Network Architecture
COS 461: Computer Networks

Nick Feamster

Spring 2015

Lectures: MW 10-10:50 am in CS 104

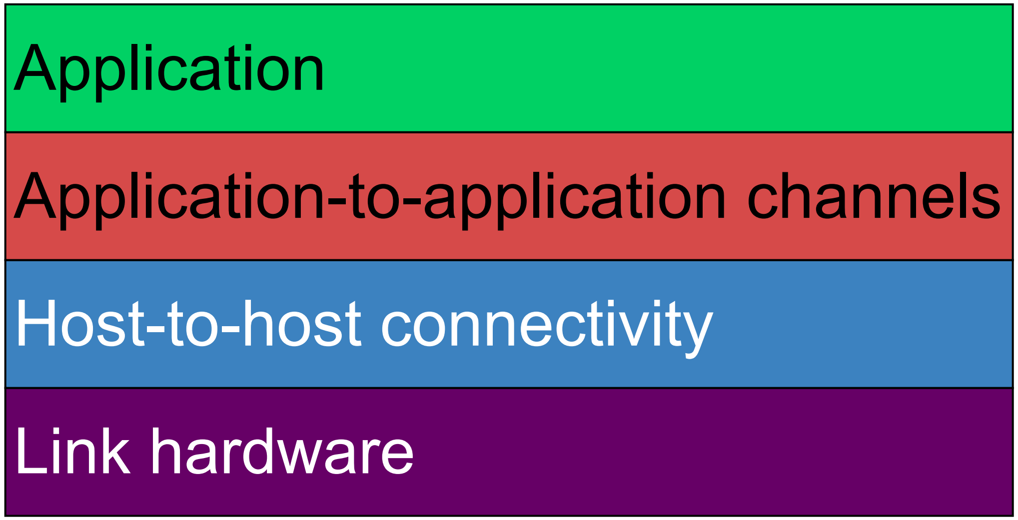
Acknowledgments: Lecture slides are from Computer networks course thought by Nick Feamster at Princeton University. When slides are obtained from other sources, a a reference will be noted on the bottom of that slide. A full list of references is provided on the last slide.

Key Concepts in Networking

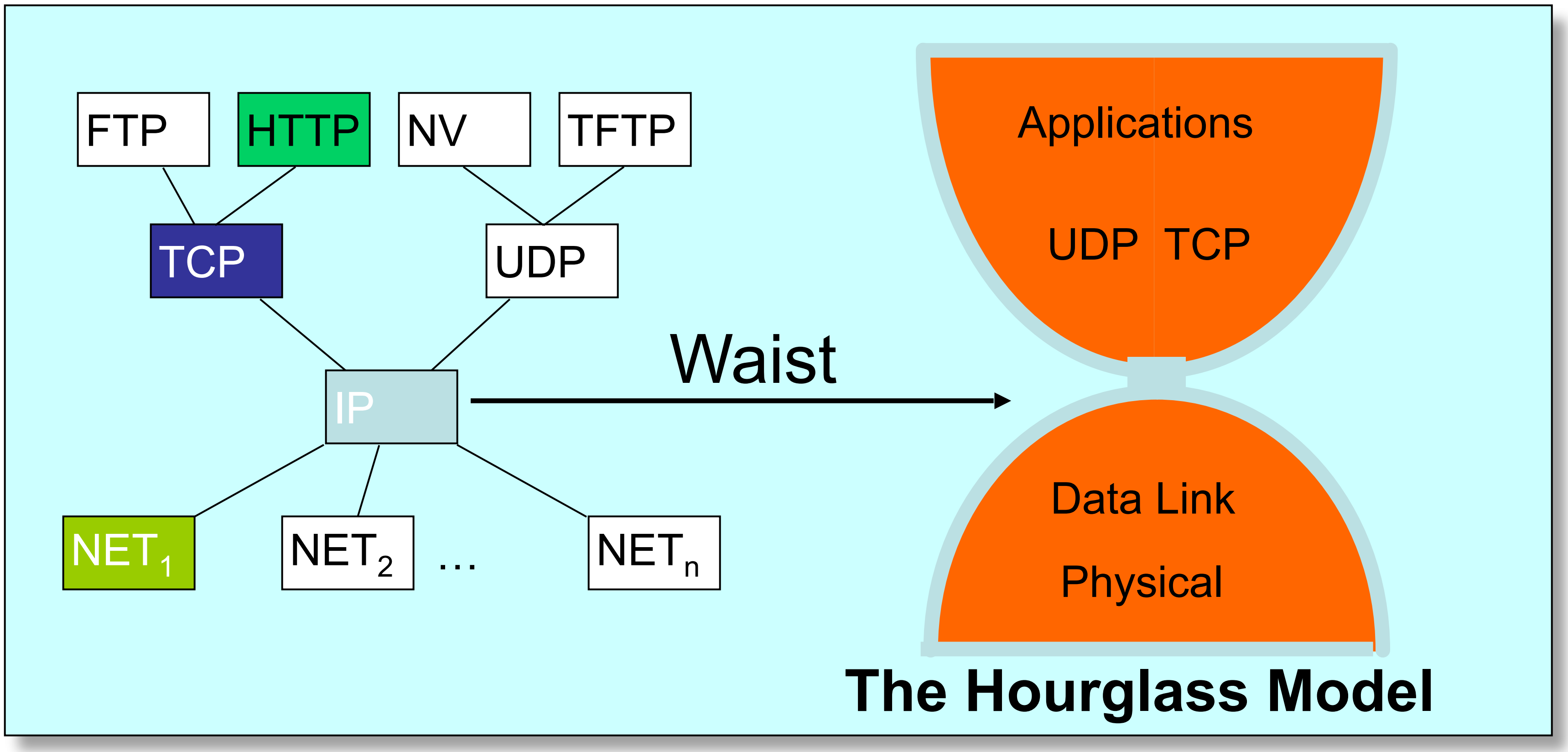
- Naming
  - What to call computers, services, protocols, …
- Layering
  - Abstraction is the key to managing complexity
- Protocols
  - Speaking the same language
  - Syntax and semantics
- Resource allocation
  - Dividing scare resources among competing parties
  - Memory, link bandwidth, wireless spectrum, paths

Abstraction through Protocol Layering

- Modularity
  - Each layer relies on services from layer below
  - Each layer exports services to layer above
- Interfaces
  - Hides implementation details
  - Layers can change without disturbing other layers



The Internet Protocol Suite



The "narrow waist" facilitates interoperability

Example: HyperText Transfer Protocol

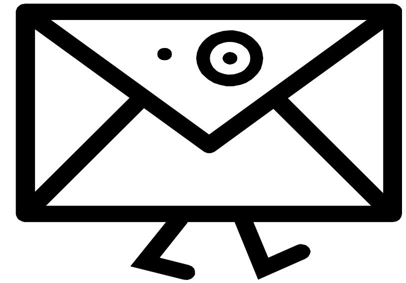
```
GET /courses/archive/spr13/cos461/ HTTP/1.1
Host: www.cs.princeton.edu
User-Agent: Mozilla/4.03
CRLF
```

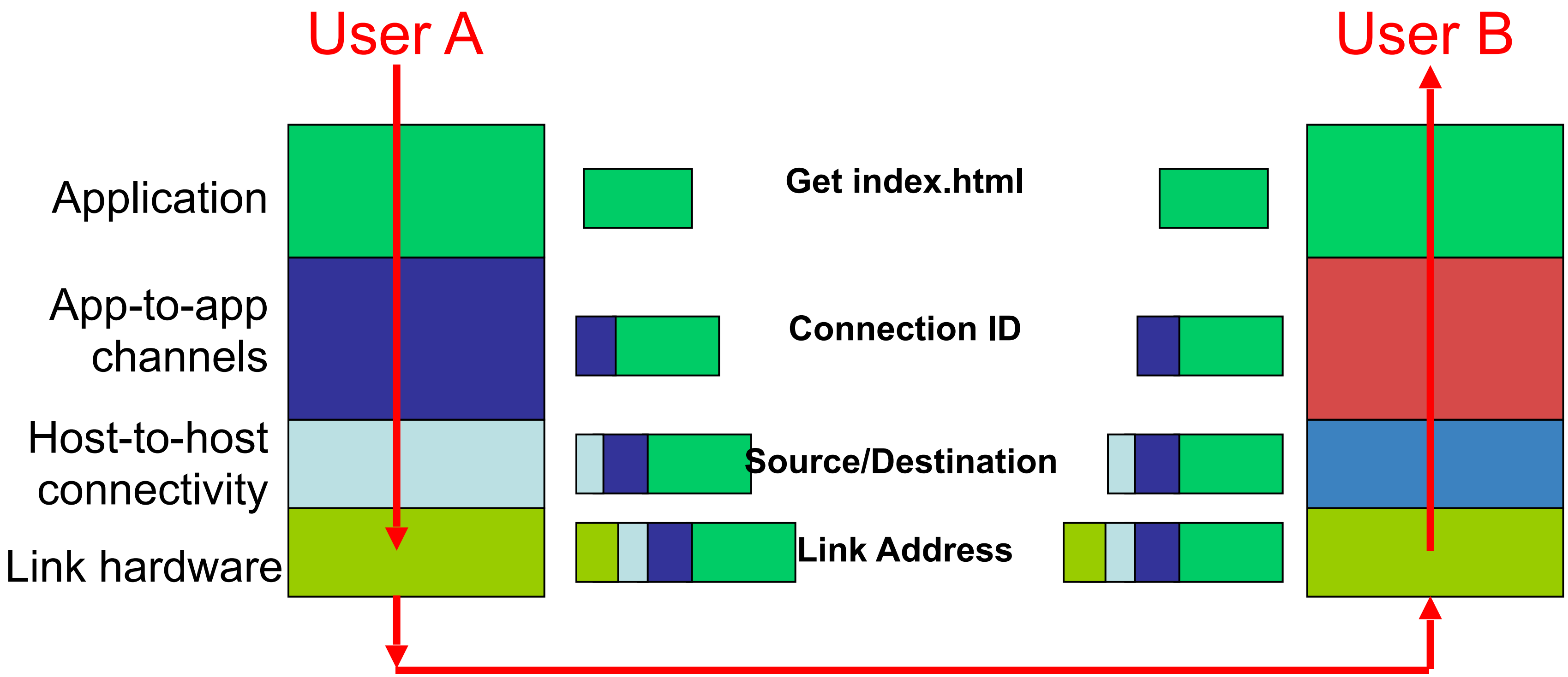
Request

Response

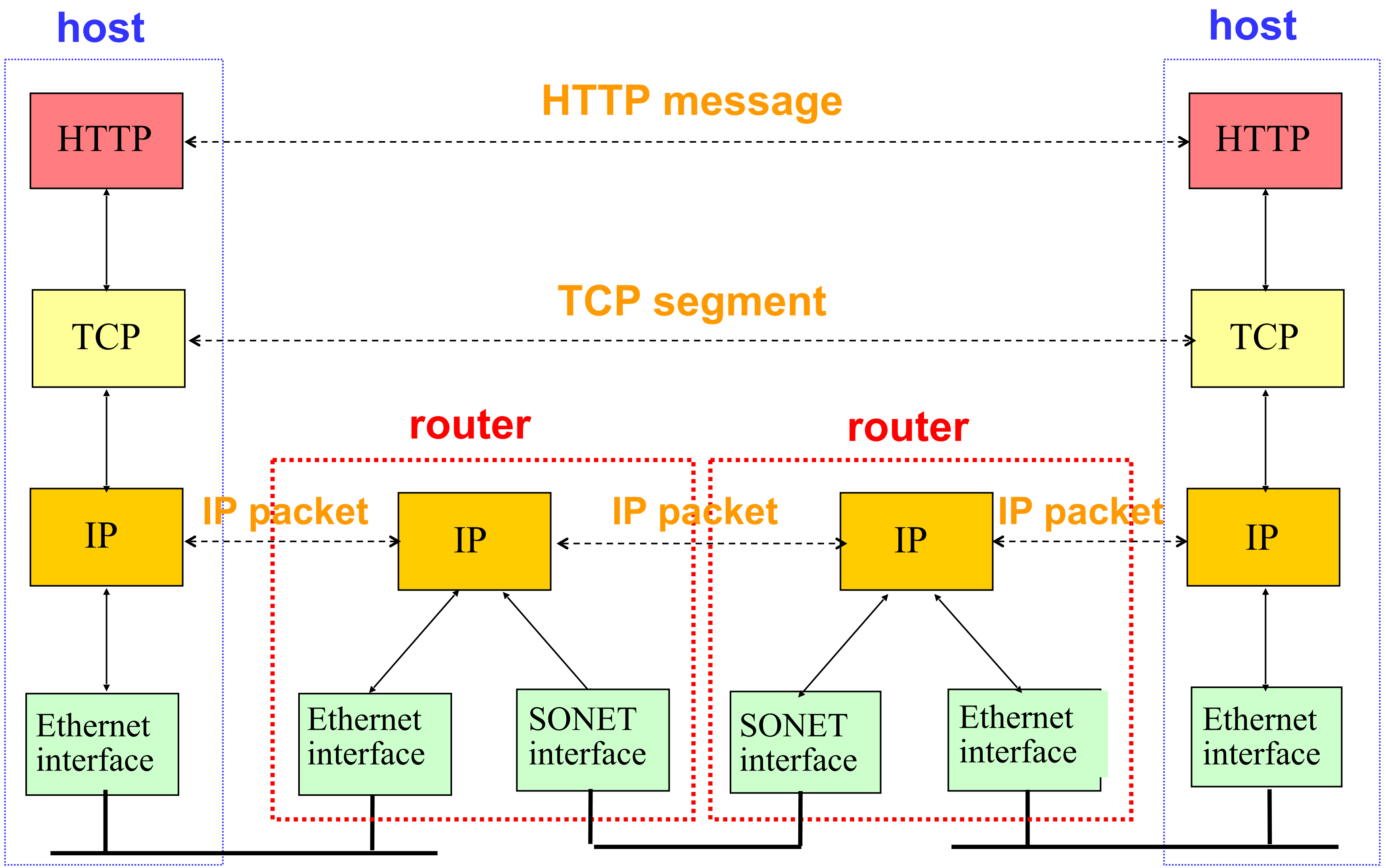
```
HTTP/1.1 200 OK
Date: Mon, 4 Feb 2013 11:09:03 GMT
Server: Netscape-Enterprise/3.5.1
Last-Modified: Mon, 2 Feb  2013 19:12:23 GMT
Content-Length: 21
CRLF
Site under construction
```

Layer Encapsulation in HTTP



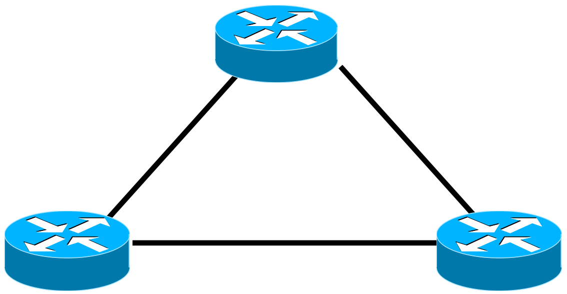


End Hosts vs. Routers



Split into Data vs. Control Plane

- Data plane: packets
  - Handle individual packets as they arrive
  - Forward, drop, or buffer
  - Mark, shape, schedule, …
- Control plane: events
  - Track changes in network topology
  - Compute paths through the network
  - Reserve resources along a path



Motivated by need for high-speed packet forwarding

Original Design Philosophy

Fundamental Goal

- “technique for **multiplexed utilization** of **existing interconnected networks**”

- **Multiplexing** (sharing)
  - Shared use of a single communications channel
- **Existing networks** (interconnection)

Packet Switching

Fundamental Goal: Sharing

- No connection setup
- Forwarding based on destination address in packet
- Efficient sharing of resources

Tradeoff: **Resource management potentially more difficult.**

Type of Packet Switching: Datagrams

- Information for forwarding traffic is contained in destination address of packet
- No state established ahead of time (helps fate sharing)
- Basic building block
- Minimal assumption about network service

Alternatives

- **Circuit Switching:** Signaling protocol sets up entire path out-of-band. (cf. the phone network)
- **Virtual Circuits:** Hybrid approach. Packets carry "tags" to indicate path, forwarding over IP
- **Source routing:** Complete route is contained in each data packet

An Age-Old Debate

Circuit Switching

- Resource control, accounting, ability to “pin” paths, etc.

Packet Switching

- Sharing of resources, soft state (good resilience properties), etc.

It is held that packet switching was one of the Internet’s greatest design choices.

Of course, there are constant attempts to shoehorn the best aspects of circuits into packet switching.

Examples: Capabilities (Lecture 21), MPLS (Lecture 15), ATM, IntServ QoS, etc.

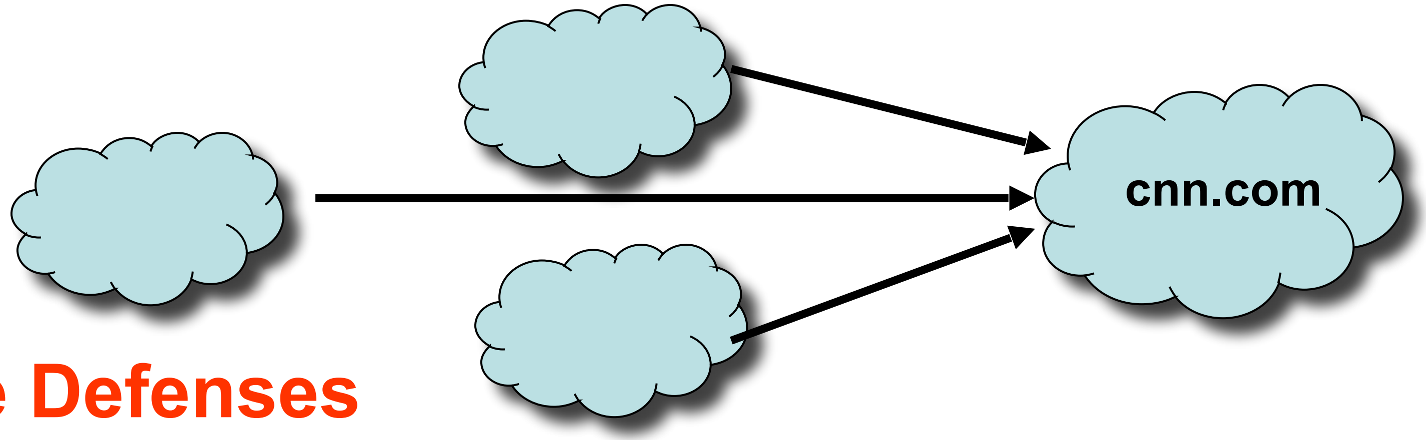
Stopping Unwanted Traffic is Hard

February 2000

March 2006

Stopping Unwanted Traffic

- *Datagram networks:* easy for anyone to send traffic to anyone else…even if they don’t want it!



Possible Defenses

- *Monitoring + Filtering:* Detect DoS attack and install filters to drop traffic.
- *Capabilities:* Only accept traffic that carries a “capability”

Stay tuned...

The Design Goals of Internet, v1

- **Interconnection/Multiplexing** *(packet switching)*
- **Resilience/Survivability** *(fate sharing)*
- Heterogeneity
  - Different types of services
  - Different types of networks
- Distributed management
- Cost effectiveness
- Ease of attachment
- Accountability

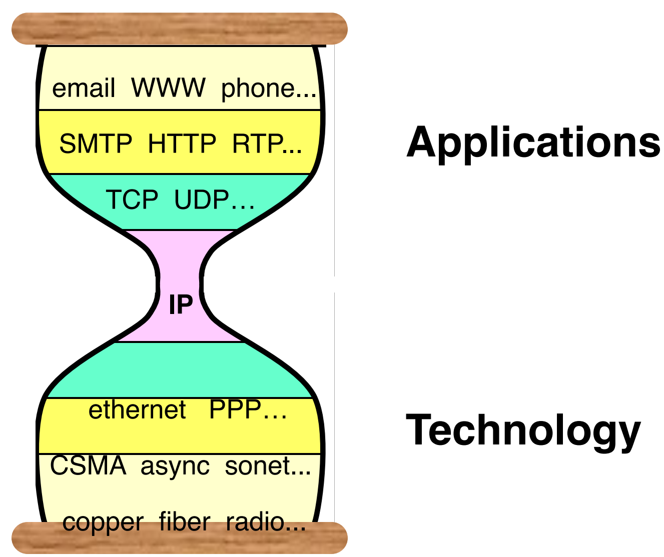
Decreasing Priority

"This set of goals might seem to be nothing more than a checklist of all the desirable network features. It is important to understand that these goals are in order of importance, and **an entirely different network architecture would result if the order were changed**."

These goals were prioritized for a military network.
Should priorities change as the network evolves?

Fundamental Goal: Interconnection

- Need to interconnect many existing networks
- Hide underlying technology from applications
- Decisions:
  - Network provides minimal functionality
  - ***"Narrow waist"***



Tradeoff: **No assumptions, no guarantees.**

The “Curse of the Narrow Waist”

- IP over anything, anything over IP
  - Has allowed for much innovation both above and below the IP layer of the stack
  - An IP stack gets a device on the Internet

- **Drawback:** very difficult to make changes to IP
  - But…people are trying
  - NSF GENI project: http://www.geni.net/

Goal #2: Survivability

- Network should continue to work, even if some devices fail, are compromised, etc.
- Failures on the Abilene (Internet 2) backbone network over the course of 6 months

	instability	unavailability	maintenance	total
node	0	2	22	24
link	0	20	65	85
peer	14	82	77	173
total	14	104	164	282

Thanks to Yiyi Huang

How well does the current Internet support survivability?

Goal #2: Survivability

Two Options

- Replication
  - Keep state at multiple places in the network, recover when nodes crash
- **Fate-sharing**
  - Acceptable to lose state information for some entity if the entity itself is lost

Reasons for Fate Sharing

- Can support arbitrarily complex failure scenarios
- Engineering is easier

Goal #3: Heterogeneous Services

- TCP/IP designed as a monolithic transport
  - TCP for flow control, reliable delivery
  - IP for forwarding

- Became clear that not every type of application would need reliable, in-order delivery
  - *Example:* Voice and video over networks
  - *Example:* DNS
  - Why don’t these applications require reliable, in-order delivery?
  - Narrow waist: allowed proliferation of transport protocols

Topic: Voice and Video over Networks

- **Deadlines:** Timeliness more important than 100% reliability.
- Propagation of errors: Some losses more devastating than others

Loss in “Anchor” Frame (I-Frame) → Propagates to “Dependent” Frames (P and B-Frames)

Goal #3b: Heterogeneous Networks

- Build minimal functionality into the network
  - No need to re-engineering for each type of network
- “Best effort” service model.
  - Lost packets
  - Out-of-order packets
  - No quality guarantees
  - *No information about failures, performance, etc.*

Tradeoff: **Network management more difficult**

Goal #4: Distributed Management

Many examples:

- Addressing (ARIN, RIPE, APNIC, etc.)
  - Though this was recently threatened.
- Naming (DNS)
- Routing (BGP)

No single entity in charge.
Allows for organic growth, scalable management.

Tradeoff: **No one party has visibility/control.**

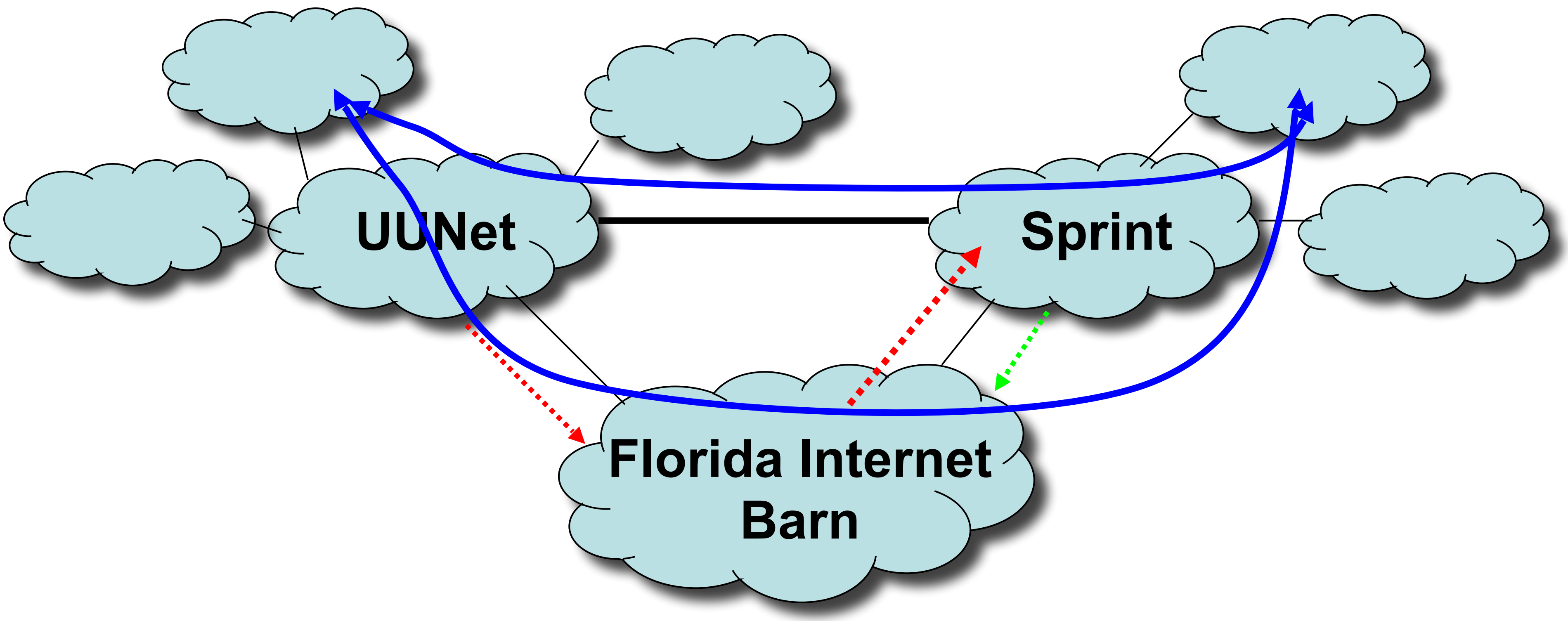
No Owner, No Responsible Party

"Some of the most significant problems with the Internet today relate to lack of sufficient tools for distributed management, especially in the area of routing."

- Hard to figure out who/what's causing a problem
- Worse yet, local actions have global effects…

Local Actions, Global Consequences

"…a glitch at a small ISP… triggered a major outage in Internet access across the country. The problem started when MAI Network Services...passed bad router information from one of its customers onto Sprint." -- *news.com*, April 25, 1997



Goal #5: Cost Effectiveness

- Packet headers introduce high overhead
- End-to-end retransmission of lost packets
  - Potentially wasteful of bandwidth by placing burden on the edges of the network

Goal #6: Ease of Attachment

- IP is “plug and play” Anything with a working IP stack can connect to the Internet (hourglass model)
- A huge success!
  - **Lesson:** Lower the barrier to innovation/entry and people will get creative (*e.g.,* Cerf and Kahn probably did not think about IP stacks on phones, sensors, etc.)
- But….

Tradeoff: **Burden on end systems/programmers.**

Goal #7: Accountability

- **Note:** Accountability mentioned in early papers on TCP/IP, but not prioritized
- Datagram networks make accounting tricky.
  - The phone network has had an easier time figuring out billing
  - Payments/billing on the Internet is much less precise

Tradeoff: **Broken payment models and incentives.**

What's Missing?

- Security
- Availability
- Accountability (the other kind)
- Support for disconnected/intermittent operation
- Mobility
- Scaling
- …

End-to-End Arguments in System Design

[Saltzer, Reed, Clark 1981]

End-to-end in a nutshell

"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.)"

Some Consequences

- In layered design, the E2E principle provides guidance on where functions belong.
- "Dumb, minimal" network and "intelligent" end-points.
- Many argue that:

  E2E principle allowed Internet to grow rapidly because innovation took place at the edge, in applications and services.

Careful file transfer

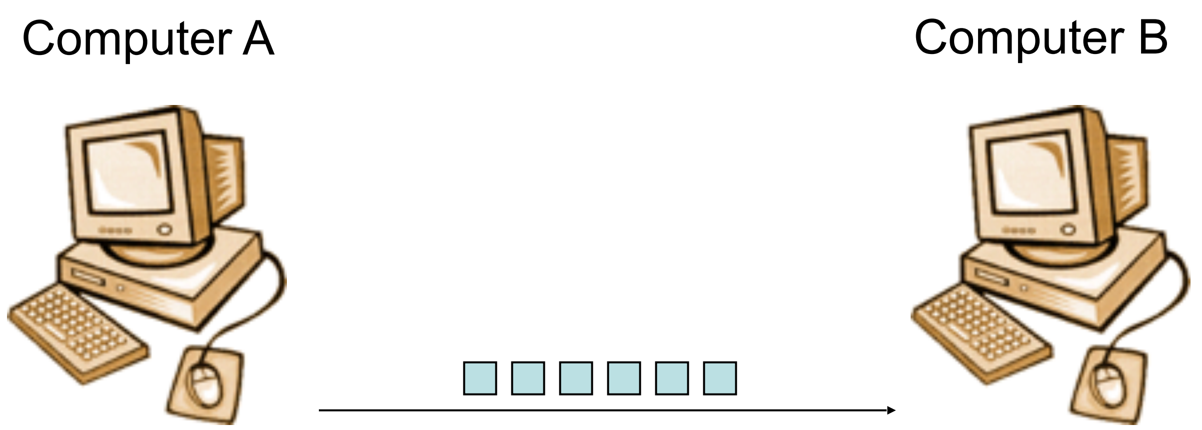
Careful file transfer

1) File transfer program on A asks file system to read F from disk

Careful file transfer

1) File transfer program on A asks file system to read F from disk
2) File transfer program on A asks communication system to send file

Careful file transfer



1) File transfer program on A asks file system to read F from disk
2) File transfer program on A asks communication system to send file
3) Communication system transmits packets

Careful file transfer

1) File transfer program on A asks file system to read F from disk
2) File transfer program on A asks communication system to send file
3) Communication system transmits packets
4) Communication system gives F to file transfer program on B

Careful file transfer

1) File transfer program on A asks file system to read F from disk
2) File transfer program on A asks communication system to send file
3) Communication system transmits packets
4) Communication system gives F to file transfer program on B
5) File transfer program on B asks file system to write F to disk

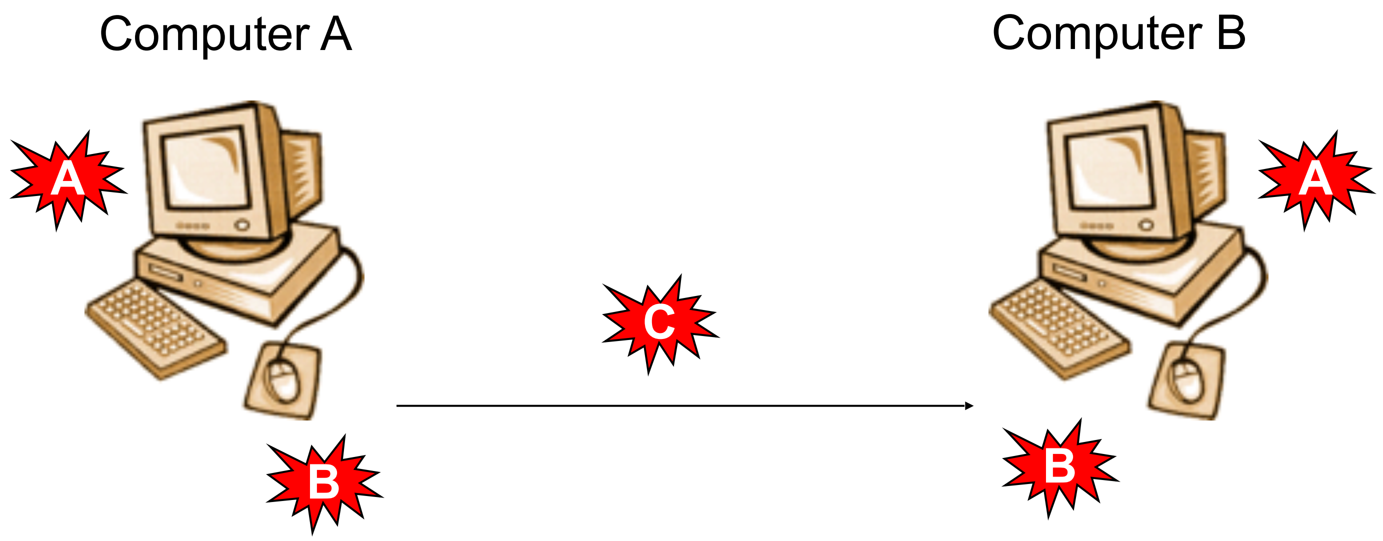
What can go wrong?

A) Reading to and writing from file system

What can go wrong?

A) Reading to and writing from file system
B) Breaking up file / reassembling file

What can go wrong?



A) Reading to and writing from file system
B) Breaking up file / reassembling file
C) Transmitting file over communication system

Possible solution #1

- Ensure each step by some form of error checking: duplicate copies, redundancy, timeout and retry, etc.
  - Packet error checking at each hop
  - Send every packet three times
  - Acknowledge packet reception at each hop

Problems with this solution

1. Not complete; still requires application level checking
2. May not be economical

Possible solution #2

- "End-to-end check and retry"
  - Application commits or retries based on checksum value.
  - If errors along the way are rare, this will most likely finish on first try.

Performance

- Lower levels can be reliable as a performance booster
  - Transferring large files
  - Regardless of data communication, end-to-end check must be done
- Tradeoff based on performance, not correctness
  - Is the amount of effort put into the reliability worth the performance gain?

On the other hand…

E2E principle appears to have been diluted: NATs, firewalls, VPN tunnel endpoints, …

- Perhaps not surprising: E2E principle grew in an era of trust among users. Now network must protect *itself*.

The network is no longer “dumb, minimal”

- Now over 6,000 RFCs.
- Router OS’s based on over 10M lines of source code.

Q: Is this a problem?