Energy Aware Multi-path and Multi-SPEED Routing Protocol in Wireless Sensor Networks

Sussan Sanati[*], Mohammad Hossein Yaghmaee[**], Asghar Beheshti[***]
[*] Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran.
s_sanati@yahoo.com
[**] Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran.
hyaghmae@ferdowsi.um.ac.ir
[***] Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran.
abeheshti@iust.ac.ir

Abstract—Wireless sensor networks are limited in energy. Any routing protocol used in wireless sensor networks should take into consideration the time sensitive nature of the traffic in such networks, along with the amount of energy left for each sensor.

In this paper we present an energy aware packet delivery mechanism for probabilistic Quality of Service (QoS) guarantee in wireless sensor networks. Each node takes routing decisions based on geographic progress towards the destination sink, required end-to-end total reaching probability, delay at the candidate forwarding node and residual energy. The simulation results demonstrate that the proposed protocol effectively improves the energy usage efficiency of the sensor nodes, maximizing the lifetime of the entire sensor network, while keeping guaranteed QoS.

Keywords—Wireleess Sensor Network; Quality of Service; Routing Protocol; Wireless Node;

I. INTRODUCTION

Wireless sensor networks (WSN) are being used more and more for critical missions in places where other types of information collection techniques are either impossible, such as space research, or inefficient, such as very large jungles. A large volume of sensor nodes are distributed in the environment to collect data, perform local processing, and communicate the results to a *base station (BS)* or *sink*.

The sensor nodes are limited in wireless communication abilities, requiring them to use a routing protocol to pass the data across the network and to the base station. This routing process takes time, since the nodes have to share the wireless media. Also, some nodes may have to queue incoming data and forward them one by one. These delays cause different data packets to take various times to reach the base station, even if they originated from a single source.

Some missions, such as target tracking in battlefields, habitat monitoring in forests, and space research on the moon and Mars, have real-time requirements for the data, meaning the sensor data is valid only for a limited time duration, and hence, needs to be delivered within a time bound called *deadline*. In these applications, timely delivery of sensory data plays a crucial role in the success of the mission. Furthermore, different types of sensory data have different deadlines depending on the dynamics of the sensed environment. Therefore, sensor network applications require delivery of various types of sensory data with different levels of real-time requirements.

Apart from real-time requirements, sensory data have reliability requirements as well. For example, in a forest monitoring application, the temperature information of the forest is valid for a specific time, which is its real-time requirement. However, temperature information that is in the range of normal temperature can suffer a higher percentage of loss in the network while being delivered to the base station, whereas sensor data containing an abnormally high temperature should be delivered to the base station with a very high probability, since it can be a sign of fire, and is less tolerant to loss.

In addition, apart from some sensory data which are created aperiodically by detection of critical events at unpredictable points in time, there are other types of sensory data for periodic monitoring of environmental status. Therefore sensor network applications have a mixture of periodic and aperiodic traffic types.

Provisioning acceptable QoS for all possible types of traffic with the above characteristics is a challenging problem due to topological aspects of sensor networks that include: Large scale with thousands of densely placed nodes and dynamic topology changes due to node mobility, failure and addition.

Recent QoS studies in sensor networks [1], [2], [3] focus on only one QoS domain, either timeliness or reliability. They are also limited in differentiating services for traffics with different levels of timeliness and reliability requirements. Another study [4] proposes a packet delivery mechanism for QoS provisioning called *Multi-Path and Multi-Speed Routing Protocol (MMSPEED)* that spans over network layer and medium access control (MAC) layer. MMSPEED addresses both real-time and reliability requirements in a decentralized fashion, offering multiple levels of delivery speed and redundancy.

The protocol relies on local routing decisions where the individual node's knowledge is restricted to its direct neighbors' location and link quality. It uses the SPEED protocol [5] to guarantee a network-wide speed. In the time domain, there is the choice between several

Proceedings of the 14th International CSI Computer Conference (CSICC'09)
978-1-4244-4262-1/09/$25.00 ©2009 IEEE

forwarding nodes that guarantee a high probability of reaching the sink in a certain time span. Each node maintains delay estimations for the transmission to the sink via its neighbors and compares the resulting speed to the chosen speed level and forwards the packet to the most promising neighbor in terms of delivery speed and closeness to the end destination.

MMSPEED handles several reliability levels in a similar fashion to SPEED, except that is uses many speed levels. Each node has one queue per speed level, where the queue with the highest speed requirements always gets priority of the node's resources over lower speed levels. Only when higher levels do not need the resources is a lower level allowed to use them.

At the source a packet is assigned a certain reliability level which is connected to the desired probability that the packet reaches the sink. As the packet traverses the network, based on different reliabilities of neighbor nodes, the packet may be duplicated and forwarded along multiple paths to ensure that initial reachability requirements are satisfied. The more redundant forwarding paths used on the way, the lower the probability of losing the packet to contention or transmission error. If at some point a node realizes the packet has progressed through the network slower than expected, it compensates for it by increasing the packets speed level.

Unfortunately MMSPEED does not care about an individual node's energy situation and does not take a node's energy reserves into account when choosing a forwarding path, therefore many packets are routed over the same route, and eventually a small set of nodes are overburdened in terms of energy and network load. On the other hand MMSPEED could use its redundant path selection for load balancing and not only for reliability enhancement, and therefore improve the overall network lifetime.

Considering MMSPEED protocol and its characteristics, we decided to add the aforementioned features to the MMSPEED protocol and try to balance the load and energy consumption of individual nodes in the network and improve the overall network lifetime. Therefore, each node makes routing decisions based on the following four parameters: geographic progress towards the destination sink, required end-to-end total reaching probability, delay and residual energy at the candidate forwarding node.

Our simulation results show that the new protocol, EAMMSPEED, provides a more stable service in the sensor network and maximizes the lifetime of the entire network while maintaining the QoS guarantees provided by MMSPEED.

The rest of this paper is organized as follows: Section II presents the MMSpeed routing protocol, Section III describes our add-on features to support the Energy Aware Multi-path and Multi-SPEED Routing Protocol (EAMMSPEED), Section IV evaluates the proposed protocol via simulation and Section V concludes the paper.

II. **MMSPEED**: **M**ULTI-PATH AND **M**ULTI-**SPEED R**OUTING **P**ROTOCOL

The MMSPEED routing protocol is designed with two important goals:

- localized packet routing decision without global network state update or a priori path setup, and
- providing differentiated QoS options in isolated timeliness and reliability domains.

For the localized packet routing without end-to-end path setup and maintenance, geographic routing mechanism based on location awareness is used.

Each sensor node is assumed to be aware of its geographical location. This location information can be exchanged with immediate neighbors with "periodic location update packets". Thus, each node is aware of its immediate neighbors within its radio range and their locations. Using the neighbor locations, each node can locally make a per packet routing decision such that packets progress geographically towards their final destinations. If each node relays the packet to a neighbor closer to the destination area, the packet can eventually be delivered to the destination without global topology information. For on-time delivery of packets with different end-to-end deadlines, MMSPEED provides multiple delivery speed options that are guaranteed network-widely. For this, the idea of SPEED protocol[5] which can guarantee a single network-wide speed was used.

If every node i in the entire network can relay a packet to a neighbor node j whose progress speed toward destination k, is higher than the pre-specified speed lower bound *SetSpeed*, then the *SetSpeed* can be uniformly guaranteed all over the network [5]. In SPEED protocol, each node i maintains delay estimation to each neighbor j, calculates its progress speed $Speed^{k}_{i,j}$,

$$Speed^{k}_{i,j} = \left(dist_{i,k} - dist_{j,k}\right) / \, delay_{i,j} \qquad (1)$$

and forwards a packet to a neighbor j whose progress speed is higher than *SetSpeed*.

However, nodes in a congested area may not be able to find any node with progress speed higher than *SetSpeed*. Those nodes start reducing workload by probabilistically dropping packets in order to retain at least one forwarding node whose progress speed is higher than *SetSpeed*. This approach compromises reliability for assuring network-wide uniform speed *SetSpeed* with a high probability [4]. Along with packet dropping, nodes also issue so-called "back-pressure packets" to reduce the incoming packet traffic from other neighboring nodes [5].

By replicating the single network-wide speed guarantee mechanism, MMSPEED provides multiple layers of network wide speed guarantees. Figure 1 depicts the protocol structure of a sensor node for multiple speed levels. Each speed layer l independently runs the above mechanism to guarantee the corresponding .

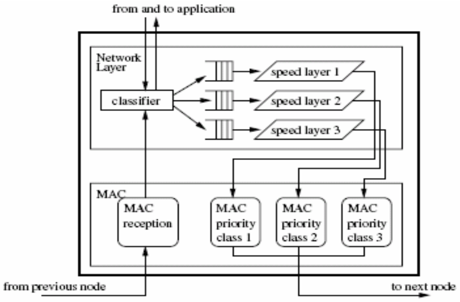

Figure 1: Protocol Structure of a Sensor Node [4]

Dynamic compensation is needed to adjust the local decisions to meet the end-to-end deadline. Specifically, the classifier of the source node *s* selects the most proper speed for a packet *x* based on the distance to final destination *d*, i.e., $dist_{s,d}(x)$,and end-to-end deadline *deadline(x)*. The minimum required speed level , $Speed^{req}(x)$ to meet the end-to-end deadline is calculated as :

$$Speed^{req}(x) = dist_{s,d}(x) \big/ deadline(x) \qquad (2)$$

Thus, the classifier of the source node picks the most proper speed layer *l* .Then, the corresponding speed layer module chooses a neighbor node *i* whose progress speed estimation, $Speed^{d}_{s,i}$,is higher than $SetSpeed_l$. However, after the packet travels several hops towards the destination *d*, an intermediate node *f* may notice that the packet has traveled slowly so far due to longer delays than the original estimation [4].

Then, the node *f* compensates the previous extra delay by boosting the speed level. For this, each intermediate node *f* adjusts speed level, based on the remaining distance to destination $dist_{f,d}$ and the remaining time to deadline.

By implementing this speed level compensation in the classifier in Figure 1, inaccuracies of localized decisions can be compensated globally as the packet travels. This ensures high probability of meeting end-to-end deadlines. However, not all packets are guaranteed to reach their destinations [4]. First, for guaranteeing network wide speed options, the routing layer of intermediate nodes can probabilistically drop packets if average delay becomes larger than a threshold. Secondly, the MAC layer can also drop the packet if it cannot be delivered with a limited number of tries. The high error rates of physical wireless channel also increase the probability of packet losses. To assure a certain level of reachability, another mechanism in the reliability domain is used.

In a dense sensor network, there exist multiple redundant paths to the final destination, even though they may not be the shortest paths [4]. A non-shortest path is acceptable as long as it can deliver a packet within end-to-end deadline. Utilizing possibly longer alternative paths is sometimes preferable for load balancing and avoiding hot spots on the shortest paths. MMSPEED protocol exploits such inherent redundancies to probabilistically guarantee the required end-to-end reliability level (end-to-end reaching probability) of a packet. The more paths we use to deliver a packet, the higher is the probability that the packet reaches its final destination, despite packet drops, node failures, and errors on wireless links. Thus, by controlling the number of forwarding paths depending on the required reliability level, service differentiation in the reliability domain can be provided [4].

The challenging task is to devise local decision mechanisms to compute and identify forwarding paths to meet the packets' end-to-end reachability requirement. To address this problem, a combination of multipath forwarding based on local estimation and dynamic compensation is used. Each node locally determines multiple forwarding nodes to meet the required reaching probability based on local error estimations and geographic hop distances to immediate neighbors. More specifically, each node *i* can maintain the recent average of packet loss rate, $e_{i,j}$, to each immediate neighbor node *j*. The packet loss includes both intentional packet drops for congestion control and errors on the wireless channel. The estimation of packet loss rate is also supported by MAC layer loss estimation. Using $e_{i,j}$, node *i* can locally estimate the end-to-end reachability of a packet from node *i* to the final destination *d* via a neighbor node *j* as follows:

$$RP^{d}_{i,j} = (1 - e_{i,j})(1 - e_{i,j})^{\left| dist_{j,d} / dist_{i,j} \right|} \qquad (3)$$

where $\left| dist_{j,d} / dist_{i,j} \right|$ is hop count estimation from node *j* to the final destination *d*.

Note that this local estimation equation is based on two assumptions: 1) packet loss rate in each of the following hops will be similar to the local loss rate of the current hop and 2) for each following hop, the geographic progress to the destination will be similar to the current progress.

From the end-to-end reachability estimation via a single neighbor node, the number of forwarding nodes is determined to satisfy the end-to-end reachability requirement, p^{req}, of a packet. More specifically, the total reaching probability *TRP* is initially set to zero. Whenever we add one forwarding node *j*, the *TRP* is updated as follows:

$$TRP = 1 - (1 - TRP)(1 - RP^{d}_{i,j}) \qquad (4)$$

Forwarding nodes are added until *TRP* becomes larger than p^{req} [4]. Once the set of required forwarding nodes are determined, the packet is delivered to them using the MAC multicast service.

However, the local decision on multiple forwarding node selection may turn out to be incorrect in the following nodes because local estimations are used to model the remaining part of the network about which the local node does not have any information. To address this problem, dynamic compensation in the reliability domain is used.

By combining aforementioned timeliness and reliability guarantee mechanisms, MMSPEED protocol can serve various packets with different timeliness and reliability requirements. Once a sensor node detects an event, it creates a packet x to be reported to the sink node. Based on the content of the sensor data, the source node selects the appropriate end-to-end deadline, *deadline(x)* and required reaching probability, p^{req}. The packet with end-to-end deadline and required reaching probability is forwarded towards its destination by MMSPEED. MMSPEED first classifies the packet into the proper speed layer based on the end-to-end deadline and the geographic distance to the destination. Then, the corresponding speed layer module *l* finds multiple forwarding nodes among those with progress speed higher than $SetSpeed_l$, such that the total reaching probability is higher than or equal to the required reaching probability. Then, the packet is delivered to the chosen forwarding nodes.

III. ENERGY AWARE MULTI PATH AND MULTI-SPEED ROUTING PROTOCOL

In typical WSN scenarios space and cost constraints make the use of large batteries impossible. Nevertheless, it is often not feasible to change batteries on a regular basis. It is therefore vital that sensor nodes save as much energy as possible and prolong the lifetime of an individual node as well as the network lifetime.

For efficient use of network resources, we decided to include energy efficiency in the MMSPEED routing protocol by taking all node's energy levels into account when making routing decisions. We compare energy reserves of neighbor nodes when choosing a forwarding path to find energy efficient paths along which the end-to-end delay and reliability requirements can be met.

We use the geographic routing mechanism for the localized packet routing based on location awareness of each node. Using beacon messages, each node i learns (x,y) position, residual energy $E_{residua}$ and node *receive-delay* of all neighbor nodes within its radio range.

For timeliness delivery, multiple network-wide packet delivery speed options are provided for different traffic types according to their end-to-end deadlines. In supporting service reliability, probabilistic multi-path forwarding is used to control the number of delivery paths based on the required end-to-end reaching probability.

Each node having the packet, calculates the progress speed of neighbor nodes. The nodes having a progress speed higher than (or equal to) the speed of the chosen speed layer are selected as the candidate forwarding nodes. The list of candidate forwarding nodes is then sorted in descending order of each candidate's energy, meaning receivers with higher energy levels are chosen first. Then, the packet is delivered to the selected forwarding nodes (Figure 2). In our proposed routing protocol, nodes posses this kind of information about each node's energy via piggybacked energy levels inserted in acknowledgement messages.

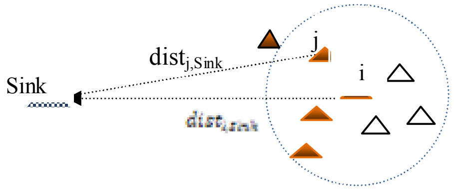

Figure 2: Selecting forwarding nodes

These methods are implemented in a localized way with dynamic compensation for the inaccuracies of local decisions. Simulations show that in dense networks taking energy reserves into account for choosing candidate forwarding nodes balances the load and energy consumption of different nodes and prolongs network lifetime.

IV. EXPERIMENTAL RESULTS

In order to evaluate the effects and efficiency of our proposed protocol, we implemented EAMMSPEED in Network Simulator (NS2) [6]. We simulate and compare EAMMSPEED with MMSPEED, which to our knowledge, provides one of the best QoS guarantees in sensor networks. Hence, any comparisons with other routing protocols used in sensor networks, such as AODV [7] and DSR [8], are pointless since they provide no QoS guarantees. General settings of the simulation can be found in Table 1.

We consider three different node settings for our simulations. The first setting, hereafter referred to as *low density* network, has 50 nodes. In this setting each node may not always have many different paths to forward packets to. The second setting with 100 nodes, *dense* network, provides nodes with the option to choose different forwarding paths to the destination. The third setting has 150 nodes, i.e. *high density* network, to depict how the protocols behave in situations where numerous paths to the destination exist that can guarantee on time packet delivery.

Table 1: Simulation Settings

BandWidth	12.8Kbps
Mac layer	802.11
Number of Nodes	50,100,150
Node placement	random
Radio range	140m
Terrain	670m×670m

Each simulation runs for 200 seconds with 3 unique event sources generating events with 0.65 requested reachability probability and deadlines of 1.5 seconds.

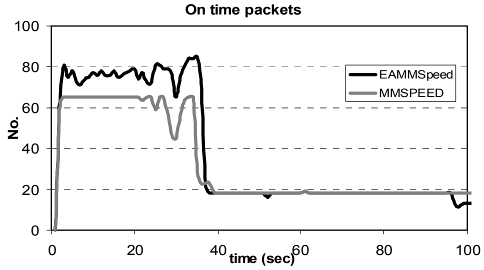

Figure 3: Number of On-time Packets in Low Density Network

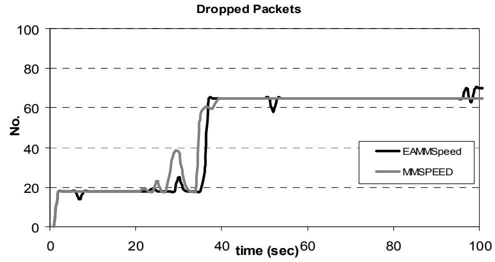

Figure 4: Number of Dropped Packets in Low Density Network

Figures 3 and 4 show how the two protocols perform in the low density network. Notice how the number of on-time and dropped packets is almost the same for both protocols. This is due to the fact that each node has at most very few forwarding candidates, typically resulting in one possible forwarding path for each packet. So being, taking energy levels into consideration has no or very little effect on the final routing decisions.

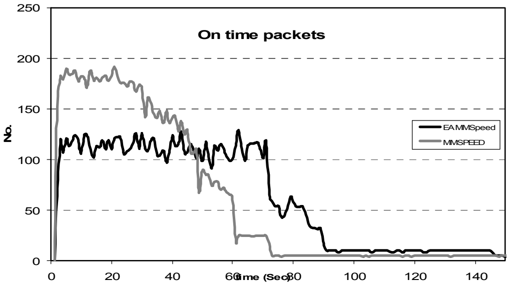

Figure 5: Number of On-time Packets in Dense Network

The number of on time and dropped packets in the dense network can be found in Figures 5 and 6 respectively. It can be seen how the two protocols start off with almost the same throughput. However, MMSPEED quickly starts exhausting the most used paths, forcing nodes along those paths to run out of energy. This in turn causes remaining nodes to find new, and longer, paths to the destination. This steadily increases the number of packets that miss their deadline, causing a rising rate in the number of dropped packets (Figure 6). At some point close to the end of the simulation, the remaining paths are too long for any packet to meet its deadline, ending in a 100% packet loss ratio.

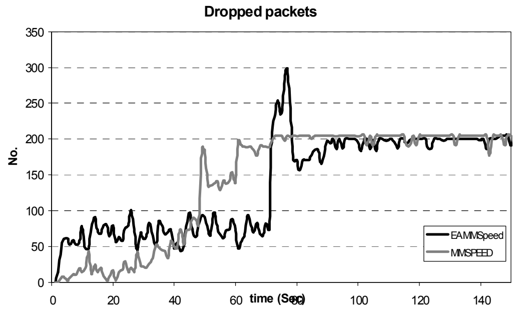

Figure 6: Number of Dropped Packets in Dense Network

Figure 7 shows how EAMMSPEED maximizes the lifetime of the network for the high density network. Results are similar for the other settings as well. A node is considered dead when it runs out of energy and loses the ability to send or receive packets. Nodes using MMSPEED start to die quickly one by one, and as a result, even though there may be many active nodes still available inside the network, the entire network loses connectivity and becomes useless. It can be seen in Figure 8, the number of on time packets in the high density network, show no more packets can be delivered in the network a bit after the 110th second, at a time in which less than 30% of the nodes have ran out of energy. In contrast, although all nodes using the EAMMSPEED protocol for delievring the packets die more or less simultaneously, the network is kept active for the most possible time. Only after 50% of the nodes die after 140 seconds does the network loose connectivity. This provides a significant increase in the life of the entire network.

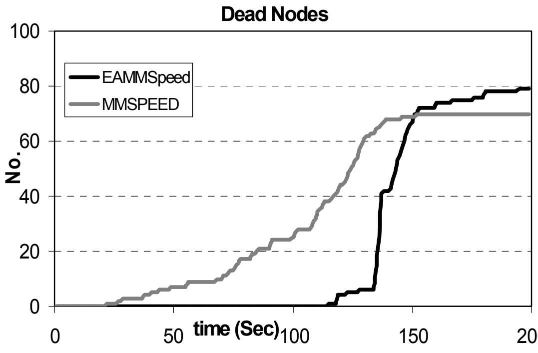

Figure 7: Number of Dead Nodes in High Density Network

Another point to notice in Figures 8 and 9 is how MMSPEED starts with a slightly higher number of on time packets. The simulations were tuned so that the average network traffic is greater than its capacity, so we would encounter packet drops in general. At the beginning of the simulation MMSPEED uses shortest possible paths, thus delivering more packets on time. On the contrary, even though EAMMSPEED has these short paths available, it also uses longer paths for delivering packets to balance energy consumption leading to a few more packet drops in the network. However, this higher rate of delivery with MMSPEED drops quickly as the short paths start to disappear therefore, the number of on-time packets drop below that of EAMMSPEED.

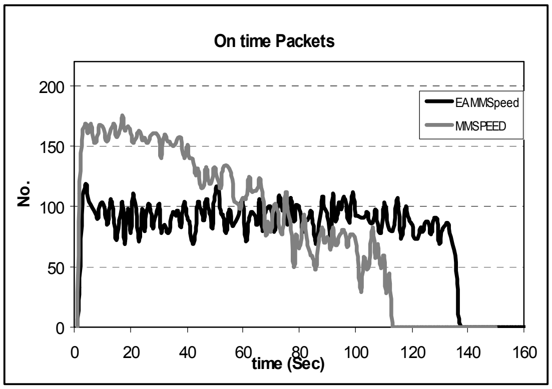

Figure 8: Number of On-time Packets in High Density Network

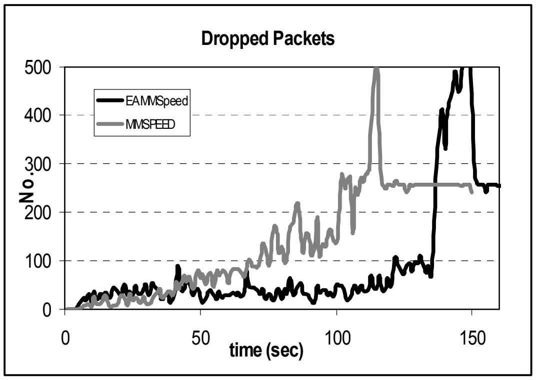

Figure 9: Number of Dropped Packets in High Density Network

In general, it is worth noting that better energy consumption efficiency comes at the price of a small increase in packet drops in the network. Nonetheless, a more stable network is achieved both in terms of QoS and lifetime.

V. CONCLUSION

Sensor networks are limited in resources. The time sensitive information collected by the nodes requires the underlying routing protocol to provide QoS guarantees. However, the limited energy of the sensors has a great impact on the usability of the whole network. Thus the routing protocol must also include energy in routing decisions.

In this paper we proposed a new routing protocol, EAMMSPEED, for wireless sensor networks that has the aforementioned characteristics. It combines QoS guarantees of MMSPEED with exchanged energy information by nodes to provide better energy consumption of the nodes.

Through simulation, our results show how EAMMSPEED provides a more stable service for wireless sensor networks and maximizes the lifetime of the sensors and the overall sensor network.

REFERENCES

[1] T. Chen, J. Tsai, and M. Gerla, “QoS Routing Performance in Multihop Multimedia Wireless Networks,” *Proc. IEEE Sixth Int. Conf. Universal Personal Communications*, vol. 2, 1997, pp.557–561.

[2] R. Sivakumar, P. Sinha, and V. Bharghavan, “CEDAR: Core Extraction Distributed Ad Hoc Routing Algorithm,” *IEEE Journal on SelectedAreas of Communications*, vol. 17, no. 8, 1999, pp. 1454–1465.

[3] S. Chen and K. Nahrstedt, “Distributed Quality-of-Service Routing in Ad Hoc Networks,” *IEEE Journal Selected Areas of Communications*, vol. 17, no. 8, 1999, pp. 1488–1505.

[4] E. Felemban, ; C. G. Lee; E. Ekici,; R. Boder ; S. Vural,, “MMSPEED: Multipath Multi-Speed Protocol for QoS of Reliability and Timeliness in Wireless Sensor Networks”, *IEEE Trans. Mobile Computing*, vol. 5, no. 6, 2006.

[5] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, “SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks,” *Proc. IEEE Int. Conf. on Distributed Computing Systems*, 2003, pp. 46–55.

[6] Marc Greis' Tutorial for the UCB/LBNL/VINT Network Simulator "ns" http://www.isi.edu/nsnam/ns/tutorial/

[7] C. E. Perkins, E. M. Royer, "Ad-hoc On Demand Distance Vector Routing", *In WMCSA’99*, New Orleans, LA, February 1999.

[8] D. B. Johnson, and D. A. Maltz. "Dynamic Source Routing in Ad Hoc Wireless Networks", *In Mobile Computing, edited by Tomasz Imielinski and Hank Korth*, Chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.