A Framework for Opportunistic Routing in Multi-hop Wireless Networks*

Niki Gazoni
ICS-FORTH
P.O. Box 1385, 71110
Heraklion, Greece
ngazoni@ics.forth.gr

Vangelis Angelakis†
ITN Linköping University
SE-60174
Norrköping, Sweden
vanan@itn.liu.se

Vasilios A. Siris‡
ICS-FORTH
P.O. Box 1385, 71110
Heraklion, Greece
vsiris@ics.forth.gr

Raffaele Bruno
IIT - CNR
Via G. Moruzzi 1 - 56124
Pisa, Italy
r.bruno@iit.cnr.it

ABSTRACT

Opportunistic routing has recently been proposed to take advantage of the broadcast nature and spatial diversity of the wireless medium and cope with unreliable transmissions. Within this "opportunistic routing hype", complex routing schemes with opportunistic features have been proposed, but their performance gains can not be clearly attributed to their opportunistic character, since they also include other strong optimization features applicable to classical routing. A goal of this work is to study how purely wireless primitives and design characteristics affect a routing scheme with opportunistic features and thus design a new such scheme. To this end we introduced a simple framework under which, through simulation, we defined the key elements of an adaptive probabilistic forwarding scheme. We show that it outperforms the opportunistic elements of two well-known opportunistic routing protocols: SOAR and Directed Transmission, in terms of delay and resource utilization, under varying channel error and misinformation conditions and due to its simplicity the gains can be clearly attributed to its core features.

Categories and Subject Descriptors

C.2.2 [**Computer**]: Communication Networks—*network protocols, routing protocols*

General Terms

Algorithms, Measurement, Performance

Keywords

Probabilistic routing, Simulation

1. INTRODUCTION

Routing in multi-hop wireless networks poses a challenge due to the unreliability of wireless links and interference among concurrent transmissions. Subsequently, traditional routing schemes that select a best path and forward the packet to a specific next hop, have proven ill-suited for wireless networks with lossy broadcast links. Recently, a new routing paradigm, opportunistic routing, is proposed to cope with unreliable transmissions by taking advantage of the broadcast nature and spatial diversity of the wireless medium. Leveraging the nodes' ability to overhear a broadcast packet, it differs from traditional routing in that forwarders are selected among the packet recipients after its transmission, hence not committing to a predetermined path. This characteristic enables opportunistic routing to combine multiple weak links to create a reliable one, as well as to exploit unexpectedly long transmissions. The increase of forwarding reliability reduces the retransmission cost, which in turn improves throughput and energy efficiency.

Previously suggested opportunistic protocols demonstrate a lack of concrete understanding of the way key wireless networking primitives and design decisions affect the performance of an opportunistic routing scheme. As a result, it is unclear to which extent the improved performance of these protocols owes to their opportunistic design and to which extent it is affected by other features that can also be applied to traditional routing. Opportunistic protocols which decide on forwarders in a centralized manner, require the exchange of node coordination messages, leading to high overhead and increased resource consumption. Furthermore they require global knowledge of the topology, making them prone to poor performance in the event of misinformation. On the other hand, localized forwarding decision protocols, designed mostly for use in sensor networks, have to trade high performance for robustness and simplicity.

*This work was partly supported by the EC in the 7th Framework Program through project EU-MESH (Enhanced, Ubiquitous, and Dependable Broadband Access using MESH Networks), ICT-215320, `http://www.eu-mesh.eu`

†During this work V. Angelakis was with ICS-FORTH

‡V. A. Siris is also with Athens University of Economics and Business

PE-WASUN'10, October 17–18, 2010, Bodrum, Turkey.

With the aid of a simulation platform, we investigated how forwarding decisions and transmission timing affect performance and under which channel error conditions and topology density it is beneficial to use opportunistic routing instead of traditional routing. With the insight gained, we designed a probabilistic procedure which can be tuned to allow for low resource consumption and high delay performance, while being robust to misinformation. In order to evaluate the suggested procedure, it is compared to single path routing and two distinguished opportunistic routing protocols. Simulation results under various channel error and misinformation conditions, demonstrate that the suggested scheme can outperform both SOAR, which uses a centralized forwarding decision scheme and Directed Transmission, which is localized and designed for sensor networks.

The remainder of this paper is organized as follows. Section 2 reviews related work and states the purpose of this work. The procedure's design elements are presented in Section 3, followed by a presentation of the simulation setup which was used to evaluate the procedure and its results, in Section 4. Finally, concluding remarks are presented in Section 5.

2. RELATED WORK

In ExOR [2], a forwarding list is generated by the source, for a batch of packets. Candidate forwarders are listed according to their ETX cost to reach the destination. Due to the centralized coordination and scheduling that is needed between forwarders and the destination, ExOR incurs high overhead when the batch of packets to transmit is small as in bursty and short-lived flows, or the number of candidate forwarders is large. SOAR [7] is proposed as an improvement to ExOR in order to support multiple flows. It constrains forwarders to be near or along the shortest path, to overhear each other, in order to avoid duplicate transmissions. SOAR uses a complex acknowledgement scheme and ACK compression to reduce its overhead. It also incorporates a data rate control scheme [8], along which the destination sends cumulative ACKS to the source, used to adjust the source's data rate, similarly to TCP's congestion window. However, there is no discrimination between loss due to degrading link quality and loss due to congestion, which may increase retransmission delay. In [11], it is emphasized that the throughput gain achieved by opportunistic routing protocols is not clearly attributed to the opportunistic selection of next hops but also partly due to their acknowledgement and scheduling features which can also be implemented by traditional routing. It is also demonstrated that using ETX can perform poorly when node coordination is imperfect. Moreover, sufficient network density is proven to be required in order to have significant gain by using opportunistic routing.

OMR [4] uses probabilistic forwarding in addition to coordination schemes similar to ExOR's, with the difference that the receiver decides on the transmission probability rather than the sender, based on link quality and feedback from neighboring nodes. This change is due to the fact that OMR has been designed for WSNs; however, adopting ExOR's coordinated approach results in substantial overhead. ROMER [9] uses a credit mechanism to build a forwarding mesh centered on the minimum cost path. It favors high quality links over poor ones by selecting higher random forwarding probability. However, it does not account for resource conservation and catering to multiple flows. The two probabilistic protocols of [1] forward a single packet with varying retransmission probability through a network of sensors, focusing on simplicity and robustness to error in distance estimation. Directed Transmission [1] improves performance by favoring nodes that are on the shortest path, yielding lower resource consumption. None of these two accounts for packet loss due to low link quality, or even multiple packets, let alone flows.

Summarily, probabilistic protocols, primarily designed for WSNs, focus on calculational simplicity and robustness, at the expense of delay performance and accounting for multiple flows [6]. Pure probabilistic forwarding is prone to percolation, leading to either network flooding by redundant copies of the same packet, or too few nodes receiving it [5]. On the other hand, centralized deterministic schemes, designed for mesh networks with static nodes, incur high overhead and are vulnerable to metric misinformation due to use of forwarding lists, which propagate such errors. Thus, resource conservation suffers in exchange for high performance. We propose an adjustable scheme which can be tuned to provide high delay performance or minimized resource consumption. It utilizes probabilistic forwarding with the addition of certain forwarders, enabling nodes to decide forwarding locally with minimal overhead, while ensuring packet progress. Moreover, the proposed scheme is robust to channel error and metric miscalculation.

3. SCHEME DESIGN

The overall protocol design is based on each node locally calculating a value for each arriving packet, according to a cost function, and assigning to it a forwarding probability and deferral duration. Better forwarding candidates are favored by higher forwarding probability and lower deferral durations. This differentiation of forwarding time between candidate forwarders reduces collisions and redundant transmissions. Passive hop by hop acknowledgments are assumed by overhearing neighboring broadcasts, to avoid additional network load and mitigate delay over lossy links.

Apart from an initial neighbor discovery phase, forwarding decisions require minimal further explicit information exchange. This is done in order to minimize protocol overhead in terms of computational load and bandwidth. Thus, candidate forwarders are self-volunteered in order to improve the performance of the end-to-end flow. Any cost metric can be used by this scheme, provided that its values can be bounded tightly. For the implementation presented in this paper, as a proof of concept we used the Euclidian distance as our forwarding metric aiming for simplicity, since we focus on the opportunistic design elements. Hence, the nodes closest to the destination of a packet are considered the best forwarding candidates for it and have the lowest metric value.

3.1 Opportunistic elements

3.1.1 Forwarding probability function

This function is common between all nodes in the network and is used to assign a probability to each node according to the value of their cost metric. More specifically, this is a decreasing function, meaning that higher probability is assigned to nodes with lower routing cost values, and it is bound, so that the minimum probability a node can assign

is 0 and the maximum is 1. For this paper the forwarding probability is assigned with respect to the distance of the receiving node from the packet destination and its relative position to the sender. Hence, when a node broadcasts a packet, it has to include its distance from the destination in its header, so that the nodes in range that receive this, can calculate their forwarding probability for it. Obviously, out of the nodes within range of its broadcast, the neighbor closest to the destination will be assigned a probability equal to 1, to ensure the progress of the packet towards the destination.

Linear forwarding probability function: Initially, a linear decreasing function with probability values from 1 to 0 would satisfy the above requirements. Formally, this forwarding probability p is expressed by:

$$p = \frac{1}{d_{min} - d_{max}} [d \cdot (P_{max} - P_{min}) + P_{min} \cdot d_{min} - P_{max} \cdot d_{max}] \quad (1)$$

where d is the distance between the candidate forwarder and the destination, P_{max} is the forwarding probability associated to the nearest possible candidate (i.e., for $d = d_{min}$), and P_{min} is the probability associated to the furthest possible candidate (i.e., for $d = d_{max}$). It is straightforward to observe that $0 \leq p \leq 1$ for $d_{min} \leq d \leq d_{max}$ if and only if $0 \leq P_{max} \leq 1$ and $0 \leq P_{min} \leq 1$.

This function provides differentiation between the forwarding probabilities of nodes that receive the same packet in a broadcast area. However, it assigns a probability equal to 1 to at most one node in each opportunistic broadcast area. That will be the node in the broadcast area having d min distance from the destination. Therefore, the packet's progress relies heavily on that particular node. If for some reason that node is not to be found any longer at this particular distance value in a topology, then there would be no "definite forwarder" for that packet, in this broadcast area.

Piece-wise forwarding probability function: To deal with the problem of not having definitive forwarders, the previous forwarding probability function can be modified to increase the number of potential forwarding candidates having probability one to forward the received packet. This is achieved by using a piecewise function composed of an initial flat region saturated to probability one, followed by a linear function. The shape of the function is demonstrated in Figure 1. In this case, the forwarding probability is given by $p = min\{1, P\}$ where P is given by (1).

The piecewise function is produced by introducing $P_{max} > 1$, hence introducing more than one certain forwarders. To ensure the packet's progressing to the destination, at least one neighbor with forwarding probability equal to 1 is needed and this is ensured by the flat region. Ideally this would be the neighbor on the shortest path to destination.

Step-wise forwarding probability function: Another variant of the forwarding probability function is the stepwise function. In this case, nodes are either assigned a forwarding probability equal to 1 or a probability equal to zero, that is they either forward the packet always or they never do. There is a threshold value of the metric, which when surpassed it is decided that the node should not forward. As previously, the number of definite forwarders can be increased or decreased by changing the position of the step threshold. The formula describing this function would be

$$p = \begin{cases} 0 & d_{threshold} < d \leq d_{max} \\ 1 & d_{min} \leq d \leq d_{threshold} \end{cases}, \quad (2)$$

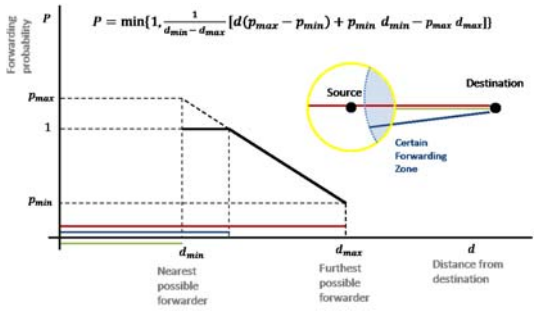


Figure 1: Piece-wise forwarding probability function

3.1.2 *Deferral window and back-off differentiation*

Having calculated the probability to forward a received packet, the node proceeds to decide when to do so. To this end, a randomized back-off mechanism is used, where each node calculates a deferral window and randomly selects a packet back-off value from that range. When this back-off timer expires, the node will proceed to forward the packet with the probability it has been previously assigned. The key is to differentiate the back-off times of different nodes, especially the ones with high forwarding probability to avoid collisions. These definite and high probability forwarders contribute much to the packet's progress towards the destination, therefore it is important to avoid collisions between their transmissions, so that the packet can actually move closer to the destination while maintaining low delay. It is therefore natural to use back-off values inversely proportional to a packet's forwarding probability.

However, in the cases of the piece-wise and the step forwarding probability, both include flat regions, hence they can assign equal probabilities to more than one nodes receiving a packet. That would result in those nodes calculating the same back-off timers for the same packet, resulting in a potentially corrupt reception. For this purpose, the linear probability function (1) is used to calculate the denominator "base probability" for each node which is then used in order to calculate its back-off window, regardless of the actual probability function used. So, the back-off window is calculated as follows:

$$win = (win_{max} - win_{min}) \cdot (1 - p_{base}) + win_{min} \quad (3)$$

Where p_{base} is the base probability for that node and win_{min}, win_{max} the minimum and maximum back-off windows that can be assigned, respectively.

3.2 Secondary elements

3.2.1 *Passive acknowledgment scheme*

In order to identify successful packet reception, the procedure takes advantage of the broadcast nature of the wireless medium. After broadcasting a packet, a node can learn if at least one of its neighbor received it by overhearing their transmissions for a sort amount of time, dependent on the topology and expected forwarding deferral. If a transmission of the last sent packet is captured, then the initiating node will drop this packet from its queue and continue to transmit the next one. In case the specified amount of time goes by without overhearing any transmission of the last sent packet, the node will retransmit it immediately, as long

as the maximum number of retransmissions has not been reached. While overhearing for neighbor transmissions, a node will be on receiver mode, so it will be unable to send.

3.2.2 *Multiple packet flow handling*

Upon successful reception of a new non-expired packet, the node has to calculate a forwarding probability and deferral window for it and select a random back-off value from the range of the latter. A windowed list with the packet and flow IDs of previously successfully forwarded packets must be kept to ensure that a node will not forward the same packet received from different sources twice. After determining all of the above, a node will store individual packets according to the back-off timer that it has calculated for each of them and try to transmit them in time.

The manner in which the node will handle the various packets it has stored can be described as a system of multiple queues, each one containing packets for which the node has selected the same back-off value. Each queue is organized in FIFO fashion. After successfully transmitting a packet and receiving its re-transmission by a neighboring node, the node will look for the queue with the smallest back-off and pick the head packet from it. This ensures that packets follow the back-off priority scheme above. Taking into account that the back-off assignment favors the most likely forwarding candidates for a packet, this queueing policy allows the node to prioritize packets with higher transmission probability and therefore more impact on the end-to-end goal.

3.3 Adjustable parameters

Maximum probability (p_{max}): The forwarding probability function slope defines how great the difference in the probability to forward the packet will be between the neighbors of the node currently holding the packet. Specifically, the steeper that slope is the more the neighbors closer to the destination will be favored. By setting the maximum probability to a value higher than 1, the slope of the forwarding probability function can be tilted, thus increasing the flat segment, which leads to more "certain" forwarders. This feature provides the forwarding function the ability to adapt in situations where more forwarders with high probability are needed and so it can be tuned online, when for example communication conditions become worse due to losses.

Acknowledgment delay : After a node broadcasts a packet, it will start overhearing its neighbors' transmission in order to verify that the packet it sent has been broadcasted by one of them. The amount of time it can wait in this overhearing mode without success, until it decides it has to retransmit the packet, is called acknowledgment delay. If this interval is too small, then the node might end up retransmitting a packet that is successfully received by the further hops, thus adding one redundant transmission. On the other hand, if it is too long and no transmission is overheard, then the packet's progress will be delayed.

Time to live (TTL): To ensure that the packets will not circulate in the network long after they have reached the destination, a mechanism to remove them from play is needed. For this purpose, each packet has a fixed number of credits which are spent each time it is broadcast. These credits can be time units or number of hops traversed, under the assumption that a time unit equals the time it takes for the packet to move one hope further. A node that receives a packet with an expired TTL will discard it without calculating any forwarding probability or back-off window.

4. EVALUATION

4.1 Simulation model

To simulate the events and conditions that occur when packets are being broadcasted in a multi-hop wireless network, some assumptions and simplifications had to be made, regarding packet propagation and channel errors. Firstly, the simulation was conducted in a Matlab time slotted model. In the model, a timeslot refers to the time it takes for a fixed size packet to be broadcast and received. For simulation simplicity, we made the assumption of a geometric propagation model. A transmission from a source is received by all nodes within a broadcast range, with an error probability for each receiver. Concurrent transmissions occurring within the broadcast range of a node are considered to be in collision at that node and neither can be decoded. All nodes in the network have the same error probability and same broadcast radius. The nodes that lie within a node's broadcast radius are considered to be the neighbors of that node. The simulated scenarios take place in a grid topology, such that a node may have four, eight or twelve neighbors, depending on the transmission radius. The nodes have fixed known positions as far as calculating metric values is concerned. More randomized topologies are studied as well, by removing nodes at random positions from the grid.

4.2 Simulation setup

In order to model the behavior of the proposed opportunistic routing scheme and optimize its parameters, we used a MATLAB-based platform as a time-driven simulator. Building from scratch allowed us to control simulation parameters at low levels fully, so as to provide fair comparison between protocols. This ensures that there are no additional model assumptions imposed by the simulator, as is the case with the popular NS-2. Experiments were conducted on a 40x40 node grid topology in order to measure delay and resource consumption for varying network density and channel error conditions. Each experiment with a given set of parameters is repeated for 500 or 1000 runs and the results are averaged over the number of runs. Delay measurements were performed on the shortest path from source to destination and single path routing has been simulated and used as the comparison reference.

Single path routing model: Single path routing is modelled as a stochastic procedure of packet forwarding, rather than simulating nodes. Delay is increased by one timeslot each time a packet manages to progress by one hop, same as each time an acknowledgment is sent, until all ten hops have been traversed. A packet's retransmission also costs one timeslot and the same TTL as in the proposed opportunistic scheme is applied.

Modelling miscalculations in the metric value: For the purpose of testing the performance of the suggested procedure in more realistic conditions, the assumption that there is full and accurate knowledge of nodes' locations in the topology should be relaxed. Regardless of the metric used, the possibility of wrong estimations of the metric value should be taken into consideration when testing the performance of a routing scheme. To this end, "noise" in metric calculation was introduced in the simulation environment.

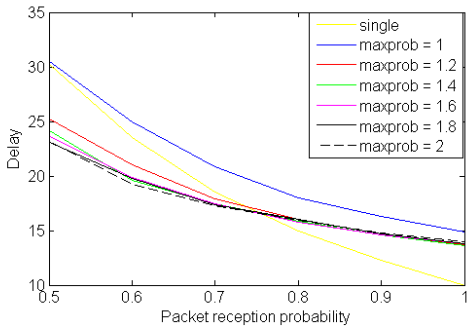


Figure 2: Tilting the slope leads to lower delays for 8 neighbor topologies.

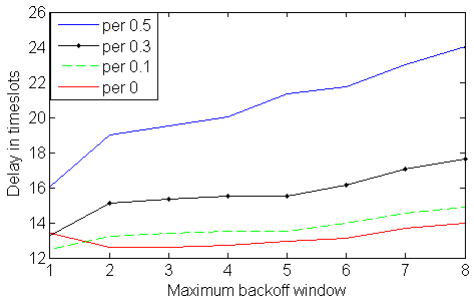


Figure 3: Delay for variable back-off schemes in 8-neighbor topology.

"Noise" in metric calculations was modelled as a randomly selected value from a uniform distribution, with mean equal to the accurately calculated value of the metric. The width of the interval from where values are chosen is equal to a percentage of the correct value of the metric.

Modelling random topologies: The limitations of the grid topology can be relaxed, by introducing "dead" nodes to random positions in the grid. This is implemented by randomly selecting nodes other than the source or destination of the packets and setting their correct packet reception probability to zero. Hence these nodes are not involved in the packet forwarding process and act as obstacles to the packet's progress.

4.3 Adjustable parameter tuning

4.3.1 Increasing forwarders to adapt to error

By setting the maximum probability to a value higher than 1, we tilt the slope of the forwarding probability function, thus increasing the number of certain forwarders. It is reasonable that values of maximum probability in the range [1.4,2] contribute to the packet's progress without diverging much from the shortest path. Furthermore, for a given packet error rate (PER) value, having more than one certain forwarders yields lower delay.

This is verified by Figure 2 which illustrates how tilting the forwarding probability function's slope leads to lower delays, in the context of a 10 hop shortest path. It should be noted that only the maximum forwarding probability parameter is examined at this point; lower delay can be achieved by adjusting the maximum back-off window value as well, which in this case is variable, dependent on the forwarding probability, taking values in [1,8]. Nonetheless, our probabilistic scheme outperforms single-path routing for PER values higher than 0.25. In fact, this effect can be observed regardless of topology density, since the same trend was observed in the denser 12 neighbor topology.

4.3.2 Back-off differentiation

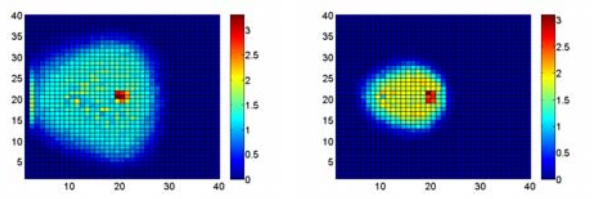
There were two general approaches to deferral window schemes with respect to the window range values. In the fixed back-off window approach, all transmitting nodes randomly select their back-off values from the same range of window values. On the other hand, when differentiating, each transmitting node randomly selects its back-off value from a different range of numbers. Specifically, the back-off window of a node will be randomly chosen between a smaller set of numbers, the larger its forwarding probability is, so as to reduce delays. Figure 3 illustrates the delay performance of the differentiated back-off window scheme, for increasing packet error rate values, in a scenario where the source is 10 hops away from the destination. There was a steeper increase in delay for the fixed back-off window scheme as the width of the back-off window interval increases, which renders fixed back-off values larger than 2 inefficient.

It can be observed that the lowest delay is measured for a back-off window of 1, which raises the question, why differentiate between nodes at all. The reason is that back-off differentiation also yields lower resource consumption.

In order to capture the effect a packet's transmission has on the network, we track the footprint its transmissions produce over time on the nodes as it is forwarded towards the destination, until all transmissions cease. The times each node has received the packet are averaged over the number of the different runs of the experiment and displayed by the color of that node. For all results referring to footprints hereon the source node's coordinates are (10,20) and the destination is at (20,20). Figure 4(a) illustrates the footprint for a flow with back-off window equal to 1. As we increase the width of the deferral window the flooding is limited to an area around the shortest path. This is shown by the plot in Figure 4(b). There is a trade-off between low delay performance and resource consumption which should be addressed by having each flow's specific requirements in mind. For example, in a network where only one flow is present at a time, a deferral window set to 1 would yield the lowest delay possible, whereas in the presence of multiple flows, a more conservative back-off scheme with the window interval set to [1,8] should be used.

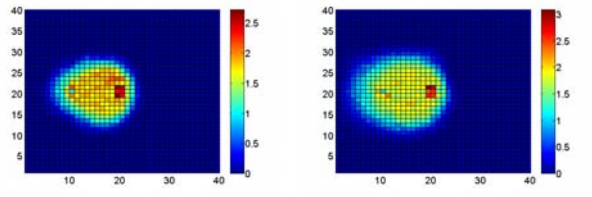
4.3.3 Robustness to metric miscalculation

In order to evaluate the robustness of our scheme to mistakes in metric estimations, we introduced the concept of metric noise. To measure resource consumption for the linear probability function with maximum probability set to 2, we track the flow's course in a scenario where there is no channel error and the source is 10 hops away from the destination and use a balanced back-off window scheme in the interval [1, 4]. Figure 5 illustrates how the linear forwarding probability scheme reacts to increasing noise. It is noteworthy that the linear forwarding probability scheme is insensitive to metric noise up to 10% of the accurate metric's value and performs decently even at the presence of noise equal to 30% of the accurate metric's value. Having



(a) (b)

Figure 4: A deferral window of size 1 will yield slightly lower delay than a variable one, but significantly higher resource consumption. Increasing the deferral window interval to [1,8] leads to lower resource consumption.



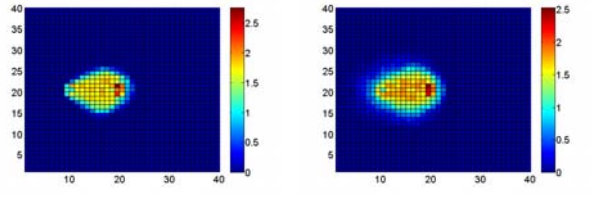
(a) (b)

Figure 5: Increasing the error in metric value estimation from 0.05 to 0.3 affects resource consumption slightly in an 8 neighbor topology, for the piece-wise probability function.

in mind that the metric values are randomly chosen from a uniform distribution in so wide intervals, it is obvious that metric noise equal to 0.3 presents a scenario of extremely inaccurate metric estimation.

We also compared the performance of the linear probability function in the presence of noise to that of the step-wise probability function.

Figure 6 illustrates the step-wise function's performance under increasing metric noise. For the same values of metric noise, the step-wise function consumes fewer resources than the linear function and is also less affected by metric noise. This is justified by the fact that even when inaccurate estimations are made, only a few nodes will calculate their metric so that they fall below the threshold or go above it, whereas the majority of nodes will get the same probability as what they would get if there was no noise. Furthermore, we measured the packet delay over the same 10 hop path



(a) (b)

Figure 6: Increasing the error in metric value estimation from 0.05 to 0.3 affects resource consumption even less, for the step-wise probability function.

for both the linear and the step-wise probability function, for various packet error rate values and observed that metric noise does not affect delay performance significantly for neither the linear nor the step-wise probability function.

4.4 Relaxing the grid assumption

In order to verify that the suggested scheme can function in more loose topologies, random topologies were generated according to the model explained in section 4.2. A random dense topology is generated by giving nodes in the grid a 5% probability to be "dead", whereas a random sparse topology is generated when there is a 10% probability that a node is "dead". Both the piece-wise forwarding probability function and the step-wise probability function were tested. The packet delay was once again measured over a 10 hop path and averaged over 1000 runs.

Tables 1 and 2 demonstrate the delay performance of the two forwarding functions in random topologies that include "dead" nodes in various positions in the grid. Delay is slightly increased compared to what was experienced in Section 4.3.1 which can be justified by the extra transmissions that are needed to route around the "dead" nodes and the alternative paths followed by the packet. It can be noted that slightly higher delay is noted when the step-wise forwarding function is used. This can be attributed to the hard-decision that this function uses to determine which nodes will certainly forward and which will not forward at all. Such a rigid decision scheme can leave out further nodes which could add to the packet's progress, and include some closer nodes that are however "dead". Contrary to this, the smoother probability assignment of the linear piece-wise function, may be more beneficial in such situations.

4.5 Multiple packets and flows

The packet-based simulation model for the suggested opportunistic scheme was extended, so as to include the concept of flows consisting of a number of packets. To this end, a module that generates packets at a configurable rate was implemented. Moreover, the nodes were supplied with queues that operate as explained in 3.2.2. These new elements added complexity thus making simulations more time consuming and adding large variation to the results, which in turn raised the need for an increase in the number of runs per scenario. Due to these circumstances, it was necessary to decrease the number of hops between the source and destination, from 10 to 5.

The robustness of the opportunistic scheme to channel error is tested by using three different uniform error conditions for the experiments: low channel error (20%), medium channel error (30%) and high channel error (40%). The scenario for the following experiments is one flow consisting of 50 packets transmitted from a source that is 5 hops away from the destination, or two flows of 50 packets each, which start from different sources and end at the same destination which is 5 hops away from both sources. The flows generate packets at a steady rate of 1 packet every 16 timeslots. Packet forwarding is decided according to the step-wise forwarding probability function.

In order to study the behavior of our scheme in the presence of multiple packets which interact with one another, a flow consisting of 50 packets is established between a source that is 5 hops away from the destination. Figure 7 demonstrates how the scheme copes with increasing packet error

Table 1: Random dense topology

PER	step-wise delay	piece-wise delay
20%	16.402545	16.312124
30%	18.479613	17.932832
40%	21.399183	20.484211

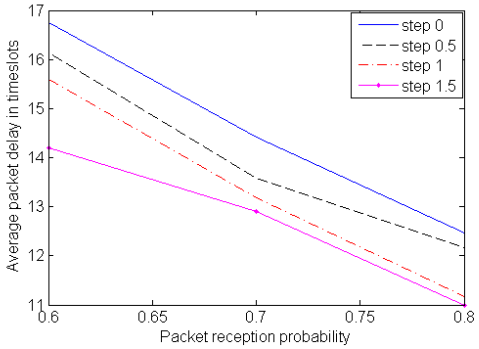


Figure 7: Delay of a 50 packet flow using step-wise probability function to adapt to increasing packet error probability

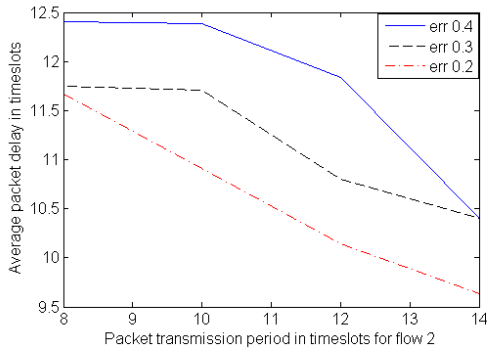


Figure 8: Delay of a 50 packet flow while competing with another flow with the same destination

caused by lossy links. It should be noted that the trend that was observed in the single packet scenario, is also present here. As the step of the forwarding probability function increases, so do the forwarders and the average packet delay is reduced due to this effect. In particular, in high error conditions, the flow benefits more from this increase, similarly to the single packet scenario presented in Section 4.3.1. In addition to verifying that the proposed scheme supports the existence of flows, we examined how two flows interact in the network and confirm that our scheme can indeed support multiple flows. The scenarios were chosen with respect to the most common cases present in wireless mesh and sensor networks. Figure 8 represents the scenario of two nodes sending packets to the same "sink" node.

4.6 Comparison to other schemes

In order to evaluate our scheme, it is compared to SOAR and Directed Transmission, which are some of the best performing opportunistic protocols, designed for mesh and wireless sensor networks respectively. SOAR is a high overhead protocol which focuses on minimizing delay at the expense of resource usage, whereas Directed Transmission is a probabilistic protocol designed for WSNs that focuses on robustness and simplicity at the expense of performance. The purpose of these comparisons is to demonstrate that a tunable scheme can cater to the demands of both network types, without suffering from the drawbacks of these two schemes.

Table 2: Random sparse topology

PER	step-wise delay	piece-wise delay
20%	16.819345	16.775213
30%	19.102419	18.420050
40%	22.439709	21.428213

4.6.1 Comparison to Directed Transmission

Directed Transmission is a parametric probabilistic routing protocol suggested in [1], which focuses on design simplicity, distributed routing decisions and robustness to metric miscalculation. When a node receives a packet it can be retransmitted with probability given by:

$$P(Ri) = \exp\{k[d(S,D) - d(R_i,D) - i]\}, \quad (4)$$

Where *d(S, D)* is the distance between source and destination, i represents a number of time steps after first packet transmission and k is a tunable parameter which controls the spread of the flow. It should be noted that Directed Transmission does not account for losses due to poor link quality, so the two protocols were compared in no-channel error scenarios, where metric miscalculation was present.

When error in metric calculation increases, directed transmission's use of resources does not increase significantly, as is the case with our scheme. However, the spread of the flow on the grid is comparable to its equivalent in the proposed scheme when the piece-wise probability function was used and larger than its equivalent when the step-wise probability function was used. Furthermore directed transmission has a larger average number of transmissions needed to deliver a packet along the 10 hop path, regardless of metric noise. These results demonstrate that a protocol can be simple enough and conserve resources to be applied in WSNs without suffering from low delay performance.

The performance of both directed transmission and our approach was not affected by increase in metric miscalculation; however our approach yields significantly smaller delays. In summary, as shown in Figure 9 our scheme outperforms Directed Transmission both in terms of resource consumption and delay, under increasing metric miscalculation.

4.6.2 Comparison to SOAR

For our comparative simulations, SOAR's algorithm for packet forwarding decisions was used, combined with the proposed passive acknowledgment scheme, in order to test the performance of its opportunistic features. SOAR initially uses ETX as a metric in order to decide on the cost of forwarding, however, for comparison purposes, hop distance was used for both protocols. SOAR behaves similarly to shortest path routing in no error conditions, constraining the flow along the shortest path from the source to the destination, so our scheme can yield lowers delays than SOAR for packet error probability over 0.25. When metric miscalculation is present, SOAR's delay increases significantly, as

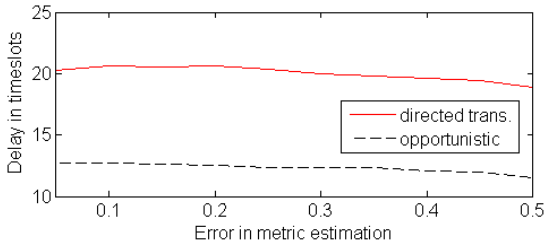
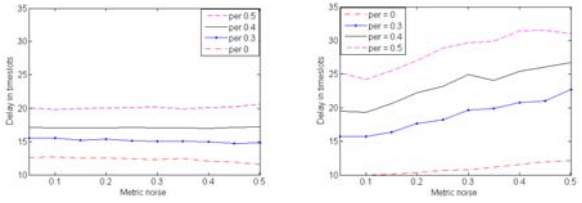


Figure 9: Both Directed Transmission and the proposed opportunistic scheme are robust to metric miscalculation but the latter yields far lower delays.



(a) (b)

Figure 10: Error in metric value estimation does not affect delay over a 10 hop distance significantly when piece-wise probability function is used.

opposed to the proposed scheme's performance which is unaffected, as shown in Figure 10(b). This can be explained by the deterministic forwarding scheme used by SOAR. When metric miscalculation occurs at the source who creates the list of forwarders, this error will propagate along with the list, since it is included in the sent packets. Therefore consequent calculations based on this list will be influenced by this error.

5. CONCLUSIONS

Simulation results demonstrate that the suggested opportunistic scheme can outperform single path routing for error values larger than 15%-20%, for an optimized slope of the forwarding probability function, with restrained use of resources. Furthermore, we showed that the optimal manner of adapting to increasing error is to increase the number of forwarders by increasing the slope of the forwarding probability function. In particular, the number of certain forwarders has the most impact on performance and they need to be increased with respect to error conditions. To reduce resource consumption, in terms of packet transmissions, utilizing a step-wise function is a sound approach, which proved robust to metric miscalculations. Finally, there is a trade-off between differentiating each forwarder's back-off value to reduce resource consumption and reducing delay. Hence, a variable back-off scheme that gives priority, by means of smaller back-off windows, to best forwarders according to their forwarding probability, is preferable to a fixed back-off window for all forwarders. It is also confirmed that the proposed scheme supports multiple flows successfully. Comparison to Directed Transmission and SOAR under metric miscalculation and channel loss conditions, shows that SOAR's use of forwarding lists exacerbates metric miscalculations, while our adjustable probabilistic scheme remains unaffected, at the same time yielding significantly lower delay than Directed Transmission, in all cases.

6. REFERENCES

[1] C. L. Barrett, S. J. Eidenbenz, L. Kroc, M. Marathe, and J. P. Smith. Parametric probabilistic sensor network routing. In *ACM WSNA*, 2007.

[2] S. Biswas and R. Morris. Opportunistic routing in multi-hop wireless networks. In *SIGCOMM*, 2005.

[3] D. C. D., M. R., A. D., and B. J. A high-throughput path metric for multi-hop wireless routing. In *MobiCom*, 2003.

[4] J. Du, H. Liu, and P. Chen. Omr: An opportunistic multi-path reliable routing protocol in wireless sensor networks. In *ICPPW*, 2007.

[5] H. Z. J, H. J. Y, and L. L. Gossip-based ad hoc routing. In *INFOCOM*, pages 1707–1716, 2002.

[6] H. Liu, B. Zhang, M. H., X. Shen, and J. Ma. Opportunistic routing for wireless ad hoc and sensor networks: Present and future directions. *IEEE Communications Magazine*, 47(12), 2009.

[7] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu. Simple opportunistic. routing protocol for wireless mesh networks. In *WiMesh*, pages 48–54, 2006.

[8] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu. Soar: Simple opportunistic adaptive routing protocol for wireless mesh networks. *IEEE Transactions on Mobile Computing*, 8(12), 2009.

[9] Y. Yuan, H. Yang, S. Wong, S. Lu, and W. Arbaugh. Romer: Resilient opportunistic mesh routing for wireless mesh networks. In *WiMesh workshop, IEEE SECON*, pages 146–158, 2005.

[10] K. Zeng, W. Lou, and H. Zhai. On end-to-end throughput of opportunistic routing in multirate and multihop wireless networks. In *IEEE INFOCOM*, pages 99–100, 2008.

[11] Z. Zhong and S. Nelakuditi. On the efficacy of opportunistic routing. In *SECON*, pages 836–843, 2007.