Adaptive and Distributed Scheduling in Heterogeneous MIMO-based Ad hoc Networks

Shan Chu Xin Wang
Department of Electrical and Computer Engineering
Stony Brook University, Stony Brook, New York 11794
{schu,xwang}@ece.sunysb.edu

Abstract

Multiple-input and multiple-output (MIMO) technique is considered as one of the most promising wireless technologies that can significantly improve transmission capacity and reliability. Many emerging mobile wireless applications require peer-to-peer transmissions over an ad hoc network, where the nodes often have different number of antennas, and the channel condition and network topology vary over time. It is important and challenging to develop efficient schemes to distributively coordinate transmission resource sharing among a heterogeneous group of nodes over an infrastructure-free mobile ad hoc network. In this work, we propose a holistic distributed scheduling algorithm that can adaptively select different transmission strategies based on the node types and channel conditions to effectively relieve the bottleneck effect caused by nodes with smaller antenna arrays, and avoid transmission failure due to violation of channel constraint. The algorithm also takes advantage of channel information to opportunistically schedule cooperative spatial multiplexed transmissions between nodes and provide special transmission support for higher priority nodes with weak channels, so that the data rate of the network can be maximized while user transmission quality requirement is supported. The performance of our algorithm is studied through extensive simulations and the results demonstrate that our algorithm is very effective in handling node heterogeneity and channel constraint, and can significantly increase the throughput while reducing the transmission delay.

1. Introduction

In recent years, we have seen a proliferation of mobile, network-enabled wireless devices. As the number, CPU power and storage space of these devices continue to grow, there is a significant increase in data transmission demand to support data intensive mobile computing and applications, such as multimedia streaming, gaming, transmission of a large amount of event data during environmental monitoring, and distributed and collaborative processing among a set of wireless devices. To meet the high data rate requirements, more and more wireless devices are equipped with multiple antennas. With multiple antennas at the transmitter and/or receiver, a MIMO (multiple-input-multiple-output) system takes advantage of *multiplexing* to simultaneously transmit multiple data streams to increase wireless data rate and *diversity* to optimally combine signals from different transmission streams to increase transmission reliability and range. The benefits of MIMO lead many to believe it is the most promising technique of emerging wireless technologies. MIMO is prominently regarded as a technology of choice for next generation wireless systems such as IEEE 802.16, IEEE 802.11n, and the third and fourth generation cellular systems. It is also being considered for supporting peer to peer mobile applications over an infrastructure free ad-hoc network.

Although MIMO techniques have been widely studied in a more centralized and infrastructure-based cellular system, there are very limited work and big challenges in extending MIMO technique into a fully distributed system over an infrastructure-free wireless ad hoc network. Different from an infrastructure-based system, it is difficult for nodes to coordinate in channel evaluations and transmissions in a distributed manner. The fast variation of channel condition and network topology, the inconsistency in node density as well as the different traffic demands and service requirements of nodes lead to more open challenges to coordinate distributed node transmissions. Moreover, in a mobile computing environment, the nodes could be very *heterogeneous* and equipped with different number of antennas. The existence of a node with a smaller-size antenna array would significantly limit the advantage of nodes with larger-size antenna arrays in the same neighborhood. Finally, the transmission environment could be very different. The number of simultaneous flows that can be transmitted not only depends on the number of antennas but also depends on the number of orthogonal channels (also called *degree of freedom*) an environment allows. Although there are some recent efforts in developing protocols for applying MIMO technique to ad hoc networks [1]–[9], to the best of our knowledge, there has not been any work that considers the heterogeneity of node antenna types and hence transmission capability and the changes of channel conditions in a distributed, peer to peer, ad hoc transmission environment. Specifically, for distributed systems, the extension from homogeneous cases to heterogeneous cases is far from trivial.

To enable more powerful mobile computing and applications in a practical distributed environment, the objective of this work is to design a holistic distributed scheduling

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

algorithm to coordinate sharing of transmission resources among heterogeneous nodes in a varying physical operation environment. The scheduling concurrently considers antenna array size, traffic demand, user service requirements, multiuser diversity and channel condition. Specifically, in order to alleviate the constraints caused by node heterogeneity and severe transmission environment, our algorithm adaptively selects different transmission strategies based on node distribution and network topology in a neighborhood, and degree-of-freedom of the transmission environment. In each transmission duration, the algorithm opportunistically schedules the nodes to transmit and determines the set of antennas to use at a selected node, by fully exploiting multiuser diversity and selection diversity to significantly improve transmission throughput and reliability. Through priority-aware scheduling, our algorithm also supports service differentiation while reducing transmission delay and ensuring fairness among nodes.

The rest of the paper is organized as follows. Section 2 discusses the background information including MIMO technologies and their application in heterogeneous networks. In Section 3, the system model is defined and the problem is formally formulated. Section 4 presents the adaptive distributed scheduling algorithm and the protocol to implement the algorithm. Simulation results are provided in Section 5 and the paper is concluded in Section 6.

2. Background and Motivation

As mentioned in Section 1, with multiple antennas at the transmitter and/or receiver, multiple data streams may be transmitted between a transmission node pair, which is called *spatial multiplexing*. At the receiver, each antenna receives a superposition of all of the transmitted data streams. In a rich scattering environment where the transmission channels for different stream are differentiable and independent, i.e., *orthogonal*, an intended receiver node can separate and decode its received data streams based on their unique spatial signatures. This multiplexing gain can provide a linear increase (in the number of antenna elements) in the asymptotic link capacity.

With multiple transmission paths, the transmission quality could be very different. Instead of sending different data through each transmitting antenna, *spatial diversity* may be exploited to improve transmission reliability. There are different types of diversity techniques. Without channel information, dependent streams can be transmitted on different antenna elements over multiple time slots and improve transmission quality through *space time coding*. When channel information is available, a subset of antennas that can transmit signals at better quality can be selected for transmissions through *selection diversity*, which is shown to outperform space-time coding [10]. To further improve transmission quality, the same data can be sent from each of the transmit antennas with each transmission weighted with an appropriate phase and gain, i.e., through *pre-coding*, such that the aggregate signal achieve the maximum strength at the receiver input.

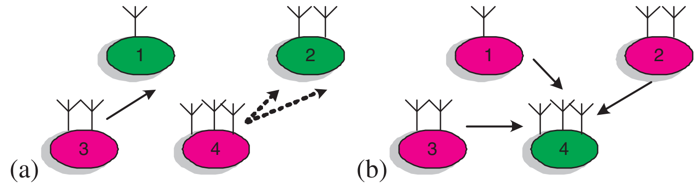


Figure 1. Illustration of heterogeneous MIMO network.

In MIMO communications, the spatial channels between two neighboring nodes n_i and n_k which have N_i^{ant} and N_k^{ant} antenna elements respectively can be represented as a $N_k^{ant} \times N_i^{ant}$ matrix:

$$\mathbf{H}_{ki} = \begin{pmatrix} h_{11} & h_{12} & \dots & h_{1N_i^{ant}} \\ h_{21} & h_{22} & \dots & h_{2N_i^{ant}} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_k^{ant}1} & h_{N_k^{ant}2} & \dots & h_{N_k^{ant}N_i^{ant}} \end{pmatrix}. \quad (1)$$

The (p, q)-th entry of $\mathbf{H}_{ki}$, h_{pq}, is the spatial channel coefficient between the p-th antenna of node n_k and q-th antenna of node n_i. The *degree-of-freedom* of a MIMO channel is an important metric to describe the dimension of space that the transmitted signals can be projected onto (so that the receiver can differentiate the signals), and the number of streams allowed to simultaneously transmitted between a pair of nodes. The degree-of-freedom is defined as the *rank* of the channel matrix $\mathbf{H}_{ki}$, or equivalently the number of non-zero eigenvalues of the matrix $\mathbf{H}_{ki}$. From (1), it is obvious that the degree-of-freedom of the channel between n_i and n_k depends on the number of antennas at nodes n_i and n_k, and the linear independency of the matrix which depends on the scattering conditions between n_i and n_k.

Instead of only allowing transmission of multiple streams between a pair of nodes as in traditional MIMO scheme, it is possible to allow multiple nodes to simultaneously transmit to a receiver that has multiple antennas, and a sender with multiple antennas can also transmit multiple streams to a set of nodes. Therefore, a group of nodes can form *virtual MIMO array* [11]. The existing work on MIMO transmissions over ad hoc networks mostly assume traditional MIMO scheme, as it is challenging to coordinate the transmissions among multiple nodes, especially in a distributed environment. Although the work in [6] took advantage of the degree-of-freedom allowed by MIMO system to support multiple groups of transmissions in a neighborhood, it did not specify how the coordination could be made among nodes to estimate the channel and form transmission pairs. The work in [9] proposed schemes to enable cooperative MIMO transmission and demonstrated the significant performance gain, however, it did not specifically consider issues in a heterogenous environment.

The literature work normally assumes the nodes are homogeneous, and channel conditions are ideal and could support full rank transmissions. However, in a generic distributed wireless system, nodes are equipped with heterogeneous number of antennas, and the physical channel conditions could vary over time. As analyzed above, both factors would impact the degree-of-freedom of the channel, hence the number of simultaneous transmissions. In Fig. 1 (a), if node 1 is a receiver, due to the constraint of its antenna number, only one stream can be accommodated from any transmitter. Additional transmissions from any node at the same time will corrupt the receiving at node 1. Moreover, even though node 4 would be able to transmit up to 2 streams to node 2 (as shown in dashed lines), if simply scheduling the transmissions based on the minimum number of streams allowed in a neighborhood [9], only one stream is allowed to transmit between node 4 and 2 at a transmission time. On the other hand, when the channel between node 4 and 2 can only support one transmission but two streams are transmitted, the streams cannot be decoded at the receiver. The examples indicate that simply applying the algorithms developed under homogeneity assumption to heterogeneous scenarios would either lead to significant throughput reduction (e.g., due to the bottleneck effect) or transmission failure. Additional issues will arise if some of the channels are weak, and cannot support good quality transmission.

These practical problems indicate that effective scheduling algorithms need to be designed to alleviate the bottleneck effect and to provide good system performance under any transmission environment. A few strategies may help. First, when the receiver has multiple antennas, the constraints to transmissions due to lower degree-of-freedom may be mitigated with formulation of cooperative virtual MIMO array. In Figure 1(b), node 1, 2 and 3 can transmit concurrently to node 4 and exploit multiplexing gain to improve throughput. Second, when the receiver has very few receiving antennas (Node 1 in Fig. 1 (a)), a transmitter could employ pre-coding to improve the data rate. Additional capacity gain can be achieved with exploration of multi-user diversity and antenna selection diversity, in which case, the transmitter nodes and the antenna to use from a node are opportunistically selected based on the channel conditions between different nodes and antennas.

3. System Model

We consider distributed channel resource allocation among an ad hoc network of nodes which have different number of antenna elements and experience different channel conditions. For a group of nodes that share transmission resource, although in traditional MIMO scheme one node pair is scheduled to transmit at a time, the chance of having multiple strong spatial paths between a node pair is small, which limits the transmission rate. In this work, we consider using virtual MIMO arrays to support many-to-many transmissions between nodes in a distributed system environment. Specifically, we design an *adaptive* and distributed scheduling algorithm that flexibly schedules node transmissions with use of different MIMO techniques, including spatial multiplexing, selection diversity, and pre-coding, based on the node constraints and channel conditions. For the convenience of presentation, in this section, we first introduce some notations used in this paper, and then formulate the problem formally and prove its NP-hardness. A centralized scheduling algorithm with provable approximation ratio is then proposed.

3.1. Stream and Stream Characteristics

A stream is defined to be an *independent* flow of signals transmitted from one or more antennas to a target node and identified by a triplet $(I^{tx}, I^{rc}, \{I^{ant}\})$, where I^{tx}/I^{rc} is the index of the transmitter/receiver node, and $\{I^{ant}\}$ is the set of antennas that involve in the transmissions of the stream. In a simple spatial multiplexing case, only one antenna is used to transmit a flow of symbols. When transmitter pre-coding is employed, a stream of data symbols could be pre-coded over space and transmitted simultaneously through several antennas. If selection diversity is assumed, only the antenna with the strongest channel condition among the candidate ones is selected to transmit the data stream.

In order for receiver nodes to decode data streams and suppress interference streams concurrently, the number of streams transmitted or received at a node is subject to certain constraint. Due to the broadcast nature of wireless channels, streams are categorized as *data streams* and *interference streams*. A data stream from node n_i to node n_k is received by n_i's neighboring node n_j as an interference stream. Denote the degree-of-freedom of the channel between n_i and n_k as $DoF(i, k)$, it is clear that n_k can differentiate streams from n_i only if the number of streams is no more than $DoF(i, k)$, as the degree-of-freedom of a channel is the largest number of streams between a node pair that can be decoded. Denote the set of all active receiving nodes (i.e., the target receivers of some transmitter nodes) around node n_i's transmission range as R_i^{active}, as the *transmitting constraint*, the number of transmitting streams from n_i should be no larger than $\mathcal{N}_i^{tx} = \min_{k \in R_i^{active}} DoF(i, k)$. Similarly, to avoid erroneous decoding at a receiver node n_k, the number of simultaneous received streams N_k^{rc} (including both data streams and interference streams) should be limited. With use of virtual MIMO array, the size of antenna array N_k^{ant} generally provides the metric of spatial resolution at a receiver n_k, and hence the total received streams should not exceed the *receiving constraint* $\mathcal{N}_k^{rc} = N_k^{ant}$.

The characteristics of a stream are captured by two parameters, *stream priority* $\mathscr{P}(s)$ and *stream capacity* $\mathscr{C}(s)$. The stream priority depends on the service type and queuing delay of the data packet to be sent with the stream. The value of $\mathscr{P}(s)$ is initially set to the service priority of the associated packet, and increases as the queuing time of the packet increases. The stream capacity describes the maximum achievable rate of a stream transmission, which depends on the transmission power of the stream and the channel condition between the transmitter antenna(s) and the receiver node. $\mathscr{C}(s)$ can be estimated at transmitters based on estimated channel conditions during scheduling.

3.2. Types of Nodes and Slots

Our algorithm is TDMA-based, in which the time domain is divided into transmission durations (TD). A TD consists of several time slots and covers one round of control signal exchange and fixed-size data frame transmission. The data transmission rate within a frame can vary based on the channel condition. For a channel with higher quality, more efficient coding can be used to encode the symbols at a higher rate. A link between a transmitter-receiver pair is half-duplex, so that a node can either transmit or receive but not at the same time.

Denote the set of nodes in the transmission range of node n_i as $\mathscr{V}_i^A$, the receiving constraint of node n_k as $\mathscr{N}_k^{rc}$, then the average receiving capability of nodes in $\mathscr{V}_i^A$ is represented as $\bar{\mathscr{N}}_i^{rc} = \sum_{k \in \mathscr{V}_i^A} \mathscr{N}_k^{rc}$. Compared with $\bar{\mathscr{N}}_i^{rc}$, if $\mathscr{N}_k^{rc} \geq \bar{\mathscr{N}}_i^{rc}$, node n_i considers n_k as a *rich* node, otherwise, a *poor* node. A poor receiver has limited decoding capability and constrains the maximum number of streams (including both data streams and interference streams) allowable in its neighborhood.

To reduce the constraint, transmission slots are divided into *P-slots* and *R-slots* for transmissions towards poor nodes and rich nodes respectively. In a P-slot, the number of concurrent transmission streams is limited by the receiving constraint of the targeted poor node, and transmitter pre-coding may be utilized to optimize link rate. In an R-slot, as only rich nodes serve as the receivers, multiuser spatial multiplexed transmissions are opportunistically scheduled for a higher throughput.

3.3. Problem Formulation

In a TDMA-based MIMO ad hoc network, packets are generated constantly. It is thus practical to schedule the transmission of packets in each transmission duration (TD) with the purpose of optimizing temporary network performance. Suppose there is a set of nodes $N = \{n_1, n_2, \ldots, n_{N_n}\}$ in the network. Based on their queuing packets, node n_i has a set of candidate streams S_i, where the destination node of the q-th stream $s_{iq} \in S_i$ is denoted as $d(s_{iq})$. Let the parameter set $\{y_{iq}\}$ ($y_i \in \{0,1\}, i = 1, \ldots, N_n, q = 1, \ldots, |S_i|$) denote whether the q-th candidate stream of node i is transmitted in the current TD. If a stream s_{iq} is transmitted, $y_{iq} = 1$; otherwise, $y_{iq} = 0$. Similarly, $\{x_i\}$ ($x_i \in \{0,1\}, i = 1, \ldots, N_n$) is used to denote the transmitter node assignment in the current TD. If node n_i is selected as a transmitter node, $x_i = 1$, otherwise $x_i = 0$. The assignment of a stream to a specific antenna of a transmitter is represented by the parameter set $\{a_{iqk}\}$ ($a_{iqk} \in \{0,1\}, i = 1, \ldots, N_n, q = 1, \ldots, |S_i|$ and $k = 1, \ldots, N_i^{ant}$), where $a_{iqk} = 1$ if stream s_{iq} is assigned to transmit from the k-th antenna of node n_i. Note that if $a_{iqk} = 1$, the transmission rate of stream s_{iq}, denoted as $\mathscr{C}(i, d(s_{iq}), k)$, depends on the triplet $(i, d(s_{iq}), k)$ and the specific transmission strategy. The priority of stream s_{iq} depends on its associated packet and is denoted as $\mathscr{P}(s_{iq})$.

The scheduling process selects a set of streams to transmit among all the candidate ones in the current TD, with the aim of jointly optimizing both data rate and priority. The problem is formulated as follows:

$$\max \sum_{i=1}^{N_n} \sum_{q=1}^{|S_i|} \sum_{k=1}^{N_i^{ant}} x_i (1 - x_{d(s_{iq})}) y_{iq} a_{iqk} \mathscr{C}(i, d(s_{iq}), k) \mathscr{P}(s_{iq}); \tag{2}$$

$$\sum_{k=1}^{N_i^{ant}} x_i y_{iq} a_{iqk} \leq N_i^{ant}, i = 1, 2, \ldots, N_n, q = 1, \ldots, |S_i|; \tag{3}$$

$$\sum_{q=1}^{|S_i|} \sum_{k=1}^{N_i^{ant}} x_i y_{iq} a_{iqk} \leq \mathscr{N}_i^{tx}, i = 1, 2, \ldots, N_n; \tag{4}$$

$$\sum_{m \in \mathscr{V}_i} (1 - x_i) \sum_{q=1}^{|S_m|} \sum_{k=1}^{N_m^{ant}} x_m y_{mq} a_{mqk} \leq \mathscr{N}_i^{rc}, i = 1, 2, \ldots, N_n; \tag{5}$$

$$x_i, y_{iq}, a_{iqk} \in \{0, 1\}.$$

Constraint (3) ensures that the data flow of a stream can be transmitted over N_i^{ant} antennas at most; equation (4) represents the constraint at a transmitter n_i, with which the total number of transmitted streams should be no more than the transmitting constraint value $\mathscr{N}_i^{tx}$; and equation (5) provides the constraint at receiver n_i where the total number of receiving streams is restricted to be no more than its receiving constraint value $\mathscr{N}_i^{rc}$ in order to decode the receiving packet. So far, we formulate the problem of heterogeneous stream scheduling as an integer programming problem with the objective function in (2) subject to constraints (3)(4)(5).

Theorem: The heterogeneous stream scheduling (HSS) problem described above is NP-hard.

Sketch of Proof: First we introduce a simplified version of HSS problem represented by a graph $G = (V, E)$. A vertex $v_i \in V$ represents a node n_i, and an edge $e = (v_i, v_k)$ denotes that n_i and n_k are neighbors in the network. Assume each node has a candidate stream s for each of its neighbors, and the gain of scheduling $\mathscr{C}(s)\mathscr{P}(s)$ is 1 for all s. The transmitting and receiving constraints for all n_i are $\mathscr{N}_i^{tx} = \mathscr{N}_i^{rc} = 1$. The optimum scheduling solution of the simplified HSS problem is then a maximum set of vertices that can transmit simultaneously while $\mathscr{N}_i^{tx}$ and $\mathscr{N}_i^{rc}$ are satisfied for transmitter and receiver nodes respectively. The simplified HSS problem can be proved to be NP-hard by reducing the NP-complete maximum independent set (MIS) problem to it. For any instance of MIS represented by a graph G', replace each edge with a dummy vertex, and connect it to the original two end vertices. The dummy vertices that represent edges connected to the same vertex are also connected. In this way, a new graph G is formed. It is then straightforward to see that the optimum scheduling solution of the simplified HSS problem in G gives an equivalent solution of MIS problem in G'.$\square$

Due to the NP-hardness of the problem, an efficient heuristic algorithm is required to solve the scheduling problem. From the formulation (2)-(5), it is clear that the scheduling problem has to determine the values of the three parameter set: $\{x_i\}$, $\{y_{iq}\}$ and $\{a_{iqk}\}$ to assign a packet to an appropriate transmitter antenna in order to maximize the total weighted rate of the network. In a practical half-duplex network, it is reasonable to divide the problem into two parts: transmitter selection and stream allocation, where the first phase determines the values of $\{x_i\}$ and $\{y_{iq}\}$ and the second phase determines the value of $\{a_{iqk}\}$ to assign a packet to a specific transmission stream. In the next section, the two subproblems are solved separately.

4. Distributed Algorithm and Protocol

In order to address the challenges discussed in Section 2, our algorithm groups transmissions into two types, transmissions to poor nodes using *P-slots* and to rich nodes using *R-slots*. The current slot type is *distributively* determined by each node and the nodes in a neighborhood reach a consensus through signaling exchange. In both types of slots, spatial multiplexing, selection diversity and transmitter precoding are adaptively utilized to deal with varying traffic demands and channel conditions to improve the overall network performance.

The distributed scheduling algorithm consists of two phases, namely *transmitter node selection / slot request* and *stream allocation*. In the first phase, a set of nodes are first selected to be transmitter nodes, and each node differentiates its packets for poor nodes and rich nodes to determine its current preference of transmission slot type. In the second phase, stream allocation is performed to allocate the data packets of the transmitter nodes to a selected set of antennas with an appropriate MIMO strategy.

In the rest of this section, we first present our scheduling algorithm in sequence of the two phases mentioned above. The complete protocol is then introduced, where we explain the detailed procedures taken to implement the algorithm and calculate the required parameters in a distributed environment.

4.1. Transmitter Node Selection and Slot Request

In this phase, nodes are distributively selected as transmitter nodes and their preference of slot type is decided. Instead of randomly selecting the transmitter nodes, the transmitter selection phase supports service differentiation and reduces transmission delay by giving higher transmission priority to the streams that are with packets in higher service class and/or have larger queuing delay. Additionally, the type of transmission slots is differentiated to support transmissions to heterogeneous nodes. We first give the main idea and define parameters used for selection, then we discuss the details of the selection process.

4.1.1. Basic Plot. In MIMO transmissions, in order to not exceed the decoding capacity of nodes, the number of streams that can be simultaneously transmitted in a neighborhood is constrained. Therefore, the number of transmitter nodes selected in our algorithm also has a limit, which will avoid unnecessary channel measurement. In addition, the decoding capacity of receivers, represented by their receiving constraints in Section 3.1, are different in a heterogeneous MIMO network. In our algorithm, each node distributively determines if it can serve as a transmitter node in a transmission duration, and selects the type of slot used for transmission based on the decoding capacity of its neighboring receivers.

Based on the receiving constraint, an active node n_i which has data to send groups its neighboring nodes into poor node set $\mathscr{V}_i^p$ and rich node set $\mathscr{V}_i^r$ based on a neighbor n_k's receiving constraint $\mathcal{N}_k^{rc}$, which is broadcast with the Hello messages sent periodically at the network layer. We introduce a threshold value $\mathcal{T}_i^{TX}$, which is calculated separately for each of the two sets. Denote the set of neighboring nodes *in concern* as $\mathscr{V}_i$, where $\mathscr{V}_i$ can correspond to $\mathscr{V}_i^p$ or $\mathscr{V}_i^r$ depending on which set is concerned at the calculation time. The parameter $\mathcal{T}_i^{TX}$ of n_i is estimated based on the number of active nodes around a neighboring node $n_j \in \mathscr{V}_i$, N_j^{active}, and the receiving constraint of node n_j, $\mathcal{N}_j^{rc}$ as $\mathcal{T}_i^{TX} = \min_{j\in\mathscr{V}_i}\left(\mathcal{N}_j^{rc}/N_j^{active}\right)$. $\mathcal{T}_i^{TX}$ represents the probability of a node n_i being a transmitter while ensuring the neighbors in $\mathscr{V}_i$ to perform correct decoding. A node n_i can be selected as a transmitter if the value of an appropriately calculated random variable is below $\mathcal{T}_i^{TX}$.

Recall that we use stream priority to represent how urgent a stream transmission is. It is therefore natural to use the average stream priority to reflect the level of priority for a node to be a transmitter. Denote all candidate streams of n_i as a set S_i and the priority of a stream s_{iq} as $\mathscr{P}(s_{iq})$, the priority of a node n_i can be represented by the average priority of its candidate streams as $\mathscr{P}_i = \sum_{s_{iq}\in S_i}\mathscr{P}(s_{iq})/|S_i|$. A node n_i can calculate the average priority $\bar{\mathscr{P}}_i$ of all the N_i^{active} active nodes in its neighborhood as $\bar{\mathscr{P}}_i = \sum_{j=1}^{N_i^{active}}\mathscr{P}_j/N_i^{active}$. The priority of a node can be attached with periodic Hello messages sent at the network layer, and updated with the data packets sent. The priority of nodes not having packets sent in a TD can be predicted as time moves forward.

To avoid extra signaling and control overhead, an active node n_i *self-decides* if it should be selected as a transmitter node by calculating an index number $\mathcal{X}_i^{TX} = (\bar{\mathscr{P}}_i - \hat{\mathscr{P}}_i)/\bar{\mathscr{P}}_i + \gamma_i$. Here the parameter γ_i is a random number uniformly distributed in the range [0,1] and generated by a node n_i at each transmission duration (TD) to provide some fairness among nodes. $\hat{\mathscr{P}}_i$ is the average priority of candidate streams at node n_i that are targeted for nodes in $\mathscr{V}_i$. The factor $(\bar{\mathscr{P}}_i - \hat{\mathscr{P}}_i)/\bar{\mathscr{P}}_i$ is used to give the higher priority node a larger probability for transmission. In a TD, if $\mathcal{X}_i^{TX} < \mathcal{T}_i^{TX}$, node n_i is selected as a transmitter node for receiver nodes in $\mathscr{V}_i$; otherwise, it has no right of transmission. Our transmitter selection algorithm prefers a node with a higher service level and/or a larger load and hence longer delay, and thus supports QoS and load balancing while ensuring certain fairness. Our selection is relatively conservative as it considers the decoding capability of all the neighboring nodes instead of only that of the actually selected receiver nodes.

4.1.2. Selection Process. At the beginning of a transmission duration, an active node n_i first determines whether it needs to initiate a transmission using P-slot based on the priority of its streams targeted for poor nodes in $\mathscr{V}_i^p$. For a poor node $n_k \in \mathscr{V}_i^p$, the average priority of streams is calculated as $\mathscr{P}_i(k) = \sum_{m \in S_{i,k}} \mathscr{P}(m)/|S_{i,k}|$, where $S_{i,k}$ is the set of streams from node n_i to the poor node n_k. Denote the poor node with the highest $\mathscr{P}_i(k)$ as $n_{i^*} = \arg\max_{n_k \in \mathscr{V}_i^p} \mathscr{P}_i(k)$. Due to the limited decoding capacity of poor nodes, a transmitter is not allowed to initiate many streams, so we only consider one candidate receiver, n_{i^*}. Let $\mathscr{V}_i = \mathscr{V}_i^p$ and substitute $\hat{\mathscr{P}}_i$ by $\mathscr{P}_i(i^*)$ for calculating the index $\mathcal{X}_i^{TX}$, which is compared with $\mathcal{T}_i^{TX}$ calculated based on nodes in $\mathscr{V}_i^p$. If $\mathcal{X}_i^{TX} < \mathcal{T}_i^{TX}$, node n_i can be a transmitter node and initiate a P-slot transmission to node n_{i^*}. Otherwise, node n_i checks if it can be a transmitter using R-slot. Similar to the previous step, $\mathcal{T}_i^{TX}$ is calculated concerning nodes in $\mathscr{V}_i^r$ and $\mathcal{X}_i^{TX}$ is obtained by letting $\hat{\mathscr{P}}_i$ equal to $\mathscr{P}_i^r = (\sum_{k \in \mathscr{V}_i^r} \sum_{m \in S_{i,k}} \mathscr{P}(m))/\sum_{k \in \mathscr{V}_i^r} |S_{i,k}|$, where $S_{i,k}$ is the set of streams which are from node n_i to a rich node n_k. Node n_i is selected as a transmitter node for receiver nodes in $\mathscr{V}_i^r$ if the updated parameters satisfy $\mathcal{X}_i^{TX} < \mathcal{T}_i^{TX}$. Note that as rich nodes have relatively higher decoding capacity and can generally accommodate more streams, the average priority of streams for nodes in $\mathscr{V}_i^r$ is used here so that n_i can initiate R-slot request towards multiple rich nodes.

If a node determines to be a transmitter node, it broadcasts an RTS message indicating the slot type as discussed in 4.3. After the transmitters and the slot type are confirmed by the receiver nodes through CTS transmission, the transmitter nodes proceed to the second phase of the scheduling described next.

4.2. Stream Allocation

Stream allocation is performed *distributively* at each of the selected transmitter nodes. The selection gives preference to streams with higher priority. For streams of the same priority, to achieve a higher data rate, the allocation process is solely based on the stream capacity by *opportunistically* assigning a channel with good condition to a selected stream. For a high-priority stream that does not have high-quality channel, the selection process reserves more of the total transmitting power for the stream to ensure a higher transmission reliability.

For a selected transmitter, there is a limit on the number of streams it is allowed to transmit, in order to meet the receiving constraints at all neighboring receivers. For a selected transmitter n_i, let N_i^0 be the number of pre-selected streams to be transmitted and N_i^{allo} be the number of streams node n_i is allowed to transmit, which is calculated based on feedbacks from neighboring receivers as described in Section 4.3. Denote the set of antennas that node n_i has as $\{A_i\}$, and the N_i^0 candidate streams have L_i distinct priority levels. The receiver nodes that the N_i^0 pre-selected streams are targeted for are partitioned into subsets $\{D_i^1\}, \{D_i^2\}, \ldots, \{D_i^{L_i}\}$ according to the descending priorities of the streams, where the set $\{D_i^j\}$ contains the target receiver nodes of the streams with j-th highest priority level. Recall that a stream s is identified by its transmitter node, transmitter antenna and receiver node. For a stream of n_i which has transmitting antenna $A_i(p)$ and receiver $D_i^j(q)$, the stream capacity is $\mathscr{C}(i, A_i(p), D_i^j(q))$. For transmitter node n_i, there is a set W_i^0 consisting of all the capacity parameters of the candidate streams $W_i^0 = \bigcup_{j=1}^{L_i} \{\mathscr{C}(i, A_i(p), D_i^j(q)) | A_i(p) \in \{A_i\}, D_i(q) \in \{D_i^j\}, p = 1, \ldots, |\{A_i\}|, q = 1, \ldots, |\{D_i^j\}|\}$.

Two stream allocation strategies are used, depending on which transmission pattern is selected in a heterogeneous MIMO network. In a TD, *one-to-one* transmission between a pair of nodes is initiated if a sender has only one target receiver and gets acknowledgement from the receiver, while the receiver either does not receive transmission requests from any other transmitters or allows only one stream to transmit (which is known from signal exchanges in Section 4.3). In this case, pre-coding is used at the transmitter to weight the signal at each antenna element through water-filling over the estimated orthogonal eigen-channels to increase the signal strength [11]. For more general *many-to-many* transmissions, we introduce the subroutine *OPPORTUNISTIC_ALLOCATION* in algorithm 2 to allocate k antennas to transmit the streams of the j-th highest priority level that are targeted for the receivers set $\{D_i^j\}$. The parameter N_i^{res} is the residual number of antennas available for allocation, the set $\{A_i\}^{res}$ contains the candidate antennas of node n_i for stream allocation, W_i^j contains the capacity parameters of the streams formulated between the antennas in $\{A_i\}^{res}$ and the receivers in $\{D_i^j\}$ and l represents the number of streams currently allocated. The allocation is based on spatial multiplexing and selection diversity, and in sequence of descending stream quality. As the allocation scheme favors stream priority than stream quality, in some cases, although the channel condition is severe, a transmission with high priority is still permitted. To reduce erroneous decoding thus packet loss under the severe channel condition, when a selected stream does not have good enough quality as indicated by a weak channel indicator include in the CTS (Section 4.3), the total number of antennas available for allocation is decreased by one to reserve extra transmitting power for the weak stream to improve its quality.

Algorithm 1 Distributed Scheduling

Initialize: $j = 1$, $\{A_i\}^{res} = \{A_i\}$, $N_i^{res} = N_i^{allo}$
if ONE-TO-ONE transmission **then**
  Do precoding
else
  while $N_i^{res} > 0$ **do**
    if $|\{D_i^j\}| \le N_i^{res}$, $N_i^{res} = N_i^{res} - |\{D_i^j\}|$ **then**
      OPPORTUNISTIC_ALLOCATION$(\{A_i\}^{res}, \{D_i^j\}, |\{D_i^j\}|, N_i^{res})$
    else
      OPPORTUNISTIC_ALLOCATION$(\{A_i\}^{res}, \{D_i^j\}, N_i^{res}, 0), N_i^{res} = 0$
    end if
    $j \Leftarrow j + 1$
  end while
end if
OPPORTUNISTIC_ALLOCATION$(\{A_i\}^{res}, \{D_i^j\}, k, N_i^{res})$
Initialize: $l = 0$, $W_i^j = \{\mathscr{C}(i, A_i^{res}(p), D_i^j(q)) | A_i^{res}(p) \in \{A_i\}^{res}, D_i^j(q) \in \{D_i^j\}, p = 1, \ldots, |\{A_i\}^{res}|, q = 1, \ldots, |\{D_i^j\}|\}$
while $l < k$ **do**
  $W_{max} \Leftarrow \max W_i^j$, $\{A_{max}, D_{max}\} \Leftarrow \arg\max W_i^j$
  Allocate the stream for the receiver D_{max} to the antenna A_{max};
  $W_i^j \Leftarrow W_i^j \setminus \{W(A_{max}, D_i^j(q)) | D_i^j(q) \in \{D_i^j\}, q = 1, \ldots, |\{D_i^j\}|\}$; if there is no other stream target for the receiver node D_{max}, also remove $\{W(A_i^{res}(p), D_{max}) | A_i^{res}(p) \in \{A_i\}^{res}, p = 1, \ldots, |\{A_i\}^{res}|\}$;
  if D_{max} has sent indicator of weak channel **then**
    if $N_i^{res} > 0$ **then**
      $k \Leftarrow k + 1$, $l \Leftarrow l + 1$, $N_i^{res} \Leftarrow N_i^{res} - 1$;
    else $\{N_i^{res} = 0\}$
      $k \Leftarrow k - 1$
    end if
  end if
  $\{A_i\}^{res} \Leftarrow \{A_i\}^{res} / A_{max}$
  $l \Leftarrow l + 1$
end while

In the main function of stream allocation, j is the index of the priority level, $\{A_i\}^{res}$ is the set of remaining available antennas of node n_i and N_i^{res} is initially set to be the total number of streams for allocation N_i^{allo}. The transmitter detects the transmission pattern based on receiver feedbacks, and determines to use pre-coding or opportunistic allocation accordingly. For many-to-many transmission, the main loop starts from the set of candidate streams which have the highest priority ($j = 1$), and calls the subroutine *OPPORTUNISTIC_ALLOCATION* for each priority level, until all the allowed streams have been allocated or the antennas of node n_i have all been assigned or reserved for streams.

4.3. Implementation of Distributed Scheduling

The implementation of the distributed scheduling algorithm is TDMA-based, where the time is divided into a serials of transmission duration consisting of four phases with different lengths. The duration of each phase is fixed and enough for the corresponding message transmission. Following the convention of IEEE 802.11 DCF, signaling messages are named RTS, CTS, DATA and ACK, which are transmitted during phase I, II, III and IV respectively. Note that slot synchronization is currently achievable in the IEEE802.11 family of protocols. By taking advantage of the selection diversity and multi-user diversity, our scheme could effectively increase the SINR of a received signal, which would help improve the accuracy of synchronization as well as mitigate the impact of a-synchronicity in a distributed scenario. The procedure of signal exchange and information acquisition for heterogeneous MIMO scheduling is as follows.

Phase I: Transmission Request and Slot Conservation. At the beginning of phase I, a node n_i which selects itself as a transmitter node as in Section 4.1 broadcasts an RTS. Before sending out the RTS, node n_i selects a set of highest-priority data packets from its queue to form $N_i^0 \le N_i^{max}$ candidate streams, where N_i^{max} is the maximum number of streams that can be transmitted in a transmission duration depending on the selected slot type, the number of antennas of n_i, and the amount of data queued. The IDs of the target receiver nodes of the selected packets, the value N_i^0, as well as the ID of node n_i are then included in the RTS. If i wants to request a P-slot towards node n_k, an RTS should further carry an indicator of P-slot and the calculated average priority $\mathscr{P}_i(k)$.

The preamble of a packet is used as the training sequence for channel estimation purpose. For both types of slot, an RTS is rotationally broadcasted through each antenna of the transmitter node with a short notice signal separating two antennas' transmissions, so that the spatial channels between each antenna of the transmitter nodes and the receiver nodes can be differentiated and estimated. An RTS is masked by another random code, called ID code, which are almost orthogonal for different nodes and assigned similarly to that in [12], so a receiver node can get the channel information of different transmitter nodes from concurrently received RTSs. Our transmitter node selection algorithm in Section4.1 adaptively selects a subset of nodes in a neighborhood to participate in channel estimations based on the decoding capabilities of nodes in the neighborhood, which not only reduces the channel estimation complexity and avoids unnecessary channel estimations but also constrains the total interference in a neighborhood for better decoding.

Phase II: Transmission Confirmation. Upon receiving multiple RTSs, a receiver correlates its received signals with each element in its set of random codes to differentiate the training sequences from different transmitter nodes, estimates spatial channels and extracts other information included in RTSs.

If a node n_k receives a request for P-slot transmission to itself, it sorts all P-slot requests it receives (for itself or for other receiver nodes) based on the request priorities. When multiple requests have the same priority, the request for the receiver with a higher ID is preferred. The receiver n_k then checks the number of P-slot transmissions allowed in the neighborhood from higher priority to lower priority until all the requests are accommodated or n_k is fully-loaded with data and/or interference streams. Denote the number of P-slot requests accommodated at node k as $N_{k,p}^{dec}$, which does not exceed the receiving constraint of n_k, $\mathcal{N}_k^{rc}$. If n_k is a target receiver of some of the accommodated requests, it considers the current transmission duration as P-slot and broadcasts a CTS with its list of confirmed P-slot requests.

If n_k is only the target receiver of some R-slot requests, while it may overhear some P-slot requests for other receivers, it checks whether it has enough residual stream $N_{k,r}^{dec} = \max\{0, \mathcal{N}_k^{rc} - N_{k,p}^{dec}\}$ for R-slot transmission. If $N_{k,r}^{dec} > 0$, it considers the current transmission duration as R-slot. Different from P-slot transmission in which a transmitter node pre-selects a target receiver, transmission streams are flexibly selected for different receivers in R-slot based on channel condition to improve aggregate data rate. After node n_k decodes the information in RTSs from all the selected transmitter nodes in its neighborhood, it learns the number of R-slot streams it may receive in the current duration, $N_{k,r}^0$, including the data streams targeted to itself and the interference streams targeted to other nodes. Denote all transmitter nodes in the one-hop neighborhood of n_k as $\mathscr{V}_k^t$, and each transmitter n_j requires N_j^0 R-slot streams for transmission, we have $N_{k,r}^0 = \sum_{j \in \mathscr{V}_k^t} N_j^0$. Node n_k then broadcasts $N_{k,r}^0$ and $N_{k,r}^{dec}$ through CTS.

A stream may have poor quality, when there is a big distance or deep-fading channel between a transmitter and a receiver. A receiver estimates the strength of a data stream based on the signal-to-noise-ratio (SNR) of the training signal. If the received SNR is lower than a threshold, it includes a weak-channel indicator in the CTS to inform the transmitter to assume more reliable transmission scheme. If a receiver only allows one stream to be transmitted or it only receives one RTS while it is the target receiver, it feeds back this information through CTS to facilitate using of precoding at the transmitter.

To allow the transmitter to estimate the spatial channels to the receiver, a CTS is rotationally broadcast from node n_k's antennas $1 \sim N_k^{ant}$ and the preamble of CTS is utilized as a short training sequence, as in the case of RTS. A CTS signal is also masked by the ID code of n_k.

Phase III: Stream Allocation and Transmission. By differentiating multiple CTSs and extracting the information included, a node n_i estimates the channel matrix $\mathbf{H}_{ki}$ between itself and each active receiver node n_k, and obtains its transmitting constraint value $\mathcal{N}_i^{tx}$. Specifically, if node n_i sends out a P-slot request in the RTS phase, it checks if its P-slot request has been confirmed by all the CTSs. If so, the number of streams allowed for transmission N_i^{allo} is N_i^0. Node n_i allocates the stream following the procedure in 4.2 according to the estimated spatial channels; otherwise, its P-slot request has a relatively lower priority so it cannot transmit in data slot.

If n_i receives a confirmation for its R-slot request, it has to determine N_i^{allo} based on the total R-slot confirmations included in the CTSs from rich neighboring receivers in the set $\mathscr{V}_i^r$. Each responding receiver n_k sends back the total number of streams it may receive, $N_{k,r}^0$, the maximum number of streams it can decode, $N_{k,r}^{dec}$, and possibly weak-channel indicators. In order to ensure all the receiver nodes in its neighborhood to have high probability of correct decoding, node n_i constrains its number of sending streams to a number as $N_i^{allo} = \min\{\mathcal{N}_i^{tx}, N_i^0 \min_{k=1}^{N_i^r} \left(N_{k,r}^{dec}/N_{k,r}^0\right)\}$.

With the estimation of all spatial channels between n_i and its target nodes, the set Q_i of stream capacity factors is constructed and the stream allocation described in 4.2 is then performed to transmit the data streams through the selected antennas. Meanwhile, receiver nodes decode streams from the neighboring transmitter nodes using channel coefficients estimated in phase I.

Phase IV: Acknowledgement. If a data stream is decoded correctly, the receiver node has to confirm the reception. An ACK is masked with the ID code of the receiver and broadcast, carrying the IDs of the transmitter nodes whose streams have been correctly received.

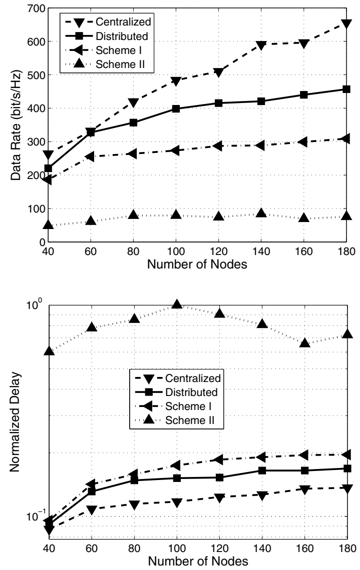
In phase IV, all transmitter nodes are in listening mode. A transmitter node extracts the information in ACKs and removes the correctly received data packets from the queue, and keeps the erroneously received or lost data packets in the queue for scheduling in the next transmission duration.

Note that random ID codes are only used for differentiation in control signal transmission. As control signals are relatively short and sent at the maximum power, there is no significant overhead induced for packet encoding and decoding and there is no need for power control.

5. Performance Evaluation

In this section, we evaluate the performance of our proposed algorithms through simulations. Nodes are distributed uniformly over a $1250m \times 1250m$ area and form an ad-hoc network with random topology. Each node has a transmission range of $250m$. To model a heterogeneous MIMO ad hoc network, we assume the degree-of-freedom of channels between node pairs are normally distributed with a given mean and variance. The channel is modeled based on the antenna array sizes, the distance between nodes and the small-scale fading coefficients following Rayleigh/Ricean model. The incoming traffic is Poisson distributed with a given mean value λ and the sources and destinations are chosen at random. A simulation result is obtained by averaging over several runs of simulations with different seeds.

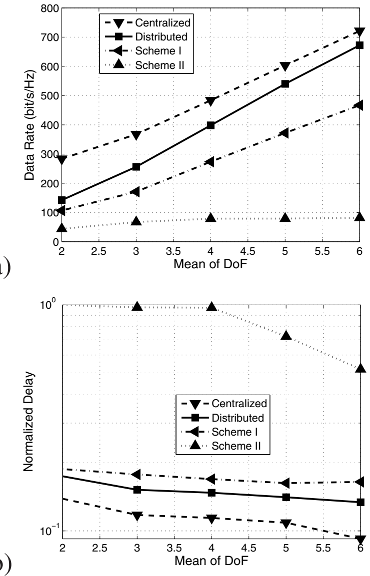
The distributed scheduling algorithm proposed in Section 4, including both transmitter nodes selection 4.1 and stream allocation 4.2, is implemented based on the protocol described in Section 4.3. Compared with conventional scheduling strategies in MIMO ad hoc networks, our dis-



(a)

(b)

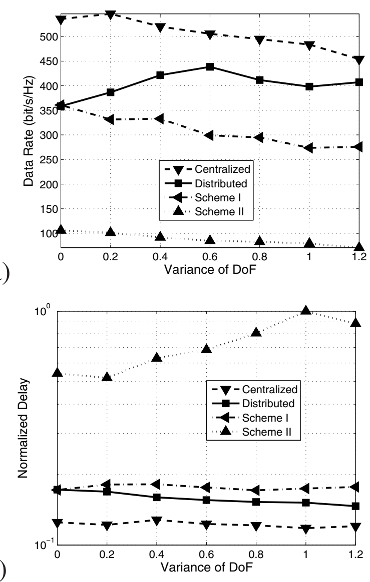
Figure 2. Impact of node density: (a) data rate; (b) delay.



(a)

(b)

Figure 3. Impact of mean of degree-of-freedom: (a) data rate; (b) delay.



(a)

(b)

Figure 4. Impact of variance of degree-of-freedom: (a) data rate; (b) delay.

tributed algorithm has the following unique features: adaptive use of different transmission strategies based on node types and channel condition, enabling multi-user to multi-user transmissions through both multiplexing and selective diversity and supporting priority-aware transmitter nodes selection. To demonstrate the benefits of these features, we design two alternative schemes here for reference. *Scheme I* is based on [9], which supports many-to-many cooperative transmission, but does not have strategies to handle the heterogeneity of nodes and channels. *Scheme II* takes the conventional scheduling strategy in MIMO ad hoc networks, where only one pair of transmitter/receiver nodes is allowed to communicate in a neighborhood using as many antenna elements as possible. In each transmission duration, the node pair with the best channel quality is selected, and transmitter node selection is also implemented here to reduce collision. To provide a benchmark for performance comparison, we also implemented the centralized scheduling algorithm proposed in Section 3.3.

The metrics we use for comparison are aggregate data rate and normalized delay. Aggregate data rate is the total data rates of the network averaged over the number of transmission durations. Delay time is defined as the number of transmission durations a packet waits in the queue before it is successfully transmitted. We investigate four impacting factors, namely node density, mean value of degree-of-freedom, and variance of degree-of-freedom. For each factor, the centralized algorithm and distributed algorithm as well as the two reference schemes are implemented, and both data rate and normalized delay are compared. If not otherwise specified, the number of nodes in the network is 100, the mean and variance of degree-of-freedom are 4 and 1 respectively and the average packet arrival rate λ is 0.5.

(1) Impact of Node Density. The impact of node density is shown in Fig 2. Irrespective of the density, the distributed algorithm has the closest performance to that of the centralized algorithm in terms of both aggregate data rate and normalized delay. Compared with scheme I, with adaptive selection of transmission strategy based on node types and channel conditions, our distributed algorithm is shown to have up to 50% higher aggregate data rate and 20% lower delay. Compared to scheme II, the distributed algorithm achieves up to 5.5 times the data rate and 75% lower delay. As expected, with only single-user to single-user links, scheme II cannot exploit the transmission potential of nodes and has the smallest data rate and the largest delay.

(2) Impact of the Mean of Degree-of-freedom. The mean value of degree-of-freedom impacts the overall capacity of the network. In Fig 3, as the mean value grows, all schemes obtain higher data rate and lower delay, except scheme II as its performance is severely constrained by the poorest node in a neighborhood. The data rate of the distributed algorithm increases much faster than that of scheme I, as the performance of the latter is constrained more by the heterogeneity of the nodes.

(3) Impact of the Variance of Degree-of-freedom. The variance of degree-of-freedom reflects the degree of heterogeneity of nodes in the network. The larger the variance is, the greater the variety of degree-of-freedom is. As shown in Fig 4, when the variance is 0, which is the homogeneous case, the distributed algorithm and scheme I have very close performance. However, as the variance increases, the performance of scheme I is constrained more by the bottleneck effect caused by channels with low degree of freedom, so its data rate constantly decrease. On the contrary, the distributed algorithm achieves relatively high data rate under all the variance values, up to 47% higher rate and 20% lower delay compared with scheme I. This demonstrates that by differentiating between poor nodes and rich nodes and adaptively scheduling transmissions in the network based on the number of antennas in the receiver nodes and channel conditions, our distributed algorithm can effectively alleviate the impact of node heterogeneity and channel variations to achieve better performance.

6. Conclusions

It is important and challenging to coordinate transmissions in a heterogeneous MIMO-based distributed system with mobile devices having different number of antennas, in presence of channel dynamics and network topology changes. In this work, we propose an effective distributed scheduling algorithm in MIMO-based ad hoc networks by concurrently considering traffic demand, service requirements, node heterogeneity, channel condition, network load, multiuser diversity and spatial diversity. Our algorithm adaptively assumes different transmission strategies for receiver nodes based on their decoding capacity to alleviate the bottlenecks caused by nodes with smaller antenna arrays, and avoid transmission failure due to channel degree of freedom constraint. Our scheduling algorithm also exploits both multiplexing and diversity to opportunistically select transmitter nodes and antennas to improve transmission rate and reliability, while supporting QoS and fairness. Nodes in a neighborhood can cooperate in transmission and form a many-to-many virtual MIMO array. We form a concrete channel model, and apply the channel model in our algorithm design to efficiently optimize network performance. Our performance results demonstrate that our proposed scheduling algorithm is very efficient in coordinating transmissions in a MIMO-based ad hoc network, achieving up to 5.5 times the data rate and reducing the transmission delay up to 74% compared with the scheme of selecting only one user pair at a time as often used in conventional MIMO schemes. Compared with the scheme not considering node heterogeneity and channel constraint, our scheduling algorithm can achieve about 50% higher data rate and 20% lower delay.

References

[1] M. Hu and J. Zhang. MIMO Ad hoc Networks: Medium Access Control, Saturation Throughput, and Optimal Hop Distance. *Journal of Communications and Networks, Special Issue on Mobile Ad Hoc Networks*, pages 317–330, 2004.

[2] J-S. Park, A. Nandan, M. Gerla, and H. Lee. SPACE-MAC: Enabling Spatial-reuse Using MIMO Channel-Aware MAC. In *Proc. IEEE ICC 2005*, May 2005.

[3] M. Levorato, S. Tomasin, P. Casari, and M. Zorzi. Analysis of Spatial Multiplexing for Cross-Layer Design of MIMO Ad Hoc Networks. In *Proc. IEEE VTC-2006 Spring*, pages 1146–1150, May 2006.

[4] M Zorzi, J Zeidler, A. Anderson, B. Rao, J. Proakis, A.L. Swindlehurst, M. Jensen, and S. Krishnamurthy. Cross-layer Issues in MAC Protocol Design for MIMO Ad hoc Networks. *IEEE Wireless Communication Magzine*, (4):62–76, August 2006.

[5] P. Casari, M. Levorato, and M. Zorzi. DSMA: an Access Method for MIMO Ad Hoc Networks Based on Distributed Scheduling. In *Proc. ACM IWCMC 2006*, July 2006.

[6] K. Sundaresan, R. Sivakumar, M. Ingram, and T-Y. Chang. A Fair Medium Access Control Protocol for Ad-hoc Networks with MIMO Links. In *Proc. IEEE INFOCOM 2004*, June 2004.

[7] K. Sundaresan and R. Sivakumar. A Unified MAC Layer Framework for Ad-hoc Networks with Smart Antennas. In *Proc. ACM MobiHoc 2004*, pages 244–255, May 2004.

[8] M. Park, R. Heath, and S. Nettles. Improving Throughput and Fairness of MIMO Ad hoc Networks Using Antenna Selection Diversity. In *Proc. IEEE Globecom 2004*, November 2004.

[9] S. Chu and X. Wang. Opportunistic and Cooperative Spatial Multiplexing in MIMO Ad hoc Networks. In *Proc. ACM MobiHoc 2008*, pages 63–72, May 2008.

[10] Y. Pan and S. Aissa. Performance Analysis of Selective Space-time Coding and Selection Diversity Under Perfect and Imperfect CSI. In *Proc. PIMRC 2005*, pages 2371–2375, September 2005.

[11] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, May 2005.

[12] R. M. de Moraes, H. R. Sadjadpour, and J. J. Garcia-Luna-Aceves. Many-to-Many Communication: A New Approach for Collaboration in MANETs. In *Proc. IEEE INFOCOM 2007*, pages 1829–1837, May 2007.