WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. (2008)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.632

Characterizing the capacity gain of stream control scheduling in MIMO wireless mesh networks

Yue Wang[1], Dah-Ming Chiu[2] and John C. S. Lui[1*,†]
[1]*Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong*
[2]*Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong*

Summary

Stream control (SC) has recently attracted attentions in the research of multiple input multiple output (MIMO) wireless networks as a potential way to improve network capacity. However, inappropriate use of SC may significantly degrade network capacity as well. In this paper, we provide the first formal study on SC scheduling in MIMO wireless mesh networks (WMNs). We derive the theoretical upper bound on network capacity gain of SC scheduling. We also provide an efficient scheduling algorithm and show that its achieved network capacity gain is close to its theoretical upper bound. Moreover, we point out the poor performance of a previous SC scheduling algorithm SCMA under the general settings of WMNs. This formal characterization provides a deeper understanding of steam control scheduling in MIMO WMNs. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: MIMO; stream control; network capacity; scheduling

1. Introduction

Recently, wireless mesh networks (WMNs) have been actively researched and developed as a key solution to provide ubiquitous network access, especially where wired networks are expensive to deploy, for example, in rural areas. Compared with mobile ad hoc networks (MANETs), wireless sensor networks (WSNs), and infrastructure-based mobile cellular networks, WMNs are (1) quasi-static in network topologies, (2) not resource constrained at mesh routers, and (3) easy and flexible to deploy. A typical WMN consists of mesh routers, Internet gateways, and mobile users. Mesh routers provide network access to mobile users and interconnect with each other *via* wireless links. Communication within a WMN can go through multiple hops. There is a special kind of mesh routers called Internet gateways which have wired connections to the Internet. Communication from or to the Internet goes through Internet gateways.

The research of WMNs focuses on improving network capacity, with various techniques from physical layer to network layer proposed, such as MIMO [1], multi-channel multi-radio [2], high-throughput routing [3], etc. In this paper, we study the effect of stream control (SC) on network capacity of MIMO WMNs. Before further discussions, we introduce the background of MIMO and SC briefly.

Multiple input multiple output (MIMO) technology is regarded as one of the most significant breakthroughs

*Correspondence to: John C. S. Lui, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong.
†E-mail: cslui@cse.cuhk.edu.hk

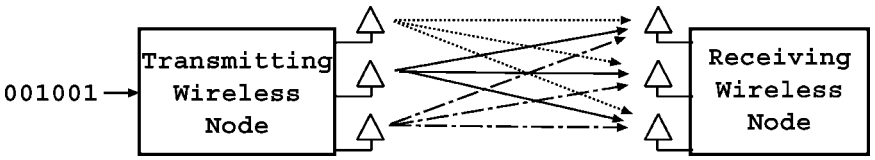


Fig. 1. MIMO wireless nodes with three antennas.

in recent wireless communication [1]. By exploiting multi-paths in indoor or outdoor environment, MIMO is able to provide very high data rate by *simultaneously* transmitting multiple *independent* data streams on the same wireless channel. This kind of simultaneous transmissions, referred to as *spatial multiplexing*, is very desirable to WMNs where *network capacity* is the main concern.

Generally speaking, MIMO can be represented as a system of multiple transmitting antennas and receiving antennas. As illustrated in Figure 1, there are three antennas for each wireless node. At the transmitting wireless node, a primary bit stream is splitted into three individual bit streams, which are then encoded and transmitted by the three antennas. The signals are mixed naturally in the wireless channel. At the receiver end, these three bit streams are separated and decoded, and then combined into the original primary bit stream. An analogy of the stream separation is similar to solving three unknowns in a group of three linear equations.

In general, we assume each MIMO node has $K \geq 1$ antennas. Therefore, the capacity of a link grows linearly with K since there are K independent streams.[‡] This link capacity gain is referred to as *spatial multiplexing gain*.[§] Roughly speaking, for the receiver to successfully separate and decode incoming streams, the following conditions must be satisfied: (a) the number of successfully decoded streams is not greater than K, and (b) the strength of interfering streams is far weaker than that of the successfully decoded ones. Otherwise, the receiver cannot decode any of the streams [4].

When a wireless link simultaneously transmits 0 *or* K streams on a channel, we call it *non-stream control* (*NSC*) scheduling. Alternatively, one can use *SC* scheduling to improve the capacity of MIMO systems. Under SC scheduling, a wireless link can simultaneously transmit k streams along a channel, where $0 \leq k \leq K$. SC scheduling provides more flexibility than does NSC scheduling since it can choose an *appropriate* number of streams so as to maximize network capacity.

SC is a unique feature of MIMO systems because of multiple antennas and space-time coding on a MIMO link. Traditionally, when two links interfere with each other, we can assign them to different time slots for transmissions. If the links are both MIMO, however, the spatial filtering capacities at the receivers and perhaps also at the transmitters enable the two links to operate co-channel with a higher network throughput than if they operate in the TDMA approach [5].

Figure 2 shows an example illustrating the improvement in network capacity by SC. There are four MIMO nodes with two flows of streams, one from node a to node b along link 1 and the other from node c to node d along link 2. The two links are placed close to each other and thus get mutual interference. Assume there are $K = 4$ antennas on each node. Under NSC scheduling, each wireless node transmits four streams at an aggregate rate of 100 Kbps. So the network capacity is 100 Kbps when the two links transmit in an alternate manner (Figure 2(b)).

Under SC scheduling, however, each link can choose to transmit k streams where $k \in \{0, 1, 2, 3, 4\}$. Specifically in MIMO, the channel gain for each streams can have quite large disparities even in the presence of interference [6]. For example, the ratio of the channel gain of the four streams is $1.0 : 0.8 : 0.7 : 0.5$. Each link can choose two best streams to transmit at an aggregate rate of $((1.0 + 0.8)/(1.0 + 0.8 + 0.7 + 0.5)) \times 100 =$ 60 Kbps, which is called *stream selection*, achieving a network capacity of 120 Kbps (Figure 2(c)).

Another kind of SC is called *partial interference suppression*. For example, suppose the interference between links 1 and 2 is weak enough in Figure 2, then they can both transmit at three streams, resulting in totally six streams transmitting per time slot.

[‡] Strictly, the capacity of a link grows linearly with the rank of the channel gain matrix of the link. This rank is usually equal to the number of antennas (K) on each node for moderate K antennas which are not extremely close packed.

[§] Another important feature of MIMO is that it has *diversity gain* that provides an increase in SNR at the receiver by partially or fully redundant$K$ streams. However, it was shown in Reference [4] that we often achieve higher throughput by exploiting spatial multiplexing instead of diversity. For the rest of this paper, we only focus on the scheduling based on spatial multiplexing.

DOI: 10.1002/wcm

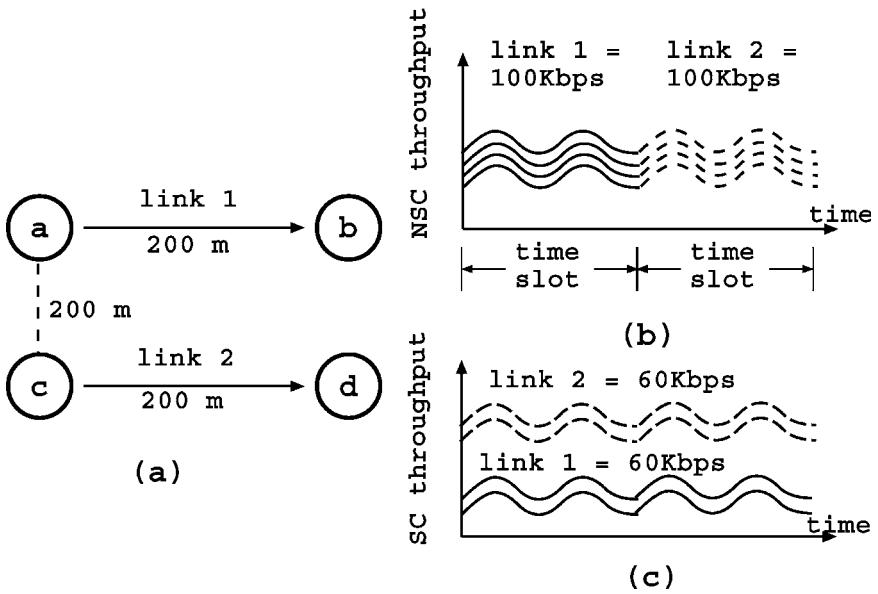


Fig. 2. Illustration of NSC and SC scheduling.

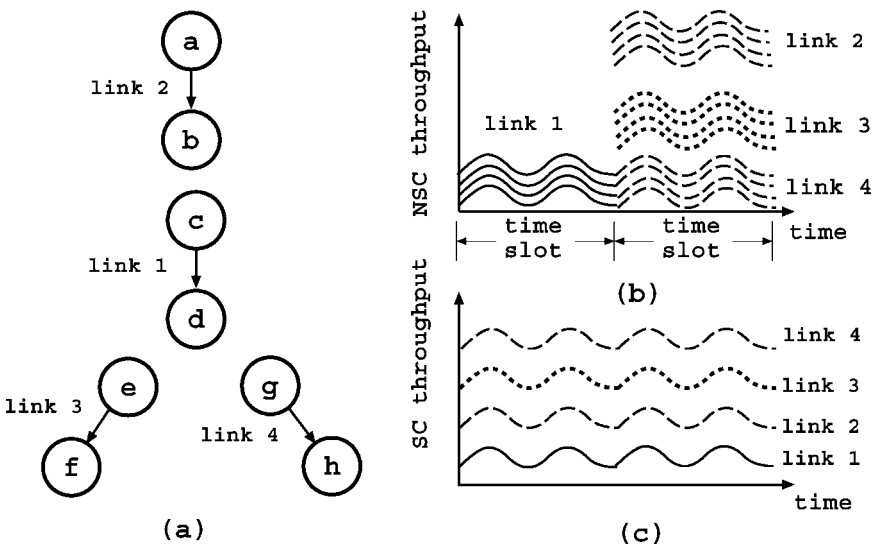


Fig. 3. Illustration of receiver overloading problem.

Previous work [6–8] showed that MIMO with SC can increase the total link throughput by 20–65 per cent for a set of *mutually interfering* links. However, it was noted in Reference [4] that when SC is not applied judiciously, one may encounter the problem of *receiver overloading* that will significantly degrade the overall network capacity. We illustrate this problem in Figure 3. Here, each wireless node has $K = 4$ antennas. Link 1 interferes with links 2–4 while the last three links do not interfere with each other. If link 1 transmits four streams in one time slot and the other three links each transmit four streams in the next time slot, it will result in an average of eight streams per time slot. If all four links use SC, then each can only transmit one stream because link 1 can only decode at most four streams. In this case, only four streams per time slot can be achieved, which has a poorer performance than NSC scheduling. The reason is that the use of SC for link 1 suppresses transmissions of the other three links.

Sundaresan *et al.* proposed a heuristic algorithm called *stream-controlled multiple access* (*SCMA*) for

DOI: 10.1002/wcm

using SC scheduling in multihop wireless networks [4]. The algorithm states that SC is used for multiple interfering links only when they belong to a single maximal clique (more discussions in later sections). Otherwise, NSC scheduling should be used. SCMA can work well for dense link interference graphs where links likely belong to a single maximal clique. However, it remains unclear how SCMA performs under the general settings of WMNs.

Another important observation is that SC can provide finer scheduling than NSC since a link can adjust its transmission rate from 0 to its channel capacity. Thus, network capacity optimization of NSC is an integer programming problem while that of SC is a non-integer programming problem (see Section 2). An important question that needs to be addressed is whether there is an *optimality gap* between NSC and SC. Note that the similar optimality gaps have been found in channel assignment [9] and routing [10].

In this paper, we provide the first formal characterization on the capacity gain of SC *from the network layer perspective*. In particular, we explore the fundamental question that how much network capacity gain can be expected from SC scheduling.

In order to answer these questions, in Section 2, we derive the theoretical upper bound on the network capacity gain of SC scheduling. In Section 3, we present a greedy SC scheduling algorithm GreedySC. In Section 4, we discuss the simulation results and compare the performance of GreedySC and that of SCMA. In Section 5, we present the related work and summarize our contribution. Finally, Section 6 discusses the limitations of our work and concludes the paper.

2. Upper Bound on Network Capacity Gain of Stream Control Scheduling

In this section, we first define notations and terminologies used in this paper. Then we propose a general model for NSC and SC scheduling and derive the theoretical upper bound on the network capacity gain of SC scheduling.

Before the discussion, let us define the concept of network capacity according to Reference [9].[‖] Given a network G and a set of flows F (each associated with a desired rate), we say that these flows are in the *capacity region* of the network if they can be realized (theoretically or practically, depending on the context). In other words, capacity region consists of all achievable flow rate vectors. For a given *traffic pattern* (which is defined as the direction of the flow rate vector or the ratio of the flow rates),the *network capacity* actually refers to the corresponding point on the boundary of the capacity region. Thus, the network capacity for a certain traffic pattern can be defined as the maximum total throughput of flows. In practice, we can calculate network capacity as follows. Given the traffic pattern, we generate the corresponding flow demand vector $\boldsymbol{x}$ (x_i is the demand of flow i in terms of bytes). Suppose by using the scheduling algorithm S, the network finishes delivering all flow demands in time T_S, then the flow rate vector is $\boldsymbol{x}/T_S$, and the network capacity under S is:

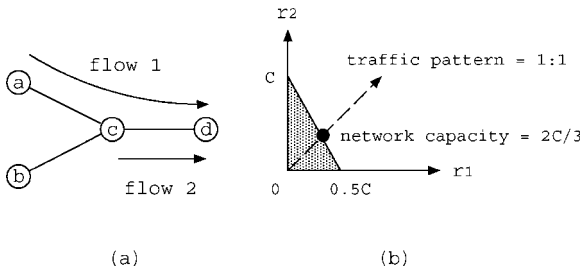


Fig. 4. Illustration of network capacity.

$$C_S = \frac{\sum_{i \in F} x_i}{T_S} \tag{1}$$

Specifically, for the *fixed* routing, the link workload vector $\boldsymbol{u}$ (u_i is the workload of link i in terms of bytes) is fixed as $\boldsymbol{u} = \boldsymbol{R}\boldsymbol{x}$, where $\boldsymbol{R}$ is the fixed routing matrix ($R_{ij} = 1$, if flow j is on link i; $R_{ij} = 0$, otherwise). So the network capacity can also be represented as

$$C_S = \frac{\sum_{e \in E} u_e}{T_S} \tag{2}$$

We illustrate the above definitions by Figure 4. There are four nodes (a, b, c, and d) and two flows ($a - c - d$ and $c - d$) in the network of Figure 4(a), resulting in three links ($L_{a,c}$, $L_{b,c}$, and $L_{c,d}$) contending the channel. Let C be the channel capacity, and r_1, r_2 be the rates of the two flows, respectively. We can easily calculate the capacity region of flow 1 and flow 2 by the constraint $2r_1 + r_2 \leq C$, shown by the shaded area in Figure 4(b). Suppose the traffic pattern $r_1 : r_2 = 1 : 1$, then the network capacity in terms of total throughput of flows is $(2/3)C$ when $r_1 = r_2 = (1/3)C$.

[‖] We adopt this definition of capacity since it isolates the capacity definition from fairness concerns.

DOI: 10.1002/wcm

We consider a multi-hop wireless network with n *static* nodes. Each node has multiple omni-directional antennas that operate on one channel. Each wireless node has the same transmission power and receiving sensitivity, resulting in a same *maximum* transmission range by a certain path loss model (e.g., two-ray ground model). Two nodes can form a link only when the distance between them is not greater than the maximum transmission range.

To derive *optimum* network capacity, we assume that the system operates in a synchronous time-slotted mode and each time slot is set very small for channel variation to be negligible. Links transmit at the beginning of each time slot, and they can transmit at the same time if they do not interfere with each other. Besides, we do not consider packet losses caused by fading. This is a reasonable assumption for deriving optimum network capacity as MIMO can significantly mitigate fading losses. Next, we assume that there are $K > 1$ antennas on each wireless node. A link can simultaneously transmit at most K independent data streams. We introduce *SC gain* g. That is, the total *normalized* transmission rate of a set of mutually interfering links using SC is *at most* g times that using NSC. Note that the transmission rate of a link is normalized by its channel capacity (assuming no interference) in this definition, because the channel capacity may be different for each link. Obviously, $1 \leq g < 2$ since $g \geq 2$ will violate the physical constraint, that is, multiple interfering links cannot transmit simultaneously at their full rates. Finally, we define network capacity gain of SC scheduling. For a given traffic pattern, let C_{NSC} and C_{SC} be the network capacity for NSC and SC scheduling, respectively, then $G_{\mathrm{SC}} = C_{\mathrm{SC}}/C_{\mathrm{NSC}}$ is defined as the *network capacity gain of SC scheduling*.

Here, we present a general model for NSC and SC scheduling in *time-varying* channels where channel capacity is a variable for different time and links. Let $c_e(t)$ be link e's channel capacity at time t, $\boldsymbol{y}(t)$ be the link transmission rate vector at t (i.e., $y_e(t)$ is the transmission rate of link e at t) and $N(e)$ be the set of interfering links (or neighbors) of e. If e transmits at t, the total normalized transmission rate of $N(e) \cup e$ must be less than or equal to g. Otherwise, receivers cannot decode the mixed signals correctly. Formally, we model the *NSC/SC scheduling constraint* as

$$\mathbf{1}_{y_e(t)>0} \cdot \left(\frac{y_e(t)}{c_e(t)} + \sum_{e' \in N(e)} \frac{y_{e'}(t)}{c_{e'}(t)} \right) \leq g, \quad \forall e \in E \text{ and } t \geq 0 \tag{3}$$

The function $\mathbf{1}$ is an indicator function, returning 1 when $y_e(t) > 0$ (or e transmits at t) and returning 0 otherwise. For NSC scheduling, $g = 1$ and $y_e(t)$ is 0 or $c_e(t)$, which indicates that all its neighbors cannot transmit when e is transmitting; for SC scheduling, $1 \leq g < 2$ and $y_e(t)$ is a real number between 0 and $c_e(t)$ since an SC link can properly choose the number of streams and its transmission rate. Note that we normalized the transmission rates in the above constraint (i.e., $y_e(t)/c_e(t)$) because channel capacity may be different for each link. In the special case of ideal channels where $c_e(t)$ is a constant, it indicates that the total transmission rate of $N(e) \cup e$ using SC can be g times the channel capacity of NSC. We now state and prove the fundamental theorem of SC scheduling.

Theorem 1. *For a given traffic pattern, the optimum network capacity of SC scheduling is at most g times that of NSC scheduling.*

Proof. We prove the theorem by two steps. First, we define SC (1) to be the SC scheduling when $g = 1$, and prove that the optimum network capacity of SC (1) is equal to that of NSC. Second, we prove that the optimum network capacity of SC scheduling is at most g times that of SC (1).

Step 1. We *only* need to prove that the optimum network capacity of NSC scheduling is greater than or equal to that of SC (1) scheduling, since NSC is a special case of SC (1). Theoretically, a time slot can be infinitesimal, within which a link's channel capacity is constant, so it is sufficient to prove that any *feasible* normalized link transmission rate vector of SC (1), $\boldsymbol{a}^{\mathrm{SC}}$ ($\sum_{e \in E} a_e^{\mathrm{SC}} \leq 1$) can be achieved by NSC (i.e., there are a sequence of normalized link transmission rate vectors of NSC $\{\boldsymbol{a}(i)^{\mathrm{NSC}} | i = 0, 2, \cdots, n-1\}$, so that $\sum_{i=0}^{n} a(i)_e^{\mathrm{NSC}}/n \geq a_e^{\mathrm{SC}}$, $\forall e \in E$). In other words, we can multiply $\boldsymbol{a}^{\mathrm{SC}}$ by a large number n to make a *virtual* integer workload vector $\boldsymbol{u} = \boldsymbol{a}^{\mathrm{SC}} \cdot n$ (real numbers can be approximated by rational ones), and prove that $\boldsymbol{u}$ can be scheduled in n *virtual* slots using NSC (0 or 1 unit of workload is scheduled per link per virtual slot). We prove this by induction for the number of *active* links m in the network. When $m = 1$ (u and a^{SC} are scalars), we can easily schedule $\boldsymbol{u}$ in n virtual slots since $a^{\mathrm{SC}} \leq 1$. Assume that we can schedule $\boldsymbol{u}$ in n virtual slots when there are m links. We consider adding the $(m+1)^{th}$ active link, say e. The neighbors of e consume at most $\sum_{e' \in N(e)} u_{e'}$ virtual slots, and we can use the remaining virtual slots (i.e., $n - \sum_{e' \in N(e)} u_{e'}$, which is greater than u_e by Equation (3))

DOI: 10.1002/wcm

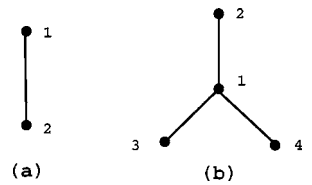


Fig. 5. Examples of link interference graphs.

to schedule u_e. So, we can schedule $\boldsymbol{u}$ for $m + 1$ links in n slots.

Step 2. We prove it by contradiction. Assume the optimum network capacity of SC is greater than g times that of SC (1), then we can get the optimum network capacity of SC (1) by dividing all link transmission rates by g, which is greater than itself. ■

Remark. From the view of optimization, SC scheduling cannot provide greater than g times network capacity gain although it can provide a finer scheduling scheme than NSC. In other words, there is no optimality gap between NSC and SC when $g = 1$. Therefore, if g is small, SC is not a worthwhile scheme for its higher computation efforts than that of NSC. Besides, the upper bound is not always achievable due to the potential receiver overloading problem.

Currently in our simulations, we use the range-based interference model [11] to determine a link's neighbors, which is shown to be realistic by Reference [12]. That is, let $L_{1,2}$ ($L_{3,4}$) be the link transmitting from node 1 to node 2 (from node 3 to 4) and let $d_{i,j}$ denote the distance between node i and node j, we define that $L_{1,2}$ is interfered by $L_{3,4}$ if $d_{3,2} \leq B \cdot d_{1,2}$, where B is called *interference range factor* determined by the SNR threshold of successful reception and the path loss model.

To represent interference between links, we introduce the *link interference graph* $G^I = (V^I, E^I)$, also known as flow contention graph in Reference [13]. (We rename it here to avoid confusion with end-to-end flows.) Here, V^I represents the set of wireless links and E^I the set of interfering link pairs. That is, if link i is interfered by link j or *vice versa*, we add two edges (i, j) and (j, i) into E^I, specifying that i and j cannot transmit at the same time. Figure 5 shows the link interference graphs for the networks in Figures 2 and 3, respectively. For example, in Figure 5(b), link 1's neighbor set $N(1) = \{2, 3, 4\}$.

3. A Stream Control Scheduling Algorithm GreedySC

In the previous section, we derived the theoretical upper bound of network capacity gain of SC scheduling. However, this upper bound may not be achievable. In this section, we present an SC scheduling algorithm *GreedySC* to get the network capacity gain G_{SC} in a practical way given the input parameter of SC gain g. Note that GreedySC is a network-layer scheduling algorithm. For maximizing SC gain g in the physical layer, readers can refer to Reference [7] which presented an algorithm to determine which streams to transmit for multiple interfering links based on channel state information.

Given a wireless network G and a link workload vector $\boldsymbol{u}$ (u_e is the workload of link e), the scheduling algorithm will calculate the time needed to transmit all workload in $\boldsymbol{u}$. Let T^{NSC} and T^{SC} be the time expended for NSC and SC scheduling, respectively, then the network capacity of NSC and SC scheduling is

$$C_{\text{NSC}} = \frac{\sum_{e \in E} u_e}{T^{\text{NSC}}} \tag{4}$$

$$C_{\text{SC}} = \frac{\sum_{e \in E} u_e}{T^{\text{SC}}} \tag{5}$$

and the network capacity gain of SC scheduling is

$$G_{\text{SC}} = \frac{C_{\text{SC}}}{C_{\text{NSC}}} = \frac{T^{\text{NSC}}}{T^{\text{SC}}} \tag{6}$$

To schedule $\boldsymbol{u}$, we design a greedy SC scheduling algorithm *GreedySC* (see Figure 6). At each time slot, the algorithm finds schedulable links of maximal workloads greedily so as to minimize the expended time. Note that GreedySC can also be applied to NSC scheduling by letting $g = 1$. The inputs to this algorithm are the link interference graph G^I, the workload vector $\boldsymbol{u}$, and SC gain g. Here, T is the elapsed time, δ is the interval of each time slot which is set small enough to adapt to time-varying channels, and R is the set of uncompleted links. $\boldsymbol{a}$ denotes the *normalized* link transmission rate vector (i.e., a_e is the normalized link transmission rate of link e), which is determined by the algorithm at each time slot, and $a_{N(e)}$ denotes the total normalized link transmission rate of e's neighbors, that is, $\sum_{i \in N(e)} a_i$.

The algorithm is described as follows. For each time slot, we first sort the links in the decreasing order of remaining workloads. In Step 1, we use the NSC mode to schedule links. That is, we iteratively pick a link e

DOI: 10.1002/wcm

GreedySC(G^I, $\boldsymbol{u}$, g)

```
T := 0;
R := {e|e ∈ E and u_e > 0};
While R ≠ φ
    T := T + δ;
    a := 0;
    Sort R in the deceasing order of u;
  /* Step 1. using NSC*/
    For each e ∈ R
      If a_{N(e)} = 0
        a_e := 1;
        u_e := u_e − a_e c_e(T)δ;
        If  u_e ≤ 0
          R := R − {e};
        fi
      fi
    end for
  /* Step 2. using SC*/
    For each e ∈ R and a_e = 0
      If a_{N(e)} = 1 and (a_{N(e')} = 0, ∀e' ∈ N(e) and a_{e'} = 1)
        a_e := g − 1;
        u_e := u_e − a_e c_e(T)δ;
        If u_e ≤ 0
          R := R − {e};
        fi
      fi
    end for
end while
```

Fig. 6. Specification of GreedySC scheduling algorithm.

from the sorted list. If no neighboring links of e have been scheduled, we schedule e for transmission (i.e., $a_e := 1$) and reduce its workload by $a_e c_e(t)\delta$; Otherwise, we do not schedule it. Then we move on to the next link on the sorted list until we finish examining all the links. In Step 2, we use the SC mode to schedule links. We iteratively pick a link e from the sorted list that was *un-schedulable* in Step 1. In order for e and its neighbors to satisfy Equation (3) when e is scheduled by SC (i.e., $a_e := g - 1$), we must ensure that only one neighbor of e, say e', was scheduled in Step 1 (i.e., $a_{N(e)} := 1$), and that no neighbors of e' were scheduled by SC in this step (i.e., $a_{N(e')} := 0$); Otherwise, we do not schedule it. Then we move on to the next link on the sorted list until we finish examining all the links. At this point, we decide the scheduling of one time slot. The above process is repeated for the next time slot until all the workloads in $\boldsymbol{u}$ is completed.

The algorithm naturally avoids the problem of receiver overloading as it schedules links using NSC first and thus guarantee the network capacity of SC scheduling is *always* greater than that of NSC scheduling. Note that, according to the scheduling sequence of links in GreedySC, an SC link e' ($a_{e'} = g - 1$) must neighbor some NSC link e ($a_e = 1$). Otherwise, e' can be scheduled using NSC. Therefore, all the other neighbors of e (or e') except e' (or e) cannot be scheduled so as to satisfy Equation (3). Therefore, for stream selection, the two links can further select other rates to achieve $a_e + a_{e'} = g$, for example, $a_e = a_{e'} = (g/2)$.

GreedySC can achieve higher network capacity gain than SCMA. To understand this, let us describe SCMA first. Briefly, multiple interfering links can be scheduled simultaneously using SC, only when they all belong to a *single* maximal clique.[¶] SCMA cannot cause receiver overloading, because links using SC do not suppress the transmissions of multiple maximal cliques. For example in Figure 5(a), links 1 and 2 belong to the single maximal clique $\{1, 2\}$, so they can be scheduled simultaneously using SC. In Figure 5(b), no two or more interfering links belong to a same single maximal clique, as link 1 belongs to three maximal cliques $\{1, 2\}$, $\{1, 3\}$, and $\{1, 4\}$. So SCMA cannot use SC here. But in fact, links belonging to multiple maximal cliques can also use SC for transmission, which is considered by GreedySC. For example in Figure 5(b), GreedySC can schedule link 2 using NSC ($a_2 = 1$) and schedule link 1 using SC ($a_1 = g - 1$). Generally, consider any two links a and b which both belong to a *single* maximal clique, and obviously, they can be scheduled using SC in SCMA. Suppose a is scheduled using NSC in GreedySC's Step 1, it is easy to see that b can be scheduled using SC in GreedySC's Step 2. Therefore, GreedySC covers the case of SCMA and thus can achieve higher network capacity gain.

4. Performance Evaluation

This section shows the simulation results in *general* WMNs. We test the network capacity gain of SC scheduling for GreedySC and SCMA, and compare it with the theoretical upper bound g.

By general WMNs, we mean static multihop wireless networks with the general settings of *network density* and *interference level* in current practices. We measure network density by the average node degree, that is, the average number of links of a wireless node, and measure interference level by the interference range factor B. The node degree in a WMN is typically

[¶] Here, a maximal clique is a maximal fully connected subgraph in the link interference graph. All links of a maximal clique interfere (or neighbor) with each other. A link may belong to multiple maximal cliques.

DOI: 10.1002/wcm

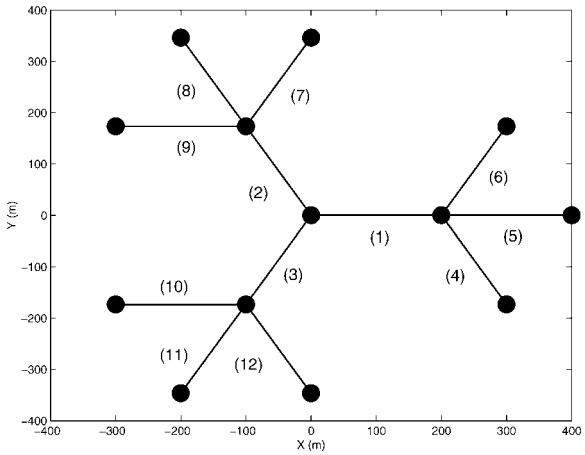


Fig. 7. A two-tier ternary tree network.

around 2 to 6 [14]; otherwise, there will be too many contending (interfering) links for a transmitting link. The maximum transmission range is set as 250 m in our simulations. The SNR threshold for successful reception is varied from 6 to 10 dB depending on the environment and coding scheme [15]. The corresponding interference range factor B varies from 1.4 to 1.8 calculated by the two-ray ground path loss model [16]. Therefore, higher SNR threshold implies more interfering links for a transmission and thus a denser link interference graph by introducing more edges in G^I. In our simulations, we assume ideal channels and use a TDMA simulator to schedule link workloads.

First, we consider tree networks which are often used in backhaul WMNs [17]. Figure 7 shows a regular two-tier ternary tree network used in our experiment. The root node are an Internet gateway and the leaf nodes are access points, while the other nodes are backhaul nodes. All links are of a distance of 200 m here and their numbers are labeled in the figure. We generate 100 flows, each with the same demand of 1000 KB. All flows in this network are from the access nodes to the gateway, or *vice versa*.

Figure 8 shows the network capacity gain of SC scheduling (G_{SC}) when GreedySC and SCMA are used for this tree network, respectively. We set $g = 1.5$ for all wireless links. When the SNR threshold is among 6–9 dB, there are three maximal cliques in the corresponding link interference graph, that is, $\{1, 2, 3, 4, 5, 6\}$, $\{1, 2, 3, 7, 8, 9\}$, and $\{1, 2, 3, 10, 11, 12\}$. Links 1–3 are the bottleneck links in the network, but they cannot use SC in SCMA because they all belong to three maximal cliques. So G_{SC} of SCMA (1.09) is much lower than that of GreedySC (1.42). When the SNR threshold is 10 dB, the corresponding link interference graph becomes so dense that there is only one maximal clique, that is, $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$. As a result, both SCMA and GreedySC can schedule all links using SC, and $G_{\mathrm{SC}} = 1.5$ in both algorithms.

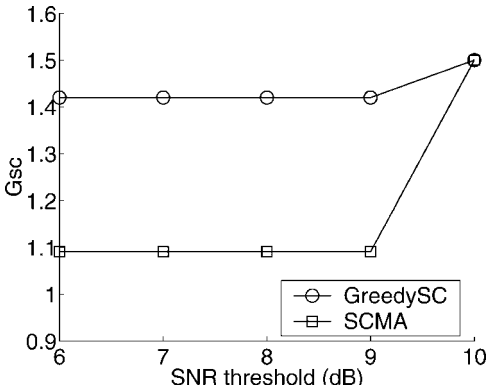


Fig. 8. Network capacity gain of SC for the tree network ($g = 1.5$).

Second, we consider random networks. We randomly place 25 wireless nodes in a square area of $1000 \times 1000\,\mathrm{m}^2$. The average node degree is 4.2. We set $g = 1.5$ for all wireless links. We generate 100 flows with sources and destinations randomly selected. Each flow has the same demand of 1000 KB. The flow demand vector $\boldsymbol{x}$ is mapped into the link workload vector $\boldsymbol{u}$ by $\boldsymbol{u} = \boldsymbol{R}\boldsymbol{x}$, where $\boldsymbol{R}$ is the routing matrix ($R_{ij} = 1$, if flow j is on link i; $R_{ij} = 0$, otherwise) and is calculated by the shortest hop-count routing for these flows.

Figure 9 shows the network capacity gain of SC scheduling G_{SC} when GreedySC and SCMA are used, respectively. SCMA provides little capacity gain because there are few links belonging to a single maximal clique, while GreedySC provides 30–40 per cent improvement on network capacity compared with NSC scheduling.

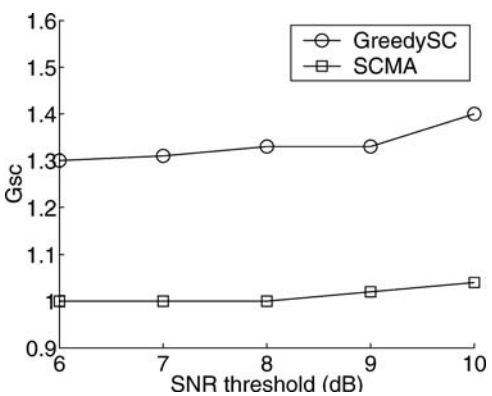


Fig. 9. Network capacity gain of SC for a random network in $1000 \times 1000\,\mathrm{m}^2$ ($g = 1.5$).

DOI: 10.1002/wcm

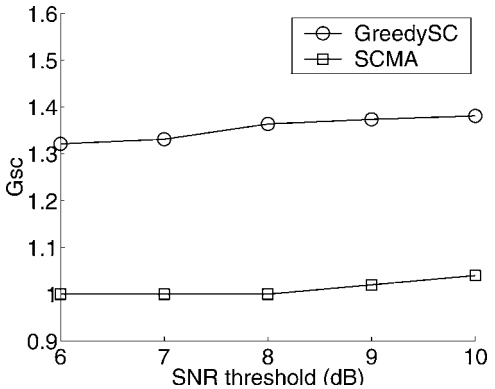


Fig. 10. Network capacity gain of SC for a random network in $800 \times 800\,\mathrm{m}^2$ ($g = 1.5$).

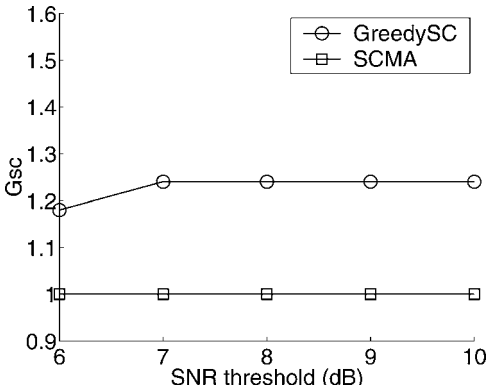


Fig. 11. Network capacity gain of SC for a grid network in $1000 \times 1000\,\mathrm{m}^2$ ($g = 1.5$).

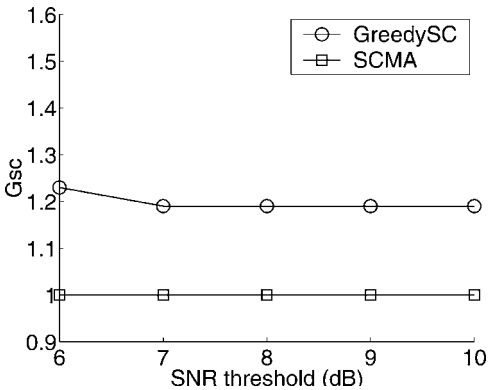


Fig. 12. Network capacity gain of SC for a grid network in $800 \times 800\,\mathrm{m}^2$ ($g = 1.5$).

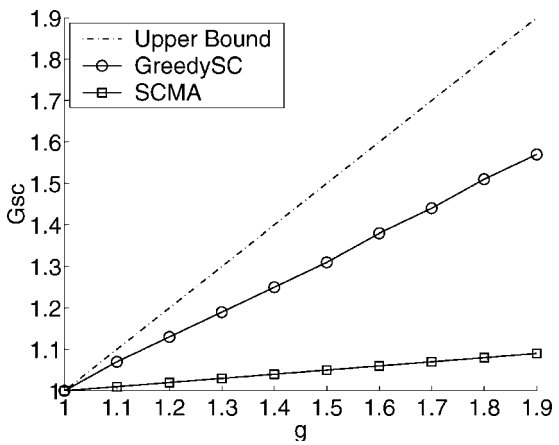


Fig. 13. Network capacity gain of SC as a function of g (SNR threshold = 10 dB).

An interesting observation is that there are still few links belonging to a single maximal clique when we increase SNR threshold to make link interference graph denser. We found the reason from the simulation traces. The average size of maximal cliques gets larger as the link interference graph becomes denser. As a result, most links are in the overlap of multiple maximal cliques. So we cannot expect that many links belong to a single maximal clique until the link interference graph is sufficiently dense (the extreme is one clique).

Next, we consider the effect of network density on the network capacity gain of SC scheduling. We keep all parameters of the previous experiment but reduce the area size to $800 \times 800\,\mathrm{m}^2$ to increase the network density. The average node degree now is 6.1. Figure 10 shows the network capacity gain of SC scheduling. The performance of GreedySC and SCMA is similar to that of $1000 \times 1000\,\mathrm{m}^2$. The reason is that both the number of links and interfering link pairs increases when the network density increases. As a consequence, the link interference graph is scaled up proportionally.

We also make simulations in grid networks. We place 25 wireless nodes in a 5×5 grid. There is a distance of 200 m (and 160 m in the later experiment) between two horizontally or vertically neighboring nodes. The other parameters are the same as that in random networks. SCMA does not work at all here because no links belong to a single maximal clique, while GreedySC provides the network capacity gain of 1.2 to 1.3 (Figures 11 and 12). In summary, SCMA often performs poorly in general WMNs.

Finally, simulations show that the network capacity gain of GreedySC is nearly a linearly increasing function of g and is closer to its theoretical upper bound (g) than SCMA. Figure 13 shows the average result for ten 25-node random networks in a square area of $1000 \times 1000\,\mathrm{m}^2$ when SNR threshold is 10 dB.

DOI: 10.1002/wcm

5. Related Work

Although MIMO and SC have been intensively studied on physical layer [1,5–8], there are only a few of works on MAC or network layer. In Reference [18], authors analyzed the strategies of exploiting different capabilities of MIMO links under different network conditions such as density, loss, and mobility. As a result, they proposed a novel routing protocol called MIR to level these strategies. In Reference [4], authors noticed that although SC can improve the network capacity, if it is not used judiciously, the problem of receiver overloading can occur which can degrade network capacity significantly. They proposed a heuristic scheduling algorithm and MAC protocol known as SCMA for utilizing the SC transmission. To our best knowledge, we are the first to provide a formal characterization of SC scheduling in WMNs. We derived the theoretical upper bound on the network capacity gain when the SC scheduling is used. Also, we proposed an SC scheduling algorithm GreedySC which greatly outperforms SCMA in the general settings of WMNs. In Reference [19], we only considered ideal channels. Furthermore, these results are extended to time-varying channels in this paper.

6. Conclusion and Future Work

In this paper, we characterized the network capacity of SC by formulating it as a network layer scheduling problem. We proved that the optimal network capacity of SC scheduling is at most g times that of NSC scheduling, where g is SC gain. We then proposed an efficient algorithm GreedySC so as to realize NSC and SC scheduling. Intensive simulations showed that the capacity gain achieved by GreedySC is close to its theoretical upper bound. Therefore, we can use GreedySC as a benchmark for other SC algorithms. In particular, we pointed out the poor performance of SCMA under the general settings of WMNs, indicating that there are plenty of rooms for enhancement.

There are some limitations in our work. First, our model assumes perfect packet scheduling. Actually, one can easily incorporate packet loss probability into our theorem and algorithm and get similar results. Second, we use a simplified physical layer model in our simulations, that is, the range-based interference model determined by SNR threshold. However, it remains an open problem to jointly consider packet scheduling and more detailed physical layer model in MIMO WMNs. Third, GreedySC currently is a centralized algorithm. The distributed implementation requires special supports from physical layer. For example, transmitters need to estimate accurately the number of streams that receivers can accommodate so as to transmit extra streams using SC. In sum, there is a lot of future work for researchers both in the area of network layer and physical layer before SC becomes a practical technique in MIMO WMNs.

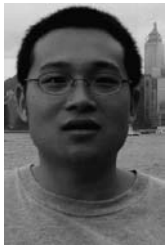
References

1. Gesbert D, Shafi M, Shiu D, Smith PJ, Naguib A. From theory to practice: an overview of MIMO space-time coded wireless systems. *IEEE Journal on Selected Areas in Communications* 2003; **21**: 281–301.
2. Mo J, So H-SW, Walrand J. Comparison of multi-channel MAC protocols. In *Proceedings of MSWiM* 2005; pp. 209–218.
3. Draves R, Padhye J, Zill B. Comparison of routing metrics for static multi-hop wireless networks. In *Proceedings of SIGCOMM* 2005; pp. 133–144.
4. Sundaresan K, Sivakumar R, Ingram MA, Chang T-Y. Medium access control in ad hoc networks with MIMO links: optimization considerations and algorithms. *IEEE Transactions on Mobile Computing* 2004; **3**: 350–365.
5. Demirkol MF, Ingram MA. Power-controlled capacity for interfering MIMO links. In *Proceedings of IEEE Vehicular Technology Conference*, Vol. 1, October 2001; pp. 187C191.
6. Demirkol MF, Ingram MA. Control using capacity constraints for interfering MIMO links. In *Proceedings of International Symposium on Personal, Indoor, and Mobile Radio Communications*, Vol. 3, 2002; pp. 1032–1036.
7. Demirkol MF, Ingram MA. Stream control in networks with interfering MIMO links. In *Proceedings of IEEE WCNC*, Vol. 1, 2003; pp. 343–348.
8. Jiang J-S, Demirkol MF, Ingram MA. Measured capacities at 5.8 GHz of indoor MIMO systems with MIMO interference. In *Proceedings of IEEE VTC*, Vol. 1, 2003; pp. 388–393.
9. Kodialam M, Nandagopal T. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *Proceedings of MobiCom* 2005; pp. 73–87.
10. Jain K, Padhye J, Padmanabhan VN, Qiu L. Impact of interference on multi-hop wireless network performance. In *Proceedings of MOBICOM* 2003; pp. 66–80.
11. Gupta P, Kumar PR. The capacity of wireless networks. *IEEE Transactions on Information Theory* 2000; **46**: 388–404.
12. Padhye J, Agarwal S, Padmanabhan VN, Qiu L, Rao A, Zill B. Estimation of link interference in static multi-hop wireless networks. In *Proceedings of USENIX IMC*, 2005; pp. 305–310.
13. Luo H, Lu S, Bharghavan V. A new model for packet scheduling in multihop wireless networks. In *Proceedings of ACM MOBICOM*, 2000; pp. 76–86.
14. De Couto D, Aguayo D, Bicket J, Morris R. High-throughput path metric for multi-hop wireless routing. In *Proceedings of MOBICOM*, 2003; pp. 134–146.
15. Kochut A, Vasan A, Shankar AU, Agrawala A. Sniffing out the correct physical layer capture model in 802.11b. In *Proceedings of IEEE ICNP*, 2004; pp. 252–261.
16. Xu K, Gerla M, Bae S. How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks. In *Proceedings of IEEE GLOBECOM*, Vol. 1, 2002; pp. 72–76.
17. Baynast A, Gurewitz O, Knightly EW. Measurement driven deployment of a two-tier urban mesh access network. In *Proceedings of MobiSys*, 2006; pp. 96–109.

DOI: 10.1002/wcm

18. Sundaresan K, Sivakumar R. Routing in ad-hoc networks with MIMO links. In *Proceedings of IEEE ICNP*, 2005.
19. Wang Y, Chiu DM, Lui JCS. Characterizing the capacity gain of stream control scheduling in MIMO wireless mesh networks. In *Proceedings of IFIP Networking*, 2007; pp. 311–321.

Authors' Biographies



Yue Wang received his B.S. degree in Mechanics from Beijing University, and his M.Sc. degree in Computer Science from Beijing University. Currently, he is a Ph.D. candidate in The Chinese University of Hong Kong. His recent research interests lie in resource allocation in wireless networks.

Dah Ming Chiu received his B.Sc. degree in Electrical Engineering from Imperial College, University of London, and his Ph.D. from Harvard University, in 1975 and 1980, respectively. He was a Member of Technical Staff with Bell Labs from 1979 to 1980. From 1980 to 1996, he was a Principal Engineer, and later a Consulting Engineer at Digital Equipment Corporation. From 1996 to 2002, he was with Sun Microsystems Research Labs. Currently, he is a Professor in the Department of Information Engineering in The Chinese University of Hong Kong. He is known for his contribution in studying network congestion control as a resource allocation problem, the fairness index, and analyzing a distributed algorithm (AIMD) that became the basis for the congestion control algorithm in the Internet. His current research interests include economic issues in networking, P2P networks, network traffic monitoring and analysis, and resource allocation and congestion control for the Internet with expanding services. Two recent papers he co-authored with students have won best student paper awards from the IFIP Performance Conference and the IEEE NOMS Conference. Recently, Dr Chiu has served on the TPC of IEEE Infocom, IWQoS, and various other conferences. He is a member of the editorial board of the *IEEE/ACM Transactions on Networking*, and the *International Journal of Communication Systems* (Wiley).

John C. S. Lui received his Ph.D. in Computer Science from UCLA. He worked in the IBM San Jose/Almaden Research Laboratory before joining the Chinese University of Hong Kong. Currently, he is the Chair of the Computer Science and Engineering Department at CUHK and he leads the Advanced Networking and System Research Group. His research interests span both in systems as well as in theory/mathematics in computer communication systems. John received various departmental teaching awards and the CUHK Vice-Chancellor's Exemplary Teaching Award, as well as the co-recipient of the Best Student Paper Awards in the IFIP WG 7.3 Performance 2005 and the IEEE/IFIP Network Operations and Management (NOMS) Conference.

DOI: 10.1002/wcm