Recap

• Rate adaptation
  – High rate users competing with low rate users
  – Robust rate adaptation
  – Effective SNR
  – Frequency aware rate adaptation
  – Rateless

• Partial packet recovery
  – Using multiple radios to recover errors
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: high-layer recovery
  – PPR: physical layer
  – SOFT: MRD + PPR
  – ...
PPR: Partial Packet Recovery for Wireless Networks

Kyle Jamieson and Hari Balakrishnan
MIT Computer Science and Artificial Intelligence Laboratory
Motivation of Partial Recovery

• Lots of packets lost due to collisions and noise in wireless networks

Can’t recover non-colliding bits today!
Three key questions

1. How does receiver know which bits are correct?
2. How does receiver know P2 is there at all?
3. How to design an efficient ARQ protocol?
How can receiver identify correct bits?

• Use physical layer (PHY) hints: **SoftPHY**
  – Receiver PHY has the information!
  – Pass this **confidence information** to higher layer as a hint

• SoftPHY implementation is PHY-specific; interface is PHY-independent

• Implemented for direct sequence spread spectrum (DSSS) over MSK (this talk) and other modulations
A new source of information

PHY conveys uncertainty in each bit it delivers up

Low uncertainty

High uncertainty

(P1)

(P2)

Preamble

Preamble
Direct seq. spread spectrum background

Transmitter:
- Data stream
  - 4 bits
  - 250 Kbits/s
- Bits to chips
  - 1 codeword (32 chips)
  - 2 Mchips/s

Receiver:
- Demodulate MSK signal
- Decide on closest codeword to received (Hamming distance)
- Many 32-bit chip sequences are not valid codewords
- Codewords separated by at least 11 in Hamming distance
- 802.11 similar
SoftPHY hint for spread spectrum

Hamming distance between received chips and decided-upon codeword

Receive: 111011010001111000011010110100010
  \( C_1 \): 11101101100111000011010100100010
  \( \Rightarrow \) SoftPHY hint is 2

Receive: 11001101000111010111011110110111
  \( C_1 \): 11101101100111000011010100100010
  \( \Rightarrow \) SoftPHY hint is 9
Three key questions

1. How does receiver know which bits are correct? **A: SoftPHY:**

2. How does receiver know P2 is there at all?

3. How to design an efficient ARQ protocol?
Postamble decoding
Receiver design with postamble

- Codeword synchronization
  - Translate stream of chips to codewords
  - Search for postamble at all chip offsets

Offset 0: Codeword 1     Codeword 2     Codeword 3
Chips: 010101001010011101010001011101001010...

Offset 3: Codeword 1     Codeword 2     Codeword 3

- Chip synchronization without preamble/postamble
Three key questions

1. How does receiver know which bits are correct?

2. How does receiver know P2 is there at all?

3. How to design an efficient ARQ protocol?
ARQ with partial packets

• ARQ today: correctly-received bits get resent

• **PP-ARQ** key idea: resend **only** incorrect bits

  Hamming distance

  1010001101010111101101010101

• **Efficiently** tell sender about what happened
  – Feedback packet
Labeling bits “good” or “bad”

- **Threshold test**: pick a threshold $\eta$
  - Label codewords with SoftPHY hint $> \eta$ “bad”
  - Label codewords with SoftPHY hint $\leq \eta$ “good”

Hamming distance
PP-ARQ protocol

1. Assuming hints correct, which ranges to ask for?
   - Dynamic programming problem
   - Forward and feedback channels

2. Codewords are in fact **correct** or **incorrect**
   - Two possibilities for mistakes
     - Labeling a correct codeword “bad”
     - Labeling an incorrect codeword “good”
Putting it together

- The sender transmits a full packet with checksum
- The receiver decodes the packet and computes the best feedback
- The receiver encodes the feedback set in its reverse-link ack packet
- The sender retransmits only (a) chunks missed at receiver and (b) checksum of the remaining runs
Implementation

**Sender:** telos tmote sky sensor node
- Radio: CC2420 DSSS/MSK (Zigbee)
- Modified to send postambles

**Receiver:** USRP software radio with 2.4 GHz RFX 2400 daughterboard
- Despreading, postamble synchronization, demodulation
- **SoftPHY** implementation

**PP-ARQ:** trace-driven simulation using data from above
Experimental design

• Live wireless testbed experiments
  – Senders transmit 101-byte packets, varying traffic rate
  – Evaluate raw PPR throughput
  – Evaluate SoftPHY and postamble improvements

• Trace-driven experiments
  – Evaluate end-to-end PP-ARQ performance
  – Internet packet size distribution
  – 802.11-size preambles

25 senders
6 receivers
PP-ARQ performance comparison

• Packet CRC (no postamble)

[Diagram of Packet CRC (no postamble)]

• Fragmented CRC (no postamble)
  – Tuned against traces for optimal fragment size

[Diagram of Fragmented CRC (no postamble)]
Throughput improvement 2.3-2.8x

Per-node offered load (fraction of nominal link rate):
- 1.4%
- 2.8%
- 5.5%
- 11%

Improvement over Packet CRC:
- Packet CRC
- PP-ARQ
PP-ARQ retransmissions are short
25% improvement over fragmented

Per-node offered load (fraction of nominal link rate)
PP-ARQ retransmissions are short
PP-ARQ feedback overhead is low
Summary

• Mechanisms for recovering correct bits from parts of packets
  – SoftPHY interface (PHY-independent)
  – Postamble decoding
• PP-ARQ improves throughput 2.3–2.8× over the status quo
• PPR Useful in other apps, e.g. opportunistic forwarding
## Comparison between MRD and PPR

<table>
<thead>
<tr>
<th></th>
<th>MRD</th>
<th>PPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leverage spatial diversity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leverage temporal diversity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leverage PHY information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td></td>
<td></td>
</tr>
<tr>
<td># errors tolerated</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inherent loss vs. collision</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implication on rate adapt.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implication on MAC protocol</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>