

# Operating Systems and Networks

## Networks Part 1: Introduction

Adrian Perrig  
Network Security Group  
ETH Zürich

# Welcome!

- Brief introduction of lecturer and TAs
  - Adrian Perrig, Professor in Department of Computer Science, Director of Network Security Group
  - Qi Li, PhD
  - Jun Han, Tae-Ho Lee, Chris Pappas, Yao Zhang
  - Network security group research area: design and implementation of secure future Internet architecture (SCION project)

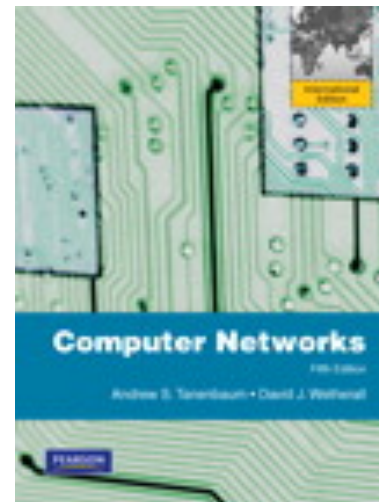
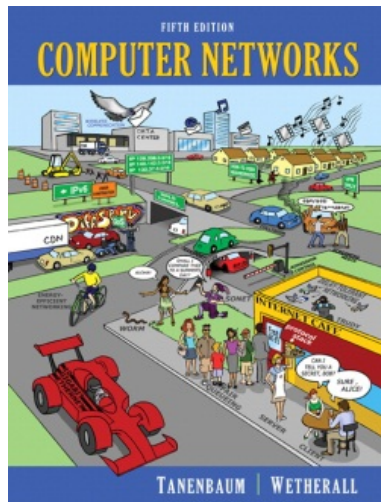


# Lecture Style

- Student interaction is encouraged!
  - Please ask questions if something is unclear
  - Please point out any errors that you spot
  - Please focus on lecture instead of facebook, twitter, etc.
    - Please turn off cell phone during class

# Textbook and Slide Credit

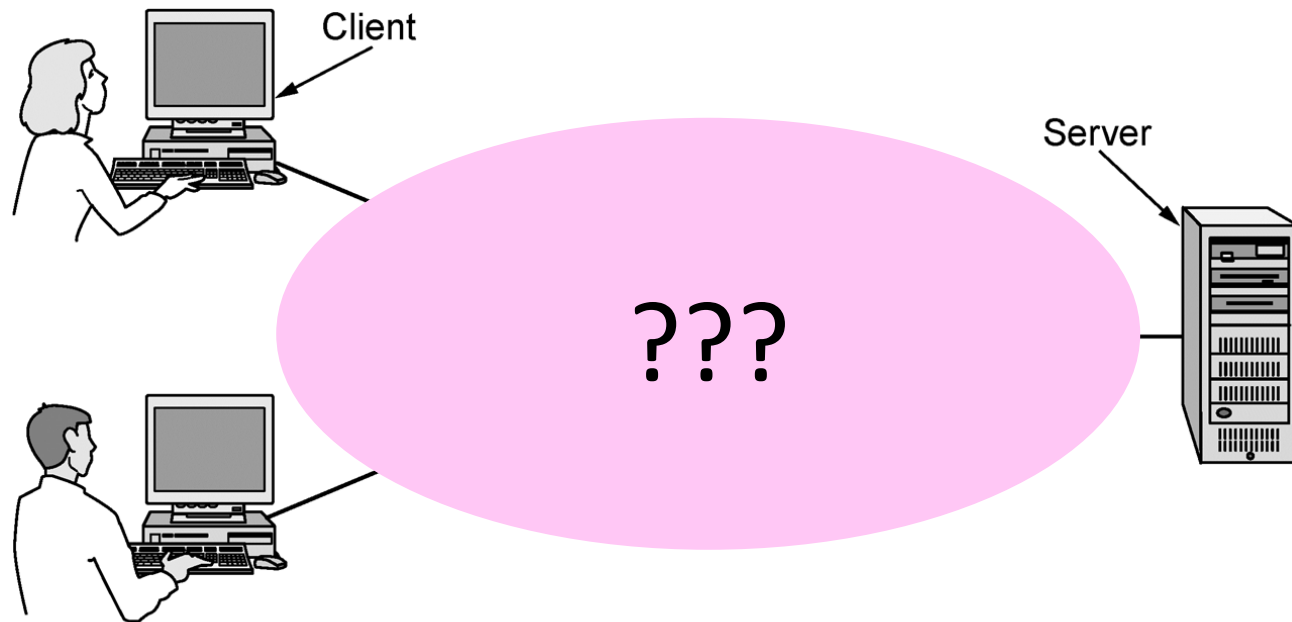
- Textbook: TANENBAUM, ANDREW S.; WETHERALL, DAVID J., COMPUTER NETWORKS, 5th Edition, 2011.
- Slides adapted from slide deck by David Wetherall



# Study Recommendations

- Make list of acronyms, concepts
- Read corresponding sections in text book
  - Available in INFK library
- Participate in exercise sessions, solve homework, and **DO THE LABS!**

# Focus of the course



## Focus of the course (2)

- Three “networking” topics:

Distributed systems
Networking
Communications

- We mainly study the Networking aspects

# The Main Point

1. To learn how the Internet works »
  - What really happens when you “browse the web”?
  - What are TCP/IP, DNS, HTTP, NAT, VPNs, 802.11 etc. anyway?
2. To learn the fundamentals of computer networks

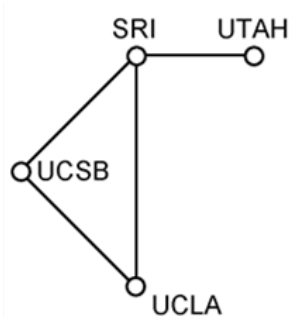


# Why learn about the Internet?

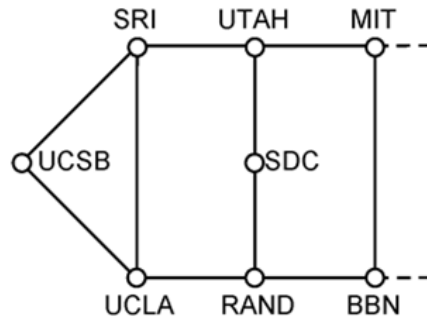
1. Curiosity »
2. Impact on our world »
3. Job prospects!

# From this experimental network ...

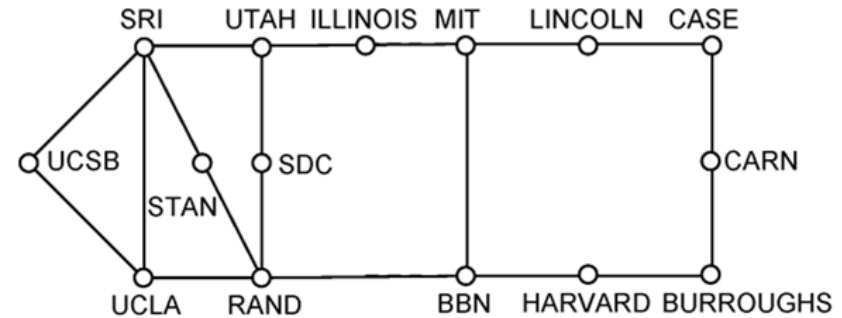
## ARPANET ~1970



(a) Dec. 1969.



(b) July 1970.



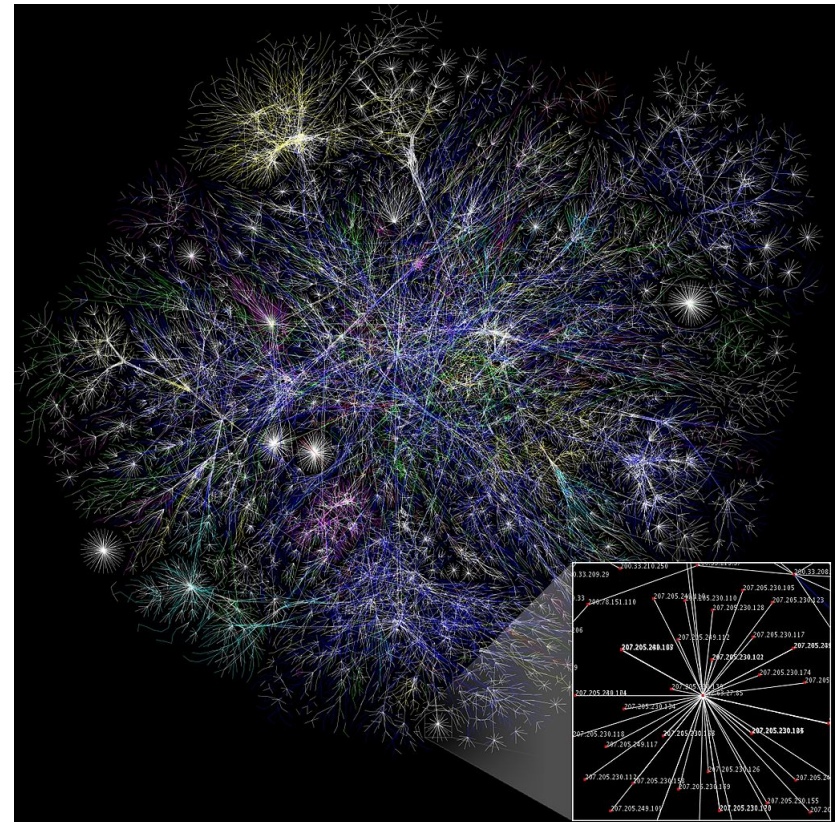
(c) March 1971.

# To this!

## Internet ~2005

- An everyday institution used at work, home, and on-the-go
- Visualization contains millions of links

Computer Networks



Attribution: By The Opte Project [CC-BY-2.5], via Wikimedia Commons

# Internet – Societal Impact

- An enabler of societal change
  - Easy access to knowledge
  - Electronic commerce
  - Personal relationships
  - Discussion without censorship



WIKIPEDIA

**PayPal**

match.com 

**Tor** 

# Internet – Economic impact

- An engine of economic growth
  - Advertising-sponsored search
  - “Long tail” online stores
  - Online marketplaces
  - Crowdsourcing



# The Main Point (2)

1. To learn how the Internet works
2. To learn the fundamentals of computer networks
  - What hard problems must they solve?
  - What design strategies have proven valuable?

# Why learn the Fundamentals?

1. Apply to all computer networks
2. Intellectual interest »
3. Change / reinvention »

# Fundamentals – Intellectual Interest

- Example key problem: Reliability!
  - Any part of the Internet might fail
  - Messages might be corrupted
  - So how do we provide reliability?
- Reliability solutions
  - Codes to detect/correct errors
  - Routing around failures ...



# Fundamentals – Intellectual Interest (2)

<b>Key problem</b>	<b>Example solutions</b>
Reliability despite failures	Codes for error detection/correction (§3.2, 3.3) Routing around failures (§5.2)
Network growth and evolution	Addressing (§5.6) and naming (§7.1) Protocol layering (§1.3)
Allocation of resources like bandwidth	Multiple access (§4.2) Congestion control (§5.3, 6.3)
Security against various threats	Confidentiality of messages (§8.2, 8.6) Authentication of communicating parties (§8.7)

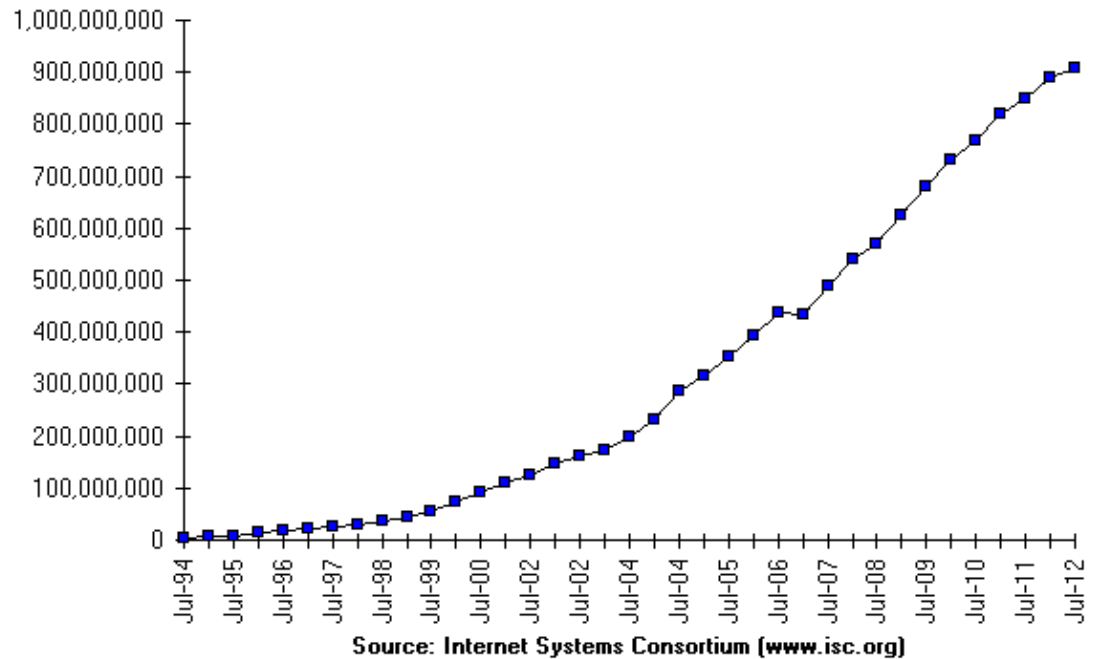
# Fundamentals – Reinvention

- The Internet is constantly being re-invented!
  - Growth over time and technology trends drive upheavals in Internet design and usage »
- Today's Internet is different from yesterday's
  - And tomorrow's will be different again
  - But the fundamentals remain the same

# Fundamentals – Reinvention (2)

Internet Domain Survey Host Count

- At least a billion Internet hosts and growing ...



# Fundamentals – Reinvention (3)

- Examples of upheavals in the past 1-2 decades

<b>Growth / Tech Driver</b>	<b>Upheaval</b>
Emergence of the web	Content Distribution Networks
Digital songs/videos	Peer-to-peer file sharing
Falling cost/bit	Voice-over-IP calling
Many Internet hosts	IPv6
Wireless advances	Mobile devices

# Not a Course Goal

- To learn IT job skills
  - How to configure equipment
    - e.g., Cisco certifications
  - But course material is relevant, and we use hands-on tools

# Example Uses of Networks

- Work:
    - Email, file sharing, printing, ...
  - Home:
    - Movies / songs, news, calls / video / messaging, e-comm
  - Mobile:
    - Calls / texts, games, videos, maps, information access ...
- What do these uses tell us about why we build networks?

# For User Communication

- From the telephone onwards:
  - VoIP (voice-over-IP)
  - Video conferencing
  - Instant messaging
  - Social networking
- Enables remote communication
  - Need low latency for interactivity

# For Resource Sharing

- Many users may access the same underlying resource
  - E.g., 3D printer, search index, machines in the cloud
- More cost effective than dedicated resources per user
  - Even network links are shared via statistical multiplexing »

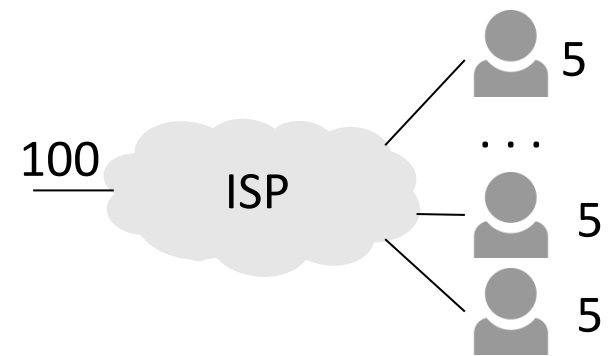


# Statistical Multiplexing

- Sharing of network bandwidth between users according to the statistics of their demand
  - (Multiplexing just means sharing)
  - Useful because users are mostly idle and their traffic is bursty
- Key question:
  - How much does it help?

# Statistical Multiplexing (2)

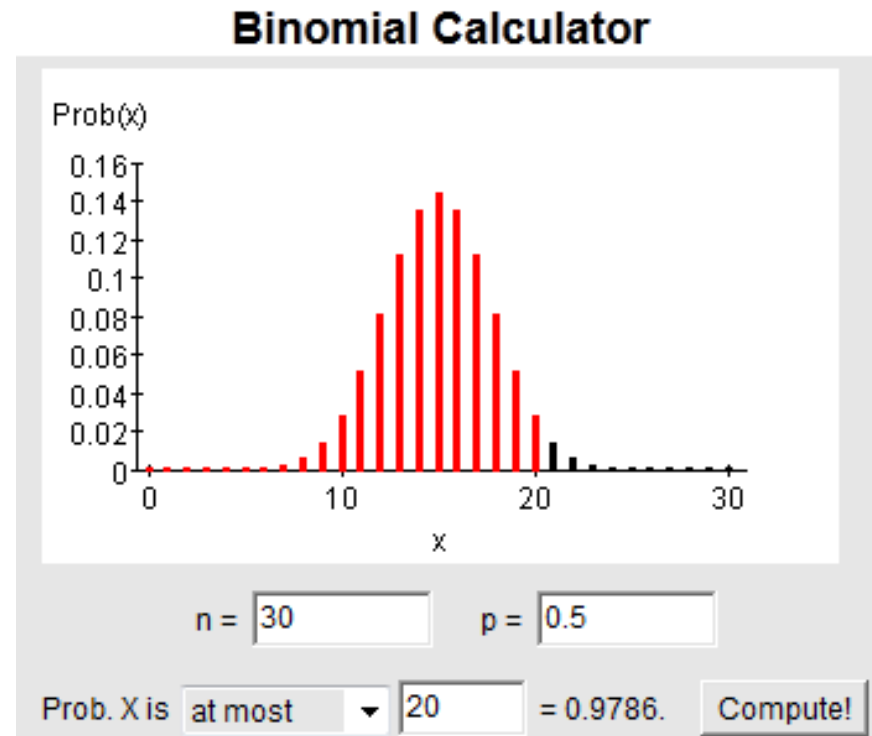
- Example: Users in an ISP network
  - Network has 100 Mbps (units of bandwidth)
  - Each user subscribes to 5 Mbps, for videos
  - But a user is active only 50% of the time ...



- How many users can the ISP support?
  - With dedicated bandwidth for each user:
  - Probability all bandwidth is used: (assuming independent users)

# Statistical Multiplexing (3)

- With 30 independent users, still unlikely (2% chance) to need more than 100 Mbps!
  - Binomial probabilities
- Can serve more users with the same size network
  - Statistical multiplexing gain is 30/20 or 1.5X
  - But may get unlucky; users will have degraded service

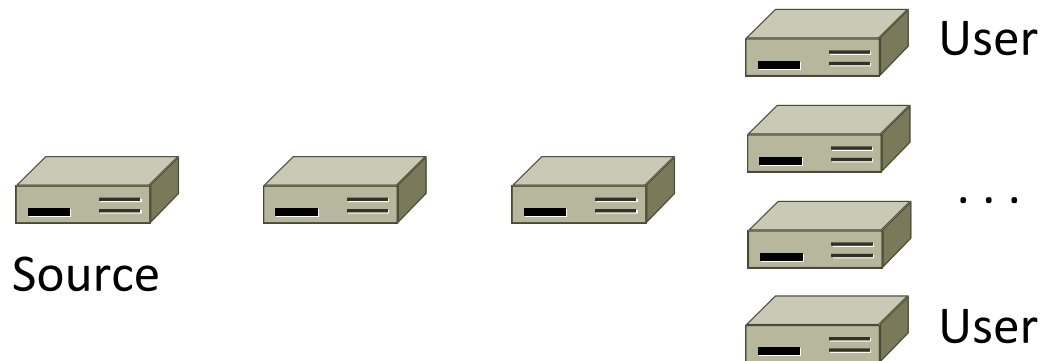


# For Content Delivery

- Same content is delivered to many users
  - Videos (large), songs, apps and upgrades, web pages, ...
- More efficient than sending a copy all the way to each user
  - Uses replicas in the network »

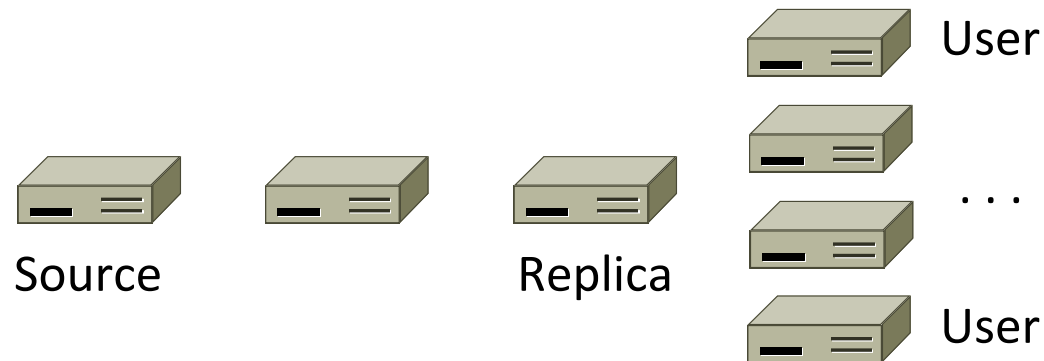
## Content Delivery (2)

- Sending content from the source to 4 users takes  $4 \times 3 = 12$  “network hops” in the example



# Content Delivery (3)

- But sending content via replicas takes only  $4 + 2 = 6$  “network hops”



# For Computer Communication

- To let computers interact with other computers
  - E.g., e-commerce, reservations
- Enables automated information processing across different parties

# To Connect Computers to the Physical World

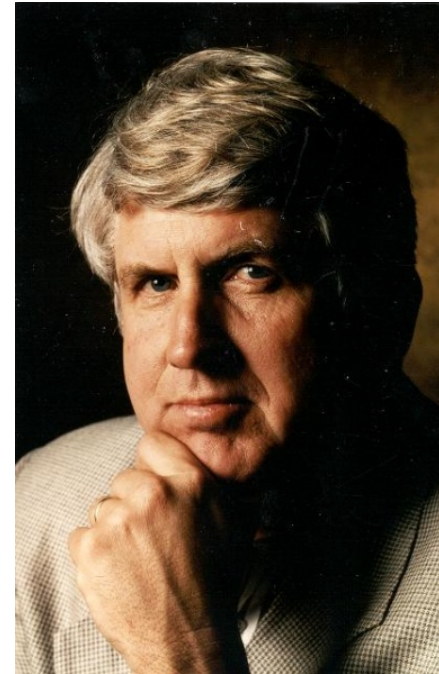
- For gathering sensor data, and for manipulating the world
  - E.g., webcams, location on mobile phones, door locks, ...
- This is a rich, emerging usage



# The Value of Connectivity

- “Metcalfe’s Law” ~1980:
  - The value of a network of  $N$  nodes is proportional to  $N^2$
  - Large networks are relatively more valuable than small ones

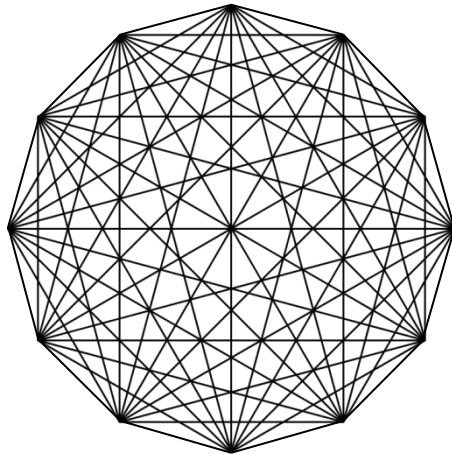
Bob Metcalfe



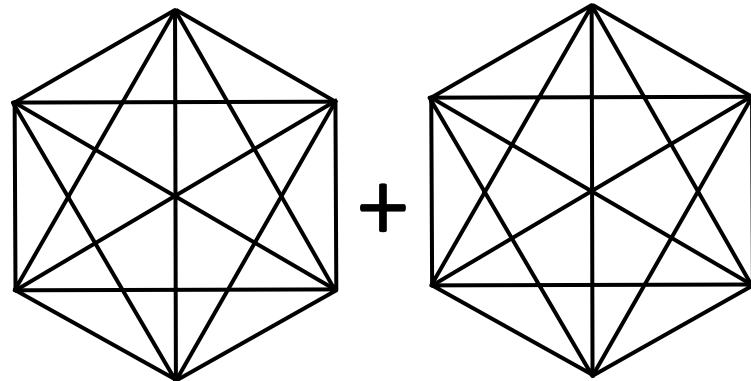
: © 2009 IEEE

# The Value of Connectivity (2)

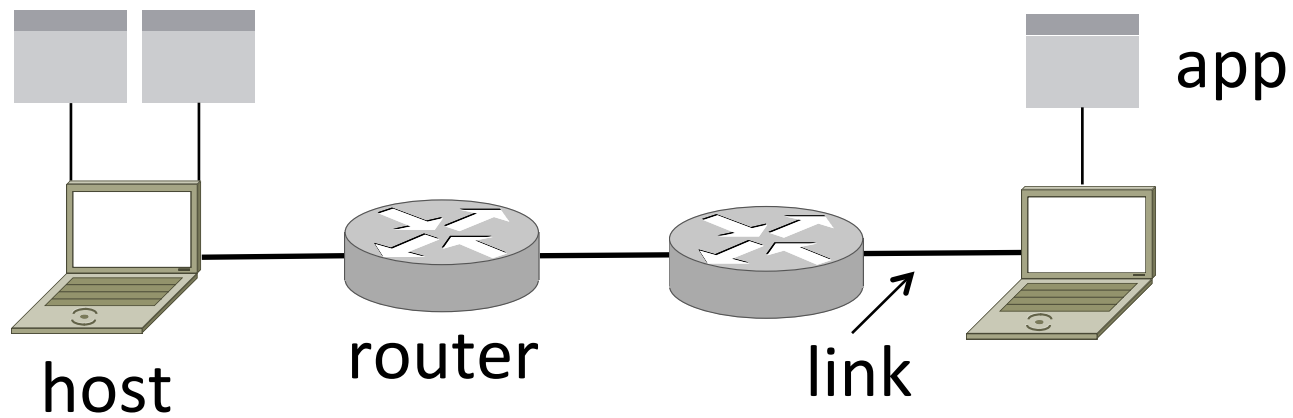
- Example: both sides have 12 nodes, but the left network has more connectivity



vs



# Parts of a Network



# Component Names

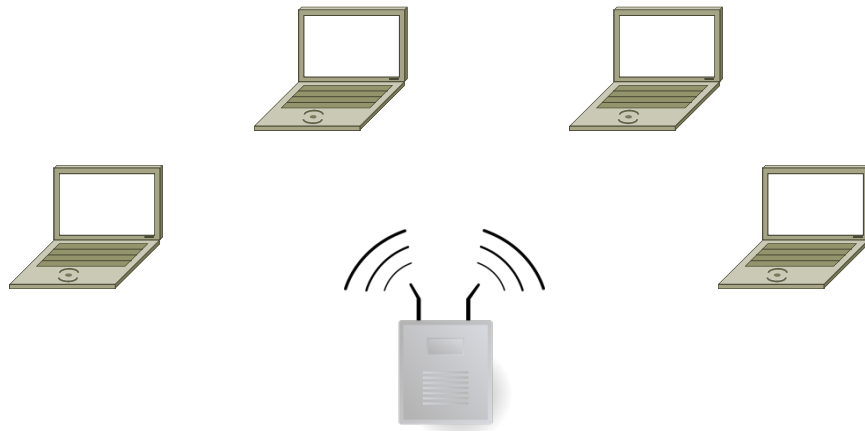
<b>Component</b>	<b>Function</b>	<b>Example</b>
<u>Application</u> , or app, user	Uses the network	Skype, iTunes, Amazon
<u>Host</u> , or end-system, edge device, node, source, sink	Supports apps	Laptop, mobile, desktop
<u>Router</u> , or switch, node, hub, intermediate system	Relays messages between links	Access point, cable/DSL modem
<u>Link</u> , or channel	Connects nodes	Wires, wireless

# Types of Links

- Full-duplex
  - Bidirectional
- Half-duplex
  - Bidirectional
- Simplex
  - unidirectional

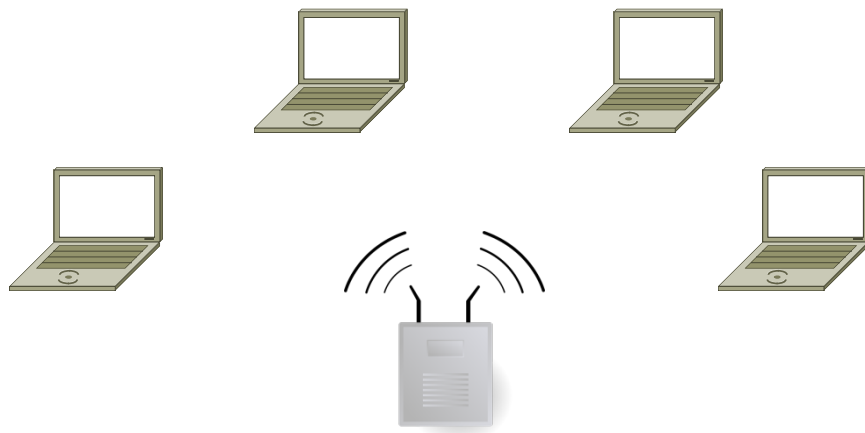
# Wireless Links

- Message is broadcast
  - Received by all nodes in range
  - Not a good fit with our model



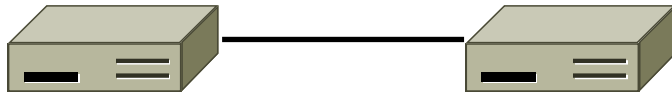
## Wireless Links (2)

- Often show logical links
  - Not all possible connectivity



# A Small Network

- Connect a couple of computers



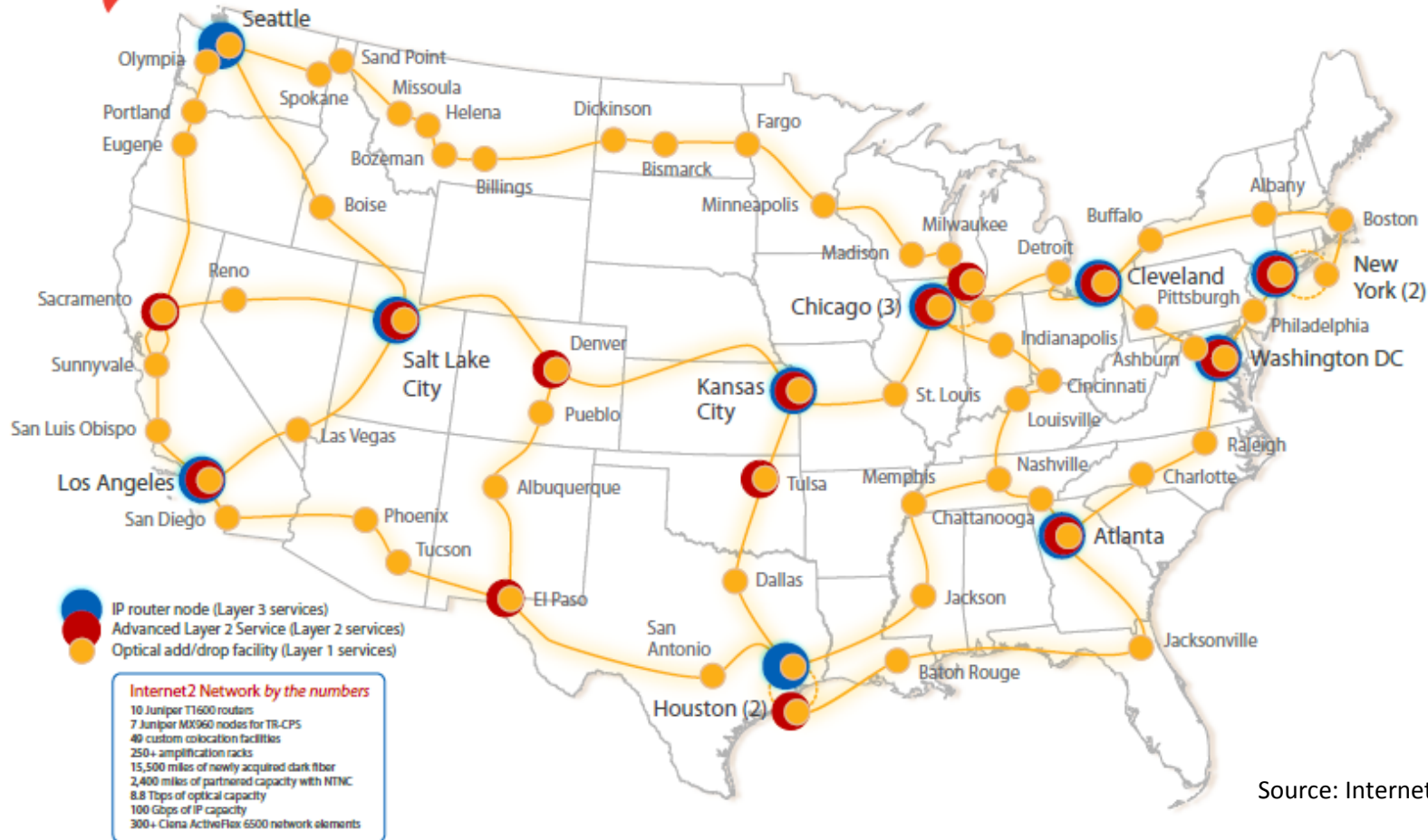
- Next, a large network ...





# Internet2 Network Infrastructure Topology

October 2012



IN SUPPORT OF  
**U.S.UCAN**

NETWORK  
PARTNERS

**ciena**



INDIANA UNIVERSITY

**Infinera**

**JUNIPER**



Source: Internet2

# Example Networks

- Commonly known by type of technology or their purpose
- [see how many you can give]

# Example Networks (2)

- WiFi (802.11)
- Enterprise / Ethernet
- ISP (Internet Service Provider)
- Cable / DSL
- Mobile phone / cellular (2G, 3G, 4G)
- Bluetooth
- Telephone
- Satellite ...

# Network names by scale

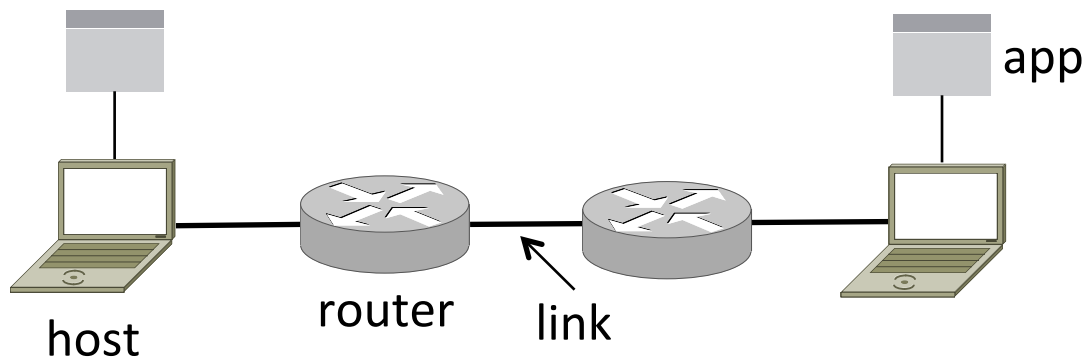
<b>Scale</b>	<b>Type</b>	<b>Example</b>
Vicinity	<u>PAN</u> (Personal Area Network)	Bluetooth (e.g., headset)
Building	<u>LAN</u> (Local Area Network)	WiFi, Ethernet
City	<u>MAN</u> (Metropolitan Area Network)	Cable, DSL
Country	<u>WAN</u> (Wide Area Network)	Large ISP
Planet	The Internet (network of all networks)	The Internet!

# Internetworks

- An internetwork, or internet, is what you get when you join networks together
  - Just another network
- The Internet (capital “I”) is the internet we all use

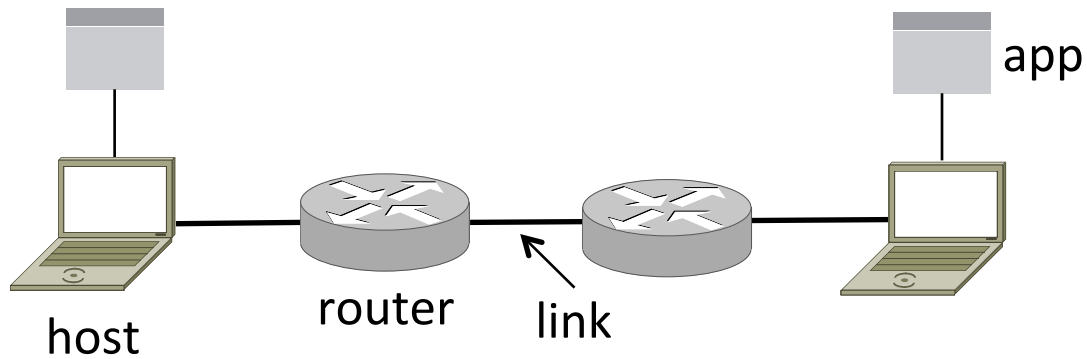
# Network Boundaries

- What part is the “network”?



# Network Boundaries (2)

- What part represents an “ISP”?



# Network Boundaries (3)

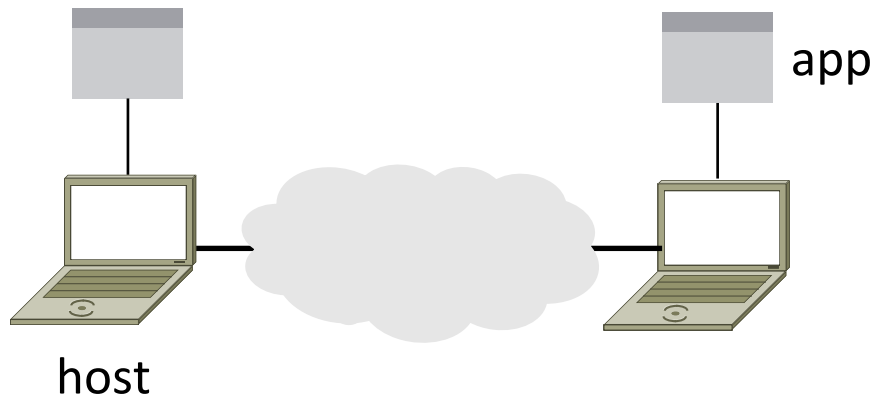
- Cloud as a generic network





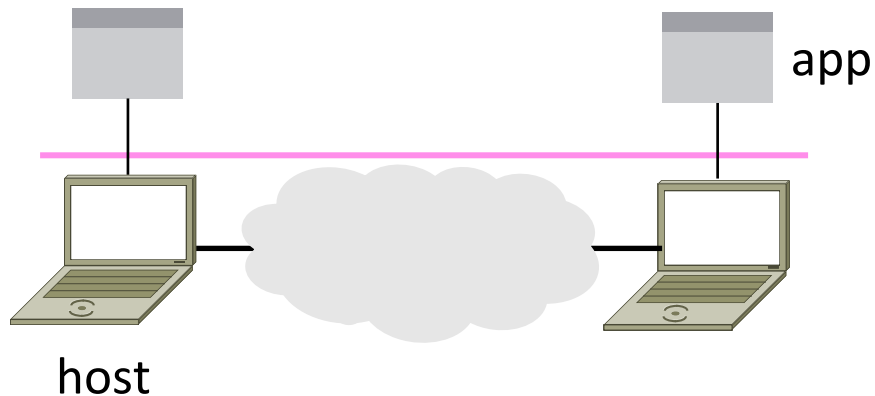
# Key Interfaces

- Between (1) apps and network, and (2) network components
  - More formal treatment later on



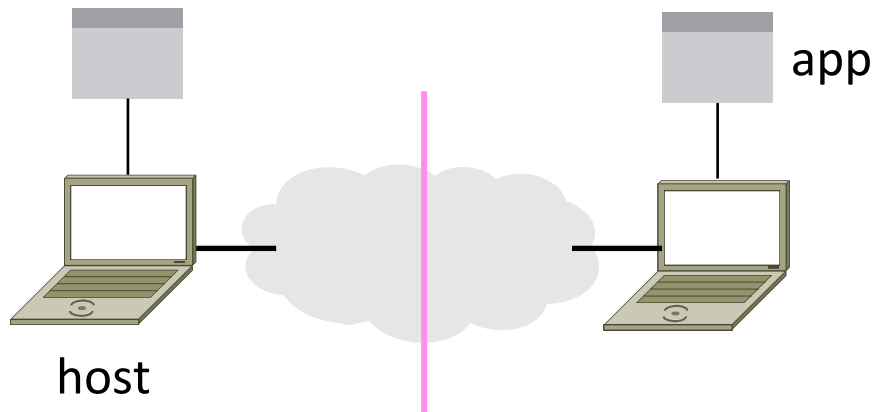
# Key Interfaces (2)

1. Network-application interfaces define how apps use the network
  - Sockets are widely used in practice



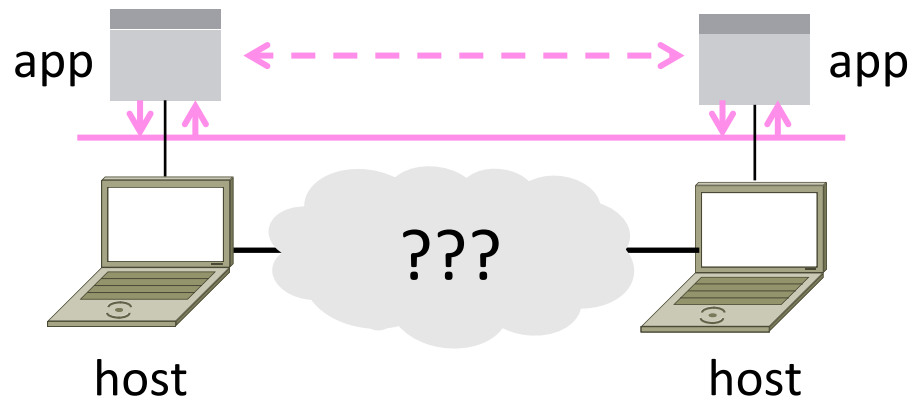
## Key Interfaces (3)

2. Network-network interfaces define how nodes work together
  - Traceroute can peek inside the network



# Network Service API Hides Details

- Apps talk to other apps with no real idea of what is inside the network
  - This is good! But you may be curious ...



# Traceroute

- Widely used command-line tool to let hosts peek inside the network
  - On all OSes (tracert on Windows)
  - Developed by Van Jacobson ~1987
  - Uses a network-network interface (IP) in ways we will explain later

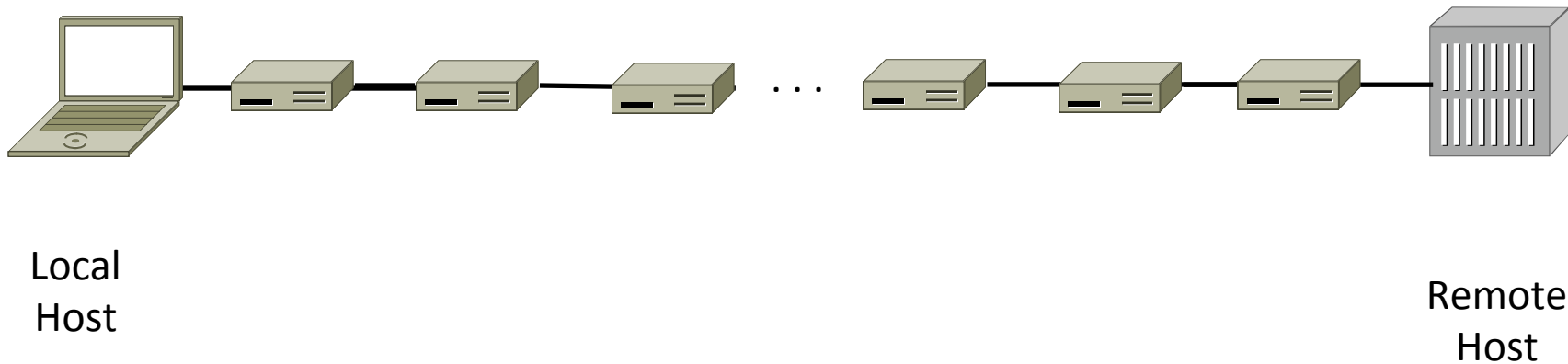
Van Jacobson



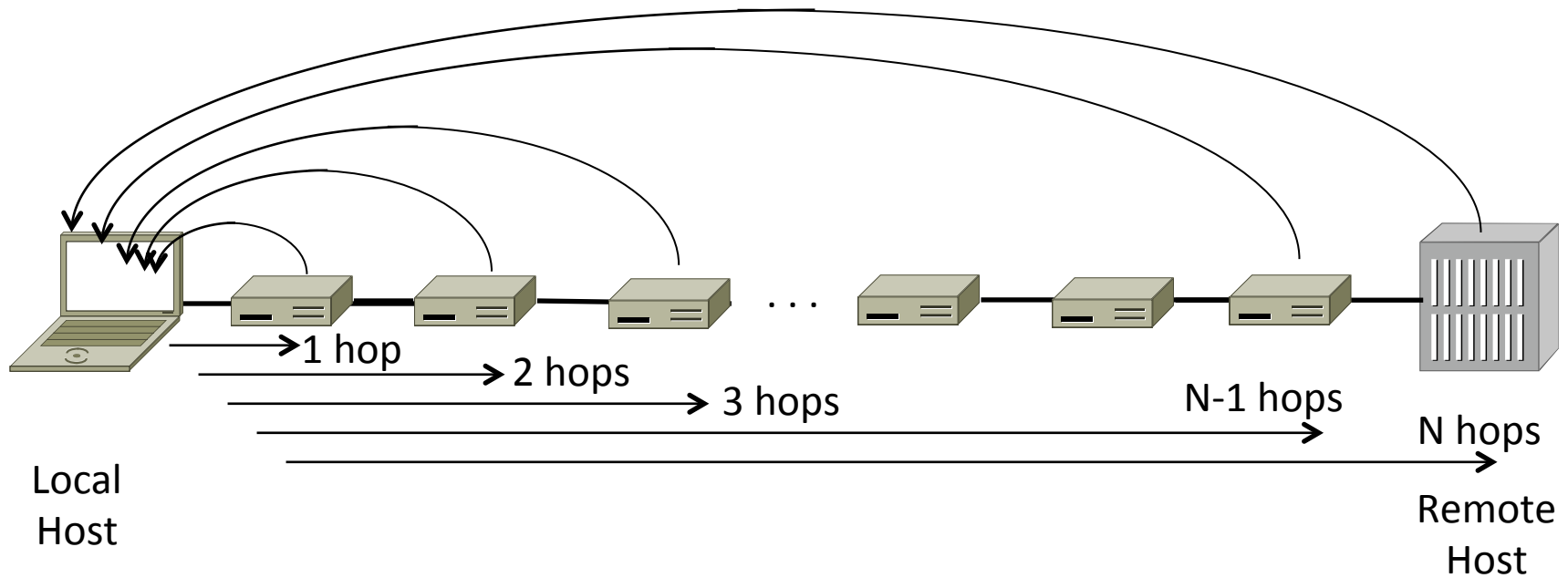
: Credit: Wikipedia (public domain)

# Traceroute (2)

- Probes successive hops to find network path



# Traceroute (3)



# Using Traceroute

```
Administrator: Command Prompt
C:\Users\djw>tracert www.uw.edu

Tracing route to www.washington.edu [128.95.155.134]
over a maximum of 30 hops:

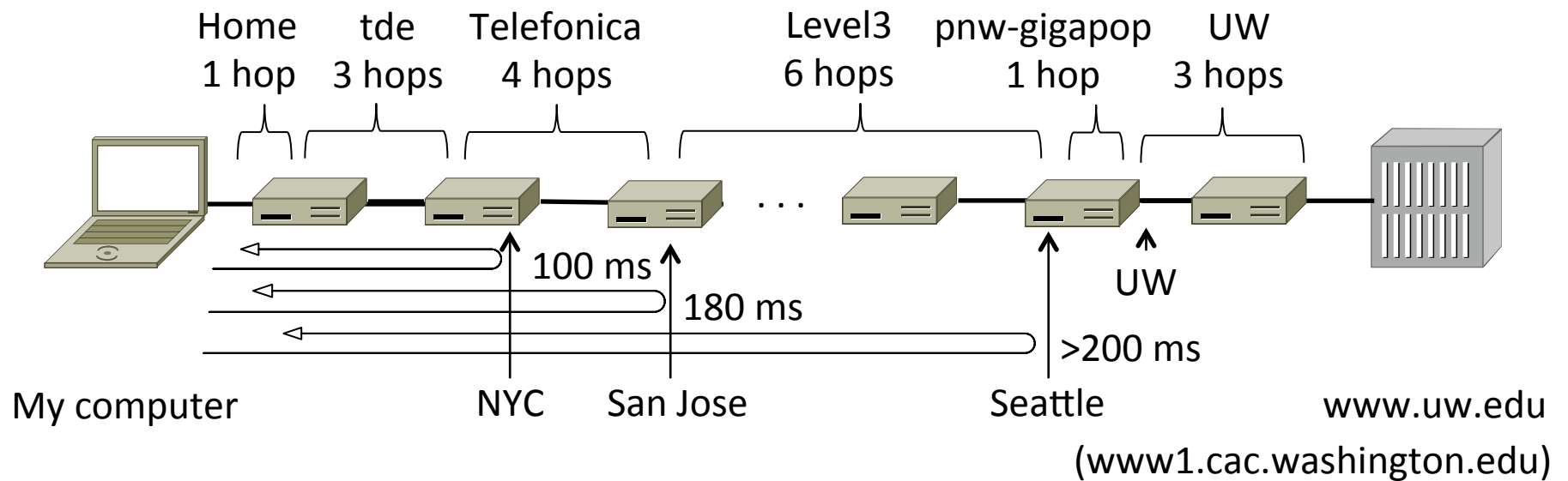
  0  1 ms  <1 ms  2 ms  192.168.1.1
  1  8 ms  8 ms  9 ms  88.Red-80-58-67.staticIP.rima-tde.net [80.58.67.88]
  2 16 ms  5 ms 11 ms 169.Red-80-58-78.staticIP.rima-tde.net [80.58.78.169]
  3 12 ms 12 ms 13 ms 217.Red-80-58-87.staticIP.rima-tde.net [80.58.87.217]
  4  5 ms 11 ms  6 ms  et-1-0-0-1-101-GRITBCNES1.red.telefonica-wholesale.net [94.142.103.205]
  5  40 ms 38 ms 38 ms 176.52.250.226
  6 108 ms 106 ms 136 ms xe-6-0-2-0-grtnycpt2.red.telefonica-wholesale.net [213.140.43.9]
  7 180 ms 179 ms 182 ms Xe9-2-0-0-grtpaopx2.red.telefonica-wholesale.net [94.142.118.178]
  8 178 ms 175 ms 176 ms te-4-2.car1.SanJose2.Level3.net [4.59.0.225]
  9 190 ms 186 ms 187 ms vlan80.csw3.SanJose1.Level3.net [4.69.152.190]
 10 185 ms 185 ms 187 ms ae-82-82.ebr2.SanJose1.Level3.net [4.69.153.25]
 11 268 ms 205 ms 207 ms ae-7-7.ebr1.Seattle1.Level3.net [4.69.132.50]
 12 334 ms 202 ms 195 ms ae-12-51.car2.Seattle1.Level3.net [4.69.147.132]
 13 195 ms 196 ms 195 ms PACIFIC-NOR.car2.Seattle1.Level3.net [4.53.146.142]
 14 197 ms 195 ms 196 ms ae0--4000.iccr-stlwa01-02.infra.pnw-gigapop.net [209.124.188.132]
 15 196 ms 196 ms 195 ms vl4000.uwbr-ads-01.infra.washington.edu [209.124.188.133]
 16 * * * Request timed out.
 17 201 ms 194 ms 196 ms ae4--583.uwar-ads-1.infra.washington.edu [128.95.155.131]
 18 197 ms 196 ms 195 ms www1.cac.washington.edu [128.95.155.134]

Trace complete.
```



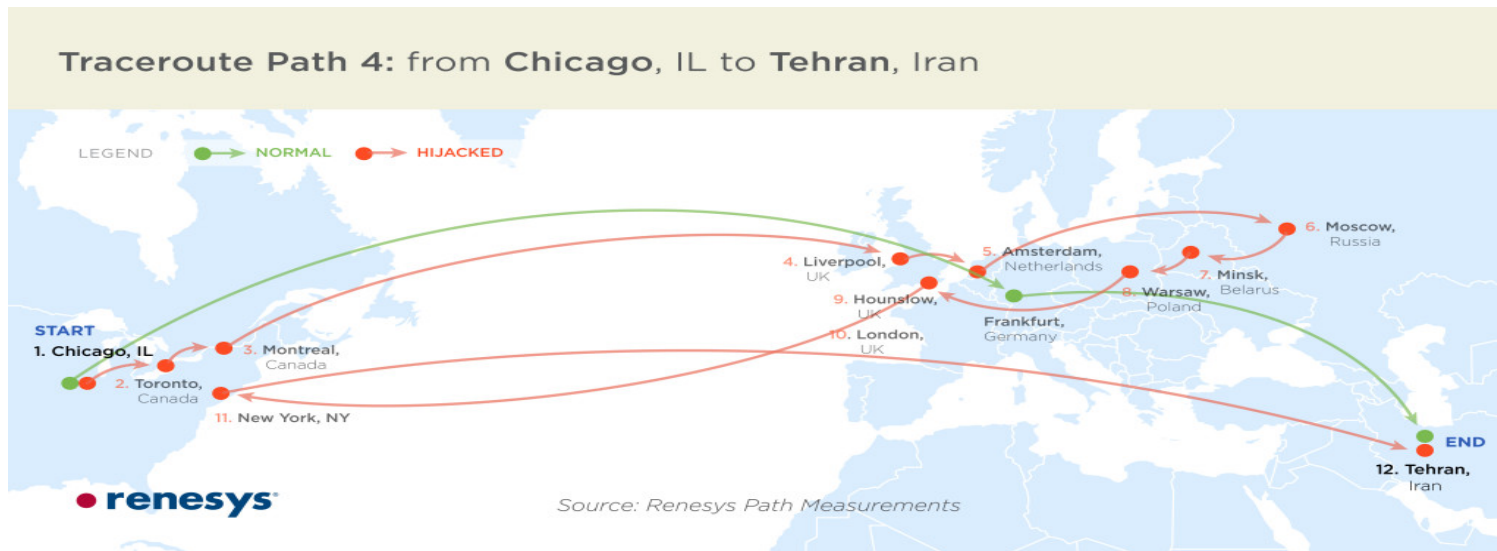
# Using Traceroute (2)

- ISP names and places are educated guesses



# Recently, some strange paths observed

- <http://www.renesys.com/2013/11/mitm-internet-hijacking/>



# Networks Need Modularity (§1.3)

- The network does much for apps:
  - Make and break connections
  - Find a path through the network
  - Transfers information reliably
  - Transfers arbitrary length information
  - Send as fast as the network allows
  - Shares bandwidth among users
  - Secures information in transit
  - Lets many new hosts be added
  - ...

# Networks Need Modularity

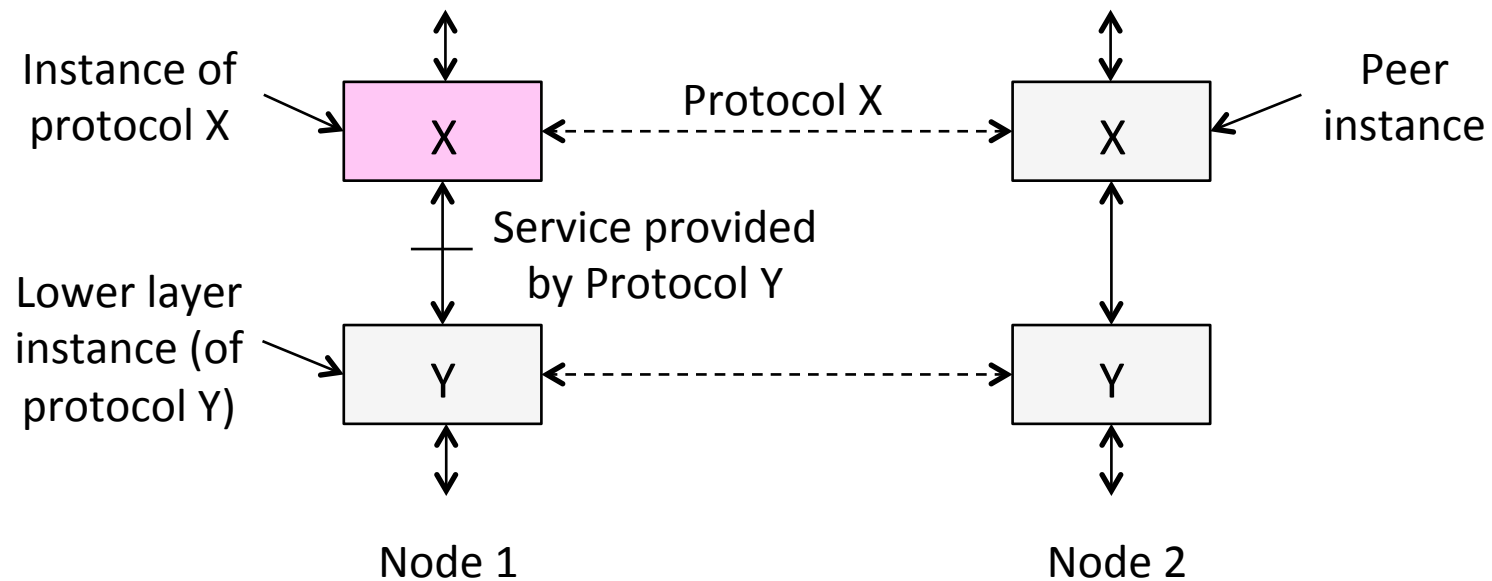
- The network does much for apps:
  - Make and break connections
  - We need a form of modularity, to help manage complexity and support reuse
  - Secures information in transit
  - Lets many new hosts be added
  - ...

# Protocols and Layers

- Protocols and layering is the main structuring method used to divide up network functionality
  - Each instance of a protocol talks virtually to its peer using the protocol
  - Each instance of a protocol uses only the services of the lower layer

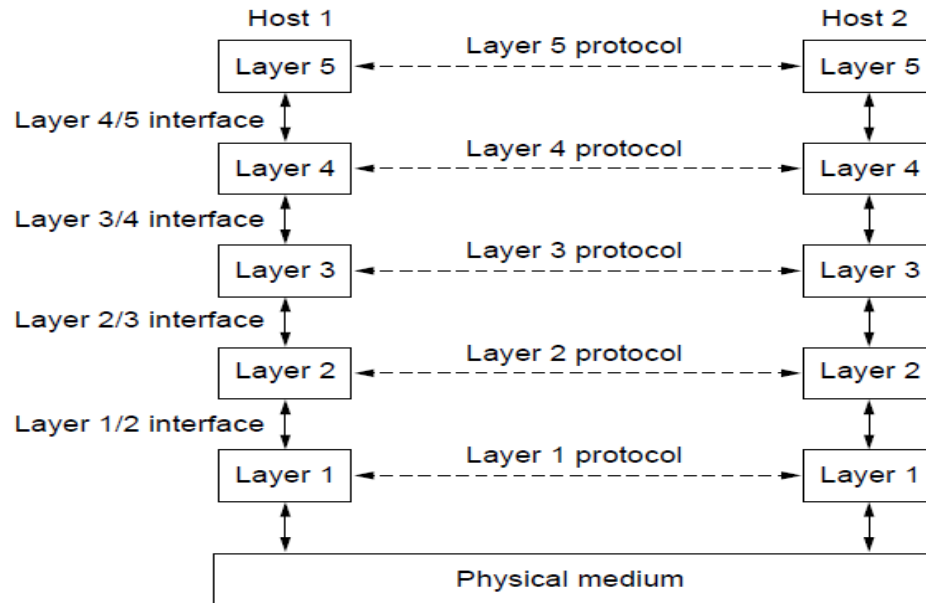
# Protocols and Layers (2)

- Protocols are horizontal, layers are vertical



# Protocols and Layers (3)

- Set of protocols in use is called a protocol stack



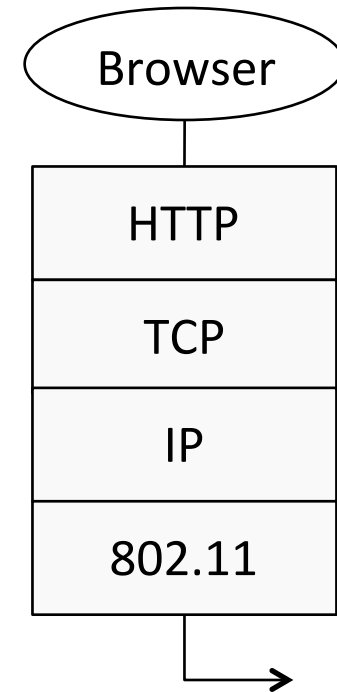
# Protocols and Layers (4)

- Protocols you've probably heard of:
  - TCP, IP, 802.11, Ethernet, HTTP, SSL, DNS, ... and many more
- An example protocol stack
  - Used by a web browser on a host that is wirelessly connected to the Internet



# Protocols and Layers (5)

- Protocols you've probably heard of:
  - TCP, IP, 802.11, Ethernet, HTTP, SSL, DNS, ... and many more
- An example protocol stack
  - Used by a web browser on a host that is wirelessly connected to the Internet

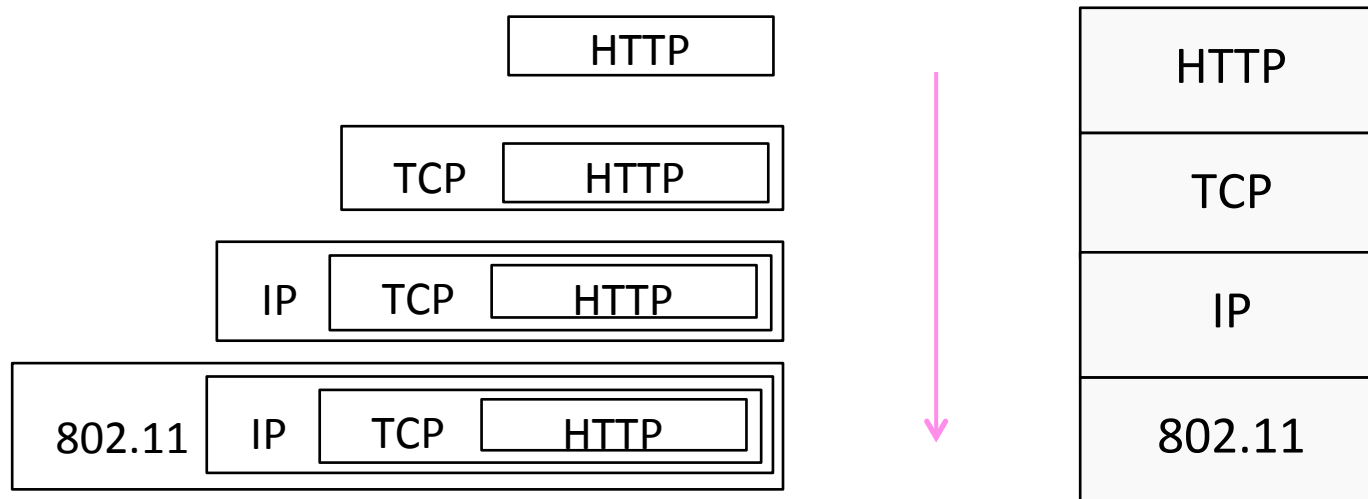


# Encapsulation

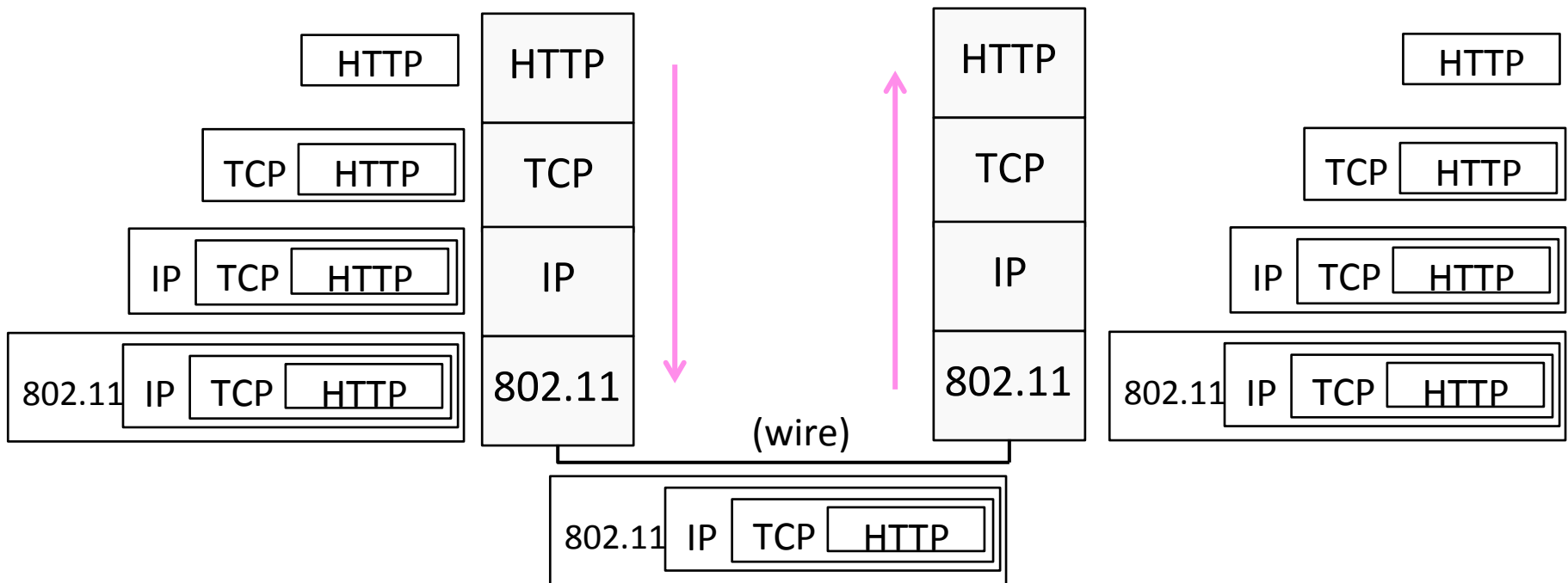
- Encapsulation is the mechanism used to effect protocol layering
  - Lower layer wraps higher layer content, adding its own information to make a new message for delivery
  - Like sending a letter in an envelope; postal service doesn't look inside

# Encapsulation (2)

- Message “on the wire” begins to look like an onion
  - Lower layers are outermost

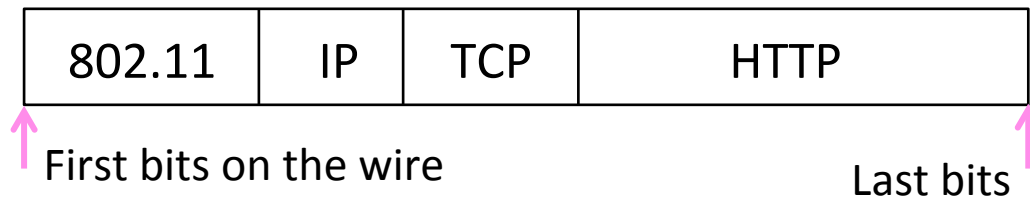


# Encapsulation (3)



# Encapsulation (4)

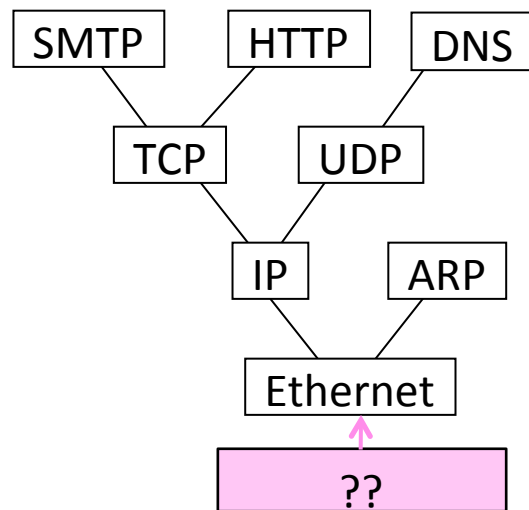
- Normally draw message like this:
  - Each layer adds its own header



- More involved in practice
  - Trailers as well as headers, encrypt/compress contents
  - Segmentation (divide long message) and reassembly

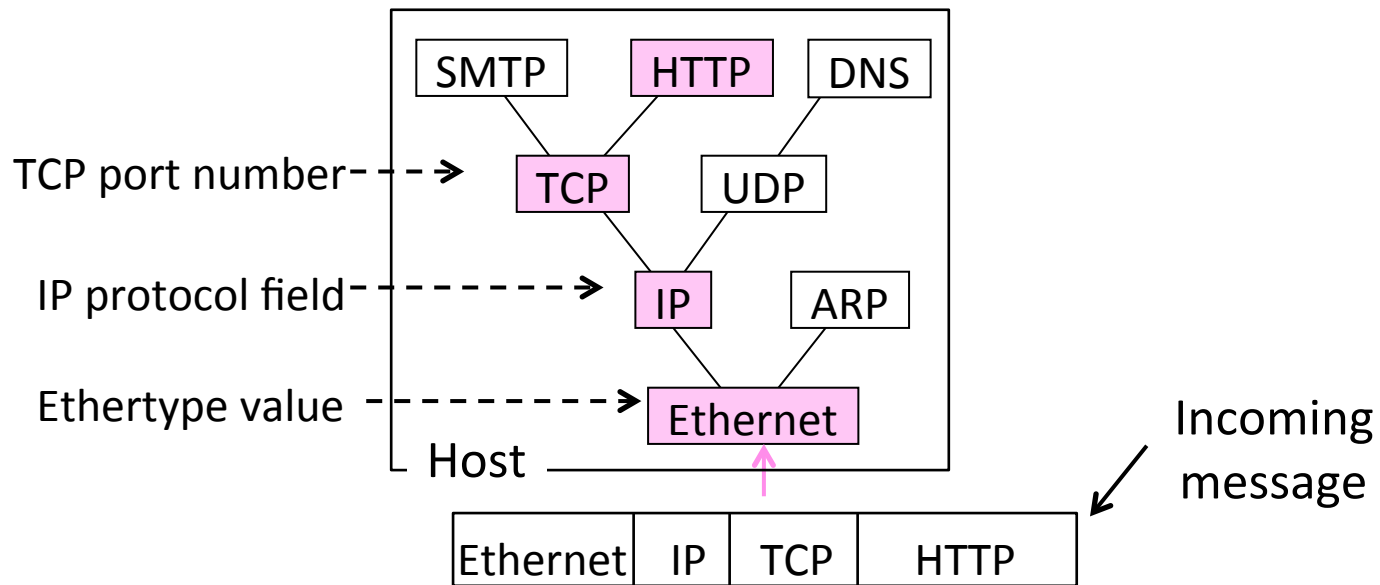
# Demultiplexing

- Incoming message must be passed to the protocols that it uses



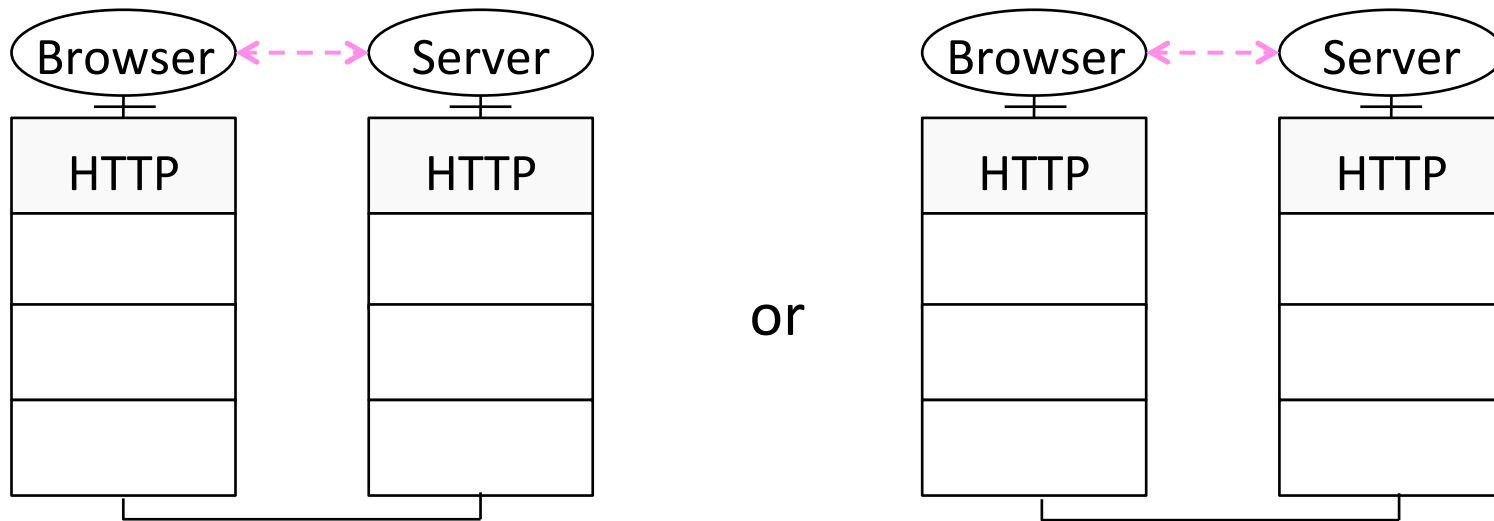
# Demultiplexing (2)

- Done with demultiplexing keys in the headers



# Advantage of Layering

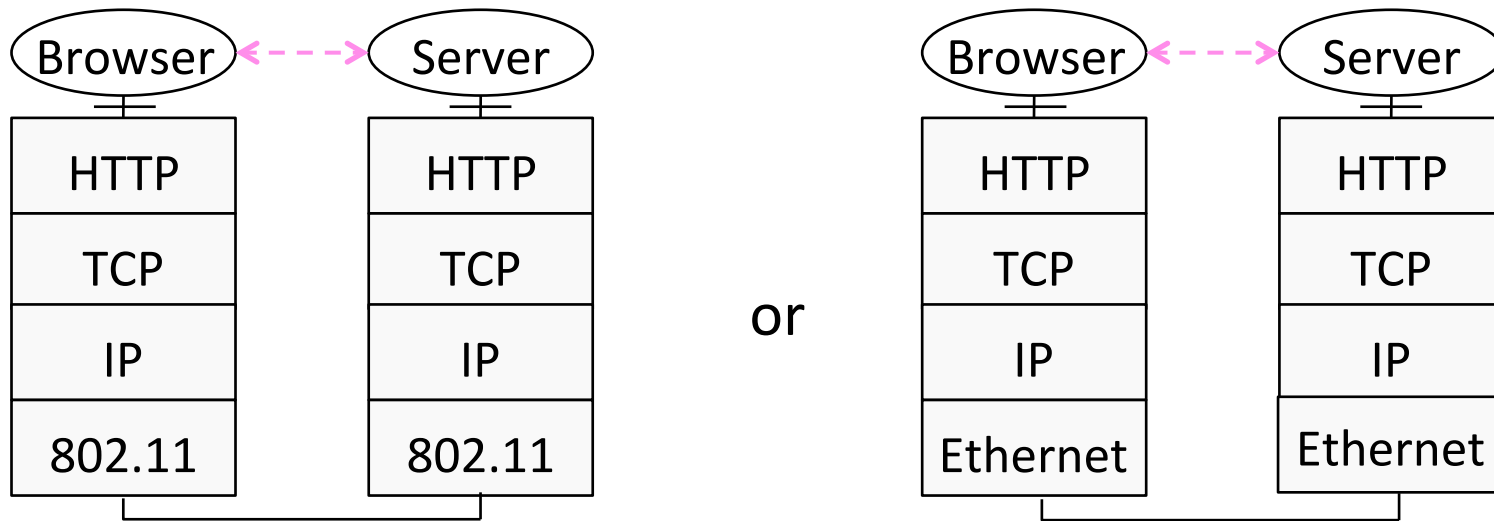
- Information hiding and reuse





# Advantage of Layering (2)

- Information hiding and reuse



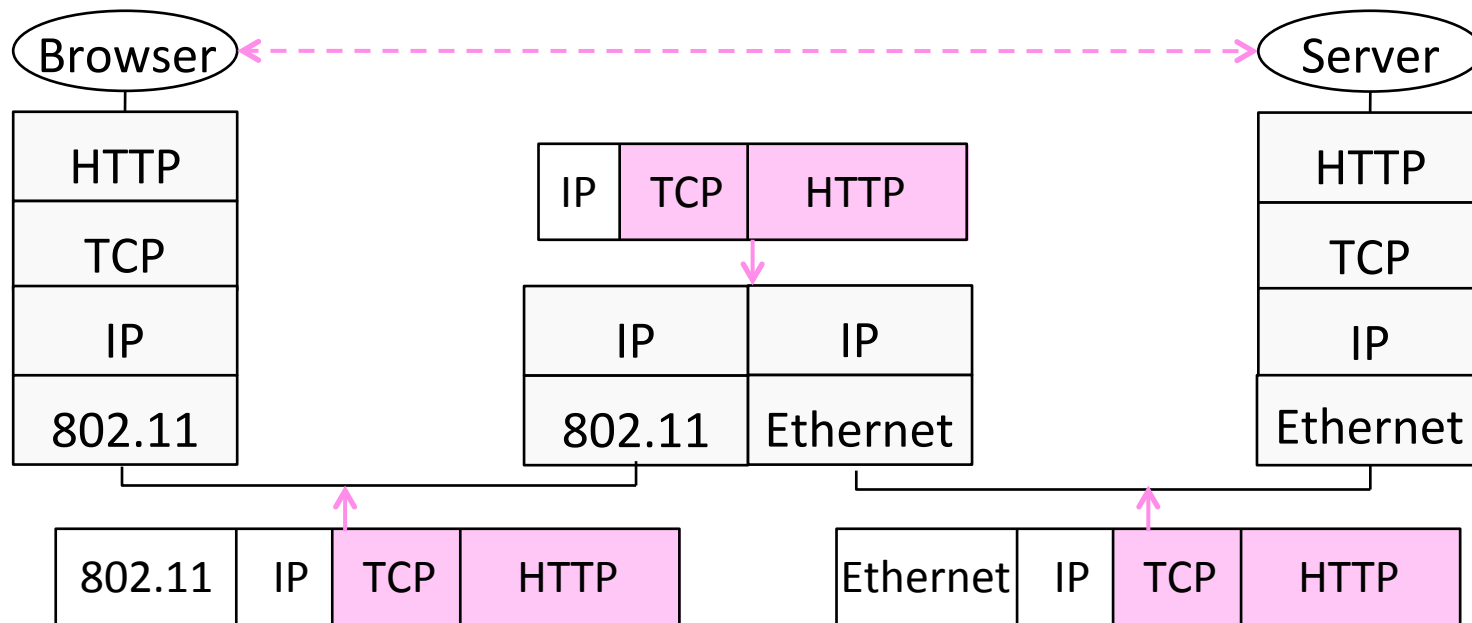
# Advantage of Layering (3)

- Using information hiding to connect different systems



# Advantage of Layering (4)

- Using information hiding to connect different systems



# Disadvantage of Layering

- Adds overhead
  - But minor for long messages
- Hides information
  - App might care whether it is running over wired or wireless!

## A Little Guidance Please ... (§1.4, §1.6)

- What functionality should we implement at which layer?
  - This is a key design question
  - Reference models provide frameworks that guide us »

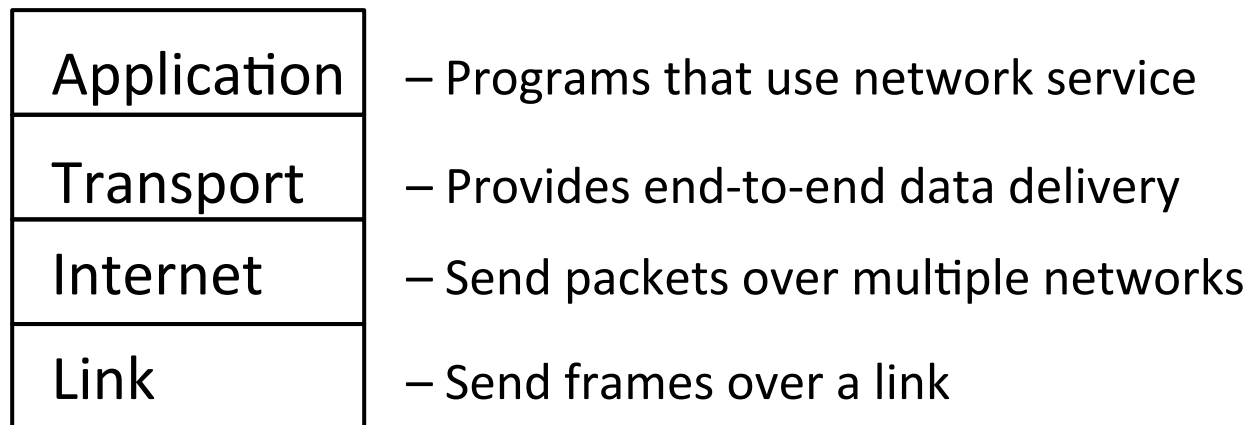
# OSI “7 layer” Reference Model

- A principled, international standard, to connect systems
  - Influential, but not used in practice. (Whoops)

7	Application	– Provides functions needed by users
6	Presentation	– Converts different representations
5	Session	– Manages task dialogs
4	Transport	– Provides end-to-end delivery
3	Network	– Sends packets over multiple links
2	Data link	– Sends frames of information
1	Physical	– Sends bits as signals

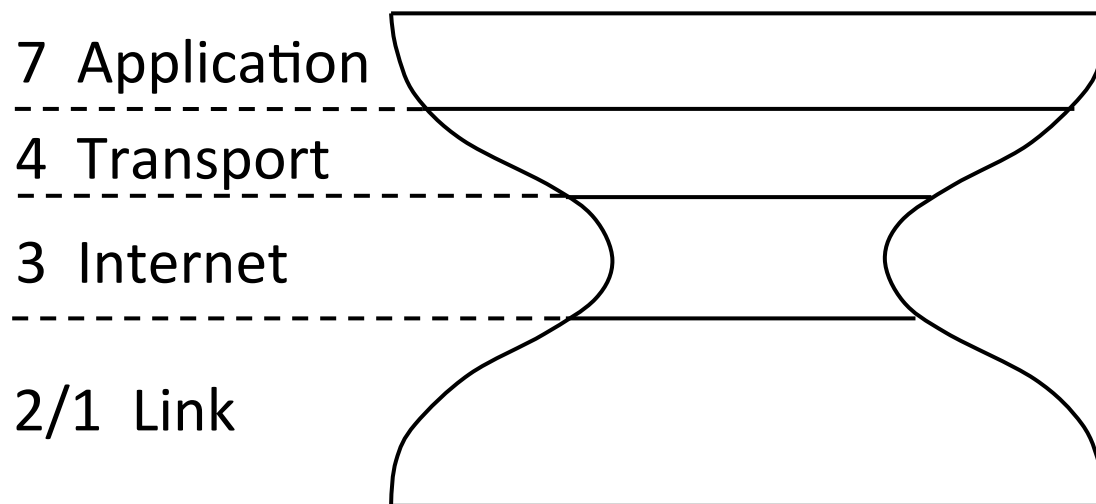
# Internet Reference Model

- A four layer model based on experience; omits some OSI layers and uses IP as the network layer.



# Internet Reference Model (2)

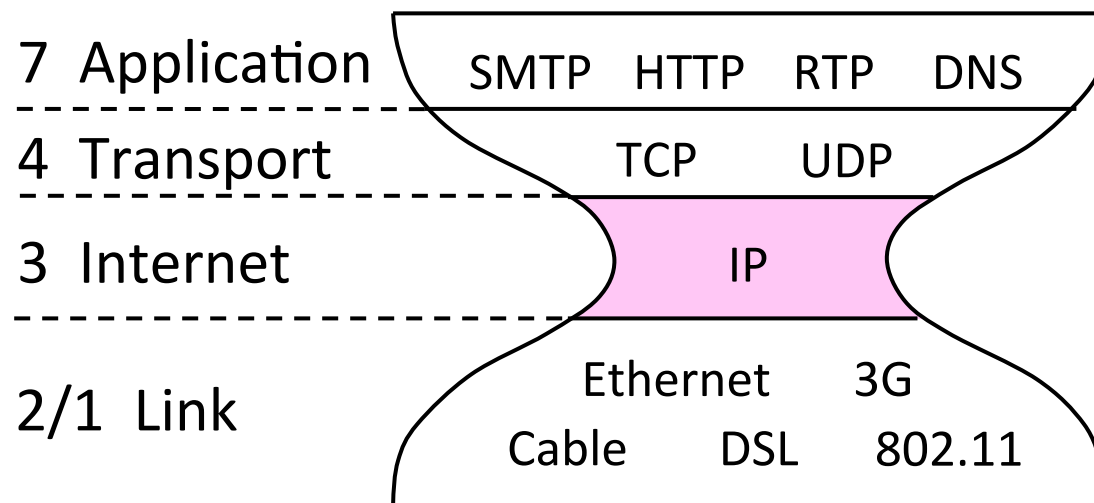
- With examples of common protocols in each layer





# Internet Reference Model (3)

- IP is the “narrow waist” of the Internet
  - Supports many different links below and apps above



# Standards Bodies

- Where all the protocols come from!
  - Focus is on interoperability

Body	Area	Examples
ITU	Telecom	G.992, ADSL H.264, MPEG4
IEEE	Communications	802.3, Ethernet 802.11, WiFi
IETF	Internet	RFC 2616, HTTP/1.1 RFC 1034/1035, DNS
W3C	Web	HTML5 standard CSS standard

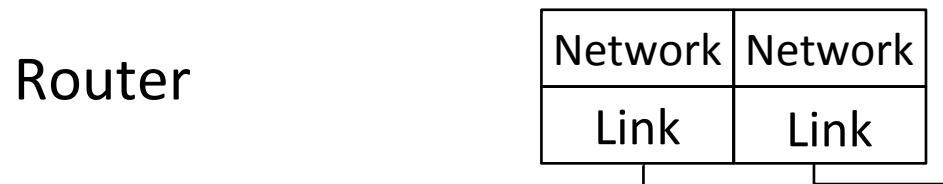
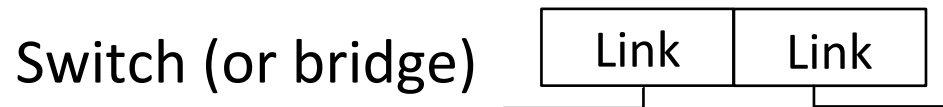
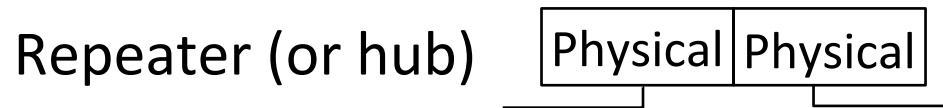
# Layer-based Names

- For units of data:

<b>Layer</b>	<b>Unit of Data</b>
Application	Message
Transport	Segment
Network	Packet
Link	Frame
Physical	Bit

# Layer-based Names (2)

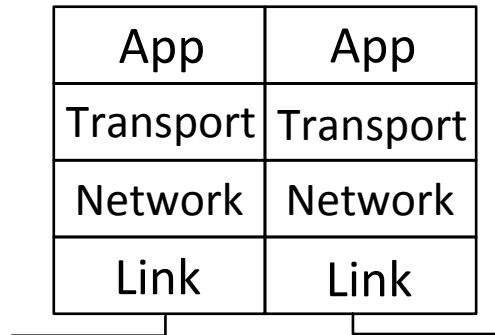
- For devices in the network:



# Layer-based Names (3)

- For devices in the network:

Proxy or  
middlebox  
or gateway



But they all  
look like this!



# A Note About Layers

- They are guidelines, not strict
  - May have multiple protocols working together in one layer
  - May be difficult to assign a specific protocol to a layer

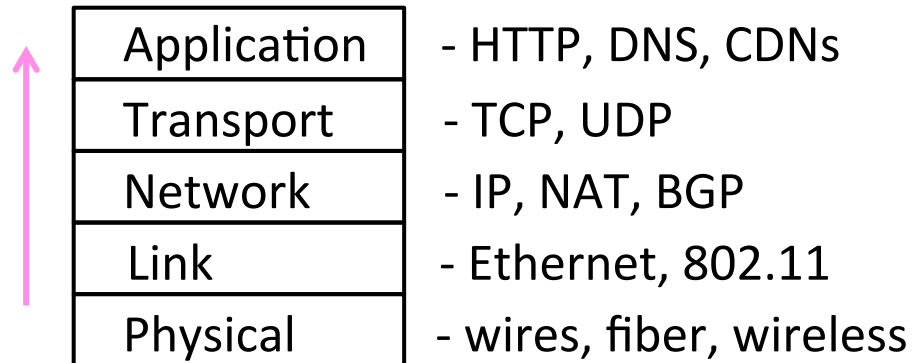
# Course Reference Model

- We mostly follow the Internet
  - A little more about the Physical layer, and alternatives

7	Application	– Programs that use network service
4	Transport	– Provides end-to-end data delivery
3	Network	– Send packets over multiple networks
2	Link	– Send frames over one or more links
1	Physical	– Send bits using signals

# Lecture Progression

- Bottom-up through the layers:



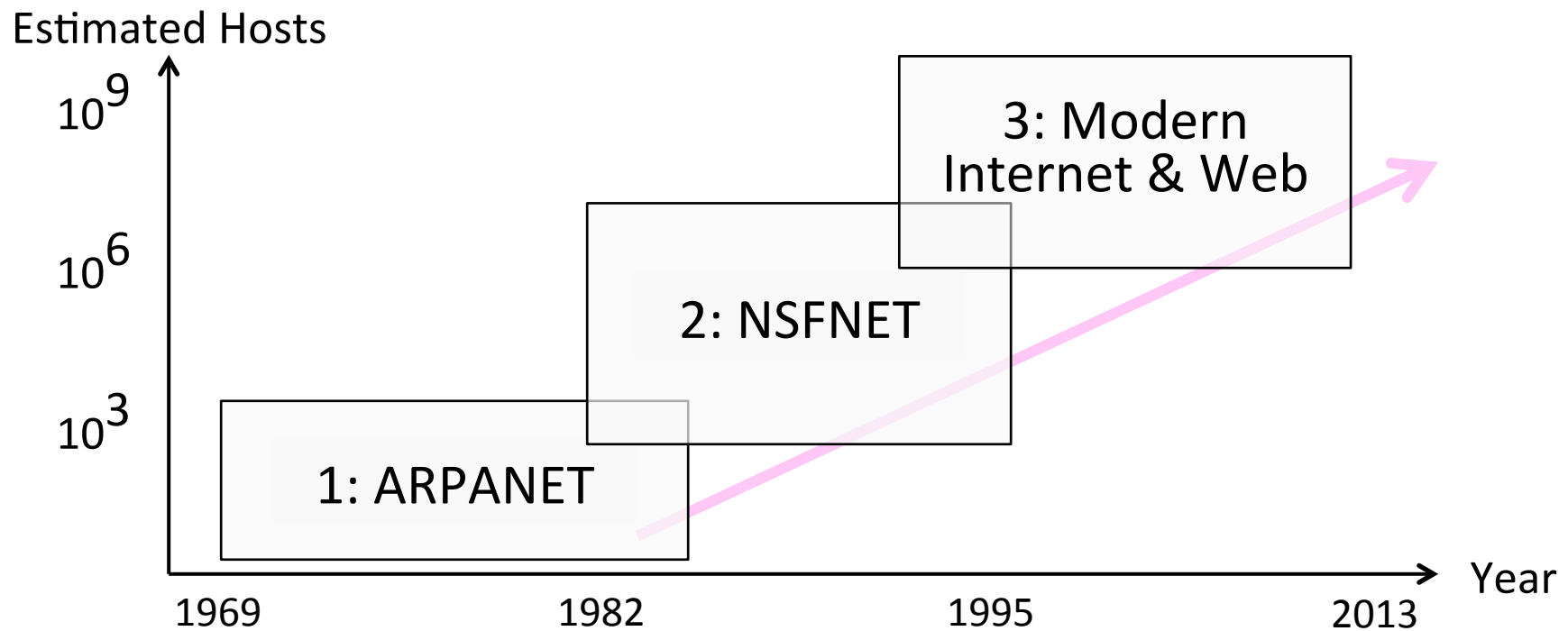
- Followed by more detail on:
  - Quality of service, Security (VPN, SSL)



# Bonus Material: History of the Internet

## History of the Internet (§1.5.1)

# Rough Internet Timeline



# The Beginning – ARPANET

- ARPANET by U.S. DoD was the precursor to the Internet
  - Motivated for resource sharing
  - Launched with 4 nodes in 1969, grew to hundreds of hosts
  - First “killer app” was email

# ARPANET – Influences

- Leading up to the ARPANET (1960s):
  - Packet switching (Kleinrock, Davies), decentralized control (Baran)

Paul Baran



Credit: Internet Hall of Fame

Donald Davies



Credit: Internet Hall of Fame

Len Kleinrock

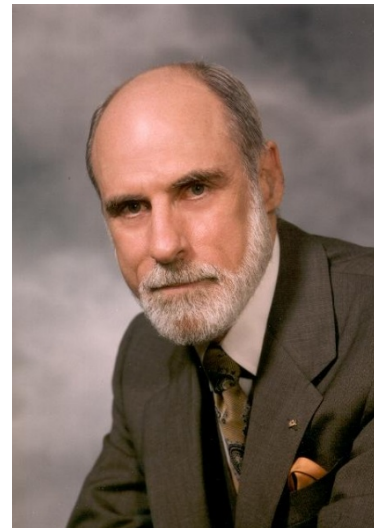


Credit: Internet Hall of Fame

# ARPANET – Influences (2)

- In the early ARPANET
  - Internetworking became the basis for the Internet
  - Pioneered by Cerf & Kahn in 1974, later became TCP/IP
  - They are popularly known as the “fathers of the Internet”

Vint Cerf



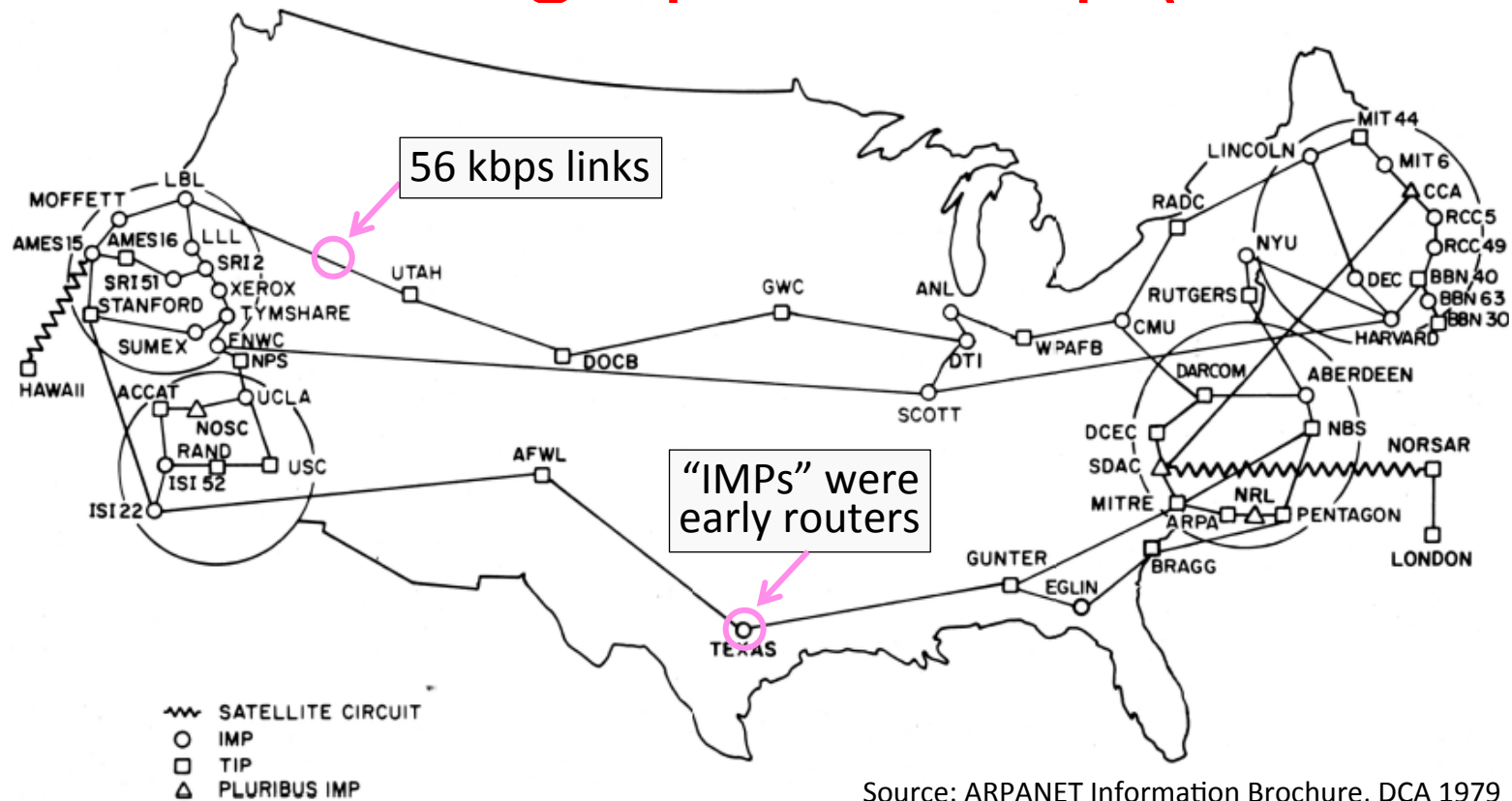
© 2009 IEEE

Bob Kahn



© 2009 IEEE

# ARPANET Geographical Map (Dec. 1978)



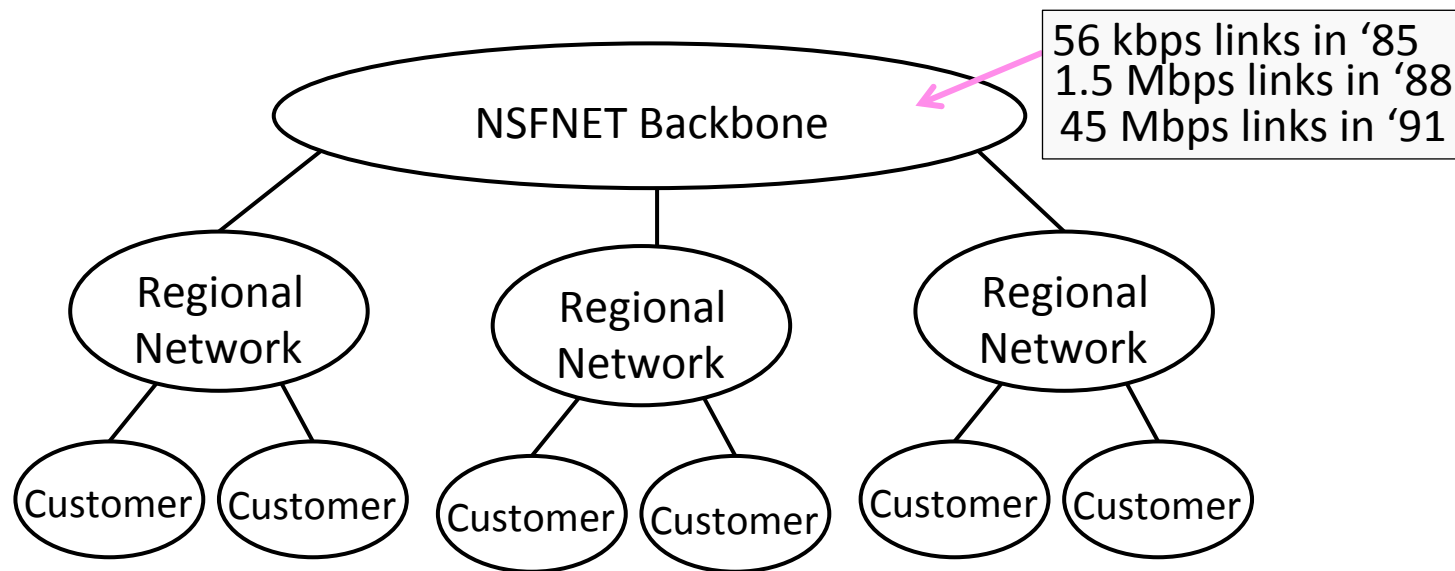
Source: ARPANET Information Brochure, DCA 1979

# Growing Up – NSFNET

- NSFNET '85 supports educational networks
  - Initially connected supercomputer sites, but soon became the backbone for all networks
- Classic Internet protocols we use emerged
  - TCP/IP (transport), DNS (naming), Berkeley sockets (API) in '83, BGP (routing) in '93
- Much growth from PCs and Ethernet LANs
  - Campuses, businesses, then homes
  - 1 million hosts by 1993 ...

# Early Internet Architecture

- Hierarchical, with NSFNET as the backbone

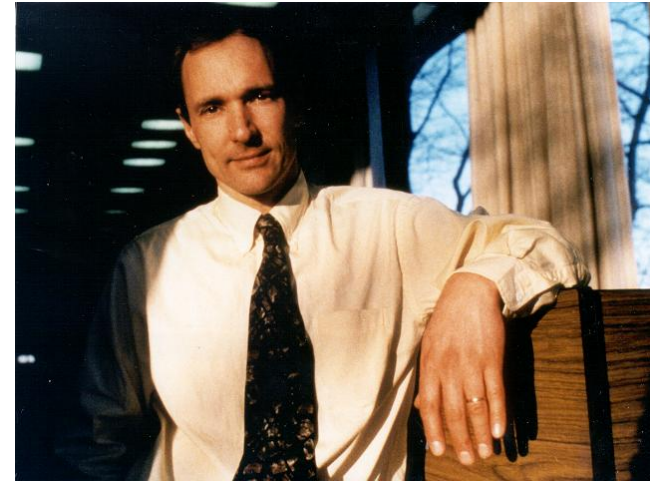




# Modern Internet – Birth of the Web

- After '95, connectivity is provided by large ISPs who are competitors
  - They connect at Internet eXchange Point (IXP) facilities
  - Later, large content providers connect
- Web bursts on the scene in '93
  - Growth leads to CDNs, ICANN in '98
  - Most bits are video (soon wireless)
  - Content is driving the Internet

Tim Berners-Lee



© 2009 IEEE

# Modern Internet Architecture

- Complex business arrangements affect connectivity
  - Still decentralized, other than registering identifiers

