CS551
Internet Architecture [Clark88a]

Bill Cheng

http://merlot.usc.edu/cs551-f12

Architecture: Definition

- A style and method of design and construction
- Orderly arrangement of parts
- The manner of construction of something and the disposition of its parts
- Design, the way components fit together
- Ex: railway system, airline system
- A single architecture can have many implementations
  - Ex:
    hub-and-spoke and United/American/Delta
    direct-flights and Southwest/JetBlue

The Internet

- The Internet is one implementation of a particular architecture
- The original Internet architecture
  - a system of store-and-forward packet-switched gateways that provides unreliable packet delivery between any two nodes in the network
  - there have been other implementations of this architecture
    - ARPANET, NSFNet, DECNet, etc.
- Other architectures
  - a virtual circuit based architecture: XUNET

Architecture Principles

- Definitions are vague, so we need guiding principles - but can people agree on what these are?
- The debate is raging on! Just browse *www.ietf.org* sometime
- Now: original principles
- End of class: look at current debate about Internet architecture

Internet Architecture Goals [Clark88a]

- Top-level goal:
  - Connect a number of distinguishable networks
  - Multiple applications and services over the Internet
- Basic design:
  - Packet switched network
  - Store and forward gateways between component networks

IP Design Principles

- Survivability
  - If a path exists, communication continues transparently
  - Fate sharing
- Hourglass design
  - IP makes minimal assumptions about underlying medium, and doesn't get in the way of applications
- Soft-state
  - Robust way to identify communication ***flows***
  - Helps survivability
- Autonomous systems
  - Each network owned and managed separately

Slogans For Computer Network Design

- Perfection is achieved not when there is no longer anything to add, but when there is no longer anything to take away
  - Antoine de Saint-Exupery
- The simplest explanation is the best
  - Occam's razor
- Be liberal in what you accept, and conservative in what you send
  - Jon Postel
- In allocating resources, strive to avoid a disaster rather than to achieve an optimum
  - Butler Lampson

The Internet Architecture

- Heterogeneous networks
- Multiplexing via packet switching
- Sub-goals:
  - *robust to network/gateway failure*
  - *multiple kinds of traffic*
  - *multiple kinds of networks*
  - distributed management
  - inexpensive
  - low effort to add host
  - resource accounting

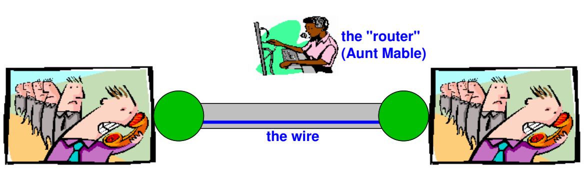
Heterogeneous Networks

- Need to run over existing networks
  - easier to get started and grow
  - pay for what you need
  - decentralized management
  - different technologies (e.g. ethernet, token ring)
  - different capabilities (e.g., wired vs. wireless)
- Multiple wired LANs, last mile, POP-to-POP, satellite, terrestrial wireless (802.11, Bluetooth) technologies
- *"Two cans and a string"*
  - Avian Carriers April Fools day RFC

Packet Switching

- Interleave packets from different sources
  - Efficient: resources used on demand
    - statistical multiplexing
  - General
    - multiple types of applications
  - Accommodates bursty traffic

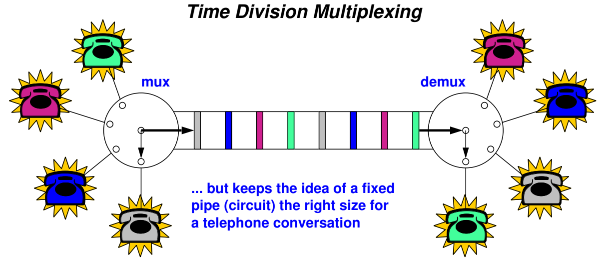
Back in the Old Days...



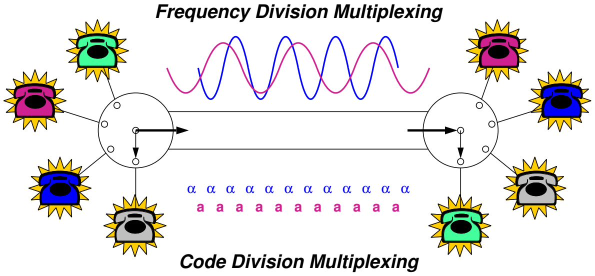
- 1920s telephony: circuits---a physical wire from one end to the other

Then Came TDM...

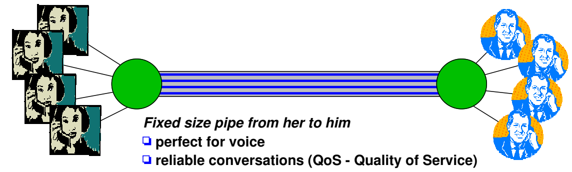
Time Division Multiplexing



And FDM and CDM...

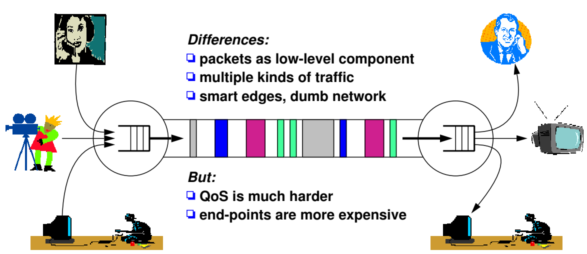


Circuit Switching



Fixed size pipe from her to him
- perfect for voice
- reliable conversations (QoS - Quality of Service)
- provisioning, good engineering
- dumb end points, smart network
- evolved for 100 years (analog to digital)

Packet Switching (Internet)



Differences:
- packets as low-level component
- multiple kinds of traffic
- smart edges, dumb network

But:
- QoS is much harder
- end-points are more expensive

Statistical Multiplexing Gain

- 1 Mbps link; Users require 0.1 mbps when transmitting; Users active only 10% of the time.
  - Circuit switching: can support 10 users.
  - Packet switching: with 35 users, probability that >=10 are transmitting at the same time = 0.0004.

Characteristics of Packet Switching

- Store and forward
  - Packets are self contained units
  - Can use alternate paths - reordering
- Contention
  - Congestion
  - Delay

Robust to Failures

- Applications should not see transient failures
- Intermediate nodes fail
  - all state at endpoints
    - datagrams
  - later: soft-state in the network and refreshed periodically (if lost, regenerated)
    - no hard-state in the network
  - *fate-sharing:* connection shares fate with the endpoints (it's okay to lose the connection if an endpoint fails)
    - state information stored at end hosts

Multiple Types of Service

- Originally just NCP, but split to {TCP,UDP}/IP soon after
- Why?
  - varying needs in speed, latency, reliability
  - not just bi-directional reliable data "virtual circuit"
- IP: best effort datagram
  - bad if link layer wants to do too much
- TCP
  - interactive, low-latency
  - bulk delivery
- UDP
  - lightweight
  - allows out-of-order to user
  - low-latency & jitter, RT possible for voice
  - reliability is biggest source of jitter

Multiple Applications

- Classes of apps
  - web
  - remote login
  - streaming audio
  - interactive audio
  - streaming/interactive video
  - file transfer (Napster, etc.)
  - computer appliances
  - distributed games
  - your app here?
- Requirements:
  - loss resilience
  - delay/jitter sensitivity
  - bursty/smooth
  - point-to-point vs. n-way (one-to-one, many-to-one, one-to-many, many-to-many)
  - numbers of sources and sinks

Multiple Kinds of Networks

- IP over X
  - compare to integrated stacks (e.g., ISO, ATM, fiber channel, Apple Desktop Bus, USB)
  - SCSI over IP?
- Requirements of X:
  - reasonable size packets/datagrams
    - but fragmentation and reassembly
  - reasonable reliability
  - addressing
- Non-requirements of X:
  - reliable, in-order, broadcast, multicast, QoS (or priority), internal knowledge of failures, speeds, or delays, etc.

Other Goals

- Distributed management
  - policy routing
  - but limitations (ex. address space portability)
- Cost effective
  - today quite cheap
  - but for small devices? for light-switch?
- Effort to deploy end-host
  - in [Clark88a]: cost of implementing stack
  - today: cost of administering machine
    - much lower today (DHCP, etc.)
    - but still lots of manual configuration

Other Goals (Cont...)

- Accountability
  - basically nothing then
    - today: PPPoE created just for authentication
- Inefficiencies
  - header too big for small payloads
  - retransmission of lost packets done at end hosts

Architecture and Implementation

- Realization: an instance of the Internet class
  - him: 1200b/s modem vs. 1Mb/s LAN
  - today: the Internet can't do X because it is Y
    - *Ex:* can't do Storage Area Networks over IP because it's too slow, so we need Fiber Channel?
    - *alternative:* build a fast Internet realization (this is why gigabit Ethernet is winning)
  - corollary: not every realization is appropriate for every app
  - also: custom stack will get last 5% of performance, but is it worth it?

TCP Features

- Features:
  - connection establishment? *Y*
  - connectionless communication? *N*
  - congestion control (not to overwhelm the network)? *Y*
  - differentiated services? *Y (sort of)*
  - duplicate packet detection? *Y*
  - flow control (not to overwhelm the receiver)? *Y*
  - loss recovery? *Y*
  - message or record boundaries? *N*
  - ordered data delivery? *Y*
  - out-of-order data delivery? *N*
  - quality-of-service guarantees? *N*
  - urgent data indication? *Y*

TCP Alternative Choices

- Stream of bytes vs. steadm of packets
  - want control over data to packet mapping, e.g., aggregate and retransmit
- Flow control
- Congestion control came later
- PSH flag
  - a weak record boundary

Other Components of IP Success

- A good, free implementation
  - BSD Unix in the mid-80's
  - compare to OSI where implementations were late
- A good API
  - BSD socket API
  - not perfect, but good
  - compare to OS's where Unix and Windows have very different APIs to open/rename/etc. files