Opportunistic Use of Client Repeaters to Improve Performance of WLANs *

Victor Bahl[1], Ranveer Chandra[1], Patrick P. C. Lee[2], Vishal Misra[2],
Jitendra Padhye[1], Dan Rubenstein[2], Yan Yu[3]
[1]Microsoft Research, [2]Columbia University, [3]Google Inc.

ABSTRACT

Currently deployed IEEE 802.11 WLANs (Wi-Fi networks) share access point (AP) bandwidth on a per-packet basis. However, the 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 high rates.

We propose a practical, deployable system, called *SoftRepeater*, in which stations cooperatively address the rate anomaly problem. Specifically, higher-rate Wi-Fi stations opportunistically transform themselves into repeaters for stations with low data-rates when transmitting to/from the AP. The key challenge is to determine when it is beneficial to enable the repeater functionality. In this paper, 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.

We evaluate our system using simulation and testbed implementation, and find that SoftRepeater can improve cumulative throughput by up to 200%.

1. 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, APs are generally sparely deployed. When heavily used, such networks suffer from the well-known *rate-anomaly problem* [13]. 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.

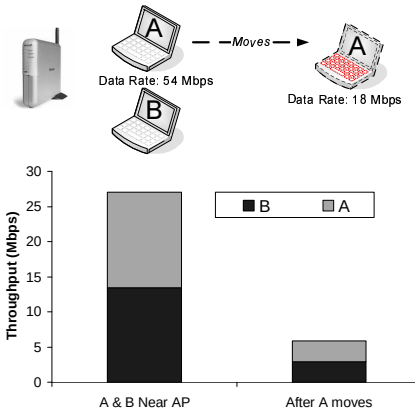

Figure 1: The Rate Anomaly problem. B's throughput drops to 25% even though it never moved.

This problem is demonstrated experimentally in Figure 1. A testbed with two laptops (stations), A and B, are associated to a single Access Point (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 5, address the rate anomaly problem. How-

*P. Lee, V. Misra, and D. Rubenstein were supported in part by the National Science Foundation under Grants CNS-0626795 and CCR-0615126.

ACM CoNEXT 2008, December 10-12, 2008, Madrid, SPAIN

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

ever, they have the following limitations: requiring dedicated hardware repeaters (e.g., [7, 8, 26]), making changes to the MAC layer (e.g., [17, 18, 19]), or constructing multi-hop networks from existing stations in ad hoc mode (e.g., [10]). 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*, that enables stations (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 the client 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. Such 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 [5]. 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 [5, 15] 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.
- 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 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 2 overviews the SoftRepeater architecture, and Section 3 discusses its implementation details. Section 4 presents evaluation results. Section 5 reviews related work, and Section 6 concludes the paper.

2. THE SOFTREPEATER APPROACH

Our opportunistic repeater framework, SoftRepeater, alleviates the rate anomaly problem [13, 24], 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 [22], and (3) co-existence 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 further away), in order to improve the overall network performance.

For example, after node A has moved in Figure 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 will 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 [5, 15]. The ad hoc network and the infrastructure networks can be on different channels. Other

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

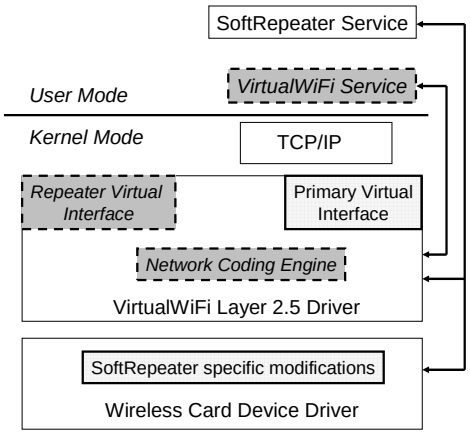

Figure 2: The SoftRepeater Architecture.

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. Further, 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 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.

3. ARCHITECTURE

As illustrated in Figure 2, the architecture of the *SoftRepeater* agent that runs on each node is based on VirtualWiFi [5], 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 Figure 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 Figure 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 [5] implementation; our current implementation allows a station to switch between the primary and the repeater networks in less than 40ms.

The SoftRepeater service constantly monitors the performance on the wireless network, estimating the utility of initiating the repeater network by polling various counters of the wireless card driver. The service also communicates with other nearby stations before finalizing the decision to initiate or join a repeater network.

In addition, 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 [5].

The Network Coding Engine is an optional module that can further improve the performance of the repeater and the client, with a tradeoff of minimal modifications to the AP. We conduct testbed experiments on using the Network Coding Engine and demonstrate the improvement of throughput as compared to without coding. Due to space constraints, we refer readers to [4] for details.

3.1 Initiating a SoftRepeater

The SoftRepeater service monitors the performance on a station's wireless interface, analyzes packets to infer the existence of the rate anomaly problem, and 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.

3.1.1 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 5 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.

3.1.2 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. This gain depends on several factors, including the desired fairness criteria, and estimates of throughput between 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 4.2.1. 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, 27], 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 inter-node loss rates, and use the loss rates to infer the best transmission rate that maximizes throughput [27]. 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 [10] 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: $\frac{\alpha * T_B}{2}$; the expected throughput of A from using a repeater is given by: $\min(\frac{\alpha * T_B}{2}, \beta * T_{A,B})$. 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.

3.1.3 Fairness

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., [12]) or mesh routing metrics (e.g., [9, 10]) cannot address both objectives.

Our current fairness derivations make two assumptions. First, we assume zero switching overhead, so that $\beta = 1 - \alpha$. For non-zero 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 [13]). 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 3.1.2).

Maximizing total throughput. First we consider maximiz-

[3]Note that $\alpha + \beta$ is less than 1 due to network switching overheads [5].

ing total throughput. The total throughput T is given by:

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

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 $\alpha * T_B = T_{A,B} + \frac{\alpha}{2} * (T_B - 2T_{A,B})$, or equivalently, $\alpha = 2\,T_{A,B}/(T_B + 2\,T_{A,B})$.

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 [16].

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

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

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

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

The max-min throughput is $T_{A,B} * T_B/(T_B + 2T_{A,B})$. 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 it gets lower throughput. The allocation is formally achieved by maximizing the sum of the $\log$ of the throughputs.

More formally, we want to maximize:

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

Due to space constraints, we refer readers to [4] for detailed derivation of the optimal α. However, we 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 throughput.

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

Multi-node 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 $T_Z = (\sum_{i=1}^{K} \frac{1}{T_i})^{-1}$, 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 with the repeater turned on, so in general, we should not turn on the repeater when there are many interfering nodes within the network. We verify this observation through simulation in Section 4.4.

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

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

We also analyze the cases for multiple channels and multiple wireless interfaces. We refer readers to [4] for details.

3.1.4 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 Figure 3.

1. The node with a high Repeater Utility, say B in Figure 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 Figure 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.

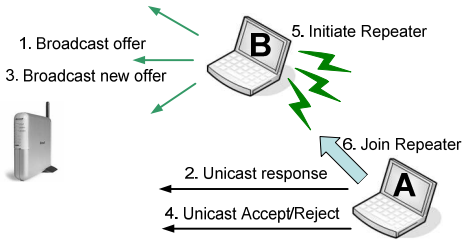

Figure 3: Steps of the Repeater Initiation Protocol.

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 through 4 of the above protocol are sent via the AP over the WLAN network, and works 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 [6, 1].

Each node that is part of the SoftRepeater network recalculates its utility function once every 10 seconds. 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 above protocol. Previous work [6] measures the time for a node to start a network, and clients to join it, to be less than 1 second. We observe similar delays in our experimental setup.

3.2 Security

SoftRepeater is designed primarily for trusted environments, such as at-home and enterprise networks. As such, our current version does not explicitly address security loopholes in 802.11. Nonetheless, we touch briefly upon both the pros and cons of using SoftRepeater in an untrusted environment. We discuss security from three perspectives: privacy, greedy stations, and malicious stations.

Privacy: SoftRepeater does not enhance network privacy. When no MAC-layer security or global-key MAC-layer security such as WEP is employed, SoftRepeater functions seamlessly, offering security similar to a SoftRepeater-free network. Pairwise-key MAC level encryption (such as with WPA) is problematic for SoftRepeater without MAC-layer modification since the repeater would require access to the key utilized by the AP and client to identify their packets for repeating. End-to-end encryption methods such as IPSec are necessary to enforce privacy in the SoftRepeater setting.

Greedy Stations: Bandwidth-greedy stations may try to game the repeater infrastructure. For instance, a node may lie about having a high transmission rate to the AP, or may choose to drop packets from stations for which it has promised to function as a repeater. To address the first issue, we note that a greedy repeater attempting to obtain additional throughput for itself must communicate with the AP at its permitted rate. Hence, a client can first observe the current transmission rate of a repeater to the AP to assess for itself the rate that the claimed repeater can access the AP. Both concerns can also be addressed in short order by having nodes verify that communication transpires as promised, and otherwise quit the SoftRepeater connection.

Malicious Stations: SoftRepeater has no explicit defense against stations that wish to jam other transmissions. However, SoftRepeater does not increase susceptibility to jamming attacks, and may in fact reduce susceptibility since the neighboring transmitters have a better signal strength.

A malicious station could lie about its rate to the AP and "pretend" to send packets at a high rate to trick clients. Clients can observe an AP's response (or lack thereof) to this malicious station. In any case, a client would identify the malicious behavior in short order when its end-to-end throughput would drop beyond the expectation. It can subsequently blacklist the repeater (based on MAC address). A sophisticated attacker could spoof MAC addresses at a high rate, confusing a client as to the identity of a valid repeater. Some type of trust or authentication mechanism would be required to obviate this problem.

4. 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 micro-benchmarks related to the repeater initiation protocol.Finally, using simulations, we investigate certain aspects of the performance of SoftRe-

peater in more detail.

4.1 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 Figure 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. 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 to switch between the two networks to a network is around 50ms. 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 3.1.2). 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 [25] 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.

4.1.1 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 Figure 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 Figure 4), which is a conference room located 3 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. Further, 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 repeater for B. The results are shown in Figure 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.

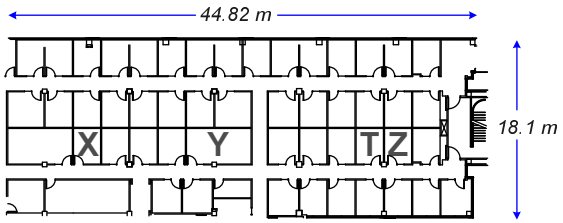

Figure 4: *Floor plan of our office*

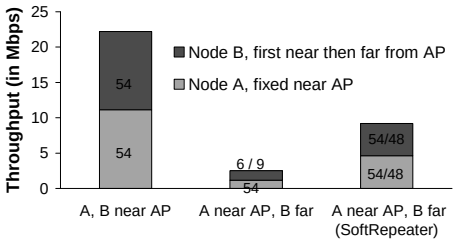

Figure 5: *Downlink UDP flows, with and without SoftRepeater.*

4.1.2 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 Figure 6. When both A and B are at location Y of Figure 4, both of them get a throughput of approximately 9 Mbps. We then move B to location T [4]. 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% [5].

4.1.3 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 Figure 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

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

[5] 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.

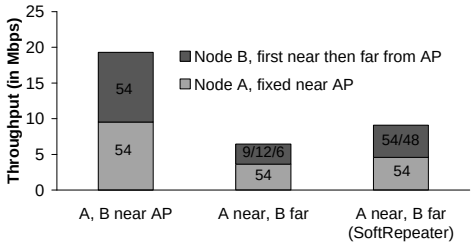

Figure 6: *Downlink TCP flows, with and without a SoftRepeater.*

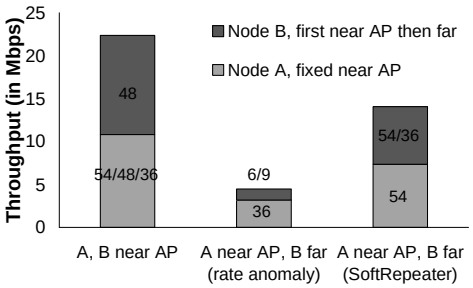

Figure 7: *Uplink UDP flows, with and without SoftRepeater.*

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.

4.1.4 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 Figure 4, and the AP was fixed at location X. We started downlink UDP flows from the AP to all the stations, and plot the throughput of each of the flows in Figure 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.

4.2 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.

4.2.1 Signal Strength vs Data Rate

The Repeater Utility function, described in Section 3.1.2, 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

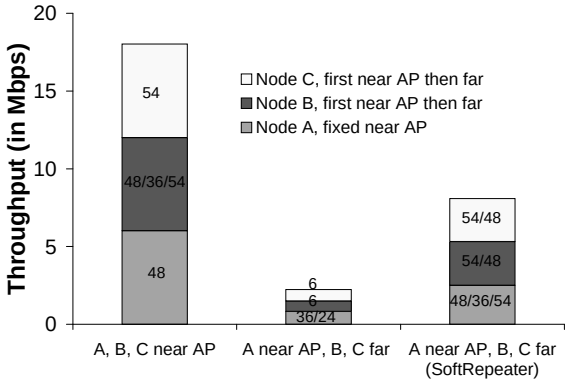

Figure 8: *Downlink UDP flows when SoftRepeater serves two clients.*

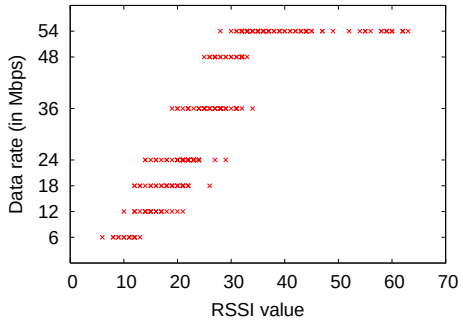

Figure 9: *Correlation between RSSI and data rate.*

location, we measured the RSSI of the received packets, and the data rates there were sent at. The results are shown in Figure 9 [6]. 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 which 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 3.1.2, we can also use a more robust approach to infer data rates like [27].

4.2.2 A 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.

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

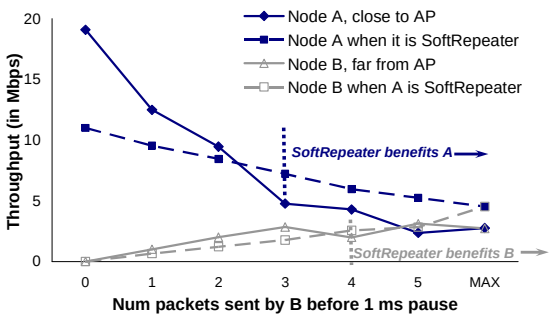

Figure 10: *AP sends full blast traffic to A, and sends bursts of packets to B with a pause for 1 ms. MAX corresponds to back to back UDP packets.*

However, if B is not sending or receiving significant traffic, there is no need for A to offer the SoftRepeater functionality.

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.

Figure 10 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 Figure 10. 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 Figure 10.

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 3 packets in each burst. Similarly, it is not in B's interest to join a repeater network until it starts to receive more than 4 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 3 packets in a burst. For this case, A calculated its expected throughput would be 5.5 Mbps, if it turned on the repeater functionality. Further, the repeater network was started (i.e. B joined it) only when B started receiving more than 4 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.

4.2.3 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.

As discussed in Section 3.1, 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 pre-set threshold (50%) (2) A rate anomaly scenario in which the ratio of packets sent to different stations (i.e., Packet Ratio denoted in Table 1) is disproportionate to their corresponding data rate ratio. We use 1/2 as the threshold. (3) The potential repeater observes a strong signal strength ($\geq$ 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 1.

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 are 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 A indeed started a repeater network, and A's and B's throughput increased to 3.24 Mbps 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 6Mbps while sending to B and 54Mbps 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

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)

Table 1: *Results of the Repeater Initiation Protocol for 5 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.*

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 Mbps 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 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.

4.3 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 [4].

4.4 Simulation Results

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 [21]. In addition to the protocol, we also built a simple model of indoor signal propagation that mimics the testbed environment used for experiments reported earlier. Our simulations will focus on the objective of maximizing the minimum (see Section 3.1.2).

4.4.1 Impact of switching overhead

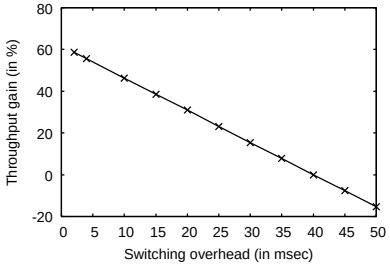

Figure 11: *Impact of switching overhead on TCP throughput.*

We simulate an indoor office environment, similar to one shown in Figure 4. The AP is located in one of the offices, station A is located 3 offices away from the AP, and station B is located 9 offices away from the AP. A sender connected to the AP via a wired link sends TCP traffic to the two stations as fast as it can. The wired link has a bandwidth of 100Mbps, and a delay of 50ms.

When station A does not act as a repeater, both A and B receive a throughput of 3.9Mbps (the baseline case). Next, we force A to always act as a repeater for B (which becomes the client). We assume a switching cycle of 200ms and vary the switching overhead from 2ms to 50ms. Figure 11 shows the improvement in throughput over the baseline case for various switching overheads. We see that repeater functionality improves performance until the switching overhead exceeds 40% of the switching cycle.

4.4.2 Effectiveness of Repeater Initiation Protocol

We now evaluate the effectiveness of the repeater initiation protocol, such that it turns on the repeater functionality only when there is throughput improvement. The AP is placed in office 0, station B is placed 9 offices away, and station A is moved from office 0 to office 9. A wired host attached to the AP sends UDP traffic to both A and B as fast as it can. The switch cycle is 200ms, and switching overhead is set to 4ms. We consider two cases: (a) both A and B are running the repeater initiation protocol, and (b) A is forced to be the repeater for B. Figure 12 shows the throughput improvement over the baseline case where no SoftRepeater is used.

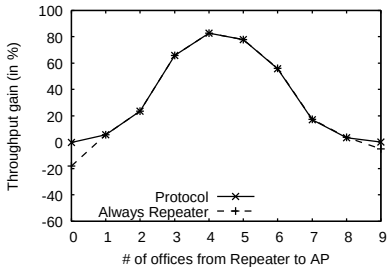

Figure 12: *Max-min fairness: (a) Repeater Initiation Protocol is used, (b) SoftRepeater is always ON.*

In most cases, the repeater functionality should be turned on. However, when A is in office 0 and office 9, turning on the repeater can introduce throughput loss. The repeater initiation protocol correctly detects this, and does not turn on the repeater functionality in those cases.

4.4.3 Larger networks

In our evaluation of the SoftRepeater protocol so far, we have focused on scenarios involving two or three stations. We now consider larger networks.

The first scenario we consider is as follows. The AP is located in one of the offices. Station A is located 3 offices away from the AP. Station B is located 9 offices away. A number of interfering nodes (see the multi-node analysis in Section 3.1.2) are located 5 offices away. A UDP sender attached to the AP sends downlink traffic to all stations as fast as possible. We consider three scenarios. First, no repeater functionality is used. Second, station A is always forced to be a repeater to serve station B (which becomes the client). Third, we run the repeater initiation protocol and let it decide whether to turn the repeater functionality on.

# other nodes	No repeater	Repeater ON		Protocol (MaxMin)	
	All nodes	Nodes A & B	Other nodes	Nodes A & B	Other nodes
0	4.55	7.53	-	7.53	-
2	3.27	2.52	7.87	3.27	3.27
4	2.56	1.50	4.70	2.56	2.56
6	2.09	1.08	3.34	2.09	2.09

Table 2: Average throughput (in Mbps) in the presence of interfering nodes (denoted by "other" nodes here).

Table 2 shows the results. Our analysis in Section 3.1.2 shows that in presence of several competing nodes, turning on the repeater functionality brings little benefit to either the client or the repeater. Thus, when the Max-Min Fairness objective is used, the repeater functionality is generally not turned on in presence of competing nodes, as confirmed in Table 2. On the other hand, we see that the total throughput does improve if the repeater functionality is turned on. We have verified that if we are to maximize the total throughput, the protocol does switch on the repeater functionality.

In the previous scenario, a large number of stations did not participate in the repeater networks. We now consider a different case. We place N repeaters three offices away from the AP, and N other clients 9 offices away. We consider the UDP sender, as before. We establish the baseline by measuring throughput without any repeater functionality. Next, we turn on the repeater functionality, but ensure that each of the N far nodes is associated with a distinct repeater in office 3. Figure 13 shows that the improvement over the baseline throughput is at least 55%. We note that this is the best-case scenario that requires coordination among the repeaters. We are currently developing a repeater coordination protocol for this purpose.

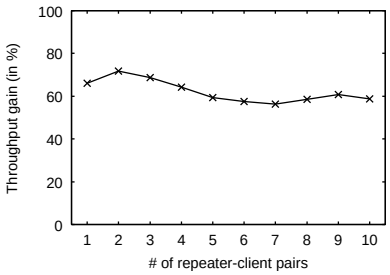

Figure 13: *Improvement in throughput with multiple repeater-client pairs.*

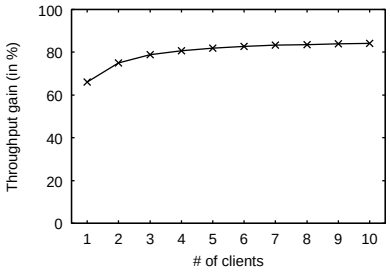

Figure 14: *Improvement in throughput with multiple clients.*

We next consider the case when a single repeater serves multiple clients. We place station A 3 offices away from the AP, and N clients 9 offices away. As before, we consider the UDP sender. We establish the baseline by measuring the throughput without the repeater. Then, we run the repeater initiation protocol, which will have station A as the repeater serve all the N clients. Figure 14 shows that the improvement over baseline throughput is more than 65%.

5. RELATED WORK

In [13], 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 [14, 20], and therefore require a new MAC and would not interoperate with the defacto 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 [23, 11] 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.

Multi-hop extensions to WLANs, such as those proposed in [17, 19], 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 [18], while having been implemented, only supports the ad-hoc mode and does not implement some of the MAC-layer features due to hardware constraints. Also, SoftRepeater can use multiple channels, while CoopMAC cannot.

Mesh routing schemes (e.g., [9, 10]) focus on increasing throughput in an ad-hoc setting. Specifically, WCETT [10] aims of 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 3.1.2). Although mesh routing schemes consider more complex cases with multiple hops, using three or more hops to mitigate rate anomaly only brings marginal benefits [17]. On the other hand, unlike SoftRepeater, these schemes cannot address fairness issues involving more than one station.

Commercially available hardware repeaters [8, 7, 26] 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.

6. 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 and a mechanism for dynamically starting a repeater network without breaking existing connections. 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 [3]. Second, we are enhancing the Repeater Utility Function with the power consumption of the SoftRepeater, 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.

7. REFERENCES

[1] A. Adya, P. Bahl, R. Chandra, and L. Qiu. Architecture and Techniques for Diagnosing Faults in IEEE 802.11 Infrastructure Networks. In *MOBICOM*, 2004.

[2] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *SIGCOMM*, 2004.

[3] Atheros. Atheros Wireless LAN. http://www.atheros.com/.

[4] Bahl et al. Opportunistic Use of Client Repeaters to Improve Performance of WLANs. Technical Report MSR-TR-2008-149, Microsoft Research, 2008.

[5] R. Chandra, V. Bahl, and P. Bahl. MultiNet: Connecting to multiple IEEE 802.11 networks using a single wireless card. In *INFOCOM*, 2004.

[6] R. Chandra, V. Padmanabhan, and M. Zhang. WiFiProfiler: Cooperative Fault Diagnosis in WLANs. In *MOBISYS*, 2006.

[7] Cisco. Aironet 1200 Series Access Point. http://www.cisco.com/en/US/products/hw/wireless/ps430/index.html.

[8] D-Link. Air Pro Wireless Acess Point. http://www.dlink.com/.

[9] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. High-throughput path metric for multi-hop wireless routing. In *MOBICOM*, 2003.

[10] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh network. In *MOBICOM*, 2004.

[11] J. Dunn, M. Neufeld, A. Sheth, D. Grunwald, and J. K. Bennett. A Practical Cross-Layer Mechanism For Fairness in 802.11 Networks. *MONET*, 11(1):37–45, 2006.

[12] C. T. Ee and R. Bajcsy. Congestion Control and Fairness for Many-to-One Routing in Sensor Networks. In *ACM SenSys*, Nov 2004.

[13] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance Anomaly of 802.11b. In *INFOCOM*, 2003.

[14] 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 *SIGCOMM*, 2005.

[15] S. Kandula, K. Lin, T. Badirkhanli, and D. Katabi. FatVAP: Aggregating AP Backhaul Bandwidth. In *Proc. of NSDI*, 2008.

[16] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. In *Journal of the Operational Research Society*, volume 49, 1998.

[17] S. Lee, S. Banerjee, and B. Bhattacharjee. The Case for a Multi-hop Wireless Local Area Network. In *INFOCOM*, 2004.

[18] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. Panwar. A Cooperative MAC protocol for Wireless LANs. *JSAC*, 25(2), Feb 2007.

[19] S. Narayanan, P. Liu, and S. Panwar. On the advantages of multi-hop extensions to the IEEE 802.11 infrastructure mode. In *WCNC*, 2005.

[20] B. Sagdehi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic Media Access For Multirate Ad Hoc Networks. In *MOBICOM*, 2002.

[21] Scalable Network Technologies. Qualnet Network Simulator.

[22] A. Sheth, C. Doerr, D. Grunwald, R. Han, and D. Sicker. MOJO: A Distributed Physical Layer Anomaly Detection System for 802.11 WLANs. In *MOBISYS*, 2006.

[23] G. Tan and J. Guttag. Time-based Fairness Improves Performance in Multi-Rate WLANs. 2004.

[24] G. Tan and J. Guttag. The 802.11 MAC Protocol Leads to Inefficient Equilibria. In *INFOCOM*, 2005.

[25] TTCP. The Story of the TTCP Program. http://ftp.arl.mil/ mike/ttcp.html.

[26] WiDeFi. Two Radio Repeaters. http://www.widefi.com/.

[27] S. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust Rate Adaptation for 802.11 Wireless Networks. In *Proc. of ACM MOBICOM*, Sep 2006.