CS 43: Computer Networks

The Network Layer

November 9, 2026



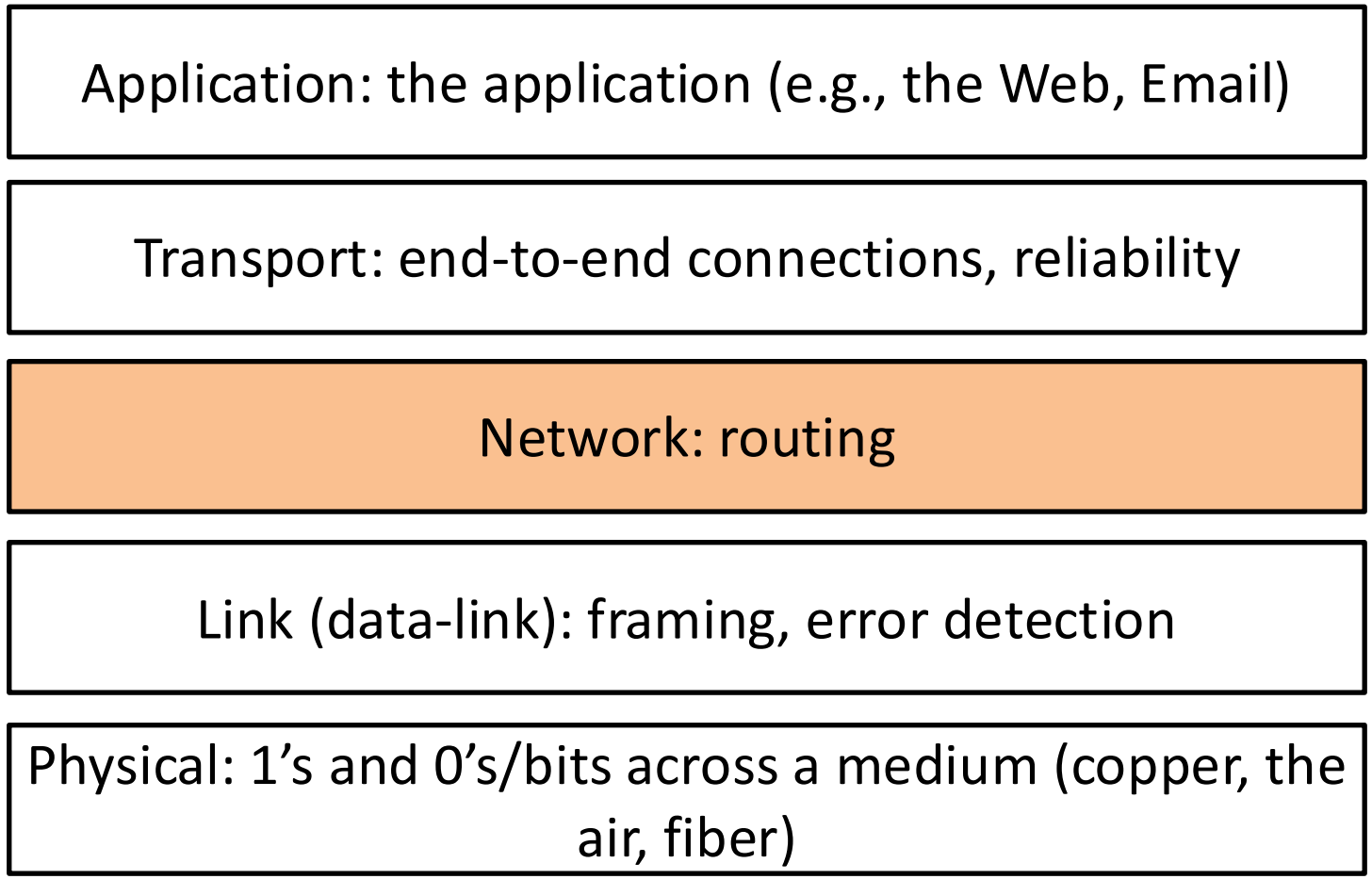
Reading Quiz
(graded for correctness)

Put devices away

Note: Reading quiz slides will have an orange border.

End of Reading Quiz!

The Network Layer!



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

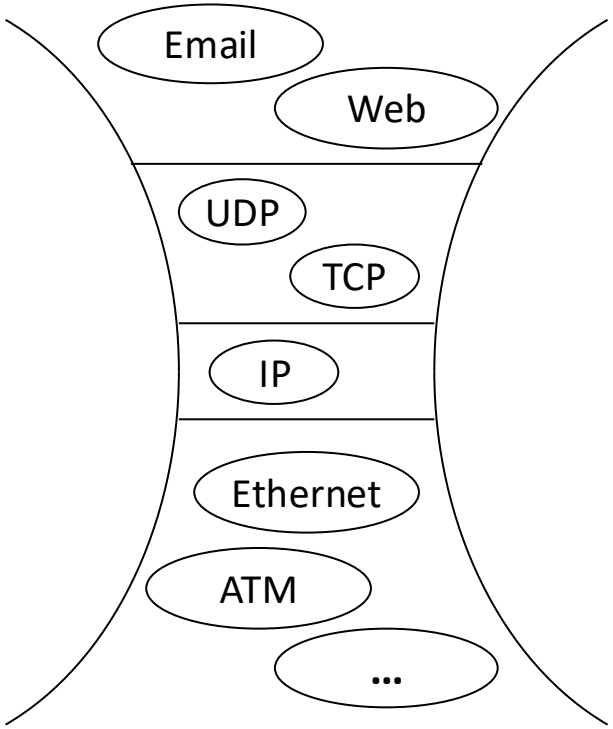
- Cerf & Kahn, in 1974,
  "*A Protocol for Packet Network Intercommunication*"
  - Foundation for the modern Internet

- 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

Internet Protocol Stack

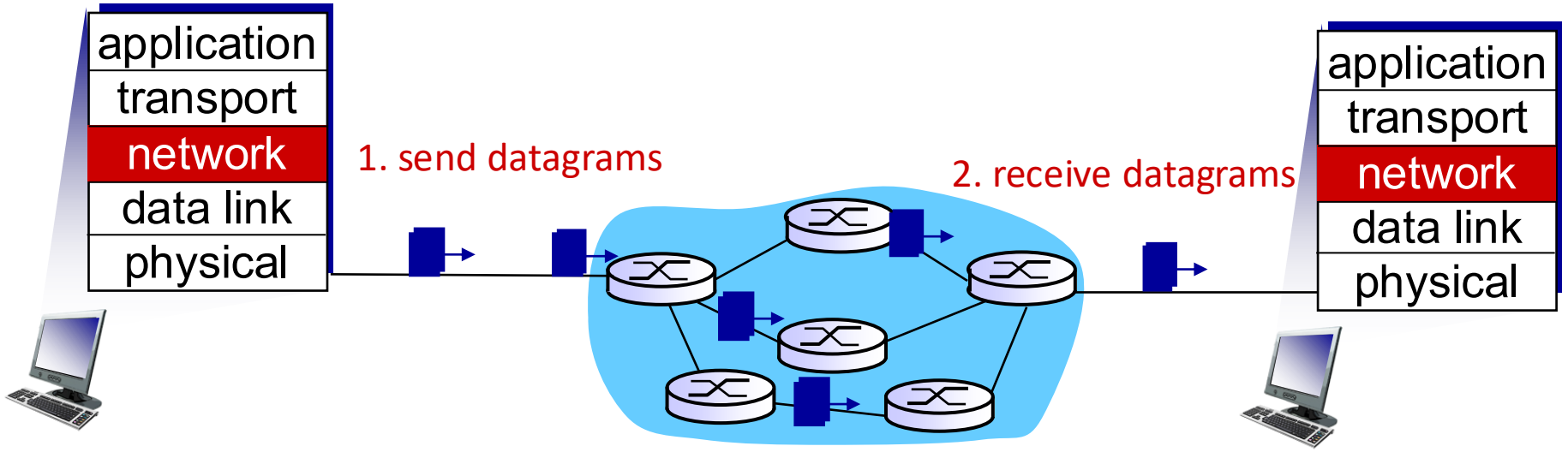
- Application: Email, Web, …
- Transport: TCP, UDP, …
- **Network: IP**
- Link: Ethernet, WiFi, ATM, …
- Physical: copper, fiber, air, …



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

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

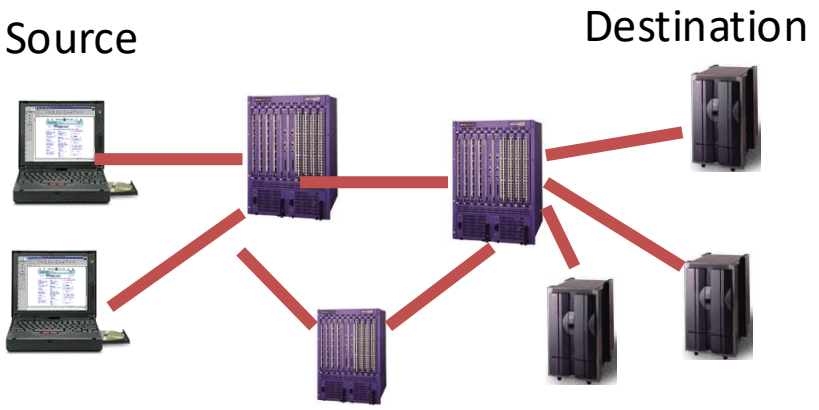
Serve Cached Data

Add Security

Provide reliability

Migrate connections

.... the list is long



The End-to-End Principle

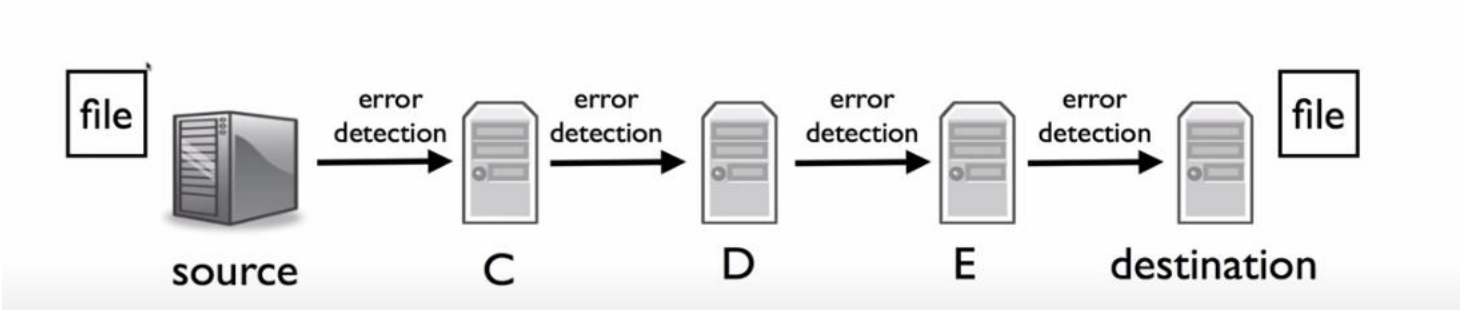
"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system."

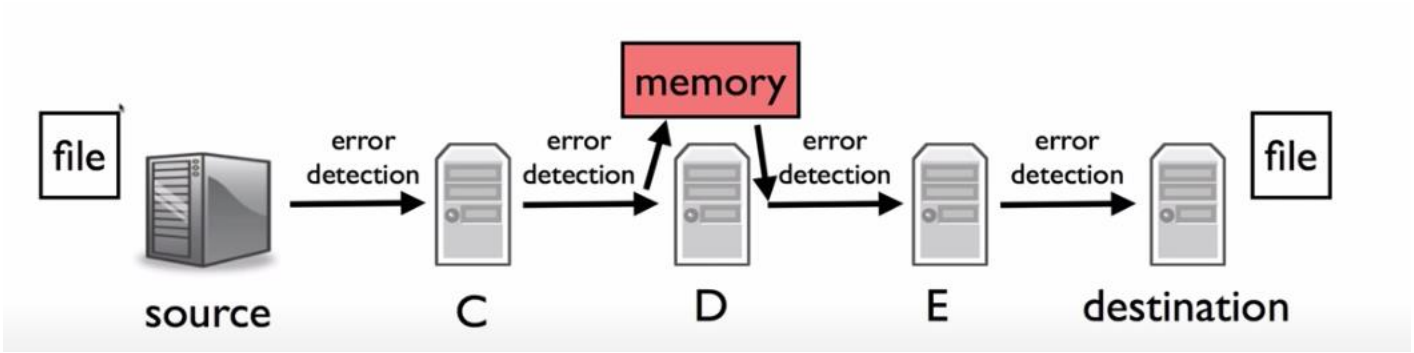
- Saltzer, Reed and Clark

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.

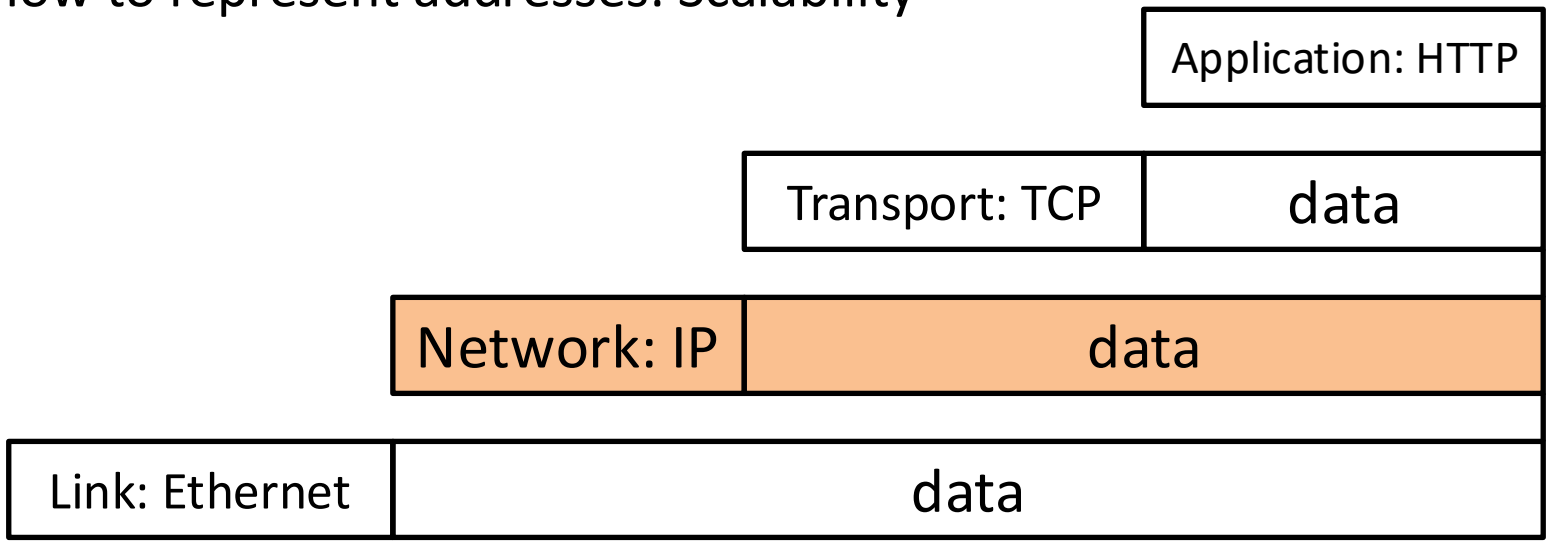
-RFC 1958

Courtesy: Stanford

Network Layer

- Function: Route packets end-to-end on a network, through multiple hops

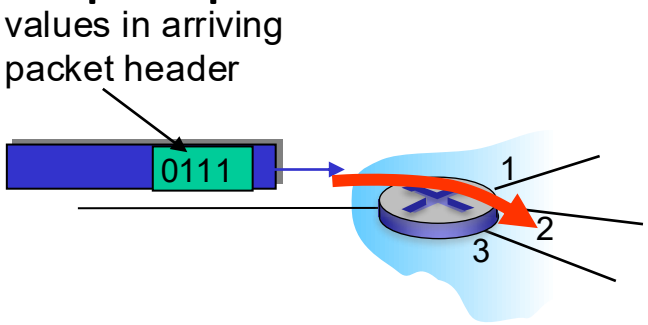
- Key challenge
  - How to route packets: Convergence
  - How to represent addresses: Scalability



Network layer: data plane, control plane

Data plane:

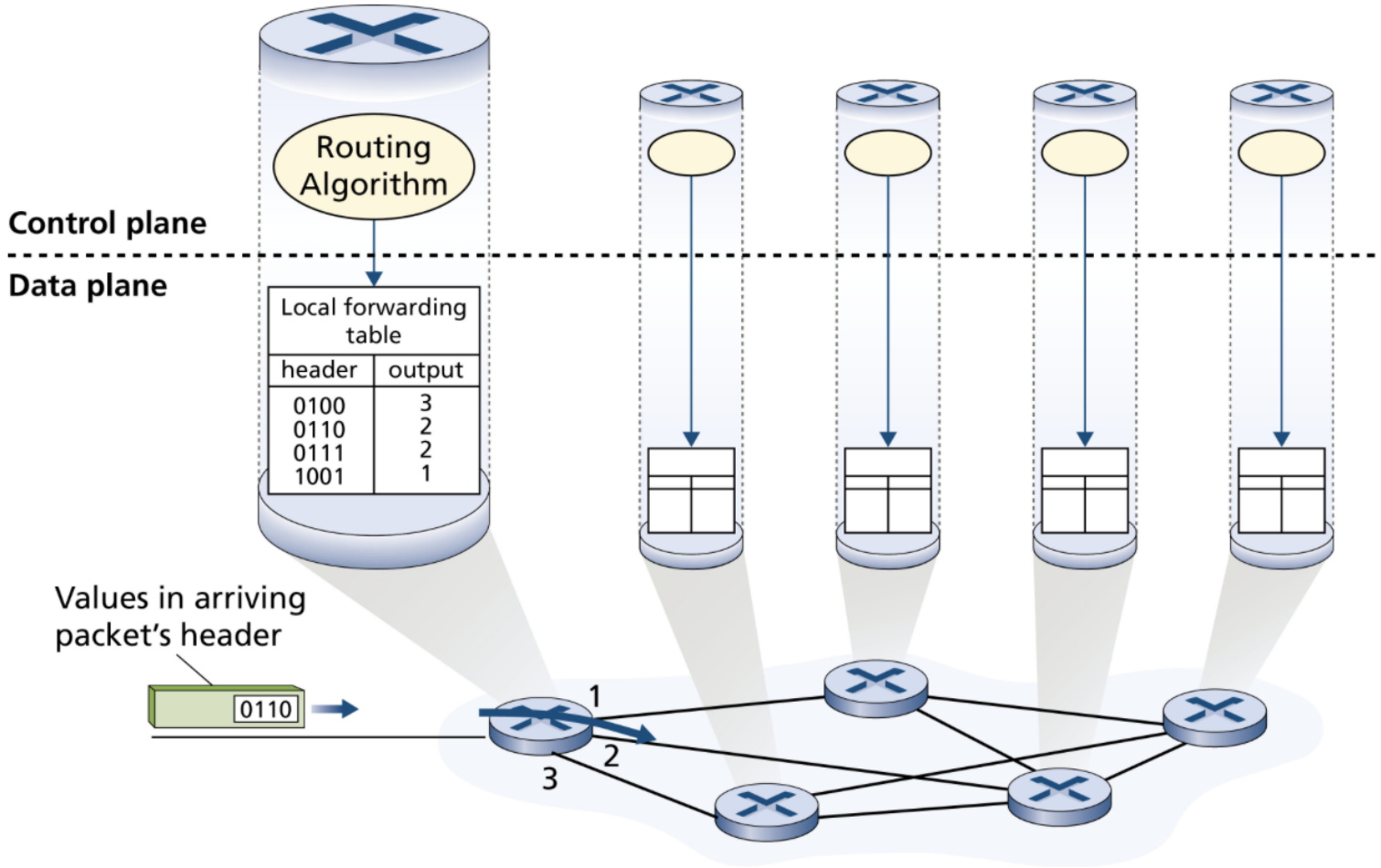
- *local*, per-router function
- determines how datagram arriving on router input port is forwarded to router output port



Control plane

- *network-wide* logic
- determines how datagram is routed among routers along end-end path from source host to destination host

Per-router control plane: Individual routing algorithm components in each and every router interact in the control plane



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. I don’t know

Discussion: 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. I don’t know



When should a router perform routing? Forwarding?

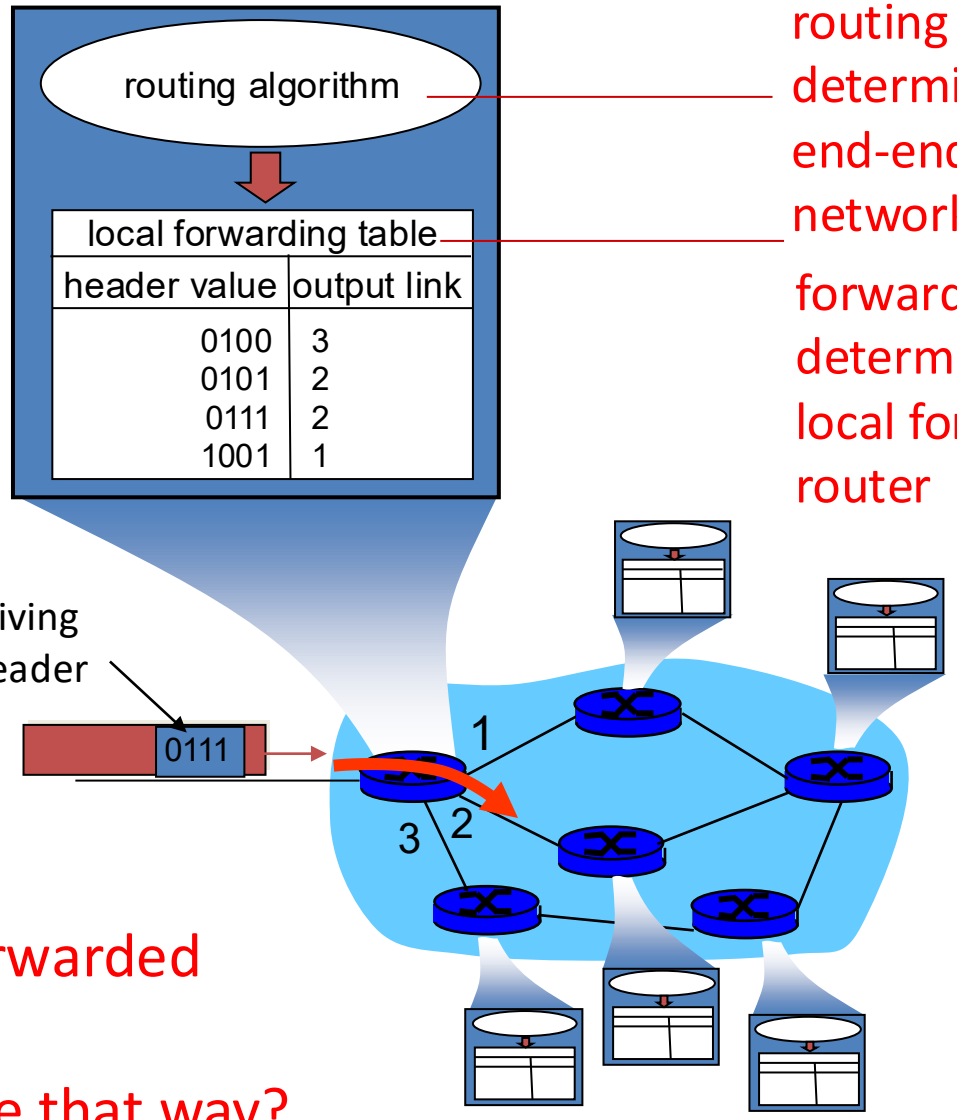
Route in advance, forward when a packet arrives.

- 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



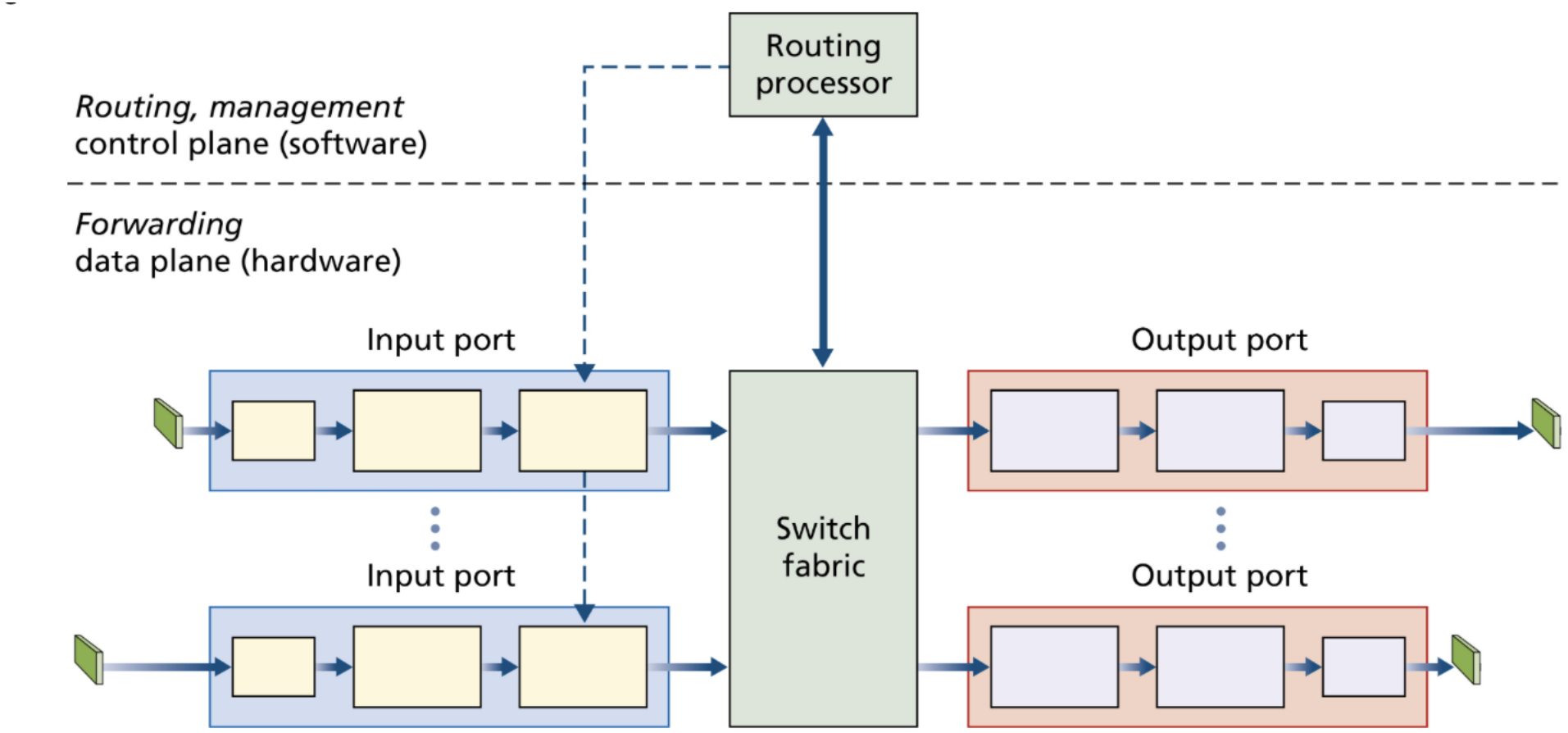
routing algorithm determines end-end-path through network

forwarding table determines local forwarding at this router

Each packet is forwarded independently.

Does it have to be that way?

Router architecture overview



Which of the following is an example of routing rather than forwarding?

A. A router looks up a packet's destination address in its table and sends it out port 2
B. A router learns from a neighbor that a link has failed and changes the output port listed for a destination
C. A router drops a packet because the queue on its output port is full
D. A router moves a packet from an input port to an output port through the switching fabric
E. I don't know

Discussion: Which of the following is an example of routing rather than forwarding?

A. A router looks up a packet's destination address in its table and sends it out port 2
B. A router learns from a neighbor that a link has failed and changes the output port listed for a destination
C. A router drops a packet because the queue on its output port is full
D. A router moves a packet from an input port to an output port through the switching fabric
E. I don't know



Discussion: Which of the following is an example of routing rather than forwarding?

A. A router looks up a packet's destination address in its table and sends it out port 2
B. **A router learns from a neighbor that a link has failed and changes the output port listed for a destination**
C. A router drops a packet because the queue on its output port is full
D. A router moves a packet from an input port to an output port through the switching fabric
E. I don't know

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.

E. I don't know

Discussion: 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.

E. I don't know



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).

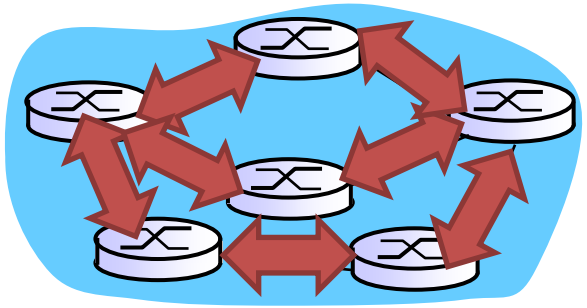
D. Some other mechanism.

Routing

Traditional

- Routers run a routing protocol to exchange state.

- Use state to build up the forwarding table.

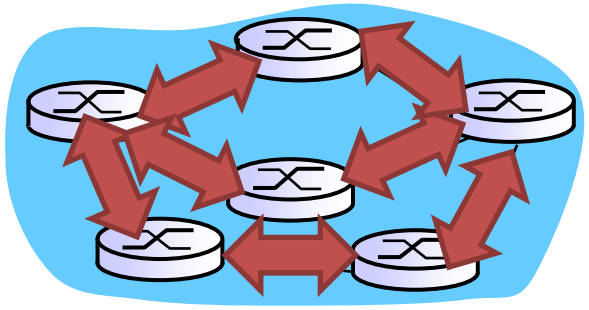


Assume this is the type of routing we're talking about unless we explicitly say otherwise!

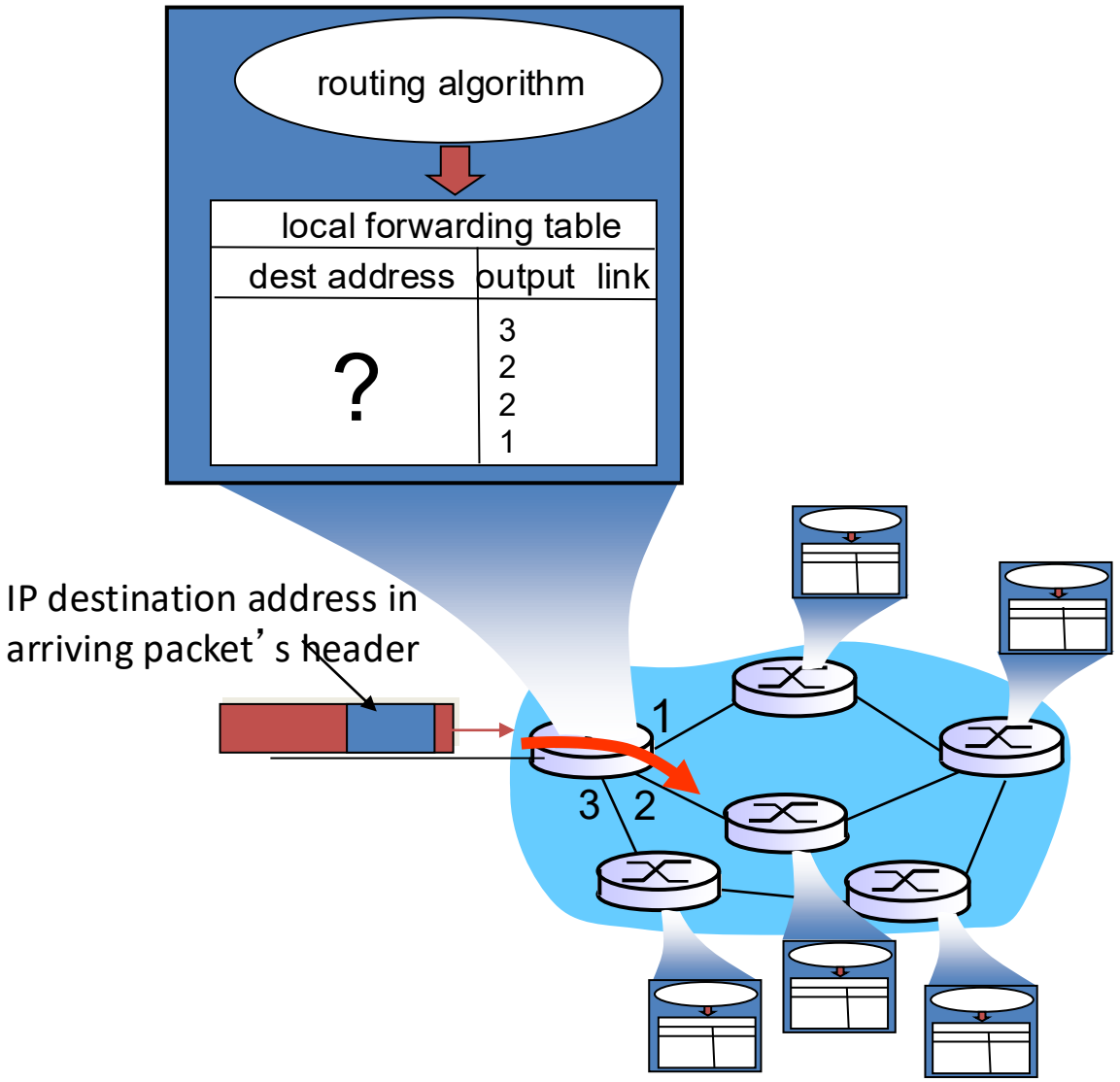
Datagram Forwarding

- Routers periodically exchange state.

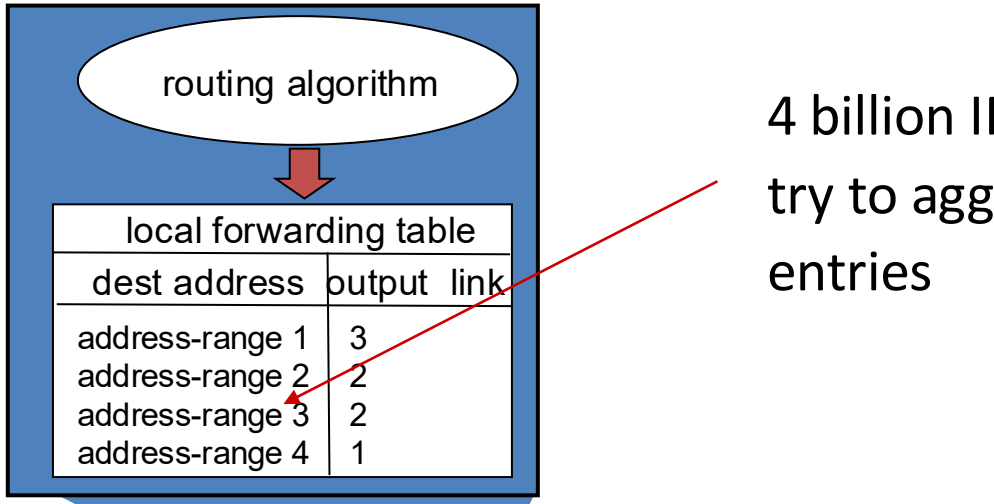
- Use the state to build a forwarding table (FIB – Forwarding Information Base)



Datagram forwarding table

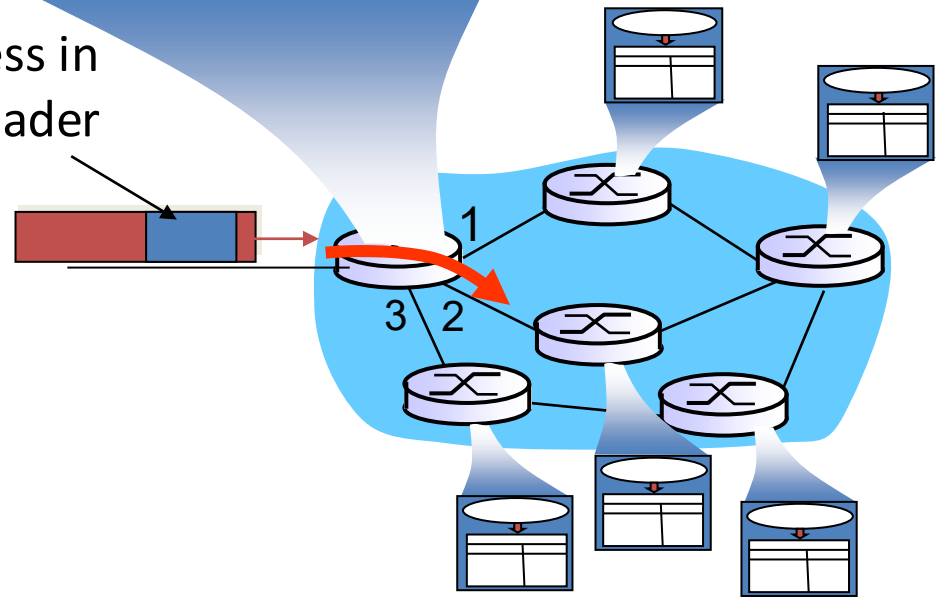


Datagram forwarding table



4 billion IP addresses, try to aggregate table entries

IP destination address in arriving packet's header



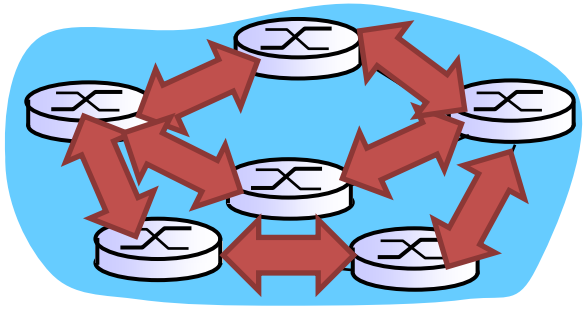
Nice things to have..

- 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

Traditional

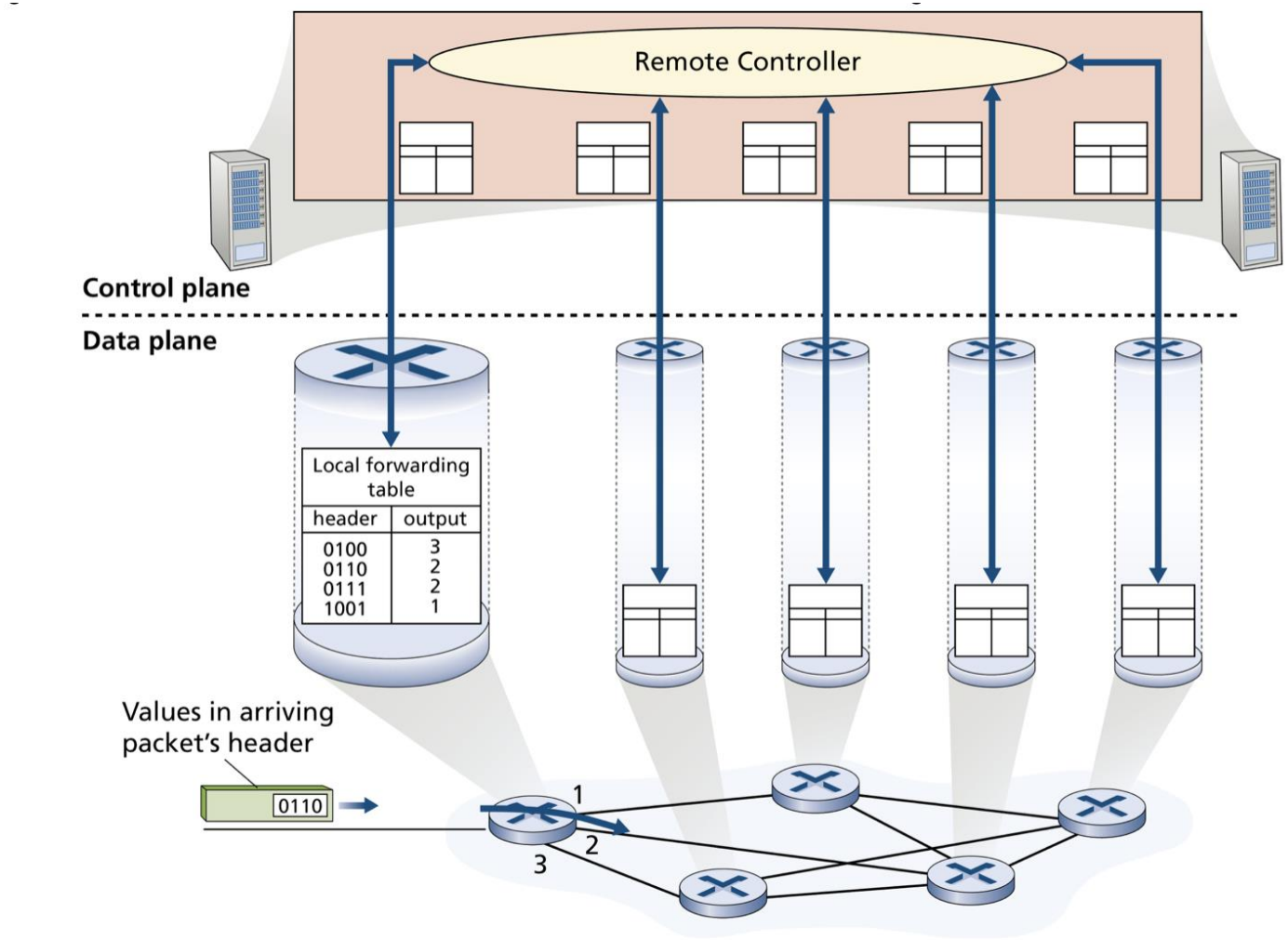
- Routers run a routing protocol to exchange state.
- Use state to build up the forwarding table.



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)



Summary

- Internet architecture: End-to-end principle and the ”thin-waist”
- 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

Reference Slides

Destination-based forwarding

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through	0
11001000 00010111 00010000 00000100 through 11001000 00010111 00010000 00000111	3
11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

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

Longest prefix matching

longest prefix match

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	3

examples:

11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

longest prefix match

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	3

match!

examples:

11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

longest prefix match

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	3

match!

examples:

11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

longest prefix match

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	3

match!

examples:

11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

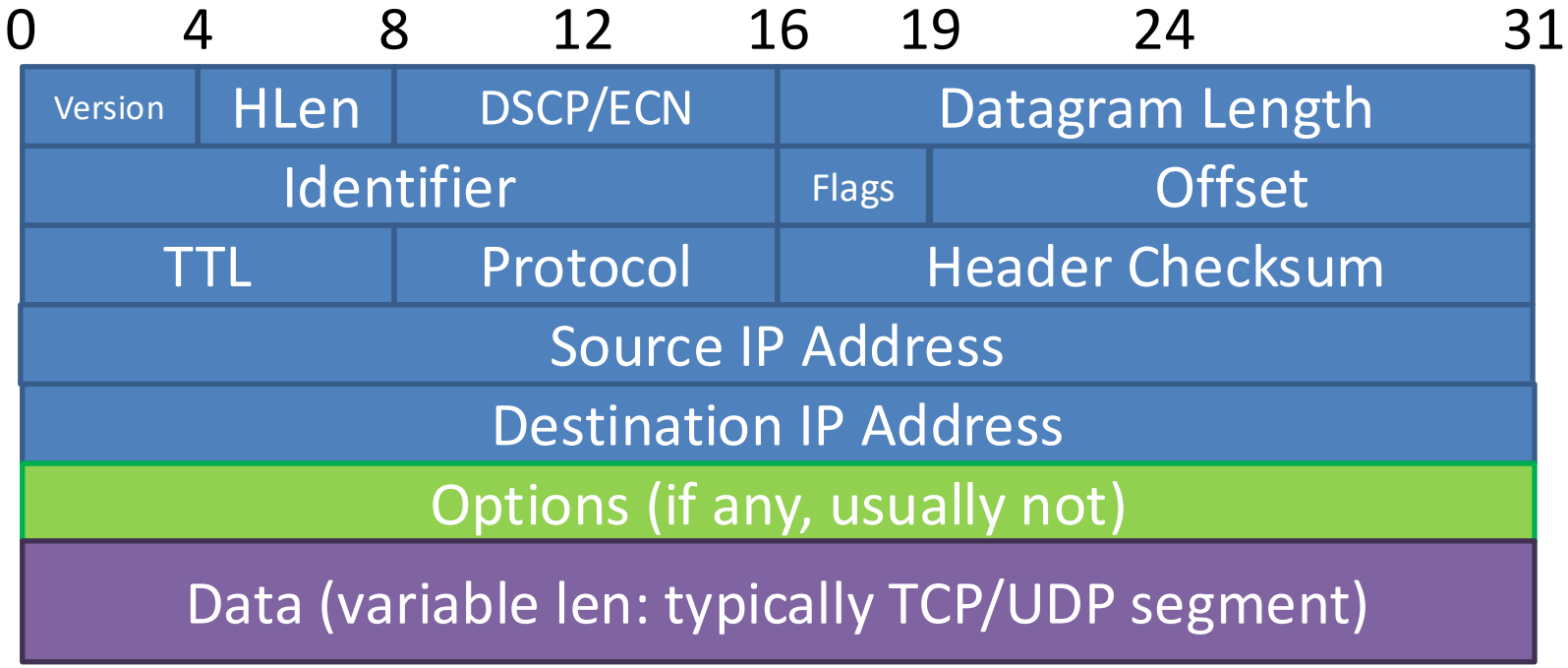
- we'll see *why* longest prefix matching is used shortly, when we study addressing
- longest prefix matching: often performed using ternary content addressable memories (TCAMs)
  - *content addressable:* present address to TCAM: retrieve address in one clock cycle, regardless of table size
  - Cisco Catalyst: ~1M routing table entries in TCAM

Network Layer Functions

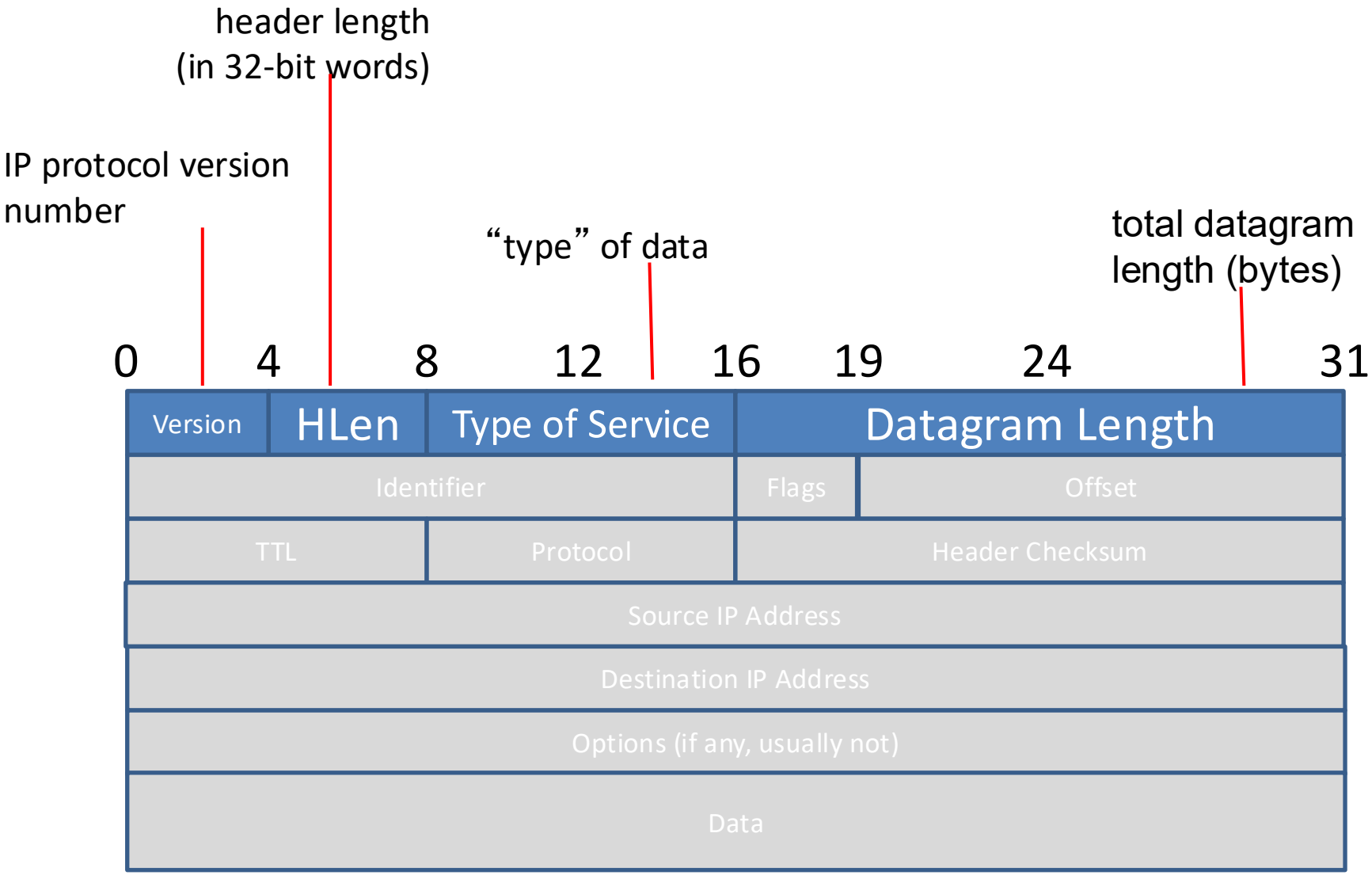
- Forwarding*:* move packets from router's input to appropriate router output
  - Look up in a table
- Routing: determine route taken by packets from source to destination.
  - Populating the table

IP Datagrams

- IP Datagrams are like a letter
  - Totally self-contained
  - Include all necessary addressing information
  - No advanced setup of connections or circuits

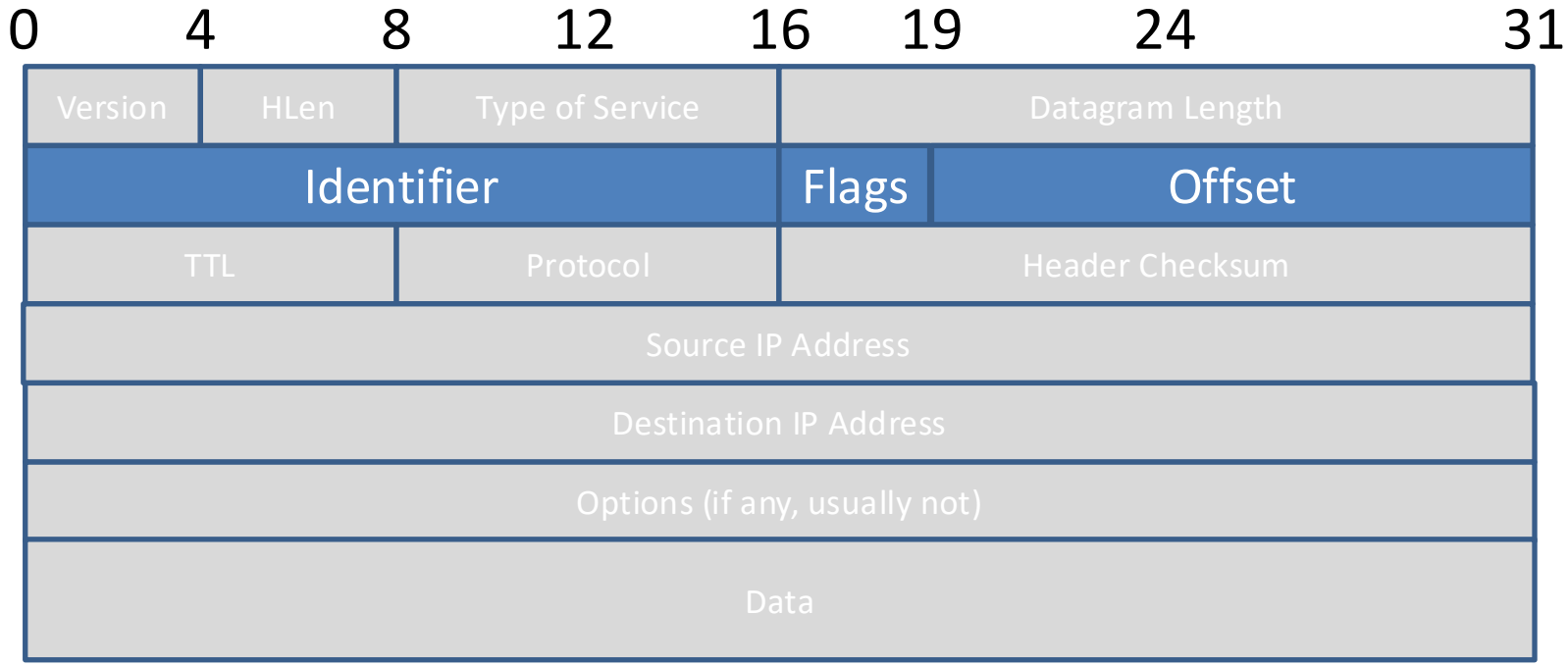


IP Datagram Format

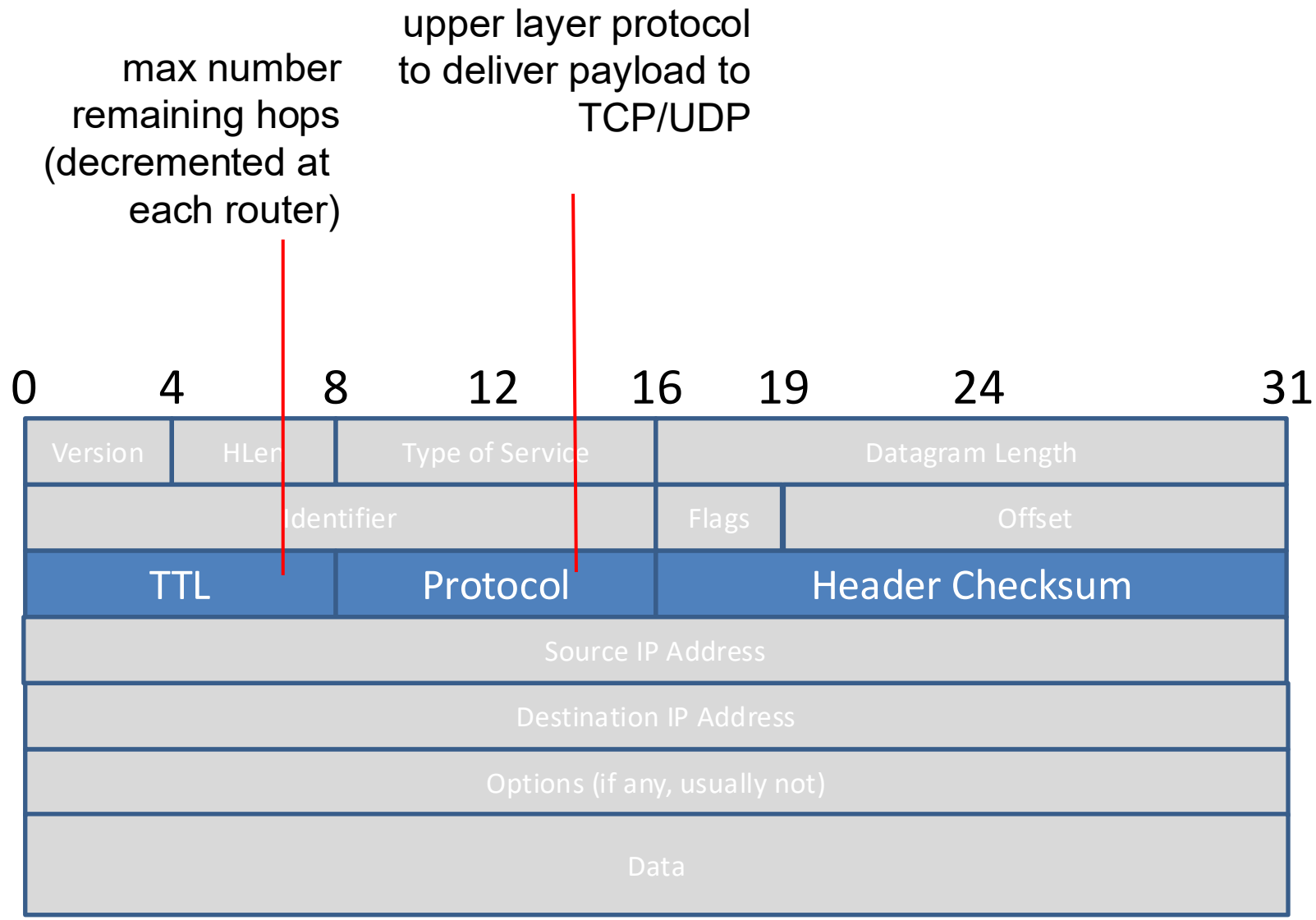


IP Datagram Format

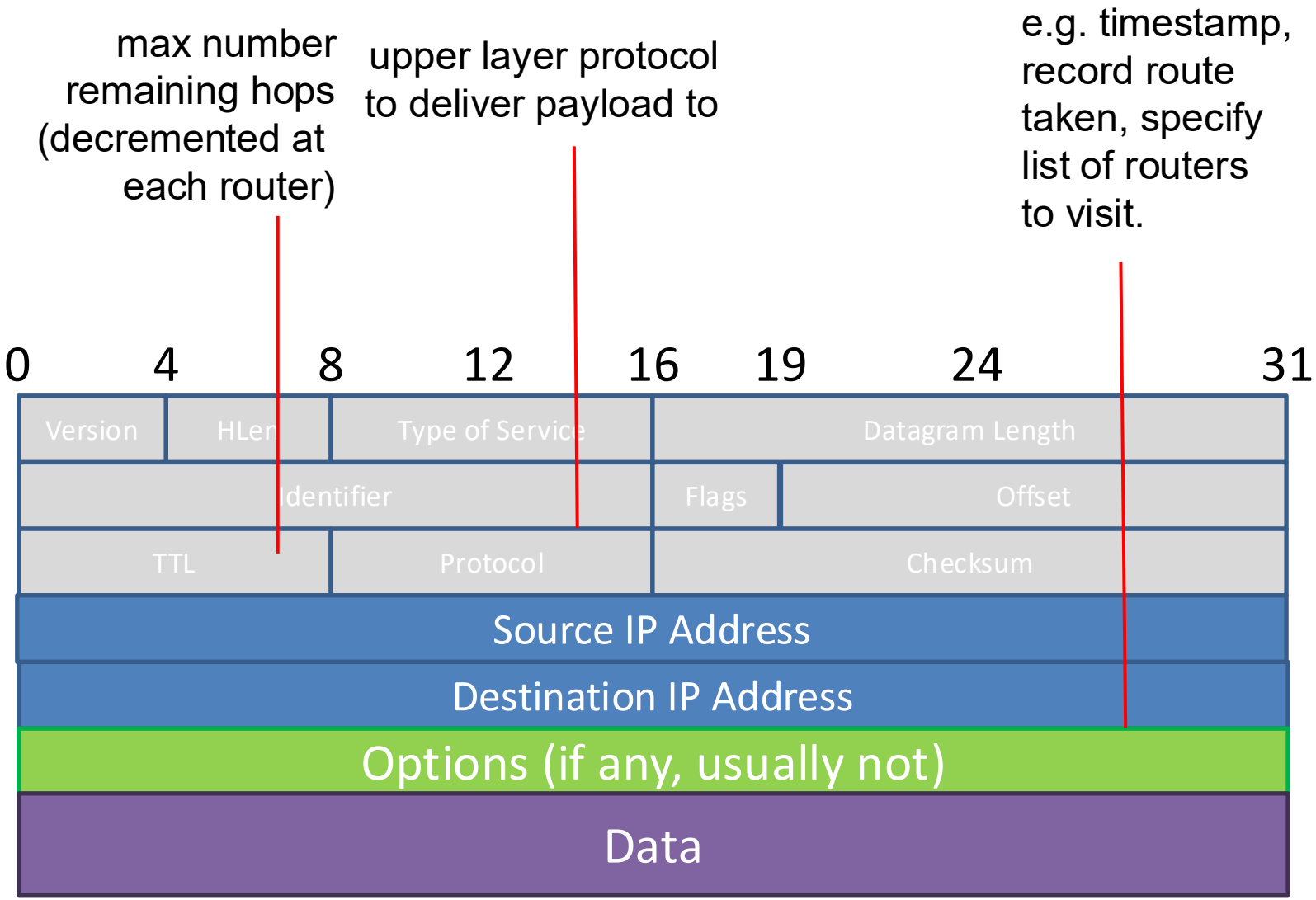
Fragmentation/ reassembly: Identifier, Flags, Offset



IP Datagram Format



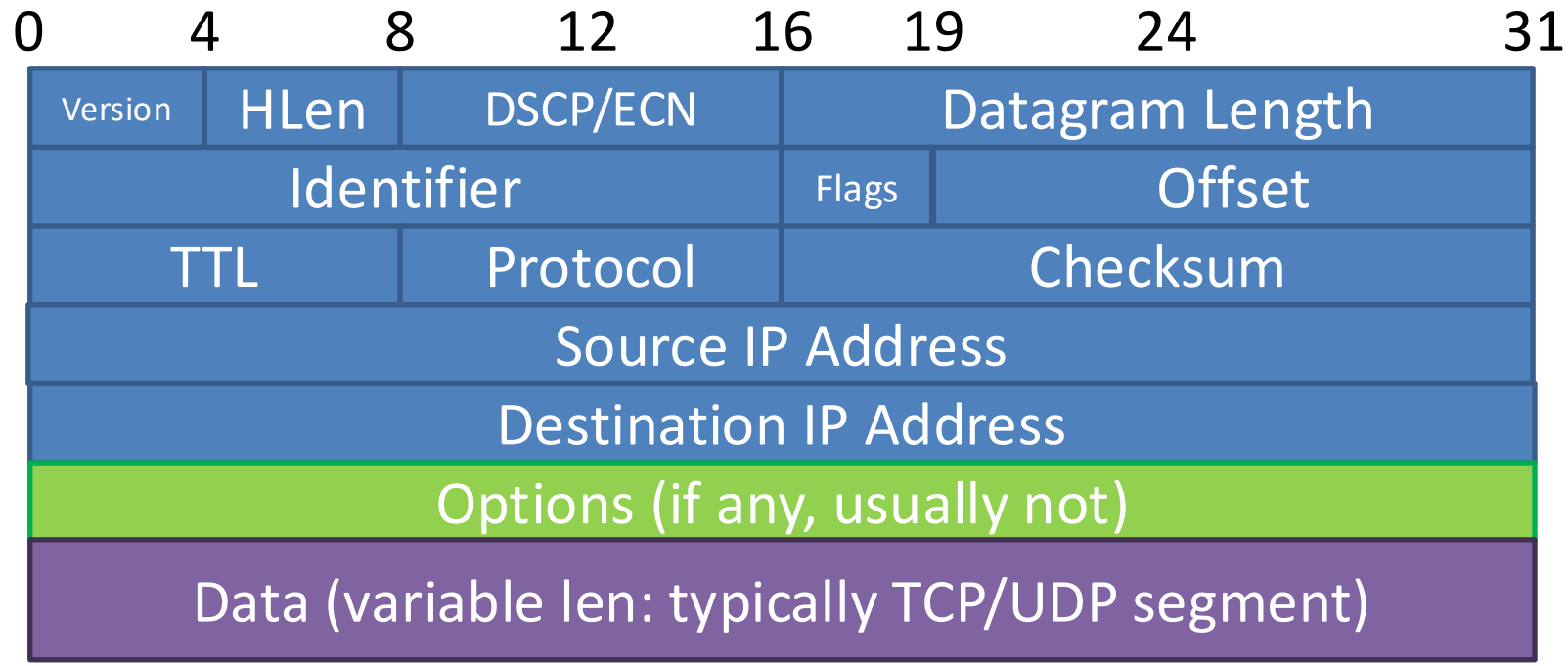
IP Datagram Format



IP Datagrams

how much overhead?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

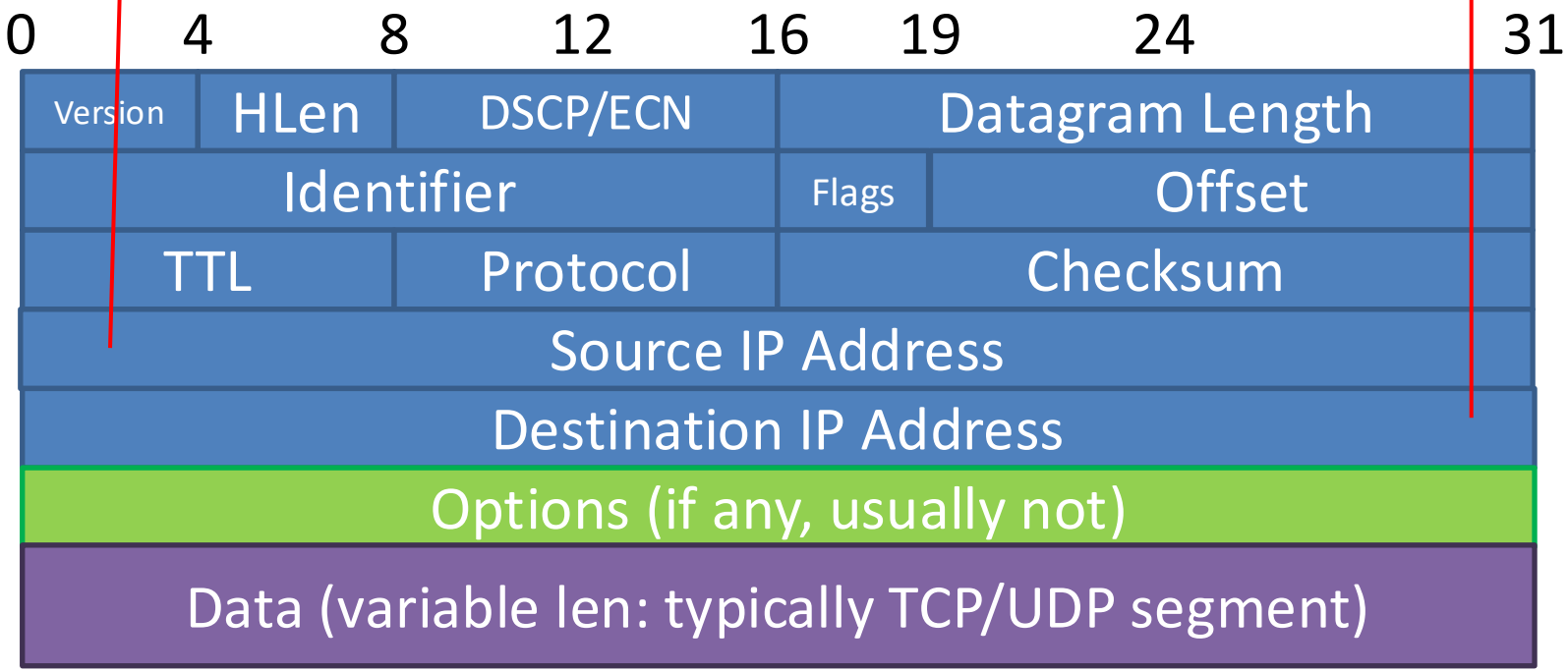


IP Datagrams

Source endpoint.

Addresses must be unique on the network!

Final destination endpoint.



What's in a name?

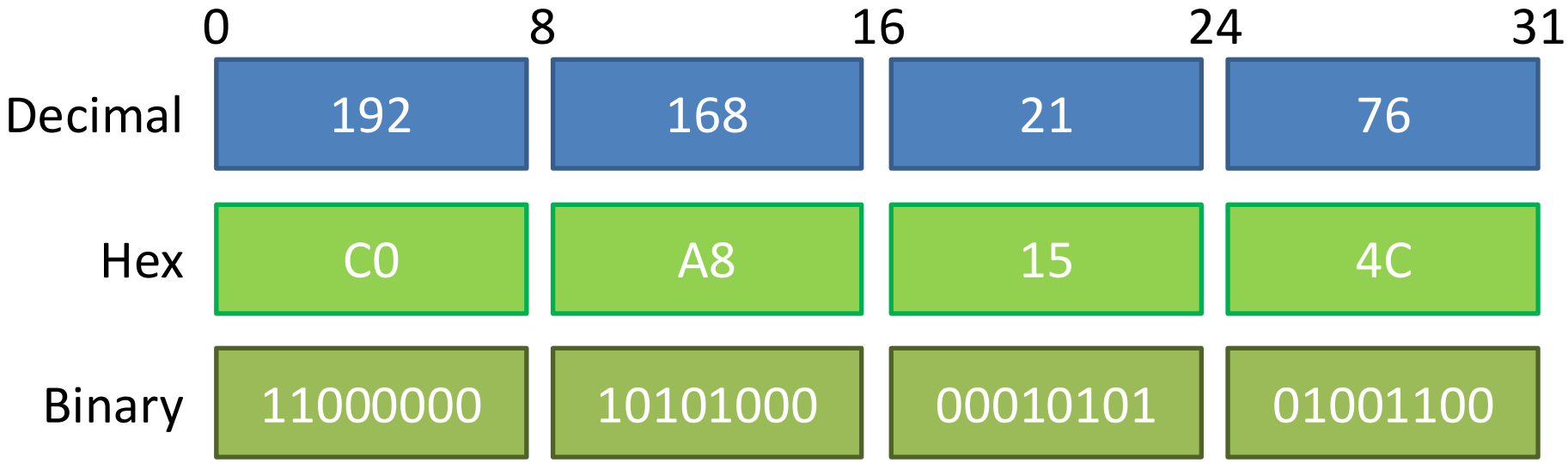
- Host name: web.cs.swarthmore.edu
  - Domain: registrar for each top-level domain (e.g., .edu)
  - Host name: local administrator assigns to each host

- IP addresses: 130.58.68.164
  - Prefixes: ICANN, regional Internet registries, and ISPs
  - Hosts: static configuration, or dynamic using DHCP

- MAC addresses: D8:D3:85:94:5F:1E
  - OIDs: assigned to vendors by the IEEE
  - Adapters: assigned by the vendor from its block

IP Addressing

- IP: 32-bit addresses
  - Usually written in dotted notation, e.g. 192.168.21.76
  - Each number is a byte
  - Stored in Big Endian order (network byte order)



IP Addresses

- 2^{32} => 4,294,967,296 possible addresses.
- In the early 80's, that's a lot!
  - Population was ~4.5 billion.
- Now…not so much.
  - Population > 7 billion.