CSC7970 – NEXT-GENERATION NETWORKING

INTRODUCTION AND TCP/IP

Instructor: Susmit Shannigrahi
sshannigrahi@tntech.edu

Tennessee TECH

Welcome

- Class website: https://csc7970.github.io/
  - Syllabus
  - Grading policies
  - Homework and assignments

- Instructor: Susmit Shannigrahi
  - Office hours: By appointment

Grading

- Coding Project + Presentation – 35%

- Paper Presentation and Discussions – 30%
  - Each student will present two papers

- Exams/Quizzes – 35%
  - 10 Quizzes over the semester
  - No midterm or final
    - Focus on learning instead

Policies

- **Participation is mandatory**
- **No late submission(s)**
- **No cheating etc….**

Cheating Policy

- **If you cheat, you will fail the class!**
  - Regardless of what you cheated in
  - Don’t do it.

- **You will also be reported for academic misconduct**
  - http://catalog.tntech.edu/content.php?catoid=18&navoid=3312

Programming Assignments

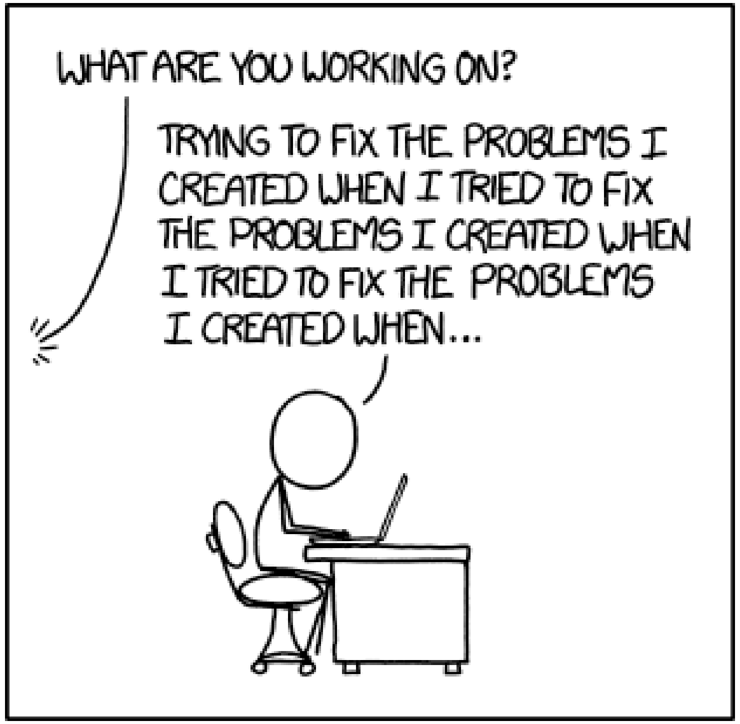
- Must run on Google Cloud Vms – Ubuntu-18.04

- Group projects – think about a problem
  - We will solve some using NDN, some using SDN/IP
  - At the end, we will compare notes

- C/C++/Python
  - Other languages, come talk to me.

How to read papers?

- Read a few papers about how to read a paper
- [How to Read a Paper]
  - https://www.cs.colostate.edu/~cs557/papers/Kes07.pdf
- [Efficient Reading of Papers in Science and Technology]
  - https://www.cs.colostate.edu/~cs557/papers/Han99.pdf



Deliverable in Next Class

- Need a volunteer for paper presentation next week!
- Shoot for a 40 minute presentation – we will discuss it for the next 20 minutes
  - **Tuesday** - J.H. Saltzer, D.P. Reed and D.D. Clark, End-to-End Arguments in System Design, ACM's TOCS 1981
  - **Thursday** - D .D. Clark, “The Design Philosophy of the DARPA Internet Protocols”, ACM SIGCOMM Computer Communication Review, 1988

Class goals

- Understand the design and the rationale behind the Internet design
  - Why the Internet was designed the way it was designed?
  - How this design has evolved over time?•Is the current design is sufficient today?
- What are the advantages and the problems of this design?
  - “Whys” and “hows” rather than specific low-level details!
  - ➢ What are the directions of designing the Internet of the future
  - ➢ Do we really need to redesign the Internet?•If so, why/how it should look like?
- How do they get applied to networked applications today?

Why bother? (other than because my advisor told me to)

- My Interest is in ML/AI/HPC/Whatever, why should I care about networking?
  - Almost all applications need networking, and increasingly more so
- How do we make applications more efficient?
  - Download a Terabyte data and run simulations
  - Write small code to invoke simulations remotely
- Networking is ubiquitous – think cloud, data center
  - Huge demand for networking professionals in industry

Textbook

- No mandatory textbook, all material will be on iLearn/Website
- You will need to brush-up your fundamentals
  - Kurose, Ross: http://www-net.cs.umass.edu/kurose-ross-ppt-6e/
  - Peterson, Davie: https://book.systemsapproach.org/

Semester Project

- Group of 2 students, with about 3.5 months to work on a problem
  - They will not be easy, start immediately
  - Open-ended problems
- You will have research and implementation components
- Propose your own topics, or choose one from samples

Paper Presentations

- Presentation of research papers
  - What is this about?
  - What is the problem they try to address?
  - Why it is important?
  - What is the solution?
  - Is it novel?
  - What are the drawbacks/limitations?
  - How can they be solved?
  - What are the lessons learned?
- Think about it! Critique it.
- One person presents (~40 minutes), others critique (~20 minutes)
  - Everyone will ask questions
  - Everyone will summarize what they have learned

Paper Presentations

- Presentation of research papers
  - What is this about?
  - What is the problem they try to address?
  - Why it is important?
  - What is the solution?
  - Is it novel?
  - What are the drawbacks/limitations?
  - How can they be solved?
  - What are the lessons learned?
- Think about it! Critique it.
- One person presents (~40 minutes), others critique (~20 minutes)
  - Everyone will ask questions
  - Everyone will summarize what they have learned

Paper Presentations

- Presentation of research papers
  - What is this about?
  - What is the problem they try to address?
  - Why it is important?
  - What is the solution?
  - Is it novel?
  - What are the drawbacks/limitations?
  - How can they be solved?
  - What are the lessons learned?
- Think about it! Critique it.
- One person presents (~40 minutes), others critique (~20 minutes)
  - Everyone will ask questions
  - Everyone will summarize what they have learned

Fundamentals

- Networking is ubiquitous
  - What did you use it for today?

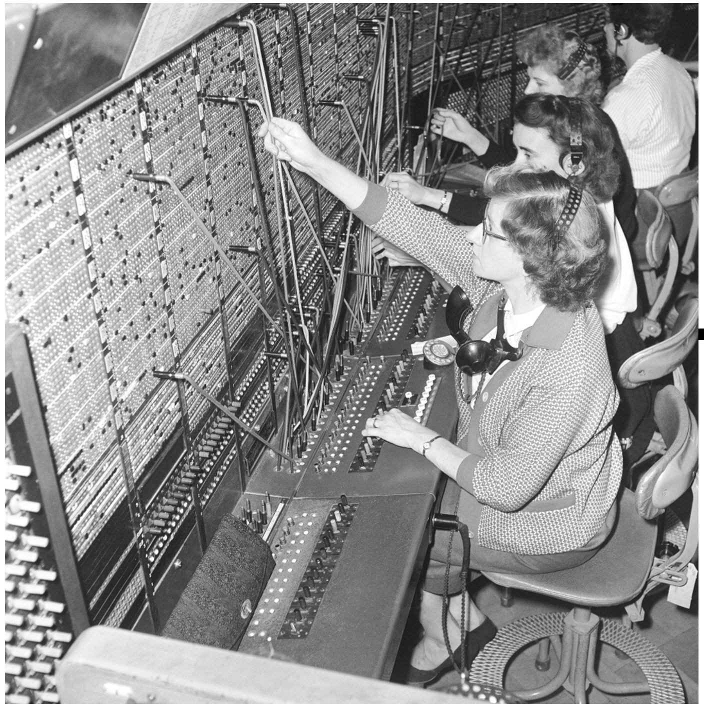
- First things first:
  - Terminology
  - Basic tools
  - What does it take to build an Internet?

Fundamentals

- Networking is ubiquitous
  - What did you use it for today?

- First things first:
  - Terminology
  - Basic tools
  - What does it take to build an Internet?

Circuit Switching – Old telephone networks



- Build physical wire:
  - Guaranteed resources
  - Great for voice

Why change?

Telephony: Design Tradeoffs

- Communication between 2 points:
  - a caller (source) and a listener(destination)
- Communication identifier:
  - telephone number
- Established circuit between source and destination
  - Highly reliable, but slow
  - Calls need to go through a switching system to be routed to the destination
- Take-away point:“Dumb” terminals, “smart” networks

Telephony: Only for Voice, Simple

- Constant bit rate - 64 Kbps
- Reliable
- Error recovery?
- Tight delay constraints (100-200ms)
- How do we send pictures?

Early days of the Internet

- US military project through Defense Advanced Research Project Agency (DARPA –at that time ARPA)
  - Outcome: ARPANET
- Enable robust communication during cold war _if_ the soviet union launched a nuclear attack against the US
  - If a link was broken, how still to communicate?
- DARPA opened it to universities in the 80s

Early days of the Internet

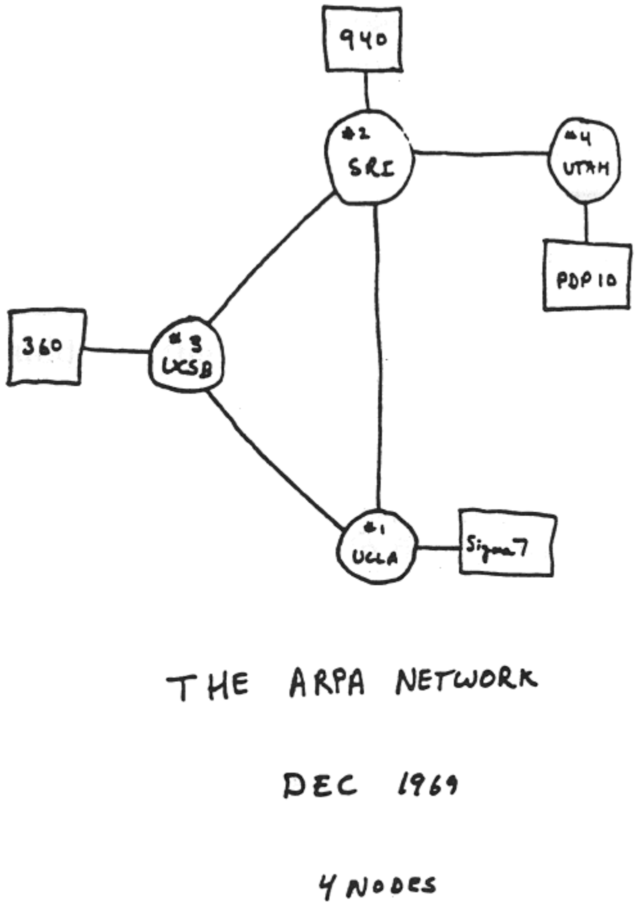
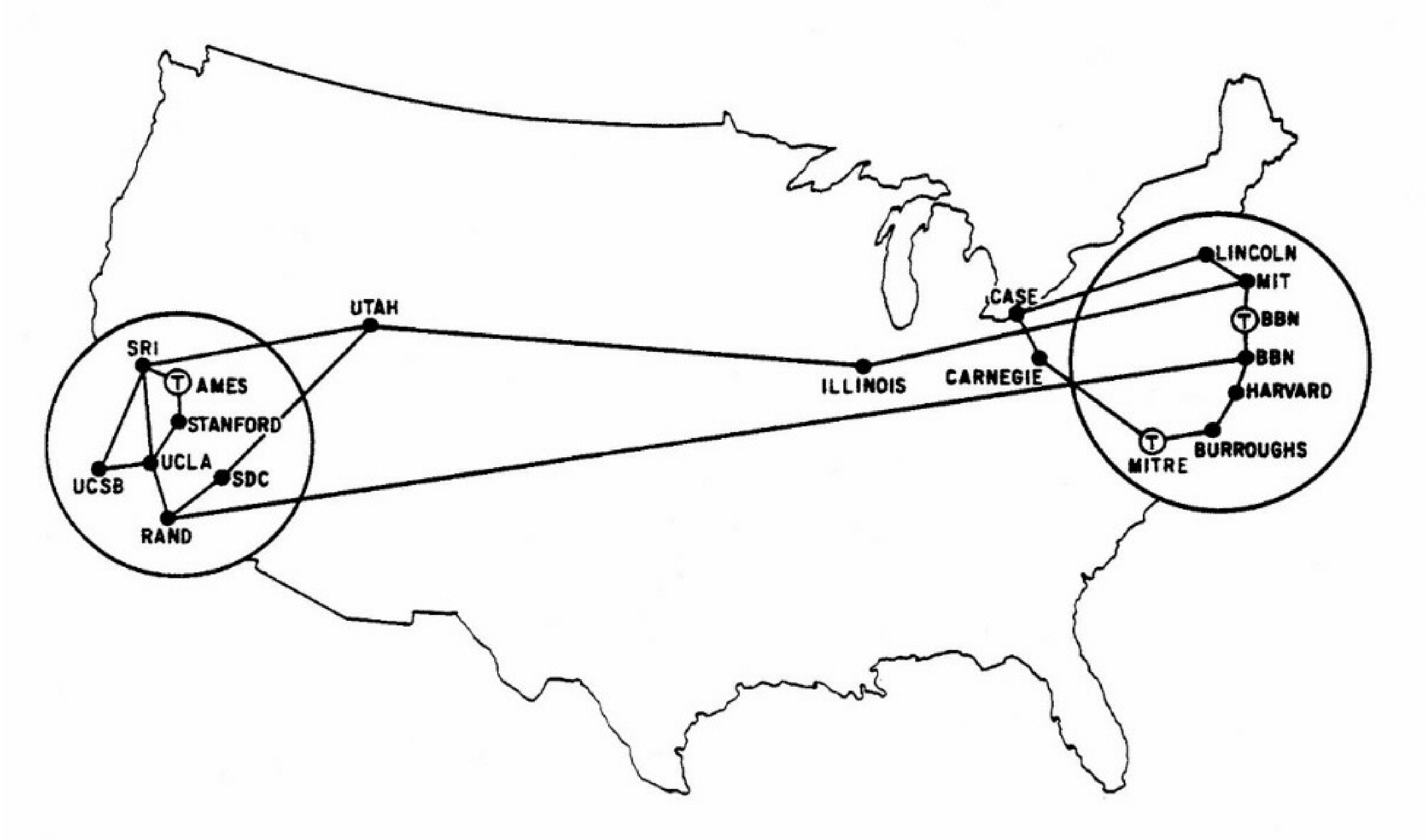


FIGURE 6.2 Drawing of 4 Node Network
(Courtesy of Alex McKenzie)

Early days of the Internet

1970, ARPANET hosts start using NCP; first two cross-country lines (BBN-UCLA and MIT-Utah)



Early days of the Internet

- 1961 – Packet switching
- 1971 - 15 nodes (23 hosts): UCLA, SRI, UCSB, Univ of Utah, BBN, MIT, RAND, SDC, Harvard, Lincoln Lab, Stanford, UIU(C), CWRU, CMU, NASA/Ames on ARPANET
- 1974 – Telnet , TCP design
- 1984 – DNS
- 1986 - NSFNET created (backbone speed of 56Kbps)
- 1993 - InterNIC (directory, information services)
- 1995 – Commercial Internet
- Then things explode…

Internet design goals

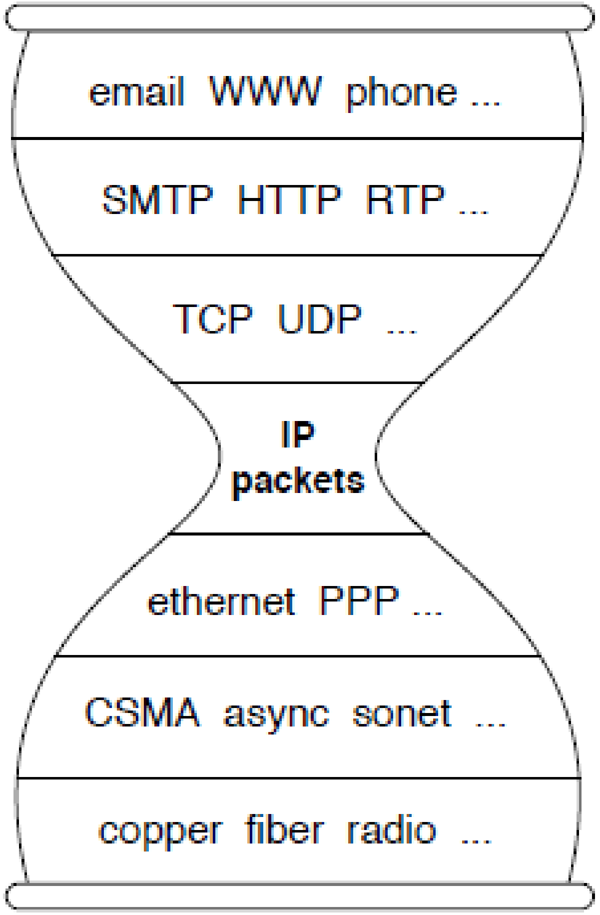
- In order of importance:
  - Inter-connect existing networks (ARPANET, packet radio, satellite, etc.)
  - Survivability (ensure communication when router and link failures happen)
  - Support different types of services/applications
  - Must accommodate a variety of networks
  - Permit distributed management of resources
  - Be cost effective
  - Permit host attachment with low levels of effort
  - Resources used must be accountable

Thoughts on Internet Design

- Different ordering of the previous goals may have resulted in a different architectural design
- Fundamentally different design compared to telephony! Needed to serve MANY different applications
- Traffic can be bursty, many different protocols.
- Best-effort delivery
  - no quality of service
- 2 fundamental design principles
  - Layering
  - End-to-end principle

Layering

- Different layers handle different functions (services)
- To provide multiple different functions, we need protocol layering
- Conceptually, we split a task into multiple (simpler) subtasks
  - each layer provides one subtask
  - Upper layers can use lower layer functionality
- Problems?
- Why the IP layer is thinner than the others?



wikipedia

End-to-End Argument

- Which layer to choose for a specific function?
  Upper or lower?
- Fundamental principle of network systems design
  - Reasoning against low-level function implementation•"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)"

  - In other words: Provide only fundamental functions in the network, since payoffs from adding features may quickly diminish

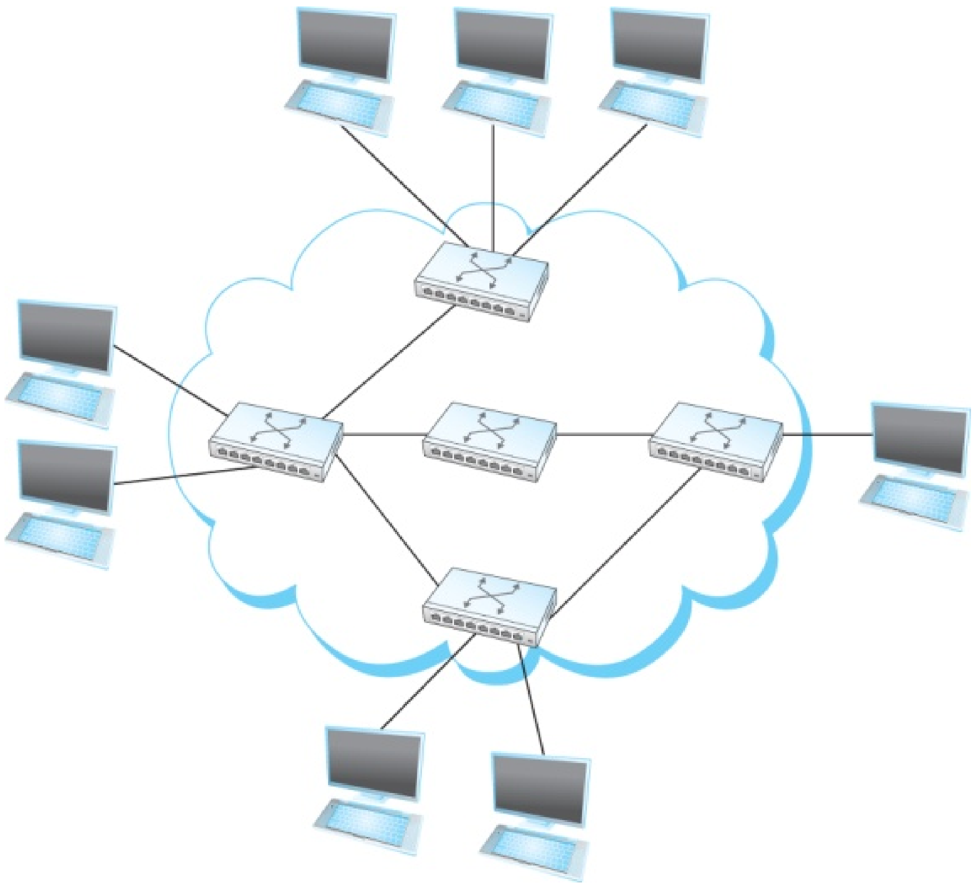
  - Application-specific functions should be implemented by end nodes
    - Different people interpret the end-to-end principle in different ways

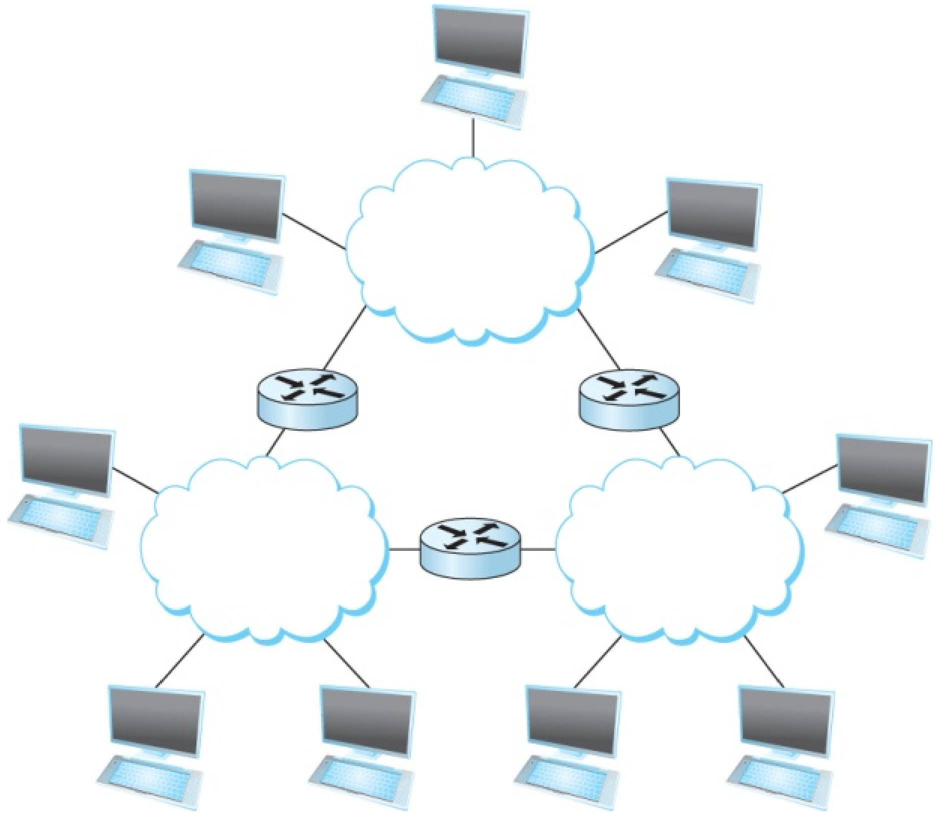
Links, Nodes, Network, Internet

- You can view the network as a graph
- Each device (a phone, a computer) is a node
- Each connection is a link
  - Wires = real links
  - Bluetooth, Radio, Infrared = virtual links
- Nodes + links = a network
  - Many connected networks = Internet

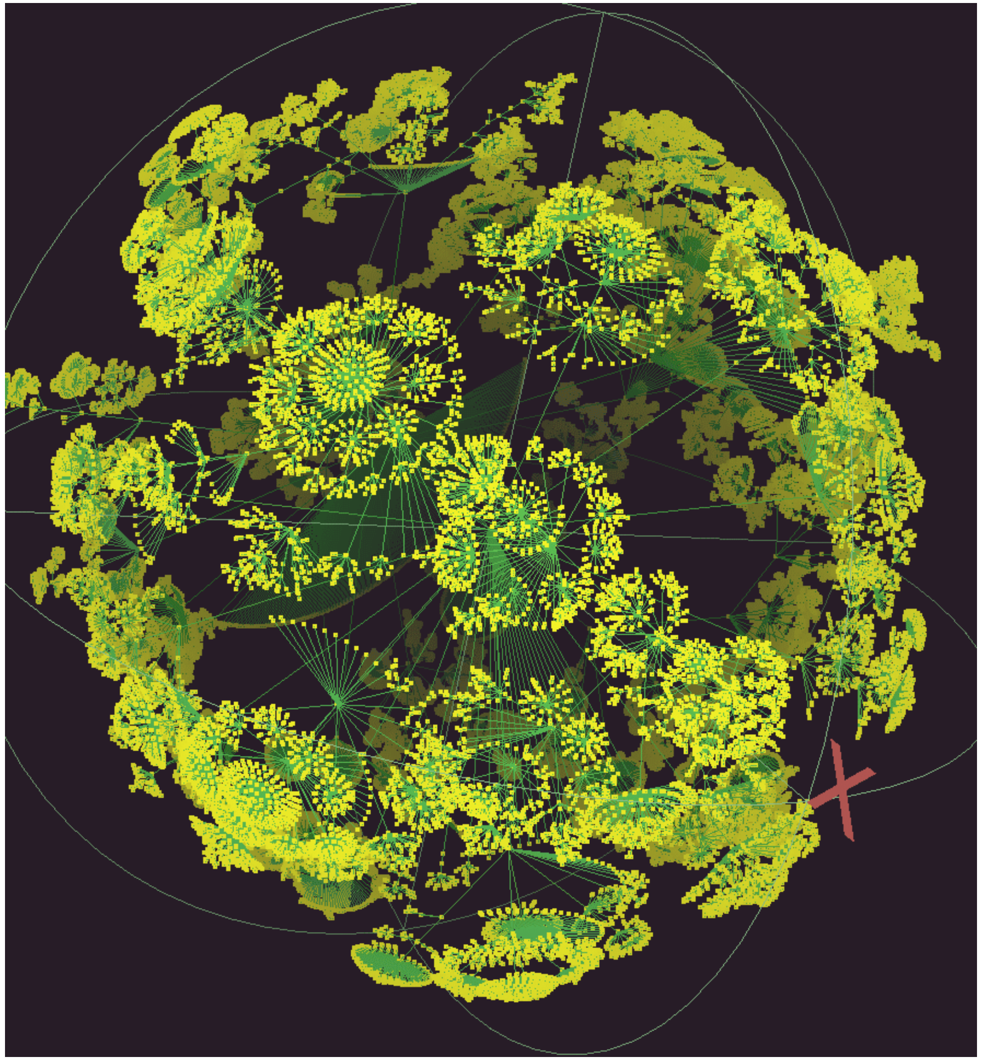
Links, Nodes, Cloud, Routers, Switches

A Network and the Internet

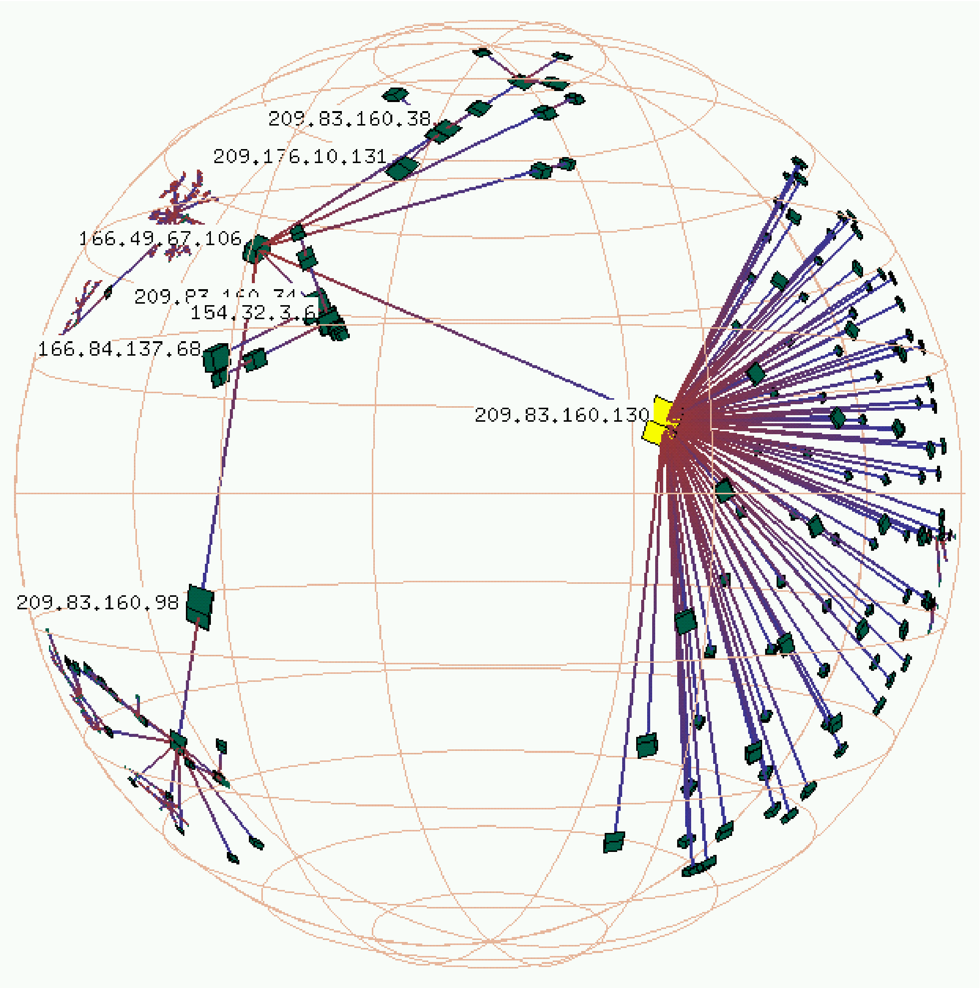




Links, Nodes, Network, Internet



Not
Actual
data



https://www.caida.org/tools/visualization/walrus/gallery1/lhr-old.png

Client and Server

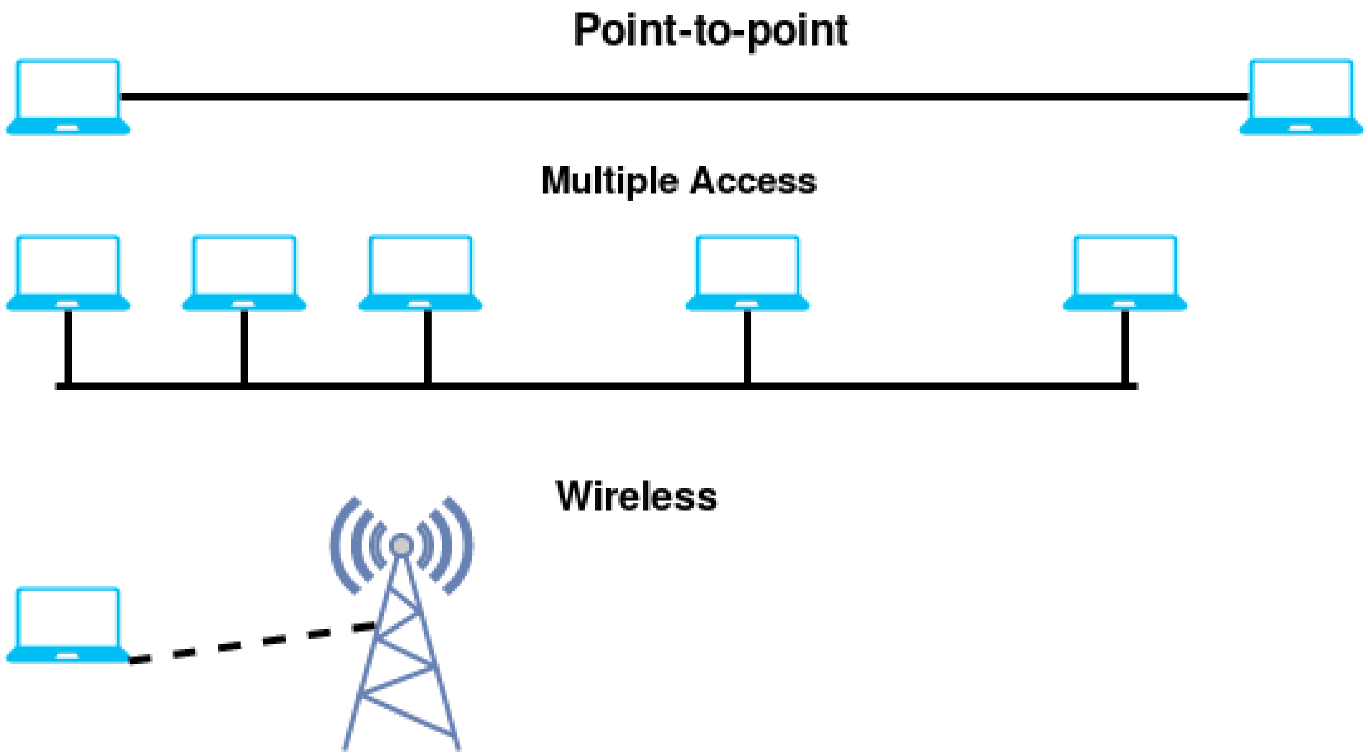
- My laptop with a browser = client
  - It requests a service
  - Email, chat, video, youtube

- A node running a program that serves the requests = server
  - Runs a service
  - Chat, video, messaging

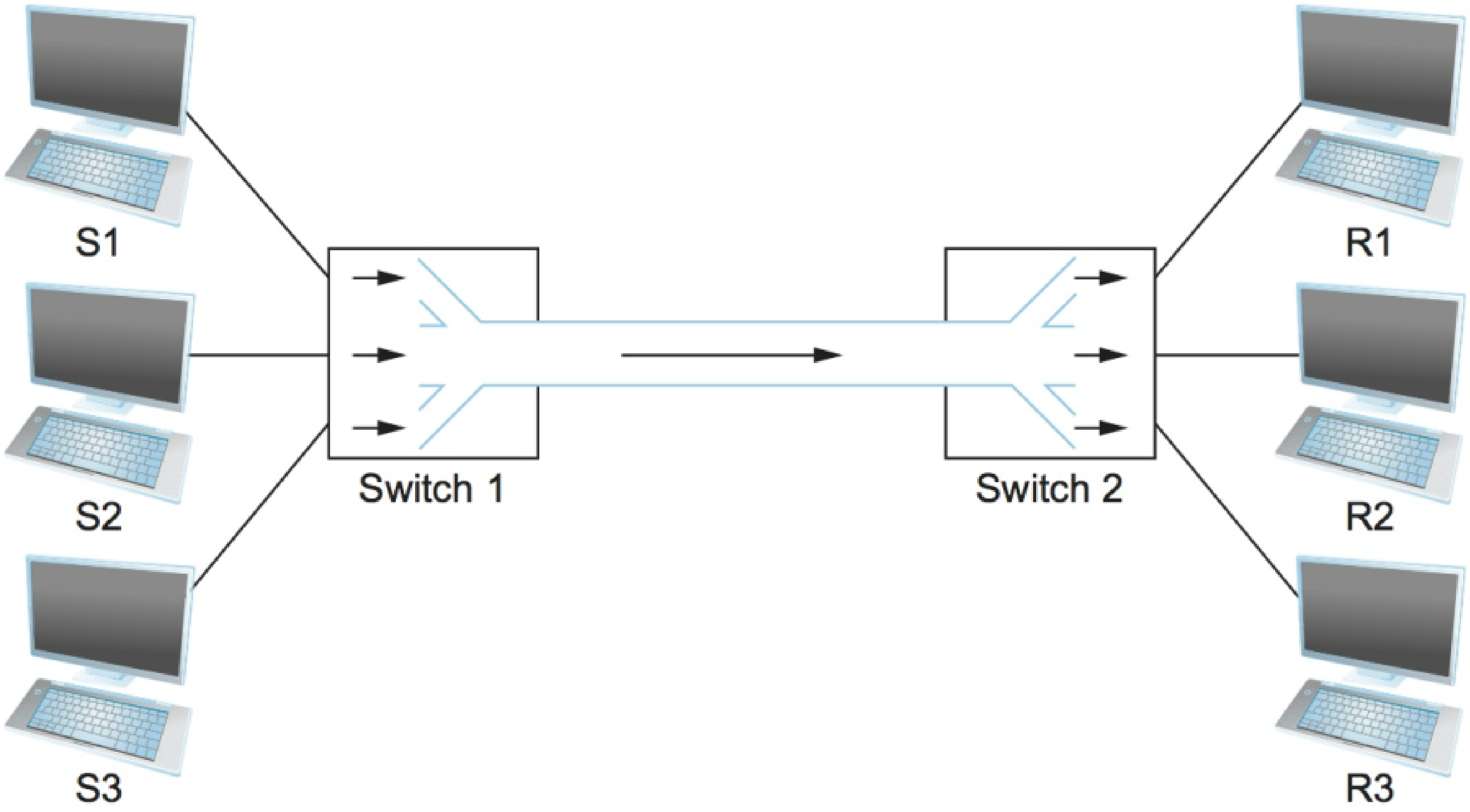
- A node can both be a client and a server

Connectivity

- Point to Point
- Multiple access
- Wireless

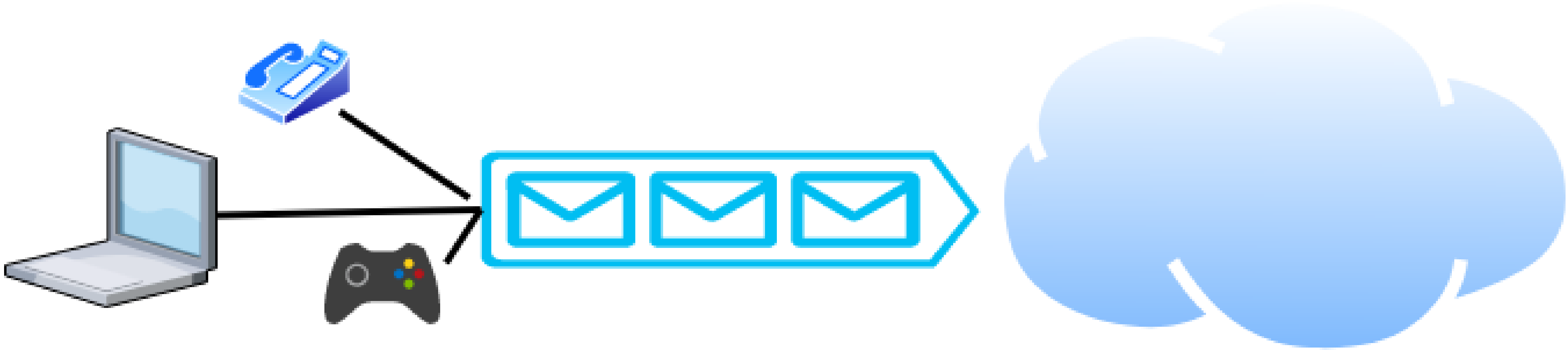


Circuit Switching – TDM and FDM



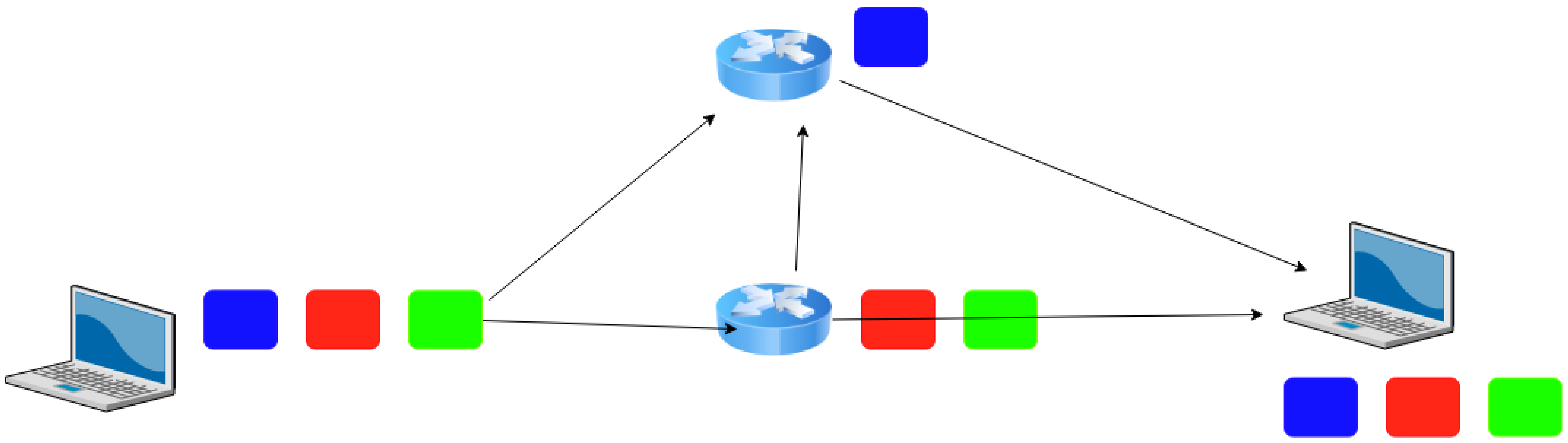
What are the problems?

Packet Switching

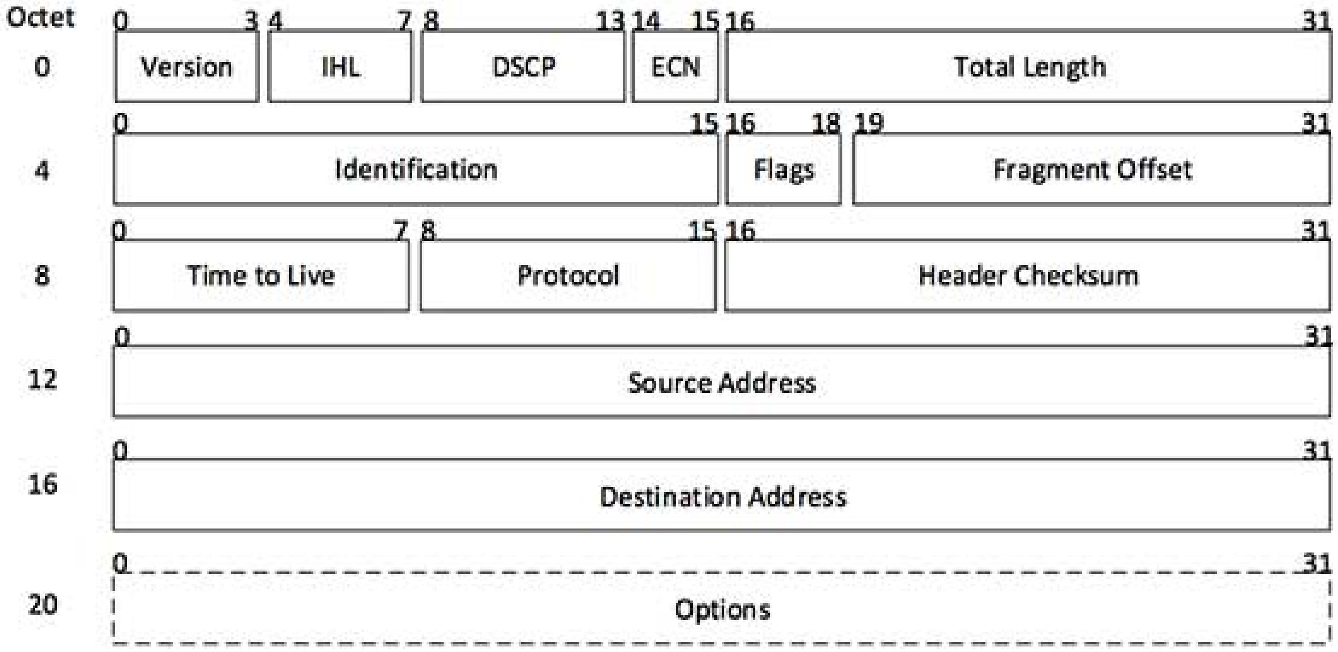


- Packets are low level components
- Multiple kind of traffic with different requirements
  - Gaming vs Phone
- Dumb network – How do you ensure quality of service?
- End points must be smart

Packet Switching



But What is a Packet?



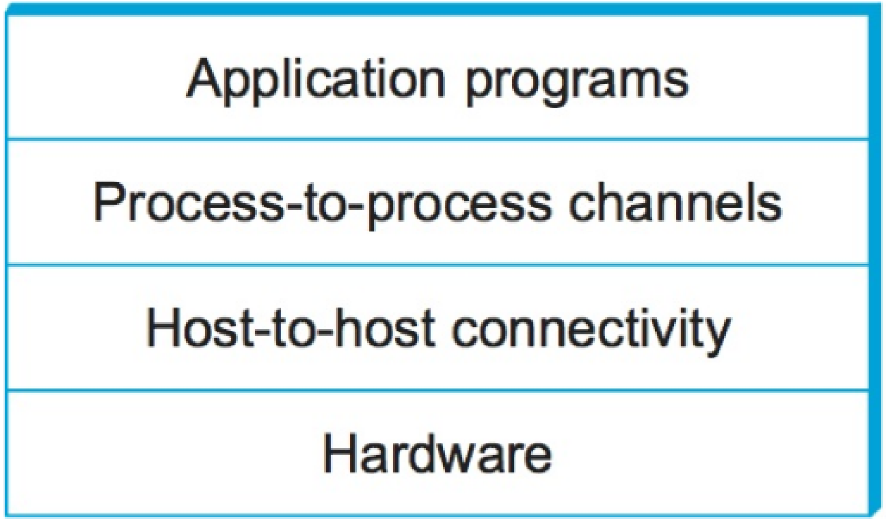
[Image: IP Header]

- Self-contained data unit
- Has two parts (generally)
  - Control information
  - Payload
- How do we transmit a dictionary?

Network Architecture

- What are the requirements from a network?

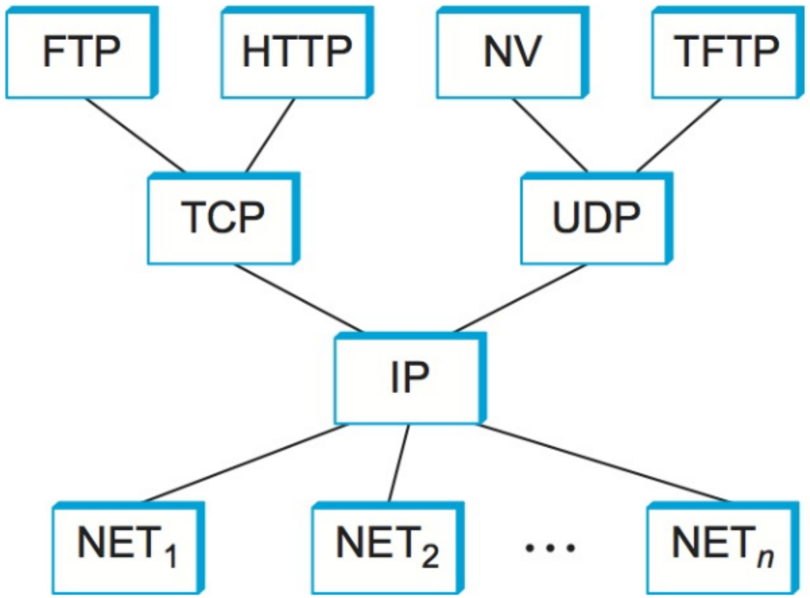
- Architecture = High-level blueprint
  - Protocols = Building blocks of the architecture
  - Layering = Break down the problem in smaller pieces

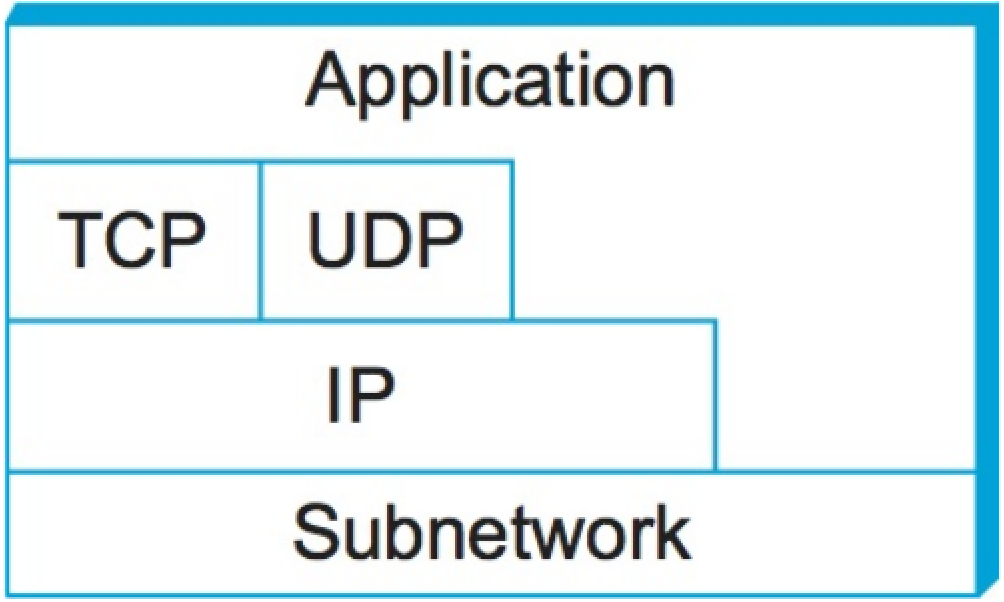


Network Layers

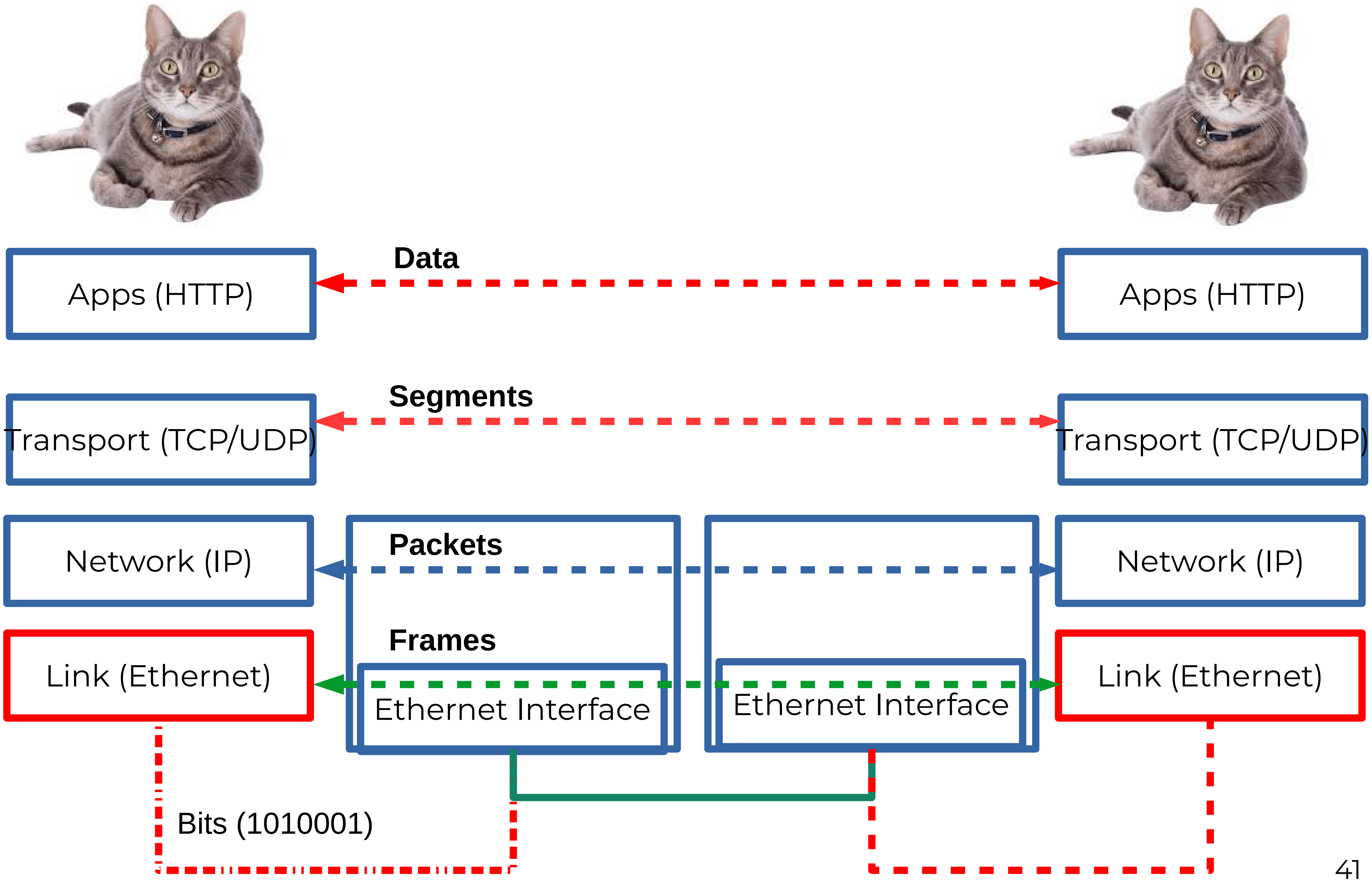
Use Service

Export Service





- Makes it easier to divide functionality
- Hides implementation details
- What else?


Data
Apps (HTTP)
Apps (HTTP)
Segments
Transport (TCP/UDP)
Transport (TCP/UDP)
Packets
Network (IP)
Network (IP)
Frames
Link (Ethernet)
Ethernet Interface
Ethernet Interface
Link (Ethernet)
Bits (1010001)

Application Layer

- Any application you can think of!
  - Skype, Facebook, Twitter, etc.

- Applications are aware of URLs (essentially some sort of names)
  - Example: www.facebook.com
  - Application data is in the payload of an HTTP/HTTPS message (another application)
  - Applications understand URLs, network understands IP addresses

  - Translation of URLs to IP addresses through the Domain Name System (DNS)
    - Why applications understand URLs and not IP addresses?

Transport Layer

- HTTP/HTTPS requests/responses are encapsulated into TCP segments or UDP datagrams
  - TCP: reliable delivery
    - Establishes a connection between 2 hosts
    - Does in-order delivery, error check, flow and congestion control

  - UDP: Best-effort datagram delivery
    - Connectionless

  - Why we need different transport layer protocols?
  - Why not having reliability implemented at the network layer?
  - Is transport the proper layer to do congestion control?

Network Layer

- Stateless, best-effort IP
  - The glue that puts the Internet together!
  - Absolutely minimum common functionality needed by applications to operate over a network

  - “Give me a destination IP address and I will try to deliver the packet to the destination”

  - Communication model between 2 hosts (source, destination IP address)
    - How to do group chat then?
    - Multicast efforts exist, but, do not scale well

Data Link Layer

- Many different protocols
  - Ethernet, WiFi, LTE, 5G, and many more!

- Why do we need so many?
  - The network layer has only one protocol
  - Adaptation to different communication environments
  - Wired, wireless, mobile, etc..

Physical Layer

- Transmits streams of raw bits (0s and 1s)
  - Has to do with electronic circuit transmission methods
- Implemented through network hardware
  - We will assume it exists and it works

Wait – where is security?

- There is no security in the Internet design
- Look through the list of priorities!
  - First application? → File transfer
- Possible oversight
- Do we need security today?

Conclusion

- The success of the Internet is because of the applications
- People are always coming up new and innovative applications
  - The network must support them
- Recent trend: Ubiquitous Augmented Reality/Virtual Reality (AR/VR)
  - Need <10ms latency
  - How?

Next Class

- Tuesday - Paper 1 presentation – volunteer?
- Thursday – Quiz on TCP/IP and Paper 1
- Thursday – Paper 2 presentation – volunteer?