Chapter 1

An Overview of Networks

The undergraduate pre-requisite course on Computer Networks (ECE 4470 or its equivalent taken elsewhere) usually covers most topics from in Chapter 1, except perhaps Sections 1.7 (Congestion) and 1.13 (Firewalls).

Please read the **entire** 1st chapter during the first week of classes so as to familiarize yourself with the general concepts. Try to understand it as much as you can. If some of these topics are required as precursors to the advanced topics in this course, a primer (or detailed tutorials) will be supplied.

Quick Review of Concepts

Layered Architectures

- Layered Protocols Design
  - Why we use layers
  - What is the “Protocol Stack”?
- Reference Models for communication
  - ISO-OSI Reference Model
  - TCP/IP Reference Model
- Layer Interaction and packet encapsulation
- Network Analyzers for traffic analysis (*e.g.* `Wireshark`, `tcpdump`)

Why use layers?

Because, layering ..

- Partitions communications process into smaller parts
  - thus, simplifies design, implementation and testing

- Allows different independently designed protocols at each layer
  - higher level protocol makes “calls” for services from protocol at layer below

- Provides flexibility for evolving protocols and services at any given layer
  - without having to change other layers

Monolithic non-layered architectures are inflexible, expensive to maintain and evolve

What layered models to use?

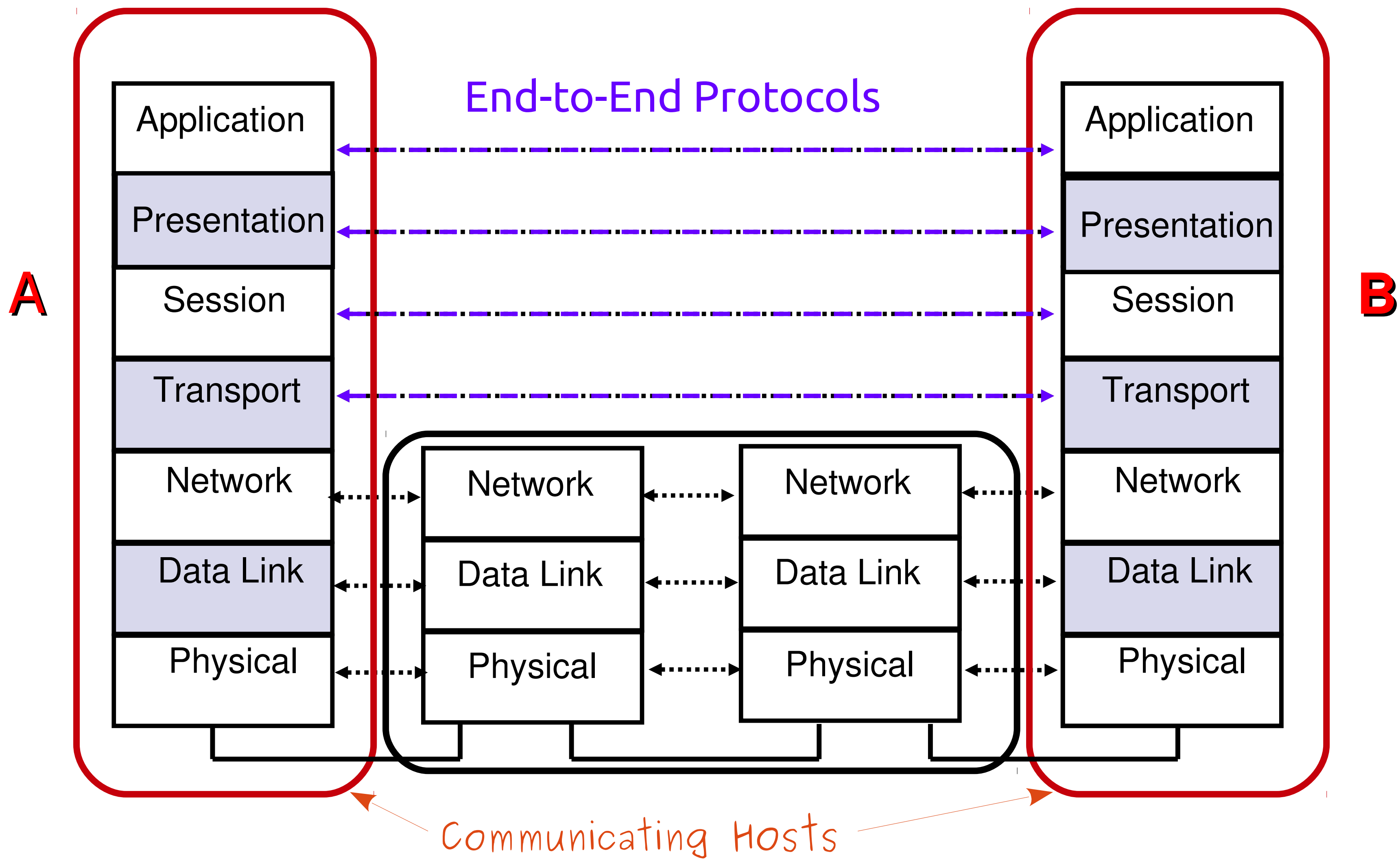
Two main models ..

1. International Standards Organization's Open Systems Interconnection (**ISO OSI**) model
   - good in theory, and used for study
   - <u>almost never used in practice</u>

2. Transmission Control Protocol over Internet Protocol (**TCP/IP**) model
   - not as neatly designed as the OSI model
   - <u>almost universally used in practice</u>

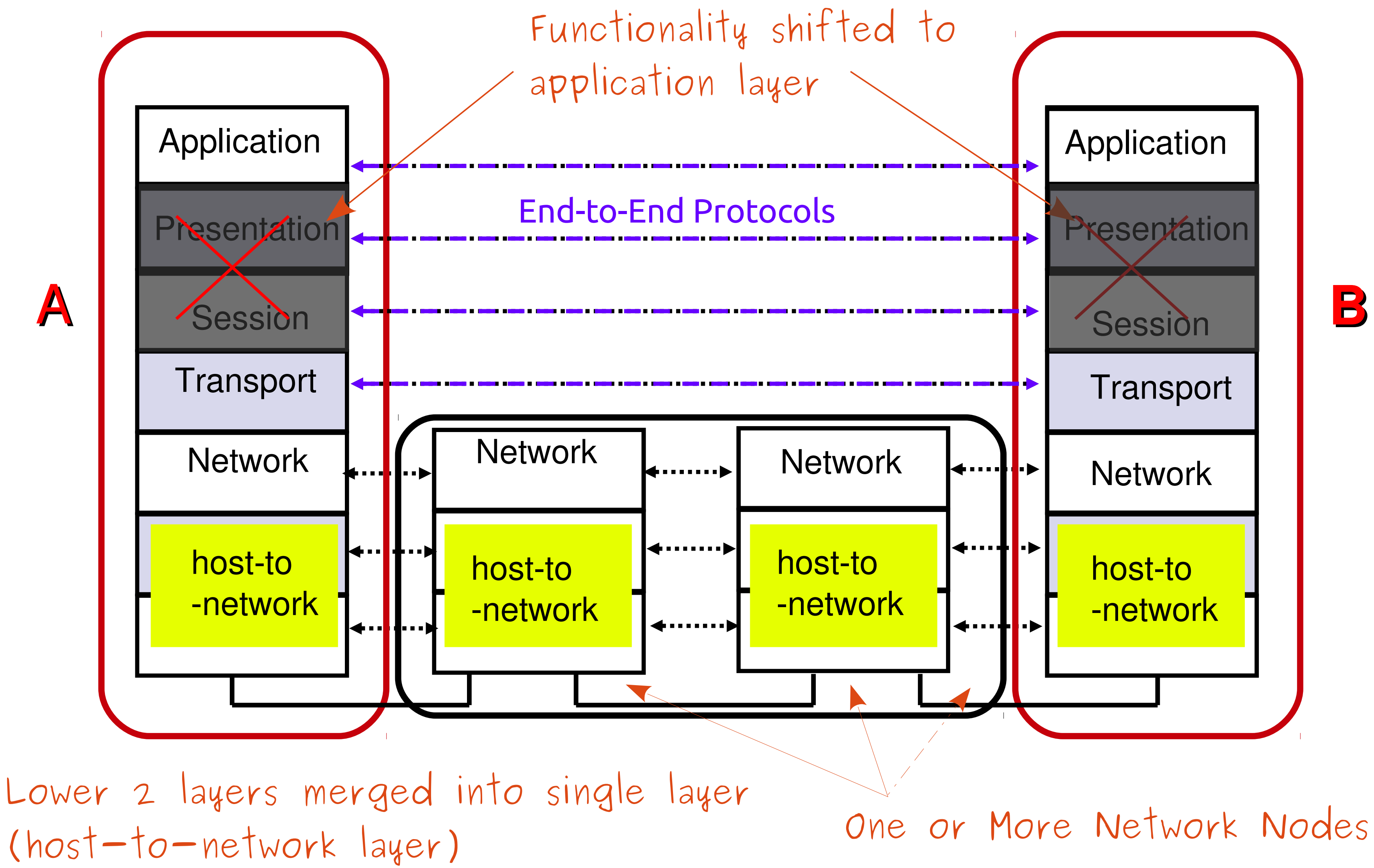
Regardless of machine architecture and internal protocol implementation, any two machines that implement the same network architectural model can *communicate* with each other.

1. The ISO OSI Model (7 layers)

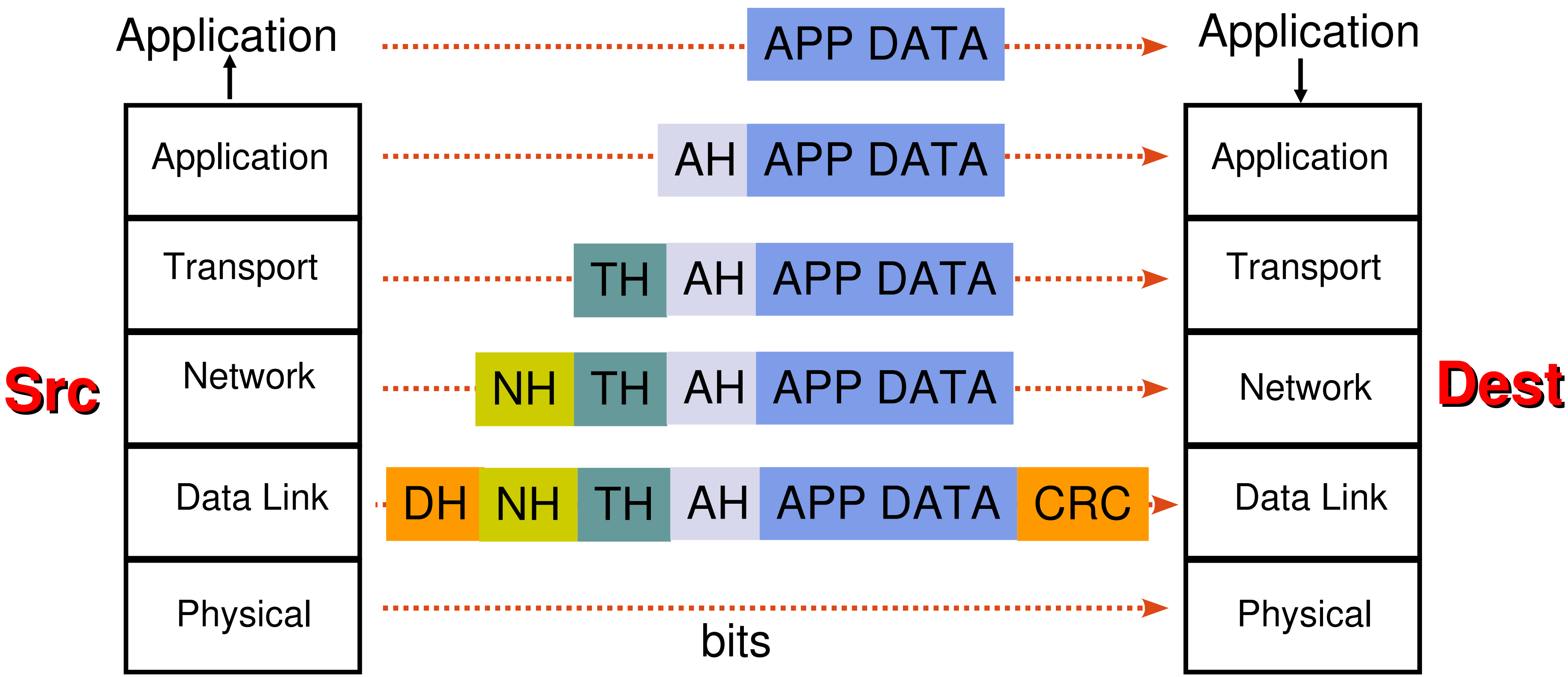


with one or more network nodes (routers) in between

2. The TCP/IP Model (4 layers)



1. Condensed OSI Model: Protocol Headers & Trailers



- Each protocol header carries addresses, sequence numbers, flag bits, length indicators, etc…
- **CRC** check bits may be appended for error detection

Information Flow in a Hypothetical 5 Layer Model: **Two Directly Connected Hosts**

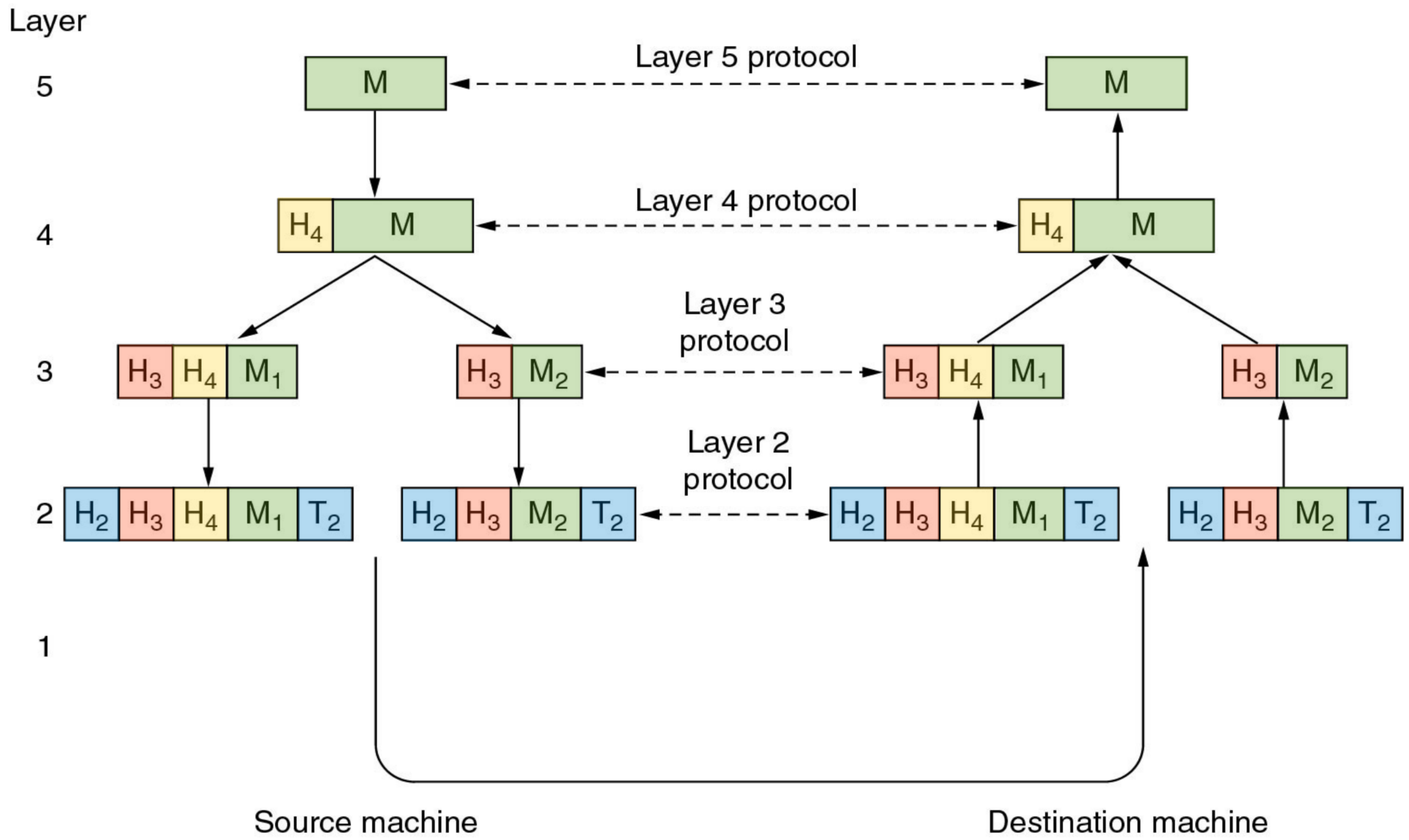


Figure: From "Computer Networks (5 ed.)" by Tanenbaum & Wetherall

Information Flow in a Hypothetical 5 Layer Model: **Hosts Connected by Network Devices**

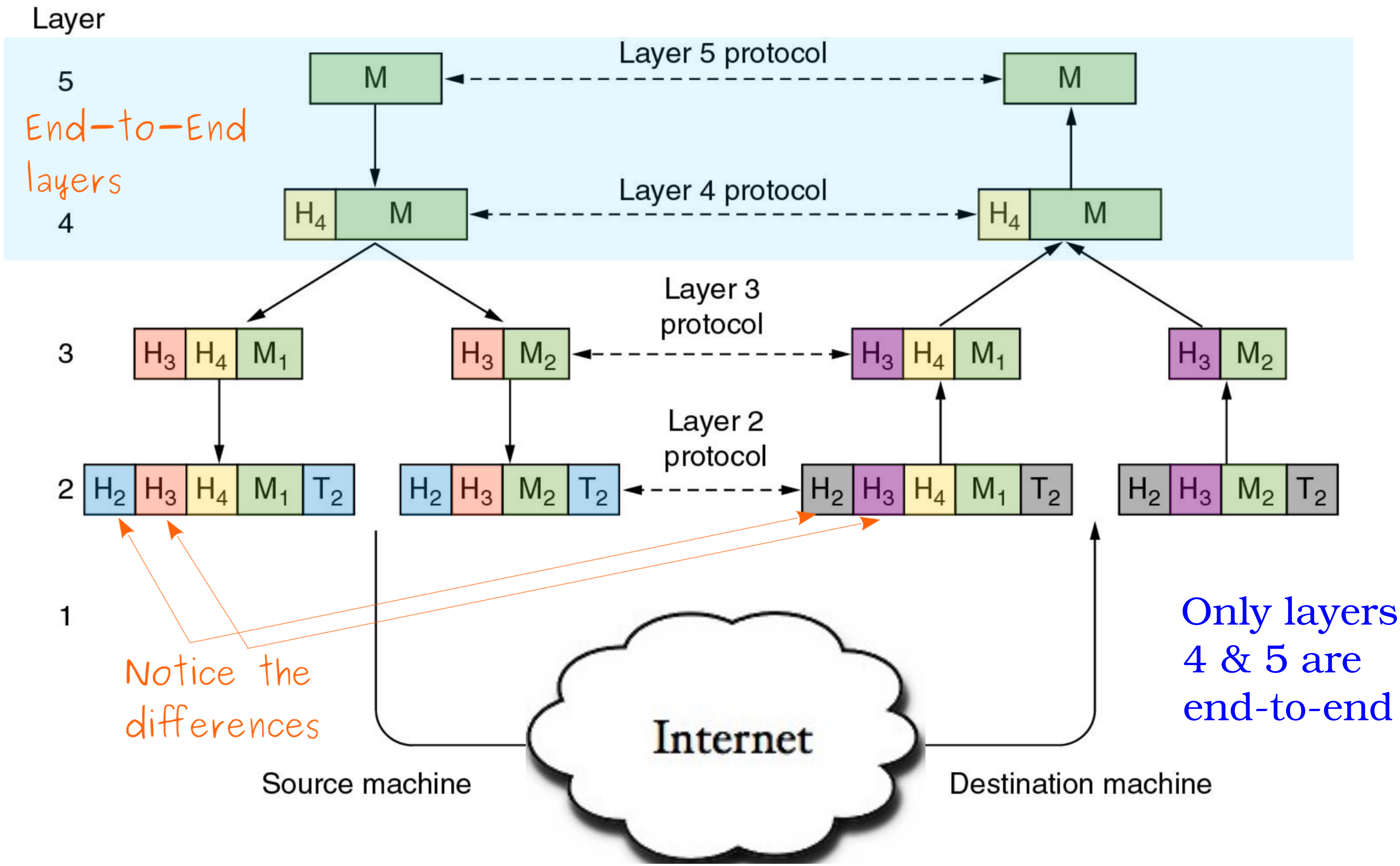


Figure: From “Computer Networks (5 ed.)” by Tanenbaum & Wetherall

Side-by-side: ISO OSI and TCP/IP Models

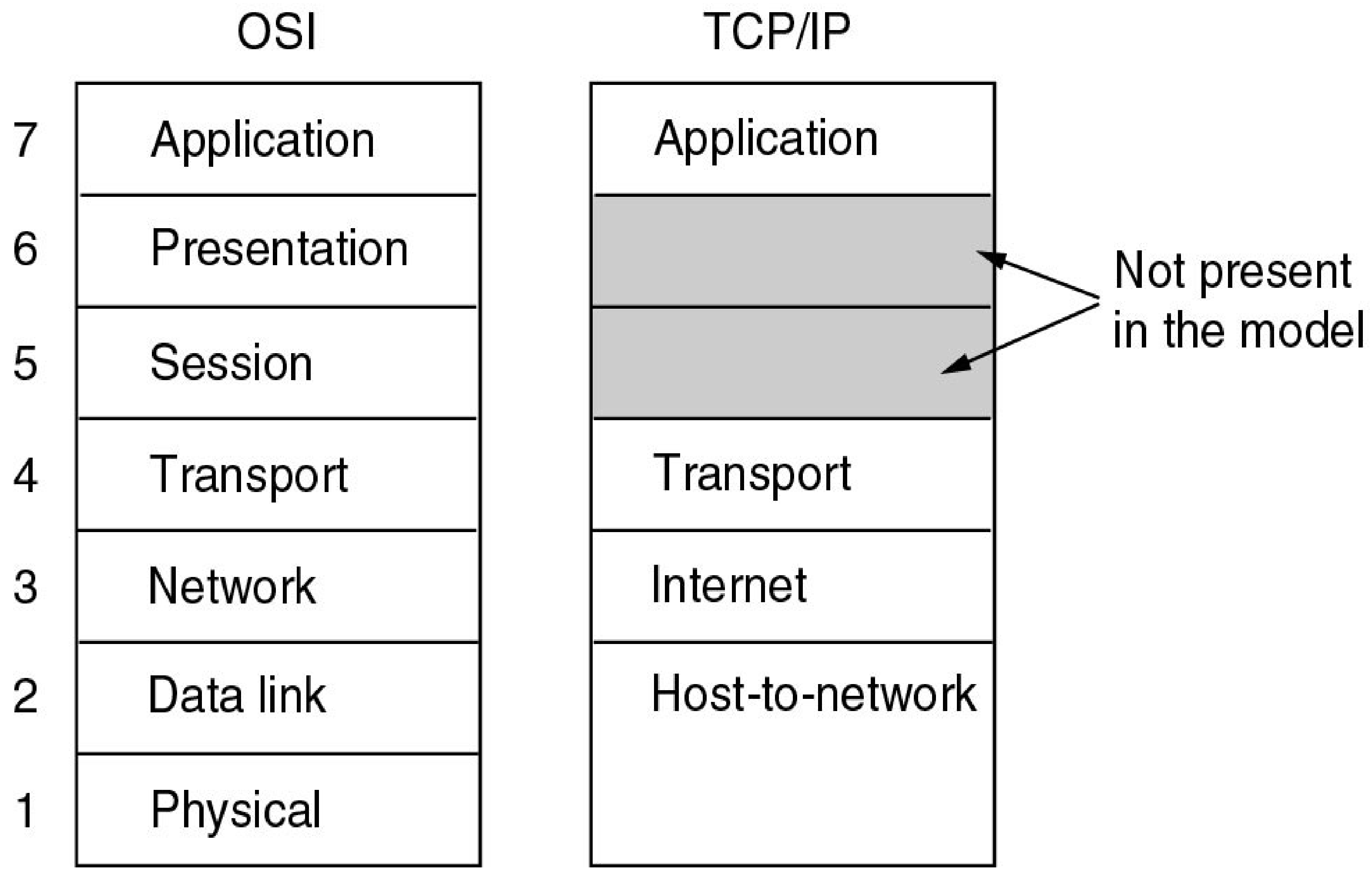
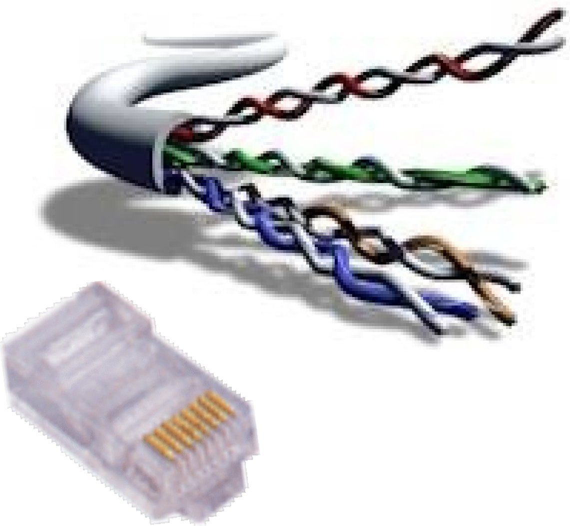


Figure: From “Computer Networks, 4th ed.,” A. Tanenbaum

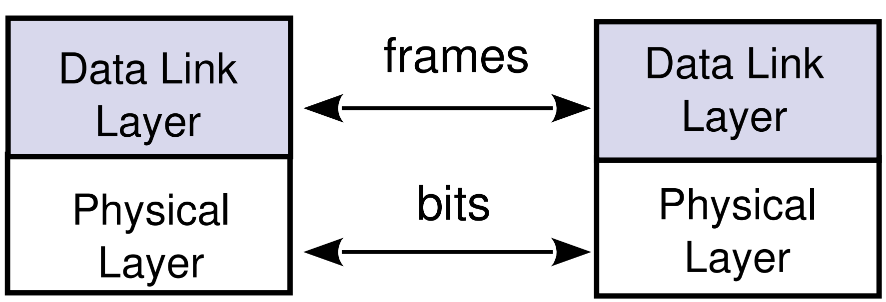
1. Condensed OSI Model: Physical Layer



- Transfers bits across link
- Definition & specification of the physical aspects of a communications link
  - Mechanical: cable, plugs, pins...
  - Electrical/optical: modulation, signal strength, voltage levels, bit times, …
  - functional/procedural: how to activate, maintain, and deactivate physical links…
- Ethernet, DSL, cable modem, telephone modems…
- Twisted-pair cable, coaxial cable optical fiber, radio, infrared, …

1. Condensed OSI Model: Data Link Layer

- Transfers *frames* across *direct* connections
- Groups bits into frames
- Detects bit errors; Retransmit errored frames
- Activates, maintains, deactivates data link connections
- Medium access control for local area networks
  (done within the MAC sublayer in the TCP/IP model)
- Flow control

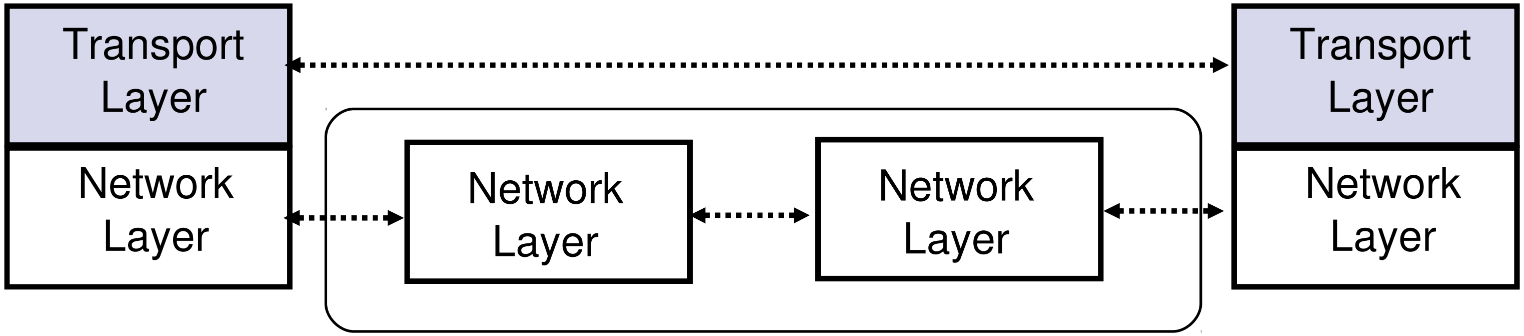


1. Condensed OSI Model: Network Layer

- Transfers *packets* across multiple links and/or networks
- Addressing must scale to large networks
  - TCP/IP uses IP addresses
- Two operations of main interest: routing, forwarding
  - ***Routing***: Nodes *co-operatively* execute routing algorithm to determine paths across network
  - ***Forwarding***: transfer packet to next hop
- Congestion control to deal with traffic surges
  - In TCP/IP model, this is done at transport layer
- Connection-less or connection-oriented. If connection-oriented, do setup, maintenance, teardown
  - IP layer in TCP/IP allows only connectionless routing

1. Condensed OSI Model: Transport Layer

- Transfers application data end-to-end between machines
- Only reliable, connection-oriented stream transfer
  - ! <u>In **TCP/IP** model:</u>
    - TCP ⇒ reliable & connection-oriented stream
    - UDP ⇒ unreliable, connection-less, quick-and-simple single-block transfer
- Port numbers enable multiplexing
- Message segmentation and reassembly
- Connection setup, maintenance, and release



Condensed OSI and TCP/IP Models: Application Layer

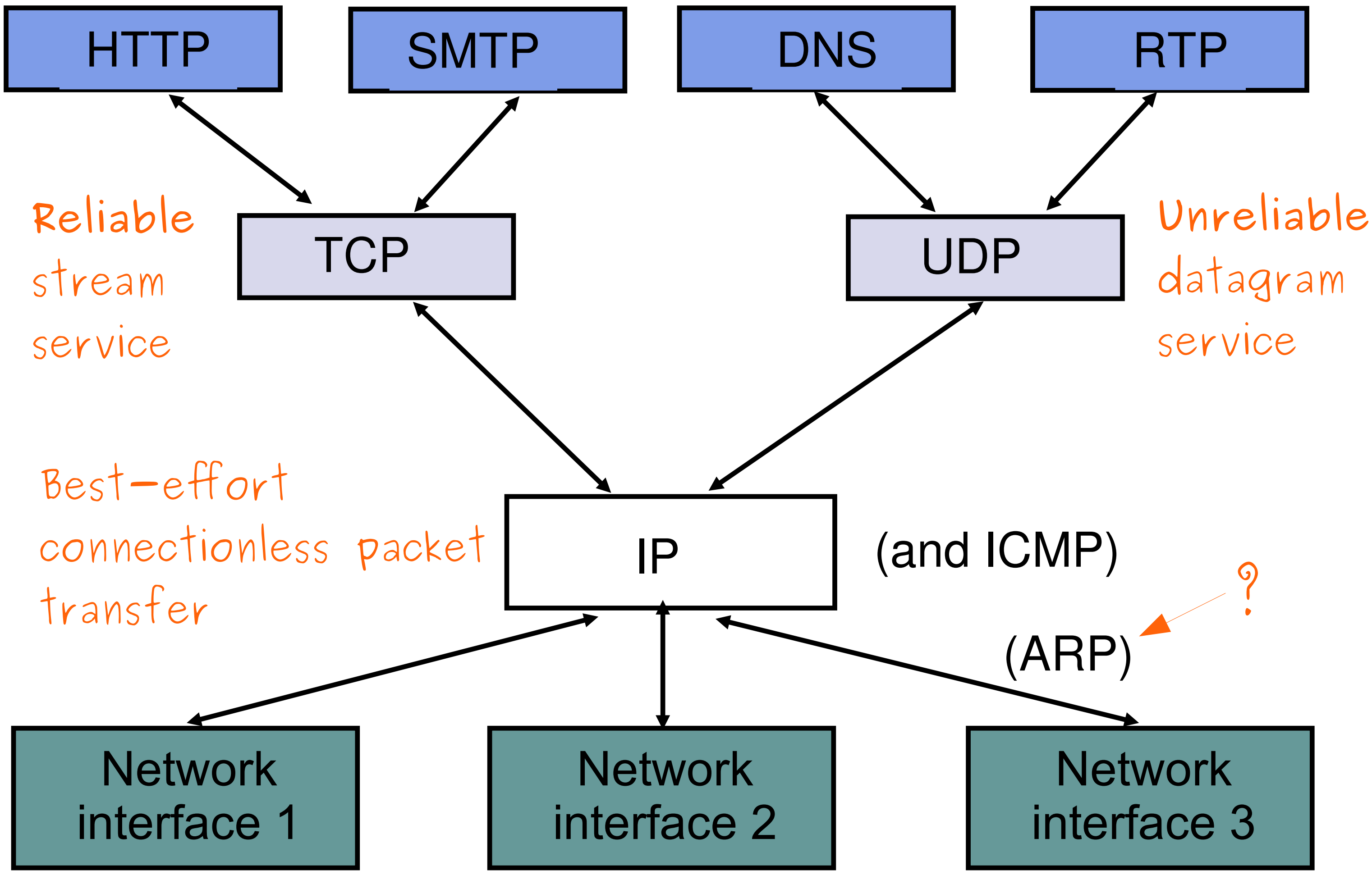
- Example application TCP/IP application protocols:
  - email (pop, imap, smtp)
  - web service (http)
  - name resolution (dns)
  - real-time transport (rtp/rtcp), *etc*.
- In TCP/IP, this layer implements functionality of application + presentation + session layers

Complete the following yourself:

- presentation functions are:

- session functions are:

2. TCP/IP Model: Protocol Suite



2. TCP/IP Model: Naming & Addressing

Internet (IP) Names

- Unique name for each host
  - Location-independent
  - Easy to memorize
  - Two-part Name
    `hostname.domain_name`
    for example:
    `acadia.villanova.edu`

- **User Addresses**
`jdoe@rhea.villanova.edu`

DNS resolves IP name to IP address

Internet (IPv4) Addresses

- Globally unique *logical* 32 bit IP address for each host*
- Separate IP address for each physical network connection
- Routing decision based on destination IP address
- IP address has two parts:
  - *netid* and *hostid*
  - *netid* unique
  - *netid* facilitates routing
- Dotted Decimal Notation:
  w.x.y.z
  e.g. 153.104.61.131
- User Address Specification
  e.g. jdoe@153.104.61.131

2. TCP/IP Model: Naming & Addressing (contd.)

- **Hardware Addresses** (for Ethernet, IEEE 802.11, *etc.*)
  - needed for hop-to-hop packet deliveries
- IP address ↔ physical address mapping needed
- *Example*: Ethernet uses 48-bit addresses
  - Each Ethernet network interface card (NIC) has globally unique physical address ➞ Medium Access Control (MAC) address
  - First 3 bytes identify NIC manufacturer; last 3 bytes are serial number
  - 00:90:27:96:a8:07 *12 hex digits*

    Intel

TCP/IP : How Layers/Protocols Work Together

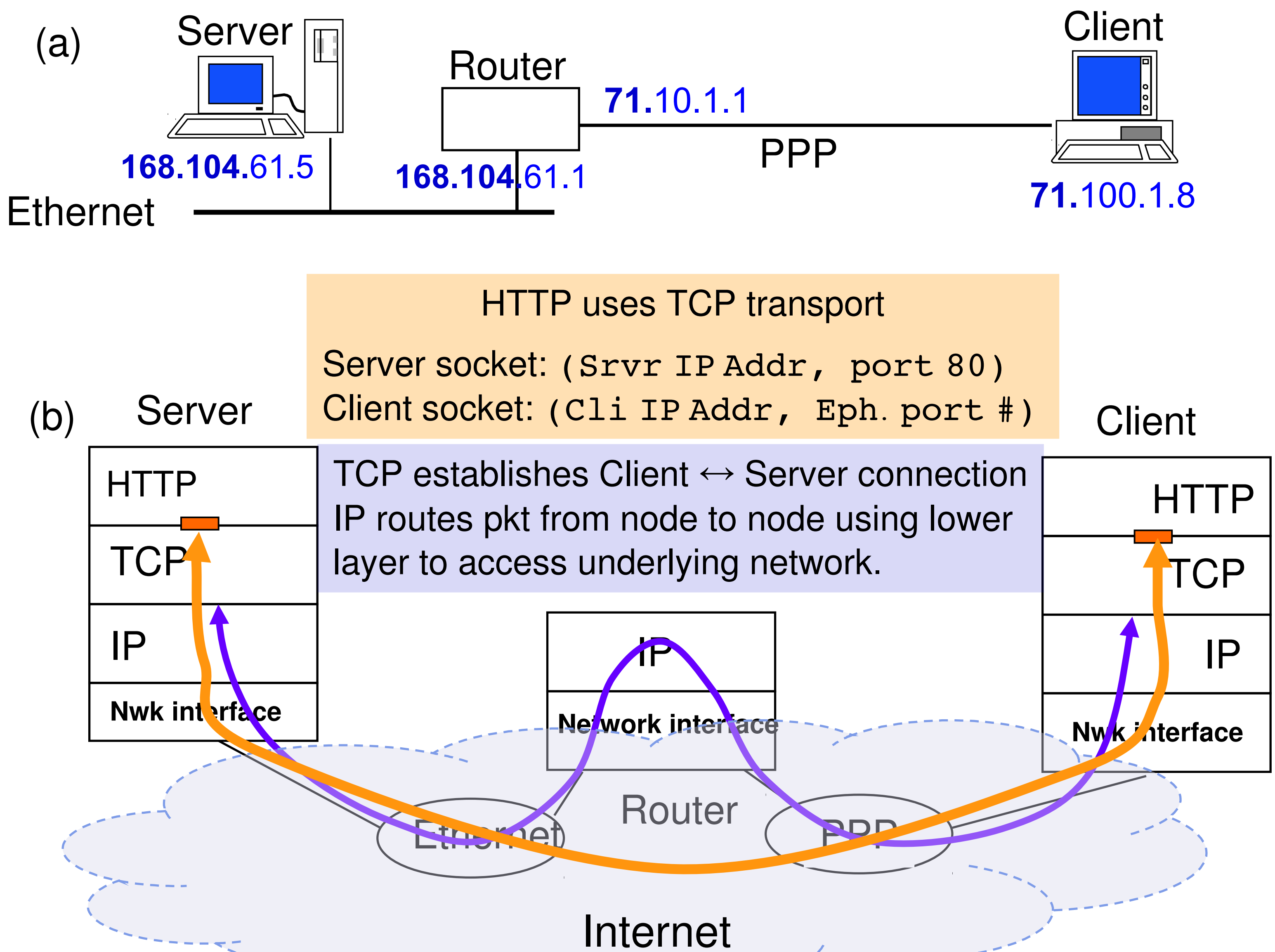


Figure: From "Comunication Networks (2 ed.)" by Leon-Garcia & Widjaja

TCP/IP : How A Packet Is Constructed

TCP header has src & dest port nos

IP header has src and dest IP addresses + transport protocol *type*

Ethernet header has src & dest MAC addresses; network protocol *type*

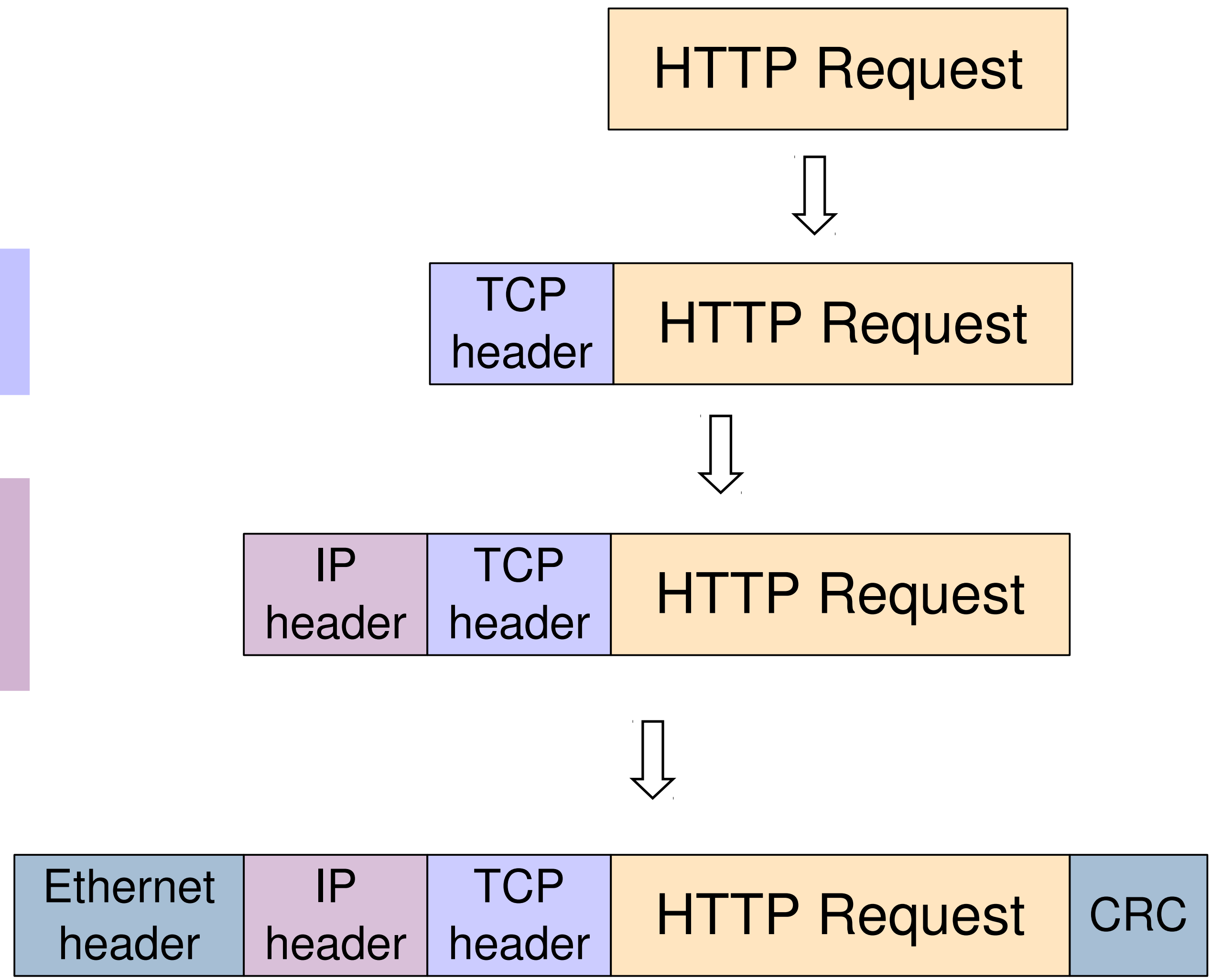


Figure: Packet **encapsulation** as it traverses protocol layers (in transmission)

TCP/IP : Encapsulation of Packets

Why encapsulate packet from one level in packet from another?

- Because this promotes layer differentiation; keeps functionality intact

Do we encapsulate packet from one level in another packet from <u>*same*</u> level?

- **Yes**. For example, we use *IP-in-IP packet encapsulation* for:
  1. VPN tunneling (encrypted IP packet is payload in another unencrypted IP packet)
  2. Forwarding packets to roaming mobile hosts (in Mobile IP protocol)
  3. Gradual transitioning from IPv4 to IPv6 protocol (*see next slide*)

Why is IP-in-IP encapsulation also called ***tunneling***?

Migration from IPv4 to IPv6

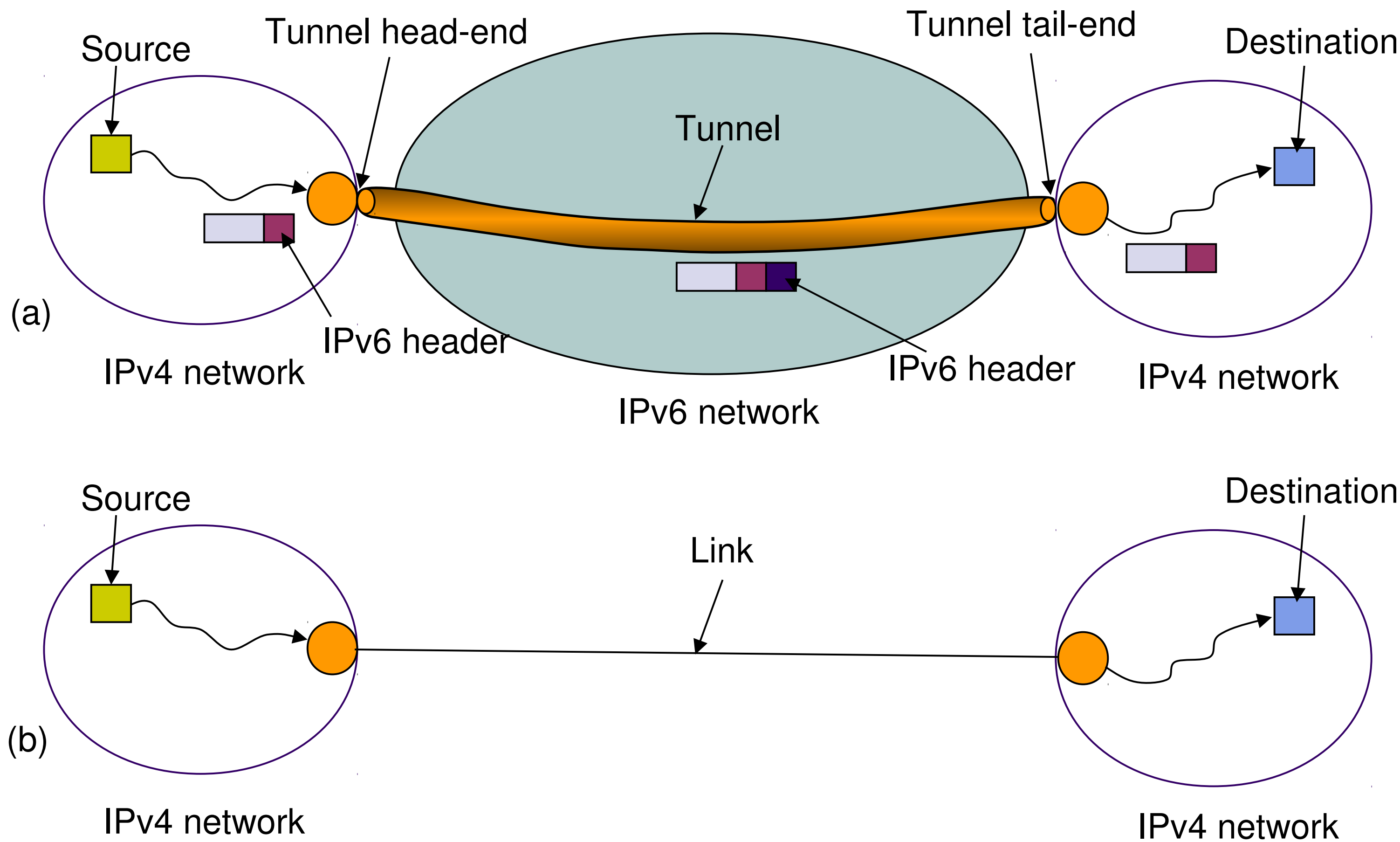


Figure: Tunneling (a) IPv4 pkt over IPv6 tunnel (b) IPv6 virtual topology
From "Communication Networks (2 ed.)" by Leon-Garcia & Widjaja

See current state of ISP-level IPv6 deployment **here**

Migration from IPv4 to IPv6 (contd.)

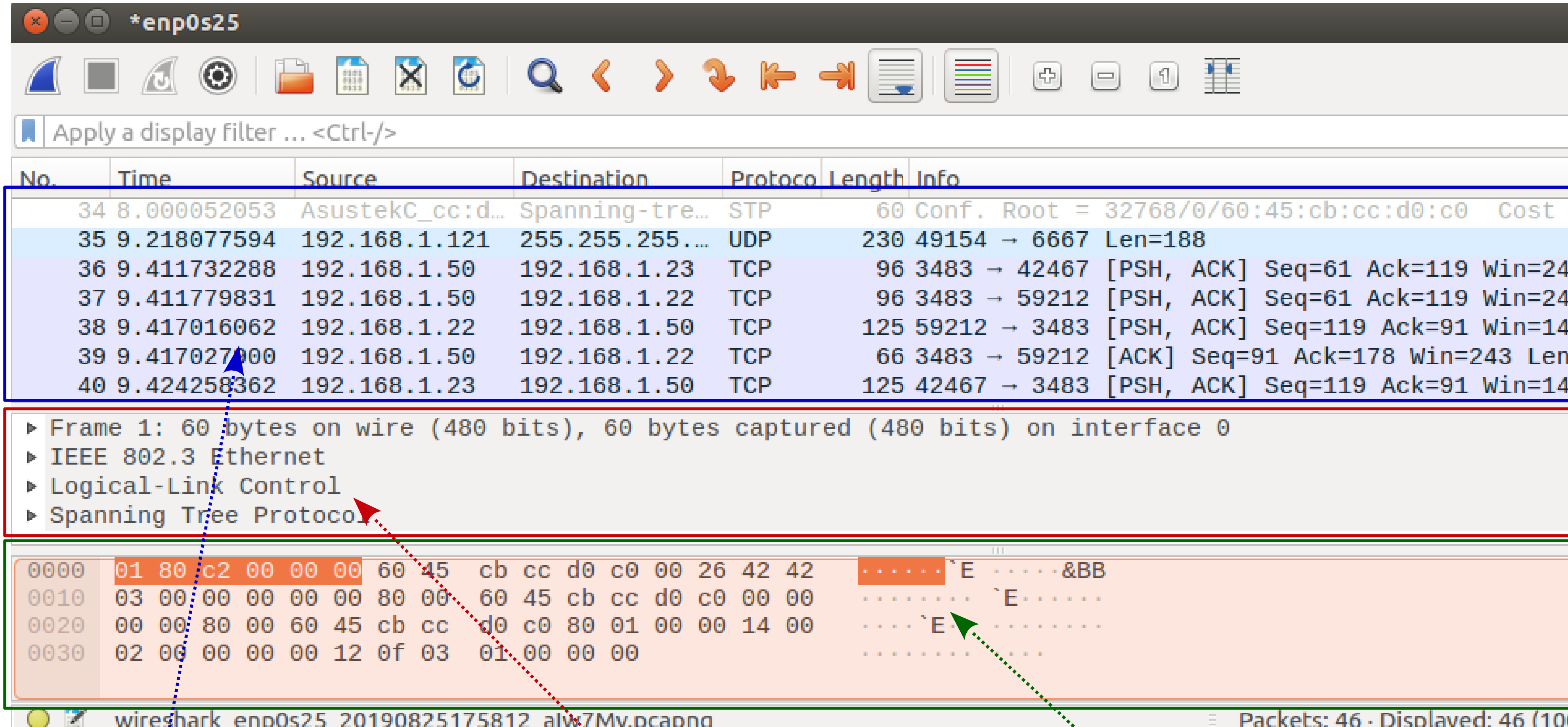
Gradual transition from IPv4 to IPv6 can be effected as follows:

- Use Dual IP stacks: routers run IPv4 & IPv6
  - `type` field in IP header ⇨ IP version
- IPv4 *islands* can tunnel across IPv6 networks (*see prev. slide*)
  - Encapsulate IPv4 packet user packet inside IPv6 packet
  - Tunnel endpoint: src host, intermediate router, or dest host
  - Can have multiple tunnels between 2 endpoints
  - A tunnel can provide forwarding service to another tunnel (*i.e*. a tunnel through a tunnel)

The *Wireshark* Network Analysis Tool

- User clicks on http://www.nytimes.com/
- *Wireshark* captures all frames observed by its Ethernet NIC
- Frames on wire (or air) and their contents can be examined down to individual bytes
- Read section 1.14 to familiarize yourself with other network utilities; download and try them out yourself.

Wireshark Display



Info about captured frame or packet

Protocol headers for frame selected in top pane

Data contained in selected frame

Summary

- Encapsulation is key to layering
- IP allows transfer of packets across diverse networks
- TCP & UDP provide universal transport services across Internet; applications use their services
- **Assigned home reading:**
  - Chapter 1 +
  - how the ARP (7.9), DHCP (7.10), DNS (7.8) protocols work
  - the sliding window protocol (undergrad notes)
  - error detection schemes – CRC, Internet Checksum (Start of 5.4 to end of 5.4.1)
  - the role of routers, and routing tables (undergrad notes)
  - Ethernet Switches and Bridges (2.4, 2.5)

Topics in red are required reading. Topics in black are for completeness