CS 352
Internet Technology

Lecture 1.1: Introduction
http://www.cs.rutgers.edu/~sn624/352
Srinivas Narayana
The Internet is an exciting place
The Internet has transformed everything

• How we communicate with other humans
• How we learn what’s going on in the world
• How we learn and acquire knowledge
• How we transact and do business
• How we entertain ourselves
• How espionage and war is conducted

• In short how we live, especially through a pandemic.
Internet growth

1995
35MM+ Internet Users
0.6% Population Penetration

2014
2.8B Internet Users
39% Population Penetration

2020
4.8B users
(61% of the world’s population)

https://www.broadbandsearch.net/blog/internet-statistics
Evolution of Internet applications

- **1992**
  - ftp
  - Web
  - email
- **1996**
  - chat
  - Games
  - IM
  - Yahoo!
- **2000**
  - news
  - Blog
  - Search
- **2004**
  - Music
  - itunes
  - Games
  - search
- **2008**
  - Wikipedia
  - Craigslist
  - Youtube
- **2010-now**
  - Twitter
  - airbnb
  - Uber

Text-heavy | Multimodal media | Augment physical world
We relied on the Internet to work

Data shows number of daily sessions in the US over a period in 2020. Source: nytimes
We relied on the Internet to “play”!

Data shows number of daily sessions in the US over a period in 2020. Source: nytimes
Threats on the Internet are growing, too

Source: CloudFlare blog
Internet Technology: This course

• The study of how the Internet (and other large networks) are designed.

• We will study the principles that make the Internet as successful an artifact as it is.

• The Internet is an example of a computer network
What is a network?

• Carrier of information between two or more entities

• Entities may be hosts/endpoints (used interchangeably)
  • your laptop, cell phone, etc.

• Entities may also be devices in the middle of the network
  • For example, your WiFi router

• The interconnection between entities is any physical medium capable of carrying information: we call physical media links
  • copper wire, lasers (over optic fibre), microwave, cable (coax), satellite link, wireless link (cellular, 802.11, bluetooth)
A single link multiple access network

- Send bits of data in **packets** or frames
- How do we differentiate among many receivers?
- Every endpoint as a link level **address**: also called a *MAC* address
- Packets have a destination address on them
- However, can’t have every computer in the world on the same link!
  - Physical limits on power / distance over which info travels over a single link
A single link multiple access network

• Even on a single link, you need to worry about a few things:
  • Converting digital data to physical signals over the medium (encode/decode)
  • How do we decide who speaks? (medium access control problem)
  • Detecting and correcting errors
A multi-link network

- Connect multiple links via routers
- Need to figure out how to move packets from one host to another host, e.g., how to reach google.com from your laptop
- Known as the routing problem
- Key Q: How should packets be moved from A to reach B?
In general, networks give no guarantees

- Packets may be lost, corrupted, reordered, on the way to the destination
  - *Best effort* delivery

- Advantage: The network becomes very simple to build
  - Don’t have to make it reliable
  - Don’t need to implement any performance guarantees
  - Don’t need to maintain packet ordering
  - Almost any medium can deliver individual packets
    - RFC 1149: “IP Datagrams over Avian Carriers”

- The early Internet thrived since (transient) disruptions are okay
Guarantees for applications

• How should endpoints provide guarantees to applications?

• Transport software on the endpoint oversees implementing guarantees on top of an unreliable network

• Need to solve the reliable data delivery problem

• For some applications, also need ordered delivery
Sending data into a multi-link network

• How quickly should endpoints send data into a network?

• Known as the congestion control problem

• Congestion control algorithms at source endpoints react to remote network congestion. Part of the transport sw/hw stack.

• Key question: How to vary the sending rate based on network signals?
Sending data into a multi-link network

• How should a router transmit packets when network resources are scarce?

• Known as the packet scheduling problem

• Key question: which packet to transmit over a constrained network link, and when?
  • Related: the buffer management problem
Components of a network: Review

• Link
  • Communication links for transmission

• Host/Endpoint
  • Computer running applications of end user

• Router
  • Computer for routing packets from input link to another output link

• Network
  • A group of hosts, links, routers capable of sending packets among its members
CS 352
Course Logistics

Lecture 1.2
http://www.cs.rutgers.edu/~sn624/352
Srinivas Narayana
About us: Management

• Faculty Instructor: Srinivas Narayana
  • [http://www.cs.rutgers.edu/~sn624](http://www.cs.rutgers.edu/~sn624)
  • [sn624@rutgers.edu](mailto:sn624@rutgers.edu)
  • Office hours on Zoom (link on Canvas) Mondays and Thursdays at 9 am ET or by appointment
  • Class is fully remote and asynchronous. Lectures every Mon+Thu

• Recitation sections 1 (Shuxin Zhong) and 2 (Ari Hayes)
  • Recitations are also fully remote and asynchronous
  • TA office hours to be announced; recitation

• Course info

• This course uses Canvas and Piazza (linked from Canvas)
Class philosophy

• We want you to learn and to be successful

• Ask questions on Piazza

• Attend office hours regularly to clarify material
  • 6 hours of office each week among all 3 instructors put together

• In summary, be proactive. Interact with us and with your fellow students and support each other
Goals

• Understand the basic design principles of computer networks

• Understand how the Internet works
  • Principles, architecture, protocols

• Text: “Computer networking, a top-down approach,” by James Kurose and Keith Ross
Course Assessments

• 30% programming projects
• 30% weekly quizzes
• 20% mid-terms (2 of 10% each)
• 20% final exam

• Full schedule of assessments available at https://www.cs.rutgers.edu/~sn624/352/syllabus.html
Programming projects (30%)

• Three programming projects (3 * 10%)
• Work in the same group of two students throughout semester  
  • Only change groups with the discretion of instructor
• Programs and short write-up required
• Background needed to get started  
  • Python (211, 214 level)  
    • Get comfortable using data structures (tuples, arrays, dictionaries)  
  • Unix (login, permissions, gcc)
Programming projects (30%)

• Hand-in programming projects via Canvas
  • Please get them in on time

• Failure to meet the due date will result in maximum 20% credit for that project for both team members.

• You must turn in all programming projects, even if they are delayed, to pass this course. Not turning in a project automatically implies a failing grade for all team members.
Weekly quizzes (30%)

• 8 weekly quizzes over the semester.
• Take them on Canvas any 40-min period over a week
• Quizzes will be announced when available
• Quizzes are closed book. Calculators are allowed

• No make-ups. You can drop 2 of the lowest scores among the 8

• Quiz schedule at  
  https://www.cs.rutgers.edu/~sn624/352/syllabus.html
Mid-Terms (2 * 10%) and Final (20%)

• Two mid-terms (1.5 hours each) and a final exam (2.5 hours)
• Take any time over a window of 3 and 5 days (resp.). You can find the exam windows in the schedule
• **Open-book:** but only use lectures, textbook, and your own notes.
• No collaboration
• No looking for answers on the Internet.
• You must notify me at least 2 weeks before the final if you need to take a makeup
24/7 Grading Policy

• You may not dispute a grade or request a regrade before 24 hours or after 7 days of receiving it.

• You may contact us if you have a legitimate regrading request…

• After 24 hours of receiving the grade: Please take the time to review your case before contacting the instructors.

• Before 7 days have elapsed since you received the grade: we don’t want to forget what the test was all about!
Academic integrity

- I encourage you to study and prepare in groups
  - Share materials: it’s helpful for everyone
- All written & programmed work you turn in must be your own
- Please, no cheating on projects and exams
  - We reserve the right to…
    - Run code similarity detectors on the projects & code review
    - Scrutinize exams for copying
- It’s much easier to just do the right thing
- Read the course academic integrity policy at https://www.cs.rutgers.edu/~sn624/352/index.html#academic-integrity
Help, Accommodations, etc.

• We’ll be happy to try and accommodate any requests that better support your learning

• Don’t hesitate to contact the course staff with any requests.

• sn624@cs.rutgers.edu