



CSE331: Introduction to Networks and Security

Lecture 3

Fall 2004



Announcements

- HW1 is due Wednesday, Sept. 15th
 - Bring hard copy to class.

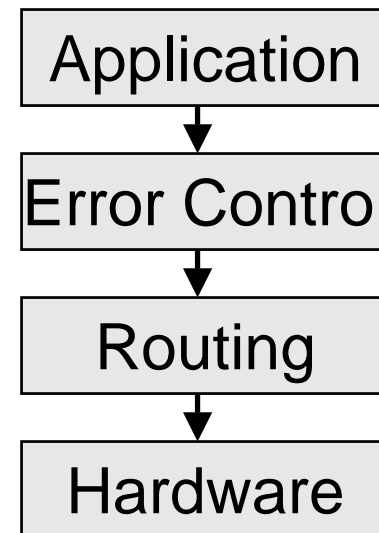


Network Architecture

- General blueprints that guide the design and implementation of networks
- Goal: to deal with the complex requirements of a network
- Use *abstraction* to separate concerns
 - Identify the useful service
 - Specify the interface
 - Hide the implementation

Layering

- A result of abstraction in network design
 - A stack of services (layers)
 - Hardware service at the bottom layer
 - Higher level services are implemented by using services at lower levels
- Advantages
 - Decompose problems
 - Modular changes

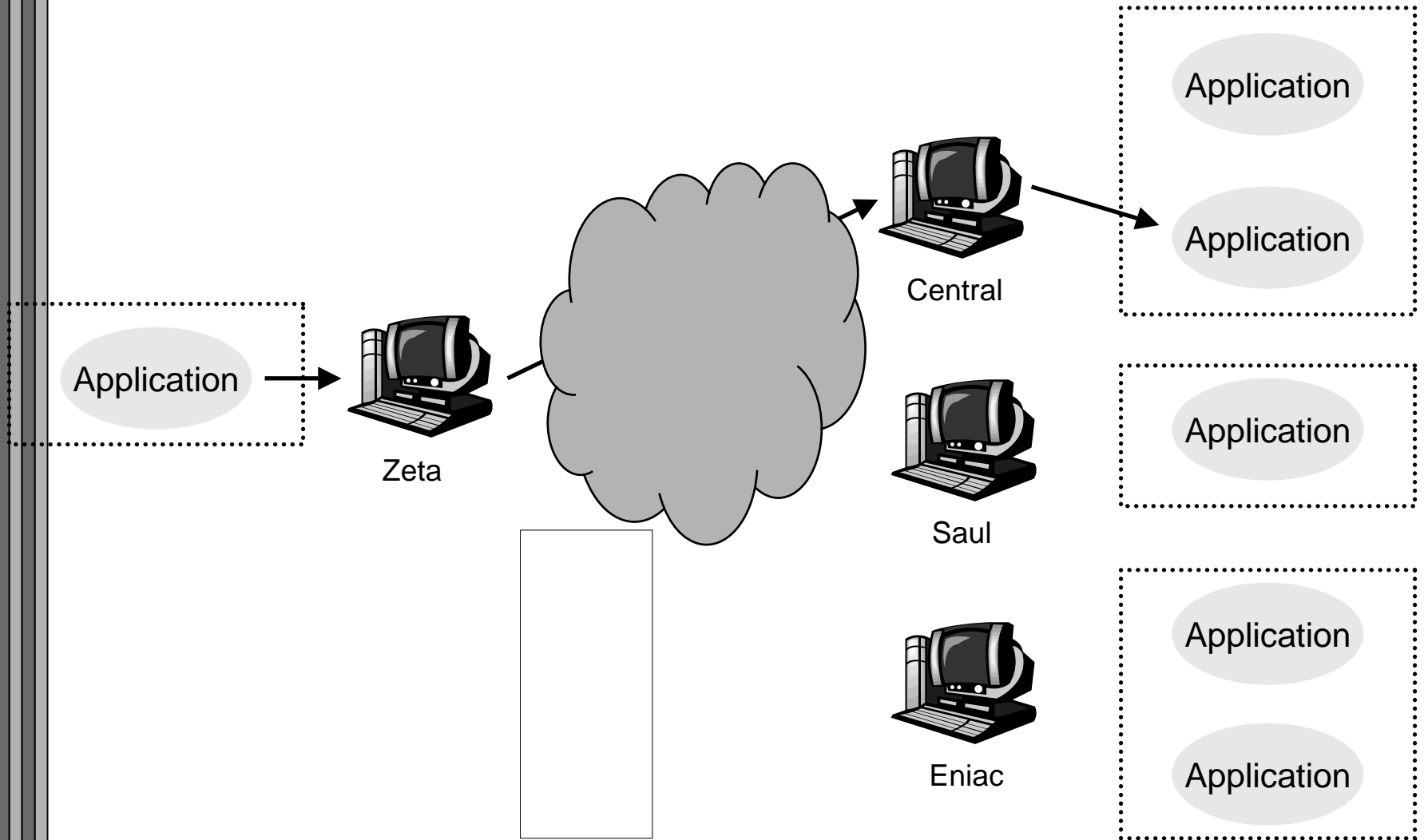




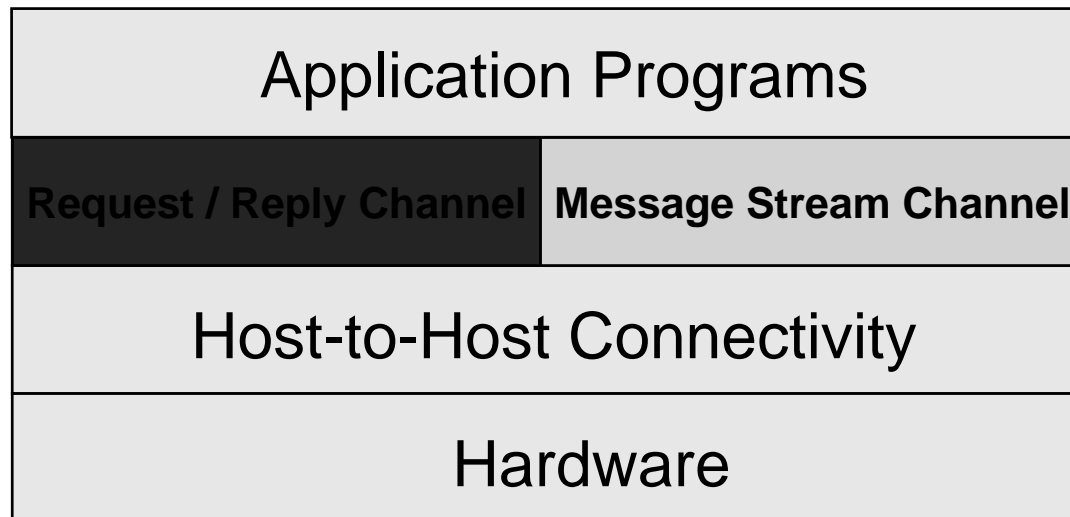
Protocols

- A *protocol* is a specification of an interface between modules (often on different machines)
- Sometimes “protocol” is used to mean the implementation of the specification.
- Examples?

Interprocess communication

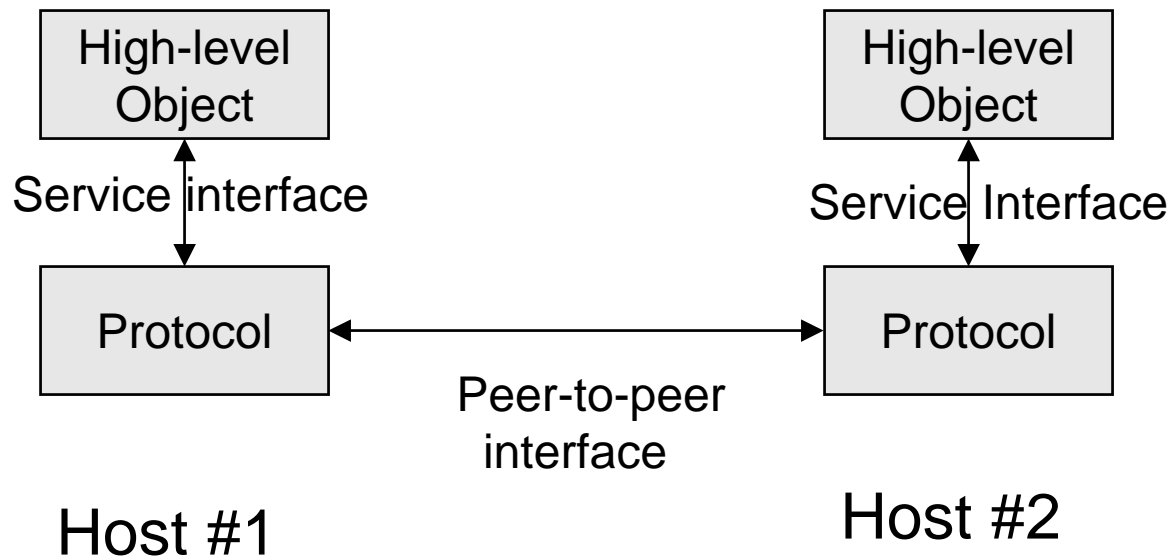


Example Protocol Stack

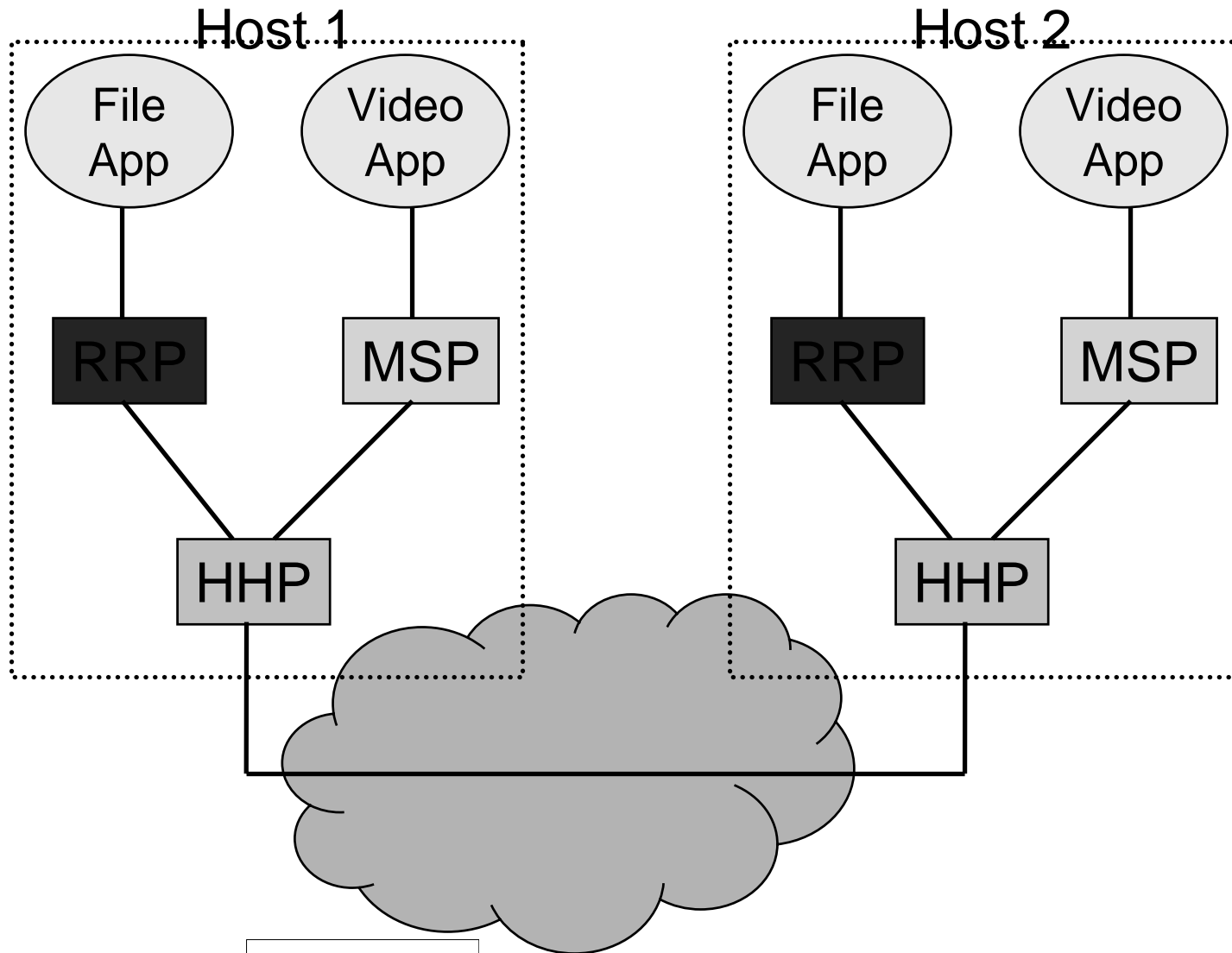


Protocol Interfaces

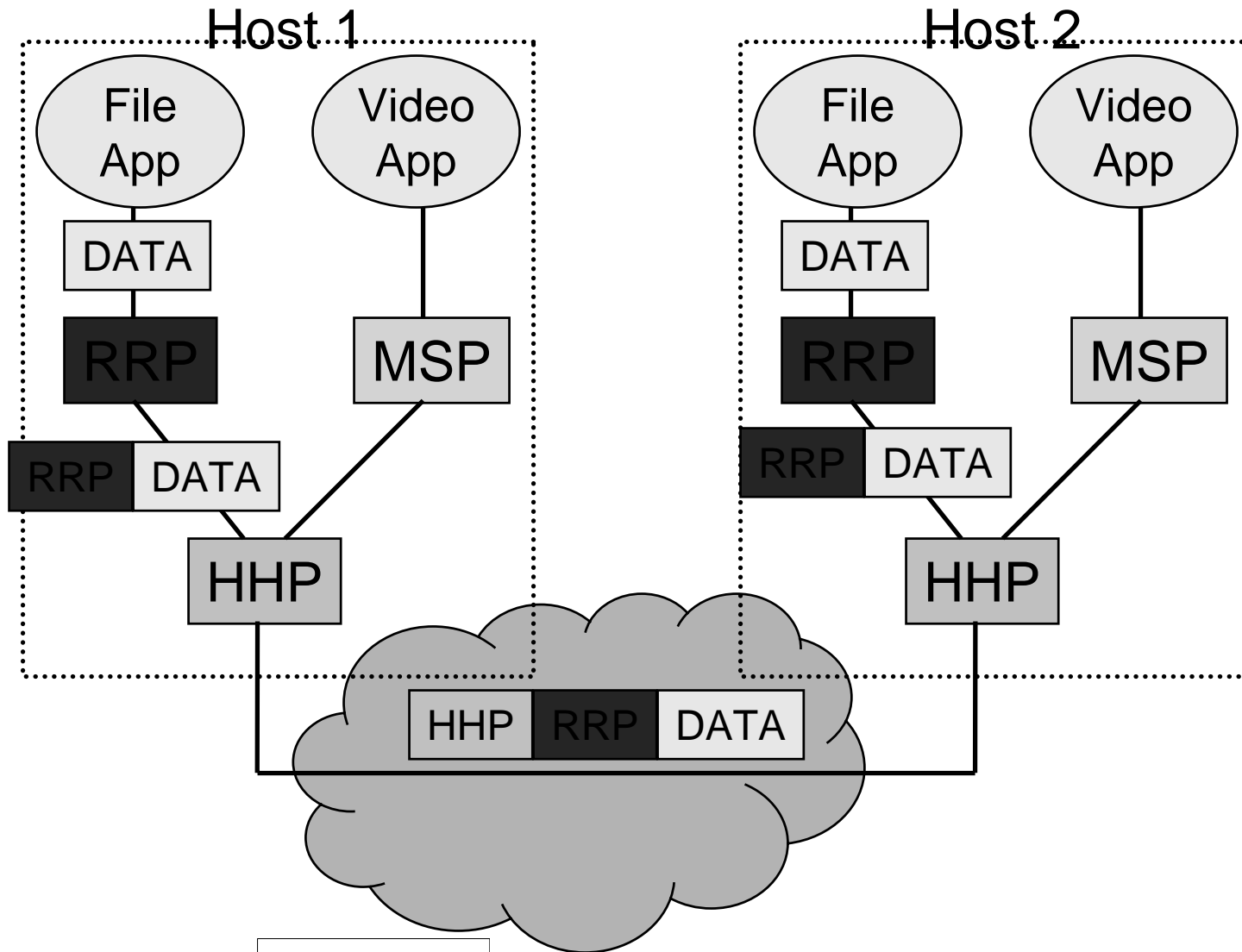
- Service Interfaces
 - Communicate up and down the stack
- Peer Interfaces
 - Communicate to counterpart on another host



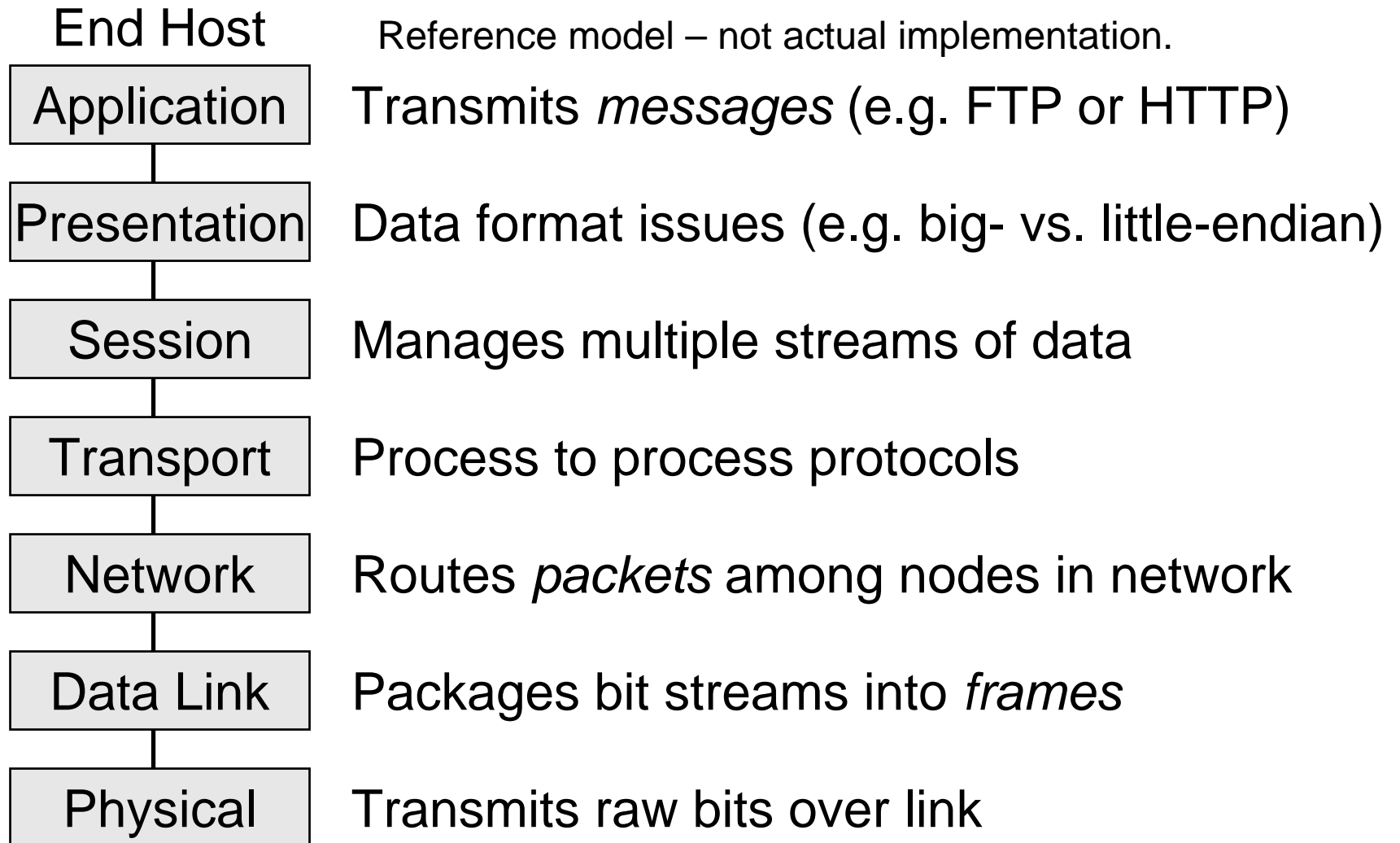
Example Protocol Graph



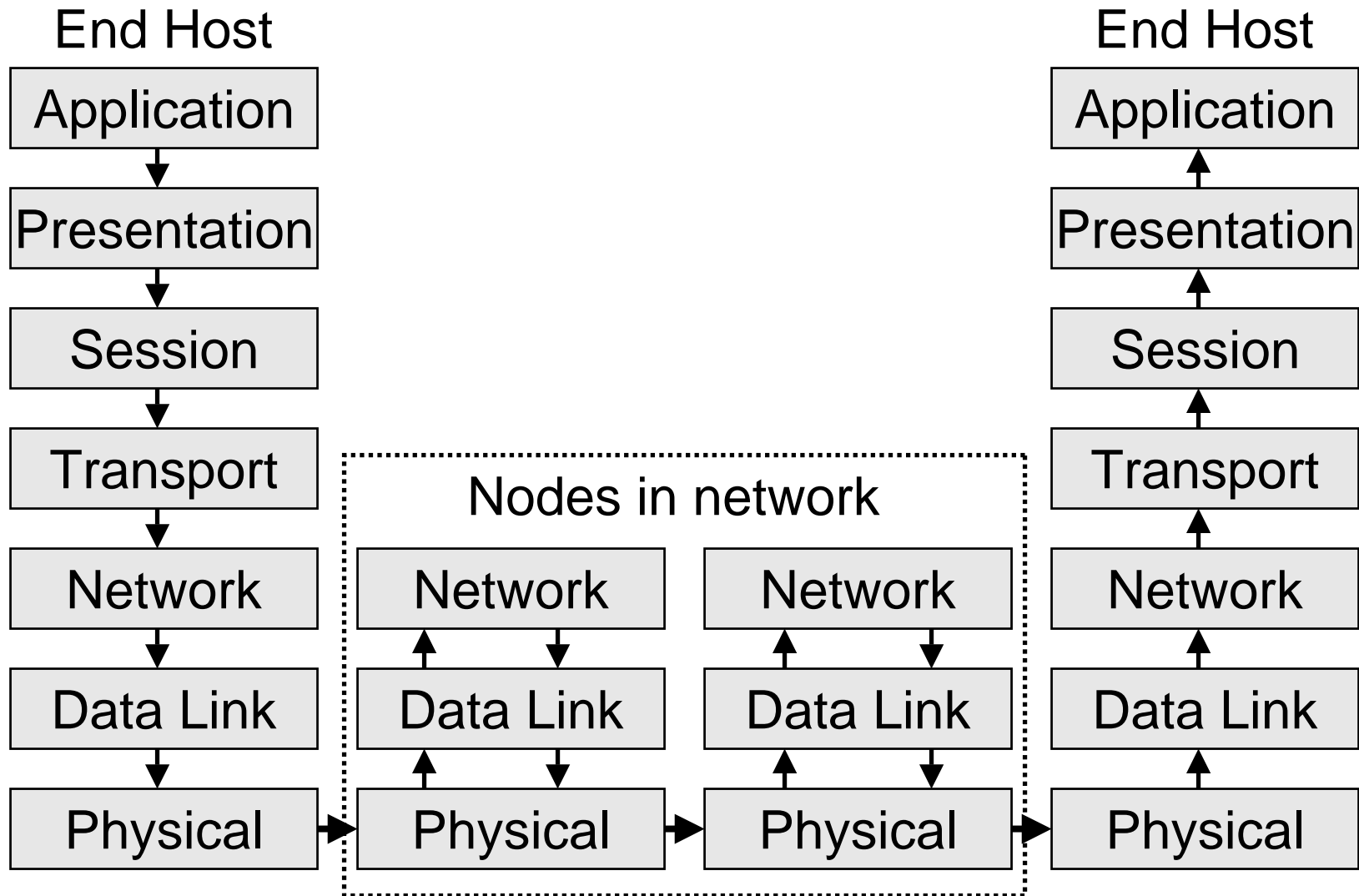
Encapsulation



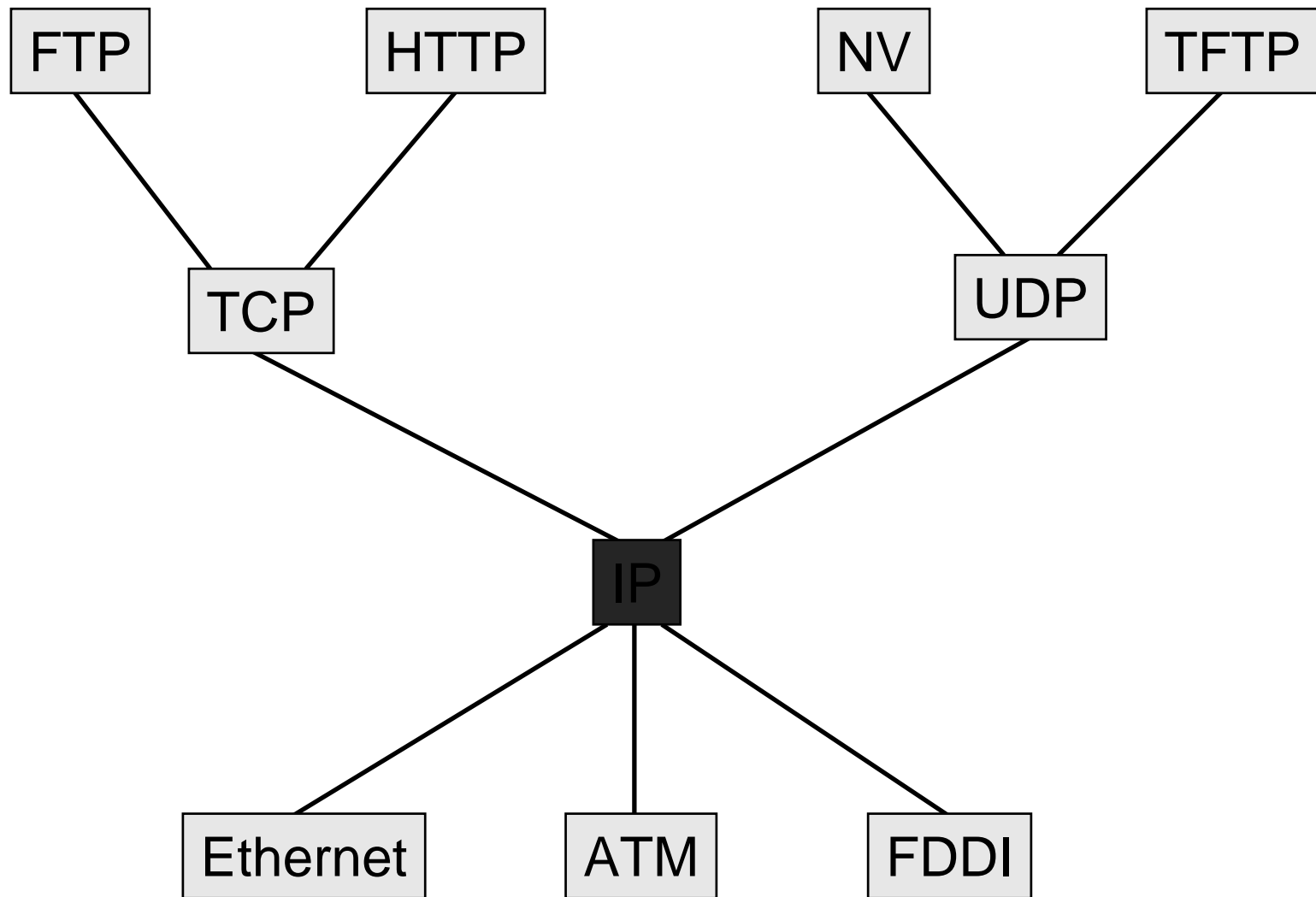
Open Systems Interconnection (OSI)



Open Systems Interconnection (OSI)



Internet Protocol Graph

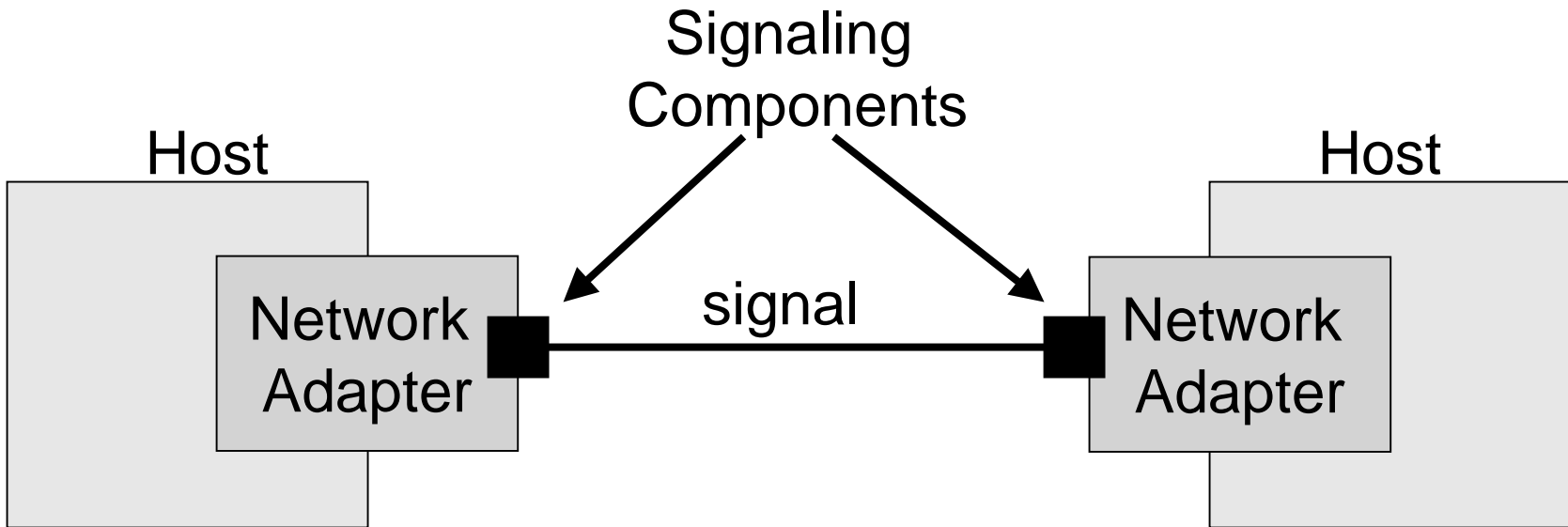




Problem: Physical connection

- Transmitting signals
- Encoding & decoding bits
- Error detection and correction
- Reliable transmission

Signaling Components

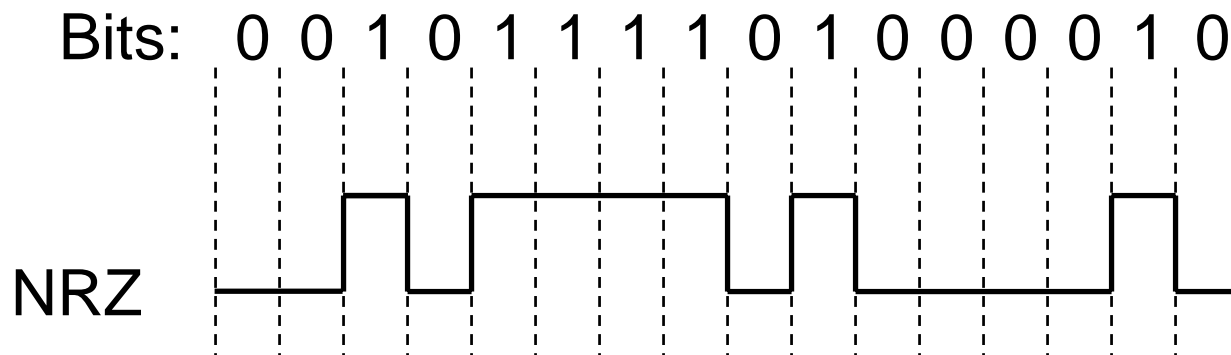


Network adapters encode streams of bits into signals.

Simplification: Assume two discrete signals—high and low.

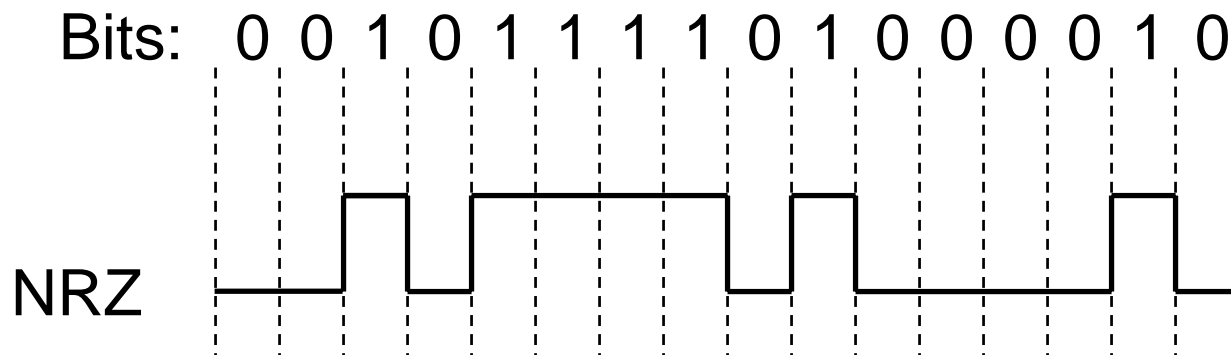
Practice: Two different voltages on copper link.

Bit Encodings



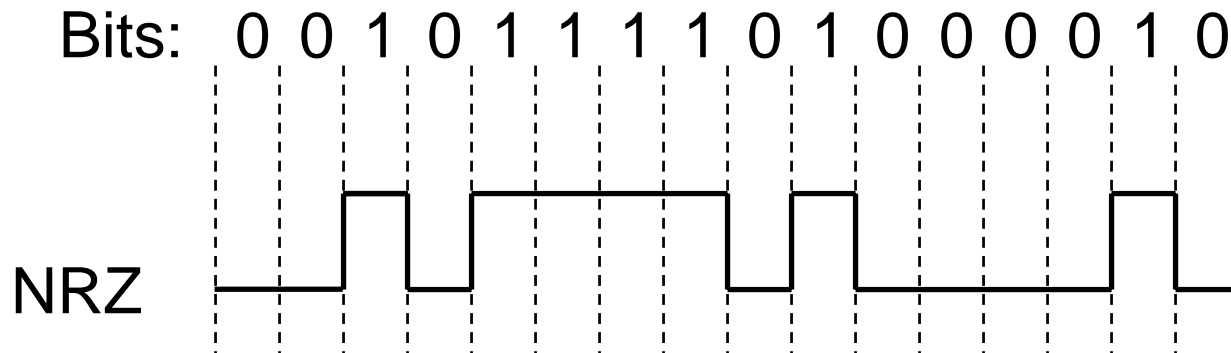
- Non-return to zero
- 1 = high, 0 = low
- Problem:
Can lead to long sequences of high or low signals.

NRZ Problem 1: baseline wander



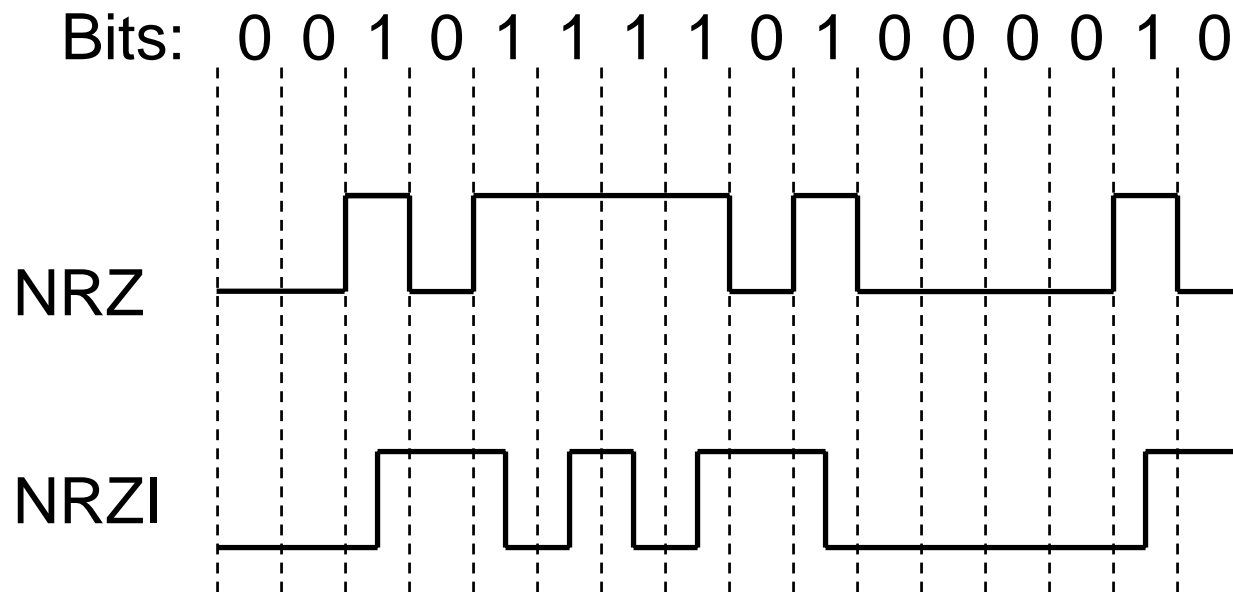
- Receivers keeps average signal seen so far
 - Significantly higher than average = 1
 - Significantly lower than average = 0
- Long periods of high/low changes average
 - Makes it harder to detect significant changes

NRZ Problem 2: clock recovery



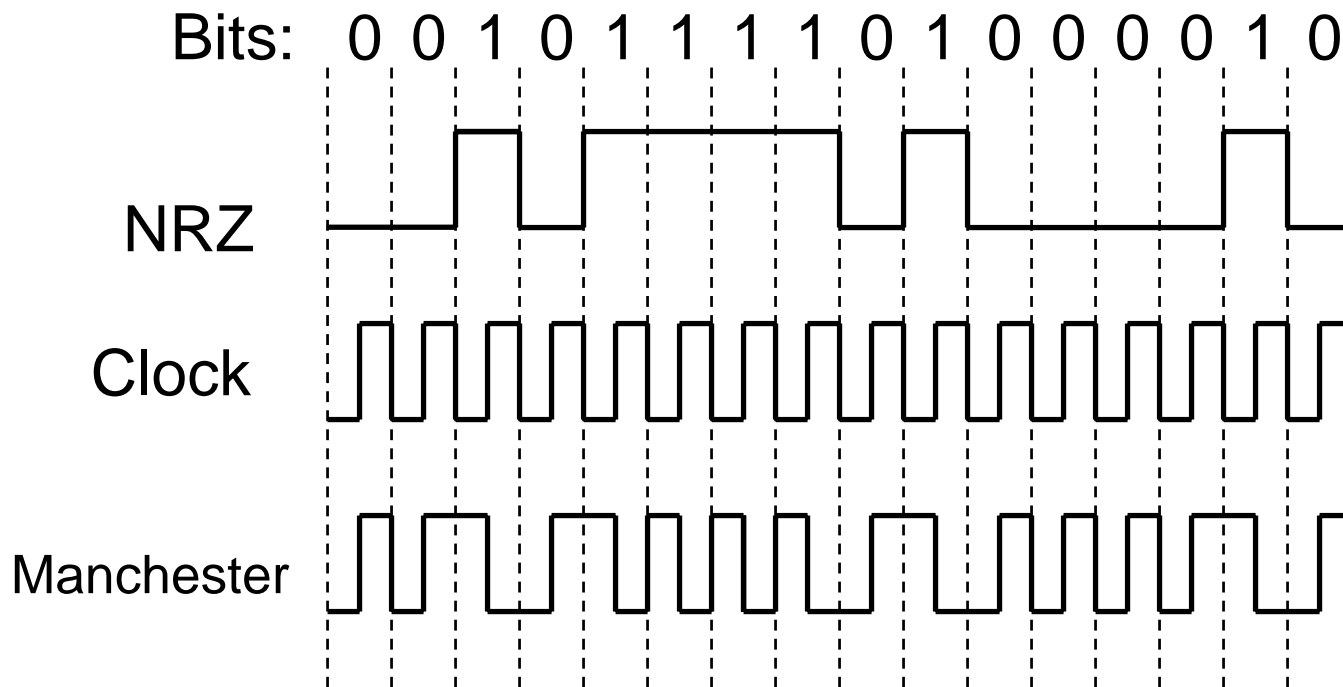
- Sender & Receiver are driven by clocks
 - Must keep clocks synchronized to recover bits
 - When signal changes, can resynchronize clocks
- Long periods of high/low lead to *clock drift*

NRZI: Non-return



- Transition from current signal to encode “1”
- Stay at current signal to encode “0”
- Does not help for consecutive 0's

Manchester Encoding



- Exclusive-OR of clock and NRZ encoding
- Doubles rate at which signal transitions are made – less time for receiver to detect changes

Efficiency

- *Bit rate vs. Baud rate*
- NRZ and NRZI: bit rate = baud rate = 100%
 - Suffer from baseline wander and clock drift
- Manchester: bit rate = $\frac{1}{2}$ baud rate = 50%
 - Prevents baseline wander and clock drift
- Can you do better?
- (Question: can bit rate exceed baud rate?)

4B/5B

- Encode 4 bits of data using 5 bits
- Choose codes to avoid long sequences of 0's
- Send 5 bit sequences using NRZI
- 80% efficiency
- Encoding
 - No more than one leading 0
 - No more than two trailing 0's

0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
...	...