# CS 43: Computer Networks

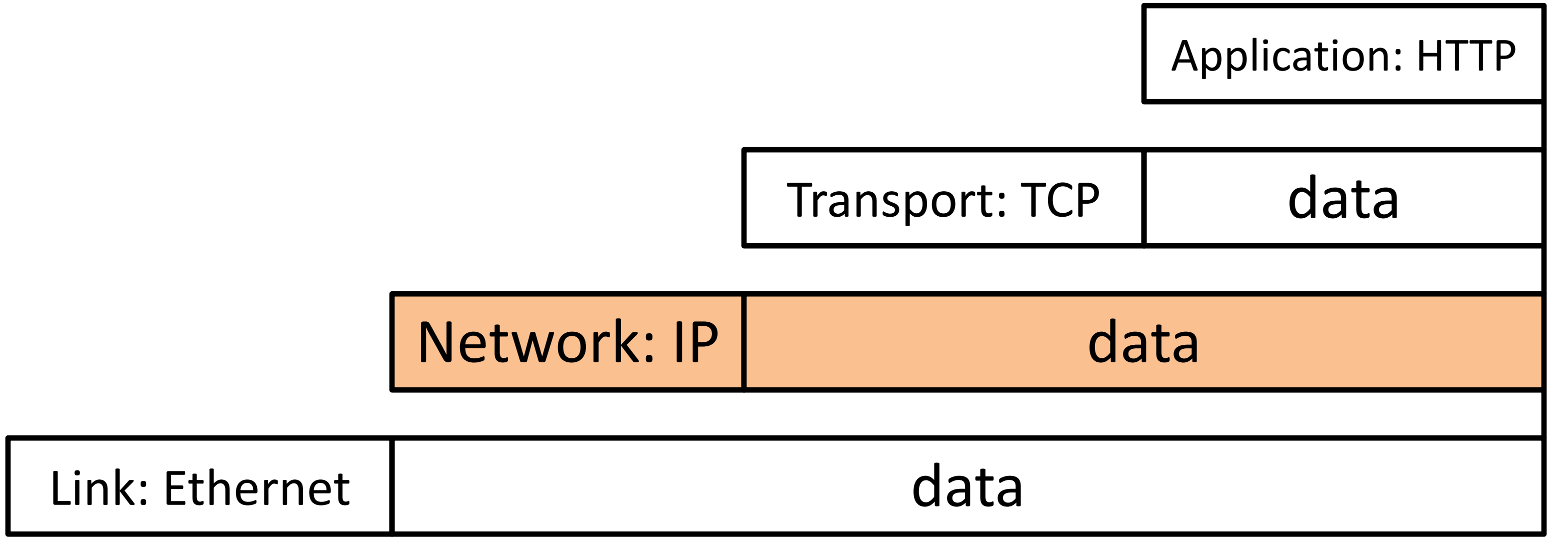
21-22: Routing on the Internet, Traffic Management
November 26, Dec 3 2024

*Adapted from Slides by: Kurose & Ross, D. Choffnes, K. Webb, J. Rexford*

SWARTHMORE COLLEGE

# Network Layer

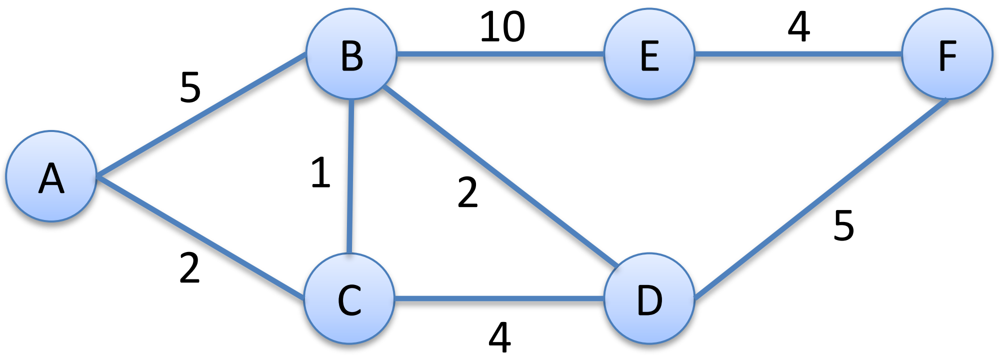
- Function: Route packets end-to-end on a network, through multiple hops



# Network Layer Functions

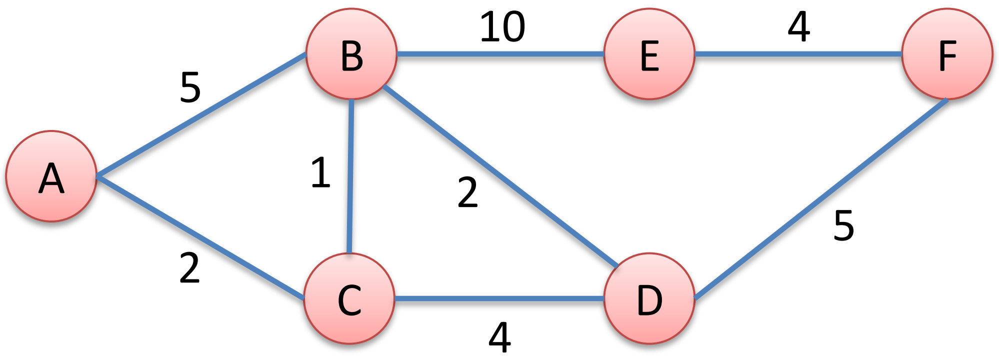
- Forwarding: move packets from router's input to appropriate router output
  - Look up in a table
- Routing: determine route taken by packets from source to destination.
  - Populating the table

# Dijkstra's Algorithm Example




- Goal: From the perspective of node A:
  - Determine shortest path to every destination

# Dijkstra's Algorithm – Done!




Lot more state
in routing table!

Final Answer

| Dest | Path | Cost D(v) |
|---|---|---|
| A | A | 0 |
| B | C, B | 3 |
| C | C | 2 |
| D | C, B, D | 5 |
| E | C, B, E | 13 |
| F | C, B, D, F | 10 |

Populate Forwarding Table

Forwarding Table

| Dest | Forward To |
|---|---|
| B | C |
| C | C |
| D | C |
| E | C |
| F | C |

# Intra-AS Routing

- Also known as *interior gateway protocols (IGP)*

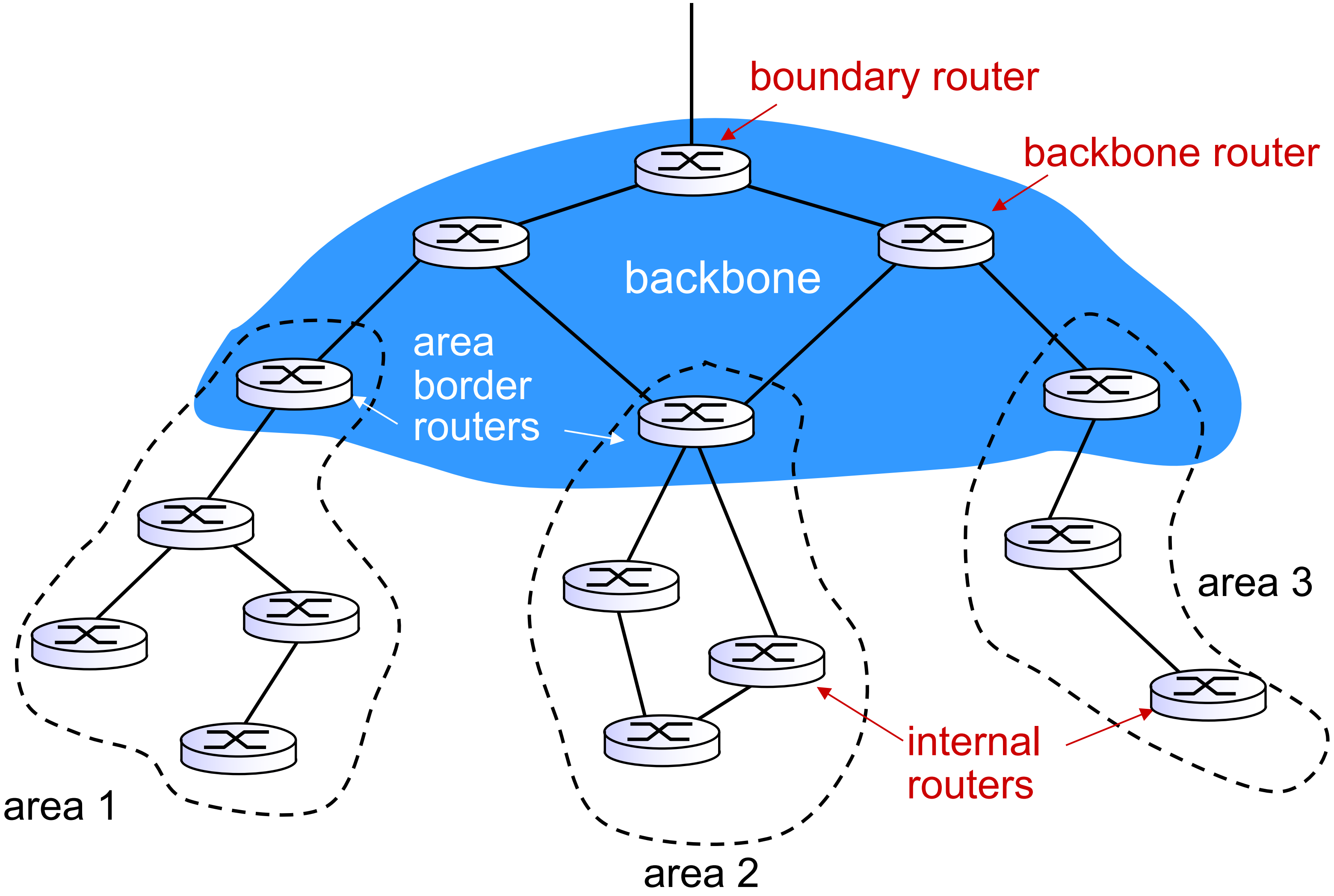
Goal:
Get traffic that is already in an AS to a destination inside that same AS.

*OSPF and IS-IS are deployed most commonly today*

# OSPF (Open Shortest Path First)

- Link state protocol (reliable flooding of LSAs)
- “Open”: standardized, publicly available implementations
- Multiple equal-cost paths allowed (load balancing)
- Additional features:
  – OSPF messages authenticated (to prevent malicious intrusion)
  – Hierarchical OSPF for large autonomous systems.

# Hierarchical OSPF

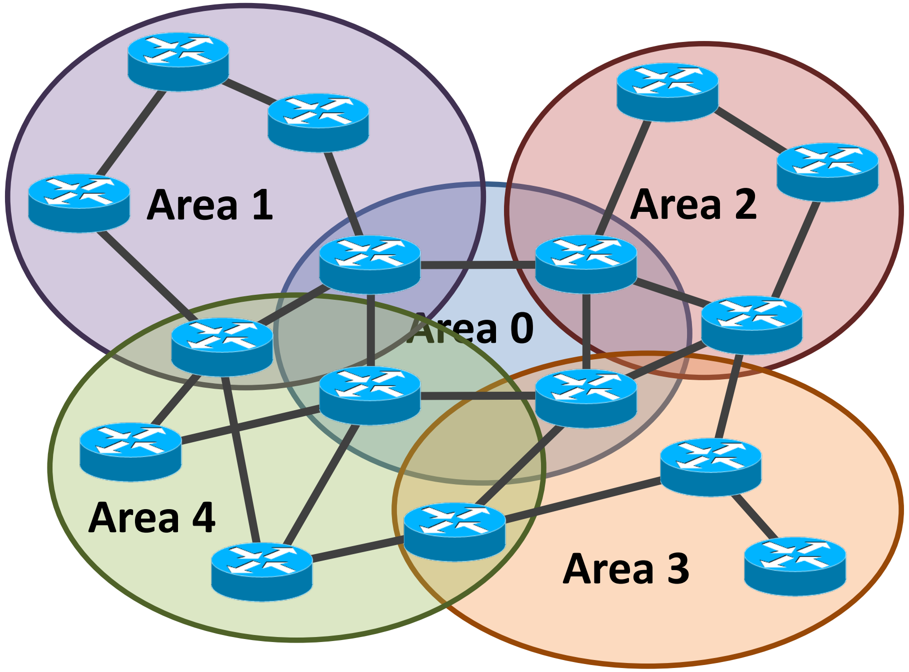


# Different Organizational Structure

| OSPF | IS-IS |
|---|---|

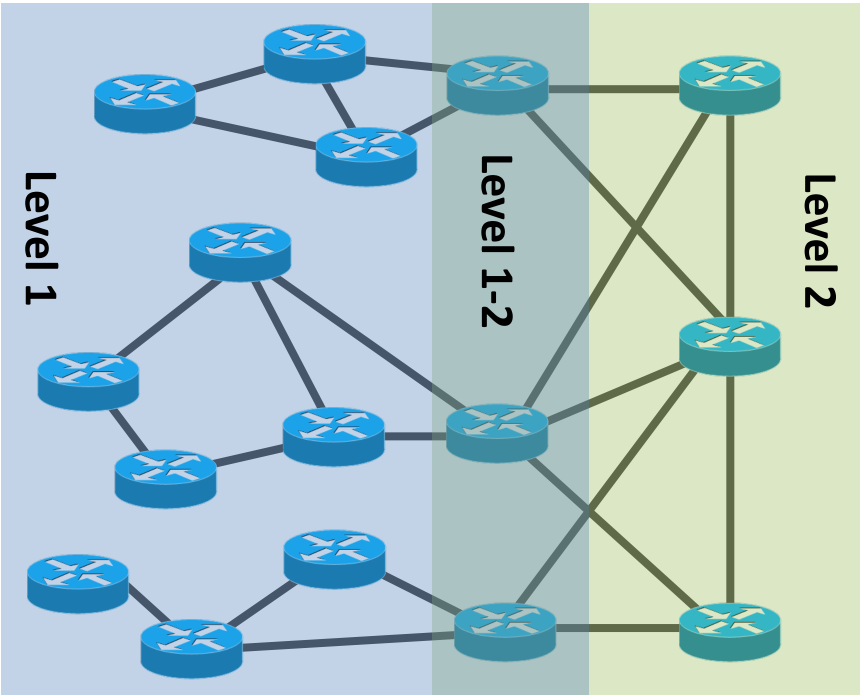
**OSPF**

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




**IS-IS**

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



# Real Protocols: OSPF vs. IS-IS

- Two different implementations of link-state routing

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

# Distance Vector Algorithm

*Iterative, asynchronous:*

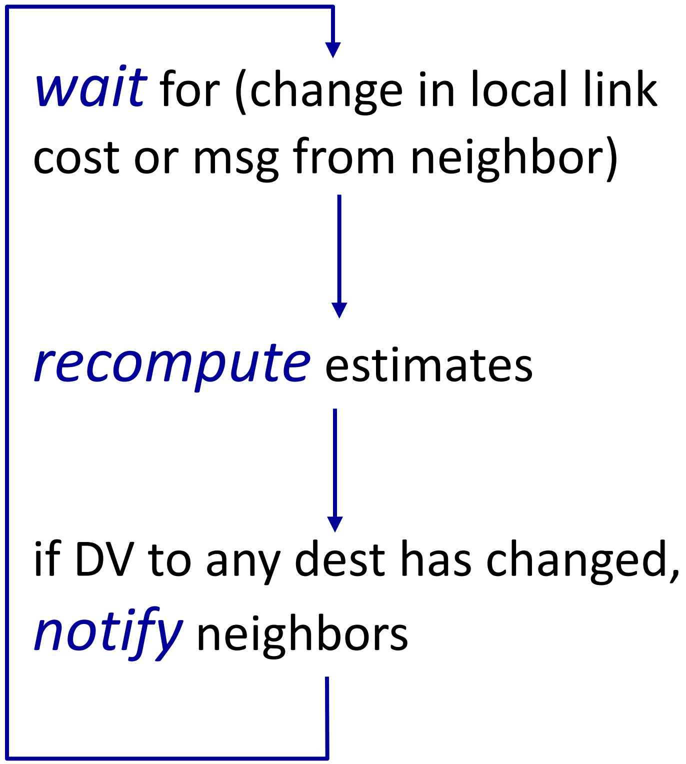
Iteration when:

- Local link cost change
- DV update from neighbor
- Periodic timer

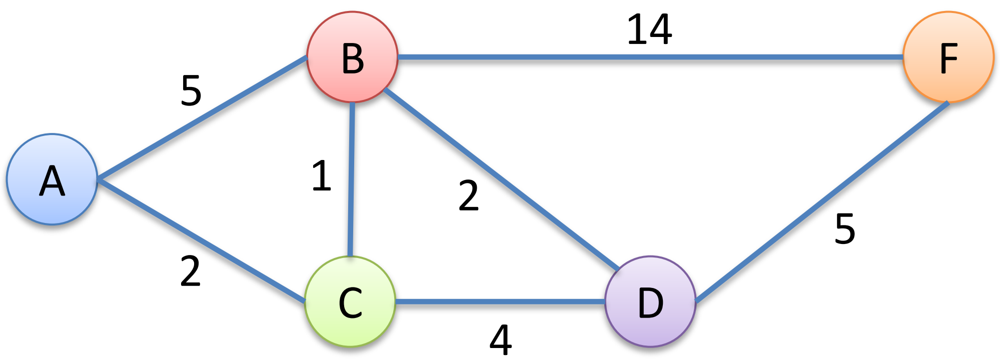
*Distributed:*

- Each node knows only a portion of global link info

*each node:*

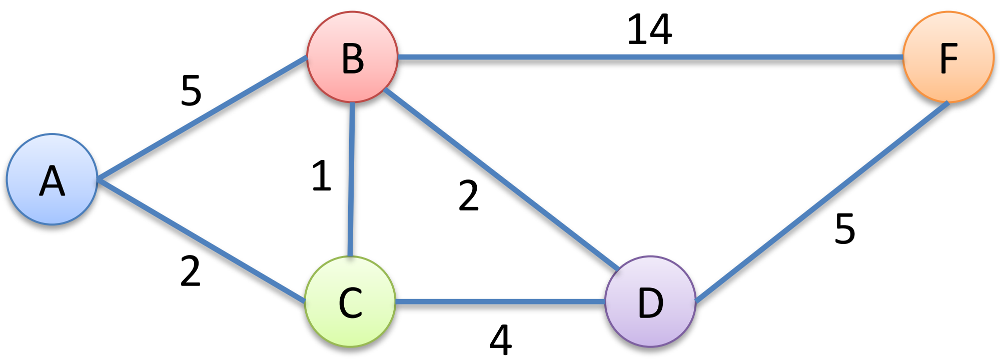


# Distance Vector Example




- Same network as Dijkstra's example, without node E.
- What I'll show you next is routing table (of distance vectors) at each router.

# Distance Vector – Round 0




Routers populate their forwarding table by taking the row minimum.

Router F

| Via→ ↓ To | B | D |
|---|---|---|
| A | | |
| B | 14 | |
| C | | |
| D | | 5 |

Router A

| Via→ ↓ To | B | C |
|---|---|---|
| B | 5 | |
| C | | 2 |
| D | | |
| F | | |

Router B

| Via→ ↓ To | A | C | D | F |
|---|---|---|---|---|
| A | 5 | | | |
| C | | 1 | | |
| D | | | 2 | |
| F | | | | 14 |

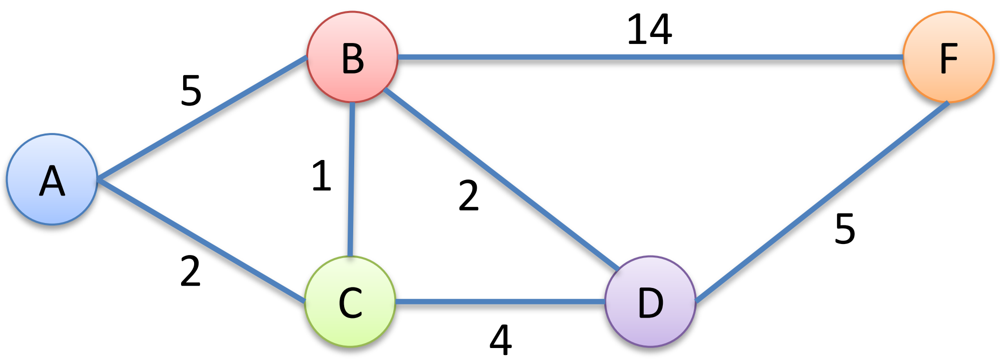
Router C

| Via→ ↓ To | A | B | D |
|---|---|---|---|
| A | 2 | | |
| B | | 1 | |
| D | | | 4 |
| F | | | |

Router D

| Via→ ↓ To | B | C | F |
|---|---|---|---|
| A | | | |
| B | 2 | | |
| C | | 4 | |
| F | | | 5 |

# Distance Vector – Convergence




Eventually, we reach a converged state.

Router F

| Via→ ↓ To | B | D |
|---|---|---|
| A | 17 | 10 |
| B | 14 | 7 |
| C | 15 | 8 |
| D | 16 | 5 |

Router A

| Via→ ↓ To | B | C |
|---|---|---|
| B | 5 | 3 |
| C | 6 | 2 |
| D | 7 | 5 |
| F | 12 | 10 |

Router B

| Via→ ↓ To | A | C | D | F |
|---|---|---|---|---|
| A | 5 | 3 | 7 | 24 |
| C | 7 | 1 | 4 | 22 |
| D | 10 | 4 | 2 | 19 |
| F | 15 | 9 | 7 | 14 |

Router C

| Via→ ↓ To | A | B | D |
|---|---|---|---|
| A | 2 | 4 | 9 |
| B | 7 | 1 | 6 |
| D | 7 | 3 | 4 |
| F | 12 | 8 | 9 |

Router D

| Via→ ↓ To | B | C | F |
|---|---|---|---|
| A | 5 | 6 | 15 |
| B | 2 | 5 | 12 |
| C | 3 | 4 | 13 |
| F | 9 | 12 | 5 |

# Why do we need different Intra and Interdomain AS routing ?

A. Scalability
B. Performance
C. A and B
D. More than just A and B

# Why do we need different Intra and Interdomain AS routing ?

*Policy:*

- inter-AS: admin wants control over how its traffic routed, who routes through its net.
- intra-AS: single admin, so no policy decisions needed

*Scale:*

- hierarchical routing saves table size, reduced update traffic

*Performance:*

- intra-AS: can focus on performance
- inter-AS: policy may dominate over performance

# Internet/inter-AS Routing

Goal:
Get traffic from one AS to another.

## The Inter-domain routing protocol, needs to be an agreed upon protocol across all Autonomous Systems

A. Yes, for interoperability

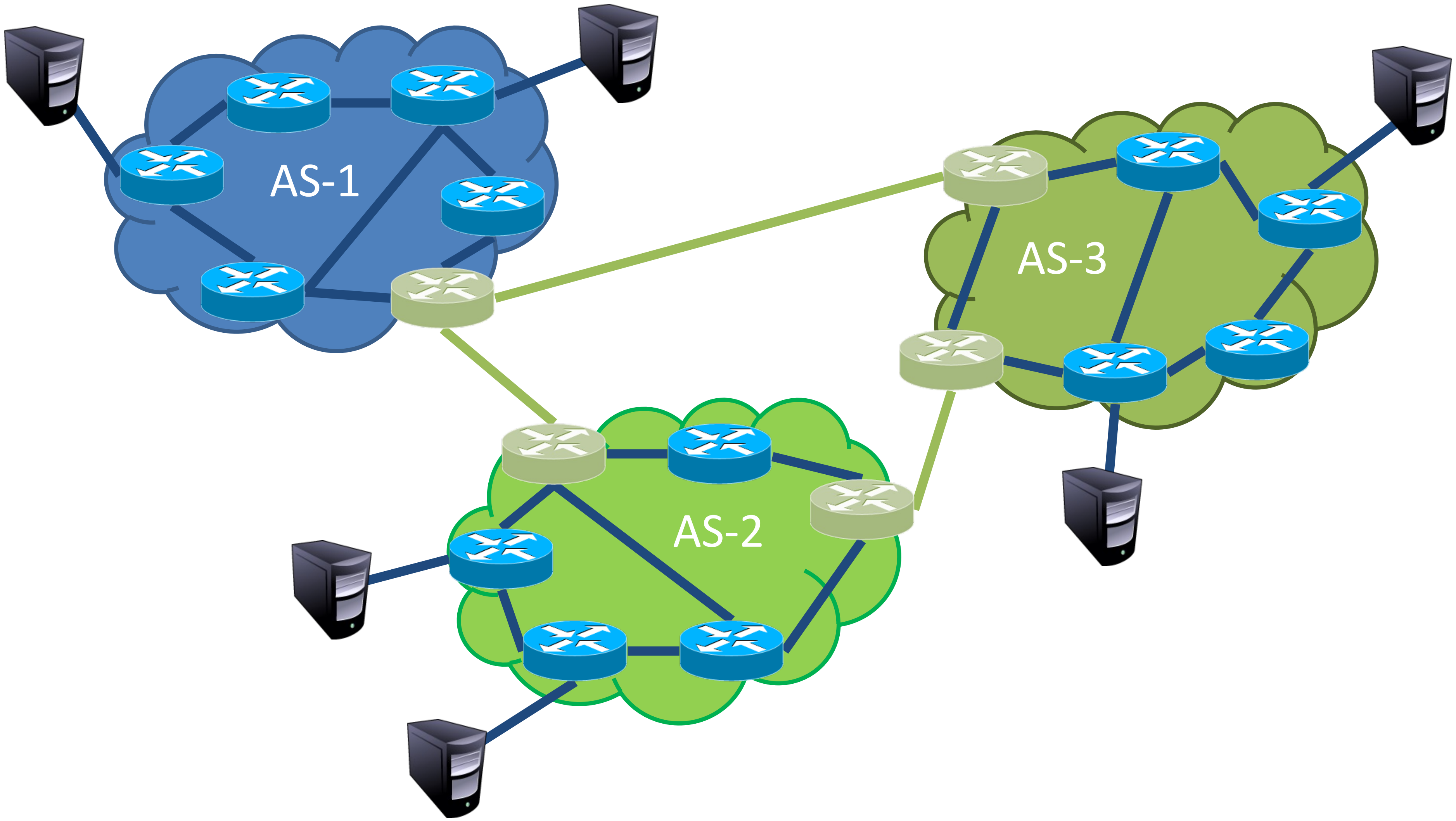
B. Not necessarily, but reduces overhead

C. No, each AS can have its own inter-domain routing protocol of choice.

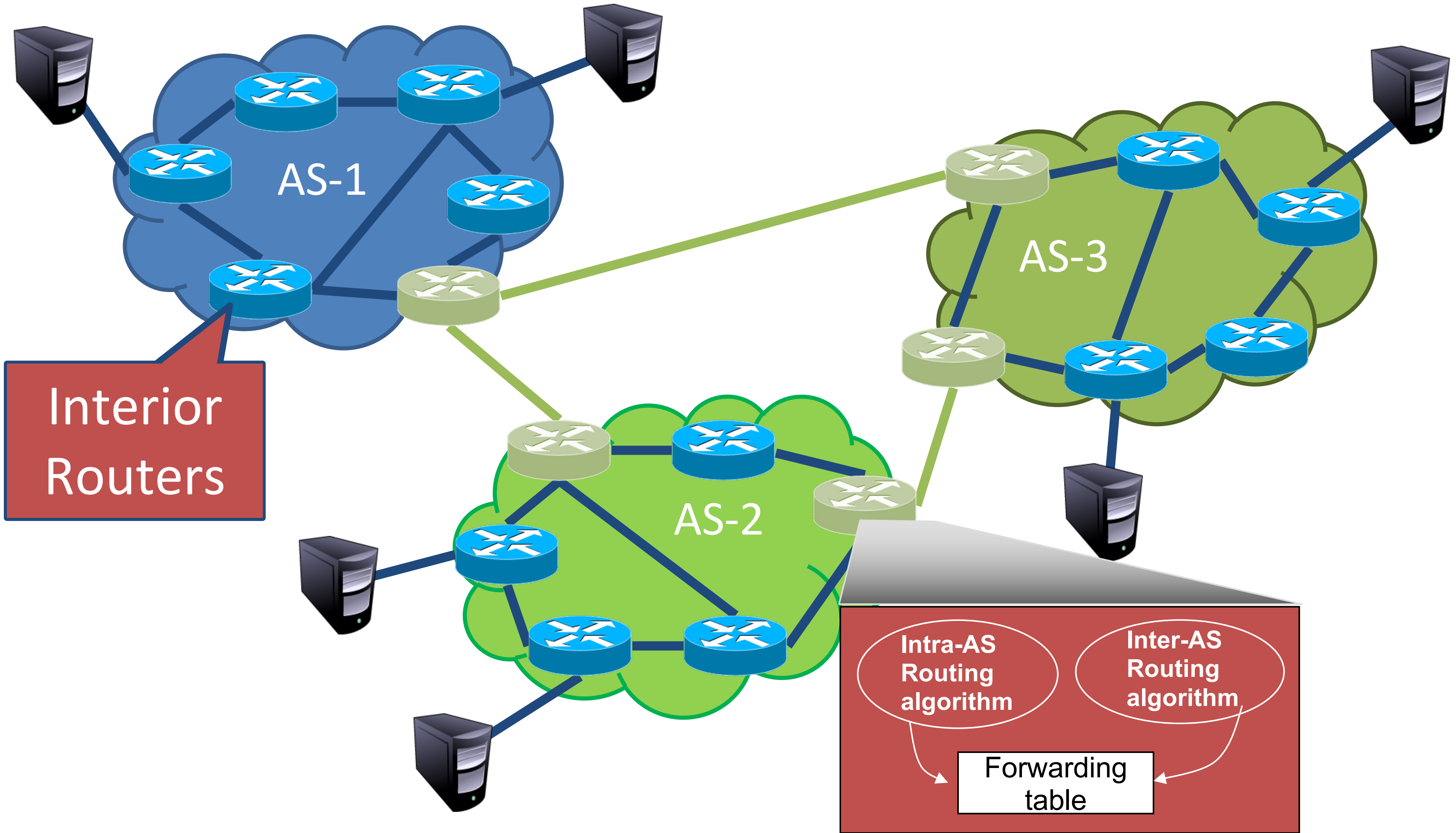
# The Inter-domain routing protocol, needs to be an agreed upon protocol across all Autonomous Systems

- Global connectivity is at stake!
  - Thus, all ASs must use the same protocol
  - Contrast with intra-domain routing
- What are the requirements?
  - Scalability
  - Flexibility in choosing routes
- Question: link state or distance vector?
  - Trick question: BGP is a path vector protocol

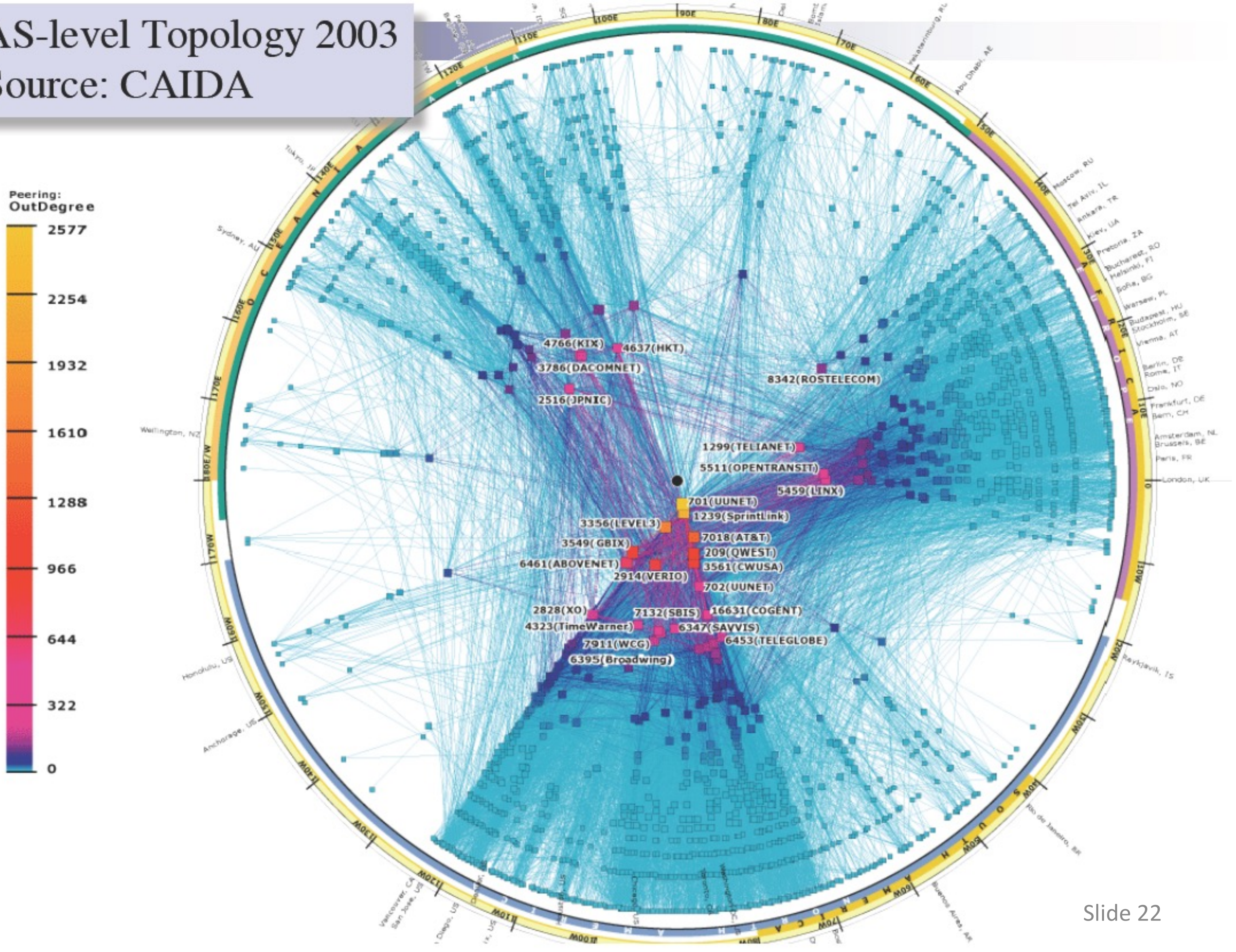
# Hierarchical routing: Autonomous Systems



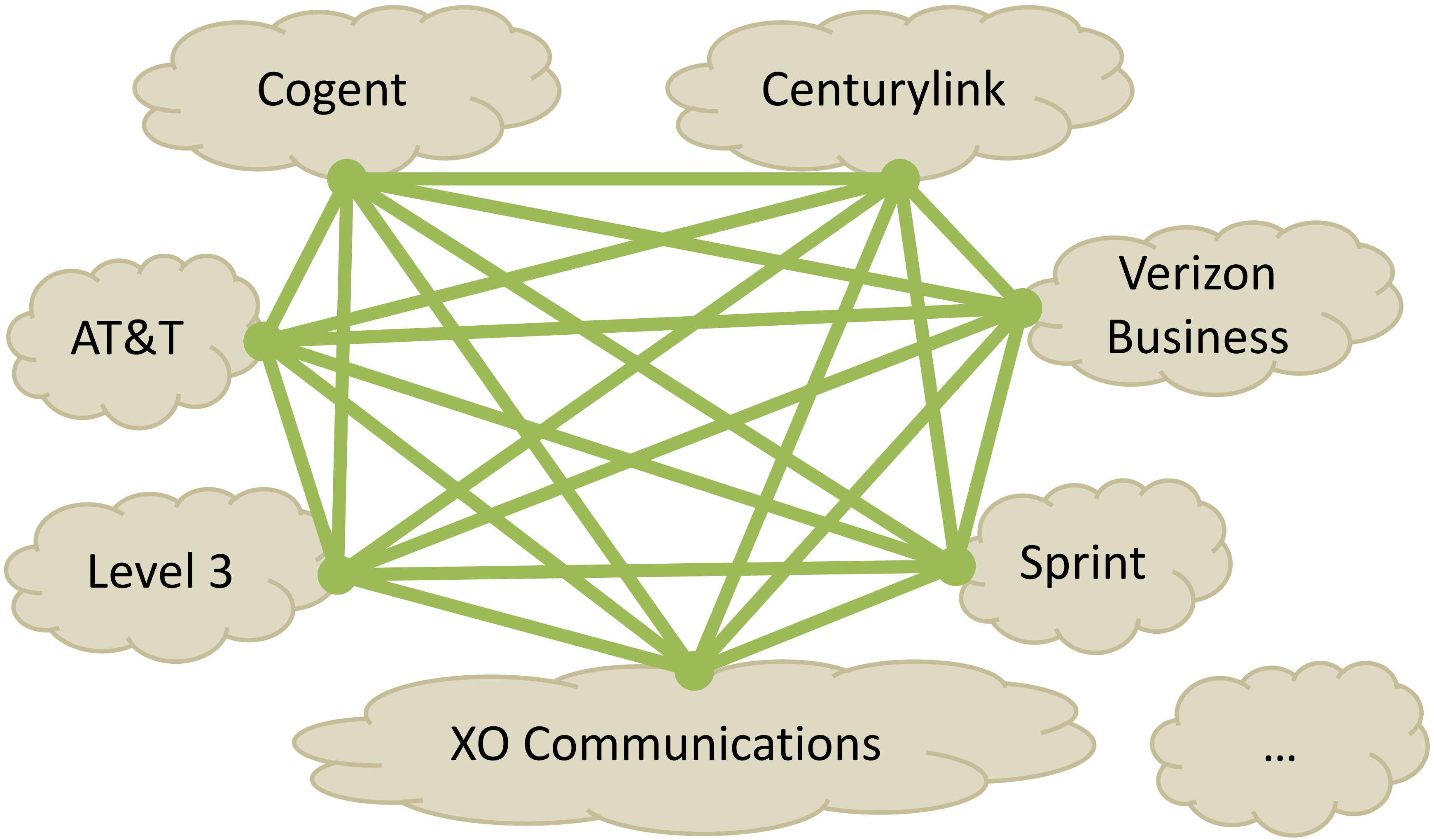
# Hierarchical routing: Interconnected ASes



# AS-level Topology 2003
Source: CAIDA

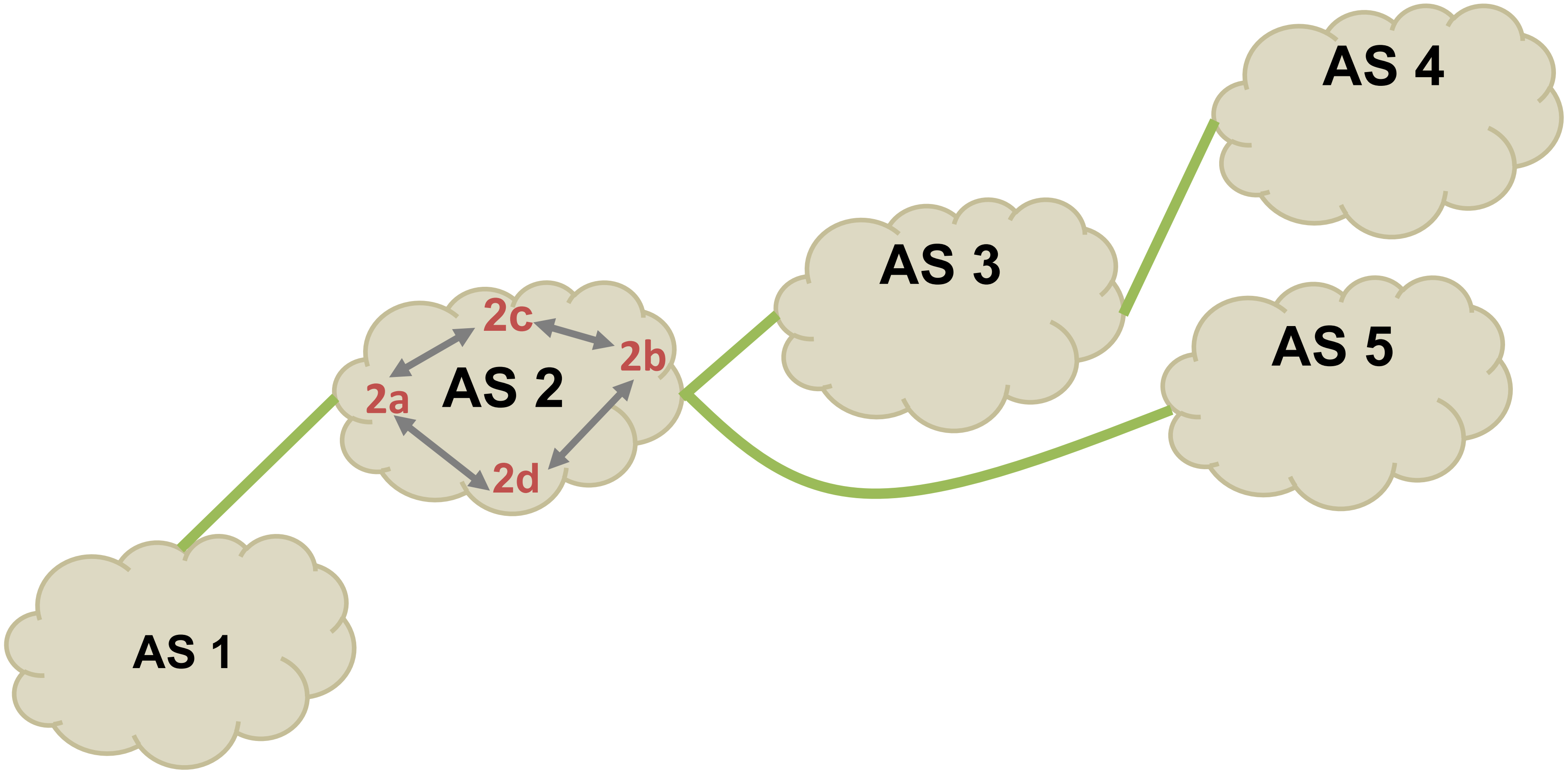


# Tier-1 ISP Peering



# Path Vector Protocol

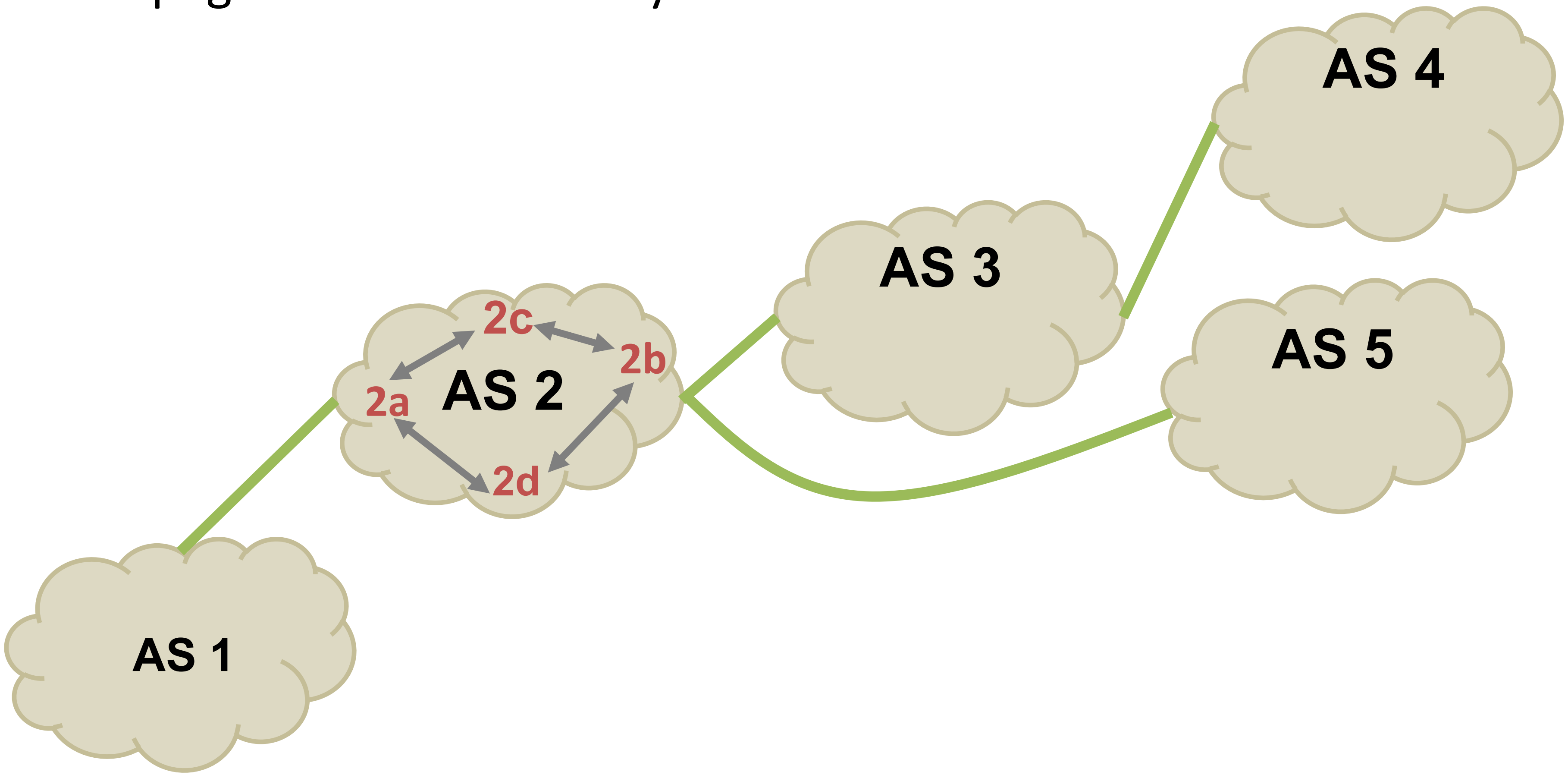
- Key idea: advertise the entire path
  - Distance vector: send *distance metric* per dest d
  - Path vector: send the *entire path* for each dest d



# Inter-domain (Inter-ISP) Routing

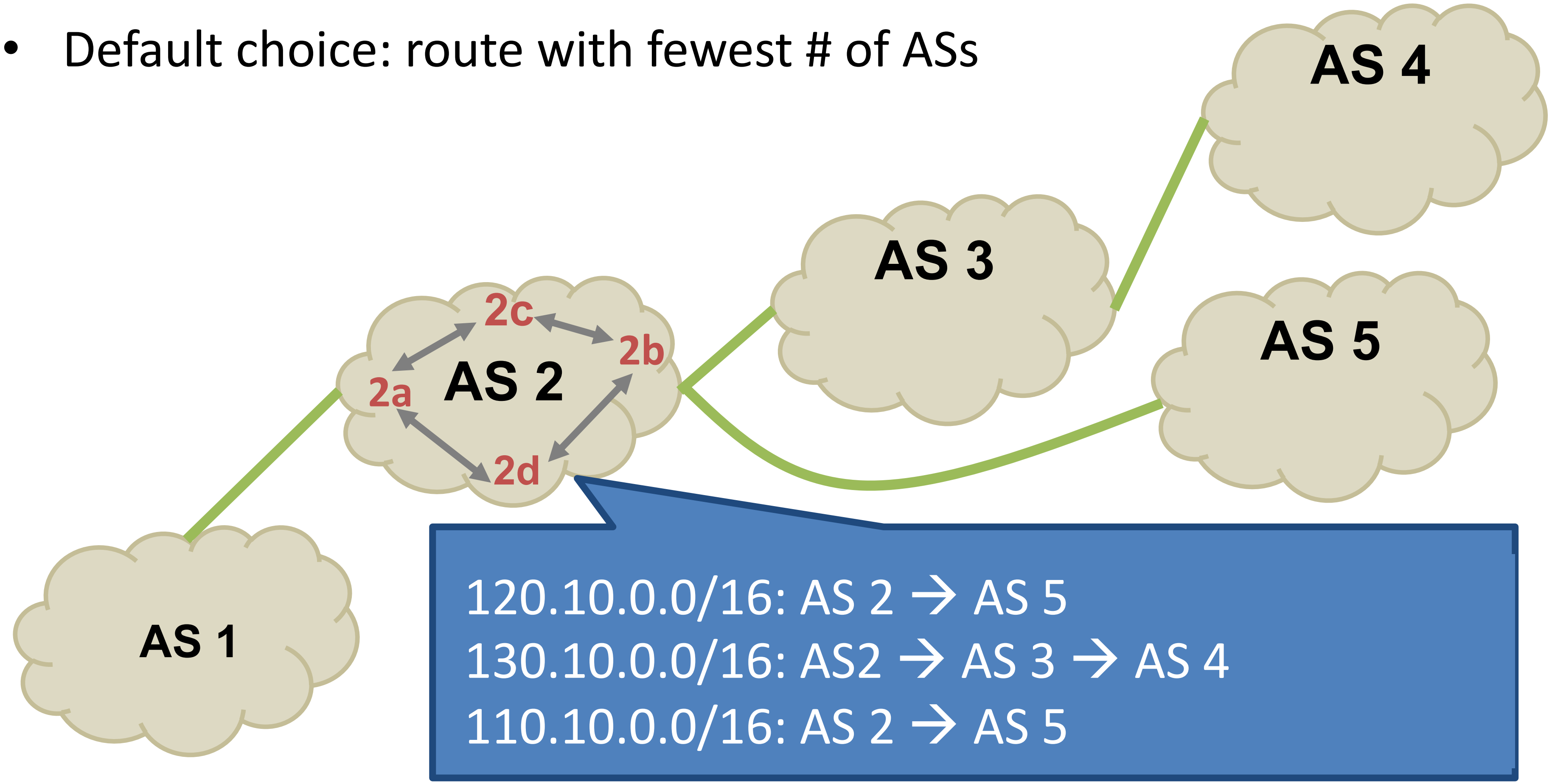
*AS2 must:*

1. Learn destinations reachable through AS2
2. Propagate this reachability info to all routers in AS2



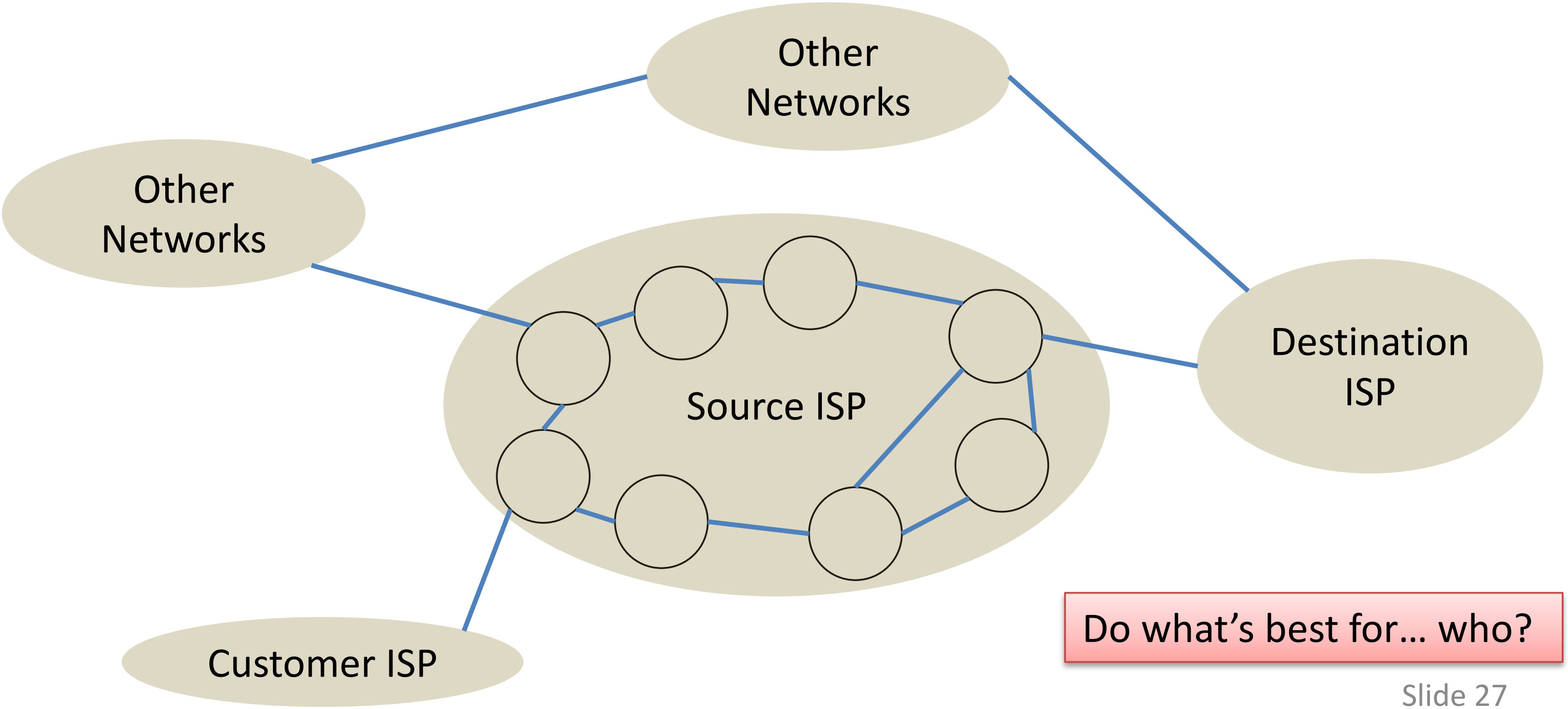
# Path Vector Protocol

- AS-path: sequence of ASs a route traverses
  - Like distance vector, plus additional information
- Used for loop detection and to apply policy
- Default choice: route with fewest # of ASs



# Routing Policy

- How should the ISP route the customer's traffic to the destination?




Do what's best for… who?

# Which routes a BGP router advertises will depend on…

A. which ISPs have contractual agreements.

B. the shortest path to a subnet/prefix.

C. which subnets are customers of an ISP.

D. More than one of the above. (which?)

# Which routes a BGP router advertises will depend on…

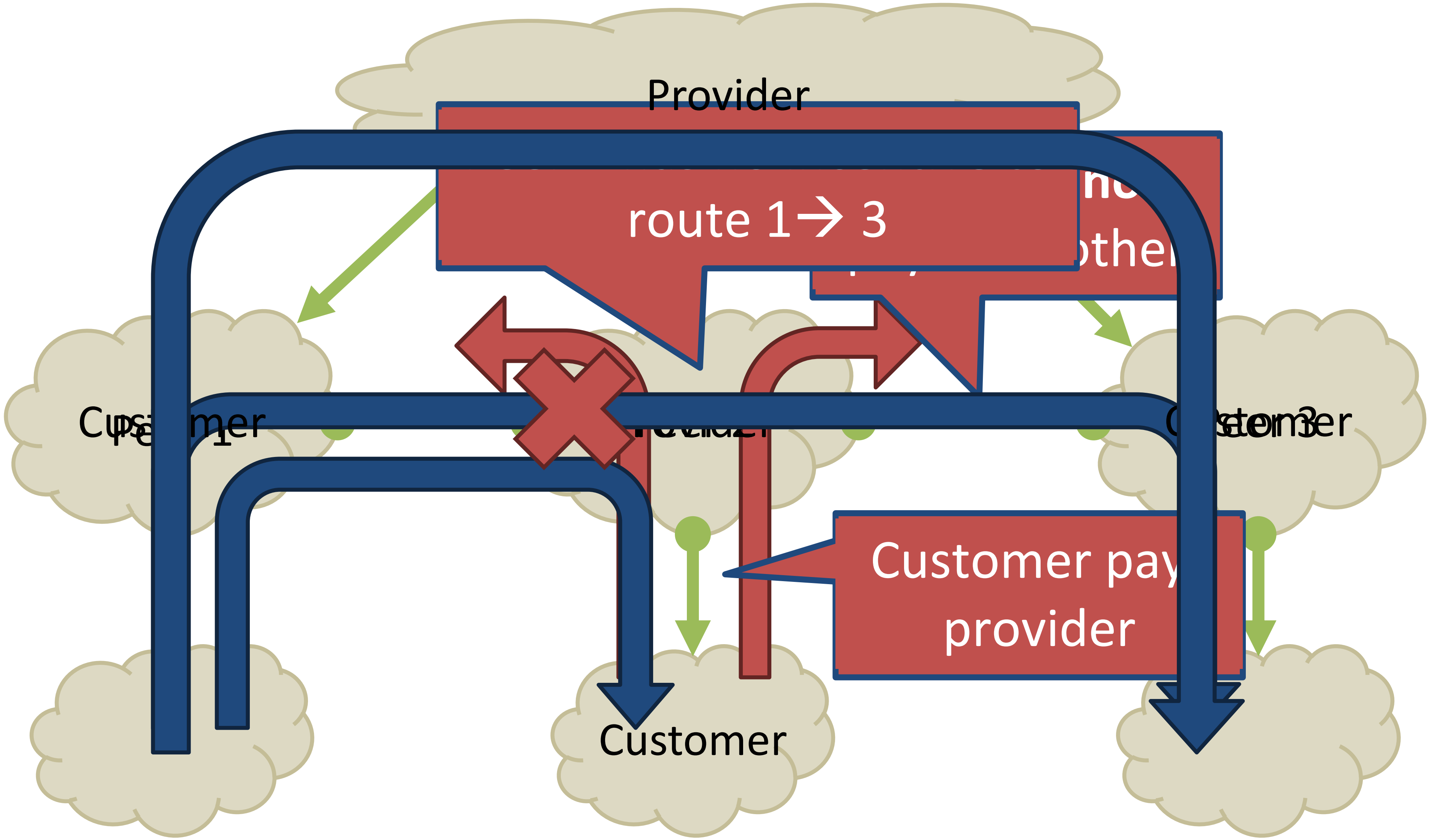
A. which ISPs have contractual agreements.

B. the shortest path to a subnet/prefix.

C. which subnets are customers of an ISP.

D. More than one of the above. (which?)

# BGP Relationships



# Peering/Interconnection Wars

- Peer

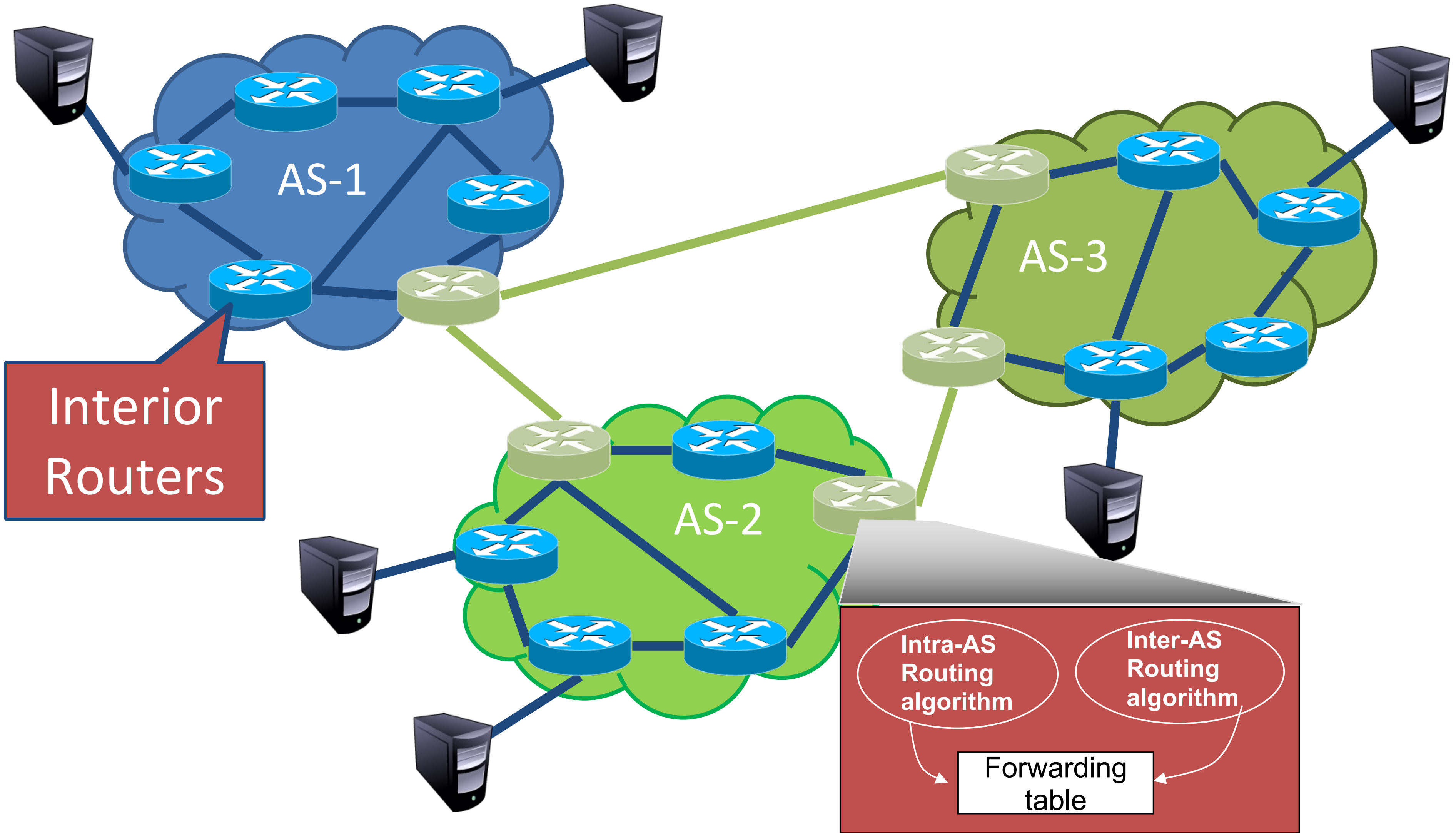
- Reduce upstream costs
- Improve end-to-end performance
- May be the only way to connect to parts of the Internet

- Don't Peer

- You would rather have customers
- Peers are often competitors
- Peering agreements require periodic renegotiation

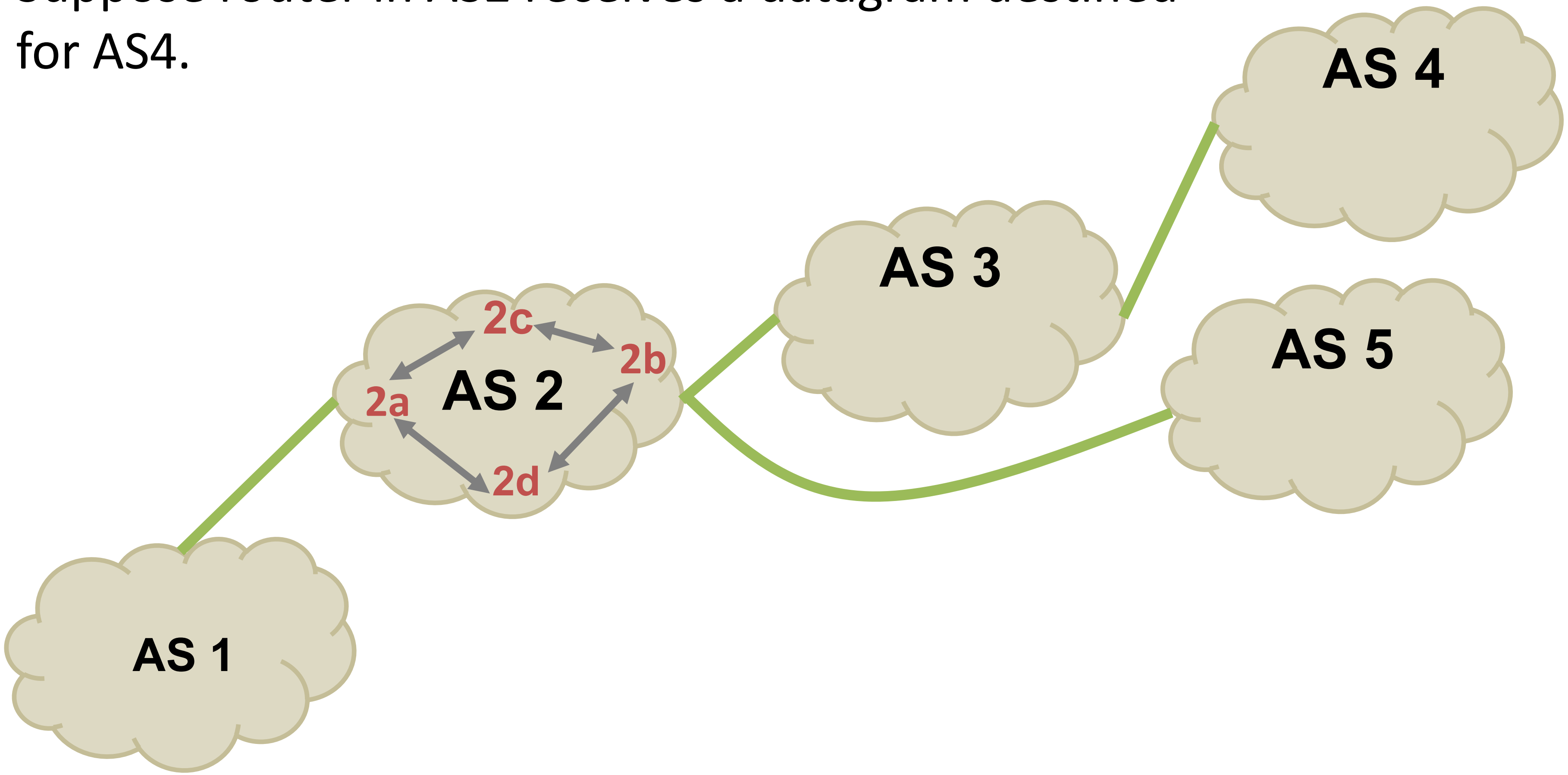
Peering struggles in the ISP world are extremely contentious, agreements are usually confidential

# Hierarchical routing: Interconnected ASes



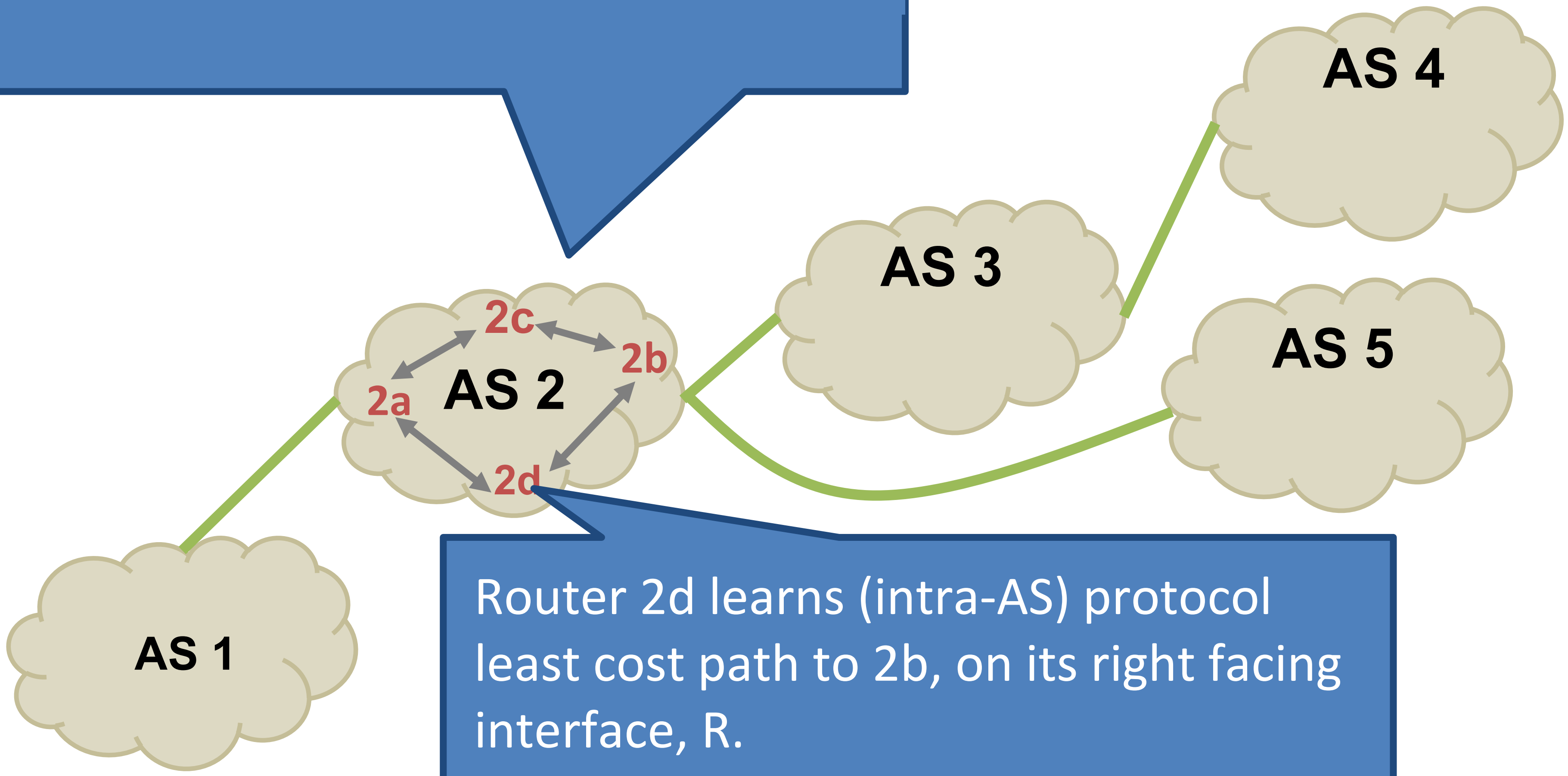
# Building the forwarding table in router 2d, for path to AS4

Suppose router in AS2 receives a datagram destined for AS4.



# Building the forwarding table in router 2d, for path to AS4

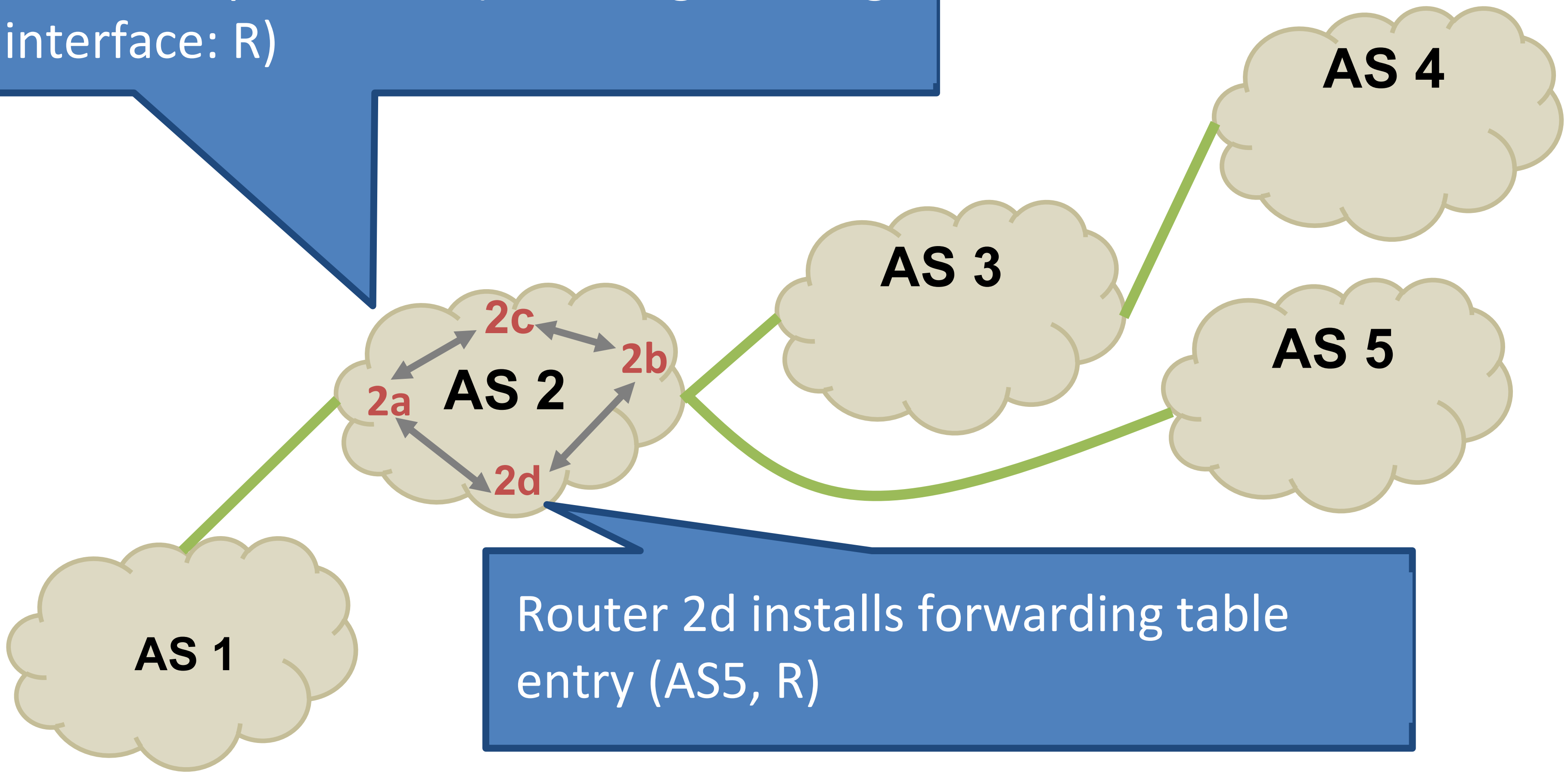
AS2 learns (Inter-AS) protocol that AS4 is reachable through AS3 (via gateway 2b).




Router 2d learns (intra-AS) protocol least cost path to 2b, on its right facing interface, R.

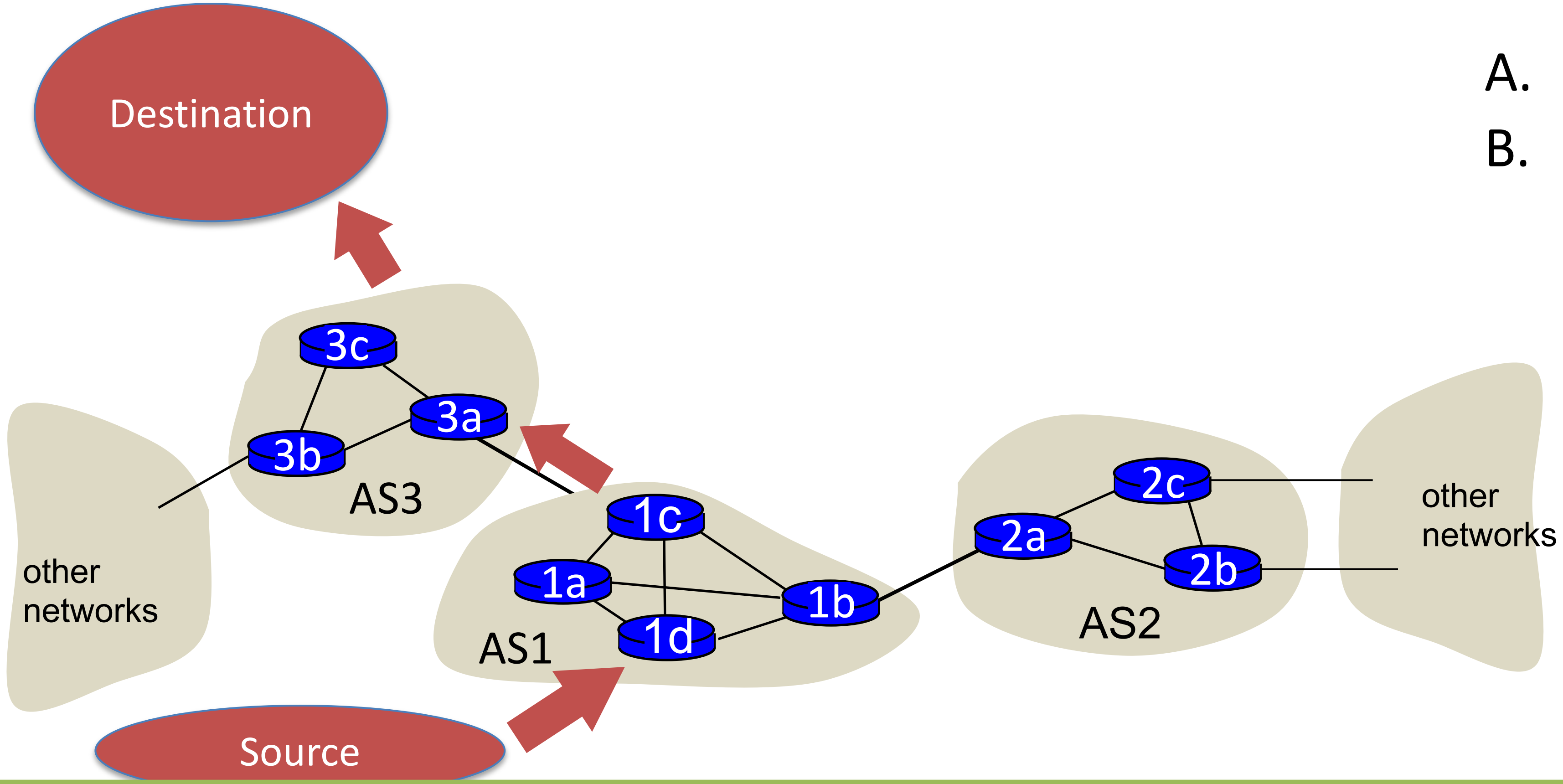
# Building the forwarding table in router 2d, for path to AS 5

Router 2d learns (intra-AS) protocol least cost path to 2b (on it's right facing interface: R)



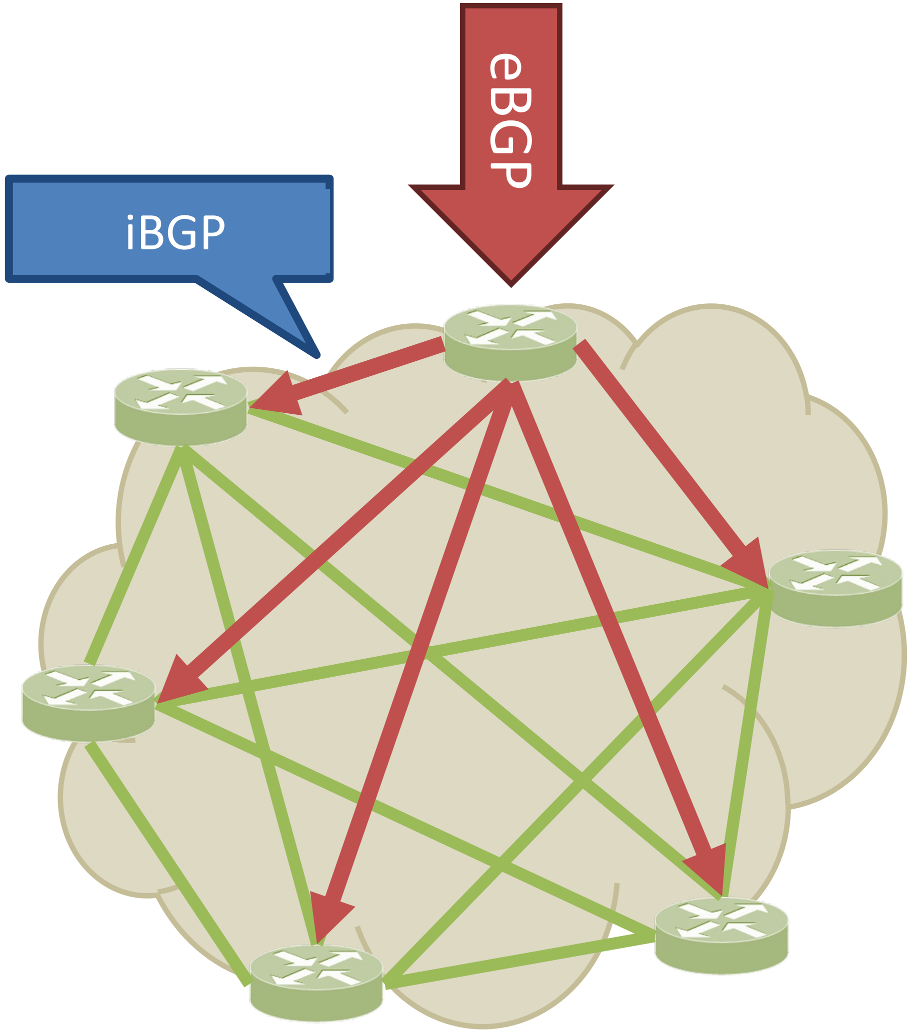

Router 2d installs forwarding table entry (AS5, R)

# Border routers: exchange AS reachability, Internal routers: exchange intra-AS reachability., Is this sufficient to route from source to destination?



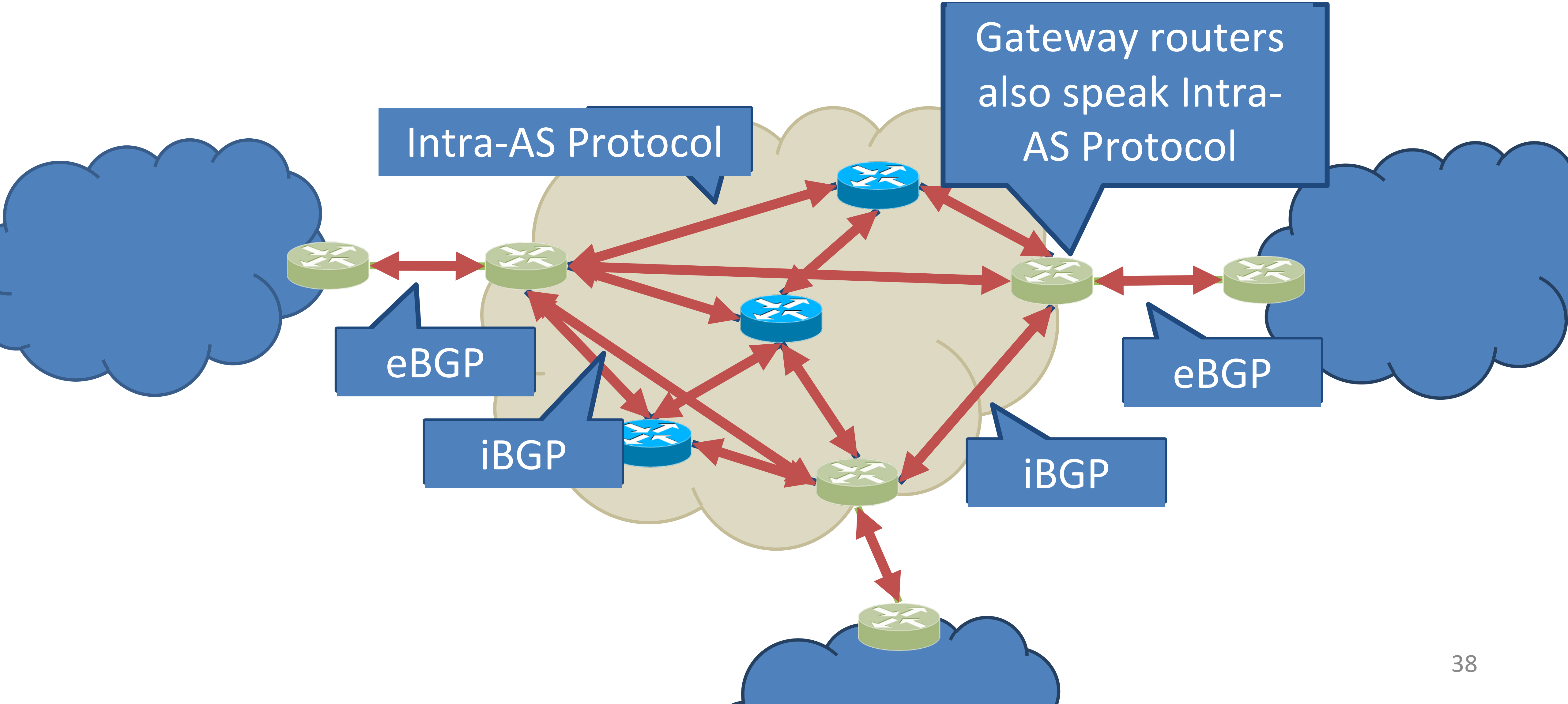

A. Yes
B. No

# Internet inter-AS routing: BGP




- Question: why do we need iBGP?
  - OSPF does not include BGP policy info
  - Prevents routing loops within the AS
- iBGP updates do not trigger announcements

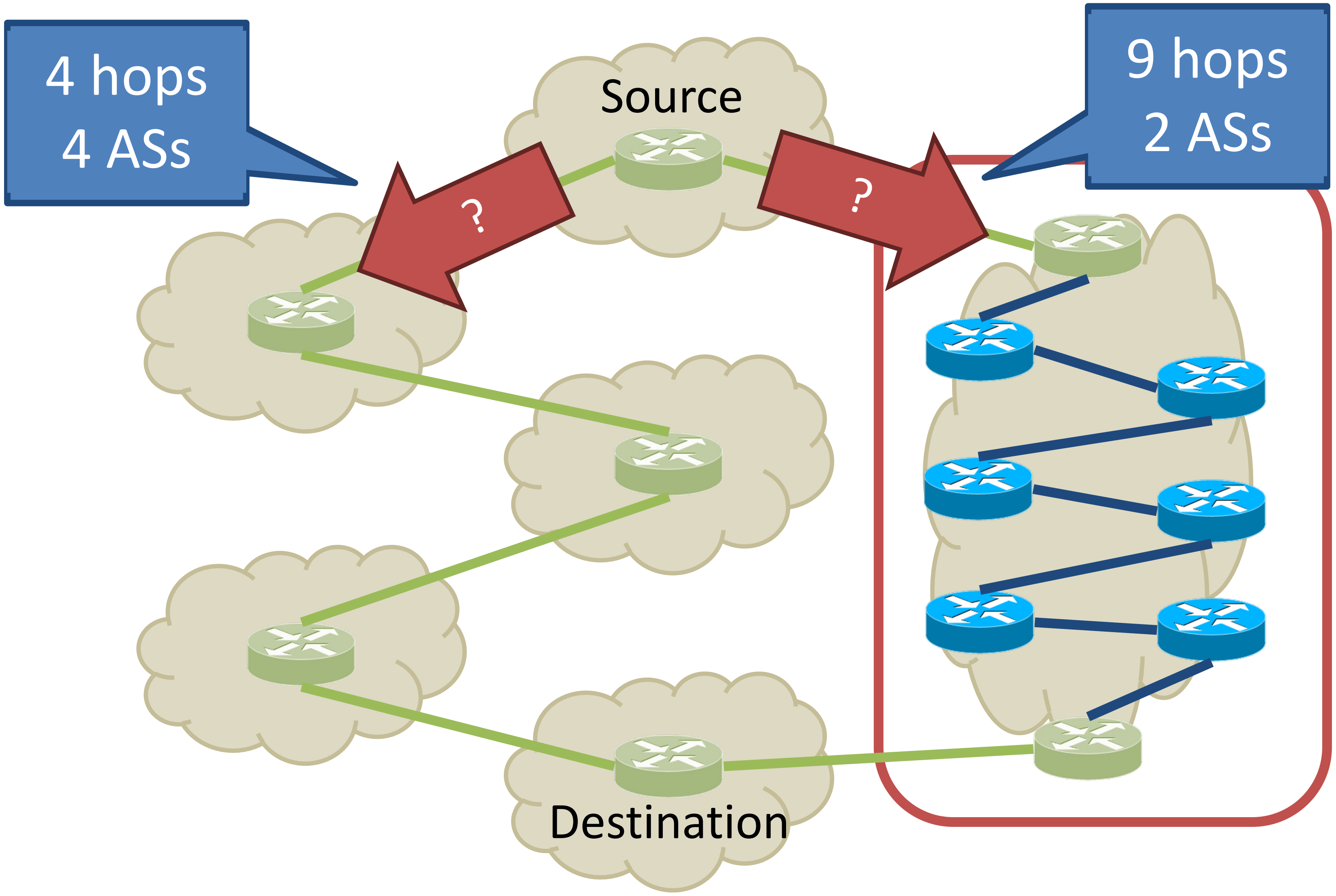
# Internet inter-AS routing: BGP



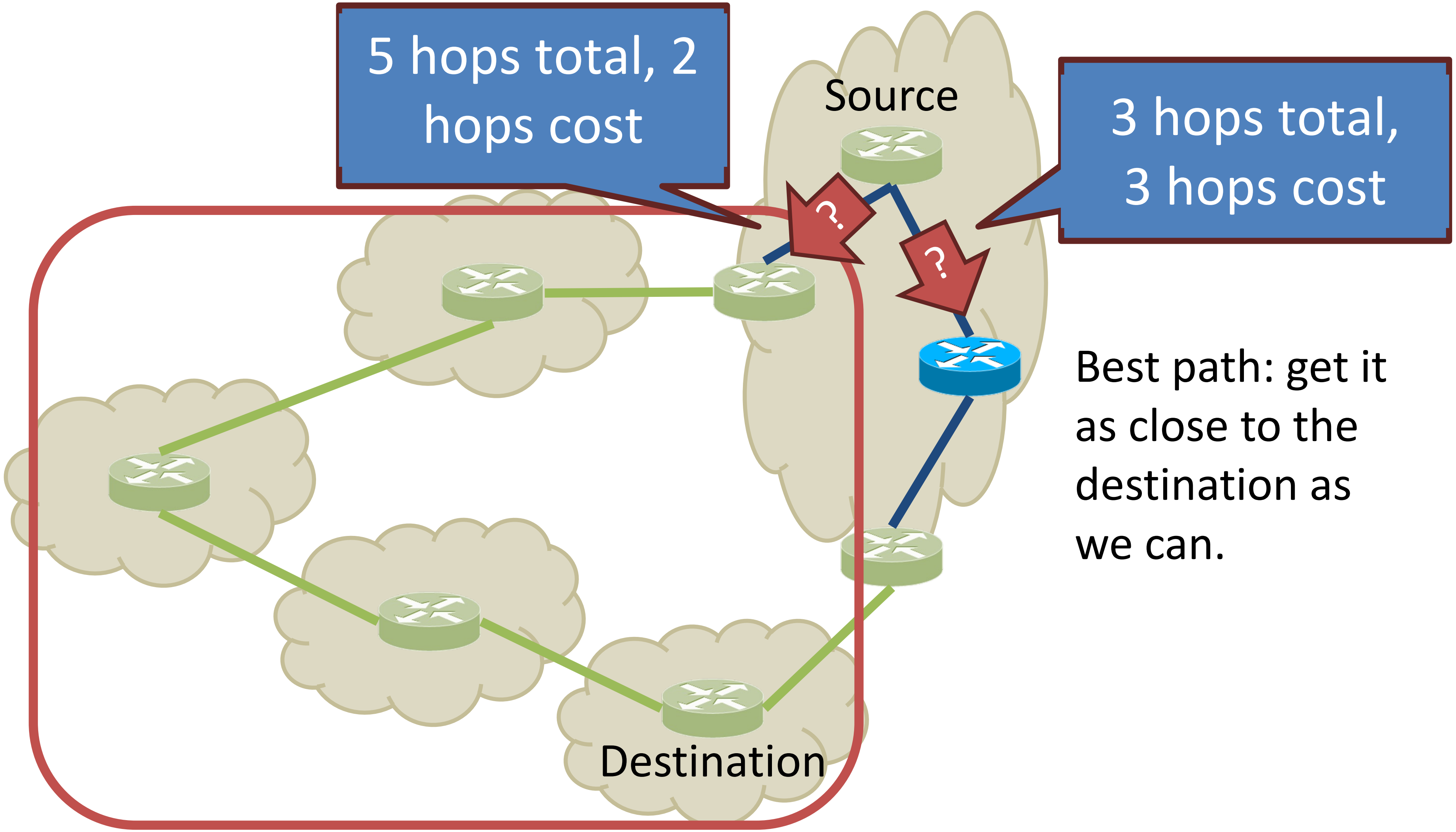
# Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol):
  *The* de facto inter-domain routing protocol
- BGP provides each AS a means to:
  - external BGP: obtain subnet reachability information from neighboring ASs.
  - internal BGP: propagate reachability information to all AS-internal routers.
  - determine "good" routes to other networks based on reachability information and policy.
- Allows a subnet to advertise its prefix to the rest of the Internet

# Shortest AS Path != Shortest Path



# Hot Potato Routing: get rid of packets ASAP!




Best path: get it as close to the destination as we can.

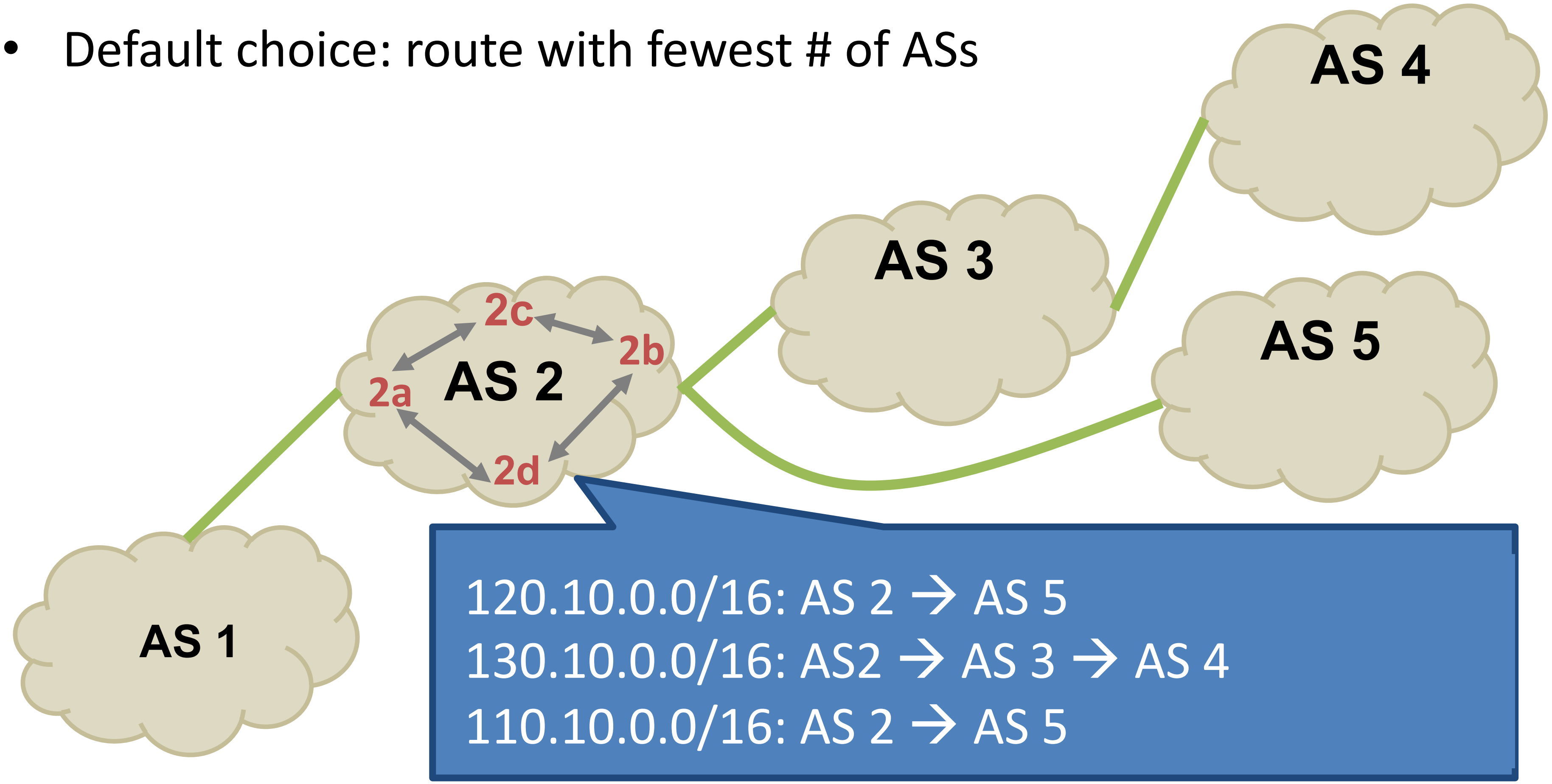
# Route Selection Summary

| | |
|---|---|
| **Highest Local Preference** | **Enforce relationships** |
| **Shortest AS Path**<br>**Lowest MED**<br>**Lowest IGP Cost to BGP Egress** | **Traffic engineering** |
| **Lowest Router ID** | **When all else fails, break ties** |

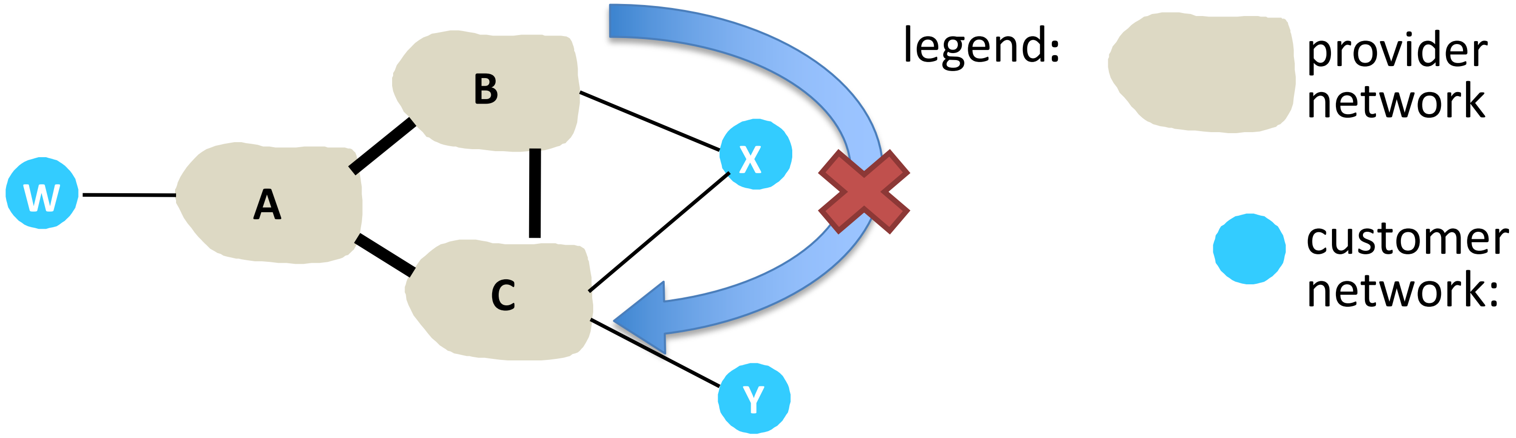
# Path Vector Protocol



- AS-path: sequence of ASs a route traverses
  - Like distance vector, plus additional information
- Used for loop detection and to apply policy
- Default choice: route with fewest # of ASs

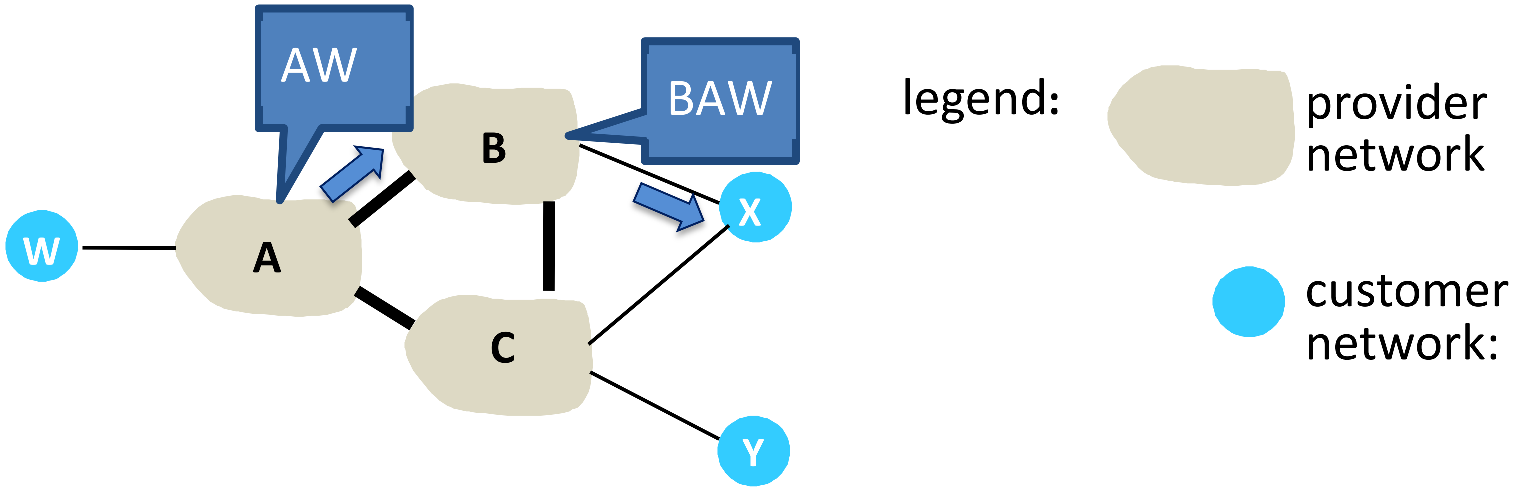


# BGP routing policy



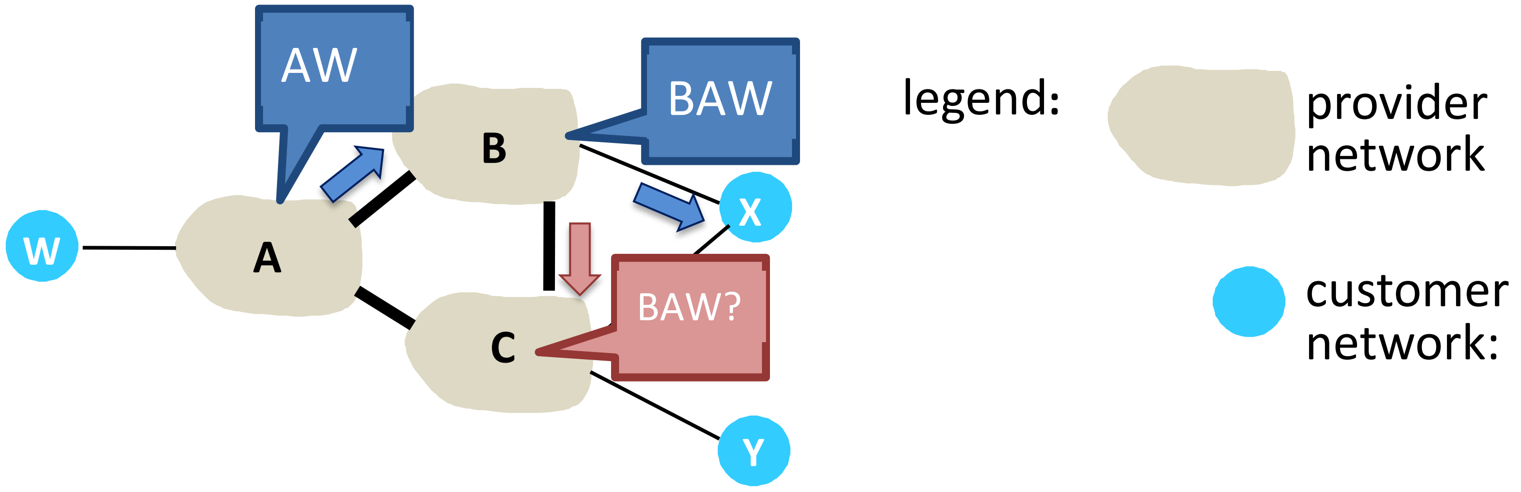

- A,B,C are provider networks
- X,W,Y are customers of the providers
- X is dual-homed: attached to two networks (B and C)
  - X does not want to route from B via X to C
  - .. so X will not advertise to B a route to C

# BGP routing policy




- A advertises path AW  to B
- B advertises path BAW to X

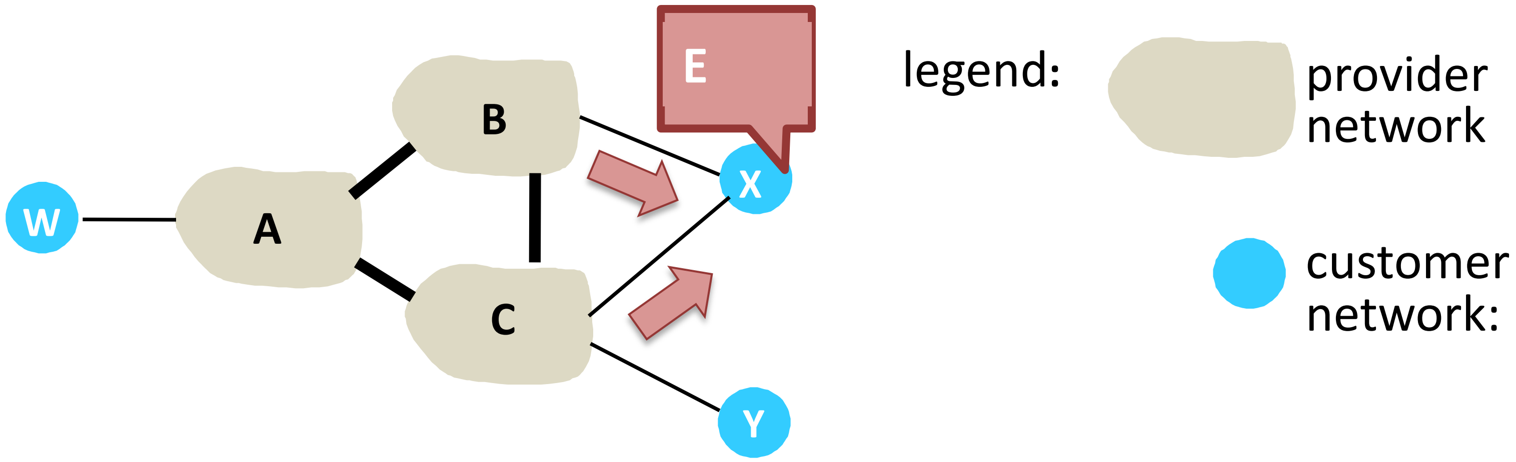
# BGP routing policy: Should B advertise path BAW to C?




Should B advertise path BAW to C?

- B gets no “revenue” for routing CBAW since neither W nor C are B’s customers
- B wants to force C to route to w via A
- B wants to route *only* to/from its customers!

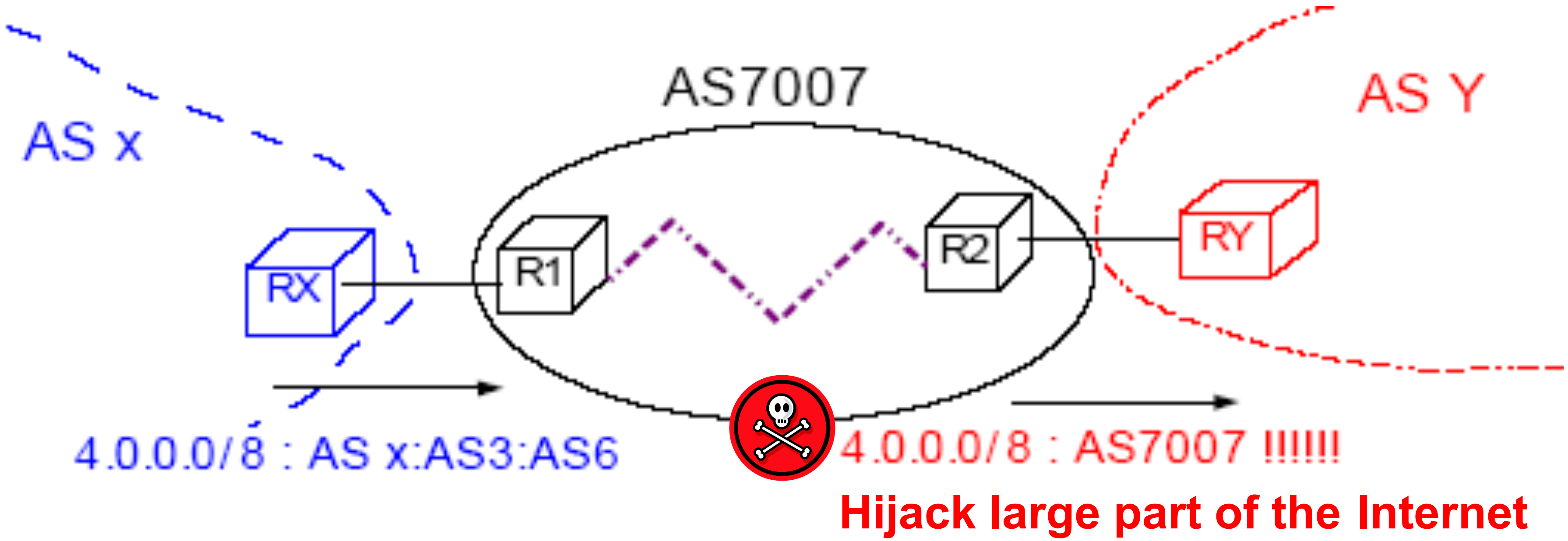
# BGP routing policy gone wrong




- x advertises a path to E (that it is not connected to).
- all traffic starts to flow into x from B and C!

# Faulty redistribution can be dangerous!

- AS7007 incident (April, 1997):



# Summary

- As we've seen before (DNS), a hierarchy can help manage state storage constraints.
  - intra-AS routing: lots of info about local routes
  - inter-AS routing: less info about far away routes
- BGP: the inter-AS routing protocol for the Internet
  - Decisions often contractual
- BGP advertises AS prefixes, including:
  - entire path of ASes along the way
  - which border router heard the advertisement (Next Hop)

# Additional Info: Inter-Domain Routing Challenges

- BGP4 is the only inter-domain routing protocol currently in use world-wide
- Issues?
  - Lack of security
  - Ease of misconfiguration
  - Poorly understood interaction between local policies
  - Poor convergence
  - Lack of appropriate information hiding
  - Non-determinism
  - Poor overload behavior

# Additional Info:
# Lots of research into how to fix this

- Security
  - BGPSEC, RPKI
- Misconfigurations, inflexible policy
  - SDN
- Policy Interactions
  - PoiRoot (root cause analysis)
- Convergence
  - Consensus Routing
- Inconsistent behavior
  - LIFEGUARD, among others

# Additional Info
# Why are these still issues?

- Backward compatibility
- Buy-in / incentives for operators
- Stubbornness

Very similar issues to IPv6 deployment

# Additional Info: Why Network Reliability Remains Hard

- Visibility
  - IP provides no built-in monitoring
  - Economic disincentives to share information publicly

- Control
  - Routing protocols optimize for policy, not reliability
  - Outage affecting your traffic may be caused by distant network

- Detecting, isolating and repairing network problems for Internet paths remains largely a slow, manual process

# Net Neutrality

- how an ISP should share/allocation its resources
  - protecting innovation, free speech, and competition on the Internet

- Example: Comcast didn't like BitTorrent, started injecting RSTs into user TCP streams.

- Scarier example: You like Netflix, but your ISP has their own video service. They degrade (or block) Netflix service unless you pay $$$.

# Net Neutrality

**Cases for:**

- End to end principle
- Prevent customer extortion
- Allow for innovation

Google, Microsoft, Yahoo, Amazon, eBay

**Cases against:**

- ISP owns their network
- Asymmetric application bandwidth usage
- We shouldn't legislate the Internet, it moves too fast

Cisco, many ISPs

# Today

- We’ve seen the behavior of TCP/IP, and routers
- We’ve joked about the option of marking packets as “urgent”
  - As a lone user, your cries for urgency will likely be ignored by one or more ISPs on the Internet
- False implication: All traffic is treated equally.

# Scenarios

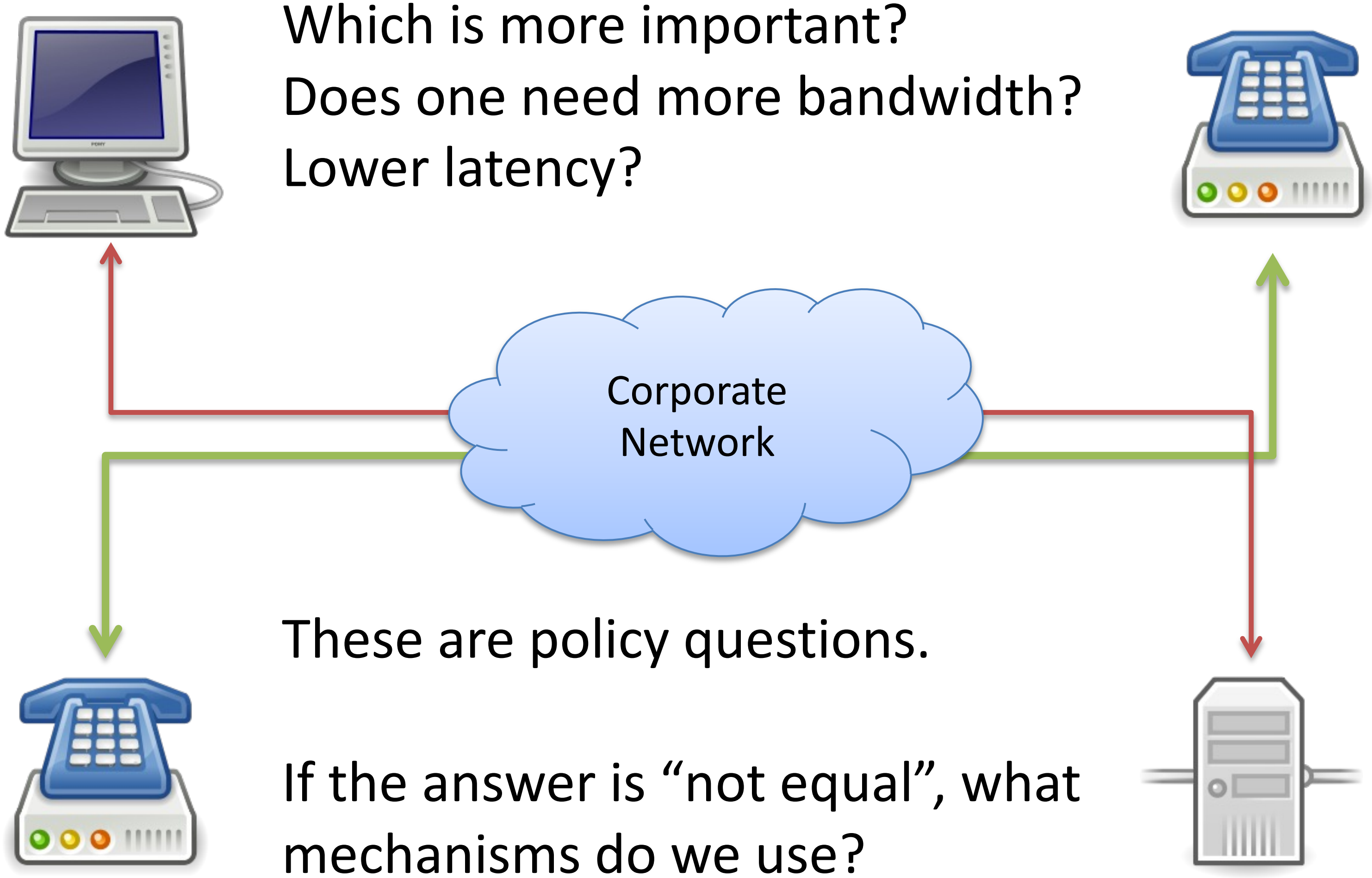
- Things we can do at the network layer to:
  - Treat traffic differently
  - Improve congestion control

- You own a private network
  - Corporate network
  - Data center
  - ISP

- You want to provide better performance to:
  - More important services
  - Customers who pay more

# Example 1: Corporate Phones

Which is more important?
Does one need more bandwidth?
Lower latency?

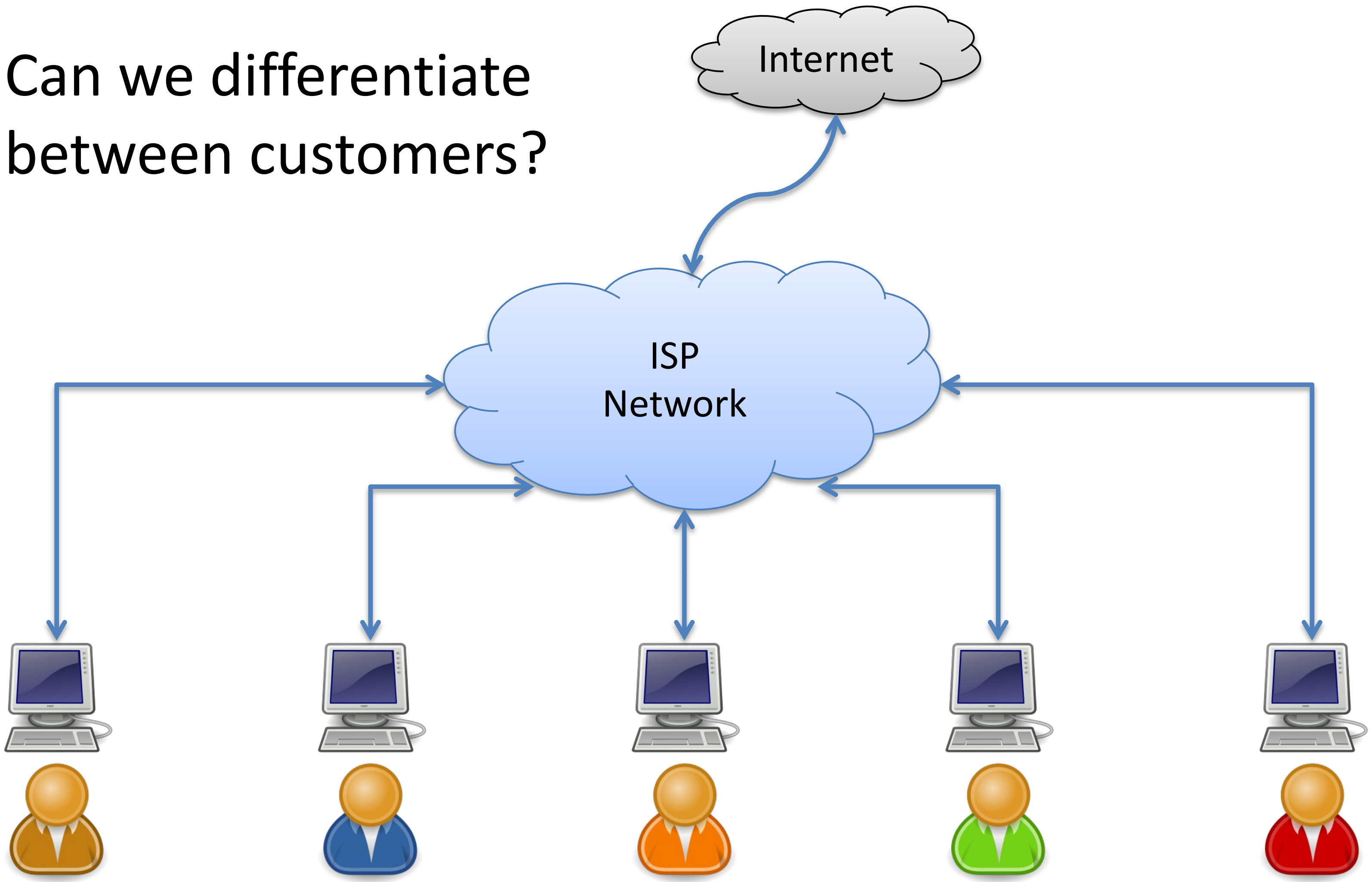



These are policy questions.

If the answer is “not equal”, what mechanisms do we use?

# Example 2: ISP Customers

Can we differentiate between customers?

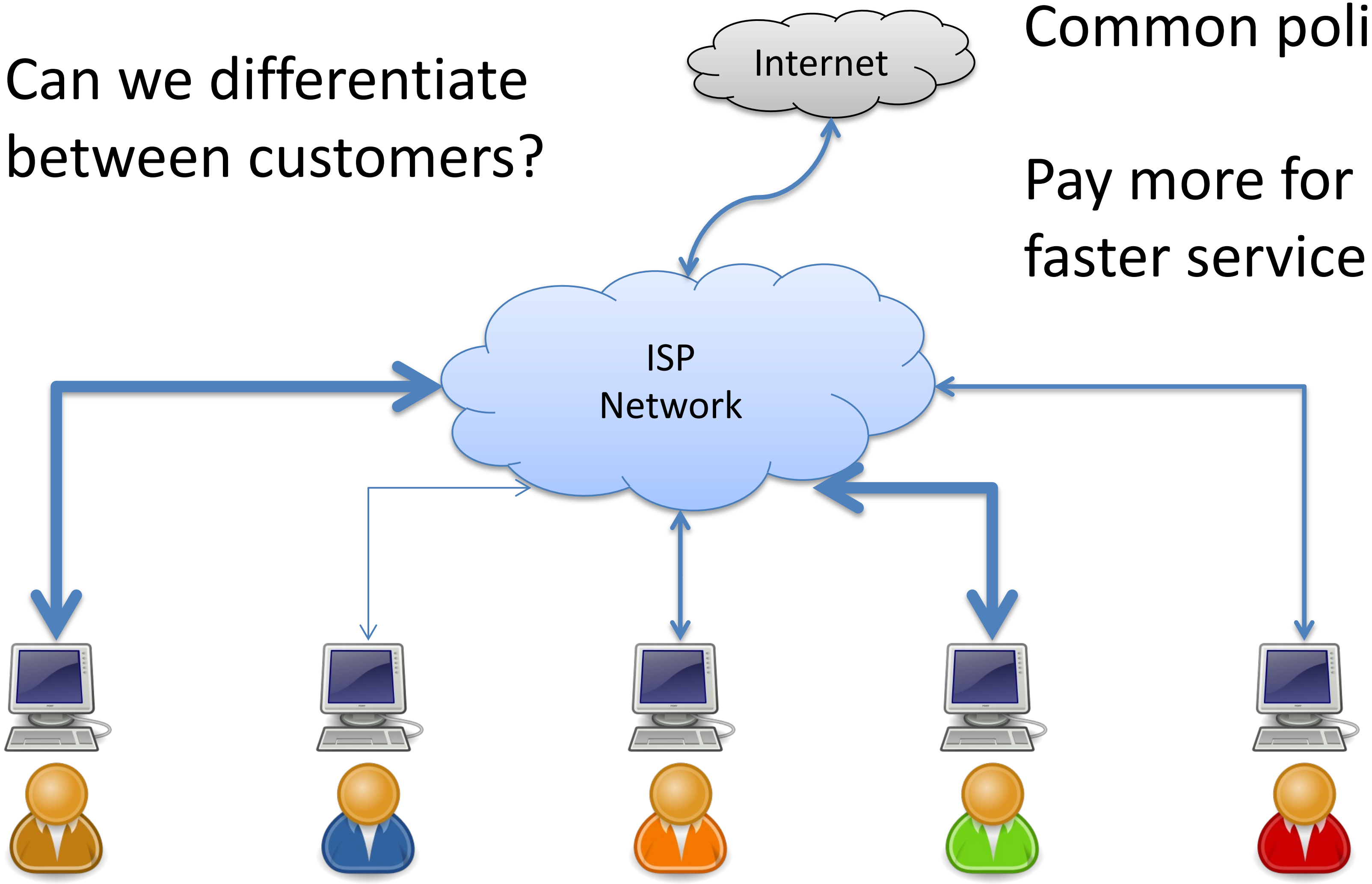


# Example 2: ISP Customers

Can we differentiate between customers?

Common policy:

Pay more for faster service!



# How might we enforce these types of policies?

A. Require that end-hosts police their traffic.

B. Change how routers queue traffic.

C. Ask users nicely to comply with policy.

D. Enforce policies some other way.

E. There is nothing we can do.

# How might we enforce these types of policies?

A. Require that end-hosts police their traffic.

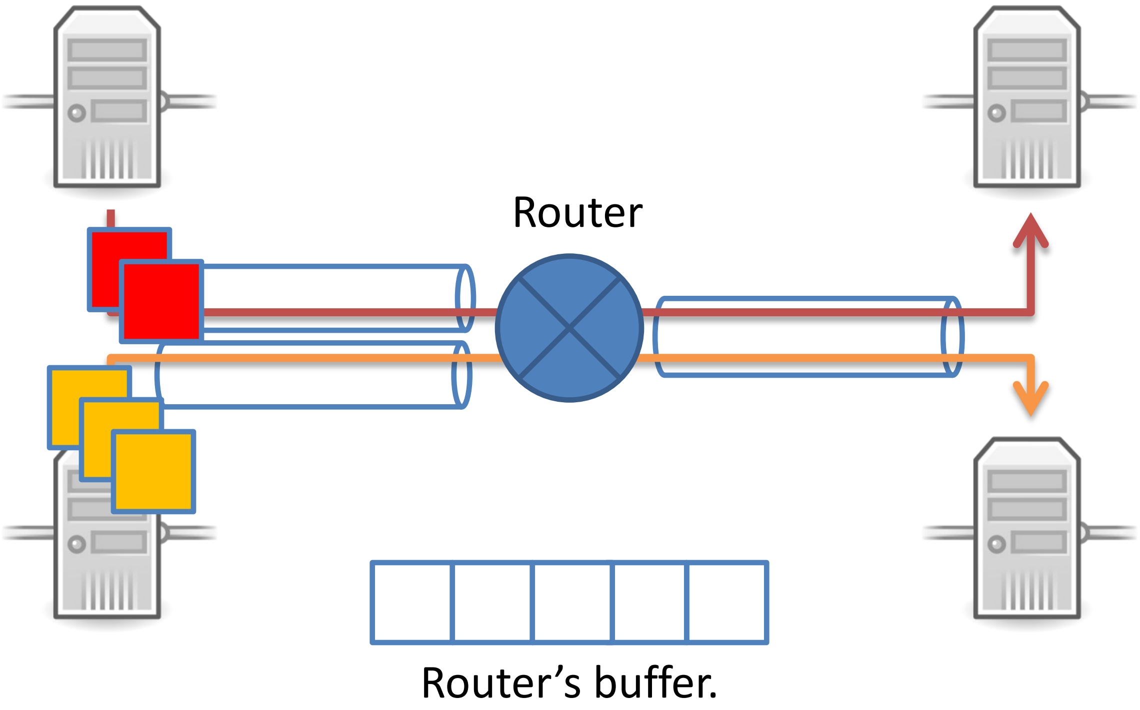
B. Change how routers queue traffic.

C. Ask users nicely to comply with policy.

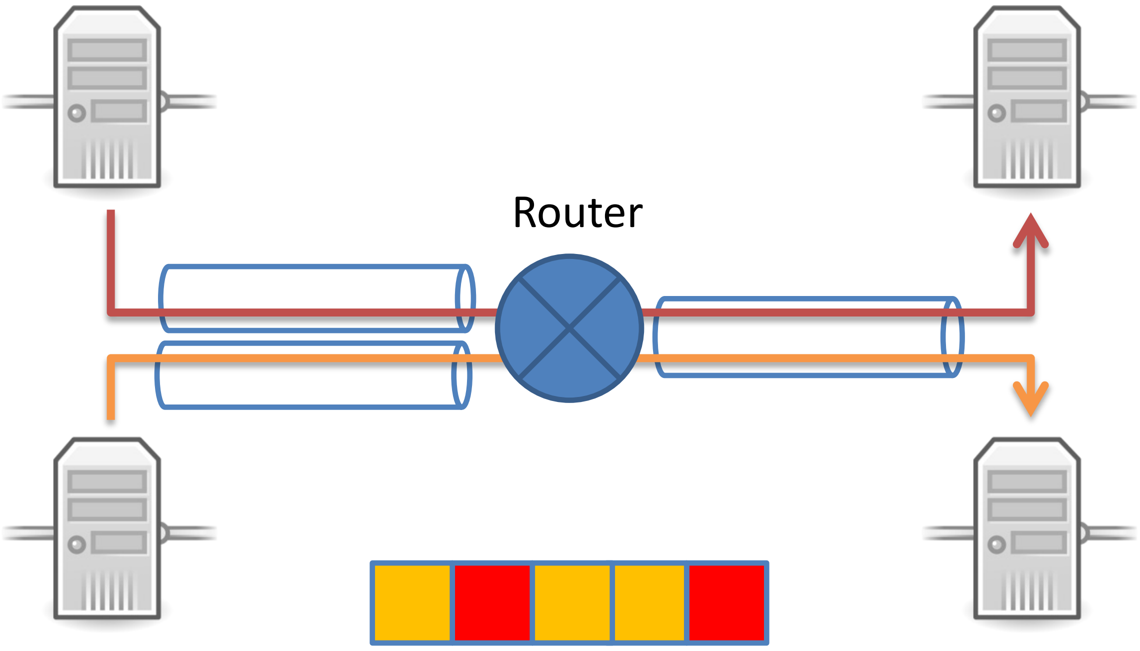
D. Enforce policies some other way.

E. There is nothing we can do.

# Recall Queueing



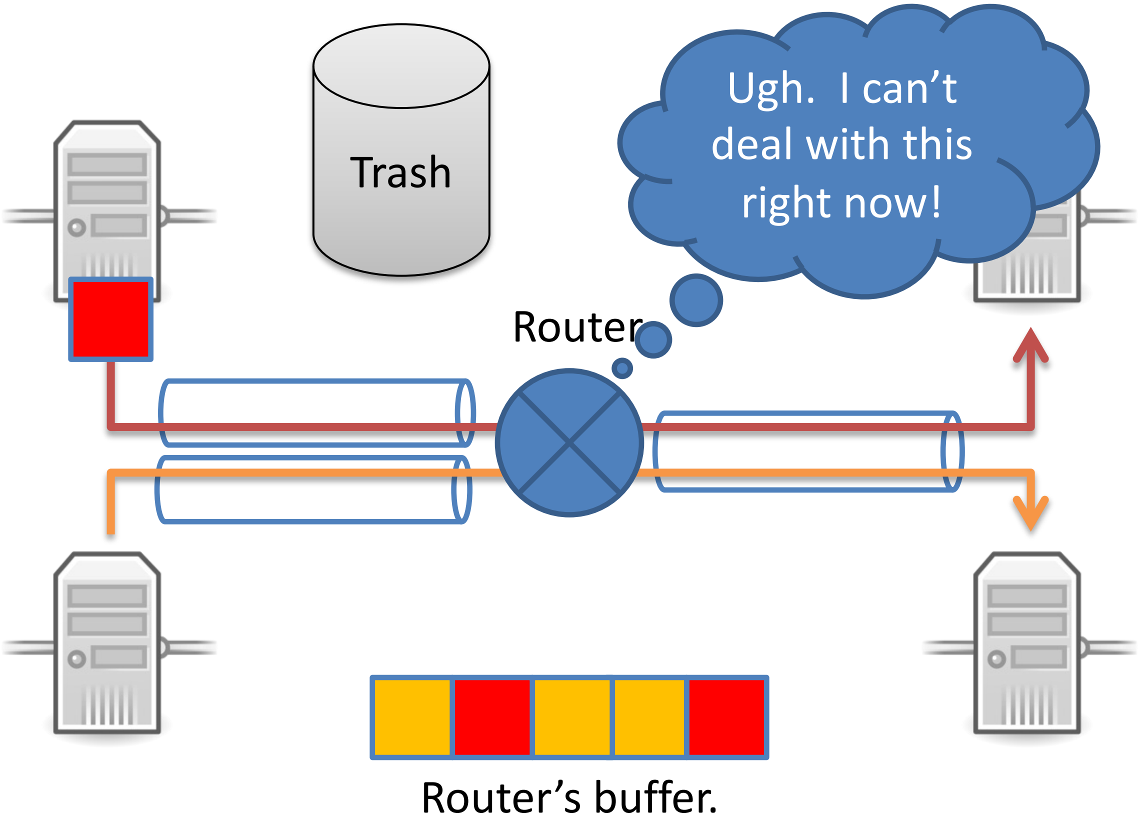
# Recall Queueing




Router's buffer.

Incoming rate is faster than outgoing link can support.

# Recall Queueing




Incoming rate is faster than outgoing link can support.

# Basic Buffer Management

- FIFO + drop-tail
  - Simplest choice
  - Used widely in the Internet
- FIFO (first-in-first-out)
  - Traffic queued in first-come, first-served fashion
- Drop-tail
  - Arriving packets get dropped when queue is full
- Important distinction:
  - FIFO: queueing (scheduling) discipline
  - Drop-tail: drop policy

# FIFO/Drop-Tail Problems

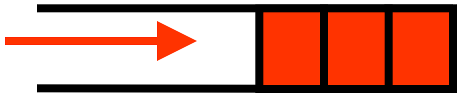
- Doesn't differentiate between flows/users
- No policing: send more, get more service
- Leaves responsibility of congestion control completely to the edges (e.g., TCP)
- Synchronization: hosts react to same events

# Quality of Service (QoS)

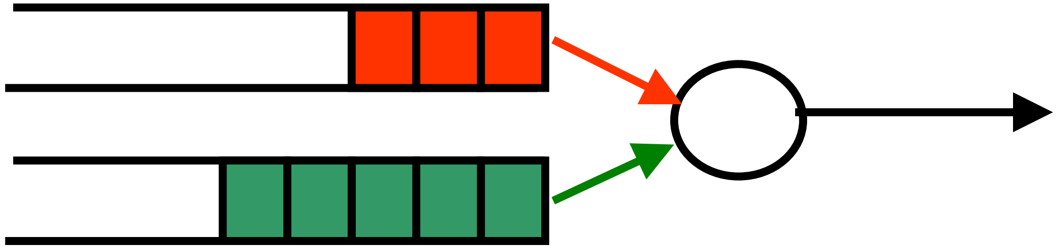
- QoS is a broad topic!  We're going to discuss:
  - Mechanism for differentiating users/flows
  - Mechanism for enforcing rate limits
  - Mechanism for prioritizing traffic

# QoS: Quality of Service

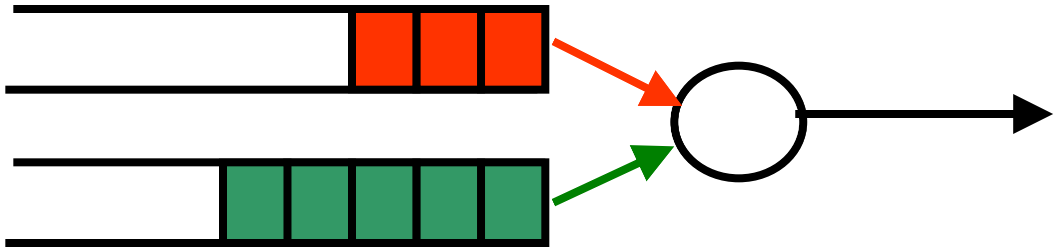
- Drop-tail FIFO queue
  - Packets served in the order they arrive
  - … and dropped if queue is full



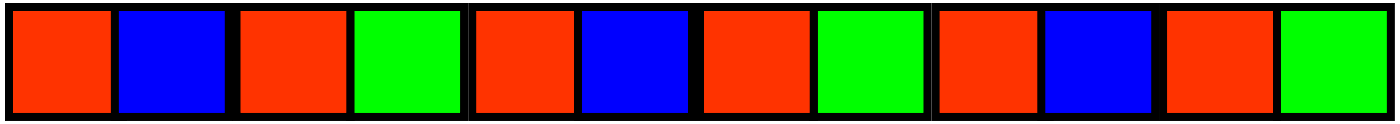
- Random Early Detection (RED)
  - When the buffer is nearly full
  - … drop or mark some packets to signal congestion
- Multiple classes of traffic
  - Separate FIFO queue for each flow or traffic class
  - … with a packet scheduler to arbitrate between them



# Packet Scheduling

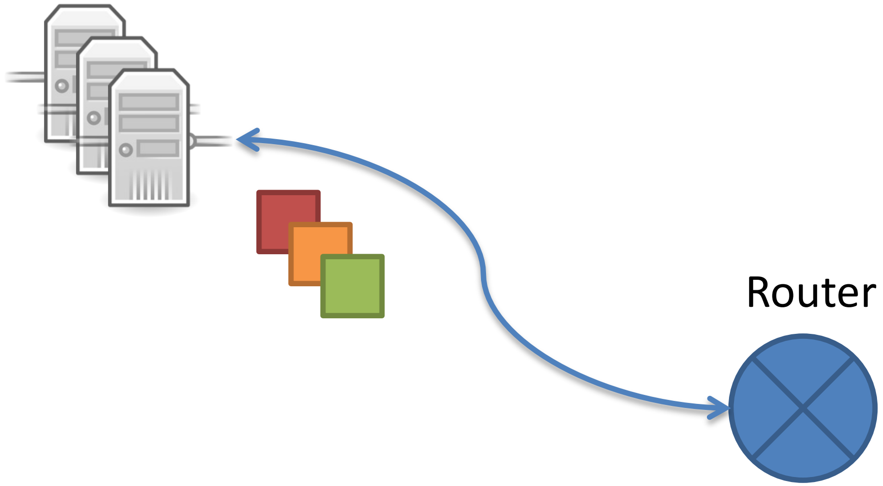


- Strict priority
  - Assign an explicit rank to the queues
  - … and serve the highest-priority backlogged queue
- Weighted fair scheduling
  - Interleave packets from different queues
  - …in proportion to weights



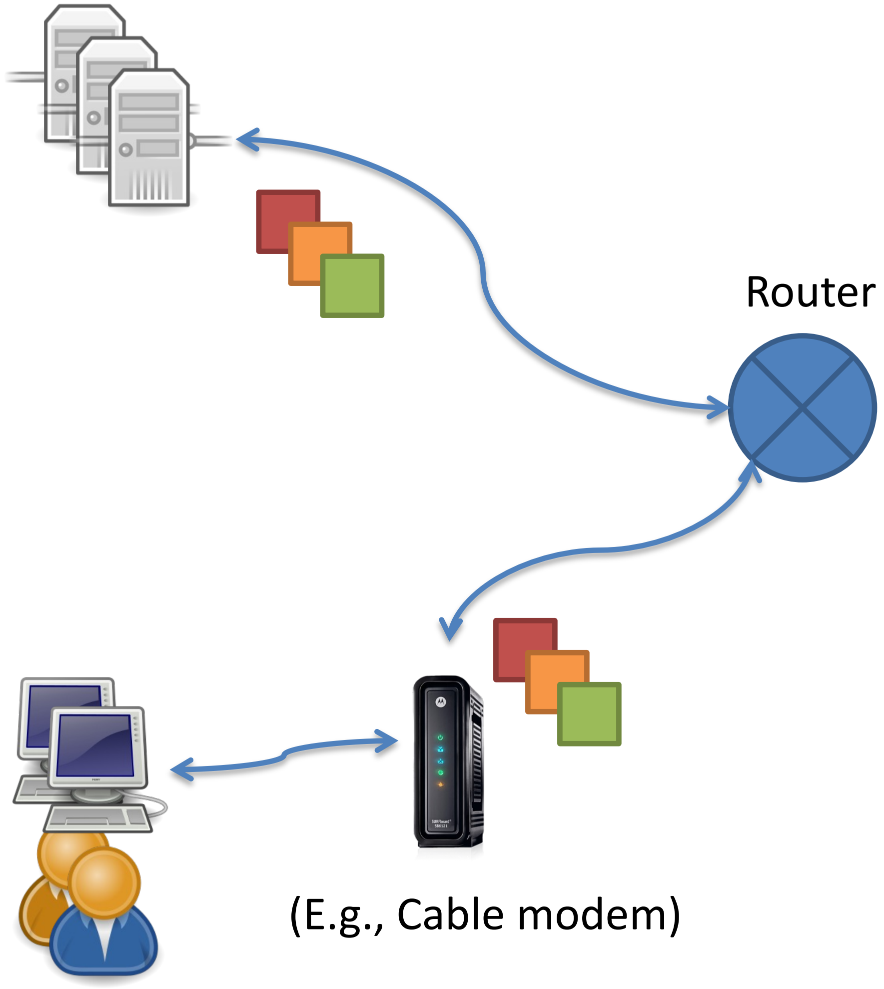
50% red, 25% blue, 25% green

# Differentiating Users




- If you control end hosts:
  - Mark packets in OS according to policy.
- Take advantage of IP's class of service or options header fields

# Differentiating Users



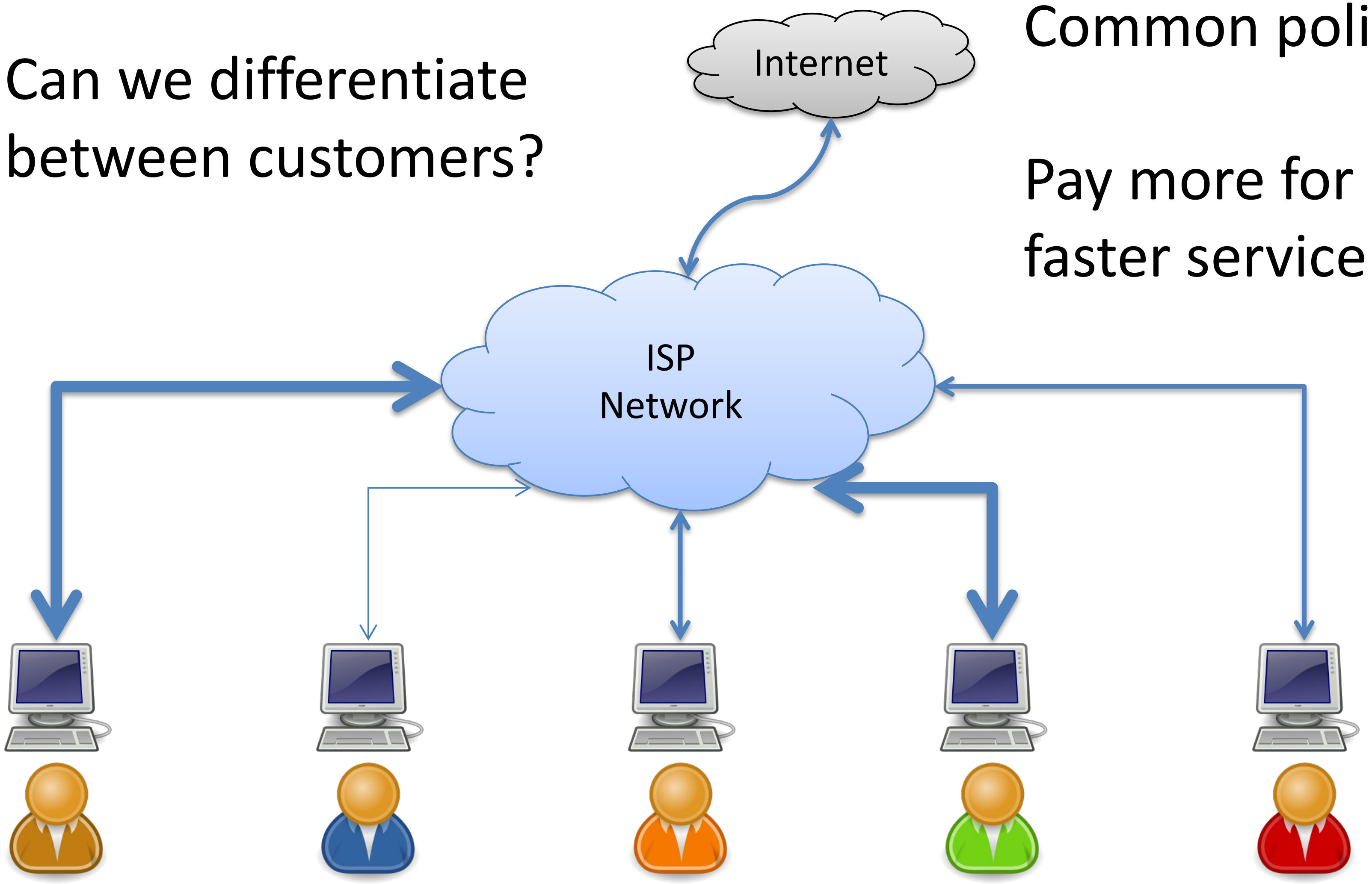

- If you control end hosts:
  - Mark packets in OS according to policy.
- Take advantage of IP's class of service or options header fields
- Otherwise:
  - Introduce an intermediate device you trust.

# Example 2: ISP Customers

Can we differentiate between customers?

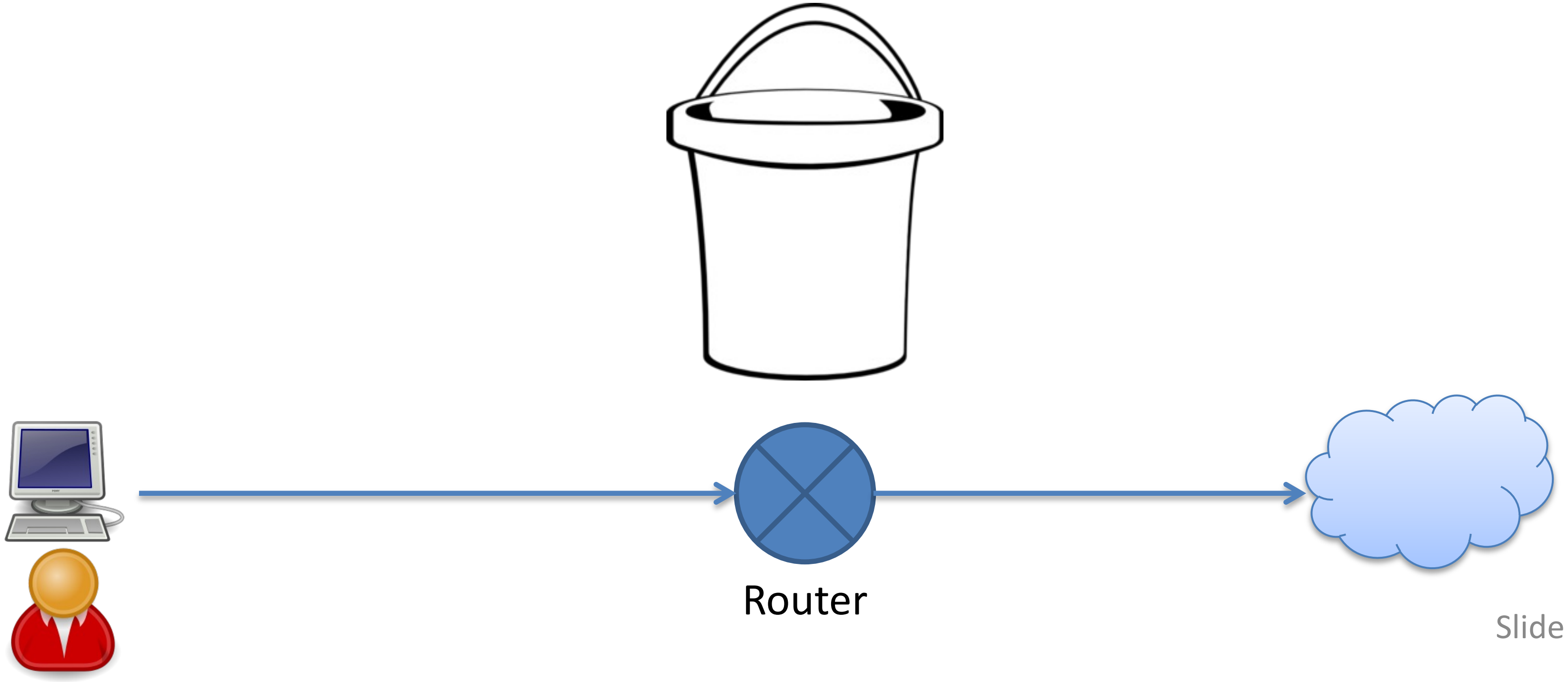
Common policy:

Pay more for faster service!



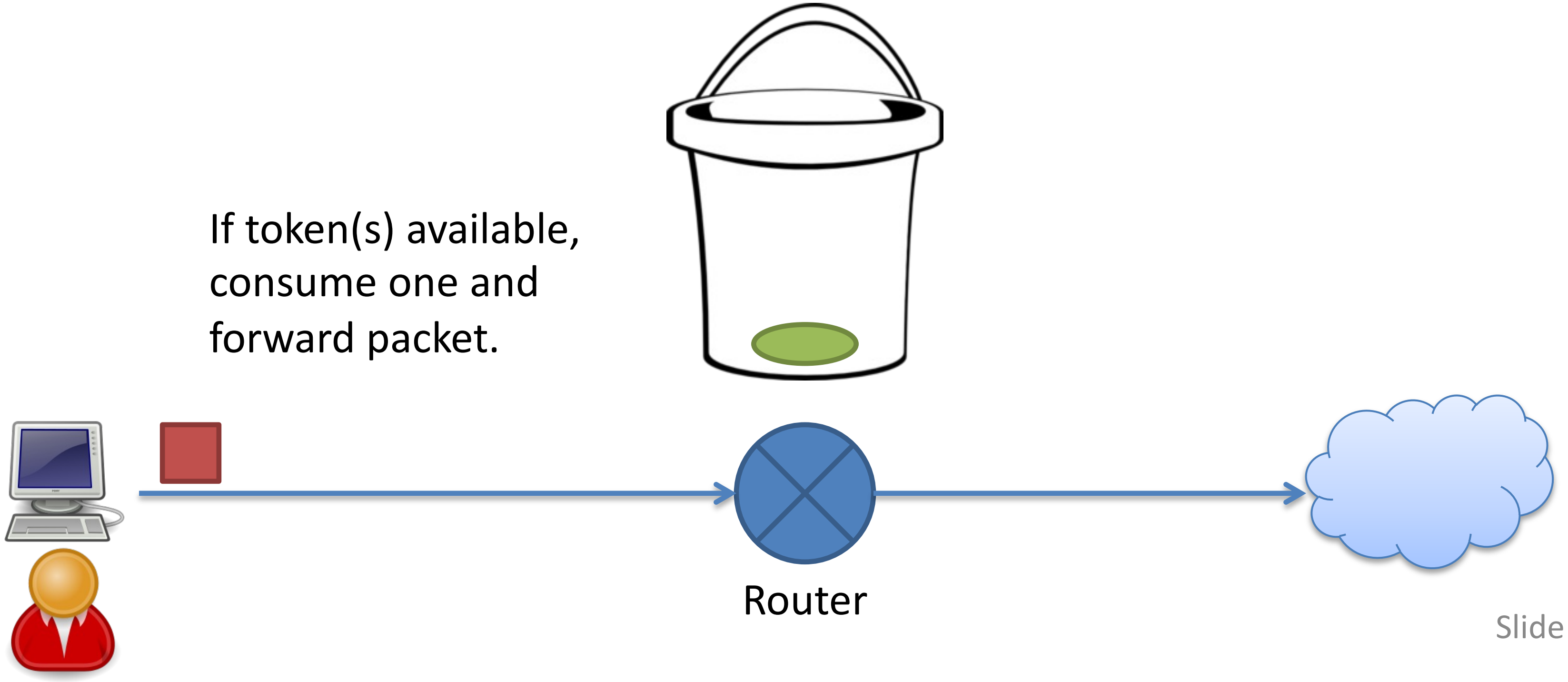
# Enforcing (Policing) Rate Limits

- Example: the red user gets at most 10 Mbps
- Solution: Token bucket



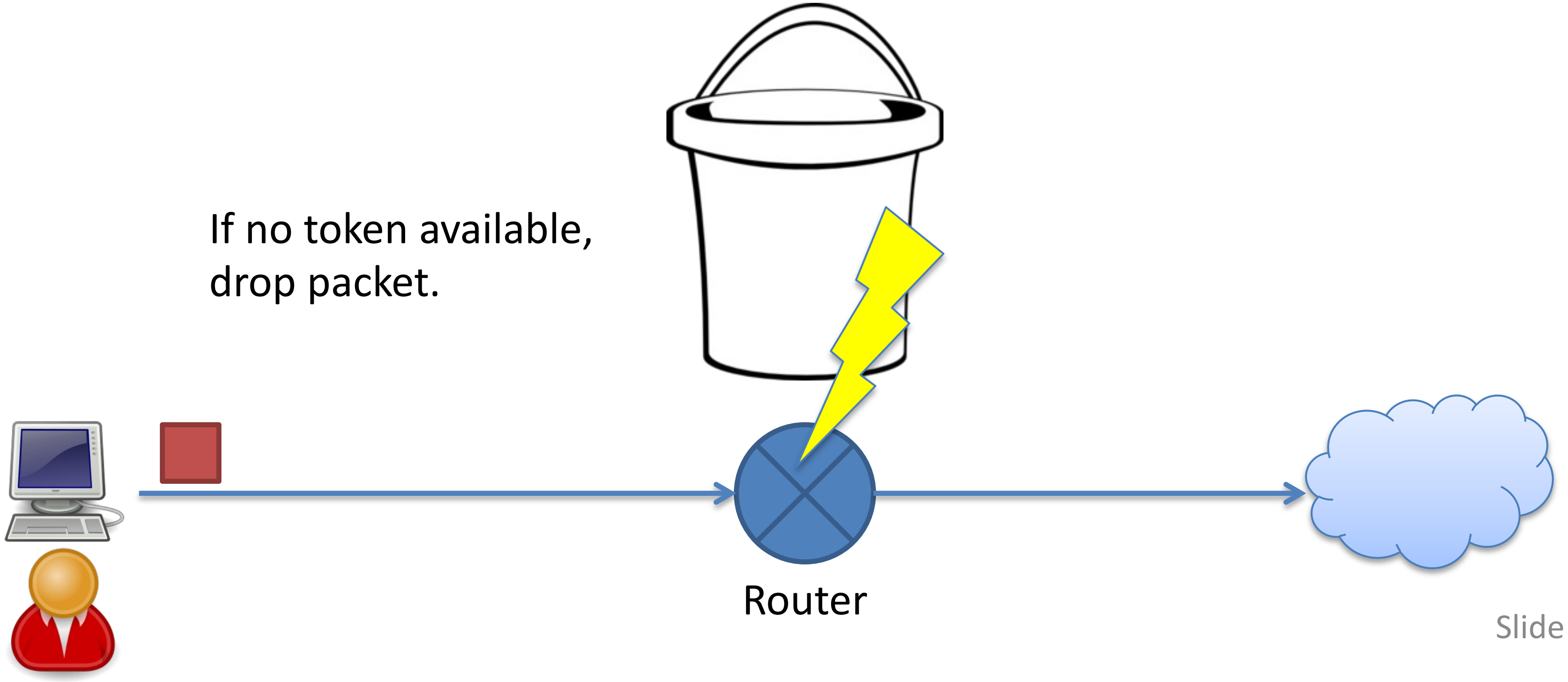
# Enforcing (Policing) Rate Limits

- Example: the red user gets at most 10 Mbps
- Solution: Token bucket



# Enforcing (Policing) Rate Limits

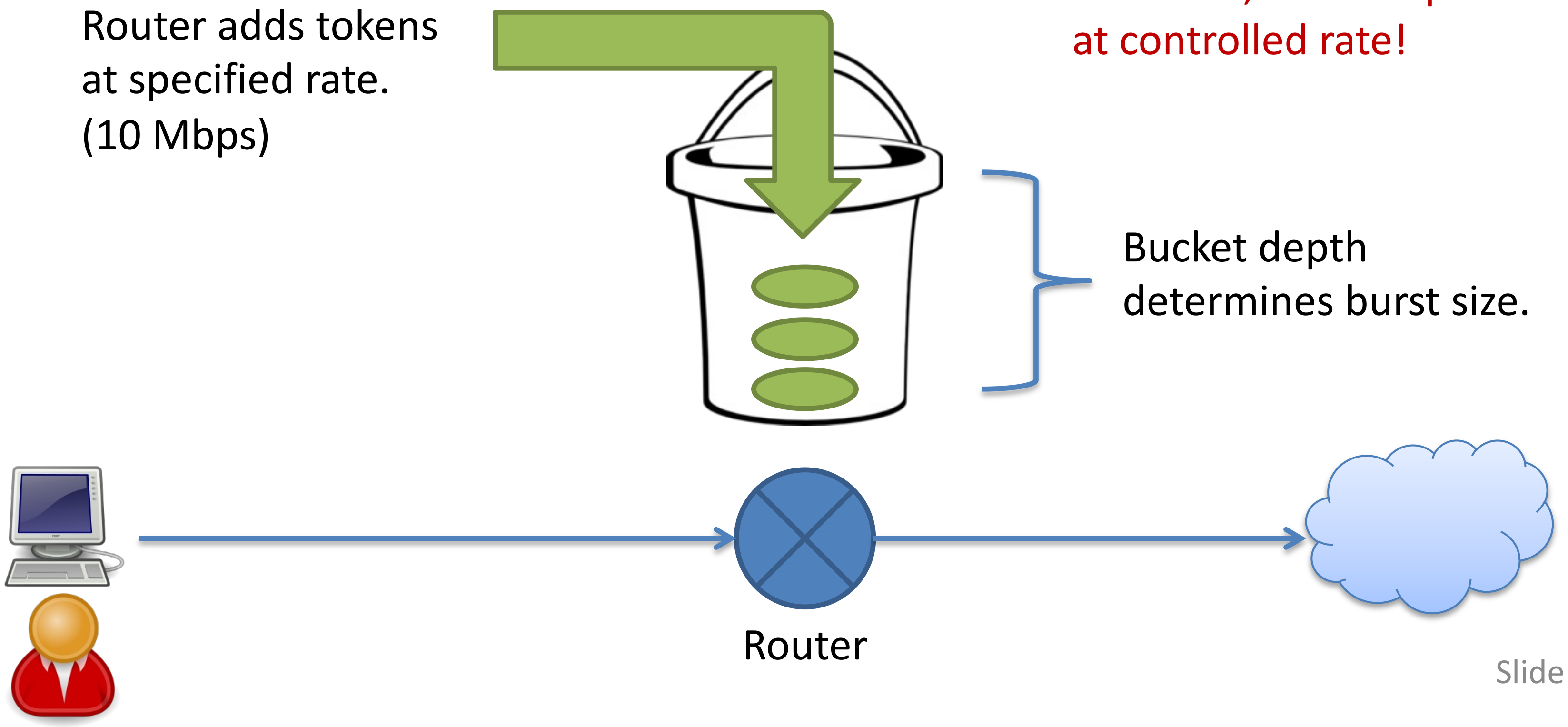
- Example: the red user gets at most 10 Mbps
- Solution: Token bucket



# Enforcing (Policing) Rate Limits

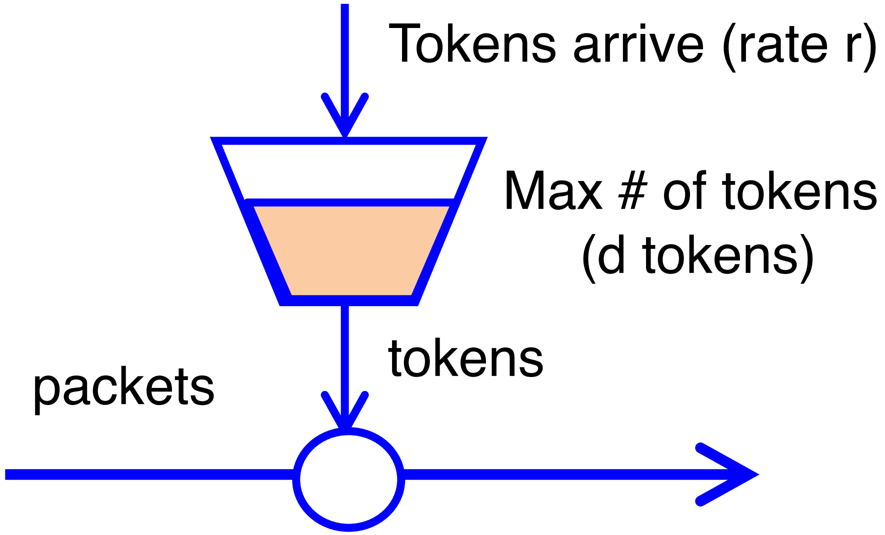
- Example: the red user gets at most 10 Mbps
- Solution: Token bucket

No matter how fast user sends, limited by number of tokens, which replenish at controlled rate!



# Traffic Shaping

- Force traffic to conform with a profile
  - To avoid congesting downstream resources
  - To enforce a contract with the customer
- Leaky-bucket shaping
  - Can send at rate r and intermittently burst
  - Parameters: token rate r and bucket depth d



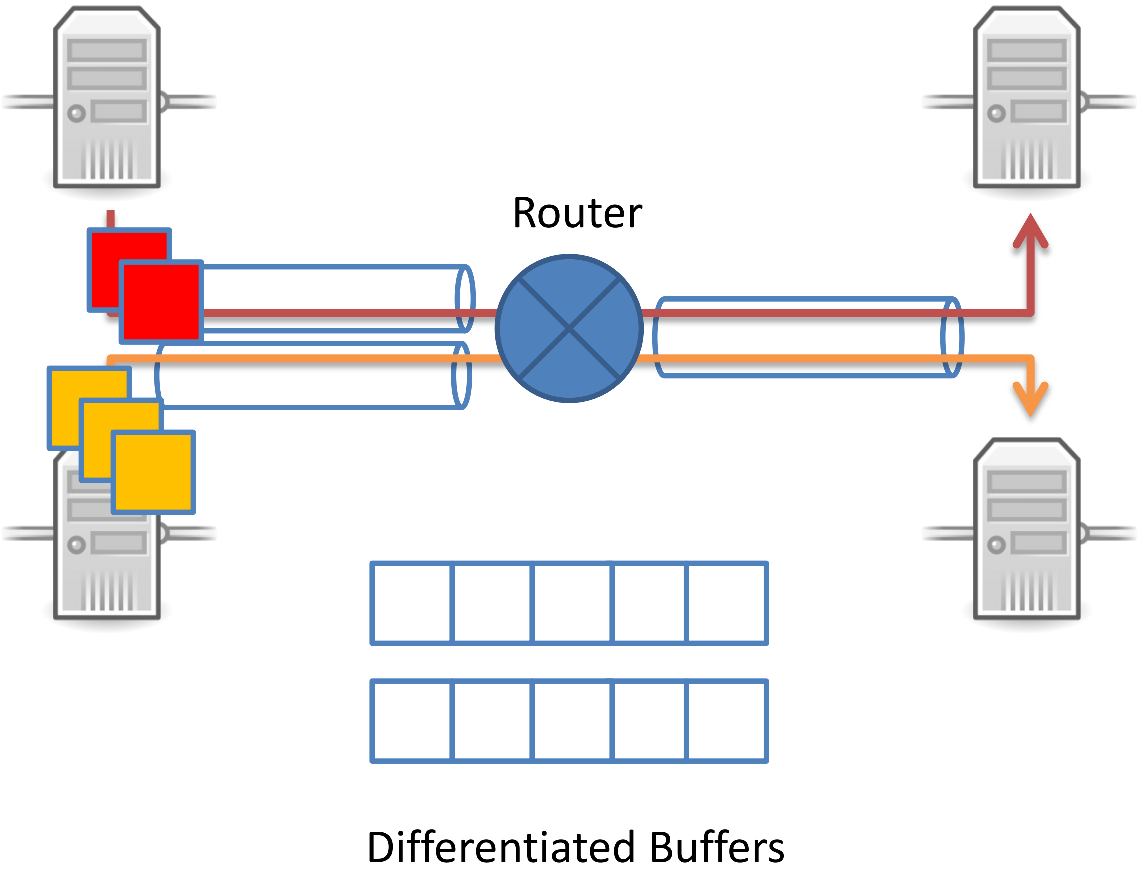

A leaky-bucket shaper for each flow or traffic class

# Prioritizing

- Been to a theme park recently?

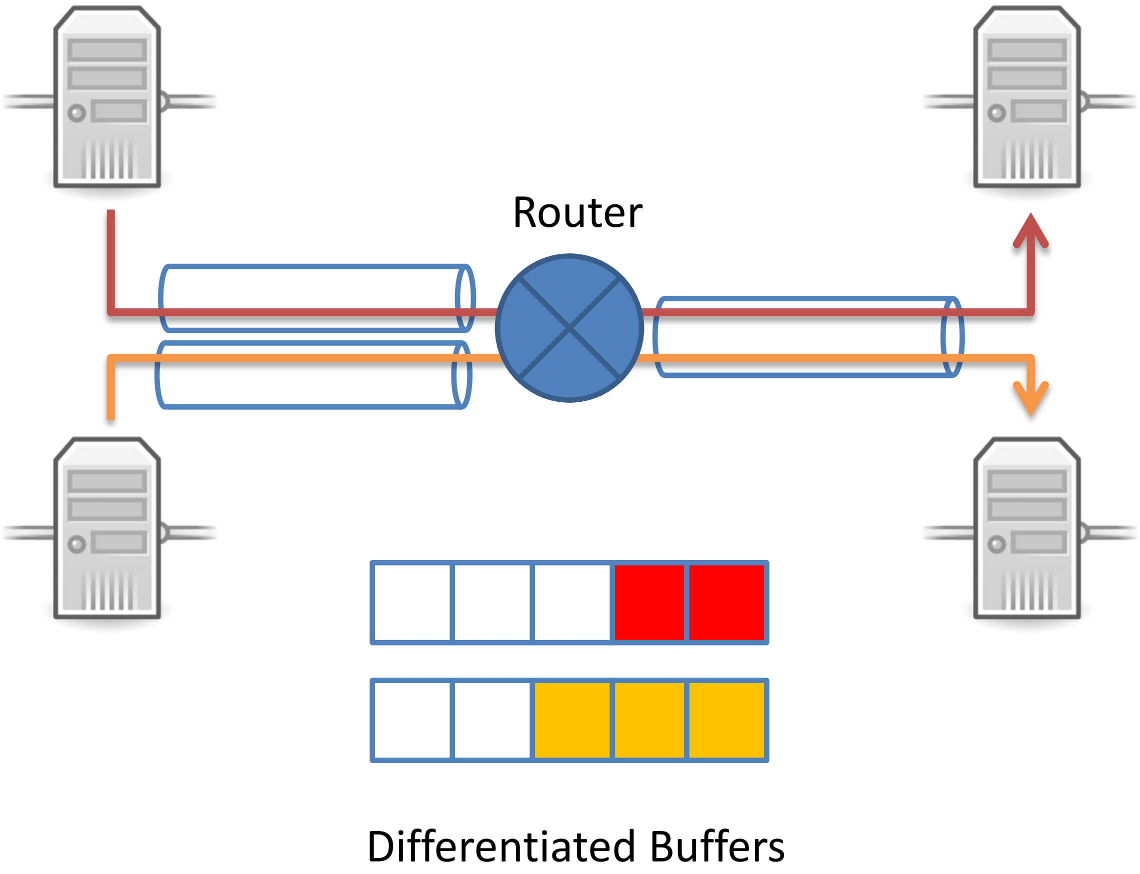
# Prioritizing Traffic

- Designate multiple classes of traffic.



# Prioritizing Traffic

- Weight queues differently.



# Weighted Fair Queueing

- Suppose orange is more important than red.

- Policy: Always empty orange's queue first.
  - Problem: Red might starve!

- Policy: Always allow 1 red packet for every N orange packets.
  - Ratio is known as weight.

# FIFO/Drop-Tail Problems

- Doesn't differentiate between flows/users
- No policing: send more, get more service
- Leaves responsibility of congestion control completely to the edges (e.g., TCP)
- Synchronization: hosts react to same events

QoS

AQM

# Active Queue Management

- Design active router queue management to aid congestion control

- Why?
  - TCP at end hosts have limited vantage point
  - Routers see actual queue occupancy

- "Hint": TCP will still do congestion control
  - We can try to help it out in the network!

# How might we take advantage of TCP's behavior to help it discover congestion in the network?

A. Drop packets, even when they could be sent.

B. Hold packets in the queue, even when they could be sent.

C. Send a congestion notification back to the sender.

D. Send a congestion notification to the receiver.

E. Some other mechanism.

# Random Early Detection (RED)

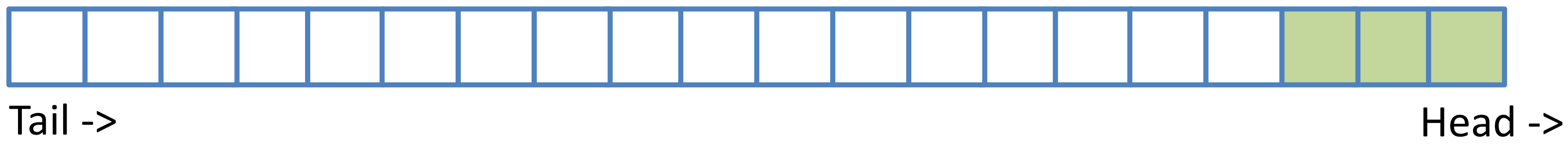
- Goal: Prevent congestion before it's a problem
- Assume hosts respond to lost packets
- Avoid window synchronization
  - Randomly mark packets
- Avoid bias against bursty traffic

# RED Algorithm

- Maintain running average of queue length
- If $avg < min_{th}$ do nothing
  - Low queuing, send packets through
- If $avg > max_{th}$, drop packet
  - Protection from misbehaving sources
- Else drop/mark packet in a manner proportional to queue length
  - Notify sources of incipient congestion

# RED

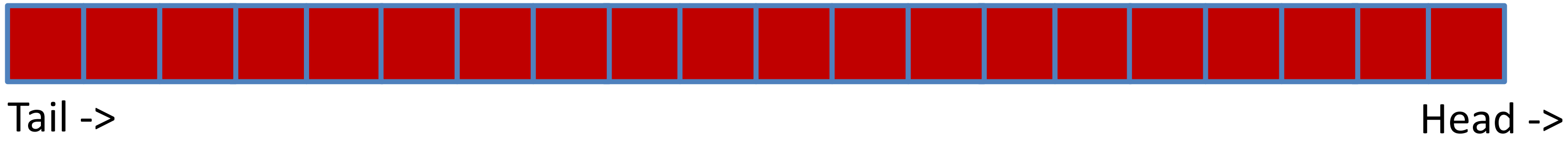
- Router queue:




- Mostly empty? Don't drop.

# RED

- Router queue:




- Mostly full?  Drop new packets.

# RED

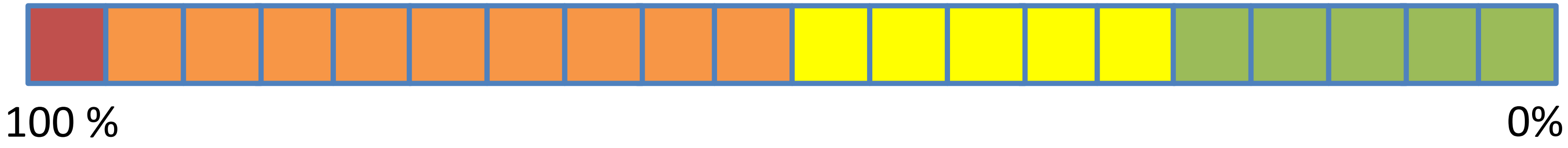
- Router queue:



- In the middle? Drop proportionally to how full the queue is!

# RED

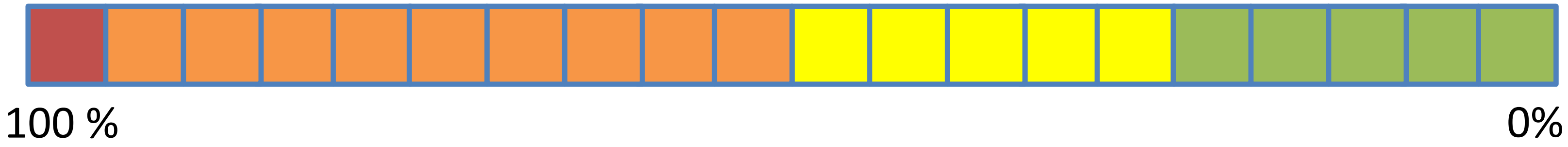
- Drop probability:




- In the middle? Drop proportionally to how full the queue is!

# ECN

- ~~Drop~~ Mark probability:




- Explicit congestion notification: Instead of dropping, set a header field, which gets returned to sender in ACK.

- Treat marked packets as “congestion events”

# Explicit congestion notification (ECN)

- TCP deployments often implement network-assisted congestion control:
  - two bits in IP header (ToS field) marked by network router to indicate congestion
- policy to determine marking chosen by network operator

# Explicit congestion notification (ECN)

TCP deployments often implement *network-assisted* congestion control:

- congestion indication carried to destination
- destination sets ECE bit on ACK segment to notify sender of congestion
- involves both IP (IP header ECN bit marking) + TCP (TCP header C,E bit marking)