Principles of Internet Design

CS204: Advanced Computer Networks

Oct 4, 2023

Adapted from Jiasi’s CS 204 slides for Spring 23

Agenda

- Internet History
- More Overview
  - Modularity through layering
  - Data, control, and management plane
  - Best-effort packet delivery
- Design philosophy of the Internet
  - What we have now
  - How it came about

Q: What are the general principles of Internet design?

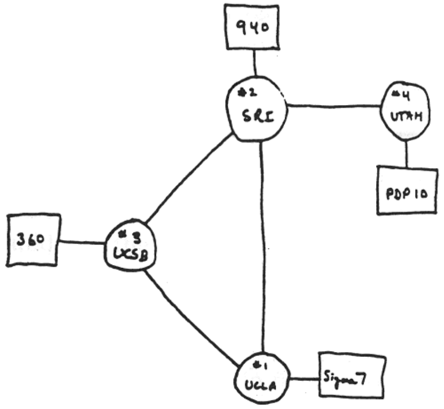
Internet History

Internet history

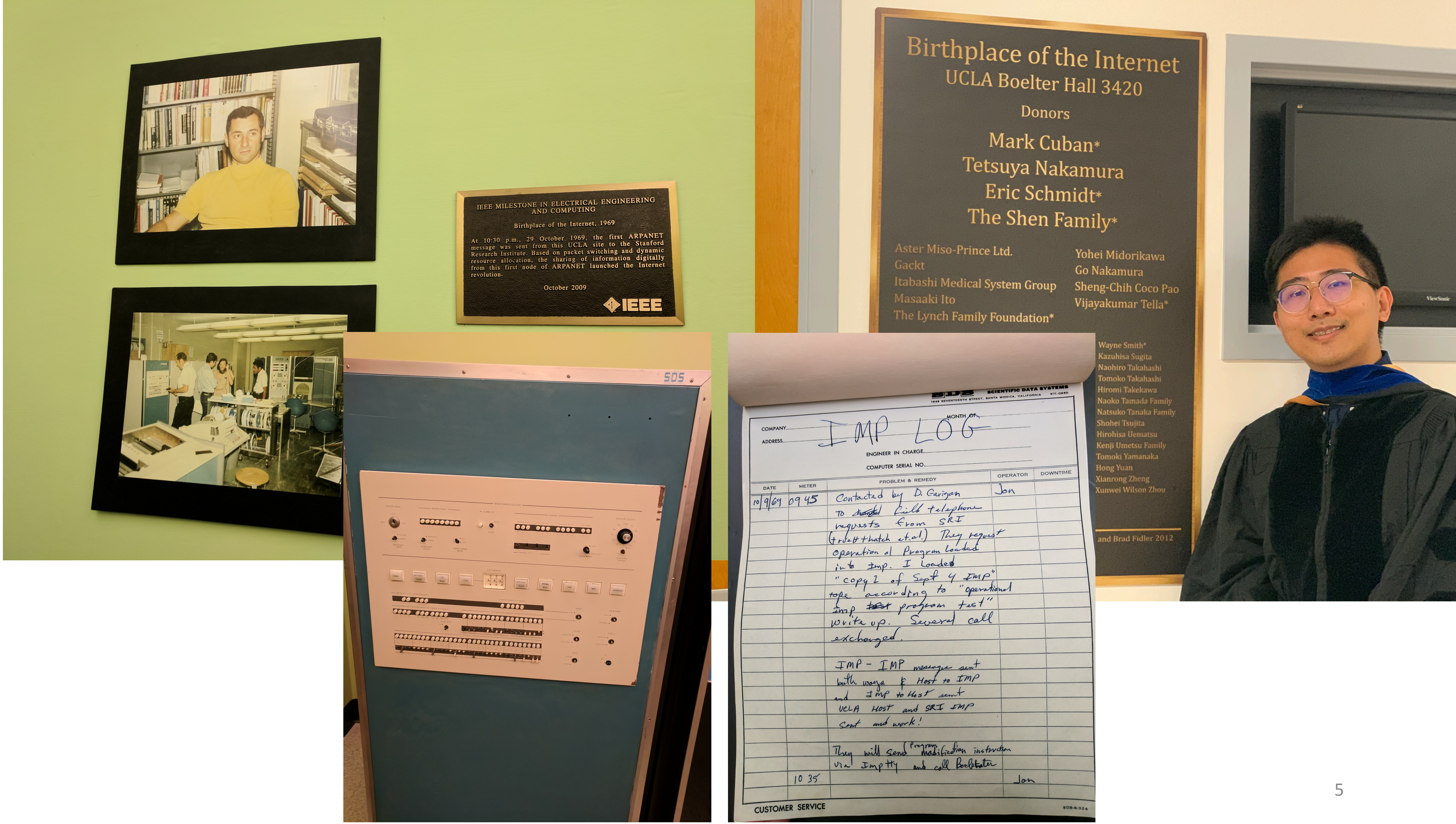
1961-1972: Early packet-switching principles

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational

- 1972:
  - ARPAnet public demo
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes



THE ARPA NETWORK



IEEE MILESTONE IN ELECTRICAL ENGINEERING AND COMPUTING

Birthplace of the Internet, 1969

At 10:30 p.m., 29 October 1969, the first ARPANET message was sent from this UCLA site to the Stanford Research Institute. Based on packet switching and dynamic resource allocation, the sharing of information digitally from this first node of ARPANET launched the Internet revolution.

October 2009

IEEE

SDS SCIENTIFIC DATA SYSTEMS
1649 SEVENTEENTH STREET, SANTA MONICA, CALIFORNIA 871-0960

COMPANY IMP LOG MONTH OF
ADDRESS
ENGINEER IN CHARGE
COMPUTER SERIAL NO.

DATE	METER	PROBLEM & REMEDY	OPERATOR	DOWNTIME
10/9/69	0945	Contacted by D. Garigan To ~~handel~~ field telephone requests from SRI (truett + thatch et.al.) They request operation of Program loaded into Imp. I Loaded "copy 1 of Sept 4 IMP" tape according to "operational imp ~~test~~ program test" write up. Several call exchanged.	Jon	
		IMP - IMP messages sent both ways & Host to IMP and IMP to Host sent UCLA Host and SRI IMP Sent and work!		
		They will send Program modification instruction via Imptty and call Beckstater		
	1035		Jon	

CUSTOMER SERVICE SDS-S-324

Birthplace of the Internet
UCLA Boelter Hall 3420

Donors

Mark Cuban*
Tetsuya Nakamura
Eric Schmidt*
The Shen Family*

Aster Miso-Prince Ltd.
Gackt
Itabashi Medical System Group
Masaaki Ito
The Lynch Family Foundation*

Yohei Midorikawa
Go Nakamura
Sheng-Chih Coco Pao
Vijayakumar Tella*

Wayne Smith*
Kazuhisa Sugita
Naohiro Takahashi
Tomoko Takahashi
Hiromi Takekawa
Naoko Tamada Family
Natsuko Tanaka Family
Shohei Tsujita
Hirohisa Uematsu
Kenji Umetsu Family
Tomoki Yamanaka
Hong Yuan
Xianrong Zheng
Xunwei Wilson Zhou

and Brad Fidler 2012

Internet history

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- late70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today's Internet architecture

Internet history

1980-1990: new protocols, a proliferation of networks

- 1983: deployment of TCP/IP
- 1982: smtp e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: ftp protocol defined
- 1988: TCP congestion control

- new national networks: Csnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

Internet history

1990, 2000's: commercialization, the Web, new apps

- early 1990's: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990's: commercialization of the Web

late 1990's – 2000's:

- more killer apps: instant messaging, P2P file sharing
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gbps

Internet history

2005-present

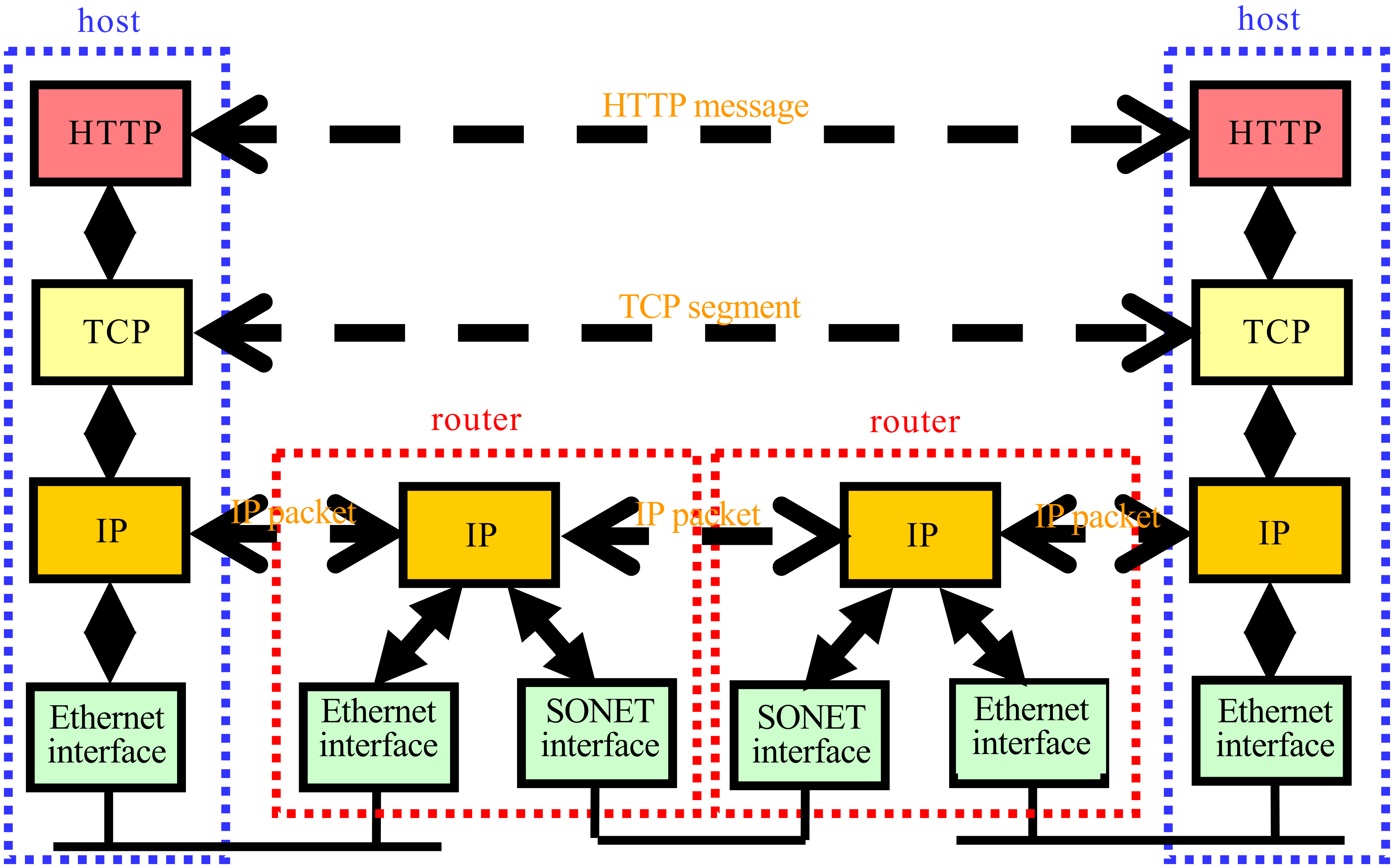
- ~1 billion hosts
  - Smartphones and tablets
- Aggressive deployment of broadband access
- Increasing ubiquity of high-speed wireless access
- Emergence of online social networks:
  - Facebook: nearly 2 billion active users
  - Tiktok: 1 billion monthly active users
- Service providers (Google, Microsoft) create their own networks
  - Bypass Internet, providing “instantaneous” access to search, email, etc.
- E-commerce, universities, enterprises running their services in “cloud” (e.g., Amazon EC2)
- Live video conference (e.g., Zoom)

More review on Internet

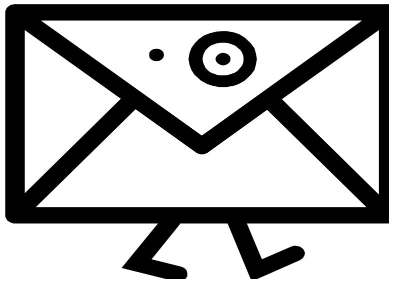
Modularity through layering

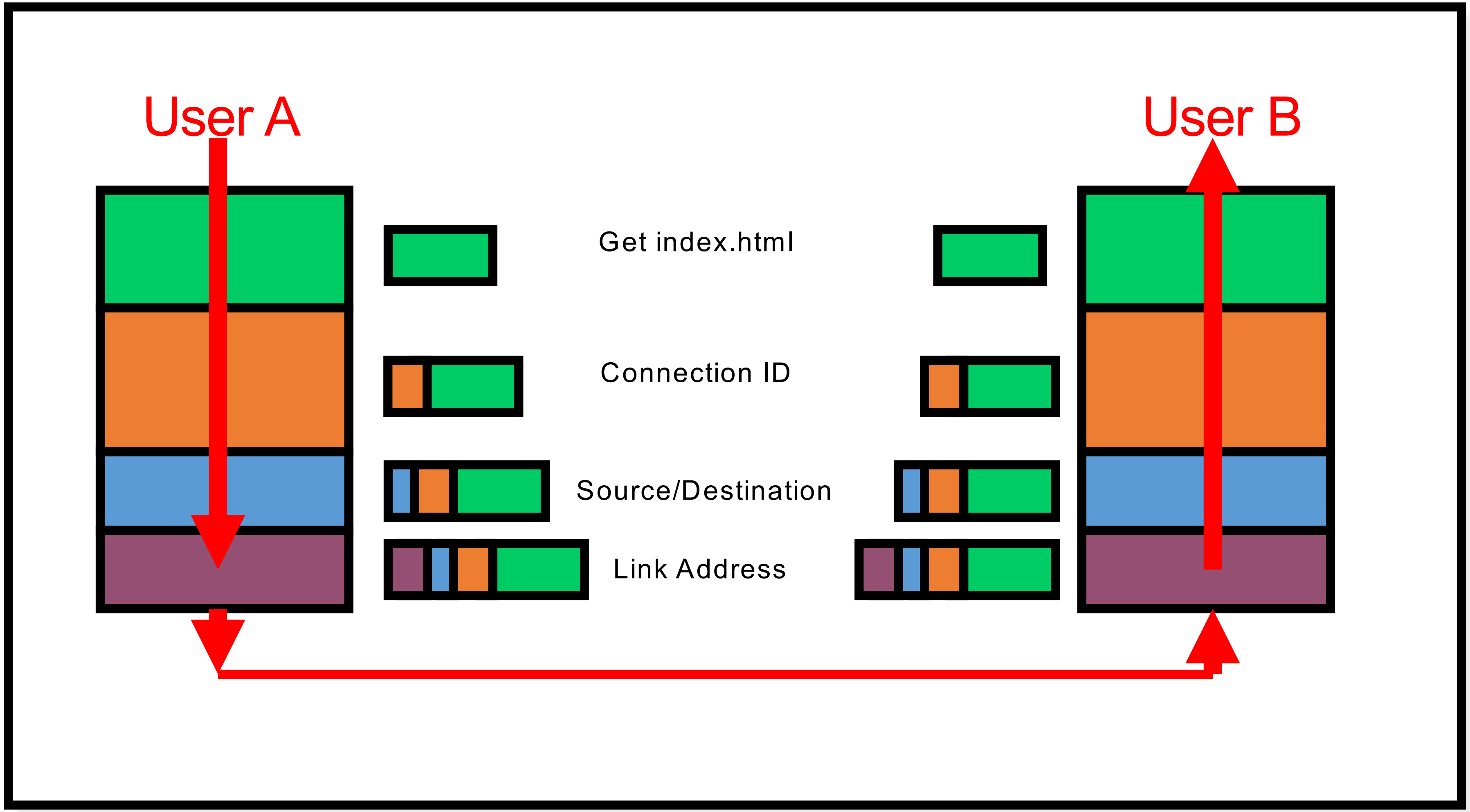
Protocol Stack

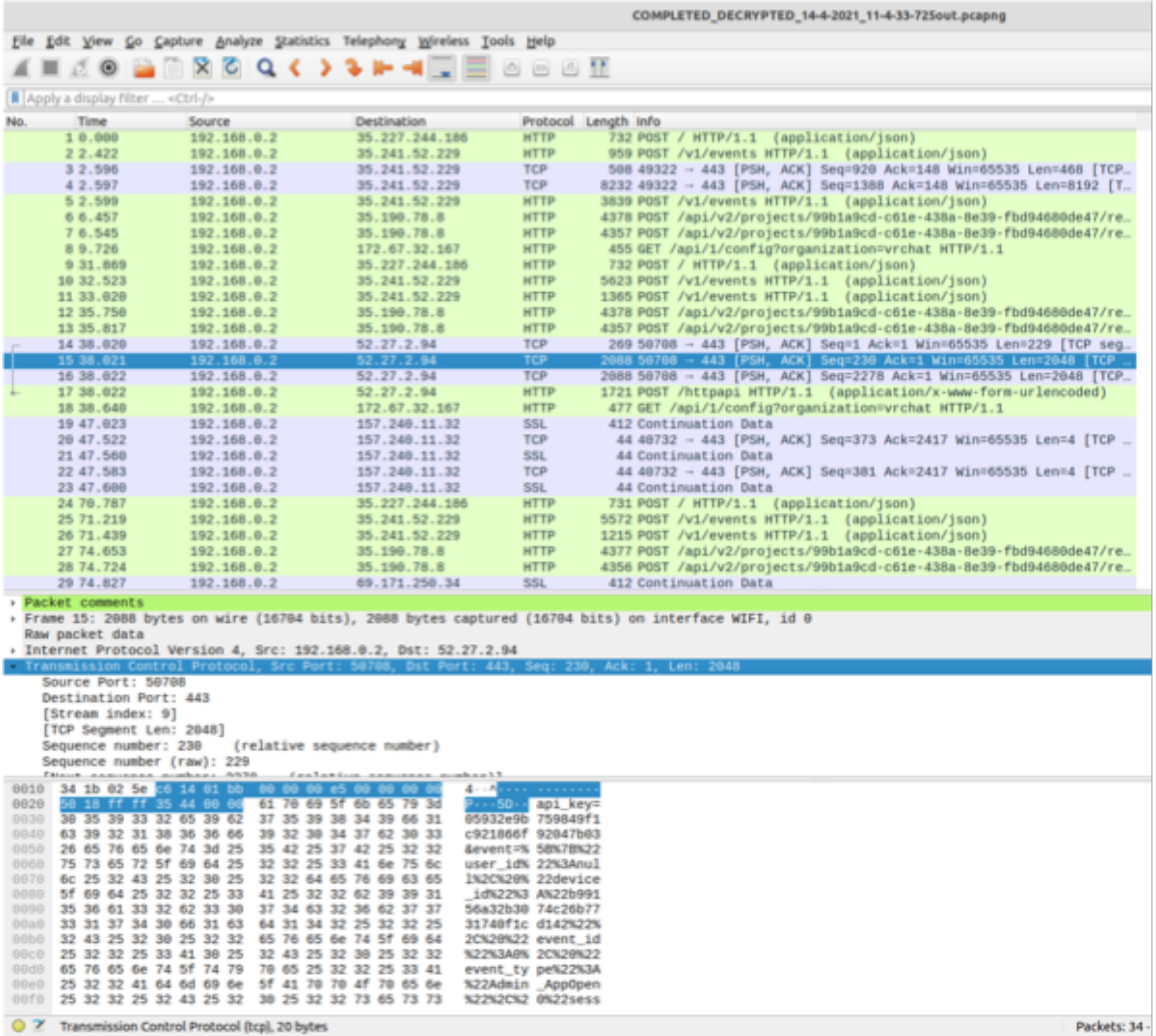
IP Suite: End Hosts vs. Routers



Layer Encapsulation







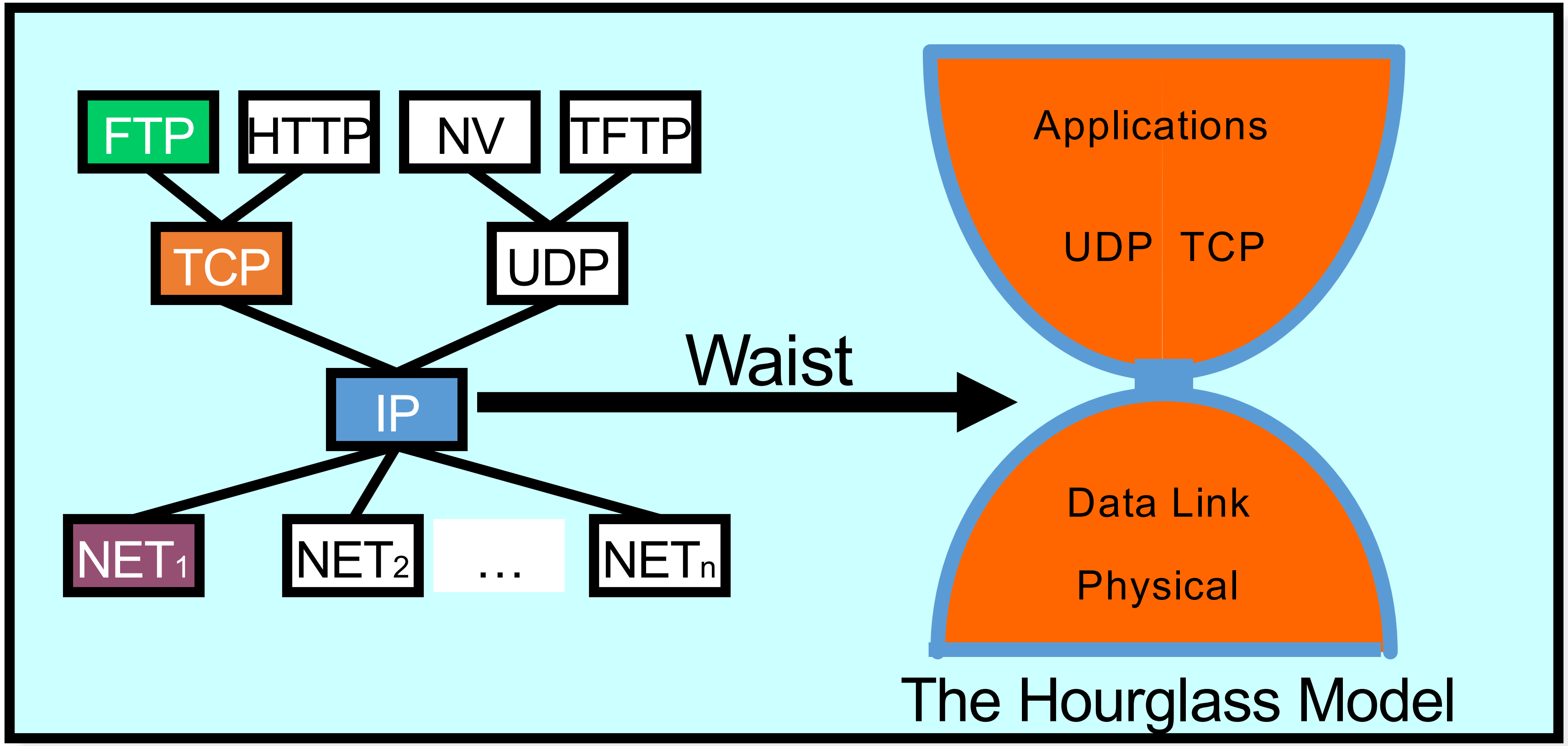
```
No.  Time    Source        Destination     Protocol Length Info
   1 0.000   192.168.0.2   35.227.244.186  HTTP      732 POST / HTTP/1.1  (application/json)
   2 2.422   192.168.0.2   35.241.52.229   HTTP      959 POST /v1/events HTTP/1.1  (application/json)
   3 2.596   192.168.0.2   35.241.52.229   TCP       508 49322 → 443 [PSH, ACK] Seq=920 Ack=148 Win=65535 Len=468 [TCP…
   4 2.597   192.168.0.2   35.241.52.229   TCP      8232 49322 → 443 [PSH, ACK] Seq=1388 Ack=148 Win=65535 Len=8192 [T…
   5 2.599   192.168.0.2   35.241.52.229   HTTP     3839 POST /v1/events HTTP/1.1  (application/json)
   6 6.457   192.168.0.2   35.190.78.8     HTTP     4378 POST /api/v2/projects/99b1a9cd-c61e-438a-8e39-fbd94680de47/re…
   7 6.545   192.168.0.2   35.190.78.8     HTTP     4357 POST /api/v2/projects/99b1a9cd-c61e-438a-8e39-fbd94680de47/re…
   8 9.726   192.168.0.2   172.67.32.167   HTTP      455 GET /api/1/config?organization=vrchat HTTP/1.1
   9 31.869  192.168.0.2   35.227.244.186  HTTP      732 POST / HTTP/1.1  (application/json)
  10 32.523  192.168.0.2   35.241.52.229   HTTP     5623 POST /v1/events HTTP/1.1  (application/json)
  11 33.020  192.168.0.2   35.241.52.229   HTTP     1365 POST /v1/events HTTP/1.1  (application/json)
  12 35.750  192.168.0.2   35.190.78.8     HTTP     4378 POST /api/v2/projects/99b1a9cd-c61e-438a-8e39-fbd94680de47/re…
  13 35.817  192.168.0.2   35.190.78.8     HTTP     4357 POST /api/v2/projects/99b1a9cd-c61e-438a-8e39-fbd94680de47/re…
  14 38.020  192.168.0.2   52.27.2.94      TCP       269 50708 → 443 [PSH, ACK] Seq=1 Ack=1 Win=65535 Len=229 [TCP seg…
  15 38.021  192.168.0.2   52.27.2.94      TCP      2088 50708 → 443 [PSH, ACK] Seq=230 Ack=1 Win=65535 Len=2048 [TCP …
  16 38.022  192.168.0.2   52.27.2.94      TCP      2088 50708 → 443 [PSH, ACK] Seq=2278 Ack=1 Win=65535 Len=2048 [TCP…
  17 38.022  192.168.0.2   52.27.2.94      HTTP     1721 POST /httpapi HTTP/1.1  (application/x-www-form-urlencoded)
  18 38.640  192.168.0.2   172.67.32.167   HTTP      477 GET /api/1/config?organization=vrchat HTTP/1.1
  19 47.023  192.168.0.2   157.240.11.32   SSL       412 Continuation Data
  20 47.522  192.168.0.2   157.240.11.32   TCP        44 40732 → 443 [PSH, ACK] Seq=373 Ack=2417 Win=65535 Len=4 [TCP …
  21 47.560  192.168.0.2   157.240.11.32   SSL        44 Continuation Data
  22 47.583  192.168.0.2   157.240.11.32   TCP        44 40732 → 443 [PSH, ACK] Seq=381 Ack=2417 Win=65535 Len=4 [TCP …
  23 47.600  192.168.0.2   157.240.11.32   SSL        44 Continuation Data
  24 70.787  192.168.0.2   35.227.244.186  HTTP      731 POST / HTTP/1.1  (application/json)
  25 71.219  192.168.0.2   35.241.52.229   HTTP     5572 POST /v1/events HTTP/1.1  (application/json)
  26 71.439  192.168.0.2   35.241.52.229   HTTP     1215 POST /v1/events HTTP/1.1  (application/json)
  27 74.653  192.168.0.2   35.190.78.8     HTTP     4377 POST /api/v2/projects/99b1a9cd-c61e-438a-8e39-fbd94680de47/re…
  28 74.724  192.168.0.2   35.190.78.8     HTTP     4356 POST /api/v2/projects/99b1a9cd-c61e-438a-8e39-fbd94680de47/re…
  29 74.827  192.168.0.2   69.171.250.34   SSL       412 Continuation Data
```

```
› Packet comments
› Frame 15: 2088 bytes on wire (16704 bits), 2088 bytes captured (16704 bits) on interface WIFI, id 0
  Raw packet data
› Internet Protocol Version 4, Src: 192.168.0.2, Dst: 52.27.2.94
˅ Transmission Control Protocol, Src Port: 50708, Dst Port: 443, Seq: 230, Ack: 1, Len: 2048
    Source Port: 50708
    Destination Port: 443
    [Stream index: 9]
    [TCP Segment Len: 2048]
    Sequence number: 230    (relative sequence number)
    Sequence number (raw): 229
```

```
0010  34 1b 02 5e c6 14 01 bb  00 00 00 e5 00 00 00 00   4··^···· ········
0020  50 18 ff ff 35 44 00 00  61 70 69 5f 6b 65 79 3d   P···5D·· api_key=
0030  30 35 39 33 32 65 39 62  37 35 39 38 34 39 66 31   05932e9b 759849f1
0040  63 39 32 31 38 36 36 66  39 32 30 34 37 62 30 33   c921866f 92047b03
0050  26 65 76 65 6e 74 3d 25  35 42 25 37 42 25 32 32   &event=% 5B%7B%22
0060  75 73 65 72 5f 69 64 25  32 32 25 33 41 6e 75 6c   user_id% 22%3Anul
0070  6c 25 32 43 25 32 30 25  32 32 64 65 76 69 63 65   l%2C%20% 22device
0080  5f 69 64 25 32 32 25 33  41 25 32 32 62 39 39 31   _id%22%3 A%22b991
0090  35 36 61 33 32 62 33 30  37 34 63 32 36 62 37 37   56a32b30 74c26b77
00a0  33 31 37 34 30 66 31 63  64 31 34 32 25 32 32 25   31740f1c d142%22%
00b0  32 43 25 32 30 25 32 32  65 76 65 6e 74 5f 69 64   2C%20%22 event_id
00c0  25 32 32 25 33 41 30 25  32 43 25 32 30 25 32 32   %22%3A0% 2C%20%22
00d0  65 76 65 6e 74 5f 74 79  70 65 25 32 32 25 33 41   event_ty pe%22%3A
00e0  25 32 32 41 64 6d 69 6e  5f 41 70 70 4f 70 65 6e   %22Admin _AppOpen
00f0  25 32 32 25 32 43 25 32  30 25 32 32 73 65 73 73   %22%2C%2 0%22sess
```

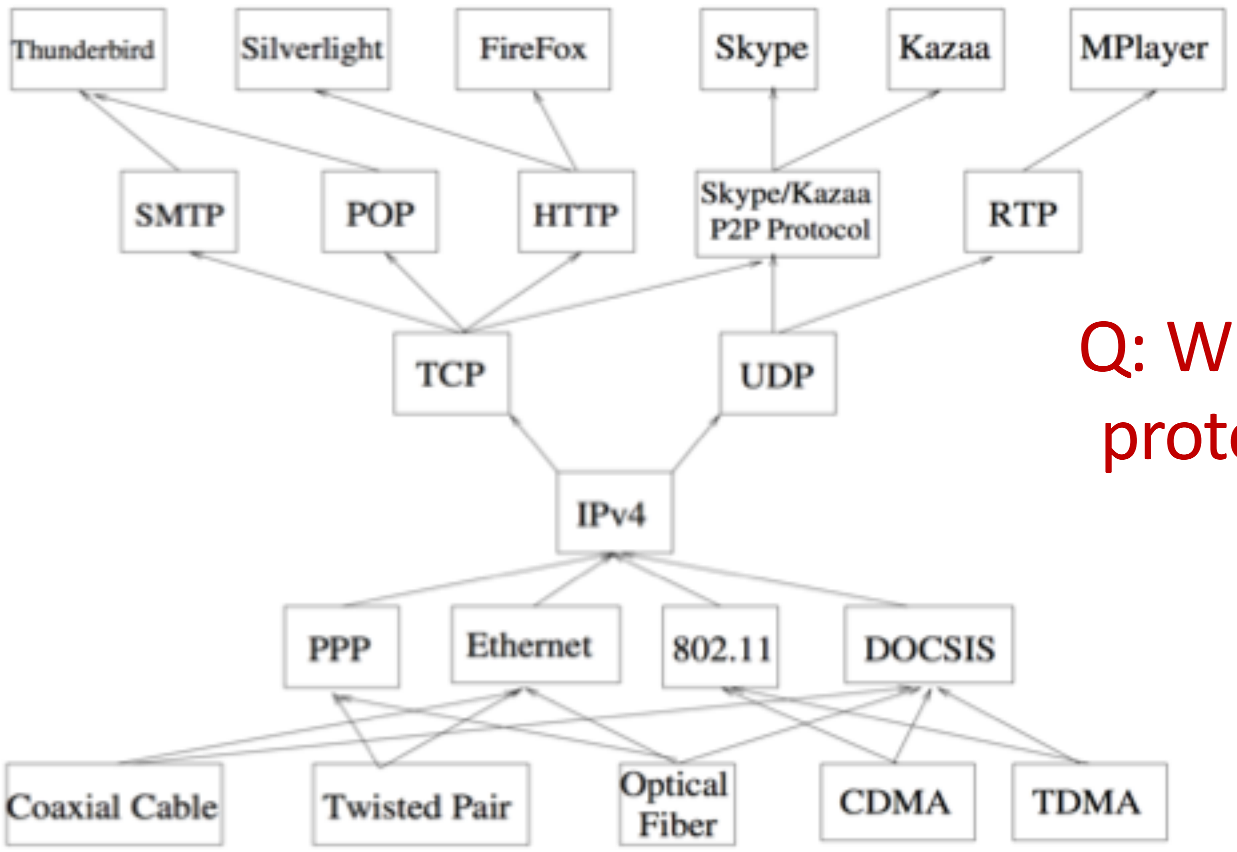
Transmission Control Protocol (tcp), 20 bytes

The “Narrow Waist” of IP



The waist facilitates interoperability

The narrow waist of IP



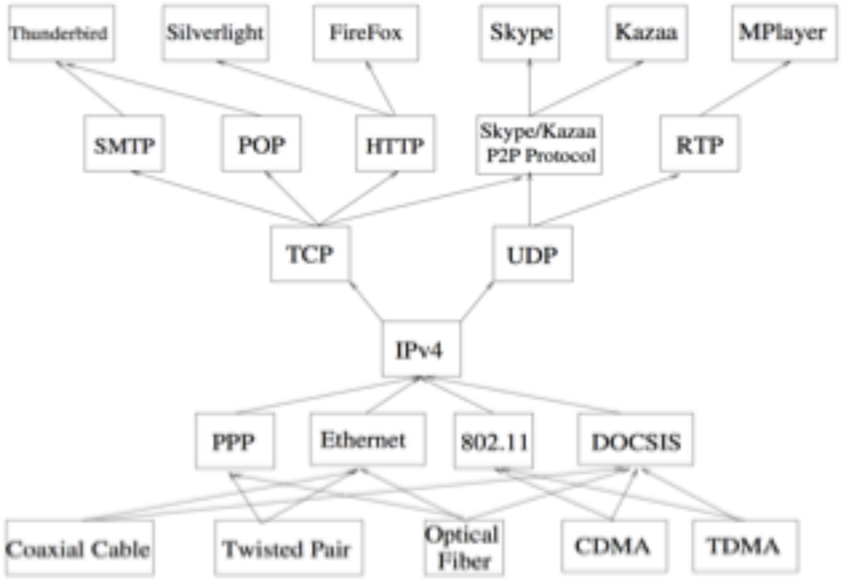
Q: Why does the Internet protocol stack resemble an hourglass?

“The Evolution of Layered Protocol Stacks Leads to an Hourglass-Shaped Architecture”, *SIGCOMM* 2011.

Q: Why does the Internet protocol stack resemble an hourglass?

- Theory 1
  - IP is a global address, so no need for two naming systems?
- Theory 2
  - Lower layers are diverse (e.g. wireless, optical, cable)
  - Higher layers are also diverse (e.g. voice, video, file transfer)

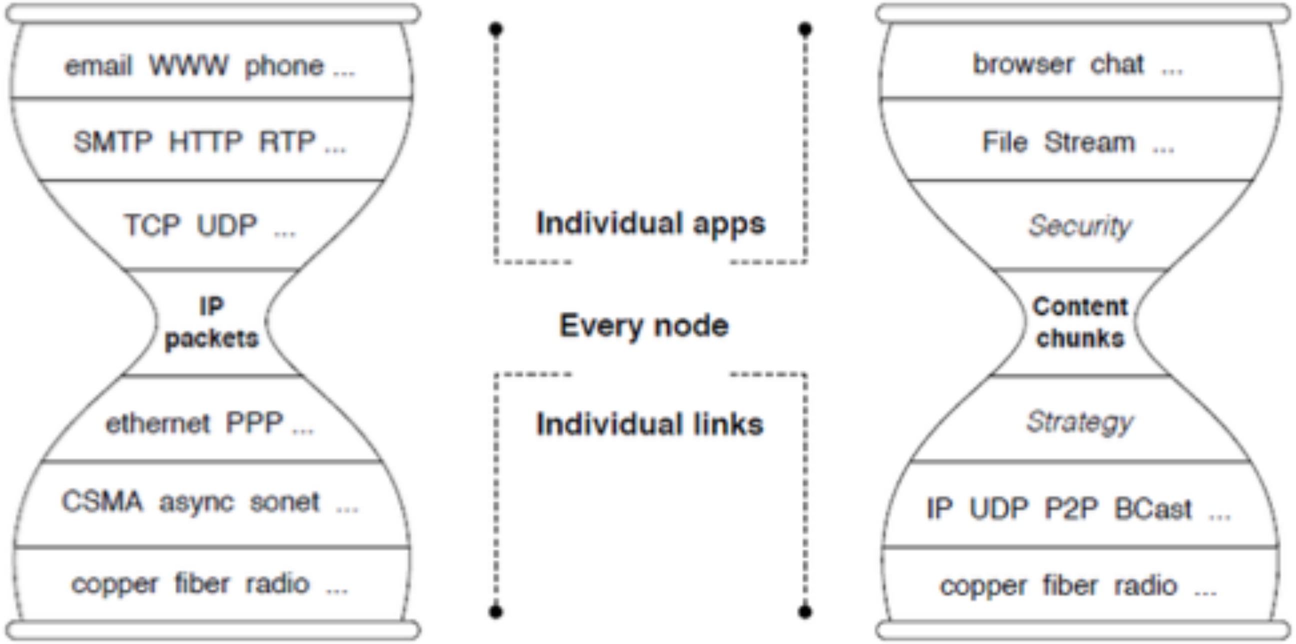
  → IP layer in the middle must be more general (and hence unique)?



- Theory 3
  - Analytic birth/death model?

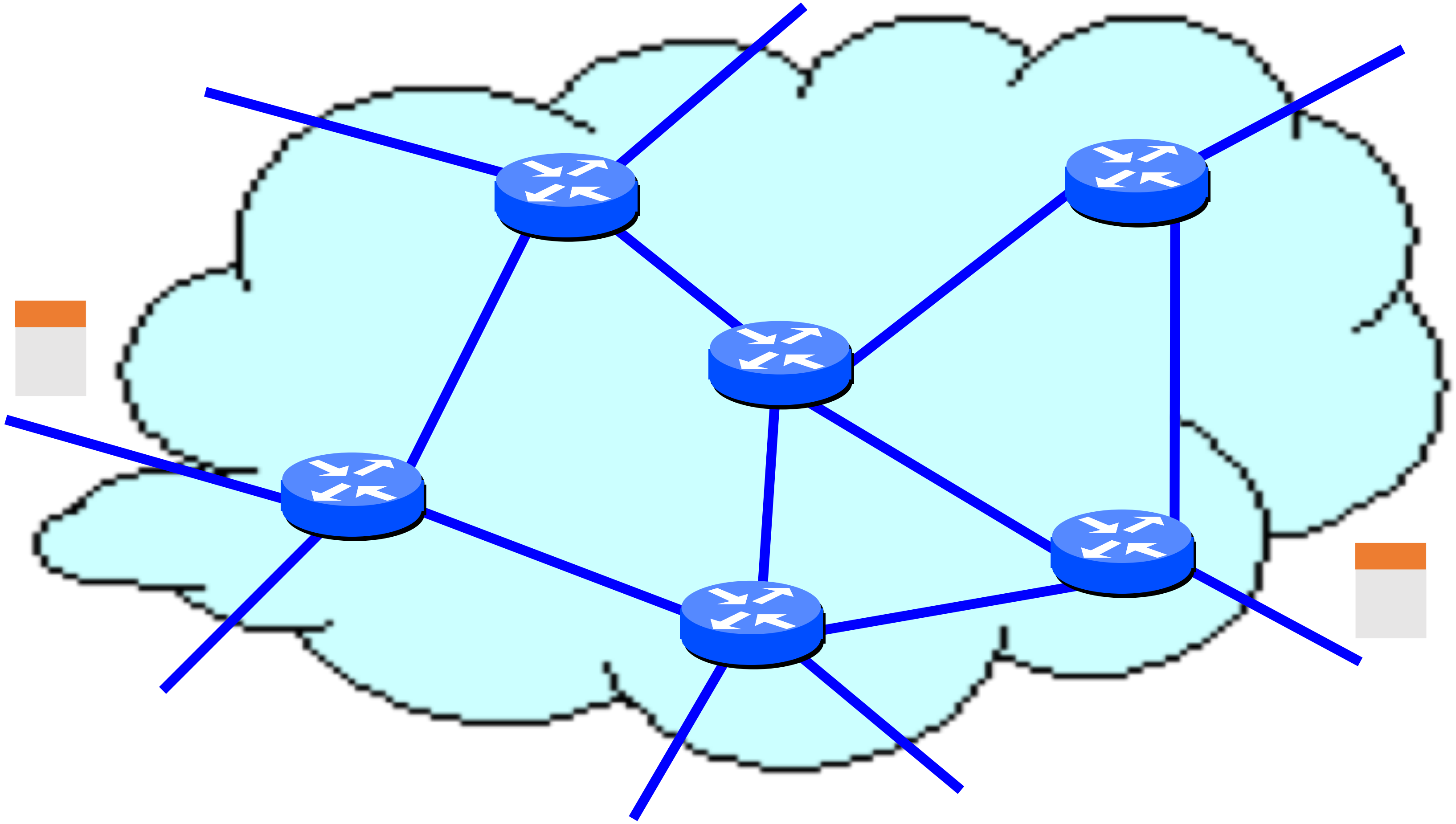
Any Alternatives to TCP/IP?

- Named data networking (NDN)
    - Universal names for content, instead of IP addresses
        - e.g., weather/riverside/yesterday, video/horror/freddy/chunk1
    - Forwarding and routing based on name prefixes



Data, Control, and Management Planes

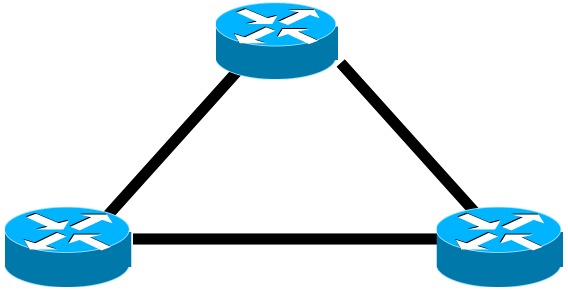
Inside the Network



Forward packets from the sender to the receiver

Split into Data vs. Control Plane

- Data plane: packets
  - Handle individual packets as they arrive
  - Forward, drop, or buffer
  - Mark, shape, schedule, …

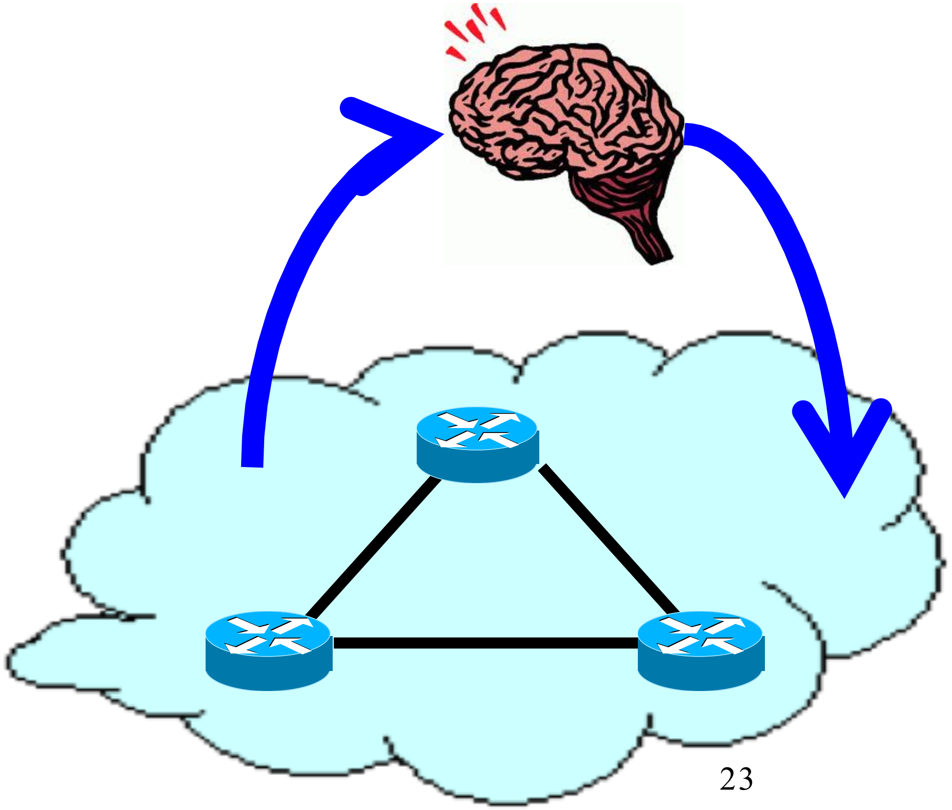


- Control plane: events
  - Track changes in network topology
  - Compute paths through the network
  - Reserve resources along a path

Motivated by need for high-speed packet forwarding

Adding the Management Plane

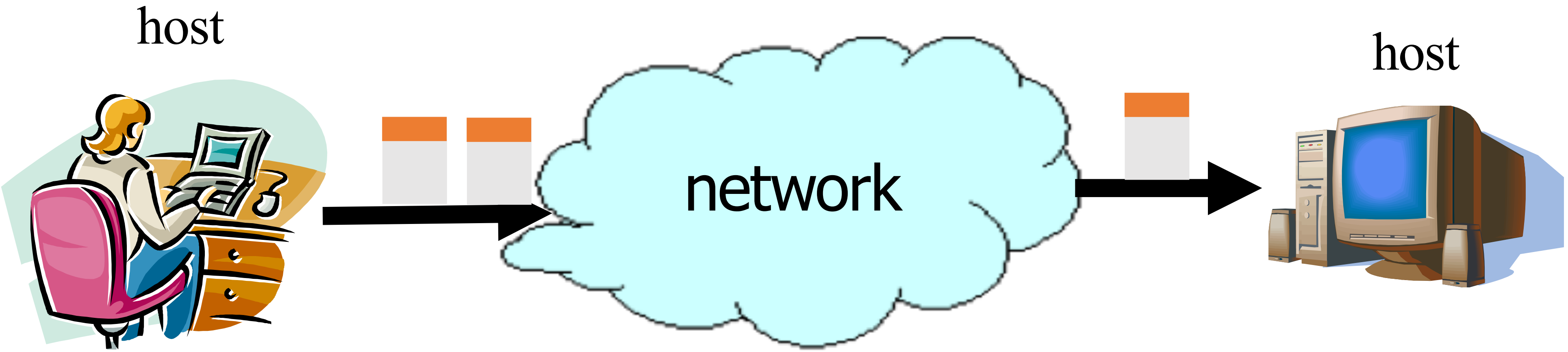
- Making the network run *well*
  - Traffic reaches the right destination
  - Traffic flows over short, uncongested paths
  - Unwanted traffic is discarded
  - Failure recovery happens quickly
  - Routers don't run out of resources
- A control loop with the network
  - Measure (sense): topology, traffic, performance, …
  - Control (actuate): configure control and data planes
- Time scales?



Best-Effort Packet-Delivery Service

Host-Network Division of Labor

- Packet switching
  - Divide messages into a sequence of packets
  - Headers with source and destination address
- Best-effort delivery
  - Packets may be lost
  - Packets may be corrupted
  - Packets may be delivered out of order



Host-Network Interface: Why Packets?

- Data traffic is bursty
  - Logging in to remote machines
  - Exchanging e-mail messages
- Don’t want to waste bandwidth
  - No traffic exchanged during idle periods
- Better to allow multiplexing
  - Different transfers share access to same links
- Packets can be delivered by most anything
  - RFC 1149: IP Datagrams over Avian Carriers

Host-Network Interface: Why Best-Effort?

- Never having to say you're sorry…
  - Don't reserve bandwidth and memory
  - Don't do error detection & correction
  - Don't remember from one packet to next
- Easier to survive failures
  - Transient disruptions are okay during failover
- Can run on nearly any link technology
  - Greater interoperability and evolution

Intermediate Transport Layer

- But, *applications* want efficient, accurate transfer of data in order, in a timely fashion
  - Let the end hosts handle all of that
  - (An example of the "end-to-end argument")
- Transport layer can optionally…
  - Detect and retransmit lost packets
  - Put out-of-order packets back in order
  - Detect and handle corrupted packets
  - Avoid overloading the receiver
  - <insert your requirement here>

Design Philosophy of the Internet

What do we have now, and why?

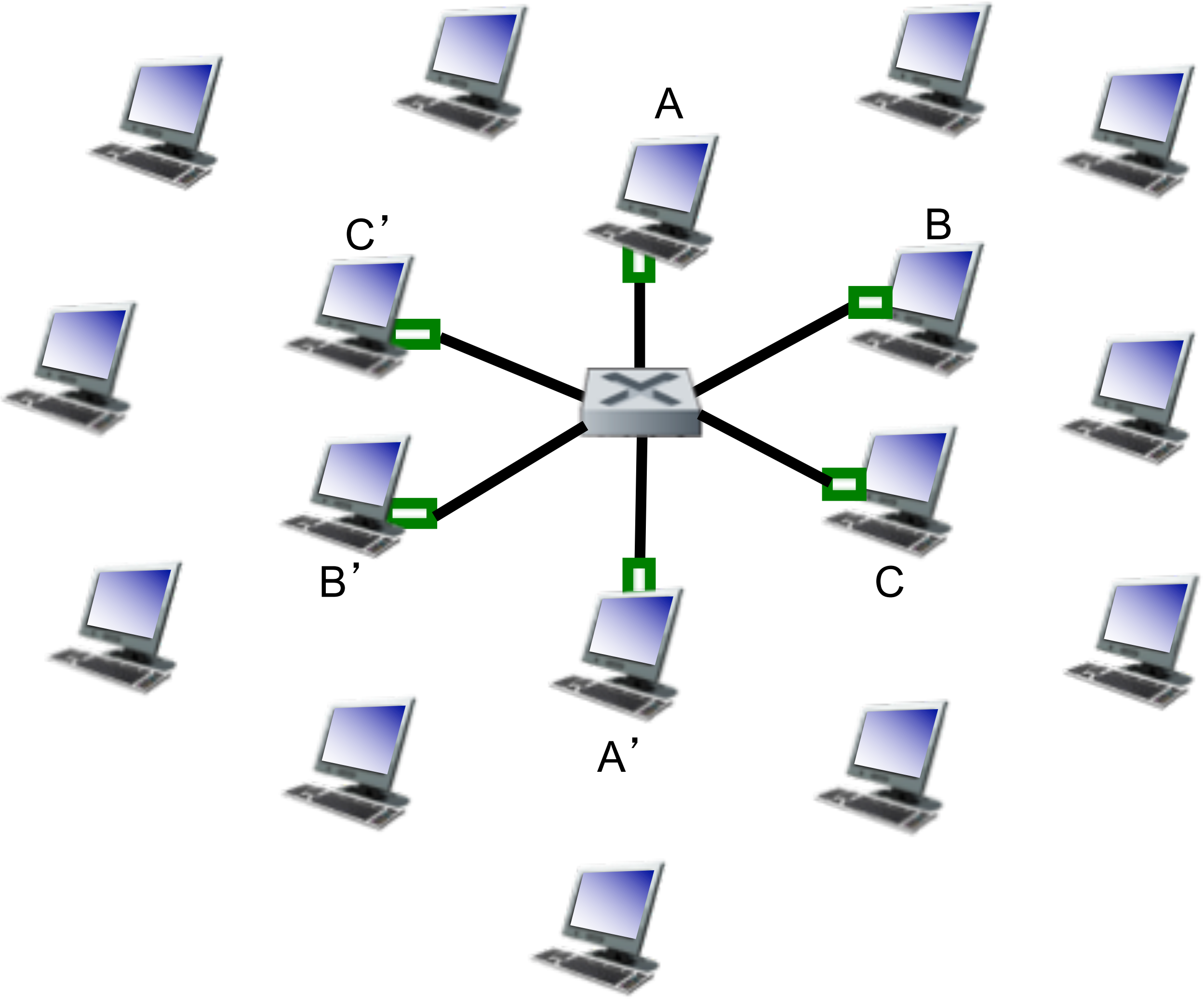
Inter-networking

- *Goal*: scalable network infrastructure that connects different smaller networks together, to enable hosts on different networks to talk to each other.

- LAN approach: connect everyone!

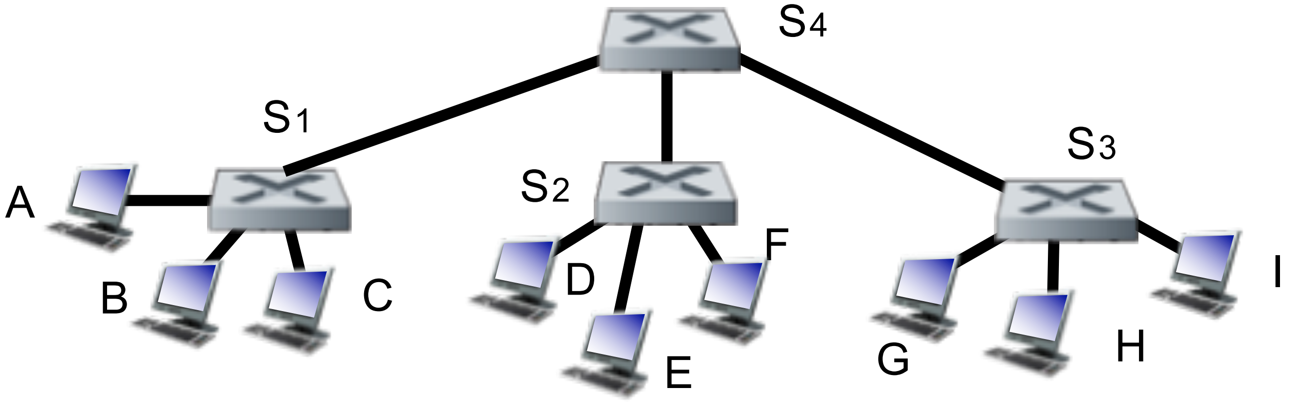
- Key challenges with the LAN approach:
  1. Scaling up
  2. Heterogeneity

Why scaling up doesn't work



What is LAN heterogeneity?

- Sources of heterogeneity
  - Addressing
  - Bandwidth and latency
  - Packet size
  - Loss rates
  - Packet routing



- Gateways provide translation between LANs

Options for gateway functionality

1. Translation: translate between different LAN “languages”
   - Updates: translation may fail if LANs get updated or new features are added
   - Scalability: have to translate between many LANs

2. Unified network layer: define some common “words” that everyone has to understand
   - This is the current design

Key Principles of the original Internet design

1. Internet communication must continue despite loss of networks or gateways.
2. The Internet must support multiple types of communications service.
3. The Internet architecture must accommodate a variety of networks.
4. The Internet architecture must permit distributed management of its resources.
5. The Internet architecture must be cost effective.
6. The Internet architecture must permit host attachment with a low level of effort.
7. The resources used in the internet architecture must be accountable.

Consequences of the robustness design goals

1. Internet communication must continue despite loss of networks or gateways.
    - State contained in the end host, only soft state in the network
        - Soft state = information that times out (goes away) unless refreshed
        - Easily recover from errors
            - E.g., routing protocols automatically update themselves periodically
        - Complicated functionality (e.g. reliability in the transport layer) implemented in the end host
        - Network gateways kept simple

    - Fate sharing of end hosts
        - If end hosts go down, state is lost
        - If gateway fails, network can recover (soft state)

    - Conservative transmission / liberal reception
        - “Be conservative in what you send; be liberal in what you accept”
        - E.g. sender receives ACK for unknown packets; silently drops

Consequences of the universality design goals

2. The Internet must support multiple types of communications service.
    - Different transport-layer protocols (e.g. TCP, UDP)
    - Datagram as fundamental unit supporting the transport protocols

3. The Internet architecture must accommodate a variety of networks.
    - Best-effort service of datagrams
        - No special treatment of different packets (ignoring QoS)
        - No loss recovery (at the network layer)
        - Makes it easier to add new networks
    - IP-over-everything
        - Common set of names (IP addresses) and routing protocols so that gateways know how to behave

Consequences of the mgmt. design goals

4. The Internet architecture must permit distributed management of its resources.
    - Multiple tier-1 ISPs
    - Different intra-domain and inter-domain routing mechanisms

Consequences of the cost design goals

5. The Internet architecture must be cost effective.
    - What is considered cost in this context? Extra header, retransmission, …
    - None of these seems to be problematic now

6. The Internet architecture must permit host attachment with a low level of effort.
    - The cost for attachment was once considered high: Each node has to implement all the protocols that are desired
    - This turns out to be the right choice! Again, “smart edge, dumb core”

Consequences of the design goals

7. The resources used in the internet architecture must be accountable.

- Understanding and monitoring the usage of the resources

These goals, are listed by the author in the order of importance

- Given the possibility of a hostile environment, survivability was put as a first goal, and accountability as a last goal
- Would you re-order the goals in the modern era?

Discussions: Any other goals moving forward?

1. Internet communication must continue despite loss of networks or gateways.
2. The Internet must support multiple types of communications service.
3. The Internet architecture must accommodate a variety of networks.
4. The Internet architecture must permit distributed management of its resources.
5. The Internet architecture must be cost effective.
6. The Internet architecture must permit host attachment with a low level of effort.
7. The resources used in the internet architecture must be accountable.