Design, Implementation and Evaluation of an Efficient Opportunistic Retransmission Protocol

Mei-Hsuan Lu
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
meihsual@ece.cmu.edu

Peter Steenkiste
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
prs@cs.cmu.edu

Tsuhan Chen
Cornell University
410 Thurston Avenue
Ithaca NY 14850-2488
tsuhan@ece.cornell.edu

ABSTRACT

This paper presents an efficient opportunistic retransmission protocol (PRO, Protocol for Retransmitting Opportunistically) to improve the performance of IEEE 802.11 WLANs. PRO is a link-layer protocol that allows overhearing nodes to function as relays that retransmit on behalf of the source $after$ they learn about a failed transmission. Relays with stronger connectivity to the destination have a higher chance of delivering the packet than the source does, thereby resulting in a more efficient use of the channel. PRO has four main features. First, channel reciprocity coupled with a run-time calibration process is used to estimate the instantaneous link quality to the destination. Second, a local qualification process filters out poor relays early. Third, a distributed relay selection algorithm chooses the best set of eligible relays among all qualified relays and prioritizes them. Finally, 802.11e Enhanced Distributed Channel Access (EDCA) is leveraged to make sure high priority relays transmit with high probability. PRO is designed to coexist with legacy 802.11 stations. We have implemented PRO in the driver of a commodity wireless card. Our extensive evaluation on both a controlled testbed and in the real world shows that PRO boosts throughput in diverse wireless environments, and especially in when there is significant contention for the channels, under fading, and with user mobility.

Categories and Subject Descriptors

C.2.2 [**Computer System Organization**]: Computer-Communication Networks—*Network Protocols*

General Terms

Design, Experimentation, Measurement, Performance, Reliability

Keywords

Opportunistic Retransmission, Wireless LANs, Relaying

1. INTRODUCTION

Wireless local area networks (WLANs) have become very popular, but the complex behavior of wireless signal propagation, particularly indoors, creates significant challenges. In this paper, we present an efficient opportunistic retransmission protocol (PRO, Protocol for Retransmitting Opportunistically) that improves network performance in dynamic infrastructure WLANs. The idea is to exploit overhearing nodes to retransmit (or relay) on behalf of the source *after* they learn about a failed transmission. Opportunistic retransmission leverages the fact that wireless networks inherently use broadcast transmission and that errors are mostly location dependent. Thus, if the intended recipient does not receive the packet, other nodes may receive the packet and then become candidate senders for that packet. With multiple senders distributed in space, the chance that at least one available sender succeeds in transmitting the packet increases. Candidate relays participate if they have a higher chance of delivering the packet successfully than the source, thus resulting in a more efficient use of the channel.

Opportunistic retransmission takes advantage of packet reception outcomes that are random and unpredictable, similar to techniques such as opportunistic routing [4, 8, 9] or opportunistic relaying [6]. There are however significant differences. In contrast to opportunistic routing, opportunistic retransmission is a *link layer* scheme operating on a per-transmission basis. This improves efficiency by avoiding, e.g., routing overhead. Opportunistic retransmission also differs from opportunistic relaying since it does not require physical layer support (e.g. decoding the combined relayed and direct signals), making it easier to deploy. We compare PRO with related work in detail in Section 10.

Opportunistic retransmission involves two key challenges. First, it requires an effective measure of link quality to decide whether a node is suitable to serve as a relay. This metric must accurately reflect channel conditions in fast changing wireless environments. PRO leverages path loss information (i.e. Received Signal Strength Indicator (RSSI)) obtained via channel reciprocity to estimate instantaneous channel quality [17]. An automatic on-line calibration scheme is used to deal with link asymmetry. RSSI information is reported by almost all wireless cards, making RSSI-based estimation a practical solution.

The second challenge is to efficiently coordinate the retransmission process given that there may be many candidate relays. The protocol needs to ensure the *best* relay that overheard the transmission forwards the packet while avoiding simultaneous retransmission attempts that can lead to

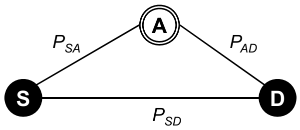


Figure 1: A three-node network containing source (S), destination (D), and a single relay (A). P_{ij} represents the packet delivery rate from i to j.

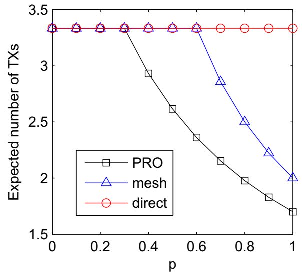


Figure 2: Analytical comparison results with $P_{SA} = P_{AD} = p$ and $P_{SD} = 0.3$.

duplicates or collisions. Prior work relies on per-transmission feedback from all the receivers to perform centralized relay selection [6, 8, 13, 18]. While feedback simplifies relay selection, it also adds overhead in time or frequency. PRO reduces the coordination overhead by using distributed relay selection. Specifically, qualified relays periodically share information about the quality of the channel with respect to sources and destinations. Using this information, relays can then (locally) decide whether they should retransmit packets for a particular source-destination pair, and if so, what their priority is. If multiple eligible relays contend to relay, PRO leverages 802.11e Enhanced Distributed Channel Access (EDCA) [1] to prioritize retransmissions so relays with a higher RSSI with respect to the destination are more likely to retransmit.

The contributions of this paper are as follows. First, we designed an efficient link-layer opportunistic retransmission protocol, PRO, which is simple, lightweight, compatible with the 802.11 standard, and allows partial deployment (i.e., PRO-enabled and legacy nodes can coexist). Second, we implemented PRO in the Madwifi driver for wireless NICs using the Atheros chipset [20], allowing for immediate deployment of PRO. Third, we explored both how opportunistic communication affects fairness, and how it can be combined with transmit rate adaptation algorithms. These issues are important yet often overlooked in prior work. Finally, we evaluated PRO in diverse environments, including a controlled testbed and two real world settings. Our results show that PRO boosts throughput, especially in high contention channels, under channel fading, or with user mobility.

The remainder of this paper is organized as follows. Section 2 describes the basic concept of opportunistic retransmission with a simple analysis. Section 3 provides an overview of PRO. Sections 4 and 5 elaborate on the main protocol components. Section 6 discusses the fairness issue. Section 7 presentes a multi-rate PRO. Sections 8 and 9 present our evaluation results on a controlled testbed and in the real world respectively. Sections 10 and 11 discuss related work and summarize the paper.

2. BASIC CONCEPT

The basic idea of opportunistic retransmission is to have intermediate nodes that overhear a failed packet to retransmit (or relay) the packet on behalf of the source. Here we provide some intuition into the potential benefits of opportunistic retransmission using the three-node network in Figure 1. We denote P_{ij} as the packet delivery rate (PDR) from i to j.

Let us look for the best strategy to deliver a packet from S to D, using the expected number of transmissions needed to deliver a packet as the metric. The simplest strategy is to simply transmit the packet directly from S to D. On average, direct communication takes

$$TX_{direct} = \frac{1}{P_{SD}} \tag{1}$$

transmissions to deliver a packet. An alternative is to exploit opportunistic retransmission. If a packet transmission from S to D fails, but the packet is overheard by A, then it is more efficient to have A retransmit (or relay) the packet rather than S if A has a higher packet delivery rate to D. Based on this philosophy, we can calculate the expected number of transmissions in the case of opportunistic retransmission, assuming no overhead, as

$$TX_{opport\_relay} = \sum_{i=1}^{\infty} iP(i), \tag{2}$$

where $P(i)$ is the probability of taking i transmissions to deliver a packet. The derivation of $P(i)$ is given in (4) at the top of the next page. Finally, in a mesh network based approach, there are two options: S sends a packet to D directly, or S sends the packet to A, which A then forwards it to D. S uses the option that requires the fewest transmissions. On average, this method takes

$$TX_{mesh\_network} = \min(\frac{1}{P_{SA}} + \frac{1}{P_{AD}}, \frac{1}{P_{SD}}) \tag{3}$$

transmissions to deliver a packet.

Figure 2 compares the three schemes for $P_{SA} = P_{AD} = p$ and $P_{SD} = 0.3$. The figure shows that with the participation of an appropriate relay PRO outperforms the mesh approach, which in turn outperforms direct communication. In fact, it can be formally shown that, ignoring overheads, opportunistic retransmission statistically always takes equal or fewer transmissions than mesh networking and direct communication [19].

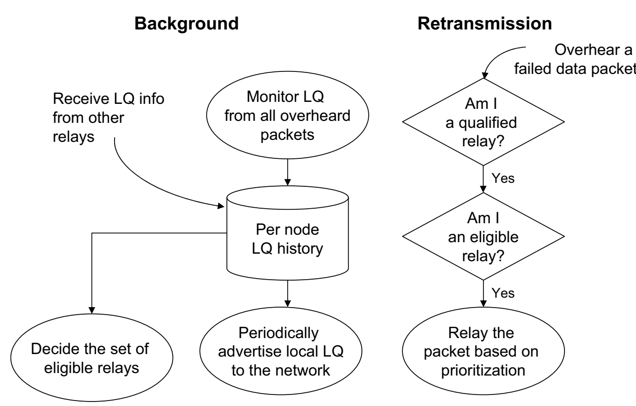


Figure 3: Protocol flowchart of PRO

3. PROTOCOL OVERVIEW

Figure 3 gives a high-level overview of PRO. In the background, candidate relays continuously monitor the link quality with respect to the source(s) and the destination(s). The channel quality to the destination shows how likely the node will successfully (re)transmit packets to the destination. The channel quality to the source indicates how often the node

$$P(i) = \begin{cases} P_{SD} & \text{if } i = 1, \\ \sum_{j=1}^{i-1} \Big((1-P_{SD})^j (1-P_{SA})^{j-1} P_{SA} (1-P_{AD})^{i-j-1} P_{AD} \Big) + (1-P_{SD})^{i-1} P_{SD} (1-P_{SA})^{i-1}, & \text{if } i \geq 2 \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

is likely to overhear packets from the source, i.e. how often the node will be in a position to function as a relay for the source. Each node locally decides whether it is a *qualified relay* for a source-destination pair based on a threshold for the quality of the channel to the destination. Qualified relays advertise their link quality with respect to both the source and the destination through periodic broadcasts.

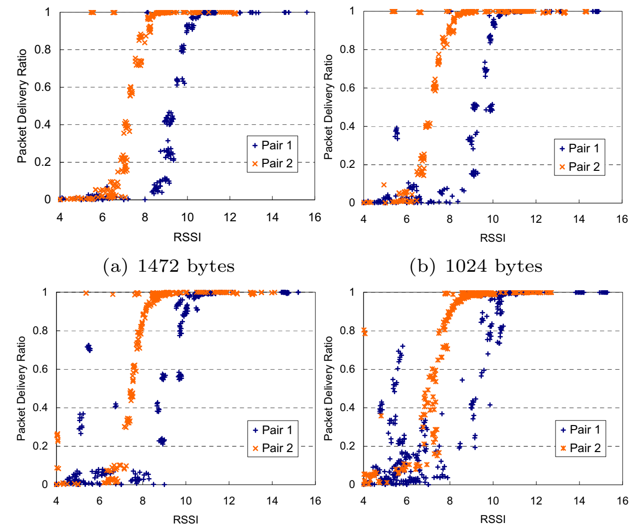
By collecting periodic link quality broadcasts, each qualified relay independently constructs a global map of the connectivity between qualified relays, the source, and the destination. Using this information, each qualified relay then decides whether it is an *eligible relay* for a source-destination pair. Only eligible relays are allowed to retransmit after a failed transmission. Clearly, the selection process should result in a set of eligible relays that is large enough so there is a high likelihood that one of them overhears the source. However, including too many relays can be harmful for several reasons. First, using too many relays can potentially increase contention in the network which may result in more collisions. Second, having poor relays retransmit can prevent or delay retransmission by better relays, thus reducing the success rate for retransmissions.

Upon a failed transmission, eligible relays that overheard the packet then participate in the retransmission of the packet as if they were retransmitting a local packet, i.e., they follow the 802.11 random access procedure. Relays stop the retransmission when they overhear an ACK that confirms a successful reception by the destination. To give precedence to relays with better connectivity to the destination, eligible relays choose the size of their initial contention window based on their priority i.e. their rank in terms of how effective they are among all eligible relays. Relays with a higher rank are associated with a smaller contention window so that they have a higher chance of accessing the channel. We elaborate on each functional component in Section 5.

4. ESTIMATING LINK QUALITY

Link quality information is needed to quantify the suitability of a node as a relay. We need a measure for link quality that is both accurate and easy to obtain. One solution is to assess link quality by monitoring the success or failure of probe messages [10, 11]. The resulting packet delivery rate is then used as an estimate of link quality. Probing-based methods do not need hardware support but respond slowly to channel dynamics. Moreover, probe messages require extra bandwidth, which may undo the gains from opportunistic retransmission. Another solution is to use location information with respect to sources and destinations based on ideas from geographical routing [22]. This requires infrastructure support for distance estimation (e.g. GPS devices) and a mechanisms to translate physical distance into link quality, which is nontrivial given shadowing and fading effects.

An alternative is to estimate link quality by monitoring signal-to-noise ratio (SNR) of packets at the receiver [7]. SNR-based solutions are attractive because they can potentially adapt quickly to the changing signal environment. The



(a) 1472 bytes (b) 1024 bytes

(c) 512 bytes (d) 16 bytes

Figure 4: Measurement results of packet delivery rates and RSSI with transmit rate equal to 11 Mbps for different packet sizes.

SNR at a receiver can be written as follows:

$$SINR = \frac{P_r}{P_{thermal} + P_{INI}}, \quad (5)$$

where P_r is the received signal strength, $P_{thermal}$ is the thermal noise and P_{INI} is the sum of the power received from all interferers. Because thermal noise is usually fairly constant and interference is reduced significantly by the use of carrier sense, SNR is largely determined by the received signal strength (RSS) [16]. This is especially true in indoor environments where multi-path effects are mitigated by dynamic equalizer in wireless network cards.

In practice RSS can be estimated using the Received Signal Strength Indicator (RSSI) [16, 21], which is reported by most wireless cards. To understand how RSSI relates to PDR, we use the CMU wireless emulator [15] to collect measurements of PDR and RSSI for UDP packets of different sizes (1472, 1024, 512, and 16 bytes). The test nodes are equipped with wireless cards using the Atheros AR5212 chipset. The path loss between the transmitter and receiver is changed from 90 to 110 dB in a step size of 0.5 dB. For each loss value, we collected the average RSSI and PDR for 10 experiments of over 1000 packets each. Figure 4 shows the results for two transmitter/receiver pairs, out of a total of 10 pairs. The other eight pairs exhibit similar behavior. We make the following observations based on these results:

1. PDR as function of RSSI is somewhat noisy, in particular for 16-byte UDP packets. However, there is generally a strong correlation between RSSI and PDR.
2. There is an RSSI high threshold (Th_h), above which packets are nearly always received.

3. The PDR-RSSI graphs for different cards have a similar shape, but are shifted by 2-4 dB. Based on channel reciprocity, forward link quality can be predicted by reverse link conditions if the amount of the shift is known.

These results suggest that RSSI is not a perfect measure for PDR, but as we will show later it suffices for our needs. PRO does not require a very accurate measure of link quality because link quality is only used to help select and prioritize a reasonable set of relays from a larger pool, and small changes in quality should not affect this process.

In practice channel conditions vary with time. To predict the current RSSI, PRO applies the time-aware prediction algorithm proposed in [17] to the RSSI history of packets. This approach improves exponential weighted moving averaging (EWMA) by giving recent samples more weight and by filtering out sharp transient fades that last for only a single packet.

5. PROTOCOL DESIGN

We elaborate on the four components of PRO: relay qualification, relay selection, relay prioritization and retransmission.

5.1 Relay Qualification

Using too many or poor relays can hurt performance since it increases the probability of collisions while offering limited opportunistic gains. To filter out poor relays early, candidate nodes must pass a qualification process by comparing their RSSI with respect to the destination with a threshold Th_h. Qualified relays periodically broadcast their link quality with respect to the source and the destination. This information is then used for relay selection, which we will describe in Section 5.2.

The main challenge in relay qualification is the fact that using reverse link conditions to predict forward link quality is imprecise when links are asymmetric. Unfortunately, conveying RSSI information from the destination to the relay, e.g., using RTS/CTS [7, 12], introduces relatively high overheads that can easily undo any performance benefits. PRO avoids such overheads by using channel reciprocity combined with on-line threshold calibration based on observed performance. Initially, relays assume a default Th_h of 10 dB (the average from our offline measurements). At run time, each relay records the transmission results - success or failure - after each transmission from itself to the destination. The value of Th_h is incremented by 1 if the packet delivery ratio over 100 transmissions is lower than 0.75 and decremented by 1 if it is equal to 1. As this threshold may vary from receiver to receiver, each transmitter maintains a Th_h for each receiver that it is communicating with and updates these thresholds independently.

The above solution is based on the observation made in Section 4 that the PDR-RSSI plots for different sized packets across different source-destination pairs have a similar shape. Note that the calibration process does not need to consider the reason for the packet losses, making it agile to deal with various conditions. For example, if packet losses are due to a jammer near the destination, then the calibration process gradually increments Th_h, making the relay less and less likely to pass the relay qualification process. The calibration process resets Th_h to the default value if no transmission to a destination occurs during 30 minutes to compensate for threshold adjustment due to the environment.

5.2 Relay Selection

Relay selection finds the best set of relay(s) among all qualified candidates to retransmit a failed packet. To assure that overhead does not overwhelm gains, PRO uses a distributed relay selection algorithm. Each qualified relay runs the algorithm to find a set of eligible relays out of all the qualified relays based on their link quality with respect to the source and the destination. We now elaborate on how relays share link quality information and how eligible relays are selected. We discuss relay prioritization in Section 5.3.

5.2.1 Sharing Link Quality via Periodic Broadcasts

The relay selection algorithm considers the link quality to the source and the destination of all the qualified relays. This information is collected by periodic broadcasts from all the qualified relays. The link quality to the destination is predicted using the RSSI of the reverse channel (Section 5.1). The link quality to the source is the packet reception rate, which is obtained by keeping track of sequence numbers in packets originated from the source. According to the 802.11 specification, sequence numbers are incremented by 1 for each packet. Thus packet losses are detectable from a gap in sequence numbers. A node can qualify as a relay for multiple sessions, so the broadcast messages should contain information for all the sessions that it is participating in.

The periodic broadcast frequency is 1 second in our implementation. This value is borrowed from the default HELLO message interval used in AODV. Relays can further reduce the broadcast overhead by adapting the broadcast frequency based on how fast the channel conditions change. They can also suppress broadcasts when the chance of becoming an eligible relay is low. When a qualified relay fails the qualification process (the predicted RSSI falls below Th_h or the relay does not hear the destination for 2 seconds), it stops broadcasting link quality information for that destination. Other relays exclude a relay from the relay selection process if they do not hear its broadcasts for 2 seconds.

5.2.2 Selecting Eligible Relays

The relay selection algorithm is based on the following design guidelines:

- Relays with stronger connectivity to the destination are favored because they have a higher chance of successfully transmitting the packet.
- Relays with stronger connectivity to the source are favored because they have a higher likelihood of overhearing the source and offer opportunistic gains.
- The resulting set must be large enough so there is a high chance that at least one eligible relay overhears the source. We also want to limit the set size to minimize any increases in collision rates.

The algorithm is iterative and starts by selecting the node that has the highest RSSI with respect to the destination. It continues to add the nodes with the next highest RSSI until the probability of having one of the selected relays hear the source is larger than a threshold Th_r. The results in Section 8 and Section 9 show that our relay selection algorithm works well. Algorithm 1 gives the psuedo code for the algorithm.

Algorithm 1 RELAYSELECTION(Q)

Require: All qualified relays Q
Ensure: All eligible relays R

Initialize Q to the set of all qualified relays
$R \Leftarrow \emptyset$
$p \Leftarrow 1$
Rank Q according to the RSSI with respect to the destination
while Q is not empty **do**
  Pick the highest ranked q in Q
  Insert q to R and delete q from Q
  Retrieve the source packet reception ratio α_q of q
  $p \Leftarrow p \cdot (1 - \alpha_q)$
  if $1 - p > Th_r$ **then**
    return R
  end if
end while

5.3 Relay Prioritization

When multiple eligible relays overhear a failed transmission, it is advantageous to have the relay with highest RSSI to the destination retransmit the packet. This can be achieved by having relays coordinate after each failed retransmission to determine who should retransmit the packet [6, 8, 18]. Explicit coordination has two problems. First, it requires scheduling among relays to decide who sends feedback when, which introduces additional complexity. Second, the overhead of distributing feedback for every failed transmission can be considerable.

PRO avoids using feedback on a per-retransmission basis and instead leverages the 802.11 protocol to prioritize the relays. The 802.11 standard provides several mechanisms for achieving this, e.g. by managing the minimal and maximal contention window size (CW_{min} and CW_{max}), backoff increasing factor, interframe space, and backoff time distribution [5]. For example, 802.11e EDCA [1] performs prioritization by manipulating interframe space or/and contention window size. In our implementation, effective relays obtain a higher priority by using a smaller CW_{min}, but other parameters can be considered. Note that the source behaves as an eligible relay after a failed transmission. We discuss the CW_{min} settings used in our implementation in Section 8.

5.4 Retransmission

Relays detect failed transmissions through the lack of an ACK. Eligible relays that overheard the packet then contend to retransmit the packet using the contention window selected during the relay prioritization procedure. If a relay is eligible for serving multiple sessions, it ignores transmission activities from other sessions when it is already in the process of relaying. If a retransmission fails, relays double the size of contention windows and again contend for the channel, similar to the process used for local retransmissions. The use of relaying is transparent to legacy 802.11 stations.

Relays terminate the retransmission process when they observe one of the following events:

- An ACK frame destined for the source is overheard, since it implies successful reception.
- The retry limit is reached after several unsuccessful transmissions.
- A new data packet (i.e. a packet stamped with a higher sequence number) originated from the source is overheard. This means that either the source has discarded the current packet, or the packet was successfully received but the relay missed the ACK.

It is possible that the relay overhears a retransmission of the packet after hearing an ACK. This means that the packet was received successfully, but either a relay or the source missed the ACK. In this case, the relay should reacknowledge the packet to avoid further retransmissions. The obvious solution, sending another ACK, does not work: if multiple relays detect the unnecessary retransmission, it will lead to systematic collisions of the ACKs. Instead, the relays send a "null" data packet, i.e. the original data packet with the payload stripped, as an acknowledgement. Since null packets are transmitted as data packets (i.e. using backoff), we avoid systematic collisions. Note that when the unnecessary retransmission was sent by the source, this approach effectively corresponds to relaying of the ACK.

When the channel between the source and destination is very poor, frequent relaying of ACKs may occur. In that case, it may be more efficient to employ a mesh network based approach (i.e. the source sends packets to a relay which forwards them to the destination) as almost all packets will be relayed anyway. We present experimental results for such a scenario in Section 9.1. It is possible to dynamically switch between opportunistic relaying and mesh-based forwarding, similar to [2]. For example, when a source observes frequent relaying of ACKs, it can switch to the mesh networking mode and notify relays by setting a flag in the packets. The source can then select the best relay to function as the forwarder, for example based on the channel state information it obtained from periodic broadcasts. When the forwarder receives a packet from the source, it generates an ACK and forwards the packet to the destination. The source can switch back to the opportunistic relaying mode when it starts overhearing traffic from the destination.

6. COLLISION AVOIDANCE AND FAIRNESS

The protocol described in the previous section has a number of parameters that control how aggressive it is with respect to legacy nodes that do not use relaying. This creates tradeoffs with respect to network efficiency and fairness.

First, the use of multiple relays to retransmit packets in PRO introduces the risk of increased collision rates, which can affect the performance of all flows in the network. PRO tries to reduce this risk through the design of its relay selection algorithm. By limiting the number of eligible relays, the algorithm limits (but not completely prevents) increases in collisions. Collision resolution is handled by having relays use 802.11's binary exponential backoff procedure: the size of contention window doubles after every attempt of retransmission until a maximum size is reached. The use of a large initial contention window can further reduce collisions but the cost is a small increase in delay. In Section 8.1.1, we show experimental results of collision rates. The results suggest that the relay selection algorithm is quite effective in mitigating collisions.

Another concern is fairness. The 802.11 exponential backoff procedure offers not only short-term collision avoidance but also long-term fairness across multiple stations. However, as multiple relays can retransmit on behalf of a single source, the joint channel access behavior is no longer uniform. This results in *unfairness* across flows with different

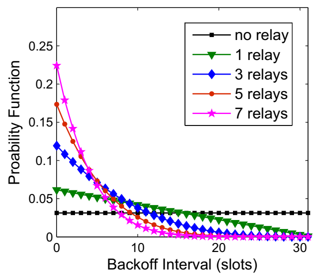


Figure 5: Probability functions of backoff intervals from multiple relays that individually have a uniform backoff distribution

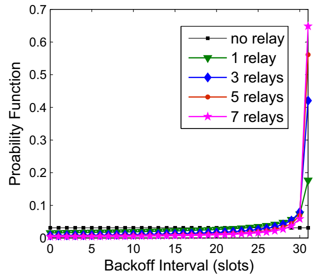


Figure 6: Probability functions of backoff intervals of individual relays that collectively yield a uniform backoff distribution

numbers of relays. Figure 5 shows the probability distribution of the backoff interval as a function of the number of relays, assuming each relay uses a uniform backoff distribution in the range [0, 31]. The figure clearly shows that source-destination pairs that use more relays are more likely to gain access to the channel.

Fairness is a policy question and different policies are possible. For example, one policy might be that it is acceptable to give priority to relayed transmissions, because opportunistic retransmission can improve network efficiency. Another policy is to force equal channel access probabilities across all flows. A particular fairness policy can be achieved by tuning the backoff distribution. One example is the use of larger initial contention windows when relaying, as we already mentioned earlier. The initial contention window can also be tuned based on the number of eligible relays. An even more aggressive solution is to have relays use non-uniform distributions for selecting slots in the contention window. This makes it possible to have the joint behavior of the relays appear as that of a single legacy 802.11 node (i.e. a node that uses a uniform distribution for selecting a slot). Figure 6 shows the probability functions of backoff intervals of individual relays that collectively yield a uniform backoff distribution in a range of [0, 31]. We have not explored these more advanced techniques, but we present an initial evaluation of how the size of the initial contention window used by relays affects performance and fairness in Section 8.3.

7. MULTI-RATE PRO

Current 802.11 provides multi-rate capabilities that allow a sender to change the bit rate adaptively, depending on the quality of the channel to the receiver. The idea is to reduce the packet error rate by lowering bitrates, i.e., rate adaptation trades longer transmit times for higher packet success rates. Opportunistic retransmission adopts a different philosophy: the source and relays continue using higher transmit rates and use relaying to improve the success rate of retransmissions. Hence, opportunistic retransmission trades more transmissions for shorter packet transmission times. This leads to the following question: upon a failed transmission, should the sender reduce the transmit rate or should it rely on relaying to combat errors?

Whether rate adaptation or opportunistic retransmission should be used is a difficult question. The answer depends in part on how well rate adaptation performs, e.g. how quickly does the rate adapt to changes in the environment. Current rate adaptation algorithms are dominated by probe-based approaches. While probe-based approaches are simple and easy to implement, several studies have shown that they perform poorly in highly dynamic environments [7, 12, 14]. Thus, recent efforts have been made to use signal strength measurements to help select the transmission rate [7, 17, 21]. In Sections 8, 9.1 and 9.2, we compare PRO with two rate adaptation algorithms, SampleRate [3] and CHARM [17]. The former is a probe-based method provided with the Madwifi driver and the latter is a SNR-based solution. Our experimental results show that in practical 802.11b scenarios PRO outperforms 802.11 with rate adaptation.

It is however important to explore how rate adaptation and opportunistic relaying can be combined, since both techniques have limitations. For example, no rate adaptation algorithm can completely eliminate packet losses and relaying can help reduce the cost of packet errors; relaying is likely to be more important in very dynamic channels. Opportunistic retransmission on the other hand can only be effective when good relays are available, so rate adaptation remains important, especially for protocols such as 802.11g, in which 12 transmit rates (including the four rates specified in 802.11b) are supported and the radio range of the highest rate (54Mbps) is fairly short. Integrating rate adaptation and opportunistic retransmission is however a complex research problem. The reason is that, in general, rate selection on the source, rate selection on the eligible relays, and the PRO algorithms for the selection and prioritization of eligible relays all depend on each other, resulting in a huge search space. Channel dynamics combined with the high cost of coordination further complicate the design of an efficient integrated solution.

As a first step, we combined PRO with the CHARM rate adaptation algorithm. Our multi-rate PRO is a minimal integration in the sense that we purposely minimized the changes to the PRO and CHARM algorithms. In multi-rate PRO, the source and relays rely on the regular CHARM algorithm to select transmission rates. CHARM uses a rate selection table that lists the minimum required RSSI threshold for each transmit rate. This table is built offline based on general card characteristics and calibrated online to deal with card differences and link asymmetry. Transmitters use channel reciprocity to estimate the received signal strength at the receiver, similar to PRO, and then use table look up to determine the transmit rate to use. For simplicity, multi-rate PRO eliminates the threshold calibration component of CHARM. Instead, it avoids errors caused by per-card differences and link asymmetry by only using three out of the possible twelve rates (18, 36, and 54 Mbps). Upon a transmission failure, multi-rate PRO executes the PRO protocol described in the previous section.

Multi-rate PRO changes the original CHARM and PRO protocols in only two ways. First, we want to avoid that relays that use a lower transmit rate than the source retransmit the packet. This should generally not happen, but this type of "rate inversion" is possible because CHARM updates channel state information more quickly than PRO. When an eligible relay observes that its transmission rate is lower than that used by the source, it disqualifies itself. Second, since relaying makes retransmission more efficient, we make rate selection on the sources more aggressive. This is done by having the sources shift the rates in their threshold table up one class.

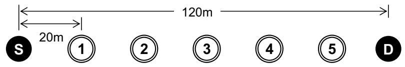


Figure 7: Static scenario topology

Multi-rate PRO works as follows. Sources use the RSSI with respect to the destination as an index to locate the transmit rate. Relays constantly overhear traffic as in PRO to collect link quality information. Upon a failed transmission, eligible relays that overheard the packet use the RSSI with respect to the destination as an index to lookup the transmit rate. The selected rate has to be higher than or equal to the rate used in the original packet; otherwise the retransmission attempt is terminated. The transmit rate of the original packet can be retrieved from the packet header. Relays retransmit the packet according to the relay prioritization procedure specified in Section 5.3. When no relay is present (i.e., no periodic broadcast is received), sources fall back to CHARM. In Section 9.3, the performance of multi-rate PRO in an 802.11g network is presented. The results show that multi-rate PRO, though not yet fully optimized, exhibits better performance over SampleRate and the mesh network based approach in an 802.11g environment.

The above protocol is clearly just a first step. Not only will a full implementation want to use all transmit rates, but some of the mechanisms can also be further optimized. For example, how much more aggressive the source can be in rate selection requires more research. In fact, it may be beneficial to have the rate selection on the source depend on the number and quality of relays.

8. EVALUATION USING THE EMULATOR

We have implemented PRO in the Madwifi driver for wireless NICs based on the Atheros chipset. Our implementation uses FlexMAC [20], a flexible software platform for developing and evaluating CSMA protocols. FlexMAC offers a number of controls in the host that are very useful in implementing PRO (e.g. the user-defined backoff mechanism and flexible retransmission policies). We use the "interoperability" mode to develop PRO so the implementation is 802.11 compatible. This section presents evaluation results in a controlled testbed. The controlled experiments use the CMU wireless network emulator [15], which supports realistic and fully controllable and repeatable wireless experiments. The emulation-based experiments are useful in studying microscopic behavior of the protocol. Real-world experimental results are given in the next section.

In the following tests, the source constantly sends back-to-back UDP packets to the destination. The UDP packets are 1472 bytes each. We run each test for 3 minutes before we start collecting the statistics to allow the run-time calibration process to converge. Each test runs for a one minute and we presents the median of five tests. The experiments in this section use 802.11b (instead of 802.11b/g) because of current testbed limitations. We present 802.11g results in Section 9.3.

8.1 Static Scenario

We construct a topology in which the distance between the source and the destination is 120 meters. Five relays are uniformly placed between the source and the destination as shown in Figure 7. A log distance large scale path loss model is used with a path loss exponent of 3. We first consider the following three scenarios:

- *freespace* is the scenario as described above. It is similar to an outdoor urban environment.
- *fading_k0* adds a Ricean fading envelope with $K = 0$ to the log distance model. This scenario exhibits severe fading.
- *fading_k5* adds a Ricean fading envelope with $K = 5$ to the log distance, so fading is less severe.

In order to differentiate collisions from errors due to corruption, we also add a monitor node in the network. The monitor node is manually configured to have perfect link quality with the other nodes. Thus, packet losses observed by the monitor node must be caused by collisions.

In the following experiments, sources use a CW_{min} of 32 slots for their initial transmissions. Eligible relays select CW_{min} based on their priority. The best two relays use a CW_{min} of 32 slots, the next two use 64 slots, and any remaining relays use 128 slots. The periodic broadcast interval is 1 second and the threshold Th_r is 0.9.

8.1.1 Overall performance

Let us first consider the two extreme scenarios, *freespace* and *fading_k0*. In this experiment, we measure the throughput for different positions of the destination corresponding to different source-destination distances. Figure 8 shows the result for PRO and 802.11 (no rate adaptation). In both scenarios, PRO improves performance over poor channels and it does not harm the performance under good channel conditions. The range of improvement is wider in fading channels.

Next we compare the throughtput of five mechanisms for all three scenarios: 1) 802.11 with rate adaption off (none); 2) 802.11 with CHARM (CHARM); 3) 802.11 with SampleRate (SampleRate); 4) a mesh network that forwards packets along the highest throughput multi-hop path using the highest transmit rate (mesh); and 5) an artificially created optimal case of PRO that involves a single relay with perfect link quality to both the source and the destination (PRO optimal). The last case corresponds to the optimal performance PRO can achieve.

Figure 9 shows the throughput results. In *freespace*, PRO and CHARM perform equally well and they both outperform SampleRate. However, the difference is not significant. In such a static environment, PRO performs close to optimal as the protocol can converge to the best operating point. In *fading_k0* and *fading_k5*, PRO performs the best because SampleRate and CHARM have trouble keeping up with the severe fading environment. As a result, senders continue to use a low rate after the channel improves, and they also suffer packet losses when the channels quality drops. This can be seen in Figure 10, which shows that the retransmission success rates of CHARM and SampleRate drop significantly in fast fading environments. In PRO, relays also sometimes make imprecise prediction under small-scale fading. This suggests that the 1-second broadcast interval for channel updates may be too long in such environments. However, the multi-relay diversity makes PRO less sensitive to imperfect prediction and thus more robust against channel dynamics. This is why PRO outperforms the two rate adaptation techniques. The mesh networking approach relies on a single forwarder, which leads to poor performance when it experiences deep fading.

To investigate the impact of concurrent transmissions from multiple relays we collected the collision rates. The number

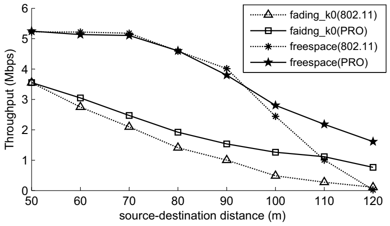


Figure 8: UDP throughput over different source-destination distances

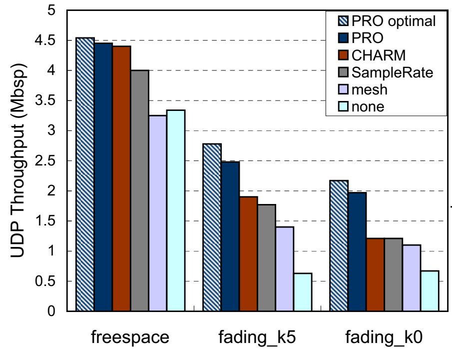


Figure 9: UDP throughput

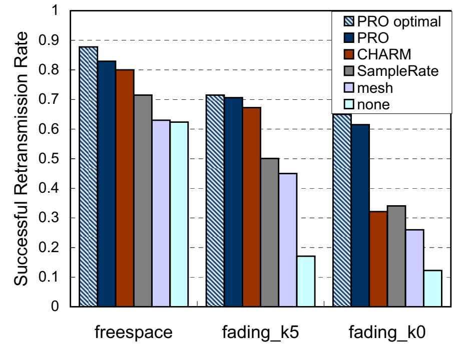


Figure 10: Successful retransmission ratio

Scenario	Collisions (%)
freespace	0.43
fading_k5	0.37
fading_k0	0.29

Figure 11: Overall collision probabilities

Relay ID	1	2	3	4	5
freespace	2	0	0	0	0
fading_k5	1	4	0	0	0
fading_k0	5	5	0	0	0

Figure 12: Online calibration result (offset to the default threshold)

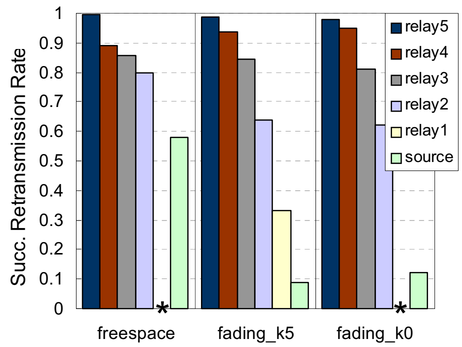


Figure 13: Per-relay retransmission rates. '*' means data points are too few to be meaningful.

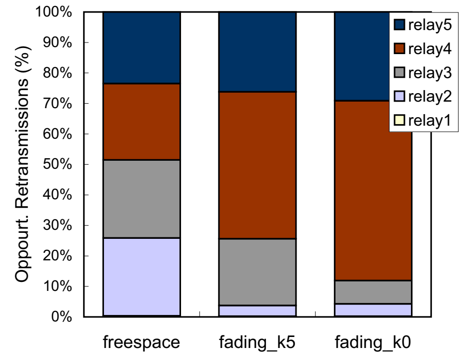


Figure 14: Percentage of opportunistic retransmissions

of collisions was calculated by subtracting the number of transmissions from relays from the number of relayed transmissions captured by the monitor node. The results in Figure 11 show that the collision probabilities are fairly low across all scenarios (< 0.5%). This suggests our choice of threshold Th_r used in the relay selection algorithms works well across scenarios with different degrees of channel fading.

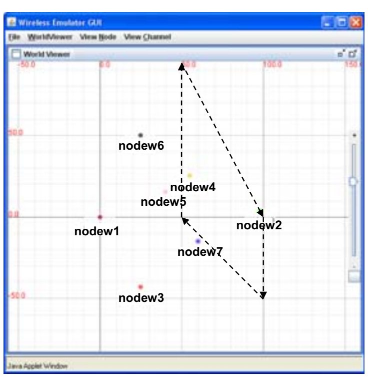


Figure 15: Mobile scenario test topology

8.1.2 Per-relay performance

To gain a better insight into the retransmission process, Figure 13 shows the packet success rates for the six transmitters. Relays closer to the source have lower IDs (see Figure 7). Not surprisingly, we see that the success rates are higher for the nodes closer to the destination. Since relays closer to the destination have a lower path loss, we would like them to have to play a bigger role as retransmitters. Figure 14 confirms this: relays closer to the destination retransmit more packets. Relay5 is an exception. The reason is that it overhears relatively fewer packets so it has fewer retransmission opportunities. These results show that the relay prioritization procedure is effective.

Figure 14 also provides insights into the effect of on-line threshold calibration. We observe that the distribution of retransmissions becomes more skewed towards the relays closest to the destination as channel fading becomes more pronounced. The reason is that the on-line calibration process gradually increases the threshold Th_h for remote relays that perform poorly under high fading conditions. Table 12 shows the calibration offets (relative to the default threshold of 10). It confirms that the threshold offsets for remote relays are higher, especially in cases with significant channel fading. Note that our implementation limits the maximal calibration offset to 5 dB.

8.2 Mobile Scenario

Next we evaluate the mobile scenario shown in Figure 15. It consists of a source (nodew1), a mobile destination (nodew2), and five relays distributed in the test area. The destination navigates a route at a speed of 5 m/s in a clockwise fashion. Because of the mobility, what relays participate in retransmission has to change over time. The channel model combines log distance attenuation with a path loss exponent of 3.3 and Ricean fading with $K = 3$.

Figure 16 shows the result. It shows that PRO outperforms CHARM and SampleRate most of the time. The exception is interval 50-58 seconds, when the destination node moves to (50, 0). At this point, it is out of the communication range of the source at 11 Mbps, so PRO performs

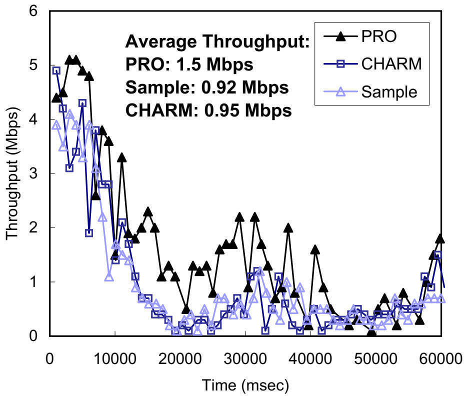
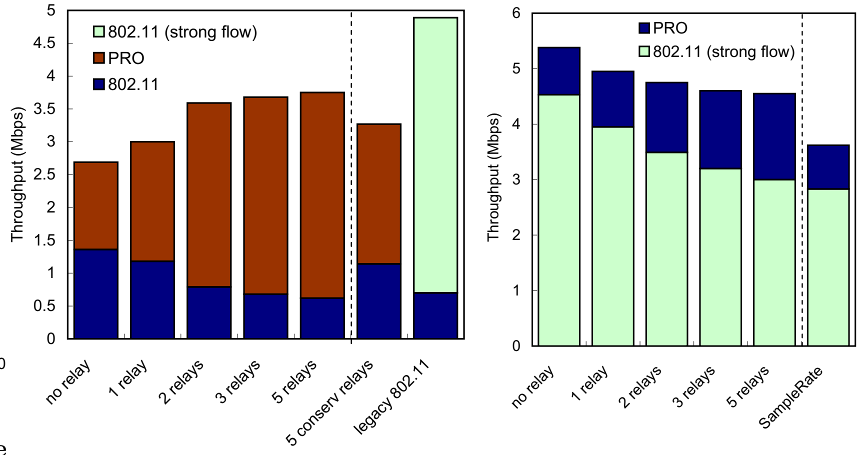


Figure 16: Throughput result of the mobile scenario



(a) Homogenous flows

(b) Heterogeneous flows

Figure 17: Fairness comparison

Figure 18: Floor plan of the office building

poorly (see Section 5.4 for a solution that dynamic switches between PRO and mesh networking). Nevertheless, PRO outperforms rate adaptation by 50% on average. The results indicate that in the presence of good relays, PRO is agile in reacting to changes in the topology.

8.3 Fairness

To understand how PRO affects fairness, we conducted experiments in which PRO and legacy 802.11 stations coexist in the network. To isolate the impact of relaying, we do not use rate adaption in the 802.11 stations, unless otherwise noted. The first scenario includes two source-destination pairs, one using PRO and the other 802.11; the distance between source and destination is 130 meters for each pair. The channel model is log distance with a path loss exponent of 3 and Ricean fading with $K = 0$ is used. We place the two sources close to each other so they defer as a result of carrier sense. For PRO, five relays are uniformly placed between the source and the destination. To study the impact of the numbers of relays, we add relays one by one, starting with the one closest to the source.

Figure 17(a) shows the throughput result. Overall, as more relays participate, PRO sees an increase in throughput but the throughput of the 802.11 station decreases. Since the increase observed by PRO is distinctly higher than the throughput drop for the 802.11 node, network capacity increases. As discussed in Section 6, this "unfair" phenomenon is due to the non-uniform channel access behavior with multiple relays. The unfairness can be reduced by increasing the contention window size for relays. For example, we tried a conservative policy by using 32 slots for the best relay (instead of the best two relays), 64 slots for the next relay (instead of the third and fourth best relays), and 128 slots for the rest. The "5 conserv relays" bar in Figure 17 shows that this strategy reduces the unfairness, although there is also a drop in aggregate throughput. If fairness is a major concern, we can further increase CW_{min} or strictly limit the number of relays. Note though that 802.11 is not a fair protocol due to its binary exponential backoff process: sessions that experience packet losses will constantly contend for the channel with larger contention windows. To illustrate this, we conducted an experiment that includes two 802.11 source-destination pairs, one with perfect channel conditions and another with a poor channel that experiences constant packet loss. We manipulated the source-destination distances such that the poor session sees a throughput equal to the 802.11 session in the 5-relay case. The rightmost bar in Figure 17 shows that there is extreme unfairness.

We next consider a heterogeneous scenario where the two flows have different source-destination distances: one is 100 meters and the other is 50 meters. The distant pair uses PRO while the close pair uses 802.11 over a nearly error-free channel. The rest of the experimental setup remains the same. Figure 17(b) shows the throughput result. In contrast to the equi-distant experiment, network capacity decreases with the number of relays: PRO sees an increase in throughput, but throughput of the 802.11 station decreases. In all cases, the increase observed by PRO is lower than the throughput drop for the 802.11 pair. PRO increases the successful retransmission ratio of the distant source-destination pair, resulting in a relatively smaller backoff interval and a better chance to gain channel access. However, the more frequent transmissions from the distant source-destination pair also reduces efficiency, thereby reducing aggregate throughput. In fact, this phenomenon also exists in 802.11 rate adaptation. To show this, we configure the remote pair to run 802.11 with SampleRate. The result is shown in the rightmost bar in Figure 17(b). We see that rate adaptation exhibits the same tradeoff: improving link quality for poor sessions increases fairness but reduces aggregate net-

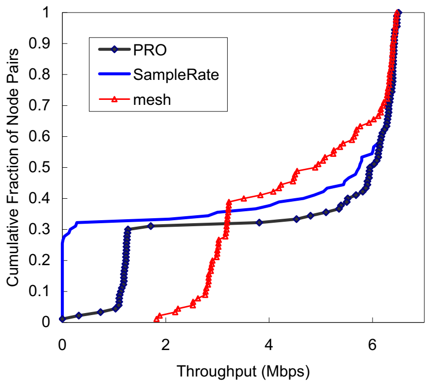


Figure 19: Throughput CDF of single session scenario in an office building

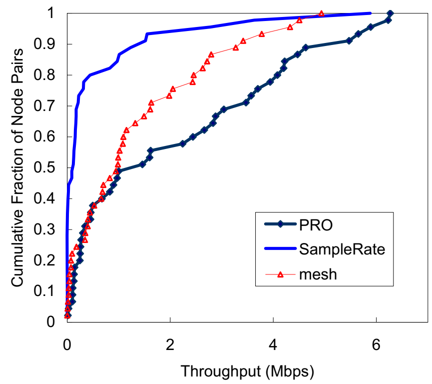


Figure 20: Throughput CDF of concurrent session scenario in an office building

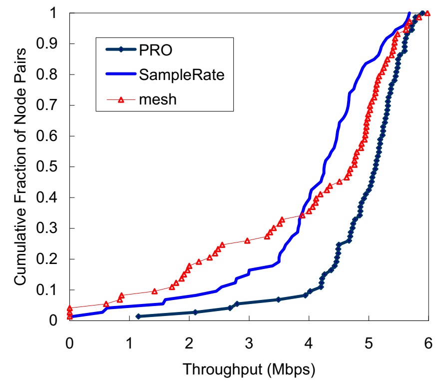


Figure 21: Throughput CDF of single session scenario in a student lounge

work capacity. Note that both the individual and aggregate throughputs are lower with SampleRate than with PRO.

The results in this section show PRO can indeed create unfairness relative to legacy 802.11 nodes, but that we can control the degree of unfairness through the selection of the CW_{min} window used by the relays. Our results also suggest that the degree of unfairness introduced by PRO is no worse than what is already present in 802.11 networks.

9. REAL-WORLD EVALUATION

To investigate how PRO performs in the real world, we conducted experiments in two indoor environments: an office building with hard partitions and a open student lounge where people passing through frequently. These experiments automatically account for all effects that are naturally present in deployed wireless networks, e.g. interference, noise, multipath fading, and shadowing. We first present 802.11b results in Sections 9.1 and 9.2 followed by 802.11g results using multi-rate PRO in Section 9.3.

9.1 Office Building

Our first experiment is conducted in an indoor office building with hard partitions. We placed ten laptops randomly in the building, as shown in Figure 18. The experiments were conducted during the night, when changes in the environment are relatively limited. In the first experiment, the laptops take turns as the source and send UDP packets to the other nine laptops one by one, resulting in ninety data points. During each of the runs, nodes other than the source and destination serve as relays. We also conducted the same experiment using SampleRate and a mesh network setup.

Figure 19 compares the cummulative distribution functions (CDFs) for the throughputs obtained using the three techniques. We can split the curves into three regions, each covering about one third of the sessions. The high throughput region contains sessions covering short distances. Sources can communicate with destinations successfully at the highest rate so all techniques achieve the maximum link capacity. The center throughput region contains source destination pairs that can communicate directly, albeit not always at the highest data rate. This region is likely to be representative of a typical infrastructure WLAN. In this case, PRO outperforms SampleRate which outperforms the mesh network setup. These results are consistent with the emulator results that use fading with a large K factor. The low throughput

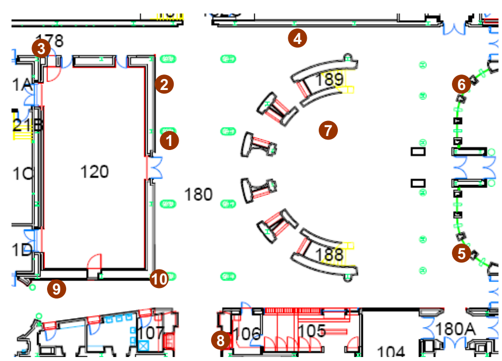


Figure 22: Floor plan of the student lounge

region contains sessions with distant sources and destinations, most of which are out of each other's communication range. In this case, SampleRate performs the worst because even the lowest transmit rate does not support communication. PRO's ability to relay ACKs allow it connect these out-of-range nodes. However, PRO has to rely on duplicate transmissions from the source to identify missing ACKs, which involves more overheads than the mesh network approach. In this scenario, support for dynamically switching between PRO and the mesh network approach described in Section 5.4 would be beneficial.

In the second experiment, we evaluate PRO with concurrent flows. Three source-destination pairs are randomly chosen every one minute and the test lasts 15 minutes resulting in 45 data points. Note that with concurrent transmissions a relay may serve multiple source-destination pairs, and destinations may serve as relays for other source-destination pairs. Figure 20 compares the throughput CDFs for SampleRate, and a mesh network setup. PRO has the best performance, followed by the mesh network, and SampleRate. SampleRate performs poorly for two reasons. First, SampleRate sometimes misjudges collisions as errors in high contention environments, which results in an overly conservative transmission rate. This problem can be severe in the presence of hidden terminals. Second, 802.11 provides equal channel access probability across contending stations. When rate adaption is used, poor channels will use a low transmit rate, using disproportionately more time than good channels. This can result in an inefficient use of the channel. PRO and the mesh network only use the highest rate so good sessions are not penalized.

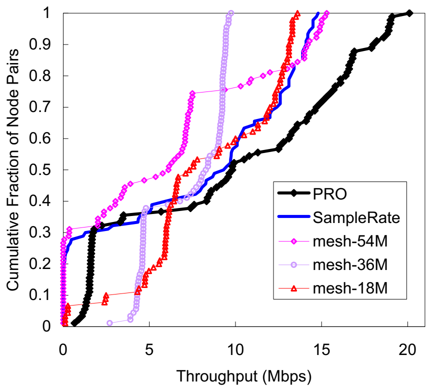


Figure 23: Throughput CDF of single session scenario in an office building (802.11g)

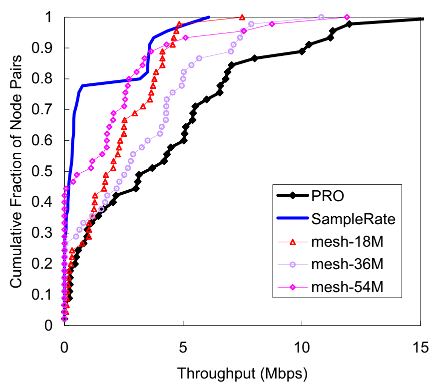


Figure 24: Throughput CDF of concurrent session scenario in an office building (802.11g)

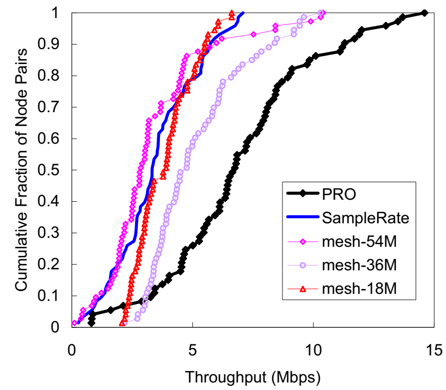


Figure 25: Throughput CDF of single session scenario in a student lounge (802.11g)

PRO consistantly outperforms the mesh network. We found this is to be the result of hidden terminals. When senders are hidden and their receivers are close (or there is a common receiver), collisions degrade the performance of the mesh network significantly. PRO alleviates the impact of hidden terminals in two ways. First, when two sources are hidden from each other and their transmissions keep colliding at a relay (or two close relays), other relays that are not involved in the hidden terminal can still successfully transmit the packet. Second, when two relays are hidden from each other and their transmissions collide at two close destinations, the online calibration process gradually filters out those hidden relays (as poor relays).

9.2 Student Lounge

We also conducted experiments in a student lounge during the day time when students come and go frequently. This creates a lot more movement in the environment. We again use ten laptops randomly placed in the lounge as shown in Figure 22. Each laptop takes turns as the source and sends UDP packets to the other nine laptops one by one.

The throughput results in Figure 21 shows that opportunistic retransmission offers significant benefits in this environment. The reason is that there is a lot of movement, resulting in significant small-scale fading that is a challenge for rate adaptation algorithms. As mentioned earlier, PRO does not need very accurate prediction of link quality, making it more robust in the presence of significant fading. Note that this result is consistent with the results obtained on the emulator testbed using fading with a small K factor. PRO also outperforms the mesh network based approach for the vast majority of the flows because this environment has few out-of-range node pairs.

9.3 802.11g with Multi-Rate PRO

We now present evaluation results for 802.11g nodes using the multi-rate PRO algorithm described in Section 7. Our implementation is built upon FlexMAC's "flexible mode", which allows the use of 802.11g transmit rates with 802.11b interframe spacings [20]. This is equivalent to an 802.11g station operating in the mixed 802.11b/g mode. We conducted experiments in the same real-world environments with the same setup as the 802.11b scenarios presented earlier in this section. Figures 23, 24, and 25 show the results for the single session and concurrent session scenarios in the office building and in the student lounge, respectively. Similar to the 802.11b results, multi-rate PRO, though not yet fully optimized, outperforms SampleRate and the mesh network based approach in both high contention and high fading environments, when few out-of-range node pairs are present.

10. RELATED WORK

The concept of opportunistic communication has been applied in many different contexts. Opportunistic retransmission takes advantage of packet reception outcomes that are random and unpredictable, similar to techniques such as opportunistic routing or opportunistic relaying. There are however significant differences:

Opportunistic routing in multi-hop wireless networks [4, 9, 19] improves the performance of static predetermined routes by determining the route as the packet moves through the network based on which nodes receive each transmission. The actual forwarding is done by the node closest to the destination. While opportunistic retransmission and opportunistic routing bear some similarity (i.e. exploiting multiple paths between the source and the destination), they are very different approaches. First, opportunistic retransmission applies to infrastructure networks so it is more generally applicable. Second, opportunistic routing requires a separate mechanism to propagate route information. Moreover, opportunistic routing is forced to use broadcast transmissions in order to enable receptions at multiple routers since it operates in the network layer. This constraint raises two issues. First, broadcasts messages are transmitted with basic rates in the link layer, which are generally overly conservative. Second, the additional gains from combining with rate adaptation are not available. In contrast, opportunistic retransmission is a link layer technique, so it automatically avoids these overheads. Finally, opportunistic retransmission does not affect (or may even decrease) packet latency and packet delivery order, while opportunistic routing often does increase latency and generate out-of-order deliveries in order to spread out scheduling and routing overheads, which is a problem for interactive applications.

The notion of *opportunistic repeating* for 802.11 WLANs is proposed in [2]. The idea is to have wireless links dynamically switch between direct transmission and two-hop mesh packet forwarding. The motivation for the work is replacing links that use low transmit rates, and are thus inefficient, by two hops that use high transmit rates. While both PRO and opportunistic repeating rely on intermediate nodes to forward packets, they differ in several facets. First, PRO explicitly considers fairness and provides a mean for controlling the degree of unfairness it introduces. Second, opportunistic repeating does not exploit per-packet opportunistic receptions at the destination and relays, but switches between two modes, similar to the switching algorithm described in Section 5.4. Finally, PRO uses multiple relays.

Recently, *opportunistic relaying* has been proposed as a practical scheme for cooperative diversity, in view of the fact that practical space-time codes for cooperative relay channels are still an open and challenging area of research [6, 22]. It relies on a set of cooperating relays that are willing to forward received information toward the destination. The challenge is to develop a protocol that selects the most appropriate relay to forward information toward the receiver. The scheme can use either digital relaying (decode and forward) or analog relaying (amplify and forward).

Opportunistic retransmission only uses relays that can fully decode the packets. From a functional perspective, opportunistic retransmission can be categorized as a lightweight, decode-and-forward opportunistic relaying mechanism. It however differs from opportunistic relaying in two aspects. First, in PRO, the destination does not combine the signals from the source and the relay, but tries to decode the information using either the direct signal or the relayed signal (in case that the direct signal is not decodable). This sacrifices some achievable rates but avoids the cost of additional receive hardware, so it is easy to deploy. Second, existing opportunistic relaying protocols require a RTS/CTS handshake to assess instantaneous channel conditions and/or to exchange the feedback of relay selection results [6, 8]. The use of RTS/CTS introduces extra overhead and delay. PRO avoids these overheads by leveraging channel reciprocity for link quality estimation.

Opportunistic communication has also been used in wireless sensor networks. For example, cluster-based forwarding (CBT) is proposed as extension to routing protocols [8]. CBT is similar to PRO but it involves several design issues. First, CBT uses a TDMA-based approach for exchanging the feedback of relay selection results. TDMA scheduling complicates the design and is not suitable for dynamic environments where relays join and leave frequently. Second, CBT only considers how good a node is to forward a packet, but it neglects how good a node is to function as a relay (i.e. the probability of overhearing a lost packet). PRO considers both in relay selection.

11. CONCLUSION

Opportunistic retransmission offers an effective means to improve throughput in wireless networks by exploiting overhearing relays to retransmit on behalf of the source upon failed transmissions. In this paper, we presented an efficient opportunistic retransmission protocol for IEEE 802.11 WLANs. Our protocol allows coexistence with legacy 802.11 stations. We have implemented PRO in the Madwifi driver for Atheros cards and conducted extensive experiments in both a controlled testbed and in the real world. Our results show that PRO boosts throughput in many wireless environments, especially in when there is significant contention for the channels, under fading, and with user mobility.

12. ACKNOWLEDGMENTS

We appreciate the constructive comments by our shepherd, Dr. Jitendra Padhye and the anonymous reviewers. This work was supported by NSF award CCR-0205266 and by the iCAST project.

13. REFERENCES

[1] Wireless Medium Access Control (MAC) and Physical Layer (PHY) specification: Medium Access Control (MAC) Quality of Service (QoS) Enhancements, IEEE Std. 802.11e-2005.

[2] P. Bahl, R. Chandra, P. Lee, V. Misra, J. Padhye, D. Rubenstein, and Y. Yu. Opportunistic Use of Client Repeaters to Improve Performance of WLANs. In *Proc. of ACM CoNEXT*, Dec. 2008.

[3] J. C. Bicket. *Bit-rate Selection in Wireless Networks*. Master's thesis, Massachusetts Institute of Technology, 2005.

[4] S. Biswas and R. Morris. ExOR: Opportunistic Multi-Hop Routing for Wireless Network. In *Proc. ACM SIGCOMM*, Sept. 2005.

[5] A. Bletsas, A. Khisti, D. Reed, and A. Lippman. A QoS-enabled MAC Architecture for Prioritized Service in IEEE 802.11 WLANs. In *Proc. IEEE GLOBECOM*, Dec. 2003.

[6] A. Bletsas, A. Khisti, D. Reed, and A. Lippman. A Simple Cooperative Diversity Method based on Network Path Selection. *IEEE Jornal on Selected Areas of Communication*, 24(3):659–672, March 2006.

[7] J. Camp and E. Knightly. Modulation Rate Adaptation in Urban and Vehicular Environments: Cross-Layer Implementation and Experimental Evaluation. In *Proc. ACM MOBICOM*, Sept. 2008.

[8] Q. Cao, T. Abdelzaher, T. He, and R. Kravets. Cluster-based Forwarding for Reliable End-to-End Delivery in Wireless Sensor Networks. In *Proc. IEEE INFOCOM*, May 2007.

[9] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading Structure for Randomness in Wireless Opportunistic Routing. In *Proc. ACM SIGCOMM*, Aug. 2007.

[10] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. In *Proc. ACM MOBICOM*, Sept. 2003.

[11] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *Proc. ACM MOBICOM*, Sept. 2004.

[12] G. Holland, N. Vaidya, and P. Bahl. A Rate Adaptive MAC Protocol for Multi-Hop Wireless Networkss. In *Proc. ACM MOBICOM*, Sept. 2001.

[13] S. Jain and S. R. Das. Exploiting Path Diversity in the Link Layer in Wireless Ad Hoc Networks. In *IEEE WoWMoM Symposium*, June 2005.

[14] A. Jardosh, K. Ramachandran, K. Almeroth, and E. Belding-Royer. Understanding Congestion in IEEE 802.11b Wireless Networks. In *Proc. of ACM SIGCOMM/USENIX IMC*, Oct. 2005.

[15] G. Judd and P. Steenkiste. Using Emulation to Understand and Improve Wireless Networks and Applications. In *Proc. of NSDI 2005*, May 2005.

[16] G. Judd and P. Steenkiste. Characterizing 802.11 Wireless Link Behavior. *Wireless Networks (WINET) Journal, Springer*, June 2008.

[17] G. Judd, X. Wang, and P. Steenkiste. Efficient Channel-aware Rate Adaptation in Dynamic Environments. In *Proc. ACM MOBISYS*, June 2008.

[18] C. Lo, W. Heath, and S. Vishwanath. Opportunistic Relay Selection with Limited Feedback. In *IEEE Vehicular Tech. Conference*, Sept. 2007.

[19] M. Lu, P. Steenkiste, and T. Chen. Video Transmission over Wireless Multihop Networks using Opportunistic Routing. In *Packet Video Workshop 2007*, Nov. 2007.

[20] M. Lu, P. Steenkiste, and T. Chen. Using Commodity Hardware Platform to Develop and Evaluate CSMA Protocols. In *Proc. of ACM WiNTECH*, Sept. 2008.

[21] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang. A Practical SNR-guided Rate Adaptation. In *Proc. IEEE Intern. Conf. on Computer Commun. (INFOCOM) (mini-symposium)*, Phoenix, Arizona, 2008.

[22] B. Zhao and M. Valenti. Practical Relay Networks: a Generalization of Hybrid-ARQ. *IEEE Jornal on Selected Areas of Communication*, 23(1):7–18, Jan. 2005.