Label-based Multipath Routing (LMR) in Wireless Sensor Networks

Xiaobing Hou, David Tipper and Joseph Kabara
Department of Information Science & Telecommunications
University of Pittsburgh, Pittsburgh, PA 15260
{xiaobing, dtipper, jkabara}@mail.sis.pitt.edu

Abstract

Current wireless sensor network routing protocols are still struggling to find valid paths between source and destination, and multipath routing for fault tolerance is quite a new research area and studied insufficiently. The multipath routing techniques designed for ad hoc network do not apply to the sensor network due to the lack of global ID in sensor networks. In this paper, we propose a novel approach called Label-based Multipath Routing (LMR) using only localized information. LMR can efficiently find a disjoint or segmented backup path to provide protection to the working path.

I. Introduction

A sensor network consists of a large number of densely deployed sensor nodes. The position of the sensor nodes is not usually predetermined, as the network may be deployed in inaccessible terrains or disaster relief operations. Therefore, the topology may be random. Some of the application areas of sensor networks are medical care, military, and disaster recovery/relief. Due to the large size of such networks compared to the transmission range of individual devices, routing protocols are necessary for end-to-end communication. Compared to ad hoc networks, sensor networks have some unique feature and application requirements [1]. First, they normally have more nodes, higher density, more limited power supply and computational capacity than nodes in mobile ad hoc networks. Second, sensor networks can be characterized as data centric networks, where users are interested in querying an attribute of the phenomenon, rather than querying an individual node. Third, sensor networks are application-specific in that the requirements on the network change with the applications. As an example, some applications require delay sensitive transmission, e.g., fire monitoring, whereas others do not, e.g., tempature control in an office building. Fourth, adjacent nodes might have similar data; therefore, sensor networks should be able to aggregate similar data to reduce unnecessary transmissions and save energy. Last, assigning unique IDs may not be suitable in sensor networks because these networks are data centric – routing to and from a specific node is not required. In addition, the large number of nodes requires long IDs and must be minimized to conserve power.

Presumably, the sensor network application requires reliable data disseminations. Given the unreliable nature of the wireless channel and the high failure rate of individual sensors [1], multiple paths are required to maintain reliability. Specifically, with current single-path routing protocols, fault tolerance can not be provided because the continuity of end-to-end communication can not be maintained without routing protection and restoration techniques. Studies done for ad hoc networks may not be applicable to the energy constrained multipath routing in sensor networks. We propose a novel approach, namely Label-based Multipath Routing (LMR) for sensor networks.

LMR broadcasts a control message throughout the network for a possible alternate path. During the process, labels are assigned to the paths the message passes through. The label information is used for segmented backup path search if a disjoint path is not achievable. Our analysis and simulation show that this label information can reduce the routing overhead and backup path setup delay.

The remainder of this paper is organized as follows. We present a brief review of sensor network routing in section II. Various of multipath routing techniques in ad hoc networks and sensor networks are surveyed in section III. In section IV, Label-based Multipath Routing (LMR) is proposed. The performance evaluation of LMR is presented in section V. We then conclude our paper in section VI.

II. Sensor Network Routing

Basically, there are two types of sensor network routing protocols in the literature, cluster-based and flat. Cluster-based routing schemes divide the network into clusters and utilize a sleep mode to save energy and prolong the network lifetime. Flat routing schemes try to reduce the routing overhead directly by using localized information only.

In cluster-based routing protocols, all nodes are organized into clusters with one node selected to be cluster-head for each cluster. This cluster-head receives data packets from its members, aggregates them and forwards data to a data sink. Examples of cluster-based routing protocols are LEACH [2], TEEN [3], and APTEEN [4].

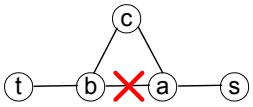

Fig. 1. Route repair

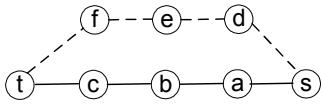

Fig. 2. Alternate routing

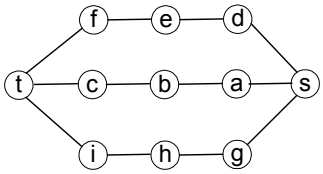

Fig. 3. Redundant routing

Low-Energy Adaptive Clustering Hierarchy (LEACH) [2] is designed for proactive networks, in which the nodes periodically switch on their sensors and transmitters, sense the environment and transmit the data. Nodes communicate with their cluster-heads directly and the randomized rotation of the cluster-heads is used to evenly distribute the energy load among the sensors. Threshold sensitive Energy Efficient sensor Network protocol (TEEN) [3] is designed for reactive networks, where the nodes react immediately to sudden changes in the environment. Nodes sense the environment continuously, but send the data to cluster-heads only when some predefined thresholds are reached. Adaptive Periodic Threshold sensitive Energy Efficient sensor Network protocol (APTEEN) protocol [4] combines the features of the above two protocols by modifying TEEN to make it send periodic data. The cluster-based routing protocols can arrange the sleep mode of each node to conserve energy at the cost of a high computational complexity and control overhead.

There are three types of flat routing schemes, namely, flooding, forwarding and data-centric based routing. Flooding is an old routing technique that can be used in sensor networks. In flooding, every node repeats the data once by broadcasting. It doesn't require costly topology maintenance and complex route discovery algorithms. But it has several deficiencies [1]:

- Implosion: duplicated messages are sent to the same node. A node with multiple neighbors may receive multiple copies of the same message.
- Overlap: if two sensors share the same observation region, both of them may sense the same stimuli at the same time. As a result, neighbor nodes receive duplicated messages.
- Resource blindness: flooding doesn't take into account the available resources, e.g. the remaining energy stored in the sensor node.

Forwarding schemes utilize local information to forward messages. Unlike the traditional routing protocols, forwarding doesn't maintain end-to-end routing information. Instead, intermediate nodes maintain only neighbor information. One example is the gossiping protocol [5], a node only forwards data to one randomly chosen neighbor, so it doesn't maintain any routing information or we can say it uses randomness to forward data. Best Effort Geographical Routing Protocol (BEGHR) [6] employs position information to forward data, and therefore requires GPS or other positioning service. Field based Optimal Forwarding employs cost field to forward data [7]. A cost field is the minimum cost from a node to the sink on the optimal path. The sink node is the destination of all of the data in the network.

In data-centric based routing, an interest message is disseminated to assign the sensing tasks to the sensor nodes and data aggregation is used to solve the implosion and overlap problems [1]. There are two types of data-centric based routing based on either the sink broadcasts the attribute for data, e.g. Directed Diffusion [8], or the sensor nodes broadcast an advertisement for the available data and wait for a request, e.g. Sensor Protocols for Information via Negotiation (SPIN) [9].

III. Related Work

In wireless ad hoc and sensor networks, nodes may be weakly connected or damaged, so that links may be asymmetric or broken for some period time. Battery-powered nodes may die out or go to sleep to save energy. A fault tolerant routing protocol must expect and overcome these problems.

Fault tolerant routing mechanisms for ad hoc networks include *route repair* [10]. After detecting a break in link *a*-*b*, node *a* can repair the route by finding another node *c* so that *a*-*b* can be replaced by *a*-*c*-*b* as shown in Fig. 1. If node *a* can not repair the route, it sneds an error message to the source(s). However, the repaired route may be suboptimal and after only a few repairs, the route may be very long and inefficient. Second, it may result in loops unless a source routing protocol (e.g. DSR [11] [12]) is used.

Alternate routing is a scheme where the source searches for a full alternate route after a failure [10]. As shown in Fig. 2, if the working route (solid line) is broken, the source receives the notification of route unavailability from the intermediate nodes and establishes a new route (dashed line). Although compared to the route repair the new path is optimal, establishing the path requires even more time and overhead. Basic AODV [13] and DSR [11] [12] protocols are using this scheme.

Redundant routing establishes alternate paths before the failure happens [10]. In Fig. 3, multiple paths are created between the source *s* and the destination *t*.

Compared with alternate routing, this approach is able to reduce the rerouting overhead since finding multiple paths at the same time (called *multipath routing* in literature) is cheaper than finding them one by one. Also the rerouting delay is smaller since the alternate path is available before the failure happens. But multiple paths having the same age may be similarly unreliable at the same time for a mobile network.

Preemptive routing proposed in [14] can improve the alternate routing by discovering an alternate path before a working path breaks. When a path is likely to be broken, a warning message is sent to the source indicating the likelihood of a disconnection. The source then initiates path discovery early, potentially avoiding the disconnection altogether. With alternate routing, when a path break occurs, the connectivity of the flow is interrupted and a hand-off delay is experienced by the packets that are ready to be sent. Preemptive routing switches a traffic flow to an alternative good path *before* a break, minimizing both the latency and jitter. Mechanisms used in cellular networks, such as the signal strength, can be used to trigger path discovery. Other warning criteria such as location/velocity and congestion can also be used as the preemptive trigger [14]. This scheme may increase the routing overhead of alternate routing protocols since some path discoveries are being carried out proactively.

Neighborhood aware Source Routing (NSR) [15] reduces the effort required to fix working routes by using alternate links available in the two-hop neighborhood of nodes. The two-hop neighborhood information is maintained by exchanging link-state information among neighboring nodes proactively. The repair delay can be alleviated since the alternate links are known before the failure occurs. Of course, extra overhead is required to maintain the proactive two-hop link state updates.

The techniques discussed above are examples of multipath routing. In each case, as is common for ad hoc networks, a global ID system is assumed so that every node has a unique ID and different paths can be easily recognized. However, this may not be the case in a sensor network. The great number of the nodes and the very low data rate make a global ID an unbearable overhead. Therefore, a multipath routing using localized information only is desirable for sensor networks. Two schemes have been proposed employing Directed Diffusion. *Disjoint Multipath* tries to find a disjoint path by randomly pick a neighbor to ask for a backup path to the sink. The request is otherwise rejected [16]. *Braided Multipath* follows the same idea but tries to form a braid around the working path. Both employ a brute force search technique, so that if a disjoint path can not be found, there is no information left for Braided Multipath to take advantage of. In the next section, we propose a new scheme to better utilize the localized information.

IV. Label-Based Multipath Routing (LMR)

Wireless sensor networks typically consist of a large number of nodes and work at a very low data rate. Therefore, assigning globally unique IDs may be extremely expensive in terms of bandwidth and power consumption. Additionally, it's not necessary because these networks are data-centric – routing to and from a specific node is not required. Similar to Disjoint Multipath and Braided Multipath [16], LMR is designed to use only the localized information to find disjoint paths or segments to protect the working path. With one flooding, LMR can either find disjoint alternate paths or several segments to protect the working path. The flooding overhead is reduced by the associated schemes used by the underlying routing protocols, e.g., location information or cached data in Directed Diffusion [8]. LMR can work with different data-centric routing protocols, e.g., SPIN and Directed Diffusion. For clarity, we introduce it over Directed Diffusion and we assume there is no mobility.

Multipath routing has been widely studied in wireline networks [17], and one of the difficulties, which also arises in wireless sensor networks, is *trap topology* [18]. In a trap topology, the working path may block all the possible disjoint paths. For example, the working path *s-a-b-c-t* in Fig. 4a has no disjoint backup path, although two disjoint paths exist between s and t. There are two solutions. One is to route the working and the back paths simultaneously. This is very difficult in a network without global ID. The second is to select multiple partially disjoint path segments to protect the working path and that's the one we are using in LMR.

A. Label

In Directed Diffusion [8], the sink node broadcasts the attributes for data, termed *interest*. The intermediate nodes create a *gradient* directed to the node from which the interest is received. After the source receives the interest, it sends an *exploratory* data message to each neighbor for whom it has a gradient at a low data rate as shown in Fig. 4b. After the sink starts receiving the exploratory data, it *reinforces* one particular neighbor by sending a *positive reinforcement* message in order to "draw down" the data at a higher data rate as shown in Fig. 4c. Similarly, a *negative reinforcement* message is used to remove a link from a path. Multiple paths may be reinforced. But this is different from the multipath routing we are studying. Firstly, there is no way we can guarantee that for each node failure we have an alternate path to protect it. Secondly, requiring every node receive data from two or more upstream nodes may result in the prohibitively high total overhead.

In LMR, after the nodes on the working path reinforce one of their links as the link to form a working path, they broadcast a *label message* to the rest of their neighbors. Both the reinforcement and label messages

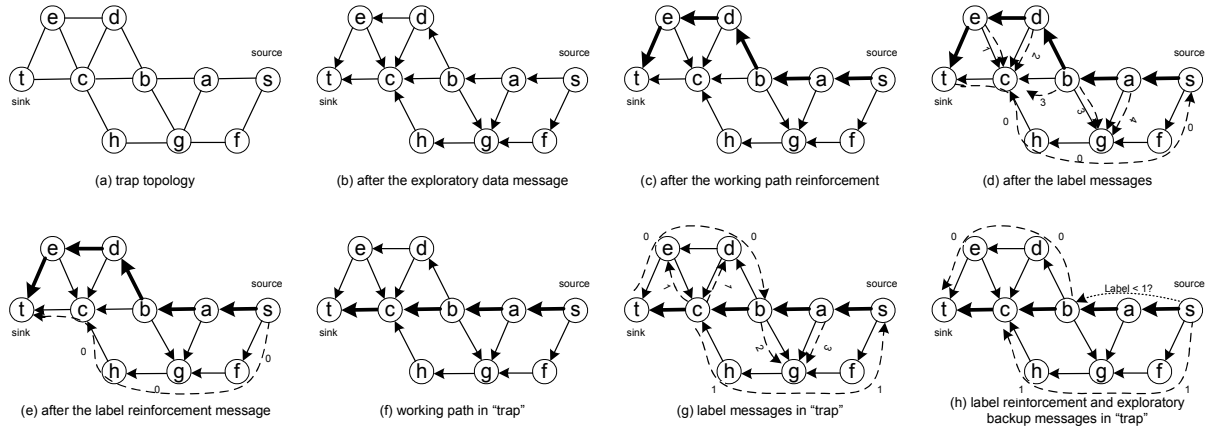

Fig. 4. Illustration of different aspects of LMR.

take an integer, termed *label*. The value of the label is increased by 1 by each working node which then broadcasts a new label message. Every working node should remember this value as its own *node label*. The label messages are forwarded towards the source along all the paths which the exploratory data messages pass through. A node receiving two or more label messages will forward the one with smaller label value only. The idea is to make the label message from the node closer to the sink go as far as possible so that the disjoint paths are possible to be found. The working nodes do not forward the label messages from any other nodes. Every node should remember all labels it has seen and the associated neighbors they are coming from. If a node receives multiple label messages with same label value from different neighbors, only the first one is recorded to find a shortest backup path. This process is shown in Fig. 4d.

B. Backoff algorithm

To avoid the excessive label message flooding, nodes must forward the smallest label message only. Therefore, a backoff algorithm is necessary to increase the probability that nodes receive the smallest label message before they start forwarding. A new label message should be delayed long enough so that a label message with a smaller label can go beyond this working node. Then the smaller label message will reach every node before the larger one if there are paths for them. However, if the delay is not long enough, the larger label message may reach the node first even with the delay. If the delay is long enough for a message to cover the entire network, we can guarantee that all nodes receive the smaller label first, but the setup delay of the backup paths may be long. So a tradeoff is necessary.

In LMR, if delay t_d is used, the working node with label w_i should broadcast a new label message after a backoff delay shown as follows,

$$T_i = w_i \times t_d \tag{1}$$

where, i=0, 1, ..., is the working node which has a new label message to broadcast and 0 is the sink. Another way to generate a new label is to make every working node increase the label by 1 no matter if it's necessary to broadcast a new label message or not. By this way, the node label $w_i = i$, and

$$T_i = i \times t_d \tag{2}$$

C. Label reinforcement

After the source receives a label message, it can immediately start label reinforcement process, since the backoff algorithm makes the smaller label message arrive first. A smaller label means we have a disjoint path segment to a working node closer to the sink. If a label 0 is received, that means we find a disjoint backup path. The source then sends a *label reinforcement* message to the node originating the label. The reinforcement continues with that node checking its memory to see which node this label comes from and then reinforcing that node. The process is a reversed reinforcement process of Directed Diffusion until the sink is reached, resulting in two disjoint paths (Fig. 4e). If the label received by the source is not 0, that means we may fall in a "trap" as shown in Fig. 4f. The label messages in this case are shown in Fig. 4g. Besides reinforcing a path segment, the source should send another message along the working path, called *backup exploratory* message. This message takes the label the source received. Any working node receiving this message whose node label is larger than this label either starts reinforcing a new backup path segment or forwards it. The new backup path segment should have a label smaller than the one the source received so that more working nodes can be protected. If the label of this new segment is not 0, a new backup exploratory message is initiated with the new label. The process is repeated until either a backup segment with label 0 is reinforced, or no new segments with smaller labels can be found, i.e. not all of the working nodes can be protected. This process is

shown in Fig. 4h.

After the backup path has been established, LMR may be repeated to find a third path. LMR can recursively find the *n* paths treating the first *n-1* paths as working paths.

V. Performance Evaluation

A. Complexity

To find the possible alternate paths, LMR incurs overhead, a flooded label message, and a label reinforce message and a backup exploratory message. We represent the sensor network as a graph $G = (N, E)$ with a diameter d in term of hops (i.e., the longest path between two nodes) and the average node degree is D. L_w represents the average length of a working path, and L_b the average length of a backup path. We consider the overhead of LMR based on two cases, local unicast, i.e. each node can only communicate with one of its neighbors at any time, and local multicast, i.e. each node can send a message to all of its neighbors at the same time. If the sensor network doesn't support local multicast but local unicast only, a node must send label messages to its neighbors individually and the number of the messages is in the order of D. If the network is not partitioned, almost all the nodes are involved except the source, therefore the label message overhead is $D \times |N|$. Since the label reinforcement message is disseminated along the backup path only, the total packet generated is L_b. Similarly, the backup exploratory message is sent along the working path only and the overhead is at most L_w. So the total overhead of LMR without multicast is $D \times |N| + L_b + L_w = D \times O(|N|)$. If the local multicast is supported, the label messages can be reduced by a factor of D. Therefore, the total overhead is $O(|N|)$ provided $N >> L_b + L_w$.

Disjoint Multipath and Braided Multipath try their neighbors one by one for the backup paths, so they can not benefit from local multicast and the complexity is same for two cases. or one failed try, two messages are involved, positive reinforcement and negative reinforcement [16]. Therefore the best case overhead is $L_b = O(d)$ and the worst case overhead is $2D \times O(|N|)$. It's worth noting that these two schemes are independent. If Disjoint Multipath fails, Braid Multipath must start over and double the overhead. LMR is efficient with local multicast and is reducing the average number of messages by $1/2D$.

Another measure of the performance of a multipath routing protocol is the delay to setup a backup path. We represent the link delay for transmission of one packet with t_p. LMR requires one round trip to set up a backup path and one of them may incur backoff delay. At the best case, a disjoint path can be found by the label message starting from the sink and no backoff delay is incurred, the total delay is $2L_b t_p = 2t_p \times O(d)$. At the worst case, all backoff delays occur at every hop and the total delay is $2t_p L_b + t_d L_w = (2t_p + t_d) \times O(d)$. Although Disjoint Multipath and Braided Multipath don't have a backoff delay, they may incur more link delay due to the brute force search. Obviously, their best case is half of LMR's. Their worst case may need to search every node in the network, so the total delay is $2t_p \times O(|N|)$. In a large network, since $O(|N|)/O(d) \gg t_d/t_p$, LMR can outperform the other two schemes in term of backup path setup delay. The above analysis is summarized in Table I.

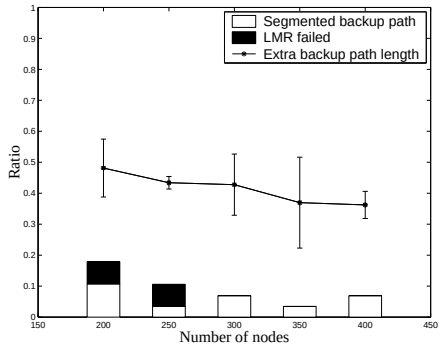

Fig. 5. Ratio of extra backup path length, segmented backup path and failure of LMR.

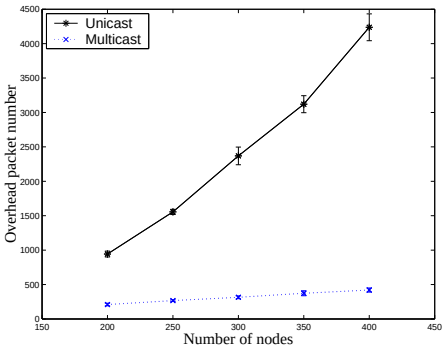

Fig. 6. Overhead of LMR.

B. Simulation

We utilized ns-2 network simulator [19], with CMU Monarch Project wireless and mobile ns-2 extensions, to study the characteristics of LMR. The distributed coordination function (DCF) of IEEE 802.11(b) for wireless LANs is used as the MAC layer. It uses Request-to-send (RTS) and Clear-to-send (CTS) messages and virtual carrier sensing for data transmission to reduce the impact of the hidden terminal problem. The radio model is similar to Lucent's WaveLAN, which is a shared media radio with a nominal bit rate of $2Mb/sec$ and a nominal radio range of 250 meters.

LMR is implemented over Directed Diffusion available in ns-2. The simulation results presented in this paper are based on scenarios randomly generated by CMU ns-2 extensions. We use 200 to 400 static nodes to study the density effects and nodes are randomly placed

TABLE I
COMPLEXITY COMPARISON
(B: BEST CASE, W: WORST CASE)

	LMR	Disjoint Multipath	Braided Multipath
Overhead(unicast)	$D \cdot O(\|N\|)$	B: $O(d)$ W: $2D \cdot O(\|N\|)$	B: $O(d)$ W: $2D \cdot O(\|N\|)$
Overhead(mcast)	$O(\|N\|)$	B: $O(d)$ W: $2D \cdot O(\|N\|)$	B: $O(d)$ W: $2D \cdot O(\|N\|)$
Setup delay	B: $2t_p \cdot O(d)$ W: $(2t_p + t_d) \cdot O(d)$	B: $t_p \cdot O(d)$ W: $2t_p \cdot O(\|N\|)$	B: $t_p \cdot O(d)$ W: $2t_p \cdot O(\|N\|)$

within a $2500m \times 2500m$ area. Besides these nodes, we put two nodes working as source and sink at the location (500, 1250) and (2000, 1250). Theoretically, at least 6 hops are needed for them to communicate. In a random topology generated by the above method, around 11 hops on average are used. For a given density, more than 30 topologies are used to get a 95% confidence interval.

Fig. 5 shows that, in most simulations, LMR can successfully find a backup bath, especially when the density is higher. In some cases, LMR cannot find a disjoint path and segmented paths are created. The ratio of extra backup path length is also shown in Fig. 5. Similar to the length of a single disjoint backup path, the length of a segmented backup path is the total hops on all the segments. This ratio is calculated as follows,

$$(L_b - L_w)/L_b \tag{3}$$

From the figure, we can see that, at lower densities, the backup paths are relatively longer since fewer alternate paths exist in the topologies and LMR has to pick up a longer one.

Fig. 6 shows the overhead of LMR in term of packets. Both local unicast and local multicast are simulated and the results match the analysis in the last subsection closely. The average node degree can be estimated by the following equation,

$$D = \pi(250)^2/(2500)^2 \times |N| - 1 \tag{4}$$

which is approximately the average number of nodes within the transmission range of a node. For example, with a 400 node network, the average degree D is about 11.6, which is close to the simulation result, i.e. 4430/420=10.6.

VI. Conclusions

In this paper, we present a review of current research on multipath routing in ad hoc networks and sensor networks. While a rich body of literature exists for ad hoc networks, few methods are appropriate for sensor networks due to the lack of global IDs. We proposed a novel approach called Label-based Multipath Routing (LMR), which employs localized information only. Analytical and simulation results show that LMR can find disjoint or segmented backup paths more efficiently compared to the Disjoint and Braided Multipath methods [16]. The label information in LMR can be used for segmented backup path search if a disjoint path is not found, reducing overhead and delay. Furthermore, LMR can take advantage of local multicast, significantly reducing the routing overhead.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, Aug. 2002.

[2] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *IEEE Hawaii International Conference on System Sciences*, 2000.

[3] A. Manjeshwar and D. P. Agrawal, "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks," in *IEEE International Parallel Distributed Processing Symposium*, 2001.

[4] A. Manjeshwar and D. P. Agrawal, "APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," in *IEEE International Parallel Distributed Processing Symposium*, 2002.

[5] S. Hedetniemi, S. Hedetniemi, and A. Liestman, "A survey of gossiping and broadcasting in communication networks," *Networks*, vol. 18, 1988.

[6] C. M. Okino and M. G. Corr, "Best effort adaptive routing in statistically accurate sensor networks neural networks," in *Proceeding Of IJCNN*, 2002.

[7] F. Ye, A. Chen, S. Lu, and L. Zhang, "A scalable solution to minimum cost forwarding in large sensor networks," in *Proceedings of the International Conference on Computer Communications and Networks*, 2001.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceeding Of ACM MOBICOM*, 2000.

[9] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination protocol for wireless sensor networks," in *Proceeding Of ACM MOBICOM*, 1999.

[10] S. Chakrabarti and A. Mishra, "Qos issues in ad hoc wireless networks," in *IEEE Communications Magazine*, Feb. 2002.

[11] D. Johnson, "Routing in ad hoc networks of mobile hosts," in *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 1995.

[12] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," 1996.

[13] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

[14] T. Goff, N. B. Abu-ghazaleh, d. S. Phatak, and R. Kahvecioglu, "Preemptive routing in ad hoc networks," in *Proceeding Of ACM SIGMOBILE*, July 2001.

[15] M. Spohn and J. Garcia-Luna-Aceves, "Neighborhood aware source routing," in *Proceeding Of ACM MOBIHOC*, 2001.

[16] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," in *ACM Mobile Computing and Communications Review*, vol. 5, no. 4, 2001.

[17] J. W. Suurballe, "Disjoint paths in a network, networks," no. 4, pp. 125–145, 1974.

[18] W. D. Grover, *Distributed Restoration of the Transport Network, Telecommunications Networks Magament in the 21st Century, Techniques, Stanstards, Technologies and Applications*. IEEE Press, 1994.

[19] K. Fall and K. Varadhan, *The ns Manual*. http://www-mash.cs.berkeley.edu/ns/, 2002.