Probabilistic QoS Guarantee in Reliability and Timeliness Domains in Wireless Sensor Networks

Emad Felemban, Chang-Gun Lee, Eylem Ekici, Ryan Boder, and Serdar Vural
Electrical and Computer Engineering
Ohio State University, Columbus, Ohio 43210
E-mails: {felembe,cglee,ekici,boder,vurals}@ece.osu.edu.

Abstract— **In this paper, we present a novel packet delivery mechanism called *Multi-path and Multi-Speed Routing Protocol (MMSPEED)* for probabilistic QoS guarantee in wireless sensor networks. The QoS provisioning is performed in two quality domains, namely, timeliness and reliability. Multiple QoS levels are provided in the timeliness domain by guaranteeing multiple packet delivery speed options. In the reliability domain, various reliability requirements are supported by probabilistic multipath forwarding. All these for QoS provisioning are realized in a localized way without global network information by employing localized geographic packet forwarding augmented with *dynamic compensation*, which compensates the local decision inaccuracy as a packet travels towards its destination. This way, MMSPEED can guarantee end-to-end requirements in a localized way, which is desirable for scalability and adaptability to large scale dynamic sensor networks. Simulation results show that MMSPEED provides QoS differentiation in both reliability and timeliness domains and, as a result, significantly improves the effective capacity of a sensor network in terms of number of flows that meet both reliability and timeliness requirements.**

I. Introduction

Wireless sensor networks can be used for many mission-critical applications such as target tracking in battlefields, habitat monitoring in forests, and space research on Moon and Mars. In these applications, reliable and timely delivery of sensory data plays a crucial role in the success of the mission. Specifically, the above-mentioned sensor network applications share the following characteristics:

- Diverse *Real-Time* Requirements: The sensory data reflects the physical status of the sensing environment. Thus, the sensor data is valid only for a limited time duration, and hence needs to be delivered within such time bound called deadline for real-time applications. More importantly, different sensory data has a different deadline depending on the dynamics of the sensed environment. For example, location sensory data for a fast moving target has shorter deadline than that for a slow moving target. In short, sensor network applications require delivery of various types of sensory data with different levels of real-time requirements.
- Diverse *Reliability* Requirements: Depending on its contents, sensory data has a different reliability requirement. For example, in the forest monitoring applications, the temperature information that is in the range of normal temperature can be delivered to the control center tolerating a certain percentage of loss. On the other hand, the sensor data that contains an abnormally high temperature should be delivered to the control center with a very high probability since it can be a sign of fire. In short, sensor network applications have various types of sensory data with different levels of reliability or so-called *reachability* requirements.
- Mixture of periodic and aperiodic data: Some sensory data are created aperiodically by detection of critical events at unpredictable points in time. In addition, there are other types of sensory data for periodic monitoring of environmental status. In short, sensor network applications have a mixture of periodic and aperiodic traffic types.

Provisioning acceptable QoS for the traffics with the above characteristics is a challenging problem due to topological aspects of sensor networks that include

- Dynamic topology changes due to node mobility, failure, and addition,
- Large scale with thousands of densely placed nodes, and
- Unreliable nature of wireless links[1].

Most of current QoS provisioning protocols [2], [3], [4], [5] in wireless ad hoc networks are based on the end-to-end path discovery, resource reservation along the discovered path, and path recovery in case of topology changes. However, such approaches are not suitable for sensor network applications with above characteristics for many reasons. Firstly, the path discovery latency is not acceptable for urgent aperiodic packets. Also, for the unpredictable aperiodic packets, it is not practical to reserve resources along the end-to-end path. Even for periodic continuous traffics, the end-to-end path based mechanisms are problematic in dynamic sensor networks since service disruption during the path recovery is not acceptable in mission critical applications. Furthermore, the approaches are not scalable due to huge overhead of path discovery and recovery in large scale sensor networks.

Recent QoS studies in sensor networks [6], [7], [8] 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 [9] can guarantee the different real-time requirements by realizing EDF packet scheduling in a decentralized way.

[1] A recent study [1] reports that 20% of neighbor nodes in a radio communication range suffer more than 10% of packet loss.

0-7803-8968-9/05/$20.00 (c)2005 IEEE

However, it is based on the assumption that most traffic is periodic and all periods are known a priori, which is not the case for many sensor network applications. Also, it is not adaptive to dynamics of sensor networks.

In this paper, we propose a novel 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. Our major goal is to provide QoS differentiation in two isolated quality domains, namely, *timeliness* and *reliability*, so that packets can choose the most proper combination of service options depending on their timeliness and reliability requirements. For the service differentiation in the timeliness domain, the proposed mechanism provides *multiple network-wide speed options* extending the idea of single network-wide speed guarantee in [7]. For the service differentiation in the reliability domain, we exploit the inherent redundancy of dense sensor networks by realizing *probabilistic multipath forwarding* depending on packet's reliability requirement.

Another important goal is to provide end-to-end QoS provisioning with local decisions at each intermediate node without end-to-end path discovery and maintenance. This goal is important to preserve properties of scalability to large sensor networks, self-adaptability to network dynamics, and appropriateness to both aperiodic and periodic traffic flows. For this, MMSPEED realizes the above QoS differentiation based on localized geographic forwarding using only immediate neighbor information. One challenge is to ensure that localized forwarding decisions result in end-to-end QoS provisioning in global sense. To handle this problem, we propose the notion of *dynamic compensation*, which compensates for inaccuracy of local decisions in a global sense as a packet progresses towards its destination. As a result, packets can meet their end-to-end requirements with a high probability even if packet delivery decision is made locally.

The rest of this paper is organized as follows: Section II presents the proposed routing protocol. Section III describes our add-on features of MAC protocol to support the routing protocol. Section IV discusses the performance evaluation of the proposed protocols. Section V summarizes the related work. Finally, Section VI concludes the paper.

II. MMSPEED: Multi-path and Multi-SPEED Routing Protocol

The proposed 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, we use a geographic routing mechanism based on location awareness. In sensor networks, information carried in a packet is more tightly associated with the geographic area where the corresponding event is sensed than with any specific sensor node. For example, after detecting a new target, to keep track of the target location, the center can establish a session with any sensor node in the surrounding area and not necessarily with the one which first detected the target. Each sensor node is assumed to be aware of its geographical location using GPS [10] or distributed location services [11], [12]. 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. Many recent protocols [13], [14], [10], [7] are also employing such geographic routing mechanisms. The localized geographic routing has the following three advantages in sensor networks:

- Scalability to a very large and dense sensor networks,
- No path setup and recovery latency—good for both critical aperiodic and periodic packets, and
- Packet-by-packet path discovery resulting in self adaptation to network dynamics.

Our goal is to provide guaranteed packet delivery services in both timeliness and reliability domains while preserving the benefits of localized geographic routing. Section II-A presents a geographic routing method that can provide multiple packet delivery speeds for multiple QoS options in timeliness domain. Then, in Section II-B, a geographic routing method that employs probabilistic multi-path forwarding for multiple QoS options in reliability domain is presented. Finally, Section II-C explains how the above two can work together to meet the combined requirement, i.e., *on-time reachability*—the percentage that a packet reaches its final destination within deadline.

A. Differentiated QoS Options in the Timeliness Domain

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, we borrow the idea of SPEED protocol [7] which can guarantee a single network-wide speed.

Consider two immediate neighbor nodes i and j in Figure 1. The geographical distances from node i and node j to the final destination k are $dist_{i,k} = 100m$ and $dist_{j,k} = 80m$, respectively. Suppose that node i forwards a packet to node j with delay (including queueing, processing, and MAC collision resolution) of $delay_{i,j} = 0.1sec$. This forwarding makes $dist_{i,k} - dist_{j,k} = 20m$ geographic progress toward the final destination k along the virtual direct line from node i to destination k. Thus, the progress speed $Speed_{i,j}^k$ from node i to node j toward the final destination k is $(dist_{i,k} - dist_{j,k})/delay_{i,j} = 20m/0.1sec = 200m/sec$. If every node i in the entire network can relay a packet to a neighbor node j whose progress speed toward destination k, i.e., $Speed_{i,j}^k = (dist_{i,k} - dist_{j,k})/delay_{i,j}$, is higher

0-7803-8968-9/05/$20.00 (c)2005 IEEE

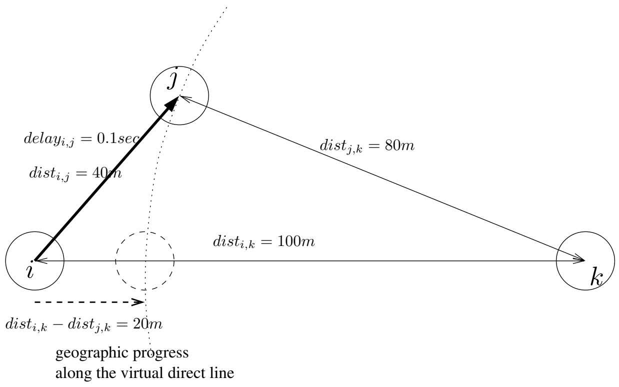


Fig. 1. Progress Speed from Node i to Node j toward Destination k.

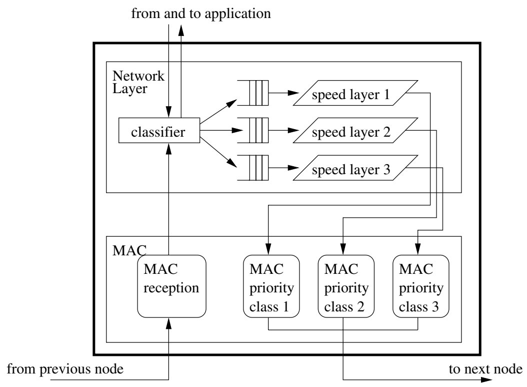


Fig. 2. Protocol Structure of a Sensor Node.

than the pre-specified speed lower bound $SetSpeed$, then the $SetSpeed$ can be uniformly guaranteed all over the network. If such network-wide guarantee of $SetSpeed$ is possible, the end-to-end packet delivery delay from any source s to any destination d can be bounded by $dist_{s,d}/SetSpeed$. For this purpose, in SPEED protocol, each node i maintains delay estimation $delay_{i,j}$ to each neighbor j, calculates $Speed_{i,j}^{k} = (dist_{i,k} - dist_{j,k})/delay_{i,j}$, and forwards a packet to a neighbor j whose progress speed $Speed_{i,j}^{k}$ 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. Along with packet dropping, nodes also issue so-called "back-pressure packets" to reduce the incoming packet traffic from other neighboring nodes. This back-pressure mechanism can also solve the void area problem, where routes may not find any neighbors that are closer to the destination than themselves.

By replicating the single network-wide speed guarantee mechanism, our protocol provides multiple layers of network-wide speed guarantees. Figure 2 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 $SetSpeed_l$. For this virtual layering, our protocol employs two important notions:

- Virtual isolation among the speed layers,
- Dynamic compensation of local decisions.

The virtual isolation of the speed layers aims to minimize the effects of lower speed packets on the delays experienced by the higher speed packets. Virtual isolation is accomplished by classifying incoming packets according to their speed classes and placing them into appropriate queues. The packets in the highest speed queue is served in FCFS discipline, followed by the next highest speed queue, and so on. This packet processing strategy prevents a packet of higher speed layer from being delayed by lower speed packets. This is the prioritized scheduling of local packets in a single node that minimizes the intra-node priority inversion. Even if this local prioritization is possible, a high speed packet can be delayed by low speed packets in neighbor nodes, since the neighbor node has no information on pending packets in other neighbors. This will cause another delay effect across the speed layer. In order to minimize such delay effect, we need a special support from MAC layer which provides distributed prioritization that minimizes the inter-node priority inversion. This issue will be discussed in Section III.

The 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

$$Speed^{req}(x) = \frac{dist_{s,d}(x)}{deadline(x)}.$$

Thus, the classifier of the source node picks the most proper speed layer l such that

$$SetSpeed_l = min_{j=1}^{L}\{SetSpeed_j | SetSpeed_j \geq Speed^{req}(x)\},$$

where L is the number of speed options. Then, the corresponding speed layer module chooses a neighbor node i whose progress speed estimation $Speed_{s,i}^{d} = (dist_{s,d} - dist_{i,d})/delay_{s,i}$ 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. 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. However, determining the remaining time to deadline at each intermediate node is not trivial due to the lack of globally synchronized clock. To handle this problem, we measure the *elapsed time* at each node and piggyback the elapsed time to the packet so that the following node f' can determine the remaining time to deadline without globally synchronized clock. For this, when a packet x arrives at a node f, its MAC layer tags $t^{arrival}$ to the packet. This packet is processed by the network layer and forwarded to the chosen forwarding node f' via MAC layer. Note that the MAC layer of f spends some time to capture the channel using RTC/CTS handshake and may transmit the packet several times until receiving and

ACK from f'. For f to piggyback the accurate elapsed time, the MAC layer updates the field of elapsed time $t^{elapsed}$ every time just before it actually transmits the packet to the physical link as follows

$$t^{elapsed} = t^{arrival} - t^{departure}$$

where $t^{departure}$ is the time when node f transmits the packet to the physical link. Thus, once node f' successfully receives the packet, the packet contains the correct measurement of the elapsed time at node f. Now, node f' can update the remaining time to deadline as follows:

$$deadline(x) = deadline(x) - t^{elapsed} - t^{transDelay} - t^{propDelay}.$$

The propagation delay $t^{propaDelay}$ between two neighbor nodes is negligibly small, and the transmission delay $t^{transDelay}$ can be computed using the transmission rate and the length of the frame containing the packet. Based on this, the intermediate node f' can check whether the current speed level of the packet is sufficient to meet the end-to-end deadline. Specifically, the current speed $SetSpeed_l$ is sufficient if the following condition holds:

$$\frac{dist_{f',d}}{SetSpeed_l} \leq deadline(x)$$

where $\frac{dist_{f',d}}{SetSpeed_l}$ is new estimation of f' on the latency from f' to d with $SetSpeed_l$ and $deadline(x)$ is the remaining time to deadline at node f'. If the current speed is insufficient due to delays in the previous path segment, node f' can boost the speed level using the following formula:

$$Speed^{req}(x) = \frac{dist_{f',d}}{deadline(x)},$$

$$SetSpeed_l = \min_{j=1}^{L}\{SetSpeed_j | SetSpeed_l \geq Speed^{req}(x)\}.$$

By implementing this speed level compensation in the classifier in Figure 2, inaccuracies of localized decisions can be compensated globally as the packet travels. This ensures high probability of meeting end-to-end deadlines.

Thanks to the network-wide speed options together with dynamic compensation, we can claim that once a packet reaches its destination, it is likely that the packet meets its end-to-end deadline. However, not all packets are guaranteed to reach their destinations. 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, we propose another mechanism in the reliability domain as described in the next section.

B. QoS Differentiation in the Reliability Domain

In a dense sensor network, there exist multiple redundant paths to the final destination [15], [16], [6] even though they may not be the shortest paths. 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. Our 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 of packet drops, node failures, and errors on wireless links. Thus, by controlling the number of forwarding paths depending on the required reliability level, we can provide the service differentiation in the reliability domain.

The challenging task is to devise local decision mechanisms to compute and identify forwarding paths to meet packet's end-to-end reachability requirement. To address this problem, we combine 1) multipath forwarding based on local estimation and 2) dynamic compensation. 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 as described in Section III. 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:

$$RP_{i,j}^{d} = (1 - e_{i,j})(1 - e_{i,j})^{\lceil dist_{j,d}/dist_{i,j} \rceil}, \qquad (1)$$

where $\lceil dist_{j,d}/dist_{i,j} \rceil$ 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, we can determine the number of forwarding nodes to satisfy the end-to-end reachability requirement P^{req} of a packet. More specifically, we initially set the total reaching probability TRP to zero. Whenever we add one forwarding node j, the TRP is updated as follows:

$$TRP = 1 - (1 - TRP)(1 - RP_{i,j}^{d}), \qquad (2)$$

where $RP_{i,j}^{d}$ is calculated as in Equation (1). We add forwarding nodes until TRP becomes larger than P^{req}. Once we determine the set of required forwarding nodes, the packet is delivered to them using MAC multicast service described in Section III.

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, we use the notion of dynamic compensation in the reliability domain. The dynamic compensation can be explained with an example in Figure 3.

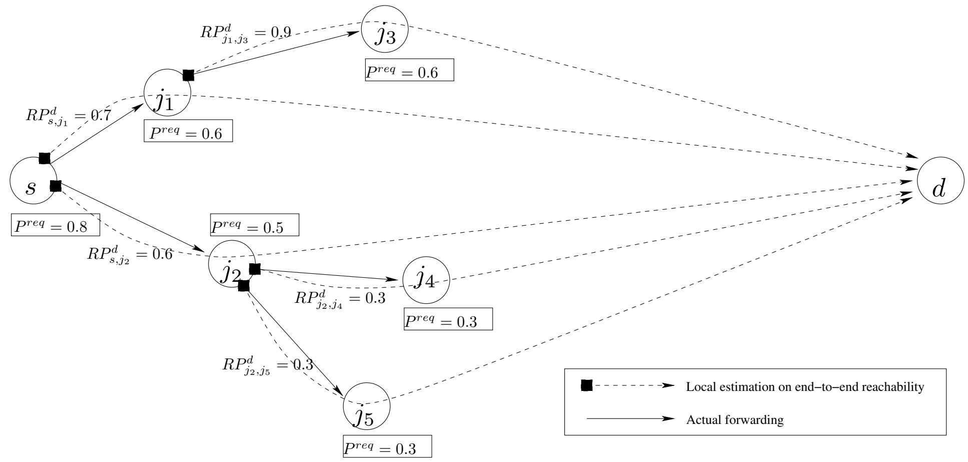


Fig. 3. Multipath Forwarding and Dynamic Compensation.

Consider a source sensor node s that detects an event that needs to be reported to the control center d with reachability $P^{req} = 80\%$. Suppose that the source node s determines to forward this packet to two immediate neighbors j_1 and j_2 based on its local estimation of $RP^d_{s,j_1} = 70\%$ and $RP^d_{s,j_2} = 60\%$. Remember that total reaching probability TRP via node j_1 and j_2 is given as

$$\begin{aligned} TRP &= 1 - (1 - RP^d_{s,j_1})(1 - RP^d_{s,j_2}) \\ &= 1 - (1 - 0.7)(1 - 0.6) = 0.88, \end{aligned}$$

which is higher than the reachability requirement $P^{req} = 80\%$. When transmitting the packet to node j_1 and j_2, new P^{req} values are assigned for each recipient. For example, recipients j_1 and j_2 may be assigned with $P^{req} = 0.6$ and $P^{req} = 0.5$, respectively, to just meet the condition that $TRP = 1 - (1 - 0.6)(1 - 0.5) = 0.8$.

When node j_1 and j_2 receive their copies with assigned $P^{req} = 0.6$ and $P^{req} = 0.5$, respectively, they make local forwarding decision to meet P^{req} as before but using their own estimations. For example, node j_1 can find a forwarding neighbor j_3 with $RP^d_{j_1,j_3} = 0.9$, and thus the assigned requirement $P^{req} = 0.6$ can be met with this single forwarding path. Thus, the packet is forwarded from j_1 to j_3 without change of $P^{req} = 0.6$. On the other hand, node j_2 finds that it needs two forwarding nodes j_4 and j_5 with $RP^d_{j_2,j_4} = 0.3$ and $RP^d_{j_2,j_5} = 0.3$ since its local loss rate estimation is worse than the original one made in the source node s. In this case, j_2 delivers the packet to j_4 and j_5 with adjusted values of $P^{req} = 0.3$ and $P^{req} = 0.3$, respectively, (the total reaching probability through these two nodes is $1-(1-0.3)(1-0.3) = 0.51$) to meet the requirement $P^{req} = 0.5$. This way, each following node dynamically compensate the previous wrong decision as the packet travels to the final destination.

Along with this hop-by-hop dynamic compensation, we also employ *reliability back-pressure mechanism* to remedy the problem of local decision in a more global scope. Since the sending node i assigns P^{req} based on its local estimation, it is possible that the receiving node j cannot satisfy the assigned P^{req} even with the hop-by-hop dynamic compensation using all possible forwarding nodes. If this over-expectation is detected by node j, it issues reliability back-pressure packet to reduce the reliability expectation of previous nodes. Specifically, the receiving node j detecting over-expectation issues a back-pressure packet with its maximum possible TRP that can be calculated by Equation (2). If the previous sender node i receives this back-pressure packet, it will use TRP as the maximum value of P^{req} that can be assigned to node j for delivering future packets. In an extreme case, this back-pressure can propagate to the original source so that P^{req} assignment can be made correctly from the beginning. This way, we can remedy the incorrectness of local decision more globally when necessary. A node that receives a back-pressure packet starts a timer called $T_{backpressure}$ to return to the normal operation expecting that the conditions that caused the generation of back-pressure packets have been resolved.

With this probabilistic multipath forwarding, we can differentiate packets with different reliability requirements and also the probability that a packet reaches the destination is likely higher than its requirement.

C. Discussion for Meeting Both Timeliness and Reliability: On-Time Reachability

By combining aforementioned timeliness and reliability guarantee mechanisms, we expect that our proposed 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, $P^{req}(x)$. 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 as explained in Section II-A. 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 as explained in Section II-B. Then, the packet is delivered to the chosen forwarding nodes.

When we deliver a packet to multiple nodes, it is important to ensure "parallel progress" along multiple paths so that each copy can meet the end-to-end deadline. Sending copies one by one to chosen neighbors may cause the later transmitted copy to miss the deadline even though the following nodes can guarantee the progress speed. For this reason, it is important to deliver a packet to multiple nodes using MAC layer multicast service based on broadcast nature of wireless medium rather than multiple calls of MAC unicast service[2]. We will discuss this MAC multicast in Section III.

Since each copy denoted by x_c of a packet x progresses in parallel and its progress speed is guaranteed by the network-wide speed mechanism, the copy that eventually reaches the destination can meet the deadline with a high probability. This can be rephrased with conditional probability—the probability that a copy x_c meets the deadline $deadline(x)$ under the condition of reaching the destination is approximately 1.0,

$$P(e2eDelay(x_c) \leq deadline(x) \mid x_c \text{ reaches the destination}) \approx 1.$$

From now on, for the simplicity of equations, we will use $x_c^{deadline}$ to represent the condition $e2eDelay(x_c) \leq deadline(x)$ and x_c^{reach} for the condition "x_c reaches the destination". Also, the number of copies of a packet is determined in a way that the total reaching probability TRP is greater than or equal to the required reachability. Thus, the probability that at least one copy reaches the destination before the deadline can be derived as follows:

$$\begin{aligned}
&P(\text{at least one copy reaches destination before deadline}) \\
&= 1 - \prod_{\forall x_c}(1 - P(x_c^{deadline} \text{ AND } x_c^{reach})) \\
&= 1 - \prod_{\forall x_c}(1 - P(x_c^{deadline} \mid x_c^{reach})P(x_c^{reach})), \\
&\text{since } P(x_c^{deadline} \mid x_c^{reach}) \approx 1, \\
&P(\text{at least one copy reaches destination before deadline}) \\
&\approx 1 - \prod_{\forall x_c}(1 - P(x_c^{reach})).
\end{aligned}$$

Note that $1 - \prod_{\forall x_c}(1 - P(x_c^{reach}))$ is TRP in Equation (2) which is greater than or equal to the required reaching probability $P^{req}(x)$. Thus, we can meet the combined metric of on-time reachability.

III. MAC Layer Features to Support MMSPEED Routing Protocol

Our proposed MMSPEED protocol alone cannot provide differentiated QoS guarantees. The proposed MMSPEED protocol relies on the premise that the underlying MAC protocol can perform the following functions:

- Prioritized access to shared medium depending on the speed layer,
- Reliable (or partially reliable) multicast delivery of packets to multiple neighbors,
- Supporting measurement of average delay to individual neighbors.
- Supporting measurement of loss rate to individual neighbors.

This section proposes extension of existing MAC protocols to best support MMSPEED routing protocol since none of current MAC protocols [17], [18], [19] can fully support the above requirements.

In the sensor network environment, it is highly desirable that the MAC protocol operates without any centralized control. Thus, our MAC protocol is mainly based on distributed CSMA with RTS/CTS collision avoidance, following the strategy of DCF (Distributed Coordination Function) mode of IEEE 802.11 standard [20]. The prioritization can be achieved by differentiating inter-frame spacings (IFS) and backoff mechanisms. The basic idea is to assign shorter IFS and backoff interval to higher priority class packets so that they can have higher chances to access the shared medium [21], [22], [17]. This way, we can realize the inter-class prioritization in a statistical sense even though it is not the ideal prioritization. The statistical prioritization is sufficient for speed layer isolation in MMSPEED. We adopt IEEE 802.11e [20] prioritization with only minor changes. Each speed layer of MMSPEED is mapped to one MAC priority class, i.e., highest-speed to highest priority, second speed to second priority, and so on. This way, we can minimize inter-node priority inversion such that a high-speed packet in one node is not likely blocked by a low-speed packet in another node.

Along with the prioritization, our MAC protocol maintains the average delay to each neighbor at each priority level. Specifically, in node i, when a request comes from MMSPEED to send a packet to neighbor j with priority-level l, a time stamp t_1 is associated with it. When node i receives ACK for the packet from node j, another time stamp t_2 is attached. Using t_1 and t_2, the MAC layer delay Δt can be calculated by

$$\Delta t = t_2 - t_1 - SIFS - ACK$$

where SIFS is the Short Inter-Frame Spacing between the data and acknowledgment frames and ACK is the transmission delay of the acknowledgement frame. With this delay measurement, we maintain the exponential moving average of MAC layer delay to neighbor j at priority-level l. This MAC layer delay is included in the overall progress delay $delay_{i,j}^{l}$ that is used by MMSPEED in Section II-A for estimating the progress speed with speed-level l to select feasible forwarding nodes.

A more challenging problem is the reliable multicast support for multi-path forwarding of MMSPEED. One simple approach is to repeatedly use the unicast sequence of RTS/CTS/DATA/ACK for reliable transmission to all recipients. However, it violates the "parallel progress" property by serializing the transmissions. Hence, later transmitted copies may experience longer delays, eventually missing their dead-

[2] A unicast based multipath forwarding also consumes more wireless channel resources.

lines. The other extreme approach is to simply use the broadcast nature of shared medium by transmitting a packet without RTS/CTS and ACK. If all the designated recipients can hear the packet successfully, all the copies received by the recipients can progress in parallel along multiple paths. However, without RTS/CTS and ACK, the probability of delivery success is very low.

Our MAC protocol aims to keep a balance between these two extremes. We select one of the recipients as the *primary recipient*, which will respond to the RTS frame with the CTS frame. Since the routing is performed based on the geographic information, we expect that there is high correlation among the locations of the multicast frame recipients and thus a single CTS frame provides a solution to the hidden node problem for most recipients with a high probability. Also, only the primary recipient has the responsibility to acknowledge a received frame. Consequently, the sender node waits for ACK only from the primary recipient. If a timer expires before the acknowledgement, the sender retransmits up to MAX times before dropping the frame. Thus, in the timing perspective, it is like RTS/CTS/DATA/ACK unicast sequence except that the designated recipients eavesdrop the data. This multicast mode, which we call a *partially reliable multicast* mode, guarantees reliable transfer to the primary recipient only. Secondary recipients never obtain the possibility of having their frames retransmitted unless they eavesdrop the retransmissions for the primary recipient. However, we can expect that secondary recipients can receive the frame containing the packet with similar probabilities as the primary recipient due to their geographic correlation.

Even though a secondary recipient does not respond to RTS and DATA, it counts the number of received frames and reports the number to the sender whenever it is selected as the primary recipient. This report is used by the sender to estimate the MAC layer loss rate. The sender node also keeps track of the number of frames it sends to each of their neighbors as secondary recipients. When the sender receives the report piggybacked in ACK frame from a recipient, it updates the exponential moving average of the loss rates to the recipient either as primary or secondary. After these calculations, both counters at the sender and the primary recipient are reset to zero.

This MAC layer loss rate is included in the overall loss rates from node i to node j, $e_{i,j}^{primary}$ for primary recipient case and $e_{i,j}^{secondary}$ for secondary recipient case, which are used by MMSPEED in Section II-B for estimating the number of forwarding nodes to meet the required reaching probability. The decision on primary and secondary recipients are made by MMSPEED protocol. $e_{i,j}^{primary}$ is applied for the primary recipient and $e_{i,j}^{secondary}$ for secondary recipients when MMSPEED calculates the total reaching probability using Equations (1) and (2). MMSPEED selects neighbor nodes as primary in a round-robin manner so that each neighbor can report its status quite frequently without starvation.

IV. Experimental Results

This section presents our simulation results. We conducted extensive simulation of the proposed MMSPEED protocol using J-SIM network simulator [23] and its performance is compared with SPEED [7], which is the only protocol available in the literature that can provide real-time services in a localized way in sensor networks. The general simulation environment is mainly drawn from [7] for fair comparison and summarized in Table I.

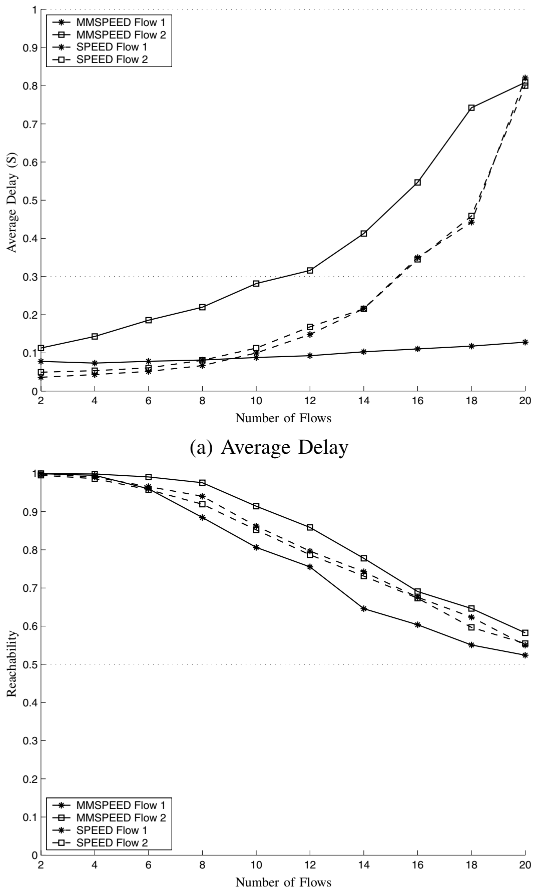
TABLE I
Simulation Environment Settings

Bandwidth	200Kbps
Payload	32 bytes
Terrain	200m×200m
Node number	100
Node Placement	Uniform
Radio Range	40 m

A. Service differentiation and guarantee

In the first experiment, we show the service differentiation in the timeliness domain by MMSPEED. For this, we randomly pick n source nodes on the left side of the network and each source node generates a traffic flow with Poisson packet arrival at a rate of 5 packets/sec. All the flows have the same destination on the center of the right edge of the network. In order to focus on the timeliness domain, we use the same and non-strict reliability requirement of 0.5 for all n flows. However, we divide n flows into two groups: one (flow group 1) has a strict real-time requirement, i.e., short end-to-end deadline of 0.3 sec and the other (flow group 2) has long end-to-end deadline of 1.0 sec. In MMSPEED protocol, we use two speed levels of 1000 m/sec and 250 m/sec while the SPEED protocol uses one highest speed level 1000 m/sec to meet the most urgent packet requirement. Figure 4(a) shows the average end-to-end delay for each flow group as increasing the number of flows n (solid lines for MMSPEED and dashed lines for SPEED). The figure shows that MMSPEED can provide clear differentiation of delay for two groups of flows with different end-to-end deadline requirements. As a result, the average end-to-end delay for each group is under the end-to-end deadline up to 20 flows. On the other hand, SPEED protocol cannot differentiate the two flow groups and thus the average delay for flow group 1 is under the deadline 0.3 sec only up to 14 flows.

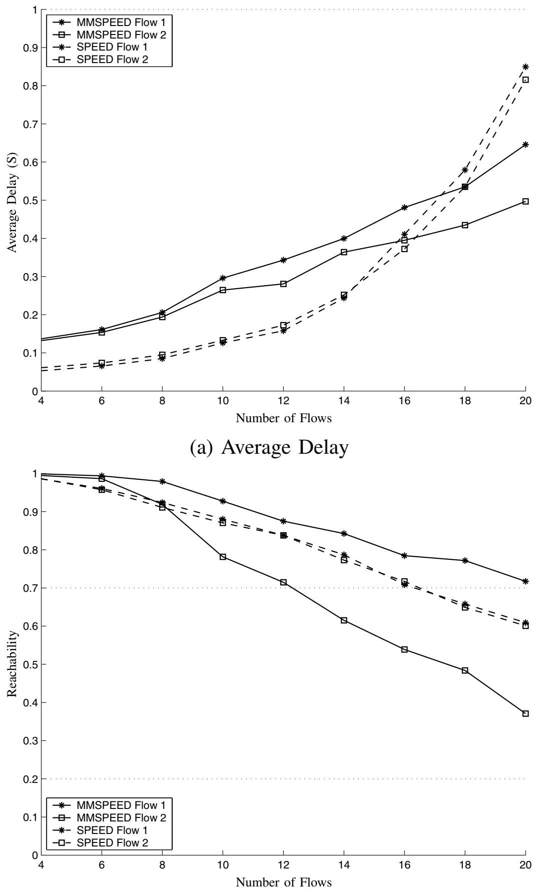
Figure 4(b) shows the reachability for each group of flows by each protocol. There is no big performance difference in the reliability domain since every flow has the same reliability requirement in this experiment. One important observation is that the reachability decreases for both MMSPEED and SPEED as increasing the number of flows. This is because both MMSPEED and SPEED regulate the workload level by probabilistically dropping packets when the injected workload becomes higher, in order to guarantee the network-wide speed with a high probability.



(a) Average Delay

(b) Reaching Probability

Fig. 4. Timeliness Differentiation.



(a) Average Delay

(b) Reaching Probability

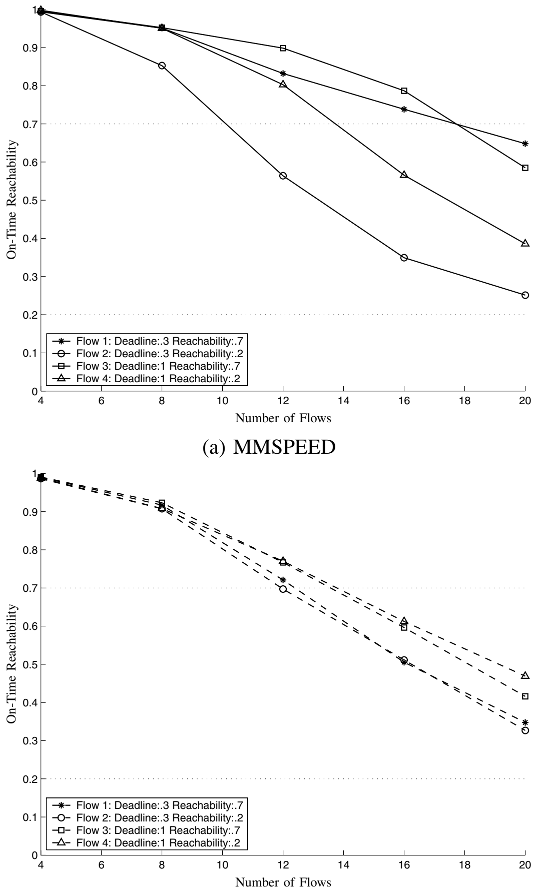
Fig. 5. Reliability Differentiation.

In the second experiment, we show the capability of our protocol to differentiate services in the reliability domain. For this, we use two flow groups with different reliability requirements (flow group 1—high reliability of 0.7 and flow group 2—low reliability of 0.2) but the same and non-strict deadline requirement of 1 sec. As before, MMSPEED has two preset speed levels of 1000 m/sec and 250 m/sec. For SPEED protocol, we used low speed level 250 m/sec so that less packets need to be dropped for speed guarantee, which gives the favor to SPEED in the reliability domain. Figure 5(b) shows that MMSPEED can provide clear service differentiation in the reliability domain and thus both flow groups can meet their own reliability requirements. This differentiation can be explained by two features in MMSPEED. First, MMSPEED controls the number of paths to deliver packets depending on their reliability requirements. This multipath routing can compensate the reliability loss by packet drops for network-wide speed guarantee. Second, when some packets need to be dropped for network-wide speed guarantee, MMSPEED drops packets according to their reliability requirements–packets with low reliability requirements are dropped with high probabilities. On the other hand, in SPEED protocol, two flow groups are mixed up with no differentiation, which makes flow group 1 miss reliability requirement of 0.7 for 18 flows and more.

Figure 5(a) shows the average delay as a reference. No big difference in delay can be observed since all flows have the same deadline requirement. The average delay for each flow group in each protocol is much lower than the non-strict end-to-end deadline requirement of 1.0 sec.

In the previous two experiments, we showed MMSPEED's capability for service differentiation in timeliness and reliability domain. The service differentiation, however, does not imply the end-to-end service guarantee in the combined metric of on-time reachability. To justify the MMSPEED protocol in the sense of guaranteeing the end-to-end on-time reachability, we conduct another experiment with mixed traffics. For this, we divide the n flows into four groups: 1) flow group 1 with short deadline 0.3 sec and high reachability 0.7, 2) flow group 2 with short deadline 0.3 sec and low reachability 0.2, 3) flow group 3 with long deadline 1.0 sec and high reachability 0.7, and 4) flow group 4 with long deadline 1.0 sec and low reachability 0.2.

Figures 6(a) and (b) show on-time reachability for each flow group by MMSPEED and SPEED, respectively. In general, the flow groups with high reachability requirement, i.e., flow



(a) MMSPEED

(b) SPEED

Fig. 6. On-Time Reachability.

groups 1 and 3 have higher on-time reachability than the other two flow groups. If we compare flow groups 1 and 3 with the same high reachability, flow group 1 with short deadline generally gives lower on-time reachability than flow group 3 with long deadline. This is because flow group 1 needs to use speed level 1000 m/sec to meet the short deadline and hence some packets experience probabilistic dropping for network-wide guarantee of high speed level 1000 m/sec while low speed class packets in flow group 3 are not likely dropped. However, the dropping can be compensated by utilizing the just adequate number of paths for routing flow group 1 packets and thus the resulting on-time reachability can be met up to 16 flows. For the two flow groups with low reachability, i.e., flow groups 2 and 4, their on-time reachability is generally lower than other two groups. If we compare flow group 2 and 4 with the same low reachability, there is a big gap. This is because the packet dropping in MMSPEED is affected not only by reliability requirement but also by required speed. The packets in flow group 4 have long deadline and thus low speed class is good enough. Thus, they are not likely to experience packet dropping for network-wide speed guarantee. However, the packets in flow group 2 have short deadlines requiring the high speed level. They are more likely to be dropped for network-wide speed guarantee. Also, their reliability requirement is low and thus they will be dropped with the highest probability if dropping is needed. Summarizing Figure 6(a), MMSPEED can afford up to 16 flows meeting on-time reachability of all flows.

On the other hand, SPEED protocol can afford only up to 12 flows meeting on-time reachability of all flows as shown in Figure 6(b). Another observation from Figure 6(b) is that flow groups 3 and 4 give higher on-time reachability than flow groups 1 and 2. This is because SPEED does not differentiate traffics and hence the average delay for each flow is similar. Therefore, packets of flow 3 and 4 with longer deadline can have higher chances to meet the deadline compared to flow 1 and 2 with shorter deadline.

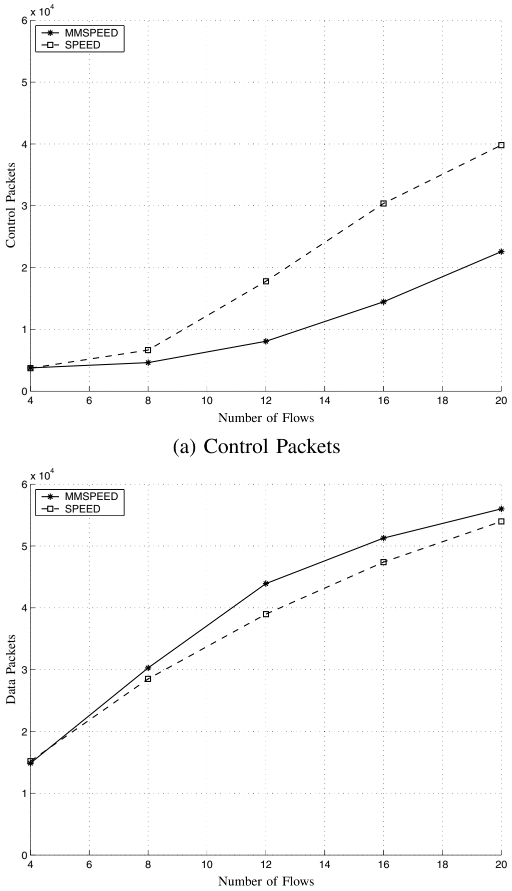
B. *Overhead Analysis*

This subsection compares the overhead of MMSPEED and SPEED protocols. We consider two types of overhead. The first type is the control overhead that includes 1) location update packets periodically broadcast to immediate neighbors, 2) timeliness back-pressure packets for speed control, 3) reliability back-pressure packets for reliability control. The first two control packets are required by both MMSPEED and SPEED protocols while the third control packets are required only by MMSPEED. The second type is data packet multiplication overhead required for leveraging multi-path routing by MMSPEED.

Figures 7(a) and (b) show the overhead of each protocol as increasing the number of flows. The flows are divided into four groups with different deadline and reliability requirements as in Figure 6. Figure 7(a) shows the total numbers of control packets generated by MMSPEED and SPEED for the whole duration of simulation. Unlike our simple intuition, the total number of control packets by MMSPEED is lower than SPEED. This can be explained as follows: 1) the number of periodic location update packets is same for MMSPEED and SPEED, 2) the number of timeliness back-pressure packets in SPEED is larger than MMSPEED since only half of traffic needs to use high speed class in MMSPEED while all traffic competes for the high speed class in SPEED resulting in many back-pressure packets, and 3) the number of reliability back-pressure packets which is an extra overhead of MMSPEED is quite small compared to the number of timeliness back-pressure packets.

To show the data packet multiplication overhead, Figure 7(b) shows the total numbers of data packets transmitted all over the network during the duration of simulation. The additional data packet transmissions for multipath routing in MMSPEED is surprisingly small compared to the total packet transmission in SPEED. This can be explained as follows. First, MMSPEED makes a just adequate number of packet multiplications mostly at early stages of route paths preventing exponential packet multiplications. Second, many copies of packets are dropped at early stages of route paths since those copies are assigned with smaller reliability requirements than the original one. This can bound the total number of packet transmissions. Finally, MMSPEED differentiates pack-

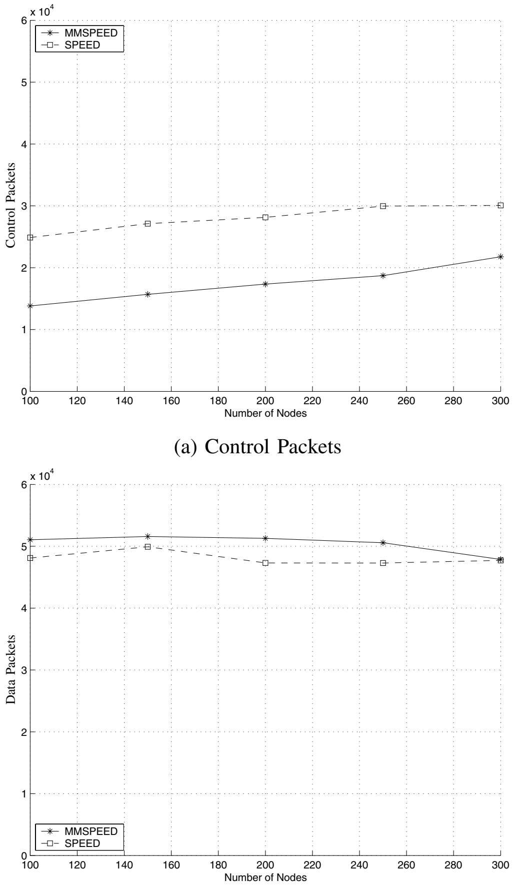
0-7803-8968-9/05/$20.00 (c)2005 IEEE



(a) Control Packets

(b) Data Packets

Fig. 7. Overhead vs Number of Flows.



(a) Control Packets

(b) Data Packets

Fig. 8. Overhead vs Node Density.

ets depending on their reliability requirement and thus we can drop more packets with low reliability requirements than SPEED, giving more resource for multipath forwarding of high reliable packets. Therefore, the aggregated number of packet transmissions of MMSPEED becomes comparable with SPEED.

In order to see the scalability of MMSPEED, we measure the overhead as increasing the node density. Starting from the total 100 nodes, we incrementally add nodes at random locations. Figure 8(a) shows that the total number of control packets only linearly increases in both MMSPEED and SPEED mainly due to the location update packets periodically generated at each node. Such linear increase of control packets is the advantage of localized routing protocols, which makes them scalable. On the other hand, proactive or reactive routing protocols utilizing global topology information cause an exponential increase of control packets as increasing the node density. As we can see in Figure 8(b), the number of data packet transmissions is generally constant in both MMSPEED and SPEED because both protocols can manage the similar hop counts regardless of node density using the geographic forwarding node selection. This is another nice property of MMSPEED and SPEED for the scalability.

C. Adaptability to Dynamic Topology Changes

Until now, we use static network topology where each node is placed at a fixed position. In order to show the adaptability of MMSPEED to the dynamic topology changes, we conduct another experiment. In this experiment, we use a network with 150 nodes randomly placed and 12 flows divided into four groups with different requirements as before. All other parameters are same as the previous experiments. For the initial 400 sec the network is static. At the time instant of 400 sec, 20% of nodes start moving randomly. Those node are in motion for the next 200 sec. After that, they stop moving. For the whole duration of simulation, we measure the on-time reachability with moving window of 1000 packets.

Figure 9 shows the time trace of on-time reachability for only one flow with most strict requirements, i.e., deadline of 0.3 sec and reliability of 0.7 among all four flow groups. From this graph, we observe that MMSPEED can guarantee the on-time reachability not only for the stationary stage but also for the motion stage continuously adapting to network topology changes. The resulting on-time reachability in the motion stage is a little bit lower than that in the stationary stage. This on-time reachability loss in the motion state is because of the gap between node's neighbor table and the actual locations of

0-7803-8968-9/05/$20.00 (c)2005 IEEE

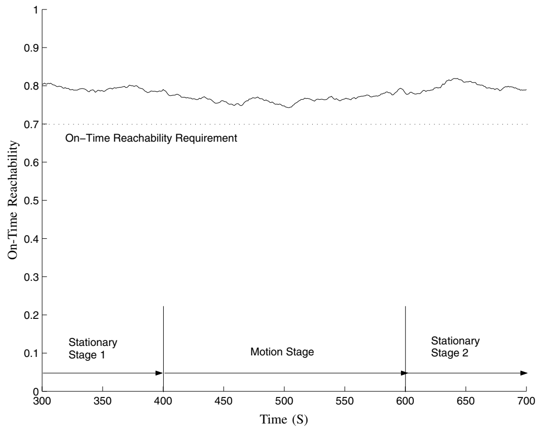


Fig. 9. Adaptability of MMSPEED.

neighbors since a node can notice that an existing neighbor leaves its radio range only after timeout without receiving location update packets from the neighbor. During the period of such gap, a node can forward packets to a node that is not within the radio range resulting in packet losses. We can reduce the period of such misforwarding, by increasing the location update frequency. However, this in turn increases the control overhead. Thus, it is a design issue to select a proper location update frequency by trading-off the adaptability and control overhead.

V. Related Work

In literature, several QoS provisioning protocols have been proposed for wireless ad hoc networks [2], [3], [4], [5]. However, they are based on end-to-end path discovery and resource reservation, which renders their application impractical for large scale dynamic sensor networks.

Recent QoS studies in sensor networks [7], [24], [9], [25], [6] focus on only one QoS domain, either timeliness or reliability. SPEED [7] protocol is designed to provide soft end-to-end deadline guarantees for real-time packets in sensor networks. However, it provides only one network-wide speed, which is not suitable for differentiating various traffic with different deadlines. RAP [24] provides service differentiation in the timeliness domain by velocity-monotonic classification of packets. Based on packet's deadline and destination, its required velocity is calculated and its priority is determined in the velocity-monotonic order so that a high velocity packet can be delivered earlier than a low velocity one. However, it is best-effort service differentiation without any guarantee in the end-to-end sense. Implicit EDF [9] can provide hard real-time guarantee based on decentralized EDF packet scheduling. However, it works only when most traffic is periodic and all periods are known a priori, which is not the case for many sensor network applications. Also, it is not adaptive to dynamics of sensor networks.

AFS [25] and ReInforM [6] are two examples that leverage path redundancy in wireless sensor networks for reliable delivery of packets. However, both protocols require the global knowledge of the network topology. They are also limited in differentiating services in the timeliness domain. Mobicast [26] aims at reliable and just-in-time delivery of alert packets to all sensor nodes in the moving delivery zone. This service is useful for waking up sensors ahead in the target trajectory being tracked. However, it assumes reliable and time-bounded transmission between every pair of sensor nodes and uses all nodes in a quite large forwarding zone to forward packets.

For QoS provisioning in timeliness and reliability domains, proper support by the MAC layer is needed. When a packet has an earlier deadline than others, it must be given preferential treatment over longer deadline packets not only within the same node but also in other nodes in the transmission radius. Furthermore, to leverage the multipath routing, it is necessary to multicast packets to a subset of neighbor nodes quickly and reliably. In literature, prioritization and reliable multicasting in MAC protocols for wireless networks have been addressed separately. Prioritization in wireless ad hoc networks can be achieved by manipulating inter-frame spacing and backoff strategies [21], [20]. Generally, a higher priority frame is assigned with a shorter inter-frame spacing and shorter congestion window range. We use this idea to support MMSPEED. Regarding MAC layer multicasting, some of the existing mechanisms like [8] do not ensure reliability at all. They simply broadcast a frame with designated recipient without RTS/CTS handshake, which results in a high collision probability. The other extreme is the reliable MAC layer multicast protocol of [19], which uses a separate RTS/CTS handshake for each of the recipients, followed by data transmission and another sequence of Request to Acknowledge (RAK)/ACK handshakes to ensure the reliable delivery to all multicast recipients. However, this incurs a long sequence of handshakes resulting in a long delay for each multicast. Our MAC protocol ensures the reliable frame transmission only to primary recipient expecting successful eavesdropping by all other recipients. This partial reliable multicasting is good enough for supporting MMSPEED since we have a reliability backup by multipath routing in the network layer.

VI. Conclusion

In this paper, we propose a novel packet delivery mechanism called MMSPEED for wireless sensor networks to provide service differentiation and probabilistic QoS guarantees in timeliness and reliability domains. For the timeliness domain, we provide multiple network-wide speed options so that various traffic types can dynamically choose the proper speed options for their packets depending on their end-to-end deadlines. For the reliability domain, we use probabilistic multi-path forwarding to control the number of packet delivery paths depending on the required end-to-end reaching probability. These methods are implemented in a localized way with dynamic compensation to compensate for the inaccuracies of local decisions as packets progress towards their destinations. Since the proposed mechanisms work locally at each node without global network state information and end-to-end path setup, it can preserve desirable properties such as scalability for large sensor networks, self adaptability to network dynamics, and appropriateness for urgent aperiodic and periodic packets. Simulation results show that MMSPEED can efficiently cater for the needs of various traffic types with different combinations of reliability and timeliness requirements. As a result, MMSPEED can significantly improve the effective capacity of a sensor network in terms of number of flows meeting both reliability and timeliness requirements.

Our future work includes finding optimal settings of number of speed levels and $SetSpeed$ values depending on network density and workload characteristics. We will also investigate how to extend the proposed mechanisms to the power consumption domain to balance power consumption in the network to prolong sensor network lifetime.

References

[1] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *In Proceedings of the first international conference on Embedded networked sensor systems*, November 2003.

[2] T. Chen, J. Tsai, and M. Gerla, "QoS Routing Performance in Multihop Multimedia Wireless Networks," *Proceedings of IEEE Sixth International Conference of Universal Personal Communications*, vol. 2, pp. 557–561, 1997.

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

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

[5] B. Hughes and V. Cahill, "Achieving real-time guarantees in mobile ad hoc wireless networks," in *In Proceedings of the Work-in-Progress session of the 24th IEEE Real-Time Systems Symposium*, December 2003.

[6] B. Deb, S. Bhatnagar, and B. Nath, "ReInForm: Reliable Information Forwarding Using Multiple Paths in Sensor Networks," *Proceedings of IEEE International Conference on Local Computer Networks*, pp. 406–415, 2003.

[7] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks," *Proceedings of IEEE International Conference on Distributed Computing Systems*, pp. 46–55, 2003.

[8] F. Stann and J. Heidemann, "RMST: Reliable Data Transport in Sensor Networks," *Proceedings of IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 102–112, 2003.

[9] M. Caccamo, L. Zhang, L. Sha, and G. Buttazzo, "An Implicit Prioritized Access Protocol for Wireless Sensor Networks," *Proceedings of IEEE Real-Time Systems Symposium (RTSS'02)*, pp. 39–48, 2002.

[10] B. Karp and H. Kung, "Greedy Perimeter Stateless Routing for Wireless Networks," *Proceedings of IEEE/ACM International Conference on Mobile Computing and Networking*, pp. 243–254, 2000.

[11] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," in *Proceedings of Mobicom*, 2003.

[12] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex Positiion Estimation in Wireless Sensor Networks," in *Proceedings of Infocom*, 2001.

[13] Y. B. Ko and N. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," in *Proceedings of ACM/IEEE 4th Int'l Conf. on Mobile Computing and Networking*, October 1998.

[14] I. Stojmenovic and X. Lin, "GEDIR: Loop-Free Location Based Routing in Wireless Networks," in *Proceedings of IASTED Int. Conf. on Parallel and Distributed Computing and Systems*, Nov. 1999.

[15] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.

[16] S. Dulman, T. Nieberg, J. Wu, and P. Havinga, "Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks," in *Proceedings of the Wireless Communications and Networking Conference*, 2003.

[17] R. Baldwin, N. Davis, S. Midkiff, and R. Raines, "Packetized Voice Transmission Using RT-MAC, a Wireless Real-time Medium Access Control Protocol," *ACM Mobile Computing and Communications Review*, vol. 5, no. 3, pp. 11–25, 2001.

[18] K. Tang and M. Gerla, "MAC Reliable Broadcast in Ad Hoc Networks," *Proceedings of IEEE Military Communications Conference (MILCOM 2001)*, vol. 2, pp. 1008–1013, 2001.

[19] M. Sun, L. Huang, A. Arora, and T. Lai, "Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks," *Proceedings of IEEE International Conference on Parallel Processing*, vol. 2, pp. 527–536, 2002.

[20] I. . WG, "Draft Supplement to Standard for Telecommunications and Information Exchange between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)," *IEEE 802.11e/D3.0*, May 2002.

[21] W. Pattara-Atikom, P. Krishnamurthy, and S. Banerjee, "Distributed Meachanisms for Quality of Service in Wireless LANs," *IEEE Wireless Communications*, vol. 10, no. 3, pp. 26–34, June 2003.

[22] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed Priority Scheduling and Medium Access in Ad Hoc Networks," *ACM/Kluwer Wireless Networks*, no. 8, pp. 466–466, 2002.

[23] "DRCL J-Sim," http://www.j-sim.org.

[24] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks," in *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2002)*, San Jose, CA, September 2002.

[25] S. Bhatnagar, B. Deb, and B. Nath, "Service differentiation in sensor networks," in *In Proceedings of Fourth International Symposium on Wireless Personal Multimedia Communications*, September 2001.

[26] Q. Huang, C. Lu, and G.-C. Roman, "Spatiotemporal multicast in sensor netwroks," in *Proceedings of the first international conference on Embedded networked sensor systems*, November 2003.