

# Employing the One-Sender-Multiple-Receiver Technique in Wireless LANs

Zhenghao Zhang, *Member, IEEE*, Steven Bronson, Jin Xie and Hu Wei

**Abstract**—In this paper, we study the One-Sender-Multiple-Receiver (OSMR) transmission technique, which allows one sender to send to multiple receivers simultaneously by utilizing multiple antennas at the sender. To study the physical layer characteristics of OSMR, we implement a prototype OSMR transmitter/receiver with GNU Software Defined Radio, and conduct experiments in a university building. Our results are positive and show that wireless channels allow OSMR for a significant percentage of the time. Motivated by our physical layer study, we propose extensions to the 802.11 MAC protocol to support OSMR transmission, which is backward compatible with existing 802.11 devices. We also note that the AP needs a packet scheduling algorithm to efficiently exploit OSMR. We show that the scheduling problem without considering the packet transmission overhead can be formalized as a Linear Programming problem, but the scheduling problem considering the overhead is NP-hard. We then propose a practical scheduler based on a two-phase algorithm that can also handle channel fluctuations. We test the proposed protocol and algorithm with simulations driven by traffic traces collected from wireless LANs and channel state traces collected from our experiments, and the results show that OSMR significantly improves the downlink performance.

**Index Terms**—One-Sender-Multiple-Receiver, Packet Scheduling, Wireless LAN.

## I. INTRODUCTION

Wireless Local Area Networks (LAN) offer convenient access to the Internet. In this paper, we study the One-Sender-Multiple-Receiver (OSMR) technique and its applications to wireless LANs. OSMR allows one sender to send distinct information to multiple receivers simultaneously on the same frequency channel by utilizing multiple antennas at the sender [2]. In wireless LANs, the Access Point (AP) may use OSMR to support multiple nodes more efficiently.

OSMR is basically *Multi-User Multiple-Input-Multiple-Output (MU-MIMO)* on the downlink [2], [3], [4]. As MU-MIMO also includes the uplink case, in this paper, we use the term OSMR for clarity. OSMR is different from the single-user MIMO transmission adopted in 802.11n [16], because the single-user MIMO still supports only one-to-one transmissions, while OSMR supports one-to-many transmissions. Allowing one-to-many transmissions will help achieving an overall higher efficiency. For example, suppose nodes *A* and *B* both have very strong channels and are already operating at the highest data rate supported by the hardware. With OSMR, the AP may be able to transmit to them simultaneously, still at the highest data rate, hence doubling the downlink

throughput. Code Division Multiplexing Access (CDMA) also allows multiple nodes to communicate simultaneously on the same frequency. Basically, OSMR takes advantage of multiple antennas and is more efficient in utilizing the bandwidth than CDMA. A CDMA transmitter has to spread the signal bandwidth to a much larger bandwidth [2], which is not required with OSMR.

The number of simultaneous receiving nodes with OSMR is limited by the number of antennas at the sender. In this paper, we focus on the case when OSMR involves two receiving nodes, which is of practical interests in wireless LANs because the AP usually has a limited number of antennas due to cost considerations. To study the physical layer characteristics and the practicability of OSMR for wireless LANs, we implement a prototype OSMR transmitter/receiver with GNU Software Defined Radio (SDR) [14], [15] that allows one sender to send to two receivers simultaneously. OSMR transmission depends on the channel states of the receivers, because it requires the sender to process the signals according to the channel states. The most critical questions related to the practicability of OSMR include: 1) how likely are two receivers *compatible*, where compatible means that the channel states of two receivers allow them to receive from the sender simultaneously at non-trivial data rates, and 2) whether the channel fluctuation speed is slow enough, such that the measured channel states remain valid until the sender finishes sending. Fortunately, our experiments reveal that two receivers are usually compatible for a significant percentage of the time. Also, in the indoor environment, the channel is typically stable during the OSMR transmission.

Motivated by our physical layer study, we propose OSMR as an enhancement technique to wireless LANs. We propose a simple extension to the 802.11 MAC protocol to support OSMR transmissions. The extension is backward compatible, i.e., it allows the OSMR-capable nodes to coexist with legacy 802.11 nodes without interfering with each other. We also note that to fully exploit OSMR, the AP needs an algorithm to determine which packet(s) to send in order to optimize the performance, e.g., maximizing the throughput under the fairness and quality of service constraints mandated by the upper layer. We show that the scheduling problem without considering the packet transmission overhead can be formalized as a Linear Programming problem, but the scheduling problem considering the overhead is NP-hard. We then propose a practical scheduler based on a two-phase algorithm that can also handle channel fluctuations. We test the proposed protocol and algorithm with simulations driven by traffic traces collected from wireless LANs and channel state traces

collected from our experiments, and the results show that OSMR significantly improves the downlink performance.

The rest of the paper is organized as follows. Section II describes our implementation of OSMR and the experiments. Section III discusses the extension to the 802.11 MAC protocol. Section IV discusses the packet scheduler. Section V evaluates the proposed protocol and the packet scheduling algorithm with simulations. Section VI discusses the case when the number of receivers is more than two. Section VII discusses the related works. Section VIII concludes the paper.

## II. OSMR IMPLEMENTATION AND EXPERIMENTS

In this section, we describe our implementation of OSMR and the experiments.

### A. OSMR Background and Implementation

OSMR can be realized by a *zero-forcing* strategy [2], as explained in the following. We assume the channel is flat-fading. If the sender sends a data symbol  $d$ , the receiver will receive  $y = hd + n$ , where  $h$  is the complex channel coefficient and  $n$  is the noise. If there are two receivers and the sender has two antennas, the sender can send two different symbols denoted as  $x_1$  and  $x_2$  on antenna 1 and antenna 2, respectively. Suppose the channel coefficient from antenna  $j$  to receiver  $i$  is  $h_{ij}$  for  $i, j \in \{1, 2\}$ . Let the received signal at receiver  $i$  be  $y_i$ , and let the noise at receiver  $i$  be  $n_i$ . The received signal is a linear combination of the signals sent from each antenna multiplied by the channel coefficients, plus the noise:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}$$

We use  $\mathbf{H}$  to denote the channel matrix. Suppose a *processing matrix*

$$\mathbf{U} = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}$$

can be found such that  $\mathbf{h}_1 \mathbf{u}_2 = 0$  and  $\mathbf{h}_2 \mathbf{u}_1 = 0$ , where  $\mathbf{h}_i$  denotes a row vector of  $\mathbf{H}$  and  $\mathbf{u}_j$  denotes a column vector of  $\mathbf{U}$ . If such matrix can be found, let  $d_1$  and  $d_2$  denote the data that should be sent to receiver 1 and receiver 2, respectively. We let

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$

Thus, receiver 1 will receive  $\mathbf{h}_1(d_1 \mathbf{u}_1 + d_2 \mathbf{u}_2) + n_1 = d_1 \mathbf{h}_1 \mathbf{u}_1 + n_1$ . Similarly, receiver 2 will receive  $d_2 \mathbf{h}_2 \mathbf{u}_2 + n_2$ . Therefore, distinct data is sent to each receiver. In this paper,  $\mathbf{h}_i \mathbf{u}_i$  is referred to as the *effective channel* of receiver  $i$ , the strength of which determines the receiving data rate. We say two receivers are *compatible* if a processing matrix can be found such that the receiving data rates are non-trivial, i.e., above the minimum data rate in the network such as 6 Mbps in 802.11g networks.

We implement a prototype OSMR transmitter/receiver based on the above zero-forcing strategy in about 2,000 lines of C++ and Python code using GNU Software Defined Radio (SDR) [14], [15]. There are two key components in the implementation, namely the channel estimation and the choice of the processing matrix. Basically, we let the sender send

a *channel estimation frame*, which consists of a sequence of special symbols, with which the receivers estimate their channels and then send the *channel estimation reports* back to the sender. With the channel information, the sender can select a processing matrix and send data. The details about the two components are described in the Appendix.

### B. OSMR Experiments

We employ our prototype OSMR transmitter/receiver to find the feasibility of OSMR. The first key question is: How likely are two receivers compatible? Because the wireless channel constantly fluctuates, two receivers may be compatible at some times while not compatible at other times. For OSMR to be applicable to wireless LANs, the percentage of the time when the receivers are compatible must be non-trivial. In our experiments, the OSMR transmission is centered at 2.42 GHz, which lies within the ISM band used by the 802.11g networks. The modulation is Differential Binary Phase Shift Keying (DBPSK) and the symbol rate is 500,000 symbols per second, resulting in a bit rate of 0.5 Mbps. We refer to the OSMR sender as  $S$  and the two OSMR receivers as  $R1$  and  $R2$ , respectively. In our experiments,  $R1$  and  $R2$  are turned on first. The OSMR transmission is then carried out in three steps:

- 1)  $S$  transmits the channel estimation frames for 0.5 second, then switches to the listening mode to wait for the channel estimation reports from  $R1$  and  $R2$ .
- 2) Both  $R1$  and  $R2$  wait until  $S$  stops sending. Then,  $R1$  sends the channel estimation report to  $S$  for 0.01 second, then switches to the listening mode to wait for the data frames. After  $S$  stops sending,  $R2$  waits for 0.01 second, then sends the channel estimation report to  $S$  for 0.01 second, then switches to the listening mode to wait for the data frames.
- 3) After getting both channel estimation reports,  $S$  waits for 0.01 second, then switches to the transmitting mode and sends distinct data frames to each receiver for 1 second. One data frame is 1524 bytes with 1500 bytes of randomly generated data and 24 bytes as the preamble and the frame header.

Our experiments are conducted in a university building. The devices used in our experiments are shown in Fig. 1. We pick 10 sender locations, and for each sender location, we conduct a set of 4 OSMR experiments at randomly selected receiver locations. Therefore, we conduct a total of 40 experiments. In the experiments, the distances between the sender and the receivers are between 6 to 30 feet. The sender location and the receiver locations in one set of the experiments, for example, are shown in Fig. 2. In each experiment, OSMR transmissions are attempted with random intervals between 2 to 5 seconds. Therefore, we basically randomly sample the channels and find the percentage of time the receivers are compatible. An OSMR transmission is considered successful if the both receivers get the first 3 data frames with no bit error. Only the first 3 frames are considered because the sender processes the signal based on the channel estimation reports received before the transmission, but the channel states may have drifted after



Fig. 1. Devices used in the experiments.

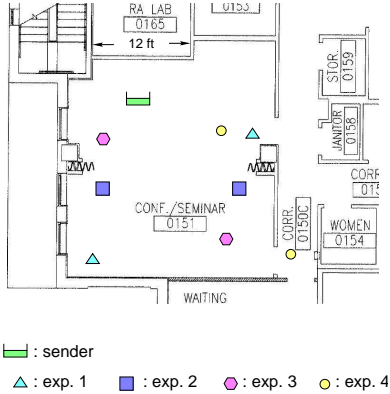


Fig. 2. The sender location and the receiver locations in one set of the experiments.

several frames, which leads to receiving errors that should not be interpreted as incompatibility.

As a quantitative measure, we define the *compatibility ratio* of an experiment as the number of successful OSMR transmissions over the number of all OSMR transmissions *that are carried out*. In our experiments, an OSMR transmission is carried out if the sender gets both channel estimation reports from the receivers. OSMR transmissions are not always carried out because of the limitations of our GNU SDR platform. Basically, the switching between the transmitting mode and the receiving mode could take a non-trivial amount of time depending on the instantaneous state of the operating system. As a result, it could happen that two receivers send reports at the same time, which results in a collision. If the sender does not get the channel estimation reports from both receivers, the sender aborts the transmission. We report the results of 35 experiments in each at least 25 OSMR transmissions are carried out, and show the Cumulative Distribution Function (CDF) of the compatible ratio in Fig. 3. We can see that roughly, the compatible ratio is uniformly distributed in  $[0, 0.9]$ . We note that in our experiment setting, the throughput gain is  $a\%$  if  $a\%$  of the channels are compatible. This is because in  $a\%$  of the times, the AP can use OSMR to double the throughput; in the other cases, the AP can use one-to-one transmissions and achieve the baseline throughput.

### C. Wireless Channel Characteristics Relevant to OSMR

As mentioned earlier, the second key question is the stability of the channel. As the wireless channel may fluctuate randomly, before starting the OSMR transmission, the sender should have the up-to-date channel states from the receivers to calculate the processing matrix. Because the sender does not have further feedbacks from the receiver, in order for the

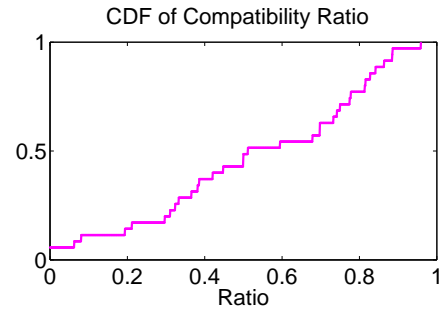


Fig. 3. The CDF of the compatibility ratio found in the experiments.

OSMR transmission to be successful, the drift of the channel during the frame transmission time must be limited. Therefore, we conduct experiments to find the channel characteristics in the indoor environments. In our experiments, there are one sender and one receiver, where the sender has two antennas and the receiver has one antenna. We pick 10 sender locations, and for each sender location, we randomly pick 4 receiver locations with both line-of-sight and non-line-of-sight paths. The sender transmits the OSMR channel estimation frame every 1 ms for a total of 50 seconds, and the receiver simply records the received samples. We show the fluctuation of the *channel ratio* in the experiments, which is used to determine the processing matrix and is defined as ratio of the channel coefficient from sender antenna 2 over that of antenna 1. If the ratio drifted from  $ae^{j\phi}$  to  $a'e^{j\phi'}$ , the drift percentage of the magnitude is defined as  $\frac{|a'-a|}{a} \times 100\%$ , and the drift of the phase is defined as the difference between  $\phi'$  and  $\phi$ . The CDFs of the channel ratio drift after 1 ms, 10 ms, 100 ms, and 1000 ms are shown in Fig. 4. We can see that for more than 90% of the times, after 10 ms, the magnitude drifts less than 10%, and phase drifts less than  $\frac{\pi}{18}$ . As an example, Fig. 5 shows a typical trace of the channel ratio magnitude. The fast fluctuation at the beginning of the trace is caused by fast movements of human beings; the rest of trace are relatively stable.

### D. Remarks

Our experiments proved that in the indoor environments, for a significant percentage of time, OSMR transmissions are possible and the wireless channels are stable. Yet, the experimental results are implementation dependent. The results reported in this section on OSMR transmissions are based on our prototype implementation with software defined radio, which has limitations at the current stage for OSMR. First, it does not allow fast switching between the transmitting mode and receiving mode, and the sender has to wait for around 20 ms before starting the transmission, because the receivers have to switch from the receiving mode to the transmitting mode. Therefore, the successful OSMR transmissions reported in our experiments belong to those cases when the channels allow OSMR transmissions *and* do not drift significantly after the channel estimation, which is a subset of the cases when the channels allow OSMR transmissions. In this regard, with a faster hardware implementation, the compatibility ratios may be higher than that in Fig. 3. However, secondly, software defined radio relies on software to process the signals, and the

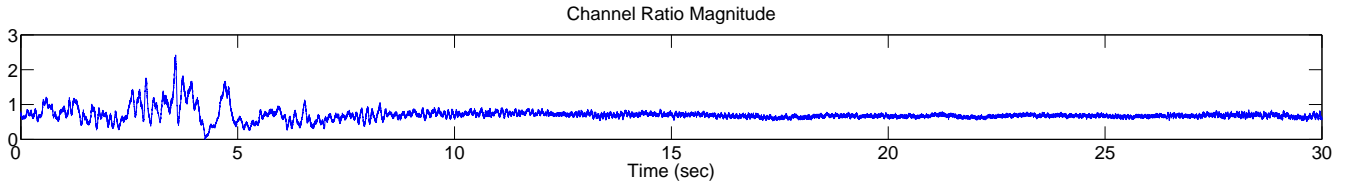


Fig. 5. A typical trace of the channel ratio magnitude.

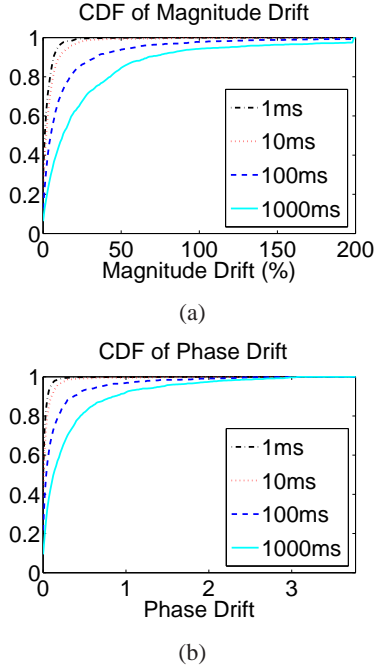


Fig. 4. The CDF of channel ratio drift. (a). Magnitude. (b). Phase (in radians).

data rate is not as high as hardware radios. When the data rate is higher, the Signal to Noise Ratio (SNR) requirements are higher, which may reduce the number of compatible pairs. Without hardware OSMR transmitters and receivers, we will use the channel state traces collected from our experiments and simulations to study the high data rate regime in Section V. Note that the channel state traces are collected by simply recording the received samples and are not subject to the same limitations in the OSMR transmission experiments.

### III. EXTENSIONS TO 802.11 MAC

The 802.11 MAC protocol needs to be extended to support OSMR transmissions. We focus on the case when only the AP acts as the OSMR sender.

#### A. The OSMR Transmission Procedure

The AP competes for medium access according to the current 802.11 MAC, i.e., monitoring the medium and backoff for a random time if needed. Thus, the fairness in the 802.11 MAC is preserved. Once the AP gains access to the medium, it may start an OSMR transmission. It may first broadcast a Channel Estimation Request (CRQ) frame, which includes a list of the nodes whose channel states are needed by the AP. The nodes send back the channel estimation reports in the Channel Estimation Report (CRP) frame one by one. The AP

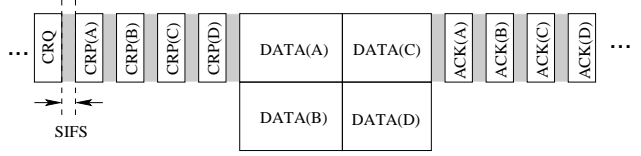


Fig. 6. Packet transmission with OSMR. CRQ: Channel Estimation Request. CRP: Channel Estimation Report.

then starts the data transmissions. After the data transmissions, the nodes send ACKs one by one. For example, an OSMR transmission procedure is illustrated in Fig. 6, in which the AP uses OSMR to transmit first to nodes *A* and *B* and then to nodes *C* and *D*.

Note that in the CRQ frame, the AP may announce a time offset value for each node to avoid the collision of the CRP frames. Similarly, it may announce another time offset value for each node to avoid the collision of the ACKs frames. The control frames may be separated by SIFS, as shown in Fig. 6. The data transmissions need not be separated by SIFS because the AP is the only transmitter. If the AP did not get the CRP frames from some nodes, it may adjust the OSMR transmission. For example, it may use OSMR only to nodes whose CRP frames have been received. After the ACKs frames, the AP removes the data that has been acknowledged; other data will be scheduled for retransmission. The CRQ frame should be sent at a data rate such that all involved nodes can decode the frame with high probability. The CRP frame can be sent at the same data rate as the ACK frame. The frame sent by OSMR is in the same format as the frame sent by a one-to-one transmission, i.e., must be preceded by the preamble, the PLCP header, and the MAC header, and must be trailed by the checksum.

We note that CRQ and CRP are generic names used to refer to the control frames needed for OSMR. In practice, they may be implemented by extending existing control frames of 802.11. Details such as frame format are out of the scope of this paper.

#### B. Fragmentation and TXOP

To improve the efficiency, two existing mechanisms in 802.11 can be exploited, namely fragmentation and Transmission Opportunity (TXOP). First, we note that the nodes may be at different data rates and the packet sizes may be different, such that it is unlikely that the AP can find packets to two nodes as a pair that occupy exactly the same amount of time. In such cases, the AP can simply send fragments of packets to match the transmission time. Second, we note that the channel estimation procedure incurs overhead. To amortize the cost, the AP may transmit for an extended period of time instead of transmitting just a few packets. Therefore, the AP



may request a TXOP time, which is first defined in 802.11e [21]. We denote the length of a TXOP as  $\gamma$  which may be several ms.

### C. Transmission Schedule and Sub-schedules

Within one TXOP, the AP may send to multiple nodes with multiple OSMR transmissions, as well as sending to some nodes without OSMR. The transmission schedule in a TXOP can be represented as a list of four-tuples. A four-tuple can be  $[(i, j), (x_{i,j}, x_{j,i})]$ , which means that the AP should send to nodes  $i$  and  $j$  simultaneously using OSMR for  $x_{i,j}$  and  $x_{j,i}$  bytes, respectively. The schedule could also have four-tuples such as  $[(i, -), (x_i, -)]$ , which means that the AP should send to node  $i$  without OSMR for  $x_i$  bytes. Each four-tuple in the list is called a *sub-schedule*. The number of bytes sent to two nodes belonging to the same sub-schedule should be proportional to their data rates and may be different.

### D. Opportunistically Piggybacking the Channel States

The AP needs to keep track of the compatibility relations of nodes in the network. We note that one possibility is to allow a node to piggyback its channel state in the ACK or the data frames. As such, the AP may receive the channel estimation reports in a timely manner for the heavily loaded nodes because they often transmit. The AP may not be able to get the channel estimation reports from the lightly loaded nodes; however, it is not as critical to optimize the transmissions to such nodes.

### E. Backward Compatibility

We note that the OSMR transmission process is completely backward compatible. This is because the AP can announce the duration of the TXOP and all nodes should backoff until the OSMR transmission finishes. In addition, all packet transmissions within the TXOP are separated by at most SIFS, such that no other nodes will attempt to transmit because they have to wait until DIFS. The AP may use OSMR transmissions only to the OSMR-capable nodes. When sending to other nodes, the AP may simply use the one-to-one transmission. Also, the uplink is unchanged because only the AP uses OSMR. Therefore, the OSMR-capable nodes and the OSMR-incapable nodes can coexist in the same LAN without interfering with each other, although the OSMR-capable nodes will receive better services from the AP.

## IV. DOWNLINK PACKET SCHEDULING

In this section, we focus on packet scheduling when OSMR is adopted. Packet scheduling is needed because the AP must make smart decisions to “pair up” packets to improve the overall downlink performance, such as the throughput. We assume the scheduler is given the number of bytes that must be sent to the nodes. The upper layer determines this based on the considerations on many issues, such as fairness and Quality of Service (QoS) requirements. For instance, when a node is running Voice Over IP (VoIP) applications, it must receive a certain number of bytes in a timely manner. The scheduler finds a schedule that meets the requirements of the upper layer while sends as many bytes as possible.

TABLE 1  
LIST OF NOTATIONS

$\gamma$	Length of TXOP
$N$	Number of nodes
$B_i$	Number of bytes buffered for node $i$
$m_i$	Number of urgent bytes to node $i$
$\mu_i$	Base data rate of node $i$
$\mu_{i,j}$	OSMR data rate of node $i$ with node $j$

### A. Definitions and Notations

We use  $\mu_i$  to denote the data rate of node  $i$  without using OSMR, and call it the *base rate*. We use  $\mu_{i,j}$  to denote the data rate of node  $i$  if the AP sends to nodes  $i$  and  $j$  simultaneously using OSMR, and call it the *OSMR rate* of node  $i$  with node  $j$ . Note that  $0 \leq \mu_{i,j} \leq \mu_i$ , because when not using OSMR, the AP focuses all power to node  $i$ . Also note that  $\mu_{i,j}$  may be different from  $\mu_{j,i}$ , because node  $i$  and node  $j$  may have different channels. We assume that the data rate is known to the AP and is stable for the scheduled transmission, because the AP can derive the data rates based on the channel states and the channel states do not change very fast. Indeed, the data rates of mobiles in the 3G networks are selected based on channel state feedbacks and vary every several ms [20]. We use  $B_i$  to denote the number of bytes in the buffer for node  $i$ , and use  $m_i$  to denote the number of bytes that must be sent to node  $i$  in this TXOP mandated by the upper layer. For convenience, we also refer to the bytes that must be sent the “urgent bytes”, and other bytes the “non-urgent bytes.” The number of nodes is denoted as  $N$ . Table 1 lists the notations.

### B. The Ideal Scheduler

We begin by considering an ideal case in which only the data bytes are sent and the overhead such as MAC header can be neglected. We use  $x_i$  to denote the number of bytes sent to node  $i$  without OSMR, and  $x_{i,j}$  to denote the number of bytes sent to node  $i$  using OSMR with node  $j$ . The optimal schedule is the solution to the following Linear Programming problem:

$$\max \sum_{i=1}^N x_i + \sum_{i=1}^N \sum_{j=1, j \neq i}^N x_{i,j} \quad (1)$$

subject to

$$x_i + \sum_{j=1, j \neq i}^N x_{i,j} \leq B_i, \text{ for all } i \quad (2)$$

$$x_i + \sum_{j=1, j \neq i}^N x_{i,j} \geq m_i, \text{ for all } i \quad (3)$$

$$\frac{x_{i,j}}{\mu_{i,j}} - \frac{x_{j,i}}{\mu_{j,i}} = 0, \text{ for all } i \neq j, \mu_{i,j}, \mu_{j,i} > 0 \quad (4)$$

$$\sum_{i=1}^N \frac{x_i}{\mu_i} + \frac{1}{2} \sum_{i=1}^N \sum_{j=1, j \neq i}^N \frac{x_{i,j}}{\mu_{i,j}} \leq \gamma \quad (5)$$

To see this, note that basically, term 1 is the total number of bytes that are sent and should be maximized. Constraint 2 states that the number of bytes sent to node  $i$  cannot be more than the total number of bytes stored in the buffer for node  $i$ . Constraint 3 states that the number of bytes sent to

node  $i$  cannot be less than the number of urgent bytes to node  $i$ . Constraint 4 states that if the AP sends to nodes  $i$  and  $j$  simultaneously, the time spent in sending to  $i$  must be the same as the time spent in sending to  $j$ . Constraint 5 states that the total amount of time must be no more than a TXOP length.

Note that if the LP is not feasible, the set of urgent bytes is not feasible, and the scheduler may send a feedback to the upper layer to recalculate the urgent bytes. In practice, to reduce the scheduling time, the upper layer may always issue urgent bytes that are guaranteed to be feasible. For example, it may make sure that the total time to send the urgent bytes at their base rates is no more than  $\gamma$ .

### C. Scheduling Considering the Overhead

The LP formulation provides theoretical insights and serves as an upper bound of the performance. However, it considers the ideal case, in which there is no overhead for a sub-schedule, where a sub-schedule is defined in Section III-C. In practice, the overhead of a sub-schedule includes the preamble, the PLCP header, the MAC header, etc., and can be more than  $20 \mu\text{s}$  in 802.11 a/g, which is non-trivial and cannot be neglected. For example, if a schedule consists of many short sub-schedules, the percentage of the overhead may be very large. Therefore, the scheduling problem must be revisited.

We assume the overhead of a sub-schedule is  $\beta$ . Given the constraints of the data rates, the number of urgent and non-urgent bytes, and the maximum transmission time  $\gamma$ , we define the optimal schedule as the schedule in which either 1) all buffered bytes are sent in a minimum time or 2) all urgent bytes are sent and as many non-urgent bytes are sent in  $\gamma$ . The OSMR Transmission With Overhead (OTWO) problem is defined as the problem to find an optimal schedule under these constraints. We prove that

*Theorem 1:* The OTWO problem is NP-hard.

*Proof:* In Appendix. ■

### D. A Practical Scheduler

In this section, we focus on the design of a practical scheduler. We note that the challenge is two fold. First, given the channel conditions, the scheduler should find a reasonably good schedule to maximize network performance. As the scheduling is in real time, the algorithm should have a low complexity and run fast. Second, the scheduler also has to handle possible channel fluctuations, as the channel conditions could change after the last channel estimation reports are sent. Given the complexity of the problem and the tight timing constraint, we will first discuss a two-phase algorithm for finding a schedule with given channel conditions. We will then describe how the scheduler handles channel fluctuations.

1) *A Two-Phase Algorithm:* We propose a two-phase algorithm to find a schedule when the channel conditions are given. In Phase 1, it considers only the urgent bytes, and exploits OSMR to find an efficient schedule using minimum time. After Phase 1, a partial schedule will be ready which likely requires less time than  $\gamma$ , and the non-occupied time is referred to as the *available time*. In Phase 2, the algorithm considers all

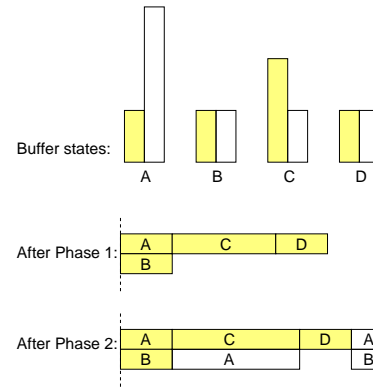


Fig. 7. An example of the scheduling algorithm. The solid rectangle represents urgent bytes and the blank rectangle represents non-urgent bytes.

buffered bytes, and makes use of the available time and sends as many bytes as possible. The algorithm schedules no more than  $2N$  sub-schedules where  $N$  is the number of nodes in the network, hence the overhead is bounded.

The motivation behind this two-phase approach is that the scheduling algorithm has two tasks: sending all urgent bytes *and* sending as many non-urgent bytes as possible. Jointly considering the two tasks may result in better schedules but will lead to a higher complexity. The two-phase algorithm, on the other hand, divides the two tasks into two phases, where Phase 1 guarantees that all urgent bytes are sent and Phase 2 packs as many non-urgent bytes as possible. This significantly narrows down the search space and reduces the complexity, making the algorithm suitable for running in real time.

An example is shown in Fig. 7, where there are 4 nodes, A, B, C, D, in the network. The numbers of urgent and non-urgent bytes for A, B, C, and D are [1000, 3000], [1000, 1000], [2000, 1000], and [1000, 1000], respectively. The base rates and the non-zero OSMR rates are:  $\mu_A = \mu_B = \mu_C = \mu_D = \mu_{A,B} = \mu_{B,A} = 1$ ,  $\mu_{A,C} = \frac{1}{2}$ , and  $\mu_{C,A} = \frac{5}{6}$ , measured in bytes per time unit.  $\gamma$  is 5000 time units. The number of urgent bytes is feasible because they can be transmitted at the base rates within 5000 time units. In Phase 1, the algorithm considers only the urgent bytes. It matches A with B and produces an available time of 1000 time units. In Phase 2, the algorithm first matches 1200 non-urgent bytes of A with the 2000 urgent bytes of C. Note that this is a change to the schedule obtained after Phase 1: the urgent bytes of C were scheduled with a one-to-one transmission and are now scheduled with an OSMR transmission. Because  $\mu_{C,A} = \frac{5}{6}$ , the new OSMR transmission requires a total of 2400 time units which is 400 units more than sending the same bytes to C using one-to-one transmission; in other words, it consumes 400 units of available time. The remaining 600 available time is assigned to 600 non-urgent bytes of A and 600 non-urgent bytes of B.

We now discuss the details of the algorithm. Basically, in every iteration of Phase 1, the algorithm searches the OSMR transmissions and adds a sub-schedule with the highest *time efficiency*; this is repeated until no further OSMR transmissions can reduce the transmission time, after which all remaining urgent bytes are sent with one-to-one transmissions. In every iteration of Phase 2, the algorithm finds a transmission with the highest *byte efficiency* and adds a sub-schedule according

to this transmission, and repeats this until no available time is left, or until all buffered bytes are scheduled. The definitions of the time efficiency and the byte efficiency are given below:

- *Time efficiency*: With regarding to a possible sub-schedule employing OSMR, the time efficiency is defined as the amount of time that can be saved comparing to sending the same number of bytes without OSMR. To be more specific, suppose a possible sub-schedule is  $[(i, j), (b, \frac{b\mu_{j,i}}{\mu_{i,j}})]$ . Note that if  $b$  bytes are sent to node  $i$  with OSMR,  $\frac{b\mu_{j,i}}{\mu_{i,j}}$  bytes must be sent to node  $j$  to match the transmission time. The time efficiency of this sub-schedule is denoted as  $\delta_{i,j,b}$  and is calculated according to

$$\delta_{i,j,b} = \frac{b}{\mu_i} + \frac{b\mu_{j,i}}{\mu_j\mu_{i,j}} - \frac{b}{\mu_{i,j}}.$$

To see this, note that the transmission time without OSMR is  $\frac{b}{\mu_i} + \frac{b\mu_{j,i}}{\mu_j\mu_{i,j}}$ , while the transmission time with OSMR is clearly  $\frac{b}{\mu_{i,j}}$ . Note that  $\delta_{i,j,b}$  increases with  $b$ . Therefore, if  $[(i, j), (b, \frac{b\mu_{j,i}}{\mu_{i,j}})]$  is the sub-schedule with the highest time efficiency, either  $b = m_i$  or  $\frac{b\mu_{j,i}}{\mu_{i,j}} = m_j$ , i.e., for at least one of the nodes, all bytes have been added to the sub-schedule.

- *Byte efficiency*: The byte efficiency is a measurement of the efficiency of a transmission in utilizing the available time to send non-urgent bytes. If a transmission has the highest byte efficiency, the algorithm will add a sub-schedule with as many bytes as possible according this transmission; note that this means that either no available time is left, or all bytes to at least one of the nodes have been added to the sub-schedule. We denote the current remaining available time as  $V$  and denote the amount of available time consumed by a sub-schedule  $t$  as  $\theta_t$ . If  $\theta_t > 0$ , the byte efficiency of the transmission is defined as the number of non-urgent bytes sent consuming one unit of available time and there are three types of such transmissions:
  - A non-OSMR transmission to a node  $i$  for some non-urgent bytes. Clearly, the byte efficiency is  $\mu_i$ .
  - An OSMR transmission to nodes  $i$  and  $j$  both for some non-urgent bytes. Clearly, the byte efficiency is  $\mu_{i,j} + \mu_{j,i}$ .
  - An OSMR transmission to node  $i$  for some urgent bytes and to node  $j$  for some non-urgent bytes. The byte efficiency is

$$\frac{\mu_i\mu_{j,i}}{\mu_i - \mu_{i,j}}.$$

Note that the urgent bytes to node  $i$  have been already been scheduled after Phase 1. This transmission piggybacks the non-urgent bytes to node  $j$  with the urgent bytes to node  $i$  by exploiting OSMR. To derive the byte efficiency, suppose when consuming one unit of available time,  $b$  bytes can be sent to node  $i$ . Therefore,  $\frac{b}{\mu_i} = 1 + \frac{b}{\mu_i}$ . Hence,  $b = \frac{\mu_i\mu_{j,i}}{\mu_i - \mu_{i,j}}$ , and the total time of this OSMR transmission is  $\frac{\mu_i}{\mu_i - \mu_{i,j}}$ . By definition,  $\frac{\mu_i\mu_{j,i}}{\mu_i - \mu_{i,j}}$  is the byte efficiency.

If  $\theta_t = 0$ , the transmission must piggyback the non-urgent bytes to a node  $j$  with the urgent bytes to a node  $i$  while  $\mu_i = \mu_{i,j}$ ; in this case, the byte efficiency is defined as  $\mu_{j,i}Z$  where  $Z$  is a large constant guaranteeing that the byte efficiency is greater than that of any transmission that consumes available time.

The algorithm is summarized in Algorithm 1. The input to the algorithm consists of  $\{m_i\}_i$ ,  $\{B_i\}_i$ ,  $\{\mu_i\}_i$ ,  $\{\mu_{i,j}\}_{i,j}$ , and  $\gamma$ , the output is the schedule  $S$ , which is a set of sub-schedules. Some sub-schedules added in Phase 2 may be merged with some sub-schedules added in Phase 1 if they involve the same pair of nodes. Note that in Phase 1, the two-phase algorithm schedules one sub-schedule and reduces the number of urgent bytes of at least one node to 0 in every iteration; hence, it will not schedule more than  $N$  sub-schedules in Phase 1. For similar reasons, in Phase 2, it will not schedule more than  $N$  sub-schedules. Therefore, no more than  $2N$  sub-schedules will be scheduled. In each iteration either in Phase 1 or Phase 2, the scheduler needs  $O(N^2)$  time. Therefore, the complexity is  $O(N^3)$ . Note that  $N$  is typically not very large in a wireless LAN.

---

**Algorithm 1** A Two-Phase Algorithm

---

```

1:  $S \leftarrow \emptyset$ .  $V \leftarrow 0$ .
2: — Phase 1 —
3: while 1 do
4:   Let  $[(i, j), (x_{i,j}, x_{j,i})]$  be the sub-schedule with the
   highest time efficiency.
5:   if  $\delta_{i,j,x_{i,j}} > 0$  then
6:      $S \leftarrow S \cup \{[(i, j), (x_{i,j}, x_{j,i})]\}$ .
7:      $V \leftarrow V + \delta_{i,j,x_{i,j}}$ .
8:   else
9:     Schedule one-to-one transmissions for the remaining
     urgent bytes.
10:  break;
11:  end if
12: end while
13: — Phase 2 —
14: while  $V > 0$  and not all bytes scheduled do
15:   Let  $t$  be the sub-schedule for the transmission with the
   highest byte efficiency.
16:    $S \leftarrow S \cup \{t\}$ .
17:    $V \leftarrow V - \theta_t$ .
18: end while

```

---

2) *Coping with Channel Fluctuation*: As the example shown in Fig. 5, the channel state of a node fluctuates with time, where the fluctuation may be faster at certain times than at other times. When the fluctuation is too fast, the processing matrix may become outdated during the transmission and the transmission will fail, because not all interferences can be canceled. To cope with this, the scheduler keeps track of the channel fluctuation speed of every node, and excludes a node from OSMR transmissions if its current fluctuation speed is above a threshold. The fluctuation speed can be estimated based on the channel state feedbacks from the node and the time when the feedbacks are received. Also, if the



channel state has not been updated for longer than a threshold, the channel state may be outdated, and the node should be excluded from OSMR transmissions. The AP can still use one-to-one transmissions to send data to such nodes. Before an OSMR transmission, the AP runs the scheduler based on its current channel state records. After getting the CRP frames, the AP may run the scheduler again if channel states of some nodes have changed significantly, i.e., above a threshold. However, because the AP schedules OSMR transmissions only to nodes with slow-varying channels, this should happen with low probability.

## V. EVALUATIONS

To evaluate the proposed protocol and algorithm, we develop an event driven simulator. We rely on the simulator for performance evaluation, because the current GNU SDR does not support very accurate timing required by the MAC protocol, and operates at a lower data rate than hardware radios. Our simulation is driven by traffic traces collected from wireless LANs [19] and the channel state traces collected from our experiments.

The simulator is configured to function as an 802.11g network. The AP is at the center and the nodes are randomly located within a certain maximum distance to the AP. Based on the path loss model in [13] and the specifications of Cisco Aironet 802.11a/b/g wireless cardbus adapter [18], the average received signal strength is assumed to be

$$P_r = -31 - 30 \log_{10} d$$

measured in dBm, where  $d$  is the distance between the sender and the receiver in meters. The channel state traces collected from our experiments with GNU SDR are amplified such that the average gain of each trace is normalized to 1. Note that the amplified traces preserve the channel fluctuation characteristics. In the simulation, the channel state of a node is obtained by randomly selecting a normalized channel state trace and multiplying it with the average signal strength between the node and the AP. The base rate is determined by the receiving power and the minimum receiving power threshold for each data rate specified in [18]. The OSMR rates are determined by the receiving power of the effective channels and the threshold in [18]; however, to account for channel fluctuation and channel estimation noise, an additional 7 dBm margin is applied when determining the OSMR rates. When not using OSMR, the SNR of a transmission is determined by the currently stronger antenna of the AP to achieve antenna diversity. In the simulation, whether or not a frame is received is determined by its data rate and the instantaneous SNR. When OSMR is used, the leaked signal intended to the other node is regarded as noise. Basically, if the instantaneous SNR is above the SNR threshold according to [18], the frame is received correctly; otherwise it is dropped.

We refer to the two-phase scheduler as OSMR-t. We implement the Linear Programming formulation with the LP solver available at [17] and refer to it as OSMR-lp. For comparison, we also run simulation disabling OSMR transmissions, and refer to it as No-OSMR. In No-OSMR, when the AP gains

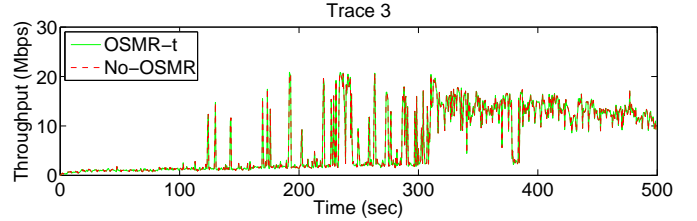


Fig. 8. Network downlink throughput in 500 seconds.

access to the medium, it transmits without OSMR for a TXOP length or until all buffered packets are sent, which is similar to the Frame Aggregation in 802.11n [16].

In the simulation, overhead such as the PLCP header and the MAC header are simulated. However, for OSMR-lp, the transmission time of a sub-schedule includes only the data transmission time because the LP formulation does not consider the overhead; this reveals the upper bound of the performance.  $\gamma$  is 3 ms. The OSMR-t and OSMR-lp scheduler consider a node not eligible for OSMR transmission if the phase drift of the channel ratio is more than  $\pi/100$  per ms or if the channel state is more than 10 ms old.

The set of urgent bytes is determined by the upper layer. In the simulation, if the total number of buffered bytes is small and can be sent at the base rates within  $\gamma$ , the set of urgent bytes is all the buffered bytes. Otherwise, the urgent bytes are determined according to the following constraints. First, they must satisfy  $\sum_{i=1}^N \frac{m_i}{\mu_i} = \gamma$ , which is basically to pessimistically assume that no OSMR transmissions can be scheduled. Second, to ensure fairness, the nodes are divided into two sets depending on their buffer states. Basically, if node  $i$  is in the first set, it has few buffered bytes, and  $m_i = B_i$ . If nodes  $j_1$  and  $j_2$  are in the second set, they have more buffered bytes, and  $\frac{m_{j1}}{\mu_{j1}} = \frac{m_{j2}}{\mu_{j2}} = t$ , where  $t \geq \frac{m_i}{\mu_i}$  for any node  $i$  in the first set.

We use four traces in [19], Trace 2 to Trace 5, which are collected by TCPDump seen at the wired port at the AP in a LAN with 75 nodes for about 10 minutes. The traces include traffic from realistic applications such as WWW and VoIP. To match the description of the trace collection in [19], we first set the maximum distance to the AP to be 20 m in our simulation, and run our simulation for 500 seconds. The results show that OSMR-t and No-OSMR have almost exactly the same throughput for all traces. For instance, the result for Trace 3 is shown in Fig. 8 where the two lines overlap. This is because the traffic load in the trace is not high. Note that the upper layer protocols, e.g., TCP, typically probe the capacity of the network to avoid overloading the network, hence the traffic load in the trace is unlikely to exceed the capacity of an AP, and therefore not high enough to reveal the benefit of OSMR.

To evaluate the performance of the network at higher traffic load, we process the trace files and combine Trace 2 to Trace 5 into one. As each trace contains 75 nodes, we create 10 merged nodes, and randomly combine the traffic of up to 30 actual nodes into one merged node. We use 30 seconds of the traffic trace from 400 seconds to 430 seconds. The maximum distance is set to be 60 m. Because the combined traffic can



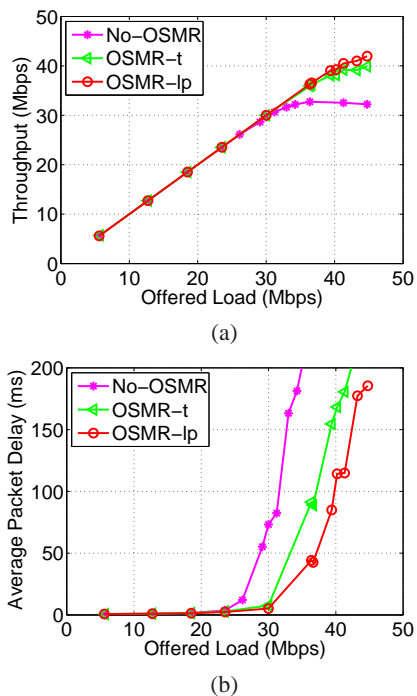


Fig. 9. Synthesized traffic combining 4 traces. The x axis is the offered load on the downlink. (a). Average throughput. (b). Average packet delay.

TABLE 2  
SUSTAINABLE THROUGHPUT (MBPS) AND OSMR-T IMPROVEMENT OVER No-OSMR

	No-OSMR	OSMR-t	OSMR-lp	Improvement
$N = 5$	38.9	48.2	51.9	24%
$N = 10$	30.7	36.5	38.8	19%
$N = 15$	29.4	37.4	42.2	27%
$N = 20$	27.2	35.8	38.1	32%

be very heavy, we allow the AP to drop a packet if the total number of buffered packets exceeds 1000. Fig. 9 shows the network performance as a function of the offered load on the downlink averaged over 20 random seeds, where Fig. 9(a) is the average downlink throughput and Fig. 9(b) is the average downlink packet delay. We can see that OSMR-t achieves a very close throughput as OSMR-lp and also performs significantly better than No-OSMR as the load increases.

To quantitatively measure the different schemes, we define the *sustainable throughput* as the maximum throughput when the average packet delay is less than 100 ms. The sustainable throughputs of different schemes are shown in Table 2 for networks of various sizes, where the results for networks with 5, 15, and 20 nodes are obtained in a similar manner as the network with 10 nodes. The improvement percentages of OSMR-t over No-OSMR are also shown. The sustainable throughput is determined by two related factors: 1) the number of nodes and 2) the number of compatible pairs. We can see that the sustainable throughput is the highest when  $N = 5$ , because least number of nodes are competing for the air time. The sustainable throughput of OSMR schemes for  $N = 15$  is higher than  $N = 10$ , because the gain by having more compatible pairs outweighs the loss of air time due to having more nodes. It is also interesting to notice that the improvement

percentage is the lowest when  $N = 10$ . This is because when  $N = 15$  and  $N = 20$ , there are more compatible pairs; when  $N = 5$ , the traffic to each individual node is heavier, such that compatible pairs are more likely to have buffered packets.

Note that the improvement is not as dramatic as one may have expected after being able to send multiple packets simultaneously. The major reason is that *compatible nodes may not both have buffered packets*. In a wireless LAN, it may happen that a node receives a large volume of traffic in a short period of time, while nodes compatible with this node receive little traffic, which causes the underutilization of the compatibility. Nevertheless, overall, we can see that OSMR is capable of improving the performance by around 20-30% for networks of various sizes.

## VI. DISCUSSIONS

We note that when the additional hardware cost and energy consumption can be tolerated, adding more antennas at the AP may increase the number of simultaneous receivers. Sending to more receivers may increase the aggregate link speed; the exact amount of gain depends on the channel conditions as well as the network traffic conditions. Although we mainly focus on the two receiver case, we note that our approaches can still be applied or can be extended to other cases. The MAC layer modifications can be trivially extended to support more simultaneous receivers. The NP-hard proof of the scheduling problem can be readily used to argue that the scheduling problem is still NP-hard when the maximum number of simultaneous receivers is greater than two, because it includes the two receiver case is a special case. The proposed scheduling algorithm can also be extended and the idea is still to schedule in two phases. Suppose up to  $r$  simultaneous receivers can be supported. In Phase 1, the algorithm may still choose the sub-schedule with the highest time efficiency. The definition of time efficiency can be extended as the time saved when the AP sends to up to  $r$  receivers simultaneously compared to sending to them individually. In Phase 2, the algorithm may still pick the transmission with the highest byte efficiency. The definition of the byte efficiency can be extended by considering transmissions which piggyback non-urgent bytes to  $r'$  receivers with urgent bytes to up to  $r - r'$  receivers for  $1 \leq r' \leq r$ . One potentially interesting issue is that the urgent bytes to different nodes may take different amount of time after piggybacking the non-urgent bytes because the data rates may change; while we leave this problem to our future work, we note that it is always possible to consider only transmissions that do not change the rates for the nodes who receive urgent bytes.

## VII. RELATED WORKS

Recently, applying advanced signal processing techniques to wireless networks has drawn much interest in the networking community [7], [9], [10], [11], [12]. We note that these works consider single-antenna systems, and cannot exploit the capacity of multiple antennas.

In [8], MU-MIMO on the uplink of wireless LANs was studied and implemented. The major issue of MU-MIMO on

the uplink is medium access, i.e., allowing multiple nodes to access the medium simultaneously without causing collisions, which is not needed in the downlink where the AP transmits to multiple nodes while the AP is aware of the buffer states of all nodes and can run intelligent algorithms.

In [5], the interference alignment and cancellation (IAC) technique was proposed which allows multiple senders to coordinate transmissions to multiple receivers. We note that IAC applies to wireless LANs with multiple APs connected by high speed wired connections, while OSMR applies to wireless LANs with only one AP, typical of residential networks. The MAC protocol proposed in [5] is an extension to the 802.11 PCF. It still has several issues unclear, e.g., how the protocol supports nodes with newly arrived VoIP traffic in the middle of the contention free period. In contrast, the MAC protocol we propose is an extension to the 802.11 DCF, which supports all types of traffic by random contention. The transmission selection algorithm proposed in [5] selects the best among a total of 4 random options. The scheduling algorithm we propose is more sophisticated while remaining acceptable in complexity.

In [6], a downlink MU-MIMO system was implemented on the WARP platform, with which indoor experiments were performed. We note that the main focus of [6] is the physical layer study; the gain of MU-MIMO is measured assuming nodes always have traffic. In contrast, we consider the realistic case for wireless LANs when both the channel conditions and the traffic conditions may vary, and propose practical scheduling algorithms. Our evaluation shows that the randomness of the traffic has significant impact on the system performance, because the AP cannot exploit the compatibility of two nodes when one of them has no buffered packets.

In our earlier works [22], [23], we studied scheduling problems with OSMR, where the focus was to send the buffered packets in minimum time. However, the scheduling problem was defined under simplified network models, in which 1) the scheduler need only maximize network throughput without considering fairness, 2) the packets cannot be fragmented such that capacity may be lost due to packet size mismatch, and 3) the channel is assumed to be stable. In this paper, we define the problem under more practical settings, in which 1) the upper layer may impose fairness requirements by issuing the urgent bytes, 2) packets fragmentation is allowed which helps achieving higher efficiency, and 3) considerations must be given to handle channel fluctuation. As such, the scheduler given in this paper is different and more advanced. In addition, a detailed physical layer study is provided in this paper, and the evaluation is driven by the channel state traces collected from experiments.

## VIII. CONCLUSIONS

In this paper, we give a systematic study on employing the One-Sender-Multiple-Receiver (OSMR) transmission technique in wireless LANs. In the physical layer, we implement a prototype OSMR transmitter/receiver with GNU Software Defined Radio that allows one sender to send to two receivers simultaneously. We conduct experiments which show that

wireless channels allow OSMR for a significant percentage of the time. We also study the characteristics of wireless channels, and show that wireless channel is stable for most of the time which is desirable for OSMR. Motivated by our physical layer study, in the MAC layer, we propose extensions to the 802.11 MAC. We study the packet scheduling problem and propose a practical scheduler capable of exploiting OSMR efficiently and handling channel fluctuations. We evaluate the proposed protocol and scheduling algorithm based on simulations driven by wireless LAN traffic traces and wireless channel traces collected from our experiments. The results show that OSMR is capable of improving wireless LAN performance significantly.

## REFERENCES

- [1] IEEE Computer Society LAN MAN Standards Committee, *IEEE Standard 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.
- [2] D. Tse and P. Viswanath, "Fundamentals of Wireless Communication," *Cambridge University Press*, May 2005.
- [3] D. Gesbert, M. Kountouris, R. W. Heath, Jr., C. B. Chae, and T. Salzer, "From single user to multiuser communications: Shifting the MIMO paradigm," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 36-46, Oct., 2007.
- [4] T. Yoo, N. Jindal, and A. Goldsmith, "Multi-antenna downlink channels with limited feedback and user selection," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 7, pp. 14781491, 2007.
- [5] S. Gollakota, S. D. Perli and D. Katabi, "Interference alignment and cancellation," *ACM SIGCOMM*, 2009.
- [6] E. Aryafar, N. Anand, Theodoros Salonidis, and Edward W. Knightly, "Design and experimental evaluation of multi-user beamforming in wireless LANs," *ACM MOBICOM 2010*.
- [7] L. E. Li, R. Alimi, R. Ramjee, H. Viswanathan, and Y. R. Yang, "muNet: Harnessing multiuser capacity in wireless networks," *IEEE INFOCOM Minisymposium*, April 2009.
- [8] K. Tan, H. Liu, J. Fang, M. Chen, J. Zhang, W. Wang, and G. M. Voelker "SAM: Enabling practical spatial multiple access in wireless LAN," *ACM MOBICOM*, Beijing, China, Sept. 2009.
- [9] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: Interference cancellation for wireless LANs," *ACM MOBICOM 2008*.
- [10] S. Gollakota and D. Katabi, "ZigZag decoding: Combating hidden terminals in wireless networks," *ACM SIGCOMM 2008*.
- [11] K. Jamieson and H. Balakrishnan, "PPR: Partial packet recovery for wireless networks," *ACM SIGCOMM 2007*.
- [12] S. Katti, S. Gollakota and D. Katabi, "Embracing wireless interference: Analog network coding," *ACM SIGCOMM 2007*.
- [13] A. Bose and C. H. Foh, "A practical path loss model for indoor WiFi positioning enhancement," *ICICS 2007*, Singapore, December 2007.
- [14] "Gnu radio - gnu fsf project," <http://www.gnu.org/software/gnuradio>.
- [15] Ettus. Inc, "Universal Software Radio Peripheral," <http://ettus.com>.
- [16] "802.11n: Next-Generation Wireless LAN Technology," [http://80211n.com/white\\_paper/802\\_11n-WP100-R.pdf](http://80211n.com/white_paper/802_11n-WP100-R.pdf).
- [17] <http://lpsolve.sourceforge.net/5.5/>
- [18] Cisco Aironet 802.11a/b/g wireless cardbus adapter, <http://www.cisco.com/>.
- [19] <http://www.winlab.rutgers.edu/~ergin/mobicom2007/>
- [20] <http://www.umtsworld.com/technology/hsdpa.htm>
- [21] <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>
- [22] Z. Zhang, M. Zhao and Y. Yang, "Enhancing downlink performance in wireless networks by simultaneous multiple packet transmission," *IEEE Transactions on Computers*, vol. 58, no. 5, pp. 706-718, May 2009.

- [23] Z. Zhang and S. Bronson, "A packet scheduling algorithm for optimizing downlink throughput in wireless LANs with the one-sender-multiple-receiver technique," *IEEE Globecom*, 2009.
- [24] Z. Zhang, S. Bronson, J. Xie and H. Wei, "The One-Sender-Multiple-Receiver technique and downlink packet scheduling in wireless LANs," *Technical Report TR-090217*, Florida State University, 2009.
- [25] Z. Zhang, S. Bronson, J. Xie and H. Wei, "Employing the one-sender-multiple-receiver technique in wireless LANs," *IEEE INFOCOM*, 2010.

## APPENDIX

### A. Implementation of OSMR

We discuss the two key components in our OSMR implementation, namely the channel estimation and the calculation of the processing matrix.

1) *Channel Estimation*: To use OSMR, the sender needs to know the channels to determine the processing matrix. Actually, to determine the processing matrix, for receiver  $i$  ( $i \in \{1, 2\}$ ), only the *channel ratio* defined as  $g_i = h_{i2}/h_{i1}$  is needed. As the same channel estimation process is carried out at both receivers, in the following, we consider one receiver and refer to it as receiver  $i$ . To estimate  $g_i$ , a channel estimation sequence is transmitted at the sender. That is, we let the sender transmit  $\{+1, -1, +1, -1, +1, \dots\}$  at antenna 1 and transmit  $\{-1, -1, -1, -1, \dots\}$  at antenna 2. At the receiver, let the received signal strengths from antenna 1 and antenna 2 of the sender be  $w$  and  $v$ , respectively. Note that  $|g_i| = v/w$ . Suppose the phase of  $g_i$  is  $\phi$ . If the receiver's phase is locked to the phase of antenna 1 of the sender, when antenna 1 is transmitting  $+1$ , the received complex symbol should be  $[w - v \cos(\phi)] + j[-v \sin(\phi)]$ ; when antenna 1 is transmitting  $-1$ , the received complex symbol should be  $[-w - v \cos(\phi)] + j[-v \sin(\phi)]$ . Therefore, if the receiver receives two consecutive samples denoted as  $S_1 = x_1 + jy_1$  and  $S_2 = x_2 + jy_2$ , where  $S_1$  and  $S_2$  correspond to the symbol when antenna 1 is transmitting  $+1$  and  $-1$ , respectively, we have

$$w = \frac{x_1 - x_2}{2}, \quad (6)$$

and

$$\phi = \tan^{-1}\left(\frac{y_1 + y_2}{x_1 + x_2}\right), \quad (7)$$

and

$$v = \frac{-y_1}{\sin(\phi)}. \quad (8)$$

Note that there are two values for  $\phi$  in  $[-\pi, \pi]$  that satisfy Eq. 7. The ambiguity is resolved by choosing the one resulting in  $v > 0$  in Eq. 8.

However, the receiver's phase will not be locked to the phase of antenna 1, because the receiver is receiving the addition of two signals with different phases. To cope with this, we let the sender transmit the same symbols  $\{+1, +1, -1, -1, +1, +1, \dots\}$  at both antennas as training symbols for the phase tracking circuit of the receiver. After the training symbols, a set of special symbols are sent to indicate the beginning of the channel estimation sequence. When the receiver receives the special symbols, it stops the phase tracking circuit. At this time, the receiver's phase is locked to the symbol when

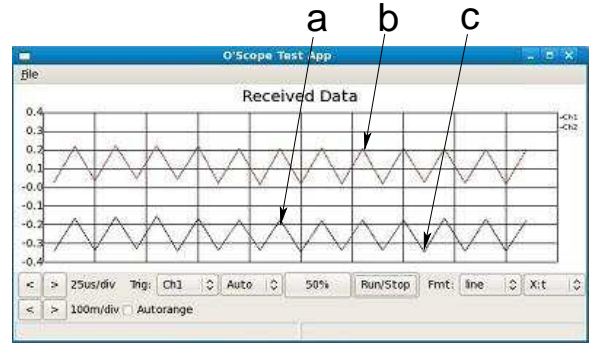


Fig. 10. A screenshot of captured channel estimation symbols.

both antenna 1 and antenna 2 are transmitting  $-1$  in the channel estimation sequence. Suppose the receiver gets two consecutive samples  $S'_1 = a + jb$  and  $S'_2 = c$ , where  $S'_1$  and  $S'_2$  correspond to the samples when antenna 1 is sending  $+1$  and  $-1$ , respectively. Note that  $S'_2$  does not have an imaginary component, because the receiver's phase is locked to the phase when both antennas at the sender are transmitting  $-1$ . We note that if the difference between the current phase of the receiver and the phase of antenna 1 of the sender is  $\theta$ ,  $S_1 = S'_1 e^{j\theta}$ ,  $S_2 = S'_2 e^{j\theta}$ . As the imaginary components of  $S_1$  and  $S_2$  are the same,

$$a \sin(\theta) + b \cos(\theta) = c \sin(\theta), \quad (9)$$

hence,

$$\theta = \tan^{-1}\left(\frac{b}{c-a}\right). \quad (10)$$

With  $\theta$ ,  $S_1$  and  $S_2$  can be found based on  $S'_1$  and  $S'_2$ , which then determine  $w$ ,  $v$ , and  $\phi$ . The ambiguity of  $\theta$  can be resolved by considering the sign of  $w$ .

For example, Fig. 10 shows a screenshot of captured channel estimation symbols, where  $a = -0.18$ ,  $b = 0.20$ , and  $c = -0.32$ . It can be found that  $\theta = -0.31\pi$ ,  $w = 0.12$ ,  $v = 0.27$ ,  $v/w = 2.25$ , and  $\phi = -0.43\pi$ .

2) *Determining the Processing Matrix*: The simplest choice of the processing matrix is the inversion of the channel matrix. In our current implementation, we take some extra measures in attempt to further optimize the performance as well as limiting the transmitting power. First, to force the interference to be 0, we require

$$\mathbf{h}_1 \mathbf{u}_2 = 0, \mathbf{h}_2 \mathbf{u}_1 = 0. \quad (11)$$

Second, we require

$$|\mathbf{h}_1 \mathbf{u}_1| \geq \eta |h_{11} + h_{12}|, |\mathbf{h}_2 \mathbf{u}_2| \geq \eta |h_{21} + h_{22}| \quad (12)$$

where  $\eta$  is a constant. This is to make sure that the effective channels are not too weak compared to the original unprocessed channels.<sup>1</sup> Third, we require

$$|u_{11} + u_{12}| \leq 1, |u_{21} + u_{22}| \leq 1, \quad (13)$$

to make sure that the transmitted signal power is within the limit of the transmitter. Note that if the data symbol to be sent

<sup>1</sup>We assume that if the sender has two antennas but does not use OSMR, it transmits the same signal at both antennas with equal power.



to user  $i$  is  $d_i$  for  $i \in \{1, 2\}$ , the signal sent by antenna  $i$  is  $u_{i1}d_1 + u_{i2}d_2$ . To make sure that each antenna is transmitting at no more than the regulated power,  $|u_{i1}d_1 + u_{i2}d_2|$  should be no more than  $|d_i|$  which is the transmitting magnitude of antenna  $i$  when OSMR is not used. The exact value of  $u_{i1}d_1 + u_{i2}d_2$  depends on  $d_1$  and  $d_2$  which are random. However, if this constraint is satisfied, the peak transmitting power is never more than the transmitting power when OSMR is not used.

From Eq. 11, we have  $u_{11} = -\frac{h_{22}}{h_{21}}u_{21}$  and  $u_{12} = -\frac{h_{12}}{h_{11}}u_{22}$ . Substituting  $u_{11} = -\frac{h_{22}}{h_{21}}u_{21}$  into the first half of Eq. 12, we have

$$|u_{21}||h_{11}| - \frac{h_{22}}{h_{21}} + \frac{h_{12}}{h_{11}} = |u_{21}||h_{11}| - g_2 + g_1 \geq \eta|h_{11} + h_{12}|,$$

therefore,

$$|u_{21}| \geq \eta \left| \frac{1 + g_1}{-g_2 + g_1} \right|. \quad (14)$$

Similarly,

$$|u_{22}| \geq \eta \left| \frac{1 + g_2}{-g_1 + g_2} \right|. \quad (15)$$

Eq. 14 and Eq. 15 give the minimum magnitude of  $u_{21}$  and  $u_{22}$ . As  $u_{11} = -g_2u_{21}$  and  $u_{12} = -g_1u_{22}$ , from Eq. 13, we have

$$|g_2u_{21} + g_1u_{22}| \leq 1, |u_{21} + u_{22}| \leq 1. \quad (16)$$

In our current implementation, to find  $u_{21}$  and  $u_{22}$ , we start with the minimum magnitude of  $u_{21}$  and  $u_{22}$  according to Eq. 14 and Eq. 15 where we set  $\eta = 0.1$ . Let the phase difference between  $u_{21}$  and  $u_{22}$  be  $\delta$ . We conduct a linear search over  $[-\pi, \pi]$  at a step of  $\frac{\pi}{360}$  for  $\delta$  to check if a  $\delta$  can be found such that both inequalities in Eq. 16 are satisfied. If no  $\delta$  can be found, the two receivers are not compatible. Otherwise, we increase the magnitude of  $u_{21}$  and  $u_{22}$  by 10% of their minimum values and conduct another search; this is continued until no  $\delta$  can be found and the  $\delta$  found in the last round is used as the solution.

### B. Proof of Theorem 1

*Proof:* We reduce the Maximum Independent Set (MIS) problem to the OTWO problem. In a graph, a set of vertices are *independent* if no two vertices in the set are adjacent to each other. A maximum independent set is an independent set with the maximum cardinality. We note that we consider MIS instances with no isolated vertices, which will not reduce the complexity of the MIS problem because the isolated vertices must belong to any maximum independent set.

Given any instance of the MIS problem, we construct an instance of OTWO problem as follows. Denote the graph in the MIS instance as  $G$  and suppose it has  $N$  vertices and  $E$  edges. The construction consists of three steps:

- *Creation of nodes.* For any vertex, say,  $v_i$ , in  $G$ , create a “first level” node  $U_i$ . For any edge in  $G$ , say,  $e_{ij}$ , create a “second level” node  $U_{ij}$ .  $U_{ij}$  is referred to as the “child” of  $U_i$ . Note that  $U_{ij}$  is also a child of  $U_j$ ; in this sense,  $U_{ij}$  and  $U_{ji}$  refer to the same node in this construction.
- *Assignment of the buffer states.* For a first level node  $U_i$ , if the degree of  $v_i$  in  $G$  is  $d_i$ , it has  $d_i C$  urgent bytes, where  $C$  is a constant. Each second level node has  $C$

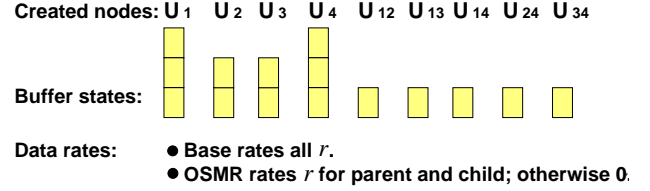
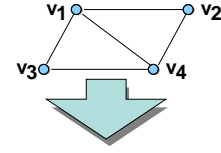


Fig. 11. The construction of the OTWO instance.

urgent bytes. No node has non-urgent bytes. Note that in this case the optimal schedule is the schedule that uses minimum time to send the urgent bytes.

- *Assignment of the rates.* Let the base rates of all nodes be  $r$ . For any two nodes, if one is not the child of the other, the OSMR rates between them are 0; otherwise, the OSMR rates are both  $r$ .

Fig. 11 shows the construction of a simple instance.

*Claim 1:* We say a second level node is saturated if all its data is sent with OSMR transmissions. In an optimal schedule, all second level nodes are saturated.

*Explanation:* To see this, we use contradiction. Suppose in an optimal schedule, some bytes of a second level node  $U_{ij}$  are sent with a one-to-one transmission. We note that if this is true, some bytes to  $U_i$  must also be sent with a one-to-one transmission. This is because  $U_i$  has  $d_i C$  bytes and only  $d_i$  children, while each child has only  $C$  bytes which can be paired up with at most  $C$  bytes of  $U_i$ . Therefore, we may improve the schedule by replacing the one-to-one transmissions to  $U_{ij}$  and  $U_i$  with an OSMR transmission, thus saving time. This, however, contradicts the optimality of the schedule.

*Claim 2:* Given any optimal schedule, we can rearrange it such that all bytes to any second level node are sent along with bytes to only one of its parents.

*Explanation:* Before proving this claim, we note that if it is true, we need only consider such optimal schedules. To see the claim is true, suppose in an optimal schedule, there exists a second level node  $U_{ij}$  which has  $x$  bytes and  $C - x$  bytes sent along with  $U_i$  and  $U_j$ , respectively. Note that according to Claim 1, no one-to-one transmission is scheduled for  $U_{ij}$ . Denote the two sub-schedules as  $[(U_i, U_{ij}), (x, x)]$  and  $[(U_j, U_{ij}), (C - x, C - x)]$ , respectively. Note that in this case, there are at least  $C - x$  bytes of  $U_i$  sent with a one-to-one transmission, denoted as sub-schedule  $[(U_i, -), (W, -)]$  where  $W \geq C - x$ . We can easily rearrange the three sub-schedules into  $[(U_i, U_{ij}), (C, C)]$ ,  $[(U_j, -), (C - x, -)]$  and  $[(U_i, -), (W - C + x, -)]$  which does not increase the total transmission time and overhead. This can be repeated for all second level nodes.

*Claim 3:* We say a first level node is saturated if all its bytes are sent along with bytes to its children with OSMR. If the number of saturated first level nodes is  $I$  in an optimal



schedule, the total amount time used by this schedule is

$$\begin{aligned} & E\left(\beta + \frac{C}{r}\right) + \left(N\beta + \frac{2EC}{r}\right) - \frac{EC}{r} - I\beta \\ &= \frac{2EC}{r} + (E + N - I)\beta. \end{aligned} \quad (17)$$

Explanation: We note that there are  $E$  second level nodes. According to Claim 2, one sub-schedule must be scheduled for each of them which takes a total of  $E\left(\beta + \frac{C}{r}\right)$  time. For the first level nodes, if no OSMR is used, they have to consume  $N\beta + \frac{2EC}{r}$  time. However, a total of  $EC$  bytes of the first level nodes are sent along with their children with OSMR, so the transmission time must be deducted by  $\frac{EC}{r}$ . In addition, if a first level node is saturated, no one-to-one sub-schedule is needed for this node and the overhead can be saved. Therefore, the transmission time is further deducted by  $I\beta$ .

*Claim 4: Let the set of saturated first level nodes in a schedule be  $A$  and the set of vertices in  $G$  corresponding to  $A$  as  $V_A$ .  $V_A$  must be an independent set.*

Explanation: Note that if two first level nodes correspond to two adjacent vertices in  $G$ , they cannot both be saturated. This is because they share a same child and the child can be paired up with only one parent.

*Claim 5: Let  $V_A$  be an independent set in  $G$  with size  $|V_A|$ . There exists a schedule for the OTWO instance with no less than  $|V_A|$  saturated first level nodes.*

Explanation: We explicitly construct the schedule as follows. First, create a sub-schedule for each node using only one-to-one transmission with which all data of this node is sent. Second, for each first level node corresponding to a vertex in  $V_A$ , piggyback all its data with the data of its children using OSMR, then remove the one-to-one transmission scheduled for this node earlier. We note that this can always be done because  $V_A$  is an independent set, and therefore no two first level nodes corresponding to vertices in  $V_A$  share a common child. Clearly, this will result in  $|V_A|$  saturated first level nodes.

Finally, we may establish the claim that given an optimal schedule  $S^*$  for the OTWO instance, the set of saturated first level nodes determines a maximum independent set in  $G$ . We denote the set of saturated first level nodes in  $S^*$  as  $A$  and the set of vertices in  $G$  corresponding to  $A$  as  $V_A$ . Based on Claim 4,  $V_A$  is an independent set. If  $V_A$  is not a maximum independent set, there exists a maximum independent set in  $G$ , denoted as  $V_B$ , with size larger than  $V_A$ . Based on Claim 5, we can make a schedule  $S'$  for the OTWO instance according to  $V_B$  with more saturated first level nodes than  $A$ . Denote the set of saturated first level vertices in  $S'$  as  $B$ . It is possible that  $S'$  may still have unsaturated second level nodes. We may scan the first level nodes not in  $B$  following any arbitrary order; for each such first level node, we may saturate its unsaturated children one by one by piggybacking its data with the data of its child. As the data of a first level node is enough to saturate all its children, all second level nodes are saturated in the end. With the same arguments as in Claim 3, it is clear that the time used by  $S'$  is also given by Eq. 17; however, as  $S'$  has more saturated first level nodes than  $S^*$ ,  $S'$  is a better schedule than  $S^*$ , hence a contradiction. ■