SHR: Stateless Hierarchical Routing for Dynamic Sensor Networks

Niwat Thepvilojanapong[†], Yoshito Tobe[‡], Kaoru Sezaki[††]
[†]Department of Information and Communication Engineering, University of Tokyo
7-3-1 Hongo Bunkyo, Tokyo 113-8654, Japan
Tel & Fax: +81-3-5452-6268, E-mail: wat@mcl.iis.u-tokyo.ac.jp
[‡]Department of Information Systems and Multimedia Design, Tokyo Denki University
2-2 Kanda-Nishikicho Chiyoda, Tokyo 101-8457, Japan
E-mail: yoshito@unl.im.dendai.ac.jp
[††]Center for Spatial Information Science, University of Tokyo
4-6-1 Komaba Meguro, Tokyo 153-8505, Japan
Tel & Fax: +81-3-5452-6268, E-mail: sezaki@iis.u-tokyo.ac.jp

ABSTRACT

In this paper, we present *Stateless Hierarchical Routing* (SHR), a new routing protocol for multi-hop, wireless sensor networks. The design of the protocol is based on requirements of sensor networks that every sensor node periodically transmits sensed data to the base station. Tree construction is initiated by the base station which will broadcast `HELLO` packet to discover child nodes. Sensor node receiving this packet decides an appropriate parent node to which it will attach, it then broadcasts `HELLO` packet to discover child nodes in the next level of tree. Consequently, hierarchical tree is rapidly created without flooding of any control packets. By knowing only parent address, each node can make forwarding decisions regardless of knowledge on other neighbors or geographical location. When comparing to other proactive routing protocols, SHR avoids periodic updating for routing maintenance but it can agilely recovers from link failures by switching to a new parent. We evaluate the performance of SHR by using ns-2 simulator and comparing its performance with that of both DSR and AODV. Simulation results demonstrate that SHR has much higher delivery ratio and lower delay on various situations, both static and dynamic networks.

Keywords: wireless sensor networks, routing protocols, hierarchical tree, performance evaluation, simulation

I. INTRODUCTION

Recent advances in MEMS-based sensor technology and low-power RF design have enabled the development of relatively inexpensive and low-power wireless sensors. A great number of such sensors can coordinate amongst themselves to achieve a larger sensing task both in urban environments and in inhospitable terrain. They can be used in various applications such as environmental monitoring, tracking system, failure detection, intrusion detection, *etc*. One example of ongoing work is habitat monitoring on the Great Duck Island [1]. Due to specific communicating pattern in such sensor networks, they need to be structured differently from traditional mobile ad hoc networks (MANETs).

To motivate the challenges in designing routing protocol, we show a scenario usually happens in any sensing application. A large number of sensors (over one thousand sensors, for example) are deployed in remote terrain. These sensors coordinate to establish a communication network, monitor specified tasks, and report sensed data periodically or on-demand to the base station. When existing sensors are out of order due to numerous reasons, they reorganize by themselves to repair failed routes. The user may deploy additional sensors to mitigate severe effect of many failed nodes, thereby enforcing sensors to reconstruct in order to take advantage of the added system resources. Hence, we consider routing protocol based on specific communication pattern which is also robust to dynamic sensor networks.

The remainder of the paper is organized as follows. Section II enumerates the detailed mechanisms of our routing protocol. Section III evaluates the performance of proposed protocol in simulated networks through ns-2 simulation tool. Section IV describes related work and we summarize our work in Section V.

II. THE PROTOCOL

We design *Stateless Hierarchical Routing* (SHR) protocol for large-scaled, dynamic sensor networks. First, we describe specific pattern of communication required in sensor networks as well as dynamic natural of sensor networks. Since SHR is based on hierarchical tree, we then present how tree is hierarchically constructed and how sensor nodes adapt to dynamic networks. The details of SHR protocol are as follows.

A. Network Model

We consider sensor networks composed of a small number of base stations or sinks and a numerous number of wireless sensors randomly distributed in an interesting area. These sensor nodes have limited processing power, storage, bandwidth, and energy, while the base stations have powerful resources to collect and process sensor readings. We assume that sensor nodes are not mobile nodes, *i.e.*, all nodes are fixed for the duration of their lifetime, however, sensor network we consider has dynamic characteristic such that new nodes may be deployed at any time or battery of the node is depleted with time. In particular, sensor nodes have omni-directional antennas and use RF to communicate. All wireless network transmissions are inherently broadcast and a node may be able to configure its network interface into promiscuous receive mode.

We design routing protocol for sensor networks whose communication pattern differs from conventional mobile ad hoc networks. Let N be a set of all nodes in the network except the base station (BS). Previous work [2, 3, 4] is point-to-point routing protocols for a set of communicating parties (s, d), where $s \in \{N, BS\}$ and $d \in \{N, BS\}$, while our work is a routing protocol for multipoint-to-point communication, where $s \in \{N\}$ and $d \in \{BS\}$. Namely, every sensor node tries to report sensed data to the base station.

B. Tree Construction

Our routing protocol is based on hierarchical tree where a base station is a root node, and sensor nodes are the internal and leaf nodes of the tree. The base station initiates tree construction by broadcasting[1] two *child request* (`CREQ`) packets separated with interval T_i. Using two broadcast packets increases reliability of the protocol because broadcast packet is prone to lose and no any retransmission mechanism supports. *Nonmember node*, a node which does not attach to the tree yet, determines its parent from received `CREQ` packets by choosing a node whose `CREQ` packet has arrived first as a parent or waiting for T_{creq} seconds in order to collect a number of candidates and choose a node whose metric is the best one (highest received power strength, highest remaining energy, for example). It then sends a *child reply* (`CREP`) packet to selected parent so as to inform that it will be a child node or a leaf node of current tree. *Member node* which is an internal or leaf node drops `CREQ` packet immediately. In our implementation, nonmember node waits for a short period of time to collect candidate parents and choose node whose power strength is highest and more than a threshold in order to avoid the problem of communication gray zones [5].

Upon receiving `CREP` packet, parent node notifies an acceptance of new child node by replying with *child acceptance* (`CACP`) packet. Child node waits `CACP` packet for a period of T_{cacp} seconds and if `CACP` packet does not arrive within this period, it sends second `CREP` packet. If T_{cacp} period has passed again and `CACP` packet still does not arrive, it sends third `CREP` packet as a last reply and chooses a new parent for the next round of T_{cacp} period. After receiving `CACP` packet from the parent, child node does the same process as its parent by broadcasting `CREQ` packet to discover its children. These procedures are performed by every node in the network. An example of hierarchical tree created by SHR is shown in Figure 1.

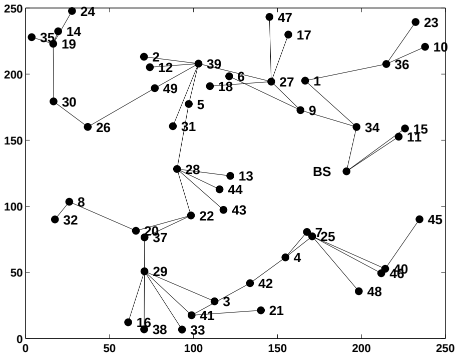


Figure 1: Hierarchical tree created by SHR protocol.

C. Joining Mechanism

Joining in sensor networks means user deploys new sensors into the current network. Newly deployed sensor or nonmember node must find a parent for communicating purpose by broadcasting *parent request* (`PREQ`) packet. Any member nodes of the tree that hear this packet reply by unicasting `CREQ` packet to joining node. Note that this `CREQ` packet is not broadcasted as described in Section II-B, therefore only one packet is sufficient. Then, the processes will follow tree construction phase, *i.e.*, joining node sends `CREP` packet to selected parent and waits for `CACP` packet before any communication can begin. We can also limit the number of children per parent in order to distribute loads and reduce congestion. If joining node does not receive any `CREQ` packet, it can infer that no any node is within its radio coverage or all of its neighboring nodes do not attach to the tree yet. In this case, it waits for incoming `CREQ` packets after one of its neighbors has attached to the tree.

D. Leaving Mechanism

Leaving means member node loses communication with other members due to numerous reasons. For instance, battery of the node is depleted with time, the node can be damaged due to harsh environment or by the enemy. Before explaining the processes in reconstructing the tree, we give some ideas on how to detect left node. As a common characteristic of sensor networks, sensor periodically transmits its sensed data to the base station. When combining this characteristic with our hierarchical protocol, if parent node does not receive packets from any children for a while, it infers that such children have completely left from the network. Another method can be done by relying on the underlying MAC sublayer protocol.We use the latter method (MAC approach) which is less complex to detect left nodes in our implementation.

If there is no acknowledgement on MAC sublayer after sending data packet to the parent, a node infers that its parent has left from the network due to any reason. It im-

---

[1] *Broadcasting* means the transmission of a packet from a source node to every node within its radio coverage. We use this definition throughout the paper.

mediately switches to a new parent by choosing the most appropriate one from candidate list and sending `CREP` packet to selected parent. If there is no any parent in the list, it broadcasts `PREQ` packet as if it is a newly deployed node. However, its child nodes will not reply to this `PREQ` packet to prevent routing loop. Every node that attaches to the orphan node does nothing because they do not know the absence of their grandparent. They can still forward packets to the orphan node as usual, and the orphan node keeps received packets in its buffer for sending later. In the worst case that orphan node does not have any parent in the list and no any response to `PREQ` packet, it broadcasts *parent query* (`PQRY`) packet to its child nodes asking whether they have candidate for parent node. Child nodes reply with *parent reply* (`PREP`) packet containing such information. Then, orphan node randomly chooses a child node that has at least one candidate parent as its new parent by sending `CREP` packet to inform new relation, and that child node will switch to a new parent chosen from the list. If all of its child nodes do not have any candidate parent, the orphan node randomly chooses one child node as a new parent by sending `CREP` packet as usual and let this selected child node find new parent by using `PREQ` packet. Note that the last scenario is very rare case that may occur in sparse network.

E. Data Communication and Discussions

A great advantage of our protocol is that it relies only on the knowledge of parent node. Therefor, the state required at each node is negligible, and independent of network density and network size which means that our protocol is very scalable. In particular, each node just forwards its sensed data and all of received packets to its parent. Thereby, our protocol is nearly stateless, *i.e.*, only one parent address suffices for routing. Routing table and geographical information are also not necessary. Route discovery does not use flooding, thereby no propagation of routing information or packets throughout the network. Moreover, our protocol does not apply periodic updating that reduces traffic load so much.

Since a main cause of energy consumption in sensor networks is communication cost compared to computational cost (transmitting a single bit of data is equivalent to 800 instructions [6]), we can decrease energy consumption by minimizing the number of transmissions. Instead of forwarding packets immediately, each node waits for a short period of time to *append* or *aggregate* data from other nodes and sends them together in one packet. Let us assume T_f denotes basic forwarding period and F_d denote delay factor, where $F_d \geq 0$. Node forwards data every $(T_f \times F_d)$ forwarding period. If $F_d = 0$, node forwards data as soon as it has new reading or receive new data. However, if the amounts of data fill up vacant space in one packet, node will immediately forward the packet regardless forwarding period. Aggregation [7, 8] which is a summarization of data is more complex than appending and depends on applications. Data compression [8] can also be done to reduce packet size as well as network loads.

The protocol described in this section is summarized as pseudo-codes in Algorithm 1 through 3. $flag\_prt$ is a flag indicating the existence of parent node. This flag is initially set to DOWN (line 5) which means that node does not have parent yet and it will turn to UP when nonmember node has attached to the tree. num_crep is the number of `CREP` packets the node sent. This variable is used to determine timeout of selected parent and it is initially set to 0 (line 6). When the sensor node is deployed in the field, it immediately broadcasts `PREQ` packet (line 7) and wait for incoming packets as described in Section II-C. The incoming packets may be both of data and routing packets. The node follows line 10–14 in the case of data packets and it conforms line 16–48 for routing packets.

III. Performance Evaluation

To evaluate the performance of our routing protocol, we use the ns-2 [9] simulation tool to run a number of simulations described in this section. We compare the performance with two well-known ad hoc routing protocols, Dynamic Source Routing (DSR) [3] and Ad-Hoc On-Demand Distance Vector (AODV) [4], which have been shown to offer higher packet delivery ratio than other ad hoc routing protocols [10].

A. Methodology and Metrics

The ns-2 simulator includes full simulation of the IEEE 802.11 physical and MAC layers. Our simulations use this MAC layer and assume symmetric links. We note that using this MAC layer does not affect evaluation result because we need to evaluate network layer of three protocols. We randomly placed 50 sensor nodes in a 250m by 250m square region. Each node has fixed radio coverage of 50 meters. Note that nodes have fixed positions without any movement for the entire simulation. We use constant bit rate (CBR) as our traffic sources. A 64-byte data packet is used for all CBR sources. The transmission rates of CBR sources are 0.25, 0.5, 1, and 2 packets per second, *i.e.*, 128, 256, 512, and 1024 bps respectively, while the bandwidth of sensor nodes is set to 19.2 kbps[2]. The CBR agent will be attached to a UDP agent, which in turn attached to the source node. For all simulations, the communication patterns are peer-to-peer and the starting time of connections is randomly selected. One node from each simulation is randomly chosen as a base station and it is only one destination for all traffic sources, while other 49 nodes are source nodes (one flow per one source). Each simulation is last for 200 seconds.

To compare between various protocols, we choose to evaluate them according to the following two metrics: data packet delivery ratio and average delay. *Data packet delivery ratio* is the ratio between the number of packets received by the destination and the number of packets sent by the source. *Average delay* measures the average

---

[2] Mica2 has bandwidth of 38.4 kbaud encoded with manchester code [11].

Algorithm 1 Main algorithm of the SHR protocol operation. Note that description within $\{\ldots\}$ is comment.

```
$flag\_prt$: whether the node has parent {DOWN, UP}
$num\_crep$: the number of CREP packets sent
$rcv\_pkt$: received packet
void $main()$ {
$flag\_prt \leftarrow$ DOWN
$num\_crep \leftarrow 0$
Broadcast PREQ packet
while $rcv\_pkt$ do
  if $rcv\_pkt$ is data packet then
    if $flag\_prt =$ UP then
      Forward $rcv\_pkt$ to parent
    else { $flag\_prt =$ DOWN }
      Buffer $rcv\_pkt$ in queue {The node drops packet if buffer is overflow}
    end if
  else {$rcv\_pkt$ is routing packet }
    if $rcv\_pkt$ is CREQ packet then
      if $flag\_prt =$ UP then
        Drop CREQ packet but keep this communicating party in parent candidate list
      else { $flag\_prt =$ DOWN }
        Add this sender in the list as a candidate parent
        if The node received CREQ packet for the first time then
          Setup timer to choose parent (call $choose\_parent()$) at $T_{creq}$ seconds later
        else
          $choose\_parent()$ {Choose parent immediately}
        end if
      end if
    else if $rcv\_pkt$ is CREP packet then
      Add communicating party in children list
      Send CACP packet
    else if $rcv\_pkt$ is CACP packet then
      if $flag\_prt =$ UP then
        Drop CACP packet
      else { $flag\_prt =$ DOWN }
        $flag\_prt \leftarrow$ UP
        $num\_crep \leftarrow 0$ {Reset this variable for future parent selection}
        Send all buffered packets in queue {Route is available now}
        Broadcast CREQ packet to find child nodes in the next level
      end if
    else if $rcv\_pkt$ is PREQ packet then
      if $flag\_prt =$ UP then
        Unicast CREQ packet to communicating party
      end if
    else if $rcv\_pkt$ is PQRY packet then
      Send PREP packet containing necessary information
    else if $rcv\_pkt$ is PREP packet then
      Choose a new parent from its children according to received information
      Send CREP packet to inform new relation
    end if
  end if
end while
}
```

Algorithm 2 $choose\_parent()$ function used in the SHR protocol.

```
void $choose\_parent()$ {
Choose node whose metric is the best as a parent
$num\_crep \leftarrow num\_crep+1$ {This variable is used as timeout}
Send CREP packet to chosen parent
Call $wait\_cacp()$ if CACP packet does not arrive within $T_{cacp}$ seconds
}
```

Algorithm 3 $wait\_cacp()$ function used in the SHR protocol.

```
void $wait\_cacp()$ {
if $num\_crep > 2$ {The node does not receive CACP packet from selected parent after sending 3 CREP packets} then
  Delete selected parent from candidate list
  $num\_crep \leftarrow 0$ {reset this variable}
  if There is at least one member in parent candidate list then
    $choose\_parent()$
  else
    Periodically broadcast PREQ packet until getting CREQ packet
  end if
else
  $num\_crep \leftarrow num\_crep + 1$
  Send CREP packet again
  Call $wait\_cacp()$ if CACP packet does not arrive within $T_{cacp}$ seconds
end if
}
```

one-way latency observed between transmitting a packet and receiving it at the destination. The parameters of our protocol used in simulations are set as follows: T_i = 0.1 second (the interval between two CREQ packets), T_{creq} = 0.1 second (time waiting for collecting candidate parents), and T_{cacp} = 0.3 second (time waiting for CACP packet from selected parent).

SHR is a proactive routing protocol, while both of DSR and AODV are reactive protocol. We try to achieve fair comparison in the simulation for such opposite approaches. In general, SHR launches tree construction at the beginning of the simulation which is an advantage of proactive approach. In contrast, both reactive protocols discover the route when there is a packet destined for new destination. To make opposite approaches begin to discover route at the same time, every traffic source is forced to issue data packet at the beginning of the simulation[3]. However, both DSR and AODV show poor performance due to flooding of a large number of control packets. Therefore, we decide to randomly start each traffic source between 0^{th} – 50^{th} second of the simulation. Since there is only one destination (base station), each source node can use discovered route for the entire simulation.

---

[3] This is feasible issue in sensor networks when a number of sensors are deployed and start to sense information at the same time.

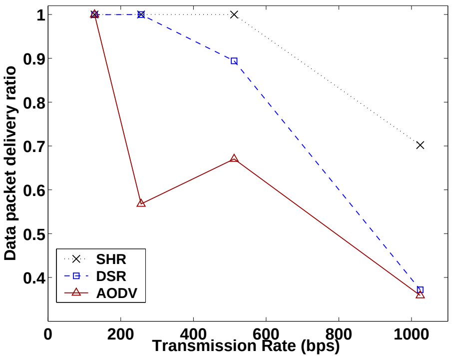


Figure 2: The fraction of data packets successfully delivered as a function of transmission rate.

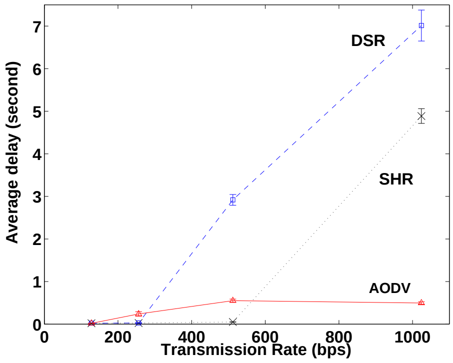


Figure 3: Average delay as a funtion of transmission rate.

Both approaches will discover new route when link failure is detected.

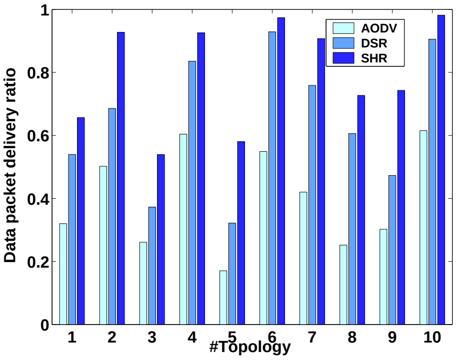
B. Simulation Results

Packet delivery ratio and average delay with 99% confidence level are shown in Figure 2 and 3 respectively. It is clearly from the figures that all of the protocols deliver a greater percentage of the originated data packets at low traffic loads. In particular, delivery ratio of three protocols is nearly 100% at 128-bps traffic load. When transmission rate is increased to 256 bps, DSR and SHR can still deliver nearly 100%, while the performance of AODV dramatically degrades to 57%. When we increase load to 512 bps, only SHR can still deliver nearly 100%, while the delivery ratio of DSR drops to 89% and AODV can deliver only 67%. At the extreme case in our simulation, *i.e.*, 1024-bps load, the delivery ratio of SHR sharply drops to 70% which is still much better than one-third delivery ratio of DSR and AODV which is unacceptable value for any applications. The main reason of dropped packets is no available route because of high congestion and high collision due to a large number of control and data packets. When considering average delay, SHR delivers slightly faster than DSR at both 128- and 256-bps load. However, SHR still does very well at 512-bps load, *i.e.*, less than 0.05 second, while DSR delivers at an average of 3 seconds. As one would expect, SHR delivers faster than DSR about 2 seconds at 1024-bps load. AODV has lower delay than both SHR and DSR at high load because there are many dropped packets especially packets that traverse long distance. Such dropped packets are not included in delay calculation.

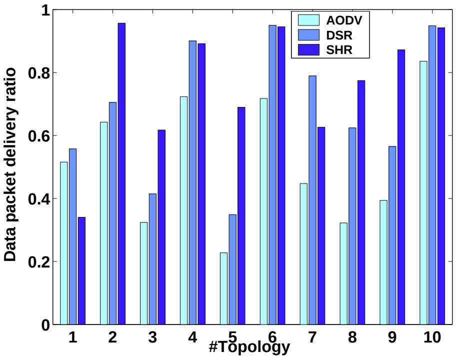
We also simulate two scenarios of dynamic networks, joining and leaving scenarios. For joining scenario, 10 nodes are randomly deployed at 100 second in addition to 50 nodes deployed at the beginning of simulation. These additional nodes need to find the parent for communicating. To simulate leaving scenario, we deploy 60 nodes and apply energy model in our simulation by providing much enough energy for entire simulations for 50 nodes and making battery of 10 nodes depletes at some point of time before simulations end. We show only the delivery ratio of 1024-bps load on ten random topologies which is appropriate for evaluating the resilience of protocol since it is extreme case. Other parameters (simulation area, radio coverage, etc.) are same as above simulations. The delivery ratio of joining and leaving scenarios are shown in Figure 4(a) and 4(b) respectively. For joining case, delivery ratio of SHR is about 10-20% better than DSR which in turn is better than AODV about 10-20%. When considering leaving case, SHR deliver greater than DSR for five topologies but DSR do better than SHR for two topologies. AODV is the worst among three protocols.

IV. RELATED WORK

DSDV [2], DSR [3], and AODV [4] are routing protocol developed for mobile ad hoc network which is very different from sensor network in an issue that nodes are mobile. Since MANET is dynamic, routing protocols must adapt to current physical topology by periodically updating states for proactive protocol (DSDV) or using reactive approach (DSR and AODV). Each node in such protocols keeps routing table for all destinations, in contrast, each node in SHR keeps only the knowledge of parent node. Moreover, SHR does not use periodical updating as DSDV or floods control packets as DSR and AODV. LAR [12], GPSR [13], and GEAR [14] are geographic routings that use location information to decrease overhead of route discovery and find routes quickly. Location-aware module increases production cost for sensor nodes, especially large-scale sensor networks. It wastes to use location-aware nodes in non-mobile networks. Note that geographical information is not required for routing in SHR. Directed diffusion [7] is a data-centric routing based on the name of data. Base stations draw interesting information by flooding interests and setting up gradients within the network. Directed diffusion also provides in-network aggregation. Directed diffusion is query-style protocol dealing with the name of data which is completely different from SHR. Therefore, directed diffusion is not appropriate for comparison with SHR.



(a) 10 nodes join.



(b) 10 nodes leave.

Figure 4: The fraction of data packets successfully delivered on ten randomly generated topologies in dynamic scenarios.

V. CONCLUSIONS

This paper has demonstrated that SHR protocol efficiently collects data packets across multi-hop wireless sensor networks while maintaining a constant amount of local state to nearly stateless and making only local decisions. Since the state required on each node is very low and independent of both network size and network density, SHR is highly scalable. In particular, there is no need to maintain the state about all neighboring nodes and the interactions between nodes are strictly local. Consequently, nodes can make quick decisions and hierarchical tree is agilely constructed. Our protocol also supports dynamic properties introduced in this paper. In other words, it is self-organized protocol according to joining or leaving nodes. Furthermore, it avoids flooding and periodic updating that incurs high traffic load. Geographical information is also not required in our protocol. We have examined the efficiency of SHR in terms of packet delivery ratio and average delay through ns-2 simulation. The results have shown that SHR achieves notably high delivery ratio and low delay, and it is also tolerable to high traffic loads. SHR is still in preliminary stage of development and it needs further investigations in many areas.

References

[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, “Wireless sensor networks for habitat monitoring,” in *Proc. of 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pp. 88–97, Sept. 2002.

[2] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,” in *Proc. of ACM SIGCOMM*, pp. 234–244, 1994.

[3] D. B. Johnson, D. A. Maltz, and J. Broch, “DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks,” in *Ad Hoc Networking* (C. E. Perkins, ed.), ch. 5, pp. 139–172, Addison-Wesley, 2001.

[4] C. E. Perkins, E. M. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) routing,” RFC 3561, IETF, July 2003.

[5] H. Lundgren, E. Nordstro, and C. Tschudin, “Coping with communication gray zones in IEEE 802.11b based ad hoc networks,” in *Proc. of International Workshop on Wireless Mobile Multimedia (WoWMoM)*, pp. 49–55, Sept. 2002.

[6] S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler, “Supporting aggregate queries over ad-hoc wireless sensor networks,” in *Proc. of the 4th IEEE Workshop on Mobile Computing and Systems Applications (WMCSA)*, pp. 49–58, June 2002.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in *Proc. of International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 56–67, Aug. 2000.

[8] D. Petrovic, R. C. Shah, K. Ramchandran, and J. Rabaey, “Data funneling: Routing with aggregation and compression for wireless sensor networks,” in *Proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, pp. 156–162, May 2003.

[9] “Network simulator - ns-2.” http://www.isi.edu/nsnam/ns/.

[10] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” in *Proc. of International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 85–97, Oct. 1998.

[11] “Mica2 mote.” http://www.xbow.com/index.htm.

[12] Y.-B. Ko and N. H. Vaidya, “Location-aided routing (LAR) in mobile ad hoc networks,” in *Proc. of International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 66–75, Oct. 1998.

[13] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks,” in *Proc. of International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 243–254, Aug. 2000.

[14] Y. Yu, R. Govindan, and D. Estrin, “Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks,” Technical Report UCLA/CSD-TR-01-0023, UCLA, May 2001.