The Importance of Being Overheard: Throughput Gains in Wireless Mesh Networks

Mikhail Afanasyev and Alex C. Snoeren
University of California, San Diego
{mafanasyev,snoeren}@cs.ucsd.edu

ABSTRACT

A flurry of recent work has focused on the performance gains that may be achieved by leveraging the broadcast nature of the wireless channel. In particular, researchers have observed that nodes other than the intended recipient of a packet may overhear the transmission in certain settings. Systems have been proposed to leverage this so-called overhearing phenomena by opportunistically adjusting forwarding paths, suppressing similar transmissions, and superimposing packet transmissions using network coding. The effectiveness of such approaches in practice depends greatly on the empirical overhearing rate, which is a function not only of the particular network and its environment, but also upon individual nodes' transmission rates.

Most existing opportunistic routing systems use a single, fixed bitrate throughout the network, leaving open significant opportunity for increased performance. We present *modrate*, a mechanism to jointly optimize rate selection and overhearing opportunities to maximize overall network throughput. We implement modrate in ExOR, an integrated routing and MAC protocol that leverages overhearing to improve bulk-data transfers, and compare its performance in a 48-node wireless mesh network testbed to ExOR, MORE, and traditional routing. While modrate increases the number of profitable overhearing instances in the network, we discover that ExOR extracts far less value from overhearing than might be expected. Instead, the majority of ExOR's performance improvement in many instances is due to its bulk-acknowledgment scheme.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design; C.2.2 [**Computer Communication Networks**]: Network Protocols

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Overhearing, 802.11 mesh networks

1. INTRODUCTION

Wireless mesh networks frequently provide connectivity in challenging environments, such as sparsely populated rural regions with extremely long distances between nodes [9] and dense, urban neighborhoods with a great deal of interference [2, 4]. The principal difficulty facing successful network operation in these deployments is channel variability: link quality can vary dramatically over time, forcing the network to constantly reevaluate the most efficient way in which to transmit packets across the mesh.

Early work focused on selecting the optimal bitrate for each link in the network. The goal of these systems was to send individual packets as efficiently as possible between intermediate hops: when channel quality is poor, nodes may employ low bitrates to ensure frame exchanges are successful, but they seek to transmit packets using the fastest bitrate the next hop can reliably decode as conditions improve. Despite attempts to select the optimal bitrate for an individual link, it is frequently the case that the chosen bitrate is overly conservative, in that the packet "overshoots" the next hop and is also successfully received at another node closer to the eventual destination. This phenomena is especially prevalent in networks with highly variable link quality, where senders are forced to be more conservative.

A flurry of recent work has focused on leveraging overhearing to improve network throughput [1, 6, 8, 16]. Some opportunistically benefit from serendipitous overhearing [1], while others deliberately seek out node configurations where packets are regularly overheard [16]. The systems that report the greatest throughput improvement for bulk data transfers, however, fundamentally redesign the way in which data is shipped from source to destination [6, 8]. In doing so, they not only alter the transport-layer behavior of nodes, but also change the way they use the MAC (e.g., by sending multicast instead of unicast packets, and turning off link-layer acknowledgments, among others).

Somewhat surprisingly, each of these previous systems has leveraged existing rate-adaptation techniques to determine the appropriate bitrate to employ for packet transmissions. Some simply fix the bitrate to a single value network wide [6, 8], while others employ traditional link-local optimization [1, 7]. In either case, the resulting transmissions are likely to be sub-optimal in the global sense. For example, while sending at 6 Mbps might be the best choice if one considers the throughput between two hops A and B on some path $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$, throttling down to 5.5 Mbps may enable 80% of A's packets to be overheard by node D, two hops further along the path. The latter is clearly a better choice if one seeks to optimize the throughput of the entire path from A to E.

The first contribution of this paper is a new rate-selection algorithm called *modrate* that seeks to jointly optimize individual link bitrate selections with network-wide overhearing opportunities. In particular, as opposed to selecting bitrates in a link-local fashion based only upon a packet's next hop, modrate selects the bitrate that minimizes a packet's expected number of transmissions along a path to its eventual destination assuming that any overhearing can be profitably exploited. We have integrated modrate with ExOR [6], one of the most effective systems currently available for leveraging overhearing, and deployed it on a 48-node indoor wireless mesh network testbed.

As our initial survey shows, our testbed presents ample opportunities for overhearing, and its prevalence varies noticeably with the particular bitrates employed. The performance of both ExOR and a more recent extension, MORE [8], vary substantially with bitrate, and most settings outperform traditional routing in our environment. We therefore expected modrate to provide significant further throughput improvement in ExOR. While modrate is able to increase ExOR's performance in some instances, the boost is surprisingly modest in many cases. Our detailed evaluation of the cause leads to the second contribution of this paper: we show that while proposed opportunistic algorithms—ExOR in particular—can provide tremendous performance improvement, in our environment at least, the vast majority of their gains come *not* from leveraging overhearing, but instead from a number of other substantial changes to the transfer protocol in the implementation.

Motivated by this observation we present a careful analysis of a spectrum of potential protocols on a smaller, 10-node controlled testbed, starting with Srcr, a state-of-the art traditional routing protocol that does not leverage overhearing [4], and incrementally applying changes to arrive at ExOR with modrate. Previous studies have compared only two points in this spectrum, typically traditional routing and their proposed protocol. By considering each modification individually, we discover that in many circumstances ExOR gains more from the relatively prosaic step of eliminating individual per-packet acknowledgments than from taking advantage of overhearing. This discrepancy is especially pronounced in networks with lossy links. Hence, Amdahl's Law explains modrate's limited improvement: even though modrate is able to increase the number of profitable instances of overhearing, it is fundamentally limited in the impact it can have on the overall throughput of ExOR.

While considerable work remains to be done to determine the generality of our findings, we believe the results may have significant implications. In particular, many researchers—ourselves included—may overestimate the ability of existing systems to effectively exploit overhearing in mesh networks. Conversely, significant gains can be extracted from far more banal protocol changes.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of related work, including a brief survey of mechanisms that have been proposed to select appropriate 802.11 bitrates and to leverage overhearing. We present our testbed environment in Section 3, and quantify the amount of latent overhearing present in the network. Section 4 describes the design and implementation of modrate. We evaluate the effectiveness of rate adaptation in Section 5 on a small, controlled testbed. Section 6 teases apart the performance gains from various protocol enhancements. Section 7 reports on results from a building-wide testbed. Finally, we summarize our conclusions in Section 8.

2. BACKGROUND & RELATED WORK

In order to place our contribution in context, we briefly survey related work in three distinct fields. First, we discuss existing 801.11 rate control algorithms. Second, we explore how traditional routing metrics consider rate selection in computing paths through the network. Finally, we detail the various ways in which researchers have proposed to harness overhearing, focusing particularly on ExOR.

2.1 Rate adaptation

Modern 802.11 PHY layers support multiple bit rates, ranging from 1–11 Mbps for 802.11b, 6–54 Mbps for 802.11a, and 1–54 Mbps for 802.11g. Because channel characteristics vary across space and time, an effective 802.11 sender will periodically reconsider the bitrate it employs. A large number of rate-adaptation techniques have been proposed in the literature [5, 14, 15, 20, 21] including several [5, 15] which have been deployed in commercial products. Each seeks the same goal, however: to optimize the goodput of the wireless link between sender and receiver.

Because the basic 802.11 standard does not provide for explicit feedback about channel quality at the receiver, senders are forced to estimate the optimal transmission rate through indirect means. The mechanism first deployed commercially, Auto Rate Fallback (ARF) [15], defaults to the highest bitrate and falls back to slower speeds if it fails to receive a link-layer acknowledgment for a transmitted frame. ARF speeds back up after a string of successive successful packet transmissions. Researchers have observed, however, that 802.11's link-layer retransmission mechanism may mask frame losses, causing ARF to over-estimate the optimal bit rate.

As an alternative, Receiver-Based Auto Rate (RBAR) [14] proposes to have the receiver report received channel quality in RTS packets, allowing the sender to dynamically adjust transmission rates according to current channel conditions. This presumes that CTS signal-to-noise ratios are effective predictors of frame-exchange success, however, which Bicket found was not always the case [5]. Instead, he proposes to send periodic probe packets at speeds higher than the one currently employed and keep track of their relative success rates in a protocol he calls SampleRate, which has been widely deployed in the MadWifi driver and employed by follow-on research projects [4, 6, 8]. Recent results, however, have shown that SampleRate can be too conservative in certain cases; indeed its poor performance has lead to its deprecation within the MadWifi driver. Instead, Starsky *et al.* have proposed combining feedback from the RTS/CTS exchange with loss-rate information gathered at the current rate into a system they call Robust Rate Adaptation Algorithm (RRAA) [21].

While researchers have proposed making opportunistic use of the link by sending packets in rapid succession when conditions allow higher transmission rates [20], none of the existing schemes rate-adaptation schemes consider what impact that choice will have on route selection or global network throughput.

2.2 Routing metrics

In order to provide end-to-end connectivity, a mesh network must compute routes between any two node pairs. In general, routing protocols attempt to compute paths that minimize some cost metric. The most natural metric, commonly used in wireline networks, is hop count. While straightforward to compute, hop count favors paths consisting of fewer, longer hops, which tend to be less reliable than shorter hops. Instead, the Roofnet urban mesh network introduced ETX, or expected transmission count, which accounts for the retransmissions that are likely to be required on less-reliable links [12]. Yet, if a particular link employs a lower bitrate which is more likely to succeed, it is also more likely to be included on a path despite other, potentially higher-throughput alternatives. To address this deficiency, Roofnet replaced ETX with ETT, or expected transmission time, that incorporates link rates in addition to retry attempts into the link cost [4].

There are multiple ways to determine ETT; Roofnet eschews collecting explicit samples between each pair of nodes [13] in favor of conducting a synchronized, network-wide survey using broadcast packets. To forward packets, Roofnet employs a source-routing protocol known as Srcr that calculates paths using ETT. When an individual node transmits a packet, however, it employs SampleRate to select the bitrate; hence, the actual transmission speeds employed may deviate from those anticipated by the ETT calculation.

2.3 Overhearing

No matter what link rate and next-hop are selected, the broadcast nature of wireless networks leads to the possibility—indeed, the probability—that nodes other than the intended recipient overhear the transmission. Researchers have proposed a number of techniques to harness overhearing to varying success.

2.3.1 Single-path routing

A number of efforts have examined methods for selecting routes that explicitly leverage opportunistic forwarding opportunities in multi-hop networks [17, 19]. We previously observed that even if routes are not deliberately selected to enhance overhearing, nodes further a long a path that overhear a transmission can squelch subsequent forwarding attempts by upstream nodes [1]. While broadly applicable, these techniques provide modest throughput gains.

2.3.2 Batching

Far greater throughput gains can be achieved if the network is designed from the ground up to leverage overhearing. In particular, both ExOR [6] and the more recent MORE [8] define new, bulk-transfer transport protocols that leverage overhearing to dramatically increase goodput. While effective at achieving high throughput, both systems are unfortunately incompatible with traditional transport protocols like TCP and latency-sensitive applications.

ExOR is a bulk-data protocol: rather than transmitting individual packets, it transfers *batches* of packets. The source gathers together the set of packets destined for a particular destination and transmits them all at once, along with a precomputed *forwarder list* enumerating any likely[1] intermediate nodes between the source and destination. The source prioritizes the forwarder list based upon its estimation of their proximity to the destination (computed using the ETX metric, described above). Any nodes contained within the forwarding list that successfully receive packets transmitted by the sender buffer them until the batch is completed.

Once the sender has finished sending the batch, the receiving node with highest priority begins forwarding any packets it has buffered. The node annotates this so-called batch *fragment* with its estimation of the highest-priority node to have received each packet in the batch, called a *batch map*. Subsequently, each node in the forwarding list takes its turn sending any packets not previously acknowledged in another's batch map until the destination has received at least 90% of of the packets in the batch. The remainder of the packets are forwarded using traditional routing.

One of the most challenging aspects of implementing ExOR is ensuring each forwarder transmits its batch fragment at the appropriate time. If transmissions are uncoordinated, fragments will collide, eliminating any potential benefits. The ExOR design requires each node to keep a transmission timer, as well as to record the fragment numbers being transmitted to estimate the effective channel rate and predict when individual forwarders will complete their fragment transmissions.

2.3.3 Network coding

MORE's operation is similar, but it uses random network coding to avoid the need for ExOR's scheduler. Mostly by increasing opportunities for spatial reuse, MORE achieves unicast throughput 22–45% higher than ExOR's [8]. While not included in the conference publication, Chachulski develops an extension to the ETX metric, called EOTX (expected opportunistic transmission count), in his Masters' thesis [7] that considers the potential decrease in expected hop count due to the use of ExOR or MORE. While theoretically more appropriate than ETX, practical experimentation indicates that EOTX provides negligible performance improvement over ETX when used in conjunction with MORE.

[1]In order to keep the list size manageable in dense networks, ExOR prunes nodes expected to overhear less than 10% of the packets.

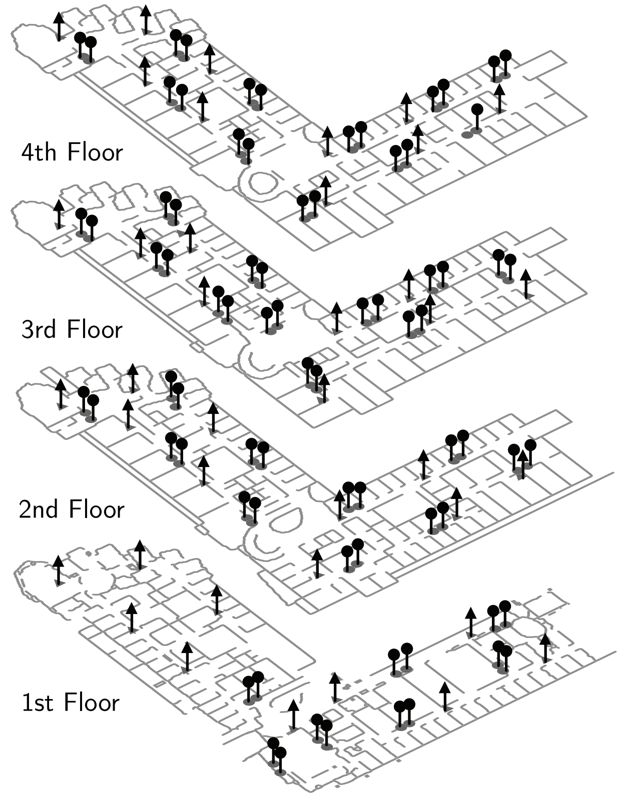


Figure 1: The UC San Diego Jigsaw wireless testbed. We use the Jigsaw mesh nodes (depicted as circles); production 802.11 access in the building is provided by infrastructure-mode access points (shown as triangles).

Another notable approach based on network coding, COPE [16], does not target opportunistic overhearing in quite the same fashion as the schemes described previously. Instead, it takes advantage of the fact that a sender in the middle of a three-node chain can be heard by both of the nearby nodes during a single transmission, allowing bidirectional traffic to be sent using three transmissions instead of four. While we believe that COPE could also benefit from joint bitrate adaptation, we leave the application of modrate to COPE as future work.

3. OVERHEARING'S SIREN SONG

The effectiveness of any overhearing scheme depends greatly on the channel characteristics of the network. Hence, we first seek to quantify the latent overhearing in our testbed.

3.1 Jigsaw testbed

In order to determine the impact of transmission rate on delivery ratio in a practical environment, we use the Jigsaw wireless testbed

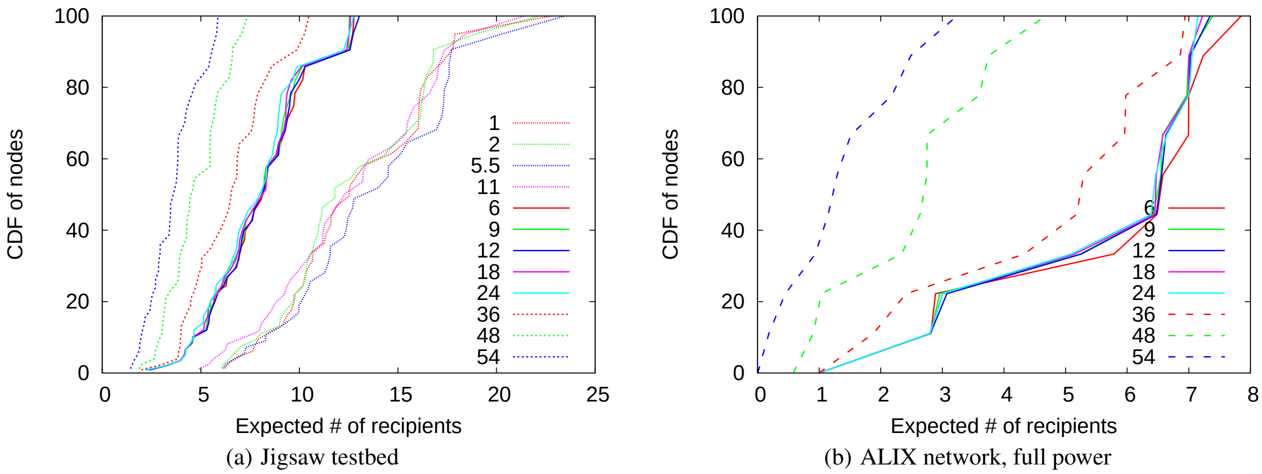


Figure 2: CDF of the number of the average number of recipients per packet as a function of bitrate.

and analysis engine at the University of California, San Diego [10, 11]. Figure 1 shows the layout of the testbed nodes (reproduced with permission from [11, Fig. 1]). The testbed consists of 48 embedded Linux nodes, each outfitted with two Atheros 802.11a/b/g radios. The testbed is deployed within the Computer Science and Engineering building on the UC San Diego campus and spans four and a half floors (the half-covered basement is not shown in the figure) covering approximately 150,000 square feet of floor space and one million cubic feet of volume. In addition to human inhabitants, the building contains thousands of workstations and a large variety of electronics operating in the same 2.4 and 5 GHz unlicensed frequency bands as 802.11, resulting in highly variable channel quality in different portions of the building and during different times of the day [11]. In our work, we use only one radio per node, and place the device into 802.11g mode. While the use of 802.11a would decrease interference from the building's production 802.11g network, increased fading in the 5 GHz band significantly decreases connectivity.

The Jigsaw analysis engine is particularly appropriate for our needs, since it is able to tightly time synchronize traces collected at multiple nodes and precisely determine which received frames are actually identical—i.e., a single transmission that was successfully decoded at multiple receivers—and which are simply duplicates (such as those that result from link-layer retransmission). The infrastructure also automatically records the RSSI and any associated hardware errors reported along with the received frame.

3.2 Channel quality

Overhearing varies as a function of various channel characteristics including noise, fading, and signal attenuation; hence, before proceeding we seek to quantity number of testbed nodes that can successfully decode packets sent by any other node in the testbed. Obviously, this number depends not only on the channel conditions, but also on the transmitter's power and selected bitrate. For all experiments in this paper, we fix transmit power to the maximum supported by the 802.11 devices in use (nominally 100 mW).

To measure the performance of the network, we conduct a network-wide link survey similar to that conducted by the Roofnet researchers [3]. In particular, we fix all nodes to the same 802.11 channel (eleven in these experiments) and set them to listen in promiscuous mode. To reduce variance, we conduct our experiments during the night.

In our initial experiments, we implemented the Roofnet procedure, iterating through each of the nodes in the network in the following fashion. At each node, we transmitted one thousand 1,500-byte packets back-to-back at a particular bitrate. We cycled through each of the twelve available 802.11g bitrates in order before moving on to the next node. The entire process takes roughly ten minutes. Once the transmission phase is complete, we submit the traces to the Jigsaw analysis engine to determine how many stations received each individual frame. The analysis takes an additional ten minutes, during which the infrastructure cannot conduct probe experiments.

We discovered, however, that this measurement technique can be highly inaccurate in our environment. In particular, a moderate-length burst of broadband interference can completely distort measurements for one or more links. Thus, we alter our survey procedure to split transmissions into groups: We divide the 1,000 packets that each node transmits into 10 groups. Then, at each node, we transmit 100 1,500-byte packets back-to-back at a particular bitrate. We cycle through every node in the system and then move to next bitrate. Once all bitrates are done, we repeat the whole process 10 times until every station has transmitted 1,000 packets at each bitrate. In addition to spreading the impact of broadband interference, this method also allows us to estimate short-term variations in link quality. We calculate the standard deviation of the reception rate of each link. In our experience, the deviation is an important parameter of the link, staying relatively consistent over each measurement. For a typical link with 0.70 transmission probability, we see deviations that range from 0.01 to 0.30.

Figure 2(a) shows the average number of recipients of a transmission as a CDF over each of the nodes in the testbed. In our testbed, the bitrate has little impact for 802.11b encodings (1, 2, 5.5, and 11 Mbps): we see that the curves for each of these speeds are very similar. In contrast, 802.11g encodings show markedly smaller reception ranges in general, and significantly different reception rates at the high end (i.e., 54, 48, 36 Mbps). For example, switching between 54 and 48 Mbps adds one additional recipient to the median node, while dropping all the way down to 14 Mbps adds four additional receivers on average, and up to ten in the best case. The lower rates on the other hand, perform almost identically to each other. Thus, it seems likely that decreasing the bitrate to increasing overhearing opportunities may be a fruitful tradeoff in many cases.

Figure 3: ALIX network map. All nodes are located on the third floor.

3.3 ALIX testbed

Due to the poor connectivity of the Jigsaw nodes when used in the 5 GHz band, we also employ a separate, 10-node testbed of ALIX nodes from PC Engines using 802.11a radios. The ALIX nodes were distributed around the third floor of the building as shown in Figure 3. Figure 2(b) repeats the same survey experiment as before, except on the ALIX testbed. The results are qualitatively similar (which is expected given that 802.11a and 802.11g use identical modulation schemes) although the absolute number of nodes receiving a single transmission is much lower due to the smaller size of the testbed, and there is far greater separation between the three high 802.11a speeds and the remaining ones.

We can vary the connectivity of the ALIX testbed by altering the transmit power. Figures 4(a) and 4(b) compare the transmission range of 24 and 54 Mbit link speeds across a range of channel powers. We conduct most of our experiments at power level 30, but return to explore the implications of higher powers in Section 6.5.

4. MODRATE

Existing batch-based opportunistic routing algorithms use a single, fixed rate for all nodes in the network—1 Mbps in the original ExOR work, and 11 Mbps for MORE (although Chachulski *et al.* also publish results for ExOR at 11 Mbps)—and defer issues of bitrate selection to future work [6, 8]. Here, we consider how one might select more efficient bitrates to improve throughput when possible. We begin by considering the case of an 802.11b network, as used in previous work, and then present *modrate*, an approach better suited for modern, 802.11a/b/g networks.

4.1 Fixed range

From the previous section, we see that all of the 802.11b rates provide approximately the same range in our testbeds, so if we consider an 802.11b-only transmitter, it likely suffices to select the bitrate for each node independently—as in traditional routing algorithms [5]—since the transmitting node's choice is unlikely to have a significant impact on the set of forwarder nodes that will receive the batch fragment. Indeed, it appears even for 802.11g the same can be said for most speeds—all but the highest three, in fact, when transmitting at the highest power in our test bed. In other words, each transmitter can disregard the presence (or absence) of overhearing, and focus on the natural goal of selecting the bitrate that minimizes the remaining expected transmission time (ETT) of the batch fragment to its ultimate destination.

Happily, this is the same goal in traditional routing: Roofnet's Srcr routing protocol [4] selects a shortest path in terms of ETT presuming each node transmits to the next hop at its optimal bitrate. In fact, ExOR uses ETT[2] to determine the priority order of the forwarding list, so it will automatically incorporate any improvements due to bitrate selection into its forwarding algorithm. Extending the notation of Chachulski *et al.* [8], let ϵ_{ij}^r denote the the expected loss probability when node i transmits to note j at rate r. If we denote the time taken to transmit a packet at rate r as $T(r)$ (a constant value regardless of the nodes in question), we can write

$$ETT_{ij}^r = \frac{T(r)}{1-\epsilon_{ij}^r}.$$

Because ExOR only transmits packets for a single destination in any given batch, a node can consider each batch fragment transmission independently. In particular, for a batch fragment originating at s destined to node d, forwarding node i selects the bitrate as follows. Assume node j is the next hop on the optimal Srcr-computed route from i to d. (Note that, due to overhearing, i may not have been on the original Srcr route from s to d.) Then, i selects a bitrate r that minimizes ETT_{ij}^r:

$$r(i,j) = \underset{r \in Rates_i}{\arg\min} \left(\frac{T(r)}{1-\epsilon_{ij}^r} \right) \tag{1}$$

where $Rates_j$ is the set of bitrates available at node i.

4.2 Variable reception range

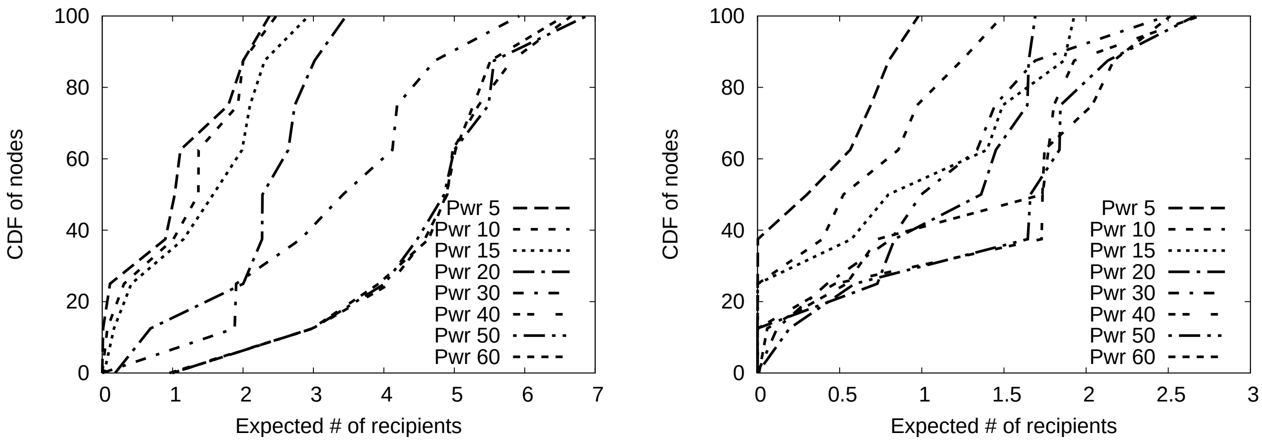
Modern systems typically use 802.11a/b/g radios, however, which have a direct correlation between transmission rate and average reception range. Thus, it is important to consider the potential impact of decreased overhearing opportunities when choosing an appropriate bitrate. We propose a rate-selection algorithm called *modrate* that jointly optimizes next-hop throughput and overhearing prevalence. Said another way, instead of trying to optimize for the expected single (Srcr) path as above, the rate instead is selected to minimize the expected transmission time over all useful paths including those that arise from overhearing.

In ExOR, a packet could be received at multiple destinations, but will be processed first by the destination with lowest ETT to the destination; to ease discussion we order all nodes in terms of their ETT to d, $s \geq i > j \geq d = 0$, Now, rather than adjusting the bitrate in view of just the next Srcr hop, we seek to consider the bitrate in view of the furthest (i.e., closest to d) recipient. If we define ρ_{ij}^r as probability that the furthest recipient of the packet sent from i at rate r will be j, we can compute the optimal bitrate r^* as

$$r^*(i) = \underset{r \in Rates_i}{\arg\min} \left(\sum_{j<i} \frac{T(r)}{\rho_{ij}^r} \right) \tag{2}$$

How could we calculate ρ_{ij}^r? One way is to assume that all transmission probabilities are independent, a frequent assumption in the

[2]ExOR actually uses ETX, but when every node in the network uses the same bitrate, ETT and ETX are equivalent.



(a) 24 MBit/sec (b) 54 MBit/sec

Figure 4: Expected number of recipients as a function of transmit power, ALIX network.

literature [8, 18]. Then, we just need to calculate the probability that a transmission would be received by j and not by any $k < j$:

$$\rho_{ij}^r = (1 - \epsilon_{ij}^r) \prod_{k<j} \epsilon_{ik}^r.$$

An alternative method is to not rely on independence, and instead measure probability of reception for all possible sets of receivers. Then, ρ_{ij}^r can be computed directly. We adopt the latter approach in our evaluation.

5. EVALUATION

This section explores the performance of various forms of rate control by comparing them to unmodified ExOR, as well as traditional, single-path routing and the MORE network coding scheme.

5.1 Testbed setup

To facilitate controlled experimentation, we conduct our initial tests on the 10-node ALIX testbed described earlier; we consider the performance of the Jigsaw testbed in Section 7. We drive the experiments with a centralized controller that has wired connectivity to each node in the network. We begin experiments by collecting the transmission probability data as described in Section 3.2 in order to produce appropriate routing and speed information, which we calculate using the set of algorithms from Section 4. This information is then communicated to the stations; thus, the stations themselves do not run any routing code, ensuring that all protocols operate with the same routes.

When conducting experiments comparing various protocols, we run all the protocols under test in sequence at each pair of source/destination nodes, before moving to the next pair. By doing so, we roughly equalize any impact of out-of-date delivery probabilities. Additionally, for long-running experiments, we update our estimates of the transmission probabilities and re-calculate routes every twenty minutes.

In order to facilitate direct comparisons, our experimental methodology largely follows those of the original ExOR [6] and MORE [8] papers, although with several slight differences. For each source/destination pair, we transfer a 1.5-megabyte file, consisting of 10 batches of 100 packets, each containing 1,500 bytes (c.f. 1,024 in the original ExOR paper) of payload. As is customary, we do not implement traditional routing of the final 10%; instead, we stop and report the throughput when the destination has received 90% of the packets in each batch. Thus, our experiments result in ten separate transmission times, each corresponding to the successful reception of at least 90% of a 150-KB file chunk.

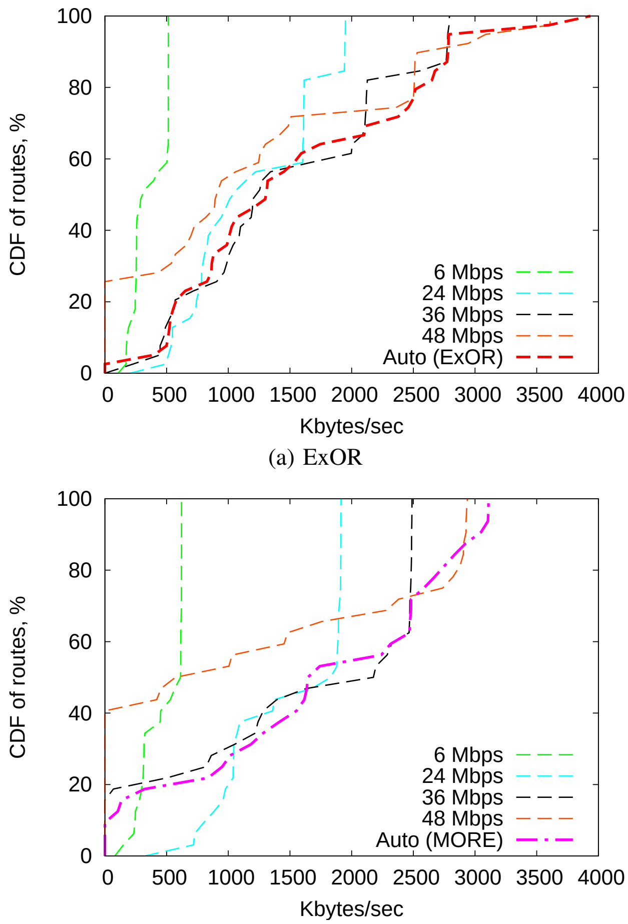
5.1.1 Traditional routing

As a baseline, we measure the throughput of traditional, single-path routing that employs both link-layer and end-to-end acknowledgments to ensure reliable delivery. In order to evaluate the most prevalent scenario in today's wireless networks—TCP data being sent over a single, rate-adapting path—we implement a simple Srcr forwarder. Our Srcr forwarder uses the link probabilities calculated by our measurement procedure and selects routes using a modified ETT metric that accounts for asymmetric links [1]. We assume that ACKs are always sent at the lowest speed for the 802.11 protocol in use (1 or 6 Mbps). This is a Click-based system which forwards all packets between two hosts along a predefined path, as provided by the experiment controller. We use the regular Linux 2.4 kernel TCP stack without modifications, and the `ttcp` application to measure the time it takes to transfer 1 megabyte of data. We refer to this mechanism in all of our results as 'trad-TCP.'

A multi-hop route might have both large latency and high losses. Those factors will interfere with TCP and prevent the window from becoming too large. Thus, non-TCP data transfer specifically optimized for the routing protocol will show better performance.

5.1.2 ExOR implementation

We were unable to obtain the original ExOR implementation, so were forced to reimplement it. Because we are unsure whether we were able to faithfully replicate the exact behavior of the transmission timer, we instead implement a scheduling "oracle" within the control server: Once a forwarding node is done transmitting a batch fragment, it notifies the control server over the wired network. The server then notifies the next node in the batch's forwarding list to begin transmission. Should that node not have any remaining packets to send, it may send a set of empty packets to propagate the batch map; regardless, it notifies the server when finished. Communication with the scheduling oracle takes time, so each station keeps track of how long it spent transmitting the batch fragment. Once the batch is successfully received at the destination, all round times are added together to get the actual transmission time without oracle communication overhead.



(a) ExOR

(b) MORE

Figure 5: Throughput of ExOR and MORE with automatic and various fixed rate selections. ALIX network with full power.

5.1.3 MORE module

We implement the MORE algorithm using the publicly available MORE source code. Since the implementation is completely separate from our ExOR code base, it does not use our experiment controller to calculate routes. Instead, we pass the reception probabilities from our surveys directly to the MORE implementation. All stations transmit simultaneously, without a central schedule (so it does not leverage our oracle scheduler, either), meaning that long paths could either perform better (due to spatial reuse) or worse (due to hidden terminal problems) than ExOR.

It is worth noting that MORE is substantially more CPU-intensive than any of the other protocols we evaluate. Even with data calculations disabled (i.e., only innovativeness verification was done), it still required more CPU power than the Jigsaw testbed can provide. Thus, we report MORE results only for the ALIX testbed.

5.2 Overhearing-oblivious rate selection

To begin, we consider the performance of ExOR as originally described by Biswas and Morris [6]. In particular, we assume that all nodes in the network use a single, fixed speed. Figure 2(b) suggests that the performance in the ALIX testbed with bit rates less than 24 Mbps are likely to be gated by link speeds rather than reception rates. Increasing speed beyond 24 Mbps, however, seems likely to markedly decrease the degree of connectivity in the network, potentially harming performance.

Figure 5(a) plots the performance of four fixed speeds, 6, 24, 36, and 48 Mbps on the ALIX testbed when all nodes transmit at maximum (60) power, roughly equivalent to 18 dBm. Recall that throughput is the total number of bytes delivered over 10 independent batches divided by the cumulative time required. For all of the graphs in this section, we report on the performance of 40 randomly selected node pairs among the 100 possible combinations. We bias the 40 routes to include longer-hop paths if possible, as one-hop paths tend to be uninteresting. None of our paths are longer than four hops. To select the 40 random paths, we first select up to 10 paths of each length—four, three, two, and one hops—and then fill in the remainder with randomly selected paths if we do not have enough of a particular length.

While performance generally improves with higher link speeds, the network becomes disconnected at 48 Mbps and no route exists for 11 of the selected route pairs; this phenomenon is even more pronounced at 54 Mbps. The globally optimal rate will obviously vary from network to network, and likely even over time. Instead, we see that an automatic rate assignment that selects the locally optimal speed for each link (neglecting overhearing potential) as specified in Equation 1 generally outperforms any fixed speed selection. We refer to this enhanced, automatic-rate-assignment ExOR implementation as 'ExOR' in all subsequent graphs.

Figure 5(b) shows the results a similar experiment for the MORE protocol. As originally described, MORE uses a single, fixed link speed for all nodes in the network. The publicly available implementation, however, selects a link-local optimal speed based upon the ETX metric in a manner similar to Equation 1 [7]. We use this improved MORE implementation in the remainder of the paper, and refer to it in the graphs as simply 'MORE.'

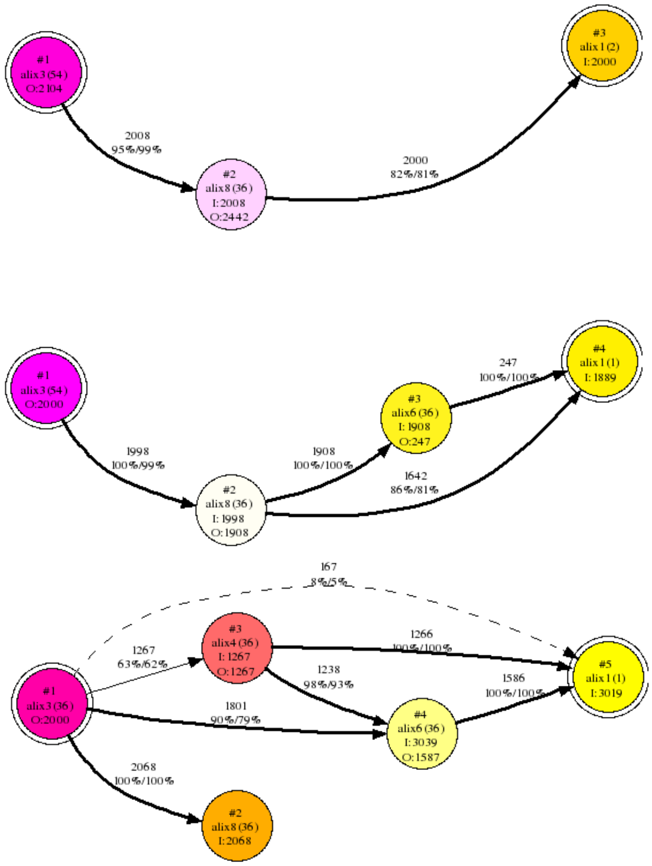
5.3 Modrate

We now consider the additional performance gains from considering the impact of link rates on overhearing opportunities. In particular, we enhance ExOR with the modrate algorithm described in Equation 2 and conduct a second experiment on the ALIX testbed at a power level of 30.

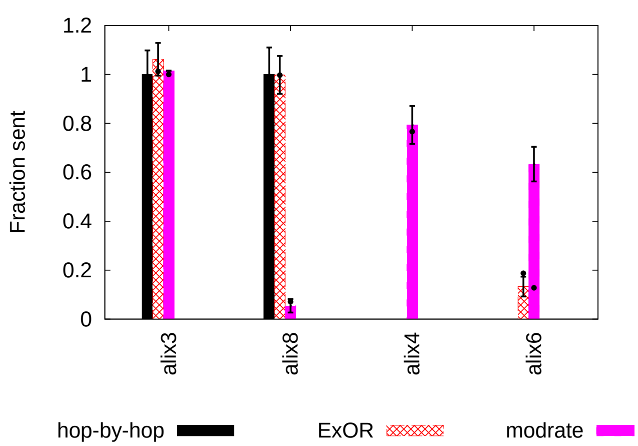
5.3.1 An example route

Figure 6(a) diagrams one particular route where modrate dramatically changes the forwarding behavior. The top portion shows a two-hop Srcr route from *alix3* to *alix1* that uses *alix8* as an intermediary when nodes transmit at full power; *alix3* transmits at 54 Mbps, while *alix8* selects 36 Mbps. In addition to link speed, each node is annotated with the number of packets it transmits (O) and receives (I). The links are labeled with both the number of packets successfully transferred as well as the experimental and predicted (by the survey) reception rate. The middle portion shows how ExOR uses the route, leveraging overhearing by node *alix6* to assist with packets on the second hop. In one particular batch, *alix6* overhears all of the packets transmitted by *alix8*, and is able to deliver 247 of them to *alix1*, saving retransmissions.

Finally, the bottom portion of the figure shows how modrate enhances overhearing by decreasing the transmission speeds of *alix3* (from 54 to 36 Mbps). By doing so, it introduces three new overhearing opportunities: First, the destination is able to directly receive packets approximately 8% of the time. Second, *alix4* and *alix6*, which are closer to the destination than *alix8*, are now able to overhear transmissions. In fact, between the two of these nodes, they are able to forward all of the packets to the destination, freeing the original intermediate hop, *alix8*, from forwarding any packets at all in this particular batch.



(a)



(b)

Figure 6: An example route using different algorithms.

The same effect is presented in another format in Figure 6(b), which depicts the same transfer from *alix1* to *alix3*. The y axis indicates the number of packets a station transmits normalized to the total number of packets transmitted along the route (although we do not plot link-layer transmissions used by the 'hop-by-hop' protocol). Node *alix3* is the source, and therefore has to transmit all packets at least once. The four transmitting nodes are laid out along the x axis, ordered in increasing proximity to the destination. Bars and error bars correspond to the average and standard deviation among all 10 batches, while dots indicate the performance predicted by the measurements.

5.3.2 Network-wide performance

Figure 7(a) plots throughput in the same fashion as Figure 5, comparing against ExOR, MORE, and traditional routing. Despite some significant changes in speed selections, the overall difference in performance between ExOR and modrate is slight. Figure 7(b) accentuates the differences by plotting the per-route throughput normalized to that of ExOR; a positive difference means that the performance is better than ExOR, and negative implies less throughput.

In theory, the performance of modrate should be strictly better than ExOR, but some variance is to be expected in practice due to time-varying delivery probabilities, and has been reported many times in the literature [8, 21]. In this experiment, modrate manages to equal or best the link-local scheme on all but 10% of the routes, and is rarely more than 10% worse. In this configuration, modrate provides limited benefit for the vast majority of routes, but brings significant improvement in around 15% of paths. This is easily explained by observing that modrate degenerates to the link-local scheme in the case of one-hop routes; even for longer routes, modrate and ExOR select identical rates 62% of the time. It is impossible to tell from the CDFs, however, precisely which routes are seeing improvement. Figure 8 presents 15 representative routes sorted according to their length and performance under traditional routing. Error bars report the standard deviation of the 10 constituent batches. We see that modrate provides performance increases in many of the two- and three-hop cases, but—as expected—none of the one-hop paths. Interestingly, MORE outperforms both schemes in 9 cases, but underperforms in the remainder, failing completely in two cases.

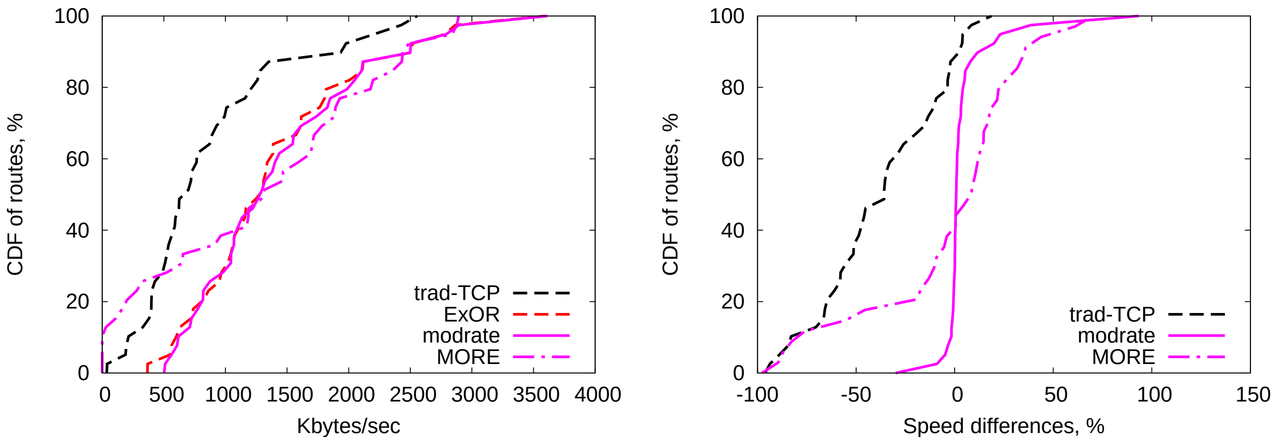
6. OVERHEARING'S ROLE

While modrate functions as expected, we were initially surprised by its modest gains given the dramatic differences in reception ranges shown in Figure 2. In particular, modrate is often able to significantly increase overhearing opportunities as shown in Figure 6, yet throughput gains are limited. Attempting to "debug" this situation leads to the second major contribution of our work, namely uncovering the reasons behind ExOR and MORE's performance. As the graphs will show, overhearing plays a relatively minor role.

In order to attribute performance gains to various aspects of the ExOR protocol, we have implemented three simplified versions of ExOR, each with a subset of ExOR's features disabled. Figure 9(a) compares the throughput of each of these versions to traditional routing, and Figure 9(b) once again shows the per-route performance relative to ExOR. Note that these graphs use the same dataset as Figure 7, so the 'trad-TCP' and 'ExOR' lines are unchanged. We describe these three versions and their relative performance in increasing levels of sophistication (and performance).

6.1 Bulk transport

Perhaps the most fundamental aspect of ExOR (and MORE) is its batch structure. Rather than transmitting packets as a stream (or window as in TCP), ExOR uses an explicit batch construct, where each node transmits an entire batch at a time before pausing to allow downstream nodes to forward them. We implement this functionality on top of traditional routing with link-level acknowledgments. In this mode, the batch map is not used (but for better comparison, it is still included as an overhead). Instead, each station transmits all packets it has once. Packets are sent in 802.11 unicast mode (as opposed to ExOR's usual broadcast), so link-level retransmissions may occur on lossy links, up to 10 times in our configuration. We note that this—not our 'trad-TCP' line—is what the ExOR paper calls 'Srcr'.



(a) CDF of path throughputs.

(b) Per-path throughput relative to ExOR.

Figure 7: The performance of modrate compared to ExOR, MORE, and traditional routing. ALIX network, power 30.

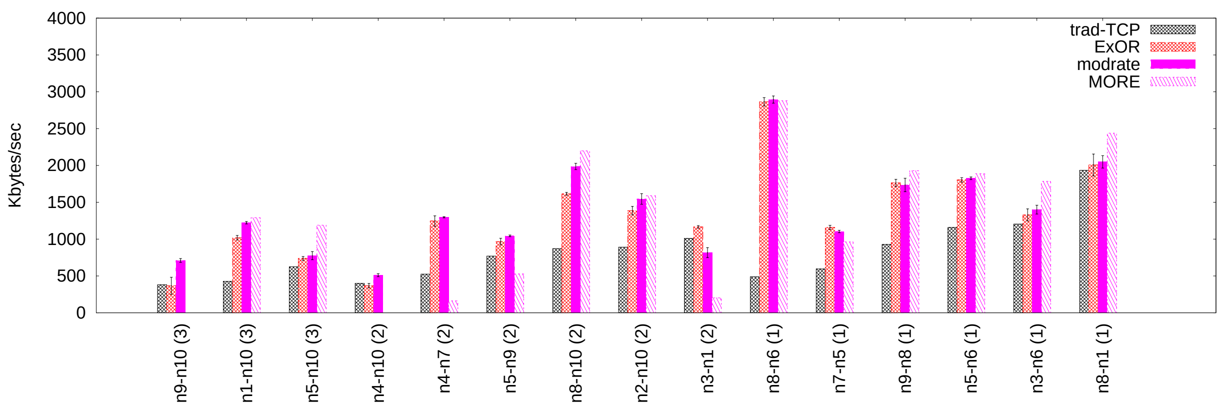


Figure 8: Path throughput for 15 representative routes. ALIX network, power 30.

It is frequently observed that TCP's back-off behavior is not ideal in wireless mesh networks. Hence, one might expect that bulk transfer, even operating on exactly the same routes at the same speeds, would perform better. Indeed, the simple bulk-transfer variant, labeled 'hop-by-hop' in the graphs, significantly out-performs 'trad-TCP', on average constituting more than 50% of ExOR's improvement. Interestingly, in almost 20% of cases, it out-performs ExOR.

6.2 Group acknowledgments

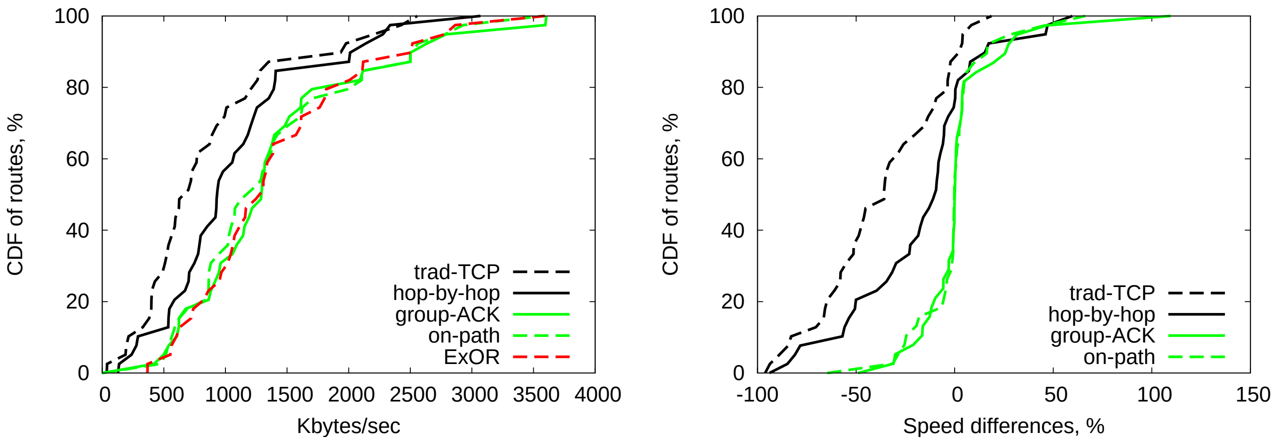
For a protocol transmitting batches at a time, it is natural to consider getting rid of individual packet acknowledgments in favor of bulk or group acknowledgments. In particular, instead of waiting for a link-level ACK after every frame, a node can send a single, combined transport-layer ACK at the end of transfer. Indeed, this is precisely what ExOR does with its batch maps. Group acknowledgments increase the latency of retransmissions, but latency is not a figure of merit for ExOR or the other protocols we study.

We have implemented a group acknowledgment scheme by simply disabling overhearing in ExOR. In particular, a node will only accept packets transmitted by the previous hop according to the underlying Srcr route. This algorithm is labeled 'group-ACK' in the graphs. We observe that 'group-ACK' is likely to perform well on low-loss links—because no time is wasted on superfluous link-level ACKs—and asymmetric links with lossy ACK channels. Given the significant improvement over the 'hop-by-hop' line in this configuration, we conjecture one or both of these instances occur frequently. We ascribe the small number of routes where 'hop-by-hop' outperforms 'group-ACK' to experimental variation.

6.3 On-path overhearing

Overhearing can be classified into two types: overhearing by nodes on the traditional route from source to destination, and incidental overhearing by nodes that would not be involved in traditional forwarding. While ExOR and MORE both take advantage of the latter, the former is easier to build into existing protocols, as the RTS-ID system showed [1]. We evaluate the effectiveness of strictly on-path overhearing by restricting ExOR's forwarder list to include nodes only on the Srcr path—as opposed to any node that is predicted to overhear at least 10% of the transmissions.

Forwarding with this restricted form of overhearing is labeled 'on-path' in the graphs. In our implementation, there can be no overhead with respect to group acknowledgments (any deviations are once again attributable to experimental noise). In this configuration, however, there is also no significant benefit. Theoretically, however, on-path overhearing can add value when there is no single



(a) CDF of path throughputs.

(b) Per-path throughput relative to ExOR.

Figure 9: The performance of various flavors of the ExOR algorithm. ALIX network, power 30.

high-quality link for a particular hop in a route, but the combination of reception rates at the the next hop and down-line forwarders combine to provide efficient performance.

6.4 Off-path overhearing

The final addition to arrive at ExOR is to enable off-path overhearing; namely, to include the full set of potential forwarders in the forwarder list. In this case, there are multiple possible paths, and packets choose the best path dynamically. We observe, however, that ExOR is not always the most efficient. In particular, these extra nodes can actually add overhead due to scheduling: it takes time to communicate the longer forwarder list and start and stop a round. Also, if the additional nodes have poor reception, they may not receive batch maps, and keep transmitting the same data over and over again. We find that ExOR works best when routes are generally poor, but there are many of them. Off-path overhearing also helps when routing information is unreliable or out of date, as extra nodes may become valuable.

6.5 Modrate

Given the small contribution that overhearing—either on-path or off-path—makes to ExOR's performance in the testbed configuration studied so far, in retrospect it is not at all surprising that modrate would have relatively modest gains. In particular, intuitively, modrate provides larger gains when ExOR runs all links at high speed (so there is room for modrate to decrease them), but reception rates are similar across a range of intermediate hops (so the best path is just one of a number of alternatives).

In order to evaluate the potential for modrate to improve performance when these conditions arise, we attempt to boost the average link rate selected by ExOR by increasing the connectivity of the network. Rather than modify the topology—which would make it hard to compare results across runs—we adjust the network-wide power level. As observed in Figure 2, different power levels have dramatically different reception ranges in the ALIX testbed. We rerun the previous experiments at three additional power levels—40, 50, and 60 (full power)—in addition to the level 30 results previously reported. As previously noted, modrate frequently chooses the same rate as ExOR. Hence, we restrict our attention to routes where modrate selects different speeds—approximately 7–10% of all possible routes in the ALIX testbed, depending on the power level employed. Figures 10(a)–10(d) present the results for all four speeds. (Note that the level-30 graph is simply a restatement of Figures 7(b) and 9(b).) To give an understanding of the magnitude of experimental noise, we run the modrate algorithm twice and plot both results ('modrate' and 'modrate2').

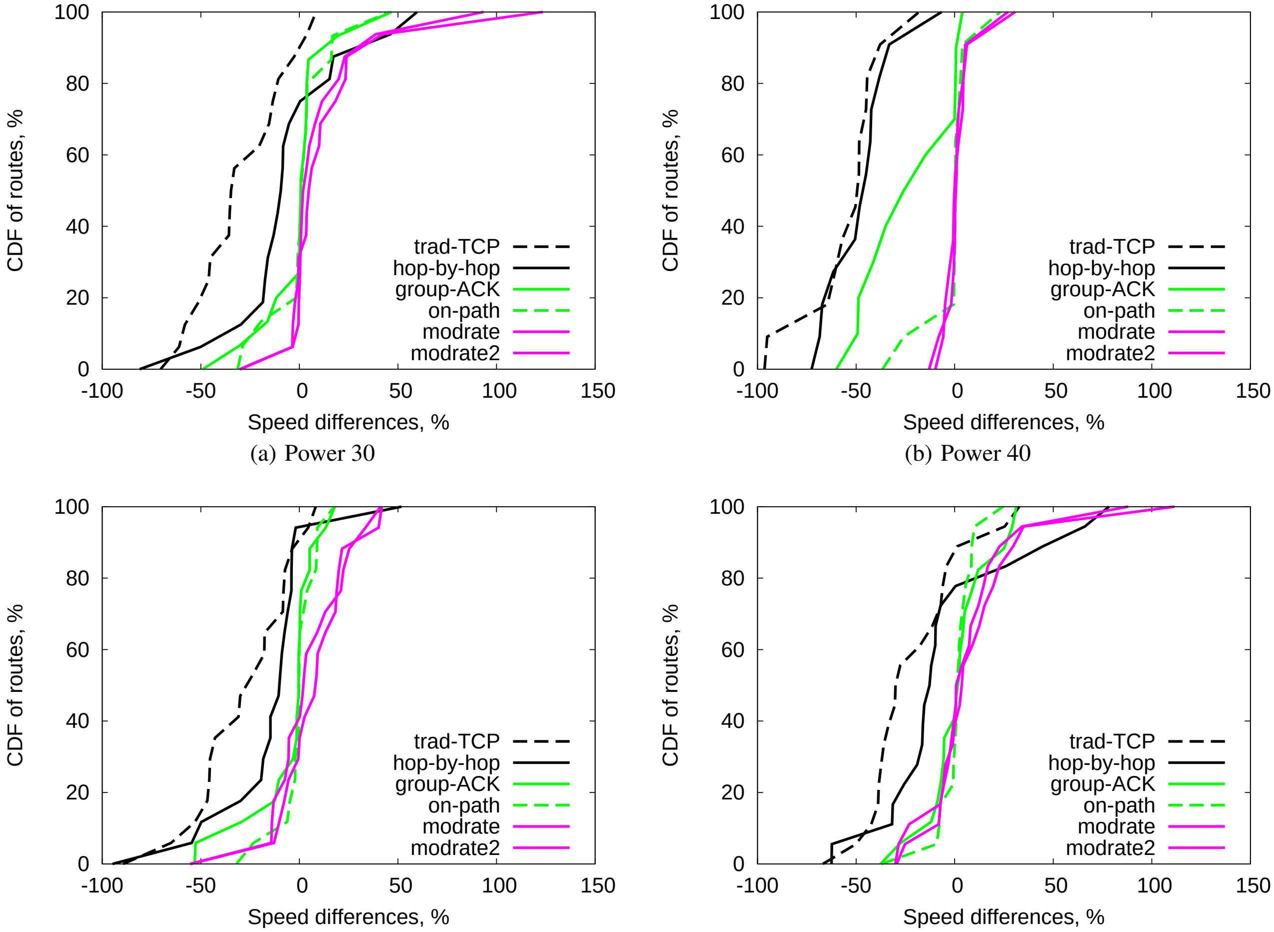
Not only does the contribution of modrate change with power level (peaking at power level 50 when connectivity is high, but still more variable than at full power), but the various components of ExOR do as well. Notably, the contribution of group acknowledgments decreases at power level 40, presumably because ExOR has selected unreliable links. Bulk acknowledgments are similarly of limited utility in the presence of lossy links. Several overall observations can be made as well: none of the techniques provide much improvement at low or full power, as poorly connected network generally has only one path made of of low quality links, while, conversely, a well-connected network with short paths does just fine with traditional routing. Networks with a range of connectivity provide the most fertile ground for all of the enhancements, but the relative importance of each can vary.

7. BUILDING-WIDE PERFORMANCE

Now that we understand the reasons for potential performance improvements, we return to consider whether such conditions exist in the Jigsaw testbed. Figure 11 shows the performance of various schemes on the top three (2nd through 4th) floors of the Jigsaw testbed. These experiments use 802.11g so they were conducted late at night in an attempt to reduce the interference from the production 802.11g network. The Jigsaw nodes are significantly less powerful than those in the ALIX testbed, and turn out to be CPU-limited when using the MORE protocol, so we do not report results for MORE.

Because it was originally deployed as a passive sniffer network, the Jigsaw nodes are quite dense. Hence, the overall results most closely resemble those from the maximum-power ALIX experiment: there is little difference between ExOR and any of its variants—including modrate. Due to the disparate layouts of the floors, however, there is significant variation between the connectivity of individual floors. Hence, we also plot the performance one floor at a time.

Figures 12 and 13 show the results for floors two and three, respectively. Each floor has a similar area and density of nodes, so comparing floors reveals differences between similar networks in



(a) Power 30 (b) Power 40

(c) Power 50 (d) Power 60

Figure 10: Modrate performance relative to ExOR on the ALIX network at various power levels.

different physical locations. Floor two, in particular, shows substantial improvement from modrate and bulk transport—similar to power 50 in the ALIX testbed. Yet floor three has some routes with dramatic improvements under modrate, some exceeding 100%, over and above the already dramatic improvements due to group acknowledgments. Floor four (not shown), on the other hand, had several routes where modrate harmed performance, likely due to significant differences between predicted and experience reception rates cause by broadband interference or hidden terminals. (Coincidentally, the hidden terminals previously reported to exist in the Jigsaw testbed [10] were located on the fourth floor.)

8. SUMMARY

Most opportunistic bulk-transfer schemes reported in the literature employ a single, fixed link rate. We find that an adaptive scheme is almost always superior, even if it only takes into account the immediate next hop in a path. While it appears that the joint optimization of link rates and overhearing potential may not significantly increase the performance of ExOR in many cases (Chachulski came to a similar conclusion regarding MORE [7]), our study reveals a somewhat surprising reason: overhearing itself is often only a minor contributor to the performance of ExOR. Instead, far more prosaic aspects of the implementation like bulk transport and group acknowledgments deliver the vast majority of the improvement: In only one of our tested configurations (the ALIX testbed at power level 40) did overhearing contribute more than a few percent of performance improvement to the median route over a simple group-acknowledgment, bulk transfer scheme. In every other instance, almost all of the improvement was gained without taking advantage of overhearing—regardless of whether link rates were optimized in an attempt to enhance overhearing.

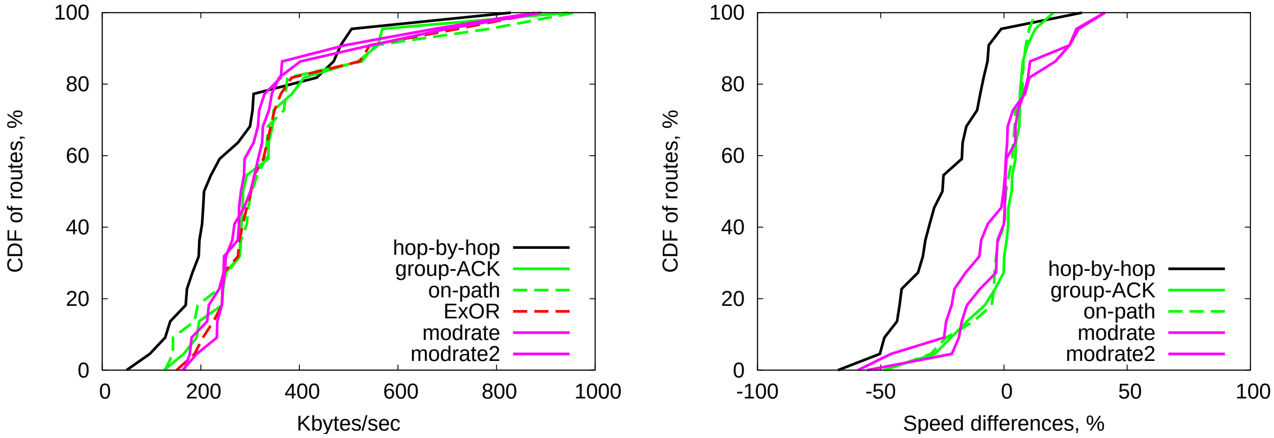
Acknowledgments

The authors are indebted to Stefan Savage, Patrick Verkaik, and Geoff Voelker for their invaluable comments on earlier versions of this manuscript, and to Szymon Jakubczak and Dina Katabi for assistance with their MORE implementation. This work is funded in part by the UCSD Center for Networked Systems (CNS), NSF CAREER grant CNS-0347949, and a grant from the Alfred P. Sloan Foundation.

9. REFERENCES

[1] M. Afanasyev, D. G. Andersen, and A. C. Snoeren. Efficiency through eavesdropping: Link-layer packet caching. In *Proceedings of ACM/USENIX NSDI*, Apr. 2008.

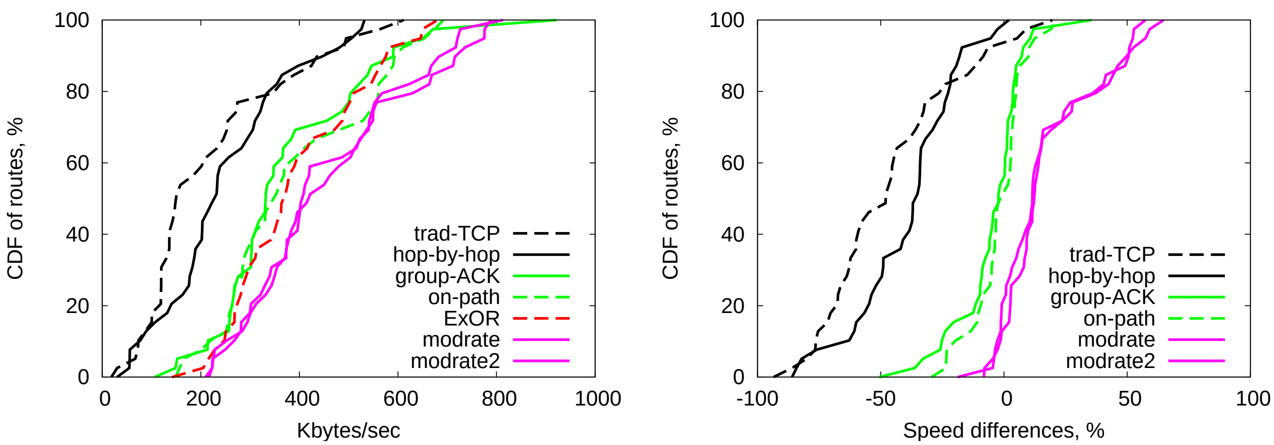
[2] M. Afanasyev, T. Chen, G. M. Voelker, and A. C. Snoeren. Analysis of a mixed-use urban WiFi network: When metropolitan becomes neapolitan. In *Proceedings of USENIX IMC*, Oct. 2008.



(a) CDF of path throughputs.

(b) Per-path throughput relative to ExOR for modrate

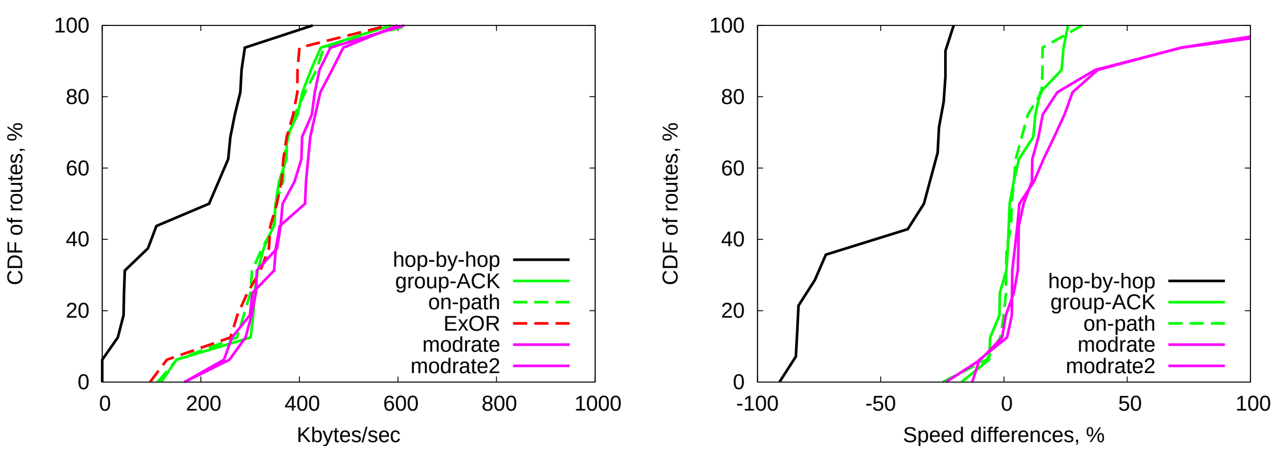
Figure 11: Jigsaw network, full power.



(a) CDF of path throughputs.

(b) Per-path throughput relative to ExOR for modrate

Figure 12: Jigsaw network, 2nd floor only; full power.



(a) CDF of path throughputs.

(b) Per-path throughput relative to ExOR for modrate

Figure 13: Jigsaw network, 3rd floor only; full power.

[3] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurement from an 802.11b mesh network. In *Proceedings of ACM SIGCOMM*, Aug. 2004.

[4] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *Proceedings of the ACM Mobicom*, Aug. 2005.

[5] J. C. Bicket. Bit-rate selection in wireless networks. Master's thesis, Massachusetts Institute of Technology, 2005.

[6] S. Biswas and R. Morris. ExOR: Opportunistic multi-hop routing for wireless networks. In *Proceedings of ACM SIGCOMM*, Aug. 2005.

[7] S. Chachulski. Trading structure for randomness in wireless opportunistic routing. Master's thesis, Massachusetts Institute of Technology, 2007.

[8] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *Proceedings of ACM SIGCOMM*, Aug. 2007.

[9] K. Chebrolu, B. Raman, and S. Sen. Long-distance 802.11b links: Performance measurements and experience. In *Proceedings of ACM Mobicom*, Sept. 2006.

[10] Y.-C. Cheng, M. Afanasyev, P. Verkaik, P. Benkö, J. Chiang, A. C. Snoeren, S. Savage, and G. M. Voelker. Automating cross-layer diagnosis of enterprise wireless networks. In *Proceedings of ACM SIGCOMM*, Sept. 2007.

[11] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage. Jigsaw: Solving the puzzle of enterprise 802.11 analysis. In *Proceedings of ACM SIGCOMM*, pages 39–50, Sept. 2006.

[12] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of ACM/IEEE MobiCom*, Sept. 2003.

[13] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of ACM/IEEE MobiCom*, Sept. 2004.

[14] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In *Proceedings of ACM Mobicom*, July 2001.

[15] A. Kamerman and L. Monteban. WaveLAN II: A high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal*, pages 118–133, 1997.

[16] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Mèdard, and J. Crowcroft. XORs in the air: practical wireless network coding. In *Proceedings of ACM SIGCOMM*, pages 243–254, Sept. 2006.

[17] P. Larsson. Selection diversity forwarding in a multihop packet radio network with fading channel and capture. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):47–54, 2001.

[18] C. Reis, R. Mahajan, M. Rodrig, and D. Wetherall. Measurement-based models of delivery and intereference. In *Proceedings of ACM SIGCOMM*, Aug. 2006.

[19] R. Roy Choudhury and N. Vaidya. MAC layer anycasting in wireless networks. In *Proceedings of ACM HotNets Workshop*, 2003.

[20] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. OAR: An opportunistic auto-rate media access protocol for ad hoc networks. *Wireless Networks*, 11(1–2):39–53, Jan. 2005.

[21] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *Proceedings of ACM Mobicom*, pages 146–157, Sept. 2006.