theory. Extensive simulations are conducted to show that the AVRS significantly outperforms the existing schemes in terms of average end-to-end transmission delay and reliability without declining network lifespan.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (61772554, 61379110, 61572528, 61572526), The National Basic Research Program of China (973 Program) (2014CB046305).

References

[1] H. Asharioun, H. Asadollahi, T.C. Wan, et al., A survey on analytical modeling and mitigation techniques for the energy hole problem in corona-based wireless sensor network, Wirel. Personal Commun. 81 (1) (2015) 161–187.
[2] Y. Cao, S. Guo, T. He, Robust multi-pipeline scheduling in low-duty-cycle wireless sensor networks, in: INFOCOM, 2012 Proceedings IEEE, 2012, pp. 361–369.
[3] H. Cheng, Z. Su, N. Xiong, Y. Xiao, Energy-efficient node scheduling algorithms for wireless sensor networks using Markov Random field model, Inf. Sci. 329 (2016) 461–477.
[4] X. Chen, M. Ma, A. Liu, Dynamic power management and adaptive packet size selection for IoT in e-healthcare, Comput. Electric. Eng. (2017), doi:10.1016/j.compeleceng.2017.06.010.
[5] X. Chen, Y. Xu, A. Liu, Cross layer design for optimal delay, energy efficiency and lifetime in body sensor networks, Sensors 17 (4) (2017) 900, doi:10.3390/s17040900.
[6] R. Fonseca, O. Gnawali, K. Jamieson, et al. Four-Bit Wireless Link Estimation[C] //HotNets. 2007.
[7] P. Guo, N. Meratnia, P. Havinga, et al., OPS: opportunistic pipeline scheduling in long-strip wireless sensor networks with unreliable links, Wirel. Networks 21 (2015) 1669–1682.
[8] S. He, J. Chen, X. Li, et al., Mobility and intruder prior information improving the barrier coverage of sparse sensor networks, IEEE Trans. Mobile Comput. 13 (6) (2014) 1268–1282.
[9] S. He, D. Shin, J. Zhang, J. Chen, Y. Sun, Area coverage in camera sensor networks: dimension reduction and near-optimal solutions, IEEE Trans. Vehicular Technol. (2016), doi:10.1109/TVT.2015.2498281.
[10] M. Kariman-Khorasani, M.A. Pourmina, A. Salahi, Energy balance based lifetime maximization in wireless sensor networks employing joint routing and asynchronous duty cycle scheduling techniques, Wirel. Personal Commun. 83 (2) (2015) 1057–1083.
[11] L. Keller, E. Atsan, K. Argyraki, C. Fragouli, SenseCode: network coding for reliable sensor networks, ACM Trans, Sensor Networks 9 (2) (2013) 25:1-25:20.
[12] Z.I. Kiss, Z.A. Polgar, M.P. Stef, et al., Improving transmission reliability in wireless sensor networks using network coding, Telecommun. Syst. 59 (4) (2015) 509–521.
[13] J. Li, J. Li, D. Xie, et al., Secure auditing and deduplicating data in cloud, IEEE Trans. Comput. 65 (8) (2016) 2386–2396.
[14] H. Lin, L. Xu, Y. Mu, et al., A reliable recommendation and privacy-preserving based cross-layer reputation mechanism for mobile cloud computing, Future Gener. Comput. Syst. 52 (2015) 125–136.
[15] A. Liu, X Jin, G Cui, Z Chen, Deployment guidelines for achieving maximum lifetime and avoiding energy holes in sensor network, Inf. Sci. 230 (2013) 197–226.
[16] A. Liu, X. Liu, Z. Tang, L.T. Yang, Z. Shao, Preserving smart sink location privacy with delay guaranteed routing scheme for WSNs, ACM Trans. Embedded Comput. Syst. 16 (3) (2017) 68. doi:http://dx.doi.org/10.1145/2990500.

[17] A. Liu, X. Liu, T. Wei, L.T. Yang, S.C. Rho, A. Paul, Distributed Multi-representative *Re*-Fusion approach for Heterogeneous Sensing Data Collection, ACM Trans. Embedded Comput. Syst. 16 (3) (2017) 73. doi:http://dx.doi.org/10.1145/2974021.

[18] Q. Liu, A. Liu, On the hybrid using of unicast-broadcast in wireless sensor networks, Comput. Electric. Eng. (2017). Doi:http://dx.doi.org/10.1016/j.compeleceng.2017.03.004.

[19] X. Liu, G. Li, S. Zhang, A. Liu, Big program code dissemination scheme for emergency software-define wireless sensor networks, Peer-to-Peer Network. Appl. (2017), doi:10.1007/s12083-017-0565-5.

[20] X. Liu, A. Liu, Q. Deng, H. Liu, Large-scale programing code dissemination for software defined wireless networks, Comput. J. (2017), doi:10.1093/comjnl/bxx014.

[21] X. Liu, A. Liu, Z. Li, S. Tian, Y.J. Choi, H. Sekiya, J. Li, Distributed cooperative communication nodes control and optimization reliability for resource-constrained WSNs, Neurocomputing (2017). doi:https://doi.org/10.1016/j.neucom.2016.12.105.

[22] X. Liu, N. Xiong, W. Li, et al., An optimization scheme of adaptive dynamic energy consumption based on joint network-channel coding in wireless sensor networks, IEEE Sensors. J. 15 (9) (2015) 5158–5168.

[23] Y. Liu, A. Liu, Z. Chen, Analysis and improvement of send-and-wait automatic repeat-request protocols for wireless sensor networks, Wirel. Personal Commun. 81 (3) (2015) 923–959.

[24] Y. Liu, A. Liu, Y. Li, Z. Li, Y. Choi, H. Sekiya, J. Li, APMD: a fast data transmission protocol with reliability guarantee for pervasive sensing data communication, Pervasive Mobile Comput. (2017) 2017, doi:10.1016/j.pmcj.2017.03.012.

[25] Z. Liu, J. Feng, Y. Ji, EAF: energy-aware adaptive free viewpoint video wireless transmission, J. Network Comput. Appl. 46 (2014) 384–394.

[26] Z. Liu, T. Tsuda, H. Watanabe, Traffic deduction exploring sensor data's intra-correlations in train track monitoring WSN, Sensors, 2015 IEEE (2015) 1–4.

[27] J. Long, M. Dong, K. Ot, A. Liu, Green TDMA scheduling algorithm for prolonging lifetime in wireless sensor networks, IEEE Syst. J. 11 (2) (2017) 868–877.

[28] JK. Min, RT. Ng, K. Shim, Aggregate query processing in the presence of duplicates in wireless sensor networks, Inf. Sci. 297 (2015) 1–20.

[29] K.P. Naveen, A. Kumar, Relay selection for geographical forwarding in sleep-wake cycling wireless sensor networks, IEEE Trans. Mobile Comput. 12 (3) (2013) 475–488.

[30] K.P. Naveen, A. Kumar, Relay selection with channel probing in sleep-wake cycling wireless sensor networks, ACM Trans. Sensor Networks (TOSN) 11 (3) (2015) 52.

[31] OMNet++ Network Simulation Framework, http://www.omnetpp.org/.

[32] Z. Rosberg, R.P. Liu, T.L. Dinh, et al., Statistical reliability for energy efficient data transport in wireless sensor networks, Wirel. Networks 16 (7) (2010) 1913–1927.

[33] J. Wang, A. Liu, T. Yan, Z. Zeng, A resource allocation model based on double-sided combinational auctions for transparent computing, Peer-to-Peer Network. Appl. (2017), doi:10.1007/s12083-017-0556-6.

[34] J. Wang, A. Liu, S. Zhang, Key parameters decision for cloud computing: insights from a multiple game model, Concurr. Comput: Practice Exp. (2017), doi:10.1002/cpe.4200.

[35] M. Wu, L. Tan, N. Xiong, Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications, Inf. Sci. 329 (2016) 800–818.

[36] N. Xiong, X. Jia, L.T. Yang, A.V. Vasilakos, Y. Li, Y. Pan, A distributed efficient flow control scheme for multirate multicast networks, IEEE Trans. Parallel Distributed Syst. 21 (9) (2010) 1254–1266.

[37] N. Xiong, A.V. Vasilakos, L.T. Yang, L. Song, Y. Pan, R. Kannan, Y. Li, Comparative analysis of quality of service and memory usage for adaptive failure detectors in healthcare systems, IEEE J. Selected Areas Commun. 27 (4) (2009) 495–509.

[38] Q. Xu, Z. Su, S. Guo, A game theoretical incentive scheme for relay selection services in mobile social networks, IEEE Trans. Vehicular Technol. 65 (8) (2016) 6692–6702.

[39] Y. Xu, X. Chen, A. Liu, C. Hu, A latency and coverage optimized data collection scheme for smart cities based on vehicular ad-hoc networks, Sensors 17 (4) (2017) 888, doi:10.3390/s17040888.

[40] R. Zhang, O. Berder, J.M. Gorce, et al., Energy–delay tradeoff in wireless multihop networks with unreliable links, Ad Hoc Networks 10 (7) (2012) 1306–1321.

[41] Y. Zhang, X. Chen, J. Li, et al., Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing, Inf. Sci. 379 (2017) 42–61.

[42] S. Zhao, A. Liu, High performance target tracking scheme with low prediction precision requirement in WSNs, Int. J. Ad Hoc Ubiquitous Comput. (2017). http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijahuc.