IEEE International Conference on Mobile Ad hoc and Sensor Systems, MASS 2014, Philadelphia, USA

Building Cross-Layer Corridors in Wireless Multihop Networks

Adrian Loch, Pablo Quesada and Matthias Hollick
TU Darmstadt, Secure Mobile Networking Lab
Mornewegstr. 32, 64293 Darmstadt, Germany
{*aloch, pquesada, mhollick*}*@seemoo.tu-darmstadt.de*

Alexander Kuehne and Anja Klein
TU Darmstadt, Communications Engineering Lab
Merckstr. 25, 64283 Darmstadt, Germany
{*a.kuehne, a.klein*}*@nt.tu-darmstadt.de*

***Abstract*—Corridor-based Routing widens traditional hop-by-hop paths to enable advanced physical layer mechanisms in Wireless Multihop Networks. The key concept is to let groups of nodes cooperate to jointly forward data. Instead of establishing a path formed by a fixed sequence of nodes, the network layer builds a "corridor" from source to destination to allow for such an approach. The corridor is divided into stages, which consist of the aforementioned groups of nodes. Existing mechanisms for Corridor-based Routing focus on the physical layer and assume a routing protocol that builds the corridor. In this paper, we design a corridor construction algorithm and a practical protocol that implements it. In particular, we address in detail the overhead introduced by our protocol, since this is crucial for performance and an open issue of Corridor-based Routing. We implement our approach both in simulation and practice. We obtain the turning point at which corridors become profitable and show that our protocol builds corridors, which enable throughput gains up to 74%.**

I. Introduction

For wireless infrastructure-based networks, advanced physical layer (PHY) techniques such as Orthogonal Frequency-Division Multiple Access (OFDMA) and Multiple-Input and Multiple-Output (MIMO) have shown to improve performance significantly. However, enabling such state-of-the-art PHY mechanisms in Wireless Multihop Networks (WMNs) is still a tough challenge due to the distributed and multihop nature of WMNs. Moreover, the coupling of advanced PHY mechanisms with routing is largely unsolved; an issue being that the channel state changes faster than the rate at which the network layer can forward control information. As a result, typical WMN implementations resort to basic PHY mechanisms that neither require coordination nor feedback among nodes at the PHY.

To address the aforementioned challenge, recent work proposes Corridor-based Routing [1], [2]. The underlaying main idea is to widen traditional hop-by-hop paths in order to span multiple nodes at each hop. As a result, instead of relaying packets from node to node, *groups of nodes* forward the packets, as illustrated in Figure 1. This provides spatial diversity along the corridor, which can be exploited by PHY techniques. For instance, OFDMA can use diversity in corridors to allocate fine-granular resources to the links with best channel conditions [2]. Two subsequent groups of nodes forming a "hop" in a corridor are called a *stage*. Nodes in a stage are fully connected. Hence, Channel State Information (CSI) required for PHY techniques used within a stage does not need to be shared over multiple hops. Also, stages can use existing one-hop techniques to implement CSI feedback efficiently. Previous work on Corridor-based Routing [2], [3] shows significant throughput gains in WMNs.

However, overhead can be an impediment for Corridor-based Routing, and gains in terms of throughput must be large enough to compensate for the effort required to set up and maintain the corridor. Today, this remains an open issue. While existing work deals with the selection of nodes that shall be part of the corridor [4], it does not consider the resulting overhead, let alone a protocol implementing the construction of a corridor. Still, the usefulness of the approach entirely depends on the incurred overhead. If the PHY operating on the corridor betters a traditional PHY in a hop-by-hop approach, after a certain amount of packets forwarded along the same corridor, the effort pays off. This raises a number of questions.

- How many packets are needed to achieve this turning point? Is the pay-off reached before the corridor has to be rebuild, e.g., due to node mobility?
- How do network characteristics such as density and connectivity impact the construction of corridors, and the resulting performance?
- How large is the overhead of corridor control messages? How many packets are needed and what information must they carry?

To answer the previous open questions, we design a protocol for corridor construction in WMNs. To this end, we combine the benefits of geographic and topological routing. First, we use location-based approaches to find a traditional hop-by-hop path to the destination *without* flooding the

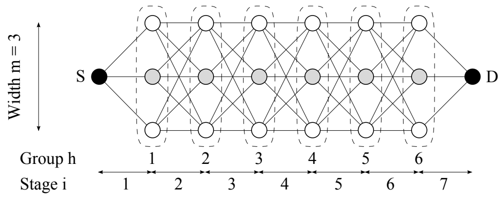


Figure 1. Corridor example [1] (regular placement for clarity reasons).

network with route discovery messages. Then, we widen the resulting path to a corridor of a given width in terms of nodes per stage. To guarantee that all nodes in a stage are fully connected, nodes locally exchange neighbor lists and identify suitable neighbors. If we lack suitable neighbors for a given corridor width, our protocol narrows the corridor and widens it again, when possible, at subsequent stages. In particular, our contributions are as follows:

1) We design a corridor construction algorithm that (a) allows to specify the intended width of the corridor, and (b) considers the CSI of potential corridor nodes.
2) We implement our approach in simulation and practice on software-defined radios (SDRs).
3) We analyze the overhead of our protocol and the turning point at which corridors pay off.
4) We evaluate our protocol on an OFDMA corridor to compare results with existing work.

This paper is structured as follows. In Section II we present related work. Then, in Section III we introduce our corridor construction algorithm, and in Section IV the protocol that implements it. After that, we delve into the evaluation of our protocol in Section V, both in practice and simulation. Finally, in Section VI we conclude the paper.

II. Related Work

Node cooperation mechanisms in WMN topologies, similar to corridors ([5], [6]), relay data exploiting spatial diversity. However, cooperation is often limited to selecting the best hop-by-hop path within a corridor-like structure. Reference [5] proposes three schemes to find such a path either having CSI of all stages, a certain number of next stages, or only the current stage. The scheme proposed in [6] allows per subcarrier path selection with OFDMA. That is, nodes cannot only choose their best outgoing link to relay data but also the best subcarriers within that link. In particular, [6] focuses on the outage performance of such a system. Mechanisms such as MIXIT [7] use random network coding to enable node cooperation. Essentially, such approaches adapt to the random nature of the wireless medium at the *network layer*. In contrast, Corridor-based Routing adapts the *physical layer* itself.

Corridor-based Routing enables nodes to (a) use advanced PHY techniques beyond simple path selection, and (b) combine multiple such techniques [1]. Previous work on corridors [2], [3] studies the specific case of OFDMA but, in contrast to [6], optimizes throughput and considers different degrees of CSI knowledge. Specifically, [2] investigates two algorithms for subcarrier and power allocation within a corridor assuming global CSI knowledge. While their first scheme follows a fair approach that allocates the same number of subcarriers to each node hop-by-hop, their second scheme assigns end-to-end resources in a greedy manner. In [3], authors solve a similar problem but only assume local CSI knowledge, i.e., nodes only know CSI of the stages they are part of. Thus, a node might receive more data than it can forward in the next stage. As a result, multiple transmissions might be needed. Both [2], [3] show that corridors using OFDMA can achieve throughput gains up to 50% compared to hop-by-hop routing. However, they assume a given corridor structure.

Corridor building remains an open issue. While [8] solves a problem named similarly, it is important not to confuse it with the challenges we deal with. The authors of [8] propose a scheme to form a set of *individual paths* which they also call a “corridor”. However, they neither form groups of cooperating nodes at the lower layers nor provide support for advanced PHYs. Their goal is to find a set of traditional paths of the same length between two nodes. To this end, [8] follows an AODV-based approach. Data packets are arbitrarily sent via the available paths to minimize the impact of potential path disruptions. Hence, it clearly stands apart from our approach. In contrast, [4] selects nodes in random WMNs to form structures suitable for Corridor-based Routing. Their algorithm first establishes a traditional path using geographic routing, which is then widened by means of geometric criteria. In particular, each node of the path selects as corridor nodes all those nodes located in a circular neighborhood around itself and pointing towards the destination. While we draw on this approach, we deal with a number of crucial practical issues not considered in [4]. First, we study the overhead caused by corridor construction. Second, we perform fine-granular node selection based on the quality of links and the intended width of the corridor. Third, we analyze our scheme on multiple networks with different density and connectivity. Finally, we validate our approach in practical terms on an SDR testbed.

Mobile backbones also structure WMNs [9], [10]. Still, they only provide limited help for constructing corridors. While the goal of mobile backbones is to introduce hierarchy into the network to reduce routing overhead and improve scalability, we focus on enabling cooperation among nodes. Mobile backbones use clustering [11] to form groups of nodes, generally without taking into account how packets flow in the network. In contrast, corridors form groups of nodes to transport data in a certain direction. In [9], only a limited set of nodes defined in advance can be part of the backbone, while in our case any node can join the corridor. Additionally, guaranteeing backbone connectivity becomes an issue in such approaches [10]. Segment routing [12] also clusters nodes, but then uses a combination of geographic and topological routing to relay data among clusters. Still, communication takes place via the backbone nodes and does not exploit cooperation among the cluster nodes.

Routing in WMNs is the basis of our construction approach. For details, we refer the reader to the literature and survey work [13], [14], [15] in this area. This includes work on multipaths [16], which however does not provide node cooperation as corridors do.

III. Corridor Construction Algorithm

Our corridor construction algorithm operates on WMNs with randomly distributed nodes. We consider an indoor scenario where nodes move at most at human speed. In Section III-C2 we sketch out how a corridor can deal with moving nodes, but otherwise we focus on the worst-case scenario, i.e., the complete structure needs to be rebuilt if nodes move. Our algorithm builds a corridor from a source S to a destination D formed by groups of nodes that forward data along multiple stages—we index the former with h and the latter with i, as shown in Figure 2. Each group has a "main" node h_m that coordinates nodes in its group. The remaining nodes of a group are called "stage neighbors". We build corridors based on two parameters. First, the *desired* corridor width m, which might be infeasible in some stages. Second, the minimum link quality SNR_{min}, which is the minimum average signal-to-noise ratio (SNR) all the links in the corridor must have.

A. Neighbor discovery

Each node broadcasts periodic beacons for neighbor discovery, including its node identifier and location. At the PHY, receiving nodes use the pilot symbols in each beacon to estimate CSI to the transmitting node.

B. Corridor construction

We build the corridor in two phases. First, we find a hop-by-hop *unipath* from S to D using geo-routing. Using this path as the initial set of h_m nodes, we then widen it to a corridor using a topological approach.

1) Unipath: Our unipath mechanism is inspired by well-known geographic routing approaches [13]. However, we do not use it to transport data packets but to find a unipath, i.e., we use this approach with a different goal than typical geo-mechanisms. Source node S—and all subsequent nodes C selected as part of the path in this phase—run Algorithm 1 to find the next node. Essentially, each node C forms a set T of neighbors N which are suitable to be the next node of the path. In particular, T contains all neighbors who (a) are not part of the path already, (b) have a link to the current node C with an SNR larger than SNR_{min}, and (c) are located in the direction of the destination D. A neighbor N lies in the direction of D if the angle between $\overrightarrow{CN}$ and $\overrightarrow{CD}$ is less than $180°$. Finally, node C chooses the neighbor in T which lies closest to the direction of D. If no suitable nodes were found, i.e., $T = \emptyset$, there is a local minima. To escape it, the algorithm can use existing mechanisms from geographic routing. However, we consider such mechanisms to be out of scope of this paper.

Algorithm 1 Single path construction algorithm.

```
Require: Location of neighbors $N$ of current node $C$,
    location of $D$, nodes $P$ already part of path, $\text{SNR}_{\text{min}}$
Ensure: Find next node of a hop-by-hop path to $D$
set $\alpha$ to angle of $\overrightarrow{CD}$ with horizontal axis
for all neighbors $N$ of $C$ do
    set $\beta_N$ to angle of $\overrightarrow{CN}$ with horizontal axis
    if $|\beta_N - \alpha| < \pi$ and $\text{SNR}_N > \text{SNR}_{\text{min}}$
    and $N \notin P$ then
        add $N$ to set $T$ of suitable neighbors
    end if
end for
if $T \neq \emptyset$ then
    return neighbor $N \in T$ with minimum $|\beta_N - \alpha|$
else
    use existing mechanisms to escape local minima
end if
```

2) Widening paths: Next, we define an algorithm to widen the unipath to a corridor hop by hop. This process starts at the destination once the last hop of the unipath is found and builds one stage at a time towards the source. For each stage, the algorithm must ensure:

- All nodes of two subsequent groups forming a stage must be one-hop neighbors (Figure 1).
- If possible, the number of nodes in each group shall be equal to the desired width m.

Algorithm 2 uses a topological approach to meet these requirements. The construction of a generic stage is shown in Figure 2. Our algorithm involves three steps: (a) neighbor list collection, (b) neighbor matching, and (c) node selection. Node h_m coordinates these three steps for the construction of stage i, which connects groups h and $h-1$. In **step (a)**, h_m collects the list of neighbors of the two next nodes along the unipath, $(h-1)_m$ and $(h-2)_m$. Additionally, h_m requires the two-hop neighbors of $(h-1)_m$ and itself to verify that links are symmetric, e.g., in Figure 2 node h_m needs the list of neighbors of X to verify that X can both send to and receive from Y, Z, h_m, $(h-1)_m$, and $(h-2)_m$. In **step (b)**, h_m removes all links with SNR $< \text{SNR}_{\text{min}}$ from the lists and then finds common nodes in all lists. This ensures that the stage being constructed is fully connected. The number of common nodes $|W|$ leads to three possible cases in **step (c)**, which we discuss next.

First, if $\boldsymbol{|W| = m - 1}$, we have exactly the number of nodes needed to form a stage of width m. Hence, all nodes in W are selected as stage neighbors at stage i. Second, if $\boldsymbol{|W| < m - 1}$, we lack one or more nodes to build a stage of

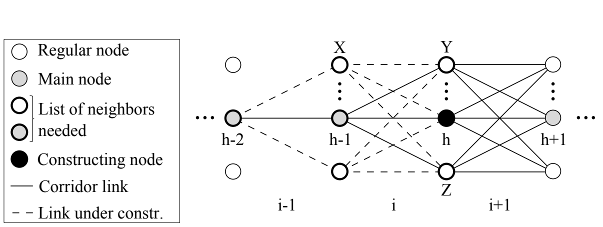


Figure 2. Generic stage construction for an intermediate stage.

width m. However, changing the main node of group $h-1$ may still allow the algorithm to find enough stage neighbors. For instance, in Figure 3a, node X is the main node of $h-1$ and only has node Y as a suitable stage neighbor. Node Z, which is a neighbor of Y and of all nodes in group h, is not a neighbor of X. Since the main node must be a one-hop neighbor of all stage nodes to enable coordination, Figure 3a only allows for a corridor of width two. Hence, in this case the algorithm checks whether any node in W allows for the desired stage width if set as the main node of $h-1$. In our example, this is the case for node Y, which is thus set as the main node in Figure 3b. Such stage reorganization is also possible for wider corridors—the general condition making this possible is given in line 7 of Algorithm 2. If changing the main node does not allow for the desired width, we narrow the corridor at stage i. Third, if $|\boldsymbol{W}| > \boldsymbol{m-1}$, we have found more nodes than needed to achieve the desired width. In this case, the algorithm chooses the $m-1$ nodes with best average SNR. However, the excess of suitable nodes also allows for optimizations (c.f. Section III-C1).

C. Extensions

1) OFDMA optimization: If the PHY to be used in a corridor is known in advance, Algorithm 2 can choose the nodes which are most beneficial for it. As a selected example, we present such an optimization for OFDMA. In OFDMA, the links at each stage share the available subcarriers. The scheme performs best when subcarriers with poor channel conditions on one link, are good on a different link and vice-versa, that is, when each subcarrier has good channel conditions on at least one link of the stage. Hence, the optimization chooses the nodes for stage i that result in the most *diverse* links. To this purpose, we compute the per-subcarrier SNR of each possible OFDMA link combination. We then choose the combination which (a) has the best average SNR, and (b) is most similar in terms of SNR over all subcarriers. This requires knowing the stage neighbors of the preceeding and subsequent groups to the currrent one. Thus, the regular construction process leaves groups with more nodes than needed "pending", and optimizes them as soon as the rest of the corridor is built.

Algorithm 2 Stage construction according to Figure 2.

```
Require: List of neighbors of all nodes potentially part of
    the current stage $i$ being built.
Ensure: Find suitable nodes for the current stage $i$.
 1: node $h$ requests list of neighbors of
      1) Stage neighbors of $h$
      2) Stage neighbors of $(h-1)_m$ (symmetry)
      3) Main node of $h-1$
      4) Main node of $h-2$ via $(h-1)_m$
 2: filter out links in lists with SNR < $\text{SNR}_{\text{min}}$
 3: find coincidences $W$ in all filtered lists
 4: if $|W| = m-1$ then
 5:     select all nodes $\in W$ as stage neighbors for $i$
 6: else if $|W| < m-1$ then
 7:     if $\exists n \in W \;/\; n$ is neighbor $\forall W \wedge n$ has $m-1-$
        $|W|$ list coincidences $W'$ with $h,\ h-2$ then
 8:         set $n$ as main node of stage $i$
 9:         set nodes $\in W \cup W'$ as stage neighbors for $i$
10:     else
11:         narrow corridor at stage $i$
12:     end if
13: else
14:     select $m-1$ nodes $\in W$ with best average SNR
15: end if
```

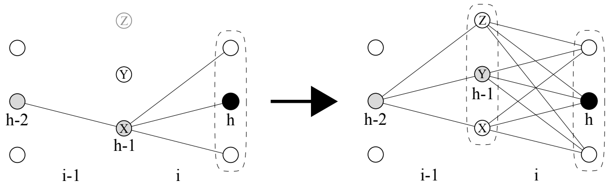
2) Corridor adaptation: Corridor nodes keep track of the periodic beacons they receive from neighbors. If the main node of group h does not receive beacons from one or more nodes of group $h-1$ for a certain time interval due to, e.g., mobility, it locally reconstructs stage i using Algorithm 2. If the main node of group $h-1$ is unavailable, h_m also reorganizes stage i as in Figure 3. That is, if a node moves, corridors can adapt *without* reconstructing the full structure. Still, in our experiments we evaluate the overhead of corridors assuming such local adaptation is not possible.

IV. PROTOCOL

To put our approach into practice, we define a protocol and specify control messages to coordinate nodes. Table I gives an overview. Due to space constraints, we only explain the details of some selected control messages.

Unipath. During unipath construction, each node on the path forwards the unipath control message to the next node it chooses based on Algorithm 1. Figure 4 depicts the details of this message. It contains the location of the destination and the corridor parameters required at the destination to start the corridor construction once the unipath is finished. This includes the "O" field, which indicates whether the corridor shall be optimized according to Section III-C1. Each node forwarding the packet adds its identifier to the list of nodes at the end of the message. This prevents nodes from selecting a neighbor which is already part of the path as a next hop. For our prototype, we encode the node identifier with six bits. Also, each node increases by one the path length field.

List Reply. Figure 5 shows the details of the "List Reply" message, which carries the lists of neighbors needed at the



(a) Stage width limited to two

(b) Stage width up to three

Figure 3. Stage reorganization. Nodes X and Z are not direct neighbors.

main nodes to construct each individual stage. Similarly to the unipath message, it includes a list of all nodes which are already part of the corridor, to prevent the algorithm from adding twice the same node to the corridor. A "List Reply" message can aggregate answers from multiple nodes in case they are relayed—for instance, in Figure 2, $(h-1)_m$ relays the list of neighbors of $(h-2)_m$ and of its own neighbors. The field "# Lists" indicates how many lists the reply aggregates. Further, the message includes the node identifiers involved in the list reply, as well as certain corridor parameters needed for correctly processing the reply.

Stage ACK. After running Algorithm 2, the main node building the current stage sends "Stage ACK" messages (Figure 6) to inform all potential stage neighbors whether they are part of the stage. This is conveyed by the binary field "P". Besides, the "T" field is set to one if the node receiving the acknowledgment shall be the main node. The message also includes the group the receiving node is part of, as well as the identifier of the main node of its group and that of the next group. As a result, each node is aware of its position and function within the corridor under construction.

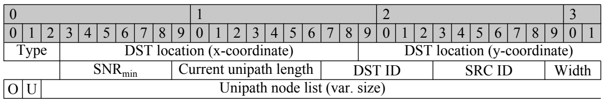


Figure 4. Control message structure for unipath construction.

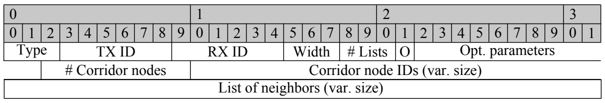


Figure 5. Control message structure for "List Reply".

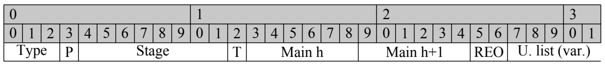


Figure 6. Control message structure for stage acknowledgments.

Table I
CONTROL PACKETS FOR CORRIDOR CONSTRUCTION

Packet type	**Contents and function**
Beacon	Node ID and location.
Unipath	Message for unipath construction. Includes destination ID, list of nodes already in unipath, and SNR$_{\text{min}}$.
List Request	Used by h_m to request neighbor list of a node.
List Reply	Answer to "List Request". Includes ID and average SNR of each neighbor of the node sending it.
Stage ACK	Notifies stage neighbors that the main node building the current stage has chosen them as part of the stage.
Corridor ACK	Final confirmation sent by the main node of each stage after the end-to-end corridor is complete.
Optimization	Sent after corridor construction to optimize "pending" stages. For OFDMA, includes per-subcarrier SNRs.

V. Evaluation

We evaluate our system both in simulation and practice. We perform practical experiments on an SDR testbed. Further, we simulate scenarios our testbed cannot recreate.

A. Experimental design

Goal. Our evaluation aims at (a) verifying that our construction mechanism is practicable regarding overhead, and (b) measuring the quality of our corridors. The corridor quality is directly related to the performance advanced PHY techniques achieve when using it, hence, we use this performance as a metric to test the corridor quality. In the literature, corridors are analyzed primarily in combination with OFDMA [2], [3], and throughput gains are known to lie in between 20% and 50% compared to hop-by-hop forwarding based on Orthogonal Frequency-Division Multiplexing (OFDM). Hence, we choose OFDM as a baseline, and OFDMA as a corridor mechanism for our experiments.

Metrics. We choose metrics that allow us to capture the overhead of corridor construction. We use (a) the throughput gain of OFDMA compared to OFDM *including* overhead, (b) the number of nodes participating in the corridor, and (c) the time required to build a corridor. To measure (a), we consider the data that the network layer delivers at the destination and the time needed for the transmission, including the time required to build the corridor. We obtain this time by counting the PHY symbols needed for signaling. Channel contention for signaling packets can be done efficiently since nodes within a corridor are already synchronized at PHY.

Measuring process. All our experiments follow the same pattern. We choose a random source-destination pair, which we use to transmit multiple packets both with OFDM and OFDMA. At the beginning of the transmission, we build either a unipath or a corridor. Still, this construction process is only needed for the *first* packet, since subsequent packets can use the same routing information as long as nodes do not move. That is, only the first packet is affected by the construction overhead. For that packet, we expect OFDMA to perform worse than OFDM, since the effort required to build a corridor is significantly harder than the one required for a unipath. However, once the corridor is available, OFDMA should perform better. Hence, after a certain number of packets, the initial penalty of OFDMA due to the corridor construction should pay off. Additionally, we use a basic adaptive modulation and coding scheme (MCS). At each OFDM link or OFDMA stage, the modulation and coding in use are improved after each successful transmission, until an error occurs. After an error, the MCS is not increased any further. Thus, the throughput initially increases with each packet, and later stabilizes when the link or stage reaches the maximum MCS it supports. Since OFDMA can combine beneficial subcarriers, we expect it to reach a higher MCS than that of OFDM, which directly translates into throughput gain. For OFDMA, we use the same MCS on all subcarriers.

Assumptions. We consider that nodes move randomly at human speeds, that is, at about 1.5 m/s. Moreover, we assume a number of worst-case conditions. First, we do not allow the corridor to reconstruct stages locally but we rebuild the entire structure when a node moves. Second, we use the aforementioned basic adaptive MCS. A more advanced approach could use feedback to estimate the maximum MCS and thus further improve corridor performance. Finally, we do not use the optimization introduced in Section III-C1, which would again improve the performance of OFDMA.

Scenarios. We test our corridor construction mechanism in four different settings: three simulated scenarios and one SDR testbed scenario. Table II gives an overview. The simulated scenarios allow us to investigate the performance of our system for large networks and diverse situations. The first one is a WMN with 50 nodes deployed randomly on an area of 250 m^2, named as **RAND**. We assume nodes are in each other's range if they are less that 12 meters far apart. At the PHY, we assume Rayleigh channels, a path loss exponent of $\alpha = 2$, and additive white Gaussian noise (AWGN) with 70 dB average SNR at a distance of one meter to a transmitter. Due to the random location of nodes, the network topology may have "holes". However, indoor networks with a large number of connected devices may be very dense. Thus, our second scenario considers a similarly large area with a uniform density, named as **UD-L**. While nodes are still placed randomly, we ensure that there is at least one node every ten meters. As a result, this scenario features high connectivity. Since our corridor construction scheme takes into account the average strength of links, our third scenario considers stronger links than RAND and UD-L. To this end, we place about 50 nodes with uniform density in a smaller area, named as **UD-S**. We lower the range, achieving a connectivity slightly lower than UD-L. Finally, our last scenario **TBED** is an SDR testbed which allows us to test our system in practice. We place 20 nodes at fixed locations in our lab as depicted in Figure 7. Since we cannot change the topology randomly for each experiment run, we choose a regular pattern to better understand our results. We adjust the transmit power of nodes to achieve typical indoor SNRs and discard packets on links which exceed 2.7 meters. As an example, Figure 7 shows which nodes are one-hop neighbors of node A for that range value. This setup allows corridors up to a width of four despite the small testbed size.

Table II
SIMULATIVE AND PRACTICAL SCENARIOS.

Name	Type	Area and Nodes	Range	Characteristic
RAND	Sim.	$50 \times 50\, m^2$, 50	12 m	Random location
UD-L	Sim.	$50 \times 50\, m^2$, 50	15 m	High connectivity
UD-S	Sim.	$21 \times 21\, m^2$, 49	6 m	Strong links
TBED	Pract.	$4.5 \times 3\, m^2$, 20	2.7 m	SDR testbed

B. Implementation

Framework. We implement our corridor construction mechanism on WARP Drive [17], which is a framework that facilitates prototyping of wireless multihop mechanisms on the Wireless Open-Access Research Platform (WARP) [18] SDR. It builds on the WARPLab Reference Design that allows to carry out over-the-air experiments with WARP directly from Matlab. We process signals in Matlab, send them via Ethernet to the WARP boards, transmit them over the wireless medium, and transfer them back to Matlab. While this results in non-real-time operation, we do *not* process data offline, but online and *interactively*. That is, over-the-air transmissions are not scheduled in advance but occur dynamically as a result of protocol actions. WARP Drive supports such operation by design. Moreover, it switches seamlessly from simulations to practical experiments, i.e., the *same code* is used for both cases.

OFDMA. In practice, OFDMA requires (a) accurate synchronization of transmitters, and (b) careful power control of simultaneous transmissions. Since we do not aim at demonstrating the feasibility of OFDMA but at studying the overhead of corridor construction, we use a simplified implementation of OFDMA. To address (a), we use wired synchronization to avoid carrier-frequency offset (CFO) in our testbed. Regarding (b), we allow OFDMA transmitters in a stage to send sequentially instead of simultaneously. Still, each transmitter only uses its assigned subcarriers. We then combine the sequential signals at the receivers and compute throughput as if transmissions had been simultaneous. This approach allows us to focus on the performance of corridor construction. For the allocation of subcarriers within each stage, we use mechanisms similar to [2], [3].

Practical setup. Our testbed consists of six WARPv3 boards with four radio interfaces each. To allow for a network with 20 nodes, WARP Drive enables us to use each interface as if it were an individual node. Still, we treat the data of each radio independently. At the PHY, we use parameters equivalent to 802.11g, i.e., 20 MHz channels, 48 usable subcarriers, 312.5 KHz subcarrier spacing, and 12.5% cyclic prefix (CP). Moreover, we use 802.11 short and long preambles—the former trains the Automatic Gain Control (AGC) and the latter allows for frame detection.

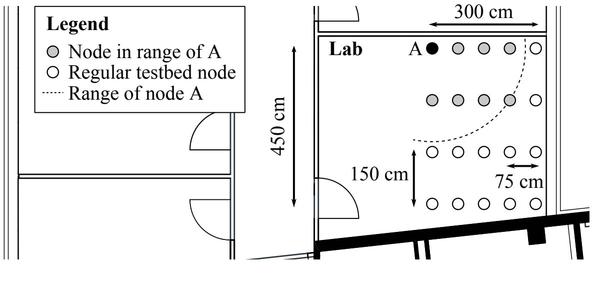


Figure 7. Practical setup for scenario TBED using SDRs.

C. Simulation

1) Construction time: Figure 8 shows that corridor construction takes significantly longer than finding a unipath. Construction time is roughly proportional to the length of the corridor/unipath, since building each additional stage/hop requires control messages. For the unipath case, we observed that the scenario barely influenced the construction time, since the algorithm only needs to choose one suitable next node. Corridor construction in UD-L is slightly slower than in RAND and UD-S. For RAND, the reason is the sparsity of the network in some areas, which results in less control messages, as nodes need to exchange less lists of neighbors to build a stage. For UD-S the reason is that its strong links allow to transmit control messages at higher rates, and thus the overall time to build the corridor is less than that of UD-L despite having similar connectivity. Finally, the corridor width and the SNR_{min} value barely impact the corridor construction time, since these parameters only determine which nodes are chosen *after* requesting the lists of neighbors. Thus, the number of exchanged control messages is mostly independent of m and SNR_{min}.

2) Turning point: Next, we analyze after how many packets the construction overhead compensates. Figure 9 depicts the average end-to-end throughput of 40 subsequent packets sent on a unipath (OFDM) and a corridor (OFDMA). For all cases, the curve initially raises due to the adaptive MCS. In UD-S, the curves stabilize at a higher value than in RAND and in UD-L since links are stronger and support a higher MCS. In all scenarios, OFDMA performs initially worse than OFDM due to the cost of corridor building for the first packet. Still, because OFDMA combines the best links at each stage, the adaptive MCS scheme can reach a higher rate, and thus OFDMA outperforms OFDM after a few packets. The vertical lines indicate the turning point, i.e., the time at which OFDMA exceeds OFDM regarding *cumulative* throughput. This occurs later than the intersection of the OFDM and OFDMA curves since OFDMA needs to compensate for its initial overhead before paying off.

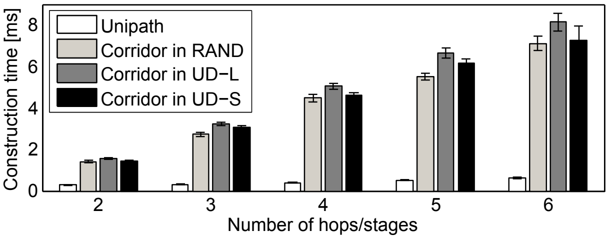


Figure 8. Time required for constructing corridors in each scenario.

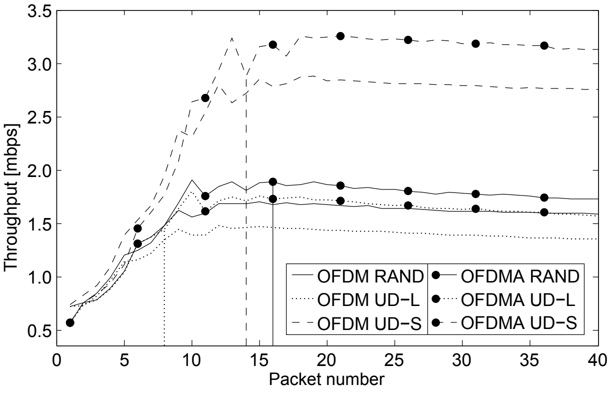


Figure 9. Turning point. The vertical lines indicate from when on the corridor compensates. Lines with markers refer to the corridor case.

RAND and UD-S reach the turning point at packets 14 and 16, respectively, while UD-L compensates for the overhead already at packet 8. Although the turning points of RAND and UD-S are similar, the reasons behind are different. For RAND, the uneven density of the network forces the algorithm to narrow the corridor more often than in UD-S and UD-L. The resulting corridors provide less spatial diversity and thus the gain of OFDMA is smaller, which translates into OFDMA needing more packets to pay off. In contrast, UD-S takes longer than UD-L to reach the turning point due to the adaptive MCS—since UD-S supports a higher MCS, the link layer needs more packets to reach it, hence delaying the turning point. Finally, UD-L reaches the turning point first because (a) the link layer finds the maximum MCS quickly, and (b) UD-L features higher connectivity than RAND, thus avoiding narrow corridors.

3) Degree of construction: Our previous experiment indicates that the performance of a corridor is closely related to the construction algorithm being able to find enough stage neighbors for each stage. We refer to this as degree of construction (DOC), which is the fraction of nodes a corridor has compared to the number of nodes it should have given its width. For instance, the ideal corridor in Figure 1 has 20 nodes. If the algorithm needs to narrow some stages resulting in a corridor of 15 nodes, the DOC is 75%. Figure 10 shows the throughput of OFDMA for different DOC ranges. As expected, it improves for larger DOCs, since diversity increases. Figure 11 depicts the distribution of the DOC for the case $m = 3$. In RAND, most corridors only achieve a DOC of 60% or less due to the low connectivity of the network. In contrast, in both UD-L and UD-S, our algorithm achieves in most cases DOCs up to 80%. UD-S has a lower DOC than UD-L due to its slightly lower connectivity.

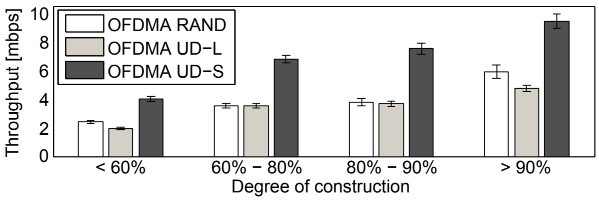


Figure 10. Throughput for different degrees of construction.

4) Width and SNR_{min}: Some effects regarding corridor width and $\mathrm{SNR}_{\mathrm{min}}$ are contrary to each other, which complicates drawing general conclusions. For instance, a larger **width** results in more diversity as more nodes participate in each stage, and thus OFDMA can choose better subcarrier allocations. However, since our allocation strategy is "fair", we assign the same number of subcarriers to each link in a stage. The more links used, the higher is the probability of a link having poor channel conditions, thus limiting the performance of the stage. Regarding $\mathbf{SNR_{min}}$, we encounter a similar trade-off. A large $\mathrm{SNR}_{\mathrm{min}}$ prevents our algorithm from including bad links in a certain stage, thus improving the performance of OFDMA. Yet, this also affects unipath construction, resulting in better OFDM performance and thus lowering gains. Overall, this limits our gains to about 15%. We address this issue in the practical case, since the aforementioned effects became critical on the SDR testbed.

D. Practical experiments

1) Limitations: The limitations outlined in Section V-C4 had a significant impact on our initial testbed experiments—our OFDMA corridors provided no gains compared to OFDM. Figure 12 depicts a sample of an actual channel measurement from our testbed. We observe two key limitations, namely, (a) the aforementioned fair allocation of subcarriers forces us to use poor links, and (b) the worst subcarrier in a stage imposes the maximum MCS since all subcarriers use the same rate. Although these issues also affected our simulations, they became more critical in practice. The reason is that the channels in our lab feature less reflections than in our simulations, and are thus less frequency selective. Due to the fair allocation strategy, the probability of assigning a poor link to a subcarrier became much larger in the practical case. Still, Figure 12 shows that our testbed does feature significant frequency selectiveness.

2) Implementation changes: To solve the above issues, we adapted the OFDMA allocation mechanism for our practical experiments. To deal with effect (a), we allow each link to get a different amount of resources according to its channel conditions and to the amount of data each node in a stage needs to forward. Further, we counteract (b) by allowing each subcarrier to use a different MCS. We design an allocation mechanism that minimizes the transmission time at each stage.

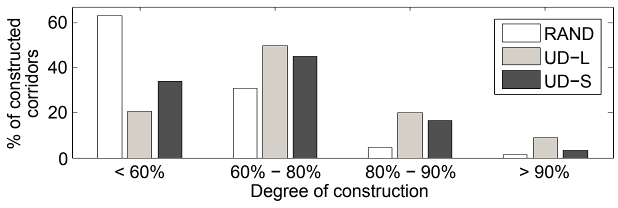


Figure 11. Degree of construction achieved for corridors of width three.

3) Results: Figure 13 depicts the throughput in our testbed of both OFDM and OFDMA for different widths and values of $\mathrm{SNR}_{\mathrm{min}}$. We achieve throughput gains mostly between 20% and 50%, which matches the results of related work. According to Section V-A, we conclude that our construction algorithm builds good corridors. Also, these results confirm our observations in Figure 12. The throughput of OFDM varies in each of the three measurements although it does not depend on the corridor width. The reason is that channels in our lab fluctuate slowly during the course of the day, leading to different results for the same $\mathrm{SNR}_{\mathrm{min}}$. Still, channels are comparable for each corridor width. We observe that the throughput of OFDM increases with $\mathrm{SNR}_{\mathrm{min}}$ since this parameter also affects the nodes chosen for unipaths. In contrast, OFDMA achieves similar throughput for all $\mathrm{SNR}_{\mathrm{min}}$ values. This shows the high degree of flexibility of OFDMA—due to the improved allocation scheme (c.f. Section V-D2), OFDMA becomes highly adaptive, and thus weak links in a stage barely make any impact on it. Regarding corridor width, we observe that throughput decreases slightly with larger stages. The reason is the incurred overhead for *OFDMA operation*, i.e., not the overhead for corridor construction, which is to a large extent independent of corridor width (c.f. Section V-C1). OFDMA exchanges CSI among all nodes of a stage. Hence, the larger a stage, the larger its overhead. For corridors of width four, this overhead even exceeds the gains of OFDMA, resulting in a negative throughput gain. Moreover, Figure 13 illustrates that the diversity available in a corridor of width two already allows for large gains. In other words, the corridor structure does not need to be wide to provide large benefits.

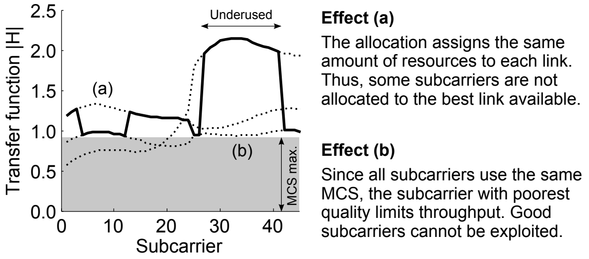


Figure 12. Testbed channels. The thick line is the OFDMA allocation.

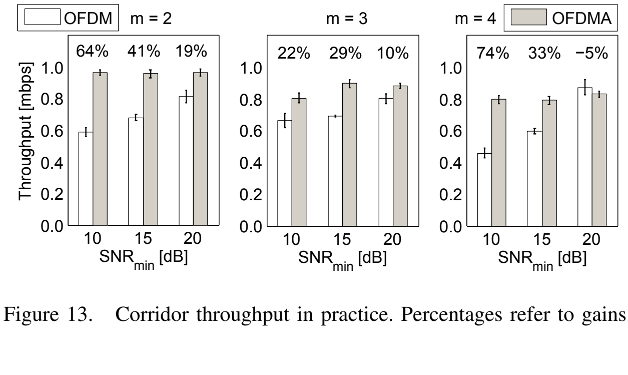


Figure 13. Corridor throughput in practice. Percentages refer to gains.

E. Discussion

In Section I we identified a number of open questions regarding the practicability of corridor construction in WMNs, and our results provide some answers. First, the **construction overhead** of a corridor depends on the length of the corridor, and increases faster per stage than a unipath does per hop. For example, for a corridor of length five, the overhead is about ten times larger than that of a unipath; it consists mostly of control packets conveying neighbor lists. This leads directly to the second issue of compensating for this overhead (**turning point**) before the corridor fails. In our experiments, the slowest scenario needed an average of 16 packets to reach this point. Assuming a packet size of 1500 bytes, a worst-case throughput of 1 mbps, and a corridor length of five stages, 16 packets require 0.96 seconds to reach the destination. Since we consider a range of six meters for our smallest scenario, on average a node needs to move three meters to leave a corridor. Hence, this requires nodes to move at about $3\,\text{m}/0.96\,\text{s} = 11.25\,\text{km/h}$ for the corridor overhead not to compensate, which is significantly larger than the average human speed. This intuitive estimation becomes more beneficial assuming more realistic throughputs, opening doors to using our construction scheme also in outdoor scenarios featuring higher mobility. Still, determining the trade-offs in such a scenario requires further practical experiments. Third, we also study the influence of **topology characteristics**. As expected, dense and connected networks result in better corridors. The less nodes are available, the more often our algorithm has to narrow a corridor. However, similarly to [4], we observe that corridors do not need to be wide to provide large gains, i.e., also sparse networks can benefit from corridors. For OFDMA, we conclude that corridor widths up to three nodes and small SNR_{min} values are best. However, SNR_{min} should be above a minimum to avoid links with bad channel conditions on all subcarriers. Finally, regarding the **operation of OFDMA**, our practical experiments show that a non-fair allocation of subcarriers to links in a stage enables high adaptability to the channel, which is crucial in wireless communication.

VI. Conclusion

We present an algorithm for corridor construction in WMNs that use Corridor-based Routing, which is a routing paradigm that widens traditional paths to exploit spatial diversity using state-of-the-art PHYs. In particular, we study whether the overhead required to build a corridor structure compensates. We evaluate our algorithm both in simulation and practice using an SDR testbed. To investigate the quality of the resulting corridors, we assess their performance for the case of OFDMA. Our results show that the effort required for corridor construction compensates if the corridor can be used for multiple packets. We analyze the number of packets needed to reach this turning point. Assuming nodes move at human speed, corridors are practicable in all scenarios we consider. We solve practical issues regarding frequency selectivity, and show that our algorithm builds corridors that allow up to 74% throughput gain when using OFDMA.

Acknowledgment

This work was supported by the LOEWE Priority Program Cocoon, the DFG CRC 1053 MAKI and LOEWE CASED.

References

[1] A. Loch, M. Hollick, A. Kuehne, and A. Klein, “Corridor-based Routing: Opening Doors to PHY Layer Advances for Wireless Multihop Networks,” in *Proc. of WoWMoM’14, WiP Session*, 2014.

[2] A. Kuehne, A. Klein, A. Loch, and M. Hollick, “Corridor-based Routing Using Opportunistic Forwarding in OFDMA Multihop Networks,” in *Proc. of PIMRC’12*, 2012.

[3] ——, “Opportunistic Forwarding in Multi-Hop OFDMA Networks With Local CSI,” in *Proc. of the International OFDM Workshop ’12*, 2012.

[4] A. Kühne, A. Loch, M. Hollick, and A. Klein, “Node Selection for Corridor-based Routing in OFDMA Multihop Networks,” in *Proc. of WCNC’13*, 2013.

[5] A. Bletsas, A. Khisti, D. Reed, and A. Lippman, “A simple cooperative diversity method based on network path selection,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, 2006.

[6] L. Dai, B. Gui, and L. Cimini, “Selective relaying in OFDM multihop cooperative networks,” in *Proc. of WCNC’07*, 2007.

[7] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, “Symbol-level Network Coding for Wireless Mesh Networks,” in *Proc. of SIGCOMM’08*, 2008.

[8] C. Vogt, M. Gerharz, and C. Waal, “Corridor Routing in Mobile Wireless Ad-hoc Networks,” in *Proc. of the 3rd Mediterranean Ad Hoc Networking Workshop*, 2004.

[9] K. Xu, X. Hong, and M. Gerla, “An ad hoc network with mobile backbones,” in *Proc. of ICC’02*, 2002.

[10] S. Chu, P. Wei, X. Zhong, X. Wang, and Y. Zhou, “Deployment of a Connected Reinforced Backbone Network with a Limited Number of Backbone Nodes,” *Trans. on Mobile Computing*, vol. 12, no. 6, 2013.

[11] J. Yu and P. Chong, “A survey of clustering schemes for mobile ad hoc networks,” *Communications Surveys Tutorials, IEEE*, vol. 7, no. 1, 2005.

[12] J. Cao, L. Zhang, G. Wang, and H. Cheng, “SSR: Segment-by-Segment Routing in Large-Scale Mobile Ad Hoc Networks,” in *Proc. of MASS’06*, 2006.

[13] M. Mauve, J. Widmer, and H. Hartenstein, “A survey on position-based routing in mobile ad hoc networks,” *IEEE Network*, vol. 15, no. 6, 2001.

[14] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Blni, and D. Turgut, “Routing protocols in ad hoc networks: A survey,” *Computer Networks*, vol. 55, no. 13, 2011.

[15] D. Niculescu and B. Nath, “Trajectory Based Forwarding and Its Applications,” in *Proc. of MobiCom’03*, 2003.

[16] S. Mueller, R. Tsang, and D. Ghosal, “Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges,” in *Performance Tools and Applications to Networked Systems*, ser. Lecture Notes in Computer Science, 2004, vol. 2965.

[17] A. Loch, M. Schulz, and M. Hollick, “WARP Drive - Accelerating Wireless Multi-hop Cross-layer Experimentation on SDRs,” in *Proc. of Software Radio Implementation Forum (SRIF)’14, Demo Session*, 2014.

[18] “Rice University WARP Project,” http://warp.rice.edu.