Course Goals

- Provide the background for understanding, appreciating and performing research in networking
  - Provide an understanding of the principles and tradeoffs involved in building a heterogeneous large scale network such as the Internet
  - Provide an understanding of the important protocols and applications that are shaping the Internet
  - Overview active research areas and discuss open research problems
- Provide training and practice in reading and evaluating research papers
- Provide experience in performing research

Course Mechanics/Grade breakdown

- Lectures and discussion
  - 5% of your grade based on class participation (attendance, and participation in discussions)
- Homeworks and Critiques (20%)
- Project/Programming assignments, (30%)
- Two midterms (25%) and a final (20%)
- Grade is relative to the rest of the class
- Will find ways to make your better work count for more
  - Will drop your worse critique, and try to factor your better work for a higher weight

CS528: Computer Networks and Data Communications

CS 528 – Computer Networks and Data Communication
Instructor: Dr. Nael Abu-Ghazaleh
CS Dept., SUNY–Binghamton Fall '07

- Agenda for today:
  - Go over syllabus: Policies, Grading, Odds and Ends
  - Intro and motivation
  - Requirements of a scalable general purpose communication network

Administrivia

- Instructor: me (nael@cs.binghamton.edu).
  - Office hours T 3-4pm, Th. 1-2pm T-12 EB, third floor
  - If you cannot make these times, email me for an appointment
- Class webpage: http://www.cs.binghamton.edu/~cs528
  - Will try to make class notes available early, but no promises
Balance Between Graduate and Undergraduate Content

- Will not assume you have had undergraduate networking, but

- But this is a graduate class
  - Will do quick reviews of core material; you will have to read some on your own if you don't have the background
  - Hitting the balance between core and research topics
    * Textbook (which is excellent) for core topics
    * Papers to augment the core material; you will read some, and I will cover additional ones
  - Exam problems and Projects from both core and research topics
  - In general, I am trying to make the course a little easier than previous years

Topics We will try to Cover

- Quick review of lower layers (physical, link layer, MAC)
- Internetworking
- Reliable Data Transfer
- Routing (Unicast and Multicast) and Traffic Engineering
- Congestion Control
- Quality of Service (IntServ/Diffserv, scheduling and queue management)
- Mobile Networking
- Network Security
- Applications
- Overlay Networks
- Peer-to-Peer networking
- Next Generation Internet
- Measurement, High Performance Networking, Router Architecture, ATM, ...
- More than will fit in a single semester!

Reading and Critiquing Papers

- Will rely on research papers for many of the topics we cover
- You have to read some papers, and I encourage you to read others
- Critiques will be used for the required papers (6 or 7)
  - A series of questions about the paper
  - Due before class
  - Will require only around 6 for the whole semester; will throw away your worse one
  - Class participation is expected (worth 5% of your grade)
  - Exams and quizzes will be partially based on the papers (only what we cover in class)

Reading Research Papers

- Several types of papers
  - Classic papers for mostly historical value (will not be required)
  - Fundamental issues/Position papers
  - New ideas papers
  - Analysis papers
- Average of one paper to read per week
  - Critique not required for all (roughly every other week)
  - I will cover additional papers that you won't be required to read or critique
First Project Assignment Next Week

- Learning NS-2: Please go through the two tutorials on the class webpage
- Assignment is straightforward, but there is a lot of stuff that you need to learn
  - Downloading/installing NS-2; make sure you have space for that or ask me for an account on a machine that does
  - Tcl/Otcl
  - awk or perl or python or your favorite scripting language to process results
- Help will be forthcoming

Projects

- A common start, then two tracks
- First project: using the NS-2 simulator, building and running some scenarios
- Second project will also be common
- After that, you can pick a standard third project or research project track
  - Standard project is well defined, non-researchy implementation project
  - Research project is a self defined (with my help) research study—Only if you are experienced/research minded

Reading for Next Week

- Required: The End to End Argument in System Design, Saltzer et al
- Recommended: Tussle in Cyberspace: Defining Tomorrow's Internet, Clark, Sollins, Wroclawski and Braden, SIGCOMM 2002.
- Recommended: The Design Philosophy of the DARPA Internet Protocols, David Clark, SIGCOMM 1988 (optional; will cover parts in class)
- Downloadable from class webpage

Some Advice/Hints

- Start on projects early (doubly true if you take the research project option)
  - Statistical correlation between when you start and your grade
  - Some groups that did not take it seriously in the past did not finish
- Keep up with the reading; classes will be much more useful to you if you do
Computer Networks – why

• Why is networking interesting?

• What's hard about networking?

Really Large Scale

Expectations

• This course is about:
  – General purpose communication networks
  – Fundamental concepts and principles; challenges and tradeoffs
  – Internet Perspective
  – Networking System Software (with overview of hardware)
  – Engineering of a Scalable Network

Expectations

• This course is NOT about:
  – Specialized networks (e.g., telephone, CATV, PCS)
  – Survey of existing protocol standards
  – OSI perspective (“layering is a good slave but a poor master”)
  – Network Hardware Design
  – Queueing Theory
Distributed

- Decentralized Control; how to manage updates in state?
- Long communication latency
- Partial failure possible

“A distributed system is one where I cannot do my work because some computer has failed that I’ve never even heard of” – Leslie Lamport

Requirement Definition

- Requirements and Constraints depend on your perspective:
  - Network users want the network to provide services that their applications need; e.g., guarantee that each message will be delivered in order, without errors, and within a pre-defined delay
  - Network designers want a cost-effective design; e.g., network resources are efficiently utilized and fairly allocated to users
  - Network providers want a system that is easy to administer and manage; e.g., faults can be easily found, system can be hot-swapped, and easy to track usage of users

- What do these requirements translate to?

Heterogeneous

- Homogeneous Network: the phone system
  - Designed for making phone calls
  - Well known behavior (call properties, bandwidth, service constraints...)
  - Small number of technologies
  - Centralized (or nearly centralized) control

- Heterogeneous Network: the Internet
  - Many types of applications
  - Few Underlying Assumptions (good and bad)
  - Wide variety of competing interacting technologies
Connectivity – Summary/Discussion

- “A network is two or more nodes connected by a direct link, or two or more networks connected by one or more nodes”

- Hosts connected directly or indirectly
  - Need global addressability
  - Need routing ability
  - Unicast/Broadcast/Multicast

- Network Edge vs. Network Core; does it make a difference?

Requirement I: Connectivity

- Building blocks:
  - Links: copper wires (coax cable, twisted pair), optical fiber, ...
  - Nodes: general-purpose workstations, dedicated routers, ...

- Direct Links:
  - Point-to-point: dedicated link connecting two nodes
  - Multiple access: many nodes access a shared “broadcast” medium
  - other?

Requirement II: Efficient Resource Sharing

- Must share the network resources (nodes and links) among multiple users

- Common multiplexing (sharing) techniques:
  - Time Division Multiplexing (TDM)
  - Frequency Division Multiplexing (FDM)
Switching Strategies

• Circuit Switching:
  - send/receive a bit stream over a dedicated circuit
  - Three phases: call, send data, hang up
  - Static allocation of resources
  - Example: Telephone network

• Packet switching:
  - A packet is the unit of switching (what is a packet?)
  - Usually store-and-forward; virtual cut-through in high performance LANs
  - Dynamic sharing of bandwidth (statistical multiplexing)

Requirement III: Functionality

• Network should provide abstractions (services) that allow application programs to communicate in a meaningful way
• How to decide what to support in the network?
  - A hardware/software tradeoff?
  - What are some useful communication abstractions?
  - Think of examples of other desirable functionality...

Example: common process-to-process channel abstractions

• Request/Reply – client server applications such as file access and web page browsing
• Message stream – example, video application
  - video: sequence of frames
  - resolution: 1/4 TV size image = 352 x 240 pixels
  - 24-bit color: frame = (352 x 240 x 24)/8 = 247.5 KBytes/frame
  - 15-30 frames/sec for usable quality video; at 30 fps need 7.5Mbyte/sec
  - video-on-demand; video-teleconferencing
  - Think of ways these applications are different; can a single network effectively support both types of application?
• Example of additional requirements: reliability; security; guaranteed level of performance
Reliability

What can go wrong in a network?

- Bit-level errors (e.g., due to electrical interference on the wire)
- Packet-level errors (e.g., due to congestion)
- Link and node failures
- Malicious interference
- Messages are delayed
- Messages are delivered out-of-order, lost or duplicated
- Messages are delivered to the wrong receiver

What level of protection should the network provide?

Functionality – Summary

- Many useful abstractions could be required by users
- Other functionality such as reliability and security
- Different performance levels required
- What should the network support?
- Balance application requirements and technology limitations
  - The semantic gap: filling the gap between the application requirements and the underlying network technology
  - Computer architecture analogy
  - The End-to-End Principle in System Design (Saltzer et al)—our first reading assignment

SUNY-BINGHAMTON – CS528 FALL ’07 LEC. #1

Requirement IV: Performance

- Bandwidth (throughput)
  - Amount of data that can be transmitted per time unit
  - Link vs. end-to-end
  - Notation
    * KB = $2^{10}$ bytes
    * Mbps = $10^6$ bits per second
  - Bandwidth related to bit-width

- Latency (delay)
  - Time it takes to send message from one point to the other
  - Sometimes we are interested in round-trip time (RTT)
  - What determines the latency?
    - Latency = Propagation + Transmit + Queue
    - Propagation is the distance / speed of light
    - Transmit = size / bandwidth
    - Queue includes delays in link and network
  - Bandwidth not relevant if size = 1 bit
  - Overhead, which can dominate if the distance is small

SUNY-BINGHAMTON – CS528 FALL ’07 LEC. #1

Performance (cont’d)
Interaction of Latency and Bandwidth

What contributes more to the performance of required operation?

• Small messages: 1ms vs 100ms latency dominates 1Mbps vs 100Mbps bandwidth
• Large messages: 1Mbps vs 100Mbps dominates 1ms vs 100ms latency

Delay Bandwidth product defines amount of data 
in flight – why is this important?

Application needs: bandwidth (burst vs stable), latency (jitter = variance in latency)

Example

Consider a network consisting of three hosts, A, B, and C. A is connected to B using a 40 kbit/sec, 10 msec delay link. B is connected to C using a 20 kbit/sec link with 20 msec delay. A sends a 2000 bit packet to B, which immediately sends a 2000 bit packet to C. If C receives the packet from B before A and sends a signal packet to A indicating that it has received the packet, how many packets has A sent?

High Speed Networks

Applications buy more bandwidth.

High speed networks – bandwidth is increasing

Think about the implications of always buying more bandwidth.

Latency is much harder to improve (limited, for example, by the speed of light)

Other Performance Metrics?

What are some other performance metrics of interest?