Today's Lecture

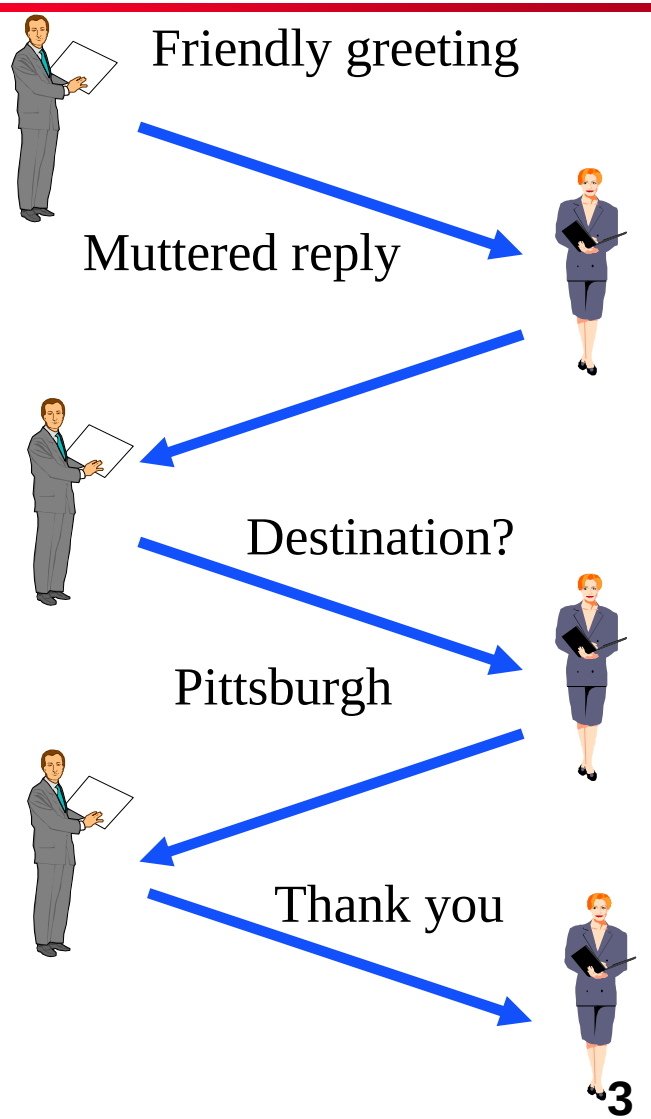
- **Last time: "Big picture"**
- **Today:**
  - **General architectural principles for networks**
  - **Introduces a few concrete models & examples**
- **Today's specifics:**
  - **What is a protocol.**
  - **Protocol stacks.**
  - **Some history.**
  - **Standards organizations.**
  - **Application layer.**

Protocols

- **Recall goals:**
  - **Interoperability**
  - **Reuse**
  - **Hiding underlying details**

Protocols

- An agreement between parties on who 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”



More on Protocols

- **Protocols are the key to interoperability.**
  - » **Networks are very heterogenous:**

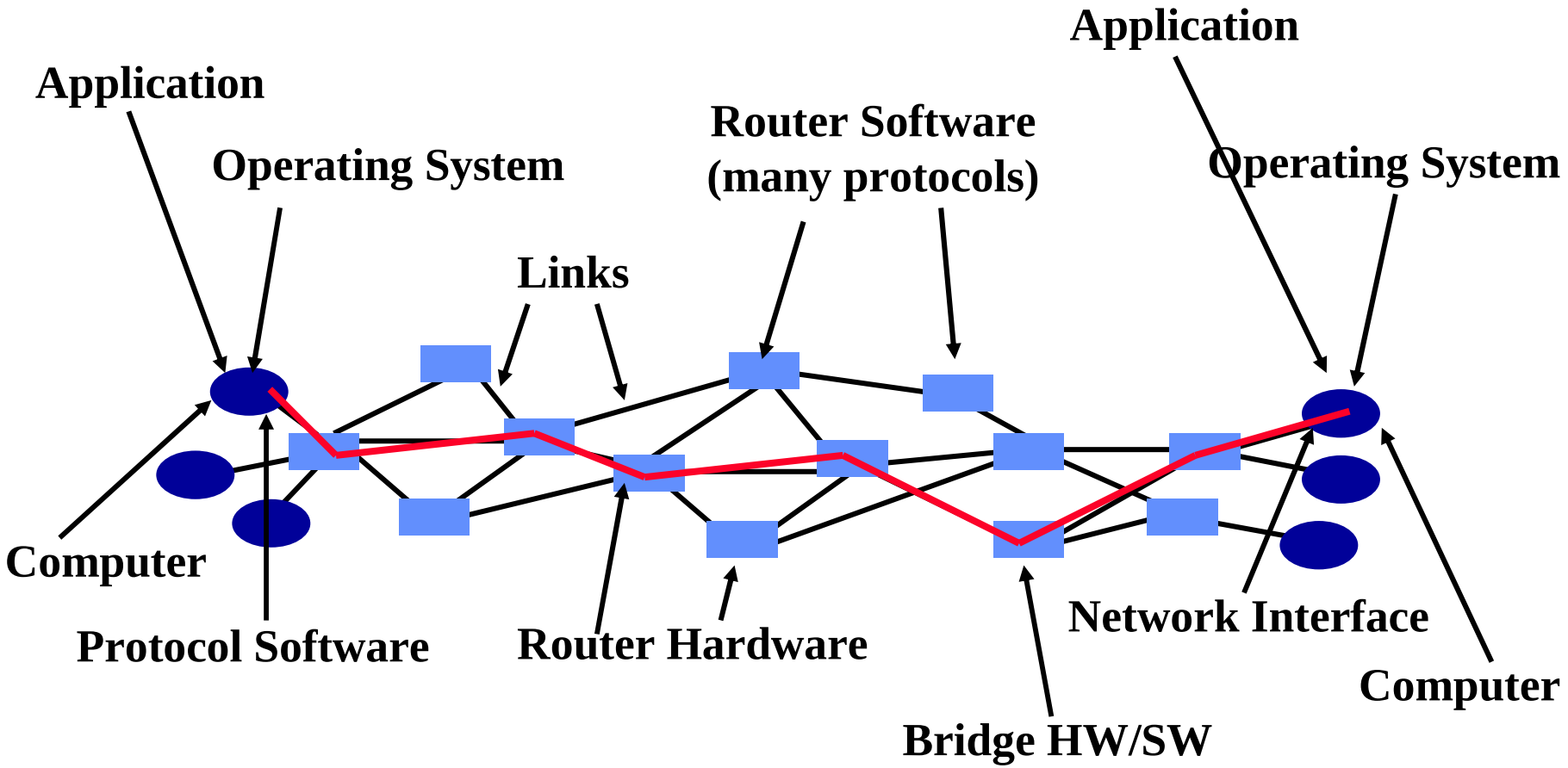
Computer: x86	**Hardware**
Ethernet: 3com	**Hardware/link**
Routers: cisco, etc.	**Network**
App: Email	**Application**

  - » **The hardware/software of communicating parties are often not built by the same vendor**
  - » **Yet they can communicate because they use the same protocol**
- **Protocols exist at many levels.**
  - » **Application level protocols, e.g. access to mail, distribution of bboards, web access, ..**
  - » **Protocols at the hardware level allow two boxes to communicate over a link, e.g. the Ethernet protocol**

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

Too Many Network Components



Too many components 2

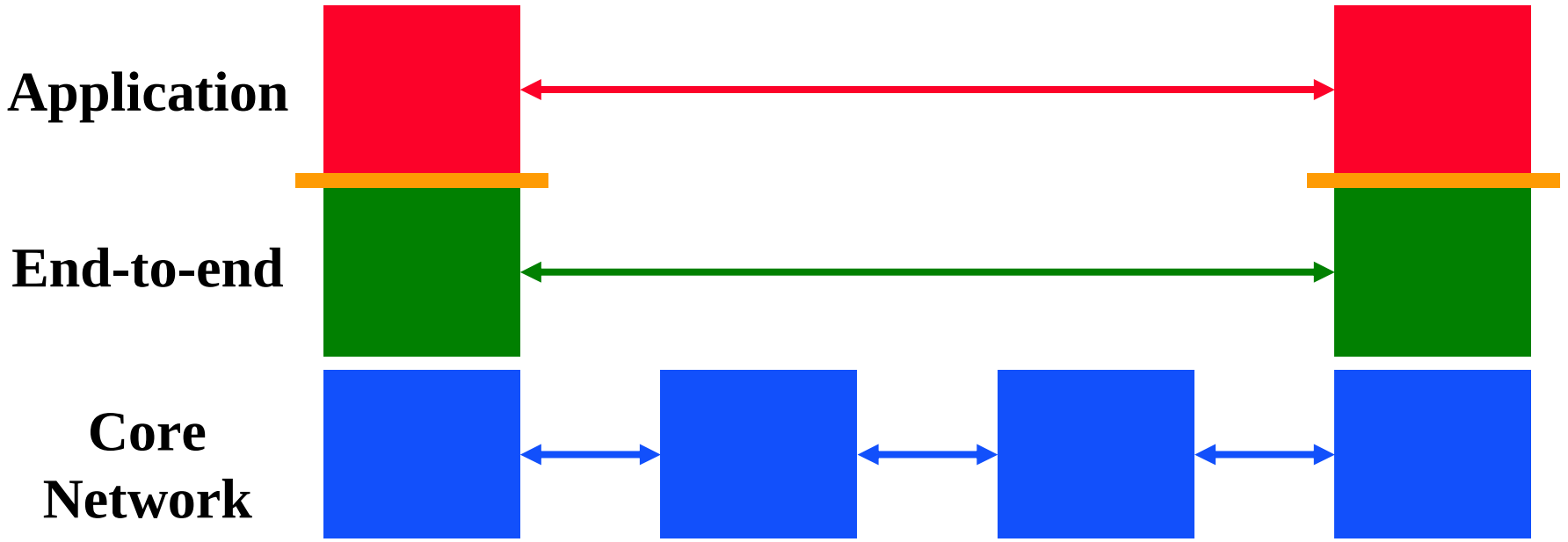
- Links: copper, fiber, air, carrier pidgeon
- Running ethernet, token ring, SONET, FDDI
- Routers speaking BGP, OSPF, RIP, …
- Hosts running FreeBSD, Linux, Windows, MacOS, …
- People using Mozilla, Explorer, Opera, …
- Protocols hide this stuff with simple abstractions.

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, …

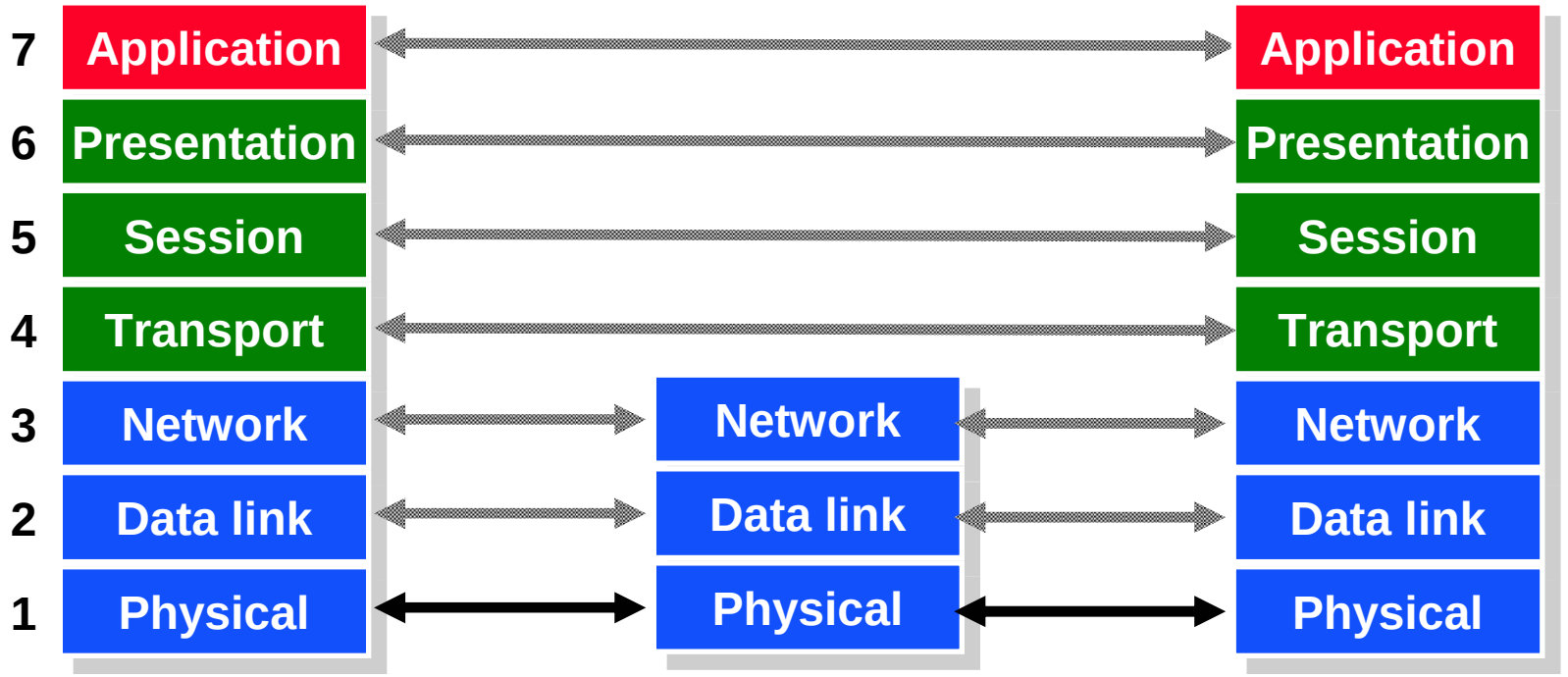
Categories not solid / religious, just a way to view things.

Protocol and Service Levels



A Layered Network Model

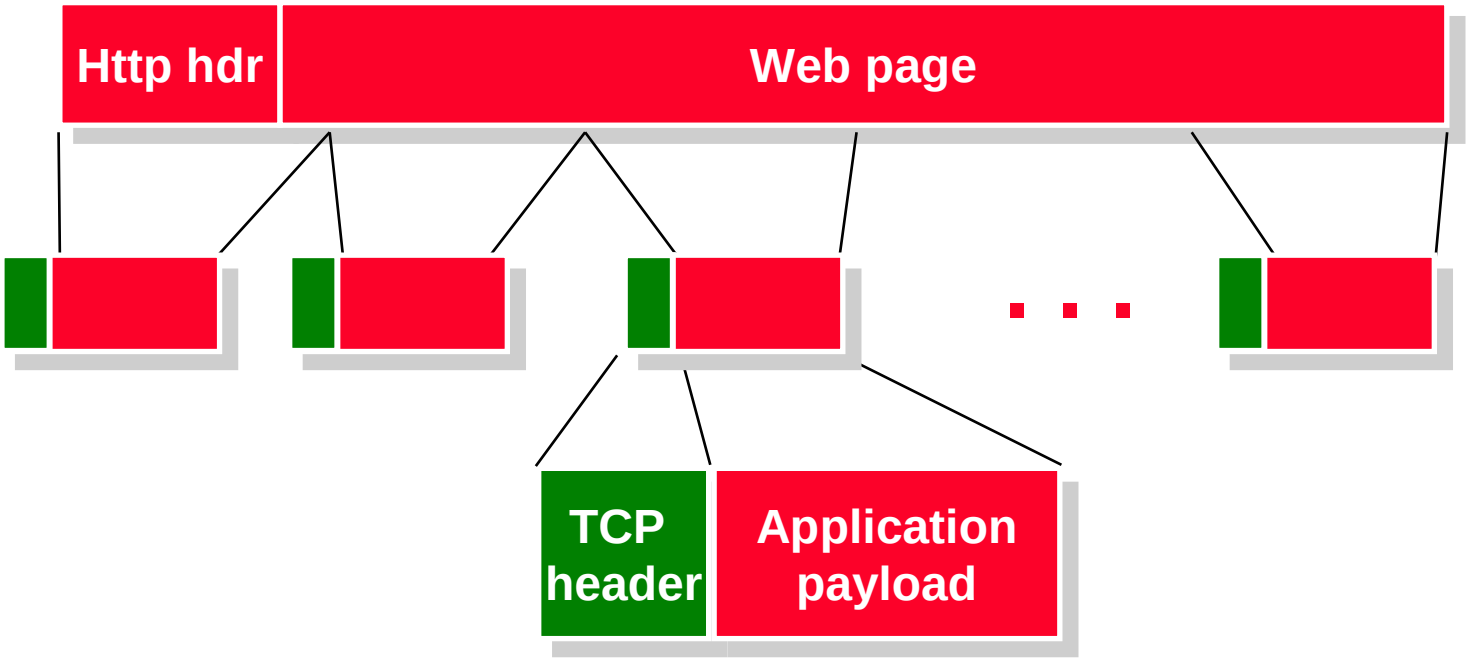
The Open Systems Interconnection (OSI) Model.

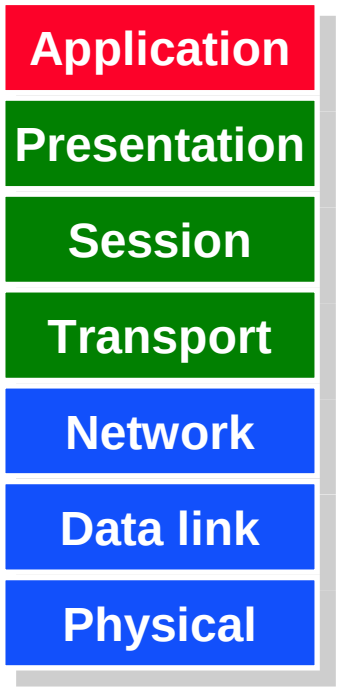


OSI Motivation

- **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.**

Example: Sending a Web Page





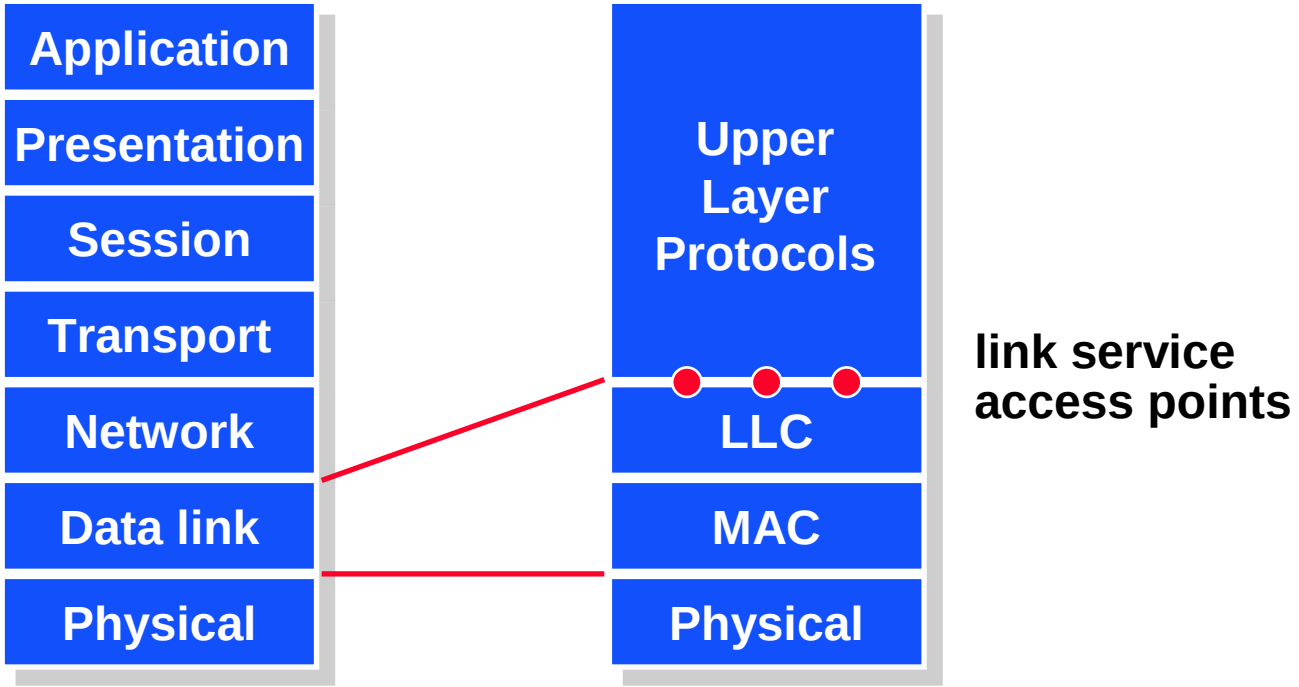
Limitations of the Layered Model

- **Some layers are not always cleanly separated.**
  - » **Inter-layer dependencies in implementations for performance reasons**
  - » **Some dependencies in the standards (header checksums)**
- **Higher layers not always well defined.**
  - » **Session, presentation, application layers**
- **Lower layers have “sublayers”.**
  - » **Usually very well defined (e.g., SONET protocol)**
- **Interfaces are not really standardized.**
  - » **It would be hard to mix and match layers from independent implementations, e.g., windows network apps on unix (w/out compatability library)**
  - » **Many cross-layer assumptions, e.g. buffer management**

The TCP/IP Model

Local Area Network Protocols

IEEE 802 standards "refine" the OSI data link layer.



Standardization

- **Key to network interoperability.**
- **A priori standards.**
  - » **Standards are defined first by a standards committee**
  - » **Risk of defining standards that are untested or unnecessary**
  - » **Standard may be available before there is serious use of the technology**
- **De facto standards.**
  - » **Standards is based on an existing systems**
  - » **Gives the company that developed the base system a big advantage**
  - » **Often results in competing “standards” before the official standard is established**

Relevant Standardization Bodies

- **ITU-TS - Telecommunications Sector of the International Telecommunications Union.**
  - » **government representatives (PTTs/State Department)**
  - » **responsible for international “recommendations”**
- **T1 - telecom committee reporting to American National Standards Institute.**
  - » **T1/ANSI formulate US positions**
  - » **interpret/adapt ITU standards for US use, represents US in ISO**
- **IEEE - Institute of Electrical and Electronics Engineers.**
  - » **responsible for many physical layer and datalink layer standards**
- **ISO - International Standards Organization.**
  - » **covers a broad area**