552: Computer Networks

Srinivas Narayana

Fall 2019 (MTh 8.40—10 AM in SEC 216)

RUTGERS
UNIVERSITY | NEW BRUNSWICK

The Internet is an exciting place

The Internet is an exciting place

- A research experiment that escaped the lab
  - ... to become a global communication infrastructure

- Ever-expanding reach
  - Now: > 3B people online
  - Future: Still potential for more users, machines, and applications

- Consistent innovation with low barriers
  - Apps: streaming video, smart grids, telesurgery, …
  - Technologies: Ethernet, cellular, optics, human-body (!), …

The Internet has transformed everything

- How we communicate with other humans
- How we learn and acquire knowledge
- How we transact and do business
- How we entertain ourselves
- How we govern ourselves
- How warfare is conducted (!)

- Computer Networking is the study of how the Internet (and other inter-networks) are designed

What is a network, anyway?

- Carrier of information between two or more entities

- Entities may be hosts: your laptop, cell phone, etc.
- Entities may also be devices in the middle of the network
  - For example, your WiFi router

- In this course, we will typically refer to communicating entities as *endpoints*

- The interconnection between entities is any physical medium capable of carrying information
  - copper wire, lasers (over optic fibre), microwave, cable (coax), satellite link, wireless link (cellular, 802.11, bluetooth)

Why should we study networking?

- Utility: tangible real-world impact
  - Easy to measure and build things
  - Artifacts can go a long way
  - *You* can build something that *you* want to use
    - … that other people then build on

- Examples
  - BitTorrent: a student at University of Buffalo
  - World wide web (WWW): one researcher at CERN
  - Bitcoin: no one knows with certainty who created it

Why should we study networking?

- Intellectual rewards: interdisciplinary & evolving problems
  - So much to learn and apply from other fields
  - Many principles to contribute to other fields

- Examples
  - Convex optimization: flow of traffic through an ISP network
  - Formal verification: ensure admin intent is reliably implemented
  - End to end argument (“don’t redo useful work”): all comp systems

Why should we study networking?

- Academic impact: a young and relatively immature field
  - *You* get to decide what the field looks like in a few years!
  - Several highly cited CS papers are networking papers
  - At least 2 Turing awards (Cerf&Kahn'04, Berners-Lee'16)

- Significant opportunities lie ahead!
  - *Define* the most interesting problems
  - *Build* your ideas using freely and openly available software
  - *Deploy* your ideas and technology over cheap platforms!

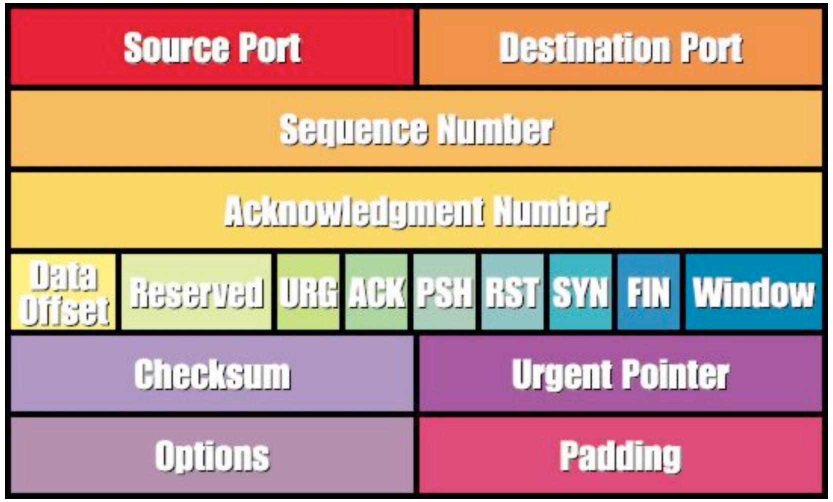
So, how *should* we study networking?

So, how *should* we study networking?

- The soup of acronyms? IP, TCP, HTTP, RSVP, …

So, how *should* we study networking?

- The soup of acronyms? IP, TCP, HTTP, RSVP, …
- The stack of layers and headers? port, option, flags, …



So, how *should* we study networking?

- The soup of acronyms? IP, TCP, HTTP, RSVP, …
- The stack of layers and headers? port, option, flags, …
- The suite of tools? ping, traceroute, wireshark, …
- The wire protocols? SCTP, DNS, SSH, ...
- The middle boxes? CDN, firewall, NAT, proxy, ...

These categorizations turn out not to be very helpful.

So, how do we do networking research?

What we will study in this course

- Principles used to build computer networks and services
  - How communication infrastructure is designed
  - … to *effectively* meet application and network designer goals
  - … in ways that can be composed constructively

Mix of classic and recent research papers

- Ways to partition functionality
  - Among machines: end points, switches, middleboxes, ...)
  - Among modules inside a machine: OS, apps, hardware, ...
  - Among entities: ISPs ("core"), users ("edge"), …

Course syllabus

1. Foundations: what are the different kinds of networks?
2. How your data is processed at the network end points (edge)
3. How your data is processed in between the end points (core)
4. How is the network shared efficiently and fairly among many end points? (congestion control and scheduling)
5. How do network operators verify that their intent is implemented reliably? (network verification)
6. What lies ahead? (frontiers)

Quick class introductions

Go around and say:

(1) your name, class/program/affiliation

(2) why are you interested in taking this class? what do you expect out of taking it?

Course logistics: Structure

- Reading papers
  - Understand, summarize, and critique papers you read
  - Find ways to improve and generalize the ideas in those papers
- In the class room:
  - Some background material
  - Spend lots of time on discussion around the papers
- At home:
  - Answers to review questions
  - Final project

Course logistics: Assessments

- Class participation (20%)
  - Show up
    - Preferably on time
  - Speak up -- there’s no harm in being wrong
  - Ask questions
  - Debate the paper’s and each other’s ideas
    - … with your own proposals if possible

- Keep conversations honest but professional
  - If you disagree, explain why
  - Attack viewpoints, approaches, methods, results, etc.
  - Do not attack the person: no *ad hominem*
  - Read “how to disagree” in the optional readings

- Try to keep conversations related to the class material

Course logistics: Assessments

- Class project (40%)

- Examples:
  - Solutions to open but narrow research questions
  - A tool that you knew you wanted but never found time to work on
  - Reproduction of results from one of the papers in this class or outside
  - Evaluating a new aspect of an existing system
  - Porting an existing system to a different platform
  - New software that makes further research possible or easy

- *Must* involve a significant course-related programming component

- I will act as a resource to consult throughout the project
  - Scoping out, overcoming obstacles, pointing to good tools, etc.

Course logistics: Class project

- Teams of 2—3 or (less preferred) work alone
- Submit initial proposal (1 page) by Friday 09/27
- Mid-term project presentation (~5—10 min per group on 10/31)
- 6-page project write-up due end of semester (Friday 12/13)
- Final presentation or poster session (tentatively 12/9)
- More information & project ideas next week

Course logistics: Assessments

- Review questions (40%) intended to:

- Test your understanding
- Identify reusable principles: stretch your imagination about where the same or similar ideas could be applied
- Get you to think critically about the weaknesses of the approach
- Help you understand what it means to improve the state of the art
- Help you appreciate research results

Course logistics: Review questions

- You are welcome to look up the paper while answering
- Sakai quiz. Answers due 10 PM the day before every lecture
  - There is no time limit on the Sakai quiz
- You are welcome to discuss and collaborate. However,
  - **all written work must be your own (no exceptions)**
  - Name all your collaborators in your review
- Read the CS academic integrity policy: https://www.cs.rutgers.edu/academic-integrity/introduction

How do you read and critique a research paper?

You’ll spend a lot of time reading

- Graduate courses
- Reviewing conference papers
- Research literature
- Understanding a closely related paper deeply
- Staying broadly educated
- Branching out into closely related areas
- Learning how to write better papers

- It’s worth knowing how to read *effectively*

There is no magic

- Reading effectively takes a lot of time and effort
- I’ve been reading research papers for about 10 years now
    - And I still sometimes spend hours or even an entire day when I closely read a paper
- You *will* get better at it over time
- There are a few tricks…

Keshav's top-down, 3-pass approach (1/3)

- Title, abstract, introduction, section titles
- Category of paper: analysis? position? new capability? measurement?
- Context: what broader body of work is this related to?
- Correctness: are conclusions plausible? are assumptions valid?
- Contributions: reusable principles? previously unknown insights?
- Clarity: Can you understand it?

Keshav’s top-down, 3-pass approach (2/3)

- Read in more detail to understand main technical ideas
  - Understand graphs and illustrations

- You must be able to summarize the paper’s main technical contribution and supporting evidence to others!

- Ignore highly detailed aspects
  - Proofs
  - Appendices
  - Mark relevant but unknown references for later reading

Keshav’s top-down, 3-pass approach (3/3)

- Re-implement the paper’s solution from scratch
- … starting from its assumptions

- Identify its innovations

- Identify implicit or flawed assumptions

- Identify hidden flaws in the solution

A strategy that I personally use

- Work through examples
    - … even if the paper doesn't show them
    - Nothing clarifies an idea like working through simple but concrete examples

Tim Roscoe: How to review a paper

- What do you think are the technical contributions of the paper?
    - What’s new about the work?
    - What are the reusable principles and insights?

- What did you find to be cool or interesting about it?
    - Specific design techniques?
    - A method of measurement, evaluation, or something else?
    - A direction that may lead to interesting follow-up work?

Tim Roscoe: How to review a paper

- Are there technical flaws in the solution or its evaluation?
  - All research is flawed or incomplete in some way!
  - Do the flaws fundamentally invalidate the claimed contributions?

- How can you improve the clarity of the paper?
  - Restructure sections?
  - Rephrase specific sentences?
  - Typos?

Course logistics summary

- Class web page: https://www.cs.rutgers.edu/~sn624/552-F19/
- Sakai web site activated
- Piazza activated to continue discussions outside the classroom

- Instructor: Srinivas Narayana (srinivas.narayana@rutgers.edu)
  - Office hour in CoRE 312 on Mon 2—3 PM or by appt

- No official class textbook. Follow the assigned readings
  - I will provide supplementary readings by student request

- Prerequisite: Undergrad computer networks (352) or by permission
  - Speak to me if you're in doubt

Computer Networks

RUTGERS
UNIVERSITY | NEW BRUNSWICK

A single link multiple access network

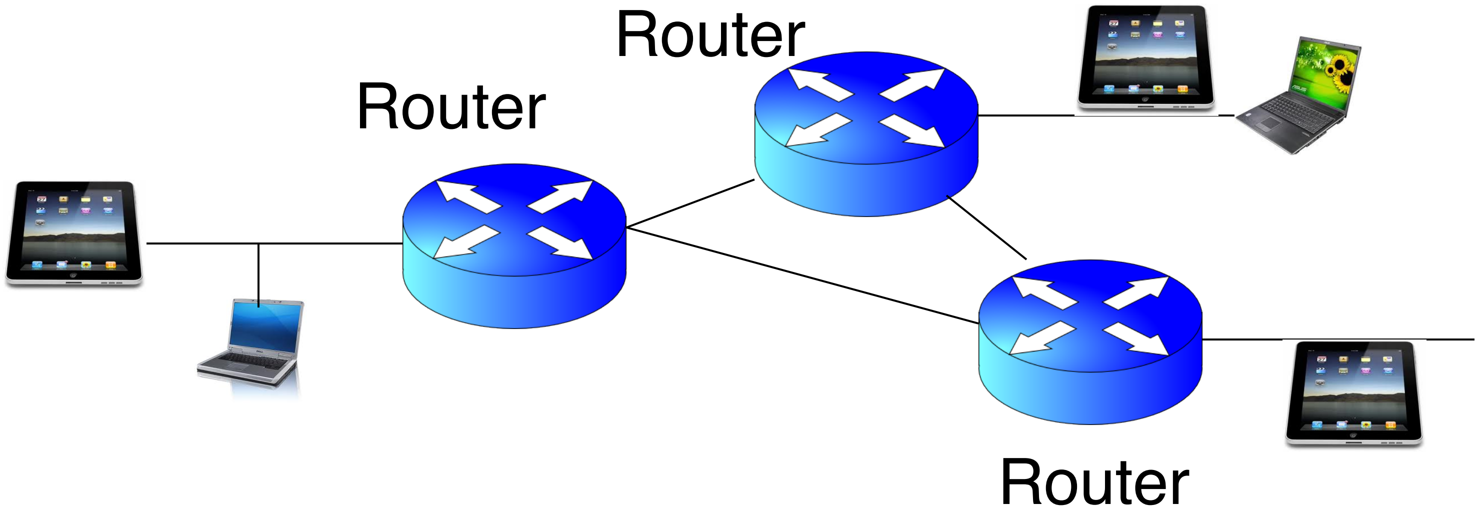
- Send bits of data in *packets* or *frames*
- How do we differentiate among many receivers?
- Every host as a link level *address*: also called a *MAC* address
- Packets have a destination address on them

- However, can't have every computer in the world on the same link!

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
- Known as the *routing* problem
- How should packets be routed from endpoint A to endpoint B?

Multi-link networks provide no guarantees

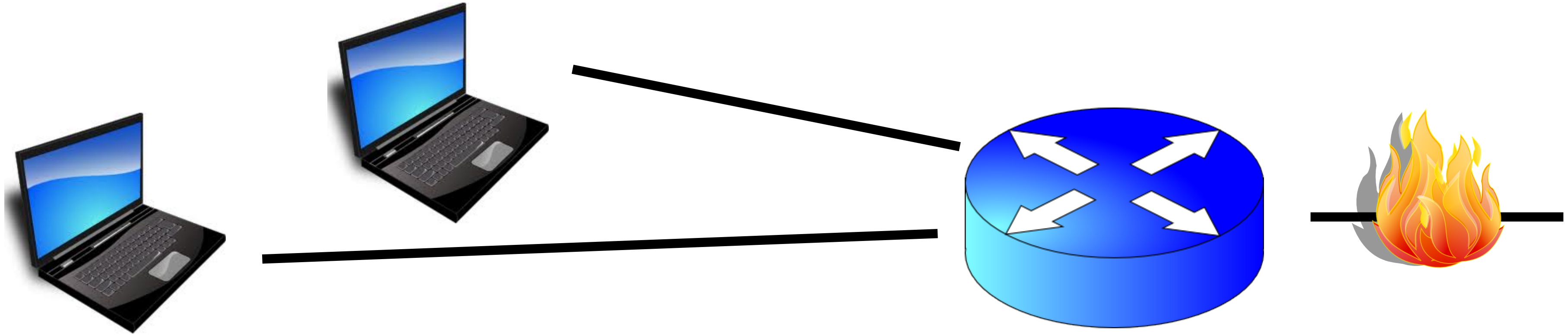
- Packets may be lost, corrupted, reordered, on the way to the destination
  - Best effort delivery

- 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

Sending data into a multi-link network

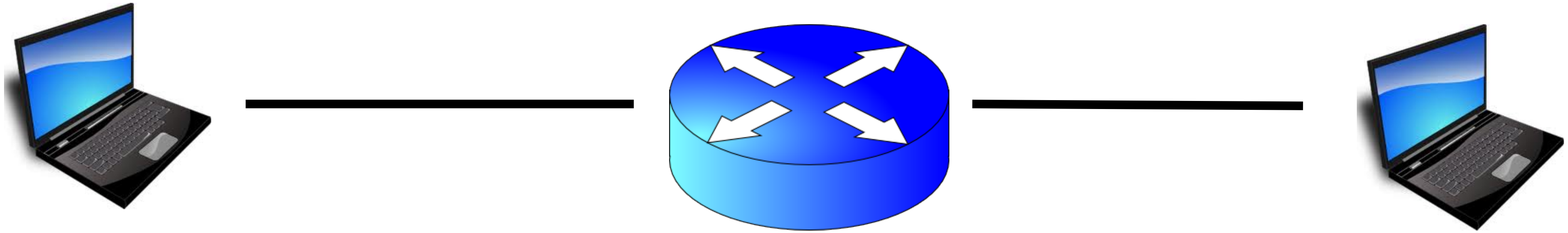
- How quickly should endpoints send data into a network?



- Known as the congestion control problem
- Congestion control algorithms at source endpoints react to remote network congestion
- How to vary the sending rate?
  - Over 30 years of research!

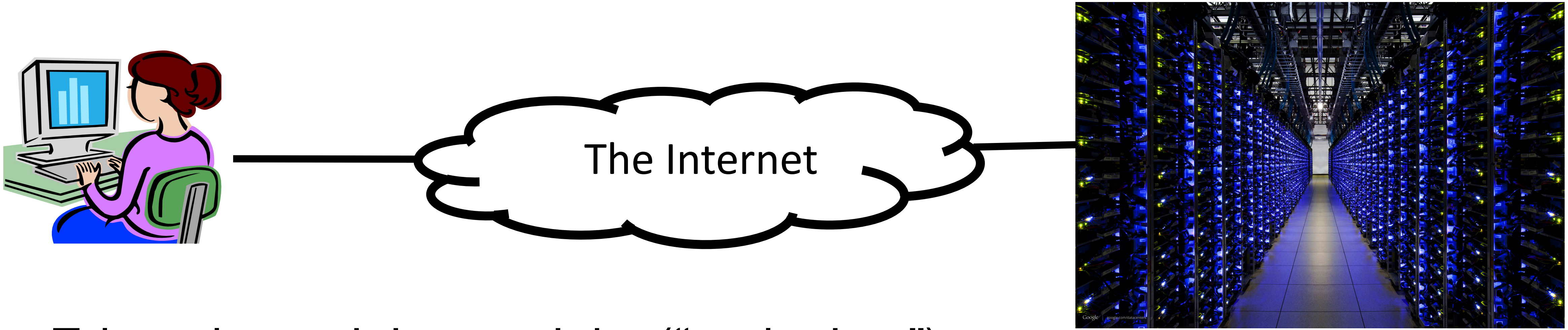
What about guarantees?

- How should endpoints provide guarantees to applications?



- Transport software on the endpoint is in charge of implementing guarantees on top of an unreliable network
    - Reliability
    - Ordered delivery
    - Packet delay not exceeding 50 ms?

Edge and *core*: a useful distinction



- Edge: data origins or sinks (“endpoints”)
  - Your laptop, mobile phone, Google’s servers
- Core: machines processing & transmitting data
  - Your WiFi router, Rutgers’s firewall, Verizon’s routers
- Varies by context: one person’s core is another’s edge

For next lecture

- Read the assigned paper
- “The Design Philosophy of the DARPA Internet Protocols”

- Answer review questions on Sakai by Sunday 10 PM