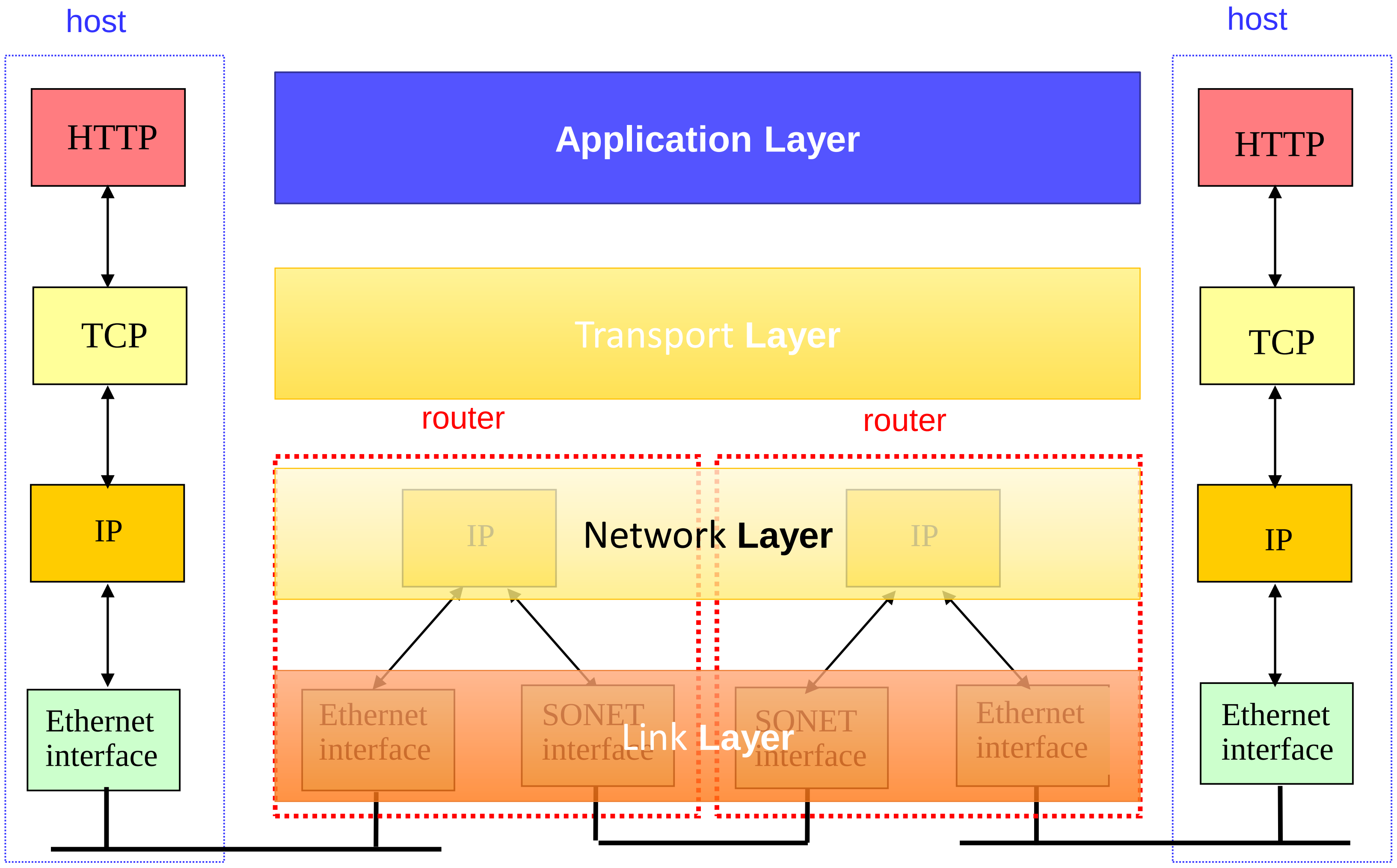
CS 43: Computer Networks
The Network Layer

Kevin Webb

Swarthmore College

March 24, 2022

TCP/IP Protocol Stack



Some background…

- 1968: DARPAnet/ARPAnet (precursor to Internet)
  - (Defense) Advanced Research Projects Agency Network
  - Bob Taylor, Larry Roberts create program to build first wide-area packet-switched network
- Mid 1970’s: new networks emerge
  - SATNet, Packet Radio, Ethernet
  - All “islands” to themselves – didn’t work together
- Big question: how to connect these networks?

Internetworking

- Cerf & Kahn, in 1974,
  "A Protocol for Packet Network Intercommunication"
  - Foundation for the modern Internet

- Routers forward packets from source to destination
  - May cross many separate networks along the way

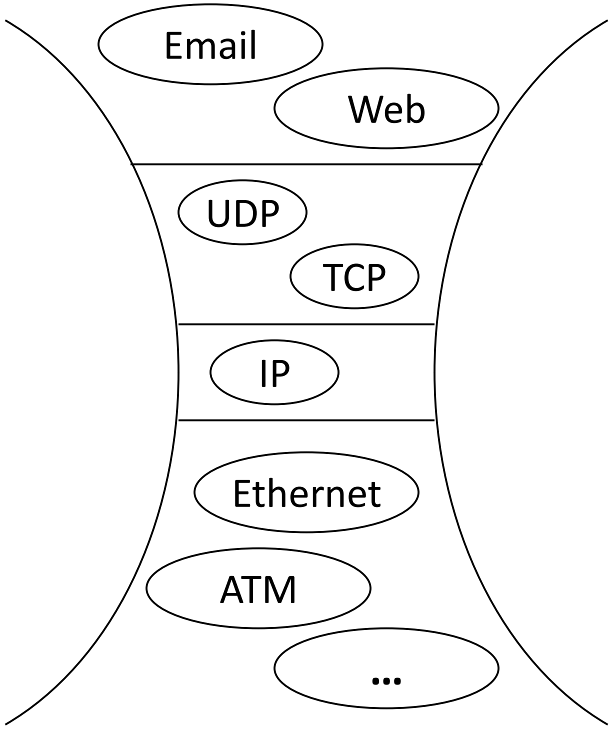
- All packets use a common Internet Protocol
  - *Any* underlying data link protocol
  - *Any* higher layer transport protocol

DARPAnet Primary Goal: Connect Stuff

- “Effective technique for multiplexed utilization of existing interconnected networks” – David Clark (1988)
  - **Minimal** assumptions about underlying networks
    - No support for broadcast, multicast, real-time, reliability
    - Extra support could actually get in the way
  - Packet switched, store and forward
    - Matched application needs, nets already packet switched
    - Enables **efficient resource sharing**/high utilization
  - “Gateways” interconnect networks
    - Routers in today’s nomenclature

Internet Protocol Stack

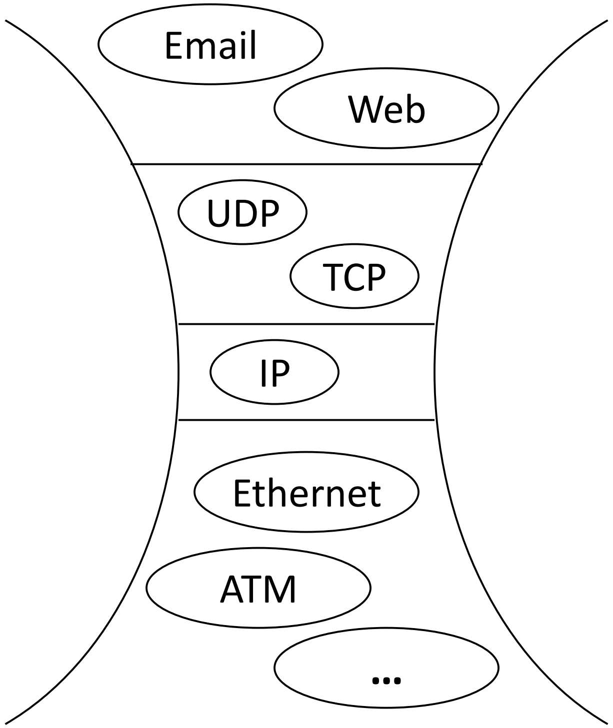
- Application: Email, Web, …
- Transport: TCP, UDP, …
- Network: IP
- Link: Ethernet, WiFi, ATM, …
- Physical: copper, fiber, air, …



- “Hourglass” model, “thin waist”, “narrow waist”

Internet Protocol Stack

- This should seem weird.
- *Everyone* uses IP?



- "Hourglass" model, "thin waist", "narrow waist"

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.

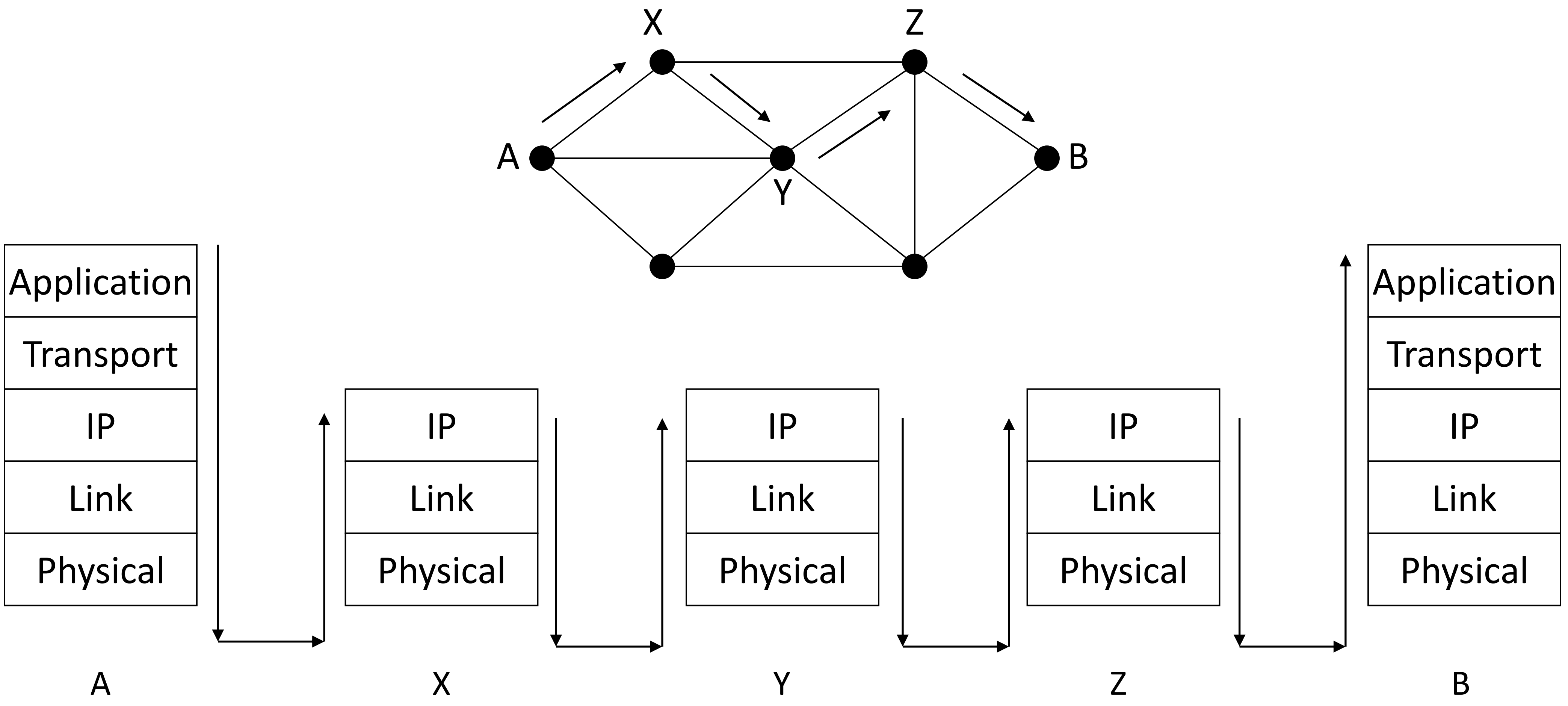
B. Route in advance, forward when a packet arrives.

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

D. Do both in advance.

E. Some other combination

Example of Internet Routing

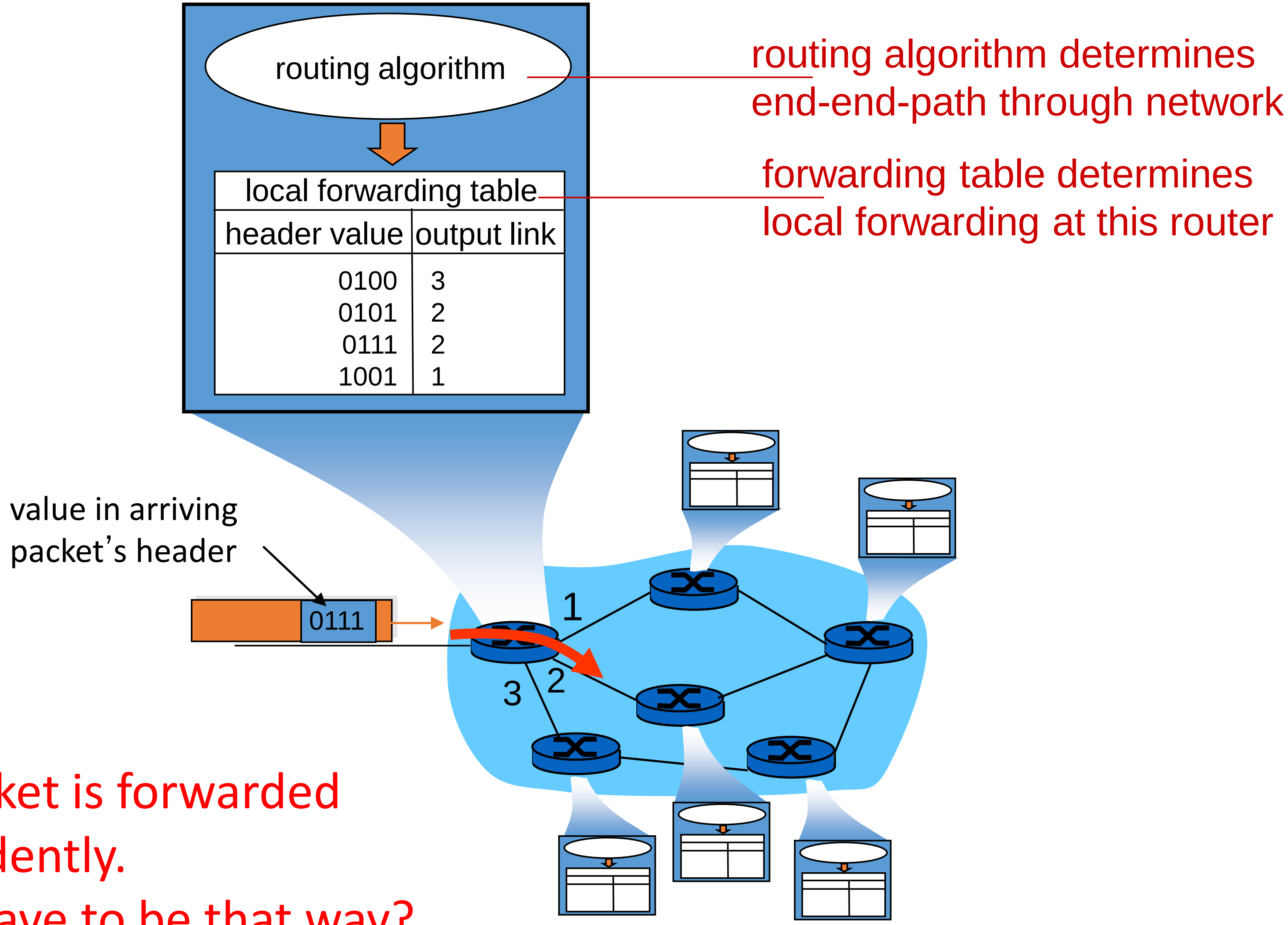


Network layer involved at every hop along the path.

Network Layer Functions

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

Interplay between routing and forwarding

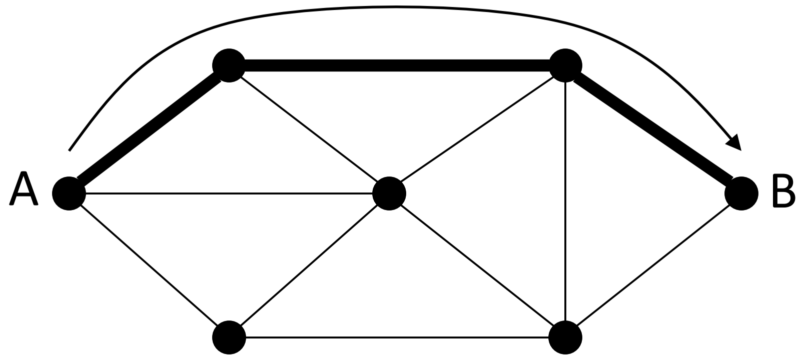


Each packet is forwarded independently.

Does it have to be that way?

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.

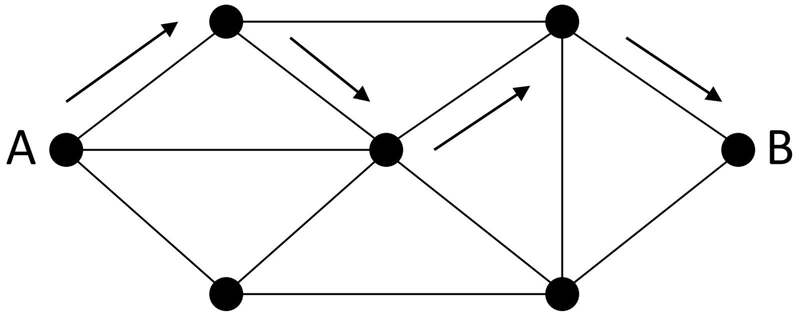
D. It requires too many resources.

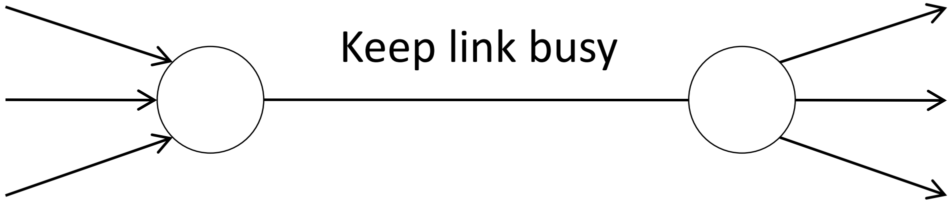
E. Some other reason.

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



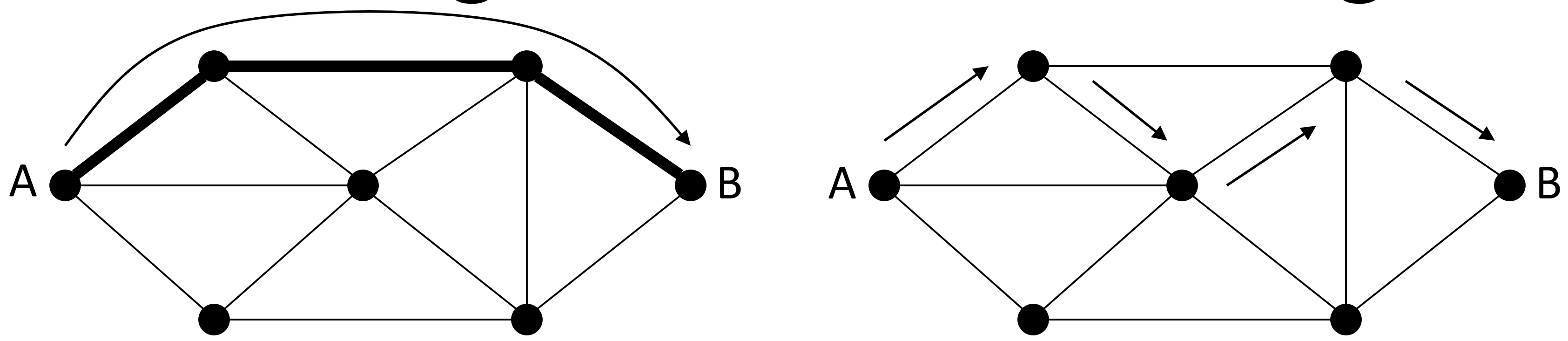


Which of the following is/are generally true of packet vs. circuit switching?

1. Packet switching has less variance in performance.
2. Circuit switching is less 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

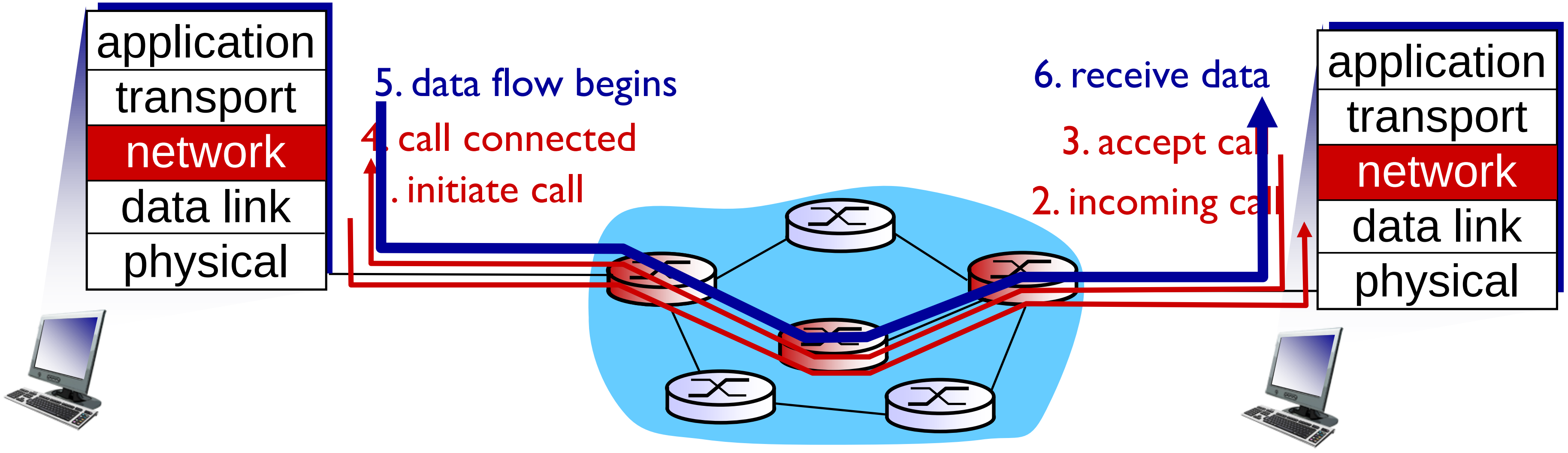
Datagram vs. “Virtual Circuit”

- *Datagram* network provides network-layer *connectionless* service (packet switching)

- *Virtual-circuit* network provides network-layer *connection* service (like circuit switching)

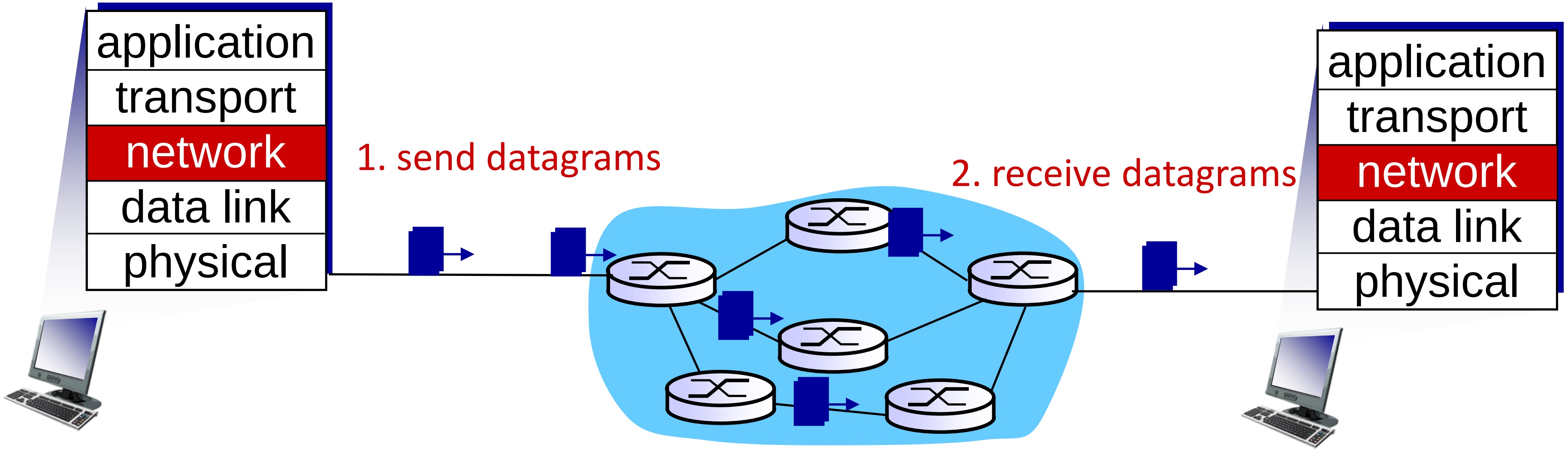
Virtual circuits: Signaling Protocols

- Used to setup, maintain, teardown VC
- Used in ATM, 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



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.

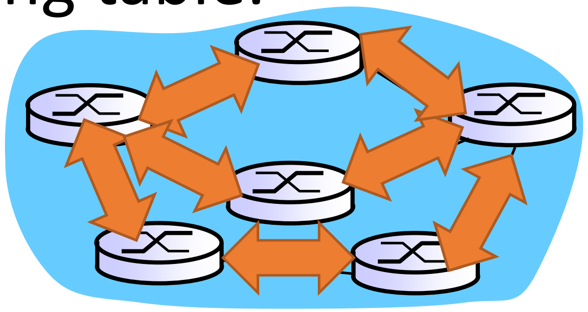
D. Some other mechanism.

Routing

Traditional

- Routers run a **routing protocol** to exchange state.

- Use state to build up the forwarding table.



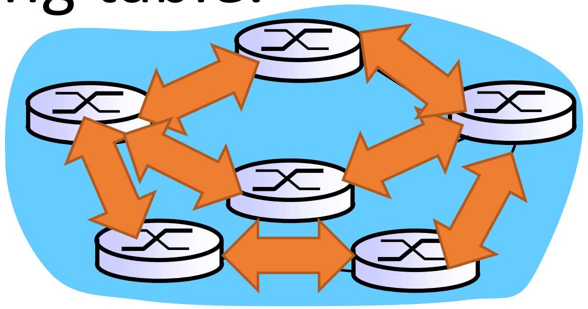
Assume this is the type of routing we're talking about unless we explicitly say otherwise!

Routing

Traditional

- Routers run a **routing protocol** to exchange state.

- Use state to build up the forwarding table.



"Software-Defined"

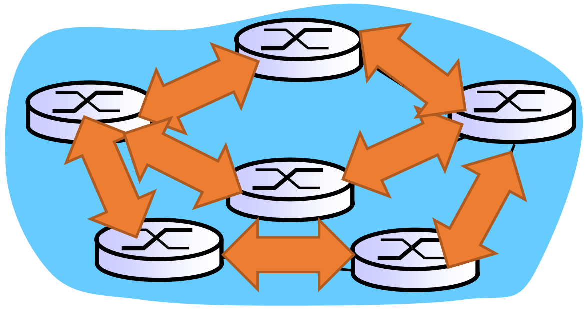
- Routers are dumb, just do what they're told.

- Controller service explicitly tells each router what to do.

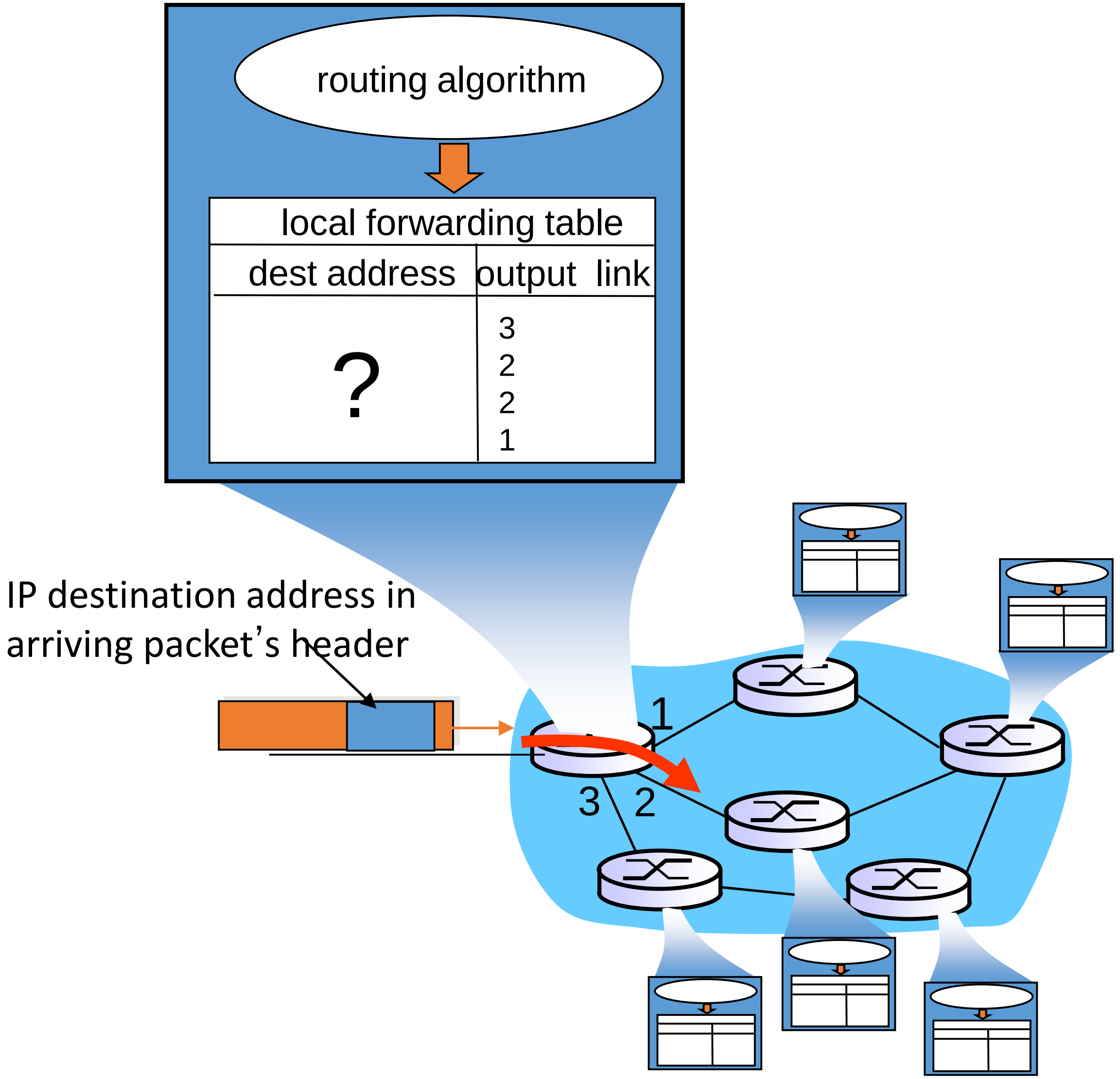
- Rare on the Internet, hot topic in data centers.

Datagram Forwarding

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



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.

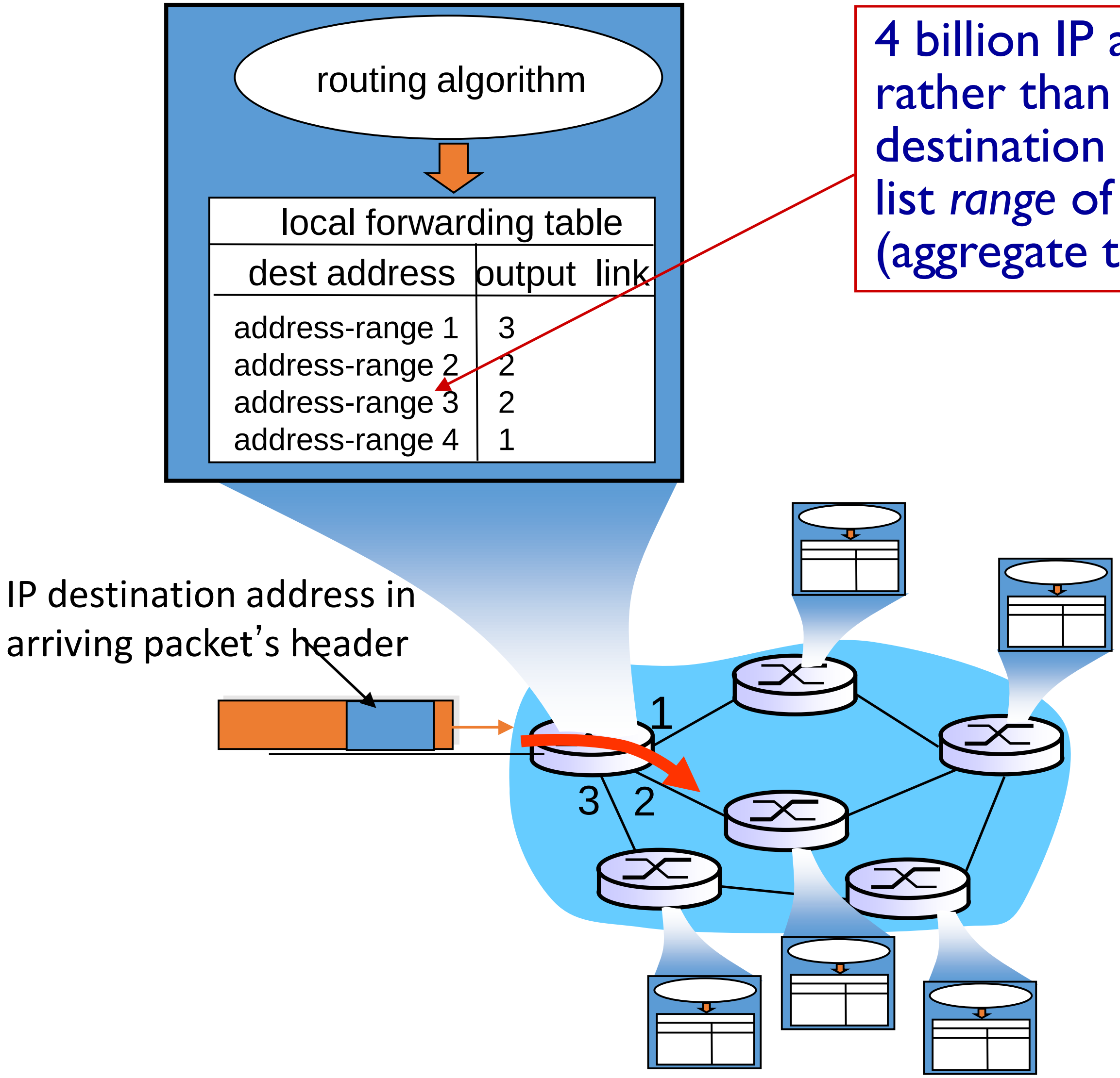
B. A hash table – look up the destination.

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.

Datagram forwarding table



4 billion IP addresses, so rather than list individual destination address list *range* of addresses (aggregate table entries)

Datagram forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
Otherwise (default gateway)	3

Q: but what happens if ranges don't divide up so nicely?

Longest prefix matching

longest prefix matching

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
Otherwise (default gateway)	3

examples:

DA: 11001000 00010111 00010110 10100001 which interface?

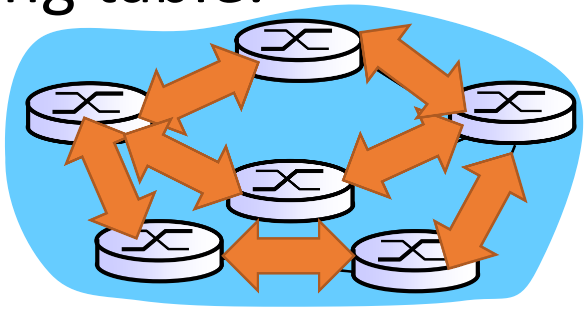
DA: 11001000 00010111 00011000 10101010 which interface?

Routing

Coming up in ~1 week.

Traditional

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

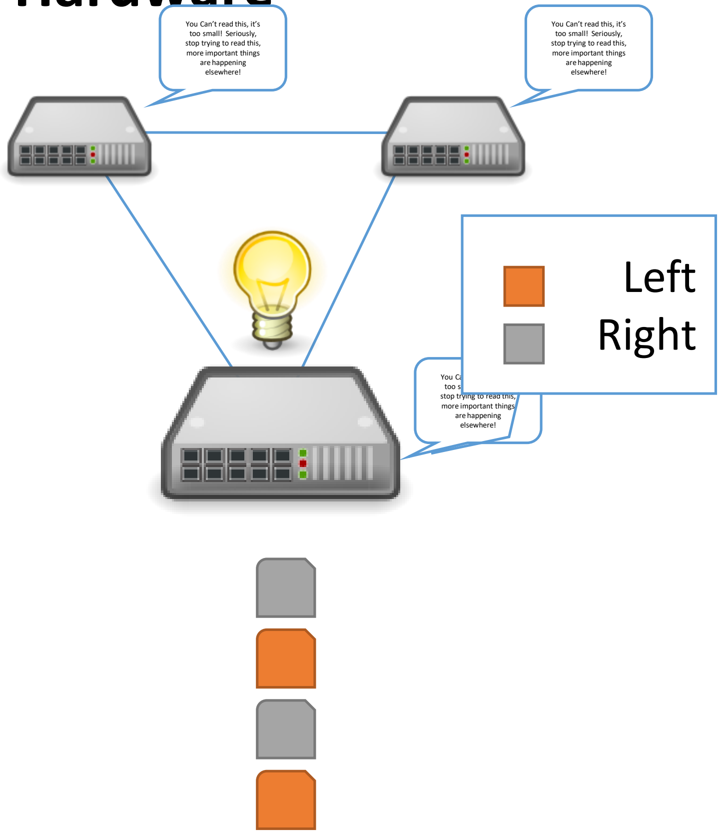


"Software-Defined"

- 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.

Software-Defined Networking (SDN)

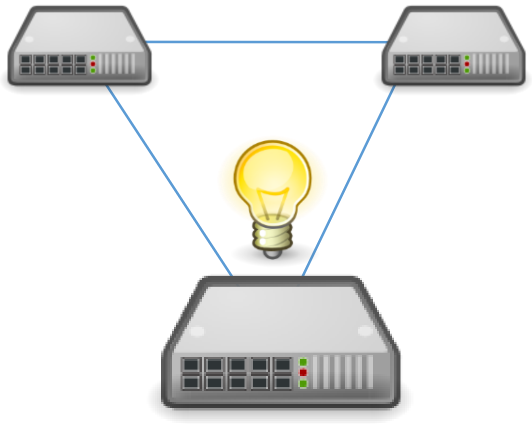
Traditional Hardware



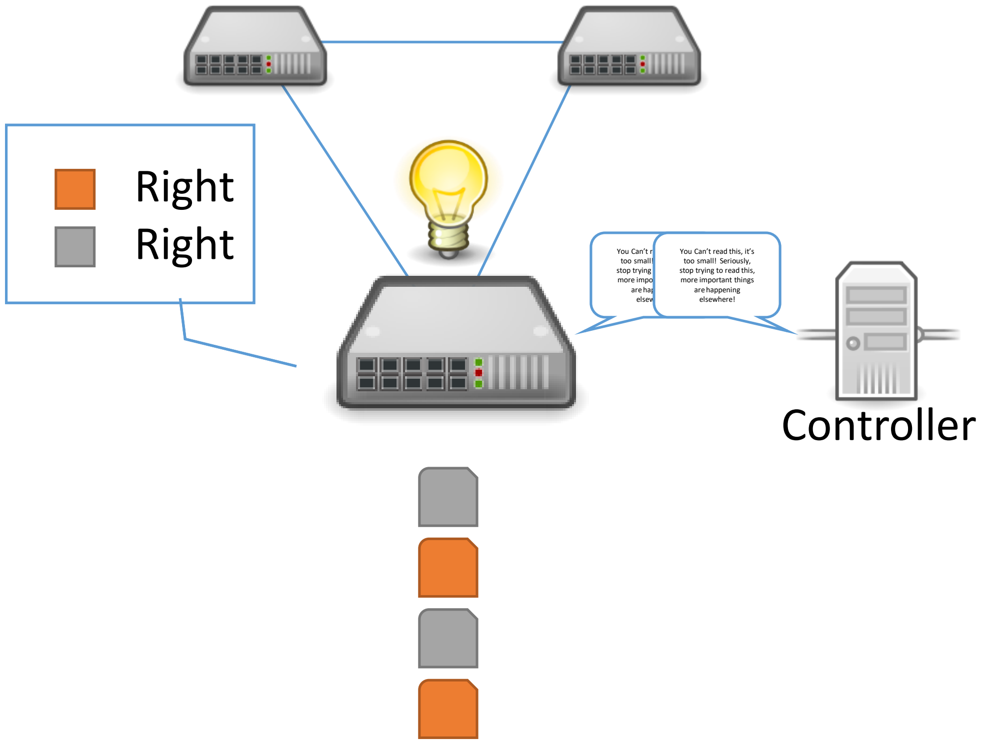
SDN Hardware

Software-Defined Networking (SDN)

Traditional Hardware



SDN Hardware



Summary

- Forwarding: moving packet from one interface to another (table lookup)
- Routing: Populating the table in advance
- On the Internet, best effort packet switching is the norm
- Hardware helps with quick forwarding using longest prefix matching