WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2006; **6**:35–48
Published online 21 April 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/wcm.241

Efficient broadcasting protocols for regular wireless sensor networks

Jang-Ping Sheu,*,† Chih-Shun Hsu and Yen-Jung Chang

Department of Computer Science and Information Engineering, National Central University, Chung-Li 32054, Taiwan

Summary

The wireless sensor network (WSN) has attracted lots of attention recently. Broadcast is a fundamental operation for all kinds of networks and it has not been addressed seriously in the WSN. Therefore, we propose two types of power and time efficient broadcasting protocols, namely one-to-all and all-to-all broadcasting protocols, for five different WSN topologies. Our one-to-all broadcasting protocols conserve power and time by choosing as few relay nodes as possible to scatter packets to the whole network. Besides, collisions are carefully handled such that our one-to-all broadcasting protocols can achieve 100% reachability. By assigning each node a proper channel, our all-to-all broadcasting protocols are collision free and efficient. Numerical evaluation results compare the performance of the five topologies and show that our broadcasting protocols are power and time efficient.

KEY WORDS: broadcast; wireless sensor network (WSN)

1. Introduction

The wireless sensor network (WSN) has attracted lots of attention recently. We can use the WSN to monitor the conditions of a place, where the traditional wired network is not available, such as battlefield, forest, outer space and human body [1]. A WSN usually consists of thousands of sensor nodes. Each sensor node is equipped with micro-electro-mechanical systems (MEMS) component, which includes sensor, radio circuit, data fusion circuitry [2] and general purpose signal processing engines [3]. The wireless sensor node collects the information from the environment by its sensor, processes the information by its signal processing engine and exchanges the information with other sensor nodes by its radio circuit.

It is known that the WSN with regular topology can communicate more efficiently than the WSN with random topology [4]. In a regular WSN, we can get the location of each sensor node without the help of GPS or other positioning devices. Besides, the biomedical sensor network is known to be regular [1]. To reduce the power consumption of communications and to get more accurate locations of sensor nodes, we can adopt the WSN with regular topology in some proper applications, such as deploying WSN to buildings, bridges, space vehicles [5] and biomedical sensors [1]. The regular WSN can be deployed by

*Correspondence to: Jang-Ping Sheu, Department of Computer Science and Information Engineering, National Central University, Chung-Li 32054, Taiwan.
†E-mail: sheujp@csie.ncu.edu.tw

robots. A deployment algorithm for regular WSN is proposed in Reference [6]. It uses an autonomous helicopter to deploy a regular WSN.

Broadcast is a fundamental operation for all kinds of networks. In WSN, we can use the one-to-all broadcasting protocol to give orders, search routes, send queries or notify important messages (e.g. intruders detected by sensors). The all-to-all broadcasting protocol can be used to acquire a new global state among wireless nodes [7], broadcast personalized information and update routing information. For example, if we want to track several objects in the WSN and predict their moving patterns, each senor node can broadcast its gathering information to all the other sensor nodes. Another example is that each sensor node can broadcast its location and sensing range to all sensor nodes. This can be used in location aware monitoring. Due to radio's broadcasting nature, designing a broadcast protocol for the WSN should be cautious to prevent redundant broadcasts and collisions. Besides, the sensor node is a low-cost, small sized and power-limited electronic device [8], such that the broadcasting protocol for WSN should be energy-efficient and should not cause complicate computation.

Many researchers focus on designing energy efficient routing and aggregation protocols [4,9–13] for WSN. SPIN [9] is the first data-centric routing mechanism. Before transmission, each sensor node negotiates with its neighbors, so that it can eliminate redundant data and save energy. A scalable routing protocol for WSN, namely directed diffusion, is proposed in Reference [10]. This paper uses a naming scheme for data, so that it can eliminate unnecessary operations of routing, and thus conserve energy. A cluster-based protocol is proposed in Reference [11], which randomly selects cluster heads to collect information in the network. Since each cluster head has to consume more power to transmit the collected information to the base station, randomly selecting cluster heads will let every node consume about the same amount of power. In protocol [12], each sensor node will decide whether it should transmit the data or not according to the variation of the collecting information and thus conserve more power than the protocol in Reference [11]. Power-efficient gathering in sensor information systems (PEGASIS) [13] also try to improve the work in Reference [11]. Instead of forming clusters, PEGASIS forms several chains in the WSN. The gathered data is transmitted along the chain and only one node in each chain needs to transmit the aggregated data to the base station, and thus save energy. A routing protocol for the wireless access network is proposed in Reference [14]. It can evenly distribute the power consumption of the transmission to every possible relay nodes in the network and thus extends the lifetime of the network. Power efficient routing protocols for five different regular WSN topologies are presented in Reference [4]. A stateless geographic real-time communication protocol for WSNs is proposed in Reference [15]. The real-time communication is achieved by maintaining a uniform packet delivery speed across the WSN. The survey of routing protocols for WSN is proposed in Reference [16]. A linear chain scheme for all-to-all broadcasting and data gathering is proposed in Reference [17]. The information of the whole network is gathered along the linear chain from the beginning to the end of the chain, and then the ending node scatters the gathered information in the reverse direction. To improve the work in Reference [17], a multiple-chain scheme for all-to-all broadcasting is proposed in Reference [18]. It divides the network into regions and generates a linear subchain in each region. The linear-chain scheme is applied to each subchain to gather or scatter information.

The routing and aggregation protocols for WSN have been studied extensively. However, no broadcasting protocol for regular WSNs has been proposed before. Here, we propose two types of power and time efficient broadcasting protocols, namely one-to-all and all-to-all broadcasting protocols, for the five different topologies proposed in References [1,4,19]. Our one-to-all broadcasting protocols not only choose as few nodes as possible to relay the broadcast packets, but also scatter the packets along the shortest path. Besides, our one-to-all broadcasting protocols can achieve 100% reachability by carefully handling collisions. By assigning each host a collision-free channel and scattering the packets along the shortest path, our all-to-all broadcasting protocols are not only collision-free but also can save lots of power and time. Numerical analysis results compare the performances of the five different topologies and show that our broadcasting protocols are power and time efficient. Our broadcasting protocols not only have good performances in the WSNs, but also can be applied to the wireless network which is static and regular, such as the packet radio network or the network consisting of wireless access points [14].

The rest of this paper is organized as follow. Section 2 describes the system environments. Section 3 presents our one-to-all broadcasting protocols for the five different topologies. Section 4 shows our all-to-all broadcasting protocols. Section 5 analyzes our broadcasting protocols and conclusions are made in Section 6.

2. System Environments

To evaluate the power consumption of each sensor node, we adopt the first order radio model proposed in Reference [11]. In this model, the power consumption rate (denoted as E_{elec}) of transmitting/receiving packets is 50 nJ/bit. To avoid the transmitting packet being interfered by the noise in the air, the sender has to consume extra 100 pJ/bit/m^2 (denoted as E_{amp}) to strengthen the transmitting signal so that the receiver can receive the packet correctly. If the sender wants to transmit k bits data to the receiver which is d meters away, the total power consumption is:

$$E_{\text{Tx}}(k,d) = E_{\text{elec}} \times k + E_{\text{amp}} \times k \times d^2 \qquad (1)$$

To receive the packet, the power consumption of the receiver is:

$$E_{\text{Rx}}(k) = E_{\text{elec}} \times k \qquad (2)$$

According to Equations 1 and 2, we can calculate the amount of power consumed by transmitting (or receiving) a packet.

Five different network topologies are considered here: namely 2D mesh with three neighbors (Figure 1), 2D mesh with four neighbors (Figure 2), 2D mesh with six neighbors (Figure 3), 2D mesh with eight neighbors (Figure 4) and 3D mesh with six neighbors

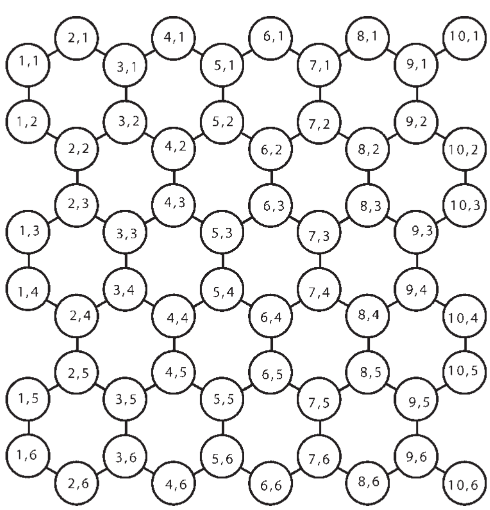

Fig. 1. 2D mesh with three neighbors.

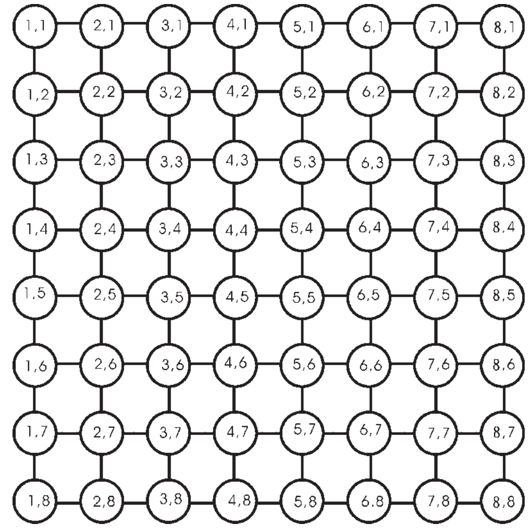

Fig. 2. 2D mesh with four neighbors.

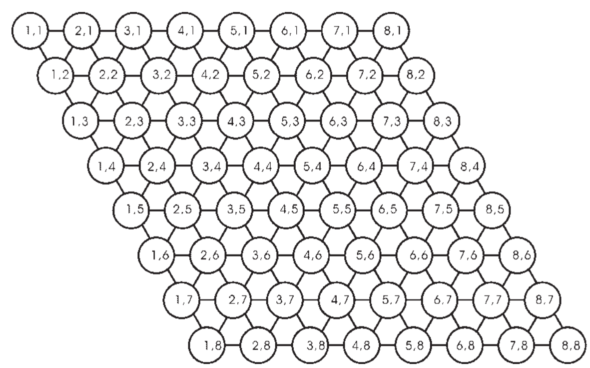

Fig. 3. 2D mesh with six neighbors.

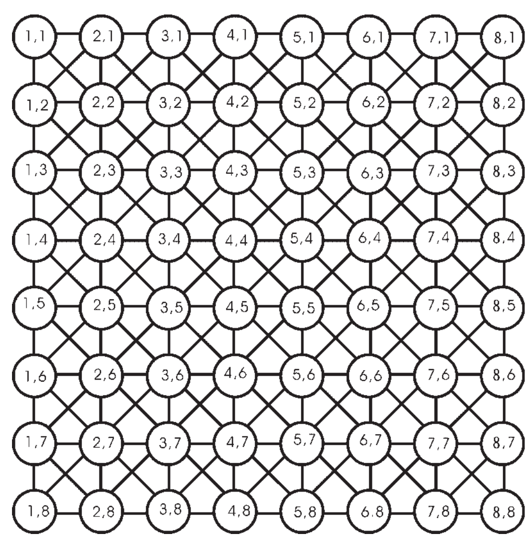

Fig. 4. 2D mesh with eight neighbors.

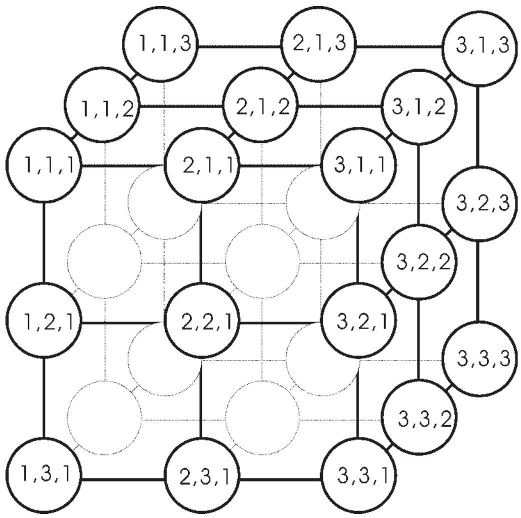

Fig. 5. 3D mesh with six neighbors.

(Figure 5). In the five figures, if there is an edge connecting two nodes, it indicates that the two nodes can communicate with each other by radio and thus they are neighbors. Each node is assigned a unique id according to its relative location in the network. The ids in 2D and 3D networks are denoted as (x, y) and (x, y, z) respectively. In practical, the WSN topology may not be so regular. However, each node can maintain the regular topology by ignoring the signal transmitted by the node that is not supposed to be its neighbor. Maintaining the regular topology will not affect the correctness of our one-to-all and all-to-all broadcasting protocols. For example, in Figure 2, nodes (3, 2), (2, 3), (4, 3) and (3, 4) are neighbors of node (3, 3). Nodes (2, 2), (2, 4), (4, 2) and (4, 4) may also be in the communication range of node (3, 3), but node (3, 3) can maintain the regular topology by ignoring the signal transmitted by these nodes.

We can synchronize the WSN according to the protocols proposed in References [20,21]. The radio channel is assumed to be symmetric, that the power required to transmit a packet from node *A* to node *B* is the same as the power required to transmit a packet from node *B* to node *A*.

3. One-to-All Broadcasting Protocols

The goal of the one-to-all broadcasting protocol is to scatter the source node's data to all the nodes in the network. In traditional broadcasting protocols, almost all the nodes need to forward the data and thus cause severe collisions. To avoid collision, some of the nodes need to wait for a period of time before forwarding the data. However, lots of time and power are wasted when the nodes are waiting. Therefore, we have to reduce the number of relay nodes and handle collisions carefully.

Due to the broadcasting nature of wireless radio (a transmission can cover all the neighboring nodes), it is not necessary for every nodes in the network to forward the broadcast packet while broadcasting packets to every node in the network. Since the network topologies are regular and fixed, we may choose the necessary relay nodes according to the network topology and thus avoid unnecessary forwarding and collisions. To conserve power, the number of relay nodes should be as few as possible, so that the total amount of consumed power can be decreased. Assuming the total number of neighbors is denoted as N and the number of neighbors that receive a non-duplicated packet after the transmission is denoted as M, the efficient transmission ratio (ETR) is defined as ETR $=M/N$. The higher the ETR is, the more efficient the transmission is. Therefore, we will choose the node which has a higher ETR as the relay node. Our goal is to reduce the number of the relay nodes and transmit the broadcast packet along the shortest path so that the delay time and the consumed power can be reduced.

Nodes in different network topology can achieve different ETR. Only the source node in the network can reach 100% ETR. For any node H_i with N neighbors, the possible optimal ETR is $N-1/N$. Assume that H_i receives the broadcast packet from one of its neighbor, H_j. Since H_j has already received the broadcast packet, there are at most $N-1$ nodes that receive the non-duplicated packet after the transmission. For example, in 2D mesh with three neighbors, the non-source node's optimal ETR is 2/3.

Choosing relay nodes according to ETR cannot guarantee a collision-free transmission. Collisions may cause some retransmissions. However, to provide a collision-free broadcast, we need to delay some transmissions, and thus increase the delay time and cause more nodes to receive duplicated packets. The larger the network the longer is the delay time. Besides, receiving duplicated packets will consume more power. Therefore, we do not delay the transmission to avoid collision, instead, we let the collision occur and retransmit the collided packets. Since retransmitting the collided packets will consume additional power, we choose as few nodes as possible to retransmit the packets.

For the ease of describing our broadcasting protocols, we assume that the size of the 2D mesh is $m \times n$, where m and n are positive integers, and the source node's id is (i, j). The nodes in the ath row and bth column of the WSN can form two axes, namely X_a and Y_b axes respectively.

3.1. 2D Mesh with Four Neighbors

To achieve high ETR in 2D mesh with four neighbors, the source node (i, j) first transmits the broadcast packet along the X_j and Y_i axes. As long as the node, whose id is $(i-3k, j)$ (or $(i+3k, j)$), where $1 \leq i-3k < i, i < i+3k \leq m$ and k is a positive integer, has received the broadcast packet, it will transmit the broadcast packet along the Y_{i-3k} (or Y_{i+3k}) axis. However, the nodes in the border of Y axis (e.g. Y_1 and Y_m axes), may still not receive the broadcast packet. Therefore, the nodes in the Y_1 and Y_m axes need to check whether the nodes in the Y_2 and Y_{m-1} axes are relay nodes or not respectively. If the nodes in the Y_2 (or Y_{m-1}) axis are not relay nodes, the nodes in the Y_1 (or Y_m) axis will become the relay nodes, otherwise, they will not become the relay nodes.

When node $(i+1+3k, j)$ is transmitting packets to the right side of the network, it will collide with the transmissions of nodes $(i+3k, j-1)$ and $(i+3k, j+1)$, and collisions occur in nodes $(i+1+3k, j-1)$ and $(i+1+3k, j+1)$ respectively, where $i \leq i+1+3k \leq m$ and k is an integer. Similarly, when node $(i-1-3k, j)$ is transmitting packets to the left side of the network, it will collide with the transmissions of nodes $(i-3k, j-1)$ and $(i-3k, j+1)$, and collisions occur in nodes $(i-1-3k, j-1)$ and $(i-1-3k, j+1)$, respectively, where $1 \leq i-1-3k \leq i$. If we delay the transmissions of nodes $(i+1+3k, j)$ and $(i-1-3k, j)$ to avoid collisions, it will cause d extra time slots delay, where $d = \max(\lceil i-1/3,\ m-i/3 \rceil)$, and nodes $(i+3k, j)$, $(i-3k, j)$, $(i+1+3k, j+1)$, $(i+1+3k, j-1)$, $(i-1-3k, j+1)$ and $(i-1 -3k, j-1)$ will receive duplicated packets. On the other hand, if we delay the transmissions of nodes $(i+3k, j-1)$, $(i+3k, j+1)$, $(i-3k, j-1)$ and $(i-3k, j+1)$ to avoid collisions, it will cause an extra time slot delay and nodes $(i+1+3k, j+1)$, $(i+1+3k, j-1)$, $(i-1-3k, j+1)$, $(i-1-3k, j-1)$, $(i-1+3k, j+1)$, $(i-1+3k, j-1)$, $(i+1-3k, j+1)$ and $(i+1-3k, j-1)$ will receive duplicated packets and thus consume more power. For example, in Figure 6, if we delay the transmissions of nodes (2, 8), (5, 8), (7, 8), (10, 8) and (13, 8), it will cause three extra time slots delay. Therefore, we do not try to avoid collisions, instead we let nodes (2, 8), (5, 8), (7, 8), (10, 8) and (13, 8) retransmit the broadcast packet in the next time slot.

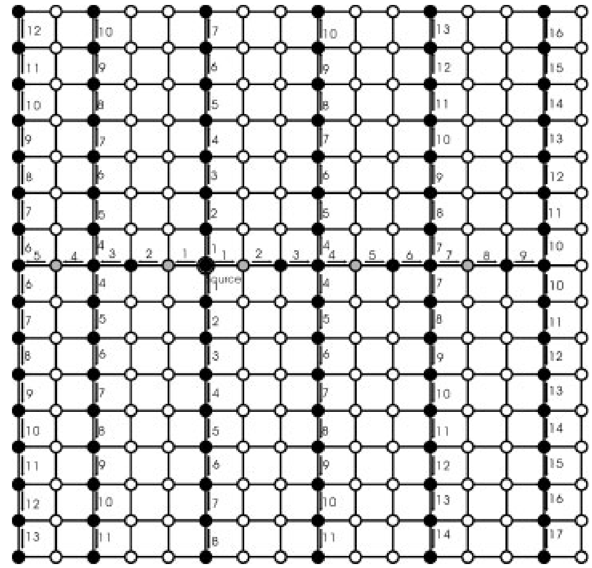

Fig. 6. One-to-all broadcast for 2D mesh with four neighbors, where source is (6, 8).

Figure 6 is an example of the one-to-all broadcast for 2D mesh with four neighbors. The nodes in black or gray color are the relay nodes, the nodes in gray color need to retransmit the broadcast packet, and the numbers beside the edges are the transmission sequences. In this protocol, most of the relay nodes can achieve optimal ETR (=3/4) and thus conserve lots of power.

3.2. 2D Mesh with Six Neighbors

For the ease of describing the broadcasting protocols of 2D mesh with six, eight and three neighbors, we define the term 'diagonal axis' as follows: for any node (i, j), where i is the coordinate in the X axis and j is the coordinate in Y axis, we define two types of diagonal axis, namely S_1 and S_2. The node (i, j) along S_1 axis is in set $S_1(c)$, if $c = i + j$, and the node (i, j) along S_2 axis is in set $S_2(c)$, if $c = i - j$. For example, nodes (5, 7), (6, 6) and (7, 5) are in set $S_1(12)$, and nodes (5, 3), (6, 4) and (7, 5) are in set $S_2(2)$. The nodes in a set will form a straight line in the network. The straight line formed by the nodes in $S_1(c)$ are named as the S_1 direction, and the straight line formed by the nodes in $S_2(c)$ are named as the S_2 direction.

Compare 2D mesh with six neighbors to 2D mesh with four neighbors. In 2D mesh with six neighbors, node (x, y) has two additional neighbors, nodes $(x-1,\ y+1)$ and $(x+1,\ y-1)$. Therefore, when broadcasting packets to the upper right and lower left areas of the source node (i, j), the packets can be forwarded along the nodes in set $S_1(i+j)$. When broadcasting packets to the upper left and lower right areas of the source node, we use the same protocol as in 2D mesh with four neighbors to choose the relay nodes and forward the broadcast packets. However, forwarding the broadcast packets along the relay nodes in set $S_1(i+j)$ cannot cover all the nodes in the upper right and lower left areas of the source node. When the packet is forwarded to the node, whose distance is $3k$ hops away from the source and id is $(i+3k, j-3k)$, where $1 \leq i+3k \leq m$ and $1 \leq j-3k \leq n$, and k is an integer, the packet will also be forwarded along the X_{j-3k} and Y_{i+3k} axes.

The ending nodes of X_j and $S_1(i+j)$ axes, for example nodes $(1, j)$, (m, j), $(1, i+j-1)$ and $(i+j-1,\ 1)$, need to check whether they need to be relay nodes or not. If the nodes in their neighboring rows or columns are relay nodes, they will not become the relay nodes, otherwise, they should become the relay nodes. The rules are summarized as follows:

- If the nodes in Y_2 axis are relay nodes ($(i-2)$ mod $3=0$) then nodes $(1, j)$ and $(1,\ i+j-1)$ will not become the relay nodes else nodes $(1, j)$ and $(1,\ i+j-1)$ will become the relay nodes.
- If the nodes in Y_{m-1} axis are relay nodes ($(m-i-1)$ mod $3=0$) then node (m, j) will not become the relay node else node (m, j) will become the relay node.
- If the nodes in Y_{i+j-2} axis are relay nodes ($(j-2)$ mod $3=0$) then node $(i+j-1,\ 1)$ will not become the relay node else node $(i+j-1,\ 1)$ will become the relay node.

For example, in Figure 7, node (5, 9) is the source node. Consider broadcasting packets to the upper right and lower left areas of node (5, 9), the broadcast packet is first forwarded along the $S_1(14)$ axis. When the broadcast packet is forwarded to nodes (2, 12), (8, 6) and (11, 3), the packet will also be forwarded along the X and Y axes. On the other hand, when broadcasting packets to the upper left and lower right areas of node (5, 9), the broadcast packet is first forwarded along the X_j and Y_i axes. When the broadcast packet is forwarded to nodes (2, 9), (8, 9), (11, 9) and (14, 9), the packet will also be forwarded along the X axis. Nodes (1, 9) and (1, 13) will not become relay nodes, because the nodes in Y_2 axis are chosen as the relay nodes.

When the broadcast packet is forwarded to the upper left and lower right areas, the collided and retransmitting nodes are the same as in 2D mesh with four neighbors. However, the packet cannot be retransmitted in the next time slot, otherwise, collisions occur again. To avoid collision, the packet is retransmitted in two time slots later. When node $(i+1+3k, j-1-3k)$ is forwarding the broadcast packet to the upper right areas of the network, it will collide with the transmissions of nodes $(i+1+3k, j-3k)$ and $(i+3k, j-1-3k)$, where $i \leq i+1+3k \leq m$ and $1 \leq j-1-3k \leq j$. Similarly, when node $(i-1-3k, j+1+3k)$ is forwarding the broadcast packet to the lower left areas of the network, it will collide with the transmissions of nodes $(i-1-3k, j+3k)$ and $(i-3k, j+1+3k)$, where $1 \leq i-1-3k \leq i$ and $j \leq j+1+3k \leq n$. Therefore, nodes $(i+1+3k, j-1-3k)$ and $(i-1-3k, j+1+3k)$ will retransmit the packet, where $i \leq i+1+3k \leq m$, $1 \leq j-1-3k \leq j$, $1 \leq i-1-3k \leq i$ and $j \leq j+1+3k \leq n$. However, if they retransmit the packet in the next time slot, collisions occur again. To avoid collision, they retransmit the packet in two time slots later. According to the above rules, four of the source node's neighbors need to retransmit the packet and collisions occur when they retransmit simultaneously. Therefore, we choose three non-neighboring nodes from the six neighbors of the source node to retransmit the packet in two time slots later and no collision occurs.

Figure 7 is an example of the one-to-all broadcast for 2D mesh with six neighbors. The nodes in black or

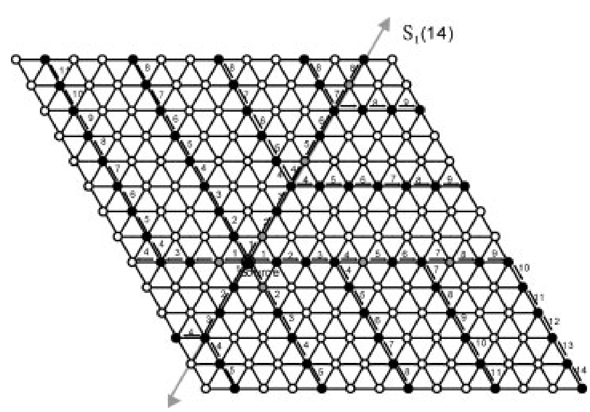

Fig. 7. One-to-all broadcast for 2D mesh with six neighbors, where source is (5, 9).

gray color are the relay nodes, the nodes in gray color need to retransmit the broadcast packet and the numbers beside the edges are the transmission sequences. Node (5, 9) is the source. When collisions occur, three of the neighboring nodes of node (5, 9), nodes (4, 9), (6, 8) and (5, 10), need to retransmit the packet in two time slots later. The nodes (9, 9) and (12, 9) in the X_9 axis and the nodes (9, 5) and (12, 2) in the $S_1(14)$ axis also need to retransmit the packet in two time slots later.

After a broadcast packet is received from the neighbor, three of the receiver's neighbors (including the sender) have already received the broadcast packet. Therefore, in 2D mesh with six neighbors, the optimal ETR is 3/6. Most of the relay nodes in our protocol can achieve the optimal ETR and thus can conserve lots of power.

3.3. 2D Mesh with Eight Neighbors

Compare 2D mesh with eight neighbors to 2D mesh with four neighbors, in 2D mesh with eight neighbors, node (i, j) has four additional neighbors, nodes $(i-1, j-1)$, $(i+1, j-1)$, $(i-1, j+1)$ and $(i+1, j+1)$. Therefore, the broadcast packet can be transmitted along the four additional neighbors. Forwarding the broadcast packet along the diagonals cannot only decrease the delay time but also can conserve more energy than forwarding along the X axis and Y axis. In Figure 8, if node (1, 4) transmits the broadcast packet along the X axis and Y axis, it takes six hops to forward the packet to node (4, 1), however, if the packet is forwarded along the diagonal, it takes only three hops to forward the packet to node (4, 1). Besides, if nodes (2, 3) forwards the broadcast packet to node (3, 2), which is along the diagonal direction, nodes (2, 2) and (3, 3) will also receive the broadcast packet, so the ETR of node (3, 2) is 5/8. However, if the broadcast packet is transmitted from node (2, 2) to node (3, 2), which is along the X axis, nodes (2, 1), (2, 3), (3, 1) and (3, 3) will also receive the broadcast packet, and the ETR of node (3, 2) is 3/8, which is much lower than transmitting along the diagonal direction.

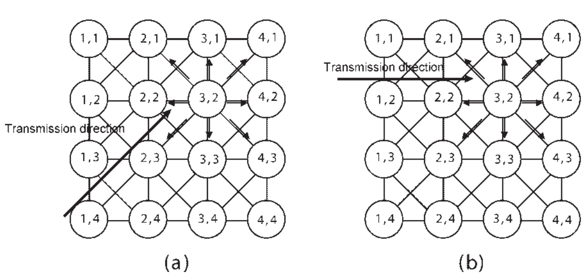

Fig. 8. Transmit packets along the diagonal and the X axis have different efficient transmission ratio (ETR).

Assuming the source node's id is (i, j). We first choose the nodes in sets $S_1(i+j)$ and $S_2(i-j)$ as the basic relay nodes, then we choose the remaining relay nodes from the S_2 (or S_1 but not both) axis. The nodes in sets $S_2(i-j+5k)$, where $-n \leq i-j+5k \leq m$, k is an integer, are chosen as the relay nodes. Collisions occur when the relay nodes those have common neighbors transmit packets simultaneously. However, not all collisions need to be resolved by retransmission. When nodes $(i+1, j+1)$ and $(i+1, j-1)$ transmit packets simultaneously, collisions occur in node $(i+2, j)$, therefore, we let node $(i+1, j-1)$ retransmit the packet. When nodes $(i+3, j-3)$ and $(i+3, j-2)$ transmit packets simultaneously, collisions occur in nodes $(i+4, j-3)$ and $(i+4, j-2)$. However, when nodes $(i+4, j-4)$ and $(i+4, j-1)$ forward the packet, nodes $(i+4, j-3)$ and $(i+4, j-2)$ will receive the packet from them respectively. Therefore, nodes $(i+3, j-3)$ and $(i+3, j-2)$ do not need to retransmit the packet.

For example, in Figure 9, node (5, 9) is the source. Nodes in $S_1(14)$, $S_2(1)$, $S_2(6)$, $S_2(11)$, $S_2(-4)$ and $S_2(-9)$ are chosen as the relay nodes. When nodes (6, 8) and (6, 10) transmit packets simultaneously, collisions occur in node (7, 9). Therefore, we let node (6, 8) retransmit the packet. In case of nodes (8, 6) and

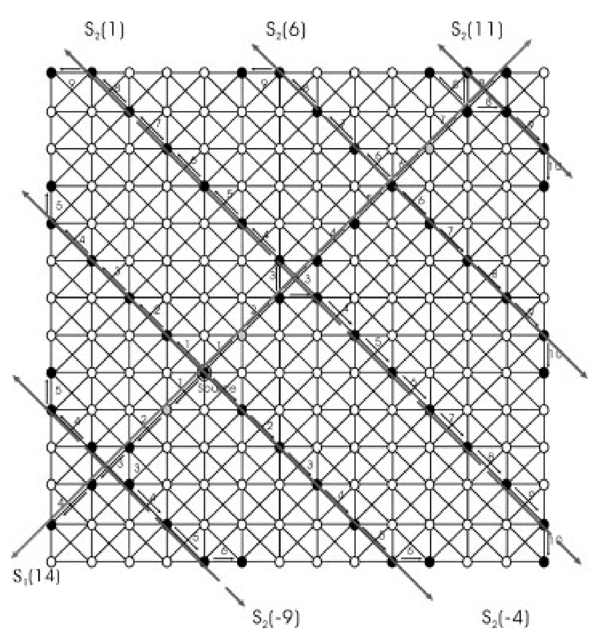

Fig. 9. One-to-all broadcast for 2D mesh with eight neighbors, where source is (5, 9).

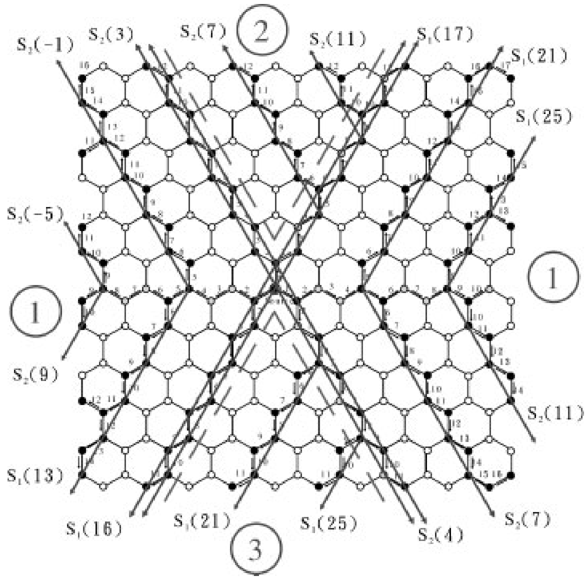

Fig. 10. One-to-all broadcast for 2D mesh with three neighbors, where source is (10, 7).

(8, 7) transmit packets simultaneously, collisions occur in nodes (9, 6) and (9, 7). However, when nodes (9, 5) and (9, 8) forward the packet, nodes (9, 6) and (9, 7) will receive the packet from them respectively. Therefore, neither node (8, 6) nor (8, 7) needs to retransmit the packet. In Figure 9, the nodes in black or gray color are the relay nodes, the nodes in gray color need to retransmit the broadcast packet, and the numbers beside the edge are the transmission sequences. We can see that, among 196 nodes, only three nodes need to retransmit the packet and most of the relay nodes can achieve optimal ETR (=5/8).

3.4. 2D Mesh with Three Neighbors

The broadcasting protocol of 2D mesh with three neighbors is more complicated than that of the other 2D topologies. To choose proper relay nodes and achieve high ETR, we divide the network into three regions as shown in Figure 10. First, the source node (i, j) will choose two nodes (denoted as nodes (i_a, j_a) and (i_b, j_b)) as the base nodes and then decide which region each node is located. If node $(i, j-1)$ is the neighbor of node (i, j), node (i, j) sets $(i_a, j_a) = (i, j-2)$ and $(i_b, j_b) = (i, j+1)$, otherwise, it sets $(i_a, j_a) = (i, j-1)$ and $(i_b, j_b) = (i, j+2)$. For any node (x, y), if $x + y \le i_a + j_a$ and $x - y \ge i_a - j_a$, node (x, y) is in region 2. Otherwise, if $x + y \ge i_b + j_b$ and $x - y \le i_b - j_b$, node (x, y) is in region 3. The node that is not in regions 2 and 3 is in region 1.

Different regions have different rules to choose relay nodes. Basically, we choose the node, which is in the X_j axis or in the two types of diagonal axes (S_1 and S_2), as the relay node. For the convenience of describing our protocol, we assume that the source node's id is (i, j) and the two sets of basic relay nodes along the two diagonal axes is denoted as $B_1(i, j)$ and $B_2(i, j)$. We set $B_1(i, j)$ and $B_2(i, j)$ according to the following rules:

If node $(i, j+1)$ is node (i, j)'s neighbor then $B_1(i, j) = S_1(i+j) \cup S_1(i+j+1)$ and $B_2(i, j) = S_2(i-j) \cup S_2(i-j-1)$
else $B_1(i, j) = S_1(i+j) \cup S_1(i+j-1)$ and $B_2(i, j) = S_2(i-j) \cup S_2(i-j+1)$

For example in Figure 1, node (5, 4) is the source. Since node (5, 5) is not node (5, 4)'s neighbor, we have $B_1(5,4) = S_1(9) \cup S_1(8)$, and $B_2(5,4) = S_2(1) \cup S_2(2)$. The nodes in X_4, $B_1(5, 4)$ and $B_2(5, 4)$ are all chosen as the basic relay nodes.

To broadcast packet to all the nodes in the network, we need to choose more relay nodes according to the following rules. We choose the relay nodes in region 1 according to R1 and R2 and we choose the relay nodes in regions 2 and 3 according to R3 and R4.

For any node (x, y) where $1 \le x \le m$ and $1 \le y \le n$:

R1: Node (x, y) is located in region 1 and in the upper right side or lower left side of node (i, j) and $(x, y) \in B_1(i + 4k, j)$, where $1 \le i + 4k \le m$ and k is an integer.
R2: Node (x, y) is located in region 1 and in the upper left side or lower right side of node (i, j) and $(x, y) \in B_2(i + 4k, j)$, where $1 \le i + 4k \le m$ and k is an integer.
R3: Source node (i, j) is located in the left side of the network, i.e. $1 \le i \le m/2$. (Node (x, y) is in region 3 and $(x, y) \in B_1(i + 4k, j)$) or (node (x, y) is in region 2 and $(x, y) \in B_2(i + 4k, j)$), where $1 \le i + 4k \le m$ and k is an integer.
R4: Source node (i, j) is located in the right side of the network, i.e. $m/2 < i \le m$. (Node (x, y) is in region 3 and $(x, y) \in B_2\ (i + 4k, j)$) or (node (x, y) is in region 2 and $(x, y) \in B_1(i + 4k, j)$), where $1 \le i + 4k \le m$ and k is an integer.

The ending nodes of X_j axis, for example nodes $(1, j)$ and (m, j), need to check whether they need to be relay nodes or not. If node $(1, j)$ does not belong to $B_1(x_1, j)$ and $B_2(x_1, j)$, where $x_1 = i - 4\lfloor i - 1/4 \rfloor$, node (1, j) will not become relay node because all its neighbors are covered by the nodes in $B_1(x_1, j)$ and $B_2(x_1, j)$. Therefore, if $(i-1)$ mod 4 =2 or 3, node (1, j) will not become the relay node, otherwise, it will become the relay node. Similarly, if $(m - i)$ mod 4 =2 or 3, all the neighbors of node (m, j) are covered by the nodes in $B_1(x_2, j)$ and $B_2(x_2, j)$, where $x_2 = i + 4\lfloor m - i/4 \rfloor$, node (m, j) will not become relay node, otherwise, it will become the relay node. For example in Figure 10, the source node's id is (10, 7), which is located on the left side of the network. The nodes in black or gray color are the relay nodes, the nodes in gray color need to retransmit the broadcast packet, and the numbers beside the edge are the transmission sequences. Node (21, 7) will not become relay node, since all its neighbors are covered by the nodes in $B_1(18, 7)$ and $B_2(18, 7)$. According to rule R1, the nodes located in region 1 and in sets $S_1(17)$, $S_1(16)$, $S_1(13)$, $S_1(12)$, $S_1(9)$, $S_1(8)$ $S_1(20)$, $S_1(21)$, $S_1(24)$ and $S_1(25)$ are chosen as the relay nodes. According to rule R2, the nodes located in region 1 and in sets $S_2(3)$, $S_2(4)$, $S_2(0)$, $S_2(-1)$, $S_2(-4)$, $S_2(-5)$, $S_2(7)$, $S_2(8)$, $S_2(11)$ and $S_2(12)$ are chosen as the relay nodes. According to rule R3, the nodes located in region 2 and in sets $S_2(7)$, $S_2(8)$, $S_2(11)$, $S_2(12)$ and the nodes located in region 3 and in sets $S_1(20)$, $S_1(21)$, $S_1(24)$ $S_1(25)$ are chosen as relay nodes. Since, most of the relay nodes can achieve optimal ETR (=2/3), our protocol can conserve lots of power.

When the broadcast packet is transmitted along the relay nodes, some collisions may occur. Since the topology of the network is predetermined, we know where the collision will occur and which node needs to retransmit the packet.

3.5. 3D Mesh with Six Neighbors

In 3D mesh with six neighbors, the optimal ETR is 5/6. The 3D mesh with six neighbors can be regarded as multiple *XY* planes of 2D mesh with four neighbors. This indicates that 3D mesh with six neighbors has an additional transmission direction, the *Z* axis. For each *XY* plane, we can use the broadcasting protocol of 2D mesh with four neighbors to scatter the packet to every node, however, this approach will consume more power and cause more collisions. Therefore, we divide our broadcasting protocol for 3D mesh with six neighbors into two parts. In the first part, we apply the broadcasting protocol of 2D mesh with four neighbors to scatter the packet to all the nodes in the same *XY* plane as the source node (i, j, k). In the second part, we select some nodes in the *XY* plane to forward the broadcast packet to other *XY* planes along *Z* axis. These selected nodes are denoted as *z*-relay nodes. As soon as the *z*-relay nodes have received the broadcast packet, they can forward the packet to other planes along the *Z* axis without waiting for the ending of part 1. Let the source be a *z*-relay node. We can recursively define the *z*-relay node as follows:

R5: Assuming the network size is $m \times n \times l$. If node (x, y, z) is a *z*-relay node then nodes (x, y, w), $(x-2, y-1, w)$, $(x-1, y+2, w)$, $(x+1, y-2, w)$ and $(x+2, y+1, w)$ are *z*-relay nodes, where $1 \leq w \leq l$.

Note that, when all of the source node's neighbors forward the packet simultaneously, collisions occur, therefore, nodes $(i-1, j, k)$, $(i+1, j, k)$, $(i, j, k-1)$ and $(i, j, k+1)$ need to retransmit the packet. However, when they retransmit the packet simultaneously, collisions also occur. Therefore, relay nodes $(i-1, j, k)$ and $(i+1, j, k)$ will retransmit the packet one slot later and *z*-relay nodes $(i, j, k-1)$ and $(i, j, k+1)$ will retransmit the packet two slots later. To avoid the packet collision occurring between the relay nodes and *z*-relay nodes in the *XY* plane with $z = k$, we also need to delay the *z*-relay nodes to forward the packet one slot later.

There are still some nodes in the border of the plane, which will not receive the broadcast packet. Therefore, we need to choose some additional nodes in the border. Figure 11 is an example of scattering the broadcast packet to other *XY* planes in 3D mesh with six neighbors. The nodes in black color are the *z*-relay nodes. The nodes in gray color are the additional relay node in the border. They will wait for two time slots and then forward the packet.

For example, assume that node (6, 8, 4) is the source node of a 3D mesh with six neighbors. The relay nodes in the *XY* plane of the source node are the same as shown in Figure 6. In addition, according to rule R5, nodes (4, 7, 4), (5, 10, 4), (7, 6, 4), (8, 9, 4), ..., are also selected as *z*-relay nodes to forward the packet to other *XY* planes along *Z* axis as shown in Figure 11.

All of the broadcasting protocols mentioned in this section, forward the broadcast packet along the shortest path and most of the relay nodes can achieve the optimal ETR (The optimal ETRs of the four topologies are shown in Table I). Therefore, our broadcasting

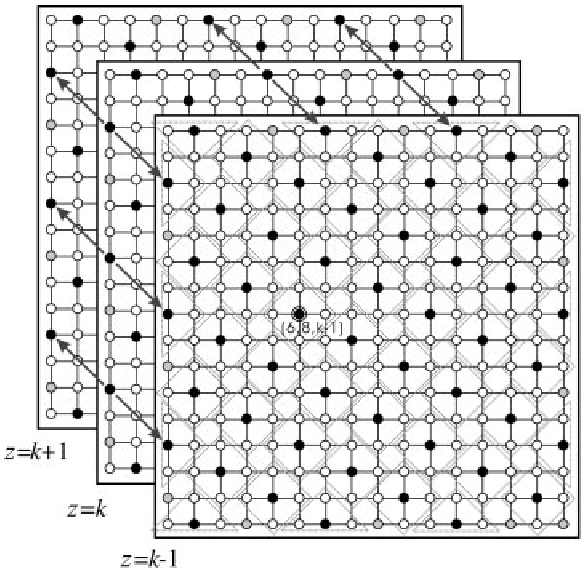

Fig. 11. Scatter the broadcast packet to each *XY* plane along the *Z* axis in 3D mesh with six neighbors, where source is (6, 8, *k*) and black nodes are *z*-relay nodes.

Table I. Optimal ETRs of the five topologies.

Topology	Optimal ETR
2D-3	2/3
2D-4	3/4
2D-6	3/6
2D-8	5/8
3D-6	5/6

protocols cannot only achieve optimal transmission time, but also conserve lots of energy. Besides, collisions are carefully handled such that our broadcasting protocols can achieve 100% reachability.

4. All-to-All Broadcasting Protocols

The purpose of the all-to-all broadcast is to let every node in the network scatter its broadcast packet to all the nodes in the network, hence, the traffic load of the all-to-all broadcast is very heavy. When the traffic load is heavy, collision and contention become a serious problem. Therefore, we divide the nodes into several groups and the nodes in the same group can transmit the packet simultaneously. To avoid collision, neighboring nodes within two hops cannot be assigned to the same group and cannot transmit packets at the same time. The grouping problem is equal to the coloring problem [22]. Our goal is to use the least number of colors to paint the network. The color of the node is the channel that the node is using to transmit packets.

Many researchers [23–26] attempt to design collision-free broadcasting protocols for mobile ad hoc networks, and model the broadcast scheduling problem as a graph-coloring problem. In our protocol, since the network topologies are regular and fixed, we can assign each node a proper color (or channel) according to the network's topology. All the neighboring nodes cannot be assigned to the same color. Hence, the minimum number of the required colors is $N+1$, where N is the number of neighbors. Figures 12–16 show how we assign colors to 2D mesh with three, four, six and eight neighbors and 3D mesh with six neighbors respectively. The detail of the algorithm to color the five topologies are shown in the appendix.

Our all-to-all broadcasting protocol works as follows. First, assign colors to all nodes in the network.

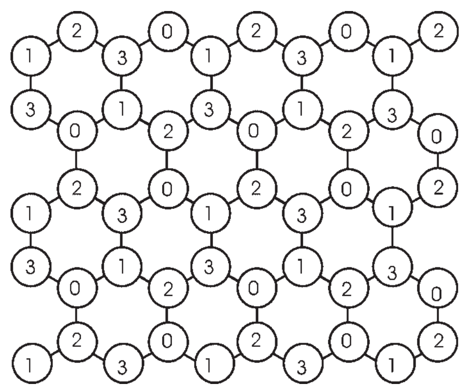

Fig. 12. Coloring a 2D mesh with three neighbors using four colors.

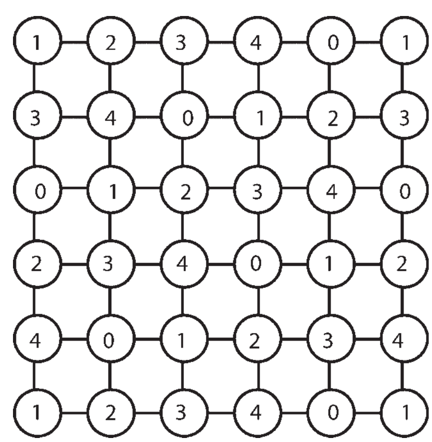

Fig. 13. Coloring a 2D mesh with four neighbors using five colors.

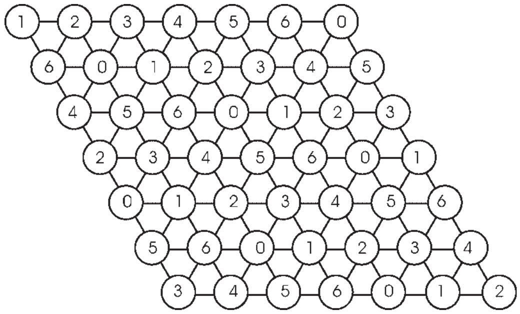

Fig. 14. Coloring a 2D mesh with six neighbors using seven colors.

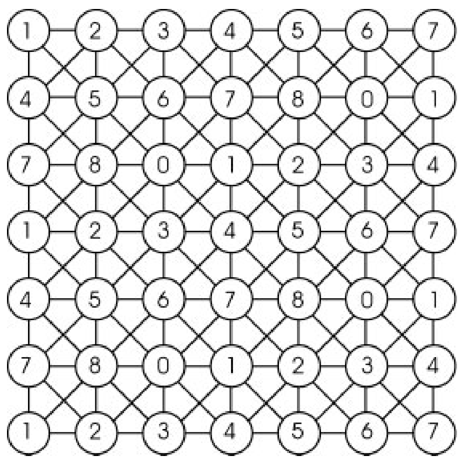

Fig. 15. Coloring a 2D mesh with eight neighbors using nine colors.

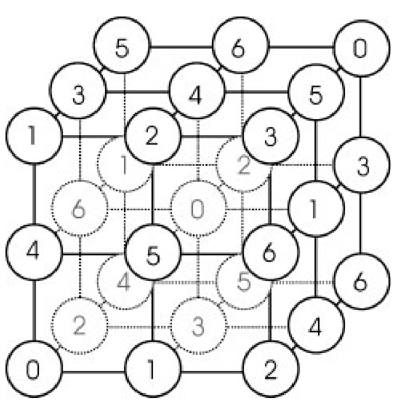

Fig. 16. Coloring a 3D mesh with six neighbors using seven colors.

Second, follow the one-to-all broadcasting protocol to choose relay nodes and forward the packet in the dedicated time slot. When the color of each node is decided, the assigned time slot is also decided. Assuming the assigned color of a node is c and the number of colors is n, the node will transmit the packet at time slot $c + k \times n$, where k is an integer and $k \geq 0$. In other time slot, the node will receive the packets. As the transmission time slot is decided, each relay node follows the one-to-all broadcasting protocols to transmit the packet in its dedicated time slot. For example, node (2, 3) in Figure 13 is assigned to color 1, it will transmit packets at time slot 1, 6, 11, $16, \ldots$, and receives packets in the remaining time slots.

By assigning each node a collision-free time slot, our all-to-all broadcasting protocols are collision-free such that all the relay nodes do not need to retransmit packets and delay the transmissions.

The all-to-all broadcast algorithm is shown as follows:

Algorithm: All-to-All Broadcast
(x, y) is the id of a node, n is the number of colors in a network, c is the assigned color of node (x, y) and S is the size of the network.

```
Begin
  For k := 0 to S − 1
    For l := 0 to n − 1
      If l = c then
        pkt = the first packet in the message_queue
        (x, y) broadcasts pkt to its neighbors
      else
        node (x, y) receives a packet pkt with
        source (i, j) from one of its neighbors.
        Apply the one-to-all broadcasting protocol
        to decide whether to relay the received
        packet or not.
        If node (x, y) has to forward the received
        packet pkt for source (i, j) then
          put the received packet pkt to message_
          queue
        End if
      End if
    End for
  End for
End
```

5. Performance Analysis

In this section, we will calculate and analyze the performance of our broadcasting protocols. We assume that there are 512 nodes on the network. These nodes can be constructed as a 32×16 2D mesh or a $8 \times 8 \times 8$ 3D mesh. The distance between any two neighboring nodes (d) is 0.5 m, the packet length (k) is 512 bits. We use Equations 1 and 2 mentioned in

Section 2 to calculate the consumed power of each transmission. We will calculate the total number of transmissions (T_x), receptions (R_x), power consumption and delay time for each communication. The total number of transmissions is the total times that the packet is transmitted by nodes in each communication. The total number of receptions is the total times that the packet is received by each node in each communication. The power consumption is the total power consumed for transmitting and receiving packets of each node in each communication. The delay time is the time from the source initiating the communication to the time the communication is over. We use the time slot as the time unit.

To show the efficiency of our one-to-all broadcasting protocols, we compare the performance of our protocols with the ideal case. In the ideal case, each relay node can achieve optimal ETR and broadcast packets without any collision. In our one-to-all broadcasting protocols, different source has different total number of transmissions, receptions, power consumption and delay time. If the source is in the center of the network, it performs better. If it is in the corner of the network, it will consume more power and has a longer delay time. Tables II–IV show the performances of the ideal case, the best case and the worst case of our broadcasting protocols. We can see that the total power consumption of our protocols is quite close to that of the ideal case, which indicates that our protocols are power efficient. Among the five different network topologies, the optimal ETR of 3D mesh with six neighbors (=5/6) is the best. However, in the first transmission part, the packet is transmitted along a 2D mesh with four neighbors, besides, more number of neighbors will increase the total number of receptions. Therefore, 3D mesh with six neighbors is not the best topology. The optimal ETR of 2D mesh with four neighbors (=3/4) is the second best but fewer number of neighbors causes fewer number of receptions. Therefore, 2D mesh with four neighbors performs best.

Table V shows the maximum delay time of the ideal case and our one-to-all broadcasting protocols. In our one-to-all broadcasting protocols, we do not delay the transmissions to avoid collisions. Besides, all the broadcast packets are scattered through the shortest path that the maximum delay time of our broadcasting protocols is the same as the ideal case, which indicates that our broadcasting protocols are time efficient. Since the diameter of the 3D mesh with six neighbors is smallest, its maximum delay time is also smallest. The diameter of 2D mesh with eight neighbors is smallest among all the 2D topologies, its maximum delay time is also smallest among all the 2D topologies.

Table VI shows the performance of our all-to-all broadcasting protocols. Our all-to-all broadcasting protocols are based on our one-to-all broadcast protocols and coloring algorithms. Since in our one-to-all broadcast protocols, the 2D mesh with four neighbors performs best, among the all-to-all broadcast protocols, 2D mesh with four neighbors also performs best.

Table II. The performance of the ideal case for the one-to-all broadcasting protocols.

Topology	T_x	R_x	Power consumption (J)
2D-3	255	765	2.61×10^{-2}
2D-4	170	680	2.18×10^{-2}
2D-6	170	1020	3.05×10^{-2}
2D-8	102	816	2.35×10^{-2}
3D-6	124	744	2.22×10^{-2}

Table III. The performance of our one-to-all broadcasting protocols (best case).

Topology	T_x	R_x	Power consumption (J)
2D-3	301	798	2.81×10^{-2}
2D-4	208	714	2.36×10^{-2}
2D-6	207	1037	3.18×10^{-2}
2D-8	143	895	2.66×10^{-2}
3D-6	167	815	2.51×10^{-2}

Table IV. The performance of our one-to-all broadcasting protocols (worst case).

Topology	T_x	R_x	Power consumption (J)
2D-3	308	816	2.88×10^{-2}
2D-4	223	778	2.56×10^{-2}
2D-6	233	1077	3.35×10^{-2}
2D-8	147	924	2.74×10^{-2}
3D-6	187	923	2.84×10^{-2}

Table V. The maximum delay times of the ideal case and our one-to-all broadcasting protocols.

Topology	Ideal case	Our protocols
2D-3	46	46
2D-4	45	45
2D-6	45	45
2D-8	31	31
3D-6	20	20

Table VI. The performance of our all-to-all broadcast protocols.

Topology	T_x (512 bits/s)	R_x (512 bits/s)	Power consumption (J)
2D-3	141 104	405 768	14.0017
2D-4	103 264	389 556	12.6175
2D-6	103 507	572 212	17.2997
2D-8	71 562	500 566	14.6474
3D-6	86 480	439 692	13.4711

6. Conclusions

In this paper, we have proposed two types of power and time efficient broadcasting protocols, namely one-to-all and all-to-all broadcasting protocols, for five different WSN topologies. Since the network topologies are all regular and fixed, we can choose as few relay nodes as possible for our one-to-all broadcasting protocols. Besides, our one-to-all broadcasting protocols can achieve 100% reachability by carefully handling collisions. As for our all-to-all broadcasting protocols, we use the coloring algorithms to assign each node a collision-free channel and use the one-to-all broadcasting protocols to scatter the packets such that our all-to-all broadcast protocols are not only collisionfree, but also are time and energy efficient.

Numerical evaluating results show that, our one-to-all broadcasting protocols are time and energy efficient such that their performances are quite close to the ideal case. Besides, we find that when the number of neighbors increase, the total number of transmissions decrease but the total number of receptions increase. Therefore, the topology that can achieve high ETR and balance the total number of transmissions and receptions performs the best. Experimental results show that the 3D mesh with six neighbors has the smallest maximum delay time in one-to-all broadcasting protocols. 2D mesh with four neighbors possesses the minimum power consumption in one-to-all and all-to-all broadcasting protocols. Our broadcasting protocols can be applied to the WSNs and the infrastructure wireless networks, where each base station (or access point) is fixed and communicates through radio.

References

1. Schwiebert L, Gupta SKS, Weinmann J. Research challenges in wireless networks of biomedical sensors. In *Proceedings 7th ACM/IEEE Mobicom*, 2001; pp. 151–165.
2. Hall D. *Mathematical Techniques in Multisensor Data Fusion*. Artech House: Boston, MA, 1992.
3. Pottie G. Wireless sensor networks. *Information Theory Workshop* 1998; **7**: 139–140.
4. Salhieh A, Weinmann J, Kochhal M, Schwiebert L. Power efficient topologies for wireless sensor networks. In *International Conference on Parallel Processing*, 2001; pp. 156–163.
5. Ulmer C, Yalamanchili S, Alkalai L. Wireless distributed sensor networks for in-situ exploration mars. In *NASA Jet Propulsion Laboratory's Technical Report*, 2001.
6. Corke P, Hrabar S, Peterson R, Rus D, Saripalli S, Sukhatme G. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation*, 2004; **6**(4): 3602–3608.
7. Imielinski T, Badrinath BR. Data management for mobile computing. *ACM SIGMOD Record*, 1993; **22**: 34–39.
8. Chandrakasan A, Amirtharajah R, Chao SH, Goodman J, Konduri G, Kulik J, Rabiner W, Wang A. Design consideration for distributed microsensor systems. In *IEEE Custom Integrated Circuits Conference*, 1999; pp. 279–286.
9. Heinzelman W, Kulik J, Balakrishnan H. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1999.
10. Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 2000.
11. Heinzelman WR, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the Hawaii International Conference on System Sciences*, 2000; pp. 3005–3014.
12. Manjeshwar A, Agrawal DP. Teen: a routing protocol for enhanced efficiency in wireless sensor networks. In *Procees-ings of International Parallel and Distributed Processing Symposium*, 2001; pp. 2000–2015.
13. Lindsey S, Raghavendra CS. 'Pegasis: power efficient gathering in sensor information systems. In *Proceedings of the IEEE Aerospace Conference*, 2002.
14. Hsiao PH, Hwang A, Kung HT, Vlah D. Load-balancing routing for wireless access networks. In *Proceedings of IEEE INFOCOM*, 2001; pp. 986–995.
15. He T, Stankovic JA, Lu C, Abdelzaher T. Speed: a stateless protocol for real-time communication in sensor networks. In *Proceedings of the 23th International Conference on Distributed Computing Systems*, 2003.
16. Akkaya K, Younis M. A survey on routing protocols for wireless sensor networks. *Journal of Ad Hoc Networks*. [To appear].
17. Lindsey S, Raghavendra CS. Energy efficient broadcasting for situation awareness in ad hoc networks. In *Proceedings of the International Conference on Parallel Processing*, 2001; pp. 149–155.
18. Du K, Wu J, Zhou D. Chain-based protocols for data broadcasting and gathering in the sensor networks. In *Proceedings of the International Parallel and Distributed Processing Symposium*, 2003.
19. Bhuvaneswaran RS, Bordim JL, Cuiu J, Nakano K. Fundamental protocols for wireless sensor networks. In *Proceedings of International Parallel and Distributed Processing Symposium*, 2001; pp. 1369–1376.
20. Elson J, Girod L, Estrin D. Fine-grain network synchronization using reference broadcasts. In *Proceedings of the Symposium on Operating System Design and Implementation*, 2002.
21. Ganeriwal S, Kumar R, Srivastava MB. Timing-sync protocol for sensor networks. In *ACM SenSys*, 2003.
22. Harary F. *Graph Theory*. Addison-Wesley Publishing Corporation: Reading, MA, 1972.
23. Basagni S, Bruschi D, Chlamtac I. A mobility-transparent deterministic broadcast mechanism for ad hoc networks. *IEEE/ACM Transactions on Networking* 1999; **7**: 799–807.

24. Ephremides A, Truong TV. Scheduling broadcasts in multihop radio networks. *IEEE Transactions on Communications* 1990; **38**: 456–460.
25. Ramanathan S, Lloyd EL. Scheduling algorithms for multihop radio networks. *IEEE/ACM Transactions on Networking* 1993; **1**: 166–177.
26. Ramaswami R, Parhi KK. Distributed scheduling of broadcasts in a radio network. *INFOCOM* 1989; **2**: 497–504.

Appendix

Algorithm: Assign color for 2D mesh with three neighbors
Input: (i, j): identity of the node
Output: the color for node (i, j)
Begin

$T1 = i \bmod 4$
$color = (T1 + (j-1) \times 2) \bmod 4$

End

Algorithm: Assign color for 2D mesh with four neighbors
Input: (i, j): identity of the node
Output: the color for node (i, j)
Begin

$T1 = i \bmod 5$
$color = (T1 + (j-1) \times 2) \bmod 5$

End

Algorithm: Assign color for 2D mesh with six neighbors
Input: (i, j): identity of the node
Output: the color for node (i, j)
Begin

$T1 = i \bmod 7$
$color = (T1 + (j-1) \times 5) \bmod 7$

End

Algorithm: Assign color for 2D mesh with eight neighbors
Input: (i, j): identity of the node
Output: the color for node (i, j)
Begin

$T1 = i \bmod 9$
$color = (T1 + (j-1) \times 3) \bmod 9$

End

Algorithm: Assign color for 3D mesh with six neighbors
Input: (i, j, k): identity of the node
Output: the color for node (i, j, k)
Begin

$T1 = i \bmod 7$
$T2 = (T1 + (j-1) \times 3) \bmod 7$
$color = (T2 + (k-1) \times 2) \bmod 7$

End

Authors' Biographies

Jang-Ping Sheu received his B.S. degree in computer science from Tamkang University, Taiwan, in 1981, and the M.S. degree and Ph.D. in computer science from National Tsing Hua University, Taiwan, in 1983 and 1987 respectively. He joined the faculty of the Department of Electrical Engineering, National Central University, Taiwan, as an associate professor in 1987. He is currently a professor at the Department of Computer Science and Information Engineering, National Central University. He was a chair of the Department of Computer Science and Information Engineering, National Central University from August 1997 to July 1999. He was a visiting professor at the Department of Electrical and Computer Engineering, University of California, Irvine from July 1999 to April 2000. His current research interests include wireless communications, mobile computing, parallel processing and distributed computing systems. He was an associate editor of *Journal of the Chinese Institute of Electrical Engineering*, from August 1996 to July 2000. He was an associate editor of *Journal of Information Science and Engineering* from August 1996 to July 2002. He is an associate editor of *Journal of the Chinese Institute of Engineers*. He was a guest editor of special issue for *Wireless Communications and Mobile Computing Journal*. He was a program chair of IEEE ICPADS'2002. He was a vice-program chair of ICPP 2003. He received the Distinguished Research Awards of the National Science Council of China in 1993–1994, 1995–1996 and 1997–1998. He was the specially granted researchers, National Science Council, from 1999 to 2002. He received the Distinguished Engineering Professor Award of the Chinese Institute of Engineers in 2003. Dr. Sheu is a senior member of the IEEE, a member of the ACM and Phi Tau Phi Society.

Chih-Shun Hsu received his B.S. degree in computer education from National Taiwan Normal University, Taiwan, in 1990, and the M.S. degree in computer science from National Taiwan University, Taiwan, in 1992. He obtained his Ph.D. in computer science from National Central University, Taiwan, in 2004. He joined the faculty of the Department of Information Management, Nanya Institute of Technology, Taiwan, as an instructor in 1996, and has become an associate professor since August 2004. His current research interests include wireless communications and mobile computing.

Yen-Jung Chang received his B.S. degree in computer science from Soochow University, Taiwan, in 2000, and the M.S. degree in computer science from National Central University, Taiwan, in 2002. He is currently a wireless device driver engineer of Delta Networks Incorporation.