CS 43: Computer Networks

16-17: The Network Layer

November 7, November 12 2024

SWARTHMORE COLLEGE

The Network Layer!

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

Transport: end-to-end connections, reliability

Network: routing

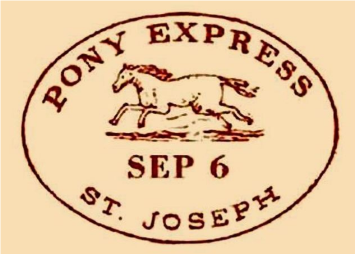
Link (data-link): framing, error detection

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

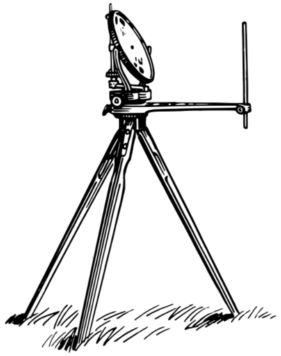
Network Layer

- 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

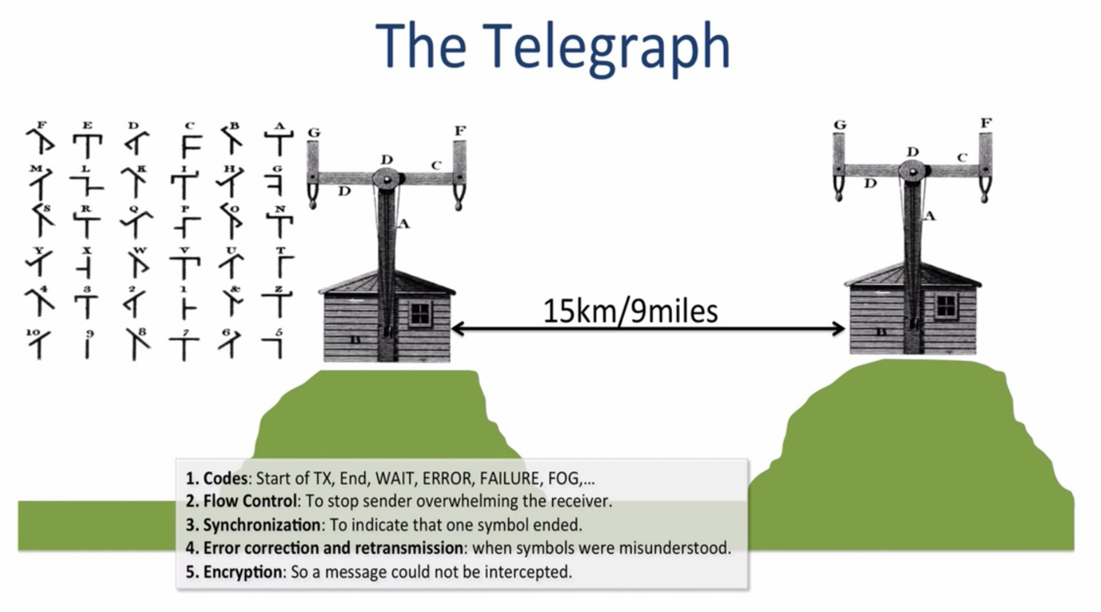
History of Communication



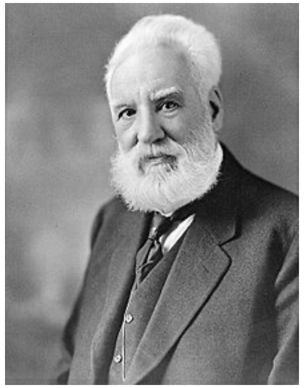
Fire Beacons
Carrier Pigeons
Human Messengers
Horse Relays – Pony Express



Wireless telegraph
- speed of light
- compression
- limited information

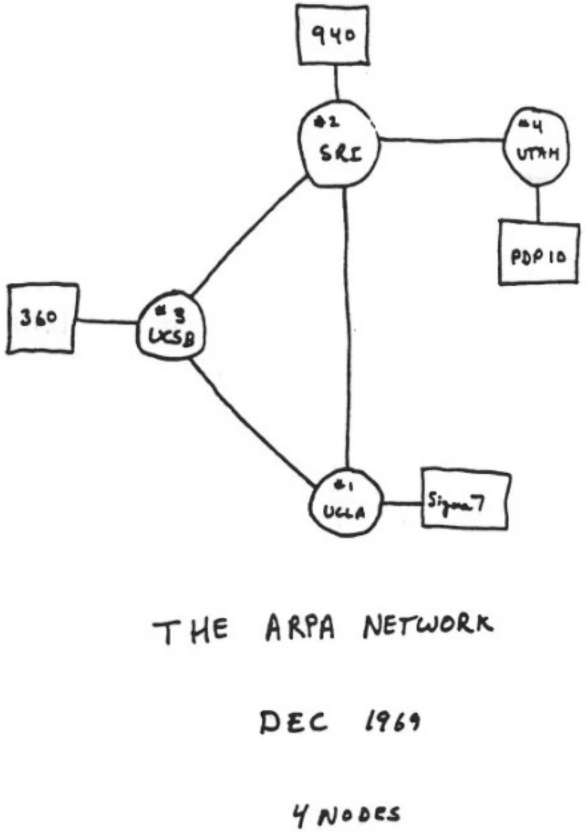


Telephone Network



Courtesy: Stanford University

History of the Internet: ARPANET



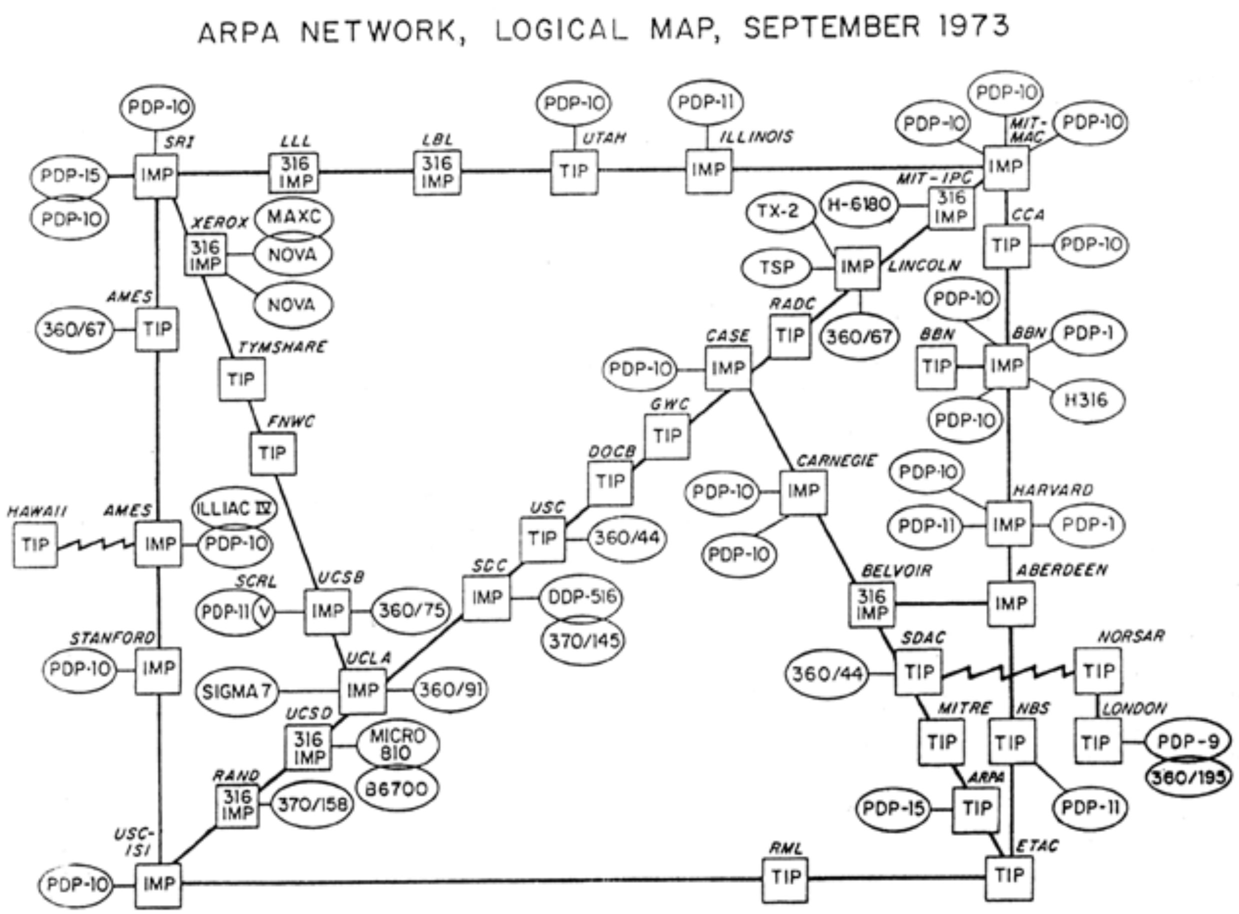
Courtesy: Scientific American

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”

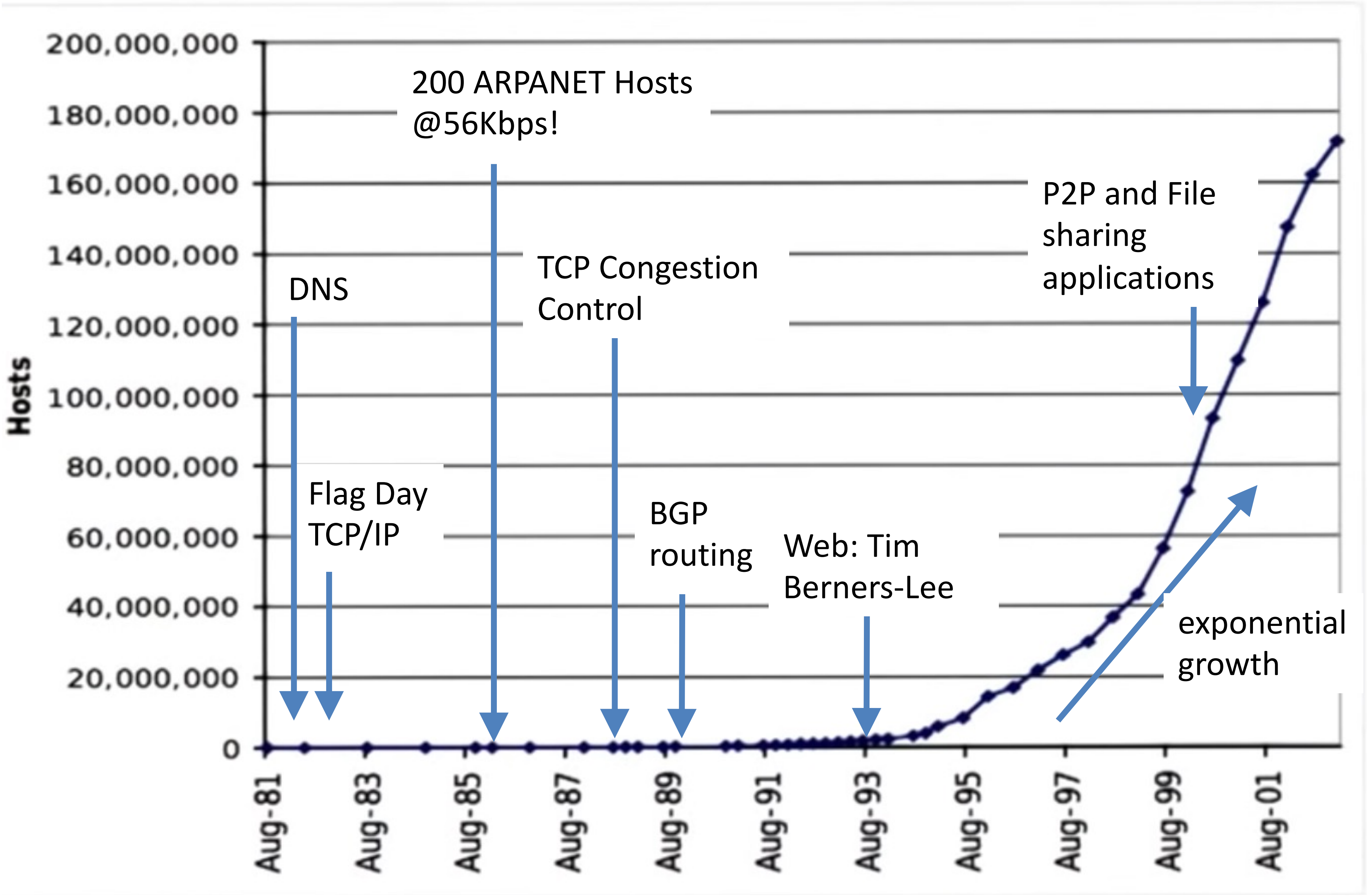
Swat Alum!

Cerf & Kahn: TCP/IP protocols

Turing Award Winners

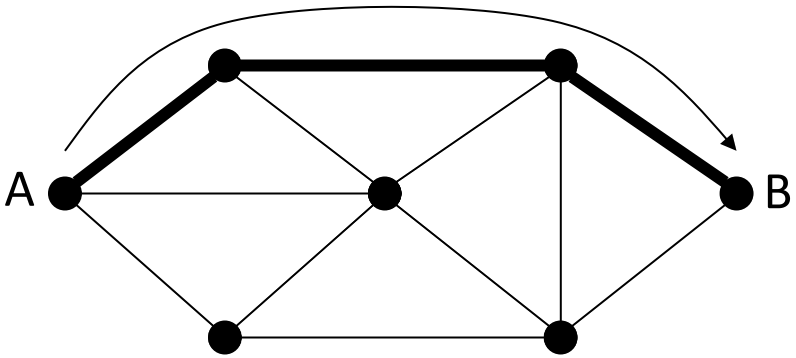
Vincent Cerf and Bob Kahn

Picture Source: Wikipedia


200,000,000
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
Hosts
Aug-81
Aug-83
Aug-85
Aug-87
Aug-89
Aug-91
Aug-93
Aug-95
Aug-97
Aug-99
Aug-01
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

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.

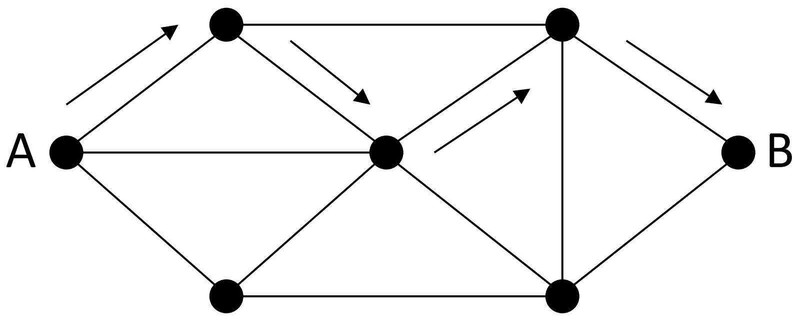
D. It requires too many resources.

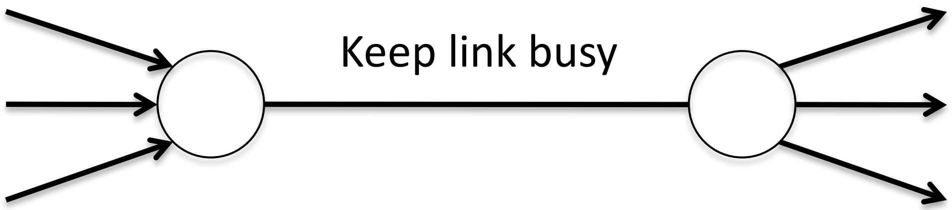
E. Some other reason.

Packet Switching

- Do we always need to reserve a link?

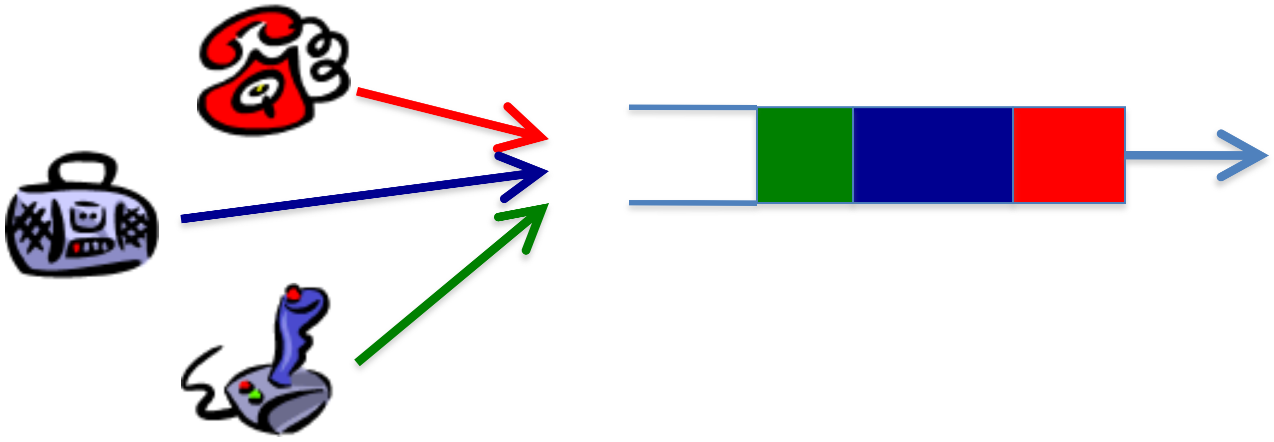
- <u>Statistical multiplexing</u>
  - Assign multiple conversations to a physical path
  - At any given time, one will have something to say





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.

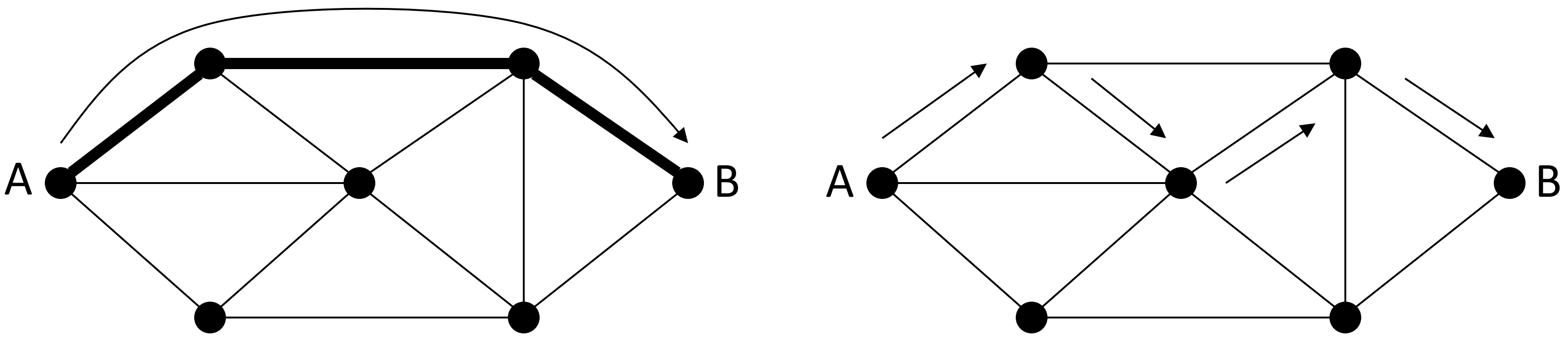
A. Only 1 is true.
B. Only 2 is true.
C. Both 1 and 2 are true.
D. Neither 1 nor 2 are true.

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

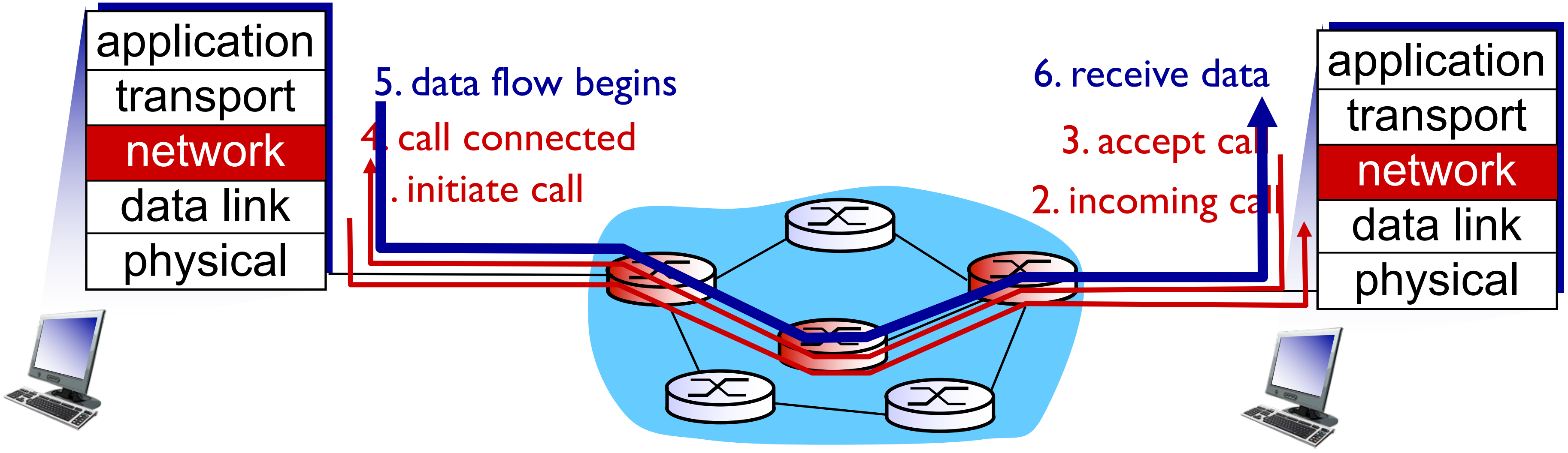
Datagram vs. "Virtual Circuit"

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

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

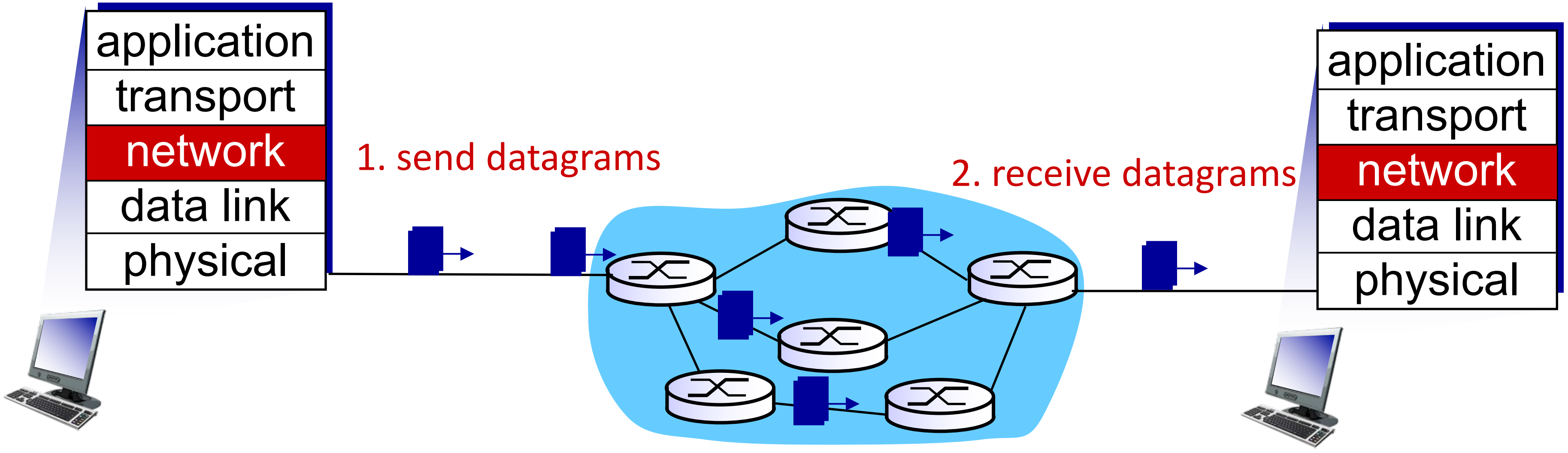
Virtual circuits: Signaling Protocols

- 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

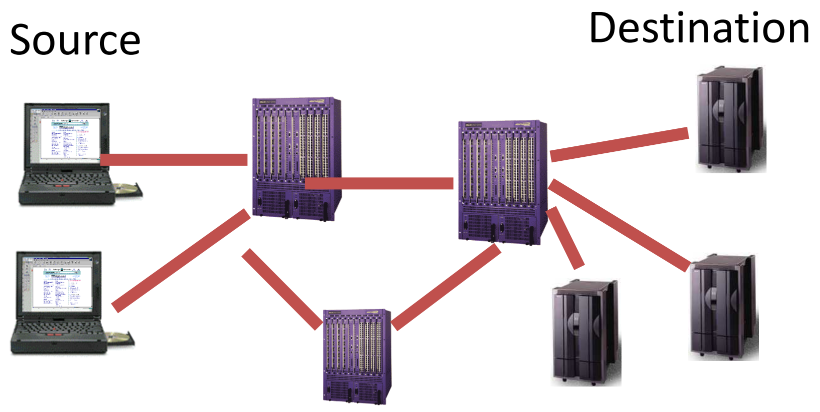
Serve Cached Data

Add Security

Provide reliability

Migrate connections

.... the list is long



The End-to-End Principle

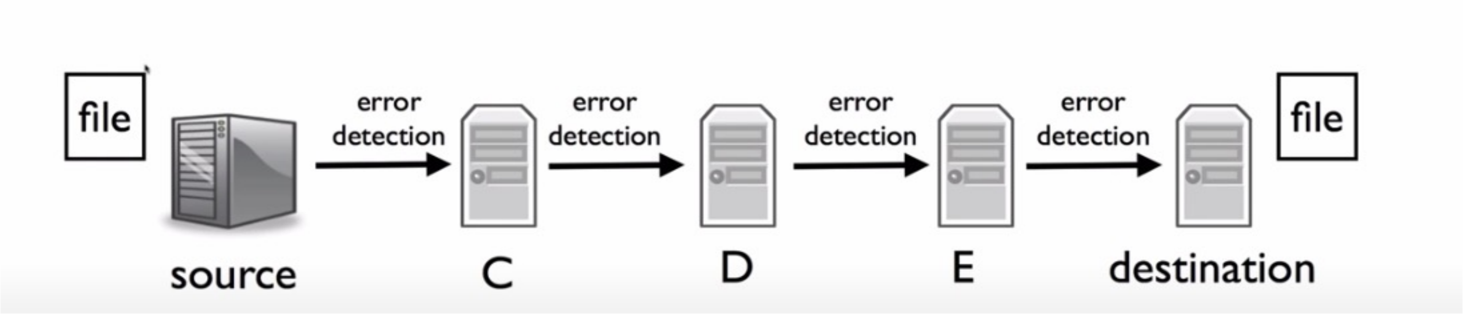
"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system."

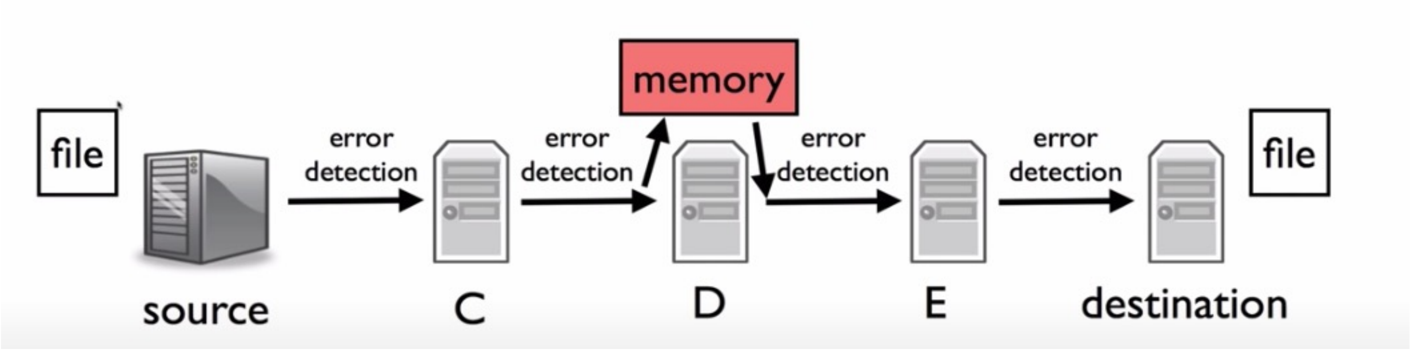
- Saltzer, Reed and Clark

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.

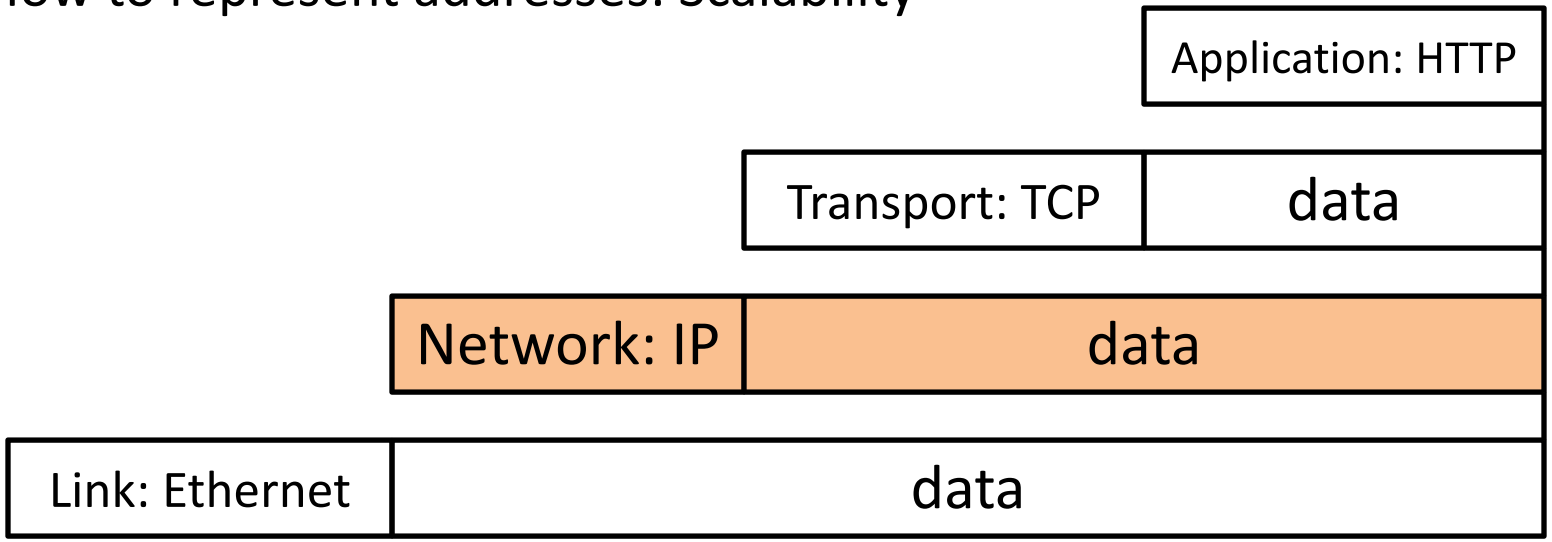
-RFC 1958

Courtesy: Stanford

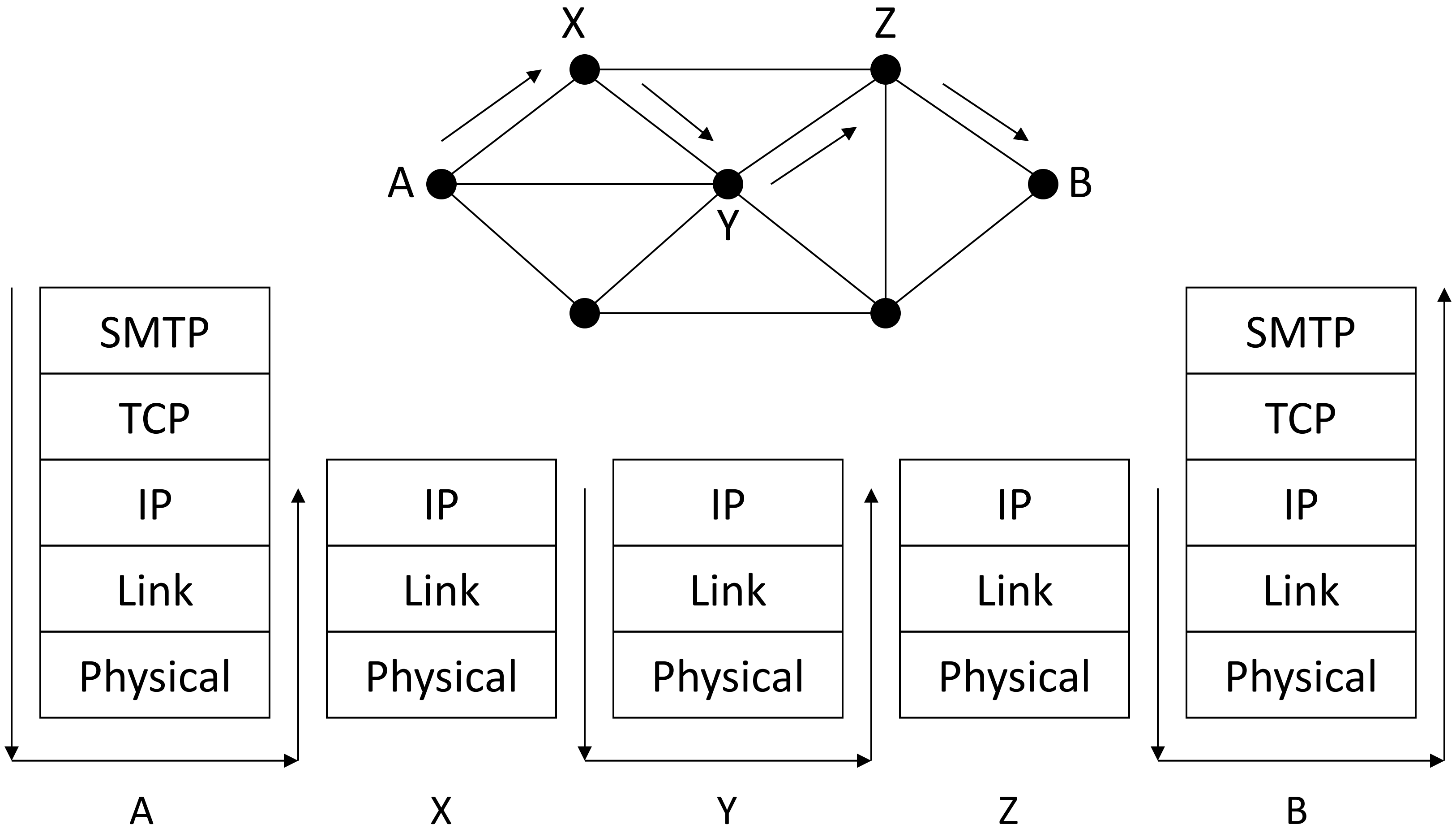
Network Layer

- 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.

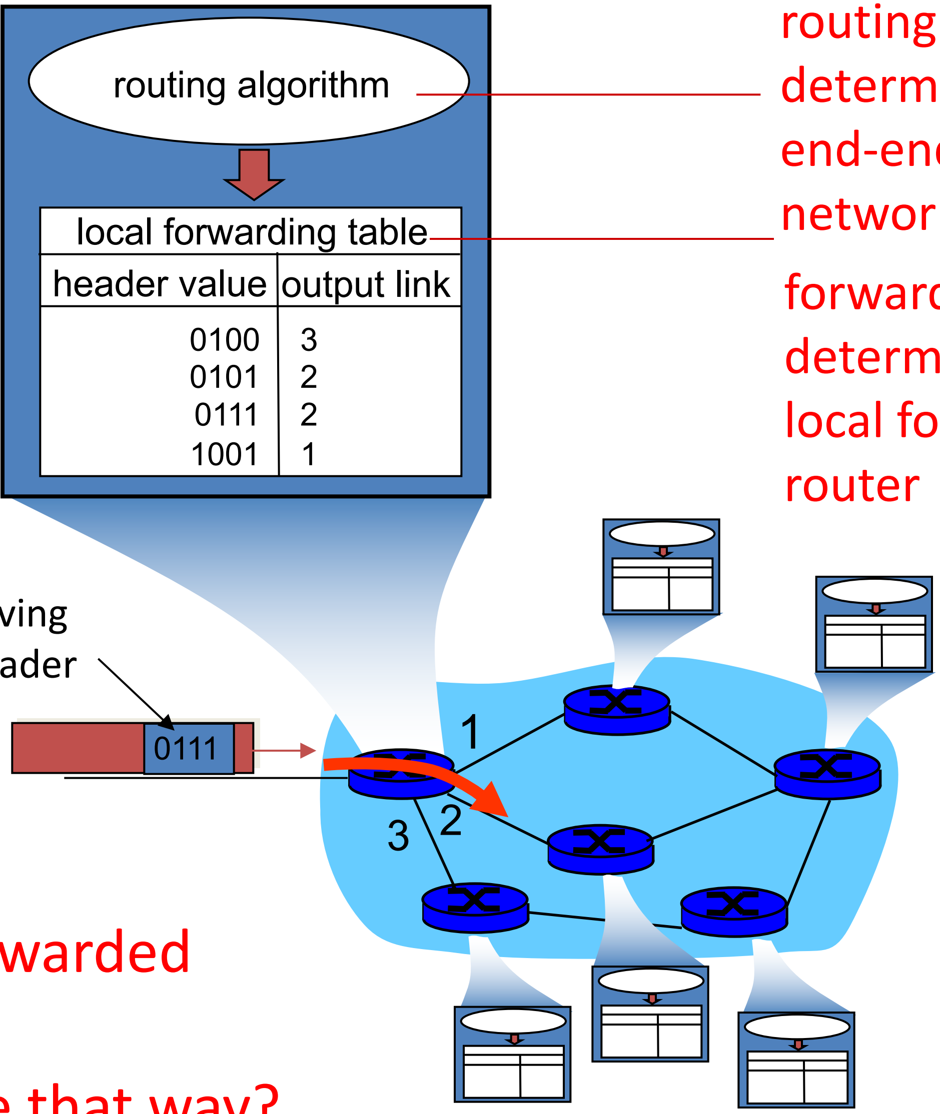
- Forwarding:
  - Copying bytes from one interface to another, can’t forward in advance
  - forwarding needs to happen very quickly: millions of packets per second
- Routing:
  - High-level decision that we do dynamically, at different time-scales than forwarding
  - route in advance and populate a table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

value in arriving packet's header



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).

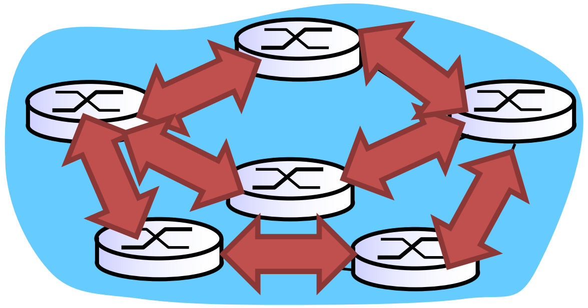
C. Routers should communicate with each other to add entries to their tables (used today).

D. Some other mechanism.

Routing

Traditional

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

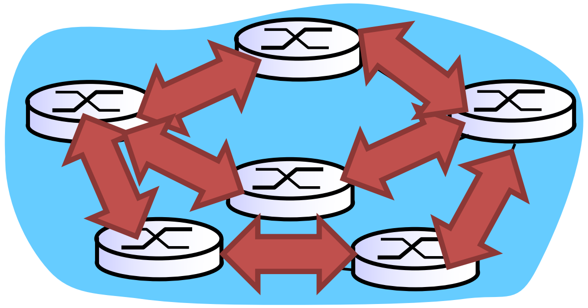


Assume this is the type of routing we're talking about unless we explicitly say otherwise!

Routing

Traditional

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



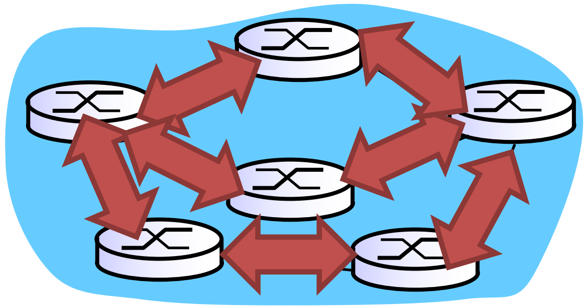
"Software-Defined"

- 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.

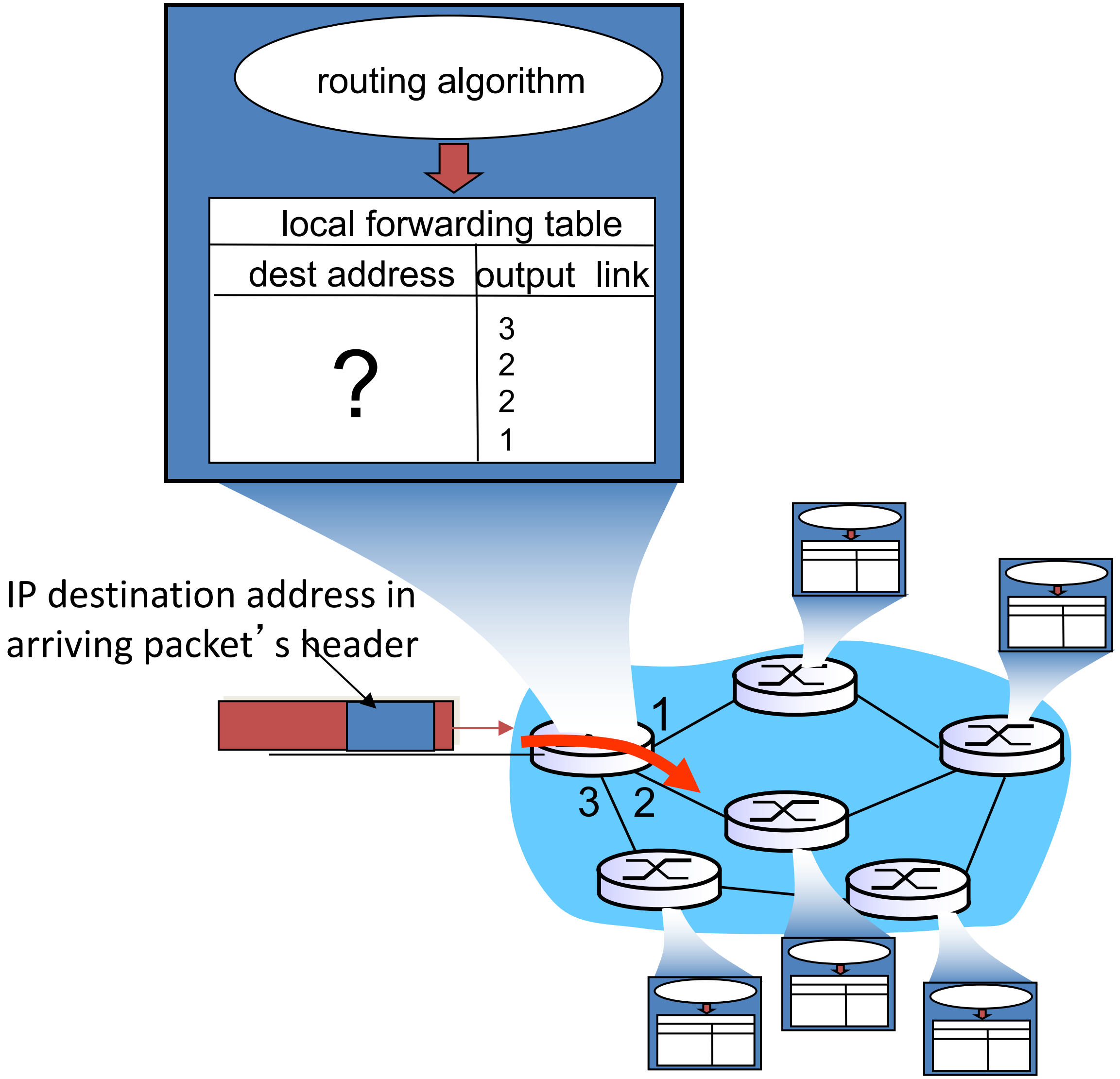
Datagram Forwarding

- 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.

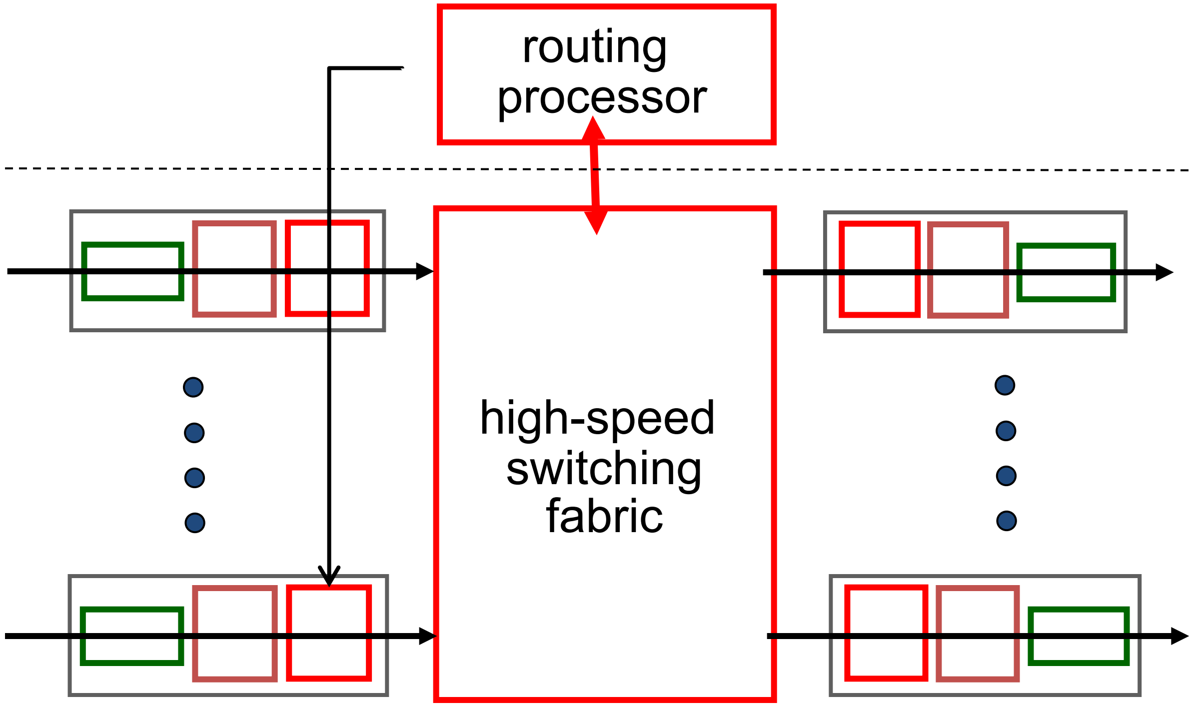
D. Some other software structure.

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

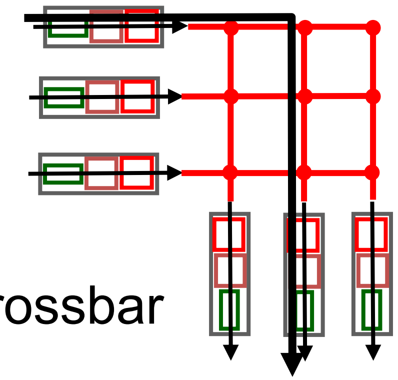
Aside: router architecture overview

- high-level view of generic router architecture:

routing, management
control plane (software)
operates in millisecond
time frame



forwarding data plane
(hardware) operates
in nanosecond
timeframe

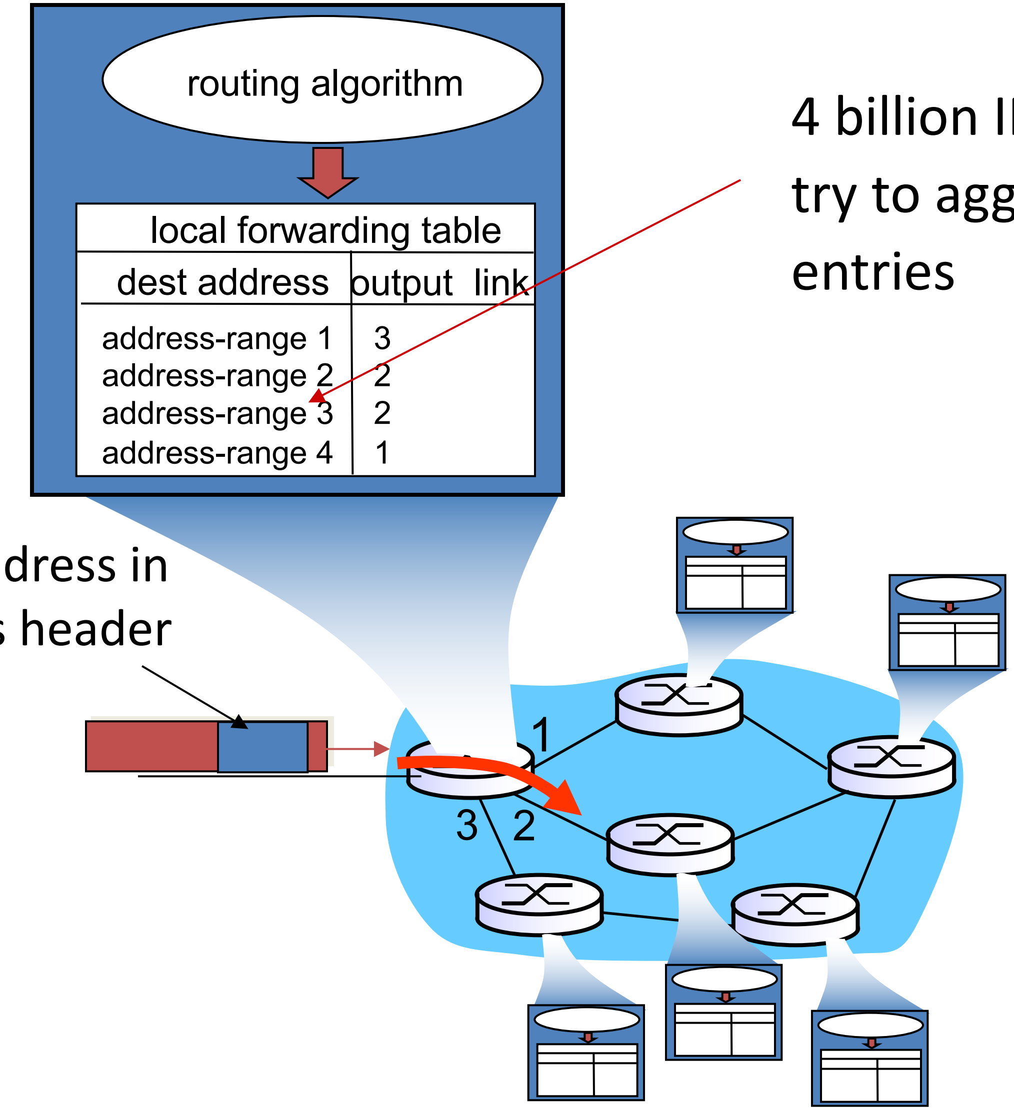


crossbar

router input ports
(different from TCP ports!!)
these are physical inputs/outputs to the router

router output ports

Datagram forwarding table



4 billion IP addresses, try to aggregate table entries

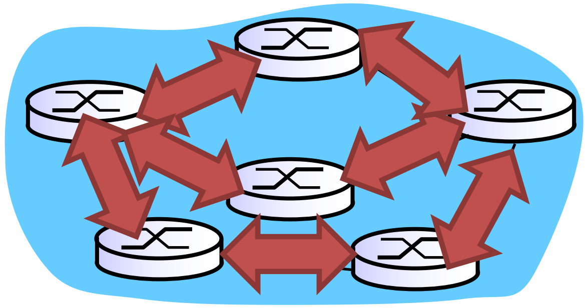
IP destination address in arriving packet’s header

Routing

<u>Traditional</u>

- 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..

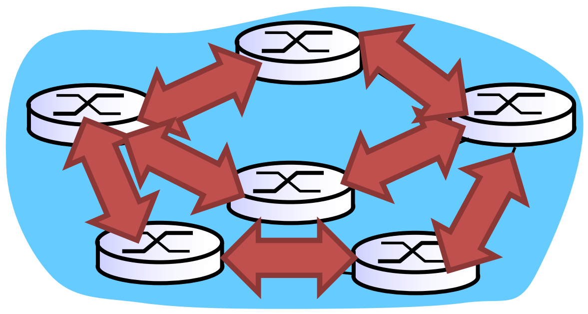
- Traffic engineering:
  - Want to avoid persistent overloads on links
  - Choose routes to spread traffic load across links
- Access Control:
  - Limit access to backend database machines.
  - Firewalls
- Network measurement

Routing

<u>Traditional</u>

- Routers run a routing protocol to exchange state.

- Use state to build up the forwarding table.



<u>Software-Defined</u>

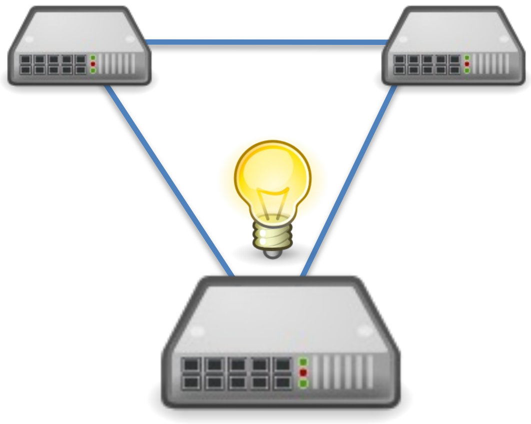
- Routers are dumb, just do what they're told.

- Controller service explicitly tells each router what to do.

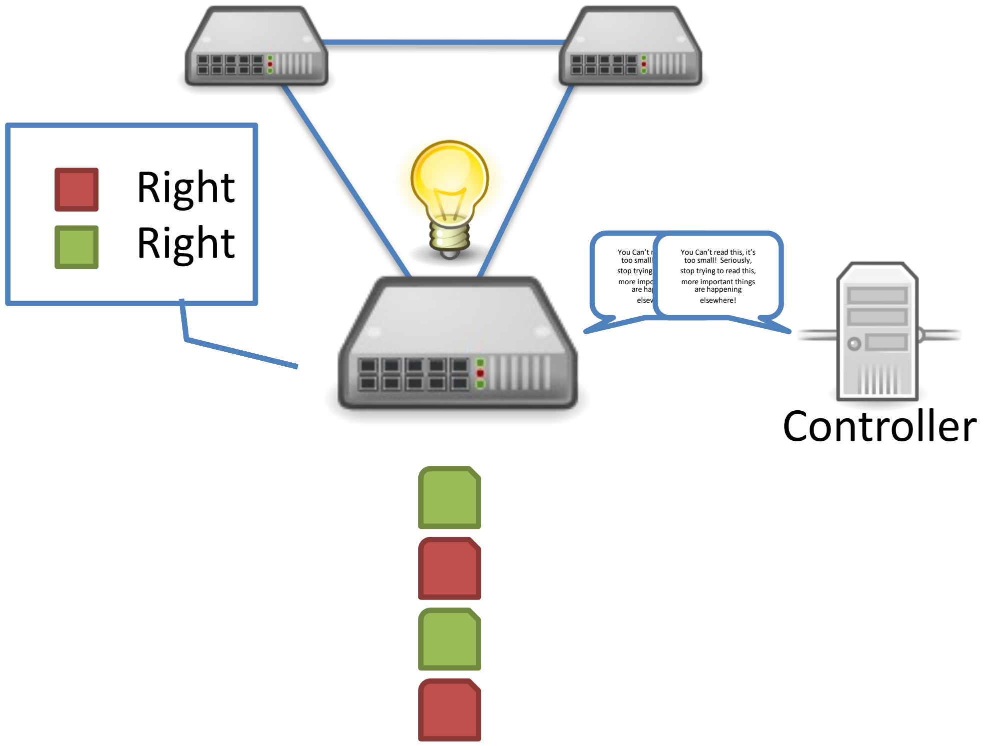
- Rare on the Internet, hot topic in data centers.

Software-Defined Networking (SDN)

<u>Traditional Hardware</u>



<u>SDN Hardware</u>



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.

CS 43: Computer Networks

Network Layer, IP

November 12, 2024

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

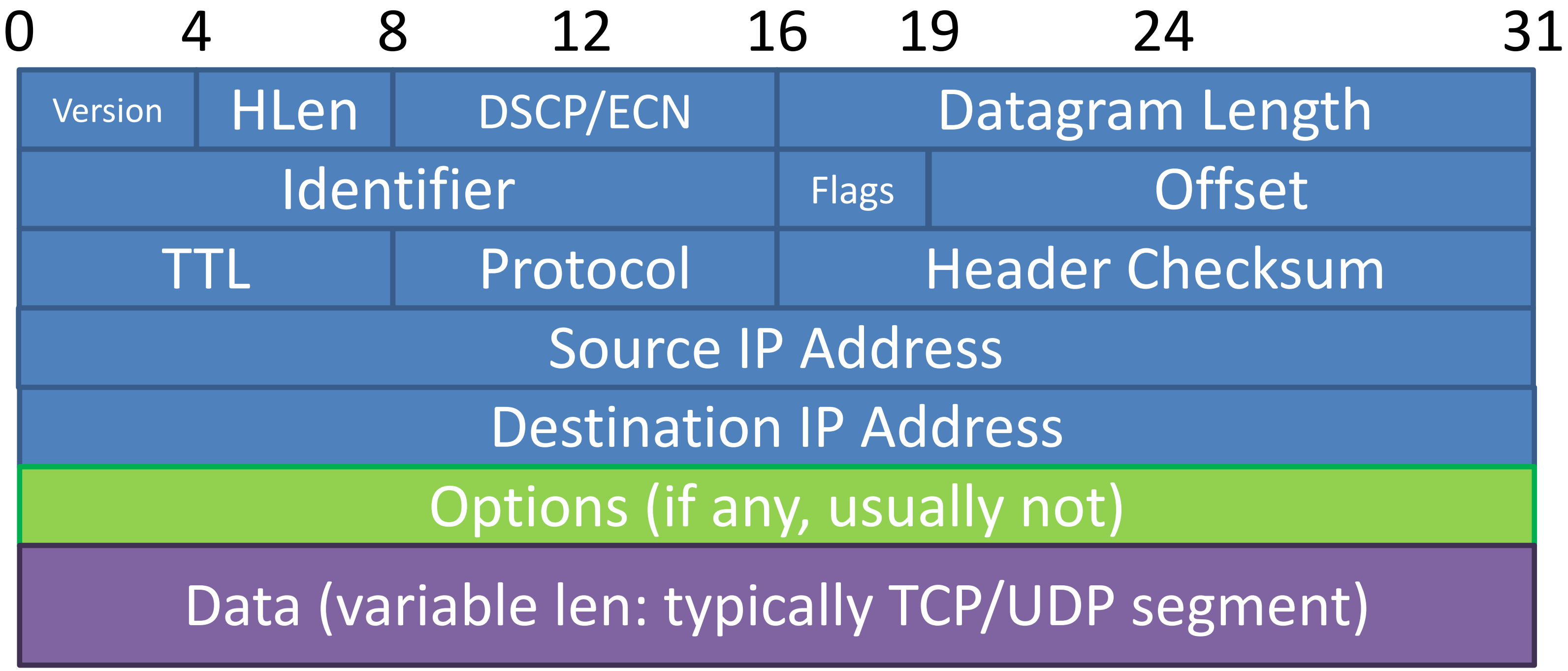
Physical: 1’s and 0’s/bits across a medium (copper, the air, fiber)

Network Layer Functions

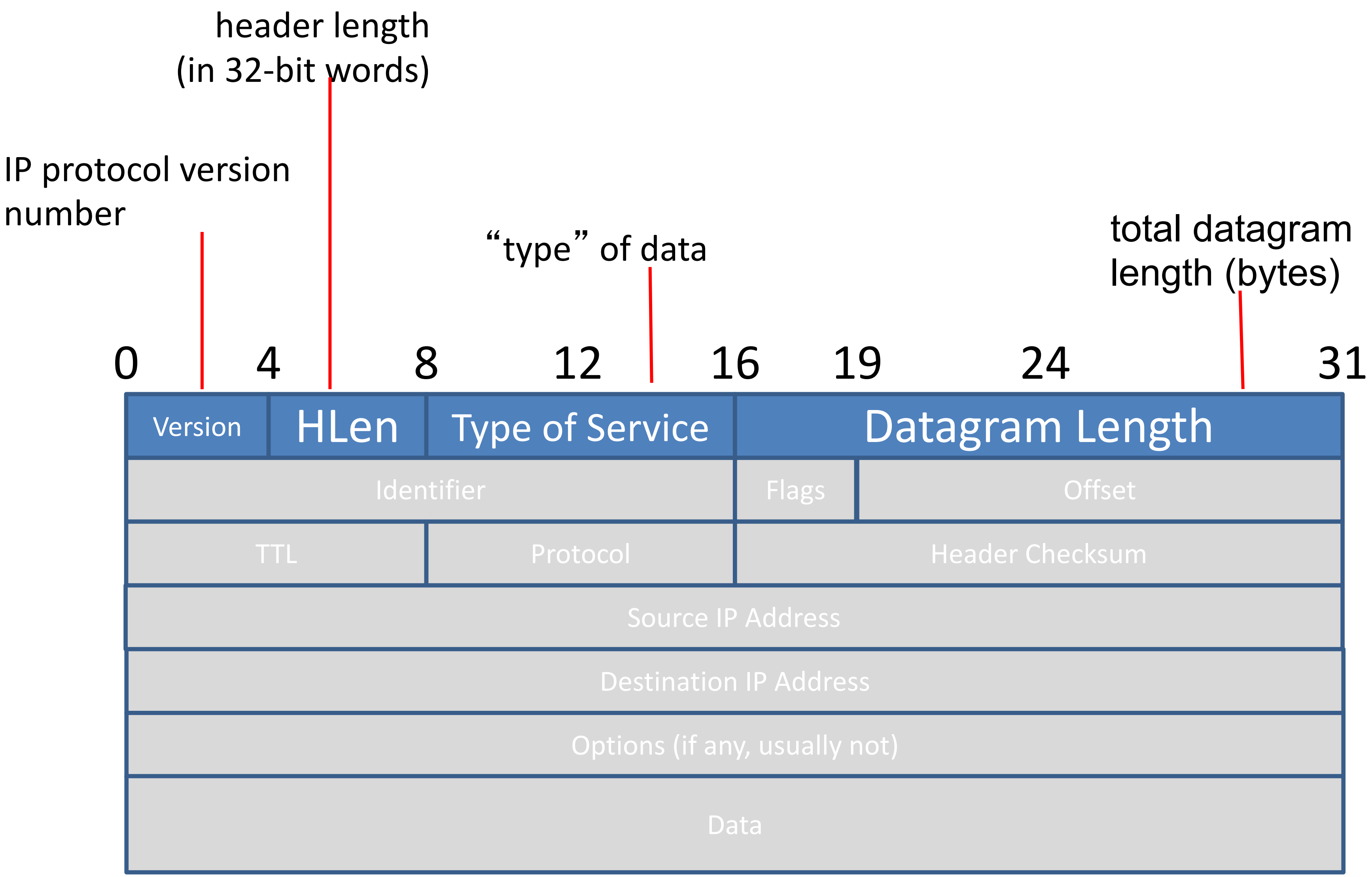
- Forwarding: move packets from router's input to appropriate router output
  - Look up in a table
- Routing: determine route taken by packets from source to destination.
  - Populating the table

IP Datagrams

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

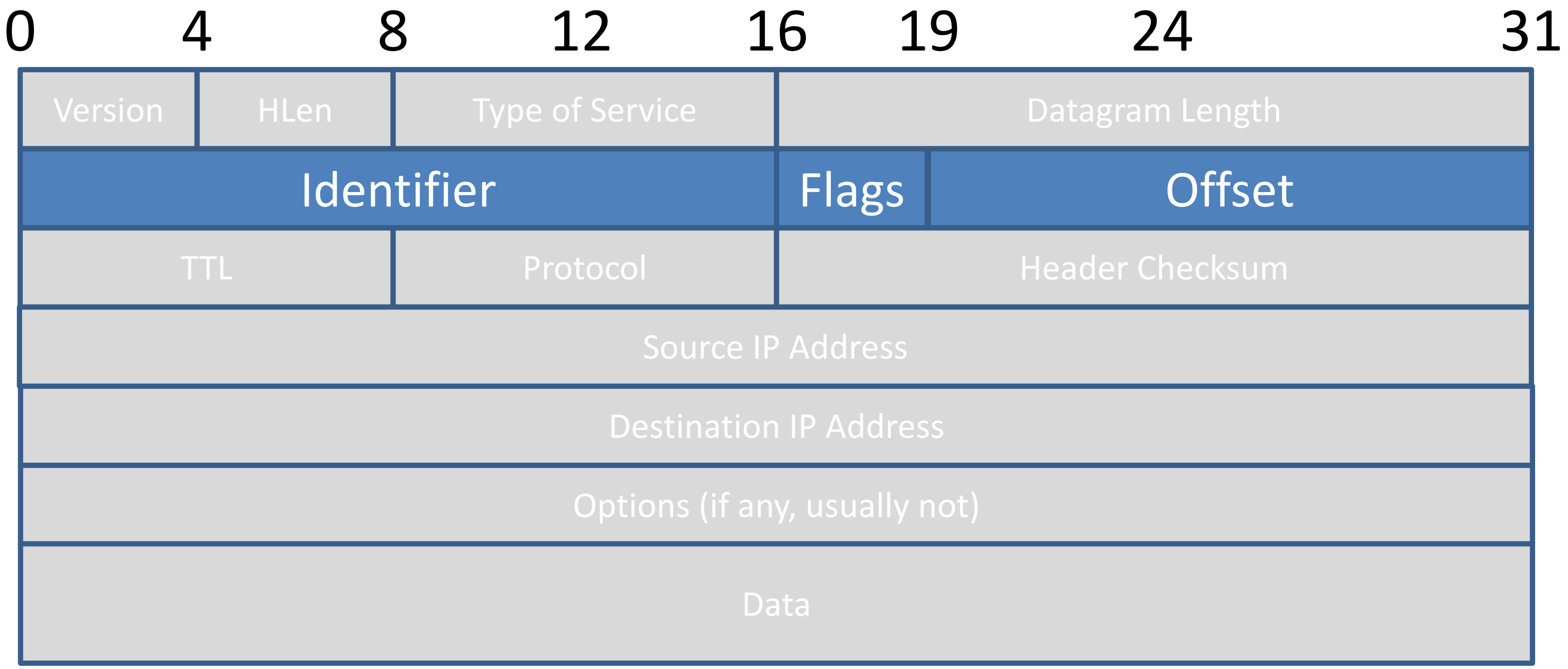


IP Datagram Format

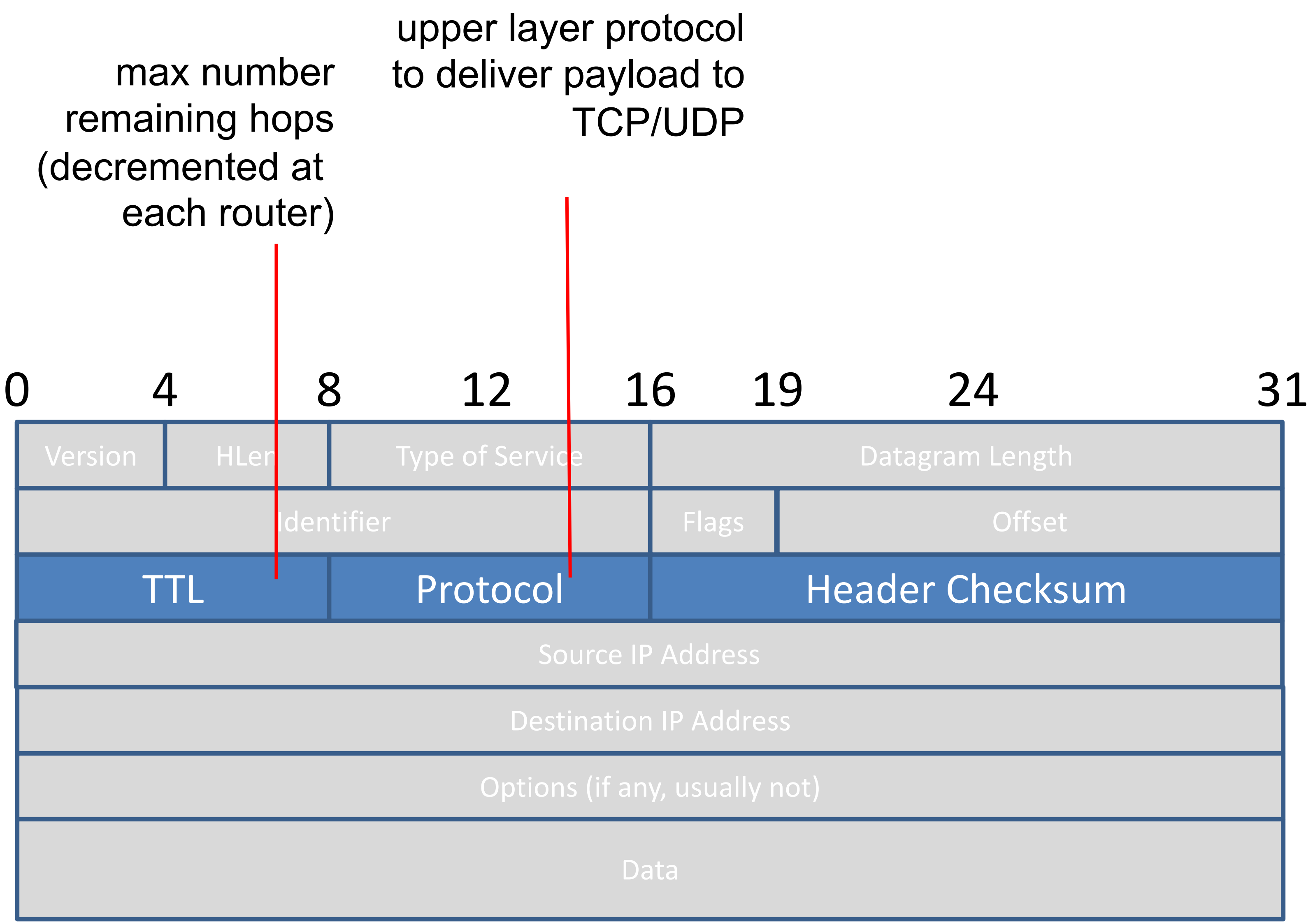


IP Datagram Format

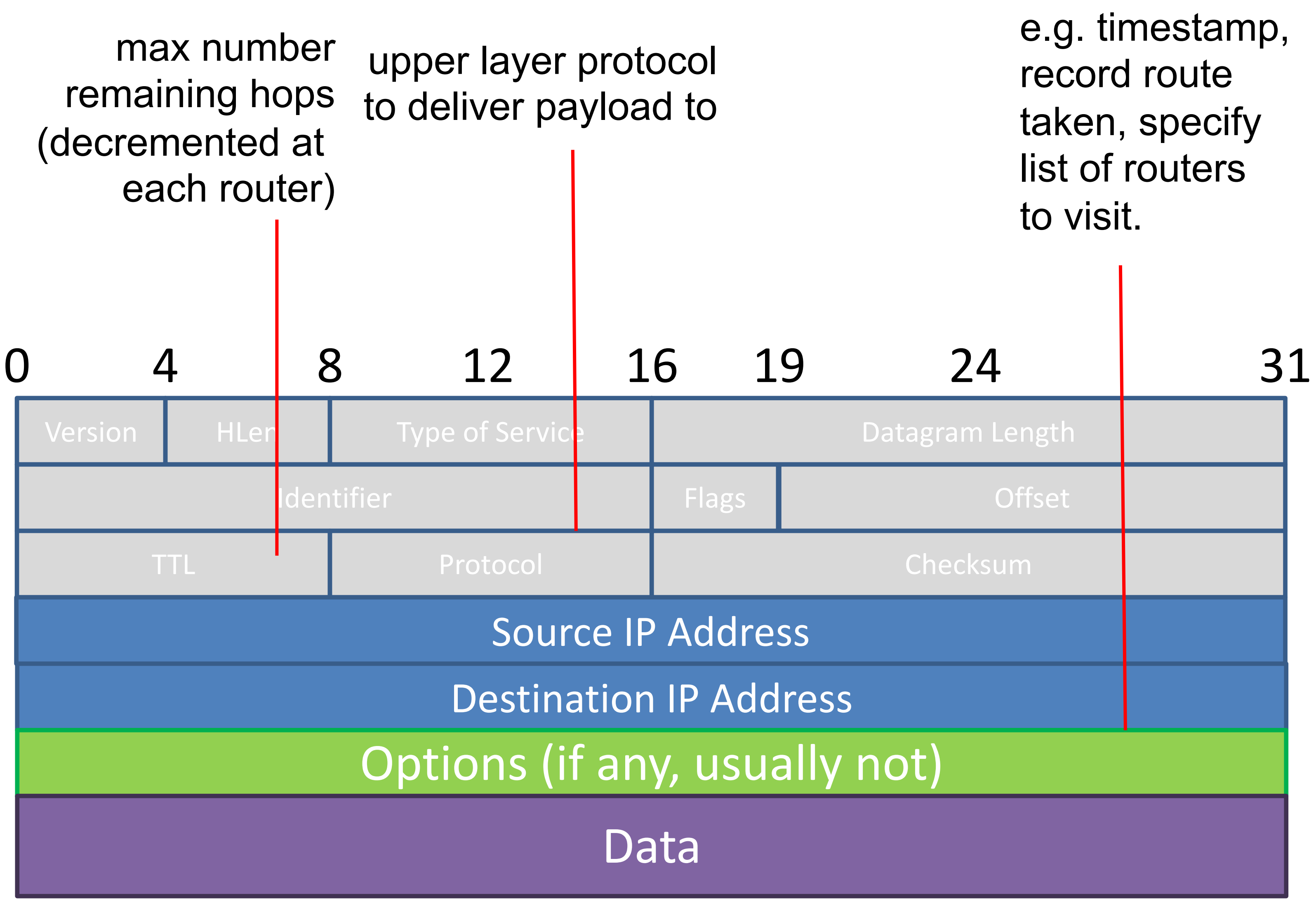
Fragmentation/ reassembly: Identifier, Flags, Offset



IP Datagram Format



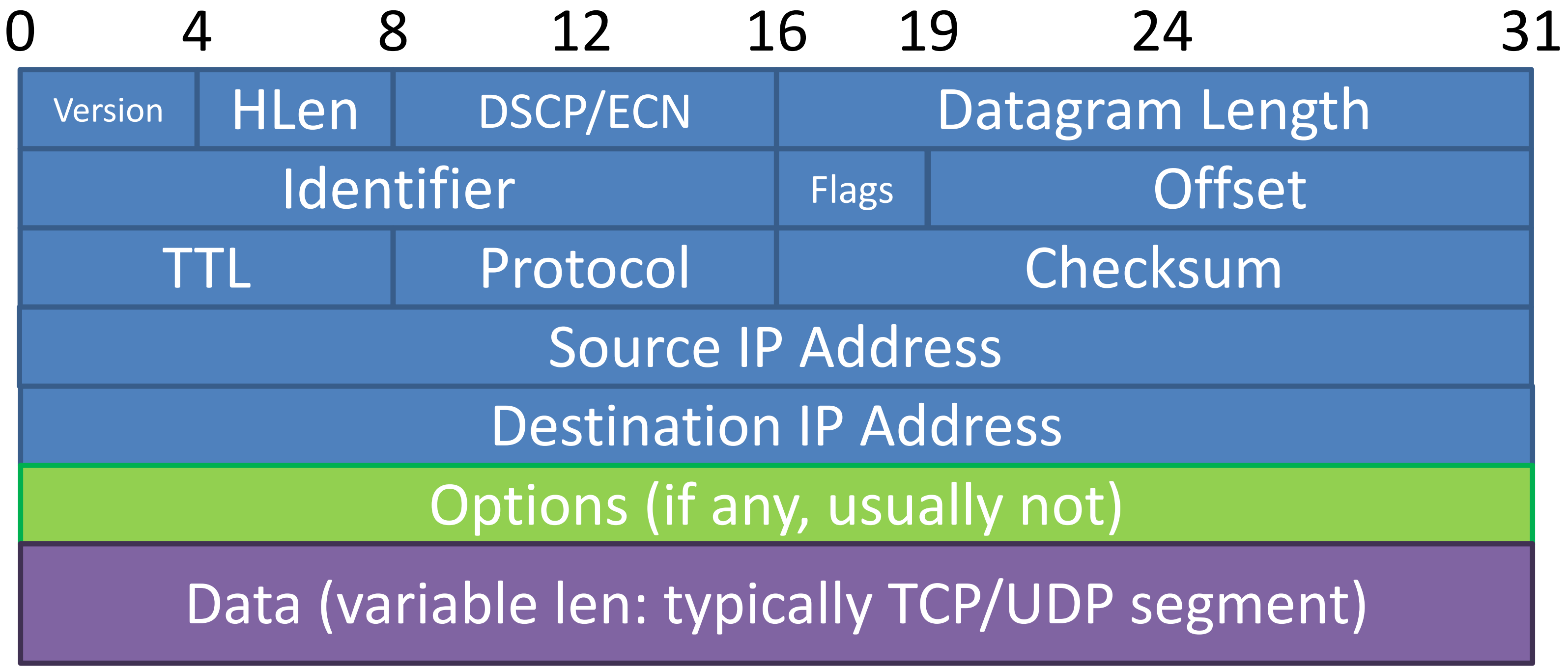
IP Datagram Format



IP Datagrams

how much overhead?

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

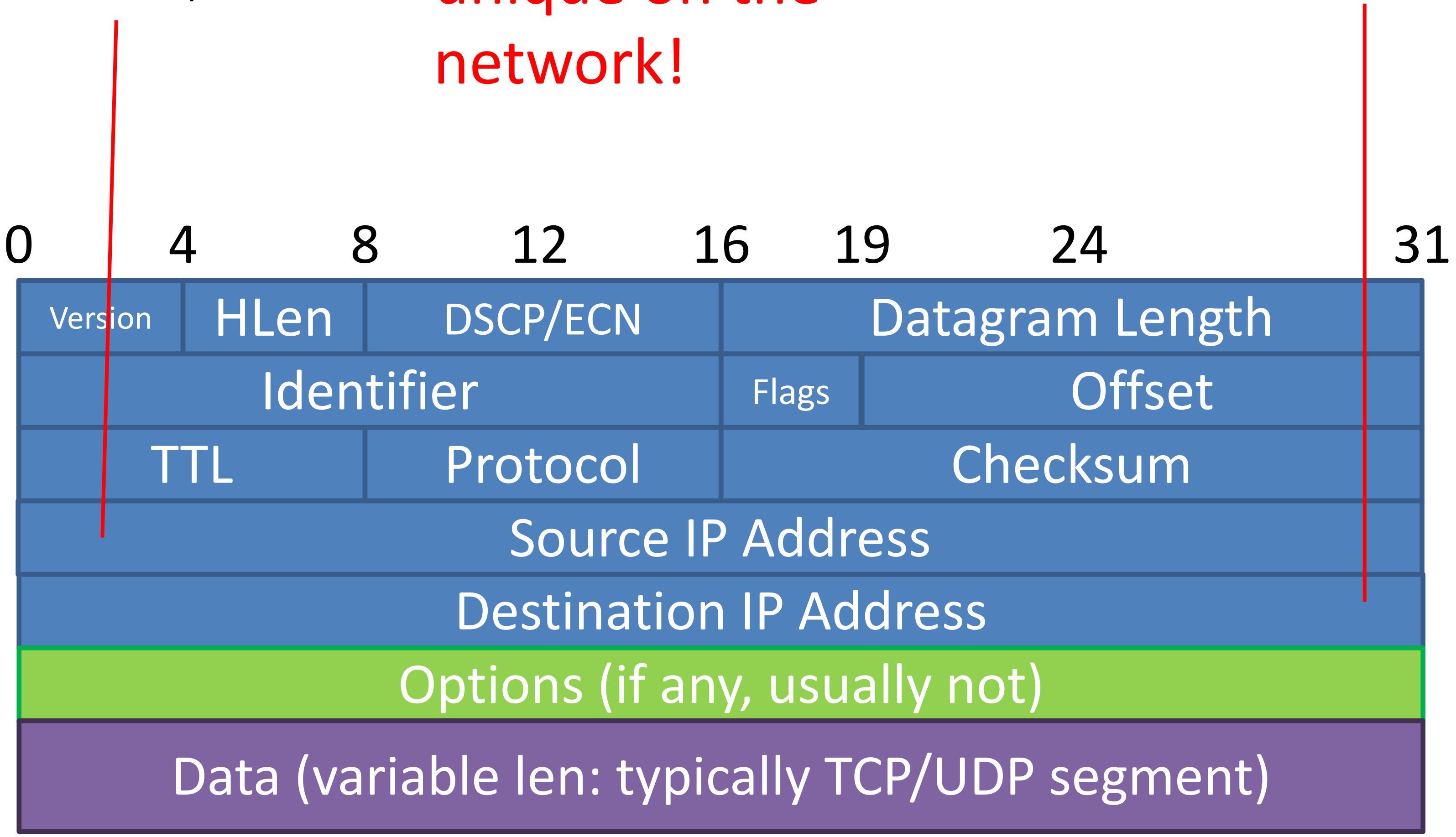


IP Datagrams

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)

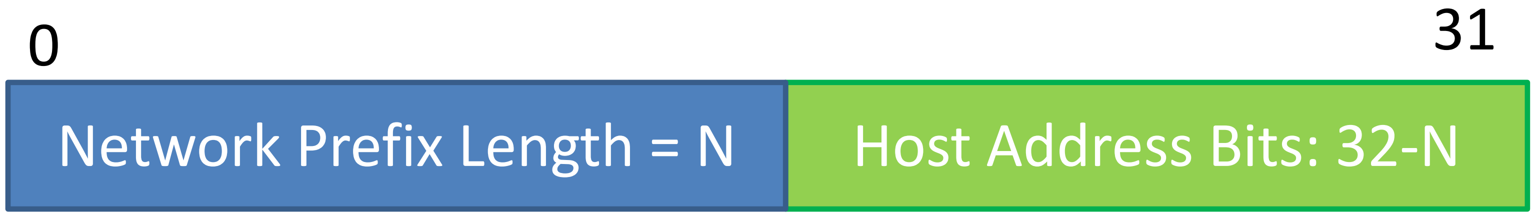
	0	8	16	24	31
Decimal	192	168	21	76	
Hex	C0	A8	15	4C	
Binary	11000000	10101000	00010101	01001100	

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.

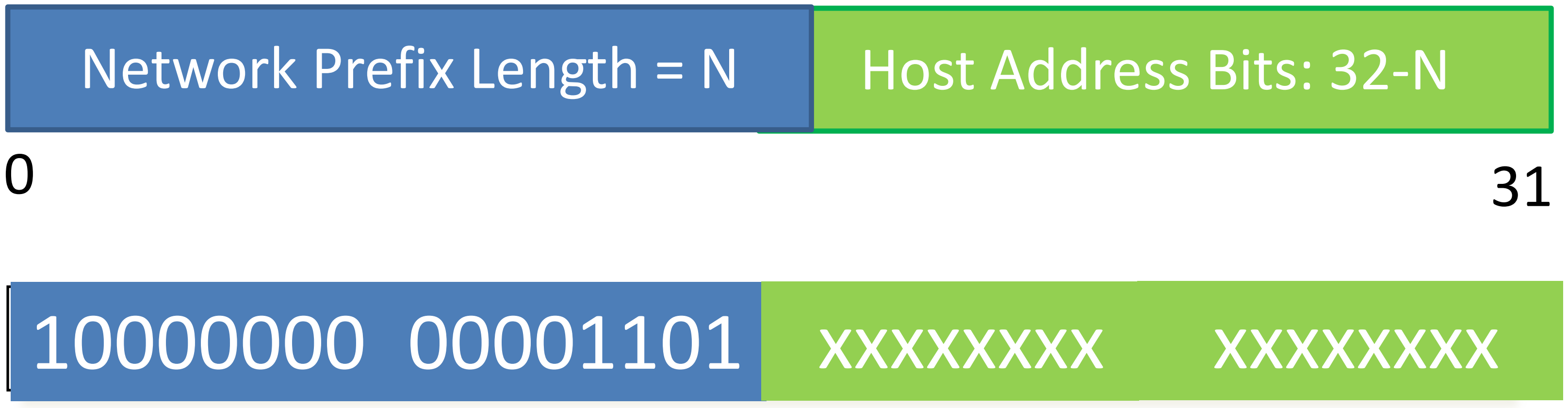
IP Prefixes

- Addresses are allocated in blocks called prefixes to organizations
- Addresses in an N-bit prefix have the same top N bits.
- If an organization has an IP/N prefix, it can allocate 2^32-N addresses to end hosts on its network



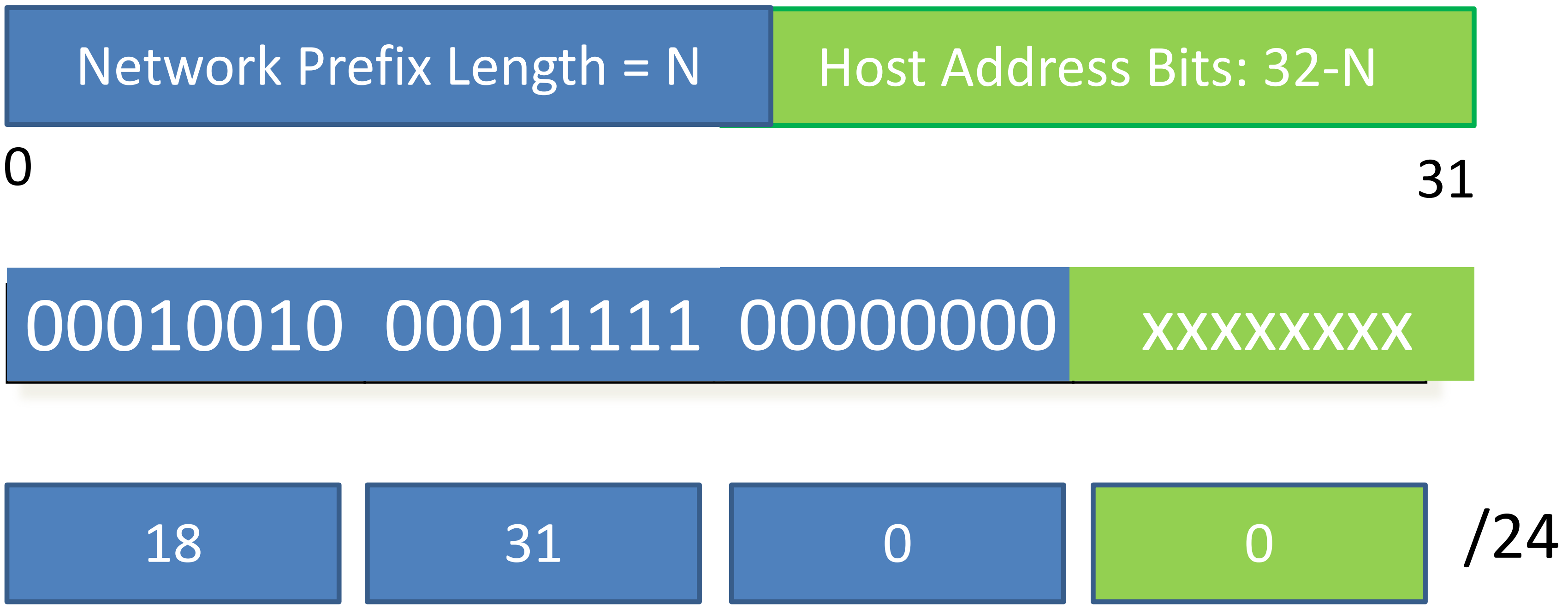
IP Prefixes

- Written in IP address/length notation
- Address is the lowest address in the allocated block. Length is prefix in bits.
- E.g. 128.13.0.0/16 is 128.13.0.0 to 128.13.255.255
  Read as: ”128.13.0.0 slash 16” prefix.



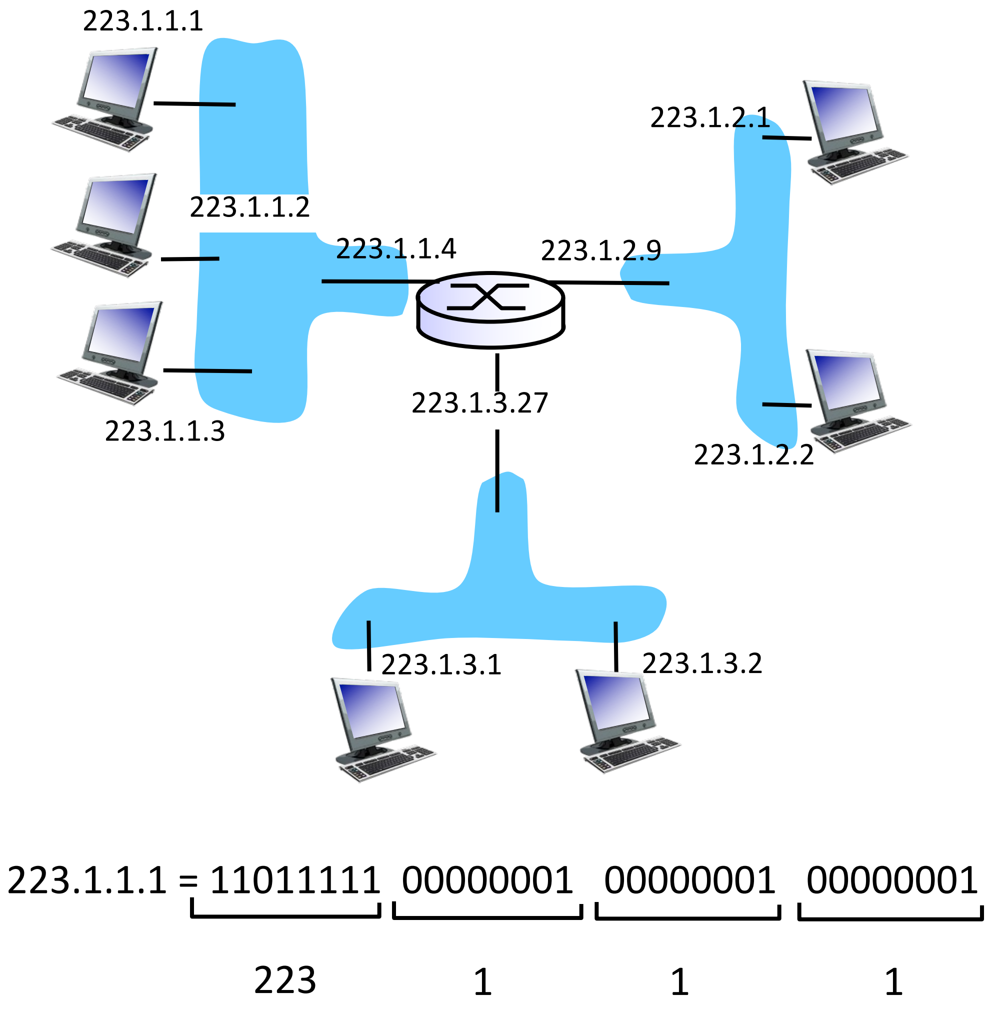
IP Prefixes

How would we express the following prefix?



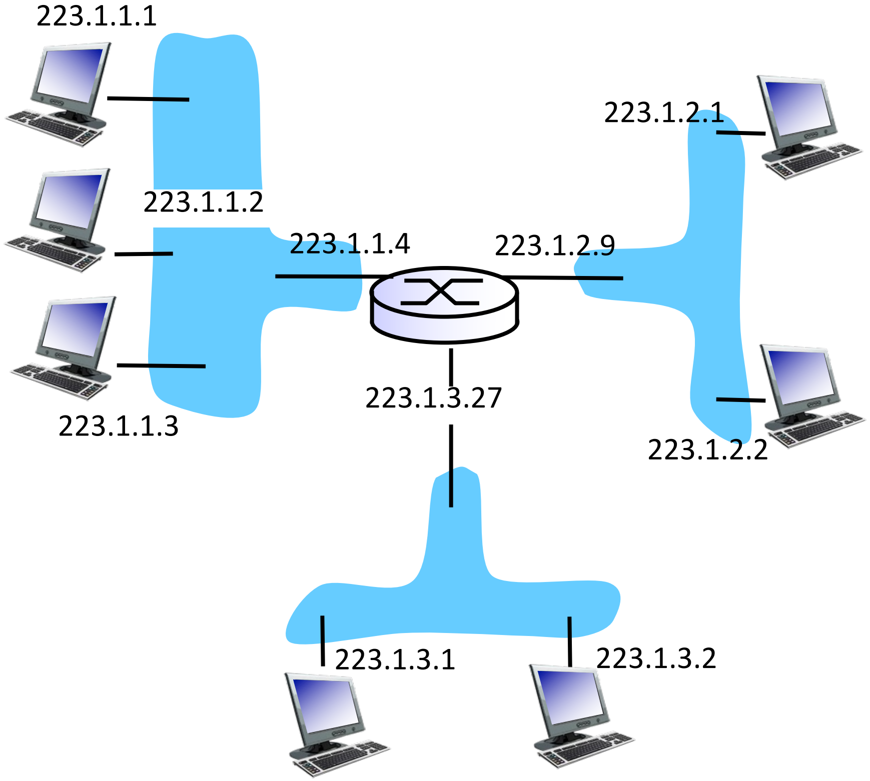
Network Interfaces

- IP address: 32-bit identifier for host, router *interface*
- interface: connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)
- IP addresses associated with each interface



Subnets

- IP address:
  - subnet part - high order bits
  - host part - low order bits
- what's a subnet?
  - device interfaces with same subnet part of IP address
  - can physically reach each other without intervening router
  - On the same link layer



223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

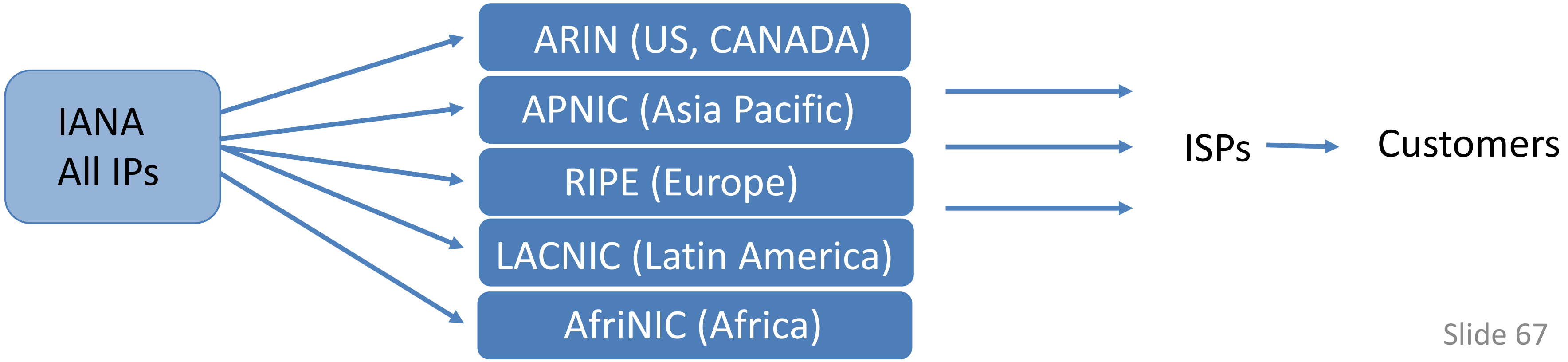
Who gets an address? How many?

- Back in the old days, you called up Jon Postel
  - “How many addresses do you need?”
  - “Here you go! I may have rounded a bit.”

Assigning Addresses

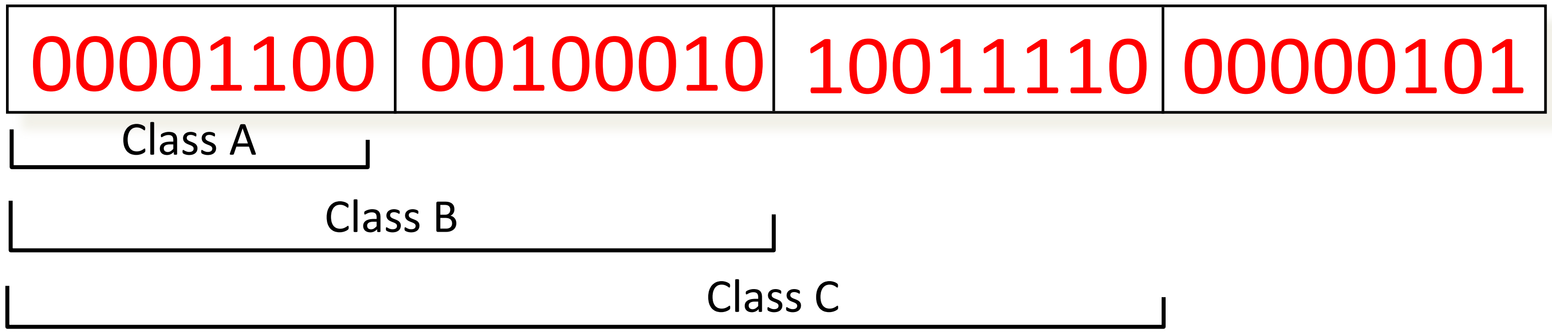
- IANA – Internet Assigned Numbers Authority
  – (Run by Jon Postel until 1988)
  – Now a part of ICANN

- ICANN: Internet Corporation for Assigned Names and Numbers
  – Manages IP addresses, DNS, resolves disputes

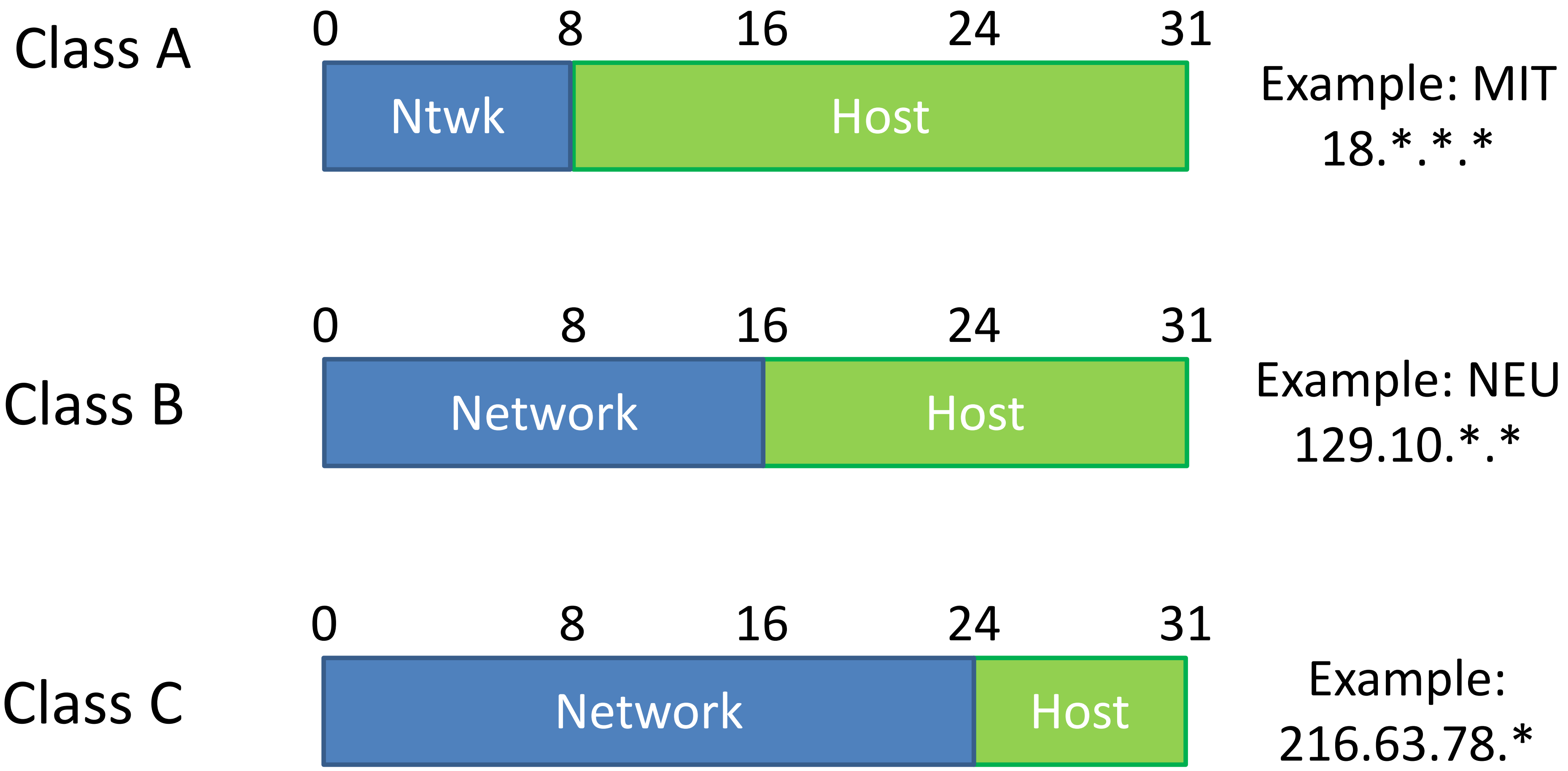


Who gets an address? How many?

- Classful Addressing
  - Class A: 8-bit prefix, 24 bits for hosts (16,777,216)
  - Class B: 16-bit prefix, 16 bits for hosts (65,536)
  - Class C: 24-bit prefix, 8 bits for hosts (256)



Classes of IP Addresses



CIDR

- Classless Inter-Domain Routing
  - Prefix (subnet) length is no longer fixed
  - (Can be division of bits rather than just 8/24, 16/16, and 24/8)

CIDR

- Classless Inter-Domain Routing
  - Prefix (subnet) length is no longer fixed
  - Address blocks come with a **subnet mask**

Classless Inter-Domain Routing (CIDR)

IP Address : 12.4.0.0   IP Mask: 255.254.0.0

Address	00001100	00000100	00000000	00000000
Mask	11111111	11111110	00000000	00000000

|← Network Prefix →|← for hosts →|

Written as 12.4.0.0/15

Use two 32-bit numbers to represent a network.
Network number = IP address + Mask

CIDR

- Classless Interdomain Routing
  - Prefix (subnet) length is no longer fixed
  - Address blocks come with a **subnet mask**
- Subnet mask written in two ways:
  - Dotted decimal: 255.255.240.0
  - /20
  - Both mean:
    11111111 11111111 11110000 00000000

/20

CIDR

- Addresses divided into two pieces:
  - Prefix portion (network address)
  - Host portion
- Given an IP address and mask, we can determine:
  - The prefix (network address) by ANDing
  - The broadcast address by ORing inverted mask

Why might a device care about its "Network or Subet Address"?

- Answers the question: is the destination on the same subnet as me?
- Address + subnet mask -> Network address
- If destination is on same network:
  - Send directly to them
- Else:
  - Send to gateway router

Network Address (Subnet Address)

IP Address & subnet mask -> Network Address

- E.g., 230.8.1.3/18 /18 => mask is 255.255.192.0

11100110	00001000	00000001	00000011	IP address
11111111	11111111	11000000	00000000	/18 Subnet mask

Network Address (Subnet Address)

- E.g., 230.8.1.3/18   /18 => mask is 255.255.192.0

11100110	00001000	00000001	00000011	IP address
11111111	11111111	11000000	00000000	/18 Subnet mask
11100110	00001000	00000000	00000000	& the two

Network address advertised by router: 230.8.0.0

Broadcast Address

- E.g., 230.8.1.3/18

11100110	00001000	00000001	00000011	IP address
~~11111111~~	~~11111111~~	~~11000000~~	~~00000000~~	/18 Subnet mask
00000000	00000000	00111111	11111111	complement of the subnet mask

Broadcast Address

- E.g., 230.8.1.3/18

11100110	00001000	00000001	00000011	IP address
00000000	00000000	00111111	11111111	complement of the subnet mask

Broadcast Address

- E.g., 230.8.1.3/18

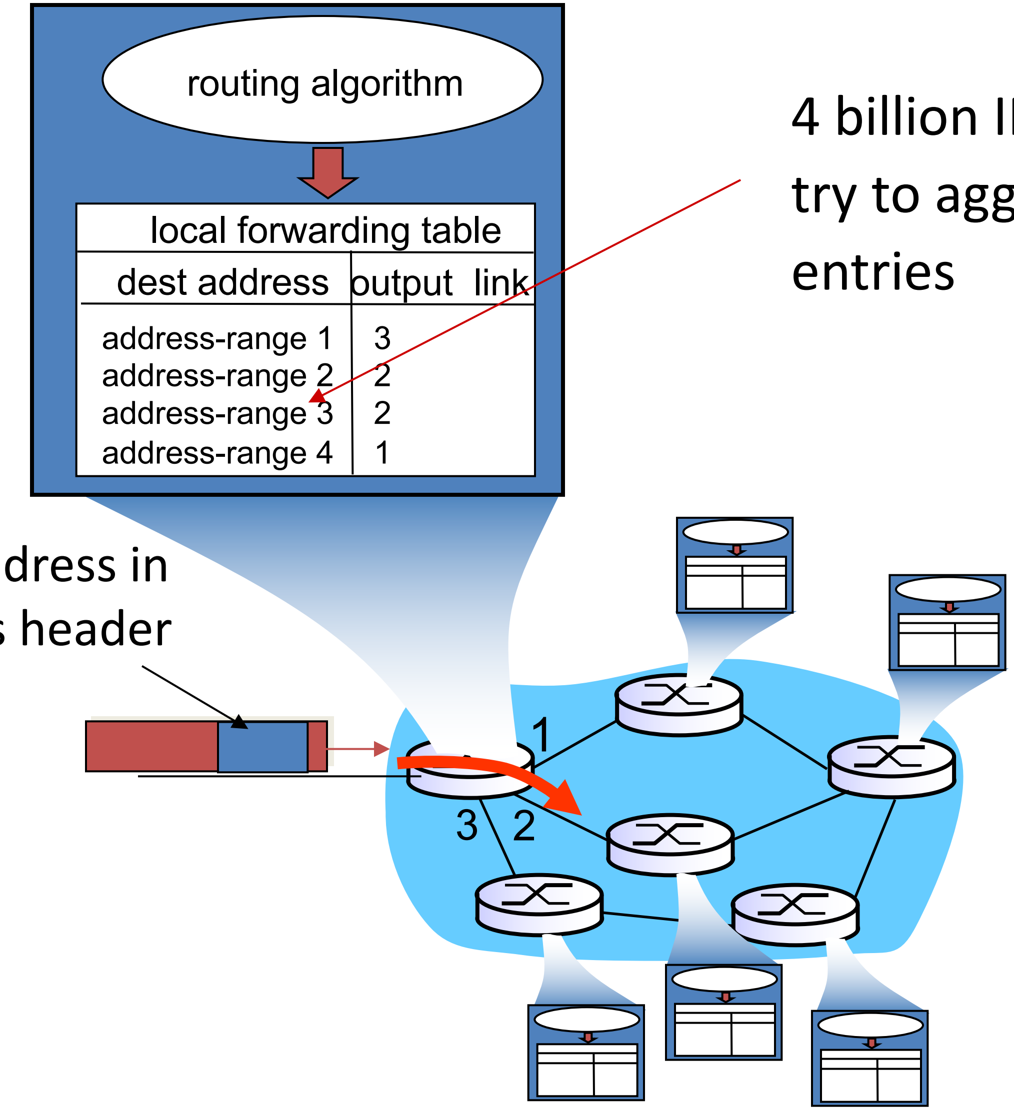
11100110	00001000	00000001	00000011	IP address
00000000	00000000	00111111	11111111	complement of the subnet mask
11100110	00001000	00111111	11111111	OR of the IP address and the complement of the subnet mask

Broadcast address: 230.8.63.255

Datagram forwarding table

4 billion IP addresses, try to aggregate table entries

IP destination address in arriving packet's header



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.

Look-up Algorithm

- Protocol: ATM (Virtual Circuits), Ethernet (Flat addresses)
  - Mechanism: Exact Match
  - Techniques: Direct lookup, Hash Tables, Binary Trees
- Protocol: IPv4, IPv6
  - Mechanism: Longest Prefix Match
  - Techniques: Prefix Trees, TCAM (Ternary Content Addressable Memories)

Datagram forwarding table

Destination Address Range	Link Interface
200.23.16.* through 200.23.23.*	0
200.23.24.0 through 200.23.24.255	1
200.23.25.* through 200.23.31.*	2
Otherwise (default gateway)	3

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

Longest prefix matching

In a forwarding table entry, use the longest address prefix that matches destination address.

Destination IP Address Range	Link interface
<upper 16 bit> 00010*** *********	0
<upper 16 bit> 00011000 *********	1
<upper 16 bit> 00011*** *********	2
Otherwise (default gateway)	3

DA: <upper 16 bits> 00011000 10101010

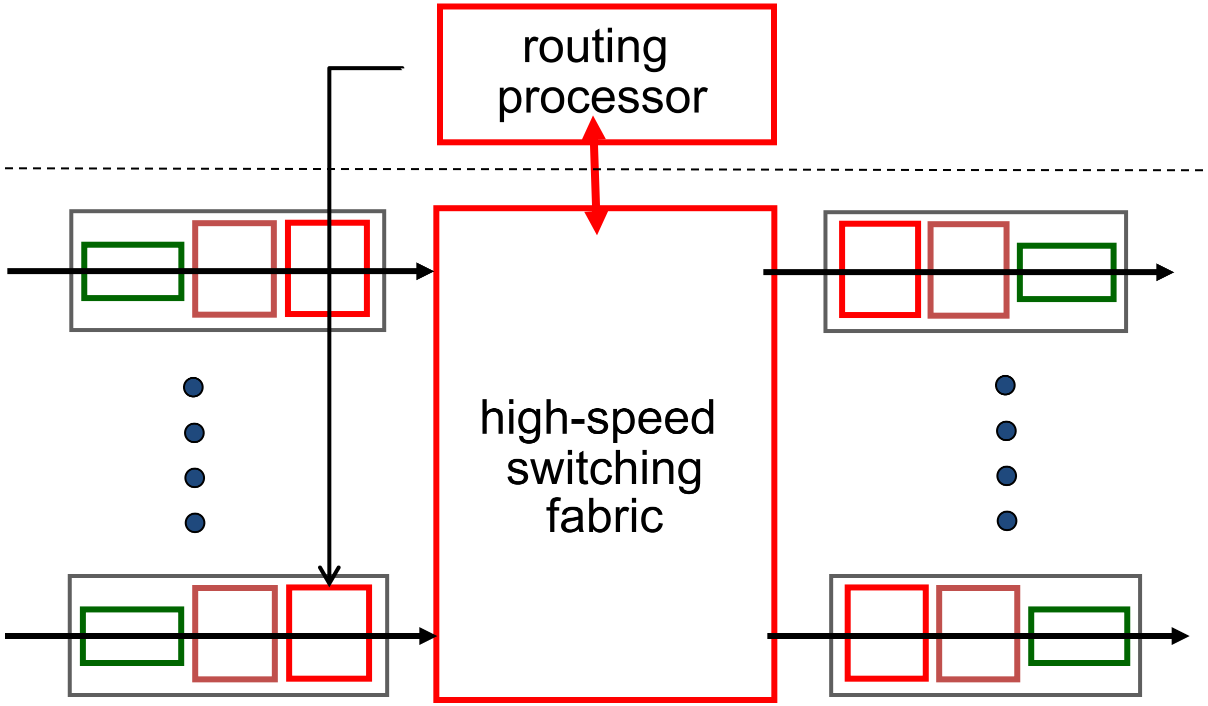
DA: <upper 16 bits> 00010110 10100001

which interface?

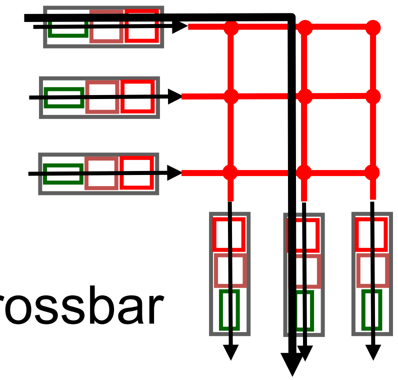
Router architecture overview

- high-level view of generic router architecture:

routing, management
control plane (software)
operates in millisecond
time frame



forwarding data plane
(hardware) operates
in nanosecond
timeframe

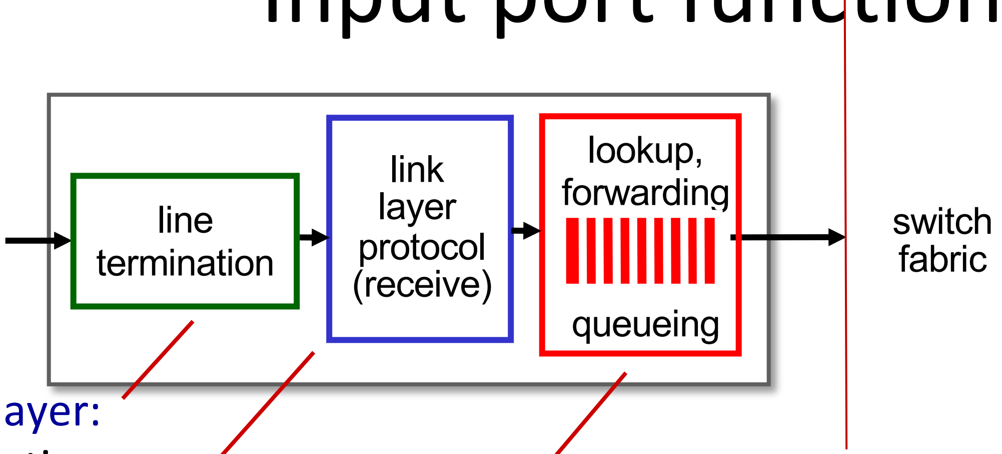


crossbar

router input ports
(different from TCP ports!!)
these are physical inputs/outputs to the router

router output ports

Input port functions



physical layer:
bit-level reception

link layer:
e.g., Ethernet
(chapter 6)

decentralized switching*:*

- using header field values, lookup output port using forwarding table in input port memory *("match plus action")*

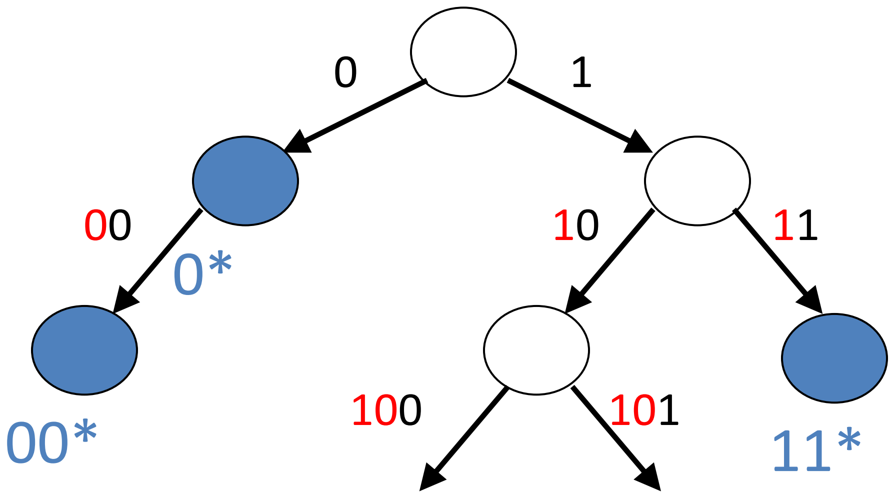
Longest prefix matching

In a forwarding table entry, use the longest address prefix that matches destination address.

Destination IP Address Range	Link interface
<upper 16 bit> 00010*** *********	0
<upper 16 bit> 00011000 *********	1
<upper 16 bit> 00011*** *********	2
Otherwise (default gateway)	3

Binary Prefix Tree

- Store the prefixes as a tree
  - Prefixes “spelled out” following a path from the root
  - One bit for each level of the tree
  - Some nodes correspond to valid prefixes
  - ... which have next-hop interfaces in a table
- When a packet arrives
  - Traverse the tree based on the destination address
  - Stop upon reaching the longest matching prefix



Prefix Range-1	0*	1
Prefix Range-2	00*	2
Prefix Range-3	11*	3

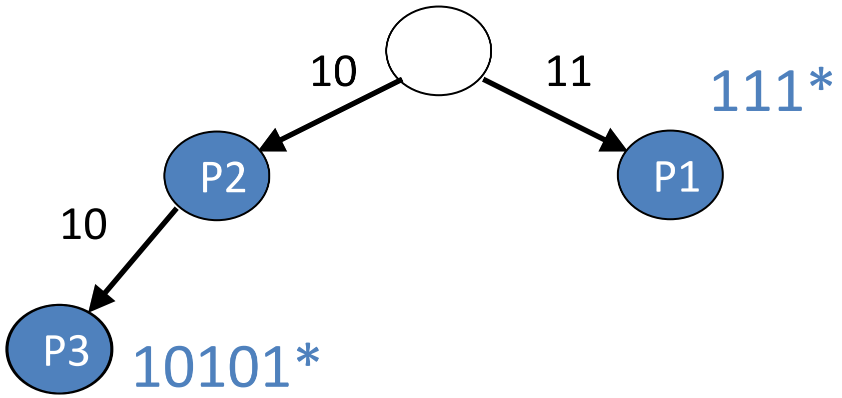
Depth = W

Degree = 2

Stride = 1 bit

Multi-bit Prefix Tree

- Store the prefixes as a tree: 4-ary tree
  - k bits for each level of the tree



Prefix Range-1	111*	1
Prefix Range-2	10*	2
Prefix Range-3	1010*	3
Prefix Range-4	10101*	4

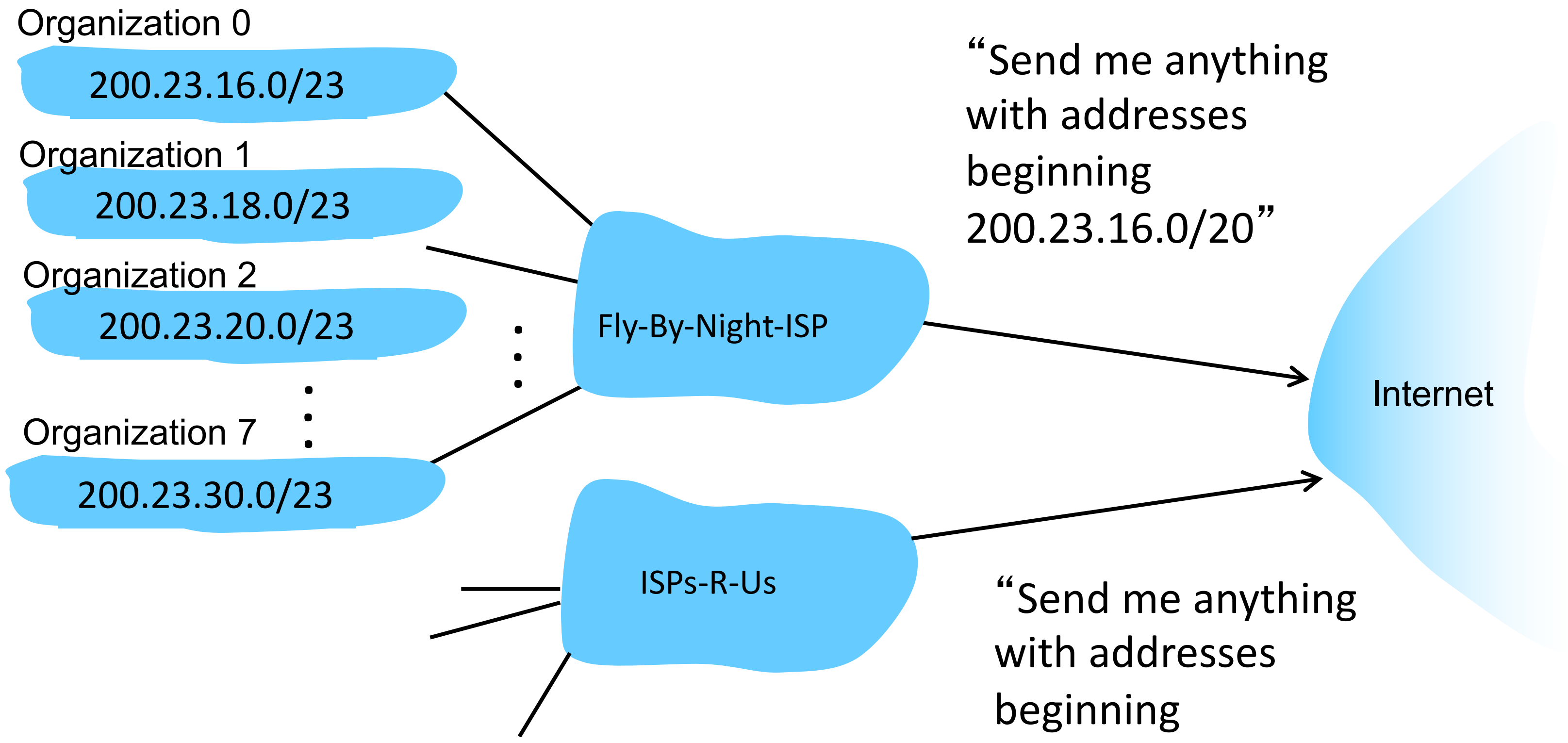
Depth = W/k
Degree = 2^k
Stride = k bits

Even Faster Lookups

- Can use special hardware
  - Content Addressable Memories (CAMs)
  - Allows look-ups on a key rather than flat address
- Huge innovations in the mid-to-late 1990s
  - After CIDR was introduced (in 1994)
  - … and longest-prefix match was a major bottleneck

Hierarchical Addressing: Route Aggregation

Hierarchical addressing allows efficient advertisement of routing information:



Hierarchical Addressing: Route Aggregation

"Send me anything with addresses beginning 200.23.16.0/20" translates to the following:

	200	23	16	0	
200.23.16.0/20 =	11001000	00010111	00010000	00000000	/20 Prefix

/20 prefix (first 20 bits)

200.23.16.0 =	11001000	00010111	00010000	00000000	Range represented by the /20 prefix
|					
200.23.31.255 =	11001000	00010111	00011111	11111111	

/20 prefix contains the range of IP addresses that match the the first 20 bits, and can have any value for the remaining 12 bits in the range of :

[first 20 bits] 0000 00000000

[first 20 bits] 1111 11111111

A total of 2^12 = 4,096 IP addresses

Route aggregation in Fly-By-Night ISP

Fly-By-Night-ISP

200.23.16.0/20 = 11001000 00010111 00010000 00000000

Individual Organizations: All of these organizations IP addresses lie withinFly-by-Night's /20 prefix (first 20 bits are the same)

- they more specifically match on the three more bits to form a /23 prefix (first 23 bits of all IP addresses within their organization are the same).
- The last 9 (32-23) bits provide 2^9 = 512 unique IP addresses within each organization.

/23 prefixes

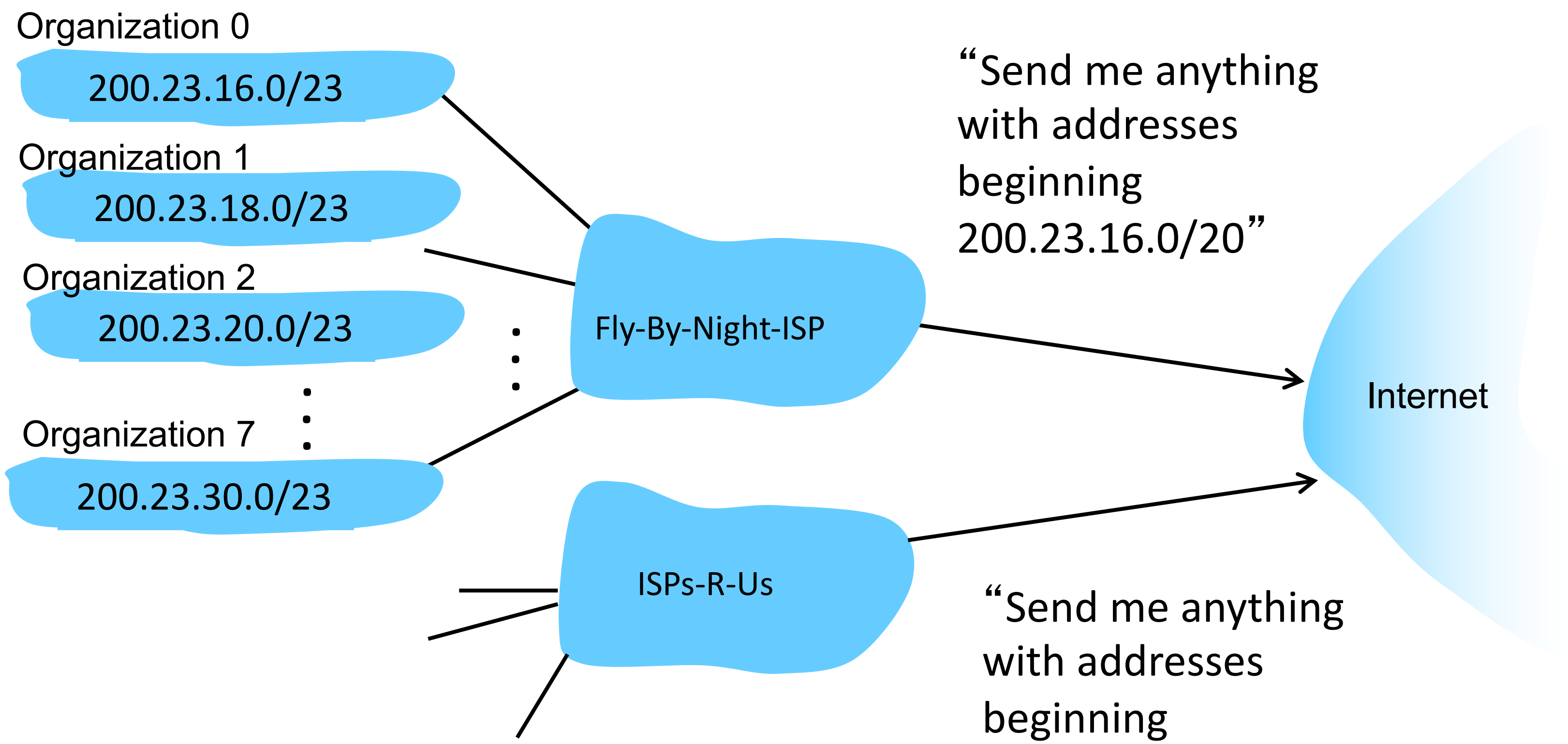
200.23.16.0/23 = 11001000 00010111 00010000 00000000

200.23.18.0/23 = 11001000 00010111 00010010 00000000

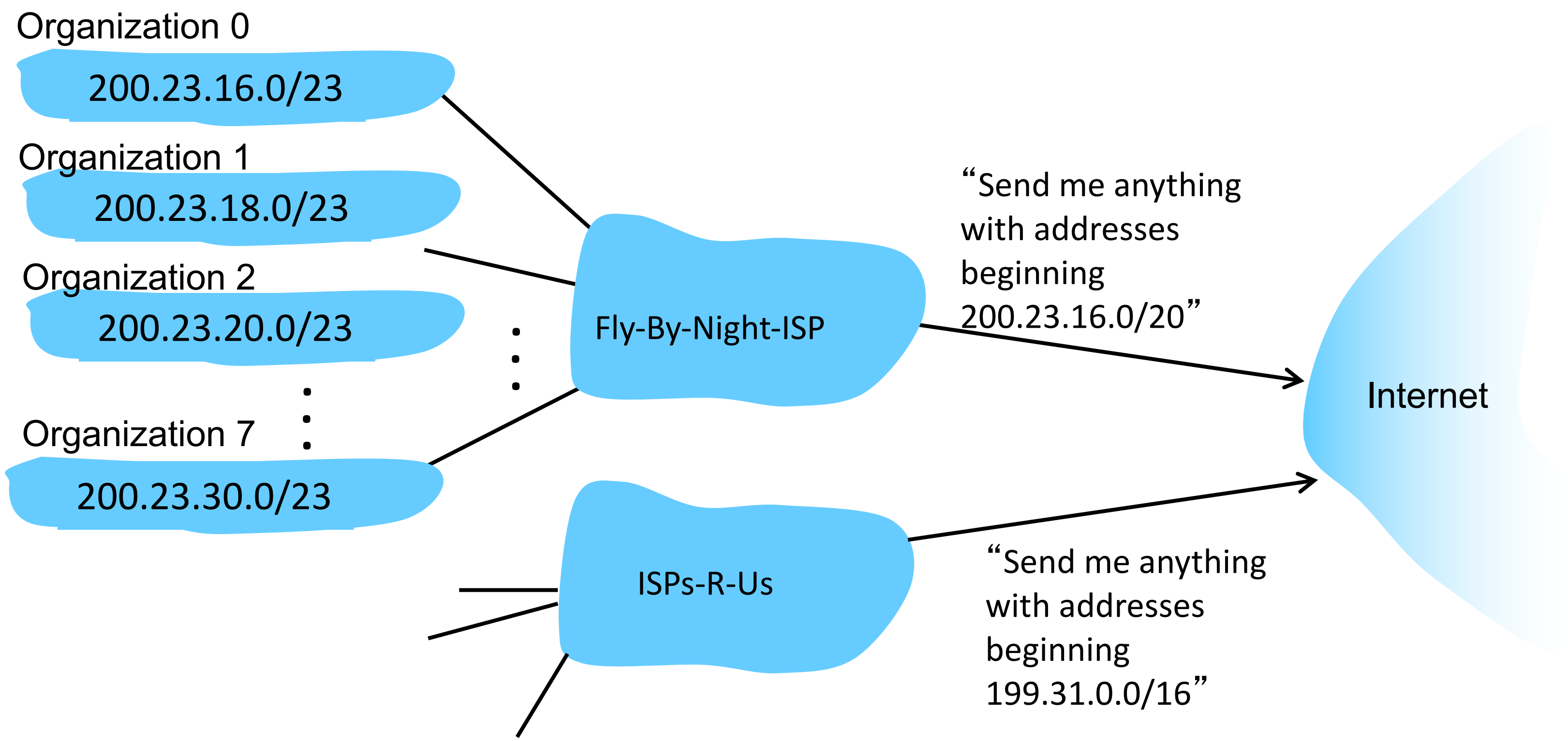
200.23.20.0/23 = 11001000 00010111 00010100 00000000

200.23.30.0/23 = 11001000 00010111 00011110 00000000

What should we do if organization 1 decides to switch to ISPs-R-Us?



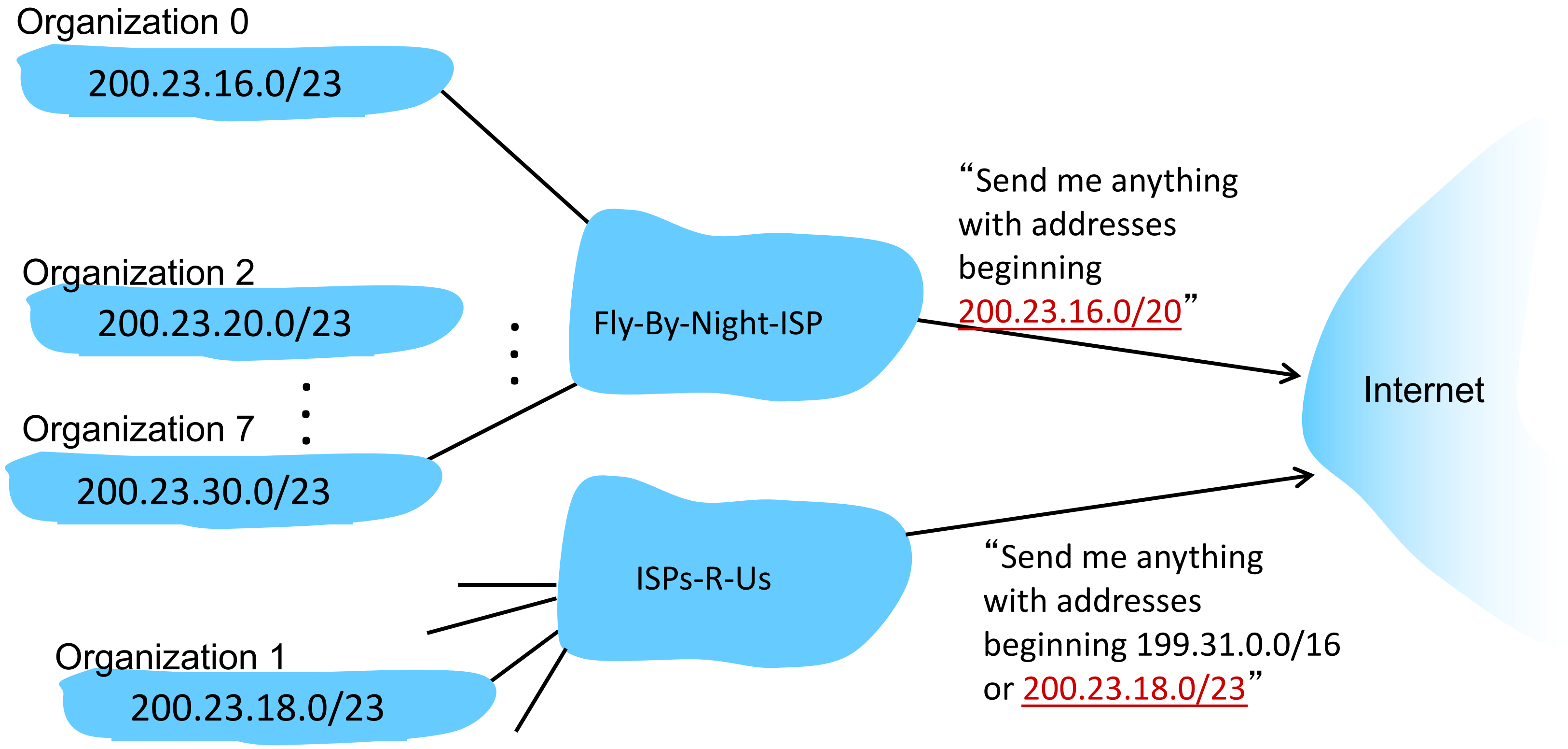
What should we do if organization 1 decides to switch to ISPs-R-Us?



A. Move 200.23.18.0/23 to ISPs-R-Us (and break up Fly-By-Night's /20 block).
B. Give new addresses to Organization 1 (and force them to change all their addresses).
C. Some other solution.

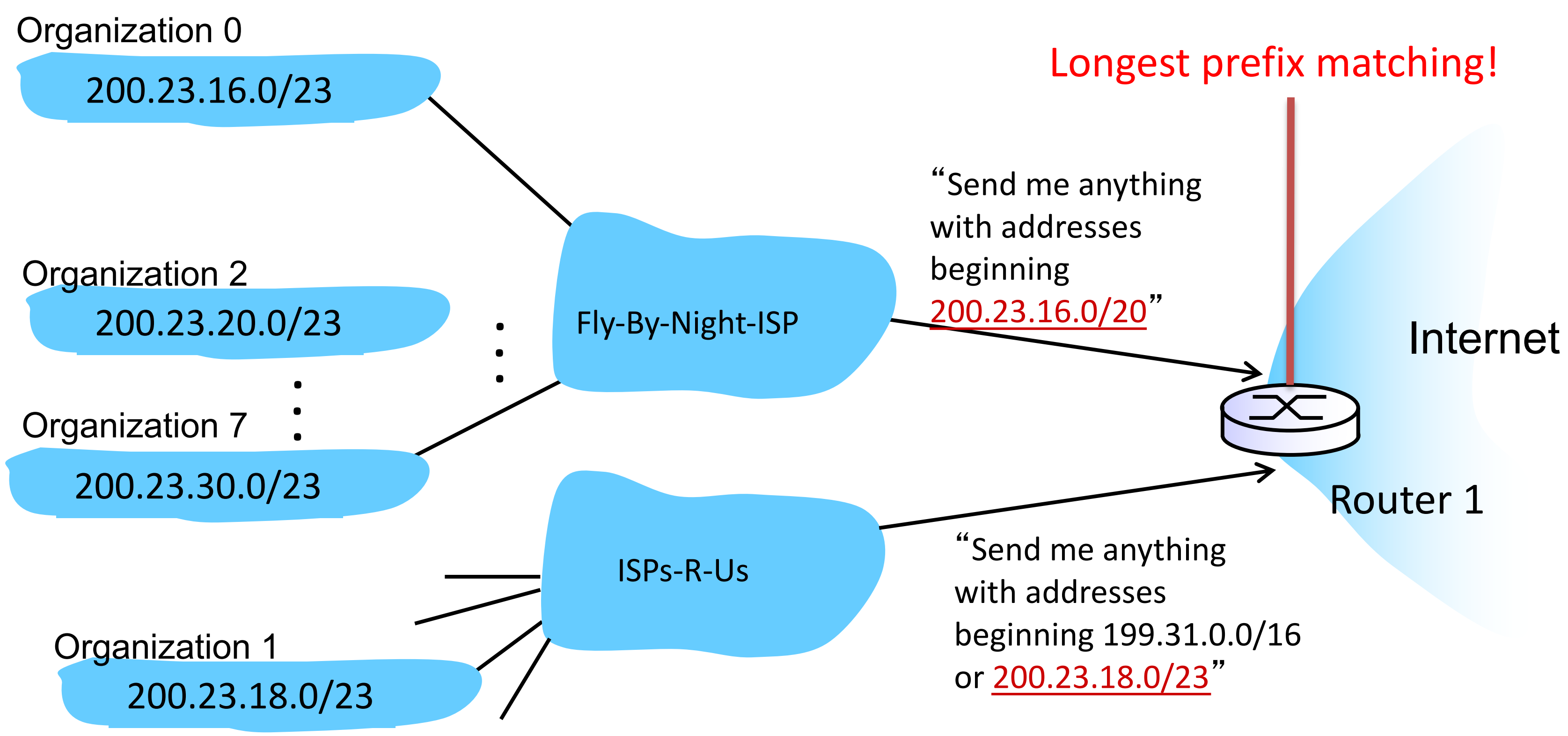
Hierarchical addressing: More Specific Routes

ISPs-R-Us has a more specific route to Organization 1

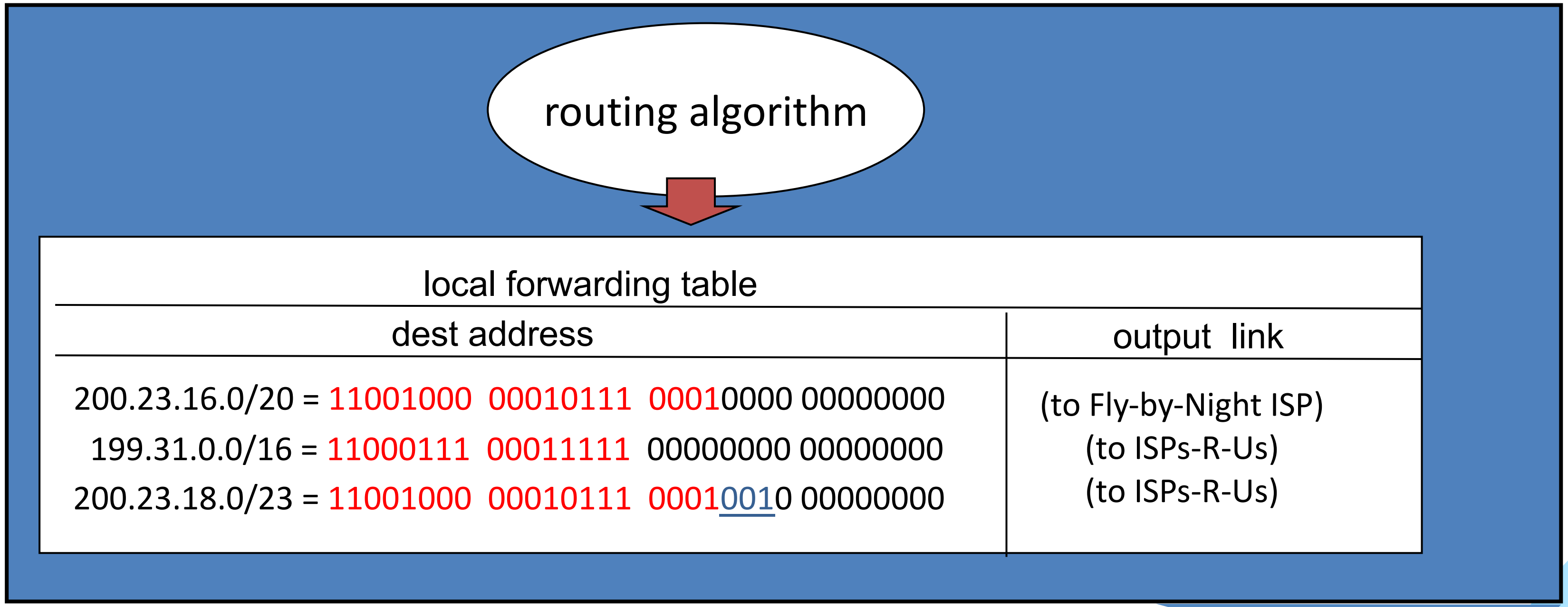


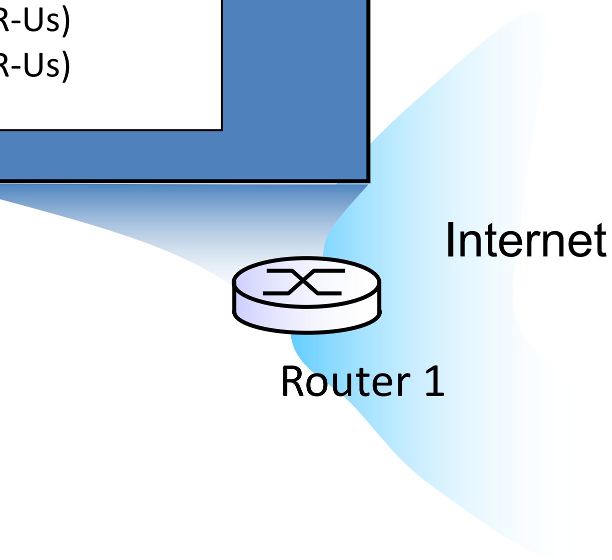
Hierarchical addressing: More Specific Routes

ISPs-R-Us has a more specific route to Organization 1



Longest Prefix Matching at Router 1





Now, when an incoming packet addressed with destination address 200.23.18.5 arrives – this address belongs to Organization 1 and the packet will be matched using longest prefix matching and will be routed to ISPs-R-Us rather than the Fly-by-Night ISP.