muNet: Harnessing Multiuser Capacity in Wireless Mesh Networks

Li (Erran) Li[⋆] Richard Alimi[†] Ramachandran Ramjee[§] Harish Viswanathan[⋆] Yang Richard Yang[†]

[⋆]Bell Labs, Alcatel-Lucent, Murray Hill, NJ [§]Microsoft Research India, India [†]Yale University, New Haven, CT

***Abstract*— We present muNet, a wireless mesh network design and implementation to harness the multiuser capacity of wireless channels. Traditionally, media access control is designed to schedule one transmission between one sender and one receiver without interference at any given time. However, this design is suboptimal in terms of achieving the multiuser capacity of multi-access wireless channels. In muNet, we implement effective physical layer techniques called superposition coding and successive interference cancellation to enable** *simultaneous unicast transmissions* **from a single transmitter to multiple receivers as well as from multiple transmitters to a single receiver. We design the first practical MAC protocol that leverages such a physical layer and exposes the multiuser capacity to upper layers. We also present a simple, effective routing protocol that increases simultaneous transmission opportunities for the MAC layer. A proof-of-concept muNet is implemented on the GNU Radio platform. Measurements on the implementation shows that the throughput gains of muNet are significant (up to 93%).**

I. Introduction

Traditional wireless mesh network architectures treat wireless channels as unicast point-to-point links. This link abstraction is simple but hides the significant potential of "multiuser" wireless channels. By multiuser capacity of a wireless channel, we mean the optimal trade-off achievable by any multiple access schemes [7]. In many situations, techniques achieving multiuser capacity may provide significant throughput gains compared with the orthogonal division techniques used in current wireless mesh networks.

Achieving these gains in practice, however, poses significant design and implementation challenges. In particular, existing MAC and routing layers are not designed to consider multiuser capacity; real implementation of physical, MAC, and routing layers harnessing multiuser capacity poses significant challenges as well. The major contribution of this paper is muNet, a design and implementation to demonstrate how to harness multiuser capacity of wireless channels.

The physical layer of muNet utilizes superposition coding and successive interference cancellation (SIC) [2]. These are well-known physical layer techniques, and we do not claim novelty on these concepts. Specifically, superposition coding and successive interference cancellation enable *simultaneous unicast transmissions* from a single transmitter to multiple receivers (downlink) as well as from multiple transmitters to a single receiver (uplink). In the downlink, the single transmitter uses superposition coding to encode information simultaneously to multiple receivers and in both uplink and downlink, receivers utilize successive interference cancellation to decode information destined for them. Extensive theoretical analysis has shown that superposition coding and successive interference cancellation are more efficient than the orthogonal division physical layer techniques used in current wireless mesh networks. For example, in [1], Bergmans has shown that superposition coding achieves optimal capacity for an additive white Gaussian noise (AWGN) physical channel, while the traditional schemes under-perform significantly.

The existence of beneficial settings does not imply that a network can harness the benefits. In large, dynamic networks, simultaneous transmission opportunities can be challenging to identify and utilize. For example, being oblivious to multiuser capacity, existing MAC layer designs cannot utilize simultaneous transmission opportunities such as that in the aforementioned example. On the other hand, the MAC layer of muNet utilizes an opportunistic scheduler that identifies opportunities for superposition coding and successive interference cancellation, and then arbitrates the medium for simultaneous transmission/reception. The MAC layer of muNet is distributed, and operates in dynamic settings.

Similarly, a key challenge for the routing layer of muNet is to discover and prefer routes for flows over a wireless mesh network that can increase multiuser opportunities. In particular, muNet employs a novel local rerouting technique to reroute flows over a wireless mesh network to increase superposition coding and successive interference cancellation opportunities.

To summarize, in this paper, we present the first practical system that takes advantage of superposition coding and successive interference cancellation in wireless mesh networks. We make two main contributions.

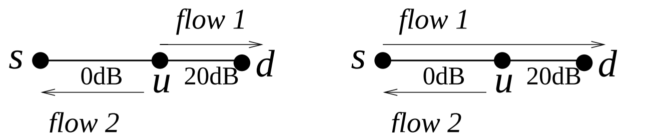
1) We design the first higher layer protocols (*i.e.*,, MAC and routing) that leverage underlying physical layer techniques to harness multiuser capacity.
2) We demonstrate proof-of-concept implementation of our techniques using the GNU Radio platform. We evaluate our design using a small-scale implementation testbed.

II. Motivating Examples

Current wireless mesh network protocols are not designed to take full advantage of the multiuser capacity of wireless channels. In this section, we highlight the need for new MAC and routing protocols that can harness the multiuser capacity of wireless channels. A major advantage of superposition coding and successful interference cancellation is the expanded capacity region [7]. Consider the one sender and two receiver case. Let P be the total transmission power, and

p the power allocated to the basic layer (*i.e.*, to receiver 1). Let $|h_1|$ and $|h_2|$ be the channel gains to receiver 1 and 2 resprectively. Then according to Shannon capacity formula for an AWGN channel, the achievable rate to receiver 1 is $\log_2\left(1+\frac{p|h_1|^2}{(P-p)|h_1|^2+N_0}\right) bits/s/Hz$ where N_0 is the background noise. On the other hand, due to successive interference cancellation at receiver 2, the achievable rate to receiver 2 is $\log_2\left(1+\frac{(P-p)|h_2|^2}{N_0}\right) bits/s/Hz$.

A. Media Access Control and Scheduling



(a) Intelligent scheduler needed (b) Side information benefits.

Fig. 1.

We now illustrate the need for an intelligent scheduler at the MAC layer that utilizes one-to-many transmissions. In Figure 1-a, at node u, suppose we have 25 packets of flow 1 and 3 packets of flow 2. Assume all packets are of the same size z bits. What is the best way to schedule the transmissions in order to minimize the total transmission time? For example, given a rate pair for the two flows of $(3, 0.9)$ (calculated using the above Equations), we would use superposition coding until one of the flow completes and then use traditional unicast to complete the other flow. Thus, we would superpose $10z$ bits of flow 1 at rate 3 simultaneously with $3z$ bits of flow 2 at rate 0.9 for time = $\frac{3z}{0.9}$ and then send the remaining $15z$ bits of flow 1 at unicast channel capacity of 6.66 for time = $\frac{15z}{6.66}$, resulting in a total transmission time of $\frac{3z}{0.9}+\frac{15z}{6.66}$. Suppose, instead we used a different rate pair of $(5, 0.6)$. In this case, the entire transmissions of flow 1 and 2 can be superimposed, resulting in a transmission time of $\frac{3z}{0.6}$. Thus, the second rate pair would reduce the transmission time by over 10% compared to the first rate pair. Can we design a scheduling algorithm that would automatically pick the optimal rate pair?

B. Superposition Coding with Side Information

Normally the first layer signal treats the second layer as noise. However, if the first layer receiver has the packet of the second layer a prior (thus, exploiting network layer information similar to Analog Network Coding [5]), then the first layer receiver does not have to treat the second layer signal as noise. It can simply subtract the second layer signal before decoding. Figure 1-b shows such a case where the receiver of the second flow S is simply the sender of the first flow. Without utilizing side information, node u can send a superposition coded packet with rate pair $(3, 0.9)$ for link ud and us. Using side information, we can transmit a superposition coded packet successfully using a rate pair $(3.85, 0.9)$, resulting in a 27% improvement over link ud. Can the MAC be designed to take advantage of this?

C. Impact of Routing

Routing agnostic of flow level information may limit opportunities for superposition coding. With limited local rerouting, we can create such opportunities.

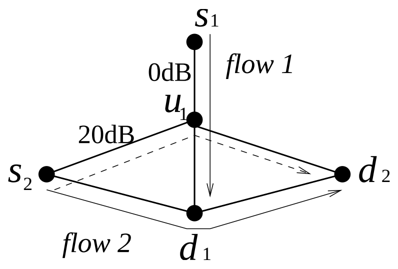


Fig. 2. Reroute flows to create superposition coding and SIC opportunities.

Figure 2 shows an example. Suppose all nodes are in the same interference domain. Suppose the SNR on links s_1u_1 and u_1d_1 are 0dB; the rest of the links all have SNRs of 20dB. Suppose flow 1 is routed from s_1 to d_1 through u_1 and flow 2 is routed from s_2 to d_2 through d_1. Suppose the two flows are backlogged. Consider the traditional 802.11 MAC that provides equal access to all nodes and schedules transmissions independently. Both flows achieve the same throughput of $1/2*(6.66/(6.66+1)) = 0.43$ bits/s/Hz. Suppose we reroute flow 2 from s_2 to d_2 through u_1 (dotted line). Now for the first transmission, both s_1 and s_2 can simultaneously transmit to u_1, say, using the rate pair $(0.9, 3)$. This represents a power split of 13.4% to link s_1u_1 and 86.6% to link s_2u_1. When u_1 receives the transmission from both senders, it decodes using successive interference cancellation. Node u_1 then transmits to d_1 and d_2 simultaneously using superposition coding, say, using the same rate pair $(0.9, 3)$. This represents a power split of 92.8% to link u_1d_2 and 7.2% to link u_1d_1. The reroute with superposition coding has allowed us to achieve a throughput of 0.45 and 1.5 for flows 1 and 2, respectively, resulting in 245% throughput gains for flow 2.

These examples illustrate the need for new MAC and routing protocols that can leverage the significant throughput gain opportunities available in a multi-user environment. Overall, our goal is to build practical, implementable solutions and thus, we try to make minimal modifications to existing protocols. We next describe our design for PHY, MAC and routing protocols.

III. Enabling multiuser channel gain: physical layer and MAC

In this section, we present changes required in physical layer and MAC that enable multiuser channel gain.

A. Physical Layer

The key requirements for the physical layer are channel estimation and functionalities at the transceivers that enable superposition coding and successive interference cancellation. Channel estimation is done using a pilot signal (SFD field of 802.11 PLCP header). The channel estimation at the receiver is then fed back to the sender by our MAC (see Section III-B).

Information	How obtained
Channel Estimation (SNR of sender)	SFD field of 802.11 PLCP header
SNR at each neighbor (SNR table)	SNR fed back in modified ACK/CTS
Rate/power for superposition coding	Modified SIGNAL field of 802.11 PLCP
Pending pkts at neighbor (Pending table)	Fed back in modified ACK/CTS
Rate/power for simultaneous uplink	Trigger header of modified DATA/RTS
Pkts sent by neighbor (packet ID cache)	Overhearing modified SIGNAL field
Sent packet cache, indexed by packet ID	Packets sent by self

TABLE I
ADDITIONAL INFORMATION REQUIRED FOR SUPPORTING SUPERPOSITION CODING AND SIC.

We further modify the SIGNAL field of 802.11 PLCP header to incorporate the rate and power for each layer (required for decoding superposition coded packets) and packet ID (required for utilizing side channel information gains). A list of additional information needed for supporting SIC and how they are obtained is summarized in Table I.

Estimated SNR of a link is stored in a *SNR table* which maps each link to the corresponding SNR. It contains the most recent SNR estimation and an averaged SNR (averaged over a few seconds). The table is periodically refreshed based on the most recent timestamp of each entry. Entries of current SNR older than the channel coherence time are deleted. In a fixed environment typical for muNet, the channel coherence time is in the order of several hundred milliseconds to a few seconds. Each entry also has flags, e.g. indicating whether a particular value has been fed back to the sender or not. This SNR table is maintained and shared by the physical, MAC and routing layers. Transmissions on links without estimated SNR values are sent using the base rate as is done today.

We now delve into more details on channel estimation and interference cancellation. Nodes promiscuously listen to transmissions. A node needs to distinguish between different *types* of transmissions, viz., single sender traditional transmission, single sender superposition transmission, and multiple sender transmissions. A superposition coded transmission is identified by multiple rate/power information fields in the physical layer header. If there is a single rate and power in the physical header, then the node tries to decode the packet. If it can not decode, then it just drops it. If it can decode, it then proceeds to decode the additional layer, if any, by first removing the first layer. If the remaining signal after subtracting the first layer signal has significant energy left, then this transmission is the result of transmissions from multiple senders. If the remaining signal's energy is comparable to noise, then the original signal is the result of a transmission from a single sender. We do channel estimation only using single-sender transmissions, though, it is theoretically possible to do channel estimation using multiple-sender transmissions using additional techniques. The IDs (source, destination, sequence number) of decoded packets are cached in a *packet ID cache*. It contains the IDs of packets recently sent by neighbors. This buffer helps finding superposition coding opportunities with side information.

B. Medium Access Control

To unlock the multiuser channel gain, we do not need a complete redesign of media access control protocols. We show how we can achieve this by modifying the dominant 802.11 MAC. We base the message flow of our MAC on 802.11 DATA/ACK and optionally RTS/CTS. We make extensions to address three issues: (1) enable feedback of estimated SNRs to the transmitter; and (2) enable many-to-one (referred to as uplink) and one-to-many (referred to as downlink) transmissions; (3) maintain long-term fairness to other competing nodes.

ACK for channel feedback and traffic indicator: As described earlier, channel estimations are done in the physical layer during the reception of DATA (and ACK) packets from single senders. The channel estimations are fed back to the sender in the ACK packets. We modify the ACK packets to enable feedback of channel estimations of one or more links.

The ACK is also modified to indicate whether there are pending packets from the receiver to the sender and if so, how many. Each node in the MAC layer maintains this *pending traffic table*. It contains a *count* entry for each neighbor, and is refreshed periodically by future ACK packets containing updated information, or cleared by time outs. The counter of each entry is subtracted with each received DATA packet from the corresponding neighbor.

Piggyback trigger header in DATA for uplink transmission: For each node A, we enable uplink many-to-one transmission to A as follows. Before scheduling A's packet for transmission, it checks the pending traffic table and SNR table to see if there are neighbors that can benefit from simultaneous transmissions to A. If such opportunities exist, A will append a trigger header to its DATA packet which contains the neighbors involved (say B and C) and their power and rate allocation using SNR table information. When B and C receives the trigger header, after a SIFS interval, they will transmit the ACK and one or more packets depending on information in the trigger header. Note that *no synchronization is needed*. This is because the receiver can always decode the first layer by design (treating the second layer as noise), subtract, and then decode the additional layer. Note that, if there is no uplink simultaneous transmissions opportunities, the trigger header's rate and power will be used to send *time overlapping ACKs* from B and C. Each node keeps recent sent packets in a small *sent-packet cache*.

Modification to RTS/CTS packets: We have described our design without the need of RTS and CTS packets. However, if RTS/CTS are enabled, they can help provide more timely channel estimations. We now describe how we modify them to suit our purpose. 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 can be addressed to two receivers, it can trigger the transmission of one CTS packet from each. If channel estimations of the receiver are recent, then the RTS also contains the rate and power allocation for time-overlapping transmissions of the two CTSs. If no channel estimations are available, the two CTSs should be separated by SIFS to avoid collision with the first addressed node sending first. Each CTS message contains the estimated SNR calculated using the pilot symbols in the preceding RTS message. The reported SNRs will be stored in the SNR table. Similarly, each superposition coded DATA packet will require two ACK packets, one from each receiver. The two ACK packets can be sent overlapping to the transmitter if there is a trigger header present. Otherwise, the two ACKs have to be sent separately with a SIFS time in between (receiver of first layer packet sending first).

For the uplink between a sender B and receiver A, the receiver A can suggest another sender C to transmit at the same time with B, and the respective rates. These information are encoded in the CTS message.

C. MAC Scheduling: A Simple, Local Scheduler

We need an intelligent scheduler at the MAC layer to take advantage of multiuser channel gains. We take an opportunistic approach. Our scheduler extends a given basic scheduler and treats it as a blackbox. Note that, if long-term fairness to long-lived flows is required, nodes may need to modify channel access parameters such as the contention window. For the basic downlink scheduling algorithm G_{opp}, please refer to [6]. It can be easily extended to accomodate side information.

For uplink scheduling, we will make use of the pending traffic table instead of the packet buffer at node u. There are two cases depending on whether we keep the total power of both transmissions fixed at P. We use $R^{(k)}(i,p)$, $\forall k = 1,2$ to denote the transmission rate for link i at transmission power level p when the packet to link i is encoded as the k-th layer. For the first case where total power is fixed at P, $Gopp$ can be readily applied based on the duality theory [4] with the following definition of rate function: $R^{(1)}(i,p) = \frac{p|h_i|^2}{(P-p)|h_j|^2+N_0}$ and $R^{(2)}(i,p) = \frac{p|h_i|^2}{N_0}$ where i, j is the link which transmits the first and second layer respectively. For the second case, we look at one rate pair. Let link i be the link with a weaker channel. Find the minimal power P_i such that link i can achieve $R^{(1)}(i,P)$. By treating i's transmission using P_i as interference, we calculate r_j, the rate link j can achieve. This design has the nice property that sender v_i's rate of link i is maximum

IV. Revealing more multiuser channel gains: two-Hop Rerouting

We take a minimalist approach in which we assume the system has chosen a basic routing protocol, e.g. ETX [3]. Our minimalist approach helps us evaluate the possible range of benefits before designing an end-to-end routing protocol. In our local rerouting approach, we ensure that rerouting is performed only when throughput is increased. In our MAC scheduling, our decision has been based on short-term SNRs. To avoid short-term routing fluctuations, in our routing protocol, we use the *average* SNRs which averages short-term SNRs for a relatively longer duration.

We design an algorithm, referred to as local 2-hop rerouting, to opportunistically take advantage of multiuser channel gain. We consider link flows, i.e. all traffic going through a set of links. In general, we need to deal with the following cases:

- only uplink SIC opportunity exists, and it happens on the first transmission;
- only uplink SIC opportunity exists, and it happens on the second transmission;
- only downlink superposition coding opportunity (on the second transmission) exists;
- Both uplink SIC and downlink superposition coding opportunities exist.

We now formalize the intuition for the last case. We will use the notations in Figure 2. We rename links s_1u_1,u_1d_1, s_2d_1, d_1d_2, s_2u_1, u_1d_1 as link 1 to 6. We assume that transmissions on link 1 to 4 occupies τ_i fraction of the channel for $i = 1,2,3,4$. These time fractions are estimated at node u_1 by overhearing transmissions from neighbors. We assume link flow conservation. The total throughput for no re-routing is $T_1 = R^{(1)}(1,P)\tau_1 + R^{(1)}(3,P)\tau_3$.

We need to achieve a better throughput after rerouting. Let the time allocated to the uplink and downlink transmission be be τ_1', and τ_2'. Let $\bar{R}^{(1)}()$ denotes the rate function for the first layer uplink transmission. $\bar{R}^{(2)}()$ is the same as $R^{(2)}()$. For flow conservation, we have $\bar{R}^{(1)}(1,p_1)\tau_1' = R^{(2)}(2,p_2)\tau_2'$ and $R^{(2)}(5,p_5)\tau_1' = R^{(2)}(6,P-p_2)\tau_2'$.

Let $T_2 = \bar{R}^{(1)}(1,p_1)\tau_1' + R^{(2)}(5,p_5)\tau_1'$. We also require that each individual flow achieve a throughput no less than what it gets without rerouting. Thus, with these constraints, we want to find p_1, p_2, p_5 that maximize T_2 as follows.

$$\begin{array}{ll} \max & T_2 = \bar{R}^{(1)}(1,p_1)\tau_1' + R^{(2)}(5,p_5)\tau_1' \\ s.t. & \bar{R}^{(1)}(1,p_1)\tau_1' = R^{(2)}(2,p_2)\tau_2' \\ & R^{(2)}(5,p_5)\tau_1' = R^{(2)}(6,P-p_2)\tau_2' \\ & \bar{R}^{(1)}(1,p_1)\tau_1' \geq R^{(1)}(1,P)\tau_1 \\ & R^{(2)}(5,p_5)\tau_1' \geq R^{(1)}(3,P)\tau_1 \\ & \sum_{i=1}^{4}\tau_i = \sum_{i=1}^{2}\tau_i'. \end{array} \tag{1}$$

This problem can be easily solved by trying all possible combinations of rates on link 1 and 2. For each rate pair, its associated p_1 and p_2 can be determined. For each p_1, p_2, we can solve for p_5 (based on first two equations in Eq. 1). Once we have these three parameters, we can check the feasibility of the rest of the constraints. We pick the best T_2 as the solution.

Our routing protocol is implemented as follows. Each node keep tracks of the fraction of time each of its link flow occupies the channel. A sender s_2 piggybacks a solicitation message in its data packet if it is transmitting without any superposition coding or SIC benefit, and its data rate is deemed to be low. Its data rate $R^{(1)}(3,P)$ is then stamped in the solicitation message. When node u_1 gets the solicitation from s_2, u_1 starts the calculation using estimated channel parameters in the local SNR table. If the calculation shows that superposition coding or SIC is beneficial, then u will notify s_2. Note that any neighbor can respond to the solicitation message. s_2 may get multiple replies. It can choose the best opportunity or choose the first one arrives.

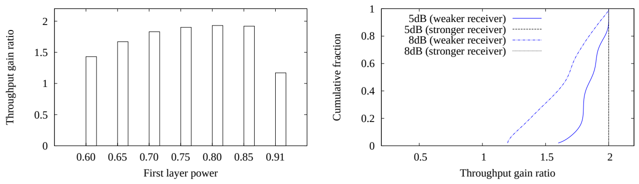
If there is no more superposition coding or SIC opportunity, and the 2-hop route is worse than the direct path, then u_1 can notify s_2 to switch back. Note that, channel allocation time τ_i depends on media access control and scheduling. Even if the MAC and scheduling do not allocate according to Eq. 1, our algorithm reduces the overall transmission time for any given set of packets which should benefit the overall network performance.

V. Experimental Results

A proof-of-concept muNet is build on the GNU Radio platform. Our overall implementation consists of 1000 lines of Python code and 3400 lines of C++ code. Each packet contains a header with a 16-bit CRC, and a 233-byte payload encoded at the rate of $\frac{233}{255}$ with Reed-Solomon encoding. The transmitter modulates each packet with BPSK at a low symbol rate (62500 symbol/s), and the receiver samples at 4x the symbol rate to process the received signal. Our performance metric is *throughput gain ratio*, which is defined as the ratio of throughputs achieved with and without using superposition coding and successive interference cancellation. Due to the software processing speed limitations of GNU radio, our experiments evaluate the benefit of muNet for realistic single-rate network settings. Our channel measurements show that the channel amplitude's standard deviation normalized by the mean is only 0.336, thus the channel in our test environment is stable.

Single AP with downlink traffic: We consider the basic case of one AP sender with two client receivers where the channels between the two clients to the AP are asymmetric (SNR difference between the two receivers is 5dB). We vary the transmission power allocated to two layers. It is intuitive that if the power level for the first layer is too low, the first layer signal will have a low decode probability, while if the power level for the first layer is too high, the second layer signal will have a low decode probability. Therefore, there is an optimal point in the middle.

Figure 3-a confirms the intuition. When 80% of total power is allocated to the first layer, the overall throughput gain ratio achieves its maximal value of 1.93. Also, note that the throughput gain ratio is not very sensitive to small changes in power allocation, with first layer power values of 70% to 90% providing gains close to the maximal value. This indicates that there is some margin in computing the optimal MAC power splits, providing robustness to channel estimation errors.



(a) Impact of power allocation (b) Throughput gain ratio

Fig. 3. Single AP downlink case

In addition, we investigate the effect of channel asymmetry on packet decode probability and throughput gain ratio. We fix the position of the receiver with stronger channel and change the position of the receiver with weaker channel so that SNR difference between the receivers is increased from 5dB to 8dB. We then perform 50 experiments for the two cases using their respective optimal power splits. Figure 3-b throughput gain ratio, for each of the receivers for the two cases. We observe that for the receiver with stronger signal, it is able to achieve close to 100% packet decode probability in both cases, resulting in a throughput gain ratio of 2. For the receiver with weaker signal, when it is near the transmitter, it can enjoy an average of 93% packet decode probability and an average throughput gain ratio of 1.87 (thus, an overall throughput gain for this case of 1.93 seen in Figure 3-a). When we move the weaker receiver farther away, it suffers from the channel attenuation and its packet decode probability drops to an average of 82%. Therefore, its average throughput gain ratio also drops to 1.63 and an overall gain of 1.82.

Single AP with uplink traffic: We consider one AP receiver and two client senders. Uplink successive interference cancellation is used for the pair of asymmetric links. Since this is the uplink case, there is no need to split the transmission power. Instead, we vary the locations of the transmitters and this results in change in the receiving power ratio between the two layers. The throughput gain ratio in this scenario reaches up to 1.93. The average gain ratio is 1.62.

Local re-routing: We also experimented on a simple topology with re-routing. The average gain with re-routing is 1.49. significant additional gain can be obtained when side information is enabled. When side information is enabled, the average gain ratio increases to 1.65.

VI. Conclusion and future work

It is our belief that the key to increase end-to-end network throughput is to treat a wireless network as a medium that propagates information rather than packets as they are originated from the sources. As a first step to move towards this goal, we replace the point-to-point link abstraction with one that can achieve the fundamental capacity limits of the multiuser wireless channel. We show how to design physical layer, MAC and routing within this new wireless network architecture. Our GNU radio implementation results demonstrate that the throughput gains can be substantial.

There are many avenues for future studies. We would like to extend our prototype network to build a larger-scale network. We would like to investigate fundamental new practical techniques and design principles that help to close the capacity gap between a wireless network that propagates information and one that transports packets.

References

[1] P. P. Bergmans. A simple converse for broadcast channels with additive white Gaussian noise. *IEEE Transactions on Information Theory*, IT-20:279–280, 1974.

[2] T. M. Cover. Broadcast channels. *IEEE Transactions on Information Theory*, IT-18:2–14, 1972.

[3] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the Ninth International Conference on Mobile Computing and Networking (Mobicom)*, San Diego, CA, Sept. 2003.

[4] N. Jindal, S. Vishwanath, and A. Goldsmith. On the duality of gaussian multiple-access and broadcast channels. *IEEE Transactions on Information Theory*, 50(5):768–783, 2004.

[5] S. Katti, S. Gollakota, and D. Katabi. Embracing wireless interference: Analog network coding. In *Proceedings of ACM SIGCOMM '07*, Kyoto, Japan, Aug. 2007.

[6] 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.

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