



Fast track article

OFDMA for wireless multihop networks: From theory to practice[☆]



Adrian Loch ^{a,*}, Matthias Hollick ^a, Alexander Kuehne ^b, Anja Klein ^b

^a Technische Universität Darmstadt, Department of Computer Science, Secure Mobile Networking Lab, Mornewegstr. 32, 64293 Darmstadt, Germany

^b Technische Universität Darmstadt, Department of Electrical Engineering, Communications Engineering Lab, Merckstr. 25, 64283 Darmstadt Germany

ARTICLE INFO

Article history:

Available online 10 July 2015

Keywords:

Wireless multihop networks
Corridor-based routing
OFDMA
Cross-layer

ABSTRACT

Orthogonal Frequency-Division Multiple Access (OFDMA) enables nodes to exploit spatial diversity in Wireless Mesh Networks (WMNs). As a result, throughput improves significantly. While existing work often considers the physical layer only, using OFDMA in a WMN also affects the link and network layers. In particular, multi-hop forwarding may result in severe bottlenecks since OFDMA resource allocations are typically based on local information only. In this paper, we design a practical system that mitigates such bottlenecks. To this end, we allow for resource allocation based on channel and traffic conditions at the physical layer, per-subcarrier coding at the link layer, and per-subcarrier packet segmentation at the network layer. We implement our approach on software-defined radios and show that it provides significant throughput gains compared to traditional schemes.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Traditional Wireless Multihop Networks (WMNs) face challenges across multiple timescales. Essentially, channel conditions at the physical layer change much faster than the time required to establish end-to-end paths at the network layer. As a result, cooperative diversity techniques for WMNs such as Orthogonal Frequency-Division Multiple Access (OFDMA) are often challenging—by the time the network layer has found a path that benefits the underlying cooperative physical layer, the Channel State Information (CSI) used to find the path is outdated. This issue strongly limits the practicability of OFDMA in WMNs since it can result in detrimental resource allocations. In other words, the timescale mismatch not only undermines the throughput gain of OFDMA in a WMN but may also result in worse performance than that of a network not using OFDMA.

Recent work introduces Corridor-based Routing [2,3] to decouple the aforementioned timescales and thus enable advanced physical layer techniques in WMNs. Essentially, this approach widens traditional paths to corridors which span multiple nodes at each hop. Fig. 1 shows an example. Each widened hop is called a stage and includes m transmitters and n receivers, which is a topology on top of which many state-of-the-art physical layers can operate. Hence, CSI only needs to be shared among direct neighbors within a stage, instead of over multiple hops. As a consequence, timely CSI dissemination becomes feasible. Moreover, the physical layer can exploit spatial diversity to adapt to the wireless channel in each stage without rebuilding the end-to-end corridor.

[☆] This paper is an extended version of Loch et al. (2014) [1], which appeared in IEEE LCN 2014 and has been selected for a fast-track publication in Elsevier PMC.

* Corresponding author.

E-mail addresses: adrian.loch@seemoo.tu-darmstadt.de (A. Loch), matthias.hollick@seemoo.tu-darmstadt.de (M. Hollick), a.kuehne@nt.tu-darmstadt.de (A. Kuehne), a.klein@nt.tu-darmstadt.de (A. Klein).

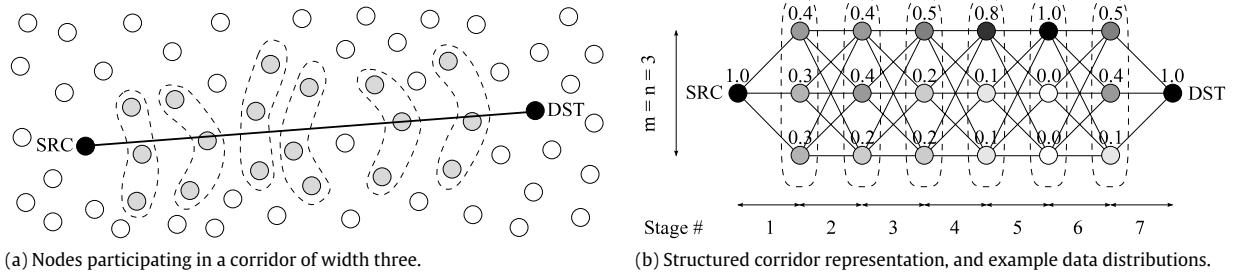


Fig. 1. Corridor example. We use a structured representation as shown in (b) for clarity. The fractions in (b) illustrate one possible distribution of data at each stage. At stage six, a potential bottleneck occurs since stage five forwards all data to the upper node.

The authors of [1,2,4] suggest using Corridor-based Routing to enable OFDMA in WMNs. Basically, OFDMA divides the bandwidth available for communication into orthogonal subcarriers and allows us to assign each subcarrier to an individual link. Existing schemes often allocate each subcarrier to the link providing best quality in terms of signal-to-noise ratio (SNR), which in turn translates into lower bit error rates (BERs) and, ultimately, higher throughput. That is, all links participating in OFDMA share the available subcarriers. When combining OFDMA with Corridor-based Routing, only the $m \cdot n$ links *within* a stage share subcarriers. However, [2,4] consider the physical layer only, while using OFDMA corridors in a WMN also has a significant impact on the link and network layers. In particular, as a result of multi-hop forwarding, data may distribute unevenly among the transmitters of each stage, and thus result in bottlenecks. Fig. 1(b) shows an example. Existing practical OFDMA resource allocation mechanisms for Corridor-based Routing do not take into account the distribution of data at each stage, and may thus severely limit the performance of corridors.

In other words, earlier work only designs a fraction of the system needed to support OFDMA corridors. Moreover, it often does not consider practical issues. In this paper, we design an OFDMA WMN system for Corridor-based Routing, and validate it in practice using software-defined radios (SDRs). We address routing *within* the corridor to mitigate the aforementioned bottlenecks, and build on corridor establishment techniques from related work [5]. Specifically, our contributions are as follows:

1. We design the operation of practical OFDMA for Corridor-based Routing, including the physical, link, and network layers.
2. We present a local resource allocation mechanism tailored to OFDMA corridors which finds close-to-optimal allocations efficiently.
3. We investigate practical issues of OFDMA corridors such as amplifier gain misadjustments and the impact of per-subcarrier scheduling.
4. We evaluate our scheme on SDRs and in simulation to gain insights into practical limitations and analyze random topologies, respectively.

The rest of this paper is structured as follows. We survey related work and explain the generic operation of Corridor-based Routing in Section 2. Section 3 introduces the design of our OFDMA system for WMNs using corridors. Section 4 gives an overview of our implementation and discusses practical issues. Next, in Section 5, we evaluate both the throughput gains of an OFDMA corridor at the physical layer, as well as the performance of our system including the link and network layers. Finally, Section 6 concludes the paper.

2. Related work and background

2.1. OFDMA for WMNs

Research on combining OFDMA with WMNs without using Corridor-based Routing falls into two main categories. On the one hand, existing work analyzes subchannel allocation algorithms [6,7] to find the optimal resource allocation in a WMN. On the other hand, some authors deal with the link layer in OFDMA WMNs [8,9]. Centralized solutions to both issues exist [10,11] but are typically not well-suited for WMNs, since WMNs often operate in a distributed manner. We build on the insights of earlier work in this area, such as adapting the medium-access control (MAC) layer to allow for simultaneous access to the medium on disjoint subcarrier sets [12], and using the per-subcarrier SNR as a simple yet effective metric to decide on subchannel allocation [13]. Further approaches suggest allocating resources in a partially probabilistic manner to reduce the resulting control and CSI feedback overhead [14]. While we also address resource allocation and link layer scheduling, our work addresses a fundamentally different challenge than the above references. We build on a structure in the network – the corridor – which (a) limits the nodes that participate in OFDMA, (b) implies a notion of “forwarding direction” towards the destination, and (c) predefines which nodes act as transmitters and receivers at each hop. In contrast, the above references assume an unstructured network. As a result, such mechanisms are not directly comparable to our approach—the result would be inherently unfair since the underlying assumptions are different in each case.

OFDMA itself has been widely studied for cellular networks [15], including the notion of collaborative OFDMA [16,17]. The latter improves the performance of the downlink by allowing mobile stations to exchange data received from the base station among each other using OFDMA. This is beneficial for the overall system since such transmissions are more energy efficient than retransmissions from the base station. Similarly to the above related work on OFDMA WMNs, this approach is based on an unstructured network of mobile stations. Further, while it also needs to deal with resource allocation, it does not consider multi-hop forwarding, which is a fundamental characteristic of our scheme based on corridors. Hence, our work stands apart from such approaches. We do not aim at improving the already largely studied domain of resource allocation but focus (a) on a system design for OFDMA WMNs based on the characteristics of corridors, and (b) on its practical performance. This includes concurrency at the physical layer, that is, allowing simultaneous transmissions of different nodes. Concurrency is the underlying technique to many state-of-the-art lower layer techniques and has proven [18,19] to overcome the well-known limitations described in [20]—namely, that throughput in WMNs does not scale with network size. This motivates our research on a practical OFDMA system based on corridors.

Previous work presents the notion of corridors [2,21] and analyzes from a theoretical perspective whether they can be beneficially combined with OFDMA. In particular, in [2], the authors optimize the network capacity by means of resource and power allocation in a given corridor with up to five stages, whereas in [21] they devise a scheme to select the nodes of a random network topology that shall participate in the corridor. Still, they consider the physical layer only, focusing on a theoretical and simulation analysis of the achievable capacity and BER performance. Further, [4] presents a theoretical resource allocation scheme which partially mitigates the aforementioned bottlenecks by taking into account the amount of data that each node needs to transmit. However, it does not consider the encoding of data, which plays a fundamental role during allocation, as we discuss in Section 3.4.1. In contrast, our allocation algorithm does take into account encoding. A concept similar to that of corridors is introduced by Gui et al. [22]. However, they study traditional multipaths within the corridor instead of OFDMA. Such closely grouped multipaths often have to deal with self-interference among paths, which is not the case for OFDMA. In earlier work [23], the same authors analyze the outage performance of OFDMA in a topology analog to a corridor. However, they do not consider the challenges resulting from transmitting encoded data, such as potential bottlenecks (cf. Fig. 1(b)), which may have a significant impact on the actual performance at the upper layers. While the above papers deal with corridor-like structures, they do not analyze their *operation*. In this paper, we close this gap.

2.2. Corridor-based routing

Ref. [3] introduces the design of Corridor-based Routing. That is, it describes how a corridor structure works regardless of the physical layer each stage uses. In the following sections, we build on the corridor concept for the particular case of OFDMA. Essentially, the operation of a corridor involves five steps. First, the *corridor construction* step builds the end-to-end corridor with a certain corridor width, ensuring that all nodes in a stage are direct neighbors. This is similar to finding a traditional path in a WMN but additionally widens the path to form a corridor. Existing protocols to build such a corridor [21,5] are often based on a combination of geographic and topologic routing. The width of the resulting corridor depends on the density of the network—if a certain network area is sparse, the protocol may need to narrow the corridor at that location, hence reducing its spatial diversity. In this paper, we use the protocol in [5] for corridor construction.

Secondly, the *stage maintenance* step monitors each stage and locally adapts, if necessary, the transmission parameters at that stage. For instance, if channel conditions change at a stage that is using OFDMA, stage maintenance locally adapts the subcarrier allocation to maximize throughput. Thirdly, the *stage mechanism* selection step chooses the physical layer technique used to forward data in a stage. In our case, we only consider OFDMA as a stage mechanism. Fourthly, *stage coordination* ensures correct node cooperation both within one stage (intra-stage) and among subsequent stages (inter-stage). For example, in OFDMA, intra-stage cooperation ensures that each subcarrier is only used by one node at a time, while inter-stage coordination can be used to enforce that each node receives an amount of data proportional to the quality of its outgoing links. However, in this paper we keep such inter-stage coordination overhead to a minimum. This may cause the aforementioned bottlenecks but our results in Section 5 show that we still achieve throughput gains on average. Finally, the *data transmission* step sends data along the corridor using the chosen stage mechanism.

3. OFDMA WMN system design

In this section, we describe the design of our OFDMA system based on corridors. We explain how our scheme achieves throughput gains, and introduce our algorithms at the physical, link, and network layers.

3.1. Scenario

3.1.1. OFDMA operation

OFDMA allows multiple nodes to transmit simultaneously without interference by assigning disjoint sets of subcarriers to each node. That is, instead of a single transmitter which uses all subcarriers to send data to a single receiver, there are m transmitters that *share* the available subcarriers to send data to n receivers. The gain comes from the assignment

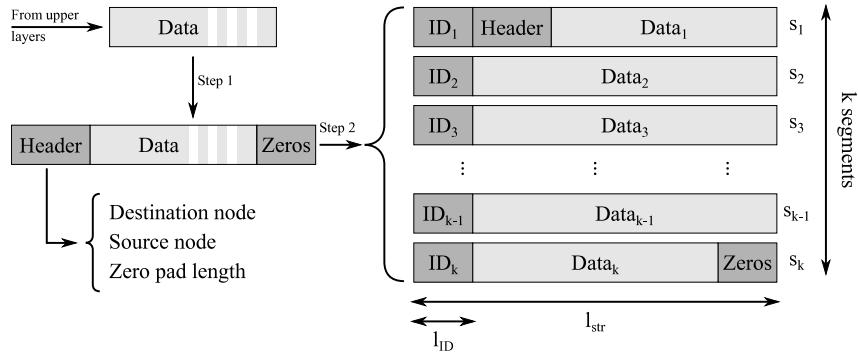


Fig. 2. Network layer operation.

of subcarriers to links which experience good channel conditions, since errors become less likely to happen. As a result, the channel quality in a stage using OFDMA is on average higher than that obtained when using Orthogonal Frequency-Division Multiplexing (OFDM) on a single link, as in routing schemes not based on corridors. Similarly to traditional forwarding, our system uses coding to ensure that data is forwarded correctly. If needed, receivers trigger additional transmissions within a stage. That is, errors do not accumulate as data flows through the corridor but are corrected at each stage.

3.1.2. Throughput gains

Since OFDMA achieves a higher overall channel quality by combining good subcarriers, it allows stages to operate at a higher throughput rate compared to OFDM. This rate is closely related to the modulation and coding scheme (MCS) nodes use. For the case of discrete-rate codes, a subcarrier experiences throughput gains when it can use a higher modulation order with OFDMA than with OFDM for a given BER. For instance, if OFDM can only use BPSK while OFDMA can employ 4-QAM and still achieve a BER similar to OFDM, the OFDMA stage only requires *half the time* to relay data. As a result, throughput improves significantly. For the case of rateless codes, we use Strider [24] as an example. Instead of sending packets with a specific discrete MCS, nodes code packets into batches. If channel quality is high, the receiver only needs a few batches to decode, whereas if channel quality is low, more batches are transmitted. The key advantage of such an approach is that no batch is in vain—all batches contribute to a successful decoding. This simplifies the operation of the link layer significantly. In contrast, retransmissions when using a discrete-rate approach typically repeat the same data at a lower rate and are thus very costly. In combination with an OFDMA stage, Strider needs less batches to transmit a packet on a certain subcarrier compared to the case when it only uses a single link. Therefore, it needs less time to forward data and thus achieves a higher throughput.

3.2. Network layer

The upper layers pass packets to the network layer as an array of bits, together with a source and a destination. If the source node does not know a corridor to the destination, it initiates a corridor construction protocol such as the one described in [5]. As soon as the corridor structure is available, the network layer at the source processes the packet as depicted in Fig. 2. First, we add a packet header and, if needed, a zero padding trailer. The latter is needed to ensure that the length of the resulting packet is a multiple of the data input length of Strider l_{str} . Secondly, we divide the packet into segments of length $l_{\text{str}} - l_{\text{ID}}$, and prepend an identifier of length l_{ID} to each segment. As a result, we obtain a set of segments that Strider can encode individually. Thus, we can transmit each coded segment over a different OFDMA subcarrier and decode it independently at each hop. This is crucial since at each stage segments are distributed differently among nodes according to the local subcarrier allocation at that stage. In other words, groups of segments split and join as nodes forward them through the corridor. The identifier in each segment allows the destination to reconstruct the data.

As depicted in Fig. 2, only the first packet segment includes the source and destination addresses. The node receiving this header becomes the stage coordinator, that is, it ensures coordination and synchronization among the nodes of the current stage. Since all nodes in a stage are one-hop neighbors, the stage coordinator can use well-known techniques, such as preambles, to achieve synchronization (cf. Section 3.3). Moreover, this also means that *any* transmitter in a stage could take on the role of stage coordinator. We follow the above convention, that is, we define that the node which receives the packet header is the stage coordinator for the current stage transmission.

3.3. Link layer

In the following, we present the link layer features of conventional WMNs that we need to adapt in order to operate a corridor. We focus on a single corridor, and thus do not cover additional features to enable the operation of multiple simultaneous corridors. For such a single corridor, we discuss (a) the MAC for an individual stage, and (b) flow control. Both (a) and (b) are particularly challenging for corridors. The former is crucial to coordinate channel access and thus enable

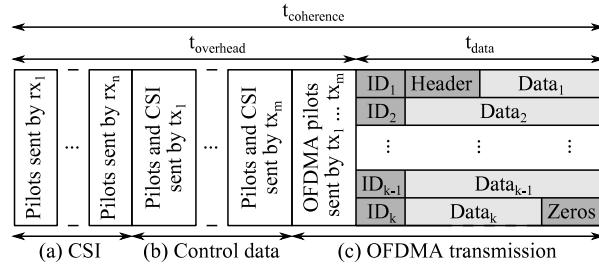


Fig. 3. Orthogonal Frequency-Division Multiple Access (OFDMA) frame structure for a stage with m transmitters tx and n receivers rx .

cooperation of nodes at the physical layer. The latter is more complex than in conventional WMNs because packets are split into multiple segments which require individual encoding.

3.3.1. Medium-access control sublayer

The MAC sublayer avoids collisions within a stage using the frame structure shown in Fig. 3. Essentially, nodes exchange control information in frame parts (a) and (b) using a time-division scheme, and after that, transmit the actual data in frame part (c) using OFDMA, that is, frequency-division. Hence, nodes need to be synchronized both in time and frequency. To this end, the node which receives the segment containing the packet header (cf. Section 3.2) acts as a stage coordinator. Prior to the frame in Fig. 3, the stage coordinator sends a preamble which allows all nodes in the stage to synchronize. We count this preamble as control overhead, that is, we do not assume that the establishment of synchronization has zero cost. Next, in frame part (a), the receivers send pilot symbols to the transmitters one at a time. We assume channels to be reciprocal, that is, channels are equal in both directions. This is valid when both directions operate at the same frequency, which is our case. Moreover, measurements in our testbed environment confirm this. Hence, after step (a) in Fig. 3, each transmitter knows about its n outgoing links, but does not know how the remaining $m \cdot n - n$ links are.

The second step (b) deals with stage coordination. In particular, each transmitter shares its CSI with all other transmitters. In order to do this efficiently, we use a codebook approach, similarly to other technologies using OFDMA, such as LTE. Basically, a codebook is a list of quantized CSI values known to all nodes of a stage. Hence, instead of encoding and sending a full CSI value, transmitters only need to share the index of a similar value of the codebook, which results in much less overhead. In Section 4, we discuss multiple codebook sizes and choose a suitable one for our purposes. In particular, we share CSI in terms of SNR on each subcarrier, since we use the SNR as a metric for allocation. This is the *intra-stage* coordination required by our OFDMA mechanism. Once all transmitters know the SNR on each subcarrier of all $m \cdot n$ links of a stage, each transmitter can independently determine the subcarrier allocation according to the algorithm we discuss in Section 3.4. That is, the outcome of the algorithm is identical on each node since all nodes use the same link SNRs as an input. Thus, no further coordination is needed. Regarding *inter-stage* coordination, the previous stage needs to tell the next stage which data segment is on which subcarrier. This information is encoded as an identifier in each segment (cf. Section 3.2). Finally, in (c) all transmitters send data at the same time using OFDMA. We count as overhead all symbols which are not actual data, including pilots.

3.3.2. Flow control sublayer

The flow control sublayer coordinates detection and correction of transmission errors *per packet segment*, including packet retransmissions. We use Strider (cf. Section 3.1.2) for rateless coding, that is, we do not need to estimate the best suitable MCS for the current channel conditions but just send batches until the receiver can decode. However, to avoid that a transmission incurs an unbounded transmission time, we limit the maximum number of batches to b_{\max} . In other words, if a receiver is not able to decode a segment after receiving b_{\max} batches, the flow control sublayer triggers a retransmission. Similarly, if Strider reports a cyclic redundancy check (CRC) error for a segment, the transmitter nodes retransmit the entire frame. The flow control sublayer allows for a certain maximum number of retransmissions. If more retransmissions are needed, the link layer assumes that the link is not usable at all and reports a transmission failure. In Section 4.3.2 we discuss specific Strider parameters that are suitable for our corridor scenario.

3.4. Physical layer

In this section, we first discuss how we map Strider segments to OFDMA subcarriers. Next, we introduce our resource allocation mechanism that mitigates the impact of bottlenecks in the corridor. Finally, we explain how nodes schedule Strider segments on the resources allocated to them.

3.4.1. Forwarding scheme

The goal of our OFDMA stage mechanism is to transport Strider segments across a stage. The stage may have any shape, which is to say it may have a different number of transmitters m than receivers n . The shape of a stage is unrelated to

the amount of resources—in other words, $m > n$ does not necessarily imply a bottleneck, and $m < n$ does not cause a waste of resources. In particular, the amount of subcarriers at each stage is the same for any m and n . While larger m and n values translate into sharing the same number of subcarriers among more links, it also means that each link transports less segments on average. However, the shape of a stage does affect its spatial diversity. For instance, at a stage that narrows down to a single receiver node ($n = 1$), transmitters must transmit to that node even if their link to the receiver exhibits poor channel conditions on all subcarriers. Still, if channel performance is good, a stage with $n = 1$ can transmit data as efficiently as a stage with large m and n . We allocate resources based on m and n , as well as the stage CSI and the amount of data at each transmitter. Hence, we must send CSI feedback for each subcarrier to the transmitters. To reduce the overhead, we group adjacent subcarriers to subchannels similarly to other wireless systems, and only send CSI feedback per subchannel. This leverages the similar behavior of adjacent subcarriers. The coherence bandwidth [25] defines up to how many adjacent subcarriers behave similarly.

The physical layer design of earlier work on OFDMA corridors (cf. Section 2) often assumes that the allocation mechanism assigns the same fair share of subchannels to each link in a stage. Further, each node typically gets exactly the same amount of data. This simplifies the design significantly since each node has the same amount of resources available for transmission as it has for reception. Therefore, bottlenecks are virtually impossible. Moreover, with this design we can simply extrapolate the operation of a single stage to a multi-hop scenario. However, such a design has significant drawbacks [5]. Specifically, nodes get a fair share of subchannels even if they have poor outgoing links. As a result, the time to transport data across the stage increases prohibitively. This worsens if the mechanism uses the same coding – discrete-rate or rateless – on all subchannels, since the poor links limit the rate on the good ones. This issue becomes particularly critical for highly frequency selective indoor channels as their conditions fluctuate strongly among different subchannels. To limit this effect and to better exploit channel capacity, we allow the allocation mechanism to (a) assign a different number of subchannels to each link of a stage, and (b) use a different coding rate on each subchannel. We expect in turn a better channel utilization. However, this increases complexity and prevents the aforementioned multi-hop extrapolation. In particular, the subchannel allocation at earlier stages has a significant impact on the current stage since it directly influences the distribution of data at the stage transmitters. To address this challenge without incurring additional inter-stage coordination overhead, in Section 3.4.2 we propose an allocation mechanism that mitigates such bottlenecks.

3.4.2. Subchannel allocation

The input parameters of the aforementioned allocation mechanism are (a) the CSI of all stage links, and (b) the amount of data at each transmitter. The goal of the allocation mechanism is to distribute the available subchannels among the stage links in a way that minimizes the *transmission time* locally at that stage, and thus mitigates bottlenecks. In the following, we formulate as optimization problems both the approaches in earlier practical work, such as [26], and our approach to highlight the conceptual difference.

Earlier work. Allocation mechanisms such as in [26] aim at finding subchannel allocations that (a) maximize the SNR, and (b) assign the same number of subchannels to each link of a stage. We formulate the corresponding optimization problem in Eq. (1), which finds the allocation \mathbf{A} that maximizes the subchannel quality \mathbf{Q} in terms of SNR. Both \mathbf{A} and \mathbf{Q} are $L \times N$ matrices, where $L = m \cdot n$ is the number of links in the current stage, and N is the number of subchannels in our OFDMA system. In particular, \mathbf{A} is a binary matrix whose element ij is one if subchannel j is allocated to link i for allocation \mathbf{A} . We must allocate each subchannel to at most one link in order to avoid interference. The second constraint in Eq. (1) captures this requirement. The objective function is the sum of the SNRs of all resources that transmitters use with allocation \mathbf{A} . To consider those SNRs only, we compute the Hadamard product (\circ) of \mathbf{A} and \mathbf{Q} , where element ij in \mathbf{Q} is the SNR of link i for subchannel j . Finally, [26] requires that the allocation algorithm assigns the same number of subchannels to each link in a stage. To enforce this, we include the first constraint in Eq. (1).

$$\begin{aligned} \max_{\mathbf{A}} \quad & \text{snr}(\mathbf{A}, \mathbf{Q}) = \sum_{i=1}^L \sum_{j=1}^N [\mathbf{A} \circ \mathbf{Q}]_{i,j} \\ \text{s.t.} \quad & \sum_{j=1}^N \mathbf{A}_{i,j} = \left\lfloor \frac{N}{L} \right\rfloor \quad \forall i \in [1 \dots L] \\ & \sum_{i=1}^L \mathbf{A}_{i,j} = 1 \quad \forall j \in [1 \dots N] \end{aligned} \tag{1}$$

Our approach. In contrast to [26], our goal is to find a subchannel allocation \mathbf{A} that minimizes the stage transmission time t . We formulate this goal as an optimization problem in Eq. (2). To determine the stage transmission time, we need to know the amount of data $d(n_{tx})$ that each transmitter n_{tx} needs to transmit, and the throughput \mathbf{S} that each transmitter can achieve on each subchannel. Concretely, \mathbf{S} is an $L \times N$ matrix whose element ij is the throughput that link i achieves on subchannel j . Each transmitter n_{tx} may require a different time to transmit its data for a given allocation \mathbf{A} and throughput matrix \mathbf{S} . Since all transmitters transmit in parallel using OFDMA, the transmission finishes when the slowest node has finished. Hence, in

Eq. (2) we define the objective function $t(\mathbf{A}, \mathbf{S})$ as the maximum out of all transmission times of the transmitters n_{tx} of the stage.

$$\begin{aligned} \min_{\mathbf{A}} \quad & t(\mathbf{A}, \mathbf{S}) = \max_{n_{tx} \in [1 \dots m]} \left(\frac{d(n_{tx})}{c(\mathbf{A}, \mathbf{S}, n_{tx})} \right) \\ \text{s.t.} \quad & c(\mathbf{A}, \mathbf{S}, n_{tx}) > 0 \quad \forall n_{tx} \in [1 \dots m] \mid d(n_{tx}) > 0 \\ & c(\mathbf{A}, \mathbf{S}, n_{tx}) \geq 0 \quad \forall n_{tx} \in [1 \dots m] \\ & \sum_{i=1}^L \mathbf{A}_{i,j} = 1 \quad \forall j \in [1 \dots N] \end{aligned} \quad (2)$$

The throughput of node n_{tx} is $c(\mathbf{A}, \mathbf{S}, n_{tx})$, which we define in Eq. (3). Essentially, it is the sum of the throughput values of the subchannels allocated to the outgoing links of n_{tx} . We use again the Hadamard product (\circ) to consider only the throughput values in \mathbf{S} relevant for allocation \mathbf{A} . For each node n_{tx} , we add the rows of $\mathbf{A} \circ \mathbf{S}$ that represent its outgoing links, which we index as $l + (n_{tx} - 1) \cdot n$, with n the number of receivers in the stage and l the index $l \in [1 \dots n]$.

$$c(\mathbf{A}, \mathbf{S}, n_{tx}) = \sum_{j=1}^N \sum_{l=1}^n [\mathbf{A} \circ \mathbf{S}]_{l+(n_{tx}-1) \cdot n, j} \quad (3)$$

In Eq. (2), the first constraint ensures that \mathbf{A} allocates at least one subchannel to all transmitter nodes n_{tx} that have some data to transmit, that is, for which $d(n_{tx}) > 0$. The second constraint limits the feasible region to non-negative throughput values, and the third constraint ensures that each subchannel is allocated at most to one link to avoid interference.

Allocation algorithm. In the following, we present a greedy iterative allocation algorithm that realizes our above approach. In Section 5.1 we compare the performance of this mechanism to the performance of the allocation scheme in [26], which maximizes the SNR, as well as the optimal allocation, which we compute based on an exhaustive off-line search.

To minimize the stage transmission time we must estimate \mathbf{S} , that is, we must determine how long it takes to transmit a segment on a certain subchannel. While for rateless codes this translates into estimating the approximate number of batches needed for decoding, for discrete-rate codes it means estimating the MCS each subchannel supports. Either way, we use the CSI as a basis. This approach cancels out the potentially open-loop operation of rateless codes. Still, it also solves one of the main limitations of such codes, namely, determining when the receiver has enough batches to decode a packet [27]. In our case, the challenge is to estimate a priori how many batches receivers need, instead of notifying the transmitters a posteriori when the receivers are ready to decode. In Section 4, we evaluate in practice the accuracy of this estimation. The key benefit of using rateless codes in our setting is that even if the estimation is wrong, all batches contribute to packet decoding, in contrast to discrete-rate approaches (cf. Section 3.1.2).

Fig. 4 summarizes our allocation mechanism. It consists of two independent phases – assignment and release – and follows a greedy iterative approach. In each iteration, the assignment phase assigns one of the available subchannels to one of the transmitter nodes. Essentially, the mechanism selects the node which would take most time to transmit its data using the subchannels currently allocated to it. Next, the mechanism allocates to that node the subchannel on which it requires the least time to transmit a Strider segment, out of the still available subchannels. This process is repeated until no more subchannels are left. In other words, in each iteration the assignment phase allocates resources to the transmitter that is currently limiting the overall stage throughput. If multiple transmitters limit the throughput, that is, if they all take the same amount of time to transmit their data, the mechanism chooses among the transmitters the one which has most data to transmit. This occurs, for instance, always in the first iteration of the mechanism—no node has any subchannels yet and thus they all take an infinite amount of time to transmit their data. Further, if multiple nodes limit throughput with the current allocation and, additionally, all of these nodes have the same amount of data, the mechanism selects one of the nodes at random. Hence, the assignment phase involves both the link CSI and the amount of data at each node, since both are needed to determine the time a node needs to transmit its data on a certain set of subchannels.

The release phase tries to further reduce the stage transmission time by reallocating subchannels—since the assignment phase follows a greedy approach, its result is not necessarily optimal. Basically, we find the node n_l which limits the throughput and compute its so-called “wish list”. The wish list of a node contains those subchannels the node does not own but which it would like to use because it experiences good channel conditions on them. The wish list is ordered, that is, the first subchannel on the list is the one on which the node n_l would perform best. Next, we estimate the stage transmission time assuming that the node n_o , owning the first subchannel on the wish list of n_l , releases that subchannel to n_l . If the new stage transmission time is shorter than the initial one, the subchannel is reallocated to n_l . Otherwise, the release phase repeats the process with the next subchannel on the wish list. The release phase ends when it reaches the end of the wish list of n_l without reallocating any subchannel. However, if the release phase reallocates a subchannel, a different node n_l may limit the throughput of the stage. In that case, it starts over again, that is, it finds the wish list of the new node n_l and steps through it from the beginning.

3.4.3. Data transmission

As a result of subchannel allocation (cf. Section 3.4.2) each node owns a set of subchannels for transmission. While the allocation mechanism considers the overall amount of data at each node, it does not specify the scheduling of each

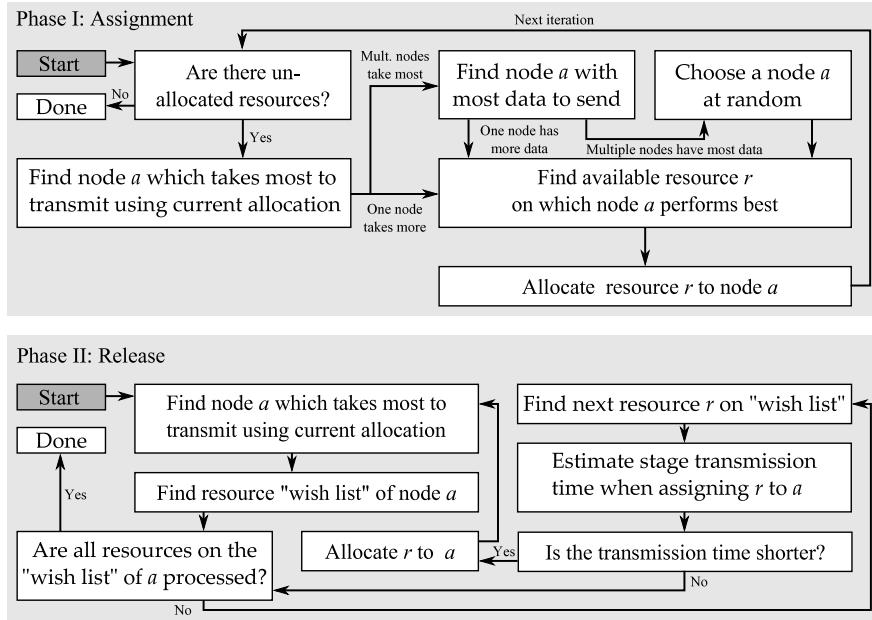


Fig. 4. Subchannel allocation mechanism.

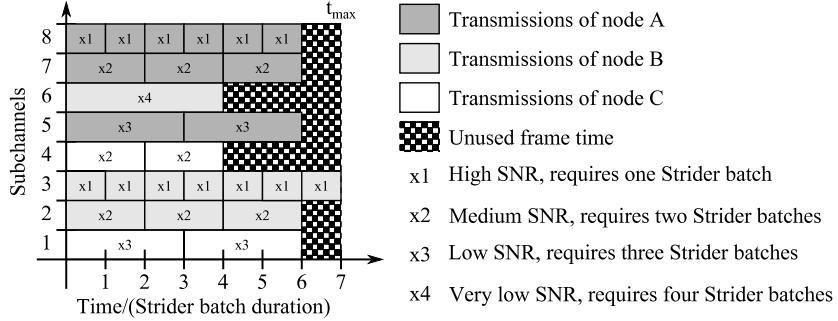


Fig. 5. Scheduling example for an OFDMA frame.

individual Strider segment. That is, each node must decide how it uses the subchannels assigned to it. The key idea is to find a scheduling which results in roughly the same transmission time on all subchannels. In other words, subchannels with high SNR should transport more Strider segments than subchannels with low SNR. Fig. 5 gives an example. The goal is to minimize the checkerboard area at the end of the frame, which stands for the time some subchannels are idle while one or more other subchannels are still in use. In Fig. 5, node B is the limiting node since it needs seven times the duration of a Strider batch to finish transmitting on all of its subchannels. However, note that no other scheduling at that node would result in a shorter stage transmission time. In particular, if it reschedules one of its packets on subchannel three to either of its other two subchannels, the overall duration increases since transmission of a packet on those subchannels takes longer than on subchannel three. The stage transmission time is given by the slowest node out of all transmitters. We use the iterative scheduling algorithm in Fig. 6 to minimize the checkerboard area in Fig. 5. Each node runs this algorithm locally, that is, we minimize the idle time among the subchannels allocated to a certain node. In each iteration our algorithm schedules a Strider segment on the subchannel of the current node on which transmitting the segment incurs the least increase possible in transmission time. The algorithm repeats this process until all segments are scheduled.

4. Implementation

4.1. Practical setup

4.1.1. Hardware platform

We use the Wireless Open-Access Research Platform (WARP), which is an FPGA-based SDR developed at Rice University [28]. It enables experiments in settings similar to an 802.11 network but with full control over the lower layers.

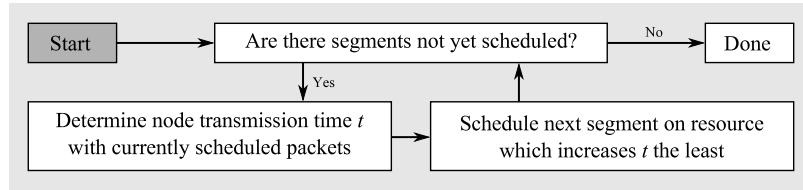


Fig. 6. Scheduling algorithm to assign Strider segments to subchannels.

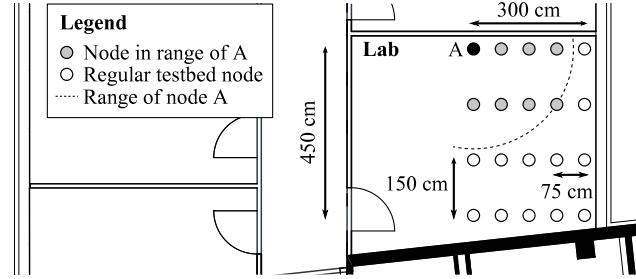


Fig. 7. Practical setup with 20 nodes using SDRs.

We implement our OFDM corridor on WARP Drive [29], a framework which supports rapid prototyping of wireless multihop techniques on WARP. It extends WARPLab [30], which is a reference design for WARP that allows for practical experiments directly from the Matlab workspace. Thus, we first calculate in Matlab the samples to be transmitted. Next, Matlab transfers these samples via Ethernet to the sending WARP board, which transmits them over the wireless medium. The receiving board samples the signal and sends it back to Matlab. Between transmissions, data is processed in Matlab. Despite this approach is not fully real-time due to delays in transferring signals to and from Matlab, we do *not* process data offline, but online and *interactively*. We only relocate processing from the WARP boards to Matlab. In other words, WARP Drive does not schedule transmissions in advance but triggers them dynamically, based on network operation. Further, it allows us to switch seamlessly between simulations and practical experiments. That is, we use the same code for both cases, which allows us to compare results and understand practical issues in detail.

4.1.2. Experiment setup

We carry out experiments on up to ten WARP boards. Each board features multiple radio interfaces. WARP Drive enables us to use each of these interfaces as if each interface were an individual node. Still, it treats data from each interface independently, and does not exploit that some nodes share the same board. Fig. 7 shows our setup. We place 20 nodes on a regular pattern to avoid undesirable side-effects, which would make drawing conclusions from the measurements difficult. Although nodes are close to each other, we achieve typical indoor SNRs by setting low transmit gains. Moreover, we discard packets on links longer than 2.7 m. This prevents nodes from being fully connected and yields a multihop environment. As an example, Fig. 7 depicts the neighbors of node A. We use an OFDMA physical layer with 20 MHz channels, 16 subchannels with three subcarriers each, 312.5 kHz subcarrier spacing, and 12.5% cyclic prefix (CP).

4.1.3. Synchronization

OFDMA requires transmitters to be synchronized in time and frequency. We achieve this via wired synchronization. However, since all nodes in a stage are direct neighbors, wireless synchronization would also be feasible using techniques from single-hop systems that build on OFDMA, e.g., LTE. While we abstract from this issue in most measurements to focus on OFDMA performance, for our practical experiments in Section 5.2 we do synchronize nodes wirelessly to show that carrier-frequency offset (CFO) estimation for OFDMA is feasible over the air. Additionally, receivers need to find out when to start receiving data. To this end, we prefix the signal with a short and long preamble as in 802.11 g. Receivers can determine the beginning by correlating the incoming signal with the long preamble. Moreover, we use the short preamble to train the Automatic Gain Control (AGC).

4.2. Simulation setup

In addition to our practical experiments, we also perform simulations in order to study larger networks. Similarly to [5], we consider two networks with about 50 nodes randomly deployed on an area of 50×50 m² and 21×21 m², respectively. We call the former *UD-L* and the latter *UD-S*. The key difference between them is the density—both have a similar number of nodes but *UD-S* deploys them on a much smaller area than that of *UD-L*. As a result, the average link SNR is significantly larger for *UD-S* than for *UD-L*. We set the range of nodes to 15 m for *UD-L* and to 6 m for *UD-S* to achieve a similar node

Table 1

Strider parameters. We refer the reader to [24] for details on each parameter.

MCS	QPSK with a 1/5 code rate
Batch size	33
Bits per Strider packet	362 bits
Bits per batch of packets	$33 \cdot 362 = 11946$ bits ≈ 1500 bytes
Number of batches b_{\max}	27
CRC bits	16 bits

degree in both cases. We generate networks with roughly uniform density, that is, we avoid “holes” in the network topology in order to recreate a dense indoor environment. At the physical layer, we assume Rayleigh fading channels with a path loss exponent of $\alpha = 2$, and additive white Gaussian noise (AWGN).

4.3. Practical considerations

4.3.1. Coherence time

Our testbed is static and thus channels are stable. Therefore, the coherence time is long. Continuous channel measurements over a time frame of 30 min reveal that this assumption holds. We exploit this to overcome the delays for transferring signals to/from Matlab. While the delays are in the order of *milliseconds*, in our testbed channels remain constant over *minutes*. Thus, CSI obtained in frame part (a) (Fig. 3) is still up-to-date in part (c). If realized in real-time, our scheme does not require long coherence times.

4.3.2. Strider parameterization

In Section 3, we motivate the use of rateless codes for OFDMA corridors, and focus on Strider [24] as a concrete example. Essentially, Strider makes a conventional MCS *rateless*. As a result, the MCS does not require a certain minimum SNR to decode. Instead, it can operate at lower SNRs as long as the transmitter transmits enough coded batches. Strider allows us to define (a) the MCS it uses as a basis to achieve rateless operation, (b) the number of packets it encodes in each batch, and (c) the size of those packets. In the following, we explain the particular values that we use for each of these parameters. To this end, we consider the characteristics of Strider discussed in [24], as well as our experiment setup. Table 1 shows an overview of the parameter values we choose.

(a) *MCS*. According to [24], the performance of Strider is insensitive to the modulation of the base MCS it uses. While the coding rate does influence performance, its impact is limited for most SNRs. Hence, we choose a 1/5 code rate as in [24]. For high SNRs, higher code rates perform better. Such codes might be beneficial for OFDMA since it can avoid subchannels with poor channel conditions, and thus achieves higher SNRs on average. Still, [24] suggests that the improvement is limited. Thus, we do not implement higher codes. Implementing such codes would improve our OFDMA results in Section 5, if at all.

(b) *Batch size*. The batch size is the number of packets that Strider encodes in each batch. The larger the batch size, the more fine-grained the improvement achieved with each additional batch. Section 9.1 in [24] experimentally shows that a value of 33 packets is well-suited for a target SNRs in the range of 3–25 dB, which is typical for indoor environments. This also holds for our testbed. In particular, the lower SNRs in that range correspond to subchannels that experience deep fades due to multi-path effects, while the high SNRs match subchannels that experience good channel conditions. Hence, we conclude that a batch size of 33 is also well-suited in our case.

(c) *Packet size*. Ref. [24] suggests that the packet size does not have a significant impact on the performance of Strider. Still, our OFDMA transmission scheme (cf. Section 3.4.3) encodes data *per subchannel*, as shown in Fig. 5, instead of stripping it across all subchannels. As a result, the transmission of a batch requires significantly more time in the former than in the latter case. Hence, we choose a small packet size – 362 bits – to avoid impractically long transmission frames.

Finally, to ensure that the transmission time of each batch is bounded, we must limit the maximum number of batches b_{\max} that a node may use to convey a packet at a receiver. This is an arbitrary limit which translates into a minimum data rate below which we consider that a transmission has failed. Similarly to [24], we set this limit to $b_{\max} = 27$ batches, which is equivalent to a minimum rate of 0.5 bits per symbol. The above discussion shows that our parameterization of Strider is valid for a large range of SNRs. OFDMA improves the average SNR since corridor stages can avoid subchannels with poor CSI. Hence, we conclude that the throughput gain of OFDMA is mostly insensitive to the parameters of Strider. That is, our results in Section 5 are not tightly coupled to the Strider parameterization that we use.

4.3.3. Throughput

The delays incurred by WARPLab and thus WARP Drive prevent us from measuring throughput directly, since they would strongly affect the result. Moreover, the large coherence times in our testbed would lead to CSI measurements at much larger intervals than in a real-world scenario. To circumvent these limitations, we obtain throughput by extrapolating our measurements. We consider an indoor scenario and assume a realistic coherence time for such a setting, namely, $t_{coh} \approx 45$ ms [31]. For overhead calculations, we consider that CSI measurements, and thus the resulting node coordination, must take place at least at intervals of size t_{coh} . In other words, while we might only need to coordinate nodes once at the

beginning of a transmission due to the stability of channels in our lab, we do take into account that such coordination would be required more frequently in a real-world environment.

In particular, we obtain throughput results as follows. We compute the end-to-end throughput at the interface of the network layer to the upper layers, that is, we include the overhead of the network, link, and physical layers. To this end, we divide the payload data at the network layer, d_{net} , by the end-to-end transmission time, which is the sum of the individual stage transmission times t_{stage}^i of a corridor of length H , with $i \in [1 \dots H]$, as shown in Eq. (4). The stage transmission time t_{stage}^i is the sum of the payload and control data transmission times. In Eq. (5), we account for the above coordination and synchronization overhead, which is required at intervals of size t_{coh} . We measure the control data transmission time t_{control}^i for the current stage i practically at the beginning of each testbed transmission. The payload data transmission time t_{data}^i is the maximum out of the individual subchannel transmission times t_{sch}^j , with $j \in [1 \dots N]$ where N is the number of subchannels, as illustrated in Fig. 5 and Eq. (6). The subchannel transmission time t_{sch}^j depends on (a) the number of packet segments scheduled on j , n_{pkt}^j , and (b) the time required to convey a single segment at the receiver on j using Strider, t_{pkt}^j , which translates directly into the number of required batches. Similar to related work on rateless codes [24], we abstract from the minimal feedback a receiver sends to a transmitter when it needs more batches to decode. Instead, we always transmit all b_{\max} batches Strider generates at the transmitter and then determine how many the receiver actually needs. This allows us to evaluate the accuracy of our estimation regarding the number of batches needed at the receiver to decode (cf. Section 3.4). We follow a similar approach for the traditional forwarding mechanism that we use as a baseline to allow for a fair comparison. However, transmitting all b_{\max} batches increases the processing time per packet segment significantly. To keep the experiment runtime reasonable, we transmit one packet segment via each subchannel and extrapolate the resulting performance to the remaining Strider segments on each subchannel. That is, we measure t_{pkt}^j for the first packet segment on subchannel j , and multiply the result with the amount of packet segments n_{pkt}^j scheduled on j , as shown in Eq. (7). Altogether, Eqs. (4)–(7) summarize how we obtain the throughput of OFDMA based on our testbed measurements t_{control}^i and t_{pkt}^j .

$$\text{thp} = \frac{d_{\text{net}}}{\sum_{i=1}^H t_{\text{stage}}^i} \quad (4)$$

$$t_{\text{stage}}^i = t_{\text{control}}^i \cdot \left\lceil \frac{t_{\text{data}}^i}{t_{\text{coh}}} \right\rceil + t_{\text{data}}^i \quad (5)$$

$$t_{\text{data}}^i = \max_{j \in [1 \dots N]} (t_{\text{sch}}^1, t_{\text{sch}}^2, \dots, t_{\text{sch}}^j, \dots, t_{\text{sch}}^N) \quad (6)$$

$$t_{\text{sch}}^j = n_{\text{pkt}}^j \cdot t_{\text{pkt}}^j \quad (7)$$

4.3.4. Estimation of the transmission duration

Our OFDMA system estimates the transmission time in terms of Strider batches to schedule data on subchannels (cf. Section 3.4). In the following, we evaluate the accuracy of this estimation in practice, and show its impact on our testbed measurements. Fig. 8 shows how much the actual transmission time of Strider segments deviates from the estimated transmission time for both simulated and practical experiments. In particular, Fig. 8(a) depicts this deviation for individual subchannels. We observe that the result roughly follows a normal distribution. However, the variance is significantly larger in the practical case than in simulation. That is, in practice we infer from CSI either too long or too short transmission times more frequently. In other words, while our estimation is mostly accurate, sometimes individual segments require significantly more batches than expected. A detailed analysis of our measurement data shows that the reason for this behavior is interference from other wireless networks operating on the same channel. The average SNR is often only slightly lower than for the cases without interference since collisions typically affect only a portion of the frame.

The aforementioned mismatches in practical Strider transmissions can have a large impact on OFDMA throughput. In order to exploit the combination of multiple frequency selective links, OFDMA must transmit each segment on an individual subchannel (per-subchannel scheduling). In contrast, a baseline mechanism not exploiting frequency selective channels can use all subchannels in parallel to transmit a segment (wideband scheduling). Fig. 9 shows an example on how this design requirement significantly limits the performance of per-subchannel scheduling. Specifically, we depict the impact of the receiver needing three times more batches to decode one of the Strider segments. In wideband scheduling, the transmitter uses all subchannels to transmit the additional batches and thus the overall frame length increases by only about 16.6%. In contrast, in per-subchannel scheduling, the transmitter must use only one subchannel per segment, resulting in 200% overhead. The checkerboard area in Fig. 9 shows the wasted time. In other words, if the input to the scheduling algorithm is inaccurate, the impact on throughput may undermine the potential gains of OFDMA entirely. Inaccurate transmission time estimations also affect the design of schemes not using OFDMA, since per-subchannel scheduling is also beneficial for transmissions on a single link. Essentially, nodes can encode data on each subchannel differently to exploit frequency

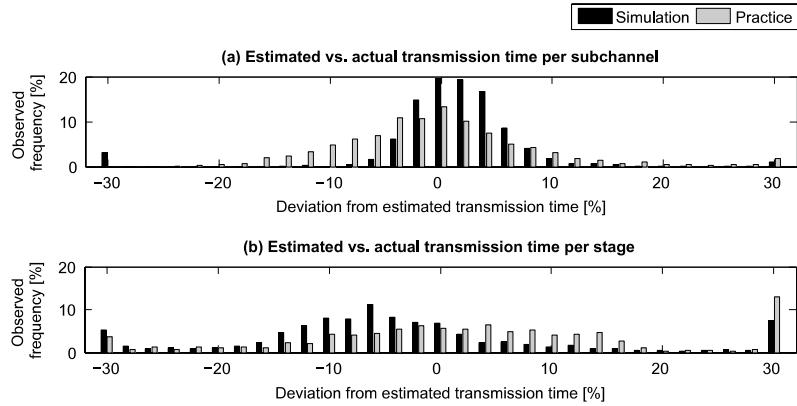


Fig. 8. Deviation from estimated transmission time in simulation and practice. While (a) shows the accuracy for an individual subchannel, (b) depicts the effect on the entire stage. The values at $\pm 30\%$ include the deviations in $[-\infty \dots -30\%]$ and $[30\% \dots \infty]$.

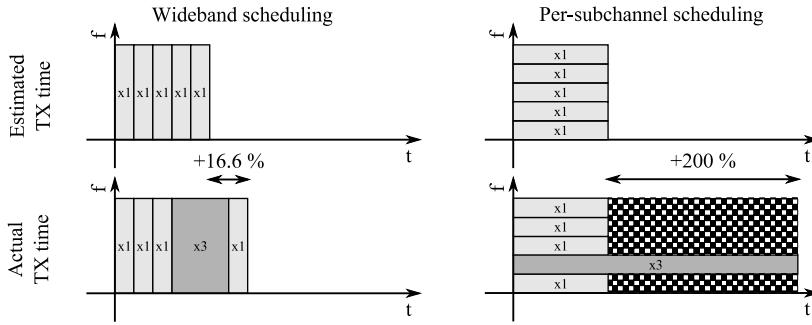


Fig. 9. Example of impact of inaccurate batch number estimation on wideband and per-subchannel scheduling. The percentage indicates the increase in frame length.

selective channels. However, such a scheme would run into problems similar to those illustrated in Fig. 9. In Section 5, we compare our OFDMA system to both a baseline which uses wideband scheduling and one which builds on per-subchannel scheduling.

In Fig. 8(b) we depict the impact of the effect in Fig. 9 on the stage transmission time. That is, we show the results for an entire frame instead of for a single subchannel. As expected, the inaccuracy of practical estimations result in significantly longer stage transmission times than in simulation. Note that the value at 30% in Fig. 8(b) includes all time deviations in the range $[30\% \dots \infty]$. Our data shows that the practical values in that range are, on average, considerably larger than the simulation ones.

4.3.5. Gain control

Gain control in OFDMA is challenging due to the overlapping of multiple signals *in time*. The receiver can decompose the resulting signal *after* quantization using the Fast Fourier Transform (FFT). Still, *before* quantization the receiver can only operate on the sum. If one signal is stronger than the others, the weaker ones suffer from higher quantization noise. The key problem is that the receiver can only adjust the strongest signal to the input range of the quantizer, while all others are sampled with less accuracy.

We show the impact of this issue in Fig. 10(a), which depicts the BER when using 64-QAM on six different stages of width two. We use the BER as a metric of the aforementioned noise impact. The OFDM baseline in Fig. 10(a) refers to the BER when only using one link of the stage, while the simultaneous OFDMA case uses all links. Additionally, we show the performance of a *sequential* OFDMA variant, which allows nodes to send in sequence on the subcarriers allocated to them. Hence, signals do not overlap in time and thus the gain control issue is circumvented. This is not how we envision the scheme to operate once deployed, but it allows us to illustrate the impact of gain control. As shown in Fig. 10(a), the impact of gain misadjustments varies for each stage, since it is directly dependent on the physical environment surrounding the stage. We observe that simultaneous OFDMA performs worse than sequential OFDMA, specially at stage E. To solve this problem, transmitter nodes must adjust their gains to ensure that all signals are received with similar power at the receivers. This is intrinsic to OFDMA and orthogonal to the gains achievable by subchannel allocation. Hence, we do not tackle it in our implementation. To obtain the actual subcarrier allocation gain, for the bulk of our experiments we show the sequential OFDMA results and compute throughput as if transmissions were simultaneous.

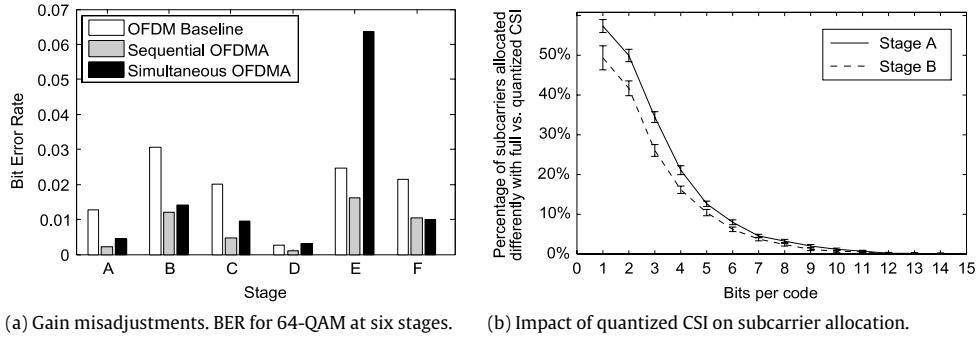


Fig. 10. Practical measurements on gain control and feedback quantization.

Table 2
BER for 64-QAM with increasing codebook size.

Codebook size	16 codes	64 codes	512 codes	8192 codes
OFDMA BER	0.0179	0.0170	0.0146	0.0139

4.3.6. Feedback

We use quantized CSI feedback and account for the resulting overhead. To find a suitable codebook size, we measure how many bits per CSI value are needed to obtain the same subcarrier allocation than with full CSI. Fig. 10(b) depicts the result for two stages of a corridor of width two. Both behave similarly, which means that the required codebook size does not depend on the specific channels of a stage. To achieve virtually identical allocations compared to full CSI, we need to quantize feedback using 13 bits per code. However, Table 2 shows that the BER achieved with much smaller codebooks is similar. While allocations using smaller codebooks are different than with full CSI, subcarriers allocated differently have similar performance. Hence, larger codebooks only provide marginal improvement. To keep overhead low, we use four bits per code, which results in 16 codebook entries.

5. Evaluation

In this section, we evaluate OFDMA for WMNs. First, we investigate the performance of the allocation algorithm that we present in Section 3.4.2. Next, we present the raw throughput gains at the physical layer using discrete-rate coding. Our goal in this first part is to investigate the potential gains of OFDMA. After that, we study the throughput gain of the complete system we introduce in Section 3. In this case, we use Strider as a rateless code. Our goal in this last part is to analyze how much of the raw gain does our OFDMA system achieve.

5.1. Allocation algorithm

In Section 3.4.2, we present a subchannel allocation algorithm tailored to corridors. The key challenge is to minimize the stage transmission time for any channel conditions and data distribution at the transmitters. In this section, we evaluate the performance of our allocation algorithm in terms of transmission time compared to the optimal allocation. To obtain the latter, we perform an off-line exhaustive search of all possible allocations. Further, we also analyze the performance of an allocation algorithm proposed in earlier work [26]. Fig. 11 shows our results for a stage with $m = n = 2$. For larger stages, we obtain equivalent results. We observe that our allocation mechanism performs close to the optimum both in simulation and practice, as well as with evenly and unevenly distributed data. In contrast, earlier work does not consider the distribution of data at the transmitters (cf. Eq. (1)). Hence, the impact on transmission time in those cases is significantly larger, reaching up to an increase of more than 30% compared to the optimal allocation. We expect such cases to be the norm rather than the exception, since the distribution of data among transmitters in a stage strongly depends on the random CSI of the channel. Hence, this highlights the benefits of our allocation mechanism.

5.2. Physical layer gain

5.2.1. Experiment design

In this experiment, we aim at determining in practice the raw throughput gain of OFDMA compared to a mechanism that does not exploit spatial diversity. To this end, we build two corridors of constant width two and four, respectively, with three stages. We then transmit data encoded with different discrete-rate MCSs and compute the BER for each case. We do not yet use the allocation and scheduling mechanisms we introduce in Section 3.4 but allocate a fair share of subchannels to each

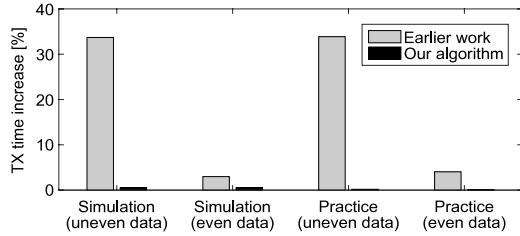


Fig. 11. Transmission time increase of our allocation algorithm (cf. Section 3.4.2) and earlier schemes compared to the optimal allocation, which we compute off-line. Data is distributed *evenly* if all transmitters of a stage have the same amount of data to transmit.

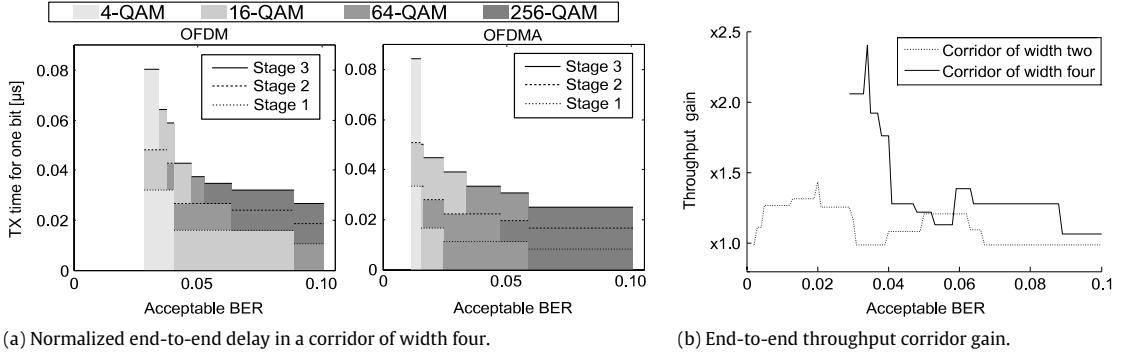


Fig. 12. Practical corridor performance at the physical layer.

node based on CSI. Moreover, we use the same modulation on all subchannels—while we use per-subchannel coding in our later system evaluation, this more simple approach allows us to decouple the gains stemming from spatial diversity from the benefits of exploiting frequency-selective fading. As a baseline, we use traditional forwarding based on OFDM. That is, we use the same corridor than OFDMA but choose a random node of each stage as a forwarder. The above experiment produces the BERs for a range of modulations. However, the actual throughput depends on the amount of bit errors the channel code in use can correct. Since we later compare the results of this experiment to a rateless code, we analyze the throughput gain for an entire range of “acceptable BERs”. Namely, we do not limit our results to the error correction capability of a particular code but consider a range of values. This allows us to obtain broader insights on the potential gains of OFDMA.

5.2.2. Practical results

We expect the gain of OFDMA to stem from being able to choose higher modulations than our baseline OFDM for a certain acceptable BER. In Fig. 12(a) we depict the end-to-end delay incurred when transporting data through our corridor of width four for any acceptable BER up to 10%. We “normalize” the result dividing the time by the amount of sent data to highlight that higher modulations transport each single bit faster. The gray tones indicate the modulation used in each stage. For example, for OFDM and 10% acceptable BER, the first stage uses 64-QAM, while the second and the third use 256-QAM. The transmission time decreases, as expected, with increasing acceptable BER, since the higher the number of errors the code can correct, the higher the modulations that become possible. For any acceptable BER, OFDMA can use higher modulations and thus requires less time than OFDM to transport data, which directly translates into throughput gain. For small acceptable BER values, both OFDM and OFDMA cannot operate, since the BER is too high even for the lowest modulation scheme in at least one of the stages. This is shown by the left white areas in Fig. 12(a). However, note that OFDMA can already operate at about 0.012 acceptable BER, while OFDM requires at least 0.029 error correction capability to become feasible.

In Fig. 12(b) we show the end-to-end throughput gains resulting from our observations in Fig. 12(a). The corridor of width four doubles throughput for certain acceptable BERs and the one of width two achieves up to $1.4 \times$ gain. We conclude that gains increase with corridor width—the more links, the higher the probability that OFDMA can allocate links with good channel conditions to each subchannel is. The curve for the corridor of width four starts at 0.029 acceptable BER since OFDM cannot operate for lower values.

5.3. System gain

In the following, we evaluate our OFDMA system both in simulation and practice. The former allows us to study random network topologies while the latter gives us insights into real-world effects.

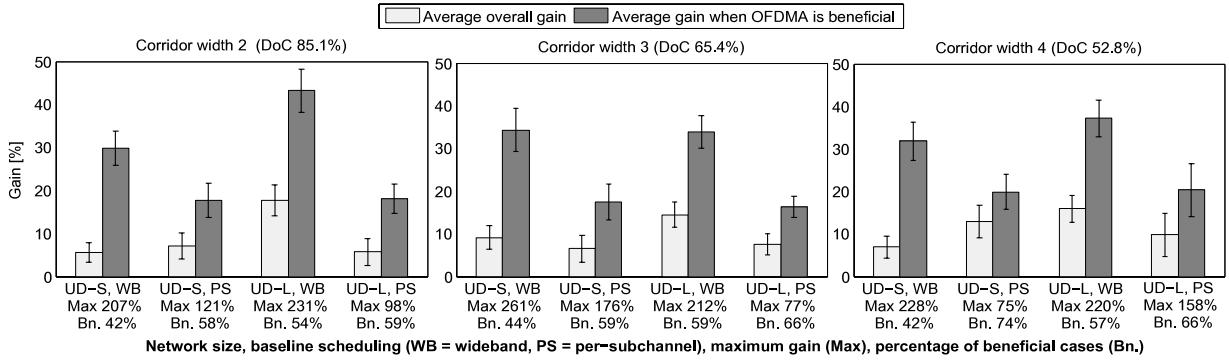


Fig. 13. Simulation results. We compare the performance of OFDMA both to a baseline using wideband scheduling and per-subchannel scheduling.

5.3.1. Simulations

Fig. 13 shows an overview of our simulation results. We depict the throughput gain of OFDMA compared to our baseline, which uses OFDM. Specifically, for our baseline we consider both wideband scheduling and per-subchannel scheduling. Further, we measure the performance of our system for corridors of widths two, three, and four. For each case, we compute the Degree of Construction (DoC), which reflects the spatial diversity of the corridor. In particular, the DoC is the percentage of the actual number of nodes that form a corridor compared to the canonic number of nodes a corridor of the given length and width should have. For instance, the corridor in Fig. 1 has a DoC of 100% since all of its stages have three nodes. If the corridor construction protocol would have needed to narrow the corridor due to network sparsity, the DoC would be lower. For each network size and scheduling type in Fig. 13, we compute the average throughput gain for (a) all experiment repetitions, and (b) the experiment repetitions where OFDMA is *beneficial*. That is, in some cases it does not pay off to use OFDMA because it results in a *negative* gain compared to our baseline. This occurs, for instance, when a forwarder with poor outgoing links needs to forward a large amount of Strider segments as a result of local allocation in previous stages. For each scenario, we show the percentage of beneficial cases out of all experiments. Moreover, we also indicate the maximum gain in each case.

Fig. 13 shows that our OFDMA system achieves up to roughly $2 \times$ throughput gain in the best cases, similarly to our results at the physical layer in Section 5.2. Further, we observe that the average gains depend strongly on whether we consider all cases or only the beneficial ones—the former yield gains in the range from 10% to 20%, while the latter achieve up to 40% better throughput than our baseline. In other words, estimating whether OFDMA is beneficial for a certain end-to-end connection is crucial. The results show that the percentage of beneficial cases itself depends on (a) the network scenario, and (b) the type of baseline scheduling. Regarding (a), OFDMA tends to pay off more frequently in UD-L than in UD-S. The reason is related to the average SNR, which is lower for UD-L than for UD-S. While UD-L achieves a lower throughput, relative gains increase as the impact of subchannels with poor channel conditions is larger. In contrast, at high SNRs, most subchannels only require few batches to convey a Strider segment. Thus, using OFDMA to combine the best subchannels provides less gains. As to (b), we observe that the number of beneficial cases is larger for per-subchannel scheduling than for the wideband case. The reason is that, in the per-subchannel case, the baseline needs to schedule transmissions similarly to OFDMA. That is, it inevitably wastes transmission time due to some subchannels finishing transmission earlier than others. Hence, OFDMA is beneficial in a wider range of situations.

Still, the percentages of beneficial cases in Fig. 13 show that the baseline using per-subchannel scheduling only wastes transmission time occasionally. In all other cases, such scheduling enables the baseline to exploit the frequency-selectiveness of channels. As a result, it achieves a higher throughput than in the wideband case, which ultimately results in OFDMA gains mostly below 20%, as shown in Fig. 13. Since OFDMA must use per-subchannel scheduling, this suggests that only a comparison with a baseline also using per-subchannel scheduling would be fair. However, the practical effect we describe in Section 4.3.4 results in the opposite behavior. That is, wideband scheduling is more beneficial in practice for our baseline than per-subchannel scheduling. This issue highlights the difference between theory and practice—we discuss this in detail along with our practical results below. Finally, in Fig. 13 we also show the average DoC for each corridor width. We observe that the DoC becomes smaller for wider corridors. Basically, the wider the intended corridor width, the more difficult it is for the corridor construction protocol to find enough nodes for each stage, forcing the corridor to narrow more often. This explains why wider corridors do not necessarily result in larger gains in Fig. 13. In contrast, our results in Section 5.2 exhibit such a relation since corridors always have 100% DoC in that case.

5.3.2. Practical results

Fig. 14 depicts our practical OFDMA measurements in our SDR testbed. Again, we consider both wideband and per-subchannel scheduling for the baseline scheme. Additionally, we investigate a low SNR case. To this end, we add a node to our testbed which continuously sends noise on all subchannels. As a result, the average SNR at the other nodes drops.

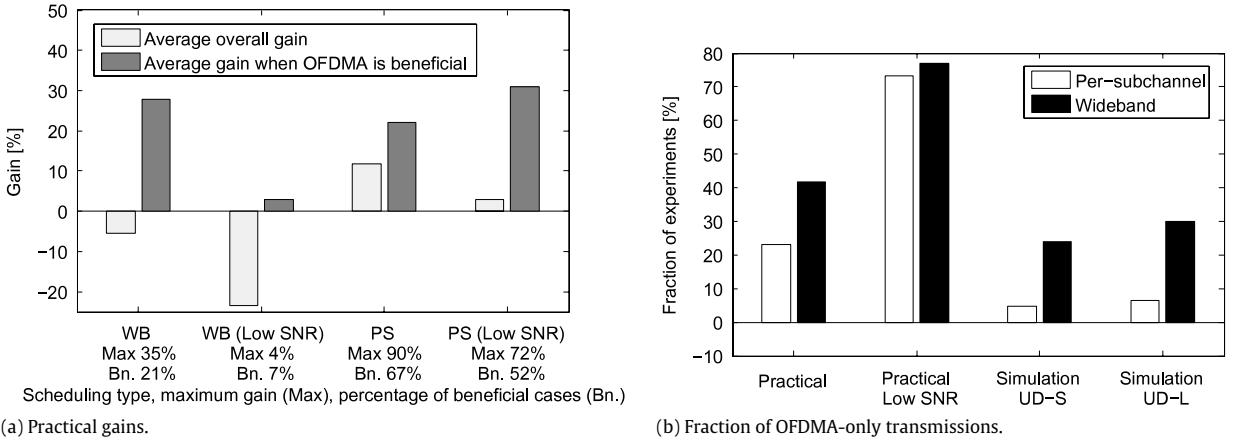


Fig. 14. Practical results. The diagram on the right refers to the fraction of experiments in which our baseline is not operable due to real-world channel conditions.

In Fig. 14(a) we show the OFDMA throughput gain. Strikingly, our OFDMA system yields *negative* gains when we compare it to our baseline that uses wideband scheduling. In other words, it performs worse than traditional hop-by-hop forwarding. The reason is the practical effect introduced in Section 4.3.4, that is, the challenge of estimating in practice the amount of Strider batches that the receiver needs to decode. Fig. 14(a) suggests that this effect is critical at low SNRs—most probably, the added noise hinders the estimation of the transmission duration. As a result, the throughput gain drops to -20% . In other words, the baseline performs better than OFDMA even though it is limited to wideband scheduling and thus cannot exploit frequency-selective link adaptation. From our above simulation results, we would expect the baseline to perform even better when switching to per-subchannel scheduling. However, in Fig. 14 we observe the opposite behavior—a baseline using per-subchannel scheduling yields a *positive* OFDMA gain. Essentially, in this case the baseline also suffers occasional mismatches regarding the estimated number of Strider batches. As a result, the baseline and OFDMA are on a par with respect to scheduling, revealing the actual spatial diversity gain of OFDMA. Other than the impact of estimating the transmission duration in practice, we come across similar effects than in simulation. For instance, per-subchannel OFDMA gains tend to become larger at lower SNRs, similarly to the simulation case. Further, we find a significant difference regarding the average gain of all experiments compared to the gain of the cases beneficial for OFDMA. That is, deciding whether OFDMA is helpful to transport data is also crucial in practice. Still, the increase in favorable cases when switching from wideband to per-subchannel scheduling is larger in practice than in simulation. Fig. 8 explains this behavior. Basically, in practice, OFDMA frames are often larger than baseline frames using wideband scheduling due to the inaccurate transmission time estimation.

Further, we observe a substantial difference between simulation and practice regarding the fraction of cases in which our baseline is not operable but OFDMA successfully delivers data. Most probably, the reason for such cases is deep fades in the channel transfer function, that is, subchannels at which multipath effects result in destructive interference for certain links. Such deep fades are more frequent in practice because our simulated channels are less frequency selective than the channels in the lab. While our baseline must use all subchannels on a link, OFDMA can allocate subchannels suffering a deep fade on a stage link to a different stage link. As a result, OFDMA can evade deep fades effectively, and is thus more robust. Our results in Fig. 14(b) show the aforementioned difference clearly. For instance, in the per-subchannel scheduling case, the fraction of experiments for which our baseline is not operable is only 6% in simulation but over 20% in practice. We observe a similar effect in the wideband case, which results in even larger fractions of OFDMA-only transmissions. Deep fades have a stronger impact in the wideband case because they affect all Strider segments instead of only the ones scheduled on the affected subchannels. In other words, a scheme not using OFDMA cannot operate *at all* in a significant number of cases. Hence, even though OFDMA may provide sometimes only moderate throughput gains, it is often essential to transport at least some data. This matches our results in Section 5.2. Finally, Fig. 14(b) shows that the fraction of OFDMA-only transmissions increases for lower SNRs both in practice and simulation. Essentially, the lower the average SNR, the higher the probability that at least one subchannel of a given link cannot transport data. Hence, the baseline becomes inoperable more often.

5.4. Discussion

Our results show that OFDMA for Corridor-based Routing is feasible in practice and improves throughput significantly. The overhead is not critical since we achieve large gains while accounting for all control messages our system needs for stage maintenance and coordination. We achieve large throughput gains even when using coarse CSI feedback. That is, gains do not stem from a fine-granular classification of subchannels according to their channel conditions, but just from coarsely identifying subchannels with very poor channel conditions. Further, corridors using OFDMA enable communication at SNR values at which a traditional forwarding scheme cannot operate at all, which highlights its robustness. While the throughput gain of OFDMA becomes larger for wider corridors, our experiments show that this effect is sometimes limited. The reasons

are (a) additional stage links only improve spatial diversity slightly, and (b) finding enough nodes to achieve a large stage width is challenging. Further, we conclude that the benefit of OFDMA strongly depends on how data splits and joins as it traverses the corridor. Hence, deciding a priori whether to use OFDMA is crucial.

When moving from OFDMA for WMNs in theory to a practical OFDMA system, we encounter a number of challenges. In particular, our results show that the combination of per-subchannel operation and the need to estimate the duration of transmissions is critical. Using individual subchannels is inevitable since OFDMA builds on top of frequency selective links. The problem is that imprecise estimation of transmission durations results in idle subchannels and thus wasted channel time. In other words, it hinders the scheduling algorithm to minimize the checkerbox area in Fig. 5. This issue raises for both discrete-rate and rateless approaches. The former causes idle subchannels due to retransmissions, while the latter results in such behavior due to additional batches. Our results show that improving the accuracy of transmission duration estimations to sort this issue out is challenging. To mitigate the impact of such inaccuracies in a scenario with multiple corridors, a stage could signal idle subchannels to nearby corridors that could use them. However, finding lightweight signaling is essential for such an approach.

In a scenario with multiple parallel unicast sessions, Corridor-based Routing constructs an individual corridor for each pair of communicating nodes. As a result, corridors may intersect and overlap, posing further challenges in terms of flow control and synchronization among corridors. Still, the overall OFDMA throughput gain does not depend on the corridor the data belongs to. Hence, the main insights of our evaluation can be translated to a scenario with multiple parallel transmissions. Further, the operation of intersecting and overlapping corridors raises the question whether OFDMA is the most suitable technique to support such transmissions. However, investigating such alternative mechanisms is beyond the scope of this paper.

6. Conclusion

We present a practical approach to enable OFDMA in Wireless Multihop Networks using Corridor-based Routing, which is a routing paradigm that supports state-of-the-art physical layer techniques. It widens traditional hop-by-hop paths in order to span a group of nodes at each hop and thus provide spatial diversity within an end-to-end path. Our OFDMA mechanism achieves performance gains in terms of throughput by allocating available subcarriers to those links providing good channel conditions. In contrast to previous practical work, we consider the amount of data that each node needs to forward. This mitigates potential bottlenecks. Moreover, we study the entire system at the physical, link, and network layers. We implement our mechanisms on software-defined radios and evaluate them in a multihop testbed. We observe that correctly estimating the transmission duration of a packet on an OFDMA subchannel is essential for scheduling, and that OFDMA can efficiently avoid subchannels affected by deep fades. Compared to traditional hop-by-hop forwarding, we achieve up to $2 \times$ throughput gain in the best cases, and $1.3 \times$ on average. We find that OFDMA gains strongly depend on CSI and traffic conditions. Hence, identifying disadvantageous cases in advance is crucial. Future work includes predicting whether OFDMA is beneficial for a certain data flow, investigating scheduling mechanisms that do not need to know the duration of packet transmissions, and designing flow control mechanisms for multiple parallel transmissions.

Acknowledgments

This work was supported by the LOEWE initiative within the NICER and COCOON projects, and by the German Research Foundation (DFG) as part of project C01 in the Collaborative Research Center (CRC) 1053 – MAKI.

References

- [1] A. Loch, M. Hollick, A. Kuehne, A. Klein, Practical OFDMA for corridor-based routing in wireless multihop networks, in: Proc. of LCN'14.
- [2] A. Kuehne, A. Klein, M. Hollick, Corridor-based routing using opportunistic forwarding in OFDMA multihop networks, in: Proc. of PIMRC'12.
- [3] A. Loch, M. Hollick, A. Kuehne, A. Klein, Corridor-based routing: Opening doors to PHY layer advances for wireless multihop networks, in: Proc. of WoWMoM 2014, WiP Session.
- [4] A. Kuehne, A. Klein, A. Loch, M. Hollick, Opportunistic forwarding in multi-hop OFDMA networks with local CSI, in: Proc. of the International OFDM Workshop'12.
- [5] A. Loch, P. Quesada, M. Hollick, A. Kuehne, A. Klein, Building cross-layer corridors in wireless multihop networks, in: Proc. of MASS'14.
- [6] S. Hoteit, S. Secci, R. Langar, G. Pujolle, A nucleolus-based approach for resource allocation in OFDMA wireless mesh networks, *Tran. Mob. Comput.* (2012).
- [7] K.-D. Lee, V. Leung, Fair allocation of subcarrier and power in an OFDMA wireless mesh network, *IEEE J. Sel. Areas Commun.* (2006).
- [8] M. Veyseh, J. Garcia-Luna-Aceves, H. Sadjadpour, OFDMA based multiparty medium access control in wireless ad hoc networks, in: Proc. of ICC'09.
- [9] M. Veyseh, J. Garcia-Luna-Aceves, H. Sadjadpour, Adaptive diversity based spectrum allocation in single-radio wireless ad hoc networks, in: Proc. of MASS'10.
- [10] S.-J. Kim, X. Wang, M. Madhian, Allocation in multi-hop OFDMA wireless networks with cooperative relay, *IEEE Trans. Wireless Commun.* 7 (2008).
- [11] G.M. Pan Zhou, B. Bing, Cross-layer congestion control and scheduling in multi-hop OFDMA wireless networks, in: Proc. of GLOBECOM'09.
- [12] M. Veyseh, J.J. Garcia-Luna-Aceves, Parallel interaction medium access for wireless ad hoc networks, in: Proc. of ICCCN'10.
- [13] M. Veyseh, J.J. Garcia-Luna-Aceves, H.R. Sadjadpour, Cross-layer channel allocation protocol for OFDMA ad hoc networks, in: Proc. of GLOBECOM'10.
- [14] M. Veyseh, J.J. Garcia-Luna-Aceves, H.R. Sadjadpour, Multi-user diversity in single-radio OFDMA ad hoc networks based on gibbs sampling, in: Proc. of MILCOM'10.
- [15] A. Richard, A. Dadlani, K. Kim, Multicast scheduling and resource allocation algorithms for OFDMA-based systems: A survey, *IEEE Commun. Surv. Tutor.* (2012).
- [16] Z. Chang, T. Ristaniemi, Energy efficiency of collaborative OFDMA mobile clusters, in: Proc. of CCNC'13.

- [17] Z. Chang, T. Ristaniemi, Efficient use of multicast and unicast in collaborative OFDMA mobile cluster, in: Proc. of VTC Spring'13.
- [18] J. Garcia-Luna-Aceves, H. Sadjadpour, Z. Wang, Challenges: Towards truly scalable ad hoc networks, in: Proc. of MobiCom'07.
- [19] S. Karande, Z. Wang, H. Sadjadpour, J. Garcia-Luna-Aceves, Optimal scaling of multicommodity flows in wireless ad hoc networks: Beyond the Gupta-Kumar barrier, in: Proc. of MASS'08.
- [20] P. Gupta, P. Kumar, *The capacity of wireless networks*, IEEE Trans. Inform. Theory (2000).
- [21] A. Kuehne, A. Loch, M. Hollick, A. Klein, Node selection for corridor-based routing in OFDMA multi-hop networks, in: Proc. of WCNC'13.
- [22] B. Gui, L. Dai, L. Cimini, Routing strategies in multi-hop cooperative networks, IEEE Trans. Wireless Commun. (2009).
- [23] L. Dai, B. Gui, L. Cimini, Selective relaying in OFDM multi-hop cooperative networks, in: Proc. of WCNC'07.
- [24] A. Gudipati, S. Katti, Strider: Automatic rate adaptation and collision handling, in: Proc. of SIGCOMM'11.
- [25] Andreas F. Molisch, *Wireless Communications*, second ed., Wiley, 2010.
- [26] R. Klose, A. Loch, M. Hollick, Evaluating dynamic OFDMA subchannel allocation for wireless mesh networks on SDRs, in: Proc. of SRIF'13.
- [27] P.A. Iannucci, J. Perry, H. Balakrishnan, D. Shah, No symbol left behind: A link-layer protocol for rateless codes, in: Proc. of Mobicom'12.
- [28] Rice University WARP Project, 2014. <http://warp.rice.edu>.
- [29] A. Loch, M. Schulz, M. Hollick, WARP Drive—Accelerating wireless multi-hop cross-layer experimentation on SDRs, in: Proc. of SRIF'14.
- [30] WARPLab Reference Design, 2014. <http://warplabproject.org/trac/wiki/WARPLab>.
- [31] A. Loch, T. Nitsche, A. Kuehne, M. Hollick, J. Widmer, A. Klein, Practical challenges of IA in frequency, Technical Report, TU Darmstadt, SEEMOO, 2013.