CS 43: Computer Networks

15: The Network Layer

November 2-5, 2020

SWARTHMORE COLLEGE

The Network Layer!

Application: the application (e.g., the Web, Email)

Transport: end-to-end connections, reliability

Network: routing

Link (data-link): framing, error detection

Physical: 1’s and 0’s/bits across a medium (copper, the air, fiber)

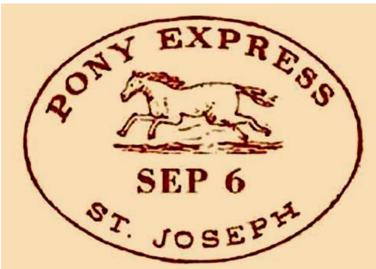
Network Layer

- DARPAnet Primary Goal: Connect Hosts

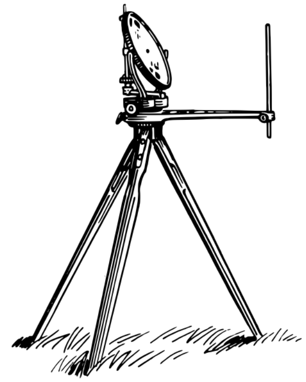
- "islands" of networks: SATNet, Packet Radio, Ethernet: how do we connect them?

- Routers forward packets using a common Internet Protocol
  - *Any* underlying data link protocol
  - *Any* higher layer transport protocol

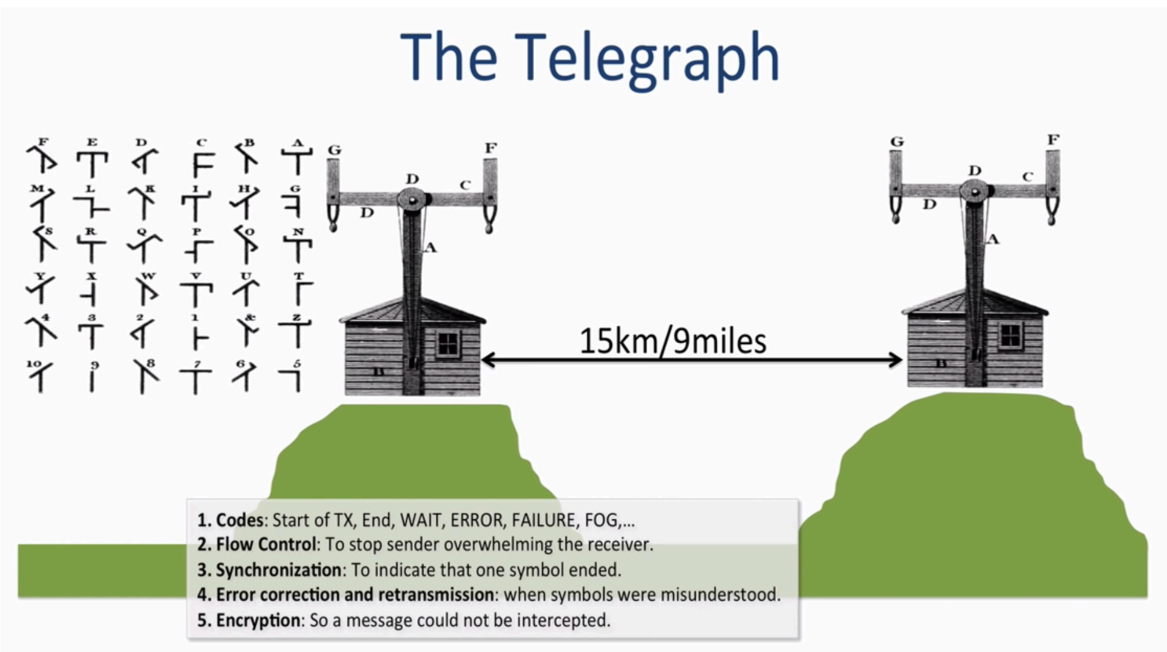
History of Communication



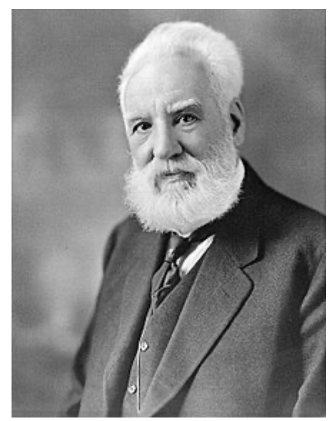
Fire Beacons
Carrier Pigeons
Human Messengers
Horse Relays – Pony Express



Wireless telegraph
- speed of light
- compression
- limited information

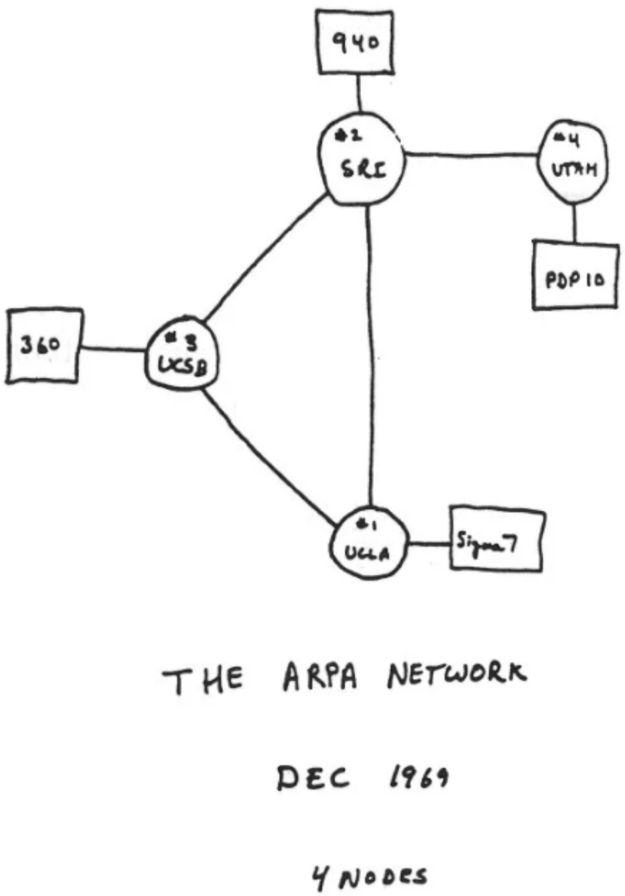


Telephone Network



Courtesy: Stanford University

History of the Internet: ARPANET



Courtesy: Scientific American

ARPANET
- Connect academic computers together
- ARPANET Nodes: UCLA, SRI, UCSB, UTAH

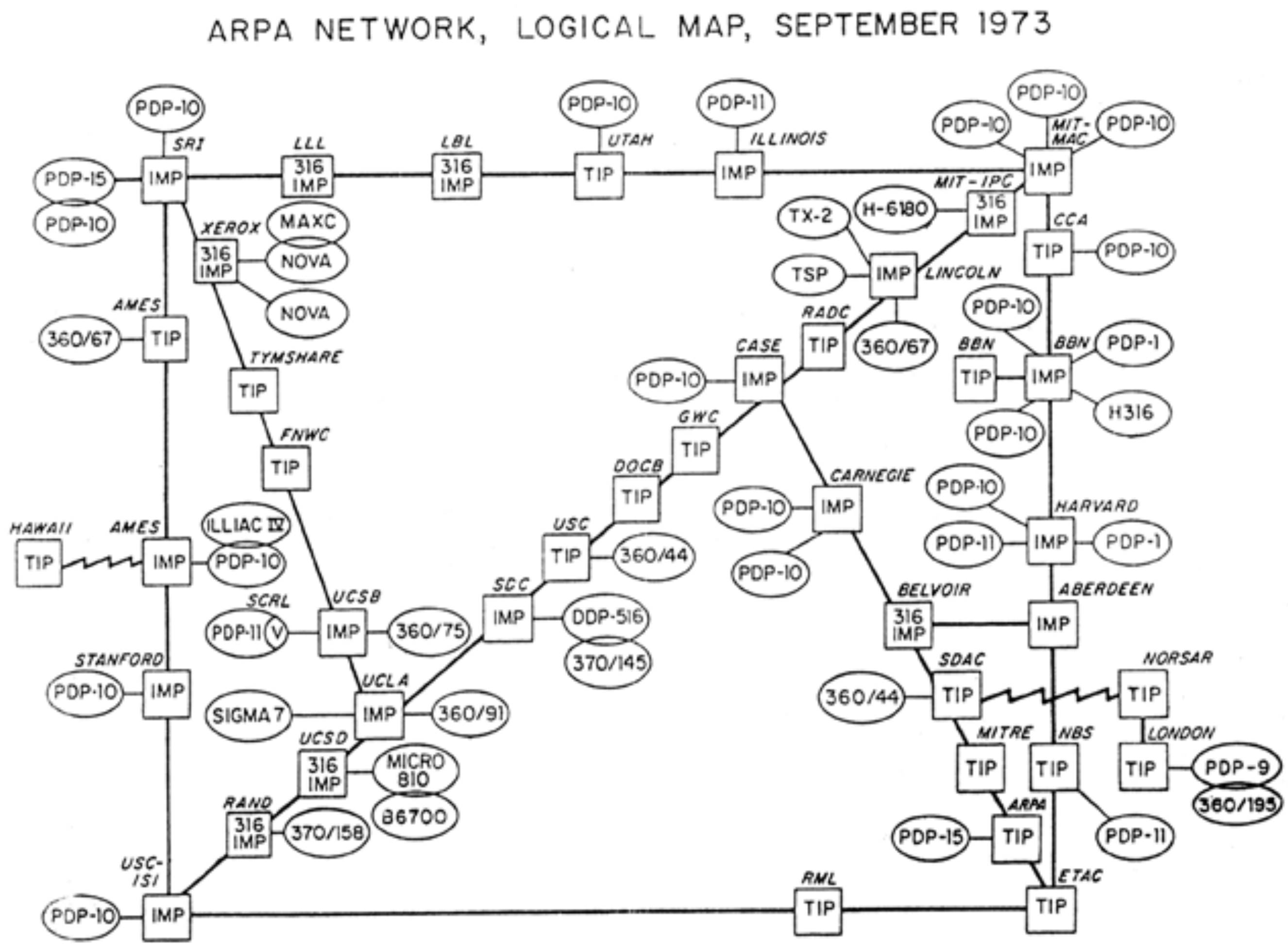
First host-to-host protocol
- two cross country links

Internet: A network of networks

- ARPANET
- NPLNET
- SATNET
- Packet radio networks
- Ethernet LAN

Network control protocol

Jan 1 1983: Flag Day Transition to TCP IP



Source: Wikimedia Commons

Pioneers of the early Internet

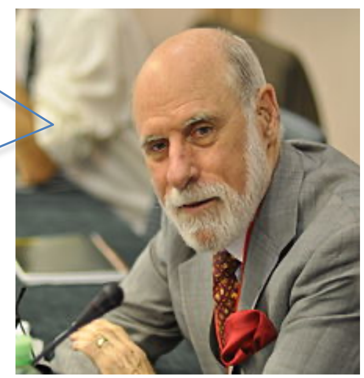
Packet Switched Networks

“Information Flow in Large Communication Nets”

Chief Protocol Architect of the Internet
"The Design Philosophy of the DARPA Internet Protocols”

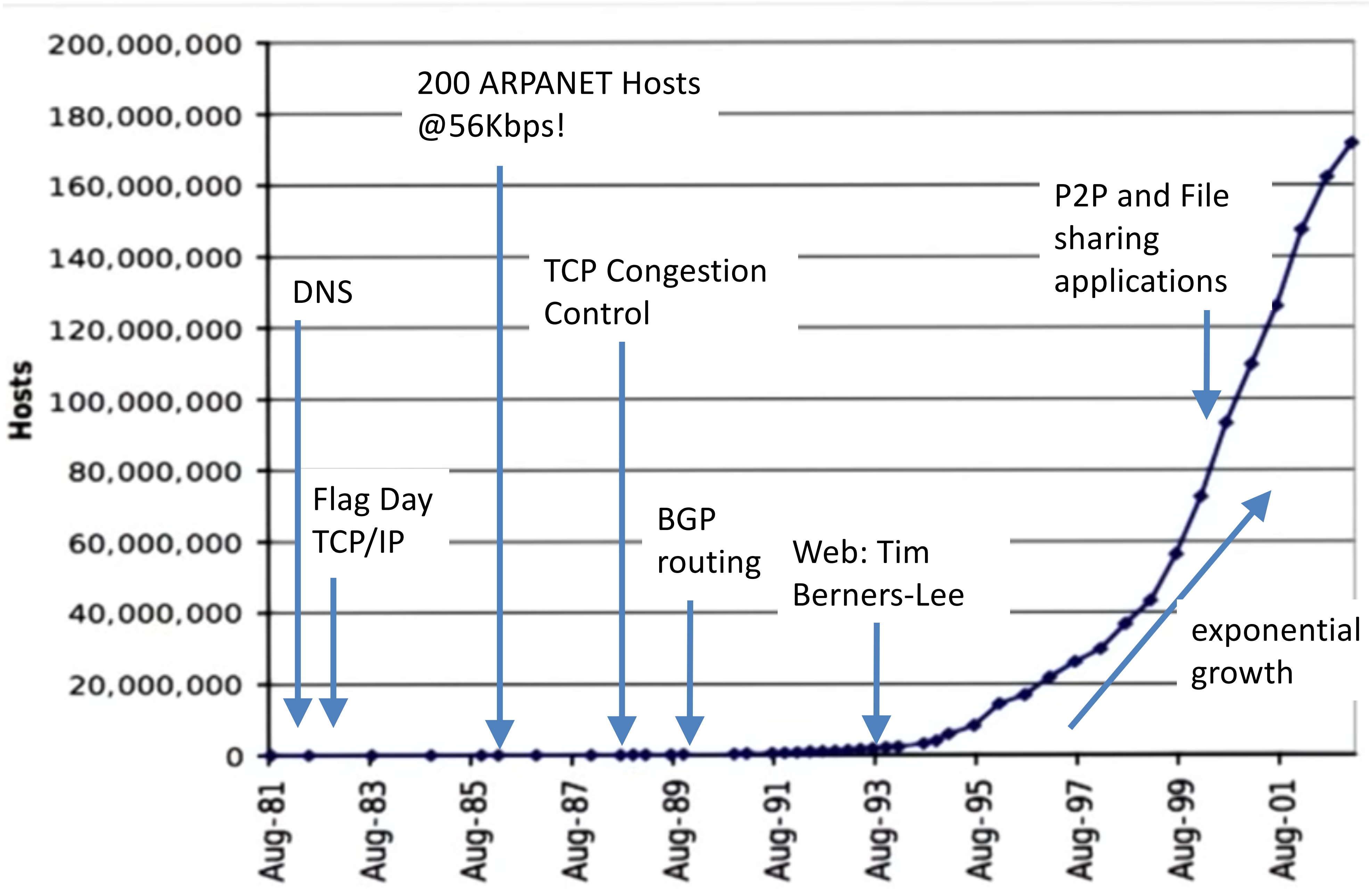
Swat Alum!

Cerf & Kahn: TCP/IP protocols
Turing Award Winners



Vincent Cerf and Bob Kahn

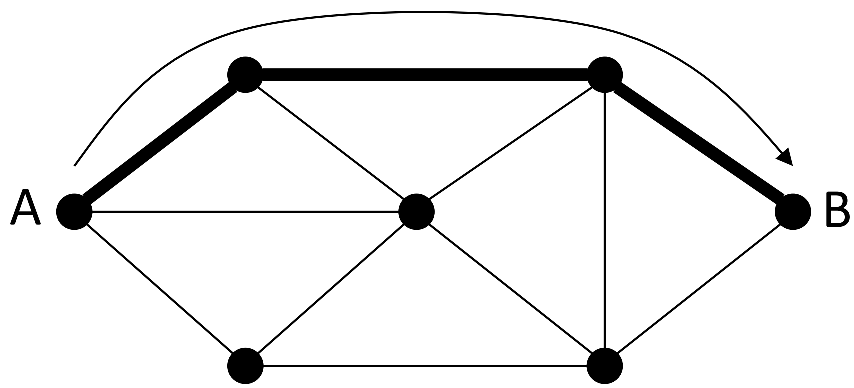
Picture Source: Wikipedia



200 ARPANET Hosts @56Kbps!
DNS
TCP Congestion Control
P2P and File sharing applications
Flag Day TCP/IP
BGP routing
Web: Tim Berners-Lee
exponential growth
Hosts
200,000,000
180,000,000
160,000,000
140,000,000
120,000,000
100,000,000
80,000,000
60,000,000
40,000,000
20,000,000
0
Aug-81
Aug-83
Aug-85
Aug-87
Aug-89
Aug-91
Aug-93
Aug-95
Aug-97
Aug-99
Aug-01

Circuit Switching

- Reserve path in advance



- (Old) telephone system

Why doesn't the Internet (typically) use circuits?

A. It's too slow to establish a connection.

B. It doesn't offer good enough performance.

C. It wastes resources.

D. It requires too many resources.

E. Some other reason.

Why doesn't the Internet (typically) use circuits?

A. It's too slow to establish a connection
   - some setup state required but not prohibitively slow per connection – but doesn't scale with growth of connections considering today's Internet applications)

B. It doesn't offer good enough performance.
   – when the end-to-end path is reserved, you have dedicated line per connection.

C. It wastes resources.

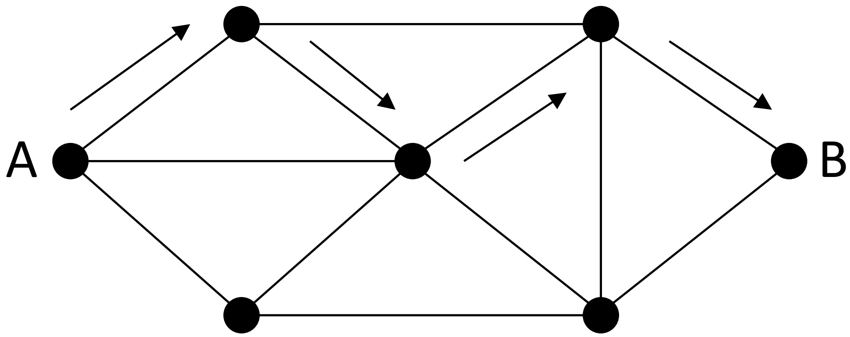
D. It requires too many resources.

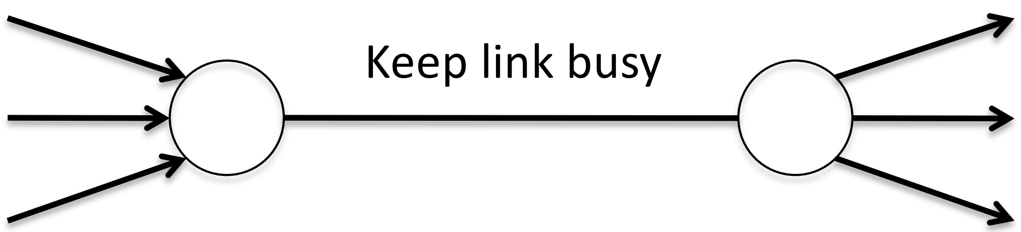
E. Some other reason.

Packet Switching

- Do we always need to reserve a link?

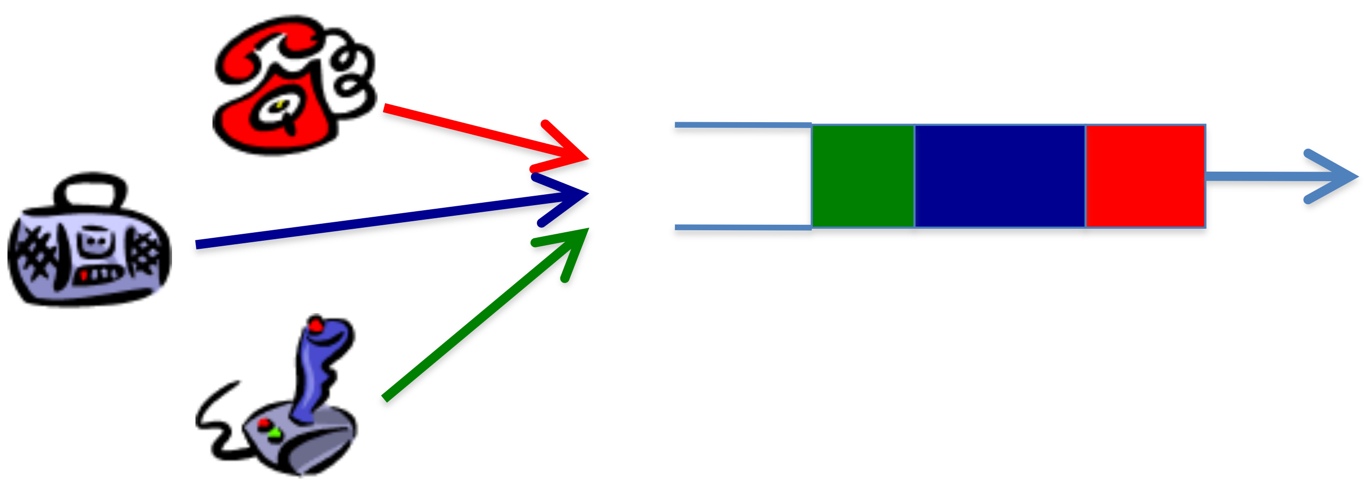
- <u>Statistical multiplexing</u>
  - Assign multiple conversations to a physical path
  - At any given time, one will have something to say





Packet Switching: Statistical Multiplexing

- Data traffic is bursty
  - Telnet, email, Web browsing, …
- Avoid wasting bandwidth
  - One host can send more when others are idle



Which of the following is/are generally true of packet vs. circuit switching?

1. Packet switching has more variance in performance.
2. Circuit switching is reliable.

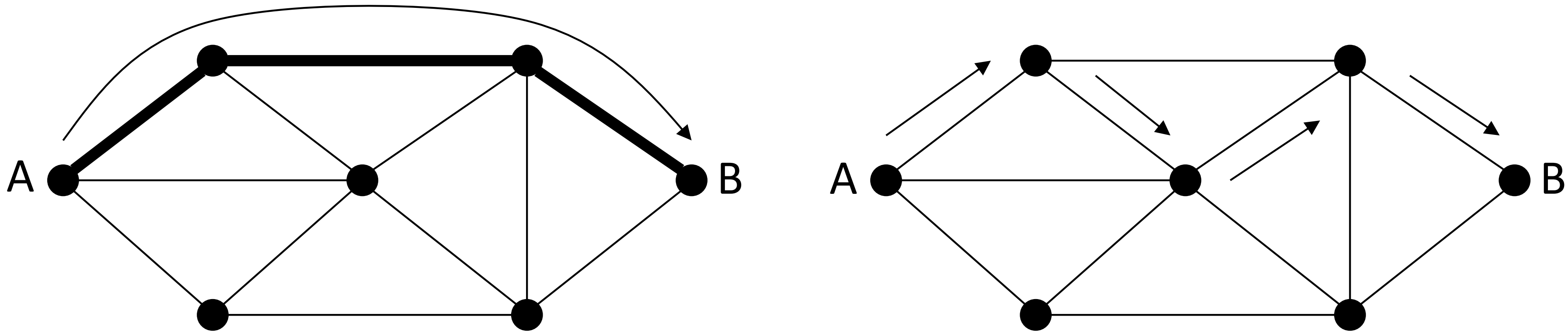
A. Only 1 is true.
B. Only 2 is true.
C. Both 1 and 2 are true.
D. Neither 1 nor 2 are true.

Which of the following is/are generally true of packet vs. circuit switching?

1. Packet switching has more variance in performance.
2. Circuit switching is reliable.

A. Only 1 is true.
B. Only 2 is true.
C. Both 1 and 2 are true.
D. Neither 1 nor 2 are true.

Circuit-switching vs. Packet switching



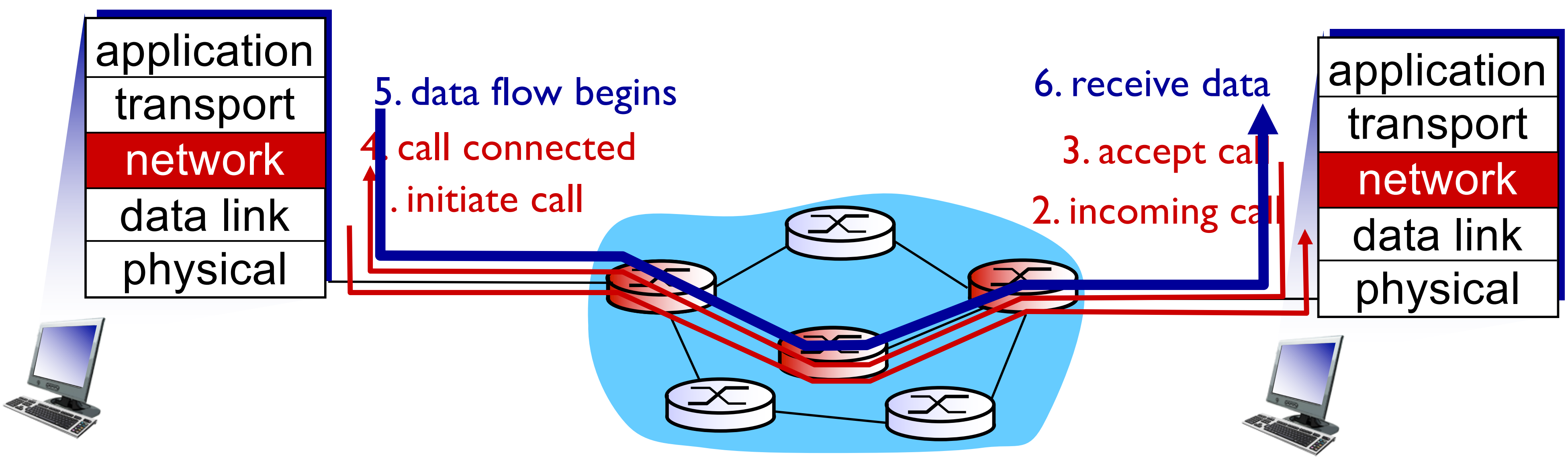
- Circuit switching: establish path, send data
  - Reserve resources, provide performance control
  - Example: telephone system
- Packet switching: forward packets hop by hop
  - Fair sharing despite bursts, statistical multiplexing
  - Example: postal system

Datagram vs. "Virtual Circuit"

- Datagram network provides network-layer connectionless service (packet switching)
- Virtual-circuit network provides network-layer connection service (like circuit switching)

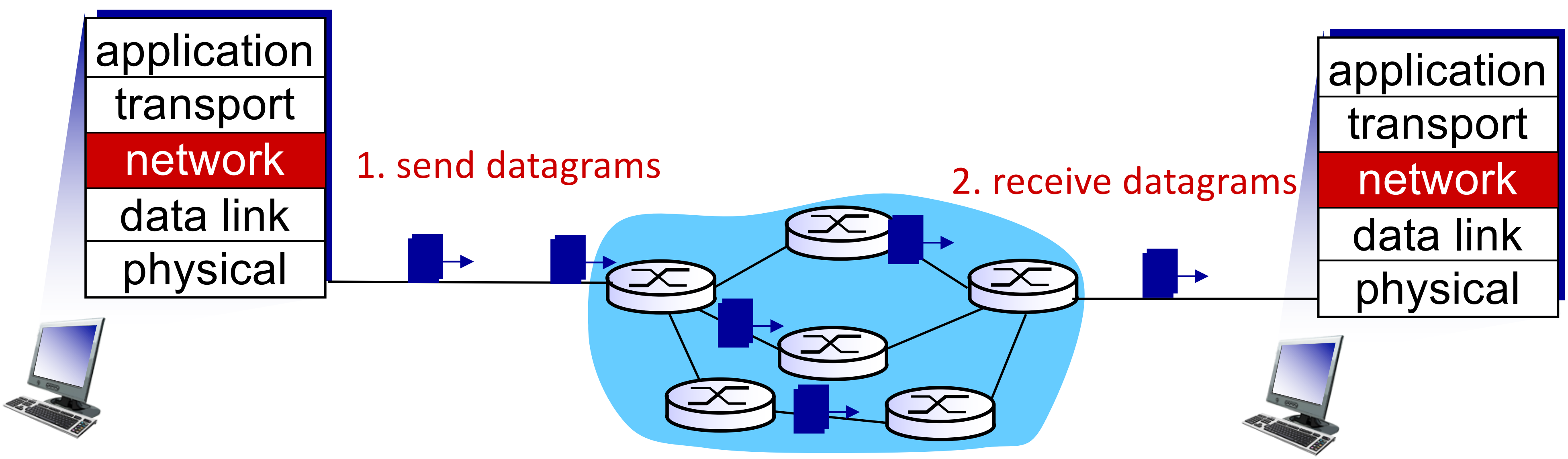
Virtual circuits: Signaling Protocols

- Used to setup, maintain, teardown VC
- Used in ATM (Asynchronous Transfer Mode), frame-relay, X.25
- Less common in today's Internet



Datagram Networks

- No call setup at network layer
- Routers: no state about end-to-end connections
  - no network-level concept of "connection"
- Packets forwarded individually towards destination



Why doesn't the network layer do more?

Compress data

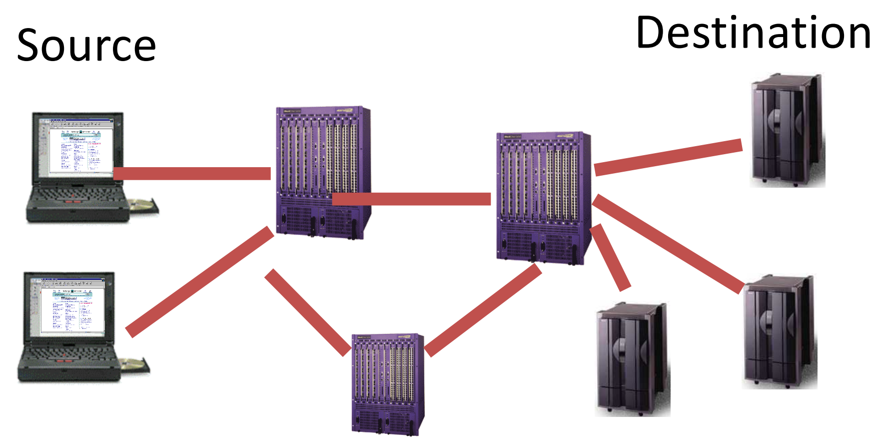
Serve Cached Data

Add Security

Provide reliability

Migrate connections

.... the list is long



The End-to-End Principle

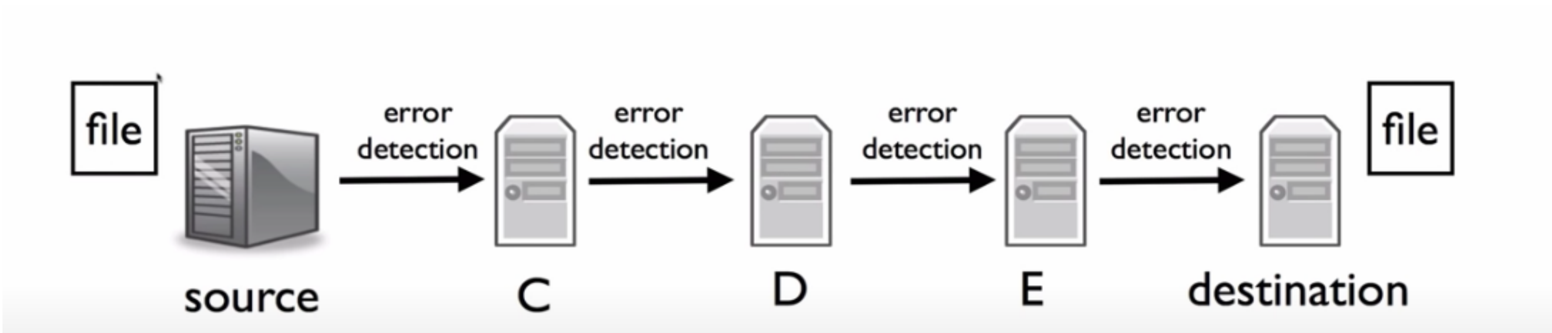
"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system."

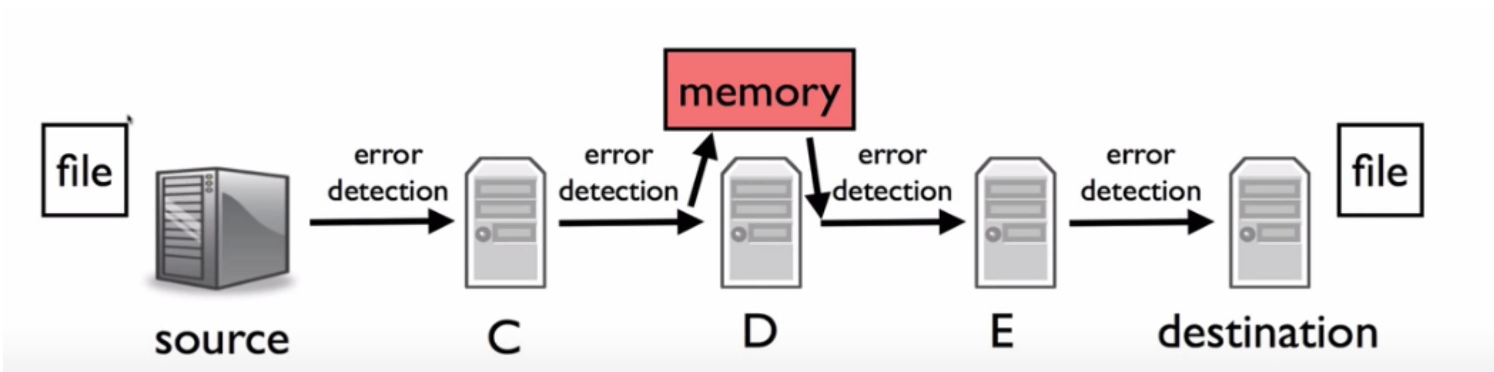
- Saltzer, Reed and Clark

End-to-end Arguments in System Design, 1984

I.e. The network can help you – but only the application is responsible for correctness. No one else has the complete picture of the requirements. You can't depend on the network.

The End-to-End Principle





No checks for errors in storage! Network can help but can't be responsible for correctness.

Courtesy: Stanford

The Strong End-to-End Principle

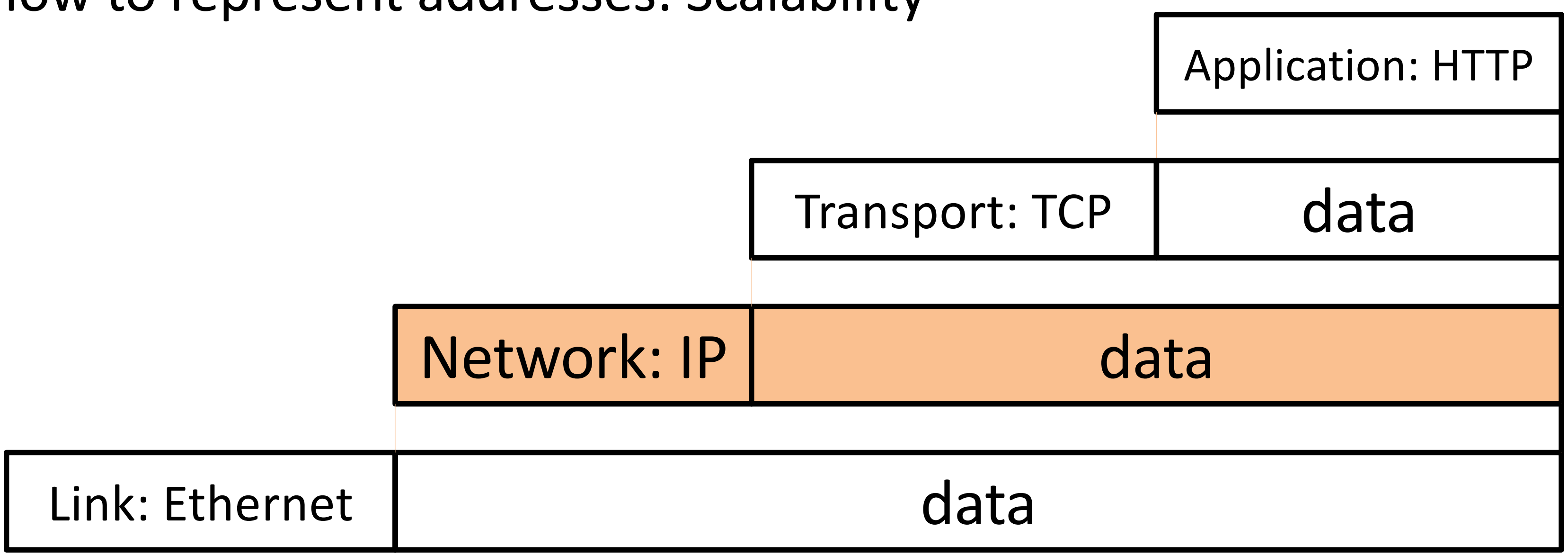
The network's job is to transmit datagrams as efficiently and flexibly as possible. Everything else should be done at the fringes.

-RFC 1958

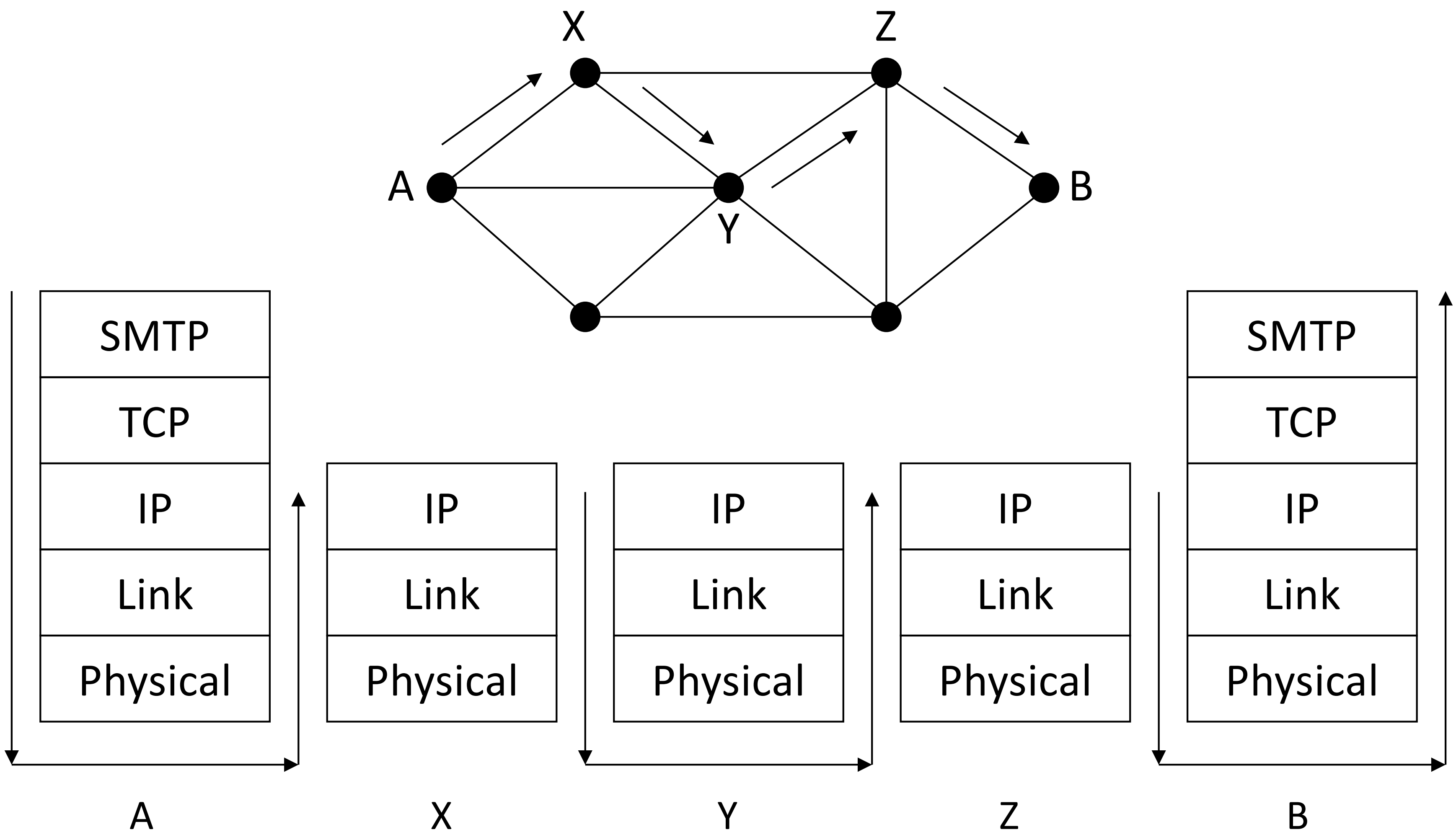
Courtesy: Stanford

Network Layer

- Function: Route packets end-to-end on a network, through multiple hops
- Key challenge
  - How to route packets: Convergence
  - How to represent addresses: Scalability



Example of Internet Routing



Network layer involved at every hop along the path.

Network Layer Functions

- Forwarding: move packets from router's input to appropriate router output
("data plane")

- Routing: determine route taken by packets from source to destination.
("control plane")

When should a router perform routing? Forwarding?

A. Do both when a packet arrives.

B. Route in advance, forward when a packet arrives.

C. Forward in advance, route when a packet arrives.

D. Do both in advance.

E. Some other combination

When should a router perform routing? Forwarding?

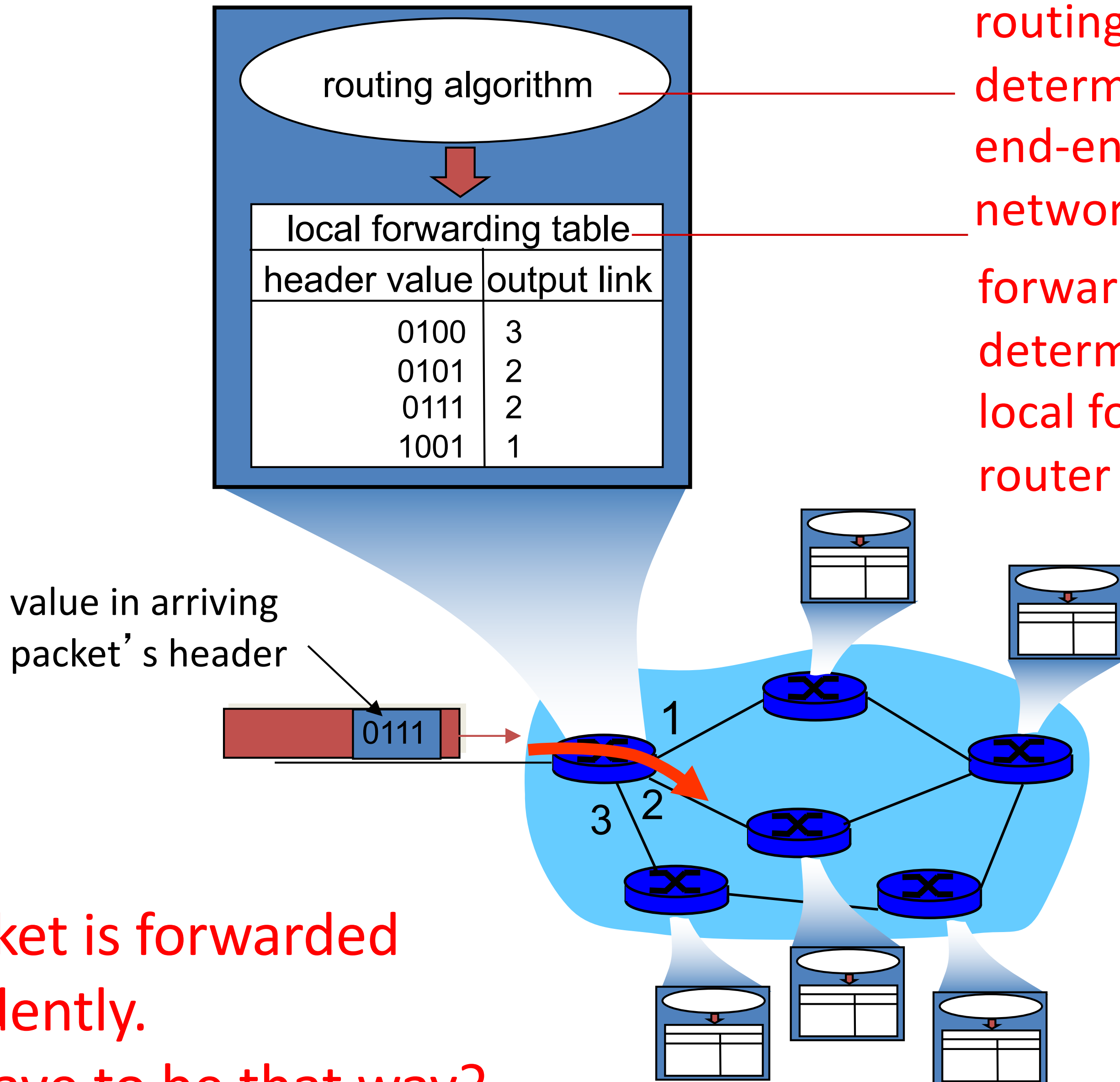
Route in advance, forward when a packet arrives.

- Forwarding:
  - Copying bytes from one interface to another, can't forward in advance
  - forwarding needs to happen very quickly: millions of packets per second
- Routing:
  - High-level decision that we do dynamically, at different time-scales than forwarding
  - route in advance and populate a table to see where it is destined

Network Layer Functions

- Forwarding: move packets from router's input to appropriate router output
  – Look up in a table
- Routing: determine route taken by packets from source to destination.
  – Populating the table

Interplay between routing and forwarding



routing algorithm determines end-end-path through network

forwarding table determines local forwarding at this router

Each packet is forwarded independently.
Does it have to be that way?

How should we populate a router's forwarding table?

A. A person should add entries to the table.

B. A program external to the router should add entries to the table.

C. Routers should communicate with each other to add entries to their tables.

D. Some other mechanism.

How should we populate a router's forwarding table?

A. A person should add entries to the table (policy decisions).

B. A program external to the router should add entries to the table (Software defined networking).

C. Routers should communicate with each other to add entries to their tables (used today).

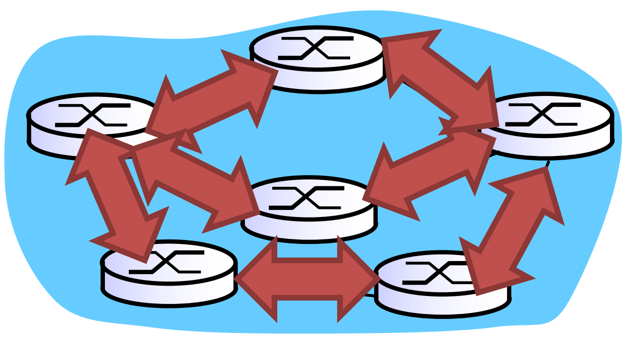
D. Some other mechanism.

Routing

Traditional

- Routers run a routing protocol to exchange state.

- Use state to build up the forwarding table.

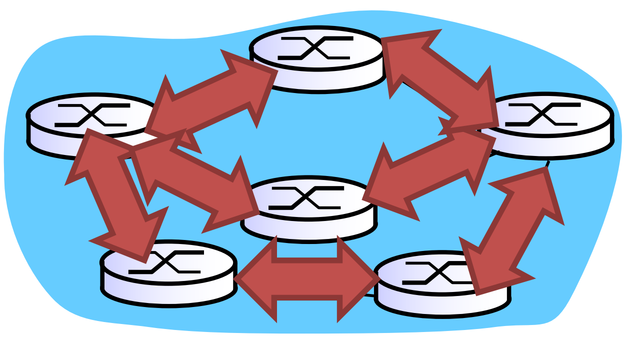


Assume this is the type of routing we're talking about unless we explicitly say otherwise!

Routing

Traditional

- Routers run a routing protocol to exchange state.
- Use state to build up the forwarding table.

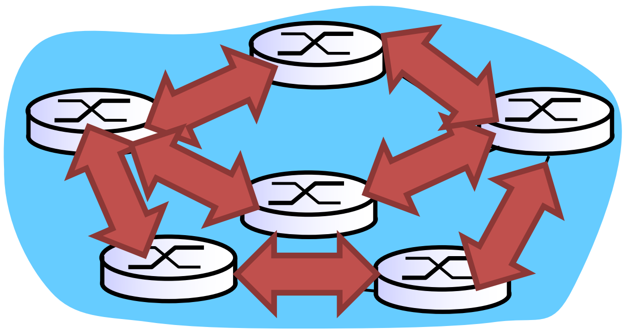


"Software-Defined"

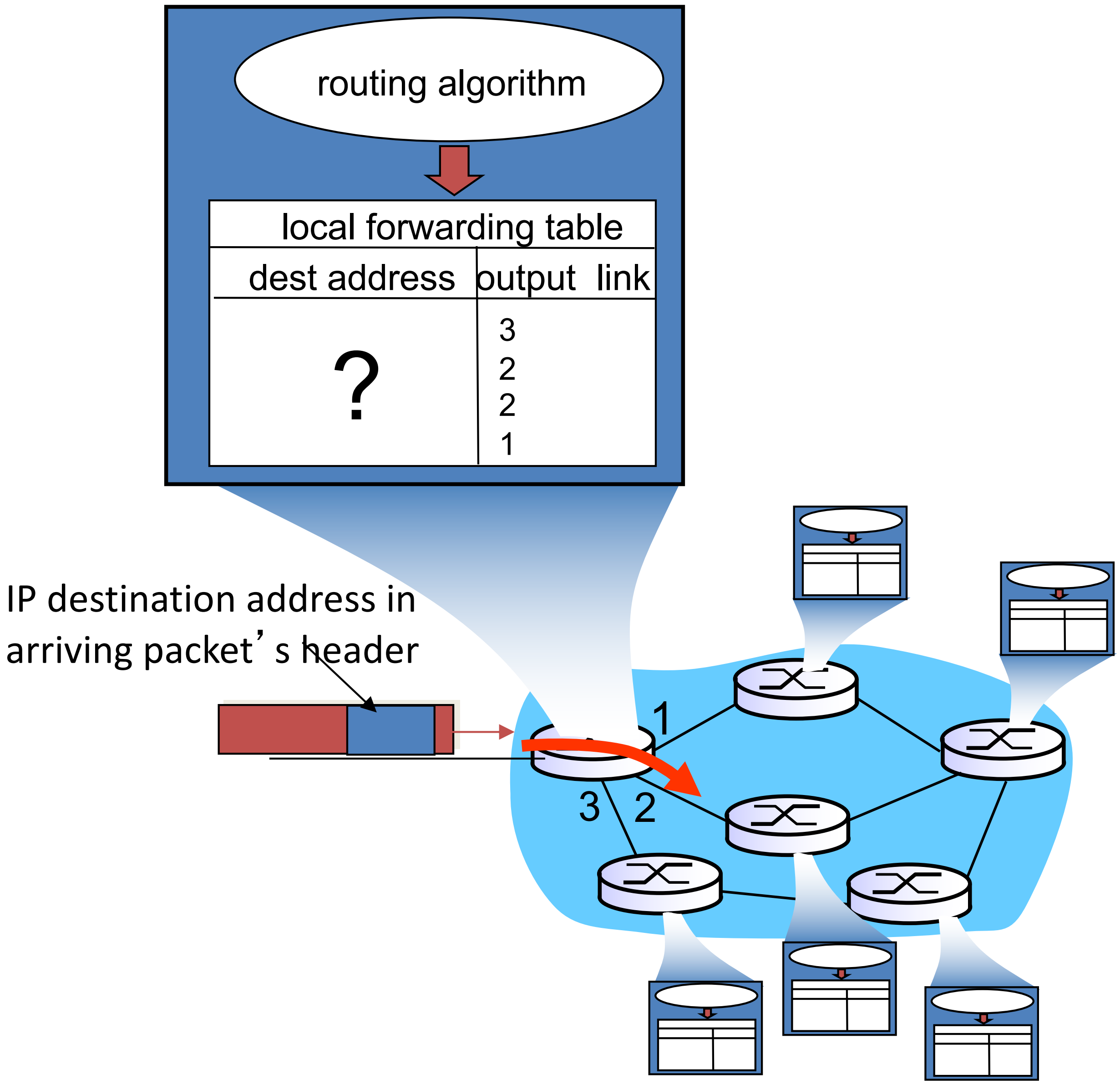
- Routers are dumb, just do what they're told.
- Controller service explicitly tells each router what to do.
- Rare on the Internet, hot topic in data centers.

Datagram Forwarding

- Routers periodically exchange state.
- Use the state to build a forwarding table (FIB – Forwarding Information Base)



Datagram forwarding table



Routers exchange state (we'll save the what and when for later). They decide, for each destination, how to get there, and build a lookup structure for their forwarding table. What should they build?

A. A list – scan for the destination.

B. A hash table – look up the destination.

C. A tree – Follow branches that lead to the destination.

D. Some other software structure.

E. We can't do this in software, we need special hardware.

Routers exchange state (we'll save the what and when for later). They decide, for each destination, how to get there, and build a lookup structure for their forwarding table. What should they build?

A. A list – scan for the destination.

B. A hash table – look up the destination.

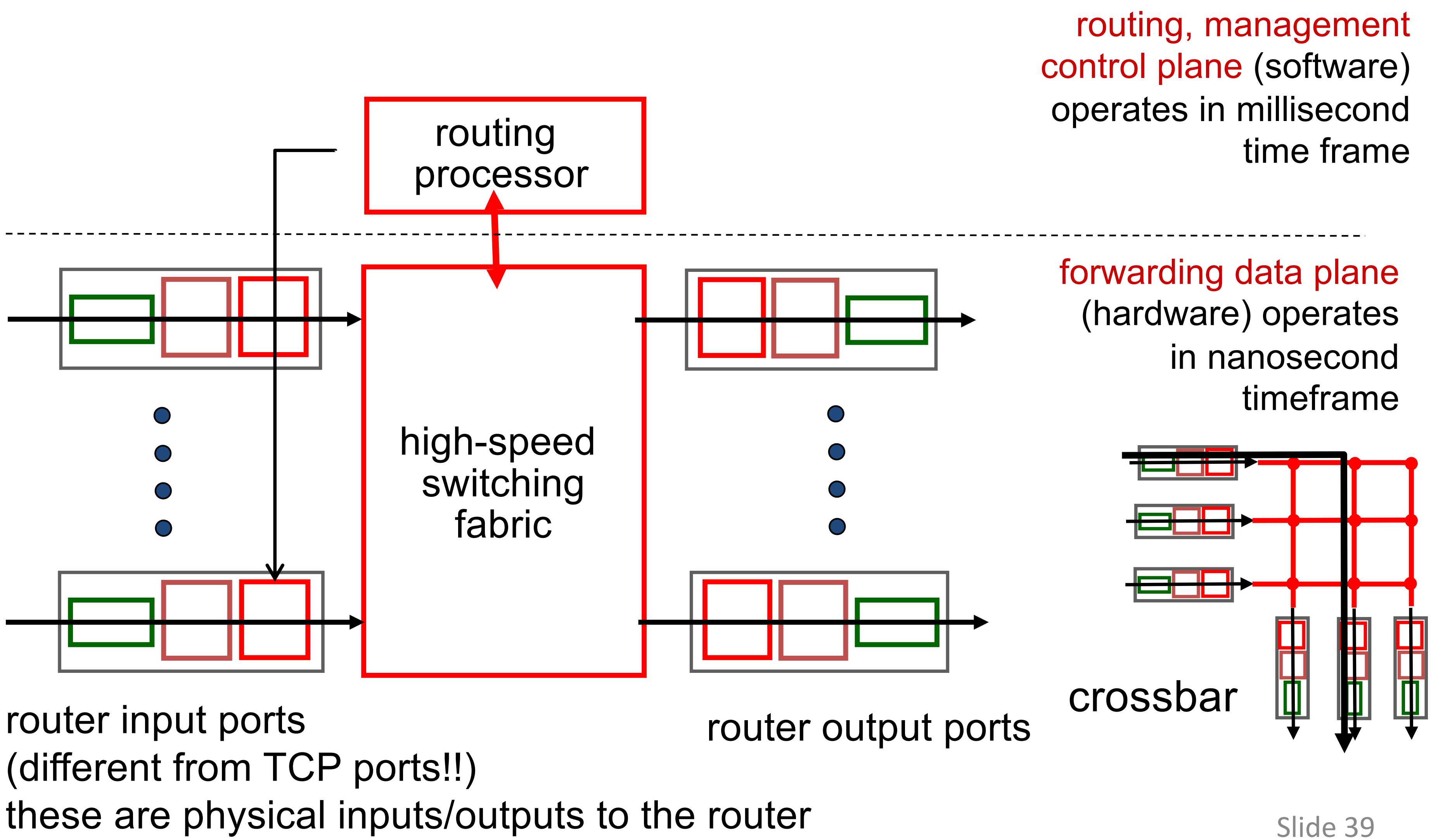
C. A tree – Follow branches that lead to the destination.

D. Some other software structure.

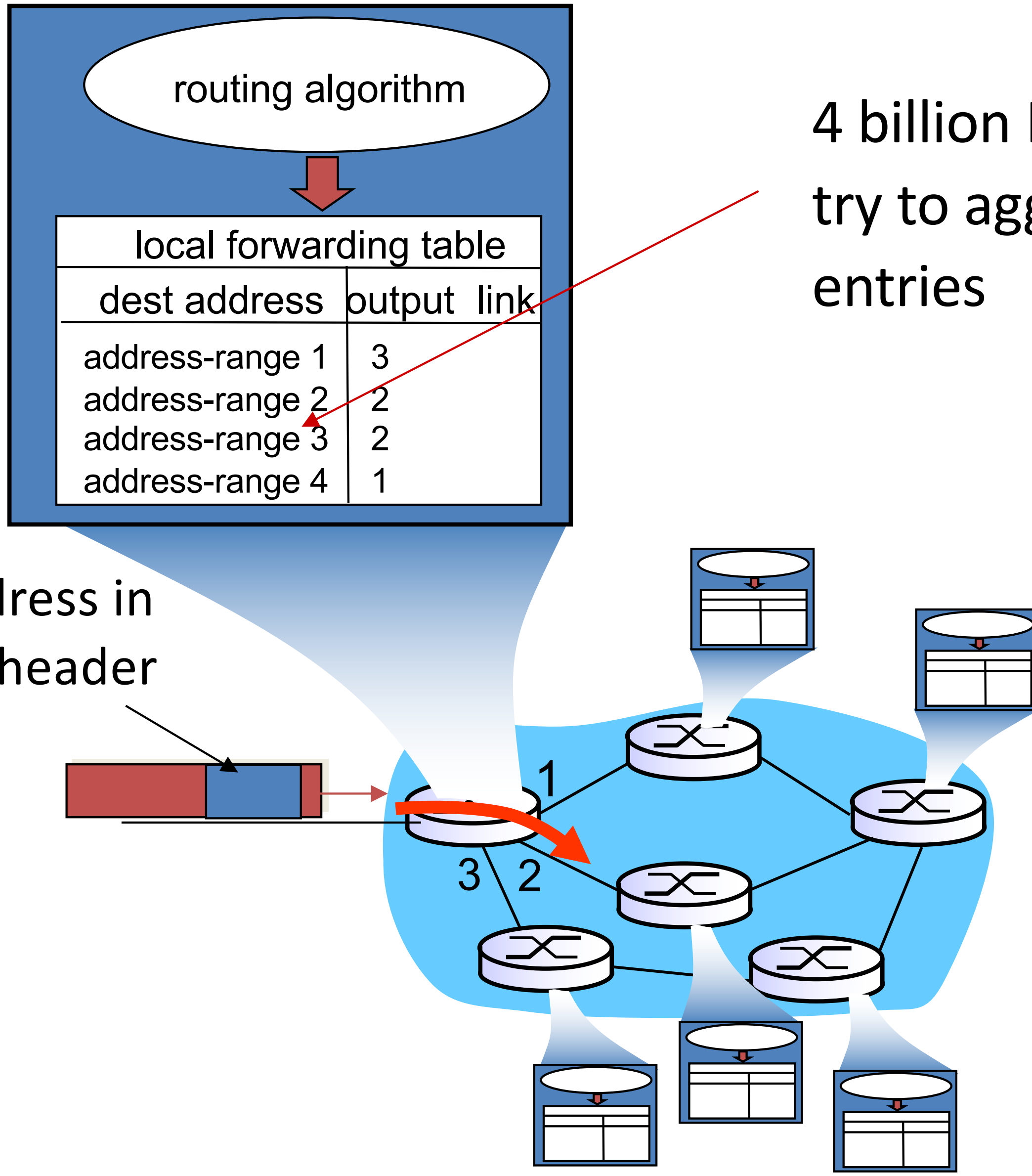
E. We can't do this in software, we need special hardware.

Aside: router architecture overview

- high-level view of generic router architecture:



Datagram forwarding table



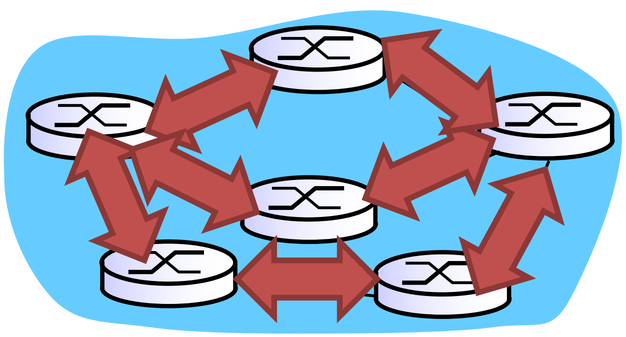
4 billion IP addresses, try to aggregate table entries

IP destination address in arriving packet's header

Routing

Traditional

- Routers run a routing protocol to exchange state.
- Use state to build up the forwarding table.



What services would we like a router to implement?

A. Basic connectivity: route packets to destination
B. Find policy-compliant paths (keep ISPs happy)
C. Traffic engineering
D. Impose limits on what can be accessed on the Internet vs. local ISP
E. All of the above

What services would we like a router to implement?

A. Basic connectivity: route packets to destination (implemented today)
B. Find policy-compliant paths (keep ISPs happy) (implemented today)
C. Traffic engineering
D. Impose limits on what can be accessed on the Internet vs. local ISP
E. All of the above

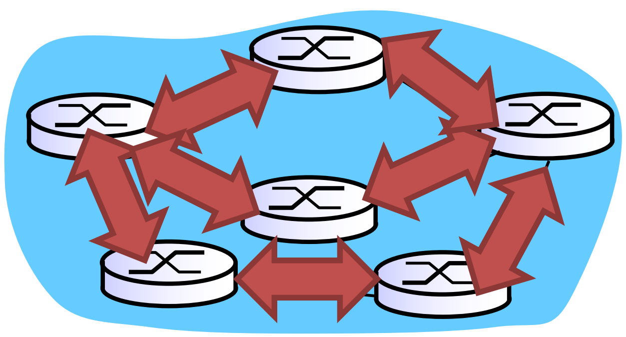
Nice things to have..

- Traffic engineering:
  - Want to avoid persistent overloads on links
  - Choose routes to spread traffic load across links
- Access Control:
  - Limit access to backend database machines.
  - Firewalls
- Network measurement

Routing

<u>Traditional</u>

- Routers run a routing protocol to exchange state.
- Use state to build up the forwarding table.

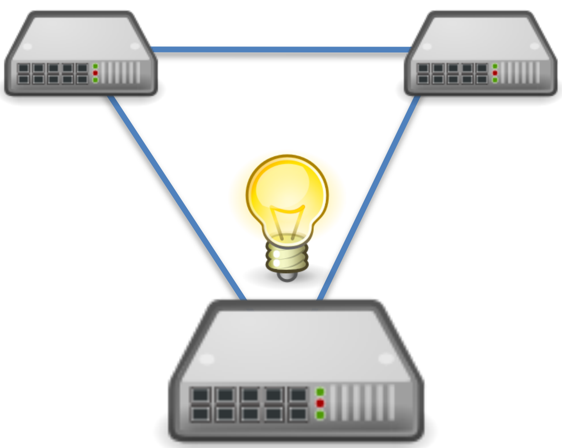


<u>Software-Defined</u>

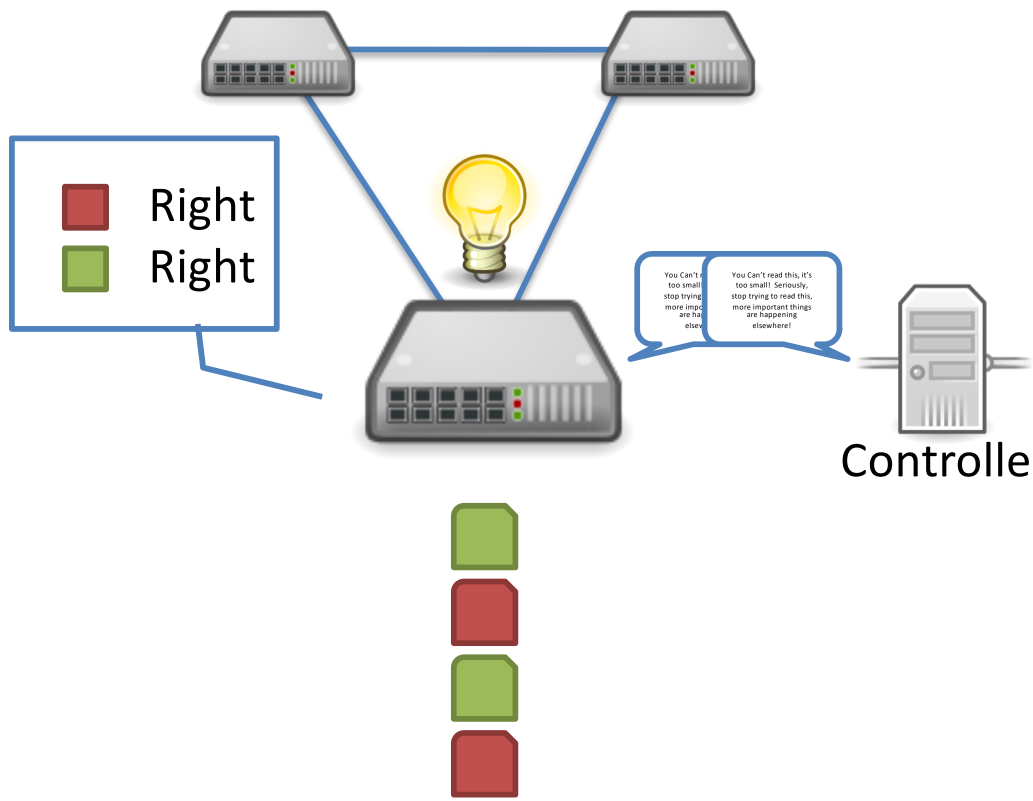
- Routers are dumb, just do what they're told.
- Controller service explicitly tells each router what to do.
- Rare on the Internet, hot topic in data centers.

Software-Defined Networking (SDN)

Traditional Hardware



SDN Hardware



Summary

- On the Internet, best-effort packet switching is the norm
- Forwarding: move packets from router's input to appropriate router output: Look up in a table
- Routing: determine route taken by packets from source to destination: Populating the table
- Hardware helps with quick forwarding using longest prefix matching.