Computer Networks

Introduction

Topics

- Use of networks
- Network structure
- Implementation of networks

Let's Get Started!

- Networking today: "Where are they?"
  - Powerful computers are cheap
  - Networks are everywhere
- Blurred lines: "What are they?"
  - multi-processors
  - devices
  - local networks
  - metropolitan networks
  - long-haul networks

Computer Networks: Our Definition

An interconnected collection of *autonomous* computers

interconnected: can exchange information
- via fiber, copper, wireless

- autonomous: no master-slave
  - no multiprocessors
  - no computer with devices

Computer Network Components

- Hardware
  - "physically" connects machines (can send signals)
- Software
  - Protocols specify services the network uses
  - Make the network hardware convenient
    - (Sound familiar? ala Operating System!)
- Software more important (hence this class)
  - (But may want to check with ECE :-))
  - (Re-work from last time I taught)

How are Networks Used by Computers?

- Autonomous Systems
  - `rsh, rcp`
- Network File System
  - NFS
- Distributed Operating Systems
  - User sees a single large virtual computer system
  - Few, none are products.
- All use client-server (Fig 1-1)

Why are Networks used by People?

- Resource Sharing
  - printers, terminals, special architectures
- Information Sharing
  - e-mail, world wide web
- Improve Reliability
- Improve Power (per cost)
  - networked PC's as powerful as a mainframe
- Killer Apps
  - Video on Demand (Last year's Project 3)
  - Online Games (This year's Project 2/3)

Effect on Society

- "Information Superhighway"
- Electronic "conversations"
  - email, bulletin boards, chat rooms
  - different than face-to-face, phone, mail
- World Wide Web
  - instant sharing of information
  - true "desk-top-publishing"
  - electronic retailing

Network Structure

- *Host* or *End-System*
  - a computer that a user logs into to do work
  - attached to network, not part of network (usually)
- *Subnet*
  - everything between hosts
  - transport data from one host to another

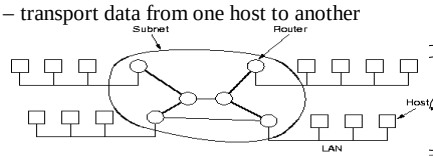


Fig. 1-5. Relation between hosts and the subnet.

Subnet

- *Point-to-Point*
  - Two machines, one at each end of a "wire"
  - Often many point-to-points in a subnet

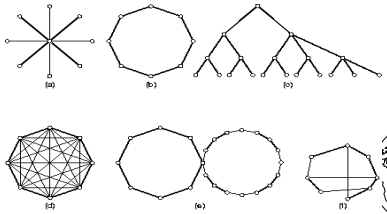


Fig. 1-6. Some possible topologies for a point-to-point subnet. (a) Star. (b) Ring. (c) Tree. (d) Complete. (e) Intersecting rings. (f) Irregular.

Subnet

- *Broadcast*
  - Many (3+) machines connected by a common link
  - When one "speaks", all hear

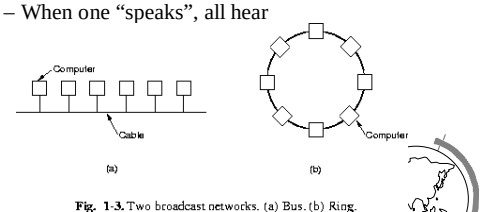


Fig. 1-3. Two broadcast networks. (a) Bus. (b) Ring.

- *Multicast* targets only some
- *Unicast* send to only one

Types of Network Structures

- LAN - Local Area Network
- MAN - Metropolitan Area Network
- WAN - Wide Area Network
- Wireless / Mobile Networks

Local Area Networks (LANs)

- Small geographic regions (e.g., building(s))
- High data rates (10-100 Mbps and up)
  - Much higher than connection to ISP
- Low cost (thousands of dollars)
- Typically broadcast

Metropolitan Area Networks (MANs, not MEN)

- Medium-size geographic regions (e.g., entire cities)
- Still no switches, single "wires"
- Example: local cable system
- IEEE 802.6--Distributed Queue Dual Bus (DQDB)
  - Uses two broadcast buses, one for each direction

Wide Area Networks (WANs)

- Larger geographic distance (e.g. entire countries)
- Low data rates (56 kbps - 1.5 Mbps (T1), bundle T1 links to get higher rates),
- High cost (tens or hundreds of thousands of dollars per year)
- The *Internet* is a specific WAN

Wireless / Mobile Networks

- Fastest growing network segment
- Notebook computers and portable digital assistants (PDAs) to base
- Portable network for military use
- Wireless is not necessarily mobile

Wireless	Mobile	Applications
No	No	Stationary workstations in offices
No	Yes	Using a portable in a hotel; train maintenance
Yes	No	LANs in older, unwired buildings
Yes	Yes	Portable office; PDA for store inventory

Fig. 1-7. Combinations of wireless networks and mobile computing.

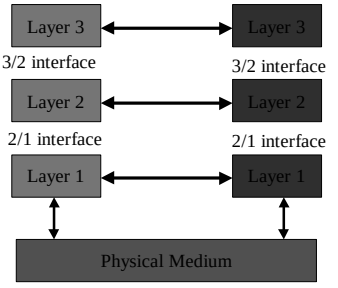
Internetworking

- The connection of different types of networks
- The *Internet*

Implementing Networks

- Need software abstraction to make hardware convenient
- Complex problem (remember OS?)
  - Where do we start?
  - Divide-and-Conquer!
- Layer up from hardware
- Only bare amount needed
- Increasingly sophisticated services

Layering



Virtual Communication

Abstraction

Transparency

Layers and protocols form *network architecture*

Network Architecture

- Two fundamental concepts
  - *messages*
  - *encapsulation*

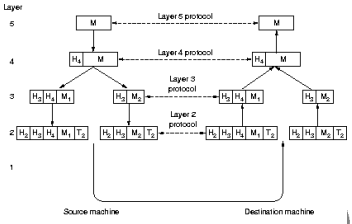


Fig. 1-11. Example information flow supporting virtual communication in layer 5.

Messages

- Each layer deals with messages
- Have maximum size (ex Ethernet 1500 bytes), 100s-1000s bytes
- Have *control* or *header*
  - used to synchronize with the remote peer
  - contain "instructions" that tell the remote peer what to do with the message
- Have *data* portion
  - arbitrary bytes
  - not of interest in this particular protocol layer

Encapsulation

- Layer N takes data from layer N+1 (above it)
  - encapsulates entire layer N+1 message in the data portion of the layer N
  - it should never look inside the data portion of the message!
- When the remote peer receives a message
  - it strips off the header information and passes only the data to the next higher layer

Network Layer Examples

- Open Systems Interconnection (OSI)
- TCP/IP

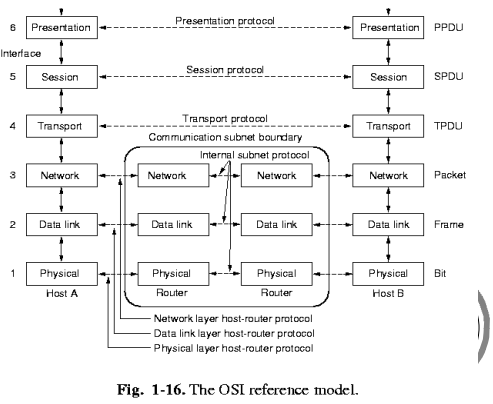


Fig. 1-16. The OSI reference model.

OSI Reference Model

- Standard attempt
- 7 layers:
  - Physical Layer
  - Data Link Layer
  - Network Layer
  - Transport Layer
  - Session Layer
  - Presentation Layer
  - Application Layer
- Layers self-contained
- Minimize messages across boundaries

Physical Layer

- Transmitting raw bits over a "wire"
  - Make sure a "1" bit is sent as a 1
- EE/ECE problem:
  - How many volts represents a "1" or "0"?
  - How long does a bit time last?
  - How many pins does the connector have?
  - How many wires does the transmission media have?
  - Are pulses electrical or optical or waves?

Data Link Layer

- Communication between two machines
- Transforms raw transmission of physical layer into error-free channel
- Divides physical layer physical layer into *frames*
  - messages containing data and control information
- Handles lost, damaged, and duplicate frames
- Handles slowing down a fast transmitter
  - *flow-control*

Network Layer

- Controls operation of the *subnet*
  - communication between hosts
- Routes *packets* from source to destination
  - not guaranteed delivery
- Handles *congestion*
  - too many packets in network
- Handles *addressing*
  - Which machine?

Transport Layer

- Makes sure data gets delivered to a specific process on a specific machine
- *End-to-end* protocol
  - sender and receiver
- Handles retransmissions, if needed
- Handles duplicates, if needed
- Also deals with addressing
  - Which process on a particular machine?
  - The *port* specification in a `socket`

Session Layer

- Long-term connections between processes
- Clean interface to the transport layer
  - Not OS specific (sockets in BSD Unix, or TLI in System V streams)
- Provides synchronization
  - recovering from transport layer failure
  - *token* for floor control

Presentation Layer

- Apply semantics to data
  - example: name, address …
- Format in agreed upon way
- General services:
  - Format data (ASCII to Unicode)
  - Compressing data
  - Encryption

Application Layer

- The user programs themselves
  - ftp
  - telnet
  - X
  - talk

Critique of OSI

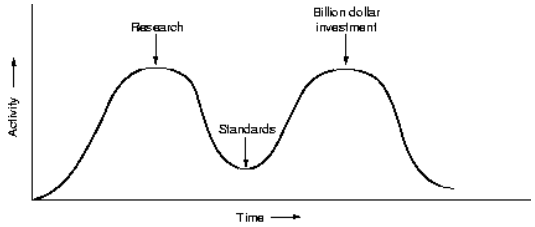


Fig. 1-20. The apocalypse of the two elephants.

- Plus, bad technology (big specification)
- Plus, bad politics (pushed by govt. orgs)

ARPANET

- Predecessor to the Internet
- Phone lines first, satellite and radio later
  - req: connect multiple networks seamlessly
- DoD worry about routers going down
  - req: survive loss of subnet hardware without losing connections
- Applications with diverse requirements
  - req: flexible architecture
- Used TCP/IP protocols
  - **then** came their reference model

TCP/IP Reference Model

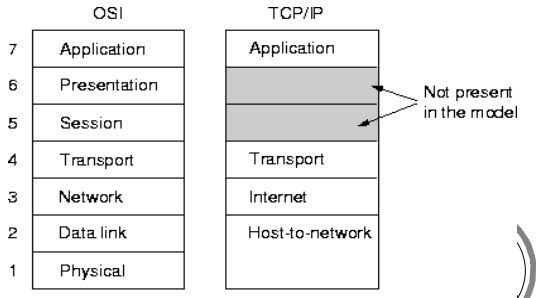


Fig. 1-18. The TCP/IP reference model.

Internet Layer

- Packet switched
- Connectionless
- Packets can be:
  - travel different routes
  - lost
  - out of order
- Called *IP* (*Internet Protocol*)

Transport Layer

- Similar to OSI Transport Layer
  - end-to-end, “conversation”
- Two protocols
  - TCP: reliable, stream, flow control, connection
  - UDP: unreliable, no flow control, connectionless

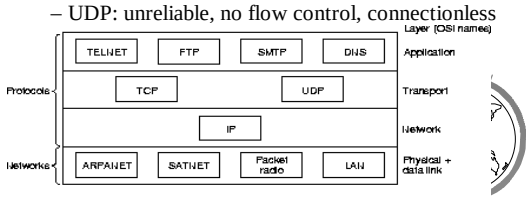


Fig. 1-19. Protocols and networks in the TCP/IP model initially.

Application Layer

- No session/presentation layers -- no need
- High-level protocols:
  - original: telnet, ftp, smtp, dns
  - new: http, nntp

Host-to-Network Layer

- Great void
- Not specified, not talked about in research literature

Critique of TCP/IP Model

- Not clean in describing service, interface and protocol
  - not a good guide for new technologies
- Not general, tied to protocols
  - hard to describe other networks
- No physical and data link layers
  - hard to abstract from physical hardware
  - “re-invent the wheel”
- IP, TCP well-thought out, but others not
  - TELNET: 10 cps, no GUI, no mouse

Model Differences: OSI and TCP/IP

- OSI concepts:
  - services: what layer does
  - interface: how processes above access it
  - protocols: how it works, private to layer
  - great for OO!
- Not so clean in TCP/IP
  - harder to replace as technology changes

Differences: OSI and TCP/IP

- OSI model before protocols
  - implementations hacked (ex - broadcast instead of point-to-point needed new layer)
- TCP/IP protocols before model
  - model does not fit other protocols
  - not useful for non TCP/IP networks
- OSI transport
  - connection oriented only
- TCP/IP transport
  - connection + connectionless

Hybrid Model

- OSI useful for discussing networks
- TCP/IP provides better protocols for using them

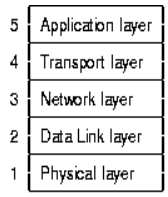


Fig. 1-21. The hybrid reference model to be used in this book.

ATM Overview

- Telephone companies coordinate multiple networks
  - ex: POTS circuit-switched, other packet-switched
- Invent network of future to manage all
  - Broadband-ISDN
- B-ISDN made possible by *Asynchronous Transfer Mode* (*ATM*)

ATM Basics

- Transmit data if fixed sized *cells*
  - Flexible (audio, video, text)
  - Fast (155 Mbps and 622 Mbps)
  - But , huge break from circuit switching

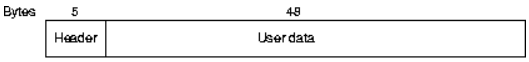


Fig. 1-29. An ATM cell.

- Connection oriented
- Niche, for now, is connecting LAN's

Outline for Rest of Course

- Intro, reference models, ch 1 (2 days)
- Physical layer, ch 2 (2 days)
- Data link layer, ch 3 (4 days)
- Medium access sublayer, ch 4 (2 days)
  - midterm exam
- Network layer, ch 5 (4 days)
- Transport layer, ch 6 (4 days)
  - UDP/TCP/IP, ch 6.4 (2-3 days)
- Upper layers, misc, chap 7
  - final exam