FRCA: A Novel Flexible Routing Computing Approach for Wireless Sensor Networks

Ping Liu, Xingfu Wang, *Member, IEEE*, Ammar Hawbani, Omar Busaileh, Liang Zhao, *Member, IEEE*, Ahmed Al-Dubai, *Senior Member, IEEE*

Abstract— In wireless sensor networks, routing protocols with immutable network policies lacking the flexibility are generally incapable of maintaining effective performance due to the complicated and rapidly changing environment situations and application requirements. The proposed "Flexible Routing Computing Approach (FRCA)" is a novel distributed and probabilistic computing approach capable of modifying or upgrading routing policies on the fly with low cost, which effectively enhances the routing flexibility. FRCA models the routing metric as a forwarding probability distribution for routing decisions. This model depends on three elements, the physical quantities collected at sensor nodes, the built-in base math functions, and the routing parameters. These elements are all user-oriented and can be specified to implement multifarious complicated network policies meeting different performance requirements. More significantly, through distributing routing parameters from the sink to end nodes, operators are allowed to adjust network policies on the fly without interrupting the network services. Through extensive performance evaluation studies and simulations, the results demonstrate that routing protocols designed based on FRCA could achieve better performance compared to its state-of-the-art counterparts regarding network lifetime, energy consumption, and duplicate packets as well as ensure high flexibility during network policies modification or upgrade.

Index Terms—wireless sensor networks, probabilistic routing, distributed routing.

1 Introduction

THE flexibility of routing protocols is an essential factor to maintain or to enhance the performance of Wireless Sensor Networks (WSNs) which are susceptible to the environment conditions [1]. Allowing operators to modify or upgrade network policies on the fly, highly flexible routing protocols are capable of keeping up with the changeable performance requirements. In contrast, current network lacks flexibility where routing protocols are incompetent to adjust network policies without nodes' redeployment. Whenever operators tend to modify or upgrade the network policies, sensor nodes have to be recycled and reprogrammed. Flexibility is important not only because it could be costly and inefficient to redeploy WSNs to achieve desired performance requirements, but also because sometimes it may be impossible to access nodes which are deployed in dangerous environments, such as battlefields, active volcanoes, and deep oceans.

It is common that the performance requirements change dynamically with the network situations. For instance, the performance requirements of a WSN deployed in forests to prevent fires change remarkably with seasons. In rainy seasons, the fire risk of forests in a humid environment is relatively low due to the plentiful rain and routing policies should be more energy efficient to prolong the network lifetime. Conversely, in dry seasons, the forest is vulnerable to fire due to the dry and combustible fallen leaves. In order to detect fire as soon as possible, routing policies should ensure high transmission reliability and low latency. Furthermore, the application requirement itself is also not always static during the lifetime of WSNs. For example, WSNs deployed at battlefields are generally expected to work longer in peacetime, while in wartime the routing efficiency of networks needs to be paid more attention. Such application requirement changes sometimes are sudden and unpredictable, which cannot be settled through preprogramming.

Unfortunately, there are few routing protocols with high flexibility that allow modifying or upgrading network policies [1], [2]. Some conventional distributed routing algorithms allow the network operators to adjust the network performance by specifying some parameters provided in algorithms. This adjustment is still under the algorithm framework, which is difficult to achieve the desired adjustment goals. The advent of the *Software Defined Network* (SDN) provides a practical solution for solving this problem [2]. SDN uses open-flow and centralized routing algorithms to fully separate the data plane and control plane. Each routing device in SDN has a flow table indicating where the packet should be forwarded. The flow tables are computed and distributed by the controller. When operators want to modify or upgrade the network policy, they need to implement it in the controller, and the controller will reconfigure all the routing devices automatically.

- *Ping L. (iacmy@mail.ustc.edu.cn), Xingfu W., A. H. {wangxfu, anmande} @ustc.edu.cn, and Omar B. (busaileh@mail.ustc.edu.cn) are with School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China;*
- *Liang Z. (lzhao@sau.edu.cn); Shenyang aerospace University; Shenyang, 110136, China;*
- *A.Al-Dubai is with School of Computing; Edinburgh Napier University;(a.al-dubai@napier.ac.uk);*

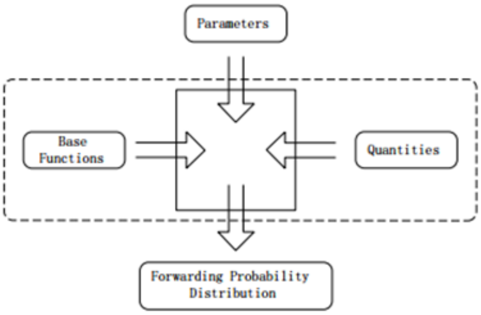


Fig. 1. The core elements of FRCA. The elements in the dotted box, base functions and quantities, need to be specified at sensor nodes before nodes' deployment while the routing parameters can be changed on the fly by operators at any time. Through modifying the routing parameters, operators can adjust the forwarding probability distribution to meet the new network performance requirements.

However, SDN which is proposed for wired networks is unsuitable for WSNs due to four reasons. First, excessive flow tables have to be calculated and distributed by the controller in WSNs where each sensor node works both as a sensor and a switch. Second, frequent flow requests from sensor nodes to the controller incur expensive communication cost and unacceptable latencies due to the unreliable link quality in wireless medium. Third, flows using to manage network traffic are inappropriate for WSNs since sensory data generated by nodes will all be sent to the sink. Last, routing paths of packets predefined by the controller using centralized algorithms are not suitable for WSNs with duty-cycled mechanism. In other words, a node in duty-cycled WSNs may fail to instantly receive the packet sent to it by other nodes due to its uncertain radio state. Therefore, SDN is not suitable for WSNs.

Motivated by the above observations and in an attempt to improve the flexibility of routing protocols efficiently in WSNs, we propose, in this paper, a Flexible Routing Computing Approach (FRCA). This work has three main contributions as follows:

1) We model the routing metric as a forwarding probability distribution which depends on three main elements, the physical quantities, the built-in base math functions, and the corresponding routing parameters, as shown in Fig. 1. FRCA not only provides flexibility for network configuration but also for policy designs. All the three elements are user-oriented and could be specified by operators to flexibly implement varieties of complicated routing policies meeting different environment situations and application requirements. Moreover, through distributing new routing parameters from the sink to end nodes, routing protocols based on FRCA are capable of modifying or upgrading routing policies on the fly without interrupting the network services after the deployment of nodes.

2) We design a routing metric for Opportunistic Routing based on FRCA which takes into account four quantities: direction, transmission distance, perpendicular distance, and residual energy. Compared with the well-known opportunistic routing protocols, ORW and ORR, the routing protocol using our metric achieves better performance in terms of network lifetime, energy consumption, and duplicate packets.

3) We recalculate the probability of multiple receivers for Opportunistic Routing, which is critical to analyze the network performance. The rest of the paper is organized as follows. Section 2 explains the related works. Section 3 introduces the preliminary knowledge of this paper. Section 4 illustrates the main idea of FRCA in detail and gives an example. The performance of FRCA is analyzed in Section 5. The simulation and evaluations are explained in section 6. Finally, section 7 concludes the paper.

2 Related Work

In WSNs, numerous routing protocols have been proposed to satisfy different application scenarios and performance requirements. These routing protocols can be classified in many categories according to different criteria. For example, these protocols can be categorized into flat or hierarchical protocols by network structure, coherent or non-coherent-based protocols by communication model, location-based or agent-based protocols by topology [3]. In order to facilitate the study of the flexibility, we categorize routing protocols based on routing computing manner to two groups, distributed and centralized protocols.

In WSNs, distributed routing protocols are more common than centralized protocols due to the distributed nature of WSNs. For distributed routing algorithms, according to the local information, routing decisions are made in sensor nodes which are scattered in the network. Distributed routing algorithms can also be divided into fixed and adaptive schemes. Fixed schemes use a constant forwarding probability to make routing decisions. The authors of [4] propose a stochastic routing framework based on the link reliability which actually can be considered as a constant for a given session of transition probability selection. While in adaptive schemes, different quantities such as transmission distance, direction, traffic load and residual energy of sensor nodes are considered to determine the forwarder. *Probabilistic Geographic Routing* (PGR) proposed in [5] assigns probabilities to each candidate node using residual energy and backward link reliability. *Distributed Heuristic Algorithm* (DHA) [6] assigns forwarding probabilities to the sender's neighbor nodes according to the distance between the sender and the sink, the maximum distance between the sender and each of its neighbor nodes as well as the distance between each neighbor node and the sink. The probability assignment scheme in DHA changes as packets get closer to the sink while in PGR the probability assignment is uniform along the routing path. Both these two protocols do not provide parameters to allow network operators to change the network performance to meet different requirements. *Zone Probabilistic Routing* (ZPR) [7] priorities the candidates with four distributions including direction, transmission distance, perpendicular distance, and residual energy. These four distributions can be controlled via a set of exponential control-parameters, which means ZPR can meet different performance for different applications.

For centralized routing algorithms, the global information is all collected at one place where all routing decisions are determined. For conventional centralized routing algorithms, decisions are made in the sink. ORW proposed in [8] for *Opportunistic Routing* (OR) [9] works on top of an asynchronous MAC protocol. ORW uses Expected Duty-Cycled Wakeups (EDC) to prioritize the candidates. ORR [10] improves the EDC by considering the residual energy of the candidates and achieves longer network lifetime. ORR provides a system parameter to control the weight on the residual energy. Recently, *Software Defined Network* (SDN) [11] which were first proposed for wired networks has been introduced into WSNs to improve the network management and configuration [12]. SDN simplifies the network management and configuration through fully separating the data plane and the control plane.

Furthermore, SDN uses a logically centralized software program to control the behavior of the entire network, which lets switches in the data plane become simple packet forwarding devices [13]. The basic requirements like supporting duty cycles, in network data aggregation, and flexible rules are discussed in [14]. Software-Defined WSN (SD-WSN) was proposed in [15] to meet these base requirements. SD-WSN uses Sensor OpenFlow (SOF) as the communication protocol between the two planes. Based on [15], SDN-WISE [16] makes sensor nodes programmable as finite state machines. Network operators are allowed to use any programming language they prefer to implement the SDN controller through the APIs offered by SDN-WISE. Unlike conventional centralized protocols, SDN brings the network management flexibility to WSNs. Network operators redefine the network behavior conveniently just by redefining the policies in the controller.

The flexibility of the above routing schemes has considerable differences. In distributed routing protocols, fixed schemes and adaptive schemes without control parameters cannot adjust the performance to meet different requirements. Adaptive schemes with control parameters can change network policies in a limited way. Such changes are still under the algorithm framework and may not be desired. In the centralized routing protocols, SDN is more convenient to manage and configure the WSNs than the conventional centralized protocols. However, SDN may be too costly for WSNs due to the challenges of expanding SDN to WSNs. These challenges could include complicated routing computation, duty cycles, high communication cost and different packets delivery mode (data-centric and address-centric), as explained in the introduction.

Unlike the aforementioned distributed routing schemes, our proposed (FRCA) approach in this article is highly flexible as a result of the consisting elements, quantities, base math function, and corresponding parameters are all highly customizable. It is convenient for the network operators to implement desired network policies flexibly. Meanwhile, through modifying the corresponding parameters, the network behavior is allowed to be redefined to meet different performances on the fly without nodes' redeployment. The approach fits into the WSNs well. Each sensor node makes routing decisions by itself and does not need to send routing requests to the controller frequently which will tremendously reduce control packets in the network. Besides, there is no doubt that the approach applies to WSNs with duty cycles.

3 PRELIMINARIES

In this section, three issues are explained. First, the assumptions of our work are illustrated in detail and the notations used in this paper are listed in Table. 1. Second, *Opportunistic Routing* (OR) used to verify the validity of FRCA is briefly introduced. Last, to facilitate the illustration of FRCA, the concept of base math functions are explained in detail.

3.1 Assumptions

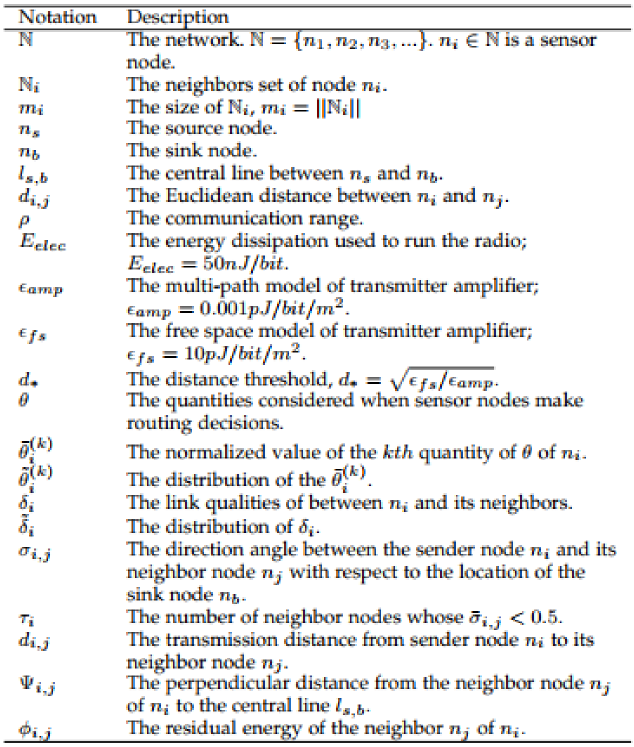
We assume that sensor nodes are all deployed at a specific region in a random way and define the network which contains all these nodes as $\mathbb{N} = \{n_1, n_2, n_{3,\dots}\}$. Any sensor node $n_i \in \mathbb{N}$ is static and knows its own geographical coordinate (x_i, y_i). All sensor nodes in $\mathbb{N}$ have the same limited communication range ρ. $d_{i,j}$ is the *Euclidean distance* from a node n_i to another node n_j. We define the neighbors set of n_i as $\mathbb{N}_i$ such that $\mathbb{N}_i = \{n_j | n_j \in \mathbb{N} \& d_{i,j} < \rho\}$. The size of neighbors set is denoted by $m_i = ||\mathbb{N}_i||$. The initial energy of n_i is also limited, denoted by e_*. We utilize the *First Order Radio Model* to calculate the energy consumption of n_i. The *Free Space Propagation Model* is utilized in this article which assumes the ideal propagation condition by representing the communication range as a circle around the transmitter.

In order to prolong the lifetime, a duty-cycle mechanism is applied into sensor nodes which helps reduce energy consumption by putting nodes into sleep mode periodically [18]. The MAC protocol used in this article is BoX-MAC [19], the default MAC protocol in TinyOS [20]. BoX-MAC is an asynchronous MAC protocol. In BoX-MAC, the sender transmits its packet repeatedly to the intended forwarder. The intended forwarder wakes up periodically to check whether there is any packet on the channel being transmitted to it. If the channel is idle, the forwarder will switch its state into sleep again. In contrast, when there is a packet on the channel, the forwarder will receive the packet and send back an ACK to the sender. Furthermore, the sender will stop transmitting the packet when its timer expires in case that the forwarder does not wake up during the transmitting process. Our approach is also suitable for WSNs in which nodes are time-synchronized.

3.2 Opportunistic Routing

Routing schemes for wireless sensor networks can be divided into two main groups, unipath routing, and anypath routing. In unipath routing, packets are delivered from the source to the sink through a single and predefined path, which is not suitable for wireless sensor networks due to the frequent transmission failure. In contrast, anypath routing enables packets to travel on a different route, which is beneficial for distributing the network's

TABLE 1
NOTATIONS



Notation	Description
$\mathbb{N}$	The network. $\mathbb{N} = \{n_1, n_2, n_3, ...\}$. $n_i \in \mathbb{N}$ is a sensor node.
$\mathbb{N}_i$	The neighbors set of node n_i.
m_i	The size of $\mathbb{N}_i$, $m_i = \|\|\mathbb{N}_i\|\|$
n_s	The source node.
n_b	The sink node.
$l_{s,b}$	The central line between n_s and n_b.
$d_{i,j}$	The Euclidean distance between n_i and n_j.
ρ	The communication range.
E_{elec}	The energy dissipation used to run the radio; $E_{elec} = 50nJ/bit$.
ϵ_{amp}	The multi-path model of transmitter amplifier; $\epsilon_{amp} = 0.001pJ/bit/m^2$.
ϵ_{fs}	The free space model of transmitter amplifier; $\epsilon_{fs} = 10pJ/bit/m^2$.
d_*	The distance threshold, $d_* = \sqrt{\epsilon_{fs}/\epsilon_{amp}}$.
θ	The quantities considered when sensor nodes make routing decisions.
$\bar{\theta}_i^{(k)}$	The normalized value of the kth quantity of θ of n_i.
$\tilde{\theta}_i^{(k)}$	The distribution of the $\bar{\theta}_i^{(k)}$.
δ_i	The link qualities of between n_i and its neighbors.
$\tilde{\delta}_i$	The distribution of δ_i.
$\sigma_{i,j}$	The direction angle between the sender node n_i and its neighbor node n_j with respect to the location of the sink node n_b.
τ_i	The number of neighbor nodes whose $\bar{\sigma}_{i,j} < 0.5$.
$d_{i,j}$	The transmission distance from sender node n_i to its neighbor node n_j.
$\Psi_{i,j}$	The perpendicular distance from the neighbor node n_j of n_i to the central line $l_{s,b}$.
$\phi_{i,j}$	The residual energy of the neighbor n_j of n_i.

load among nodes. Conventional unipath and anypath routing do not consider the usefulness of the broadcast characteristic of the wireless medium and the duty-cycle mechanism. *Opportunistic Routing* (OR) is a new anypath routing scheme which can effectively improve the performance of duty-cycled WSNs by taking advantages of the benefits of overhearing wireless signals [9]. FRCA can be applied into all of these routing schemes. In this article, we will apply FRCA into OR, the most complicated routing scheme. To facilitate the subsequent illustration, OR will be briefly introduced in this subsection.

In *OR*, the sender selects a group of nodes from its neighboring nodes as candidate forwarders. This group of nodes is called *Candidate Set* (CS). In general, when data transmission starts, the sender broadcasts its packet first, the candidates in CS which have received the packet coordinate among each other to determine the final forwarder. This final forwarder will continue the forwarding process while other candidates discard the packet that they have received. Therefore, the performance of *Opportunistic Routing* protocols depends on three main elements: *Routing Metric*, *Candidates Set Selection*, and *Candidates Coordination* [21].

The formulation of candidate set depends on the *Routing Metric* and the *Candidates Set Selection*. The *Routing Metric* is a mathematical model to measure the ability of the neighbor nodes to reach the sink node with lower forwarding cost. FRCA models the routing metric as a forwarding probability distribution where the metric value $\tilde{\delta}_{i,j}$ between the neighbor node n_j and the sender n_i is a probability value such that $\sum_j^{||\mathbb{N}_i||} \tilde{\delta}_{i,j} = 1$. Higher probability implies that the node can reach the destination with lower cost. The procedure for designing a routing metric based on FRCA will be illustrated in Section 4. *Candidates Set Selection* is the mechanism that allows the sender to sort and select its candidate forwarders using the routing metric. In this article, we assume only the neighbor node n_j whose metric value $\tilde{\delta}_{i,j}$ meets $\tilde{\delta}_{i,j} > 1/m_i$, where $m_i = ||\mathbb{N}_i||$, will be added into the candidate set of n_i. *Candidates Coordination* is the mechanism that ensures only one node in the candidate set can finally forward the packet to the next hop. In this article, in order to save transmission energy in the forwarding process, our coordination among candidates is Acknowledgment-based as in ORW. The sender will broadcast a beacon packet instead of the data packet to candidates first. The awaken candidates will receive the beacon and send back an ACK. According to the ACK received, the sender can identify which of the candidates are in active state and forwards the data packet to the active candidate with highest forwarding probability. This final forwarder will continue the forwarding process.

3.3 Base Math Functions

In FRCA, we prepare a base math function pool at each sensor node which contains several base math functions specified by network operators, such as Gauss and Boltzmann function. Each base function in the base function pool has a unique number. Network operators can add any math function they want to use into the pool and select the appropriate math function to calculate the priority value for candidates by specifying the function number. With this base math function pool, operators are allowed to design varieties of complicated forwarding probability distribution to meet different requirements and flexibly modify the distribution on the fly without interruption of services. The procedure for using these base math functions to design the final forwarding probability distribution will be illustrated in Section 4. We introduce three base math functions here, the exponential function, the Gauss function, and the Boltzmann function.

1) The exponential function expressed in (1) has three parameters, y_e, A_e and t_e. Through specifying these three parameters, the exponential function can depict four types of curves as shown in Fig. 2 (a) and (b).

$$y = y_e + A_e \cdot e^{-x/t_e} \qquad (1)$$

2) The Gauss function expressed in (2) has four parameters, y_g, A_g, x_g, and w_g. The type of curve produced is like Fig. 2 (c).

$$y = y_g + \frac{A_g}{w_g \cdot \sqrt{\pi/2}} \cdot e^{-\frac{2(x-x_g)^2}{w_g^2}} \qquad (2)$$

3) If the desired function curve is like a sigmoid line as shown in Fig. 2 (d), then the Boltzmann function expressed in (3) will be the best math function. It has four parameters,

A_{b1}, A_{b2}, x_b, and dx.

$$y = \frac{A_{b1} - A_{b2}}{1 + e^{(x - x_b)/dx}} + A_{b2} \qquad (3)$$

4 THE PROPOSED APPROACH

In this section, we will illustrate FRCA in detail. The implements of desired distributions for quantities concerning routing decisions and the flexible mechanism for combining these distributions to calculate the final forwarding probability distribution will be detailedly explained. Furthermore, an instance applying FRCA into opportunistic routing protocols is given, in which we design a novel forwarding probability distribution for the routing metric of opportunistic routing based on FRCA. The quantities that we selected, the distributions designed for these quantities, and the final forwarding probability distribution will all be demonstrated in subsection 4.2. Other two main components of the opportunistic routing protocol, *Candidates Set Selection,* and *Candidates Coordination*, are explained and set in Section 3.

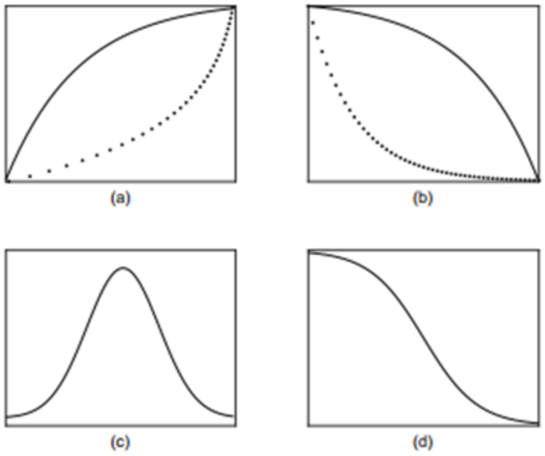


Fig. 2. Different types of function curves.

4.1 A Flexible Routing Computing Approach (FRCA)

All routing decisions made by sensor nodes not only depend on the algorithm of routing protocols, but also on the given physical quantities. There are some physical quantities which are commonly used in routing protocols such as transmission distance, direction, and channel fading. Some unusual quantities could also be used in some application scenarios, like the speed of cars in a vehicular ad-hoc network, depth of the communicating nodes in underwater wireless sensor networks, and temperature of the skin in wireless body area networks. We denote the quantities used in the routing algorithm as $\theta = (\theta^{(1)}, \theta^{(2)}, \theta^{(3)}, \ldots, \theta^{(p)})$, where p represents the number of quantities considered. For a given sender node n_i, its quantities are expressed by a vector $\theta_i = (\theta_i^{(1)}, \theta_i^{(2)}, \theta_i^{(3)}, \ldots, \theta_i^{(p)})$. For each $\theta^{(k)}$, where $1 \le k \le p$, the quantity values between n_i and each of its neighbor nodes would be collected to a random variable $\theta_i^{(k)} = (\theta_{i,1}^{(k)}, \theta_{i,2}^{(k)}, \theta_{i,3}^{(k)}, \ldots, \theta_{i,m_i}^{(k)})$, where $m_i = ||\mathbb{N}_i||$. In order to facilitate subsequent calculations, $\theta_{i,j}^{(k)}$ is then normalized to $\overline{\theta}_{i,j}^{(k)}$ using (4) such that $0 \le \overline{\theta}_{i,j}^{(k)} \le 1$, where γ_k is the corresponding normalization function of $\theta^{(k)}$. Consequently, the random variable $\theta_i^{(k)}$ is normalized to $\overline{\theta}_i^{(k)} = (\overline{\theta}_{i,1}^{(k)}, \overline{\theta}_{i,2}^{(k)}, \overline{\theta}_{i,3}^{(k)}, \ldots, \overline{\theta}_{i,m_i}^{(k)})$.

$$\overline{\theta}_{i,j}^{(k)} = \gamma_k(\theta_{i,j}^{(k)}) \qquad (4)$$

So far, we have prepared the physical quantities to be used in routing computing. In the rest of this subsection, the distribution of each quantity will be calculated. Then these distributions will be combined to calculate the final forwarding probability distribution.

The distribution of n_i for the quantity $\theta^{(k)}$, denoted by $\tilde{\theta}_i^{(k)} = (\tilde{\theta}_{i,1}^{(k)}, \tilde{\theta}_{i,2}^{(k)}, \tilde{\theta}_{i,3}^{(k)}, \ldots, \tilde{\theta}_{i,m_i}^{(k)})$, specifies the priority for its neighbor nodes to be forwarders in terms of this quantity, which is formulated by the mass function (5).

$$\tilde{\theta}_{i,j}^{(k)} = \frac{\zeta(\overline{\theta}_{i,j}^{(k)}, \xi^{(k)}, \varphi^{(k)})}{\Sigma_v^{||\mathbb{N}_i||} \zeta(\overline{\theta}_{i,v}^{(k)}, \xi^{(k)}, \varphi^{(k)})} \qquad (5)$$

$\zeta(\overline{\theta}_{i,j}^{(k)}, \xi^{(k)}, \varphi^{(k)})$ formulated by (6) is a *priority value function* which measures the priority of the neighbor node n_j to act as the next forwarder by considering the normalized quantity value $\overline{\theta}_{i,j}^{(k)}$. $\varphi^{(k)}$ represents the number of the base math function selected to depict the priority value curve of the quantity value $\overline{\theta}_{i,j}^{(k)}$, as explained in Section 3. The $\xi^{(k)} = (\xi_e^{(k)}, \xi_g^{(k)}, \xi_b^{(k)})$ is the coefficient vector which contains the corresponding parameters of the selected base math function. Through specifying the function number $\varphi^{(k)}$ and the corresponding parameters $\xi^{(k)}$ of a given quantity, network operators can fully control its priority value function.

$$\zeta(\overline{\theta}_{i,j}^{(k)}, \xi^{(k)}, \varphi^{(k)})$$

$$= \begin{cases} \xi_{e,0}^{(k)} + \xi_{e,1}^{(k)} \cdot e^{-\frac{\overline{\theta}_{i,j}^{(k)}}{\xi_{e,2}^{(k)}}} & \varphi^{(k)} = 0 \\ \xi_{g,0}^{(k)} + \frac{\xi_{g,3}^{(k)}}{\xi_{g,2}^{(k)} \cdot \sqrt{\frac{\pi}{2}}} \cdot e^{-\frac{2[\overline{\theta}_{i,j}^{(k)} - \xi_{g,1}^{(k)}]^2}{(\xi_{g,2}^{(k)})^2}} & \varphi^{(k)} = 1 \\ \frac{\xi_{b,0}^{(k)} - \xi_{b,1}^{(k)}}{1 + e^{(\overline{\theta}_{i,j}^{(k)} - \xi_{b,2}^{(k)})/\xi_{b,3}^{(k)}}} + \xi_{b,1}^{(k)} & \varphi^{(k)} = 2 \end{cases} \qquad (6)$$

For instance, Fig. 3 depicts two potential priority value function curves of different physical quantities, signal strength and node loads. From Fig.3, we can see that the changed tendencies of priority values for these two quantities are opposite.

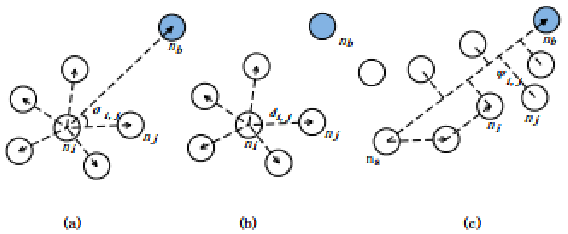


Fig. 4. n_b is the sink node, n_s is the source node, n_i is the current sender node and n_j is one of the neighbor nodes of n_i. The coordinates of these nodes are (x_b, y_b),(x_s, y_s),(x_i, y_i) and (x_j, y_j). (a) shows the direction angel $\sigma_{i,j}$ from n_i to n_j with respect to the location of n_b. (b) shows the transmission distance from n_i to n_j. (c) shows the perpendicular distances from the neighbor nodes of the current sender node to the central line $l_{s,b}$ between the source node n_s to the sink node n_b.

Through modifying the customizable parameters $\varphi^{(k)}$ and $\xi^{(k)}$, the changed tendency of priority value as a certain quantity value increases can be adjusting. We define the base math function number specified by operators for all quantities considered in routing computing as a random variable $\xi = (\xi^{(1)}, \xi^{(2)}, \xi^{(3)}, \ldots, \xi^{(p)})$, and the corresponding parameters as $\varphi = (\varphi^{(1)}, \varphi^{(2)}, \varphi^{(3)}, \ldots, \varphi^{(p)})$. The distributions of quantities considered at the sender node n_i can be expressed as a random variable $\tilde{\theta_i} = (\tilde{\theta}_i^{(1)}, \tilde{\theta}_i^{(2)}, \tilde{\theta}_i^{(3)}, \ldots, \tilde{\theta}_i^{(p)})$.

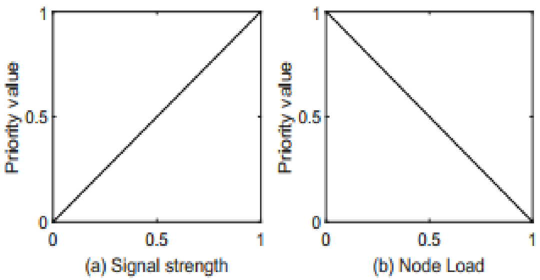


Fig. 3. The priority value functions of the normalized signal strength and node load.

The distributions attained can be combined using (7) to calculate the final forwarding priority value $\delta_{i,j}$ between the sender node n_i and its neighbor node n_j. R represents the item number of the equation. $w = (w_1, w_2, w_3, \ldots, w_R)$ are coefficients of items, which represent the weights of items. $\mu = (\mu^{(1)}, \mu^{(2)}, \mu^{(3)}, \ldots, \mu^{(R)})$ is a vector containing the corresponding exponents for each item. Through adjusting these three parameters, R, w and μ, network operators are able to conveniently implement a complicated distribution function in a flexible way.

$$\delta_{i,j} = \sum_{r}^{R} w_r \cdot \prod_{k}^{p} [\, \tilde{\theta}_{i,j}^{(k)}]^{\mu_k^{(r)}} \qquad (7)$$

Lastly, the final forwarding probability distribution of the sender node n_i is formulated by (8), which can be expressed as a random variable $\tilde{\delta}_i = (\tilde{\delta}_{i,1}, \tilde{\delta}_{i,2}, \tilde{\delta}_{i,3}, \ldots, \tilde{\delta}_{i,m_i})$.

$$\tilde{\delta}_{i,j} = \frac{\delta_{i,j}}{\sum_{v}^{||\mathbb{N}_i||} \delta_{i,v}} \qquad (8)$$

The procedures for computing forwarding probability distribution is described in Algorithm 1.

Algorithm 1 Procedures for Computing Forwarding Probability Distribution $\tilde{\delta}_i$ at Node n_i.

```
Input:
    The quantity values collected at node $n_i$: $\theta_i$;
    The parameters set by network operators: $\xi, \varphi, w, R, \mu$;
Output:
Normalize $\theta_i$ to $\bar{\theta}_i$ using (4)
for each quantity $\bar{\theta}_i^{(k)}$ in $\bar{\theta}_i$ do
    for each neighbor node $n_j$ do
        Calculate priority value $\zeta_{i,j}$ of $n_j$ using (6)
    end for
    Calculate the distribution $\tilde{\theta}_i^{(k)}$ for $\bar{\theta}_i^{(k)}$ using (5)
end for
Combine all distributions $\tilde{\theta}_i$ to calculate the final priority
    value $\delta_{i,j}$ using (7)
Calculate the forwarding probability distribution $\tilde{\delta}_i$ using (8)
return $\tilde{\delta}_i$
```

4.2 A Novel Forwarding Probability Distribution for OR Metric

In this section, we design a novel forwarding probability distribution for OR metric based on FRCA. The main goal of this routing metric is to prolong the lifetime of the network while maintaining a relatively low waiting time for senders. The sender node uses this metric to prioritize its candidates. The distributions of quantities considered in this article are the direction distribution, the transmission distance distribution, the perpendicular distance distribution, and the residual energy distribution.

4.2.1 Direction Distribution

We use $\sigma_{i,j}$ to denote the direction angle as shown in Fig. 4 (a) between the sender node n_i and its neighbor node n_j with respect to the location of the sink node n_b. Using vectors $\vec{\alpha} = (x_j - x_i, y_j - y_i)$ and $\vec{\beta} = (x_b - x_i, y_b - y_i)$, $\sigma_{i,j}$ can be calculated according to (9). The angles between n_i and each of its neighbor nodes are expressed as a random variable $\sigma_i = (\sigma_{i,1}, \sigma_{i,2}, \sigma_{i,3}, \ldots, \sigma_{i,m_i})$, $m_i = ||\mathbb{N}_i||$. These angles need to be normalized to $\overline{\sigma_i} = (\overline{\sigma}_{i,1}, \overline{\sigma}_{i,2}, \overline{\sigma}_{i,3}, \ldots, \overline{\sigma}_{i,m_i})$ by (10) such that $0 \le \overline{\sigma}_{i,j} \le 1$.

$$\sigma_{i,j} = arccos \frac{\vec{\alpha} \cdot \vec{\beta}}{||\vec{\alpha}|| \cdot ||\vec{\beta}||} \qquad (9)$$

$$\bar{\sigma}_{i,j}=\frac{\sigma_{i,j}}{\pi} \qquad (10)$$

The physical quantity, direction angle, indicates the location relationship among the sender and its neighbor nodes. If $\bar{\sigma}_{i,j}$ is smaller than 0.5, then the neighbor node n_j is closer to the sink and can provide routing progress with lower cost than other neighbor nodes whose $\bar{\sigma}_{i,j}>0.5$. Therefore, the goal of the direction distribution is to give a higher priority for the neighbor node that has a smaller direction angle. The priorities for neighbor nodes whose direction angle are larger than 0.5 must be relatively low to avoid poor routing. We find that the Boltzmann function described is quite suitable for characterizing such property. Therefore, the priority value of direction $\tilde{\sigma}'_{i,j}$ is formulated by (11). The function curve of (11) is shown in Fig. 5 (a).

$$\begin{cases}\tilde{\sigma}'_{i,j}=\dfrac{A_{b1}-A_{b2}}{1+e^{(\tilde{\sigma}_{i,j}-x_b)/dx}}+A_{b2}\\ A_{b1}=1,A_{b2}=0\\ x_b=0.5,dx=0.05\end{cases} \qquad (11)$$

In Fig. 5 (a), if the normalized direction angle is larger than x_0 in (11), then the priority value of the corresponding neighbor node will be lower than 0.5. Thus, according to (11), only the neighbor nodes whose $\bar{\sigma}_{i,j}<0.5$ will have high priority values. However, in different network situation or even in different areas at the same network, the local densities of sensor nodes have great differences, as shown in Fig. 6.

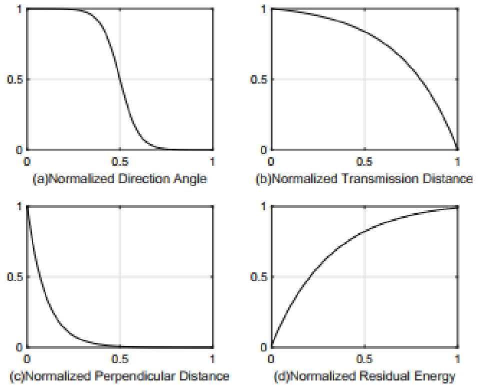


Fig. 5. The curves of priority value function used in this article.

Using the static $x_0=0.5$ in (11) is inappropriate. In Fig. 6 (a), there is a few neighbor nodes whose $\bar{\sigma}_{i,j}<x_0$ while in Fig. 6 (b) there are too many neighbor nodes available. Therefore we introduce the quantity τ_i which represents the number of neighbor nodes whose $\bar{\sigma}_{i,j}<0.5$ to modify x_0 in (11). We define x_0 after modification as λ_i which is formulated by (12). The function curve of (12) is shown in Fig. 7 (a). The values of parameters are shown in Table. 2.

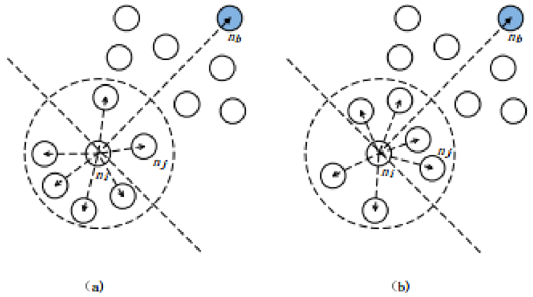


Fig. 6. Sender nodes which have a different number of neighbor nodes. The number of neighbor nodes whose $\bar{\sigma}_{i,j}<0.5$ of the sender node n_i in (a) is much less than that in (b).

$$\begin{cases}\lambda_i=y_g+\dfrac{A_g}{w_g\cdot\sqrt{\pi/2}}\cdot e^{-\frac{2(\tau_i-x_g)^2}{w_g^2}}\\ y_g=0.07053,A_g=8.11192\\ w_g=5.26696,x_g=-1.85373\end{cases} \qquad (12)$$

Therefore, (11) can be modified to (13).

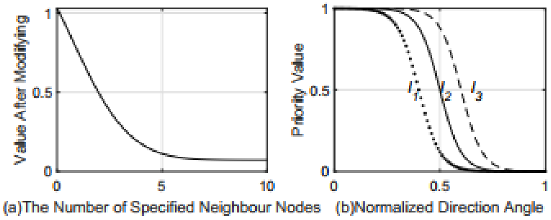


Fig. 7. (a) shows the function curve of (12). (b) shows function curves of the priority value function in (11) where x_0 has been modified to different values. l_2 is the curve when $x_0=0.5$ is assigned as in (11). The curves l_1 and l_3 depict the priority value function when x_0 is modified to $x_0-0.1$ and $x_0+0.1$.

$$\begin{cases}\tilde{\sigma}'_{i,j}=\dfrac{A_{b1}-A_{b2}}{1+e^{(}\tilde{\sigma}_{i,j}-x_b)/dx}+A_{b2}\\ A_{b1}=1,A_{b2}=0\\ x_b=\lambda_i,dx=0.05\end{cases} \qquad (13)$$

Based on the normalized direction angle $\bar{\sigma}_i$, and λ_i, the direction distribution $\tilde{\sigma}_i=(\tilde{\sigma}_{i,1},\tilde{\sigma}_{i,2},\tilde{\sigma}_{i,3},\ldots,\tilde{\sigma}_{i,m_i})$ is obtained by the mass function according to (5), as shown in (14).

$$\tilde{\sigma}_{i,j}=\frac{\zeta(\bar{\sigma}_{i,j},\xi^{(\sigma)},\varphi^{(\sigma)})}{\Sigma_v^{\|\mathbb{N}_i\|}\zeta(\bar{\sigma}_{i,v},\xi^{(\sigma)},\varphi^{(\sigma)})} \qquad (14)$$

The values of parameters in (14) are shown in Table. 2.

4.2.2 Transmission Distance Distribution

The transmission distance from the sender node n_i to its neighbor node n_j, denoted by $d_{i,j}$, is formulated by (15). As shown in Fig. 4 (b), $d_{i,j}$ is the *Euclidean distance* from n_i to n_j. Let $d_i = (d_{i,1}, d_{i,2}, d_{i,3}, \dots, d_{i,m_i})$ be a random variable which represents the transmission distance from the sender node to each of its neighbor nodes. These values are then normalized to $\overline{d}_i = (\overline{d}_{i,1}, \overline{d}_{i,2}, \overline{d}_{i,3}, \dots, \overline{d}_{i,m_i})$, $0 \le \overline{d}_{i,j} \le 1$, using (16), where ρ represents the communication radius. In order to save energy and prolong the lifetime of sensor nodes, high priority should be given to neighbor nodes with short transmission distance from the sender node. Hence, let $\tilde{d}'_{i,j}$ in (17) be the priority function value of $\overline{d}_{i,j}$. The function curve of (17) is shown in Fig. 5 (b). Finally, the transmission distance distribution $\tilde{d}_i = (\tilde{d}_{i,1}, \tilde{d}_{i,2}, \tilde{d}_{i,3}, \dots, \tilde{d}_{i,m_i})$ is formulated by the mass function (18).

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (15)$$

$$\overline{d}_{i,j} = \frac{d_{i,j}}{\rho} \quad (16)$$

$$\begin{cases} \tilde{d}'_{i,j} = y_e + A_e \cdot e^{-\overline{d}_{i,j}/t_e} \\ y_e = 1.03918, A_e = -0.03918 \\ t_e = -0.30506 \end{cases} \quad (17)$$

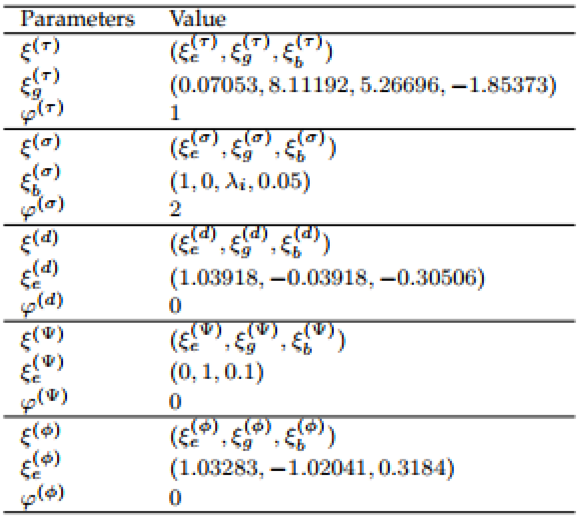
$$\tilde{d}_{i,j} = \frac{\zeta(\overline{d}_{i,j}, \xi^{(d)}, \varphi^{(d)})}{\Sigma_v^{||\mathbb{N}_i||} \zeta(\overline{d}_{i,v}, \xi^{(d)}, \varphi^{(d)})} \quad (18)$$

The values of parameters in (18) are shown in Table. 2.

4.2.3 Perpendicular Distance Distribution

Let the perpendicular distance from the neighbor node n_j to the central line $l_{s,b}$ be $\Psi_{i,j}$ which is formulated by (19), as shown in Fig. 4 (c). $\Psi_i = (\Psi_{i,1}, \Psi_{i,2}, \Psi_{i,3}, \dots, \Psi_{i,m_i})$ represents the perpendicular distance from each neighbor node of the sender node to the central line $l_{s,b}$ and is normalized to $\overline{\Psi}_i = (\overline{\Psi}_{i,1}, \overline{\Psi}_{i,2}, \overline{\Psi}_{i,3}, \dots, \overline{\Psi}_{i,m_i})$ using (20). Smaller perpendicular distance of candidates implies shorter distances between the routing path and the central line $l_{s,b}$ and shorter transmission distance. Thus, higher priorities should be assigned to neighbor nodes with small perpendicular distance. The priority value $\tilde{\Psi}'_{i,j}$ of $\overline{\Psi}_{i,j}$ is formulated by (21) and the function curve is shown in Fig. 5 (c). Furthermore, the perpendicular distance distribution $\tilde{\Psi}_i = (\tilde{\Psi}_{i,1}, \tilde{\Psi}_{i,2}, \tilde{\Psi}_{i,3}, \dots, \tilde{\Psi}_{i,m_i})$ is formulated by (22). The values of parameters are shown in Table. 2.

TABLE 2
PARAMETERS FOR PRIORITY VALUE FUNCTIONS



Parameters	Value
$\xi^{(\tau)}$	$(\xi_e^{(\tau)}, \xi_g^{(\tau)}, \xi_b^{(\tau)})$
$\xi_g^{(\tau)}$	$(0.07053, 8.11192, 5.26696, -1.85373)$
$\varphi^{(\tau)}$	1
$\xi^{(\sigma)}$	$(\xi_e^{(\sigma)}, \xi_g^{(\sigma)}, \xi_b^{(\sigma)})$
$\xi_b^{(\sigma)}$	$(1, 0, \lambda_i, 0.05)$
$\varphi^{(\sigma)}$	2
$\xi^{(d)}$	$(\xi_e^{(d)}, \xi_g^{(d)}, \xi_b^{(d)})$
$\xi_e^{(d)}$	$(1.03918, -0.03918, -0.30506)$
$\varphi^{(d)}$	0
$\xi^{(\Psi)}$	$(\xi_e^{(\Psi)}, \xi_g^{(\Psi)}, \xi_b^{(\Psi)})$
$\xi_e^{(\Psi)}$	$(0, 1, 0.1)$
$\varphi^{(\Psi)}$	0
$\xi^{(\phi)}$	$(\xi_e^{(\phi)}, \xi_g^{(\phi)}, \xi_b^{(\phi)})$
$\xi_e^{(\phi)}$	$(1.03283, -1.02041, 0.3184)$
$\varphi^{(\phi)}$	0

$$\begin{cases} \vec{v}_{sb} = (x_b - x_s, y_b - y_s) \\ \vec{v}_{sj} = (x_j - x_s, y_j - y_s) \\ \Psi_{i,j} = ||\vec{v}_{sj}|| \cdot sin[arccos \dfrac{\vec{v}_{sb} \cdot \vec{v}_{sj}}{||\vec{v}_{sb}|| \cdot ||\vec{v}_{sj}||}] \end{cases} \quad (19)$$

$$\overline{\Psi}_{i,j} = \frac{\Psi_{i,j}}{\rho + \Psi_{i,i}} \quad (20)$$

$$\begin{cases} \tilde{\Psi}'_{i,j} = y_e + A_e \cdot e^{-\overline{\Psi}_{i,j}/t_e} \\ y_e = 0, A_e = 1 \\ t_e = 0.1 \end{cases} \quad (21)$$

$$\tilde{\Psi}_{i,j} = \frac{\zeta(\overline{\Psi}_{i,j}, \xi^{(\Psi)}, \varphi^{(\Psi)})}{\Sigma_v^{||\mathbb{N}_i||} \zeta(\overline{\Psi}_{i,v}, \xi^{(\Psi)}, \varphi^{(\Psi)})} \quad (22)$$

4.2.4 Residual Energy Distribution

The residual energy of the node n_j which is the neighbor of n_i is denoted by $\phi_{i,j}$. $\phi_i = (\phi_{i,1}, \phi_{i,2}, \phi_{i,3}, \dots, \phi_{i,m_i})$ represents the residual energy of each neighbor node and is normalized to $\overline{\phi}_i = (\overline{\phi}_{i,1}, \overline{\phi}_{i,2}, \overline{\phi}_{i,3}, \dots, \overline{\phi}_{i,m_i})$ using (23), where e_* is the initial energy of n_j. The priority value $\tilde{\phi}'_{i,j}$ of $\overline{\phi}_{i,j}$ is formulated by (24) and the function curve is shown in Fig. 5 (d). The residual energy distribution $\tilde{\phi}_i = (\tilde{\phi}_{i,1}, \tilde{\phi}_{i,2}, \tilde{\phi}_{i,3}, \dots, \tilde{\phi}_{i,m_i})$ is formulated by (25). The values of parameters are shown in Table. 2.

$$\overline{\phi}_{i,j} = \frac{\phi_{i,j}}{e_*} \quad (23)$$

$$\begin{cases} \tilde{\phi}'_{i,j} = y_e + A_e \cdot e^{-\overline{\phi}_{i,j}/t_e} \\ y_e = 1.03283, A_e = -1.02041 \\ t_e = 0.3184 \end{cases} \quad (24)$$

$$\tilde{\phi}_{i,j} = \frac{\zeta(\overline{\phi}_{i,j}, \xi^{(\phi)}, \varphi^{(\phi)})}{\Sigma_v^{||\mathbb{N}_i||} \zeta(\overline{\phi}_{i,v}, \xi^{(\phi)}, \varphi^{(\phi)})} \quad (25)$$

4.2.5 Final Forwarding Probability Distribution

Based on the predefined four quantities which have distributions $(\tilde{\sigma}_i, \tilde{d}_i, \tilde{\Psi}_i, \tilde{\phi}_i)$, the final forwarding probability distribution, denoted by $\delta_i = (\delta_{i,1}, \delta_{i,2}, \delta_{i,3}, \ldots, \delta_{i,m_i})$, between the sender n_i and each of its neighbor nodes is formulated by (26). According to (26), when the neighbor node is not on the correct direction, the corresponding forwarding probability will be pretty low. Using FRCA to implement (26), the parameters in (7), R, $w = (w_1, w_2, w_3, \ldots, w_R)$ and $\mu = (\mu^{(1)}, \mu^{(2)}, \mu^{(3)}, \ldots, \mu^{(R)})$ need to be assigned according to Table. 3, where $\mu^{(i)} = (\mu_\sigma^{(i)}, \mu_d^{(i)}, \mu_\Psi^{(i)}, \mu_\phi^{(i)})$, $1 \le i \le R$. The calculation process is derived in (27). Finally, δ_i is normalized to $\tilde{\delta}_i = (\tilde{\delta}_{i,1}, \tilde{\delta}_{i,2}, \tilde{\delta}_{i,3}, \ldots, \tilde{\delta}_{i,m_i})$ using (28).

$$\delta_{i,j} = 0.25 \cdot \tilde{\sigma}_{i,j} \cdot \tilde{d}_{i,j} + 0.25 \cdot \tilde{\sigma}_{i,j} \cdot \tilde{\Psi}_{i,j} + 0.5 \cdot \tilde{\sigma}_{i,j} \cdot \tilde{\phi}_{i,j} \quad (26)$$

TABLE 3
PARAMETERS FOR (27)

Parameters	Value
R	3
w	$(0.25, 0.25, 0.25)$
$\mu^{(1)}$	$(1, 1, 0, 0)$
$\mu^{(2)}$	$(1, 0, 1, 0)$
$\mu^{(3)}$	$(1, 0, 0, 1)$

$$\begin{aligned}\delta_{i,j} &= \sum_r^R w_r \cdot \prod_k^p [\,\tilde{\theta}_{i,j}^{(k)}]^{\mu_k^{(r)}} \\ &= w_1 \cdot (\tilde{\sigma}_{i,j})^{\mu_\sigma^{(1)}} \cdot (\tilde{d}_{i,j})^{\mu_d^{(1)}} \cdot (\tilde{\Psi}_{i,j})^{\mu_\Psi^{(1)}} \cdot (\tilde{\phi}_{i,j})^{\mu_\phi^{(1)}} \\ &+ w_2 \cdot (\tilde{\sigma}_{i,j})^{\mu_\sigma^{(2)}} \cdot (\tilde{d}_{i,j})^{\mu_d^{(2)}} \cdot (\tilde{\Psi}_{i,j})^{\mu_\Psi^{(2)}} \cdot (\tilde{\phi}_{i,j})^{\mu_\phi^{(2)}} \\ &+ w_3 \cdot (\tilde{\sigma}_{i,j})^{\mu_\sigma^{(3)}} \cdot (\tilde{d}_{i,j})^{\mu_d^{(3)}} \cdot (\tilde{\Psi}_{i,j})^{\mu_\Psi^{(3)}} \cdot (\tilde{\phi}_{i,j})^{\mu_\phi^{(3)}} \\ &= 0.25 \cdot (\tilde{\sigma}_{i,j})^1 \cdot (\tilde{d}_{i,j})^1 \cdot (\tilde{\Psi}_{i,j})^0 \cdot (\tilde{\phi}_{i,j})^0 \\ &+ 0.25 \cdot (\tilde{\sigma}_{i,j})^1 \cdot (\tilde{d}_{i,j})^0 \cdot (\tilde{\Psi}_{i,j})^1 \cdot (\tilde{\phi}_{i,j})^0 \\ &+ 0.50 \cdot (\tilde{\sigma}_{i,j})^1 \cdot (\tilde{d}_{i,j})^0 \cdot (\tilde{\Psi}_{i,j})^0 \cdot (\tilde{\phi}_{i,j})^1 \\ &= 0.25 \cdot \tilde{\sigma}_{i,j} \cdot \tilde{d}_{i,j} + 0.25 \cdot \tilde{\sigma}_{i,j} \cdot \tilde{\Psi}_{i,j} + 0.5 \cdot \tilde{\sigma}_{i,j} \cdot \tilde{\phi}_{i,j}\end{aligned} \quad (27)$$

$$\tilde{\delta}_{i,j} = \frac{\delta_{i,j}}{\Sigma_v^{||\mathbb{N}_i||} \delta_{i,v}} \quad (28)$$

5 ANALYSIS

In this section, we analyze the performance of the proposed routing computing approach FRCA. We will mathematically express the energy cost, the number of redundant packets, and the average waiting times.

As mentioned previously, FRCA can be applied to any routing schemes like *Unipath* and *Anypath* routing. Compared to general *Unipath* and *Anypath* routing, *Opportunistic Routing* improves the network performance by considering benefits of the broadcast characteristic of the wireless medium and the duty-cycle mechanism of sensor nodes. It is clear that the forwarding process of *OR* is much more complicated than other routing schemes. The performance analysis of FRCA for other routing schemes is similar with that for *OR* without considering multiple receivers problem. Therefore, the performance of FRCA for *OR* will be analyzed as follows while the analysis for other routing schemes will not be covered here.

In *OR*, the number of active candidates assigned to the sender node in each hop has a significant influence on the performance of FRCA. Hence, before analyzing the energy consumption, the redundant packets and the waiting times, we will find the probability of the multiple receivers.

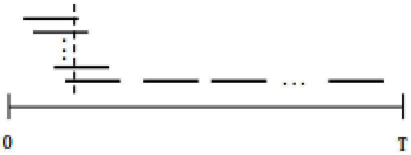


Fig. 8. The intervals of candidates.

We assume sensor nodes are asynchronous. The length of each node's active period which can be considered as a continuous interval with a randomly defined starting point are equal, denoted by t'. The starting point of each interval is randomly selected within $[0, T]$, where T is the initialization time of the network, as shown in Fig. 8. At a certain point in time, if there are more than one intervals intersecting with each other, then the sender node will have multiple active candidates. To facilitate the computation of the multiple receivers probability, t' and $[0, T]$ are normalized to t and $[0,1]$. We assume the maximum number of candidates assigned for a given sender node n_i is v_i which is formulated by (29), where $\kappa_{i,j}$ is defined by (30) and m_i is the number of n_i's neighbors. We are interested in finding the multiple receivers probability $Pmr(v_i, \eta)$ when there are $\eta (1 \le \eta \le v_i)$ active candidate(s) assigned for the sender n_i. The probability is actually equal to the probability that there are exactly η overlapping intervals while none of the rest $v_i - \eta$ intervals are intersected.

$$v_i = \sum_{j=1}^{m_i} \kappa_{i,j} \quad (29)$$

$$\kappa_{i,j} = \begin{cases} 1, & \tilde{\delta}_{i,j} \ge \dfrac{1}{m_i} \\ 0, & \tilde{\delta}_{i,j} < \dfrac{1}{m_i} \end{cases} \quad (30)$$

When $\eta \geq 2$, there are η overlapping intervals. Among these overlapping intervals, we assume that the distance between the starting points of the first and the last interval is $t - x$, where $0 \leq x \leq t$. With the exception of the starting points of the first and the last interval, the rest starting points of the overlapping intervals are effectively constrained to a region of length $t - x$. Besides, there are still $\nu_i - \eta$ intervals which do not intersect with each other. The starting points of the first interval in the overlapping intervals and the remaining $\nu_i - \eta$ intervals can be randomly placed in a span of length $1 - (\nu_i - \eta + 1) \cdot t + x$. The η overlapping intervals are randomly selected from ν_i intervals but starting points of $\eta - 2$ intervals are placed without considering the order in the region between starting points of the first and last intervals. Therefore, the probability $Pmr(\nu_i, \eta)$ can be formulated by (31).

$$\begin{aligned} Pmr(\nu_i, \eta) &= \frac{P(\nu_i, \eta)}{(\eta - 2)!} \\ &\cdot \int_0^t (t - x)^{\eta - 2} \cdot [1 - (\nu_i - \eta + 1) \cdot t + x]^{\nu_i - \eta + 1} dx \\ &= \nu_i! \cdot (\nu_i - \eta + 1) \\ &\cdot \left\{ \frac{t^{\nu_i}}{\nu_i!} + \sum_{k=1}^{\nu_i - \eta + 1} \frac{t^{\nu_i - k}}{k!\,(\nu_i - k)!} \cdot [1 - (\nu_i - \eta + 1) \cdot t]^k \right\} \end{aligned} \quad (31)$$

When $\eta < 2$, none of the intervals intersects with each other. The probability of this case is expressed in (32).

$$Pmr(\eta < 2) = [1 - (\nu_i - 1) \cdot t]^{\nu_i} \ (32)$$

5.1 Expected Energy Cost

According to the *First Order Radio Model*, the energy cost $T_X(i, j, k)$ for transmitting a packet of size k from n_i to n_j is formulated by (33), where E_{elec}, ϵ_{fs} as well as ϵ_{amp} are constants and $d_* = \sqrt{\epsilon_{fs} / E_{elec}}$ is the distance threshold. $d_{i,j}$ is the *Euclidean distance* between the sender n_i to n_j. The energy cost $R_X(j, k)$ for receiving a data packet of size k by n_j is formulated by (34).

$$T_X(i, j, k) = \begin{cases} E_{elec} \cdot k + \epsilon_{fs} \cdot k \cdot d_{i,j}^2 & d_{i,j} < d_* \\ E_{elec} \cdot k + \epsilon_{amp} \cdot k \cdot d_{i,j}^4 & d_{i,j} \geq d_* \end{cases} \ (33)$$

$$R_X(j, k) = k \cdot E_{elec} \ \ (34)$$

When a sender node wants to forward a packet to one of its candidates, it will first transmit a beacon packet to its candidates. If the candidates are active, they will receive the beacon packet and send back an ACK packet to the sender. After receiving the ACKs from the candidates, the sender node will identify which of its candidates are active. According to the network routing policy, the sender node determines the final candidate forwarder and send the data packet to it. The expected total energy cost when there are η active candidate(s) assigned for the sender n_i is denoted by $T_E(i, j, k, \eta)$. $T_E(i, j, k, \eta)$ can be divided into three parts, the energy cost $T_B(i, j, k_b, \eta)$ for transmitting a beacon packet, the energy cost $T_A(i, j, k_a, \eta)$ for transmitting ACK packets and the energy cost $T_D(i, j, k)$ for transmitting the packet. k_b is the size of the beacon packet, and k_a is the size of the ACK packet. The $T_B(i, j, k_b, \eta)$, $T_A(i, j, k_a, \eta)$ and $T_D(i, j, k)$ are respectively expressed in (35), (36) and (37).

$$\begin{aligned} T_B(i, j, k_b, \eta) &= T_X(i, j, k_b) + \sum_{j=0}^{\eta} R_X\,(j, k_b) \\ &= k_b \begin{cases} E_{elec} \cdot + \epsilon_{fs} \cdot d_{i,j}^2 + \eta \cdot E_{elec} & d_{i,j} < d_* \\ E_{elec} \cdot + \epsilon_{amp} \cdot d_{i,j}^4 + \eta \cdot E_{elec} & d_{i,j} \geq d_* \end{cases} \end{aligned} \quad (35)$$

$$\begin{aligned} T_A(i, j, k_a, \eta) &= \sum_{j=0}^{\eta} T_X\,(j, i, k_a) + \sum_{j=0}^{\eta} R_X\,(i, k_a) \\ &= k_a(\eta \cdot E_{elec} + \sum_{j=0}^{\eta} \begin{cases} E_{elec} \cdot + \epsilon_{fs} \cdot d_{j,i}^2 & d_{i,j} < d_* \\ E_{elec} \cdot + \epsilon_{amp} \cdot d_{j,i}^4 & d_{i,j} \geq d_* \end{cases}) \end{aligned} \quad (36)$$

$$\begin{aligned} T_D(i, j, k) &= T_D(i, j, k) + R_D(j, k) \\ &= k \begin{cases} E_{elec} \cdot + \epsilon_{fs} \cdot d_{j,i}^2 + E_{elec} & d_{i,j} < d_* \\ E_{elec} \cdot + \epsilon_{amp} \cdot d_{j,i}^4 + E_{elec} & d_{i,j} \geq d_* \end{cases} \end{aligned} \quad (37)$$

Therefore, the expected energy cost $T_E(i, j, k, \eta)$ when there are η active candidate(s) assigned for the sender n_i can be formulated by (38).

$$T_E(i, j, k, \eta) = T_B(i, j, k_b, \eta) + T_A(i, j, k_a, \eta) + T_D(i, j, k)(38)$$

Finally, according to the probability of exactly η active candidates assigned for the sender node n_i in (31), the expected energy cost $T_{hop}(i, j)$ of each hop is obtained by (39).

$$T_{hop}(i, j) = \sum_{\eta=1}^{\nu_i} P\,mr(\nu_i, \eta) \cdot T_E(i, j, k, \eta) \ (39)$$

Based on (39), the expected energy cost for delivering the data packet through the routing path $\mathbb{P} = \{n_1, n_2, \dots, n_\varrho\}$ is formulated by (40) where ϱ denotes the number of nodes in the path $\mathbb{P}$.

$$\begin{aligned} T_{path} &= \sum_{y}^{\varrho} T_{hop}\,(y, y + 1) \\ &= \sum_{y}^{\varrho - 1} \sum_{\eta=1}^{\nu_y} P\,mr(\nu_y, \eta) \cdot T_E(y, y + 1, k, \eta) \end{aligned} \quad (40)$$

5.2 Waiting Times

The *Average Waiting Times* (AWT) represents the average number of times the sender waits for at least one of the potential forwarders to wake up and receive the data packets. Based on (32), the AWT of the packet in one hop can be obtained by (41).

$$AWT_i = [1-(\nu_i - 1)\cdot t]^{\nu_i} \quad (41)$$

Thus, the AWT along the routing path $\mathbb{P} = \{n_1, n_2, \ldots, n_\varrho\}$ is expressed by (42).

$$AWT_{path} = \sum_{y=0}^{\varrho} AWT_y = \sum_{y=0}^{\varrho} [1-(\nu_y - 1)\cdot t]^{\nu_y} \quad (42)$$

5.3 Redundant Packets

Before determining the final forwarder, the sender node needs to identify which of its candidates are active. It broadcasts a beacon packet. All its awaken candidates will receive the beacon packet and send back an ACK packet. According to the network routing policy, the sender selects one of its candidates to forward the packet while the other candidates will drop the received packets. These aborted packets are considered as redundant. The expected number of redundant packets in each hop is formulated by (43).

$$AxRP_i = \sum_{\eta=2}^{\nu_i} \eta \cdot Pmr(\nu_i, \eta) \quad (43)$$

Consequently, the expected number of redundant packets along the path $\mathbb{P} = \{n_1, n_2, \ldots, n_\varrho\}$ is expressed by (44).

$$AxRP_{path} = \sum_{y=1}^{\varrho} \sum_{\eta=2}^{\nu_y} \eta \cdot Pmr(\nu_y, \eta) \quad (44)$$

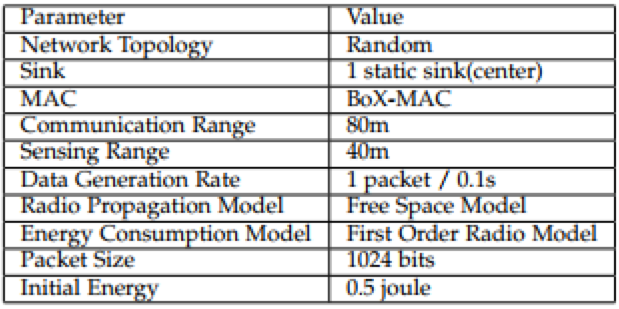
6 PERFORMANCE EVALUATION

In this section, two simulations will be conducted to verify the validity and flexibility of FRCA respectively. First, the performance of the routing protocol based on FRCA and the distribution proposed in Section 4 will be evaluated and compared with two well-known WSN protocols, ORW and ORR. Second, the flexibility of the routing protocol based on FRCA will be intuitively demonstrated by modifying the network policy on the fly.

6.1 Simulation Settings

All the simulations are conducted within an on-purpose simulator written in visual studio 2017 (C# WPF). The source code and the documents are available on our website: http://www.123.com. In our simulations, the sensor nodes are randomly deployed while the sink is placed in the center of the field. All these nodes are homogeneous. All of them run BoX-MAC and have the same duty cycles. The propagation model used here is the *free space model*, and the communication range of nodes is 80m. Each node which consumes the energy according to the *First Order Radio Model* has the battery of 0.5 Jouls. We assume the network generates one packet from a random sensor node to the sink in each 0.1s. The length of the data and control packets are 1024 bits and 512 bits respectively. The forwarding probability distribution of the routing protocol based on FRCA in the simulation is related to the neighbors' residual energy. In order to save energy, we assume that the sensor node only broadcasts its residual energy information to its neighbors at each time it loses 5% of the energy. The default parameters for simulations are shown in Table. 4. We compare the performance of our approach with two well-known WSN protocols:

TABLE 4
DEFAULT PARAMETERS



Parameter	Value
Network Topology	Random
Sink	1 static sink(center)
MAC	BoX-MAC
Communication Range	80m
Sensing Range	40m
Data Generation Rate	1 packet / 0.1s
Radio Propagation Model	Free Space Model
Energy Consumption Model	First Order Radio Model
Packet Size	1024 bits
Initial Energy	0.5 joule

1) ORW, a low latency opportunistic routing protocol using the *Expected Duty-Cycled wakeups* (EDC) as its routing metric. EDC which measures the expected number of wakeups in the packet path from the source to the sink is a global routing metric and needs recursive computation. Sender nodes always select candidates with high EDC to forward packets. ORW does not consider the load-balancing problem, so the nodes with high EDC will deplete their energy too early due to the concentrated traffic load. The EDC and candidate set of each node are only calculated at the network initialization and will not change until the network's topology changes again.

2) ORR, a load-balanced opportunistic routing protocol that improves ORW by using a *Forwarder Score* (FS) as its routing metric and limiting the number of candidates explicitly. FS is also a global routing metric which considers the residual energy of sensor nodes by EDC and also needs recursive computation. The sink periodically collects the energy information of the entire network, calculates the optimal number of candidates and candidate set of nodes and distributes the information to each sensor node. In our simulations, the α parameter is set to 1.0. The authors of ORR evaluated the performance of their work by varying α from 0 to 4 and pointed out that α greater than 1.0 has small impacts on the performance.

6.2 Evaluation Metrics

The following evaluation metrics are considered in our simulations. 1) ***Energy Consumption*** **(EC)**: The total energy consumption required to deliver a given number of packets from the source nodes to the sink. The energy unit is Joule. 2) ***Average Number of Redundant Packets*** **(AxRP)**: When the sender node broadcasts a packet to its candidate set, all its awaken candidates will receive the packet. Only one candidate is allowed to finally forward the packet while the other nodes drop the received packet after coordination. All these aborted packets are considered as redundant. AxRP is the average number of redundant packets after delivering a given number of packets from the sources to the sink. 3) ***Average Waiting Times*** **(AWT)**: When all the candidates of the sender node are at the sleep state, it has to wait until at least one of its candidates wakes up. AWT is the average number of times the sender waits after delivering a given number of packets from each randomly selected source to the sink. 4) ***Network Lifetime*** **(NL)**: The number of packets delivered to the sink until the first node of the entire network completely depletes its energy.

6.3 Verifying the Validity

In this subsection, the performance of the network varying the number of nodes and the network varying the wake-up intervals are evaluated respectively in detail.

6.3.1 Varying the Number of Nodes

We evaluate the network varying the number of nodes from 100 to 200. Each sensor node has the same active time 1s and sleep time 2s. Other parameters are the same with the default simulation settings presented previously. The total number of data packets generated by randomly selected sensor nodes in this simulation is set to 3000. The *Energy Consumption*, *Average Number of Redundant Packets*, and *Average Waiting Times, as well as the Network Lifetime,* are demonstrated and discussed as follows.

(a) **Energy Consumption**: The results of evaluating the energy consumption varying the number of nodes are shown in Fig. 9. The energy consumption of the three routing protocols all increase as the number of nodes becomes larger for two reasons. First, the density of nodes increases as the number of nodes becomes larger, which will increase the size of candidate sets. In consequence, sender nodes with more candidates generate more redundant packets and consume more energy. Second, increasing nodes may also increase the size of the network. As a result, the routing distance and the number of hops increase, which leads to a rise in the total energy consumption. FRCA outperforms ORW and ORR, as shown in Fig. 9, due to its better limit on the candidate number. ORW does not limit the number of candidates explicitly while ORR collects information of the entire network to calculate the optimal number of candidates n_{max} for all nodes of the network. However, it is inappropriate to use a global n_{max} to restrict the candidate number of nodes with a different number of neighbors. In FRCA, the forwarding probability distribution of each sender adapts to its neighbor number. Thus, FRCA can limit the candidate number more efficiently to avoid generating excessive, redundant packets which in turn reduces the energy consumption.

(b) **Average Number of Redundant Packets**: The results of evaluating the average number of redundant packets varying the number of nodes are depicted in Fig. 10. The redundant packets of the three protocols all increase as the number of nodes become larger since higher network density undoubtedly leads to an increase in the size of the candidate set of sender nodes. FRCA outperforms ORW and ORR for its better limit of candidate number, as explained above.

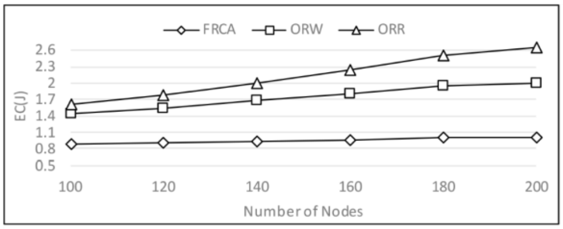


Fig. 9. Energy consumption varying number of nodes.

(c) **Average Waiting Times**: The results of evaluating the average waiting times varying the number of nodes are presented in Fig. 11. Higher network density imposes the sender to select more candidates which leads to an increase of both AxRP and energy consumption as illustrated previously but meanwhile may also reduce the average waiting times. More candidates imply that the sender does not need to wait for a specific candidate to wake up, which will effectively reduce the number of waiting times. Consequently, the average waiting times of ORW and ORR is slightly reduced as the number of nodes increases. In FRCA, the size of each node's candidate set grows slightly with the network density since the direction distribution concerning with the local density of nodes limits the number of candidates. Hence, the average waiting times of FRCA does not change significantly when the number of nodes varies from 100 to 140. However, the average waiting times not only depends on the network density but also on the network size. The increasing network size due to the increasing number of nodes will enforce packets to travel through a longer routing path and wait for more times. Thus, when the number of nodes increases to 160, the AWT of FRCA even slightly increases due to the growth of the network size. When the number of nodes continues to grow, the network density starts to have a greater impact

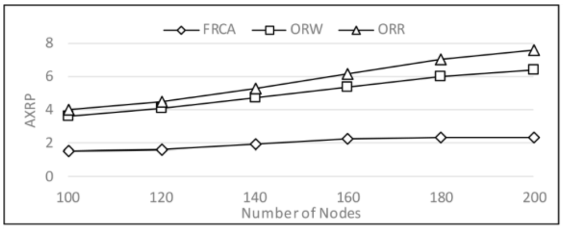


Fig. 10. Average number of redundant packets varying the number of nodes.

on the average waiting times than the network size, which leads to a reduction in the average waiting times.

(d) **Network Lifetime**: The results of the network lifetime varying the number of nodes are shown in Fig. 12. From Fig. 12, we can see that the performance of ORW is the worst among the three protocols and the performance of FRCA is nearly equal to or even better than that of ORR. ORW does not consider the load-balancing problem, so sensor nodes deplete their battery earlier. In order to prolong the network lifetime, ORR periodically collects residual energy information of all sensor nodes in the network to the sink and then calculates and distributes the candidate set for each node based on the global routing metric FS. Without the help of the sink, FRCA deploys distributed routing metric to determine the candidate set of each sender node and achieves a better lifetime performance. FRCA uses four distributions to reduce energy waste and prolong the network lifetime. First, the residual energy distribution enforces packets to travel through nodes with higher residual energy. Second, the direction distribution strictly limits the number of candidates. Third, the perpendicular distance distribution enforces packets to travel through a shorter routing distance. Last, the transmission distance distribution enforces packets to be delivered to the targeted candidate with lower transmission cost. All the quantities in the forwarding probability distribution work out to prolong the network lifetime.

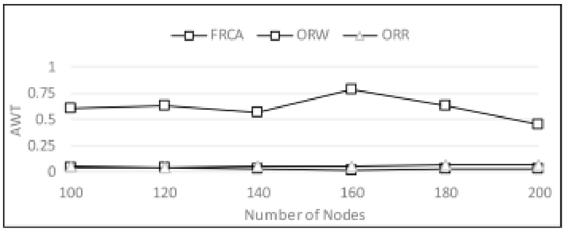


Fig. 11. Average waiting times varying number of nodes.

6.3.2 Varying the Wake-up Intervals

We evaluate the performance of the network for delivering 3000 data packets from randomly selected sources to the sink varying the wake-up intervals. The sleep time of nodes is fixed to 2s while the active time varies from 1s to 5s. Other parameters are the same as default simulation settings.

(a) **Energy Consumption**: The results of the energy consumption varying the active periods are depicted in Fig. 13. The multiple receivers probability of the sender increases as the active period grows, which will inevitably lead to a rise in the number of redundant packets and the total energy consumption. Therefore, the energy consumption of the three protocols all increase as the active period grows. We can see from Fig. 13 that the increasing active period has a less impact on FRCA since the number of the sender's candidates is under the strict limit to reduce energy consumption as explained before.

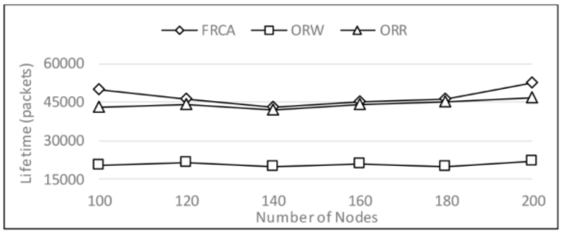


Fig. 12. Network lifetime varying number of nodes.

(b) **Average Number of Redundant Packets**: The results of the average number of redundant packets varying the active periods are shown in Fig. 14. Longer active periods implies the sender will generate more redundant packets since its multiple receivers probability increases. FRCA outperforms ORR and ORW due to its strictly limited candidate number as explained previously.

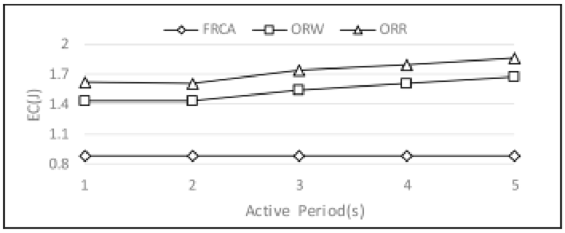


Fig. 13. Energy consumption varying active periods.

(c) **Average Waiting Times**: The results of the average waiting times varying the active periods are shown in Fig. 15. The increment of multiple receivers probability will increase the number of active candidates, which means that the sender does not need to wait for a specific candidate to wake up. Thus, the waiting times for each path gets smaller as the active periods increase. As shown in Fig. 15, the average waiting times of the three protocols all reduced.

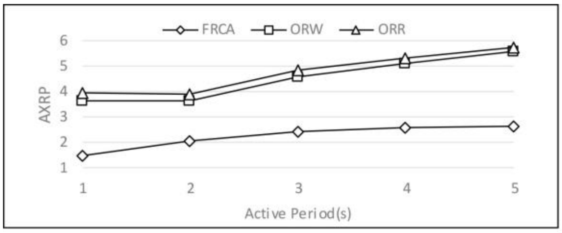
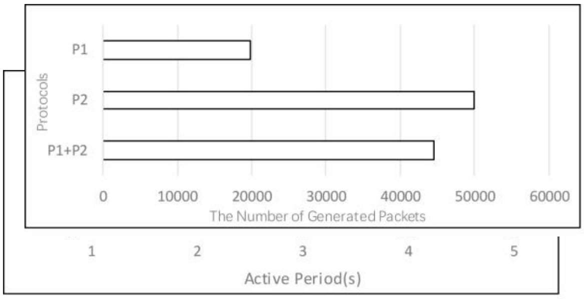


Fig. 14. Average Number of Redundant Packets varying active periods.

(d) **Network Lifetime**: The results of the network lifetime varying the active periods are depicted in Fig. 16. Longer active periods means more redundant packets will be generated by the sender, which will undoubtedly lead to a growing energy consumption. Therefore, we can conclude that the network lifetime decreases as active periods increase. As shown in Fig. 16, FRCA outperforms ORW and ORR for the same reasons explained previously.

6.4 Verifying the Flexibility

In this subsection, we are going to verify the flexibility of routing protocols designed based on FRCA. The number of nodes is set to 100. The active/sleep periods of each sensor node are set to 1s and 2s respectively. Other parameters are set to the default simulation settings.



ng
ol-
icy P2 respectively.

Fig. 15. Average Waiting Times varying active periods.

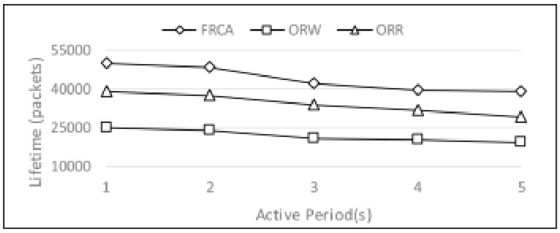


Fig. 16. Network lifetime varying active periods.

At the initialization of the network, sensor nodes make routing decisions according to the initial built-in network policy which is called Policy P1. Policy P1 is a simple routing policy, in which the sender node broadcasts a beacon packet to its neighbor nodes that are closer to the sink and lets the first node that receives the beacon become the final forwarder. When the total number of packets reaches 2000, Policy P1 will be updated to the new routing policy that we propose in Section 4, called Policy P2, using the parameters shown in Table. 2 and Table. 3. There are many approaches to deliver these parameters to each end node. Flooding is the simplest approach but incurs high energy consumption. Hierarchical approaches [22] establishing a virtual hierarchy of nodes can deliver the parameters to end nodes efficiently, such as grid-based [23], cluster-based [24], tree-based [25], and area-based [26] approaches. In this simulation, we use a simple parameters delivery method implemented in our source code. The variations of the network performance between Policy P1 and Policy P2 are shown in Fig. 17, Fig. 18, Fig. 19, and Fig. 20. We record and recalculate the *Energy Consumption*, *Average Number of Redundant Packets*, and *Average Waiting Times* every 500 packets.

(a) **Energy Consumption**: The variations of the energy consumption are depicted in Fig. 17. We conclude that Policy P1 consumes more energy than Policy P2. In Policy P1, the sender generates excessive, redundant packets since all the neighbor nodes that are closer to the sink will become candidates, which undoubtedly consumes much more energy than Policy P2 which strictly restricts the candidate number. We can intuitively see from Fig. 17 that Policy P1 has been upgraded to Policy P2 when the number of packets generated by the network reaches 2000 due to the sharp reduction in energy consumption.

(b) **Average Number of Redundant Packets**: The variations of AxRP are shown in Fig. 18. The average number of redundant packets reduces sharply after changing Policy P1 to Policy P2. As discussed before, the redundant packets of Policy P2 are less than Policy P1 due to the decline in sender nodes' candidate number.

(c) **Average Waiting Times**: The variations of AWT are shown in Fig. 19. The average waiting times of the network does not change dramatically as shown in Fig. 19. We can conclude that Policy P2 is better than Policy P1 because Policy P2 successfully reduces the energy consumption and the average number of redundant packets without increasing the average waiting times.

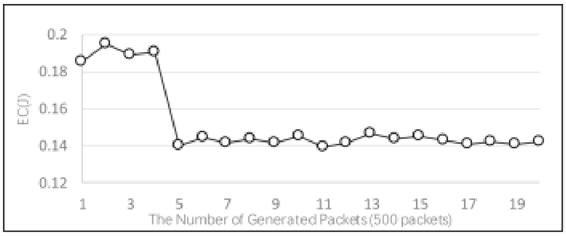


Fig. 17. The variations of the Energy consumption after changing network routing policy.

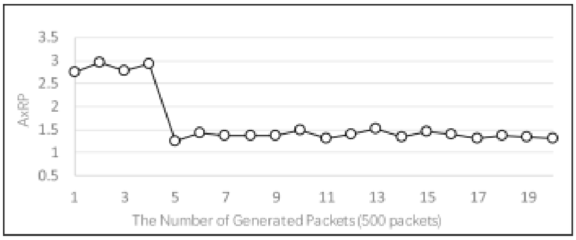


Fig. 18. The variations of the average number of redundant packets after changing network routing policy.

(d) **Network Lifetime**: The lifetime of the network running Policy P1, Policy P2 and Policy P1+Policy P2 are shown in Fig. 20. The network running Policy P1 can deliver approximately 20000 packets from randomly selected nodes to the sink until the first node depletes its battery while the network running Policy P2 can deliver approximately 50000 packets. In this simulation, we change the routing policy of the network running Policy P1 at first to Policy P2 when the total number of packets reaches 2000. As a result, the network lifetime increases from the original 20000 packets to approximately 45000 packets.

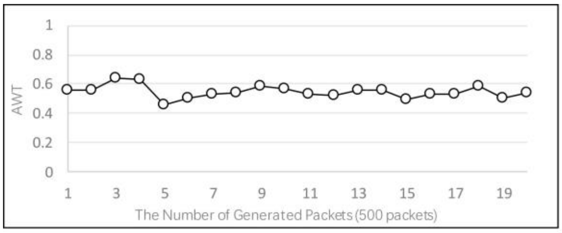


Fig. 19. The variations of the average waiting times after changing network routing policy.

From the simulation results, we conclude that routing protocols which are designed based on FRCA have high flexibility to adjust network routing policies to meet various and complicated requirements on the fly.

7 CONCLUSION

Considering the complicated and high dynamicity of environment situations and application requirements of WSNs, flexibility is a significant factor that enhances the performance and needs to be considered when designing routing protocols. This paper proposes a flexible routing computing approach (FRCA) for WSNs which can efficiently improve the flexibility of routing protocols. Routing protocols designed based on FRCA allow network operators to change network behaviors by modifying or upgrading the network policies on the fly without interrupting the network services while conventional routing protocols lacking flexibility have to recycle and redeploy sensor nodes, which is too costly. FRCA consists of three parts, the physical quantities collected at the sensor nodes, the built-in base math functions and the routing parameters which are all highly customizable. The simulation results demonstrated that routing protocols designed based on FRCA achieve better performance compared to the state-of-the-art solutions regarding network lifetime, energy consumption and duplicate packets as well as ensure high flexibility.

ACKNOWLEDGMENTS

This paper is supported by the "Fundamental Research Funds for the Central Universities NO. WK2150110007" and by the National Natural Science Foundation of China (NO.61772490, 61472382, 61472381 and 61572454).

REFERENCES

[1] W. Gong, K. Liu, and Y. Liu, "Directional diagnosis for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1290–1300, 2015.

[2] H. Xu, X.-Y. Li, L. Huang, H. Deng, H. Huang, and H. Wang, "Incremental deployment and throughput maximization routing for a hybrid sdn," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 3, pp. 1861–1875, 2017.

[3] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: A survey," *IEEE Communications surveys & tutorials*, vol. 15, no. 2, pp. 551–591, 2013.

[4] U. Wijetunge, A. Pollok, and S. Perreau, "Fault-tolerant stochastic routing for wireless sensor networks with unreliable links," in *Communications Theory Workshop (AusCTW), 2013 Australian*. IEEE, 2013, pp. 87–92.

[5] T. Roosta, M. Menzo, and S. Sastry, "Probabilistic geographic routing protocol for ad hoc and sensor networks," *Proceeding (474) Wireless Networks and Emerging Technologies*, 2005.

[6] F. Sivrikaya, T. Geithner, C. Truong, M. A. Khan, and S. Albayrak, "Stochastic routing in wireless sensor networks," in *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*. IEEE, 2009, pp. 1–5.

[7] A. Hawbani, X. Wang, A. Abudukelimu, H. Kuhlani, A. Qarariyah, and A. Ghannami, "Zone probabilistic routing for wireless sensor networks," *IEEE Transactions on Mobile Computing*, 2018.

[8] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, "Low power, low delay: opportunistic routing meets duty cycling," in *Proceedings of the 11th international conference on Information Processing in Sensor Networks*. ACM, 2012, pp. 185–196.

[9] A. Boukerche and A. Darehshoorzadeh, "Opportunistic routing in wireless networks: Models, algorithms, and classifications," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, p. 22, 2015.

[10] J. So and H. Byun, "Load-balanced opportunistic routing for duty-cycled wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 7, pp. 1940–1955, 2017.

[11] N. McKeown, "Software-defined networking," *INFOCOM keynote talk*, vol. 17, no. 2, pp. 30–32, 2009.

[12] I. T. Haque and N. Abu-Ghazaleh, "Wireless software defined networking: A survey and taxonomy," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2713–2737, 2016.

[13] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

[14] S. Costanzo, L. Galluccio, G. Morabito, and S. Palazzo, "Software defined wireless networks: Unbridling sdns," in *Software Defined Networking (EWSDN), 2012 European Workshop on*. IEEE, 2012, pp. 1–6.

[15] T. Luo, H.-P. Tan, and T. Q. Quek, "Sensor openflow: Enabling software-defined wireless sensor networks," *IEEE Communications letters*, vol. 16, no. 11, pp. 1896– 1899, 2012.

[16] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "Sdn-wise: Design, prototyping and experimentation of a stateful sdn solution for wireless sensor networks," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 513–521.

[17] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on*. IEEE, 2000, pp. 10–pp.

[18] X. Wang, X. Wu, and X. Zhang, "Optimizing opportunistic routing in asynchronous wireless sensor networks," *IEEE Communications Letters*, vol. 21, no. 10, pp. 2302–2305, 2017.

[19] D. Moss and P. Levis, "Box-macs: Exploiting physical and link layer boundaries in low-power networking," *Computer Systems Laboratory Stanford University*, vol. 64, no. 66, p. 120, 2008.

[20] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer *et al.*, “Tinyos: An operating system for sensor networks,” in *Ambient intelligence*. Springer, 2005, pp. 115–148.

[21] A. Hawbani, X. Wang, Y. A. AL-SHARABI, A. Ghannami, H. Kuhlani, and S. Karmoshi, “Load-balanced opportunistic routing for asynchronous duty-cycled wsn,” *IEEE Transactions on Mobile Computing*, 2018.

[22] C. Tunca, S. Isik, M. Y. Donmez, and C. Ersoy, “Distributed mobile sink routing for wireless sensor networks: A survey,” *IEEE communications surveys & tutorials*, vol. 16, no. 2, pp. 877–897, 2014.

[23] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, “Ttdd: Two-tier data dissemination in large-scale wireless sensor networks,” *Wireless networks*, vol. 11, no. 1-2, pp. 161–175, 2005.

[24] C.-J. Lin, P.-L. Chou, and C.-F. Chou, “Hcdd: hierarchical cluster-based data dissemination in wireless sensor networks with mobile sink,” in *Proceedings of the 2006 international conference on Wireless communications and mobile computing*. ACM, 2006, pp. 1189–1194.

[25] H. S. Kim, T. F. Abdelzaher, and W. H. Kwon, “Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks,” in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003, pp. 193–204.

[26] E. B. Hamida and G. Chelius, “A line-based data dissemination protocol for wireless sensor networks with mobile sink,” in *Communications, 2008. ICC’08. IEEE International Conference on*. IEEE, 2008, pp. 2201–2205.