CS 640 Introduction to Computer Networks

Lecture 1

Today's lecture

- Introduction to communication networks
  - Purpose
  - History
  - Underlying technologies
- Fundamental problems
  - Reliability
  - Resource sharing, cooperation and competition
  - Structure, growth and evolution

Purpose of communication networks

- Carrying messages from sender to receiver
- Supporting conversations between people
- Enabling access to sources of information
- Dissemination of information
- Simulating user's presence at remote locations
- Remote monitoring

- The list keeps evolving

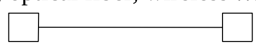
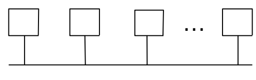
A brief history of the Internet

- Leonard Kleinrock published work in ‘61
  - Showed packet switching effective for bursty traffic
- DARPA contract for BBN to build switches
  - First network had four nodes in ‘69
- Email first application – Ray Tomlinson, ’72
- In ’74 Vint Cerf and Robert Kahn developed open architecture for Internet
  - TCP and IP

History of the Internet (cont.)

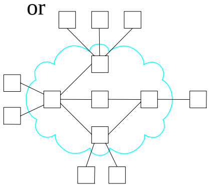
- By ’79 the 200 nodes and by’89 over 100K!
  - Much growth fueled by connecting universities
  - Larry Landweber from UW had important role
- In ’89 Van Jacobson made MAJOR improvements to TCP
- In ’91 Tim Berners-Lee invented the Web
- In ’93 Marc Andreesen invented Mosaic
- The rest should be pretty familiar…

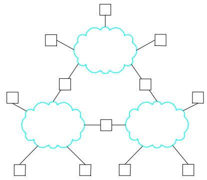
Building Blocks

- Nodes: PC, special-purpose hardware…
  - hosts (PCs, cell phones, toasters, etc.)
  - routers (and switches, bridges, hubs, etc.)
- Links: coax cable, optical fiber, wireless …
  - point-to-point 
  - multiple access 

Nodes and links form networks

- A network can be defined recursively as...
  - two or more nodes connected by a link, or
  - two or more networks connected by nodes





Reliability

- Links corrupt transmissions, nodes fail
- Information transmitted over the network usually has to arrive unaltered
- Basic techniques against data corruption
  - Error detection
  - Error correction
- Basic technique against equipment failures
  - Dynamic reconfiguration

Error detection

- Underlying technology can sometimes indicate failure (e.g. signal too weak)

- Bits are grouped into packets (a.k.a. frames)
- Checksum added to the packet by sender
- Receiver verifies the checksum
  - If it matches, bits in packet assumed correct
  - Otherwise receiver assumes some bits corrupted

Error correction

- Sender can retransmit corrupted packets
- Network discards some packets without notice
  - Timeouts at sender trigger retransmission
- Forward error correction: encode data redundantly – can recover from a few errors
- Some applications (voice) can tolerate loss or corruption of a few packets

Dynamic reconfiguration

- Nodes keep gossiping about state of network
- Handle packets based on map learned this way
- When nodes or links fail, packets take detour
- Such dynamic reconfiguration is implemented by routing protocols and learning bridges
  - It's more complicated in practice

Today's lecture

- Introduction to communication networks
  - Purpose
  - History
  - Underlying technologies
- Fundamental problems
  - Reliability
  - Resource sharing, cooperation and competition
  - Structure, growth and evolution

Sharing, cooperation and competition

- Multiplexing (methods for sharing)
- Medium access control (who's turn is it?)
- Congestion control (can have too much traffic)
- Quality of service (I want good service)
- Denial of service (you don't get any service)
- Cryptography – privacy, authentication
- Pricing, accounting, incentives

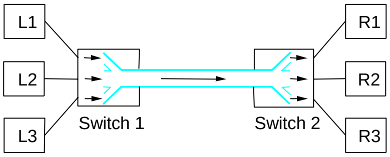
Multiplexing

- Frequency Division Multiplexing (FDM)
  - Each radio station uses a different frequency
- Synchronous Time Division Multiplexing
  - Each "user" gets a small time slot periodically
  - Works for phone calls (steady data)

  - Data communication very bursty

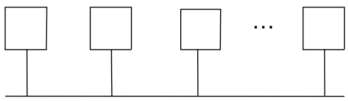
Statistical multiplexing

- Whoever has data to send, sends
- Like CPU sharing among processes in OS



Who’s turn is it to send?

- Scheduling at routers decide which packets to send, which ones to buffer, which ones to drop
- Medium access control
  - If many hosts send at the same time, it gets garbled
  - Pick which host sends onto multiple-access link



Congestion

- There is more traffic than the network can take
- Solution 1: senders all slow down (TCP)
- Solution 2: drop packets of biggest senders
- Solution 3: build a faster network
- Solution 4: use pricing incentives

- What if the bad guys are congesting your network on purpose (a.k.a. DDoS)?

Today’s lecture

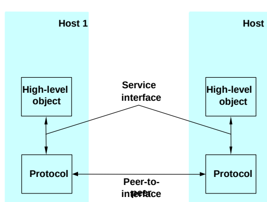
- Introduction to communication networks
  - Purpose
  - History
  - Underlying technologies
- Fundamental problems
  - Reliability
  - Resource sharing, cooperation and competition
  - Structure, growth and evolution

Structure, growth and evolution

- Protocols – clear rules for interaction
- Modularization – layered architecture
- Encapsulation
- Addressing and naming (IP, MAC, DNS)
- Caching – better response times, lower traffic
- New/changing protocols and applications
  - Incremental deployment
  - Interoperability/backward compatibility

Protocols

- Building blocks of a network architecture
- Each protocol object has two different interfaces
  - *service interface*: operations on this protocol
  - *peer-to-peer interface*: messages exchanged with peer (format of messages and behavior)

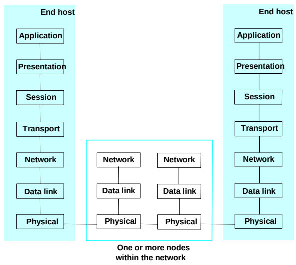


Modularization: layering

- Use abstractions to hide complexity
- Abstraction naturally lead to layering
- Alternative abstractions at each layer

Application programs	
Request/reply channel	Message stream channel
Host-to-host connectivity	
Hardware	

ISO Architecture



CS 640

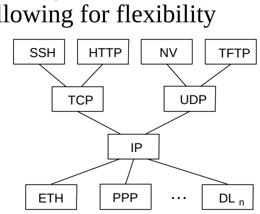
Internet Architecture

- Defined by Internet Engineering Task Force (IETF)
  1. Application: interacts with user to initiate data transfers (browser, media player, command line)
  2. Transport: TCP reliable, in-order delivery of data; UDP
  3. Network: addressing and routing (IP)
  4. Data Link: how hosts access physical media (Ethernet)
  5. Physical: how bits are represented on wire (Manchester)
- Information passed between layers – encapsulation
  - Header information attached to data passed down layers
- Layers access other layers via API's (e.g. sockets)
- Communication at specific layer enabled by a proto.

CS 640

Hourglass Design

- Single protocol at network level ensures packets will get from source to destination while allowing for flexibility



CS 640