CSC4200/5200 – COMPUTER NETWORKING

CONNECTING MACHINES TO A NETWORK

Instructor: Susmit Shannigrahi
sshannigrahi@tntech.edu

Tennessee TECH

Office hours

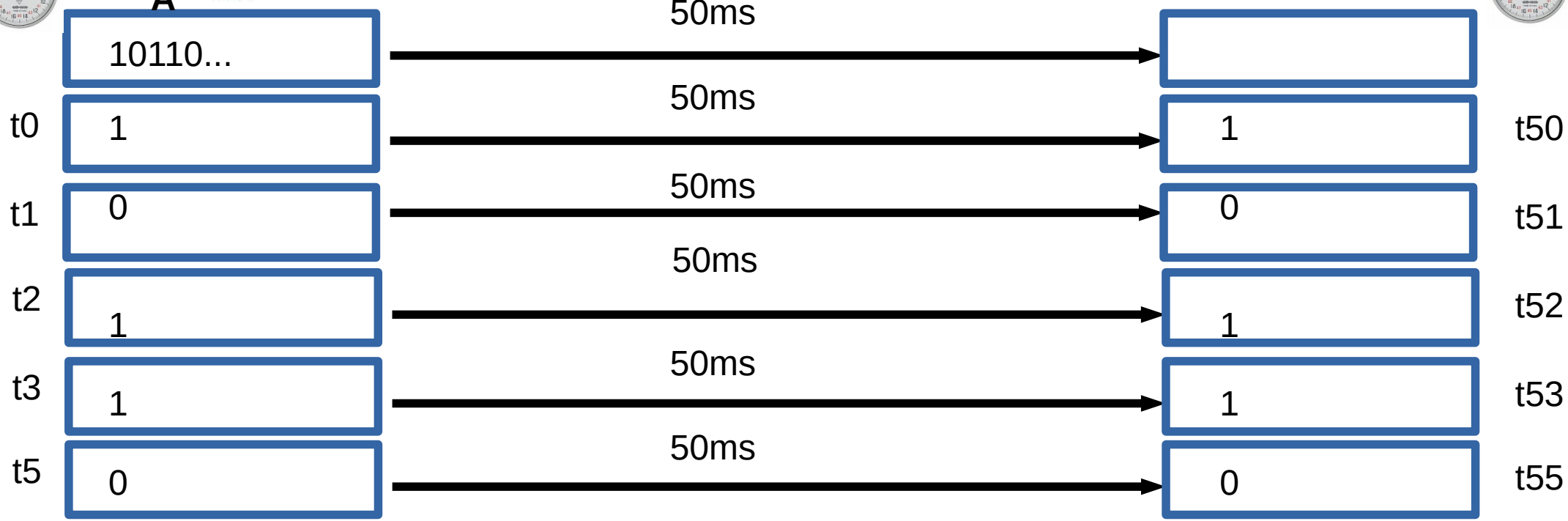
Instructor: Susmit Shannigrahi

- Office hours: Monday 2:30-4:30 PM, Wednesday 11-12AM in Bruner 417
- Email : sshannigrahi@tntech.edu

- GTA: David Reddick
  - Office hours: Wednesday 2PM-4PM in Bruner 406
  - Email: dereddick42@students.tntech.edu

Recap – Propagation and Transmission delay

Packets are made of bits. All bits must make it the next router before it can be forwarded.
Propagation delay = 50ms (time it takes for a bit to go from A to B)
Transmission delay = 1 ms (time it takes for each bit to be converted into signal)



Performance – Example

- Calculate the total time required to transfer a 1000-KB file in the following case, assuming bandwidth is 1.5 Mbps, an RTT of 50 ms, a packet size of 1 KB data, and an initial 2 × RTT of “handshaking” before data is sent. (Peterson-Davie Exercise 3, Chapter 1)

  Delay = Handshake + Transmission + Propagation + Queuing

  Delay = 2*50ms + (1000*1024*8)/(1.5*1000*1000) second + 50/2ms + 0 = 5.586seconds

- **Propagation delay = First bit from sender to receiver**

Performance – Example

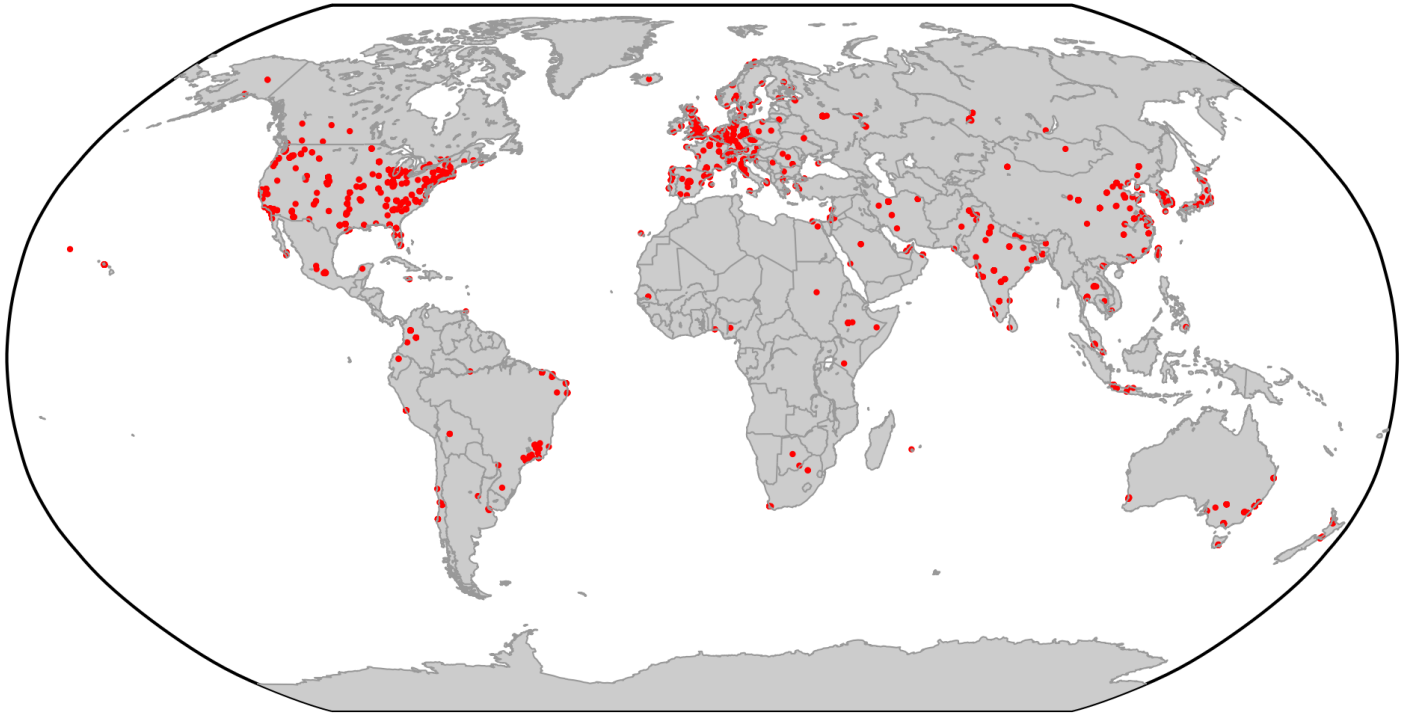
- Calculate the total time required to transfer a 1.5-MB file in the following cases, assuming an RTT of 80 ms, bandwidth= 10Mbps, a packet size of 1 KB data, and an initial 2 × RTT of “handshaking” before data is sent:

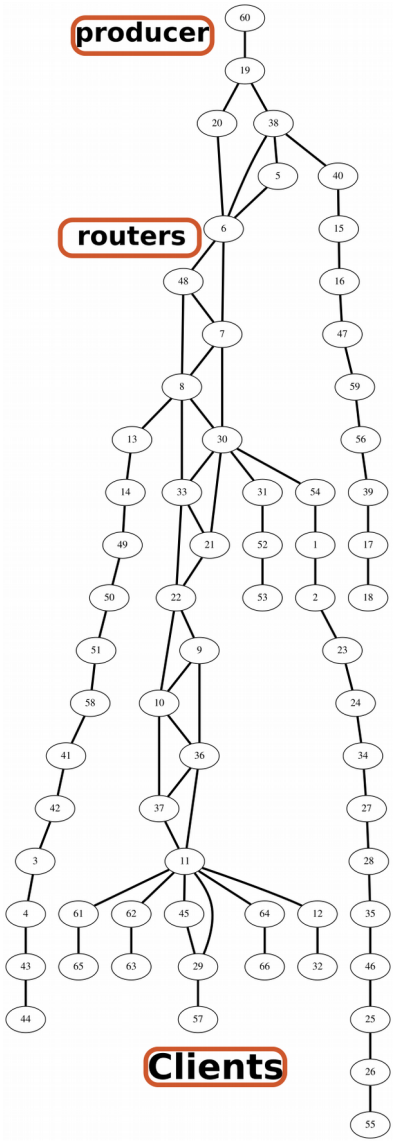
  Delay = Handshake + Transmission + Propagation + Queuing

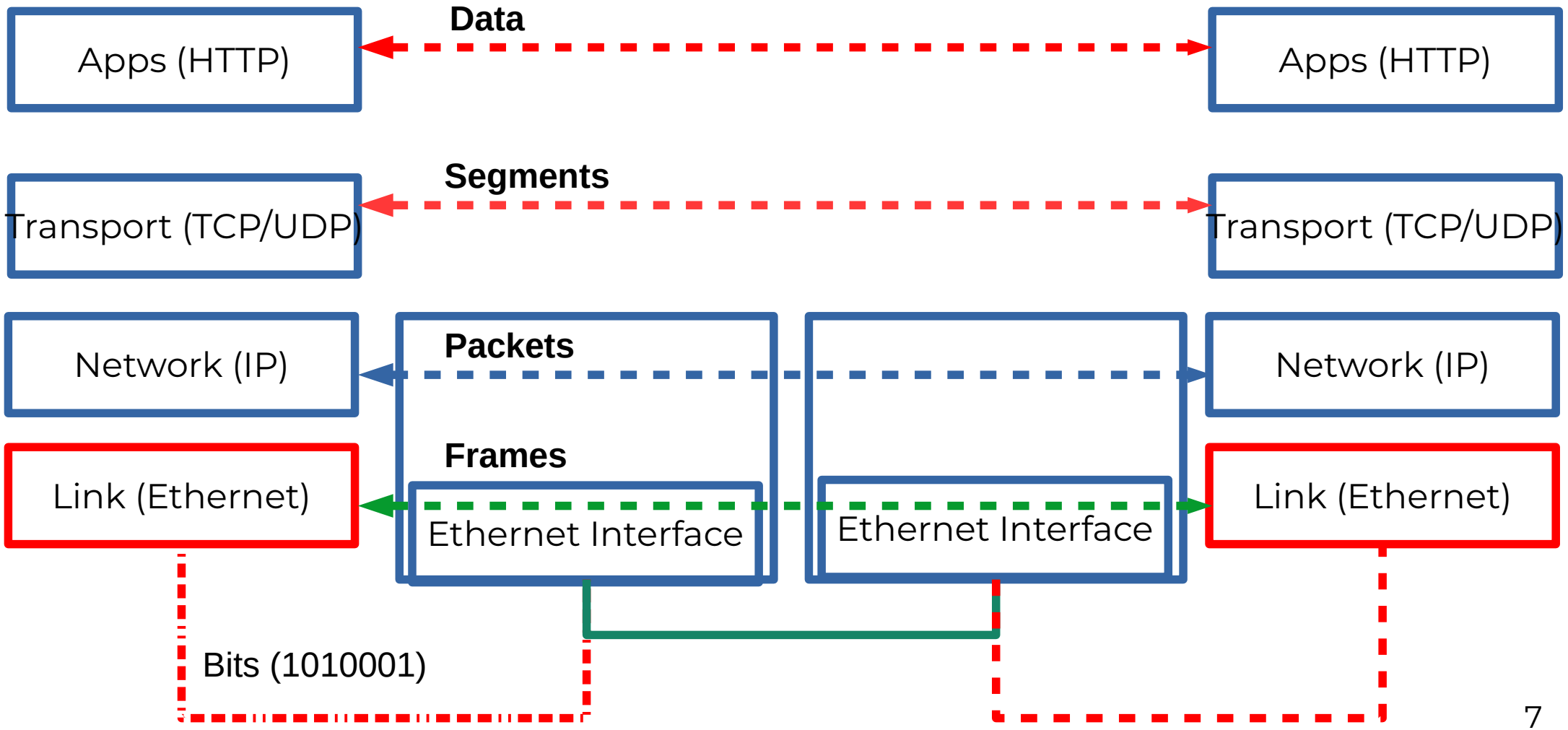
  1.5 MB = 12582912 bits (How?) 1.5*1024KB = 1.5*1024*1024Bytes = 1.5*1024*1024*8 bits
  Total delay = 2 initial RTTs (160 ms) + 12,582,912/10,000,000 bps (transmit) + RTT/2 (propagation) ≈ 1.458 seconds.

- **Propagation delay = First bit from sender to receiver**

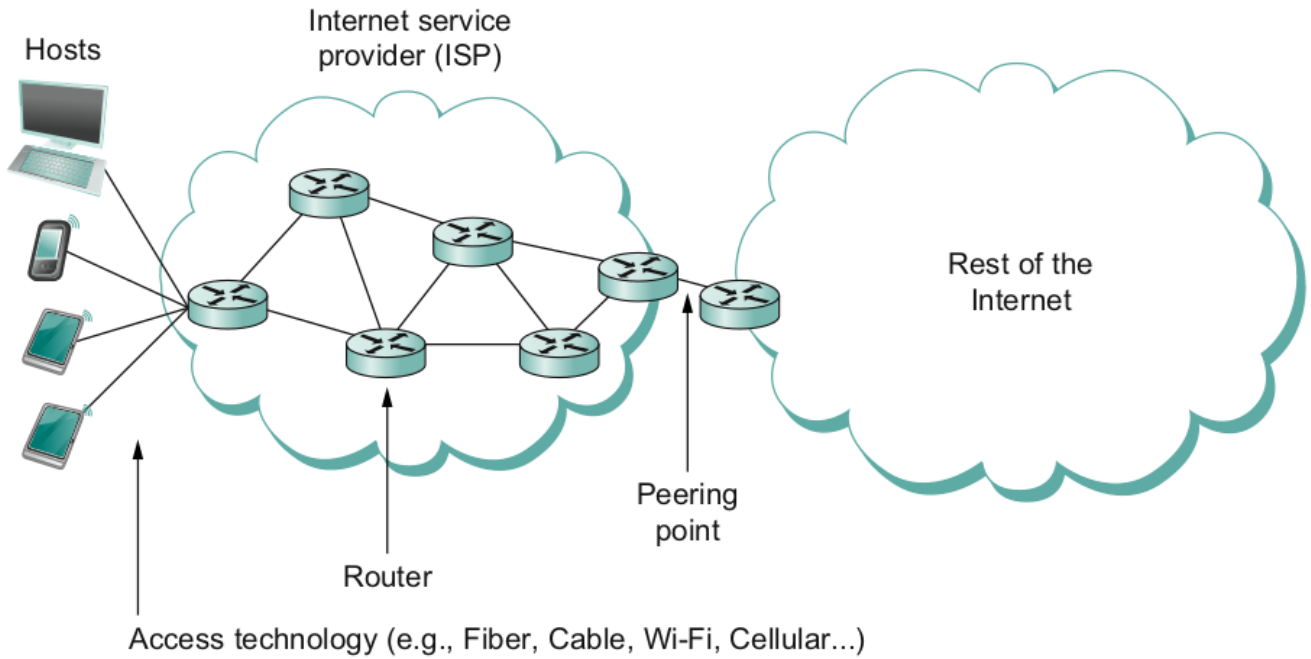
Recap – Network = Graph (Nodes + Links)






Data
Apps (HTTP)
Apps (HTTP)
Segments
Transport (TCP/UDP)
Transport (TCP/UDP)
Packets
Network (IP)
Network (IP)
Frames
Link (Ethernet)
Ethernet Interface
Ethernet Interface
Link (Ethernet)
Bits (1010001)

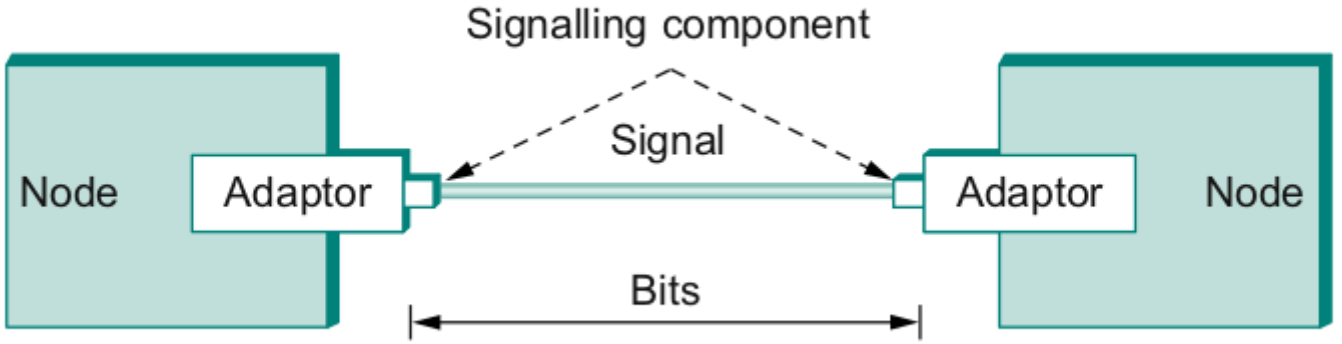
What does it take to create a link?



- Common abstractions
  - Why?

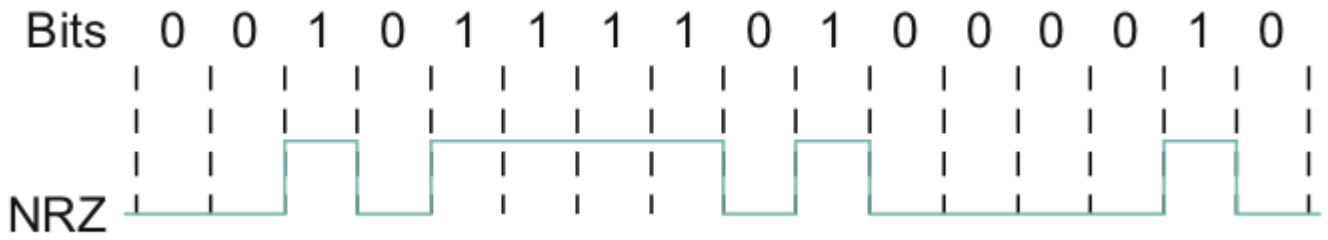
Packet to Low level Signals

- Bit pattern - 0101001
  - Must encode it into electrical signals and then decode it on the other end!



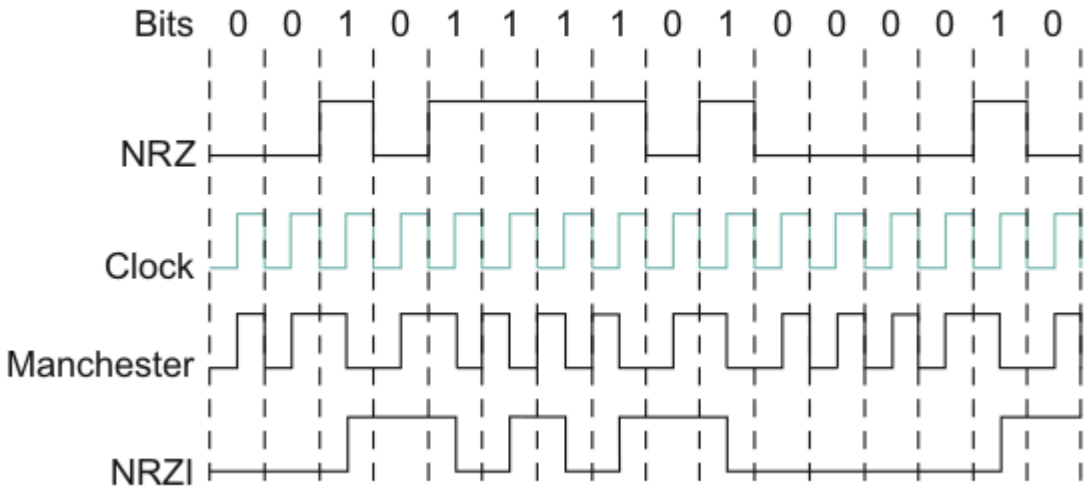
One easy way - NRZ

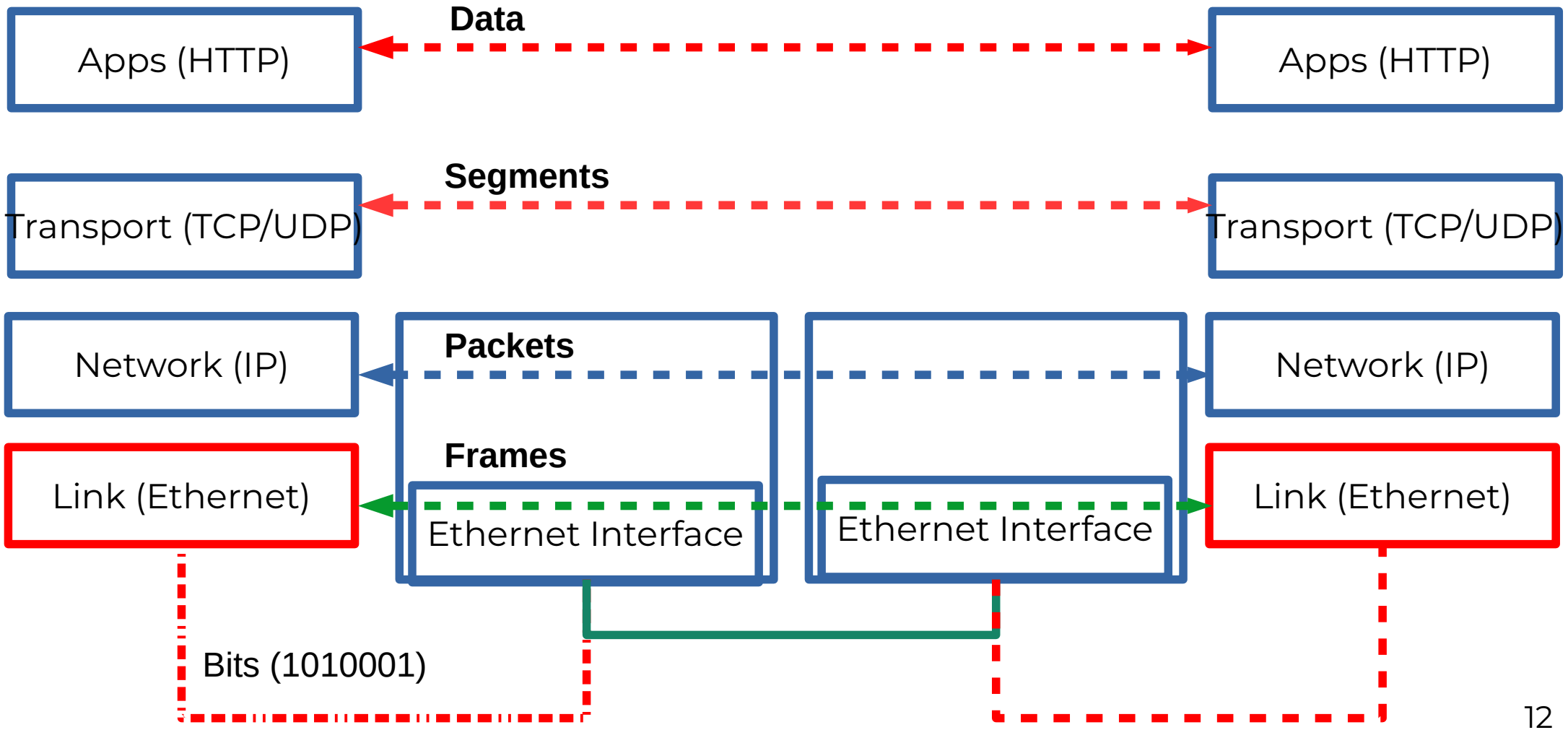
- 0 = low, 1= low
- Problem if you have too many 0s or 1s in a row
  - Baseline wander (read it in the book)



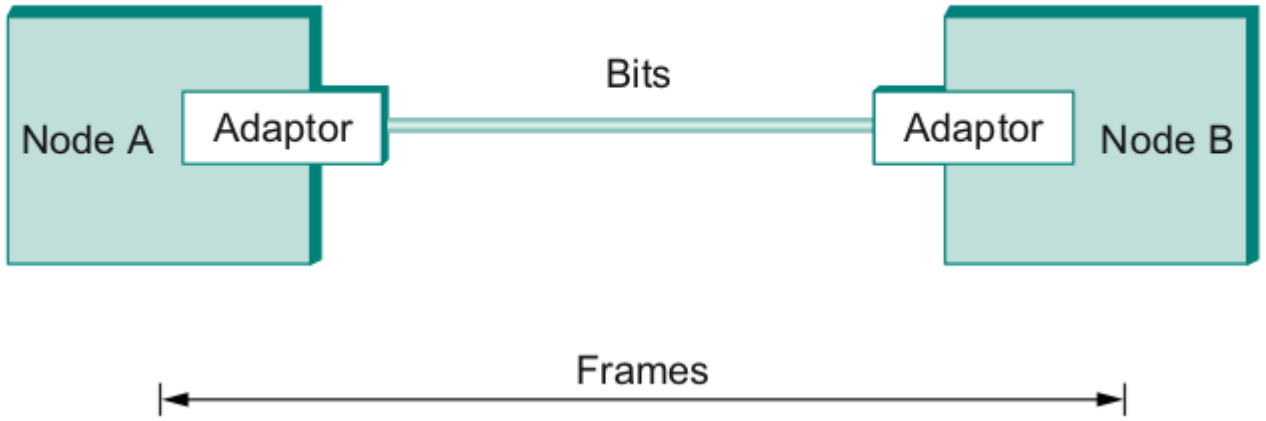
Other ways – NRZI/ Manchester Encoding

- NRZI – 1=transition, 0= Don’t
- Machester encoding – XOR of clock + NRZ data




Data
Apps (HTTP)
Apps (HTTP)
Segments
Transport (TCP/UDP)
Transport (TCP/UDP)
Packets
Network (IP)
Network (IP)
Frames
Link (Ethernet)
Ethernet Interface
Ethernet Interface
Link (Ethernet)
Bits (1010001)

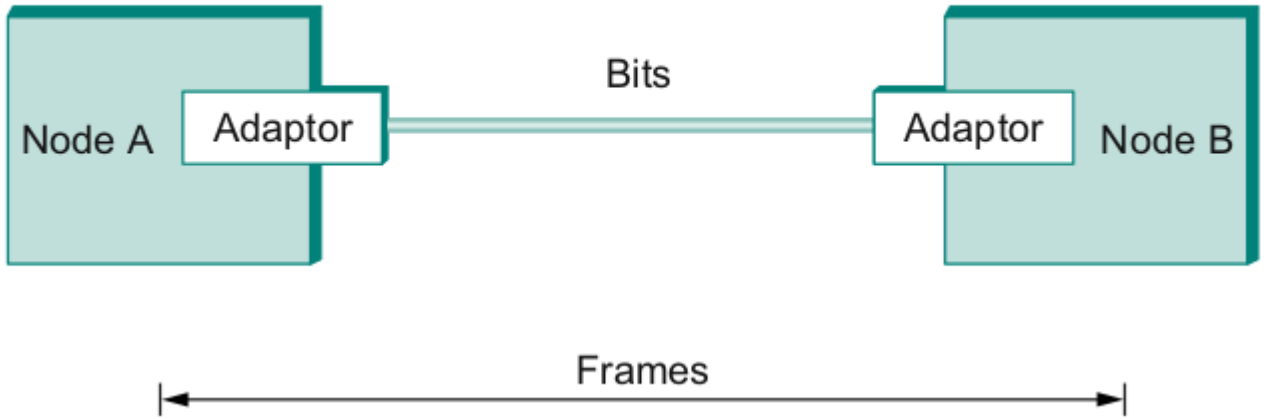
Frames – bag of bits



- Bits - between adaptors
- Frames – between hosts (two computers)
  - The job of adaptors is to find frames in a bit sequence

- Frames are link layer protocols

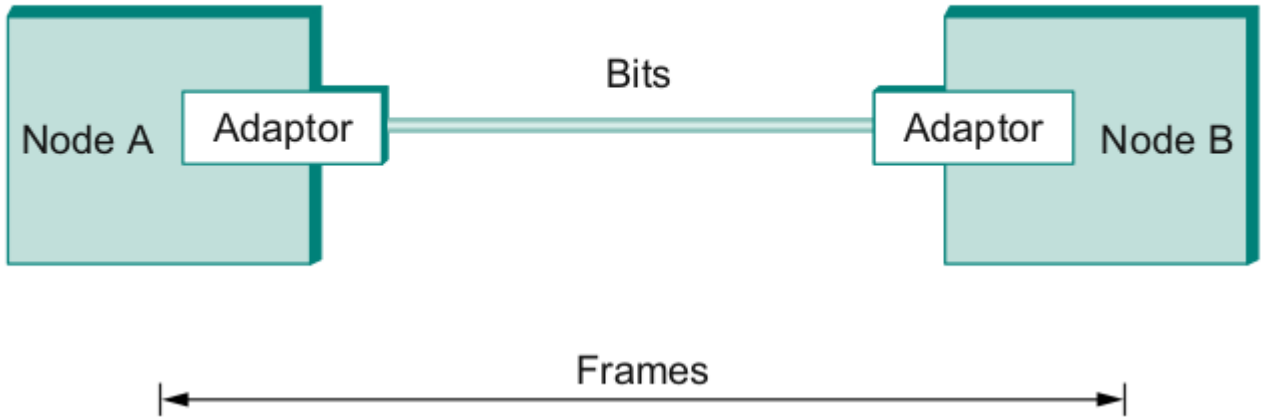
Frames – bag of bits



Bits flow between adaptors, frames between hosts

Let’s see it in wireshark!

Frames – bag of bits



- Sending side – encapsulation, add error check bits, flow control
- Receiving side – extract frames, check for error, flow control

Framing

- Point-to-point
  - Special start of text character denoted as Flag
    - 01111110
  - Address, control : default numbers
  - Protocol for demux : IP / IPX
  - Payload : negotiated (1500 bytes)
  - Checksum : for error detection

Point to Point Links

- Requirements:
  - Packet framing: encapsulation of network-layer datagram in data link frame
  - Bit transparency: Does not care about what its carrying
  - Error Detection: Correction is on upper layers
  - Connection liveness
  - Network layer address negotiation

Point to Point Links

- Flag: 01111110
  - Address: Does nothing
  - Control; Does nothing
  - Protocol: IP/ATM
  - Payload: Whatever
  - Checksum: Discussing later
- Flag: 01111110

Point to Point Links – Byte stuffing

- Flag: 01111110
  - Data transparency means it can also carry 01111110
  - How do you differentiate between data and flag?
    - Two back to back sequences of 01111110
    - Two sequences = Data, discard first
    - One sequence = Flag

Error Detection

- Bit errors are introduced into frames
  - Because of electrical interference and thermal noises
- Detecting Error
- Correction Error
- Two approaches when the recipient detects an error
  - Notify the sender that the message was corrupted, so the sender can send again.
    - If the error is rare, then the retransmitted message will be error-free
  - Using some error correct detection and correction algorithm, the receiver reconstructs the message

Error Detection

- Common technique for detecting transmission error
  - CRC (Cyclic Redundancy Check)
    - Used in HDLC, DDCMP, CSMA/CD, Token Ring
  - Other approaches
    - Two Dimensional Parity (BISYNC)
    - Checksum (IP)

Error Detection

- Basic Idea of Error Detection
  - To add redundant information to a frame that can be used to determine if errors have been introduced

0	1	0	1	0	**0**

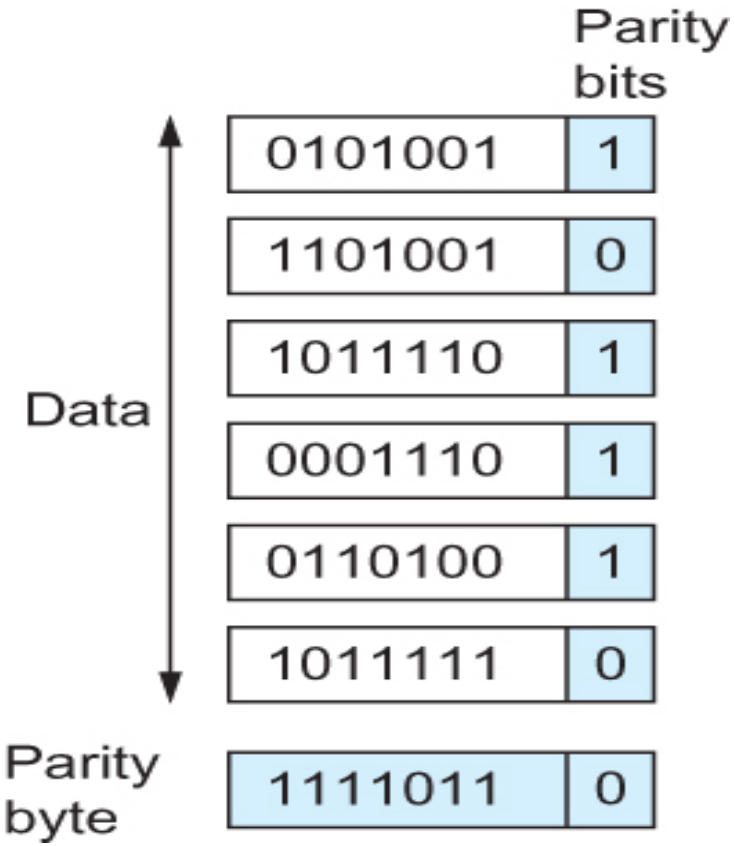
0	1	0	1	1	**1**

Number of 1s
- Odd 1s = Parity bit 0
- Even 1s = Parity bit 1

Two-dimensional parity

- Two-dimensional parity does a similar calculation
- Extra parity byte for the entire frame, in addition to a parity bit for each byte
- Two-dimensional parity catches all 1-, 2-, and 3-bit errors and most 4-bit errors

Two-dimensional parity



Two Dimensional Parity

Internet Checksum Algorithm

- Not used at the link level
- Add up all the words that are transmitted and then transmit the result of that sum
  - The result is called the checksum
- The receiver performs the same calculation on the received data and compares the result with the received checksum

Internet Checksum Algorithm

Server Side

1. It treats segment contents as sequence of 16-bit integers.

2. All segments are added. Let's call it sum.

3. Checksum : 1's complement of sum.(In 1's complement all 0s are converted into 1s and all 1s are converted into 0s).

4. Sender puts this checksum value in UDP checksum field.

Client Side:

1. Calculate checksum

2. All segments are added and then sum is added with sender's checksum.

3. Check that any 0 bit is presented in checksum. If receiver side checksum contains any 0 then error is detected. So the packet is discarded by receiver.

Internet Checksum Algorithm (RFC 1071)

- A = 0110011001100110
- B = 0101010101010101

  1011101110111011
- C = 0000111100001111

  1100101011001010 (sum of all segments)

  0011010100110101 (1's complement, 1→0, 0→ 1) <= this is the checksum

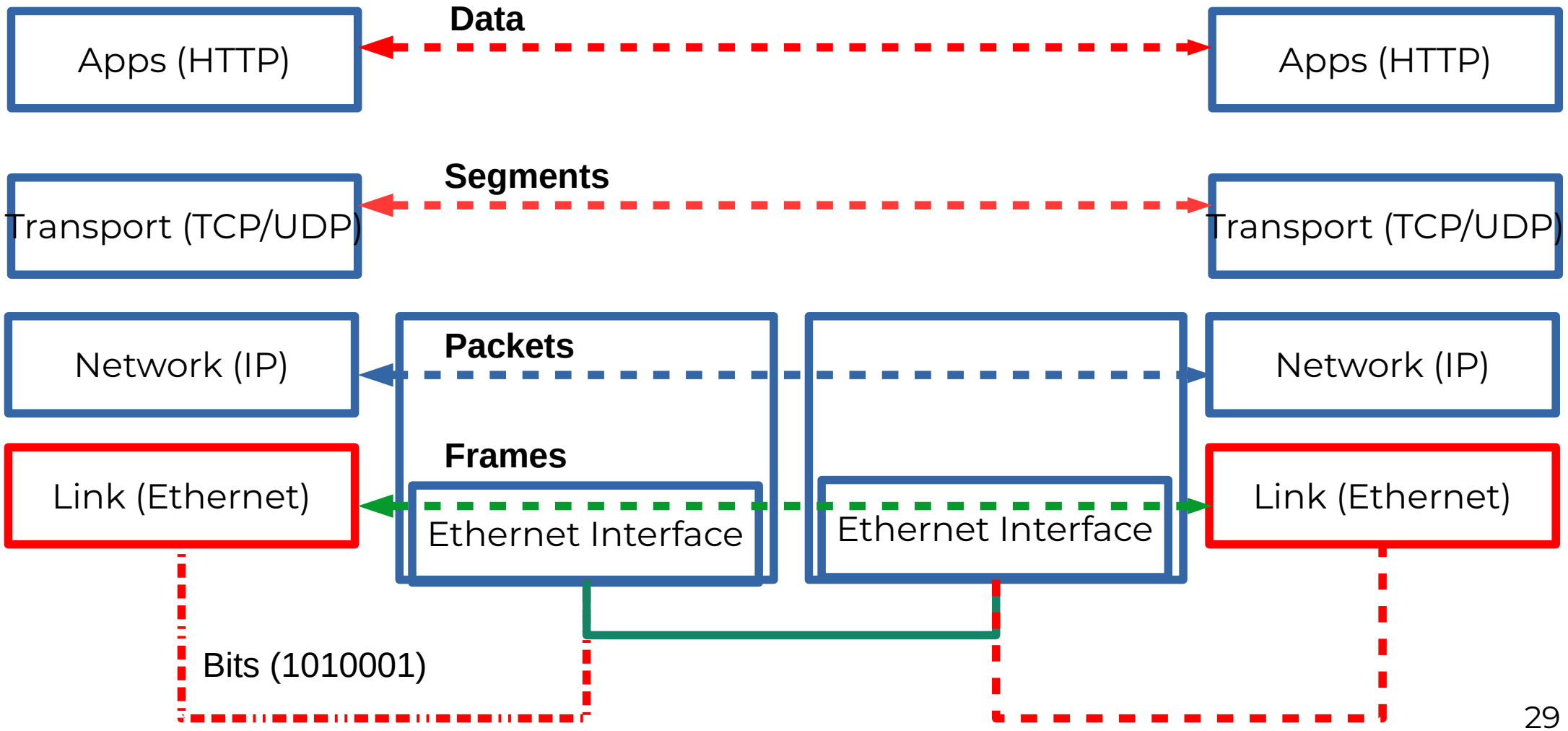
  At receiver:

  Add sum of all segments and checksum

  - If correct, all 1s

Others - Cyclic Redundancy Check (CRC)

- Reduce the number of extra bits and maximize protection
- Very interesting math but we will skip it for now.


Apps (HTTP)
Data
Apps (HTTP)
Transport (TCP/UDP)
Segments
Transport (TCP/UDP)
Network (IP)
Packets
Network (IP)
Link (Ethernet)
Frames
Ethernet Interface
Ethernet Interface
Link (Ethernet)
Bits (1010001)

All these for what?

- Handles bit transmission errors that might occur
- Link access – who can access the link at a given time?
- Flow control – control the pace that does not overwhelm sending and receiving devices
- **Reliable delivery – next topic**
- It permits the transmission of data to Layer 3, the network layer, where it is addressed and routed.

Next Steps

- Read Chapter 2

- Next lecture – Reliable Delivery