# Introduction to Computer Networks

COSC 4377

Lecture 20

Spring 2012

April 4, 2012

# Announcements

- HW9 due this week
- HW10 out
- HW11 and HW12 coming soon!
- Student presentations

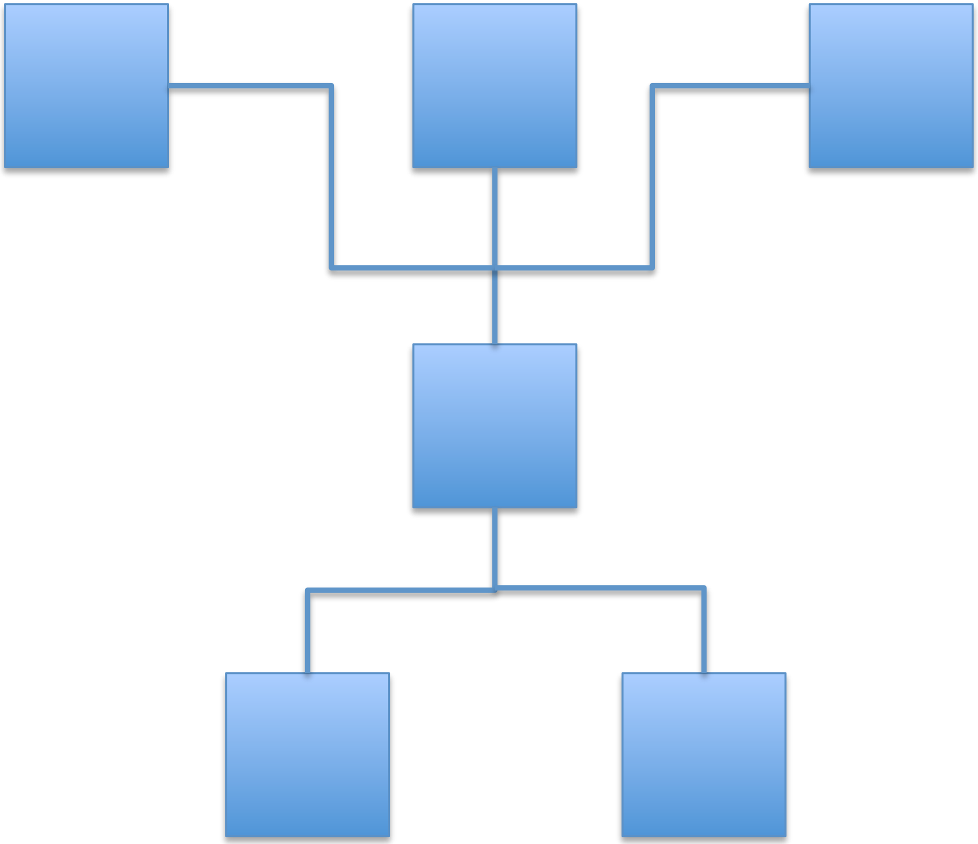
# HW9

- Capture packets using Wireshark
- Plot CDF

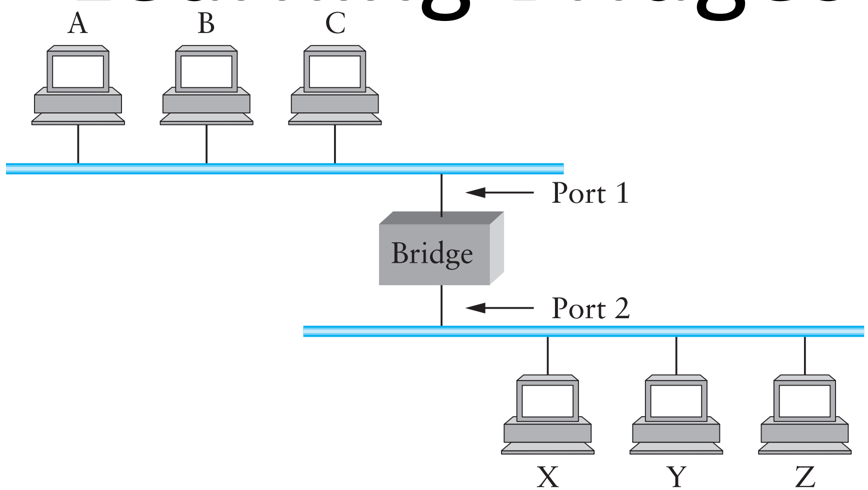
# Today's Topics

- Switching
- Physical Layer
  - Bandwidth
  - Modulation
  - Encoding
  - Framing

# Wired Media Access



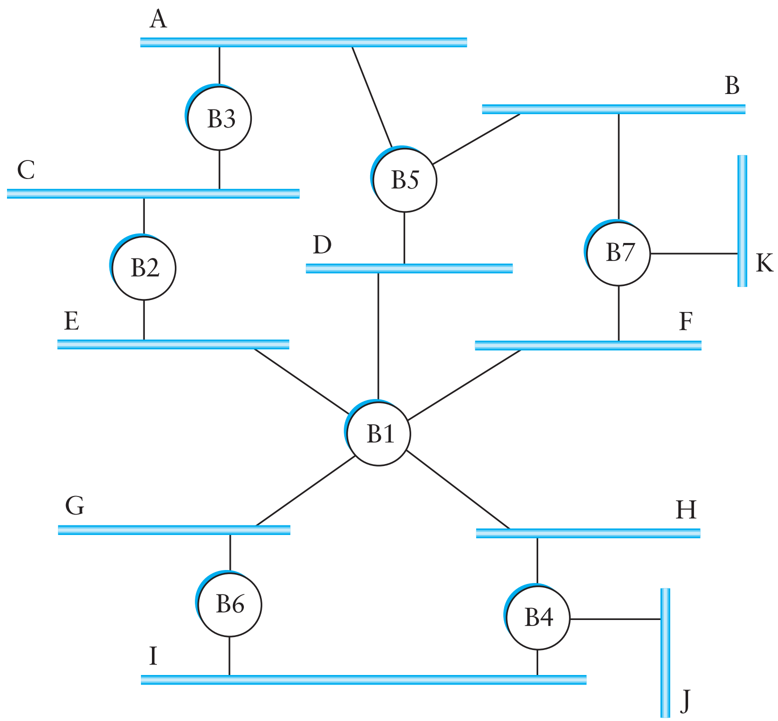
# Learning Bridges




- Idea: don't forward a packet where it isn't needed
  - If you know recipient is not on that port
- Learn hosts' locations based on source addresses
  - Build a table as you receive packets
- Table says when *not* to forward a packet
  - Doesn't need to be complete for *correctness*

# Dealing with Loops

- Problem: people may create loops in LAN!
  - Accidentally, or to provide redundancy
  - Don't want to forward packets indefinitely

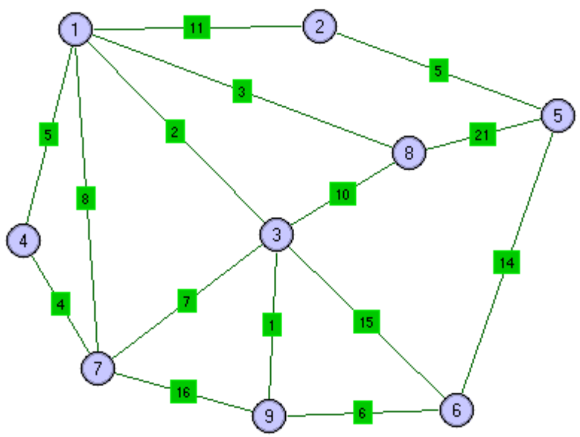


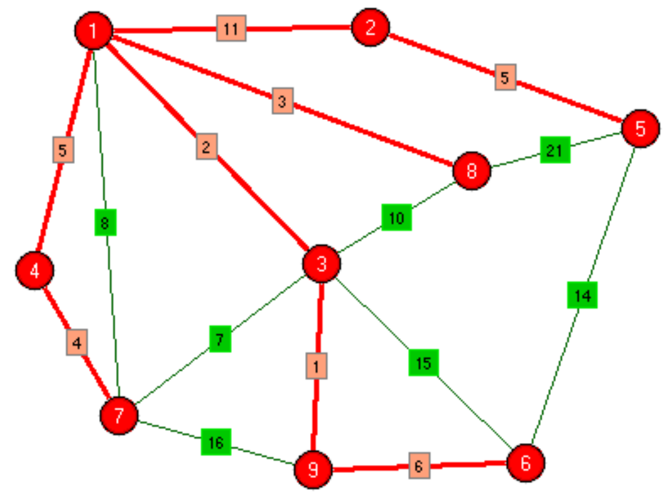
# Spanning Tree

In the mathematical field of graph theory, a spanning tree T of a connected, undirected graph G is a tree composed of all the vertices and some (or perhaps all) of the edges of G. Informally, a spanning tree of G is a selection of edges of G that form a tree spanning every vertex. That is, every vertex lies in the tree, but no cycles (or loops) are formed.

From: http://en.wikipedia.org/wiki/Spanning_tree

# Spanning Tree



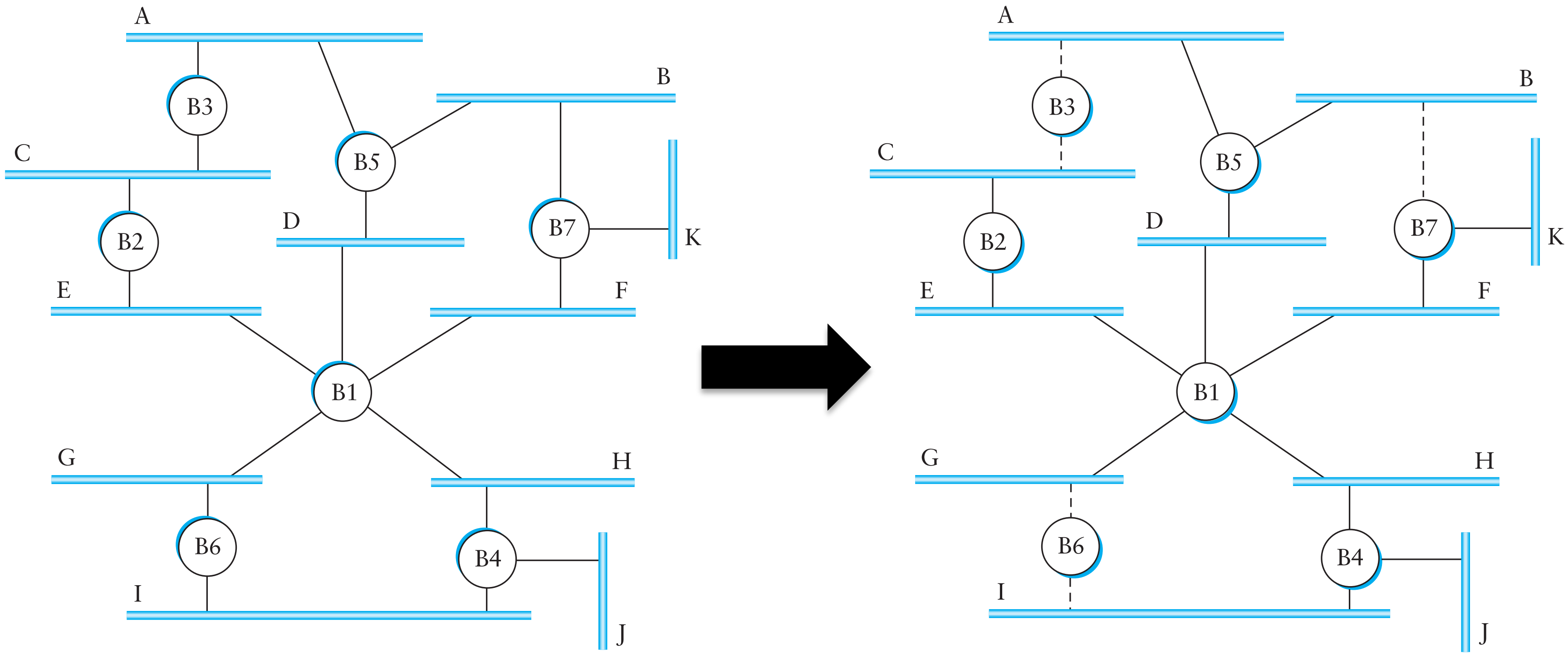




http://www.graph-magics.com/articles/min_spantree.php

# Spanning Tree Algorithms

- Graph search algorithms
- Dijkstra’s algorithm
- Minimum-spanning Tree Algorithms

# Spanning Tree




- Need to disable ports, so that no loops in network
- Like creating a spanning tree in a graph
  - View switches and networks as nodes, ports as edges

# Distributed Spanning Tree Algorithm

- Every bridge has a unique ID (Ethernet address)
- Goal:
  - Bridge with the smallest ID is the root
  - Each segment has one designated bridge, responsible for forwarding its packets towards the root
    - Bridge closest to root is designated bridge
    - If there is a tie, bridge with lowest ID wins

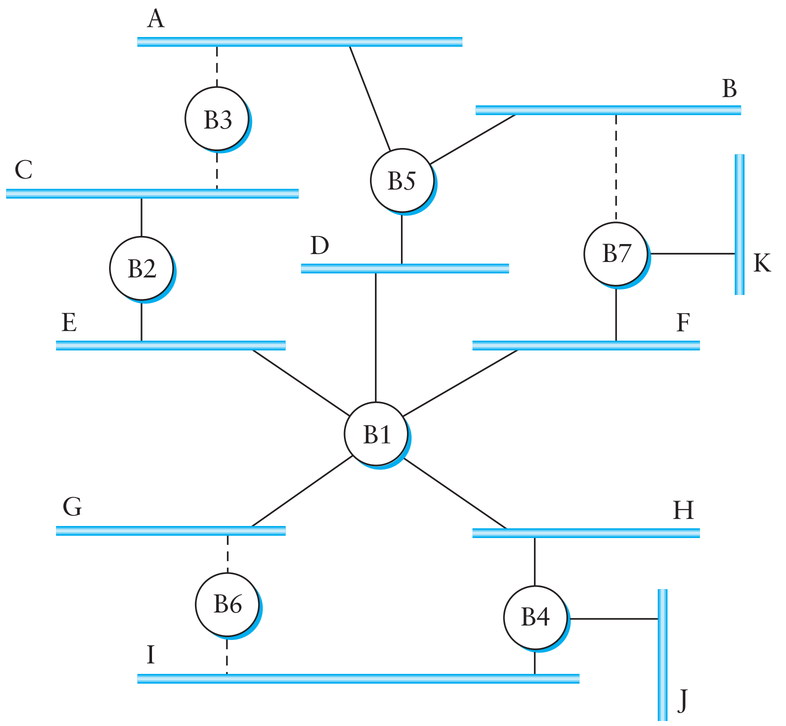
# Spanning Tree Protocol

- Spanning Tree messages contain:
  - ID of bridge sending the message
  - ID sender believes to be the root
  - Distance (in hops) from sender to root
- Bridges remember best config msg on each port
- Send message when you think you are the root
- Otherwise, forward messages from best known root
  - Add one to distance before forwarding
  - Don’t forward if you know you aren’t dedicated bridge

# Limitations of Bridges

- Scaling
  - Spanning tree algorithm doesn’t scale
  - Broadcast does not scale
  - No way to route around congested links, *even if path exists*
- May violate assumptions
  - Could confuse some applications that assume single segment
  - Much more likely to drop packets
  - Makes latency between nodes non-uniform
  - Beware of transparency

# Local Area Network



# Physical Layer

- Responsible for specifying the physical medium
  - Type of cable, fiber, wireless frequency
- Responsible for specifying the signal (modulation)
  - Transmitter varies *something (*amplitude, frequency, phase)
  - Receiver samples, recovers signal
- Responsible for specifying the bits (encoding)
  - Bits above physical layer -> *chips*

# Specifying the signal

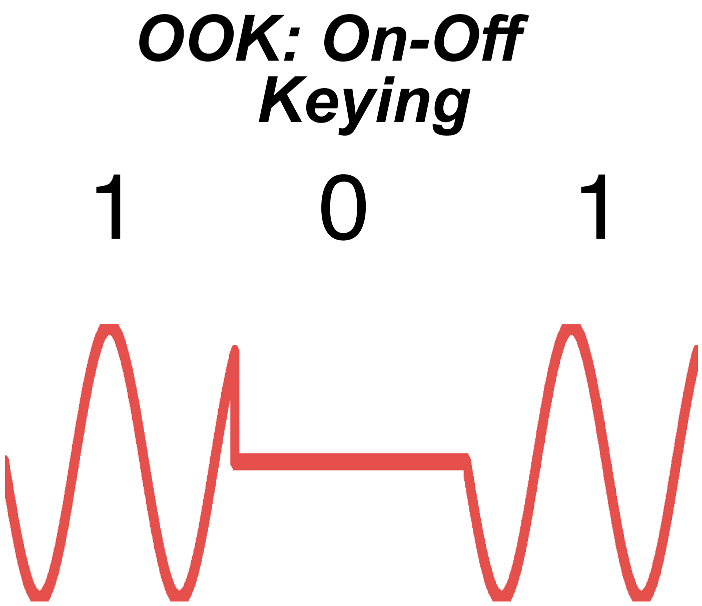
- Chips vs bits
  - Chips: data (in bits) at the physical layer
  - Bits: data above the physical layer
- Phy layer specifies Analog signal ⟷ chip mapping
  - On-off keying (OOK): voltage of 0 is 0, +V is 1
  - PAM-5: 000 is 0, 001 is +1, 010 is -1, 011 is -2, 100 is +2
  - Frequency shift keying (FSK)
  - Phase shift keying (PSK)

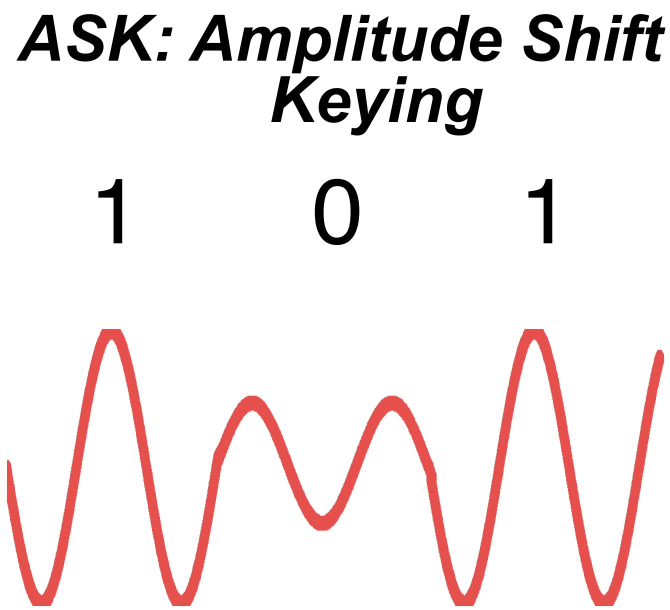
# Modulation

- Specifies mapping between digital signal and some variation in analog signal
- Why not just a square wave (1v=1; 0v=0)?
  - Not square when bandwidth limited
- Bandwidth – frequencies that a channel propagates well
  - Signals consist of many frequency components
  - Attenuation and delay frequency-dependent

# Use Carriers

- Idea: only use frequencies that transmit well
- *Modulate* the signal to encode bits

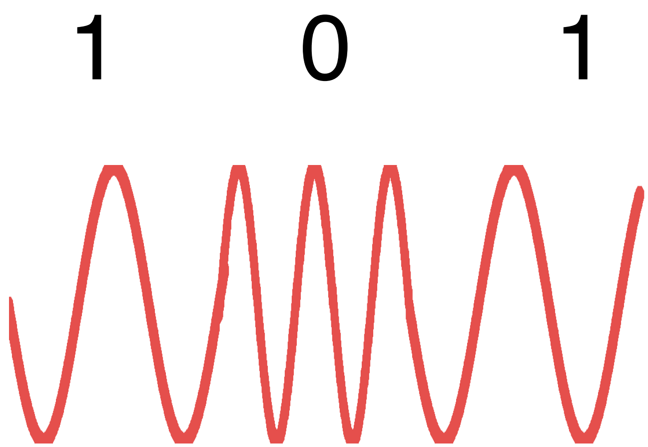


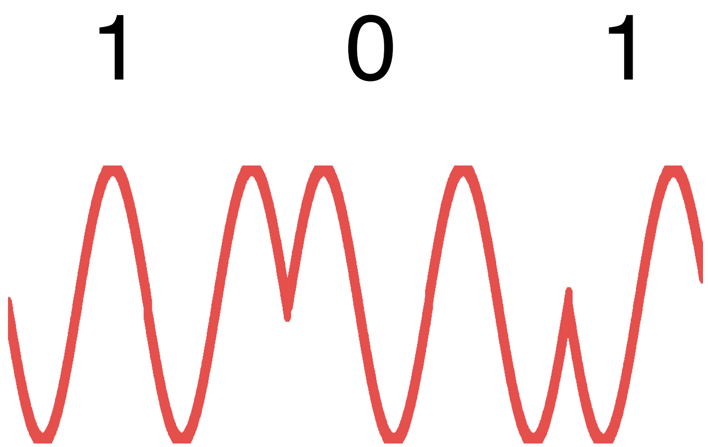
# Use Carriers

- Idea: only use frequencies that transmit well
- *Modulate* the signal to encode bits

***FSK: Frequency Shift Keying***




***PSK: Phase Shift Keying***

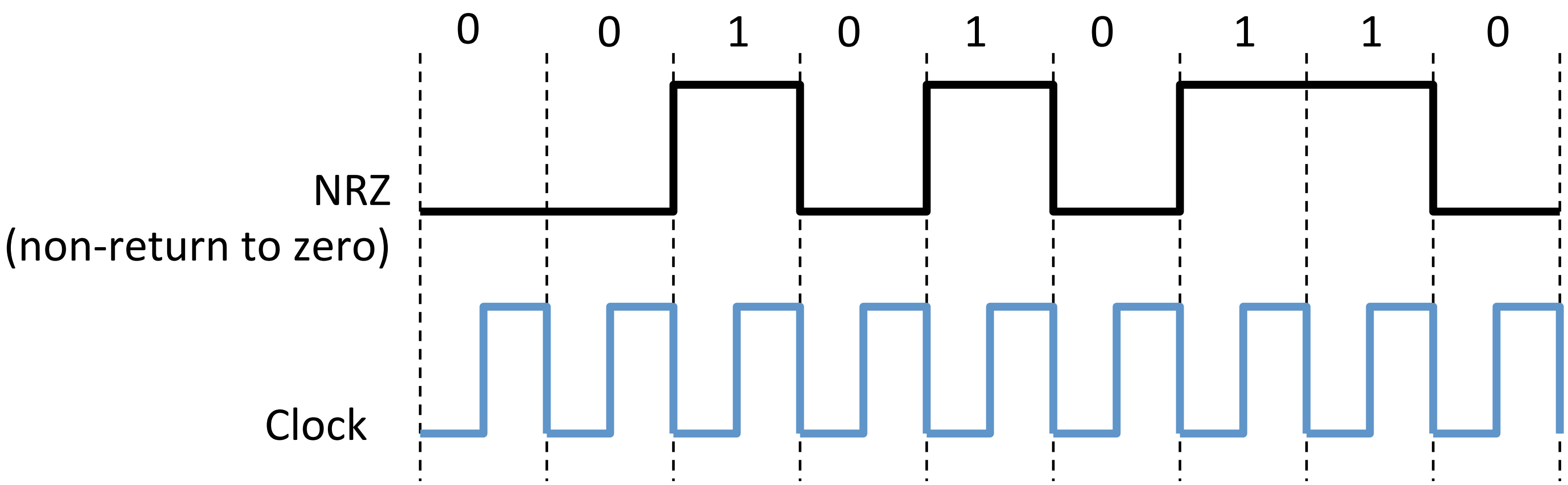


# How Fast Can You Really Send?

- Depends on frequency and signal/noise ratio
- Shannon: $C = B \log_2(1 + S/N)$
  - C is the channel capacity in bits/second
  - B is the bandwidth of the channel in Hz
  - S and N are average signal and noise power
- Example: Telephone Line
  - 3KHz b/w, 30dB S/N = 10ˆ(30/10) = 1000
  - C ≈ 30 Kbps

# Encoding

- Now assume that we can somehow modulate a signal: receiver can decode our binary stream
- How do we encode binary data onto signals?
- One approach: Non-return to Zero (NRZ)
  - Transmit 0 as low, 1 as high!

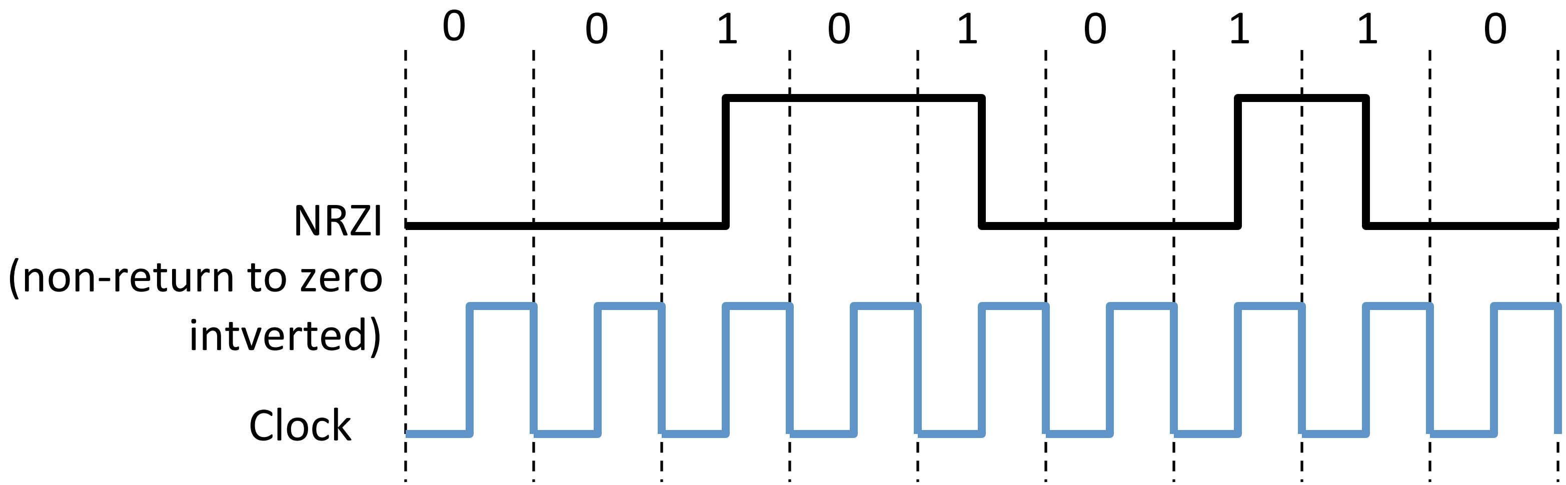


# Drawbacks of NRZ

- No signal could be interpreted as 0 (or vice-versa)
- Consecutive 1s or 0s are problematic
- Baseline wander problem
  - How do you set the threshold?
  - Could compare to average, but average may drift
- Clock recovery problem
  - For long runs of no change, could miscount periods

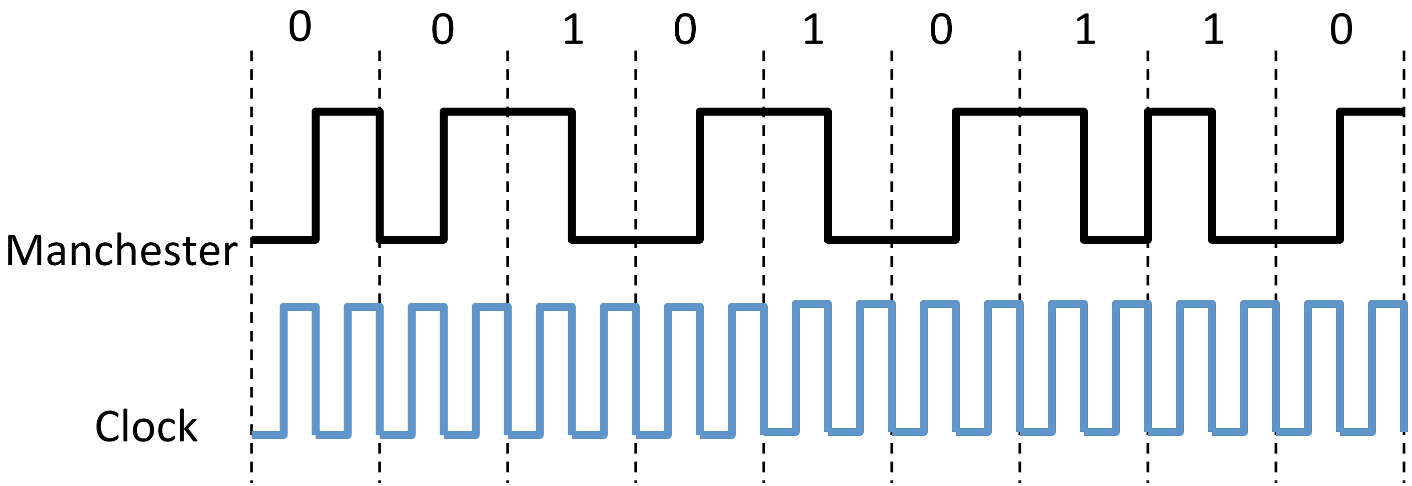
# Alternative Encodings

- Non-return to Zero Inverted (NRZI)
  - Encode 1 with transition from current signal
  - Encode 0 by staying at the same level
  - At least solve problem of consecutive 1s



# Manchester

- Map 0 → chips 01; 1 → chips 10
  - Transmission rate now 1 bit per two clock cycles
- Solves clock recovery, baseline wander
- But cuts transmission rate in half



# 4B/5B

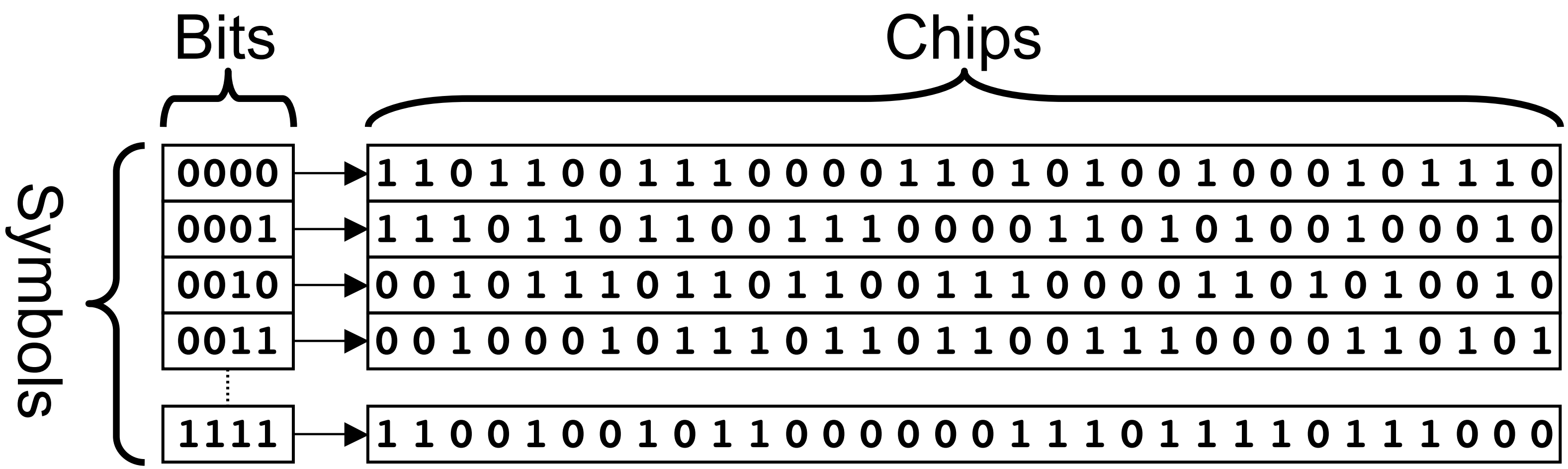
- Can we have a more efficient encoding?
- Every 4 bits encoded as 5 *chips*
- Need 16 5-bit codes:
  - selected to have no more than one leading 0 and no more than two trailing 0s
  - Never get more than 3 consecutive 0s
- Transmit chips using NRZI
- Other codes used for other purposes
  - E.g., 11111: line idle; 00100: halt
- Achieves 80% efficiency

# Encoding Goals

- DC Balancing (same number of 0 and 1 chips)
- Clock synchronization
- Can recover some chip errors
- Constrain analog signal patterns to make signal more robust
- Want near channel capacity with negligible errors
  - Shannon says it's possible, doesn't tell us how
  - Codes can get computationally expensive
- In practice
  - More complex encoding: fewer bps, more robust
  - Less complex encoding: more bps, less robust

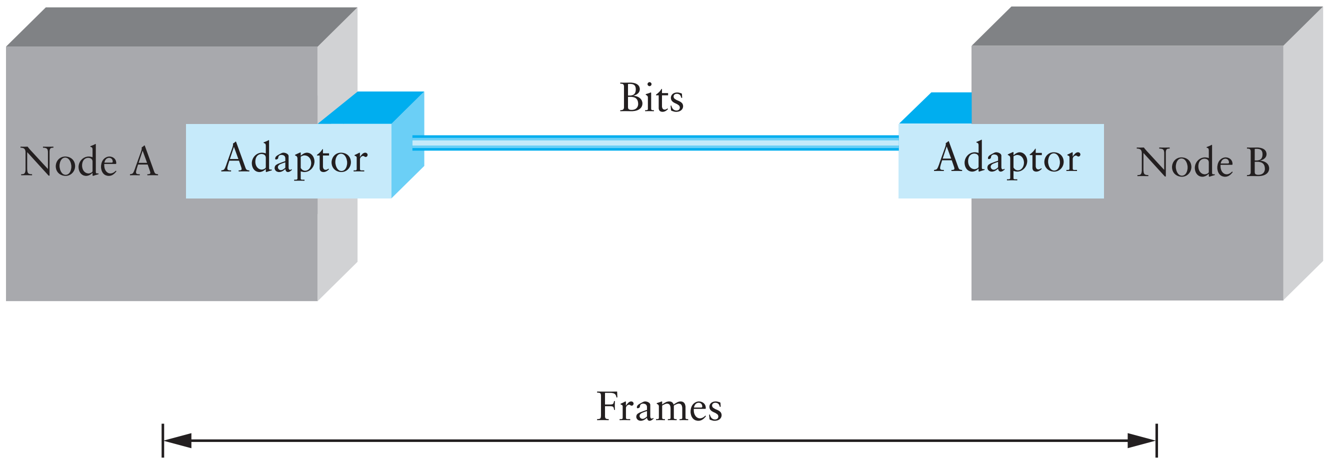
# 802.15.4

- Standard for low-power, low-rate wireless PANs
  - Must tolerate high chip error rates
- Uses a 4B/32B bit-to-chip encoding



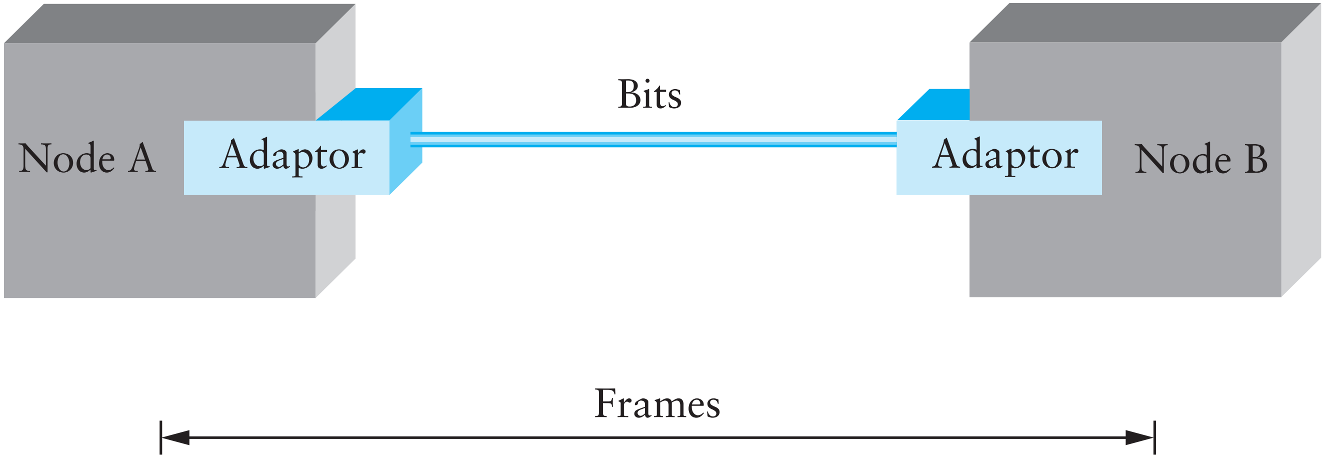
# Framing

- Given a stream of bits, how can we represent boundaries?
- Break sequence of bits into a frame
- Typically done by network adaptor



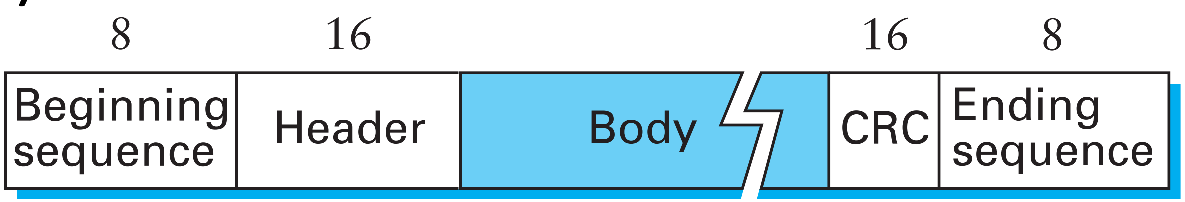
# Representing Boundaries

- Sentinels
- Length counts
- Clock-based



# Bit-Oriented Protocols

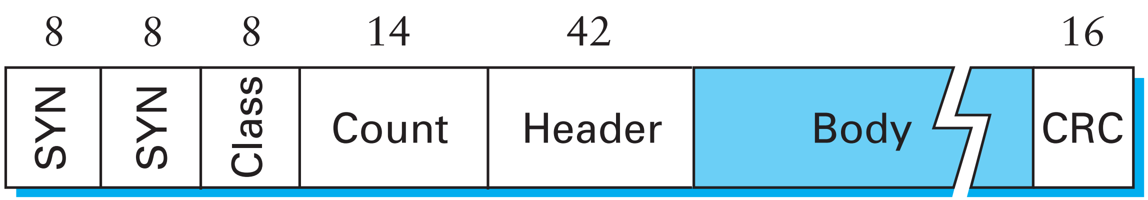
- View message as a stream of bits, not bytes
- Can use sentinel approach as well (*e.g.*, HDLC)




  - HDLC begin/end sequence 01111110
- Use *bit stuffing* to escape 01111110
  - Always append 0 after five consecutive 1s in data
  - After five 1s, receiver uses next two bits to decide if stuffed, end of frame, or error.

# Length-based Framing

- Drawback of sentinel techniques
  - Length of frame depends on data
- Alternative: put length in header (*e.g.*, DDCMP)




- Danger: Framing Errors
  - What if high bit of counter gets corrupted?
  - Adds 8K to length of frame, may lose many frames
  - CRC checksum helps detect error

# Representing Boundaries

- Sentinels
- Length counts
- Clock-based