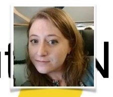
15-441/6 Comput Networks oduct



15-441 Fall 2019
Profs **Peter Steenkiste** & Justine Sherry

Fall 2019
https://computer-networks.github.io/fa19/

Carnegie Mellon University

Overview

- Course overview and administrative issues
- Packet-switched networking fundamentals
- The Web as a motivating application (P1 preview)
- How to build an Internet: the protocol stack
- The life of a packet
- The end-to-end argument

What makes up a network?

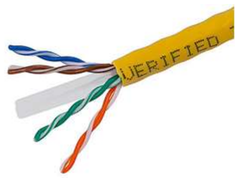
- A **communication medium** is the means by which information (the message) is transmitted between a speaker or writer (the sender) and an audience (the receiver).
- **Data transmission** is the process of sending digital or analog data over a communication medium to one or more computing, network, communication or electronic devices
- **Data encoding** is the process of applying a specific code, such as letters, symbols and numbers, to data for conversion into an equivalent cipher

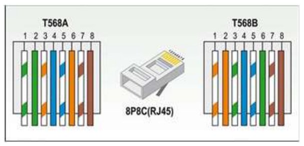
Do you remember this movie?

Why did I just show you this video?

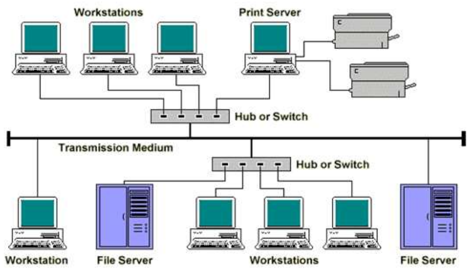
- The Great Wall of China used **beacon towers** to transmit information
  - The *communication medium* was light and darkness
  - The *data encoding* had only two values:
    - Light means danger! No light means safe!
  - *Data transmission* occurs as the light travels from outpost to outpost — at the speed of light!

A modern communication medium





- **Ethernet over copper**
  - What is the medium?
  - How is data encoded?
  - How is data transmitted?



When you connect two or more endpoints with a communication medium, you have a network.

Modern Networks

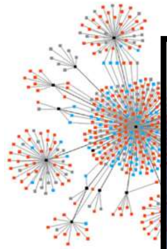
Datacenters

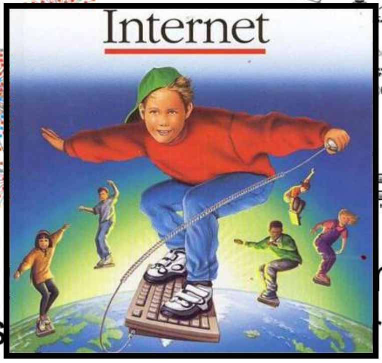
Low Earth Orbit (LEO) Satellites

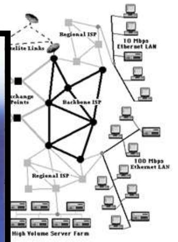
University Campus

How is building each of these networks different from each other?

What might make it more challenging to develop, say, web applications for each scenario?



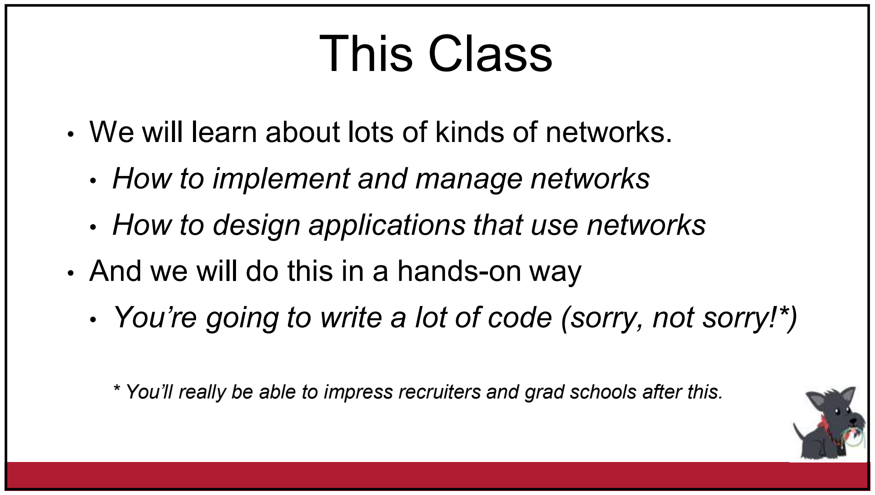




When networks r more network.

Why should I care about this class?

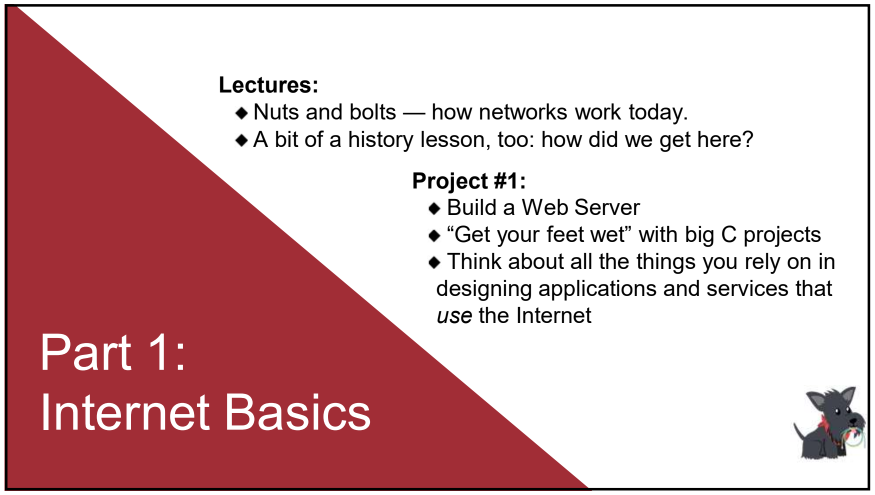
Would you rather have a super computer without a network connection, or a simple cell phone with connectivity to the Internet?



This Class

- We will learn about lots of kinds of networks.
  - *How to implement and manage networks*
  - *How to design applications that use networks*
- And we will do this in a hands-on way
  - *You're going to write a lot of code (sorry, not sorry!*)*

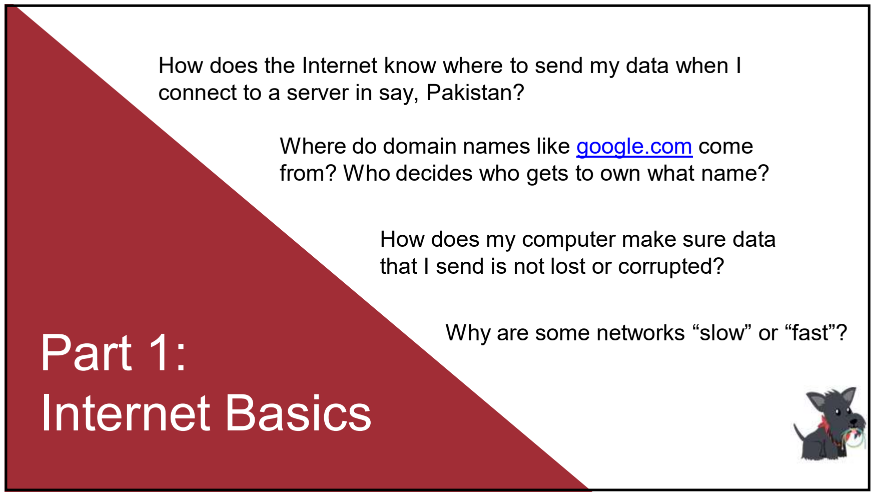
** You'll really be able to impress recruiters and grad schools after this.*



Lectures:
- Nuts and bolts — how networks work today.
- A bit of a history lesson, too: how did we get here?

Project #1:
- Build a Web Server
- "Get your feet wet" with big C projects
- Think about all the things you rely on in designing applications and services that *use* the Internet

Part 1: Internet Basics



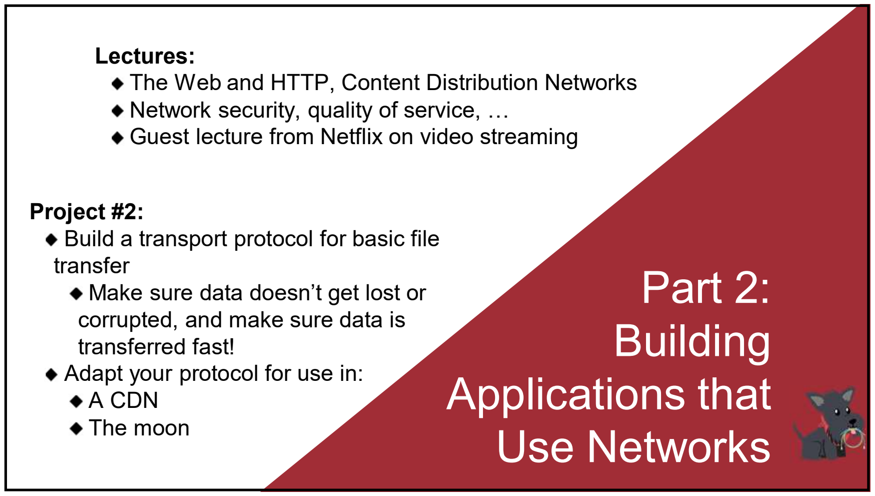
How does the Internet know where to send my data when I connect to a server in say, Pakistan?

Where do domain names like google.com come from? Who decides who gets to own what name?

How does my computer make sure data that I send is not lost or corrupted?

Why are some networks "slow" or "fast"?

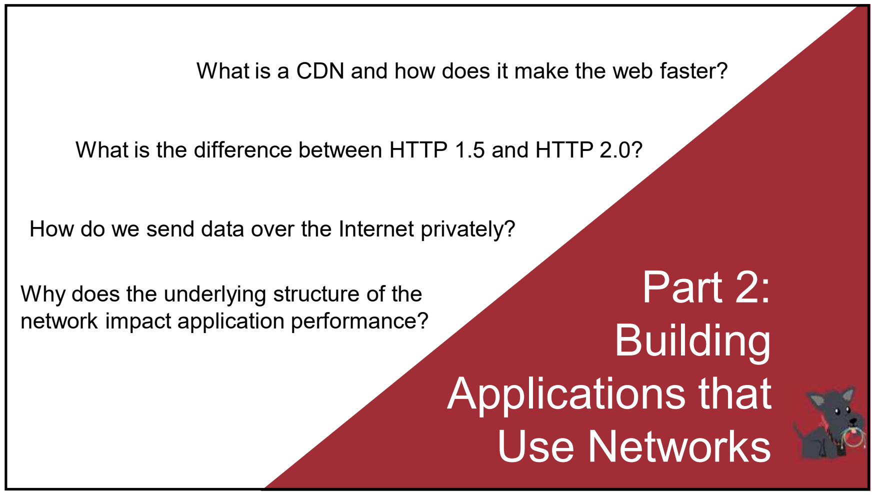
Part 1: Internet Basics



Lectures:
- The Web and HTTP, Content Distribution Networks
- Network security, quality of service, …
- Guest lecture from Netflix on video streaming

Project #2:
- Build a transport protocol for basic file transfer
  - Make sure data doesn't get lost or corrupted, and make sure data is transferred fast!
- Adapt your protocol for use in:
  - A CDN
  - The moon

Part 2: Building Applications that Use Networks



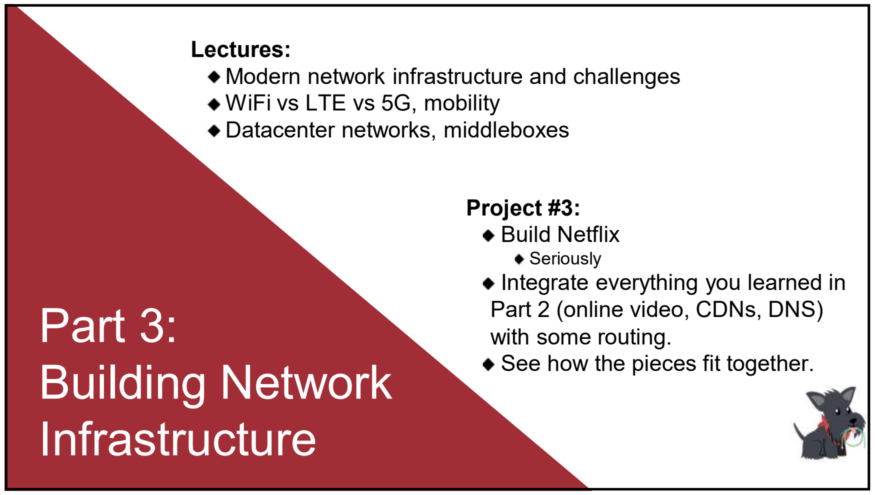
What is a CDN and how does it make the web faster?

What is the difference between HTTP 1.5 and HTTP 2.0?

How do we send data over the Internet privately?

Why does the underlying structure of the network impact application performance?

Part 2: Building Applications that Use Networks



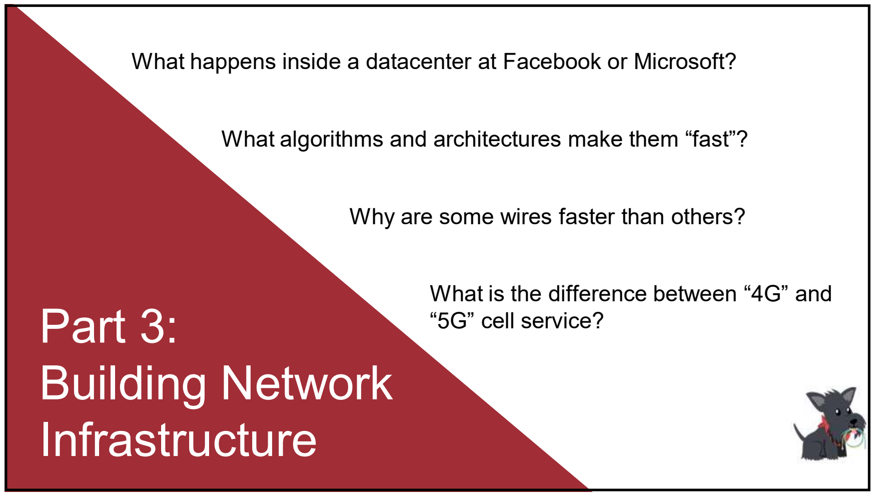
Lectures:

- Modern network infrastructure and challenges
- WiFi vs LTE vs 5G, mobility
- Datacenter networks, middleboxes

Project #3:

- Build Netflix
  - Seriously
- Integrate everything you learned in Part 2 (online video, CDNs, DNS) with some routing.
- See how the pieces fit together.

Part 3: Building Network Infrastructure



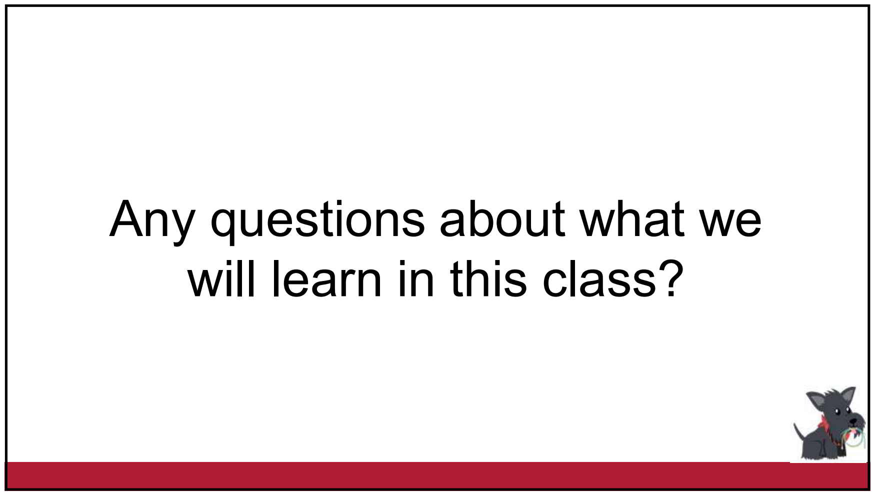
What happens inside a datacenter at Facebook or Microsoft?

What algorithms and architectures make them “fast”?

Why are some wires faster than others?

What is the difference between “4G” and “5G” cell service?

Part 3: Building Network Infrastructure



Any questions about what we will learn in this class?

Course Policies

Mostly outlined in the syllabus - this is an overview

MOST IMPORTANT WAY TO SUCCED IN THIS CLASS:
The majority of your grade comes from class projects

- **45**% for Projects I, II and III
- 18% for Midterm exam
- 27% for Final exam
- 10% for Homework

This means: START EARLY! Use office hours to ask for help! Debug your code with your own testing scripts!

Late Work

- We will give you two “late days” for free.
  - You don’t need to tell us which days you are using — we calculate late days at the end of the semester to your advantage (e.g., if you turn in both a project and a homework two days late, we will give you full credit on the project since that is worth more points.)
- Any other late assignments are penalized 15% per day late. No assignments are accepted more 48 hours after the deadline.
- See the syllabus about dealing with emergencies.

Don’t Cheat. Seriously.

- Working together is important
  - Discuss course material in general terms
  - Work together on program debugging, ..
  - Collaborating on projects P2 and P3
- Final submission must be your own work
  - Homeworks, midterm, final, projects….
- Submitting or using someone else’s work is an academic integrity violation (i.e., cheating)
  - We will follow the university policy on reporting violations
  - Voluntarily sharing your work is also a policy violation
- Web page has details, e.g., university policy, etc.

REALLY don't cheat on the projects

- The project code you submit must be your work!
  - Exception is the starter code provided by us, standard libraries, packages mentioned in the project handout
  - If in doubt, ask the course staff
- We use tools to compare submissions
  - These tools are very good
  - Don't compete with them (the odds are against you)
- Some students have put their projects on the web
  - Posting and using the code is a form of cheating
  - If you can find the code, so can we

CMU's Disability Services Office is Great

- If you need their help — for any reason — we do what they tell us to, no questions asked (we don't need to know why you need accommodations).
- Please email the professors a copy of the accommodations sheet for us to sign, or bring a paper copy to either of us in Office Hours.

CAPS is also great.

- Whenever are worried about a student, we call CAPS and they give great advice.
- Many people think CAPS is just for people with severe mental health troubles. You can also go just because you're feeling a little stressed about *anything* and you need someone to talk to.
  - Seriously, no problem is too small.
  - If you think about visiting them, just go ahead and do it.

How do I get off of the waitlist?

<u>213/513 is a pre-requisite for this course – for 641 you need a B or better</u>

- We check the prerequisite and enroll students who meet it
  - But grades for the 513 summer session will only be available later this week so please be patient
- If you meet the prerequisite but you are still on the mailing list, email us when you took 15-213/15-513 and what your grade was
- If you did not take 213/513 but you think you have equivalent experience, email us with some information on your background and we will admit you if there is room

Any policy questions?

Your TAs are amazing.

- Alex Bainbridge
- Aneek Mukherjee
- Kartik Chitturi
- Ines Potier
- Mingran Yang

Back to technical stuff! (Fun!)

Overview

- Course overview and administrative issues
- Packet-switched networking fundamentals
- The Web as a motivating application (P1 preview)
- How to build an Internet: the protocol stack
- The life of a packet
- The end-to-end argument

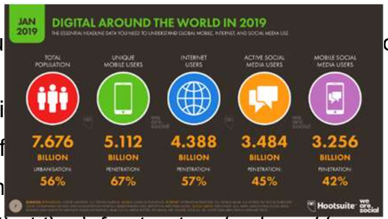
What is a Network?

- An infrastructure that allows distributed “users” to communicate with each other
  - People, devices, …
  - By means of voice, video, text, …
  - We focus on electrical/optical/RF/.. (not trucks)
- It is assumed that the infrastructure is shared by many users
  - Value increases with the number of users!

34

What is a Network?

- An infrastructu communicate with each other
  - People, devi
  - By means of
  - We focus on
- It is assumed that the infrastructure is shared by many users
  - Value increases with the number of users!
- Our goal: build a network for 4.39 billion users … and growing

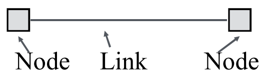


* Graphic: https://wearesocial.com/blog/2019/01/digital-2019-global-internet-use-accelerates

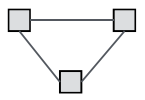
35

Basic Building Block: Nodes, Links

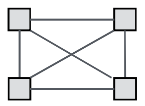
- Simplest example: 2 nodes
  - Sender changes voltage, frequency, …
  - Or maybe it is optical or wireless?



- But receiver must “understand” sender – protocols
  - More on this later

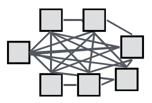


- Okay… what about more nodes?
- How about a million?

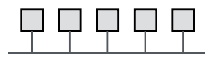


36

Scaling the Network

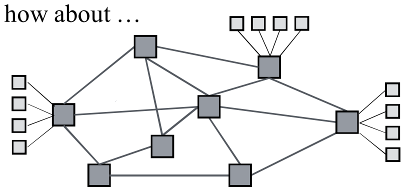


(N^2) Wires for everybody!



Or one wire

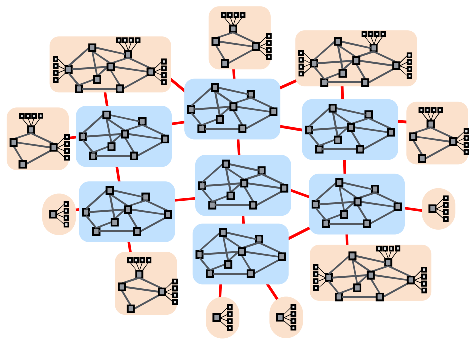
Or how about …



- We have switches and links connecting devices in a toplogy
- Data needs to travel along a path from a source node to a destination node
- Networks are owned by a person or organization

37

How about an Internet?



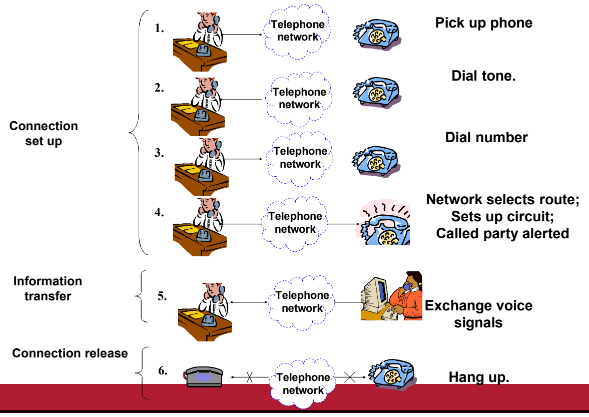
- An Internet is a network of networks
  - Network = Domain
  - Domains are ISPs or client networks (stubs)
- The Internet infrastructure is owned by many companies!
- Paths now traverses two types of links
  - Inter-domain links and intra-domain links

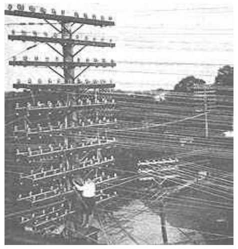
How is the Internet Unique?

- Alllow diverse applications on diverse devices to communicate …
  - Web, peer-to-peer, video streaming, distributed processing, transactions, map-reduce, video and audio conferencing, …
- … over very diverse infrastructures
  - The “Internet”, WiFi and cellular, data center networks, corporate networks, dedicated private networks, …
- In contrast: previous networks were special purpose and fairly homogeneous in terms of technology
- Context: it is the 1960’s and you are asked to design an Internet …
- … your starting point is the telephone network

39

POTS: A Circuit-switched Network

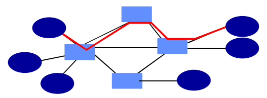




Plain Old Telephone System

Circuit Switching

- Source first establishes a connection (circuit) to the destination
  - Each switch along the way stores info about the circuit, e.g., two wires are connected (POTS)
- Source sends the data over the circuit
  - No need to include the destination address with the data since the switches know the path
- The circuit is explicitly torn down
- The circuit has a fixed, guaranteed bandwidth
- Example: telephone network (analog – remember 1960’s)



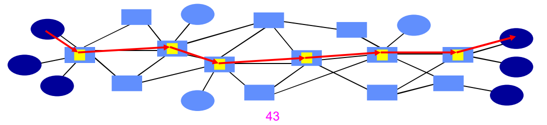
41

Circuit Switching Discussion

- Circuits have some very attractive properties.
  - Fast and simple data transfer, once the circuit has been established
  - Predictable performance since the circuit isolate users from each other
  - E.g. guaranteed bandwidth
- But it also has some shortcomings.
  - How about bursty traffic?
    - Do you need a permanent circuit to Facebook?
      - And are you willing to pay for it
    - Circuit will be idle for significant periods of time
  - In practice you will need circuits to many destinations
  - How about users with different bandwidth needs?

Contrast this with Packet Switching (our focus)

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

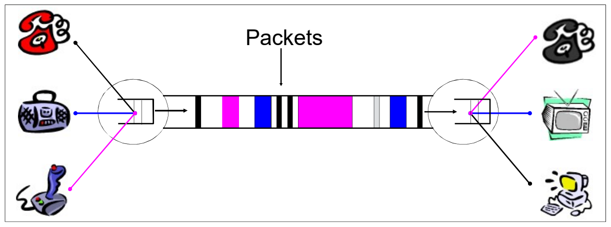


Packet Switching Discussion

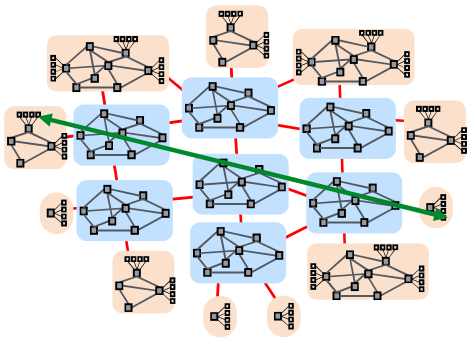
- General: Multiple types of applications
- Efficient: Can send whenever input that is ready
- Accommodates bursty traffic efficiently
  - Statistical multiplexing (next slides)
- Store and forward architecture
  - Packets are self contained units with destination addresses
  - Can use alternate paths – potentially more robust
  - Requires buffering to absorb bursts
- Many challenges, e.g., contention (no isolation)
  - Buffer overflow, congestion, variable delay, high packet rate, …

Key Benefit: Statistical Multiplexing

- Users share the wires at a fine grain - packets
- Links are never idle when there is traffic - Efficient!
- Requires queues to buffer packets



Building an Internet: Talk to me!

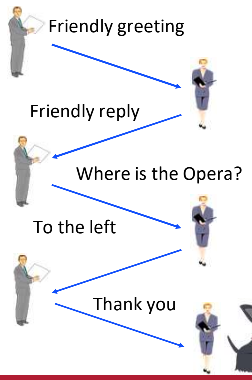


- To support communication two applications need to be able to talk to each other
- Also, two switches, …
- And device-switch, …
- And two networks, …
- You can talk to the person next to you (in English), but talking switches, nodes, software, ..?

46

Protocol: Enable Communication

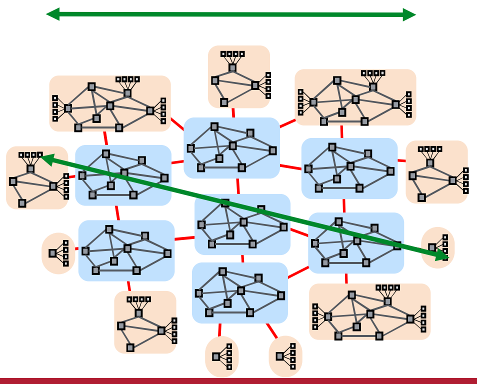
- An agreement between parties on how communication should take place.
- Protocols have to define many aspects of the communication:
  - Syntax: data encoding, language, etc.
  - Semantics: data, error, start/end, …
- Example: Asking for directions
  - English, facial expression, …
- Example: Buying airline ticket by typing.
  - English, ascii, lines delimited by "\n"



Overview

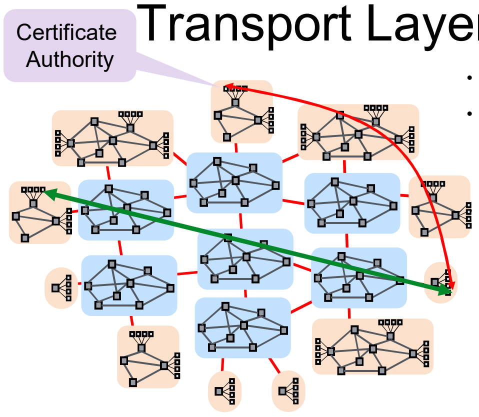
- Course overview and administrative issues
- Packet-switched networking fundamentals
- The Web as a motivating application (P1 preview)
- How to build an Internet: the protocol stack
- The life of a packet
- The end-to-end argument

How Many Protocols do we Need?



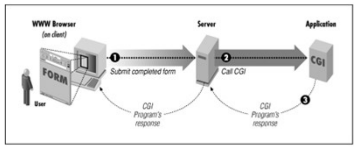
- Social networking web site:
  - http://www.fun.com/my-friends
- What protocols do we need?

Transport Layer Security



- TLS secures the connection:
- Client authenticates server
  - Needs the help from a Certificate Authority (CA)
- Ensures integrity and confidentiality of the data
  - Crypto magic – covered later in the course
  - Transparent to application developers but not to the developer of Web server!

CGI: Generating Dynamic Content



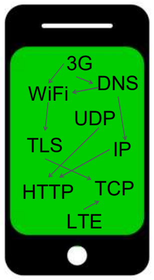
- Web server forward request plus additional information to an external application using a Common Gateway Interface
  - Where the user is connecting from, other user information
  - The CGI can access other data sources, e.g., databases
- CGI returns a response for the browser, e.g., HTTP document

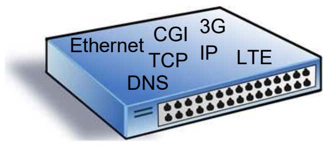
Graphic: https://www.oreilly.com/openbook/cgi/ch01_01.html

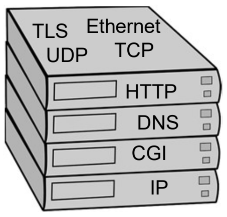
Overview

- Course overview and administrative issues
- Packet-switched networking fundamentals
- The Web as a motivating application (P1 preview)
- How to build an Internet: the protocol stack
- The life of a packet
- The end-to-end argument

Solution #1

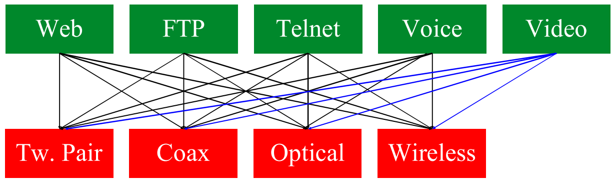






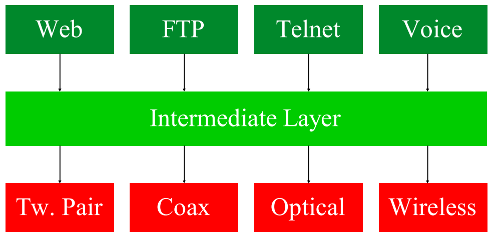
Solution #2

- What is absolutely essential: applications and wires!



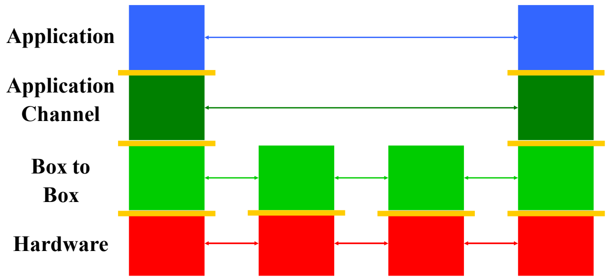
Solution #3

- Too many interfaces!

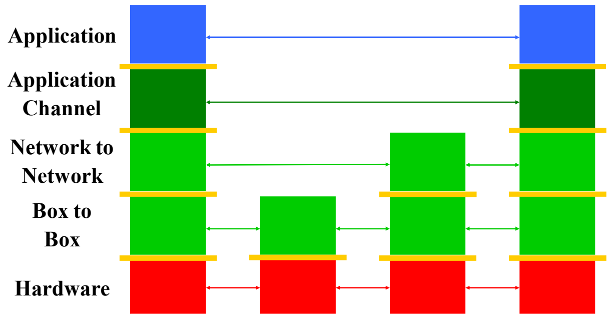


Solution #4

- But where are the protocols?



But we Need an Internet!



Layering: modular approach to network functionality

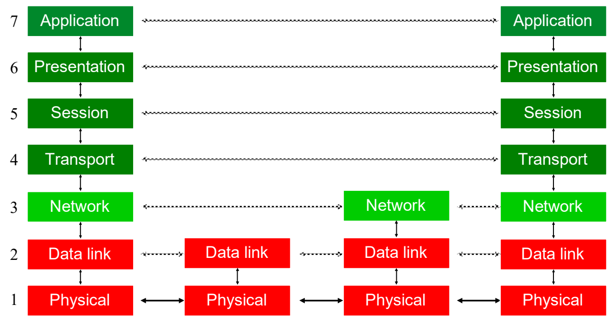
Systems Engineering Wisdom

"Modularity based on abstraction is the way things get done."

Barbara Liskov
Turing Award Winner
+ von Neumann Medal, Computer Pioneer Award,…
+ Pretty much all the things.

A Layer Network Model

The Open Systems Interconnection (OSI) Model



OSI Model: 7 Protocol Layers

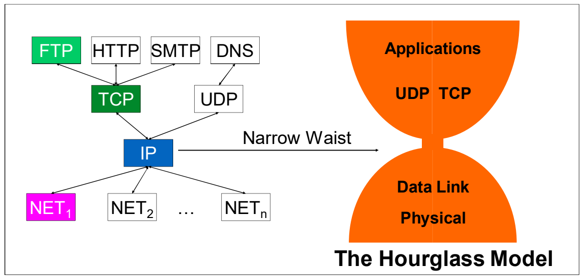
- Physical: how to transmit bits
- Data link: how to transmit frames
- Network: how to route packets
- Transport: how to send packets end2end
- Session: how to tie flows together
- Presentation: byte ordering, security
- Application: everything else

- TCP/IP has been amazingly successful, and it is not based on a rigid OSI model. The OSI model has been very successful at shaping thought

Layering Characteristics

- The stack has two types of interfaces
  - Service: each layer relies on services from layer below and exports services to layer above
  - Protocol: defines interaction with peer on other hosts
- Modules hide implementation - layers can change without disturbing other layers
- A layer can implement multiple protocols that offer the same/similar or different services
  - Datalink: Wifi versus Ethernet
  - Transport: TCP versus UDP

The Internet Protocol Suite

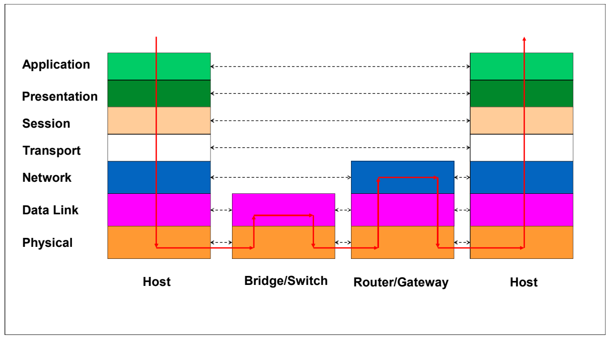


The narrow waist facilitates interoperability
… but evolution is hard

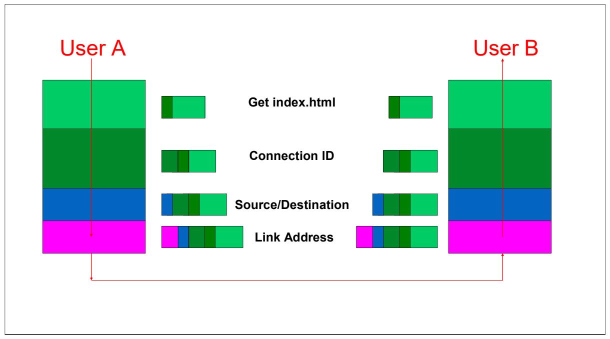
Overview

- Course overview and administrative issues
- Packet-switched networking fundamentals
- The Web as a motivating application (P1 preview)
- How to build an Internet: the protocol stack
- The life of a packet
- The end-to-end argument

Life of Packet

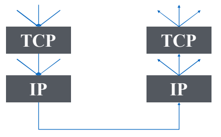


How do Peers Communicate: Headers and Layer Encapsulation



Multiplexing and Demultiplexing

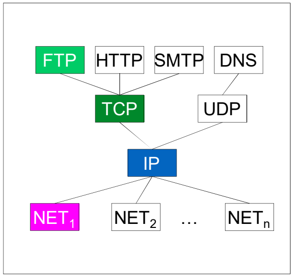
- Multiple choices at each layer: how does a protocol on receiver know what protocol is next?
- Headers include a *demultiplexing field* that is by receiver used to identify the protocol in the next layer.
  - Filled in by the sender
  - Used by the receiver

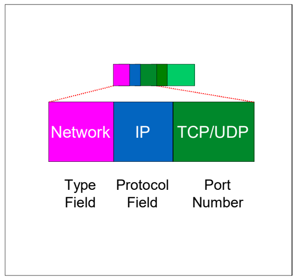


V/HL	TOS	Length
ID		Flags/Offset
TTL	Prot.	H. Checksum
Source IP address		
Destination IP address		
Options..		

Protocol Demultiplexing

- What layers do not need a demultiplexing field?

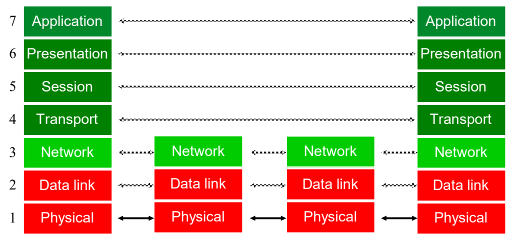




Overview

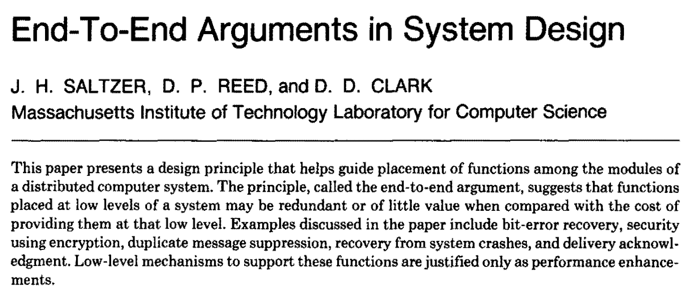
- Course overview and administrative issues
- Packet-switched networking fundamentals
- The Web as a motivating application (P1 preview)
- How to build an Internet: the protocol stack
- The life of a packet
- The end-to-end argument

So Many Protocols to Choose From!



- At what layer should we implement feature X?
  - All layers?
  - The highest?
  - The Lowest?
  - Maybe the Average?
- We need some guidelines?
  - Clearly the answer depends on X!

The End-to-End Argument

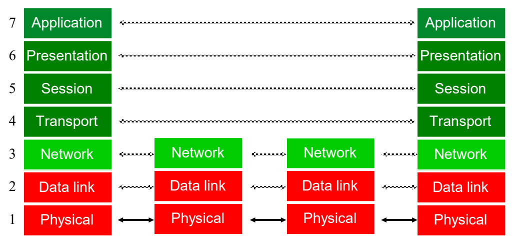


End-To-End Arguments in System Design

J. H. SALTZER, D. P. REED, and D. D. CLARK
Massachusetts Institute of Technology Laboratory for Computer Science

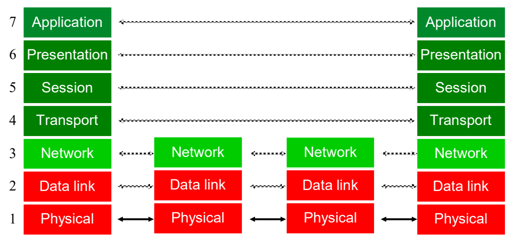
This paper presents a design principle that helps guide placement of functions among the modules of a distributed computer system. The principle, called the end-to-end argument, suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level. Examples discussed in the paper include bit-error recovery, security using encryption, duplicate message suppression, recovery from system crashes, and delivery acknowledgment. Low-level mechanisms to support these functions are justified only as performance enhancements.

So Many Protocols to Choose From!


7 Application Application
6 Presentation Presentation
5 Session Session
4 Transport Transport
3 Network Network Network Network
2 Data link Data link Data link Data link
1 Physical Physical Physical Physical

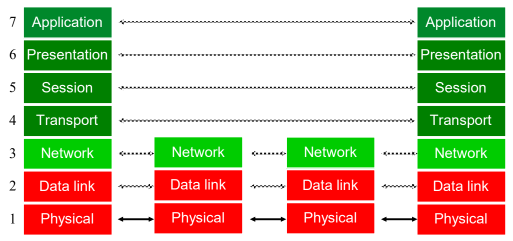
- At what layer should we implement feature X?
  - All layers?
  - The highest?
  - The Lowest?
  - Maybe the Average?
- We need some guidelines?
  - Clearly the answer depends on X!

Example: Reliability

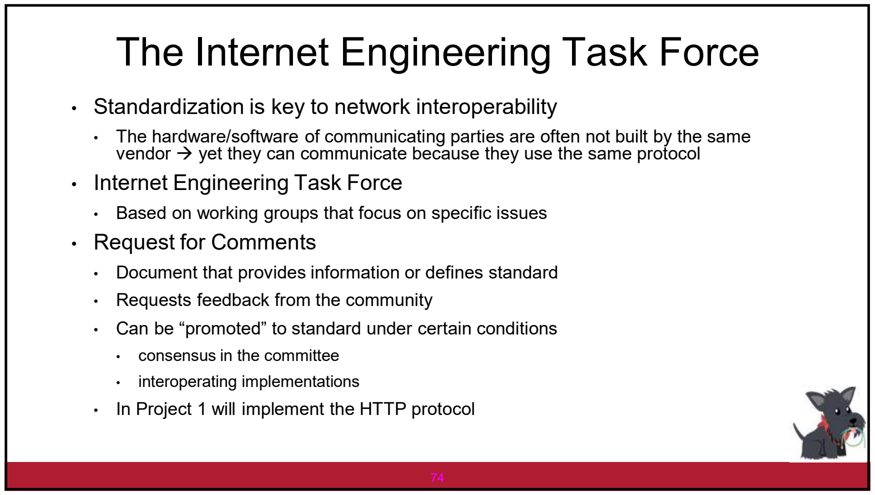

7 Application Application
6 Presentation Presentation
5 Session Session
4 Transport Transport
3 Network Network Network Network
2 Data link Data link Data link Data link
1 Physical Physical Physical Physical

- At what layer should we implement reliability?
- End-to-end?
  - Why the right answer?
  - When does it fail?
- "In" the network?
  - Why the wrong answer?
  - When does it work?

Example: Security


7 Application Application
6 Presentation Presentation
5 Session Session
4 Transport Transport
3 Network Network Network Network
2 Data link Data link Data link Data link
1 Physical Physical Physical Physical

- Authentication, integrity, confidentiality
- End-to-end?
  - Why the right answer?
  - When does it fail?
- "In" the network?
  - Why the wrong answer?
  - When does it work?
- But what about intrusion detection, firewalls?



The Internet Engineering Task Force

- Standardization is key to network interoperability
  - The hardware/software of communicating parties are often not built by the same vendor → yet they can communicate because they use the same protocol
- Internet Engineering Task Force
  - Based on working groups that focus on specific issues
- Request for Comments
  - Document that provides information or defines standard
  - Requests feedback from the community
  - Can be “promoted” to standard under certain conditions
    - consensus in the committee
    - interoperating implementations
  - In Project 1 will implement the HTTP protocol

74