Opportunistic and Cooperative Spatial Multiplexing in MIMO Ad hoc Networks

Shan Chu, *Student Member, IEEE,* and Xin Wang, *Member, IEEE*

***Abstract*—With the fast progress of MIMO technology and its growing applications in networks, it is important to develop techniques to enable more efficient MIMO network communications. However, it is very challenging to coordinate node transmissions in a MIMO-based ad hoc network. In this work, we propose schemes that take advantage of the meshed topology of ad hoc networks to fully exploit the multiuser diversity and spatial diversity, in order to maximize the data rate of the network, while supporting different transmission priorities, reducing transmission delay and ensuring fair transmissions among nodes. We formulate a concrete physical model, and present cross-layer centralized and distributed scheduling algorithms which exploit physical layer channel information to opportunistically schedule cooperative spatial multiplexed transmissions between nodes. We also propose a new MAC scheme to support our distributed algorithm. The performance of our algorithms are studied through extensive simulations and the results demonstrate that our algorithms are very effective and can significantly increase the network throughput while reducing the transmission delay.**

***Index Terms*—Spatial multiplexing, MIMO, ad hoc networks, scheduling, cross-layer design.**

I. 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 users 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]–[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, where all nodes are peer-to-peer in nature and connected through a mesh topology. Different from an infrastructure-based single-hop cellular network, it is difficult for nodes to coordinate in channel evaluations and transmissions in a dynamic meshed 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 [6]–[9], [11], [13], [14], 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 lead 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 demands, 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 antenna sets are selected opportunistically to exploit the multiuser diversity and spatial diversity to a large 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 a graph approach, and divide it into two subproblems.
- 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.

Shan Chu and Xin Wang are both with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY, 11794 USA. E-mail: ({schu,xwang}@ece.sunysb.edu). This research was supported by U.S. NSF under grant numbers CNS-0751121 and CNS-0628093.

- We develop schemes to specifically consider the service requirements of the user traffic, the transmission delay, and the fairness among nodes.
- 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 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 II discusses the related work. We introduce the system architecture in Section III and describe our physical model in Section IV. We formally formulate the problem in Section V, and propose our centralized and distributed algorithms in Section VI and VII respectively. In Section VIII, we present our MAC protocol. Simulation results are given in Section IX and the paper is concluded in Section X.

II. 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 [6], spatial diversity (e.g. space time coding (STC)) is explored to combat fading and achieve robustness. SPACE-MAC, proposed in [7], enables denser spatial reuse patterns with the aid of transmitter and receiver beamforming. Authors in [8] 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 [9]. A high-level discussion about cross-layer issues in MAC protocols design for MIMO ad hoc networks is further presented in [10]. In [11], spatial multiplexing with antenna subset selection for data packet transmission is proposed. In [8], [9], [11], 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 [12], the physical layer approximation is studied to facilitate cross-layer design of MIMO-BLAST ad hoc network. However, it does not provide a complete algorithm/protocol that can be actually implemented. In [13], 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, while these parameters are critical for MIMO network design and challenging to gain 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 [14]. In these literature works, spatial diversity and multiuser diversity are not fully exploited. There are no support of QoS and consideration of the difference in node traffic demands. We have made an effort to address some of these issues in [15]. In this paper, we more clearly formulate the problem based on network graph, and further design the distributed algorithm to better support packet transmission priority and user service requirements. We also perform more extensive simulations to demonstrate the functionality of the proposed algorithms.

III. 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 data at a 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.

Spatial diversity can be adopted to further improve the transmission gain thus reliability and capacity. 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 could be selected for transmissions through selection diversity, which is shown to outperform space-time coding in [16]. In this work, antenna selection diversity is exploited at a node to select a subset of stronger streams for transmission. In addition, the proposed many-to-many transmission with use of virtual MIMO array also helps to select stronger streams from candidate transmission node pairs, taking advantage of multi-user diversity to provide additional reliability and throughput. As we focus on spatial multiplexing instead of topology control in this paper, spatial diversity is only used for diversity gain and transmission range is assumed to be uniform.

A stream is identified by a triplet $(I_{TX}, I_{RX}, I_{ANT})$, where I_{TX} is the index of the transmitter node, I_{RX} is the index of the receiver node, and I_{ANT} is the index of the transmitter antenna. 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 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 [20].

As it is difficult 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 node selection* and *stream allocation*. 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 to appropriate antennas.

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 a node which depends on both the service type and the delay time of its queued 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. In this phase, a selected transmitter node first determines a set of packets to transmit based on their priority and the allowed number of streams to transmit in the neighborhood. As discussed later in Section IV, multiple antennas at both ends of a link create multiple independent spatial channels with different channel gains in a multi-path or rich scattering environment, which makes channel capacities or achievable data rates of the streams different. It is thus beneficial to allocate the selected packets to transmit over channels that have stronger channel gain thus higher data rate, i.e. with *opportunistic stream allocation*, in order to maximize the temporal throughput of the network.

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 the priority of the stream p is higher. In other words, the stream whose data packet has a higher service priority and/or experiences a longer delay is given a higher priority for transmission.

. **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).

IV. Physical Model

In wireless communications, time-varying fading is commonly observed due to user mobility or the variation of propagation environments [20]. 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 the transmission ranges of each other, and the numbers of antenna elements are n_i and n_k respectively. The spatial channel between i and k can be represented as an $n_k \times n_i$ matrix $\mathbf{H}_{ki}$:

$$\mathbf{H}_{ki} = \begin{pmatrix} h_{11} & h_{12} & \dots & h_{1n_i} \\ h_{21} & h_{22} & \dots & h_{2n_i} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n_k1} & h_{n_k2} & \dots & h_{n_kn_i} \end{pmatrix}, \tag{2}$$

where h_{mn} is the spatial channel coefficient between the m-th antenna of node k and n-th antenna of node i, and can be represented as in (1). In general cases, the number of independent eigenchannels [20] between i and k is equal to the number of non-zero eigenvalues of the matrix $\mathbf{H}_{ki}^*\mathbf{H}_{ki}$. In a rich scattering environment and if the separation of antenna elements at each node is large enough, the spatial channels between node i and k undergo i.i.d fading and there are $\min\{n_i, n_k\}$ eigenchannels in total. For the convenience of discussion, we assume the rich scattering environment all through this paper, and our results can be easily extended to scenarios with less scattering, i.e. with LOS, by calculating the actual number of eigenchannels.

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

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

where $s_1, s_2, \dots, s_{n_i}$ are signals transmitted from antenna $1, 2, \dots, n_i$. Note that $s_1, s_2, \dots, s_{n_i}$ 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:

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

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

$$= \mathbf{r}_{ki,sig} + \mathbf{r}_{ki,int}, \qquad (4)$$

where X_{ki} is the set of streams from 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 (SINR) of received stream p at node k as $SINR_p$, the sum data rate that receiver node k gets from transmitter node i is:

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

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:

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

The calculation of $SINR_p$ depends on the decoding capacity at the receiver node. According to [20], 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:

$$\mathbf{v}_p = \mathbf{K}_{\mathbf{z}_p}^{-1}\mathbf{h}_p. \qquad (7)$$

The corresponding SINR achieved is

$$SINR_p = \sigma_p^2 \mathbf{h}_p^* \mathbf{K}_{\mathbf{z}_p}^{-1}\mathbf{h}_p, \qquad (8)$$

where $\mathbf{h}_p$ is the $n_k \times 1$ channel vector for stream p to a receiver k with n_k antennas, $\mathbf{K}_{\mathbf{z}_p}$ is the covariance of $\mathbf{z}_p$, which is the noise plus interference faced by data stream p: $\mathbf{z}_p = \sum_{q>p}^{N_k} \mathbf{h}_q s_q + \mathbf{n}$. Here N_k is the number of transmission streams (including both data and interference streams) around the receiver k. In SIC decoding, received streams are initially sorted according to their received strength, and the strongest stream is first recovered and subtracted from the received vector. Therefore, only the weaker streams create interference at a stream p. Although the quality of SIC decoding may be impacted by error propagation and the accuracy of channel estimation, it works well if the streams are well coded and the data block length is large [20]. As the design of receiver structure is beyond the scope of this paper, we do not deal with the problems due to channel estimation and decoding errors.

In point-to-point transmissions, when channel information is known, a transmitter node can assign different power to different transmission streams based on their channel conditions using water-filling method [20] to maximize the data rate. As described in Section VI, our centralized algorithm schedules a stream for transmission by comparing its priority and channel gain with those of other candidate streams in the network, thus the streams to select from a node are not known in a scheduling step, and water-filling could not be easily applied to divide the total power among multiple streams in advance. However, water-filling can be used to assign transmission power in our distributed algorithm proposed in Section VII where the determination of the number of streams to use and the allocation of streams are decoupled. As the transmissions of multiple streams from one transmitter node would lead to lower channel gain for individual streams, in many-to-many transmission scenario, it may help to schedule transmissions from multiple nodes than transmitting multiple streams from the same node given the same degree of freedom constraints. Therefore, there is a lower likelihood for a node to transmit multiple streams and the need of power splitting among streams using water-filling. For better performance comparison between our centralized algorithm and distributed algorithm, we consider equal power allocation in this paper. The performance of our distributed scheduling would be further improved without much change to the algorithm if water-filling is used, but extra processing complexity is required for power assignment considering multiple channel matrices for transmissions to multiple receiver nodes.

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_q associated with data stream q and N_0 as the noise variance, we can explicitly calculate $\mathbf{K}_{\mathbf{z}_p}$ as

$$\mathbf{K}_{\mathbf{z}_p} = N_0\mathbf{I}_{n_r} + \sum_{q>p}^{N_k} P_q \mathbf{h}_q \mathbf{h}_q^*, \qquad (9)$$

which is invertible. Note that in order to avoid significant signaling overhead, nodes are assumed to perform channel estimation through communications with their one-hop neighbors using MAC protocol in Section VIII. As nodes are only able to estimate the channels between themselves and nodes in their receiving range, the signals coming from non-estimated channels may constitute a noise floor. Moreover, the channel estimation capacity of a node is always limited in any channel access strategy. The noise floor could potentially reduce the achievable receiving rate, as does in any transmission scheme. However, our MAC design exploits multi-user diversity and antenna selection diversity to significantly increase the transmission signal strength, which helps to increase the received SINR and thus mitigate the problem due to noise floor. Substitute (9) into (8), the output SINR for stream p can be calculated as:

$$SINR_p = P_p \mathbf{h}_p^* \left(N_0 \mathbf{I}_{N_r} + \sum_{q>p}^{N_k} P_q \mathbf{h}_q \mathbf{h}_q^* \right)^{-1} \mathbf{h}_p. \qquad (10)$$

Substitute (10) into (6), we can calculate the data rate for each receiver node. Therefore, the aggregate data rate of the network is $R = \sum_{k \in S_r} R_k$, 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 $\|\mathbf{h}_p\|^2 = \mathbf{h}_p^* \mathbf{h}_p$ is, the higher 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:

$$Q(p)_{sim} = P_p(\mathbf{h}_p^* \mathbf{h}_p) = P_p \|\mathbf{h}_p\|^2. \qquad (11)$$

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, it is very difficult to estimate the complete interference formation before scheduling is performed. Thus we define a normalized stream quality index to capture the interference a stream creates to its neighbors hence the impact of interference streams on scheduling:

$$Q(p)_{nor} = \frac{P_p \|\mathbf{h}_p\|^2}{\sum_{q \in X_{int}} P_q \|\mathbf{h}_q\|^2} = \frac{\|\mathbf{h}_p\|^2}{\sum_{q \in X_{int}} \|\mathbf{h}_q\|^2}, \qquad (12)$$

where X_{int} is the set of interference streams towards neighboring active receivers caused by the transmission of data stream p. By normalizing the strength of a data stream with the strength of interference stream(s) it results in, streams that have higher channel gain yet cause smaller interference in the neighborhood are preferred during scheduling. The definition of stream quality is then used in the following sections for stream allocation.

So far, we have formulated a concrete physical model and provided a stream quality metric to facilitate scheduling. Although the above analysis is based on MMSE-SIC receiver, which helps investigate the impact of physical layer parameters on network performance, the scheduling algorithms we propose next do not depend on a specific receiver model. Other receiver strategies can be easily adopted using our algorithms without much modification.

V. Problem Formulation

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

A. Graph Construction

A directed graph $G = (V, E)$ is used to model the topology and 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. Specifically, the source/destination vertex of an edge is the transmitter/receiver of the corresponding stream, solid edges represent data streams, and dashed edges represent interference streams. At the beginning of a transmission duration, if node i has one packet targeted for node k, there is a candidate transmission stream from i to k. A solid 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 dashed edges are formulated from i to $j_1, \ldots, j_n$. If node i has n_i packets for transmission, there are a total of n_i solid edges originated from i to some of its neighbor nodes.

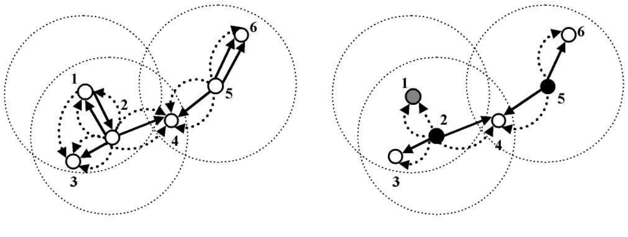


Fig. 1. Examples: (a) graph representation of the network; (b) feasible scheduling.

Figure 1 (a) shows an example of the graph construction. In the figure, a dotted circle represents the transmission range of the centered node. The network consists of six nodes. As node 2 has a packet for each of the nodes 1, 3 and 4, there is one solid edge between node 2 and each of these nodes. A solid edge from node 2 to a target receiver node(e.g., node 1) is accompanied by a set of dashed edges to other neighbors (e.g., nodes 3 and 4). Therefore, nodes 1, 3, and 4 each has two incoming dashed edges from node 2 as a result of the three solid edges originated from node 2. A node may have multiple incoming solid edges from the same neighbor node which has more than one packet for it. For example, node 6 in the figure has two incoming solid edges from node 5, accompanied by two dashed edges from node 5 to node 4.

Edges are scheduled in sets. Each set $\{e\}_j$ consists of one solid edge $e_j = (s(j), t(j))$ and $N_j - 1$ dashed edges $\{\bar{e}\}_j$, where $s(j)$ / $t(j)$ are the source/destination vertices of e_j and N_j is the number of nodes within the transmission range of transmitter node $s(j)$.

The stream parameters defined in Section III thus become the parameters of edges. The stream priority depends on the data packet, thus one solid edge and its corresponding dashed 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 III. It is obvious that e_i and $\{\bar{e}\}_i$ have different stream quality as they are associated with different stream triplets. 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.

In 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 solid edges connected to a vertex i as $d_i(out)$, the number of incoming solid edges connected to a vertex i as $d_i(in.data)$, the number of incoming dashed 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 Constraints (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 [21], 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.

Figure 1 (b) shows an example of the feasible scheduling, where the degree constraint is satisfied for every node. In the figure, nodes 2 and 5, which have non-zero number of outgoing edges, are colored black and scheduled as transmitter nodes. Nodes 3, 4 and 6, with zero outgoing edges and non-zero number of solid incoming edges, are colored white and serve as receiver nodes. Node 1 has neither outgoing edge nor solid incoming edge, so it is colored grey as an idle node. Assume $n_i = 4, i = 1, \ldots, 6$ and $\alpha = 0$ here, then all the receiver nodes satisfy the third inequation of the constraint (2), which indicates that the data streams can be correctly decoded. Node 4, which has 4 incoming edges, is an example of being a fully loaded receiver node.

B. 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. Rather than scheduling transmissions over time dimension, it is more practical to model the scheduling problem as an iterative optimum subgraph selection problem in each transmission duration (TD), where the temporal network performance is optimized. To consider interactions between consecutive TDs, the residual graph of each TD is updated and left to the next TD for processing. The stream priority metric is calculated accumulatively based on increasing delay time and the stream quality metric is updated according to the channel variation. 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 solid 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 constraints. Second, the higher priority streams are preferably 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
```

In many cases, the channel associated with the transmitter and receiver of a packet with the highest priority may not have the best quality. There is a tradeoff between optimizing priority and optimizing capacity. In our problem, the optimality of priority is satisfied before the optimality of capacity condition is checked in order to assure the transmissions of high-priority streams first.

Our scheme is TDMA based by scheduling transmissions in each transmission time duration. Although promising [17], the application of TDMA in ad-hoc networks leads to the known NP-complete Broadcast Scheduling Problem (BSP) [18]. Therefore, we will provide suboptimal solution with our centralized and distributed scheduling algorithms next.

VI. Centralized Algorithm

In this section, we propose a centralized algorithm (CMUMSS) 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 provides a basis for the distributed algorithm.

In the algorithm, directed graphs are formed as described in Section V-A with each data edge associated with a candidate transmission between a transmitter and a receiver. The scheduling algorithm ranks all the packets in the system (in implementation, only the priority of the head of line packets of different nodes need to be compared) and greedily schedules transmissions from higher priority to lower priority. For packets with the same priority, transmissions are scheduled from the higher channel quality to lower quality, while ensuring that the overall scheduled network transmissions satisfy the degree constraints. The scheduling therefore meets the constraints (1) and (2) of OSGP formulated in Section V-B and provides performance with a fixed approximation ratio in terms of constraint (3). The centralized algorithm is given as below.

CMUMSS: Centralized MUMSS Algorithm

1. Initialization

The central controller checks the queue of data packets at every node, and constructs a graph G_0 according to Section V-A. G_0 contains the data edges and interference edges to be scheduled at the initial phase and is updated to form graph G_k in a subsequent transmission duration k $(k = 1, 2, \ldots)$.

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 solid 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 queuing delay of corresponding packets. The updated graph is denoted as G_k. Let the optimum subgraph $G_k^{opt} = NULL$. Create another subgraph called blocked graph G_k^b, used to save edge sets that cannot be scheduled in the current duration, and set $G_k^b = NULL$. Each node is allowed either to be a transmitter node or a receiver node at this stage.

II. Stream Allocation

Select the 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 solid edge associated with a candidate data transmission denoted as $e_j = (s(j), t(j))$, where $s(j)$ and $t(j)$ are the source and destination vertices of edge e_j respectively. e_j and its corresponding dashed edges form a set $\{e\}_j$. Sort all the elements in $\{e\}^h$ according to their priority. The set $\{e\}^h$ can then be partitioned into a series of subsets $\{e\}_1^h, \{e\}_2^h, \ldots, \{e\}_{N_{prio}}^h$, where N_{prio} is the number of different priority values in $\{e\}^h$ and elements in the set $\{e\}_l^h$ are edges in $\{e\}^h$ that have the same priority P_l.

```
for $l = 1 \rightarrow N_{prio}$
  - Denote the $q$-th element $e_q$ in $\{e\}_l^h$ along with its dashed edges as $\{e\}_q$. For an edge $e_q$, data transmission can be scheduled from any of the unassigned antennas of its transmitter and the scheduler assigns the antenna based on the channel quality. Construct a set consisting of the channel quality factors associated with all the possible stream allocations for solid edges with priority $P_l$: $S_l = \{Q(a,B) | e_q = (s(q), t(q)), a \in A_{s(q)}, B = t(q), \forall e_q \in \{e\}_l^h\}$ where $A_{s(q)}$ is the set of unused antennas at $s(q)$, $Q(a,B)$ is the stream quality factor for a stream between antenna $a$ and node $B$;
  - for $q = 1 \rightarrow |\{e\}_l^h|$
    – Find the largest element in $S_l$, denote it as $Q_{max}$, and the corresponding transmitter node, receiver node and antenna as $s_{max}$, $t_{max}$ and $a_{max}$ respectively. $e_{max} = (s_{max}, t_{max})$ is the corresponding edge of $Q_{max}$ that has $s_{max}/t_{max}$ as its source/destination node.
    – If $s_{max}$ is marked as a receiver node or $t_{max}$ is marked as a transmitter node from previous scheduling steps, the edge $e_{max}$ is not eligible for scheduling. Remove the set $\{e\}_{max}$ containing $e_{max}$ and its corresponding interference edges from $G_k$ and add it to $G_k^b$;
    – Else:
      Tentatively add $\{e\}_{max}$ to $G_k^{opt}$. Check whether DC is still satisfied for $G_k^{opt}$.
      . If no, remove $\{e\}_{max}$ from $G_k$ and add it to $G_k^b$;
      . Else, the edge $e_{max}$ is eligible for scheduling. Mark $s_{max}$ as a transmitter node and $t_{max}$ as a receiver nodes if they are not currently marked. Assign $e_{max}$ to the antenna $a_{max}$, add $\{e\}_{max}$ along with the allocation information to $G_k^{opt}$. Update $A_{s_{max}}$ to remove $a_{max}$ from the unused antenna set. Meanwhile, if any vertex associated with $\{e\}_{max}$ becomes fully loaded, remove all edge sets that may overload it from $G_k$ and add them to $G_k^b$. Remove elements associated with $a_{max}$ from $S_l$.
    – Delete $Q_{max}$ from $S_l$.
  - 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_k^{opt}. Add the edges in G_k^b back to G_k, which will be used for scheduling in the next transmission duration.

Next, we use the example in Figure 1 to explain our CMUMSS algorithm. In a specific transmission duration k, the graph G_k is constructed as in Figure 1(a). Assume the data edges (solid edges) in the figure, from left to right, have index numbers and priorities as in the following table.

data edge index	1	2	3	4	5	6	7
source vertex	2	2	1	2	5	5	5
destination vertex	3	1	2	4	4	6	6
stream priority	5	2	1	3	4	3	1

Initially, $\{e\}^h$ consists of edges e_1, e_3, e_5, which are respectively the highest priority edge from candidate transmitter nodes 2, 1, and 5. As e_1 has the highest priority 5, it is scheduled first, with node 2 and 3 identified as transmitter and receiver nodes respectively. Similarly, e_5 with priority 4 is scheduled next, with node 5 and 4 assigned as transmitter and receiver nodes. When scheduling e_3 in $\{e\}^h$, which has the lowest priority, as its destination node 2 has already been scheduled as a transmitter (when scheduling e_1), e_3 cannot be scheduled for transmission any more in duration k, and is deleted from G_k and added to G_k^b. In the second run of stream allocation, $\{e\}^h$ consists of edges e_4 and e_6, which have the same priority 3. The stream quality set S_1 is then constructed based on the two data edges and the available antennas at their source nodes. Assume e_4 is the edge corresponding to the largest element Q_{max} in S_1, it is then scheduled first. As the addition of e_6 into G_k^{opt} does not violate the DC, it is also scheduled. At this moment, node 4 has two incoming data edges and two incoming interference edges and is fully loaded. Therefore, the rest two edges e_2 and e_7 are removed from the candidate scheduling set G_k and added to G_k^b as it would overload node 4 if they are scheduled for transmissions. Finally, the optimum subgraph G_k^{opt} for the current transmission duration is formulated as in Figure 1(b).

As stated at the beginning of this section, the CMUMSS algorithm is optimum to the first two constraints of OSGP problem defined in Section V-B. Suppose the stream quality factor $Q(p)$ directly reflects the value of stream capacity $C(p)$, we can further prove that CMUMSS achieves a fixed approximation ratio compared with the optimum solution that obtains the highest aggregate data rate.

Theorem: The CMUMSS algorithm can achieve an approximation ratio of of $1/\left((2+\alpha+\mathcal{D})\max_i\{n_i\}+2\right)$ in terms of constraint (3) of OSGP, where $\mathcal{D}$ is the maximum node degree in the network.

Proof: Let *sol* be our solution, and *opt* be the optimum solution that satisfies the degree constraints. Obviously, in *sol*, some of the candidate data edges are suppressed by the selection of a specific edge set $\{e\}$ (i.e., removed from S_l) due to their conflicting with transmission of $\{e\}$ but these streams may be the selected streams in *opt*. According to the selection process, the selection of $\{e\}_{max}$ eliminates the possibility for s_{max}/t_{max} to be receiver/transmitter as well as for any other outgoing data edge of s_{max} to use the antenna a_{max}. Moreover, the assignment of transmitter/receiver eliminates their opportunity of being an idle node, while an idle node does not constrain the number of streams it perceives in the neighborhood. Denote the maximum node degree in the network as $\mathcal{D}$, the number of suppressed streams due to this reason should be no more than $\mathcal{D}\max_i\{n_i\}$, where $\max_i\{n_i\}$ is the maximum antenna array size of nodes in the network. As a result, the number of suppressed data streams that may be transmitted for each priority level in a TD should be no more than $(2+\alpha+\mathcal{D})\max_i\{n_i\}+1$. A data edge $e' \in$*opt* is considered to be associated with a data edge $e'' \in$*sol* either because they are identical or because e' is suppressed by e'' during the process of greedy selection. For each data edge e_q in *sol*, there is a set W_q containing the data edges in *opt* that are associated with it, and $\bigcup_{e_q \in sol} W_q = opt$. The number of streams in W_q, $|W_q|$, has an upper limit $(2+\alpha)\max_i\{n_i\}+2$. As the selection of data edge in *sol* is greedy and looks for the one with the largest stream quality at a time, thus $C(e_q) \geq C(e_m), \forall e_m \in W_q$. Considering the utility function U as the total data rate, we have:

$$\begin{aligned}\frac{U(sol)}{U(opt)} &= \frac{\sum_{e_q \in sol} C(e_q)}{\sum_{e_q \in opt} C(e_q)} = \frac{\sum_{e_q \in sol} C(e_q)}{\sum_{e_q \in sol}\sum_{e_m \in W_q} C(e_m)} \\ &\geq \frac{\sum_{e_q \in sol} C(e_q)}{\sum_{e_q \in sol}\sum_{e_m \in W_q} C(e_q)} = \frac{\sum_{e_q \in sol} C(e_q)}{\sum_{e_q \in sol} |W_q| C(e_q)} \\ &\geq \frac{\sum_{e_q \in sol} C(e_q)}{((2+\alpha+\mathcal{D})\max_i\{n_i\}+2)\sum_{e_q \in sol} C(e_q)} \\ &= \frac{1}{(2+\alpha+\mathcal{D})\max_i\{n_i\}+2}. \square\end{aligned}$$

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

VII. Distributed Algorithm

As introduced in Section 3, the scheduling algorithm includes two phases, namely transmitter nodes selection and stream allocation. The two phases are obviously dependent on each other. 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. For better stream selection, the channel condition between the selected transmitter nodes and their target receiver nodes need to be evaluated. To avoid transmission collision from the selected set of transmitting nodes, the channel measurement signals are encoded using pseudo-random codes as discussed in Section VIII. 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).

A. Transmitter Nodes Selection

As the transmission is half-duplex and a node cannot be a transmitter and a receiver at the same time, there is a need to select a subset of the nodes to serve as transmitters 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. By reducing the transmission delay of each node, DTNS can balance the load in a neighborhood and ensure transmission fairness. In addition, adaptively selecting a subset of nodes in a neighborhood to participate in channel estimations based on the decoding capabilities of nodes in the neighborhood would help reduce the estimation complexity and avoid unnecessary channel estimations.

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:

$$P^{TX} = \min_{j=1}^{n_i^n} \left(\frac{N_j^{dec}}{n_j^a} \right) \tag{13}$$

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:

$$r_i^{TX} = \frac{\bar{p} - p(i)}{\bar{p}} + \gamma_i = P_i + \gamma_i \tag{14}$$

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.

The distributed transmitter nodes selection algorithm is therefore summarized as below.

```
DTNS: Distributed Transmitter Nodes Selection
for each node $i$:
    - Calculate $P_i^{TX}$ based on Eq. 13;
    - Calculate $r_i^{TX}$ based on Eq. 14;
    - If $r_i^{TX} \leq P_i^{TX}$, $i$ determines itself to be a transmitter
      node.
end
```

B. Stream Allocation

In the distributed allocation algorithm, we first assume that nodes can receive RTSs/CTSs from multiple transmitter/receiver nodes simultaneously and decode them correctly if the number of simultaneous RTSs/CTSs is less than a certain limit number. The feasibility of this assumption will be discussed in Section VIII.

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. However, there is a tradeoff for taking either of the options. If the decisions are made at the transmitter nodes, channel information should be made available at the transmitter side first. A transmitter node can properly allocate streams to transmit antennas through pre-coding and cancel the interference partially. However, if all the transmitter nodes make the stream allocation independently, it is very likely that the total number of streams (including data streams and interference streams) arriving at a receiver node exceeds the node's decoding capability. If the decisions are made at the receivers, as a receiver node has full knowledge of all data and interference streams it will receive, it can better select the set of streams to turn off so as to maximize the throughput locally. The disadvantage is that different receivers may decide to turn off different streams and lead to conflicting decisions, so extra coordination is still needed at transmitter nodes to finalize the decision. Additionally, the cost for feeding back the selected stream set is much higher compared with feeding back only a small number of relevant parameters, i.e. each receiver only has to feed back two parameters in our scheme.

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 (based on the channel estimation from the reverse direction) 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. In each transmission duration, the DSA takes the following steps in sequence.

(1) Step 1: actions at the transmitter nodes

At this step, a transmitter node i 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:

$$N_k^0 = \sum_{j=1}^{n_k^t} n_j^0 \tag{15}$$

In the reply slot, if a node is the target receiver of any data stream, it 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 constrains its number of sending streams to a number $n_i^{allo} = n_i^0 \min_{k=1}^{n_i^r} \left(N_k^{dec}/N_k^0\right)$. The value n_i^{allo} may be a fraction number. Instead of directly calculating n_i^{allo}, in our algorithm, n_i^{allo} is estimated based on the probability that one stream can be allocated, which is: $P_i^{allo} = \min_{k=1}^{n_i^r} \left(N_k^{dec}/N_k^0\right)$. 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}.

```
Initialize: 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.

The selection gives preference to packets with higher priority. For packets of the same priority, the selection is solely based on the stream quality in order to achieve a higher data rate. Denote the set of antennas that node i has as $\{a_i\}$, the set of priority levels of the n_i^0 packets as $\{P_i\}$, and the set of receiver nodes which the n_i^0 preselected packets are targeted for as $\{B_i\}$. The set $\{B_i\}$ is partitioned into subsets $\{B_i^1\}, \{B_i^2\}, \ldots, \{B_i^{|\{P_i\}|}\}$ according to the descending priorities of the packets. The j-th subset $\{B_i^j\}$ contains the target receiver nodes of the packets with priority $P_i(j)$.

Recall that a stream p is identified by its transmitter node, transmitter antenna and receiver node, and each stream p has a unique stream quality parameter $Q(p)$, which depends on the transmission power and channel condition of the specific spatial channel. If the normalized stream quality parameter defined in (12) is used here, X_{int} only includes the interference streams towards the active receivers in the neighborhood, i.e. those that have sent back CTSs but are not the targeted receivers of stream p. For transmitter node i, there is a set S_i^0 consisting of all the stream quality parameters of the candidate streams:

$S_i^0 = \{Q(a_i(p), B_i(q)) | a_i(p) \in \{a_i\}, B_i(q) \in \{B_i\}, p = 1, \ldots, |\{a_i\}|, q = 1, \ldots, |\{B_i\}|\}$

```
Subroutine: stream_allocation ($\{a_i'\}, \{B_i^j\}, k$)
Initialize: $l = 0$;
$S_i^j = \{Q(a_i'(p), B_i^j(q)) | a_i'(p) \in \{a_i'\}, B_i^j(q) \in \{B_i^j\}, p = 1, \ldots, |\{a_i'\}|, q = 1, \ldots, |\{B_i^j\}|\}$;
while $l < k$
    - Find the largest element in $S_i^j$, 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 $a_{max}$;
    - Remove $\{Q(a_{max}, B_i^j(q)) | B_i^j(q) \in \{B_i^j\}, q = 1, \ldots, |\{B_i^j\}|\}$ from $S_i^j$, as $a_{max}$ is no longer available; if there is no other packet target for the receiver node $B_{max}$, also remove $\{Q(a_i'(p), B_{max}) | a_i'(p) \in \{a_i'\}, p = 1, \ldots, |\{a_i'\}|\}$ from $S_i^j$;
    - Remove $a_{max}$ from $\{a_i'\}$;
    - $l + +$;
end
```

Assume $\{a_i'\}$ contain the set of available antennas of node i that can be used for stream allocation, the set S_i^j contain the quality parameters of the streams formulated between the antennas in $\{a_i'\}$ and the receivers in the set $\{B_i^j\}$. Let l represent the number of streams currently allocated. The subroutine *steam_alloation* is used to allocate k streams to transmit the packets which are targeted for the receivers in the priority set $\{B_i^j\}$. Note that $\{a_i'\}$, the set of available antennas of node i, is updated as the subroutine is executed. Let j be the index of the priority level and n_a be the number of streams that have been allocated. Based on the subroutine *stream_allocation*, n_i^{allo} streams can be allocated to appropriate antennas in a loop as below.

```
Initialize: $j = 1$, $n_a = 0$, $\{a'_i\} = \{a_i\}$;
while $n_a < n_i^{allo}$
    - If $|\{B_i^j\}| \leq n_i^{allo} - n_a$, do stream_allocation ($\{a'_i\}, \{B_i^j\}, |\{B_i^j\}|$), $n_a = n_a + |\{B_i^j\}|$;
    - else, do stream_allocation ($\{a'_i\}, \{B_i^j\}, n_i^{allo} - n_a$), $n_a = n_i^{allo}$;
    - $j++$;
end
```

The data packets that cannot be scheduled in the current transmission duration will be kept in the transmission queue and wait to be 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.

VIII. 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 III, a transmission duration (TD) consists of several time slots and covers one round of control signal exchange and fixed-size data frame transmission. Follow the paradigm of IEEE802.11, a TD consists of four slots, namely RTS, CTS, DATA and ACK, which have different slot lengths. 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 [9]. Although distributed transmissions may increase the asynchronicity at the symbol level and impact decoding quality, as our scheme could effectively increase the SINR of received signals by taking advantage of the antenna selection diversity and multi-user diversity, it would help improve the accuracy of synchronization as well as mitigate the impact of asynchronicity in a distributed scenario. 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. The protocol consists of the following five phases.

(1) RTS transmission

In this phase, nodes which determine themselves to be the selected transmitter nodes as in Section VII-A 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 and the IDs of node k's targeted set of receiver nodes selected by step (1) in Section VII-B. The preamble of an RTS can be used as a training sequence for channel estimation at the 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. Recall that we assume the neighbor density is limited to ensure the possibility of channel estimations and hence decoding performance. 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 [19]. An RTS signal from node i is rotationally transmitted through node i's antennas $1 \sim n_i$, and there are 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 sequences 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 VII-B. In a CTS slot, a node k that is the targeted receiver in any RTS request broadcasts a CTS signal masked by the ID code of k, which includes its ID, the number of total streams it may receive N_k^0, and the number of streams it is able to decode N_k^{dec}. Similarly, the preamble of CTSs can be used for training and channel estimation purpose. 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 step (3) of Section VII-B, 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 parameters, which are used to allocate streams to antennas. After stream allocation is completed, spatial multiplexed data streams are transmitted through the selected antennas in a 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 IV. 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.

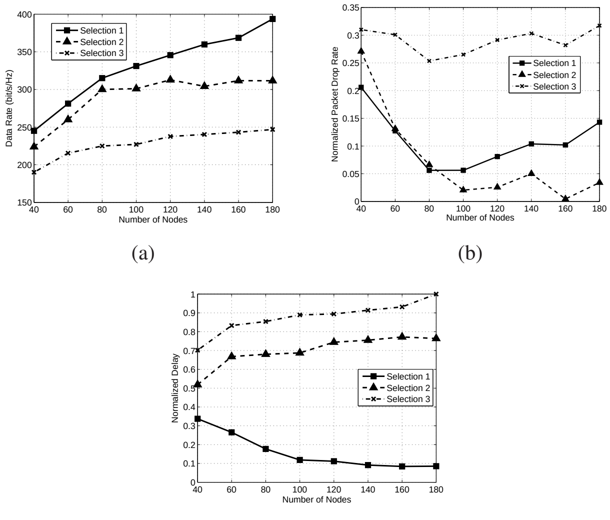
Note that random ID codes are only used for differentiation in control signal transmission. As control signals are relatively short and sent at the maximum power, there is no significant overhead induced for packet encoding and decoding and there is no need for power control.

IX. 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 $250m$. The MIMO channel between node pair is modeled based on the distance between nodes and the small-scale fading coefficients following Rayleigh model. White Gaussian noise with $SNR = 10dB$ is added to include environment noise and interference that cannot be canceled. 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 VIII and the algorithms proposed in Section VII. The centralized multi-user multi-stream scheduling algorithm (CMUMSS) described in Section VI 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, i.e. overloaded. The drop rate is defined to be the total number of dropped packets divided by the total number of transmitted packets. For the convenience of comparison, the results of drop rate are normalized to a maximum value. 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, the number of antenna elements at each node is 4, and the overload factor α defined in Section V-B is 0.



(a)

(b)

(c)

Fig. 2. Performance of DTNS with different types of transmitter nodes selection: (a) data rate; (b) packet drop rate; (c) normalized delay.

(1) Performance of DTNS

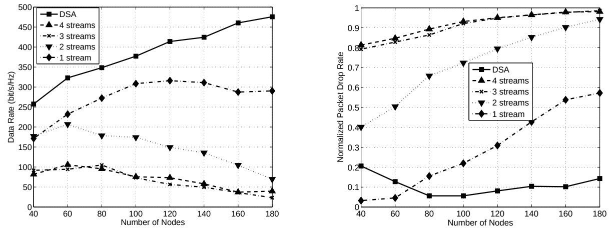
We first evaluate the performance of DTNS by varying the node density. We consider three types of distributed transmitter nodes selection:

- *Selection 1*: Use DTNS as described in VII-A;
- *Selection 2*: Use P^{TX} as described in VII-A, 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, average packet drop rate and normalized delay for the three selection schemes are compared in Figure 2. Selection scheme 3 is seen to have the lowest aggregate rate and the highest dropping rate and normalized delay, as it does not consider node density and load condition in node selection. 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%. In Figure 2 (b), scheme 2 achieves the lowest drop rate in high density case, as its r^{TX} calculation is not impacted by the priority factor which depends on network load and can hence better control the transmission node selection based on the number of active nodes in a neighborhood. As a tradeoff, Figure 2 (c) shows that scheme 1 has much lower average delay compared to scheme 2, as packets with longer queuing delay are favored for transmission in scheme 1. Although scheme 2 has lower packet drop rate than scheme 1 at high node density, its aggregate data rate is lower than scheme 1. This is because the scheduling decision of scheme 1 can better adapt to the traffic demands of nodes and increase the total transmission rate.

(2) Performance of DSA

In Section VII-B, the number of streams allocated is adaptively adjusted according to the traffic condition in the



(a)

(b)

Fig. 3. 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.

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 3 (a) and (b). 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 and the rate is lower when the fixed stream is set at a larger number, for more collisions are induced. In Section IV, a normalized stream quality factor is introduced, which is demonstrated to outperform the simple stream quality factor as in Figure 4.

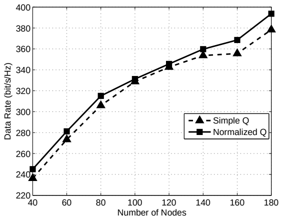
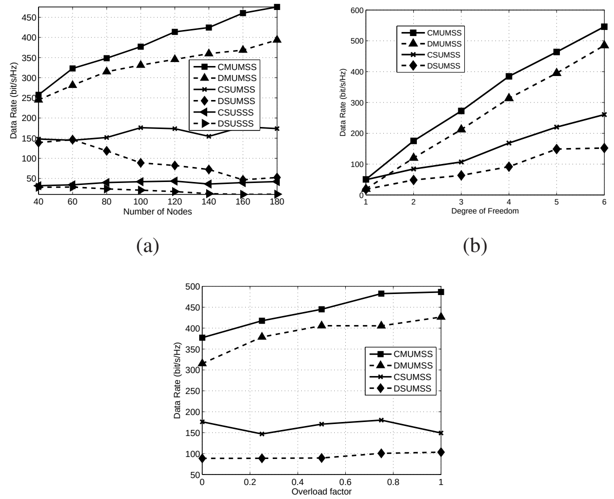


Fig. 4. Data rate with simple and normalized stream quality factor.

(3) Performance of DMUMSS

The overall performance of DMUMSS is evaluated in Figure 5, with CMUMSS, CSUMSS and DSUMSS as references. According to Figure 5(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. We also present the performance of centralized and distributed single-user single-stream scheduling algorithms, denoted as CSUSSS and DSUSSS respectively, where each node only has one antenna and best user pairs are selected opportunistically over the network. As expected, using



(a)

(b)

(c)

Fig. 5. 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.

MIMO transmission especially with multi-user multi-stream scheduling can significantly improve data rate. Figure 5(b) illustrates the changing of data rate with varied values of DoF (degree of freedom). Again, data rate of MUMSS increases almost linearly. In comparison, limited by the single user constraint, the increasing of data rate of SUMSS, especially DSUMSS, is much slower as the number of antennas grows. This figure indicates that MUMSS can be expected to outperform SUMSS as long as there exists some level of DoF. In Section V-A, 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 5(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 between 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.

(4) Robustness to Topology Change Rate

In Figure 6, the aggregate data rate achieved by DMUMSS is further investigated under the different topology update rate v. The topology of the network changes every v number of transmission durations. For all the three representative values of node density simulated, the aggregate data rate remains almost constant with only slight variations. The result shows that our DMUMSS algorithm is robust to topology changes in the network, as it is always able to coordinate the transmissions based on traffic demand and schedule high-quality streams in any topology. This indicates that our scheme will perform well in a mobile ad hoc network with frequent topology change.

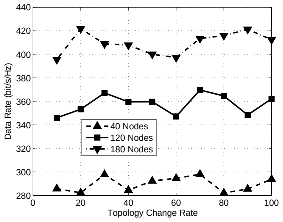


Fig. 6. Data rate of DMUMSS with different topology change rates.

X. Concluding Remarks

In this work, we propose a centralized and a distributed scheduling algorithms in MIMO-based ad hoc networks by concurrently considering traffic demand, service requirements, network load, multiuser diversity, and spatial diversity. Our algorithms fully exploit 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%.

Besides spatial multiplexing, several other techniques can be utilized to further exploit the advantage of MIMO to improve network performance. For instance, space-time coding can be used to increase the reliability of transmissions. It would be intricate but promising to design a cross-layer scheme to adaptively utilize these techniques. These issues will be studied as part of our future work.

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] L. Bao and J.J. Garcia-Luna-Aceves, Transmission Scheduling in Ad Hoc Networks with Directional Antennas, *Proc. ACM/IEEE MobiCom 2002*, 2002.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] M. Levorato, P. Casari, S. Tomasin and M. Zorzi, Physical Layer Approximations for Cross-Layer Performance Analysis in MIMO-BLAST Ad Hoc Networks, *IEEE Trans. on Wireless Communications*, December 2007.

[13] 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.

[14] 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.

[15] S. Chu and X. Wang, Opportunistic and Cooperative Spatial Multiplexing in MIMO Ad hoc Networks, *Proc. ACM MobiHoc 2008*, May. 2008.

[16] Y. Pan and S. Aissa, Performance Analysis of Selective Space-time Coding and Selection Diversity Under Perfect and Imperfect CSI, *Proc. PIMRC 2005*, Sep. 2005.

[17] 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.

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

[19] 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.

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

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

Shan Chu received her B.S. and M.S. degrees in electrical engineering from Zhejiang University, Hangzhou, China. She is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering of the State University of New York at at Stony Brook, Stony Brook, NY. Her current research interests include MIMO and cooperative communications, ad hoc networks and cross-layer design. She is a student member of IEEE.

M'1 / ACM'4 received the B.S. and M.S. degrees in telecommunications engineering and wireless communications engineering from Beijing University of Posts and Telecommunications, Beijing, China, and the Ph.D. degree in electrical and computer engineering from Columbia University, New York, NY.

She is currently an Assistant Professor in the Department of Electrical and Computer Engineering of the State University of New York at Stony Brook, Stony Brook, NY. Before joining Stony Brook, she was a Member of Technical Staff in the area of mobile and wireless networking at Bell Labs Research, Lucent Technologies, New Jersey, and an Assistant Professor in the Department of Computer Science and Engineering of the State University of New York at Buffalo, Buffalo, NY. Her research interests include analysis and architecture design in wireless networks and communications, mobile and distributed computing, infrastructure design and performance enhancement across network layers, applications and heterogeneous networks, network and mobility management, QoS, signaling and control, as well as support for advanced network services and applications.