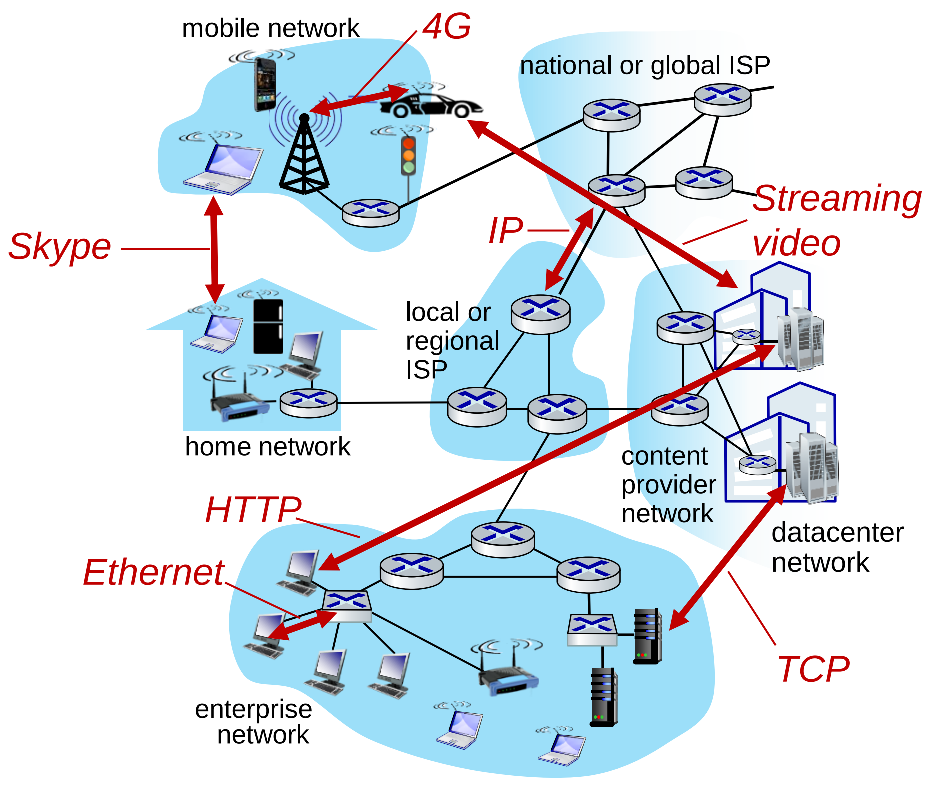
CSC358: Principles of Computer Networks

Week 2 Practical:

A Day In Life of A Packet



Winter 2025, University of Toronto, Mississauga

Today’s Outline

- In week 1 lecture, you saw
  - What is the Internet made of?
  - What is a protocol?
  - Layering and encapsulation
- In today’s practical session, you will see what a packet goes through when you visit a website.
- This example helps us better understand protocols, layering, and encapsulation
- But, before this example, we’ll review some fundamental concepts

Review: Layers; Multiplexing

Applications with Multiple Streams

- Name three applications that require tying multiple streams together.
  - Web Browsers:
    - Browsers make multiple HTTP(S) requests in parallel (*e.g.*, fetching HTML, images, and scripts).
    - The application layer (browser logic) integrates these responses to render the web page seamlessly.
  - Video Conferencing:
    - Audio, video, and screen-sharing streams are transmitted as separate connections (often using RTP over UDP).
    - The application layer synchronizes these streams for real-time playback.
  - File Downloading:
    - Download managers split a file into chunks and download each chunk via a separate stream (*e.g.*, parallel HTTP/FTP connections).
    - The application layer reassembles these chunks into the original file.

Applications with Multiple Streams, cont’d

- In the context of OSI model, which layer manages tying multiple streams together?
  - Session layer
- In the context of 4-layer Internet model, which layer manages tying multiple streams together?
  - Application layer

Transport Layer and Multiplexing

- Which of the following statements are true:
  a) Transport layer multiplexing assigns different IP addresses to multiple data streams.
  b) Transport layer multiplexing manages multiple logical sessions over a single application-level connection, such as synchronizing audio and video streams in a video call.
  c) Transport layer multiplexing ensures that data from multiple applications on the same device is sent and received correctly using port numbers.
  d) Application layer is responsible for splitting data into chunks to be sent over multiple streams.

Transport Layer and Multiplexing, cont’d

- Which of the following statements are true:
  a) Transport layer multiplexing assigns different IP addresses to multiple data streams --> False
     - IP address is not used for multiplexing multiple data streams.

  b) Network layer multiplexing manages multiple logical sessions over a single application-level connection, such as synchronizing audio and video streams in a video call --> False
     - Session Layer does this.

  c) Transport layer multiplexing ensures that data from multiple applications on the same device is sent and received correctly using port numbers --> True

  d) Application layer is responsible for splitting data into chunks to be sent over multiple streams --> False
     - Transport layer does this.

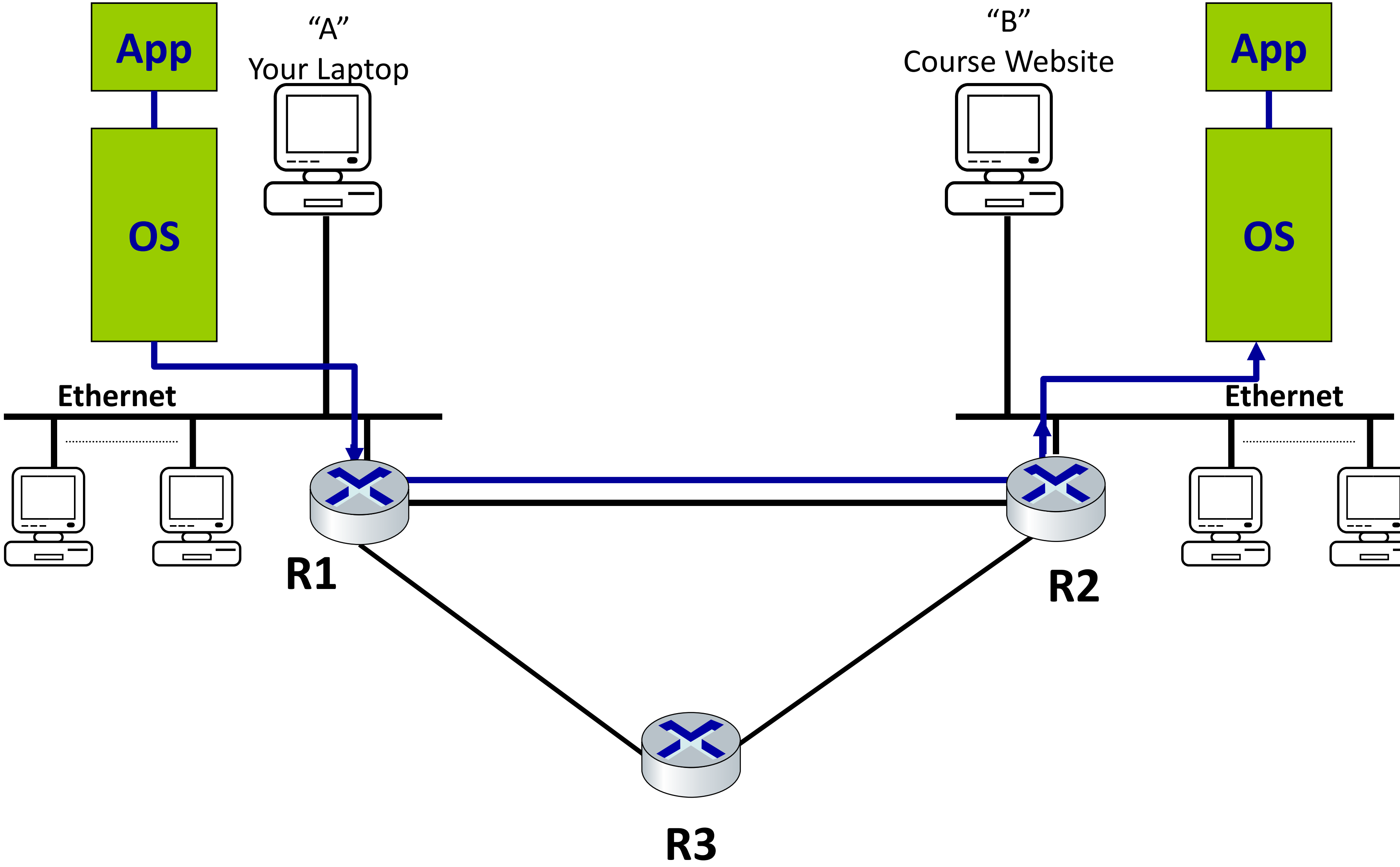
Transport Layer and Multiplexing, cont’d

- There are likely to be many processes running on any given host
- So, it is required to add a level of multiplexing/demultiplexing,
  - thereby allowing multiple application processes on each host to share the network.
- Transport layer protocols, e.g., UDP and TCP, support a multiplexing/demultiplexing mechanism
  - This allows multiple application programs on any given host to simultaneously carry on a conversation with their peers.
- TCP and UDP use source and destination port numbers for multiplexing/demultiplexing.
  - Port numbers will be discussed later in the course

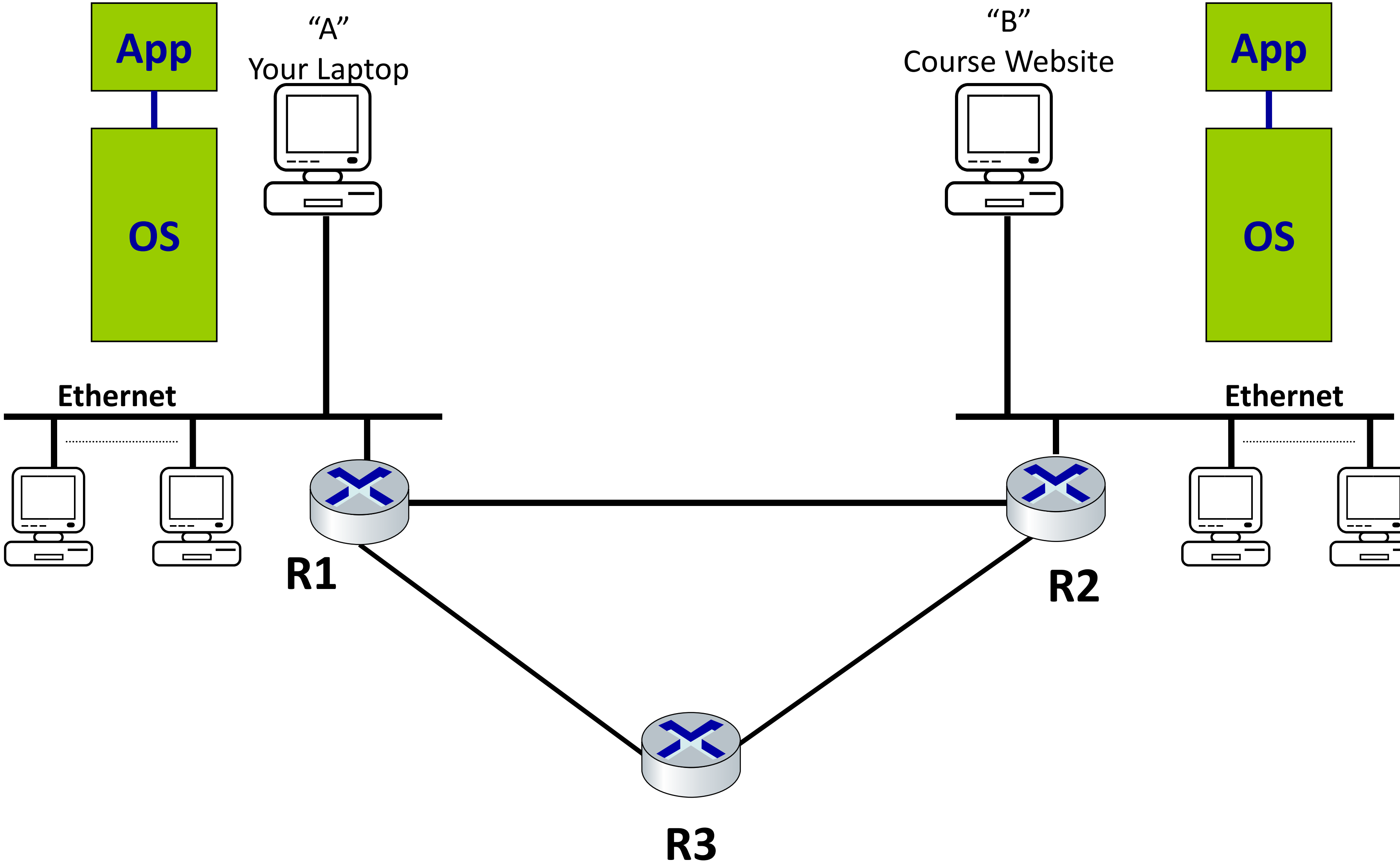
A Day In Life of A Packet

HTTP over the Internet, with TCP/IP and Ethernet

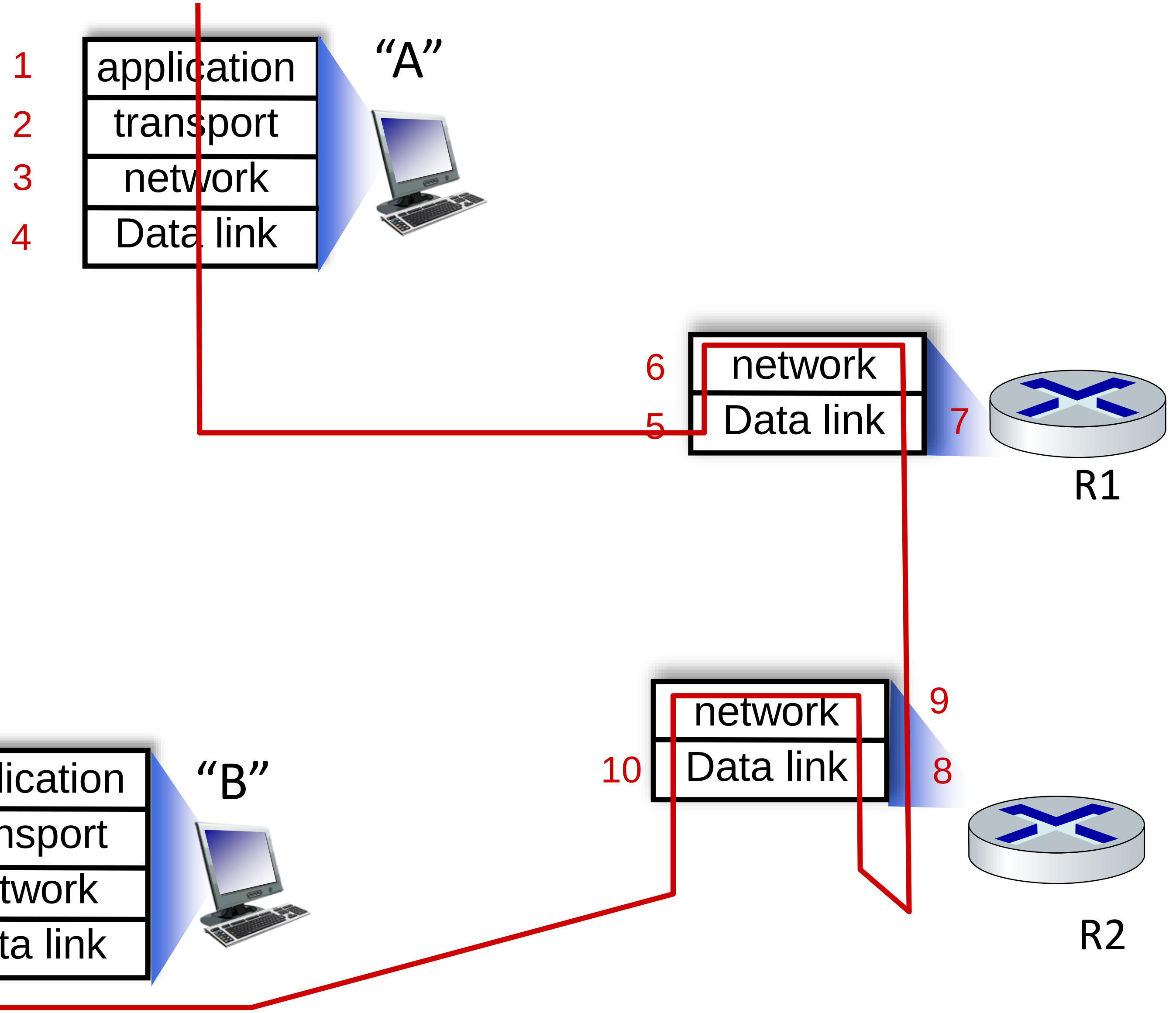
HTTP over the Internet, with TCP/IP and Ethernet



HTTP over the Internet, with TCP/IP and Ethernet



Encapsulation: an end-end view



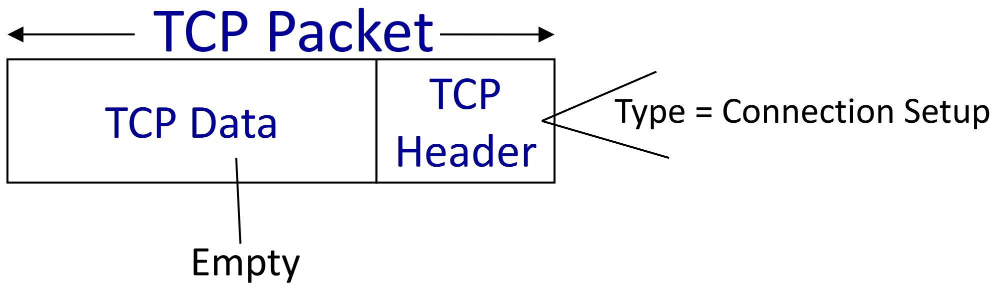
In the Sending Host

1. Application

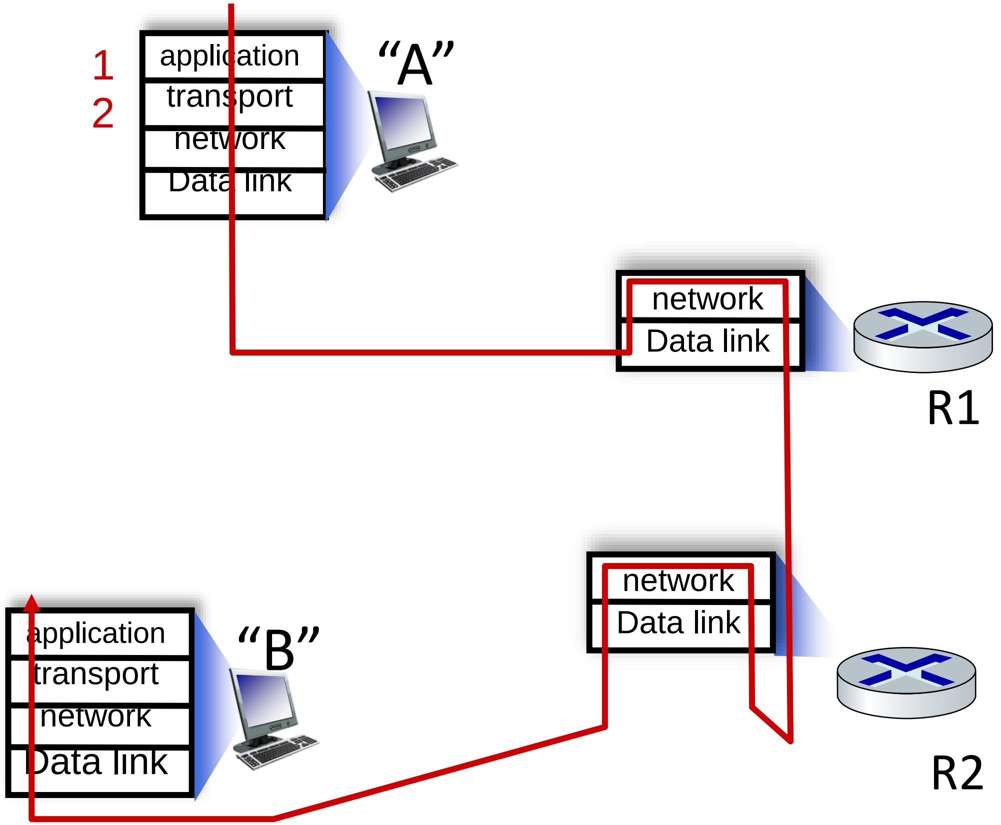
- requests a TCP connection with “B”, using the Application-Programming Interface (API)

2. The Transmission Control Protocol (TCP)

- creates a TCP “Connection setup” packet



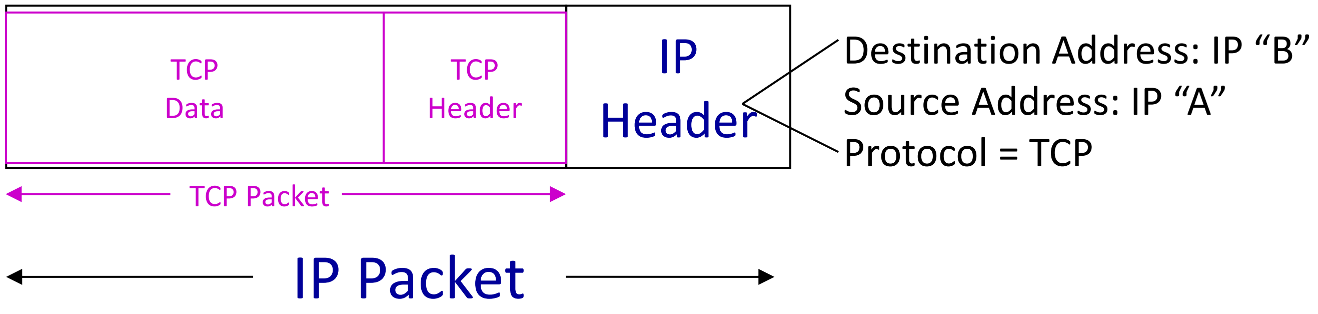
- TCP requests IP packet to be sent to “B”



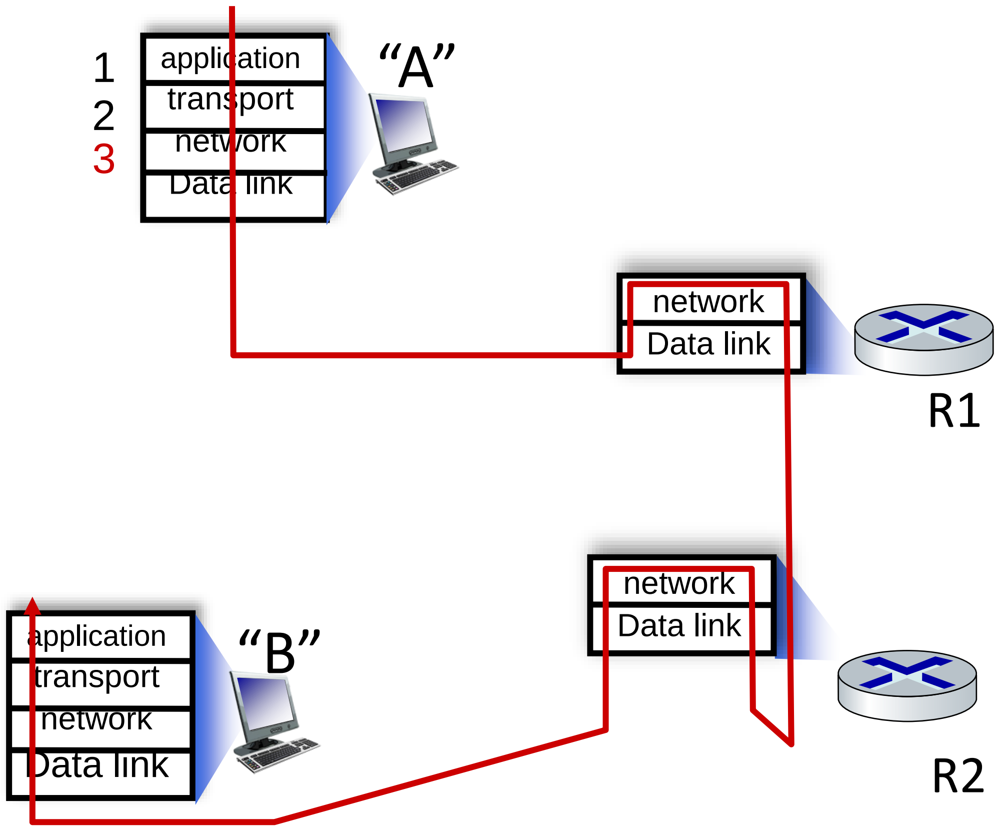
In the Sending Host – cont'd

3. Internet Protocol (IP)

- creates IP packet with correct addresses.



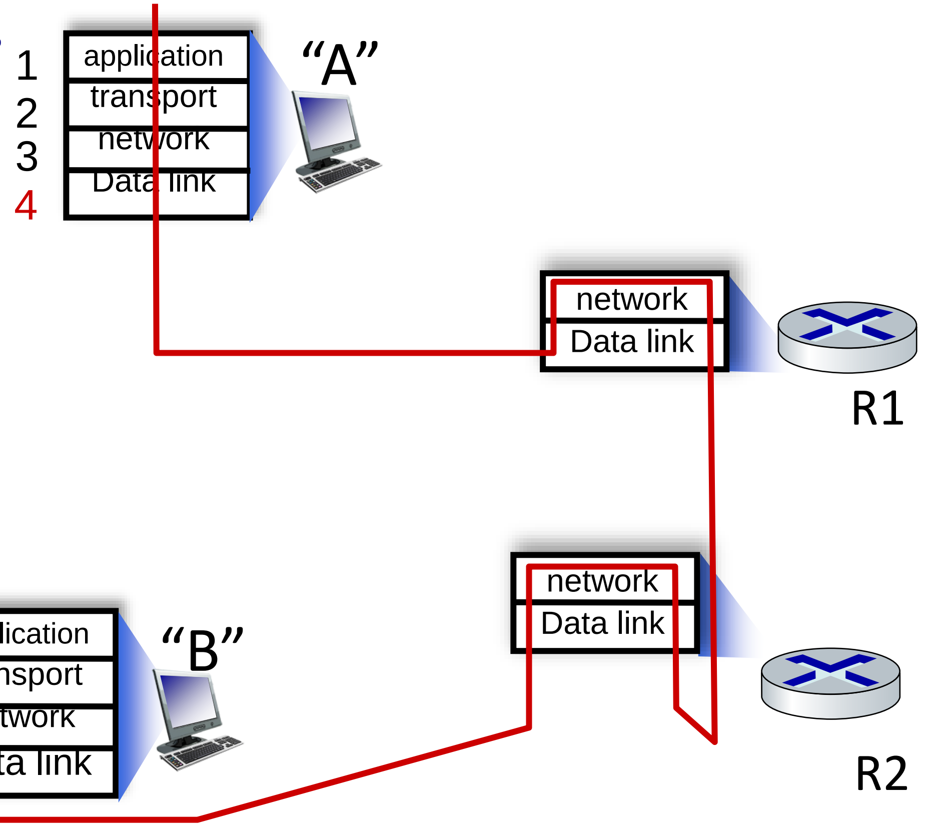
- IP requests packet to be sent to router.

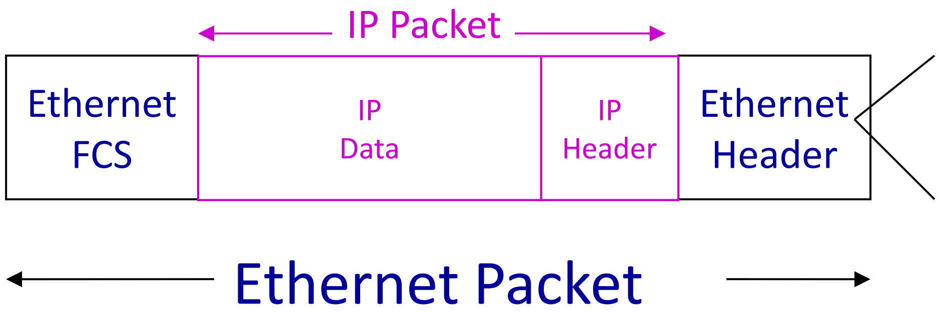


In the Sending Host – cont’d

4. Link (“MAC” or Ethernet) Protocol

- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.



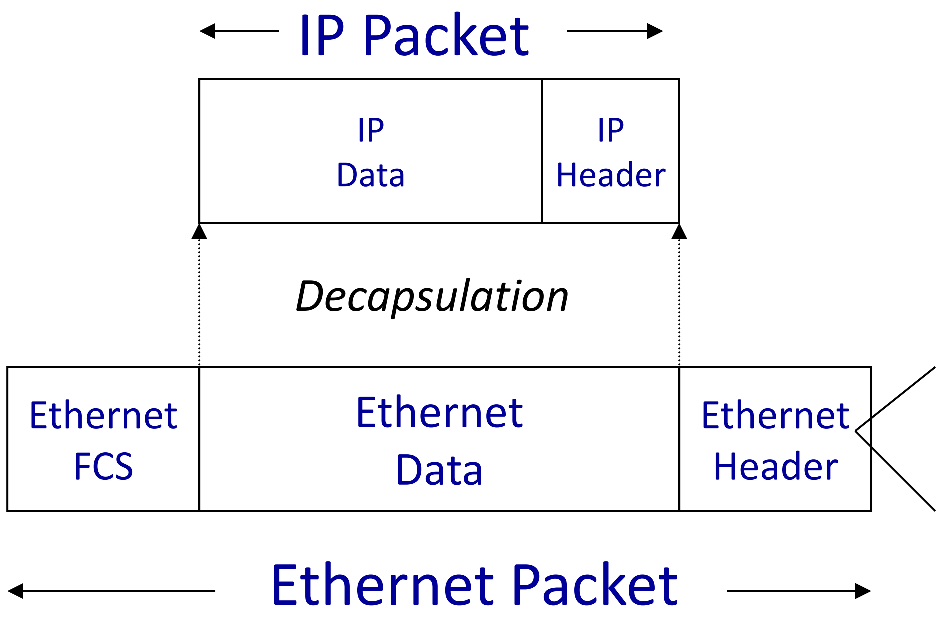


Destination Address: MAC “R1”
Source Address: MAC “A”
Protocol = IP

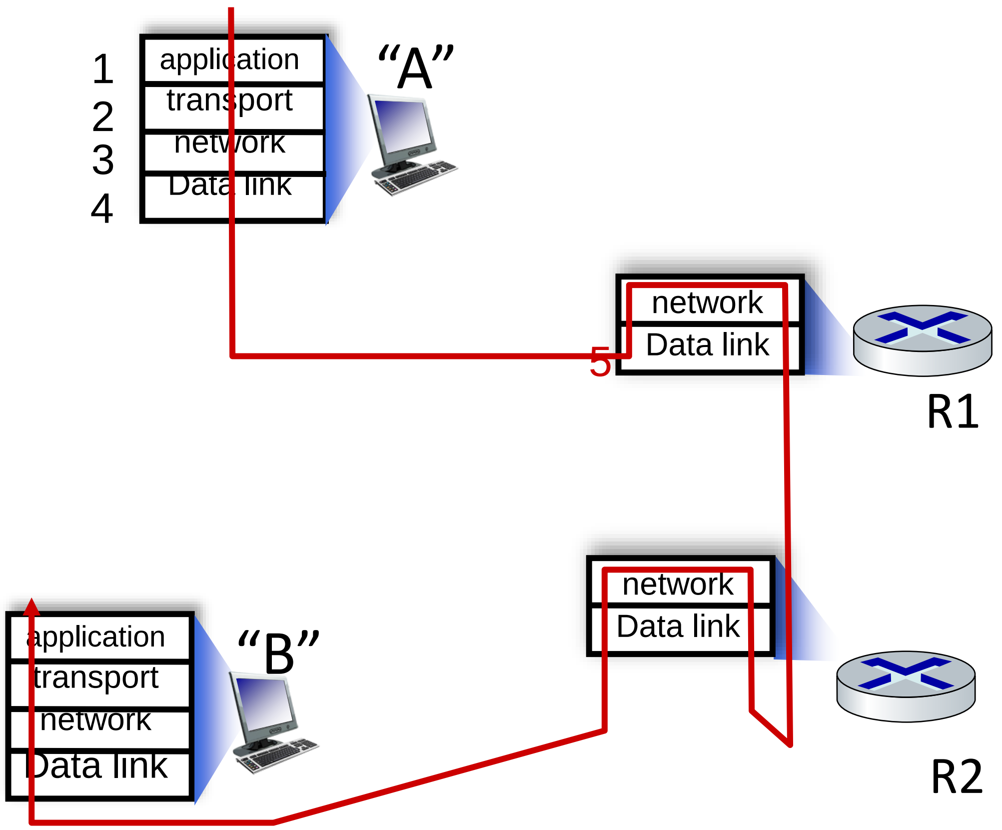
In Router R1

5. Link (“MAC” or Ethernet) Protocol

- Accept MAC frame, check MAC address and Frame Check Sequence (FCS).
- Pass data to IP Protocol.



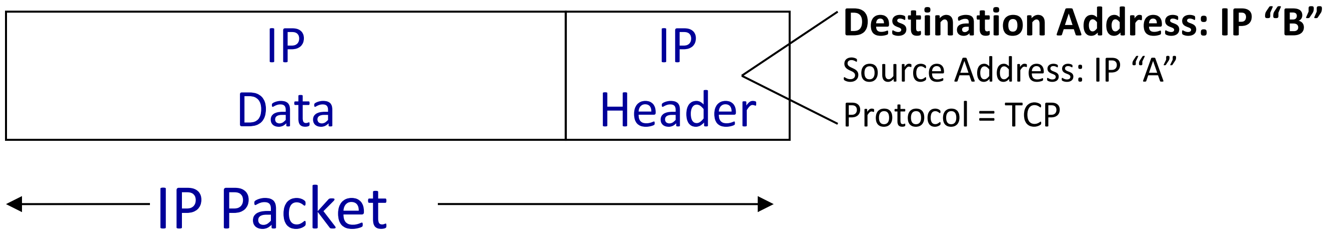
Destination Address: MAC “R1”
Source Address: MAC “A”
Protocol = IP

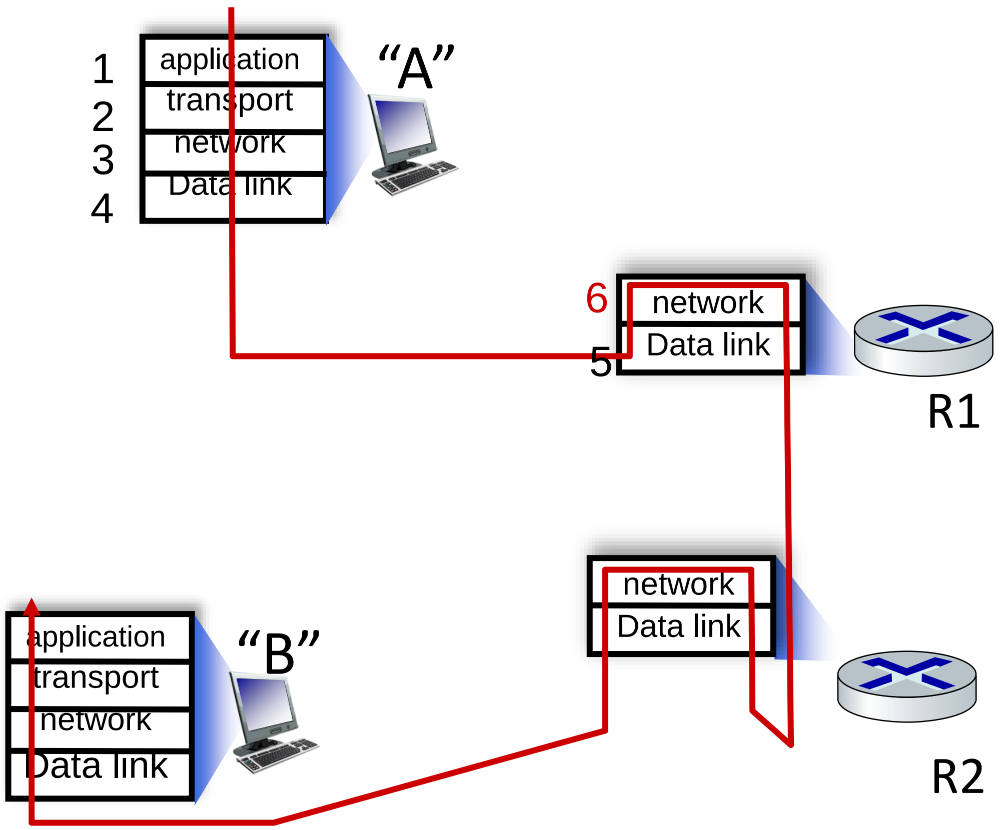


In Router R1 – cont’d

6. Internet Protocol (IP)

- Use IP destination address to decide where to send packet next (“next-hop routing”).
- Request Link Protocol to transmit packet.

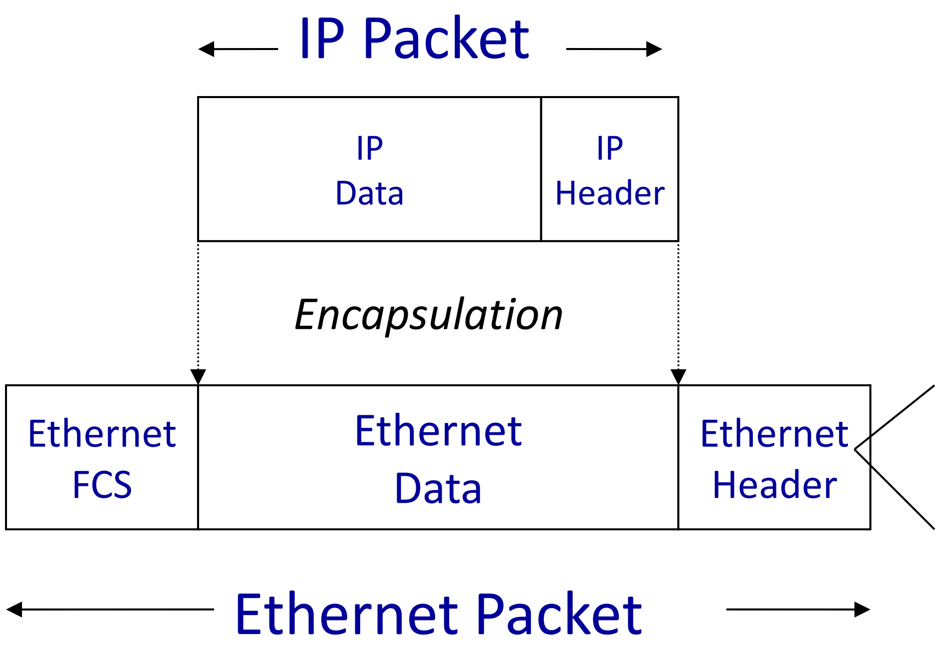




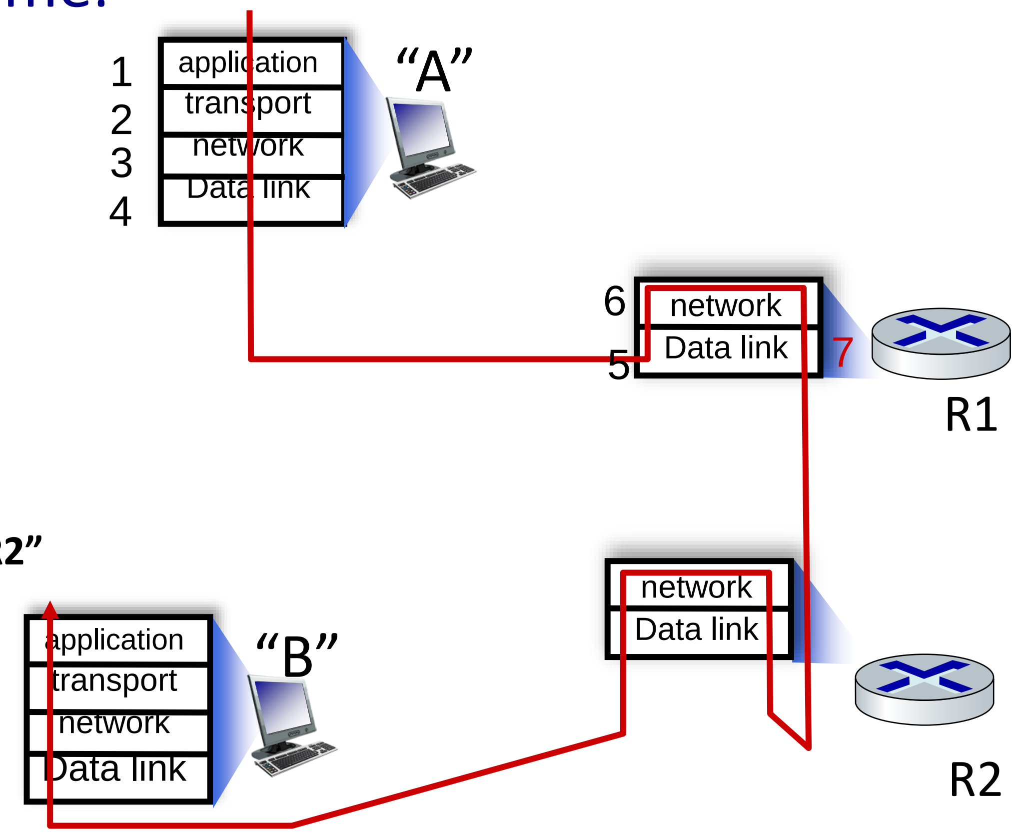
In Router R1 – cont’d

7. Link (“MAC” or Ethernet) Protocol

- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.



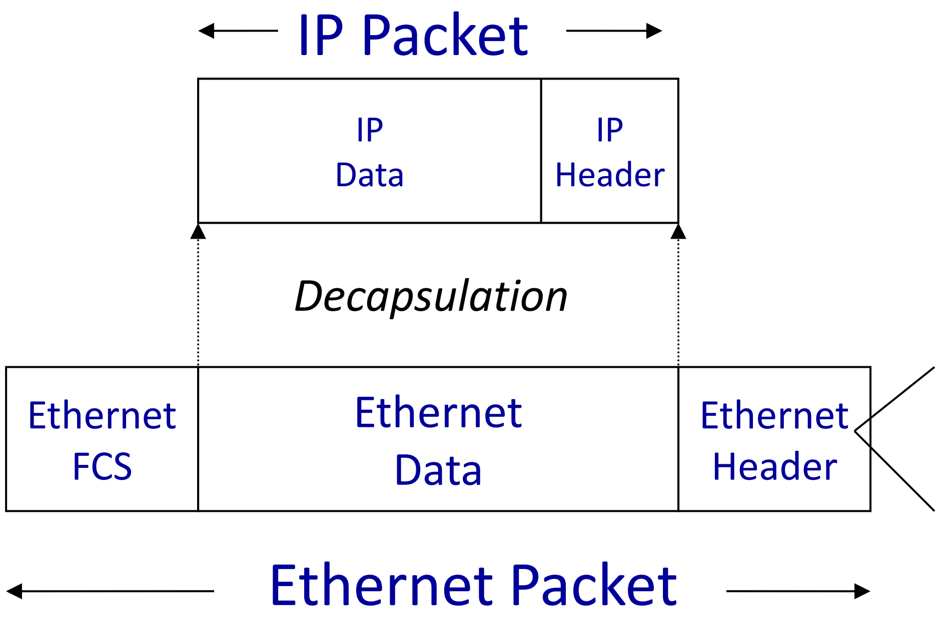
Destination Address: **MAC “R2”**
Source Address: **MAC “R1”**
Protocol = IP



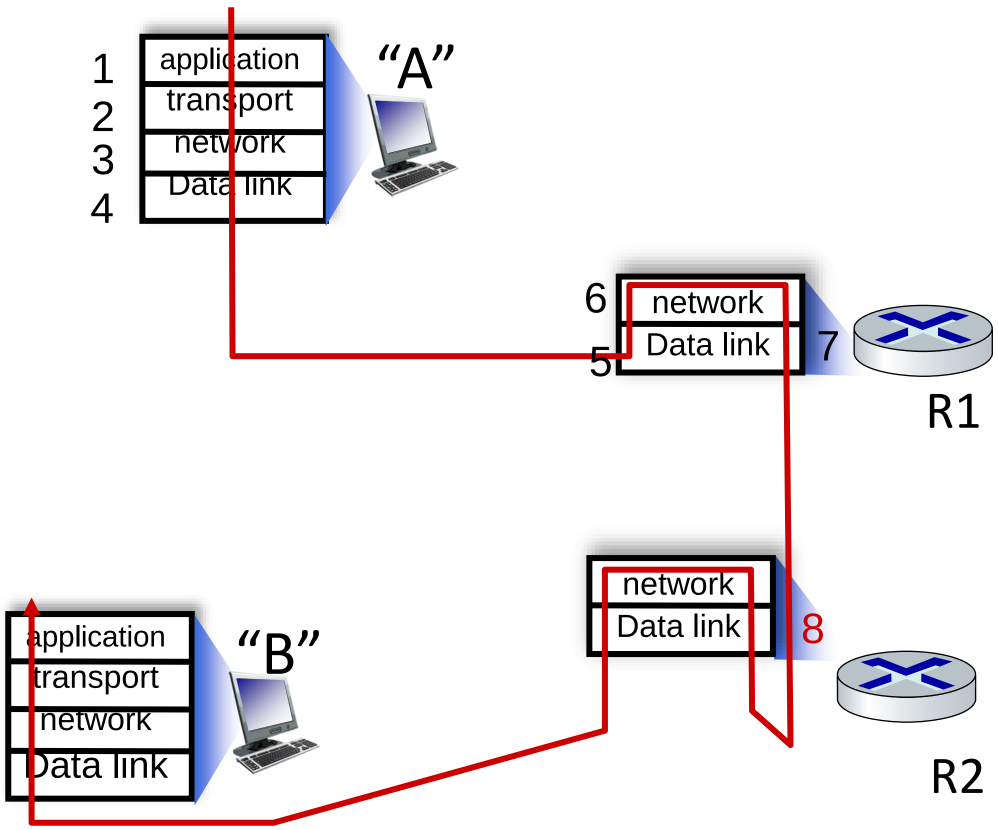
In Router R2

8. Link ("MAC" or Ethernet) Protocol

- Accept MAC frame, check MAC address and Frame Check Sequence (FCS).
- Pass data to IP Protocol.



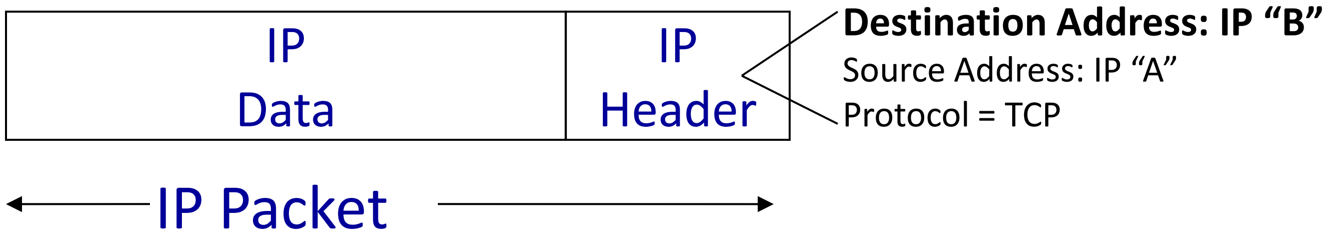
Destination Address: MAC "R2"
Source Address: MAC "R1"
Protocol = IP

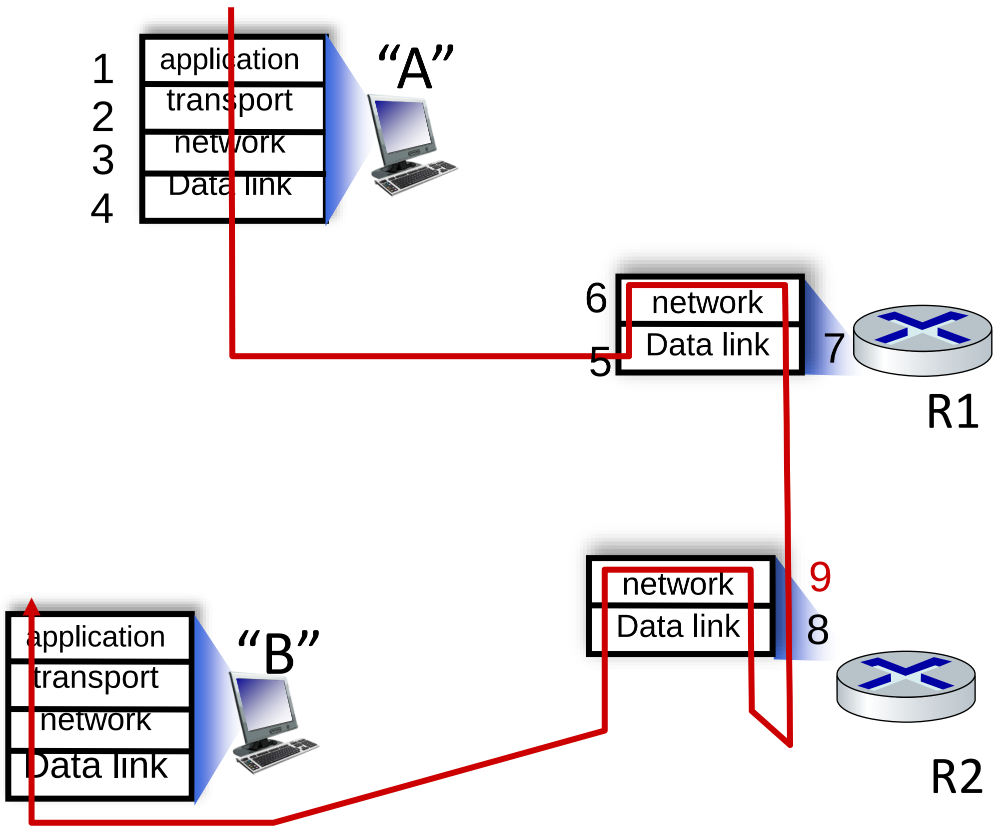


In Router R2 – cont’d

9. Internet Protocol (IP)

- Use IP destination address to decide where to send packet next (“next-hop routing”).
- Request Link Protocol to transmit packet.

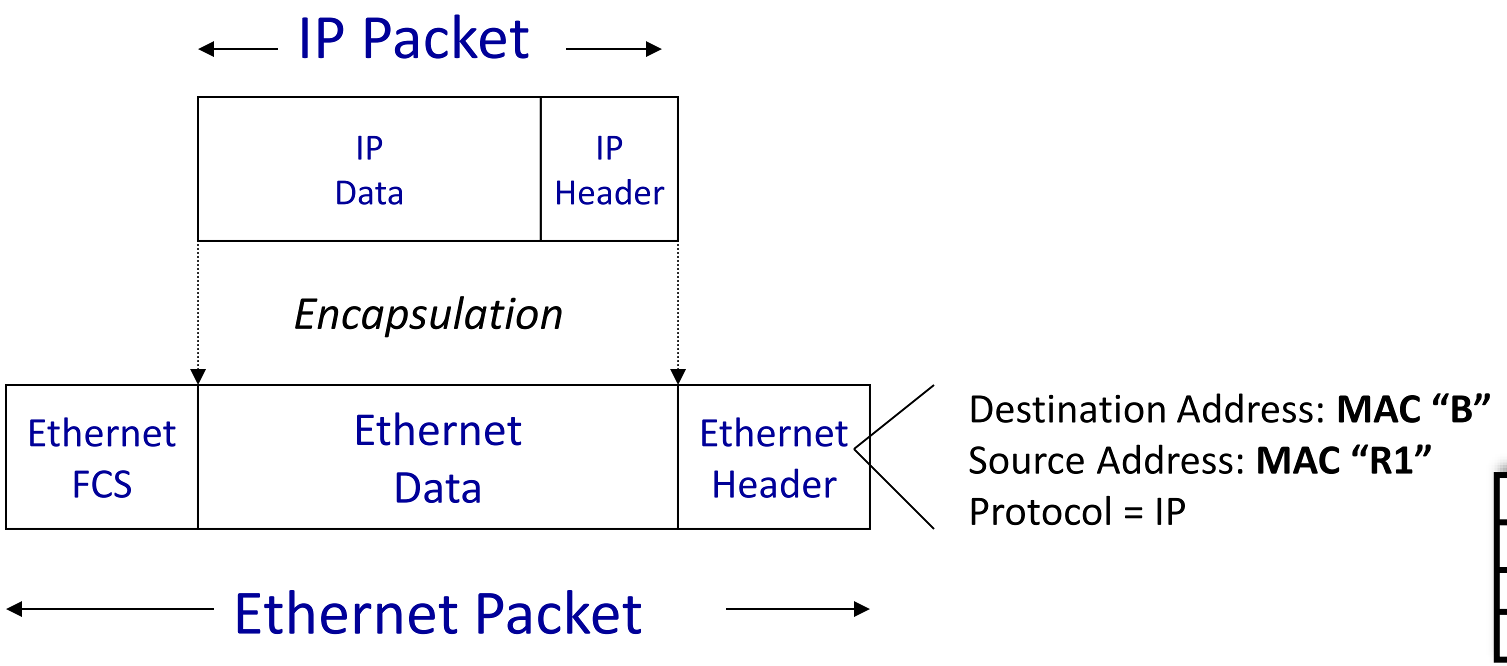


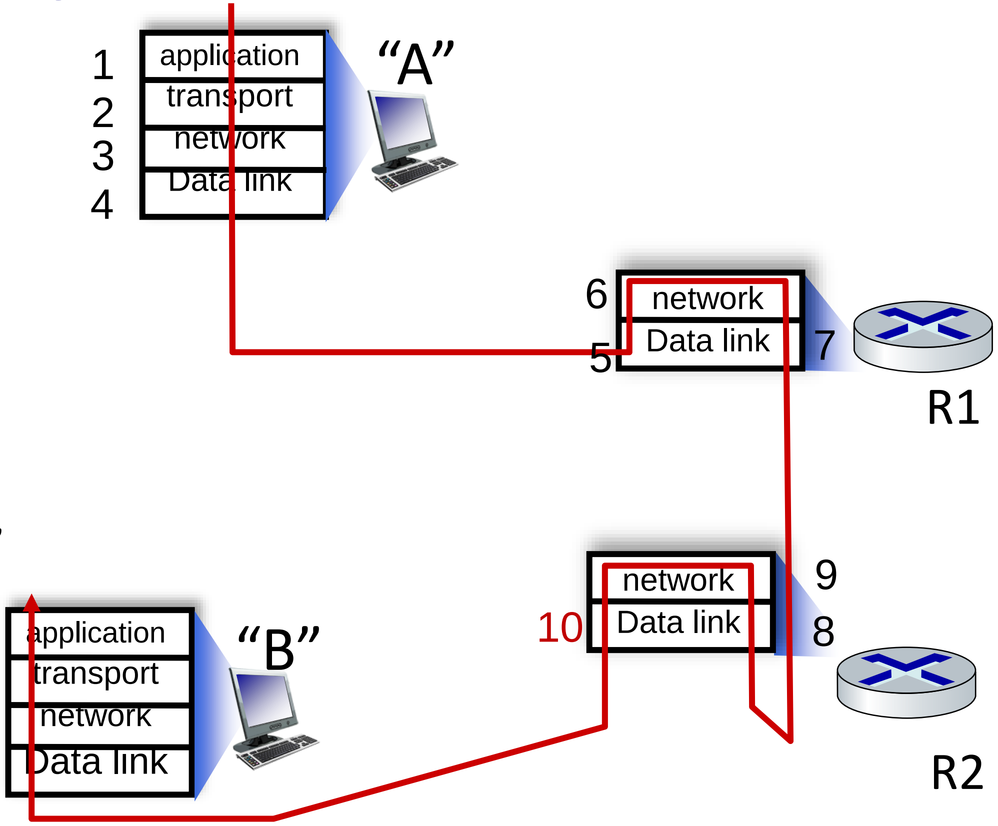


In Router R1 – cont'd

10. Link ("MAC" or Ethernet) Protocol

- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.

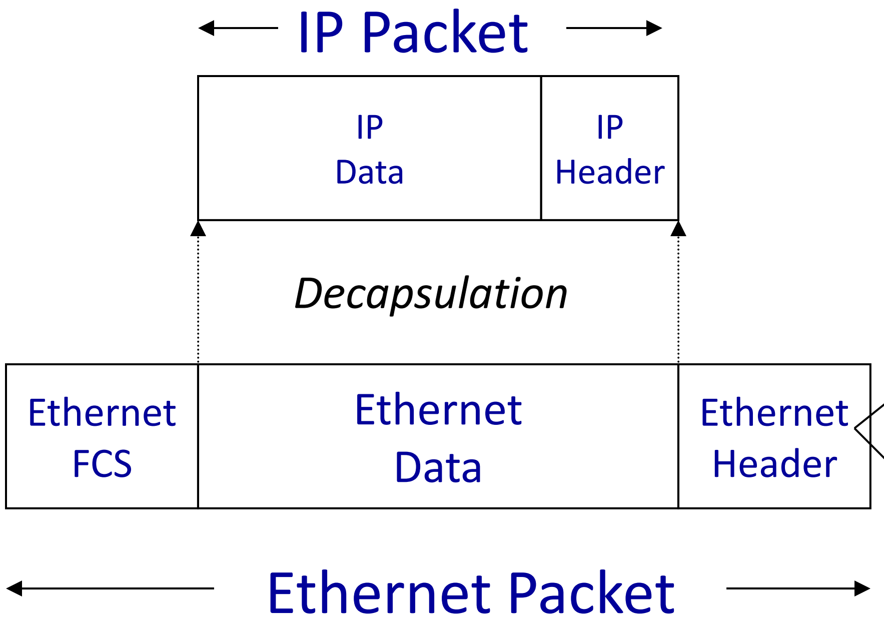




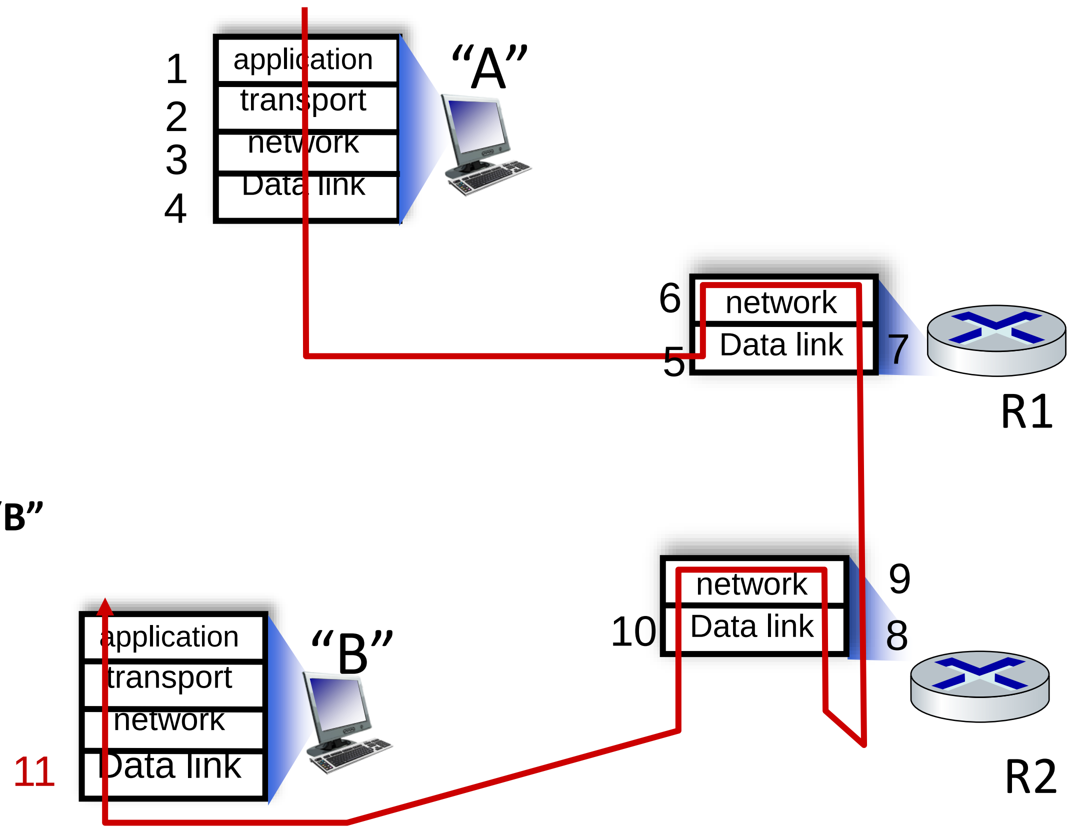
In The Receiving Host

11. Link (“MAC” or Ethernet) Protocol

- Accept MAC frame, check address and Frame Check Sequence (FCS).
- Pass data to IP Protocol.



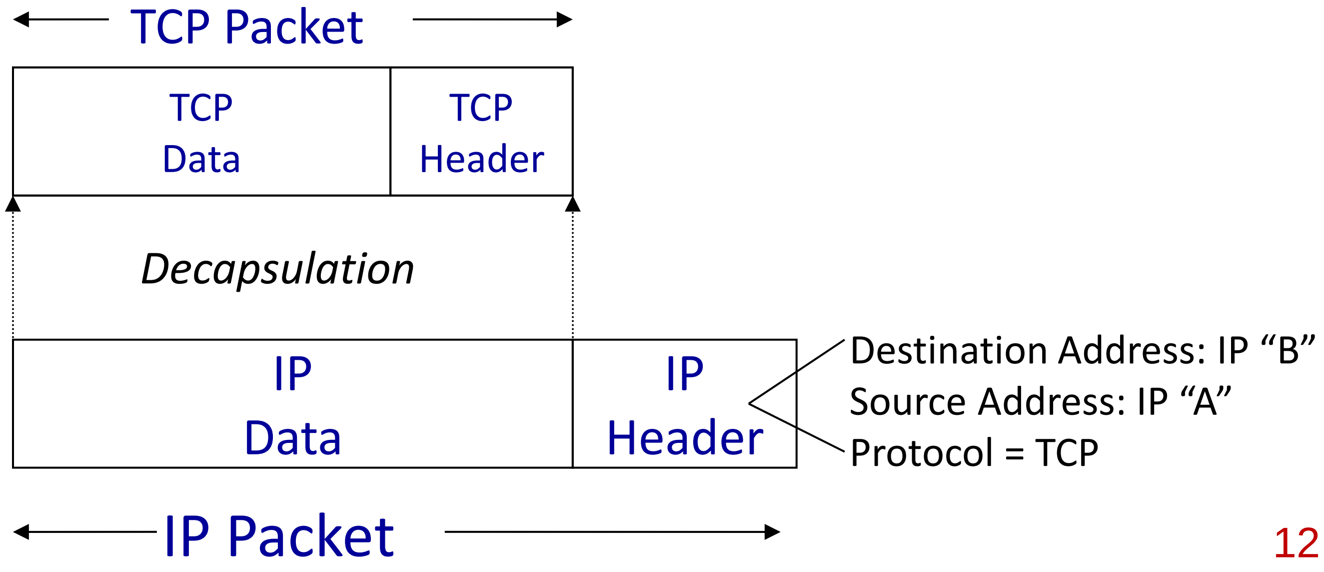
Destination Address: **MAC “B”**
Source Address: **MAC “R2”**
Protocol = IP

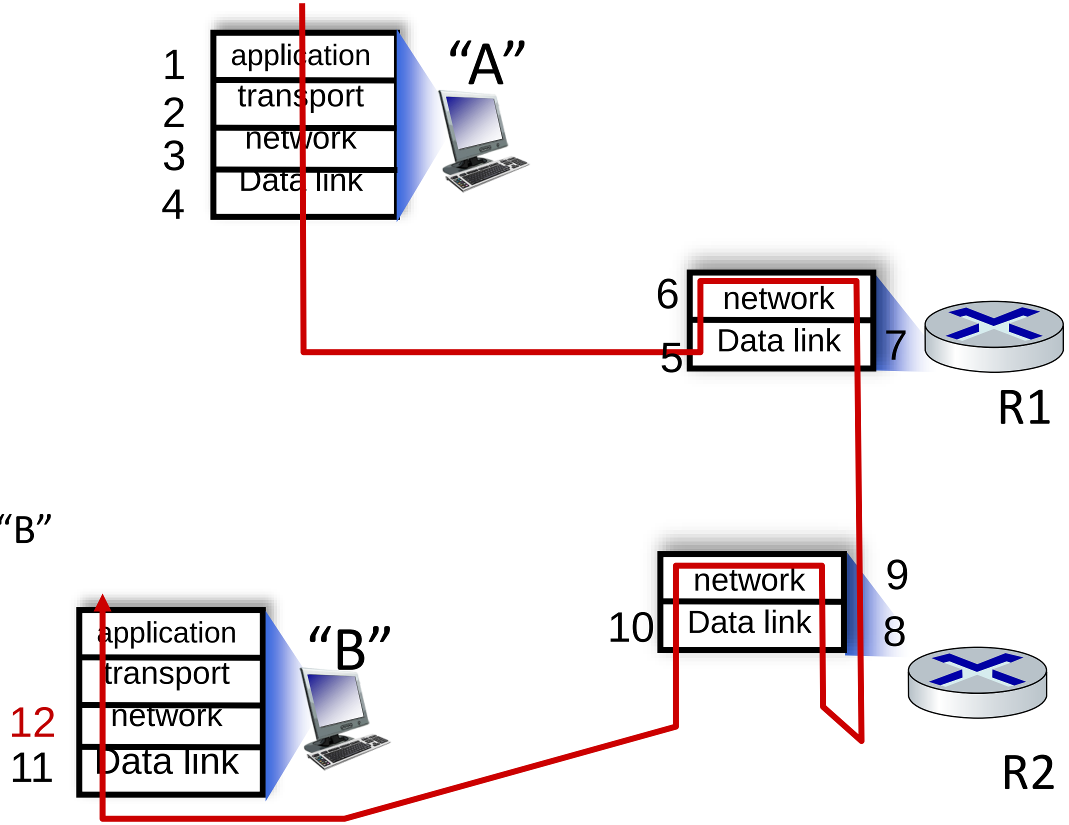


In The Receiving Host – cont’d

12. Internet Protocol (IP)

- Verify IP address.
- Extract/decapsulate TCP packet from IP packet.
- Pass TCP packet to TCP Protocol.





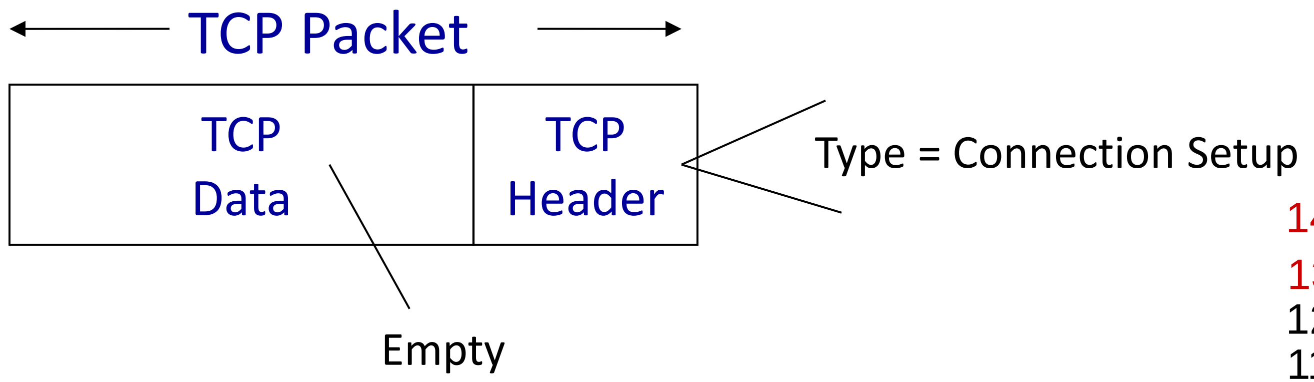
In The Receiving Host – cont’d

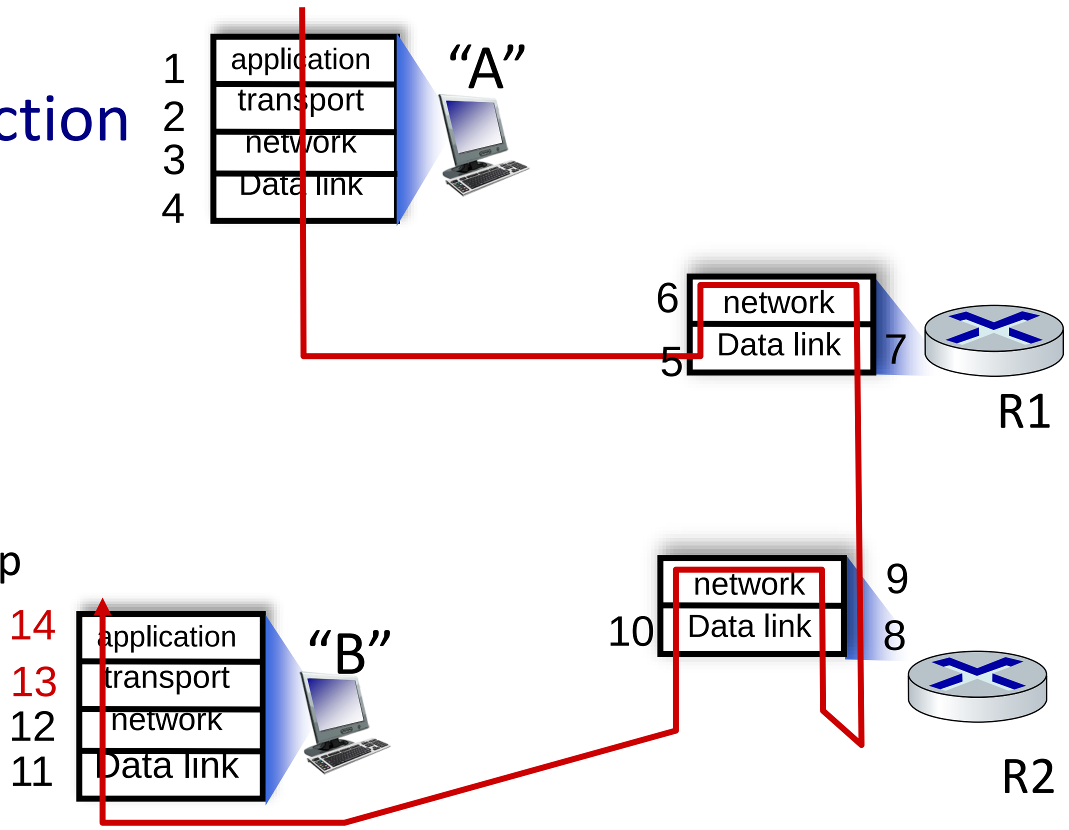
13. Transmission Control Protocol (TCP)

- Accepts TCP “Connection setup” packet
- Establishes connection by sending “Ack”.

14. Application-Programming Interface (API)

- Application receives request for TCP connection with “A”.





What is TCP, IP, IP Address, Ethernet, MAC, MAC Address, and FCS?

- Throughout this example, we saw many terminologies that you may not know.
  - What is TCP? What is a Router exactly? What is Ethernet? What is an IP address and what is a MAC address? What is MAC anyway? What is Ethernet? Why do we need to wait for “Access” to the line? What is FCS?
- Don’t worry. That’s normal. We will study each of them in details in this course.

A Day in Life of A Packet: Takeaway

- The takeaway message is:
  - A packet goes through multiple devices and layers throughout its path.
  - It will be encapsulated and decapsulated each time that it goes from one layer to another.
  - Routers only require Network layer and below
    - In future, we'll see that switches need only the data link layer.
  - End-hosts need all layers
  - As we saw, there are multiple addressing schemes, e.g., IP address and MAC address. Later in the course, we'll see why we need multiple addresses.