CS 640: Introduction to Computer Networks

Aditya Akella

Lecture 1
Introduction

1

Today…

- Administrivia
- Whirlwind tour of networking!

2

Administrative Details

- Instructors
  - Aditya Akella
    - akella@cs.wisc.edu
    - Office: #7379, 890-0122
- Teaching assistant
  - Ashutosh Shukla
    - shukla@cs.wisc.edu
- Course web page
  - http://www.cs.wisc.edu/~akella/CS640/F06/
  - News, lecture notes (morning of the lecture), readings…
- Office hours
  - Aditya: T 1:30 to 3:00PM
  - Ashutosh: F 1:30 to 3:00PM

3

Goals

- Understand principles and practice of networking
- Learn how network applications work; Learn to write applications that use the network
- Hands-on approach to understand network internals
- Make you ready for a career in networking!

4

Format

- ~25 lectures
  - Readings before lectures
- 4 paper/lab homeworks
  - Loosely tied to lecture material
- 3 programming assignments
  - Group projects (groups of two)
  - Get an early start
  - Evaluation by demos
- Midterm and final
  - Actually, two midterms
  - Roughly equal weight

5

Grading

- Split
  - 35% for Programming assignments
  - 20% for Homework
  - 20% for Midterm
  - 25% for Final exam
  - Roughly equal weight in assignments and exams
- Must pass both assignments and tests!

6

Collaboration & Late Submission

- Working together is encouraged
  - Discussion of course material, debugging issues, ..
- But final submission *must* be your own work!
  - Homeworks, midterm, final
- Programming assignments: Teams of two
  - Both must contribute!
  - Collaboration, group skills
- Late penalty: 10% per day
  - No more than 2 days late

7

Today…

- Administrivia
- Whirlwind tour of networking!

8

Goal of Networking

- Enable *communication* between *network applications* on different *end-points*
  - End-points? computers, cell phones….
  - Application? Web, Peer to Peer, Streaming video, IM
  - Communication? transfer bits or information across a "network"
- Network must understand application needs/demands
  - What data rate?
  - Traffic pattern? (bursty or constant bit rate)
  - Traffic target? (multipoint or single destination, mobile or fixed)
  - App sensitivity? (to delay, "jitter", loss)
  - Difficulty: Network may not know these in the first place!
- How does the application "use" the network?
  - Peer to peer: how to find nearest host
  - Web: how to modulate sending rate? Coexist with other users/apps?

9

Defining a "Network"

- Network = nodes + links
  - Will build on this soon
- Intentionally vague. There are several different networks:
  - The Internet
  - Wisc CS network
  - Telephone network
  - Home wireless networks
  - Others – sensor nets, "On Star", cellular networks
- Our focus on Internet
  - Also explore important common issues and challenges

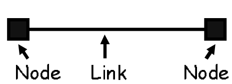
10

Common Principles ➔ Challenges for Networking

- Accommodate different geographic scopes
  - The Internet vs. home network
- Enable scale
  - CS network vs. the Internet
- Seamlessly integrate different application types
  - Email vs. video conferencing
- Independent administration and Trust
  - Corporate network – owned by one entity
  - Internet owned and managed by 17,000 network providers
    - Independent, conflicting interests

11

Network Building Block: Links

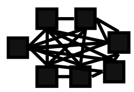


- "Physical"-layer questions
  - Wired or wireless
  - Voltage (Electrical) or wavelength (optical)
- "Link"-layer issues: How to send data?
  - Medium access – can either side talk at once?
  - Data format?

12

Basic Building Block: Links

- … But what if we want more hosts?



Wires for everybody?

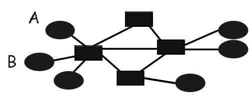
How many wires?

- How many additional wires per host?
- Scalability?

13

Key Idea: Multiplexing

- Multiplex: share network resources
  - Resources need “provisioning”
  - Grow at slower rate than number of nodes

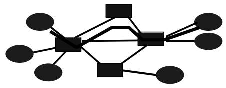


- How to share? Switched network
  - Party “A” gets resources sometimes
  - Party “B” gets them sometimes
- Interior nodes act as “Switches”

14

Circuit Switching

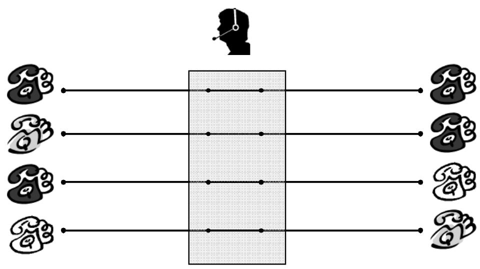
- **Source first establishes a circuit to destination**
  - Switches along the way stores info about connection
    - Possibly allocate resources
    - Different srs-dst’s get different paths



- **Source sends the data over the circuit**
  - No address required since path is established beforehand
- **The connection is explicitly set up and torn down**
- Switches use TDM (digital) or FDM (analog) to transmit data from various circuits

15

Switching in the Telephone Network



16

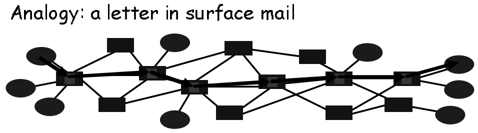
Circuit Switching Discussion

- Positives
  - *Fast and simple* data transfer, once the circuit has been established
  - Predictable performance since the circuit provides *isolation* from other users
    - E.g. guaranteed max bandwidth
- Negatives
  - How about bursty traffic
    - Circuit will be idle for significant periods of time
    - Also, can't send more than max rate
  - Circuit set-up/tear down is expensive
  - Also, reconfiguration is slow
    - Fast becoming a non-issue

17

Packet Switching

- Source sends information as self-contained packets
  - Packets have an address.
  - Source may have to break up single message in multiple packets
- Packets travel independently to the destination host
  - Switches use the address in the packet to determine how to forward the packets
  - "Store and forward"
- Analogy: a letter in surface mail

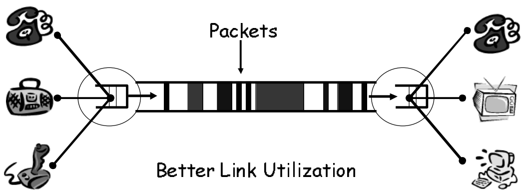


18

Benefits of Statistical Multiplexing

TDM: Flow gets chance in fixed time-slots

SM: Flow gets chance on demand; no need to wait for slot



19

Packets vs. Circuits

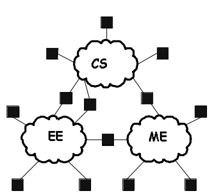
- Efficient
  - Can send from any input that is ready
  - No notion of wastage of resources that could be used otherwise
- Contention (i.e. no isolation)
  - Congestion
  - Delay
- Accommodates bursty traffic
  - But need packet buffers
- Address look-up and forwarding
  - Need optimization
- Packet switching pre-dominant
  - Circuit switching used on large time-scales, low granularities

20

Internetwork

- A collection of interconnected networks
- **Networks**: Different depts, labs, etc.
- **Router**: node that connects distinct networks
- **Host**: network endpoints (computer, PDA, light switch, …)
- Together, an independently administered entity
  - Enterprise, ISP, etc.

Internet[work]

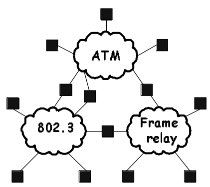


21

Internetwork Challenges

- Many differences between networks
  - Address formats
  - Performance – bandwidth/latency
  - Packet size
  - Loss rate/pattern/handling
  - Routing
- How to translate and inter-operate?

Internet[work]



22

"The Internet"

- Internet vs. internet
- The Internet: the interconnected set of networks of the Internet Service Providers (ISPs) and end-networks, providing data communications services.
  - Network of internetworks, and more
  - About 17,000 different ISP networks make up the Internet
  - Many other "end" networks
  - 100,000,000s of hosts

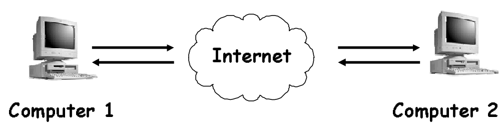
23

Challenges of the Internet

- Scale & Heterogeneity
  - 18,000+ independently administered domains
  - Thousands of different applications
  - Lots of users/hosts
  - Fast links, slow links, satellite links, cellular links, carrier pigeons
- Diversity of network technologies
  - Commercialization: different vendors, different features/formats
- Adversarial environment
  - Users/network operators could be malicious, or just buggy
- All participating networks have to follow a common set of rules
  - To avoid anarchy; but rules must be minimal and not stifle growth
- Oh, and let's make it easy to use…
  - Should support any application; minimal involvement of users…

24

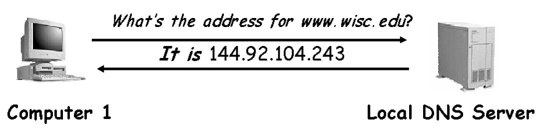
Some Key "Internet" Design Issues



Need:
(1) naming,
(2) addressing and
(3) routing
(4) ...

25

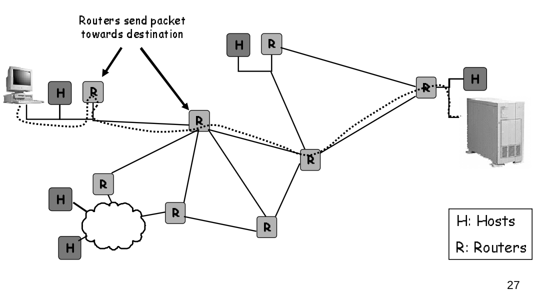
Key Issues: Naming/Addressing



Translates human readable names to logical endpoints

26

Key Issues: Routing



27

Key Issues: Network Service Model

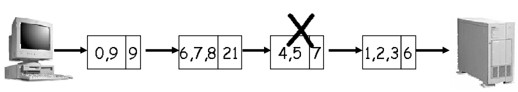
- What is the *service model?*
  - Defines what to expect from the network
  - ***Best-effort:*** packets can get lost, no guaranteed delivery
- What if you want more?
  - Performance guarantees (QoS)
  - Reliability
    - Corruption
    - Lost packets
  - In-order delivery for file chunks
  - Etc…

28

What if the Data gets Corrupted?

Problem: Data Corruption

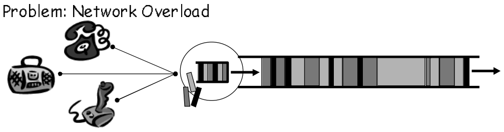
Solution: Add a *checksum*



29

What if Network is Overloaded?

Problem: Network Overload



Solution: Buffering and Congestion Control

- **Short bursts: buffer**
- **What if buffer overflows?**
  - Packets dropped
  - Sender adjusts rate until load = resources → “congestion control”

30

What if the Data gets Lost?

Problem: Lost Data

Solution: Timeout and Retransmit

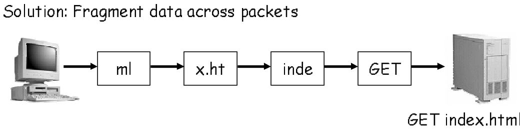
31

What if the Data Doesn't Fit?

Problem: Packet size

- On Ethernet, max packet is 1.5KB
- Typical web page is 10KB

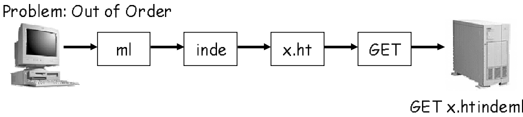
Solution: Fragment data across packets



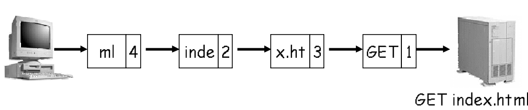
32

What if Data is Out of Order?

Problem: Out of Order



Solution: Add Sequence Numbers



33

Meeting Application Demands

- Sometimes network can do it
  - E.g., Quality of Service
    - Benefits of circuit switching in packet-switched net
    - Hard in the Internet, easy in restricted contexts
    - Lecture 20
- OR hosts can do it
  - E.g., end-to-end *Transport protocols*
    - TCP performs end-to-end retransmission of lost packets to give the illusion of a reliable underlying network.
    - Lectures 16-19

34

To Summarize…

Networks implement many functions

- Links
- Sharing/Multiplexing
- Routing
- Addressing/naming
- Reliability
- Flow control
- Fragmentation
- Etc….

35

Next Lecture

- Split of functionality
  - Across protocol layers
  - Across network nodes/entities

36