International Journal of Computer Trends and Technology
ISSN: 2231 – 2803 / https://doi.org/10.14445/22312803/IJCTT-V71I2P111
Volume 71 Issue 2, 68-77, February 2023

Original Article

Research on Improvement of Self-Selective Routing in Wireless Sensor Networks

Chi-Chang Chen[1], Jung-Hung Ko[2]

[1,2] *Department of Information Engineering, I-Shou University, Taiwan.*

Received: 10 January 2023 Revised: 11 February 2023 Accepted: 20 February 2023 Published: 28 February 2023

Abstract - *This article focuses on the improvement of Self-Selective Routing (SSR) protocol in wireless sensor network routing. In SSR, when two points want to transmit data to each other, the source point does not need to select and record the path as the transmission but only needs to record the hop count information on the transmitted packet. When the source point wants to send a message to the destination, it uses the established hop count to send a broadcast message to the nearby node closest to the target to be responsible for transmitting. This intermediary node is chosen arbitrarily and is responsible for transmitting the message by the quickest responding sensing node. This process is repeated until the packet reaches its destination. This article aims to improve the flooding method used during the initial search for the target point in the SSR protocol. We observed that under the SSR protocol, if the sensing nodes are evenly deployed in the sensing area, the actual path used when transmitting a message between two points is close to the straight line near these two points, and other remote area nodes are not used. Therefore, we need to limit the scope of flooding in the initial stage. Our method is to create zones for all nodes in the network and use the zones to limit the transmission range of flooding to exclude unnecessary nodes in the block from receiving and transmitting messages to conserve energy.*

Keywords - *Self-Selective Routing, Flooding mechanism, Wireless Sensor Network, Routing protocol.*

1. Introduction

The earliest appearance of Wireless Sensor Networks was in a research project at the University of California, Berkeley. The developers of the project used micro-electromechanical systems (MEMS) technology to develop a small-sized (similar in size to a regular coin) sensor called "Smart Dust" [1,2]. The project was funded by the Defense Advanced Research Projects Agency (DARPA) of the United States and was originally aimed at military applications.

Wireless Sensor Network (WSN) is typically a network formed by connecting small, low-cost, low-power, simple-computing, and short-range transmitting sensor nodes [4,5]. Such sensor nodes can be placed in large quantities in the environment to form a sensor network for detection tasks. These sensors usually rely on batteries to maintain normal operation, can work on their own without supervision, have a lifespan of several weeks to months, and are usually used to replace human labor in unfavorable environments.

Due to the random distribution of wireless sensors, each sensor cannot know its relative position to other sensors. Therefore, the sensor network must use self-configuration protocols to automatically develop a communication network between sensors so that the data sensed by all sensors in the network region can be transmitted to the data-collecting base station through the self-configured network.

In wireless sensor networks, data transmission is the part that consumes the most network energy [6]. Sensors must communicate through wireless transmission technology to send and receive messages with other sensors or base stations. Due to the limited transmission range of sensors and to conserve energy, sensors typically use multi-hop methods to transmit data. Currently, the design of most sensors sets the energy consumption target to -20dbm [6].

There are three commonly used data transmission methods in the physical layer of wireless sensor networks: (i) Flooding Flooding transmits data by broadcasting the data to be transmitted from the sensor to all neighboring nodes within its transmission range, and all neighbors then broadcast the data to their neighbors until the base station receives the data or the number of broadcasts reaches the limit. This method of transmitting data is not affected by changes in network topology. However, it will result in a large amount of repeated data being transmitted in the network, affecting the energy consumption of the network. (ii) Unicast Sensors communicate directly with the base station, possibly using multi-hop routing protocols or communicating one-on-one with the cluster head. This type of communication is called unicast. (iii) Multicast In multicast mode, neighboring sensors form groups as required by the application, and data transmission can be directly sent to each member of their own group.

Studies have shown that broadcasting is a better data transmission option for wireless sensor networks because sensors do not need to establish and maintain routing paths to the base station. The network architecture can be divided into static and dynamic based on whether the sensor and base station move or not. The sensor and base station are stationary in a static sensor network. This type of network often optimizes its network structure because the additional energy consumed by optimization is usually much less than the energy saved. The optimization algorithms are less complex and consume less energy. The optimization algorithm for a dynamic sensor network is relatively more complex and consumes more additional energy. At the same time, when the network architecture changes, the network often needs to be optimized again, which not only temporarily suspends the network's detection function but also consumes additional energy, so the optimization of dynamic sensor networks and the total additional energy consumption must be considered.

Regardless of whether it is a static or dynamic sensor network, data aggregation is an inevitable function to save energy consumption. In order to perform data aggregation, a certain number of neighboring sensors will form a group to perform data aggregation. Therefore, the sensor network can be seen as composed of groups of sensors. Further, multiple groups may form larger groups, and the entire network forms a hierarchical structure.

2. Background

Due to the characteristics of wireless sensor networks, such as decentralization, dynamic distribution, and multi-hop transmission, broadcasting is often used in message transmission. In a network topology with connectivity, this method can guarantee that all nodes receive the packet without the help of routing tables. Although the broadcast method is simple and ensures that the target node can be found, it also results in nodes having to check the packet multiple times, which causes excessive energy loss for the sensors. Moreover, not all broadcasts contribute to data transmission. Many routing protocols use broadcasting in the path search, such as FEB[7], DSR[8], AODV[9], ZRP[10], and LAR[11], which are used in sensor searches like Directed Diffusion and service search like NC-DD [12].

It is mentioned in [23] that using the broadcast method for message transmission in a MANET (mobile ad hoc network) environment will result in factors such as redundancy, contention, and collision, leading to broadcast storm problems and affecting network performance.

- **Redundancy :** When a node first receives a broadcast message and rebroadcasts it, some of the nodes within its transmission range may have already received this message, causing unnecessarily repeated broadcasts.
- **Contention :** When a node transmits a broadcast message, its neighboring nodes almost simultaneously receive it and decide to rebroadcast it. Due to the nature of wireless network bandwidth sharing and the similarity of rebroadcasting time will result in obvious transmission competition, causing transmission delay.
- **Collision :** Due to the almost simultaneous rebroadcasting phenomenon, collisions will occur, causing the receiving end not to be able to receive the message normally, resulting in wasting transmission bandwidth. At the same time, since the acknowledgement mechanism cannot confirm broadcast packet transmission, it will not be possible to confirm the success of rebroadcasting, leading to a decrease in broadcast coverage.

Traditional multi-hop routing, which uses routing tables to specify that neighbors forward packets to reach the destination, is known as multi-hop routing. Examples of this include MintRoute [14], Directed Diffusion, and AODV. These basic methods, which emulate traditional wired network communication, require maintaining the status of their neighbors to provide routing decisions. In addition, technology for measuring wireless link conditions is also required, and these routing protocols typically require additional overhead to provide typical wireless sensor network operations, particularly to support fault tolerance mechanisms.

2.1. Self-Selective Routing (SSR)

Most wireless networks use broadcast antennas for communication, unlike typical wired networks that use point-to-point communication. For multi-hop routing, SSR [15,16] has the characteristic of using broadcast communication without emulating point-to-point communication [22]. Therefore, in SSR, a sensing node transmitting a packet does not need to use a forwarding table to specify a neighbor to send the packet to the destination. Instead, the node closest to the destination with the lowest cost (or the minimum number of hops) among the neighbors receiving the broadcast packet will choose to be responsible for forwarding the packet to the destination. This concept is illustrated in Figure 1, which uses a network plane gradient to represent the transmission of packets to the destination at decreasing cost. The points indicated by the arrows in the figure are the nodes that receive the broadcast packet, and only one of the lowest-cost points will continue to forward.

The prominent features of SSR include fault tolerance, optimal path selection, and congestion avoidance, which are attributed to the self-selection technology. The self-selection technology relies on three broadcasts and one back-off delay, as shown in Figure 2. The first broadcast is a temporal implicit synchronization point at the beginning of the back-off delay, and the use of the back-off delay is to find the node with the highest expected characteristics when selecting. The second broadcast is a notification from the node with the shortest back-off delay to the previously broadcasted node to continue transmitting to the target node. The third broadcast informs other nodes participating in the competition of the result of this self-selection.

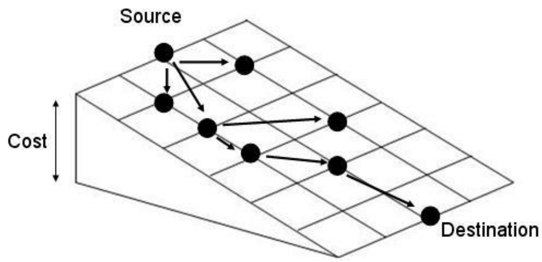


Fig. 1 SSR uses Cost index for transmission

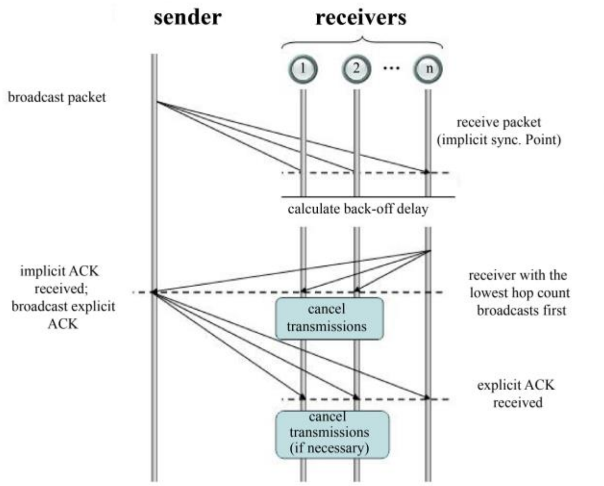


Fig. 2 SSR Algorithm

The SSR protocol consists of two stages: a path search program and a data transmission program.

2.1.1. Path Search Program

The data structure used by the SSR protocol is simple. Each node maintains a target node cost table. The table directory consists of: (i) the working node ID (source or destination), (ii) the latest sequence number observed from the target node and (iii) the hop distance from the target node to the current node.

The path search process consists of two stages, described as follows:

Destination Request Stage

When a source node wants to send a data packet to a target node and has not yet established a cost table directory, the source node uses a broadcast protocol to send a destination request (DREQ) packet and increments its own sequence number. Each DREQ packet has an actual hop count field to record the number of hops from the source node to the current receiving node, as shown in Figure 3 (a).

Source ID	Sequence Number	Destination ID	Actual Hop Count

(a)

Source ID	Sequence Number	Destination ID	Actual Hop Count	Expected Hop Count

(b)

Fig. 3 (a) destination request packet of SSR (b) destination reply packet of SSR

Suppose a relay node receives a DREQ packet from a source node and has not established a directory for the DREQ packet. In that case, the node will generate a new directory in the destination node's cost table, including the source node ID, the sequence number, and the actual number of hops. If the directory already exists, the directory update may occur (i) when the received DREQ packet sequence number is higher than that contained in the cost table or (ii) when the actual number of hops in the DREQ packet is lower than that recorded in the cost table.

Target Reply Phase

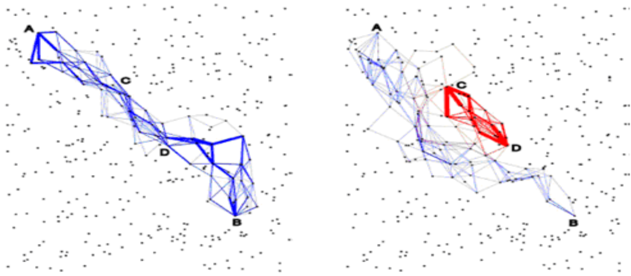
The destination node will reply with a destination reply (DREP) packet upon receipt of a new DREQ packet, as shown in Figure 2.3(b). The header of this packet contains the same fields as the DREQ packet and the expected hop count field, which indicates the expected number of hops for the packet to return to the source node (where the expected value is the number of hops from the destination to itself minus 1). The destination node simply broadcasts the DREP packet without specifying the next hop node. Instead, it stores the value of the expected number of hops minus 1 in the expected hop count field of the DREQ packet, using the number of hops in the destination node's cost table. Once the DREP packet is detected, each node checks the expected hop count field of the packet.

The main concept of the SSR protocol is to drive back-off delay based on the number of hops measured from the relay node to the destination node, with nodes closer to the destination node given higher priority to transmit packets. Through passive listening to all packets and observing their actual hop count fields, a relay node only knows the distance from the destination node to itself.

2.1.2. Data Transmission Program

When the source node receives a DREP packet, the source node can begin to transmit data packets to the target node. Data packets can be transmitted to the target node in the same way that the target node transmits the DREP packet to the source node, using the actual hop count field and self-selection. However, upon receipt of any data or DREP packet, the receiving node will update its own target node cost table to match the directory of the target node for that packet.

A less obvious feature of SSR is that it can automatically avoid congestion. In dense areas, nodes may have to wait for too many packets in the MAC queue. Even if a node has a large MAC queue assigned to a small back-off delay, it is highly likely that it will not be able to select itself quickly in self-selection, like other nodes in less congested areas.



(a) (b)

Fig. 4 (a) Normal communication between nodes A and B. (b) Due to congestion between nodes C and D, packet transmission between nodes A and B will automatically avoid congested paths.

Figure 2.4 illustrates the actual paths on that packets are transmitted. Figure 2.4 (a) depicts communication from node A to node B. Figure 2.4 (b) also depicts the same network, with the addition of communication between node C and node D. Figure 2.4 (b) shows that the SRR packet transmission environment is in a congested area. Despite the increase in path length from node A to node B, the end-to-end delay between endpoints is likely to decrease, as the increase in each hop delay in the new path is less than the original shorter path.

2.2. Path Initialization Problem

In the target request stage of the SSR protocol, the source node needs to find the target node. Initially, the broadcast protocol is used to transmit the target request packet to all nodes in the network. Therefore, as long as different nodes need to communicate with each other, the source node will broadcast the packet to the entire network again. Every time different nodes initialize the path, all nodes in the network need to consume a lot of energy to receive and broadcast the target request packet to find the target node.

We found that under the condition of uniform distribution of sensing nodes in the sensing area, the communication path between any two points is usually near the shortest distance between the two points. As shown in Figure 4, we can see that the actual communication path between points A and B is almost near the straight line connecting points A and B and the nodes at the bottom left. The upper right parts of the entire sensing range are the least likely to need to perform relay work between points A and B. During transmitting the target request packet from node A, the nodes in these two parts must receive and broadcast the packet. Therefore, we hope that during the target request stage, the transmission range of the target request packet is minimized to the required range, saving unnecessary nodes to conserve energy.

Under the SSR protocol, to shorten the range of broadcast target request packets without the need for sensor node coordinate positioning, our proposed method is to divide the entire network nodes into multiple blocks using hop count, each node will record its own block, and when sending a target request packet, it is limited by the established block range. If the received target request node is within the required range, it will continue broadcasting the packet; otherwise, the packet will be discarded, and the broadcast action will stop. The next section will discuss the method of dividing the network nodes into multiple blocks and limiting the transmission range.

3. Improvement of Flooding Mechanism

Before executing our improved flooding method, we need some environment settings to ensure the accuracy of the results after the flooding. First, we place a data collection node (referred to as x sink and y sink) in each of the two adjacent corners of a square sensing area. Then, we evenly deploy sensing nodes in the sensing area. The sensing nodes need to reach a certain density (in the simulation experiment in Chapter 4, we set the sensing area to be 200 meters * 200 meters, with 500 nodes and a node transmission range of 25 meters). Otherwise, it will affect the integrity of the block establishment and may not be able to achieve the shortest path transmission during data transmission. After setting the required environmental conditions, we can start to execute our improved flooding method, which can be divided into the following steps: (i) establish hop numbers, (ii) draw partition blocks, and (iii) restrict the flooding transmission range [16].

3.1. Establish Hop Numbers

After all the nodes are deployed, the x and y converge points will respectively broadcast jump numbers to create packets, the packets containing converge point ID and jump number fields, with the jump number field set to 1 before broadcasting. Sensing nodes near the converge points to receive the packet; they will check the contents of the packet and record the converge point ID and jump number, then increase the jump number by 1 and broadcast it again. Repeat this process until all nodes in the network receive the jump number packets from the x and y converge points. If a node receives a packet with the same converge point ID and a different jump number, it will compare and update the recorded jump number with the smaller value received if the packet received has a smaller jump number. The node will then broadcast again to find the closest distance to the converging point. Otherwise, the packet will be discarded, and no action will be taken.

3.2. Draw Partition Blocks

After recording two hop numbers at each sensing node, our block establishment is divided based on these two values. Each node will store this information (x_hop, y_hop). Nodes with the same x and y hop values will be classified as the same block. After all the sensing nodes and the x and y collection points are set up in Figure 5, Figures 6(a) and 6(b) establish the x and y collection point jumps. Each node can then know its own block and return its block information to the collection point for later source node searching for the target node block.

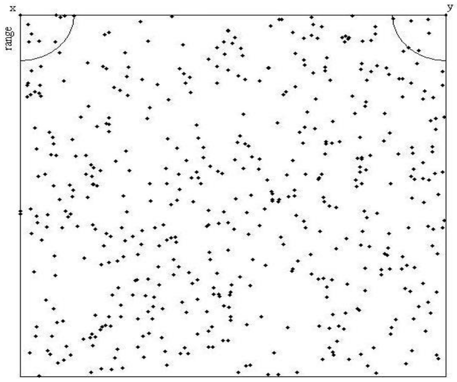


Fig. 5 Deployment of sensors and x, y collection points. Collection points are on the left and right upper corners

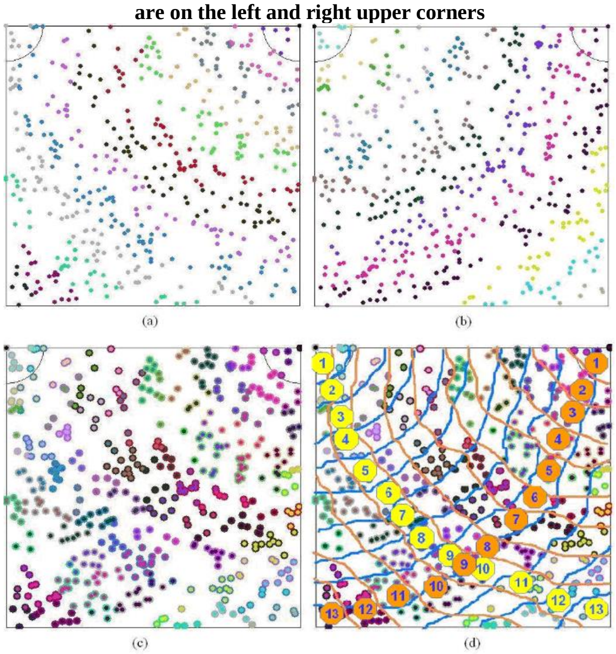


Fig. 6 (a) establishes the hops with the x collection point. (b) establishes the hops with the y collection point. (c) The blocks divided by the two hop numbers are combined and belong to the same block of the same color. (d) Mark the range of the x block and y block.

3.3. Restrict the Flooding Transmission Range

After the block establishment is completed, when a source node wants to find the target node to transmit data, it will first obtain the target node's block position from the collection point and then broadcast the DREQ packet with the block range itself, according to the calculation rule of the range, as in formula (1). x_s and y_s represent the source node's block, and x_d and y_d represent the target node's block.

$$Range_{flooding} = x\_band[\min(x_s, x_d), \max(x_s, x_d)] \cap y\_band[\min(y_s, y_d), \max(y_s, y_d)] \quad (1)$$

Suppose that the hops from collection points x and y to the origin are C_x and C_y, and the band blocks formed by x collection point are x_band[1], x_band[2], …, x_band[C_x], and those formed by y collection point are y_band[1], y_band[2], ..., y_band[C_y]. The successive bands from x_band[n_1] to x_band[n_2] are denoted as x_band [n_1, n_2], where $1 \leqq n_1$, $n_2 <= C_x$. y_band[m_1, m_2] are defined similarly as x_band[n_1, n_2].

The DREQ packet transmission area can extend to the outer edge of the required range at most. Because nodes at the edge receive DREQ packets broadcast by nodes within the broadcast range, after checking, they find that they are not part of the broadcast range and discard the packets, stopping the spread of the broadcast range.

This method can be used in general circumstances. However, the following special circumstances must be considered because the divided blocks are part of the sector when the difference of the x (or y) block between two points is relatively small ($0 \leqq \triangle x \leqq 3$). The difference of the y (or x) block is relatively large ($\triangle$ y>5); the original limited broadcast range will be difficult to include nodes near the line between the two points. It will be difficult to achieve the shortest transmission path, as shown in Figure 8(a). Therefore, in this situation, we must modify the coverage range to obtain the nodes with the shortest path. Our method is to expand n blocks inward in the small difference block and modify formula (1) as follows. The values of ε and δ depend on the sensing area's size and the sensing points' communication range. In this simulation, ε=3 and δ=5.

$$Range_{flooding} = \begin{cases} x\_band[\min(x_s, x_d) - n, \max(x_s, x_d)] \cap & \text{if } \Delta x \le \varepsilon \text{ and } \Delta y \\ \text{y\_band}[\min(y_s, y_d), \max(y_s, y_d)] & > \delta, \text{then } n = \varepsilon - \Delta x \\ & \\ x\_band[\min(x_s, x_d), \max(x_s, x_d)] \cap & \text{if } \Delta y \le \varepsilon \text{ and } \Delta x \\ \text{y\_band}[\min(y_s, y_d) - n, \max(y_s, y_d)] & > \delta, \text{then } n = \varepsilon - \Delta y \end{cases} \quad (2)$$

In Figure 8(a), we assume that the source point and the target point are located in blocks (10, 13) and (10, 1), respectively. After using formula (1), the broadcast range is limited to the intersection of x_band[10] and y_band[1,23], as shown by the red dotted line in the figure. This would result in a longer transmission path and waste energy. In this case, formula (2) must be used to expand the transmission range to cover the shortest path nodes, as shown in Figure 8(b). The broadcast range is modified to x_band[7,10] and band[1,23].

4. Experiment Results and Analysis

In our experiment, we conducted both traditional flooding and partitioned flooding methods on the SSR protocol, comparing the number of hops established between source and target points and the resulting data transmission. Finally, we displayed the transmission path as a graph to observe the results.

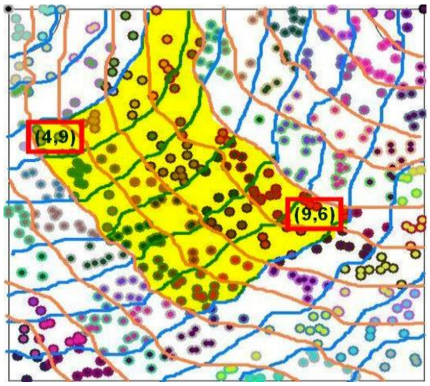
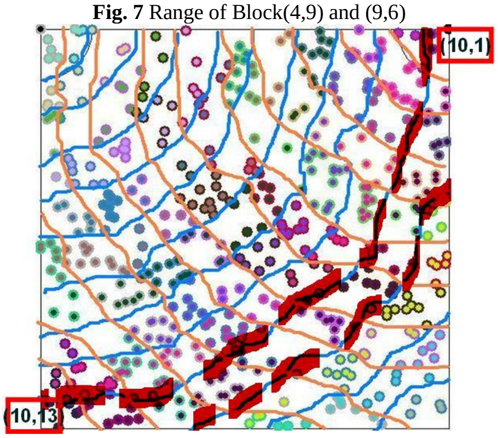
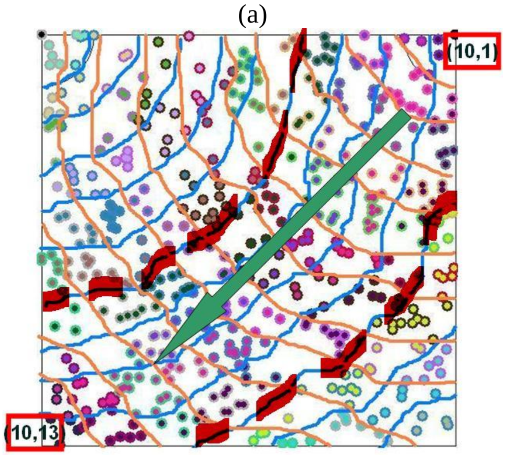


Fig. 7 Range of Block(4,9) and (9,6)



(a)



(b)

Fig. 8 (a) restricts range from the original rule (b) restricts range from the modified rule

Both methods were performed under the same environment parameters and nodes, with two randomly selected points as data transmission points. Additionally, to test the feasibility of our proposed method, we divided the distance between the two points into short, medium, and long distances. Short distances were represented by 60-70 meters, medium distances by 120-130 meters, and long distances by 180-200 meters. We used SENSE (Sensor Network Simulator and Emulator) [20] as our simulation tool, and the next section will introduce SENSE briefly.

4.1. Introduction of SENSE

SENSE is a proven effective sensor network simulation tool; compared to general sensor network simulation tools, it has three main features:

4.1.1. Extensibility

The simulation developed using component-based development has the ability to extend the network simulation architecture. Component-based modules can make the simulation modules extensible: in a compatible situation, a new component can replace an old component and inheritance is not required. The simulation component classes can extend the simulation engine: advanced users can freely develop new simulation engines to meet their needs.

4.1.2. Reusability

Removing dependencies between modules makes them reusable. If the interface and semantics meet the requirements of other users, a specific module development component can be reused. On the other hand, reusability is achieved using C++ templates. A component is typically declared a template class, allowing it to handle different data types.

4.1.3. Scalability

Provides parallel processing. In SENSE, the parallel processing simulation engine can only handle components composed of compatible parts. If the default sequential simulation engine meets the user's needs, then the components in the module library will be reusable.

SENSE is built on COST, which is a library composed of classes. COST is characterized by its use of CompC++, an extension of C++ that is component-oriented, for developing discrete event simulations. COST adopts component-oriented technology, where a discrete event simulation can be seen as a collection of components that interact with each other by exchanging messages through communication ports. Additionally, the simulation also includes a simulation engine responsible for synchronizing the processing of components. Components have one or more event handlers that perform corresponding actions in response to each event's arrival. Events can be divided into two categories: synchronous events, which are transmitted to the component itself through the input port by its neighboring component, and asynchronous events, which are connected to a timer, a special type of port between the component and the simulation engine. Components receive and schedule asynchronous events through the timer.

SENSE v3.03 has been tested and can run on the following operating systems:
Linux (Ubuntu "Breezy Badger", kernel 2.6.12) with GCC version 4.0.2.
Mac OS X v10.4 "Tiger" with GCC version 4.0.1
Windows XP with cygwin (version unknown)
To create a simulation of a wireless sensor network using

SENSE, the following steps are required:
(i) Set up components of a sensor node
(ii) Construct a sensor network drive from CostSimEng
(iii) Build the entire system and run the simulation.

4.2. Analysis of Simulations

Table 1 represents the setting of our sensing environment, source and target points, and the energy consumption of sensing nodes using the MICA2 configuration, as shown in Table 2 [21].

Figure 9 represents each block node's average energy consumption change when returning block data to x sink. After all, nodes have set their blocks and the last block node will start to return its x and y blocks to the sink in a decremental manner for the x block. The nodes that receive the packets will also add their block information to the packets for return. According to Figure 10, the total energy consumption of all nodes (500 nodes) returning information to x sink is 1272.632 Joules, and the energy consumption will approximately double when returning to y sink.

Table 1. Simulation parameters

Parameters	
Area of WSN	200m*200m
Sensor number	500 modes, 2 sinks
Transmission range	25m
Simulation time	20 sec.
Data generation rate	2 packets/second
Distance between two sensors	60-70m, 120-130m, 180-200m

Table 2. Energy Consumptions

Operation Current (mA)	**MICA2**
Processor, full operation	12(7.37MHz)
Processor, sleep	0.010
Radio, receive	7
Radio, transmit	10
Radio, sleep	0.001

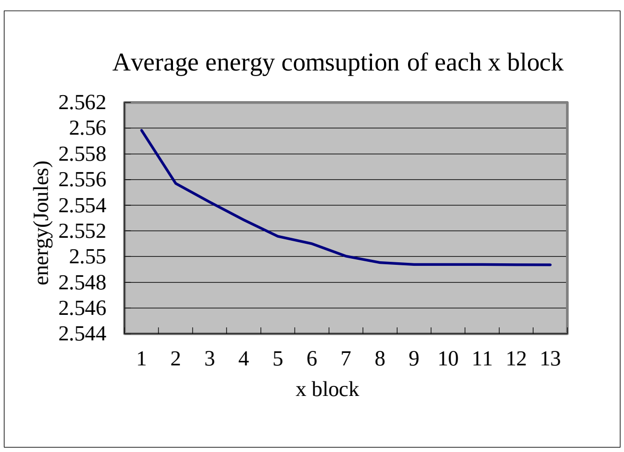


Fig. 9 Average of Energy Consumption

Figure 12 shows the path of the two-point transmission, with the same color representing the transmission process of the same packet. The blocks covered by the broadcast range include nodes near the two points in a straight line, so by establishing the minimum number of hops, the path can achieve the same effect as traditional broadcasting and reduce some unnecessary node hops.
This

We randomly selected ten pairs of points from three different distances as the source and target points for simulation testing and made Figures 12 and 13 with the data. Figure 12 shows the average number of hops of the transmission under each distance and demonstrates the difference in average hops between the two transmissions under different distances. The comparison result shows a very small difference. In Figure 4.7, it can be seen that with the block restriction on the broadcast range, when the distance between the two points is small, the broadcast block also decreases accordingly, saving more nodes from receiving the packet. When the distance is large, the reduction gradually decreases. On the other hand, in the case of unrestricted broadcast range, almost all nodes must receive the target request packet from the source, and only a few nodes may fail to receive the packet message due to collisions during packet transmission.

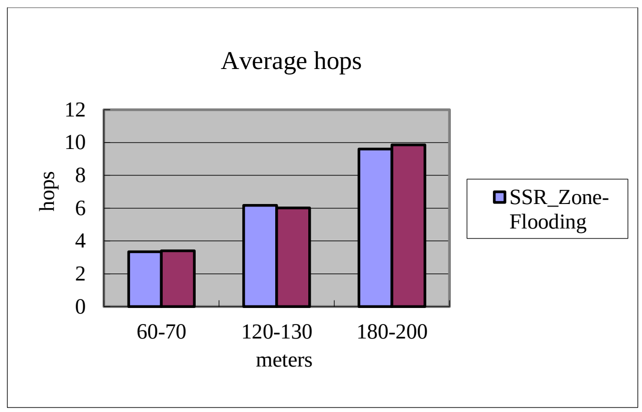


Fig. 10 Average hop numbers

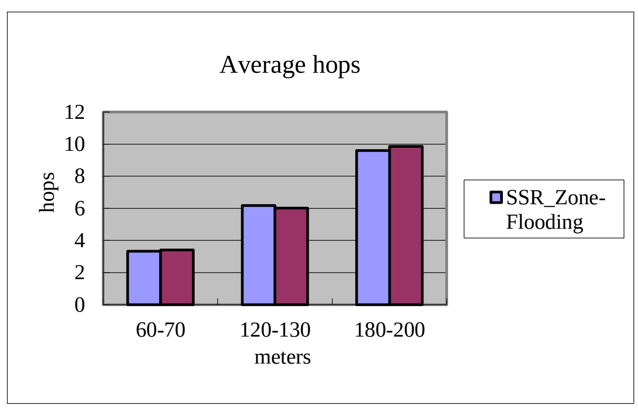
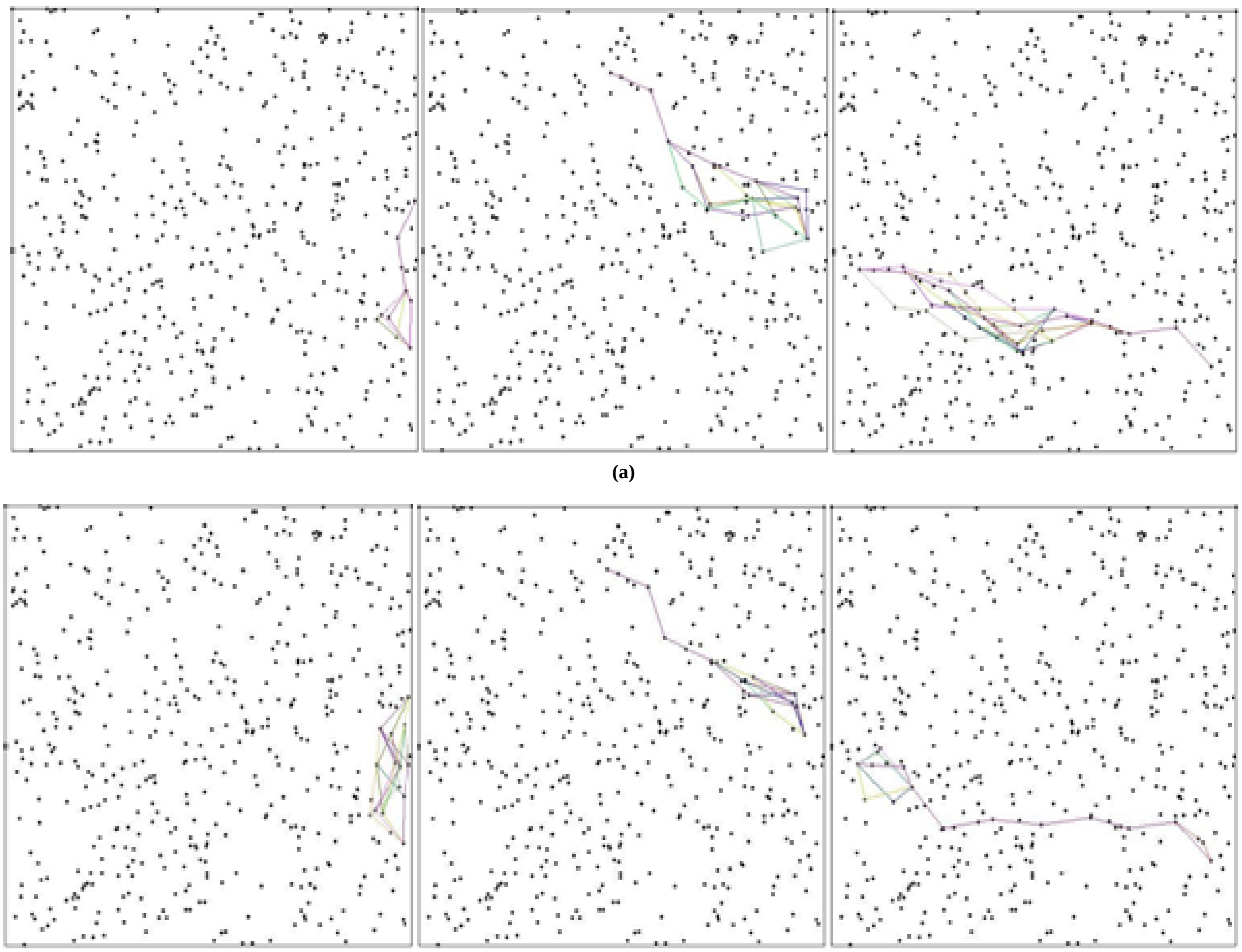


Fig. 11 Average of nodes not receiving any packets

4.3. Improved Simulation under Special Conditions

Next, we test if modifying the block range of broadcasting is necessary under special circumstances, that is, if the block difference between two points is $0 \leqq \triangle x \leqq 3$ in x or y, and the difference between y or x is larger ($\triangle y>5$), then using formula (2) can achieve the shortest path.

Figure 13 shows the source and target points with blocks (10,13) and (10,1), respectively. If the source and target points were to broadcast based on their original positions, the node that can receive the packets would follow a longer path. After modifying this scenario with our formula (2), the resulting transmission path can achieve the same result as the original SSR protocol for the shortest path. Even when the source and target points are located at the farthest distance in the entire sensing area, as shown in Figure 13, the node that needs to receive the target request packet does not have to receive from all the sensing nodes and establish the number of hops, thus saving the nodes in the furthest left-top and right-bottom part, and improving the contribution of the broadcast range area.



(a)

(b)

Fig. 12 (a) depicts the transmission path using traditional broadcast by SSR, from left to right, respectively representing short, medium, and long-distance transmissions. (b) depicts the transmission path using block-constrained broadcast by SSR, from left to right, respectively representing short, medium, and long-distance transmissions.

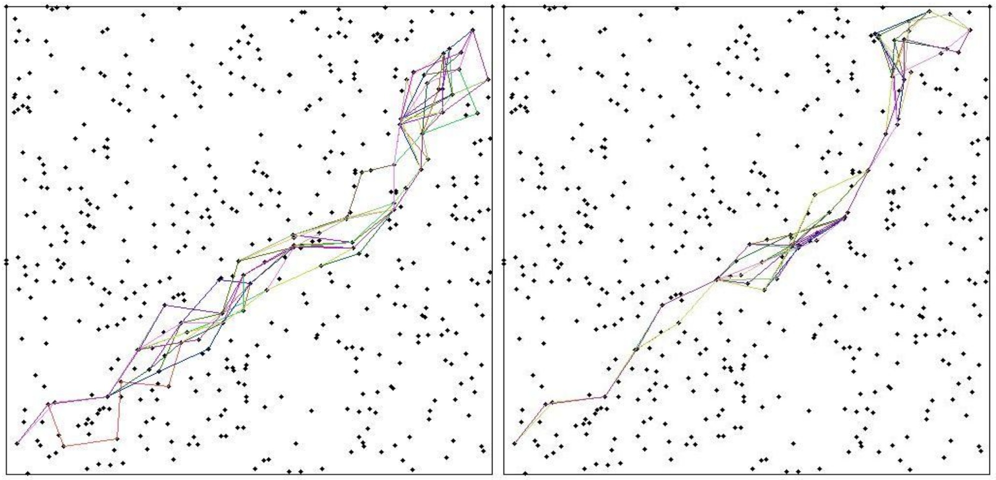


Fig. 13 Shows the transmission using formula (2) in special circumstances, with the left side using the block flooding SSR protocol and the right side using the original SSR protocol.

5. Conclusion

In wireless sensor network routing protocols, broadcast protocols are often used to find and establish paths. However, broadcasts can have negative impacts, such as message collisions, competition, and excess transmission. Therefore, we propose a partitioned broadcast method to reduce the negative effects of broadcasts while maintaining the original transmission path. In the experiment, when there is a certain density of sensing nodes evenly deployed in the sensing area, our method can still establish the shortest transmission path and eliminate the farthest node from receiving and sending messages to save energy.

In the future, we hope to make modifications to the partitioned blocks. After the number of hop broadcast blocks is established, if the source node experiences disconnection with the target node after a period of transmission, we hope to add a fault tolerance mechanism, which is to expand the current broadcast range of the source node to achieve a connection with the target node.

References

[1] B. Warneke et al., "Smart dust: Communicating with a Cubic-millimeter Computer," *Computer*, vol. 34, no. 1, pp. 44-51, 2001. *Crossref*, https://doi.org/10.1109/2.895117

[2] J. M. Kahn, R. H. Katz, and K.S.J. Pister, "Next Century Challenges: Mobile Networking for "Smart Dust"," *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networki*ng, pp. 271-278, 1999. *Crossref*, https://doi.org/10.1145/313451.313558

[3] Eugene Shih et al., "Physical Layer Driven Protocol and Algorithm Design for Energy-efficient Wireless Sensor Networks," *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 272-286, 2001. *Crossref*, https://doi.org/10.1145/381677.381703

[4] Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Heinzelman, "A Taxonomy of Wireless Micro-sensor Network Models," *Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28-36, 2002. *Crossref*, https://doi.org/10.1145/565702.565708

[5] Dionisis Kandris et al., "Applications of Wireless Sensor Networks: An Up-to-Date Survey," *Applied System Innovation*, vol. 3, no. 1, 2020. *Crossref*, https://doi.org/10.3390/asi3010014

[6] I.F. Akyildiz et al., "Wireless Sensor Network: A Survey," *Computer Networks*, vol. 38, no. 4, pp. 393-422, 2002. *Crossref*, http://dx.doi.org/10.1016/S1389-1286(01)00302-4

[7] Hui Cheng, Lei Tao, and Xinming Zhang, "A Fast and Efficient Broadcast Protocol with a Mobile Sink Node in Asynchronous Wireless Sensor Networks," *IEEE Access*, vol. 7, pp. 92813-92824, 2019. *Crossref*, https://doi.org/10.1109/ACCESS.2019.2927627

[8] David B. Johnson, and David A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile computing, pp. 153-181, 1996. *Crossref*, https://doi.org/10.1007/978-0-585-29603-6_5

[9] C. Perkins, E. Belding-Royer, and S. Das, "Ad Hoc On-demand Distance Vector (AODV) Routing (No. rfc3561)," *Browse RFC*, 2003. *Crossref*, https://doi.org/10.17487/RFC3561

[10] Wendi Beth Heinzelman, "*Application-Specific Protocol Architectures for Wireless Networks*," Massachusetts Institute of Technology, 2000.

[11] Z. J. Haas, and M. R. Pearlman , "The Zone Routing Protocol (ZRP) for Ad Hoc Networks", INTERNET DRAFT - Mobile Ad hoc

NETworking (MONET) Working group of the Internet Engineering Task Force (IETF), vol. 9, 1997.

[12] Weixin Huang, and Linghua Zhang, "An Improved Protocol Based on Directed Diffusion Routing Protocol with Network Coding," *Proceedings of the 2019 8th International Conference on Networks, Communication and Computing*, pp. 159-162, 2019. *Crossref*, https://doi.org/10.1145/3375998.3376030

[13] Revathi B, and Arulanandam K, "Design and Development of Robust and Secure Cluster Routing Algorithm for Manet Based IOT," *International Journal of Computer Trends and Technology*, vol. 69, no. 8, pp. 15-19, 2021. *Crossref*, https://doi.org/10.14445/22312803/IJCTT-V69I8P104

[14] Sze-Yao Ni et al., "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, vol. 8, no. 2/3, pp. 151-162, 1999. *Crossref*, http://dx.doi.org/10.1145/313451.313525

[15] Erol Gelenbe et al., "Cognitive and Self-selective Routing for Sensor Networks," *Computational Management Science*, vol. 8, pp. 237-258, 2011. *Crossref*, https://doi.org/10.1007/s10287-009-0102-y

[16] Richa Sharma, Vasudha Vashisht, and Umang Singh, "Modelling and Simulation Frameworks for Wireless Sensor Networks: A Comparative Study," *IET Wireless Sensor Systems*, vol. 10, no. 5, pp. 181-197, 2020. *Crossref*, https://doi.org/10.1049/iet-wss.2020.0046

[17] Prakash Sonwalkar, and Vijay H Kalmani, "Improved ACO Oriented Efficient Cluster Head Selection Mechanism for Energy Aware Routing Scheme in WSN," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 9, no. 8, pp. 49-59, 2022. *Crossref*, https://doi.org/10.14445/23488379/IJEEE-V9I8P106

[18] Chi-Chang Chen, Chi-Yu Chang, and Yan-Nong Li, "Range-free Localization Scheme in Wireless Sensor Networks Based on Bilateration," *International Journal of Distributed Sensor Networks*, vol. 9, no. 1, 2013. *Crossref*, https://doi.org/10.1155/2013/620248

[19] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 174–185, 1999. *Crossref*, https://doi.org/10.1145/313451.313529

[20] SENSE-3.0: Sensor Network Simulator and Emulator. [Online]. Available: http://www.ita.cs.rpi.edu/

[21] Crossbow MPR-MIB Users Manual Revision A, 2007.

[22] Gilbert G. Chen, Joel W. Branch, and Boleslaw K. Szymanski, "A Self-selection Technique for Flooding and Routing in Wireless Ad-hoc Networks," *Journal of Network and Systems Management*, pp.359–380, 2006. *Crossref*, https://doi.org/10.1007/s10922-006-9036-7

[23] E. Woodrow, and W. Heinzelman, "SPIN-IT: A Data Centric Routing Protocol for Image Retrieval in Wireless Networks," *Proceedings. International Conference on Image Processing*, vol. 3, pp. 913-916, 2002. *Crossref*, https://doi.org/10.1109/ICIP.2002.1039121