Research Article

Borel Cayley Graph-Based Topology Control for Consensus Protocol in Wireless Sensor Networks

Junghun Ryu,¹ Jaewook Yu,² Eric Noel,³ and K. Wendy Tang¹

¹ Department of Electrical and Computer Engineering, Stony Brook University—SUNY, Stony Brook, NY 11794-2350, USA

² Intel Labs, 2111 Northeast 25th Avenue, Hillsboro, OR 97124, USA

³ AT&T Laboratories, 200s Laurel Avenue, Middletown, NJ 07747, USA

Correspondence should be addressed to Junghun Ryu; junghun.ryu@stonybrook.edu

Received 15 June 2013; Accepted 11 July 2013

Academic Editors: C.-Y. Chow and S. Srinivasan

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

Borel Cayley graphs have been shown to be an efficient candidate topology in interconnection networks due to their small diameter, short path length, and low degree. In this paper, we propose topology control algorithms based on Borel Cayley graphs. In particular, we propose two methods to assign node IDs of Borel Cayley graphs as logical topologies in wireless sensor networks. The first one aims at minimizing communication distance between nodes, while the entire graph is imposed as a logical topology; while the second one aims at maximizing the number of edges of the graph to be used, while the network nodes are constrained with a finite radio transmission range. In the latter case, due to the finite transmission range, the resultant topology is an “incomplete” version of the original BCG. In both cases, we apply our algorithms in consensus protocol and compare its performance with that of the random node ID assignment and other existing topology control algorithms. Our simulation indicates that the proposed ID assignments have better performance when consensus protocols are used as a benchmark application.

1. Introduction

An adhoc wireless sensor network (WSN) is a self-organized and distributed network consisting of a large number of small and light sensor nodes [1, 2]. A sensor node includes a processor, a wireless radio, and various sensors to monitor and sense environmental parameters such as temperature, moisture, and pressure. In a WSN, sensor nodes interchange information and collaborate with each other to achieve a common mission. The flexibility, fault tolerance, high sensing fidelity, low cost, and rapid deployment characteristics of sensor networks create many new and exciting application areas for remote sensing [3]. Examples of ad-hoc wireless sensor networks applications include building monitoring [4], environmental sensing [5–7], traffic monitoring [8], and surveillance [9].

Some WSN applications require very dense networks. Hundreds to several thousands of nodes can be deployed throughout a sensor field. For example, some machine

diagnosis applications use up to 3000 nodes in a 100 m by 100 m area [10] or sensors can be deployed within tens of feet of each other for object tracking [11]. The topology of a large and dense sensor network is important to its performance. For example, topology control algorithms are essential in reducing energy consumption and radio interference [12], thus expanding the network's lifetime.

According to [13], topology control algorithms can be classified as location-based, direction-based, or neighbor-based approaches. In all these approaches, a network topology is formed in an adhoc manner based on sensor nodes' location, direction, or some sort of neighbor ordering. However, since these approaches do not generate a predefined graph, there is no guarantee on the topology's overall properties such as a bounded diameter and average path length.

We approach topology control with predefined graphs. Structured graphs have been studied for a long time and are good candidates for interconnection networks [14, 15]. Various graph-based interconnection networks have been

applied to wavelength division multiplexed optical networks [15], satellite constellations [16], and chip design [17]. In peer-to-peer overlay network schemes, various structured graphs are investigated and compared to unstructured P2P overlay network [18]. Examples of P2P overlay networks include k ring lattices with Chords [19], de Bruijn graphs with Koorde, and distance halving [20, 21]. There is theoretic analysis to apply de Bruijn and Cayley graphs to P2P [22, 23]. Graph-based wireless sensor networks have also been explored by other researchers [24–28]. In general, a predefined graph topology with its deterministic connection rule facilitates performance analysis. In addition, some offer symmetry, hierarchy, and hamiltonicity, and can have a constant low degree and existing distributed routing protocol, all preferable properties for WSNs.

We focus on Borel Cayley graphs, a member of Cayley graph family [29]. In [30], it was shown that Borel Cayley graphs have potential to be efficient logical topologies for dense WSNs because of their small degree and short diameter. Also, BCGs are symmetric graphs, a property that enables distributed routing [31]. One of BCG applications is the consensus protocol, a distributed node-to-node message exchange rule to drive nodes to an agreement for a quantity of interest [32]. BCGs showed good performance when compared to mesh, torus, and small world networks [33].

In this paper, we propose two ID assignment algorithms for applying BCGs as logical topologies in WSNs. The first one, the chordal ring based node ID assignment algorithm, uses a specific representation (the chordal ring representation) of BCG to assign node IDs of the entire graph. Its goal is to minimize long distance communications. In the second algorithm, the distributed node ID swapping assignment, we consider the realistic constraint of network nodes' finite transmission constraint, and thus the imposition of the entire BCG is not always possible.

In the former case, when the goal is to minimize communication distance, the proposed chordal ring-based node ID assignment algorithm requires a smaller radio range (57% that of the random ID assignment). In the latter case, when the nodes are subject to a finite transmission range and the imposition of the entire BCGs is not feasible, the proposed distributed node ID swapping assignment imposes more edges of the original graph (43% more edges in comparison to random ID assignment).

It must be noted that a shorter paper describing an initial version of Section 5 was published in [34]. The main differences between two papers are the addition in this paper of the experiment to evaluate a consensus protocol comparing with known topology control algorithms.

This paper is organized as follows. Section 2 reviews the basic concept of BCGs and consensus protocol. Section 3 presents the goals of this paper. Section 4 describes our proposed algorithm to reduce the average communication distance. Section 5 presents our proposed algorithm to maximize connections of BCGs under transmission range constraints. Section 6 shows our topology control's performance in a consensus protocol. Conclusions are presented in Section 7.

2. Preliminaries

In the following, we provide a definition of Cayley graphs, Borel subgroup, and Borel Cayley graphs.

Definition 1 (Cayley graph [29]). A graph $C = C(V, G)$ is a Cayley graph with vertex set V if two nodes $v_1, v_2 \in V$ are adjacent $\Leftrightarrow v_1 = v_2 * g$ for some $g \in G$, where $(V, *)$ is a finite group and $G \subset V \setminus \{I\}$. G is called the generator set of the graph and I is the identity element of the finite group $(V, *)$.

The definition of a Cayley graph requires vertices to be elements of a group but does not specify a particular group.

Definition 2 (Borel subgroup). If V is a Borel subgroup of general linear 2×2 matrices set, then

$$V = \left\{ \begin{pmatrix} x & y \\ 0 & 1 \end{pmatrix} : x = a^t \pmod{p}, y \in Z_p, t \in Z_k \right\}, \quad (1)$$

where a fixed parameter $a \in Z_p \setminus \{0, 1\}$, p is prime, and k is the order of a . That is, k is the smallest positive integer such that $a^k = 1 \pmod{p}$.

Definition 3 (Borel Cayley graph (BCG) [29]). Let V be a Borel subgroup, and let G be a generator set such that $G \subseteq V \setminus \{I\}$; then $B = \mathcal{B}(V, G)$ is a Borel Cayley graph with vertices for the 2×2 matrix elements of V and with directed edge from v to u if $u = v * g$, where $u \neq v \in V$, $g \in G$, and $*$ is a modulo- p multiplication chosen as a group operation.

Definition 4 (GCR [29]). A graph R is a generalized chordal ring (GCR) if nodes of R can be labeled with integers modulo the number of nodes N , and there is a divisor q of N such that node i is connected to node j if and only if node $i + q \pmod{N}$ is connected to node $j + q \pmod{N}$.

The connection rules of elements are defined by connection constants. Connection constants of i and $i + q$ are the same according to the definition. There are four connection constants when the graph is four regular. For example, Figure 1(a) shows a degree 4 GCR with 21 nodes and $q = 3$ classes. The connection rules can be defined as follows: let $V = 0, 1, 2, \dots, 20$. For any $i \in V$, if $i \pmod{3} =$

- “0” : i is connected to $i + 3, i - 3, i + 4, i - 10 \pmod{21}$,
- “1” : i is connected to $i + 6, i - 6, i + 7, i - 4 \pmod{21}$,
- “2” : i is connected to $i + 9, i - 9, i - 7, i + 10 \pmod{21}$. (2)

The connection constants for class 0 are +3, -3, +4, and -10.

A chordal ring (CR) is a special case of GCR, in which every node has +1 and -1 modulo N connections. In other words, all nodes on the circumference of the ring are connected to form a Hamiltonian cycle. A Hamiltonian cycle is a graph cycle through a graph that visits each node exactly once.

Proposition 5. For any finite Cayley graph with vertex set V and any $T \in V$ such that $T^m = I$, there exists a GCR representation of C with divisor $q = N/m$, where I is the identity element.

Proposition 6. All degree-4 Borel Cayley graphs have CR representations.

The proofs of these propositions are given in [29] and not repeated here. T is referred as the transform element. a_i are class representing elements.

For the simplicity of GCR representation, we chose T and a_i for simple representation as follows [29]:

$$T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad a_i = \begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix}^i. \quad (3)$$

Any vertex $v \in V$ is represented with T and a_i as follows [29]:

$$v = T^j * a_i = \begin{pmatrix} 1 & j \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a^i & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a^i & j \\ 0 & 1 \end{pmatrix}. \quad (4)$$

BCGs are defined over a group of matrices. The systematic representation of BCGs from the group domain to the integer domain is useful for routing because nodes are defined in the integer domain and there is a systematic description of connections. The node ID representation in GCR ($ID_g(v)$) is denoted as follows [29]:

$$ID_g(v) = q * j + i, \quad (5)$$

where q is k .

Symmetry or vertex-transitivity is a preferable attribute for an efficient interconnection network topology. Informally, a symmetric or vertex-transitive graph looks the same from any node. This property allows to use an identical routing table at every node. Mathematically, this implies that for any two nodes a and b in the graph, there exists an automorphism of the graph that maps a to b . This property is very useful for practical implementation of interconnection networks. Most of the well-known interconnection graphs, such as the toroidal mesh, hypercube, and cube-connected cycle, exhibit such property.

In Table 1, we list the parameter values for BCGs used in this paper. Parameters p and a determine N and BCG parameter k . Parameters t_1 and y_1 were used to construct the first generator. Parameters t_2 and y_2 were used to construct the second generator. Using two different generators and their inverse, we construct undirected BCGs. We arbitrarily chose parameters t_s and y_s for generators.

3. Problem Statement

A WSN can be represented as a graph, where each node of the graph corresponds to a sensor node and each edge represents a radio connection between nodes. Obviously, for dense WSNs, the number of physical neighbors (nodes within radio range) is large. For ease of description, we call the topology representing the physical neighbors a *host* graph. Besides representing the number of neighbors within communication

TABLE 1: Parameters of Borel Cayley Graphs.

N	p	k	a	t_1	t_2	y_1	y_2
272	17	16	3	1	2	1	1
506	23	22	7	1	7	1	1
1081	47	23	2	1	7	1	1
1474	67	22	3	1	7	1	1
2265	151	15	2	1	7	1	1

reach of each other, an edge of a host graph is also weighted by the communication distance or cost between nodes.

We are interested in the problem of imposing a BCG with a small degree on a dense WSN. We call the BCG topology the *target* graph, and the host graph after the imposition, the *resultant* graph. We consider the two following cases:

- (i) the radio range of each node covers the whole sensor deployment area. In this case, any ID assignment always produces a resultant graph with a fully connected BCG topology. But depending on the ID assignment, the communication distances between neighbors vary. So, in this case, the goal of this ID assignment is to reduce the communication distance between connected nodes;
- (ii) the radio range does not cover the whole sensor deployment area. A careful node ID assignment is required to produce a resultant graph as similar as possible to the target graph (a BCG topology). In this case, the goal is to find the ID assignment that establishes the most communication edges following BCG connection rules.

3.1. Minimize the Average Communication Distance When the Radio Range Covers the Deployment Area. Our node ID assignment problem can now be described in terms of finding the assignment that yields the minimum sum of weights of the resultant graph after imposing a target graph onto the host graph. Figure 2 illustrates these terminologies. Figure 2(a) is the host graph with each node's physical neighbors and weights on the edges that represent the connections communication cost. Figure 2(b) is the target graph. Figures 2(c) and 2(d) show two example resultant graphs, where each graph has a different sum of weights, depending on how node IDs were assigned.

Let a host graph $G = (V, E)$ in the Euclidean space represent the underlying network before applying our proposed methods, with V being the set of sensor nodes and E representing the set of communication distances between nodes. We assume that the radio range of each node in the host graph is large enough to cover the whole deployment area. So the host graph is a fully connected graph. The weight of $E(u, v)$, denoted by w_{uv} , represents the Euclidean distance between nodes u and v . The weight between two nodes u and v with coordinates (u_x, u_y) and (v_x, v_y) is computed as

$$w_{uv} = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2}. \quad (6)$$

FIGURE 1: A Borel Cayley graph in GCR and CR representations with $p = 7, a = 2, k = 3, t_1 = 0, t_2 = 1, y_1 = 1$, and $y_2 = 1$.

FIGURE 2: Graph representation with radio range enough to cover the whole deployment area.

3.2. Completing Communication Edges with Finite Radio Range. This node ID assignment problem can be described as finding the maximum number of edges of the resultant graph after imposing a target graph onto the host graph. Figure 3 illustrates these terminologies. Figure 3(a) is the host graph that shows the physical neighbors of each node. Figure 3(b) is the target graph. Figures 3(c) and 3(d) show two different ways to impose the target graph on the host graph, where each results in a different number of edges, depending on how node IDs were assigned.

The edge $E(u, v)$ is determined by the radio range and the Euclidean distance between nodes u and v . The distance between two nodes u and v with coordinates (u_x, u_y) and (v_x, v_y) is w_{uv} . The edge $E(u, v)$ is defined as follows:

$$E(u, v) = \begin{cases} 1 & \text{if } w_{u,v} \leq \text{radio range} \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

In the remaining sections, we will propose algorithms to solve these two problems: (a) reducing the average of communication ranges of BCG-based networks and (b) completing as many as possible communication edges that follow BCG connection rules. Each problem is dealt with a distributed method. Regardless of the application type, a distributed method is preferred in wireless sensor networks.

4. Chordal Ring Based Node ID Assignment

4.1. CR Representation and Node ID Conversion between GCR and CR. Recall that nodes of ID n in a chordal ring (CR) have $+1$ and -1 modulo N connections. In other words, all nodes on the circumference of the ring are connected to form a Hamiltonian cycle. In the CR representation of BCG, the transform element and class representing elements are function of BCG parameters. Let $\mathbb{T}$ be a transform element in CR, and let $\mathbb{a}_i$ be the class representing elements in CR. Then, any vertex $v \in V$ is represented by $\mathbb{T}$ and $\mathbb{a}_i$ as follows:

$$v = \mathbb{T}^{j'} * \mathbb{a}_{i'}. \quad (8)$$

Since there is no systematic representation for CR and no conversion method between GCR ID and CR ID, we propose (a) a CR representation in the integer domain and (b) a conversion method between GCR ID and CR ID. The number of classes, q , can be different in the CR and GCR domains [29]. Therefore, we use q_g for the number of classes in the GCR domain and q_c in the CR domain. The node ID representation in the CR domain ($\text{ID}_c(v)$) is given as follows:

$$\text{ID}_c(v) = q_c * j' + i', \quad (9)$$

where q_c is k or p and $\text{ID}_c(v)$ is the node ID of v in the CR domain. Figure 1(b) shows the CR representation of a Borel Cayley graph in the integer domain. The graphs shown in Figure 1 represent the same BCG represented in both the CR and GCR domains.

FIGURE 3: Graph representation with finite radio range.

FIGURE 4: Illustration of CR assignment. Note that each alphabet represents a physical sensor node.

By combining (5) and (8), the conversion formulation from CR ID to GCR ID is

$$\begin{aligned} \text{ID}_c(v) &= q_c * j' + i', \\ v &= \mathbb{T}^{j'} * \mathfrak{a}_{i'} = \begin{pmatrix} a^i & j \\ 0 & 1 \end{pmatrix}, \\ \text{ID}_g(v) &= q_g * j + i, \end{aligned} \quad (10)$$

where $\text{ID}_g(v)$ is the node ID representation in the GCR domain.

Similarly, the conversion formulation from GCR ID to CR ID is

$$\begin{aligned} \text{ID}_g(v) &= q_g * j + i, \\ v &= T^j * \mathfrak{a}_i = \begin{pmatrix} a^i & j \\ 0 & 1 \end{pmatrix} = \mathbb{T}^{j'} * \mathfrak{a}_{i'}, \\ \text{ID}_c(v) &= q_c * j' + i'. \end{aligned} \quad (11)$$

Based on the set of class representing elements $\mathfrak{a}$ given in [29], the integers i' and j' can be calculated from Algorithm 1. Note that Algorithm 1 does not have any infinite loop since the matrix v is $\mathbb{T}^{j'}$ multiplied by $\mathfrak{a}_{i'}$.

We already discussed the relationship between GCR ID and the CR ID. The GCR and CR representations of BCG have different advantages. The GCR representation of BCG has an optimal routing algorithm to identify the shortest paths between any sources and destination pairs [31]. On the other hand, the CR representation supports a Hamiltonian cycle and a simple suboptimal routing algorithm. Therefore, we

assign IDs in the CR domain to minimize the communication distance and then map them to the GCR domain for optimal routing performance.

4.2. Algorithm. The chordal ring-based method (CR assignment) consists of three main steps:

- (1) making a Hamiltonian cycle in the CR domain;
- (2) converting node IDs from the CR domain to the GCR domain;
- (3) establishing edges.

Figure 4 illustrates the CR assignment algorithm. There, the algorithm starts by choosing the lowest weighted edge of sensor node ID 0 and then assigns the ID 1 plus node ID 0 to the corresponding node (Figure 4(b)). The next node selects the edge with the lowest weight among edges that are not connected to already assigned nodes. The algorithm repeats until the last node ID is assigned (Figure 4(c)). Then, the node IDs are mapped from the CR domain to the GCR domain (Figure 4(d)). Finally, the actual connections are established using GCR connection constants following the BCG connection rule (Figure 4(e)). Algorithm 2 summarizes the CR assignment.

The CR assignment guarantees that at least the lowest weighted edges of consecutive node IDs are selected as communication links, except for the edge between the starting node and the last assigned node. As a result, we optimize two out of four edges per node excluding first and last ones. We expect the total weights of edges from the CR assignment to be smaller than those of a random ID assignment. Moreover,

Require: $\mathbb{T}$ is nonsingular.

```

(1) procedure CALCULATING  $j'$  AND  $i'(\mathbb{T}, v)$ 
(2)    $j', i' \leftarrow 0$ 
(3)   while Not  $v \in \mathbf{a}$  do                                ▷ Find  $j'$  of  $\mathbb{T}$  using (8)
(4)      $v \leftarrow \mathbb{T}^{-1}v$ 
(5)      $j' \leftarrow j' + 1$ 
(6)   end while
(7)   while  $v \neq \mathbf{a}_{i'}$  do                                ▷ Find  $i'$  of  $\mathbf{a}_{i'}$  using (8)
(8)      $i' \leftarrow i' + 1$ 
(9)   end while
(10)  return  $j', i'$ 
(11) end procedure

```

ALGORITHM 1: Calculating j' and i' .

```

(1) procedure CR ASSIGNMENT ( $\mathbb{T}, \mathbf{a}$ )
(2)    $v \leftarrow v_0$                                 ▷  $v_0$ : starting node
(3)    $ID_c(v) \leftarrow 0$ 
(4)   while  $ID_c(v) \neq N$  do                        ▷ Until all nodes are assigned
(5)      $v_n \leftarrow$  select unassigned node closest to  $v$ 
(6)      $ID_c(v_n) \leftarrow ID_c(v) + 1$ 
(7)      $v \leftarrow v_n$ 
(8)   end while
(9)   Convert all IDs from CR to GCR                    ▷ Using (10)
(10)  Establish the remained edges using GCR constants
(11) end procedure

```

ALGORITHM 2: CR assignment.

the CR assignment is a fully distributed algorithm and does not require preassigned node IDs.

4.3. Average Communication Distance. We define the average communication distance as the average weights of the target graph communication distance between pairs of nodes. Our goal is to minimize the average communication distance. To evaluate our assignment algorithm, we calculated the expected communication distance analytically. From (6), the expected communication distance between randomly chosen positions is as follows:

$$E(D) = \frac{1}{a^2 b^2} \int_0^a \int_0^b \int_0^a \int_0^b w_{uv} du_x dv_x du_y dv_y, \quad (12)$$

where a and b represent the horizontal and vertical dimensions, respectively.

Figure 5 shows the average communication distance for each node ID assignment algorithm where $a, b = 100$ m. The random assignment algorithm uniformly and randomly assigns a unique BCG node ID to all the sensors. We call the resultant network topology with Random assignment BCG-0 and the resultant network topology with CR assignment BCG-1. The average weight of the random assignment and the expected distance are both approximately equal to 52.

We also calculated the standard deviation for the average communication distance from 100 CR assignment samples.

FIGURE 5: Average communication distance.

The results show that selection of the initial node does not affect the average communication distance. For graphs of sizes $N = 272, 506, 1081, 1474$, and 2265 , the standard

deviations for the CR assignment were 0.58, 0.53, 0.62, 0.72, and 0.60, respectively.

5. Distributed Node ID Swapping Assignment

In the previous section, we showed the CR assignment algorithm for reducing the average communication distance without radio range constraint. In this section, we propose the distributed node ID swapping assignment for completing communication edges with radio range constraint. The distributed node ID swapping assignment (Dist-swap assignment) consists of four main steps executed by the current node:

- (1) broadcast its node ID to its physical neighbors;
- (2) collect IDs of physical neighbors that can be swapped;
- (3) select the best-fit node ID to be swapped with;
- (4) swap node IDs and update each logical neighbor table of its physical neighbors.

5.1. Terminologies. We denote the logical node ID of node x by x_{id} and define the following.

- (i) $D(x_{id})$: set of logical neighbors of node ID x_{id} in the target graph;
- (ii) $T(x)$: set of logical IDs of node x 's physical neighbors;
- (iii) $L(x_{id}, y) = |D(x_{id}) \cap T(y)|$: the number of logical neighbors of node ID x_{id} that are also physical neighbors of node y .

We define four different packet types.

- (1) **Token packet**: forwarded by nodes and used to initiate the node swapping algorithm by the node recipient. Also contains counter $repCnt$.
- (2) **Info packet**: used by the current node holding the token to broadcast x_{id} to its physical neighbors so as to identify candidate nodes to be swapped.
- (3) **Request packet**: Supports collection of candidate nodes. If a physical neighbor node y receiving an Info packet from node x determines that it can be swapped, it sends a Request packet back to node x . The Request packet contains y_{id} , $L(y_{id}, y)$, and $L(x_{id}, y)$.
- (4) **Swap packet**: used by the current node x to announce to its physical neighbors swapping with node y . Contains the pair (x_{id}, y_{id}) . Nodes having those node IDs in $T()$ need to update the swapped IDs.

5.2. Assumption. We assume that (a) a host graph is a connected graph (there is a path between all node pairs), (b) nodes have preassigned unique IDs ranging from 0 to $N - 1$, (c) $T(x)$ (a set of logical IDs of its physical neighbors) at each node $x \in V$ is obtained before the swapping process, and (d) the order of nodes to perform operation is based on token method.

5.3. Algorithm. The Dist-swap assignment is performed by one node at a time using a token method (i.e., there is one token in a network, and only the node holding the token executes the swapping algorithm). To prevent an infinite loop, we rely on a token counter ($repCnt$) that tracks the number of times the token is being passed around. At the beginning of the Dist-swap assignment, the node that starts the operation sets $repCnt$ to $repTotal$ that was heuristically selected based on previous simulation studies. As the token travels around the network, each node decreases $repCnt$ by one before forwarding the token to the next node, and the Dist-swap assignment ends once $repCnt$ reaches zero.

We describe in detail the Dist-swap assignment as follows:

(i) the following describes the Dist-swap assignment operation after node x receives a Token packet.

- (1) A node x receives a Token packet.
  - (a) If $|D(x_{id})| = L(x_{id}, x)$, forward the Token packet randomly to one of its physical neighbors.
  - (b) Otherwise, send an Info packet to all its physical neighbors.
- (2) Node x collects the Request packets.
  - (a) If there is no Request packet received, forward the Token packet randomly to one of its physical neighbors.
- (3) Node x creates a list of candidate nodes. A node y is a candidate node if $L(y_{id}, x) + L(x_{id}, y) - L(x_{id}, x) - L(y_{id}, y) > 0$.
  - (a) If there is no candidate node, forward the Token packet randomly to one of its physical neighbors.
  - (b) Otherwise, select the node y maximizing the number of logical neighbors when two node IDs are swapped.

(4) Node x sends Swap packet with (x_{id}, y_{id}) .

(5) Node x sends Token packet to node y .

First, node x determines whether or not it needs to execute the swapping process based on its current number of logical neighbor IDs in the physical neighbors and its target number of logical neighbors in the target graph. If those are not the same, node x sends an Info packet to its physical neighbors. After receiving Request packets from its physical neighbors, node x determines which node ID swapping is the most beneficial. Because the Request packet from node y contains y_{id} , $L(y_{id}, y)$, and $L(x_{id}, y)$, node x can calculate the change in the number of logical neighbors from the physical neighbors when node IDs x_{id} and y_{id} are swapped. If a candidate node for swapping exists, node x sends a Swap packet with its own node ID and the selected node ID to its neighbors.

(ii) The following illustrates the operation after node y receives an Info packet.

- (1) A node y receives an Info packet.
- (2) Node y calculates the number of its logical neighbors among its physical neighbors if it was assigned node ID x_{id} .
  - (a) If that number is larger than or equal to the number of logical neighbors of the current ID, send the Request packet.
  - (b) Otherwise, ignore the Info packet.

To determine whether or not to reply with a Request packet, node y checks the number of its logical neighbors that would be assigned to its physical neighbors if it were assigned node ID x_{id} . If it is larger than that with the current node ID, node y sends a Request packet.

(iii) The following shows the operation when node z receives a Swap packet.

- (1) A node z receives a Swap packet that contains node IDs (x_{id}, y_{id}) .
  - (a) If z_{id} is equal to y_{id} , change z_{id} to x_{id} and send a Swap packet with node IDs (z_{id}, x_{id}) to its physical neighbors.
  - (b) If z_{id} is equal to x_{id} , ignore the Swap packet.
  - (c) Otherwise, change x_{id} to y_{id} in $T(z)$.

Swap packet contains node IDs (x_{id}, y_{id}) . If y_{id} is the same as z_{id} , then the Request packet sent by z is accepted. So, node z changes its node ID to x_{id} and sends a Swap packet with (z_{id}, x_{id}) to its physical neighbors so that they update their $T(\cdot)$ s. When x_{id} is the same as z_{id} , node z ignores that packet. Otherwise, node z updates $T(z)$ to reflect that node ID x_{id} is changed to y_{id} .

5.4. Example. Next, we use an example to illustrate the Dist-swap assignment operation. The physical nodes of the host graph are denoted by nodes x , y , and z , and the logical node IDs of the target graph are denoted by $1, 2, \dots$, and so forth. Figure 6 shows that logical node IDs 5, 8, and 15 are assigned to nodes x , y , and z , respectively. We assume that the target graph has a simple connection rule; namely, node ID i is connected to $i-4, i-2, i+2$, and $i+4$. Given the connection rule in the target graph, the neighbors of node ID 5 in the target graph are: $D(5) = \{1, 3, 7, 9\}$. Similarly, $D(8) = \{4, 6, 10, 12\}$, and $D(15) = \{11, 13, 17, 19\}$.

First, node x receiving a Token packet decides whether or not to perform the swapping process based on the current number of logical neighbors among its physical neighbors ($L(5, x)$). The number of logical neighbors among the physical neighbors is not the same as the maximum number of logical neighbors 4 since $T(x) = \{4, 6, 8, 13, 15\}$, $D(5) = \{1, 3, 7, 9\}$ and $L(5, x) = 0$. So, node x sends an Info packet to its physical neighbor nodes. Swapping node IDs is beneficial to node y because $L(8, y)$ is zero and $L(5, y)$ is two, which means node y gains two more logical neighbors after swapping IDs with node ID 5. Thus, node y sends the Request packet to node x . Node z also sends a Request packet to node x through the same process. Finally, node x determines to

FIGURE 6: Host graph used as example of Dist-swap assignment.

TABLE 2: Summary of our proposed algorithms.

Resultant graphs	Algorithms
BCG-0	Random assignment
BCG-1	Chordal ring based node ID assignment
BCG-2	Distributed node ID swapping assignment
BCG-3	Distributed node ID swapping assignment and random node selection

swap IDs with node y because $L(8, x) + L(5, y) = 4$ is greater than $L(15, x) + L(5, z) = 3$.

5.5. Completeness of Resultant Graph. Recall that an efficient ID assignment method should incorporate as many as possible target graph edges in the resultant graph from a given host graph, where sensor nodes are limited to a specific radio range. Therefore, to measure the performance of our assignment methods, we define R_c , the ratio of the number of edges of the resultant graph over that of the target graph. That is,

$$R_c = \frac{\text{Number of edges in resultant graph}}{\text{Number of edges in target graph}}. \quad (13)$$

For a given radio transmission range, the assignment method with the largest R_c is the most efficient since most edges of the target graphs are incorporated in the resultant graph. We define the resultant network topology with Dist-swap assignment BCG-2. Figure 7 shows R_c for BCGs target graphs with $N = 1081$. The Dist-swap assignment (BCG-2) shows larger R_c than that of the random assignment (BCG-0). Especially, with 50 m radio range in 100 m by 100 m area, it showed 43% more edges.

When applying BCGs to wireless sensor networks, we also need to consider network connectivity. Because the Dist-swap assignment maximizes R_c without considering network connectivity, we apply a random node selection algorithm on processed nodes that do not have the right number of logical neighbors. We define the resultant graph from this method BCG-3. Table 2 summarizes resultant graphs from our proposed algorithms.

Figure 8 shows the percentage of connected network function of assignments for 100 network samples. BCG-0 with 105 m radio range results in a 100% connected network. BCG-2 and BCG-3 require 65 m and 10 m radio

FIGURE 7: Edge construction percentage with distributed node ID assignment.

FIGURE 8: Connected network percentage with node ID assignments.

range, respectively, to produce a connected network. All connections of BCG-2 are satisfied by the connection rules of BCGs. However, BCG-3 has non-BCG connections since some are logical neighbors randomly selected for improved connectivity.

6. Simulation

6.1. Consensus Protocol. We use the consensus protocol as a benchmark to evaluate our topology control algorithms. Consensus protocol is a distributed node-to-node message exchange rule to reach a network-wide agreement over a quantity of interest (e.g., average of sensory data). Consensus protocol has a long history in distributed computing and has been used in a variety of applications. Moreover, it has been widely accepted as a reliable measure of data fusion performance of network topologies [35, 36]. In WSN, consensus protocol research focused on time synchronization and gossip algorithms [37–39]. Readers interested in more detail on consensus protocol and its application are referred to [40].

The consensus agreement value can be the average, the maximum, the minimum, or any function and only depends on the initial states of the nodes in the network. Furthermore, the speed of consensus is a good measure of the efficiency of a network topology to distribute information. In this study, we consider the distributed average consensus protocol proposed in [40] which we summarize next.

Let us consider a network system of which logical network topology is represented by an unweighted, undirected graph. Each node $v \in V$ in the system communicates its state value x_v to its immediate neighbors $N(v) := u : e_{v,u} \in E(V)$, where E is an edge set. At each iteration τ , nodes exchange their current state values $x_v(\tau)$ with their immediate neighbors. Given the state values $x_u(\tau)$ received from their neighbors $u \in N(v)$, each node u updates its state according to

$$x_u(t+1) = x_u(t) + \frac{1}{\omega} \sum_{v \in N(u)} (x_v(t) - x_u(t)), \quad (14)$$

where ω is typically $0 < 1/\omega < 1/2d_{\max}(G)$ and $d_{\max}(G)$ is the largest nodal degree. We set ω to $2 \times d_{\max}(G) + 1$ in this paper. Consensus convergence can be controlled by ω value. However, finding an optimal ω value is out of scope for this paper.

Following the method in [41], we use the average consensus protocol (14) to measure the information fusion performance of the network generated by the proposed topology control protocol. BCGs have already been shown to exhibit better consensus protocol performance than mesh, torus, and small world networks [33]. However, that research compared only topology level not considering nodes' physical geometric information and radio range. In this paper, we compare a BCG with other network topologies when applied to wireless sensor network.

6.2. Simulation Setup. We evaluated our proposed topology control algorithms in terms of consensus speed and power consumption. The simulations were executed on 100 host graphs, each with sensors uniformly and randomly distributed over a $100 \text{ m} \times 100 \text{ m}$ area.

Consensus protocol was initialized with nodes state value set to integers randomly chosen between -5 and 5 inclusive. We declared a network topology to have reached an

FIGURE 9: Histograms of communication distance by the proposed algorithms with $N = 1081$.

agreement once all node values equal the average of all initial state values within a precision of 0.001. The performance of the consensus protocol was measured as the number of steps needed to reach a network-wide consensus.

We also computed the energy needed for all the nodes in the network to reach a consensus using the average consensus protocol. To do this, we utilized the radio model described in [42]. For the sake of simplicity, we only considered the energy consumptions from data transmission (E_{tx}) and reception (E_{rx}) defined as follows:

$$\begin{aligned} E_{tx} &= B(\beta_1 + \beta_2 \omega(v, u)^n), \\ E_{rx} &= B\gamma, \end{aligned} \quad (15)$$

where n is the path loss exponent which, in typical, ranges from 2 to 6. The constants β_1 , β_2 , and γ are the energy dissipated by the transmitter module, transmit amplifier, and the receiver module, respectively. We denote the estimated distance between nodes v and u by $\omega(v, u)$ and the length of

TABLE 3: Radio model parameters.

Parameters	Values
n	2
β_1	45 nJ/bit
β_2	10 pJ/bit
γ	135 nJ/bit
B	320 bits (=40 bytes)

the message by B . We set $n = 2$, equivalent to the free-space pass loss model, and assume that a sensor node v , regardless of the topology control protocols considered, can adjust its transmission power to reach its neighbors w . The parameters used in the simulation are summarized in Table 3.

Using the average consensus protocol, a node transmits and receives to and from each of its neighbors at every

FIGURE 10: Power consumption comparison between BCG-0 and BCG-1.

iteration. Thus, the node's energy consumed by the network at the τ th iteration is given by

$$P_0(v) = \tau B \left(\sum_{\forall (v,u) \in E, v \neq u} (\beta_1 + \beta_2 \omega(v,u)^n) + \gamma \deg(v) \right), \quad (16)$$

where $\deg(v)$ is the number of logical neighbors, (v,u) is a logical connection, and E denotes the edge set. The average nodal energy consumption is given by $\bar{P}_0 = \sum_{v \in V} P_0(v) / |V|$. We call this power consumption model *Power model 0*.

In wireless sensor networks, a network can support multicasting routing to transmit at once to multiple nodes inside the radio range. Then power consumption is calculated not by each communication edge but by the maximum distance communication edge as follows:

$$P_1(v) = \tau B (\beta_1 + \beta_2 \text{Max} \{ \omega(v,u), u \in \text{adj}(v) \}^n + \gamma \deg(v)). \quad (17)$$

We call this power consumption model *Power model 1*.

6.3. Comparison between BCG-0 and BCG-1. Figure 9 shows the histograms of the communication distance for $N = 1081$ generated by the proposed node ID assignment algorithms. The histogram of the CR assignment (BCG-1) exhibits a right skewed distribution with a high frequency of short edges. The Dist-swap assignment (BCG-2) shows that all connection distance are under 80 m. We obtained similar histograms for the network sizes listed in Table 1. We compared power consumption between the random assignment and the CR assignment because they require a radio range that covers the whole node deployment area. Figure 10 summarizes the

resulting consensus protocol energy consumption. We found that BCG-1 consumed 8% less energy than BCG-0 with power model 0 and 2% less with power model 1.

6.4. Comparison between Our Proposed Topology Control and Existing Topology Controls. We compared our topology control algorithms with existing topology controls. The evaluation was done in terms of diameter, average path length, consensus steps, and nodal energy consumption.

We used the following topologies for comparison against our approach.

- (i) Max: network where all nodes within the maximum radio range are logical neighbors. Without any topology control, a network is called Max topology.
- (ii) E-MST [43]: the Euclidean minimum spanning tree is a minimum spanning tree whose edge weight is the distance between nodes. This algorithm minimizes the summed weight of edges and generates a connected network.
- (iii) K-Neigh [44]: network where the number of neighbors of a node equals to or is slightly below a given value k .
- (iv) Gabriel Graph [45]: a planar graph (no edges cross one another) supports geographical routings and is defined such that an edge (v,u) exists if no other node is inside the circle with the diameter uv .

Figure 11 provides illustrations of the topologies constructed with (a) the random assignment (BCG-0), (b) the Dist-swapping assignment (BCG-2), (c) Max, (d) E-MST, (e) K-Neigh, and (f) Gabriel graph. For ease of description, we show sample graphs with $N = 272$. However, we simulated with $N = 1081$ in this section.

Figures 12 and 13 show the diameter and average path length function of the transmission range, respectively. We only compared the cases where the resultant graph is a connected graph. From these figures, we confirm again that the BCGs have the smallest diameter and the shortest average path length for a given radio range except Max topology. This is because as a radio range increases, BCGs have a constant number of logical neighbors, while Max topology has much more logical neighbors than others.

From Figure 14, at radio range 30 m, consensus protocol convergence for BCG-3 was 35.5, 2932, and 15.9 times faster than that of Gabriel, E-MST, and K-Neigh topologies. At radio range 50 m, convergence of BCG-3 was 81.2, 6756.9, and 39.9 times faster than that of Gabriel, E-MST, and K-Neigh topologies. Even though Max topology has the best consensus steps, due to its large number of logical neighbors, it consumed 23 times more nodal power than BCG in Figure 15.

When considering multicast routing (Power model 1), the Max topology has an obvious advantage as only the maximum radio range is used in the power consumption computation. However, even with this advantage, our simulations showed that BCG continues to outperform the Max topology in Figure 16. This better performance can be explained by the

FIGURE 11: BCG-0, BCG-3 with radio range 40 m, Max with 25 m, E-MST, K-Neigh with $k = 10$ and 25 m, Gabriel graph. $N = 272$.

FIGURE 12: Comparison of diameters.

FIGURE 13: Comparison of average path lengths.

fact that the Max topology has a larger number of logical neighbors and hence larger power is consumed at the receiver modules.

7. Conclusions

Our goal was to use Borel Cayley graphs, a family of predefined graph topology, to overlay on dense wireless sensor

networks for topology control. In this paper, we proposed node ID assignment methods to reduce the communication distance between nodes or to increase the number of logical connections following predefined graph connection rules in a distributed manner. In particular, we proposed the chordal ring-based node ID assignment method and the distributed node ID swapping assignment method.

FIGURE 14: Comparison of consensus steps.

FIGURE 16: Comparison of power model 1 for consensus protocol.

FIGURE 15: Comparison of power model 0 for consensus protocol.

In the chordal ring-based method, we exploit that all BCGs have CR representations which imply the existence of a Hamiltonian cycle. With this observation, node IDs can simply be progressively assigned in numerical order based on their physical locations. Once the node IDs are assigned in the CR domain, we provide an explicit formulation to convert the CR domain ID to GCR domain ID which supports efficient routing. Our simulation results over a range of network sizes between 300 and 2000 nodes showed that this node ID

assignment requires a smaller radio range (57%) than the random ID assignment.

In the distributed node ID swapping assignment method, we consider that nodes have a limited transmission range. In WSNs, it is challenging to impose an entire logical graph to a physical network of finite radio range. An efficient node ID assignment will allow more connections to be imposed. For a range of network sizes from 200 to 1,000 nodes, the simulation showed that our proposed method achieved 43% more connections of all desired edges maximally than that of the random assignment. We also showed that our proposed topology control algorithms are efficient in consensus protocol comparing existing topology controls such as K-Neigh, Gabriel, and E-MST.

In summary, in this paper, we showed that BCGs are good candidates as a family of predefined graphs for topology control for WSNs. However, there are many practical issues in the application of BCG as predefined graphs. For example, wireless packet transmission is interfered by simultaneous transmission of packets among nodes in the network. In future work, we will investigate the effects of interferences on BCG-based networks.

Acknowledgment

The authors are partially supported by the National Science Foundation under Grant no. CNS 0829656 and IIP 0917956. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- [1] D. Culler, D. Estrin, and M. Srivastava, "Introduction: overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, 2004.
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [3] C. E. Perkins, "Ad hoc networking: an introduction," in *Ad Hoc Networking*, pp. 1–28, Addison-Wesley Longman, Boston, Mass, USA, 2001.
- [4] T. Schmid, H. Dubois-ferriere, and M. Vetterli, "Sensorscope: experiences with a wireless building monitoring sensor network," in *Proceedings of the 1st Workshop on Real-World Wireless Sensor Networks (REALWSN '05)*, 2005.
- [5] G. Werner-Allen, K. Lorincz, M. Welsh et al., "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, 2006.
- [6] L. Selavo, A. Wood, Q. Cao et al., "LUSTER: wireless sensor network for environmental research," in *Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys '07)*, pp. 103–116, November 2007.
- [7] K. Martinez, J. K. Hart, and R. Ong, "Environmental sensor networks," *Computer*, vol. 37, no. 8, pp. 50–56, 2004.
- [8] S. Coleri, S. Y. Cheung, and P. Varaiya, "Sensor networks for monitoring traffic," in *Proceedings of the Allerton Conference on Communication, Control and Computing*, 2004.
- [9] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 51–62, November 2003.
- [10] E. Shih, S.-H. Cho, N. Ickes et al., "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 272–287, New York, NY, USA, July 2001.
- [11] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 56–67, August 2000.
- [12] R. Rajaraman, "Topology control and routing in ad hoc networks: a survey," *SIGACT News*, vol. 33, pp. 60–73, 2002.
- [13] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys*, vol. 37, no. 2, pp. 164–194, 2005.
- [14] S. B. Akers and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," *IEEE Transactions on Computers*, vol. 38, no. 4, pp. 555–566, 1989.
- [15] K. N. Sivarajan and R. Ramaswami, "Lightwave networks based on de Bruijn graphs," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 70–79, 1994.
- [16] J. Sun and E. Modiano, "Capacity provisioning and failure recovery for Low Earth Orbit satellite constellation," *International Journal of Satellite Communications and Networking*, vol. 21, no. 3, pp. 259–284, 2003.
- [17] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Computing Surveys*, vol. 38, no. 1, pp. 71–121, 2006.
- [18] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.
- [19] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup service for internet applications," in *Proceedings of the Applications, Technologies, Architectures, and Protocols for Computer Communications (ACM SIGCOMM '01)*, pp. 149–160, New York, NY, USA, August 2001.
- [20] M. F. Kaashoek and D. R. Karger, "Koorde: a simple degree-optimal distributed hash table," in *Peer-to-Peer Systems II*, vol. 2735 of *Lecture Notes in Computer Science*, pp. 98–107, 2003.
- [21] M. Naor and U. Wieder, "Novel architectures for P2P applications: the continuous-discrete approach," *ACM Transactions on Algorithms*, vol. 3, no. 3, Article ID 1273350, 2007.
- [22] D. Loguinov, J. Casas, and X. Wang, "Graph-theoretic analysis of structured peer-to-peer systems: routing distances and fault resilience," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 1107–1120, 2005.
- [23] C. Qu, W. Nejdl, and M. Kriesell, "Cayley dhTs—a group-theoretic framework for analyzing dhTs based on cayley graphs," in *Proceedings of the International Symposium on Parallel and Distributed Processing and Applications (ISPA '04)*, Springer, 2004.
- [24] S. S. Iyengar, D. N. Jayasimha, and D. Nadig, "A versatile architecture for the distributed sensor integration problem," *IEEE Transactions on Computers*, vol. 43, no. 2, pp. 175–185, 1994.
- [25] A. A. Taleb, J. Mathew, and D. K. Pradhan, "Fault diagnosis in multi layered De Bruijn based architectures for sensor networks," in *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops '10)*, pp. 456–461, April 2010.
- [26] A. A.-B. Al-Mamou and H. Labiod, "ScatterPastry: an overlay routing using a DHT over wireless sensor networks," in *Proceedings of the International Conference on Intelligent Pervasive Computing (IPC '07)*, pp. 274–279, October 2007.
- [27] M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea, and A. Rowstron, "Virtual ring routing: network routing inspired by dhTs," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '06)*, pp. 351–362, ACM, New York, NY, USA, 2006.
- [28] F. Araujo, L. Rodrigues, J. Kaiser, C. Liu, and C. Mitidieri, "Chr: a distributed hash table for wireless ad hoc networks," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops*, pp. 407–413, 2005.
- [29] K. W. Tang and B. W. Arden, "Representations of borel cayley graphs," *SIAM Journal on Discrete Mathematics*, vol. 6, pp. 655–676, 1993.
- [30] B. W. Arden and K. W. Tang, "Representations and routing for Cayley graphs," *IEEE Transactions on Communications*, vol. 39, no. 11, pp. 1533–1537, 1991.
- [31] K. W. Tang and B. W. Arden, "Vertex-transitivity and routing for Cayley graphs in GCR representations," in *Proceedings of the ACM/SIGAPP Symposium on Applied Computing (SAC '92)*, pp. 1180–1187, March 1992.
- [32] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [33] J. Yu, E. Noel, and K. W. Tang, "A graph theoretic approach to ultrafast information distribution: borel Cayley graph resizing algorithm," *Computer Communications*, vol. 33, no. 17, pp. 2093–2104, 2010.

- [34] J. Ryu, J. Yu, E. Noel, and K. Tang, "Node id assignment in group theoretic graphs for wsns," in *Proceedings of the Wireless Telecommunications Symposium (WTS '11)*, pp. 1–8, 2011.
- [35] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 63–70, April 2005.
- [36] Y. Hatano and M. Mesbahi, "Agreement over random networks," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, vol. 2, pp. 2010–2015, December 2004.
- [37] L. Schenato and G. Gamba, "A distributed consensus protocol for clock synchronization in wireless sensor network," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 2289–2294, December 2007.
- [38] P. Braca, S. Marano, and V. Matta, "Running consensus in wireless sensor networks," in *Proceedings of the 11th International Conference on Information Fusion (FUSION '08)*, pp. 1–6, July 2008.
- [39] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.
- [40] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [41] R. Olfati-Saber, "Ultrafast consensus in small-world networks," in *Proceedings of the American Control Conference (ACC '05)*, vol. 4, pp. 2371–2378, June 2005.
- [42] M. Marta and M. Cardei, "Using sink mobility to increase wireless sensor networks lifetime," in *Proceedings of the 9th IEEE International Symposium on Wireless, Mobile and Multimedia Networks (WoWMoM '08)*, pp. 1–10, June 2008.
- [43] N. Li, J. C. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 1195–1206, 2005.
- [44] D. M. Blough, M. Leoncini, G. Resta, and P. Santi, "The K-Neigh protocol for symmetric topology control in ad hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC '03)*, pp. 141–152, New York, NY, USA, June 2003.
- [45] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, August 2000.