

Problem / Overview

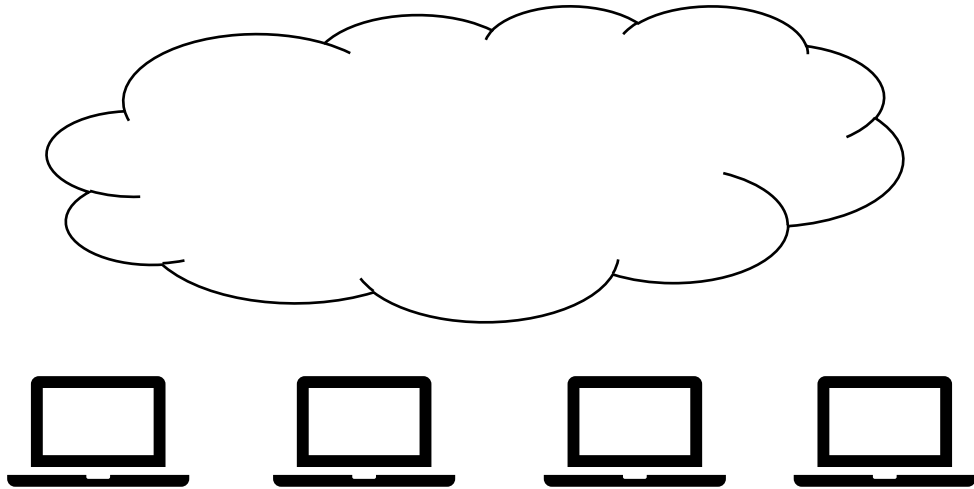
Course: Networking Fundamentals
Module: Network



University of Colorado **Boulder**

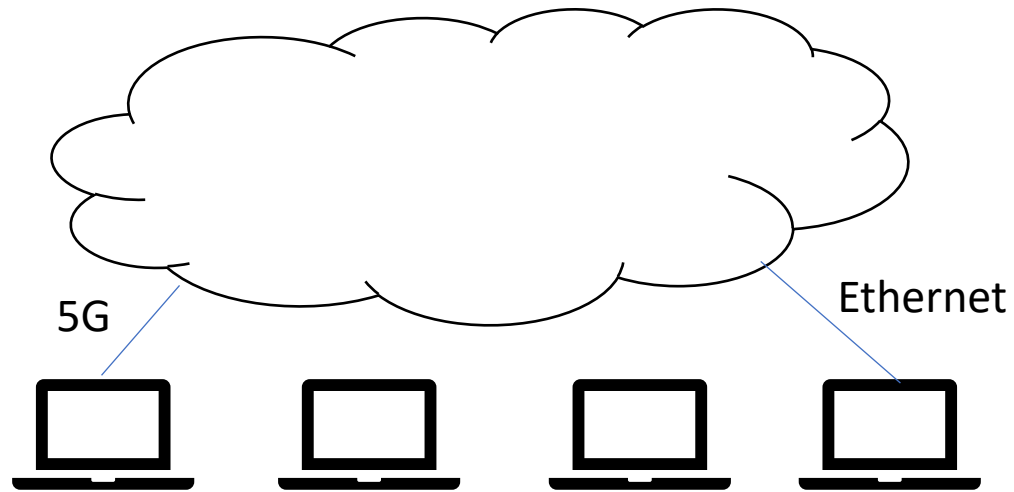
Intro to the Network

Link layer recap: Structuring data, handling errors, providing for multiple access, and scaling beyond single links.



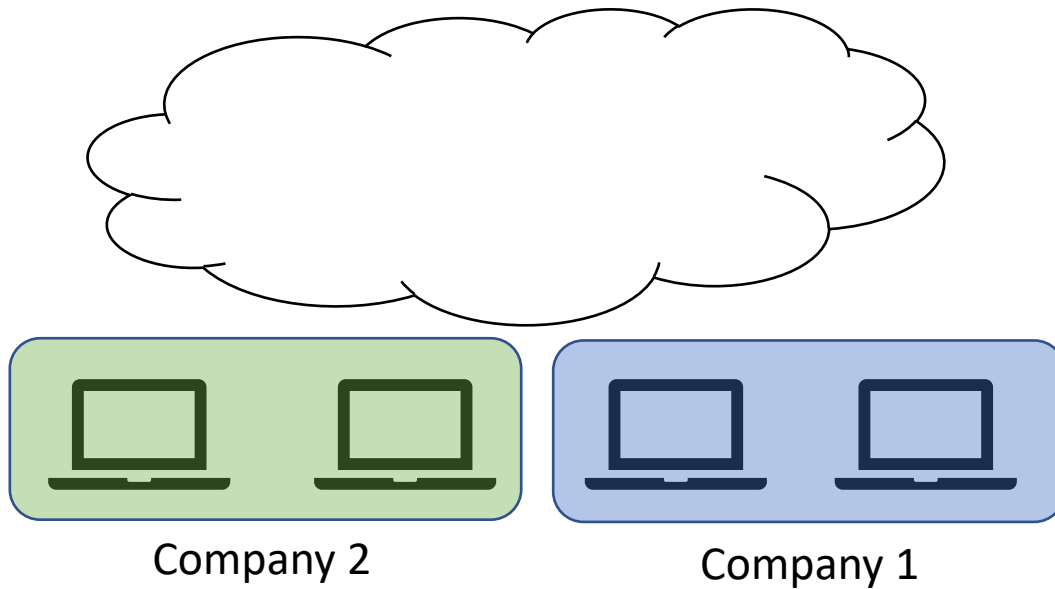
What if the nodes are on different links?

e.g., data is structured differently, error handling is different



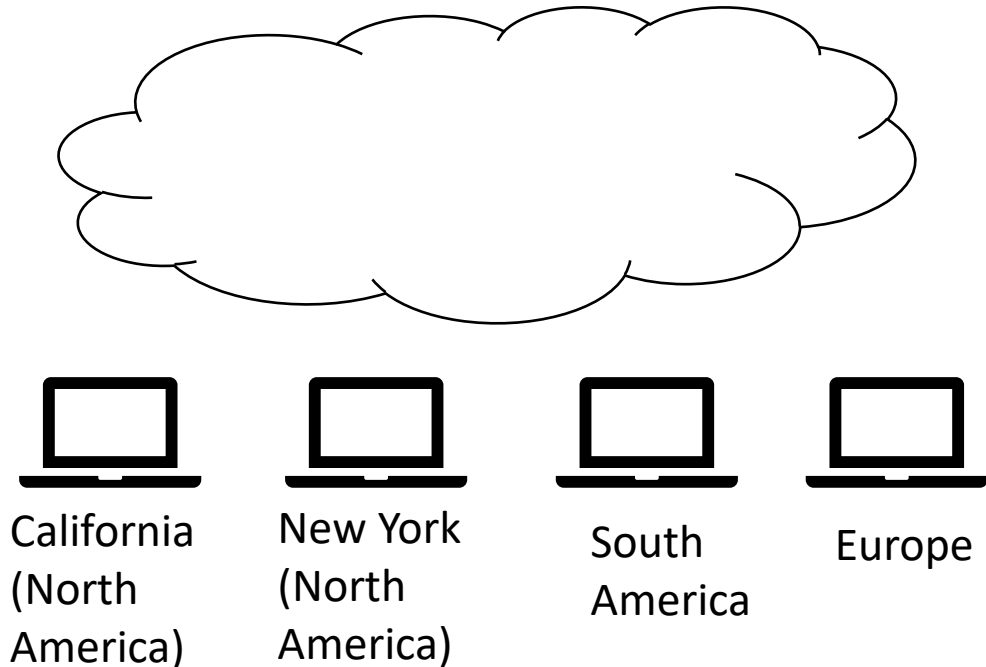
What if the nodes have different admins?

e.g., does company 1 really want company 2's broadcast traffic, do they agree on how and when to access the network



What if the nodes are in different locations?

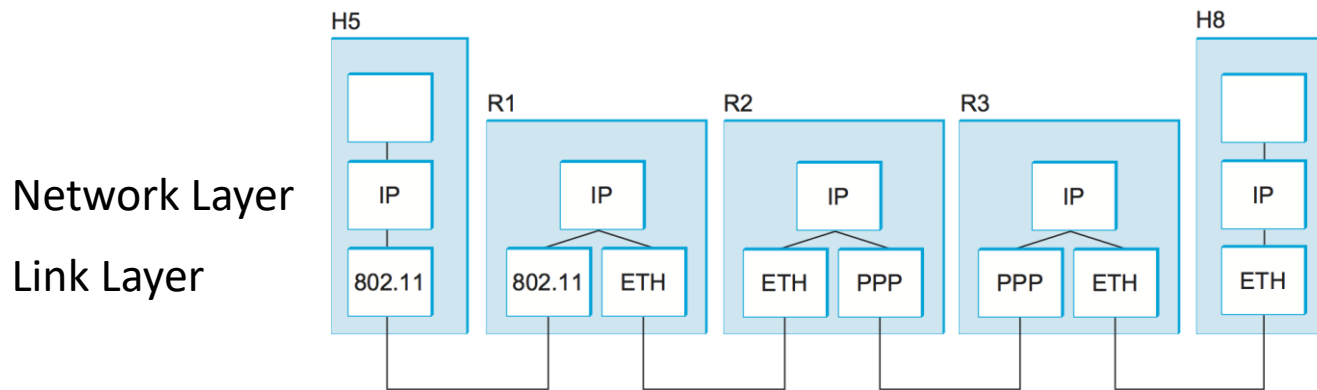
e.g., are MAC table look ups (and learning) going to scale to billions of devices, are broadcasts going to work globally at that scale



Network Layer

The assumptions at the link layer, do not work here

Introduce a new layer, the Network Layer, that addresses these needs



Network Layer Overview

- How do we address nodes scalably across different networks
- Given a destination address, how do we forward it (scalably)
- How do the different networks coordinate, so each can reach destinations beyond their boundaries
- How are addresses assigned and mapped to the link layer address
- What tools can an admin use to troubleshoot beyond its network boundaries





University of Colorado **Boulder**

Internet Protocol

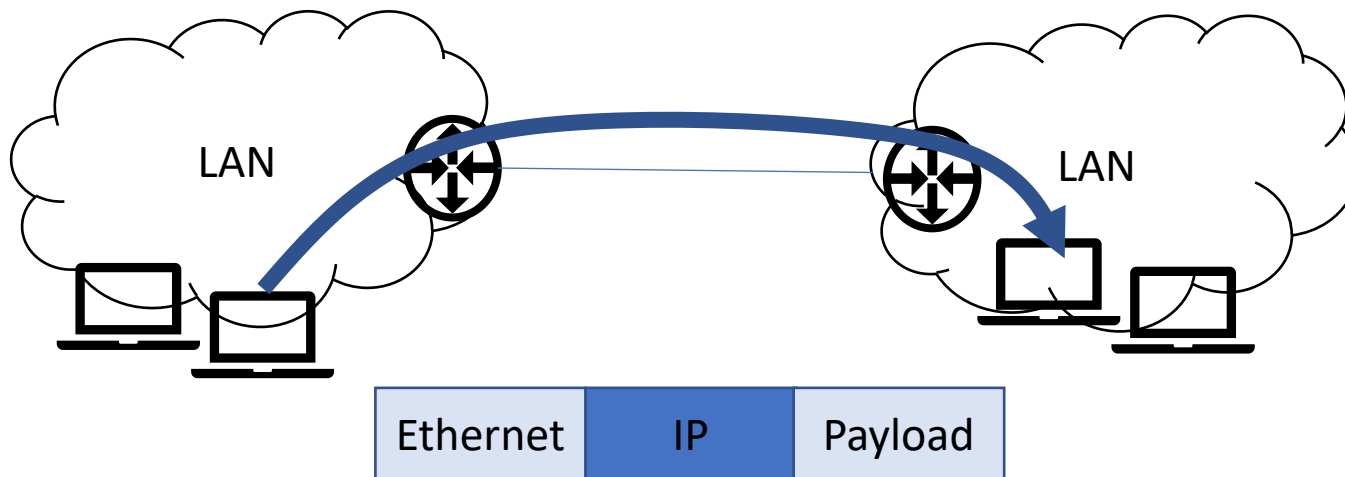
Course: Networking Fundamentals
Module: Network



University of Colorado **Boulder**

Internetworking

- Enable communication between multiple, heterogeneous networks
- Key: Router at edge of each network
(called Gateway in 1974 paper from Cerf & Kahn “A Protocol for Packet Network Intercommunication”)



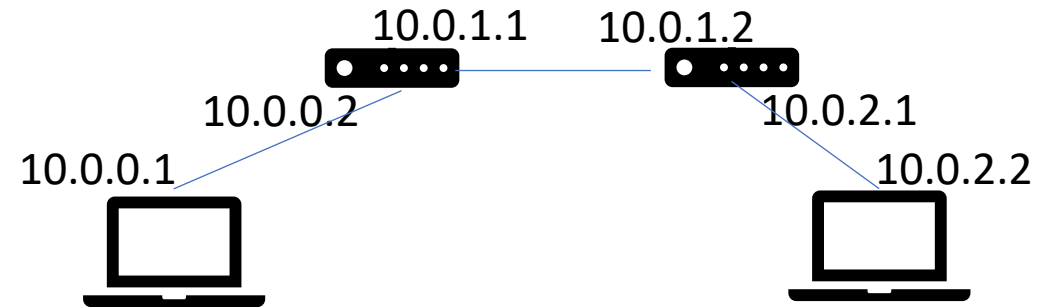
Service Model of the Internet Protocol

- Best Effort
 - Delivery – packets may get dropped
 - Timing – no guarantee on how long it takes to deliver a packet
 - Order – packets may get re-ordered in the network
- Reasoning: inter-connecting different link layers, so needs to be lowest common denominator



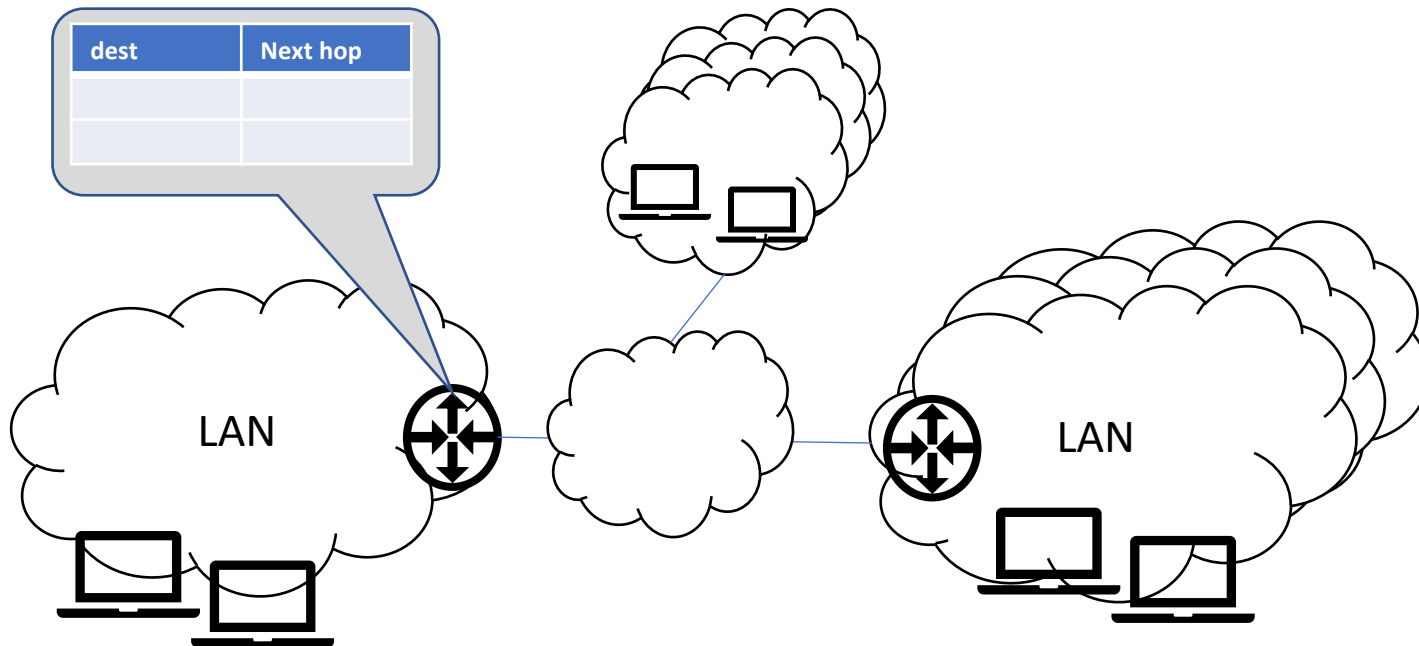
Addressing in IP

- Each interface gets an IP address
- IPv4 - 32 bits in dotted decimal
 - 192.168.0.1
- IPv6 - 128 bits in hextets (16 bits) separated by a colon
 - `2001:0db8:0000:0000:0000:ff00:0042:8329`
 - *Can drop leading zeros*
`2001:0db8:0:0:0:ff00:0042:8329`
 - *Can replace consecutive zeros with ::*
`2001:0db8::ff00:0042:8329`



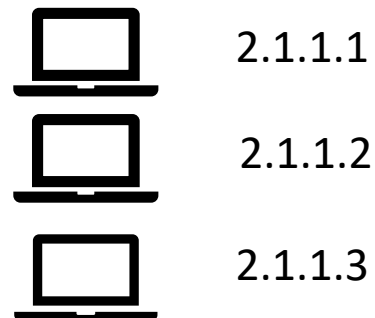
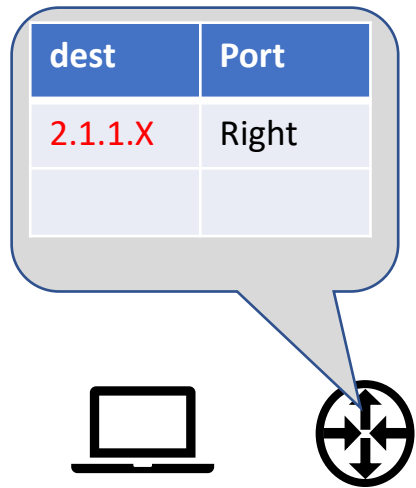
Structure Adds Scalability

- In Ethernet we saw tables had 1 entry per destination
- That doesn't scale to billions of devices worldwide
- Solution – add structure to addresses so devices can be grouped



Subnet – a set of consecutive IP addresses

- High order bits in a subnet all the same (2.1.1 in picture)
- Low order bits identify host uniquely within that network (.1, .2, .3)
- Now, just need to know how to get to the subnet



Recall – IPv4 address is 32 bits:
00000010_00000001_00000001_00000001

A red bracket underlines the first 8 bits (00000010) of the binary string, and a red circle highlights the first bit (2) of the first octet in the dotted decimal representation.

2 . 1 . 1 . 1

CIDR (pronounced cider)

- Classless Inter-Domain Routing
- Format: a.b.c.d/x
(also called a prefix)
- X is the length of the prefix – num. high order bits that are the same
- a.b.c.d are values which are the same (use 0 for low order bits)

Prefix: 2.1.1.0/24



2.1.1.0



2.1.1.1

...

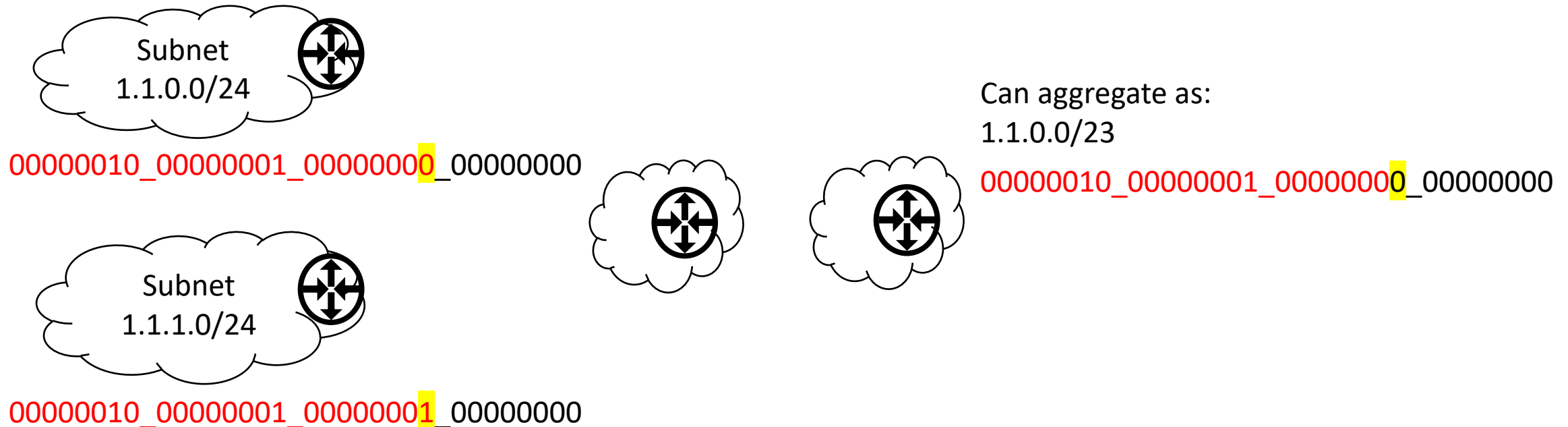


2.1.1.255



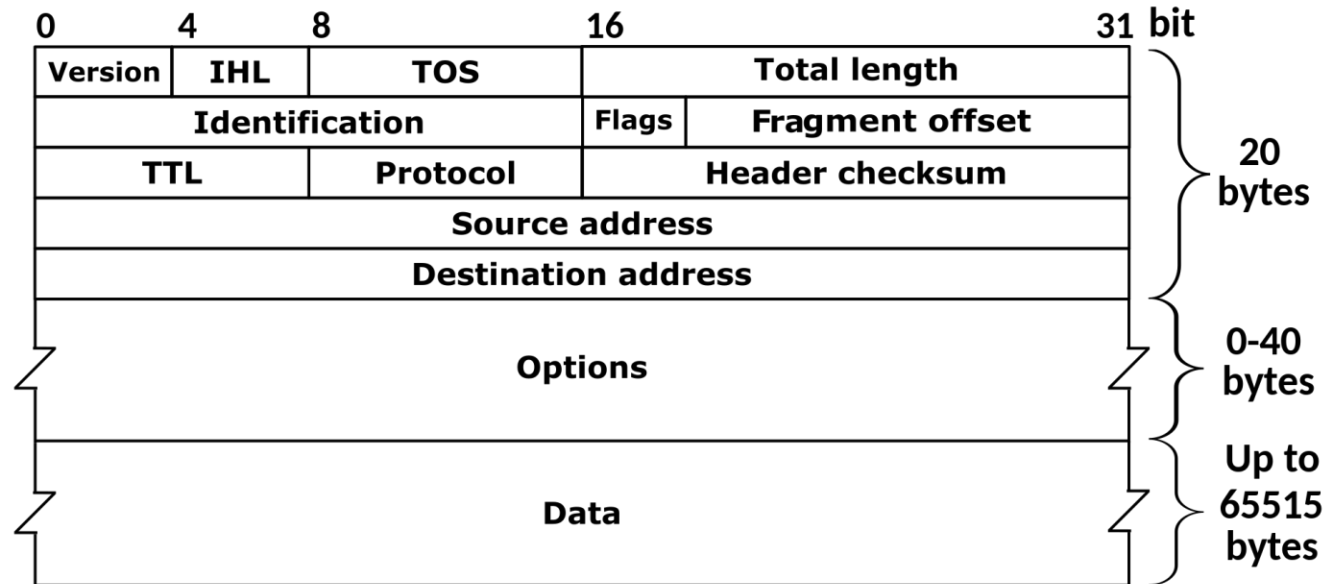
Hierarchical

- Two subnets can combine
- $1.1.0.0/24$ and $1.1.1.0/24 = 1.1.0.0/23$ (show as picture)



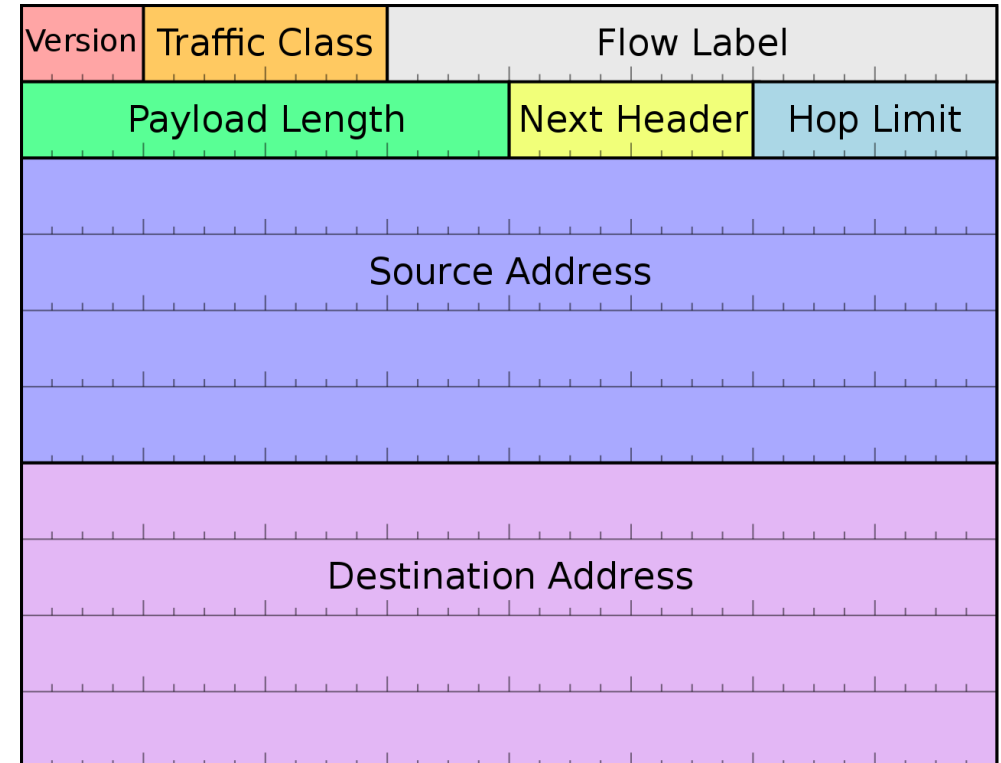
Full Header

IPv4



- Note Source, Destination
- Note TTL / Hop Limit
- Note Protocol / Next Header
- Note header checksum

IPv6





University of Colorado **Boulder**

Router Data Plane

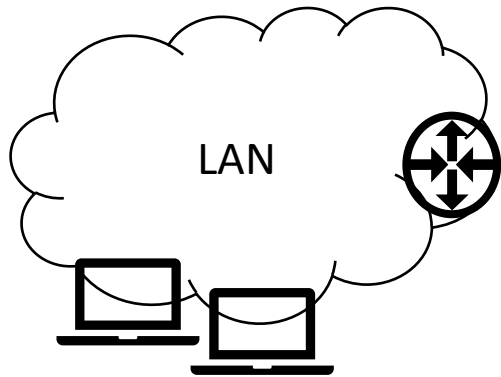
Course: Networking Fundamentals
Module: Network



University of Colorado **Boulder**

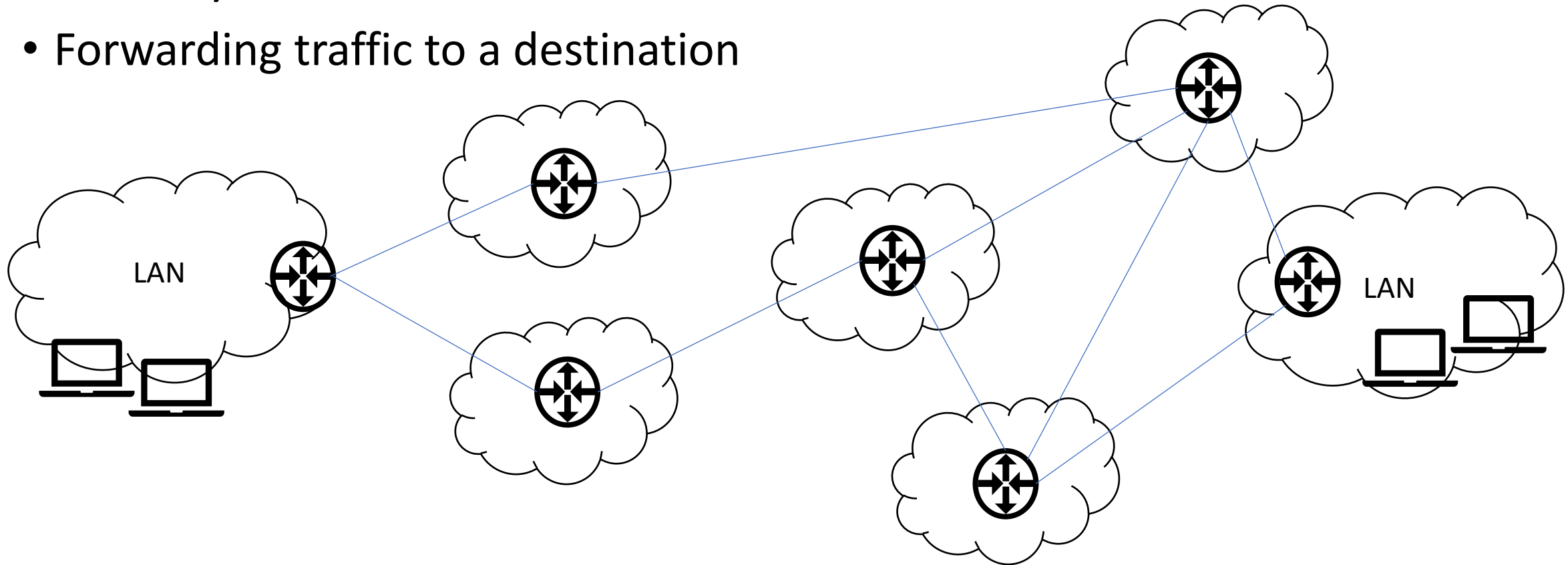
The Role of a Router

- Gateway



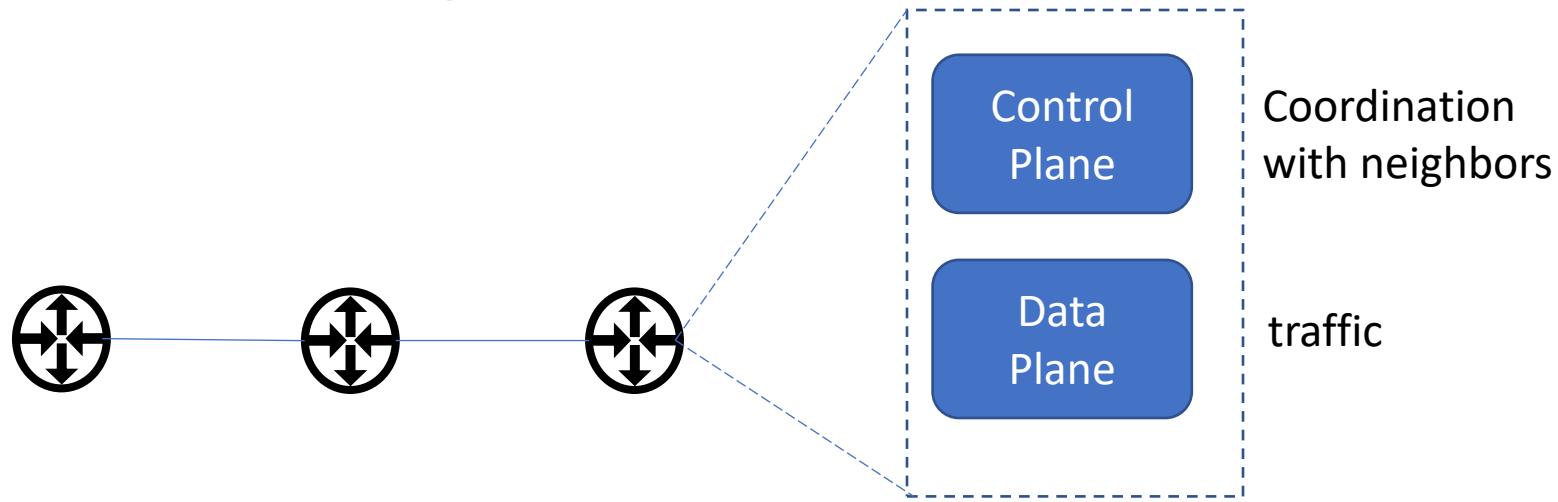
The Role of a Router

- Gateway
- Forwarding traffic to a destination



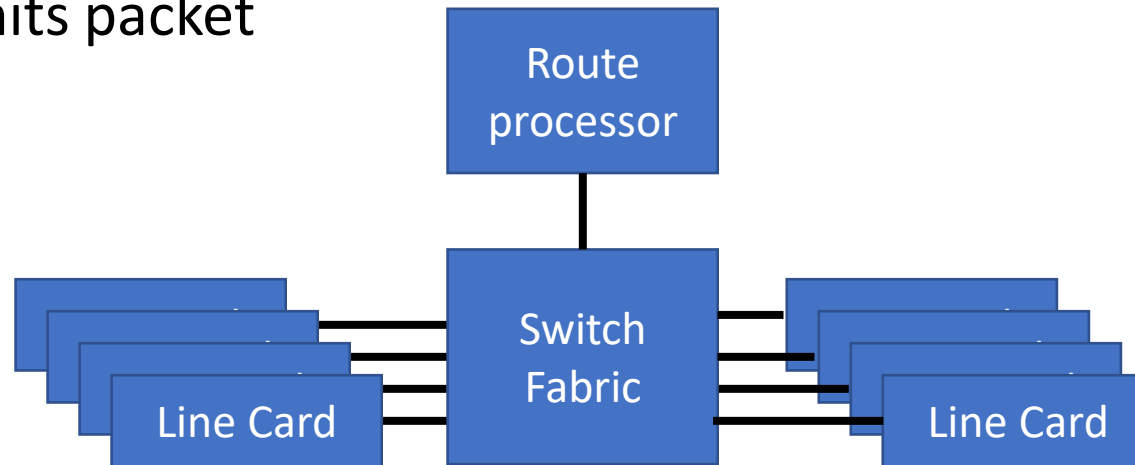
Forwarding vs Routing

- Forwarding: Data plane
 - Direct a data packet to an output port/link
 - Uses a forwarding table
- Routing: Control plane
 - Computes paths by coordinating with neighbors
 - Creates the forwarding table



Packet Forwarding

- Control plane (route processor) calculates the forwarding table
- Data plane – life of packet
 - Received at ingress of line card
 - Lookup destination in forwarding table (determines output port)
 - Send packet over switch fabric to output port
 - Line card transmits packet



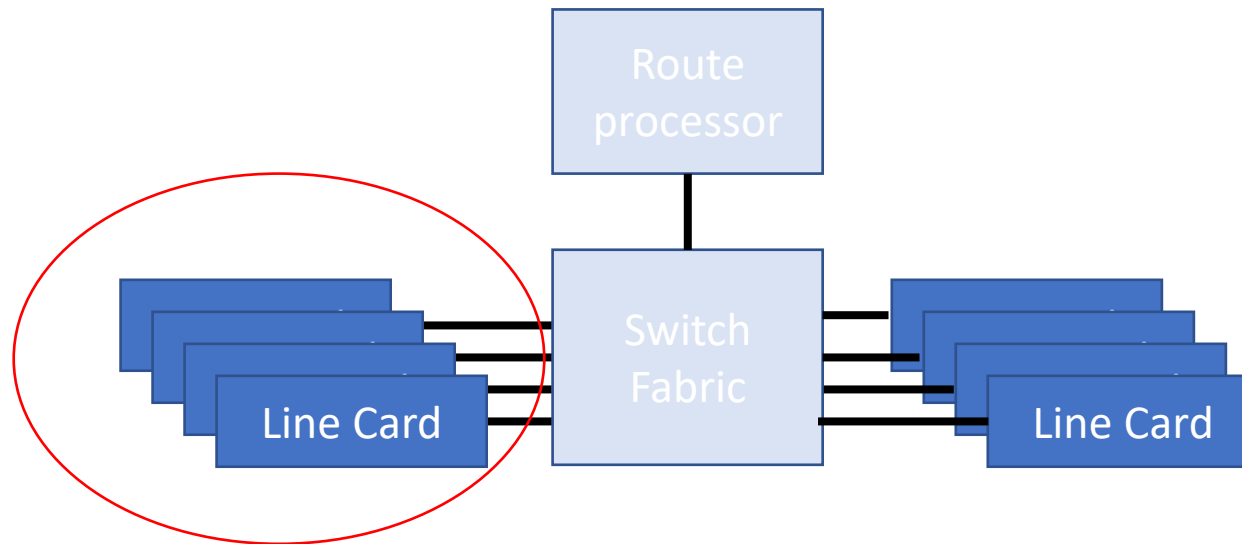
Data Plane Processing

- Processing a packet
- Switch fabric



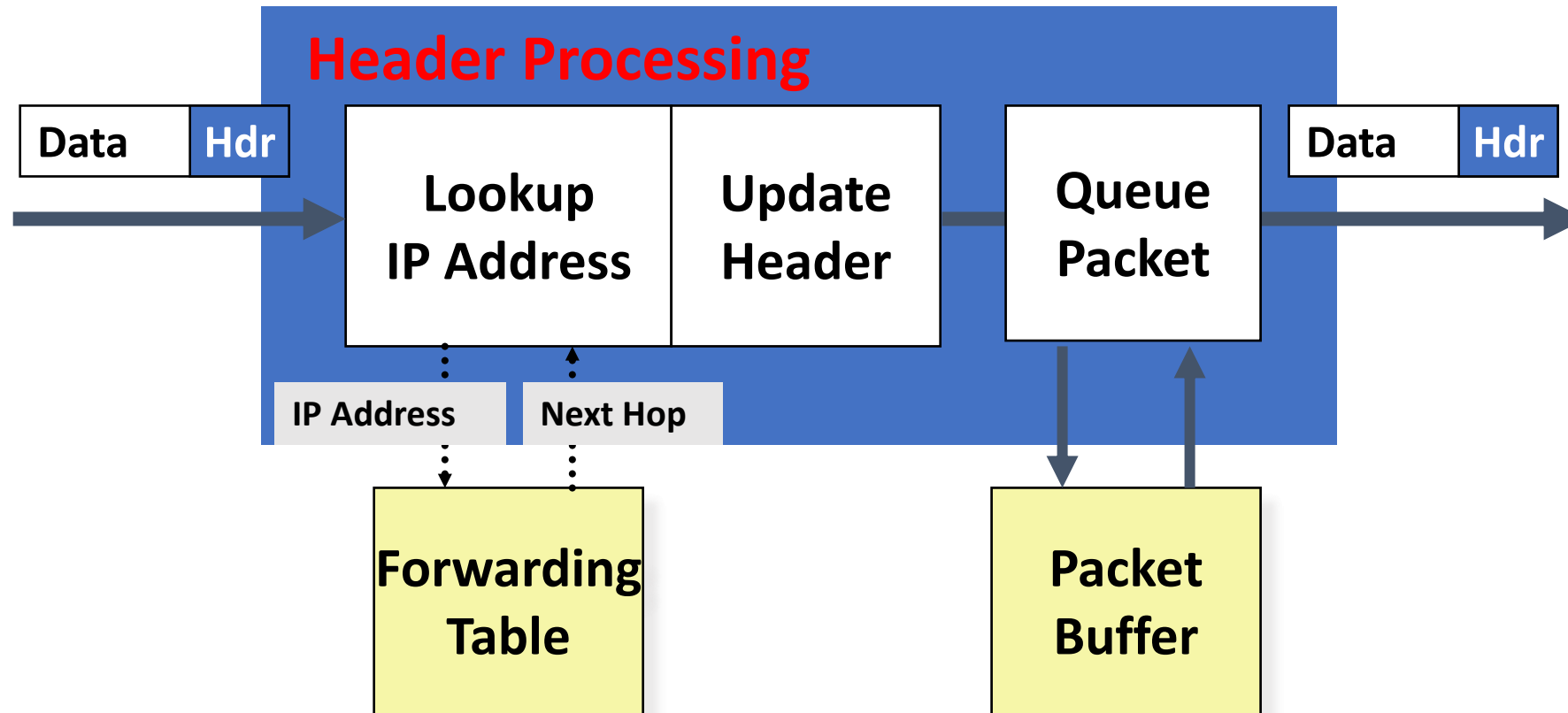
Processing

- Processing a packet
- Switch fabric



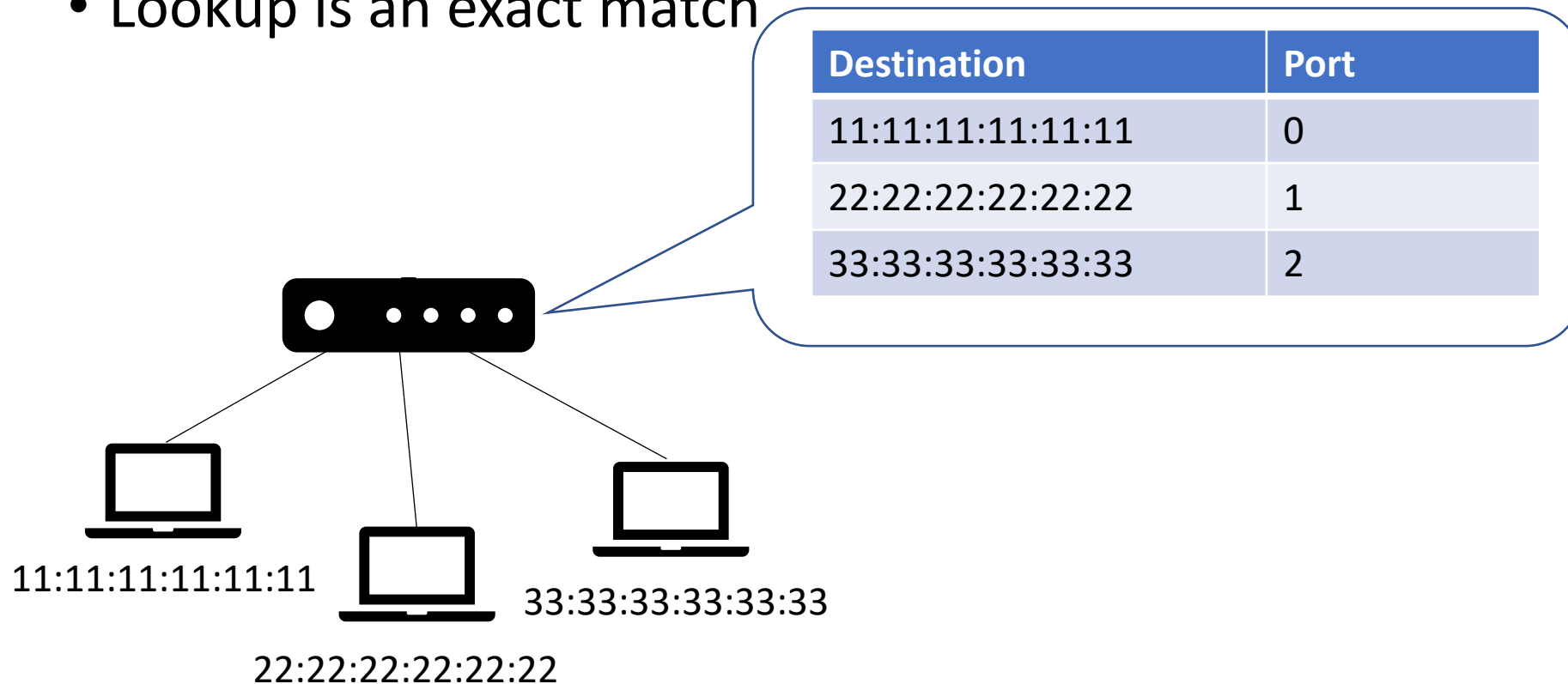
Generic Router Architecture

- Key challenge - Lookup



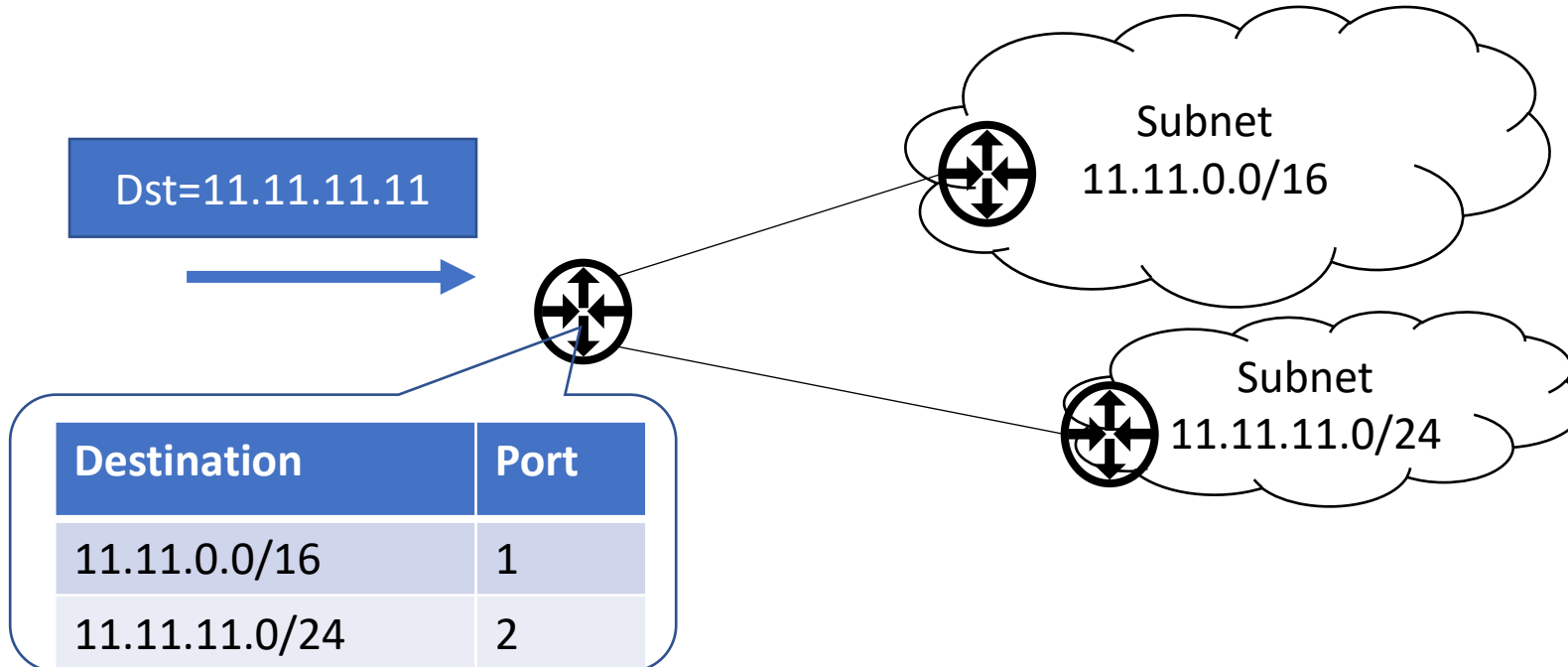
Recall: Ethernet – Match on Destination MAC

- Addresses structure around vendors
- Lookup is an exact match



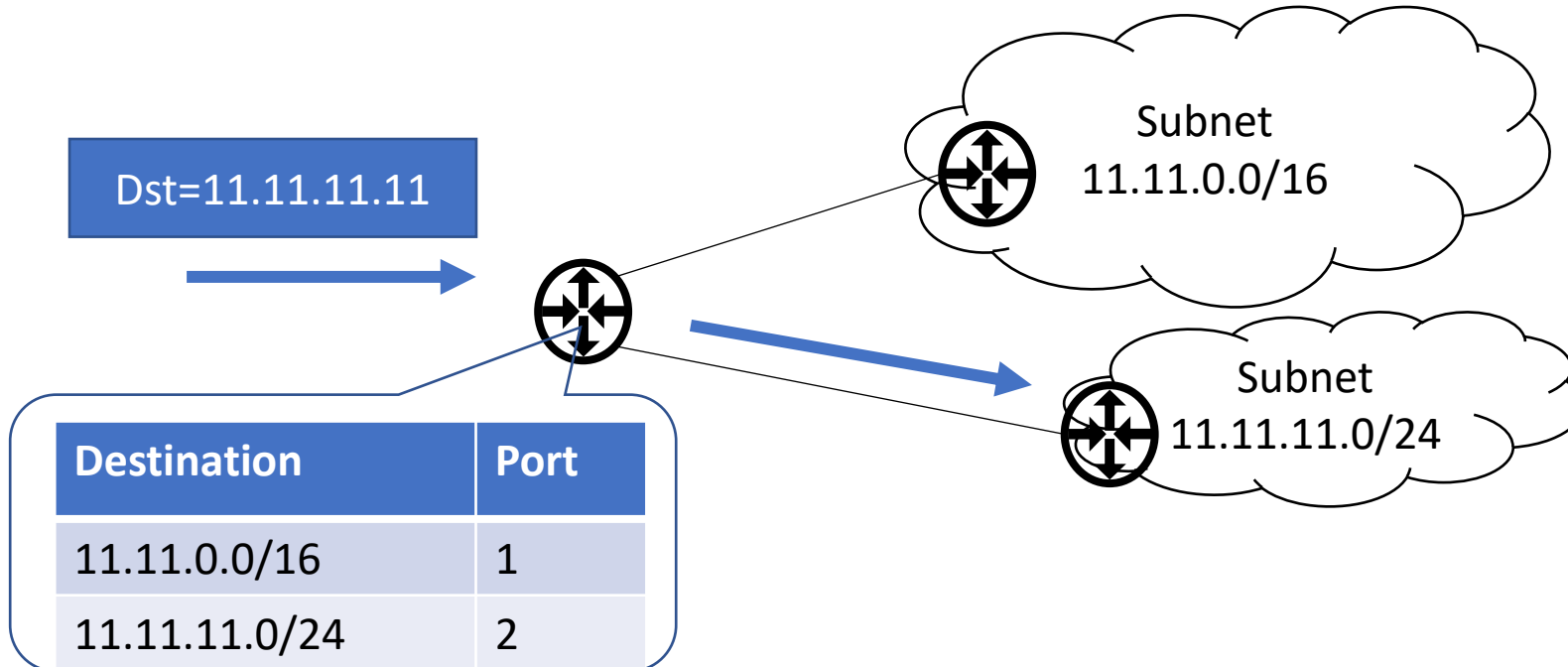
IP – Match on IP Prefix

- Variable length prefixes
- Prefixes may overlap



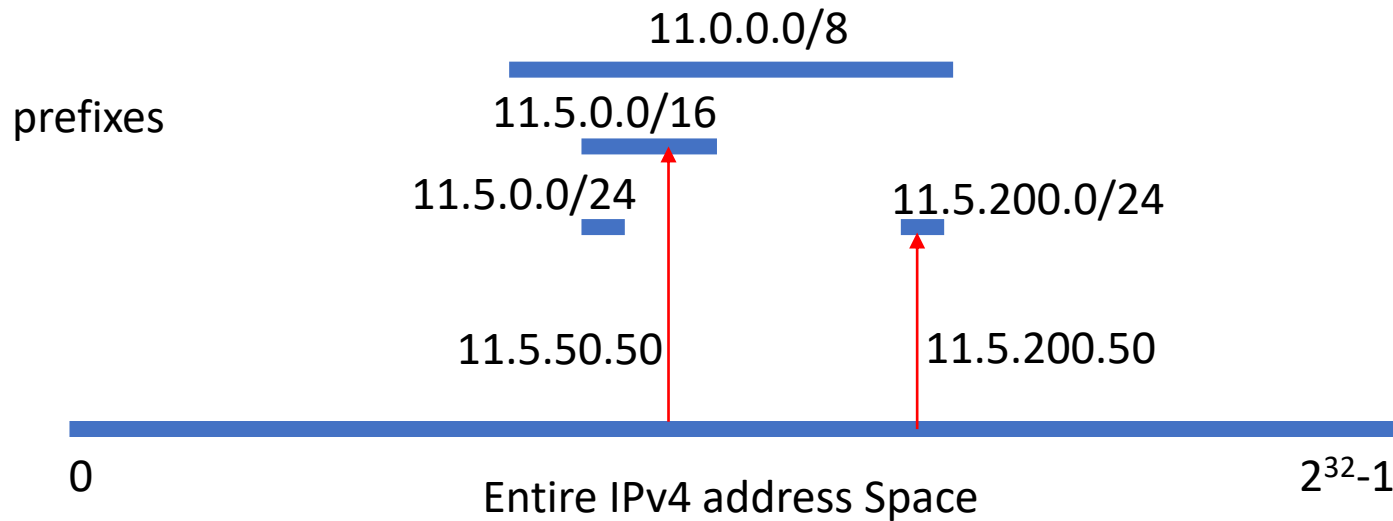
Longest Prefix Match (LPM)

- Find the most specific prefix that matches the destination



Longest Prefix Match (LPM)

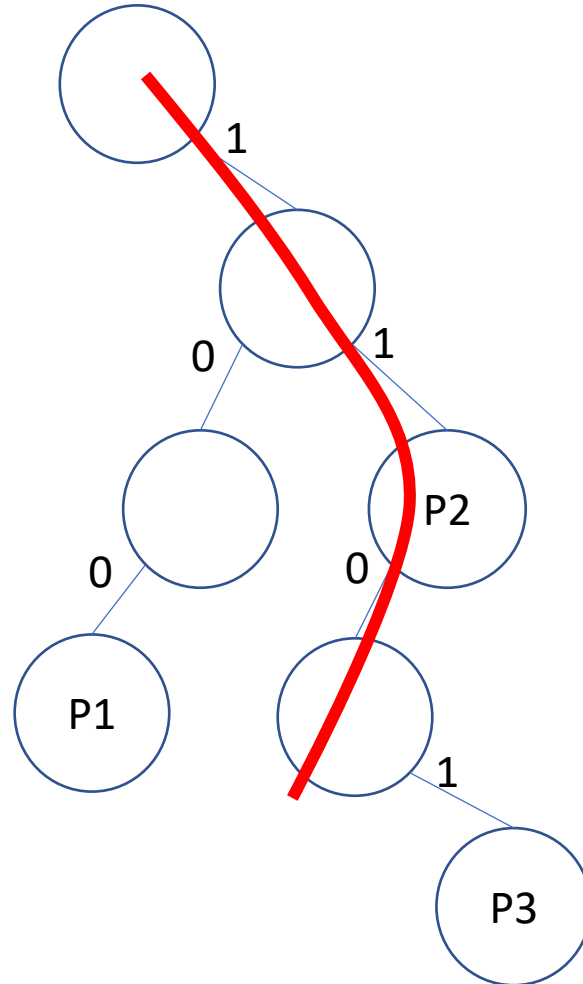
- Find the most specific prefix that matches the destination



Address Lookup using Tries

| | | |
|----|-------|--------|
| P1 | 100* | Port 0 |
| P2 | 11* | Port 3 |
| P3 | 1101* | Port 7 |

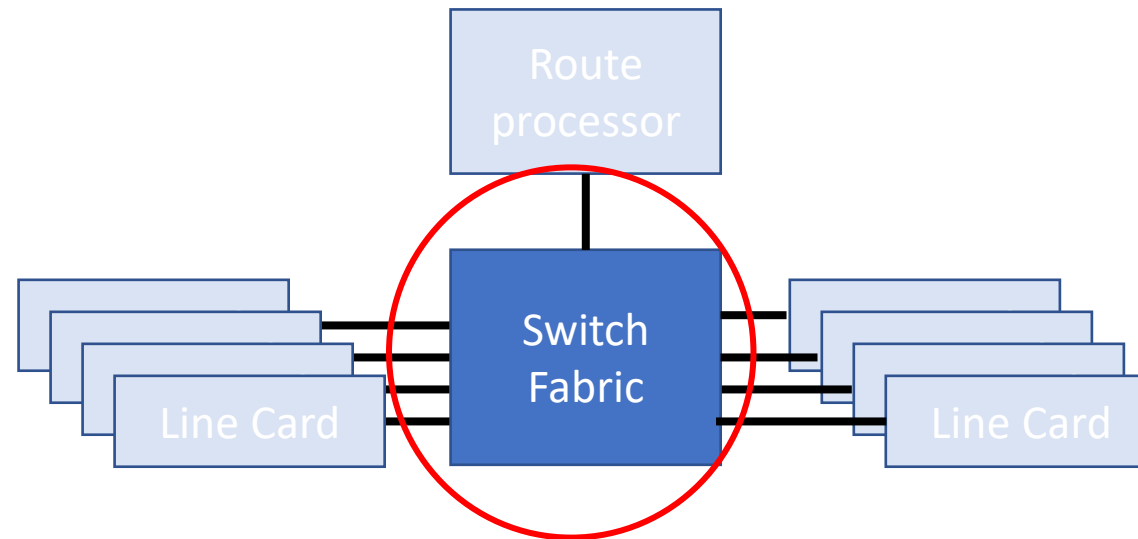
Lookup: 11001111
Match: P2



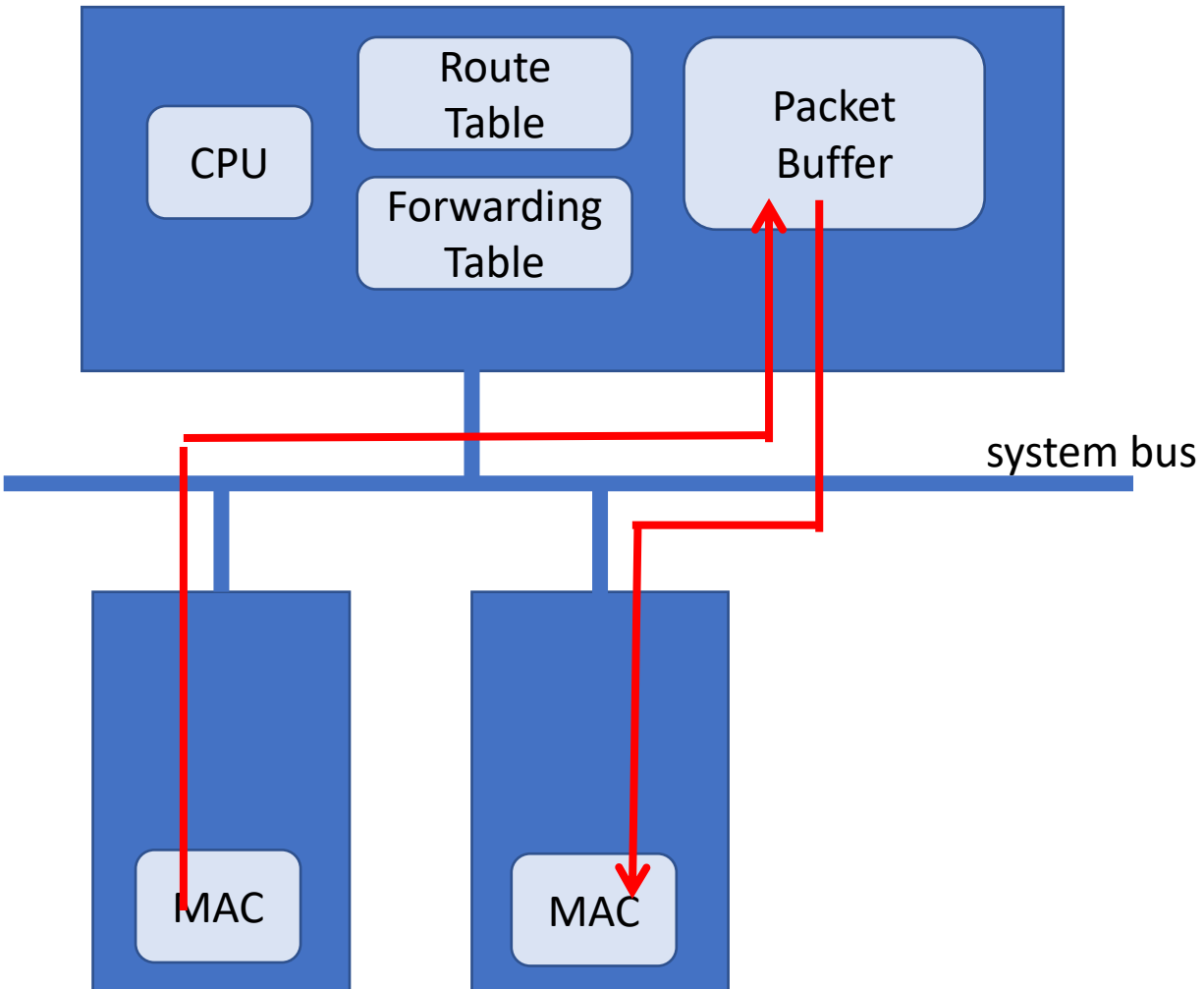
- Type of search tree
- Left branch = 0, Right = 1
- Walk the tree to perform a lookup

Processing

- Processing a packet
- Switch fabric

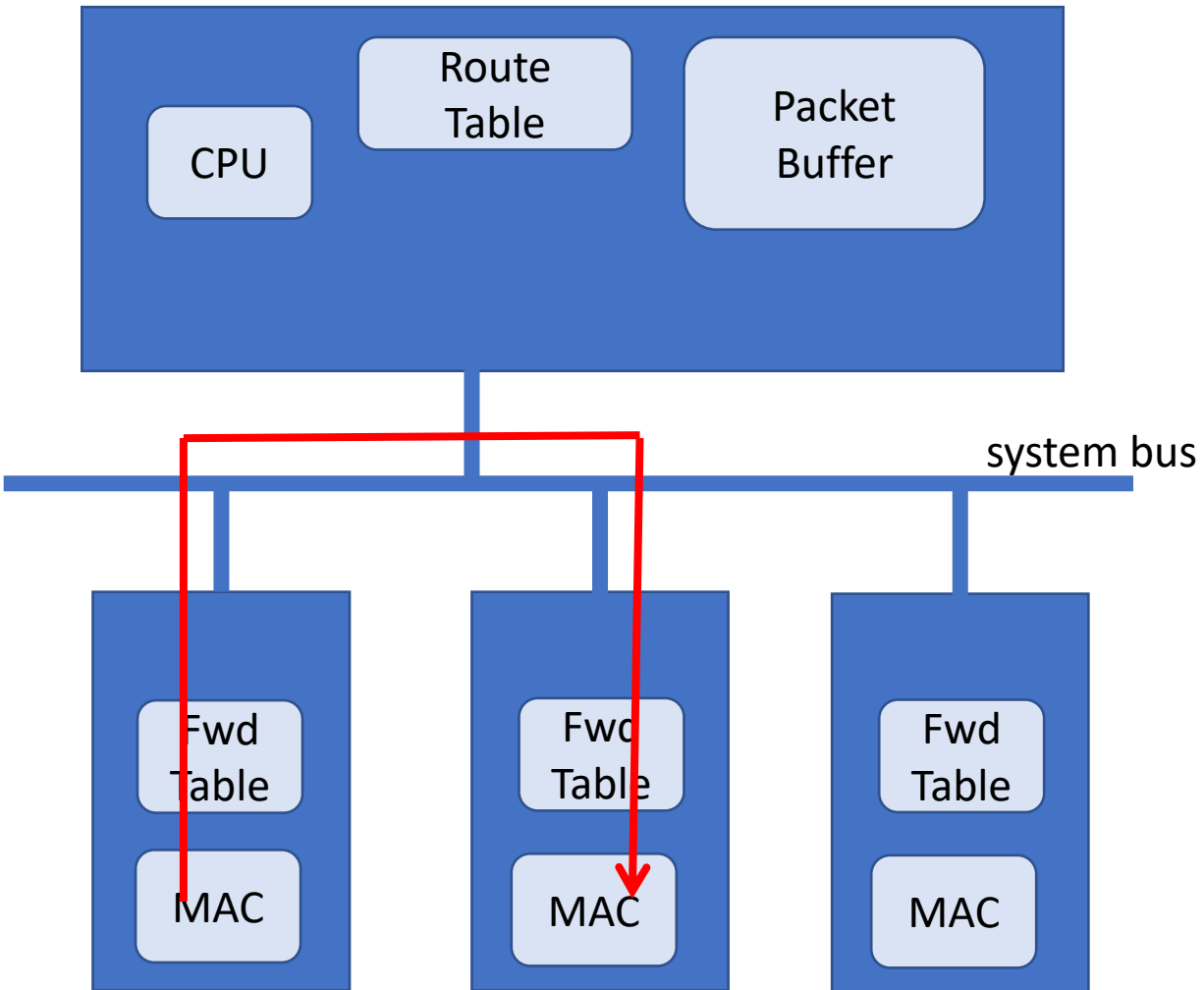


Generation 1: Switching via Memory



- Interface gets packet off the wire
- Transfers packet from interface to memory
- Processor determines next hop
- Transfer packet from memory to output interface
- Limitation: CPU processing

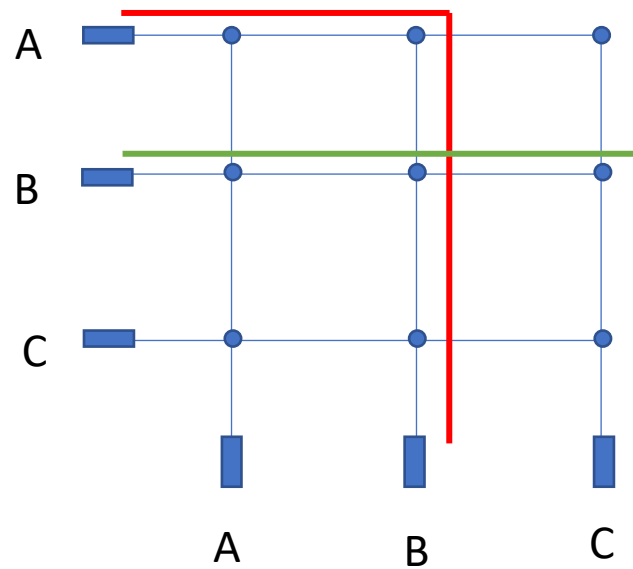
Generation 2: Switching via Bus



- Innovation: Each line card has forwarding table
- Transfer packet over the system bus
- Limitation: Can only transfer 1 packet at a time over bus

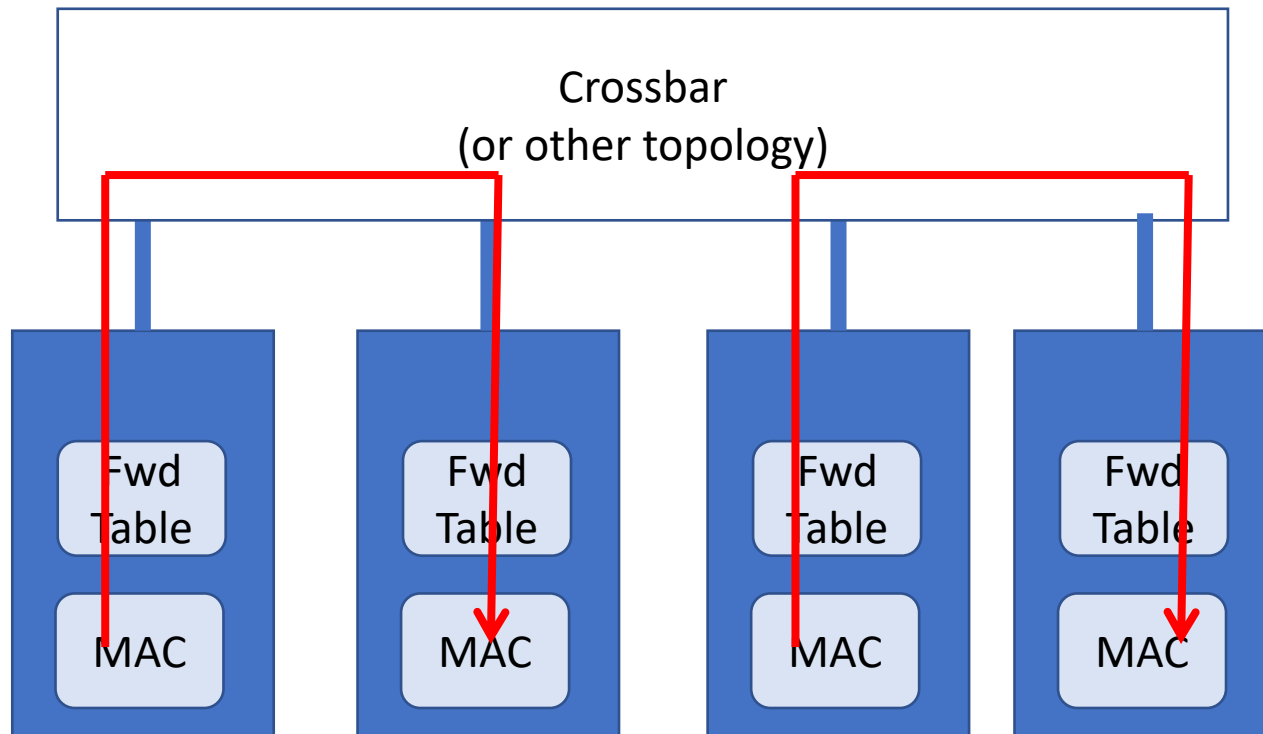
Generation 3: Switching Fabric

- Innovation: parallelism via a switching fabric (shown is a crossbar, but other topologies exist)
- During each timeslot, each input connected to 0 or 1 output

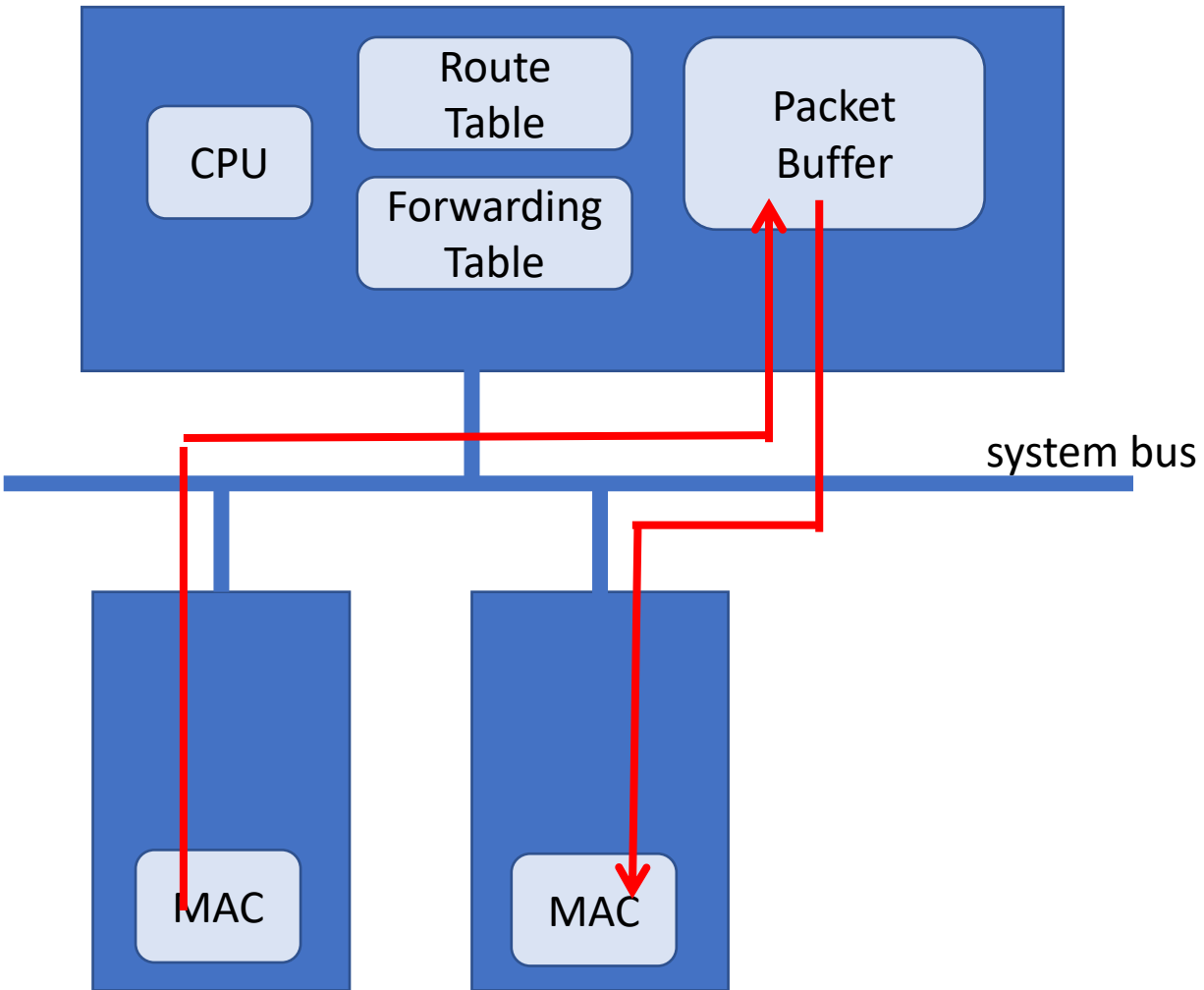


Generation 3: Switching Fabric

- Switching fabric can transfer many packets simultaneously
- Limitation: Requires special hardware



Generation 4: Switching via Memory



- Processors are getting faster
- Cloud / virtualized environments gaining popularity

Some additional things to look into

- Link scheduling to determine what to put onto the wire
(should it be round robin, weighted, priority)
- Dropping packets on congestion
(Drop-tail, Random Early Detection)
- Traffic shaping to force traffic to conform to a policy





University of Colorado **Boulder**

Routing

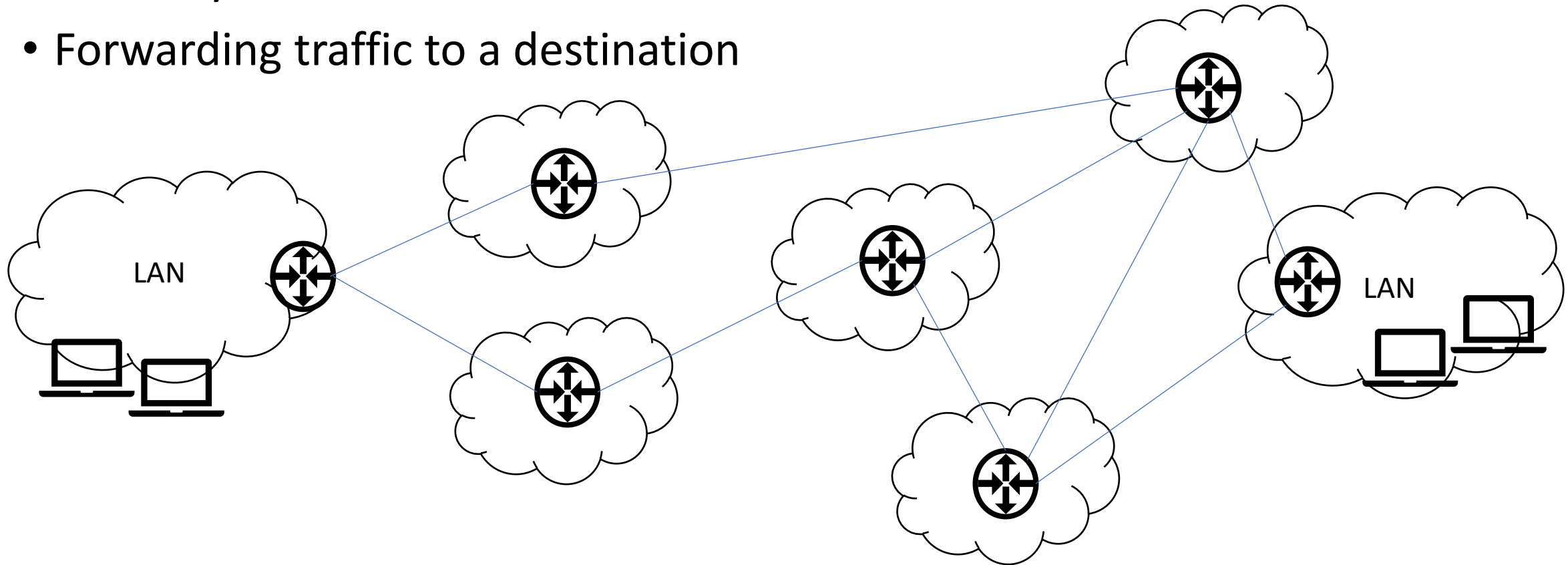
Course: Networking Fundamentals
Module: Network



University of Colorado **Boulder**

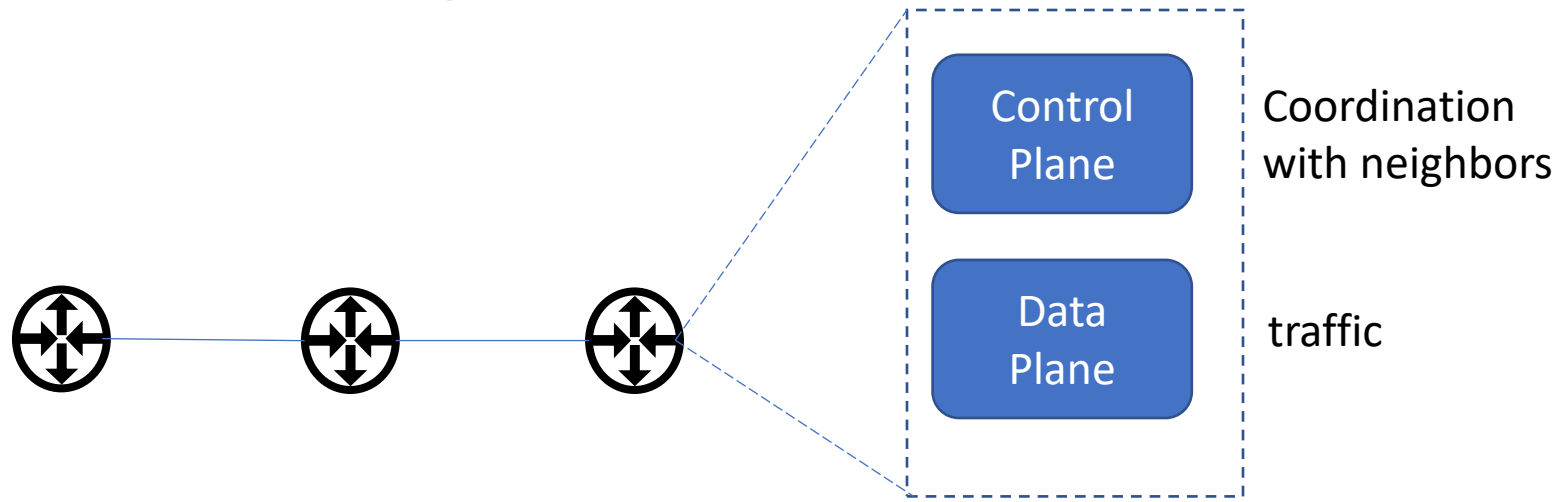
The Role of a Router

- Gateway
- Forwarding traffic to a destination



Forwarding vs Routing

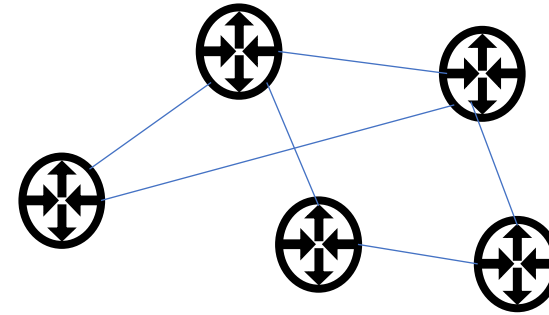
- Forwarding: Data plane
 - Direct a data packet to an output port/link
 - Uses a forwarding table
- Routing: Control plane
 - Computes paths by coordinating with neighbors
 - Creates the forwarding table



Key Function of Control Plane: Calculate the Forwarding Table

Distributed routing

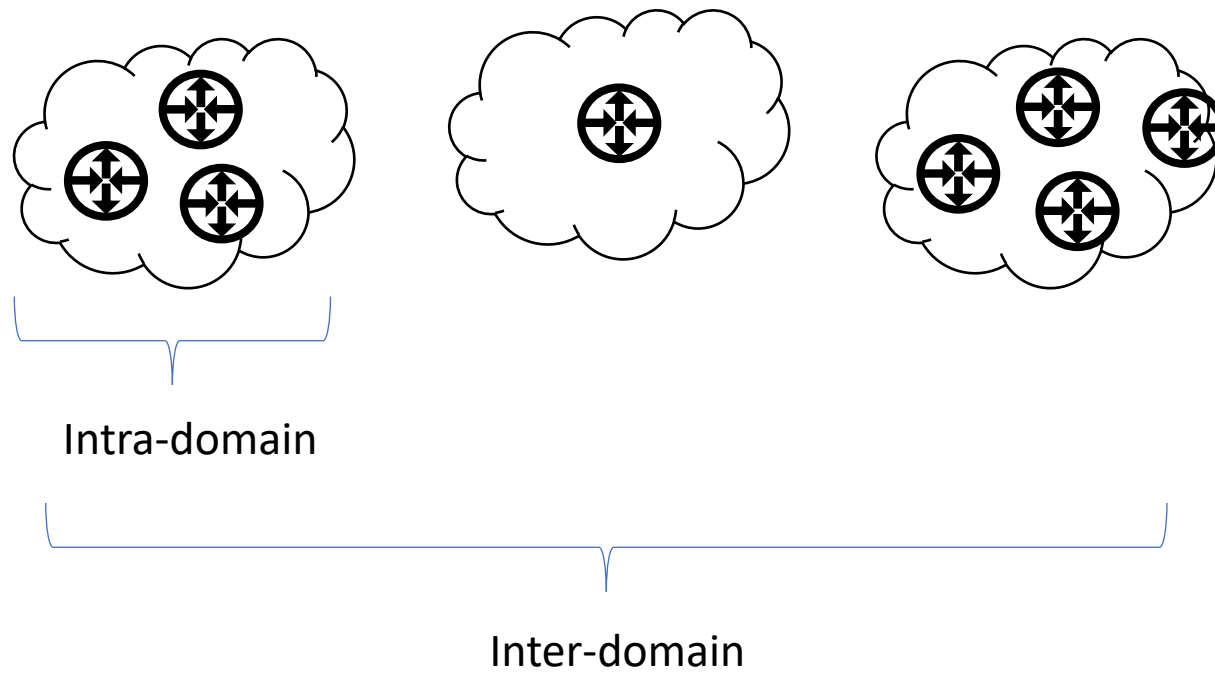
- What info each router knows
(entire topology or just its neighbors)
- What info each router exchanges
(all routes or single route)
- What calculation is performed
(shortest path or what's locally best)



Network as a graph

Decentralized Control Plane

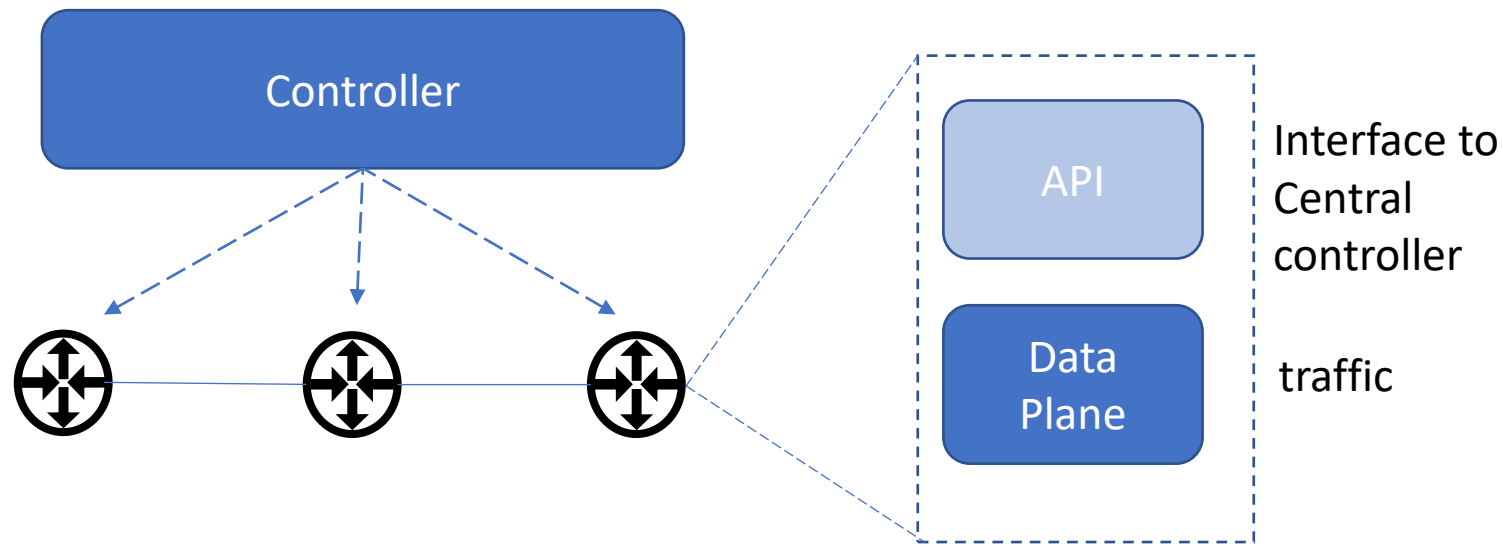
These are traditionally you'd know as routing protocols.



- Link-state protocol – OSPF (Intra-domain)
- Path-vector protocol – BGP (Inter-domain)

Centralized Control Plane (Software-defined Networking)

- Central controllers know entire topology, and makes decision for entire network
- Routers become just the data plane





University of Colorado **Boulder**

Routing: Link State Protocol

Course: Networking Fundamentals
Module: Network



University of Colorado **Boulder**

Decentralized Control Plane

These are traditionally you'd know as routing protocols.

- What info each router knows
(e.g., entire topology or just its neighbors)
- What info each router exchanges
(e.g., everything it has learned or just what it has selected)
- What calculation is performed
(e.g., shortest path or what's locally best)



OSPF – Example of a Link State Protocol

- Widely used in LANs
- OSPFv2 - <https://www.ietf.org/rfc/rfc2328.txt>
(244 pages)
- OSPFv3 (for IPv6) - <https://www.rfc-editor.org/rfc/rfc5340.html>
(94 pages)



OSPF Process Simplified

Each router establishes peering with neighbors

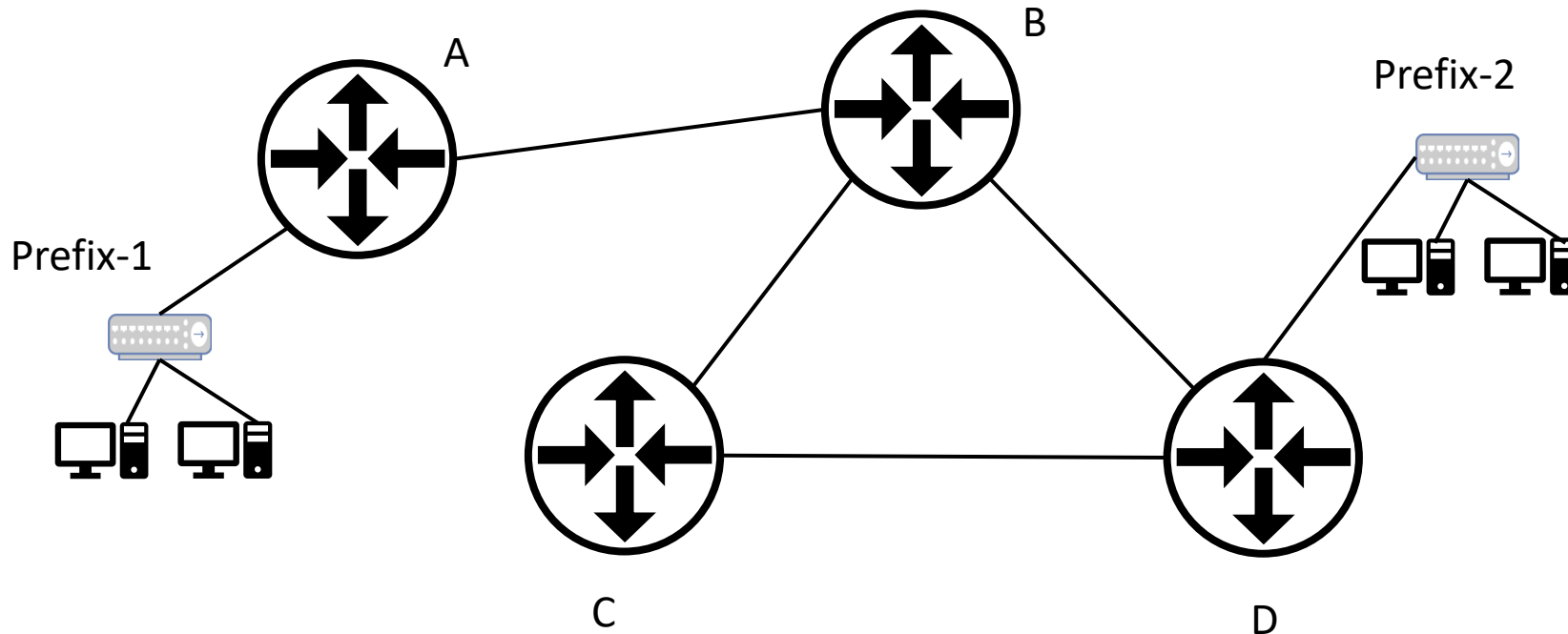
Link states are flooded to entire network

Each router calculates shortest path(s)



Link State

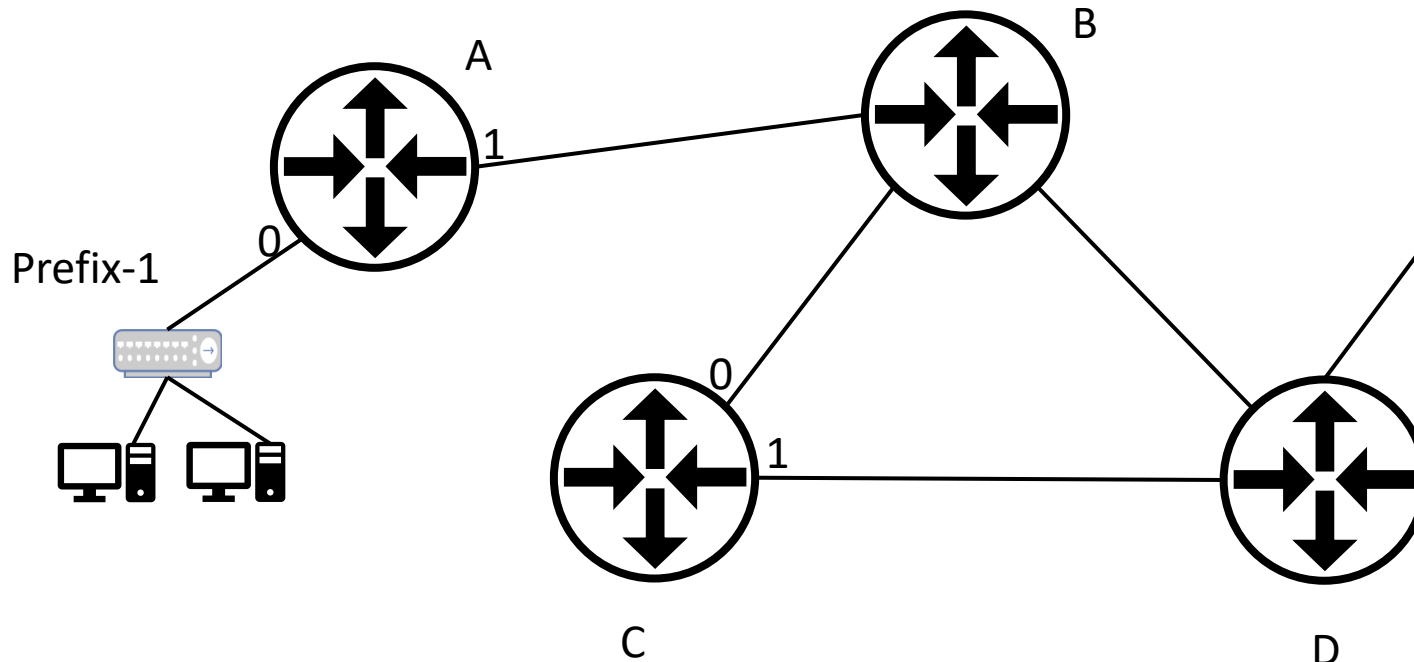
- Connection to other routers – tells who the neighbor is
- Connection to network with end hosts – tells prefixes reachable



Link State Examples

Link State for A

- Interface0 – A connected to Prefix-1
- Interface1 – A connected to B



Link State for C

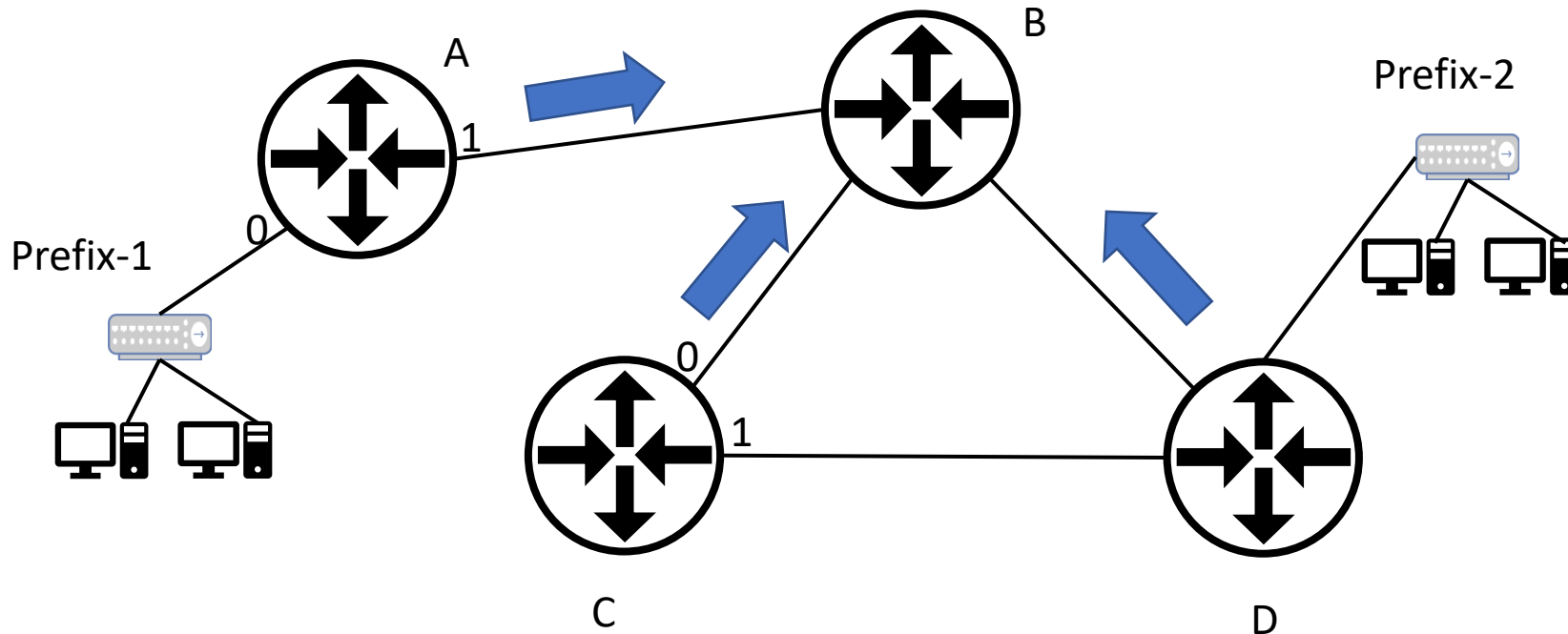
- Interface0 – C connected to B
- Interface1 – C connected to D

Link State Updates – Tell Neighbors

A will tell B that A is connected to Prefix-1

C will tell B That C is connected to D

D will tell B that D is connected to C and Prefix-2



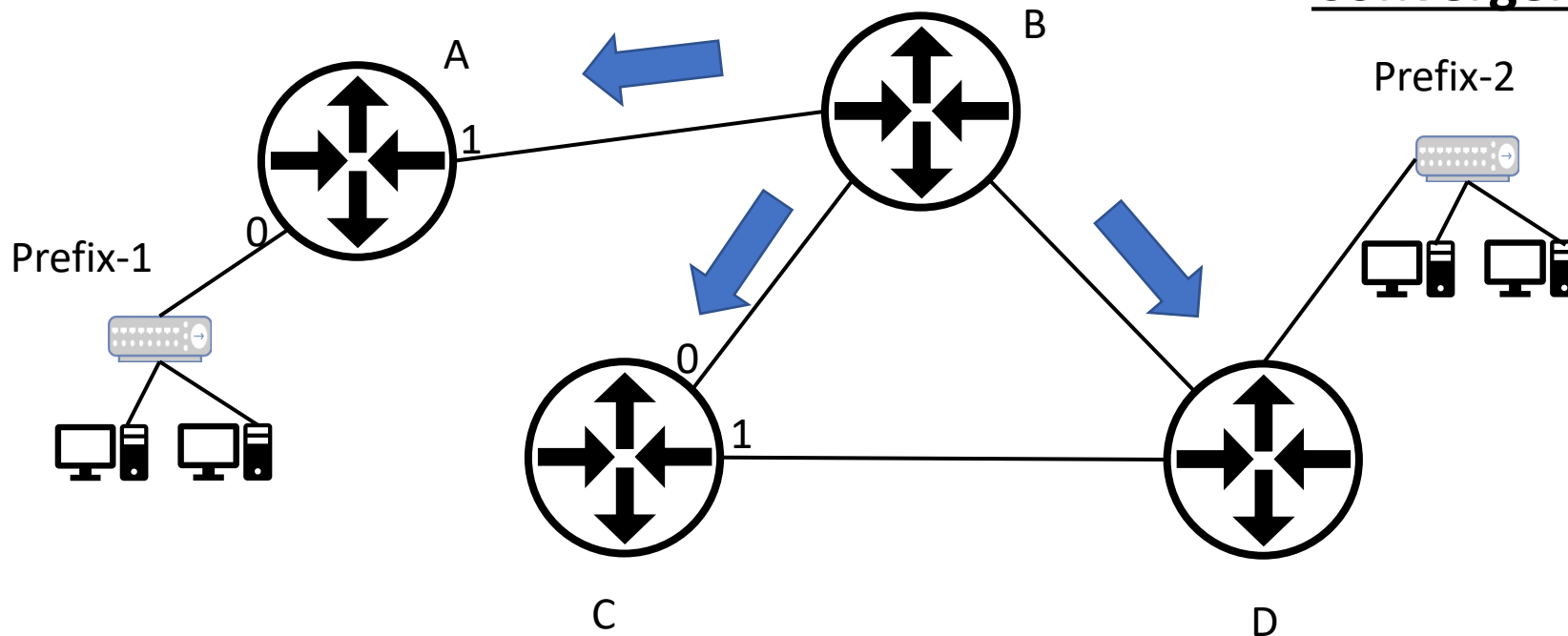
Link State Database

B's Link State Database now includes

- The link states for its interfaces
- The link states it has learned

B will now send a Link State Update to its neighbors

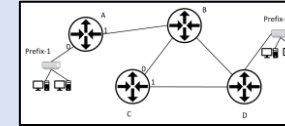
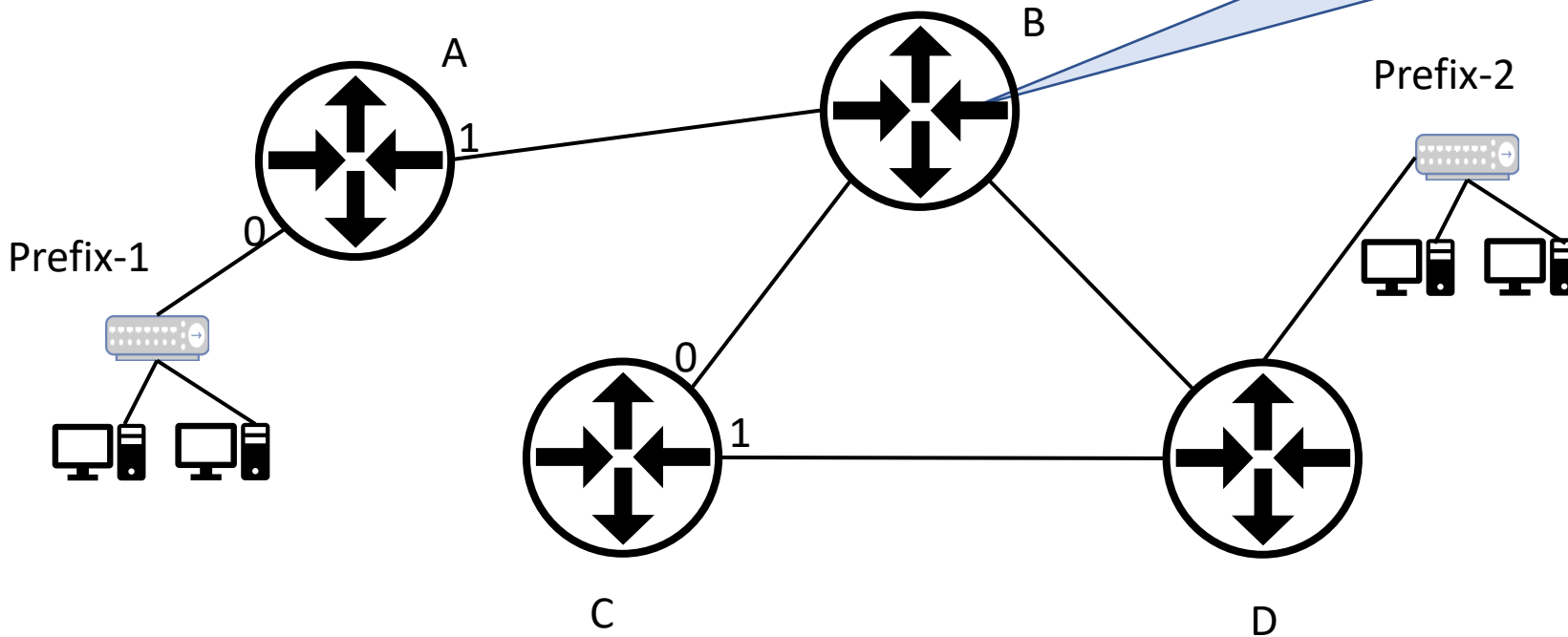
Process continues until network reaches
Convergence



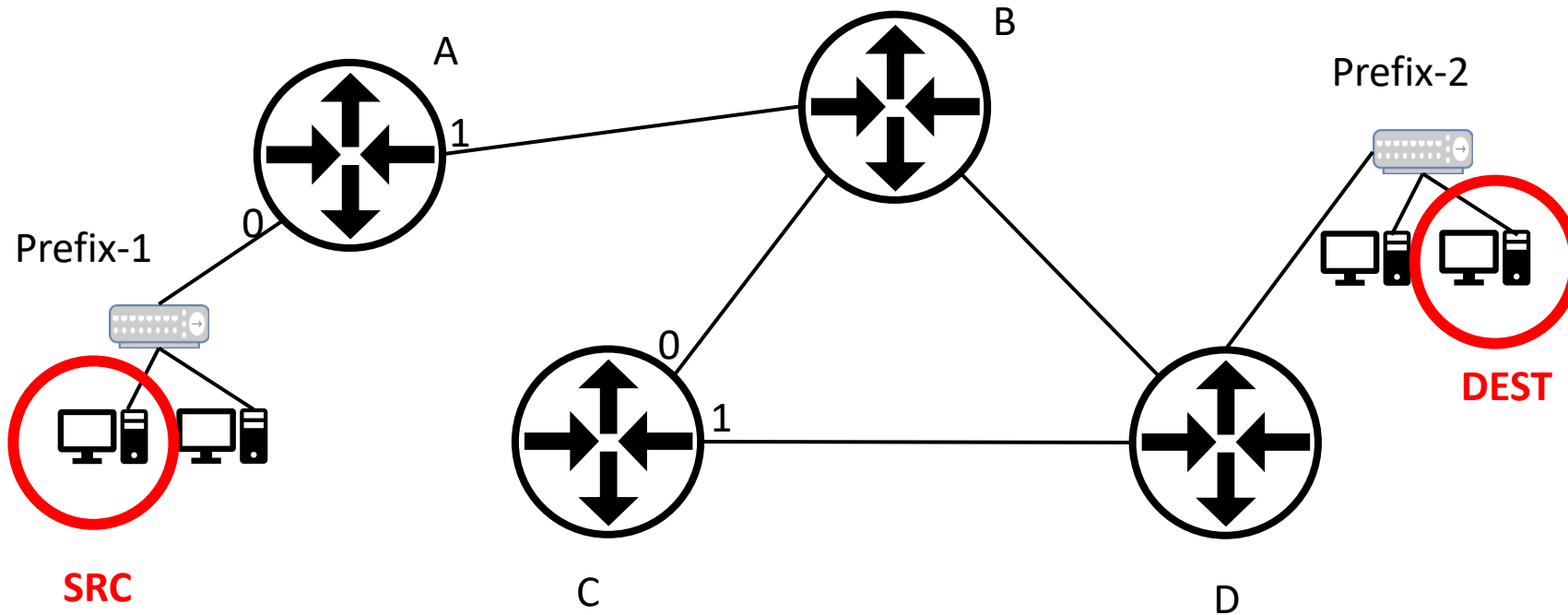
Converged State of Network

At the end B has learned the entire topology
(in fact, so did A, C, and D)

And they all agree

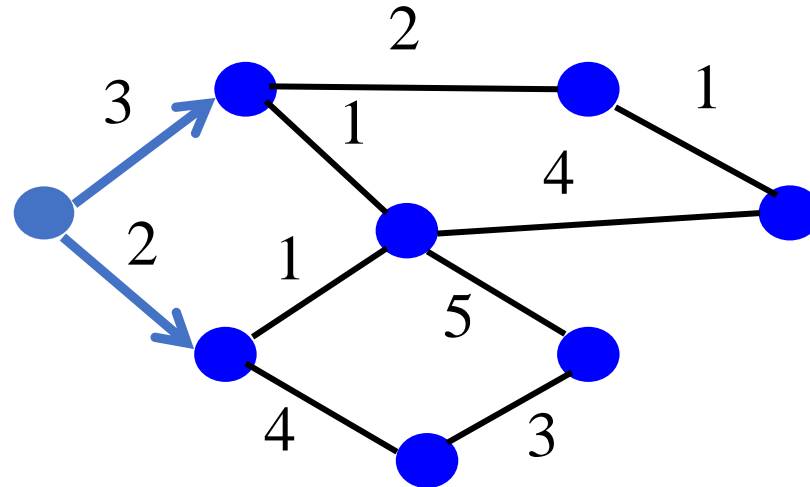


But, how does this help forward traffic?



Each Router Performs Shortest Path Calculation

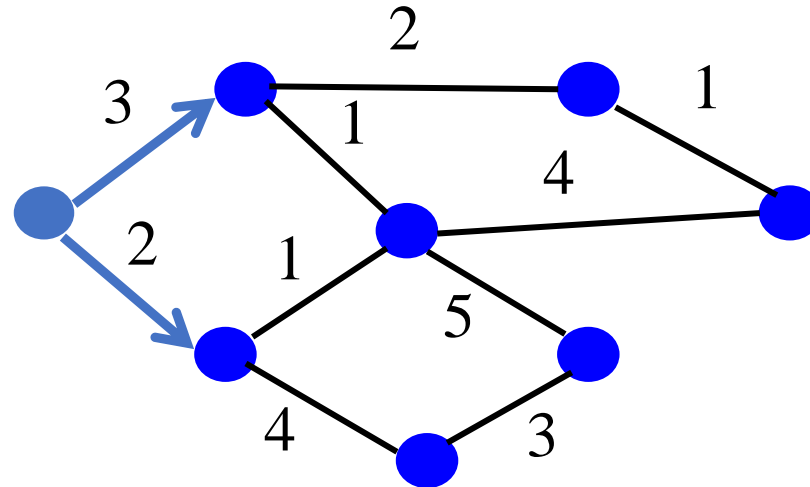
Dijkstra's Algorithm
(recall – a network is just represented as a graph)



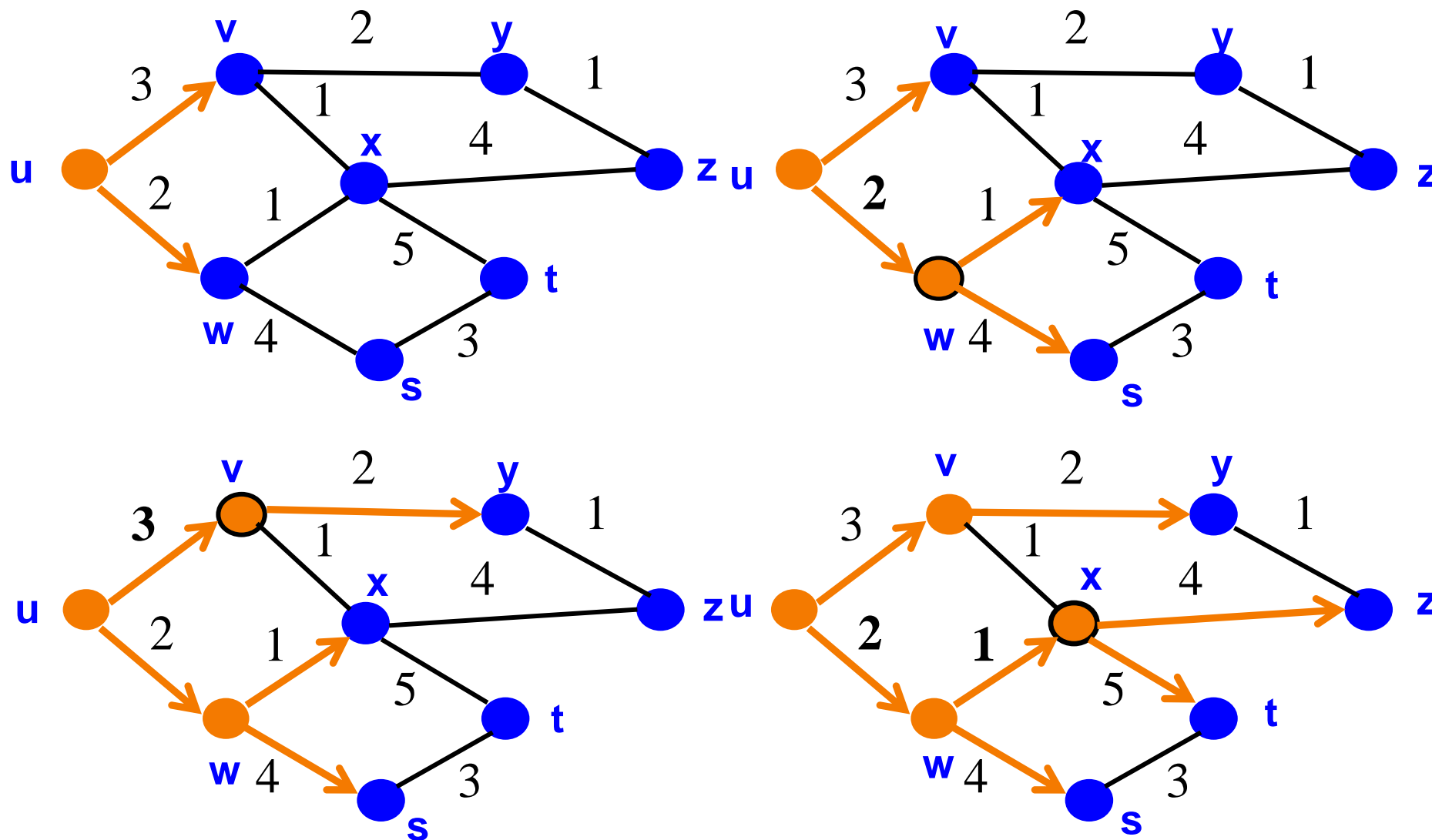
Each Router Performs Shortest Path Calculation

Dijkstra's Algorithm
(recall – a network is just represented as a graph)

Side note: OSPF links can have cost

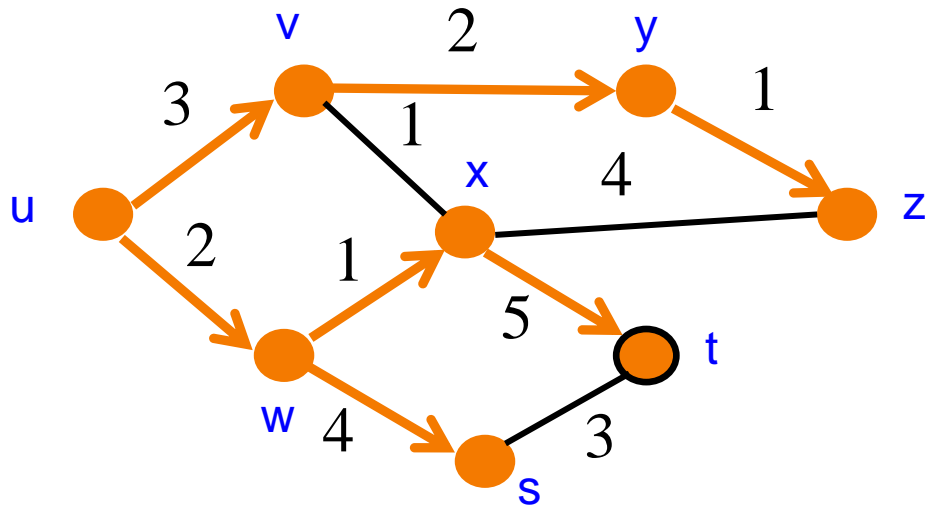


Example of Dijkstra's in Action



End Result of Shortest Path Calculation

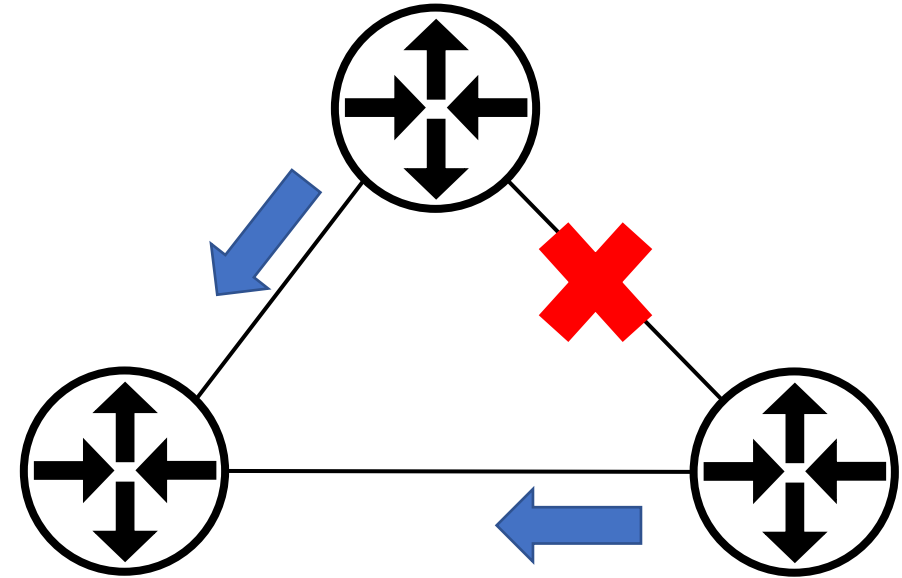
- Shortest-path tree from u
- Forwarding table at u



| | link |
|---|-------|
| v | (u,v) |
| w | (u,w) |
| x | (u,w) |
| y | (u,v) |
| z | (u,v) |
| s | (u,w) |
| t | (u,w) |

Changes in the Network

- New neighbor
 - Failure of a link or node
 - Adding a prefix
- ➔ All are changes in the Link State



Send Link State Updates (until convergence)

Calculate Shortest Path(s)

Update forwarding table



University of Colorado **Boulder**

OSPF (make as in video quiz)

- What info each router knows
- What info each router exchanges
- What calculation is performed

OSPF

- What info each router knows
 - ➔ All of the link states in the network.
- What info each router exchanges
- What calculation is performed

OSPF

- What info each router knows
 - ➔ All of the link states in the network (i.e., entire topology)
- What info each router exchanges
 - ➔ All of the link states the router knows about
- What calculation is performed

OSPF

- What info each router knows
 - ➔ All of the link states in the network (i.e., entire topology)
- What info each router exchanges
 - ➔ All of the link states the router knows about
- What calculation is performed
 - ➔ Shortest path



University of Colorado **Boulder**

Routing: Path Vector Protocol

Course: Networking Fundamentals
Module: Network



University of Colorado **Boulder**

Decentralized Control Plane

These are traditionally you'd know as routing protocols.

- What info each router knows
(e.g., entire topology or just its neighbors)
- What info each router exchanges
(e.g., everything it has learned or just what it has selected)
- What calculation is performed
(e.g., shortest path or what's locally best)



BGP

A Border Gateway Protocol 4 (BGP-4)

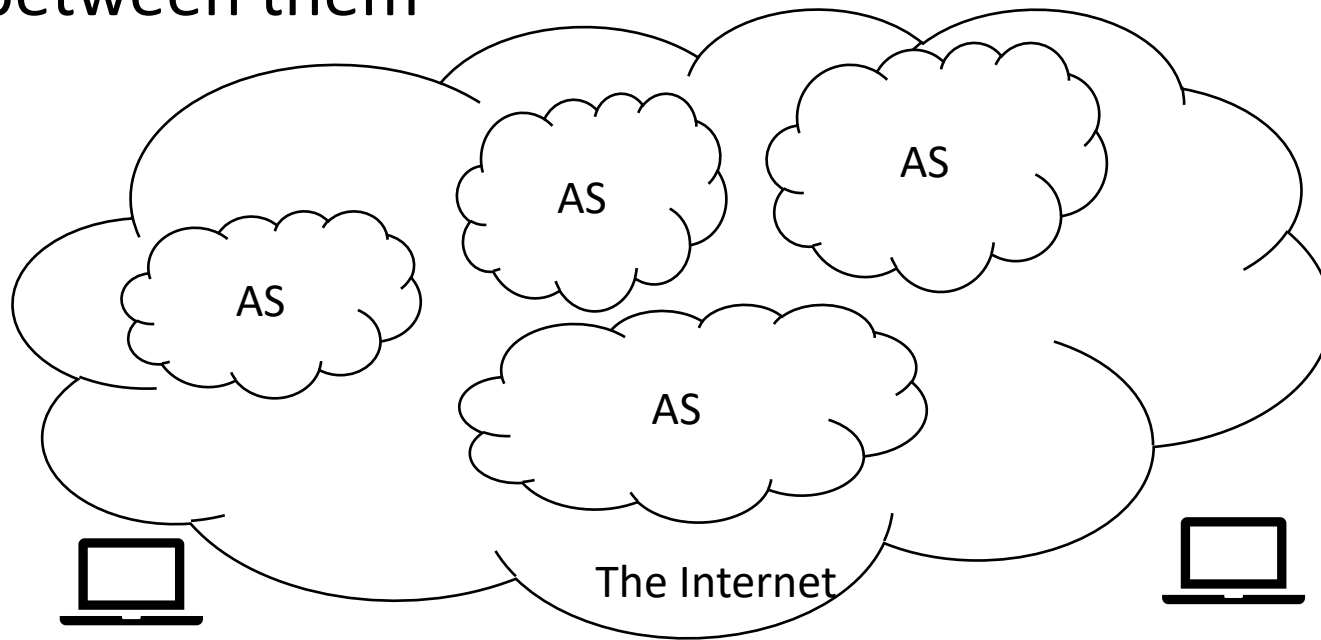
<https://datatracker.ietf.org/doc/html/rfc4271>



The Internet

Many inter-connected networks called Autonomous Systems

Border Gateway Protocol (BGP) is the protocol used to coordinate between them



BGP Process simplified

Peer with Neighbors

For any route update received or failure event or config change

- Update Routing Table (list of known routes)

- Perform locally best route calculation for that prefix

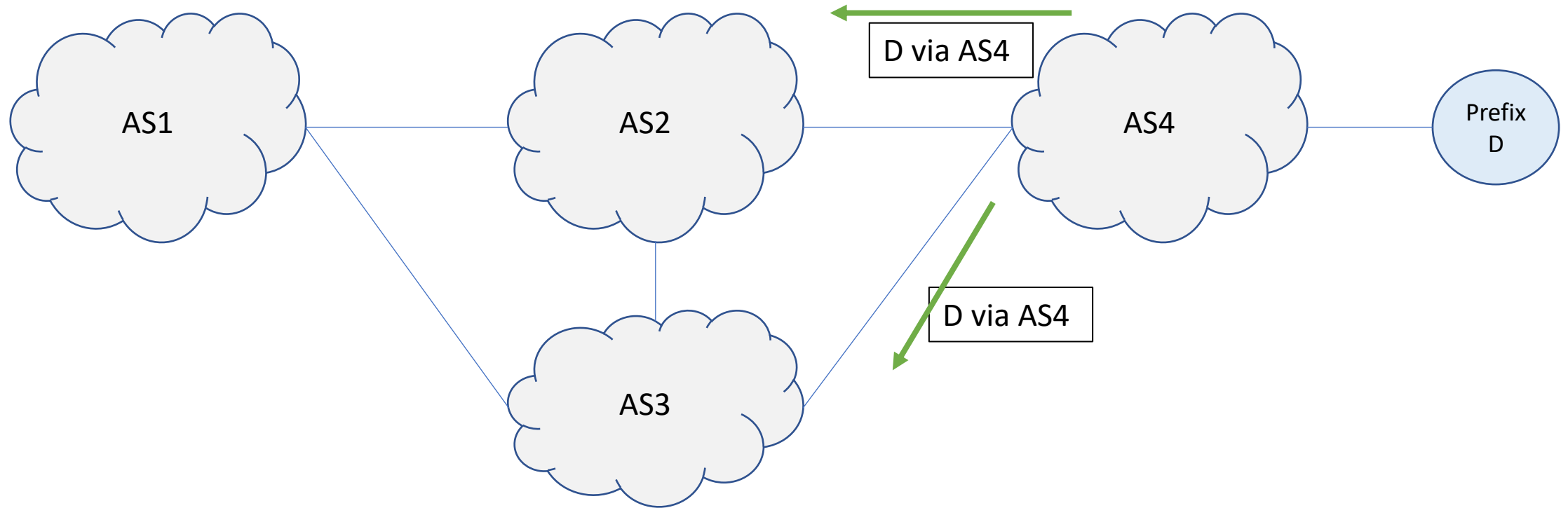
- If any change, send update to neighbors

Route = path to a prefix

Path = sequence of ASes

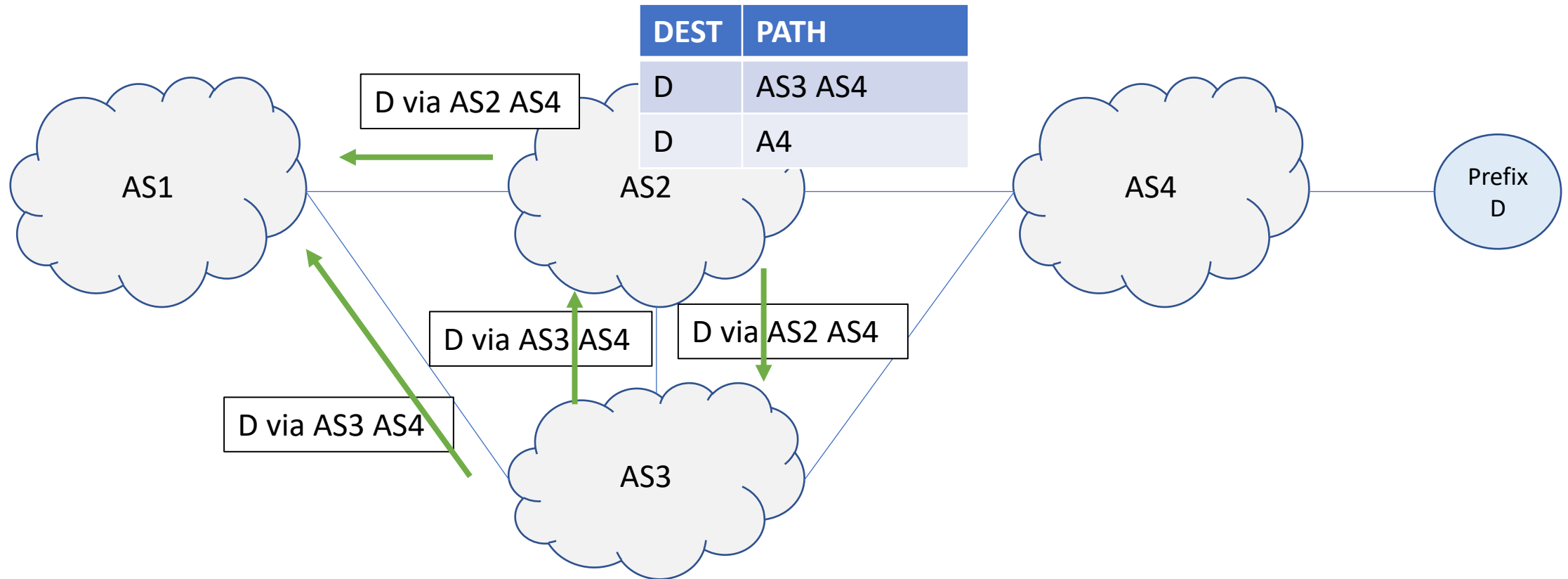


Example of route exchange



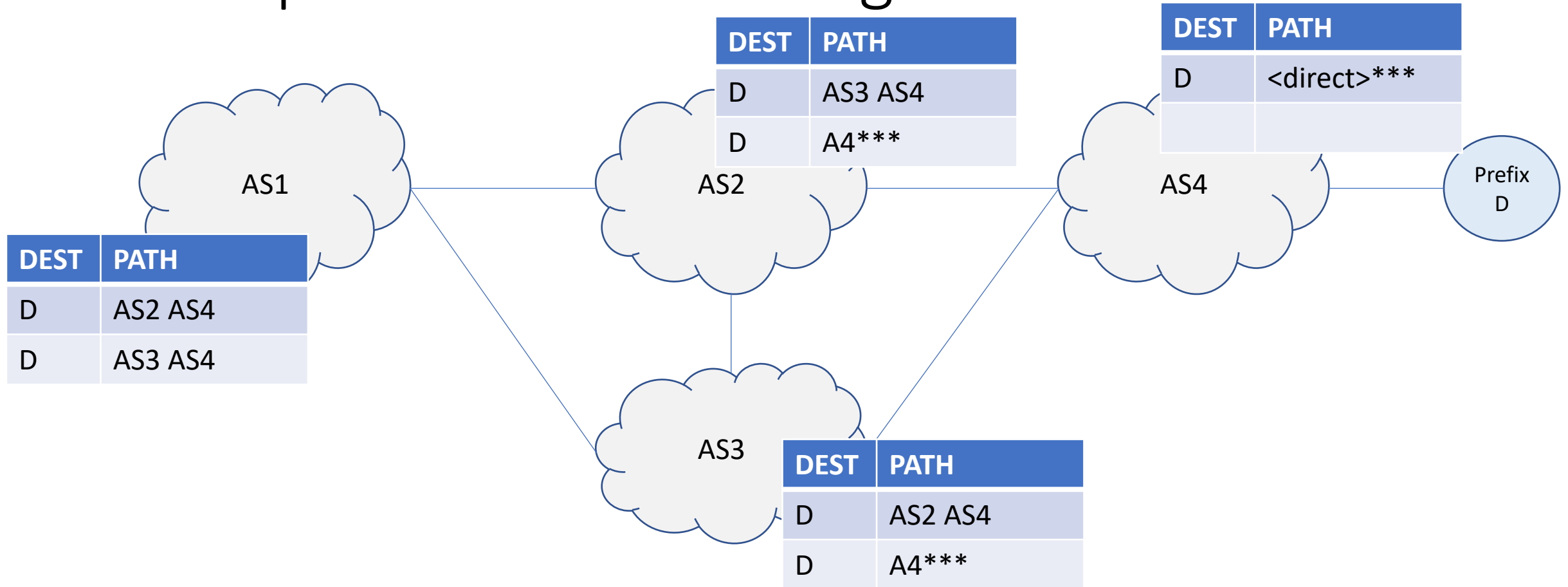
- AS4 announces to its neighbors (AS2 and AS3) that it can reach D with a path of AS4.
- AS2 and AS3 will insert that into the Routing Table and then choose best path to D

Example of route exchange



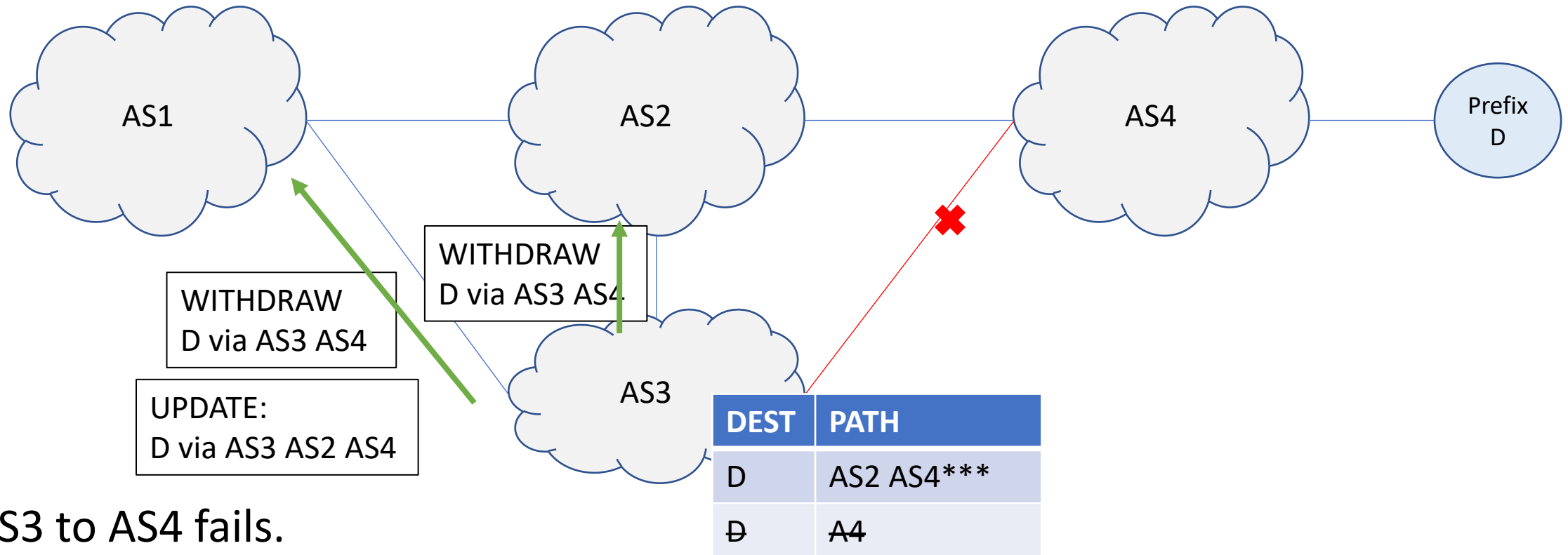
- AS2 and AS3 announce to their neighbors.
 - AS2 and AS3 now each know about 2 paths to D
 - e.g., AS2 knows (i) via AS4, (ii) via AS3 AS4. Will choose (i)

Example of route exchange



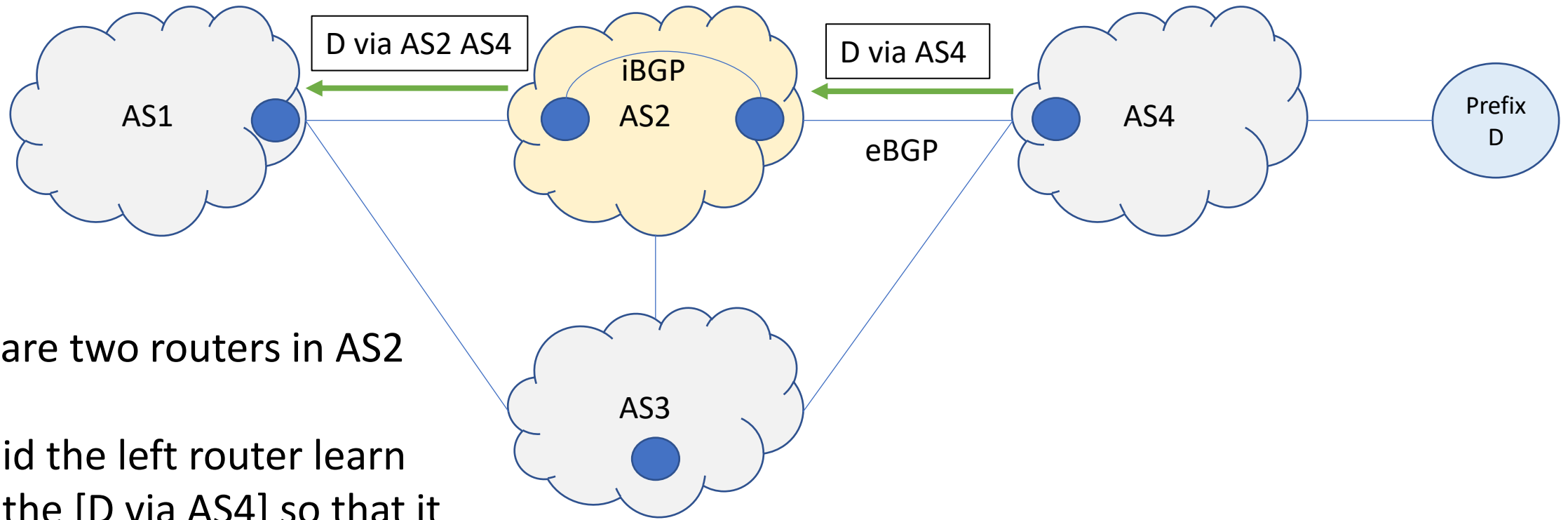
- AS1 knows 2 paths (i) via AS2 AS4, (ii) via AS3 AS4. Which will it choose?

Example of route exchange



- Say AS3 to AS4 fails.
- AS3 withdraws that route - tells AS1 and AS2
- AS3 has another path available (via AS2), so it chooses that, and announces to its neighbor AS1

iBGP vs eBGP

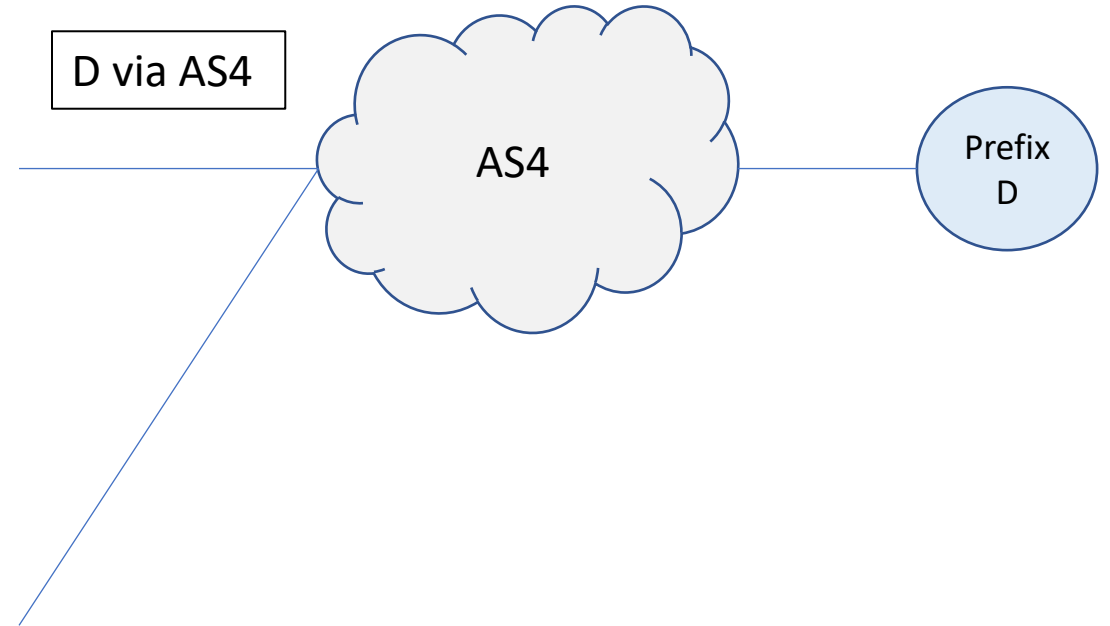


There are two routers in AS2

How did the left router learn about the [D via AS4] so that it can announce it to AS1?

Learning about Prefixes

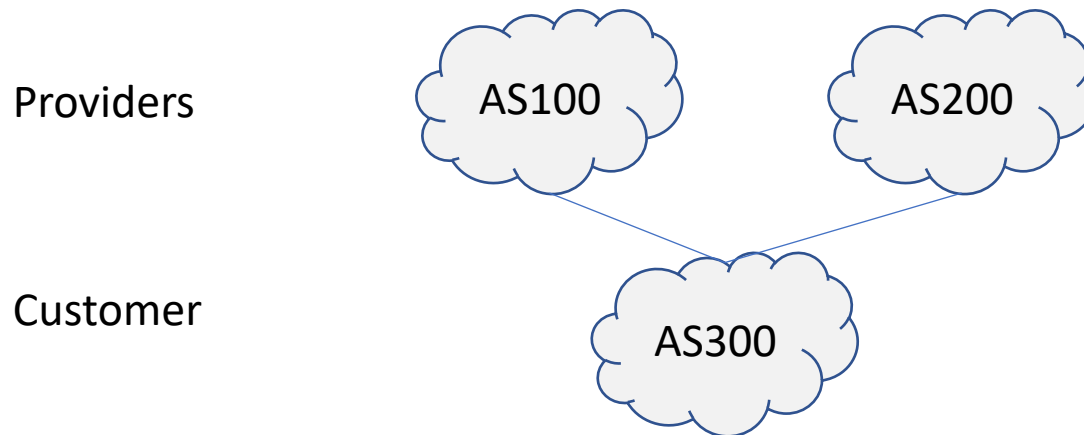
- How did AS4 learn about D?
 - Could be a local protocol (OSPF)
 - Or, static routes



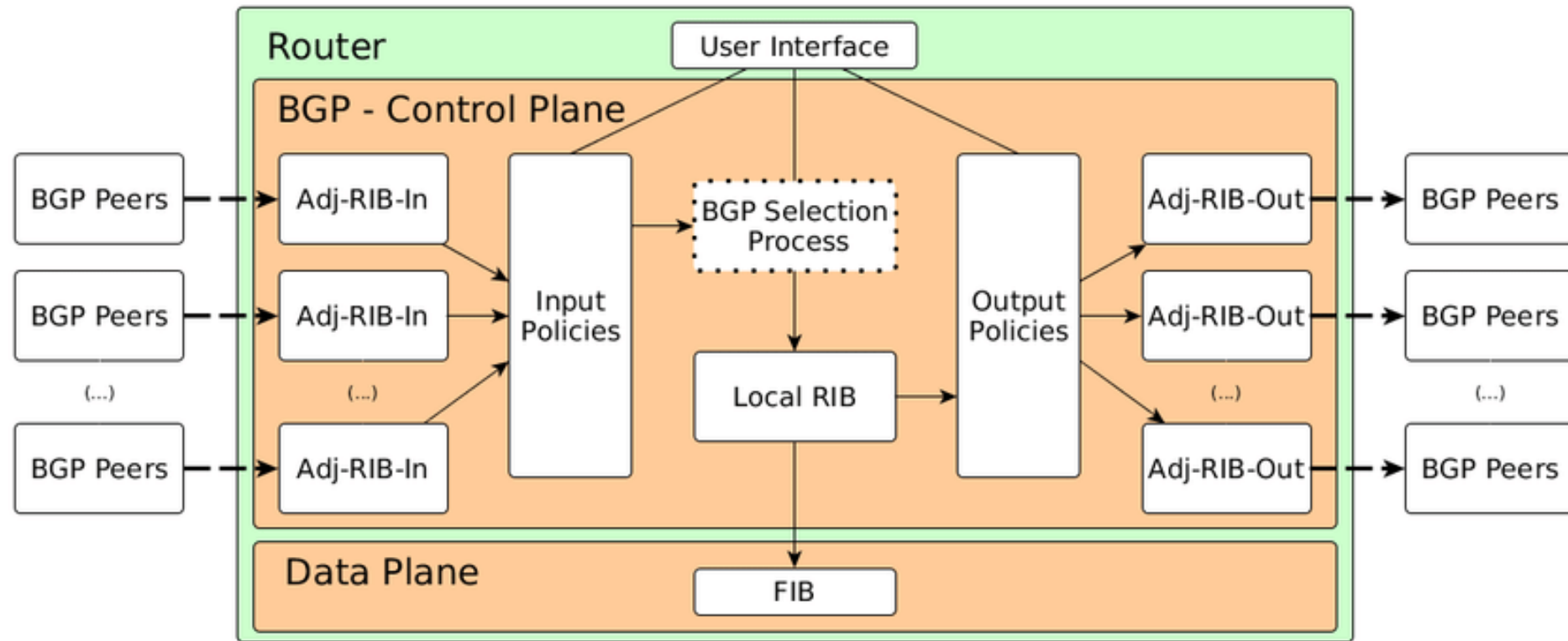
BGP Policies (and the Internet Business)

- Provider - is paid money to transit traffic
- Customer - pays money to a provider to have its traffic carried
- Peer - financially neutral arrangement where two ASes connect to each other and will send traffic through each other

BGP Policies to control what is being announced



Router's BGP Processing Pipeline



Best Path Calculation

| Local preference | numerical value assigned by routing policy. |
|-----------------------------------|--|
| AS path length | number of AS-level hops in the path |
| Multiple exit discriminator (MED) | allows one AS to specify that one exit point preferred |
| eBGP over iBGP | Learned through external neighbor over internal |
| Shortest IGP path cost | Exit this network as quickly as possible |
| Router ID | arbitrary tiebreaker |





University of Colorado **Boulder**

BGP (make as in video quiz)

- What info each router knows
- What info each router exchanges
- What calculation is performed

BGP (make as in video quiz)

- What info each router knows
 - ➔ Only its neighbors and routes it's received
- What info each router exchanges
 - ➔ Best route for each prefix (subject to filtering)
- What calculation is performed
 - ➔ Best local decision (incorporating business needs)



University of Colorado **Boulder**

Mapping IP Addresses

Course: Networking Fundamentals
Module: Network



University of Colorado **Boulder**

Network Layer Overview

- How do we address nodes scalably across different networks
- Given a destination address, how do we forward it (scalably)
- How do the different networks coordinate, so each can reach destinations beyond their boundaries
- **How are addresses assigned and mapped to the link layer address**
- What tools can an admin use to troubleshoot beyond its network boundaries



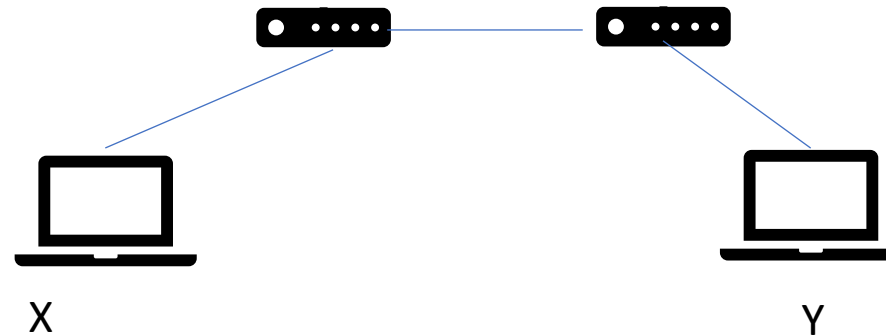
Allocation

- IANA – Internet Assigned Numbers Authority – global coordination
- Regional registries (RIR) manage local IP address allocation
<https://www.arin.net/resources/guide/request/>
- ISPs own large blocks and provide them to their customers



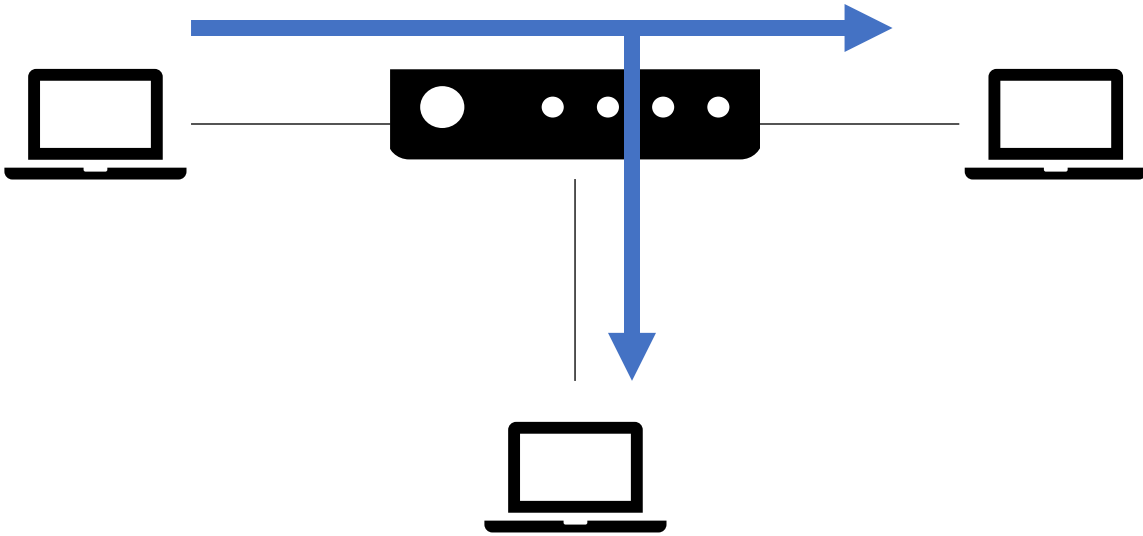
Mapping IP to MAC

- X wants to talk to Y on a LAN
- X more likely knows IP address of Y
 - Name of server (we'll cover DNS)
 - Consistent communication in larger network
- But, link layer comm is done with MAC addresses

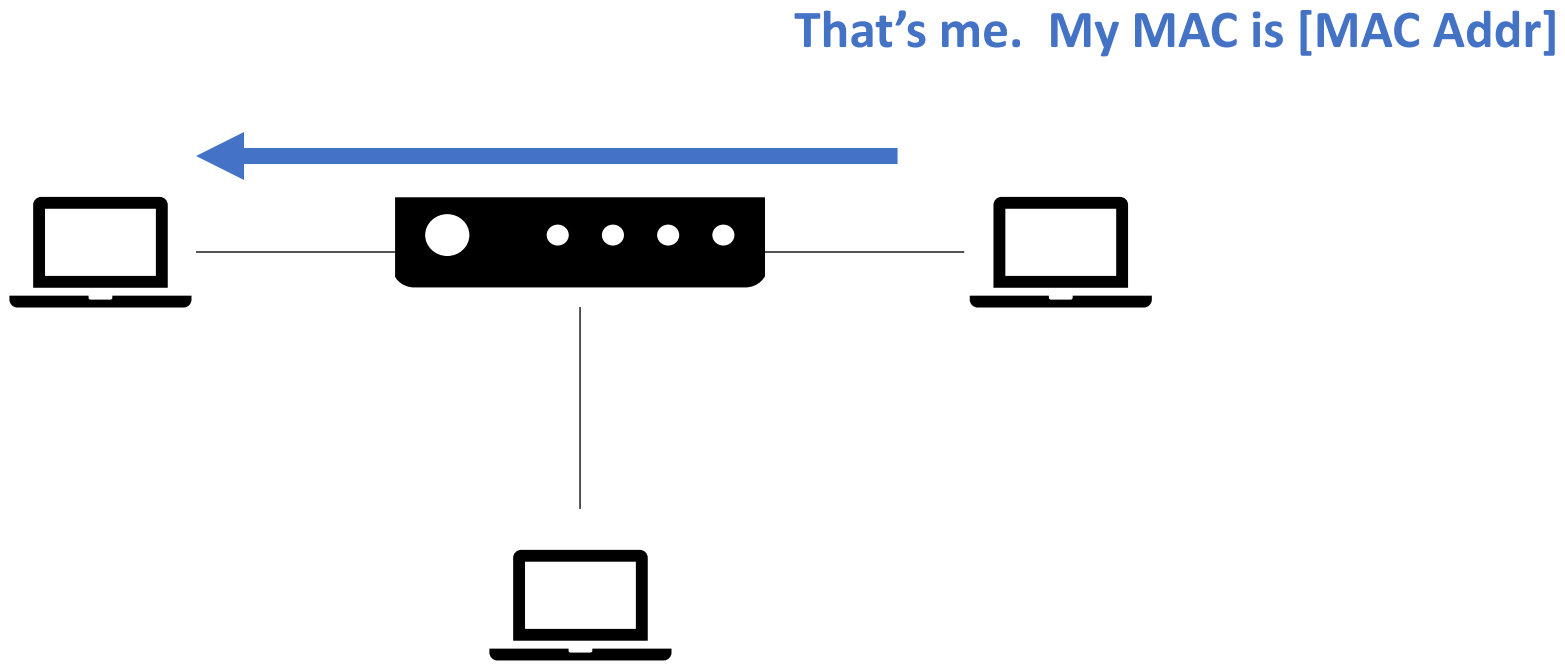


ARP – address resolution protocol

Broadcast: I'm looking for [IP address], what is your MAC?



ARP – address resolution protocol



201114_ARP_flags.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp.isprobe or arp.isgratuitous or arp.isannouncement

| No. | Time | Source | Destination | Protocol | Length | Is gratuitous | Is probe | Is announcement | Info |
|------|-------------|-------------------|-------------------|----------|--------|---------------|----------|-----------------|--------------------------------------|
| 3883 | 1354.931086 | 00:26:ab:58:f5:c5 | ff:ff:ff:ff:ff:ff | ARP | 64 | True | | True | ARP Announcement for 192.168.200.145 |
| 3937 | 1373.994258 | 00:05:cd:9b:f3:5c | ff:ff:ff:ff:ff:ff | ARP | 64 | True | | True | ARP Announcement for 192.168.200.204 |
| 3986 | 1390.279377 | 00:05:cd:3c:a6:90 | ff:ff:ff:ff:ff:ff | ARP | 60 | True | | True | ARP Announcement for 192.168.200.201 |
| 4064 | 1415.695237 | 00:26:ab:58:f5:c5 | ff:ff:ff:ff:ff:ff | ARP | 64 | True | | True | ARP Announcement for 192.168.200.145 |
| 4159 | 1462.902666 | 00:05:cd:3c:1b:dd | ff:ff:ff:ff:ff:ff | ARP | 60 | True | | True | ARP Announcement for 192.168.200.203 |
| 4174 | 1465.564231 | a0:a8:cd:68:29:73 | ff:ff:ff:ff:ff:ff | ARP | 60 | | True | | Who has 169.254.90.111? (ARP Probe) |
| 4175 | 1465.564585 | a0:a8:cd:68:29:73 | ff:ff:ff:ff:ff:ff | ARP | 60 | | True | | Who has 192.168.200.200? (ARP Probe) |
| 4177 | 1466.564934 | a0:a8:cd:68:29:73 | ff:ff:ff:ff:ff:ff | ARP | 60 | | True | | Who has 169.254.90.111? (ARP Probe) |
| 4180 | 1467.564167 | a0:a8:cd:68:29:73 | ff:ff:ff:ff:ff:ff | ARP | 60 | | True | | Who has 169.254.90.111? (ARP Probe) |
| 4183 | 1468.573678 | a0:a8:cd:68:29:73 | ff:ff:ff:ff:ff:ff | ARP | 60 | True | | True | ARP Announcement for 169.254.90.111 |
| 4210 | 1476.673236 | 00:26:ab:58:f5:c5 | ff:ff:ff:ff:ff:ff | ARP | 64 | True | | True | ARP Announcement for 192.168.200.145 |
| 4344 | 1537.660574 | 00:26:ab:58:f5:c5 | ff:ff:ff:ff:ff:ff | ARP | 64 | True | | True | ARP Announcement for 192.168.200.145 |
| 4495 | 1598.638727 | 00:26:ab:58:f5:c5 | ff:ff:ff:ff:ff:ff | ARP | 64 | True | | True | ARP Announcement for 192.168.200.145 |

< >

> Frame 4175: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
 > Ethernet II, Src: a0:a8:cd:68:29:73, Dst: ff:ff:ff:ff:ff:ff
 v Address Resolution Protocol (ARP Probe)
 Hardware type: Ethernet (1)
 Protocol type: IPv4 (0x0800)
 Hardware size: 6
 Protocol size: 4
 Opcode: request (1)
 [Is probe: True]
 Sender MAC address: a0:a8:cd:68:29:73
 Sender IP address: 0.0.0.0
 Target MAC address: 00:00:00:00:00:00
 Target IP address: 192.168.200.200

```

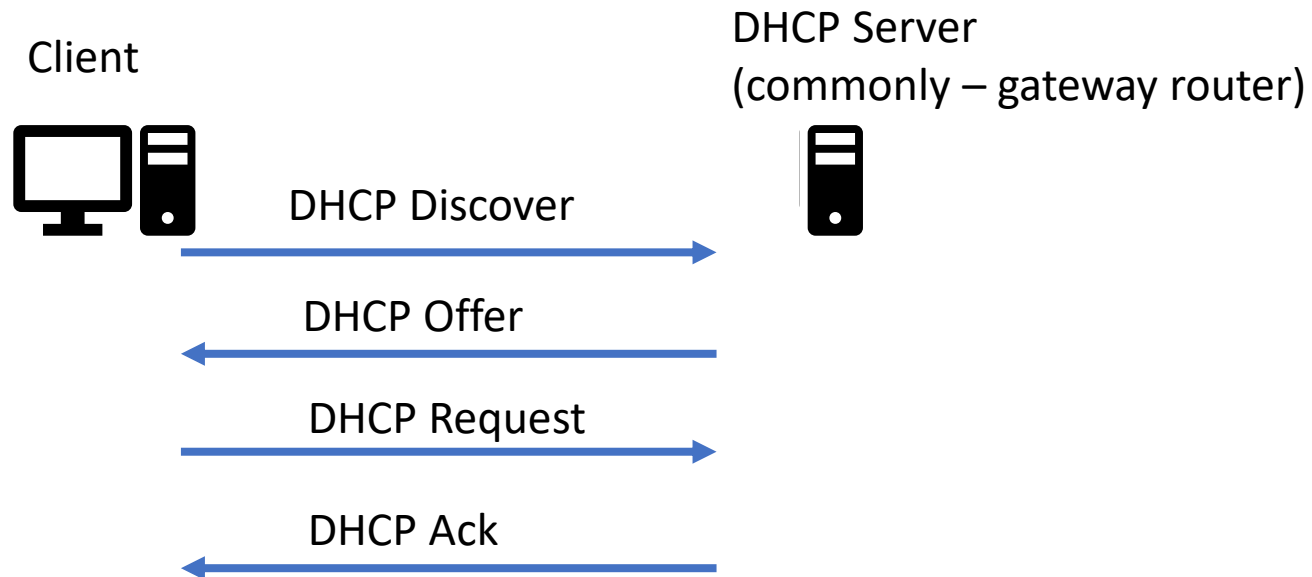
0000 ff ff ff ff ff ff a0 a8 cd 68 29 73 08 06 00 01 ..... -h)s...
0010 08 00 06 04 00 01 a0 a8 cd 68 29 73 00 00 00 00 ..... -h)s...
0020 00 00 00 00 00 00 c0 a8 c8 c8 00 00 00 00 00 .....
0030 00 00 00 00 00 00 00 00 00 00 00 00 .....
  
```

Is probe (arp.isprobe)

Packets: 4562 · Displayed: 47 (1.0%) · Dropped: 0 (0.0%) · Load time: 0:0.33 | Profile: Default

Assigning an IP on a LAN

- Statically assign
`sudo ip addr add 10.1.1.10/24 dev eth0`
- Dynamic – DHCP (dynamic host configuration protocol)





University of Colorado **Boulder**

Tools for Troubleshooting

Course: Networking Fundamentals
Module: Network



University of Colorado **Boulder**

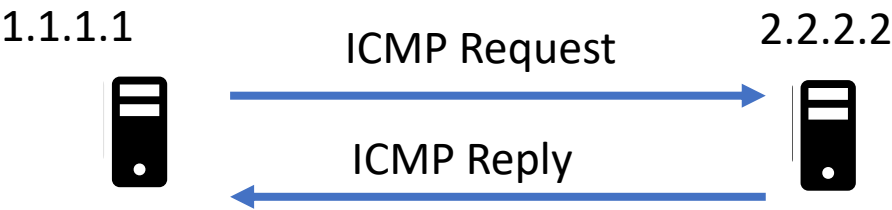
ping

- Quick check if a server is up
- Also includes round trip time, so can identify latency issues

```
node0:~> ping colorado.edu
PING colorado.edu (151.101.130.133) 56(84) bytes of data.
64 bytes from 151.101.130.133 (151.101.130.133): icmp_seq=1 ttl=56 time=9.21 ms
64 bytes from 151.101.130.133 (151.101.130.133): icmp_seq=2 ttl=56 time=7.13 ms
64 bytes from 151.101.130.133 (151.101.130.133): icmp_seq=3 ttl=56 time=7.49 ms
```



Ping uses ICMP Echo



Protocol = 1 (ICMP)

| IPv4 header format | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------|-------|------------------------|---|---|---|-----|---|---|---|----------|---|----|----|----|----|-----|----|-----------------|----|----|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|
| Offsets | Octet | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
| Octet | Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 0 | 0 | Version | | | | IHL | | | | DSCP | | | | | | ECN | | Total Length | | | | | | | | | | | | | | | |
| 4 | 32 | Identification | | | | | | | | | | | | | | | | Flags | | | Fragment Offset | | | | | | | | | | | | |
| 8 | 64 | Time To Live | | | | | | | | Protocol | | | | | | | | Header Checksum | | | | | | | | | | | | | | | |
| 12 | 96 | Source IP Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16 | 128 | Destination IP Address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 20 | 160 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Type =

| ICMP header format | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------|-------|----------------|---|---|---|---|---|---|---|------|---|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Offsets | Octet | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
| Octet | Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 0 | 0 | Type | | | | | | | | Code | | | | | | | | Checksum | | | | | | | | | | | | | | | |
| 4 | 32 | Rest of header | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

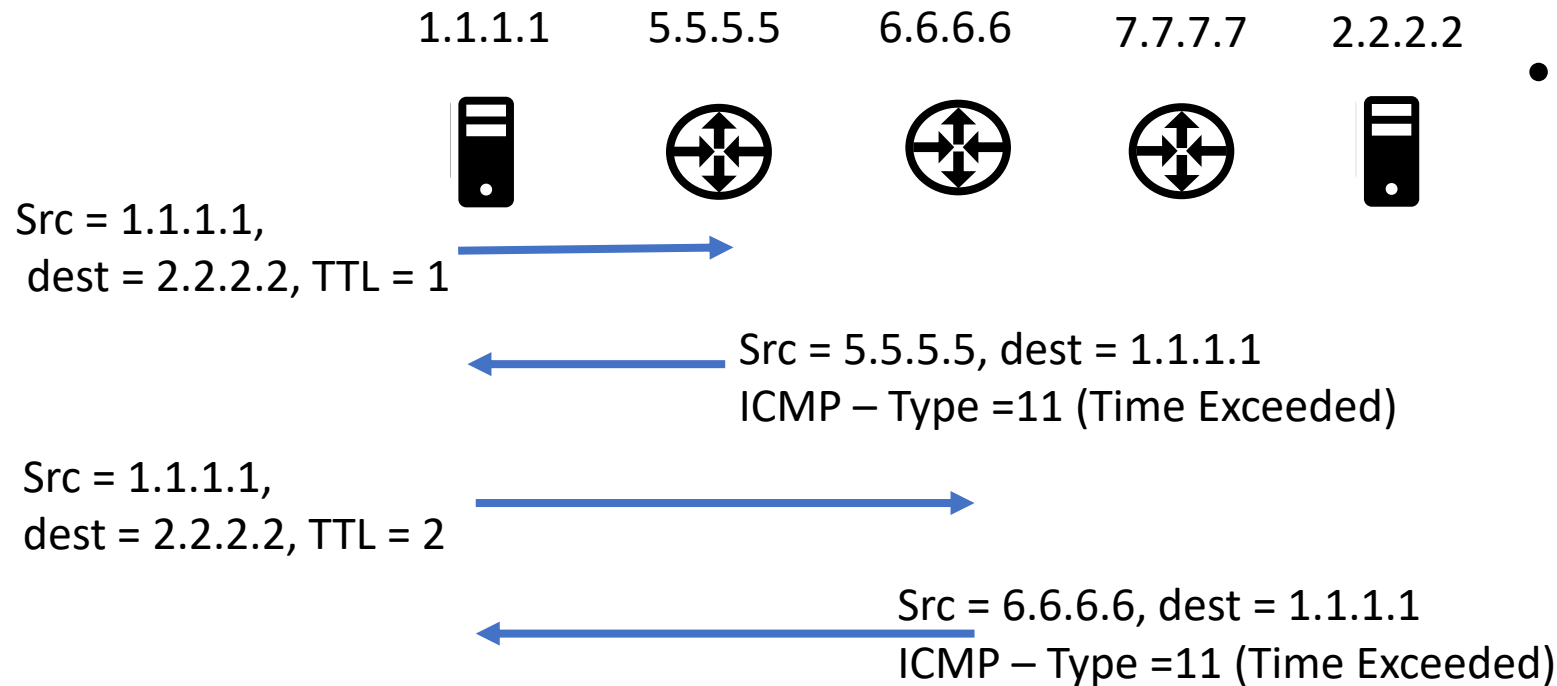
Type = 0 (Request)
Type = 1 (Reply)

Traceroute – path to destination

From a server hosted in cloudfab in Wisconsin

```
node0:~> traceroute nsf.gov
traceroute to nsf.gov (128.150.221.106), 30 hops max, 60 byte packets
 1  128.104.222.2 (128.104.222.2)  0.457 ms  0.638 ms  0.615 ms
 2  146.151.164.179 (146.151.164.179)  0.580 ms  0.556 ms  0.532 ms
 3  146.151.164.18 (146.151.164.18)  0.499 ms  0.728 ms  0.451 ms
 4  r-uwmadison-animal-ae5-187.uwsys.net (143.235.41.130)  0.597 ms  0.573 ms  0.549 ms
 5  r-uwmadison-cssc-ae3-3439.uwsys.net (143.235.33.94)  13.696 ms  13.672 ms  13.649 ms
 6  12.90.18.37 (12.90.18.37)  13.854 ms  13.775 ms  13.586 ms
 7  * * *
 8  * * *
 9  * * *
10  32.130.24.47 (32.130.24.47)  22.686 ms  22.690 ms *
11  4.68.39.1 (4.68.39.1)  25.900 ms  27.208 ms  27.142 ms
12  ae2.3609.edge2.Dallas2.level3.net (4.69.206.169)  27.083 ms *  27.130 ms
13  Lumen-level3-Dallas.Level3.net (4.68.110.70)  27.205 ms  27.046 ms  27.247 ms
14  dcx2-edge-01.inet.qwest.net (67.14.28.70)  46.264 ms  46.623 ms  45.394 ms
15  stn-mtps-10.inet.qwest.net (205.171.251.62)  47.111 ms  45.887 ms  47.041 ms
16  65.126.14.94 (65.126.14.94)  45.703 ms  45.858 ms  45.731 ms
17  dca-priv-14.inet.qwest.net (65.116.153.241)  46.222 ms  46.183 ms  45.941 ms
18  stn-priv-20.inet.qwest.net (65.126.14.89)  46.066 ms  47.637 ms  47.268 ms
19  63-146-9-178.dia.static.qwest.net (63.146.9.178)  47.490 ms  46.536 ms  46.241 ms
20  * * *
21  * * *
22  * * *
```

Traceroute – how its implemented



- Set TTL = desired hop
- When TTL expires, router will send an ICMP message that the Time Exceeded

iperf3

- Determine the max throughput between a client and server
- Can be used to identify performance issues

1.1.1.1



iperf3 -c 2.2.2.2

2.2.2.2



iperf3 -s

```
[ 5] local 1.1.1.1 port 34938 connected to 2.2.2.2 port 5002
[ ID] Interval            Transfer          Bitrate          Retr   Cwnd
[ 5]  0.00-1.00    sec   4.60 MBytes    38.6 Mbits/sec      0   1.51 MBytes
[ 5]  1.00-2.00    sec  11.2 MBytes    94.4 Mbits/sec      0   1.72 MBytes
```

...





University of Colorado **Boulder**