# Cooperative Packet Recovery in Enterprise WLANs

Mahanth Gowda
Duke University

Souvik Sen
Hewlett Packard Laboratories

Romit Roy Choudhury
Duke University

Sung-Ju Lee
Hewlett Packard Laboratories

***Abstract*—Cooperative packet recovery has been widely investigated in wireless networks, where corrupt copies of a packet are combined to recover the original packet. While previous work such as *MRD* (Multi Radio Diversity) and *Soft* apply combining to bits and bit-confidences, combining at the symbol level has been avoided. The reason is rooted in the prohibitive overhead of sharing raw symbol information between different APs of an enterprise WLAN.**

**We present *Epicenter* that overcomes this constraint, and combines multiple copies of incorrectly received "symbols" to infer the actual transmitted symbol. Our core finding is that symbols need not be represented in full fidelity — coarse representation of symbols can preserve most of their diversity, while substantially lowering the overhead. We then develop a rate estimation algorithm that actually exploits symbol level combining. Our USRP/GNURadio testbed confirms the viability of our ideas, yielding 40% throughput gain over *Soft*, and 25-90% over 802.11. While the gains are modest, we believe that they are realistic, and available with minimal modifications to today's EWLAN systems.**

## I. Introduction

Enterprise Wireless LANs (EWLANs) are composed of multiple WiFi access points (APs) connected over a wired backbone. When a client transmits a data packet to its associated AP, the broadcast nature of the wireless channel enables multiple other APs to overhear copies of this packet. In 2006, MRD [1] observed that when a reception fails at an AP, the retransmission of that packet is not always necessary. The packets overheard by other EWLAN APs can be forwarded to a common AP over the wired backbone, and these erroneous packets may be combined to recover the failed transmission. The core intuition is that the errors observed by each AP are likely to be diverse (due to diversity of link qualities), and hence, cooperative error recovery can be effective.

While MRD showed that error correction can occur entirely at the MAC layer (using bit-level voting), later, Soft [2] demonstrated that physical layer (PHY) hints can be valuable to optimize MRD. Specifically, *Soft* showed that the PHY layer can estimate a confidence for every bit that it decodes, and this confidence should be leveraged to improve the combining process. While *Soft* attained consistent throughput gains over MRD, we believe additional gains are possible by combining packets at the *PHY layer symbol level*. The key insight is that PHY layer symbols capture the true diversity of the channel and combining them is more likely to yield the correct result. As an example, consider the scenario in Fig. 1(a), where a client has transmitted a symbol corresponding to the constellation point $P_6$. The client's associated AP did not receive this symbol correctly,

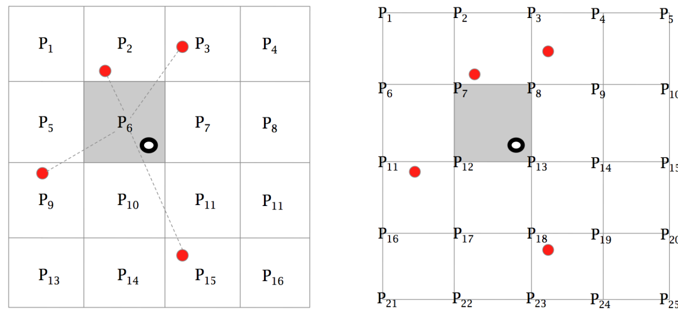



Fig. 1. (a) The red circles depict the 4 copies of the corrupt symbols, received by 4 different APs. The expectation is that the (weighted) centroid of these symbols – denoted by the oval – falls in the box of the transmitted symbol ($P_6$ in this case). (b) Example of a higher density constellation – symbols represented by mapping to the closest point on this constellation.

but has gathered 4 erroneous symbols from 4 different APs (including itself). Now, by computing some function on these symbols (e.g., centroid), and mapping the result (the black oval) to the nearest constellation point ($P_6$), the actual transmitted symbol can be recovered. If the same operation were to be performed on bits (by first mapping the 4 symbols to $P_2$, $P_3$, $P_9$, and $P_{15}$), then some channel information would get lost early, resulting in higher decoding error.

Symbol level combining is not a new idea and has been known for decades – *maximal ratio combining* (MRC) and its variants have extensively studied them in the past [3]. However, these schemes are primarily confined to single devices with multiple antennas, where gathering the symbols from the antennas is trivial. Symbol combining in EWLANs has been far less explored, primarily because of its prohibitive bandwidth requirement [2], [4]. In other words, each AP will be required to forward its overheard symbols over the wired backbone; given each symbol in full fidelity is 4 bytes, they can easily overwhelm a Gbps Ethernet. This paper asks: *what if symbols are forwarded at a coarser granularity to reduce overhead; will the diversity combining gains reduce dramatically when using such symbols?*

We find that the tradeoff between diversity combining and granularity favors practical systems, i.e., gains are modest even when the symbols are forwarded at coarse granularity. Fig. 1(b) shows one simple way of coarse-grained representation – a 25 point constellation. When the APs map their received symbols to one of these constellation points and forward them to the common AP, diversity combining is noticeably better compared to a 16 point constellation. As we will see later, additional changes in the structure of the constellation can further improve the combining accuracy, making EWLANs amenable to cooperative packet recovery.

Of course, reducing the symbol overhead alone is not sufficient. Rate estimation in diversity combining systems has assumed that received symbols follow a Gaussian distribution. This property is no longer preserved if one uses, say, the 25 point constellation representation. Moreover, channel qualities at different APs are significantly different compared to MRC where multiple antennas are on the same AP. To this end, Epicenter develops a rate selection algorithm that captures the channel quality and virtually replays it across various modulation/decoding schemes to converge on the optimal rate. Since the optimal algorithm proves to be CPU intensive, Epicenter designs a *lookup heuristic* that performs close to the optimal.

Epicenter addresses additional questions such as (1) how many symbols are best for combining, (2) which symbols should be discarded, and (3) how to mitigate interference. We implement Epicenter on a prototype of USRP/GNURadios and perform trace-based simulations. We show consistent gains for the upload scenario, and some benefits in the download case as well. For example, an AP need not retransmit many times to a client – the client can cache prior corrupt packets and combine them with the newly received packet to correct errors. The gains are modest in general, and sizable when the channel quality is weak.

Our overall contributions are summarized as follows:
**(1) We identify the viability of symbol combining in EWLAN settings.** The overhead of forwarding symbols over the wire is alleviated by representing symbols on a modified constellation.

**(2) We design a rate estimation algorithm that exploits symbol combining.** The proposed algorithm is provably optimal; however, to reduce complexity, we propose a simple yet effective heuristic.

**(3) We implement and evaluate our system, Epicenter, on a USRP/GNURadio testbed, and compare it against Soft and 802.11.** Results show Epicenter supports 1 to 2 notch higher bitrates than Soft for OFDM based WiFi networks, leading to $1.2x-1.8x$ improvement in throughput. When compared to 802.11 (which does not exploit cooperative combining), the improvements are naturally higher.

## II. Background and Overview

We provide some background on PHY layer (de)modulation of symbols and (de)coding, followed by Epicenter's overview.

### A. Symbols and Bits

The PHY layer at the transmitter translates a string of bits into a string of symbols, a method called *modulation*, where each symbol represents one or more bits. These predefined symbols are represented as complex numbers and they can be organized in a grid structure, called the *constellation diagram*. Fig. 2(a) shows a 16QAM constellation diagram,

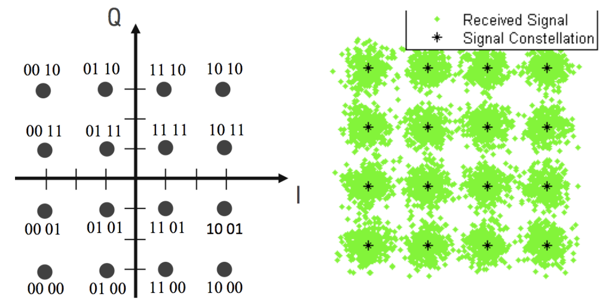



Fig. 2. Symbol constellation for 16QAM: (a) Each symbol corresponds to 4 bits. (b) Symbols received after suffering channel-induced errors.

where each symbol represents 4 bits. If a packet contains the bit string, say "1011", the transmitter uses the corresponding $x$ and $y$ axes values, also called $< I, Q >$ values, to regulate the amplitudes and phases of the carrier signal [5], [6]. The signal is then transmitted by the radio hardware.
When the signal arrives at the receiver, it applies a number of operations (such as synchronization, channel equalization, etc.), to recover the $< I, Q >$ values. However, since the channel distorts the signal, the received $< I, Q >$ values are *dispersed* from the transmitted values (green points in Fig. 2(b)). The receiver attempts to map these received I/Q values to the correct transmitted symbol, which is one of the 16 constellation points in 16QAM. For this, it simply picks the closest constellation point. If this constellation point is the same as the transmitted symbol (corresponding to "1011" in this example), the transmission is successful; otherwise, it is corrupt. This overall process is called *demodulation*.

### B. Epicenter: High Level Overview

Fig. 3 sketches the architectural overview of Epicenter. A wireless client in an EWLAN associates to a specific AP as prescribed by 802.11. It then transmits a data packet at a rate selected by a rate control algorithm. This packet might be overheard by a number of APs in the vicinity. These APs transfer the overheard $< I, Q >$ values over the wired backbone to the designated AP. To ensure that the designated AP can compute the CRC and send back the ACK in real-time, the other APs forward the symbols in batches, even before the client has completed transmission. The designated AP combines batches of symbols (detailed later), and demodulates them into a string of bits. When all the bits have been generated, they are passed through the decoding algorithm for error correction. If the CRC passes, an ACK is transmitted back; otherwise the client retransmits the packet (potentially with suitable rate adjustments). Of course, transmitting full precision $< I, Q >$ values could be too voluminous to be sent over the Ethernet backbone – Epicenter must design for this constraint.

A natural question is *should one care about upload transmissions in EWLANs?* We argue that upload traffic is increasing at a rapid pace due to a variety of factors, including cloud computing, cloud storage, P2P file access, machine-to-machine traffic, video conferencing, online gaming, etc.

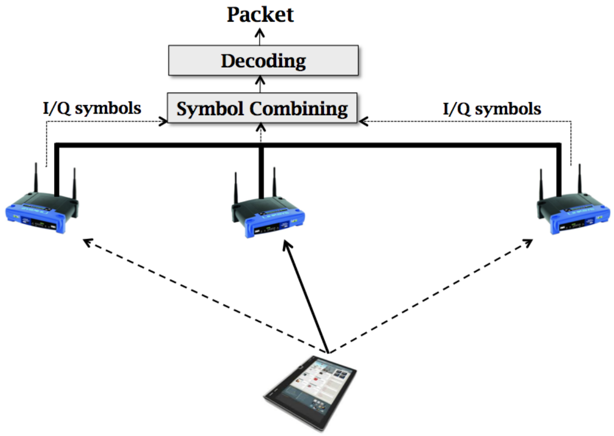


Fig. 3. Batches of (optimized) I/Q symbols transferred from different APs to the designated AP, which are in turn combined towards packet recovery.

In addition, mobile devices such as phones and tablets are equipped with a number of sensors (cameras, accelerometers, microphones, etc.), and they stream large quantities of data into the Internet [7]. Recent studies suggest rapid inflation in upload traffic in the next few years, warranting reconsideration of traditional designs that favor download.

Of course, symbol combining may be applicable to downlink scenarios as well. A client can cache packets from failed retransmissions and combine them, reducing the number of retransmissions from the AP. This might be effective when communicating to clients at the edge of the transmission range, or when channel conditions are poor.

## III. EPICENTER: SYSTEM DESIGN

We present the 3 building blocks underlying Epicenter, namely, (1) the symbol combining scheme, (2) low fidelity symbol representation, and (3) rate selection.

### A. Symbol Combining Scheme

A client's transmission is composed of a sequence of symbols. The AP associated to this client receives these symbols, as well as multiple copies of each symbol forwarded by other APs. All these symbols are stored in a matrix wherein the $i^{th}$ row corresponds to the $i^{th}$ AP, and the $j^{th}$ column corresponds to the $j^{th}$ symbol in the packet. The AP's task is to combine the elements of each column in this matrix to recover the correct symbol (i.e., the one transmitted). A naïve combining approach is the centroid algorithm. However, given that the symbols were overheard by different APs, each AP affected by diverse channel conditions, computing a basic centroid may not be ideal. Thus, Epicenter APs also transmit their respective channel state information (CSI), which are then translated to weights for the centroid algorithm. An AP with a strong channel – indicating higher confidence in the position of the symbol – is weighed proportionally in the centroid algorithm. An AP that was weak or heavily interfered, could be ignored. The final centroid position is mapped to the closest constellation point, and the corresponding bits are extracted. In the example in Fig. 1(a), symbol combining yields $P_6$, which in $16QAM$ corresponds to bits "0111".

### B. Forwarding Symbols over Wire

If all APs were to communicate raw symbols for combining (32 bits per symbol), the bandwidth requirement would amount to several Gbps [2], [4]. On the other hand, combining at the bit level (e.g., 4 bits per symbol for 16QAM) reduces the bandwidth significantly, but also offsets the combining gains. Our hypothesis is that there is a sweet spot in this tradeoff – symbols represented by just a few more bits can still offer most of the combining benefits. *The key intuition is that every additional bit available to represent symbols, increases the constellation state space exponentially* (e.g., 4 bits offer a 16 point constellation, 5 bits offer 32, and 6 bits offer 64). Representing in an exponentially denser constellation implies more channel information retained, which ultimately results in better recovery.

To verify this, we designed an experiment to measure the degradation in diversity combining when symbols are represented with fewer bits. Our experiment includes AP positions mimicking real deployments, and a client transmitting to an AP using 16QAM. We compare the following schemes:

| Name | Description |
|---|---|
| 802.11 | does not perform combining |
| Bit Level Combine (Bits) | symbols represented by 4 bits |
| Full Fidelity Combine (Optimal) | symbols represented by 32 bits |
| 25-point Const. (4+ Bits) | symbols rep. by 4.6 bits on avg. |
| 36-point Const. (5+ Bits) | symbols rep. by 5.4 bits on avg. |

To quantify diversity combining (i.e., whether the centroid falls in a good position), we use a metric called *SlackDistance* defined as: distance from the centroid to the correct constellation point minus the distance from the centroid to the closest incorrect constellation point. A negative SlackDistance indicates that the centroid will be demodulated correctly; a positive SlackDistance indicates error. Fig. 4 shows the distribution of SlackDistance – observe that $Optimal$ maintains a consistent gap over $Bits$, emphasizing the value in symbol level combining. More importantly, we find that $4{+}Bits$ makes 5% less symbol errors than $Bits$ (see Y when X=0), and $5{+}Bits$ is almost overlapping with $Optimal$. Note that 5% reduction in symbol errors can be significant – the decoding algorithm can correct many more packets if errors are lowered by 5%. Thus, Fig. 4 suggests that combining symbols is more effective than combining bits, so long as it is slightly more fine-grained than the latter.

A natural question is *how can symbols be represented in fractional bits such as* 4.6 *and* 5.4*?*. We observe that this can be achieved on average, if multiple symbols are represented together (e.g., 10 symbols represented by 46 bits). For this, a dictionary can be created for every permutation of the 10 symbols, and the total number of permutations should precisely be $2^{46}$. This implies that each permutation is 46 bits long, proving that the average number of bits per symbol is indeed 4.6. This form of representing symbols in blocks allows us fractional bits, which in turns helps us generate any constellation diagram for representing symbols. The optimal choice of the constellation is a study by itself, and

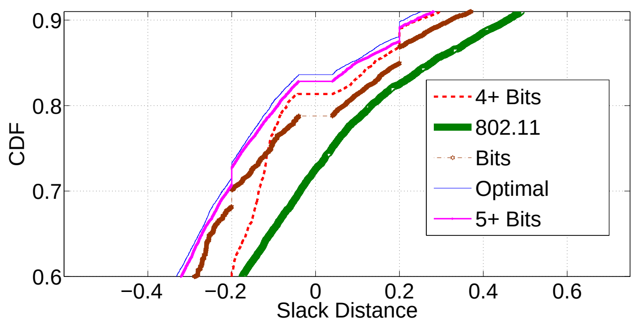



Fig. 4. Low precision I/Q symbols offer gains comparable to full precision.

is left for future work. We pick two simple ones – a 25 point constellation (4.6 bits), and a 41 point constellation (5.4 bits) shown in Fig. 5. Epicenter APs use these representations, reducing overhead while upholding combining accuracy.

### C. Rate Selection with Symbol Combining

Symbol level combining reduces bit errors, and as a result should boost the transmission rate. One might view the Epicenter APs as a giant MIMO system (each AP equivalent to a MIMO antenna) and expect the MIMO rate selection algorithms to apply here. This is because the error vectors for each symbol are Gaussian, and the combination of multiple Gaussian random variables should preserve the Gaussian property (as required by MIMO). However, recall that the Epicenter APs map the received symbols to a different constellation (e.g., 41 point constellation) before forwarding to the combiner AP. This violates the Gaussian assumption and standard MIMO rate selection is inapplicable. A separate design is necessary.

*1) "Replay and Verify" Algorithm:* We first define an optimal rate selection algorithm that Epicenter could employ for rate selection – thereafter, we refine the algorithm to reduce its computational complexity. Optimality is defined as follows: upon receiving a packet, the receiver should be able to estimate the maximum rate at which the packet *could have been* transmitted. The receiver prescribes this rate for the subsequent transmission. Of course, the prescribed rate is optimal for the subsequent packet only if the channel remains coherent, otherwise the optimal rate might change. However, this is the best possible without instantaneous channel state feedback.

To this end, the Epicenter AP first computes the *error vector* of the centroid. This is a vector joining the centroid to the actual transmitted symbol (which is known when the packet has been received correctly). Epicenter then asks: *what happens if the same error vector is applied on other modulation schemes?* Thus, it encodes the transmitted symbols at different modulation and coding schemes, applies the error vectors and verifies if correct decoding is feasible. One may view this as "replaying" the error vectors at different modulation schemes, such as AccuRate [8]. The operation repeats for every modulation and coding combination (i.e., every rate) and the highest successful rate is recorded.

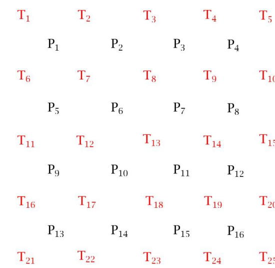



Fig. 5. 41 point constellation for representing 16QAM symbols.

Although the weighted centroid is expected to be less affected by symbols from weak APs, somewhat surprisingly, we find that discarding symbols from weaker APs helps improves recovery (we verify this observation by extensive measurements in Section IV). To leverage this, Epicenter further runs the Replay and Verify algorithm for all subsets of APs. It picks the maximum successful rate across different AP combinations and prescribes it as the bitrate of the next packet. If the channel remains coherent for the next packet, the prescribed rate is guaranteed to be successful.

*2) Heuristic for Lower Complexity:* The proposed Replay and Verify algorithm incurs $O(2^K M)$ time complexity, where $K$ is the number of APs that forwarded symbols, and $M$ is the number of rate (<modulation, coding>) schemes. Clearly, it is difficult for an AP processor to absorb this computation load. Therefore, we optimize the algorithm to bring down the complexity, in exchange for slight sub-optimality. To this end, we note the following opportunities:

(1) Of all the symbol copies, there is not much benefit in including a relatively far-away symbol, and excluding a nearby one. Therefore, we rank the symbols in increasing order of error vector magnitudes (i.e., strongest symbol first), and select subsets of $Q$ *strongest symbols*, $Q$ varying from 2 to $K$. This brings down the complexity term from $2^K$ to $(K-1)$, a substantial reduction.

(2) The second optimization is more drastic – an attempt to substitute the replaying mechanism with a simple threshold-based heuristic. The heuristic essentially uses a "threshold distance" corresponding to different rates. It checks whether certain fractions of the received symbols are within various threshold distances from the correct constellation point. The maximum rate that passes this test is the prescribed rate for the next packet. We describe the selection of "fractions" and "thresholds" next.

We use a 16 point constellation as an example, and illustrate our method of BER computation. Assume that a transmitter transmits symbol $P_i$. We first define three boxes $B_1$, $B_2$ and $B_3$ pivoted at $P_i$ (see Fig. 6). $B_1$ is the inner most grid box around $P_i$. $B_2$ is the grid box that includes all immediate neighbors of $P_i$, but not $B_1$. $B_3$ is the grid box that includes all 2-hop neighbors of $P_i$, but not $B_2$ and $B_1$.

Recall that the error vector of the centroid – denoted EVC henceforth – is the vector joining the correct constellation

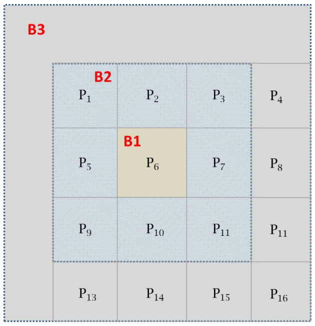



Fig. 6. Bit errors depend on the degree of dispersion.

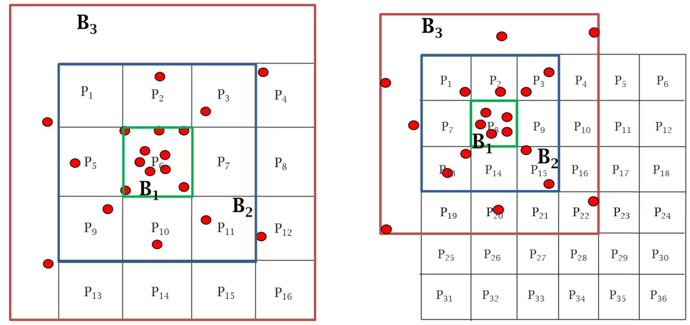



Fig. 7. Red dots denote EVC distribution (a) Boxes and Fractions at a lower rate $r_1$ (b) Boxes and Fractions at a higher rate $r_2$.

point to the centroid. Consider the distribution of the EVC derived from a received packet. We can estimate the BER of a packet from the EVC distribution as follows. Let $F_1$ denote the fraction of centroids that fall within $B_1$, with respect to the transmitted point $P_i$. Similarly, let $F_2$ and $F_3$ denote the fractions of centroids falling within $B_2$ and $B_3$ respectively. Clearly, $F_1$ fraction of symbols will be received with zero bit errors. A fraction $F_2$ of symbols will encounter 1.5 bit errors on average, because adjacent symbols in a constellation differ by 1 bit and diagonally adjacent ones differ by 2 bits (see Fig. 2). Similarly, a fraction $F_3$ symbols will incur 3 bit errors approximately. Thus, the effective BER can be estimated as $F_1 * 0 + F_2 * 1.5 + F_3 * 3$.[1]

Observe that this effective BER is statistical in nature (i.e., on average). If we simulate multiple packets whose EVCs are drawn from these fractions ($F_1$, $F_2$ and $F_3$), some of these packets will be correctly decoded, while others will fail. This implies that, for that given rate, we can estimate the delivery ratio (DR) for each <$F_1$, $F_2$ and $F_3$> tuple. Put differently, for each received packet, the Epicenter AP can create a record as follows: [Rate $r$, $F_1$, $F_2$, $F_3$, $DR$]. For another packet, another such record will get created. Over time, for packets received at different rates, Epicenter creates one table per rate. This per-rate table has records of the form [$F_1$, $F_2$, $F_3$, $DR$]. Each table has many records, each record corresponding to a unique value of <$F_1$, $F_2$, $F_3$> measured from the past. These per-rate tables are generated when Epicenter is installed. Tables can be built completely offline also, by simulating various distributions of $F_i$. For a reasonable precision, the table will only occupy a few (1-20) MBs of memory.

In the steady state, when a packet arrives at rate $r_1$, the Epicenter AP computes the packet's <$F_1$, $F_2$, $F_3$> tuple from the EVCs. Now, the AP looks up the table for rate $r_1$, and obtains the delivery ratio, $DR_1$. The throughput of the next packet, if transmitted at the same rate $r_1$ is expected to be $r_1 \times DR_1$. Now, Epicenter must compute the throughput if this packet was transmitted at a higher rate, say $r_2$. For this, it needs to first determine what <$F_1$, $F_2$, $F_3$> would have been, had this packet been transmitted at rate $r_2$. Fig. 7 explains this operation with a visual example.

[1]We intend to clarify that use of three boxes and fractions is a heuristic – one can use less or more boxes to regulate the tradeoff between BER estimation accuracy and complexity.

Observe that from the packet at rate $r_1$ (sparser constellation grid as shown in Fig. 7(a)), 10 centroids are in box $B_1$, 6 in box $B_2$, and 4 in $B_3$, implying that <$F_1 = \frac{10}{20}$, $F_2 = \frac{6}{20}$, $F_3 = \frac{4}{20}$>. Now, for the same centroids, if we apply the next higher rate $r_2$ (i.e., denser constellation grid as shown in Fig. 7(b)), fewer centroids fall inside box $B_1$. To be precise, 5 centroids now fall within $B_1$, 8 in box $B_2$, and 7 in $B_3$, implying that for $r_2$, <$F_1 = \frac{5}{20}$, $F_2 = \frac{8}{20}$, $F_3 = \frac{7}{20}$>. Thus, Epicenter uses this <$\frac{5}{20}$, $\frac{8}{20}$, $\frac{7}{20}$> tuple as the key to lookup the table for rate $r_2$, and obtains the corresponding delivery ratio, $DR_2$. The throughput, if the packet was sent at $r_2$, would have been $r_2 \times DR_2$. The exact same operation repeats for every rate, and the corresponding throughputs computed. Finally, Epicenter selects the rate that offers maximum throughput, and prescribes it back to the client. Algorithm 1 shows the pseudocode for the rate estimation engine. Clearly, table lookups is a lightweight operation and needs to be performed once for each rate, and yet, the rate selected can be quite close to the optimal.

**Algorithm 1** Rate Estimation Algorithm

$maxThroughput \leftarrow 0$
$selectedRate \leftarrow 1$
Input: $EV_{array}$: An array of Error Vectors computed from the previous packet
Input: $Table$: Table indexed by $i$, where $i$ is the $i^{th}$ rate. Table($i$) holds <$F1$, $F2$, $F3$, $DR$> for rate $i$
**for all** rate $r$ in the list of supported rates **do**
  Determine sizes $B_1$, $B_2$ and $B_3$ corresponding to the modulation scheme used in $r$.
  Determine fractions $F_1$, $F_2$ and $F_3$ among $EV_{array}$ that lie within $B_1$,$B_2$ and $B_3$ respectievly.
  Find a closest match to <$F_1$,$F_2$,$F_3$> in Table($r$)
  Let the corresponding Delivery Ratio from the match is $DR_r$
  $throughput \leftarrow r * DR_r$
  **if** $throughput > maxThroughput$ **then**
    $maxThroughput \leftarrow throughput$
    $selectedRate \leftarrow rate$

*3) Selection of APs for Packet Combining:* As mentioned earlier, symbol combining can be improved when a certain subset of all the overhearing APs is used (i.e., symbols from certain weak APs ignored). For this, we rank all the $N$ APs

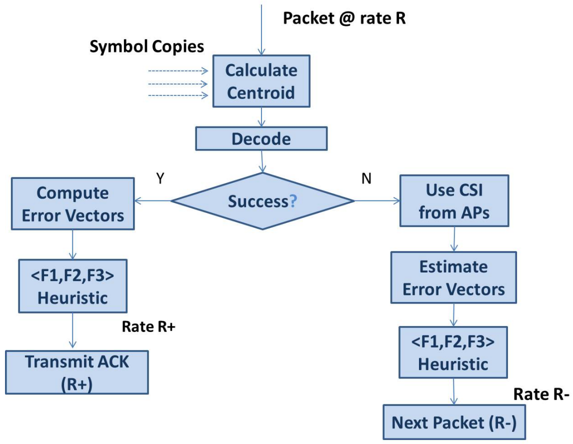



Fig. 8. The flow of operations with Epicenter. Observe that error vectors are "computed" from symbols upon packet success, but "estimated" from CSI upon failure.

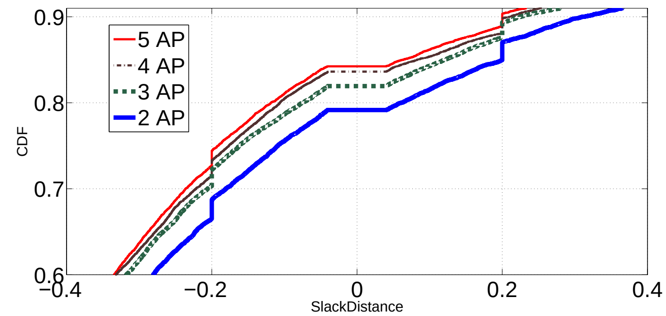



Fig. 9. SlackDistance: difference between the $EV$ of correct constellation point and the minimum of $EV$ of incorrect constellation points.

in decreasing order of signal strength, select the top $k$ APs, and compute the best rate for this combination. We vary the value from $k=2$ to $N$, indicating that the above pseudo-code is invoked $(N-1)$ times. The maximum of all these rates is prescribed for the next packet, and the corresponding value of $k$, say $k_{max}$ is recorded. When the next packet indeed arrives, the top $k_{max}$ strong APs are used for combining.

*D. Putting All Modules Together*

Fig. 8 summarizes Epicenter's flow of operations. During bootstrap, Epicenter pre-computes $< F_1^*, F_2^*, F_3^* >$ thresholds for each modulation scheme and shares it with all APs. When the next packet arrives, the designated AP gathers all the symbol copies from other APs, and applies the weighted centroid algorithm for cooperative demodulation. This process is pipelined (i.e., APs start forwarding symbols even before the full packet has been received). Once all the symbols are demodulated, the buffered bits are passed through the decoder, followed by a CRC check.

If the CRC check is successful, the AP immediately knows the precise error vectors for all symbol copies. Thus, it can execute the $< F_1^*, F_2^*, F_3^* >$ heuristic test for all rates higher than the rate at which this packet was received. If rate $R^+$ is prescribed by the heuristic, the Epicenter AP includes $R^+$ in an ACK and sends it back to the client. However, if the packet failed the CRC, the AP does not send an ACK as per 802.11 specifications. Instead, it uses the channel state information (CSI) forwarded by each AP to *estimate* the error vectors, and runs the $< F_1^*, F_2^*, F_3^* >$ heuristic using them.[2] The prescribed rate in this case is communicated back to the client in a subsequent download packet to that client. The client adapts its rate for the next transmission.

*E. Point of Discussion*

Nearby APs are often allocated to orthogonal channels, implying that not many APs may overhear a client's transmission. Will diversity combining be really beneficial?

[2]Observe that the channel state information is a complex number and indicative of the error vector, when added with a Gaussian noise component.

This is a valid concern and we tested the efficacy of symbol-based combining against varying number of symbol copies. To design the experiment, we placed 6 APs in locations that mimic the actual AP placement in our building. We then placed the client at arbitrary positions, associated it to the strongest AP (as prescribed by 802.11), and characterized the best bitrate to that AP (i.e., the bitrate that supports $>90\%$ delivery ratio). Let's call this bitrate $R$. The client then transmitted a batch of packets at this bitrate. The traces were collected and used to test the sensitivity to the "number of symbol copies".

We use $SlackDistance$, defined in Section III-B, to understand the results of the above experiment. Recall that a negative value of $SlackDistance$ implies correct demodulation. Since we know the transmitted packet, we precisely know the correct constellation points, and hence, $SlackDistance$ could be computed. Fig. 9 shows the distribution of $SlackDistance$ for increasing number of symbol copies (at bitrate QPSK). Evidently, even with 2 copies, around 80% of the symbols are decoded correctly, and more symbols offer diminishing returns. This suggests promise with Epicenter – we discuss this in more detail and translate these results to throughput in Section IV.

As an aside, MRD [1] suggested that APs could be instrumented with a passive radio on other channels, solely for the purpose of overhearing. If this is feasible (perhaps given that WiFi radios will become even cheaper), then greater diversity gains may be extracted. We are also investigating whether combining symbols from signals leaked from adjacent channels would be effective.

## IV. Epicenter Implementation

Epicenter builds on the OFDM codebase for the USRP/GNURadio platform. We implement an 802.11a/g-like transceiver that uses interleaving, puncturing, and soft input soft output viterbi decoding (SOVA) [9] (used to implement Soft [2] for purposes of comparison). The bandwidth in our testbed experiment is fixed at 2MHz. Our OFDM implementation uses 1024 point FFT, 394 occupied subcarriers, 8 pilot tones, and a cyclic prefix of length 256.

**Experiment Design:** The high latency incurred in procuring RF samples from the USRP front end prohibits real-time evaluation of Epicenter. Thus, we approximate Epicenter's performance using traces collected in our 10 node USRP testbed. We co-located 6 USRPs with the existing APs in our engineering building. All the USRP APs are tuned to the same 2.43GHz channel. We collected 74 walking traces (each of 1000 packets) starting from random client locations. To collect these traces, we move a USRP client on a wheel chair while it communicates with the nearest AP. The idea is to capture real-life channel state information (CSI) and error vector (EV) variations using these traces. APs detect client packets, compute and forward CSI and per symbol EV information to the central server. The central server runs various protocols for comparison:

• **MRC Symbol:** This is full fidelity combining using maximal ratio combining (MRC). APs forward full precision symbols (<I,Q> values) to the central combiner, which in turn combines them to correct erroneous packets.

• **MRC Bits:** APs map received full precision symbols to the closest constellation point, and forward the mapped bits to the combiner. For 16QAM, each symbol maps to 4 bits.

• **Epi 1x:** For a given modulation, Epi 1x creates a grid by placing points at the corner of the grid boxes – like points $T_i$ in Fig. 5. Thus, for QPSK, the constellation would be a 9-point constellation; for 16QAM, it would be a 25-point constellation. The number of bits used to represent symbols with, say a 25-point constellation is 4.6 bits, slightly greater than what *MRC Bits* would use for the same 16QAM.

• **Epi 2x:** For a given constellation, Epi 2x is the union of the constellation points in that modulation and its Epi 1x version – (points $T_i \cup P_i$ in Fig. 5). For 16QAM, the number of points in the Epi 2x constellation is 41; the number of bits used to represent symbols with Epi 2x is 5.4 bits.

In addition, we compare Epicenter's performance with the following protocols:

• **802.11:** A legacy 802.11 client associates and communicates only with the strongest AP, without any combining. We use the AccuRate [8] rate control protocol to realize the best throughput 802.11 can achieve. This offers a strong baseline to understand the merits in symbol combining.

• **Soft:** We implement Soft as described in [2]. Soft combines bit-level confidence across multiple receptions to correct errors. However, unlike Epicenter, Soft does not propose any method of estimating rates under diversity combining. For a fair comparison, we empirically find the highest bit-rate Soft can support for a given packet (by replaying the packet at increasing bit-rates using CSI and error vectors (EV)). The highest successful bit-rate is used.

### *A. Performance Evaluation*

We target answering the following key questions: (1) How effective is the <$F_1$, $F_2$, $F_3$> heuristic rate selection scheme in comparison to the optimal choice of rate? (2) How effective is MRC to low precision <I,Q> values? (3) How much throughput gain can Epicenter provide over 802.11 and Soft? (4) For download scenarios, how much throughput can be gained by reducing retransmissions?

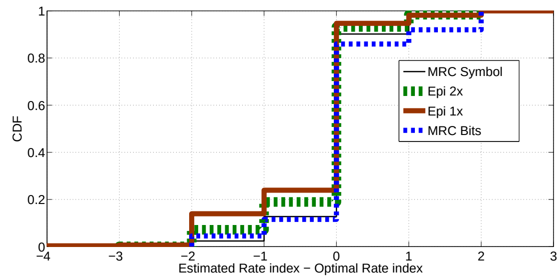
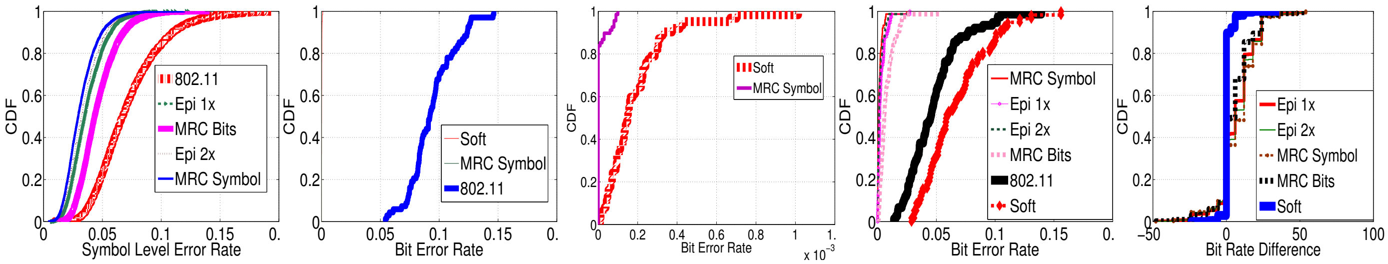


Fig. 11. Rate estimation accuracy of Epicenter vs. optimal.

*1)* **Rate Selection Accuracy:** Fig. 11 shows the rate selection efficacy of different protocols, compared to the optimal rate (computed from the replay and verify algorithm). Note that a vertical line at $X = 0$ is the optimal rate; deviations from this line indicates errors in rate selection. Evidently, Epicenter (i.e., all its variants) overlap quite reasonably with the optimal vertical line, achieving around 70−80% accuracy. Of course, the system under-selects rates slightly more often. This is because the thresholds on delivery ratio (DR) in Section III-C2 are designed to favor under-selection, since this leads to fewer packet losses.

*2)* **Robustness of Low Precision Symbols:** Recall that full precision *MRC Symbol* is prohibitive because of its bandwidth requirement over the wire [2], [4]. To this end, we adopt coarse-grained representation of symbols through (shifted and interspersed) constellations. How well do these coarse-grained versions, namely $1x$ and $2x$ interspersion perform? Figures 10(a) demonstrates that $1x$ and $2x$ representation of symbols are consistently effective, producing comparable symbol error rates, in comparison to $Optimal$. Further, Epicenter outperforms 802.11 and Soft in terms of symbol errors because of its ability to decode more errors. In sum, the symbol communication overhead is sizably reduced while the combining degradations are marginal.

*3)* **Upload BER and Throughput Gain:** BER is a strong indicator of system throughput, hence, we compare the achieved BER between Epicenter and Soft. First we operate on weak links at the optimal rate at which Epicenter will successfully decode. Fig. 10(b) shows the resulting BER comparison with 802.11 (both Soft and Epicenter are invisible since they are almost aligned with the Y axis, and are much better than 802.11). Fig. 10(c) zooms in on the left side of the X axis, and evidently, Soft exhibits a lower BER than 802.11, but higher than Epicenter. However, when we operate the same link at a higher rate (12 Mbps), Soft's performance falls below 802.11 (Fig. 10(d)). This is because Soft's PHY confidence incurs high errors at over-optimal rates. Epicenter on the other hand outperforms both. Reduced BER enables an Epicenter client to increase its rate further without suffering losses – Fig. 10(e) shows that across several traces, Epicenter support higher rates in 55% cases. It also shows how *Epi 1x* and *Epi*




(a) (b) (c) (d) (e)

Fig. 10. Performance comparison of Epicenter with 802.11 and Soft in terms of (a) symbol error rate, (b) bit error rate on a channel operated at near optimal rate, (c) resolution among BER curves of Soft and MRC at near optimal rates, (d) bit error rate on a channel operated at aggressive rate, and (e) rate supported for upload scenarios.

*2x* closely follow the *MRC Symbol* scheme.

Fig. 12 demonstrates the actual throughput gains with Epicenter over Soft and 802.11. These gains of 50% over 802.11 and 40% over Soft are a direct fallout of supporting higher rates with symbol combining. In contrast, *MRC Bits*, which does not use interspersed constellations, is unable to exploit diversity, resulting in a lower gain.

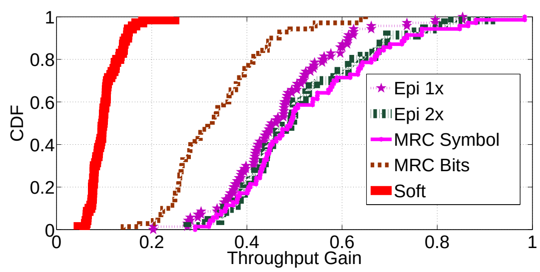



Fig. 12. CDF of throughput gain of Epicenter, Soft over 802.11.

*Correct AP selection:* Fig. 13(a) shows how performance suffers if symbols from all APs are blindly combined. Also, combining symbols from the strongest three APs is always worse than all six APs, because there is some loss in diversity with the former. We find that the best subset of APs to be used for diversity combining varies over traces and packets (Fig. 13(b)). Thus, AP selection must be a dynamic decision.

Fig. 13(a) also shows the best throughput obtained by combining from ideal number of APs for each packet. Our proposed constellation mapping heuristic from Section III-C2 estimates the best set of APs that maximizes performance. If the estimated set of APs fails to decode the packet, Epicenter tries combining from a different $k$ strongest APs, which had second best optimal rate as computed from the previous packet, and so on. Fig. 13(c) shows that such an attempt doesn't increase the overhead. The average number of decodings per packet was approximately 2. With only 2 decodings on average, Epicenter is able to achieve throughput as the ideal number of APs (Fig. 13(a)). Fig. 13(d) shows that as the number of overhearing APs increase, the throughput gain over 802.11 increases. However, the symbol communication overhead is a constant factor to that of bit-level combining and MRC. More APs provide better diversity combining opportunity, allowing Epicenter to be more prudent in its AP selection.

*Performance at different rates*: Fig. 14 shows Epicenter's 1x throughput gain over 802.11 for different rates. Evidently, the gain reduces at higher rates. This is because the MAC layer throughput gap is known to reduce with increase in bit-rates (owing to the constant protocol overhead [10]). Thus even though Epicenter supports a higher bit-rate than 802.11, the relative gain decreases. With multiple stream MIMO, we believe that the gain should be proportinal to the corresponding single stream data-rates. At the end of MIMO decoding, the residual signals could be combined with the Epicenter approach.

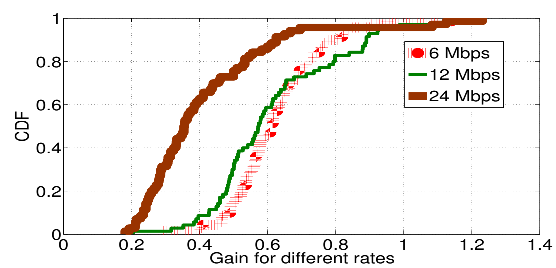
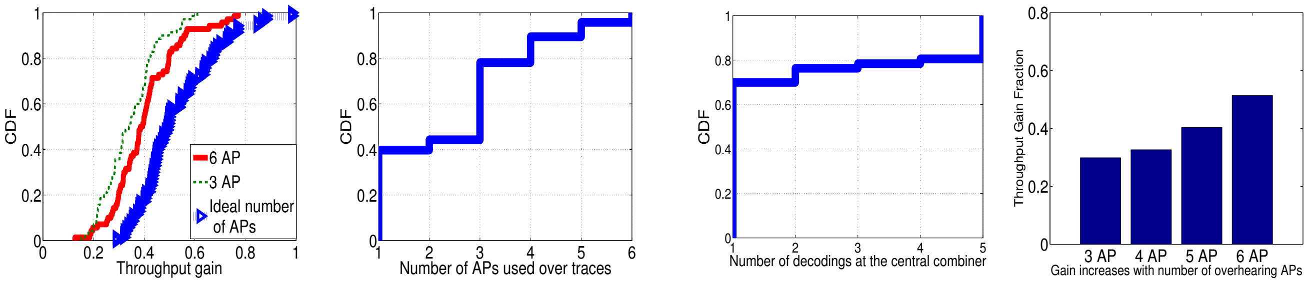



Fig. 14. Epicenter's throughput gain for various rates.

*4)* **Reducing Download Retransmissions:** Packet losses are prompted by channel fluctuations and interference. 802.11 retransmits to recover from such losses, often 5 to 7 times if the channel is weak. A combining approach can reduce these retransmissions. For example, Soft may combine the correct bits from multiple failed download retransmissions to correctly decode the packet. Epicenter can employ symbol combining to correct more errors than Soft, requiring fewer retransmissions. Observe that, Epicenter can use full precision <I,Q> values for download transmissions because of no communication overheads. To evaluate download gain, we transmit packets over randomly chosen links. Fig. 15 shows modest gains from Epicenter over 802.11 and Soft.

*5)* **BackHaul Bandwidth:** Soft [2] requires $3x$ bandwidth compared to just bits. Epicenter 1x requires approximately $2x$ the overhead on a similar scale, while 2x requires $2.5x$. In contrast, *MRC Symbol* requires 9 times. Constellation points selected by *Epicenter 1x* would be a practical choice. Thus, *Epicenter 1x* achieves performance close to that of pure symbol combining, but uses up bandwidth comparable to Soft – the best of both worlds.




(a) (b) (c) (d)

Fig. 13. (a) Performance gain of Epicenter over 802.11 when suboptimal set of APs are chosen for symbol combining. (b) Number of APs actually used by Epicenter. (c) Number of times an attempt is made to decode packet by combining from different AP combinations. (d) Epicenter's performance (*Epi 1x*) improves as more APs overhear the same client transmission.

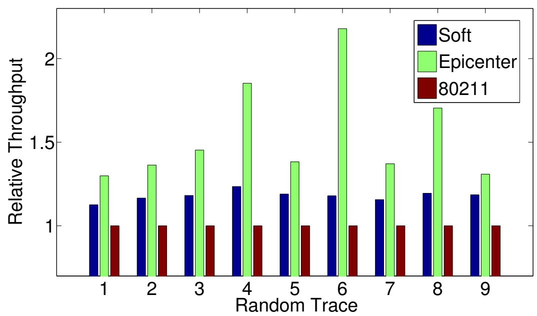



Fig. 15. Download throughput gain of Epicenter and Soft over 802.11.

## V. Related Work

Improving performance by exploiting spatial diversity has been an active area of research [11]–[13]. Epicenter, MRD [1] and Soft [2] explore diversity across APs, but other proposals combine information from multiple interfaces and antennas at the same AP. Techniques include choosing the best transmit or receive antenna [14], and combining signals across multiple antennas (Maximal Ratio Combining) [5], [15]. These approaches yield high performance gain but require tight time synchronization. Spatial diversity combining techniques also appear in the context of soft-handoff in CDMA networks [16] where a cellphone can simultaneously connect to two or more cells during a call. Properties of CDMA signaling schemes enable cellphones to combine the received signals from two different base stations, improving reliability and performance.

PHY confidences have been used in several existing schemes. PPR (Partial Packet Recovery) [17] identifies corrupt portions of the packet by examining per bit confidences, and only retransmits those. PHY confidence information has also been used for collision detection [18], rate control [19], multi hop routing [20], etc. Constellation symbol information were shown to be also useful for optimizing video traffic in APEX [21]. Epicenter is complimentary to all these schemes.

## VI. Conclusion

We presented Epicenter, a system that utilizes *symbol* level combining for cooperative packet recovery in enterprise WLAN. We showed that the PHY layer symbols, even in coarse representation, capture the true diversity of the channel and hence can be effectively used for combining without generating large backhaul traffic. Epicenter also performs rate selection that exploits symbol level combining. Our Epicenter implementation on USRP/GNURadio testbed showed that it provides 40% throughput gain over *Soft*, and 25-90% over 802.11. Our ongoing work includes further reducing the backhaul overhead, utilizing multiple antennas in the APs, and leveraging soft decoding.

## References


[1] A. K. Miu *et al.*, "Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks," in *ACM Mobicom*, 2005.

[2] G. R. Woo *et al.*, "Beyond the bits: Cooperative packet recovery using phy information," in *ACM MobiCom*, 2007.

[3] D. Brennan, "On the maximal signal-to-noise ratio realizable from several noisy signals," *Proc. IRE*, 1955.

[4] S. Gollakota, S. D. Perli, and D. Katabi, "Interference alignment and cancellation," in *ACM Sigcomm*, 2009.

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

[6] T. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall, 2001.

[7] I. Constandache *et al.*, "Enloc: Energy-efficient localization for mobile phones," in *IEEE Infocom Mini Conference*, 2009.

[8] S. Sen *et al.*, "Accurate: constellation based rate estimation in wireless networks," in *NSDI*, 2010.

[9] J. Haganauer and P. Hoeher, "A viterbi algorithm with soft-decision outputs and its applications," in *IEEE GlobeCom*, 1989.

[10] S. Sen, R. Roy Choudhury, and S. Nelakuditi, "No time to countdown: Migrating backoff to the frequency domain," in *ACM MOBICOM*, 2011.

[11] M. Win and J. Winters, "Analysis of hybrid selection/maximal-ratio combining in rayleigh fading," in *IEEE ICC*, 1999.

[12] M. Alouini and A. Goldsmith, "Capacity of rayleigh fading channels under different adaptive transmission and diversity-combining techniques," *IEEE Trans. Veh. Technol*, 1999.

[13] T. Eng *et al.*, "Comparison of diversity combining techniques for rayleigh-fading channels," *IEEE Trans. Commun.*, 1996.

[14] S. Sanayei and A. Nosratinia, "Antenna selection in mimo systems," *IEEE Communications Magazine*, 2004.

[15] D. Brennan, "Linear diversity combining techniques," *Proc. IRE*, 1959.

[16] A. Viterbi *et al.*, "Soft handoff extends cdma cell coverage and increases reverse link capacity," *IEEE JSAC*, 1994.

[17] K. Jamieson and H. Balakrishnan, "Ppr: partial packet recovery for wireless networks," in *ACM Sigcomm*, 2007.

[18] S. Sen, R. Roy Choudhury, and S. Nelakuditi, "Csma/cn: carrier sense multiple access with collision notification," in *ACM MOBICOM*, 2010.

[19] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," *ACM Sigcomm*, 2009.

[20] S. Katti *et al.*, "Symbol-level network coding for wireless mesh networks," *ACM Sigcomm*, 2008.

[21] S. Sen *et al.*, "Design and implementation of an "approximate" communication system for wireless media applications," in *ACM SIGCOMM*, 2010.