

---

## Lecture 2

# Protocol Stacks and Layering

David Andersen  
School of Computer Science  
Carnegie Mellon University

15-441, Computer Networks

1

---

## Last Time

- The Big Picture
  - Goals:
    - Efficiency
    - “ilities” (scalability, manageability, availability),
    - Ease of creating applications
  - Challenges:
    - Scale
    - Geography
    - Heterogeneity (\*\* today’s focus!)
- A few specific details:
  - Circuits vs. packets
  - Little bit about routing
  - Service model and how to construct services (\*\* today!)

2

---

## Today’s Lecture

- Last time: “Big picture”
- Today:
  - General architectural principles for networks
  - Introduce a few concrete models & examples
- Where we are going:
  - Tuesday: Socket programming review\*\* (for project)
  - Thursday: Application examples (still high level)
  - After that: Diving into the details, ground up
- Today’s specifics:
  - What is a protocol.
  - Protocol stacks.
  - Some history.
  - Standards organizations.
  - Application layer.

3

---

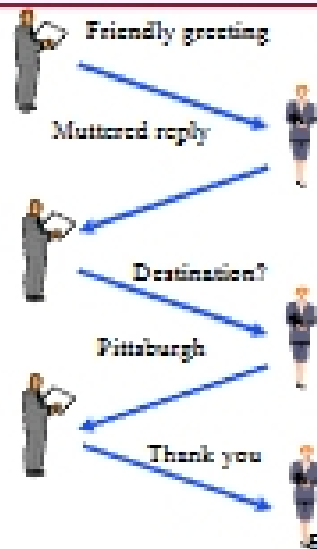
## Why protocols and layering?

- Interoperability
- Reuse
- Hiding underlying details

4

## What is a Protocol

- An agreement between parties on how communication should take place.
- Protocols may have to define many aspects of the communication.
- **Syntax:**
  - Data encoding, language, etc.
- **Semantics:**
  - Error handling, termination, ordering of requests, etc.
- Protocols at hardware, software, all levels!
- Example: Buying airline ticket by typing.
- Syntax: English, ASCII, lines delimited by "\n"

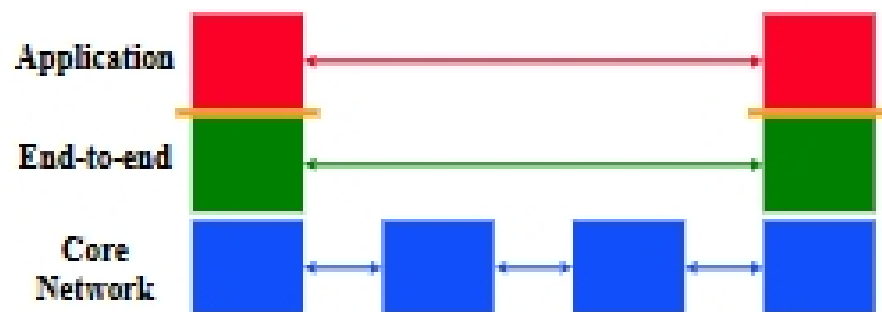


## Interfaces

- Each protocol offers an interface to its users, and expects one from the layers on which it builds
  - Syntax and semantics strike again
    - Data formats
    - Interface characteristics, e.g. IP service model
- Protocols build upon each other
  - Add value
    - E.g., a reliable protocol running on top of IP
  - Reuse
    - E.g., OS provides TCP, so apps don't have to rewrite

8

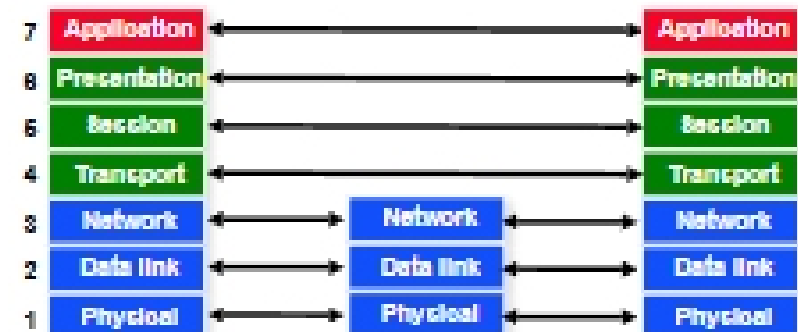
## Protocol and Service Levels



7

## A Layered Network Model

The Open Systems Interconnection (OSI) Model.



8

## OSI Motivation

- Standard way of breaking up a system in a set of components, but the components are organized as a set of layers.
  - Only horizontal and vertical communication
  - Components/layers can be implemented and modified in isolation
- Each layer offers a service to the higher layer, using the services of the lower layer.
- "Peer" layers on different systems communicate via a protocol.
  - higher level protocols (e.g. TCP/IP, Appletalk) can run on multiple lower layers
  - multiple higher level protocols can share a single physical network
- "It's only a model!" - TCP/IP has been crazy successful, and it's not based on a rigid OSI model. But the OSI model has been very successful at shaping thought.

9

## OSI Functions

- (1) Physical: transmission of a bit stream.
- (2) Data link: flow control, framing, error detection.
- (3) Network: switching and routing.
- (4) Transport: reliable end to end delivery.
- (5) Session: managing logical connections.
- (6) Presentation: data transformations.
- (7) Application: specific uses, e.g. mail, file transfer, telnet, network management.

Multiplexing takes place in multiple layers

10

## Looking at protocols

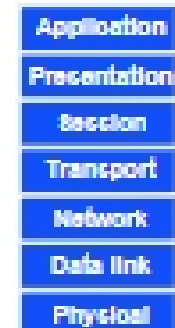
- Hop by hop / link protocols
  - ethernet
- End-to-end protocols
  - TCP, apps, etc.
- Management / "control plane" protocols
  - Routing, etc.
    - Can be either link or e2e themselves
    - Definition somewhat vague.
- Standards
  - File formats, etc.
    - E.g., JPEG, MPEG, MP3, ...

Categories not solid / religious, just a way to view things.

11

## Heterogenous Sources of Components

- Application: web server/browser, mail, distributed game, ...
- Presentation/session.
  - Other part of application
  - Serialization, ... library
- Transport/network.
  - Typically part of the operating system
- Datalink.
  - Often written by vendor of the network interface hardware
- Physical.
  - Hardware, serial and link



12