Multi-Channel Opportunistic Routing in Multi-Hop Wireless Networks

ANATOLIJ ZUBOW, MATHIAS KURTH and JENS-PETER REDLICH

Humboldt University Berlin

Unter den Linden 6, D-10099 Berlin, Germany

(zubow|kurth|jpr)@informatik.hu-berlin.de

Abstract. We propose and investigate Multi-Channel Extremely Opportunistic Routing (MCExOR) which is a protocol that extends Extremely Opportunistic Routing by utilizing multiple RF channels in multi-hop wireless networks. Large numbers of transmissions per end-to-end delivery combined with interference are the main reasons for the low capacity of wireless multi-hop networks. MCExOR reduces the overall number of transmissions in wireless multi-hop networks by opportunistically skipping nodes in a packet's forwarding path. The use of multiple non overlapping RF channels contributes to the reduction of overall interference.

In contrast to other approaches MCExOR only needs one RF transceiver per device. We present algorithms for route discovery and packet forwarding. A significant benefit of MCExOR is that the selection of RF channels is independent of the routing function. Finally, with the help of simulations we show that MCExOR outperforms traditional protocols like ad-hoc on-demand distance vector routing through the simultaneous use of multiple RF channels. In combination with realistic radio propagation models an increase in the throughput is observed due to the opportunistic feature of MCExOR. With the increasing number of RF channels the overall throughput increases superproportionally. Unlike other multi channel approaches even a single packet flow can benefit from the existence of multiple channels.

Keywords: Mesh networks, wireless multi-hop networks, ad-hoc networks, wireless routing, opportunistic routing, multi channel, interference, ExOR, MCExOR, Berlin RoofNet.

1. Introduction

Wireless multi-hop mesh networks play an increasingly important role as backbones for sensor networks and as community networks that provide Internet access in urban areas [8]. Nevertheless, one of their biggest challenges is the insufficient scalability with increasing number of nodes and users [1]. The most important reason for this phenomenon can be found in the structure of a multi-hop network: a node is responsible not only for the transmission of its own data, but also for forwarding packets of other nodes. No less significant is the fact that wireless network nodes in close proximity interfere with each other because they share the same medium (RF spectrum).

Extremely Opportunistic Routing (ExOR) is a promising approach for improving the throughput of wireless multi-hop networks [2]. While most wireless network models use wire-like point-to-point links that try to mask the fact that wireless transmissions are broadcasts by nature, ExOR uses this fact to its advantage: In a wireless network a link exists between every pair of nodes, although the error rate may be rather high for some of these links. All packet transmissions (which are layer 2 broadcasts) can potentially be received by every remote node, with a certain non-zero probability. This brings up the opportunity that a packet might skip a few nodes on its forwarding path if current radio propagation conditions are favorable. ExOR uses this approach to significantly reduce the average path length of most end-to-end transmissions.

IEEE 802.11 provides several non overlapping RF channels. If multiple channels are used within one region (collision domain) multiple transmissions can take place simultaneously without interference resulting in a positive impact on overall network throughput. Although routing protocols that use multiple channels have been studied before [5][7], they are not applicable in most 'real' IEEE 802.11 multi-hop installations because they require nodes with more than one transceiver. Most IEEE 802.11 devices are equipped with only one transceiver. This leads to the problem that nodes which operate on different channels cannot communicate with each other. Nevertheless, devices with just one transceiver can still make use of multiple channels by quickly switching to the channel of the intended receiver. Today's IEEE 802.11 hardware is capable of switching the RF channel within a fixed delay of 80 μs [5].

The multi-channel routing protocol MCRP [7] provides an interesting approach for using devices with only one transceiver on multiple channels. MCRP is an extension of the well known ad-hoc on-demand distance vector routing protocol (AODV [9]). With the help of simulations So et al. [7] showed the superiority of MCRP in comparison to AODV.

The present paper is organized as follows. At first we describe existing routing protocols for multi-hop networks and the idea behind our multi-channel opportunistic routing protocol. In the following section we present design details of MCExOR like link probing, route discovery and packet forwarding. In the final section the results of measurements using the JiST/SWANS [12] simulator are presented and compared to AODV and ExOR.

2. The Idea behind MCExOR

Many routing protocols are known today which were developed particularly for multi-hop mesh networks. For example, Dynamic Source Routing (DSR [15]) and AODV, as well as protocols especially designed for wireless mesh networks like ExOR. Recently, new protocols for the use of multiple RF channels like MCRP were introduced. In this section we present the idea on which MCExOR is based – an opportunistic routing protocol that utilizes multiple RF channels in wireless multi-hop networks.

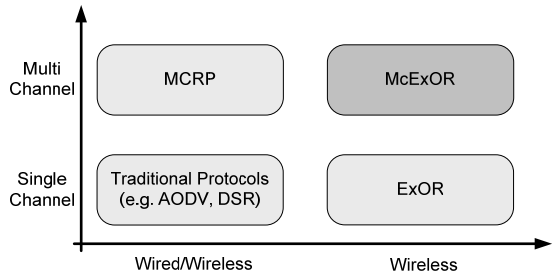

Figure 1: A classification of routing protocols.

2.1 Traditional Protocols

Routing protocols can be characterized as being proactive or reactive. Reactive protocols like DSR or AODV work on-demand which means that they exchange routing information only when it is required and not periodically like proactive protocols (e.g. Optimized Link State Routing Protocol (OLSR) [20]). Reactive protocols introduce route discovery and route maintenance phases. The route discovery mechanism is used to find a route from a source to a destination node. The source node emits a network-wide route request (RREQ) broadcast which is answered by the destination node with a route reply packet (RREP). This way, the source node learns about routes to a destination node. The route maintenance algorithm refreshes entries in the routing data base and decides on their validity.

On the other hand, routing protocols can be distinguished in regard to their knowledge of the network's topology. With distance vector routing (e.g. AODV) only local information about the network is stored on a node (a node memorizes only the next hop towards a destination and the associated cost of the end-to-end path). In addition, with link state routing (e.g. DSR), a global view of the network is necessary to perform routing functions, i.e. the current state of all links in the network needs to be known to all nodes. With this information it is possible for a node to compute an optimal forwarding path towards any destination locally, e.g. using Dijkstra's shortest path algorithm.

2.2 Extremely Opportunistic Routing

Traditional routing protocols do not sufficiently take the basis for MCExOR. Extensive observations about the quality of a wireless link were made [1]. In a nutshell, in a wireless network the majority of links has a delivery probability different from one. Hence, a hop count metric is not useful for finding a good end-to-end forwarding path. Instead the minimization of the number of transmissions towards the destination leads to better results [10]. However, in general, the majority of links are of poor quality. But this is not necessary a disadvantage, since there are quite many of those poor quality links, which can be used simultaneously by ExOR.

2.2.1 Working Method

The following example demonstrates the principles behind ExOR. Consider the network in Figure 2. Many routes exist between node A and D. For instance, it is possible for node A to transmit a packet to node D directly in one hop. However, because the probability of a successful transmission from A to D is very low, a packet will likely be retransmitted multiple times. Alternatively, node A can send a packet via nodes B and C towards the final destination D. In this case, a packet must be transmitted multiple times too (multi hop), but possibly without many retransmissions.

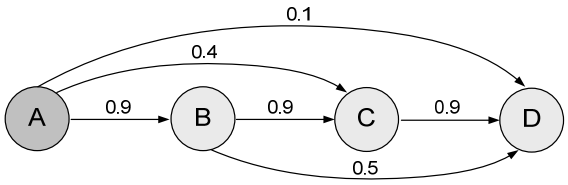

Figure 2: Network with delivery ratios (e.g. the probability of a successful transmission from node A to C is 0.4).

When transmitting a packet from node A to node B towards the final destination D, it is possible that the packet is successfully received not only by node B, but also by node C or even D. In this case an additional transmission of the packet from node B to C is unnecessary and a waste of resources. Instead, the node that is closest to the final destination should continue the forwarding process. Similarly, if node A tries to send a packet directly to D and the transmission fails, it is likely that the packet reached node B or even node C. Hence, it could make sense to transmit the packet from B or C to D instead of retransmitting it from node A. This mechanism was first introduced by the ExOR protocol [2]. ExOR uses a so-called 'candidate set' which contains all nodes useful for the forwarding of the packet towards the destination. In contrast to traditional protocols like AODV, ExOR uses multiple potential nodes for the next hop. From this point of view AODV can be seen as ExOR with a candidate set of size one. Both protocols provide a layer-3 unicast service. However, different mechanisms (unicast vs. local broadcast) are used on the data link layer.

ExOR determines the candidate set by taking all possible paths from the source to the final destination into account. The size of the candidate set is a configurable parameter. The candidate set contains the first node of every path towards the destination, sorted by ascending path length. Ties are broken using the ETX metric [10]. ExOR is able to use distant links with a high loss rate in order to skip intermediate hops towards the destination. In contrast, AODV completely ignores these links. Candidate nodes acknowledge the successful reception of a packet in a prioritized manner, i.e. a candidate with higher priority sends its acknowledgement before any lower prioritized candidate (slotted acknowledgment). Among all nodes of a candidate set that successfully received a packet, the node with the highest priority (i.e. the first node to acknowledge) is responsible for forwarding the packet towards the final destination. Each time a packet is forwarded, a new candidate set is computed by the forwarding node.

The performance of ExOR is positively affected by high network density because nodes in close proximity of a node are likely included in its candidate set. The size of the candidate set is essential for the performance of ExOR. With small candidate sets ExOR tend to behave like AODV and uses links with low quality, making it likely that packets do not make any progress towards the final destination.

The main differences between AODV and ExOR can be characterized as follows. Both protocols use different mechanism for packet forwarding. AODV only specifies the next hop towards the destination. In contrast, ExOR uses a set of candidates as potential receivers for the next hop to increase the probability of skipping intermediate nodes towards the destination. The main differences are summarized Table 1.

	AODV	ExOR
Medium	wired/wireless	wireless
Forwarding	next hop	candidate set
Topology	any	dense

Table 1: Comparison between AODV and ExOR

2.3 Strategies for Using Multiple Channels

A promising approach to increase the capacity of wireless multi-hop mesh networks is the simultaneous use of multiple RF channels for layer-2 packet transmission. However, this is at the expense of additional channel management. With the use of multiple channels the capacity of the network is increased even further because of the reduction of packet loses due to interference (collisions). However, this approach introduces new problems. For example, it is not possible for a node with only one transceiver to operate on multiple channels at the same time. Hence, we will use devices with only one transceiver that are able to switch from one channel to another within a short time.

Routing protocols for such a platform have to deal not only with route discovery, but also with the assignment of a proper channel to each node. Nodes A, B and C in Figure 3 belong to the same network; however, they use different channels 1, 2 and 3. At first we consider the case where node A sends a packet to C and all nodes use the same channel. With ExOR, both nodes B and C would belong to the candidate set of the transmission A→C.

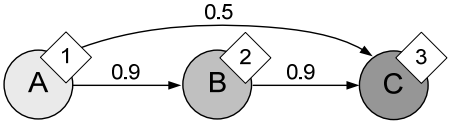

Figure 3: Network with nodes A, B and C. The node's channel is specified in the upper right corner. For instance, channel 1 is assigned to node A.

If multiple channels are available, the selection of the candidate set depends on the particular channels used by each node. In the present example node A has the choice between candidate set {B} (channel 2) and {C} (channel 3). Nevertheless, the candidate set {B, C} is not an option, because nodes B and C do not operate on the same channel. Algorithms for the computation of candidate sets are introduced in Section 3.4.

2.3.1 Channel Assignment - Nodes vs. Flows

Using multiple RF channels in one wireless network requires new algorithms for channel assignment and management. From [7] we know at least two approaches: In the first approach, channels are assigned to nodes independently of packet flows. A node along a path only needs to know the next node towards the destination as well as the channel this node is operating on. If this information is available the sending node can transmit packets by switching to the channel of the destination node. The advantage of this approach is that the channel assignment is independent of the routing algorithm. Therefore, both aspects can be addressed independently. In the easiest case every node randomly chooses a channel and informs its neighbors via multi-channel broadcast (MCBC) which is a broadcast on all available channels. However, nodes operating on different channels create a new problem: 'deafness' [4]. Deafness occurs if two nodes cannot communicate with each other because they operate on different channels. As we will see later, this problem is less pronounced in MCExOR which uses multiple nodes (a candidate set) as potential next hop nodes. Deafness is the main reason why MCRP uses a second approach to channel assignment and management: channels are assigned to flows (between nodes). After the successful establishment of a route from the source to the destination all nodes along this route have to be assigned the same channel as long as the flow exists.

However, assigning channels to flows has two significant disadvantages. First, the available capacity along a path (flow) is substantially reduced by self-interference [6][1][16]. It is commonly assumed that the interference area of a packet is twice as big as the transmission area itself. If multiple packets are transferred along a path, self-interference reduces the number of simultaneously active links. If too many links are active, collisions will occur that provoke retransmissions. The second disadvantage of MCRP is that the routing function implies the channel assignment which leads to very complex protocols. So the latter approach does not seem to be very promising. We will focus on the assignment of channels to nodes independently of packet flows and independently of the routing function.

	MCRP	MCExOR
Channel assignment	flow	node
Channel assignment decoupled from routing	no	yes
Protocol complexity	high	low

Table 2: Comparison between MCRP and MCExOR

2.3.2 Comparison with MCRP

MCExOR possesses important advantages compared to MCRP. It inherits the opportunistic nature of ExOR along with its advantages (Table 1). Furthermore, MCRP assigns channels to flows instead of nodes which leads to self-interference along the path. MCRP does not scale with the number of simultaneous flows in the network [7]. In MCExOR, however, multiple concurrent flows pose no problem, because nodes along a flow do not necessarily operate on the same channel. In addition, the deafness problem of MCRP plays a minor role in MCExOR. It is rather unlikely that during a transmission all nodes of the candidate set are 'deaf'. Moreover, the MCExOR protocol is easier and more elegant: no special cases must be handled like with MCRP. Several optimizations are not possible with MCRP

due to the tight integration of the routing function and channel assignment. Table 2 summarizes the main differences between both protocols.

3. Design of MCExOR

MCExOR extends ExOR by utilizing multiple RF channels. It improves the network performance by choosing the RF channel with the most promising candidate set for every transmission. Furthermore, it uses spectrum diversity so that multiple simultaneous data transmissions can take place within a region without interference. In this section we describe the design of MCExOR. At first the local neighbor and route discovery are presented. Afterwards, algorithms for packet forwarding are introduced. Finally, we illustrate the modifications to IEEE 802.11 MAC layer as well as optimizations like duplicate suppression.

3.1 RF Channel Assignment

MCExOR assumes that each node is equipped with only a single wireless transceiver capable of sending and receiving on a fixed number of available RF channels. The transceiver unit is able to tune from one channel to another within a fixed delay. During this time, the transceiver is not able to send or receive. Thus, the node is 'deaf'. These assumptions hold for today's IEEE 802.11 hardware [5]. Furthermore, a so-called home channel is assigned to each node. The node announces its home channel to its neighbors. Data packets are sent on the home channel of the receiving node.

The RF channel assignment for nodes is decoupled from the routing protocol. MCExOR merely needs the information about a node's assigned channel to construct a candidate set. Hence MCExOR is not restricted to a fixed channel assignment. So the following approaches are only examples. The random strategy assigns channels to nodes in a random fashion. The main advantage of this approach is that no global view of the network is required. This strategy is simple to implement.

Alternatively a node chooses its channel based on the decision of its neighbors. It selects the least utilized channel in order to minimize the influence of neighboring nodes. The algorithm requires only local information, but it does not consider the characteristics of the wireless medium on the RF channel. Especially in indoor and urban scenarios, the quality of a wireless link depends on the used channel [11]. Instead of randomly choosing a channel the observed link quality information could be taken into account. For example, a node starting up could measure link qualities and select its home channel according to this measurement. It is also possible to periodically repeat the measurements and change the home channel dynamically. This way a node can adapt to the (possibly changing) characteristics of the wireless medium.

It should be noted that local information is perhaps not sufficient to make good decisions. It may be necessary to use global information about the network to optimize channel assignments, in recognition of certain flow's QoS requirements and of typical traffic patterns. Nevertheless, such a mechanism is very complex and cannot be considered in this paper.

3.2 Local Neighbor Discovery and Link Delivery Probabilities

Nodes discover neighbors through link probe packets. Every node periodically broadcasts link probes and neighboring nodes receive them. On receiving a link probe a node updates its neighbor list accordingly. Furthermore, it maintains a history of link probe receptions to calculate the delivery probability of the link from the sender to it. This proceeds similar to the calculation of ETX, except that links are considered unidirectional, i.e. every link has a delivery probability in forward and in backward direction. Within a link probe packet the calculated delivery probabilities of all neighbors are locally distributed so that every node knows the adjacent nodes and corresponding delivery probabilities within the hop count distance of 2. In the following sections we will refer to this algorithm as local neighbor discovery (LND).

Besides the link quality the used metric also takes foreign traffic into account. When foreign traffic emerges and congests the wireless medium, the reception probability of a link probe also decreases. Routing decisions are based on link quality information. If the quality of a link decreases due to foreign traffic, the routing layer will try to find a better path. So the protocol is suited for 'green field' deployments without foreign traffic as well as urban areas where it must coexist with devices running other protocols. However, besides foreign traffic the metric also considers own traffic since it cannot distinguish whether a link probe got lost due to a collision with a foreign or an own packet. So the metric is also a very basic means of load sharing.

MCExOR is a multi-channel protocol. Adjacent nodes do not necessarily operate on the same RF channel. For this reason consecutive link probes are broadcasted on a different channel in a round robin manner.

3.3 Route Discovery

The MCExOR routing protocol uses two kinds of algorithms for the discovery of new routes towards a destination: proactive and reactive. Both algorithms have pro and cons. For example, proactive route discovery results in knowledge about the global network topology, so that more sophisticated algorithms could be used (Section 3.4.3). However, the proactive algorithm does not scale with the size of the network. Furthermore it is not applicable in high dynamic networks, where the route to a node frequently changes and so the information becomes invalid. In this case the reactive 'on-demand' route discovery is used. Basically, the sending node initiates a route request which is forwarded by each receiving node on all available channels (flooding). Eventually the route request is replied by the destination. The route discovery overhead is low compared to the proactive algorithm. However, each sending node only knows a subset of the network topology. For traditional routing protocols like DSR this is not a problem, but as we will see this could to be a handicap for opportunistic protocols like MCExOR (Section 0).

3.3.1 Reactive Route Discovery

The reactive route discovery algorithm finds routes from a sender towards a destination in an 'on-demand' manner. The most important difference to traditional reactive protocols like DSR is that route requests (RREQ) are sent via multi channel broadcast. This is necessary because network nodes can operate on different channels. During a multi channel broadcast, the node emits a

RREQ not only on one channel, but on all available channels. Thereafter it quickly returns to its home channel. However, the delay that is introduced by the route discovery process is of importance and will be later observed in greater detail (see Section 4.4.2).

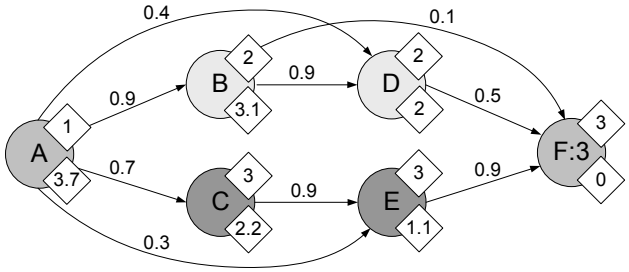

Figure 4: Example network with link delivery probabilities shown along the edges; RF channels indicated by the number in the upper right corner and the estimated transmission count to node F from each node of the graph indicated by the number in the lower right corner.

The following example illustrates the route discovery process. Figure 4 shows a network with 6 nodes (A to F) operating on 3 different channels (1 to 3). For the transmission of a packet from node A towards the destination F, node A needs a valid route. Furthermore, node A is using channel 1 and node F channel 3 as their home channel, respectively. The route discovery algorithm starts with the generation of a route request: Node A creates a RREQ packet and adds itself together with its home channel (1 in our case) in the packet's route header. Then node A performs a multi channel broadcast, i.e. it emits a RREQ packet on all available channels and switches back to its home channel. On receiving a RREQ a node adds its address together with its channel to the packet's route and multi-channel broadcasts the packet further. In the example, node F will eventually receive the RREQ on channel 3 and reply with a RREP packet. In contrast to traditional reactive protocols the RREP is opportunistically forwarded back to the originator of the RREQ (Section 3.4). This is important since otherwise a simple source routed unicast packet could encounter a 'deaf' node on the next hop. Furthermore it is not always possible to forward the RREP along the inverted route (collected by the RREQ) back to the originator. The reason is that we cannot assume that all links are symmetric [1].

Moreover, the approach leads to another problem. Again, consider our example from Figure 4. As described above node F is the destination of the RREQ. Imagine furthermore that the RREQ has taken the route from A over B and D to F (A→B→D→F). Now node F could decide to forward the packet along node E. In this case node E will be responsible for further forwarding the RREP packet towards A. In this simple example this would not be a problem for node E, since A and E are direct neighbors. However, this is generally an exception and node E has to find a route towards A. This could lead to a new route discovery process. To avoid cascading route discoveries each opportunistically forwarded RREP packet contains a so called fallback route being the shortest path from sender to destination (e.g. node F to A). In our example this could be the route obtained from the received RREQ packet (A→B→D→F). The exact forwarding algorithm with fallback route is described in Section 3.4.4.

Both reactive and proactive versions of the route discovery protocol uses information obtained from LND. So it is not always necessary to start a route discovery for each destination. Sometimes the destination could be either a neighboring node or a neighbor of a neighboring node. So the route towards such a destination could be easily obtained from the LND algorithm.

Finally it is worth to mention that if multiple channels were used some optimizations are possible. In the case that a node has neighbors operating only on a small fraction of available channels a broadcast on all channels introduces unnecessary overhead. Instead, the broadcast should only consider channels used by neighboring nodes. This optimization is possible because the required neighborship information could be easily obtained from the LND algorithm.

3.3.2 Proactive Route Discovery

Basically, the proactive version of the MCExOR route discovery algorithm can be understood as the natural extension of the local neighbor discovery algorithm described in Section 3.2. The algorithm works as follows: Each node periodically broadcasts discovery packets containing his home channel as well as the link delivery probabilities obtained from the LND algorithm to its neighbors. On receiving such a packet each node stores the home channel of the node initiating the discovery, updates its link table and forwards this packet via broadcast on all available RF channels (flooding). Again, one optimization is that we only need to forward the packet on the RF channels of our neighbors. Eventually this discovery packet will be received by each node in the network. Additionally each node knows the home channel of all nodes in the network. As it will be shown later this information is essential for one of our candidate set selection algorithms (Section 3.4.3).

Finally it is important to note that as a result of the multi-channel support an additional delay is introduced by the route discovery process. However, in case of the proactive route discovery, where the flooding interval is around 30s a delay of some milliseconds could be safely ignored.

3.4 Packet Forwarding

If a node needs to send a packet to a certain destination it makes use of one of the route discovery algorithms described in the previous section. As a result the node is able to construct a set of candidates for the packet forwarding. Within this section, we address the problem of selecting a route and forwarding the packet along this path. The main idea of MCExOR as well as ExOR is to use a set of forwarding candidates instead of only a single forwarder. Especially in dense networks it is possible to construct many different candidate sets. With MCExOR the additional problem of choosing a transmission channel is introduced. We subdivided the mentioned problems into two tasks. At first, candidate sets for every RF channel are constructed and finally, the most promising candidate set along with its channel is selected for transmission.

3.4.1 Construction of Candidate Sets per Channel

The algorithm for the construction of a candidate set is similar to the one used by ExOR. Unlike ExOR, in MCExOR we have to construct a candidate set per channel. This becomes clear, because two candidates on different RF channels cannot communicate with each other. Our algorithm works as follows: At first the

cumulated expected transmission count for the current node and each neighbor towards the destination is calculated. Only neighbors with a better metric than the current node are further considered. Thereafter the candidates are grouped according to their home channels. A fixed number of candidates is chosen according to the ETX metric of the path from the current node to the final destination using the candidate as first hop. Finally the most promising candidate set is selected for transmission. There are two alternative algorithms which differ in their knowledge about the current network's state they use to make a decision. Both algorithms are presented in the following two sections.

3.4.2 Local Algorithm for the Selection of a promising Candidate Set

Unlike ExOR, in MCExOR we have to choose between a set of candidate sets. In general we can select between k candidate sets, where k is the number of different home channels used by neighboring nodes. Now the question arises which candidate set should be used? Consider the network illustrated in Figure 4, where node A needs to forward a packet to node F. Based on this network node A constructs the following two candidate sets (Figure 5):

- *(D,B)*, when channel 2 is used
- *(E,C)*, when channel 3 is used.

The size of the resulting candidate sets is two. If node C and E would also operate on channel 2, there would be only one candidate set of size four. If the maximum size of a candidate set is three for example, one candidate had to be removed from the set.

For the description of the algorithm for the selection of a promising candidate set we have to formulate our problem more precisely: A wireless mesh network is a collection of directed links connecting transmitters, forwarders, and receivers. Such a communication network may be represented by a directed graph $G=(V,E,f)$ with a vertex set $V=\{A_1,\ldots,A_n\}$ and an edge set $E\subseteq V\times V$. Further a non-negative number $f(e)$ is associated to each link $e\in E$, called the link delivery probability of e. Based on this graph we can define the expected transmission count $g(x,y,z)$ of the path from node x followed by y to destination z:

$$g(x,y,z)=\sum_{e\in sp(x,y,z)}\frac{1}{f(e)},\ \text{where}\ x,y,z\in V,(x,y)\in E$$

Furthermore $sp(x,y,z)$ calculates the shortest path $(e_1,\ldots,e_n)\in E^n$ in the network from node x followed by node y ($(x,y)\in e_1$) to z regarding to the link delivery probability f. In order to allow communication between two nodes in the network we define a flow c as $(u,v)\in V\times V$, where $u=source(c)$ is the source of c and $v=sink(c)$ is the sink of c.

Our algorithm $chooseCs(c,w,CSS)$ calculates to a given flow $c=(u,v)$, a forwarding node w and a set of available candidate sets $CSS:P(V^n)$ the candidate set with the lowest metric towards the destination node $v=sink(c)$, where $P(V^n)$ denotes the power set of V^n:

$$chooseCs(c,w,CSS)=cs\ \text{ where}$$
$$\forall cs'\in CSS: csm(c,w,cs')\geq csm(c,w,cs)$$

Whereas $csm(c,w,cs)$ associates a non-negative number to each candidate set cs selected by a node w to a given flow c, called the metric of the candidate set:

$$csm(c,w,cs=(c_1,\ldots,c_n))=\sum_{i=1}^{n}g(w,c_i,\text{sink}(c))\frac{pcs(w,i,cs)}{1-pncs(w,i,cs)}$$

Further $pcs(w,i,cs)$ calculates the probability that the i-th node in the candidate set cs will be the next forwarder when the packet is transmitted by node w, whereas $1- pncs(w,cs)$ is the probability that the packet was received by none of the nodes in cs:

$$pncs(w,cs=(c_1,\ldots,c_n))=\prod_{j=1}^{n}1-f(w,c_j)$$

$$pcs(w,i,cs=(c_1,\ldots,c_n))=f(w,c_i)\prod_{j=1}^{i-1}1-f(w,c_j)$$

In our example network of Figure 4 we have a flow from A to F ($c=(A,F)$). Node A would calculate:

$$csm(((A,F),A,(D,B)))=(2+2.5)\frac{0.4}{0.94}+(2+1.1+1.1)\frac{0.54}{0.94}=4.33$$

$$csm(((A,F),A,(E,C)))=(3.33+1.1)\frac{0.3}{0.79}+(1.43+1.1+1.1)\frac{0.49}{0.79}=3.95$$

$$chooseCs(c=(A,F),w=A,CSS=\{(B,D),(E,C)\})=(E,C)$$

Therefore node A would decide in favor of (E,C).This is true in the case of only a single RF channel, but in case of multiple channels some further work is required. The reason why MCExOR uses a strategy for channel assignment different from MCRP is the reduction of self-interference. If multiple packets are transferred along a path, self-interference between these packets reduces the number of simultaneously active transmissions. That's why MCExOR tries to minimize the use of identical RF channels along a path. To achieve this not only the metric of the candidates in the candidate set is considered, but also their RF channels. Reconsider the example of Figure 4. Imagine further that the nodes A, E and C operate on the same channel 1 and that node A receives a packet from a preceding node X on his home channel. In this case (E,C) is not a good choice, since a packet will be forwarded on the same channel (here 1) twice:

- From node X to A on channel 1
- From node A to (E,C) on channel 1

To avoid multiple successive transmissions on the same channel, each opportunistically forwarded packet contains the RF channels of the last j hops, where j is the number of available channels (node X_1 to X_j in Figure 5). If a packet is forwarded j-times on the same channel than the observed bandwidth is smaller than B/j. That's why we have to adopt our csm function accordingly:

$$csm'(c,w,cs,p)=uch(p,ch(cs))\cdot csm(c,w,cs)$$

A natural number $ch(cs)$ is associated to each candidate set $cs\in V^n$ which represents the home channel of the candidates. Further $uch(p,i)$ calculates how often the packet p was transmitted on channel i plus one.

Therefore node A would calculate:

$$csm'(((A,F),A,(E,C),p)=uch(p,ch(((E,C)))\cdot 3.95=2\cdot 3.95=7.9$$

So it makes sense to use *(D,B)* and therefore to transmit the packet on channel 2.

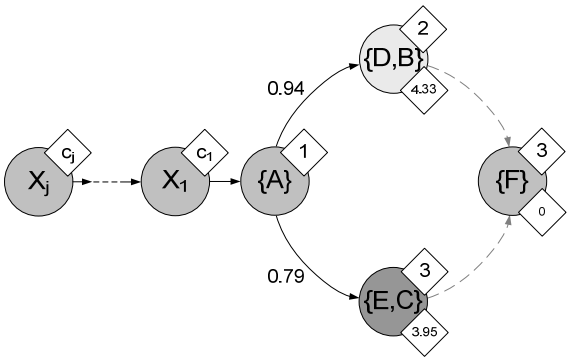

Figure 5: Local candidate set selection algorithm calculates the metric for each candidate set (4.33 for (D,B) and 3.95 for (E,C)). Furthermore the channel of each candidate set is displayed by the upper left number. The two numbers on the edges represents the probability that a packet is successfully received by at least one node in the candidate set. In addition to avoid multiple successive transmissions on the same channel j pseudo nodes are created (X_1 to X_j).

3.4.3 Look-Ahead Algorithm for the Selection of a promising Candidate Set

Sometimes it is possible that the algorithm described in the previous section makes suboptimal decision. We refer to it as the local candidate set selection algorithm. Reconsider the example illustrated in Figure 4. According to the local algorithm we would decide in favor of *(E,C)*. However this is not a good decision since the home channel of the final destination is 3: Packet forwarding by *(E,C)* would lead to two successive transmissions on the same channel (3). In other words: by selecting a candidate set we also define the RF channel for the next transmission.

To overcome the shortcomings of the local algorithm we extend the approach by a look-ahead. At first a so-called candidate set graph is constructed from the given network topology. The corresponding candidate set graph of the network in Figure 4 is illustrated in Figure 6.

The candidate set graph is constructed as follows: Candidate sets are modeled as nodes in the graph, where the node's outgoing edges represent the available candidate set choices. In node *(D,B)* we have to distinguish whether node D or B is responsible for the packet forwarding. In the first case the only available candidate set is *(F)*. In the second case two candidate sets *(F)* and *(D)* on channels 3 and 2 are available. The rest of the graph is constructed similarly for candidate set *(E,C)*.

Now the question arises how far do we have to make the look-ahead towards the final destination. In the optimum we have to construct to full candidate set graph for a given network. It is obvious that this approach is not feasible, because of the high node degree (fanout) of the constructed graph: if we have an average of k channels in the neighborship and the average size of the candidate set is m, than the resulting candidate set graph has a fanout of $(k \cdot m)$. So we use the following heuristic: Only the highest candidate in the candidate set is expanded. In the example of Figure 6 the nodes *(D)* and *(E)* will be ignored. Finally, the look-ahead is at most the number of available channels in the network. This becomes clear, since the basic idea behind this algorithm is to determine the fewest used RF channel for the next hop. The best performance along a route (path) could be achieved if all available channels are equally used. So it is important to know when an already used channel along a route could be reused.

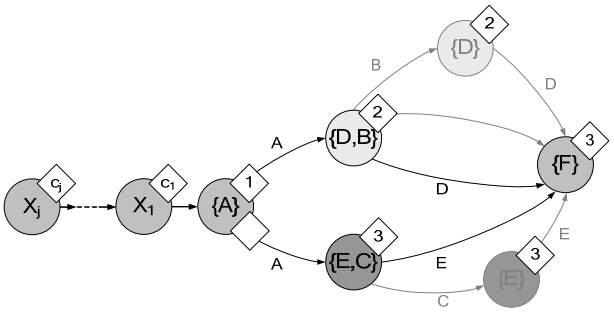

Figure 6: Candidate set graph for the network in Figure 4. The label on the edge between two candidate sets (nodes) represents the particular forwarder.

Practically speaking, each edge in the candidate set is further annotated with a metric which reflects not only the distance of the candidate set towards a destination, but also the used channel. In our example the metric on edge between nodes *(E,C)* and *(F)* would be increased two times because the channel 3 would we used twice. Finally a shortest path algorithm is used to find the best route in this graph. According to the look-ahead algorithm node A would decide in favor of *(E,C)*.

3.4.4 Packet Producer, Consumer, and Forwarder

In MCExOR we can identify three kinds of nodes: packet producer and consumer as well as packet forwarder (relay nodes). In general there are only few producers and consumers, but many forwarding nodes.

For the producer of a packet it doesn't make any difference which candidate of the selected candidates forwards the packet. It is only important that the packet makes 'progress' towards the destination. However, the last hop to the packet consumer requires a more sophisticated solution. It is possible that a packet consumer node is selected by other nodes as forwarder. In this case a node becomes a consumer and a forwarder at the same time. This could lead to the problem of 'deafness' since in general a node forwards a packet on a channel different from its home channel. This problem and our solution are addressed in more detail in Section 3.4.8.

All packet producers should prefer the use of candidate sets on their home channels. There are two reasons for this decision. The first is a technical one. For the originator of a packet flow it is hard to utilize the whole bandwidth due to the delay introduced by switching the channel for each packet which finally results in a reduced transmission rate. The other reason is that in general the originator of a packet flow is also a destination of a reversed packet flow (e.g. TCP/IP traffic). In this case a node becomes the producer as well as the consumer of packets. Preferring the home channel prevents the node from being 'deaf'. Otherwise packets destined for this node would fail on the last hop. This becomes clear, since this node would otherwise operate most of the time on a channel different from his home channel. Finally, pure

forwarding nodes are free to choose any candidates on any home channel.

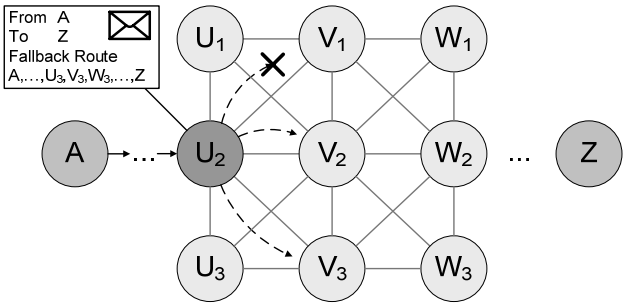

Figure 7: Example network with physical links shown as gray edges and dashed arrows indicating the forwarder's decision at node U_2 with the help of the fallback route.

3.4.5 Opportunistic Routing with Fallback-Route

As described in Section 3.3 we can freely choose between two kinds of route discovery algorithms: proactive and reactive. In case of the reactive route discovery we have to make some further restrictions on the selection of forwarding candidates. This is required to avoid cascading route discoveries due to the character of opportunistic forwarding. For example, it is not always possible to send a RREP along the inverted route of the associated RREQ due to the existence of asymmetric links. Furthermore it is very likely that an opportunistically transmitted data packet will be received by a forwarding node which has not a valid route to the destination. In both cases an additional route discovery would reduce the overall performance of the network.

Consider the network in Figure 7. Node U_2 needs to opportunistically forward the incoming packet towards the final destination Z. However, for the selection of a candidate set node U_2 cannot longer freely choose between all available options described in the previous sections. It is important that each selected forwarder in the candidate set is able to forward the packet towards the destination without initiating an additional route discovery process. That's why an additional restriction on the selection of a forwarder has to be made when the reactive route discovery algorithm is used: Each node in the candidate set has at least one neighbor which is listed in the fallback route with a better ETX metric towards the destination than the current node.

This restriction together with the information obtained from the fallback route and the local neighbor discovery algorithm (Section 3.2) guarantees that each forwarder is able to calculate the next forwarder without possibly initiating a new route discovery process. In our example node U_2 will make a decision in favor of node V_2 and V_3. Both nodes have neighbors listed in the fallback route with a better metric than U_2 (e.g. V_3 and W_3 respectively). Node V_1 cannot be an option, because none of his neighbors are listed in the fallback route.

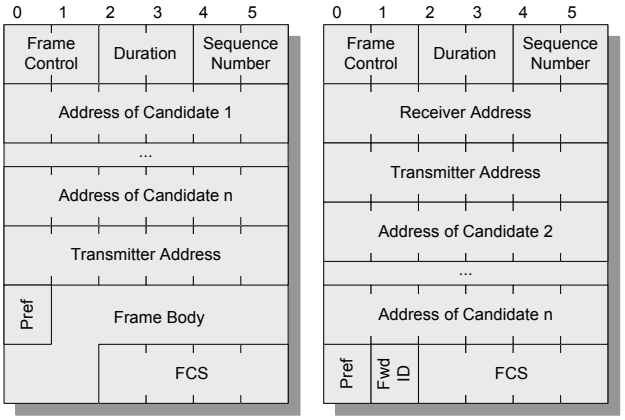

Figure 8: Format of MCExOR data and acknowledgement MAC frames.

3.4.6 MAC Layer Framing and Transmission

We use an IEEE 802.11-like Medium Access Layer, extended by the capability of switching RF channels and processing slotted acknowledgements. The following paragraphs cover MCExOR framing as well as the details of data transmission.

The structure of MCExOR MAC data and acknowledgement frames is shown in Figure 8. The *Frame Control* field is similar to the corresponding field in the IEEE 802.11 MAC. The first byte identifies the *type* of the frame (MCExOR data or acknowledgement) and the remaining 8 bits are reserved for control information. MCExOR only uses the *Retry* bit to mark retransmitted packets. The 2 byte *Duration* field is used to update the network allocation vector. It contains the number of microseconds the medium is expected to remain busy for the currently active transmission. Both data and acknowledgment frames carry the *Sequence Number* of the data. In combination with the sender of the data (*Transmitter Address* for data or *Receiver Address* for acknowledgement frames, respectively) it identifies the data packet and is used to track duplicates. The *Frame Check Sequence* is similar to the corresponding IEEE 802.11 field for both packet types.

The data frame contains the *Transmitter Address* and the *Addresses of all Candidates*. It is assumed that the maximum number of candidates is fixed and does not change during operation. The *Pref* field holds the forwarding preference of the transmitting node (only 4 bits used) and is followed by the payload (*Frame Body*). An acknowledgement frame holds the *Receiver* and *Transmitter Address*. Additionally it carries the *Addresses of all Candidates* except the highest prioritized one. The index of the forwarder is stored in the *Fwd ID* field. The forwarding preference of both the transmitting and forwarding node are stored in the *Pref* field (4 bit each).

The packet transmission starts with a channel switch, if necessary. Within this period, the node is deaf (Section 3.1). After the RF hardware proceeded the channel switch, the state of the MAC is reset (back-off, collision window, retry counter, mode, etc.). The network allocation vector (NAV) is not reset because of the risk of collisions. Instead the MAC tries to adapt the NAV to the new channel through advancing the NAV by the transmission time of the maximum fragment size. So the packet is not sent until the NAV is updated. After that, it is annotated with source address

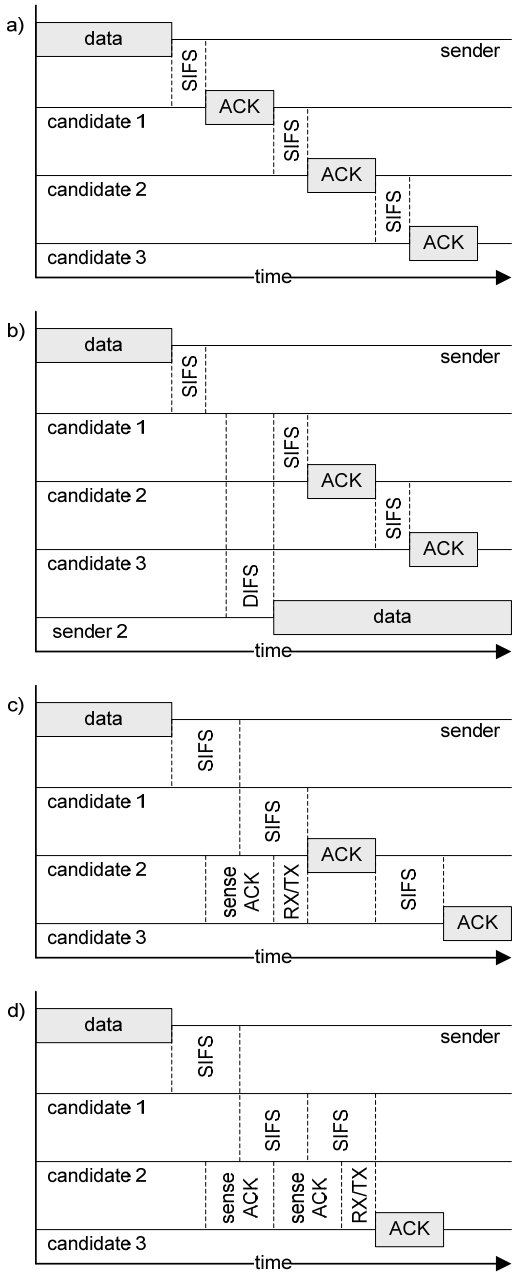

Figure 9: a) Slotted acknowledgement with three candidates. b) Slotted acknowledgement with the first ACK missing. Subsequent ACKs collide with a data transmission started within the delay of the missing ACK. c) Compressed slotted acknowledgement with first ACK missing. d) Compressed slotted acknowledgement with first and second ACK missing.

and in general multiple destination addresses taken from the candidate set and it is transmitted via the wireless medium.

Slotted acknowledgements [2] determine which candidate forwards the packet. They are a generalization of the link level acknowledgements of the IEEE 802.11 protocol. Every candidate sends an acknowledgement packet (ACK). The highest prioritized candidate sends the first ACK with a delay of *SIFS* after the data packet was received. The other candidates send their ACK in order of decreasing priorities, each separated by *SIFS*. A slotted acknowledgement with 3 candidates is depicted in Figure 9a. The ACK packet additionally contains an identification of the highest prioritized candidate which did successfully receive the packet. Therefore the MAC maintains a forwarder field which is initialized with the own identification. If a candidate receives an ACK, the forwarder identification is extracted and stored in the forwarder field, if the announced forwarder has a higher priority. With reaching the assigned time slot the candidate sends the ACK with the previously determined forwarder. This way the ACK packets propagate in a multi-hop fashion from the highest prioritized candidate to the sender. So it is possible to use low-quality and asymmetric links for data transmission.

A serious problem arises with the usage of slotted acknowledgements. In the traditional IEEE 802.11 MAC the ACK is sent after a delay of *SIFS*. Since the ACK packet has a constant size the initial sender could determine whether to retransmit the packet after a fixed delay. Another node willing to transmit a packet has to sense the medium for a period of *DIFS* which is larger than *SIFS*. Thus the contention based medium access does not allow that another node starts to send a packet within the delay between data reception and ACK transmission. But using the slotted acknowledgement the mentioned problem may occur if a candidate misses the data packet and does not send an ACK. Since an ACK packet is larger than *DIFS* another node may experience an idle medium and decide to start a transmission which will collide with subsequent ACK packets. The described scenario is depicted in Figure 9b. Virtual carrier sensing does not solve the problem because the node that is willing to send could have missed the data packet. Therefore it is not able to update its NAV accordingly.

We address this problem by refining the presented mechanism [2] to a compressed slotted acknowledgement. The main idea is the following. If a candidate detects that an ACK from a higher prioritized candidate is missing, it prematurely sends its ACK. This way spaces where the medium is idle are kept smaller than *DIFS* (for a candidate set of a fixed size). In order to prevent collisions, the points in time when a candidate prematurely sends its ACK are ordered by decreasing priority.

The compressed slotted acknowledgement works in the following way. With a delay of *SIFS* after the data packet was received the highest prioritized candidate sends the ACK packet. From that point in time all other candidates wait for the period $P = SIFS - RX/TX$ whether they 'hear' the recently sent ACK (The receive/transmit turnaround *RX/TX* delay occurs when the radio turns from receive to transmit mode. Within this period, the node is deaf.) Because not all candidates necessarily receive this ACK, we use signal strength as an indicator. If within the waiting period the signal strength did increase significantly, the ACK packet is considered as sent (It is not necessary that the candidate successfully receives the packet.) On the other hand if no such increase in signal strength is observable, the other candidates conclude that the highest prioritized candidate did miss the data packet. In that case, the second highest prioritized candidate starts to transmit its ACK prematurely. The radio switches from receive to transmit within delay *RX/TX*, so the ACK is sent *SIFS* after the expected ACK and *2*SIFS* after the data packet was received. Up

from this point the acknowledgement process is continued like in the no-error case, except that all subsequent events happen earlier. The previously described scenario is illustrated in Figure 9c. However, it is also possible that the two highest prioritized candidates miss the data packet. This 'two-missed' scenario from Figure 9d starts like the 'one-missed' case above: After data packet reception all other candidates wait for the period *P* and note that the highest prioritized ACK is missing. So they wait for *SIFS* to hear the next (premature) ACK. But this ACK is also missing since none of the two highest prioritized candidates received the data packet. So the third node in decreasing priority order sends the ACK. After turnaround from receive to transmit the radio finally sends the ACK packet *3*SIFS* after the data packet was received. All other 'many-missed' cases are analogously. For a candidate set size of less than six the medium is idle for not longer than *DIFS*.

3.4.7 Dealing with Duplicates

The selection of a forwarder relies on a distributed agreement among the sender and the candidates. But there is no guarantee that all nodes agree on the same outcome. The links between candidates are not necessarily perfect and may vary over time. If, for example, a lower prioritized candidate does not receive a higher prioritized ACK, the candidate may wrongly advertise itself as forwarder. So a duplicate is created because two nodes forward the data packet (multiple-forwarder duplicate). Another reason for duplicate creation is the retransmission of a data packet because no ACK reached the sender node (retransmission duplicate). Figure 10 depicts typical scenarios which result in packet duplication. In every sub-picture node A sends a packet using the candidate set (C, B) and C as highest prioritized candidate. In Figure 10a all transmissions proceed without error and eventually C forwards the packet. Figure 10b shows the creation of a multiple-forwarder duplicate. Node B does not receive the ACK from C and vice versa, so both advertise themselves as forwarder and relay the packet. In Figure 10c, both B and C receive the data packet, but their ACK packets do not reach the sender A. So node A retransmits the packet and node B and C are able to detect the duplication. Figure 10d illustrates a more complex scenario where both retransmission and multiple-forwarder duplication occur.

It turned out that duplicates are a serious problem for the opportunistic forwarding, since they waste bandwidth and therefore lower the effective throughput. On the other hand the usage of the ETX metric produces paths with not only high quality links, but also poor links with a low delivery probability. So using the best path according to that metric does not mean that there are no retransmissions and duplicates. We are using different techniques to reduce the number of duplicates. The IEEE 802.11 MAC keeps track of duplicates by using sequence numbers. For every successfully received packet the sequence number along with the MAC address of the sender are stored and used to identify subsequent data packets as duplicates. This way duplicates due to retransmission are recognized. We extended this mechanism in a way that every node aggressively tracks sequence numbers. The IEEE 802.11 MAC considers only information of data packets, whereas the MCExOR MAC tracks the sequence number of every packet, regardless if it is data or ACK. So a candidate is able to identify a retransmission as duplicate even if he missed the data packet and only received a preceding ACK.

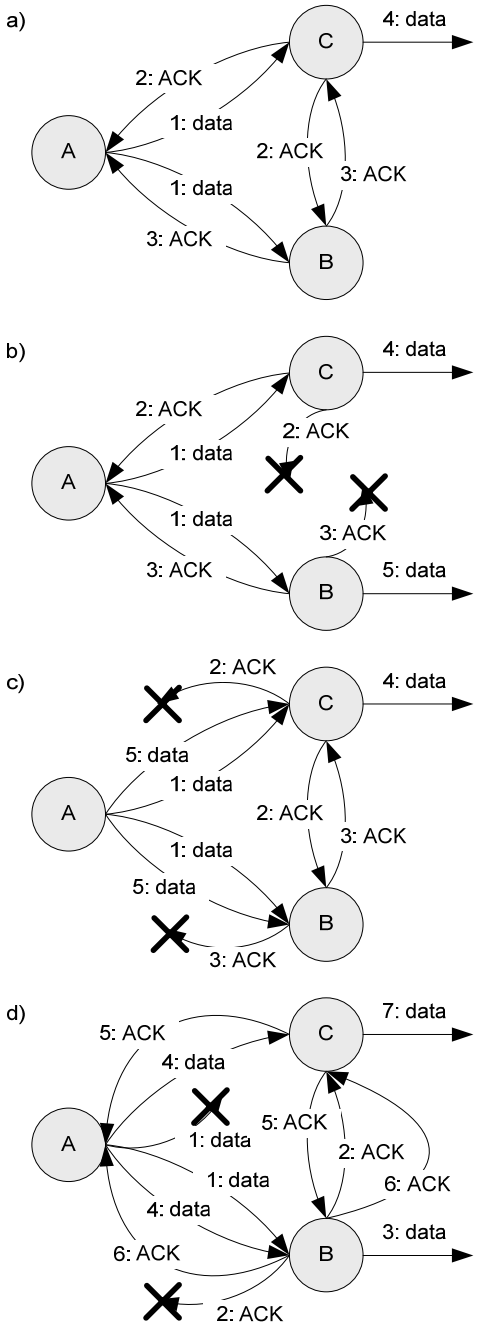

Figure 10: a) Slotted acknowledgement with 2 candidates. b) Slotted acknowledgement where the candidates do not receive their ACKs, so that both forward the packet (multiple-forwarder duplicate). c) Slotted acknowledgement where no ACK reaches the sender A. Thus node A retransmit the packet and B and C identify it as duplicate (retransmission duplicate). d) Slotted acknowledgement with both retransmission and multiple-forwarder duplicates.

This way node C in Figure 10d recognizes the retransmission of A as duplicate and does not forward the packet twice.

Multiple-forwarder duplicates as shown in Figure 10b still exist because sequence numbers are valid only for a single hop. We introduced a duplicate detection mechanism for the routing layer. A sequence numbers is assigned to every data packets on the routing layer on the initial sender node and is not changed along the routing path. Analogously to the MAC the MCExOR routing layer stores sequence number and source identification of incoming packets and uses them to track duplicates on the destination node.

We further use passive acknowledgements to reduce duplicates due to retransmission. After a node has sent a packet, it waits for the corresponding ACK. If the ACK gets lost on the way back, the sender will process an exponential back-off and try again several times. During the back-off process it is possible that the sender recognizes that one of its candidates forwards the data packet. In this case the sender assumes that the candidate has successfully received the data packet and only the ACK got lost and therefore cancels the retransmission.

A candidate may not have received the data packet but instead a corresponding ACK. The ACK packet contains the MAC addresses of all candidates, so a node can decide whether it was a candidate within the transmission of the data packet. If so and its transmission slot has not already past, it takes part in the acknowledgement process despite the fact that it did not receive the data. These additional ACK packets increase the probability that the sender and other candidates take note of the actual forwarder and therefore prevent duplicates.

3.4.8 Relay Preferences

With ExOR only the sender selects the candidates. If there are multiple flows in the network, an exclusive sender-based selection is problematic. Figure 11a illustrates a scenario where the sink of the first flow (node E) becomes forwarder for packets of the second flow. It is very likely that a forwarder node has to change the channel, so node E has to change from its home channel 3 to channel 4. Packets from the first flow could not be delivered on the last hop (from A and D to E) as long as E is occupied with relaying. Even worse, nodes A and D do not get any ACK packets, so they process an exponential back-off. The achievable throughput of the first flow decreases.

However, exclusive sender-based forwarder selection also results in a poor performance in the case that an intermediate node becomes forwarder for packets from multiple flows like node G in Figure 11b. Simulations have shown that under high load these nodes are not able to forward packets from two flows with the same rate as nodes serving only one flow. Often these nodes became the bottleneck because they were not able to forward packets at the same rate as they arrive, so their packet queue did grow. A better schedule would be achievable if not only the sender, but also the candidate could influence the forwarder selection. This way the nodes B and D would get a feedback from G and B could decide to use E instead of the congested G.

There are different alternatives for a sender and receiver based forwarder selection. In a simple solution a congested node does not acknowledge new packets until its queue length drops under a predetermined limit. But this solution has some obvious drawbacks. The sender is not able to distinct between congestion and packet loss. So the congested node will be used as candidate as long as its ETX metric does not change, but independent from its acknowledgement. Thus some opportunities are wasted, like using node E instead of G in Figure 11b.

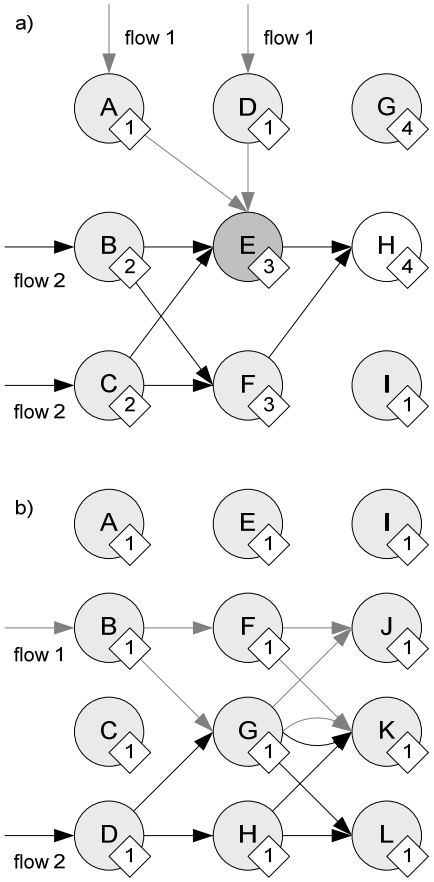

Figure 11: a) Part of a network (channel of every node is annotated) with two flows having the destinations E and H respectively. Node E relays packets for flow 2 on channel 4 and is deaf within this period. b) Two flows sharing nodes G and K as forwarder.

We realized a combined sender and receiver based forwarder selection using relay preferences. The relay preference is the willingness of the node to forward packets. It is influenced by the number of flows originate or terminate on the node and its congestion quantified by queue sizes. An idle node has the maximum forwarding preference. If it becomes source or destination of a packet flow or its packet queue grows, it lowers its preference. Analogously the preference is increased in the opposite cases. Every node propagates its current preference in all data and ACK packets it sends. Other nodes maintain a preference table for all its neighbors and update it accordingly whenever they receive a packet with annotated forwarding preference. Furthermore, the ACK packet contains not only the index but also the preference of the forwarder. The slotted acknowledgement is modified in a way that not the initial ordering of candidates is the solely criterion for the forwarder selection. The forwarding preference announced in each ACK packet is most important. The

sender-based candidate set ordering is considered afterwards to break ties. So if a less prioritized candidate receives an ACK from a higher prioritized one and notices that it has a higher preference value than the forwarder placed in the ACK packet, it announces itself as forwarder. In this way it can prevent the highest prioritized candidate from sending the packet.

Relay preferences can be used to suppress duplicates. If a candidate identifies the current data packet as duplicate, it acknowledges the reception with a symbolic preference value which indicates the highest preference and is not used otherwise. So it becomes the forwarder and discards the duplicate. On the other hand relay preferences increase the risk of duplicates. If a higher prioritized node does not receive the ACK from a lower prioritized one with a higher preference, both nodes will forward the packet and hence, a duplicate is created. But changes in routing preference of a neighbor are immediately reflected in routing decisions, so that the number of resulting duplicates is kept acceptable small.

By using relay preferences both scenarios from Figure 11 can be improved. In Figure 11a node E recognizes that it is the sink for flow 1 and lowers its preference accordingly. Node B and C update their neighbor preference table when they receive an ACK from E. Because the preference of E is lower than the preferences of all surrounding nodes, potential senders will try to find a candidate set without E. If it is not possible to replace E, at least the priority of the node is lowered so that it is not the highest prioritized candidate anymore. In Figure 11b node G lowers its relay preference when it gets congested. So both senders B and D will try to avoid using G as candidate when they recognize the preference change. The result could be that node B changes its candidate set from {F, G} to {F, E}. An interesting observation is that relay preferences are a simple means of dynamical load sharing.

3.4.9 Hot potato routing

An interesting question arises when the packet reaches its last hop and the destination node is the forwarder with the highest priority. If no other candidates are added to the candidate set besides the destination then it is not possible to use the opportunistic nature, i.e. the protocol degrades to a traditional protocol like AODV. So the candidate set is filled up with additional nodes which have a routing metric equal to or better than the sender. If the destination node is not able to receive the packet, possibly another candidate receives and forwards it. If there are no other candidates, the sender would experience a timeout while waiting for the acknowledgement and start the back-off process. By adding other candidates the back-off is avoided.

On the other hand, the risk of cycles in the routing path is increased through this approach. If another candidate and not the final destination receives the packet, it will also put additional candidates into its candidate set. This way it is possible that two or more candidates exchange the packet in a 'hot potato' fashion and the packet is within a loop, because the relation between distance and link delivery probability is not linear. There are often no significant differences in link quality for links between nodes within small distances. Fortunately, TTL fields introduced by the routing prevent the packet from looping forever. Furthermore, in this particular case the final destination is always the candidate with the highest priority. So the probability of cycles is reduced. A packet only loops if the final destination is not able to receive it, e.g. if it is deaf. In this case the hot potato routing even saves unnecessary back-offs.

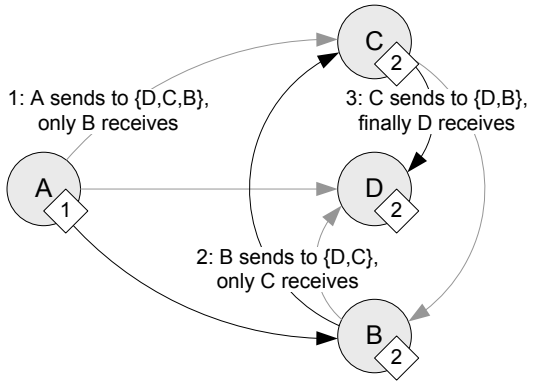

Figure 12: Node A relays a packet for the destination D. Since A, B and C all use multiple candidates the packet is forwarded in a hot potato fashion.

Furthermore, deafness is reduced in the case of multiple channels. In the example of Figure 12 node A tries to transmit a packet to destination D on a channel different from its own home channel. Without hot potato routing it would send the packet unicast to D. Node D does not receive the packet multiple times, so A had to stay on a different channel to perform the back-offs and retries and is deaf for transmissions on its home channel. By using hot potato routing node A uses multiple candidates having a common home channel. So it is able to deliver the packet to the destination D, in the best case, or to another node with D's home channel. After A has successfully relayed the packet, it switches back to its home channel and is able to receive further packets. Furthermore, the candidates B and C operate on the home channel of D. They do not need to change the channel in order to forward the packet to D, so no additional deafness is introduced.

4. Simulation

We implemented a prototype of MCExOR using the JiST/SWANS [12] wireless network simulator. The following sections cover implementation details, measurement methodology and a theoretical upper bound for the one-hop distance gain. Our outcomes show that MCExOR outperforms traditional protocols like AODV by the simultaneous use of multiple RF channels. In conjunction with realistic radio propagation models (shadowing) a further increase in the throughput is observed due to the opportunistic feature of MCExOR. With increasing number of channels the observed overall throughput superproportionally increases. Furthermore, even a single flow can benefit from the existence of multiple channels.

4.1 Implementation Details

JiST/SWANS offers two radio propagation models: free space and two-ray ground. Both models are based on the assumption that the received signal power is a deterministic function of the node's distance. According to the free space model the received signal power Pw_r depends on the transmission power Pw_t, wave length λ and distance d in the following way:

$$Pw_r^{dB}(d) = Pw_t^{dB} + 10 \cdot \log_{10}\left(\frac{\lambda}{4 \cdot \pi \cdot d}\right)^2$$

Therefore nodes within the communication range of a transmitting node always receive the packet. On the other hand, if a node is not within the communication range, it will never receive the packet. So the delivery probability of every wireless link within that range is one and for all links with greater distances is zero.

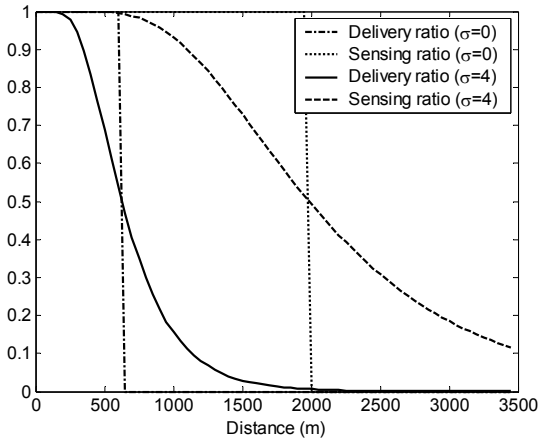

Figure 13: Delivery and sensing probability versus distance according to the free space and shadowing model.

A more realistic propagation model is shadowing [17]. The received signal power is modeled as log-normal distributed random variable. The mean of the distribution is determined by one of the propagation models mentioned above[1] and the standard deviation depends on the environment. For an outdoor environment typical values are from 4 to 12 dB. Whether a wireless card is able to receive or sense a packet depends on the reception threshold and sensitivity of the radio. Using the parameters from Table 3 the packet delivery and sensing probabilities are calculated in the following way (Φ denotes the standard normal cumulative distribution function):

$$P_{deliv}(d) = 1 - \Phi\left(\frac{-81dBm - Pw_r^{dB}(d)}{\sigma}\right)$$

$$P_{sens}(d) = 1 - \Phi\left(\frac{-91dBm - Pw_r^{dB}(d)}{\sigma}\right)$$

The resulting probabilities for standard deviations of 0 and 4 dB are displayed in Figure 13, whereas the former case corresponds to the non-probabilistic free-space model. We implemented the shadowing model in JiST/SWANS using free space propagation and a configurable standard deviation.

Furthermore, JiST/SWANS does not support multiple RF channels. We realized a simple multi-channel radio without cross-channel interference. The radio is extended by a fixed number of RF channels. Switching from one channel to another is possible within a fixed delay. Within this period of time the radio is not able to process any packets, so the node is deaf.

We used the IEEE 802.11 MAC implementation in JiST/SWANS as a starting point for our MCExOR MAC. The RTS/CTS mechanism was completely removed. We made modifications in order to realize the compressed slotted acknowledgement in combination with multi-channel support. Table 3 lists the parameter values we used.

The network layer is a slightly modified IPv4 implementation as offered by JiST/SWANS. We introduced a jittering between network and medium access layer in order to reduce the problem of simultaneous transmissions in discrete simulators.

Address resolution (ARP) and reverse ARP (RARP) are realized in a straightforward fashion. A lookup table of IP to MAC addresses is prepared in advance and made public to all nodes. So ARP operations are processed locally without network communication, because this paper aims to investigate the performance of the MCExOR protocol. An efficient realization of ARP without relying on broadcast communication is kept for future work.

Simulation parameter	Value
Propagation model	Shadowing, Free Space
Path loss exponent β	2.0
Shadowing standard deviation σ	0..4 dB
Communication data rate	1 Mbit/s
Transmission strength	15 dBm
Radio reception sensitivity	-91 dBm
Radio reception threshold	-81 dBm
Radio receive/transmit turnaround time	5 µs
Radio frequency	2.4 MHz
Signal to noise ratio	10
Radio channel switch turnaround time	80 µs
Slot time	20 µs
SIFS	10 µs
Retry limit	7
Collision window	31..1023

Table 3: Simulation parameters

4.2 Deriving a theoretical upper bound for the one-hop distance gain

In dense networks a sending node generally has many potential candidates, so there is the problem of choosing the optimal candidate set. In this section we investigate the influence of the candidate selection on the performance of opportunistic protocols like MCExOR and ExOR. In order to assess selected candidate sets we introduce the one-hop distance gain metrics *dg(d),* which is the expected increase in distance for the transmission over a single hop. The optimization of the metrics results in a theoretic upper bound for the one-hop distance gain which could be used to estimate the expected hop count. Our results are based on the assumption of a probabilistic shadowing propagation mentioned in section 4.1. Furthermore we only consider the static case, i.e. we do not consider semi-dynamic and dynamic parameters like ETX ratings and queue length. We also do not consider retransmissions, back-offs and related implementation details.

[1] The reference distance d_0 [13] has no influence because of a path loss exponent of 2.

Thus our solution is an upper bound and no solution to the scheduling problem.

$$dg_{AODV}(d) = P_{deliv}(d) \cdot P_{deliv}(d) \cdot d$$

At first we derive the one-hop distance gain for AODV dg_{AODV} in the continuous case, i.e. the position of candidates could be freely varied in the three-dimensional room. Using the assumption of omnidirectional radio propagation the optimal candidates are located on the vector from the sender to the final destination. Therefore all other cases are reducible to the one-dimensional case and are not considered further. By using AODV the candidate selection problem is to select exactly one candidate at the distance d which maximizes the expected distance gain $dg_{AODV}(d)$. The one-hop transmission is considered successful if both the data and acknowledge packets are delivered successfully. So the expected distance gain $dg_{AODV}(d)$ is calculated as product of distance d and delivery probability of the data and acknowledgement packets $P_{deliv}(d)$. The optimal distance gain for AODV with the parameters $\beta=2$ and $\sigma=4$ for the shadowing model is 283m and is achieved using a node distance of 367m.

Using MCExOR and ExOR the slotted acknowledgement has to be considered in the calculation of the distance gain. Depending on which candidate has received the packet there are in general multiple ways an acknowledgement could be delivered to the sender. Consider the scenario with three candidates in Figure 14 where node 0 sends a packet with candidates 3, 2 and 1. The packet delivery probability of a link from node x to node y is named $p_{x,y}$ and its complement is $\overline{p_{x,y}} = 1 - p_{x,y}$.

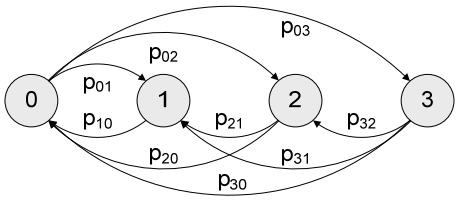

Figure 14: Nodes 0 sends a packet with candidates 3, 2, 1. The packet delivery probabilities are annotated on the edges.

For a successful transmission several cases have to be taken into account. If for example the highest prioritized candidate 3 in Figure 14 receives the data packet, the acknowledgement could flow directly to 0 or traverse nodes 2 or 1. The distance gain dg for the usage of n candidates is calculated as product of data and acknowledgement reception probabilities P^{data} and P^{ack}, weighted by distance d and summarized over all candidates:

$$dg(d_1,...,d_n) = \sum_{j=1}^{n} P_{j,n}^{data} \cdot P_{j,n}^{ack} \cdot d_j$$

We assume without loss of generality that the candidates are ordered by ascending priority, i.e. the lowest and highest prioritized candidates are nodes *1* and *n*, respectively. The data reception probability P^{data} of candidate j is the probability that the node j and no higher prioritized candidate $j+1 \ldots n$ receives the data packet.

$$P_{j,n}^{data} = p_{0,j} \cdot \prod_{k=j+1}^{n} \overline{p_{0,k}}$$

The acknowledgement reception probability P^{ack} is determined by the following recursive definition. The probabilities of all different paths the acknowledgement packet could use to reach the sender are summarized, where all non-direct paths are recursively calculated using the acknowledgement reception probability of the associated intermediate node. The recursion terminates for $j=0$ with a probability $P^{ack}=1$.

$$P_{j,n}^{ack} = \sum_{i=0}^{j-1} p_{j,i} \cdot P_{i,n}^{ack} \cdot \prod_{k=0}^{i-1} \overline{p_{j,k}}$$

$$P_{0,n}^{ack} = 1$$

Figure 15 illustrates the derivation of the acknowledgement reception probability on a decision tree for the candidate 3. There are four different paths from candidate 3 to the sender 0 (marked dark gray). The different recursion levels are displayed as pyramids.

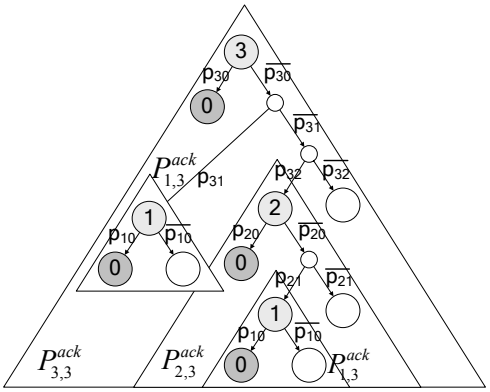

Figure 15: The derivation of the acknowledgement reception probability P^{ack} illustrated on a decision tree for candidate 3 (n=3).

For a number of two and three candidates the expected distance gain could be calculated using the following equations. With 3 candidates and shadowing parameters $\beta=2$ and $\sigma=4$ the maximum distance gain is 535m which is higher than with using 2 candidates (407m) and 1 candidate/AODV (283m).

$$dg(d_1,d_2) = d_2 \cdot p_{0,2}(p_{2,0} + \overline{p_{0,2}} p_{2,1} p_{1,0}) + d_1 \cdot \overline{p_{0,2}} p_{0,1} p_{1,0}$$

$$dg(d_1,d_2,d_3) = d_2 \cdot \overline{p_{0,3}} p_{0,2}(p_{2,0} + \overline{p_{0,2}} p_{2,1} p_{1,0}) + d_1 \cdot \overline{p_{0,3}}\,\overline{p_{0,2}} p_{0,1} p_{1,0} +$$
$$d_3 \cdot p_{0,3}(p_{3,0} + \overline{p_{3,0}}(p_{3,1} p_{1,0} + \overline{p_{3,1}} p_{3,2}(p_{2,0} + \overline{p_{2,0}} p_{2,1} p_{1,0})))$$

$$dg_{\max}(461,352) = 407$$

$$dg_{\max}(756,571,299) = 535$$

An interesting observation about this result is that the placement of the candidates depends on the number of candidates. By increasing the number of candidates (e.g. from 2 to 3), another candidate with a greater distance is added (e.g. a candidate in distance 756m). But the position of the previous candidates also changes (e.g. the last candidate moves from 352m to 299m). It turned out that the reason for that is the slotted acknowledgement. Without considering the acknowledgement reception probability in the calculation of the distance gain, the position of candidates does not change when increasing the number of candidates.

A further observation is the increase in distance gain when increasing the number of candidates. The distance gain is almost doubled when using three instead of only one candidate. This outcome indicates that the expected hop count decreases with using MCExOR instead of AODV. It has also a positive effect on the achievable throughput, since an increased distance gain means that fewer transmissions have to be made to deliver the data packet from the initial sender to the final destination.

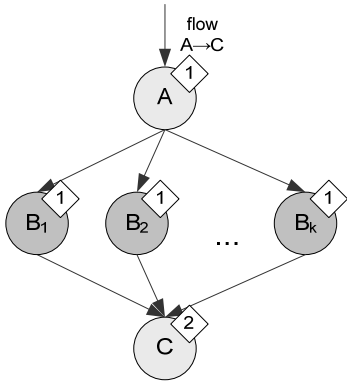

Figure 16: Simple network with one packet flow from node A to C demonstrates the influence of the candidate set size on the overall throughput.

4.3 Methodology

The goal of our experiments was to analyze the performance of the MCExOR routing protocol and to compare it with ExOR and a traditional protocol like AODV. The simulation scenario consists of a grid of nodes. Within a field with a fixed dimension the nodes were regularly placed using a fixed density. In case of MCExOR the radio channels were uniformly assigned to all nodes. We used a simple communication model for our simulations with a constant number of traffic flows. The source and destination of a flow are placed on the left and right borders of the grid. The flows are uniformly distributed in the horizontal dimension of the grid. We used constant bit-rate UDP traffic with packet sizes of 1400 bytes.

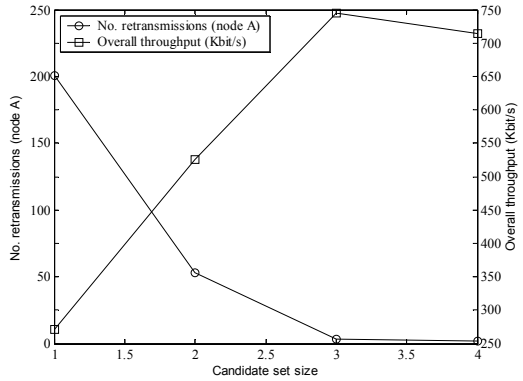

Figure 17: The diagram shows the impact of the candidate set size on the throughput of the packet flow of the network in Figure 16.

We identified the following metrics to compare the protocols with each other. The *route discovery latency* is the latency from sending out the route request until the first route reply arrived. The *throughput* is the ratio of the number of received bytes to the used time. The *packet delivery ratio* is the number of successfully received packets in relation to the number of sent packets. The *number of transmissions* summarizes the necessary transmissions of a packet until the final destination is reached or the packet is discarded. Accordingly the *number of retransmissions* covers all transmissions in the case that the first transmission was not successful completed. Further we count *duplicates* which could occur using opportunistic forwarding. Finally, the *number of hops* is the average number of transmissions from the source to the destination along a route.

4.4 Results and Discussion

4.4.1 Deafness

In this section we present simulation results regarding the 'deafness' problem. How does MCExOR solve this problem? Instead of choosing the next forwarder, MCExOR selects a set of potential forwarding candidates to reduce 'deafness'.

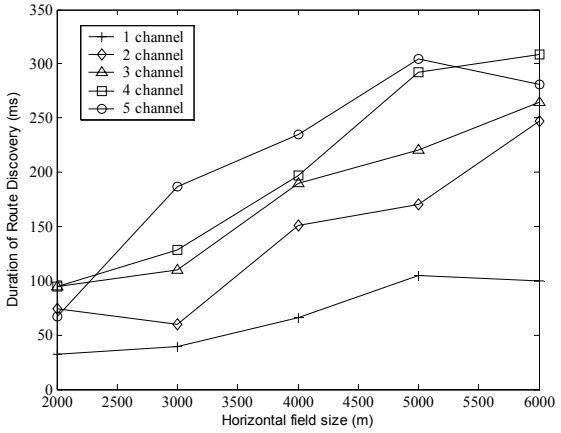

Figure 18: Delay in the reactive route discovery due to the support of multiple channels (example network with constant vertical field size of 200m).

Consider the example from Figure 16. There is a packet flow from node A to C with the help of the forwarding nodes ($B_{1..k}$). The first hop (A to $B_{1..k}$) is on channel 1, whereas the last hop ($B_{1..k}$ to C) is on channel 2. Therefore each forwarding node ($B_{1..k}$) has to switch from channel 1 to 2 in order to transmit the packet to node C. During the transmission on channel 2 each forwarding node is 'deaf'. The idea behind MCExOR is that is very unlikely that all nodes in the candidate set are 'deaf' at the same time. The impact of the candidate set size on the overall throughput is depicted in Figure 17. With increasing number of candidates the throughput of the flow increases. It seems that at least 3 candidates are required. Additional candidates do not further increase the performance. Furthermore, 'deafness' results in a great number of retransmissions at node A.

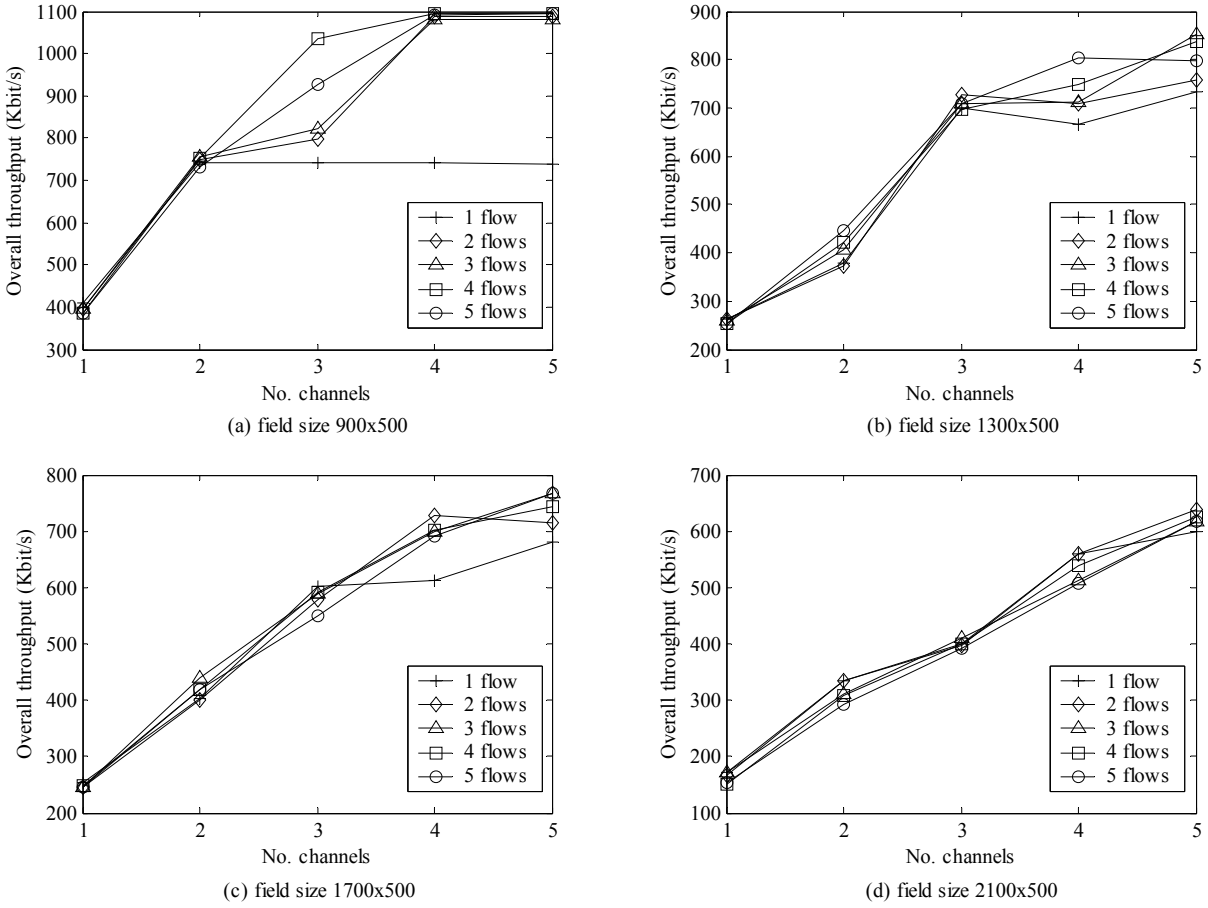

Figure 19: AODV (No. channels = 1) vs. MCExOR (No. channels = 2-5) with free space radio propagation model.

4.4.2 Latency due to Reactive Route Discovery

In section 3.3.1 we described the reactive route discovery algorithm used by the reactive version of MCExOR. The problem was that due to the multi channel feature there is some noticeable delay in the route discovery. Figure 18 presents the latency in the route discovery in regard to the number of used channels. With the increase of the number of channels the delay also increases. However the delay per channel does not increase with the number of channels.

4.4.3 AODV vs. MCExOR using Free Space Propagation

In this section we compare AODV with MCExOR. In order to show the multi channel advantage introduced by MCExOR we use the free space radio propagation model. With this radio model the advantage of being opportunistic only plays an inferior role. In a subsequent section we will compare AODV with MCExOR under more realistic conditions to show the opportunistic feature of MCExOR.

The results of our simulations are displayed in Figure 19. We selected the horizontal field size in a way that AODV has to make 2, 3, 4 and 5 hops on the average. The resulting sizes are 900m, 1300m, 1700m and 2100m, respectively, with a vertical dimension of 500m. Furthermore, we used the non-probabilistic free-space radio propagation model. We varied the number of simultaneous horizontal traffic flows from 1 to 5. The AODV protocol uses only one channel, whereas the MCExOR protocol varies the number of channels from 2 to 5. Furthermore the influence of the number of flows on the overall throughput was measured.

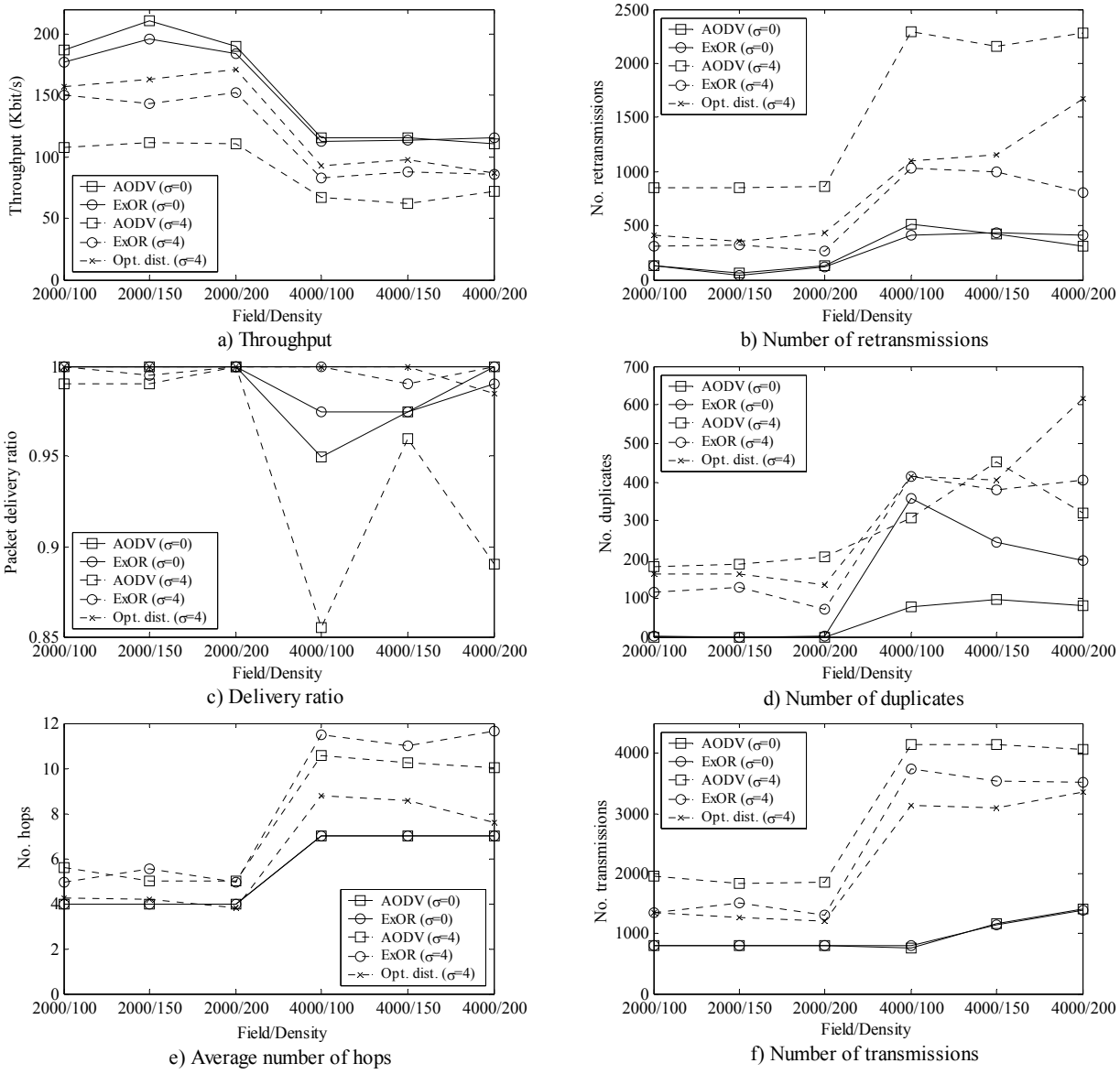

Figure 20: AODV vs. ExOR with Free Space (σ=0) and Shadowing Model (σ=4).

In the smallest network (Figure 19a) MCExOR scales with the number of channels and flows. Using more than 2 channels in the case of only one flow does not lead to an increase in the throughput. This is clear, because the packet route has an average length of 2 hops. In the network of Figure 19b a route has an average length of 3. That means that also a single flow can benefit from 3 channels. Finally in the largest network (Figure 19d) all 5 available channels could be simultaneously used by one flow. Again, in the smallest network (Figure 19a) one flow cannot benefit from a further increase in the number of channels above 2. However, by increasing the number of simultaneous flows the load imbalance among all available channels is reduced. In contrast to other approaches MCExOR assigns channels to nodes and not to flows. So also a single flow (e.g. Figure 19d) can benefit from the existence of multiple channels (here 5).

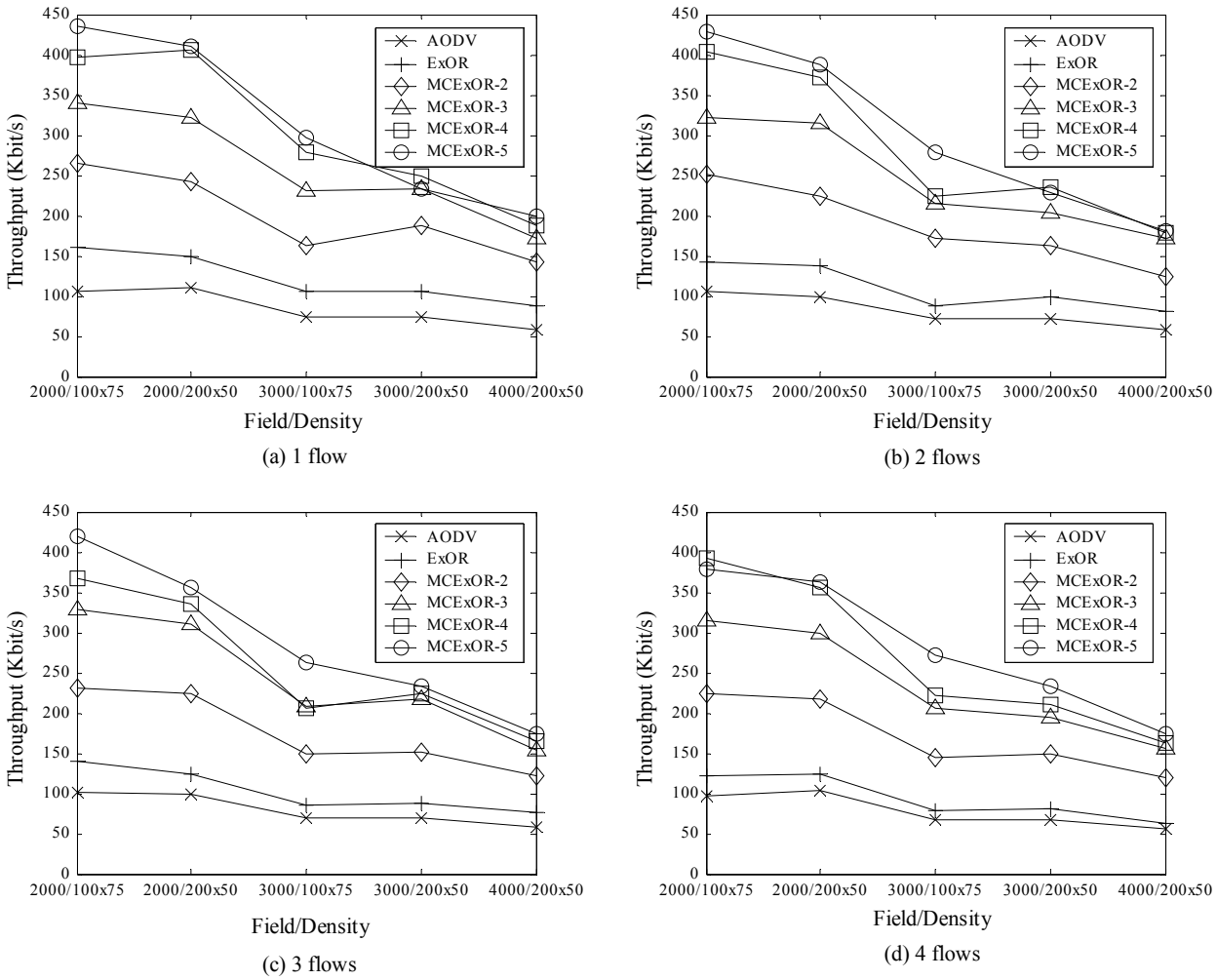

Figure 21: AODV, ExOR and 4 versions of MCExOR with 2, 3, 4 and 5 RF channels with shadowing model.

4.4.4 AODV vs. ExOR using Shadowing

The results of the measurements using AODV and ExOR and the free-space and the shadowing model are displayed in Figure 20. Although the focus of this paper lies on MCExOR we compare these two protocols because we could not use their original versions. AODV originally uses hop count as routing metric. However, hop count is not suited for the probabilistic radio model. Therefore we replaced hop count by ETX. Unfortunately the source code of the ExOR protocol is not publicly available, so we implemented our own version. Both protocols were used in its proactive versions, because we focus on packet forwarding instead of route discovery. Besides the two protocols we also used a candidate selection algorithm based on the node's positions. Therefore candidates where chosen which are closest to the analytically derived optimal distance (see Section 4.2).

The simulation took place on grids with the horizontal dimension of 2000m and 4000m and a constant vertical dimension of 300m. The nodes were regularly placed with a fixed vertical displacement of 75m. The horizontal displacement was varied between 100m, 150m and 200m. The abscissa of every diagram shows the used horizontal dimension and displacement. We used a single horizontal traffic flow. The ETX parameters were set to $\tau=4s$ and $w=100s$. Furthermore, we used a shadowing spread of $\sigma=0$ and $\sigma=4$, so the radio propagation corresponds to the non-probabilistic free-space and an outdoor shadowing scenario, respectively.

Figure 20a shows the achieved throughput for AODV and ExOR for both radio propagation models. Using free space propagation the performance of AODV is slightly better compared to ExOR. Due to the delivery probability distribution of the simple propagation model as depicted in Figure 13 there are no 'opportunities' for ExOR. It degrades to AODV with the additional overhead of slotted acknowledgement. Using shadowing the throughput of both protocols decrease, but ExOR outperforms AODV by 30%. Furthermore, the distance-based candidate selection increases the throughput by 40% compared to AODV.

Figure 20 b and f point out that the number of link-level transmissions and retransmissions of ExOR in the shadowing model is significantly lower compared to AODV. Using free

space propagation the differences in transmissions and retransmissions are marginal for both protocols. ExOR in combination with the shadowing model has also a higher packet delivery ratio than AODV, as depicted in Figure 20c. Another observation about ExOR is the increased number of duplicates compared to AODV especially on the larger grids caused by the slotted acknowledgement (Figure 20d). By using the probabilistic radio model there is always a probability that two candidates overhear each others acknowledgements and forward the packet. Furthermore, ExOR has to make as much hops as AODV when using free space propagation (Figure 20e). With shadowing the number of hops increases for AODV and ExOR, whereas ExOR has to make more hops. By using the distance-based candidate selection the hop count could be significantly reduced for the shadowing case.

A noticeable point is that the distance-based candidate selection algorithm which uses the theoretical derived distances between candidates, reaches a higher throughput and significantly reduces hops and transmissions compared to ExOR in the shadowing case. We realized different candidate selection algorithms, but the results indicate that these algorithms do not find the most promising candidates and therefore do not reach the same results compared to the distance-based approach. So a task for our future is to improve the candidate selection in order to minimize the difference to the distance-based approach.

4.4.5 AODV, ExOR, and MCExOR using Shadowing

The results of the measurements using AODV, ExOR, and MCExOR and the shadowing model are displayed in Figure 22. The simulation took place on regular grids with the horizontal dimension of 2000m, 3000m and 4000m, a constant vertical dimension of 300m and different field densities. In order to show the combined advantage introduced by MCExOR – the opportunistic behavior as well as the multi-channel support – we used the more realistic shadowing radio propagation model. Furthermore we measured up to 4 simultaneous horizontal traffic flows. Figure 22a shows the achieved throughput for AODV, ExOR and MCExOR for a single flow. ExOR outperforms AODV by an average of 46%. In turn, MCExOR with 2 RF channels outperforms ExOR by an average of 64%. The most interesting point is that MCExOR with 2 channels surpasses AODV by an average of 140% – doubling the number of channels results in more than doubling of the observed throughput. Furthermore, from the practical point of view the case with 3 channels is of interest since IEEE 802.11b only offers 3 non-overlapping channels. In this case MCExOR outperforms AODV by 210%. Finally, MCExOR also performs very well with an increasing number of simultaneous flows.

4.5 Additional Observations and Future Work

The MCExOR packet forwarding relies on link qualities. It is crucial that the measurement of link delivery probabilities produces accurate and dependable data. So there is a tradeoff between link probing overhead and convergence time of link qualities: if the probing interval is small, than changes in link quality are propagated quickly, but the overhead is high. For example, by reducing the ETX parameter τ from *4s* to *1s* the average hop count is increased by at least 1 (2000m field length) and 2 (4000m field length), respectively. On the other hand, if the probing interval is big, the overhead is reduced at the expense of convergence time.

Our experiments show that inaccurate delivery probabilities lead to poor results. For example, consider a scenario in which a forwarder has to decide between a long link with a low delivery ratio and a 'good' metric to the final destination and several short links with contrary properties. If the long link's delivery ratio gets overestimated, it will be favored over the short links until its delivery ratio has normalized. So unnecessary retransmissions and possibly packet drops occur and the overall performance decreases.

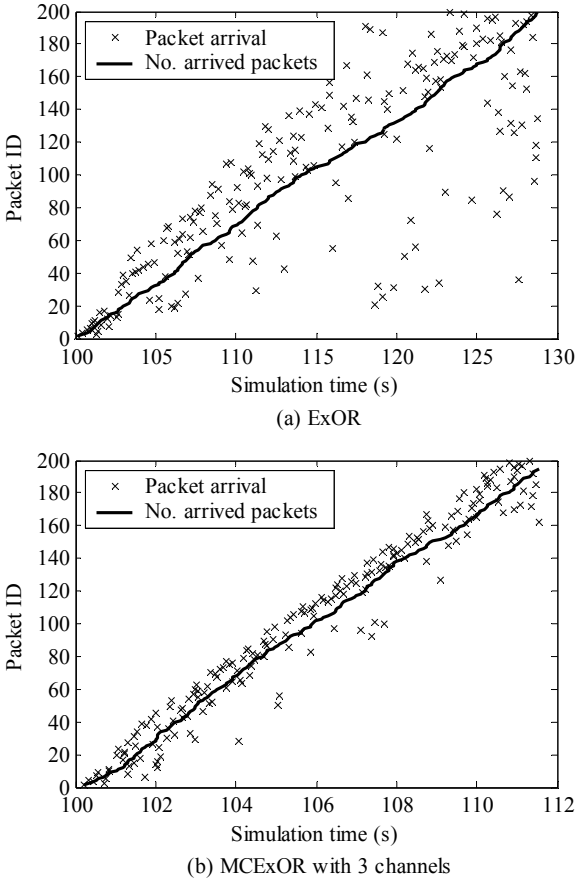

Figure 22: Packet arrival and number of arrived packets in relation to the simulation time for ExOR (a) and MCExOR with 3 RF channels (b) (shadowing σ=4, field 4000x300 m, node density 100x75 m, 4 candidates).

In opposite to AODV ExOR as well as MCExOR use multiple routing paths towards a destination. The ordering in which packets are sent is not necessarily the same in which they arrive. Figure 22 illustrates the ordering in which packets arrive at the destination for an example network. Figure 22a shows the results for ExOR whereas Figure 22b represents the outcome for MCExOR using 3 RF channels. The problem of both protocols is packet reordering which could lead to problems with TCP/IP. However, with MCExOR the variation of the packet arrival time is smaller than the one in ExOR. In case of ExOR the reason for

the high variation in the packet arrival time is fairness to get access to the medium. Our simulations show that there are nodes in the network which are heavily used by other nodes as forwarding nodes. This, however, leads to the problem, that such nodes are not able to relay packets with the same rate as new packet arrive. An evidence for this assumption is the high number of packets in the network queue of such nodes. With MCExOR this problem plays a minor role because of the forwarding policy: Generally, on receiving a packet a node immediately switches to another channel to forward the packet. Within this time the node is 'deaf' and therefore cannot receive further packets. After the transmission the node returns to its home channel and is able to receive further packets. A task for the future is to make MCExOR more TCP-friendly and investigate the performance.

The problem of an initial assignment of home channels is not considered in this paper and left for future work. Another task for our future work is the implementation of MCExOR on real hardware and measurements whether performance can be reproduced in a real world deployment.

5. Conclusions

In this paper we have introduced the multi channel opportunistic routing protocol MCExOR which enables devices with only one transceiver to operate on multiple channels. In a wireless multi-hop mesh network MCExOR minimizes the number of data transmissions and reduces interference to avoid packet collisions. This leads to an increase of the network's capacity as well as to a reduction of latency.

The simulation results presented in this paper show that MCExOR outperforms traditional protocols like AODV by the simultaneous use of multiple RF channels. In conjunction with realistic radio propagation models (shadowing) a further increase in the throughput is observed due to the opportunistic feature of MCExOR. With the increasing number of channels the observed overall throughput superproportionally increases.

In contrast to other approaches MCExOR assigns channels to nodes and not to flows. So also a single flow can benefit from the existence of multiple channels.

6. References

[1] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level Measurements from an 802.11b Mesh Network. SIGCOMM 2004.

[2] Sanjit Biswas and Robert Morris. Opportunistic Routing in MultiHop Wireless Networks. HotNets-II, Cambridge, Massachusetts, 2003.

[3] Sanjit Biswas, and Robert Morris, "ExOR: Opportunistic Muliti-Hop Routing for Wireless Networks", SIGCOMM 2005, August 2005.

[4] R. Choudhury and Nitin Vaidya. Deafness: A mac problem in ad hoc networks when using directional antennas. IEEE ICNP, 2004.

[5] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multiradio multi-hop wireless mesh networks. ACM Mobicom, 2004.

[6] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In Mobile Computing and Networking, pages 61–69, 2001.

[7] Jungmin So and Nitin Vaidya. A Routing Protocol for Utilizing Multiple Channels in Multi-Hop Wireless Networks with a Single Transceiver. University of Illinois at Urbana-Champaign, 2004.

[8] Berlin Roof Net project. Humboldt University Berlin. Systems Architecture Group. www.berlinroofnet.de.

[9] C. Perkins, C. E. Perkins, Ad-hoc on-demand distance vector routing, in MILCOM '97 panel on Ad Hoc Networks, Nov. 1997.

[10] D.S.J.De Couto, D.Aguayo, J.Bicket and R.Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. MOBICOM 2003, San Diego.

[11] Multi-Channel Link-level Measurements in 802.11 Mesh Networks, Mathias Kurth, Anatolij Zubow and Jens-Peter Redlich, IWCMC, July 2006.

[12] JiST/SWANS, Rimon Barr, Wireless Networks Laboratory, Cornell University, jist.ece.cornell.edu/.

[13] Ns2 manual, The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, www.isi.edu/nsnam/ns/doc/.

[14] Estimation of Link Interference in Static Multi-hop Wireless Networks, J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill, Internet Measurement Conference, 2005.

[15] Dynamic Source Routing in Ad Hoc Wireless Networks, David B Johnson and David A Maltz, Mobile Computing, Kluwer Academic Publishers, 1996.

[16] Capacity of Ad Hoc Wireless Networks, Jinyang Li, Charles Blake, Douglas S. J., Hu Imm Lee and Robert Morris, Proceedings of the 7th International Conference on Mobile Computing and Networking, July 2001.

[17] T. S. Rappaport. Wireless communications, principles and practice. Prentice Hall, 1996.

[18] David Kotz, Calvin Newport, and Chip Elliott. The mistaken axioms of wireless-network research. Dartmouth College Computer Science Technical Report TR2003-467, 2003.

[19] Matthew S. Gast. 802.11 Wireless Networks Second Edition. O'Reilly 2005. ISBN 0-596-10052-3.

[20] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot and T. Clausen. Optimized Link State Routing Protocol (OLSR), RFC 3626. http://www.olsr.net/, http://www.olsr.org/