Opportunistic Use of Client Repeaters to Improve Performance of WLANs

Paramvir (Victor) Bahl, *Fellow, IEEE*, Ranveer Chandra, Patrick P. C. Lee, Vishal Misra, Jitendra Padhye, Dan Rubenstein, *Member, IEEE*, and Yan Yu

***Abstract*—Currently deployed IEEE 802.11 WLANs (Wi-Fi networks) share access point (AP) bandwidth on a per-packet basis. However, various stations communicating with the AP often have different signal qualities, resulting in different transmission rates. This induces a phenomenon known as the *rate anomaly problem*, in which stations with lower signal quality transmit at lower rates and consume a significant majority of airtime, thereby dramatically reducing the throughput of stations transmitting at higher rates.**

We propose *SoftRepeater*, a practical, deployable system in which stations cooperatively address the rate anomaly problem. Specifically, higher rate Wi-Fi stations opportunistically transform themselves into repeaters for lower rate stations when transmitting data to/from the AP. The key challenge is to determine when it is beneficial to enable the repeater functionality. In view of this, we propose an initiation protocol that ensures that repeater functionality is enabled only when appropriate. Also, our system can run directly on top of today's 802.11 infrastructure networks. In addition, we describe a novel, zero-overhead network coding scheme that further alleviates undesirable symptoms of the rate anomaly problem. Using simulation and testbed implementation, we find that SoftRepeater can improve cumulative throughput by up to 200%.

***Index Terms*—IEEE 802.11, rate anomaly, wireless.**

I. Introduction

As CORPORATIONS move to all-wireless offices and a culture of mobility takes root, performance of such networks becomes paramount. In traditional corporate Wi-Fi networks, access points (APs) are generally sparely deployed. When heavily used, such networks suffer from the well-known *rate anomaly problem* [19]. This problem arises when multiple Wi-Fi stations transmit packets at different transmission rates. The IEEE 802.11 protocol arbitrates channel access requests on a per-packet basis. Assuming that all stations transmit packets of equal size, the stations that use lower transmission rate consume more airtime. This often severely limits the throughput of stations that are able to transmit at higher rates.

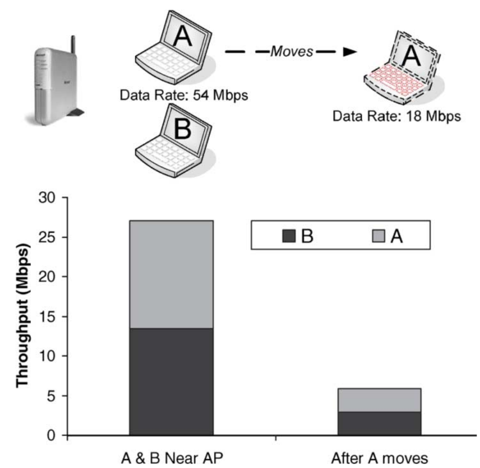


Fig. 1. The rate anomaly problem. B's throughput drops to 25% even though it never moved.

This problem is demonstrated experimentally in Fig. 1. A testbed with two laptops (stations), A and B, are associated to a single AP in IEEE 802.11a mode. Each station sends UDP packets to the AP as fast as it can. When both stations are close to the AP, both have good signal strength and transmit packets at their highest possible rate; each station receives a UDP throughput of 13 Mbps.[1] When station A moves away from the AP, its signal strength lowers, and a built-in auto-rate algorithm reduces A's transmission rate to 18 Mbps, increasing the time needed for A to transmit and receive packets. Since A and B share the medium on a packet-by-packet basis, B's throughput decreases as well—in this case of our experiment, by 75%, even though B never moved. This experiment conclusively demonstrates that rate anomaly can occur, and when it does, it reduces throughputs substantially in Wi-Fi networks.

A variety of proposed solutions, discussed in more detail in Section VI, address the rate anomaly problem. However, they have the following limitations: requiring dedicated hardware repeaters (e.g., [11], [12], [35]), making changes to the MAC layer (e.g., [26]–[28]), or constructing multihop networks from existing stations in *ad hoc* mode (e.g., [14]). Hence, they either increase cost, do not conform to currently deployed infrastructure networks, or cannot be activated on demand only when providing benefit.

In this paper, we describe a different approach: a practical, deployable system called *SoftRepeater*, which enables stations

Manuscript received January 23, 2009; revised June 09, 2009; approved by IEEE/ACM Transactions on Networking Editor K. Ross. First published July 21, 2009; current version published August 19, 2009. An earlier and shorter conference version of this paper appeared in ACM CoNEXT'08 [5].

P. Bahl, R. Chandra, and J. Padhye are with Microsoft Research, Redmond, WA 98052 USA (e-mail: bahl@microsoft.com; ranveer@microsoft.com; padhye@microsoft.com).

P. P. C. Lee is with Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong (e-mail: pclee@cs.columbia.edu).

V. Misra and D. Rubenstein are with Department of Computer Science, Columbia University, New York, NY 10027 USA (e-mail: misra@cs.columbia.edu; danr@cs.columbia.edu).

Y. Yu is with Google Inc., Mountain View, CA 94043 USA (e-mail: yanyu2005@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2009.2026414

[1]The sum is less than 54 Mbps due to protocol overheads.

(known as *repeaters*) with good signal strength and high transmission rates to *opportunistically* act as relays for stations (known as *clients*) with poor connectivity to the AP and low transmission rates. Our system requires no changes to the 802.11 MAC. Also, it is implemented entirely in software that runs on participating stations, thereby requiring no changes to the AP.

One key challenge is to ensure that the system is activated only when beneficial to all parties who suffer from the rate anomaly problem. For example, if the overall network utilization is low, there is no need for repeaters. It is necessary to have *practical* prediction algorithms that identify when the system would offer benefit.

Another key challenge is that once our system is activated, we require that the repeater can *reliably* send and receive traffic to/from both the AP and the client. This can only be achieved by having the repeater alternately switch between the infrastructure mode (for communication with the AP) and the *ad hoc* mode (for communication with the client). The practical needs of switching between the two modes are detailed in [8]. Thus, our system needs to efficiently switch between the two modes and determine the fraction of time spent on each mode to ensure the fairness of throughput of both the repeater and the client.

The algorithms and protocols are embodied in the *SoftRepeater* agent that runs on participating stations. The agent uses VirtualWiFi [8], [22] to support the repeater functionality in the common case where each station has only one radio available. This implementation is particularly attractive because the repeater is able to exploit available frequency channels to provide good performance without requiring extra hardware. If multiple radios are available, SoftRepeater can use them in conjunction with multiple channels to further boost the performance of the network.

In the context of our system, our important research contributions are the following:

- formalizing how the SoftRepeater system addresses the rate anomaly problem as a set of utility maximization problems for different fairness requirements;
- an algorithm that enables stations to detect the rate anomaly problem in a Wi-Fi network and then predict when invoking SoftRepeater will alleviate the problem;
- the protocol utilized by stations to negotiate, reach consensus, and subsequently activate SoftRepeater functionality;
- descriptions of multiple-channel and low-overhead network coding techniques similar to [23] that further alleviate the rate anomaly problem and further boost overall throughputs;
- an implementation of the SoftRepeater system in Windows XP, with its performance evaluated in both Qualnet simulation and extensive experiments using our implementation on a testbed.

The results from our experiments and simulations show that, under the right conditions, the SoftRepeater protocol can improve the performance of Wi-Fi networks by up to 200%. Furthermore, the protocol is able to correctly determine when it is beneficial to turn on the repeater functionality.

The rest of the paper is organized as follows. Section II overviews the SoftRepeater architecture, and Section III discusses its implementation details. Section IV presents evaluation results. In Section V, we discuss various scenarios where SoftRepeater is useful. Section VI reviews related work, and Section VII concludes the paper and presents future work.

II. SoftRepeater Overview

Our opportunistic repeater framework, SoftRepeater, alleviates the rate anomaly problem [19], [33], which arises when stations within interference range of one another send packets at different data rates. This occurs commonly in practice, mostly due to the auto-rate algorithm of IEEE 802.11 that adjusts the transmission rate of a wireless card based on RF signal quality and excessive packet loss. These two properties are often quite varied across stations in a network. They can be due to: 1) topological placement, with nodes further from the AP having weaker signal and, hence, lower rate; 2) heterogeneous receiver sensitivities for different wireless cards [31]; and 3) coexistence of different, competing bands, like IEEE 802.11g with older, lower rate IEEE 802.11b stations.[2] Note that in each of the above scenarios, the interfering stations do not have to belong to the same network; it is sufficient that they interfere with one another.

SoftRepeater allows some stations (usually those near the AP) to act as repeaters for other clients (usually those that are farther away) in order to improve the overall network performance.

For example, after node A has moved in Fig. 1, node B turns on the SoftRepeater functionality and acts as a repeater for node A. Node A now sends its packets to node B instead of sending it to the AP. Since node A is close to node B, the auto-rate algorithm at node A uses higher transmission rate to send these packets. The throughput of node B can also go up because it is not contending for airtime with packets sent at a lower data rate.

The decision to turn on repeater functionality is taken by each station independently using locally available information. A station initiates the repeater functionality (i.e., becomes a SoftRepeater) by starting an ad hoc network and then quickly switching between the original infrastructure (AP-based) network and the newly formed ad hoc network using VirtualWiFi [8], [22]. The ad hoc network and the infrastructure networks can be on different channels. Other clients join the newly formed ad hoc network and use the SoftRepeater as a relay, if it improves their performance.

SoftRepeater works with ordinary, off-the-shelf wireless cards and is entirely software-based, not requiring any changes to the firmware or the hardware of the wireless cards. Most such cards cannot be turned into transparent, MAC-level ("layer 2") repeaters. Consequently, our system is implemented in the "layer 2.5" of the OSI network stack.

An alternative to the SoftRepeater approach is to deploy hardware repeaters. The main drawback of this scheme is that it requires dedicated hardware and cannot be deployed opportunistically. Furthermore, since stations do not face performance problems all the time, it is difficult to justify dedicated hardware to address this problem.

Besides solving the rate anomaly problem, the SoftRepeater system has other applications as well. For example, one could

[2]Similar problems occur when IEEE 802.11n stations have to coexist with pre-IEEE 802.11n stations.

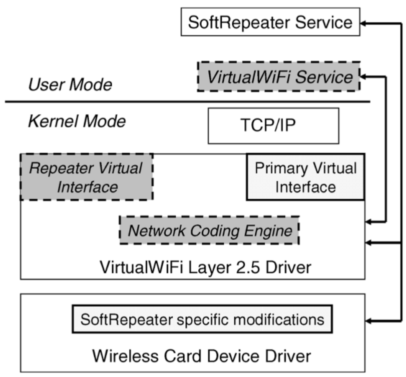


Fig. 2. The SoftRepeater architecture.

use our SoftRepeater framework to dynamically extend the range of a WLAN. A node at the edge of a WLAN could provide coverage to areas that are outside the range of the AP. However, in this paper, we focus only on the rate anomaly problem.

III. SoftRepeater Design

A. Architecture

As illustrated in Fig. 2, the architecture of the *SoftRepeater* agent that runs on each node is based on VirtualWiFi [8], which is a virtualization architecture for wireless network cards. It abstracts a wireless card into multiple virtual instances, and each virtual instance appears as an independent network interface to the user, allowing the user to connect each virtual card to a separate wireless network. VirtualWiFi provides an illusion to the user of simultaneous connectivity on all wireless networks using efficient switching and buffering techniques. It is implemented as an intermediate layer driver and a user-level service, shown as *VirtualWiFi Layer 2.5 Driver* and *VirtualWiFi Service* in Fig. 2. The mechanisms of switching and buffering are implemented in the kernel, while the logic and policies are implemented as a user-level service.

SoftRepeater uses VirtualWiFi to implement a repeater using a single wireless card. It abstracts the wireless card into two virtual instances, shown as *Repeater Virtual Interface* and *Primary Virtual Interface* in Fig. 2. The shaded components are disabled when a station is not using the repeater network. Thus, when a station is performing well, the wireless card is always connected on the primary wireless network. When a station wishes to initiate a repeater network (i.e., become a repeater), it starts the VirtualWiFi service and plugs in the details of the repeater network to the Repeater Virtual Interface. We have made several modifications to significantly reduce the switching time in comparison to the original VirtualWiFi [8] implementation; our current implementation allows a station to switch between the primary and the repeater networks in less than 40 ms.

When the SoftRepeater service initiates the repeater functionality, it buffers packets for the primary (repeater) network if the repeater (primary) is currently used so as to ensure reliable packet delivery. Note that the buffering mechanism can be implemented without modifications to the AP. The implementation details are found in [8].

The Network Coding Engine is an optional module that can further improve the performance of the repeater and the client. However, modifications to the AP are required to use the network coding engine and are detailed in Section III-F.

In summary, the SoftRepeater service works as follows. It constantly monitors the performance on the wireless network, analyzes packets to infer the existence of the rate anomaly problem, and estimates the utility of initiating the repeater network by polling various counters of the wireless card driver. The service executes a four-way handshake protocol to confirm that all participating stations have the necessary incentive to initiate SoftRepeater. If this is confirmed, SoftRepeater is activated. We explain the implementation details of SoftRepeater in the following discussion.

B. Detecting Rate Anomaly

The SoftRepeater service infers the existence of the rate anomaly problem if the wireless network interface is consistently backlogged; i.e., the station is trying to use the network at a higher rate than what is offered, and nearby nodes send approximately the same number of packets, but at a lower data rate.

The service collects information about nearby stations and their transmitted packets by setting the wireless card to promiscuous mode and logging aggregate information for each station. This aggregate information is maintained in a table, where each row corresponds to a MAC identifier of another node whose packets were overheard, plus one additional row for itself. Each row has five entries: the number of packets heard, the average size, RSSI and data rate of data packets received, and the BSSID of the associated network. This information is updated once every second and is maintained as a moving average over five update intervals.

The utilization of the wireless medium is calculated by adding the airtime consumed by all neighboring nodes, where a neighbor's airtime is calculated using the size and number of packets received from that node and the data rate at which the packets were sent. If the utilization of the medium is greater than 50%, and the SoftRepeater service observes another neighbor sending approximately the same fraction of packets but at a lower data rate, it predicts that the rate anomaly problem exists.

C. Repeater Utility Function

Once a repeater predicts the existence of the rate anomaly problem, it must next estimate its gain in throughput if SoftRepeater is invoked. We propose a notion called *Repeater Utility Function*, which captures the throughput gain of the stations when SoftRepeater is used. The Repeater Utility Function is defined based on the desired fairness requirement and is a function of the estimates of throughput among stations whose rates have yet to be determined and need to be estimated.

To motivate the challenge of invoking SoftRepeater, consider two stations A and B connected to the same AP, where the transmission rates of A and B are R_A and R_B, respectively. Suppose B infers the existence of the rate anomaly problem and considers instantiating itself as a repeater for A. Then, it must estimate the rate $R_{A,B}$ of transmissions between A and B.

The rate $R_{A,B}$ is approximated by assuming a symmetric channel and mapping the received signal strength of packets from A $(RSSI_{A,B})$ to the corresponding data rate. Each node maintains an expected data rate table, which maps an RSSI range to its expected data rate. The table is built from local measurements, as described in Section IV-B1. We emphasize that by no means do we suggest that the use of physical-layer metrics can accurately infer transmission rates, as shown in previous work [2], [36], but our approach here serves as a starting point. Given the physical-layer complexities, a more robust approach for inferring data rates is to use link-layer statistics, such that each node (assumed in promiscuous mode) periodically broadcasts probes and monitors internode loss rates, and use the loss rates to infer the best transmission rate that maximizes throughput [36]. We plan to evaluate this approach in future work.

In addition to the data rate table, each node also maintains an expected throughput table, which maps data rate to the expected throughput achievable for a given data rate. This is required as the throughput is usually smaller than the data rate due to protocol overheads and background interference. For example, even when a node sends packets at a data rate of 54 Mbps, its effective TCP/UDP throughput is of the order of 20 Mbps. We populate this table from local measurements under normal operating conditions to account for background interference and other physical-layer complexities. For instance, the expected throughput can be computed using 1/ETT, where ETT [14] is the expected transmission time of a packet over a link and is measured from link-level probing. The expected throughput $T_{A,B}$ and T_B can then be obtained from table lookups indexed by $R_{A,B}$ and R_B.

The resulting throughputs also depend on parameters α and β, where α is the fraction of time that the repeater spends on the primary network forwarding both its and its supported clients' packets to/from the AP, and β is the fraction of time that the repeater spends on the repeater network relaying its clients' packets.[3] If α and β are fixed constants and both A and B have the same throughput, then in our example, by invoking SoftRepeater, the expected throughput of B from using a repeater is given by $\alpha \times T_B/2$; the expected throughput of A from using a repeater is given by $\min((\alpha \times T_B/2), \beta \times T_{A,B})$. If the expected throughput for both A and B is greater than their current respective throughput, then there is an incentive for B to start the repeater network as well as for A to use it.

The proposed utility function does not take into account the added power consumption at the repeater. This is likely to be a concern for mobile stations. In our future work, we plan to modify the utility function to take power consumption into account.

D. Generalizing Repeater Utility Function for Different Fairness Criteria

Rather than simply have static values for α and β, SoftRepeater can implement different fairness criteria by appropriately setting α and β as a function of the known and estimated throughputs that will occur when SoftRepeater is enabled. In this subsection, we generalize our utility function based on different fairness criteria. Our analysis serves two purposes. First, we want to decide whether switching on SoftRepeater can benefit all clients *and* the repeater. Second, if we decide to switch on SoftRepeater, we want to know the fractions of time being allocated for the primary and repeater networks. Previous studies on fairness issues in wireless (e.g., [17]) or mesh routing metrics (e.g., [13] and [14]) cannot address both objectives.

Our current fairness derivations make two assumptions. First, we assume zero switching overhead, so that $\beta = 1 - \alpha$. For nonzero switching overhead (denoted by $s\%$ of airtime), we can simply set $\beta = 1 - s\% - \alpha$. Second, we assume the saturated case where there is always backlogged data available for all stations involved, implying that each station has equal long-term channel access (e.g., see [19]). This assumption conforms to file-transfer-like applications where throughput optimization is a concern. Under these assumptions, the value of α is determined by what the repeater wishes to optimize. Let T_B and $T_{A,B}$ be the achievable throughputs for data rates R_B and $R_{A,B}$, respectively (see Section III-C).

Maximizing Total Throughput: First, we consider maximizing total throughput. The total throughput T is given by

$$\begin{aligned} T &= \frac{\alpha \times T_B}{2} + \min\left(\frac{\alpha \times T_B}{2}, (1-\alpha) \times T_{A,B}\right) \\ &= \min\left(\alpha \times T_B, T_{A,B} + \frac{\alpha}{2} \times (T_B - 2T_{A,B})\right). \end{aligned}$$

Let us consider two cases. If $T_B \geq 2T_{A,B}$, then T is monotonically increasing with α. Thus, T is maximized when $\alpha = 1$. On the other hand, if $T_B < 2T_{A,B}$, then the LHS of the $\min$ is increasing with α while the RHS of the $\min$ is decreasing with α. Thus, T is maximized when $\alpha \times T_B = T_{A,B} + (\alpha/2) \times (T_B - 2T_{A,B})$, or equivalently, $\alpha = 2T_{A,B}/(T_B + 2T_{A,B})$.

However, setting $\alpha = 1$ implies that the client will be starved, an undesirable outcome always for the client node. Instead, we investigate two commonly employed fair allocation schemes in networking, namely Max-Min Fairness and Proportional Fairness [24].

Max-Min Fairness: To maximize the minimum, it suffices to equalize the throughput of the client and the repeater.

$$\frac{\alpha \times T_B}{2} = \min\left(\frac{\alpha \times T_B}{2}, (1-\alpha) \times T_{A,B}\right).$$

Thus, we have $(\alpha \times T_B/2) = (1-\alpha) \times T_{A,B}$. The optimal α is

$$\alpha = \frac{2T_{A,B}}{T_B + 2T_{A,B}}.$$

The max-min throughput is $T_{A,B} \times T_B/(T_B + 2T_{A,B})$. If the result is greater than the current throughput of A and B, SoftRepeater is invoked.

Proportional Fairness: Proportional Fairness achieves a compromise between maximizing throughput and maximizing the minimum. The philosophy of proportional fair allocation is that "expensive" flows achieve a lower quality of service without getting starved. In our scenario, the client is the expensive flow since it consumes significantly higher airtime compared to the repeater and, hence, gets lower throughput. The allocation is formally achieved by maximizing the sum of the $\log$ of the throughputs.

More formally, we want to maximize

$$\log\left(\frac{\alpha \times T_B}{2}\right) + \log\left(\min\left(\frac{\alpha \times T_B}{2}, (1-\alpha) \times T_{A,B}\right)\right).$$

We can show that either $\alpha = 0.5$ or $\alpha = 2T_{A,B}/(T_B + 2T_{A,B})$, so the optimal α is the one that maximizes the

[3]Note that $\alpha + \beta$ is less than 1 due to network switching overheads [8].

throughput. In the interest of space, the derivation is in the technical report of [8].

In our experiments, we focus mainly on Max-Min Fairness. However, if higher cumulative throughput is desired, then our framework can utilize Proportional Fairness.

Multinode Case: We now generalize the case to multiple nodes, focusing on Max-Min Fairness. Suppose that the repeater is serving one client, while there are K interfering nodes that do not participate in the repeater service but have traffic that occupies the channel. Note that these interfering nodes and the SoftRepeater nodes may be associated with the same or different APs, but they share the same contention domain. In the absence of the repeater and the client, the expected throughput of each of those K interfering nodes is $T_Z = (\sum_{i=1}^{K}(1/T_i))^{-1}$, where T_i is the achievable throughput of interfering node i. Note that all interfering nodes have the same expected throughput because they have equal long-term channel access. Now, by taking into account that the repeater (when it is on the primary network) and the client (when it is on the repeater network) need to compete for airtime with those K nodes, the throughput of the client is

$$\min\left(\frac{\alpha}{2(1/T_B+1/T_Z)}, \frac{1-\alpha}{1/T_{A,B}+1/T_Z}\right).$$

By equalizing the LHS and the RHS of the min function, we can show that the optimal α is

$$\alpha = \frac{2(1/T_B+1/T_Z)}{2/T_B+1/T_{A,B}+3/T_Z}.$$

For the special case when there is no interfering node, we can set $1/T_Z = 0$.

Thus, when the repeater is turned on, the resulting throughput is $(2/T_B+1/T_{A,B}+3/T_Z)^{-1}$. Note that without the repeater, the throughput is $(1/T_A+1/T_B+1/T_Z)^{-1}$. Thus, the presence of interfering nodes can reduce throughput, so in general, we should not turn on the repeater when there are many interfering nodes within the network.

Using similar arguments, we can extend our analysis to the case where the repeater is serving $M \geq 1$ clients. Thus, the optimal α is

$$\alpha = \frac{(M+1)(1/T_B+1/T_Z)}{(M+1)/T_B+M/T_{A,B}+(M+2)/T_Z}.$$

Multichannel Case: When the repeater switches to the repeater network using VirtualWifi, it can use a new channel different from the primary network's, thereby avoiding the contention with the interfering nodes. Thus, the throughput of the client is

$$\min\left(\frac{\alpha}{2(1/T_B+1/T_Z)}, \frac{1-\alpha}{1/T_{A,B}}\right).$$

Hence, the optimal α is

$$\alpha = \frac{2(1/T_B+1/T_Z)}{2/T_B+1/T_{A,B}+2/T_Z}.$$

E. Repeater Initiation Protocol

To determine whether invoking SoftRepeater can improve throughput for a given fairness requirement, stations can carry out the Repeater Initiation Protocol, which gathers consensus from nearby stations using a four-way handshake. The protocol steps are illustrated in Fig. 3.

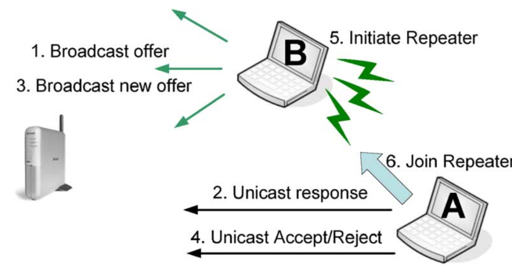


Fig. 3. Steps of the repeater initiation protocol.

1) The node with a high Repeater Utility—say B in Fig. 3—creates a message with the IP addresses of clients it intends to serve and the estimated data rate of each client. It then broadcasts this message in its IP subnet.
2) When an intended client—say A in Fig. 3—receives this message, it computes its Utility of using B as the repeater. It then unicasts this Utility and its estimated data rate to B. Note that the data rate is calculated from the signal strength of overheard packets sent by B.
3) B recalculates its Utility based on the number of the updated data rates of clients whose responses had Utility improved (assuming the clients will accept to use the repeater). It then rebroadcasts a message with a revised set of client IP addresses.
4) When A receives the second request, it recomputes its utility and sends a message to B either accepting or refusing to join the repeater network. If A accepts, it will start monitoring the medium for B's repeater network.
5) When B receives sufficient acceptances from authorized clients, it turns itself into a SoftRepeater.
6) Authorized clients then join the repeater network.

Note that messages in steps 1)–4) of the protocol are sent via the AP over the WLAN network and work only if all the clients are connected to the AP. Currently, we do not support scenarios where a client is disconnected from the network. However, we note that these scenarios can be implemented using schemes similar to the ones proposed in [1] and [9].

Each node that is part of the SoftRepeater network recalculates its utility function once every 10 s. When a station does not receive any benefit from being a repeater or being part of a SoftRepeater network, it stops the repeater network or leaves the network, respectively. In our current implementation, we do not allow clients to join a SoftRepeater network without going through the entire repeater initiation protocol.

SoftRepeater uses a simple scheme to maintain a client's existing TCP and UDP connections. A client that joins the SoftRepeater network keeps its original IP address, and the repeater sends a gratuitous ARP to the AP with the client's IP address. Therefore, when the AP receives a packet for the client, it sends it to the repeater instead. The repeater then forwards it to the client. Note that when a client decides to leave the SoftRepeater network, it sends a gratuitous ARP to the AP with its IP address.

The biggest overhead in initiating a SoftRepeater is the time to complete steps 5) and 6) of the protocol. Previous work [9] measures the time for a node to start a network and clients to join it to be less than 1 s. We observe similar delays in our experimental setup.

F. Zero-Overhead Network Coding

Hardware repeaters reduce the capacity of a wireless network by resending every received packet. These packets consume double the airtime and consequently reduce the throughput of nearby clients. If some modifications to the AP are allowed, we can limit this reduction in throughput by using network coding.

Our approach is similar to COPE [23], which is a proposal for using network coding to increase the throughput of wireless mesh networks. SoftRepeater is ideally suited to take advantage of the fundamental idea of network coding without requiring many of the additional overheads of COPE. In the SoftRepeater setting, all packets relayed by a SoftRepeater are either sent by an AP or destined to it. As a result, a repeater can encode at most two packets in each transmission, i.e., a packet sent by a client to the AP and a packet sent by the AP to that client. Opportunities for coding together more packets may exist, but are likely to be rare at best. The authors of [23] empirically show that the number of packets that can be encoded in each transmission is two to four in most cases, and the authors of [25] provide a theoretical upper bound for this number in a general topology. Therefore, we assume that a repeater only encodes at most two packets at a time. This insight allows us to propose a lightweight network coding protocol for the SoftRepeater architecture.

Consider a packet X sent by the AP to the client through the SoftRepeater and a packet Y flowing in the reverse direction. The fundamental idea of COPE is to have the SoftRepeater receive both X and Y prior to forwarding these packets, and then instead to forward $Z = X \oplus Y$. Upon receiving Z, since the AP is the sender of X and has a copy of the packet, the AP decodes Y via $Y = Z \oplus X$, and the client can similarly decode X via $X = Z \oplus Y$. In this case, the SoftRepeater has delivered both X and Y to their respective destinations while transmitting only a single packet.

Since $A \oplus A = 0$ for any constant A, XOR packets are identifiable by a value of 0 in any constant packet header field, such as the IP version field. For IPv4, this field is assigned a constant value of 4; hence, in an XOR packet, the field will have a value of 0.[4]

Once a packet Z is identified as an XOR packet, the recipient of the packet must decode the packet to retrieve the internal data. The recipient then XORs Z with packets in its send buffer, $S_1, S_2, \cdots$ to produce a set of potentially decoded packets, $D_1, D_2, \cdots$ with $D_i = S_i \oplus Z$. If the SoftRepeater generated the XOR packet Z using a particular S_i, then D_i is indeed the original packet delivered, and confirmation that this is the correct packet is obtained by verifying the checksum within the packet header of D_i. If Z was encoded using some other S_j, then D_j will (with very high probability) contain an invalid checksum.[5]

If the repeater receives packets from only one node (either the AP or the client), then no coding opportunity is available. In this case, the repeater simply forwards the packets as in the no-coding case.

Note that our technique has no additional transmission overhead: We do not add extra bytes in the packet header and do not require nodes to send additional control messages. There are several alternate compromises that are needed to implement this technique. In particular:

— **AP modification**: The AP must be explicitly configured to decode received, coded packets. Note that the use of network coding is optional. The core SoftRepeater protocol *does not* require any changes to the AP.

— **Improved transmission**: COPE increases the delivery rate of broadcasts by unicasting packets to a neighbor and having other neighbors overhear the channel (assuming nodes are in promiscuous mode). We take a step further by addressing the rate-range tradeoff of IEEE 802.11 [14], [15], such that we unicast packets at a data rate that ensures transmission to the farthest destination. Any node closer to the sender has a higher likelihood of overhearing the packets. We find that our scheme increases the rate of delivery by 40% beyond that of COPE.

— **Packet Format Limitations**: Alternately formatted packets, such as ARP packets, that do not have easily identifiable constant fields cannot utilize this zero-overhead technique because coded packets are not easily identified. We, therefore, do not apply our coding technique to these types of packets and instead continue to transmit the raw packets.

— **Buffering**: The AP and client must buffer their sent packets to use as potential candidates to decode received, coded packets. SoftRepeater uses each packet in at most one codeword and uses the received packets in order, making it easy for the AP and client to determine which packets can be flushed from any buffer they maintain solely for decoding purposes. In our current implementation, we maintain the buffered packets as a ring buffer and garbage-collect packets from these buffers during a send or a receive operation.

IV. Experimental Results

In this section, we evaluate the performance of the SoftRepeater system. We begin by demonstrating the benefits of using SoftRepeater with controlled experiments in a simple testbed. Then, we present several microbenchmarks related to the repeater initiation protocol and the benefits of using network coding as part of the SoftRepeater system.

A. Benefits of Using SoftRepeater

We demonstrate the benefits of SoftRepeater using a simple testbed that consists of two laptops, A and B (running Windows XP), and one 802.11a AP. The testbed is set up on one floor of a typical office building, as shown in Fig. 4. We fixed the location of the AP and station A and placed station B at different locations. The locations we used are labeled X, Y, T, and Z. We placed the AP at location X, station A at location Y. The location of station B varies depending on the experiment. For some of the experiments, A serves as the repeater for B, which becomes the client. The wireless network operates on channel 36 (802.11a). When the repeater functionality is used, the repeater network is also established on the same channel. The worst-case time

[4] If the network contains a mix of IPv4 and IPv6 packets, the IP version field in coded packets will be either 0 or 2, both of whose values would be unexpected in raw IP packets.

[5] Under certain conditions, XORing a packet with IP headers of consecutive packets gives the same checksum. To avoid these collisions, we randomly assign an IP ID to packets and preserve the same ID across multiple IP fragments.

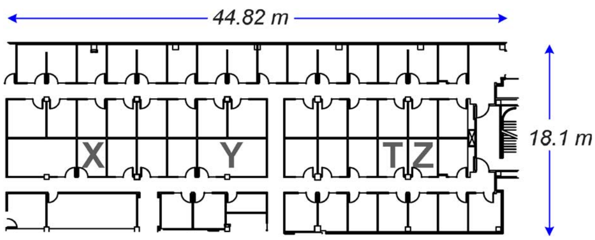


Fig. 4. Floor plan of our office.

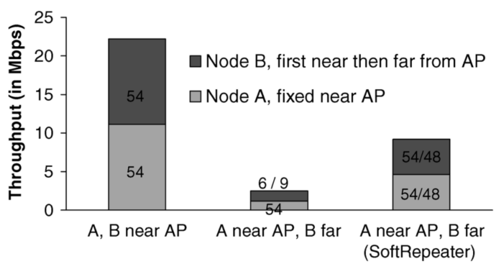


Fig. 5. Downlink UDP flows, with and without SoftRepeater.

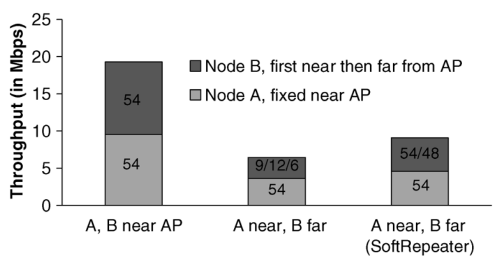


Fig. 6. Downlink TCP flows with and without a SoftRepeater.

to switch between the two networks to a network is around 50 ms. In our experiments, we use Max-Min Fairness to determine whether to switch on SoftRepeater and the fractions of time spent on the primary and repeater networks if SoftRepeater is switched on (see Section III-C). RTS/CTS exchange was turned off for all experiments.

We study the impact of SoftRepeater on both UDP and TCP flows. The UDP traffic consists of 1400-byte (payload) packets sent as fast as possible. The TCP traffic is generated using a variant of TTCP [34] for Windows. We enabled the TCP windows scaling option and use asynchronous send and receive with large send and receive buffers. We also set the receive buffer to be 1 MB. All our throughput measurements are averaged over 10 runs.

We first evaluate the SoftRepeater architecture with both uplink and downlink traffic. We then study the performance of SoftRepeater when a node helps multiple clients.

1) Downlink UDP Flows: In the first experiment, we evaluate the throughput of downlink UDP flows from the AP to the stations with and without SoftRepeater. A sender is connected to the AP via wired Ethernet. The sender sends UDP flows to both A and B. We plot the throughput received by both stations at different locations in Fig. 5. The values inside the bars denote the data rate of packets sent to each station.

Initially, both A and B are at location Y (see Fig. 4), which is a conference room located three offices away from the AP's location, X. Both stations have a good connection to the AP and get approximately the same throughput. We then move station B to location Z, which reduces its connection quality to the AP. The AP can only send packets at 6 or 9 Mbps to B, and hence, the throughput of the flow to B drops. Furthermore, the throughput of the flow to A also drops significantly due to rate anomaly. However, the throughput of both A and B goes up if A turns itself into a repeater for B. The results are shown in Fig. 5. B gets better throughput because it receives packets at a higher data rate from A, and A gets better throughput since it does not suffer from rate anomaly due to low data rate packets. The overall network throughput nearly triples when SoftRepeater is used.

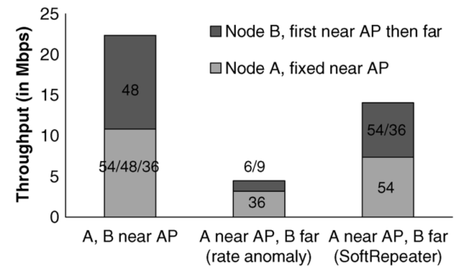


Fig. 7. Uplink UDP flows with and without SoftRepeater.

2) Downlink TCP Flows: We set up downlink TCP flows from a wired host to the two wireless stations A and B. The results are shown in Fig. 6. When both A and B are at location Y of Fig. 4, both of them get a throughput of approximately 9 Mbps. We then move B to location T.[6] With this, the throughputs of A and B drop significantly. When A turns into a repeater, it increases the TCP throughput of both itself and B, and the overall network throughput goes up by 50%.[7]

3) Uplink Flows: Although the predominant traffic in wireless networks is downlink flows, there is usually a small fraction of uplink flows as well. We now show that SoftRepeater also provides throughput improvement for uplink flows. We performed experiments for TCP as well as UDP flows, and the performance in both of these scenarios was similar. We only present the UDP results here.

We initiated UDP flows from stations A and B to a host on the wired network. The results are shown in Fig. 7. When both stations are at location Y, they get approximately the same throughput. However, we see fluctuating data rates due to collisions and auto-rate at the stations. When station B is moved to location Z, the throughput of both A and B drops due to the impact of rate anomaly. We see that the throughput of A is slightly higher than that of B. The reason is that station A is closer to the AP, and its packets can sometimes be decoded by the AP even when they collide with packets sent by station B (capture effect). When station A functions as a repeater, the throughputs of both stations increase and the overall network throughput is doubled.

4) Performance With Multiple Clients: In another experiment, we studied the performance of SoftRepeater when it repeated traffic from two clients instead of one. We first placed three stations A, B, and C at location Y of Fig. 4, and the AP was fixed at location X. We started downlink UDP flows from the AP

[6] We could not get a stable connection from B to the wired host when B is at location Z.

[7] In addition, we note that when A is used as a repeater, B was able to establish a stable TCP connection from location Z as well.

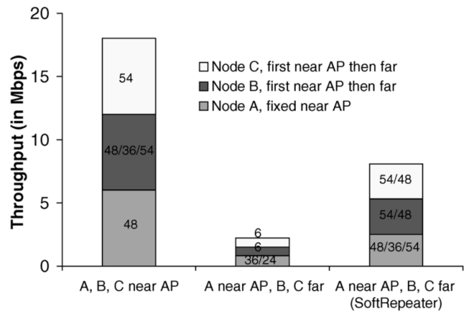


Fig. 8. Downlink UDP flows when SoftRepeater serves two clients.

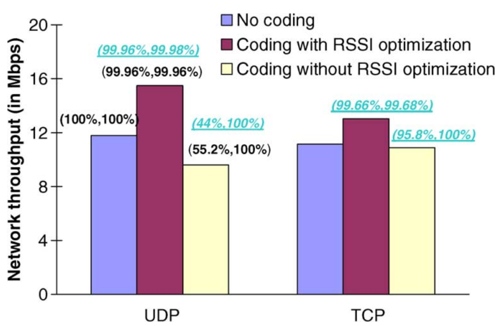


Fig. 9. Throughput improvement due to network coding and impact of RSSI optimization.

to all the stations and plot the throughput of each of the flows in Fig. 8. We then moved B and C to location Z. We saw a significant decrease in throughput due to rate anomaly. However, the throughput of all the stations increased when A (which is at location Y) acted as a repeater for both B and C at location Z. Using A as a repeater nearly triples the network throughput.

5) Further Improvement With Network Coding: As discussed in Section III-F, we expect that network coding further improves the performance of SoftRepeater. To quantify the improvement, we carry out experiments with bidirectional TCP and UDP traffic. As before, the AP is at location X, station A is at location Y, and station B is either at location T (for TCP experiments) or at location Z (for UDP experiments). We initiate bidirectional TCP or UDP flows between B and the AP; i.e., A is the encoder, and the AP and B are the decoders.

As discussed in Section III-F, we propose to address the rate-range tradeoff by unicasting coded packets from repeater A using the data rate that ensures packets are reachable at the farthest node (i.e., either the AP or client B). The farthest node has the lowest RSSI, and the corresponding data rate is obtained from the RSSI measurements stored in the expected data rate table (see Section III-C). We call this heuristic *RSSI optimization*. We compare our approach with a scheme that does not use network coding and a scheme that uses network coding but does not use RSSI optimization [23] (i.e., it unicasts packets using the data rate for transmissions to the closer node). In Fig. 9, the underlined numbers denote the link-layer delivery ratio of packets sent from A to B, and packets from A to the AP, respectively. The nonunderlined numbers denote the end-to-end delivery ratio of the flow between B and the AP and between the AP and B, respectively.

Fig. 9 shows that network coding scheme significantly improves the network throughput and that RSSI optimization is critical to achieve this improvement. Without RSSI optimization, the coded packets are unicast at the data rate for the closer node. Therefore, the receiver that is farther away from the repeater is unable to decode the packet. As a result, we see significantly uneven link layer delivery ratios (100% and 44% for UDP, and 100% and 95.8% for TCP) for the two receivers when RSSI optimization is not used. A drop in link-layer delivery ratio for one receiver significantly reduces network throughput. In fact, for UDP experiments, network coding without RSSI optimization *reduced* the network throughput by 20%, as opposed to a 30% increase in the case of network coding with RSSI optimization. For TCP traffic, we see that network coding offers no improvement in performance without RSSI optimization. With RSSI optimization, network coding improves the performance by 15%.

B. Protocol Validation

SoftRepeater requires nodes to dynamically detect rate anomaly and initiate SoftRepeater on the fly. In this section, we validate the correctness of our system. We first show the feasibility of mapping between RSSI and the data rate. We then demonstrate the correctness of the Repeater Utility Function and the Repeater Initiation Protocol using a carefully controlled, simple traffic scenario. Finally, we validate our Repeater Initiation Protocol in five other scenarios.

1) Signal Strength versus Data Rate: The Repeater Utility function, described in Section III-C, requires a mapping from RSSI to the expected data rate. We now show that this mapping is feasible. We set up a sender at a fixed location on our floor and moved a receiver to 267 different locations. The sender transmitted a stream of packets to the receiver using its default auto rate algorithm. At each location, we measured the RSSI of the received packets and the data rates at which they were sent. The results are shown in Fig. 10.[8] Note that a WiFi sender determines the transmission rate of the packets based on a variety of factors such as loss rate and the signal strength of packets (such as ACKs) that it has received from the receiver. Yet, we see that there is a reasonable correlation between the signal strength with which each packet was *received* and the data rate it was *sent at*. In other words, the wireless channel is somewhat (but not completely) symmetric. We use these measurements to build a table that predicts the most likely data rate given an RSSI value. Note that these numbers do not have to be exact. A repeater network is started only when the expected throughput (calculated from the likely data rate) is significantly higher than the current throughput. We note that these measurements are supplementary to the ones presented in [2], which showed the correlation between loss rate and RSSI at a fixed data rate. As described in Section III-C, we can also use a more robust approach (e.g., [36]) to infer data rates.

2) Simple Traffic Scenario: We now demonstrate that the repeater functionality is initiated only when it benefits both the repeater and the client. As before, we place the AP at location X, station A at location Y, and station B at location Z. We know from previous experiments that rate anomaly will exist in this situation. However, if B is not sending or receiving significant traffic, there is no need for A to offer the SoftRepeater functionality.

[8]Note that multiple locations may give the same RSSI and data rate values.

TABLE I
RESULTS OF THE REPEATER INITIATION PROTOCOL FOR FIVE SCENARIOS. STATION A IS THE POTENTIAL REPEATER. PACKET RATIO IS THE RATIO OF THE NUMBER OF PACKETS SENT TO THE HIGH-RATE STATION DIVIDED BY THE NUMBER OF PACKETS SENT TO THE LOW-RATE STATION, AND RATE RATIO IS THE RATIO OF THE DATA RATE USED BY THE HIGH-RATE STATION TO THE DATA RATE USED BY THE LOW-RATE STATION. OUR PROTOCOL TURNS ON SOFTREPEATER AT STATION A FOR THE THIRD AND FIFTH SCENARIOS

Scenario	Station A's observations				Throughput at A		Throughput at B	
	Busy Airtime	Packet Ratio	Rate Ratio	RSSI from B	Measured without Repeater (Mbps)	Measured with Repeater (Mbps)	Measured without Repeater (Mbps)	Measured with Repeater (Mbps)
Healthy Network	44%	1	1	79	12.0		11.9	
No Congestion	12%			67	1.2		0.6	
Rate Anomaly	87%	0.48	9	35	2.4	3.24	1.9	3.22
No SoftRepeater	88%	0.6	9	12	3.0		3.1	
Complex Setting	85%	0.28	9	29	0.6	0.6	0.8	3.88 (4 for C)

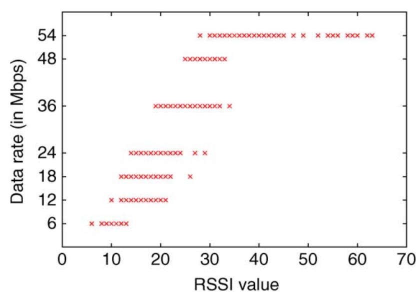


Fig. 10. Correlation between RSSI and data rate.

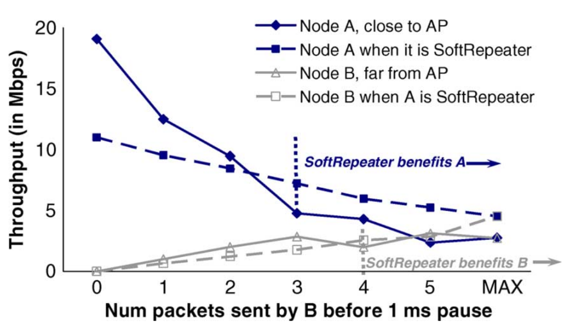


Fig. 11. AP sends full-blast traffic to A and sends bursts of packets to B with a pause for 1 ms. MAX corresponds to back to back UDP packets.

To illustrate this, we start a full-blast UDP transfer from a wired host connected to the AP to A. Another flow is started from the wired host to B. We send packets to B in bursts, with a 1-ms pause between bursts.

Fig. 11 plots the throughput of flows to A and B upon changing the number of packets that are sent to B in each burst. MAX in the figure corresponds to back to back UDP packets to B with no pause. Each UDP packet in our experiments is 1400 bytes long.

We first carried out the experiment when A never becomes a SoftRepeater. The throughputs of A and B are shown by solid lines in Fig. 11. Next, we repeated the experiment when we forcibly turn A into a repeater and B into a client. The throughput of the two stations with SoftRepeater turned on is shown by dashed lines in Fig. 11.

We see that starting a repeater at A will hurt its performance if B is not receiving enough traffic. Turning on repeater functionality benefits A only when the throughput it gets with the repeater functionality turned on is higher than the throughput it gets with the functionality turned off. This happens when B starts receiving more than three packets in each burst. Similarly, it is not in B's interest to join a repeater network until it starts to receive more than four packets per burst.

Finally, we repeated the experiment once again and allowed A to become a repeater when it saw benefits. Our system correctly detected that A should become a repeater only when B was sending more than three packets in a burst. For this case, A calculated its expected throughput would be 5.5 Mbps if it turned on the repeater functionality. Furthermore, the repeater network was started (i.e., B joined it) only when B started receiving more than four packets in a burst. This experiment demonstrates the correctness of our rate anomaly detection routine and the calculation of the utility function. In the next section, we consider more complex traffic scenarios to validate our protocol.

3) Other Traffic Scenarios: We now validate the Repeater Initiation Protocol under five different scenarios. We fix the AP at location X, and place stations A and B at locations X, Y, and Z for different experiments. In all these scenarios, we initiate UDP traffic from a host that is connected to the AP over Ethernet.

Here, we assume that a station starts the repeater network only if the following conditions are satisfied. 1) The network is heavy-loaded when the percentage of busy airtime consumed by data packets is over a preset threshold (50%). 2) A rate anomaly scenario in which the ratio of packets sent to different stations (i.e., Packet Ratio denoted in Table I) is disproportionate to their corresponding data rate ratio. We use 1/2 as the threshold. 3) The potential repeater observes a strong signal strength (≥ 26) from the client. A signal strength of 26 corresponds to an expected data rate of 36 Mbps from our measurements. 4) For the repeater, the expected throughput from using the repeater network is higher than its current throughput. 5) For the client, the expected throughput from using the repeater network is higher than its current throughput.

We now describe each scenario that we tested against. The results are summarized in in Table I.

— **Healthy Network:** We place the AP and stations A and B at location X. We send full-blast UDP traffic to both A and B. Traffic in both connections is sent at 54 Mbps. Both A and B receive a high throughput of around 12 Mbps. There is no rate anomaly in this scenario, and so the repeater network is not started.

— **No Congestion:** We place the AP at location X, station A at location Y, and station B at location Z. We send UDP traffic at 1.2 Mbps to A and 0.6 Mbps to B. In this scenario, A observes that the network is busy transmitting data packets 12% of the time, which is less than the 50% threshold. Therefore, the repeater network is not started.

— **Rate Anomaly:** We place the AP at location X, station A at location Y, and station B at location Z. We send full-blast UDP traffic to both A and B. A receives packets at 54 Mbps, while B receives packets at 6 Mbps. The throughputs of both A and B are approximately 2 Mbps. This is a typical rate anomaly scenario, and all conditions for initiating a SoftRepeater are satisfied: The percentage of busy airtime (87%) >50%, and the utility function indicates that there is value in starting the repeater functionality. Note thatA indeed started a repeater network, and A's and B's throughput increased to 3.24 and 3.22 Mbps, respectively.

— **No Available SoftRepeater:** We place the AP and A at location X, and B at location Z. We send full-blast UDP traffic to both A and B. The AP uses transmission rate of 6 Mbps while sending to B and 54 Mbps when sending to A. The UDP throughput to both A and B is about 3 Mbps. Station A recognizes that this is a rate anomaly scenario. However, the observed RSSI from B is 12, which is less than 26. Therefore, A is not in a good position to help B, and the repeater network is not started. This happens because A is too close to the AP (rather than being midway between the AP and B, as in the previous scenario), and B is likely to get the same poor performance from talking to A as that it is getting from talking to the AP. To verify this, we manually started the repeater network. With the repeater switched on, B's throughput dropped from 3.1 to 2.1 Mbps.

— **Complex Setting:** We introduce another station C. We place both C and AP at location X, A at location Y, and B at location Z. We send full-blast UDP traffic to both B and C and a small amount of traffic (0.6 Mbps) to A. AP sends packets to B at 6 Mbps, to C at 54 Mbps, and both B and C achieve 0.8-Mbps throughput. A's moderate bandwidth requirement is satisfied. However, A is the only one that is in a good location to help B. A observes strong RSSI (29) from B; and the utility function indicates that the repeater should be started. After the repeater network is started, A's throughput stays at 0.6 Mbps (since it is not bottlenecked), while B's throughput improves to 3.88 Mbps and C's throughput improves to 4.0 Mbps. In summary, after A becomes a repeater, it significantly improves the throughput of other clients around it.

C. Summary

The experiments in this section show that using SoftRepeater increases the throughput of the repeater as well as of the client(s) being helped. This increases the overall throughput of the system. We have shown that SoftRepeater works in many different traffic scenarios, with multiple clients, with both uplink and downlink traffic, and benefits TCP as well as UDP flows. We also showed that using network coding with SoftRepeater further improves the overall throughput. Certain aspects of the SoftRepeater protocol are difficult to evaluate using a testbed. For example, we cannot easily change the switching overhead in our implementation. For such cases, we turn to simulations. Simulations also allow us to evaluate the protocol on larger networks. To this end, we have implemented the SoftRepeater protocol using Qualnet [30]. We refer readers to our conference version [8] for the simulation results.

V. Discussion

SoftRepeater works best when all stations cooperate and follow the Repeater Initiation Protocol. Stations that refuse to cooperate reduce the overall possible gain.[9] Stations may simply be inherently uncooperative or on another (competing) network. Let us explore these factors in three scenarios.

— **Enterprise:** We expect SoftRepeater to be the most useful in enterprise wireless networks. Trust, and therefore cooperation, among employees can be enforced by the IT administrators. Since enterprise networks usually span a reasonably large area, most stations in a region will be connected to the same wireless network.

— **Home:** We expect people in the same residence to trust one another and therefore cooperate. However, if there is a poorly performing station on a neighboring wireless network on the same frequency channel, the SoftRepeater approach may not provide significant gains. Obviously, the right solution for this problem is for the AP to switch to another channel.

— **Hotspots:** We expect people to be wary of sending packets through an untrusted user's laptop. However, a number of hotspots still have an open wireless network without any layer 2 security primitives. Since users are not deterred by potential sniffing, they might also be willing to use SoftRepeater.

SoftRepeater can also dynamically extend the range of a wireless network, which would be useful in large homes where people currently use more costly commercially available hardware repeaters.

VI. Related Work

In [19], the rate anomaly problem in 802.11b WLANs was first exposed and analyzed. Our experimental results confirm this problem for 802.11a WLANs.

Various solutions to the rate anomaly problem suggest changing the MAC to be "time-fair" rather than the current "packet-fair" scheme that is used in practice [20], [29] and, therefore, require a new MAC and would not interoperate with the *de facto* standard 802.11 deployed in conventional LANs. Furthermore, unlike SoftRepeater, the above ideas have not been demonstrated on top of real systems.

In contrast to SoftRepeater, other practical solutions, such as [16] and [32] require changes to the AP. Another drawback of prior work is that they further degrade the performance of the low-rate stations such that the incentive to affect the change is not global among stations, as is the case for SoftRepeater.

Multihop extensions to WLANs, such as those proposed in [26] and [28], have demonstrated in simulation that they too can alleviate the rate anomaly problem. However, because they require substantial modifications to the MAC layer, they have not been tested in practice. CoopMAC [27] and CRS [18], while having been implemented, only support the *ad hoc* mode. Also, SoftRepeater can use multiple channels, while CoopMAC and CRS cannot.

[9] The Repeater Initiation Protocol will not start the SoftRepeater if there is no throughput gain.

Various mesh routing schemes, [6], [7], [13], [14] focus on increasing throughput in an *ad hoc* setting. In particular, WCETT [14] aims to minimize the transmission times of a mix of high-rate and low-rate senders. To account for background interference, we can use a similar idea of WCETT to determine the link throughput via link-level probing (see Section III-C). Although the mesh routing schemes consider more complex cases with multiple hops, using three or more hops to mitigate rate anomaly only brings marginal benefits [26]. On the other hand, unlike SoftRepeater, these schemes cannot address the fairness issues involving more than one station.

Commercially available hardware repeaters [11], [12], [35] blindly repeat everything they overhear over the air without considering the effects. Consequently, they double the traffic transmitted over the air, and each new repeater reduces the network capacity by half. They are mainly useful as range extenders instead of addressing the rate anomaly problem.

Finally, we compare our network coding scheme against previous proposals. Initial work, such as [3], [10], [21], and [37], focus on multicast traffic and require prior knowledge of topology. The scheme proposed and implemented in [23], which utilizes network coding in a wireless unicast setting, applies naturally to the SoftRepeater setting.

VII. Conclusion

We have presented a new approach, called SoftRepeater, to alleviate the rate anomaly problem in IEEE 802.11 WLANs. As part of the SoftRepeater design, we also propose new algorithms to determine the presence of rate anomaly, a mechanism for dynamically starting a repeater network without breaking existing connections, and a new low-overhead network coding approach. Our scheme does not require any changes to the 802.11 MAC and works over commercially available wireless cards. We have implemented SoftRepeater on Windows XP and our evaluations show that SoftRepeater can improve the total network throughput by up to 200% in some of the scenarios that we explored.

We are exploring ways to improve the performance of SoftRepeater. First, to reduce the switching overhead of VirtualWiFi, we are exploring a hardware implementation of VirtualWiFi with Atheros chipsets, whose newer versions allow simultaneous associations to multiple BSSIDs [4]. Second, we are enhancing the Repeater Utility Function to address more complex traffic models, power consumption, and mobility of the repeater and the clients. Third, we are exploring an alternative architecture in which some nodes have multiple WiFi radios and are, therefore, more likely candidates to become a SoftRepeater. Finally, we are developing a protocol for repeaters to coordinate their actions to improve their cumulative performance.

References

[1] A. Adya, P. Bahl, R. Chandra, and L. Qiu, "Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks," in *Proc. MobiCom*, 2004, pp. 30–44.

[2] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proc. SIGCOMM*, 2004, pp. 121–132.

[3] R. Ahlswede, N. Cai, R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[4] *"Atheros Wireless LAN,"* Atheros [Online]. Available: http://www.atheros.com/

[5] Bahl, R. Chandra, P. P. C. Lee, V. Misra, J. Padhye, D. Rubenstein, and Y. Yu, "Opportunistic use of client repeaters to improve performance of WLANs," in *Proc. ACM CoNEXT*, Dec. 2008, Article No. 29.

[6] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," in *Proc. ACM SIGCOMM*, Aug. 2005, pp. 133–144.

[7] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. ACM SIGCOMM*, Aug. 2007, pp. 169–180.

[8] R. Chandra, V. Bahl, and P. Bahl, "MultiNet: Connecting to multiple IEEE 802.11 networks using a single wireless card," in *Proc. INFOCOM*, 2004, pp. 882–893.

[9] R. Chandra, V. Padmanabhan, and M. Zhang, "WiFiProfiler: Cooperative fault diagnosis in WLANs," in *Proc. MobiSys*, 2006, pp. 205–219.

[10] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Allerton Conf. Commun., Control and Comput.*, 2003, pp. 40–49.

[11] *"Aironet 1200 Series Access Point,"* Cisco [Online]. Available: http://www.cisco.com/en/US/products/hw/wireless/ps430/index.html

[12] *"Air Pro Wireless Access Point,"* D-Link [Online]. Available: http://www.dlink.com/

[13] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. MobiCom*, 2003, pp. 134–146.

[14] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh network," in *Proc. MobiCom*, 2004, pp. 114–128.

[15] R. P. Draves, J. Padhye, and B. D. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *Proc. SIGCOMM*, 2004, pp. 133–144.

[16] J. Dunn, M. Neufeld, A. Sheth, D. Grunwald, and J. K. Bennett, "A practical cross-layer mechanism for fairness in 802.11 networks," *MONET*, vol. 11, no. 1, pp. 37–45, 2006.

[17] C. T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks," in *Proc. ACM SenSys*, Nov. 2004, pp. 148–161.

[18] L. Guo, X. Ding, H. Wang, Q. Li, S. Chen, and X. Zhang, "Cooperative relay service in a wireless LAN," *IEE J. Sel. Areas Commun.*, vol. 25, no. 2, pp. 355–368, Feb. 2007.

[19] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proc. INFOCOM*, 2003, pp. 836–843.

[20] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless LANs," in *Proc. SIGCOMM*, 2005, pp. 121–132.

[21] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, K. J. S. Egner, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1973–1982, Jun. 2003.

[22] S. Kandula, K. Lin, T. Badirkhanli, and D. Katabi, "FatVAP: Aggregating AP backhaul bandwidth," in *Proc. NSDI*, 2008, pp. 89–104.

[23] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proc. SIGCOMM*, 2006, pp. 243–254.

[24] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, 1998.

[25] J. Le, J. C. S. Lui, and D. M. Chiu, "How many packets can we encode?—An analysis of practical wireless network coding," in *Proc. IEEE INFOCOM*, 2008, pp. 371–375.

[26] S. Lee, S. Banerjee, and B. Bhattacharjee, "The case for a multi-hop wireless local area network," in *Proc. INFOCOM*, 2004, pp. 894–905.

[27] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. Panwar, "CoopMAC: A cooperative MAC protocol for wireless LANs," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 2, pp. 340–354, Feb. 2007.

[28] S. Narayanan, P. Liu, and S. Panwar, "On the advantages of multi-hop extensions to the IEEE 802.11 infrastructure mode," in *Proc. WCNC*, 2005, pp. 132–138.

[29] B. Sagdehi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *Proc. MobiCom*, 2002, pp. 24–35.

[30] *"Qualnet Network Simulator,"* Scalable Network Technologies.

[31] A. Sheth, C. Doerr, D. Grunwald, R. Han, and D. Sicker, "MOJO: A distributed physical layer anomaly detection system for 802.11 WLANs," in *Proc. MobiSys*, 2006, pp. 191–204.

[32] G. Tan and J. Guttag, "Time-based fairness improves performance in multi-rate WLANs," 2004.

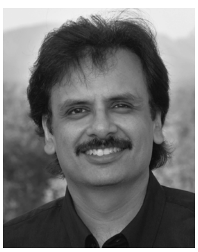
[33] G. Tan and J. Guttag, "The 802.11 MAC protocol leads to inefficient equilibria," in *Proc. INFOCOM*, 2005, pp. 1–11.

[34] TTCP, "The story of the TTCP program," [Online]. Available: http://ftp.arl.mil/ mike/ttcp.html

[35] WiDeFi, Two radio repeaters, [Online]. Available: http://www.widefi.com/

[36] S. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proc. ACM MobiCom*, Sep. 2006, pp. 146–157.

[37] Y. Wu, P. A. Chou, and S. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," MSR-TR-2004-78, 2004.



Paramvir (Victor) Bahl (F'08) received the Ph.D. degree from the University of Massachusetts, Amherst, in 1997.

He is currently a Principal Researcher and Founding Manager of the Networking Research Group in Microsoft Research, Redmond, WA. He is responsible for directing research activities that push the state of the art in computer networking.

Dr. Bahl has served on the board of six IEEE and Association for Computing Machinery (ACM) journals and on several U.S. National Science Foundation and National Research Council panels. He received Digital's prestigious Doctoral Engineering Award in 1995 and SIGMOBILE's Distinguished Service Award in 2001. In November 2003, he became an ACM Fellow for "contributions to wireless communication systems, and leadership in the mobile computing and communications community."

Ranveer Chandra completed undergraduate studies in computer science from Indian Institute of Technology, Kharagpur, India, and received the Ph.D. in computer science from Cornell University, Ithaca, NY, in 2005.

is a Researcher in the Networking Research Group at Microsoft Research, Redmond, WA. His research is focused on systems issues in computer networks, and he has published more than 25 research papers and filed over 30 patents. Currently he is working on four different projects under this broad umbrella: white space networking, energy-conserving PC architecture, mobile systems, and network management.

Patrick P. C. Lee received the B.Eng. degree (first-class honors) in information engineering and the M.Phil. degree in computer science and engineering from the Chinese University of Hong Kong in 2001 and 2003, respectively, and the Ph.D. degree in computer science from Columbia University, NY, in 2008.

He is now an Assistant Professor in the Department of Computer Science and Engineering at the Chinese University of Hong Kong. His research interests are in network robustness and security.

Vishal Misra received the B.Tech. degree from Indian Institute of Technology, Bombay, India, and the Ph.D. degree from the University of Massachusetts, Amherst.

He is an Associate Professor in Computer Science and Electrical Engineering at Columbia University. His research emphasis is on mathematical modeling of computer systems, bridging the gap between practice and analysis. His recent work includes the areas of overlay security, congestion control mechanisms, wireless modeling and efficient scheduling policies.

He has received an NSF CAREER Award, a DoE CAREER Award, and an IBM Faculty Award. He has served as the Guest Editor for the *Journal of Performance Evaluation* and Co-Chair of the Opensig 2003 and NYMAN 2002 workshops. He has participated as member of program committees for conferences such as IEEE INFOCOM, ACM SIGMETRICS, ACM SiGCOMM, IFIP Performance, and IEEE ICNP.

Jitendra Padhye received the B.E. degree in computer engineering from Victoria Jubilee Technical Institute, Mumbai, India, in 1992; the M.S. degree in computer science from Vanderbilt University, Nashville, TN, in 1995; and the Ph.D. degree from the Computer Science Department, University of Massachusetts, Amherst, in 2000.

After receiving the Ph.D. degree, he then spent two years working at ACIRI, now called ICIR, in Berkeley, CA. He has been at Microsoft Research, Redmond, WA, since April 2002. His research interests are in the area of Internet technologies and applications—specifically, Internet congestion control.

Dan Rubenstein (M'03) received the B.S. degree in mathematics from the Massachusetts Institute of Technology, Cambridge, the M.A. degree in math from the University of California, Los Angeles, and the Ph.D. degree in computer science from the University of Massachusetts, Amherst.

He is now an Associate Professor of electrical engineering and computer science at Columbia University, NY. He is generally interested in the design and performance of networked systems and also the security and integrity of such systems as well as the general design of distributed communication algorithms.

Dr. Rubinstein has received a U.S. National Science Foundation CAREER Award, an IBM Faculty Award, an ACM CoNEXT Best Paper Award, an IEEE ICNP Best Paper Award, and (as a student) an ACM SIGMETRICS Best Student Paper Award.

Yan Yu received the Ph.D. degree in computer science from the University of California, Los Angeles, in January 2005.

She is currently an engineer at Google. Her research interests include distributed systems and wireless sensor networks.