Empirical Evaluation of Querying Mechanisms for Unstructured Wireless Sensor Networks

*§Joon Ahn[1], Shyam Kapadia[2], Sundeep Pattem[1], Avinash Sridharan[1], Marco Zuniga[3],
Jung-Hyun Jun[4], Chen Avin[5], Bhaskar Krishnamachari[1]
[1]Dept. of Electrical Engineering, University of Southern California, USA
[2] Cisco Systems Inc., San Jose, CA, USA
[3] Xerox Research Center Webster, Webster, NY, USA
[4]Dept. of Electrical and Computer Engineering, University of Cincinnati, USA
[5] Communication Systems Engineering Dept., Ben Gurion University of The Negev, Israel

ABSTRACT

In the last few years, several studies have analyzed the performance of flooding and random walks as querying mechanisms for unstructured wireless sensor networks. However, most of the work is theoretical in nature and while providing insights into the asymptotic behavior of these querying mechanisms, does not account for the non-idealities faced by the network in real deployments. In this paper, we present an empirical evaluation of the performance of both flooding and random walks in real environments. The metrics considered are delay, reliability and transmission cost. Our results suggest that flooding is better suited for low-interference environments, while random walks might be a better option in networks with high interference.

1. INTRODUCTION

Querying in sensor networks can be classified in the following ways: (a) based on the type of data – *continuous* streams, or *one-shot* information (b) based on the type of queries – *pull-based* (where the sink issues the queries for information) or *push-based* (where the detection of events triggers notifications to the sink), or a *hybrid* of these two and (c) based on the query process – *structured* (using an index or hash table) or *unstructured* (stored at locations unknown to the querying sink node). Our study explores the area of unstructured systems; more precisely, we focus on one-shot, pull-based data querying in unstructured wireless sensor networks. Some typical examples of such queries are: 'give me any location where a lion was seen?', 'give me all locations where the temperature is above 100° F?' etc.

Three important querying mechanisms for one-shot queries in unstrucutured networks are: flooding, controlled flooding (using expanding rings), and random walks. These querying mechanisms have been extensively studied in the context of wireless sensor networks [4, 9, 10, 11, 25, 16, 14]. However, most of these studies are theoretical or based on simulations, and they make simplified assumptions about the communication model. Recent studies [1, 2] indicate that the behavior of real wireless networks may differ significantly from that observed in analytical/simulation studies. Hence, in order to determine the best querying mechanism for a given application, it is important to complement the initial theoretical/simulation studies with insights derived from real deployments. The aim of this work is to provide some initial design insights based on the behavior of flooding and random walks in real environments.

Our work is motivated by the fact that, while different variants of flooding have been evaluated in experimental test-beds [1, 3], and limited realistic simulations have been reported recently expanding ring-based controlled floods [4], *there has been no prior work on evaluating the performance of random walks in a realistic wireless system.* Our results suggest that a an initial design decision between flooding and random walks could be made based on the interference level of the environment. For scenarios with low interference, flooding appears as a better querying mechanism, while for high-interference environments, random walks provide a superior performance. Our results also give useful insights into use of memory for improving random walk performance.

The rest of the paper is organized as follows. In Section 2, we discuss some related work. Our implementation of the flooding and random walk based querying mechanisms are discussed in Section 3 and Section 4 respectively. In Section 5 we provide details of our experimental methodology. Subsequently in Section 6, we describe the main metrics and the results of our empirical evaluation. Finally, in Section 7 we conclude with a discussion of our findings and describe the major future research directions that we intend to pursue.

---

*§ The first five authors are listed in alphabetical order. All authors participated in the work described in this paper while they were affiliated with the Autonomous Networks Research Group at the University of Southern California. This work is supported in part by the National Science Foundation (NSF) under the NETS NOSS grant 0435505 and CAREER grant 0347621. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

2. RELATED WORK

Querying in wireless sensor networks can be broadly classified into two main categories: structured and unstructured mechanisms[1]. In structured systems (Geographic Hash Tables [6], DIMENSIONS [7], DIFS [8]), the content of event data is used to determine where they should be stored. The query mechanisms for such systems can be quite efficient because the sink can use the same indexing or hashing mechanism to determine the storage location of desired content and retrieve it.

In contrast, unstructured systems minimize the overhead of pre-configuration. The simplest implementation of a query dissemination protocol for a sensor network is a basic flooding mechanism. An efficient enhancement over this mechanism is the sequential TTL-based controlled flooding (expanding rings) [4, 9, 10, 11, 12].

Random walks are also finding increasing use in the context of unstructured wireless sensor networks. Different variants of random-walk-based protocols have been proposed and analyzed by several research groups. The authors of [24, 25] highlight their inherently load-balanced advantages. The ACQUIRE protocol [14, 15] combines random walks with controlled floods, while the rumor routing algorithm [16] is a hybrid push-pull mechanism that provides a rendezvous point for queries and events. However, to the best of our knowledge, despite so many proposals for using random walks in sensor networks, they have never before been evaluated on a real wireless network test-bed.

Indeed, the topic of experimental evaluation of multi-hop wireless sensor network protocols itself could be said to be in a relatively nascent stage today. Ganesan *et al.*[1] performed a seminal experimental study of basic flooding on a real test-bed, and revealed several surprising observations. This was followed by several experimental studies investigating the behavior of real wireless links [17, 18, 19, 20, 21, 22, 23]. Our work shares the spirit of this growing literature on realistic evaluation of wireless network protocols and covers new ground.

3. RANDOM WALKS

In this study, we consider both the simple random walk, and its enhanced version called the self-avoiding random walk with memory [13]. They differ in the method of selecting the next hop neighbor when forwarding the random walk token (the token in this case is the query packet).

- **Simple RW**: At each step, a node chooses one of its neighbors randomly to forward the walk. Every neighbor has an equal probability of being chosen. One weakness of the simple random walk is its tendency to revisit recently visited nodes. It can spend a significant amount of time in the vicinity of the starting node before exploring the rest of the network - causing significant delays and increased energy cost.

- **Self avoiding RW with k-Memory**: This walk avoids the most recently visited k nodes that were part of its trajectory. The identity of the last k nodes is stored within the random walk query packet itself. If the random walk finds itself trapped at a node such that all its neighbors have been visited in the last k steps, only then does it violate this avoidance rule to escape, by picking one of the neighbors uniformly at random.

3.1 Salient Features

Communication over lossy links, asymmetric links, irregular network topologies, transient link failures, dynamic network topology changes, node failures, interference from other networks (e.g. IEEE 802.11 a/b/g, Bluetooth etc.) are some of the characteristics of real environments. To counter some of these vagaries, two key elements are employed in our random walk protocol implementation: neighbor discovery and reliable forwarding.

3.1.1 Neighbor Discovery

As described earlier, at each step, the random walk propagates by forwarding a token to a chosen neighbor. If the link to the chosen neighbor is lossy, a high communication cost may be incurred. These issues are handled as follows in the neighbor discovery phase.

Link Quality Estimation: To ensure that during forwarding, a node chooses a good 'neighbor', we first perform link quality estimation across the network. A simple two-phase TDMA scheme is implemented for packet reception rate (PRR) measurements. In the first phase, every node is allocated 100 time slots. In every time slot, a designated node broadcasts a packet. Thus, at the end of phase one, each node ends up transmitting a total of 100 packets. Since the packets are broadcasted, all nodes log the packets received from other nodes noting the identity of the sender. At the end of this phase, each node has a record of their in-degree PRR with respect to every other node i.e. the number of packets (out of 100) successfully received from every other node. In the second TDMA phase, every node exchanges their in-degree PRR information with every other node. Hence, at the end of the second phase, all nodes are aware of their in-degree and out-degree PRR values with respect to every other node.

Blacklisting: The link quality estimation stage provides each node the PRR for its incoming and outgoing links to every other node in the network. Good neighbors for each node are determined using a blacklisting threshold for bi-directional links. Only neighbors with links with PRR above this threshold in both directions are chosen as potential next hop token receivers.

3.1.2 Reliable Forwarding

A practical implementation of a random walk based querying scheme must ensure that the random walk token is not "lost" during transmissions. Moreover, with energy being a key resource, the walk should be able to propagate around transient link failures without expending a lot of energy. Hence, a node may attempt to forward the token to a chosen neighbor a certain number of times before giving up and choosing an alternate neighbor. Also, a single random walk should not branch off into multiple instances, as this would lead to increased transmission costs and complexity in terms of protocol management. In this study, our focus is on *single* threaded random walks. To ensure reliability and duplicate walk suppression, the forwarding process for the random walk token employs a three-way handshaking protocol similar to that employed by TCP [26].

---

[1] The nomenclature is derived from the analogous classification of peer-to-peer networks [5]

We briefly explain the protocol considering the forwarding of the random walk token from a node A to node B, its chosen neighbor. Node A initiates the forwarding process by sending a RW_PKT packet to node B. Node B on successful reception replies with a RW_ACK packet. When A receives the acknowledgement it sends a RW_REL packet indicating to node B that henceforth it is the responsibility of node B to forward the token. We employ suitable timers at both the sender and receiver ends along with session state information (for e.g. sequence number etc.) to recover from the possibility of lost packets and to avoid multiple simultaneous random walk threads.

The timer values employed have an impact on the delay for network coverage and in turn the time taken for query resolution. Too large a timer value will inflate the delay while too small a value will cause frequent timeouts and result in higher energy expenditure. We empirically calculated the average time taken for transmission of a packet between two neighboring nodes and chose the timer value to be twice the average packet transmission time plus a certain tolerance . Note that successful transmission of a token from one node to another involves three successive back-to-back transmissions and receptions for reliable delivery. This manifests itself in the form of a higher average delay observed with random walks for query resolution.

We discuss briefly another enhancement to the random walk protocol. In wireless networks, the transmissions are inherently broadcasted. Hence, random walks in wireless networks allow for the possibility that the packet being forwarded is overheard by other nodes in the neighborhood. Particularly in dense deployments where neighboring nodes remain awake, this broadcast advantage can provide a drastic advantage in terms of the time taken until a node containing the desired information hears the query (*the hitting time*) or the time taken to explore all nodes in the entire network (*the cover time*). To enable this, the token being passed during the walk is sent as a broadcast packet, but the intended recipient's ID is recorded in the data, and only that node is allowed to forward the packet at the next step. All other nodes passively record the query. The disadvantage here is that it complicates how the random walk may be terminated once a node receiving a query via a broadcast resolves it and sends the reply to the sink. Implementation of random walks using broadcast advantage is part of of ongoing and future work.

4. FLOODING

We have implemented a version of the basic flooding scheme in which there is no explicit mechanism to ensure reliability. To prevent an "explosion" due to repeated rebroadcasts of flooding packets heard from multiple neighbors, each instance of flooding has a FLOOD_ID. When a node receives a FORWARD_FLOOD packet with a particular FLOOD_ID, it will rebroadcast only if it is hearing this FLOOD_ID for the first time. This first reception and its time is logged for the delay calculation. Suitable timers and sequence number checks ensure that back-to-back floods do not overlap.

5. EXPERIMENTAL METHODOLOGY

5.1 Deployment description

We performed experiments both in an indoor and outdoor environment. The characteristics of the two environments are as follows:

- **Indoor:** This environment comprised off a large empty room on the fourth floor of one of the campus buildings. There is an operational 100 node sensor network test bed on the third and fourth floors of this building. Other sources of interference are several research labs using wireless devices and the presence of a large number of IEEE 802.11b/g networks.
- **Outdoor:** This environment comprised off a large open area outside one of the campus buildings where we found that there is significantly lower interference from external sources as compared to the indoor environment.

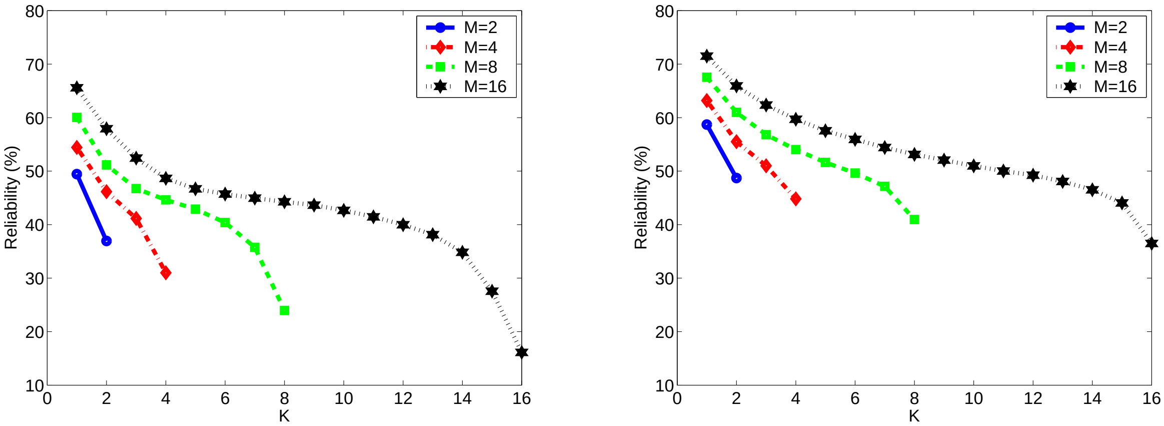
The experiments employed Moteiv Tmote-Sky devices, popularly known as the telosb motes [27]. The experimental topology comprises a square grid, however, as compared to an ideal symmetric grid where each node has 4 neighbors (except border nodes), in practical scenarios not all neighbors only send and receive packets from their immediate neighbors. We deployed networks of different grid sizes: $\{5 \times 5,\ 7 \times 7,\ 8 \times 8,\ 9 \times 9\ \}$.

The experiments were conducted on different days, with no discernible changes in the environments - this is important to substantiate later observations on reliability of flooding. This ensures that the results are true in general for the conditions described and are not artifacts of flawed implementation or rare events.

5.2 Experiment description

The following steps describe our experimental methodology:

1. Nodes were programmed with unique node ID's and tagged correspondingly so that their locations could be matched with the collected trace information.
2. A laptop, serving as a base-station, was used to monitor and issue commands to the nodes to start/stop/reset an experiment run for the appropriate querying protocols (flooding and random walk).
3. The initial placement phase is followed by a neighbor discovery phase, where the nodes first measure pairwise packet reception rates (PRRs) between them using a TDMA-based scheme (see Link Quality Estimation in Section 3). The PRRs are only recorded once at the beginning since the environment is expected to be static. Initially, the nodes were calibrated at the minimum power level.
4. The nodes sent their collected PRR information to the base-station where a verification procedure was executed to ensure that all the nodes formed a single connected component. If the collected PRRs indicated that the network was disconnected, the power level of the nodes was increased till a single connected component was formed. Alternatively, the distances between the nodes could be reduced to have a similar effect.



(a) 25 node grid network (b) 49 node grid network

Figure 1: Reliability for query resolution with flooding for indoor environments.

5. After the PRR phase, we conducted various runs of flooding and random walk. During these experiments, each node logged various details for packet transmissions and receptions such as the source, destination, time of transmission, time of reception, type of protocol, etc.

6. A START_FLOOD command was issued from the base-station specifying the node from which the flood should start (sink node) and the number of back-to-back floods to be issued. Appropriate timer settings ensured that successive floods did not overlap.

7. After the flooding phase, a START_RW command was issued from the base-station initiating a single iteration of a random walk (with or without memory) at the same sink node. Ideally, as soon as the cover time (meaning the earliest time when all the nodes have been visited by the random walk at least once) is reached we would have liked to terminate the walk. Since doing this precisely is difficult, we terminate after the walk had hopped a certain number of times (say RW_STEPS). If RW_STEPS is really large value almost certainly the cover time would be reached, however, to prevent running over the limited flash memory that the motes possess for logging purposes (only 1MB), we empirically determined an appropriate value for RW_STEPS[2]. This marked the termination of a single iteration of the random walk.

8. We performed several such random walk iterations and in post processing the traces only considered those where the cover time was reached.

9. For both the indoor and the outdoor environments, we performed a 1000 flooding iterations, followed by 10 random walk iterations each with different memories varied as {1, 2, 3, 4 }.

6. EVALUATION AND RESULTS

We consider one-shot queries of the type 'find any K out of a total of M events?'. We assume that a total of M events exist in the network; the location of these events may be drawn from an arbitrary distribution. For simplicity, we consider a uniform distribution for the location of the M events. Queries desiring exactly K out of M events are issued at a sink node ($0 \leq K \leq M$). The sink node is assumed to be located at one corner of the grid network. The query initiates a random walk from the sink node and proceeds to find the desired K events. When $K = 1$, the query is resolved when it hits any event node for the first time, while when $K = M$, the query is resolved when it visits the last of the non-visited event nodes.

6.1 Metrics

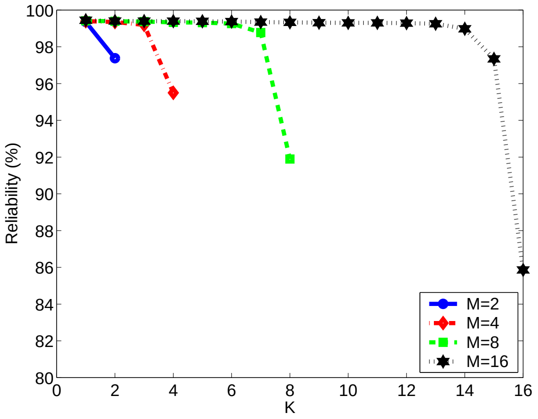
The key application metrics considered in this study are average delay, average energy (in terms of number of packet transmissions), and reliability for query resolution. The metrics of delay and energy do not consider the cost of the query response to the sink. Hence, the delay metric considered here is the one way searching delay (complete query delay would also need the return path delay). Similar is the case for the energy metric. For flooding, the transmission cost is fixed independent of the value of K and M but the delay is the earliest time at which the K events are found. For random walk, both the number of transmissions and delay depend on the earliest hop at which the K^{th} event is found. The assumption is that the random walk would have stopped at this point. Reliability is calculated as the percentage of successfully resolved queries.

6.2 Results

We emulate the K out of M type queries on the traces. In postprocessing the trace data, we are able to emulate the existence of arbitrary numbers and locations of event sources. For each value of M and K, we consider a 1000 different placements for the event nodes, hence, the results presented are averages over 1000 iterations. The error bars capturing the 95% confidence intervals for the metric results were quite tight, hence, we do not present them here. Due to space constraints, we cannot exhaustively study all possible

---

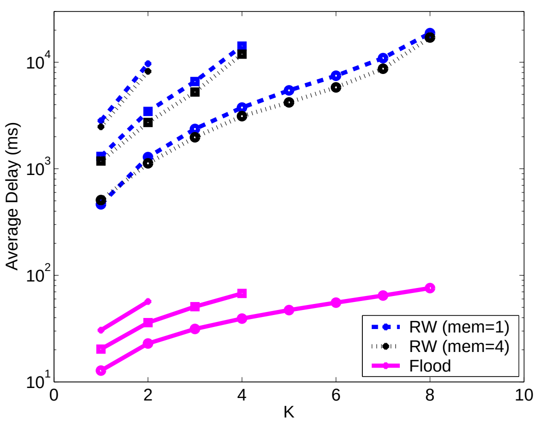
[2] While theoretically it is known that cover time for a random walk on a symmetric grid of size N is $O(N^2 \cdot \log N)$, we found that this value was an over-estimate especially in practical scenarios where one cannot get an 'ideal' grid



(a) 64 node grid network

(b) 81 node grid network

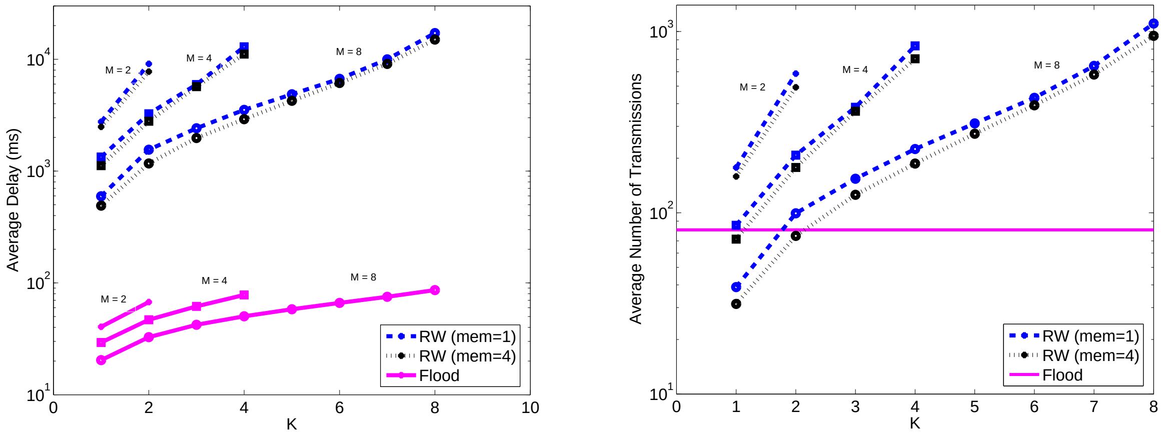
Figure 2: Reliability for query resolution with flooding for outdoor environments.



(a)

(b)

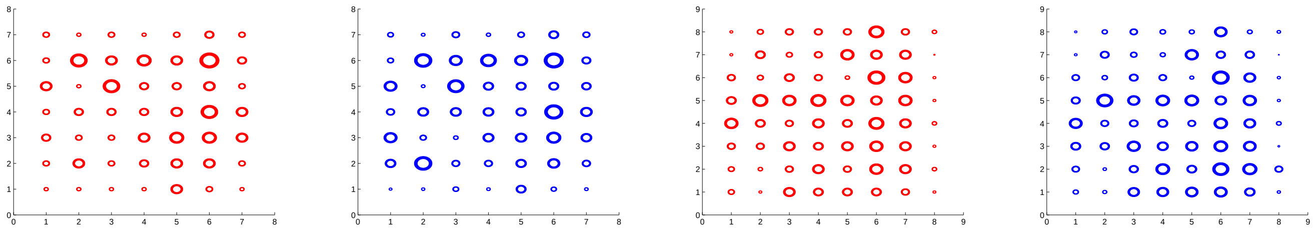
Figure 3: (a) Average delay and (b) Average energy in terms of number of transmissions, for query resolution with the flooding and random walk querying schemes for a 64-node grid network in an outdoor environment.



(a)

(b)

Figure 4: (a) Average delay and (b) Average energy in terms of number of transmissions, for query resolution with the flooding and random walk querying schemes for a 81-node grid network in an outdoor environment.



(a1) (a2) (b1) (b2)

Figure 5: (a1), (b1) Relative node degree and (a2), (b2) relative number of visits of random walk with 1-step memory for the indoor 49 node and outdoor 64 node networks.

scenarios in the evaluation, but rather focus on illustrating some key observations.

Figures 1 (a) and (b) show the reliability of flooding for the indoor environment for 25 and 49 nodes respectively. It is observed that the success percentage is pretty low for all values of K and M. The success percentage for the random walk was 100%. As described earlier, this is a high interference environment and packet losses occurred frequently. It was observed that in most flooding runs, a significant number of nodes could not receive a single packet. For the random walk, packet forwarding was observed to stall many times due to repeated packet losses but eventually recovered every time.

Clearly, simple flooding is not a feasible querying mechanism under such conditions. The random walk has an increased cost due to the packet losses but achieves 100% reliability. In this regime, a reliability mechanism is needed for flooding to be useful as a querying mechanism. It will be interesting to see with the increase in cost for this reliability how flooding will compare with random walks.

For the benign outdoor environment with little or no interference, Figures 2(a) and (b) show that simple flooding has high reliability for most K and M values. Figures 3(a) and 4(a) show the average delay and Figures 3(b) and 4(b) show the relative energy performance for flooding and random walks in the outdoor environment for 64 and 81 node networks respectively. Here, we find that the reliability of flooding is also quite high and comparable to the 100% reliability provided by the random walk scheme. However, as compared to random walks, the delay for flooding is lower by an order of magnitude and the energy cost is also lower for most K and M values. This suggests that simple flooding is a very option in such conditions. In this regime, it would be interesting to see by how much performance of random walks can be improved by exploiting the broadcast advantage (see the end of Section 3).

Self-avoiding random walks with memory improved the cost and delay when compared to simple random walks. However, from Figures 3 and 4, we see that having a memory of 4 previous steps does not give significant gains over having a memory of 1 step i.e there are diminishing returns with increasing memory size. This indicates that most of the gains can be obtained with minimal overhead of carrying one step memory in the packets. Figure 5 shows the close correspondence between node degree and number of visits made by the random walk with a memory of 1 for both indoor (49 nodes) and outdoor(64 nodes) environments.

7. CONCLUSION

Random walk based protocols have been gaining prominence as a useful tool for querying as well as other useful operations in sensor networks due to their simplicity, natural load-balancing and global coverage properties. We have provided the first-ever experimental characterization of such random walks. In the process, we have developed and implemented memory-based self-avoiding walks that show better coverage properties than the simple random walk.

We compare the performance of random walks and simple flooding as querying mechanisms under a range of settings. Our results suggest that for isolated networks with little or no interference, flooding has high reliability and outperforms random walks in terms of delay and energy costs. For networks with considerable interference, reliability of flooding is drastically reduced and random walk based approaches might be better suited. Our results show there are diminishing returns with increasing memory size for self-avoiding random walks. For a more complete understanding and characterization of the design space, the following need to be studied:

- The use of broadcast advantage can significantly decrease both delay and cost of random walks, particularly in dense deployments.
- In lossy environments, how will the higher cost of implementing reliable flooding compare with random walks?
- Incorporation of the reply path costs in both the delay and energy metrics
- Controlled flooding or expanding ring search
- Experiments in several environments with varying levels of interference and multi-path.

8. REFERENCES

[1] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin and S. Wicker, Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks. UCLA CS Technical Report UCLA/CSD-TR 02-0013, 2002.

[2] D. Kotz and C. Newport and C. Elliott, The mistaken axioms of wireless-network research, Technical Report TR2003-467, Dept. of Computer Science, Dartmouth College, July 2003.

[3] F. Stann, J. Heidemann, R. Shroff, and M. Murtaza, Rbp: Reliable broadcast propagation in wireless networks, In USC/ISI Technical Report ISI-TR-2005-608, 2006.

[4] B. Krishnamachari, J. Ahn, Optimizing data replication for expanding ring-based queries in wireless sensor networks, WiOpt, Boston, MA, April 2006.

[5] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, Search and replication in unstructured peer-to-peer networks, Proceedings of the 16th annual ACM International Conference on supercomputing (ICS) 2002.

[6] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, Data-Centric Storage in Sensornets with GHT, A Geographic Hash Table, In Mobile Networks and Applications (MONET), Special Issue on Wireless Sensor Networks, 8:4, Kluwer, August 2003.

[7] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann, An Evaluation of Multi-resolution Storage for Sensor Networks, The First ACM Conference on Embedded Networked Sensor Systems (Sensys03), November 2003.

[8] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker, DIFS: A Distributed Index for Features in Sensor Networks, Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, May 2003.

[9] N. B. Chang and M. Liu, Revisiting the ttl-based controlled flooding search: Optimality and randomization, In Proceedings of the Tenth Annual International Conference on Mobile Computing and Networks (ACM MobiCom), 2004.

[10] N. B. Chang and M. Liu, Controlled flooding search in a large network, accepted for publication, IEEE/ACM Trans. on Networking, 2006.

[11] Z. Cheng and W. Heinzelman, Flooding Strategy for Target Discovery in Wireless Networks, ACM/Baltzer Wireless Networks, 2005.

[12] Z. Cheng and W. Heinzelman, Searching Strategy for Multi-Target Discovery in Wireless Networks, 4th Workshop on Applications and Services in Wireless Networks (ASWN '04), August 2004.

[13] D. Aldous and Fill, Reversible Markov Chains and Random Walks on Graphs, Draft Monograph, online at http://www.stat.berkeley.edu/ ∼ aldous/RWG/book.html

[14] N. Sadagopan, B. Krishnamachari, and A. Helmy, The ACQUIRE Mechanism for Efficient Querying in Sensor Networks, IEEE International Workshop on Sensor Network Protocols and Applications (SNPA'03), May 2003.

[15] N. Sadagopan, B. Krishnamachari, and A. Helmy, Active Query Forwarding in Sensor Networks (ACQUIRE), Journal of Ad Hoc Networks, Elsevier, Vol 3, Issue 1, pp. 91-113, January 2005.

[16] D. Braginsky and D. Estrin, Rumor Routing Algorithm For Sensor Networks, The First Workshop on Sensor Networks and Applications (WSNA'02), October 2002.

[17] J. Zhao and R. Govindan, Understanding Packet Delivery Performance in Dense Wireless Sensor Networks, ACM Sensys, November 2003.

[18] D. Lal, A. Manjeshwar, F. Herrmann, E. Biyikoglu, A. Keshavarzian, Measurement and characterization of link quality metrics in energy constrained wireless sensor networks, GLOBECOM 2003 - IEEE Global Telecommunications Conference, no. 1, Dec 2003 pp. 446-452

[19] A. Cerpa, N. Busek, and D. Estrin, SCALE: A tool for Simple Connectivity Assessment in Lossy Environments, CENS Technical Report, September 2003.

[20] A. Cerpa, J. Wong, L. Kuang, M. Potkonjak and D. Estrin, Statistical Model of Lossy Links in Wireless Sensor Networks, In the ACM/IEEE Fourth International Conference on Information Processing in Sensor Networks (IPSN'05), UCLA, Los Angeles, California, USA, April 25-27, 2005.

[21] M. Zuniga, B. Krishnamachari, Analyzing the Transitional Region in Low Power Wireless Links, First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON), Santa Clara, CA, October 2004.

[22] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, D. Culler, Exploiting the Capture Effect for Collision Detection and Recovery, The Second IEEE Workshop on Embedded Networked Sensors (EmNetS-II). Sydney, Australia. May 30-31, 2005.

[23] D. Son, B. Krishnamachari, and J. Heidemann, Experimental Analysis of Concurrent Packet Transmissions in Low-Power Wireless Networks, Technical Report ISI-TR-609, USC/Information Sciences Institute, December, 2005.

[24] S. D. Servetto, G. Barrenechea, Constrained Random Walks on Random Graphs: Routing Algorithms for Large Scale Wireless Sensor Networks, In the Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, September 2002.

[25] C. Avin and C. Brito, Efficient and Robust Query Processing in Dynamic Environments Using Random Walk Techniques, In Proceedings of the Third IEEE/ACM International Symposium on Information Processing in Sensor Networks (IPSN 2004), pp. 396-404, Berkeley, California, April 26-27, 2004.

[26] R. Stevens and G. Wright TCP/IP Illustrated Vol.2 - The Implementation, Addison-Wesley, 1995.

[27] http://www.moteiv.com/products/tmotesky.php.