Wireless Communication Systems
@CS.NCTU

Lecture 12: Soft Information

Instructor: Kate Ching-Ju Lin (林靖茹)

PPR: Partial Packet Recovery for Wireless Networks

ACM SIGOCMM, 2017

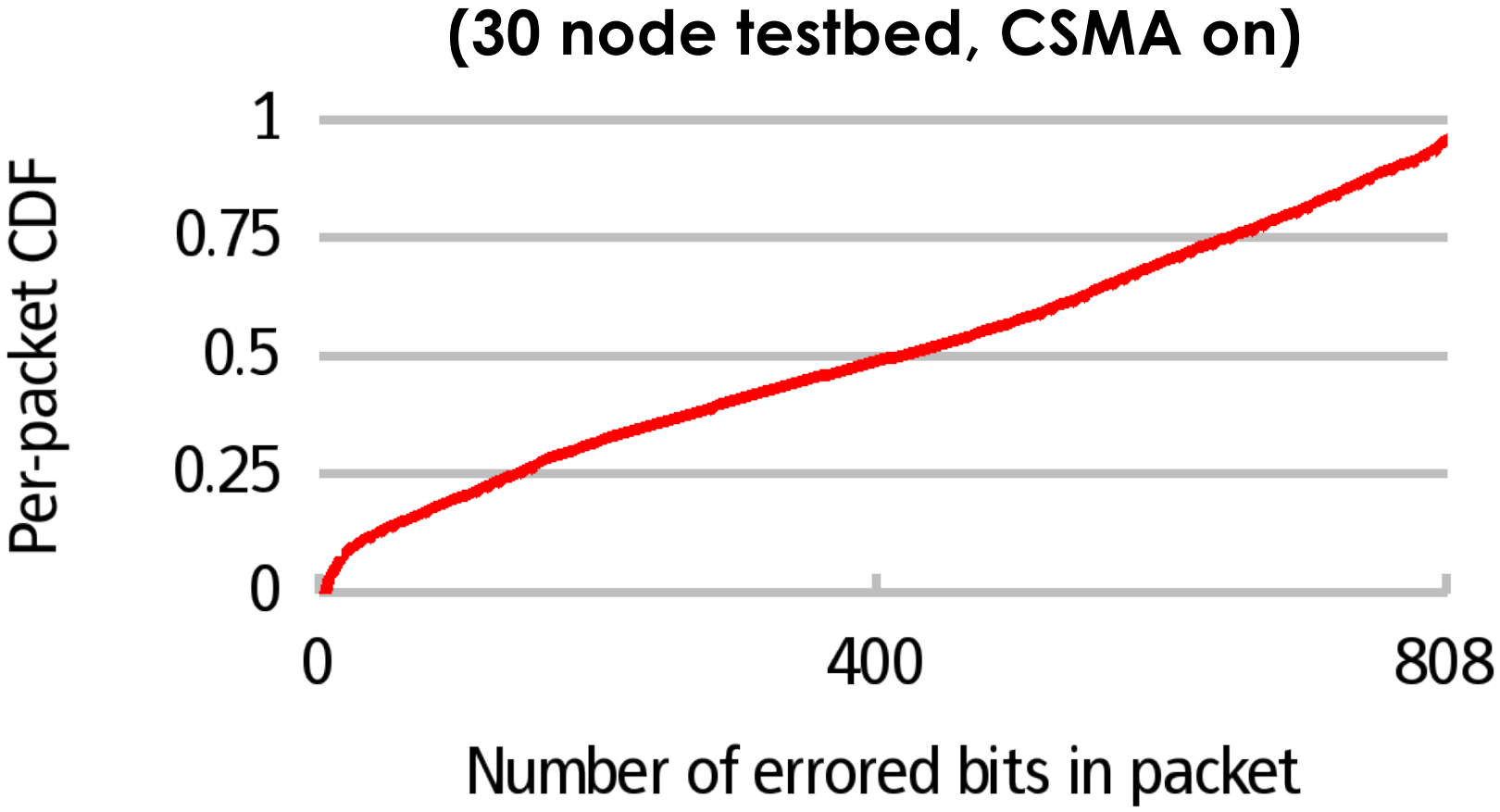
Kyle Jamieson and Hari Balakrishnan
CSAIL, MIT

What is Partial Packet Error?

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

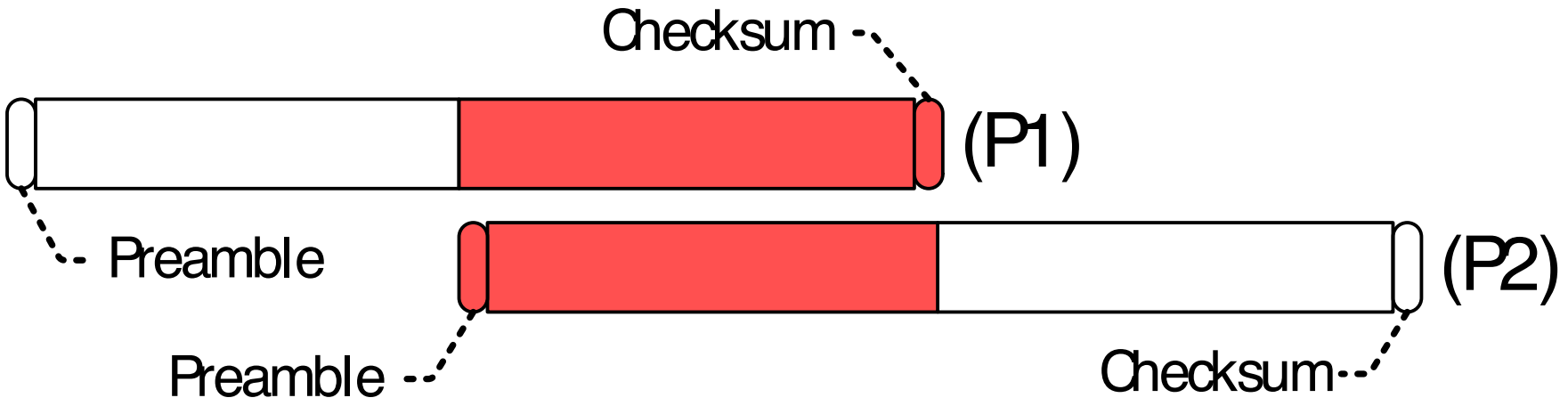
Can't receive non-colliding bits today!

Bits in a packet don't share fate



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

Three Key Questions



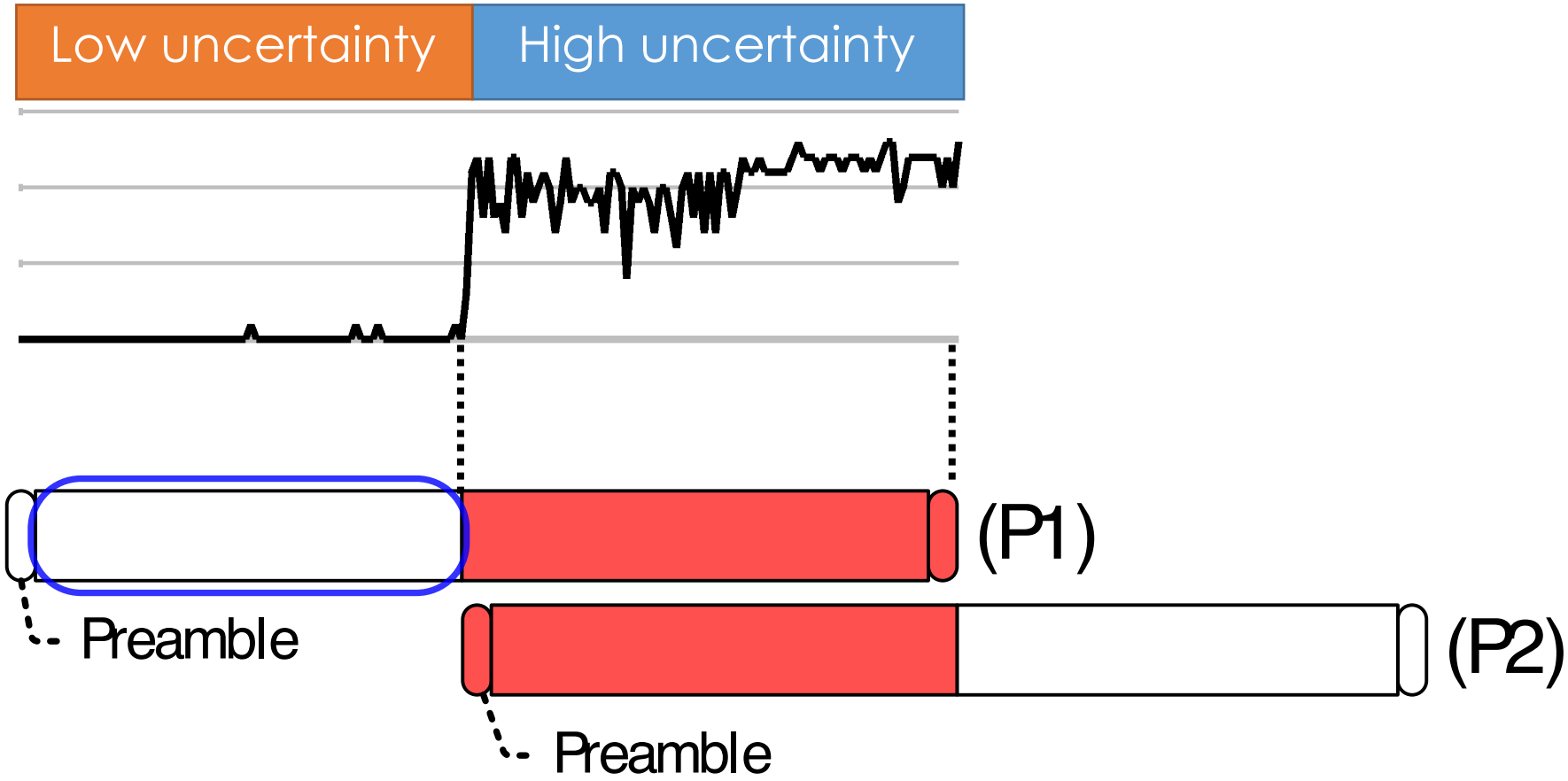
1. How does receiver know which bits are correct?
2. How does receiver know P2 is there at all?
3. How to design an efficient ARQ protocol?

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 and other modulations

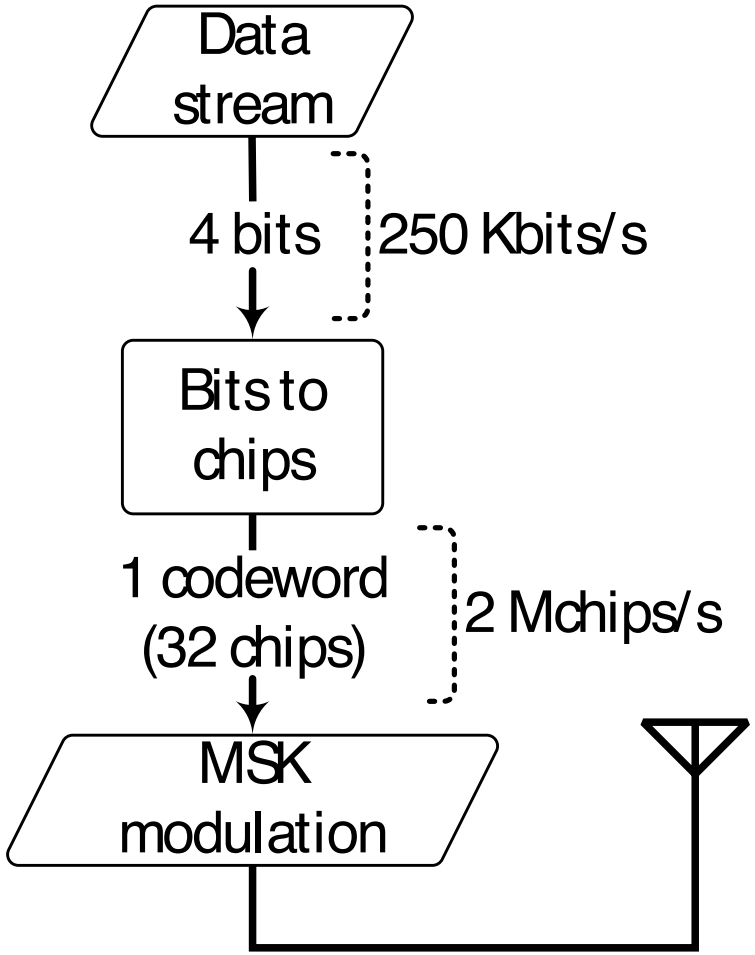
Can We Leverage Soft Info?

PHY conveys uncertainty in each bit it delivers up



Direct Sequence Spread Spectrum

Transmitter:



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: 11101101000111000011010110100010
C_1: 11101101100111000011010100100010

→ SoftPHY hint is **2**

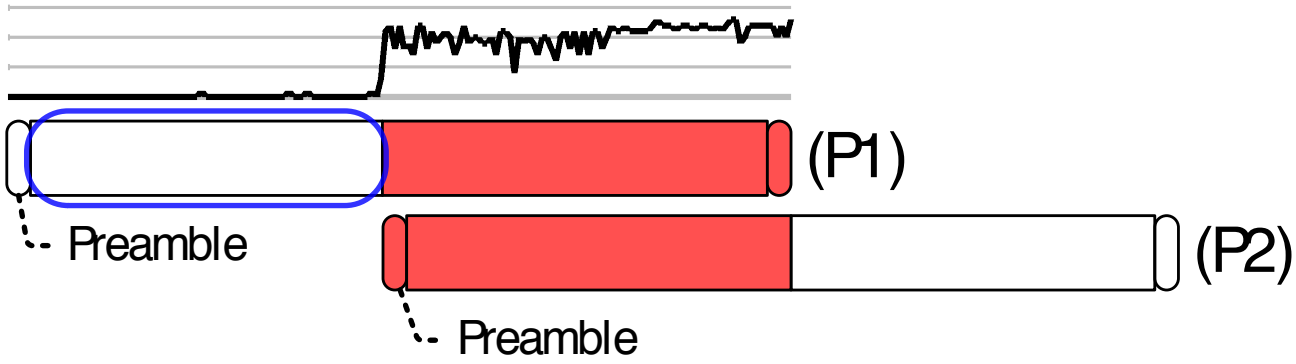
Receive: 11001101000111010111011101101111
C_1: 11101101100111000011010100100010

→ SoftPHY hint is **9**

Three Key Questions

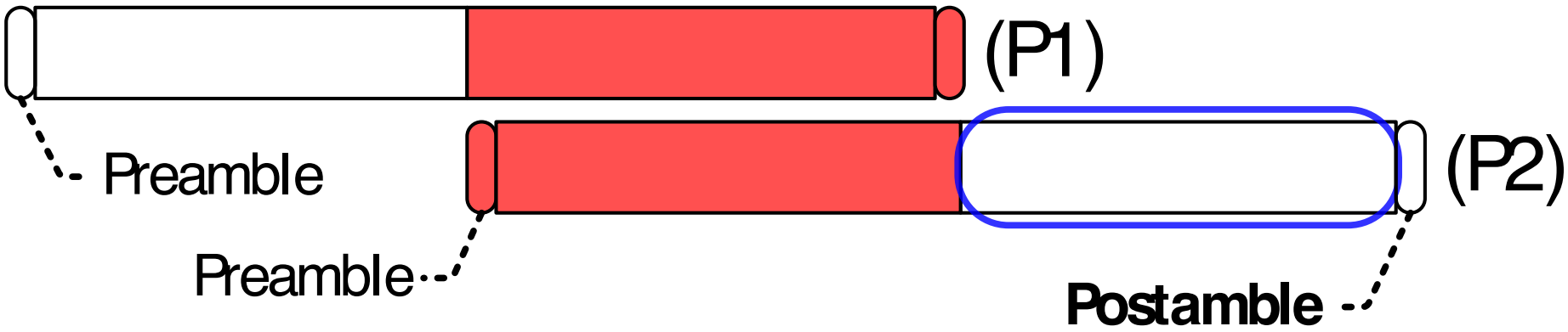
1. How does receiver know which bits are correct?

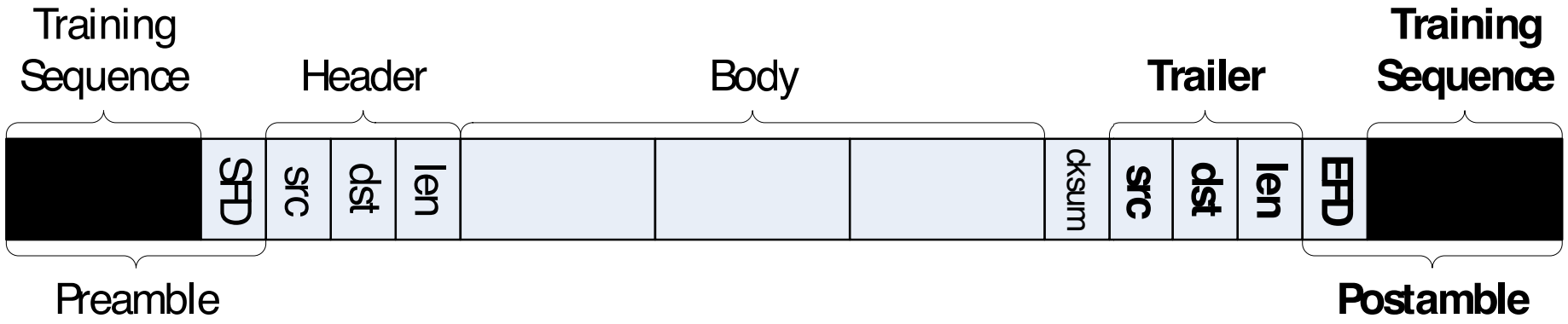
A: SoftPHY:



2. **How does receiver know P2 is there at all?**
3. How to design an efficient ARQ protocol?

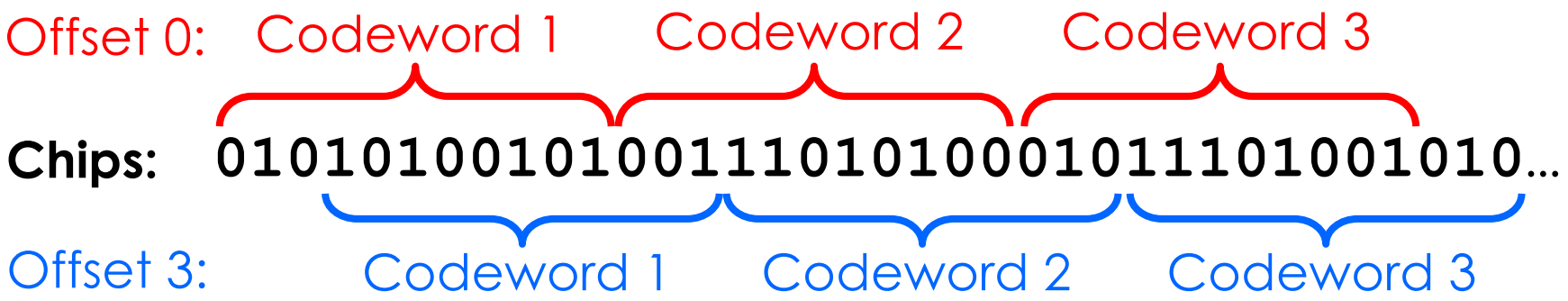
Postamble decoding





Receiver Design with Postamble

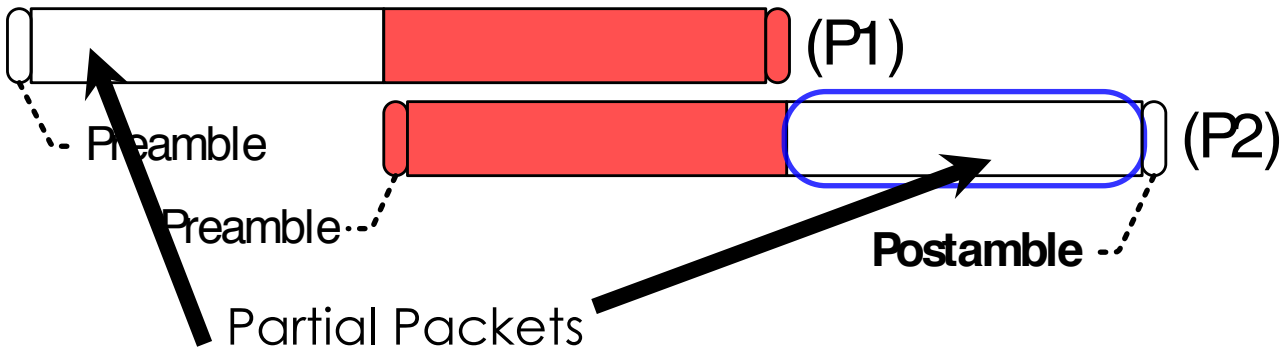
- Codeword synchronization
  - Translate stream of chips to codewords
  - Search for **postamble** at all chip offsets



Three Key Questions

1. How does receiver know which bits are correct?
2. How does receiver know P2 is there at all?

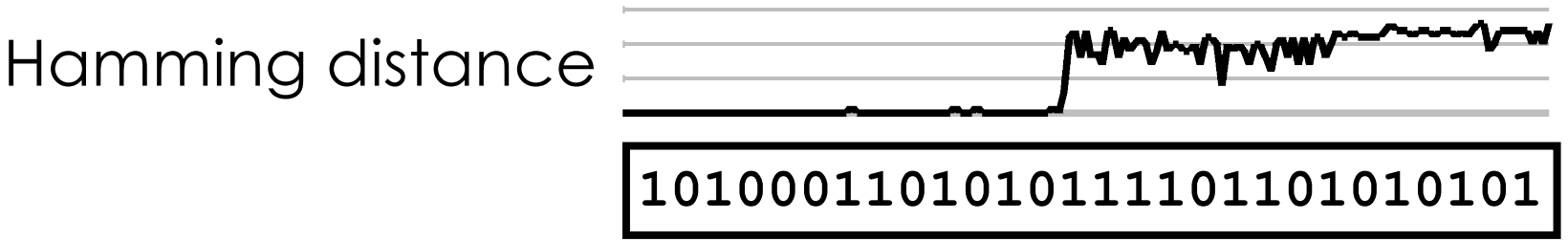
A: Postamble:



3. **How to design an efficient ARQ protocol?**

ARQ with partial packets

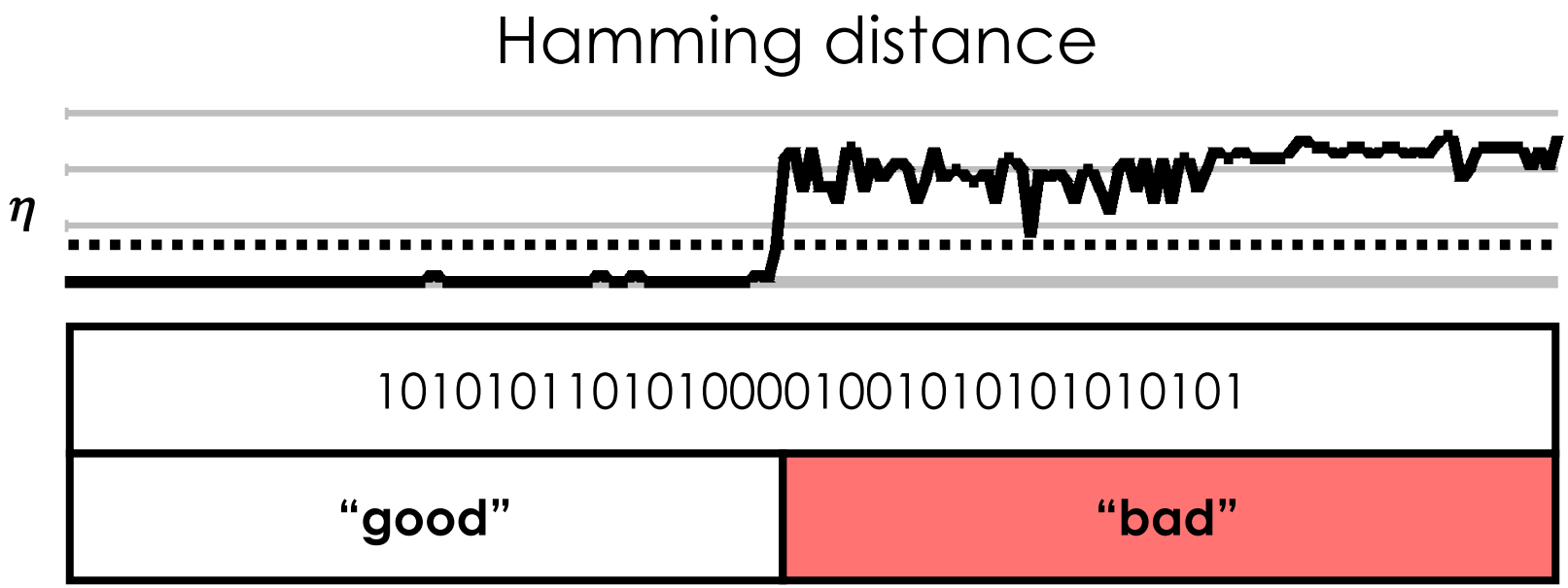
- ARQ today: correctly-received bits get resent
- **PP-ARQ** key idea: resend **only** incorrect bits



- **Efficiently** tell sender about what happened
  - Feedback packet

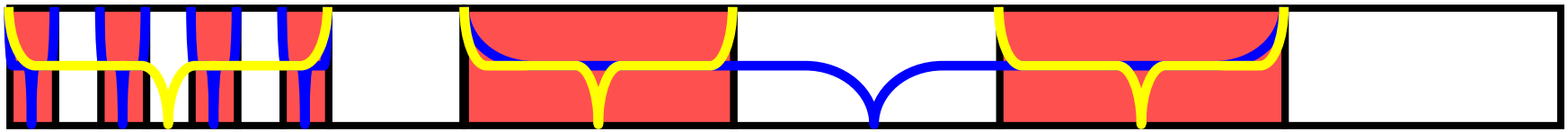
Labeling Bits “good” or “bad”

- **Threshold test**: pick a threshold η
  - Label codewords with SoftPHY hint > η **“bad”**
  - Label codewords with SoftPHY hint ≤ η **“good”**



PP-ARQ protocol

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

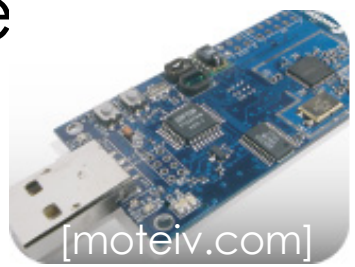


2. Codewords are in fact **correct** or **incorrect**
   - Two possibilities for mistakes
     - Labeling a correct codeword "bad"
     - Labeling an incorrect codeword "good"

Implementation

Sender: telos tmote sky sensor node

- Radio: CC2420 DSSS/MSK (Zigbee)
- Modified to send **postambles**



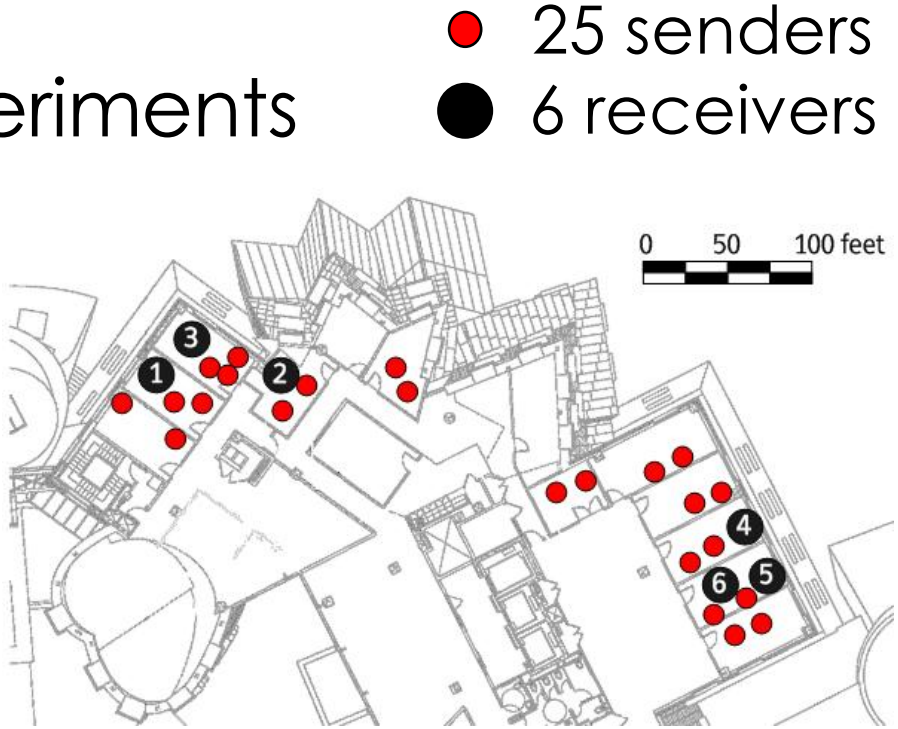
Receiver: USRP software radio with 2.4 GHz RFX 2400 daughterboard

- Despreading, postamble synchronization, demodulation
- **SoftPHY** implementation

PP-ARQ: trace-driven simulation

Experimental design

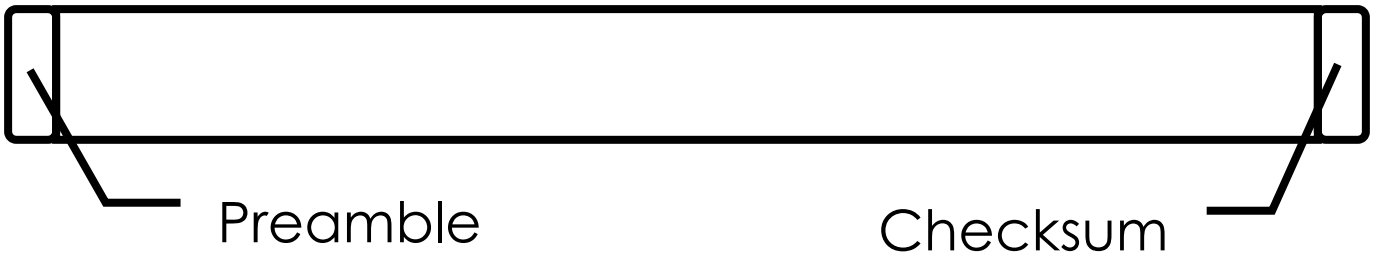
- Live wireless testbed experiments
  - Senders transmit 101-byte packets, varying traffic rate
  - Evaluate raw PPR throughput
  - Evaluate SoftPHY and postamble improvements



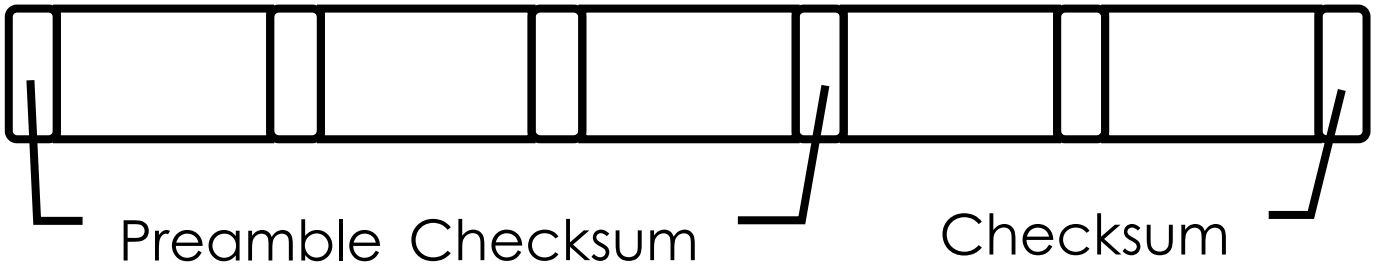
- Trace-driven experiments
  - Evaluate end-to-end PP-ARQ performance
  - Internet packet size distribution
  - 802.11-size preambles

PP-ARQ performance comparison

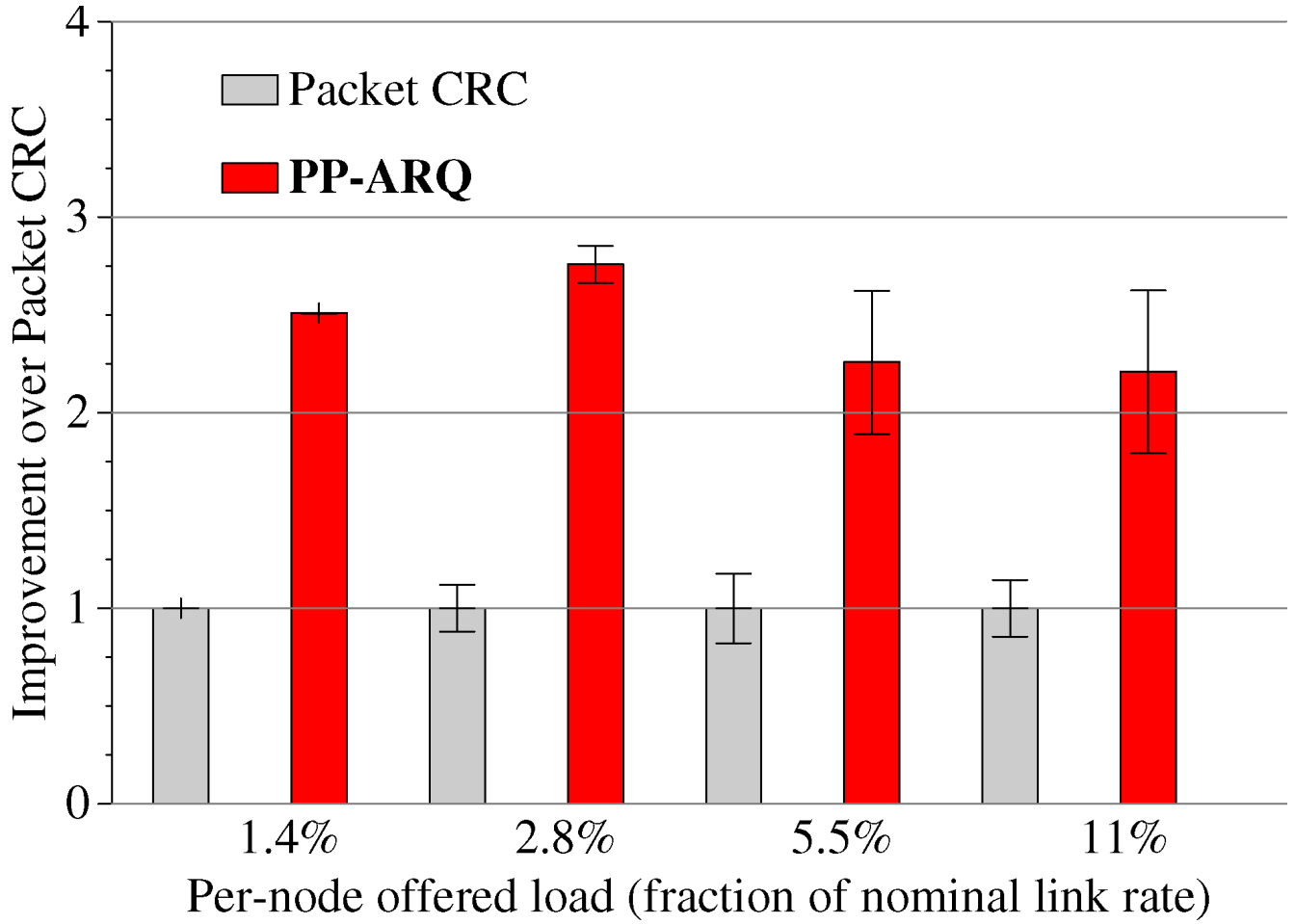
- Packet CRC (no postamble)



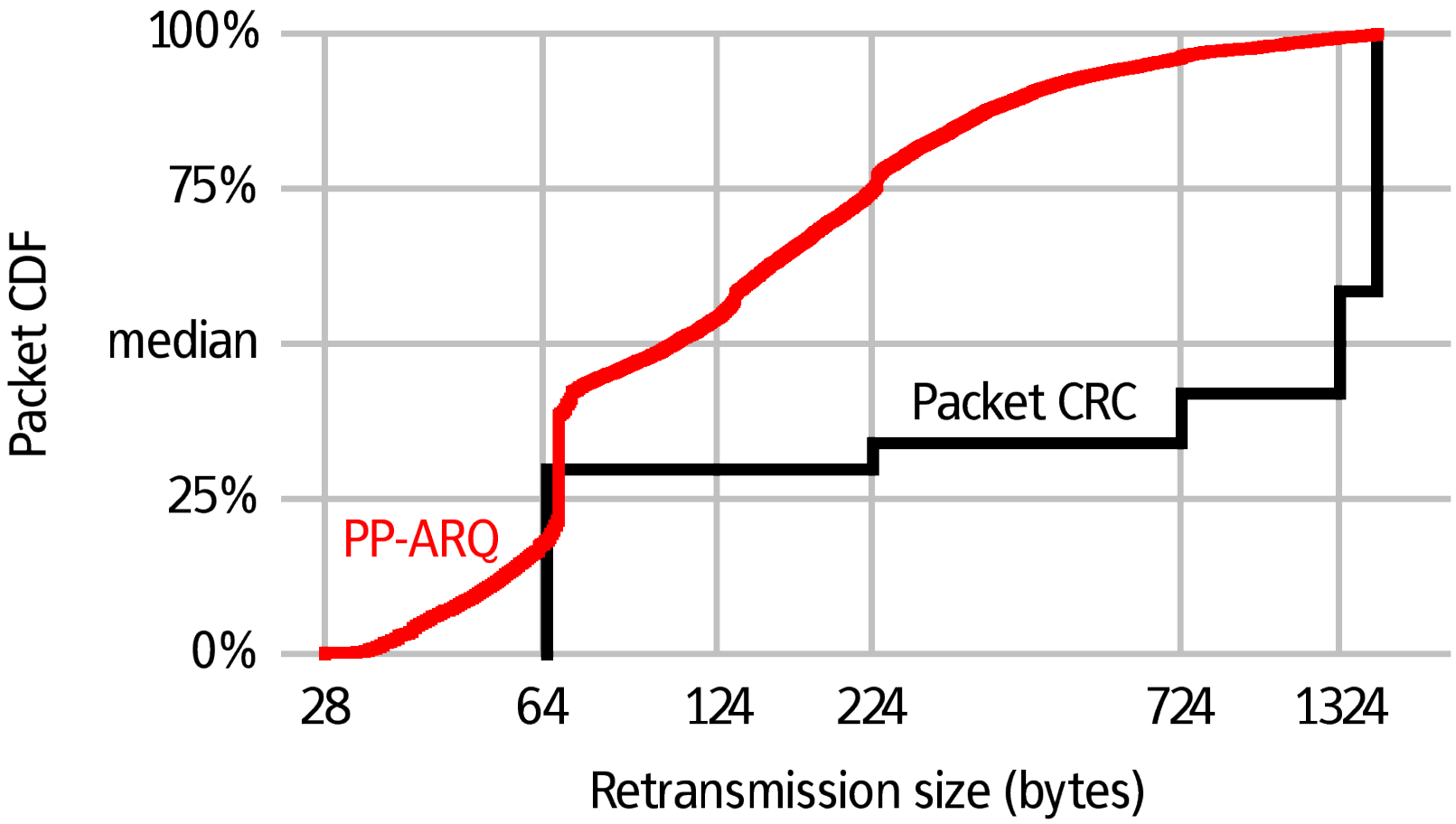
- Fragmented CRC (no postamble)
  - Tuned against traces for optimal fragment size



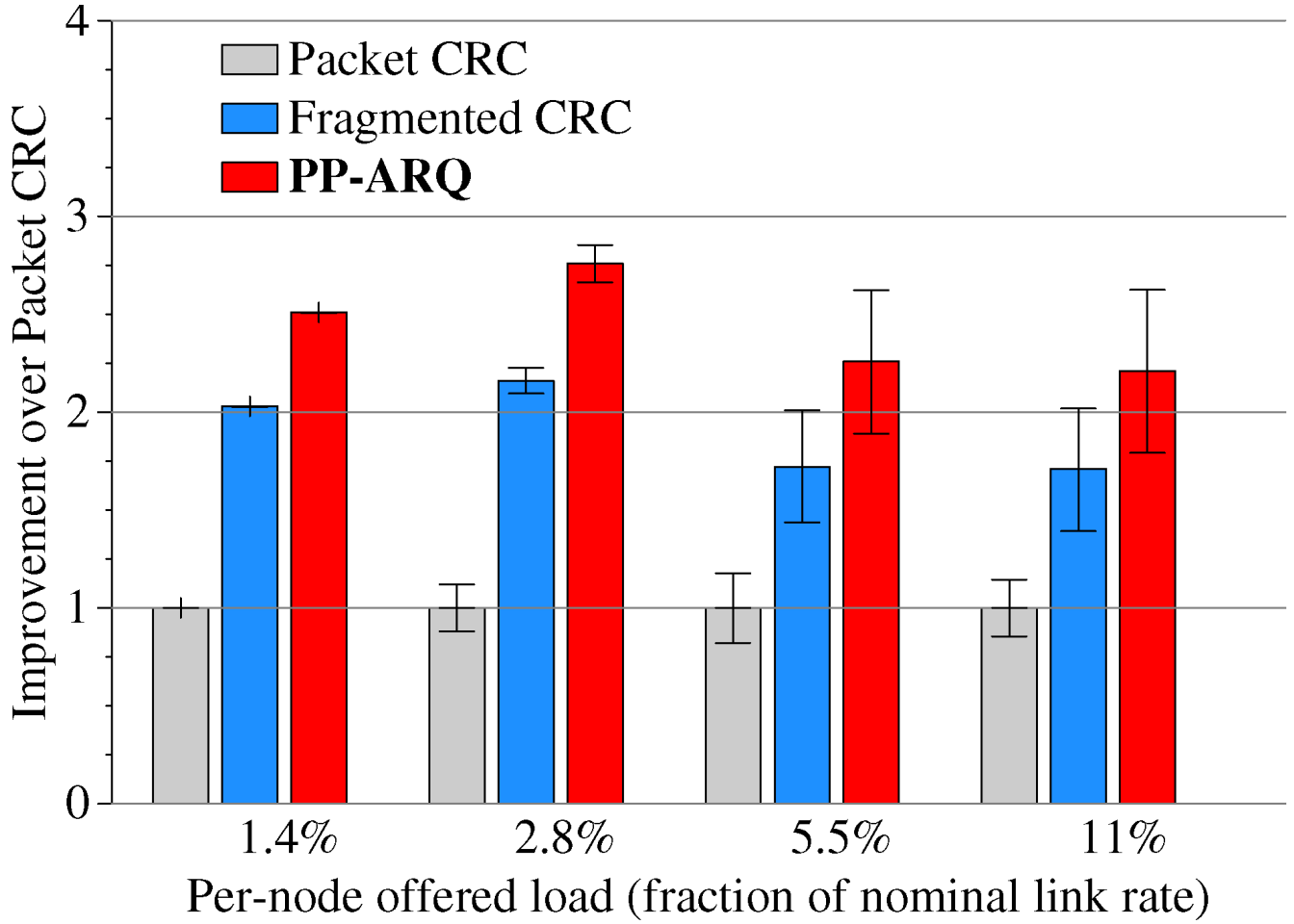
Throughput Gain: 2.3-2.8x



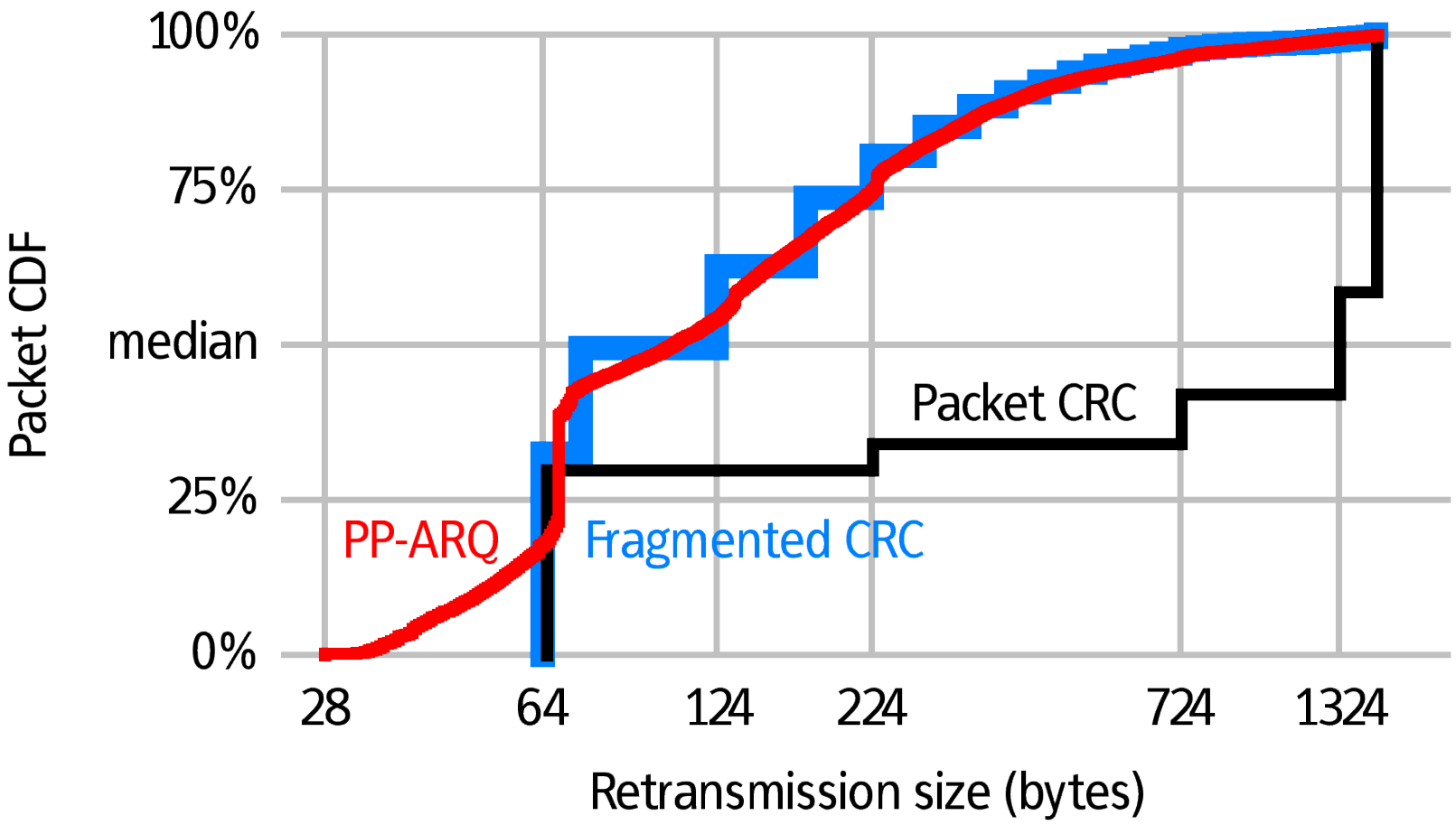
PP-ARQ Retransmissions are Short



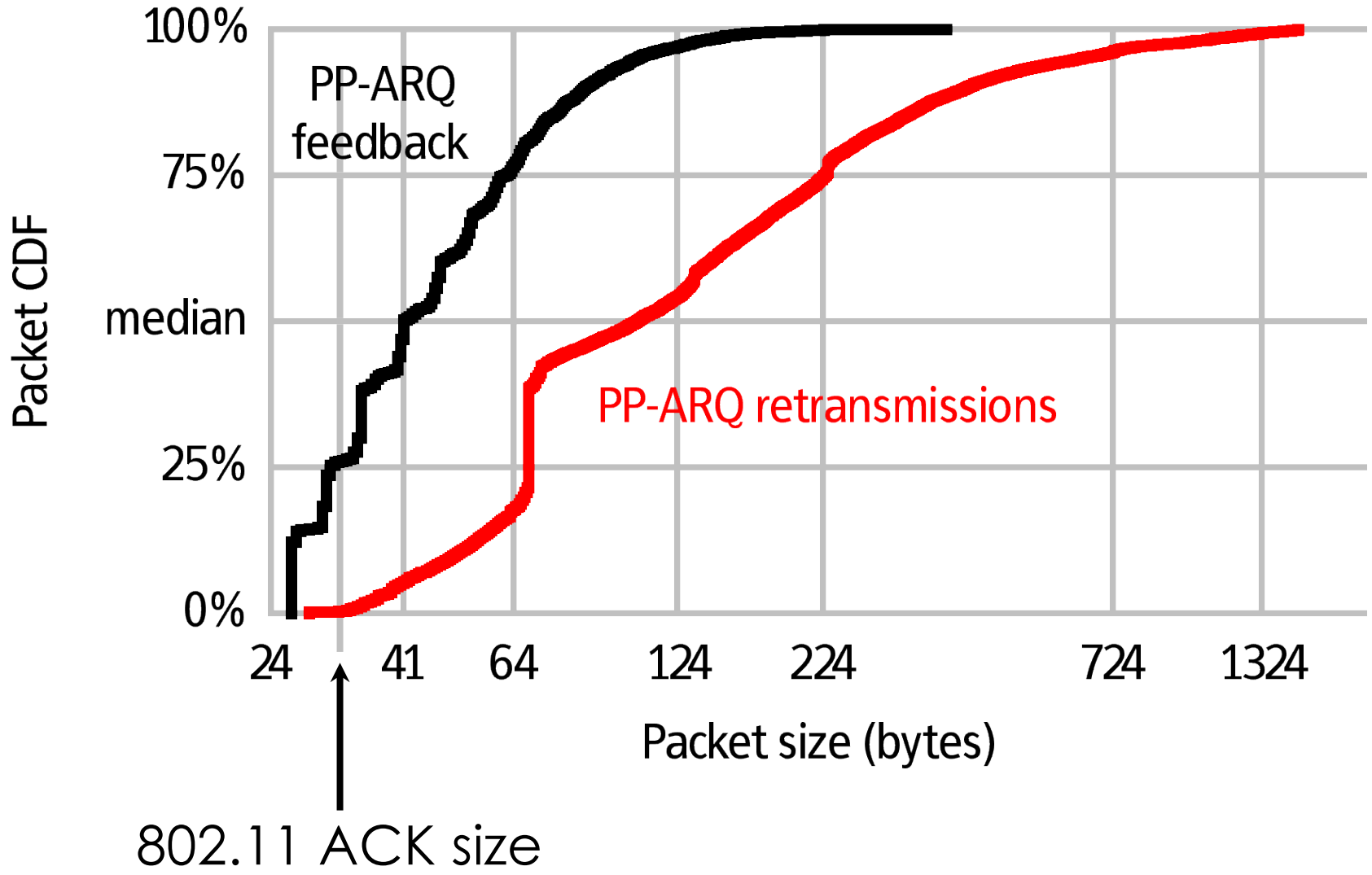
25% Gain over Fragmented



PP-ARQ Retransmissions are Short



Low PP-ARQ Feedback Overhead



Related work

- ARQ with memory [Sindhu, IEEE Trans. On Comm. '77]
  - Incremental redundancy [Metzner, IEEE Trans. On Comm. '79]
  - Code combining [Chase, IEEE Trans. On Comm. '85]
- Combining retransmissions
  - **SPaC** [Dubois-Ferrière, Estrin, Vetterli; SenSys '05]
- Diversity combining
  - Reliability exchanging [Avudainayagam et al., IEEE WCNC '03]
  - **MRD** [Miu, Balakrishnan, Koksal; MobiCom '05]
  - **SOFT** [Woo et al.; MobiCom '07]
- Fragmented CRC
  - **Seda** [Ganti et al.; SenSys '06], 802.11 fragmentation

Conclusion

- 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