Extended Abstract: Superposition Coding for Wireless Mesh Networks

Li (Erran) Li⋆ Richard Alimi† Ramachandran Ramjee§
Jingpu Shi‡ Yanjun Sun‡ Harish Viswanathan⋆ Yang Richard Yang†

Bell Labs, Alcatel-Lucent⋆ Rice University‡ Microsoft Research, India§ Yale University†

ABSTRACT

A major barrier for the adoption of wireless mesh networks is severe limits on throughput. In this paper, we apply superposition coding to substantially improve network capacity of large, dense wireless mesh networks. Superposition coding is a physical layer technique that allows a transmitter to simultaneously send independent packets to multiple receivers. While superposition coding has been studied extensively by the physical layer community, we present the first design of practical and effective MAC protocols to take advantage of superposition coding in wireless mesh networks. Extensive evaluations show that superposition coding can be a practical method to increase the throughput of large, dense wireless mesh networks. Specifically, in a mesh network with 2 to 64 active receivers and one gateway, we show that our system can increase throughput up to 154%, with average gain ranging from 10% to 21%. When there are multiple gateways forming a mesh network, our system gains up to 98%, with average gain ranging from 24% to 46%. These results clearly demonstrate the potential benefits of our system. We also present results from an implementation of superposition coding using GNU Radio.

Categories and Subject Descriptors: C.2.1 [Computer Communication Networks]: Network Architecture and Design – *Wireless communications*; C.2.5 [Computer Communication Networks]: Local and Wide-Area Networks – *Access schemes*

General Terms: Algorithms, Performance, Design.

Keywords: Superposition Coding, Scheduling, Wireless Mesh

1. INTRODUCTION

Wireless mesh networks are becoming a major paradigm for constructing user access networks that provide community or city-wide Internet connectivity. However, recent theoretical analysis (*e.g.,* [3]) and experimental measurements (*e.g.,* [2]) have shown that the current wireless mesh networks are severely limited in throughput and do not scale as they become large and dense.

In this paper, we conduct the first study and design of wireless mesh networks that use superposition coding to substantially improve network capacity. Unlike previous studies, which focus mainly on physical layer issues of applying superposition coding, our study proposes simple, practical, and effective MAC protocols to take advantage of superposition coding.

2. BACKGROUND

We first provide a brief introduction to the superposition coding technique in the physical layer. While the focus of this paper is on the MAC layer, we need a basic understanding of the physical layer to design an effective MAC layer. For more details, please refer to [6].

Specifically, superposition coding is a physical layer technique by which a transmitter can *simultaneously* send independent messages to multiple receivers. In this paper, for simplicity, we restrict ourselves to the two-receiver case, in which the transmitter *superimposes* an additional message destined to a secondary receiver (receiver 2) on a basic message destined for a primary receiver (receiver 1). We also refer to the basic message as the first, primary or lower layer, and the additional message as the second, or upper layer. It is natural to extend our technique to more than two layers.

In implementation, a transmitter using superposition coding splits the available transmission power between the two layers, selects the transmission rate for each of the layers, then encodes and modulates each of the packets separately at the selected rate. The modulated symbols are scaled appropriately to match the chosen power split and summed to obtain the transmitted signal. For example, if quadrature phase shift keying (QPSK) is used for modulation of both layers, then the superposed transmitted signal will have a 16-point constellation.

The two receivers decode their received signal using different schemes. Receiver 1 decodes its packet treating the superimposed additional layer as interference. Receiver 2 first decodes the basic layer, re-encodes it, and then subtracts it from the original signal. It then decodes the remaining signal. This process is referred to as *successive interference cancellation* (SIC). To allow receiver 2 to decode the basic layer whenever receiver 1 can and to ensure that the remaining signal after subtraction has enough strength over noise, the channel quality to receiver 2 should be better than that to receiver 1; that is, the two channels should be asymmetric. Thus, we also refer to receiver 1 as the weaker receiver and receiver 2 the stronger.

To better appreciate the potential benefits of superposition coding, consider applying superposition coding to 802.11 like wireless mesh networks. Recent measurement studies (*e.g.,* [4]) have shown that nodes in a wireless network may be distributed unevenly and the channel quality of the links in such mesh networks varies widely. Such rate diversity can drastically reduce network throughput using standard 802.11. However, this channel diversity presents a setting where it can be particularly effective for superposition coding. Consider a simple scenario of one transmitter and two receivers. The channel quality to receiver 1 is low and can support only 6 Mbps; the channel quality to receiver 2 is high and can support 54 Mbps. Without superposition coding, the transmitter using FIFO scheduling may alternate transmissions to the two receivers. Thus, the transmitter spends $54/(54+6) = 90\%$ of the time transmitting to receiver 1 and 10% to receiver 2. The total throughput is only $10.8(= 6*0.9+54*0.1)$ Mbps. One way to improve total throughput and still maintain fairness is to use a different scheduler

MobiCom'07, September 9–14, 2007, Montréal, Québec, Canada.
ACM 978-1-59593-681-3/07/0009.

such as proportional fairness. However, this reduces the throughput of the receiver with poor quality by 44%. With superposition coding, the transmitter can superimpose the messages to receiver 2 as additional messages on the basic messages to receiver 1. Thus, the transmitter maintains a constant throughput of $60(= 54+6)$ Mbps. This throughput is $5.55\times$ that of the scenario without superposition coding.

3. MAC FOR SUPERPOSITION CODING

In this section, we present our MAC protocols utilizing superposition coding.

3.1 Medium Access Control

To maximize reuse of previous design and increase backwards compatibility, we base the message flow of our MAC on 802.11 RTS/CTS/DATA/ACK. We make small extensions to address two issues: (1) enable superposition coded messages to be sent to two receivers; and (2) enable feedback of estimated SINRs to the transmitter.

Specifically, we extend RTS by adding an extra address. The first address denotes the receiver of the basic packet, while the second address denotes the receiver of additional packet. Since an RTS is addressed to two receivers, it triggers the transmission of one CTS packet from each. These two CTSs are separated by SIFS to avoid collision. Each CTS message contains the estimated SINR calculated using the pilot symbols in the preceding RTS message. The reported SINRs will be stored in the SINR table. Each node maintains such a table which maps each link to the corresponding estimated SINR. Superposition coding will be done based on the stored SINR values. Each superposition coded DATA packet will require two ACK packets, one from each receiver. This is handled similarly to the two CTS packets. For channel estimation if RTS/CTS is disabled, please refer to our technical report [5].

3.2 MAC Scheduling: A Greedy Scheduler

Besides specifying message flow, the MAC layer will also need a scheduling algorithm to select the data packets to transmit. We design a scheduling algorithm that takes advantage of superposition coding. Our scheduler extends a given basic scheduler and treats it as a blackbox.

Consider a node u and assume that the links to its n_u neighbors are numbered $1,\ldots,n_u$. Let $v_1,\ldots,v_{n_u}$ be the corresponding neighbors. Let the estimated channel gain to link i be h_i. The routing algorithm determines the next hop and thus the link to be used for each data packet. Let Q_i be the queue of packets waiting for transmission on link i. We use $|Q_i|$ to denote the length of Q_i.

For simplicity of presentation, we assume that the total power of each transmission is fixed to be P, all packets are of the same size s, and the background noise is fixed at N_0 at all nodes.

For each transmission, we assume that the transmitter picks one rate for each layer. Let the set of discrete transmission rates be $\mathcal{R}$. We use the rate functions $R^{(1)}(i,p)$ and $R^{(2)}(i,p)$ to determine the transmission rate for link i at transmission power level p, when the packet to link i is encoded in the first layer and second layer, respectively. For $R^{(1)}(i,p)$, since the packet in the second layer with transmission power $P-p$ will become interference to the first layer packet, the SINR used is $\frac{p|h_i|^2}{(P-p)|h_i|^2+N_0}$. On the other hand, due to successive interference cancellation when a receiver recovers a second layer packet, the SINR to determine $R^{(2)}(i,p)$ is just $\frac{p|h_i|^2}{N_0}$. With SINR, we search a table generated from [1] showing the relationship between packet error rate (PER) and SINR at different 802.11 rates. We pick a rate so that the PER is small, say 10%.

We impose two objectives on our scheduling algorithm. First, since a network may already have a MAC scheduler (*e.g.,* FIFO, round robin, or proportional fairness) to determine which data packet to transmit next, our scheduling algorithm shall be generic and easily integrated with the existing scheduler to take advantage of superposition coding. We refer to the existing scheduler as the basic scheduler and our scheduler the SC scheduler. Second, superposition shall only increase throughput.

We design a greedy scheduler, referred to as *Gopp*, to achieve the preceding objectives. Figure 1 shows the complete scheduler.

We keep track of the arrival time of each packet. We use logical time. If we have a new packet arrive at the queue, the time is increased by one unit. Let pkt_1 be the packet to be transmitted next as determined by the basic scheduler. We assume that its arrival time is τ_1. Without loss of generality, we assume the next hop of pkt_1 is v_1. For ease of presentation, let pkt_1 be transmitted in the first layer. It is also possible that pkt_1 is transmitted in the second layer. To handle this, we run the algorithm the second time assuming pkt_1 is in the second layer and take the better of the two solutions.

We first consider each link $i \neq 1$ to determine the maximum total number of packets that can be transmitted if we use the packets for link i as the second layer. Specifically, the total number of packets N_i that can be transmitted using superposition coding during the time to transmit a packet at the first layer at a power level p is:

$$N_i = 1 + min\left(\frac{R^{(2)}(i,P-p)}{R^{(1)}(1,p)}, |Q_i|\right), \qquad (1)$$

where $\frac{R^{(2)}(i,P-p)}{R^{(1)}(1,p)}$ is the number of second layer packets we can transmit and $|Q_i|$ is the number of backlogged packets at link i whose arrival time is smaller than $\tau_1 + D$. D is a parameter which bounds on how far we can look ahead in the queue for second layer packets. D is needed to prevent the unfair situation where one flow gets a lower throughput than that without using superposition coding. Thus, the number of packets we transmit is upper bounded by $|Q_i|$.

It is straightforward to show that quantity $\frac{R^{(2)}(i,P-p)}{R^{(1)}(1,p)}$ that determines the number of packets in the second layer correcty accounts for the coding rates and modulations used in the 802.11a data rates.

Dividing N_i by the transmission time $s/R^{(1)}(1,p)$, and ignoring s since it is the same for all packets, we define normalized effective throughput:

$$T_i = N_i R^{(1)}(1,p). \qquad (2)$$

We vary p to maximize T_i. Let p_i^* be the optimal power. We choose the best performing link i^*, and denote the power p_i^* by p^*.

However, we apply superposition coding on links 1 and i only when it achieves better throughput than scheduling the two links separately during the same time interval. That is, $\frac{s}{R^{(1)}(1,P)}$ is used to transmit pkt_1 and the remaining time $\frac{s}{R^{(1)}(1,p)} - \frac{s}{R^{(1)}(1,P)}$ to transmit packets of link i. Thus, we have the following constraint:

$$\begin{aligned} &1 + min\left(\frac{R^{(2)}(i,P-p)}{R^{(1)}(1,p)}, |Q_i|\right) > \\ &\quad 1 + min\left(\left(\frac{1}{R^{(1)}(1,p)} - \frac{1}{R^{(1)}(1,P)}\right)R^{(2)}(i,P), |Q_i|\right). \end{aligned} \qquad (3)$$

If the link i is backlogged, the preceding equation is simplified to:

$$\frac{R^{(1)}(1,p)}{R^{(1)}(1,P)} + \frac{R^{(2)}(i,P-p)}{R^{(2)}(i,P)} > 1. \qquad (4)$$

Finally, we need to update certain scheduler specific parameters. For example, if the scheduler is proportional fair, one needs to update the average rate of a given next hop.

```
Process($pkt_1$) – On getting a packet $pkt_1$ from basic
scheduler. Let its next hop be $v_1$
let $T^* = 0$, $i^* = -1$ for all $i \neq 1$
for each link $i \neq 1$
  for each discrete rate $c_j$
      determine $p$ such that $c_j = R^{(1)}(1,p)$
      $N_i = 1 + min\left(\frac{R^{(2)}(i,P-p)}{R^{(1)}(1,p)}, |Q_i|\right)$
      $N_s = 1 + min((\frac{1}{R^{(1)}(1,p)} - \frac{1}{R^{(1)}(1,P)})R^{(2)}(i,P), |Q_i|)$
      if ($N_i R^{(1)}(1,p) > T^*$ and $N_i > N_s$) then
          $N^* = N_i$
          $T^* = N_i R^{(1)}(1,p)$
          $i^* = i$
          $p_i^* = p$
      endif
  endfor
endfor
if ($i^* \neq -1$) then
  superposition coding $pkt_1$ using power $p^*$ and
  $N^* - 1$ packets of $i^*$
else
  schedule $pkt_1$ only
endif
update scheduler specific parameters
```

Figure 1: Superposition scheduling algorithm *Gopp* at node *u*.

Assuming a FIFO queue, it is possible to show that our scheduling algorithm ensures that each link flow gets a throughput no smaller than it would get without superposition coding with a sufficient finite value for *D*. The full proof is in our technical report [5].

We can further extend the scheduler to directly handle PER. Let q_1 be the PER for the first layer of link (u, v_1). Let q_2, q_3 be PER of the first and second layer of link (u, v_i) respectively. Then the expression for N_i, which reflects expected number of delivered packets under independent losses, is revised to $(1-q_1) + min\left(\frac{R^{(2)}(i,P-p)}{R^{(1)}(1,p)}), |Q_i|\right) \times (1-q_2)(1-q_3)$. We can use the revised expression to compute N_i and N_s in our scheduler and add a loop for PER to search for the best combination.

We now illustrate with a simple example. Suppose we have two links l_1 and l_2. The maximum individual rates achievable by the two receivers are 9 and 36 respectively. In other words, $R^{(1)}(1,P) = 9$ and $R^{(1)}(2,P) = 36$. Assume that when they are transmitted using superposition coding, they can achieve 6 and 24. Thus, $R^{(1)}(1,p^*) = 6$ and $R^{(2)}(2,P-p^*) = 24$. Assuming sufficient queued packets, then $N^* = N_2 = 1 + 24/6 = 5$; $N_s = 1 + (1/6 - 1/9)36 = 3$. Since $N^* > N_s$, it is beneficial to use superposition coding.

As stated, the *Gopp* algorithm treats the basic scheduler as a blackbox. We can also design a scheduling algorithm that jointly schedules two packets. For details, please see the technical report [5].

4. SIMULATION EVALUATIONS

4.1 Methodology

We have implemented superposition coding as well as our MAC protocol and scheduling algorithm in ns-2 (Ver. 2.28). We assume that the basic scheduler is FIFO.

For our simulation setup, we use the 2-ray ground model with 0 dBi antenna gains and antenna heights of 1 meter. We use wireless nodes with maximum transmit power $P = 200$ mW. To implement a realistic packet decoding model, we use a lookup table for packet error rates (PER) [1] given an observed SINR. There is a separate PER curve for each 802.11g data rate. The SINR depends on the received signal power P_r, noise power $N_0 = -86$ dBm, and implementation margin $I_m = 5$ dB as specified in the 802.11g standard. These parameters produce maximum transmission ranges similar to those of the Cisco Aironet 802.11g in a typical outdoor environment.

When a packet is received in the physical layer, we decode it as follows. If the packet is not a superposition coded packet, we compute SINR $= \frac{P_r I_m}{N_0}$ and look up the PER correponding to the data rate at which the packet was transmitted. The packet is dropped with a probability equal to the PER.

If the packet is a superposition coded packet, we compute the SINR for the basic layer as SINR $= \frac{P_r p_1 I_m}{P_r(1-p_1)+N_0}$ where $p_1 \in [0,1]$ is the fraction of power allocated to the basic layer packet. The SINR lookup and decoding for the basic layer are done in the same way as a non-superposition coded packet. If the basic layer is decoded successfully, we compute the SINR for the additional layer as SINR $= \frac{P_r(1-p_1)I_m}{N_0}$. SINR lookup and decoding for the additional layer are also done in the same way as the non-superposition coded packet.

Note that there may be multiple packets packed into the additional layer of a superposition coded packet. These are unpacked in the MAC layer before being delivered further up the protocol stack.

To handle channel estimation, the physical layer calculates the channel gain $h = \frac{P_r}{P}$ for each received transmission. This value along with the transmission source address is passed up to the MAC layer, which manages a table of channel gains for each link. These channel gains are used to compute SINR values in the scheduling and routing layers.

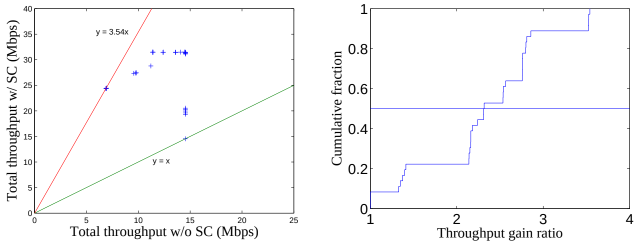
We assume that all transmitted packets are 1500 bytes. We also disable RTS/CTS.

4.2 Simulation Results

4.2.1 Single gateway

We start with the case of two receivers and a single gateway transmitter. The location of receiver 2 is fixed at 10 m away from the transmitter. We vary the position of receiver 1 at 100 random locations. We run two UDP flows for a total of 60 seconds. Each UDP flow generates CBR traffic.

Figure 2-a is a scatter plot comparing the total throughput with and without using our SC scheduler. The *x*-axis is the throughput with SC scheduling turned off, and the *y*-axis repeats each experiment with SC scheduling turned on. We add two lines in the figure: $y = x$ and $y = 6x$. We observe that with standard 802.11 scheduler, the transmitter achieves total throughput in the range from 6 to 14 Mbps; with our SC scheduler, the total throughput improves to the range from 14 to 32 Mbps.



(a) Throughput gain with SC scheduler (b) CDF of throughput gain ratio

Figure 2: Single transmitter and two receivers (receiver 2 has a fixed position close to the transmitter).

Figure 2-b plots the cumulative distribution function (CDF) of throughput gain ratio, where the throughput gain ratio is defined as the ratio of throughput with SC scheduling to that without it. The median gain ratio is 2.5, corresponding to a throughput increase of 150%. We observe that the typical ratio is around 2 to 3, as indicated by the steep increase of the curve.

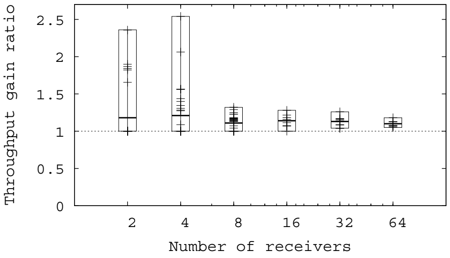


Figure 3: Throughput gain ratio of multiple receivers. There is a single transmitter.

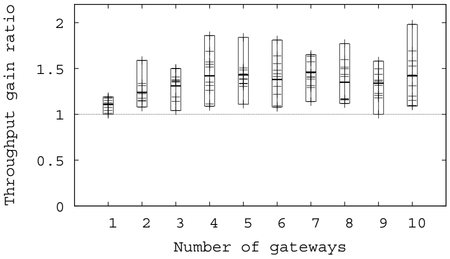


Figure 4: Throughput gain ratio of multiple flows and multiple gateways forming a mesh network.

Next we study the effect of the number of receivers. In this class of experiments, we place all receivers randomly. Thus, this class of experiments is different from the preceding one which always has a receiver close to the transmitter. Typically gateways are placed close to a set of receivers, so the preceding experiment might be closer to reality and the results from this set of experiments provide a lower bound. We vary the number of receivers from 2 to 64. Figure 3 plots the min (the bottom of each bar), max (the top), average (the heavy middle horizontal line), and individual results for each given number of receivers. We observe that when we initially increase the number of receivers from 2 to 4, it increases diversity and creates opportunities for superposition coding. For example, the average gain increases from 18% when the number of receivers is 2 to 21% when the number of receivers is 4. The maximum gain we observed is 154% when the number of receivers is 4. Further increasing the number of receivers can reduce the average gain. However, even when there are 64 simultaneous active receivers, we still achieve an average gain of 10%.

4.2.2 *Gateway mesh networks*

Finally, we evaluate the throughput gain when there are multiple gateways forming a mesh network. This evaluates the scalability of our system with respect to the number of transmitters. We randomly place 50 nodes in a square area of 600 meters by 600 meters. We designate a number of nodes with highest degreee as gateway nodes. The number of gateways ranges from 1 to 10, thus making the network dense relative to typical deployments. We fix the number of flows to 25. All flows are from gateway nodes to non-gateway nodes. A non-gateway node communicates with the closest gateway nodes through a multi-hop path. We use the inverse of the link data rate as the routing metric. Only links with PER $< 10\%$ are used. This routing metric tries to minimize the total transmission time for a given flow.

Figure 4 plots the results. We see that, our system consistently achieves average gains ranging from 24% to 46% when there are multiple gateways. When the number of gateways is 10, our system gains up to 98%, with average at around 42%.

5. IMPLEMENTATION

Finally, we present results from measurement on an implementation of our algorithm using the GNU Radio platform. In this setup, there is a transmitter and two receivers. The transmitter transmits 20 packets. When superposition coding is enabled, the transmitter sends packets with BPSK modulation in both layers, with 70% of the power allocated to the first layer and 30% to the second layer.

Scheme	Norm. exp. trans. time	Gain ratio
No Coding	3.92	1
Superposition	2.88	1.4

Table 1: Testbed performance of superposition coding.

We report in Table 1 the mean normalized total transmission time for the two receivers to recover their packets, where one unit of time is the time to transmit one packet at 6 Mbps. We observe that throughput gains are largely consistent with our simulation results.

6. CONCLUSIONS AND DISCUSSIONS

This paper is part of our larger project founded on the belief that the key to increase network capacity is to treat a wireless network as a medium that propagates information rather than packets as they are originated from the sources. Many information transmission techniques such as superposition coding, relay channel, network coding, and MIMO must be accommodated in higher layers to realize their full potential.

This paper presented our first step in designing a MAC protocol to take advantage of superposition coding as an information propagation technique. In our next step, we have formulated the *transmitter packet mixing problem* and designed a MAC protocol utilizing both network coding and superposition coding. Our preliminary evaluations show that this integrated protocol can improve throughput in a wide range of scenarios and outperform the better performer of network coding and superposition coding by more than 30% in a wide range of scenarios.

The ultimate goal is to design a framework that can accommodate a wide range of information theory and physical layer techniques. To achieve this goal, there is a need to re-consider the whole wireless network architecture starting from its basic link abstraction.

7. ACKNOWLEDGMENTS

Richard Alimi and Y. Richard Yang were supported in part by grants from the U.S. NSF. We are grateful to Mehul Motani, our shepherd, for his help in revising the paper. Gerhard Kramer and Hao Wang have given us valuable comments.

8. REFERENCES

[1] K.-Y. Doo, J. young Song, and D.-H. Cho. Enhanced transmission mode selection in IEEE 802.11a WLAN system. In *Proceedings of Vehicular Technology Conference*, pages 5059–5062, Sept. 2004.

[2] Ugly truth about mesh networks. `http://www.dailywireless.org/2004/06/28/ugly-truth-about-mesh-networks/%`.

[3] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, Jan. 2001.

[4] D. Kotz and K. Essien. Analysis of a campus-wide wireless network. In *Proceedings of the Eighth International Conference on Mobile Computing and Networking (Mobicom)*, pages 107–118, Atlanta, GA, Nov. 2002.

[5] L. E. Li and *et al.* Supernet: Superposition coding for wireless mesh networks. Technical report, Computer Science, Yale University, available at http://www-net.cs.yale.edu/publications/supernet.pdf, 2007.

[6] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, May 2005.