TDTS21: Advanced Networking

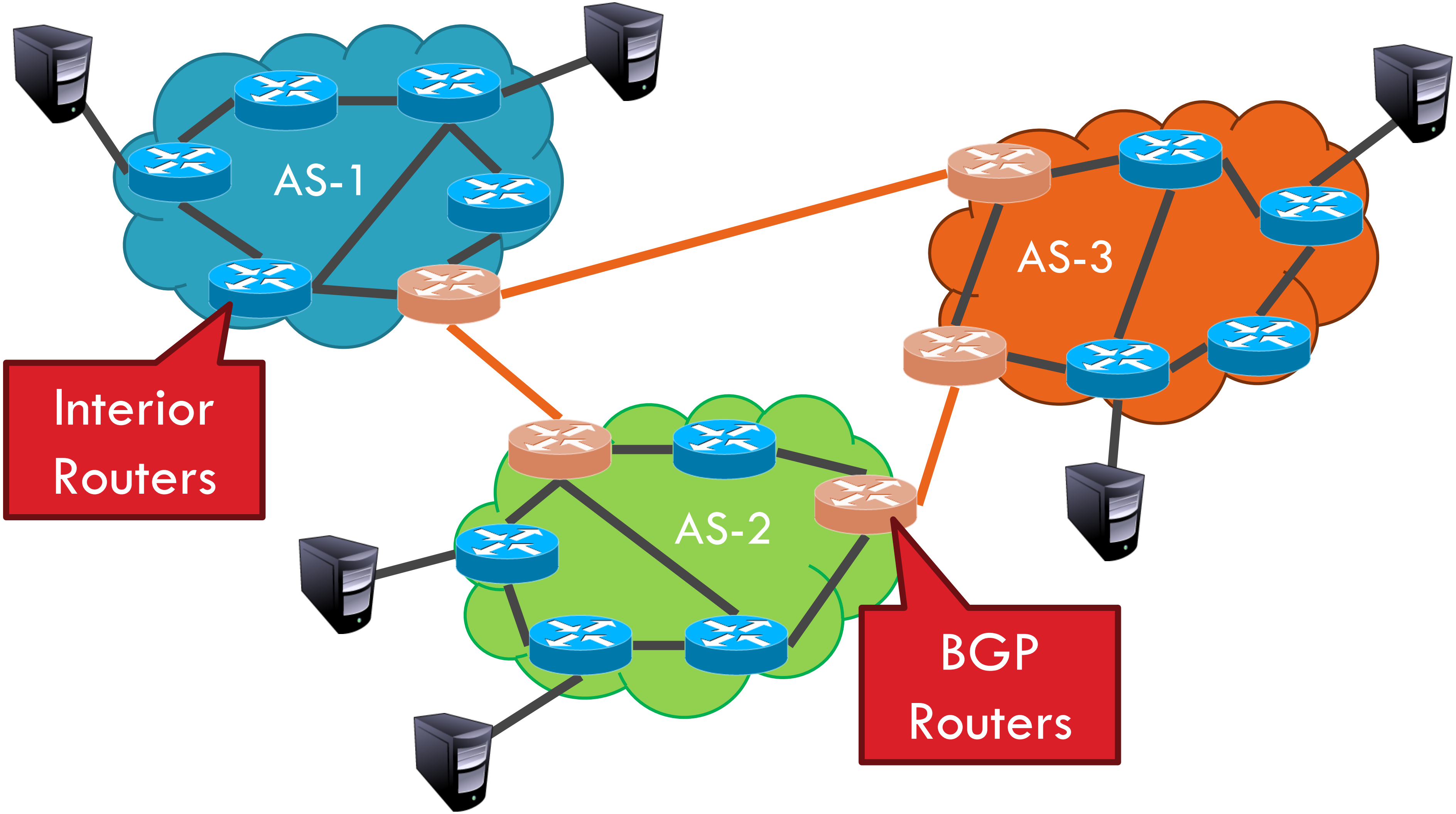
Lecture 7: IP and Intra Domain Routing

Based on slides from P. Gill and D. Choffnes
Revised 2015 by N. Carlsson

Internet Routing

- Internet organized as a two level hierarchy
- First level – autonomous systems (AS's)
  - AS – region of network under a single administrative domain
  - Examples: Comcast, AT&T, Verizon, Sprint, etc.
- AS's use intra-domain routing protocols internally
  - Distance Vector, e.g., Routing Information Protocol (RIP)
  - Link State, e.g., Open Shortest Path First (OSPF)
- Connections between AS's use inter-domain routing protocols
  - Border Gateway Routing (BGP)
  - De facto standard today, BGP-4

AS Example



Interior Routers

BGP Routers

Why Do We Need ASs?

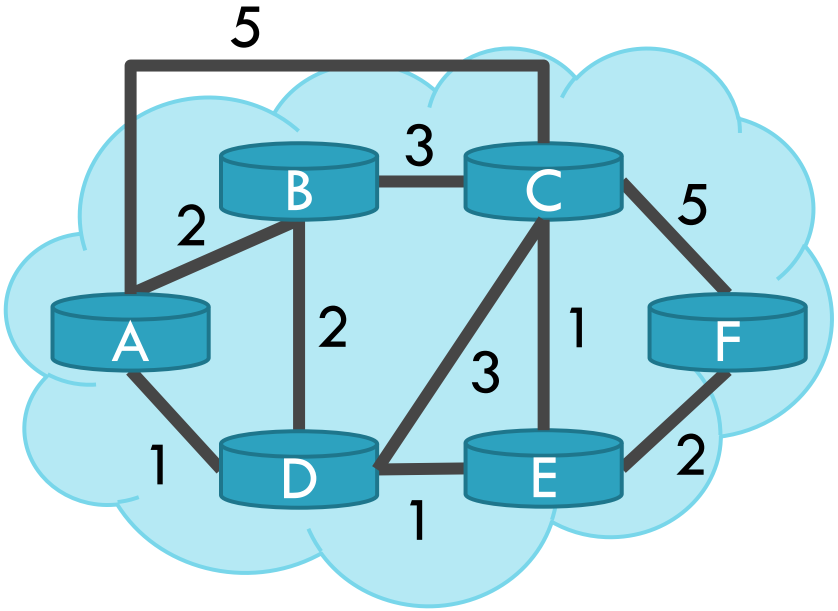
- Routing algorithms are not efficient enough to execute on the entire Internet topology
- Diffe policies
- Allo struc
- Allo s each othe

- Easier to compute routes
- Greater flexibility
- More autonomy/independence

How to find a good path?

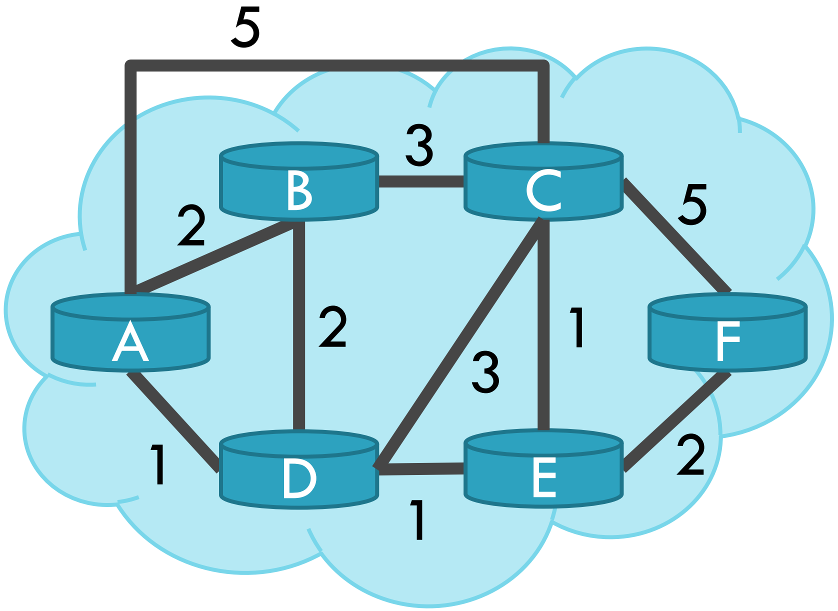
Routing on a Graph

- Goal: determine a "good" path through the network from source to destination
- What is a good path?
  - Usually means the shortest path
  - Load balanced
  - Lowest $$$ cost
- Network modeled as a graph
  - Routers → nodes
  - Link → edges
    - Edge cost: delay, congestion level, etc.



Routing Problems

- Assume
  - A network with N nodes
  - Each node only knows
    - Its immediate neighbors
    - The cost to reach each neighbor
- How does each node learn the shortest path to every other node?



Intra-domain Routing Protocols

- Distance vector
  - Routing Information Protocol (RIP), based on Bellman-Ford
  - Routers periodically exchange reachability information with neighbors
- Link state
  - Open Shortest Path First (OSPF), based on Dijkstra
  - Each network periodically floods immediate reachability information to all other routers
  - Per router local computation to determine full routes

Distance Vector Routing

- What is a distance vector?
  - Current best known cost to reach a destination
- Idea: exchange vectors among neighbors to learn about lowest cost paths

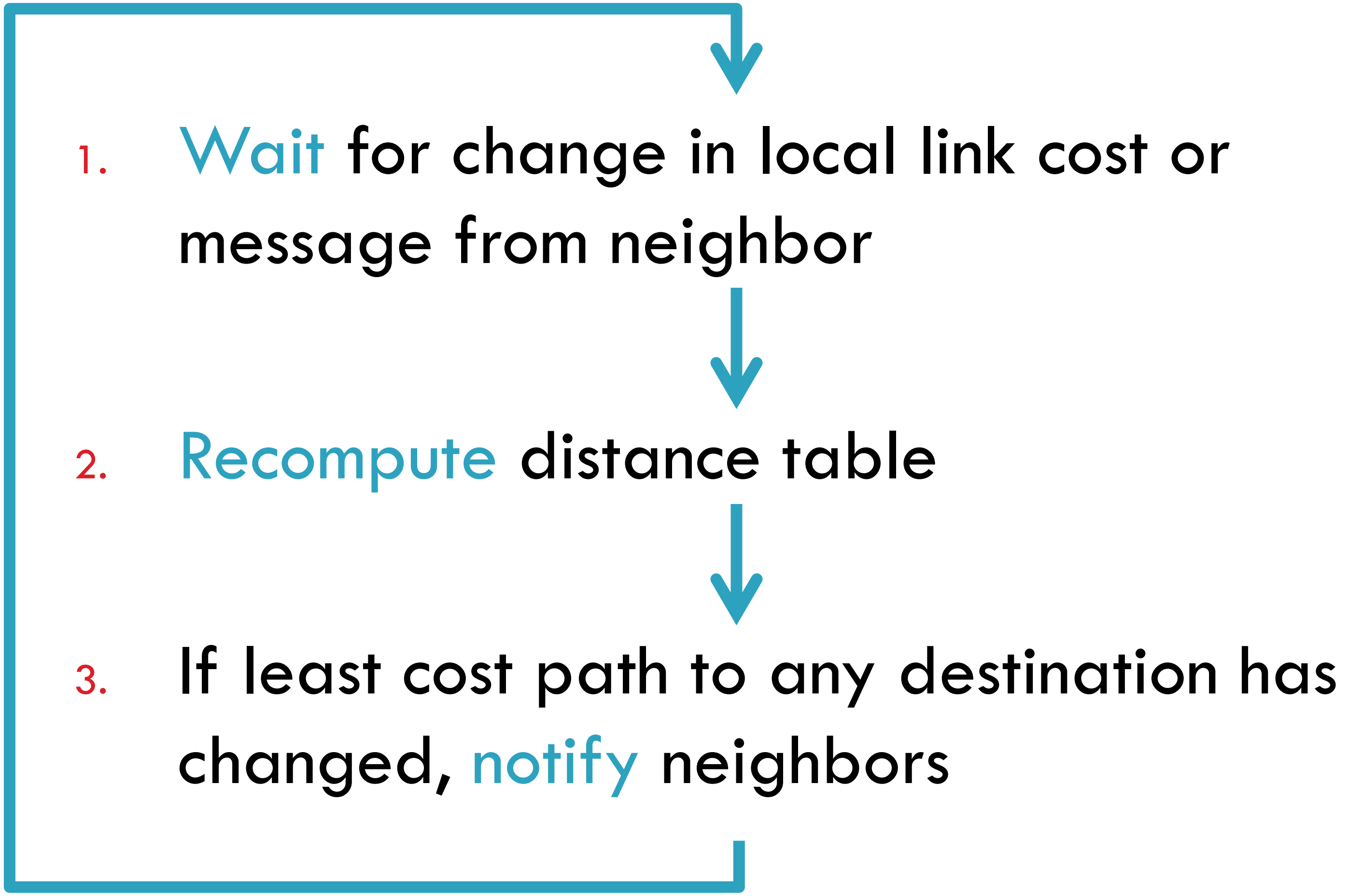
DV Table at Node C

Destination	Cost
A	7
B	1
D	2
E	5
F	1

- No entry for C
- Initially, only has info for immediate neighbors
  - Other destinations cost = ∞
- Eventually, vector is filled

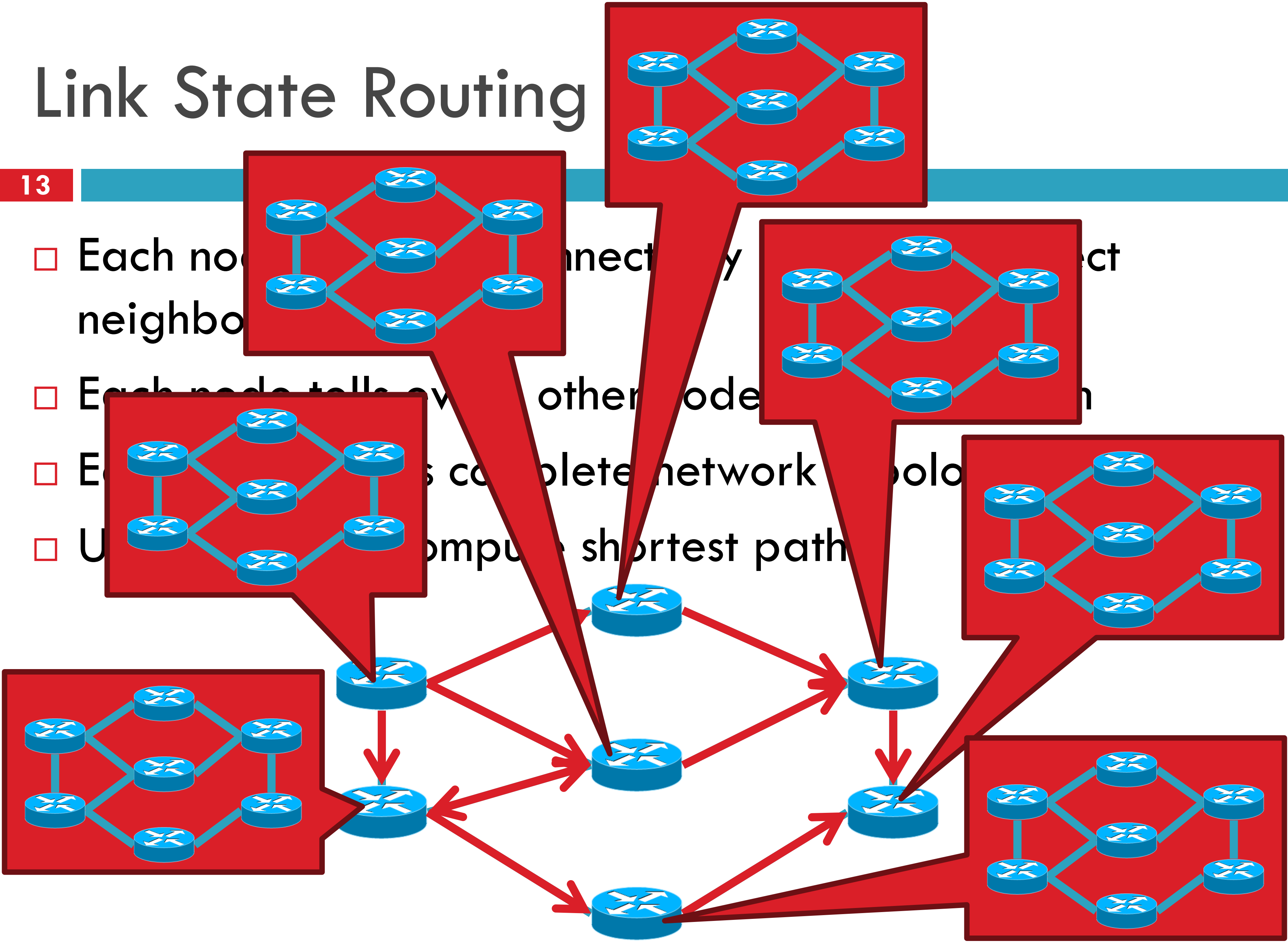
- Routing Information Protocol (RIP)

Distance Vector Routing Algorithm



1. Wait for change in local link cost or message from neighbor
2. Recompute distance table
3. If least cost path to any destination has changed, notify neighbors

Link State Routing



- ☐ Each no nnect y ct neighbo
- ☐ Each node tells ev other ode n
- ☐ E s co plete network polo
- ☐ U ompu e sh rtest path

Link State vs. Distance Vector

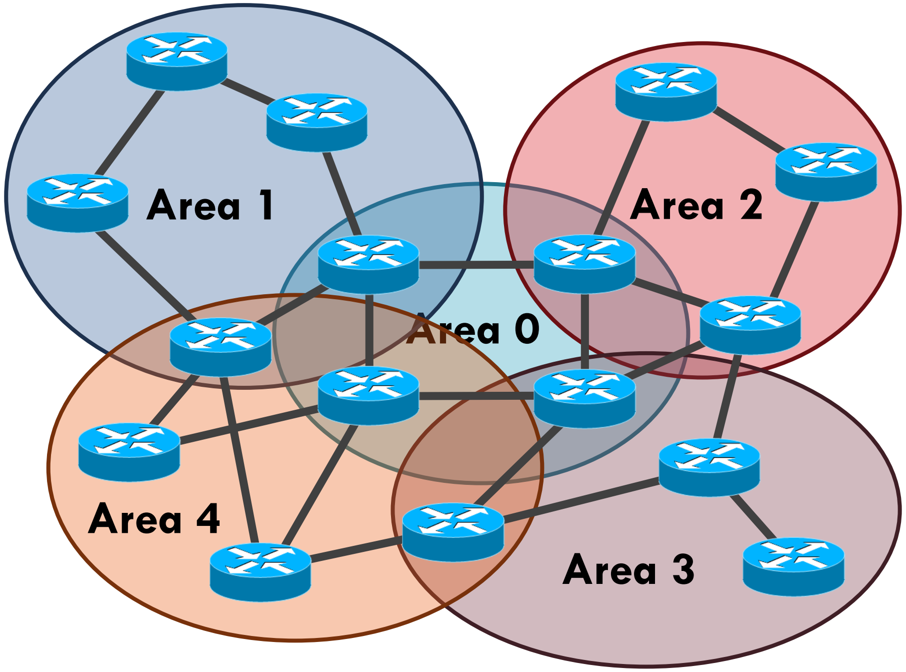
	Link State	**Distance Vector**
Message Complexity	$O(n^2*e)$	$O(d*n*k)$
Time Complexity	$O(n*\log n)$	$O(n)$
Convergence Time	$O(1)$	$O(k)$
Robustness	• Nodes may advertise incorrect link costs • Each node computes their own table	• Nodes may advertise incorrect path cost • Errors propagate due to sharing of DV tables

- Which is best?
- In practice, it depends.
- In general, link state is more popular.

Additional organization in Large ASes

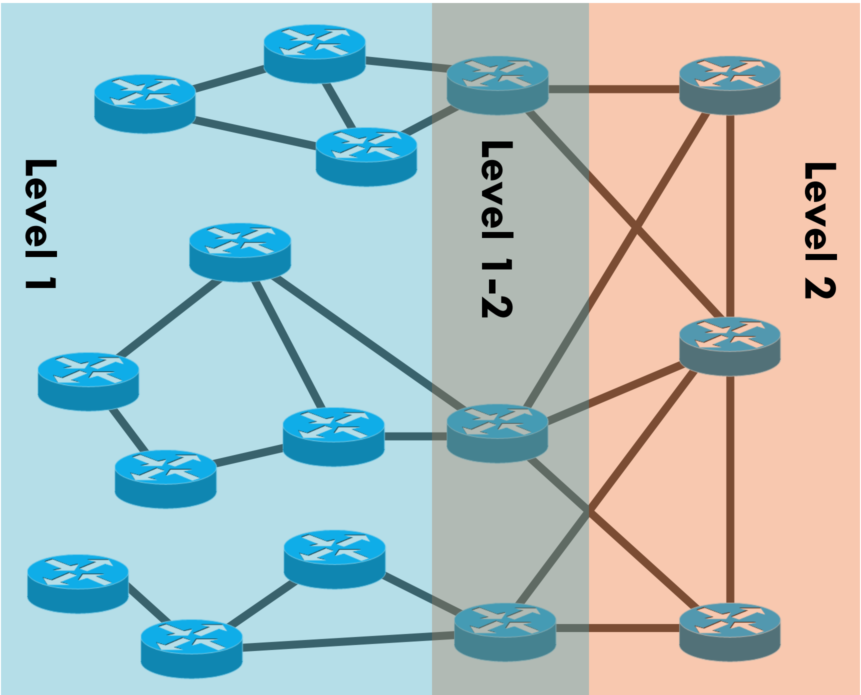
OSPF

- Organized around overlapping areas
- Area 0 is the core network



IS-IS

- Organized as a 2-level hierarchy
- Level 2 is the backbone



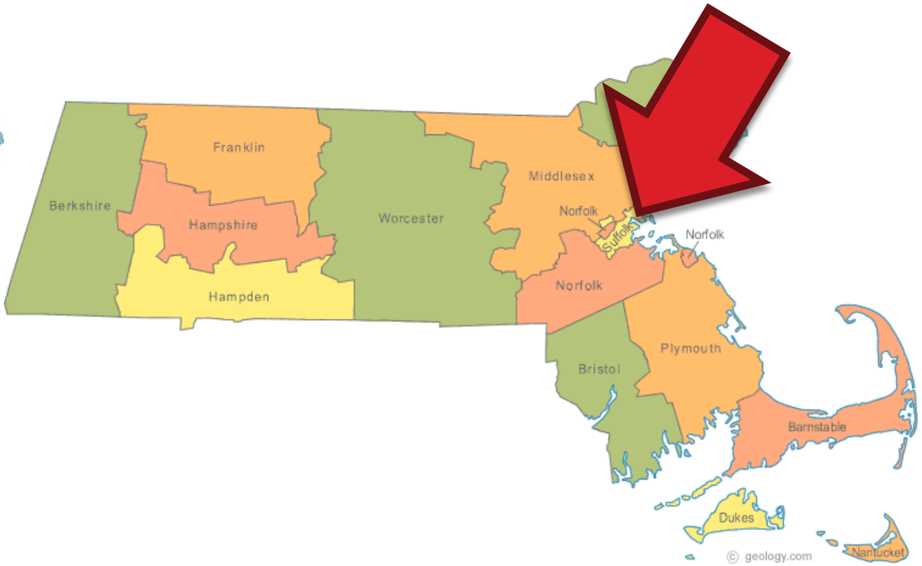
Possible Addressing Schemes

- Flat
  - e.g. each host is identified by a 48-bit MAC address
  - Router needs an entry for every host in the world
    - Too big
    - Too hard to maintain (hosts come and go all the time)
    - Too slow (more later)
- Hierarchy
  - Addresses broken down into segments
  - Each segment has a different level of specificity

Example: Telephone Numbers

1-617-373- 3278

Very General



Northeastern University

West Village H
Room 256

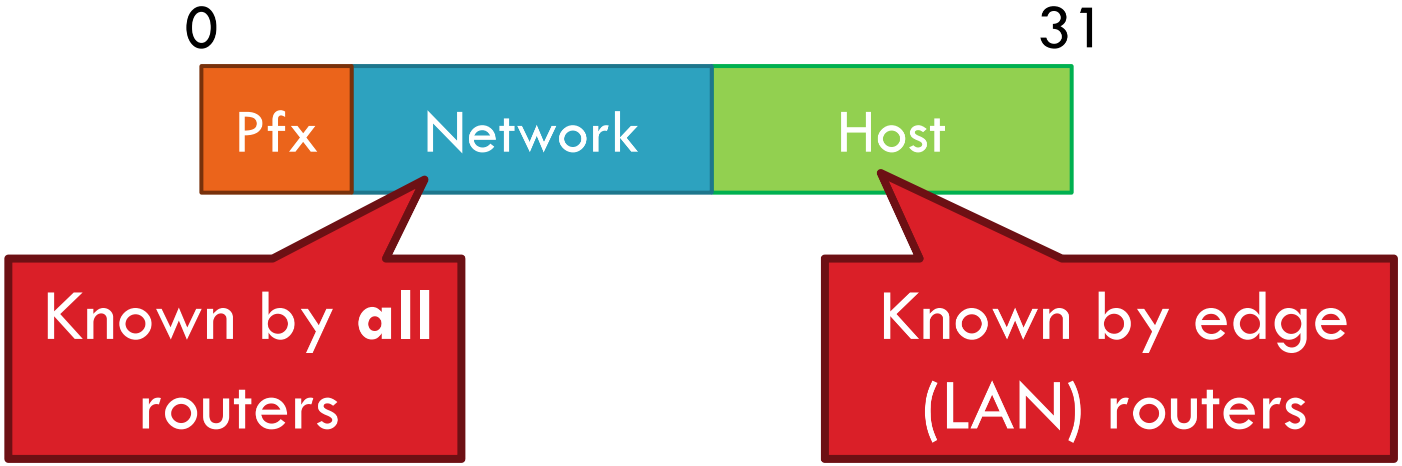
West Village G
Room 1234

Updates are Local

Very Specific

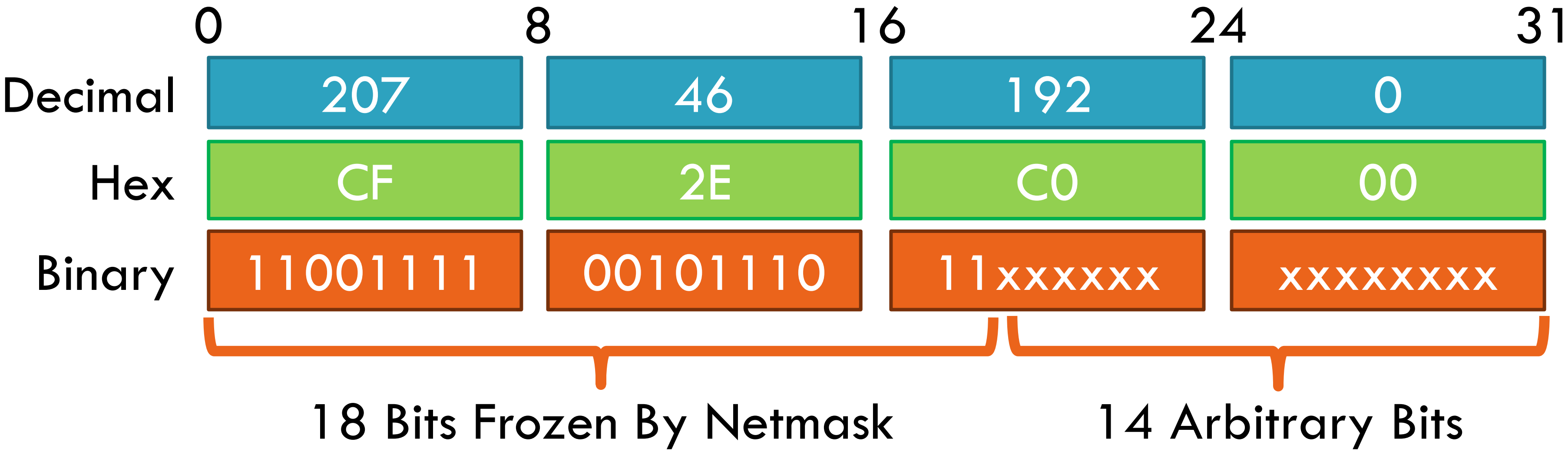
IP Addressing and Forwarding

- Routing Table Requirements
  - For every possible IP, give the next hop
  - But for 32-bit addresses, 2^{32} possibilities!
  - **Too slow:** 48GE ports and 4x10GE needs 176Gbps bandwidth
    DRAM: ~1-6 Gbps; **TCAM** is fast, but 400x cost of DRAM
- Hierarchical address scheme
  - Separate the address into a network and a host



Aggregation with CIDR

- Classless inter-domain routing (CIDR)
  - Allow variable sized network parts (prefixes)
- One organization given contiguous IP ranges
  - Example: Microsoft, 207.46.192.* – 207.46.255.*
  - Specified as CIDR address 207.46.192.0/18



Decimal	207	46	192	0
Hex	CF	2E	C0	00
Binary	11001111	00101110	11xxxxxx	xxxxxxxx

18 Bits Frozen By Netmask

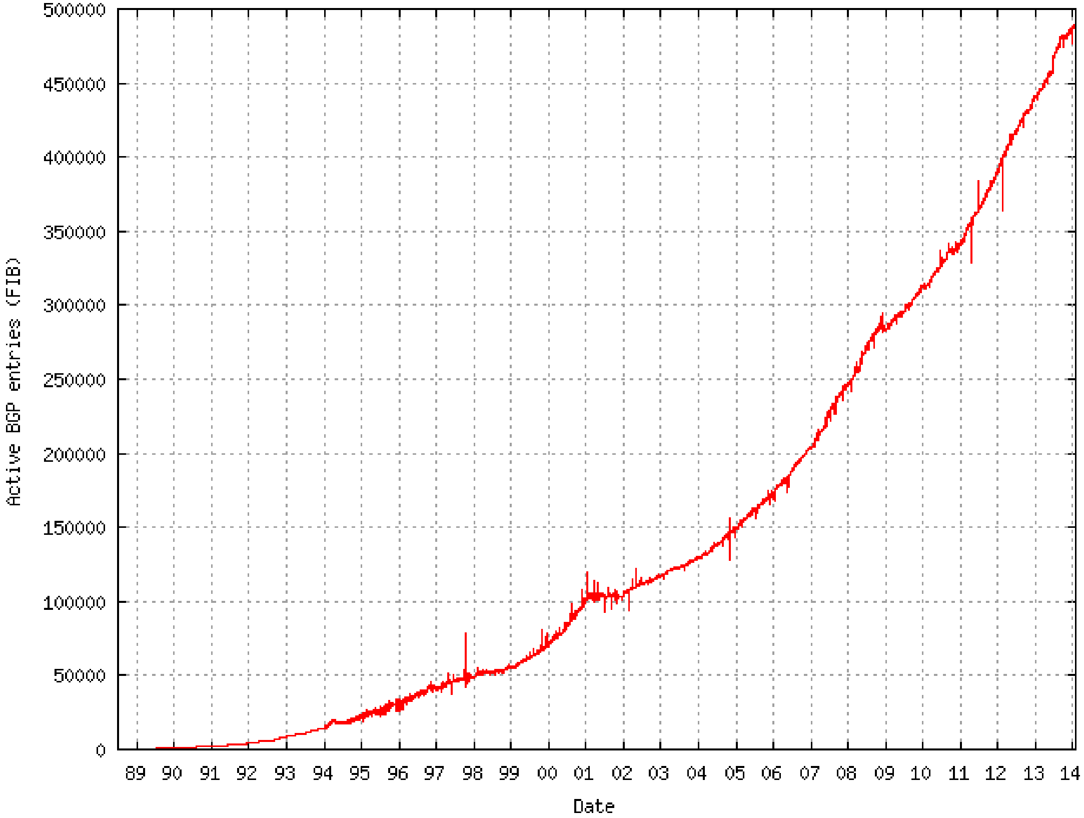
14 Arbitrary Bits

Example CIDR Routing Table

Address	Netmask	Third Byte	Byte Range
207.46.0.0	19	000xxxxx	0 – 31
207.46.32.0	19	001xxxxx	32 – 63
207.46.64.0	19	010xxxxx	64 – 95
207.46.128.0	18	10xxxxxx	128 – 191
207.46.192.0	18	11xxxxxx	192 – 255

Hole in the Routing Table: No coverage for 96 – 127
207.46.96.0/19

Size of CIDR Routing Tables



- From www.cidr-report.org
- CIDR has kept IP routing table sizes in check
  - Currently ~500,000 entries for a complete IP routing table
  - Only required by backbone routers

We had a special day this summer!

- 512K day – August 12, 2014
- Default threshold size for IPv4 route data in older Cisco routers → 512K routes
    - Some routers failed over to slower memory
        - RAM vs. CAM (content addressable memory)
    - Some routes dropped
- Cisco issues update in May anticipating this issue
    - Reallocated some IPv6 space for IPv4 routes
- http://cacm.acm.org/news/178293-internet-routing-failures-bring-architecture-changes-back-to-the-table/fulltext

How Do You Get IPs?

- IP address ranges controlled by IANA
  - Internet Assigned Number Authority
  - Roots go back to 1972, ARPANET, UCLA
  - Today, part of ICANN
- IANA grants IPs to regional authorities (RIRs)
  - E.g., RIPE (Europe, Middle East), ARIN (North America), APNIC (Asia/Pacific), AfriNIC (Africa), and LACNIC (Latin America) may grant you a range of IPs
  - You may then advertise routes to your new IP range
  - There are now secondary markets, auctions, …

The IPv4 Address Space Crisis

- Problem: the IPv4 address space is too small
  - 2^{32} = 4,294,967,296 possible addresses
  - Less than one IP per person
- Parts of the world have already run out of addresses
  - IANA assigned the last /8 block of addresses in 2011

Region	Regional Internet Registry (RIR)	Exhaustion Date
Asia/Pacific	APNIC	April 19, 2011
Europe/Middle East	RIPE	September 14, 2012
North America	ARIN	13 Jan 2015 (Projected)
South America	LACNIC	13 Jan 2015 (Projected)
Africa	AFRINIC	17 Jan 2022(Projected)

IPv6

- IPv6, first introduced in 1998(!)
  - 128-bit addresses
  - 4.8 * 10^{28} addresses per person
- Address format
  - 8 groups of 16-bit values, separated by ‘:’
  - Leading zeroes in each group may be omitted
  - Groups of zeroes can be omitted using ‘::’

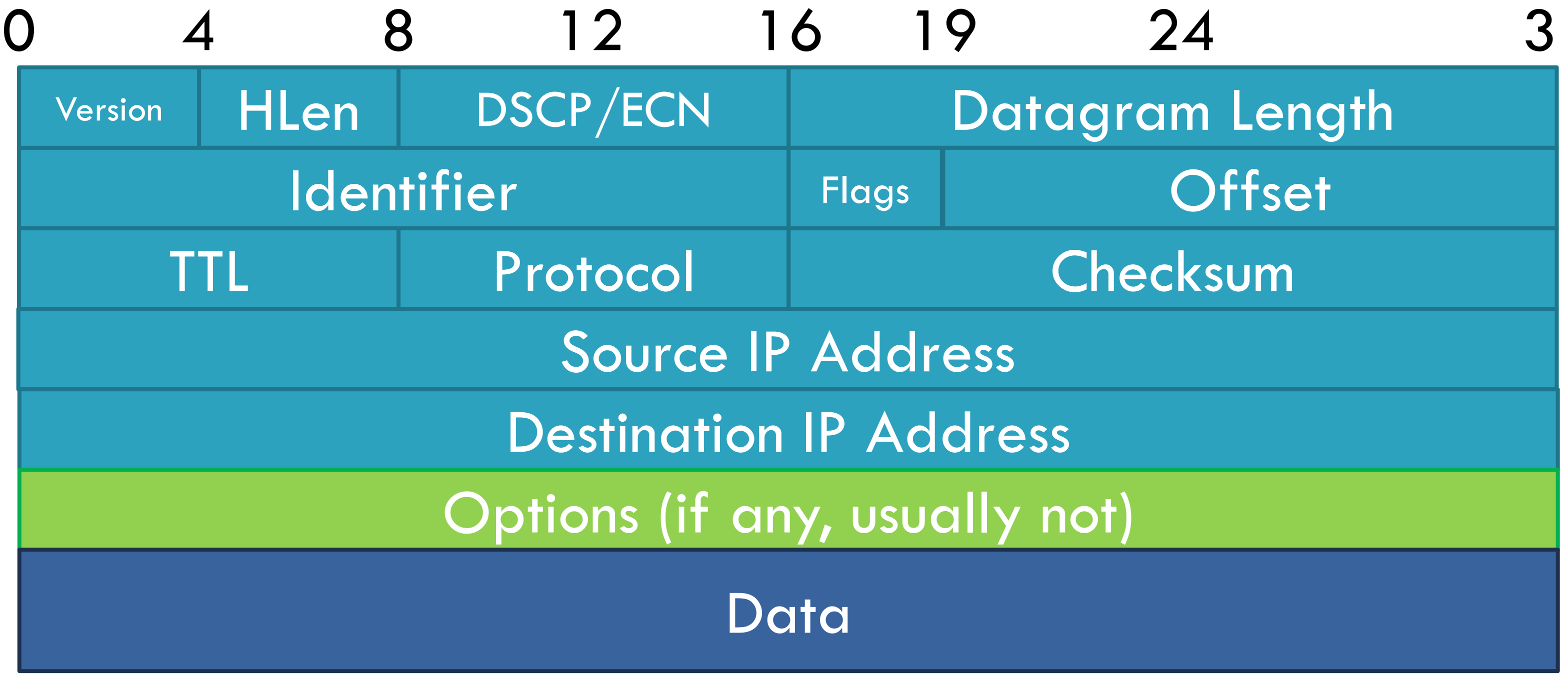
2001:0db8:0000:0000:0000:ff00:0042:8329

2001:0db8:0:0:0:ff00:42:8329

2001:0db8::ff00:42:8329

IPv4 Header

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

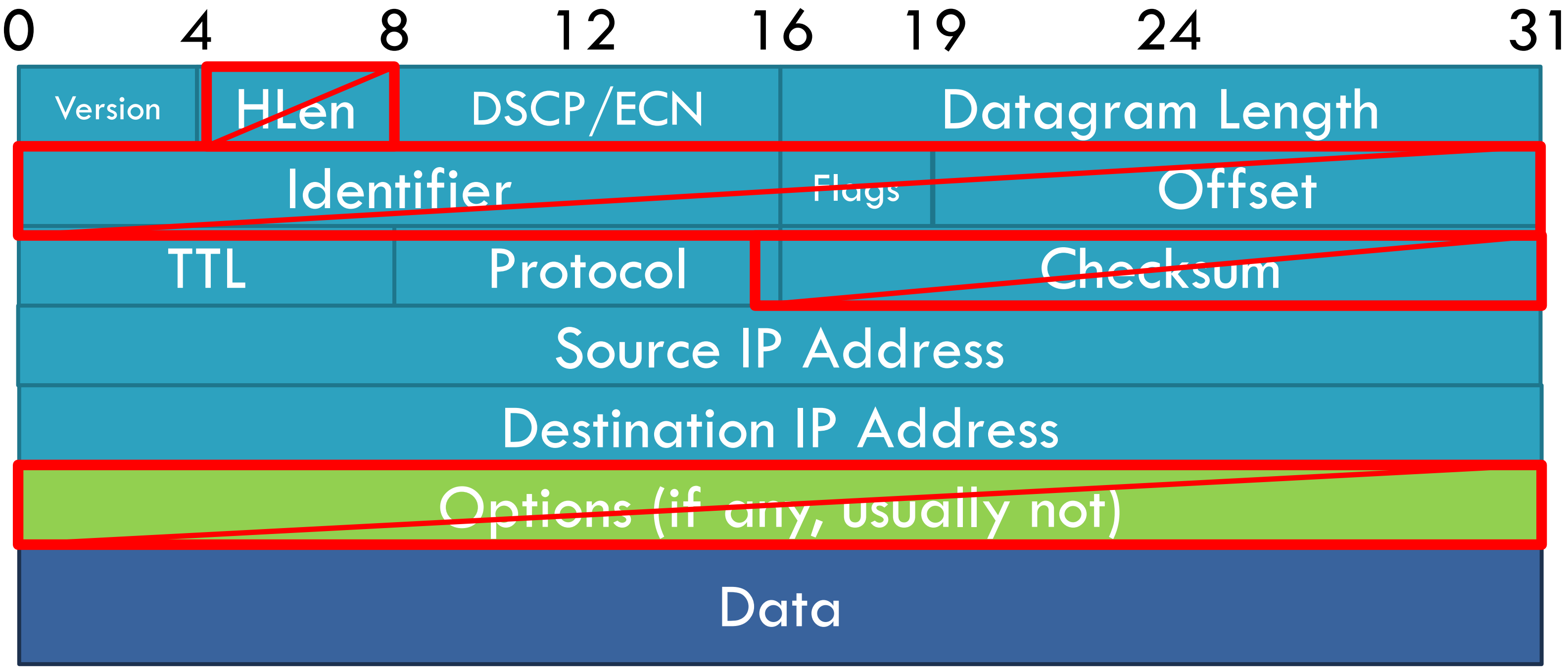


0 4 8 12 16 19 24 31

<table>
<tr><td>Version</td><td>HLen</td><td colspan="2">DSCP/ECN</td><td colspan="3">Datagram Length</td></tr>
<tr><td colspan="4">Identifier</td><td>Flags</td><td colspan="2">Offset</td></tr>
<tr><td colspan="2">TTL</td><td colspan="2">Protocol</td><td colspan="3">Checksum</td></tr>
<tr><td colspan="7">Source IP Address</td></tr>
<tr><td colspan="7">Destination IP Address</td></tr>
<tr><td colspan="7">Options (if any, usually not)</td></tr>
<tr><td colspan="7">Data</td></tr>
</table>

IPv4 Header

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



0 4 8 12 16 19 24 31

Version ~~HLen~~ DSCP/ECN Datagram Length

~~Identifier~~ ~~Flags~~ ~~Offset~~

TTL Protocol ~~Checksum~~

Source IP Address

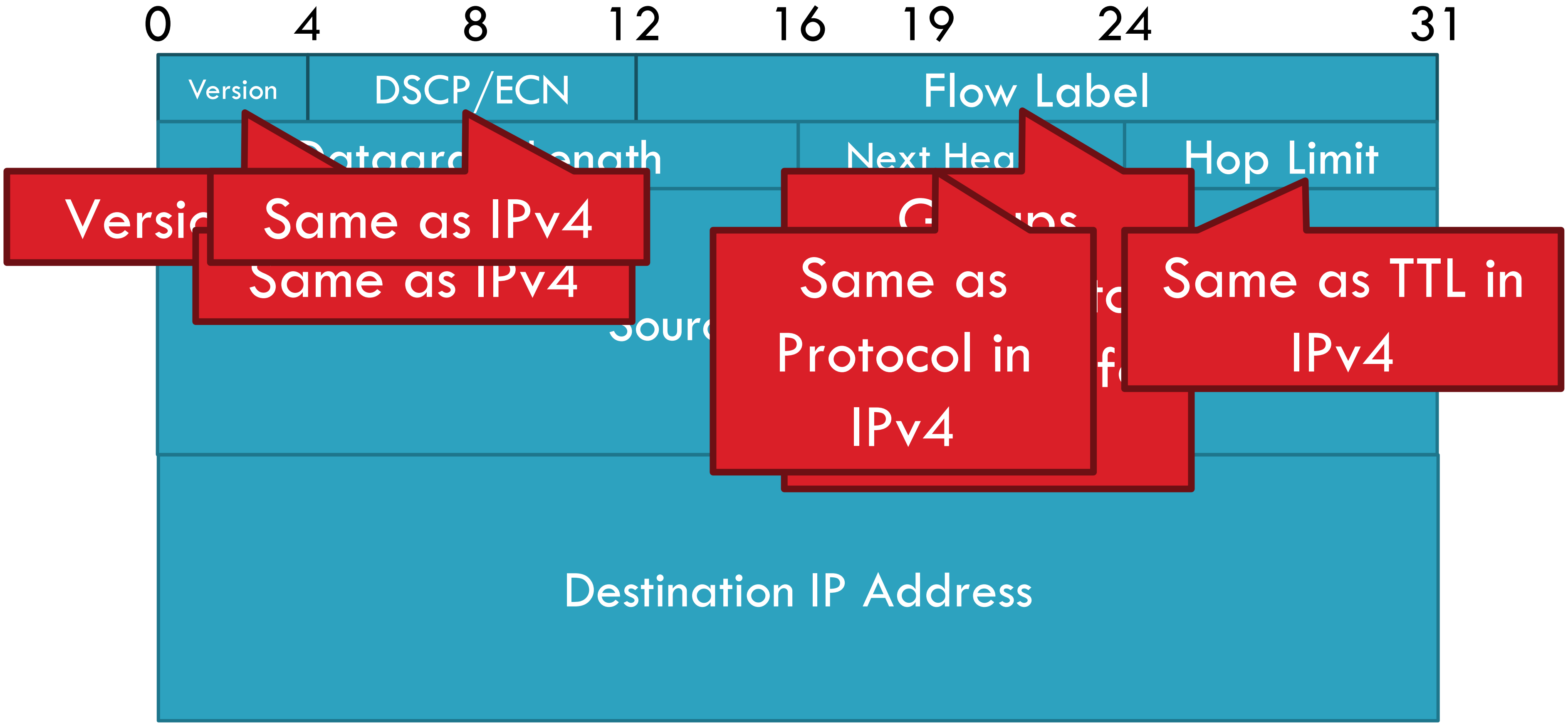
Destination IP Address

~~Options (if any, usually not)~~

Data

IPv6 Header

- Double the size of IPv4 (320 bits vs. 160 bits)



0 4 8 12 16 19 24 31

Version

DSCP/ECN

Flow Label

atagra ength

Next Hea

Hop Limit

Sourc

Destination IP Address

Versi

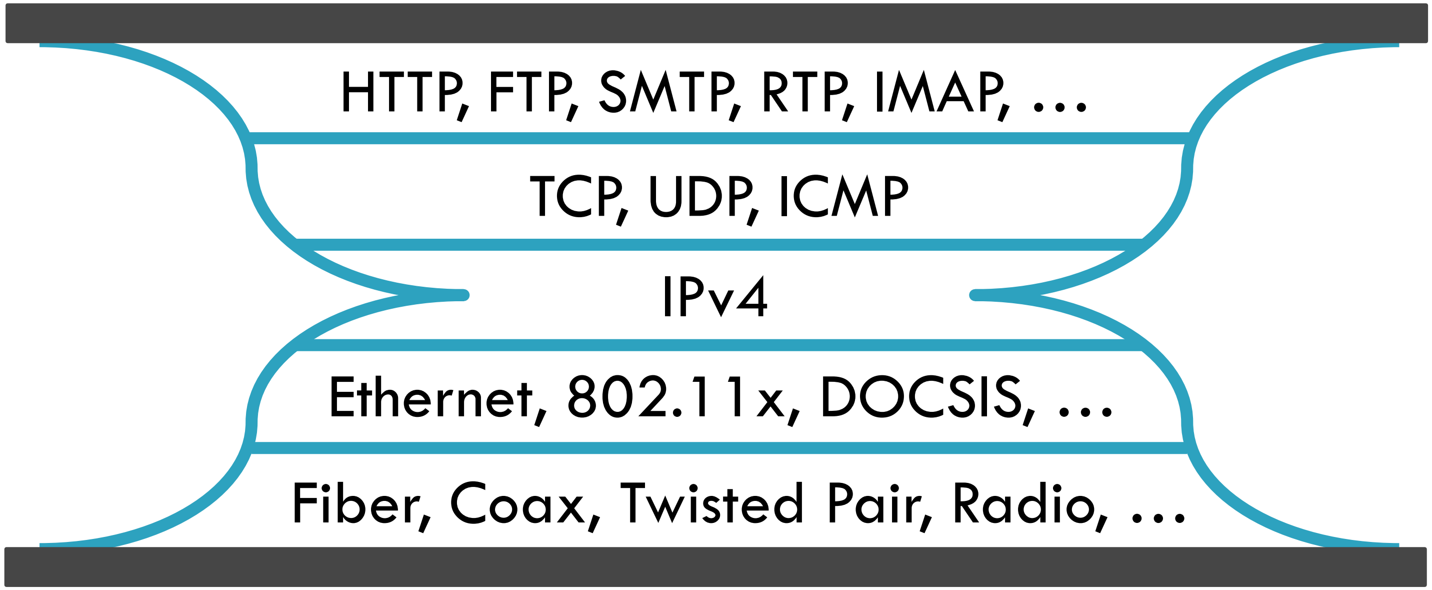
Same as IPv4

Same as IPv4

Same as Protocol in IPv4

Same as TTL in IPv4

Deployment Challenges



HTTP, FTP, SMTP, RTP, IMAP, …

TCP, UDP, ICMP

IPv4

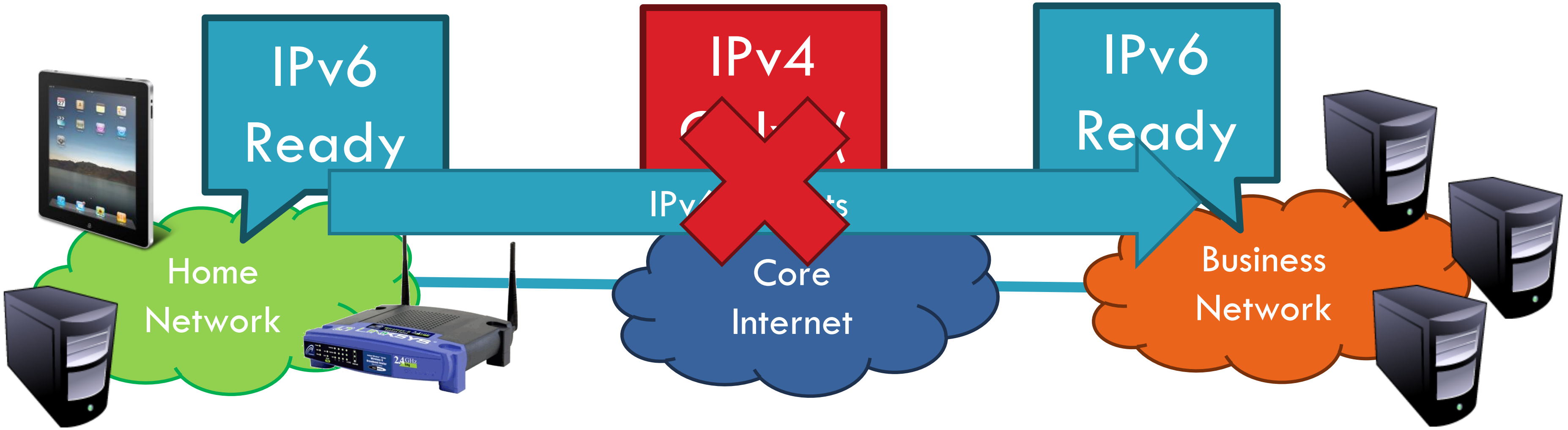
Ethernet, 802.11x, DOCSIS, …

Fiber, Coax, Twisted Pair, Radio, …

- Switching to IPv6 is a whole-Internet upgrade
  - All routers, all hosts
  - ICMPv6, DHCPv6, DNSv6
- 2013: 0.94% of Google traffic was IPv6, 2.5% today

Transitioning to IPv6

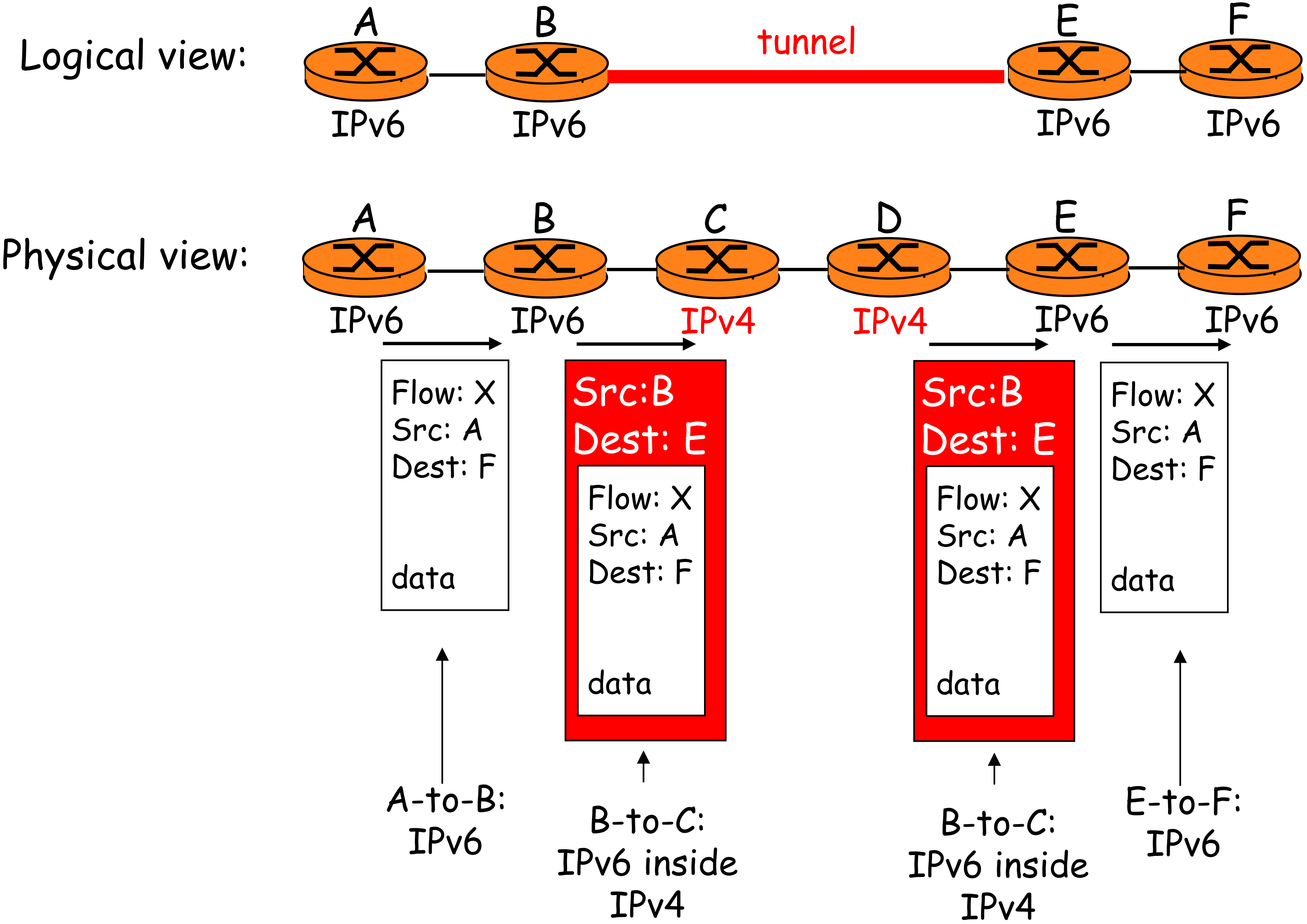
- How do we ease the transition from IPv4 to IPv6?
  - Today, most client devices are IPv6 ready
    - Windows/OSX/iOS/Android all support IPv6
    - Your wireless access point probably supports IPv6
  - The end-to-end network is harder to upgrade
  - … but a IPv4 core cannot route IPv6 traffic



Transition Technologies

- How do you route IPv6 packets over an IPv4 Internet?
- Transition Technologies
  - Use tunnels to encapsulate and route IPv6 packets over the IPv4 Internet
  - Several different implementations
    - 6to4
    - IPv6 Rapid Deployment (6rd)
    - Teredo
    - … etc.

Tunneling



Flow: X
Src: A
Dest: F

data

Src:B
Dest: E

Flow: X
Src: A
Dest: F

data

Src:B
Dest: E

Flow: X
Src: A
Dest: F

data

Flow: X
Src: A
Dest: F

data

A-to-B:
IPv6

B-to-C:
IPv6 inside
IPv4

B-to-C:
IPv6 inside
IPv4

E-to-F:
IPv6

More slides …

Differences from IPv4 Header

- Several header fields are missing in IPv6
  - Header length – rolled into Next Header field
  - Checksum – was useless, so why keep it
  - Identifier, Flags, Offset
    - IPv6 routers do not support fragmentation
    - Hosts are expected to use path MTU discovery
- Reflects changing Internet priorities
  - Today's networks are more homogeneous
  - Instead, routing cost and complexity dominate

Performance Improvements

- No checksums to verify
- No need for routers to handle fragmentation
- Simplified routing table design
  - Address space is huge
  - No need for CIDR (but need for aggregation)
  - Standard subnet size is 2^{64} addresses
- Simplified auto-configuration

Additional IPv6 Features

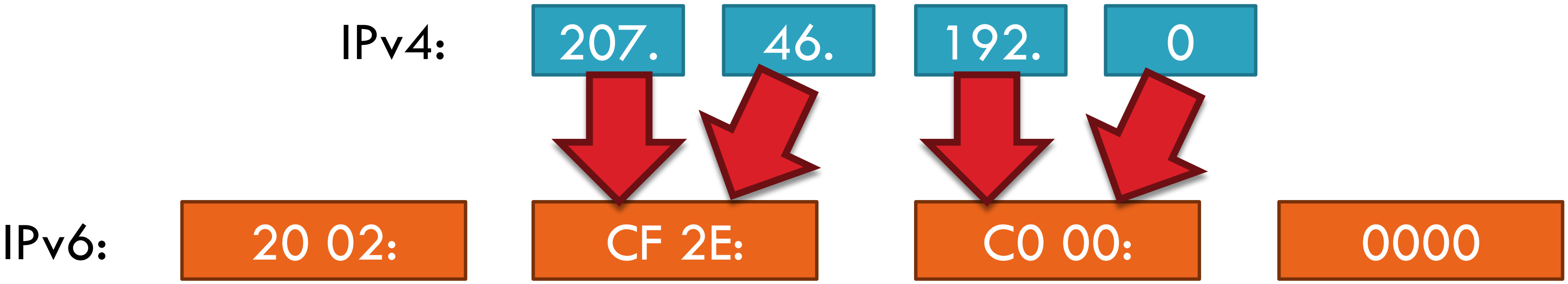
- Source Routing
  - Host specifies the route to wants packet to take
- Mobile IP
  - Hosts can take their IP with them to other networks
  - Use source routing to direct packets
- Privacy Extensions
  - Randomly generate host identifiers
  - Make it difficult to associate one IP to a host
- Jumbograms
  - Support for 4Gb datagrams

Consequences of IPv6

- Beware unintended consequences of IPv6
- Example: IP blacklists
  - Currently, blacklists track IPs of spammers/bots
  - Few IPv4 addresses mean list sizes are reasonable
  - Hard for spammers/bots to acquire new IPs
- Blacklists will not work with IPv6
  - Address space is enormous
  - Acquiring new IP addresses is trivial

6to4 Basics

- Problem: you've been assigned an IPv4 address, but you want an IPv6 address
  - Your ISP can't or won't give you an IPv6 address
  - You can't just arbitrarily choose an IPv6 address
- Solution: construct a 6to4 address
  - 6to4 addresses always start with 2002::
  - Embed the 32-bit IPv4 inside the 128-bit IPv6 address

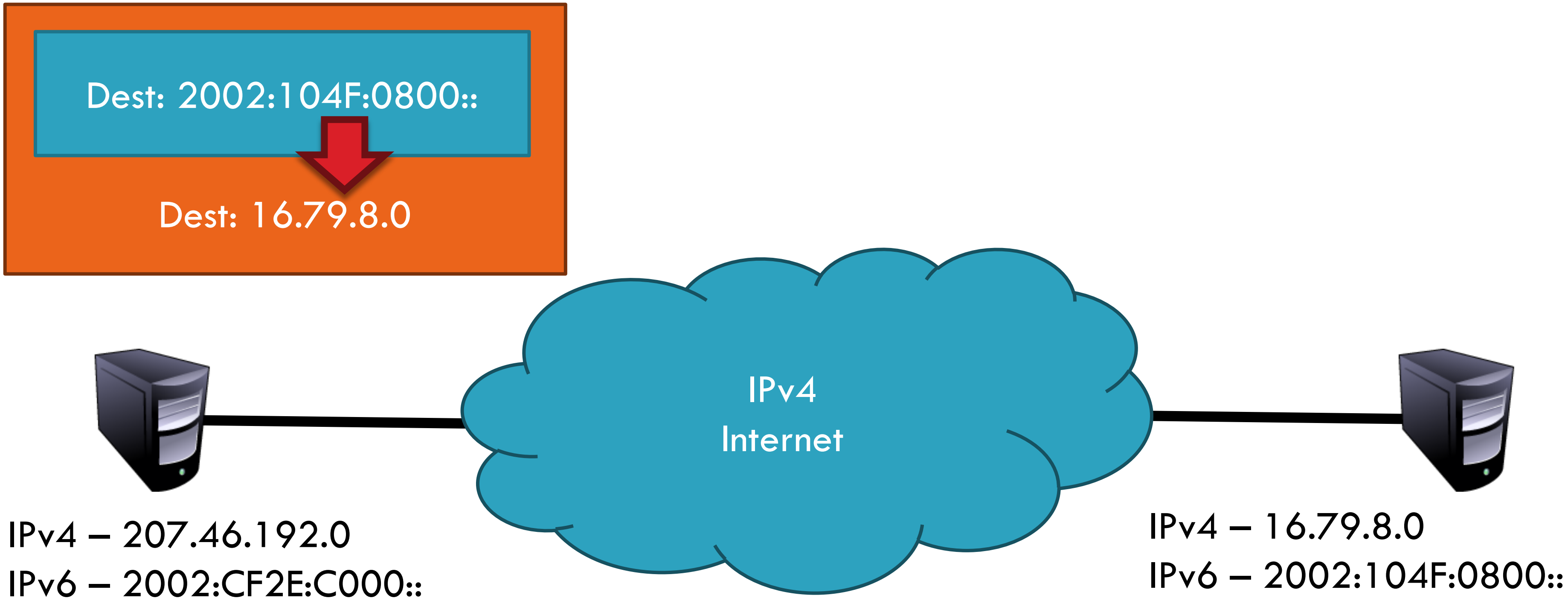


IPv4: 207. 46. 192. 0

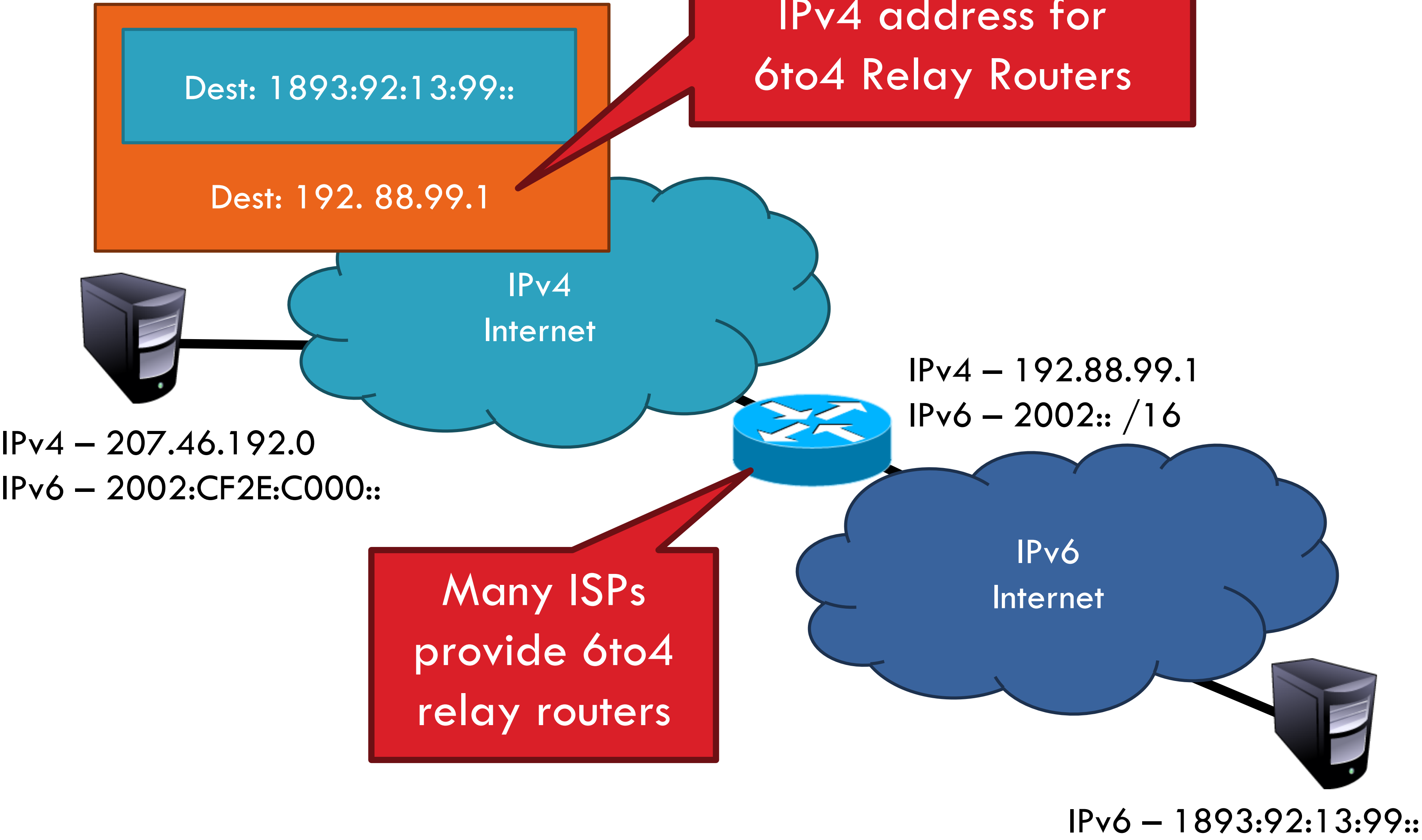
IPv6: 20 02: CF 2E: C0 00: 0000

Routing from 6to4 to 6to4

- How does a host using 6to4 send a packet to another host using 6to4?



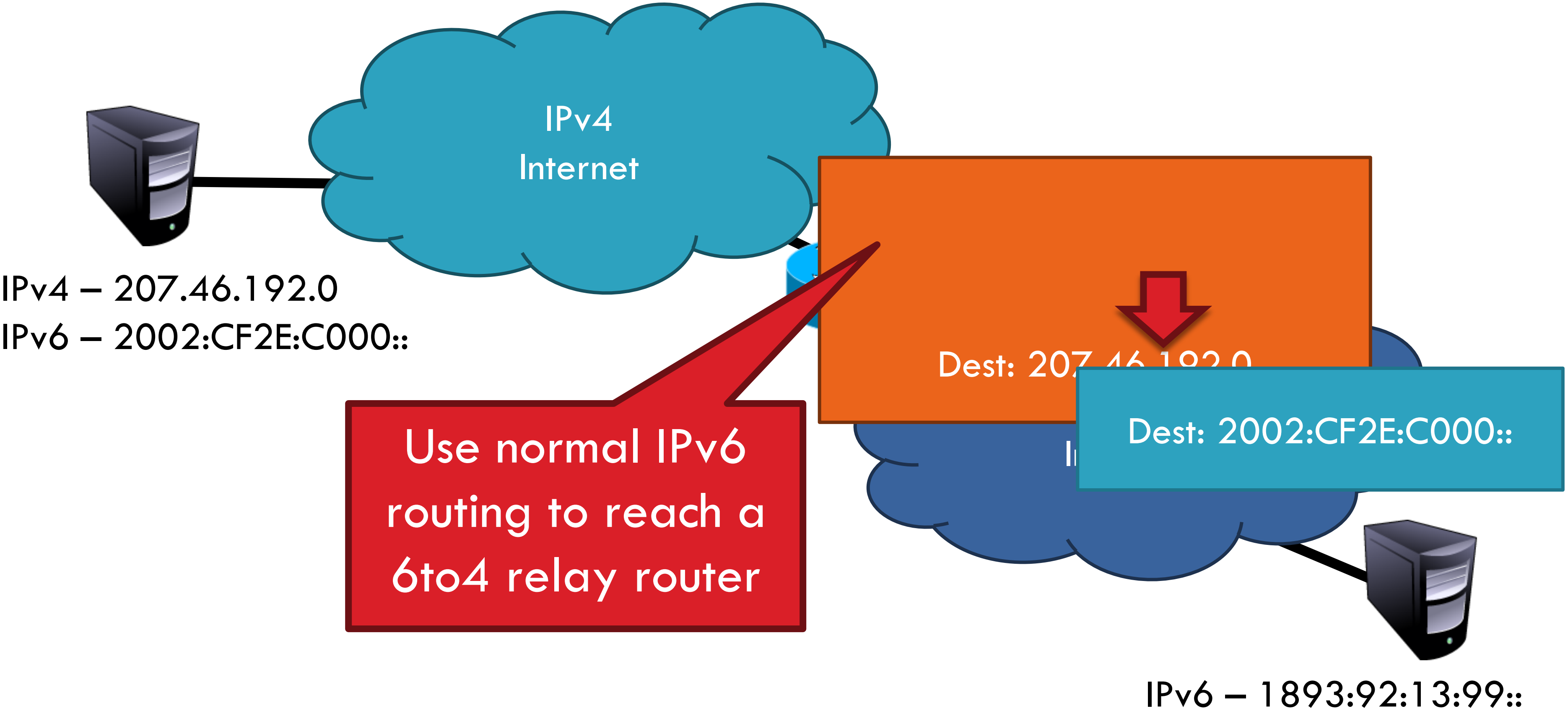
Routing from 6to4 to Native IPv6



Special, anycasted IPv4 address for 6to4 Relay Routers

Many ISPs provide 6to4 relay routers

Routing from Native IPv6 to 6to4



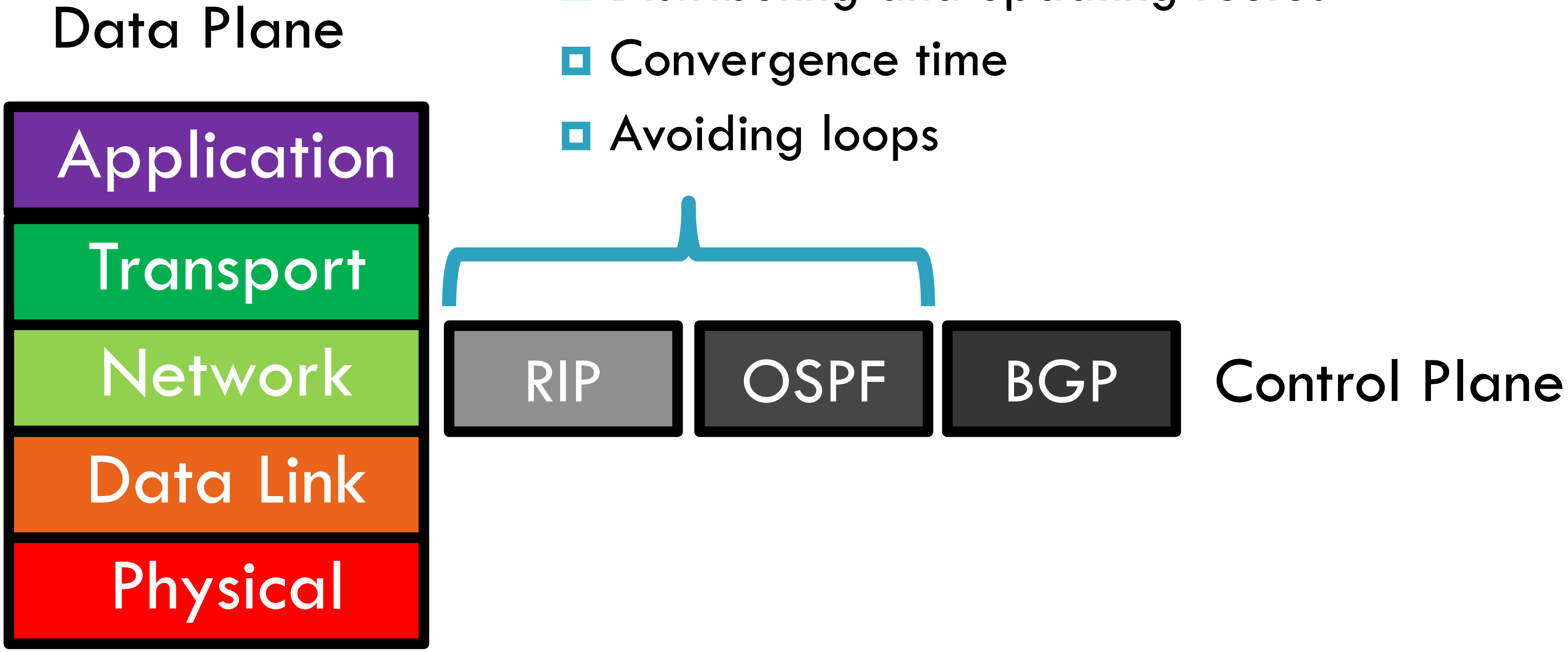
Use normal IPv6 routing to reach a 6to4 relay router

Problems with 6to4

- Uniformity
  - Not all ISPs have deployed 6to4 relays
- Quality of service
  - Third-party 6to4 relays are available
  - …but, they may be overloaded or unreliable
- Reachability
  - 6to4 doesn’t work if you are behind a NAT
- Possible solutions
  - IPv6 Rapid Deployment (6rd)
    - Each ISP sets up relays for its customers
    - Does not leverage the 2002:: address space
  - Teredo
    - Tunnels IPv6 packets through UDP/IPv4 tunnels
    - Can tunnel through NATs, but requires special relays

Network Layer, Control Plane

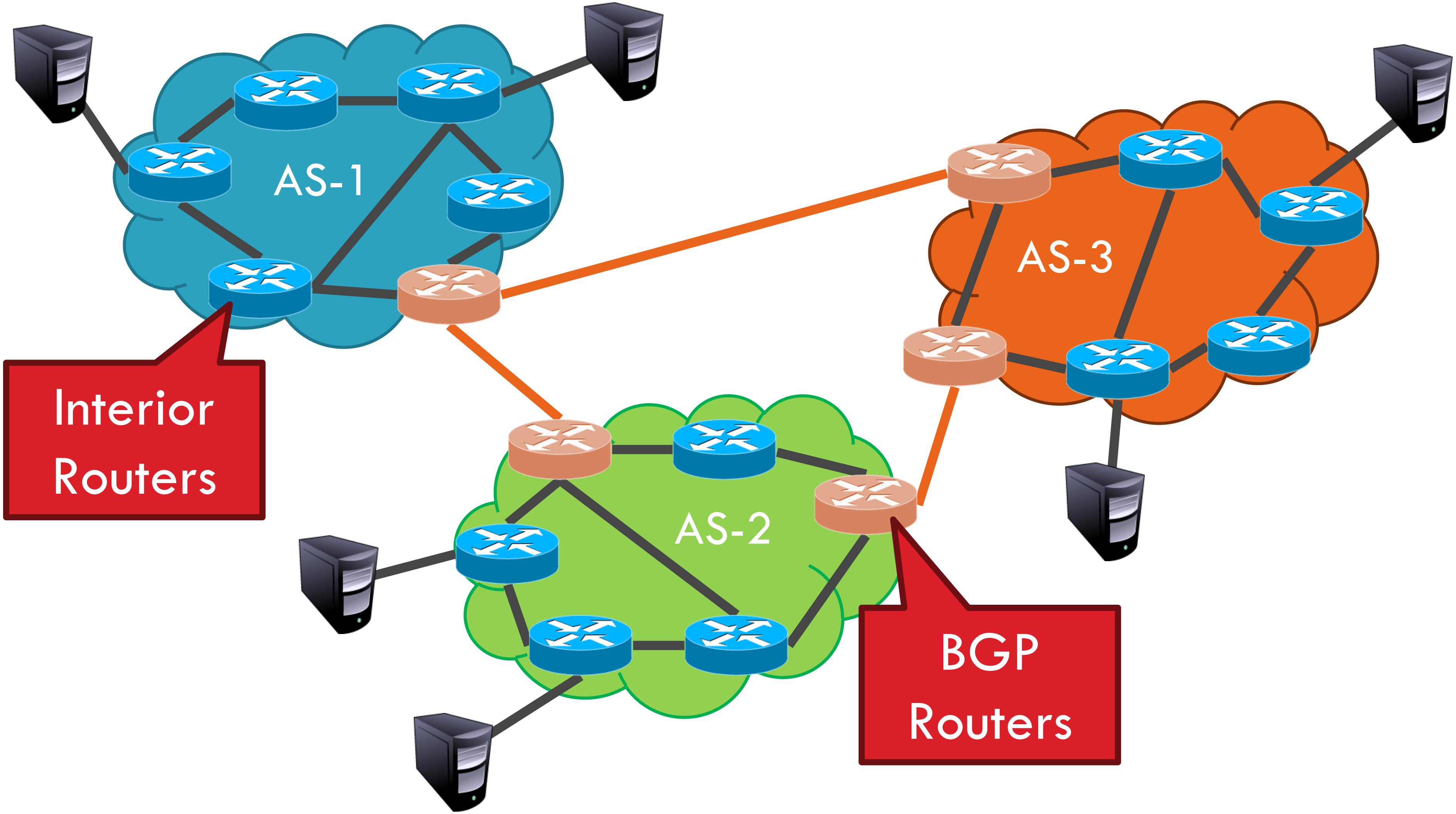
- Function:
  - Set up routes within a single network
- Key challenges:
  - Distributing and updating routes
  - Convergence time
  - Avoiding loops



Internet Routing

- Internet organized as a two level hierarchy
- First level – autonomous systems (AS's)
  - AS – region of network under a single administrative domain
  - Examples: Comcast, AT&T, Verizon, Sprint, etc.
- AS's use intra-domain routing protocols internally
  - Distance Vector, e.g., Routing Information Protocol (RIP)
  - Link State, e.g., Open Shortest Path First (OSPF)
- Connections between AS's use inter-domain routing protocols
  - Border Gateway Routing (BGP)
  - De facto standard today, BGP-4

AS Example



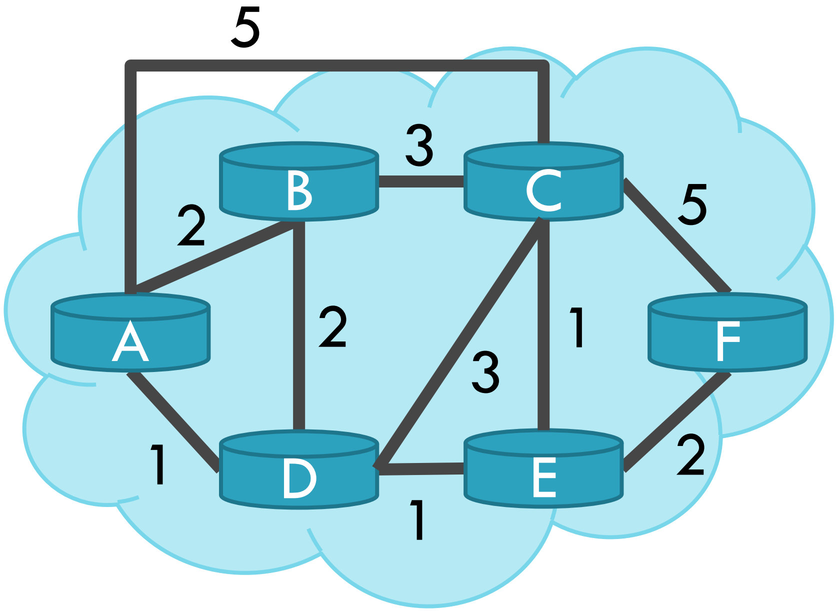
Why Do We Need ASs?

- Routing algorithms are not efficient enough to execute on the entire Internet topology
- Diffe policies
- Allo
  struc
- Allo s each
  othe

- Easier to compute routes
- Greater flexibility
- More autonomy/independence

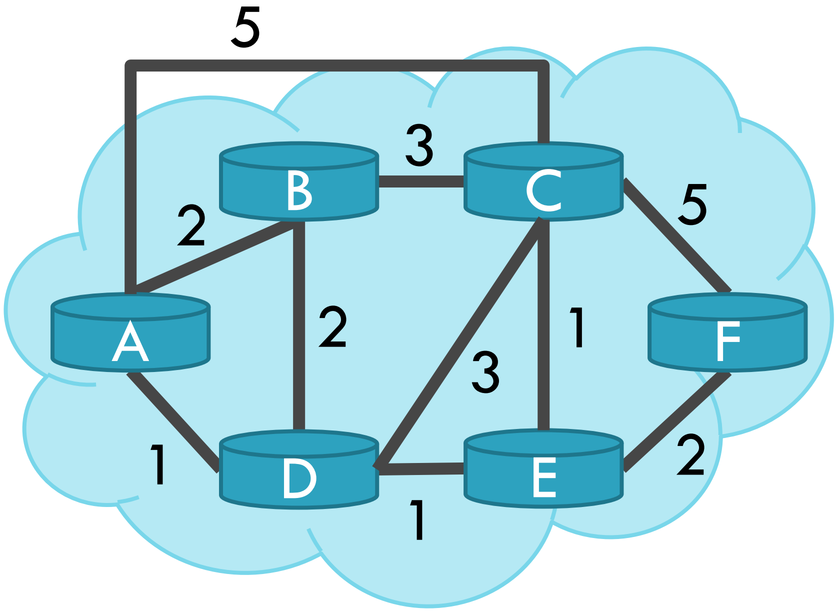
Routing on a Graph

- Goal: determine a “good” path through the network from source to destination
- What is a good path?
  - Usually means the shortest path
  - Load balanced
  - Lowest $$$ cost
- Network modeled as a graph
  - Routers → nodes
  - Link → edges
    - Edge cost: delay, congestion level, etc.



Routing Problems

- Assume
  - A network with N nodes
  - Each node only knows
    - Its immediate neighbors
    - The cost to reach each neighbor
- How does each node learn the shortest path to every other node?



Intra-domain Routing Protocols

- Distance vector
  - Routing Information Protocol (RIP), based on Bellman-Ford
  - Routers periodically exchange reachability information with neighbors
- Link state
  - Open Shortest Path First (OSPF), based on Dijkstra
  - Each network periodically floods immediate reachability information to all other routers
  - Per router local computation to determine full routes

Outline

- Distance Vector Routing
  - RIP
- Link State Routing
  - OSPF
  - IS-IS

Distance Vector Routing

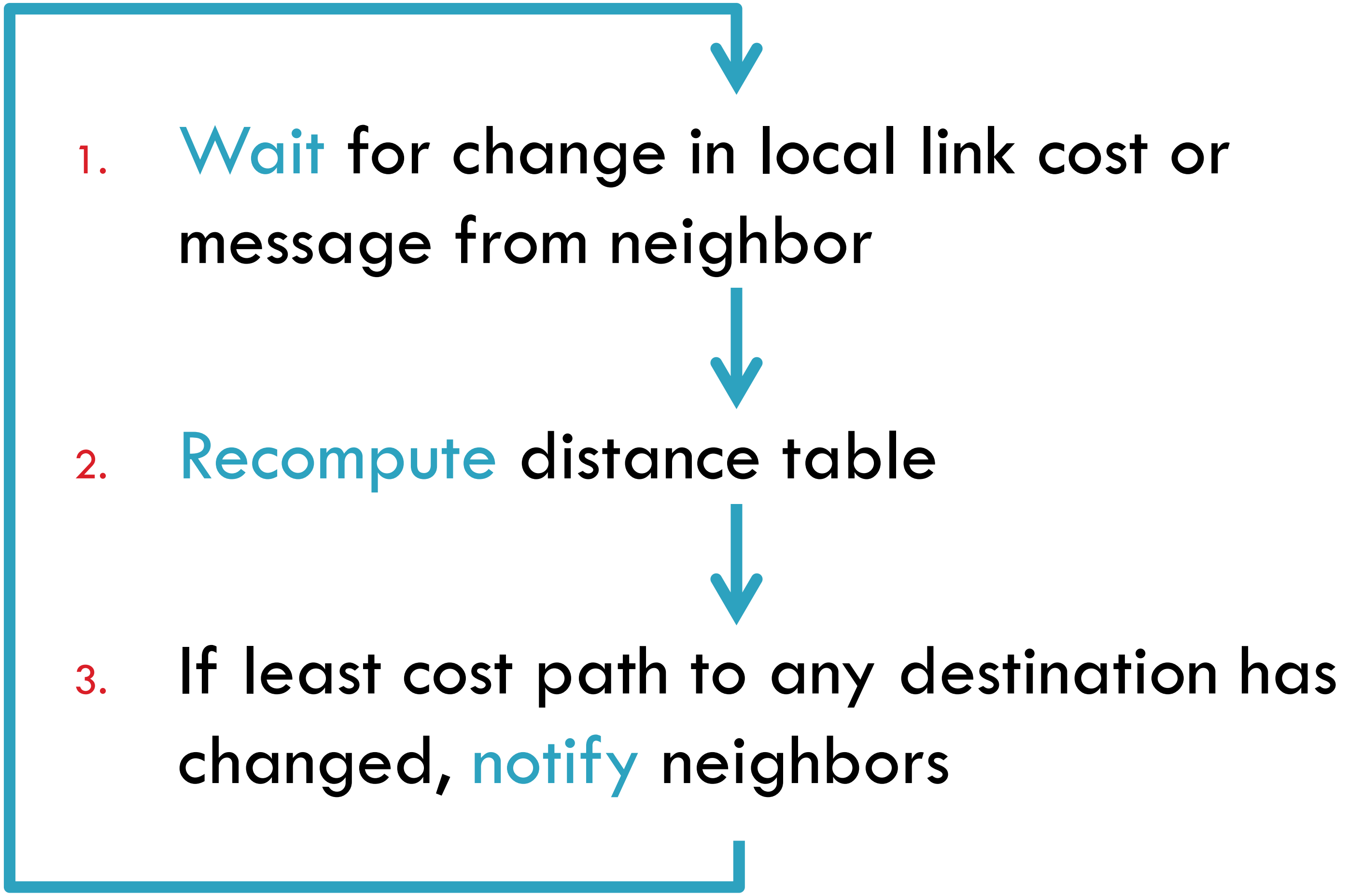
- What is a distance vector?
  - Current best known cost to reach a destination
- Idea: exchange vectors among neighbors to learn about lowest cost paths

DV Table at Node C

Destination	Cost
A	7
B	1
D	2
E	5
F	1

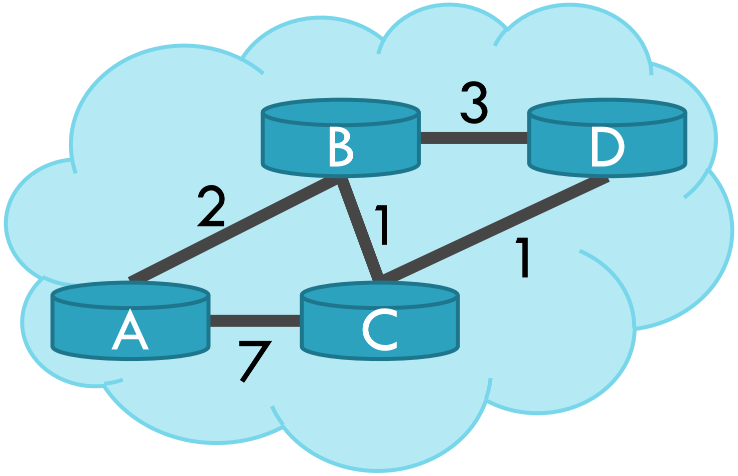
- No entry for C
- Initially, only has info for immediate neighbors
  - Other destinations cost = ∞
- Eventually, vector is filled
- Routing Information Protocol (RIP)

Distance Vector Routing Algorithm



1. Wait for change in local link cost or message from neighbor
2. Recompute distance table
3. If least cost path to any destination has changed, notify neighbors

Distance Vector Initialization



Node A

Dest.	**Cost**	**Next**
B	2	B
C	7	C
D	∞	

Node B

Dest.	**Cost**	**Next**
A	2	A
C	1	C
D	3	D

```
Initialization:
  for all neighbors V do
    if V adjacent to A
      $D(A, V) = c(A,V)$;
  else
      $D(A, V) = \infty$;
…
```

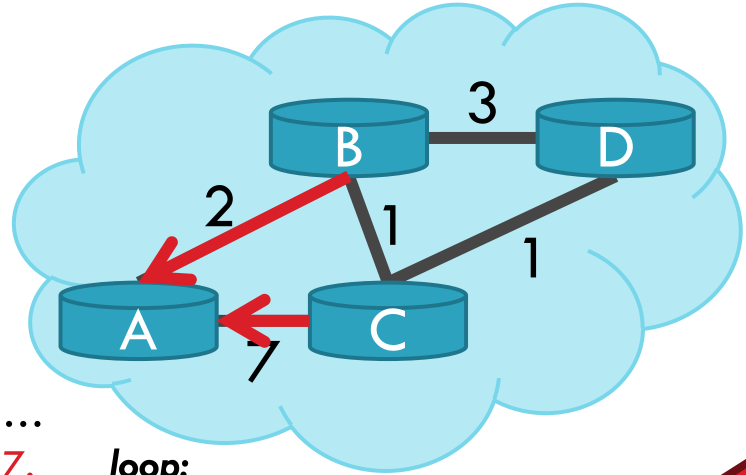
Node C

Dest.	**Cost**	**Next**
A	7	A
B	1	B
D	1	D

Node D

Dest.	**Cost**	**Next**
A	∞	
B	3	B
C	1	C

Distance Vector: 1st Iteration



```
…
loop:
…
else if (update D(V, Y) recei
  for all desti
    if (destina
      D(A,Y) =
    else
      D(A, Y) =
              mi
              D(A, V) + D(V, Y));
if (there is a new min. for dest. Y)
  send D(A, Y) to all neighbors
forever
```

Node A

Dest.	Cost	Next
B	2	B
C	3	B
D	5	B

Node B

Dest.	Cost	Next
A	2	A
C	1	C
D	2	C

B	1	B
D	1	D

e D

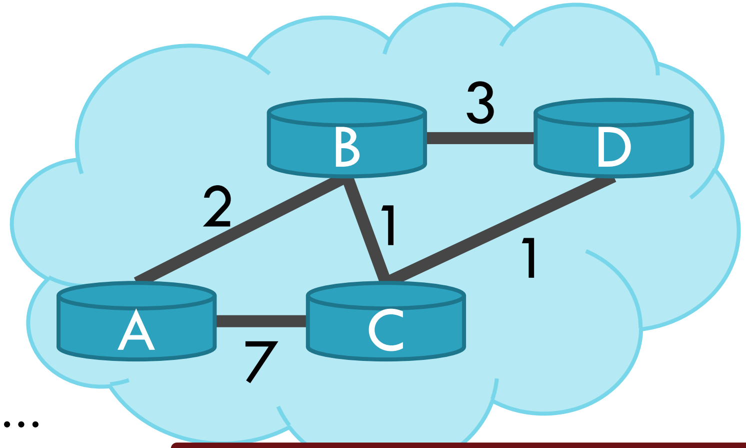
		Next
	4	B
B	3	B
C	1	C

D(A,C) A,C), D(A,B)+D(B,C))

D(A,D) = min(D(A,D), D(A,B)+D(B,D))

= min(8, 3 + 3 B = 5

Distance Vector: End of 3rd Iteration



Node A

Dest.	Cost	Next
B	2	B
C	3	B
D	4	B

Node B

Dest.	Cost	Next
A	2	A
C	1	C
D	2	C

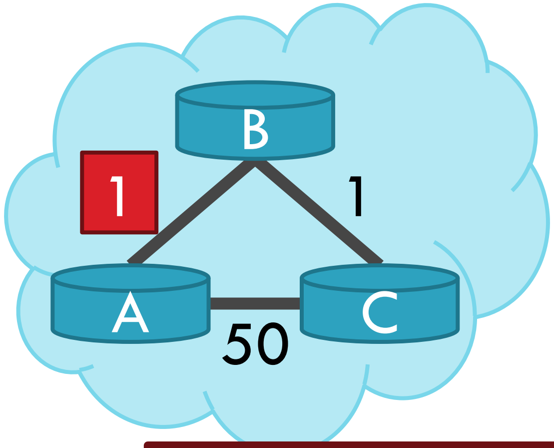
```
…
    loop
…
    else
      fo
        i

        else
          D(A, Y) =
                  min(D(A, Y),
                  D(A, V) + D(V, Y));
    if (there is a new min. for dest. Y)
      send D(A, Y) to all neighbors
    forever
```

- Nothing changes, algorithm terminates
- Until something changes…

A	3	B
B	1	B
D	1	D

		Next
A	4	C
B	2	C
C	1	C

```
loop:
  wait (link cost update or update message)
  if (c(A,V) changes by d)
    for all destinations Y through V do
      D(A,Y) = D(A,Y) + d
  else if (update D(V, Y) received from V)
    for all destinations Y do
      if (destination Y through V)
        D(A,Y) = D(A,V) + D(V, Y);
      else
        D(A, Y) = min(D(A, Y), D(A, V) + D(V, Y));
```



Link Co

Algor

Good news travels fast

Algorithm

erminates

Node B

D	C	N
A	4	A
C	1	B

	C	N
A	1	A
C	1	B

D	C	N
A	1	A
C	1	B

D	C	N
A	1	A
C	1	B

Node C

D	C	N
A	5	B
B	1	B

D	C	N
A	5	B
B	1	B

D	C	N
A	2	B
B	1	B

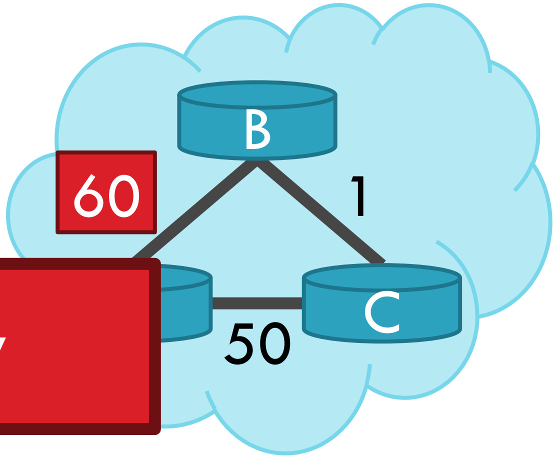
D	C	N
A	2	B
B	1	B

Time

Count to Infinity Problem

- Node B knows D(C, A) = 5
- However, B does not know the path is C
- Thus, D(B,A

Bad news travels slowly



Node B

D	C	N
A	4	A
C	1	B

D		N
A	6	C
C	1	B

D	C	N
A	6	C
C	1	B

D	C	N
A	8	C
C	1	B

Node C

D	C	N
A	5	B
B	1	B

D	C	N
A	5	B
B	1	B

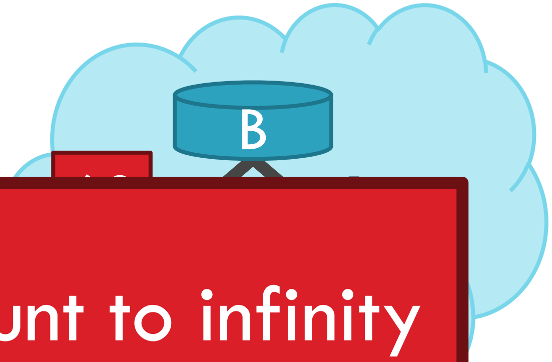
D	C	N
A	7	B
B	1	B

D	C	N
A	7	B
B	1	B

Time

Poisoned Reverse

- If C routes through B to get to A



Does this completely solve this count to infinity problem?

NO

Multipath loops can still trigger the issue

Node C	A	5	B	A	5	B	A	**50**	**A**	A	50	A
	B	1	B	B	1	B	B	1	B	B	1	B

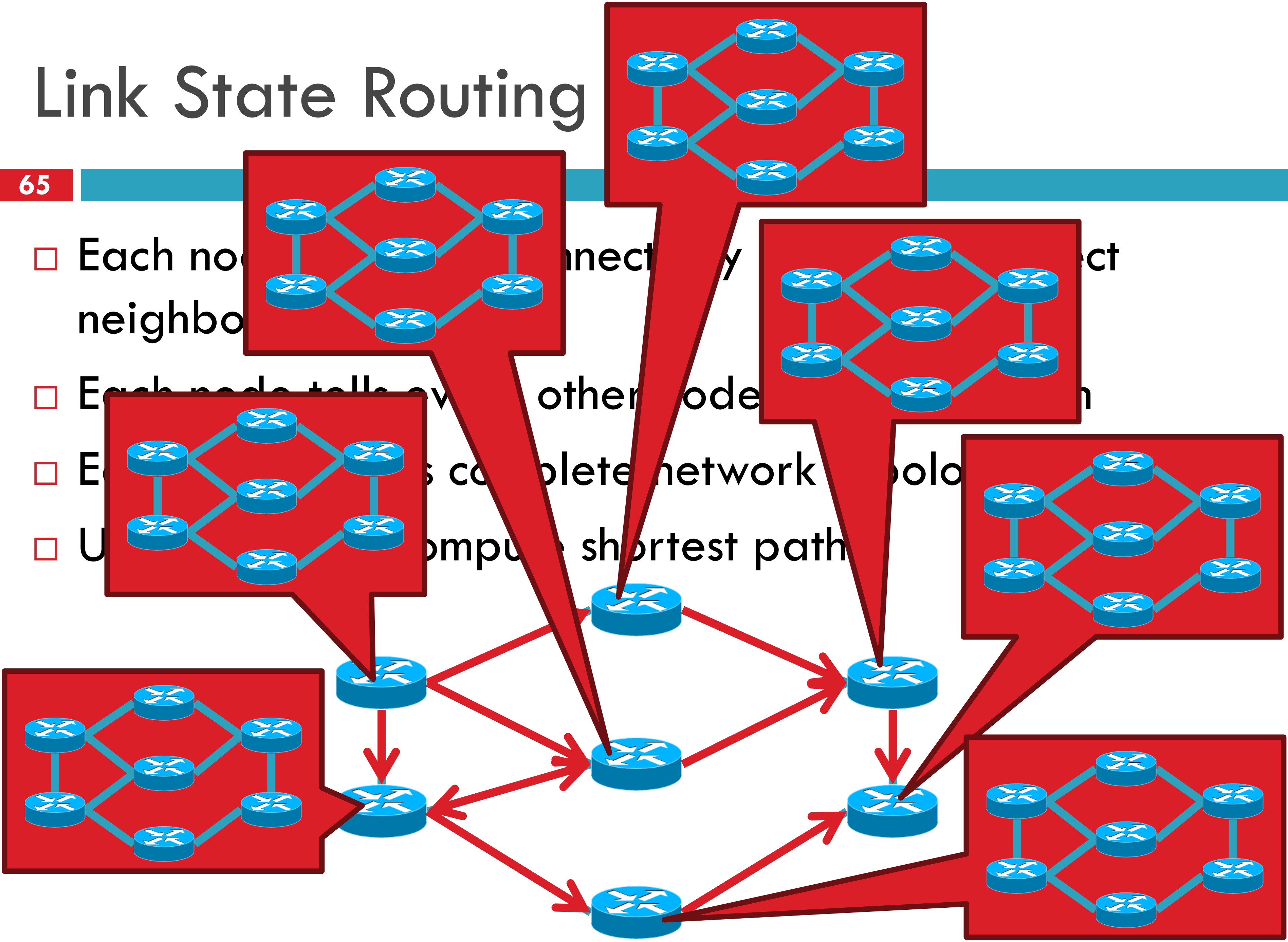
Time

Outline

- Distance Vector Routing
  - RIP
- Link State Routing
  - OSPF
  - IS-IS

Link State Routing

- Each no nnect y ect neighbo
- Each node tells ev other ode n
- E s co plete network polo
- U ompu e sh ortest path

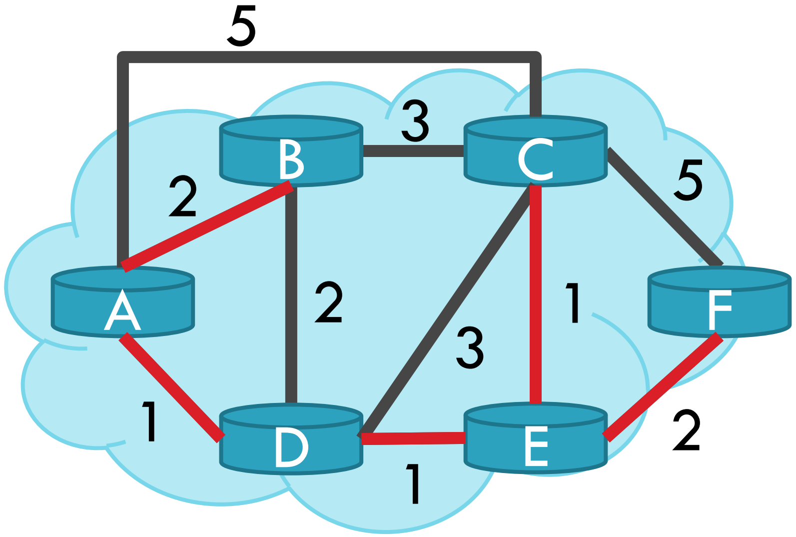


Flooding Details

- Each node periodically generates Link State Packet
  - ID of node generating the LSP
  - List of direct neighbors and costs
  - Sequence number (64-bit, assumed to never wrap)
  - Time to live
- Flood is reliable (ack + retransmission)
- Sequence number “versions” each LSP
- Receivers flood LSPs to their own neighbors
  - Except whoever originated the LSP
- LSPs also generated when link states change

Dijkstra’s Algorithm

Step	Start S	→B	→C	→D	→E	→F
0	A	2, A	5, A	1, A	∞	∞



```
…
8.  Loop
9.    find w not in S s.t. D(w) is a minimum;
10.   add w to S;
11.   update D(v) for all v adjacent
      to w and not in S:
12.     D(v) = min( D(v), D(w) + c(w,v) );
13. until all nodes in S;
```

```
1. Initialization:
2.   S = {A};
3.   for all nodes v
4.     if v adjacent to A
5.       then D(v) = c(A,v);
6.       else D(v) = ∞;
```

OSPF vs. IS-IS

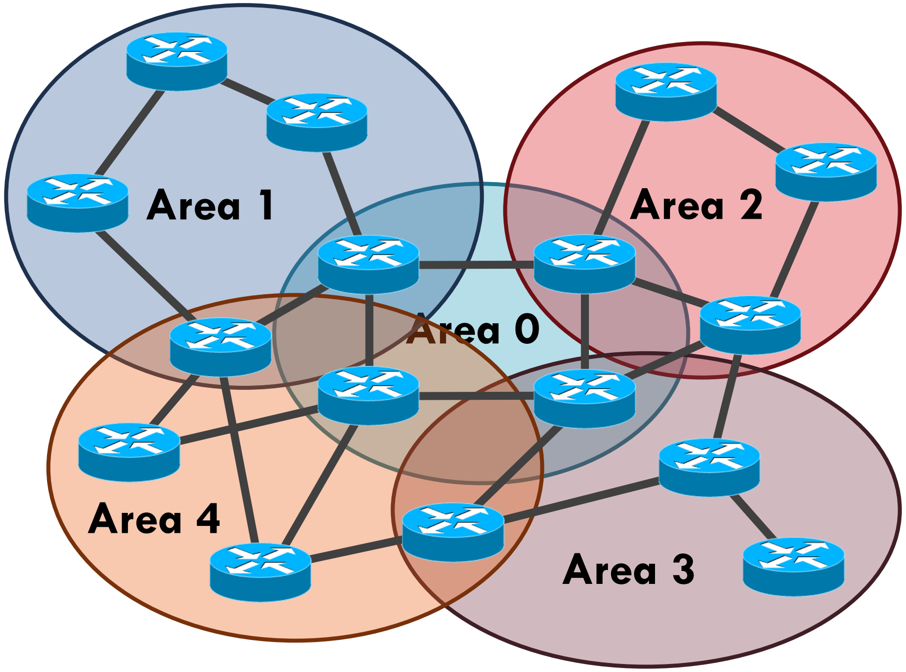
- Two different implementations of link-state routing

OSPF	IS-IS
Favored by companies, datacenters	Favored by ISPs
More optional features	Less "chatty" - Less network overhead - Supports more devices
Built on top of IPv4 - LSAs are sent via IPv4 - OSPFv3 needed for IPv6	Not tied to IP - Works with IPv4 or IPv6

Different Organizational Structure

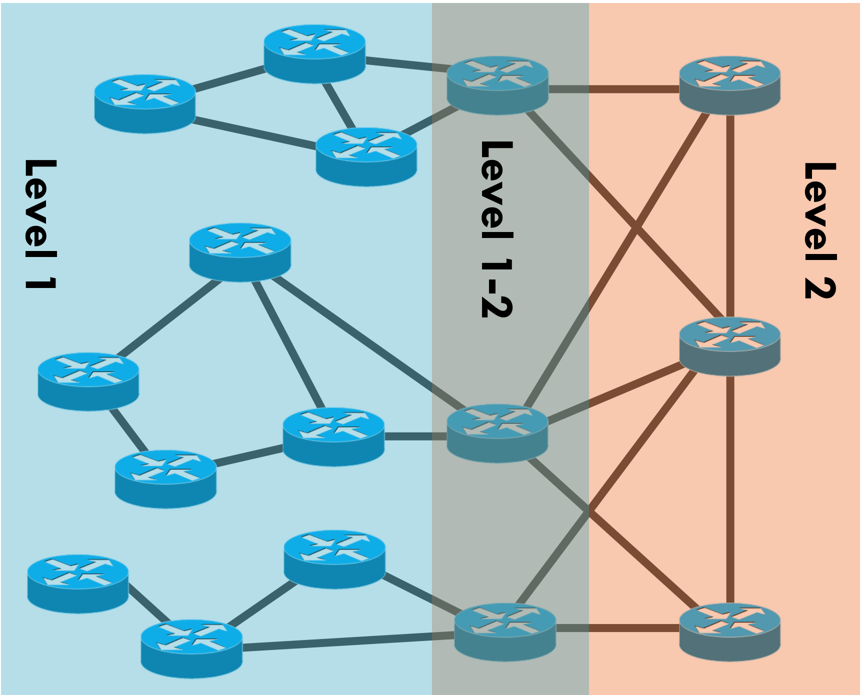
OSPF

- Organized around overlapping areas
- Area 0 is the core network



IS-IS

- Organized as a 2-level hierarchy
- Level 2 is the backbone



Link State vs. Distance Vector

	Link State	**Distance Vector**
Message Complexity	$O(n^2*e)$	O(d*n*k)
Time Complexity	O(n*log n)	O(n)
Convergence Time	O(1)	O(k)
Robustness	• Nodes may advertise incorrect link costs • Each node computes their own table	• Nodes may advertise incorrect path cost • Errors propagate due to sharing of DV tables

- Which is best?
- In practice, it depends.
- In general, link state is more popular.