On the Performance of Multiuser MIMO Mesh Networks

Mohammad Taha Bahadori, Konstantinos Psounis
University of Southern California
{mohammab, kpsounis}@usc.edu

***Abstract*—Over the last five years both the academia and the industry have produced evidence that wireless multi-hopping suffers from low performance. Even putting real-world constraints aside and assuming deployment of optimal schedulers and rate controllers, the fact of the matter is that wireless multi-hop networks are severely constrained by interference. Advances in antenna technologies make it possible to create efficient multi-hopping architectures utilizing multiple antennas per node. Specifically, a Multiuser MIMO system could offer a significant boost in performance by utilizing spatial multiplexing, which allows nodes to send and receive multiple packets concurrently, and by reducing interference via beamforming and interference cancellation techniques. But, it comes with the need to dynamically and efficiently orchestrate the capabilities of the nodes in order to carefully balance between exploiting spatial diversity and inducing interference in the media.**

In this work we propose a distributed, scalable MAC mechanism to achieve this task which addresses all major issues associated with Multiuser MIMO wireless networks. Our mechanism consists of two sublayers. The first, called MU-MIMO sublayer, allows nodes to estimate the channel during a short, collision free period made possible via CDMA techniques, and to exchange control plane information used for scheduling. The second sublayer, called scheduling sublayer, offers an efficient, distributed scheduler which decides which nodes will send and which will receive packets each time, and selects multiple senders per receiver and multiple receivers per sender. The scheduler bases its decisions on local SINR measurements and queue differential information. We show using simulations that our system achieves more than 60% of the throughput achieved by the centralized optimal scheduler in the scenarios that we test, and outperforms prior, state of the art distributed designs.

I. Introduction

Wireless multi-hop networks suffer from low performance. Specifically, the performance of these networks is significantly degraded by the interference of nodes' transmission with each other. For example, the capacity [1], the routing performance [2], and the efficiency of transport layer operations [3], [4] are adversely affected by the limiting effects of interference.

One of the most promising ways to enhance the performance of such networks by alleviating interference is to use multiple antennas and the signal processing capabilities available for MIMO systems. MIMO systems can reduce interference in two ways: by beamforming (focusing the transmitted power on a particular direction while reducing the power in other directions to materialize a directional antenna) at the transmitter side, and by performing interference cancellation at the receiver side. What is more, in Multiuser MIMO (MU-MIMO) systems it is possible to create multiple beams to transmit multiple packets at the same time, and to receive from multiple sources at the same time.

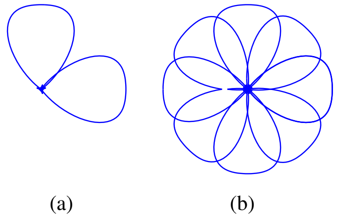


Fig. 1. Spatial diversity usage: (a) only two beams used, (b) all eight possible beams used.

Challenges: In MU-MIMO wireless networks, there is a natural trade-off between exploiting spatial diversity and inducing interference in the media. For example, consider the linear antenna configuration with eight antenna elements, as shown in Figure 1. As the transmitter node uses more spatial diversity branches, it induces more interference in the network. In the extreme, if it uses all of the available diversity branches, its interference is equally sever as that of an omni-directional transmitter. Thus, it is important to design an efficient scheduler to identify which diversity branches to use, that is, to which receivers to transmit simultaneously.

Another scheduling challenge is for nodes to decide whether they want to transmit or receive, inform their neighbors about this decision, and establish concurrent communications based on the scheduler's output. Further, when some transmitters want to send to the same receiver, their requests can collide in the receiver which results in low performance [5]. Moreover, in systems using beamforming a receiver may also fail to detect the communication between two neighbors because it can be located in the null direction of the beamforming pattern. Beamforming systems are also known to be prone to "Directional Hidden Terminal" problems, [6], [7], where the mismatch between the communication range of different beamforming patterns in different steps of the communication establishment can make neighbors unreachable.

Approach and Contributions: We use a two layer architecture shown in figure 2 which enables us to apply prior work in wireless networks scheduling without being too concerned about MU-MIMO implementation intricacies. The MU-MIMO (MMI) sublayer has the task of hiding the lower layer issues associated with MU-MIMO systems from the scheduling sublayer.

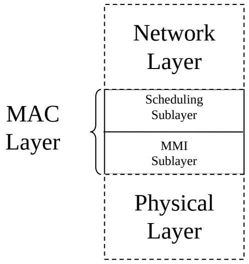


Fig. 2. The layered architecture of our design.

MMI sublayer: The high-level operation of the MMI sublayer (synchronous mode) is as follows: Nodes are synchronized and time is slotted. Assume that the scheduler has determined which nodes will transmit and which will receive in the current time slot. Receivers omni-directionally broadcast a ready-to-receive message coded with orthogonal CDMA codes to be detectable in the presence of interference. Then, transmitters send a request-to-transmit message to receivers and receivers reply back by grant messages based on local decisions by the scheduler. Both request and grant messages are sent in directional mode and are coded with orthogonal CDMA codes. After this process, the data communication starts and ends with acknowledgments as usual. For data and acks we use high rate OFDM modulation. We also design an asynchronous MMI sublayer which provides the same functionalities of the synchronous MMI sublayer via the help of an additional radio.

Scheduling sublayer: The scheduling algorithm has two tasks: (i) to determine the state of every node (transmitting or receiving) at the current time slot, and (ii) to efficiently select set of nodes to which each transmitter will concurrently transmit and, thus, the set of nodes from which each receiver will concurrently receive. This takes place via the request-grant-accept process described above. All scheduling decisions are based on local queue differential information and SINR measurements. In order to evaluate performance of our system, we carry out a number of simulations.

The paper is structured as follows. After the related work in Section II, Section III presents MU-MIMO channel models and general assumptions about our system. Section IV presents the design of the MMI sublayer and the communication establishment process, Section V presents the scheduler sublayer design and Section VII evaluates the performance of the whole design with simulations. Finally, in Section VI we briefly discuss real-world concerns, and Section IX concludes the paper.

II. Related Work

There is a large body of work on MIMO systems and scheduling in the context of wireless networks. However, to the best of our knowledge, our work is the first one to address the mentioned implementation challenges and propose a distributed scheduling algorithm for multiuser MIMO systems with multiple packet transmission and reception.

A large body of work, see, for example [7]–[11], has investigated protocols for wireless mesh networks with nodes equipped with single beam directional antennas and shown that the capacity of wireless mesh networks can be improved substantially. We are interested in combining beamforming with multiple packet transmission and reception.

Despite the large body of work in single beam directional antenna networks and MIMO systems, only few protocols have been proposed for enabling multiple packet transmission and reception in multiuser MIMO systems. Crichigno *et al* [12] and Chin *et al* [13] have studied centralized algorithms for achieving high throughput in networks with directional antennas and multiple packet reception. We are interested in practical, distributed approaches.

One of the early distributed system designs which supports multiple packet reception in directional antenna systems is ROMA, proposed by Bao and Garcia-Luna-Aceves [14]. In this work, nodes exchange omni-directionally their future mode (transmission or reception) during a short period at the beginning of every time slot. Because of the relatively low communication range of omni-directional mode, it is not possible to exchange information with distant nodes. This causes inefficiencies similar to those caused by directional hidden terminal problems. Furthermore, since all nodes use their antenna in omni-directional mode, the network scales poorly with the number of nodes and this omni-directional period becomes the bottleneck of network performance.

Li *et al* [15] have proposed a simple distributed MAC for using multiple packet transmission and reception in mesh networks, but they have not taken into account collisions of simultaneous RTS packets transmitted toward a receiver in neither their analysis nor simulation. This results in unfairness and under-utilization [5]. They also have not proposed any algorithm for determining the state (transmitter/receiver) of nodes and offer no solution for the directional hidden terminal problem. Lal *et al* [16] have proposed a protocol which uses omni-directional RTS/CTS messages for establishing communication. Thus, this design suffers from range mismatch and the directional hidden terminal problem, similarly to prior designs.

In [17]–[19] the authors study the scheduling problem in multiuser MIMO cases, but make the unrealistic assumption that the receiver is able to decode as many signals as the number of the antenna elements, since they do not take SINR constraints into consideration. In [20], Gelal *et al* studied the system under more realistic models, but they did not use beamforming nor the multiple packet transmission capabilities of MIMO systems.

III. Multiuser MIMO Networks Primer

Multiuser MIMO wireless networks are networks in which some nodes are equipped with one or more antennas. Those devices with multiple antennas can simultaneously transmit to multiple destinations or receive multiple packets from multiple sources, as shown in Figure 3.

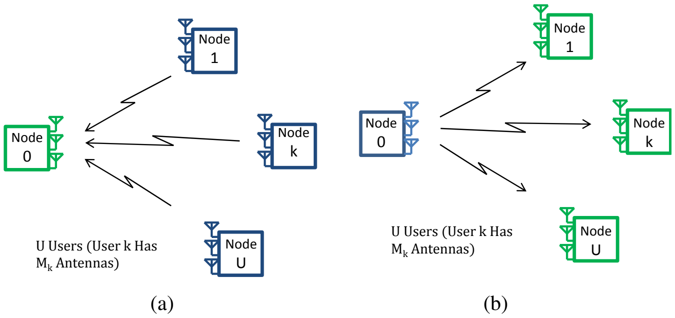

Node 1
Node 0
Node k
Node U
U Users (User k Has M_k Antennas)
(a)
Node 1
Node 0
Node k
Node U
U Users (User k Has M_k Antennas)
(b)

Fig. 3. In multiuser MIMO networks, a node can simultaneously (a) receive multiple packets from multiple sources, or (b) transmit multiple packets to multiple destinations.

When a node, say node 0 in Figure 3a, receives multiple packets from multiple transmitters k, $1 < k < U$, the received signal vector can be written as:

$$\mathbf{y}_0 = \sum_{k=1}^{U} \mathbf{H}_k^H \mathbf{x}_k + \mathbf{n}_0, \tag{1}$$

where $\mathbf{x}_k$ is the $M_k \times 1$ signal vector of node k, M_k is the number of antenna elements at node k, and $\mathbf{n}_0$ is the (additive Gaussian) noise vector at node 0. The matrix $\mathbf{H}_k \in \mathbb{C}^{M_k \times M_0}$ represents the channel matrix between nodes k and 0 and H is the Hermitian operator. In order to detect the signal of the k^{th} user, there are several receiver structures [21], most notably, Decorrelator, MMSE, and MMSE-SIC, where the later structure achieves the optimal capacity of the channel.

In the transmission mode, as shown in Figure 3b, the transmitter transmits to a subset of nodes S, $|S| = K$. The received signal vector from node 0 at the k^{th} user is:

$$\mathbf{y}_k = \mathbf{H}_{0,k}\mathbf{w}_{0,k}s_{0,k} + \mathbf{H}_{0,k} \sum_{l=1, l\neq k}^{K} \mathbf{w}_{0,l}s_{0,l}, \tag{2}$$

where $\mathbf{w}_{0,k} \in \mathbb{C}^{M_k \times 1}$ and $\mathbf{w}_{0,l} \in \mathbb{C}^{M_l \times 1}$ are the precoding (beamforming) vectors between node 0 and k and between node 0 and l respectively, and $s_{0,l}$ is the signal sent from node 0 to the l^{th} user. Note that a noise term and the interference from other nodes' transmissions in the media should be added to this signal to get the aggregate signal received by node k. If the set of destinations is specified, the precoding vector can be found by a suitably normalized ZF or MMSE inverse of the multiuser matrix, see [22], [23] for details. In this case, it can be assumed for simplification that the receivers have only one antenna and $M_k = 1$ [24]. This results in $\mathbf{H}_{0,k} = \mathbf{h}_{0,k}$ becoming a vector, so we can rewrite Equation (2) as:

$$y_k = \mathbf{h}_{0,k}\mathbf{w}_{0,k}s_{0,k} + \mathbf{h}_{0,k} \sum_{l=1, l\neq k}^{K} \mathbf{w}_{0,l}s_{0,l}. \tag{3}$$

In this paper, while we have designed our MU-MIMO sublayer (see Figure 2) for general conditions, in the design of our scheduling algorithm we have assumed that all of the nodes have the same number of antennas, $M_k = M \ \forall k$. We use decorrelator beamforming, i.e. $\mathbf{w}_{0,k} = \mathbf{h}_{0,k}^H$. As usual, full channel state information is assumed to be available at both the receiver and transmitter side, because otherwise spatial multiplexing is not possible in MU-MIMO systems (except in very special situations) [24]. The channels are assumed to be free-space channels without fading. Last, for illustration and evaluation purposes we have used linear antenna array configurations. However, our algorithms can be applied to general antenna configurations.

Since we want to study the effects of interference in MU-MIMO networks, we assume that the network operates in the high SNR regime. This assumption results in the following observations which simplify the design space: (a) the interplay between power and throughput can be neglected, in other words, in the absence of interference nodes can successfully transmit to up to M receivers and (b) the power allocation problem reduces to allocating equal power to all beams.

IV. MU-MIMO SUBLAYER

In this section we describe our MU-MIMO (MMI) sublayer design in detail. Due to the need to coordinate transmitters and receivers to establish multiple concurrent communications we *synchronize the network* by one of the available methods, see, for example, [25]–[27]. (We briefly discuss a fallback mechanism in the absence of synchronization in Section VI.) We separate the time in phases called timeslots. At each timeslot, the network will establish a number of concurrent communications between transmitters and receivers and exchange data.

At the beginning of every timeslot, the receiver nodes *broadcast* a brief message to their neighbors to announce that they like to receive in the current timeslot (see $T_{T/R}$ in Figure 4). This eliminates any possible ambiguity about the state of nodes (sending or receiving) and prevents unsuccessful channel reservation attempts. For brevity, we will refer to this message as the broadcast or the ready-to-receive message from now on.

To prevent collisions of request-to-transmit messages towards the same receiver, we choose to have a *short collision-free period* at the beginning of every timeslot using CDMA techniques. Specifically, our system uses two different receivers: A CDMA receiver at the collision-free period (used for broadcast, request-to-transmit, and grant messages) and a higher rate MIMO-OFDM receiver at the data transmission period (used for data and ack packets). Since the network is synchronized and the CDMA mode does not need to be high-rate, we use orthogonal CDMA codes like the Welsh-Hadamard codes and eliminate concerns about power control and near-far issues. The process of mapping the signature of a code to a node is similar to the algorithms used in CDMA cellular systems, see, for example, [28].

During the collision-free period, $T_{T/R}$ in Figure 4, receivers broadcast their ready-to-receive message *omni-directionally*. Further, receivers increase the range of omni-directional communication by adding redundancy via channel coding mechanisms, in order to match it with the maximum communication range used in the directional transmission mode and ensure all relevant nodes will hear the message. These design choices ensure that receivers avoid directional hidden terminal problems, and do not fail to detect a communication between two neighbors because they are located in the null direction of the beamforming pattern.

During the collision-free period, nodes estimate the channel for reception purposes as usual. Nodes use the corresponding measurements also to *compute SINR values which are later used by the scheduler* (see Section V for more details). This short collision-free period is also well suited to exchange queue differential information as discussed in Section VI. This queue differential information is used for scheduling decisions too (see Section V for details).

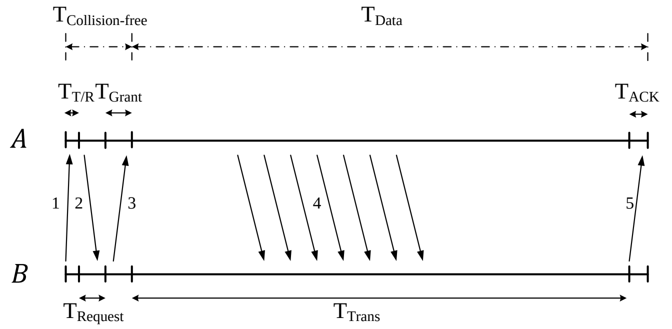


Fig. 4. A simple example of the 5-step communication establishment process in a timeslot.

A. Communication Establishment Process

This section describes in detail the sequence of events to establish communication between nodes.

Consider a node which wants to receive data from its neighbors. As shown in Figure 4, a receiver node like B first broadcasts its ready-to-receive message. Then, for a fixed duration $T_{Request}$, it monitors the channel. During this period the receiver node receives request-to-transmit packets from transmitters, estimates the channel, analyzes the signature ($\mathbf{h}_k$) of the received signals, and finds the set of nodes that are requesting to send data to it. Based on the scheduling sublayer (see Section V) the receiver node decides to grant a subset of the requests during the time period T_{Grant}. Then, it waits for data transmissions and if it receives data without any error, it replies with an acknowledgment to each of the corresponding senders.

Now, consider a node which wants to transmit data to its neighbors. As shown in Figure 4, a transmitter node like A listens to the messages transmitted by the receivers during $T_{T/R}$ to identify the nodes ready to receive in the current timeslot. Then, during the time period $T_{Request}$, the node sends its request-to-transmit messages to all receivers for which it has a packet. Assuming that the node receives one or more grants, it uses the scheduling sublayer (see Section V) to decide which grants to accept and transmits the corresponding data packets. Upon reception of an acknowledgment, the corresponding packet is removed from the queue. Clearly, it is possible for the sender to receive acknowledgements for a subset of the packets that it has transmitted during the current time slot.

Putting everything together the sequence of events is as follows (see Figure 4):

1) During $T_{T/R}$, node B announces that it is ready for reception of packets from neighbors by broadcasting a ready-to-receive message. Node A listens to the channel.
2) At the beginning of the time period $T_{Request}$, A sends its request-to-transmit messages, including a message to B. During this period, node B, like other receiver nodes, monitors the channel and analyzes the signature of the received requests.
3) Assuming that the scheduling algorithm running at B decides to grant A's request to B, B sends a grant message to A at the beginning of the time period T_{Grant}.
4) Upon reception of all grant messages by A, the scheduler algorithm finds the set of grants to be accepted. Assuming A accepts B's grant, A transmits the corresponding data to B during T_{Trans}.
5) If the reception of data by node B is successful, B sends an acknowledgment to A and the packet is removed from A's queue.

V. Scheduling Sublayer

The scheduler has to perform two tasks. First, it should determine which nodes should transmit and which should receive packets in the given timeslot. Let $\mathcal{S}_T$ be the set of transmitters and $\mathcal{S}_R$ be the set of receivers.

Second, the scheduler should determine to which receivers each transmitter will send packets, and from which transmitters each receiver will receive packets. If transmitter $i \in \mathcal{S}_T$ has a packet for receiver $j \in \mathcal{S}_R$, we indicate this with an edge from i to j and set a corresponding variable $x_{i \to j} \in \{0, 1\}$ to 1. Now, we have a bipartite graph over which we want to choose a matching.

At first glance this problem looks like a classical matching problem where the maximum weight matching yields the best scheduler. However, there are two issues. First, we need to carefully identify what to use as a weight. The size of the queue of the transmitter, which is the standard weight used, is not a good metric for reasons that will become clear shortly. Recently proposed weights for ad hoc networks based on queue differentials [29] are not directly applicable either. The second, more fundamental issue, is that contrary to the classical definition of bipartite matching where each transmitter has to be matched with at most one receiver and each receiver with a most one transmitter, here each transmitter can be matched with up to b receivers, and each receiver with up to b transmitters. (The value of b depends on the number of antenna elements and other physical layer considerations.) In the literature, this generalized matching problem is called *b-matching* and has received attention from the combinatorial optimization community, see, for example, [30]. Recently, belief-propagation ideas have been explored in an effort to offer more efficient ways to solve the problem, e.g. [31].

To the best of our knowledge there are no known algorithms for the maximum weight b-matching problem that can be used in practice in our context, since existing algorithms are either centralized, or incur a prohibitively large communication overhead and take too long to converge. For this reason, while we will compare the performance of our scheduler to that of the optimal scheduler for small scenarios where we can use brute force to compute it, we seek practical schedulers for our system. In the rest of this section, we first introduce some notation and state the optimization problem which solves the scheduling problem optimally. Then, in Section V-B we present a practical algorithm to find the sets of transmitters, $\mathcal{S}_T$, and receivers, $\mathcal{S}_R$, and in Section V-C we present a distributed, practical algorithm which matches transmitters and receivers.

A. *Optimal algorithm*

To state the optimal algorithm, we need to specify the right weights to be used in maximum weight matching. We use queue differential ideas, and borrow notation and ideas from the wGDP algorithm [29] for this task.

Each node i maintains *per-destination queues*, denoted by Q_d^i for destination d. Note that d is the final destination and it is possible that many flows originating from different sources and crossing node i have the same final destination d. The packets from all these flows will share this queue. Let q_d^i denote the size of Q_d^i, $n(i,d)$ denote the next hop towards destination d and $w_d^i = q_d^i - q_d^{n(i,d)}$ denote the queue differential at node i for destination d. In [29], the authors use $\max_d w_d^i$ as the weight of each node to solve the maximum weight matching problem. Here, we need a different weight for each potential receiver j of transmitter i, and this weight should be the maximum among all queue differentials corresponding to destinations with next hop receiver j. Thus, we use the following weights for each link $i \rightarrow j$:

$$w^{ij} = \max_{d:n(i,d)=j} w_d^i.$$

Now, the optimal scheduler is the solution to the following optimization problem:

$$\begin{array}{ll} \text{maximize} & \sum w^{ij} x_{i\rightarrow j} \\ \text{subject to:} & i \in \mathcal{S}_T, j \in \mathcal{S}_R, \\ & \text{SINR constraint} \end{array} \tag{4}$$

where we have also added an SINR constraint, since, due to physical layer considerations, a link $i \rightarrow j$ should only be considered as part of the final matching if it satisfies the SINR constraint.

B. *Transmit/Receive Algorithm*

The Transmit/Receive algorithm determines if a node will be on transmission or reception mode at each time slot. Due to real-world considerations we propose a simple, distributed scheme which is later shown to perform quite efficiently using simulations. The scheme uses queue differential information to access how "loaded" a node is. The scheme naturally selectes those nodes with the larger load as transmitters.

Specifically, let $\mathcal{D}_i$ be the set of destinations currently served via node i, and $\mathcal{N}_i$ be the set of one-hop neighbors of i. The idea is to compare the average queue differential at node i, $\overline{w^i} = \sum_{\mathcal{D}_i} w_d^i / |\mathcal{D}_i|$, with the median of its one-hop neighbors. In particular, node i will be a transmitter if $\overline{w^i} > median_{j:j\in\mathcal{N}_i}(\overline{w^j})$, otherwise it will be a receiver. Note that this choice is not particularly optimized. It is a simple choice which works quite well in practice.

C. *Request-grant-accept algorithm*

In traditional switching problems, the academia has used the so called request-grant-accept idea [32] to approximate the maximum weight matching solution in an iterative fashion. At each iteration, transmitters request to be matched with receivers. Receivers select (grant the request for) the edge with the larger weight among all requests they received, and transmitters accept the edge with the larger weight among all grants they received. The edges which are finally selected for the matching during this iteration are removed from further consideration. After a small number of iterations this procedure yields a matching that is quite close to the maximum weight matching [32].

In theory we could follow a similar procedure in our problem. In particular, both at the grant and accept phases, nodes would pick the set of edges such that their sum of weights is the maximum among all sets of edges of cardinality up to b. But, doing this at the grant phase has the following two real-world shortcomings: (i) there are many sets of edges to consider which makes the selection of the best set computationally intensive, and (ii) the SINR constraints depend on the final set of edges selected in the matching, which is not known at the time of the grant phase, thus, at best one could conservatively and inefficiently assume that all grants will be accepted for the purpose of computing the SINR values. For this reason, we follow a probabilistic approach to decide which requests to grant, as described below.

At the grant phase, each edge is selected with a probability proportional to the the queue differential of the edge and the SINR value. The rational is that the larger the queue differential the larger the urgency to serve the edge, and the larger the SINR value the larger the probability that the communication will be successful. (Note that in the switch scheduling literature there are many results about the high efficiency of policies which choose edges with a probability proportional to their weights.) As a final note, we choose to normalize both the queue differential and the SINR values with the channel vector (see Section III) which describes the correlation between the multiple beams, since when beams are spatially separated by a significant amount, they interfere a lot less than when they are close, and one needs to take this into account. We are now ready to state our scheduling mechanism:

Request: Node i will send requests to all one-hop neighbors j for which there are packets in its queue.

Grant: Node j will grant the request associated with link $i \rightarrow j$ with probability

$$p_{ij} = [\delta q_{ij}]_0^1 \left[\frac{\gamma_{ij}}{\gamma_{th}}\right]_0^1, \qquad (5)$$

where δq_{ij} and γ_{ij} are defined below, γ_{th} is the SINR threshold that guarantees 95% successful packet reception in the receiver, and $[x]_0^1 = x$ if $0 \le x \le 1$, $[x]_0^1 = 0$ if $x < 0$, and $[x]_0^1 = 1$ if $x > 1$, to ensure that $0 \le p_{ij} \le 1$.

γ_{ij} is the SINR term for the corresponding stream in the MMSE receiver [21] and it equals

$$\gamma_{ij} = P_{ij}\mathbf{h}_{i,j}^H \left(N_0\mathbf{I}_M + \sum_{l,k} P_{lk}\mathbf{h}_{l,k}\mathbf{h}_{l,k}^H\right)^{-1} \mathbf{h}_{i,j}, \qquad (6)$$

where P_{ij} is the power at link $i \rightarrow j$, $\mathbf{h}_{i,j}$ is the channel vector defined in Section III, H is the hermitian operator, N_0 is white noise, $\mathbf{I}_M$ is the identity matrix, M is the number of antenna elements, and the summation is over all received request signals (from some node l to node $k = j$) and over all interfering requests destined for other users (from some node l to some other node k).

Similarly, δq_{ij} equals:

$$\delta q_{ij} = w^{ij}\mathbf{h}_{i,j}^H \left(\sum_{l} w^{lj}\mathbf{h}_{l,j}\mathbf{h}_{l,j}^H\right)^{-1} \mathbf{h}_{i,j}, \qquad (7)$$

where now the summation is only over the requests which are destined from some node l to the receiver node j.

Accept: Node i computes the sum of queue differentials for all feasible sets of grants and selects the set with the largest value. That is, if $\mathcal{G}_i$ is the set of feasible sets of grants and $g_i \in \mathcal{G}_i$ denotes a specific such set, then the node accepts the set which maximizes

$$max_{g_i \in \mathcal{G}_i} \sum_{i \rightarrow j \in g_i} w^{ij}.$$

Note that computing whether a set is feasible or not has to do with whether the antenna can concurrently transmit to all the corresponding receivers without violating the SINR constraints. Since the number of sets is quite small, we do this computation for each set. As an aside, in practice the number of concurrent transmissions is unlikely to be equal or even too close to the number of antenna elements M.

Remarks: While the scheduling algorithm is based on a number of informed heuristics and thus one cannot make formal performance claims about it (the next section investigates its performance using simulations), it has plenty of desirable properties which make the algorithm promising and are worth highlighting. First, the term γ_{ij}/γ_{th} prioritizes the requests with higher SINR, which results in higher capacity and lower packet drop ratios. Further, if there are too many requests interfering with each other, this term drops and slows down the request rate of the transmitters, offering some kind of implicit congestion control. Second, the term δq_{ij} prioritizes the requests with larger queue differentials, which results in higher capacity and better fairness. Third, we choose to normalize the term δq_{ij} with the channel vector similarly to the standard practice in directional antenna systems for SINR terms like γ_{ij}. This keeps the granting process for two transmitters independent when the corresponding receptions at the receiver are independent, which better utilizes the directional antenna capabilities. Prior work has done similar normalizations for other quantities rather than the SINR, for example for the so called NAV table [33], [34].

As a final remark, note that the granting probabilities p_{ij} in Equation (5) could be scaled up to increase the chances that many requests are granted, and similarly, we could assign a small probability rather than 0 to grants for which δq_{ij} turns out to be negative. Computing a good scaling factor is not trivial. For example, scaling these probabilities such that the largest one is equal to 1 at each receiver i is not efficient, since some receivers should assign larger probabilities than others when, for example, their SINR levels are better. We leave as future work to investigate how much the performance may be improved by such scaling, and whether there is a practical way to compute such a scaling factor.

VI. Asynchronous operation

Sometimes node synchronization may be lost. In this case it is useful to have a fallback method to communicate despite of the lack of synchronization. The main challenge is to efficiently coordinated a number of transmitters for a particular receiver and a number of receivers for a particular transmitter. Since in asynchronous mode, nodes can be in transmission mode while another node is broadcasting its ready-to-receive message, this part of communication establishment process should be done in another frequency to make sure that all nodes can keep track of the status of their neighbors. In this section we discuss the changes that we should make in the synchronous MMI sublayer to be able to use the same Scheduling algorithm described in the previous section.

A. Radios and receivers

As Kyasanur et al pointed out in [35], there are small slices of bandwidth (1-2 MHz) available in the lower frequencies (lower than 900MHz) that can be used for exchanging control messages. Nodes use one of these small channels to exchange control messages (we call this channel "Control Channel") and the entire of the main high frequency channel for the data communication (we call this channel "Data Channel"). The frequency separation between two channels is large enough to assume that communications in them are independent.

Table I summarizes the properties of the signals in the asynchronous system; the rest of configurations are the same as synchronous system, except that two little simplifications can be done: First, since the set of active CDMA codes are already known by the CDMA receiver from reception of the broadcast message in a lower data rate, the MIMO-CDMA receiver at the $T_{R:Request}$ period in the data channel only needs to search among the known signature set which makes it far simpler. Second, because the control channel has small bandwidth, transmitters can allocate enough power to the control messages in order to make the communication range of the control messages beyond the communication range of data channel. Thus, less coding overhead is needed for the control message transmission in the control channel.

Message	Channel	CDMA coded	Redundancy codes
Broadcast message	Control	Yes	A little
Request	Data	Yes	Yes
Grant	Control	Yes	A little
Data	Data	No	No
ACK	Control	No	No

TABLE I
PROPERTIES OF DIFFERENT MESSAGES IN THE ASYNCHRONOUS SYSTEM.

B. Communication Establishment in Asynchronous mode

This section describes the sequence of events to establish communication between nodes in more detail. Consider a node which wants to receive data from its neighbors. As shown in Figure 5, it first broadcasts its Broadcast message. Then, for a fixed duration $T_{R:Request}$, it monitors the channel. In this period, called the *Request Reception* period, the node receives request packets. If no requests are received during the request reception period, the node restart the communication establishment process by determining whether it should transmit or receive. Otherwise, if there are some requests, the receiver node decides to grant a subset of the requests based on the scheduling sublayer output. After the Request Reception period, since some of the transmitter may have already started transmission toward other receivers, the receiver nodes need to transmit grant packets to the selected transmitter in the control channel. After grant transmission, the receiver waits for data transmissions and after reception of data, if data is received correctly, it sends an ACK to each sender.

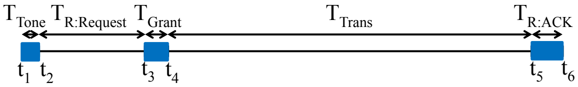


Fig. 5. Timing of operations required to receive a packet.

Now consider a node which wants to transmit data to its neighbors. As shown in Figure 6, the node listens to the control channel to find a suitable time for sending a request to the neighbors who are ready to receive. The transmitter nodes know that $T_{R:Request}$ seconds after the transmission of the Broadcast Message by a particular node, it will no longer be available for reception of a packet. Therefore, the transmitter node waits until hearing the best destination (determined by the scheduler); then during the Request Reception period of the best destination, it transmits requests to other available receivers. Just before the end of Request Reception period of the best destination, it transmits a request message to the best destination[1]. Note that nodes do not wait forever for a particular receiver; if after $T_{R:Request} + T_{Trans}$[2] seconds the particular node did not show up, the node starts communication with the current available receivers and if there is no available receiver, it restarts the communication establishment process. After request transmission, the transmitter node monitors the control channel to receive grants. At the end of grant reception period, the node starts data transmission to all the granted requests. After completion of the transmission, the transmitter node should wait for $T_{T:ACK}$ to receive acknowledgments. $T_{T:ACK}$ should be at most as long as $T_{R:Request}$ to ensure that all ACKs have enoough time to be received.

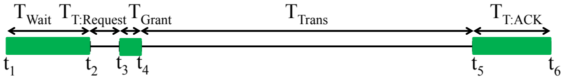


Fig. 6. Timing of operations required to transmit a packet to its next hop.

VII. PERFORMANCE EVALUATION

In this section we compare our scheduler against the optimal scheduler, as well as against other schedulers, as we vary a number of parameters like the network density, the number of antenna elements per node, and the traffic load.

A. Simulation setup

We assume each node has a linear antenna array configuration. Thus, the antenna elements are half-wavelength dipoles, spaced uniformly by a quarter of wavelength from each other. Nodes use an MMSE receiver to detect the incoming signals. We assume that control messages like the ready-to-receive broadcast message and the request-to-transmit message are never dropped due to physical layer issues, whereas data messages are dropped if they have an SINR less than a threshold.

We present results from four type of network configurations: random, grid, star, and clique configurations like the ones shown in Figures 7, 12, 14, and 16, respectively. A random configuration is constructed by randomly distributing nodes in a 200 × 200 grid. If two nodes are closer that 50 distance units apart, we setup a duplex flow between them. Each flow injects packets with probability p at every timeslot. The SINR threshold is set to $10dB$. The grid configuration has ten flows crossing the network as shown in the figure. All flows are CBR flows with equal rates. The SINR threshold for successful reception of a signal is set to $6dB$.

B. Comparison with the optimal scheduler

1) Varying the node density: Figure 8 compares the performance of our scheduling algorithm with the multiuser optimal scheduler described in Section V-A, as well as the single user optimal scheduler, which is the optimal scheduler when

[1] In Section VII-G, we will show that nodes become almost aligned in time after a while. Thus, the described process equals to transmission of request messages to all available receivers.

[2] This time is almost equal to the slot length in the synchronous system.

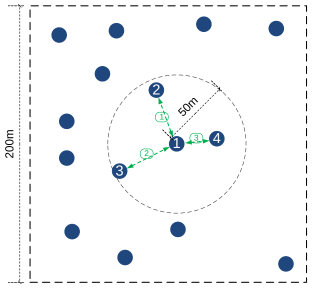
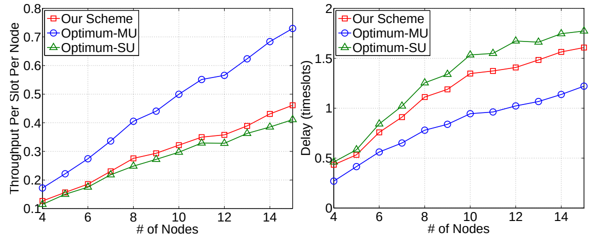


Fig. 7. A random topology.

multiple packet reception and transmission is not possible. The figure reports the average throughput and average delay where the average is taken over all nodes/flows and over 10 runs on different random topology instantiations. It is evident that the throughput achieved by our scheme is within 60% of the multiuser optimal scheduler and it is higher than that achieved by the single user optimal scheduler. The lagging in terms of delay is even smaller.



(a) (b)

Fig. 8. Performance of our system as node density increases. (a) Achieved per node throughput. (b) Average end-to-end delay.

2) Varying the number of antenna elements: Figure 9 shows the performance of our scheduling algorithm versus the optimal multiuser and single user schedulers. In order to make the comparison easier, in Figure 9a we have plotted the throughput of our algorithm as a ratio of the throughput achieved by the multiuser optimal scheduler. The figure shows that the performance of our scheduler (with respect to throughput and delay) is getting closer to that of the optimal as the number of antennas increases. This is because as we increase the number of antennas, the receivers' resolution increases and the transmitters' beams become narrower with smaller side lobes, thus the reception of multiple requests becomes more independent, which effects the grant probabilities and the quality of the matching. The reduction of interference as the number of antenna elements increase is also evident by the fact that almost all transmissions satisfy the SINR constraint, see Figure 9c. Note that scaling the grant probabilities as discussed in the previous section does improve the performance further, but we choose not to report results which depend on fine tuning of parameters.

3) Varying the traffic load: Figure 10 shows, as expected, that in low traffic rates the performance of our algorithm is very close to that of the optimal schedulers. This is because low traffic rates decrease the possibility of having multiple nodes with correlated signatures, or, put it differently, there is no contention in the scheduler: all requests are granted, and all grants are accepted, resulting in a scheduling decision very similar to the optimal one.

C. Comparison with other schedulers

We have already seen that our system outperforms the optimal single user scheduler, which is the best scheduler utilizing beamforming but no multiple packet reception or transmission.

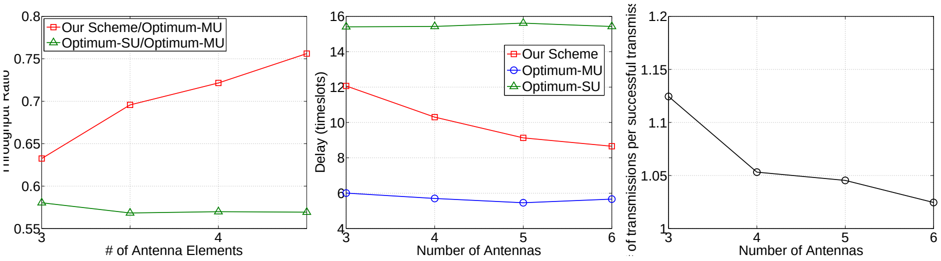
Our system uses both multiple packet reception and transmission. There are schemes which use only one of these MIMO capabilities. As an example, we compare our scheme against D-MUSIC [20] which uses only multiple packet reception. We have run simulations in identical scenarios with those presented in [20]. As expected, our scheme achieves higher throughput. For example, in networks with average node degree 4 and 8, our scheme our system achieves 2.5 and 1.77 times higher throughputs then D-MUSIC respectively.

We also compare our scheme against to somewhat naive schemes which are often used for comparison in the literature. The first is an *all-grant* scheme, where all requests are always granted. The second is a *random-grant* scheme, where each request is granted with some probability, which we set to 1/2. Figure 11a compares the throughput of our scheme against that of the all-grant and random-grant schemes. As expected, reducing the interference improves the throughput, evident from the fact that the random-grant scheme is better than the all-grant one, but, as expected, doing so in an informed manner further improves the performance since our scheme is better than the random-grant one. Figure 11b shows the average number of transmission required for a successful transmission in the network. This result shows that a key factor behind the superior scalability of our algorithm is its interference reduction ability since the interference created per transmission is almost constant as we increase the density of the network. For example, the γ_k/γ_{th} term in Equation (5) decreases the transmission rate in high interference regions and prevents further packet losses.

D. High throughput grid topology

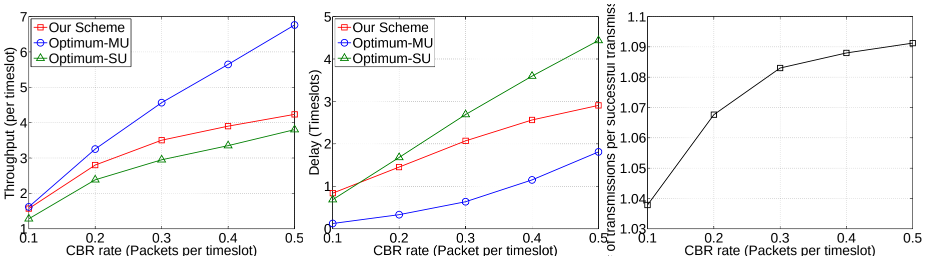
We evaluate the performance of our scheme in a grid configuration with multi-hop flows shown in Figure 12.

Figure 13a shows the throughput achieved by our scheme as we increase the number of antenna elements. Note that 8 antenna elements are enough for the optimal to schedule all flows concurrently and achieve a throughput equal to 5. (This is because the only constraint in the transmission of packets from 10 flows is that intermediate nodes cannot transmit and receive at the same time.) With 8 antenna elements, our scheme achieves a throughput that is very close to that of the optimal. Interestingly, the large jumps in the throughput, for example, when the number of antenna elements increases from 5 to 6, is due to the creation of an additional main load which allows to send one more packet concurrently, and the smaller



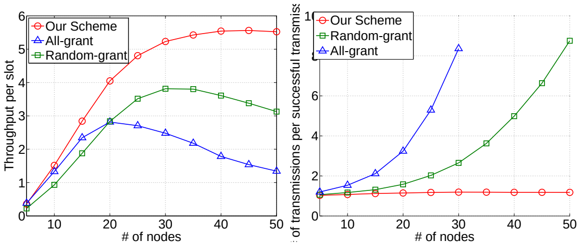
(a) (b) (c)

Fig. 9. Performance of our system as the number of antenna elements increases. (a) Achieved throughput as a ratio of the throughputs achieved by the multiuser optimal scheduler. (b) Average delay. (c) Average number of transmission attempts per successful transmission.



(a) (b) (c)

Fig. 10. Performance of our system as traffic rate changes. (a) Average aggregate throughput. (b) Average delay. (c) Average number of transmission attempts per a successful transmission.



(a) (b)

Fig. 11. (a) Achieved throughput. (b) Average number of transmission attempts per a successful transmission.

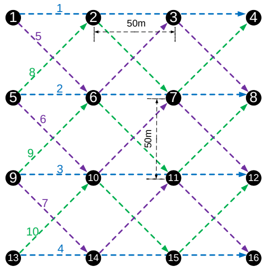
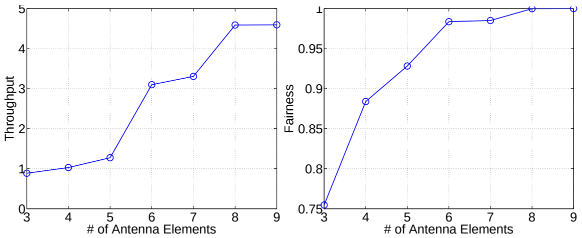


Fig. 12. The grid topology.

jumps are due to the reduction of the interference from side lobes, which increases the SINR. In Figure 13b we have used Jain's Fairness index to evaluate the fairness of our system. We can see that for a fixed traffic rate, fairness increases as the number of antenna elements increases. This is due to the fact that with more antenna elements there is less interference and more capacity, and the network can schedule almost all flows concurrently.



(a) (b)

Fig. 13. (a) Achieved throughput. (b) Fairness.

E. Star Topology

Figure 14 shows the Star topology. In this topology several client nodes transmit packets toward a common node. This topology can model several scenarios, including: the uplink of base station mode in wireless local area networks, the sink node in wireless sensor networks, and the wired network connection node in wireless mesh networks. This scenario is one of the configurations in which multiple packet reception shows a significant gain over single packet reception systems.

In absence of any interferer other than the client nodes, the optimal grant algorithm searches among all subsets with cardinality smaller than M and finds the best subset. The optimal downlink operation has recently attracted attention of researchers [24], [36]. We test our system in this scenario to assure that it achieves a high performance in this scenario, too.

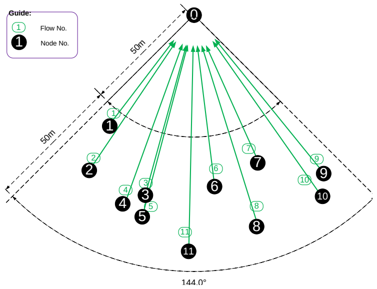
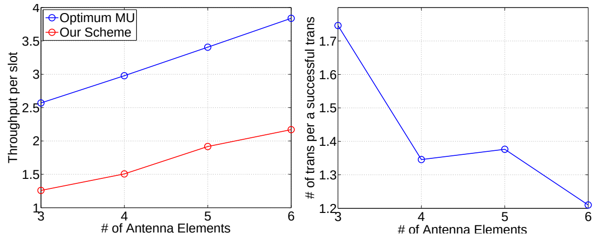


Fig. 14. The Star topology.

In our simulation, there are 21 clients; CBR flows are injecting one packet every three timeslots and the SINR reception threshold is set to $10dB$. As Figure 15 shows, our scheme achieves a reasonable performance, compared to the optimal scheduler.



(a) (b)

Fig. 15. (a) Achieved throughput. (b) Average number of transmission attempts per a successful transmission.

F. Clique Topology

The Clique topology is shown in Figure 16. While the Clique topology can model configurations like MANET, it can be also considered as a building block of bigger networks in which multiple packet transmission and reception can result in significant throughput gains. In this topology there are ten duplex flows with the equal CBR rates and the reception SINR criteria is $10dB$. Figure 17 compares the performance of our scheme with the optimum scheduler in the Clique configuration.

G. The asynchronous mode

We present results from the clique topology shown in Figure 16.

Figure 18 shows the number of packets delivered per node in each $T_{Trans} + T_{R:Request}$. It shows that the throughput is almost independent of the length of the request reception time ($T_{R:Request}$). Analysis of the trace shows that this is because the communication establishment mechanism aligns nodes in time and creates an effective multiple packet transmission and reception.

This alignment mechanism is due to the Broadcast Message transmission process in the receivers and the waiting process in the transmitters. If a receiver node is not aligned with the transmitters, it will not receive any request; thus it will extend it request reception period to receive at least one request, which makes the node aligned with the transmitter. The transmitters also wait until hearing a Broadcast Message which further helps the nodes to be aligned with the receivers. Successive operations of these two processes makes the nodes aligned, eventually.

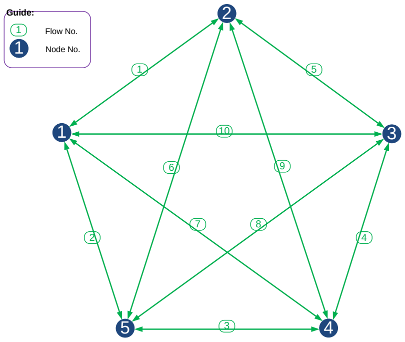


Fig. 16. The clique configuration.

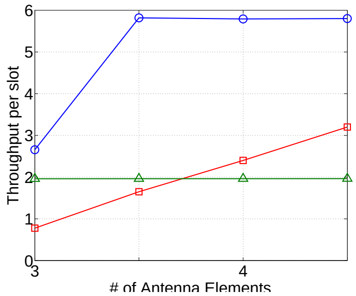


Fig. 17. Number of packets delivered in a timeslot in the Clique topology.

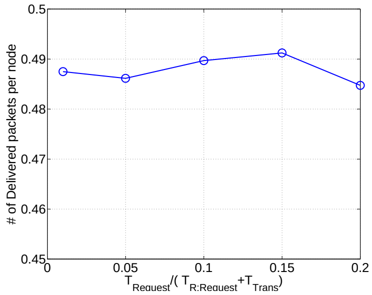


Fig. 18. Number of packets delivered per unit time per node. Unit time is defined to be $T_{R:Request} + T_{Trans}$.

Figure 19 shows the throughput per node achieved in our network configuration. Since the length of $T_{R:Request}$ does not influence the number of delivered packets, it is better to keep it as short as possible to decrease the overhead in the network.

Finally, Figure 20 shows the throughput of the asynchronous system as a ratio of the throughput achieved by the synchronous system. The figure confirms that with $T_{R:Request} = 0.01$, the asynchronous system is within 60% of the performance of the synchronous system. This is made possible by the

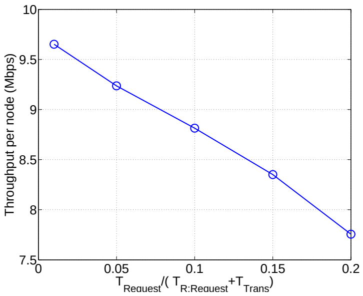


Fig. 19. Throughput per node.

alignment property of our system which increases the chances of multiple packet transmission and reception.

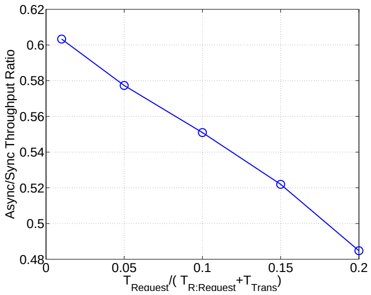


Fig. 20. Throughput of the asynchronous system as a fraction of the synchronous one.

VIII. Discussion

Exchange of queue differential information: Our scheduler uses queue differential information to make decisions. We have not yet discussed how this information is shared among nodes. First, all potential receivers should send the size of their per destination queue for all destinations d they serve, $q_d^{n(i,d)}$, to the corresponding upstream potential transmitter node i. This way node i can compute the queue differential w_d^i. (Note that in practice the number of such destinations is expected to be small.) Then, assuming all nodes i have computed w_d^i's, all nodes should send their average queue differential value, $\overline{w^i}$, to all one hop neighbors to decide whether they will transmit or receive (Section V-B), and transmitters should send the maximum queue differential along link $i \rightarrow j$, w^{ij}, to receivers to compute δq_{ij} (Equation (7)). There are two approaches to do the above. The first uses two additional control messages to do so, sent omni-directionally using CDMA techniques exactly the same way the broadcast ready-to-receive message is sent. This is what we assumed in our simulations. To avoid this overhead, another approach is to insert this information into broadcast and request messages which are transmitted anyway. The downside in this case is that the information will be outdated, since nodes will make decisions at the beginning of the current time slot based on information exchanged at the previous time slot. We leave as future work to investigate which approach yields a better accuracy-versus-overhead tradeoff.

IX. Conclusions and Future Work

In this paper we designed a distributed, scalable, high performance MAC mechanism for MU-MIMO wireless networks. The mechanism is implemented in two layers, the first deals with MIMO specific issues, and the second proposes a distributed scheduler to efficiently schedule concurrent multiple packet transmissions and receptions.

For future work we plan to investigate further improvements to our scheduler, attempt to analytically bound the performance of the system as compared to the optimal performance, and implement our system in a testbed to test our design choices under real-world conditions.

Acknowledgment

The authors would like to thank Andreas F. Molisch for his helpful comments about the system design.

References

[1] P. Gupta and P. Kumar, “The capacity of wireless networks,” *IEEE Trans. on Info. Theory*, vol. 46, no. 2, pp. 388–404, Mar 2000.

[2] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *MobiCom*. New York, NY, USA: ACM, 2003, pp. 134–146.

[3] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, “The impact of multihop wireless channel on tcp throughput and loss,” in *INFOCOM*, vol. 3, March-3 April 2003, pp. 1744–1753 vol.3.

[4] S. Rangwala, A. Jindal, K.-Y. Jang, K. Psounis, and R. Govindan, “Understanding congestion control in multi-hop wireless mesh networks,” in *MobiCom*. New York, NY, USA: ACM, 2008, pp. 291–302.

[5] G. D. Celik, G. Zussman, W. F. Khan, and E. Modiano, “MAC for Networks with Multipacket Reception Capability and Spatially Distributed Nodes,” in *IEEE INFOCOM*, Apr. 2008, pp. 1436–1444.

[6] A. P. Subramanian and S. R. Das, “Addressing deafness and hidden terminal problem in directional antenna based wireless multi-hop networks,” *Wireless Networks*, 2008.

[7] R. Choudhury, X. Yang, R. Ramanathan, and N. Vaidya, “On designing mac protocols for wireless networks using directional antennas,” *IEEE Trans. on Mobile Comp.*, vol. 5, no. 5, pp. 477–491, May 2006.

[8] A. Nasipuri, S. Ye, J. You, and R. Hiromoto, “A mac protocol for mobile ad hoc networks using directional antennas,” in *IEEE WCNC*, vol. 3, 2000, pp. 1214–1219 vol.3.

[9] G. Jakllari, I. Broustis, T. Korakis, S. Krishnamurthy, and L. Tassiulas, “Handling asymmetry in gain in directional antenna equipped ad hoc networks,” in *PIMRC*, vol. 2, Sept. 2005.

[10] T. Korakis, G. Jakllari, and L. Tassiulas, “A mac protocol for full exploitation of directional antennas in ad-hoc wireless networks,” in *ACM MobiHoc*. New York, NY, USA: ACM, 2003, pp. 98–107.

[11] J. Wang, Y. Fang, and D. Wu, “Syn-dmac: a directional mac protocol for ad hoc networks with synchronization,” in *IEEE MILCOM*, Oct. 2005, pp. 2258–2263 Vol. 4.

[12] J. Crichigno, M. Y. Wu, and W. Shu, “Throughput optimization in wireless networks with multi-packet reception and directional antennas,” in *IEEE WCNC*, ser. WCNC'09. Institute of Electrical and Electronics Engineers Inc., The, 2009, pp. 1655–1950.

[13] K.-W. Chin, S. Soh, and C. Meng, “A Novel Spatial TDMA Scheduler for Concurrent Transmit/Receive Wireless Mesh Networks,” in *IEEE AINA*. IEEE, Apr. 2010, pp. 481–488.

[14] L. Bao and J. Garcia-Luna-Aceves, “Transmission scheduling in ad hoc networks with directional antennas,” in *MobiCom*. New York, NY, USA: ACM, 2002, pp. 48–58.

[15] X. Li, Y. Zhang, and M. G. Amin, “Priority-Based Access Schemes and Throughput Performance in Wireless Networks Exploiting Multibeam Antennas,” *IEEE Trans. on Vehic. Tech.*, vol. 58, no. 7, pp. 3569–3578, Sep. 2009.

[16] D. Lal and D. Agrawal, “A multiple-beam antenna protocol at a wireless access point for exploiting spatial parallelism,” in *IEEE/Sarnoff Symp. on Advances in Wired and Wireless Comm.*, Apr 2004, pp. 23–26.

[17] K. Sundaresan, R. Sivakumar, M. A. Ingram, and T.-Y. Chang, “A fair medium access control protocol for ad-hoc networks with MIMO links,” in *IEEE INFOCOM*, vol. 4, Mar. 2004, pp. 2559–2570 vol.4.

[18] B. Mumey, J. Tang, and T. Hahn, *Joint Stream Control and Scheduling in Multihop Wireless Networks with MIMO Links*. IEEE, May 2008.

[19] S. Chu and X. Wang, “Opportunistic and cooperative spatial multiplexing in MIMO ad hoc networks,” *ACM MobiHoc*, pp. 63–72, 2008.

[20] E. Gelal, K. Pelechrinis, T.-S. Kim, I. Broustis, S. V. Krishnamurthy, and B. Rao, “Topology control for effective interference cancellation in multi-user MIMO networks,” *IEEE INFOCOM*, pp. 2357–2365, 2010.

[21] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. New York, NY, USA: Cambridge University Press, 2005.

[22] C. Peel, B. Hochwald, and A. Swindlehurst, “A Vector-Perturbation Technique for Near-Capacity Multiantenna Multiuser Communication; Part I: Channel Inversion and Regularization,” *IEEE Trans. on Comm.*, vol. 53, no. 1, pp. 195–202, Jan. 2005.

[23] M. Joham, W. Utschick, and J. Nossek, “Linear transmit processing in MIMO communications systems,” *IEEE Trans. on Sig. Proc.*, vol. 53, no. 8, pp. 2700–2712, Aug. 2005.

[24] D. Gesbert, M. Kountouris, R. Heath Jr., C.-b. Chae, and T. Salzer, “Shifting the MIMO Paradigm,” *IEEE Sig. Proc. Mag.*, vol. 24, no. 5, pp. 36–46, Sep. 2007.

[25] F. Sivrikaya and B. Yener, “Time synchronization in sensor networks: a survey,” *Network, IEEE*, vol. 18, no. 4, pp. 45–50, 2004.

[26] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, 2002.

[27] Q. Li and D. Rus, “Global clock synchronization in sensor networks,” *IEEE Trans. on Computers*, vol. 55, no. 2, pp. 214–226, 2006.

[28] A. F. Molisch, *Wireless Communications*. John Wiley & Sons, 2006.

[29] U. Akyol, M. Andrews, P. Gupta, J. Hobby, I. Saniee, and A. Stolyar, “Joint Scheduling and Congestion Control in Mobile Ad-Hoc Networks,” in *IEEE INFOCOM*. IEEE, Apr. 2008, pp. 619–627.

[30] W. R. Pulleyblank, *Matchings and extensions*. MIT Press, 1996.

[31] M. Bayati, C. Borgs, J. Chayes, and R. Zecchina, “Belief-Propagation for Weighted b-Matchings on Arbitrary Graphs and its Relation to Linear Programs with Integer Solutions,” Sep. 2007. [Online]. Available: http://arxiv.org/abs/0709.1190

[32] N. McKeown, “The iSLIP scheduling algorithm for input-queued switches,” *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 188–201, Apr. 1999.

[33] V. Shankarkumar and N. Vaidya, “Medium access control protocols using directional antennas in ad hoc networks,” in *IEEE INFOCOM*. IEEE, 2000, pp. 13–21.

[34] R. Ramanathan, “On the performance of ad hoc networks with beamforming antennas,” *ACM MobiHoc*, p. 95, 2001.

[35] P. Kyasanur, J. Padhye, and P. Bahl, “On the efficacy of separating control and data into different frequency bands,” in *2nd Int. Conf. on Broadband Networks*. IEEE, 2005, pp. 646–655.

[36] G. Caire, N. Jindal, M. Kobayashi, and N. Ravindran, “Multiuser MIMO Achievable Rates With Downlink Training and Channel State Feedback,” *IEEE Trans. on Info. Theory*, vol. 56, no. 6, pp. 2845–2866, Jun. 2010.