The Cayley Graph Implementation in TinyOS for Dense Wireless Sensor Networks

Lei Wang, K. Wendy Tang

Abstract— **Wireless Sensor Network (WSN) consists of a group of sensors or nodes, linked by a wireless medium (infrared or radio frequency) to perform distributed sensing tasks. The dynamic and lossy nature of wireless communication poses major challenges for highly dense WSN. In our previous work, we proposed a new networking topology, the Cayley Pseudo-Random Protocol [1] [2] for large wireless sensor networks. In this paper, we focus on the implementation of Cayley Graph on Crossbow Technology Inc's sensor nodes [3]. The Cayley Graph topology is implemented via the TinyOS emulator, Power Tossim [4]. The performance of the network is evaluated in terms of energy consumption, network lifetime and fairness. Comparison is made with the Xmesh protocol and showed that our Cayley Graph Implementation consumes less power but trade off with fairness in a relatively small amount.**

I. Introduction

Wireless Sensor Networks (WSN) have emerged as a new information-gathering platform with a large number of self-organized sensing nodes. These networks can be used in many environments such as intelligent battlefields, smart hospitals, earthquake response systems, and learning environments. In most applications, energy supply and communication bandwidth are constrained for sensor nodes [5]. Therefore in order to shorten network lifetime and efficiently use the limited bandwidth, energy efficiencies need to be improved. Such constraints challenge researchers to design and manage WSNs with energy-awareness at all layers, especially for a typical deployment of a large scale sensor network [6]. 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, network dynamics etc. The dynamic and lossy nature of wireless communication poses major challenges to reliable, self-organizing multihop networks. Especially for dense WSN with a few hundred nodes, the energy conservation, scalability and self-configuration are primary goals [7] [8], while per-node fairness and protocol simplicity are less important.

When designing routing algorithms for large-scale WSN, other factors potentially interact with routing, such as realistic connectivity of nodes [9] [10]. For an actual sensor network, the connectivity graph should be discovered by sharing local communication quality measurements. A nearby node could have the better communication link, but due to multipath, collision, congestion or other realistic factors, it is not guaranteed. Thus when designing routing algorithms for large-scale WSNs, the communication quality needs to be taken into account since geographically proximate nodes may not produce optimal routes.

Inspired by our own protocol - Cayley Pseudo-Random Protocol [2] and our implementation [11] for a Wireless System in Treatment of Sleep Apnea, we propose and implement our Cayley Graph into a single-transceiver platform [12], Xbow's Mica2 [3]. Also inspired by the self-organized protocol-Xmesh [9] [13], which has Link Estimator to evaluate motes' realistic connectivity likelihood, we combined our highly scalable Cayley Graph Topology [1] with Xbow's Mica2 Xmesh. We overlayed a degree 4 Cayley graph to Xmesh routing algorithm so that selection of routes with quality measured by Xmesh Link Estimator is based on the underlying Cayley graph. We implemented this topology in TinyOS's emulator, PowerTossim [4]. For benchmark, we also propose a simplified protocol based on Cayley Graph, Random Degree-4. Random Degree-4 does not search the shortest path for communication, instead nodes just select their neighbors in Cayley Graph topology randomly. We can imagine there will be more collision and longer paths to Base Station, therefore more energy consumption.

In this paper, Section II describes the routing algorithm for Cayley Graph on single-transceiver platform; Section III covers the design and implementation of Cayley Graph in TinyOS's emulator, PowerTossim; Section IV provides simulation results, analysis and discussion whereas conclusion is included in Section V.

II. The Routing Algorithm for Cayley Graph on Single-Transceiver Platform

A. Cayley Graph Overview

Symmetric, regular, undirected graphs are useful models for the interconnection of multicomputer systems. Dense graphs of this sort are particularly attractive. Based on group theoretic constructions, Cayley Graph [14] are in this category of graphs [1]. 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 $\Longleftrightarrow$ $v_1 = v_2 * g$ 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 paper, we are interested in undirected, degree-4 Cayley graphs. In other words, we are dealing with Cayley graphs whose generator set consists of two group elements and their inverses. In

Lei Wang,K. Wendy Tang are with Electrical and Computer Engineering Department, Stony Brook University, Stony Brook, NY 11794-2350, Email: leiwang,wtang@ece.sunysb.edu

1-4244-0697-8/07/.00 ©2007 IEEE.

Cayley graphs, vertex is transitable and vertex density is high. 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 very useful for routing. It means that a Cayley graph "looks the same from any node", which maps path-searching between two arbitrary vertices to a path already known from a fixed vertex. In other words, routing between vertices i and j can be determined by finding paths between 0 and j. This property is the basis for a distributed routing algorithm, the vertex transitive routing in. Figure 1 is an example of a 21-node, degree-4, Borel Cayley graph in the integer domain. The graph has $\mathbf{V} = \{0, 1, ..., 20\}$, and the connection are defined as [1]:

Let $\mathbf{V} = \{0, 1, ..., 20\}$. For any $i \in V$, if i mod 3 = :

0: i is connected to i+3, i-3, i+4, i-10; mod 21

1: i is connected to i+6, i-6, i+7, i-4; mod 21

2: i is connected to i+9, i-9, i+10, i-7; mod 21

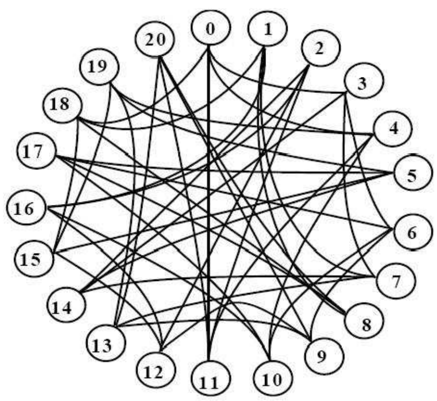

Fig. 1. 21 Node Borel Cayley Graph

B. *Shortest Path Algorithm for Cayley Graph*

In order to implement Cayley Graph into Xbow's motes, we developed a new algorithm of path-searching to find the shortest path between any two vertices. In [2], each vertex has 4 degree, 2 for incoming communication and 2 for outgoing communication. In this paper, the 2 incoming and 2 outgoing channels merge into one channel for single half-duplex transceiver. Therefore multiple choices of shortest-path selecting in terms of **hops** (the concern of realistic connectivity/communication cost is taken into account by embedding Link Estimator, see Section III) exist. To better understand this, see Figure 2: Tree View with node 0 as root. Clearly, for instance, from Cayley graph node 9 has four connected neighbors: node 6, 12, 13, 20, among which node 6 and node 20 have the shorter paths than node 12 and 13. Thus for node 9, two possible and equivalent "shortest paths" (in terms of hops) do exist.

In our Cayley graph, each node has 4 degree connection (link $\alpha, \beta, \gamma, \theta$), and not any information of communication cost is provided before this Cayley graph is implemented in real Sensor Networks. Therefore for implementation,

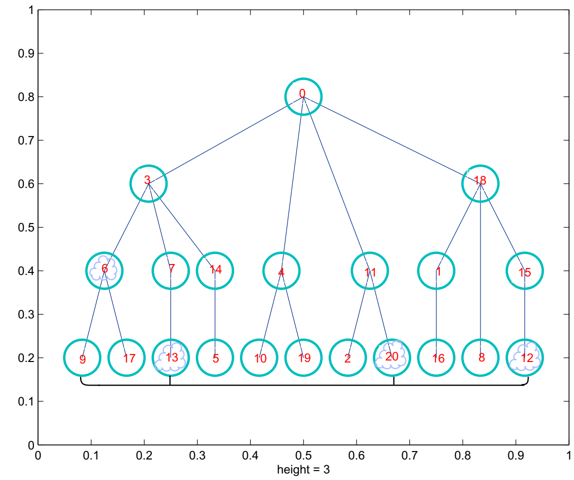

Fig. 2. Tree View with node 0 as root

typical Dijkstra algorithm faces the problem of equivalent path selection and falls into the category of breadth or depth search algorithms. Also the complexity of Dijkstra algorithm is $O(n^2)$, in Sensor Networks it could consume more energy of CPU and scales poorly to large networks if we leave motes to find the shortest path instead of pre-loading the routing table in this paper (for simplicity). Realizing the above factors and inspired by "On-Demand Route Discovery" by Johnson [16], we chose our own algorithm combined with Depth-First-Search [17] and Breadth-First-Search algorithm [18]. Below is the description of this algorithm.

- **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 j1/j2/j3/j4 is not visited through this link $\alpha/\beta/\gamma/\theta$, in other words, distance = infinity, and j1/j2/j3/j4 is not root of this tree, go to the following procedure:
     a) Update the distance and path information for node j1/j2/j3/j4;
     b) Find the next four connected nodes of node j4, flag this link $\alpha/\beta/\gamma/\theta$, 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 [2], generate the routing table for other vertices.

The complexity of this algorithm is $O(16log_4(n))$, which ensures a better searching efficiency. We searched the "shortest path" for 21-node, 55-node, 110-node 253-node and 465-node Cayley graph, for the densest 465-node graph, it only took around 1 second to finish the searching on a Dell GX620 desktop.

III. Implementation of Cayley graph in TOSSIM

We demonstrate the effectiveness and measure the performance of our protocol in a testbed emulator, which can be considered as a real testbed (See Section III.A), for wireless sensor network of motes developed by University of California, Berkeley and commercialized by Xbow [3], MoteIV [19], Inc. The mote is the hardware platform which consists of Processor/Radio boards (MPR, Mote Processor Radio) and has a 8-bit Atmel AT90LS8535 microcontroller running at 4 MHz. It has a low power radio transceiver module CC1000 which operates at 916.5/433/315 MHz and provides a transmission rate of 19.2 Kbps.

The mote runs on a small event-driven operating system called TinyOS. The TinyOS operating system is open-source, extendable, and scalable. The TinyOS system, libraries, and applications are written in nesC, a new language for programming structured component-based applications and primarily intended for embedded systems such as sensor networks.

A. The Cayley Graph Implementation

Our Cayley Graph Implementation can support large scale networks, but implementing large scale WSN for real motes is limited by budget and space factors. Therefore, our approach is to simulate the Cayley Graph Implementation in the TinyOS sensor network emulator, Power TOSSIM. Implementation for a few motes could be carried out after simulation in Power TOSSIM, for test purpose. PowerTOSSIM [4], an extension to TOSSIM, is a scalable simulation environment for wireless sensor networks and 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. Our implementation is based on Power/Radio Model of Mica2 in Power TOSSIM and the operating frequency is 915 MHz.

1) Introduction to Xmesh: The multihop routing protocol , Xmesh [9], is implemented in a TinyOS's application called Surge. Base on Surge, we integrate our routing algorithm into it. Xmesh is a distributed routing process that has three local processes: link quality estimation, neighborhood management, and connectivity-based route selections (Figure 3). Link quality of each node's neighbors is characterized by the component of Link Estimator. 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 run periodically to select one of the potential neighbors for routing. The route messages has fields of 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

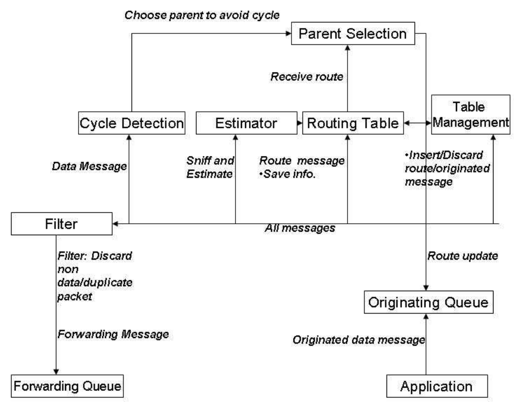

Fig. 3. Message flow in Xmesh

In Xmesh, if a node is triggered by some sensing event, it will send the message to the sink for reporting and this message will be put into the originating FIFO queue. If a node get a message from its children, it will check its table periodically to decide its parent and route information according to link estimation and cost calculation and put this message into forwarding queue. All messages go to Table management component first, then in Routing Table node's neighbors information will be provided in order to estimated in Estimator. Component of Cycle Detection protect cycle forming. Aftermath, component of Parent Selection decide node's parent and the message will be put into Forwarding queue.

2) Overlaying Cayley Graph with Xmesh: Our program reads the routing table from a local file (before compiling), and selects the next neighbor/parent node when the TX buffer is ready for this local node. In our routing table, "-1" indicates the corresponding link is not to the shortest path. We first generated the routing table for vertex 0 as root and routing tables for other vertices are derived by vertex transitive formula given by [1].

Each node does not necessarily store the whole routing table since it only needs its own parents information. For example, if node 11 needs to send message to Base, node 0, node 11 just stores the first entry (the possible parents to reach node 0) in its corresponding routing table. This concern is critical especially for large scale sensor networks since motes have limited memory size, 4 KB for Mica2 [12]and 10 KB for Tmote [19].

According to Cayley graph topology, some non-leaf nodes will undertake more forwarding traffic through themselves, while leaf nodes only forward less traffic. This will lead to low fairness (See Section IV). Future improvement is also covered in Section V. When the local node receives a message from other nodes, which means RX buffer has something for this node, it forwards this message to its parent. This receiving/forwarding behavior is implemented by modifying the radio stack of this node.

B. Integration of Link Estimator for Cayley Graph

As described above, in Cayley Graph topology for single half-duplex transceiver, some nodes have multiple choices of selecting parents. Inspired by the concept of Link Estimation [9] [10], we adopt Link Estimator Interface in the Multihop Routing protocol for sensor networks called "Reliable Route" or "XMesh" [13] on MAC layer, designed to satisfy the dynamic and lossy characteristics of WSN. As we know, communication cost is an abstract measure of distance. Hop count, transmission and retries and reconfigurations over time can be the cost. Xmesh defines Minimum Transmission(MT) [9] as cost:

- MT cost = $\frac{1}{linkquality_{forward}} times \frac{1}{linkquality_{backward}}$

This kind of quality is evaluated in terms of Received Signal Strength Indicator (RSSI) and therefore ensures the realistic communication factors to be taken account into. In our Cayley graph, nodes with multiple parents selection choose their parents according to parents' real connection cost to Base Station. To compute link quality, in TinyOS a node snoops [10] on the packets sent by each neighbor, and checks the addresses. A node determines the link quality to a neighbor by monitoring the ratio of packets received from that neighbor to the number of packets sent by that neighbor. In our implementation, we only compare the link qualities of potential parents by integrating Link Estimation interface [20] into our configuration. A higher return value (ranged in [0,255]) from LinkEstimator.getQuality() implies that the link to the parent is estimated to be of a higher quality than the one that results in a smaller return value.

C. Random Degree-4 Implementation

In order to have a better scope how our implementation works in Power TOSSIM and a wider testing scope, we also propose the very simplified version of our protocol: Random Degree-4 protocol. In Random Degree-4, each node does not need a routing table for its own, and it simply take any of its four parents randomly according to Cayley graph connectivity. For example, in 21-node network, node 9 has four neighbors-node 6, 12, 13 and 20, and it just takes one of them randomly as parent. The concern of creating loops is solved automatically since Surge program takes care of it with a component so called Cycle detection [10].

IV. Simulation Results and Analysis

A. Simulation and Results

In order to scale the network size for our Cayley graph protocol and evaluate the performance of each network, 21-node, 55-node and 110-node network were simulated in Power TOSSIM. Due to limited computing ability, larger networks were not simulated. In future, we will seek the methods to release the simulator memory stack and speed up our simulation for larger network.

All simulations of Cayley Graph/Random Degree-4/Surge were executed for 737 virtual/simulated seconds and with the same random seed to ensure the same booting sequence. Also in our simulation, same random location deployment schemes were applied to the specific sized network, for example, for 110-node, all three applications of Cayley Graph/Random Degree-4/Surge had the same physical deployment. 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 (bit error rate between motes according to their location and empirical models of radio connectivity) which is based on an outdoor trace of packet connectivity. The duty cycle, which determine the percentage of time of periodic sleeping, listening, transmitting and receiving, were set with same default value-mode 0 (see Section B).

To evaluate Power Consumption, we used Power Profiling which provides direct view of power consumption for each application and for each hardware component. The power consumption of Radio, CPU, LED, EEPROM and Total were recorded into a matrix file for analysis.

B. Power Consumption Analysis

1) With Full Duty Cycle: Power Consumption of each component for different simulation are in the unit of Mili-Joules. For comparison purpose, the Power Profiling information of different applications were plotted in the same figure for same specific sized network. For instance, Figure 4 shows the power consumption of Cayley Graph/Surge/Random Degree-4 with 55 nodes. The different bars represent consumption of different components, and the whole bars mean the total consumption. For 21-node and 110-node network, we had the similar plots. From Figure 4, one can tell that Surge and Random Degree-4 distribute energy with a almost uniform manner, whereas Cayley Graph obviously assigns the consumption of each node with relatively wider difference.

To visually show the effects of network size and different applications, we calculated the average total consumption of each application for different network size and plotted into Figure 5. From this figure, it turns out that:

- As networks grow larger the total power consumption of each application will also increase with a certain amount;
- Compared with Random Degree-4 and Surge, Cayley Graph consumes least power in average.

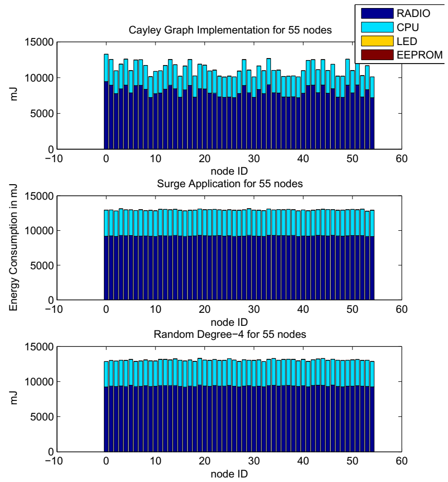

Fig. 4. Power Consumption for 55-node network

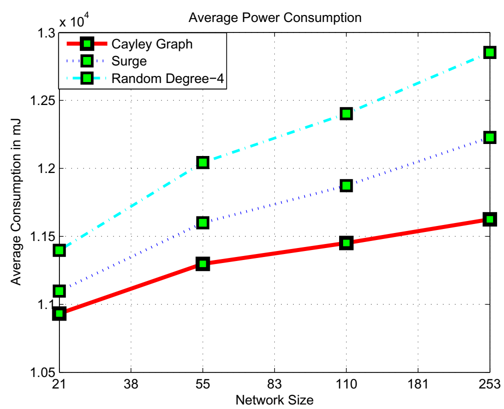

Fig. 5. Average of Total Consumption For Different Applications

2) With Various Duty Cycle: In order to save power consumption for applications, CrossBow motes use duty cycles to conduct the radio operations: sleep, initialize radio, radio crystal start-up, receive/transmit, sample, sleep [13]. TinyOS provides functions called SetListeningMode()and SetTransmitMode() to achieves Low Power Listening and different duty cycles. Basically there are four duty cycle modes: 0, 1, 2, 3 $(100\%, 35.5\%, 11.5\%, 7.5\%)$ [21]. We integrated the above functions into our Cayley Graph to change duty cycle modes and simulated our applications with different network sizes. Clearly with larger duty cycle mode applications consume less energy. Figure 6 shows the duty cycle effect on WSN with 55-node for all applications.

C. Statistic Results

Histogram of total power consumption were also plotted and analyzed in order to have the view in statistic domain. Simulations show that Cayley Graph has a more diversified energy distribution than Random Degree-4 and Surge for all sized network. Figure7 is the histogram for 55-node network. From this figure, one can see: energy is distributed in a wider range of [10.26 13.1]J in Cayley Graph simulation,

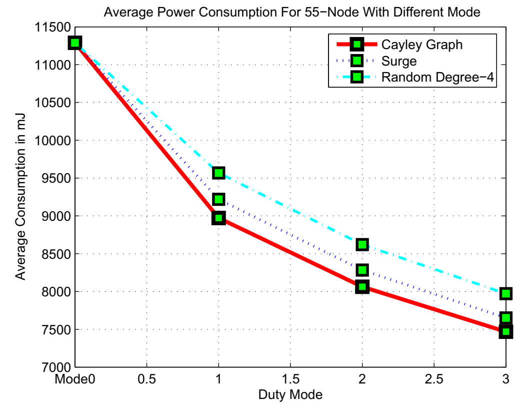

Fig. 6. Average of Total Consumption For 55-Node With Different Mode

4 histogram bars with node number above 8 are dispersed in most of this range and rest of bars with less node number are filled in between; while energy consumption of nodes only differs in range of [12.8 13.13]J and [12.9 13.25]J in Surge and Random Degree-4 simulation.

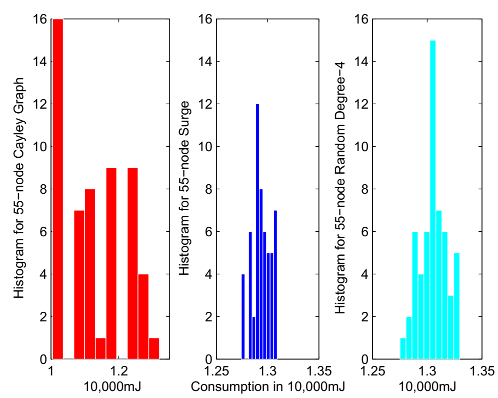

Fig. 7. Histogram of Energy for 55-node network

Standard deviation of each simulation was evaluated and compared in the same plot. From Figure 8, one can see that for Cayley Graph has the largest normalized standard deviation (standard deviation over corresponding average value) for all of component consumption. The normalized standard deviation for Cayley Graph falls into the range of $[6\%, 7\%]$, whereas Random Degree-4 and Surge/Xmesh only oscillate around 2%. Thus we can draw the conclusion that Cayley Graph consumes less energy on the average but trade off with fairness. This variability in power consumption can make a few nodes in Cayley Graph run out of power earlier than other nodes. New techniques to deflect some traffic through these nodes are required.

Duty cycle also has effects on the standard deviation of power consumption. Figure 9 shows with radio always on (mode 0), the deviation difference between Cayley Graph and Random-4/Surge is the largest; when other duty cycles are applied to applications, collisions/backoff/retransmit scheme

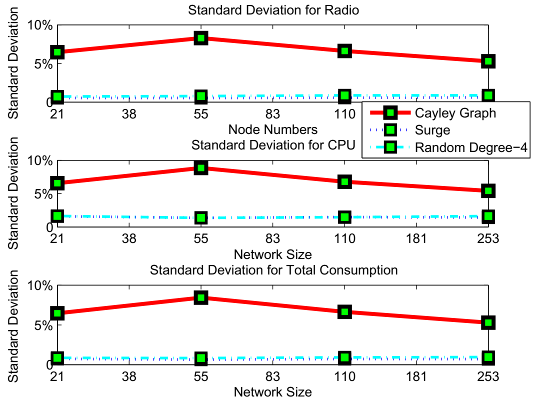

Fig. 8. Normalized Standard Deviation for Radio, CPU, Total Consumption

causes Surge/Random-4 nodes can not behave fairly as mode 0. At this point, Cayley Graph Implementation will trade off less fairness, which is promising.

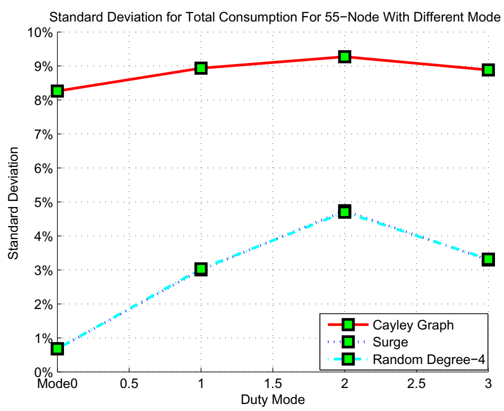

Fig. 9. Standard Deviation for Total Consumption For 55-Node With Different Mode

D. Discussion and Analysis

By looking into the tree view of Cayley Graph topology for single half-duplex transceiver, the above simulation and analytical results can be understood: in graph domain, Cayley Graph treats nodes differently, in other words, some non-leaf nodes will undertake more forwarding traffic through themselves, while leaf nodes only forward less traffic; Surge and Random Degree-4 do not "discriminate" nodes and therefore distribute energy almost uniformly; Cayley Graph ensures very few hops to reach the Base Station, the average hops are 2.1, 2.89, 3,74, 5.39 for 21/55/110/253-node network respectively; Cayler Graph has the less average power consumption (1.3%/2.8%/3.6%/5.2% less than Surge/Xmesh). In Figure 9, by fitting the power-saving data, we expect to achieve 6% power consumption saving for 500-node network relative to Surge's consumption.

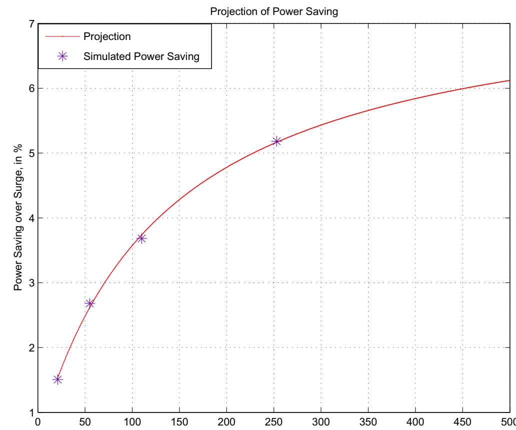

Fig. 10. Expected Consumption Improvement for larger network

V. Conclusion

In Cayley graph, each node has 4 degree connection (link $\alpha, \beta, \gamma, \theta$), shortest paths in terms of hops exist. To reach to the Base Station in sensor networks, this Cayley Graph Implementation on Single Half-Duplex transceiver platform has multiple shortest paths, which leads to the multiple parent selection. By adopting the Link Estimator Interface in TinyOS, our implementation not only takes advantage of Cayley Graph topology, but also utilizes the realistic communication cost (Minimum Transmission) to satisfy the dynamic and lossy characteristics of Wireless Sensor Networks. Our implementation consumes less power, but at the same time the fairness is trade off with a relatively mediocre amount (with duty cycles less than 100% the amount will be relatively smaller). In the future research, scaling network size beyond 400-node and achieving a good value of fairness by dynamically changing positions of those heavy-traffic nodes in Cayley Graph, are our primary goals.

VI. Acknowledgement

This project is supported by the Sensor Consortium in Stony Brook University and is funded by the National Science Foundation under Grant No. EEC-0332605. The goal of the consortium is to promote and increase awareness of entrepreneurship and technology transfer activities on Long Island with a focus on national security and medical sensor systems. Any opinions, findings and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Graphs Bruce W. Arden and K. Wendy Tang, Representations and Routing for Cayley Graphs", IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 39, NO. 11, NOVEMBER 1991
[2] Eric Noel and K. Wendy Tang, Novel Sensor MAC Protocol Applied to Cayley and Manhattan", IEEE International Workshop on Wireless Ad-hoc and Sensor Networks,2006
[3] Crossbowweb Web, http://www.xbow.com

[4] Victor Shnayder, Mark Hempstead, Bor-rong Chen and Matt Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications", UNDER SUBMISSION
[5] JAMAL N. AL-KARAKI, AHMED E. KAMAL "ROUTING TECHNIQUES IN WIRELESS SENSOR NETWORKS: A SURVEY", IEEE Wireless Communications, December 2004
[6] I. Akyildiz et al., "A Survey on Sensor Networks", IEEE Commun. Mag., vol. 40, no. 8, Aug. 2002, pp. 102C14.
[7] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", Proc. 33rd Hawaii Intl. Conf. Sys. Sci., Jan. 2000.
[8] Joseph M. Hellerstein, Wei Hong, Samuel R. Madden, "The Sensor Spectrum: Technology, Trends, and Requirements", ACM SIGMOD Volume 32, Issue 4, December 2003
[9] Alec Woo, Terence Tong, David Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks", SenSys03,November 5C7, 2003, Los Angeles, California, USA.
[10] 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.
[11] Lei Wang, Eric Noel, Cheung Fong, Ridha Kamoua and K. Wendy Tang, "A Wireless Sensor System for Biopotential Recording in the Treatment of Sleep Apnea Disorder", IEEE International Conference On Networking, Sensing and Control,2006
[12] Crossbow, "Hardware Framework for Sensor Networks", In Presentation of Day one in Crossbow Seminar, Towson,December, 2005.
[13] Crossbow, "Multihop-Mesh-Network", In Presentation of Day one in Crossbow Seminar, Towson,December, 2005.
[14] 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.
[15] C.L.Liu,"Shortest Path In Weighted Graphs", In Elements Of Discrete Mathematics, Second Edition, page-147, McGraw-Hill, 1998
[16] JOHNSON, D. B., AND MALTZ, D. A. , "Dynamic source routing in ad hoc wireless networks", In Mobile Computing, Imielinski and Korth, Eds., vol. 353. Kluwer Academic Publishers, 1996.
[17] Ming-Syan Chen and Kang G. Shin, "Depth-First Search Approach for Fault-Tolerant Routing in Hypercube Multicomputers", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. I, NO. 2, APRIL 1990
[18] Jaime Silvela and Javier Portillo, "Breadth-First Search and Its Application to Image Processing Problems", IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 10, NO. 8, AUGUST 2001
[19] MoteIV Website, http://www.moteiv.com/
[20] Rodrigo Fonseca, Sylvia Ratnasamy, et al., "Beacon-Vector Routing: Scalable Point-to-Point Routing in Wireless Sensor Networks",In Proceedings of NSDI 2005.
[21] MICA2 Radio Stack for TinyOS, http://www.tinyos.net/tinyos-1.x/tos/platform/pc/CC1000Radio/CC1000RadioIntM.nc