Artery: A Data-Centric Architecture for Wireless Sensor Networks

Lan LIN
Department of Computer Science, University of Denver
Denver, CO 80208, USA

Hailin WU
Department of Computer Science, University of Denver
Denver, CO 80208, USA

ABSTRACT

Sensor networks, composed of large amount of micro-sensors, are considered promising, both in academic research and in real life applications. To ensure highly efficient communications between event observers and sensor network users, new infrastructures and algorithms are being developed. This paper describes Artery, a novel architecture that delivers queries and data between multiple observers and multiple mobile users. Simulation results show that Artery outperforms some major data dissemination algorithms.

Keywords: Sensor Network, Routing Scheme, Network Topology, Data-Centric, and Clustering.

1. INTRODUCTION

The wireless communication technologies have advanced in such a great scale that the deployment of large sensor networks has become possible. Thousands of sensor nodes, deployed in a vast area, perform functions like event monitoring, data computation and aggregation, and collaboration through communications. Potential applications of these networks, such as, environmental monitoring, will be in all aspects of our daily lives.

The sensor networks are to be self-configuring, scalable, and robust in order to adjust to changing topologies. The applicable algorithms are distributed, as long-ranged communications are expensive due to stringent power constraints. The applications must be able to gather information from all parts of the network without taxing its limited power and bandwidth [2, 3].

One of the main research topics has been data dissemination, that is, how to efficiently transmit queries and data between sensor nodes who observe events and network users who try to gather interesting information. Previous work has been more focused on flooding part of or the entire network with queries [4, 5]. In this paper we propose a network infrastructure that is highly efficient when dealing with queries and data analysis. It is based on two observations. One is that data transmissions from multiple sources to multiple sinks are not sufficiently aggregated. When there are multiple pairs of source and sink in a field, it is very likely that certain sections of the paths between them can be combined. The other observation is that the paths established between source and sink are often not for repeated use. Due to energy constraints, it is costly to build new paths frequently. Since mobile sinks generally do not move at a rapid speed, the path established one time period earlier should still be valid for the next time period after minor adjustments.

We call the infrastructure Artery. Like its name, it is a ring of sensor nodes connecting to each other through short-ranged radios. It is located in the mid-way from the center to the boundary of the network, with paths connecting it to every node in the network. Data collected by sensors are flown back to Artery through the paths. Sinks use the paths to send query requests to Artery. It achieves high data aggregation rate especially when multiple sinks query on the same event. The sinks can all tap into Artery so no individual paths need to be built. Artery, a database by itself, can also answer queries on the whole network, such as, the total number of events in the whole network in the past ten minutes. This has huge advantages when the sinks need to do in-network decision-making instead of sending the data outside of the network for processing. One such example is a battle field scenario, where soldiers move around as mobile sinks collecting data from sensors and making decisions in the field.

In the rest of paper, we give an overview of Artery architecture (section 2), a detailed scheme (section 3) and the performance evaluation (section 4). We also briefly discuss some related algorithms and future work.

2. OVERVIEW

We make the following assumptions on the sensor networks we use in this paper:

- A large amount of homogeneous sensor nodes densely cover a vast field, that is, each node is within communication range of some other nodes.
- Sensor nodes are stationary and location-aware. The location information is attainable by receiving GPS signals or through techniques described in [6].
- Sinks (users) are mobile and may not be location-aware.
- Events are all within a fixed region of the network, not a whole network phenomenon.

The Artery routing scheme comprises of five parts:

a) Sensor nodes are organized into clusters. Within each cluster, a cluster-leader is elected periodically. To save energy, only the cluster-leader may be awake, monitoring and reporting on any events occurring within its sensing range.

b) Artery is formed in the sensor network. In a ring shape, Artery is made up of clusters around the center of the network. Initially, Artery nodes are positioned in the mid-way from the center to the boundary. One example of it is shown in Figure 1. Its main functions include spreading across Artery event information gathered by sensors and matching up query requests with events.

c) Paths are established between Artery and sensor nodes, that is, every sensor node finds a path to at least one Artery leader. This is accomplished by Artery leader nodes broadcasting "path establishment" packets that travel to the boundary of the network and establish reverse paths along the way.

d) When a sensor node discovers an event, it collects related information and reports the event to Artery, using the path gradients established in c). The Artery node that received the information then floods only the Artery with the event.

e) A sink collects event information by broadcasting queries which can then be picked up by the leader of the cluster in which the sink resides. The cluster-leader forwards the query to Artery using the path gradients established in c).

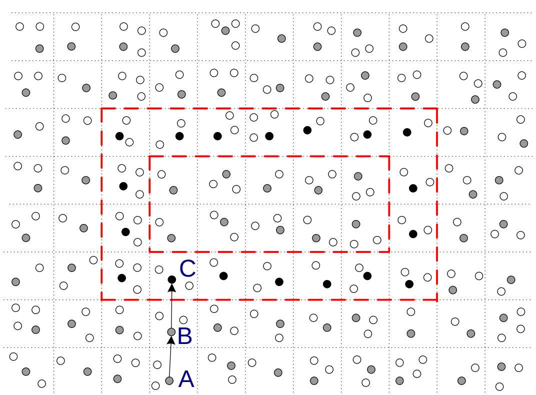

Figure 1. Sensor nodes are organized into clusters. Each cluster has a cluster-leader (gray node). Artery is bounded by the two rectangular shapes in the middle of the network. Black nodes are Artery leaders. Every sensor node has a path to at least one Artery leader. For example, A->B->C.

3. ARTERY ARCHITECTURE

3.1 Cluster Formation

Clusters are formed such that all nodes in two adjacent clusters can communicate with each other. Suppose clusters are in square shape with side length of x, and the sensor's transmission range is r. Then to enable the property, we have $x < \frac{r}{\sqrt{5}}$ as illustrated in Figure 2. For example, if the sensor's transmission range is 100m, the side length of square cluster can not be greater than 44.7m.

It takes several steps for sensors to organize themselves into local clusters. First, sensors are densely deployed in a randomly manner in a large region. One additional sensor, carrying the information of the approximate width and length of the field, is deployed in the center of the field. We call it the seed node. Based on the transmission range of the sensors, the seed node is able to draw up a grid that covers the whole sensor field. The seed node floods the network with a reference packet

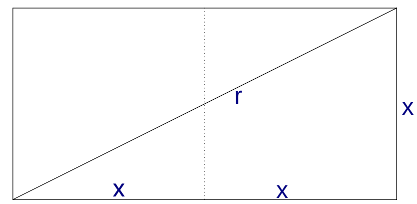

Figure 2. If the cluster is in square shape,

we have $x^2 + (2x)^2 < r^2 \Rightarrow x < \frac{r}{\sqrt{5}}$

containing its own location information which is used as a reference point. Upon receiving the reference packet, each node compares its own location with the location in the packet, and calculates its own cluster ID. Note that the cluster establishment is a one-time only process.

A cluster is composed of sensors with the same cluster ID. Periodically, each cluster elects a cluster-leader. Each node may probabilistically elect itself to be the cluster leader by broadcasting a "leadership intention" packet. After the packet is received and acknowledged by other nodes in the same cluster, a new leadership role and a new round start.

Since only the cluster leader is active during one round, all other non-cluster-leader nodes can be powered off, thus saving considerable amount of energy. Besides, all the data generated within a cluster can be aggregated by the cluster leader, greatly reducing the amount of outbound data.

Our cluster-leader selection algorithm is similar to the one used in LEACH[7]. The difference is that we have clusters with fixed size of boundaries but in LEACH each round of cluster generation may result in clusters of different sizes. Our scheme is better suited for data aggregation and fits into the Artery architecture well.

3.2 Artery Formation

Initially, Artery is located in the center of the network. Started from the seed sensor, the Artery is growing outwardly until it reaches the mid-way between the center and the boundary. An analogy of this is the ripples. Dropping a rock into a pool, you see rings of ripples. Initially the seed broadcasts messages asking its neighboring nodes to form a ring. After this is done, nodes in the ring broadcast messages asking its outward neighbors to form a new ring. The seed sensor node is not necessarily to be deployed in the center of the network. For certain deployment environments when the center is not reachable, the seed sensor can be placed on the boundary of the field. However, this requires different formation scheme.

After Artery is formed, Artery cluster leaders enter Path Establishment Phase in which they attempt to establish paths to all non-Artery cluster leaders. First, they broadcast a "build path" packet with hop counter set to zero $\langle num\_of\_hops\_towards\_artery = 0 \rangle$. On receiving the packet, each node increments the hop count by one and compares it to its own counter $\langle num\_of\_hops\_towards\_artery \rangle$. If its own counter is smaller than the hop counter in the packet, it discards the packet. If it is larger than the one encoded in the packet, it has found a shorter path to Artery. It then updates its own counter and sets its next_hop pointer to the neighbor from which the packet was sent.

After the Path Establishment Phase, every node in the network has path gradients toward Artery.

3.3 Routing Scheme

After a sensor discovers an event, it starts collecting data and sending reports to Artery. The complete path to Artery is not stored in the sensors since it can be costly and the path is prone to constant changes. A sensor only saves the information of its neighbors who are one-step above and below it on the path. This information is available after Path Establishment Phase described in the previous section. After receiving the data from its neighbor, the sensor searches its own memory and forwards the data to its neighbor on the path toward Artery. Since all the nodes on the path have only local information about the path, when any nodes are rotated out as cluster-leader or need to power themselves off due to low battery, they only need to turn over the information of its two neighboring nodes to its replacement in order to reestablish the connections.

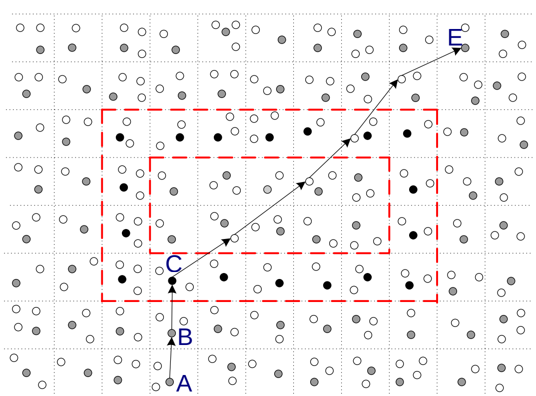

Figure 3. Sink A queries for event originated at E. First, A forwards the query to Artery cluster leader C via path A->B->C. Since C is an Artery leader, it knows the location information of all the events, therefore, it knows the location of target event E, and uses GPSR[8] to forward the query towards E.

It takes a limited number of transmission hops before the data reaches an Artery node. That Artery node then broadcasts the event information across the whole Artery. It first sends the data to its right-hand neighbor. The recipient node checks if it had received it before. If not, it saves the data and forwards it to its right-hand neighbor until the data packet is returned to the original sender. Otherwise, the data packet is considered obsolete and is dropped.

If one of the Artery nodes dies, Artery will still be connected but not in a ring shape. The data packet will not be circulated and returned to its original sender. A timestamp is needed to help determine whether the packet is considered lost and should be sent in the other direction.

When a sink collects data from the sensor field, it first broadcasts a query request within the cluster where it resides. Since the leader of the cluster is always awake, it picks up the query and forwards it toward Artery in the same way as event information is forwarded. When the query reaches Artery, the Artery node checks its memory to find if there is a match of event information. If so, it sends the event data back to the sink along the reverse path. If there is no data found, the Artery node broadcasts the query in Artery in the same way as an event is broadcasted.

In case the Artery node cannot find a match for the query, and it has known the location of source event information, which is considered as the "minimum knowledge" of the events, the Artery node can make use of the event source location information and use that in combination with GPSR[8] to forward the query to the original event source. This query delivery process is illustrated in Figure 3.

3.4 Floating Artery

From what is described above, we can see that Artery nodes take a much heavier load of responsibilities than non-Artery nodes thus consuming a lot more energy. If Artery is fixed and all nodes have the same amount of battery power, the Artery nodes will die much earlier than non-Artery nodes. The network will then be partitioned into two, an inner and outer part, disconnected from each other. To solve this problem, we propose a slightly different design – a structure we call Floating Artery. When the energy level of an Artery node is lower than certain threshold, the node transfers its duties to one of its non-Artery neighboring node. First, it contacts its two non-Artery neighbors to see if their energy levels allow them to take over the job. If not, which means the neighbors have less energy, the Artery node has no choice but to remain on the job. If one of its neighbors is capable of the duty, the Artery node turns over all the information it maintains, including its left and right neighbors in Artery, all its non-Artery nodes connections, and all the event data and queries it keeps in the memory. Then the new Artery node establishes the connections and all the old paths are resumed. The whole handover process uses only local information.

It is crucial to maintain the connectivity of Artery since it is the "heart" of the whole network. The reason for choosing the ring shape is that it remains connected even when one of the nodes dies accidentally. After a replacement for the dead Artery node is elected, the nodes originally connected to the dead Artery node can be switched to the replacement node. Since the neighbors of the dead node were close enough to listen to and to take notes of all of its transmissions, it is suitable to pick one of the neighbors to be the new cluster-leader so that no information is lost because of the accident.

4. PERFORMANCE EVALUATION

We implement the Artery routing protocol in ns-2 [9]. We compare Artery with Rumor routing, a recent routing protocol for wireless sensor networks. In order to make the comparison, we also port the implementation of rumor routing from lecsSim[10] to ns-2. The ns-2 implementation of Rumor routing [11]deviates from the original implementation quite a lot, due to the nature of ns-2 platform. However, we have tried our best to preserve the logic of the original algorithm. We use the optimization techniques for Rumor routing described in [12], thus the version of Rumor routing we are comparing with is the optimized one. To our best knowledge, ours is the first implementation of both algorithms in ns-2.

In both Artery and Rumor routing protocols, we use the same energy models as adopted in ns-2.1b9a, and its underlying 802.11 DCF MAC. A sensor node's transmitting and receiving power consumption rates are 0.66w and 0.395w. We do not count the idle power consumption because it largely depends on the system load , query and event timing, yet it constitutes the major portion of power cost, thus it does not reflect the true performance of protocols. We set sensor node's transmission range as 100 meters and cluster grid side as 44 meters to enable any two nodes in adjacent grids to reach each other.

The default simulation setting has 4 event sources and 10 sinks. We simulate 5 topologies of different sizes. They are $200x200m^2$, $400x400m^2$, $600x600m^2$, $800x800m^2$, and $1000x1000m^2$, with 44, 178, 401, 713, 1000 nodes respectively. The number of nodes for each topology is so chosen such that the node density remains roughly constant. We have also tested other topologies and node densities, and have observed similar results; therefore, the results presented here are representative of a wide range of settings. Each simulation result is averaged over three random topologies of each fixed size.

We use four metrics to evaluate the performance of Artery routing. **Path length** is defined as the average number of hops queries take to reach their respective event sources from their origins. It indicates the quality of path gradients in hops. **Delay** is defined as the average time between the moment a sink originates a query and the moment the query is successfully delivered to the event source. Delay, same as Path Length, is averaged over all source-sink pairs in all topologies. Since data packets take reverse paths from event source to sink, delay indicates the time efficiency of the routing algorithm as well as the freshness of data packets. **Energy consumption** is defined as the total communication (transmitting and receiving) energy the network consumes. In [12], energy consumption is modeled as proportional to the number of transmissions only. This can be very inaccurate because though the unit cost of transmitting is larger than that of receiving, the sheer number of receiving nodes in a node's transmission range makes the cost of receiving not negligible. In our simulation, we are able to evaluate the energy cost more accurately, considering both transmitting and receiving cost of all nodes. **Success rate is** defined as the ratio of the number of successfully delivered queries over the total number of queries generated in the first place, averaged over all source-sink pairs.

Path Length

We denote Artery-R as the variant of Artery routing protocol which restricts routing among grid leaders, i.e., a sensor node can only select next-hop grid leader neighbor to route a packet. This scenario happens if we want to turn those non-grid leader nodes off to achieve further power savings. We denote Rumor routing with parameter number of agents equals to n as Rumor_n.

Artery routing consistently performs better than other alternatives, even when the number of agents per event source is increased to 10. At $800x800^2$ and number of agents per source event equals to 2, Rumor routing produces paths 44% longer than that of Artery routing. This shows that Artery routing coupled with GPSR is able to deliver query via near-optimal paths, which consist of the shorter paths from sink to the Artery, and the shortest paths from Artery to event sources.

Artery-R performs better than most Rumor routing configurations, but not as well as Artery routing, because it restricts its next hop neighbor selection among grid leaders, though in some circumstances there may be non-grid leader nodes closer to the event source. This shows that if we keep only the grid leader awake, and power off the rest of the nodes in a grid, we may save energy, but at the expense of longer paths.

Path Length	200x	400x	600x	800x	1000x
Artery	1.60	2.87	4.00	5.20	7.00
Artery_R	1.93	3.20	4.45	6.05	7.55
Rumor_1	1.60	2.90	5.83	7.26	7.83
Rumor_2	1.60	2.93	5.50	7.51	7.21
Rumor_3	1.60	2.77	4.70	6.63	8.56
Rumor_4	1.60	2.87	5.07	7.10	8.44
Rumor_5	1.60	2.80	4.63	6.54	7.44
Rumor_6	1.60	2.90	4.63	6.97	7.78
Rumor_7	1.60	2.83	4.53	6.67	8.63
Rumor_8	1.60	2.77	4.60	6.11	8.47
Rumor_9	1.60	2.77	4.83	6.09	9.12
Rumor_10	1.60	2.83	4.83	6.83	8.25

Table 1. Path length in hops.

We also observe that in smaller topologies such as $200x200^2$, $400x400^2$, the advantage of Artery routing is not obvious. This is because when topologies are small, the average distance between all nodes is short, thus the advantage of building an Artery and routing packets by first tapping into the Artery is not very high.

We also observe that the performance of Rumor routing is inconsistent, i.e., increasing the number of agents per event source doesn't always yield better results. This is due to the nature of path gradients establishment. In Rumor routing, each agent chooses its path randomly and the timing of the paths overlay affects the quality of path gradients.

Delay

The average query delay is closely related to path length. From the results we observe that Artery routing

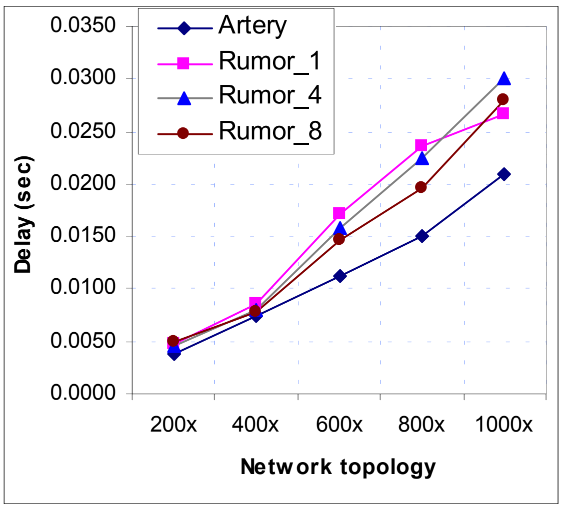

Figure 4. Delay comparison.

consistently performs better than Rumor routing with improvement between 10% and 56%. Artery routing is able to deliver queries with less delay due to its near-optimal paths.

Energy consumption

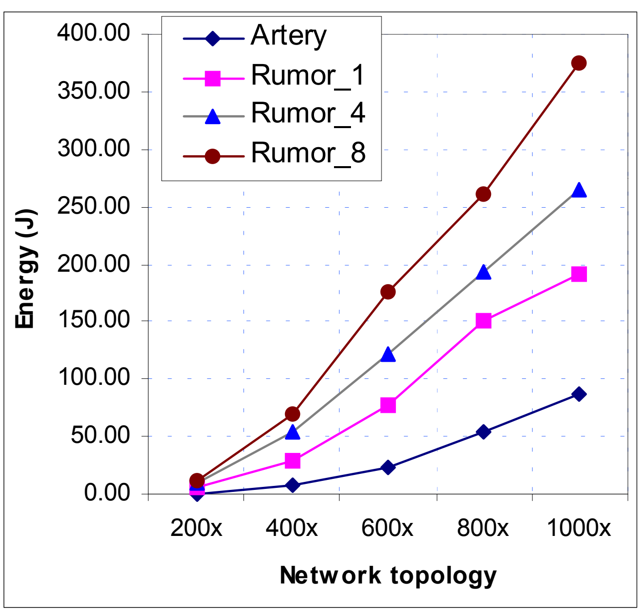

Figure 5. Energy Consumption. For clarity, we only show Rumor routing with three variants of number of agents per event.

Total energy is counted from the setting up of event sources to the end of query phase. It doesn't include the Artery setting up phase, which consists of the setting up of Artery and path gradients leading to it. This is because the Artery setting up phase is an infrequent operation, thus it should not be amortized over the queries in short time span.

Artery costs less energy because for each event, it broadcasts the location packet to the Artery, while for Rumor routing each event source generates several agent packets, each to be broadcast for AGENT_TTL times. In Artery routing, suppose the topology is L x $L\,m^2$ with grid of l x $l\,m^2$ and suppose the Artery is in the middle of the topology, each side of the Artery is of $L/2l$ grids size, thus there are total of $2L/l$ number of grids. Therefore, each event is broadcast [hops of path between event source and Artery]+ $2L/l$ number of times while in Rumor routing, each event costs (AGENT_TTL * # of agents/per event) number of broadcasts.

Success rate

For Artery routing, the success rate is always close to 100% because a node, via path gradients to the Artery, can always first find the Artery to obtain the source event location, before using GPSR[8] to further reach the event source.

Notice that the success rate in Rumor routing is less than 100% when topology is greater than 600x600 m^2 , meaning the delivery ratio of Rumor routing varies and there is no guarantee even if the number of agents per event source is large.

5. DISCUSSIONS

An important feature of the Artery architecture is its usefulness in data analysis. Since Artery stores all the events and queries, it can act like a database. Queries concerning the whole network can be answered, for example, the total number of queries on a certain event within a certain period of time. In the performance study, we haven't considered data aggregation aspect of Artery, that is, we have compared the worst-case scenario of Artery with the best-case scenario of Rumor routing. According to LEACH [7], all the events/queries in a grid can be aggregated to the leader, with savings as high as (avg # of nodes per grid) * 100%. This indicates an even bigger performance improvement that Artery may achieve We plan to explore that in a companion paper.

6. RELATED WORK

A lot of research has been done on sensor networks in the past several years. One of the first research topics is energy-efficient data dissemination. Here are several algorithms that address this issue.

- Directed diffusion[13]– does an initial limited data flooding and sets up reverse gradients to reinforce the best path. It results in high quality paths, but requires an initial flooding for exploration.

- Rumor routing[12]– does not flood the network with queries or data. Paths from event are set up by randomly walking “agents” sent out from the source. Queries also randomly walk in the field until they encounter an event path. It is a highly efficient algorithm, but it does not guarantee 100% successful delivery and does not handle mobile sinks.
- Two-tier Data Dissemination[14]– builds a grid structure from data source so that mobile sinks may receive data continuously by flooding queries within a local grid cell. Grid is rebuilt frequently when the queried event is moving. We plan to compare Artery with this work soon.
- Data-centric storage [15]–names data and hashes names to certain geographic regions in the network. It can efficiently deliver queries to named events. But it relies on a global coordinate system and a geo-routing framework.

7. FUTURE WORK

We will work on determining the optimal values for the thresholds used in this paper, such as cluster size and transmission range. More rigorous proofs and simulations will also be conducted.

We will also investigate further the impact of mobile sinks on Artery and develop algorithms to efficiently maintain the structure of Artery and the paths to it. Besides, we’d like to pursue on the database features of Artery.

Scalability of a network is essential. By adding hierarchies to Artery, we expect that the architecture scales well in very large network topologies.

8. CONCLUSIONS

We have shown in this paper Artery, a data-centric architecture for sensor networks. Aiming at good improvement on data aggregation, Artery acts as a bridge linking together multiple event sources and network users efficiently. Floating Artery design ensures the longevity of the whole network. Simulation results show that Artery outperforms some existing major algorithms on data dissemination and that Artery is a feasible architecture for wireless sensor networks.

9. ACKNOWLEDGEMENTS

The authors would like to thank David Braginsky and Deborah Estrin at UCLA for providing the lecsSim[10] source code, based on which we developed the ns-2 simulation code of Rumor routing.

10. REFERENCES

[1] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: Application driver for wireless communications technology," presented at ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, Costa Rica, 2001.

[2] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," presented at MobiCOM '99, 1999.

[3] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for Ad Hoc routing," presented at MobiCom'01, 2001.

[4] M.-J. Lin, K. Marzullo, and S. Masini, "Gossip versus Deterministic Flooding: Low Message Overhead and High Reliability for Broadcasting on Small Networks," University of California, San Diego Technical Report CS1999-0637, November 1999.

[5] H. Wang, D. Estrin, and L. Girod, "Preprocessing in a Tiered Sensor Network for Habitat Monitoring," presented at EURASIP JASP special issue of sensor networks, 2002.

[6] J. Albowitz, A. Chen, and L. Zhang, "Recursive Position Estimation in Sensor Networks," presented at ICNP, 2001.

[7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Transactions on Wireless Communications, 2002.

[8] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," presented at the sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking(MobiCom), Boston, MA, USA, 2000.

[9] H. Wu, "http://www.cs.du.edu/~hwu/Artery/Artery.html."

[10] D. Braginsky and D. Estrin, "http://lecs.cs.ucla.edu/~daveey/art/code.html."

[11] H. Wu, "http://www.cs.du.edu/~hwu/RumorRouting/Rumor.html."

[12] D. Braginsky and D. Estrin, "Rumor Routing Algorithm For Sensor Networks," presented at First ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia, U.S.A., 2002.

[13] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," presented at MobiCom 2000, Boston, MA, 2000.

[14] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A Two-Tier Data Dissemination Model for Large-Scale Wireless Sensor Networks," presented at MobiCom'02, Atlanta, GA, 2002.

[15] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, "Data-Centric Storage in SensorNets," presented at First Workshop on Hot Topics in Networks (HotNets-I), 2002.