Opportunistic and Cooperative Spatial Multiplexing in MIMO Ad hoc Networks

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

ABSTRACT

It is important and challenging to develop efficient schemes to coordinate node transmissions in a MIMO-based ad hoc network. In this work, we propose a scheme to fully exploit the multiuser diversity and spatial diversity by taking advantage of the meshed topology, while also supporting user transmission quality requirement. We formulate a concrete physical model, and present cross-layer algorithms which take advantage of physical layer channel information to opportunistically schedule cooperative spatial multiplexed transmissions between nodes, so that the data rate of the network can be maximized. The performance of our algorithm is studied by extensive simulations and the results demonstrate that our algorithm is very effective and can significantly increase the throughput while reducing the transmission delay.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Algorithms, Design, Performance

Keywords

Spatial multiplexing, MIMO, ad hoc networks, scheduling, cross-layer design

1. INTRODUCTION

Multiple-input multiple-output (MIMO) technology utilizes multiple antennas at the transmitter and the receiver with the aim of improving transmission reliability and providing higher raw data rates. A transmitter node can divide its data into multiple data streams and transmit them simultaneously over multiple antenna elements, which is known as spatial multiplexing. As a rich scattering environment can provide independent transmission paths (multi-channels) between different transmitting and receiving antenna pairs, an intended receiver node can separate and decode its received data streams based on their unique spatial signatures. In addition to improving the transmission capacity through spatial multiplexing, in a network with multiple users, the channels between different user and antenna pairs are different and vary over time. In cellular networks, multiuser diversity could be exploited by scheduling the user with the best channel condition to communicate with the base station [2, 3, 4].

With the fast progress of MIMO technology, it is now being adopted in 802.11n [1] and is also considered for ad hoc networks. Different from an infrastructure-based single-hop cellular network, it is difficult for nodes to coordinate in channel evaluations and transmissions in a dynamic ad hoc network. Different nodes may have different number of antennas, and the peer relationship changes as network topology changes. The quick variation of channel condition and network topology as well as the inconsistency in node density would lead to more challenges in ad hoc network design. Instead of simply extending the algorithms used in cellular networks, an efficient algorithm is needed to better exploit the peer-to-peer nature of the network and the varying channel condition to maximize the data rate of the network. Although there are many recent efforts in developing MAC protocols for applying MIMO technique to ad hoc networks [5, 6, 8, 10, 11, 12, 13], there is very limited work to fully exploit the meshed topology of ad hoc networks and consider both multiuser diversity and spatial diversity to maximize network capacity. In addition, the traffic at each node may be different and the user packets may have different service requirements, which leads to more open problems for the MAC protocol design in MIMO-based ad hoc networks.

In this work, we propose an integrated scheduling scheme to improve the network throughput and transmission quality in MIMO-based ad hoc networks by jointly considering traffic demand, service requirements, network load, multiuser diversity, and channel condition. In our scheme, a sender node can transmit to multiple downstream nodes using different antennas, while a receiver node can receive packets from multiple upstream nodes. Therefore, a group of neighboring nodes can take advantage of the meshed network topology to cooperate in transmission and form a *virtual* MIMO array. In a transmission duration, transmitter nodes and their antenna set are selected opportunistically to exploit the multiuser diversity and spatial diversity to a large

MobiHoc'08, May 26–30, 2008, Hong Kong SAR, China.

degree, while supporting different transmission priorities, reducing transmission delay and ensuring fair transmissions among nodes. Our scheduling scheme is *cross-layer*, with the consideration of physical channel condition and transmission power in MAC design. The main contributions of this paper are summarized as follows.

- We formally formulate the multiuser MIMO scheduling problem using graph approach, and divide it into two subproblems.
- We form a concrete physical layer model, and provide efficient methodologies to evaluate channel coefficients and interference, in the presence of a large number of nodes competing in transmission. This can reduce the gap between physical layer theoretical studies and practical implementation of the algorithm in network to improve performance.
- We propose a centralized algorithm to use as performance benchmark, and a distributed algorithm for practical implementation. Both algorithms take advantage of the multiuser diversity and spatial diversity by opportunistically selecting the nodes and antennas with good channel conditions to form virtual transmission array and maximize the spatial multiplexing gain.
- We develop schemes to specifically consider the service requirements of the user traffic, the transmission delay, and the fairness among nodes.
- We propose a new MAC scheme to better work in a MIMO-based multi-packet reception network, and to support our distributed algorithm design.

The rest of the paper is organized as follows. Section 2 discusses the related work. We introduce the system architecture in Section 3 and formally formulate the problem in Section 4. We describe our physical model in Section 5, and propose our centralized and distributed algorithms in Section 6 and 7 respectively. In Section 8, we present our MAC protocol. Simulation results are given in Section 9 and the paper is concluded in Section 10.

2. RELATED WORK

Over the past several years, the application of MIMO technology in networks has undergone a fast development.

Many studies have been performed to develop scheduling schemes to select the best user to transmit based on certain criteria in a multiuser MIMO-based cellular network. In [2], an overview of scheduling algorithms in MIMO-based fourth-generation wireless systems is given, and the relationship of spatial and multiuser diversity is also investigated. Paper [3] addresses the design of the optimal space time scheduler for multiuser MIMO system based on an information theory approach. In [4], the authors argue that both multiuser and spatial diversity can be exploited with more bits of feedback information.

In recent years, many efforts have been made to support MIMO transmission in ad hoc networks. In [5], spatial diversity (e.g. space time coding (STC)) is explored to combat fading and achieve robustness. SPACE-MAC, proposed in [6], enables denser spatial reuse patterns with the aid of transmitter and receiver beamforming. Authors in [10] introduce a distributed scheduling (DSMA) scheme within the CSMA/CA framework where the stream allocation depends on the transmitter-receiver distance. Layered space-time multiuser detection and its role in PHY-MAC cross-layer design are analyzed in [8]. In [13], spatial multiplexing with antenna subset selection for data packet transmission is proposed. In [10, 8, 13], a user can only be scheduled to transmit to one receiver node, and the selected user is allowed to use all or a subset of its antennas for transmission. In [11], the authors discuss key optimization considerations for MAC layer design in ad hoc networks with MIMO links, and develop a centralized algorithm and a distributed algorithm. However, there is no description on how to obtain the parameters necessary for stream selection and performance optimization, which is very challenging in ad hoc networks. A unified representation of the physical layer capabilities of different types of smart antennas, and unified medium access algorithms are presented in [12]. In these literature works, spatial diversity and multiuser diversity are not fully exploited. There is no support of QoS and consideration of the difference in node traffic demand.

3. SYSTEM ARCHITECTURE

We consider an ad hoc network where each node is equipped with an antenna array. The number of antenna elements may vary from node to node. Our MAC design 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 higher rate. Due to the peer-to-peer nature of nodal interaction in ad hoc networks, the total transmit power at each node is considered to be fixed, while the transmit power of each antenna is different when a node uses a different number of antennas for transmission. 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. A node can transmit multiple streams to several downstream nodes or receive multiple streams from several upstream nodes simultaneously. Therefore, a *virtual* MIMO array can be formed among a group of nodes. As we focus on spatial multiplexing in this paper, spatial diversity is not considered for range extension and transmission range is assumed to be uniform.

A stream is identified by a triplet $(I_{TX}, I_{RX}, I_{ANT})$, where $I_{TX}/I_{RX}/I_{ANT}$ are the index of the transmitter node/ receiver node/ transmitter antenna respectively. At a transmitter node, independent data streams are transmitted from selected antenna elements. The total number of transmitted streams from a node is obviously limited by the total number of antenna elements of the node. Due to the broadcast nature of wireless links, a stream transmitted from a node i to its one-hop neighbor j is also received by all the other one-hop neighbor nodes of i, which causes interference at these nodes. To differentiate the streams received at a node j, we call the streams targeted for j as *data streams*, and the streams not for j as *interference streams*. Thanks to multiple antennas, a node is endowed with multiple packet reception (MPR) capability so that it can receive data streams and suppress interference streams concurrently. Note that the total number of data streams and interference streams received at a node is also constrained by its degree of freedom (DOF), which is approximately equal to its number of antennas in a rich scattering environment.

As it is hard to maintain a central controller in a practical ad hoc network and a node can not be a transmitter and receiver at the same time, our distributed scheduling algorithm has two phases, namely *transmitter nodes selection* and *stream allocation*. That is, a set of nodes are first selected to be transmitter nodes based on their priority and the current network topology, then the streams with higher priority and/or better quality are allocated from the selected set of transmitter nodes.

In the first phase of the scheduling, instead of randomly selecting a set of transmitter nodes, our scheduling algorithm only selects active nodes that have packets for transmissions, and the selection is based on the priority of packets which depends on both the service type and the delay time of packets. In the second phase of scheduling, stream allocation is performed so that data packets of the transmitter nodes are allocated to a selected set of antennas for transmission. As discussed later in Section 5, multiple antennas at both ends of a link create multiple independent spatial channels with different channel gains in a rich scattering environment, which makes channel capacity or achievable data rates of the streams different. To maximize the temporal throughput of the network, it is important to allocate streams opportunistically by taking advantage of different channel gains, while considering the priority of data packets.

To capture the characteristics of a stream p, two parameters are defined below.

. **stream priority** $P(p)$: It depends on the type of the data to be sent with the stream and the delay time of the current data packet. A higher value of $P(p)$ indicates higher priority of stream p.

. **stream quality** $Q(p)$: It describes the reliability of a stream transmission, which depends on the transmission power of the stream (which will reduce when more streams are selected from the same sending node) and the channel condition between the transmitter antenna and the receiver node of this stream (which can be represented by a vector function as discussed later).

4. PROBLEM FORMULATION

In this section, we use graph representation to formally formulate the two-phase scheduling problem described in Section 3. We first describe graph construction guideline and constraints for scheduling, and then formulate the problem formally.

4.1 Graph Construction

A directed graph $G = (V, E)$ is used to model the traffic demand of the network. Each node is represented by a vertex $v \in V$. A directed edge in the graph denotes a candidate transmission stream between a sender and a receiver. We name edges corresponding to data streams as *data edges*, and edges representing interference streams as *interference edges*. For instance, at the beginning of a transmission duration, node i has one packet targeted for node k, so there is a candidate transmission stream from i to k. A data edge appears in G with i as the source vertex and k as the destination vertex; meanwhile, if $j_1, \ldots, j_n$ are nodes in the one-hop neighborhood of i, a set of interference edges are formulated from i to $j_1, \ldots, j_n$.

Edges are scheduled in sets. Each set $\{e\}_i$ consists of one data edge $e_i = (i_s, i_t)$ and $N_i - 1$ interference edges $\{\bar{e}\}_i$, where i_s /i_t are the source/destination vertices of e_i and N_i is the number of nodes within the transmission range of i_s.

The stream parameters defined in Section 3 thus become the parameters of edges. The stream priority depends on the data packet, thus one data edge and its corresponding interference edge(s) share the same stream priority parameter. The stream quality of an edge depends on the spatial channel between the transmit antenna of the stream and the target receiver node, and is associated with the stream triplet $(I_{TX}, I_{RX}, I_{ANT})$ described in Section 3. As e_i and $\{\bar{e}\}_i$ are associated with different stream triplets, they have different stream quality. The assignment of the triplet to a data stream is decided by the scheduling algorithm, while the interference streams are caused by the existence of the data stream. The achievable data rate of a data stream e_i, $C(e_i)$ can be calculated based on the stream quality of e_i and all the interfering streams received at node i_t.

At a certain transmission duration k, all candidate streams of the network form a graph G_k. The scheduling is performed to select a subset of the data streams for transmission. The selected data streams and their resulted interference streams along with their senders and receivers form a subgraph of G_k, which is called G^k_{opt}. Denote the number of outgoing data edges connected to a vertex i as $d_i(out)$, the number of incoming data edges connected to a vertex i as $d_i(in.data)$, the number of incoming interference edges connected to a vertex i as $d_i(in.int)$, and the number of antennas at the node of vertex i as n_i. Due to the limitation of decoding capability of nodes and the half-duplex characteristic of links, the degrees of nodes are subject to the following constraints.

Degree Constraint (DC)

At a transmission duration k, one and only one of the three constraints is satisfied for a vertex i in subgraph G^k_{opt}:
(1) $0 < d_i(out) \leq n_i$;
(2) $d_i(out) = 0$, $d_i(in.data) \neq 0$, and $d_i(in.data) + d_i(in.int) \leq (1 + \alpha)n_i$;
(3) $d_i(out) = 0$ and $d_i(in.data) = 0$.

If constraint (1) is satisfied, the node is classified as a *transmitter node*, and the total number of outgoing streams at a certain time cannot exceed its number of antennas. If constraint (2) is satisfied, a node receives some streams targeted for it, so it is an active *receiver node*. The parameter $\alpha \geq 0$ is called overload factor, which depends on the decoding capacity of the receiver node [18], and the condition $d(in.data) + d(in.int) \leq (1 + \alpha)n$ is used to constrain the total number of incoming streams at a receiver node so that data streams can be decoded while interference streams can be suppressed. If constraint (3) is satisfied, the node is an *idle node*, and is not currently involved in either transmitting or receiving in the network. A node is called *fully loaded* if $d_i(out) = n_i$ for a transmitter node or $d_i(in.data) + d_i(in.int) = (1 + \alpha)n_i$ for a receiver node.

4.2 Problem Definition

In dynamic networks, a node gets data packets from its upper layers from time to time, and it is impossible to have the information of all data packets in advance. Moreover, the spatial channels between nodes may vary over time. Therefore, the scheduling can be modeled as an iterative optimum subgraph selection problem at each transmission duration (TD). The residual graph is updated and left to the next

TD for processing. We first define the optimum subgraph problem as follows.

Optimum Subgraph Problem (OSGP)

Select a subgraph G_{opt} of graph G, with antenna allocation L_{opt}, such that:
(1) G_{opt} satisfies **constraint DC**;
(2) Optimum Priority: Denote a residual graph $G_- = G - G_{opt}$. For any edge e_x in G_- whose stream priority is higher than the lowest stream priority of the edges in G_{opt}, **DC** cannot be held if e_x is added to G_{opt}.
(3) Optimum Capacity: Denote the set of data edges in G_{opt} as $\{e\}_{data}$. The total achievable data rate of G_{opt} is therefore $C(G_{opt}) = \sum_{p\in\{e\}_{data}} C(p)$. There does not exist another subgraph $G(k)$ with antenna allocation $L(k)$, which also satisfies (1) and (2), such that $C(G(k)) > C(G_{opt})$.

Basically, OSGP is to find a solution that satisfies all three conditions. First, the subgraph selected should meet the degree constraint. Second, the higher priority streams are selected to form the subgraph. Third, the subgraph selected should achieve optimum aggregate capacity.

If OSGP can be solved, the multi-user multi-stream scheduling can be performed in an iterative way as below.

Multi-User Multi-Stream Scheduling(MUMSS)

```
Initialization: $G_0 \leftarrow G$
for transmission duration $k = 1, 2, \ldots$
  - Update $G_{k-1}$ according to new traffic demands and
    updated priority/quality, the new graph is $G_k$;
  - OSGP($G_k$), get graph $G^k_{opt}$;
  - Send data frames according to $G^k_{opt}$;
  - $G_k \leftarrow G_k - G^k_{opt}$;
end
```

Our scheme is TDMA based by scheduling transmission in each transmission time duration. Although promising [14], the application of TDMA in ad-hoc networks leads to the known NP-complete Broadcast Scheduling Problem (BSP) [15]. Therefore, efficient algorithms to provide suboptimal solution of the problem need to be developed.

So far, we formulate the problem of scheduling using graph representation. In the next section, we discuss the physical layer model which is the basis of our MUMSS algorithm design.

5. PHYSICAL MODEL

In wireless communications, time-varying fading is commonly observed due to user mobility or the variation of propagation environments [17]. A fading channel can generally be expressed as

$$h = ae^{j\phi} + b, \tag{1}$$

where $ae^{j\phi}$denotes the LOS component and b denotes the time-varying component of the fading. When the LOS component is very weak, the channel can be well modeled by Rayleigh fading.

Consider two nodes i and k which are within transmission range of each other, the numbers of antenna elements are n_i and n_k respectively. When the separation of antenna elements at each node is large enough, the spatial channels between node i and k are i.i.d fading channels, which can be represented as a $n_k \times n_i$ matrix $\mathbf{H_{ki}}$:

$$\mathbf{H_{ki}} = \begin{pmatrix} h_{11} & h_{12} & \ldots & h_{1n_i} \\ h_{21} & h_{22} & \ldots & h_{2n_i} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n_k 1} & h_{n_k 2} & \ldots & h_{n_k n_i} \end{pmatrix} \tag{2}$$

where h_{pq} is the spatial channel coefficient between the p-th antenna of node k and q-th antenna of node i, and can be represented as in (1).

Let node i be the transmitter node at a particular time slot, then the transmitted signal can be represented as a vector

$$\mathbf{s} = \begin{pmatrix} s_1 & s_2 & \ldots & s_{n_i} \end{pmatrix}^T \tag{3}$$

where $s_1, s_2, \ldots, s_{n_i}$ are symbols transmitted at antenna $1, 2, \ldots, n_i$. Note that $s_1, s_2, \ldots, s_{n_i}$ may have different target receiver nodes.

Consider an active node k with n_k antennas within the transmission range of node i. A receiving node is considered active if it is either a target receiver or a passive listening node of a transmission. Therefore, the faded signal from node i received at node k can be represented as:

$$\mathbf{r}_{ki} = \mathbf{H}_{ki}\mathbf{s} = \begin{pmatrix} \sum_{p=1}^{n_i} h_{1p}s_p \\ \sum_{p=1}^{n_i} h_{2p}s_p \\ \vdots \\ \sum_{p=1}^{n_i} h_{n_k p}s_p \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{p\in X_{ki}} h_{1p}s_p \\ \sum_{p\in X_{ki}} h_{2p}s_p \\ \vdots \\ \sum_{p\in X_{ki}} h_{n_k p}s_p \end{pmatrix} + \begin{pmatrix} \sum_{p\notin X_{ki}} h_{1p}s_p \\ \sum_{p\notin X_{ki}} h_{2p}s_p \\ \vdots \\ \sum_{p\notin X_{ki}} h_{n_k p}s_p \end{pmatrix}$$

$$= \mathbf{r}_{ki,sig} + \mathbf{r}_{int}, \tag{4}$$

where X_{ki} is the set of antennas at node i that transmit signals to node k. Due to the broadcast nature of wireless channels, all signal streams transmitted by node i are received at node k. Therefore, node k has to differentiate streams targeted for itself (data streams) from streams targeted for other nodes (interference streams). Denote the signal to interference and noise ratio of received stream p at node k as $SINR_p$, the sum data rate that receiver node k gets from transmitter node i is:

$$R_{ki} = \sum_{p\in X_{ki}} \log(1 + SINR_p) \tag{5}$$

Denote the set of transmitting nodes that are within the receiving range of node k as J_k, the total sum rate at receiver node k is therefore the summation over all transmitter nodes in J_k:

$$R_k = \sum_{i\in J_k} R_{ki} = \sum_{i\in J_k} \sum_{p\in X_{ki}} \log(1 + SINR_p) \tag{6}$$

The calculation of $SINR_p$ depends on the decoding capacity at the receiver node. According to [17], a way to get optimum performance for multiple stream decoding is using Minimum Mean Square Error Sequential Interference Cancelation (MMSE-SIC) receiver. In this case, the linear MMSE receiver for a stream p is represented by the vector:

$$v_p = K_{z_p}^{-1} h_p \tag{7}$$

The corresponding SINR achieved is

$$SINR_p = \sigma_p^2 h_p^* K_{z_p}^{-1} h_p \tag{8}$$

where h_p is the $n_r \times 1$ channel vector for stream p to a receiver with n_r antennas, K_{z_p} is the covariance of z_p, which is the noise plus interference faced by data stream p: $z_p = \sum_{l>p}^{N_k} h_l s_l + n$, where the set N_k contains all the transmission streams around the receiver k of stream p. Since a receiver using SIC decodes the strongest stream first, all the weaker streams create interference at p. As we consider each node has a fixed transmitting power, the transmitting power of a stream only depends on the number of streams allocated from this node. For instance, denote the total transmitting power of node i as P_i, the number of allocated streams of node i as n_i^{allo}, then the transmitting power of a single stream p is $P_p = P_i / n_i^{allo}$ if the total power is uniformly allocated to each stream. With power P_j associated with data stream j, we can explicitly calculate K_{z_p} as

$$K_{z_p} = N_0 I_{n_r} + \sum_{j>p}^{N_k} P_j h_j h_j^* \tag{9}$$

which is invertible. Substitute (9) into (8), the output SINR for stream p can be calculated as:

$$SINR_p = P_p h_p^* \left(N_0 I_{N_r} + \sum_{j>p}^{N_k} P_j h_j h_j^* \right)^{-1} h_p \tag{10}$$

Substitute (10) into (6), we can calculate the data rate for each receiver node. Therefore, the aggregate data rate of the network is:

$$R = \sum_{k \in S_r} R_k \tag{11}$$

where S_r is the set of all receiver nodes.

Based on the analysis above, stream quality $Q(p)$ introduced in Section 3 can be quantitatively specified here. From (10), it is obvious that the larger the value of $\|h_p\|^2 = h_p^* h_p$ is, the better is the strength of stream p. So a straightforward way to define $Q(p)$ is to simply use the channel vector and the transmitting power:

$$Q(p)_{sim} = P_p(h_p^* h_p) = P_p \|h_p\|^2 \tag{12}$$

However, in order to achieve better aggregate data rate of the whole network, the strength of interference streams caused by a data stream should also be taken into consideration. Unfortunately, an explicit expression can not be deduced from equation (10). Here we define a normalized stream quality to capture the impact of interference streams:

$$Q(p)_{nor} = \frac{P_p \|h_p\|^2}{\sum_{s \in S_{int}} P_s \|h_s\|^2} = \frac{\|h_p\|^2}{\sum_{s \in S_{int}} \|h_s\|^2} \tag{13}$$

where S_{int} is the set of interference streams caused by the data stream p. The definition of stream quality is then used in the following sections for stream allocation.

6. CENTRALIZED ALGORITHM

In this section, we propose a centralized algorithm to solve the MUMSS problem where all the stream information is assumed to be known at a central controller. The design of the centralized algorithm can provide a basis for the distributed algorithm.

CMUMSS: Centralized MUMSS Algorithm

1. Initialization

Check the queue of data packets at every node, and construct a graph G_0 according to Section 4.1. The graph is updated in each transmission duration.

2. Greedy Scheduling

For a transmission duration k ($k = 1, 2, \ldots$), perform the following steps in sequence based on graph G_{k-1}.

I. Pre-scheduling Update

This step is performed at the beginning of a transmission duration. Each vertex i keeps a list L_i^{prio} where its outgoing data edges (associated with to-be scheduled data streams) are ordered in decreasing sequence according to the priority of the corresponding data packets, with the priority calculated based on the service type and delay time of a packet. After checking the new data packets from upper layers for every node, the list for each vertex is updated according to the priority of the new packets. The new edges from all the lists are then added to graph G_{k-1} and the existing weights of G_{k-1} are updated based on the packet delay. The updated graph is denoted as G_k. Let the optimum subgraph $G_{opt} = NULL$. Create another subgraph called blocked graph G_b, used to save edge sets that cannot be scheduled in the current duration, and set $G_b = NULL$. Each node is allowed either to be a transmitter node or a receiver node at this stage.

II. Stream Allocation

Select one edge with the highest priority in L_i^{prio} from each vertex i to form a set $\{e\}_h$. The j-th element in $\{e\}_h$ is a data edge denoted as $e_h(j) = (s_j, t_j)$, where s_j and t_j are the source and destination vertices of $e_h(j)$ respectively. $e_h(j)$ and its corresponding interference edges form a set $\{e_h(j)\}$. Sort all the elements in $\{e\}_h$ according to their priority. The set $\{e\}_h$ can then be reorganized into another set $\{e\}_{prio}$, where each element in the set consists of edges in $\{e\}_h$ that have the same priority.

```
for $j = 1 \rightarrow |\{e\}_{prio}|$
  - Denote the $j$-th element of $\{e\}_{prio}$ as a set $e_{prio}(j)$,
    the $q$-th element in $e_{prio}(j)$ along with its interference edges as $\{e_h(q)\}$;
  - Construct a set:
    $S_j = \{Q(a,B) | e_h(q) = (s_q, t_q), a \in A_{s_q}, B = t_q, \forall e_h(q) \in e_{prio}(j)\}$
    where $A_{s_q}$ is the set of unused antennas at $s_q$,
    $Q(a,B)$ is the stream quality factor for stream between antenna $a$ and node $B$;
  - for $q = 1 \rightarrow |e_{prio}(j)|$
    – Find the largest element in $S_j$, denote it as $Q_{max}$, and the corresponding
      transmitter node, receiver node and antenna as $s_{max}$, $t_{max}$ and $a_{max}$ respectively;
    – If $s_{max}$ is marked as a receiver node or $t_{max}$ is marked as a transmitter node,
      remove the set $\{e\}$ where $e = (s_{max}, t_{max})$ from $G_k$ and add it to $G_b$;
```

```
        – Else:
          Tentatively add $\{e\}$ to $G_{opt}$. Check whether
          DC is still satisfied for $G_{opt}$.
            . If no, remove $\{e\}$ from $G_k$ and add it to
              $G_b$;
            . Else, mark $s_{max}$ as a transmitter node
              and $t_{max}$ as a receiver nodes if they are
              not currently marked. Assign $e$ to the
              antenna $a_{max}$, add $\{e\}$ along with the allocation
              information to $G_{opt}$, and update
              $A_{s_{max}}$. Meanwhile, if any vertex associated
              with $\{e\}$ becomes fully loaded, delete
              all edge sets that may overload it from $G_k$
              and add them to $G_b$. Delete elements associated
              with $a_{max}$ from $S_j$.
        – Delete $Q_{max}$ from $S_j$.
    - end
end

III. End Check
Check whether there is still any edge set in $G_k$. If yes,
go to (II); else got to (IV).

IV. Post-scheduling update
The optimum subgraph for this transmission duration
is generated. Schedule the transmissions according to
graph $G_{opt}$. Add the edges in $G_b$ back to $G_k$, which
will be used for scheduling in the next transmission
duration.
```

The centralized algorithm is used as a benchmark to evaluate the performance of the distributed algorithm to be presented next.

7. DISTRIBUTED ALGORITHM

As introduced in Section 3, the scheduling algorithm actually involves two phases, namely transmitter nodes selection and stream allocation. Although the two problems can be considered together in the centralized algorithm to achieve better overall performance, in the distributed case without a central controller, a node always has to decide whether it is a transmitter node first. Then the candidate outgoing streams of the selected transmitter nodes are compared, and the streams with higher priority and/or better quality are allocated for transmissions. In this section, we describe our algorithms for distributed transmitter nodes selection (DTNS) and distributed stream allocation (DSA) in detail, and DTNS and DSA jointly form the distributed MUMSS solution (DMUMSS).

7.1 Transmitter Nodes Selection

As the transmission is half-duplex and a node cannot be a transmitter and a receiver at the same time, it is necessary to select a subset of the nodes to serve as the transmitter in a transmission duration. Instead of randomly selecting the transmitter nodes, our DTNS supports service differentiation and reduces transmission delay by giving higher transmission priority to the packets that are in higher service class and/or have larger queuing delay.

We consider a node with packets to transmit an *active* node. To select a subset of nodes to be transmitter nodes in a neighborhood, we introduce a probability P^{TX}, below which an active node can be selected as a transmitter node. The parameter P^{TX} is estimated by each node based on the number of active nodes around each neighboring node j and the maximum number of simultaneous flows allowed by j in its neighborhood. That is, a node estimates P^{TX} based on its two-hop information announced through a Hello message at network layer. In a neighborhood with n nodes, in order to not exceed the decoding capacity of any node at data transmission time, the number of streams that can be simultaneously transmitted in the neighborhood is constrained. Therefore, we constrain the number of transmitter nodes as well to this value to avoid unnecessary channel measurement, reduce processing complexity at a receiver, and better serve higher priority packets. For each active node i, denote the number of its neighboring nodes as n_i^n, the number of streams that can be decoded at its neighboring node j as N_j^{dec}, and the number of active nodes around j as n_j^a, P^{TX} at node i is calculated as follows:

$$P^{TX} = \min_{j=1}^{n_i^n} \left(\frac{N_j^{dec}}{n_j^a} \right) \qquad (14)$$

Note that our selection is more conservative for a node to consider the decoding capability of all its neighbors instead of only the selected receiver nodes, whose information is not available at the selection time.

An active node will then decide if it can be selected as a transmitter node based on P^{TX} and the priority of its packets, which depends on the service type and delay time of the packets. A possible way to integrate both factors into the priority calculation is to let a packet to have its initial priority equal to its service priority number, and the priority of the packet will be increased as its queuing time increases. Assume node i has N_i^{pkt} packets and the priority of the m-th packet in queue is $p_i^{pkt}(m)$, the priority of node i can be calculated as $p(i) = \sum_{m=1}^{N_i^{pkt}} p_i^{pkt}(m)/N_i^{pkt}$. Before a node has any data transmission, it can attach its initial priority with the Hello message sent out. Thereafter, the updated priority is attached with each packet it sends out. A node with priority 0 is idle.

A node can calculate the average priority, $\bar{p}$, of all the active nodes in its neighborhood as $\bar{p} = (\sum_{i=1}^{n^a} p(i))/n^a$. Nodes with higher priority should be given higher transmission opportunity. To avoid extra signaling and control overhead, an active node i has to *self-decide* if it should be selected as a transmitter node by calculating an index number r_i^{TX} as follows:

$$r_i^{TX} = \frac{\bar{p} - p(i)}{\bar{p}} + \gamma_i = P_i + \gamma_i \qquad (15)$$

where γ_i is a uniformly distributed random number with value in the range [0,1], which is generated at a node i at each transmission duration. The random number γ_i is introduced to provide some fairness among nodes, while the factor $\frac{\bar{p}-p(i)}{\bar{p}}$ is used to give a higher priority node the larger probability of transmission. If $r_i^{TX} < P^{TX}$, node i is selected as a transmitter node in the current transmission duration; otherwise, it has no right of transmission. Therefore, a node with higher service level and/or larger load and hence longer delay has higher chance of being selected as a transmitter node, and our selection algorithm supports QoS and load balancing while ensuring certain fairness.

7.2 Stream Allocation

In distributed scheduling, as there is no centralized control mechanism, the stream allocation decision can be made either at the transmitter nodes or at the receiver nodes, and there is a tradeoff for taking either of the options. In this section, we propose a distributed stream allocation algorithm (DSA) which makes decision first at the transmitter nodes, then at the receiver nodes and finalizes the decision at the transmitter nodes to concurrently consider the priority and quality of the streams and constrain the number of transmission streams to be within the decoding capability of the receivers. The algorithm is given below.

DSA: Distributed Stream Allocation Algorithm

At the beginning of each transmission duration, take the following steps in sequence.

(1) Step 1: actions at the transmitter nodes
At this step, a transmitter node selects n_i^0 data packets from its queue. Denote the number of antennas at a transmitter node i as n_i. If the total number of packets in the queue is less than n_i, all of them are selected, i.e., $n_i^0 < n_i$; otherwise, only the n_i packets with the highest priority are selected. The IDs of the target receiver nodes of the selected packets, the value n_i^0, and a training signal are then rotationally broadcasted through each antenna of the transmitter node.

(2) Step 2: actions at the receiver nodes
After a receiver node k decodes the information sent from all the selected transmitter nodes in its neighborhood, it learns the number of streams it may receive in the current duration, N_k^0, including the data streams targeted to itself and the interference streams targeted to other nodes. Assume there are n_k^t transmitter nodes in the one-hop neighborhood of k, we have:

$$N_k^0 = \sum_{j=1}^{n_k^t} n_j^0 \tag{16}$$

At the reply slot, the receiver will broadcast N_k^0 and the maximum number of streams it can decode N_k^{dec} along with a training sequence.

(3) Step 3: actions at the transmitter nodes
Upon the reception of messages from neighboring receiver nodes, a transmitter node estimates the channel coefficients using the training sequence inserted in the messages, and make the final decision for stream allocation based on the receiving stream information at all its neighboring receivers. Denote the number of receiver nodes within the transmission range of a transmitter node i as n_i^r. Each receiver k sends back the total number of streams it may receive, N_k^0, and the maximum number of streams it can decode, N_k^{dec}. In order to ensure all the receiver nodes in its neighborhood to have high probability of meeting degree constraint, node i constrain its number of sending streams to a number n_i^{allo} as follows:

$$n_i^{allo} = n_i^0 \min_{k=1}^{n_i^r} \left(\frac{N_k^{dec}}{N_k^0} \right). \tag{17}$$

Note that the value n_i^{allo} may be a fraction number. Instead of directly calculating n_i^{allo} according to equation 17, in our algorithm, n_i^{allo} is estimated based on the probability that one stream can be allocated, which is:

$$P_i^{allo} = \min_{k=1}^{n_i^r} \left(\frac{N_k^{dec}}{N_k^0} \right) \tag{18}$$

The stream allocation scheme of a selected transmitter node is then as follows.

```
1. Determine the number of streams that can be allowed
   for transmission $n_i^{allo}$.
   Set $n_i^{allo} = 0$;
   for $j = 1 \to n_i^0$
       -Generate a uniformly distributed random variable
       $\beta_j$ in the range[0,1];
       -If $\beta_j \leq P_i^{allo}$, $n_i^{allo} + +$;
   end

2. Allocate streams to antennas. Since node $i$ can transmit up to $n_i^{allo}$ number of streams, it needs to select $n_i^{allo}$ packets among the $n_i^0$ packets selected at step(1) and assign them to the $n_i^{allo}$ best antennas. In order to achieve higher data rate, the selection in this step is solely based on the stream quality. Denote the set of antennas of node $i$ as $\{a_i\}$, and the set of target receiver nodes selected in step(1) as $\{B_i\}$. Therefore, there is a set $S_i$ consisting of all the stream quality factors:
   $S_i = \{(Q(a_i(p), B_i(q))|a_i(p) \in \{a_i\}, B_i(q) \in \{B_i\}\}$
   for $i = 1 \to n_i^{allo}$
       - Find the largest element in $S_i$, denote it as $Q_{max}$, and the corresponding antenna and receiver node as $\{a_{max}$,$B_{max}\}$;
       - Allocate the packet for the receiver $B_{max}$ to the antenna $a_{max}$;
       - Delete $\{(Q(a_{max}, B_i(q))|B_i(q) \in \{B_i\}\}$ from $S_i$; if there is no other packet with $B_{max}$ as the target node, also delete $\{(Q(a_i(p), B_{max})|a_i(p) \in \{a_i\}\}$ from $S_i$;
   end
```

Note that the data packets that cannot be scheduled in this transmission duration will be kept in the transmission queue and scheduled in the next duration. Due to the increase in delay time, the unscheduled packets will have their priority increased, and hence have higher chance of being scheduled.

8. PROTOCOL DESCRIPTION

In order to realize our distributed algorithm, we devise a MAC protocol based on the RTS/CTS mechanism of the IEEE802.11 distributed coordination function (DCF). As mentioned in Section 3, a transmission duration consists of four slots, namely RTS, CTS, DATA and ACK, which have different slot length. The duration of each slot is fixed and long enough for the corresponding messages to complete their tasks. Note that slot synchronization is currently achievable in the IEEE802.11 family of protocols [8]. As a node has to decode multiple control signals from nodes in its neighborhood, a multiple-access scheme is required for multiuser detection. Generally, TDMA and CDMA are two commonly used schemes. In our design, we combined both schemes to facilitate multi-user and multi-antenna access.

(1) RTS transmission

In this phase, nodes which determine themselves to be the selected transmitter nodes as in Section 7.1 broadcast RTSs to receiver nodes in its one-hop neighborhood at the beginning of an RTS slot. An RTS contains the ID of node k, the IDs of node k's targeted set of receiver nodes as selected in step (1) in Section 7.2, and a short training sequence. The training sequence is a fixed pseudo-random code with specific structure for channel estimation at receiver nodes. An RTS is masked by another random code, called ID code, which is assigned to each node according to its node ID. ID Codes for different nodes are almost orthogonal, which means that the cross-correlation of different nodes' codes is close to zero. Such code series can be constructed in a similar way as in CDMA systems, e.g. using OVSF code. The code length is related to the node density of the network. Each node keeps a set of random codes, where the size of the set is large enough to cover the maximum number of nodes in its neighborhood. The assignment of codes can be done in a similar way as [16]. An RTS signal from node i is rotationally transmitted through node i's antennas $1 \sim n_i$, and there is a short notice signal between two antennas' transmissions to separate them.

(2) RTS reception and CTS transmission

In an RTS slot, a receiver node is in listening mode using all its antenna elements. Upon the reception of multiple RTSs, a receiver correlates its received signal with each element in its set of random codes to differentiate training sequence from different transmitter nodes and estimate spatial channels. Then information included in RTSs can be extracted to be used in receiver action as in step(2) of Section 7.2. At a CTS slot, a receiver node k broadcasts a CTS signal masked by the ID code of k, which includes its ID, number of total streams it may receive N_k^0, number of streams it is able to decode N_k^{dec}, and a short training sequence. To inform the transmitter nodes of full channel condition information, a CTS is rotationally transmitted from node k's antennas $1 \sim n_k$, as in the case of RTS. Therefore, each independent spatial channel between a transmitter/receiver pair can be estimated at transmitter nodes.

(3) CTS reception and DATA transmission

In a CTS slot, transmitter nodes are in listening mode. Similar to the case at receivers, a transmitter node has to extract the information included in multiple CTSs. Specifically, as described in Section 7.2, it has to extract N_k^{dec} and N_k^0 from all its neighbor receiver nodes to determine the number of streams allowed for transmission, and estimates all spatial channels to construct the set S_i of stream quality factors, which are used to allocate streams to antennas. After stream allocation is completed, spatial multiplexed data streams are transmitted through the selected antennas in DATA slot.

(4) DATA reception and ACK transmission

In a DATA slot, receiver nodes receive streams from the neighboring transmitter nodes. With channel coefficients estimated in phase (2), streams are decoded using MMSE-SIC as described in Section 5. If a data stream is decoded correctly, the receiver node has to confirm with the transmitter node through ACK broadcast. An ACK thus includes the IDs of the transmitter nodes whose streams have been correctly received and is also masked by the ID code of the receiver.

(5) ACK reception

In an ACK slot, all transmitter nodes are in listening mode. Using channel coefficients estimated in phase (3), a transmitter node extracts information in ACKs and checks whether the streams it transmits in this transmission duration are all received correctly. Correctly received data packets are removed from the queue of the node, and erroneously received or lost data packets remain in the queue, waiting to be scheduled 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.

9. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed algorithms through simulations. We consider an ad hoc network with random topology. Nodes are distributed uniformly over a $1250m \times 1250m$ area. Each node has a transmission range of $250m$. A simulation result is obtained by averaging over several runs of simulations with different seeds.

The distributed multi-user multi-stream scheduling algorithms (DMUMSS) is implemented based on the MAC framework described in Section 8 and the algorithms proposed in Section 7. The centralized multi-user multi-stream scheduling algorithm (CMUMSS) is also implemented, which serves as a benchmark for performance comparison. To demonstrate the benefit of using many-to-many cooperative transmission by fully taking advantage of multiuser diversity in a meshed network and through antenna selection, the performance of our algorithms is compared with corresponding centralized and distributed schemes of Single-User Multi-Stream Scheduling (SUMSS), which is based on conventional multiuser selection. In SUMSS, only one pair of transmitter/receiver nodes is allowed to communicate in the neighborhood, and both transmitter and receiver nodes use all their antenna elements. In each transmission duration, the node pair with the best channel quality is selected, and transmitter node selection is also implemented in SUMSS to reduce collision.

The metrics we use for comparison are aggregate data rate, average drop rate and normalized delay. Aggregate data rate is the total data rates of the network averaged over the number of transmission durations. Packets are dropped due to erroneous decoding when the total number of streams received at a receiver exceeds its decoding capability. The drop rate is defined to be the total number of dropped packets divided by the total number of transmitted packets. Delay time is defined as the number of transmission durations a packet waits in the queue before it is successfully transmitted. The two phases of distributed scheduling, namely Distributed Transmitter Nodes Selection (DTNS) and Distributed Stream Allocation (DSA), are first studied separately; then the overall performance of DMUMSS is evaluated and compared with CMUMSS, centralized SUMSS (CSUMSS) and distributed SUMSS (DSUMSS). If not otherwise specified, the number of nodes in the network is 100,

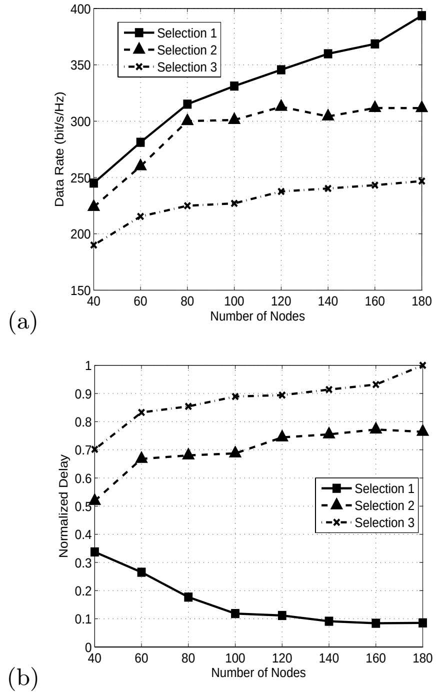

Figure 1: Performance of DTNS: (a) data rate with different types of transmitter nodes selection; (b) normalized delay with different types of transmitter nodes selection.

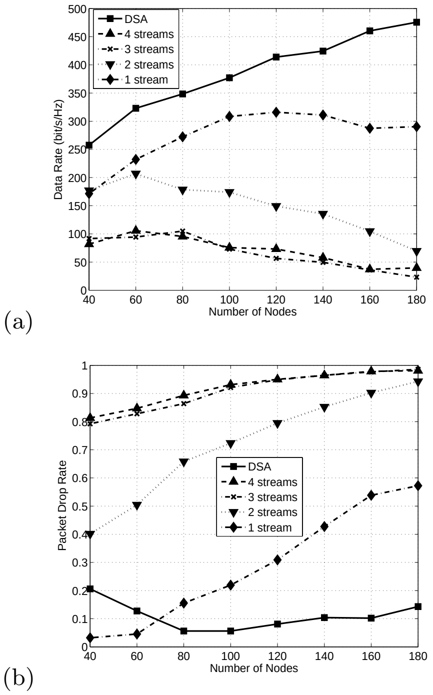

Figure 2: Performance of DSA: (a) data rate with DSA and non-adaptive distributed stream allocation; (b) packet drop rate with DSA and non-adaptive distributed stream allocation.

the number of antenna elements at each node is 4, and the overload factor α defined in Section 4.2 is 0.

(1) Performance of DTNS

We first evaluate the performance of DTNS by varying the node density. We consider three types of distributed transmitter nodes selection:

Selection 1: Use DTNS as described in 7.1;

Selection 2: Use P^{TX} as described in 7.1, but does not consider node priority in r^{TX} calculation;

Selection 3: Use a fixed P^{TX}, which is 0.5 in the simulation, and does not consider node priority in r^{TX} calculation.

Aggregate data rate and normalized delay for the three selection schemes are compared in Figure 1. Selection scheme 3 is seen to have the lowest aggregate rate and the highest normalized delay, as it does not consider node density and load condition in node selection. Scheme 2, although outperforms scheme 3, loses to scheme 1 in both aggregate data rate and normalized delay as it can not fully adapt to the traffic demands of nodes. By considering the active node density and traffic load in a neighborhood to reduce collision and delay, selection scheme 1 is seen to achieve more than 60% higher aggregate rate at the highest node density studied while reducing the delay up to 90%.

(2) Performance of DSA

In Section 7.2, the number of streams allocated is adaptively adjusted according to the traffic condition in the neighborhood. To demonstrate its advantage, we implement an alternative of DSA where the number of streams allocated is fixed. The number of streams is fixed to different values in the simulation. The performance of DSA and the alternative scheme is illustrated in Figure 2. It is evident that by adjusting the number of streams according to traffic condition, DSA outperforms its alternative by providing significantly higher data rate and lower packet drop rate. As the node density increases, data rate for the alternative scheme reduces. Moreover, the rate is lower when the fixed stream is set at a larger number as it leads to severe collisions.

(3) Performance of DMUMSS

The overall performance of DMUMSS is evaluated in Figure 3, with CMUMSS, CSUMSS and DSUMSS as references. According to Figure 3 (a), the aggregate rates of DMUMSS and CMUMSS are close, but the rate of DMUMSS is more than double that of CSUMSS and almost eight times of the rate of DSUMSS. This demonstrates that the data rate can be greatly increased in a meshed network through many-to-many cooperative transmissions by fully exploiting multiuser diversity and spatial diversity. Moreover, as the number of nodes in the network increases, the data rates of both CMUMSS and DMUMSS increase, while the data rate of CSUMSS saturates at a maximum value and the rate of DSUMSS even decreases, as it cannot fully take advantage of the multiuser diversity to achieve higher rate. Figure 3(b) illustrates the changing of data rate with varied number of antenna elements. Again, data rate of MUMSS increases almost linearly. In comparison, limited by the single user constraint, the increasing of data rate of SUMSS slows down as the number of antennas grows. In Section 4.1, we have mentioned overload factor α, which allows more streams to be correctly decoded than the number of antenna elements at receiver nodes. The impact of factor α is studied in Figure 3(c). SUMSS can not take advantage of the higher decoding capability to improve data rate, since only interference-free one-to-one communication is allowed in a neighborhood, and the number of streams transmitted be-

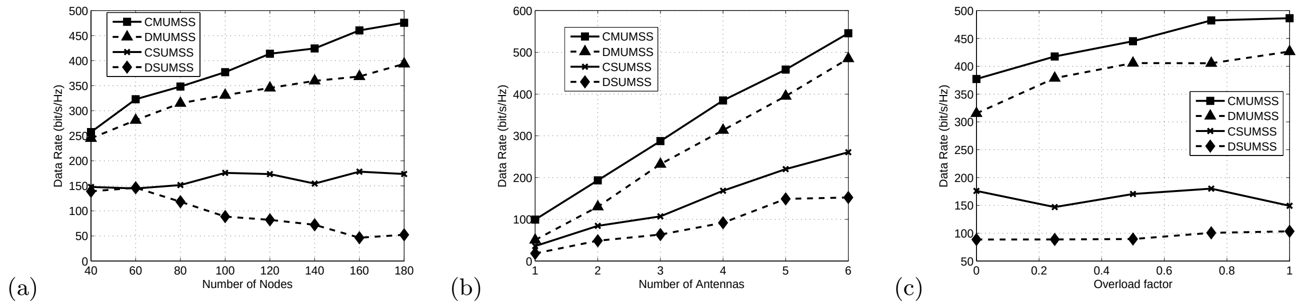

Figure 3: Performance of DMUMSS: (a) data rate with different numbers of nodes in the network; (b) data rate with different numbers of antenna elements; (c) data rate with different values of overload factor.

tween a node pair is constrained by the number of antennas at the transmitter node. Both CMUMSS and DMUMSS achieve higher data rates as overload factor increases from 0 to 1; however, the increasing slope reduces due to the limitation in the number of antennas at transmitter nodes, and the aggregate data rate becomes flat when the overload factor is between 0.75 and 1.

10. CONCLUSIONS

In this work, we propose a centralized and a distributed scheduling scheme in MIMO-based ad hoc networks by concurrently considering traffic demand, service requirements, network load, multiuser diversity, and spatial diversity. Our scheme fully exploits multiuser diversity and spatial diversity to opportunistically select transmitter nodes and transmission antennas 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 physical layer model, and apply the physical model in our MAC design to efficiently optimize network performance. Our performance results demonstrate that our proposed algorithms are very efficient in coordinating transmissions in a MIMO-based MPR network. Up to eight times data rate is achieved as compared to the scheme of selecting only one user pair at a time as often used in cellular networks, while the transmission delay is reduced up to 90%.

11. REFERENCES

[1] IEEE Strandards, *IEEE Standard 802.11n*.

[2] W. Ajib, D. Haccoun, An overview of scheduling algorithms in MIMO-based fourth-generation wireless systems, *IEEE Network*, Volume 19, Issue 5, pp. 43 - 48, Sept.-Oct. 2005.

[3] V.K.N., Lau, Y. Liu, T. A. Chen, Optimal multi-user space time scheduling for wireless communications, *Proc. IEEE VTC 2002-Fall*, Volume 4, pp. 1939- 1942, 2002.

[4] D. Aktas, H. El Gamal, Multiuser scheduling for MIMO wireless systems, *Proc. IEEE VTC 2003-Fall*, Volume 3, pp. 1743 - 174, 2003.

[5] 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*, pp. 317 - 330, 2004.

[6] J-S. Park, A. Nandan, M. Gerla, and H. Lee, SPACE-MAC: enabling spatial-reuse using MIMO channel-aware MAC, *Proc. IEEE ICC 2005*, May 2005.

[7] G. Jakllari, S. V. Krishnamurthy, M. Faloutsos, P. V. Krishnamurthy, O.A Ercetin, Framework for Distributed Spatio-Temporal Communications in Mobile Ad Hoc Networks, *Proc. IEEE INFOCOM 2006*, pp 1 - 13, April 2006.

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

[9] Zorzi, M. *et al.*, Cross-layer issues in MAC protocol design for MIMO ad hoc networks, *IEEE Wireless Communication Magzine*, Volume 13, Issue 4, pp. 62 - 76, Aug. 2006.

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

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

[12] K. Sundaresan and R. Sivakumar, A unified MAC layer framework for ad-hoc networks with smart antennas, *Proc. ACM MobiHoc 2004*, pp. 244 - 255, 2004.

[13] M. Park, R. Heath, and S. Nettles, Improving Throughput and Fairness of MIMO Ad hoc Networks using Antenna Selection Diversity, *Proc. IEEE Globecom*, Nov. 2004.

[14] L. Bao and J.J. Garcia-Luna-Aceves, Distributed Channel Access Scheduling for Ad Hoc Networks, *Algorithms and Protocols for Wireless and Mobile Networks*, CRC/Hall Publisher, 2004.

[15] A. Ephremides and T.V. Truong, Scheduling broadcasts in multihop radio networks, *IEEE Transactions on Communications*, 38(4): 456 - 460, Apr 1990.

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

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

[18] B. Vucetic and J. Yuan, *Space-Time Coding*, New York: Wiley, 2003.