Research Article

A Multipopulation Firefly Algorithm for Correlated Data Routing in Underwater Wireless Sensor Networks

Ming Xu¹,² and Guangzhong Liu¹

¹ College of Information Engineering, Shanghai Maritime University, 1550 Haigang Avenue, Shanghai 201306, China
² Shanghai Key Laboratory of Intelligent Information Processing, 220 Handan Road, Shanghai 200433, China

Correspondence should be addressed to Ming Xu; mingxu@shmtu.edu.cn

Received 30 August 2012; Revised 3 December 2012; Accepted 31 January 2013

Academic Editor: Shabbir Merchant

Copyright © 2013 M. Xu and G. Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Low data delivery efficiency and high energy consumption are the inherent problems in Underwater Wireless Sensor Networks (UWSNs) characterized by the acoustic channels. Existing energy-efficient routing algorithms have been shown to reduce energy consumption of UWSNs to some extent, but still neglect the correlation existing in the local data of sensor nodes. In this paper, we present a Multi-population Firefly Algorithm (MFA) for correlated data routing in UWSNs. We design three kinds of fireflies and their coordination rules in order to improve the adaptability of building, selecting, and optimization of routing path considering the data correlation and their sampling rate in various sensor nodes. Different groups of fireflies conduct their optimization in the evolution in order to improve the convergence speed and solution precision of the algorithm. Moreover, after the data packets are merged during the process of routing path finding, MFA can also eliminate redundant information before they are sent to the sink node, which in turn saves energy and bandwidth. Simulation results have shown that MFA achieves better performance than existing protocols in metrics of packet delivery ratio, energy consumption, and network throughput.

1. Introduction

Underwater Wireless Sensor Networks (UWSNs) have a lot of potential application areas such as oceanographic data collection, disaster prevention, pollution monitoring, offshore exploration, and military surveillance [1–3]. UWSNs are a new area of wireless sensor network in underwater environments which has challenges to be overcome such as long propagation delay, severely limited bandwidth, and time-varying multipath propagation [4–6]. All the above distinct features of UWSNs give birth to new challenging areas for every level of the network protocol suite. UWSNs consist of a variable number of sensors and vehicles that are deployed to perform collaborative monitoring tasks over a given area. To achieve this objective, sensors and vehicles self-organize in an autonomous network in order to adapt to the characteristics of the underwater environment. Many problems arise with UWSNs that need to be solved in order to enable underwater monitoring in the new environment. Among them, providing efficient routing is very challenging due to the unique characteristics of UWSNs [7, 8]. UWSNs must rely on underwater acoustic communications because high-frequency radio signals used in traditional ground-based sensor networks can be rapidly absorbed by water. Compared with terrestrial wireless sensor networks, the channel of UWSNs has the following characteristics: (1) high bit error ratio caused by noise, multipath loss and Doppler propagation; (2) high energy consumption; (3) limited bandwidth; (4) long and unstable delivery delay [9, 10]. Therefore, many research results in land-based sensor networks as well as traditional ad hoc networks cannot be applied to UWSNs directly, which requires new routing protocol to be designed for the new features of the UWSNs in order to ensure that the performance of UWSNs can meet the actual underwater environmental needs.

Low data delivery efficiency and high energy consumption are the inherent problems in UWSNs characterized by the acoustic channels. Acoustic channel imposes higher energy consumption as well as lower bandwidth than radio signal. Therefore, improving packet delivery ratio and energy efficiency for UWSNs becomes even more critical than in traditional sensor networks.
According to their architectures, the routing protocols of UWSNs can be divided into three categories: location-based routing, flat routing, and hierarchical routing [11]. Location-based routing has good scalability, but it requires a positioning system or positioning algorithm to help the nodes calculate the location information. Flat routing protocols have better robustness, but the excessive overhead for maintaining routing information restricts their application to small-scale underwater acoustic circumstances. Hierarchical routing also has good scalability, but the cluster maintenance overhead and the failure of key nodes will affect the routing efficiency. Consequently, these routing protocols cannot fully meet performance requirements described above.

Firefly algorithm is the development of simulated evolutionary computation in the recent years [12]. It is simple, but appropriate to parallel processing with strong robustness, which extends its application to a number of fields such as solving economic dispatch and cryptography problems [13–15]. As the characteristic of firefly intelligence is similar to that of UWSN’s self-organized property, this paper presents a Multi-population Firefly Algorithm (MFA) for correlated data routing in UWSNs. In MFA, three kinds of fireflies coordinate to improve the adaptability of building, selecting, and optimization of routing path considering the data correlation and their sampling rate in various sensor nodes. As a result, MFA minimizes the total data transmission volume and optimizes the routing path concurrently. Furthermore, in order to increase the energy efficiency in resource constraint underwater environment, MFA balances the energy consumption during the process of routing in case some sensor nodes exhaust their energy too early.

The remainder of the paper is organized as follows. Section 2 gives a brief overview of related work. Section 3 presents the network model, while Section 4 introduces the technical details of our routing protocol. Performance evaluation is described in Section 5. Finally, we conclude the paper in Section 6.

2. Related Work

Tan et al. [16] proposed a new protocol based on hop-by-hop hybrid; implicit/explicit acknowledgment scheme is proposed for a multihop UWSN. In the protocol, data packets forwarded by downstream nodes can work as implicit ACKs for previous transmitted data packets, (where ACK is an abbreviation that refers to the Acknowledgement to the receipt of data packet). Pompili et al. [17] introduced two distributed routing algorithms for delay-insensitive and delay-sensitive applications, respectively, with the objective of minimizing the energy consumption taking the varying condition of the underwater channel and the different application requirements into account. Vector-Based Forwarding (VBF) [18] is a geographic approach, which allows the nodes to weigh the benefit to forward packets and reduce energy consumption by discarding low benefit packets. Therefore, over a multi-hop path, only the nodes that are located within a pipe of given width between the source and the destination are considered for relaying. However, in the areas of low density of nodes VBF may not find the path close to the routing vector. Similarly, Jornet et al. [19] proposed Focused-Beam Routing (FBR) protocol that is suitable for networks containing both static and mobile nodes. The objective of FBR is to determine which nodes are candidates for relaying. Candidate nodes are those that lie within a cone of angle $\pm \theta/2$ emanating from the transmitter towards the final destination. An RTS/CTS handshake is set up to isolate closer nodes within this cone. If a node determines that it is within the transmitter’s cone, it will respond to the RTS. Those nodes that are outside the cone will not respond. A theoretical argument supporting geographic routing has been discussed in [20] based on simple propagation and energy consumption models for underwater networks. In view of this, several position-based routing algorithms are proposed and compared; results show that selecting relays closer than a given maximum distance before seeking farther ones achieves in fact optimal energy consumption.

Depth-Based Routing (DBR) [21] can handle network dynamics efficiently without the assistance of a localization service. DBR forwards data packets greedily towards the water surface (i.e., the plane of data sinks). In DBR, a data packet has a field that records the depth information of its recent predecessor and is updated at every hop. But DBR has only greedy forwarding mode, which alone is not able to achieve high delivery ratios in sparse areas. A similar idea can be found in [22], a hydraulic pressure-based anycast routing protocol that exploits the measured pressure levels to route data to surface buoys where relays are chosen based on a weighted average of advancement towards the sources and probability of packet delivery, and an efficient underwater dead end recovery method is added to handle the absence of a relay node at lower depth than the current packet holder. Yang and Su [23] designed an energy-efficient routing protocol called EUROP, where they tried to reduce large amount of energy consumption by reducing broadcast hello messages. The sensor nodes use RREQ and RREP packets to communicate with each other, and the next hop can be determined by the rule of from deep to shallow and so on. In EUROP, installing the depth sensor and electronic module is not a simple decision because cost per node will increase and the additional devices will burden the critical node energy, hence decreasing the life of the sensor node.

In order to remove the constraints imposed by special hardware (e.g., every node should be equipped with depth or pressure sensor, which not only increases the cost of the network but also becomes a burden for extra energy consumption), Ayaz and Abdullah [24] proposed a Hop-by Hop-Dynamic Addressing Based (H2-DAB) routing protocol to provide scalable and time efficient routing for UWSN. The H2-DAB routing protocol does not require any dimensional location information or any extra specialized hardware compared with many other routing protocols in the same area. Dynamic addresses are used for sensor nodes in order to solve the problem of water currents, so that sensor nodes will get new addresses according to their new positions at different depth intervals. However, the problem of multi-hop routing still exists as it is based on multi-hop architecture, where nodes near the sinks drain more energy because they are used more frequently. Domingo [25] presented a distributed
energy-aware routing protocol called DUCS (Distributed Underwater Clustering Scheme) where the whole network is divided into clusters using a distributed algorithm. Sensor nodes are organized into local clusters where one node is selected as a cluster head for each. After receiving the data packets from all the cluster members, cluster head performs signal processing function like aggregation on the received data and transmits them towards the sink using multi-hop routing through other cluster heads. DUCS achieves a relatively high packet delivery ratio and reduces the network overhead. But frequent division of sectors can be a burden on the network as the setup phase is repeated many times. Moreover, during the network operation phase, a cluster head can only transmit its collected data towards another cluster head, which restricts its flexibility and availability. For example, when water currents move two cluster head nodes away, they cannot communicate directly even if a few noncluster head nodes are available between them.

Packet redundancy and multiple paths can be exploited in order to increase the reliability of UWSNs. Ayaz et al. [26] provided a two-hop acknowledgment reliability model in order to insure the reliable data deliveries to the surface sinks, where two copies of the same data packet are maintained in the network without extra burden on the available resources. A relay node that has data packets to forward will not reply the acknowledgment until it cannot find the next hop towards the destination. But if a node is unable to find the next hop due to any failure, or even if it is lost, then packets in the buffer are not considered lost. All the nodes that send the data packets towards this node will wait for a certain amount of time before trying again for the next hop. Xu et al. [27] proposed a Multiple-path Forward Error Correction (M-FEC) approach that integrated multiple-path communications and Hamming Coding to eliminate retransmission and enhance reliability in underwater sensor networks. Moreover, a Markov model and a dynamical decision and feedback scheme were developed to decrease the number of the paths in order to save energy and ensure the desirable packet error rate. However, M-FEC may cause much long delay because of additional process of encoding and decoding the data packets, and it does not tackle the issues of node and link failures.

**3. Network Model**

The routing algorithm to be presented in this paper is built on a Peer-to-peer (P2P) based UWSN platform that consists of acoustic sensor nodes, relay nodes, sink nodes, and P2P nodes as shown in Figure 1.

Acoustic sensor nodes are deployed in the monitoring region, which are responsible for data collection, processing, and communications. Each acoustic sensor node has two data communication interfaces. One is acoustic communication interface, through which the acoustic transducer can communicate with acoustic sensor nodes and others; the other interface can be used to connect relay nodes through CAN (Controller Area Network) bus or UART (Universal Asynchronous Receiver/Transmitter). When acoustic sensors are transmitting data, they deliver nodes’ states and data to relay nodes at the same time. The adoption of relay nodes makes the underwater nodes free from the bondage of cables, facilitating the deployment and control of underwater acoustic sensors and improving the structural integrity and robustness of UWSN. It can also expand the scope of monitoring. Relay nodes send information to sink nodes through wireless communication. At last, sink nodes communicate with P2P nodes through the serial ports and collect data to P2P nodes from various acoustic sensor nodes. After data stream arrives at P2P nodes, it will be put into the appropriate queue scheduler, where it can be processed by multithread operation processor. After that, the results will be sent to the router, which will decide the direction of the data: passing to archive or processing for output. Load balance strategies are used for preventing the platform from overloading. If the system performance is rather low, it notifies the load balancer, which will take measures to reduce the load on the system until it reaches the normal. The arriving data stream is dynamic. So even if the average load of a node is not high, a node is still likely encountering a temporary peak loads in the busy period, which extends its waiting time for data processing. Therefore, in order to minimize the data processing time, appropriate load balancing strategy is needed to avoid overloading.

We consider that UWSNs are composed of a large number of nodes uniformly scattered in monitoring fields and represented by $G = (V, E)$. The sensor nodes are assumed static, or they have low mobility with respect to signal propagation speed. Every sensor node has the same transmission range and each sensor node is able to communicate with other sensor nodes within its range. We also assume that each node is assigned with a triplet of coordinates $(x, y, z)$, where each coordinate represents the hop distance of the node from one anchor. We assume that the set of $n$ sensor nodes is represented as a set $N$ with $n = |N|$. All sensor nodes have the same
communication range of \( r \), which is represented as a sphere volume of radius \( r \) in UWSNs. Moreover, we assume all sensor nodes know their own locations through a certain localization service [28]. Such assumption is justified in underwater systems where fixed bottom-mounted nodes have location information upon deployment. In fact, the underwater localization is a nontrivial task for which relatively very few options are available. For example, employing Global Positioning System (GPS) does not work well because Radio Frequency (RF) waves are heavily attenuated in underwater environment. Many researchers have proposed a variety of localization schemes and techniques to address this issue specially [29, 30]. It is not always feasible to deploy anchor nodes at the sea floor for deep water environment. In this case, mobile beacon nodes such as Autonomous Underwater Vehicles (AUVs), which are equipped with internal navigation systems, are exploited as reference nodes to assist in corresponding distributed localization algorithms. This paper takes advantage of these research results as existing preconditions.

Definition 1. The function \( \delta(u, v) \) defines the distance between two nodes \( n_u \) and \( n_v \) in a 3D Euclidean space as

\[
\delta : N \times N \rightarrow \Gamma : \delta(u, v),
\]

\[
\delta(u, v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2 + (u_z - v_z)^2}.
\] (1)

Two nodes \( n_u \) and \( n_v \) are neighbors and connected by a link if \( \delta(u, v) < r \) and the link between \( n_u \) and \( n_v \) is denoted by \( e(u, v) \). We construct the network topology with Relative Neighborhood Graph (RNG) [31], then two nodes become neighbor nodes if and only if for any arbitrary node \( n_{j_p} \), \( \delta(u, v) \leq \max\{\delta(p, u), \delta(p, v)\} \). For a three-dimensional Euler space embedded, if the arcuate area formed by the intersection of two spheres centered at \( n_u \) and \( n_v \) (with radius \( \delta(u, v) \)) is empty, then \( n_u \) and \( n_v \) are adjacent nodes. RNG algorithm is simple and is easily built in a distributed way. There is no crossing edges in an RNG because at least one edge in any pair of crossing edges must be removed according to their definitions, and the time complexity is \( O(n^2) \). While constructing from a Delaunay Triangulation Graph structure, its complexity of lower bound is \( O(n \log(n)) \) [32]. In addition, a computational method with the complexity of \( O(n^2) \) for the RNG in a three dimensional space is given in [33]. As underwater sensors float with currents, their movements are constrained in different horizontal planes and they are likely to maintain a steady position relative to each other. The construction of RNG does not require that the exact positions of nodes and their neighbors are known. For each node, only the corresponding mutual distances to its neighbors are required. Therefore, RNG is expected to be more suitable in modeling UWSN, which achieves more accurate results and behaves more consistently than other models.

Considering two sensor nodes at minimum hop distance \( h \), there exist two values \( u(h) \) and \( v(h) \) such that the Euclidean distance \( \delta(u, v) \) between the two nodes is bounded, that is, \( u(h) \leq \delta(u, v) \leq v(h) \). The quality of the bounds depends on the network density \( \rho \). In particular for each \( h > 0 \) it holds that

\[
\lim_{\rho \to \infty} v(h) - u(h) = r,
\] (2)

where \( r \) is the transmission range of the sensors nodes.

Sensing devices generally have widely different theoretical and physical characteristics. Thus, numerous models of varying complexity can be constructed based on application needs and device features. However, for most kinds of sensors, the sensing ability diminishes as distance increases.

Definition 2. Given a source sensor node \( n_s \) and a destination sensor node \( n_d \), then the sensor model is defined as

\[
S(u, v) = \frac{\varphi}{\delta(u, v)^k + \xi}
\] (3)

with \( \lambda > 0 \) and \( \xi > 0 \), where \( \varphi \) is the signal amplitude, \( k \) is the sensor technology dependent parameter and \( \xi \) is a predefined parameter, for handling the situation when the source node and the destination node locate at the same position.

4. Routing Protocol

4.1. Data Correlation and Energy Consumption Model. Data aggregation is important in energy constraint UWSNs where spatial and temporal correlations as well as redundant data can be detected to reduce the number of packets exchanged in the network. In this paper, spatial correlation means that multiple sensor nodes in a nearby region record the same or similar data about a single event; whereas temporal correlation means that the change pattern of the data in current sensor node is equal or similar to the data observed at previous times, and the degree of correlation between consecutive observations may vary according to the temporal variation characteristics of the phenomenon.

Given two sensor node sets \( A \) and \( B \), where \( B \subseteq A \subset V \). When packets in all sensor nodes of the set \( A-B \) can be correctly received by the sensor nodes in the set \( B \), it means the data can be aggregated from the set \( A \) to the set \( B \).

Definition 3. Given a data sequence \( X_1, X_2, \ldots, X_N \) about the value of the field being measured with the sampling time \( T_1, T_2, \ldots, T_N \), then the \( k \)-step self-correlation of this data sequence is defined as

\[
SR_X(k) = \sum_{i=1}^{N-k} X_i \cdot X_{i+k}.
\] (4)

Definition 4. Given two data sequences \( X_1, X_2, \ldots, X_N \) and \( Y_1, Y_2, \ldots, Y_N \) about the value of the field being measured with the sampling time \( T_1, T_2, \ldots, T_N \) then the \( k \)-step cross-correlation of these two data sequences is defined as

\[
CR_{XY}(k) = \sum_{i=1}^{N-k} X_i \cdot Y_{i+k}.
\] (5)

Given a sensor node \( n_u \) and its homogeneous neighbor node set \( N_u = \{ n_{i_1}, n_{i_2}, \ldots, n_{i_m} \} \), then the correlation
combination of the data sequence generated by \( n_\alpha \) and the data sequences generated by its homogenous neighbor node \( n_d \) \((i = 1, 2, \ldots, m)\) are called homogenous data correlation (SC). Similarly, given a sensor node \( n_\alpha \) and its heterogenous neighbor node set \( \text{ND}_\alpha = \{n_{u1}, n_{u2}, \ldots, n_{um}\} \), then the correlation combination of the data sequence generated by \( n_\alpha \) and the data sequences generated by its heterogenous neighbor node set \( n_{u\beta} \) \((j = 1, 2, \ldots, n)\) are called heterogenous data correlation (DC).

The correlated data packets can be merged in order to decrease communication traffic, which in turn cuts down the overall energy consumption of the network. Suppose \( d_i \) is the data packet created at the sensor node \( n_i \) and \( d_j \) is the data packet created at \( n_j \)’s neighbor \( n_j \). Let \(|d_i|\) and \(|d_j|\) denote the size of these two packets. When they meet together, they are merged into a new packet and its size is calculated as

\[
|d_i \cdot d_j| = \max (|d_i|, |d_j|) + (1 - \rho_{ij}) \min (|d_i|, |d_j|),
\]

where \( \rho_{ij} \) denotes the correlation coefficient between the data packet \( d_i \) and the data packet \( d_j \).

Let \( N_v = \{n_u | \delta(v, u) < r, n_u \in N\} \) denote the set of neighbors of node \( n_v \), then the neighbors that are closer to the sink node \( n_v \) are called downstream nodes of \( n_v \), which is represented as \( D_v = \{n_k | \delta(k, v) < \delta(v, t), n_k \in N\} \). Similarly, the set of upstream nodes of \( n_v \) is denoted by \( A_v = \{n_k | \delta(k, v) > \delta(v, t), n_k \in N\} \). During the routing process, a sensor node usually forwards the data packets created in the upstream nodes or local nodes to the downstream nodes. Suppose \( \omega_{uv} \) denotes the ratio of packets sent from the upstream nodes of \( n_u \) to the node \( n_v \) and \( \theta_{uv} \) denotes the ratio of packets sent from the node \( n_v \) to the node \( n_u \), then we get

\[
\sum \omega_{uv} + \sum \theta_{uv} = 1, \tag{7}
\]

where the node \( n_u \) is the upstream node compared with the node \( n_v \).

Let \( \lambda_u \) and \( \lambda_v \) denote the transmission rate of the node \( n_u \) and the node \( n_v \), then

\[
\lambda_v = \sum_{n_k \in A_v} (\lambda_u \omega_{uv} + \eta_u \theta_{uv} (1 - P_{uv})), \quad 0 \leq P_{uv} \leq 1, \tag{8}
\]

where \( \eta_u \) denotes the data generating rate of node \( n_u \), and \( P_{uv} \) denotes the average correlation coefficient between the packets at the node \( n_u \) and the packets at the node \( n_v \). Therefore, \( \sum_{n_k \in A_v} (\lambda_u \omega_{uv}) \) denotes the transmission rate that is forwarded from the upstream nodes of \( n_u \) to the node \( n_v \), and \( \sum_{n_k \in A_v} (\eta_u \theta_{uv} (1 - P_{uv})) \) denotes the transmission rate that is directly generated by the set of upstream nodes of \( n_u \).

Acoustic signal has different transmission modes in shallow water (where the depth of the water is less than 100 meters) and deep water (where the depth of the water is more than 100 meters). In shallow water, the transmission of the acoustic signal is limited to a cylindrical area from the bottom to the surface. The energy consumption for transmission in the shallow water is calculated as

\[
e_t = 10 \log_5 \delta (u, v) + \alpha \delta (u, v) \cdot 10^3, \tag{9}
\]

where \( \delta(u, v) \) denotes the Euler distance between the sender and the receiver, and \( \alpha \) is the absorption coefficient.

In deep water, the transmission of the acoustic signal is mainly with spherical diffusion. The energy consumption is caused by spherical diffusion and water absorption, which can be calculated as

\[
e_t = 20 \log_5 \delta (u, v) + \alpha \delta (u, v) \cdot 10^3. \tag{10}
\]

4.2. Routing Algorithm. Routing in UWSNs has its own particularity. Firstly, computation, storage, and communication capabilities of the sensor nodes are relatively weak, and therefore cannot be achieved on a complex routing algorithm. Secondly, UWSNs have data-centric architectures, which are different from those choosing the path according to nodes addresses in the traditional networks. Firefly algorithm uses the points of the search space to simulate firefly individuals in nature and transfers the process of search and optimization into the process of firefly individuals attraction and move. It simulates solving problems in the objective function metrics into the pros and cons of the individual location. It continues to change the decision-making domain size based on the number of neighbor nodes with a strong global search ability and convergence.

Firefly algorithm has two essential factors: fluorescence intensity and attractiveness. Fluorescence intensity reflects the pros and cons of the firefly location and determines its direction of movement, while the degree of attractiveness determines the distance that a firefly has moved. Fluorescence intensity and attractiveness are constantly updated in order to achieve the objective of optimization. From the mathematical point of view, the optimization mechanism of the firefly is described as follows.

Definition 5. Fluorescence intensity of a firefly is defined as

\[
I(r) = I_0 e^{-\gamma r^2}, \tag{11}
\]

where \( I_0 \) is the original fluorescence intensity at the location of \( r = 0 \), and \( \gamma \) is the fluorescence absorption coefficient.

As firefly attractiveness is proportional to the fluorescence intensity seen by adjacent fireflies, we can now define the attractiveness of a firefly as follows.

Definition 6. Attractiveness of a firefly is defined as

\[
\beta(r) = \beta_0 e^{-\gamma r^2}, \tag{12}
\]

where \( r \) is the distance between each two fireflies and \( \beta_0 \) is their attractiveness at \( r = 0 \), that is, when two fireflies are found at the same point of search space.

The distance between any two fireflies \( i \) and \( j \) at \( x_i \) and \( x_j \) can be expressed as Cartesian distance \( r_{ij} = ||x_i - x_j|| \). The firefly \( i \)'s displacement that is attracted to another more attractive firefly \( j \) is determined by

\[
\Delta x_i = \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \cdot \epsilon, \tag{13}
\]
which is randomized with the vector of random variable $e_i$ being drawn from a Gaussian distribution and step factor $\alpha \in [0, 1]$. The movements of underwater sensors are driven not only by the flow of water, but also by other uncertain factors such as tides, animal interference, and ships. Therefore, it is not easy to build a precise kinematic model to characterize the underlying mobility patterns of the underwater sensors. For simplicity, we assume that the sink nodes are stationary and other sensor nodes follow the random-walk mobility pattern in simulations. The firefly’s displacement is used to capture the network mobility properties.

In order to realize the multi-population firefly-based routing protocol, we define three kinds of fireflies: searching firefly, listening firefly, and updating firefly.

**Definition 7.** Searching firefly (sf): each sensor node can create one or more searching fireflies, which can move among the nodes in the network and store the collected sensor nodes’ information into its mobile data stack (ds) in order to share with other searching fireflies.

The task of sf is to find a routing path from the source node to the sink. When a source node creates an sf, it also gives a TTL value to this sf and the value decreases by 1 after this sf moves from one node to another. When the TTL value equals 0 or this sf moves to a sink node, then the lifetime of this sf is terminated. As TTL value can affect the performance of packet delivery ratio, energy consumption, and network throughput of different routing protocols in UWSNs, we have conducted a couple of simulations to reveal their quantitative relationships. The default TTL value is empirically set to 6 according to the size of region and the number of sensor nodes in simulation. Therefore, the choice of TTL value is not related to the type of data and its temporal/spatial correlation.

**Definition 8.** Listening firefly (lf): each sensor node in wakeup state owns a listening firefly, which is responsible for listening the information of mobile data stack in searching firefly and changing its routing path.

The searching firefly decides its moving direction according to the fluorescence intensity of the listening fireflies in its neighbor nodes. Suppose node $n_i$ has a searching firefly $sf_i$ and, $n_i$’s neighbor node $n_j$ has a listening firefly $lf_j$. The probability that $sf_i$ moves to the node $n_j$ is

$$p_{ij} = \frac{I_j - I_i}{\sum_{k \in N_i} (I_k - I_i)}.$$

where $N_i$ is the neighbor set of node $n_i$.

When the searching firefly $sf_i$ moves to the node $n_j$ in the routing path, the listening firefly $lf_j$ at the node $n_j$ loses part of fluorescence and the wastage can be calculated as

$$\Delta I_j = |I_i - I_j|^{\mu} \cdot e^{-\sigma t'},$$

where $0 < \mu < 1$ represents the fluorescence depletion rate. The listening firefly $lf_j$ at the node $n_j$ loses its fluorescence only when other searching fireflies move to the node $n_j$ if $n_j$ is a source node. In these cases, the node $n_j$ is considered as a relay node that belongs to other routing paths.

When two searching fireflies created by the same source node meet at the same node, then the searching firefly with smaller TTL value stops moving and transfers its mobile data stack to the listening firefly at this node. The searching firefly with bigger TTL value continues to search until it reaches a sink node or runs out its lifetime.

**Definition 9.** Updating firefly (uf): Each sink node can create an updating firefly after a searching firefly’s arriving, which is in charge of updating the fluorescence intensity of the nodes along the routing path for further selection and optimization of the routing path.

When the searching firefly reaches a sink node, an updating firefly is created to update the fluorescence intensity of the nodes along the routing path. The initial fluorescence intensity of the updating firefly is

$$I(u) = \frac{I_s}{\|x_u - x_i\|^r},$$

where $I_s$ is the fluorescence intensity of the searching firefly at the sink node. $0 < \sigma < 1$ represents the fluorescence update rate. When an updating firefly moves to a node $n_i$ in the routing path, it will transfer a few fluorescence to the listening firefly at that node and the quantity is

$$\Delta I_i = I(u) \cdot \frac{\|x_u - x_i\|}{\sum_{n_j \in P} \|x_n - x_j\|},$$

where $n_j$ is a random node in the routing path $P$.

Algorithm 1 describes our Multi-population Firefly Algorithm for correlated data routing in detail.

In order to prevent flooding, all packets at the relay nodes should have limited lifetime, which are controlled by TTL (Time-To-Live) information. At first, the routing path mes.path is set to empty after initialization and the node $n_i$ retrieves the message mes as described from line 1 to line 2. If mes hits at the node $n_{n_i}$, then $n_i$ will create an updating firefly $uf_s$ according to (16). After that, $uf_s$ travels along the routing path and updates listening firefly’s fluorescence intensity for each node it visited according to (17). Otherwise, mes.path will be added to the mobile data stack of the current routing node. And then, the routing path mes.path is returned, which indicates a successful routing process is finished as described from line 3 to line 12. If the node $n_i$ is not the sink node, then $n_i$ will retrieve a data packet $d_i$ and create a searching firefly $sf_i$ in order to find a routing path. And then, $n_i$ is added to the routing path mes.path. In the next procedure, $sf_i$ moves to the neighbor node $n_j$ according to (14) and calculates the displacement value according to (13). At the same time, the listening firefly $lf_i$’s fluorescence intensity at the node $n_j$ should be updated according to (15) as described from line 13 to line 21. Moreover, the data correlation of packet $d_i$ is checked with the data in the neighbor nodes. If their correlation relationship is within the predefined correlation threshold interval, then the two packets are merged into a new
Input: source node \( n_s \), sink node \( n_t \), correlation threshold interval \( \text{Init}_{\text{val}} \) and the related parameters for the firefly algorithm;

Output: optimized routing path \( \text{mes}\cdot path \);

1. \( \text{mes}\cdot path \leftarrow \Phi \) //Initialization
2. Node \( n_s \) retrieves \( \text{mes}\);
3. if \( \text{mes} \) hits at the node \( n_t \) then
4. Creates a updating firefly \( u_f_s \) according to formula (16);
5. for all sensor nodes in \( \text{mes}\cdot path \) do
6. if \( n_p \in \text{mes}\cdot path \) then
7. Updates \( n_p \)'s listening firefly’s fluorescence intensity according to formula (17);
8. else
9. \( n_p\cdot ds-add(\text{mes}\cdot path) \)
10. endfor
11. return \( \text{mes}\cdot path \);
12. else
13. if \( n_s \) is not the sink node then
14. Node \( n_s \) retrieves a data packet \( d_i \);
15. Node \( n_s \) creates a searching firefly \( s_f_s \);
16. \( \text{mes}\cdot path\cdot add(n_s) \);
17. while \( s_f_s\cdot TTL > 0 \) do
18. Moves \( s_f_s \) to the neighbor node \( n_j \) according to formula (14);
19. Calculates the displacement value according to formula (13);
20. Updates the listening firefly \( f_j \)'s fluorescence intensity at \( n_j \) according to formula (15);
21. if \( d_i \in n_j\cdot attr_j \) then
22. Calculates \( d_i \)'s homogenous data correlation \( SC_d_i \) with relation to \( n_j\cdot attr_j \);
23. else
24. Calculates \( d_i \)'s heterogenous data correlation \( DC_d_i \) with relation to \( n_j\cdot attr_j \);
25. endif
26. if \( SC_d_i \in \text{Init}_{\text{val}} \) or \( DC_d_i \in \text{Init}_{\text{val}} \) then
27. \( d_i \leftarrow \prod (d_i, d_j) \);
28. \( \lambda_j = \lambda_j - \psi(n_j\cdot SC_d_i + n_j\cdot DC_d_i) / 0 < \psi < 1 \)
29. else
30. \( d_i \leftarrow d_i \cup d_j \);
31. \( \lambda_j = \lambda_j + (n_j\cdot SC_d_i + n_j\cdot DC_d_i) / \psi; / 0 < \psi < 1 \)
32. endif
33. \( \text{mes}\cdot path\cdot add(n_j) \);
34. Sends the data packet \( d_i \) to the Node \( n_j \);
35. \( n_j \leftarrow n_j \);
36. \( s_f_s\cdot TTL = s_f_s\cdot TTL - 1 \);
37. endwhile
38. else
39. if \( s_f_s\cdot TTL = 0 \) then
40. return \( \text{mes}\cdot path \);
41. endif
42. endif
43. endif
44. endif

Algorithm 1: Multi-population Firefly Algorithm (MFA).

packet. For the sake of saving energy, the corresponding node will decrease its sampling rate as calculated according to line 29, where \( 0 < \psi < 1 \) denotes the extent of sampling rate variation. Otherwise, if their correlation relationship is beyond the predefined correlation threshold interval, then the two packets will execute union operation. And then, the corresponding node will increase its sampling rate as calculated according to line 32 in order to improve data processing speed. After that, the data packet \( d_i \) is sent to the node \( n_j \) and the TTL value is decreased by 1 for the next
iteration process as described from line 34 to line 39. If the TTL value equals 0, then the routing path mes.path is also returned as described from line 40 to line 44.

When traditional firefly algorithm runs at its late stage, individual firefly may hover near the peak or even oscillate. In addition, when the search space is too larger or distributes with uneven density, some fireflies’ neighbor set becomes empty set, which results in slow convergence and easily falls into local optimum. In our multi-population firefly-based routing algorithm, after defining three kinds of firefly groups, different groups of fireflies access the information not only from their own group, but also from other fireflies groups to conduct their own optimization in the evolution. In this way, three groups of fireflies coevolve in order to improve the convergence speed and solution precision of the algorithm.

Collision occurs when two or more nodes send data at the same time over the same transmission channel. Medium Access Control (MAC) protocols have been developed to assist each node to decide when and how to access the channel, which allows the sensor nodes to transmit data packets on the basis of a predefined schedule that will not cause the packet collision. If underlying MAC protocols are not available or out of service, MFA can adopt a round-robin scheduling method in order to avoid collision. The whole process is divided into two phases: round determination phase and timeslot allocation phase. During the routing process, each node calculates its round number based on the hops in current routing path and each round is scheduled sequentially. As a result, a downstream node usually gets bigger round number than an upstream node does. After that, each node will assign timeslots in pairs with its neighbors for packet transmission. The length of each round is determined by the number of timeslots that are necessary to avoid conflicts. Due to nonuniform deployment, some nodes may experience high traffic loads and cause more collisions than other nodes. Therefore, the number of timeslots varies from one round to another and is proportional to the fluorescence intensity of the listening fireflies in the neighbors. In this way, the initial setting of timeslot values may not be suitable for every node in the network. It achieves better collision avoidance as MFA approaches optimization in the evolution.

MFA requires location awareness of neighbors, but no topology information needs to be exchanged among neighboring nodes. It is a localized and distributed routing algorithm.

4.3. Theoretical Analysis. Suppose all \( n \) sensor nodes are evenly deployed in a monitoring region with size \( h \times h \times h \). The transmission range of each sensor node is \( r \). The average node degree is \( d \). We use \( w \) to denote the routing pipe radius in VBF and \( l \) (\( l < \text{TTL} \)) to denote the average path length in MFA. The comparison of the complexity of different routing protocols is listed in Table 1.

In the best case, MFA achieves the best routing performance among three routing protocols. This happens when searching flies find a shortest path from the source node to the sink. In the worst case, DUCS needs more routing hops than others because cluster head nodes may not always reachable and have to be selected again from normal nodes. MFA consumes the most memory space because each searching fly needs to store the collected sensor nodes’ information into its mobile data stack along the routing path. But \( O(dL) \) memory consumption is not a burden in MFA as the length of the routing path is restricted by the TTL value and the average node degree in UWSN is usually not high.

### 5. Performance Evaluation

5.1. Simulation Settings. We use Aqua-Sim [34] as simulation framework to evaluate our approach. The data packets are also generated by the simulator. Aqua-Sim is an ns-2-based underwater sensor network simulator developed by underwater sensor network lab at University of Connecticut. Aqua-Sim can simulate the attenuation and propagation of acoustic signals. It can also simulate packet collision in underwater sensor networks.

Simulation parameters and their default values are listed in Table 2. We assume that the sink nodes are stationary and the sensor nodes follow the random-walk mobility pattern. Each sensor node randomly selects a direction and moves to the new position with a random speed between the minimal speed and maximal speed, which are \( 1 \text{ m/s} \) and \( 5 \text{ m/s} \), respectively.

We use the following metrics to evaluate the performance of routing protocols:

1. **Packet delivery ratio** is defined as the ratio of the number of distinct packets received successfully at the sinks to the total number of packets generated at the sinks.
source node; although a packet may reach the sinks several times, these redundant packets are considered as only one distinct packet.

(2) **Energy consumption** takes into account the total energy consumed in packet delivery, including transmitting, receiving, and idling energy consumption of all nodes in the network.

(3) **Network throughput** equals the total data bits received at the sinks divided by the simulation time.

We compared the performance of Multi-population Firefly Algorithm (MFA) with that of Vector-Based Forwarding (VBF) protocol and Distributed Underwater Clustering Scheme (DUCS). As MFA and DUCS exploit data aggregation to eliminate redundancy in order to enhance their routing performance, we conducted a couple of simulations to reveal their quantitative relationships and the correlation coefficients are set from 0.0 to 1.0 with a fixed interval of 0.1. In order to evaluate other metrics, the correlation coefficients are set to a medium value that equals to 0.5 in MFA and DUCS.

### 5.2. Simulation Results

In the first set of simulations, we compared the packet delivery ratio with the number of nodes in different routing protocols. The TTL value for each protocol is set to 6. As shown in Figure 2, the packet delivery ratio of three routing protocols is proportional to the number of nodes. MFA performs better than other routing protocols in the same circumstances. When the number of nodes does not exceed 400, the packet delivery ratio of DUCS is higher than that of VBF, but the result reverses as the number of nodes is more than 400 because DUCS needs more operations for the maintenance of clusters. Specifically, MFA improves 17.3% of the packet delivery ratio than that of DUCS and 19.6% of the packet delivery ratio than that of VBF on average.

Figure 3 illustrates the comparison of the packet delivery ratio with the TTL value in different routing protocols. The number of nodes is set to 400 for each protocol. The packet delivery ratio of three routing protocols is proportional to the TTL value. MFA achieves higher packet delivery ratio than that of DUCS and VBF when their TTL values are the same. When the TTL value does not exceed 5, the packet delivery ratio of VBF is higher than that of DUCS, but the result reverses as the TTL value reaches 6 and above because with the increase of the TTL value, VBF takes more cost in order to find the path close to the routing vector. Overall, MFA improves 11.8% of the packet delivery ratio than that of DUCS and 13.6% of the packet delivery ratio than that of VBF on average.

Figure 4 shows the comparison of the packet delivery ratio with the correlation coefficient in different routing protocols. The TTL value is set to 6 and the number of nodes is set to 400 for each protocol. MFA performs better than other routing protocols in the same circumstances. The packet delivery ratio of MFA and DUCS is proportional to the correlation coefficient, while the packet delivery ratio of VBF is fixed to 57% as VBF does not exploit data aggregation during the routing process. When the correlation coefficient does not exceed 0.3, the packet delivery ratio of DUCS is lower than that of VBF. When the correlation coefficient is not less than 0.4, the packet delivery ratio of DUCS is higher than that of VBF, which means the using of data aggregation technique is effective for improving the performance of DUCS. On
average, MFA improves 14.1% of the packet delivery ratio than that of DUCS and 14.8% of the packet delivery ratio than that of VBF.

In the second set of simulations, we compared the energy consumption with the number of nodes in different routing protocols. As shown in Figure 5, the energy consumption of three routing protocols is proportional to the number of nodes. MFA performs better than other routing protocols in the same circumstances. Moreover, the curve of MFA has more gentle slope compared with that of DUCS and VBF. Specifically, when the number of nodes increases from 100 to 600, the energy consumption growth rate of MFA is 37.5%, while the energy consumption growth rates of DUCS and VBF obtain 52.5% and 58.2%, respectively. Overall, MFA decreases 25.7% of the energy consumption than that of DUCS and 38.8% of the energy consumption than that of VBF on average.

Figure 6 illustrates the comparison of the energy consumption with the TTL value in different routing protocols. The number of nodes is set to 400 for each protocol. The energy consumption of three routing protocols is proportional to the TTL value. MFA consumes less energy than that of DUCS and VBF when their TTL values are the same. Moreover, the curve slope of MFA is gentler than that of DUCS and VBF. For example, when the TTL value is 3, MFA consumes 84.6% of the energy more than that of DUCS and 77.9% of the energy more than that of VBF; when the TTL value reaches 10, MFA consumes 67.2% of the energy more than that of DUCS and 53.1% of the energy more than that of VBF. On average, MFA decreases 28.1% of the energy consumption less than that of DUCS and 34.4% of the energy consumption less than that of VBF.

Figure 7 shows the comparison of the energy consumption with the correlation coefficient in different routing protocols. The TTL value is set to 6 and the number of nodes is set to 400 for each protocol. MFA performs better than DUCS and VBF in the same circumstances. The energy consumption of MFA and DUCS is inversely proportional to the correlation coefficient. However, the curve of MFA is gentler than that of VBF. Specifically, when the correlation coefficient is 0.1, MFA consumes 67.2% of the energy less than that of VBF; when the correlation coefficient is 0.9, MFA consumes 61.8% of the energy less than that of VBF. On average, MFA decreases 31.6% of the energy consumption less than that of VBF.
coefficient, while the energy consumption of VBF is fixed to \(1.49 \times 10^4\) mJ as VBF does not exploit data aggregation during the routing process. With the aid of data aggregation, DUCS eliminates redundant information before they are sent to the sink nodes, which makes it save 18.9\% of the energy compared with that of VBF. With the increase of correlation coefficient, the curve of MFA indicates sharper decline than that of DUCS, which means MFA is more effective than DUCS in cutting down the energy consumption. Overall, MFA decreases 25.4\% of the energy consumption less than that of DUCS and 39.5\% of the energy consumption less than that of VBF on average.

In the third set of simulations, we compared the network throughput with the average data rate in different routing protocols. The TTL value is set to 6 and the number of nodes is set to 400 for each protocol. As shown in Figure 8, the network throughput of three routing protocols is proportional to the average data rate. The front parts of the curves indicate rapid increases in network throughput while the rear parts of the curves show very slow growth rates after average data rate is above some value. The reason is that each node is responsible for sending its own traffic towards the sink nodes as well as forwarding all traffic from its upstream nodes towards the sink nodes. But both upstream nodes and downstream nodes have the same packet generating rates, and downstream nodes cannot receive packets from several of its upstream nodes simultaneously. MFA performs better than other routing protocols in the same circumstances. When the average data rate does not exceed 800 bits/s, the network throughput of VBF is higher than that of DUCS, but the result reverses as the average data rate is above 1 k bits/s, which means that DUCS can obtain more benefit than VBF when the average data rate is relatively high. On average, MFA improves 13.3\% of the network throughput more than that of DUCS and 15.2\% of the network throughput more than that of VBF.

Figure 9 illustrates the comparison of the network throughput with the number of nodes in different routing protocols. The TTL value is set to 6 for each protocol. The network throughput of three routing protocols is proportional to the number of nodes. MFA achieves high network throughput than that of DUCS and VBF when the number of nodes are the same. When the number of nodes is below 400, the network throughput of DUCS is higher than that of VBF, but the result reverses as the number of nodes reaches 500 and above because in the high density network, VBF contains more candidate sensor nodes that are close to the routing vector for the packet delivery. Overall, MFA improves 21.6\% of the network throughput more than that of DUCS and 25.6\% of the network throughput more than that of VBF on average.

Figure 10 depicts the comparison of the network throughput with the TTL value in different routing protocols. The number of nodes is set to 400 for each protocol. The network throughput of three routing protocols is proportional to the TTL value. MFA achieves higher network throughput than that of DUCS and VBF when their TTL values are the same. When the TTL value does not exceed 5, the packet delivery ratio of VBF is higher than that of DUCS, but the result reverses as the TTL value reaches 6 and above, which means the network throughput of VBF is restrained as the TTL value becomes relatively high. Specifically, MFA improves 16.7\% of the network throughput than that of DUCS and 16.3\% of the network throughput than that of VBF on average.
suitable for building a shorter routing path or not. Therefore, the packet delivery ratio in VBF mainly depends on the node density and it cannot benefit from the deployment of multiple sink nodes if they are not close to each other. DUCS organizes the sensor nodes into clusters and one sensor node is selected as a single cluster head for each cluster. Each cluster head is responsible for processing data aggregation received from cluster member nodes and relaying packets to other cluster heads until a sink node is reached, which restricts its flexibility and availability when adjacent cluster heads cannot communicate directly for uncertain reasons. In MFA, three kinds of fireflies cooperate to find an optimized routing path. The searching firefly is in charge of finding a reachable path from the source node to the sink node with a predefined TTL value. The listening firefly is used to guide the movement of the searching firefly according to its fluorescence intensity. If a data packet cannot arrive at a sink node within TTL hops, the lost fluorescence would not be replenished. Otherwise, the sink node generates an updating firefly and transfers some fluorescence to the listening firefly at each node along the routing path. As a result, the successful routing path is strengthened while the unsuccessful routing path becomes more unattractive, which brings a higher packet delivery ratio in MFA compared with VBF and DUCS regardless of node density. The inherent adaptive nature of such algorithm is one of the main attractions of biologically inspired approaches.

MFA also consumes less energy compared with VBF and DUCS. The reason is that MFA merges the data packets according to their correlation coefficient before they are sent to the sink node, which in turn eliminates redundant information and saves nodes’ residual energy. VBF does not exploit data aggregation during the process of routing. DUCS adopts aggregation techniques for extracting nonredundant data in clusterheads, but in order to avoid fast draining the batteries of specific sensor nodes, frequent division of clusters is needed, which can be a burden on the network as the setup phase is repeated many times. Therefore, MFA is more effective than DUCS in cutting down the energy consumption.

MFA achieves higher network throughput than that of DUCS and VBF. The reason lies in two aspects. First, the higher packet delivery ratio in MFA ensures more effective data packets arrive at the sink nodes, while unsuccessful data packet delivery does not contribute to the network throughput. Second, with the convergence of the algorithm, many overlapping edges in different routing paths emerge in MFA. Thus, some searching fireflies may have higher probability of choosing shorter paths to the sink nodes.

6. Conclusion

In order to improve the routing efficiency of UWSNs, this paper proposed a Multi-population Firefly Algorithm (MFA) for correlated data routing. We have designed three kinds of fireflies and their coordination rules in order to improve the adaptability of building, selecting, and optimization of routing path considering the data correlation and their sampling rate in various sensor nodes. Different groups of fireflies access to information not only from their own groups, but also from other firefly groups to conduct their optimization.
in the evolution. Moreover, in consideration of spatial and temporal correlations, MFA merges the correlated data packets during the process of routing, which in turn eliminates redundant information before they are sent to the sink nodes for saving energy and bandwidth. We have demonstrated through simulations that MFA achieves better performance than existing protocols in metrics of packet delivery ratio, energy consumption, and network throughput.

Acknowledgments

This work was sponsored by the National Natural Science Foundation of China (61202370), the Shanghai Pujiang Program from Science and Technology Commission of Shanghai Municipality (11PJ1404300), the Open Program of Shanghai Key Laboratory of Intelligent Information Processing (IIPL-2011-008), and the Science and Technology Program of Shanghai Maritime University (20110049).

References


