



Bit-Switched Wireless Networks

**Increasing Throughput By Increasing
Concurrency**

**Hari Balakrishnan
MIT CSAIL**

With Kyle Jamieson

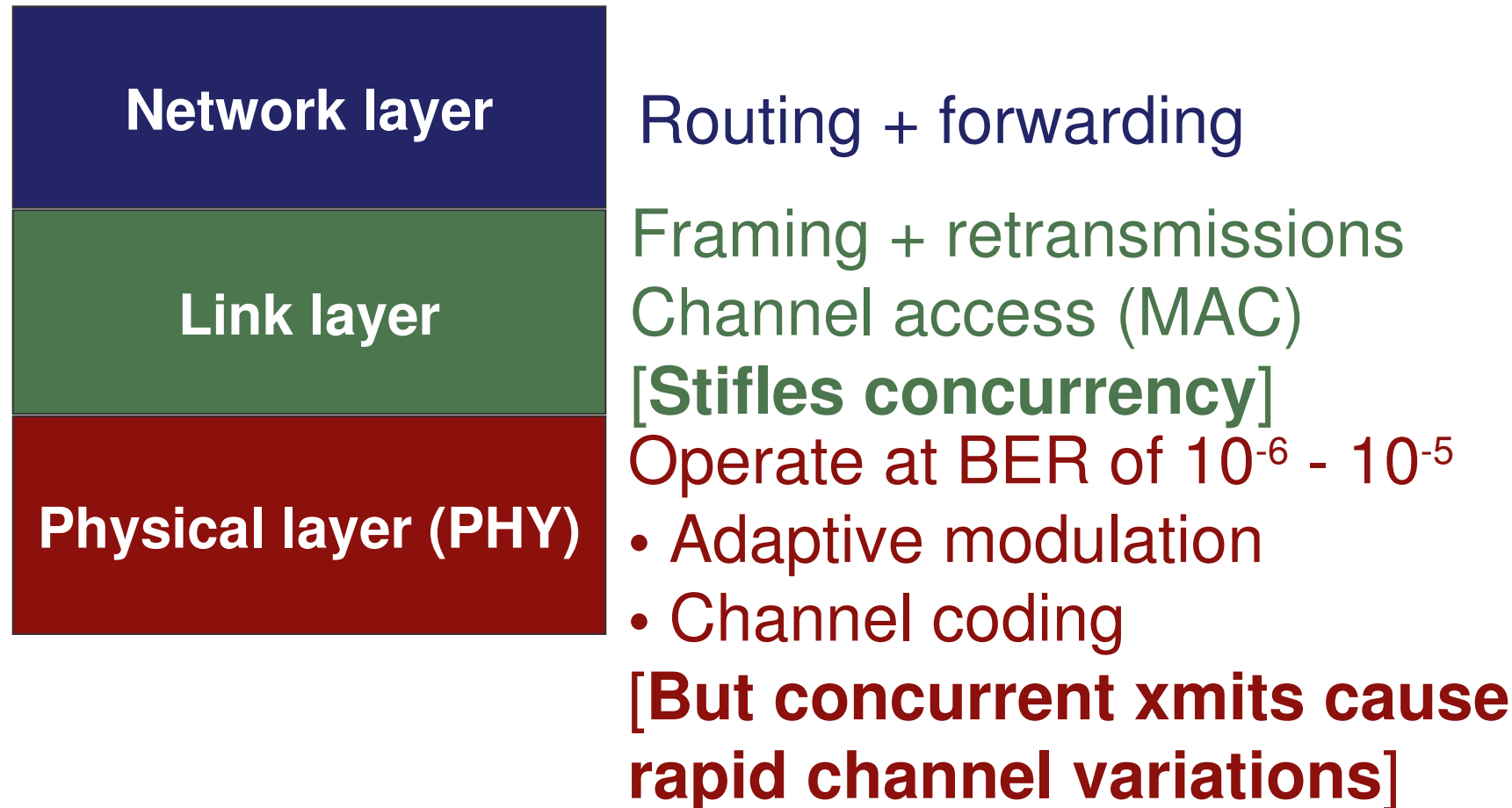
**This talk builds on the partial packet recovery
scheme described at SIGCOMM 2007**

Radio Propagation is Complex

- Much harsher medium than a wire
- Data delivery depends on many factors
 - **Interference**
 - Noise, attenuation (distance), fading, multi-path
 - Mobility (of both nodes and in environment)
- Fundamentally, a shared medium with hard-to-model time-varying properties
- Result: Poor *aggregate network* throughput and capacity

Coping with Radio Complexity: Current Approach

- Hide vagaries under abstractions that make a radio look like a wire



The Approach: Change the Abstraction

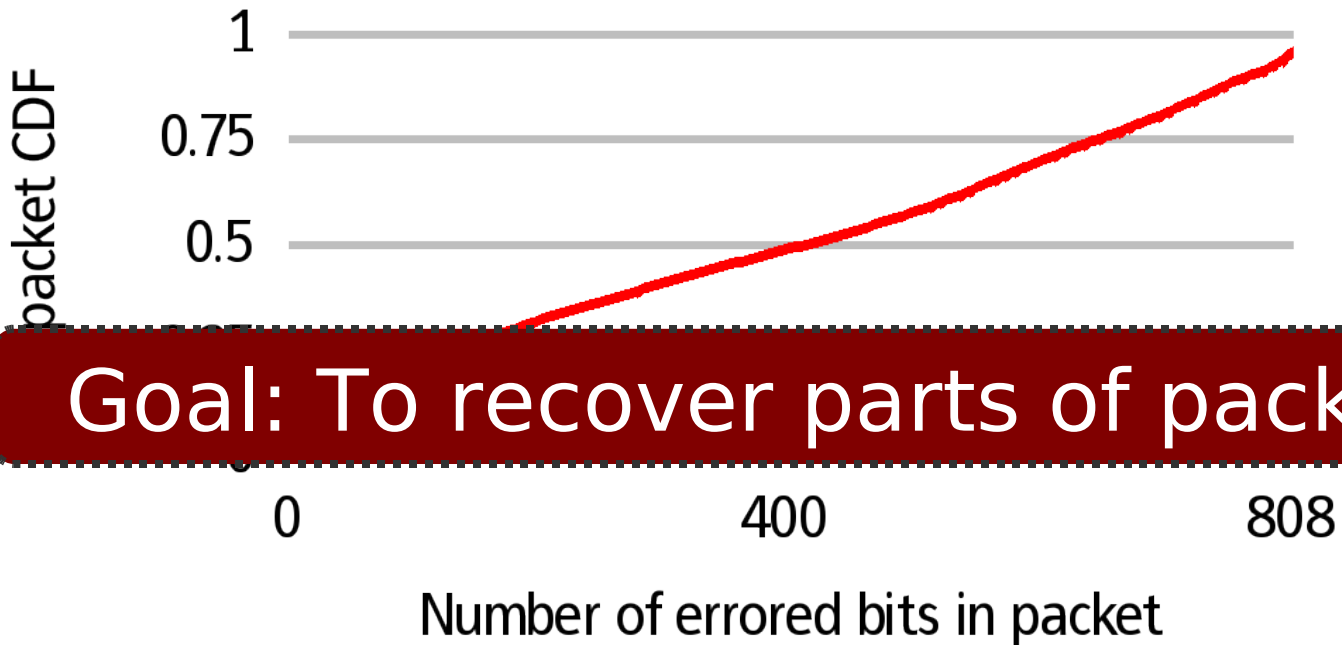
- Current systems have two ideas:
 - Layering, great idea
 - Wired abstraction, bad idea
- Approach: Don't mimic an “error free” wire
 - Embrace radio peculiarities
- Key idea is **SoftPHY**: A different abstraction
 - Receiver PHY annotates each bit it passes to higher layer with a decoding *confidence*
- *Granularity* of data recovery and routing is not a packet, but dynamic *groups of bits*
- Hence, a ***bit-switched wireless network***



Outline

- SoftPHY design and implementation
- Bit-switched network architecture
 - Using SoftPHY in various ways to improve capacity

Bits in a Packet Don't Share Fate



Goal: To recover parts of packets

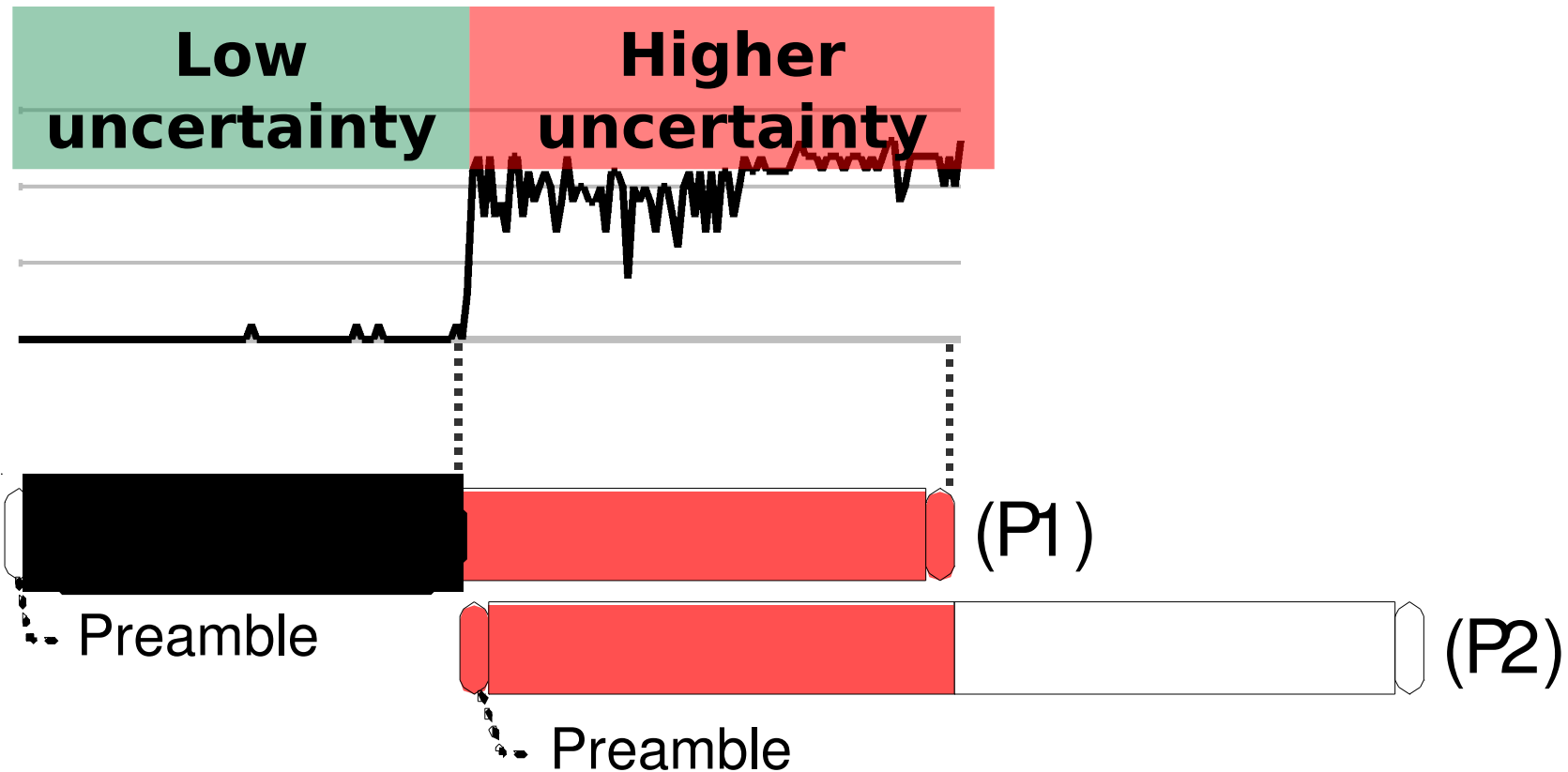
(30-node testbed, CSMA on)

Many bits from corrupted packets are correct,

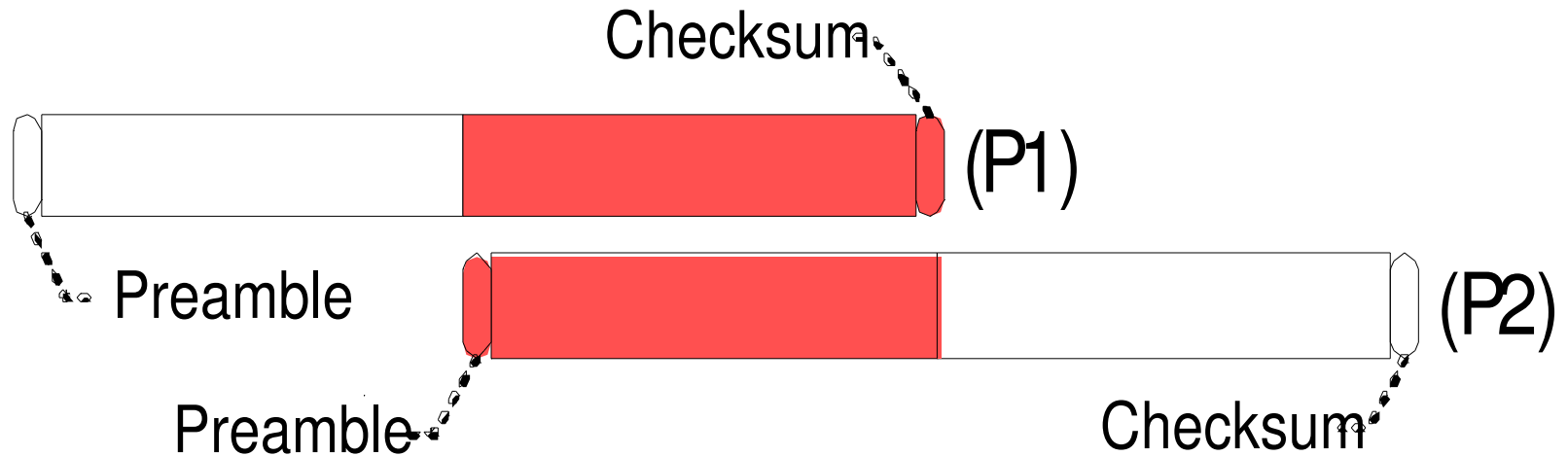
but current receivers don't know which!

SoftPHY: A New Source of Information

Receiver PHY conveys uncertainty in each bit
it delivers up to higher layers

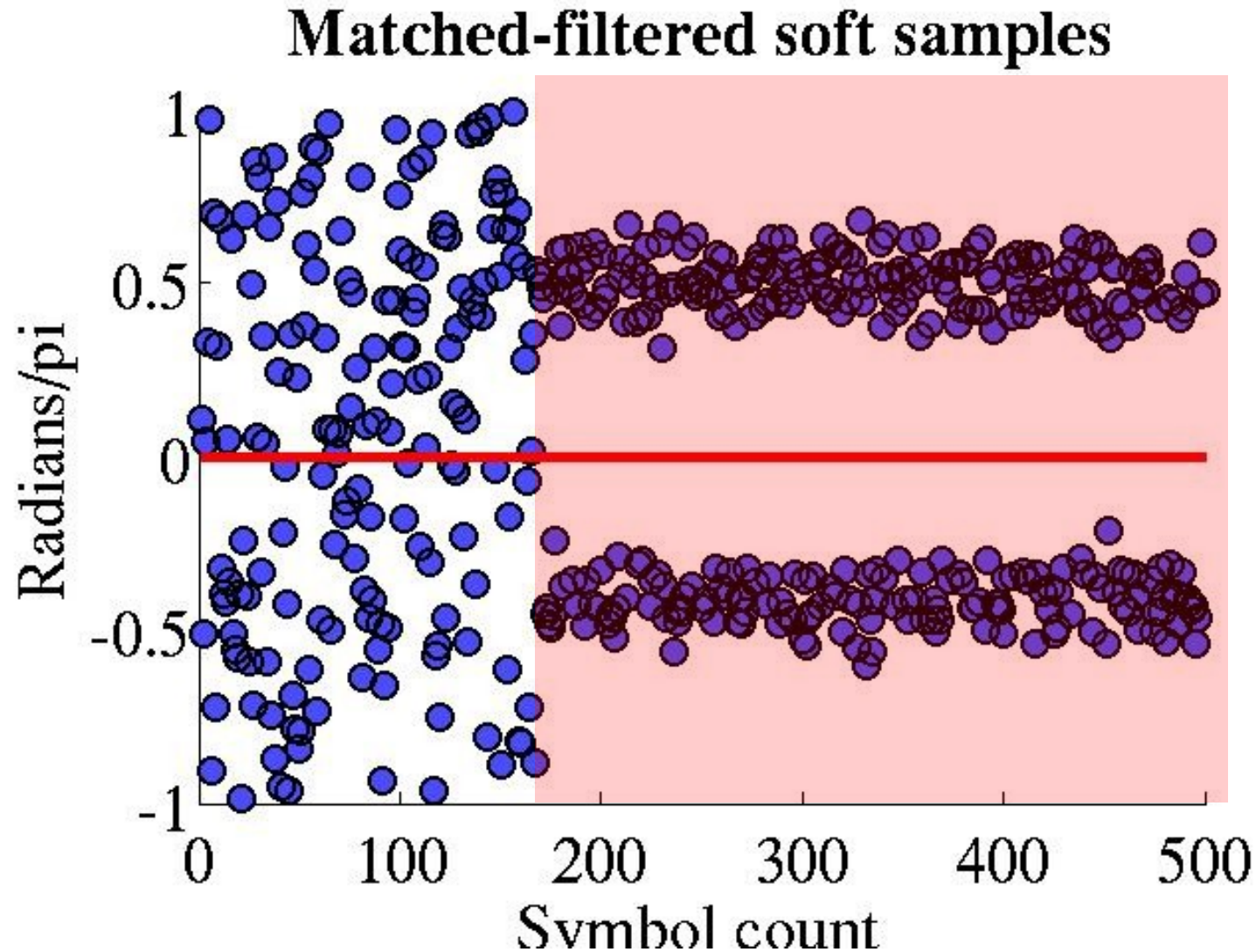


SoftPHY: Two Questions



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

SoftPHY Intuition: Receiver PHY's “Soft” Samples for Symbols

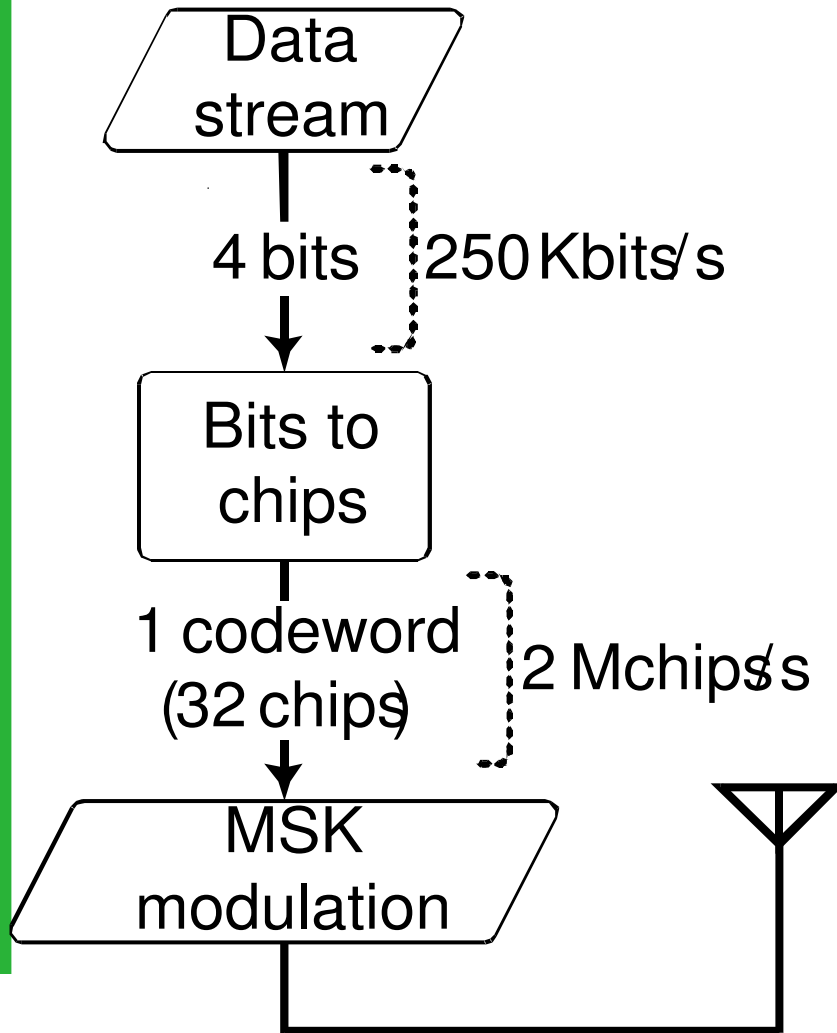


Identifying “Correct” Bits

- Receiver PHY has the information in its maximum-likelihood estimator
- Pass this confidence to higher layer as a *hint*
 - Inspired by soft-decision decoding
 - Fundamental difference: export to higher layers
 - Goal is to increase concurrency, not single-link performance
- SoftPHY implementation is PHY-specific; interface is PHY-independent
- Implemented for direct sequence spread spectrum (DSSS) over MSK and some other modulations
[Jamieson/Balakrishnan SIGCOMM 2007]

Direct Sequence Spread Spectrum Background

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 DSSS

Hamming distance between
received chips and **decided-upon**
codeword

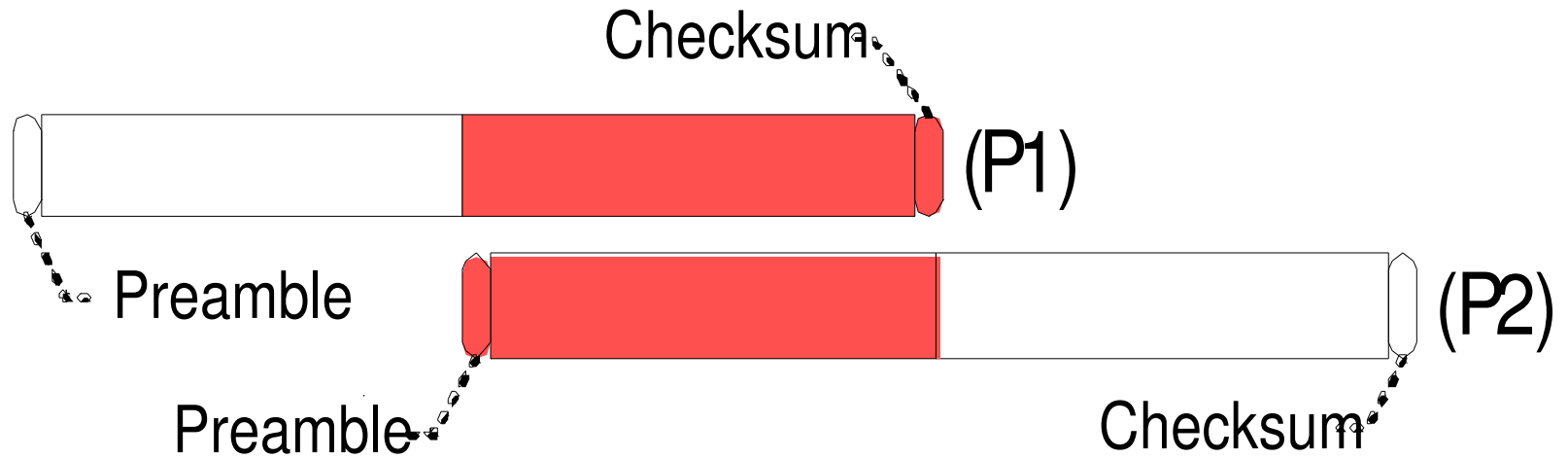
Receive: 11101101000011100000110101101000010
 C_1 : 11101101100011100000110101001000010

⇒ SoftPHY hint is
2

Receive: 110011010000111010111011110110111
 C_1 : 11101101100011100000110101001000010

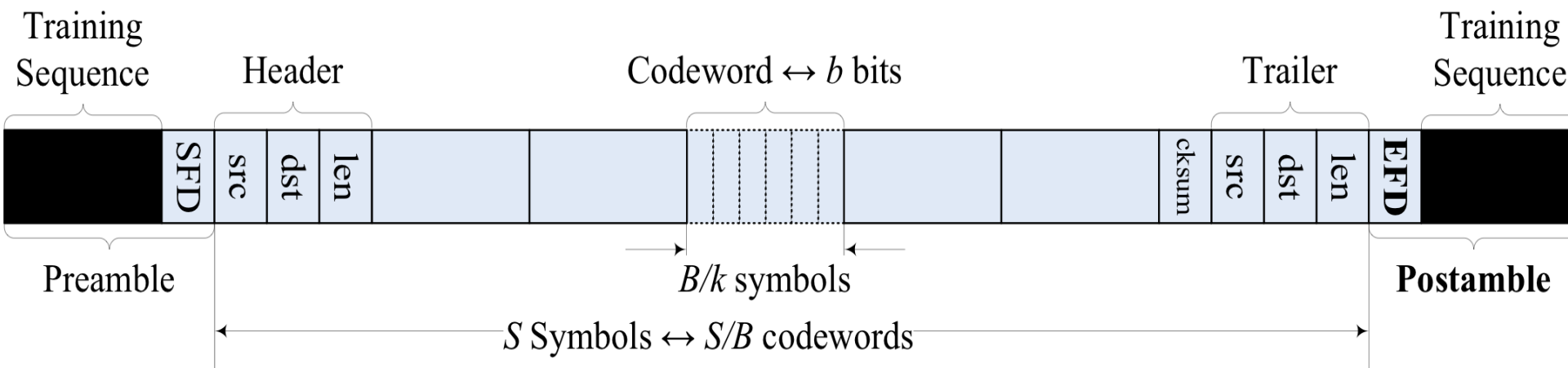
⇒ SoftPHY hint is **9**

SoftPHY: Two Questions



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

Coping with Preamble Corruption: Postamble Decoding



Details in paper [Jamieson/Balakrishnan SIGCOMM 2007]

How Well Does SoftPHY + Work?

Sender: CC2420 DSSS/MSK (Zigbee)

- Modified to send postambles

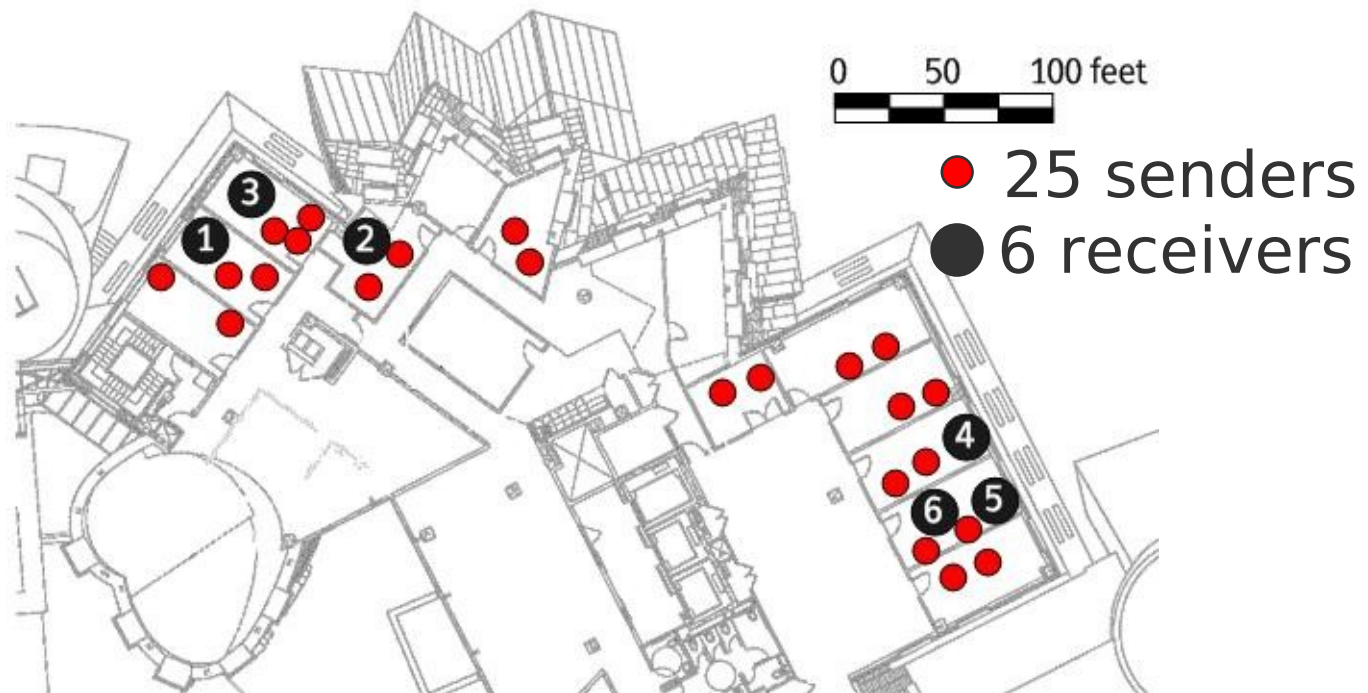


Receiver: USRP software radio with 2.4 GHz RFX 2400 daughterboard

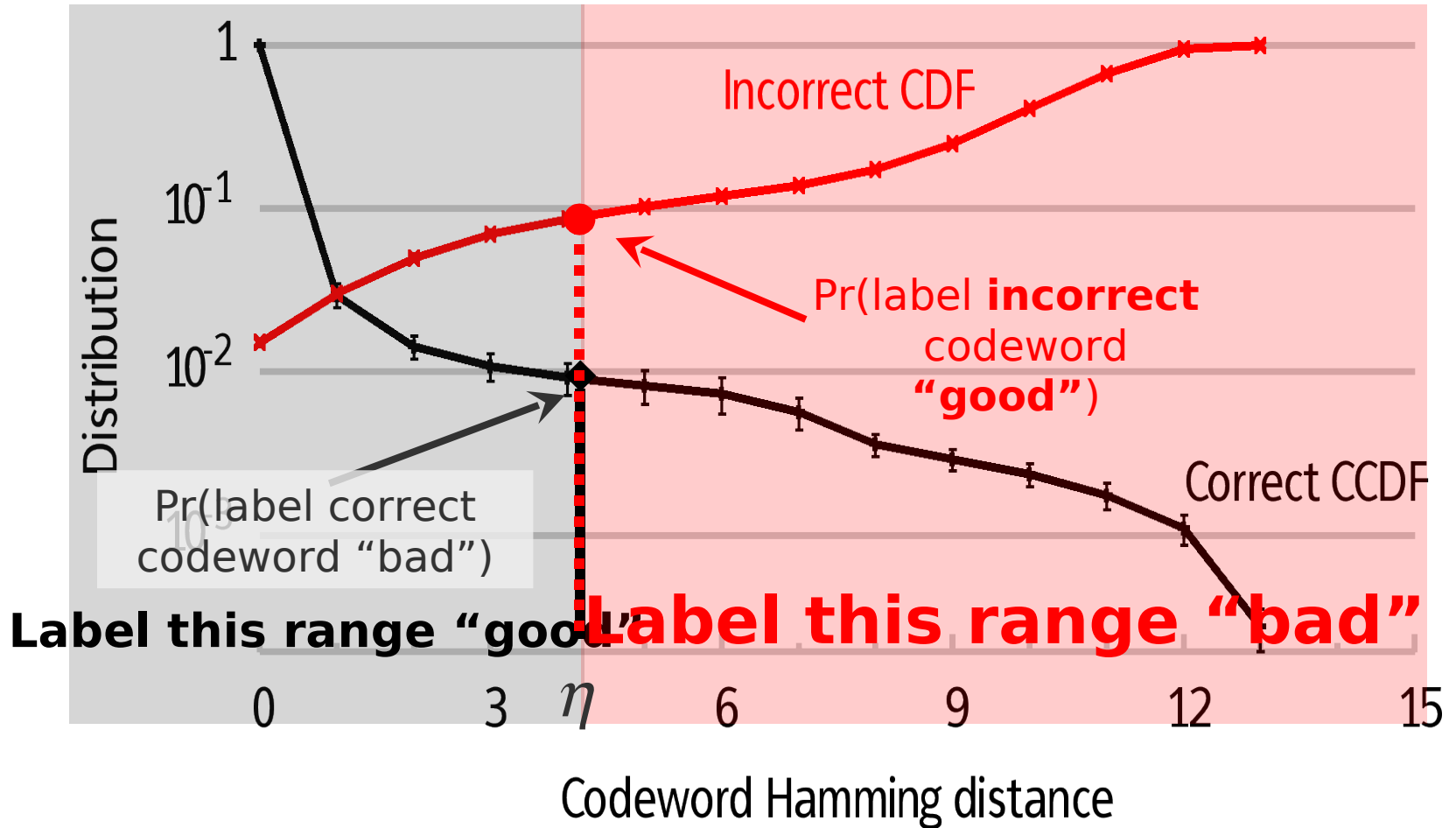
- Despreading, postamble synchronization, demodulation
- SoftPHY implementation in C/Python

Experiments on Testbed

- Senders xmit 101-byte packets, varying rate
- Measure SoftPHY and postamble improvements

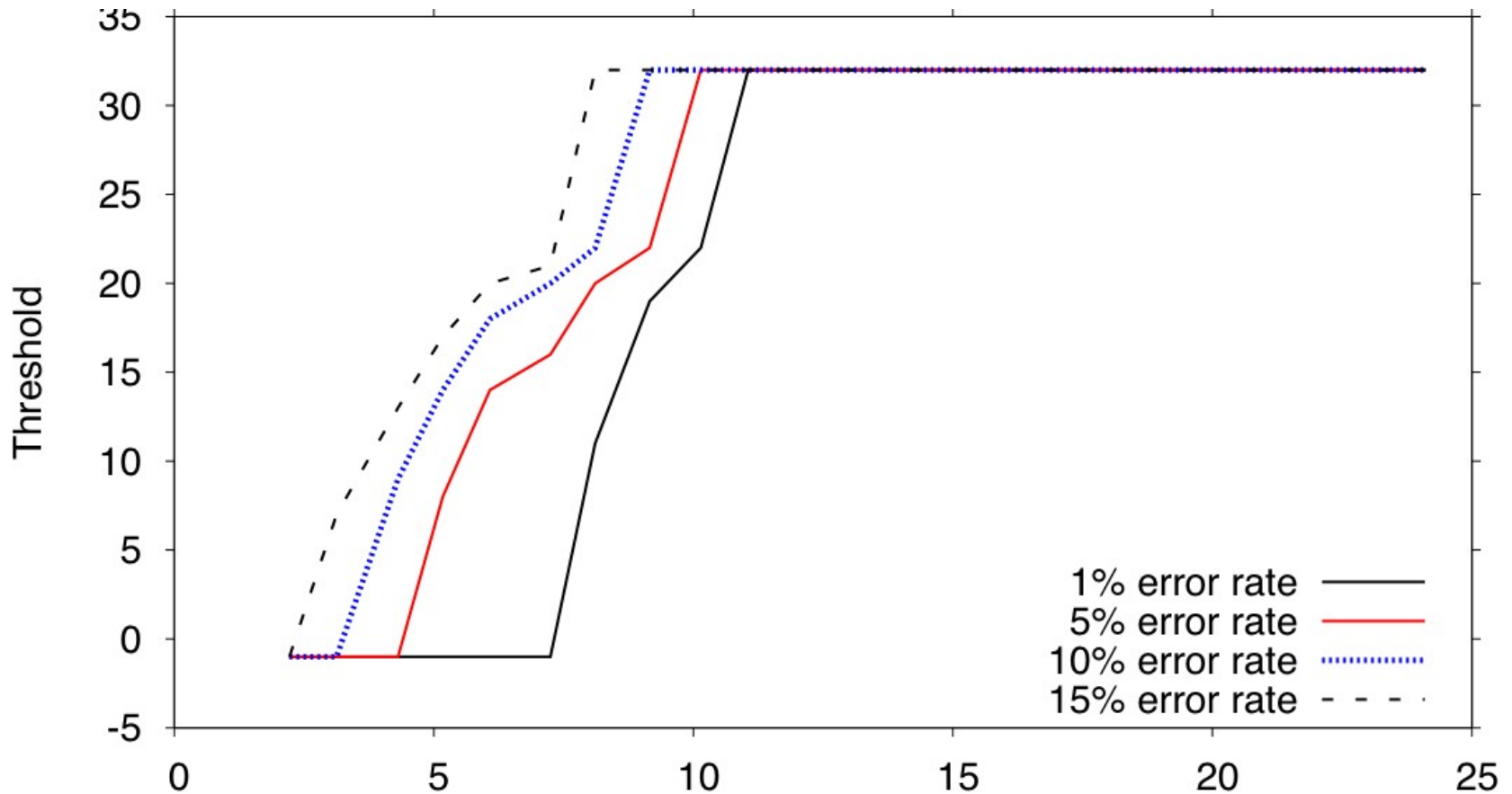


SoftPHY Hints are Robust



Similar results under wide range of traffic load, power levels, interference conditions

Optimization: Pick Threshold Using SINR Information



Each curve shows threshold v SINR at different SoftPHY desired error rates
So, given a target, pick using current SINR



Outline

- SoftPHY design and implementation
- Bit-switched network architecture
 - Using SoftPHY to improve capacity

SoftPHY Embraces Radio “Peculiarities”

- Bits in a packet don't all share the same fate
 - SoftPHY gives *partial packet recovery* of ‘good’ bits
- Trade-off between BER and data rate
 - With SoftPHY, links can run at high BER (10^{-3} - 10^{-2})
 - Higher BER-tolerance ==> much higher data rates
- Successful data delivery depends on conditions near receiver, not sender
 - Concurrent transmissions interact in complex ways
 - SoftPHY allows receiver to determine interferers
- Radio is inherently broadcast
 - SoftPHY enables better cooperative combining

Bit-Switched Wireless Networks

Forward only
“good” bits
(coop. mesh networks)

PP-ARQ (rxmit
only “bad”)

Bits

PHY

Multi-radio combining
(access points)



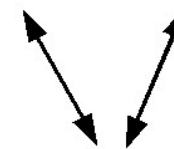
Remote Host

Divert Controller (DC)

Wired Backbone



AP/Divert Monitor (DM)



Mobile
Receiver

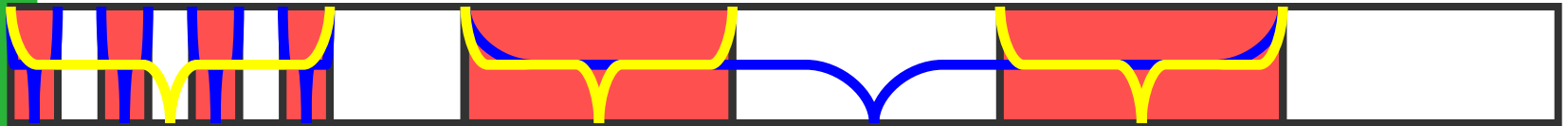
Miu et al., Mobicom '05

PP-ARQ: ARQ with Partial Packets

- ARQ today: even correctly received bits get resent
- PP-ARQ key idea: resend only incorrect bits using receiver feedback

PP-ARQ Protocol: Challenges

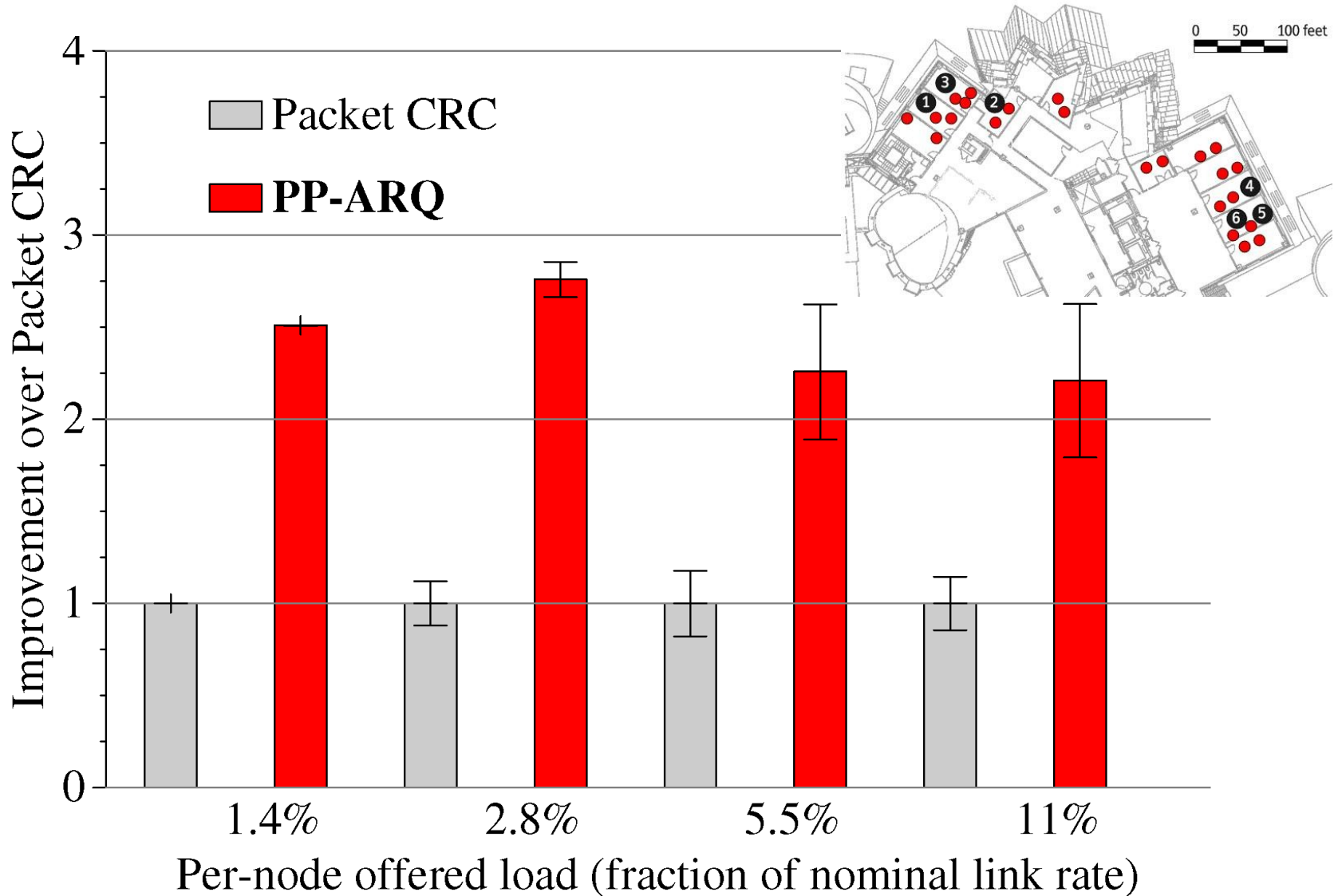
- " Assuming hints correct, which ranges to ask?
 - Dynamic programming problem: need to aggregate ranges carefully
 - Account for both forward and feedback channels



□ 'Good' bits
■ 'Bad' bits

- But codewords are in fact **correct** or **incorrect**
 - Mislabeled a correct codeword "bad"
 - Mislabeled an incorrect codeword "good"

PPR+ARQ Improves Throughput 2.3-2.8x



Related Work

- ARQ with memory [Sindhu IEEE Trans. Comm. '77]
 - Incremental redundancy [Metzner, IEEE Trans. Comm. '79]
 - Code combining [Chase, IEEE Trans. Comm. '85]
- Soft-decision decoding uses soft information at PHY
 - Works well for noise, doesn't handle multiple transmissions
 - Smears confidence across many bits to decode
- Diversity combining
 - Reliability exchanging [Avudainayagam et al., IEEE WCNC '03]
 - MRD [Miu, Balakrishnan, Koksal, MobiCom '05]
 - SOFT [Woo et al., MobiCom '07]
- Combining retransmissions
 - SPaC [Dubois-Ferrière, Estrin, Vetterli; SenSys '05]
- Scores of channel access schemes

The Status Quo is Too Conservative

Link recovery & routing
unit

Whole packet

Channel access

Conservative: CSMA and/or
RTS/CTS

Rate adaptation

Too conservative: decrease
rate, add redundancy in
coding

PHY coding

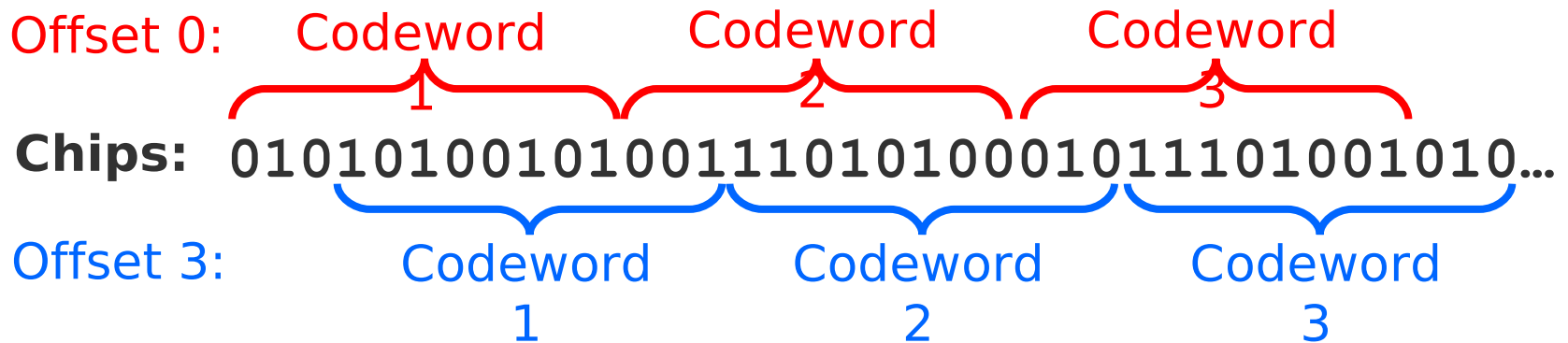
- With SoftPHY, protocols don't assume the worst when bits are corrupted
- Groups of bits are better granularity
- Leads to higher concurrency and throughput

Conclusion

- Exterminate the wired abstraction for wireless nets!
 - Embrace radio peculiarities, don't hide them
- SoftPHY enables *bit-switched wireless networks*
 - Propagate PHY information as hints to higher layers
 - Change atomic unit of recovery/forwarding to group of bits
- SoftPHY abstraction has many uses: ARQ, concurrent MAC, coop. forwarding, bitrate adaptation, ...
- An approach that deserves detailed study and development; promises dramatic throughput gains

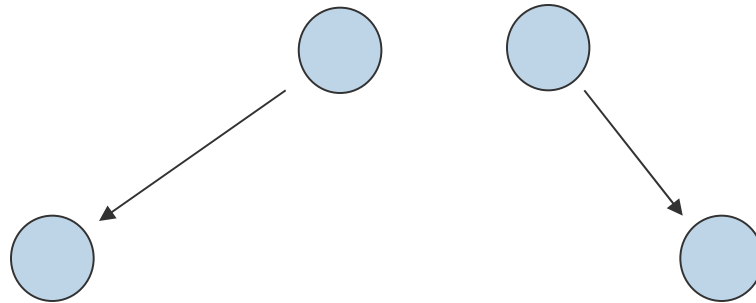
Receiver Design with Postamble

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



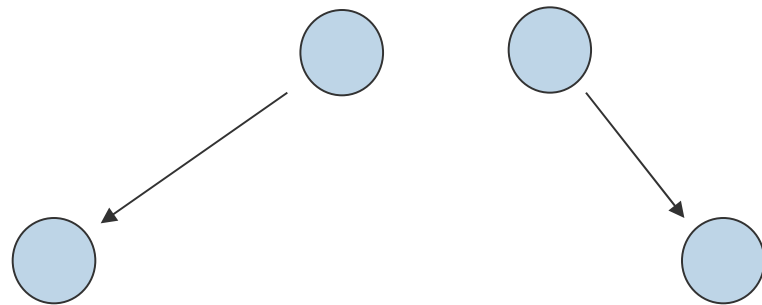
- Chip synchronization without preamble/postamble

CMAP: Reactive Channel Access

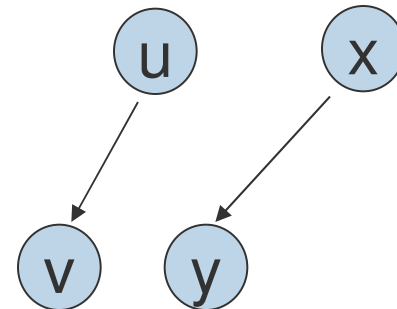


- Current channel access schemes are proactive
 - Presume that collisions will occur, defensive
- Instead, what if each node “just sends”?
 - SoftPHY can help recover from collisions, so don’t focus on minimizing collisions
 - Try to maximize throughput (concurrency)

Building a Conflict Map



No conflicts



$(u \rightarrow v, x \rightarrow y)$ *conflict*

- “Just send” works OK, but we can do better
- Conflict map: Distributed data structure built from empirical observations of bit receptions
 - Determines which communications *cannot* be concurrent because sequential throughput better
 - Use SoftPHY to determine interferers correctly

Determine Interferers at a Receiver

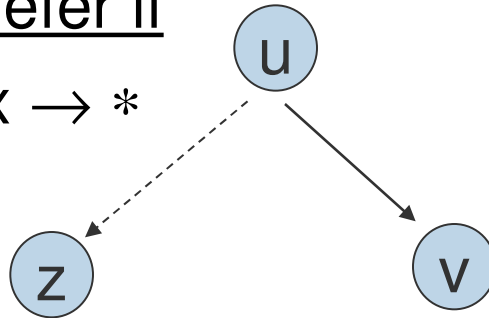
Des

t

Defer if

v

$x \rightarrow *$



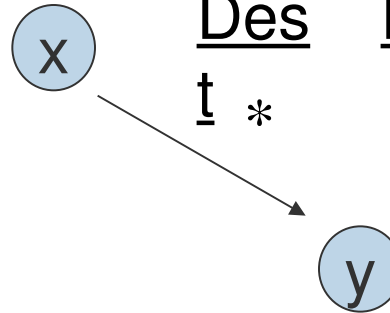
x

Des

Defer if

t *

$u \rightarrow v$



Build interferer list at v

Sourc

Interferer

e u

x

If x adversely interferes with v's reception of u's transmission, then v adds x to its interferer list as shown

Postamble + SoftPHY makes this process easy