Novel Sensor MAC Protocol Applied to Cayley and Manhattan Street Networks with CrossBow MICA2

Eric Noel
AT&T Labs Research
Middletown, NJ
Email: eric.noel@att.com

Wendy Tang
Department of Electrical and Computer Engineering
SUNY Stony Brook, NY
Email: wtang@ee.sunysb.edu

Abstract— **With the rapid growth of wireless sensor technology, we foresee a need for MAC protocols to allow efficient simultaneous peer-to-peer communications in large and dense wireless sensor networks. To meet such a need, we propose a novel MAC protocol that exploits network graphs property to assign frequencies such that the number of intermediate hops to reach destinations is minimized. Via a simulation model parameterized to CrossBow MICA2, we evaluate the performance of our MAC protocol with Cayley and Manhattan Street networks.**

I. INTRODUCTION

The recent development of small and affordable microsensors that can communicate with each other via radio transceivers have resulted in the rapid growth of wireless sensor networks[1], [2], [3]. When deployed in large numbers, they provide unprecedented opportunities for monitoring applications such as real-time traffic monitoring, nuclear factory surveillance, military sensing and reconnaissance, disaster relief networks, wildfire detection, wildlife tracking, ... [4].

We propose a novel MAC protocol to allow efficient simultaneous peer-to-peer communications in large and dense wireless sensor networks. The proposed MAC protocol is inspired by SMACS[5]. But instead of assigning frequencies uniformly at random, we propose to exploit network graphs property to assign frequencies such that the number of intermediate hops to reach destinations is minimized. The proposed MAC protocol is designed to address "dense" sensor networks.

In the following, we present our novel MAC protocol (Section II), introduce the two evaluation networks (Section III), overview the simulation model used to evaluate our proposal (Section IV), present the simulation results (Section V) and summarize findings in the conclusion (Section VI).

II. NOVEL MAC PROTOCOL

Before deployment, each sensor is programmed to receive and transmit via a small set of dedicated frequency channels following the connection pattern of the underlying graph (either Cayley[6] or Manhattan Street Network[7]). A routing table that lists all optimal outgoing links (two outgoing links for the networks considered here) for routes between node 0 and all other nodes in the network is generated and loaded to each of the sensor nodes (only graphs with such routing property are considered). Sensors being pre-programmed with the topology there is no direct relationship between logically adjacent and geographically nearby sensors.

Because there is only one single half-duplex transceiver on each sensor (CrossBow MICA2), the multiple channels have to be used in a "distributed time-division manner". The proposed protocol borrows the well known RTS (Request to Send), CTS (Consent to Send), and ACK (Acknowledgement) concept of the IEEE 802.11 DCF protocol.

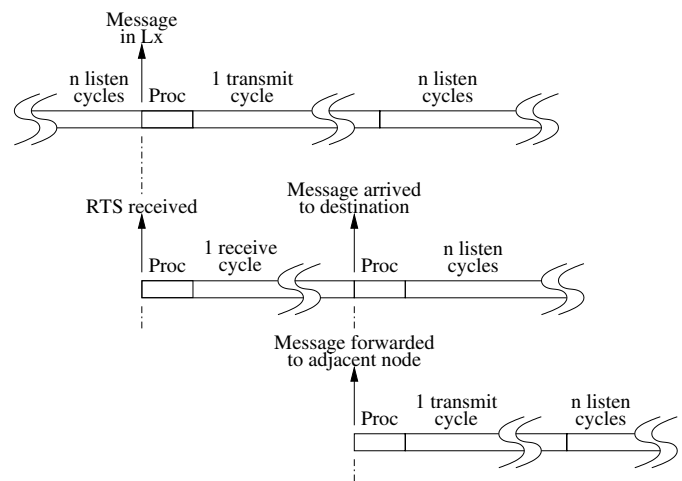Each sensor node evolves within a Transmit, Receive, Listen and Sleep cycle (see Figure 1).

Fig. 1. Medium access control cycles.

A sensor node switches to the Transmit cycle whenever there is data to send, provided that it is not busy receiving or processing data. If it is busy receiving some data from a neighboring node, it will wait until the data has been completely received and an ACK has been sent out before switching to the Transmit cycle. The Transmit cycle is completed once the sensor receives an ACK from the destination sensor.

During the Receive cycle, a sensor node receives data from its designated frequency neighbor (see Figure 2 for details). Once the data message is received, an ACK control message is sent to the source sensor.

During the Listen cycle, a sensor tunes its antenna to each adjacent sensor for a fix time period (τ). The sensor cycles through each adjacent sensor (D sensors) until it receives an RTS control message or it has a message ready for transmission. Upon receiving an RTS, a sensor sends out an
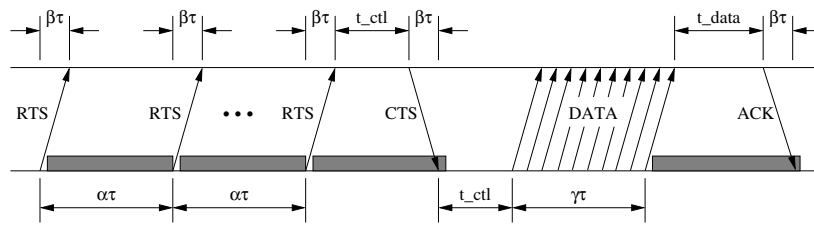
626

Fig. 2.   Transmission cycle (grey area indicates listening period).

CTS and switches over to Receive cycle. One Listen cycle lasts $D\tau$ time units if there is no activity.

Right before switching to a Listen cycle, the sensor will check whether or not a local message is ready for transmission. If such messages exist, the first queued message will be moved to the appropriate Tx buffer to prepare for transmission.

During the optional Sleep cycle (not implemented), sensor places itself in power saving mode for a fixed time duration (either a listen cycle duration or a transmit cycle estimated duration).

III. NETWORK TOPOLOGIES

We limit our attention to degree 2 graphs (each node has two incoming and two outgoing links) mainly because current technology only allows a sensor node to have a small number of transmission and receiving frequencies.

Each node in the sensor network corresponds to a vertex of the graph under consideration. And each edge of the graph corresponds to a dedicated frequency channel between two nodes of the sensor network.

A. Cayley Graphs

To define a Cayley graph, let $(G, *)$ be a group. A graph $C = (V, E)$ is a Cayley graph with a vertex set V if two vertices v_1, v_2 in V are adjacent, that is (v_1, v_2) is in E if and only if $v_1 = v_2 * g$ for some nonidentity element $g \in G$. The set $G - I$, where I denotes the identity element in G, is called the generator set of the graph. The exclusion of I from the generator set prevents the corresponding Cayley graph from having self-loops.

Cayley graphs are advantageous because (1) Cayley graphs are generally dense; and (2) all Cayley graphs are vertex transitive. 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"[6], [8] and simplifies the problem of finding a path between two arbitrary vertices to one of finding a path 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[9], [10].

Figure 3 is an example of a 21-node, degree-4, Borel Cayley graph in the integer domain. The graph has $V = \{0, 1, ..., 20\}$
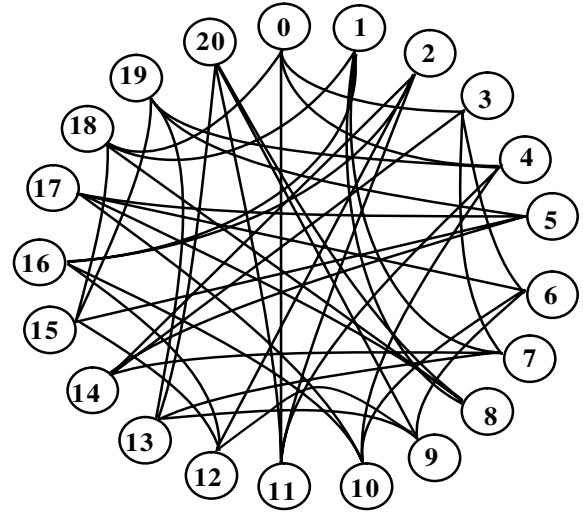

Fig. 3.   Example of a Borel Cayley graph.

nodes. Connections are defined as:

$$\text{For all } i \in V \begin{cases} \text{if } i \bmod 3 = 0 & i \text{ is connected to } i+3, \\ & i-3, i+4, i-10 \bmod n \\ \text{if } i \bmod 3 = 1 & i \text{ is connected to } i+6, \\ & i-6, i+7, i-4 \bmod n \\ \text{if } i \bmod 3 = 2 & i \text{ is connected to } i-9, \\ & i+9, i+10, i-7 \bmod n \end{cases}$$

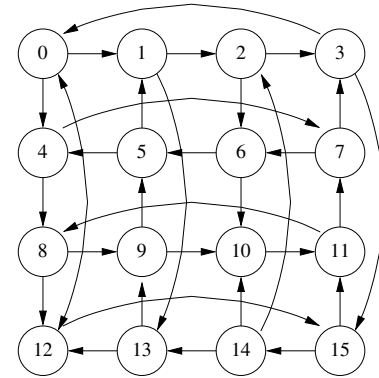### B. Manhattan Street Network

Fig. 4.   Example of a 4x4 Manhattan Street Network.

The Manhattan Street Network (MSN) is a degree 2 directed

627

mesh connected network, with its links resembling the one-way streets and avenues of Manhattan (even number of rows and columns). The diameter (longest distance between two nodes) of an R rows by C columns ($R \times C$) Manhattan Street Network maybe calculated as follows:

$$\Delta = \begin{cases} C/2 + R/2 + 1 & \text{if } R \bmod 4 = 0 \wedge C \bmod 4 = 0, \\ C/2 + R/2 & \text{else,} \end{cases}$$
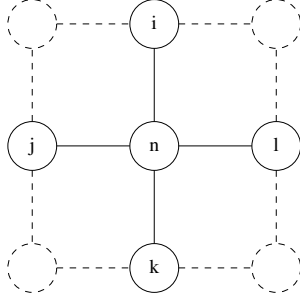
Fig. 5.   Definition of connected neighbors for the Manhattan Street network.

Neighboring nodes are defined by (see Figure 5)

$$\begin{aligned} n &= rC + c \\ i &= (r - 1 \bmod R)C + c \\ j &= rC + (c - 1 \bmod C) \\ k &= (r + 1 \bmod R)C + c \\ l &= rC + (c + 1 \bmod C). \end{aligned}$$

and the direction of connections is defined by

$$\begin{aligned} n &\to i & \text{if } c \bmod 2 = 1 \\ n &\to k & \text{if } c \bmod 2 = 0 \\ n &\to j & \text{if } r \bmod 2 = 1 \\ n &\to l & \text{if } r \bmod 2 = 0. \end{aligned}$$

For each node, the routing matrix consists of a set of outgoing links in the shortest path towards destination node. By symmetry of the network, routing matrices at any node can be derived from the routing matrix at node 0.

IV. SIMULATOR OVERVIEW

The simulator is an extension of the one used in [11]. It is written in the C programming language, is discrete event driven and includes a heap scheduler.

Within the simulation model, sensors are randomly placed in a circular region whose coordinates (x_s and y_s) are randomly selected within a region of diameter the sensor range. All sensors are logically connected with one another according to either a Cayley or Manhattan Street Network graph.

A. Sensor Model

Our model for sensors is parametrized to match the Cross-Bow Mica2 sensor specifications (i.e. processing delays, message lengths, and battery consumption are derived from Cross-Bow Mica2 data sheet). The equivalent software simulation
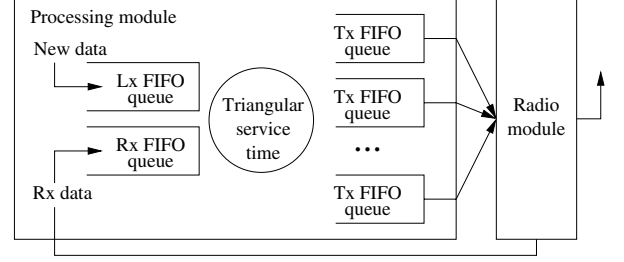
Fig. 6.   Sensor model.

model is depicted in Figure 6. It consists of an input queue for transit messages (Rx), an input queue for new data (Lx), a server (triangular service time distribution), and output queues (Tx, one per adjacent sensor). The queues are located in the logger flash memory while the server is the application running in the processor. Communication to and from adjacent nodes traverses the radio module. New data queued in the local queue (Lx) are moved to the output queue (Tx) destined to the appropriate sensor just before its transmission. All queues are FIFO served.

All aspects of the MAC protocol previously discussed are accounted for. In the current simulation implementation, no transmission errors are accounted for and no timeout procedure was implemented. So that a sensor could wait indefinitely for CTS. And the only queues that may overflow and result in message loss are the local queues (Lx in Figure 6).

B. Power Model

Using the CrossBow Mica2 sensor datasheet, we summarize the battery consumption as function of sensor activity as follows:

μProcessor (ATmega128L)		
$P_{\text{Fo}}^{\mu P}$	8ma/hr	Full operation
$P_{\text{Sl}}^{\mu P}$	15μa/hr	Sleep
Radio		
$P_{\text{Rx}}^{\text{Rd}}$	10ma/hr	Receive
$P_{\text{Tx}}^{\text{Rd}}$	27ma/hr	At 500ft (maximum range)
	15ma/hr	At 100ft
$P_{\text{Sl}}^{\text{Rd}}$	1μa/hr	Sleep
Logger		
$P_{\text{Wr}}^{\text{Lg}}$	15ma/hr	Write
$P_{\text{Rd}}^{\text{Lg}}$	4ma/hr	Read
$P_{\text{Sl}}^{\text{Lg}}$	2μa/hr	Sleep
Sensor Board		
$P_{\text{Fo}}^{\text{Bd}}$	5ma/hr	Full operation
$P_{\text{Sl}}^{\text{Bd}}$	5μa/hr	Sleep
Battery Capacity		
E^{Bt}	3000ma-hr	

Then from CrossBow Mica2 datasheet, we quantified battery current draw for message transmission as function of distance between communicating sensors (Figure 7).

In our simulation model the following activities drew current from the battery:
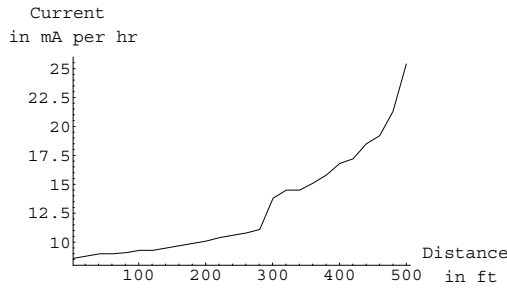
628

Current
in mA per hr

Fig. 7. CrossBow sensor transmission current draw in mA/hr as function of distance in feet.

- Reading or writing to queues: Logger read or write and sensor board full operation. While both μProcessor and radio in sleep mode.
- Sending or receiving messages: Radio receive or transmit. While μProcessor, logger and sensor board in sleep mode.
- Listening for new messages: Radio in receive mode while μProcessor, logger and sensor board in sleep mode.
- μProcessor processing: μProcessor and sensor board in full operation. While both radio and logger in sleep mode.

C. Traffic Patterns

Two traffic patterns have been considered: single node accumulation where all sensors send messages at the same rate to an accumulation sensor, and random traffic pattern where 20% of all sensors are randomly selected to act as traffic source and send messages at the same rate to randomly selected destinations.

Traffic sources generate messages with exponentially distributed interdeparture time. Each new message is placed in the associated sensor local buffer (Lx) for transmission.

D. Simulator Parameters

In addition to the parameters presented in the power model section (see Section IV-B), the simulator uses the following input parameters:

```
N                    52          Number of nodes in sensor network
D                    2           Sensor network degree
R                    13          Number of rows in sensor network
C                    4           Number of columns in sensor network
MAX_Lx               10          Local buffer maximum size
MAX_Tx               1           Transmit buffer maximum size

msg_size(B)          40          Data message size
preamble_size(B)     16          Overhead and control mesage size

data_rate(bps)       38400       Transmitted bandwidth
logger_uart_rate(bps) 40000      Logger bandwidth

tau(msec)            10          listen time period per adjacent sensor
alpha_rts            1           RTS transmission time in tau units

t_ctl(msec)          2.5 5       Min/max control message service time
t_data(msec)         5 10        Min/max data message service time

ListenCycleOrder     Sequential  Listen cycle adjacent nodes ordering

seed                 377003613   Pseudo random number generator seed
total_departures     1e7         Number of simulated data messages
initial_departures   1e4         Limit for reseting statistics
```

Message sizes and data rate where derived from CrossBow Mica2 datasheet. Control messages (RTS, CTS and ACK) and data messages processing delays where educated guesses whose delay distributions follow a triangular distribution.

E. Simulator Metrics

The following quantities reported by the simulator are used to assess sensor networks:

- Sensor utilization: The fraction of time a sensor is processing messages, in a transmit or a receive cycle.
- Battery consumption: Sensor current draw as a function of time.
- Transmission delay: Measured from the instant a data message enters its associated sensor local buffer Lx to the instant it reached its destination sensor,
- Data message loss probability: The ratio of data messages lost due to local buffer Lx overflow to the total number of transmission attempts.
- Network lifetime: The time required for the first sensor to exhaust its battery.

V. SIMULATION RESULTS

We compared Cayley and Manhattan Street networks with 465, 110 and 55 nodes. For each reported quantity, the error spread was estimated to be approximately 10%.

A. Single Node Accumulation

In the following table, we capture key simulator statistics for the single accumulation traffic pattern:

	Network sizes		
	465	110	55
Accumulation node	247	26	26
Traffic intensity (msg/sec)	0.0625	0.26	0.5
Blocking threshold (msg/sec)			
Cayley	0.075	0.29	0.6
MSN	0.075	0.29	0.6
Mean delay (msec)			
Cayley	340	271	242
MSN	470	300	258
Proportion of node whose utilization < 5%			
Cayley	89%	65%	37%
MSN	83%	47%	29%
Peak battery consumption in 1hr (ma-hr)			
Cayley	15	15	15
MSN	15	15	15

We found that beyond the reported blocking thresholds, end-end delay and message loss grow significantly, indicating the tested networks throughput limit was reached. For the blocking threshold, there were little differences between Cayley and Manhattan Street networks.

Inspection of delay distributions in Figure 8 shows the delay distribution for the Cayley network to be "tighter" than that of the Manhattan Street network. These differences are amplified with increasing network sizes. Moreover, Cayley network mean delay is always less than that of the Manhattan Street network.

From inspection of the sensor utilizations in Figures 9, even for a centralized traffic pattern such as single node accumulation, we observe that under the Cayley network sensor utilization is more uniformly distributed than under
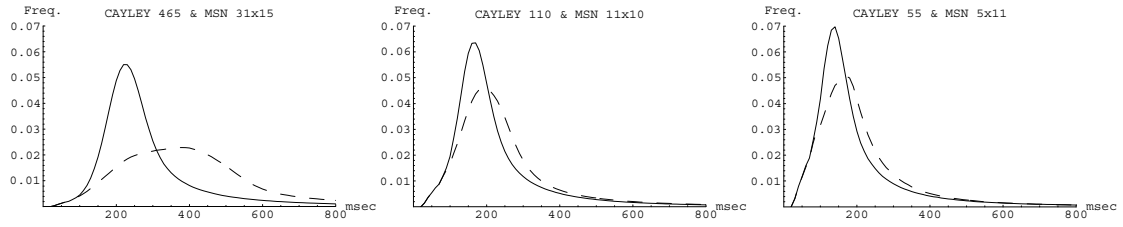
Fig. 8. End-to-end message delay histograms (10msec bins) for Manhattan Street Network (dashed) and Cayley networks (continuous) of 465, 110 and 55 nodes when subject to the single node accumulation traffic pattern.
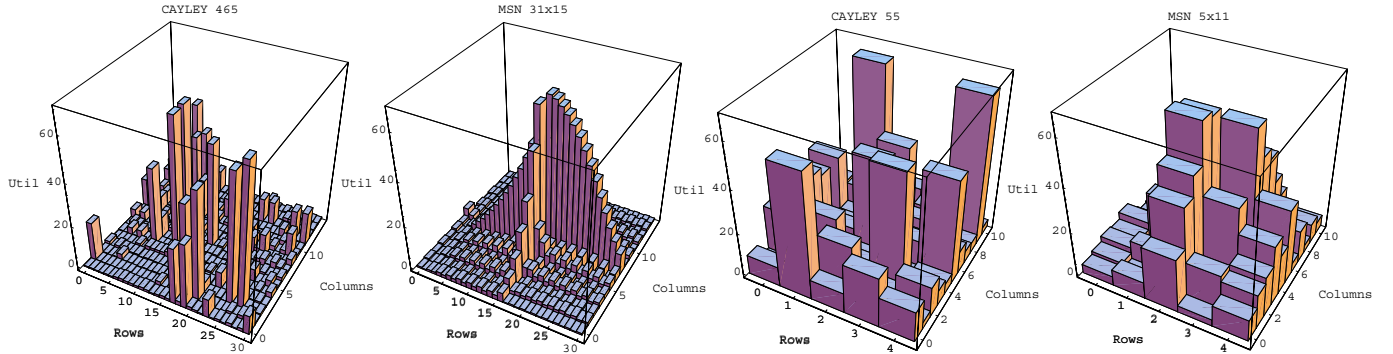

Fig. 9. Sensor utilization for Cayley and Manhattan Street networks of 465 and 55 nodes when subject to the single node accumulation traffic pattern.

the Manhattan Street network. While the proportion of sensor whose utilization is less than 5% is always largest with the Cayley network.

Lastly, we found little differences for peak battery consumption between Cayley and Manhattan Street networks. Moreover, because the sleep cycle was not implemented, battery consumption spread between lightly loaded and loaded sensors was small.

B. Random Traffic Pattern

In the following table, we capture key simulator statistics for the random traffic pattern:

	Network sizes		
	465	110	55
Number of active node pairs	86	24	10
Traffic intensity (msg/sec)	0.05	0.09	0.5
Blocking threshold (msg/sec)			
Cayley	0.5	0.6	0.6
MSN	0.07	0.1	0.6
Mean delay (msec)			
Cayley	202	135	119
MSN	298	161	134
Peak battery consumption in 1hr (ma-hr)			
Cayley	15	15	15
MSN	15	15	15

For the blocking threshold metric, we found with increasing network size blocking threshold for the Cayley network to be significantly better than that of the Manhattan Street network.

As with the single node accumulation traffic pattern, inspection of delay distributions in Figure 10 shows the delay distribution for the Cayley network to be "tighter" than that of the Manhattan Street network. These differences are amplified with increasing network sizes. Moreover, Cayley network mean delay is always less than that of the Manhattan Street network.

From inspection of the sensor utilizations in Figures 11 we observed little differences between Cayley Manhattan Street networks.

Again, we found little differences for peak battery consumption between Cayley and Manhattan Street networks. And because the sleep cycle was not implemented, battery consumption spread between lightly loaded and loaded sensors was small.

VI. CONCLUSION

We presented a novel MAC protocol inspired by SMACS[5] that exploits network graphs property to assign frequencies for efficient simultaneous peer-to-peer communications in large and dense wireless sensor networks. Via a simulation model parameterized to CrossBow MICA2, we evaluated the performance of our MAC protocol with Cayley and Manhattan Street networks.

We found our MAC protocol to perform better with the Cayley network than with the Manhattan Street network and that differences were amplified with increasing network size.

In future work, we expect to implement the sleep cycle capability, correlate logically adjacent nodes to geographically nearby nodes and implement our MAC protocol on CrossBow Mica2 sensors.
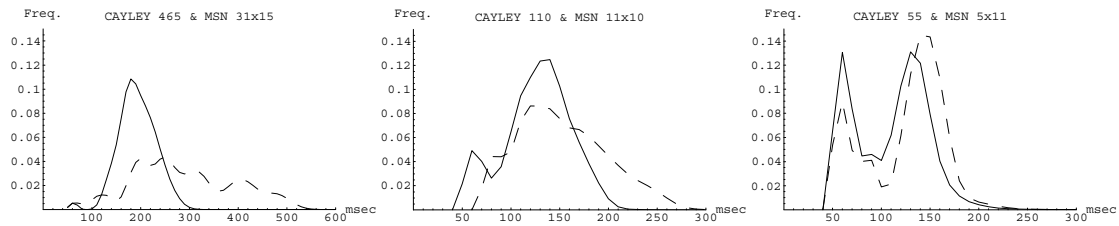
Fig. 10. End-to-end message delay histograms (10msec bins) for Manhattan Street Network (dashed) and Cayley networks (continuous) of 465, 110 and 55 nodes when subject to the random traffic pattern.
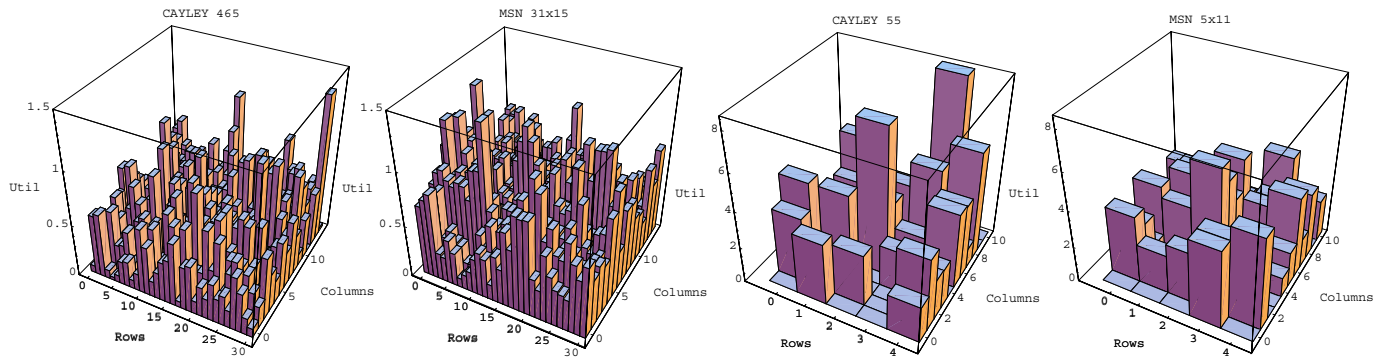

Fig. 11. Sensor utilization for Cayley and Manhattan Street networks of 465 and 55 nodes when subject to the random traffic pattern.

ACKNOWLEDGMENT

REFERENCES

[1] Gregory T. Huang, *Casting the Wireless Sensor Net*, Technology Review, July/August 2003, pp. 50-56.

[2] David Culler, *10 Emerging Technologies that will Change the World*, Technology Review, February 2003, pp. 33.

[3] *Demo: Sensor World*, Technology Review, April 2004, pp.70-73.

[4] Chee-Yee Chong and Srikanta P. Kumar, *Sensor Networks: Evolution, Opportunities, and Challenges*, Proceedings of the IEEE, 91(8): pp. 1247-1256, August 2003.

[5] Katayoun Sohrabi, Jay Gao, Vishal Ailawadhi and Gregory J. Pottie, *Protocols for Self- Organization of a Wireless Sensor Network*, IEEE Personal Communications, October 2000, pp. 16-27.

[6] A. Cayley, *On the Theory of Groups*, Proceedings of London Mathematics Society, 9:126-133, 1878.

[7] N. F. Maxemchuck, *The Manhattan Street Network*, IEEE GLOBE-COM'85, pages 255–261, New-Orleans, September 1985.

[8] N. Biggs, *Algebraic Graph Theory*, Cambridge University Press, London, 1974.

[9] K. Wendy Tang. Dense, *Symmetric Interconnection Networks*, PhD thesis, Department of Electrical Engineering, University of Rochester, Rochester, New York, 1991.

[10] B.W. Arden and K. Wendy Tang, *Representations and Routing for Cayley Graphs*, IEEE Transactions on Communications, 39(11):1533-1537, November 1991.

[11] E. Noel and K. W. Tang, *Performance Modeling of Multihop Networks Subject to Uniform and Non-Uniform Geometric Traffic*, IEEE/ACM Transaction in Networking, volume 8, pages 763–774, December 2000.