Exploiting Spatial Reuse in Multi-hop MIMO Networks: Models, Algorithms, and Evaluation

Deke Guo*, Yuan He†, Yunhao Liu†, Panlong Yang‡,and Xiangyang Li §
* College of Information Systems and Management, National University of Defense Technology, Changsha, China
† Department of Computer Science and Engineering, Hong Kong University of Science and Technology
‡ Department of Computer Science and Technology, Nanjing University, China
§ Department of Computer Science, Illinois Institute of Technology, Chicago, IL, 60616

***Abstract*— MIMO has great potential for enhancing the throughput of multi-hop wireless networks via spatial multiplexing or spatial reuse techniques. Spatial reuse outperforms spatial multiplexing in terms of potential network throughput and is effective in a wider range of wireless contexts than spatial multiplexing. The network throughput when previous approaches of spatial reuse are employed, however, is still considerably constrained due to the limited concurrency among the data streams transmitted in the same time slot and vicinity. In this paper, we address the issue of MIMO link scheduling to maximize the spatial reuse gain and thus network throughput. We propose two interference suppression models, transmitter oriented (TOIS) and receiver oriented (ROIS), based on which we design both centralized and distributed link scheduling algorithms to fully exploit spatial reuse in multi-hop MIMO networks. Further, we address the traffic-aware link scheduling problem by injecting non-uniform traffic load into the network. Through theoretical analysis and comprehensive performance evaluation, we achieve the following results: (1) Link scheduling based on TOIS achieves equivalent network throughput as stream control, while the latter is the best approach prior to our work. (2) Link scheduling based on ROIS achieves higher network throughput than that based on TOIS, irrespective of the interference range, the number of antennas, and the average hop length of data flows. (3) The traffic-aware link scheduling is necessary and beneficial, which realizes near-optimal network throughput.**

I. Introduction

Recently MIMO (Multiple-Input-Multiple-Output) has often been referred to as an affordable solution to increase the network capacity of wireless networks. MIMO enhances the potential network throughput by spatial multiplexing or spatial reuse. Spatial multiplexing transmits independent and separately encoded data streams over a MIMO channel. Spatial reuse of the spectrum allows multiple simultaneous transmissions in the same vicinity with interference suppression.

Many efforts have been made to deal with single-user MIMO communication [1]–[4]. Multiple antennas yield degree-of-freedom gain. The additional degree-of-freedom can be fully exploited by spatial multiplexing multiple independent data streams onto one MIMO channel, hence resulting in increased network capacity without extra bandwidth or power cost. Recently, the co-channel interference in multi-user MIMO communications [5]–[7] draw considerable attention from the community, such as broadcast channel (BC) [8] and multiple-access channel (MAC) [5].

In multi-hop MIMO networks, MIMO enhances the potential network throughput via *spatial multiplexing* by achieving high data rates or *spatial reuse* of the spectrum by allowing multiple simultaneous communications in the same vicinity. In the case of spatial reuse, each MIMO link only employs partial degree-of-freedom to transmit and receive data streams. Basically there are three approaches. The first one is spatial reuse with *stream control* [9]–[16]. The best degree-of-freedoms are selected for data transmissions, while the other degree-of-freedoms at the receiver are used to suppress interfering data streams. In the second approach, referred to as spatial reuse with *non-stream control*, degree-of-freedoms for transmission are randomly selected while the receiver uses the other degree-of-freedoms in the same way as the first approach [17], [18]. In the third approach referred to as spatial reuse with *interference avoidance* [17], the transmitter selects appropriate degree-of-freedom to transmit data streams such that the remainder degree-of-freedoms can *null* its data streams at undesired nearby receivers.

Many existing work have revealed that stream control increases the total link throughput by 20-65% for a set of *mutually interfering* links [9]–[11]. In [11], [12], the authors demonstrate that stream control improves the overall throughput compared to the TDMA-based spatial multiplexing[1]. In [13], [14], the authors point out that stream control with optimal antenna selection is an attractive alternative to the TDMA-based spatial multiplexing. Moreover, stream control is effective in a wide range of wireless environments, while spatial multiplexing exhibits its benefits only under rich scattering or strong multi-path conditions associated with urban and indoor applications. The network throughput when spatial reuse is employed, however, is still considerably constrained due to the interference effect and limited concurrency among the data streams transmitted in the same time slot and vicinity.

Bearing these points in mind, we address the issue of link scheduling to fully exploit spatial reuse in multi-hop MIMO networks, and to maximize the potential network throughput. The challenges to this work include how to deal with the interference effect so we can increase the number of transmission streams in the same time slot, frequency slot,

[1]For ease of presentation, we use the term "stream control" to represent "spatial reuse with stream control" in the rest of this paper, and explain in Section II-C why stream control can outperform spatial multiplexing.

and vicinity. Our studies show that while the stream control outperforms TDMA-based spatial multiplexing, spatial reuse is not fully exploited. Even after the stream control mechanism is realized, there still exist large numbers of additional data streams which can be transmitted concurrently in certain time slots and vicinities. Moreover, the gain of spatial reuse can be further exploited by interference avoidance on the basis of stream control. To address this problem, we model the distributed spatial reuse mechanism as a link scheduling problem under more accurate interference constraint, which incorporates both spatial reuse with *stream control* and spatial reuse with *interference avoidance*. We study how to achieve maximum network throughput through efficient centralized and distributed link scheduling. The main contributions of this paper are as follows.

1) We propose the transmitter oriented (TOIS) and receiver oriented suppression (ROIS) models to deal with interference, including the necessary and sufficient conditions for these two models. Each link that has data to transmit is scheduled by TOIS, or by ROIS if TOIS cannot assign the smallest time slot to it.
2) We design both centralized and distributed link scheduling algorithms to optimize the overall network throughput. Further,we design traffic-aware link scheduling algorithms to meet the traffic demands. The improved algorithms can significantly enhance the overall throughput.
3) Through comprehensive simulations, we show that irrespective of interference range, number of antennas, and average hop length of flows, the link scheduling based on ROIS achieves higher network throughput than that based on TOIS. Further, the traffic-aware link scheduling achieves the largest network throughput which is near-optimal.

The rest of this paper is organized as follows. In Section II we briefly describe the background and preliminaries of our work. Section III introduces the network model and interference suppression models used in this paper. The formulation of the link scheduling problem is presented as well. We elaborate the centralized and distributed link scheduling algorithms, and study how to schedule links when each link has a requirement of the least number of data streams in Section IV. Our simulation studies are reported in Section V. We review the related work in Section VI, and conclude this paper in Section VII.

II. Background and Preliminaries

In this section we briefly discuss single-user MIMO communication in Section II-A. The transceiver model used in this paper is introduced in Sections II-B. Then in Section II-C, we illustrate the basic idea of spatial reuse with co-channel interference.

A. Single-user MIMO communication

In rich scattering environments, a non-selective block-fading MIMO channel with m transmitting and n receiving omnidirectional antennas can be modeled by a $n{\times}m$ channel matrix H, where $m \leq n$ [2]. The $n-$dimensional signal at the output of the receiving antennas can be written as $y = Hx + a$ where x accounts for the $m-$dimensional transmitted symbols, and a denotes the $n \times 1$ additive white Gaussian noise vector. The real and imaginary parts of the noise on each receiving antenna are independent, each with a variance σ^2. The entries of H are independent with uniformly distributed phase and normalized Rayleigh distributed magnitude, modeling a Rayleigh fading channel. Let h_{ij} represent the gain of the channel from transmitting antenna j to receiving antenna i.

At the transmitter side, an encoder with a channel coding, interleaving, and symbol-mapping blocks is responsible for producing the input symbol vector, $x_1, ..., x_m$. The transmitter directly delivers the symbol vector to the receiver from respective antennas. At the receiver side, number of n signals are received, each signal y_i being a linear combination of the m transmitted symbols plus an additive noise where $1 \leq i \leq n$. That is

$$y_i = \sum\nolimits_{j=0}^{m} h_{ij} * x_j + n_i.$$

The objective of the receiver is to estimate x with $\bar{x}$ from the given y and H.

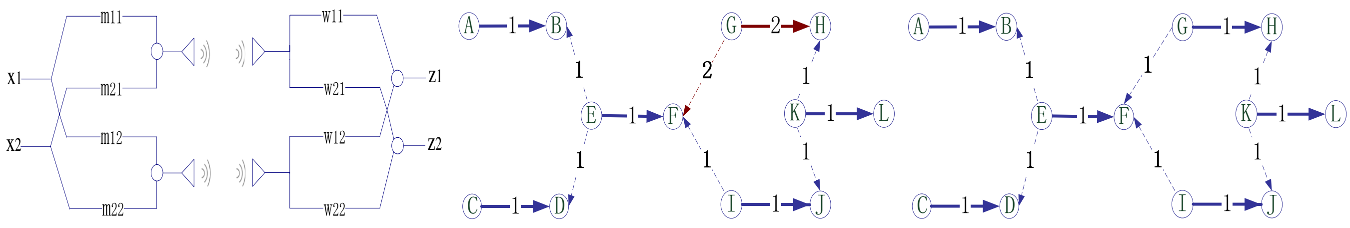
B. Transceiver model

Throughout this discussion, we focus on MIMO transceiver model that includes linear pre-processing and post-processing performed at the transmitter and receiver:

$$z = W(HMx + a)$$

where z is a data vector of dimension n, and the actual transmitted signals HMx is generated using an $m \times m$ pre-processing matrix M in which each column is referred to as a transmit weight vector for a symbol of x. As shown in Figure 1(a), number of $m = 2$ weighted copies of each symbol in x are produced and sent to the m transmit antennas, one on each antenna. A mixed signal of m weighted symbols is produced at each transmitting antenna. The m mixed signals are then delivered to an intended receiver through a MIMO channel H. The received vector of signals is $HMx + a$, and is converted into an estimation of the original vector signal x by a $n{\times}n$ post-processing matrix W in which each row is referred as a receive weight vector.

If H is perfectly known by the transmitter, a MIMO channel with m transmitting antennas and n receiving antennas can be decomposed into $\min\{m, n\}$ independent parallel channels [19]. In this case, channel capacity is achieved by using *linear precoding* known to be capacity-optimal [20] where M and W are chosen as the right and left singular vectors of H, respectively. If H is unknown to the transmitter, the ergodic capacity can be achieved by choosing $M{=}\alpha I$ [21]. In multi-hop MIMO networks, the benefit of spatial multiplexing scheme is achieved by a TDMA-based channel access model which ensures that no co-channel interference exists in any time slot, frequency slot, and vicinity.



(a) MIMO processing (b) TOIS model (c) ROIS model

Fig. 1. MIMO processing and illustrative examples of interference suppression models, where each transceiver is equipped with two antennas.

C. Spatial reuse with interference

In a multi-hop MIMO network, a receiver with m antennas should not observe more than m data streams in order to decode its desired streams correctly.

A MIMO channel can be decomposed into multiple parallel channels whose capacities are not equal, and have quite large disparities for moderate or low SNR [12]. As mentioned in Section I, three approaches can exploit the spatial reuse gain. Those approaches are different in the selection of sub-channel (degree-of-freedom) for transmission, and thus result in different throughput on the same MIMO link. By using the linear precoding technique [20], stream control selects the best channels for transmissions and yields the largest capacity among these approaches. On the other hand, if stream control and spatial multiplexing transmit same number of data streams during a same time period, stream control outperforms TDMA-based spatial multiplexing in terms of network throughput. Yet it is still not clear whether spatial reuse with non-stream control and interference avoidance generally outperform the TDMA-based spatial multiplexing. Therefore, researchers focus on realizing stream control in multi-hop MIMO networks during the past years [12], [15], [16]. For the same reason, we focus on how to fully exploit spatial reuse on the basis of stream control in multi-hop MIMO networks.

Here, we give an example to illustrate the basic idea of spatial reuse. As shown in Fig.2, each node has four antennas. Consider transmissions from node A to node B along link 1, and those from node C to node D along link 2. Since links 1 and 2 are close to each other, they encounter mutual interference. If TDMA-based spatial multiplexing is adopted, only one transmission is allowed to take place in any slot while the transmission may proceed with all the four streams. In the case of the spatial reuse with stream control or non-stream control, the two transmissions proceed simultaneously but the number of streams transmitted by each node is two. Although nodes B and D observe two additional interfering streams from nodes C and A, they are able to decode two desired streams from nodes A and C, respectively. Since stream control selects the best degree-of-freedoms for data transmissions, it obviously delivers higher throughput than TDMA-based spatial multiplexing although the same number of data streams are transmitted per time slot in both cases. Nevertheless, this is not always true for the non-stream control approach. Without careful selection of degree-of-freedoms, worse sub-channels might be chosen for data transmissions.

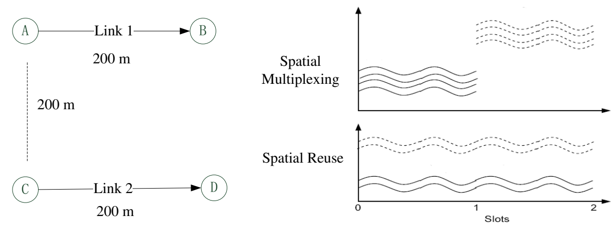


Fig. 2. Illustrative examples of the spatial multiplexing and spatial reuse.

As for spatial reuse with interference avoidance, node A uses stream control or non-stream control to deliver two data streams over link 1 since no streams are currently received by nearby nodes within A's interference range. Node C nulls its signals at interfered receiver B prior to transmitting its two data streams, namely directing signal energy towards the intended node D and minimizing interference to node B. Similar with stream control and non-stream control approaches, this approach transmits four data streams per time slot. The resulting network throughput, however, is less than that of stream control.

III. System model and assumptions

We start with a brief introduction to the network model used in this paper and propose two interference suppression models for multi-hop MIMO networks. We then model link scheduling problem and outline our approaches.

A. Network model

We consider a multi-hop wireless network with homogeneous terminals deployed in a rich scattering environment. Each terminal has m antennas, following the aforementioned transceiver model. The communication graph is modeled as a directed graph G=(V, L) with a vertex set V and a edge set L. A vertex of the graph corresponds to a terminal with m antennas. A directed edge (u, v) denotes a MIMO link l through which u can transmit to v directly and v can correctly decode the received signal from u. The vertex u and vertex v are referred to as the transmitter $t(l)$ and receiver $r(l)$ of l.

Each vertex u has an associated transmission range (decode range) denoted by $d_t(u)$. A necessary but not sufficient condition for a transceiver v to hear u is that v is within the Euclidean distance $d_t(u)$ from u. Edge $(u, v) \in L$ if and only if v is within the distance $d_t(u)$ from u. We assume that all vertices have the identical transmission power and thus identical transmission range. The transmission range can be enlarged when all antennas are used to achieve *array* and *diversity gains*, which will be discussed in our technical report [22]. Each vertex u has an interference range $d_i(u)$ such that vertex v is interfered by u only if v resides within a distance $d_i(u)$ of u. The interference ranges of all the nodes are also assumed to be identical, which are typically larger than the transmission ranges. Given a set of pairs of source and destination, the end-to-end flow for each pair of nodes can be discovered by existing routing protocols [23], thus determining the corresponding communication graph.

In this work we mainly consider multi-hop MIMO networks with homogeneous terminals. Yet it is worth noticing that the proposed interference suppression models and link scheduling algorithms can also be applied to other network models with more complex assumptions. For example, every terminal may have its own transmission power, transmission range, and interference range; it may have its own number of antennas and degree-of-freedoms.

B. Interference suppression model

Due to the *radio constraint*, a radio either transmits or receives, but cannot transmit and receive at the same time. When multiple separate streams reach several receivers, the receivers observe superposition signals of these streams and noise. Thus we propose the following two interference suppression models, which ensure that receivers still successfully decode desired data streams from the superposition signals.

Our first model is TOIS (transmitter oriented interference suppression) that exploits the gains of *stream control*. More specifically, each transmitter employs the linear precoding technology to pursue the highest data rates for data streams while does not intend to reduce interference to other receivers. That is, each receiver with m antennas can suppress interferences caused by undesired nearby transmitters while successfully receiving desired signals, as long as it does not observe more than m data streams. The receiver that observes more than m streams cannot decode its desired streams correctly. As shown in Fig.1(b), terminals B, D, and J can decode their desired streams from terminals A, C, and I at time slot 1, respectively. Terminal F, however, cannot decode its desired streams from terminal E if link l_{GH} is also active at time slot 1 such that it has two antennas while observes three streams.

Inspired by the third type of spatial reuse, namely interference avoidance, streams[2] that do not satisfy the TOIS model can be transmitted concurrently in some time slot and vicinities if their transmitters employ interference avoidance technique. That is, after realizing the TOIS model some potential transmitters can null their signals at those nearby receivers which have observed at least m streams while ensuring acceptable signal gains at desired receivers. We name this procedure including the TOIS as the ROIS (receiver oriented interference suppression) model. It appropriately combines two schemes of spatial reuse (stream control and interference avoidance) for enhancing the number of concurrent stream transmissions. As shown in Fig.1(c), using ROIS, terminal F correctly decodes its desired stream from terminal E even if link l_{GH} is active at the same time slot, as long as node G nulls its signals at terminal F. Note that terminals need not null their signals at nearby receivers that observe less than m streams because those receivers are already able to decode the desired streams.

[2]At least one of them makes an undesired receiver no longer satisfy the TOIS model.

C. Problem formulation

A feasible schedule of MIMO links describes the specific time slots at which packets are moved over the links. Our objective is to allocate each link $l \in L$ an interference-free transmission schedule such that the overall throughput of the network is maximized. Let $X_{l,t} \in \{0, 1, ...m\}$ be an indicating variable, $X_{l,t} = 0$ only if link l is inactive and no packet is sent over l at time slot t, and $X_{l,t} = i > 0$ if link l is active and i concurrent streams are sent over link l at time slot t ($0 < i \leq m$). We employ a periodic schedule with period length of T such that $X_{l,jT+t} = X_{l,(j+1)T+t}$ for any time slot t and integer j. For any two links u and v, link u interferes with link v only if the transmission on link u interferes with the reception on link v. However, this does not necessarily imply that link v also interferes with link u. Let I_l^+ denote the set of links whose receivers are interfered by the transmission on link l at the same time slot, and let I_l^- denote the set of links whose transmissions interfere with the reception on link l.

The objective of link scheduling is to find the shortest interference-free schedule if traffic load on links are unknown in advance. If we schedule all links within a period T in an interference-free model, at least one packet can be delivered over each communication link in every T time slots. Thus, $1/T$ is often used to estimate the *network throughput* with this schedule. The second scenario is that the average traffic load $\alpha(l)$ of each link is known in advance. We assume that each link l has a weight $w(l) = a(l)/c(l)$ which specifies the minimum number of time slots required in each period. $c(l)$ denotes a time-invariant capacity of link l over which only one stream is transmitted, and is determined by the transmission vectors and power allocation for the data stream. Our main focus is how to schedule the communication links in an interference-free manner such that the network throughput is maximized.

Let $N_i(l,t)$ and $N_u(l,t)$ denote the number of intended and unintended streams observed by the receiver of link l, respectively, and $N(l,t)$ denote the number of total streams observed by the receiver of link l with $N(l,t) = N_i(l,t) + N_u(l,t)$ at time slot t. According to the *radio constraint*, a necessary condition to schedule link l at time slot t is that

$$X^t_{l',t}{=}0,\ \forall l' \in \{l' | l' \in L, t(l){=}r(l') \text{ or } r(l){=}t(l')\}. \quad (1)$$

Note that whether a link can be active at a time slot depends on the interference suppression model, e.g., TOIS model or ROIS model, as well as the *radio constraint*. Thus, for a given link l and time slot t, $X_{l,t}$ might be different. Specifically, an active link in TOIS model is still active in ROIS model, and an inactive link in TOIS model might be active in ROIS model. To avoid confusion, we use $X^{ts}_{l,t}$ and $X^{rs}_{l,t}$ to denote the state of link l at time slot t in TOIS model and ROIS model, respectively. Note that $X_{l,t}^{ts}$ and $X_{l,t}^{rs}$ are two exclusive metrics, and the state of link l can be one of the following three possibilities. That is, $X_{l,t}^{ts} > 0$ shows that link l is active in TOIS model, $X_{l,t}^{rs} > 0$ means that link l is active in ROIS model, and $X_{l,t}^{rs} = 0$ denotes that link l is still inactive even in ROIS model. In TOIS model, we have

$$\begin{aligned} N_i(l,t) &= \sum\nolimits_{l' \in L, r(l')=r(l)} X_{l',t}^{ts} \\ N_u(l,t) &= \sum\nolimits_{l' \in I_l^-, r(l') \neq r(l)} X_{l',t}^{ts} \end{aligned}$$

In ROIS model, there is

$$\begin{aligned} N_i(l,t) &= \sum\nolimits_{l' \in L, r(l')=r(l)} X_{l',t}^{ts} + X_{l',t}^{rs} \\ N_u(l,t) &= \sum\nolimits_{l' \in I_l^-, r(l') \neq r(l)} X_{l',t}^{ts} + X_{l',t}^{rs} \end{aligned}$$

In TOIS model, a receiver with m antennas can decode all received streams together (at most m), or just decode those intended streams after suppressing all unintended streams as interferences. The sufficient and necessary condition of an interference-free schedule is that the *radio constraint* is satisfied and no more than m streams are observed by the receiver of each active link at any time slot. That is, a schedule S is *interference-free* only if the formulas (1) and (2) hold for each active link l.

$$N(l,t) \leq m, \forall t \in T. \tag{2}$$

In ROIS model, no matter how many additional streams a receiver observes after it has received m streams, the receiver can suppress those additional streams as interferences and decode the first m streams or only intended streams. The sufficient and necessary condition of an *interference-free* schedule in ROIS is that the schedule is *interference-free* under TOIS model and satisfies the following two conditions for each active link l scheduled in ROIS model. First, the total number of intended incoming streams occurred at those receivers, which have received at least m streams and stay within the interference range of the transmitter of link l, is less than m. Second, the number of streams observed by the receiver of link l should be less than m prior to the transmission of link l. That is, a schedule S is interference-free only if it is interference-free under TOIS model and the formula (3) holds for each additional active link l scheduled in ROIS model.

$$\begin{cases} \sum_{l' \in I_l^+, N(l',t) \geq m} N_i(l',t) < m \\ N(l,t) < m, \forall t \in T \end{cases} \tag{3}$$

IV. Link Scheduling

In this section, we present centralized and distributed link scheduling algorithms under different interference suppression models. We then improve algorithms by addressing the traffic load issue.

A. Interference graph

The conflict graph has been used to indicate which groups of links mutually interfere and cannot be active simultaneously [24]. It, however, cannot characterize the special constraints of MIMO link scheduling as discussed in Section III. Therefore, we define an interference graph, F_G, to model the effect of wireless interference and radio constraint in a communication graph G. Each vertex xy of F_G corresponds to a *directed* link (x, y) in the communication graph G. For any two vertices xy and pq, if y=p or q=x, there is an undirected edge between uv and pq in F_G. Otherwise, there is a *directed* edge from uv to pq in F_G if the transmission on link (x, y) interferes with the reception on link (p, q) in G, or a directed edge from pg to xy in F_G if the transmission on link (p, g) interferes with the reception on link (x, y) in G. Note that an undirected edge in F_G indicates that a pair of links in G suffer from the *radio constraint* and thus cannot be active simultaneously.

For simplicity, we use a single character to label a vertex in F_G, and define basic concepts for F_G used in the rest of the paper. The set of vertices adjacent to vertex pq through a undirected edge is called the *conflict-neighbors* of vertex pq. The set of vertices incident on pq is called the *in-neighbors* of vertex pq. Similarly, the set of vertices incident from pq is called the *out-neighbors* vertex pq. The in-degree of pq is the cardinality of its *in-neighbors*, and the out-degree of is the cardinality of its *out-neighbors*.

B. Centralized link scheduling

As illustration of our basic scheduling approach, we first present our centralized link scheduling method for MIMO. The interference-free link scheduling problem has been studied as the *vertex coloring* of conflict graph. This paper focuses on two different interference-free link scheduling problems due to the distinct features of MIMO links and associated interference suppression models. The *vertex coloring* of our interference graph cannot reveal the advantages of MIMO links, so we propose two dedicated approaches.

The basic idea of link scheduling is to first sort all links and then to assign each link the smallest available time slots such that an interference-free link scheduling is achieved. The sorting order and assigning order of links are two important factors that influence the scheduling result. Our centralized algorithms will sort links in two special orders, and then assign links in the reverse order of the sorting order. The first special one is as follows: we pick a vertex with the smallest in-degree in the remainder interference graph F_G, and then remove the picked vertex and associated edges from the graph; repeat such a process until the graph is empty. The second special one is similar, but each time it picks the vertex with the smallest out-degree in the remainder of the graph. In summary, a link with larger in-degree or out-degree will likely be processed earlier by our algorithms. As shown in Algorithms 1 and 2, the centralized scheduling algorithms under TOIS and ROIS models use the first order to sort links. They can also use the second order to sort links.

Algorithm 1 Centralized Scheduling under TOIS Model

Require: A communication graph $G = (V, E)$ of $|E|$ links.
Ensure: An interference-free link scheduling.

Construct the interference graph F_G and let $G' = F_G$.
SortLinks(G')
for $i = 1$ to $|E|$ **do**
  Assign link l_i the smallest time slot t with $X^{ts}_{l_i,t} = 1$ such that, prior to the transmission of l_i, $N(l_i, t) < m$, $N(l'_i, t) < m$ for each l'_i where $l'_i \in I^+_l$ and $X^{ts}_{l',t} > 0$, and none of *conflict-neighbors* of l_i are active at that slot.
for $i = 1$ to $|E|$ **do**
  Increase $X^{ts}_{l_i,t}$ as many as possible for active link l_i at time slot t if the new scheduling remains interference-free.

SortLinks(G')
while G' is not empty **do**
  Find the vertex with the *smallest* in-degree in G' and remove this vertex and all associated edges from G'. Let l_k denote the $(|E| - k + 1)$th vertex removed.

For link scheduling under TOIS model, Algorithm 1 processes links from l_1 to l_m and assign each l_i the smallest time slot t if following conditions hold prior to the transmission of l_i. First, none of *conflict-neighbors* of l_i are active at time slot t. Second, the receiver of each active *out-neighbors* of l_i at time slot t observes less than m streams at that time slot, and so does the receiver of l_i. Note that there is an upper bound on the largest number of streams can be transmitted over link l_i at time slot t. In other words, the largest value of $X^{ts}_{l_i,t}$ is

$$\min\left\{m - N(l_i, t), m - N(l'_i, t) \text{ for } l'_i \in I^+_{l_i} \text{ and } X^{ts}_{l'_i,t} > 0\right\}.$$

As shown in Algorithm 2, the link scheduling under ROIS model will be accomplished by two sequential stages at each time slot. In the first stage, Algorithm 2 will process links from l_1 to l_m, and allocate a time slot t to unscheduled links as much as possible such that an interference-free schedule under TOIS model is achieved at that time slot. In the second stage, Algorithm 2 will process links in the same order, and assign each unscheduled link l_i the same time slot t under ROIS model if following conditions hold prior to the transmission of l_i. First, none of *conflict-neighbors* of l_i are active at the time slot t. Second, the sufficient and necessary condition of interference-free schedule under ROIS model holds for l_i and t. If unscheduled links still exist after this round, the process repeats in the next time. Recall that $X^{rs}_{l_i,t}$ denotes the number of streams which are transmitted over an active link l_i at time slot t. According to inequation (3), it is clear that the value of $X^{rs}_{l_i,t}$ should not be larger than

$$\gamma = \min\left\{m - N(l_i, t),\ m - \sum\nolimits_{l'_i \in I^+_{l_i}, N(l'_i,t) \geq m} N_i(l'_i, t)\right\}.$$

The inequation (3) ensures that each active link can transmit one stream in order to achieve an interference-free schedule. If each active link wants to transmit streams as many as possible without compromising the existing interference-free schedule, the value of $X^{rs}_{l_i,t}$ should not be larger than $m - N(l'_i, t)$ for any active link l'_i with $N(l'_i, t) < m$ in out-neighbors of l_i at time slot t. This guarantees that those active links can still decode their desired streams when link l_i is active at the same

Algorithm 2 Centralized Scheduling under ROIS Model

Require: A communication graph $G = (V, E)$ of $|E|$ links.
Ensure: An interference-free link scheduling.

Construct the interference graph F_G and let $G' = F_G$.
SortLinks(G')
Let t denote an initial time slot.
while Exist links that have not been scheduled **do**
  for $i = 1$ to $|E|$ **do**
    if Link l_i has not been scheduled **then**
      Let $X^{ts}_{l_i,t} = 1$ for l_i and time slot t only if, prior to the transmission of l_i, $N(l_i, t) < m$, $N(l'_i, t) < s$ for each l'_i where $l'_i \in I^+_l$ and $X^{ts}_{l',t} > 0$, and none *conflict-neighbors* of link l_i are active at that slot.
  for $i = 1$ to $|E|$ **do**
    if Link l_i has not been scheduled **then**
      Let $X^{rs}_{l_i,t} = 1$ for l_i and time slot t if inequation (3) holds for l_i and t prior to its transmission, and none of *conflict-neighbors* of l_i are active at that slot.
  Let t denote the next time slot.
for $i = 1$ to $|E|$ **do**
  Increase $X^{ts}_{l_i,t}$ (if $X^{ts}_{l_i,t} = 1$) or $X^{rs}_{l_i,t}$ (if $X^{rs}_{l_i,t} = 1$) as many as possible for active link l_i at time slot t as long as an existing interference-free schedule is still kept.

time slot. Thus, the largest number of streams over an active link l_i under ROIS model at time slot t is

$$\min\left\{\gamma, m - N(l'_i, t) \text{ for } l'_i \in I^+_{l_i,t}, N(l'_i, t) < m\right\}.$$

In Algorithms 1 and 2, an active link l at a given time slot t is only allowed to transmit one stream in order to make more links be active at the same time slot. After each link is scheduled to a given time slot, Algorithms 1 and 2 process all links from l_1 to l_m and append additional streams as many as possible over an active link l_i at time slot t if the sufficient and necessary condition of interference-free scheduling is satisfied.

C. Distributed link scheduling

In a wireless network, centralized algorithms are often difficult to implement, and even they are adopted, due to the dynamic features of wireless networks, it is inefficient to update the schedule using a centralized algorithm. To address this issue, we further design distributed algorithms to provide an interference-free schedule. Intuitively, we can design the distributed versions for Algorithms 1 and 2. However, finding the link with the global maximum in-degree or out-degree in F_G iteratively is not trivial. The node having packets to transmit over a link does not know the in-neighbors or out-neighbors of that link in advance, no matter the knowledge of F_G. Thus, our distributed scheduling as shown in Algorithm 3 neither sorts links nor processes links in order.

Algorithm 3 shows the distributed link scheduling under TOIS model and that under ROIS model together. For that under TOIS model, each node schedules its links which have packets to transmit, according to TOIS model based on collected information about link scheduling results from interfering nodes. For that under ROIS model, each node first tries to schedule its links having packets to transmit according to TOIS model, and then ROIS model if the necessary and sufficient condition of TOIS model are not satisfied. The process to provide each node with necessary information about

Algorithm 3 Distributed Scheduling

Require: A communication graph $G = (V, E)$ of $|E|$ links.
Ensure: An interference-free link scheduling.

1: Node v_i identifies a set of links, say L_i, over which v_i will transmit packets.
2: **while** $Omes_i$ is not empty, or L_i is not empty **do**
3:   **while** Node v_i failed to obtain channel **do**
4:     **MonitorChannel**(v_i) and competes for the channel.
5:   **if** Node v_i selects a link $l_{i,k}$ from L_i successfully **then**
6:     **if** ROIS Model **then**
7:       **if** v_i received overload messages before time slot t and inequations (3) holds for $l_{i,k}$ at a smallest time slot $t' < t$. **then**
8:         v_i assigns $l_{i,k}$ the time slot t'.
9:       **else**
10:         v_i assigns $l_{i,k}$ a smallest slot t such that v_i does not meet an overload message at slot t and $s_t < m$ at v_k.
11:     **else**
12:       v_i assigns a link $l_{i,k}$ a smallest time slot t at which no overload messages are received by v_i and $s_t < m$ at v_k.
13:     v_i broadcasts a message $color(v_i, v_k, t)$ to all nodes within distance $d_i(v_i)$ of v_i, and removes link $l_{i,k}$ from L_i.
14:   **if** a message $omes(v_i, t, Cmes(v_i, t))$ exists in $Omes_i$ and v_i even received a message $color(*, v_i, t)$ **then**
15:     Such messages are broadcasted to all nodes within a distance $d_i(v_i)$ of node v_i, and then removed from $Omes_i$.
16: **MonitorChannel**(v_i)

MonitorChannel(v_i)

1: **if** v_i receives a message $omes(v_j, t, Cmes(v_j, t))$ **then**
2:   v_i records that $s_t \geq m$ for node v_j.
3:   **if** $Cmes(v_j, t)$ is not empty, and v_i is the transmitters of at lest one link in $Cmes(v_j, t)$ **then**
4:     v_i reschedules all links in $Cmes(v_j, t)$ with v_i as transmitters by adding them into the set L_i.
5: **if** v_i receives a message $color(v_j, v_k, t)$ **then**
6:   **if** v_i has received message $color(v_j, v_k, t')$ **then**
7:     v_i removes the $color(v_j, v_k, t')$, and decreases s'_t by one.
8:   v_i records the message, and increments s_t by one. If s_t=m, v_i adds an overload message $omes(v_i, t, Cmes(v_i, t))$ to the set $Omes_i$ where $Cmes(v_i, t)$ is an empty set. If $s_t > m$, v_i adds the link l_{jk} to the set $Cmes(v_i, t)$.

interfering nodes is common in scenarios of both models, and involves two types of messages as follows. The first type is $color$ message which records the time slot assigned to a link represented by a pair of transmitter and receiver. The second type is overload message $omes$ which indicates that a node receives m messages of type $color$ at the same time slot and records the number of intended streams received.

Each node v_i, having unscheduled links or overload messages, competes channel randomly. If successes, node v_i assigns one unscheduled link if exists with a smallest time slot; creates a $color$ message and broadcasts it with the overload messages to all nodes residing within its interference range. If no unscheduled link exists, node v_i only broadcasts the overload messages.

If node v_i fails or need not to compete channel, it monitors channel and waits to receive $omes$ and $color$ messages from all interfering nodes. The simple broadcast operation of the two types of messages can ensure that all related nodes obtain necessary information to allocate a smallest time slot for each link. An overload message is created if v_i has received m messages of type $color$ at the same time slot t, that is s_t=m at v_i.

As introduced above, receivers of links to be scheduled need to broadcast overload messages once those messages are produced. Such policy, however, is not sufficient to guarantee an interference-free schedule in general scenarios. Specifically, when a transmitter T wants to assign a time slot t to a link, it might not receive an overload message of time slot t from the receiver R even if it has been overloaded at that time slot. The reason is that no links with node R as receivers are active at the time slot t. Thus, the receiver R cannot decode desired streams from the transmitter T as it receives more than m streams if T assigns that link the same time slot. Algorithm 3 assumes that a certain competition based MAC layer (e.g., 802.11 with RTS/CTS) is available for a node to obtain the channel and to address that scenario. Concretely speaking, the CTS control packet is used to contain information of streams observed by a desired receiver as shown in line 10 and line 12 in Algorithm 3. Algorithm 3 can guarantee an interference-free schedule under any scenario, and is easy to be implemented without much additional computation on each node.

The distributed algorithm mentioned above shows a fundamental process through which each communication link is scheduled to transmit only one stream at a given time slot. According to the centralized algorithms, we know that each scheduled communication link may have opportunity to transmit more streams besides the one allocated by Algorithm 3. Such opportunity can be captured if an additional process is appended at the end of Algorithm 3 to allocate additional streams as many as possible for each scheduled link. All necessary information to make decision on how many additional streams can be transmitted over a link can be obtained by the transmitter in the same way as Algorithm 3 does. Besides above discussions, we address another two key issues, the transmission of the two types of messages and the delayed overload message.

1) Transmission of Color and Overload Messages: In our distributed algorithm, each node should broadcast its $color$ and overload messages to all nodes within its interference range. This can ensure that those nodes obtain necessary information to allocate a smallest time slot for each of its unscheduled links under TOIS model as well as ROIS model. To enable these messages decode-able by nodes out of the transmission range but within the interference range, the transmission range of $color$ and overload messages needs to be extended by a factor along the following two approaches.

In the first approach, a transmitter uses multiple antennas to transmit two types of messages via spatial diversity, in which dependent streams instead of parallel independent streams are transmitted on multiple antennas. In this case, we exploit the diversity gain of MIMO to extend the transmission range without enhancing transmission power. This approach possesses particular advantages, but it has two disadvantages which directly affect its usability in practice. First, the range extension is not the same in all directions and depends on the radiation pattern used by each transmitter. Second, each receiver uses all antennas to exploit the diversity gain, whereas it cannot decode desired packets from multiple transmitters which broadcast packets simultaneously in the same vicinity due to hidden terminal problem.

In the second approach, each transmitter uses one antenna to broadcast packets, whereas leaves corresponding receivers the chance to overcome the hidden terminal problem by decoding all received packets from less than m transmitters successfully. The hidden terminal problem is serious in the context of broadcasting the two kinds of messages. The cost of the second approach is that each transmitter must allocate more transmission power to extend the transmission range. In summary, the second approach is more suitable to our distributed link scheduling algorithm than the first approach.

2) Delayed Transmission of Overload Message: In our distributed algorithm, each node v_i is required to construct an overload message and then broadcast the message as soon as possible once it has observed number of m *color* messages at time slot t. Node v_i, however, sometimes cannot broadcast the overload message in time because it may not obtain the channel quickly if multiple neighborhood nodes compete the same channel simultaneously. Thus, all nodes within the interference range of node v_i may not receive the overload message, and each node v_j may assign a link with the same time slot t if it obtains channel successfully. Furthermore, node v_j broadcasts a *color* message to all nodes within its interference range, and may make those nodes schedule their links wrongly. In this case, v_i cannot decode its desired streams, and our distributed algorithm cannot guarantee an interference-free link schedule.

To address this issue, each node v_i monitors channel and inserts link l_{jk} into a set $Cmes(v_i, t)$ if link l_{jk} is contained by any received $color(v_j, v_k, t)$ message. After obtaining a channel, node v_i broadcasts an overload message $omes(v_i, t, Cmes(v_i, t))$. All nodes that are transmitters of links contained in the overload message of node v_i will find that they even assigned those links with incorrect time slots after receiving the overload message. In order to eliminate negative impacts of the links scheduled incorrectly, those links will be reassigned with smallest time slots again, and all nodes received *color* messages about those links will be notified to update related records. The **MonitorChannel** part of Algorithm 3 shows the detail process.

D. Link scheduling with traffic load

In the above discussions, we assume that the number of packets that need to go though each communication link is the same. Such an assumption, however, is not always true and it is possible that some communication link carries more traffic than others.

Let us consider a simple example of multi-hop MIMO network as shown in Fig.3(a) in which each node has $m = 2$ antennas and there are k flows, with flow f_i starting at a source node s_i and ending at a destination node d_i for $1 \leq i \leq k$. In Fig.3(a), we can see that the transmission of link $v_2 d_i$ will interfere the reception of any link with v_1 as receiver. Thus, at

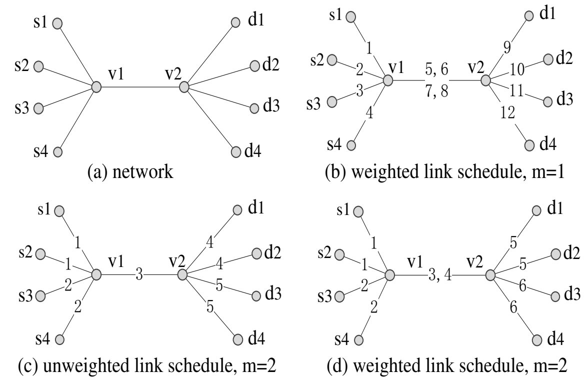
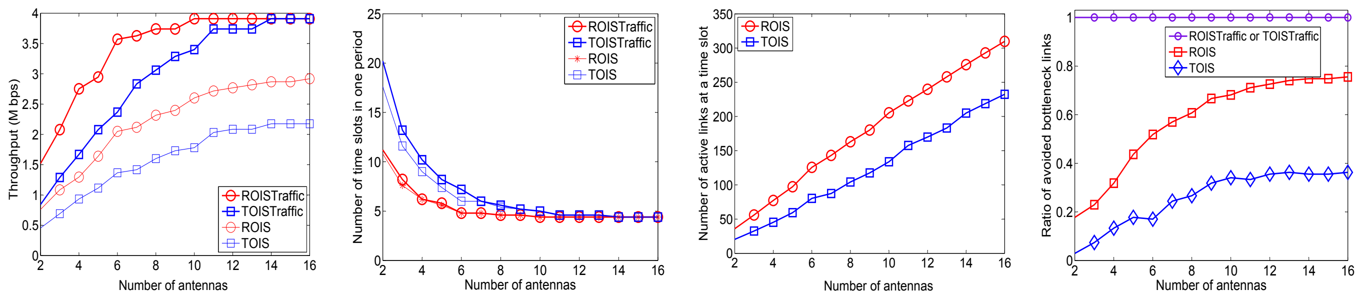


Fig. 3. Simple example: unweighted link schedule is inefficient.

least $k+1$ time slots are required to implement an interference-free link schedule, which can be obtained by assigning time slot $\lceil i/2 \rceil$ to link $s_i v_1$ for $1 \leq i \leq k$, time slot $k/2+1$ to link $v_1 v_2$, and time slot $\lceil i/2 \rceil + k/2 + 1$ to link $v_2 d_i$ for $1 \leq i \leq k$ as in Fig.3(c). In this schedule, only one stream is transmitted over links $s_i v_1$ and $v_2 s_i$, whereas at most two streams can be transmitted over link $v_1 v_2$. We assume the rate at which data can be sent over links $s_i v_1$ and $v_2 s_i$ is a bps and that of link $v_1 v_2$ is $2a$ if each link uses all time slots. It is obvious that all k flows go through link $v_1 v_2$, and node v_1 will receive ka data and can send only $2a$ data every $k+1$ slots. Thus, link $v_1 v_2$ becomes the bottleneck and the overall network throughput is $\frac{2a}{k+1}$ bps. For each flow, its throughput is approximately $\frac{2a}{k(k+1)}$ bps. Clearly, we should generalize the link scheduling that takes the traffic on each link into account.

Given a set of k flows, we adopt a certain routing algorithm to determine the routing path for each flow, and then assign a weight to each link used by at least one flow. The weight of a link l, for example $w(l)$, is the total flow passing through l divided by the bandwidth $c(l)$ of link l, that is $w_l = \frac{\sum_{f_i:\ f_i \ using \ l} f_i}{c(l)}$. We will show how a weighted link schedule can improve the network throughput using the example shown in Fig.3(d). By assigning weight 1 to each link $s_i v_1$ and $v_2 d_i$ for $1 \leq i \leq k$ and $k/2$ to link $v_1 v_2$, obviously, $3k/2$ time slots can guarantee an interference-free schedule. The total network throughput is now $2a/3$ bps and each flow has a throughput of $2a/3k$ bps, being increased compared with the unweighted link schedule in an order of k. We also show how MIMO links can improve the network throughput by comparing examples shown in Fig.3(d) and Fig.3(b). Fig.3(b) illustrates a weighted link schedule where each node has only one antenna. In this case, the total network throughput is $a/3$ bps and each flow has a throughput of $a/3k$ bps. How to obtain a valid weighted link scheduling is as follows.

Each link l is assigned a given time slot to transmit one data stream by our centralized or distributed algorithm, and then is allocated a weight where the capacity of each link is assumed to be the same. Recall that the last step of our centralized and distributed algorithms is to allocate additional streams as many as possible for each scheduled link. Operations in this step should take into account the traffic on each link. Specifically, the centralized algorithm processes scheduled



(a) Network throughput. (b) Enhancement of throughput. (c) Number of active links per time slot. (d) Ratio of avoided bottleneck links.

Fig. 4. The impact of number of antennas at each terminal under distributed algorithms.

links in descending order of weight of each link, while in the distributed algorithm, each link competes channel with a probability which is proportional to its weight. Such simple modifications will allocate links with high weight more data streams, so as to increase the capacity for links which are shared by many flows. If there are still links with a weight larger than the number of allocated data streams, the links will be re-scheduled by repeating the above process.

V. Performance Evaluation

This section starts with an introduction to the simulation methodology and scenarios, and then evaluates achievable throughput in multi-hop MIMO networks under different scheduling algorithms and network parameters.

A. Simulation methodology and scenarios

Our simulations are based on SWANS++ [25] which extends the JiST/SWANS framework and provides a good platform for simulation of wireless networks. We randomly generate a static multi-hop MIMO network with 500 homogeneous terminals. All terminals are uniformly distributed in a $800m \times 800m$ square. We use the shadowing path loss model as the signal propagation model, where the exponent for path loss formula is 2.8 and the shadowing deviation is 6.0. The default transmission power is 15dBm. The default radio reception sensitivity is set to be -72dBm, so the initial interference ranges is 78m. The default radio reception threshold is -67dBm, and thus the transmission range is 51m. The default bandwidth of each link to transmit one stream is 1024 bytes. We randomly generate 100 pairs of source and destination, and discover an end-to-end flow for each reachable pair of terminals using the DSR routing protocol [23].

A flow q is defined as the collection of all the links between a pair of source and destination. The flow rate f_q is defined as the throughput of its bottleneck links. Given a set of flows, denoted as Q, we use the $\sum_{q \in Q} f_q$ as a metric to characterize the network throughput under a schedule. We also use other three metrics to explain the changes of network throughput. They are the period T to achieve an interference-free link schedule, the ratio of avoided bottleneck links, and the number of links which can be active at one time slot.

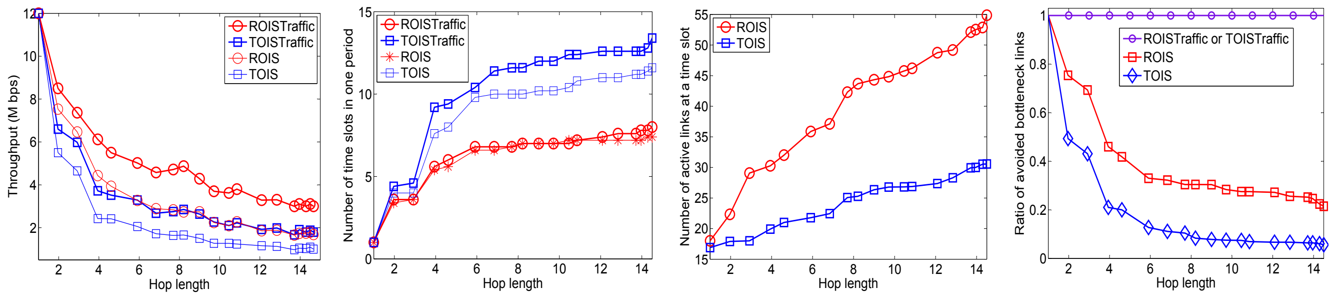
In groups of simulations, we first regulate the following parameters to evaluate their impact on the performance of the proposed distributed link scheduling algorithms. They are the number of antennas, the average hop length of data flows, and the interference range. The default number of antennas is 3. The default number of interference range is 1.5. We then compare the network throughput under the distributed and centralized link scheduling algorithms with TOIS and ROIS models. Let terms "ROISTraffic" and "TOISTraffic" denote the traffic-aware link scheduling algorithms under ROIS and TOIS models, respectively.

B. Impact of number of antennas on distributed link scheduling

In the simulations, the number of antennas at each terminal varies from 2 to 20. We try to discover and schedule end-to-end flows for 100 pairs of source and destination terminals, but, DSR protocol only successfully responses 17 flows. The average hop length of flows is 22.11. The largest link weight is 3. The ratio of interference range to transmission range is set as default. Fig.4(a) shows that regardless the scheduling algorithm, the throughput first increases as the number of antennas increases, then reaches and keeps at the highest value after the number of antennas exceeds a threshold. The period of a schedule also stops decreasing simultaneously, as shown in Fig.4(b). The reason is that after the number of antennas reaches the threshold, all solvable interference constraint can be solved and hence the radio constraint becomes the dominating factor.

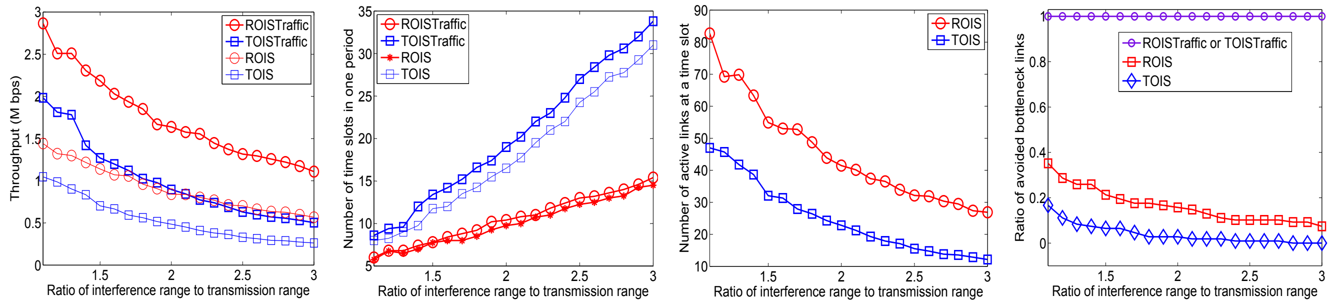
Results in Fig.4(a) indicate that each algorithm based on ROIS model outperforms the corresponding algorithm based on TOIS model. The reason is that irrespective of the number of antennas, ROIS model can allocate more links to be active at the same time slot than TOIS model, as shown in Fig.4(c). The algorithms based on ROIS model hence use less time slots to achieve an interference-free schedule than that based on TOIS model, as shown in Fig.4(b), and obtain a higher throughput.

On the other hand, each traffic-aware algorithm outperforms the correlated standard algorithm although the scheduling period of the former is larger than the latter, as shown in Fig.4(b). The reason is that traffic-aware algorithms allocate sufficient time slots for those bottleneck links, whereas those bottleneck links cannot be avoided completely by the standard algorithms, as shown in Fig.4(d). Consequently, more flows obtain higher transmission rates using the traffic-aware algorithms.



(a) Network throughput. (b) Enhancement of throughput. (c) Number of active links per time slot. (d) Ratio of avoided bottleneck links.

Fig. 5. The impact of average hop length under distributed algorithms.



(a) Network throughput. (b) Enhancement of throughput. (c) Number of active links per time slot. (d) Ratio of avoided bottleneck links.

Fig. 6. The impact of interference range under distributed algorithms.

As a short summary to this group of simulations, the ROIS-based algorithm is recommended if traffic information is not known. Otherwise, ROISTraffic algorithm should be adopted.

C. Impact of hop length on distributed link scheduling

In this scenario, each flow is partitioned into a series of disjointed flows to be scheduled in each round, where the average hop length ranges from 1 to 15. We consider 20 different hop lengths and generate random sets of flows for each hop length. The largest link weight among links is still 3. The number of antennas and the ratio of interference range to transmission range are set default, respectively. Fig.5(a) shows that irrespective of the scheduling algorithm, the network throughput decreases as the average hop length increases. This can be explained as follows. First, the average out-degree and in-degree of nodes in an interference graph increase as the average hop length increases. Potentially there are more link interfering links along the flows that need to be scheduled. Second, the longer the multi-hop flows, the more bottleneck links likely arise, as shown in Fig.5(d). Thus, more time slots are required for an interference-free schedule, as shown in Fig.5(b), and hence result in a reduced throughput.

Fig.5(c) shows that the number of links assigned with a same time slot increases as the average hop length increases although the number of antennas is fixed. More links are added to be dealt with whereas the link density does not increase. Fig.5(d) shows that the ratio of avoided bottleneck flows decreases as the hop length increases. More bottleneck flows arise while each active link has less chance to append additional streams due to more out-neighbors. These two findings can explain why the throughput decreases as the hop length increases.

D. Impact of interference range on distributed link scheduling

In this scenario, the average hop length of flows is 22.11. The largest link weight of links is 3. The number of antennas is set default. In this set of simulation, we vary the ratio of interference range to transmission range from 1.1 to 3. Fig.6(a) shows that irrespective of the scheduling algorithm, the network throughput decreases as the interference range increases. The increase of interference range directly increases the probability of interference in the vicinity. As shown in Fig.6(c), less links can then satisfy the scheduling conditions defined in Section IV. As a result, more time slots are required for an interference-free schedule, as shown in Fig.6(b). For the similar reason, each link has less chance to expand additional streams at the same time slot after it is assigned a given time slot. Thus, less bottleneck links are addressed as the interference range increases, as shown in Fig.6(d).

E. Comparisons among models and algorithms

Curves in Fig.7 compare the ratios to which the algorithms approximate the optimal network throughput, when the number of antennas varies. The results indicate that all the approximate ratios, except the approximate ratio of the centralized ROISTraffic algorithm, increase first and then reach the highest value after the number of antennas exceeds a threshold. The results again demonstrate that centralized algorithms outperform their distributed counterparts. Meanwhile, ROIS-based algorithms always achieves higher approximation ratio than TOIS-based ones. Moreover, the distributed ROISTraffic algorithm performs best among the four distributed algorithms.

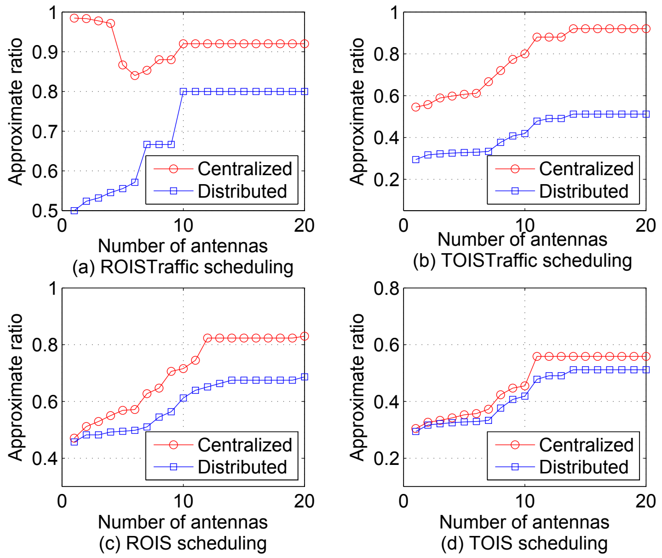


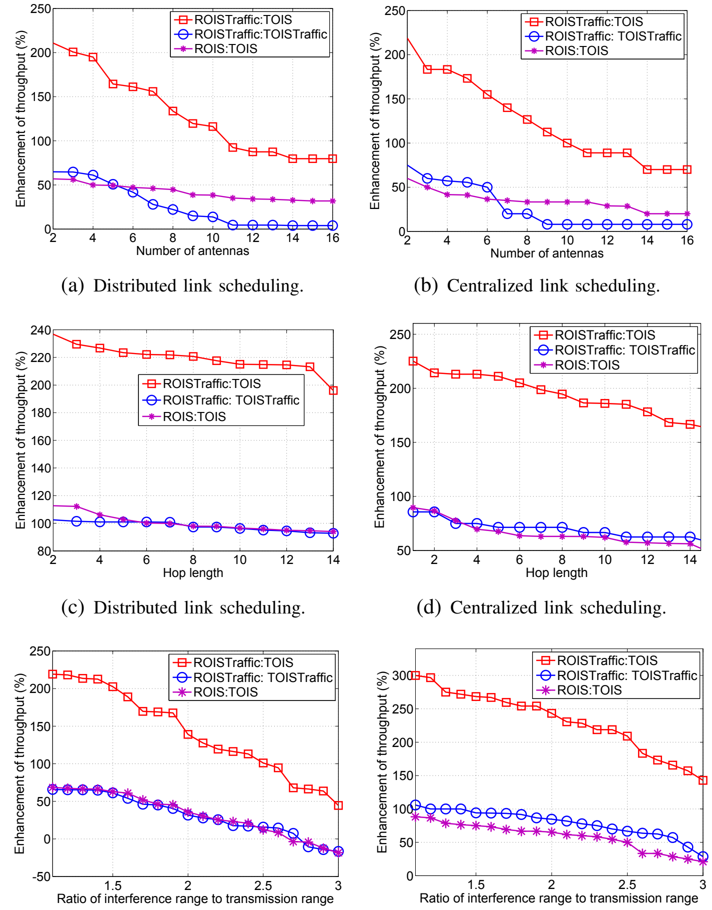
Fig. 7. The approximate ratio of different link scheduling algorithms.

The centralized ROISTraffic algorithm realizes near-optimal network throughput, irrespective of the number of antennas.

For further comparisons, we regulate parameters like the hop length and the interference range as well. Very similar results are obtained but omitted due to the page limit. We then examine the enhancement of throughput when comparing pairs of algorithms. For simplicity, let “ROISTraffic:TOISTraffic”, “ROISTraffic:TOIS”, and “ROIS:TOIS” denote the relative ratios of enhancement, respectively. As shown in Fig.9, the relative enhancement decreases along the increases of the number of antennas, the hop length, and the interference range. Nevertheless, ROISTraffic scheduling always significantly enhances network throughput over the standard TOIS scheduling (i.e. stream control). For example, under the default setting, the enhancement of throughput ROISTraffic:TOIS is over 200%. Moreover, ROIS-based scheduling apparently outperforms TOIS-based scheduling, under both distributed and centralized cases.

VI. Related work

Spatial reuse has recently attracted attentions in the research of multi-hop MIMO networks as a potential way to improve network capacity. We have discussed the benefits and three types of spatial reuse in Section I and Section II-C, and have shown the distinguishing features of stream control than the other two types. Some efforts have been taken in the past few years to explore effective and efficient approaches to facilitate spatial reuse in wireless networks. Sundaresan et al. propose a heuristic scheduling algorithm called SCMA to exploit the spatial multiplexing and stream control [12]. Wang et al. derive the theoretical upper bound on network capacity gain by stream control scheduling. They further design a scheduling algorithm named GreedySC, which outperforms SCMA under the general settings of wireless mesh networks and the potential network capacity is close to its theoretical upper bound [16]. Mundarath et al. propose a specific MAC scheme to realize both spatial multiplexing and spatial reuse



(a) Distributed link scheduling. (b) Centralized link scheduling.

(c) Distributed link scheduling. (d) Centralized link scheduling.

(e) Distributed link scheduling. (f) Centralized link scheduling.

Fig. 8. The relative enhancement of throughput under different network parameters.

[18] such that each terminal seamlessly allocates its available degrees of freedom to either spatial multiplexing or spatial reuse. In the way of cross-layer optimization, Bhatia et al. study the effect of stream control on network capacity of MIMO networks [15].

Hamdaoui et al. study three protocols for multi-hop MIMO networks, spatial multiplexing only (SMP), spatial reuse/ multiplexing (SRMP), and spatial reuse only (SRP) [17]. Actually, the SMP and SRMP are just the TDMA-based spatial multiplexing and spatial reuse used in this paper and other literatures, respectively. SRP is a special instance of spatial reuse in which each transmission session delivers only one stream so as to maximize the number of concurrent streams in the same vicinity. They formulate the packet-level constraints as a standard multi-commodity flow instance, and solve a linear programming problem after relaxation. By solving many instances, they achieve the maximum network throughput under each protocol. Nevertheless, they have not provided neither distributed nor centralized algorithms to realize the maximum. The preferred interference model in literature [17] called CiM, requires either the transmitter (using interference avoidance) or the receiver be responsible for spatial reuse. The CiM model, however, does not discuss the details at the transmitter side when its own receiver and interfered receivers are responsible to suppress interference. Even if the transmitter adopts stream control when receivers are responsible to suppress interference, CiM achieves at most the same throughput as stream control (as discussed in Section II-C).

Scheduling has been studied extensively to allocate time slots in TDMA MAC protocols that eliminate collision and guarantee fairness. Link scheduling is often reduced to coloring problems in conflict graphs. We find that simple *vertex coloring* on the interference graph cannot utilize the advantages of MIMO links. Thus we propose two dedicated link scheduling algorithms for centralized and distributed environments. Link scheduling based on TOIS achieves equivalent network throughput as those previous work belonging to the category of stream control. Link scheduling based on ROIS outperforms stream control, and achieves near-optimal network throughput.

VII. Conclusion

This paper presents our study on efficient link scheduling for multi-hop MIMO networks. We propose two dedicated interference suppression models to fully utilize the unique characteristics of MIMO. Based on these models, both efficient centralized and distributed link scheduling algorithms are devised to maximize the network throughput. For better applicability in practice, we further improve the proposed algorithms to address the non-uniform traffic demands. As shown through the comprehensive performance evaluation, link scheduling based on ROIS model outperforms that based on TOIS model in terms of network throughput. The traffic-aware link scheduling algorithm achieves near-optimal network throughput.

Following the work in this paper, we plan to study several issues in the future. The first issue is how to efficiently collect the information of interfering links for a MIMO link, which is an input element of link scheduling. The results in [18] provide some insights on this problem, yet using a model simpler than the one used in this paper. Second, link scheduling with dynamic traffic load on the links is another potential direction. It may help to realize an efficient scheduling that is adaptive to the traffic dynamics in wireless networks.

References

[1] E. Visotsky and U. Madhow, "Space-time transmit precoding with imperfect feedback," *IEEE Transactions on Signal Processing*, vol. 47, no. 6, pp. 2632–2639, Sept. 2001.

[2] C. Chuah, D. Tse, J. Kahn, and R. Valenzuela, "Capacity scaling in MIMO wireless systems under correlated fading," *IEEE Transactions on Information Theory*, vol. 48, no. 3, pp. 637–650, 2002.

[3] L. Zheng and D. N. C. Tse, "Diversity and multiplexing: A fundamental tradeof in multiple-antenna channels," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1073–1096, 2003.

[4] M. Vu and A. Paulraj, "MIMO wireless linear precoding," *EEE Signal Processing Magazine*, vol. 24, no. 5, pp. 86–105, sep 2007.

[5] W. Rhee and J. M. Cioffi, "On the capacity of multiuser wireless channels with multiple antennas," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2580–2595, 2003.

[6] Z. Pan, K. K. Wong, and T. S. Ng, "Generalized multiuser orthogonal space-division multiplexing," *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, pp. 1969–1973, 2004.

[7] N. Jindal and A. Goldsmith, "Dirty-paper coding versus TDMA for MIMO broadcast channels," *IEEE Transactions on Information Theory*, vol. 51, no. 5, pp. 1783–1794, May 2005.

[8] H. Dai, A. F. Molisch, and H. V. Poor, "Downlink capacity of interference-limited MIMO systems with joint detection," *IEEE Transactions on Wireless Communications*, vol. 3, no. 2, pp. 442–453, 2004.

[9] J. S. Jiang, M. F. Demirkol, and M. Ingram, "Measured capacities at 5.8 GHz of indoor MIMO systems with MIMO interference," in *Proc. IEEE VTC*, 2003, pp. 388–393.

[10] M. F. Demirkol and M. A. Ingram, "Control using capacity constraints for interfering MIMO links," in *Proc. International Symposium on Personal, Indoor, and Mobile Radio Communications*, 2002.

[11] ——, "Stream control in networks with interfering MIMO links," in *Proc. IEEE Wireless Communications and Networking Conference*, Louisiana, USA, Mar. 2003, pp. 343–348.

[12] K. Sundaresan and R. Sivakumar, "A unified mac layer framework for ad-hoc networks with smart antennas," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 546–559, 2007.

[13] S. Gaur, J. S. Jiang, M. A. Ingram, and M. F. Demirkol, "Interfering MIMO Links with Stream Control and Optimal Antenna Selection," in *Proc. IEEE GLBOCOM*, Dallas, TX, Nov. 2004.

[14] S. Gaur and M. Ingram, "Stream control for interfering MIMO links with linear MMSE receivers," in *Proc. 41th Annual Asilomar Conference on Signals, Systems, and Computers*, California, USA, Nov. 2007.

[15] R. Bhatia and L. Li, "Throughput optimization of wireless mesh networks with MIMO links," in *Proc. 26th IEEE INFOCOM*, Alaska, USA, May 2007, pp. 2326–2330.

[16] Y. Wang, D. M. chiu, and J. C. Lui, "Characterizing the capacity gain of stream control scheduling in MIMO wireless mesh networks," *Wireless Communications and Mobile Computing*, 2008.

[17] B. Hamdaoui and K. G. Shin, "Characterization and analysis of multi-hop wireless MIMO network throughput," in *Proc. the 8th ACM Mobihoc*, Quebec, Canada, 2007, pp. 120–129.

[18] J. C. Mundarath, P. Ramanathan, and B. D. V. Veen, "Exploiting spatial multiplexing and reues in multi-antenna wireless ad hoc networks," *Elsevier Journal of Ad Hoc Networks*, no. 7, pp. 281–293, 2009.

[19] G. G. Raleigh and J. M. Cioffi, "Spatio-temporal coding for wireless communication," *IEEE Transactions on communication*, vol. 46, p. 357C366, mar 1998.

[20] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor, *MIMO Wireless Communications*. Cambridge university press, 2007.

[21] G. J. Foschini and M. J. Gans, "On limits of wireless personal communications in a fading environment when using multiple antennas," *IEEE Personal Communications Magazine*.

[22] D. Guo, P. Yang, Y. Liu, and X. Li, "Link scheduling for static wireless networks with mimo links," CSE of HKUST available at http://www.cse.ust.hk/ liu/guodeke/mimo.pdf, Tech. Rep., 2009.

[23] D. B. Johnson, "Routing in ad hoc networks of mobile hosts," in *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, USA, Dec. 1994, pp. 158–163.

[24] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proc. 9th ACM MOBICOM*, 2003, pp. 66–80.

[25] D. Offnes, A. Hamilov, E. Atsan, and K. K. Udayashankar. Swans++: Extensions to the scalable wireless ad-hoc network simulator, http://www.aqualab.cs.northwestern.edu/projects/swans++/.