

Topology-Based Routing for Xmesh in Wireless Sensor Networks

Lei Wang and K. Wendy Tang

Abstract Large wireless sensor networks can contain hundreds or thousands of sensor nodes. Due to wireless sensor network's properties of low-energy-efficiency, large-scale, low cost, and lossy nature, the development of efficient routing protocols for these large and dense wireless sensor networks is an interesting research topic. This research focuses on the design and implementation of protocols for dense and wireless sensor networks. More specifically, we propose to combine an underlying topology with *Xmesh*, the multihop routing strategy of Crossbow Technology's motes. In an effort to limit the path lengths, we propose to impose an underlying connectivity graph for *Xmesh*. The underlying connectivity graph is a virtual topology of the network, hence the name "topology-based routing." Instead of being always forwarded to the node with the best link quality among all neighbors, a packet is being routed according to the shortest path routing of the underlying graph which guarantees a bounded path length. *Cayley* graphs from the *Borel* subgroup have been known as the densest degree-4 graphs and all *Cayley* graphs are vertex transitive. In this work, we propose a topology-based routing for *Xmesh* with *Cayley* graphs as the underlying topology. We show that, indeed, by imposing a *Cayley* graph as an underlying graph, the average path lengths between nodes is smaller and that the averaged power consumed is less than the original *Xmesh*.

1 Introduction

Wireless sensor network (WSN) consists of a large number of nodes with different kinds of sensors, linked by a wireless medium (radio frequency) to perform distributed sensing tasks. These networks can be applied in many environments such as intelligent battlefields, smart hospitals, environment response systems, and

L. Wang (✉)

State University of New York, Stony Brook, New York, USA

e-mail: leiwang@mail.ee.sunysb.edu

surveillance systems. In most applications, mainly unwired power supply and communication bandwidth are constrained for sensor nodes [1]. Therefore, to shorten network lifetime and to use the limited bandwidth efficiently, researchers emphasize energy conservation in the design and management of WSNs [2]. At the network layer, finding methods for energy-efficient route discovery and relaying of data from the sensor nodes to the base station is highly desirable. There are still other concerns when designing WSN protocols, such as fairness, fault tolerance, node/link heterogeneity, and network dynamics. The dynamic and lossy nature of wireless communication poses major challenges to reliable, self-organizing multihop networks. For large and dense WSN with a few hundred or thousands of nodes, energy conservation, scalability, and self-configuration are the primary goals [3, 4]. Many protocols have been proposed for WSN [1–6].

Crossbow Technology Inc. has been one of the major vendors for wireless sensor networks. Its powerful battery-powered platform runs on the open-source TinyOS operating system. With this operating system, developers can control low-level event and maintain task management. Its multihop routing protocol called “Xmesh” is a distributed routing process [7, 8]. Routing decisions are based on a minimum transmission cost function that considers link quality of nodes within a communication range. However, there are no limits on the path length. In extreme cases and for large networks, it is conceivable that a packet may need to hop through many intermediate nodes before reaching its intended destination.

In an effort to limit the path lengths, we propose to impose an underlying connectivity graph for Xmesh. The underlying connectivity graph is a virtual topology of the network. Instead of being forwarded to the best link quality node among all neighbors within communication range, a packet is being routed according to the shortest path routing of the underlying graph. In the event that multiple shortest paths exist, the one with the best link quality is chosen. The purpose of the underlying connectivity graph is to impose a virtual topology that facilitates routing and guarantees a bounded path length. An ideal underlying graph should guarantee a small number of hops between nodes and should possess a simple routing algorithm.

Cayley graphs from the Borel subgroup [9] have been known as the densest degree-4 graphs and all Cayley graphs are vertex transitive or symmetric. Furthermore, our earlier work has shown that Cayley graphs can have very effective integer representation and symmetric routing strategies [9–12]. In this chapter, we propose a topology-based routing for Xmesh with Cayley graphs as the underlying virtual topology. To evaluate the effect of imposing such a virtual topology on Xmesh, we simulated our proposed protocol via Power Tossim, an emulator for wireless sensor networks. We show that, indeed, by imposing a Cayley graph as an underlying graph, the average path lengths between nodes is smaller and that the averaged power consumed is less than the original Xmesh.

This chapter is organized as follow: Section 2 provides an overview of Cayley graphs and its routing algorithm. Section 3 gives a detailed description of our proposed topology-based routing for Xmesh. Simulation results and analysis are described in Section 4. A summary and conclusions are described in Section 5.

2 Routing for Cayley Graphs

2.1 Cayley Graph Overview

Symmetric, regular, undirected graphs are useful models for interconnection of multicomputer systems. Dense graphs of this sort are particularly attractive. Based on group theoretic constructions, Cayley Graphs [13] are in this category of graphs [9]. The construction of Cayley graphs is described by finite (algebraic) group theory.

Definition A graph $C = (V, G)$ is a Cayley graph with vertex set V if two vertices $v_1; v_2 \in V$ 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.

Note that the identity element I is excluded from G . This prevents the graph from having self-loops. In this chapter, we are interested in undirected, degree-4 Cayley graphs. In other words, the generator set consists of two group elements and their inverses. Cayley graphs are vertex transitive. Furthermore, the densest known degree-4 graphs are Cayley graphs from the Borel group [13].

The dense property of Cayley graphs implies that they can connect a large number of nodes via a small number of hops through intermediate nodes. The vertex-transitive property is useful for routing. It means that a Cayley graph “looks the same from any node” which means the same routing algorithm and routing table can be used at every node. More specifically, routing between vertices i and j can be determined by finding paths between vertices 0 and j' , where j' is a function of i and j . This property is the basis for a distributed routing algorithm, the vertex-transitive routing in [10–12]. Figure 1 is a 21-node, degree-4, Borel Cayley graph in the integer domain.

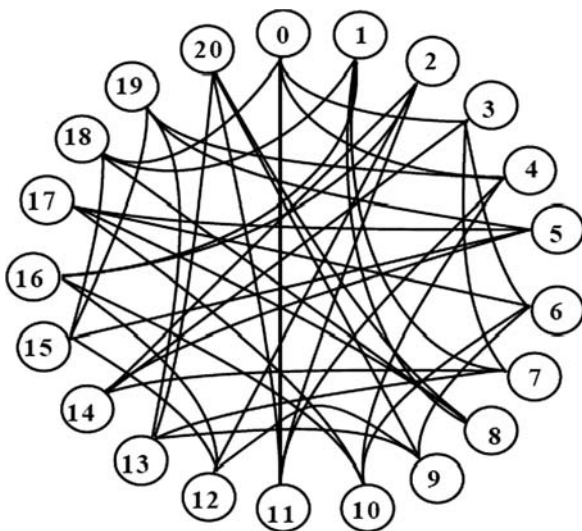


Fig. 1 A 21-node Cayley graph

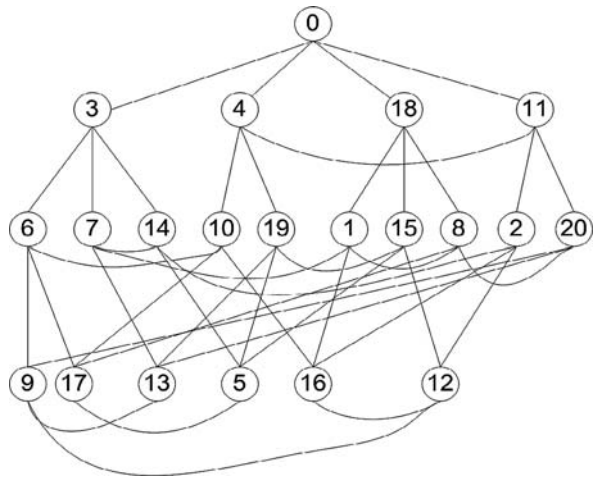
The graph has $V = \{0, 1, \dots, 20\}$, and the connection is defined as [9–12]:

- Let $V = \{0, 1, \dots, 20\}$. For any $i \in V$, if $i \bmod 3 = :$
- 0: i is connected to $i+3, i-3, i+4, i-10; \bmod 21$
 - 1: i is connected to $i+6, i-6, i+7, i-4; \bmod 21$
 - 2: i is connected to $i+9, i-9, i+10, i-7; \bmod 21$

2.2 Shortest Path Algorithm for Cayley Graph

To implement Cayley Graph into Xbow’s notes, we implemented an algorithm for path searching to find the shortest path between any two vertices. In [5], each vertex has 4 degree, 2 for incoming communication and 2 for outgoing communication. In this chapter, the two incoming and two outgoing channels merge into one channel for single half duplex transceiver. Therefore, there are multiple choices of shortest path selecting in terms of hops. For example, in Fig. 2, there are two shortest paths (with a hop count of three) from noted 0 to node 9.

Fig. 2 Tree view with node 0 as root



Inspired by Dijkstra and “On-Demand Route Discovery” algorithms [14, 15], we chose an algorithm that combines depth-first-search [16] and breadth-first-search [17]. Below is the description of the algorithm that we use to generate routing tables for all nodes. A routing table at a node stores the optimal outgoing links from that node to all other nodes. Because of the vertex-transitive property of Cayley graphs, routing tables for all the nodes can be generated by a simple integer mapping of the routing table from node 0 Table 1 [9–12].

Table 1 An algorithm to generate routing tables for all nodes

-
- Step 1: Initialize the four possible paths and corresponding distance of each node with infinite numbers.
 - Step 2: Starting from the four nodes j directly connected to node 0 (distance = 1), Breadth-first search the nodes below these four nodes with the recursive function.
 - 1) While the depth of searching $\leq \log_4(n) + 1$, do 2).
 - 2) If next connected node $j_1/j_2/j_3/j_4$ is not visited through this link $a/b/c/d$, in other words, distance = infinity, and $j_1/j_2/j_3/j_4$ is not root of this tree, go to the following procedure:
 - a) Update the distance and path information for node $j_1/j_2/j_3/j_4$;
 - b) Find the next four connected nodes of node $j_1/j_2/j_3/j_4$, flag this link $a/b/c/d$, call procedure 1).
 - Step 3: Among the paths of each node, select the those with minimum distance, and flag those corresponding links in routing table for the tree with node 0 as root.
 - Step 4: Manipulating the vertex transitive formula given by [12, 22, 23], generate the routing table for other vertices.
-

3 Topology-Based Routing with Xmesh

3.1 Introduction to Xmesh

The multihop routing protocol, Xmesh [7], is implemented in a TinyOS application called Surge. Xmesh is a distributed routing process that has three local processes: *link quality estimation*, *neighborhood management*, and *connectivity-based route selections*. The link quality estimator estimates the link quality of all nodes within communication range. The neighborhood management process decides how the node chooses neighbors for paths. Link estimation and neighborhood management build a probabilistic connectivity graph. The routing process then builds topologies upon this graph. These three processes together form a holistic approach with the goal of minimizing total cost and providing reliable communications. The core component of Xmesh is the neighbor table which contains status and routing entries for neighbors; its fields include MAC address, routing cost, parent address, child flag, reception (inbound) link quality, send (outbound) link quality, and link estimator data structures.

The Component of Parent selection is running periodically to select one of the potential neighbors for routing. A packet has fields for parent address, estimated routing cost to the base, and a list of reception link estimations of neighbors. When a node receives a route message already in its neighbor table, the corresponding entry is updated. If not, the neighbor table manager decides whether to insert the node or drop the update. Originated data packets, such as outputs of local sensor processing, are queued for sending with the parent as the destination. Incoming data packets are selectively forwarded through the forwarding queue. To avoid cycles the corresponding neighbor table entry is flagged as a child in parent selection. Duplicate forwarding packets are eliminated. When cycles are detected on forwarding packets, parent selection is triggered with the current parent demoted to break the cycle.

3.2 Topology-Based Routing with Xmesh

In our proposed topology-based routing with Xmesh, we use a Cayley graph as an underlying topology for Xmesh routing. In Xmesh, a packet is forwarded to an intermediate node that has the best link quality within the communication range of the source node. The packet will be forwarded until it reaches its intended destination. In extreme cases and for large networks, such multihop routing can result in long path length, i.e., a packet will go through large number of intermediate nodes.

By routing packets in Xmesh according to an underlying graph, we can impose a limit on the path length. In our proposed topology-based routing, a packet is forwarded to an intermediate node that has the best link quality *among neighbors of an underlying graph*. By imposing a topology on forwarding packets, the path length of the message is bounded by the *diameter* (maximum of the minimal distance) of the graph. Obviously, it is important to choose an underlying graph that is *dense* (small diameter for large number of nodes). We choose to use Cayley graphs as the underlying topology because of its vertex-transitive and potentially dense properties.

The vertex-transitive property of Cayley graphs enables us to use the same routing table at each node. A routing table is first generated off-line that stores the *optimal* outgoing links from node 0 to all other nodes in the network. An outgoing link is *optimal* if it contributes to a shortest path between the source and the destination. This routing table of size $O(n\delta)$, where n is the number of nodes and δ is the degree of the graph, is stored at every node. A vertex-transitive mapping formula [9–12] is used to identify the appropriate entry in the routing table to determine what are the optimal outgoing links.

4 Simulation Results and Analysis

4.1 Power TOSSIM

We implement the topology-based routing (based on Cayley graphs) for Xmesh in the TinyOS sensor network emulator, Power TOSSIM [18]. It is an extension to TOSSIM, a scalable simulation environment for wireless sensor networks, that provides an accurate, estimation of node power consumption. Power TOSSIM can capture the detailed, low-level energy requirements of the CPU, radio, sensors, and other peripherals based on the Mica2/Mica2dot/MicaZ sensor node platform of Crossbow Technology Inc [19]. Our implementation is based on Power/Radio Model of Mica2 in Power TOSSIM and the operating frequency is 915 MHz

4.2 Simulation and Results

To evaluate the effect of topology-based routing for Xmesh, we compare the performance of three different routing strategies: *original Xmesh*, *Xmesh based on Cayley graphs and shortest path routing*, and *Xmesh based on Cayley graphs but*

with random selection of routes. For the original Xmesh, a packet is forwarded to intermediate nodes with the best link quality. For Xmesh based on Cayley graphs, a Cayley graph is the virtual topology of nodes in the network, packets are forwarded to intermediate nodes that is part of the shortest path between the source and the destination. In the event that multiple shortest paths exist, the best quality link will be chosen. For Xmesh based on Cayley graphs and random selection of routes, a Cayley graph is still the virtual topology of the networks, but packets are randomly forwarded to one of the neighbors of the virtual topology.

As an example, for a 21-node network, a packet is being sent from source node 0 to node 9. Using the original Xmesh routing strategy, the packet will be forwarded to intermediate nodes with the best link quality from source node 0. Using the Xmesh based on the 21-node Cayley graph (Fig. 2), the packet will be sent to either node 3 or node 4, depending on the link quality between nodes 0 and 3, and nodes 0 and 4. Using the Xmesh based on random selection of routes, the packet will be randomly forwarded to any one of node 0's neighbors, i.e., nodes 3, 4, 18, or 11. We expect that, in general, messages routed according to Xmesh based on Cayley with shortest path routing to have the shortest path length, while that of random routing will impose longer path lengths.

These three routing strategies were simulated via Power TOSSIM for networks of sizes 21, 55, 110, 253, and 465 nodes. The simulation scenario is that each node continuously sends packets to the base station (node 0) at the transmission rate of 38.4 kbps with the packet size of 46 bytes. These numbers are used according to the specification of MICA2 motes by Crossbow Technology Inc.

By comparing these three strategies, we can see the effect of imposing a virtual topology on Xmesh. We expect the strategy of Xmesh based on Cayley with shortest path routing to have shorter path lengths than that of the original Xmesh. By imposing the Cayley topology with a random routing strategy, path lengths generated will be longer than the original than the original Xmesh.

All simulation were executed for the same amount of virtual/simulated seconds and with the same random seed and with the same random location deployment for networks of the same size regardless of their routing strategies. The radio model in Power TOSSIM sets the bit error rate between motes according to their location and various models of radio connectivity. We used the CC1000's "Empirical" radio model.

4.3 Power Consumption Analysis

To compare the power consumption required for the different strategies, Fig. 3 plots the averaged power consumption versus different network sizes. The averaged power consumption is the average of the total power consumption for all nodes in the network. Among the three strategies, the *random* routing strategy consumed the most power, while the topology-based (*Cayley*) on X-mesh consumed the least power. This is expected as the averaged path lengths for the *Cayley*

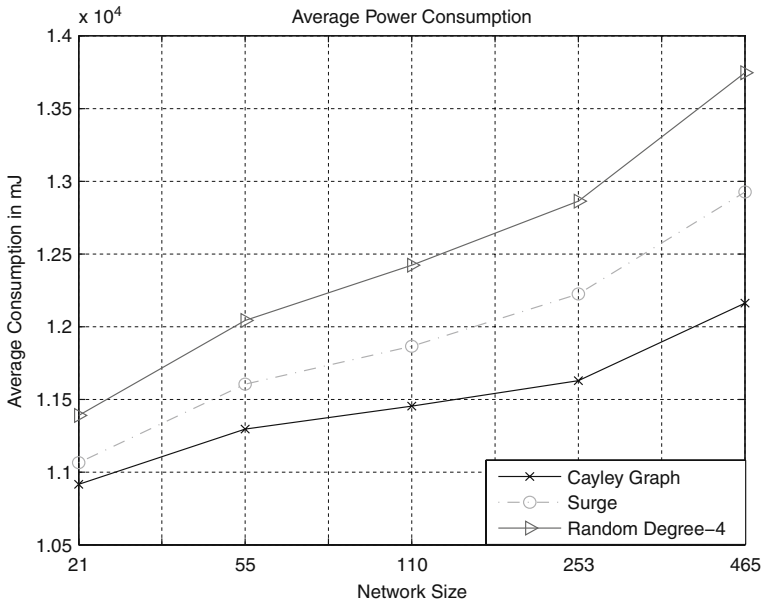


Fig. 3 Average power consumption for different routing strategies

	21-node	55-node	110-node	253-node	465-node
Cayley	2.65	3.01	3.95	5.84	6.91
Surge	2.81	3.55	4.53	6.27	7.46
Random	2.93	3.84	4.98	6.93	8.04

strategy is the shortest and that of the *random* (Cayley based with random routing) is the longest.

As expected, the averaged power consumed grows with the size of the network. Furthermore, the differences among the three strategies also grow with the size of network. For the small size network with 21-node, Cayley and Surge are about the same. For the largest size network, 465-node, Cayley’s averaged power consumed is about 93% that of the original Xmesh (*Surge*). The difference is a consequence of the path lengths of the three routing strategies. Figure 4 plots the averaged hop counts among the three strategies for the different size networks. Indeed, the topology-based with random routing (*Random*) has the largest hop count, whereas the Cayley with shortest path routing (*Cayley*) has the shortest hop count. Furthermore, such difference grows with the size of the networks.

To further analyze the power consumption among the three different strategies, Fig. 5 plots the histogram of the energy consumed for the 55-node network. From the histogram, we observe that the Cayley strategy has a wider energy distribution than that of the original Xmesh (*Surge*) and the *random* case. For Cayley, the range is from 10,000 to 13,000 mJ, whereas the original Xmesh has a narrow peak around

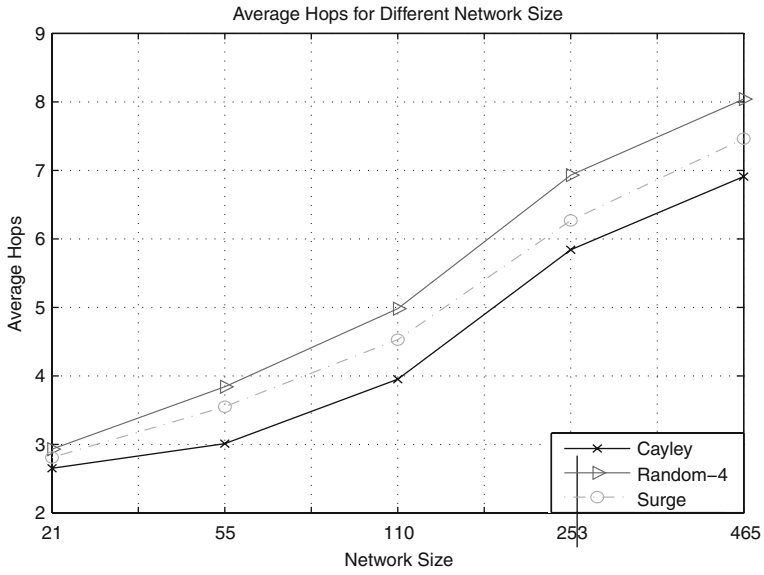


Fig. 4 Averaged hop count vs. network size

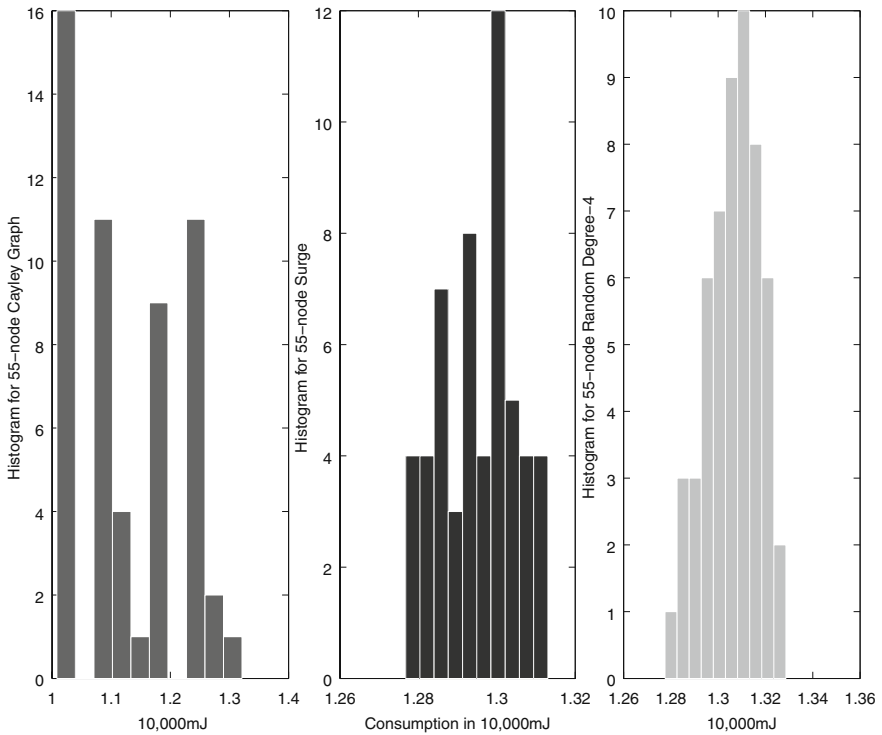


Fig. 5 Histogram of power consumption for 55-node network

13,000 mJ. As for the topology-based with random routing (Random), the range of the power consumption is between 12,700 and 13,300 mJ. Based on this result, we can conclude that the Cayley strategy conserved power but impose a wider energy distribution among nodes in the network. This implies that some nodes of the network may “die” first before others. To further investigate on this point, we computed the normalized standard deviation of energy consumption among nodes in a network for the three different routing strategies. Indeed, we found that the standard deviation for the topology-based routing with shortest path (*Cayley*) has a standard of deviation between 6 and 7% for different network sizes, whereas those of the Xmesh (*Surge*) and random (*Random*) routing is about 2%.

5 Conclusions

In this chapter, we have proposed a topology-based routing for Xmesh that combines the dense and vertex-transitive property of Cayley graphs. The dense property of Cayley graphs implies that path lengths are shorter and that the vertex-transitive property allows the same routing strategy and routing table be used at every node. Through the Power TOSSIM emulator, we implemented our proposed protocol for the Mica2 motes from Crossbow Technology Inc. [19]. Our simulation result for network size ranges from 21-node to about 500-node showed that the proposed topology-based routing consumes less power than the original Xmesh strategy. Furthermore, this power saving advantage of the proposed protocol increases with the network size.

References

1. J.N. Al-Karaki and A.E. Kamal, “Routing Techniques in Wireless Sensor Networks: A Survey”, *IEEE Wireless Communications*, 11, 6, December 2004, 6–28.
2. I. Akyildiz et al., “A Survey on Sensor Networks”, *IEEE Commun. Mag.*, 40, 8, Aug. 2002, 102C14.
3. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks”, *Proc. 33rd Hawaii Intl. Conf. Sys. Sci.*, Jan. 2000.
4. J.M. Hellerstein, W. Hong, and S.R. Madden, “The Sensor Spectrum: Technology, Trends, and Requirements”, *ACM SIGMOD* 32, 4, December 2003, 22–27.
5. E. Noel and K.W. Tang, “Novel Sensor MAC Protocol Applied to Cayley and Manhattan”, *IEEE International Workshop on Wireless Ad-hoc and Sensor Networks*, 2006
6. L. Wang, E. Noel, C. Fong, R. Kamoua, and K.W. Tang, “A Wireless Sensor System for Biopotential Recording in the Treatment of Sleep Apnea Disorder”, *IEEE International Conference On Networking, Sensing and Control*, 2006.
7. A. Woo, T. Tong, and D. Culler, “Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks”, *Sen-Sys03*, November 5C7, 2003, Los Angeles, California, USA.
8. A. Woo and D. Culler, “Evaluation of efficient link reliability estimators for low-power wireless networks”, *Technical Report UCB//CSD-03-1270*, U.C. Berkeley Computer Science Division, September 2003.

9. B.W. Arden and K.W. Tang, "Representations and Routing for Cayley Graphs", *IEEE Transactions on Communications*, 39, 11, November 1991, 1533–1537.
10. K.W. Tang and B.W. Arden, "Vertex-Transitivity and Routing for Cayley Graphs in GCR Representations", in the 1992 Symposium on Applied Computing, pp. 1180–1187, March 1–3, 1992, Kansas City, MO.
11. K.W. Tang, "Dense, Symmetric Interconnection Networks", Ph.D. Thesis, University of Rochester, 1991.
12. K.W. Tang and B.W. Arden, "Representations of Borel Cayley Graphs", *SIAM Journal on Discrete Mathematics*, 6, 4, November 1993, 655–676.
13. D.V. Chudnovsky, G.V. Chudnovsky, and M.M. Denneau, "Regular graphs with small diameter as models for interconnection networks", Tech. Rep. RC 13484–60281, IBM Res. Division, Feb. 1988.
14. C.L. Liu, "Shortest Path In Weighted Graphs", In *Elements Of Discrete Mathematics*, 2nd Edition, p. 147, McGraw-Hill, New York, 1998
15. D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks" In T. Imielinski and H.F. Korth, Eds., *Mobile Computing*, vol. 353, Kluwer Academic Publishers, Dordrecht, 1996.
16. C. Ming-Syan and K.G. Shin, "Depth-First Search Approach for Fault-Tolerant Routing in Hypercube Multicomputers", *IEEE Transactions On Parallel and Distributed Systems*, 1, 2, April 1990, 152–159.
17. J. Silvela and J. Portillo, "Breadth-First Search and Its Application to Image Processing Problems", *IEEE Transactions on Image Processing*, 10, 8, August 2001, 1194–1199.
18. V. Shnayder, M. Hempstead, C.Bor-rong, and M. Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications", *SenSys'04*, November 3–5, 2004.
19. Crossbow Technology Inc's homepage, <http://www.xbow.com>, accessed in August 2007.
20. Crossbow, "Hardware Framework for Sensor Networks", In Presentation of Day one in Crossbow Seminar, Towson, December 2005.
21. Crossbow, "Multihop-Mesh-Network", In Presentation of Day one in Crossbow Seminar, Towson, December, 2005.
22. MoteIV Website, <http://www.moteiv.com/>, accessed in August 2007.
23. R. Fonseca, S. Ratnasamy et al., "Beacon-Vector Routing: Scalable Point-to-Point Routing in Wireless Sensor Networks", In *Proceedings of NSDI 2005*.
24. MICA2 Radio Stack for TinyOS, <http://www.tinyos.net/tinyos-1.x/tos/platform/pc/CC1000Radio/CC1000RadioIntM.nc>