An Internet Worthy of Society's Trust

Jennifer Rexford

Princeton University

How Is It Possible?

Shawn Fanning,
Northeastern freshman
Napster

Tim Berners-Lee
CERN Researcher
World Wide Web

Meg Whitman
E-Bay

So, I Went to Wikipedia…

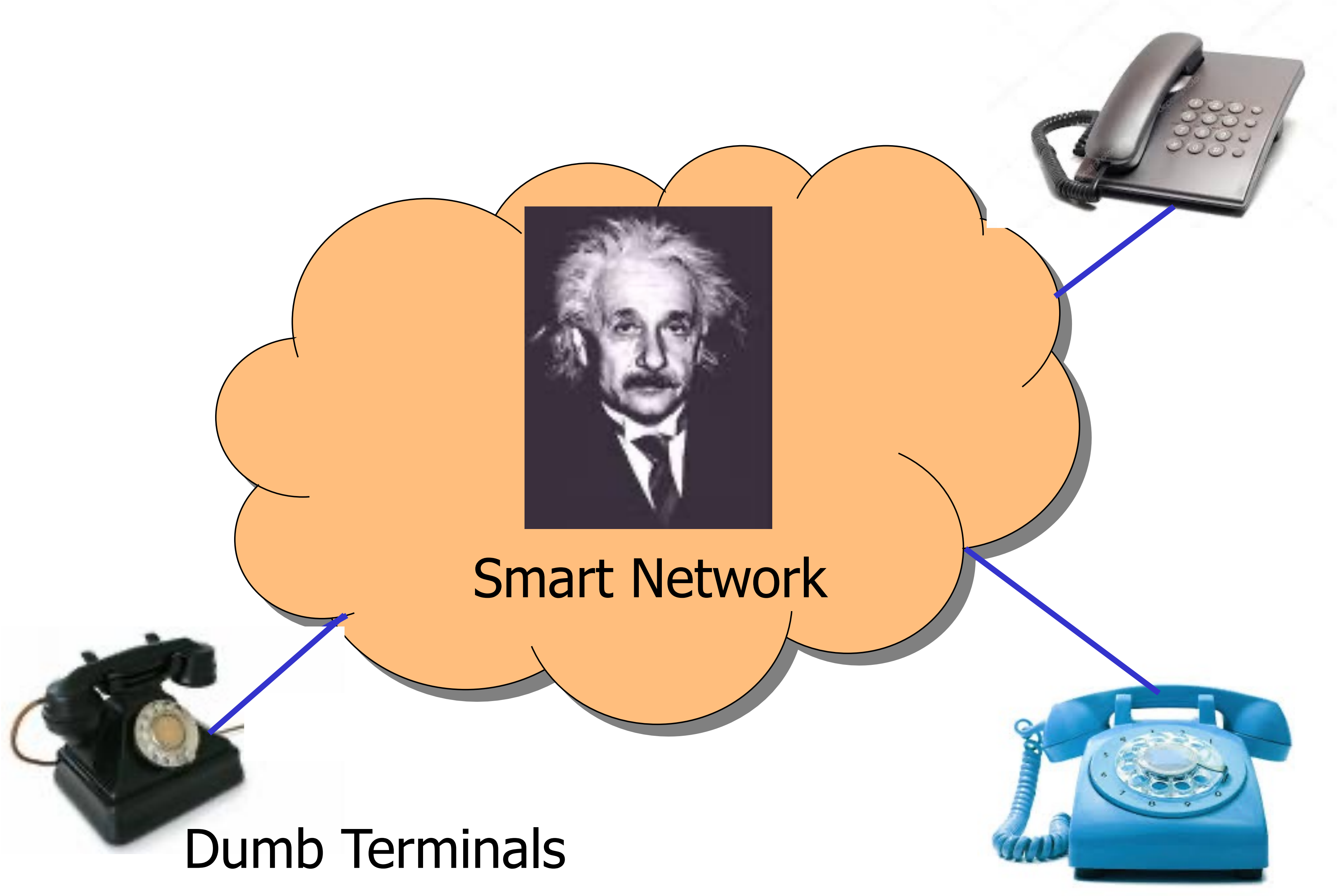
The **Internet** is the worldwide, publicly accessible network of interconnected computer networks that transmit data by packet switching using the standard Internet Protocol (IP). It is a "network of networks" that consists of millions of smaller domestic, academic, business, and government networks, which together carry various information and services, such as electronic mail, online chat, file transfer, and the interlinked Web pages and other documents of the World Wide Web.

http://en.wikipedia.org/wiki/Internet

Key Ideas Underlying the Internet

Idea #1: The rise of the stupid network

Telephone Network



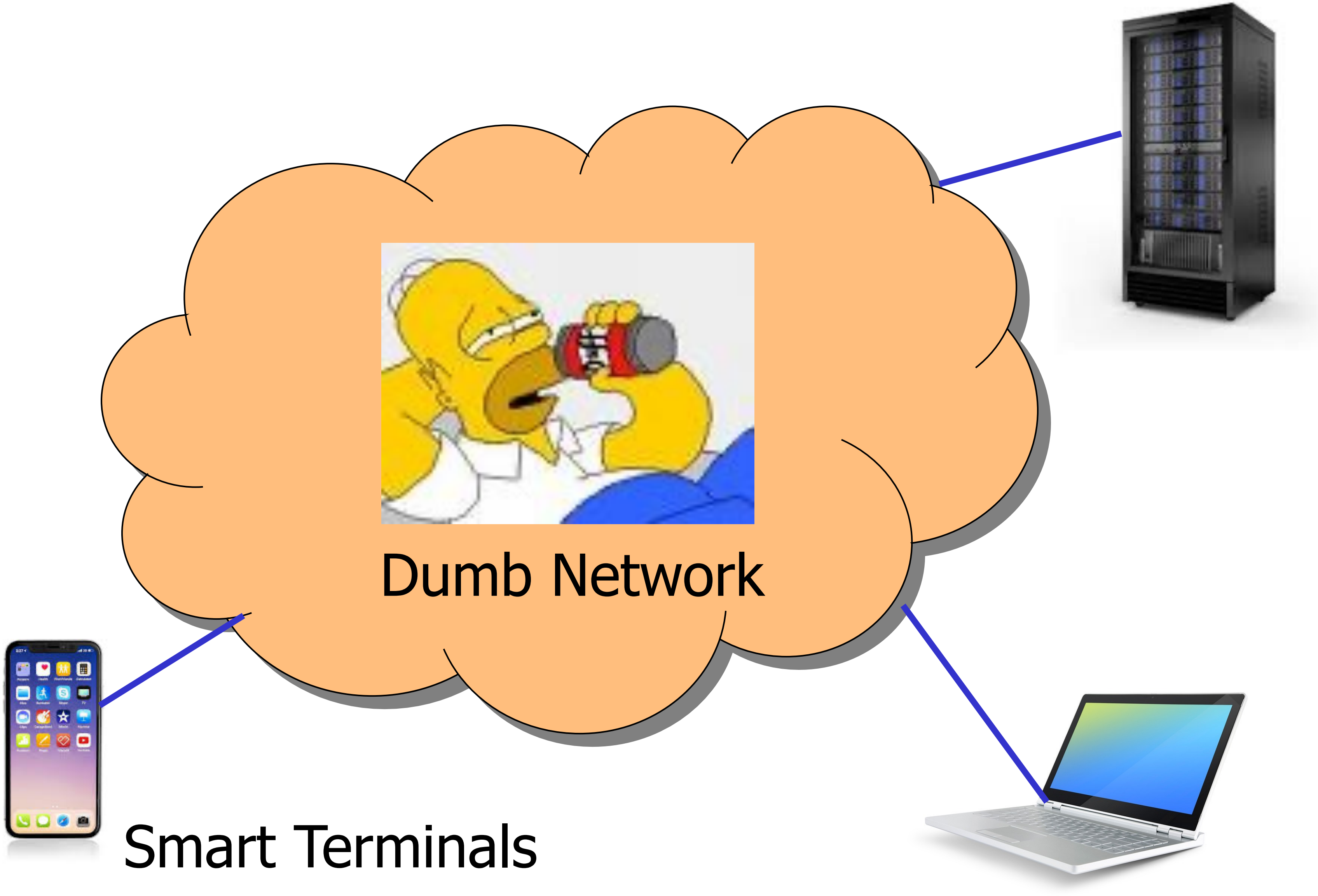
Smart Network

Dumb Terminals

Telephone Network

- Dumb phones
  - Dial a number
  - Speak and listen
- Smart switches
  - Set up and tear down a circuit
  - Forward audio along the path
- Limited services
  - Audio
  - Later, fax, caller-id, …
- A monopoly for a long time

Internet



Dumb Network

Smart Terminals

Power at the Edge

End-to-End Principle

Whenever possible, communications protocol operations should be defined to occur at the end-points of a communications system.

Programmability

With programmable end hosts, new network services can be added at any time, by anyone.

And then end hosts became powerful and ubiquitous….

Idea #2: Going Postal

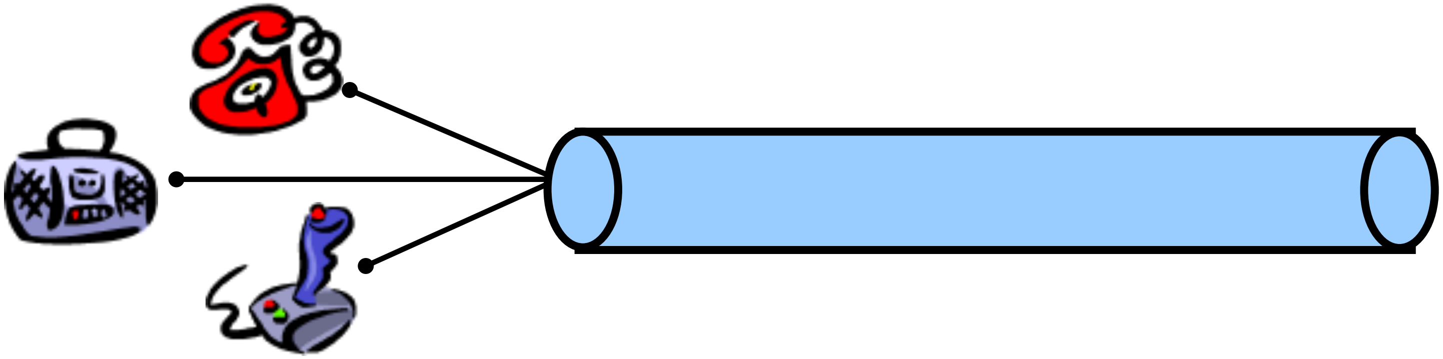
Internet Protocol (IP) Packet Switching

- Much like the postal system
  - Divide information into letters
  - Stick them in envelopes
  - Deliver them independently
  - And sometimes they get there

- What's in an IP packet?
  - The data you want to send
  - A header with the "from" and "to" addresses

Why Packets?

- Data traffic is bursty
  - Logging in to remote machines
  - Exchanging e-mail messages
- Don’t waste bandwidth
  - No traffic exchanged during idle periods
- Better to allow multiplexing
  - Different transfers share access to same links



Why Packets?

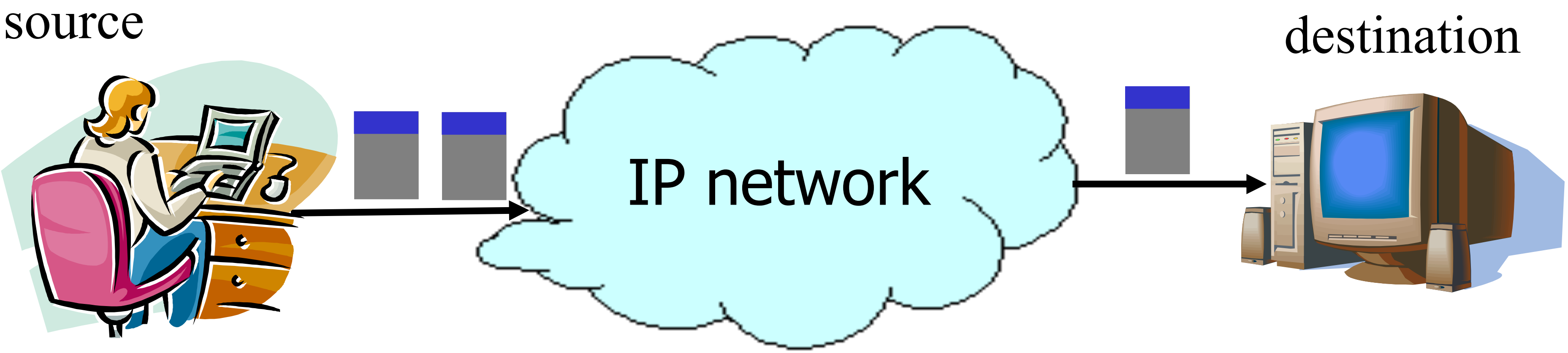
- Packets can be delivered by most anything
  - Serial link, fiber optic link, coaxial cable, wireless
- Even birds
  - RFC 1149: IP Datagrams over Avian Carriers

IP over Avian Carriers was actually implemented, sending 9 packets over a distance of approximately 5km (3 miles), each carried by an individual pigeon, and they received 4 responses, with a packet loss ratio of 55%, and a response time ranging from 3000 seconds to over 6000 seconds.

Idea #3: Never having to say you’re sorry

Best-Effort Packet-Delivery Service

- Best-effort delivery
  - Packets may be lost
  - Packets may be corrupted
  - Packets may be delivered out of order



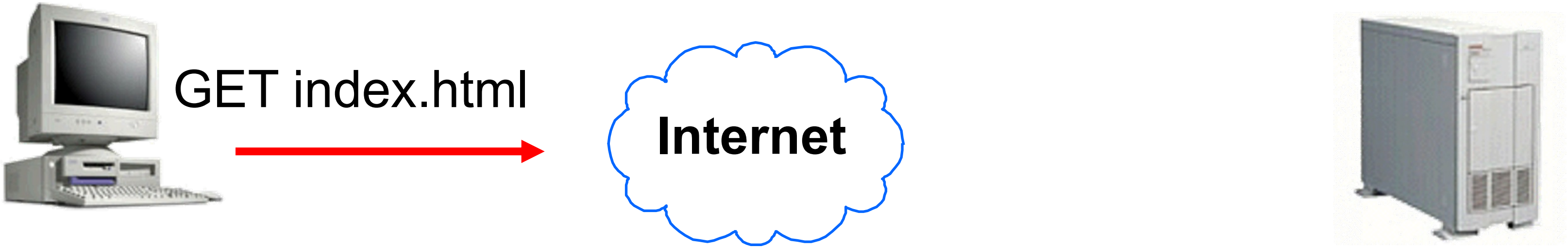
IP Service Model: Why Best-Effort?

- I never promised you a rose garden
  - No error detection and correction
  - Don’t remember from one packet to next
  - Don’t reserve bandwidth and memory
- Easier to survive failures
  - Transient disruptions are okay during failover

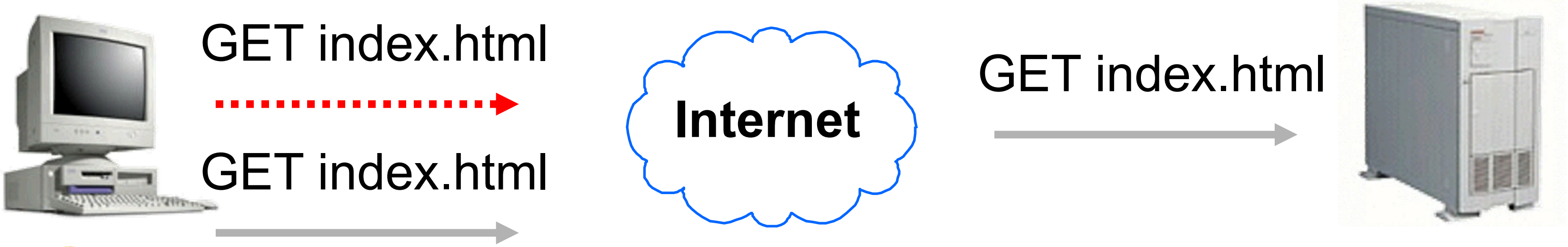
- … but, applications *do* want efficient, accurate transfer of data in order, in a timely fashion
- Let the end host take care of that!

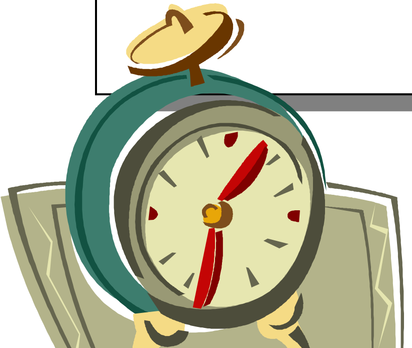
What if Packets are Lost or Delayed?

Problem: Lost or Delayed Data



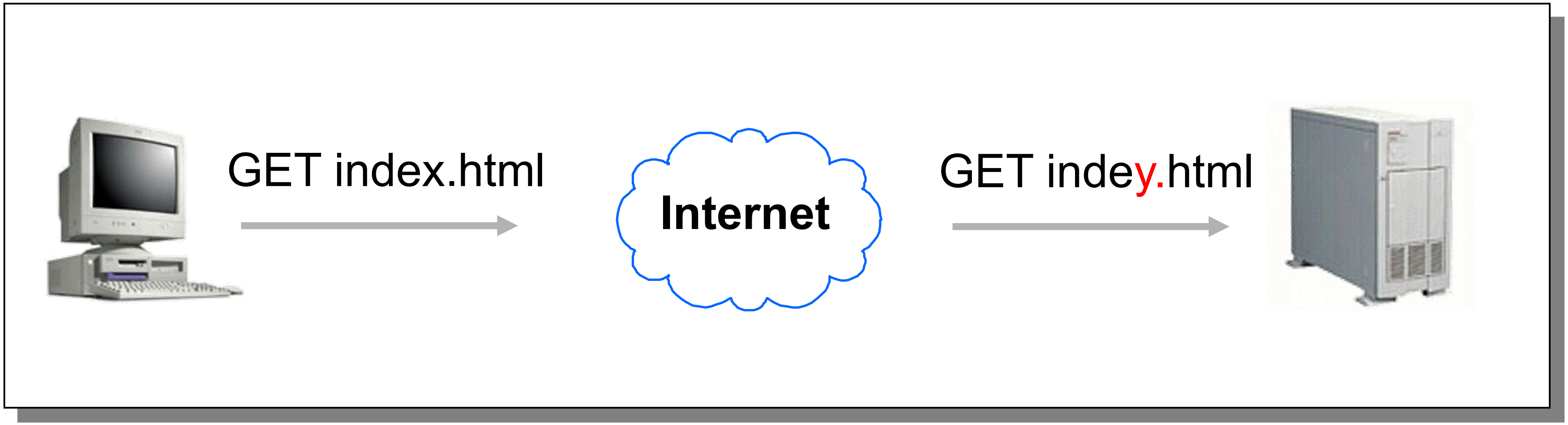
Solution: Timeout and Retransmit





Waiting for an acknowledgment…

What if Packets are Corrupted?



- Sender computes a checksum
  - Sender sums up all of the bytes
  - And sends the sum to the receive

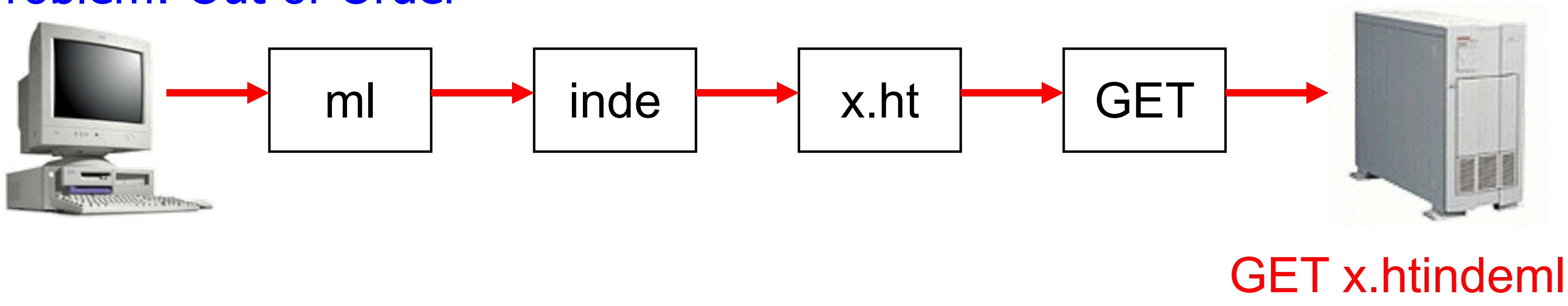
$$\begin{array}{r} 134 \\ +\ 212 \\ \hline =\ 346 \end{array}$$

- Receiver checks the checksum
  - Received sums up all of the bytes
  - And compares against the checksum

$$\begin{array}{r} 134 \\ +\ 216 \\ \hline =\ 350 \end{array}$$

What if the Data is Out of Order?

Problem: Out of Order



ml

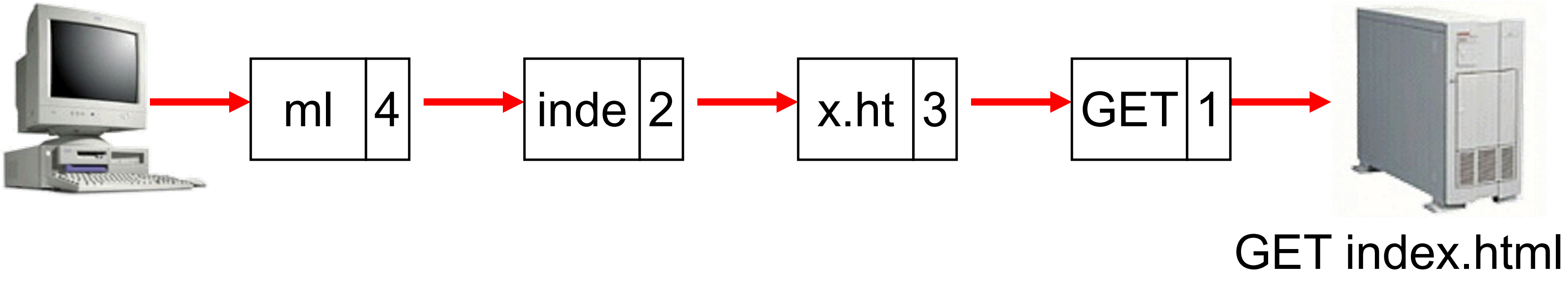
inde

x.ht

GET

GET x.htindeml

Solution: Add Sequence Numbers



ml 4

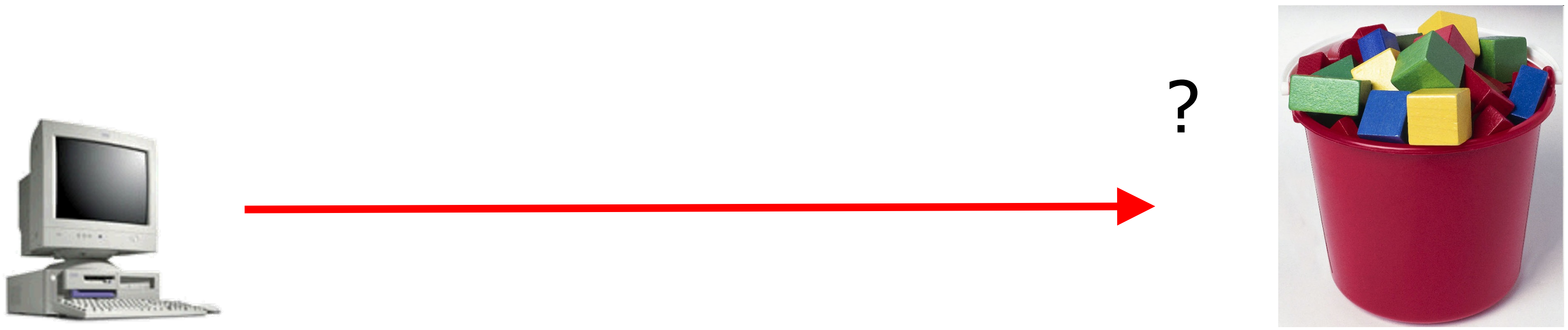
inde 2

x.ht 3

GET 1

GET index.html

What if the Receiver is Out of Space?



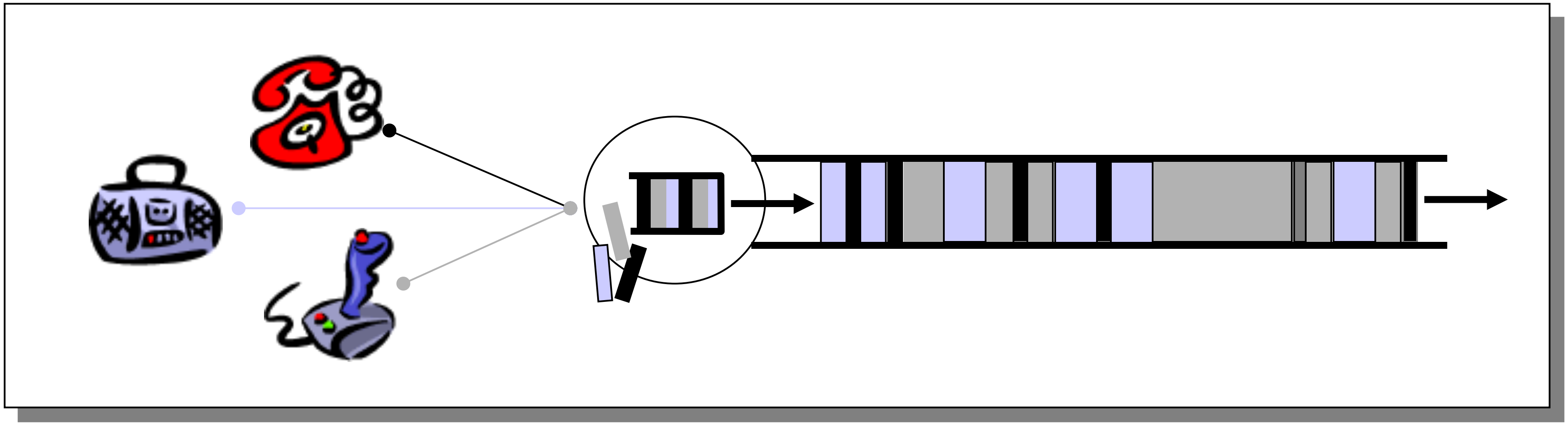
- Receiver maintains a *window size*
    - – Amount of data it can buffer
- Advertises window to the sender
    - – Amount sender can send without acknowledgment
- Ensures that sender doesn’t send too much
    - – While sending as much as possible

What if Too Many Hosts Send at Once?

- Some folks need to slow down…

Idea #4: Think globally, act locally

Congestion

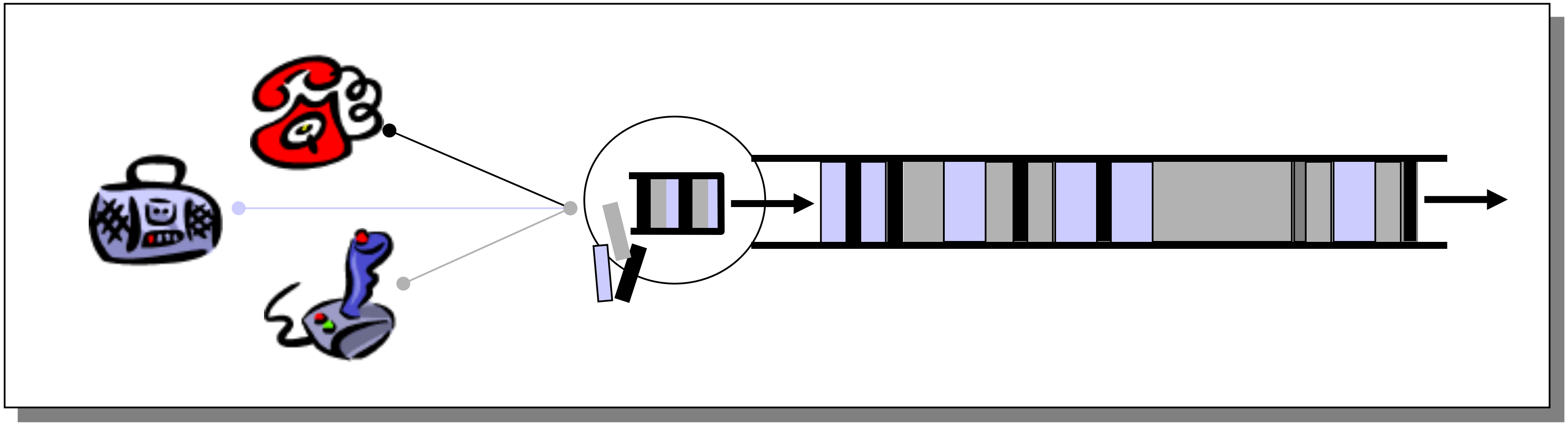


- Too many hosts sending packets at once
  - – Some packets have to wait in line
  - – Eventually the queue runs out of space
  - – And some packets gets dropped on the floor

Sharing the Limited Resource

- Reserve resources
  - Room for ten phone calls
  - Block the 11th call
- Sub-divide resources
  - Tell the 11 transfers to each use 1/11 of the bandwidth
  - How????
- Local adaptation
  - Each transfer slows down
  - Voluntarily, for greater good

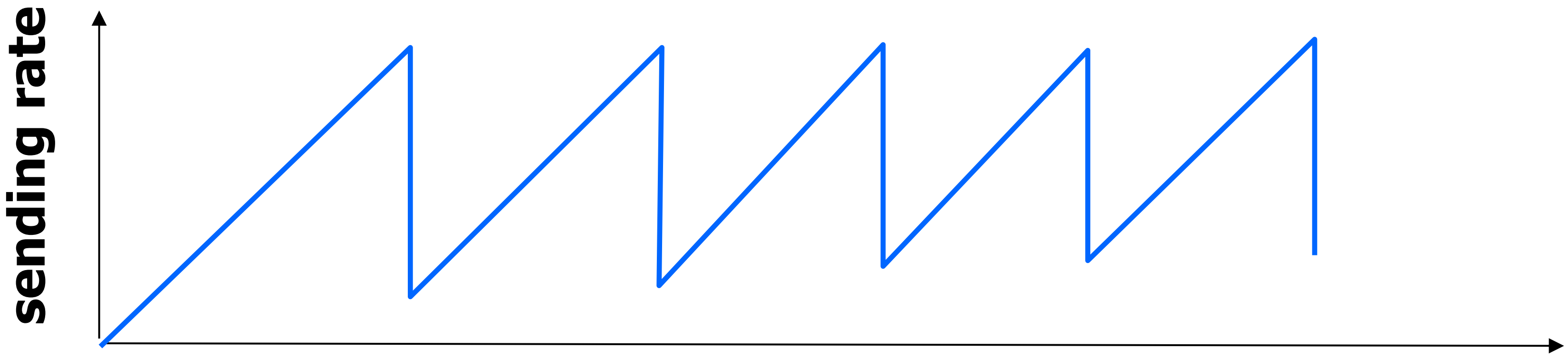
Congestion Control



- What if too many folks are sending data?
  - Senders agree to slow down their sending rates
  - … in response to their packets getting dropped
  - For the greater good

Congestion Control

- Detecting congestion
  - My packet was lost
- Reacting to congestion
  - I voluntarily reduce my sending rate (by 2X)
- Testing the waters
  - I gradually increase my sending rate (linearly)



Transmission Control Protocol (TCP)

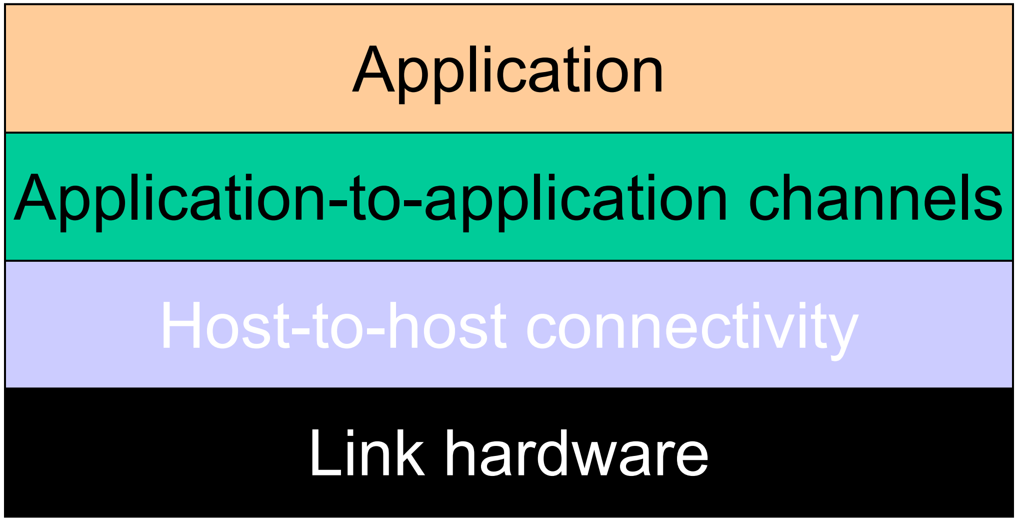
- Runs on the end host
  - Puts data into packets and sends them
- Congestion control
  - Speeds up and slows down
- Ordered reliable byte stream
  - Sender retransmits lost packets
  - Receiver discards corrupted packets
  - Receiver reorders out-of-order packets

Reliable service on an unreliable network

Key idea #5: Standing on the shoulders of giants

Layering: A Modular Approach

- Sub-divide the problem
  - Each layer relies on services from layer below
  - Each layer exports services to layer above
- Interface between layers defines interaction
  - Hides implementation details
  - Layers can change without disturbing other layers



Application

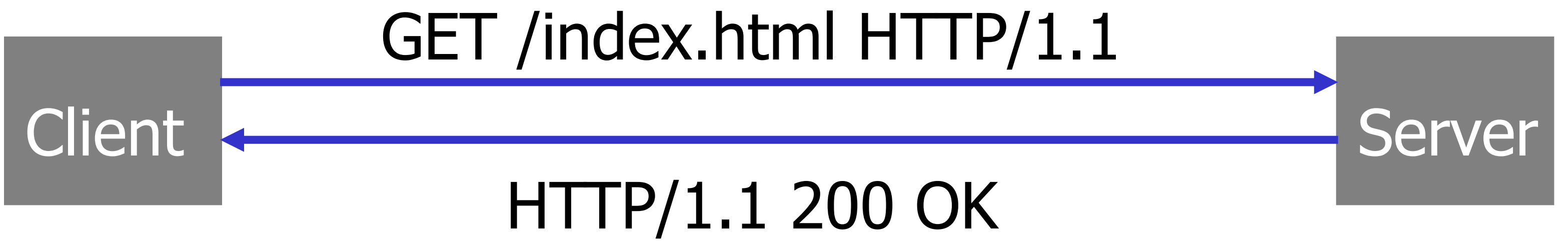
Application-to-application channels

Host-to-host connectivity

Link hardware

Application-Layer Protocols

- Messages exchanged between applications
  - Syntax and semantics of the messages between hosts
  - Tailored to the specific application (e.g., Web, e-mail)
  - Messages transferred over transport connection (e.g., TCP)
- Popular application-layer protocols
  - Telnet, FTP, SMTP, NNTP, HTTP, …



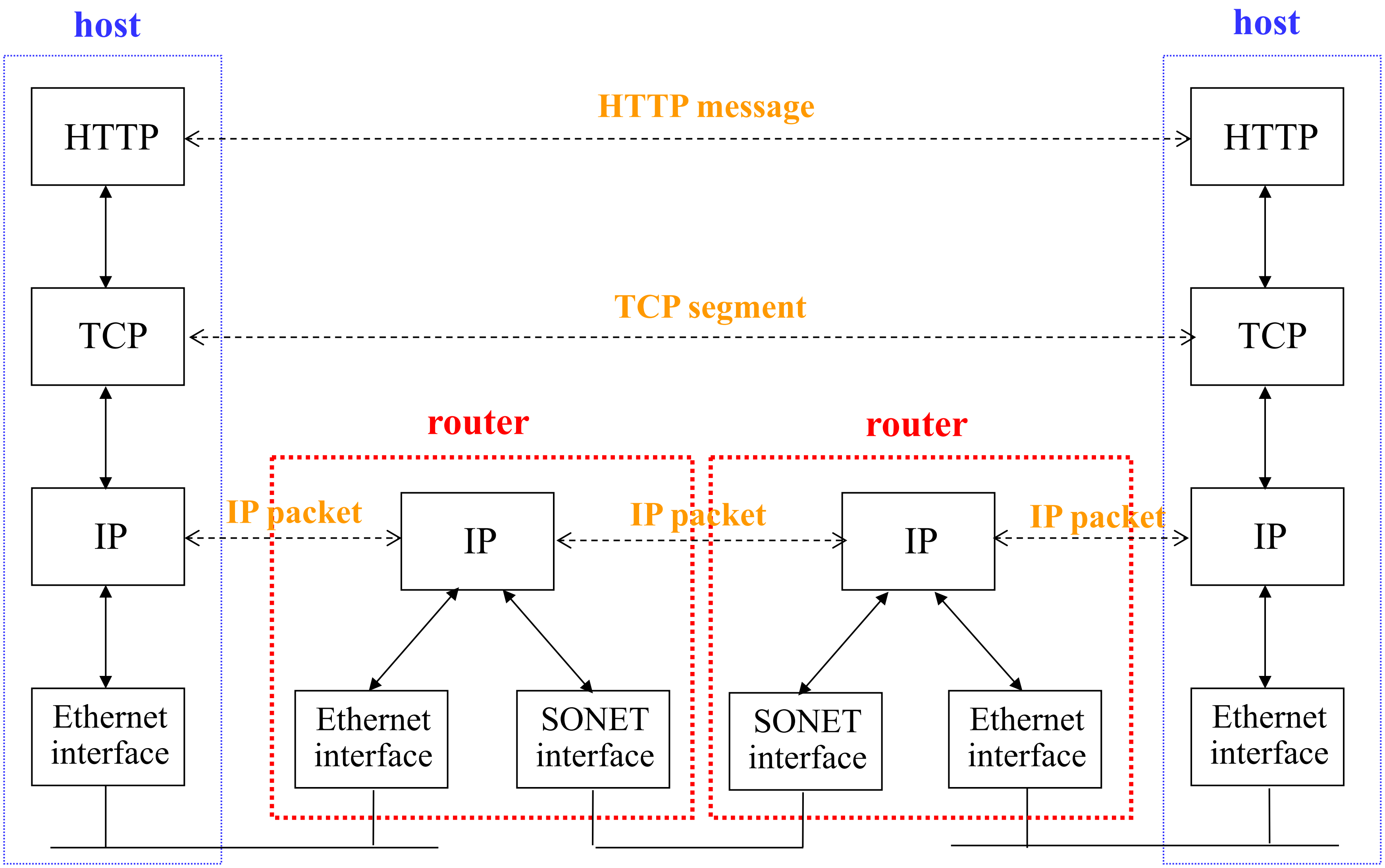
GET /index.html HTTP/1.1

Client

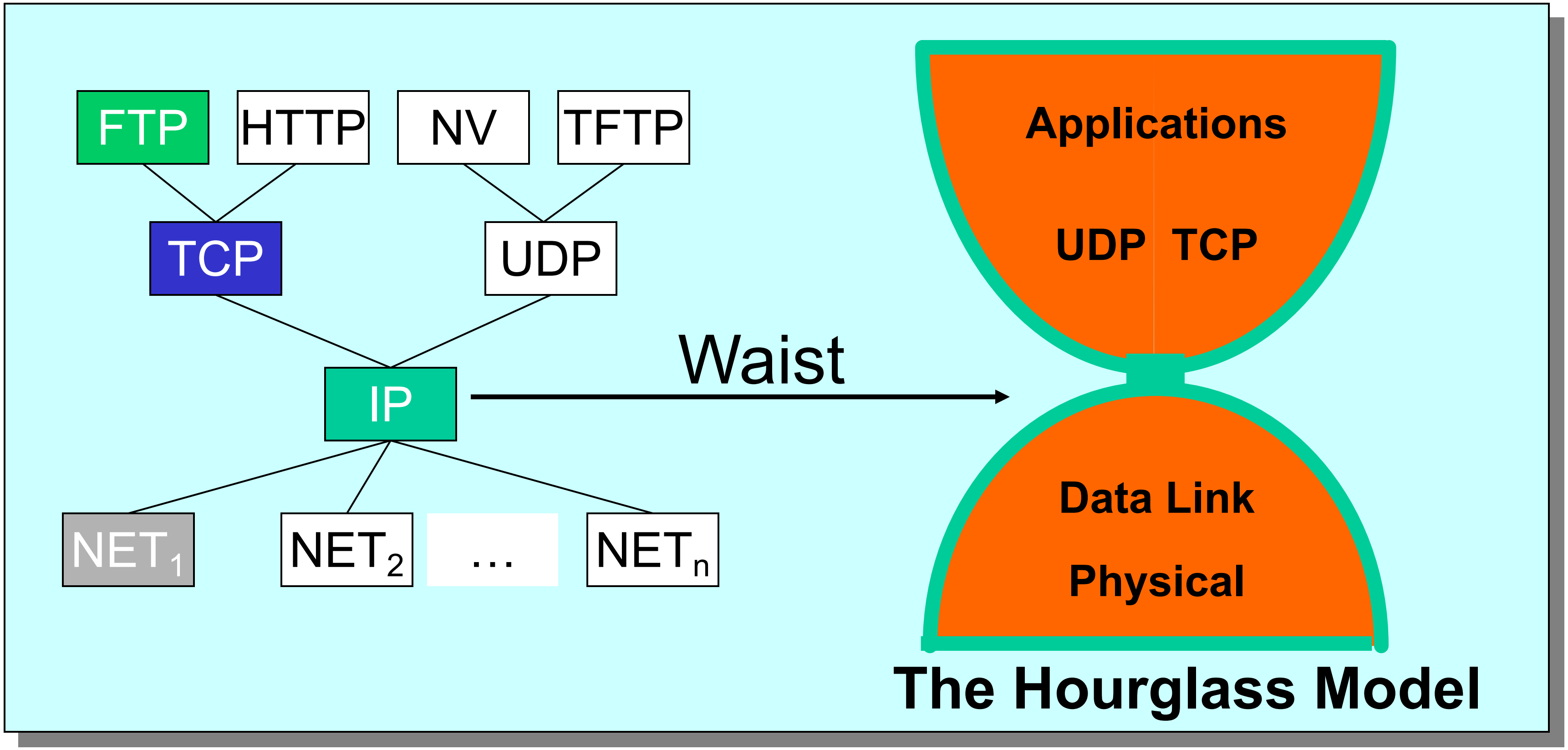
Server

HTTP/1.1 200 OK

Layering in the Internet



The Narrow Waist of IP



The waist facilitates interoperability

Idea #6: A rose by any other name

Separating Naming and Addressing

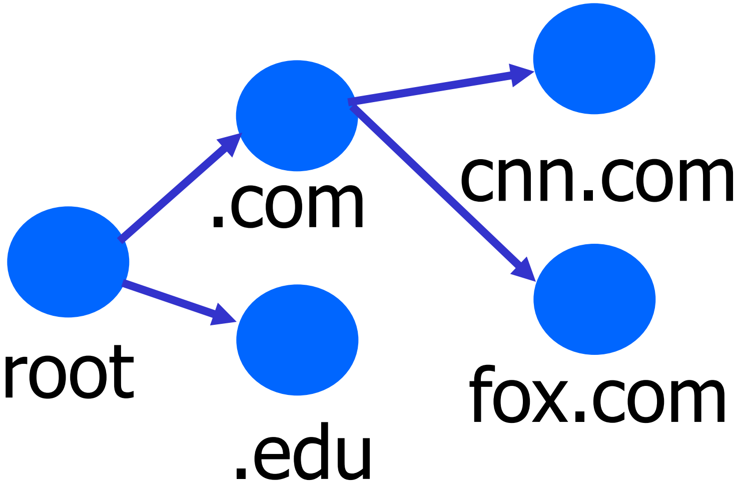
- Host names
  - Mnemonic name appreciated by humans
  - Variable length, alpha-numeric characters
  - Provide little (if any) information about location
  - Examples: www.cnn.com and ftp.eurocom.fr
- IP addresses
  - Numerical address appreciated by routers
  - Fixed length, binary number
  - Hierarchical, related to host location
  - Examples: 64.236.16.20 and 193.30.227.161

Separating Naming and Addressing

- Names are easier to remember
  - www.cnn.com vs. 64.236.16.20
- Addresses can change underneath
  - Move www.cnn.com to 64.236.16.20
- Name could map to multiple IP addresses
  - www.cnn.com to multiple replicas of the Web site
- Map to different addresses in different places
  - Address of a nearby copy of the Web site
  - E.g., to reduce latency, or return different content
- Multiple names for the same address
  - E.g., aliases like ee.mit.edu and cs.mit.edu

Domain Name System (DNS) Hierarchy

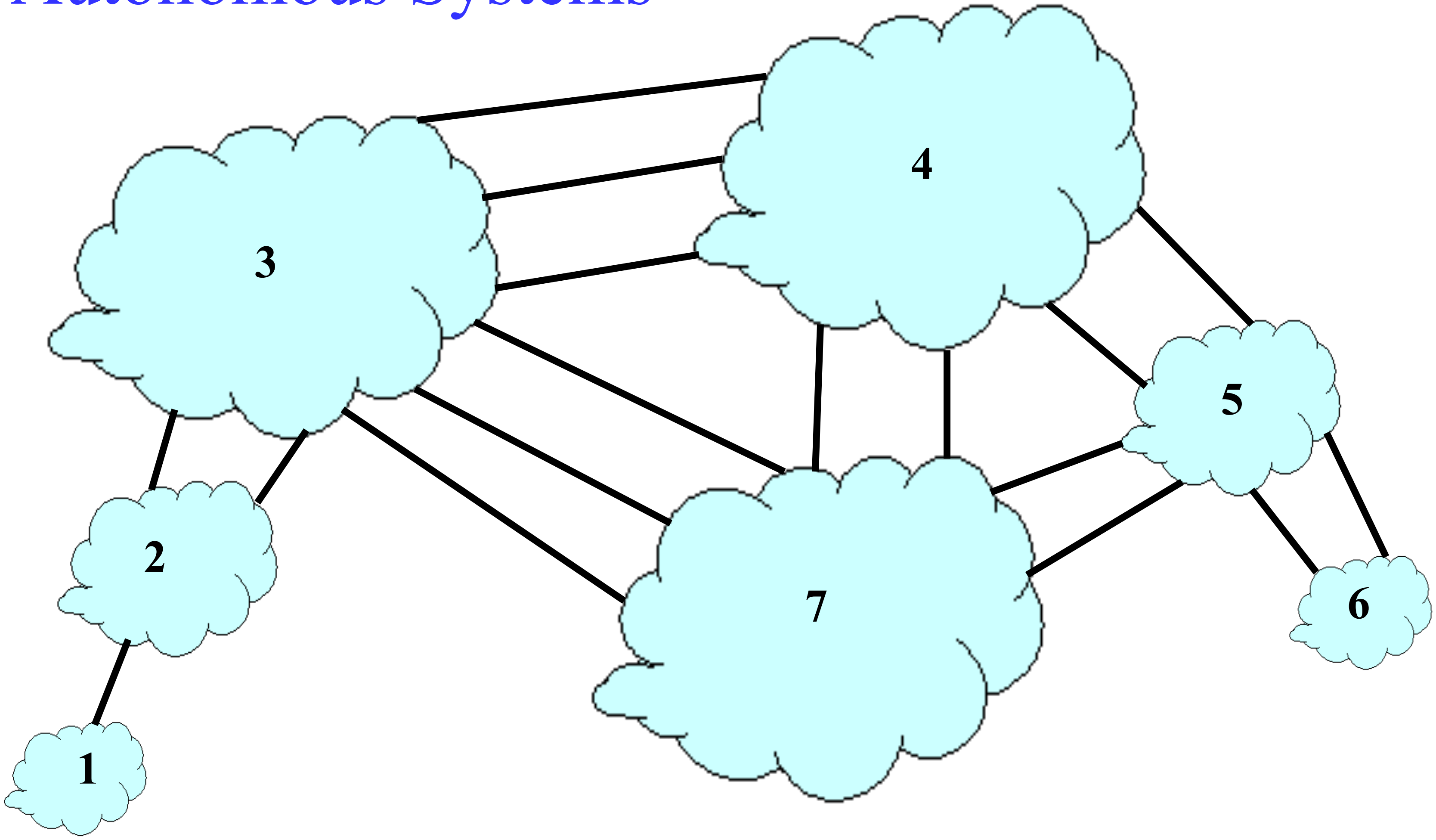
- Distributed “phone book”
  - Multiple queries to translate name to address
- Small number of “root servers”
  - Tell you where to look up “.com” names
- Larger number of “top-level domains”
  - Tell you where to look up “cnn.com” names



Idea #7: You scratch my back…

Network of Networks

Autonomous Systems

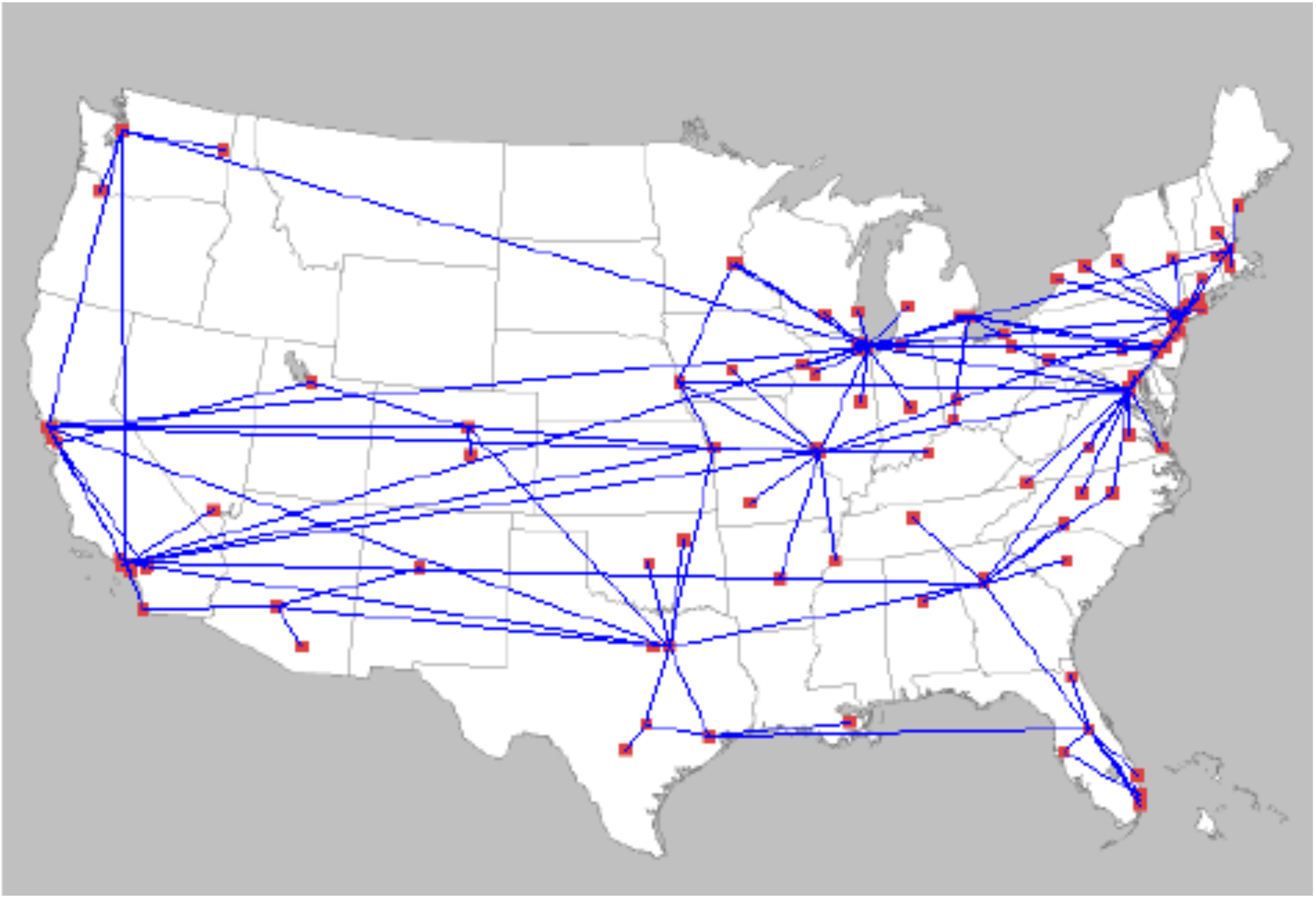


Autonomous Systems (Ases)

Currently around 77,543 ASes.

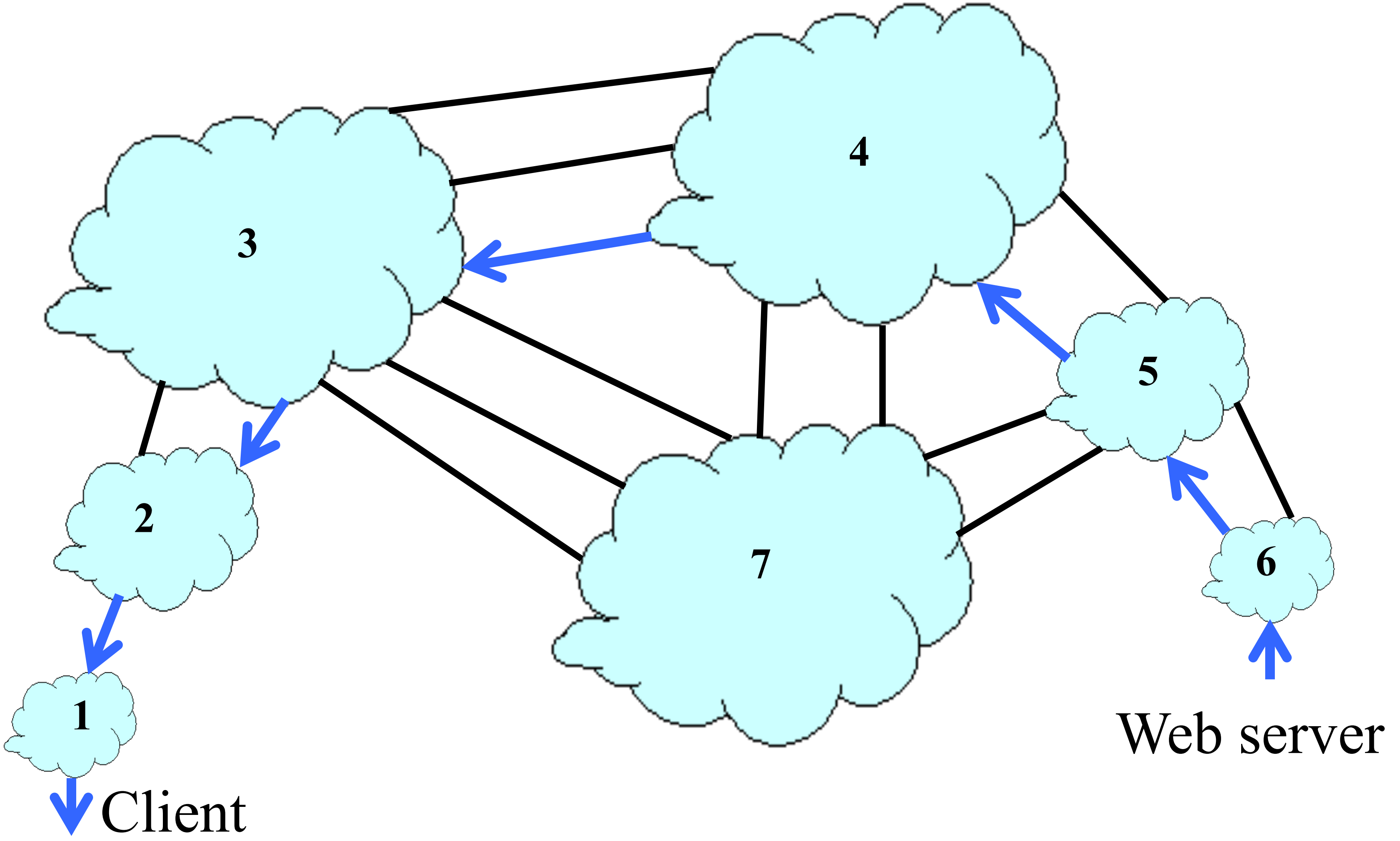
- Level 3: 1
- MIT: 3
- Princeton: 88
- AT&T: 7018, 6341, 5074, …
- Verizon: 701, 702, 284, 12199, …
- …

Inside an AS: AT&T Backbone Network



Cooperation and Competition

Traffic flows through many ASes



Problems With the Internet: Cheaters do win

No Strict Notions of Identity

"On the Internet, nobody knows you're a dog."

- Leads to
  - – Spam
  - – Spoofing
  - – Denial-of-service

Nobody in Charge

- Traffic traverses many Autonomous Systems
  - Who’s fault is it when things go wrong?
  - How do you upgrade functionality?
- Implicit trust in the end host
  - What if some hosts violate congestion control?
- Anyone can add any application
  - Whether or not it is moral, good, etc.
- Nobody knows how big the Internet is
  - No global registry of the topology
- Spans many countries
  - So no government can be in charge

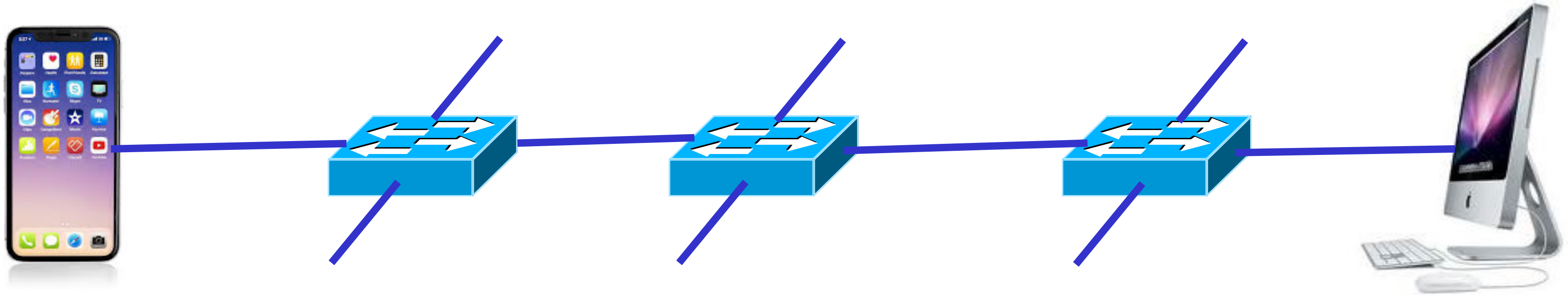
The Internet of the Future

- Can we fix what ails the Internet
  - Security
  - Performance
  - Upgradability
  - Managability
  - <your favorite gripe here>

- Without throwing out the baby with bathwater
  - Ease of adding new hosts
  - Ease of adding new services
  - Ease of adding new link technologies
- An open technical and policy question…

Toward a Better (Smarter) Internet

- Programmability is key
  - Enables innovation over time
  - Enables different solutions in different places
  - Determines *who* gets to innovate
- Inside of the Internet
  - Historically *not* programmable
  - Drives network speed, performance, security, etc.

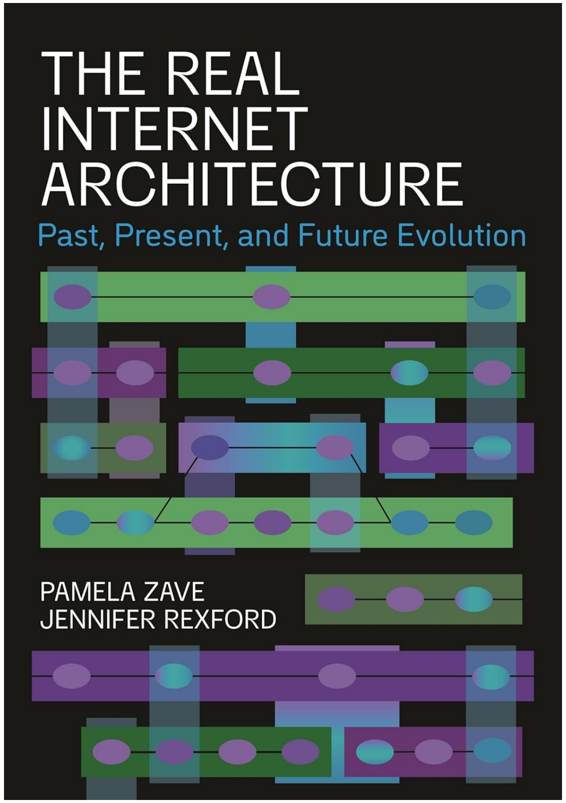


Programmable Networks

- Programmable network hardware
  - – Flexibility at the level of individual packets
  - – … though with limitations for high speed
- My research
  - – Programming languages for high-speed networks
  - – Working within severe resource constraints
  - – Improving Internet performance, security, etc.
- Highly collaborative
  - – Interdisciplinary research
  - – Industry engagement

Internet Evolution

- The Internet continues to evolve
- Describing accurately how it has evolved
- … and how it can evolve in the future


THE REAL INTERNET ARCHITECTURE
Past, Present, and Future Evolution
PAMELA ZAVE
JENNIFER REXFORD

Thanks!