



# CS 348

## Computer Networks

### Lec 4

Spring 2020 IIT Goa

Course Instructor: Dr. Neha Karanjkar

Disclaimer: These slides are adapted from Computer Networking: A Top-down Approach by Kurose & Ross, 6<sup>th</sup> ed. For copyright information visit: <http://www-net.cs.umass.edu/kurose-ross-ppt-6e/>

The Internet is a highly **complex** system

- integrates a wide variety of end-systems and links
- supports an ever-growing number of applications
- resilient to link failures and packet loss

How can we grasp this complexity?

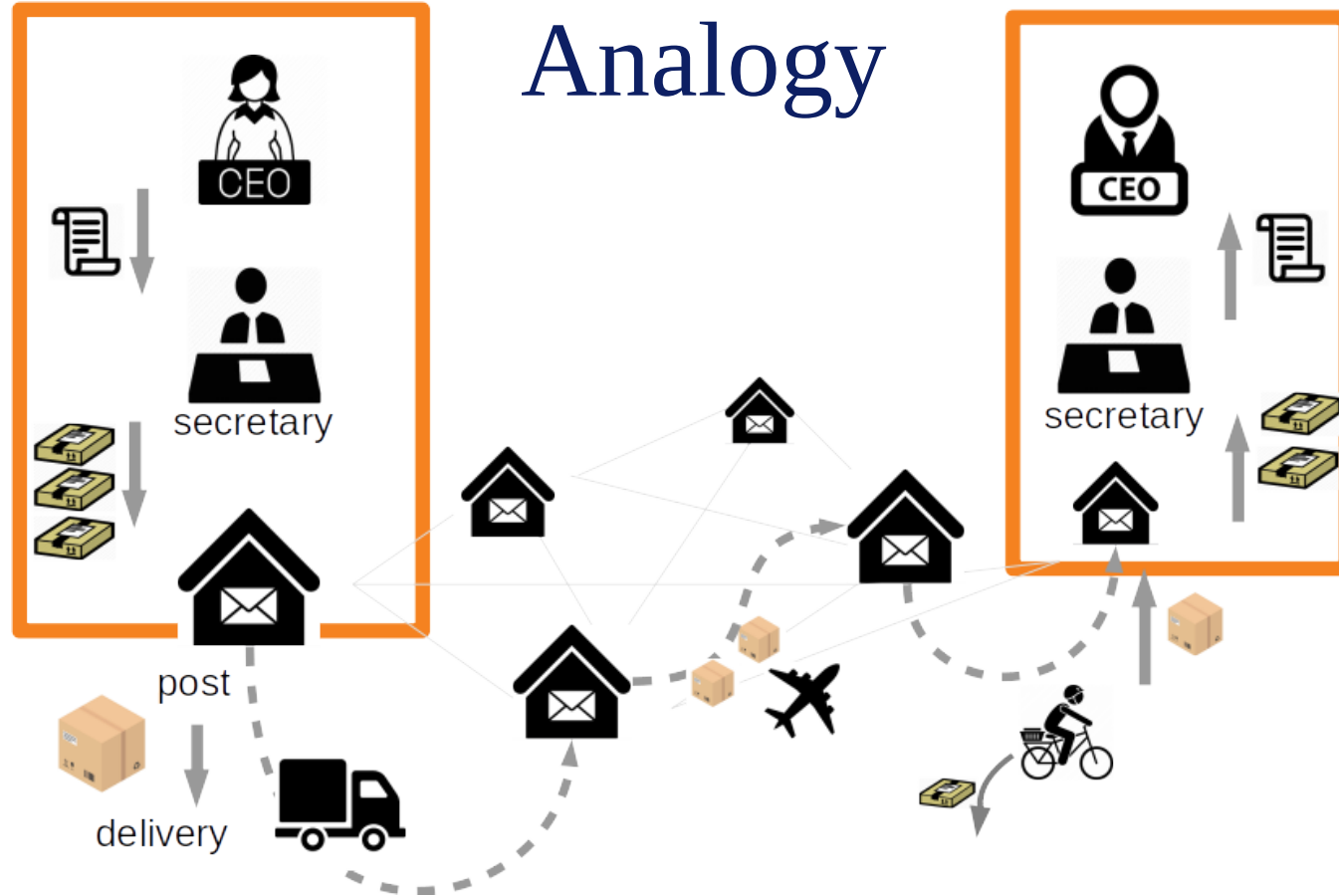
# Layering

- A common engineering approach:  
Divide the functionality into parts or “**layers**”
- **Fixed:**
  - **What** each layer does, what **service** it provides
  - **Interface:** how one layer interacts with another layer
- **Variable:**
  - **How** each layer does its job

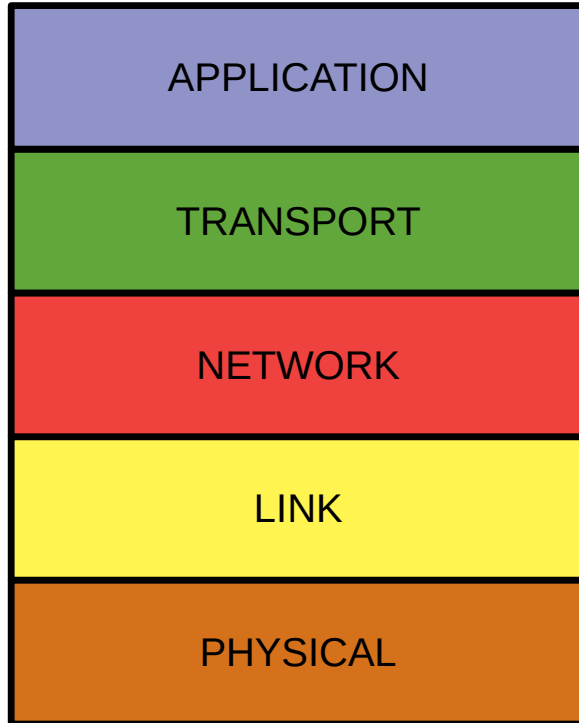
# Advantages of Layering

- Separation of concerns:
  - easier for humans to understand/design/engineer
- Each layer can be improved/changed independent of the other layers.

# A Postal Analogy



# Layers of the Internet (TCP/IP Model)



# The Application Layer

- **Consists of Applications (or Processes) running on hosts**
- Applications determine “**what**” is the data/message being sent across, and what to do with received messages.
- Use the services of the layers below to send/receive messages across the network (not concerned with “**how**” the messages get across)

Different **applications** might **require different types of services** from the layers below

- **Reliable, In-order** delivery of a stream of Bytes

Example: Web browsing, Email ...

OR

- **Simple, unreliable delivery** of a stream of Bytes (ok if some Bytes get lost)

Example: Video streaming, Skype, ...



# The Transport Layer

- **Provides Process-to-process message delivery service to Applications**
- Applications can choose between **different types of services** offered by the Transport layer:
  - **TCP: Reliable, In-order delivery** of a stream of Bytes
    - Performs numbering and re-ordering of packets, re-sending lost packets, etc.
    - Also does congestion control, flow control
  - **UDP: Simple, best-effort, unreliable delivery** of Bytes
    - Just send the packets across

# The Network Layer

- **Provides Host-to-host delivery service to the Transport layer**
- Responsible for:
  - **Routing a packet** to a given destination host (determine which path to take)

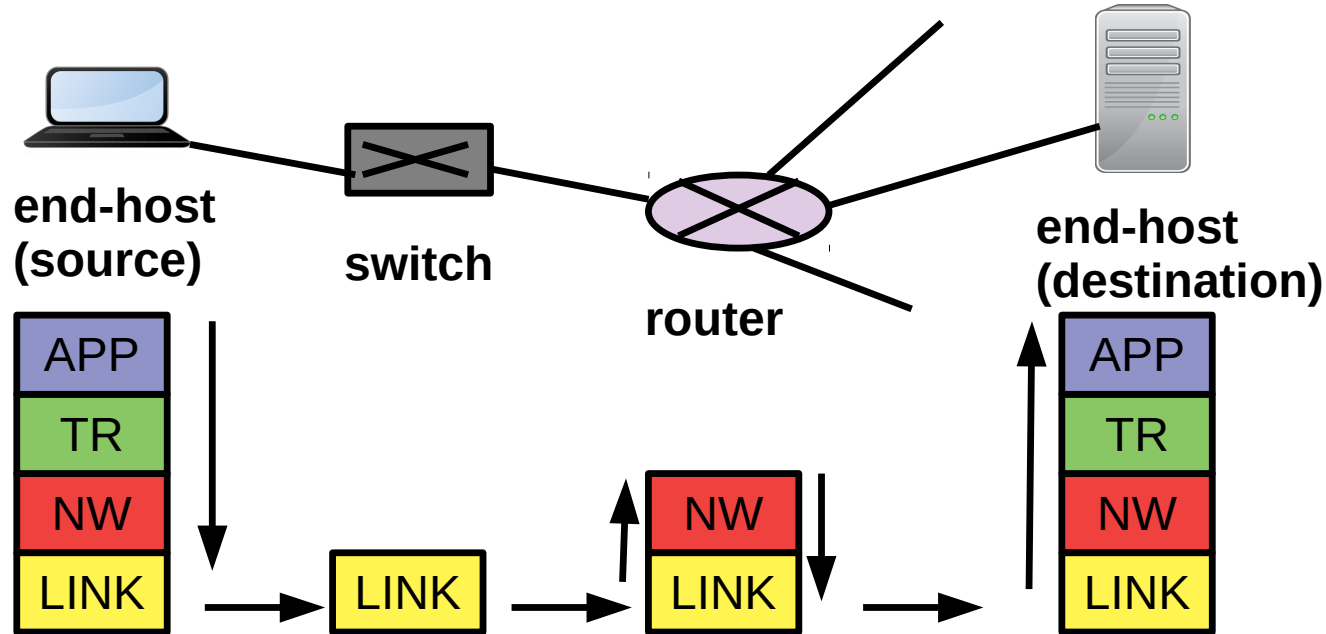
# The Link Layer

- Responsible for **delivering a packet along a single link/hop**
- Different types of links: Ethernet, WiFi etc.
- The link layer hides details of “how” to send a packet along a single type of link.
- Some responsibilities of the Link layer:
  - In case of a broadcast medium, who gets to transmit? What happens on a collision? How to send packet to only one recipient?
  - Some links may provide error detection/error correction

# The Physical Layer

- Responsible for:
  - How to **transmit “bits” over a physical medium?**
  - How are 0s and 1s encoded? What is the maximum range/length supported
- Many protocols/standards exist, for different types of communication links

# Where are the Layers Implemented?



# Protocols

- **Application Layer:**

HTTP (Web browsing) , SMTP (Email)

FTP (File Transfer), DNS (Translation of urls to IP address), ....

- **Transport Layer:**

TCP (Reliable, in-order delivery), UDP (best-effort delivery), ...

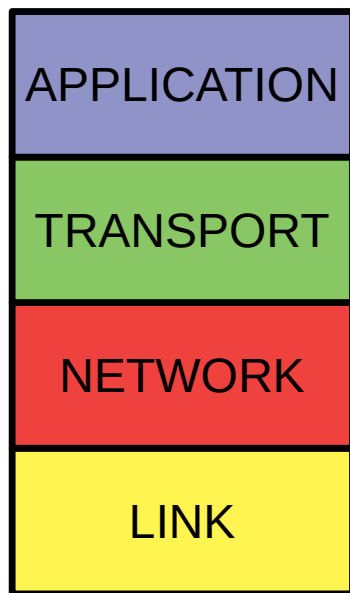
- **Network Layer:**

IP (IPv4, IPv6, ...), Routing Protocols ...

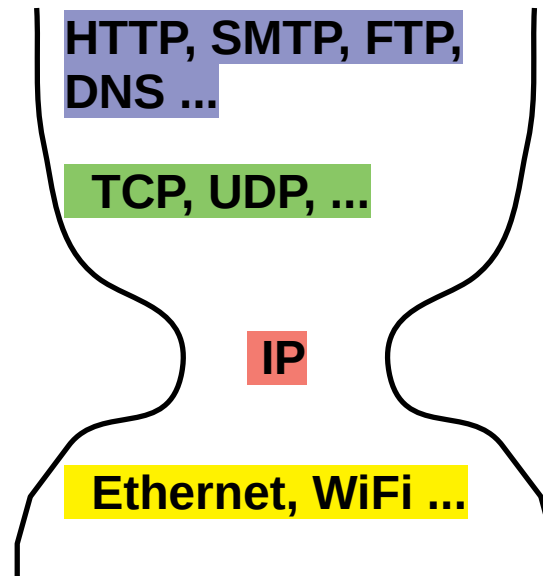
- **Link Layer:**

Ethernet, WiFi, ...

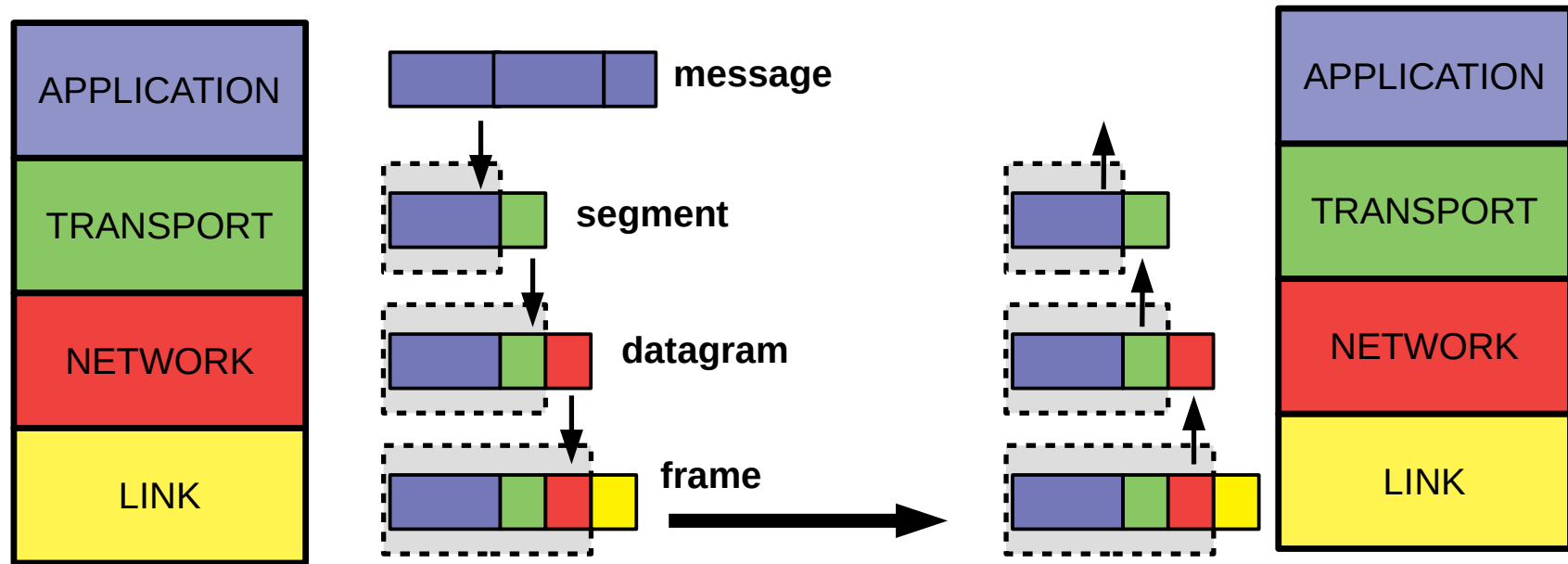
# The Protocol Stack



## PROTOCOLS



# Key Idea: Encapsulation





# Some Questions

- What exactly happens within each layer? How does each layer function?
- **Where** are these layers implemented?
- How exactly do the layers **interact** with each other?
- If I write an application, how can I use the services of the Transport layer?
- Why is this layered model called TCP/IP? Who proposed/designed this layered model? Are there any alternative models?
- Are there any disadvantages to such a layered approach?
- Difference between a Process, Program and an Application?

# Questions

- Can I see/examine packets leaving my NIC? How?
- Can I see/examine other people's packets on a shared medium?
- Is my MAC ID visible to the destination host (such as a web-server)?  
What other information about me is visible?

# Reading Assignment

- Kurose and Ross: Section 1.5