Partial Recovery
Loss Recovery

• **Wireless medium is inherently lossy**
  - Large distance between a sender and receiver
  - Obstacles
  - Collisions (e.g., hidden terminals)

• **Current solutions to cope with loss?**
Existing Solutions

• Current solutions to cope with loss
  - Automatic repeat request (ARQ)
  - FEC
  - Rate adaptation
  - MIMO
  - Network coding

• Pros & Cons?
Bits in a packet don’t share fate

Many bits from corrupted packets are correct, but status quo receivers don’t know which!
Partial Recovery

• Partial Recovery
  - Segment CRC
  - MRD
  - PPR
  - SOFT
  - ...

• Is partial recovery useful for wireline traffic?
Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks

Allen Miu, Hari Balakrishnan
MIT Computer Science and Artificial Intelligence Laboratory

C. Emre Koksal
Computer and Communication Sciences, EPFL
Today’s wireless LAN (e.g., 802.11)

- Uses only one communication path
Multi-Radio Diversity (MRD) - Uplink

- Allow *multiple APs* to simultaneously receive transmissions from a single transmitter

Loss independence $\rightarrow$ simultaneous loss $= 2\%$
Multi-Radio Diversity (MRD) - Downlink

• Allow *multiple client radios* to simultaneously receive transmissions from a single transmitter

![Diagram showing MRD concept with AP1 and AP2 connected to Internet and a mobile device]

MRDC
Demodulation

bits before FEC

FEC

bits after FEC

MRD
Are losses independent among receivers?

- Broadcast 802.11 experiment at fixed bit-rate:
  6 simultaneous receivers and 1 transmitter

- Compute loss rates for the 15 receiver-pair (R1, R2) combinations
  - Frame loss rate FLR(R1), FLR(R2) vs. simultaneous frame loss rate FLR(R1 ∩ R2)
Individual FLR > Simultaneous FLR

$\text{FLR} = x$

$\text{FLR}(R1 \cap R2)$
Challenges in developing MRD

• How to correct simultaneous frame errors?
  – Frame combining

• How to handle retransmissions in MRD?
  – Request-for-acknowledgment protocol

• How to adapt bit rates in MRD?
  – MRD-aware rate adaptation
Bit-by-bit frame combining

1. Locate bits with unmatched value

2. Select bit combination at unmatched bit locations, check CRC

Problem: Exponential # of CRC checks in # of unmatched bits.
Block-based frame combining

- Observation: bit errors occur in bursts
- Divide frame into $N_B$ blocks (e.g., $N_B = 6$)
- Attempt recombination with all possible block patterns until CRC passes
  - # of checks upper bounded by $2^{N_B}$
  - Explore bursty bit-error
  - Failure rate increases with $N_B$ under uniform error
Failure decreases with $N_B$ and burst size.
How to Perform Error-Control?
Option 1: Directly use 802.11 retransmission scheme

- Conventional link-layer ACKs do not work
  - Final status known only to MRD
Option 2: Disable 802.11 retransmission scheme
Option 2: Disable 802.11 retransmission scheme

• Problems
  - Sending our own ACK is more expensive than sending 802.11 ACK (Why?)
  - Backoff and rate control both rely on MAC-layer ACKs
Retransmission in MRD

• **Two levels of ACKs**
  - Use 802.11 ACK for per-packet acknowledgement
    • 802.11 ACK can be directly used for CSMA
    • No need to contend for the medium
  - Send MRD-ACK via ACK compression
  - Sender retransmits when MRD-ACK not received upon timeout
Request-for-acknowledgment (RFA) for efficient feedback
(a) Headers in the transmitted data frame

2 Bytes | 1 Byte | N bits
MAGIC  | SEQ   | TX STATE

(b) MRD-ACK Packet
MRD-aware rate adaptation

- **Standard rate adaptation does not work**
  - Reacts only to link-layer losses from 1 receiver
  - Uses sub-optimal bit-rates

- **MRD-aware rate adaptation**
  - Reacts to losses at the MRD-layer

**Implication:** First use multiple paths, *then* adapt bit rates.
Experimental setup

- 802.11a/b/g implementation in Linux (MADWiFi)
- L transmits 100,000 1,472B UDP packets w/ 7 retries
- L is in motion at walking speed, > 1 minute per trial
- Variants: R1, R2, MRD (5 trials each)
MRD improves throughput

Throughput (Mbps)

- **R1**: 8.25 Mbps
- **R2**: 8.25 Mbps
- **MRD**: 18.7 Mbps

2.3x Improvement

Each color shows a different trial
MRD maintains high bit-rate

<table>
<thead>
<tr>
<th>Selected bit rate (Mbps)</th>
<th>Fraction of transmitted frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0.19</td>
</tr>
<tr>
<td>12</td>
<td>0.36</td>
</tr>
<tr>
<td>18</td>
<td>0.54</td>
</tr>
<tr>
<td>24</td>
<td>0.71</td>
</tr>
<tr>
<td>36</td>
<td>0.88</td>
</tr>
<tr>
<td>48</td>
<td>1</td>
</tr>
<tr>
<td>56</td>
<td>1</td>
</tr>
</tbody>
</table>

Frame recovery data (% of total losses at R1)
via R2: 42.3%
frame combining: 7.3%
Total: 49.6%
Delay Analysis

User space implementation caused high delay.

Fraction of delivered packets vs. One way delay ($10^x$ s)