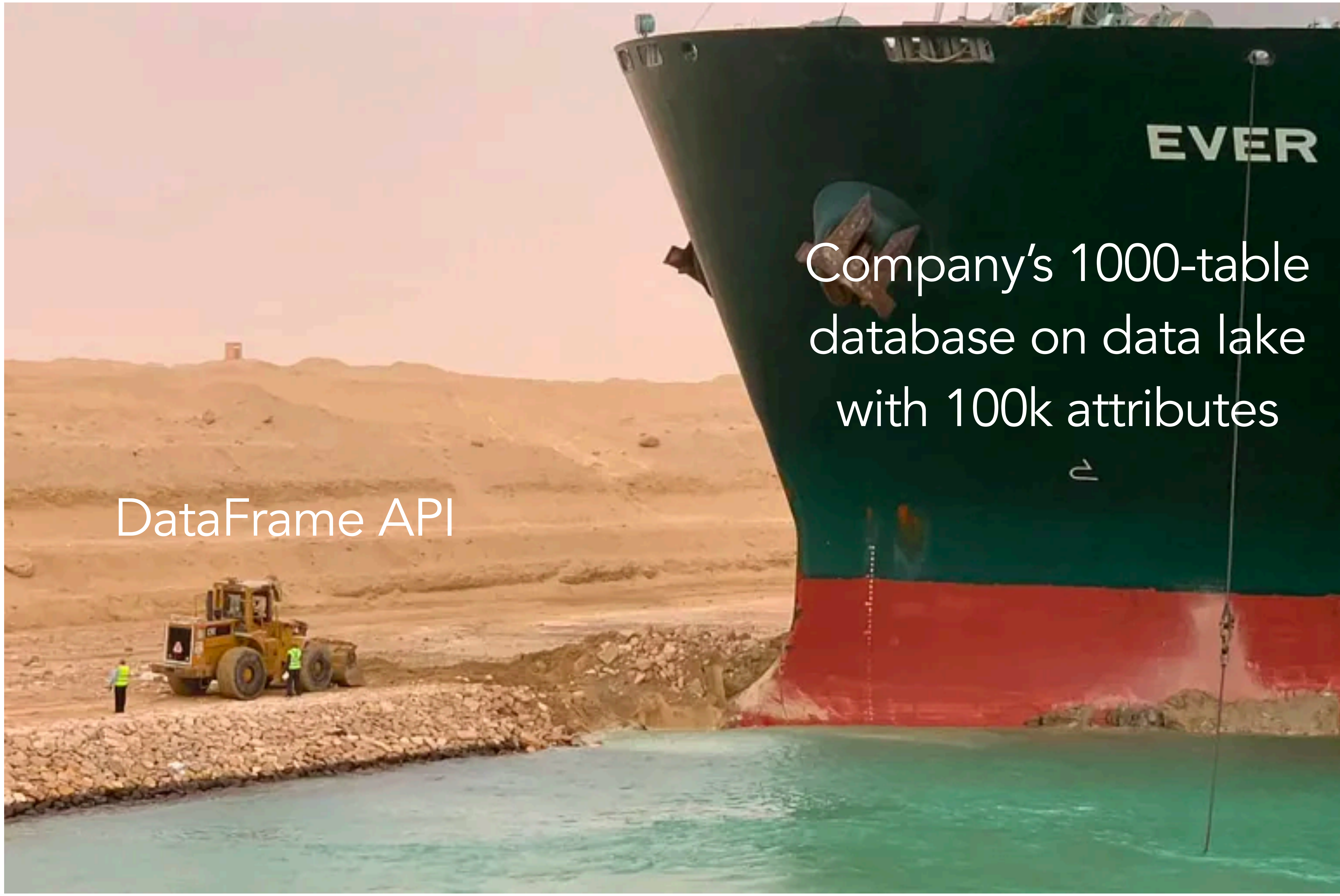
DSC 204a Scalable Data Systems

- Haojian Jin



Meme idea credit: https://datasystemsfun.tumblr.com/

Where are we in the class?

Foundations of Data Systems (2 weeks)

- Digital representation of Data → Computer Organization → Memory hierarchy → Process → Storage

Scaling Distributed Systems (3 weeks)

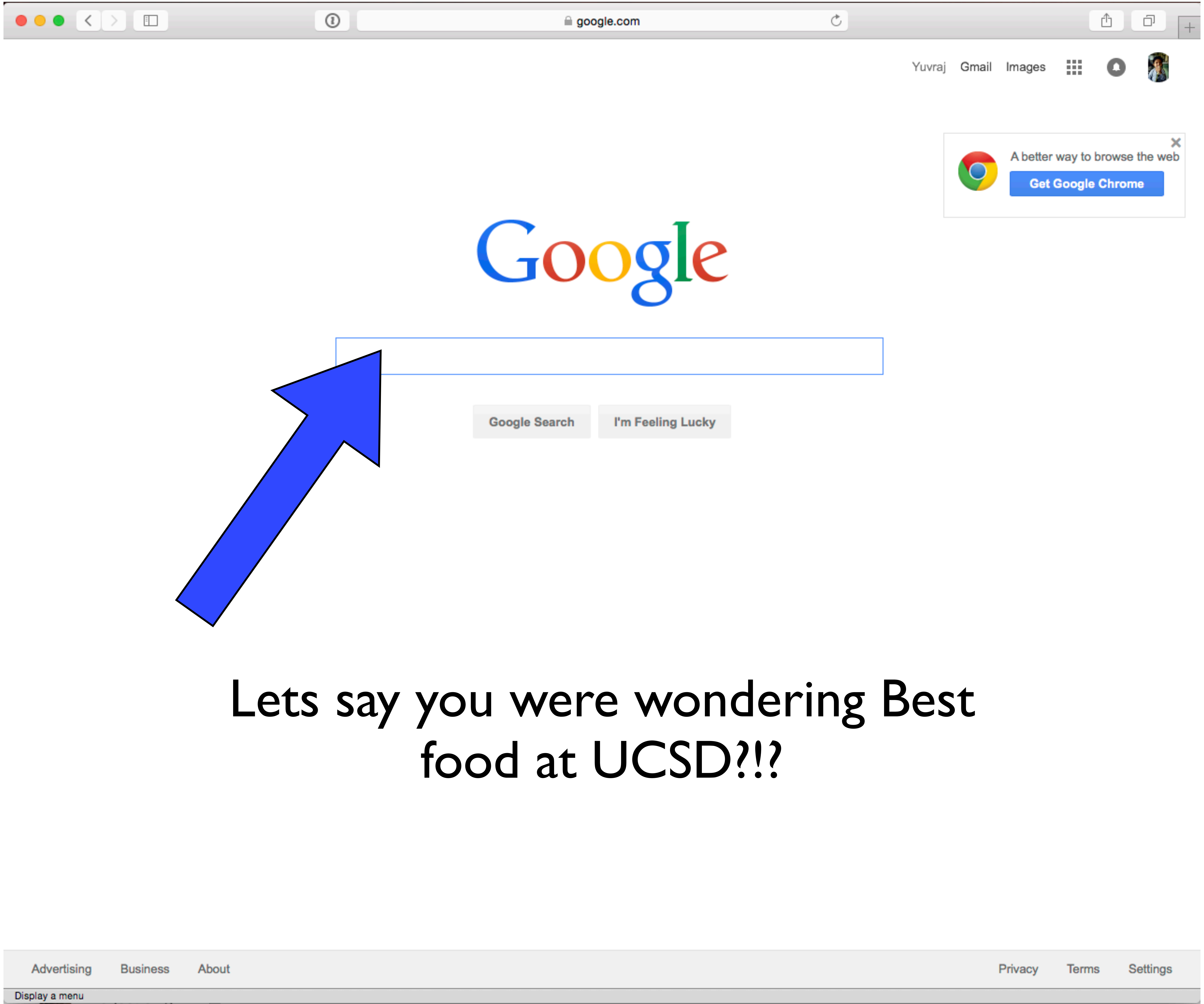
- Cloud → **Network** → Distributed storage → Partition and replication (HDFS) → Distributed computation

Data Processing and Programming model (5 weeks)

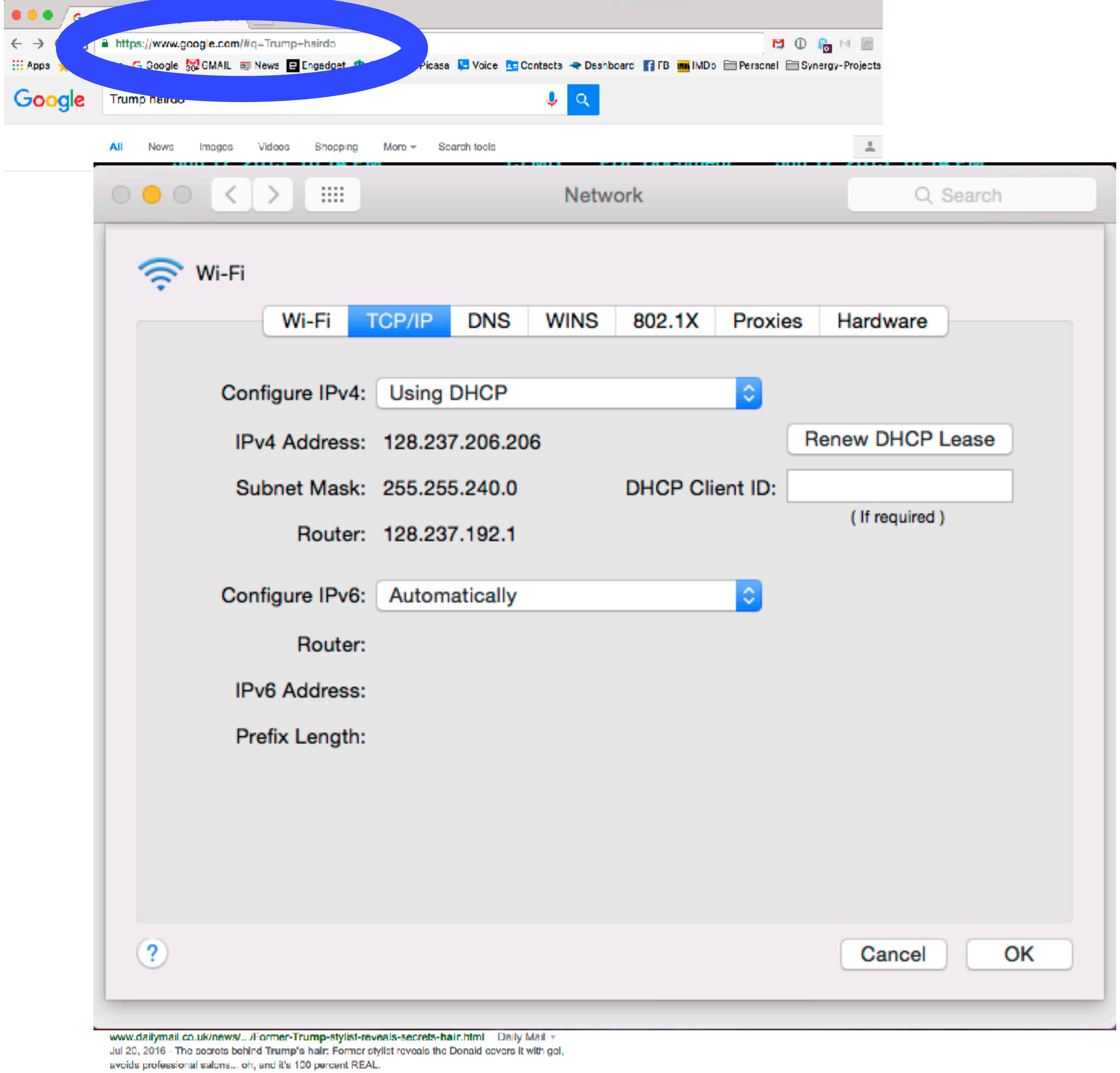
- Data Models evolution → Data encoding evolution → → IO & Unix Pipes → Batch processing (MapReduce) → Stream processing (Spark)

Today's topic

- An example of distributed system
- Network



Lets say you were wondering Best food at UCSD?!?


https://www.google.com/#q=Trump+hairdo
Apps
Google
GMAIL
News
Engadget
Picasa
Voice
Contacts
Dashboard
FB
IMDb
Personal
Synergy-Projects
Google
Trump hairdo
All
News
Images
Videos
Shopping
More
Search tools
Network
Search
Wi-Fi
Wi-Fi
TCP/IP
DNS
WINS
802.1X
Proxies
Hardware
Configure IPv4:
Using DHCP
IPv4 Address:
128.237.206.206
Renew DHCP Lease
Subnet Mask:
255.255.240.0
DHCP Client ID:
(If required)
Router:
128.237.192.1
Configure IPv6:
Automatically
Router:
IPv6 Address:
Prefix Length:
?
Cancel
OK
www.dailymail.co.uk/news/.../Former-Trump-stylist-reveals-secrets-hair.html Daily Mail
Jul 20, 2016 - The secrets behind Trump's hair: Former stylist reveals the Donald covers it with gel, avoids professional salons... oh, and it's 100 percent REAL.

er IP...

hosts.txt

www.google.com
66.233.169.103
www.cmu.edu 128.2.185.33
www.cs.cmu.edu 128.2.56.91
www.areyouawake.com
66.93.60.192
...

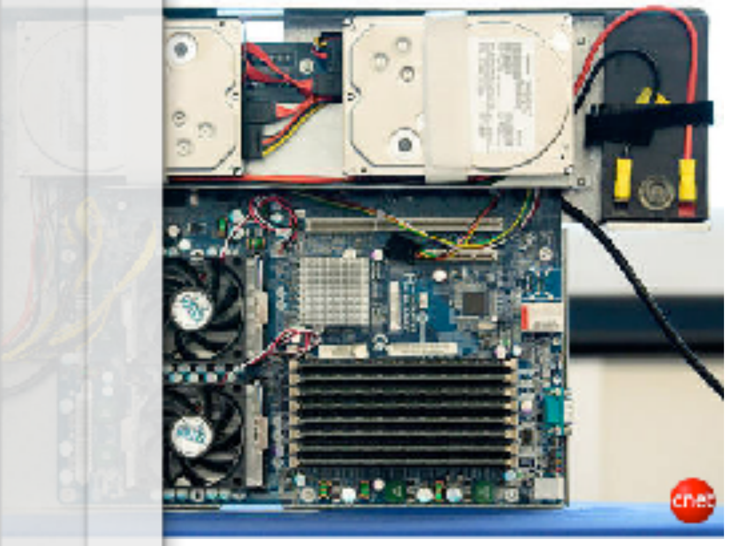
X | Proxies | Ethernet

Domains:

s.cmu.edu
u.edu
apository.net

IPv4 or IPv6 addresses

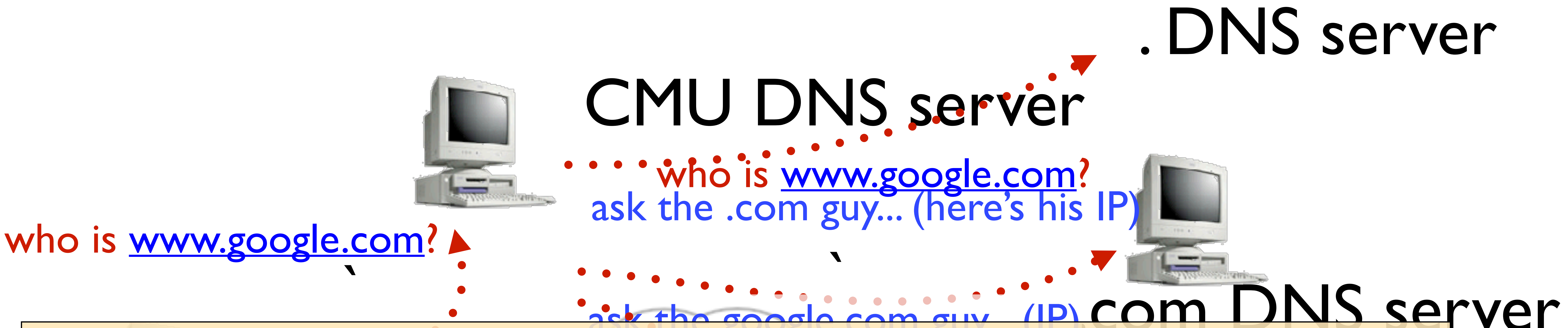
Cancel OK



From: 128.237.206.206
To: 66.233.169.103

<packet contents>

Domain Name System

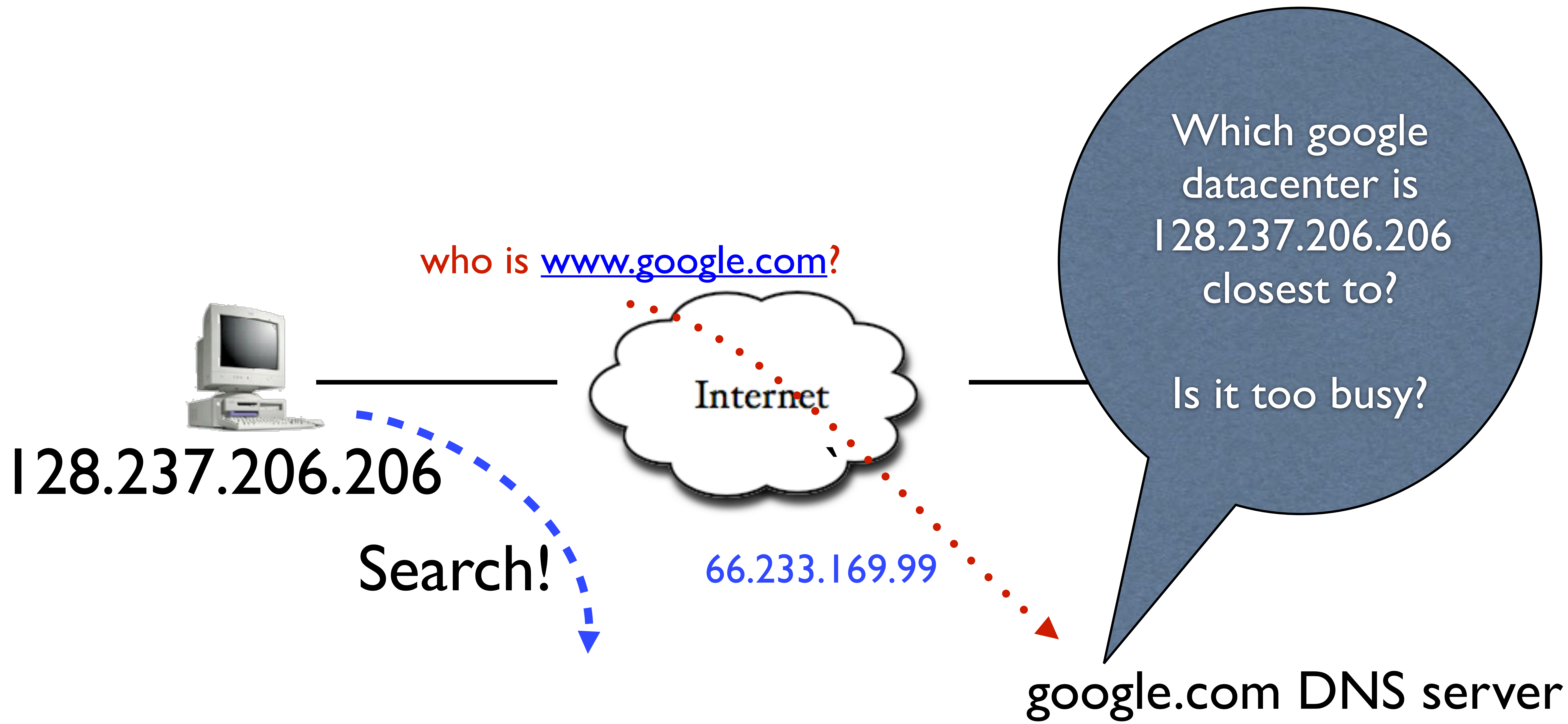


Decentralized - admins update own domains without coordinating with other domains

Scalable - used for hundreds of millions of domains

Robust - handles load and failures well

But there's more...

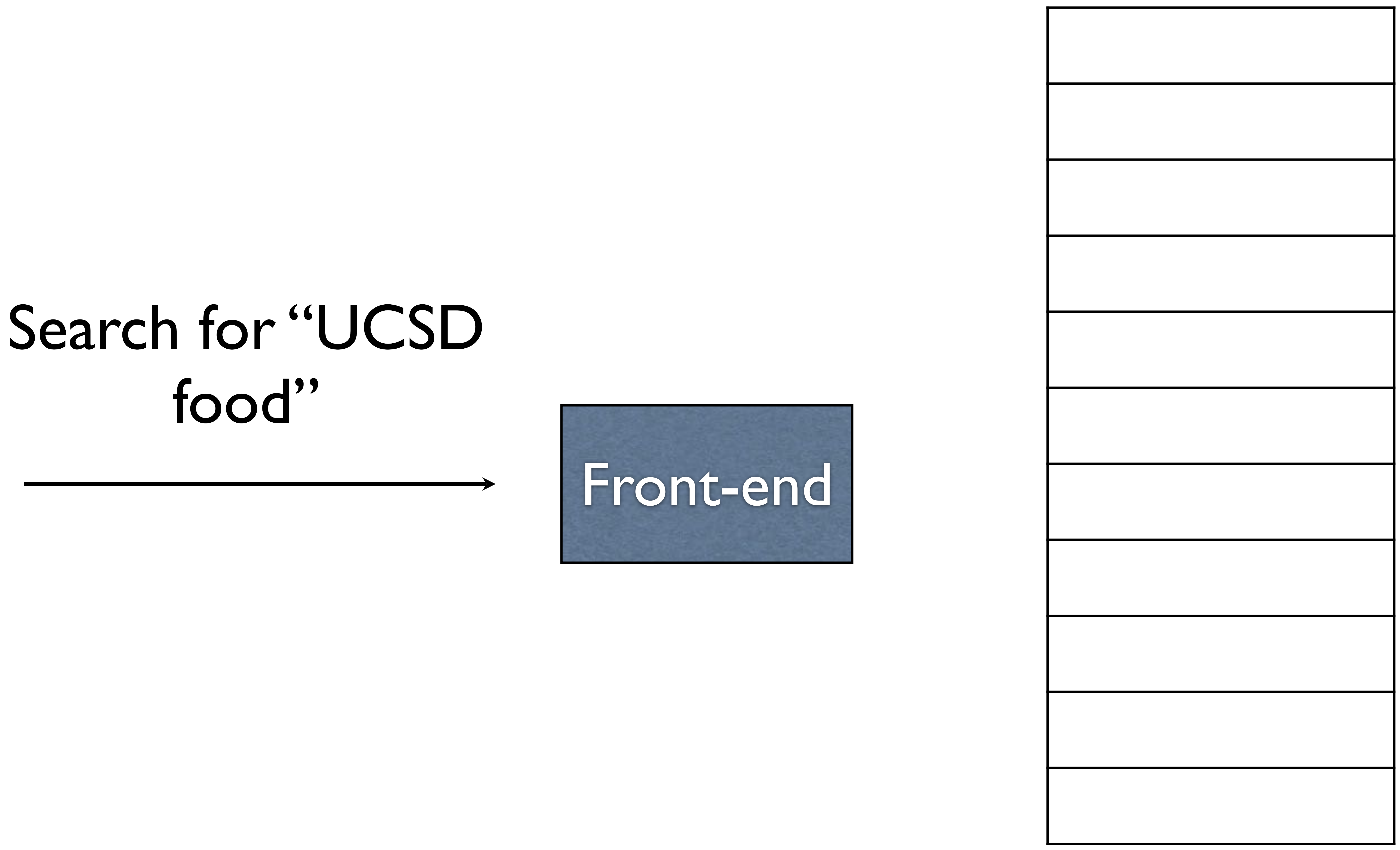


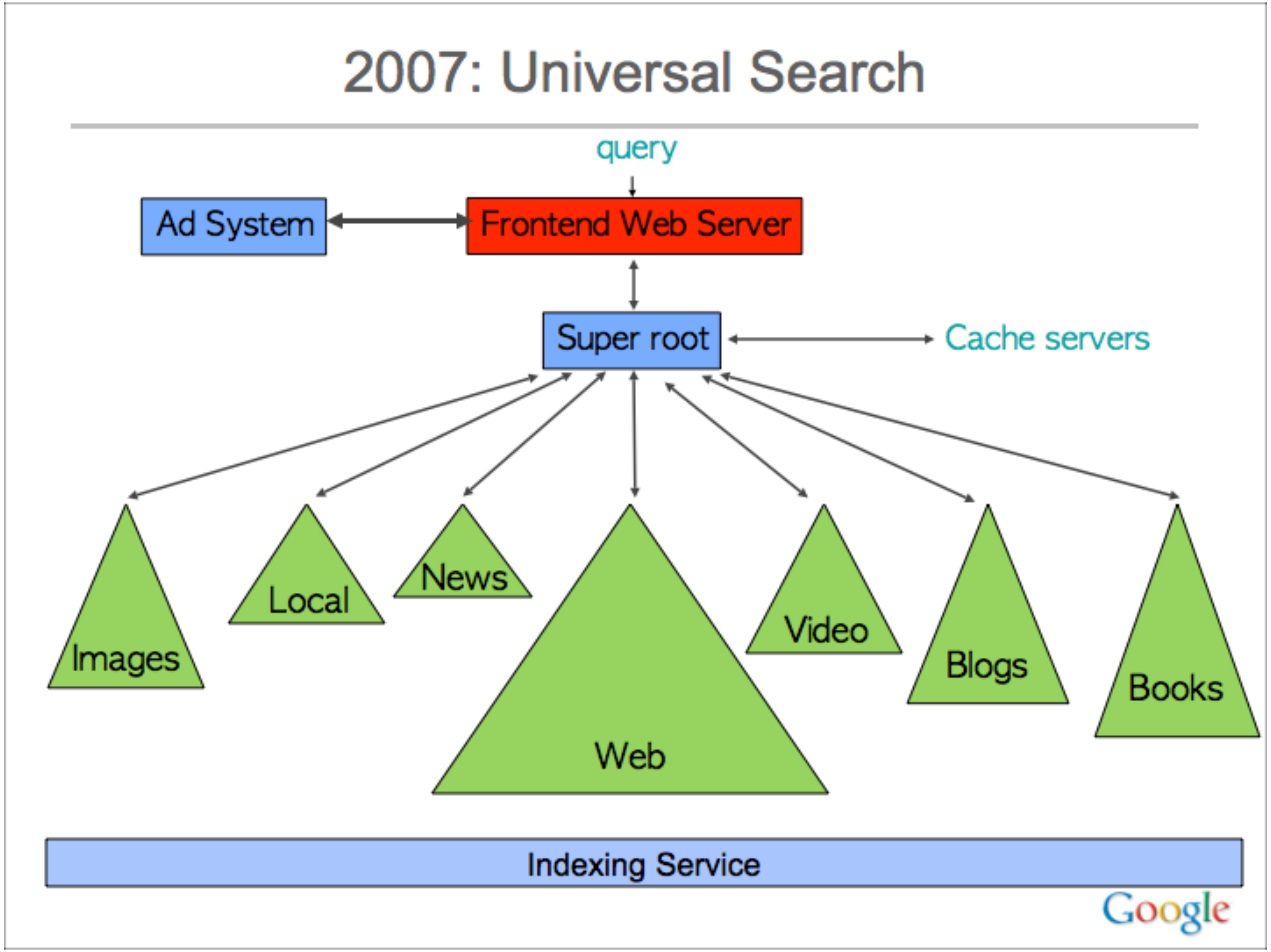
A Google Datacenter

How big? Perhaps one million+ machines

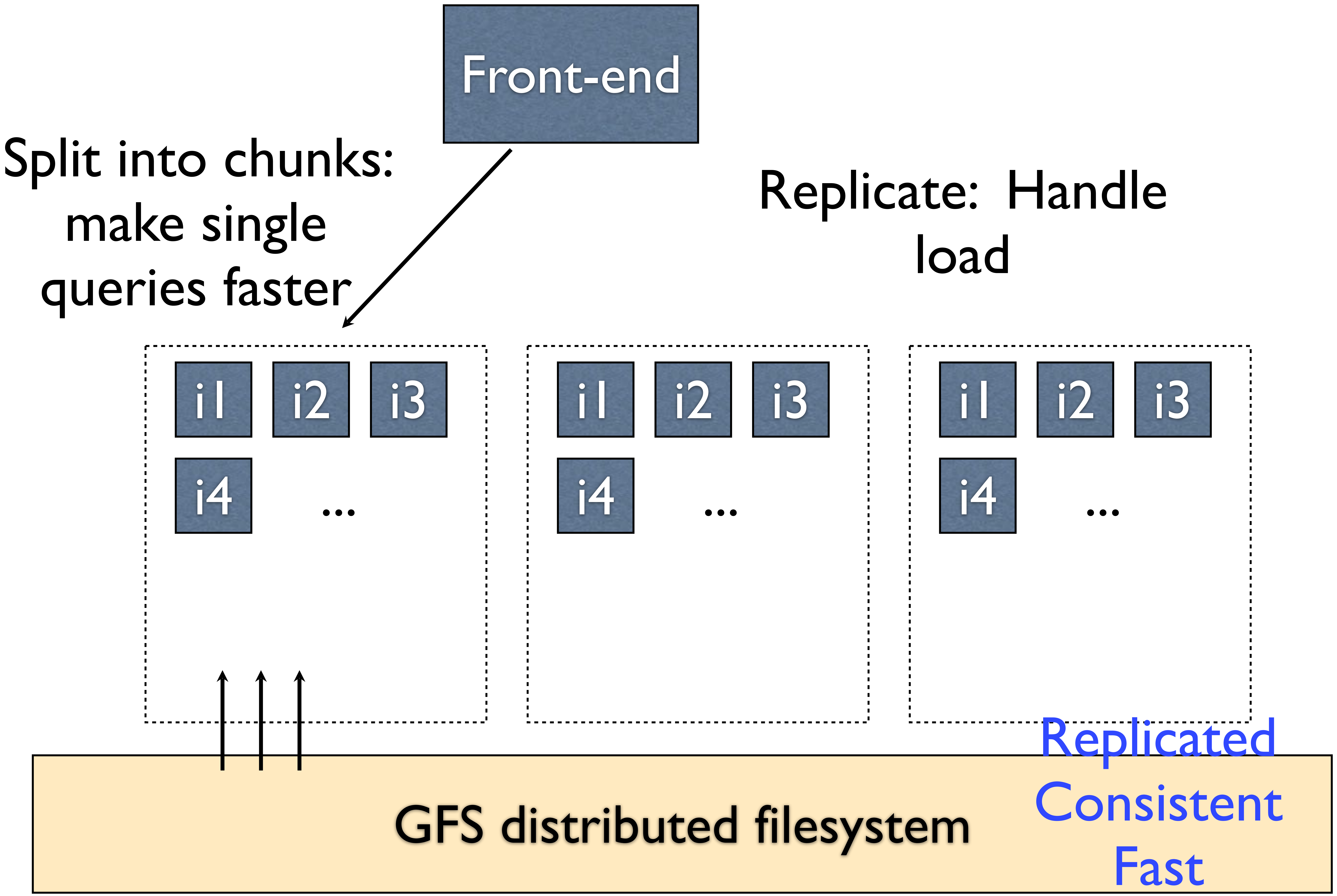
but it's not that bad...

usually don't use more than **20,000** machines to accomplish a single task. [2009, probably out of date]


Search for "UCSD food"
Front-end



slide from Jeff Dean, Google


Front-end
Split into chunks: make single queries faster
Replicate: Handle load
i1
i2
i3
i4
...
i1
i2
i3
i4
...
i1
i2
i3
i4
...
GFS distributed filesystem
Replicated
Consistent
Fast

How do you index the web?

1. Get a copy of the web.
2. Build an index.
3. Profit.

There are over 1 trillion unique URLs

Billions of unique web pages

Hundreds of millions of websites

30?? terabytes of text

=

- *Crawling* -- download those web pages
- *Indexing* -- harness 10s of thousands of machines to do it
- *"Data-Intensive Computing"*

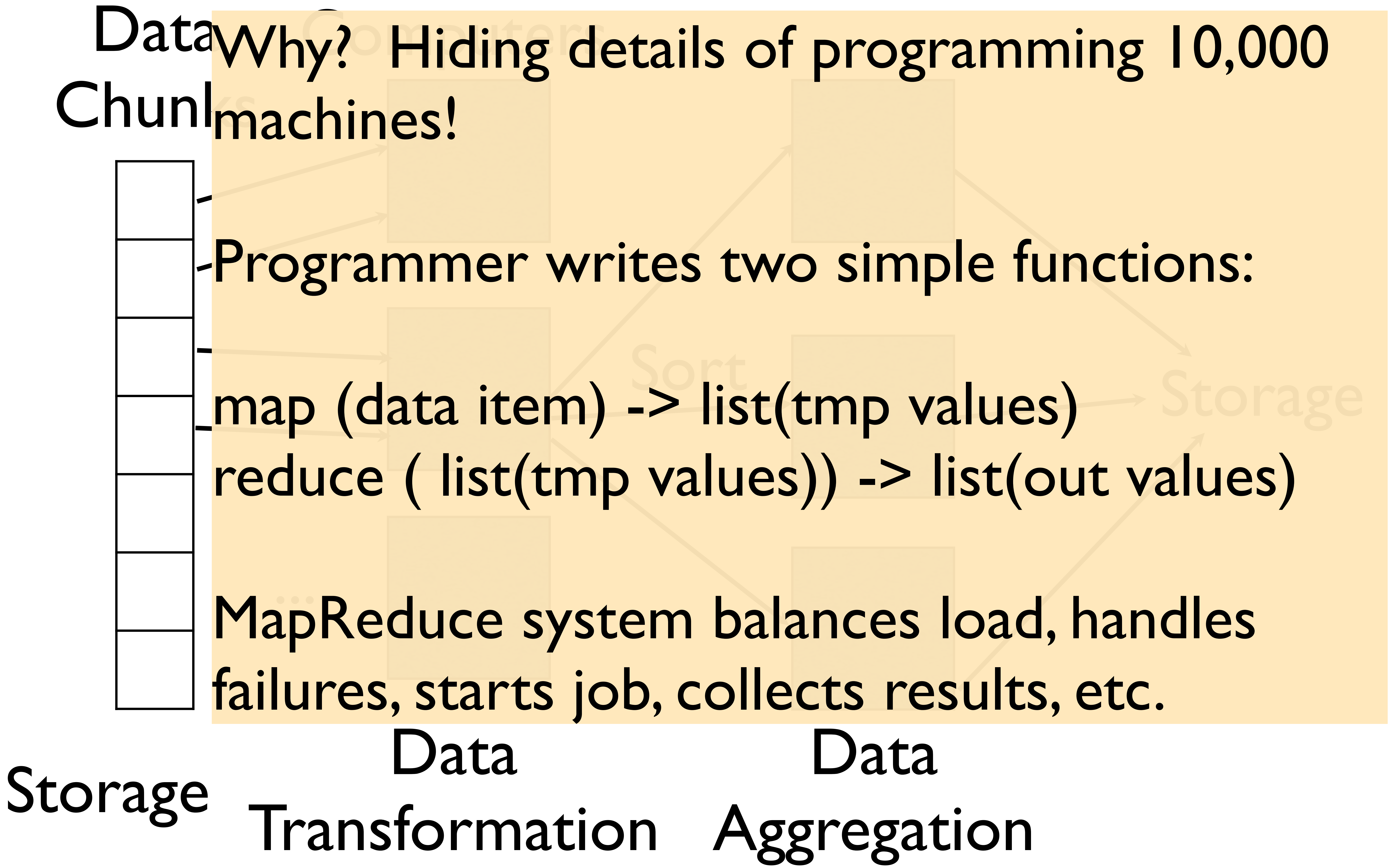
MapReduce / Hadoop

Why? Hiding details of programming 10,000 machines!

Programmer writes two simple functions:

map (data item) -> list(tmp values)
reduce (list(tmp values)) -> list(out values)

MapReduce system balances load, handles failures, starts job, collects results, etc.



All that...

- Hundreds of DNS servers
- Protocols on protocols on protocols
- Distributed network of Internet routers to get packets around the globe
- Hundreds of thousands of servers

Today’s topic

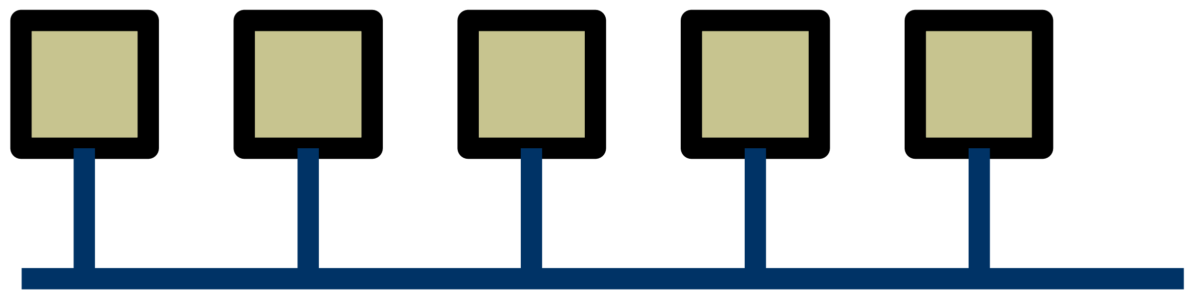
- An example of distributed system
- Network
  - Network links and LANs
  - Layering and protocols
  - Internet design
  - Transport protocols

Basic Building Block: Links

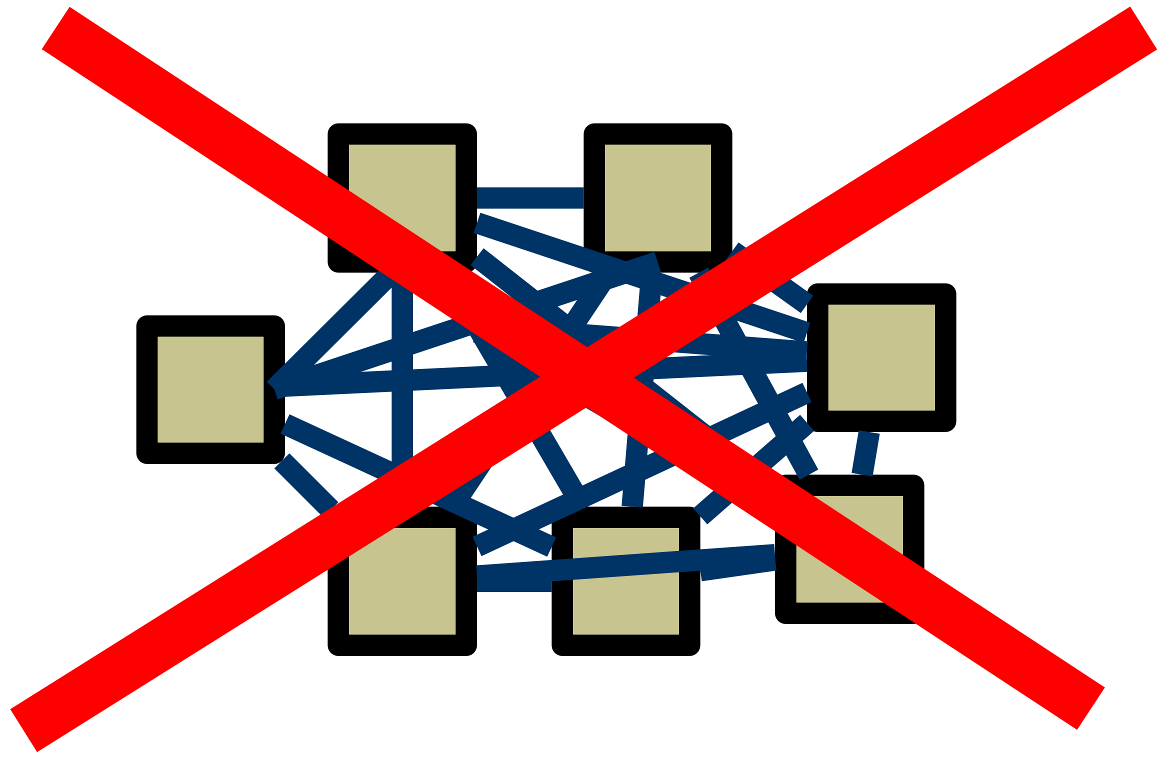
- Electrical questions
  - Voltage, frequency, …
  - Wired or wireless?
- Link-layer issues: How to send data?
  - When to talk – can either side talk at once?
  - What to say – low-level format?

Basic Building Block: Links

- … But what if we want more hosts?



One wire

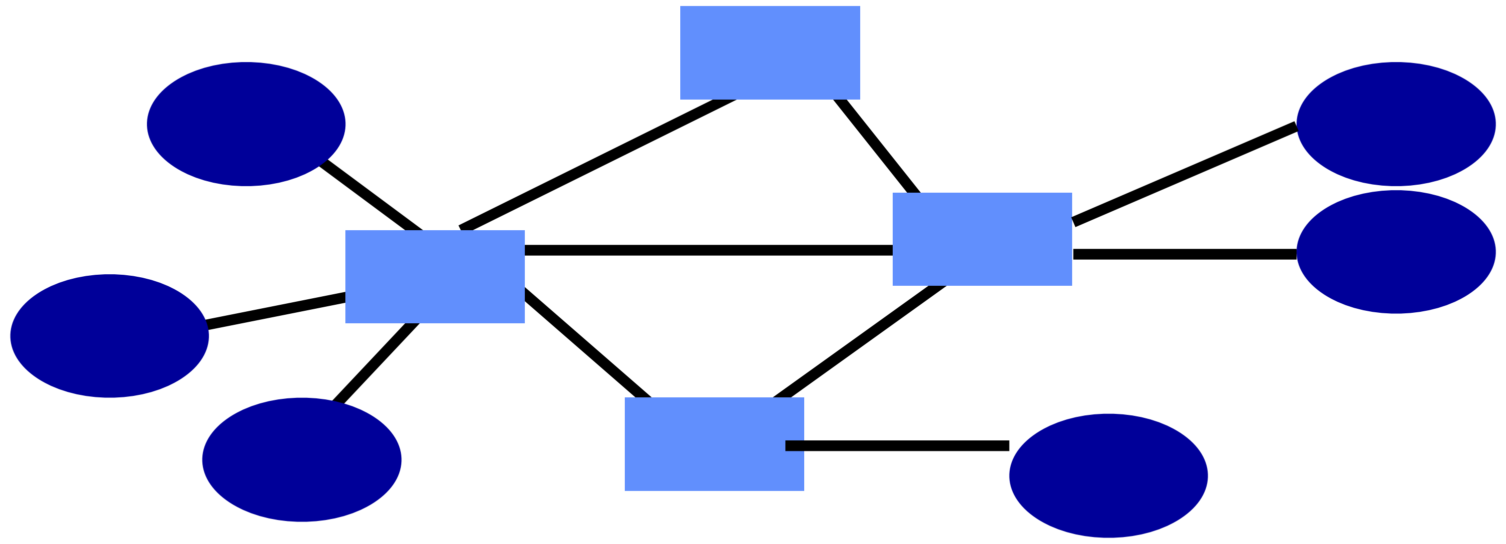


Wires for everybody!

- Scalability?!

Multiplexing

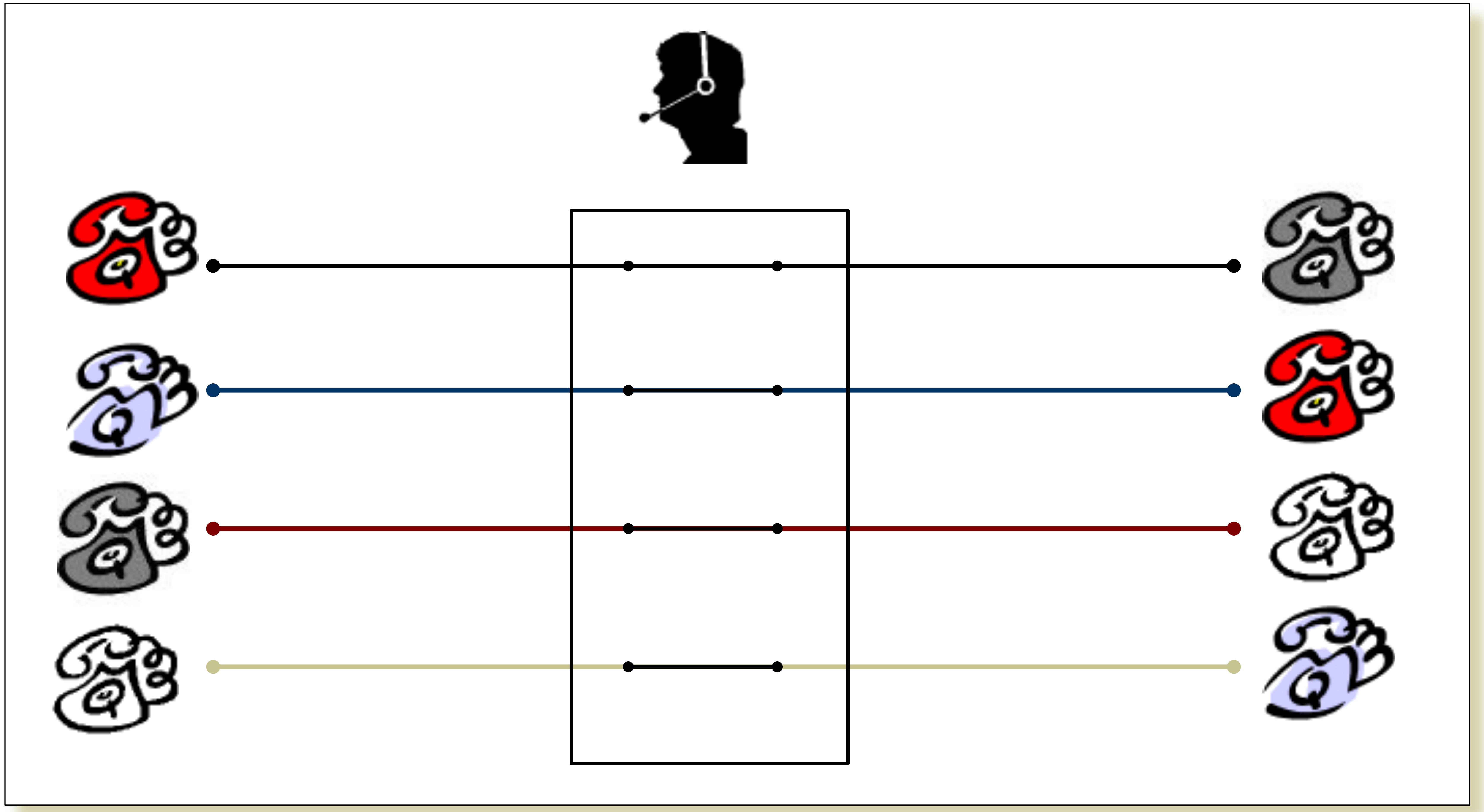
- Need to share network resources



- How? Switched network
  - Party "A" gets resources sometimes
  - Party "B" gets them sometimes
- Interior nodes act as "Switches"

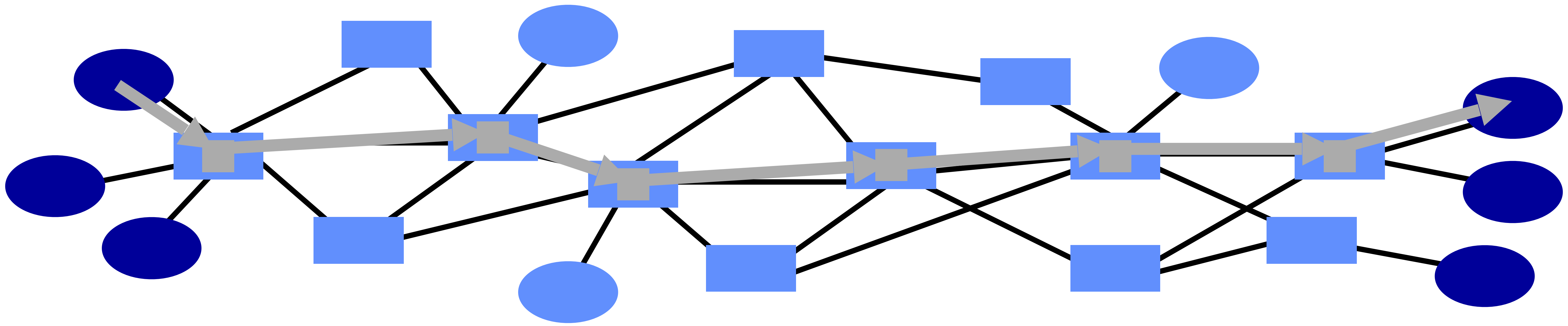
- What mechanisms to share resources?

In the Old Days...Circuit Switching

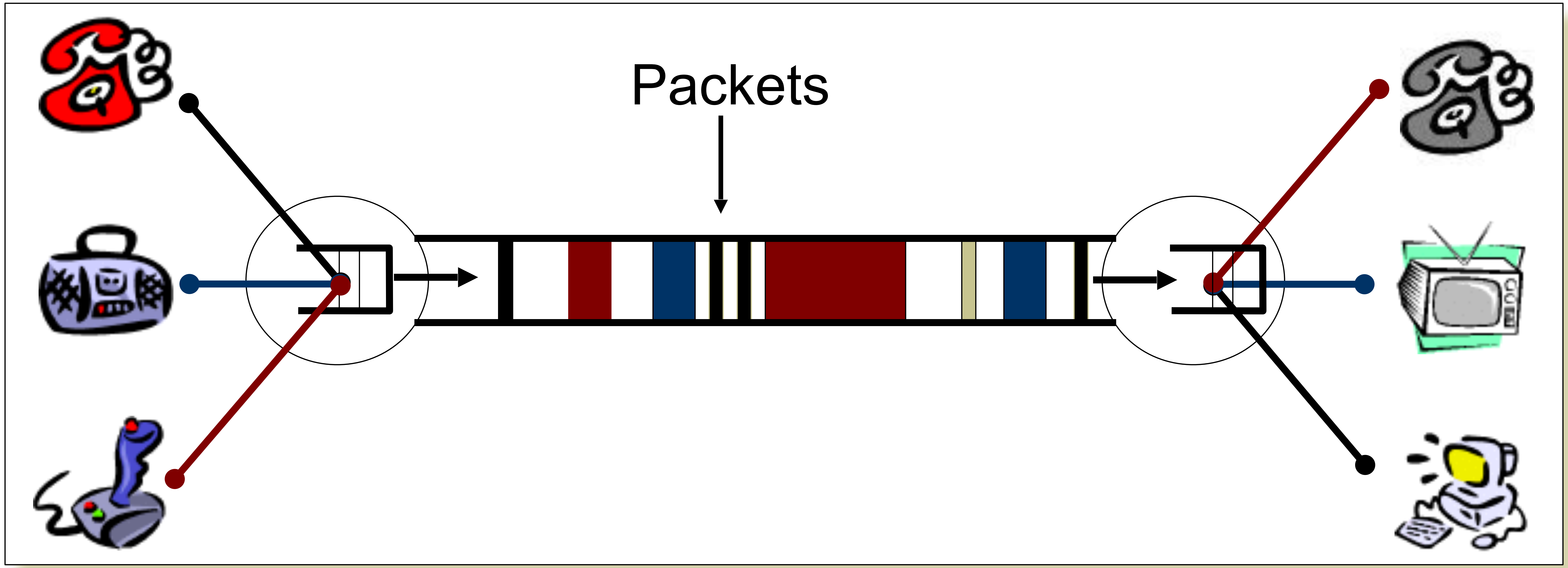


Packet Switching

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



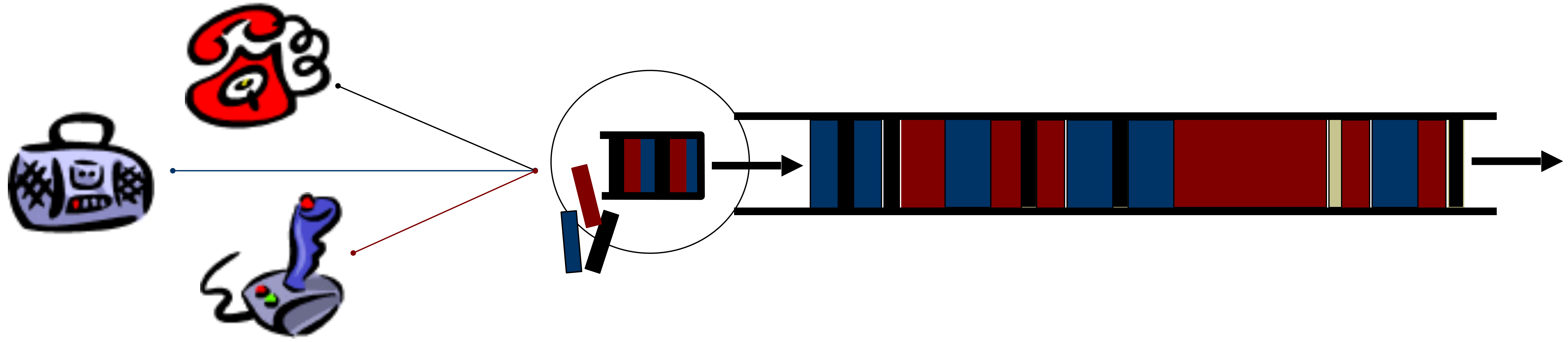
Packet Switching – Statistical Multiplexing



- Switches arbitrate between inputs
- Can send from *any* input that's ready
  - Links never idle when traffic to send
  - (Efficiency!)

What if Network is Overloaded?

Problem: Network Overload



Solution: Buffering and Congestion Control

- Short bursts: buffer
- What if buffer overflows?
  - Packets dropped
  - Sender adjusts rate until load = resources → "congestion control"

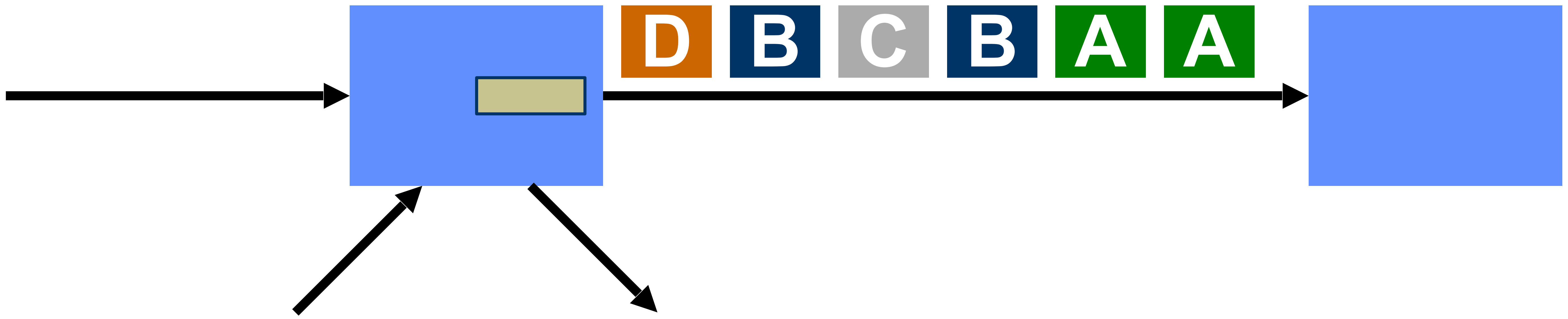
Model of a communication channel

- Latency - how long does it take for the first bit to reach destination
- Capacity - how many bits/sec can we push through? (often termed "bandwidth")
- Jitter - how much variation in latency?
- Loss / Reliability - can the channel drop packets?
- Reordering

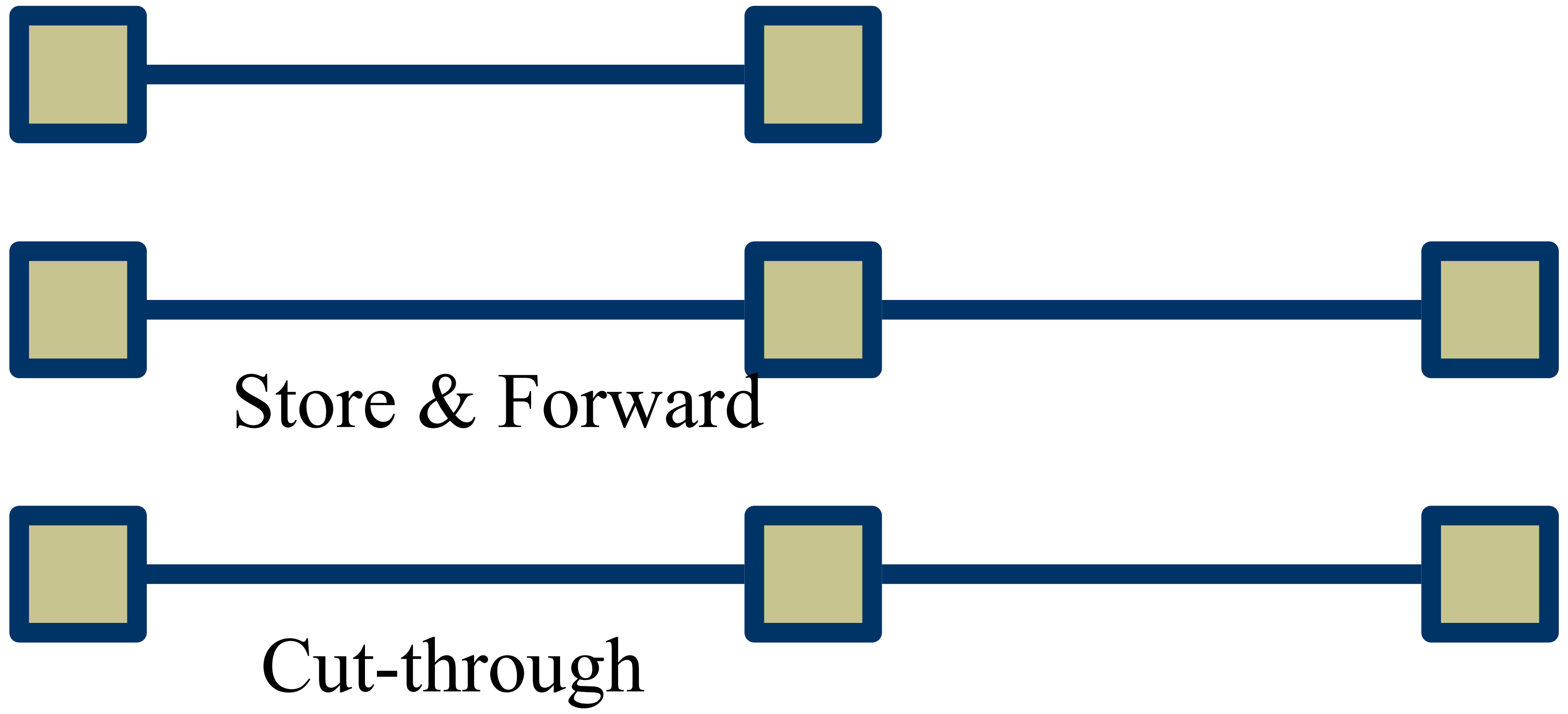
Packet Delay

- Sum of a number of different delay components:

- Propagation delay on each link.
  - Proportional to the length of the link
- Transmission delay on each link.
  - Proportional to the packet size and 1/link speed
- Processing delay on each router.
  - Depends on the speed of the router
- Queuing delay on each router.
  - Depends on the traffic load and queue size



Packet Delay



Prop + xmit

2*(Prop + xmit)

2*prop + xmit

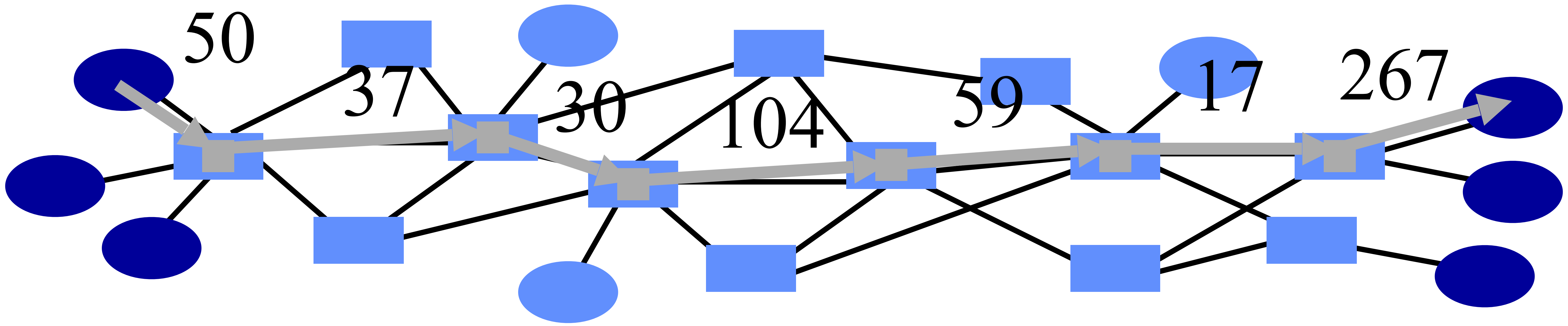
When does cut-through matter?

Next: Routers have finite speed (processing delay)

Routers may buffer packets (queueing delay)

Sustained Throughput

- When streaming packets, the network works like a pipeline.
  - All links forward different packets in parallel
- Throughput is determined by the slowest stage.
  - Called the bottleneck link
- Does not really matter why the link is slow.
  - Low link bandwidth
  - Many users sharing the link bandwidth



Some simple calculations (mbps/kbps)

- Cross country latency
  - Distance/speed = 5 * 10^6m / 2x10^8m/s = 25 * 10^-3 s = 25ms
  - 50ms RTT
- Link speed (capacity) 100Mbps
- Packet size = 1250 bytes = 10 kbits
  - Packet size on networks usually = 1500 bytes across wide area or 9000 bytes in local area
- 1 packet takes
  - 10k/100M = .1 ms to transmit
  - 25ms to reach there
  - ACKs are small → so 0ms to transmit
  - 25ms to get back
- Effective bandwidth = 10kbits/50.1ms = 200kbits/sec ☹

https://www.quora.com/Which-is-faster-the-speed-of-electric-current-through-a-copper-wire-or-the-speed-of-light

Some Examples

- How long does it take to send a 100 Kbit file?
  - Assume a perfect world

Throughput / Latency	100 Kbit/s	1 Mbit/s	100 Mbit/s
500 μsec	1.0005	0.1005	0.0015
10 msec	1.01	0.11	<u>0.011</u>
100 msec	1.1	0.2	<u>0.101</u>

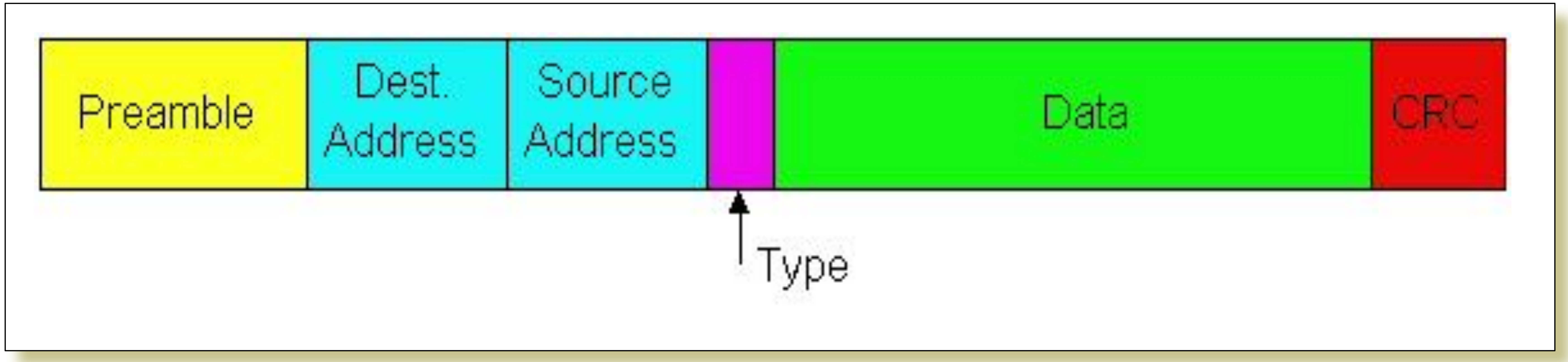
Some Examples

- How long does it take to send a 10 Kbit file?
  - Assume a perfect world

Latency \ Throughput	100 Kbit/s	1 Mbit/s	100 Mbit/s
500 μsec	0.1005	0.0105	0.0006
10 msec	0.11	0.02	0.0101
100 msec	0.2	0.11	0.1001

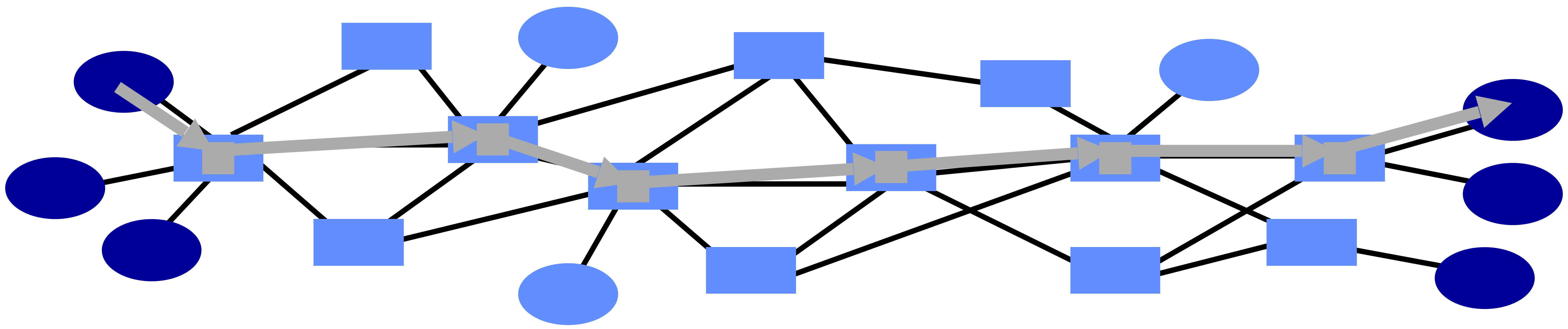
Example: Ethernet Packet

- Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame



Packet Switching

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

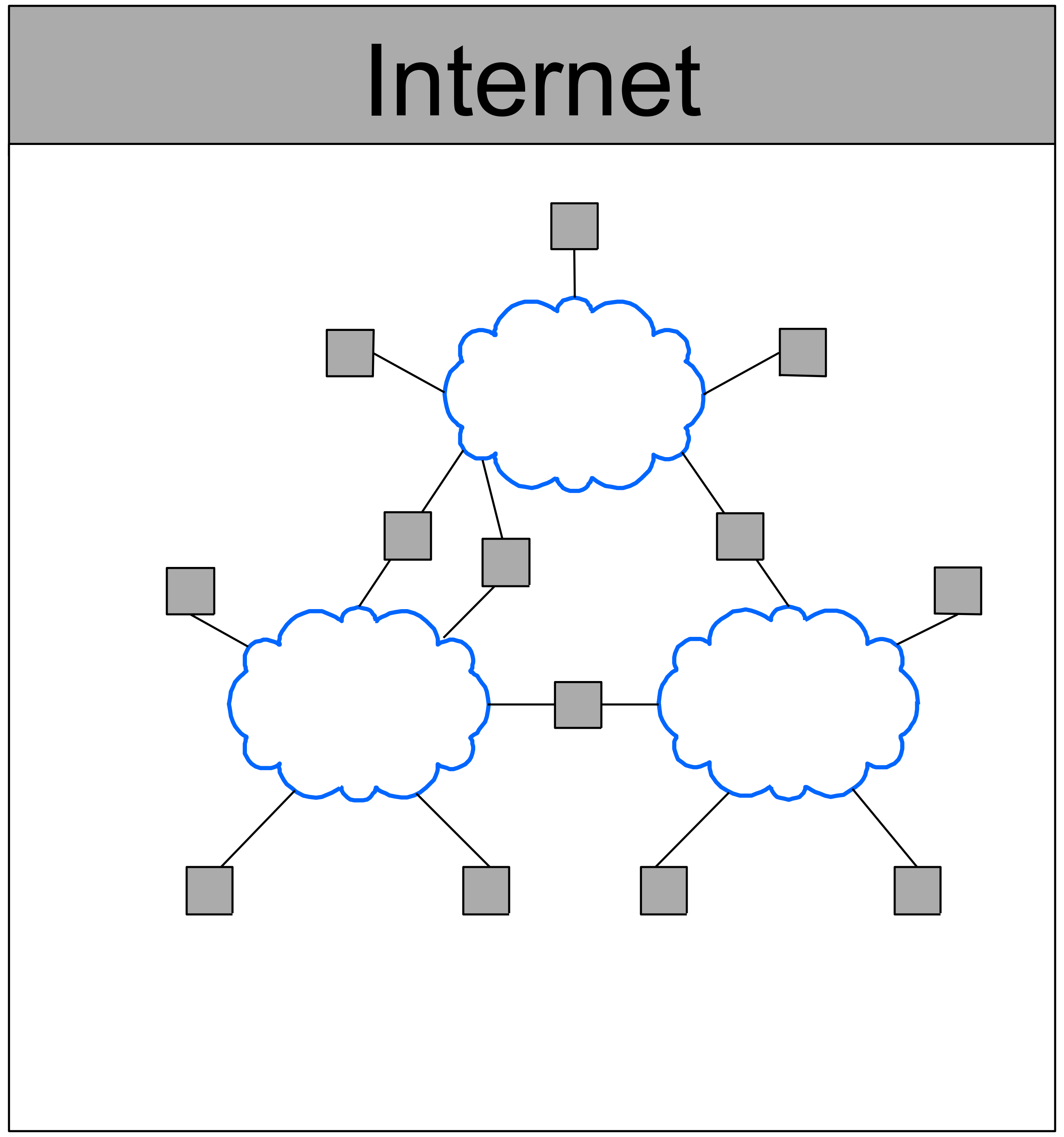


Today's topic

- Network links and LANs
- Layering and protocols
- Internet design
- Transport protocols

Internet

- An inter-net: a network of networks.
  - Networks are connected using routers that support communication in a hierarchical fashion
  - Often need other special devices at the boundaries for security, accounting, ..
- The Internet: the interconnected set of networks of the Internet Service Providers (ISPs)
  - About 17,000 different networks make up the Internet

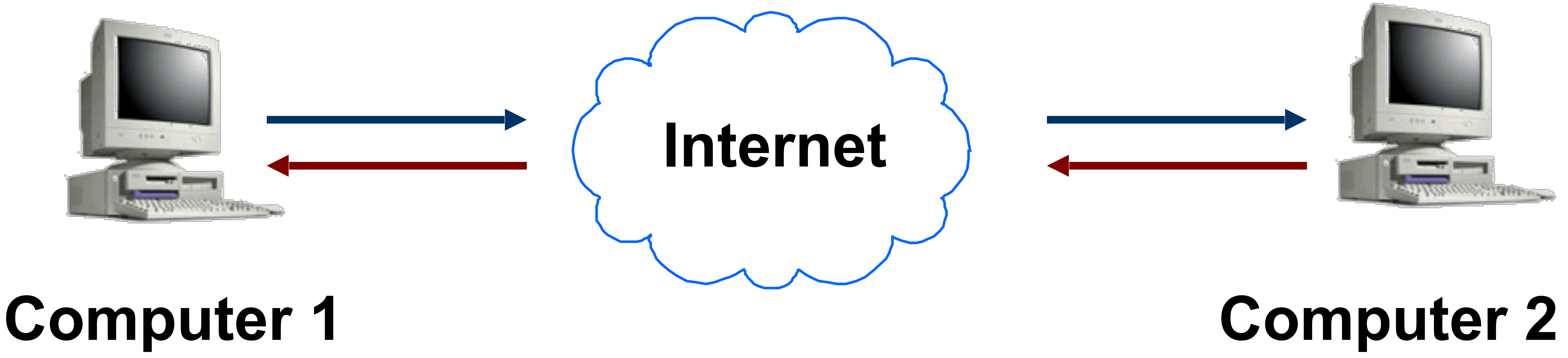


Challenges of an internet

- Heterogeneity
  - Address formats
  - Performance – bandwidth/latency
  - Packet size
  - Loss rate/pattern/handling
  - Routing
  - Diverse network technologies → satellite links, cellular links, carrier pigeons
  - In-order delivery

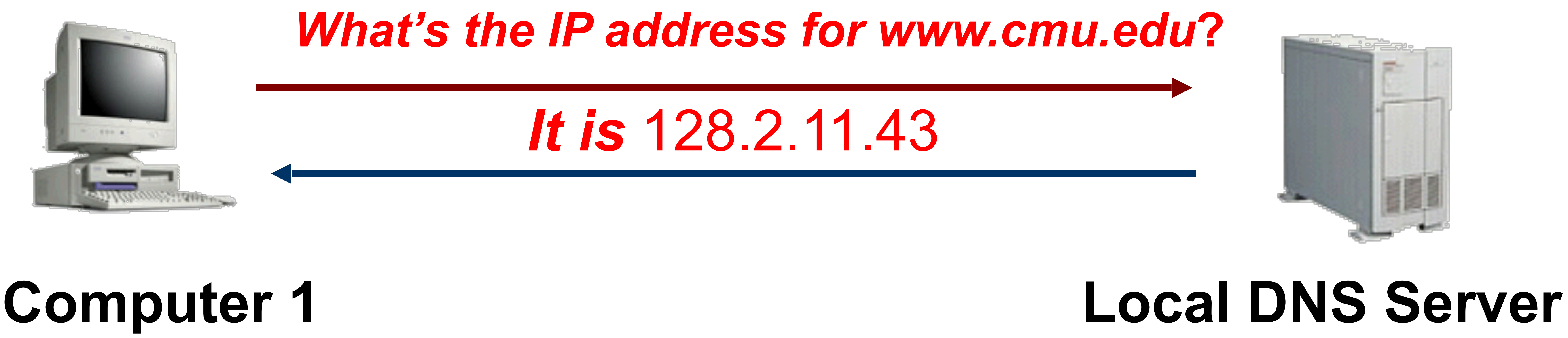
- Need a “standard” that everyone can use → IP

How To Find Nodes?



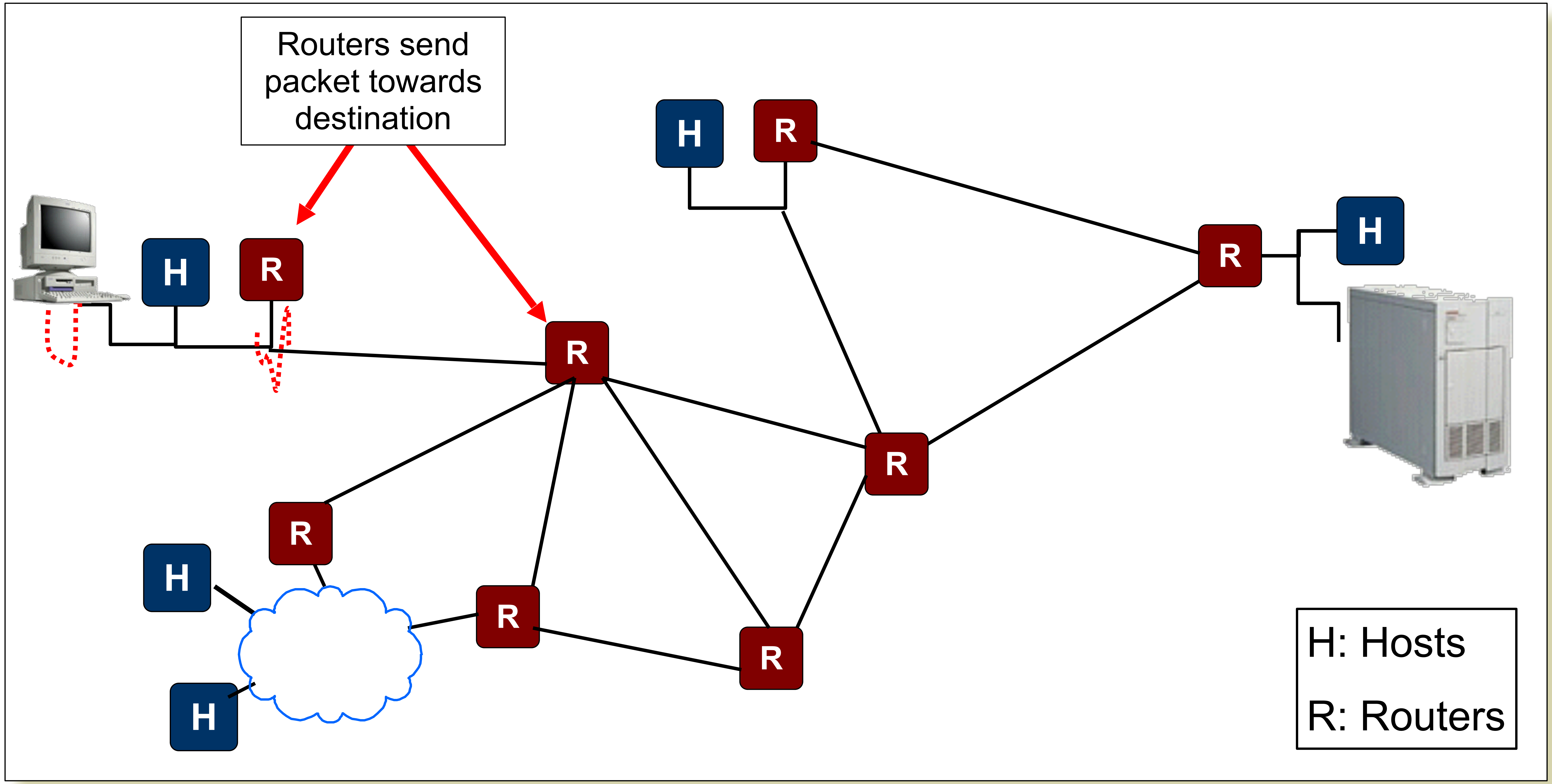
Need naming and routing

Naming



Translates human readable names to logical endpoints

Routing


Routers send packet towards destination
H
R
H
R
H
R
R
H
R
R
H
R
H
H: Hosts
R: Routers

Network Service Model

- What is the *service model* for inter-network*?*
  - Defines what promises that the network gives for any transmission
  - Defines what type of failures to expect
- Ethernet/Internet: *best-effort* – packets can get lost, etc.

Possible Failure models

- Fail-stop:
  - When something goes wrong, the process stops / crashes / etc.
- Fail-slow or fail-stutter:
  - Performance may vary on failures as well
- Byzantine:
  - Anything that can go wrong, will.
  - Including malicious entities taking over your computers and making them do whatever they want.
- These models are useful for proving things;
- The real world typically has a bit of everything.

- Deciding which model to use is important!

Fancier Network Service Models

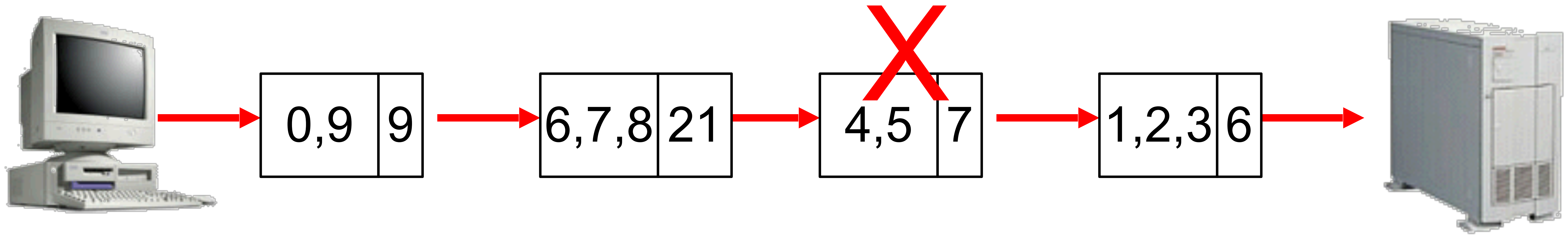
- What if you want more?
  - Performance guarantees (QoS)
  - Reliability
    - Corruption
    - Lost packets
  - Flow and congestion control
  - Fragmentation
  - In-order delivery
  - Etc…

- If network provided this, programmers don't have to implement these features in every application
- But note limitations: this can't turn a byzantine failure model into a fail-stop model…

What if the Data gets Corrupted?

Problem: Data Corruption

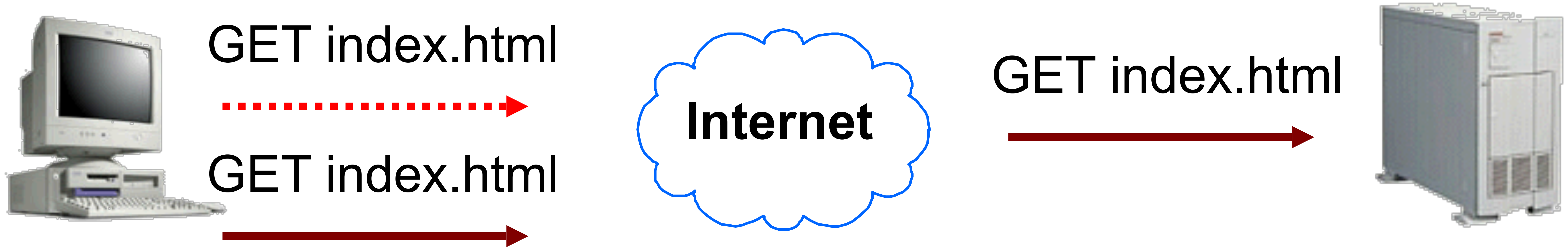
Solution: Add a *checksum*



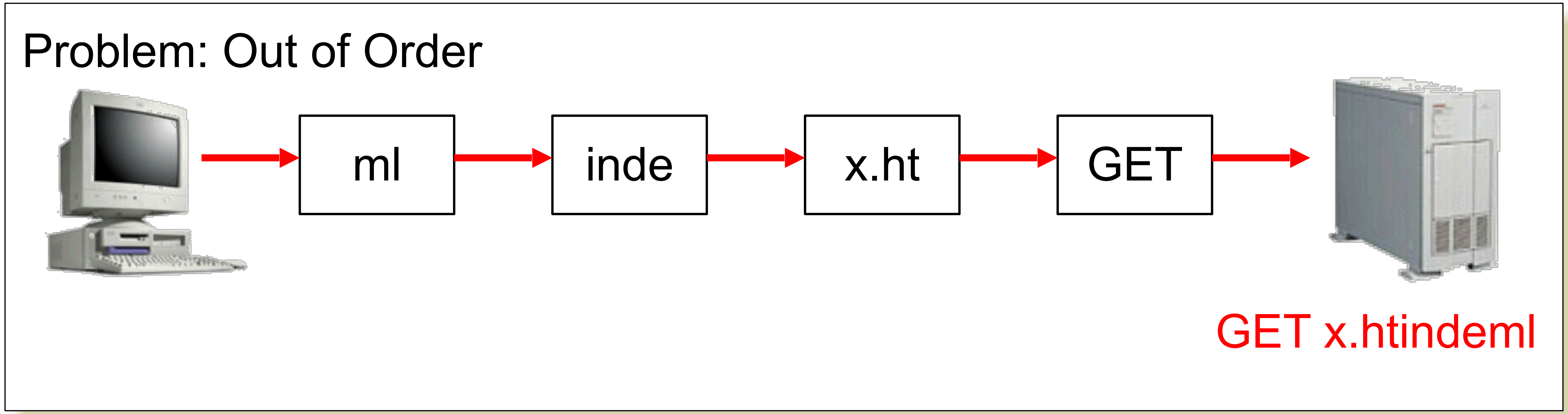
What if the Data gets Lost?

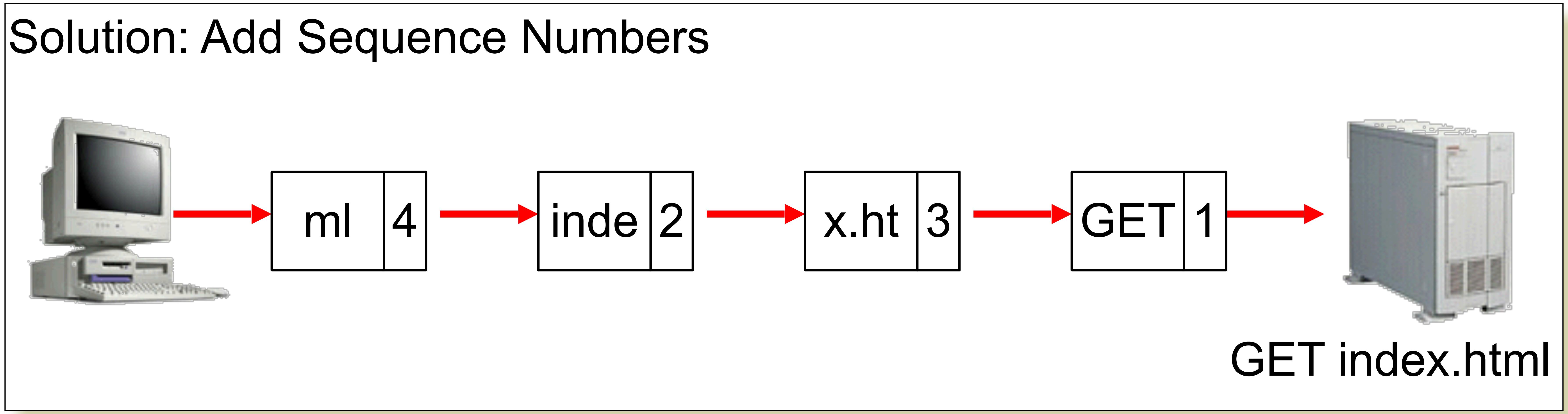
Problem: Lost Data

Solution: Timeout and Retransmit



What if the Data is Out of Order?



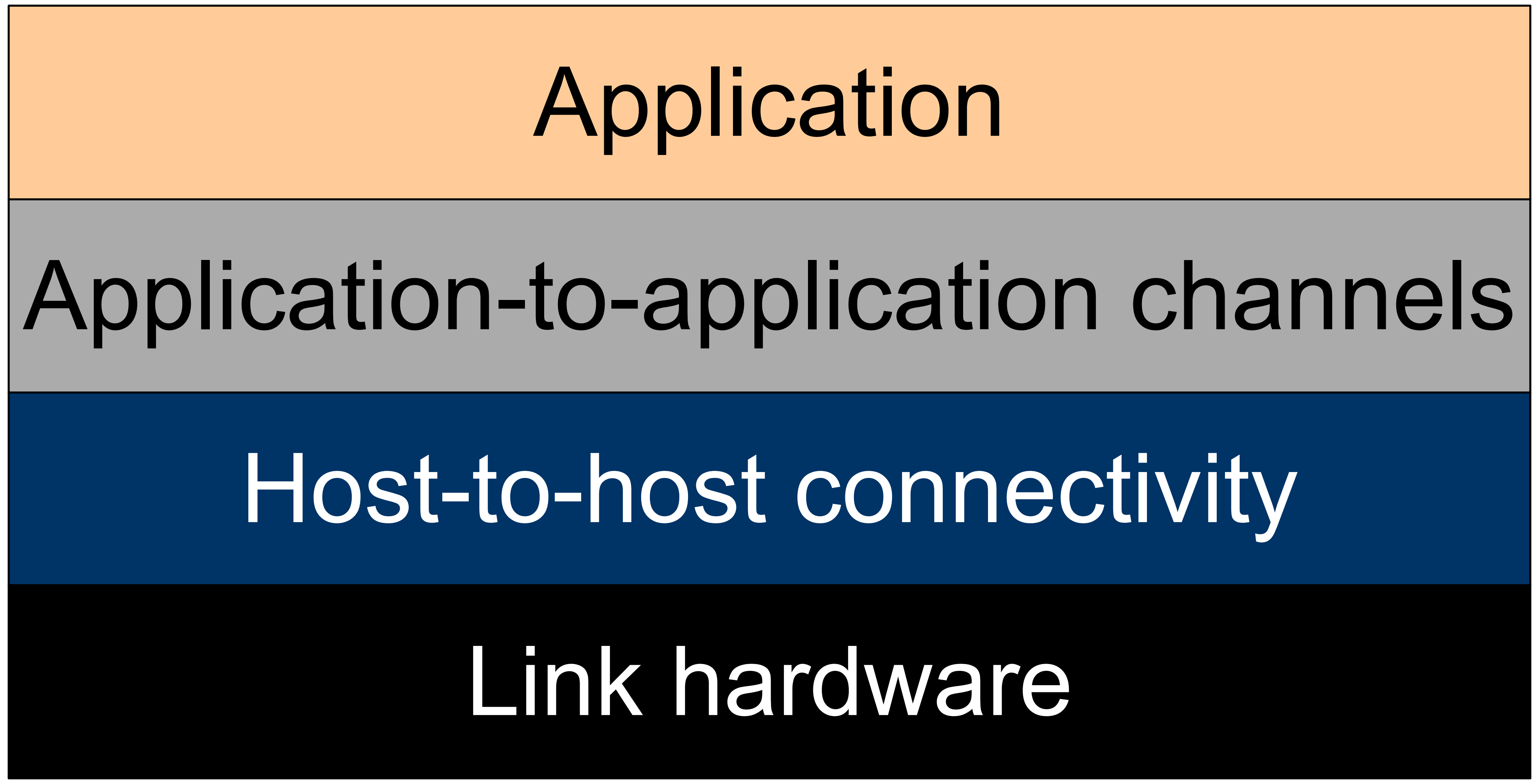


Networks [including end points] Implement Many Functions

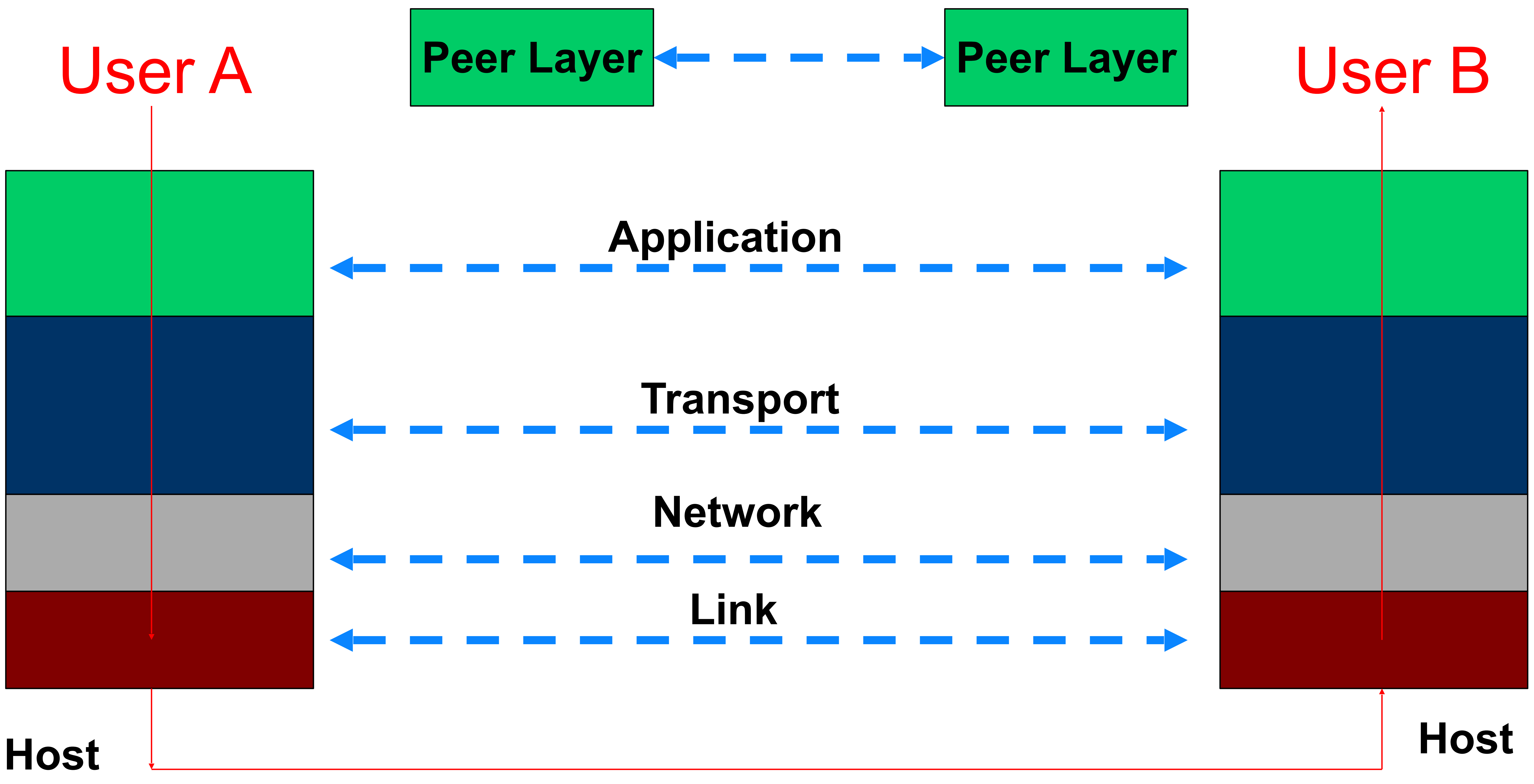
- Link
- Multiplexing
- Routing
- Addressing/naming (locating peers)
- Reliability
- Flow control
- Fragmentation
- Etc….

What is Layering?

- Modular approach to network functionality
- Example:



What is Layering?



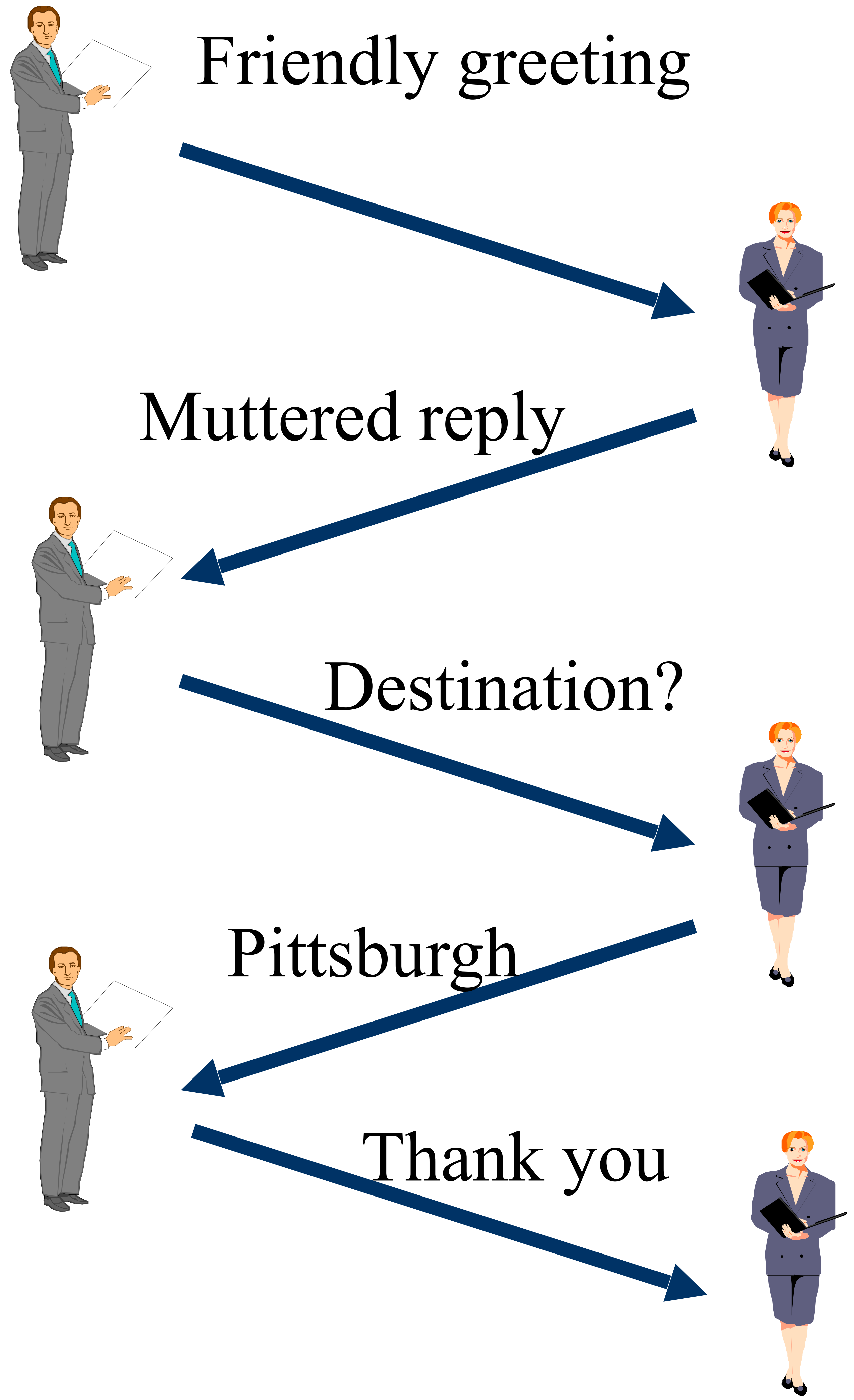
Modular approach to network functionality

Layering Characteristics

- Each layer relies on services from layer below and exports services to layer above
- Interface defines interaction with peer on other hosts
- Hides implementation - layers can change without disturbing other layers (black box)

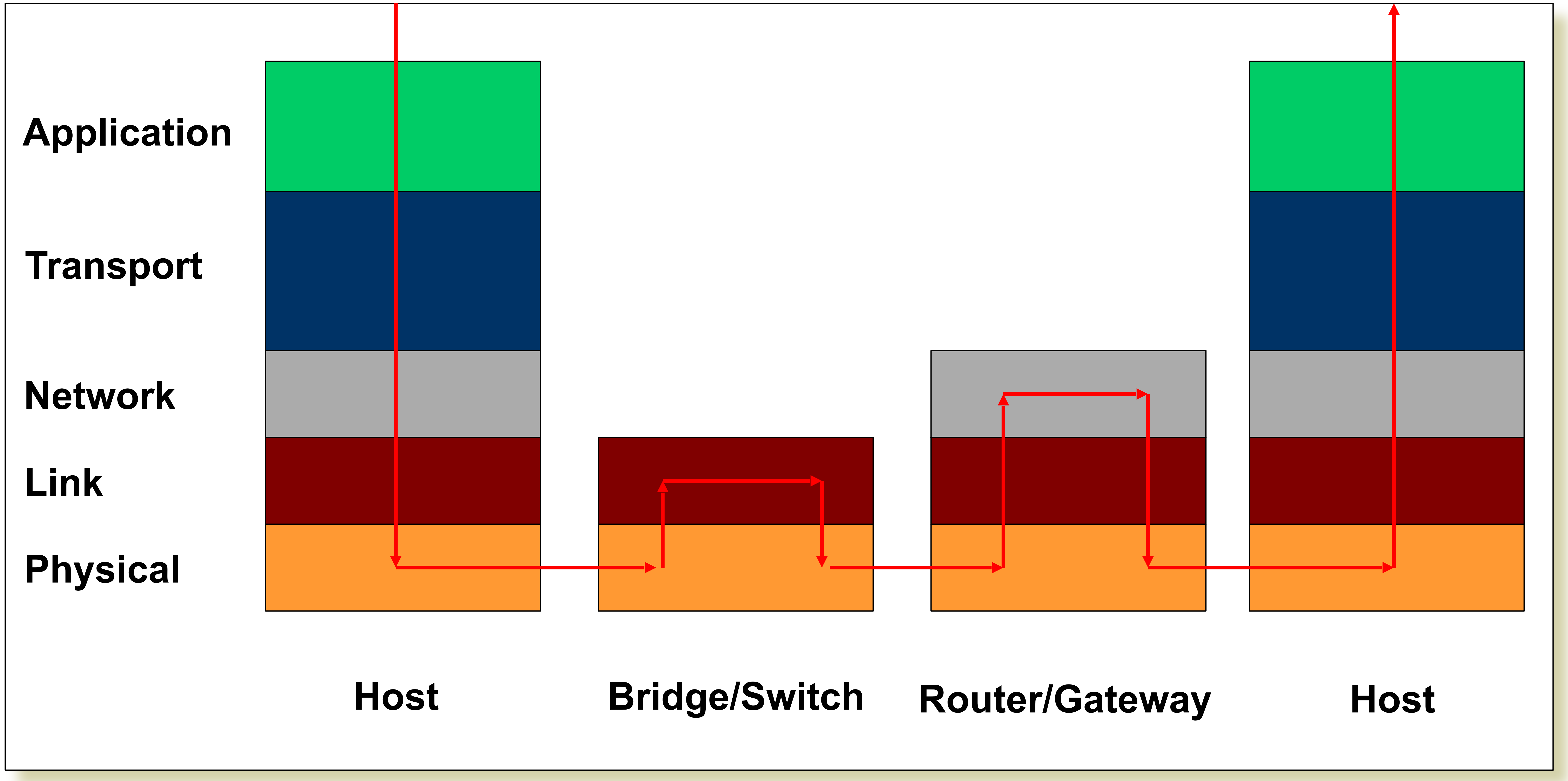
What are Protocols?

- An agreement between parties on how communication should take place
- Module in layered structure
- Protocols define:
  - Interface to higher layers (API)
  - Interface to peer (syntax & semantics)
    - Actions taken on receipt of a messages
    - Format and order of messages
    - Error handling, termination, ordering of requests, etc.
- Example: Buying airline ticket

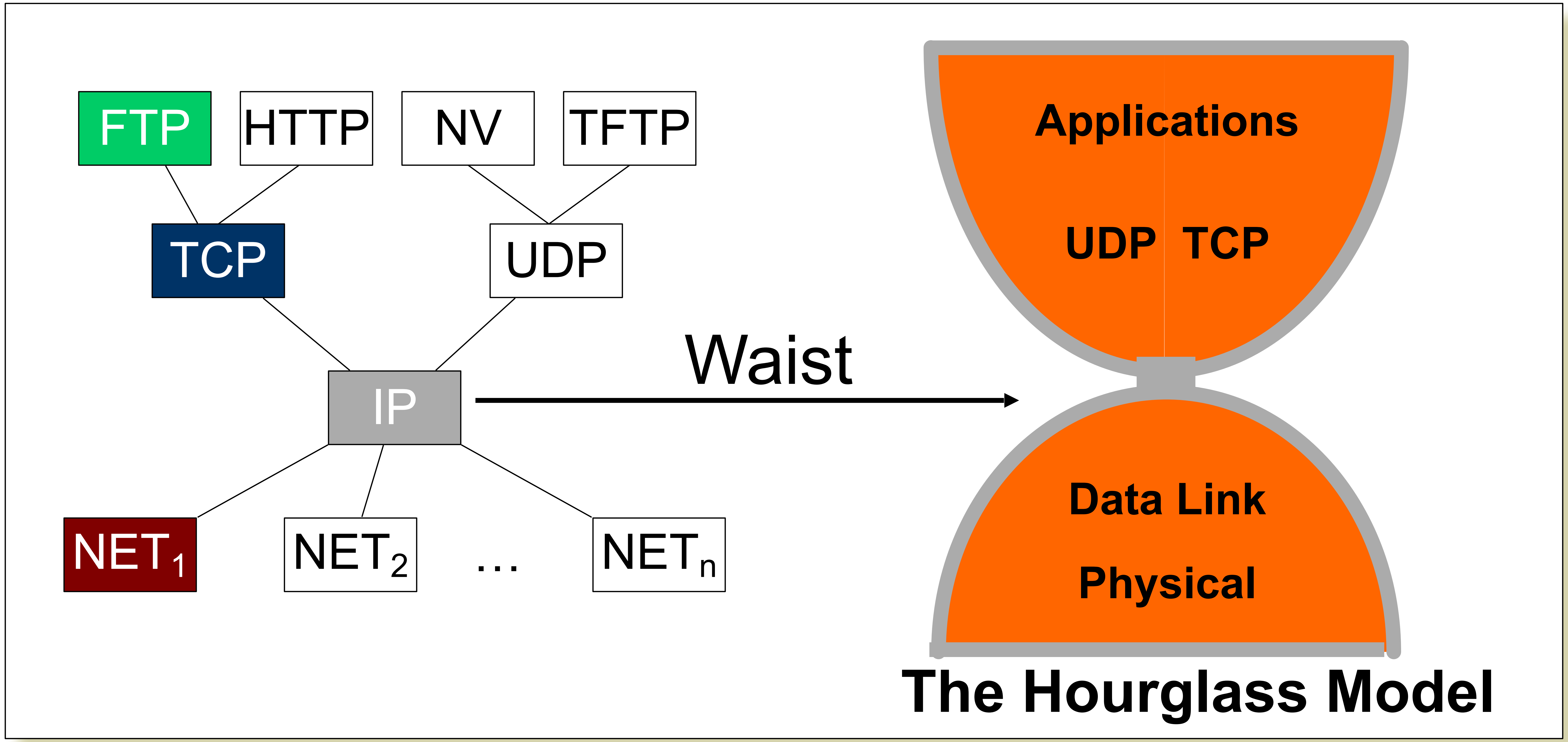


IP Layering

- Relatively simple

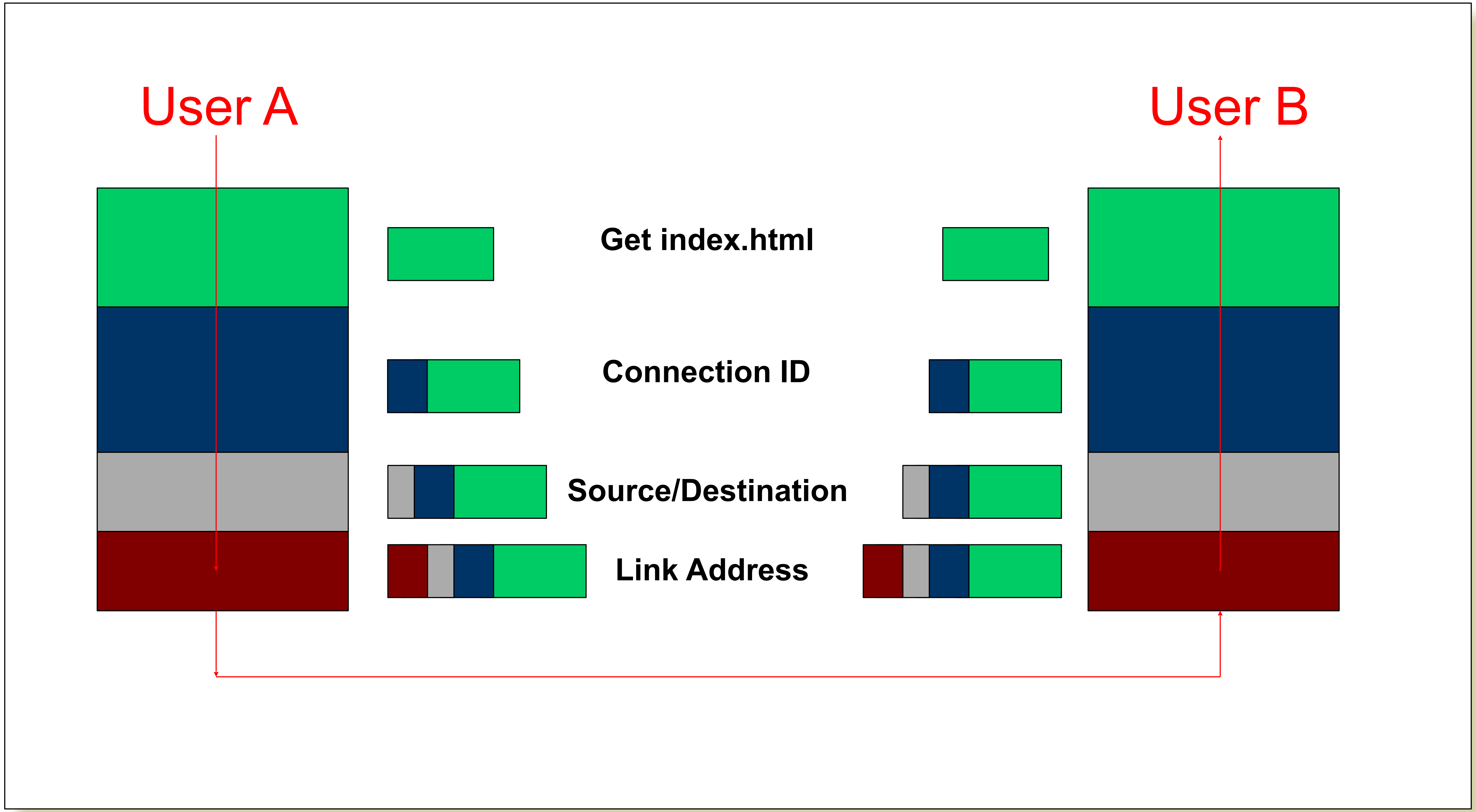


The Internet Protocol Suite



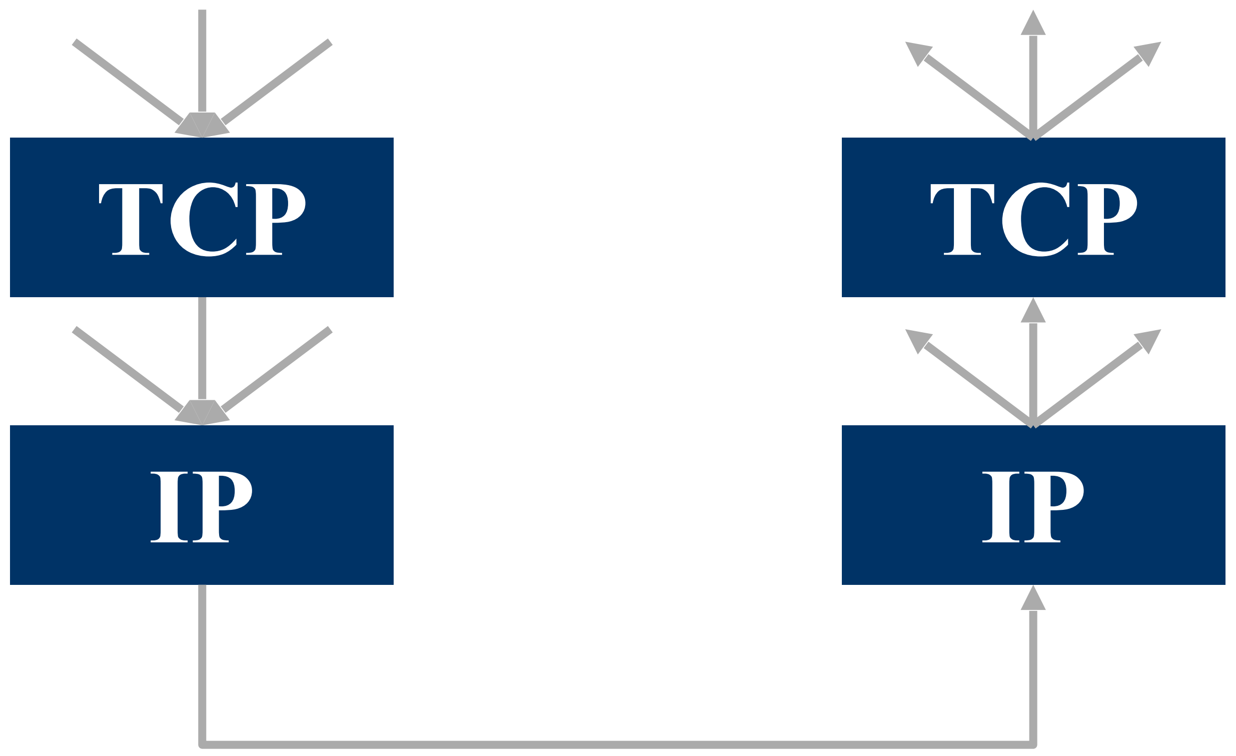
The waist facilitates interoperability

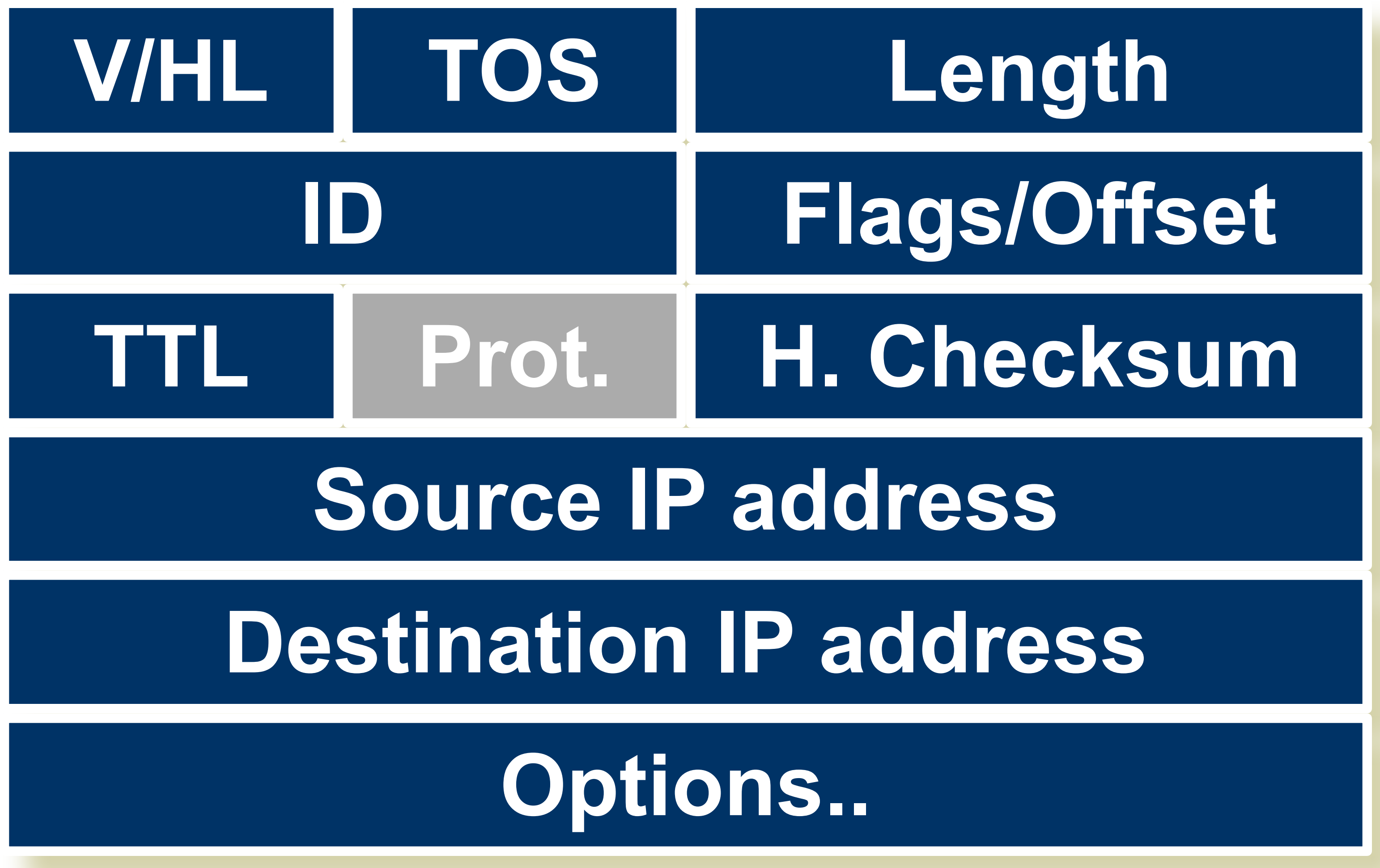
Layer Encapsulation



Multiplexing and Demultiplexing

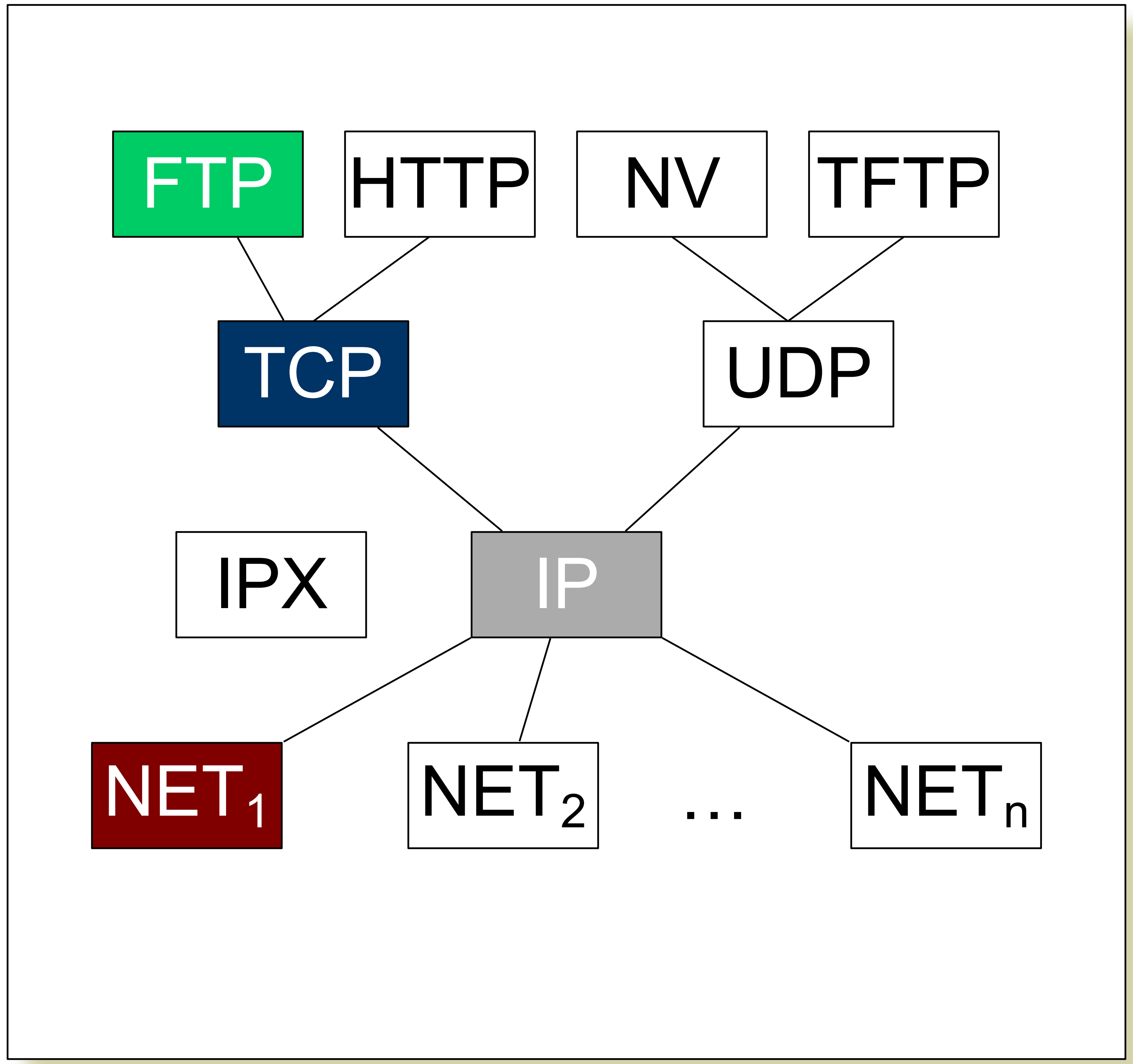
- There may be multiple implementations of each layer.
  - How does the receiver know what version of a layer to use?
- Each header includes a demultiplexing field that is used to identify the next layer.
  - Filled in by the sender
  - Used by the receiver
- Multiplexing occurs at multiple layers. E.g., IP, TCP, …

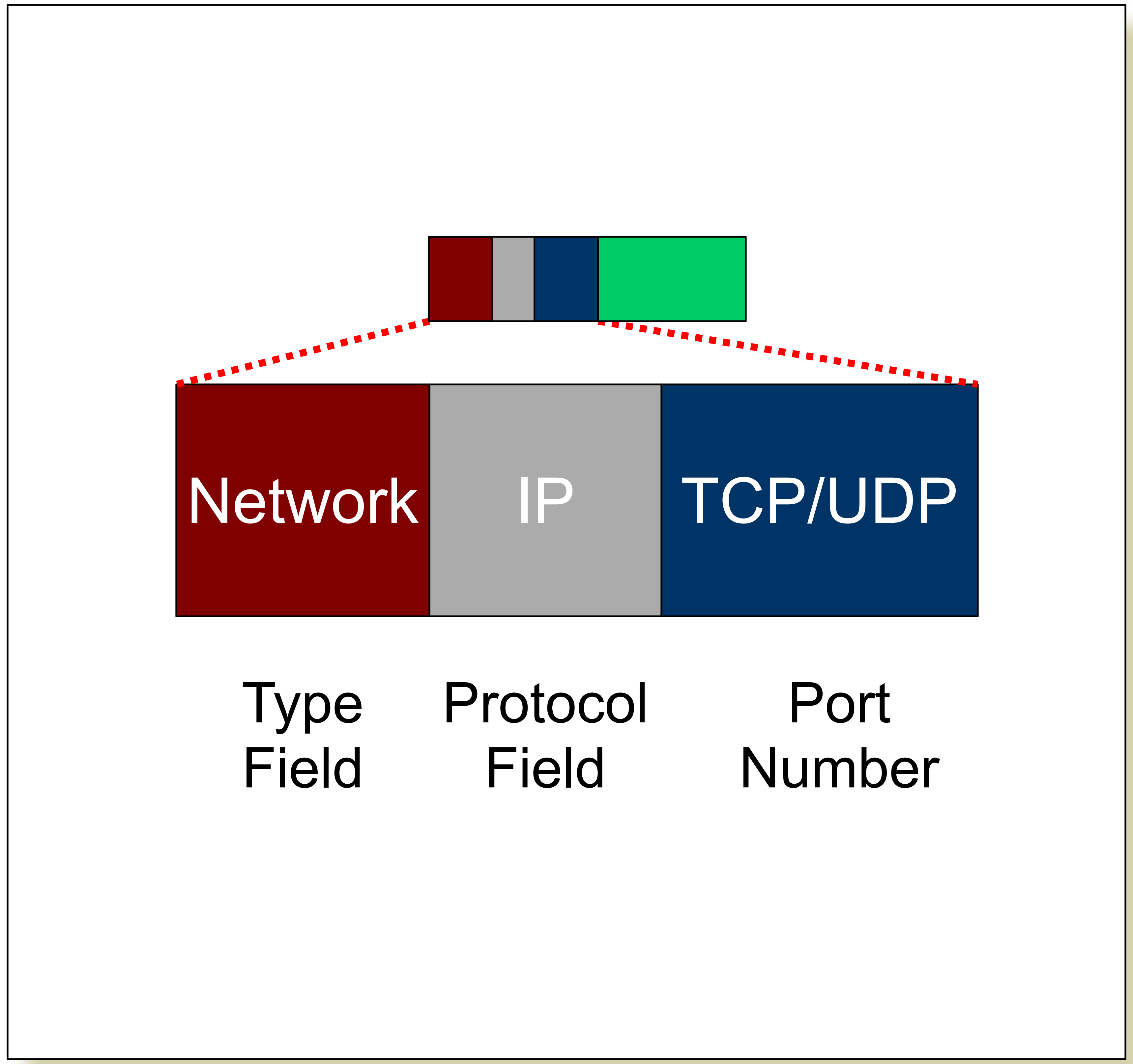




Protocol Demultiplexing

- Multiple choices at each layer





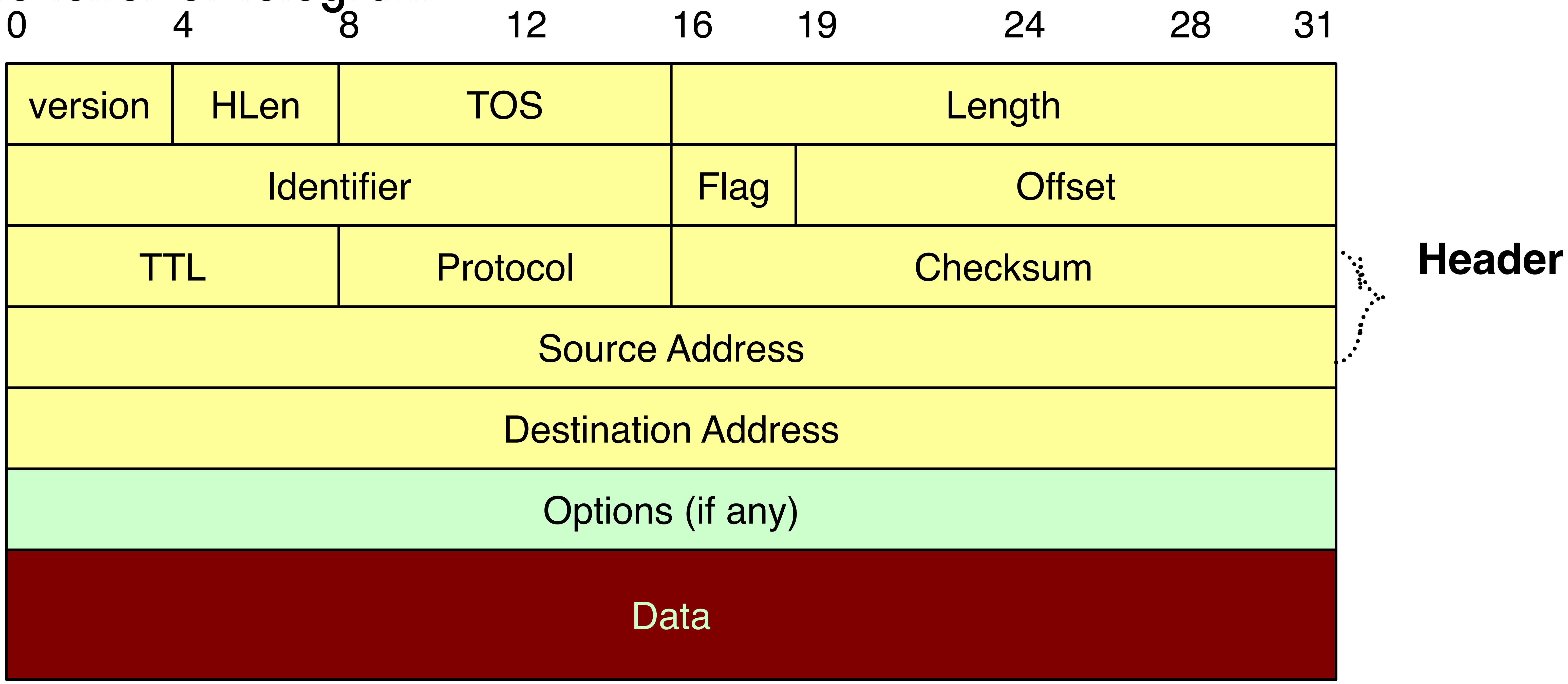
Today’s topic

- Network links and LANs
- Layering and protocols
- Internet design
- Transport protocols

IP Packets/Service Model

- Low-level communication model provided by Internet
- Datagram
  - Each packet self-contained
    - All information needed to get to destination
    - No advance setup or connection maintenance
  - Analogous to letter or telegram

IPv4 Packet Format



IP Addresses: How to Get One?

- Network (network portion):
- Get allocated portion of ISP's address space:

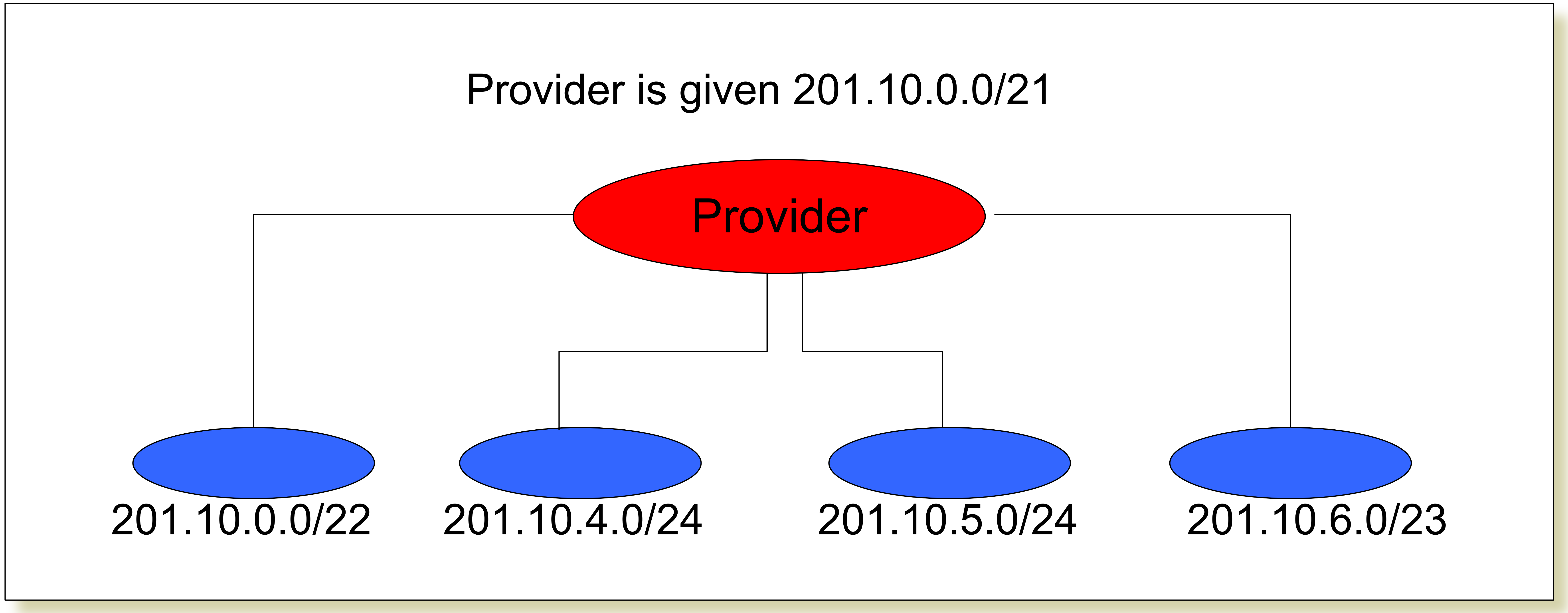
ISP's block	<u>11001000 00010111 0001</u>0000 00000000	200.23.16.0/20
Organization 0	<u>11001000 00010111 0001000</u>0 00000000	200.23.16.0/23
Organization 1	<u>11001000 00010111 0001001</u>0 00000000	200.23.18.0/23
Organization 2	<u>11001000 00010111 0001010</u>0 00000000	200.23.20.0/23
...	....	....
Organization 7	<u>11001000 00010111 0001111</u>0 00000000	200.23.30.0/23

IP Addresses: How to Get One?

- How does an ISP get block of addresses?
  - From **Regional Internet Registries** (RIRs)
    - ARIN (North America, Southern Africa), APNIC (Asia-Pacific), RIPE (Europe, Northern Africa), LACNIC (South America)

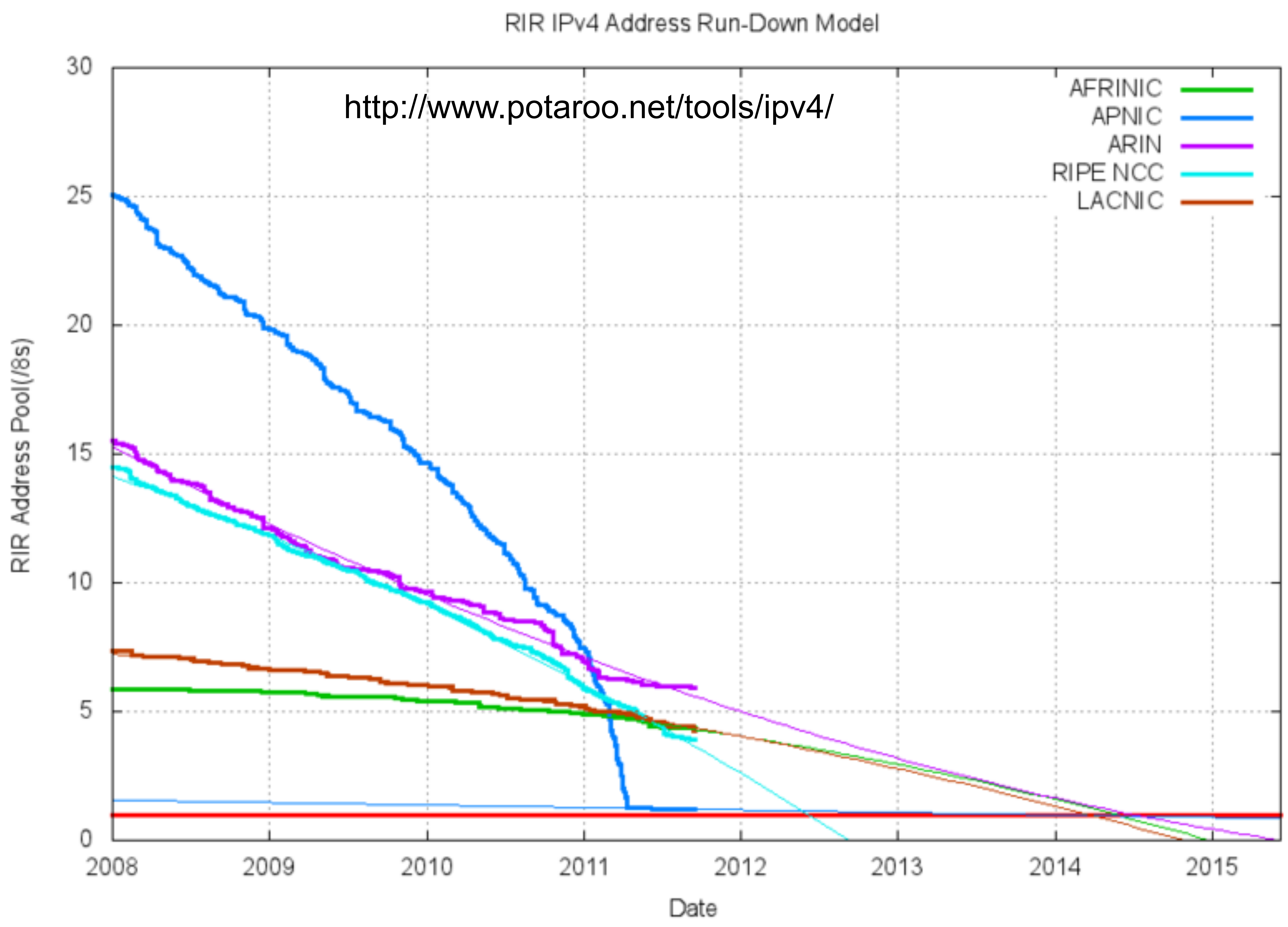
- How about a single host?
  - Hard-coded by system admin in a file
  - DHCP: Dynamic Host Configuration Protocol: dynamically get address: "plug-and-play"
    - Host broadcasts "DHCP discover" msg
    - DHCP server responds with "DHCP offer" msg
    - Host requests IP address: "DHCP request" msg
    - DHCP server sends address: "DHCP ack" msg

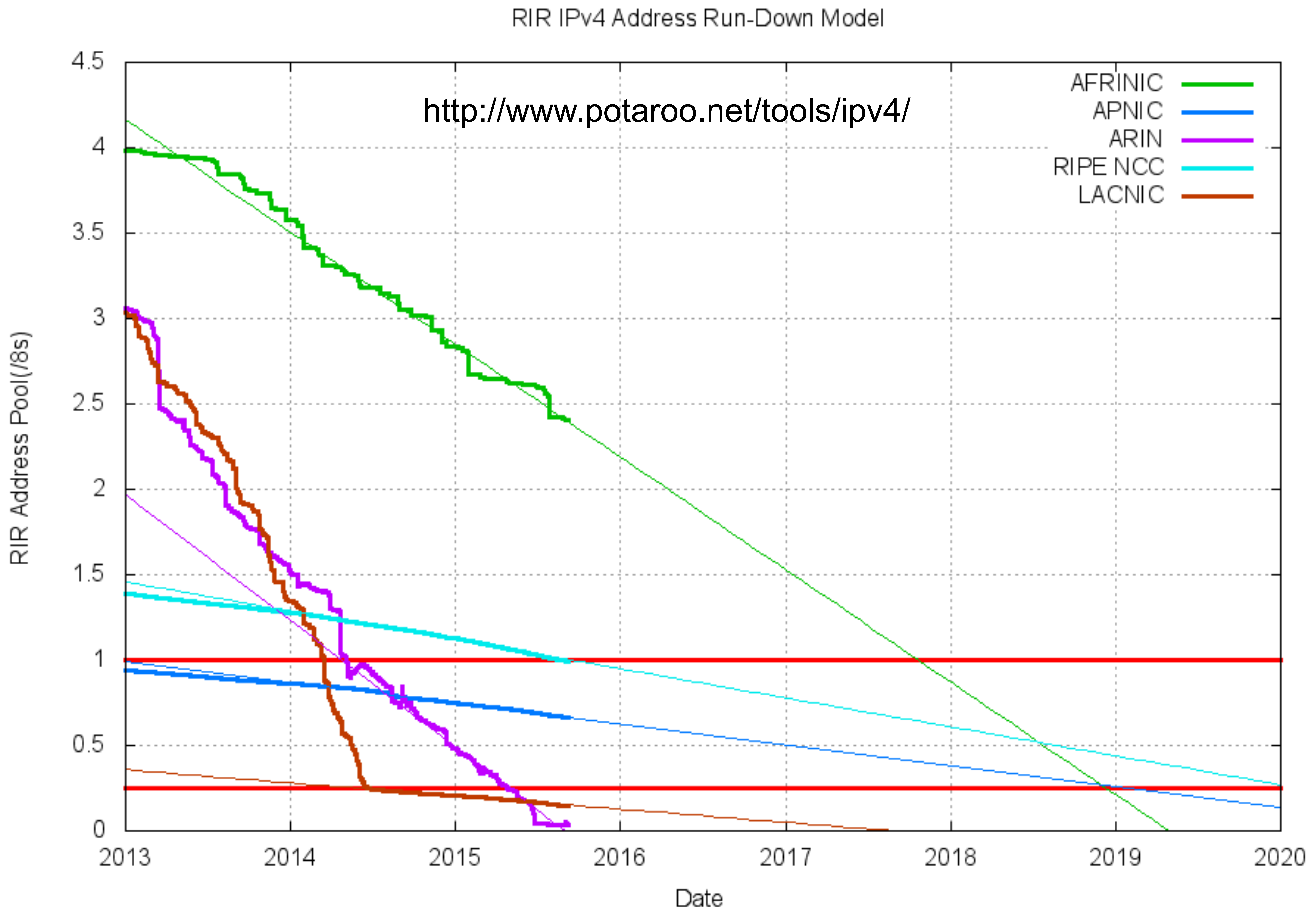
CIDR IP Address Allocation



IP Address Utilization (‘06)

http://xkcd.com/195/


RIR IPv4 Address Run-Down Model
http://www.potaroo.net/tools/ipv4/
AFRINIC
APNIC
ARIN
RIPE NCC
LACNIC
RIR Address Pool(/8s)
30
25
20
15
10
5
0
2008
2009
2010
2011
2012
2013
2014
2015
Date


RIR IPv4 Address Run-Down Model
http://www.potaroo.net/tools/ipv4/
AFRINIC
APNIC
ARIN
RIPE NCC
LACNIC
RIR Address Pool(/8s)
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
2013
2014
2015
2016
2017
2018
2019
2020
Date

What Now?

- Last /8 given to RIR in 1/2011
- Mitigation
  - Reclaim addresses (e.g. Stanford gave back class A in 2000)
  - More NAT?
  - Resale markets
  - Slow down allocation from RIRs to LIRs (i.e. ISPs)
- IPv6?
  -

Host Routing Table Example

- From "netstat –rn"
- Host 128.2.209.100 when plugged into CS ethernet
- Dest 128.2.209.100 → routing to same machine
- Dest 128.2.0.0 → other hosts on same ethernet
- Dest 127.0.0.0 → special loopback address
- Dest 0.0.0.0 → default route to rest of Internet
  - Main CS router: gigrouter.net.cs.cmu.edu (128.2.254.36)

Destination	Gateway	Genmask	Iface
128.2.209.100	0.0.0.0	255.255.255.255	eth0
128.2.0.0	0.0.0.0	255.255.0.0	eth0
127.0.0.0	0.0.0.0	255.0.0.0	lo
0.0.0.0	128.2.254.36	0.0.0.0	eth0

Today's & Tuesday's Lecture

- Network links and LANs
- Layering and protocols
- Internet design
- Transport protocols
- Application design

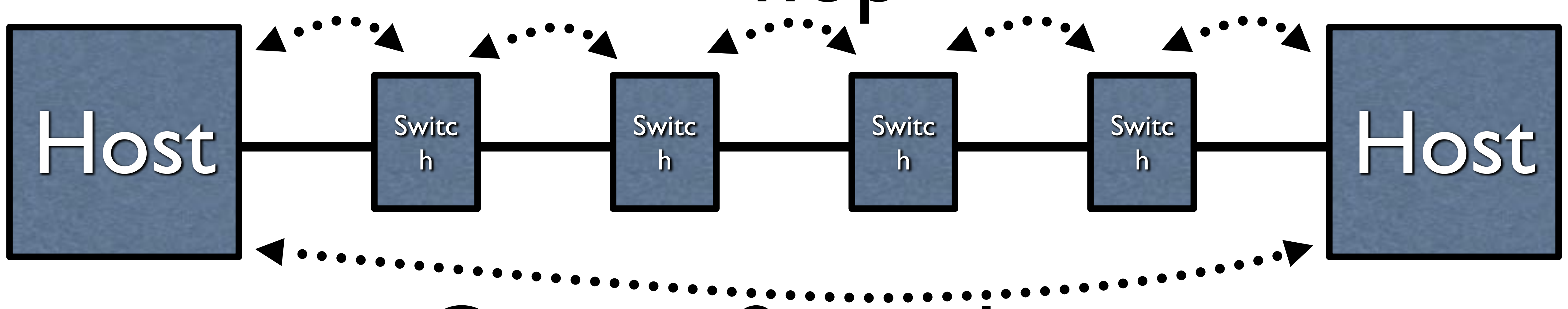
Networks [including end points] Implement Many Functions

- Link
- Multiplexing
- Routing
- Addressing/naming (locating peers)
- **Reliability**
- Flow control
- Fragmentation
- Etc....

Design Question

- If you want reliability, etc.
- Where should you implement it?

Option 1: Hop-by-hop



Option 2: end-to-end

A question

- Is hop-by-hop enough?
  - [hint: What happens if a switch crashes? What if it's buggy and goofs up a packet?]

End-to-End Argument

- Deals with where to place functionality
  - Inside the network (in switching elements)
  - At the edges
- Argument
  - If you have to implement a function end-to-end anyway (e.g., because it requires the knowledge and help of the end-point host or application), **don't implement it inside the communication system**
  - Unless there's a compelling performance enhancement

- Key motivation for split of functionality between TCP,UPD and IP

Further Reading: "End-to-End Arguments in System Design." Saltzer, Reed, and Clark.

User Datagram Protocol (UDP): An Analogy

UDP	Postal Mail
• Single socket to receive messages • No guarantee of delivery • Not necessarily in-order delivery • Datagram – independent packets • Must address each packet	• Single mailbox to receive letters • Unreliable ☺ • Not necessarily in-order delivery • Letters sent independently • Must address each letter

Example UDP applications
Multimedia, voice over IP

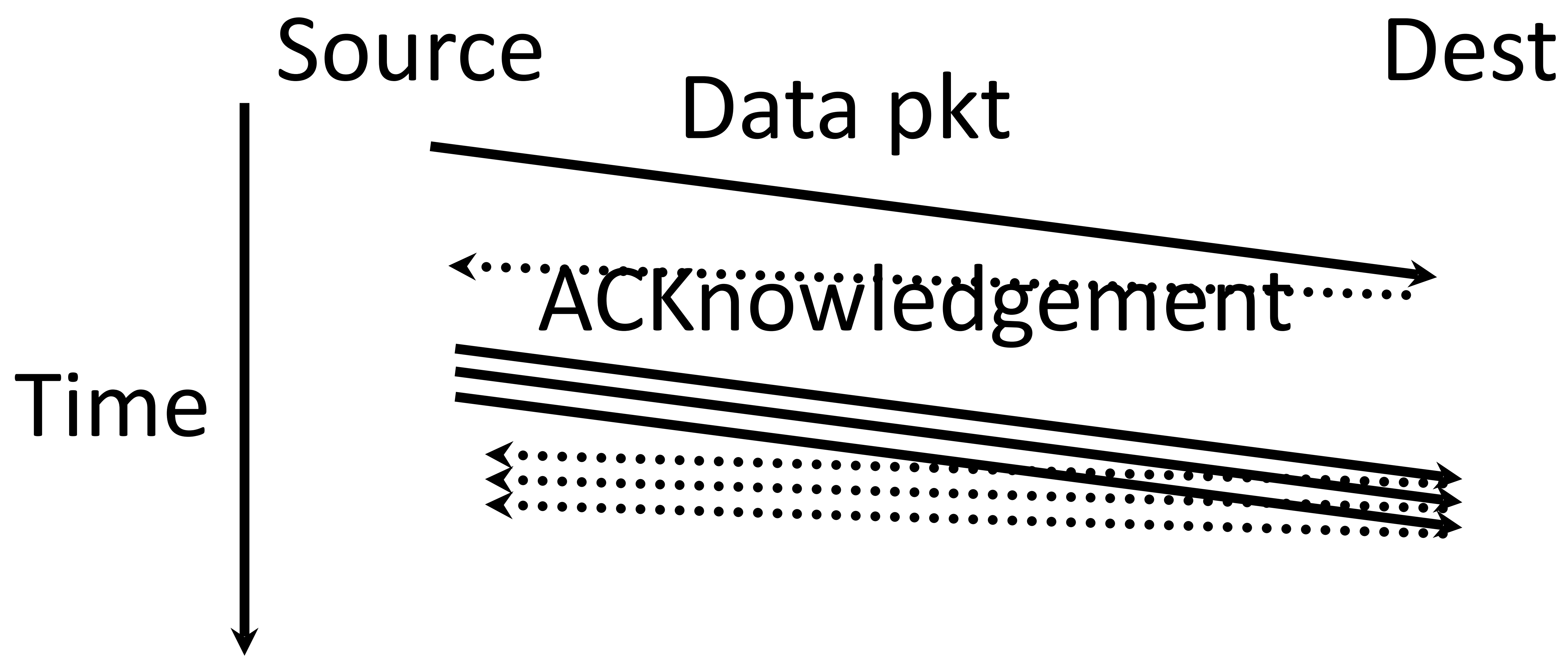
Transmission Control Protocol (TCP): An Analogy

TCP	Telephone Call
• Reliable – guarantee delivery • Byte stream – in-order delivery • Connection-oriented – single socket per connection • Setup connection followed by data transfer	• Guaranteed delivery • In-order delivery • Connection-oriented • Setup connection followed by conversation

Example TCP applications
Web, Email, Telnet

Rough view of TCP

(This is a *very* incomplete view. :)



What TCP does:

1) Figures out which packets got through/lost
2) Figures out how fast to send packets to use all of the unused capacity,

- But not more
- And to share the link approx. equally with other senders

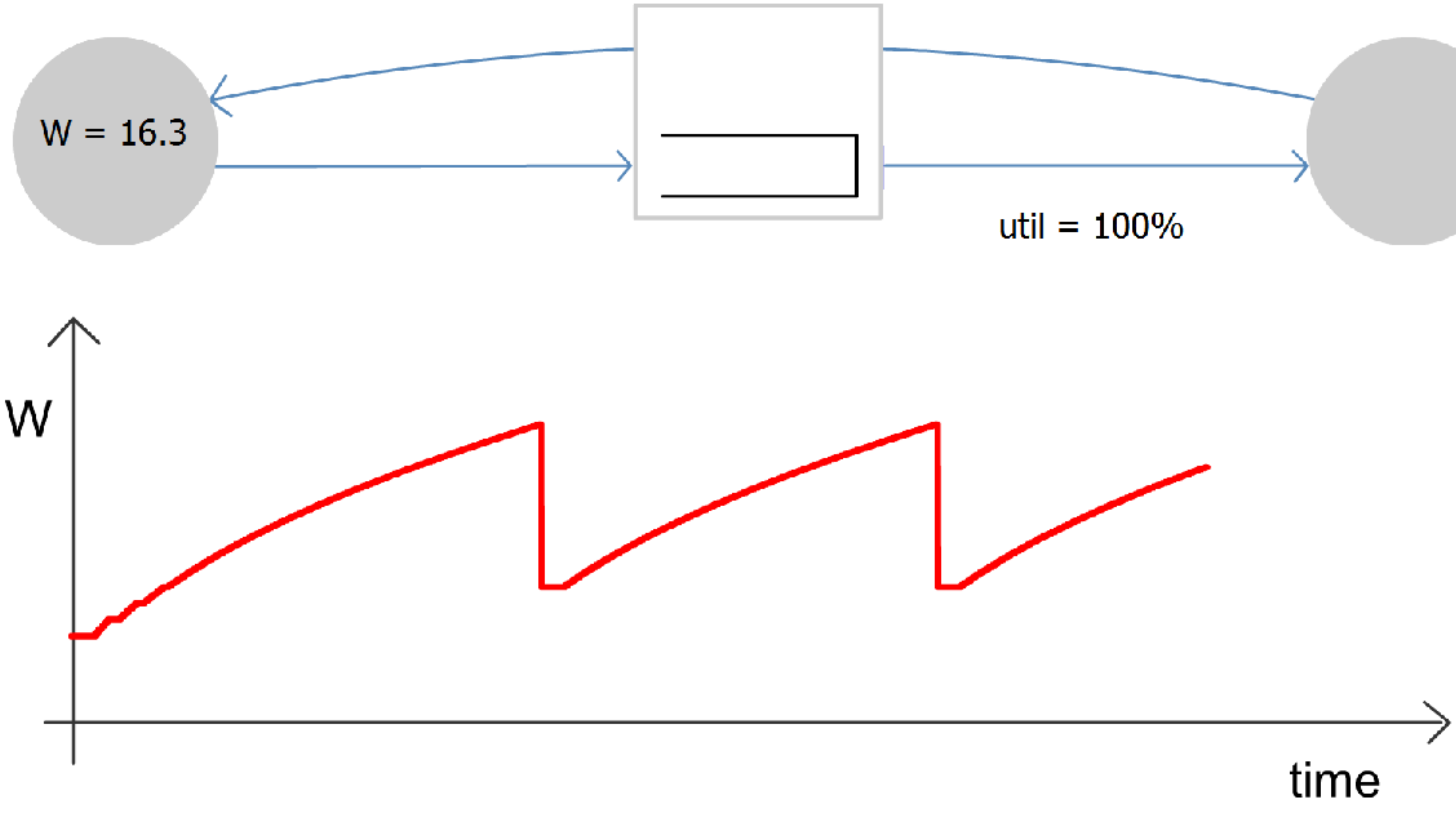
Questions to ponder

- If you have a whole file to transmit,
  how do you send it over the Internet?
  - You break it into packets (packet-switched medium)
  - TCP, roughly speaking, has the sender tell the receiver “got it!” every time it gets a packet. The sender uses this to make sure that the data’s getting through.
  - But by e2e, if you have to acknowledge the correct receipt of the entire file... why bother acknowledging the receipt of the individual packets???

- This is a bit of a trick question – it’s not asking e2e vs in-network. :)
  The answer: Imagine the waste if you had to retransmit the entire file because one packet was lost. Ow.

Single TCP Flow

Router with large enough buffers for full link utilization



Why not always use TCP?

- TCP provides “more” than UDP
- Why not use it for everything??

- A: Nothing comes for free…
  - Connection setup (take on faith) – TCP requires one round-trip time to setup the connection state before it can chat…
  - How long does it take, using TCP, to fix a lost packet?
    - At minimum, one “round-trip time” (2x the latency of the network)
    - That could be 100+ milliseconds!
  - If I guarantee in-order delivery,
    what happens if I lose one packet in a stream of packets?

Design trade-off

- If you’re building an app…

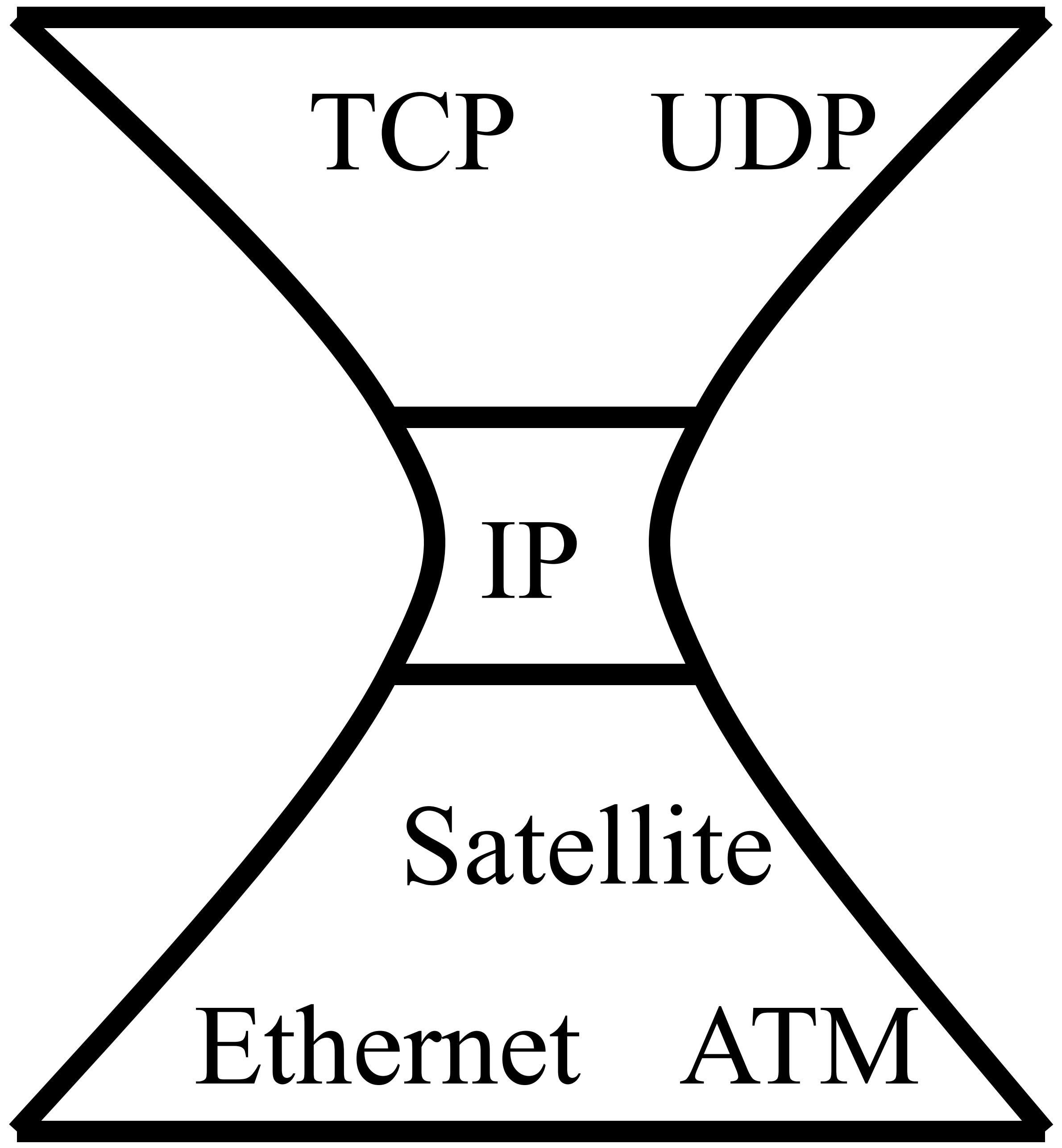
- Do you need everything TCP provides?
- If not:
  - Can you deal with its drawbacks to take advantage of the subset of its features you need?
  - OR
  - You’re going to have to implement the ones you need on top of UDP
    - Caveat: There are some libraries, protocols, etc., that can help provide a middle ground.
    - Takes some looking around - they’re not as standard as UDP and TCP.

In contrast to UDP

- UDP doesn't figure out how fast to send data, or make it reliable, etc.
- So if you write() like mad to a UDP socket...
- It often silently disappears. *Maybe* if you're lucky the write() call will return an error. But no promises.

Summary: Internet Architecture

- Packet-switched datagram network
- IP is the “compatibility layer”
  - Hourglass architecture
  - All hosts and routers run IP
- Stateless architecture
  - no per flow state inside network



Summary: Minimalist Approach

- Dumb network
  - IP provide minimal functionalities to support connectivity
    - Addressing, forwarding, routing
- Smart end system
  - Transport layer or application performs more sophisticated functionalities
    - Flow control, error control, congestion control
- Advantages
  - Accommodate heterogeneous technologies (Ethernet, modem, satellite, wireless)
  - Support diverse applications (telnet, ftp, Web, X windows)
  - Decentralized network administration