

# ReInForM: Reliable Information Forwarding using Multiple Paths in Sensor Networks

Budhaditya Deb, Sudeept Bhatnagar and Badri Nath  
 Dataman Lab, Dept. of Computer Science  
 Rutgers University, Piscataway, NJ, USA  
 {bdeb,sbhatnag,badri}@cs.rutgers.edu

## Abstract

Sensor networks are meant for sensing and disseminating information about the environment they sense. The criticality of a sensed phenomenon determines its importance to the end user. Hence data dissemination in a sensor network should be *information-aware*. Such information-awareness is essential firstly to disseminate critical information more reliably and secondly to consume network resources proportional to criticality of information.

In this paper, we describe a protocol called ReInForM to support information awareness in sensor networks. Using ReInForM, data can be delivered at desired levels of reliability at proportional cost, in spite of the presence of significant channel errors. It uses the concept of dynamic packet state in context of sensor networks to control the number of paths required for the desired reliability using only local knowledge of channel error rates and does not require any prior computation or maintenance of these multiple paths. We also show that for uniform unit disk graphs, multiple edge-disjoint paths numbering as many as the average node degree, exist between any source and sink with very high probability. These paths exist in a thin band between source and sink having low deviation from the optimal path. ReInForm utilizes this property in its randomized forwarding mechanism which results in use of all possible paths in this band. Thus as a side effect ReInForM leads to load balancing as well.

## I. INTRODUCTION

A sensor network is comprised of small devices (or sensor nodes) with computation, communication and sensing capabilities [1], [2], [3], [4]. Since sensor nodes are highly resource constrained, all applications and protocols should consume minimal network resources. Hence research in sensor networks focused on energy efficiency while providing basic networking and information dissemination functionality [5], [4].

However the main purpose of a sensor network is to disseminate information about the environment it senses and hence the data dissemination process should be *information-aware*. Consider a case where a temperature sensor in a forest, senses 60F on a normal spring day. On the other hand another sensor senses a temperature of 1000F at the same time. From a monitor's point of view, the packet containing the 1000F temperature is much more important as it could indicate a forest fire and hence should reach the monitoring node with high reliability and low latency. Note that reliability in data delivery is an important consideration since sensor networks typically would have high channel errors. Since nodes may be deployed at random, sometimes in hostile terrain, there could be significant deviation in the local channel errors from the global estimate. Also if the number of hops from source to sink is high, the probability of a packet reaching the sink becomes very low.

Data centric routing [5] considers the information dissemination aspect of a sensor network by giving more importance to the data, rather than the source of the data. However it doesn't consider the fact that the reliability in data delivery should depend on the *information content* in data being sent. Flooding can be used to deliver packets with high reliability but the overhead incurred is significant. Using default routing for normal data and flooding for critical data, does not provide a graceful transition, both in terms of attained reliability and incurred overhead. Also this kind of a scheme would overlook the fact that achieved reliability is heavily dependent on prevalent channel error rates. Any data dissemination protocol which is not adaptive to channel error rates and does not have a notion of information awareness would either be spending too much resources for unimportant information or would not be delivering really important information with high reliability.

The above highlight the need for information-awareness in sensor networks and adaptability to channel errors along with differential consumption of network resources based on the criticality of data. This paper is intended to solve the above problems. Based on the criticality of data inside a packet different priority levels are assigned. Each priority level maps to a desired reliability in data delivery. The main concept used in this paper is introduction of redundancy in data delivered to increase probability of data delivery. This redundancy is in the form of multiple copies of the same packet which travels to the destination among multiple paths.

We propose a multi-path forwarding protocol called ReInForM for providing desired reliability in data delivery based on packet priority. It uses *only local knowledge* of error rates and neighborhood at each node. We achieve this by a technique analogous to dynamic packet state [8], [9]. Packets carry minimal state aiding in forwarding decisions at a node so as to provide desired reliability at the global scale. Our algorithm does not require any data caching at any node which is important because sensor nodes have limited memory. Use of dynamic packet state and randomized forwarding mechanism leads to use of all nodes in the area between source and sink at random. Thus the mechanism as a side effect achieves load balancing effect as well. Even for single path forwarding, ReInForM is desirable because of its load balancing effect.

The protocol is supported by our observations on the existence of multiple edge-disjoint paths in connected unit disk graphs of uniform density. Our simulations showed that for connected unit disk graphs the number of edge disjoint paths is equal to the average node degree with high probability. Moreover very few of these paths have significant deviation in the number of hops from the optimal. In our simulations the *average* deviation was less than 2 hops for all cases considered. This implies that almost all these disjoint paths exist in a thin band between the source and the sink.

The paper is organized as follows: In section 2 we discuss some of the related work. In section 3, we describe the network model and assumptions used in our work. In section 4 we discuss our study of multiple paths in random unit disk graphs which supports the idea of multipath approach to providing reliability. In section 5 we describe the ReInForM in detail. Section 6 gives some simulation results for the performance of our algorithm. Finally in section 7, we discuss conclusions and future work.

## II. RELATED WORK

Support for information-awareness in sensor networks can be provided using a framework similar to differentiated services [6]. Every data packet is assigned a priority level based on its information content and criticality. When a node receives a packet, it looks at its priority level and acts accordingly to provide the desired reliability. The concept of such a differentiated services and *data prioritizing* framework for sensor networks was introduced in [7]. A multipath-multipacket forwarding protocol was proposed for delivering data packets at required reliability based on data priority. By using redundancy in packets the protocol controls the reliability of packet delivery. The protocol uses a probabilistic flooding scheme to create multiple paths from source to sink. It adapts to any channel error and topological changes. However the main drawback of the algorithm is that it requires periodic update of forwarding parameters to adapt to the changes. ReInForM does not require any periodic reinforcements.

In [12], multiple paths from source to sink is used in diffusion routing framework [5] to quickly recover from path failure. The multiple paths provided by such protocols could be used for sending the multiple copies of each packet. However it incurs extra overhead of multiple path formation and maintenance of path state in each node and is not adaptive to channel errors.

Similarly for mobile ad-hoc networks different multipath extensions to well known routing algorithms have been proposed [13], [14], [15], [16]. However the main purpose of these multipath routing schemes is to increase robustness in paths by quickly recovering from broken paths. they are also not adaptive to local channel error rates.

For streaming video over a Wireless Lan, different schemes for prioritized forward error correcting codes have been proposed [18], [17]. They allow different resolution layers in streaming video to be protected by different channel codes based on their importance. Based on the local channel error and packet importance similar approach could be adopted to provide the desired reliability. However for sensor networks, computationally simpler algorithms have to be designed since nodes have limited computation power and memory. This is an area of future work.

## III. NETWORK MODEL & TERMINOLOGY

In this section, we describe the network model that we assume for the underlying sensor network. The network graph  $G = (V, E)$  has  $|V| = n$  vertices and  $|E| = m$  edges. The vertices represent the sensor nodes which have uniform random distribution. The edges represent the wireless communication links between nodes. The links are assumed to be symmetric (the graph is undirected) and all nodes have equal communication radius  $R$ . Thus,  $G$  is a *unit disk graph* in which an edge  $(v_i, v_j) \in E$  iff  $distance(v_i, v_j) \leq R$ .

There is a single monitoring node (also called the sink) to which each sensor sends its data.<sup>1</sup> The sink periodically broadcasts a routing update packet which is flooded throughout the network with each node forwarding it at most

<sup>1</sup>Having multiple monitoring nodes does not require any changes to our protocol. We assume so for the ease of exposition.

TABLE I  
THE AVERAGE DEVIATION IN NUMBER OF HOPS FROM OPTIMAL FOR DIFFERENT NUMBER OF NODES

Radius(m)	10	12.5	15	17.5	20	22.5	25	27.5	30
100 nodes	0.02	1.76	1.1	1.07	1.03	0.94	0.89	0.96	0.79
200 nodes	2.2	2.03	1.2	1.12	1.2	0.98	0.91	0.9	0.96
300 nodes	1.28	1.32	1.2	0.99	0.92	0.89	0.82	0.85	0.89

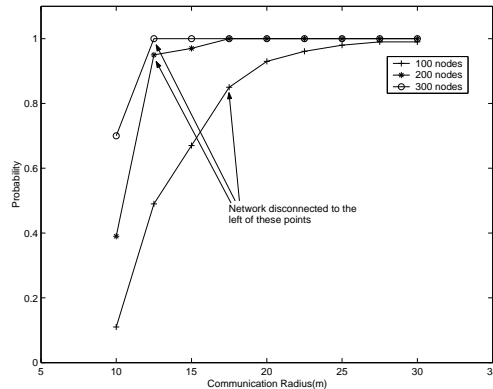


Fig. 1. Probability of Edge Disjoint Paths equal to node Degree against density

once. During each routing update a node  $i$  becomes aware of its neighbors and the number of hops it is away from the sink  $h_i$ . Node  $i$  also notes down the number of its neighbors which are  $h_i - 1, h_i, h_i + 1$  hops away from the sink.

We assume a TDMA MAC layer in the network as proposed in [11] for collision free transmissions. Each node is also assumed to have knowledge of the local channel error which can be obtained over a period of time. The number of neighbors of node  $i$  which is also its degree in  $G$ , is represented by  $d_i$ . A packet has a *priority level*  $l$  where  $l \in l_1, l_2, \dots, l_n$ . The number of priority levels  $n$  may depend upon the application and the levels of differentiation desired by the user. Each priority level  $l_i$  is associated with a *reaching probability*  $P_i$ , i.e., a packet with priority level  $l_i$  is expected to reach the monitor with probability  $P_i$ .

#### IV. STUDY OF MULTIPLE PATHS IN RANDOM UNIT DISK GRAPHS

ReInForM provides desired reliability in packet delivery in presence of any channel error rate by sending multiple copies of a single packet along multiple paths. Hence it relies heavily on the existence of multiple paths from source to sink. Only if sufficiently large number of paths exist from source to sink without too much deviation in the number of hops from the optimal, the multipath approach would succeed. When the number of paths between any two nodes is not sufficient for providing the reliability required, there may be more one copy of a packet per path.

The reliability that can be achieved using multipath forwarding depends on the expected number of such paths. The number of edge disjoint paths gives a good bound for the above because number of paths would be at least as large the number of disjoint paths. We note that the maximum possible number of disjoint paths between a source and a sink is the minimum of node degrees of source and sink. We conducted several experiments to find relation between the number of edge-disjoint paths between a source and a sink and the  $\min\{degree(source), degree(sink)\}$ . To find edge-disjoint paths, we used a greedy algorithm which finds the shortest path between source and sink and removes all edges from the graph and finds the next shortest path in the current graph and so on.

Figure 1 shows the results for the case of a unit disk graph of uniform density. For a reasonable density, the number of disjoint paths between *any* pair of nodes is equal to the average node degree of the graph with very high probability. Thus there is a direct correlation in the average node degree with the number of edge-disjoint paths. Also the result indicated that the probability that the number of edge disjoint paths equals the maximum number possible (which is limited by the node degree), is very high when the network is connected. Reference [10] shows through simulation studies that expected neighborhood size for a network to be connected is six. Our simulations indicate that for a uniform random unit disk graph, a network slightly more dense than a minimal connected graph is

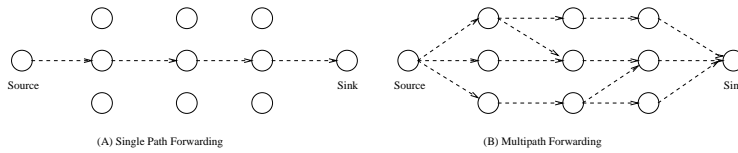


Fig. 2. Single Path Forwarding v/s Multi-Path Forwarding

sufficient to provide edge-disjoint paths equal to the average node degree.

Our second experiment studies the expected deviation (in hops) of these edge-disjoint paths from the optimal path. The simulation results shown in table I indicate that, for all the topologies and densities considered, the average deviation is not more than two hops. In fact, the maximum deviation for any single pair of node occurs for the nodes which are just 1 hop away from the sink (which is expected because the path remaining after removing all but one neighbors of the last-hop node would likely have a path length large relative to the optimal path length).

These two results significantly uphold the idea of multipath forwarding for reliability. The fact that the number of edge-disjoint paths from any node to the sink is nearly equal to the node's degree implies that the node has a lower bound on the number of paths at its behest. These paths being of nearly equal length means that choosing any path randomly for data delivery does not incur a penalty in terms of latency and at the same time allows us to do load balancing. These observations enable a node to take *local* decisions to determine the number of forwarding paths and the next hop for data forwarding. This coupled with corrections for local information form the basis of the ReInForM protocol.

## V. REINFORM - RELIABLE INFORMATION FORWARDING USING MULTIPLE PATHS

In this section we describe the multi-path forwarding algorithm, ReInForM for providing desired reliability in data delivery. The main idea is to provide reliability in data delivery by introducing redundancy in the form of multiple copies of each packet delivered through multiple paths from source to sink. Based on the local knowledge of some network conditions (channel error, hops to sink, out-degree etc.), the source decides to send multiple copies of packets through multiple paths. Each packet header contains information about network conditions from the previous step, which is used for forwarding decisions. As a packet travels toward the sink, these fields are updated at each node to account for local deviation in network conditions. The method is analogous to *Dynamic Packet State* (DPS) approach used for data networks [8], [9]. The key idea in DPS is to carry some state information in the packet header enabling the network to serve the packet in a desired manner even though the forwarding nodes do not maintain any packet state. While in data networks this technique has been used for providing bandwidth and delay guarantees to flows, we use it to provide a lightweight and localized mechanism for reliability in information dissemination. The mechanism is useful in sensor networks because it does not require *any* caching or state maintenance at intermediate nodes which would be required in any acknowledgment based scheme. It adapts to any channel errors and topological deviations using only local knowledge while maintaining the desired level of reliability at global scale. The overhead incurred is proportional to the desired reliability.

In sensor networks, wireless channel can have significant channel error. Hence the probability of a packet reaching the sink can be low if the number of hops from source to sink is high. In our approach we try to increase this probability by introducing redundancy in packets by forwarding copies of packets along multiple paths. This approach is depicted in figure 2. Note that while forwarding multiple packets along a single path can also be used to provide a desired reliability, it would overuse the default path and create hot spots. Moreover, the expected latency in such a case would be higher than that of multi-path approach where multiple copies of the packet can travel in parallel. We now describe the ReInForM protocol and later show its effectiveness.

### A. Protocol Description

On generating a packet the source node determines the importance of the information it contains and decides the desired reliability  $r_s$  for it. It also knows the local channel error  $e_s$  and its hop distance from sink  $h_s$ . Using these values, the source computes the number of paths (or equivalently, the number of copies of the packet to be sent)  $P$  required for delivering the packet at desired reliability to the sink as:

$$P(r_s, e_s, h_s) = \frac{\log(1 - r_s)}{\log(1 - (1 - e_s)^{h_s})} \quad (1)$$

The source broadcasts the packet and puts certain DPS(dynamic packet state) fields in its header. The key aspect of our algorithm is the DPS mechanism which allows the receiving nodes to take *local* decisions about forwarding the packets. Before describing the DPS fields, we bring forth some issues which need to be addressed using DPS:

- Each receiving node has to ascertain the reliability that the source expects it to provide from itself to sink. Since the source's computation was done using its local channel error and its distance from sink, the reliability expected of the receiving node should be adjusted accordingly.
- We don't want to cache any packets. Each receiving node should take a forwarding decision based on the contents of the packet's DPS fields.
- The direction of flow of each packet should be to-towards the sink as much as possible. While we could initially deviate from the optimal path length, once the required number of paths have been created, the copies of the packet should have a gradient to-towards the sink.

We address these issues in our algorithms as follows:

The source  $s$  is at a distance  $h_s$  from the sink. The neighborhood set of the source is divided into 3 subsets  $H_s^-$ ,  $H_s^0$  and  $H_s^+$  which contain neighbors at distance  $h_s - 1$ ,  $h_s$  and  $h_s + 1$  respectively. The source selects one of the members of  $H_s^-$  set at random, and designates it the *next hop*. Since the graph is assumed to be connected, there is at least one node in set  $H_s^-$ . This ensures that at least one member to-towards the sink forwards. Choosing a random next hop helps in load balancing. The source also has to compute the following DPS fields for the packet:

- $P_{H_s^-}, P_{H_s^0}, P_{H_s^+}$ : These values represent the number of paths that *each* node in  $H_s^-$ ,  $H_s^0$  and  $H_s^+$  respectively is required to create. I.e. if this value is greater than 1, then each node would forward the packet and more than one of its neighbors would forward.
- $e_s$ : The channel error at source. This value is used in association with local channel error at a receiving node to make local corrections to DPS fields.
- $h_s$ : The hop distance of the source to sink.

At each step, our aim is to create the desired number of paths while maintaining a forwarding bias to-towards the nodes having lower distance to sink. Thus paths for  $H_s^-$ ,  $H_s^0$ ,  $H_s^+$  are assigned in order such that first set is assigned paths first. Only if more paths are left, the second set gets assigned and so on. Since packets from nodes in  $H_s^0$  and  $H_s^+$  take one and two hops more than  $H_s^-$ , the paths are assigned with the following ratios.

$$P_{H_s^-} = \frac{P_{H_s^0}}{1 - e} = \frac{P_{H_s^+}}{(1 - e)^2} \quad (2)$$

$P_{H_s^0}$  is assigned only if number of paths  $P$  exceed number of nodes in  $H_s^-$  and  $P_{H_s^+}$  is assigned only if number of paths exceed number of nodes in  $H_s^-$  and  $H_s^0$ . The expected number of paths  $P$  to be spawned at the source is computed by subtracting  $(1 - e_s)$  from equation 3 as follows:

$$P = \frac{\log(1 - r_s)}{\log(1 - (1 - e_s)^{h_s})} - (1 - e_s) \quad (3)$$

This is done to compensate for the default path created by designating one next hop node to forward as described earlier. The probability of this packet reaching the default node is  $(1 - e_s)$ . Based on the expected number of paths  $P$ , the distribution among the three neighbor sets is done in ratios given by equation 2.

If a receiving node is the default next hop, it always forwards the packet. Other receiving nodes determine whether or not to forward the packet using the following criteria: If the path value  $P$  associated with it is greater than 1, it always becomes a forwarding node. If the value is less than 1, it becomes a forwarding node with a probability equal to the value. The nodes which decide not to forward the packet simply drop it. The other nodes which decide to forward, have to adjust the values of the DPS fields so that they remain consistent with the source's calculations. To do this each node  $i$  has to update  $(r_s, e_s, h_s)$  to  $(r_i, e_i, h_i)$  while taking into account the three local H-sets:  $H_i^-$ ,  $H_i^0$ ,  $H_i^+$ . Node  $i$  computes the required reliability  $r_i$  as:

$$r_i = 1 - (1 - (1 - e_s)^{h_s - 1})^P \quad (4)$$

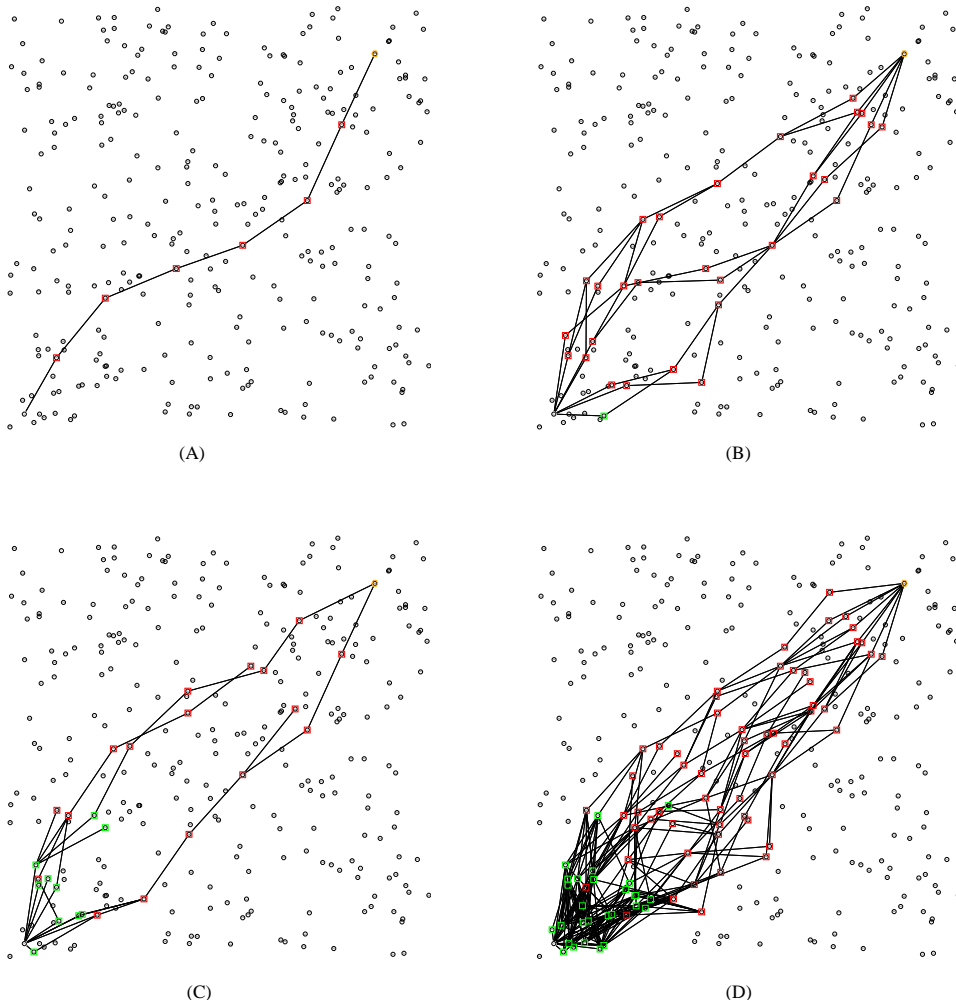


Fig. 3. Illustration of the Band Based Multipath Forwarding Paths. For all four cases the required reliability is 70%. The source is in lower left corner and the sink at the upper right corner of the graph. (A) shows the path taken by a single packet for zero channel error and (B) shows the band for zero channel error for 10 packets. (C) shows the paths taken for a single packet when channel error is 30% and (D) shows the band for 10 packets when channel error is 30%.

Effectively, the node now becomes a source in its own right. Using its local channel error  $e_i$  and its hop distance to sink  $h_i$ , the node uses the same process to compute the path values for its neighbors and the process continues.

The algorithm essentially creates a band of parallel paths between source and sink. The width of the band and the number of paths, depend on the error rates, desired reliability and hop distance between source and sink. Figure 3(A) and 3(C) show paths for a *single packet* between a pair of source and sink for error rates 0 and 0.3 respectively and desired reliabilities of 0.7 on the same topology. Figures 3(B) and 3(D) show the bands formed for 10 different packets between same pair of source and sink under the same conditions as earlier. Note that the forwarding duty is distributed across the band and is not concentrated on a single path. Similarly, the band is limited in width which can not be done using flooding.

## VI. SIMULATION

In this section we show some of the simulation results illustrating features of ReInForM. The simulations are done on a uniform topology consisting of 300 nodes spread in a square area of 100mx100m. The channel error rate is normally distributed across the field with the mean and deviation varying for different simulations. In all simulations, the source is located in lower left zone and the sink at the upper right zone as shown in figure 3.

The first simulation shows the reliability-overhead control of ReInForM. For this simulation, the communication

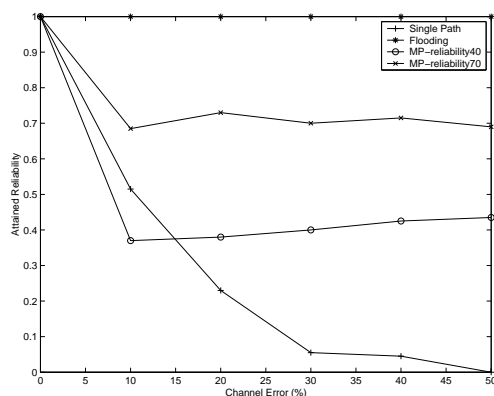


Fig. 4. Attained Reliability by flooding, single path forwarding and multipath forwarding targeting 40% and 70% reliability for ncreasing channel error

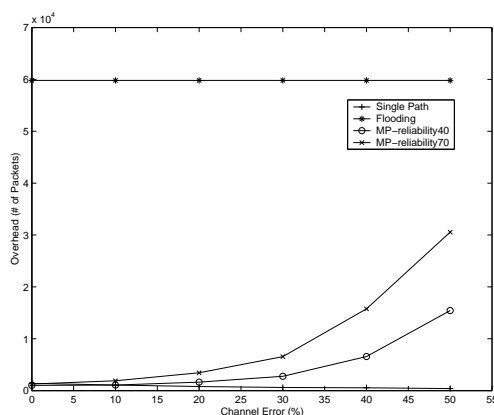


Fig. 5. Overhead incurred by flooding, single path forwarding and multipath forwarding targeting 40% and 70% reliability for ncreasing channel error

range of all nodes is set to 20m. This results in a 7 hop distance between source and sink. The source sends 200 packets to the

sink. The mean channel error is increased from 0 to 50%. Figure 4 shows the reliability attained for single hop forwarding, flooding and two instances of multipath forwarding at desired reliability of 40% and 70% respectively. We see that flooding always provides high reliability close to 100% whereas the reliability provided by single path forwarding is negligible. We see that ReInForM provides reliabilities close to their desired reliabilities.

Figure 5 shows the overhead incurred in terms of number of packets transmitted for each of the above cases. The overhead incurred by flooding is very high and that incurred by single path forwarding very low. The overhead incurred by the multipath forwarding schemes increases with the error rate but is significantly less than that of flooding even for an error rate of 50%. This experiment shows that even for small channel errors, the reliability provided by single path forwarding is very low. Flooding on the other hand incurs a huge overhead even when the channel conditions are good. Our multipath forwarding scheme adapts to the network condition and provides desired reliability at a reasonable cost. We see that when the expected number of paths required to provide desired reliability is close to the total number of nodes in the network (for example, with 60% error rate and required reliability of around 80%, the expected number of paths is well beyond 300 nodes), it is better to flood the network in a controlled manner, rather than rely on ReInForM.

The second simulation shows the impact of increasing desired reliability for a constant channel error. For this simulation the communication range of all nodes is 20m and the channel error is fixed at 25%. Figure 6 shows the attained reliability for flooding, single path forwarding and multipath forwarding. The corresponding overhead incurred is shown in figure 7. Again the adaptive nature of our algorithm is highlighted vis-a-vis the two extreme forwarding methods: flooding and single path forwarding.

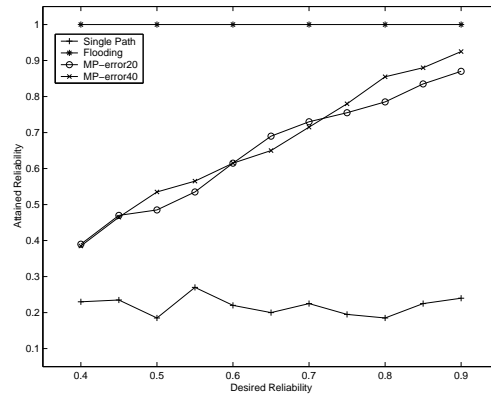


Fig. 6. Attained Reliability vs Desired Reliability for flooding, single path forwarding and multipath forwarding for channel error fixed at 20% and 40%

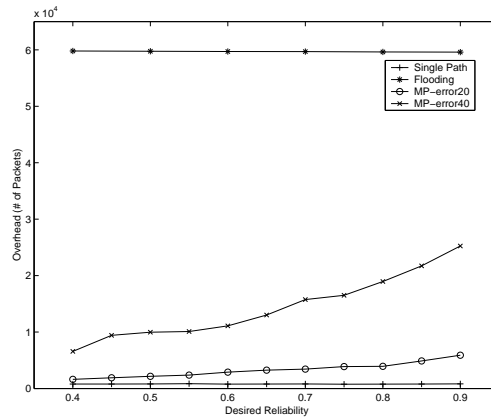


Fig. 7. Overhead incurred by flooding, single path forwarding and multipath forwarding under 20% and 40% channel errors

## VII. DISCUSSION

In this paper, we argue that mechanisms to support information awareness and service guarantees for information dissemination is essential for sensor networks. This is a preliminary investigation into the issue of information awareness in sensor networks. The work raises some important issues with respect to information awareness:

- What is Quality of Service(QoS) from the perspective of sensor networks? What metrics other than Reliability and Latency in data delivery are relevant for sensor networks?
- In an actual sensor network, what network services are realizable, i.e. are the lightweight mechanisms as proposed in this paper light enough?
- Is QoS a stand alone service that a network provides, or should it be used in conjunction with other network protocols. For example, in content aware data dissemination like directed diffusion, the sink propagates an interest in network and the nodes satisfying that interest reply back. It may be more relevant to have a push based mechanism where data dissemination is triggered based on the information content in the sensed data. In this regard, a node could use a QoS capable network to provide the sink with important information quickly. The sink in turn could generate a corresponding interest message to ask if there are other nodes which sense that important information as well. This removes the responsibility of “expecting the unexpected” from the sink.
- This paper assumes that packets are not cached in sensor nodes because of memory constraints. This may lead to multiple transmissions of the same packet at a node. It can be shown that overhead could be reduced significantly by caching the packets. To what degree is caching permissible in sensor networks in practice?
- This work assumes that the nodes can compute the information content and importance of sensed data. For example pattern matching or machine learning techniques may be used for this purpose. This aspect needs to be investigated further.



## REFERENCES

- [1] Embedded Everywhere: A Research Agenda for Networked System of Embedded Computers. NRC Report (2001)
- [2] G.J. Pottie and W.J. Kaiser: Wireless integrated network sensors. *Communications of the ACM*, vol. 43 no. 5 (2000) 51–58
- [3] J. Haartsen, M. Naghshineh, J. Inouye, O. Joeressen, and W. Allen: Bluetooth: Vision, goals, and architecture. *Mobile Computing and Communications Review*, vol. 2, no. 4, (1998) 38–45
- [4] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister: System architecture directions for networked sensors. 9th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, (2000) 93–104.
- [5] C. Intanagonwiwat, R. Govindan, and D. Estrin: Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. *Proc. of ACM MOBICOM*, (2000)
- [6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss: An Architecture for Differentiated Services. RFC 2475, IETF, (1998)
- [7] S. Bhatnagar, B. Deb, and B. Nath: Service Differentiation in Sensor Networks. In *Proc. of Wireless Personal Multimedia Communications*, (2001)
- [8] I. Stoica, S. Shenker, and H. Zhang: Core-Stateless Fair Queuing: Achieving Approximately Fair Bandwidth Allocations in High Speed Networks. *Proc. of ACM SIGCOM*, (1998)
- [9] I. Stoica, and H. Zhang: Providing Guaranteed Services Without Per-flow Management. *Proc. of ACM SIGCOMM*, (1999)
- [10] B. Krishnamachari, S. Wicker, and R. Bejar: Phase Transition Phenomenon in Wireless Ad-Hoc Networks. In *Symposium of Ad-Hoc Wireless Networks, Globecom* (2001)
- [11] K. Sohrabi, G. J. Pottie: Performance of a novel self-organization protocol for wireless ad hoc sensor networks. *IEEE Vehicular Technology Conference*, Volume 2, (1999)
- [12] D. Ganesan, R. Govindan, S. Shenker and D. Estrin Highly Resilient, Energy Efficient Multipath Routing in Wireless Sensor Networks. *Mobile Computing and Communications Review (MC2R) Volume 1*, (2002)
- [13] M. Marina and S. Das On Demand Multipath Distance Vector Routing in Ad Hoc Networks *IEEE International Conference on Network Protocols (ICNP)*, (2001)
- [14] A. Nasipuri and S.R. Das On Demand Multipath Routing for Mobile Ad Hoc Networks 8th International Conference on Computer Communications and Networks (IC3N). Boston (1999)
- [15] M.R. Pearlman, Z.J. Haas, P. Sholander, and S.S. Tabrizi On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks. *Proceedings of IEEE/ACM Mobihoc 2000*, Boston Aug. (2000)
- [16] On Demand Multipath Routing for Mobile Ad Hoc Networks 4th European Personal Mobile Communication Conference (EPMCC), Vienna (2001)
- [17] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan Priority Encoded Transmission *IEEE Transactions on Information Theory* , (1996)
- [18] A.E. Mohr, E.A. Riskin, and R.E. Ladner Graceful Degradation over Packet Erasure channels through forward error correction *Proc. Data Compression Conference*, Snowbird , UT, (1999)