ECE 598HH: Advanced Wireless Networks and Sensing Systems

Lecture 6: Rate Adaptation & Soft Information

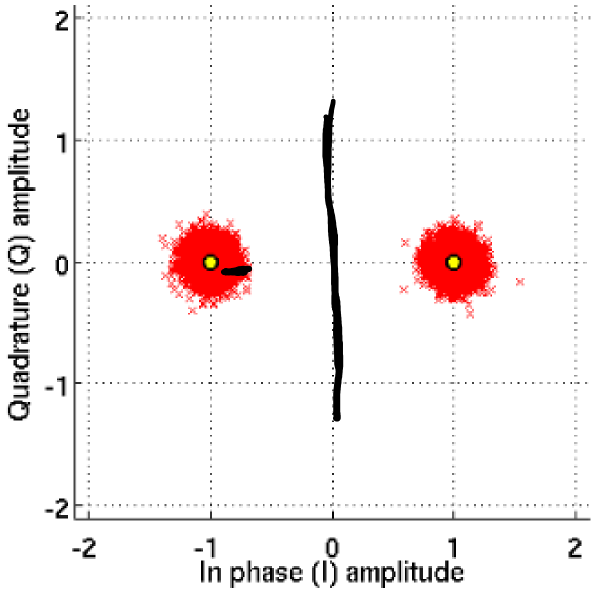
Haitham Hassanieh

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

CSL: COORDINATED SCIENCE LAB

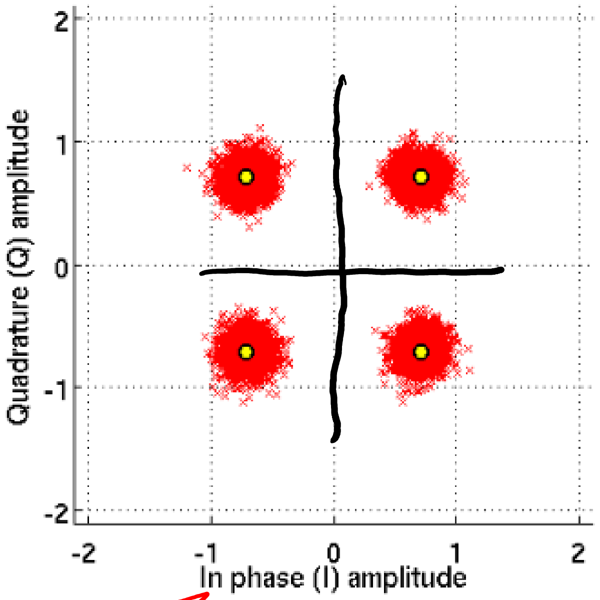
Modulation Schemes

Binary Phase-Shift Keying (BPSK)



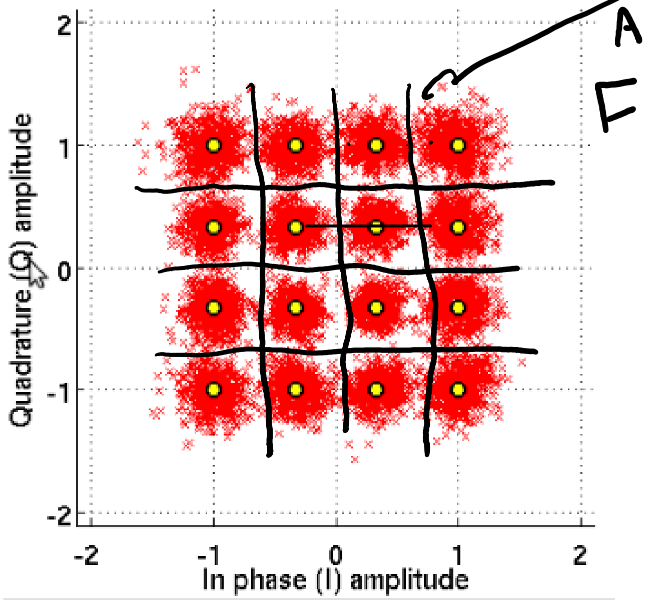
1 bit/value

Quadrature Phase-Shift Keying (QPSK)



2 bits/value

16-Quadrature Amplitude Modulation (16-QAM)



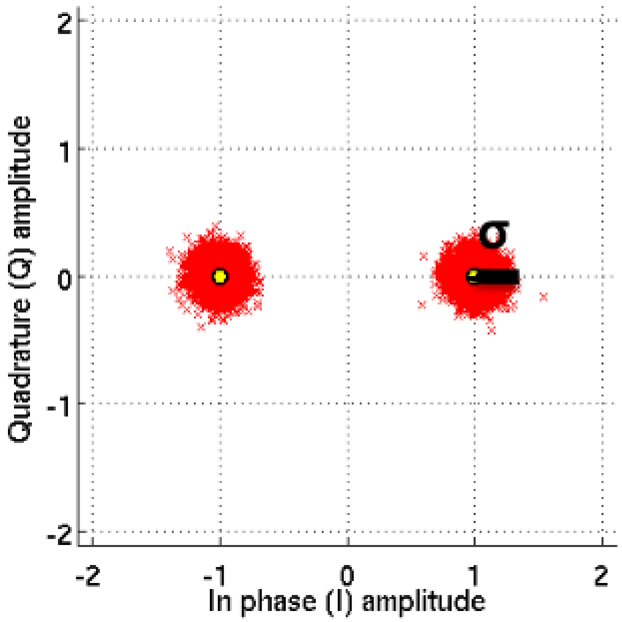
Add FEC

4 bits/valu

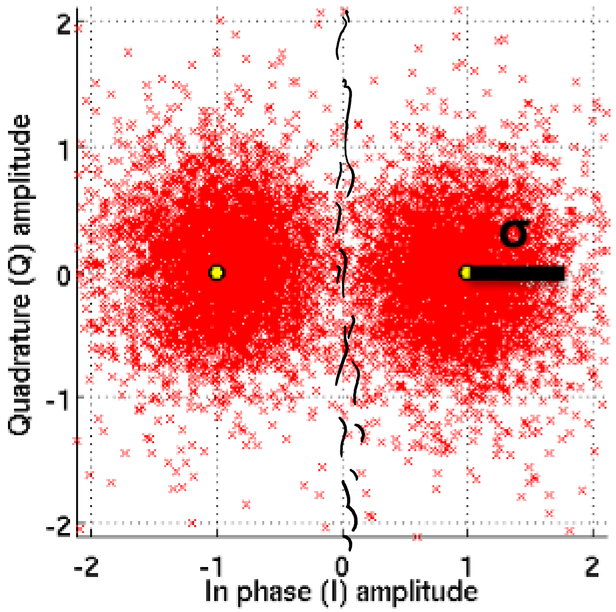
Data rate 4x higher.

Choice of Modulation

High SNR

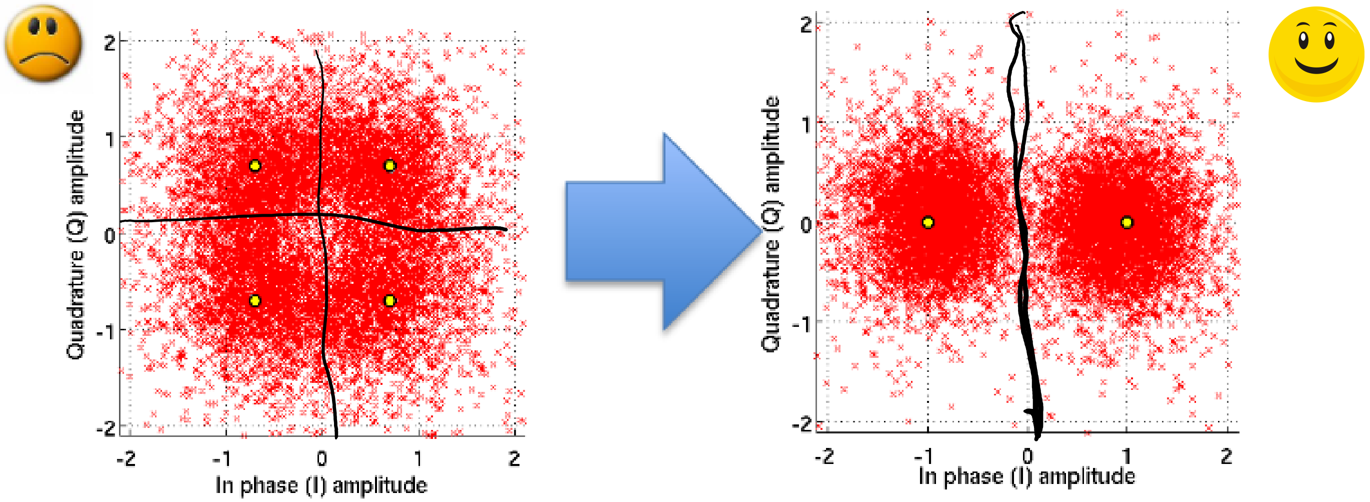


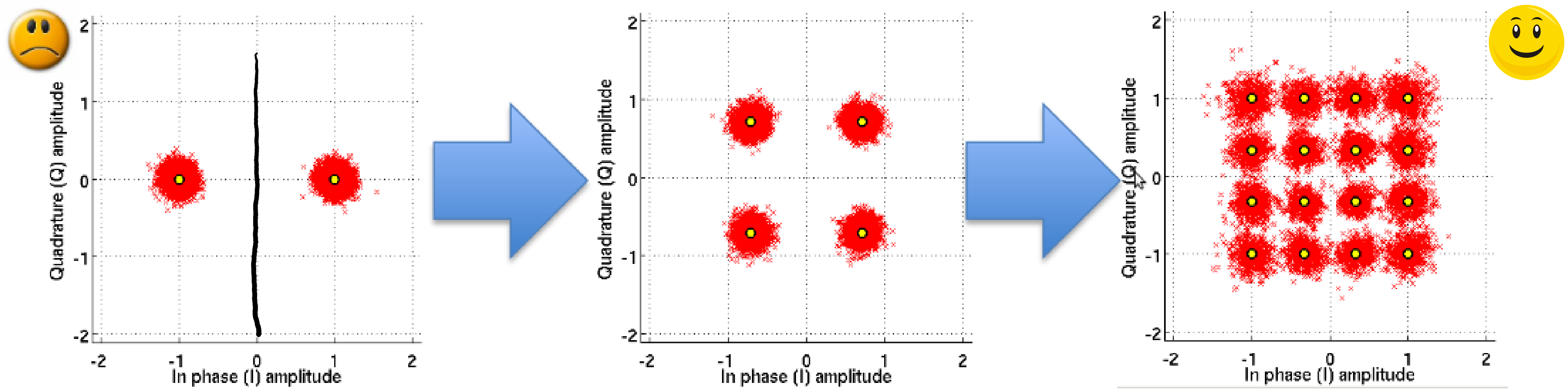
Low SNR



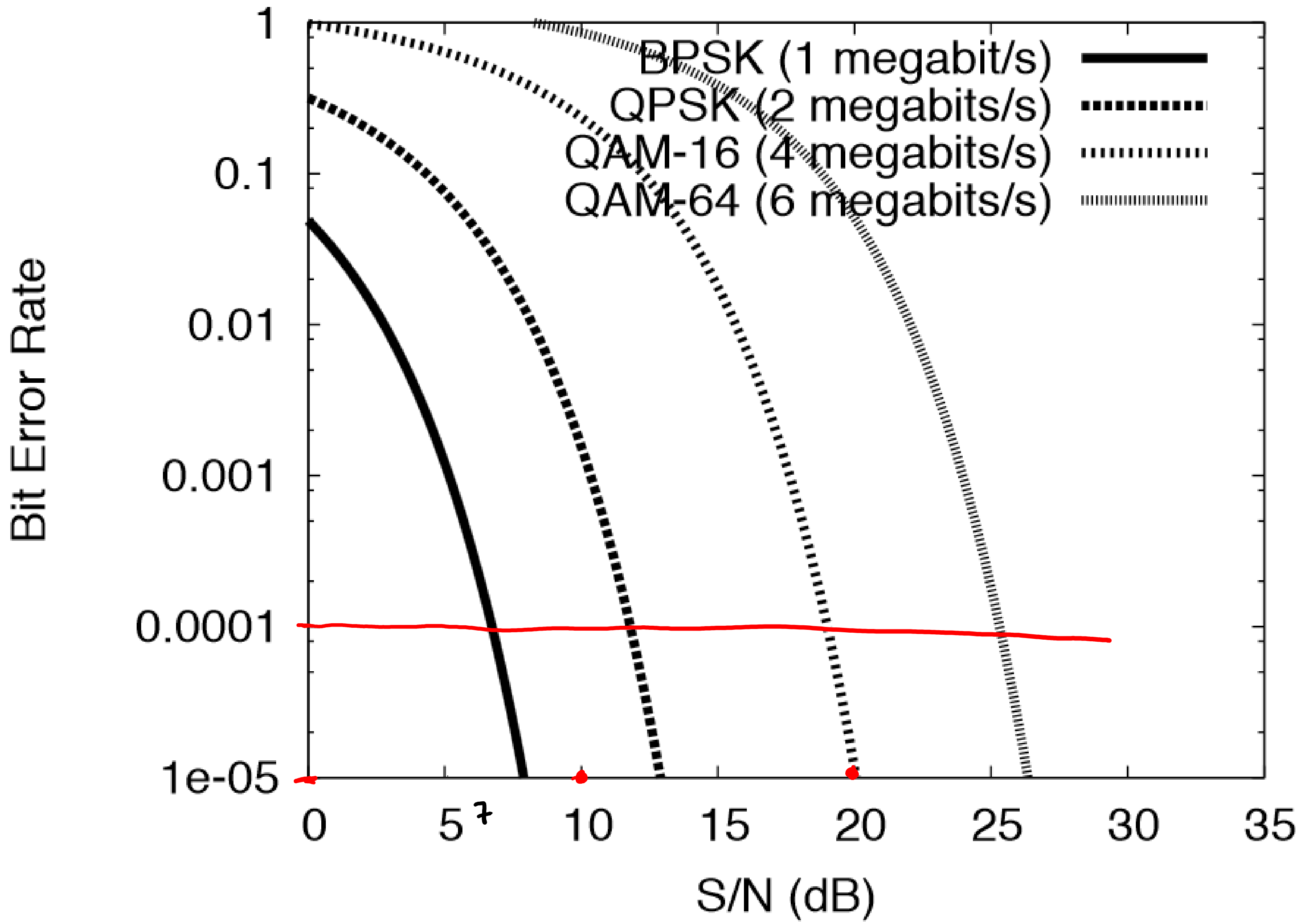
- Signal power normalized to 1
- Gaussian noise with std. dev. σ
- $SNR = 10\log_{10} 1/\sigma^2 = -20\log_{10}\sigma$

Choice of Modulation





BER vs SNR


Bit Error Rate
1
0.1
0.01
0.001
0.0001
1e-05
0
5
7
10
15
20
25
30
35
S/N (dB)
BPSK (1 megabit/s)
QPSK (2 megabits/s)
QAM-16 (4 megabits/s)
QAM-64 (6 megabits/s)

Throughput vs SNR

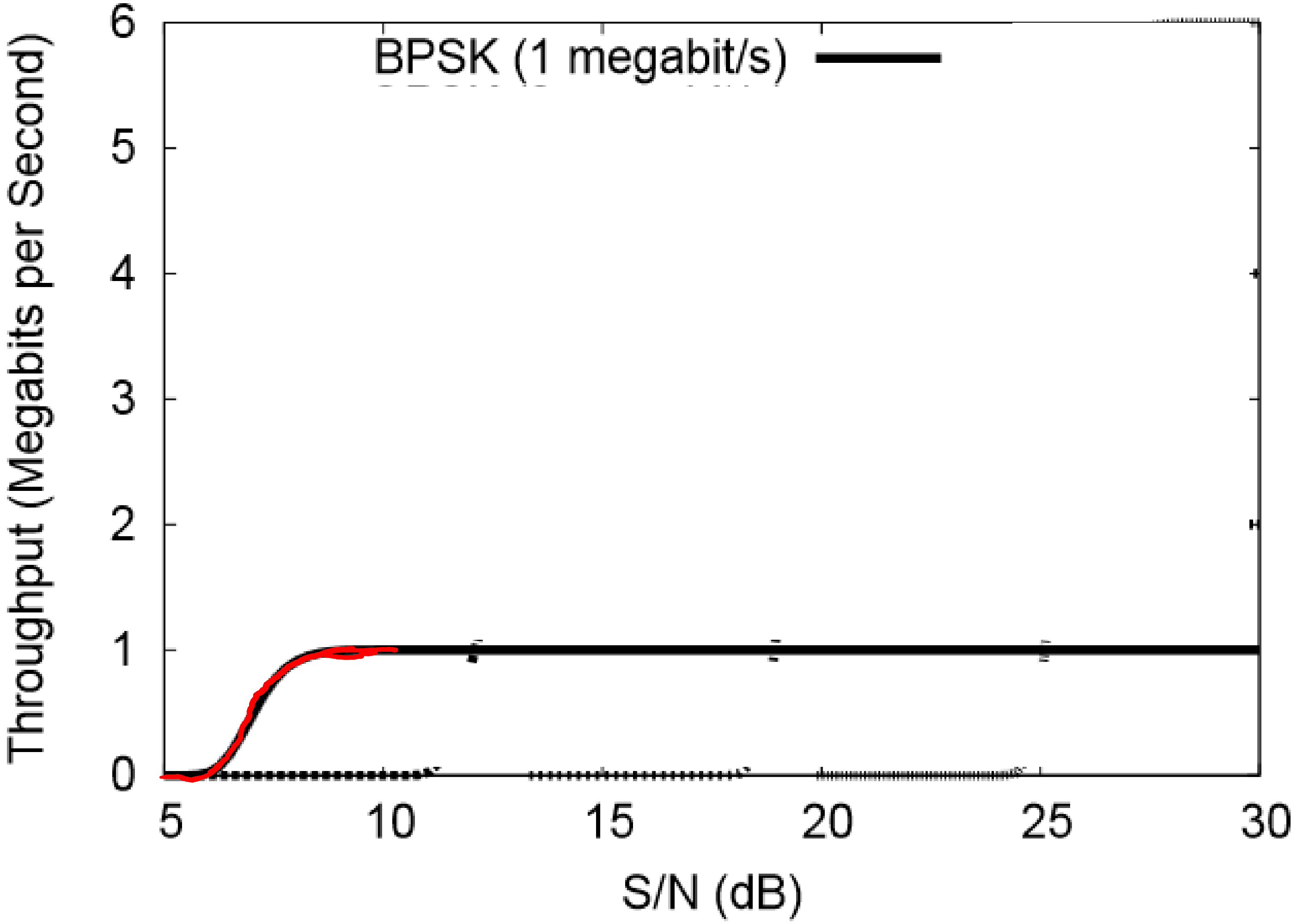
$$Data\ Rate = Bandwidth \times Bits/sample \times Code\ Rate$$

$$Capacity = Bandwidth \times \log_2(1 + SNR)$$

Throughput: number of bits correctly received per second

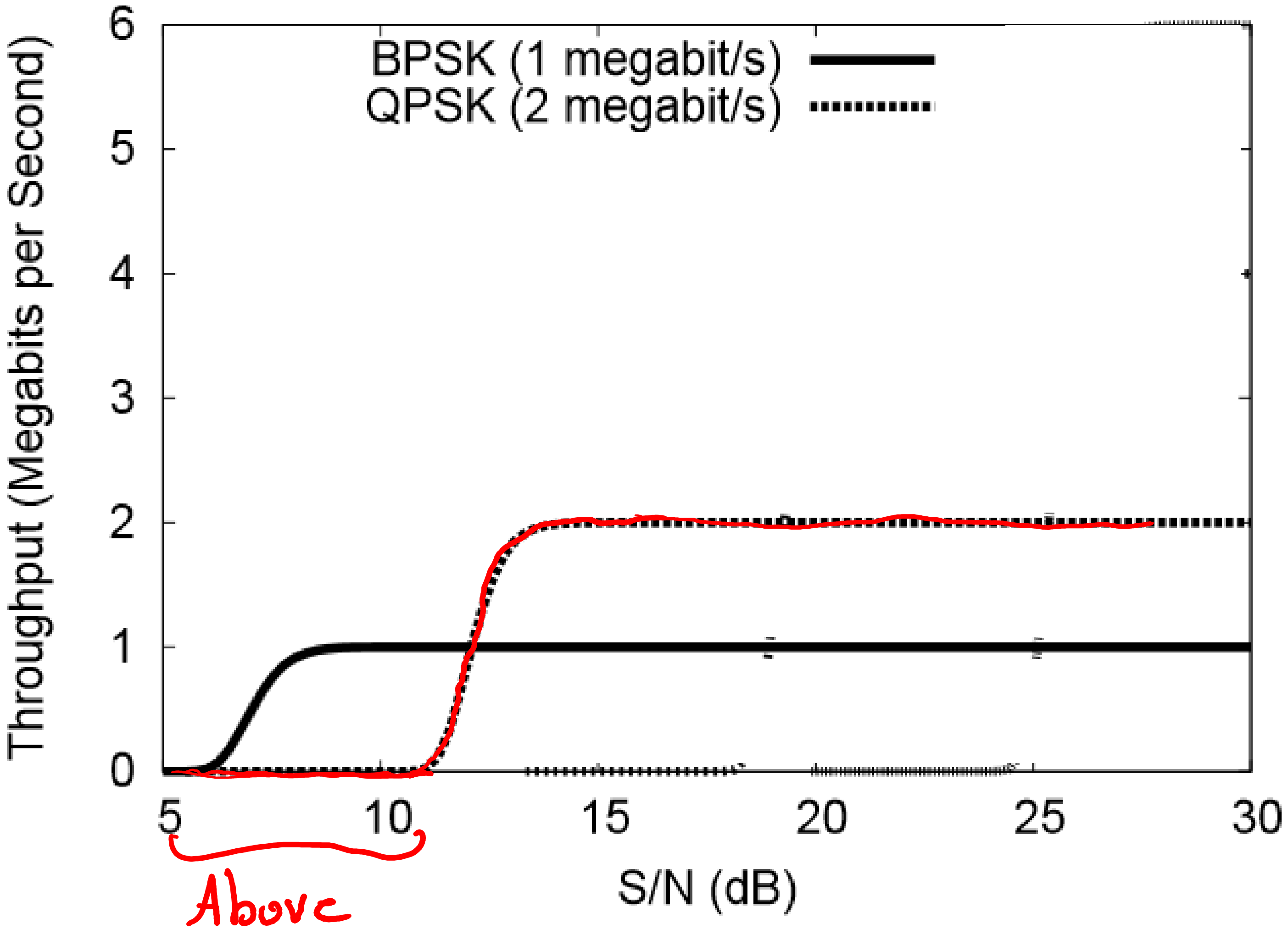
$$Throughput \leq Data\ Rate < Capacity$$

Throughput vs SNR



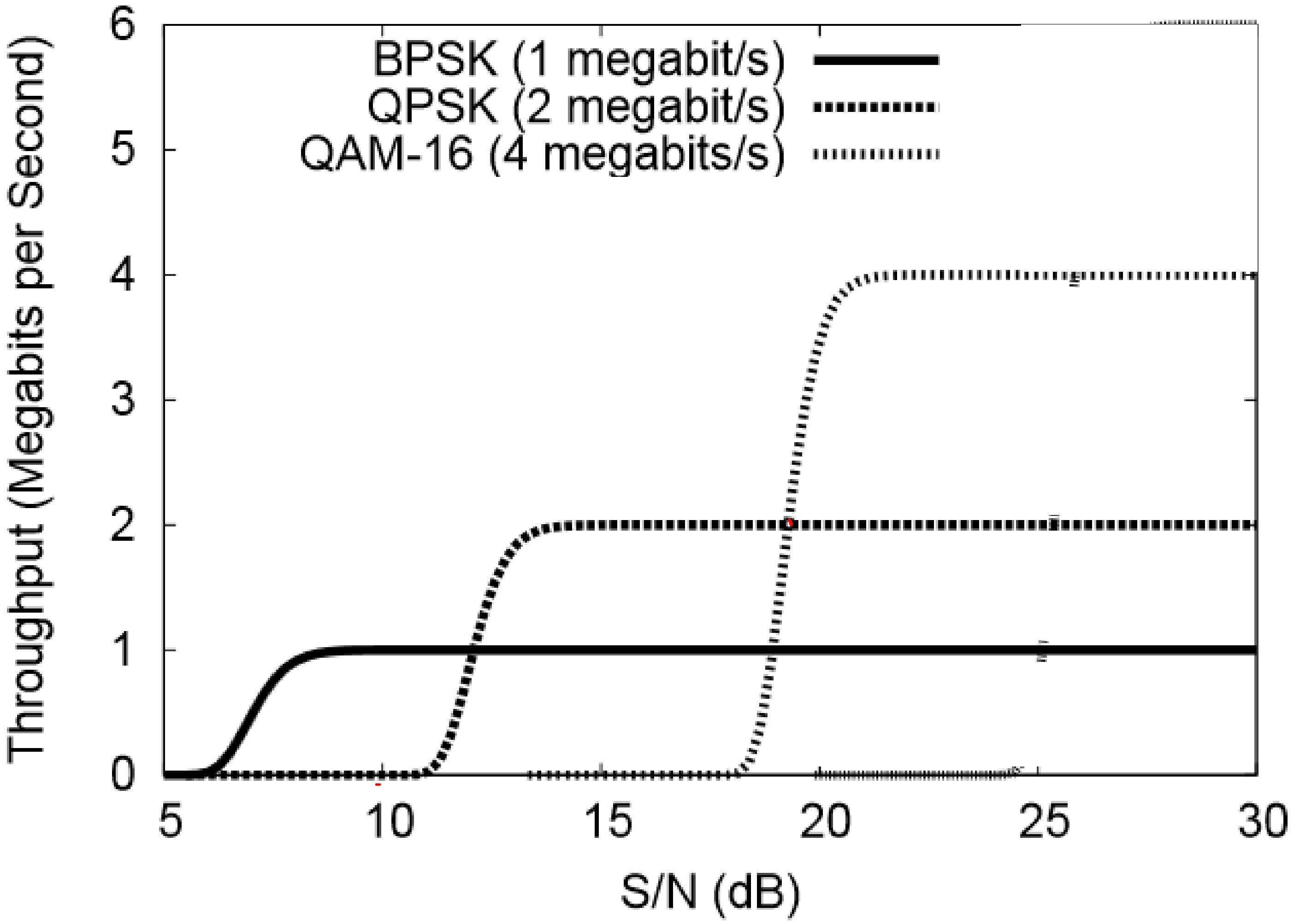
Throughput vs SNR

1 MHz channel

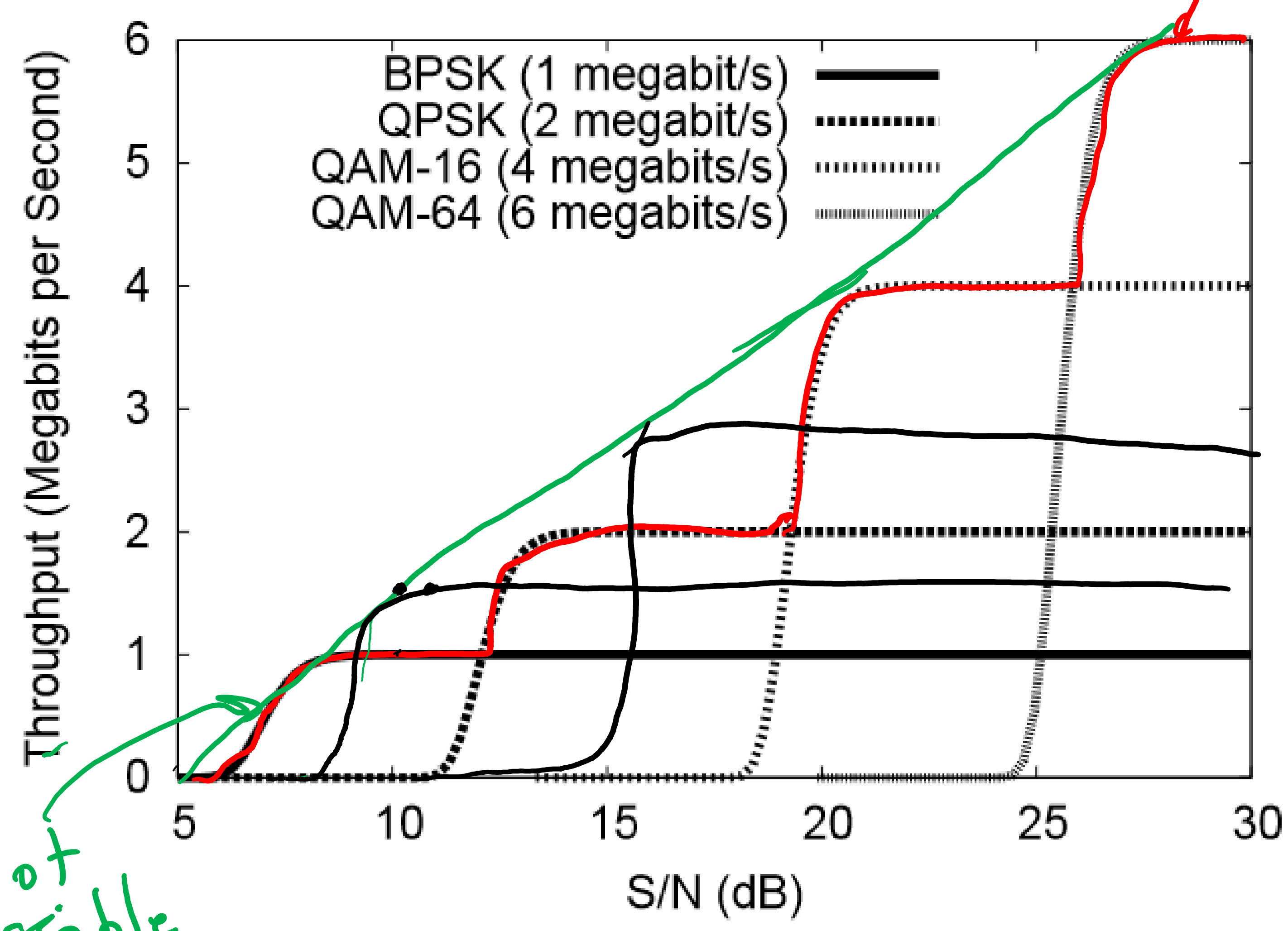


Above capacity

Throughput vs SNR



Throughput vs SNR



goal is to be on this curve

Not possible in practice?

Cannot implement everything in hardware.

Rate Adaptation

Choose the best modulation and coding scheme that maximizes the throughput that can be supported by the channel.

Challenges:

- Few modulation and coding rates supported by standards/hardware → must choose for discrete set
- TX does not know the channel and noise at the RX before choosing the modulation & coding.

How to measure channel quality in Practice?

- Loss Rate:
  - keep track of ACKs received.
  - channel can change drastically!
- Throughput:
  - Success of a bitrate used → maximizes exactly what we want.
  - Average over window? → large window ⇒ good estimate
    → small window ⇒ bad estimate
- SNR:
  - Hard to measure
  - 802.11 gives us RSSI (Not very correlated with SNR) } Effective SNR!!
- Probe Packets

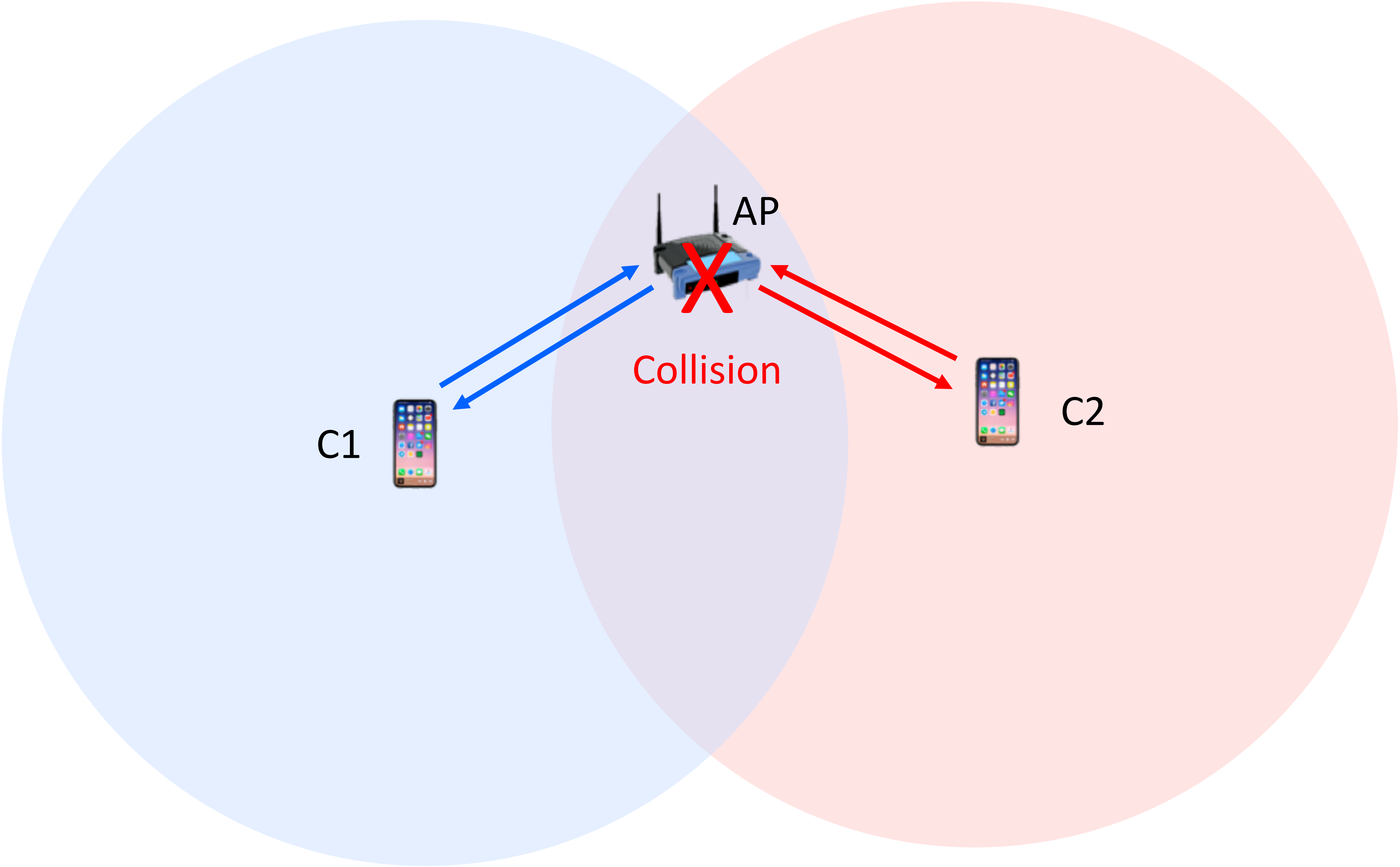
Rate adaptation is hard

- Channel changes quickly
- Any metric:
  - Good estimate → Need many samples to average
  - Small number of sample before the channel changes. ⇒ bad estimate.
- Cannot tell difference:
  - Bad Channel & Noise: Reduce bit rate
  - Interference: Do not reduce bit rate

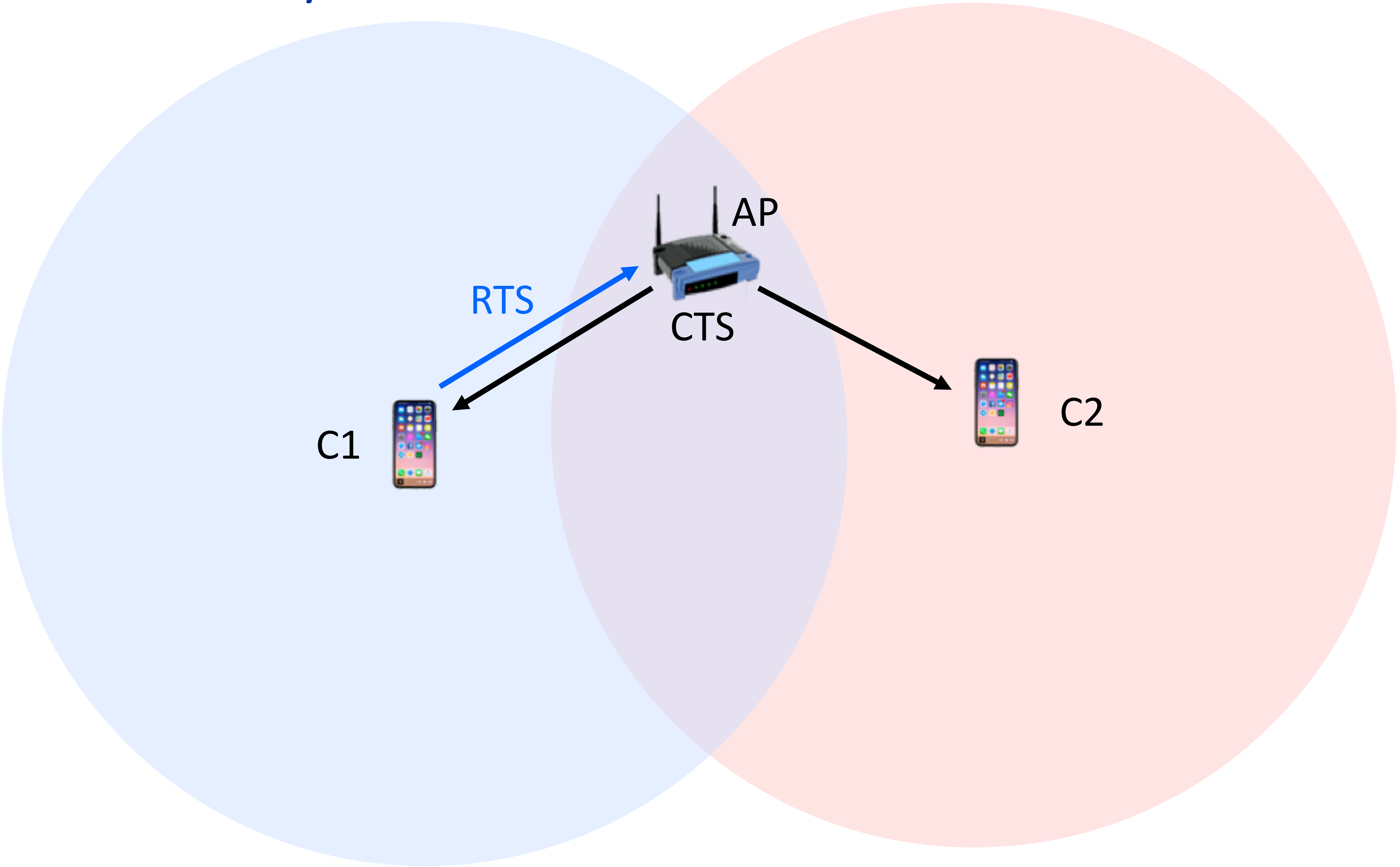
$$\text{Packet time} = \frac{\text{\# of bits/packet}}{\text{Data Rate}}$$; ↓ bitrate ⇒ ↑ packet time

⇒ ↑ collisions

Hidden Terminals



RTS/CTS and Hidden Terminals



Robust Rate Adaptation Algorithm

- Measure loss rate over 100ms window
  - Long enough to get good measurement
  - Short enough that the channel does not change.

- RTS/CTS has high overhead

- Adaptively uses RTS & CTS
  - Loss without RTS/CTS→ more RTS/CTS
  - Loss with RTS/CTS → reduce RTS/CTS usage.

PPR: Partial Packet Recovery

- Problem:
  - Fate sharing among bits
  - After FEC, even 1 bit error in packet
  - Checksum:

    If it passes → accept packet.

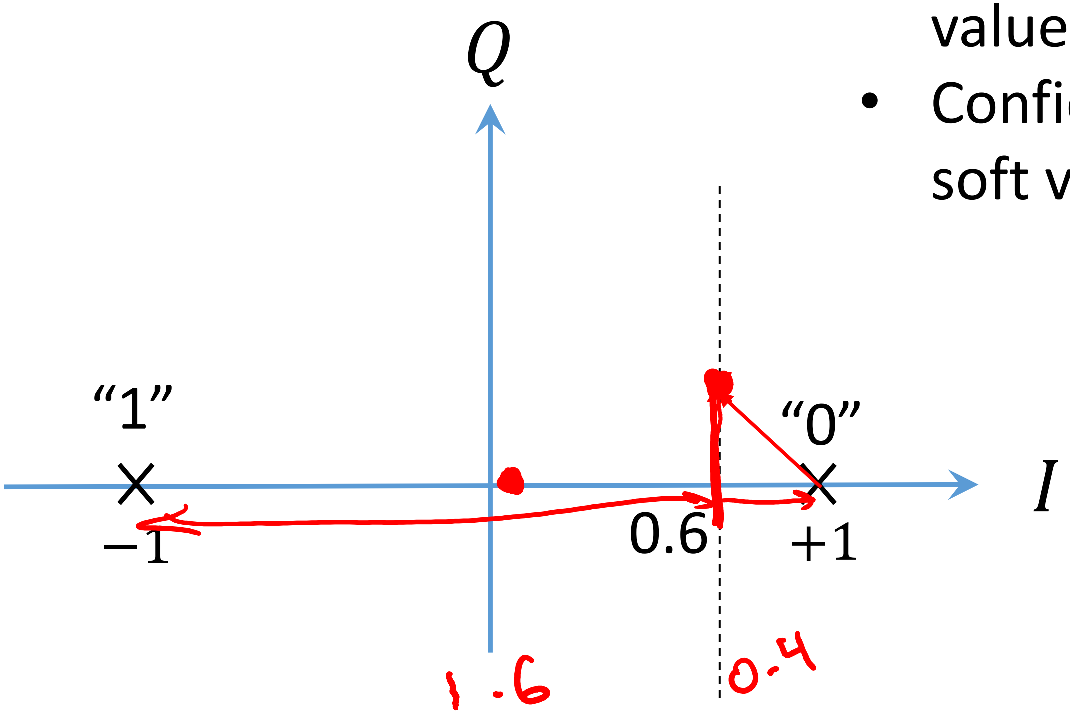
    If it fails →drop entire packet.
  - Huge waste because most bits are correct.

PPR: Partial Packet Recovery

- Solution:
  - Accept packets with errors and try to correct them.
  - Ask sender to retransmit the incorrect bits.
- How to tell which bits might have errors?
  - Soft values can be used a confidence measure
  - PHY layer can say:
    - “looks more like a 1 bit”
    - “looks more like a 0 bit”

PPR: Partial Packet Recovery

- BPSK Example



Soft Value: 0.6

- How far it is from two accepted values of 0 and 1 bits.
- Confidence metric: inverse of soft value.

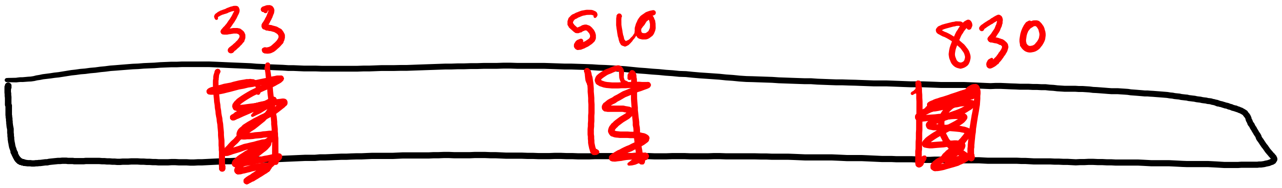
PPR: Partial Packet Recovery

- Soft Value is up to us to define
  - We are never sure the bits are in error
  - We are just hopping that our guess is reasonably correct.
- PPR uses Hamming distance:
  - Zigbee: low power, low complexity
  - Maps 4 bits to 32 bit code words (2^4 values to 2^{32} values)
  - Hamming distance: number of flipped bits between received code word and closest code word.

PPR: Partial Packet Recovery

- Retransmit bits that are in error

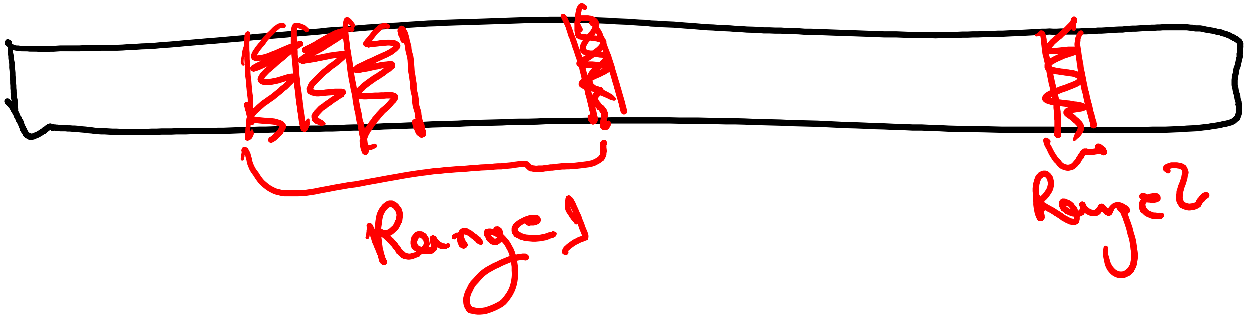
$2^{10} = 1024$ bit packet



* 10 bits for each wrong bit

* 3 bit errors ⇒ 30 bits

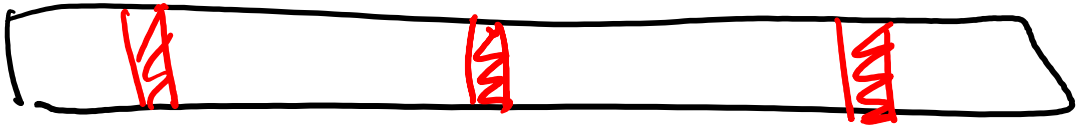
* 100 bit errors ⇒ 1000 bits X



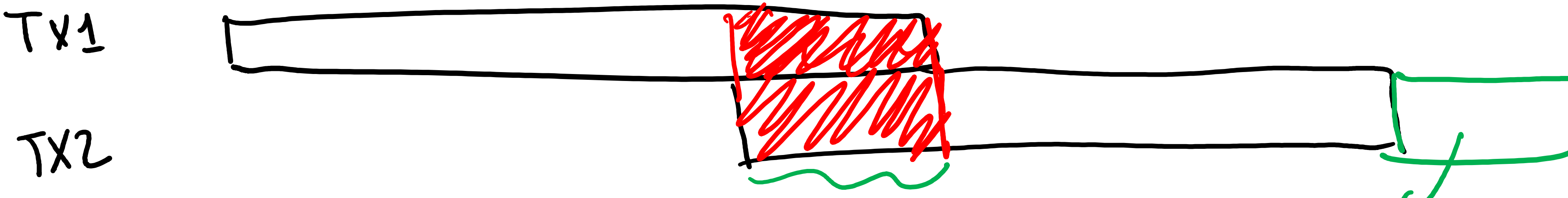
* PPR: Ask for a range of bits.

PPR: Partial Packet Recovery

- Bit errors are due to:
  - Noise : errors are dispersed : Not ideal for PPR

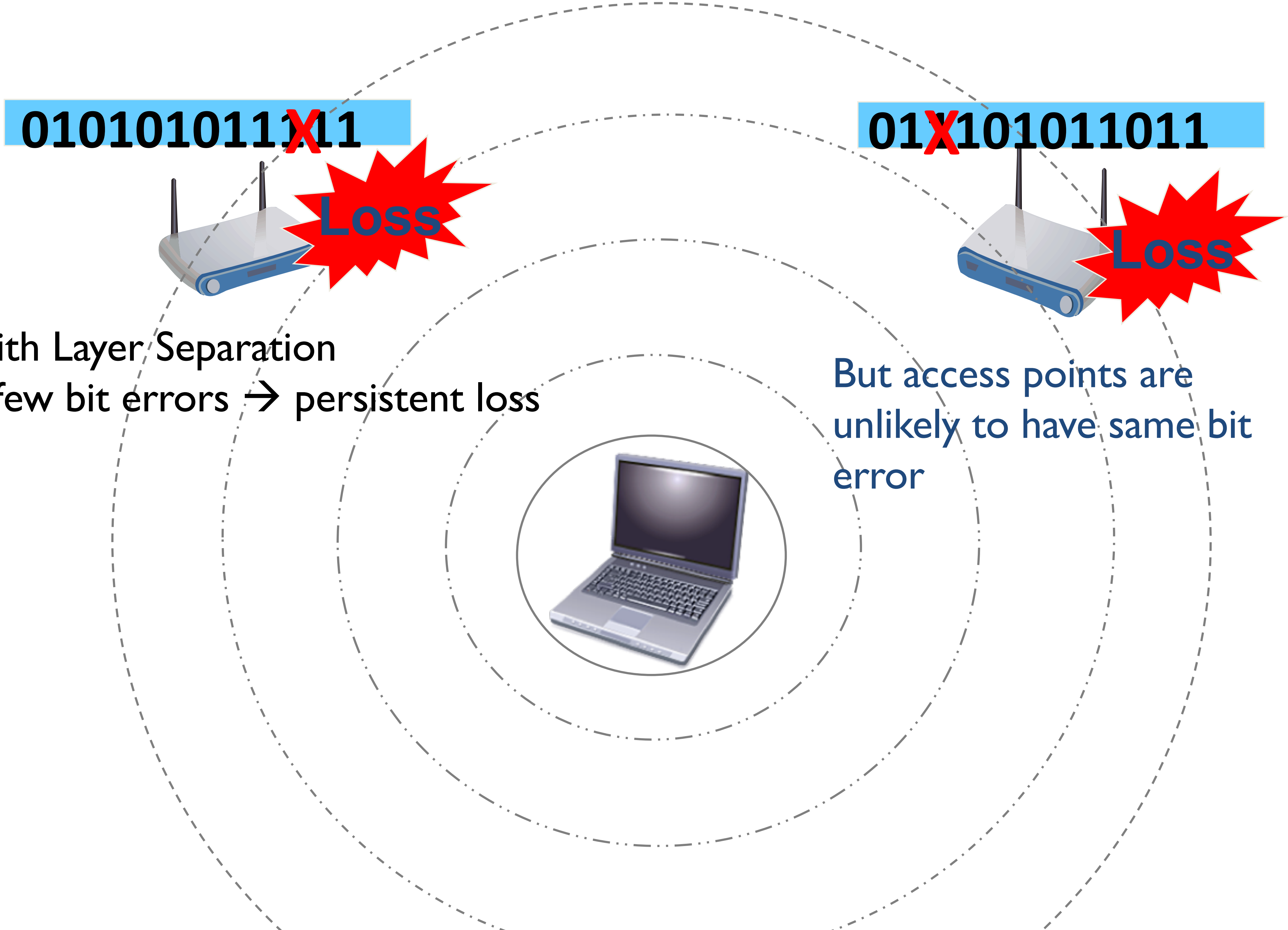


  - Collisions : errors are clustered!



Ask only for range.

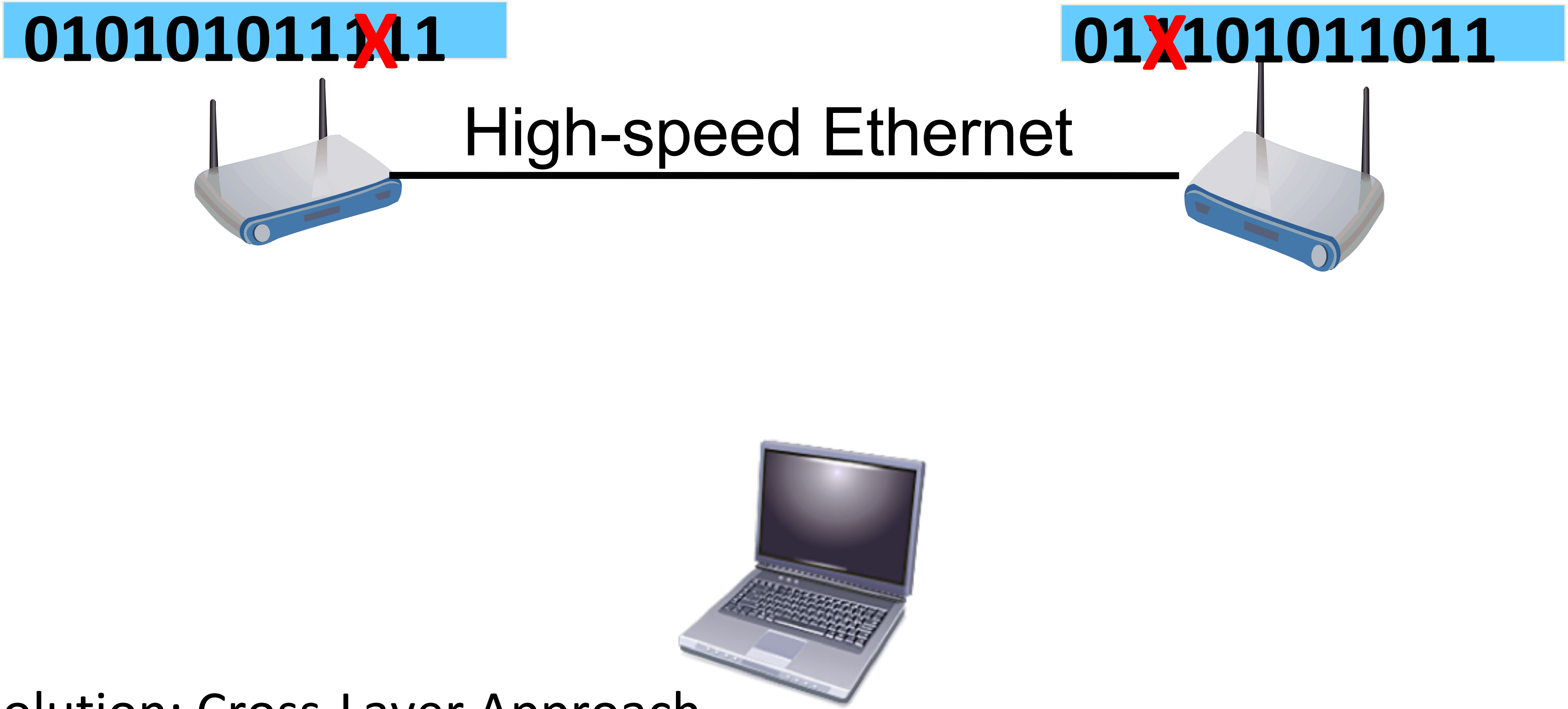
Scenario: Laptop in a Dead Spot



With Layer Separation
a few bit errors → persistent loss

But access points are unlikely to have same bit error

Scenario: Laptop in a Dead Spot



Solution: Cross-Layer Approach

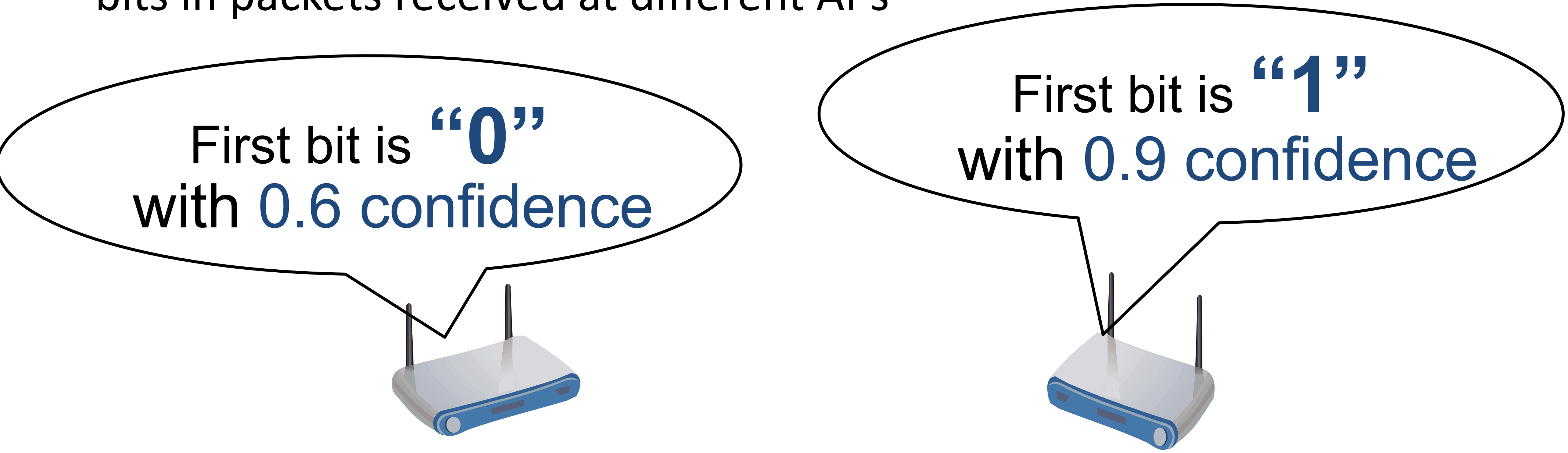
- Allow the layers to collaborate instead of acting separately
- PHY layer delivers partially correct packets
- Network layer combines correct bits across different access points to obtain correct packet

Solution: Network cooperates with physical layer

Physical Layer

010110111

- Physical layer already estimates a confidence in its 0-1 decision
- If we expose this information to the network layer, we can compare bits in packets received at different APs

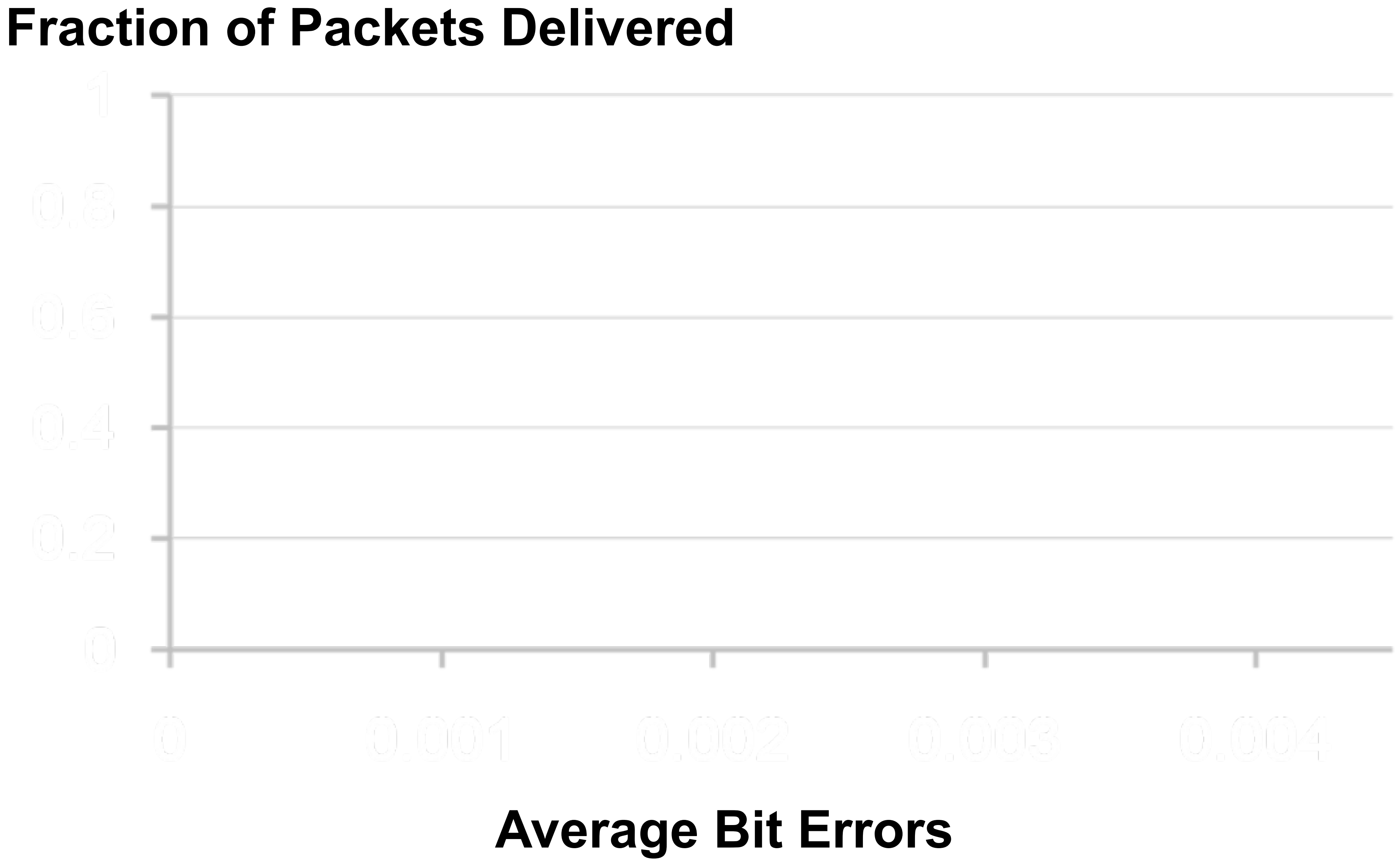


First bit is **“0”**
with 0.6 confidence

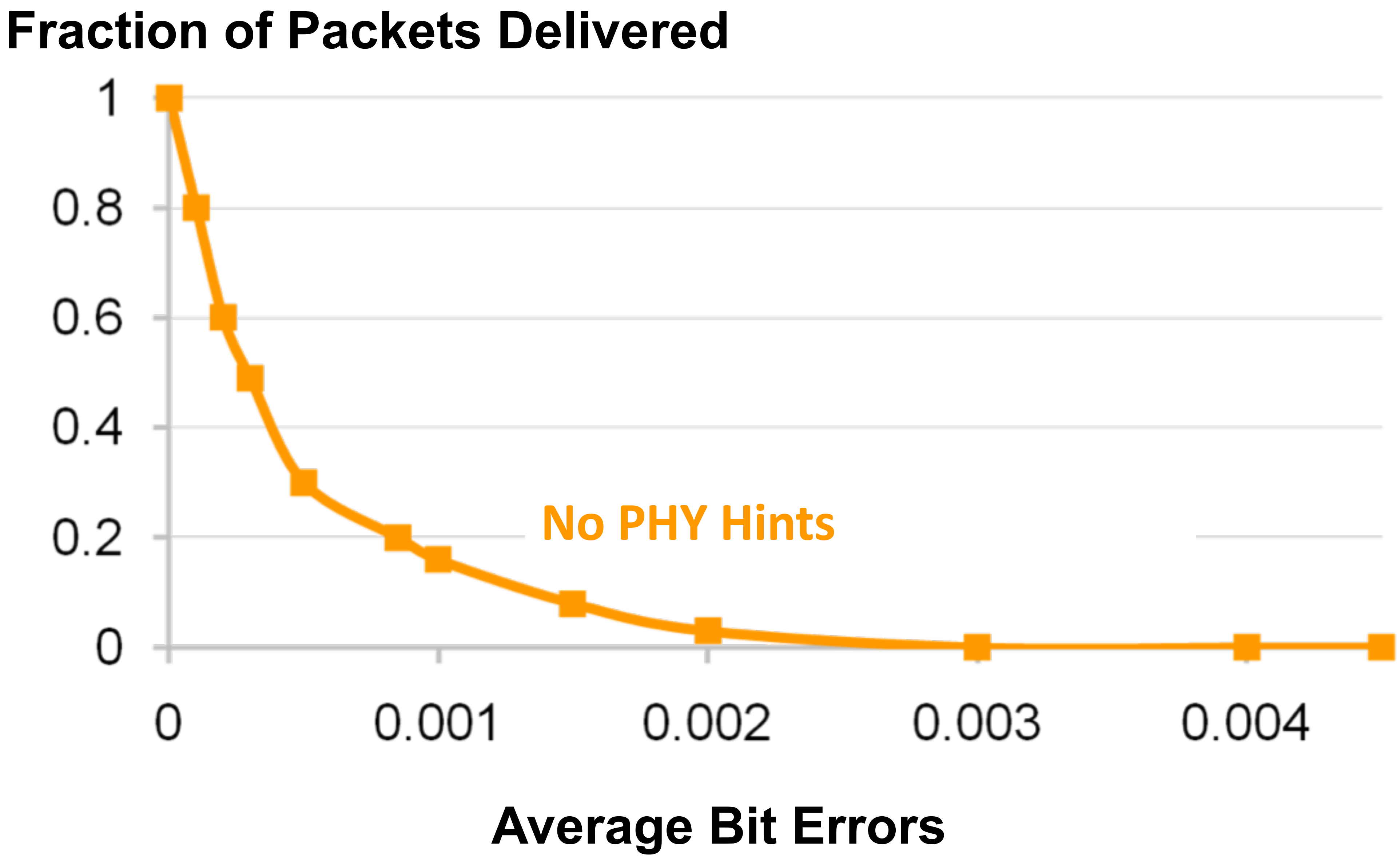
First bit is **“1”**
with 0.9 confidence

- Assign to each bit the value that corresponds to a higher confidence

Experiment: Packet Delivery vs. Poor Coverage



Experiment: Packet Delivery vs. Poor Coverage



Experiment: Packet Delivery vs. Poor Coverage