UNIVERSITY OF CRETE
FACULTY OF SCIENCES AND ENGINEERING
COMPUTER SCIENCE DEPARTMENT

A Framework for Opportunistic Routing in Wireless Multi-hop Networks

Niki Gazoni

Master of Science Thesis

Heraklion, February 2010

UNIVERSITY OF CRETE
FACULTY OF SCIENCES AND ENGINEERING
COMPUTER SCIENCE DEPARTMENT

A Framework for Opportunistic Routing in Wireless Multi-hop Networks

A thesis submitted by

Niki Gazoni

in partial fulfillment for the

Master of Science Degree

Author:

Niki Gazoni

Commitee:

Apostolos Traganitis, Professor
Supervisor

Vasilios Siris, Assistant Professor
Athens University of Economics and Business
Advisor

Panagiotis Tsakalides, Associate Professor
Member

Accepted:

Panos Trahanias, Professor
Director of Graduate Studies

February 2010

A Framework for Opportunistic Routing in Wireless Multi-hop Networks

Niki Gazoni

Master of Science Thesis

Computer Science Department
University of Crete

Abstract

Multi-hop wireless networks, such as mobile ad hoc networks (MANETs), wireless sensor networks (WSNs), and wireless mesh networks (WMNs), have been increasingly popular due to their easy deployment at low cost and their broad applications, ranging from last-mile network access to environment monitoring and vehicular networks. Routing in multi-hop wireless networks poses a great challenge mainly due to the existence of unreliable wireless links and interference among concurrent transmissions. Due to these unique characteristics, traditional routing schemes that select the best path towards a destination 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.

However, existing opportunistic protocols can be characterized by lack of concrete understanding of the way design affects the performance of an opportunistic routing scheme and to which extent the latter is owed to its opportunistic elements, or to other features that can also be applied to traditional routing, such as acknowledgment schemes.

In order to study how these primitives affect an opportunistic routing scheme and to facilitate my own scheme design I developed a framework, in which, using simulation, I was provided with the required insight for the design of a new adaptive opportunistic forwarding scheme. Moreover, the suggested scheme is compared to the opportunistic elements of two distinguished opportunistic routing protocols, SOAR and Directed Transmission, under varying channel error and misinformation conditions, outperforming both in terms of delay and resource consumption. Finally, the suggested procedure is extended to support the existence of multiple flows in a network, which is confirmed by simulations.

Supervisor
Apostolos Traganitis
Professor

Computer Science Department
University of Crete

Ένα πλαίσιο για την ευκαιριακή δρομολόγηση σε ασύρματα δίκτυα πολλαπλών ζεύξεων

Νίκη Γκαζώνη

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Τα ασύρματα δίκτυα πολλαπλών ζεύξεων όπως τα κινητά δίκτυα χωρίς υποδομή (MANETs) , τα ασύρματα δίκτυα αισθητήρων (WSNs) και τα ασύρματα δίκτυα πλέγματος (Mesh networks) , έγιναν ευρέως γνωστά χάρις στην εύκολη εγκατάστασή τους με χαμηλό κόστος και στο εύρος των εφαρμογών τους, που ποικίλλουν, από τελευταίο μίλι δικτυακής πρόσβασης μέχρι την παρατήρηση περιβάλλοντος και τα δίκτυα οχημάτων.

Η δρομολόγηση σε αυτά τα δίκτυα αποτελεί πρόκληση, κυρίως λόγω της αναξιοπιστίας των ασυρμάτων ζεύξεων αλλά και των παρεμβολών μεταξύ των ταυτόχρονων εκπομπών. Εξαιτίας αυτών των ιδιαίτερων χαρακτηριστικών, οι παραδοσιακές τεχνικές δρομολόγησης που επιλέγουν ένα βέλτιστο μονοπάτι και προωθούν τα πακέτα σε έναν συγκεκριμένο επόμενο κόμβο, αποδεικνύονται ανεπαρκείς. Τα τελευταία χρόνια η ευκαιριακή δρομολόγηση έχει προταθεί για την αντιμετώπιση των αναξιόπιστων εκπομπών, χάρις στην ικανότητά της να εκμεταλλεύεται την ευρυεκπομπή που είναι βασική ιδιότητα του ασύρματου μέσου.

Παρά το γεγονός αυτό, τα υπάρχοντα πρωτόκολλα ευκαριακής δρομολόγησης διακρίνονται από ελλειπή κατανόηση των συνεπειών που έχει ο σχεδιασμός ενός τέτοιου πρωτοκόλλου στην απόδοση του. Μάλιστα ειναι ασαφές σε ποιο βαθμό η βελτίωση στην απόδοση σε σχέση με τις παραδοσιακές τεχνικές δρομολόγησης, οφείλεται στα ευκαιριακά στοιχεία των πρωτοκόλλων αυτών και κατά πόσο οφείλεται σε επιπρόσθετα στοιχεία τα οποία μπορούν να εφαρμοστούν και στα παραδοσιακά πρωτόκολλα, όπως μέθοδοι επιβεβαιώσεων.

Προκειμένου να μελετηθεί πώς τα βασικά στοιχεία επηρεάζουν μια μέθοδο ευκαριακής δρομολόγησης, ανέπτυξα ένα πλαίσιο όπου με τη βοήθεια προσομείωσης προέκυψαν τα απαραίτητα συμπεράσματα πάνω στα οποία βασίστηκε η σχεδίαση μιας νέας προσαρμόσιμης διαδικασίας ευκαιριακής δρομολόγησης. Στη συνέχεια, η προτεινόμενη διαδικασία συγκρίθηκε, ως προς την καθυστέρηση και την κατανάλωση πόρων, με δυο διακεκριμένα πρωτόκολλα ευκαιριακής δρομολόγησης, τα SOAR και Directed Transmission, υπό διάφορες συνθήκες σφαλμάτων λόγω κακής ποιότητας καναλιού και λάνθασμένων εκτιμήσεων. Τέλος, η προτεινόμενη διαδικασία δρομολόγησης επεκτάθηκε ώστε να υποστηρίζει την συνύπαρξη πολλών ροών στο δίκτυο, γεγονός που επιβεβαιώνεται μέσω προσομειώσεων.

Επόπτης
Απόστολος Τραγανίτης
Καθηγητής

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Acknowledgements

I would like to express my gratitude to my advisor, Vasilios A. Siris, for his valuable advice during our cooperation throughout the past two years. I would also like to thank my supervisor, Apostolos Traganitis, for his support. Lastly, I would like to thank Vangelis Angelakis without whose guidance this Thesis would not be complete.

Contents

List of Figures

List of Tables

Abbreviations

WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network
WSN	Wireless Sensor Network
DSR	Dynamic Source Routing
AODV	Ad-hoc On Demand Distance Vector
DSDV	Destination-Sequenced Distance Vector
LQSR	Link Quality Source Routing
SNR	Signal to Noise Ratio

Chapter 1

Introduction

1.1 Introduction

Routing in multi-hop wireless networks poses a great challenge mainly due to the existence of unreliable wireless links and interference among concurrent transmissions. Due to these unique characteristics, traditional routing schemes that select the best path towards a destination 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, opportunistic routing differs from traditional routing in that forwarders are selected among the recipients of the packet 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 in one transmission reduces the retransmission cost, which in turn improves the 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 design 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 which makes them prone to poor performance in the event of misinformation. On the other hand, localized forwarding decision protocols, that have been designed mostly for use in sensor networks, have to trade high performance for robustness and simplicity. In fact some of them partly rely on flooding techniques, thus demonstrating a percolation behavior, which leads to unnecessary transmissions.

In the design of an opportunistic routing scheme, there are two key design decisions. Firstly, how the receiver of a packet will decide whether to forward or not. Secondly, provided that a receiver has decided to forward a packet, when is the most appropriate time to do so. With the aid of a MATLAB based simulation platform, this thesis investigated how the 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, I designed an opportunistic procedure which can be tuned to allow for low resource consumption and high delay performance, at the same time being robust to misinformation. Moreover, I use simulations to confirm that the suggested protocol can support multiple flows in a network. In order to evaluate the suggested procedure, it is compared to single path routing and two distinguished opportunistic routing protocols, SOAR and Directed Transmission. Simulation results under various channel error and misinformation conditions, demonstrate that the suggested protocol outperforms both SOAR which uses a centralized forwarding decisions scheme and Directed Transmission which is localized and designed for sensor networks.

1.2 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 summarizes the main characteristics of single path, multi-path and opportunistic routing, reviews related work and states the purpose of this thesis. The procedure's design elements are presented in Chapter 3, followed by a presentation of the simulation setup which was used to evaluate the procedure and its results, in Chapter 4. Finally, concluding remarks and directions for future work are presented in Chapter 5.

Chapter 2

Background

2.1 Routing Schemes in multi-hop wireless networks

2.1.1 Single path routing

Initially, the routing techniques that have traditionally been used in wired networks were adopted for use in multi-hop wireless networks as well. These routing protocols would rely on choosing the best sequence of nodes between the source and destination and then forward each packet through that path. However, the performance of single path routing would often prove unsatisfactory due to the fact that, in wireless multi-hop networks link conditions are highly variable and such a path may not always be found. Most of the existing routing protocols, such as DSR [JMB01], AODV [RP00], DSDV [EP94], and LQSR, fall into this category.

2.1.2 Multi-path routing

For the purpose of improving routing performance, multi-path routing takes advantage of the existence of multiple alternative paths between a source and destination pair in multi-hop wireless networks. The multiple paths computed might be overlapped, edge-disjointed or node-disjointed with each other. Data traffic is split into multiple routes to avoid congestion and to use network resources efficiently. Multi-path routing can yield a variety of benefits such as fault tolerance, increased bandwidth, or improved security. It is typically proposed in order to increase the reliability of data transmission or to provide load balancing [MTG04].

2.1.3 Opportunistic routing

Opportunistic routing differs from traditional routing in that it leverages the broadcast nature of wireless medium and defers route selection after packet transmissions. As opposed to multipath routing, it does not commit to any number of paths. Among the nodes that receive the packet, the one closest to the destination is chosen to forward. This can mitigate the effect of unreliable and unpredictable wireless links. There are two major benefits in opportunistic routing. Firstly, opportunistic routing can combine multiple weak links into one strong link. Secondly, opportunistic routing can take advantage of transmissions that are unexpectedly long, thereby achieving high throughput.

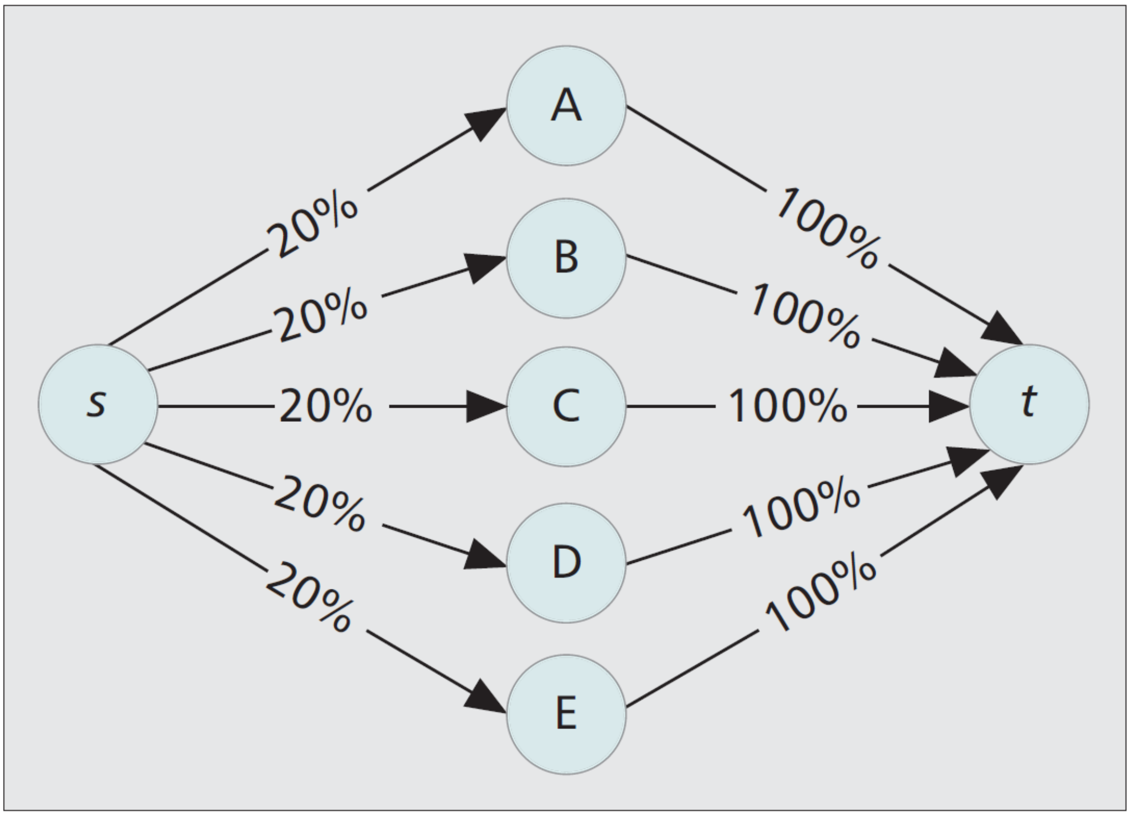


Figure 2.1: Multiple relays combine weak links to a create a reliable link [LZH+09]

2.2 Related work on opportunistic routing

2.2.1 ExOR

In ExOR [BM05], forwarding is decided after the packet has been received by a subset of nodes, according to a forwarding list that is generated by the current transmitter. Nodes need to discover and agree on which nodes of the subset received the packet, and only the node in

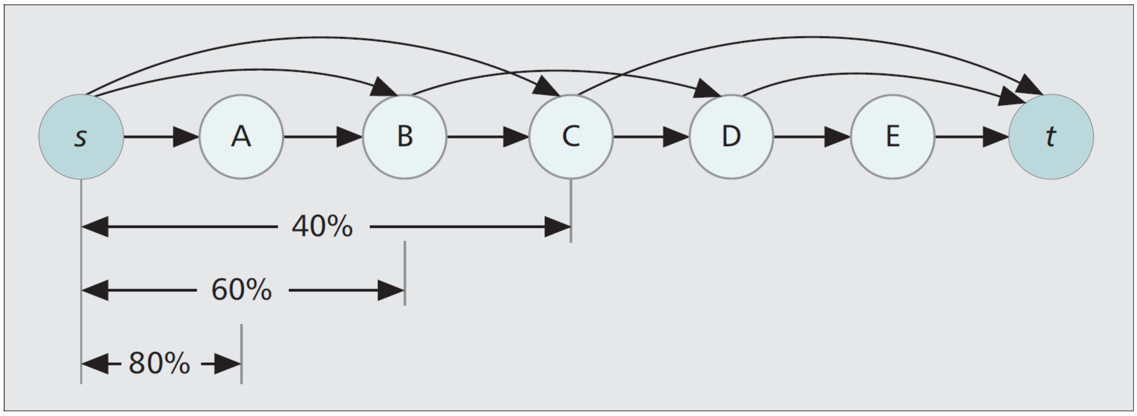


FIGURE 2.2: Leveraging long transmissions in a tandem topology [LZH$^+$09]

that subset that is "closest" to the destination forwards the packet. After collecting a batch of packets, which is uniquely identified by a BatchID, the source broadcasts each packet in the batch, listing the forwarding nodes in priority order in the packet's header. The cost metric is ETX [JRDJ], the expected number of transmissions necessary to forward a packet along a route. It is unlikely that a forwarder receives the entire batch correctly, so the nodes that have stored fragments of the batch will need to schedule their transmissions. To that end, each forwarder uses a forwarding timer set to five packet durations times the number of higher priority nodes in the forwarder list, which is a prediction of the time at which the node should start forwarding packets from its packet buffer. After each schedule cycle, the batch maps need to be updated by means of negative acknowledgments and when the destination has received 90% of the batch, the rest of it is sent using traditional routing, because the overhead would be forbidding otherwise. 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.

2.2.2 SOAR

The SOAR protocol [RSMQ06] is proposed as an improvement to ExOR in order to support multiple flows. In SOAR the candidate forwarders are constrained to be along or near the shortest path from source to destination. The cost metric is again ETX. Another significant difference between ExOR and SOAR is that SOAR performs the routing decision process on a per-packet basis rather than on an entire batch. Finally, in SOAR the forwarder list is limited to 5 relays. Overall, SOAR incurs slightly less overhead than ExOR and restricts flows as close to the shortest path from source to destination as possible. To make the protocol reliable to ACK losses, selective ACKs are used. Basically, each ACK packet contains the starting sequence number of out-of-order ACKs and a bit-map of out-of-order ACKs. In addition, SOAR uses piggyback ACKs and ACK compression to reduce the overhead of acknowledgments. When a node does not have much data to send, it should also send stand-alone ACKs to provide timely feedback. It is considered crucial to avoid diverging paths, and select forwarders that can hear each other because overhearing is the only means to avoid duplicate transmissions. However, the increase in throughput that is observed when SOAR is in use is partly due to its opportunistic scheme and partly due to its complex acknowledgment scheme, therefore making it unclear at what extent this protocol contributes to opportunistic routing.

2.2.3 ROMER

In ROMER [YYW+05], a credit mechanism is used to build a forwarding mesh on the fly centered on the minimum cost path. It aims to avoid the cost of repetitive retransmissions over persistently poor routes. An amount of credit is assigned to each packet that equals to the amount of credit that is needed for the packet to reach the destination along the minimum cost path plus some extra. Each node needs to spend some credit to forward. Before forwarding, it calculates the remaining credit that the packet would have if forwarded over a link and compares it to a threshold value. This scheme makes routes diverge sufficiently close to the source to ensure additional progress but converge and stay close to the minimum cost path, near the destination. Greedy forwarding is used to deliver the data packet along the instantaneously highest rate link with probability one and along other high-rated downstream links with high probability. Thus, it favours high quality links over low quality ones by selecting higher random forwarding probability. However, it does not account for resource conservation and catering to multiple flows.

2.2.4 Destination Attractor and Directed Transmission

The Parametric probabilistic sensor network routing scheme of [BEK+07] proposes two protocols that forward a single packet with varying retransmission probability through a network of sensor nodes, focusing on simplicity and robustness to errors in distance estimation. Destination Attractor [BEK+07] assigns a higher retransmission probability to the packet, as it moves closer to the destination and reduces it, as the packet moves away from the destination. Distance check is performed by comparing the distance of the source node in hops from the destination to the distance of the node currently holding the packet. The primary concern of this approach is to deliver as many copies of a single packet to the destination sensor as possible, without accounting for resource usage and delay. Directed Transmission [BEK+07] improves probabilistic routing's performance by assigning exponentially higher forwarding probability to nodes that are on the shortest path from the source to the destination and decreasing it as the packet strays from the shortest path. This leads to lower resource consumption than Destination Attractor and can be tuned to resemble shortest path routing, when the misinformation is low enough. Both protocols are compared to shortest path routing as an ideal case.

2.2.5 Pure Gossip

One of the simplest probabilistic routing protocols is Pure Gossip [JYL02]. It operates by assigning a retransmission probability to a packet at the source, and having every node to receive the packet retransmitting with that probability. This leads to percolation behavior, in that for a given retransmission probability it is most likely that either very few nodes on the network will receive the packet, or almost all will.

2.2.6 OMR

OMR [DLC07] uses probabilistic forwarding in addition to coordination schemes similar to these of ExOR, with the difference that the receiver is the one to decide 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 to forwarding results in substantial overhead.

2.2.7 Efficacy of opportunistic routing

In [ZN07], it is emphasized that the throughput gain achieved by opportunistic routing protocols is not clearly attributed to the opportunistic selection of next hops but is also partly due to their robust acknowledgement and scheduling features that can also be implemented by traditional routing. Moreover, the authors point out the risk of duplicate forwarding that is experienced in opportunistic schemes, in cases when the multiple forwarders are unaware of others' transmissions, leading to collisions and potentially more retransmissions than single path routing. It is also demonstrated that use of ETX metric by an opportunistic scheme can result to low performance when node coordination is imperfect. Finally, it is noted that sufficient network density is required in order to have significant improvement by using opportunistic routing.

2.3 Statement of purpose and research methodology

Previously suggested opportunistic protocols can be characterized by lack of concrete understanding of the way key wireless networking primitives and design decisions affect the performance of an opportunistic routing scheme. In order to study how these primitives affect an opportunistic routing scheme and to facilitate my own scheme design I developed a framework, in which, using simulation, I was provided with the required insight for the support of the design of a new adaptive opportunistic forwarding scheme.

Chapter 3

Procedure design and elements

3.1 Procedure design

3.1.1 Design principles

The procedure was designed having simplicity and distributed decision- making in mind. Apart from an initial neighbor discovery phase, routing decisions should require minimal information to be exchanged between nodes and no co-operation, to avoid imposing computational load on nodes with limited computational capacity and wasting bandwidth in exchanging control packets when bandwidth is limited. However, the procedure does not achieve this by utilizing flooding, as seen in many sensor network schemes. The reason for this is that the procedure also opts for restricted resource consumption, so as to allow for multiple flows to exist simultaneously in the network. Summarily the procedure is intended as a flexible routing solution that can be applied in a variety of networks, provided that lossy areas are observed due to low link quality and that the topology is sufficiently dense to allow for opportunistic routing.

3.1.2 Design parameters

The major issues that need to be addressed in the design of the routing procedure are deciding which nodes should forward a packet, when to do so, which nodes should acknowledge successful packet reception and how to do so. The candidate forwarders should be selected in order to improve the performance of the end-to-end flow. On the other hand the number of forwarders should be limited to those necessary to ensure the packets' progress towards their destination, in order to avoid excess resource consumption and allow for more than one flows to utilize network resources simultaneously. Furthermore, the differentiation of forwarding times between the candidates may mitigate the generation of redundant transmissions. Lastly, since broadcast packets do not implement link-layer acknowledgements, an acknowledgment scheme should be devised. The destination could generate acknowledgements upon packet reception, which would be propagated back to the source, either opportunistically or via the shortest path. This would impose significant load on the network if the acknowledgements were stand-alone packets, competing with the data packets and being susceptible to lossy links, which would cause significant delay until retransmission of a lost packet. However, the option of piggy-backing the acknowledgments on data packets is limiting since it can be applied only to two-direction flows between source and destination. On the other hand, hop-by-hop acknowledgments provide robustness to the lossy links and mitigate delay. Moreover, a passive acknowledgment scheme that utilizes overhearing of other nodes' transmissions, avoids imposing additional load to the network, contrary to explicit acknowledgment packets.

3.2 Procedure elements

3.2.1 Routing cost metric

For the purpose of determining which nodes are the best forwarding candidates for a particular packet, a routing cost metric needs to be used. The implementation of the procedure presented in this thesis, uses euclidian distance as a metric, for simplicity. However, any cost metric can be used to this end, providing that it has a known minimum and maximum, or such bounds can be tightly estimated. In order for a node to be able to calculate its distance from the destination of the packet, a neighbor discovery phase should take place between the nodes in network, before data packets can be exchanged. Hence, the nodes closest to the destination of a packet are considered the best forwarding candidates for it and have the lowest metric value.

3.2.2 Forwarding probability function

In order to determine which nodes will forward a packet and with what probability will they do so, a forwarding probability function is implemented. 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 function is decreasing, meaning that higher probability is assigned to nodes with lower routing cost values, and it is bound, so that the minimum probability it can assign is 0 and the maximum probability is 1. 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 neighbors of the sender node, that is to say, the nodes that are within range of its broadcast, the neighbor that is closest to the destination should be assigned a probability equal to 1, to ensure the progress of the packet towards the destination.

3.2.2.1 Linear forwarding probability function

Initially, a linear decreasing function with probability values from 1 to 0 would satisfy the above requirements. (Figure 3.1). Formally, this forwarding probability p is expressed by:

$$p = \frac{1}{d_{min} - d_{max}}[d \cdot (P_{max} - P_{min}) + P_{min} \cdot d_{min} - P_{max} \cdot d_{max}] \tag{3.1}$$

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 with differentiation between the forwarding probabilities of different nodes that receive the same packet in a broadcast range. However, it assigns a probability equal to 1 to one node maximum in each broadcast area, the node with d_{min} distance from the destination. Therefore, the packet's progress would relies heavily on that particular node. Moreover, if no node is to be found with this particular distance value in a topology, then there would be no certain forwarder for that packet, in this broadcast area.

3.2.2.2 Piece-wise linear forwarding probability function

To deal with this problem, the previous forwarding probability function was modified to increase the number of potential forwarding candidates having probability one to forward the received packet. This was 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 Fig. 3.2. In this case, the forwarding probability is given by the following formula:

$$p = min\{1, \frac{1}{d_{min} - d_{max}}[d \cdot (P_{max} - P_{min}) + P_{min} \cdot d_{min} - P_{max} \cdot d_{max}]\} \tag{3.2}$$

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

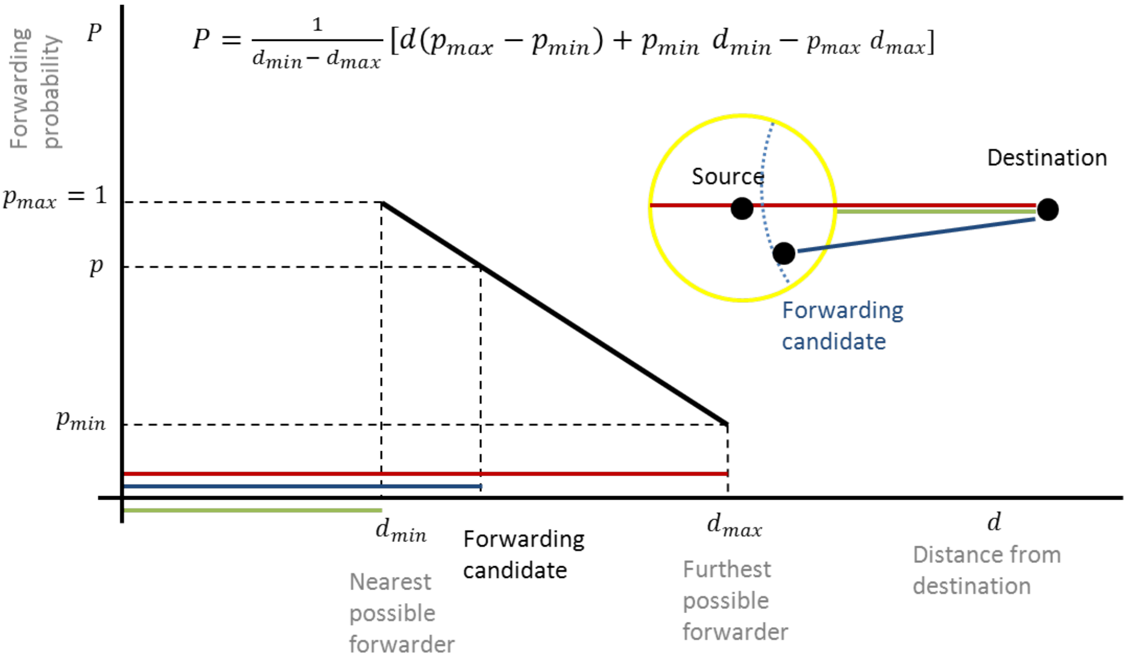


Figure 3.1: Linear forwarding probability function and a node's broadcast radius

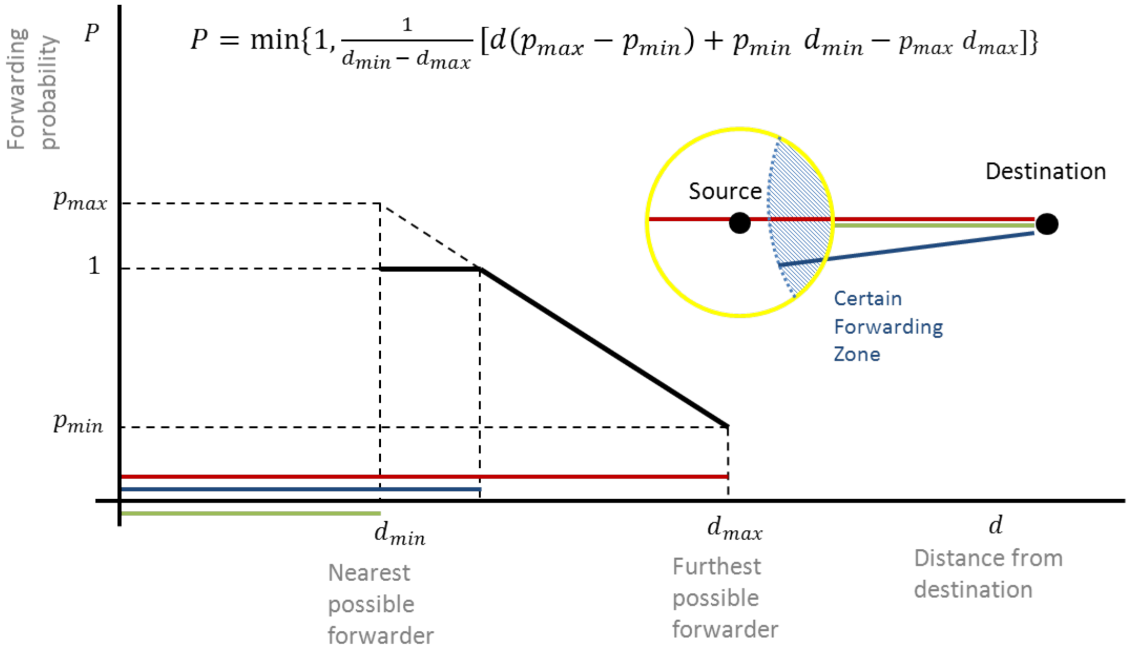


Figure 3.2: Piece-wise forwarding probability function and a node's broadcast radius

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.

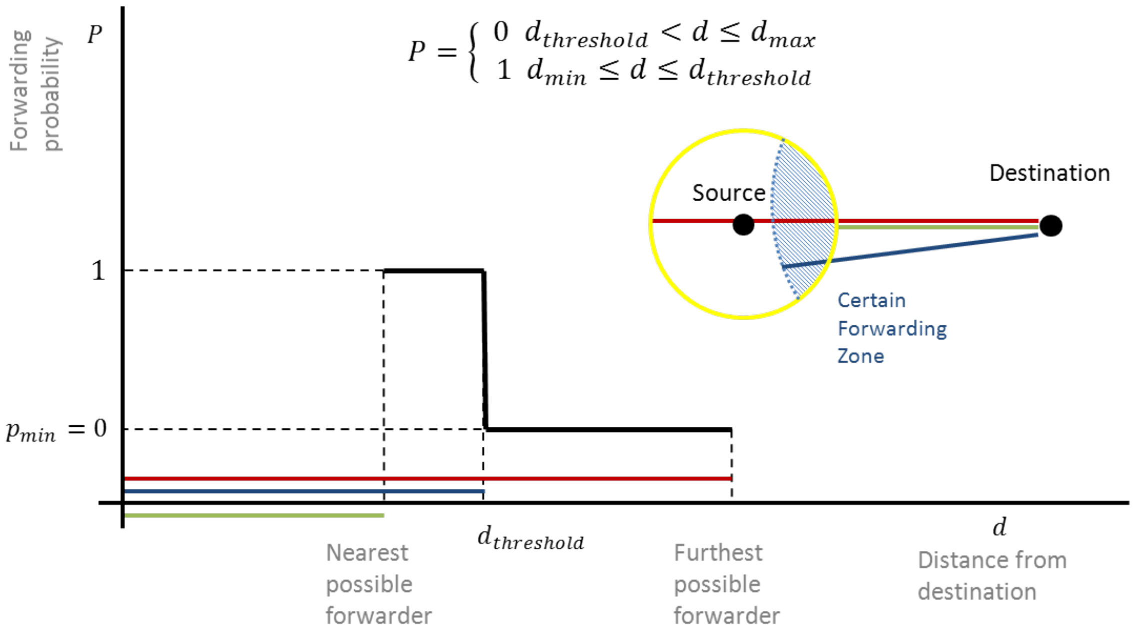


Figure 3.3: Step-wise forwarding probability function and a node's broadcast radius

3.2.2.3 Step-wise forwarding probability function

Another variant of the forwarding probability function is the step-wise function. Figure 3.3 illustrates the step-wise 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. Forwarding is again based on the value of the metric and there is a threshold value of the metric, called step, which when surpassed it is decided that the node should not forward. Yet again, forwarders can be increased or decreased by changing the value of the step threshold. The formula describing this function would be

$$p = \begin{cases} 0 & d_{threshold} < d \leq d_{max} \ , \\ 1 & d_{min} \leq d \leq d_{threshold} \end{cases} \tag{3.3}$$

3.2.3 Back- off window differentiation

Having calculated its probability to forward a certain packet that it received, a node should proceed to decide when to do so. To this end, a randomized back-off mechanism is used, where each node calculates its back-off window and randomly selects a back-off value from that range. After this back-off timer has expired, the node will proceed to forward the packet with its designated probability. The focus here is to differentiate the back-off times of different nodes, especially the ones with high forwarding probability. These optimal forwarders would 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 move closer to the destination as soon as possible. Hence, it is straightforward to conclude that back-off values should be inversely proportional to a node's forwarding probability.

However, the piece-wise and the step forwarding probability functions both include flat regions which can assign probability equal to 1 to more than one nodes which would result in them calculating the same back-off timers. For this purpose, the linear probability function (3.2) is used to calculate a base probability for each node which will then be used in order to calculate its back-off window. Finally, the back-off window is calculated as follows:

$$win = (win_{max} - win_{min}) \cdot (1 - p_{base}) + win_{min} \tag{3.4}$$

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.

3.2.4 Passive hob by hop acknowledgment and retransmission scheme

In order to acknowledge 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 neighboring nodes received it by overhearing its neighbors' transmissions for a sort amount of time. If a transmission of the last sent packet is overheard, then a node will drop this packet from its queue and continue to transmit the next packet. In case the specified amount of time goes by without overhearing any transmission of the last sent packet, then 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.

3.2.5 Multiple packet handling

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

3.2.5.1 Queuing

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 FIFO. After having successfully transmitted a packet and overheard its retransmission by a neighboring node, the node will look for the queue with the smallest back-off and pick the first packet from that. This ensures that a packet for which the node has a small back-off will have priority over one for which the node has a large back-off value. Taking into account that the back-off assignment favors the optimal forwarding candidates for a packet, this queueing policy allows the node to give priority to packets to whose progress it can contribute more.

3.2.6 Adjustable parameters

3.2.6.1 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.

3.2.6.2 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.

3.2.6.3 Time to live (TTL)

To ensure that the packets will not circulate in the network long after they have reached the destination, a mechanism that renders them obsolete 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 for it.

Chapter 4

Evaluation

4.1 Simulation Model

For the purposes of emulating 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. Primarily, it should be noted that the model is time-slotted. Specifically, the time unit is one timeslot, which refers to the time it takes for a packet to be broadcast and received.

4.1.1 Propagation model and reception errors

For simulation simplicity, I have made the assumption of a geometric propagation model. A transmission from a source can only be received by all nodes within a broadcast radius, with a given correct reception probability for each potential receiver. Moreover, all nodes in the network share the same correct packet reception probability and same broadcast radius. The nodes that lie within a node's broadcast radius are hereby referred to as neighbors of that node. The probability of packet loss due to channel errors is modelled as the complement of the correct packet reception probability. Hence, a packet may either be correctly received, or considered as not received at all.

4.1.2 Network topology

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 (Fig. 4.1). The nodes have fixed known positions as far as calculating metric values is concerned. However, more randomized topologies will be studied as well, by introducing "dead" nodes to random positions in the grid.

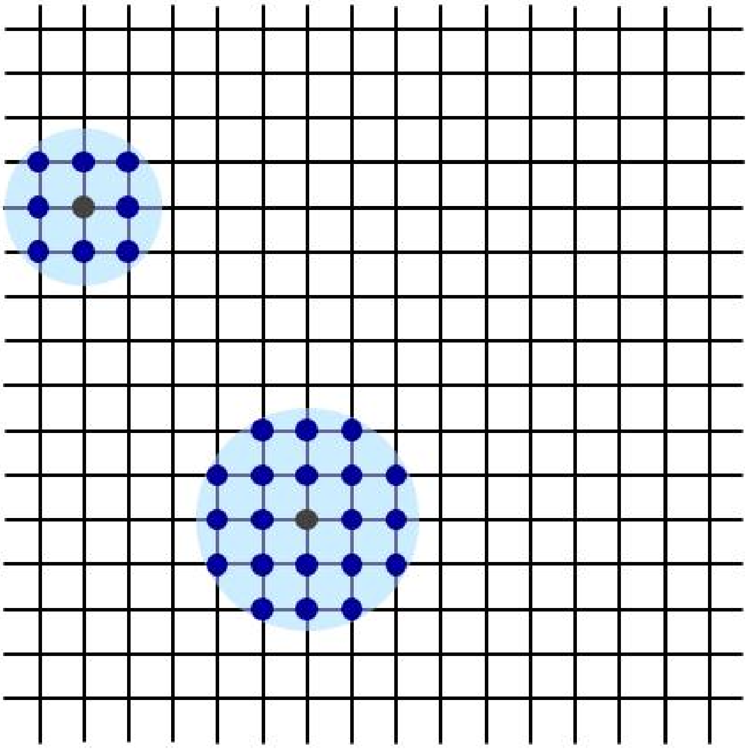


Figure 4.1: Grid topology with transmission radius for 8 and 12 neighbors.

4.2 Simulation setup

In order to model the behavior of the proposed opportunistic routing scheme and optimize its parameters, I used a MATLAB-based platform as a time-driven simulator. Experiments were conducted on a 40x40 node grid topology in order to measure delay, loss ratio 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 traditional single path routing has been simulated and used as the basis of comparison.

4.2.1 Single path routing model

Single path routing was modelled as a stochastic procedure of packet forwarding, rather than simulating nodes and their queues. 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.

4.2.2 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. More generally, 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. "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. For example when metric noise is 0.5, noisy metric values are randomly chosen from the range [0.75*accurate_metric_value, 1.25*accurate_metric_value].

4.2.3 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 Forwarding probability function

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. To ensure the packet's progressing to the destination, at least one neighbor with forwarding probability equal to 1 is needed; ideally this would be the neighbor on the shortest path to destination. This leads to a linear forwarding probability function. As the PER increases, more "certain" forwarders are needed, to make up for failed packet receptions, but the need arises to relax potential collisions among them.

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. Figure 4.2 illustrates how tilting the slope increases the number of certain forwarders in an 8 neighbor topology. Initially there is only one certain forwarder, the one that is closest to the destination. Two

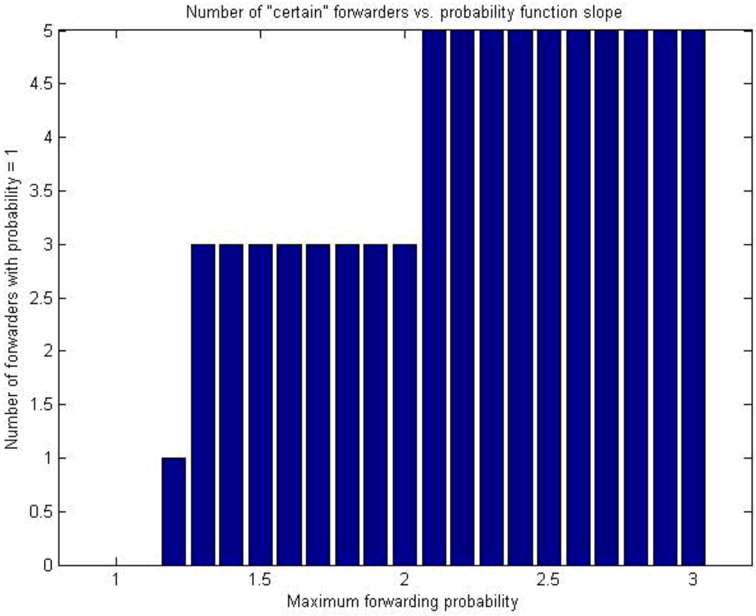


Figure 4.2: Increasing the number of nodes that always forward by tilting the probability function slope.

more forwarders are added that are closer to the destination than the node that transmitted the packet. If the slope is increased, two more forwarders are added, that are slightly further away from the destination, which will make the packet progress diverge sideways from the shortest path. It is reasonable that values of maximum probability in [1.4,2] contribute to the packet's progress without diverging much from the shortest path. Furthermore, for a given PER value, having more than one certain forwarders yields lower delay.

This is verified by Figure 4.3 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. Figure 4.4 shows the same behavior in a 12 neighbor denser topology.

4.3.2 Back-off window differentiation

There were two general approaches to back-off window schemes with respect to the window range values. One one hand, in the fixed back-off window option, all transmitting nodes would 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

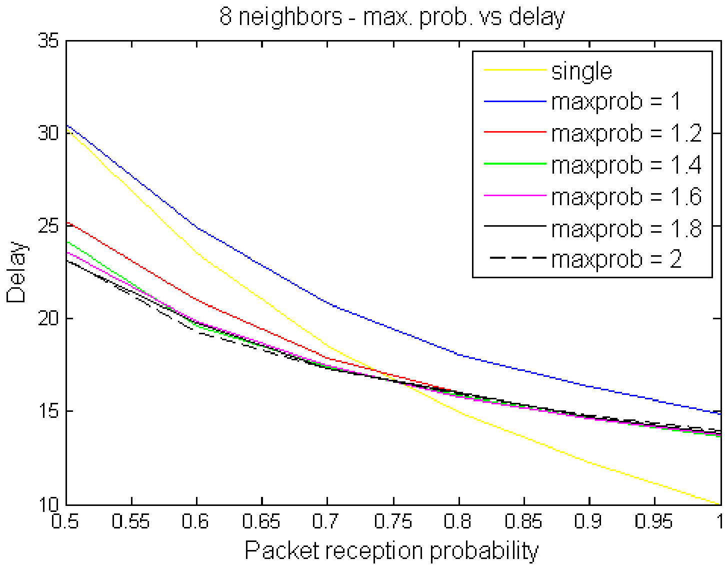


FIGURE 4.3: Tilting the slope creates the piece-wise probability function, which leads to lower delays for 8 neighbour topologies, on a 10 hop shortest path distance. The back- off window scheme used was variable with range [1, 8].

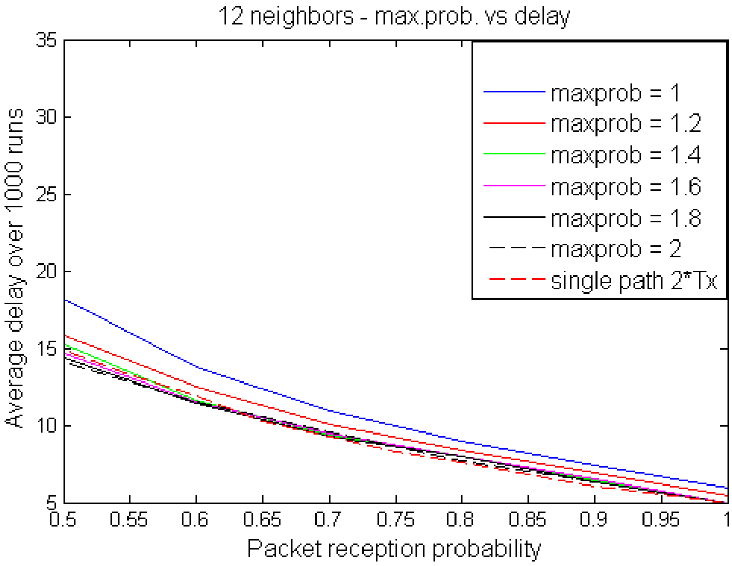


FIGURE 4.4: The piece-wise probability function yields lower delays for larger slope values in the case of 12 neighbor topologies as well. The back-off window scheme used was variable back-off window with range [1, 12].

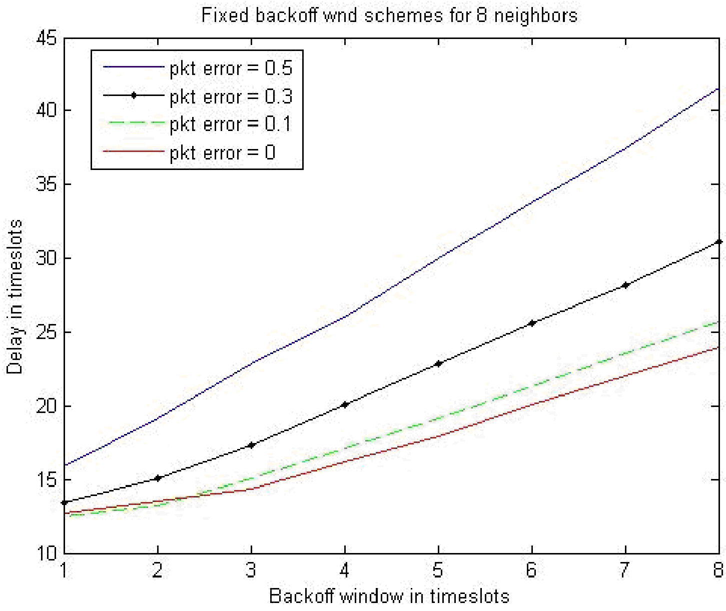


FIGURE 4.5: Delay performance over a 10 hop distance for fixed back-off schemes in an 8-neighbor topology.

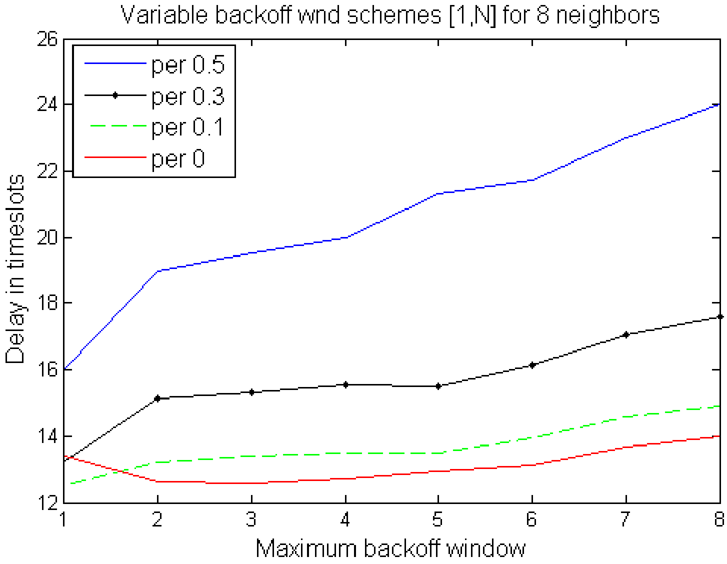


FIGURE 4.6: Delay performance over a 10 hop distance for variable back-off schemes in a 8-neighbor topology.

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. Figures 4.5 and 4.6 illustrate the delay performance of the fixed and differentiated back-off window schemes respectively, for increasing packet error rate values, in a scenario where the source is 10 hops away from the destination. There is 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.

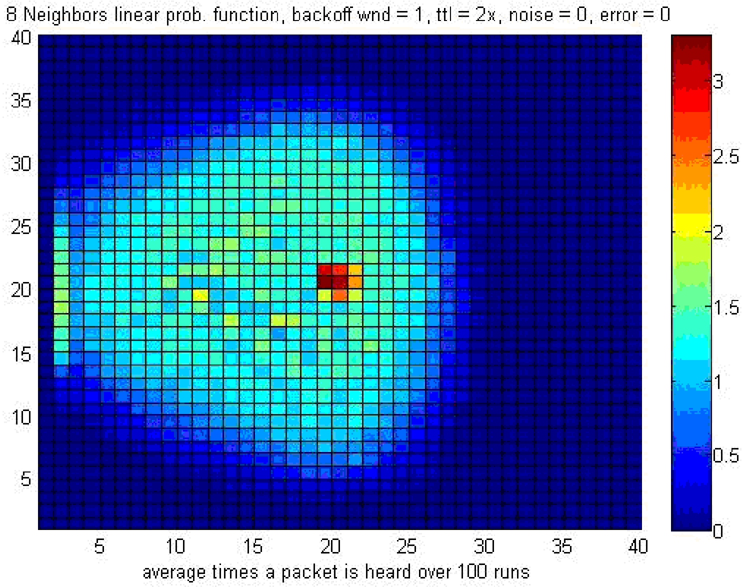
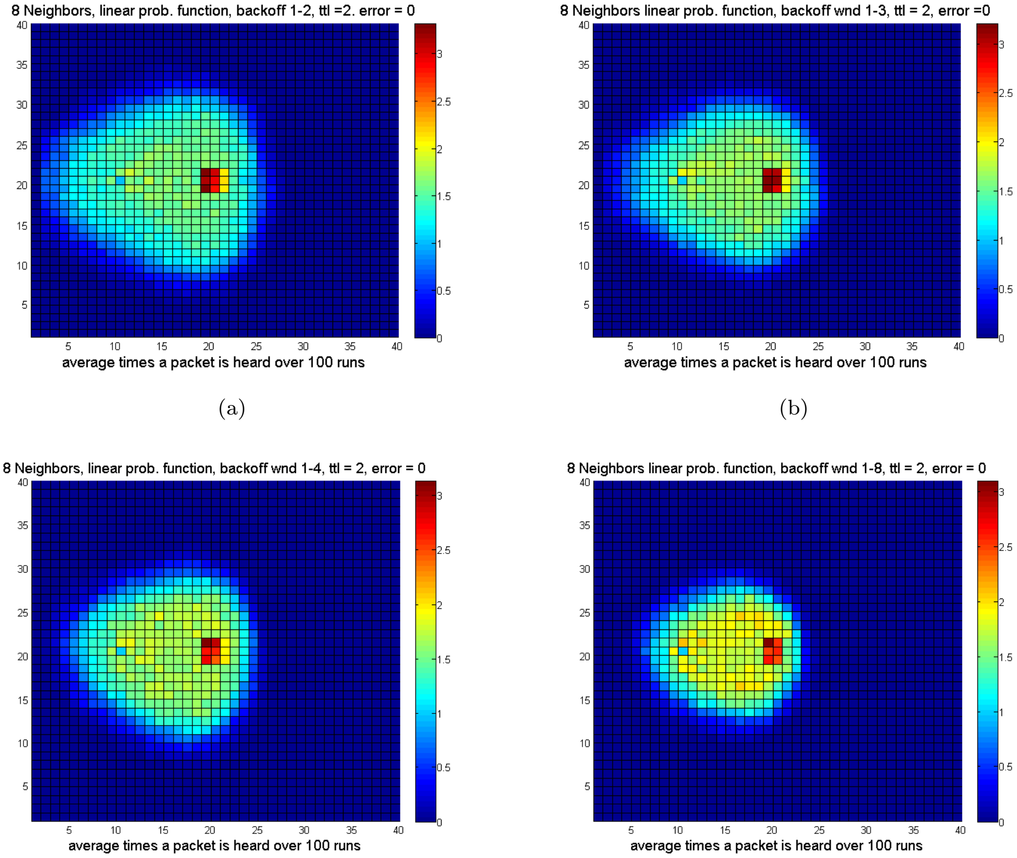


Figure 4.7: A back-off window of size 1 will yield slightly lower delay than a variable back-off window, but significantly higher resource consumption.

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.

For all results referring to footprints hereon the source node's coordinates are (10,20) and the destination is at (20,20). Figure 4.7 illustrates the footprint for a flow with back-off window equal to 1. As we increase the width of the back-off window the flooding is limited to an area around the shortest path. This is shown by the plots in Figure 4.8. It should be noted that a linear forwarding probability function with maximum probability equal to 2 was used for these experiments. 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 back-off 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.



(a) (b)

(c) (d)

Figure 4.8: Increasing the range of the back-off window interval from [1, 2] to [1, 8] leads to lower resource consumption when piece-wise probability function is in use.

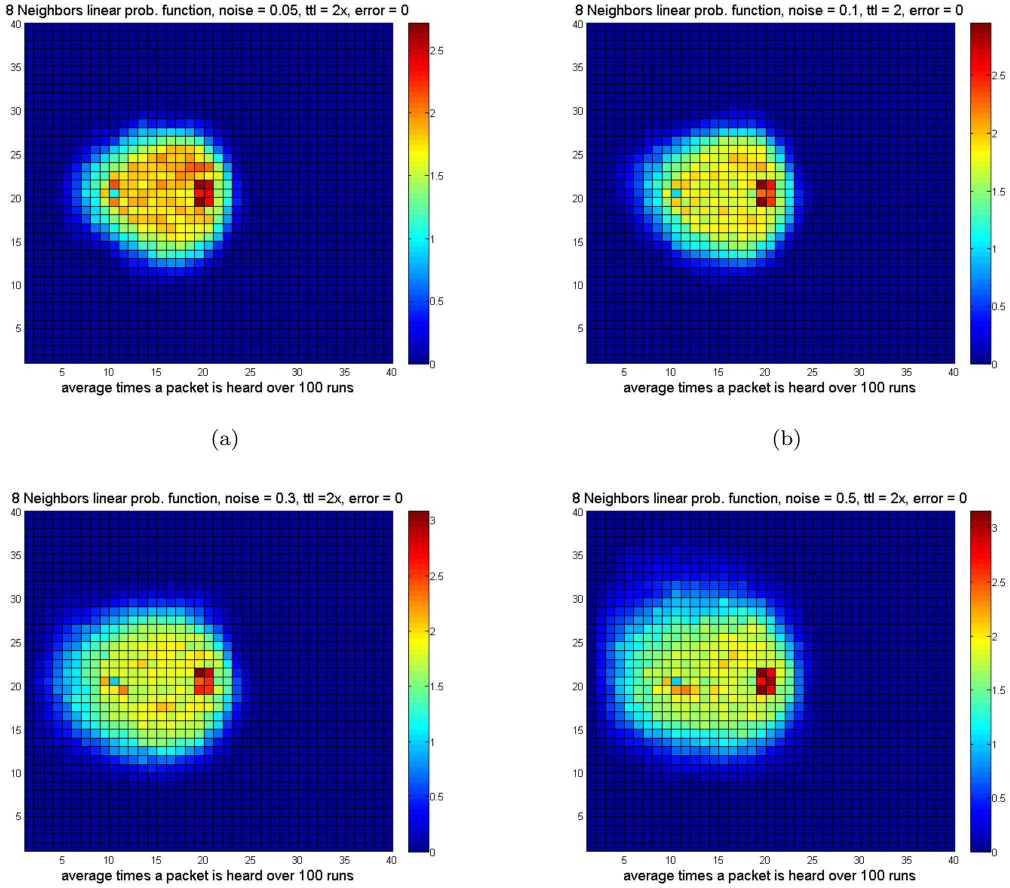
4.3.3 Resource consumption vs. metric miscalculation

In order to evaluate the robustness of our scheme to mistakes in metric estimations, I introduced the concept of metric miscalculation noise.

4.3.3.1 Linear probability function

To measure resource consumption for the linear probability function with maximum probability set to 2, I 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 I use a balanced back-off window scheme in the interval [1, 4].

Figure 4.9 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 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.



(a) (b)

(c) (d)

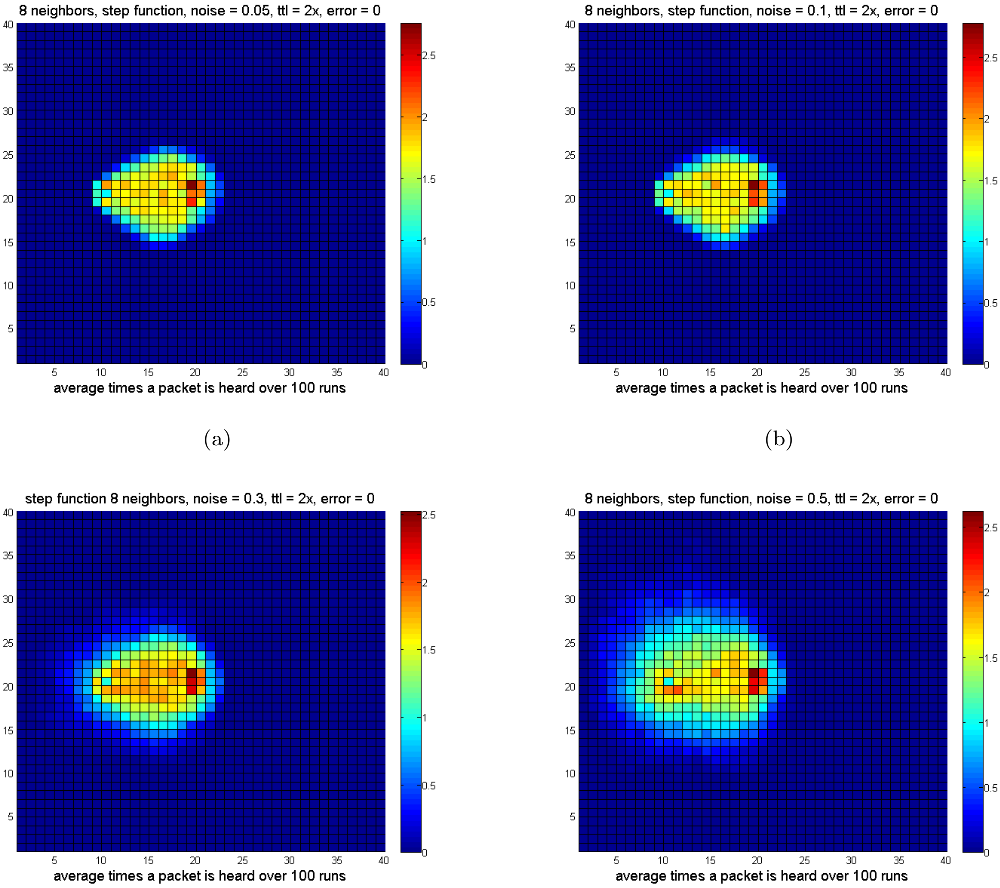
Figure 4.9: Increasing the error in metric value estimation affects resource consumption slightly in an 8 neighbor topology, for piece-wise probability function.

4.3.3.2 Step-wise probability function

Furthermore, I compare the performance of the linear probability function in the presence of noise to that of the step-wise probability function.

For the purposes of these experiments the step was set to 0, such as the neighbors that have a smaller distance from destination, than the node currently holding the packet, will forward the packet. The source is again at (10,20) and the destination at (20,20) and the back-off scheme is again set to [1, 4] interval.

Figure 4.10 illustrates the step-wise function's performance under increasing metric noise conditions. For the same values of metric noise, the step-wise function consumes fewer resources than the linear function and it 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.



(a) (b)

(c) (d)

Figure 4.10: Increasing the error in metric value estimation affects resource consumption even less in an 8 neighbour topology, for step-wise probability function.

4.3.3.3 Metric noise vs. delay

To further understand the impact that metric noise has on our scheme, I also measured the delay over the same 10 hop path for both the linear and the step-wise probability function, for various packet error rate values. Figures 4.11 and 4.12 demonstrate that metric noise does not affect delay performance significantly for neither the linear nor the step-wise probability function.

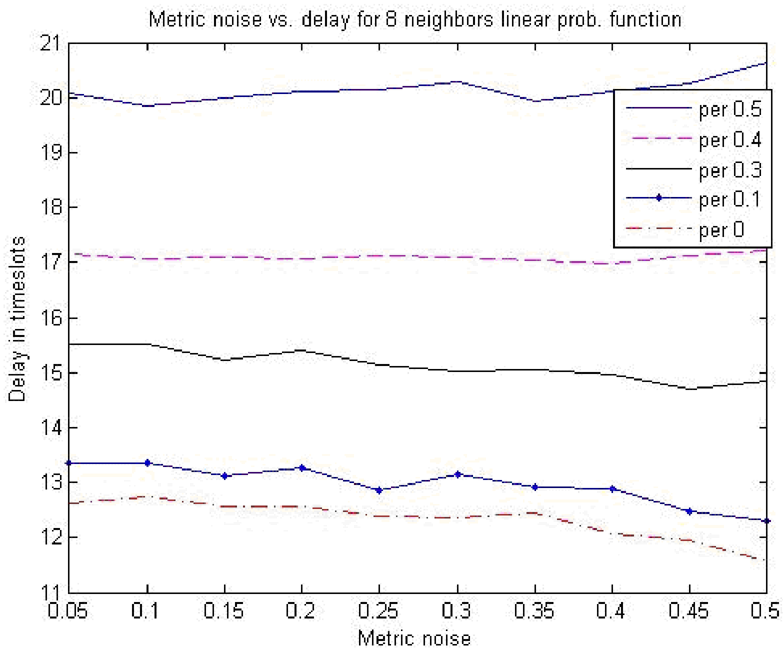


Figure 4.11: Error in metric value estimation does not affect delay over a 10 hop distance significantly when piece-wise probability function is used.

Table 4.1: Delay performance over a 10 hop path in a randomly generated dense topology.

	step-wise	piece-wise
PER 0.2	16.402545	16.312124
PER 0.3	18.479613	17.932832
PER 0.4	21.399183	20.484211

4.3.4 Random topologies

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 4.2.3. 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 4.1 and 4.2 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 section4.3.1 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

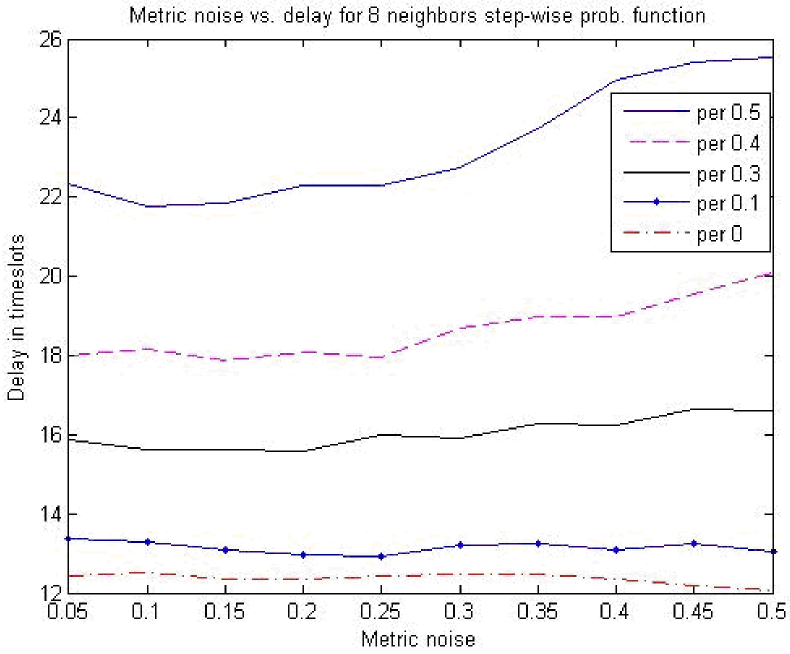


Figure 4.12: Error in metric value estimation does not affect delay over a 10 hop distance notably when step-wise probability function is used.

Table 4.2: Delay performance over a 10 hop path in a randomly generated sparse topology.

	step-wise	piece-wise
PER 0.2	16.819345	16.775213
PER 0.3	19.102419	18.420050
PER 0.4	22.439709	21.428213

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 deciding 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.

4.4 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 3.2.5.1. These new elements added complexity to the simulator 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 and

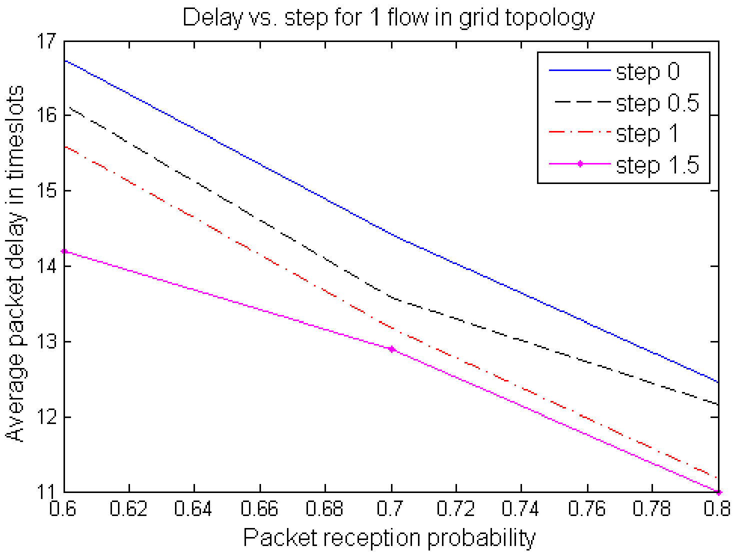


FIGURE 4.13: Delay performance of a 50 packet flow using step-wise probability function to adapt to increasing packet error probability

the need of averaging delay results. 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.

4.4.1 Multiple packets in one flow

In order to study the behavior of the proposed 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 4.13 demonstrates how the scheme copes with increasing packet error 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

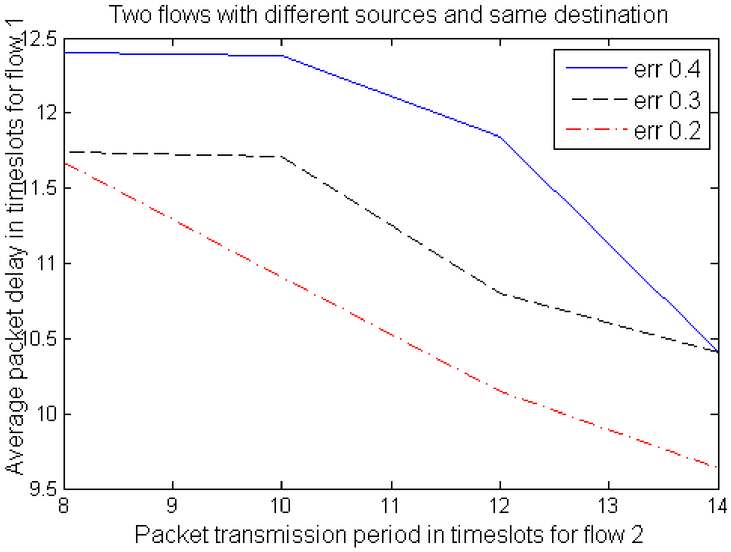


Figure 4.14: Delay performance of a 50 packet flow while competing with another flow with the same destination

4.4.2 Coexistence of two flows in the network

In addition to verifying that the proposed scheme supports the existence of flows, it is interesting to examine how two flows interact in the network and confirm that the proposed 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 4.14 represents the scenario of two nodes sending packets to the same "sink" node, whereas Fig. 4.15 refers to the case of a "downlink" in a wireless mesh network. The third scenario is a worst case scenario of both flows sharing the same source and destination and is illustrated by Fig. 4.16 It should be noted that the packet loss of the flows did not exceed 5% even in the third scenario.

4.5 Comparison to other opportunistic routing schemes

In order to evaluate the suggested 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.

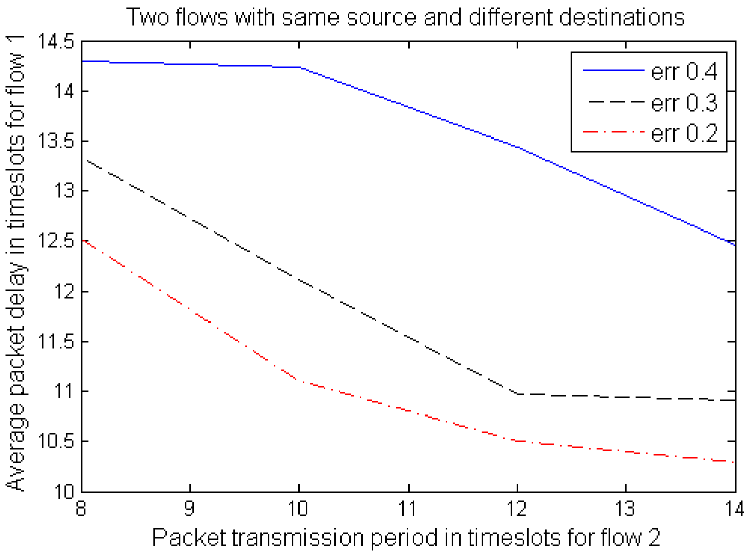


FIGURE 4.15: Delay performance of a 50 packet flow while competing with another flow with the same source

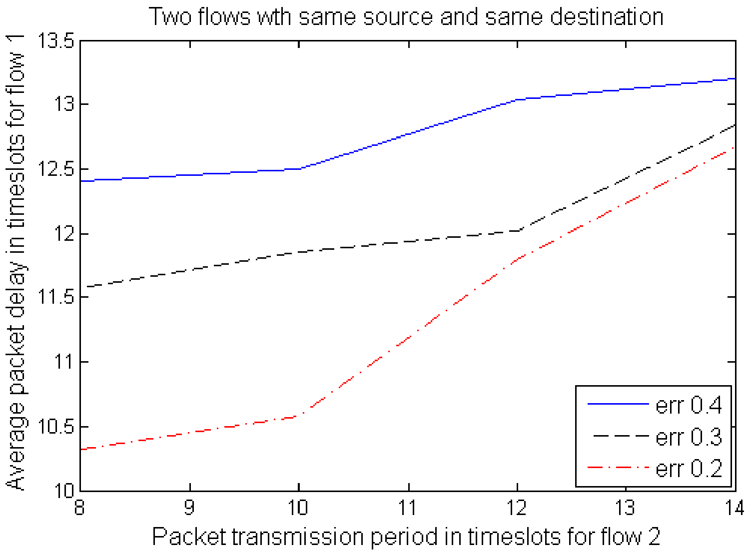


FIGURE 4.16: Delay performance of a 50 packet flow while competing with another flow with the same source and destination

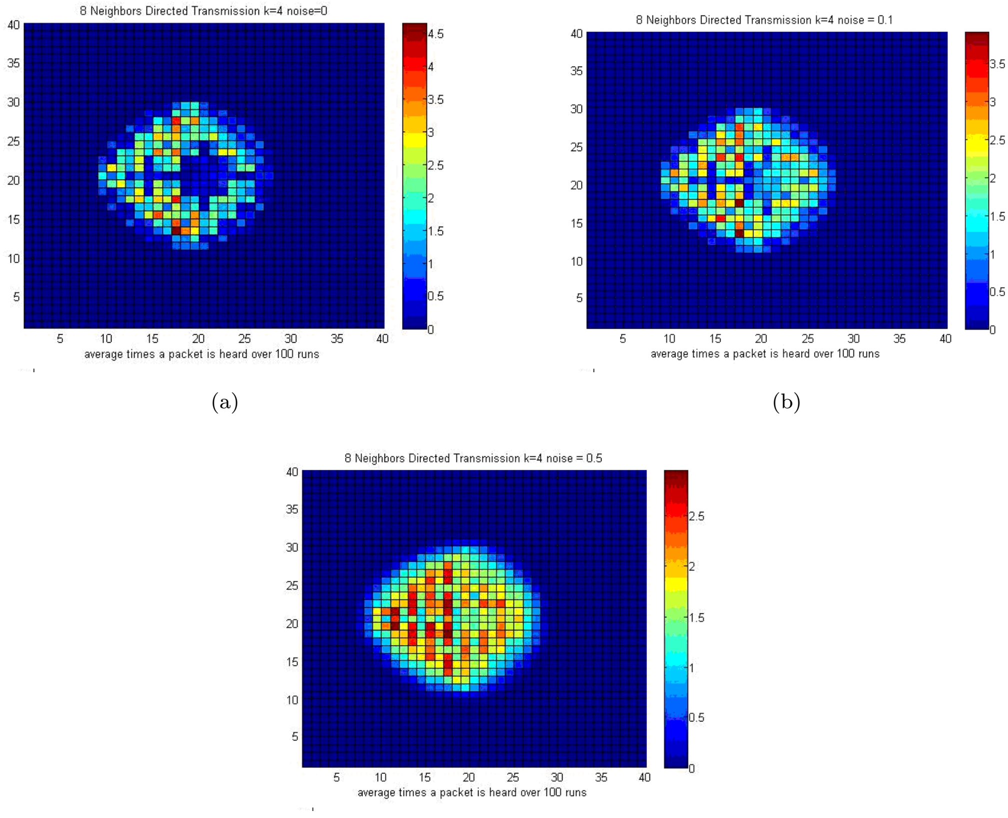
4.5.1 Comparison to Directed Transmission

Directed Transmission is a parametric probabilistic routing protocol suggested in [BEK$^+$07], 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:

$$P(Ri) = \exp\{k[d(S,D) - d(R_i,D) - i]\}, \tag{4.1}$$

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 the scheme I suggest. 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 are depicted in Figure 4.17. These results demonstrate that a protocol can be simple enough and conserve resources to be applied in WSNs without suffering from low delay performance.



(a) (b)

(c)

Figure 4.17: Directed Transmission is equally robust to metric noise as the suggested scheme, however it requires more retransmissions to deliver a packet on average.

As far as delay is concerned, the performance of both directed transmission and my approach was not affected by increase in metric miscalculation; however our approach yields significantly

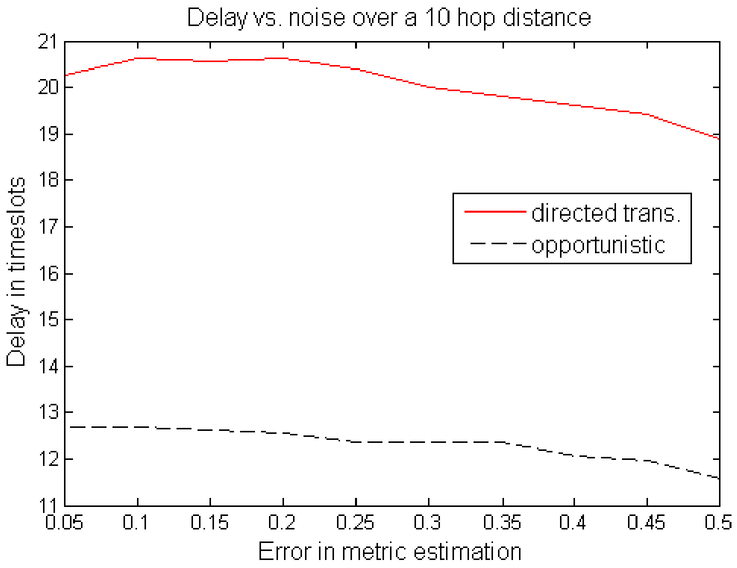


Figure 4.18: Both Directed Transmission and the proposed opportunistic scheme are robust to metric miscalculation but the latter yields far lower delays.

smaller delays. In summary, as shown in Figure 4.18 our scheme outperforms Directed Transmission both in terms of resource consumption and delay, under increasing metric miscalculation.

4.5.2 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, as shown in Figure 4.19.

When metric miscalculation is present, SOAR's delay increases significantly, as opposed to the proposed scheme's performance which is unaffected, as shown in Figure 4.20. 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.

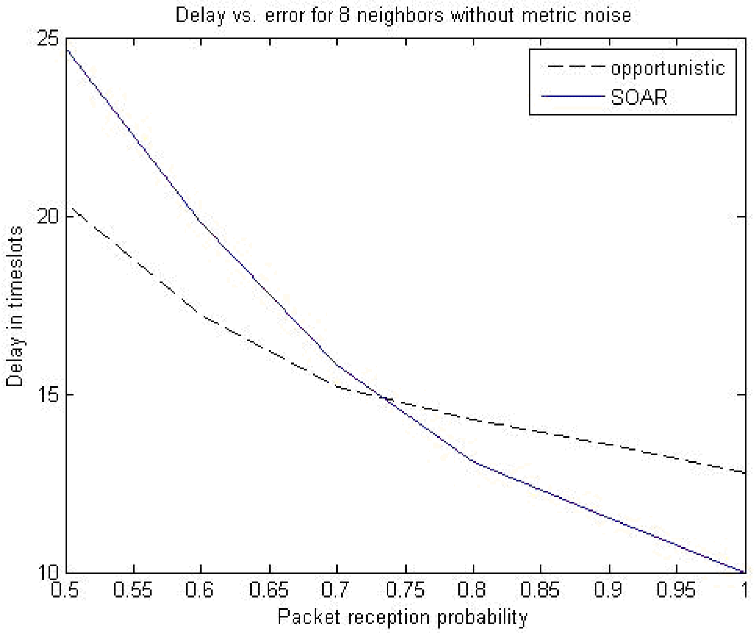


FIGURE 4.19: The proposed scheme yields lower delays than SOAR for packet error probability over 0.25.

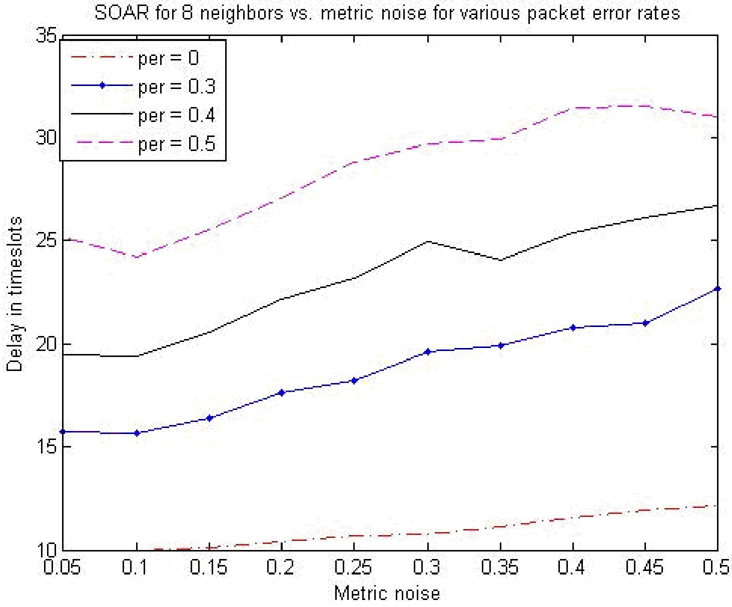


FIGURE 4.20: SOAR's delay performance degrades as metric miscalculation is increased.

Chapter 5

Conclusions and Further Work

5.1 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, I have shown 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 tradeoff between differentiating each forwarder's back-off value to reduce resource consumption and reducing delay. Simulation results show that 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.

Finally, it is confirmed that the suggested model supports flows successfully, in a variety of topologies and can adapt to the probability of packet error, utilizing a step-wise probability function. In order to evaluate the proposed scheme, it has been compared to Directed Transmission and SOAR under metric miscalculation and channel loss conditions.

5.2 Further Work

The assumption of the geometric propagation model should be relaxed, so that the behavior of the suggested opportunistic scheme can be examined with the help of a more realistic SNR model. In this manner, transmissions will not be received only in a specific radius with a uniform correct packet reception probability, but in different radii with decreasing correct packet reception probability as the radius increases. Thus, it will also be possible to apply the opportunistic scheme in tandem topologies.

In order to be adaptive to channel error conditions when there is also significant collision probability, the opportunistic scheme will need a mechanism that can recognize the difference between loss due to collisions and loss due to channel errors. This can be achieved in the following ways. A node can make SNR measurements when it acts as receiver. High SNR measurement combined with a possible "corrupt" packet from which only the header can be decoded, imply that there was likely a collision at that node. Conversely, low SNR measurements and no packets captured imply that packets are likely to be lost due to low channel quality. However, these measurements are taken at the receiving nodes. A means to propagate this information about losses to the packet's sender can be negative acknowledgements (NACKs) which will propagate opportunistically, limiting redundant transmissions at the same time. In a different approach, each node, regardless if they are receiving or transmitting, can measure how often the channel is utilized. High channel utilization would imply that collisions are more likely whereas low channel utilization may imply low channel quality.

Bibliography

[BEK$^+$07] C. L. Barrett, S. J. Eidenbenz, L. Kroc, M. Marathe, and J. P. Smith. Parametric probabilistic sensor network routing. In *ACM"07*, 2007.

[BM05] S. Biswas and R. Morris. Opportunistic routing in multi-hop wireless networks. In *SIGCOMM"05*, 2005.

[DLC07] Junzhao Du, Hui Liu, and Ping Chen. Omr: An opportunistic multi-path reliable routing protocol in wireless sensor networks. In *ICPPW '07*, 2007.

[EP94] Perkins C. E. and Bhagwat P. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *ACM SIGCOMM Computer Communication Review*, 24(4), 1994.

[JMB01] David B. Johnson, David A. Maltz, and Josh Broch. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In *In Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5*, pages 139–172. Addison-Wesley, 2001.

[JRDJ] De Couto D. S. J, Morris R., Aguayo D., and Bicket J. A high-throughput path metric for multi-hop wireless routing. In *International Conference on Mobile Computing and Networking.*

[JYL02] Haas Z. J, Halpern J. Y, and Li L. Gossip-based ad hoc routing. In *INFOCOM '02*, pages 1707–1716, 2002.

[LZH$^+$09] Haitao Liu, Baoxian Zhang, Mouftah H., Xiaojun Shen, and Jian Ma. Opportunistic routing for wireless ad hoc and sensor networks: Present and future directions. *IEEE Communications Magazine*, 47(12), 2009.

[MTG04] Stephen Mueller, Rosep. Tsang, and Dipak Ghosal. Multipath routing in mobile ad hoc networks: Issues and challenges. In *In Performance Tools and Applications to Networked Systems, volume 2965 of LNCS*, pages 209–234. Springer-Verlag, 2004.

[RP00] Elizabeth M. Royer and Charles E. Perkins. An implementation study of the aodv routing protocol. In *In Proceedings of the IEEE Wireless Communications and Networking Conference*, 2000.

[RSMQ06] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu. Simple opportunistic. routing protocol for wireless mesh networks. In *WiMesh"06*, pages 48–54, 2006.

[YYW⁺05] 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 2005*, pages 146–158, 2005.

[ZLZ08] Kai Zeng, Wenjing Lou, and Hongqiang Zhai. On end-to-end throughput of opportunistic routing in multirate and multihop wireless networks. In *IEEE INFOCOM '08*, pages 99–100, 2008.

[ZN07] Z. Zhong and S. Nelakuditi. On the efficacy of opportunistic routing. In *SECON '07: In proceedings of the 4th IEEE International Conference on Sensor and Ad Hoc Communications and Networks*, pages 836–843, 2007.