

Research Article

Cross-Layer Design for End-to-End Throughput Maximization and Fairness in MIMO Multihop Wireless Networks

Jain-Shing Liu,¹ Chun-Hung Richard Lin,² and Kuang-Yuan Tung²

¹Department of Computer Science and Information Engineering, Providence University, 200 Chung Chi Rd., Shalu, Taichung Hsien 43301, Taiwan

²Department of Computer Science and Engineering, National Sun Yat-Sen University, No. 70, Lienhai Rd., Kaohsiung 80424, Taiwan

Correspondence should be addressed to Jain-Shing Liu, chhliu@pu.edu.tw

Received 12 August 2009; Accepted 7 February 2010

Academic Editor: A. Lee Swindlehurst

Copyright © 2010 Jain-Shing Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

MIMO links can significantly improve network throughput by supporting multiple concurrent data streams between a pair of nodes and suppressing wireless interference. In this paper, we study joint rate control, routing, and scheduling in MIMO-based multihop wireless networks, which are traditionally known as transport layer, network layer, and MAC layer issues, respectively. Our aim is to find a rate allocation along with a flow allocation and a transmission schedule for a set of end-to-end communication sessions so as to maximize the network throughput and also to achieve the proportional or weighted fairness among these sessions. To this end, we develop Transmission Mode Generating Algorithms (TMGAs), and Linear Programming- (LP-) and Convex Programming- (CP-) based optimization schemes for the MIMO networks. The performances of the proposed schemes are verified by simulation experiments, and the results show that the different schemes have different performance benefits when achieving a tradeoff between throughput and fairness.

1. Introduction

Recent advances in wireless communications and computing technologies enable a broad range of network applications. To facilitate these applications for the fast-growing number of mobile users and services, the communication society intensifies the interest in the development of novel approaches that can increase the overall network capacity. With the enlarged requirement, future multihop wireless networks such as wireless backhaul networks (WBNs) and wireless mesh networks (WMNs) are conducted to support various data and multimedia transmissions that are usually bandwidth-consumed. In such networks, the Multiple-Input Multiple-Output (MIMO) antenna system, which can offer multiple Degree of Freedom (DOFs) for communications in a node while reducing interference and improving network throughput, is one of the technologies to this end, and attracts much attention of recent research on communication [1–3]. However, when considered with networking, it is still in its early stage. For example, in [4], the authors devise a

MIMO-based MAC protocol and develop analytical methods to characterize the corresponding saturation throughput and study the impact of MIMO MAC on routing. As another example, the authors in [5] give their key optimization considerations such as spatial multiplexing for MAC layer design in ad hoc wireless networks.

On the other hand, cross-layer schemes have been proposed to improve throughput and fairness for multihop wireless networks. For example, in [6, 7], the joint rate control and scheduling problems have been studied for wireless ad hoc networks with either a scheduling-based MAC [6] or an Aloha-based MAC [7], in which routes are assumed to be given a priori, however. In [8], the authors propose Integer Linear Programming formulations and a heuristic algorithm to solve the joint scheduling and power control problems in WMNs. Nevertheless, they consider only the wireless networks without MIMO links. In addition, the authors in [9] also present a centralized algorithm to solve the corresponding joint routing, scheduling, and stream control problem for WMNs. With MIMO links,

the authors provide a three-step approach to this end, but they do not explicitly consider the problem of providing different object functions of fairness for each session in the network. Similarly, a cross-layer optimization for solving the joint stream control and scheduling problem for MIMO-based wireless networks is proposed in [10]. In this work, the authors aim to seek a stream control solution and a transmission schedule with minimum frame length to satisfy traffic demands of links, and pay no attention to the issue of supporting service differentiation for different traffic sessions.

In this paper, we aim to develop cross-layer optimization schemes for multihop wireless networks with MIMO to provide end-to-end throughput optimization among different sessions and support service differentiation for these sessions with proportional or weighted fair share. To this end, we first divide communication flows into components containing only weak contentions. Then, we use Transmission Mode Generating Algorithms (TMGAs, including TMGA1 and TMGA2) to generate a TDMA-based scheduling matrix for these components to be scheduled without interference. As expected, generating all transmission modes for a network (with, e.g., TMGA1) would be time consuming. Thus, we design a polynomial time heuristic (TMGA2) to compute a subset of transmission modes with time efficiency. Given the scheduling matrix, we conduct the cross-layer optimization schemes to address joint network routing, link scheduling, and rate control in such networks with a scheduling-based MAC, which are traditionally known as transport layer, network layer, and MAC layer issues, respectively. In particular, with Linear Programming (LP) and Convex Programming (CP), we seek a rate allocation along with a flow allocation and a transmission schedule such that the network throughput can be maximized and the desired fairness can be achieved. The performances of the proposed schemes are verified by simulation experiments, and their impacts on throughput and fairness for the networks are evaluated. The numerical results show that the proposed schemes can satisfy our design aims and can provide their unique performance benefits when achieving a tradeoff between throughput and fairness. The results also show that TMGA2 can achieve nearly the same performance gains with only a few numbers of iterations and can be very time efficient, when compared with TMGA1 that generates all transmission modes.

The rest of this paper is organized as follows. In Section 2, we summarize the system model for MIMO links. Following that, in Section 3, we introduce our flow scheduling algorithms for the system. In Section 4, we present the cross-layer optimization schemes for the throughput maximization and fairness problem. Finally, these schemes are examined with experiments in Section 5, and conclusions are drawn in Section 6.

2. System Model

2.1. Node Topology Graph. In this work, we consider a multihop wireless network that consists of a finite set V of

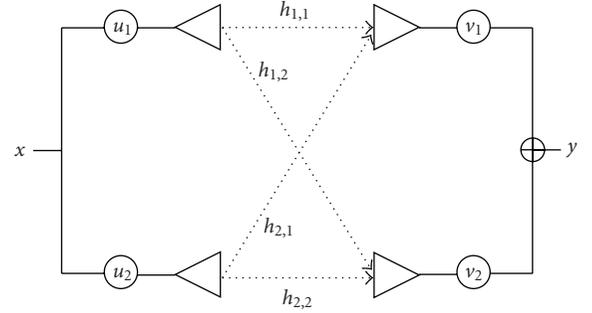


FIGURE 1: Multiantenna wireless channel.

nodes where each node $m \in V$ is stationary and has a MIMO antenna array with K_m elements or DOFs for each node to transmit and receive signals. All nodes transmit at the same fixed power level in a signal common channel, and each node has a uniform transmission range R_T and a uniform interference range $R_I \geq R_T$. Let E denote the set of all pairs (m, n) of distinct nodes in V such that m and n are within each other's transmission range. An ordered pair of nodes (m, n) in E is said to form a flow $f = \overrightarrow{(m, n)}$ if node m needs to transmit to node n . f is said to be *active* if m is currently transmitting to n ; otherwise, f is said to be *inactive*. Let F denote the set of all flows. Hereafter, the graph $G = (V, E, F)$ defined above is referred to as *node topology graph*.

2.2. Overview of MIMO Antenna Array Processing. Figure 1 illustrates the transmitter and receiver antenna array and the MIMO channel for the case when $N = 2$ antennas are used at the both ends. To transmit a signal $x(t)$ through a transmit beamformer over the 2-antenna array, the transmitter sends two weighted copies of the signal, one on each antenna. That is, $u_1x(t)$ is sent over antenna 1, and $u_2x(t)$ is sent over antenna 2, with $\mathbf{u} = [u_1, u_2]^T$ called the *transmitting weight vector*. Then, the two signals are weighted by the receiver with a *receiving weight vector* $\mathbf{v} = [v_1, v_2]^T$ and summed to produce $y(t)$. Let \mathbf{H} denote the matrix of channel coefficients between the transmitter and the receiver. The above can thus be written as $y(t) = (\mathbf{u}^T \mathbf{H} \mathbf{v})x(t)$, and the complex gain experienced by $x(t)$ is then a consequence of transmit beamforming, the channel, and receive beamforming of $\mathbf{u}^T \mathbf{H} \mathbf{v}$. Now with appropriate weight vectors \mathbf{u} and \mathbf{v} , the received signal $y(t)$ can achieve a unit gain ($\mathbf{u}^T \mathbf{H} \mathbf{v} = 1$) if it is received at the desired receiver or a zero gain ($\mathbf{u}^T \mathbf{H} \mathbf{v} = 0$) if it is received at a undesired receiver. In other words, we can ensure that the signal is received with a certain gain or is perfectly nulled by appropriately choosing the weight vectors if no power limit on a node's receiving capacity. In general, we could design \mathbf{u} if \mathbf{v} is given, and vice versa. Thus, by considering whether the transmitter and the receiver are a desired communication pair or potentially interference with one another, we have the following *beamforming conditions*.

- (1) If the receiver corresponding to \mathbf{v} is the desired receiver of $x(t)$ and \mathbf{v} is fixed, then we could require \mathbf{u} to satisfy $\mathbf{u}^T \mathbf{H} \mathbf{v} = 1$ so that $x(t)$ can be received with unit gain.



FIGURE 2: Node topology graph.

- (2) If the receiver corresponding to \mathbf{v} is already involved in another link, then \mathbf{v} is fixed and we could require \mathbf{u} to satisfy $\mathbf{u}^T \mathbf{H} \mathbf{v} = 0$ so that the transmitter does not create interference at this receiver.
- (3) If the transmitter is communicating with a different user by using a fixed \mathbf{u} , and the receiver corresponding to \mathbf{v} wants to receive signals from different transmitter without interference, then we could require \mathbf{v} to satisfy $\mathbf{u}^T \mathbf{H} \mathbf{v} = 0$ so as to null the contribution of \mathbf{u} at the receiver.

2.3. Transmission Constraints on MIMO Links. Consider the multihop wireless network $G = (V, E, F)$ shown in Figure 2, where $V = \{1, 2, 3, 4\}$, $E = \{(2, 1), (3, 2), (4, 3)\}$, and $F = \{f_1 = (2, 1), f_2 = (4, 3)\}$. Assume that (1) every link has the same R_T and $R_I = 1.5 R_T$, (2) the numbers of antennas are $K_1 = K_2 = 1$ and $K_3 = K_4 = 2$, and (3) f_1 and f_2 are the only two flows in the network. With the above, we consider the example that node 4 wants to transmit to node 3 while node 2 is currently transmitting to node 1 (i.e., f_1 is active). Let \mathbf{u}_2 be the (1×1) transmitting weight vector that node 2 is currently using to weight its transmitted signal and let \mathbf{v}_1 be the (1×1) receiving weight vector that node 1 is currently using to weight the signal received from node 2. In this case that f_1 is currently active, if node 3 wants to receive an interference-free signal from node 4, it must design its receiving weight vector \mathbf{v}_3 to suppress the interference caused by node 2's transmission while assuring an acceptable gain of its intended signal coming from node 4. In terms of equations, the above can be written as $(\mathbf{u}_4^T \mathbf{H}_{4,3} \mathbf{v}_3) = 1$ and $(\mathbf{u}_2^T \mathbf{H}_{2,3} \mathbf{v}_3) = 0$, where $\mathbf{u}_4 = [u_{4,1}, u_{4,2}]^T$ is the transmitting weight vector of node 4, and $\mathbf{v}_3 = [v_{3,1}, v_{3,2}]^T$ is the receiving weight vector of node 3. Now, given \mathbf{u}_4 , \mathbf{u}_2 , $\mathbf{H}_{4,3}$ and $\mathbf{H}_{2,3}$, the system of these two equations would have a unique solution \mathbf{v}_3^* because the elements of each of $\mathbf{H}_{4,3}$, and $\mathbf{H}_{2,3}$ are i.i.d in general. That is, if there is no power limit on node 3's receiving capacity, it is always possible for node 3 to receive one interference-free flow from node 4 concurrently with the signals from node 2. In other words, f_1 and f_2 are always possible to be active simultaneously in this example.

By reserving the admission order of these flows, we now consider another example in which node 2 wants to transmit to node 1 while node 4 is currently transmitting to node 3 (i.e., f_2 is active). With similar considerations for the above, this example can be represented by the equations of $(\mathbf{u}_2^T \mathbf{H}_{2,1} \mathbf{v}_1) = 1$ and $(\mathbf{u}_4^T \mathbf{H}_{4,3} \mathbf{v}_3) = 0$, but now the objective to be solved becomes \mathbf{u}_2 , which is (1×1) weight vector. That is, the single variable \mathbf{u}_2 involves the system of the two equations associated with $\mathbf{H}_{2,1}$ and $\mathbf{H}_{4,3}$ that are i.i.d in general. Obviously, the over-determined system has no solution.

By summarizing these examples, we can find that the admission of (f_1, f_2) is order dependent. That is, if f_1 is admitted first, then f_2 is possibly admitted. On the contrary, if f_2 is admitted first, then f_1 could not be admitted without inference. Clearly, these could be verified with the beamforming conditions given previously. However, to be specific, we define the transmission constraints as follows. Let node m be the transmitter and node n the receiver, and suppose that there are β streams currently received by nodes located within m 's interference range, and γ streams currently transmitted by nodes located within n 's interference range, where a steam denotes a copy of data signal transmitted by the MIMO system. Suppose further that m wants to transmit an α -stream flow f to n . Then, for interference-free communications in the system, we have the following constraints.

Theorem 1 (transmit DOF constraint). *m can transmit an α -stream flow f to n without interfering with the β streams concurrently received by m 's neighbors if*

$$\alpha + \beta \leq K_m. \quad (1)$$

Proof. Let $\{m_1, m_2, \dots, m_p\}$ denote the set of m 's neighbors that are receiving the β streams and β_i denote the number of streams that node m_i is currently receiving, for all $i \in \{1, 2, \dots, p\}$ (i.e., $\sum_{i=1}^p \beta_i = \beta$). Suppose that for each i , m knows the receiving weight vector of m_i , and the channel matrix between it and m_i , for all i . Then, m can transmit its α -stream flow f to n with the weight vectors $\mathbf{u}_{m,j}$, $j \in \{1, 2, \dots, \alpha\}$ subject to the conditions of $\mathbf{A}_m \mathbf{u}_{m,j} = \mathbf{a}_j, \forall j$, where \mathbf{a}_j is the column vector of length $\alpha + \beta$ defined as $[0 \ 0 \ \dots \ 0 \ 1 \ 0 \ 0 \ \dots \ 0]^T$ with 1 at the j th position, and \mathbf{A}_m is the $(\alpha + \beta) \times K_m$ matrix defined as $[\hat{\mathbf{A}}_{m,n} \ \hat{\mathbf{A}}_{m,1} \ \hat{\mathbf{A}}_{m,2} \ \dots \ \hat{\mathbf{A}}_{m,p}]^T$ with $\hat{\mathbf{A}}_{m,n} = [\mathbf{H}_{m,n} \mathbf{v}_{n,1} \ \mathbf{H}_{m,n} \mathbf{v}_{n,2} \ \dots \ \mathbf{H}_{m,n} \mathbf{v}_{n,\alpha}]$ and $\hat{\mathbf{A}}_{m,i} = [\mathbf{H}_{m,m_i} \mathbf{v}_{m_i,1} \ \mathbf{H}_{m,m_i} \mathbf{v}_{m_i,2} \ \dots \ \mathbf{H}_{m,m_i} \mathbf{v}_{m_i,\beta_i}]$, for all $i \in \{1, 2, \dots, p\}$. Because the elements of each \mathbf{H} involved would be i.i.d in general, the system of equations $\mathbf{A}_m \mathbf{u}_{m,j} = \mathbf{a}_j$, for all j , with \mathbf{A}_m of the size of $(\alpha + \beta) \times K_m$, would have no solution if $(\alpha + \beta) > K_m$. \square

Theorem 2 (receive DOF constraint). *n can receive an α -stream flow f from m without interfering with the γ streams concurrently transmitted by n 's neighbors if*

$$\alpha + \gamma \leq K_n. \quad (2)$$

Proof. Let $\{n_1, n_2, \dots, n_q\}$ denote the set of n 's neighbors that are transmitting the γ streams and γ_i denote the number of streams that node n_i is currently transmitting, for all $i \in \{1, 2, \dots, q\}$ (i.e., $\sum_{i=1}^q \gamma_i = \gamma$). Suppose that for each $i \in \{1, 2, \dots, q\}$, n knows the transmitting weight vector of n_i , and the channel matrix between it and n_i , for all i . Then, n can receive its α -stream flow f from m with the weight vectors $\mathbf{v}_{n,j}$, $j \in \{1, 2, \dots, \alpha\}$ subject to the conditions of $\mathbf{B}_n \mathbf{v}_{n,j} = \mathbf{b}_j, \forall j$, where \mathbf{b}_j is the column vector of length $\alpha + \gamma$ defined as $[0 \ 0 \ \dots \ 0 \ 1 \ 0 \ 0 \ \dots \ 0]^T$ with 1 at the j th position, and \mathbf{B}_n is

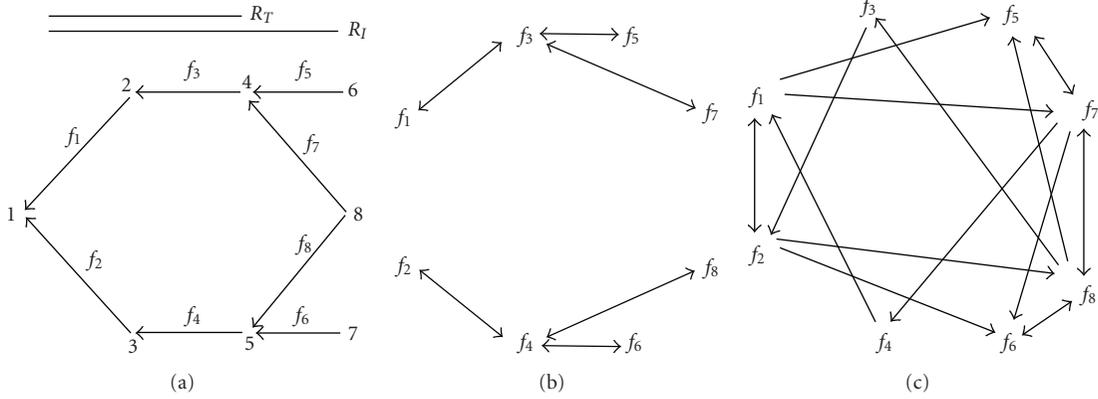


FIGURE 3: An example of weak and strong contentions: (a) the node topology graph G , (b) the corresponding strong contention graph G_{strong} , and (c) the corresponding weak contention graph G_{weak} .

the $(\alpha + \gamma) \times K_n$ matrix defined as $[\hat{\mathbf{B}}_{m,n} \hat{\mathbf{B}}_{1,n} \hat{\mathbf{B}}_{2,n} \cdots \hat{\mathbf{B}}_{q,n}]^T$ with $\hat{\mathbf{B}}_{m,n} = [\mathbf{H}_{m,n}^T u_{m,1} \mathbf{H}_{m,n}^T u_{m,2} \cdots \mathbf{H}_{m,n}^T u_{m,\alpha}]$ and $\hat{\mathbf{B}}_{i,n} = [\mathbf{H}_{i,n}^T u_{i,1} \mathbf{H}_{i,n}^T u_{i,2} \cdots \mathbf{H}_{i,n}^T u_{i,\gamma}]$, for all $i \in \{1, 2, \dots, q\}$. Because the elements of each \mathbf{H} involved would be i.i.d in general, the system of equations $\mathbf{B}_n \mathbf{v}_{n,j} = \mathbf{b}_j$, for all j , with \mathbf{B}_n of the size of $(\alpha + \gamma) \times K_n$ would have no solution if $(\alpha + \gamma) > K_n$. \square

3. Flow Scheduling Algorithms

In this section, we present our flow scheduling algorithms for MIMO networks. As the first step to this end, we divide the flow contentions in a MIMO system into two categories: *strong interference* and *weak interference*, similar to that given in [10]. The strong interference denotes that an incoming flow into a node cannot be scheduled in the same time slot with any outgoing flow from the same node and vice versa. This is because a node in wireless networks is usually half-duplexing and thus it cannot simultaneously transmit and receive. In other words, any two flows $f_i = (u_i, v_i)$ and $f_j = (u_j, v_j)$ are said to strongly contend with each other if and only if $u_j = v_i$ or $u_i = v_j$. Given that, we define $G_{\text{strong}}(V_{\text{strong}}, E_{\text{strong}})$ as the *strong contention graph*, where a vertex in V_{strong} corresponds to a flow in F , and a bidirectional edge in E_{strong} denotes the corresponding flows in F strongly contend with each other in the node topology graph G .

On the other hand, the weak interference denotes that a pair of flows would contend for resources but they could be scheduled in the same time slot if the related DOFs can be properly arranged. Similarly, we define $G_{\text{weak}}(V_{\text{weak}}, E_{\text{weak}})$ as the *weak contention graph*. In this graph, a directional edge between a pair of vertices corresponding to $f_i = (u_i, v_i)$ and $f_j = (u_j, v_j)$ in G arises if any one of the three weak-interference conditions holds: (1) $u_i = u_j$, (2) $v_i = v_j$, and (3) there exists a $f_k = (u_k, v_k)$ (possibly $k = i$ or $k = j$) such that $\|v_k - u_i\| \leq R_I$ and/or $\|v_k - u_j\| \leq R_I$. To be specific, we consider a MIMO network of 8 nodes and 8 flows as an example, showing its node topology graph G in

Figure 3(a) and the derived G_{strong} and G_{weak} in Figures 3(b) and 3(c), respectively. For example, we can see in Figure 3(a) that when node 4 is transmitting to node 2 (with f_3), node 2 cannot simultaneously transmit to node 1 (with f_1), and vice versa. This is a strong interference, and is denoted by a bidirectional edge between f_1 and f_3 in Figure 3(b). Similarly, we can also see in Figure 3(a) that when node 8 simultaneously transmits to node 4 and node 5, f_7 and f_8 will contend with each other. This corresponds to the first weak-interference condition, and is shown by a bidirectional edge between f_7 and f_8 in Figure 3(c). In addition, the second weak-interference condition is exemplified by the scenario in Figure 3(a) that nodes 2 and 3 simultaneously send to node 1, and thus f_1 and f_2 interfere with each other. Therefore, there is a bidirectional edge between f_1 and f_2 in Figure 3(c). Finally, the third weak-interference condition can be found in Figure 3(a) that when node 4 transmits to node 2 (with f_3), this node can interfere with node 1's receipt of f_2 from node 3. This interference is shown by a directional edge from f_3 to f_2 in Figure 3(c). Besides, the other strong and weak interferences not exemplified in the above can be also observed easily.

Clearly, the flows strongly contending with each other cannot be scheduled at the same time. Thus, the first step of our algorithms is to divide the flows into a set of components, say $\{C_o\}$, that contain only weak contentions and can be represented by $\{C_{o_{\text{weak}}}\}$. This step is done by finding a valid coloring for G_{strong} with, for example, the greedy algorithm given in [11] that sorts vertices in G_{strong} by decreasing vertex degrees, and according to the order, colors them one by one using a first-fit greedy approach. Then, the flows of G with the same color in G_{strong} compose a C_o , and the corresponding subset of G_{weak} composes a $C_{o_{\text{weak}}}$.

3.1. Transmission Modes. After obtaining $\{C_o\}$ and $\{C_{o_{\text{weak}}}\}$, we now proceed to find an interference-free scheduling for each of the components with a scheduling-based MAC. To this end, we define *transmission mode* as the flows in G that can be active simultaneously. A $t_\Gamma \times m_\Gamma$ matrix Γ is used to represent the set of transmission modes, and

called *scheduling matrix*, in which t_Γ denotes the number of transmission modes, and m_Γ denotes the number of flows in G . In the matrix, a row represents a transmission mode T_M , and its element Γ_{T_M, f_i} represents the number of traffic streams utilized by flow f_i in this mode. That is, a T_M is a vector denoting a possible set of transmissions from all m_Γ flows in a time slot. By definition, all flows with traffic streams in a transmission mode can be activated simultaneously. However, the concurrent flows may interfere with each other in the MIMO network. Thus, the scheduling algorithms should firstly generate interference-free T_M s for constructing a desired Γ . Next, with these modes representing the time slots to be used, they are expected to determine the time fraction p_{t_m} for every transmission mode or time slot to compose a TDMA frame that can satisfy the scheduling target. Accordingly, the scheduling algorithms should secondly find the frame length and the number of active time slots of each transmission mode in one frame. This can be done by considering that if the value of p_{t_m} for every T_M is given, the frame length is then the smallest positive integer I such that $p_{t_m} \cdot I$ is an integer for every transmission mode, and the number of active times for a T_M is simply its $p_{t_m} \cdot I$. Now, given the two aims of the scheduling algorithms, we proceed to develop the transmission mode generating algorithms for T_M s in the next subsection, and leave the methods for determining p_{t_m} s till Section 4.

3.2. Transmission Mode Generating Algorithm 1. To obtain the transmission modes for MIMO networks, we design Transmission Mode Generating Algorithm 1, or shortly TMGA1. As shown in Algorithm 1, the inputs of TMGA1 include the set of source DOFs $\{K_m\}$, the set of destination DOFs $\{K_n\}$, and the set of weak contention graphs $\{C_{o\text{weak}}\}$, for the flow components $\{C_o\}$ obtained previously. For each C_o , the algorithm generates all possible transmission modes T_M s, according to the component's K_m and K_n . Then, each of the modes is examined for its validity conservatively or nonconservatively. Two different versions of TMGA1, namely, TMGA1-con and TMGA1-non, are conducted here according to Section 2.3 revealing that the admission of flows is order dependent. In TMGA1-con, a transmission mode is considered to be valid if and only if all its admission orders, A_O s, can lead to the mode. On the contrary, in TMGA1-non, the mode is said to be valid if there exists at least one A_O to confirm its validity. For either of the versions, the algorithm requires Step 1 to update α , β , and γ with

$$\begin{aligned} \alpha(f_i) &= T_M(f_i), \\ \beta(f_i) &= \sum_{\forall f_j \in N_S^i(f_i)} \alpha(f_j), \\ \gamma(f_i) &= \sum_{\forall f_j \in N_R^i(f_i)} \alpha(f_j), \end{aligned} \quad (3)$$

where f_i is the flow in G_o now considered for the proceeding A_O and T_M , and $N_S^i(f_i)$ is the set of neighbors, in G_o , located within R_I of f_i 's source node, and $N_R^i(f_i)$ is that for f_i 's destination node.

Following that, in Step 2, these updated values are used to verify whether the $\alpha(f_i)$ -stream traffic can be established on f_i without interfering with the other flows according to the DOF constrains given in (1) and (2). If all f_i 's in the A_O are verified successfully, the corresponding T_M could then be conservatively or nonconservatively considered to be an element of Γ_o (Γ for C_o). In addition, to reduce the size of Γ_o , the T_M 's with their capabilities being subset of the others will not be included. Then, all the reduced Γ_o 's are merged to form a single scheduling matrix Γ so that its size (t_Γ) can be tractable for the cross-layer design. Finally, after added with an empty T_M as the default element for scheduling, the complete Γ is given in Step 3.

To be clear, let us review the example in Figure 2. Obviously, this example presents no strong contention and thus TMGA1 only needs to consider a single flow component $C_o = \{f_1, f_2\}$. As already shown, this example has $K_1 = K_2 = 1$ and $K_3 = K_4 = 2$. However, they are represented here by $\{K_m\} = \{1, 2\}$ and $\{K_n\} = \{1, 2\}$ to be the inputs of TMGA1 in addition to $C_{o\text{weak}} = (V_{\text{weak}}, E_{\text{weak}})$, where $V_{\text{weak}} = \{f_1, f_2\}$ and $E_{\text{weak}} = \{f_1 f_2\}$. With these inputs, the algorithm generates a set of all possible transmission modes $\{T_M\} = \{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2)\}$. Among these, we consider the mode of (1, 1) with its two possible admission orders, $\{A_O\} = \{(f_1, f_2), (f_2, f_1)\}$, as before. For the first order, TMGA1 initializes $\{\alpha\} = \emptyset$, $\{\beta\} = \emptyset$, and $\{\gamma\} = \emptyset$. Then it updates for f_1 with $\alpha(f_1) = T_M(f_1) = 1$, $\beta(f_1) = \sum_{\forall f_j \in N_S^i(f_1)} \alpha(f_j) = \alpha(f_2) = 0$, and $\gamma(f_1) = \sum_{\forall f_j \in N_R^i(f_1)} \alpha(f_j) = 0$ (since $\nexists f_j \in N_R^i(f_1)$), and the updated values satisfy the DOF constrains by $\alpha(f_1) + \beta(f_1) (= 1) \leq K_m(f_1) (= 1)$, and $\alpha(f_1) + \gamma(f_1) (= 1) \leq K_n(f_1) (= 1)$. After that, it updates for f_2 with $\alpha(f_2) = T_M(f_2) = 1$, $\beta(f_2) = \sum_{\forall f_j \in N_S^i(f_2)} \alpha(f_j) = 0$ (since $\nexists f_j \in N_S^i(f_2)$), and $\gamma(f_2) = \sum_{\forall f_j \in N_R^i(f_2)} \alpha(f_j) = \alpha(f_1) = 1$, and these also satisfy the DOF constrains by $\alpha(f_2) + \beta(f_2) (= 1) \leq K_m(f_2) (= 2)$ and $\alpha(f_2) + \gamma(f_2) (= 2) \leq K_n(f_2) (= 2)$. Thus, the first order (f_1, f_2) is valid.

Similarly, after the initialization, the update for the second order will start with f_2 , resulting in $\alpha(f_2) = T_M(f_2) = 1$ and $\beta(f_2) = 0$, as before; but now $\alpha(f_1) = 0$ and thus $\gamma(f_2) = \alpha(f_1) = 0$. Nevertheless, the above still satisfies the DOF constrains. When then updating for f_1 , it results in $\alpha(f_1) = T_M(f_1) = 1$ in addition to $\alpha(f_2) = 1$ updated previously. Thus $\beta(f_1) = \alpha(f_2) = 1$ instead of 0, while $\gamma(f_1) = 0$ without change. Consequently, it can only satisfy the receive DOF constrain by $\alpha(f_1) + \gamma(f_1) (= 1) \leq K_n(f_1) (= 1)$, but fails on the transmit DOF constraint since $\alpha(f_1) + \beta(f_1) (= 2) \not\leq K_m(f_1) (= 1)$. Finally, with the two A_O 's results, if the nonconservative version (TMGA1-non) is adopted, the transmission mode $T_M = (1, 1)$ is accepted; otherwise, if the conservative version (TMGA1-con) is adopted, the mode is rejected.

For the time complexity of TMGA1, we recall $f = \overline{(u, v)}$ and denote by $\xi(f) = \delta_{\text{in}}(u) + \delta_{\text{out}}(v)$ the number of flows that f has a strong contention with, where $\delta_{\text{in}}(v)$ and $\delta_{\text{out}}(v)$ refer to *in* and *out* degrees of the node v , respectively. With that, we can let $\xi = \max_f \xi(f)$ and denote it by $\xi = \Delta(G_{\text{strong}})$. Then, the number of C_o in the network would

```

(1) INPUT:  $\{C_o\}, \{K_m\}, \{K_n\}, \{C_{o_{\text{weak}}}\}$ ;
(2) for all  $C_o$  do
(3)   Generate all transmission modes,  $T_M$ s, for  $C_o$ 
(4)   for all  $T_M$  do
(5)     Generate all admission orders,  $A_O$ s, for  $T_M$ ;
(6)     for all  $A_O$  do
(7)       Initialize  $\{\alpha\} = \emptyset, \{\beta\} = \emptyset, \{\gamma\} = \emptyset$ ;
(8)       for each  $f_i$  sorted with the increasing order of  $A_O$  do
(9)         {Step 1: update  $\alpha, \beta$  and  $\gamma$ }
(10)         $\alpha(f_i) = T_M(f_i)$ ; {Update  $\alpha$ }
(11)         $\beta(f_i) = \sum_{\forall f_j \in N_s^i(f_i)} \alpha(f_j)$ ; {Update  $\beta$ }
(12)         $\gamma(f_i) = \sum_{\forall f_j \in N_r^i(f_i)} \alpha(f_j)$ ; {Update  $\gamma$ }
(13)        {Step 2: verify for the Transmit/Receive DOF constraints}
(14)        Success( $f_i$ ) = 1;
(15)        if  $\alpha(f_i) + \beta(f_i) > K_m(f_i)$  or  $\alpha(f_i) + \gamma(f_i) > K_n(f_i)$  then
(16)          Success( $f_i$ ) = 0;
(17)        end if
(18)      end for
(19)      if Success( $f_i$ ) = 1,  $\forall i$  then
(20)        Success( $A_O$ ) = 1;
(21)      end if
(22)    end for
(23)    if ((Conservative == 1 and Success( $A_O$ ) = 1,  $\forall A_O$ ) or
(24)      (Conservative == 0 and Success( $A_O$ ) = 1,  $\exists A_O$ )) then
(25)       $\Gamma_o = \Gamma_o \cup T_M$ ;
(26)    end if
(27)  end for
(28)  Reduce  $\Gamma_o$  to be that in which  $\nexists \hat{T}_M, T_M$  s.t.  $\hat{T}_M(i) \leq T_M(i), \forall i$ 
(29)  {Step 3: merge and output the scheduling matrix}
(30)  Merge  $\Gamma_o$ s to form a single  $\Gamma = \{\Gamma_1, \dots, \Gamma_p\}$ , where  $p = |\{C_o\}|$ ;
(31) OUTPUT:  $\Gamma = \Gamma \cup T_M$  of  $\emptyset$ ;

```

ALGORITHM 1: Transmission mode generating Algorithm 1 (TMGA1).

be $p \leq \Delta(G_{\text{strong}}) + 1 = \xi + 1$ if the coloring algorithm in [11] is adopted. Now consider that the j th $C_o = \{f_i\}$ has the number of transmission modes $n_j \in O(\prod_{i=1}^{i=s_j} |f_i|)$, where $s_j \in O(m/\xi)$ (with m denoting the total number of flows in G) is the size of this C_o , and $|f_i|$ is the DOF of its element flow f_i . In the C_o , each transmission mode has o_j admission orders, which is the total number of permutations for the s_j elements (flows). With the above, we can show the time complexity of TMGA1 as $O(\sum_{j=1}^{j=p} n_j \cdot o_j)$.

Although the time complexity could be high as shown in what mentioned above, TMGA1 is the only way to explore all possible transmission modes that are feasible according to the transmit DOF constraint and receive DOF constraint for the MIMO networks. In fact, designing interference-free scheduler for multihop wireless networks is considered to be a hard problem in the literature, and recent works have shown that it is in fact NP complete [12, 13]. Thus, as a rule of thumb, if one has no time to find all the modes with an algorithm such as TMGA1, a possible solution is using a polynomial time heuristic such as the algorithm (TMGA2) shown in what follows to generate a subset of these modes satisfactory enough, within a reasonable time limit.

3.3. Transmission Mode Generating Algorithm 2. In what mentioned before, TMGA1 can find all T_M 's for each C_o . However, the number of T_M s will grow exponentially with the increase of the size of C_o , which would be intractable when it is relatively large. Therefore, we propose here a polynomial time heuristic algorithm, namely, Transmission Mode Generating Algorithm 2, or shortly, TMGA2, that can generate a good subset of T_M 's for each C_o . The central idea of TMGA2 is to consider a $f_i \in C_o$ as the first flow to be admitted for a S -stream flow, and then randomly choose other f_j 's $\in C_o$ to see if they could cooperatively construct a valid T_M . Note that the starting f_i is chosen with an increasing order and the following f_j s could be the same of f_i , implying that multiple S -streams could be established in a single f_i . Furthermore, the seeking process could be repeated several times according to the iteration limit \mathcal{L} . With that, we can control its time complexity to be satisfactory enough while obtaining a good Γ_o that can cover all flows in C_o and can evenly distribute the number of times that each flow is included in certain T_M s.

To fulfill the design aim, we add $\{V\}$ and $\{W\}$ in this algorithm in addition to $\{\alpha\}$, $\{\beta\}$, and $\{\gamma\}$ given previously.

```

(1) INPUT:  $\{C_o\}, \{K_m\}, \{K_n\}, \{C_{o_{weak}}\}, \mathcal{L}, \delta$ ;
(2) for all  $C_o$  do
(3)   Initialize  $\{V\} = \emptyset$ ;
(4)   for  $l = 1$  to  $l = \mathcal{L}$  do
(5)     Initialize  $\{W\} = \min\{\{K_m\}, \{K_n\}\}, \{\alpha\} = \emptyset, \{\beta\} = \emptyset, \{\gamma\} = \emptyset$ ;
(6)     for  $i = 1$  to  $i = |C_o|$  do
(7)       Initialize  $\{\alpha\} = \emptyset, \{\beta\} = \emptyset, \{\gamma\} = \emptyset, \{T_M\} = \emptyset, k = \text{TRUE}$ ;
(8)        $\alpha(f_i) += \delta, V(f_i) = \text{random}; W(f_i) -= \delta$ ;
(9)       while  $k == \text{TRUE}$  do
(10)        if  $W(f_i) \geq 0$  then
(11)          {Primary update;}
(12)           $W(f_j) -= \delta, \forall f_j \in N_S^l(f_i) \cup N_R^l(f_i);$  {Update  $W$ }
(13)           $\beta(f_i) = \sum_{\forall f_j \in N_S^l(f_i)} \alpha(f_j);$  {Update  $\beta$  #1}
(14)           $\gamma(f_i) = \sum_{\forall f_j \in N_R^l(f_i)} \alpha(f_j);$  {Update  $\gamma$  #1}
(15)          {Optional update;}
(16)           $\beta(f_j) += \alpha(f_i), \forall f_j \in N_R^l(f_i)$  with  $\alpha(f_j) > 0;$  {Update  $\beta$  #2}
(17)           $\gamma(f_j) += \alpha(f_i), \forall f_j \in N_S^l(f_i)$  with  $\alpha(f_j) > 0;$  {Update  $\gamma$  #2}
(18)        end if
(19)        Success( $f_i$ ) = 1;
(20)        {Primary verification;}
(21)        if  $\alpha(f_i) + \beta(f_i) > K_m(f_i)$  or  $\alpha(f_i) + \gamma(f_i) > K_n(f_i)$  then
(22)          Success( $f_i$ ) = 0;
(23)        end if
(24)        {Optional verification;}
(25)        for all  $f_j \in N_S^l(f_i) \cup N_R^l(f_i)$  with  $\alpha(f_j) > 0$  do
(26)          if  $\alpha(f_j) + \beta(f_j) > K_m(f_j)$  or  $\alpha(f_j) + \gamma(f_j) > K_n(f_j)$  then
(27)            Success( $f_j$ ) = 0;
(28)          end if
(29)        end for
(30)        if Success( $f_i$ ) = 0 then
(31)          Roll back  $V, W, \alpha, \beta, \gamma,$  and stamp  $f_i$  as failed; {Roll back if  $f_i$  is not valid}
(32)        end if
(33)        Find the next  $f_i$  with  $V(f_i) = \min\{V\}$  and  $W(f_i) \geq \delta$ , among  $\forall f_i$  not yet examined and not yet failed;
(34)        if no  $f_i$  can be found then
(35)           $k = \text{FALSE}$ ;
(36)        else
(37)           $\alpha(f_i) += \delta, V(f_i) = \text{random}, W(f_i) -= \delta$ ;
(38)        end if
(39)      end while
(40)       $T_M(f_i) = \alpha(f_i), \forall i$ ;
(41)      if  $\nexists \hat{T}_M \in \Gamma_o == T_M$  then
(42)         $\Gamma_o \cup = T_M$ ;
(43)      end if
(44)    end for
(45)  end for
(46)  Reduce  $\Gamma_o$  to be that in which  $\nexists \hat{T}_M, T_M$  s.t.  $\hat{T}_M(i) \leq T_M(i), \forall i$ ;
(47) end for
(48) Merge  $\Gamma_o$ s to form a single  $\Gamma = \{\Gamma_1, \dots, \Gamma_p\}$ , where  $p$  is the number of  $\Gamma_o$  found;
(49) OUTPUT:  $\Gamma = \Gamma \cup T_M$  of  $\emptyset$ ;

```

ALGORITHM 2: Transmission mode generating Algorithm 2 (TMGA2).

More precisely, $\{V\}$ is the set of values associated with f_i 's chosen probabilities. In the current version, the values are given with uniform random numbers, and TMGA2 chooses the next f_i with the lowest value; that is, it will choose uniformly and randomly among the f_i 's. On the other hand, $\{W\}$ denotes the set of link capabilities for the flows and is

obtained by $\{W\} = \min\{\{K_m\}, \{K_n\}\}$. In addition, two new update rules are considered for β and γ , respectively, as

$$\begin{aligned}
 \beta(f_j) &+= \alpha(f_i), \quad \forall f_j \in N_R^l(f_i) \text{ with } \alpha(f_j) > 0, \\
 \gamma(f_j) &+= \alpha(f_i), \quad \forall f_j \in N_S^l(f_i) \text{ with } \alpha(f_j) > 0.
 \end{aligned} \tag{4}$$

In what mentioned previously, f_i is the flow now considered in C_o . Clearly, each neighbor of f_i 's destination node would consume $\alpha(f_i)$ DOFs to null its interference at the receiver if it has data to transmit right after f_i . Thus, conservatively it should take f_i 's α into account as a part of its own β and then check the transmit DOF constraint to avoid the interference. Symmetrically, each neighbor of f_i 's source node will be interfered with f_i 's transmission if it wants to concurrently receive its own traffic right after f_i . Hence, by taking f_i 's α into account as a part of its own γ for the receive DOF constraint, it could be interference-free in the situation.

In TMGA2, if the optional update and the optional verification (as shown in lines 16 to 17, and lines 25 to 29 in Algorithm 2, resp.) are considered, the algorithm is operated conservatively, and called TMGA2-con. On the other hand, if these optional parts are not involved, then it is TMGA2-non. Obviously, the two versions correspond to those of TMGA1, and their performances will be compared in the experiments.

Let us use Figure 2, again, as our example and set $\delta = 1$ so that the algorithm will consider 1-stream flow for each admission. With the initial $\{\alpha\} = \{0, 0\}$ and $\{W\} = \{1, 2\}$, TMGA2 starts by assuming f_1 to be admitted and accordingly updating the related parameters to be $\{\alpha\} = \{1, 0\}$ and $\{W\} = \{0, 2\}$. In the same time, the neighbor of f_1 , that is, f_2 , should also consume its link capability (DOF) to be interference-free from f_1 's transmission. Thus, $\{W\}$ is further changed to be $\{0, 1\}$. In what follows, it updates f_1 's α and β with the equations in (3) as TMGA1 does. If TMGA2-non is considered, the algorithm will proceed to examine the next f_i that has the lowest random value V and has its link capability W equal to or larger than $\delta (= 1)$. Now f_2 is the only candidate, and the following process for updating α and β and verifying the DOF constraints is the same as that for TMGA1. Finally, $\{\alpha\} = \{1, 1\}$ found is used to update T_M as $(1, 1)$, which is the result of this case. Then, with f_2 as the new start, the process continues to search other valid T_M , and will end after the searching. In fact, the algorithm is designed to repeat the whole process \mathcal{L} times, and with the random nature of V , each of the iterations may lead to a different T_M complying with our design aim.

Now, let us turn out to focus on the conservative version of TMGA2. As shown in the TMGA1 example, $T_M = (1, 1)$ is accepted by TMGA1-non but is rejected by TMGA1-con, because the latter considers both A_{OS} of (f_1, f_2) and (f_2, f_1) , and finds the second order to be unacceptable. In TMGA2-con, this is done implicitly. To see why, let us reconsider the above process started with f_1 . After checking α and β for f_1 , TMGA2-con must also make the optional checks for f_1 's neighbors. In this case, no f_i is the neighbor of f_1 's destination node, and thus no update for its β is needed. On the other hand, f_2 is the neighbor of f_1 's source node. However, f_2 currently has no established traffic and thus has no need to change its γ . With the unchanged values, the optional verification for f_2 is easily passed, in addition to the primary verification for f_1 . Consequently, TMGA2-con may go to check the next, that is, f_2 , as TMGA2-non would do.

As expected, TMGA2-con first makes the primary update for f_2 , which keeps $\beta(f_2) = 0$ and changes $\gamma(f_2)$ to be 1, then it makes the optional update and finds that f_1 is the neighbor

of f_2 's destination node and has 1-stream traffic established before. Thus, it changes $\beta(f_1)$ to be 1. Meanwhile, it finds no neighbor of f_2 's source node, and thus it changes nothing. However, since the optional update changes at least one value ($\beta(f_1)$), its optional verification may produce nontrivial results. In fact, it has $\alpha(f_1) + \beta(f_1) (= 2) \not\leq K_m(f_1) (= 2)$, indicating that $T_M = (1, 1)$ is not valid. Clearly, this example shows that while $A_O = (f_1, f_2)$ is examined by the primary update and verification, $A_O = (f_2, f_1)$ is also verified by the optional counterpart in TMGA2-con.

For the time complexity of TMGA2, we note that the time complexity for a single C_o is $O(\mathcal{L}^2 \cdot e + \mathcal{L} \cdot e^3)$, where e denotes the number of edges (flows) in the C_o , that is, $e = |E_{C_o}|$. Then, considering $p \leq \xi + 1$ graph components (C_o s), and denoting by \tilde{e} the maximal e in the network, that is, $\tilde{e} = \max_e \{e = |E_{C_o}|\}$, we can have the polynomial time complexity for TMGA2 as $O(p \cdot \mathcal{L}^2 \cdot \tilde{e} + p \cdot \mathcal{L} \cdot \tilde{e}^3)$.

4. Cross-Layer Schemes

Now, given a network G with MIMO links, the source and destination nodes of K end-to-end communication sessions, and the scheduling matrix Γ obtained previously, in this section we aim to find a rate allocation \mathbf{r} specifying the rate r_k for each session k , along with a flow allocation vector \mathbf{f}^k specifying the amount of traffic f_e^k of session k routed through link e , and a transmission schedule vector \mathbf{p} specifying time fraction p_{im} for each transmission mode T_M . More precisely, we want to solve the following optimization problems.

Definition 1. The Maximum throughput Rate Allocation (MRA) problem seeks a feasible rate allocation vector $\mathbf{r} = [r_1, r_2, \dots, r_K]$, along with a feasible flow allocation vector and a feasible transmission schedule vector such that the throughput $\sum_{k=1}^K r_k$ is maximized.

Definition 2. The Proportional fair Rate Allocation (PRA) problem seeks a feasible rate allocation vector \mathbf{r} , along with a feasible flow allocation vector and a feasible transmission schedule vector such that the utility function $\sum_{k=1}^K \log(r_k)$ is maximized.

Definition 3. The Weighted fair Rate Allocation (WRA) problem seeks a feasible rate allocation vector \mathbf{r} , along with a feasible flow allocation vector and a feasible transmission schedule vector such that $r_i/\psi_i = r_j/\psi_j, 1 \leq i \neq j \leq K$, where ψ_i denotes the positive weight of session i , with the assumption of $1 = \psi_1 > \psi_2, \dots, > \psi_K > 0$, and the throughput $\sum_{k=1}^K r_k$ is maximized.

For these problems, we propose our cross-layer schemes with the same basic steps. First, we identify all possible transmission modes or a subset of transmission modes by means of TMGA1 or TMGA2 given previously. Second, we formulate the problems as Linear Programming problems (LP)s and Convex Programming problems (CP)s based on the transmission modes found in above. More precisely, we have the following.

Linear Programming 1 (LP1): MRA

Given.

- (i) G , node topology graph of the MIMO network.
- (ii) SS , set of session source nodes, where s_k is the source node of session k .
- (iii) SD , set of session destination nodes, where d_k is the destination node of session k .
- (iv) Γ , scheduling matrix.

Variables. (i) \mathbf{r} , rate allocation vector, where r_k is the rate allocated for session k , $\forall k$.

(ii) \mathbf{f}^k , flow allocation vector for session k , where f_e^k is the session k 's traffic routed through link e .

(iii) \mathbf{p} , transmission schedule vector, where p_{t_m} is the time fraction for the transmission mode T_M .

Optimize. (i) Maximize the total throughput of sessions

$$\max \left(\sum_{k=1}^K r_k \right). \quad (5)$$

Constraint. (i) Flow conservation for source nodes

$$\sum_{e \in E_{s_k}^{\text{out}}} f_e^k - \sum_{e \in E_{s_k}^{\text{in}}} f_e^k = r_k, \quad 1 \leq k \leq K, \quad (6)$$

where $E_{s_k}^{\text{out}}$ ($E_{s_k}^{\text{in}}$) denotes the set of outgoing (incoming) edges of source node s_k .

(ii) Flow conservation for intermediate nodes

$$\sum_{e \in E_v^{\text{out}}} f_e^k - \sum_{e \in E_v^{\text{in}}} f_e^k = 0, \quad 1 \leq k \leq K, \quad \forall v \in V \setminus \{s_k, d_k\}, \quad (7)$$

where E_v^{out} (E_v^{in}) denotes the set of outgoing (incoming) edges of node $v \in V \setminus \{s_k, d_k\}$.

(iii) Bandwidth conservation

$$\sum_{k=1}^K f_e^k \leq \sum_{T_M: \Gamma_{T_M, f_i} \neq 0} p_{t_m} \cdot c_{f_i}, \quad \forall e \in E, \quad (8)$$

where c_{f_i} is the link capacity or rate of Γ_{T_M, f_i} .

(iv) Scheduling constraint

$$\sum_{t=1}^{t_\Gamma} p_{t_m} = 1. \quad (9)$$

(v) Flow rate validity:

$$f_e^k \geq 0, \quad 1 \leq k \leq K, \quad \forall e \in E. \quad (10)$$

(vi) Scheduling validity

$$p_{t_m} \geq 0, \quad 1 \leq t_m \leq t_\Gamma. \quad (11)$$

(vii) Session rate validity

$$r_k \geq 0, \quad 1 \leq k \leq K. \quad (12)$$

Remark 1. (i) Constraint (6) ensures that the net amount of traffic going out of the source node of a session is equal to that of the end-to-end session rate.

(ii) Constraint (7) ensures that the amount of traffic of a session entering any intermediate node is equal to that existing the intermediate node.

(iii) Constraint (8) ensures that the total traffic on a link is no more than the average link transmission rate.

(iv) Constraint (9) ensures that the summation of all elements in a transmission schedule vector is equal to 1.

(v) Constraints involving f_e^k imply that a session k can be routed through different links, es . That is, a session can go through several different routes towards its destination, which is called traffic splittable.

Linear Programming 2 (LP2): WRA

We have

$$\max \left(\sum_{k=1}^K r_k \right) \quad (13)$$

subject to the constraints (6)–(12), and

$$\frac{r_i}{\psi_i} - \frac{r_j}{\psi_j} = 0, \quad 1 \leq i \neq j \leq K \quad (14)$$

When compared with the objective of LP1 that only maximizes the network throughput and involves no consideration for fairness, LP2 has the extra constraint (14) for the weighted fairness among the session traffics. That is, LP2 aims to maximize the throughput while preserving the weighted fair shares in the sessions.

On the other hand, the PRA problem can be formulated as a convex program because it has the same linear constraints as the MRA problem and the objective is to maximize a concave utility function. That is,

Convex Programming 1 (CP1): PRA

We have

$$\max \left(\sum_{k=1}^K \log(r_k) \right) \quad (15)$$

subject to the constraints (6) to (12).

There are efficient algorithms for solving LPs and CPs [14, 15]. In our experiments, we use MATLAB to solve the LPs, and its CVX package [16] to solve the CPs. Their results are given in the following section.

5. Experiment Results

In this section, we report on simulation experiments made in order to verify the cross-layer schemes designed previously. To this end, different sets of experiments are conducted to exhibit their distinct performances on different network topologies frequently used. In addition, we take into account that throughput is in fact affected by link capability or rate.

Thus, to focus on the schemes' correctness and compare their performance, we assume that each antenna (DOF) in the MIMO system has the same capability (of 1). The rate allocated to each session, r_k , and the system throughput, $\sum_{k=1}^K r_k$, are normalized by the capability to provide their values independent of a certain system.

5.1. Wireless Backhaul Network. A wireless backhaul network (WBN) is considered as a collection of access points (APs), along with the uplink (to the Internet) and downlink (from the Internet) demands for each AP. The MAC layer adopted is usually assumed to schedule data to multiple receivers across timeslots using a TDMA-based scheme, which complies with our scenarios. For the network, we consider only uplink traffics conveyed with a common wireless channel shared by the MIMO links. In addition, we consider also that access traffic from the users to their respective APs is transmitted in a separate frequency band, and does not interfere with the wireless backhaul traffic considered here. For WBNs, the cross-layer schemes can be used to schedule the MIMO links without interference, and to maximize the system throughput according to the traffic demands from APs.

5.1.1. Topology 1. Let's re-examine the topology in Figure 2, and regard it as a backhaul network as follows. That is, $N = \{1, 2, 3, 4\}$ now represents the set of APs, each with DOF of 2, and node 1 denotes the AP connecting to the Internet, which is usually referred to as *Transit Access Point* (TAP). With these APs, $f_1 = \overrightarrow{(2, 1)}$, $f_2 = \overrightarrow{(3, 2)}$, and $f_3 = \overrightarrow{(4, 3)}$ are so conducted to compose the flow set F , reasonably representing that all uplink traffics are destined to the wired Internet. Clearly, every AP has its own traffic toward TAP, and thus the three sessions involved would be $(s_1 = 2, d_1 = 1)$, $(s_2 = 3, d_2 = 1)$, and $(s_3 = 4, d_3 = 1)$ (where (s_k, d_k) denotes the source-destination pair of session k). Given the above, it is easy to derive G_{strong} showing that f_1 and f_2 strongly contend with each other, and so do f_2 and f_3 . Accordingly, with the coloring algorithm, they are divided into two flow components, $C_1 = \{f_1, f_3\}$ and $C_2 = \{f_2\}$, containing only weak contentions. For these components, TMGA1 and TMGA2 both can easily produce $\Gamma_1 = \{\{0, 2\}, \{1, 1\}, \{2, 0\}\}$ and $\Gamma_2 = \{\{2\}\}$. After padding its T_M 's with zeros for the flows not in the component, respectively, the two Γ_o 's are merged to be the complete scheduling matrix $\Gamma = \{\{0, 0, 0\}, \{0, 0, 2\}, \{1, 0, 1\}, \{2, 0, 0\}, \{0, 2, 0\}\}$, wherein an empty $T_M = \{0, 0, 0\}$ is added as the default element for scheduling.

Now, with a DOF to represent a unit of transmission capability, LP1 produces $\{f_e^k\} = \{f_1^1 = 2, f_2^1 = 0, f_3^1 = 0, f_1^2 = 0, f_2^2 = 0, f_3^2 = 0, f_1^3 = 0, f_2^3 = 0, f_3^3 = 0\}$. That is, only by routing session 1 through $\{f_1\}$ and sacrificing all other sessions (with $r_2 = 0$ and $r_3 = 0$), the system can have the maximum system throughput 2. In addition, it produces $\{p_{i_m}\} = \{0, 0, 0, 1, 0\}$, saying that only $T_M = \{2, 0, 0\}$ should be scheduled in the MIMO network. In other words, the system throughput is dominated by the APs closet to TAP. The unfairness problem has also been reported in [17, 18]. However, unlike the

previous works, we address here joint rate control, routing, and scheduling for the MIMO-based wireless backhaul networks with a scheduling-based (TDMA-based) MAC. To be specific, with LP2 we can achieve the weighted fairness among the sessions while maximizing the aggregated system throughput. More precisely, LP2 produces, for example, $\{r\} = \{r_1 = 0.7692, r_2 = 0.3846, r_3 = 0.1538\}$, exactly complying with the given weights $\{\psi\} = \{\psi_1 = 1, \psi_2 = 0.5, \psi_3 = 0.2\}$. The corresponding routes are constituted by $\{f_e^k\} = \{f_1^1 = 0.7692, f_2^1 = 0, f_3^1 = 0, f_1^2 = 0.3846, f_2^2 = 0.3846, f_3^2 = 0, f_1^3 = 0.1538, f_2^3 = 0.1538, f_3^3 = 0.1538\}$. For the requirements on link capability, a TDMA-based MAC over the MIMO PHY is scheduled with Γ and $\{p_{i_m}\} = \{0, 0.01, 0.1338, 0.5869, 0.2692\}$. Accordingly, the link capabilities $\{p_{i_m}\}_{1 \times 5} \cdot \Gamma_{5 \times 3} = \{1.3076, 0.5384, 0.1538\}$ can fulfill the requirement of f_1 ($0.7692 + 0.3846 + 0.1538 = 1.3076$), that of f_2 ($0 + 0.3846 + 0.1538 = 0.5384$), and that of f_3 ($0 + 0 + 0.1538 = 0.1538$). Finally, for solving the PRA problem, CP1 produces $\{r\} = \{r_1 = 0.\bar{6}, r_2 = 0.\bar{3}, r_3 = 0.\bar{2}\}$ and improves the fairness problem when compared with LP1. However, it lacks the capability of achieving weighted fairness, and it would inevitably sacrifice the system throughput as LP2 may do.

5.1.2. Topology 2. Let us now consider the example in Figure 3. In principle, it can be also regarded as a wireless backhaul network with node 1 as TAP and other nodes as APs. With the more complex topology, our aim is to show how the cross-layer schemes can find multiple routes for a session to fulfill their specific maximization goals. To this end, three sessions $(s_1 = 6, d_1 = 1)$, $(s_2 = 7, d_2 = 1)$, and $(s_3 = 8, d_3 = 1)$ are conducted for the leaf nodes (6, 7, and 8). Given that, LP1 produces $\{r\} = \{r_1 = 0.3781, r_2 = 0.3781, r_3 = 0.5771\}$. To support these session rates, a single route to TAP (node 1) is allocated to the first two sessions, respectively. That is, the route for the first session is $f_5 \rightarrow f_3 \rightarrow f_1$ and that for the second is $f_6 \rightarrow f_4 \rightarrow f_2$, in which every single flow contributes the data rate of 0.3781 to its session. On the other hand, LP1 finds two routes for the third session: $f_7 \rightarrow f_3 \rightarrow f_1$ and $f_8 \rightarrow f_4 \rightarrow f_2$. In what mentioned above, each flow provides its rate of 0.2885. Then, by combining the two routes, it can support $r_3 = 0.5771$. Similarly, by setting $\psi_i = 1, \forall i$, LP2 can give each session i the same rate allocation $r_i = 0.\bar{4}$, with the same routes obtained in what mentioned above. Finally, CP1 also finds the same equal rate allocation as LP2, and thus, it has the same implication on the fairness in this case.

5.2. Wireless Mesh Network. In this set of experiments, we randomly generate wireless mesh networks (WMNs) with n nodes located in a $1000 \times 1000 \text{m}^2$ region. The transmission range (R_T) and the corresponding interference range (R_I) are set to 400 m and 600 m, respectively. In addition, these networks are so conducted to ensure their connectivity of the resulting topologies. Then, the cross-layer schemes are examined on these networks to provide their performances on rate allocated to each session, throughput, and weighted fairness. In particular, the sessions are sorted in

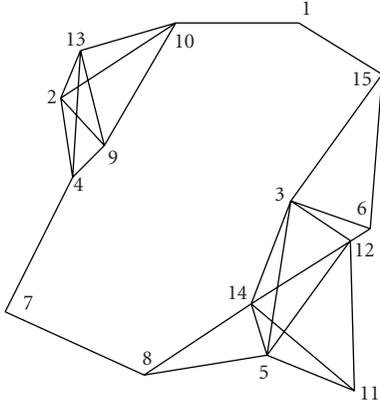


FIGURE 4: Random topology for the experiment.

the increasing order of their rate values to clearly show their performance differences.

As the first part of this experiment set, we examine our cross-layer schemes on a network with 15 nodes, as shown in Figure 4. In the network, we equip each node with 2 antennas and generate 10 communication sessions with their source and destination nodes to be randomly generated so that no two sessions have the same sources and destinations. Note that, in this example, the maximum number of flows s_j in a C_o is 18, and clearly, the number of permutations for the $s_j = 18$ elements (flows), that is, its o_j , is generally an intractable value for computation.

A similar situation also happens to the problem of finding Maximal Independent Sets (MISs). Although the algorithm in [19] can be used to find all MISs, it is still intractable that the number of MISs will grow exponentially with the increase of the graph size, and this is the reason why we need to develop TMGA2. In the experiment, TMGA2 is used to generate transmission modes with $\mathcal{L} = 1$ and $\mathcal{L} = 10$, respectively. The corresponding rate allocation results are shown in Figure 5. As expected, the MRA scheme can give certain sessions the highest r_i 's, but it also results in a severe unfairness on the rate allocation. This can be seen in the figure that the first several sessions sorted have their rates equal to zero but the latter ones obtain very high values. On the contrary, the WRA scheme performs best in terms of fairness. In fact, with $\psi_i = 1, \forall i$, the rate allocated to each session i is the same. Note that the WRA scheme has the capability to achieve arbitrary weighted fairness among the sessions. However, in the experiments, we simply show the equal weight results. Between the two extremes, the PRA scheme is much better than the MRA scheme on the fairness, but it cannot achieve an absolutely even distribution, and certainly it cannot achieve the weighted fairness. Finally, it could be seen that with the conservative generating approach, labeled with "(con)," the three schemes tend to have their rate allocations lower than those with the nonconservative counterparts, labeled with "(non)." This trend is further verified in the following experiments.

In the second part of the experiments, we aim to compare the conservative and the nonconservative transmission mode

generating approaches in TMGA1 and TMGA2 and to evaluate the efficiency of TMGA2. In fact, TMGA1 is used here as a benchmark because it can generate all possible transmission modes (T_M s) and can give the cross-layer schemes the most complete scheduling matrix (Γ). However, to be numerically tractable for TMGA1, we run the experiments on a smaller network that has 6 nodes and 6 sessions with the same setting given previously. In this network, each node is randomly equipped with 1 or 2 antennas, and three numbers of iteration limit, $\mathcal{L} = 1$, $\mathcal{L} = 2$, and $\mathcal{L} = 10$, are examined for TMGA2. In addition, to quantitatively analyze the fairness performances for these schemes, we let S_i denote the throughput of session i , and let ψ_i denote the associated weight, and we use these parameters to obtain the *fairness index* in [20] as follow:

$$F_s = \frac{\mu(S_i/\psi_i)}{\mu(S_i/\psi_i) + \alpha(S_i/\psi_i)}. \quad (16)$$

The results are shown in Figure 6, wherein "All" denotes that for TMGA1. With this figure, we summarize our observations from the following two aspects. First, from the throughput aspect in Figure 6(a), we can see that the MRA scheme achieves the highest values in spite of \mathcal{L} . That is to say, although a larger \mathcal{L} may lead to a larger number of T_M 's in Γ , it does not affect MRA's throughput performance here. This is because MPA could always maximize a single session while sacrificing all other sessions with Γ s obtained, as indicated in Section 5.1. Similarly, \mathcal{L} does not affect much PRA and WRA with conservative Γ 's. However, if given nonconservative Γ 's, it has stronger impacts on the two schemes. This is because a larger Γ resulted may open more opportunities for these schemes to optimize their target functions, and the nonconservative Γ 's found would have their sizes larger than the conservative counterparts. Apart from these differences, we note also that the Γ 's resulted from $\mathcal{L} = 1$ or 2 would be enough for most of the schemes. Even so, one may still expect a nonconservative Γ for higher throughputs, despite \mathcal{L} . However, the higher throughputs are obtained with the costs of providing a more strict admission control that can correctly admit its sessions with the admission orders (A_O 's) recorded to realize the T_M 's in a given nonconservative Γ . On the other hand, with a conservative Γ , every possible A_O would be already considered for a T_M , and thus no such overheads would be involved.

As the second aspect, we consider the fairness results in Figure 6(b). Clearly, it is shown that the WRA scheme perfectly achieves the weighted fairness of $\psi_i = 1, \forall i$, and its fairness index values are all of 1 in spite of \mathcal{L} . On the other hand, the PRA scheme has its values ranging from 0.7 to 0.9, and the MRA scheme does not exceed 0.52. In addition, similar trends for the throughput results also hold here. For example, the nonconservative Γ 's usually provide better performances than the conservative counterparts. However, we note that a higher \mathcal{L} improves most the PRA scheme on the throughput, but it improves most the MRA scheme on the fairness. Thus, it is suggested that one may choose \mathcal{L} depending on the performance metric most concerned.

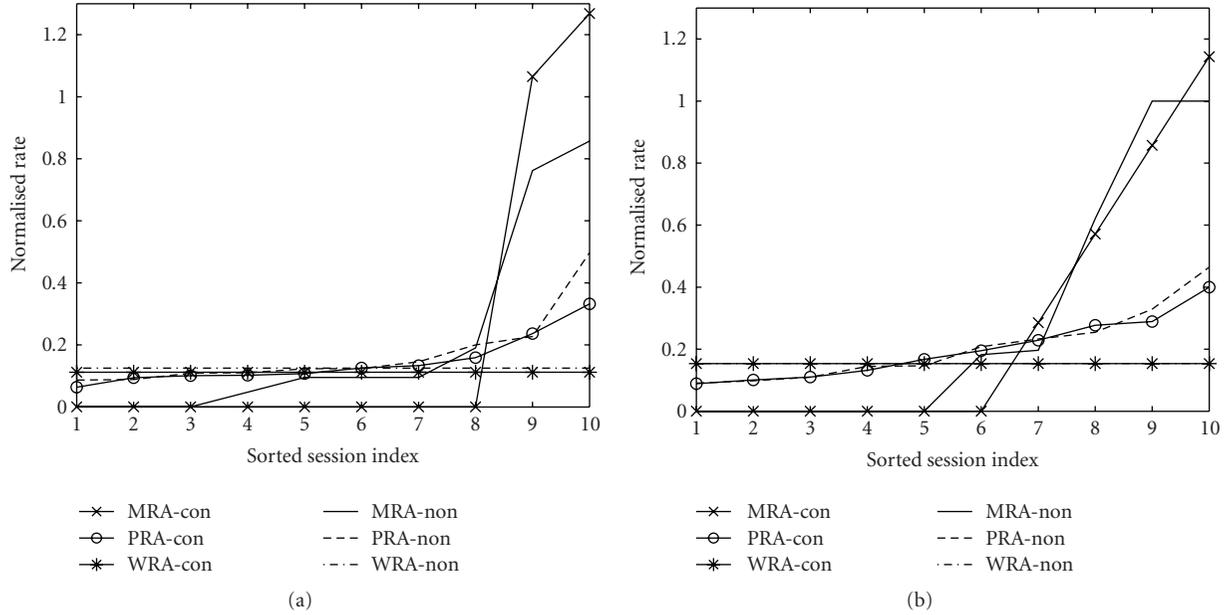


FIGURE 5: Rate allocation results for the experimented sessions: (a) $\mathcal{L} = 1$, and (b) $\mathcal{L} = 10$.

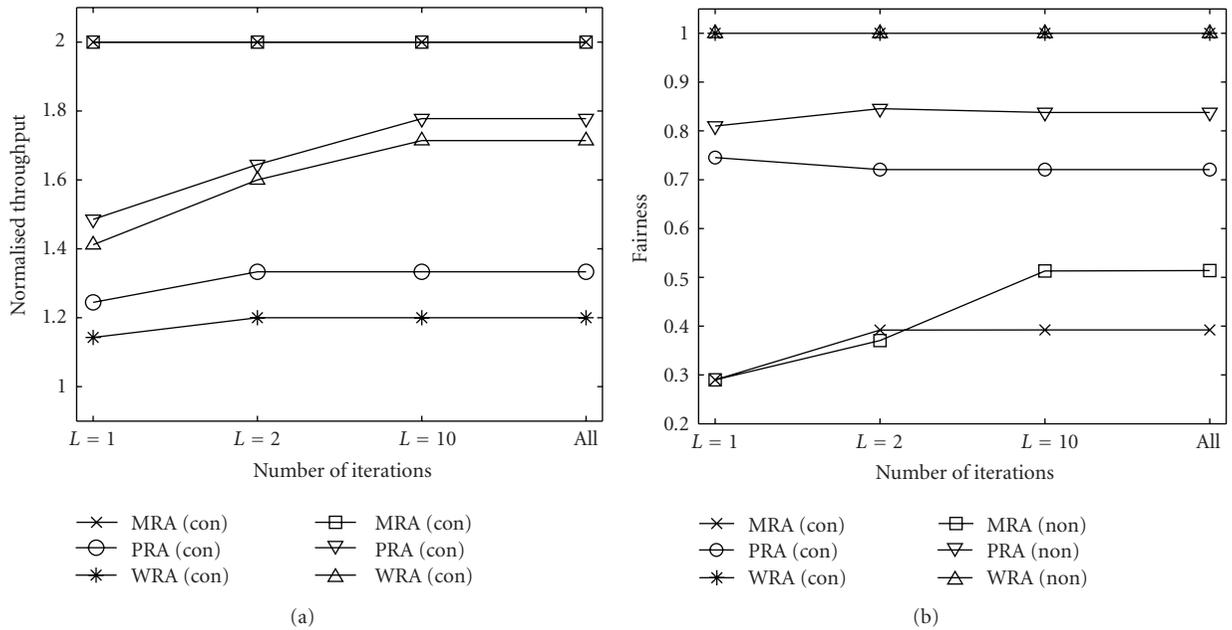


FIGURE 6: Throughput and fairness results for the experimented sessions: (a) throughput and (b) fairness.

Nevertheless, in general, $\mathcal{L} = 1$ or 2 could satisfy these schemes with low time complexity, as indicated previously.

As the final part of the experiments, we examine TMGA2 with more topologies to know its effectiveness. To be specific, we let $\mathcal{L} = 10$ and conduct two sets of experiments that are numerically tractable for this aim. Specifically, by randomly deploying 6 nodes and 6 sessions in the network, we conduct 30 different topologies as the first set of experiments. Note that although this set has the same numbers as the

above, with the variation it actually results in very different topologies and traffic conditions to be considered. With the same way, we conduct another 30 topologies as the second set of experiments, but now there are 10 nodes and 8 sessions randomly deployed to reasonably represent the different numbers of nodes and sessions that may involve.

The throughput and fairness results for the first set are given in Figures 7(a) and 7(b), respectively. From these figures, we can easily see that the two performance

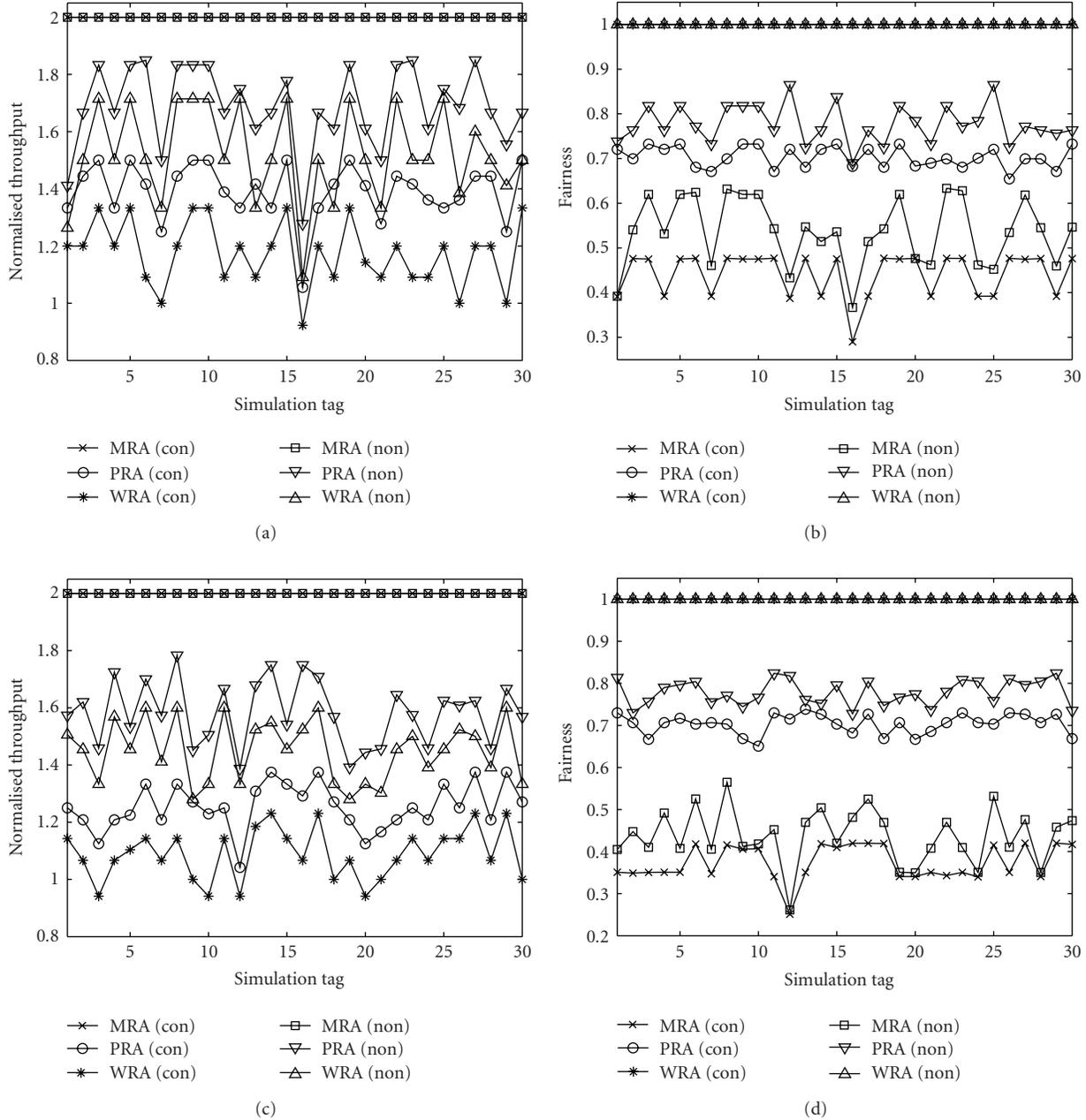


FIGURE 7: Throughput and fairness results for the different network scenarios: (a) throughput for the 6-node topologies, (b) fairness for the 6-node topologies, (c) throughput for the 10-node topologies, and (d) fairness for the 10-node topologies.

metrics are significantly varied by the different topologies, as expected. However, for each single topology, the relative relationships among the results of these schemes have the same trend as Figure 6 has shown. Clearly, this trend can be also observed in Figures 7(c) and 7(d) for the results of the second set. With the above, it could be said that the proposed schemes in TMGA2 are able to generate the transmission modes that can efficiently fulfill the design aim of solving the MRA, PRA, and WRA problems in spite of the topologies with the different numbers of nodes and sessions.

6. Conclusion

In this work, we take into account the fact that for fully realizing the potential of MIMO technology, higher layer must be designed to be cognizant of the MIMO link capability. To this end, instead of simply translating the achievable gain for individual MIMO links into end-to-end gain in the network, we present a mathematical framework that can express the cross-layer gain on throughput as a function of network routing, link scheduling, and stream control in the presence of interference. With that, we propose Transmission Mode

Generating Algorithms (TMGAs) to generate TDMA-based scheduling matrices and give our Linear Programming- (LP-) based and Convex Programming- (CP-) based schemes to maximize the network throughput, and to achieve certain fairness (such as weighted fairness, in particular) at the same time. The simulation experiments' results show that the proposed schemes are all capable on achieving our design aims, and every scheme has its own unique performance benefit and tradeoff between throughput and fairness.

References

- [1] S. K. Jayaweera and H. V. Poor, "Capacity of multiple-antenna systems with both receiver and transmitter channel state information," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2697–2709, 2003.
- [2] R. S. Blum, "MIMO capacity with interference," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 5, pp. 793–801, 2003.
- [3] R. Narasimhan, "Spatial multiplexing with transmit antenna and constellation selection for correlated MIMO fading channels," *IEEE Transactions on Signal Processing*, vol. 51, no. 11, pp. 2829–2838, 2003.
- [4] M. Hu and J. Zhang, "MIMO ad hoc networks: medium access control, saturation throughput, and optimal hop distance," *Journal of Communications and Networks*, vol. 6, no. 4, pp. 317–330, 2004.
- [5] K. Sundaresan, R. Sivakumar, M. A. Ingram, and T.-Y. Chang, "Medium access control in ad hoc networks with MIMO links: optimization considerations and algorithms," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 350–365, 2004.
- [6] L. Chen, S. H. Low, and J. C. Doyle, "Joint congestion control and media access control design for ad hoc wireless networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 3, pp. 2212–2222, Miami, Fla, USA, March 2005.
- [7] X. Wang and K. Kar, "Cross-layer rate control for end-to-end proportional fairness in wireless networks with random access," in *Proceedings of the 6th International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '05)*, pp. 157–168, Urbana-Champaign, Ill, USA, May 2005.
- [8] J. Tang, G. Xue, C. Chandler, and W. Zhang, "Link scheduling with power control for throughput enhancement in multihop wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 3, pp. 733–742, 2006.
- [9] R. Bhatia and L. Li, "Throughput optimization of wireless mesh networks with MIMO links," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 2326–2330, Anchorage, Alaska, USA, May 2007.
- [10] B. Mumey, J. Tang, and T. Hahn, "Joint stream control and scheduling in multihop wireless networks with MIMO links," in *Proceedings of IEEE International Conference on Communications (ICC '08)*, pp. 2921–2925, Beijing, China, May 2008.
- [11] S. Ramanathan, "Unified framework and algorithm for channel assignment in wireless networks," *Wireless Networks*, vol. 5, no. 2, pp. 81–94, 1999.
- [12] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MOBICOM '03)*, pp. 66–80, San Diego, Calif, USA, September 2003.
- [13] M. Kodialam and T. Nandagopal, "Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MOBICOM '03)*, pp. 42–54, San Diego, Calif, USA, September 2003.
- [14] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Networks Flows*, John Wiley & Sons, New York, NY, USA, 3rd edition, 2005.
- [15] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [16] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab Software for Disciplined Convex Programming," 2008.
- [17] V. Gambiroza, B. Sadeghi, and E. W. Knightly, "End-to-end performance and fairness in multihop wireless backhaul networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM '04)*, pp. 287–301, Philadelphia, Pa, USA, September 2004.
- [18] J. Lee, W. Liao, and M. C. Chen, "An incentive-based fairness mechanism for multi-hop wireless backhaul networks with selfish nodes," *IEEE Transactions on Wireless Communications*, vol. 7, no. 2, pp. 697–704, 2008.
- [19] D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou, "On generating all maximal independent sets," *Information Processing Letters*, vol. 27, no. 3, pp. 119–123, 1988.
- [20] D. Qiao and K. G. Shin, "Achieving efficient channel utilization and weighted fairness for data communications in IEEE 802.11 WLAN under the DCF," in *Proceedings of the 10th International Workshop on Quality of Service (IWQoS '02)*, pp. 227–236, 2002.