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] Dept. of Computer Science & Engineering, The Chinese University of Hong Kong
{ywang, cslui}@cse.cuhk.edu.hk
[2] Dept. of Information Engineering, The Chinese University of Hong Kong
dmchiu@ie.cuhk.edu.hk

Abstract. Stream control has recently attracted attentions in the research of MIMO wireless networks as a potential way to improve network capacity. However, inappropriate use of stream control may significantly degrade network capacity as well. In this paper, we provide the first formal study on stream control scheduling in MIMO wireless mesh networks. We derive the theoretical upper bound on network capacity gain of stream control 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 stream control scheduling algorithm SCMA under the general settings of wireless mesh networks. This formal characterization provides a deeper understanding of steam control scheduling in MIMO wireless mesh networks.

Key words: MIMO, stream control, network capacity, scheduling

1   Introduction

Multiple-input multiple-output (MIMO) technology is regarded as one of the most significant breakthroughs 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 wireless mesh networks (WMNs) where *network capacity* is the main concern.
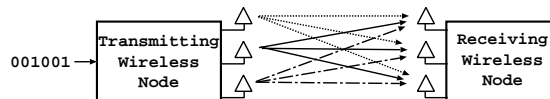

Fig. 1. MIMO Wireless nodes with three antennas.

Generally speaking, MIMO can be represented as a multiple transmitters and multiple receivers system. As illustrated in Fig. 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, each MIMO node has $K \geq 1$ antennas. Link capacity grows *linearly* with K since there are K independent streams. This link capacity gain is referred to as *spatial multiplexing gain*.[1] 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 additional streams (treated as noise by the receiver) is far weaker than that of the successfully decoded streams. Otherwise, it is not possible to decode any stream[3].

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 *Stream Control (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. Recent studies [4][5][6] show that SC can be applied to multiple *interfering* links to improve network capacity.
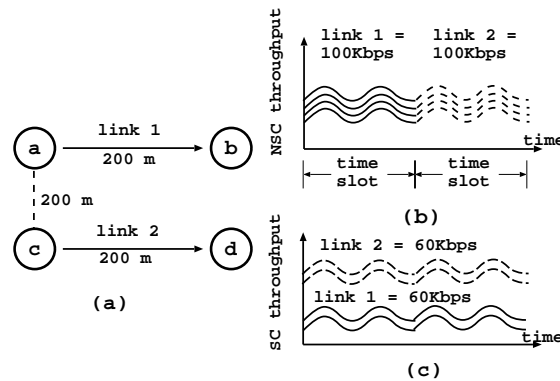

Fig. 2. An Illustration of Network Capacity for NSC and SC scheduling

An example to illustrate the improvement in network capacity by SC is shown in Fig. 2. 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

---

[1] 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 [3] 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.

are placed close to each other and thus get mutual interference. Assume there are four antennas on each node ($K = 4$). Under NSC scheduling, each wireless node transmits 4 streams at an aggregate rate of 100 Kbps. The maximum network capacity is 100 Kbps when the two links transmit in an alternate manner (Fig. 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, these streams have different transmission rate [2]. For example, the ratio of transmission rate for the 4 streams is $1.0 : 0.8 : 0.7 : 0.5$. Each link can choose the two best streams to transmit with an aggregate rate of $\frac{1.0+0.8}{1.0+0.8+0.7+0.5} \cdot 100 = 60$ Kbps (this is called *stream selection*), achieving a network capacity of 120 Kbps (Fig. 2(c)).
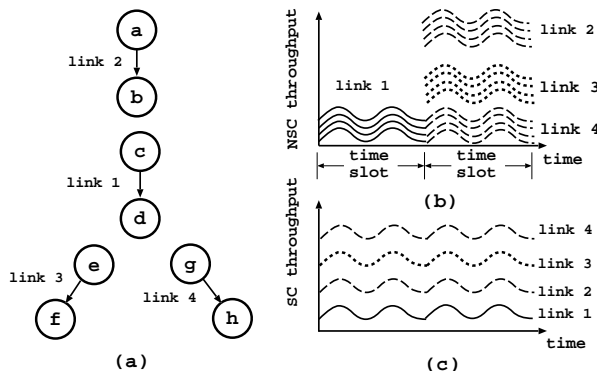

Fig. 3. Illustration of Receiver Overloading Problem

Previous work [4][5][6] showed that MIMO with stream control can increase the network capacity by $20 - 65\%$ for a set of *mutually interfering* links. However, it was noted in [3] that when stream control 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 Fig. 3. Here, each wireless node has $K = 4$ antennas. Link 1 interferes with link $2, 3$ and 4 while the last three links do not interfere with each other. If link 1 transmits 4 streams in one time slot and the other three links each transmit 4 streams in the next time slot, it will result in an average of 8 streams per time slot. If all four links use SC, then each can only transmit one stream (upper bounded by 4 streams for link 1). In this case, only 4 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 full transmissions of the other three links.

Sundaresan et al proposed a heuristic algorithm called *SCMA (Stream-Controlled Multiple Access)* for using SC scheduling in multihop wireless networks [3]. 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). In other cases, NSC scheduling should be used. SCMA can work well for dense link interference graph where links likely belong to a single maximal clique. In this paper, we provide the first formal characterization on the capacity gain of SC *from the angle of scheduling*. In particular, we

address 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 to NSC scheduling. In Section 3, we present a greedy scheduling algorithm when NSC or SC is used. In Section 4, we discuss the simulation results and compare the network capacity of NSC and SC scheduling as well as that of SCMA. Finally, Section 5 discusses the limitations of our work and concludes the paper.

2   Upper Bound on Capacity Gain of Stream Control Scheduling

In this section, we first define the terminologies used in wireless networks, then we propose a simplified model for MIMO WMNs and derive the theoretical upper bound on the network capacity gain of SC scheduling.

We consider a multihop wireless network with n *static* nodes. Each node has multiple omnidirectional antennas that operate on one channel. We model it as a directed graph $G = (V, E)$ where V represents the set of wireless nodes and E represents the set of wireless links. Each wireless node has the same transmission power and receiving sensitivity, resulting in the same *maximum* transmission range by a certain path loss model (e.g. two-ray ground model). Therefore, two nodes can form a link when the distance between them is not greater than the maximum transmission range.

We use the range-based interference model[7] that is shown to be realistic by [8]. Let $L_{1,2}$ ($L_{3,4}$) be the link between node 1 and 2 (node 3 and 4). Let d_{ij} denote the distance between node i and node j. $L_{1,2}$ is interfered by $L_{3,4}$ when node 1 or node 2 is interfered by node 3 or node 4, precisely, $min(d_{1,3}, d_{1,4}, d_{2,3}, d_{2,4}) \leq S \cdot d_{1,2}$. Here we call S the *interference range factor* that is determined by the SNR threshold for successful receiving and the underlying path loss model. Note that the above reception and interference models are actually implemented in ns-2 simulator.
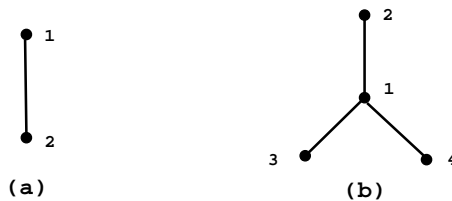

Fig. 4. Examples of Link Interference Graph

To represent the interference between links, we introduce the *link interference graph* $G^I = (V^I, E^I)$, also known as flow contention graph [9]. we rename it here to avoid confusion with end-to-end flows. 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 can not transmit at the same

time. Fig. 4 shows the link interference graphs for the networks in Fig. 2 and Fig. 3, respectively.

To derive the upper bound on network capacity, we assume that the system operates in a synchronous time-slotted mode. Links transmit at the beginning of each time slot in a TDMA mode. They can transmit at the same time, provided that they do not interfere with each other. Besides, we do not consider packet losses caused by fading. This is a reasonable assumption since MIMO can significantly mitigate fading losses. Next, we assume that there are $K > 1$ antennas on each wireless node, and thus a link can simultaneously transmit at most K independent data streams. We introduce the *stream control gain* g, a factor for all wireless links. The total capacity of multiple mutually interfering links using SC is at most g times that using NSC. $1 < g < 2$ because $g \geq 2$ will violate the physical constraint (i.e. multiple interfering links can not transmit simultaneously at their full rate). Finally, we define network capacity gain using SC. Assume that the traffic pattern (i.e., the ratio of average link rate) is fixed. Let C_{NSC} and C_{SC} be the respective network capacity for NSC and SC scheduling. We have $C_{SC} = G_{SC} \cdot C_{NSC}$, where G_{SC} is defined as the *network capacity gain of SC scheduling*.

We now present the interference constraint in MIMO wireless networks. Without the loss of generality, we normalize the capacity of a MIMO link to 1 and define the link transmission rate as a fraction of 1. Let $\boldsymbol{y}(t)$ be the link transmission rate vector at time t (i.e., $y_e(t)$ is the transmission rate of link e), and $N(e)$ the set of neighbors (i.e., interfering links) of e. If e transmits at time slot t, the sum of transmission rate in $N(e) \cup e$ must be less than or equal to g, otherwise, receivers can not decode the mixed signals correctly. Formally, we define the *MIMO interference constraint* as,

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

The function $\mathbf{1}$ is an indicator function, returning 1 when $y_e(t) > 0$, and returning 0 otherwise. For NSC scheduling, $g = 1$ and $y_e(t)$ is 0 or 1. For SC scheduling, $1 < g < 2$ and $y_e(t)$ is a real number between 0 and 1 since a MIMO link can properly choose the number of streams and their transmission rate. Thus, NSC scheduling can be seen as a special case of SC scheduling.

We now state and prove the fundamental theorem of SC scheduling.

Theorem 1. *Assume that the traffic pattern (i.e., the ratio of average link rate) is fixed, the network capacity of SC scheduling is at most g times that of NSC scheduling.*

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

Step 1: We *only* need to prove that the capacity of NSC scheduling is greater than or equal to that of SC(1) scheduling, since the capacity of SC(1) scheduling is at least equal to that of NSC scheduling. As we are interested in the average link rate, it is sufficient to prove that for any feasible link transmission rate vector $\boldsymbol{y}^{SC}$ of SC(1) at some time slot, there is a sequence of link transmission rate vectors of NSC $\{\boldsymbol{y}^{NSC}(t) | t = 1, 2, ..., T\}$

for T time slots, whose average is greater than or equal to $\boldsymbol{y}^{SC}$. In other words, we scale $\boldsymbol{y}^{SC}$ by a large number of time slots T to make it an integer workload vector $\boldsymbol{u} = \boldsymbol{y}^{SC} \cdot T$ (real number can be approximated by rational), and prove that $\boldsymbol{u}$ can be scheduled in T time slots using NSC. We prove this by induction for the number of links m in the network. When $m = 1$ (u and y^{SC} are scalars), we can easily schedule u in T since $y^{SC} \leq 1$. Assume that we can schedule $\boldsymbol{u}$ for m links in T. We consider adding the $(m+1)^{th}$ *active* link e. The neighbors of e consume at most $\sum_{e' \in N(e)} u_{e'}$ time slots, and we can use the remaining $(T - \sum_{e' \in N(e)} u_{e'})$ time slots (which is greater than u_e by Eq. (1) for $g = 1$) to schedule u_e. So, we can schedule $\boldsymbol{u}$ for $m + 1$ links in T.

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

Remark: SC scheduling can not provide greater than g times network capacity gain although it provides a finer scheduling scheme. Therefore, if g is of small value, SC is not worthwhile to consider as the computational cost of SC scheduling is higher than that of NSC scheduling. Besides, the upper bound is not always achievable due to the potential problem of receiver overloading.

3   A Stream Control Scheduling Algorithm

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 a centralized SC scheduling algorithm and obtain the capacity gain in a practical way. It is important to point out that this algorithm provides a *lower bound* on the achievable network capacity.

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

$$C_{NSC} = \frac{\sum_{e \in E} u_e}{T^{NSC}}, \tag{2}$$

$$C_{SC} = \frac{\sum_{e \in E} u_e}{T^{SC}}, \tag{3}$$

and the network capacity gain of SC scheduling is,

$$G_{SC} = \frac{C_{SC}}{C_{NSC}} = \frac{T^{NSC}}{T^{SC}}. \tag{4}$$

To schedule $\boldsymbol{u}$, we design a greedy scheduling algorithm *GreedySC*. At each time slot, the algorithm finds schedulable links of maximal workloads greedily so as to minimize the total number of time slots used. Fig. 5 depicts the algorithm. The inputs to this algorithm are the link interference graph G^I, the workload vector $\boldsymbol{u}$ and stream control

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

$T_s := 0$;
$A := E$;
While $A \neq \phi$
    $T_s := T_s + 1$;
    $\boldsymbol{y} := \boldsymbol{0}$;
    Sort A in the deceasing order of u;
  / Step 1. using NSC*/*
    For each $e \in A$
      If $y_{e'} = 0, \forall e' \in N(e)$
        $y_e := 1$;
        $u_e := u_e - 1$;
        If $u_e \leq 0$
         $A := A - \{e\}$;
        fi
      fi
    end for
  / Step 2. using SC*/*
    For each $e \in A$ and $y_e = 0$
      If $y_{N(e)} \leq 1$ and $(y_{N(e')} \leq 1, \forall e' \in N(e)$ and $y'_e > 0)$
        $y_e := g - 1$;
        $u_e := u_e - (g - 1)$;
        If $u_e \leq 0$
         $A := A - \{e\}$;
        fi
      fi
    end for
end while

Fig. 5. Specification of GreedySC Scheduling Algorithm

gain g. Here, T_s is the number of time slots expended, A is the set of uncompleted links, and $y_{N(e)}$ is the sum of transmission rate for links in $N(e)$. Note that GreedySC can also be applied to NSC scheduling by letting $g = 1$.

The algorithm is described as follows. For each time slot, we first sort the links in decreasing order of the remaining workloads. In *Step 1*, we use the NSC mode to schedule links. That is, we iteratively pick a link e from the sorted list. If no neighboring links of e have been scheduled, we schedule e for transmission ($y_e := 1$). 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. Again, we iteratively pick a link e from the sorted list that was *un-schedulable* in step 1. If Eq. (1) can be satisfied for e and all its neighboring links, we schedule e for SC transmission ($y_e := g - 1$). 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 for one time slot. The above process is repeated in the next time slot until all the traffic workload 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, a SC link e' ($y_{e'} = g - 1$) must neighbor some NSC link e ($y_e = 1$). Otherwise, e' can be scheduled using NSC. And all neighboring links of e or e' can not be scheduled to satisfy Eq. (1). Therefore, for stream selection, the two links can select streams of other rate to achieve $y_e + y_{e'} = g$ (e.g., $y_e = y_{e'} = \frac{g}{2}$).

GreedySC achieves higher network capacity than SCMA. To understand this, let us describe SCMA first. In short, multiple interfering links can be scheduled simultaneously using SC, only when they all belong to a *single* maximal clique[2]. SCMA can not cause receiver overloading, because links using SC do not affect the transmission of other maximal cliques. Besides, SCMA can be implemented distributedly [3]. For example in Fig. 4 (a), link 1 and 2 belong to the single maximal clique $\{1, 2\}$, so they can be scheduled simultaneously using SC. In Fig. 4 (b), there are no two interfering links both belonging to a single maximal clique, as link 1 belong to 3 maximal cliques $\{1, 2\}$, $\{1, 3\}$ and $\{1, 4\}$. So SCMA can not 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 Fig. 4 (b), GreedySC will schedule link 1 using NSC ($y_1 = 1$) and link 2 using SC ($y_2 = g - 1$). Formally, consider any two links a and b belonging to a single maximal clique. Suppose a is scheduled using NSC in step 1 of GreedySC. It is easy to see that b can be scheduled using SC in step 2 of GreedySC. In this way, GreedySC covers the case of SCMA and thus achieves higher network capacity.

4   Performance Evaluation

This section shows the simulation results in *general* WMNs. We test the 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 (1) *network density* and (2) *interference level* in current practices. We measure network density by node degree, i.e., the number of links a wireless node has, and measure interference level by the interference range factor S. The node degree in a WMN is typically around 2 to 6, otherwise, there will be too much interference. For example, our simulation shows that a link is interfered by about 16 other links when the average node degree is 3. We vary the SNR threshold for successful receiving from 6 to 10dB [11] depending on the environment and coding scheme. And the resulting interference range factor S varies from 1.4 to 1.8 by the two-ray ground path loss model[10]. Note that higher SNR threshold implies more interfering links. We designed a TDMA simulator to generate WMNs and schedule link workloads.

First, we randomly place 25 wireless nodes in an area of 1000m×1000m, with the maximum transmission range of 250m. The average node degree is 4.2. $g = 1.5$ for all wireless links. We generate 100 flows with sources and destinations randomly selected.

---

[2] Here a maximal clique is a maximal fully connected subgraph of G^I. All links of a maximal clique interfere (neighbor) with each other. A link may belong to multiple maximal cliques.
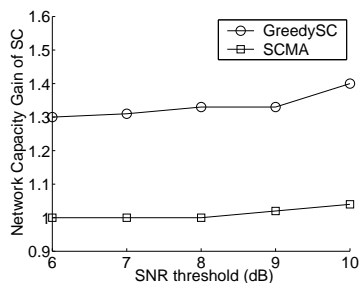
Fig. 6. Avg. Network Capacity Gain of SC for 1000m×1000m Random Topologies ($g = 1.5$)
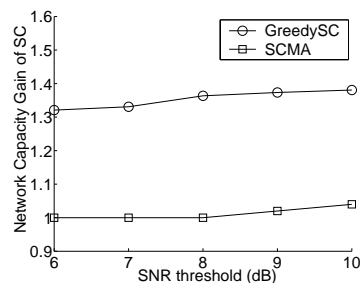
Fig. 7. Avg. Network Capacity Gain of SC for 800m×800m Random Topologies ($g = 1.5$)

Each flow has the same traffic demand of 1000. And we map the flow traffic demand x into link traffic workload u by $u = Rx$, where R is the routing matrix ($R_{ij} = 1$, if flow j is on link i; $R_{ij} = 0$, otherwise). We use the shortest path routing for these flows and run the simulation for tens of times to get the average.

Fig. 6 shows the network capacity gain of SC scheduling G_{SC} when GreedySC and SCMA are used. SCMA provides little capacity gain because there are few links belonging to a single maximal clique, while GreedySC provides $30-40\%$ improvement on network capacity compared with NSC scheduling.

An interesting observation is that there are still a small number of links belonging to single maximal cliques when we increase SNR threshold to make link interference graph denser (introduce more edges in G^I). 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 can not 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 800m×800m to increase the network density. The average node degree now is 6.1. Fig. 7 shows the network capacity gain of SC scheduling. The performance of GreedySC and SCMA is similar to that of 1000m×1000m. 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 did simulation in grid topologies. In this experiment, we place 25 wireless nodes in a 5×5 grid. There is a distance of 200m (and 160m later) between two horizontally or vertically neighboring nodes. The other parameters are the same as that in random topologies. 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 (Fig. 8 and Fig. 9). In short, SCMA performs poorly in general WMNs.

Finally, simulation shows that the network capacity gain of GreedySC is a linearly increasing function of g and is closer to its theoretical upper bound (g) than SCMA.
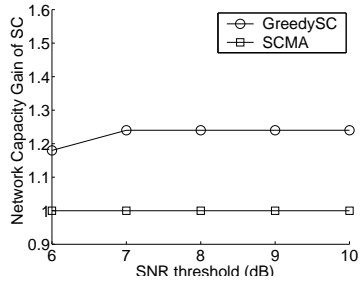
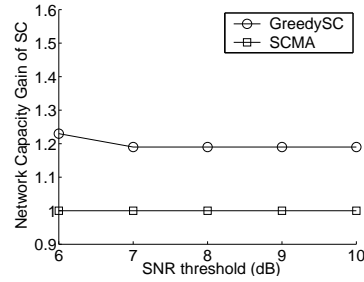**Fig. 8.** Avg. Network Capacity Gain of SC for 1000m×1000m Grid Topologies ($g = 1.5$)

Fig. 9. Avg. Network Capacity Gain of SC for 800m×800m Grid Topologies ($g = 1.5$)
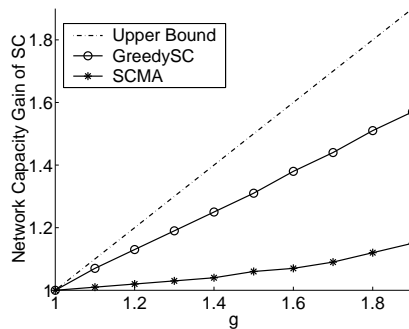

Fig. 10. Network capacity gain of SC as a function of g (SIR threshold = 10dB)

Fig. 10 shows the result for 25-node random topologies in an area of 1000m×1000m when SIR threshold is 10dB.

5 Conclusion and Future Work

In this paper, we characterized the network capacity of SC by formulating it as a mere scheduling problem. We proved that the network capacity of SC scheduling is at most g times that of NSC scheduling, where g is stream control 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 losses into our theorem and algorithm to get similar results. Second, we use a simplified physical layer model and do not account for time-varying fading channels. 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. D. Gesbert, M. Shafi, D. Shiu, P.J. Smith, and A. Naguib, "From Theory to Practice: An Overview of MIMO Space-Time Coded Wireless Systems," *IEEE J. Selcted Areas in Comm.*, vol. 21, pp. 281-301, 2003.
2. David Tse and Pramod Visvanath, "Fundamentals of Wireless Communication", *Cambridge University Press*, 2005.
3. K. Sundaresan, R. Sivakumar, M.A. Ingram, and T.-Y. Chang, "Medium access control in ad hoc networks with MIMO links: optimization considerations and algorithms," *IEEE Trans. on Mobile Computing*, vol. 3, pp. 350 - 365, 2004.
4. M.F. Demirkol and M.A. Ingram, "Control Using Capacity Constraints for Interfering MIMO Links," Procs., *Intl Symp. Personal, Indoor, and Mobile Radio Comm.*, vol. 3, pp. 1032-1036, 2002.
5. M.F. Demirkol and M.A. Ingram, "Stream Control in Networks with Interfering MIMO Links," Procs., *IEEE WCNC*, vol. 1, pp. 343-348, 2003.
6. J.-S. Jiang, M.F. Demirkol, and M.A. Ingram, "Measured Capacities at 5.8 GHz of Indoor MIMO Systems with MIMO Interference," Procs., *IEEE VTC*, vol. 1, pp. 388-393, 2003.
7. P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. on Info. Theory*, vol. 46, pp. 388-404, 2000.
8. J. Padhye, S. Agarwal, V. N. Padmanabhan, Lili Qiu, A. Rao and Brian Zill, "Estimation of Link Interference in Static Multi-hop Wireless Networks," Procs., *USENIX IMC*, pp. 305-310, 2005.
9. H. Luo, S. Lu, V. Bharghavan, "A new model for packet scheduling in multihop wireless networks," Procs., *ACM MOBICOM*, pp. 76-86, 2000.
10. K. Xu, M. Gerla, and S. Bae, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks," Procs., *IEEE GLOBECOM*, vol. 1, pp. 72-76, 2002.
11. A. Kochut, A. Vasan, A. U. Shankar, A. Agrawala, "Sniffing out the correct physical layer capture model in 802.11b," Procs., *IEEE ICNP*, pp. 252-261, 2004.